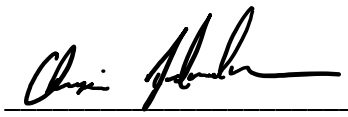
Created by

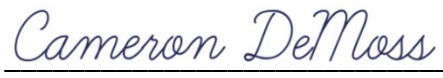
Chris Holschuh, Aidan Burns, and Cameron Demoss

Submitted to
the Faculty of the School of Information Technology
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Technology



Chris Holschuh — 4/25/2020 Date

Aidan Burns — 4/25/2020 Date



Cameron Demoss — 4/25/2020 Date

Faculty Advisor: Ryan Moore — 4/25/2020 Date

University of Cincinnati
College of
Education, Criminal Justice, and Human Services

April 2020

Prepared by:

Aidan Burns, Project Manager and UI/UX Lead
Chris Holschuh, Software Developer
and Cameron Demoss, Security Analyst

Students of
University of Cincinnati
College of Education, Criminal Justice, and Human Services School
of Information Technology
cech.uc.edu/it

April 2020

Table of Contents

LIST OF ILLUSTRATIONS ..... **II**

TABLES: II
FIGURES: II

ABSTRACT **1**

INTRODUCTION: **2**

PROBLEM: 2
SOLUTION: 3
PROJECT GOALS & METHODOLOGY: 4
OVERVIEW: 5

DISCUSSION: **6**

PROJECT CONCEPT: 6
DESIGN OBJECTIVES: 6
USER PROFILE: 7
USE CASE DIAGRAM: 8

TECHNICAL ELEMENTS: **9**

METHODOLOGY & TECHNICAL APPROACH 9
NETWORK (HARDWARE AND INFRASTRUCTURE) 9
APPLICATION 10
SECURITY 12

VISUALS **13**

USER INTERFACE: 13

TESTING: **16**

PROJECT MANAGEMENT **25**

BUDGET: 25
DESIGN OBJECTIVES/DELIVERABLES: 26
PROJECT SCHEDULE: 27
WORK BREAKDOWN STRUCTURE: 27
GANTT CHART: 30
PROBLEMS ENCOUNTERED 31
FUTURE RECOMMENDATIONS 32

CONCLUSION **33**

FALL 2019: 33
SPRING 2020: 33

APPENDIX: REFERENCES **35**

List of Illustrations

Tables:

Table 1: User Profile 7
Table 2: Beacon Functionality 19
Table 3: Launch Tests 20
Table 4: Page Launch Tests 21
Table 5: Beacon Test 21
Table 6: Navigation Testing 22
Table 7: Scenario Testing 22
Table 8: Project Milestones 26
Table 9: Work Breakdown Structure 27

Figures:

Figure 1: Use Case Diagram 8
Figure 2: Main Menu 13
Figure 3: Floors Page 13
Figure 4: Maps Page 14
Figure 5: Email Pop-up 14
Figure 6: The Check-in Page 14
Figure 7: Beacon Discovery 15
Figure 8: Location Services Pop-up 15
Figure 9: The Beacon Card 15
Figure 10: Budget 25
Figure 11: Gantt Chart 30
Figure 12: Poster 36

Abstract

Have you had a chance to visit the 1819 Innovation hub yet? If not, you're probably not familiar with all the exciting opportunities and services the building has to offer. To some, the possibilities at 1819 can be quite overwhelming, but with the 1819 Mobile app, we make these opportunities more approachable. With our location based informational beacons and seamless check-in pre-registration process, you can skip to the front of the line and get right to work on your latest ideas and inventions. Using Apple's Core Location services, the power of Swift, and the latest in Bluetooth low energy beacon technology, the 1819 Mobile app provides you with up to date contextual information about key locations within the University of Cincinnati's 1819 Innovation Hub. The 1819 Mobile app ensures that you have access to the tools and information needed to succeed in your visit.

Introduction:

With its state-of-the-art technology and tools, the 1819 Innovation Hub is quickly becoming a popular place for creative minds to come and make their ideas come to life. This facility needs a way to quickly inform visitors about opportunities that are available at the Innovation Hub, and seamlessly allow guests to register ahead of time for the makerspace check-in process. Using 1819 Mobile and Bluetooth low energy iBeacons, we have made the building a more accessible resource for faculty, students, and staff.

Problem:

Currently the University of Cincinnati does not have a resource or application that aggregates data about specific aspects of the university and the areas surrounding the campus. While researching areas at the University of Cincinnati that are in need of an application to showcase their services offered, we stumbled across 1819. The 1819 Innovation Hub is a great resource for the University as a whole, with many services that are available to students, faculty and staff. However, there is a lot packed in to such a small building, which can make it difficult to find exactly what you're looking for.

After our initial meeting with 1819, we re-grouped as a team to discuss what we felt could be done to help showcase 1819 and all they have to offer. During our visit, employees were happy to share ideas that they had on how an app could improve the day to day operations of the building as well as strongly expressing some concerns they had about guests wandering aimlessly around the building. When further researching this concern, we spoke with Harry LeMaster, the technology coordinator of the building who said "We have a need for

a wayfinding tool with the possibility of automating tours. Our areas are subject to numerous tours and community members so an application leveraging way-finding beacons could guide guests through the space as well as provide knowledge of the building and its units function within it." We also spoke with Gregory Glevicky, 1819's Digital Communications Specialist, about the long lines that can sometimes form at the makerspace check in desk which we realized can be a major inconvenience for students with busy schedules. When students were asked about the services that 1819 offers, they could not list anything beyond the 3D printers that are available in the makerspace.

Solution:

Our Solution to these problems was to design a mobile application that would make this information easily accessible to anyone interested in 1819. Using Apple's Core Location services, the power of swift, and the latest in Bluetooth low energy beacon technology, the 1819 Mobile app provides users with up-to-date contextual information about key locations within the 1819 Innovation Hub. In the app, users will have access to building information whether on or off site, as well as a fun and interactive way of accessing that information on-site using the app's Beacon Discovery feature.

Instead of having students, faculty, staff and potential patrons wandering around the building asking for directions, our application makes navigating the building easy and interactive. Through the app, users will have access to detailed color-coded maps to help them navigate the building, as well as the locations of the 1819 mobile beacons. To cut down on the

long lines that guests encounter at the makerspace, we implemented a pre-registration check-in feature that will allow users to get right to work when they arrive.

When we originally developed our idea, we compared it with other options that would potentially be available. The first potential solution could be having physical pamphlets that would be handed out to guests to replace the need for the general information section of our app. Although with this solution, 1819 would be spending illogical amounts of money in printing costs. Given that UC is a green campus, our application provides an environmentally friendly solution to the problem. Another solution would be a guided tour service instead of the iBeacon self-guided tour that our app will come with. However, with this solution 1819 would need to hire and train tour guides to be familiar with all aspects of the 1819 Innovation Hub therefore our application once again provides the simpler solution. We believe that this application is unique because it will utilize a technology that we think has a lot of untapped potential. Currently an application using this type of technology is not being used at the University of Cincinnati, and if implemented we believe users could benefit from the assistance and convenience that our application will provide.

Project Goals & Methodology:

The overall goal for our project was to provide 1819 with a simple and green solution to the problems that we identified. We wanted to streamline their building tour process to allow visitors to explore the building on their own. Additionally, we wanted to cut down on the wait times that some guests experience at the makerspace on the first floor.

To achieve these goals within the given time frame that we had, our group elected to go with a very agile design and development process. This allowed us to quickly adapt and change the application when we ran in to any issues.

Overview:

The rest of this report covers the process that our group went through in order to design and build the 1819 Mobile application. We will also go over the process for how we tested and redesigned certain parts of our application. Our project goals and the deliverables that we've produced will be outlined, the schedule that we used for our project will be provided as well. Finally, we will explain the core functionality of our application and show the visual design that we went with.

Discussion:

Project Concept:

1819 Mobile is a smart phone application built for Apple devices running iOS 12 or later. With 1819 Mobile, our group aims to make information about the exciting opportunities and services offered at the 1819 Innovation Hub more widely available to anyone interested. By utilizing Bluetooth low energy iBeacons located around the Innovation Hub, our application present users with information about the area that they are in. This information is also easily accessible off-site for those who can't make it to the building. Guests that intend to use the makerspace at 1819 will also be able to pre-register for the check-in process before their visit to save time.

Design Objectives:

As previously stated, our primary goal for the 1819 Mobile application is to make information about the 1819 Innovation hub more easily accessible for anyone that wants to learn more about the services offered. In achieving this, we also wanted to make the experience for visitors more fun and interactive while also streamlining the experience for those who are interested in the makerspace.

To achieve the first goal, we intended to implement a guided tour service within the app that would direct users to the next beacon along a predetermined route. Due to hardware limitations and our timeframe we had to rework some aspects of this feature. For our second goal we had planned to connect our app with the existing Envoy check in service at 1819,

however due to Envoy's lack of third-party application support, the functionality of this feature had to be redesigned as well.

User Profile:

Table 1: User Profile breaks down the targeted audience of our application and the experience that we want to provide to these users.

Table 1: User Profile

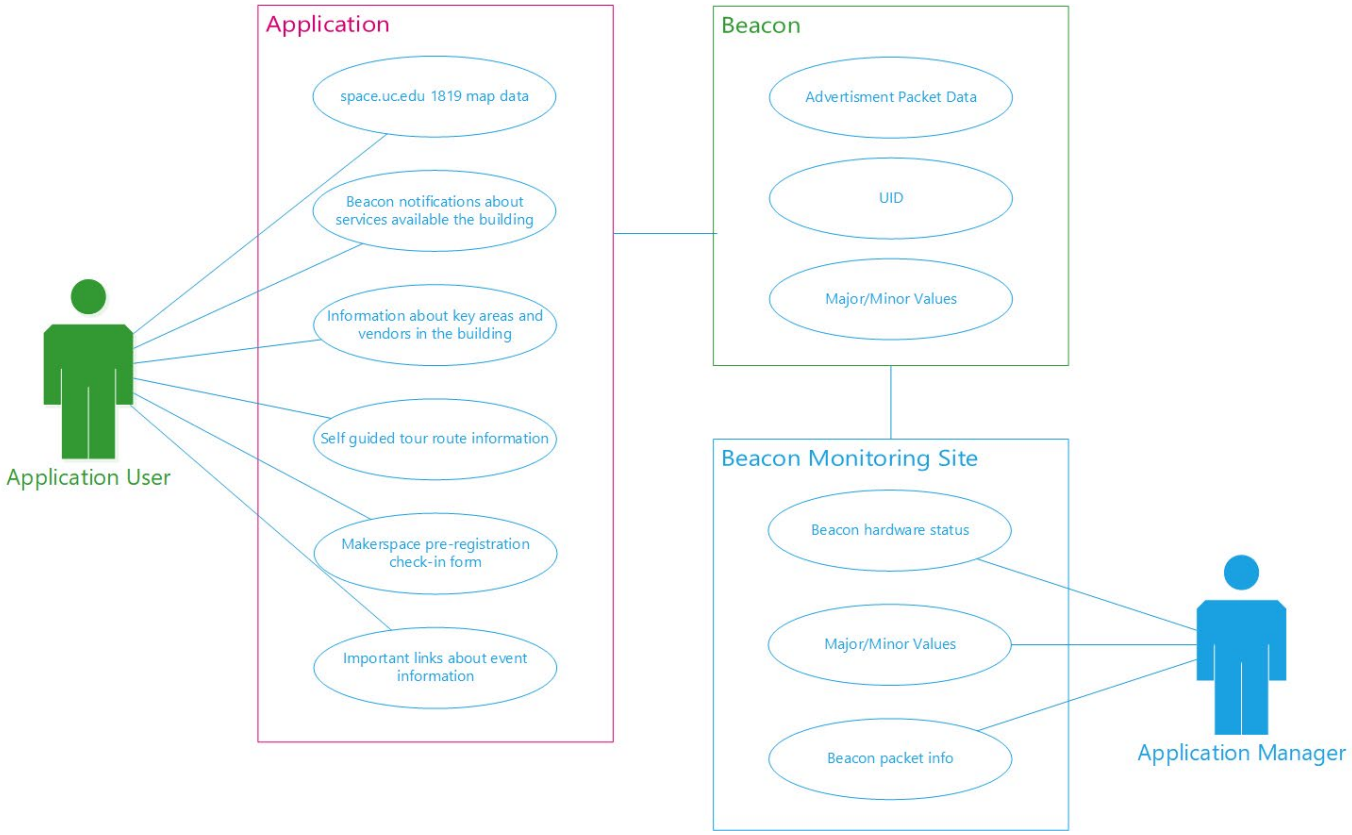
Project: Mobile Application for the 1819 Innovation Hub that utilizes beacon technology
Potential Users: • Students • Faculty • Staff • Guests • 1819 Staff
Software, Interface, and Related Experience: The 1819 Mobile app has been designed for members of the community who are interested in learning more about the exciting tools, services, and opportunities offered by the 1819 Innovation Hub. The application has been designed with accessibility in mind, by utilizing the latest in beacon technology, our application will act as a lighthouse, guiding and informing visitors about key areas in the building, as well as providing the same level of information to off-site users. Our goal is to make each feature within the application accessible to users with a wide array of technical knowledge. We hope it will allow everyone to realize the opportunities available to them at 1819. After the initial deployment there will need to be an 1819 Staff member with access to the kontakt.io admin portal to monitor the battery life of their beacons and reconfigure them if they fall out of sync.
Experience with Similar Applications: There are several applications that utilize beacon technology in a similar manner however there is no direct competition that 1819 could pursue instead of our application.
Task Experience: Users of our application should be familiar with general mobile device and app usage. For the admin portal, users will need to have a basic understanding of the beacons that are being used.
Frequency of Use:

The 1819 mobile app can be used at any time of the day, when the user would like to find out more information about the 1819 Innovation Hub. The application will also be used daily within the building for users to check-in and interact with self-guided tours.
Key Project Requirements based on profile: • Beacon pop-ups that provide information to the user • A clean and accessible UI/UX that is easy to pick up and use • A simple and easily accessible 1819 check-in process • Very little setup required

Use case diagram:

Figure 1: Use Case Diagram depicts each component of our application that users will interact with. Additionally, we have outlined which components are going to be utilized by the students, faculty and staff that visit the building and which components will be used by the application managers.

Figure 1: Use Case Diagram



Technical Elements:

Methodology & Technical Approach

Throughout the development process of 1819 Mobile, we used a very agile development approach in order to help evolve the application as quickly as possible. Since we were on a very rapid timeframe, there would not be much time to do any type of large-scale re-designs on the app as you would see done in a more waterfall development approach. Because of that we decided to do any redesigning along the way if it was necessary. Along with that, most of our testing occurred parallel to the development of our application to save time. We also met weekly as a team to go over our progress on assigned task, as well as any issues that we encountered.

Network (Hardware and Infrastructure)

Our kontakt.io iBeacons utilize Bluetooth low energy technology to broadcast advertisement packets that the 1819 Mobile app can use to present the user with location-based information. (6, p. 1) Our application depends on a consistent connection with the internet so that it can communicate with the kontakt.io server. This ensures that the beacons function properly and it also allows us to use the iBeacon secure shuffling feature that we will discuss later. We created a view controller that would alert the user if their device was ever to lose its internet connection while using the beacon discovery feature of our app. The reason behind this is unlike our prototype, the beacon discovery feature in our main application makes use of secure shuffling and needs internet access to contact Kontakt.io API to retrieve the unshuffled beacon values of a detected beacon.

Application

Our application is coded using Apple's swift programming language and utilizes Apple's core location services to communicate with the 1819 mobile beacons. Our application also utilizes apple's core data services to store building information for 1819. We chose to use core data because it provides flexibility and ease of use for us within Apple's ecosystem. The building and room information for our application is stored in a json file that is read and used to fill all of the application's content. The structure of our app mimics the structure of this file to ensure that it can be easily imported. To achieve this, we've included an array of floor objects. Each floor object consists of a floor name, map image file name, and an array of room objects. Each of these room objects includes, the room's name, number, description (information), it's assigned beacon minor and major values, and an array of photo objects. Each of those photo objects include a file name of a photo associated with the room and alt text describing that photo. When the application is launched for the first time, the app decodes the included json data to swift objects, these swift objects are then passed into the core data store of the application, separating the data into their respected core data entities.

When a user first opens the Beacon Discovery section of our application, they are presented with a view controller that prompts them to enable location services which is required to use our beacon discovery feature. If a user was to select "Allow While Using App" or "Allow Once" the app would then segue to the beacon discovery view controller, and if a user was ever to select "Don't Allow" it would prompt the user with instructions on how to reenable location services for the app in the phones settings. We also added a shortcut to this prompt that would take users directly to the phone's location services settings page for 1819 Mobile so

they can reenable this feature. Once location services have been enabled, the application will start ranging for iBeacons. Once a beacon is discovered by our app, the securely shuffled beacon ID is sent to the Kontak.io cloud server. The Kontakt.io cloud servers decode the Beacon ID and send back the true major and minor values to the device. Once the values are received and the room is found that matches those values, the application will display that room's information. The kontakt.io SDK is used so the app can process the data from the beacons correctly and display information to the user. To make use of the kontakt.io SDK we have use CocoaPods dependency manager to import their libraries into the 1819 Mobile application. (5, p. 1)

The user interface and experience of our application is crafted using Apple's Storyboard UI tools for simplicity and ease of use as well as device flexibility. (4, p. 1) We used Apple's auto layout tools to improve the device flexibility as well and to make sure that it could run on older devices. These tools allowed us to create constraints for images and text fields so the application can be supported on a wide variety of screen sizes. Apple's Human Interface guidelines were followed when designing the UI of our application to ensure accessibility and a consistent design language. We were also able to include alt text properties on each room photo for accessibility purposes. This allows Apple's screen reader to also read alt text descriptions of each of the room's photos.

Our email check-in feature utilizes Apple's Message UI framework which allows us to send emails through the default mail application. This makes it so that we can pre-fill an email using a very basic form and all the user has to do is click send. We even set up the forum to allow users to enter a preferred email if they wanted to use an email address other than the one that their default mail client was configured to be used with.

Security

Kontakt.io secure shuffling, will shuffle a beacons major and minor values, as well as it's MAC-address daily. The Major and minor values will allow us to precisely see which beacon is being targeted with our app and will give only the detail that is related to that beacon. The hardware is encrypted at a firmware level, and have set range limits, so access is limited to only application users inside the facility or near an 1819 Mobile beacon. (2, p. 1) Users do not have access to the beacon information and the application does not have privileges to write to the beacon. These beacons are also assigned to specific owners who have the ability to grant access to other administration staff of their choosing through the Kontakt.io administration console. This allows management to be very secure as there are very few people who have access to manage the device hardware.

Visuals

User interface:

Figure 2: The main menu is the central hub of our app that allows the user to access all the features that are available within each section.

Figure 2: Main Menu

Figure 3: The maps and room information floor navigation pages allow users of our app to select which floor's details they would like to view.

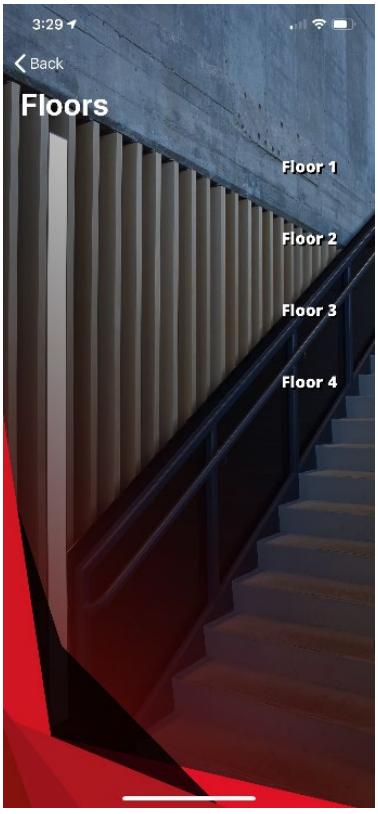


Figure 3: Floors Page

Figure 4: The Maps Page will allow users to access a color-coded map of each floor in the Innovation Hub. These maps also highlight the location of 1819 Mobile beacons in the building so that users can find their way around the building.

Figure 4: Maps Page

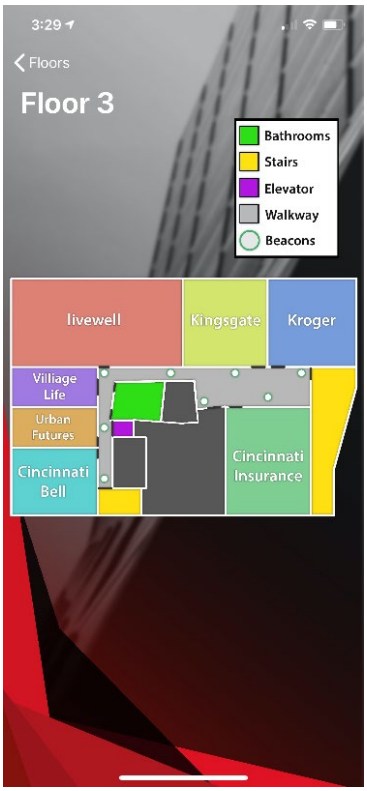


Figure 5 & 6: The Check-in Page has been designed with simplicity in mind and gets straight to the point. All that the user needs to input is their name and optionally a preferred email address. The application will then generate a pre-formatted email that can be sent directly from the pop-up.

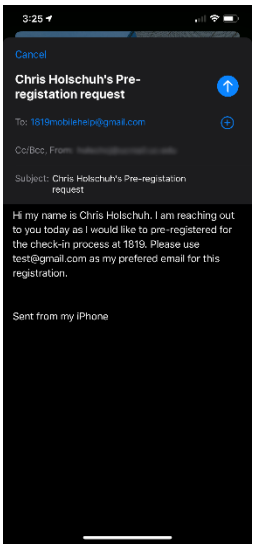


Figure 5: Email Pop-up

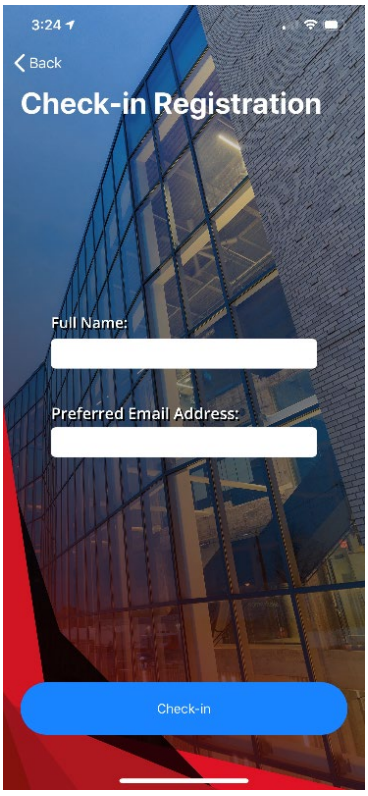


Figure 6: The Check-in Page

Figure 7 & 8: The beacon discovery page features a simple system prompt that requests for users to enable location services on the device. Once enabled the user will be taken to a second screen where the app starts to scan for beacons. Once a beacon is in range the Beacon card will appear.

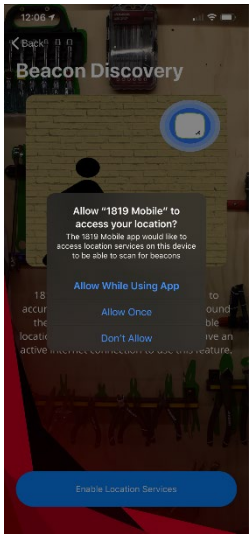


Figure 8: Location Services Pop-up

Figure 9: The Beacon Card, When the user walks near one of the Bluetooth beacons, a card will pop up on the screen that provides information about the area of the building that they're currently in.

Figure 7: Beacon Discovery

Figure 9: The Beacon Card

Testing:

Overview

This project document provides an outline for how we tested and analyzed our application to ensure the functionality of 1819 Mobile's core features. We cover in depth information about the scope of what our tests covered and the objectives of our tests. We also cover the procedures that we came up with by using potential scenarios that might occur while someone is using our app in the 1819 Innovation Hub. Then we outline the conditions that we set for our tests in order for the application to pass each scenario as well as the data that was gathered during these tests. Finally, we provide the timeline that we used for our testing scenarios as well as a conclusion of what we've learned.

Methodology

The 1819 Mobile application has been designed using a very agile development approach. We started with a prototype application that was designed to test the beacon detection functionality. From there we began work on an application with the framework and content that we needed for the informational portions of our app while continuing to develop the beacon prototype separately. Once the beacon concept was fully fleshed out, we began to export those features over from the prototype to our main application. We've incorporated many common practices of agile development into our design approach including the use of a Trello board to track tasks. Additionally, we held weekly scrum meetings to discuss the progress of those tasks and any issues team members maybe encountering. We also performed most of our internal testing as we developed and built out the application's features.

Testing Scope

The tests that we ran covered not only the core features of our application but also the functionality of our beacon devices. We started our beacon testing last semester with the prototype application that we built to identify the beacon signals. This semester our testing focused more on the core application. These tests covered the stability of our application including the launch, navigation, and menu stability. Additionally, we tested the more specific features of our application, like the beacon pop ups that lead to information pages as well as the pre-check-in registration menu. Finally, we tested the minor details of our app including auto layout changes that may affect larger screen sizes and how other apps may affect or interact with our application.

Objective

The objective of these tests was to ensure that features were functioning as intended and to improve the overall experience. We wanted to make our application as user friendly and as easy to pick up as possible, so user feedback was also crucial during these tests. Since the beacon technology is relatively new and still being developed in some respects, it was also crucially important that we test the beacons in a variety of locations around 1819 to ensure that they function properly in varying conditions.

Testing Procedures

These test procedure guidelines were used while designing the scenarios that we conducted on our application. We also established the pass-fail conditions based off of these

a. Beacon Functionality Test – This set of tests will determine whether the beacons are functioning properly or not. They will also determine if the application is able to detect the correct beacon that it is closest to since certain beacons in the building will be closer than others.

b. Launch Stability Test – This test will evaluate the stability of our application at launch. We also want to ensure that the application can consistently launch quickly by launching it repeatedly over a short period of time.

c. Application Stability Test – The application stability test will cover the basic features of our app and ensure that none of them cause the application to slow down or crash. This includes testing the beacon pop-ups to make sure they don't cause any instability.

d. Application Navigation Test – We will also be conducting tests on the navigation elements to ensure that menu and page transitions load quickly and don't cause any slowdowns within the app.

e. User Scenario Tests – Finally we are going to perform more specific tests on our application that will go through more specific user workflows. This includes simple tests like making sure buttons lead to the correct informational pages as well as

more complex tests where we ask potential users to find their way through the application.

Scenarios

a. Beacon Functionality Testing

i. Does the application detect the beacon?

Pass: The beacon is successfully detected by our application

Fail: The application does not detect the beacon, and nothing happens

ii. Does the application detect the correct beacon that it is nearest to?

Pass: The application detects the beacon that it is ranging for

Fail: The application detects one of the other beacons

Table 2: Beacon Functionality

Beacon	Detected	Correct Beacon
OtIPNQ	Yes	Yes
OtfEj0	Yes	Yes
Ot1n0N	Yes	Yes
Otnzgt	Yes	Yes

b. Launch Stability Testing

i. Does the program launch?

Pass: The program launches

Fail: The program doesn't Launch

ii. Does the application consistently launch quickly?

Pass: The application is able to launch multiple times in a row without crashes or slowdowns

Fail: Launching the application eventually causes slowdowns or instability

iii. Does the application ever crash shortly after launching?

Pass: The application stays open after launching

Fail: The application has instability or crashes after the user launches it

Table 3: Launch Tests

Device	Times Tested	Successes	Post Launch Crash
iPhone 6	25	25	None
iPhone XS Max	25	25	None
xCode Simulator	25	25	None

c. Application Stability Testing

i. Can the application function for long periods of time?

Pass: The app can remain open for long periods of time without crashing

Fail: The app crashes or is unstable after being open for too long

ii. Do the beacon pop-ups cause any slowness?

Pass: Beacon pop ups appear consistently each time

Fail: The app is slowed down or crashes if a beacon pop up appears

Table 4: Page Launch Tests

Feature	Time Tested	Crashes	Was there instability?
Main Menu	5 menus tested	None	No issues, all sections open
Maps	4 maps tested	None	All maps open and can be moved
Building Info	16 pages tested	None	All information visible
Beacon Discovery	4 beacons detected	None	All beacon pop-ups are visible
Email Check-in	1 email tested	None	The email is sent to our account
Settings	3 pages tested	None	Email for bug report sent to account

Table 5: Beacon Test

Beacon	Time to pop-up	Crash	Instability:
OtIPNQ	2.69 sec	No	
OtfEj0	2.29 sec	No	
Ot1n0N	2.70 sec	No	
Otnzgt	3.8 sec	No	

d. Application Navigation Testing

i. Is each page accessible from the main menu?

ii. Do the pages load quickly?

Pass: Each page loads quickly and doesn't crash once open

Fail: A page opens slower than others or causes the application to crash

Table 6: Navigation Testing

Page name	Successful Load	Time to load	Page crash
Maps	Yes	Instant	No
Building Info	Yes	Instant	No
Beacon Discovery	Yes	Instant	No
Email Check-in	Yes	Instant	No
Settings	Yes	Instant	No

e. User Scenario Testing

i. Do the beacons make the correct information show up on screen?

ii. Is the user able to successfully send a pre-registration email?

iii. Does the application save its current state after launching another app?

iv. Are location photos tied to the correct room/floor combo?

v. Is any information lost due to auto layout changes?

vi. Are users able to infer directions from the maps?

vii. Does the “clear visited rooms” function work correctly?

viii. Is a user able to grant the app Bluetooth/location services access?

Table 7: Scenario Testing

Test	Conditions	Pass/Fail
Information	When a beacon is found, is the correct floor/room page shown?	Yes
Email	When the user selects preregistration, is an email generated?	Yes
Saved State?	If the user exits the reopens the app, does it save their location?	Yes

Photos	Do photos display in the correct floor/room page?	Yes
Auto Layout	Does the app's layout change correctly depending on the device?	Yes
Maps	Can users navigate the building with the maps in our app?	Yes
Visited Rooms	Does the app remember when the user clears the visited Rooms?	Yes
Bluetooth	Does the application correctly change the system settings?	Yes

Pass/Fail

Our pass/fail conditions were set within each application test. If the application did not pass a specific test, we would have retested the application to ensure the functionality of that specific feature. For the user scenario testing, the pass/fail conditions were simply defined by the goal that was set for each scenario.

Timeline

Since our group elected to go with a more agile approach to our development lifecycle, the bulk of our internal testing happened as we continued to develop our application. Once we completed the core development of our main application, we put our application in the hands of a few potential users in order to beta test the core features and scenarios.

Conclusion

Through our own internal testing that we've done so far, we learned a lot of interesting details about our application. For example, while developing the check in section of our application we discovered that there would be some compatibility issues between our application and the service that 1819 currently has implemented. Due to this we reworked how this feature operates to ensure that users can still save time by pre-registering for a visit to the makerspace. This testing also helped us discover some issues with our development tools. While developing the beacon receiver prototype Apple released an update to iOS and Xcode that prevented us from testing the application on a physical device and restricted us to only using the built in Xcode simulator. Unfortunately, the beacon detecting portion of our application requires the use of a physical device. After further research into the errors we were receiving, we discovered that we were not the only ones having this type of problem. With an update release to Xcode and iOS, it seems as if Apple has unintentionally or intentionally broke the use of reference frameworks with their free tier of Apple developer accounts. Apple has yet to address or acknowledge this as a known issue with their update. Due to this update, our software developer had to pay for an apple developer account in order to continue testing our application on physical devices.

The remainder of our application testing went extremely well. The app was able to pass all the tests that we ran through. No errors were detected, and the application did not experience any slowness or crashes.

Project Management

Budget:

Figure 2: Budget illustrates the financial requirements of the 1819 Mobile application. When calculating the total cost of this project, we factored in the material cost, an estimate of total time required for completion, and the average ongoing pay rate for young professionals in our given roles. In this budget, we have also included our project risk score. This score illustrates the risk of failure based off the complexity and time we will need to complete the project.

Figure 10: Budget

Project Asset Type						**Funding Source (if applicable)**		
Revenue Generating	Comments:					Self	Comments:	
Risk Identification (See Risk Types tab)						**Project Stakeholder(s)**		
	Risk Rating 1-5 (5 is	*Comments*	*Weight*	*Score*		Students, Faculty, Staff, Guests, 1819 staff and administrators.		
Work Effort (days)	2		40%	0.80				
Complexity	2		60%	1.20				
Project Risk Score:				**2.00**				
Estimate of Benefits								
If project will generate revenue, estimate 1 year here:								
Select other benefits the project may bring a customer or								
Risk Avoidance	☐							
Improved customer	☑							
Increased system availability	☐							
Productivity or process	☑							
Reduced costs	☑							
Estimated Cost Rough Order of Magnitude:								
	Rate Per/Hr	**Work Effort (Hours)**	**1 X Costs**	**Ongoing Annual Rate Per/Hr**	**Ongoing Annual Work Effort (Hours)**	**1 X Support Cost**	**Comments: Our re-occuring cost is the price for the API that the beacons use, as well as the Apple Developer plan.**	
Labor - IT	20	200	$ 4,000.00	20	10	$ 200.00		
Labor - External			$ -		0	$ -		
Software - External			$ 100.00			$ 115.00		
Hardware - Externa			$ 600.00					
Misc.								
TOTAL			$ 4,700.00			$ 315.00		
5-Year ROI Analysis								
Description	**5- Year Expected**		**Conservative (1.5)**					
Total Costs	**$ 6,275.00**		**$ 9,412.50**					
Total Benefit	**$ 4,500.00**		$2,250					
Total Costs/Benefit Differential	**$ (1,775.00)**							
Conservative Costs/Benefit Differential	**$ (7,162.50)**							

Design Objectives/Deliverables:

Table 2: Design Objectives and Deliverables outlines the major milestones for our project as well as the assignment deadlines that we have to meet for the class. The left column shows the project milestones and the right side displays the assignments and deliverables that need to be completed during both semesters.

Table 8: Project Milestones

MAJOR PROJECT MILESTONES (DELIVERABLES)			
Fall 2019 Milestones			
Project Initiation	9/2/19	Team Contract Delivered	9/18/19
Begin Research	9/3/19	Completed Project Abstract	10/9/19
Hardware Procured	10/10/19	Presented 3 Minute Elevator Pitch	10/21/19
Use Case/User Profile Complete	10/21/19	Environments Set-up	10/28/19
Prototypes Designed	11/10/19	Final Fall Presentation	11/18/19
Spring 2020 Milestones			
UI/UX Complete	1/8/18	Testing Plan Outlined	2/10/20
Individual Sections Finished	1/20/18	Tech Expo Poster Complete	3/9/20
Alpha and Beta tests	2/29/18	Final Report Submitted	
Bug Fix and Redesign Phase	3/7/18	Tech Expo	4/14/20
Security Penetration Testing	3/15/18		

Project Schedule:

Table 3: Work Breakdown Structure outlines the entire schedule for our project across both semesters. Figure 3: Gantt Chart is the visual timeline of this two-semester schedule.

Work Breakdown Structure:

Table 9: Work Breakdown Structure

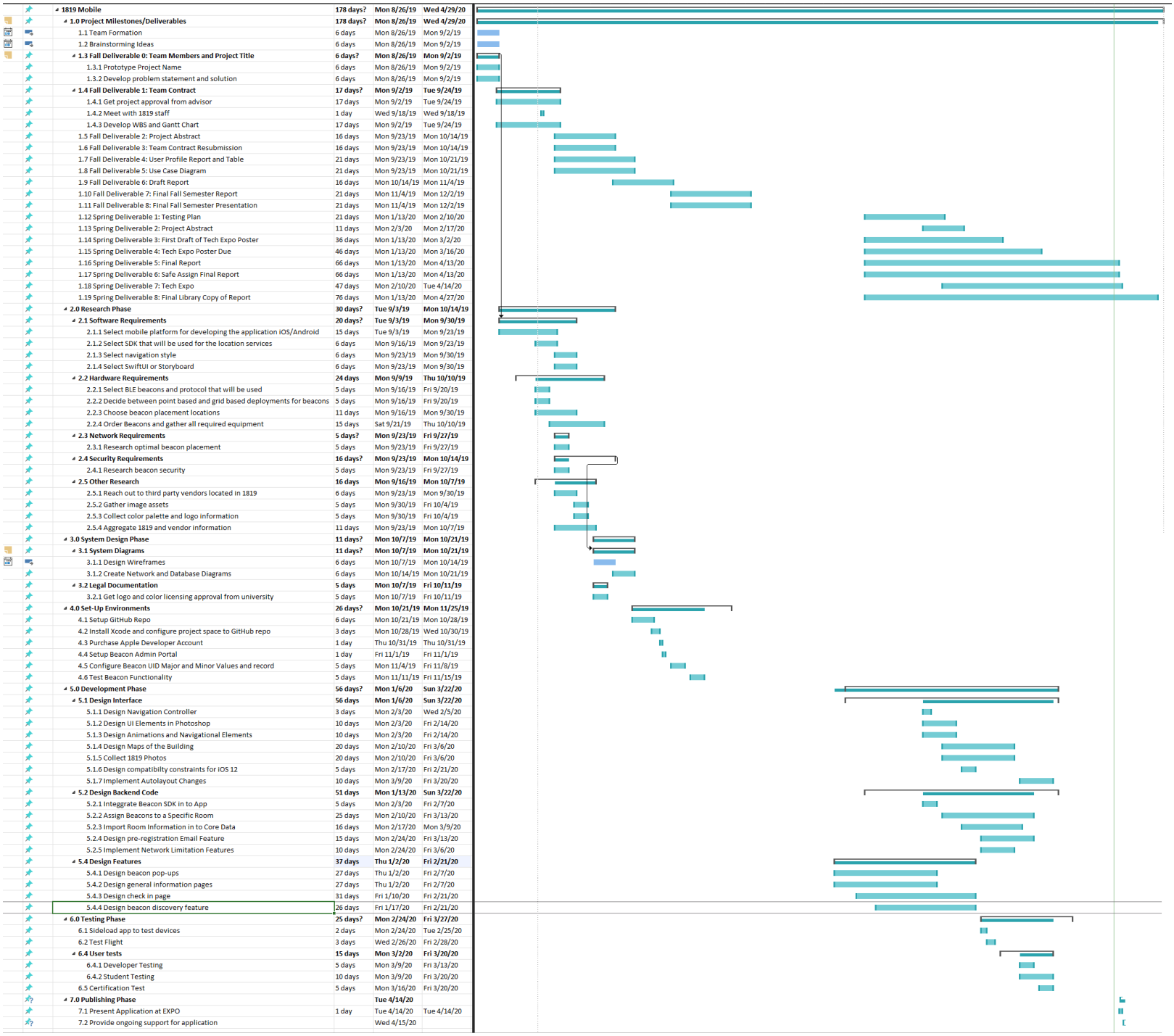
Task Name	Duration	Start	Finish
1819 Mobile	**178 days?**	**Mon 8/26/19**	**Wed 4/29/20**
1.0 Project Milestones/Deliverables	**178 days?**	**Mon 8/26/19**	**Wed 4/29/20**
1.1 Team Formation	6 days	Mon 8/26/19	Mon 9/2/19
1.2 Brainstorming Ideas	6 days	Mon 8/26/19	Mon 9/2/19
1.3 Fall Deliverable 0: Team Members and Project Title	**6 days?**	**Mon 8/26/19**	**Mon 9/2/19**
1.3.1 Prototype Project Name	6 days	Mon 8/26/19	Mon 9/2/19
1.3.2 Develop problem statement and solution	6 days	Mon 8/26/19	Mon 9/2/19
1.4 Fall Deliverable 1: Team Contract	**17 days?**	**Mon 9/2/19**	**Tue 9/24/19**
1.4.1 Get project approval from advisor	17 days	Mon 9/2/19	Tue 9/24/19
1.4.2 Meet with 1819 staff	1 day	Wed 9/18/19	Wed 9/18/19
1.4.3 Develop WBS and Gantt Chart	17 days	Mon 9/2/19	Tue 9/24/19
1.5 Fall Deliverable 2: Project Abstract	16 days	Mon 9/23/19	Mon 10/14/19
1.6 Fall Deliverable 3: Team Contract Resubmission	16 days	Mon 9/23/19	Mon 10/14/19
1.7 Fall Deliverable 4: User Profile Report and Table	21 days	Mon 9/23/19	Mon 10/21/19
1.8 Fall Deliverable 5: Use Case Diagram	21 days	Mon 9/23/19	Mon 10/21/19
1.9 Fall Deliverable 6: Draft Report	16 days	Mon 10/14/19	Mon 11/4/19
1.10 Fall Deliverable 7: Final Fall Semester Report	21 days	Mon 11/4/19	Mon 12/2/19
1.11 Fall Deliverable 8: Final Fall Semester Presentation	21 days	Mon 11/4/19	Mon 12/2/19
1.12 Spring Deliverable 1: Testing Plan	21 days	Mon 1/13/20	Mon 2/10/20
1.13 Spring Deliverable 2: Project Abstract	11 days	Mon 2/3/20	Mon 2/17/20
1.14 Spring Deliverable 3: First Draft of Tech Expo Poster	36 days	Mon 1/13/20	Mon 3/2/20
1.15 Spring Deliverable 4: Tech Expo Poster Due	46 days	Mon 1/13/20	Mon 3/16/20
1.16 Spring Deliverable 5: Final Report	66 days	Mon 1/13/20	Mon 4/13/20
1.17 Spring Deliverable 6: Safe Assign Final Report	66 days	Mon 1/13/20	Mon 4/13/20
1.18 Spring Deliverable 7: Tech Expo	47 days	Mon 2/10/20	Tue 4/14/20
1.19 Spring Deliverable 8: Final Library Copy of Report	76 days	Mon 1/13/20	Mon 4/27/20
2.0 Research Phase	**30 days?**	**Tue 9/3/19**	**Mon 10/14/19**
2.1 Software Requirements	**20 days?**	**Tue 9/3/19**	**Mon 9/30/19**
2.1.1 Select mobile platform for developing the application iOS/Android	15 days	Tue 9/3/19	Mon 9/23/19
2.1.2 Select SDK that will be used for the location services	6 days	Mon 9/16/19	Mon 9/23/19
2.1.3 Select navigation style	6 days	Mon 9/23/19	Mon 9/30/19

2.1.4 Select SwiftUI or Storyboard	6 days	Mon 9/23/19	Mon 9/30/19
2.2 Hardware Requirements	**24 days**	**Mon 9/9/19**	**Thu 10/10/19**
2.2.1 Select BLE beacons and protocol that will be used	5 days	Mon 9/16/19	Fri 9/20/19
2.2.2 Decide between point based and grid based deployments for beacons	5 days	Mon 9/16/19	Fri 9/20/19
2.2.3 Choose beacon placement locations	11 days	Mon 9/16/19	Mon 9/30/19
2.2.4 Order Beacons and gather all required equipment	15 days	Sat 9/21/19	Thu 10/10/19
2.3 Network Requirements	**5 days?**	**Mon 9/23/19**	**Fri 9/27/19**
2.3.1 Research optimal beacon placement	5 days	Mon 9/23/19	Fri 9/27/19
2.4 Security Requirements	**16 days?**	**Mon 9/23/19**	**Mon 10/14/19**
2.4.1 Research beacon security	5 days	Mon 9/23/19	Fri 9/27/19
2.5 Other Research	**16 days**	**Mon 9/16/19**	**Mon 10/7/19**
2.5.1 Reach out to third party vendors located in 1819	6 days	Mon 9/23/19	Mon 9/30/19
2.5.2 Gather image assets	5 days	Mon 9/30/19	Fri 10/4/19
2.5.3 Collect color palette and logo information	5 days	Mon 9/30/19	Fri 10/4/19
2.5.4 Aggregate 1819 and vendor information	11 days	Mon 9/23/19	Mon 10/7/19
3.0 System Design Phase	**11 days?**	**Mon 10/7/19**	**Mon 10/21/19**
3.1 System Diagrams	**11 days?**	**Mon 10/7/19**	**Mon 10/21/19**
3.1.1 Design Wireframes	6 days	Mon 10/7/19	Mon 10/14/19
3.1.2 Create Network and Database Diagrams	6 days	Mon 10/14/19	Mon 10/21/19
3.2 Legal Documentation	**5 days**	**Mon 10/7/19**	**Fri 10/11/19**
3.2.1 Get logo and color licensing approval from university	5 days	Mon 10/7/19	Fri 10/11/19
4.0 Set-Up Environments	**26 days?**	**Mon 10/21/19**	**Mon 11/25/19**
4.1 Setup GitHub Repo	6 days	Mon 10/21/19	Mon 10/28/19
4.2 Install Xcode and configure project space to GitHub repo	3 days	Mon 10/28/19	Wed 10/30/19
4.3 Purchase Apple Developer Account	1 day	Thu 10/31/19	Thu 10/31/19
4.4 Setup Beacon Admin Portal	1 day	Fri 11/1/19	Fri 11/1/19
4.5 Configure Beacon UID Major and Minor Values and record	5 days	Mon 11/4/19	Fri 11/8/19
4.6 Test Beacon Functionality	5 days	Mon 11/11/19	Fri 11/15/19
5.0 Development Phase	**56 days**	**Mon 1/6/20**	**Sun 3/22/20**
5.1 Design Interface	**56 days**	**Mon 1/6/20**	**Sun 3/22/20**
5.1.1 Design Navigation Controller	3 days	Mon 2/3/20	Wed 2/5/20
5.1.2 Design UI Elements in Photoshop	10 days	Mon 2/3/20	Fri 2/14/20
5.1.3 Design Animations and Navigational Elements	10 days	Mon 2/3/20	Fri 2/14/20
5.1.4 Design Maps of the Building	20 days	Mon 2/10/20	Fri 3/6/20
5.1.5 Collect 1819 Photos	20 days	Mon 2/10/20	Fri 3/6/20
5.1.6 Design compatibilty constraints for iOS 12	5 days	Mon 2/17/20	Fri 2/21/20
5.1.7 Implement Autolayout Changes	10 days	Mon 3/9/20	Fri 3/20/20
5.2 Design Backend Code	**51 days**	**Mon 1/13/20**	**Sun 3/22/20**
5.2.1 Integgrate Beacon SDK in to App	5 days	Mon 2/3/20	Fri 2/7/20
5.2.2 Assign Beacons to a Specific Room	25 days	Mon 2/10/20	Fri 3/13/20

5.2.3 Import Room Information in to Core Data	16 days	Mon 2/17/20	Mon 3/9/20
5.2.4 Design pre-registration Email Feature	15 days	Mon 2/24/20	Fri 3/13/20
5.2.5 Implement Network Limitation Features	10 days	Mon 2/24/20	Fri 3/6/20
5.4 Design Features	**37 days**	**Thu 1/2/20**	**Fri 2/21/20**
5.4.1 Design beacon pop-ups	27 days	Thu 1/2/20	Fri 2/7/20
5.4.2 Design general information pages	27 days	Thu 1/2/20	Fri 2/7/20
5.4.3 Design check in page	31 days	Fri 1/10/20	Fri 2/21/20
5.4.4 Design Self Guided Tour	26 days	Fri 1/17/20	Fri 2/21/20
6.0 Testing Phase	**25 days?**	**Mon 2/24/20**	**Fri 3/27/20**
6.1 Sideload app to test devices	2 days	Mon 2/24/20	Tue 2/25/20
6.2 Test Flight	3 days	Wed 2/26/20	Fri 2/28/20
6.4 User tests	**15 days**	**Mon 3/2/20**	**Fri 3/20/20**
6.4.1 Developer Testing	5 days	Mon 3/9/20	Fri 3/13/20
6.4.2 Student Testing	10 days	Mon 3/9/20	Fri 3/20/20
6.5 Certification Test	5 days	Mon 3/16/20	Fri 3/20/20
7.0 Publishing Phase		**Tue 4/14/20**	
7.1 Present Application at EXPO	1 day	Tue 4/14/20	Tue 4/14/20
7.2 Provide ongoing support for application		Wed 4/15/20	

Gantt Chart:

Figure 11: Gantt Chart



	Task Name	Duration	Start	Finish
	1819 Mobile	**178 days?**	**Mon 8/26/19**	**Wed 4/29/20**
	1.0 Project Milestones/Deliverables	**178 days?**	**Mon 8/26/19**	**Wed 4/29/20**
	1.1 Team Formation	6 days	Mon 8/26/19	Mon 9/2/19
	1.2 Brainstorming Ideas	6 days	Mon 8/26/19	Mon 9/2/19
	1.3 Fall Deliverable 0: Team Members and Project Title	**6 days?**	**Mon 8/26/19**	**Mon 9/2/19**
	1.3.1 Prototype Project Name	6 days	Mon 8/26/19	Mon 9/2/19
	1.3.2 Develop problem statement and solution	6 days	Mon 8/26/19	Mon 9/2/19
	1.4 Fall Deliverable 1: Team Contract	**17 days?**	**Mon 9/2/19**	**Tue 9/24/19**
	1.4.1 Get project approval from advisor	17 days	Mon 9/2/19	Tue 9/24/19
	1.4.2 Meet with 1819 staff	1 day	Wed 9/18/19	Wed 9/18/19
	1.4.3 Develop WBS and Gantt Chart	17 days	Mon 9/2/19	Tue 9/24/19
	1.5 Fall Deliverable 2: Project Abstract	16 days	Mon 9/23/19	Mon 10/14/19
	1.6 Fall Deliverable 3: Team Contract Resubmission	16 days	Mon 9/23/19	Mon 10/14/19
	1.7 Fall Deliverable 4: User Profile Report and Table	21 days	Mon 9/23/19	Mon 10/21/19
	1.8 Fall Deliverable 5: Use Case Diagram	21 days	Mon 9/23/19	Mon 10/21/19
	1.9 Fall Deliverable 6: Draft Report	16 days	Mon 10/14/19	Mon 11/4/19
	1.10 Fall Deliverable 7: Final Fall Semester Report	21 days	Mon 11/4/19	Mon 12/2/19
	1.11 Fall Deliverable 8: Final Fall Semester Presentation	21 days	Mon 11/4/19	Mon 12/2/19
	1.12 Spring Deliverable 1: Testing Plan	21 days	Mon 1/13/20	Mon 2/10/20
	1.13 Spring Deliverable 2: Project Abstract	11 days	Mon 2/3/20	Mon 2/17/20
	1.14 Spring Deliverable 3: First Draft of Tech Expo Poster	36 days	Mon 1/13/20	Mon 3/2/20
	1.15 Spring Deliverable 4: Tech Expo Poster Due	46 days	Mon 1/13/20	Mon 3/16/20
	1.16 Spring Deliverable 5: Final Report	66 days	Mon 1/13/20	Mon 4/13/20
	1.17 Spring Deliverable 6: Safe Assign Final Report	66 days	Mon 1/13/20	Mon 4/13/20
	1.18 Spring Deliverable 7: Tech Expo	47 days	Mon 2/10/20	Tue 4/14/20
	1.19 Spring Deliverable 8: Final Library Copy of Report	76 days	Mon 1/13/20	Mon 4/27/20
	2.0 Research Phase	**30 days?**	**Tue 9/3/19**	**Mon 10/14/19**
	2.1 Software Requirements	**20 days?**	**Tue 9/3/19**	**Mon 9/30/19**
	2.1.1 Select mobile platform for developing the application iOS/Android	15 days	Tue 9/3/19	Mon 9/23/19
	2.1.2 Select SDK that will be used for the location services	6 days	Mon 9/16/19	Mon 9/23/19
	2.1.3 Select navigation style	6 days	Mon 9/23/19	Mon 9/30/19
	2.1.4 Select SwiftUI or Storyboard	6 days	Mon 9/23/19	Mon 9/30/19
	2.2 Hardware Requirements	**24 days**	**Mon 9/9/19**	**Thu 10/10/19**
	2.2.1 Select BLE beacons and protocol that will be used	5 days	Mon 9/16/19	Fri 9/20/19
	2.2.2 Decide between point based and grid based deployments for beacons	5 days	Mon 9/16/19	Fri 9/20/19
	2.2.3 Choose beacon placement locations	11 days	Mon 9/16/19	Mon 9/30/19
	2.2.4 Order Beacons and gather all required equipment	15 days	Sat 9/21/19	Thu 10/10/19
	2.3 Network Requirements	**5 days?**	**Mon 9/23/19**	**Fri 9/27/19**
	2.3.1 Research optimal beacon placement	5 days	Mon 9/23/19	Fri 9/27/19
	2.4 Security Requirements	**16 days?**	**Mon 9/23/19**	**Mon 10/14/19**
	2.4.1 Research beacon security	5 days	Mon 9/23/19	Fri 9/27/19
	2.5 Other Research	**16 days**	**Mon 9/16/19**	**Mon 10/7/19**
	2.5.1 Reach out to third party vendors located in 1819	6 days	Mon 9/23/19	Mon 9/30/19
	2.5.2 Gather image assets	5 days	Mon 9/30/19	Fri 10/4/19
	2.5.3 Collect color palette and logo information	5 days	Mon 9/30/19	Fri 10/4/19
	2.5.4 Aggregate 1819 and vendor information	11 days	Mon 9/23/19	Mon 10/7/19
	3.0 System Design Phase	**11 days?**	**Mon 10/7/19**	**Mon 10/21/19**
	3.1 System Diagrams	**11 days?**	**Mon 10/7/19**	**Mon 10/21/19**
	3.1.1 Design Wireframes	6 days	Mon 10/7/19	Mon 10/14/19
	3.1.2 Create Network and Database Diagrams	6 days	Mon 10/14/19	Mon 10/21/19
	3.2 Legal Documentation	**5 days**	**Mon 10/7/19**	**Fri 10/11/19**
	3.2.1 Get logo and color licensing approval from university	5 days	Mon 10/7/19	Fri 10/11/19
	4.0 Set-Up Environments	**26 days?**	**Mon 10/21/19**	**Mon 11/25/19**
	4.1 Setup GitHub Repo	6 days	Mon 10/21/19	Mon 10/28/19
	4.2 Install Xcode and configure project space to GitHub repo	3 days	Mon 10/28/19	Wed 10/30/19
	4.3 Purchase Apple Developer Account	1 day	Thu 10/31/19	Thu 10/31/19
	4.4 Setup Beacon Admin Portal	1 day	Fri 11/1/19	Fri 11/1/19
	4.5 Configure Beacon UID Major and Minor Values and record	5 days	Mon 11/4/19	Fri 11/8/19
	4.6 Test Beacon Functionality	5 days	Mon 11/11/19	Fri 11/15/19
	5.0 Development Phase	**56 days?**	**Mon 1/6/20**	**Sun 3/22/20**
	5.1 Design Interface	**56 days**	**Mon 1/6/20**	**Sun 3/22/20**
	5.1.1 Design Navigation Controller	3 days	Mon 2/3/20	Wed 2/5/20
	5.1.2 Design UI Elements in Photoshop	10 days	Mon 2/3/20	Fri 2/14/20
	5.1.3 Design Animations and Navigational Elements	10 days	Mon 2/3/20	Fri 2/14/20
	5.1.4 Design Maps of the Building	20 days	Mon 2/10/20	Fri 3/6/20
	5.1.5 Collect 1819 Photos	20 days	Mon 2/10/20	Fri 3/6/20
	5.1.6 Design compatibilty constraints for iOS 12	5 days	Mon 2/17/20	Fri 2/21/20
	5.1.7 Implement Autolayout Changes	10 days	Mon 3/9/20	Fri 3/20/20
	5.2 Design Backend Code	**51 days**	**Mon 1/13/20**	**Sun 3/22/20**
	5.2.1 Integgrate Beacon SDK in to App	5 days	Mon 2/3/20	Fri 2/7/20
	5.2.2 Assign Beacons to a Specific Room	25 days	Mon 2/10/20	Fri 3/13/20
	5.2.3 Import Room Information in to Core Data	16 days	Mon 2/17/20	Mon 3/9/20
	5.2.4 Design pre-registration Email Feature	15 days	Mon 2/24/20	Fri 3/13/20
	5.2.5 Implement Network Limitation Features	10 days	Mon 2/24/20	Fri 3/6/20
	5.4 Design Features	**37 days**	**Thu 1/2/20**	**Fri 2/21/20**
	5.4.1 Design beacon pop-ups	27 days	Thu 1/2/20	Fri 2/7/20
	5.4.2 Design general information pages	27 days	Thu 1/2/20	Fri 2/7/20
	5.4.3 Design check in page	31 days	Fri 1/10/20	Fri 2/21/20
	5.4.4 Design beacon discovery feature	26 days	Fri 1/17/20	Fri 2/21/20
	6.0 Testing Phase	**25 days?**	**Mon 2/24/20**	**Fri 3/27/20**
	6.1 Sideload app to test devices	2 days	Mon 2/24/20	Tue 2/25/20
	6.2 Test Flight	3 days	Wed 2/26/20	Fri 2/28/20
	6.4 User tests	**15 days**	**Mon 3/2/20**	**Fri 3/20/20**
	6.4.1 Developer Testing	5 days	Mon 3/9/20	Fri 3/13/20
	6.4.2 Student Testing	10 days	Mon 3/9/20	Fri 3/20/20
	6.5 Certification Test	5 days	Mon 3/16/20	Fri 3/20/20
	7.0 Publishing Phase		**Tue 4/14/20**	
	7.1 Present Application at EXPO	1 day	Tue 4/14/20	Tue 4/14/20
	7.2 Provide ongoing support for application		Wed 4/15/20	

Problems Encountered

During the development lifecycle of our solution we encountered several problems which forced us to rethink some of the functionality of certain core features of our application. The first issue we ran into was with the Envoy integration with our check-in Pre-registration feature. Their service is designed to work in tandem with the first party application that they designed in house, so they are very strict about it being used within apps other than their own. In its place we developed the pre-registration check-in page so that users can send an email directly to the makerspace in order to get on the list ahead of time.

The second issue we encountered was with our “Guided Tour” feature. With the limited time frame that we had there would have been no simple way that we could determine the direction of a user to a beacon. In theory this could be achieved by retrieving the location information and direction of a user’s device and then compare that to stored geo-location data about a beacon’s location within 1819. However, as our app is only a prototype these beacons do not have any defined locations yet. Hardware wise these beacons also do not transmit geo-location data, so geo-location data would need to be discovered and stored for each beacon manually in the app. The biggest issue with this concept is that GPS location information is very inaccurate when used indoors, so the retrieved geo-location data from a device would not be accurate enough for us to know which way we need to direct the user. The beacon discovery and maps feature work together to provide a more self-guided experience for users of our application as opposed to making them go through a full guided tour.

There were also a few visual design choices that we had to adapt due to their complexity and the time that we had available. Originally we wanted to have the beacons

appear as cards at the bottom of the screen that could be swiped up on to view the information but due to the way we have our pages set up to connect to each other this was not feasible. We were also unable to make the floor list and room list in one page like we had originally planned for the same reason.

Future Recommendations

We believe that 1819 Mobile has a lot of future potential. Since we use an editable json file and Apple's core data framework the platform that we designed could very easily be adapted for use by another organization. Given the ongoing expansion that is currently happening around 1819, we believe that this application could very easily be expanded to include all of the buildings within the Uptown Innovation Corridor. A future goal for this project is to set up an AWS S3 bucket to host this json file, so we can update the room data without having to update the app. One more feature that we think could be beneficial if we were to implement the AWS integration is a page showcasing upcoming events at 1819

Conclusion

Fall 2019:

Our goal for the fall semester has been to learn about the technologies that we'll be using for our project and how we'll have to integrate them with each other. Initially we hoped to have a better idea of what 1819 wanted to see in the application and what situations they would want to use it for, however during our initial meeting with 1819 they established that they wanted us to decide which features would be best for the application. With that in mind we spoke with 1819 staff to get more ideas about what would be useful. Chris has made a lot of progress on his swift training, and we now have a working prototype that demonstrates the beacon functionality. During the development of the prototype, we discovered that some of the documentation about implementing and using Kontakt.io's SDK was created using an older version of XCode. To ensure compatibility, we had to conduct further research into the implementation and usage of their libraries. We are currently working on learning how to design the user interface for our application using Apple's Storyboards. Initially, we intended to design and code the user interface and experience using Apple's SwiftUI, however due to compatibility issues with older device models, we decided that Storyboards would be the better option.

Spring 2020:

1819 Mobile has become a fantastic resource which could be utilized by students, faculty, and staff at the university who want to explore the opportunities that are available to them at the 1819 Innovation Hub. There are several monetary and environmental advantages to our solution compared to

other ideas that we came up with while analyzing the problem that we discovered. 1819 Mobile provides a more fun and interactive experience for users that are on-site utilizing the 1819 Mobile beacons that are scattered throughout the building. We also streamline the pre-registration experience for the makerspace, simplifying the process of getting access to all of the tools and services that are available.

In addition to the advantages that our application provides to its users, our team as a whole gained a lot out of this project. Throughout the process of designing and building this application our team learned a lot about the various technologies that we chose to use for 1819 Mobile. Before starting this project most of us knew very little about Low Energy Bluetooth iBeacons, but we have come to learn a lot about the possibilities that are available with them.

Appendix: References

1. Lee, Wei-Meng. “Understanding and Using IBeacons.” *CODE Online*, Codemag, 2014, https://www.codemag.com/article/1405051/Understanding-and-Using-iBeacons
2. Ziobro, Adrian. “FAQ About Kontakt.io Secure.” *Kontakt.io*, Kontakt.io, 2017, https://support.kontakt.io/hc/en-gb/articles/206770089-Frequently-Asked-Questions-about-Kontakt-io-Secure
3. “Human Interface Guidelines.” *Apple*, Apple, 2019 https://developer.apple.com/design/human-interface-guidelines/ios/overview/themes/
4. “Swift Overview.” *Apple*, Apple, 2019 https://developer.apple.com/swift/
5. Mojica, Gregg. “How To Use CocoaPods in Swift and Objective C Projects.” AppCoda, AppCoda, 24 June 2016, https://www.appcoda.com/cocoapods/
6. Adarsh, Monika. “BLE Beacon Technology Made Simple: A Complete Guide to Bluetooth Low Energy Beacons.” *Beaconstac RSS*, MobStac, 18 June 2019. https://blog.beaconstac.com/2018/08/ble-made-simple-a-complete-guide-to-ble-bluetooth-beacons/
7. G. Glevicky, personal communication, September 18th, 2019
8. H. Lemaster, personal communication, October 14th, 2019

Figure 12: Poster

College of Education,
Criminal Justice,
and Human Services
School of Information Technology

1819 Mobile

Cameron DeMoss | Aidan Burns | Chris Holschuh
Group #11 | Advisor Ryan Moore

University of CINCINNATI

Abstract

- The 1819 Innovation Hub is full of exciting opportunities and services.
- 1819 Mobile makes these opportunities more approachable and interactive with tools needed to succeed during your visit.
- Wayfinding beacons help users to navigate the building and find where they are.

Problem

- The University does not offer a resource that aggregates information about the 1819 innovation hub.
- 1819 is filled with great resources for faculty staff and students however it is tightly packed into a small building.
- It can be difficult to find exactly what you are looking for when you visit.

Solution

- 1819 Mobile is an application designed for Apple's iOS that takes advantage of Apple's core data framework.
- Bluetooth Beacons throughout the building present users with information about key locations in the building.

iBeacons are small wireless devices that continuously broadcast a Bluetooth Low Energy signal

Using the Beacon Discovery feature, the device will begin ranging for these iBeacons. Once a beacon is discovered the securely shuffled beacon ID is sent to the Kontak.io cloud server.

The Kontakt.io cloud servers decodes the securely shuffled Beacon ID, then sends back the true major and minor values to the device.

The application receives these values from the server and performs a search on our backend core data store for any rooms that are tied to those values. If a room is found, the application will display that room's information.

1819 Mobile

Chris Holschuh

Aidan Burns

Cameron DeMoss

Agenda

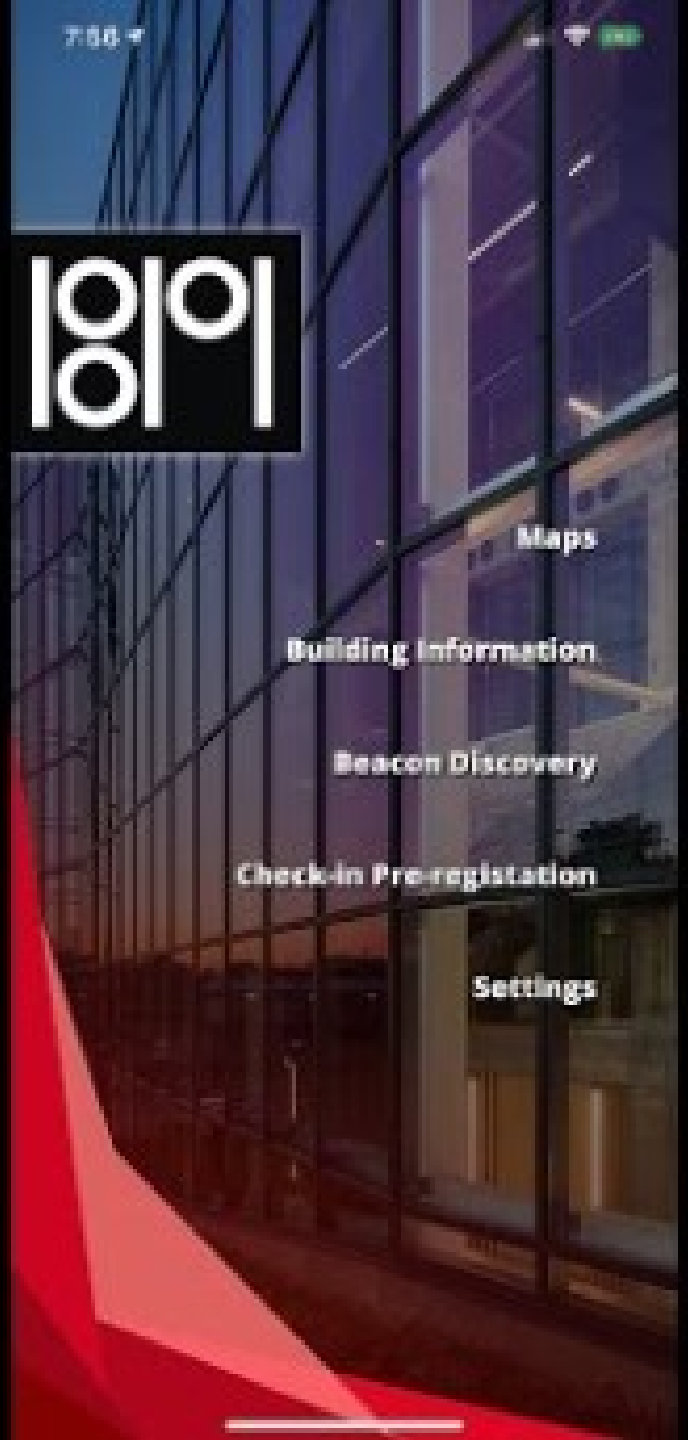
- The problem
- Our solution
- Application Demonstration
- Benefits with our solution
- Conclusion

The Problem

- The university does not offer a resource that aggregates information about the 1819 Innovation Hub
- 1819 has a lot of services and resources packed into a small building
- It can be difficult to find what you are looking for
- Staff take time out of their days to give tours
- The facility does not have a way to register for the makerspace ahead of time

Our Solution

- A mobile application that utilizes Bluetooth Low Energy beacon technology
- Maps that allow users to easily navigate the building
- Building information that is easily accessible on and off-site
- Check-in pre-registration service



Application Demo

https://www.youtube.com/watch?v=xoCUVojXy0E

Benefits

- Helps users to navigate the building
- Cuts down on long wait times for the makerspace
- Simple and easy to understand layout
- Greener and more cost effective when compared with other solutions
- Very adaptable platform for other uses

Conclusion

- 1819 Mobile is a fantastic resource for students, faculty, and staff
- 1819 Mobile makes navigating the innovation hub a more fun and interactive experience
- Provides the university with a greener and more affordable solution for the problem

Questions?

Thanks for watching!