

ABSTRACT

Title of Document: A STATIONLESS BIKESHARE PROOF OF CONCEPT FOR COLLEGE CAMPUSES

Luke Boegner, Yong Cho, Nicholas Fleming, Tyler Gilman, Teng Kuan Huang, Kyle King, Nathaniel Kruder, Joshua Lafond, Timothy McLaughlin, Sye Hoon Noh, William Poh, Emily Ruppel, Libby Wei

Directed by: Dr. Robert Newcomb
Professor, Department of Electrical and Computer Engineering,
University of Maryland, College Park

Bikeshares promote healthy lifestyles and sustainability among commuters, casual riders, and tourists. However, the central pillar of modern systems, the bike station, cannot be easily integrated into a compact college campus. Fixed stations lack the flexibility to meet the needs of college students who make quick, short-distance trips. Additionally, the necessary cost of implementing and maintaining each station prohibits increasing the number of stations for user convenience. Therefore, the team developed a stationless bikeshare based on a smartlock permanently attached to bicycles in the system. The smartlock system design incorporates several innovative approaches to provide usability, security, and reliability that overcome the limitations of a station centered design. A focus group discussion allowed the team to receive feedback on the early lock, system, and website designs, identify improvements and craft a pleasant user experience. The team designed a unique, two-step lock system that is intuitive to operate while mitigating user error. To ensure security, user access is limited through near field

communications (NFC) technology connected to a mechatronic release system. The said system relied on a NFC module and a servo working through an Arduino microcontroller coded in the Arduino IDE. To track rentals and maintain the system, each bike is fitted with an XBee module to communicate with a scalable ZigBee mesh network. The network allows for bidirectional, real-time communication with a Meteor.js web application, which enables user and administrator functions through an intuitive user interface available on mobile and desktop. The development of an independent smartlock to replace bike stations is essential to meet the needs of the modern college student. With the goal of creating a bikeshare that better serves college students, Team BIKES has laid the framework for a system that is affordable, easily adaptable, and implementable on any university expressing an interest in bringing a bikeshare to its campus.

A STATIONLESS BIKESHARE PROOF OF CONCEPT FOR COLLEGE CAMPUSES

By

Luke Boegner, Yong Cho, Nicholas Fleming, Tyler Gilman, Teng Kuan Huang, Kyle King, Nathaniel Kruder, Joshua Lafond, Timothy McLaughlin, Sye Hoon Noh, William Poh, Emily Ruppel, Libby Wei

Thesis submitted in partial fulfillment of the requirements of the Gemstone Program
University of Maryland, 2016

Advisory Committee:

Mr. Jeffrey Holliday - UMD Bike Shop Director
University of Maryland, College Park

Mr. Amol Deshpande, PhD - Computer Science Associate Professor
University of Maryland, College Park

Ms. Nil Gurel - Graduate Research Assistant at Autonomous Vehicle Laboratory
University of Maryland, College Park

Mr. Majid Aroom - Machine Shop & PIRLS Lab Manager
University of Maryland, College Park

Ms. Alison Donlan - Director Product Portfolio, Kryptonite Allegion

© Copyright by

Luke Boegner, Yong Cho, Nicholas Fleming, Tyler Gilman, Teng Kuan Huang, Kyle King, Nathaniel Kruder, Joshua Lafond, Timothy McLaughlin, Sye Hoon Noh, William Poh, Emily Ruppel, Libby Wei

2016

Acknowledgements

First, the team would like to thank Dr. Robert Newcomb for his invaluable mentorship through the duration of the project. His expertise in electrical systems and continued interest in the progress of the overall project propelled the team to research more creative solutions than first imagined. The team also wants to express gratitude to the team's librarians, Ms. Robin Dasler and Ms. Elizabeth Soergel, for their constant support throughout the entire research process. Also, the team wants to thank Mr. Majid Aroom and Dr. Robert Bonenberger for allowing the use of their labs, as well as technical guidance and support. The team wants to thank the University of Maryland Sustainability Fund for providing essential financial support for this research. Finally, the team would like to thank all of the Gemstone staff for their never-ending support and encouragement throughout the past four years.

Table of Contents

Executive Summary	1
Introduction	8
Research Background	8
Research Focus	10
Team Organization	13
Thesis Organization	15
Literature Review	17
Current State of Bikeshares	17
Security and Locking	23
Access Control	37
Geolocation	53
Development	60
Discovery	61
Iteration	87
Integration	151
Integration Demo 1: Arduino to Website through XBee	152
Integration Demo 2: NFC User ID Wireless Authentication	153
Integration Demo 3: Multiple Smartlock Prototype Units	156
Integration Demo 4: Locking & Access	157
Bikeshare Design	158
Conclusion	174
Technological Achievements	174
Lessons Learned	177
Future Directions	179
Appendix A – Drawing	182
Appendix B – System Integration Diagrams	183
Appendix C - Focus Group Questions	186
References	187

Table of Figures

Figure I. Final smartlock design	4
Figure II. Data transfer from ID card to web server	5
Figure III. Administrator bike history view	6
Figure IV. User bike locator view	6
Figure 1. Proposed proof-of-concept of a stationless bikeshare	13
Figure 2. Team BIKES organization tree	15
Figure 3. Image from Kryptonite's original U-lock patent	26
Figure 4. The patent for Kryptonite's K4 U-lock	28
Figure 5. Torsion test machine twisting metal rod	33
Figure 6. Table comparing types of RFID tag and its applications	47
Figure 7. Latest iteration of the loop lock	64
Figure 8. FEA analysis of cutting test on straight portion of U-lock	67
Figure 9. FEA analysis of cutting test of OnGuard K9 folding lock	68
Figure 10. Testing fixtures with pins and bolt cutter head piece	69
Figure 11. Test fixtures attached to Tinius Olsten model H25K-T benchtop tester	70
Figure 12. Comparison of tested NFC modules	74
Figure 13. Comparison of antenna types	75
Figure 14. Kryptonite Keeper 12 U-lock tension test one	79
Figure 15. Kryptonite Keeper 12 U-lock tension test two	80
Figure 16. Kryptonite Keeper 12 U-lock after tension testing	82
Figure 17. OnGuard K9 folding lock tension test one	83
Figure 18. OnGuard K9 folding lock tension test two	83
Figure 19. OnGuard K9 folding lock during testing	85
Figure 20. A picture of a typical student's schedule	90
Figure 21. The top portion of the about page with a hero image and nav bar	91
Figure 22. The middle section of the about page including additional details	92
Figure 23. The first version of the modified U-lock	102
Figure 24. The second version of the modified U-lock	103
Figure 25. The third version of the modified U-lock	104
Figure 26. First, second, and third iterations of the Loop Lock concept	106
Figure 27. Iteration 4 assembly	108
Figure 28. Upper collar and lower collar	109
Figure 29. Electronics housing	109
Figure 30. Final lock prototype	112
Figure 31. Electronics box with pin housing	113

Figure 32. Pin with slot for servo	113
Figure 33. Architecture of the NFC-based access control system	114
Figure 34. Example network test configuration	120
Figure 35. Example XBee Packet Analysis of Various MAC Addresses	120
Figure 36. XCTU's Range Test Program	122
Figure 37. McKeldin Mall Range Test Site	123
Figure 38. Chapel and Fraternity Fields Test Site	123
Figure 39. Coverage Region of Single Node	125
Figure 40. Heat Map Data Collector	127
Figure 41. Heat Map Data Collector Version Two	128
Figure 42. CSV format	129
Figure 43. McKeldin Heat Map Test Leaflet Program	130
Figure 44. McKeldin Heat Map Test Leaflet with Path	131
Figure 45. McKeldin Heat Map Google Maps	132
Figure 46. UMD Campus Web Map Displaying Bicycle Rack Locations	134
Figure 47. Ellicott Community to McKeldin Mall	135
Figure 48. Ellicott Community Heat Map	139
Figure 49. Ellicott Network Routing Map	139
Figure 50. McKeldin Network Routing Map	140
Figure 51. Cambridge Community to Regents Dr. Buildings	141
Figure 52. Cambridge to Regents Routing Map One	143
Figure 53. Cambridge to Regents Routing Map Two	144
Figure 54. Proposed Campus-Wide Network	148
Figure 55. CSV string	153
Figure 56. Membership and usage fees for users of Capital Bikeshare	159
Figure 57. Number of bikes per 10,000 people in bikeshares around the country	170
Figure 58. Cost Analysis	172

Executive Summary

The concept of a bikeshare is an intuitive one: give individuals the opportunity to borrow a bicycle, ride it to another destination, and leave it behind once locked. The complexities arise when contriving a means of securing, locating, and accessing the bicycles in a secure and user friendly fashion. Large, metropolitan bikeshares currently answer these questions by using bicycle docking stations from which riders rent and return bicycles. Each station is equipped with a kiosk connected to a remote server that verifies user access privileges and a number of locking bays for bicycles. Stations are located around the city, and users may rent and return bicycles to and from any station. These distributed stations provide a high degree of mobility for visitors and daily commuters alike, and one could postulate that any area with a dense population that moves about, like a college campus, could benefit from such a system.

The flaw in concluding that a conventional, stationed bikeshare model will meet the needs of students on a college campus is the assumption that the movements of individuals on a college campus mirrors the movement of individuals around a city. Around a college campus, individuals are more likely to make short, erratic, high frequency trips between classes, meetings, and social events. These erratic trips contrast against the more regular and predictable routes of city users and riders who use the bikeshare to commute. For the commuters, it is very likely that a station will be located closer to their final destination than their starting location. On a college campus, however, it is quite possible for the sum of the distances from a student's starting point to the first station and from the second station to the student's final destination to exceed the

distance between the start and end points in the first place. For this reason, a bikeshare that is stationless, one that allows users to securely park bicycles to existing bike racks rather than a small set of custom docking stations, would be better suited to a college campus.

The objective of this thesis is to provide a proof of concept of a stationless bikeshare which relies on a smartlock permanently attached to the bicycle that overcomes the technical challenges associated with creating a bikeshare system tailored to the needs of a college campus. This was accomplished by dividing the task among four subteams of students each with a different focus so that development of the smartlock system's components could be carried out in parallel. The Locking subteam was responsible for the mechanical design of the lock. The Geolocation subteam devised a wireless network to locate the bicycles in the system in addition to a web server to process the bicycle locations and provide this data to bikeshare users. The Access Control subteam configured the electronic interface between the network communications, the user identification system, and the mechanical locking mechanism. Finally, the Business subteam generated recommendations for the operation of the stationless bikeshare.

Before beginning to design the system, the team carried out a thorough review of existing bikeshares, commercially available locks, wireless communication protocols, and web server frameworks. The literature review generated several hypotheses that were tested by each of the subteams. To determine the failure characteristics of commercial locks and influence the physical design of the smartlock, the Locking

subteam conducted several stress-strain analyses of existing locks. The Geolocation and Access Control teams developed code for the electronic components and web framework selected in the literature review. The Geolocation team also tested the signal strength of the modules used to form the wireless mesh network to determine their limitations and ensure that technology identified during the literature review worked as expected. Finally, the Business subteam spearheaded a focus group discussion to gauge the reaction of potential users to the physical design of the smartlock, the web application, and the business plan initially generated by the team. During the focus group discussion, participants had the opportunity to interact with 3D printed models of several different smartlock designs and the mobile web application. This resulted in valuable feedback that the team used in addition to knowledge gained through rapid prototyping to create a final design for a functional smartlock and the outline of a business strategy to market the system.

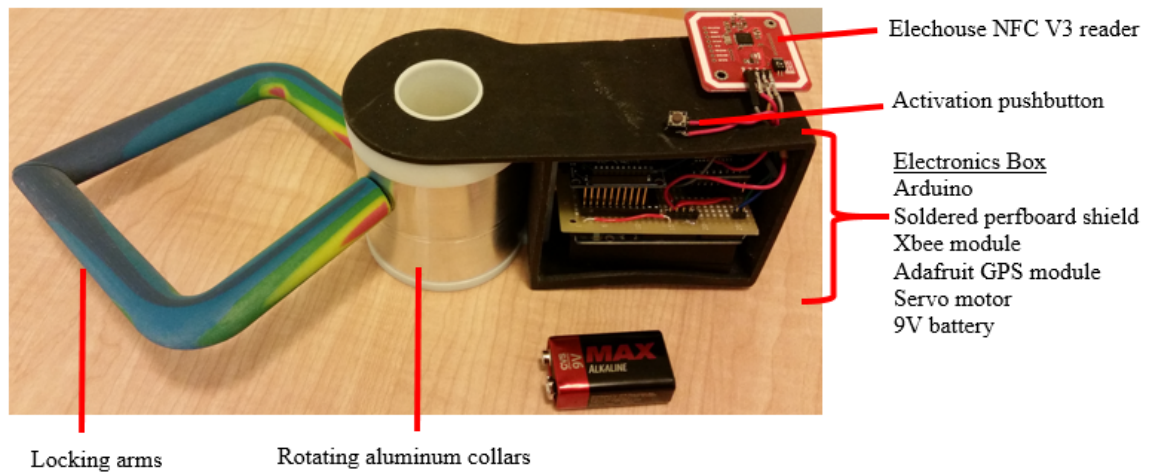


Figure I. Final smartlock design

The final smartlock design, shown in Figure I, includes a mechanical lock comprised of locking arms that open and close as two aluminum collars rotate. The motion of these collars is restricted by a pin, not visible in Figure I, which prevents rotation unless the pin is retracted into the adjacent electronics box. The hardware, protected inside the electronics box, is controlled by an Arduino Uno microcontroller that sends and receives signals from a number of different electronic modules used to provide location data, user identification, wireless network connectivity, physical stimuli, and power to the system. For user identification, an Elechouse NFC V3 module reads a user's ID off of a MIFARE smartcard, analogous to a metro SmarTrip card, using near field communication (NFC). In this case, NFC relies on an active reader circuit, the Elechouse module, to procure the tag ID stored inside of the passive circuitry in a MIFARE smart card. An XBee module that operates using the Zigbee wireless protocol was selected for

providing wireless communication and relaying data from the bicycles to the web application. The module can be operated as part of a mesh network so messages from the bike, an end device, can be carried through a system of routers to a coordinator. The coordinator sends the message to the server and upon a response can redistribute the message through the router system. The entire flow of information from tag to server is illustrated in Figure II.

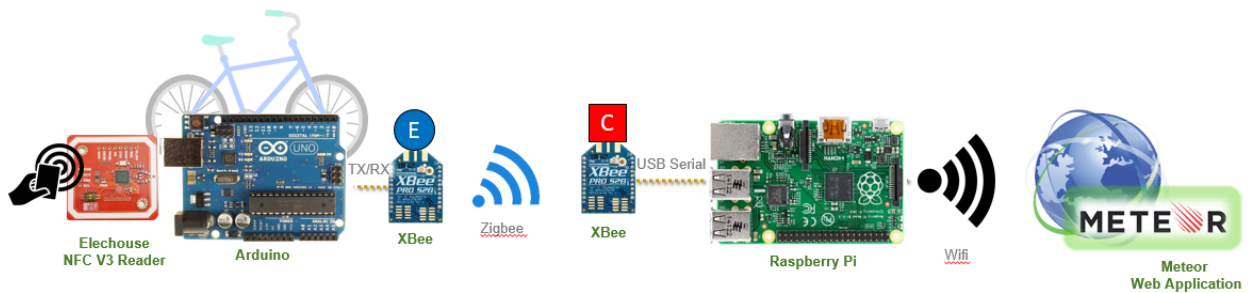


Figure II. Data transfer from ID card to web server

To demonstrate the value of the Zigbee communication protocol to our system, the Xbee module range was tested in several different locations around the University of Maryland's campus. This testing allowed the team to accurately assess the effect of buildings and other obstructions on the signal strength. From the results, the team extrapolated a proposed 56 Xbee node system that could cover every bike rack on the University of Maryland campus. Furthermore, once data is transferred from the smartlock to the web server, the web application can respond in real time to verify a user's ID or to update the location of a bicycle in the online database. The website can

then utilize this database to provide real time bicycle locations, administrator dashboards, or other web tools to confer relevant information to users and system administrators, as seen in Figures III and IV.

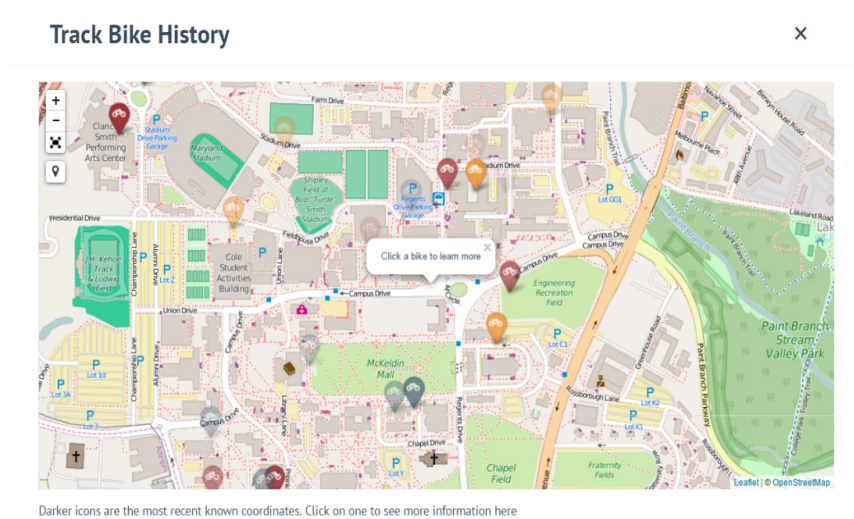


Figure III. Administrator bike history view

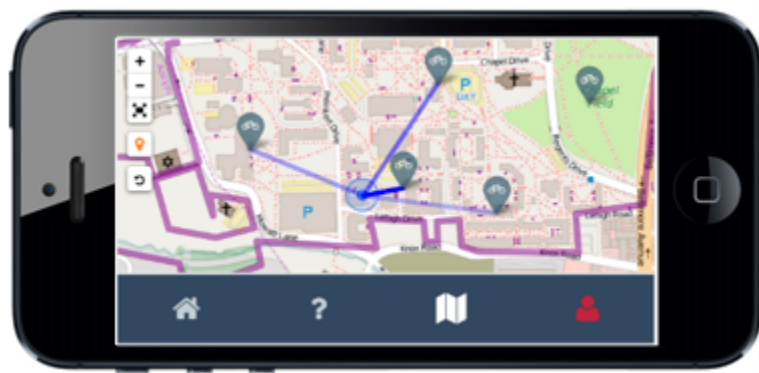


Figure IV. User bike locator view

Beyond the engineering required to implement a stationless bikeshare, this thesis also provides a discussion of the business decisions that must be made in order to realize the system. The suggested business plan provides an overview of the responsibilities that the system operators will have to take on, ranging from typical bikeshare operational activities such as maintaining the fleet size and managing users, to challenges unique to a stationless system such as expanding the wireless network coverage. The plan goes on to propose a pricing system that allows for flexible enrollment periods and imposes a fee for users who exceed the maximum rental time allotted. Finally, the plan addresses the information that the operators must provide to users in order to promote the safety and security of the riders and the bikeshare system as a whole. This includes clear bicycle safety training and a detailed description of how to operate the bikeshare.

The completed smartlock prototype and proposed implementation model demonstrate a new paradigm for stationless bikeshares utilizing a novel mesh network, a real time web application, and an innovative bike lock design. The conclusion that we drew from the process of engineering the smartlock is that the technological challenges presented by a stationless bikeshare can be overcome in order to deliver the flexibility demanded by individuals moving about on a college campus. Removing the stations from the system greatly reduces the initial capital required for a bikeshare system and exponentially increases the user's' freedom.

Introduction

Research Background

As global industrialization and urbanization continues, lack of convenient transportation lingers as a challenge for many individuals. When considering the escalating environmental and monetary costs of today's most popular modes of transportation, commuters must strive to find viable alternatives. Municipalities first officially implemented bikeshares in 1965 to combat the problem of rising costs of public transportation (Shaheen, Guzman, & Zhang, 2010). Today, the typical bikeshare systems operate by using bike stations dispersed throughout the surrounding popular metropolitan destinations. During an allotted time period, users can rent a bicycle from any station, use it for any number of pursuits - be it work, leisure, or tourism - and return the bicycle to any station (Bikeshare, 2013). However, the number and distribution of stations can limit a user's options for his or her final destination. A metropolitan city with a large number of stations overcomes this drawback, but the problem still exists in smaller, though populous, areas such as college campuses.

In the United States, bikeshares are relatively new institutions, but a growing body of research indicates that bikeshares are successful in increasing community awareness of bikers, increasing bicycle usage, and by extension, reducing carbon emissions (Group, 2012). Exceeding the trend of the past decade, in which many bikeshares emerged in densely populated areas around the world, bikeshares have begun

to sprout up in small towns, suburban areas and university campuses. Although the existing body of research concerning bikeshares on university campuses is rather limited, the University of California-Irvine's experience operating Zotwheels, a university supported bikeshare, serves to illustrate the difficulties of implementing the current bikeshare model on a college campus (Fleming & Harris, 2010).

Zotwheels serves the UCI's campus of 1,500 acres and approximately 30,000 students. As reported by the University of California's parking authority, Zotwheels has helped UCI to meet emissions reduction targets, and is a major part of the parking authority's long term environmental vision for the university (Fleming & Harris, 2010). The administrators of Zotwheels estimate that in the future, the bikeshare has the potential to reduce greenhouse gas emissions by over 30 metric tons per year (Fleming & Harris, 2010). Presently, Zotwheels has 25 bicycles in operation and has reached up to 49 rentals per day (Fleming, Harris, & Davis, 2010). While Zotwheels has had a positive effect on the University of California-Irvine campus, that effect has come at a tremendous cost. The initial budget for the project was over a quarter of a million dollars, attributing roughly \$200,000 towards installing the bicycle stations (Fleming, Harris, & Davis, 2010). The resulting initial costs of over \$10,000 for each bicycle in the bikeshare call into question the worth of the endeavor. By discarding the conventional bikeshare model in favor of a stationless bikesharing system, ridership could be increased and overhead costs would be decreased while still preserving the benefits of the bikeshare.

Research Focus

Similar to the University of California-Irvine, the University of Maryland has sought out and planned for Capital Bikeshare, a bikeshare operating out of the nation's capital and surrounding Virginia and Maryland suburbs. The Diamondback, the university's independent, student operated newspaper, published a story in 2013 that discussed about the proposed bike station locations in campus. The plan was due to be implemented by the Spring of 2014; however, these plans never came to fruition as the company that manufactures the bikes for Capital Bikeshare filed for bankruptcy. Three years after the failed attempt to install Capital Bikeshare, UMD is currently continuing with new plans to install Zagster bikeshare in partnership with the city of College Park (Lang, 2016). Team BIKES believes that the tendency of college students to use bicycles to make relatively short trips would have limited the usefulness of the planned traditional system. Since trips may be so brief, students may find that it takes longer to return a bicycle to an inconveniently located station and then walk to class than it would take to walk to class instead of biking. Furthermore, another limiting factor is the cost of stations, which restricts the number of bicycles that can be introduced into the system, thus hindering its functionality. The overall result is that such a bikeshare would likely be unpopular with students if it were implemented. Instead of docking stations, Team BIKES has planned and tested the possibility of creating a stationless bikeshare. To replace the clunky and expensive stations, the team needed to develop three main components. These components include: a way to track the location of the bikes, a way

to securely lock the bikes, and a way for any user to unlock bikes in the system. To replace the stationary docks, the team tested the feasibility of implementing a ZigBee network around campus by configuring XBee devices and conducting coverage tests. This technology, similar to a wireless router, allows the tracking of any bike in the system as long as it remains on campus and is parked at a bike rack covered within the network. The routers act as nodes in a wireless mesh network through which the locations of the bikes can be recorded in a central database and displayed on a phone or website application. This front-end application has a multi-faceted purpose as it provides a central location where users can access their accounts and administrators can monitor usage rates, location, and maintenance needs.

With bicycle tracking under control, the team also needed to develop a secure bike lock. The lock needs to meet and hopefully exceed the security standards of other consumer locks on the market. The lock will securely lock and unlock, intelligently actuated from the onboard electronics. It also needs to house a means of powering the electronic hardware and provide modest protection to the delicate components that would be used to both unlock and locate the bike.

As a means of having any user in the system be able to unlock a bike, the team needed to explore various wireless communication methods of radio frequency identification (RFID) technologies - including near field communication (NFC), and Bluetooth to allow any potential user access to bikes. With a simple electronic authentication, the user can be off riding to their next class in no time.

Through the combination of security, location, and web application, Team BIKES has created a smartlock that can be fitted onto any bicycle, thus enabling a campus friendly bikeshare system. The team also proposes guidelines for operations, terms of use, subscription cost and fees, both general and unique problems to a stationless bikeshare system, and potential ways in which to overcome these challenges. With the goal of creating a bikeshare that better serves college students, Team BIKES has laid the framework for a system that is affordable, easily adaptable, and implementable on any university expressing an interest in bringing a bikeshare to its campus. The team details its project from start to finish beginning with the literature review that documents the research into general bikeshares, locking materials and properties, RFID technologies, and the various means of locating and recording bike locations. The team then documents the methods it took to develop and design the bikeshare. Working through a variety of technical and logistic issues along the way, Team BIKES has created the bikeshare suitable for all college campuses. Figure 1 summarizes the proof-of-concept of the proposed stationless bikeshare design integrating various components.

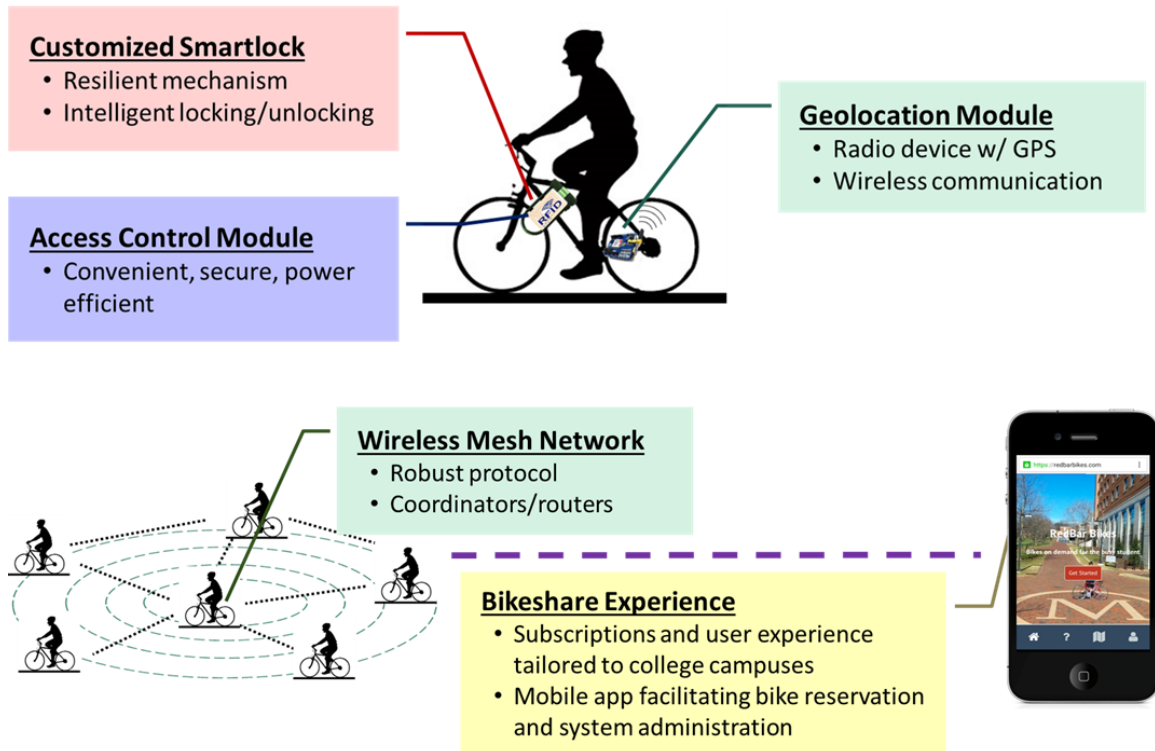


Figure 1. Proposed proof-of-concept of a stationless bikeshare.

Team Organization

In order to delegate tasks and accomplish the goal of proving the feasibility of a stationless bikeshare, the team decided that it was best to break the team off into four subteams, each with particular focus on a different part of the bikeshare design. The subteams include: Business, Locking, Access Control, and Geolocation. The Business subteam took on the responsibility of organizing the focus group discussion and the details related to the operation and design of the business side of the bikeshare. The remaining three subteams fell under the category of technological development. The Locking subteam's responsibilities were testing lock fixtures and developing a smartlock.

The Access Control subteam's responsibilities were developing a means of unlocking the lock using RFID technologies. The Geolocation subteam was tasked with developing a way of tracking bikes, storing this information, and creating a front-end website for users and administrators. With the nature of the project, each subteam needed to communicate with other subteams to address problems and make sure that there was consistency, especially when it came to software compatibility and lock measurements.

In addition to the three liaisons per the Gemstone Department's guidelines, the team created the roles of project manager, to oversee the entire project and lead meetings, and system engineer, to oversee the technology subteams. The team's organizational tree can be found in Figure 2.

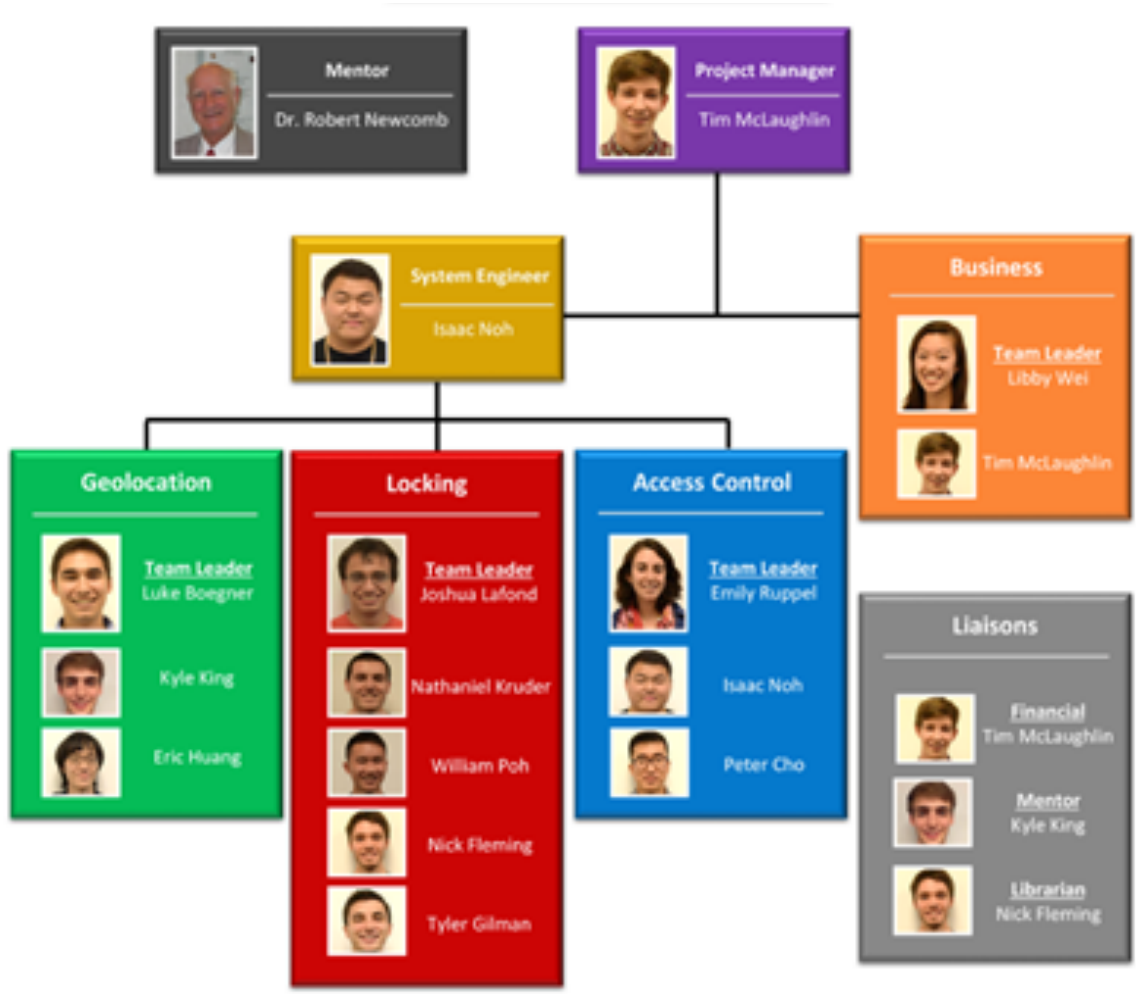


Figure 2. Team BIKES organization tree.

Thesis Organization

This thesis is organized based on the process Team BIKES followed for the product development of the smartlock and the bikeshare system. The second chapter presents a literature review where the team collects past work done in regards to bikeshare systems as well as related technologies in the fields of bicycle locks, access

control, and geolocation. The third chapter transitions to the development process. At the beginning of the development chapter, a discovery section first outlines the progression from deciding on technologies to exploring the capabilities of the selected technologies to experimenting to begin development while learning how all of the selected technologies function. Following the discovery stage, an iteration period occurred where the team conducted a focus group to obtain feedback and gain more information regarding potential customer desires. With a refined set of goals based off of the feedback received from the focus group discussion, the team entered an implementation period. During this phase, the subteams independently conducted all the research and data collection needed for developing the components needed for their subsystems. The fourth chapter then describes the integration period where all of the subteam components were compiled together in the form of compounding demonstrations of capabilities, followed by an overarching bikeshare design. The thesis then wraps up with the fifth chapter's conclusions outlining technological achievements, lessons learned, and future directions.

Literature Review

The first step towards developing an innovative product is to explore the existing work done in related fields. Team BIKES elected to research the existing work by breaking the complex system into subtopics following the same divisions of subteams introduced in the previous section. The business subteam was tasked with researching the current state of bikeshares, noting the different business strategies, costs, and effectiveness of various existing bikeshares. The locking subteam focused on collecting information on what types of bicycle locks proved to be the most functional in regards to security and usability. The access control subteam gathered information on currently existing systems used for close range wireless communication identification techniques, the corresponding security protocols, and the power needed for operating these components. The geolocation subteam researched the various options available for maintaining constant communications with the bicycles in order to track and display the locations to users. Compartmentalizing the research into specific fields, Team BIKES hoped to achieve a broader and deeper understanding of all of the existing technologies and strategies dealing with the team's goal of developing the stationless bikeshare system.

Current State of Bikeshares

Bikeshare systems can be sorted into three different categories: community sharing, manual, and automatic systems. Each system has unique features, but they all

try to accomplish the same goal: to create a bikesharing system that can be used by various members of the community.

Community bikeshares. Community bicycle shares (CBS) are run on a trust-based system. Using public funds, bicycles are placed around a community where anyone is able to borrow a bike and return it to a new location. The bikes may be purchased second hand, donated, or entirely custom with distinct markings that clearly distinguish CBS bikes from privately owned bikes. These systems come in various forms, as they tend to crop up in small communities, campuses, and other areas to which a small system would cater. The bikes are free to be used by anyone in the community, and users generally do not encounter any fees for participation. These programs have the lowest start-up cost and are the best for small communities to encourage tourism and weekend leisure use (dell'Olio, Ibeas, & Moura, 2011).

Although it can be inexpensive to gather a collection of second hand bicycles, community bicycle shares often struggle with the large investment in bicycle repair, unpredictable maintenance, and inability to locate bikes in the system. Since these programs are generally available to the public at no cost, replacing missing bikes and performing repairs hinders the fiscal ability of these programs to continue operation. As David Mozer notes: “[CBS’s] proved hard to sustain because users didn't return the bicycle to public places or because of heavy use, or simple abuse, the fleet dwindled because it was hard to keep the bicycles maintained.” As a result of these problems,

bikeshares began to develop solutions that would address these major flaws (Mozer, 2016).

Manual bikeshares. As a result of all the problems that community bicycle shares faced, bikeshare programs began to move towards the manual bikeshare model. These bikeshares are based around a library model, where one can rent a bike out for a certain amount of time. These systems have bicycle points near or in tourist buildings, libraries, and other public buildings. Users must identify themselves to the staff at these buildings and will then receive a bicycle to use for the day. Many times, these types of bikeshares are free and used to promote bicycle use in towns (dell'Olio, Ibeas, & Moura, 2011). These systems are better suited for small or medium sized towns where an automatic bikeshare would be cost prohibitive due to low demand. Having the ability to rent also means that the fleet of bikes can be properly maintained as well as easily located since users of the system need to sign bikes out for use.

Automatic bikeshares. For high demand cities, automatic systems feature bicycle points throughout a city that are accessed by a user's card or mobile phone rather than an attendant. There are annual or weekly service charges or alternatively, pay per 30 minutes schemes, which have penalty charges if the bicycle is returned late. The most cost extensive bikeshare systems to maintain are in Washington, D.C, Paris, and Barcelona and are operated automatically (Kurtzleben, 2012). Automatic systems are better suited for large metropolitan areas and college campuses where the potential user demand is greater.

Although bikeshares have proved to be successful in large cities, there is a surprisingly limited amount of research data available on creating a bikeshare specifically for college campuses. Existing studies indicate that college communities were generally more accepting of biking and other non-motorized transport compared to other settings (Balsas, 2003). College students were more conscious of the environment and tended not to have access to personal motor vehicles due to cost and limited parking (Balsas, 2003). Therefore, walking and biking provided green, reliable alternatives that are much more campus friendly. Extensive pedestrian infrastructure allowed for easy walking across campuses, but campus roads allowed bikers to travel quickly and safely. Active biking and strong bicycle law enforcement created a safer, more effective environment for bicycle users on and off campus (Balsas, 2003).

Capital Bikeshare and Velib. Various cities employ different bikeshare systems depending on the demand, population density, geographical features, and bicycle infrastructure (Cervero & Duncan, 2003). As a case study, consider the closest bikeshare to College Park: Capital Bikeshare based in Washington, D.C., with service in Alexandria, Virginia; Arlington County, Virginia; and Montgomery County, Maryland. In 2014, the sponsored program had 1,650 bicycles and 175 stations servicing 22,000 subscribing members. Bicycles can be returned to different stations giving patrons the freedom to travel to various destinations with either two-way or one-way travel. Capital Bikeshare participants also use the bicycles for a variety of reasons. Fifty-eight percent of respondents who are subscribers said they use Capital Bikeshare for commuting, while

seventy percent said that they use the service for social activities and errand runs (Bikeshare, 2013).

Capital Bikeshare uses a payment system with four options: pay-per-ride, three-day rental, or a monthly or yearly subscription. The fee for one-time use is \$8, a three-day pass is \$17, a monthly subscription is \$28, and yearly subscriptions are \$85. The first 30 minutes of a paid subscription are free, but any riding time after the 30 minutes is an additional charge. The bicycles have three speeds, equipped with adjustable seats, and both a front and rear LED light, as per bicycle law (Md. TRANSPORTATION Code Ann. § 21-101, 2016). Once finished riding, the user returns the bicycle to a designated station. The stations vary in size and are powered via solar panels (Bikeshare, 2013).

The Velib bikeshare system in Paris is the largest in the world, but it is plagued with high levels of theft and vandalism (Maynard, 2013). Velib bicycles are not anchored when locked and can be easily stolen or tossed in the River Seine. Some bicycles have even been shipped off to northern Africa to sell the parts on the black market (Kazis, 2010). Other bikeshare systems have different, more reliable locking mechanisms that reduce theft, making bicycle replacement costs negligible. For example, during the first two years of operation, Capital Bikeshare has only lost five out of 1,100 bicycles (Kazis, 2010). Capital Bikeshare uses several tactics that lead to their low rate of bicycle theft. The aesthetics of their bicycles are not especially desirable, and they are most clearly from Capital Bikeshare. The parts in the bicycles are also custom made and unique. They also require special tools to disassemble, which discourages thieves from

taking apart the bicycles to sell individual parts (DeMaio & Gifford, 2004). The potential demand of the users for any bikeshare system will decrease if the locking mechanism is not reliable (dell'Olio, Ibeas, & Moura, 2011). Since bike replacements including all associated costs can range in the \$1,000s, it is important for both riders and operators alike to have good security measures to avoid such strenuous costs. Bikeshares also face logistical challenges such as the constraints of bicycle docking. For instance, Capital Bikeshare subscribers often have to search for an available bike or an empty spot on a dock. Due to the uneven distribution of bicycles, some stations may be completely empty while others will be full and unable to accept additional bikes. Capital Bikeshare operates six vans that move approximately 1,000 bicycles around each night to fix the problem of disproportionate quantities of bicycles at each station (Chavez, 2013). With a system that eliminates docking stations, students are free to leave a bicycle at any location. Returned bicycles would not have to funnel down to one individual location. Rather, students could leave the bicycle almost anywhere on campus.

However, there is still the very likely issue that students will ride downhill from the residence halls, leave the bicycles at the classroom building, and walk back to their residence halls, a common issue identified in other bikeshares. An example of how current bikeshares combat this problem is demonstrated by the Velib program that offers users a free extra fifteen minutes of bicycle use if they return bicycles to stations located on top of hills (Vogel, Greiser, & Mattfeld, 2011).

College bikeshares. Some college campuses have already installed bikeshares. For example, Washington State University in Pullman installed \$140,000 automated bikeshare system where students could swipe their university ID card to unlock a bicycle from a station (Tang, 2010). Using the pre-existing infrastructure of student swipes, on which students already rely for everything from housing to food, the bikeshare was tailored to student life. These systems are challenged by short-distance student needs where docks are difficult to incorporate. Student use peaks between classes when students are most likely to borrow a bike to travel to the next class building.

Security and Locking

Modern bikeshare programs rely on powered docking stations to store bicycles when not in use (Midgley, 2011). However, these stations are expensive, time consuming, and the power requirements limit where they can be placed (DeMaio, 2009). Although these stations do keep bikes as secure as possible, their limited placement and mobility hampers the usefulness of these bikeshares. Since rented bicycles must be borrowed from and returned to these stations, there is always the risk that there will not be a bicycle available for use or the station is already full. Both cases cause serious problems for people on a tight schedule, such as commuters who are most likely to use bikeshares consistently, which will result in negative feedback of the systems.

An alternative to station-based bikeshares is to use smartlocks that allow for a stationless system (Rzepecki, 2010). Current concepts such as Bitlock and Lock8 may soon be on the market. Both are wireless bicycle locks with keyless locking and

unlocking as well as location tracking. They offer Bluetooth connection, Global Positioning System (GPS) location retrieval with a smartphone, and backup entry in the event a user's phone runs out of battery (Bitlock, 2015; Velolock Germany, 2015). Lock8 also has an alarm that sounds if its sensors indicate possible theft. Each lock design, however, still has its shortcomings. Bluetooth is readily hackable with today's technology, and individual components of bicycles can still be removed and stolen (Kumar, 2013). Bitlock boasts the potential to make a bikeshare, but has not been implemented yet as such. Recently, Lock8 has pivoted from directly selling a smartlock to customers to managing a bikeshare service after substantially revamping their original product and refunding most of their Kickstarter backers (Velolock Germany, 2015). At the time of writing, Lock8 had not yet launched a bikeshare service.

Bikeshare systems past and present have easily distinguishable bicycles to deter would-be thieves from being able to easily turn a bicycle for a profit. Clearly marking bicycles to indicate that they belong to a city bikeshare system reduces the incentive to steal from the bikeshare, as it poses a greater risk for those with ill intentions. Easily distinguishable bicycles are also more convenient to locate when trying to borrow. In addition, the theft of individual bicycle parts is a significant issue that is affecting current bikeshares. Screwdrivers, wrenches, and Allen keys are commonly used to steal bicycle parts (Van Lierop, Grimsrud, & El-Geneidy, 2015). In order to combat this widespread problem, bikeshares, such as Citibike in New York, use parts that require special proprietary tools to disassemble (Allyn, 2013). To prevent general misuse and theft,

some systems use disabling methods such as automatic wheel locks. GPS tracking allows these bicycles to be recovered or tracked if they are being moved far out of the city (DeMaio, 2009).

One of the biggest challenges when attempting to deal with the more user-friendly stationless bikeshare is determining the safest and most effective lock type. Bicycle theft criminals are rarely caught, so prevention is key, especially with an expensive bikeshare system (Johnson, Sidebottom, & Thorpe, 2008). There are many common techniques for breaking bicycle locks, so a combination of different types of locks, or an extra-strengthened type of lock will be needed to ensure that the bicycles in a stationless system are still secured and accessible (Van Lierop, Grimsrud, & El-Geneidy, 2015).

Locking mechanisms. In order to create a lock for a stationless bikeshare, it is important to understand the history and development of locking technology meant to deter potential thieves. Lock experts' design choices, driven by thieves' attack methods, give insight into the advantages and disadvantages of past products. The team can learn from these past designs and use this understanding to improve on current locks. In this subsection, the Locking subteam goes into detail about past and present locks on the market in order to develop design considerations for the team's smartlock.

Before the development of locks specifically for bicycles, bikers secured their bikes using lengths of chain connected by padlocks. This chain and padlock combination could easily be cut using bolt cutters. In 1972, Kryptonite introduced the U-lock, offering a marked advantage over previous bicycle locking methods. Their U-locks were

heavier and thicker than typical chains, making them more resistant to bolt cutters (Welch, 2013). Kryptonite's first U-lock was made of iron, featuring a flat U-shaped shackle through which a flat crossbar could be inserted and locked in place with a padlock. The lock also included a socket to cover the padlock and protect it from bolt cutters (Kaplan, 1974), as displayed in Figure 3 below.

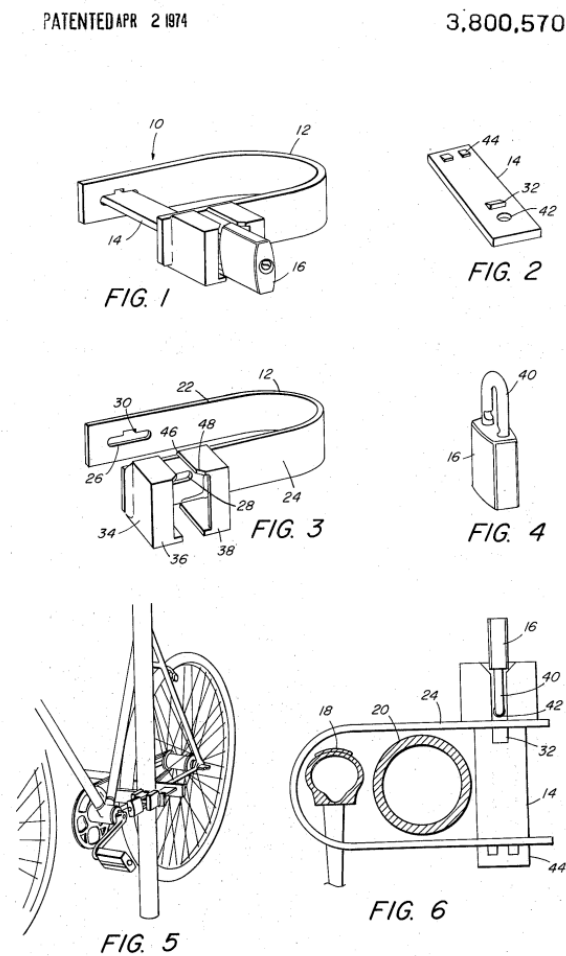


Figure 3. Image from Kryptonite's original U-lock patent. Depicts the complete system, with the crossbar 14 fitting into the u-shaped shackle 12 and being secured by a padlock 16. In Fig. 3, the socket is more clearly represented 34, 36, and 38 as a protective casing for the padlock (Kaplan, 1974).

The brand Kryptonite gained recognition when a New York bike shop used Kryptonite's U-lock to secure a bike to a parking meter outside the shop. After a month, the wheels and seat were gone but the bike's frame was still locked to the meter. Further advertising and design iteration established Kryptonite in the bike locking industry. Kryptonite learned from thieves' methods and improved their lock, thickening and rounding out the previously flat shackle and crossbar, changing the material to hardened steel, and moving the key to the center of the shackle (Welch, 2013).

Kryptonite's next lock design, the K4 lock, boasted ease-of-use by allowing the user to insert the U-shaped shackle into the crossbar rather than threading the crossbar through the shackle. First made in 1978, the K4 U-lock featured a "bent-foot," which represented another important change in U-lock design. As can be seen in Figure 4, the bent foot design allows the user to first slide one end of the shackle into the crossbar then the other end, rather than having to simultaneously thread both ends of the shackle into the crossbar (Smithsonian Institution, 2003).

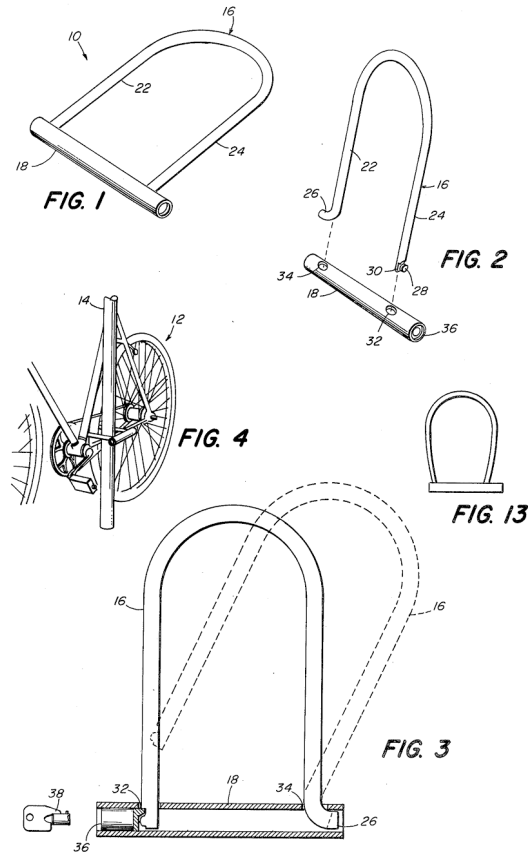


Figure 4. The patent for Kryptonite's K4 U-lock. Notice the bend 26 in the bottom portion of the shackle in patent Fig. 2. The "bent foot" design allows the user to first slide in one end of the shackle, then the other, as demonstrated in patent Fig. 3 (Zane & Zane, 1979).

Kryptonite was sold to Ingersoll Rand (IR) in 2001, which led to a line of new, even stronger locks. The standout Kryptonite product to this day is the Fahgettaboudit Mini (Howard, 2013; Lock, 2016; Welch, 2013). The Fahgettaboudit Mini's resiliency comes from its ability to resist thieves' typical attacks on U-locks: angle grinders and

levering with a jack or long pipe. U-locks are made from thick bars of hardened steel designed to resist bolt cutters, which causes thieves to resort to these more involved methods of attack. This is especially true of the Fahgettaboudit Mini, which has an 18mm thick shackle (Kryptonite). If a thief can fit a modified car jack or bottle jack inside a U-lock, they can use the jack to force the lock apart. Similarly, a long pipe can lever the lock apart if a thief inserts it through the U-lock.

The Kryptonite Fahgettaboudit Mini resists these types of attacks because of its small size. With a width of 3.25 in and a height of 6 in, the Fahgettaboudit Mini is too small to insert a jack or pipe while properly locked around a bike rack and the bike (Kryptonite). In order to break the Fahgettaboudit Mini, expert Hal Ruzal had to use an angle grinder to cut through it. Another feature of this lock that makes it even more secure is that it secures both ends of the shackle when locked instead of just one. When only one side is secured, a thief can cut a U-lock in one place and easily remove the unsecured end. By securing both ends of the shackle to the crossbar, thieves have to make two cuts through the Fahgettaboudit in order to remove a portion of the lock and thereby remove the lock (Howard, 2013).

Ruzal tested a wide variety of locks when he tested the Fahgettaboudit. He found that cable locks were all easily breakable with bolt cutters or wire cutters and said that they are only good for securing the front wheel of a bike when used in conjunction with another type of lock. Like U-locks, chain locks have developed over the years to be stronger with more hardy locking components. The best chain lock, according to Ruzal,

was the OnGuard Mastiff 8020. It resisted his angle grinder for almost three minutes before the angle grinder's battery died. The Mastiff 8020 features a 10 mm titanium reinforced square link steel chain, which is what gave it the strength to withstand Ruzal's angle grinder (Howard, 2013; OnGuard). In addition, it only weighs 2 lbs. to the Fahgettaboudit Mini's 4.5 lbs. (Kryptonite; OnGuard). The Mastiff 8020 is lighter than the Fahgettaboudit Mini, but both are incredibly strong locks that supply protection geared for tough environments.

An innovative new bike lock created by father and son Bob and John Laughlin, the TiGr lock, is the first lock to use all titanium (Welch, 2013). The TiGr lock comes in a few different sizes, with the 1.25 in version being the strongest. The TiGr lock is unique in its composition and in a number of its other features. It is composed of a locking cylinder connected to a thin loop of titanium that looks similar to a stretched out belt. Since it is made entirely of titanium, it is extremely light, with its heaviest variety weighing less than 2 lbs. It is also flexible due to how thin it is, but still very hard to break. The locking cylinder spins 360 degrees to allow the lock to rotate. While the bike is in motion, the lock can be fixed to the top tube of the bike for convenient storage. It comes in three sizes, the longest of which can lock both wheels and the bike frame to a rack (TiGr).

The TiGr lock performs respectably against attacks. Since it rotates around the key cylinder, trying to lever the lock with a pipe only causes it to spin. When the TiGr lock creators tested it against a car jack, the jack reached full extension without breaking

the lock, a testament to the lock's flexibility and strength (Welch, 2013). However, Ruzal was able to sever it with an angle grinder in 17 seconds (Howard, 2013). It does emit a loud shrieking noise under the angle grinder, which might be enough to draw a crowd and discourage thieves, note its creators (Welch, 2013). The TiGr lock is portable, innovative, strong, and attractive. However, it does not provide the same level of safety as the Fahgettaboudit Mini or the Mastiff 1020.

In addition to functionality, bikers weigh cost when choosing a bike lock (Lock, 2016). The Fahgettaboudit Mini costs \$110. The OnGuard Mastiff 8020 costs slightly less, at \$90 (Howard, 2013). Of the three locks discussed in detail above, the TiGr is by far the most expensive. The TiGr lock's most secure model costs \$199 (TiGr). The amount of money a biker is willing to spend on a lock will depend on the environment and the value of the bike secured (Lock, 2016).

From this analysis of leading bike locks, the team compiled a list of critical qualities for a lock design and methods of achieving those goals. Important qualities of a bike lock are resistance to cutting, resistance to leveraging by jack or pipe, resistance to grinders, weight, and cost. Characteristics to consider when attempting to meet these goals include thickness, hardness, size, flexibility, and material. The team will consider these characteristics in order to design the most effective bike lock design possible by maximizing resistance to cutting, leveraging, and grinding while minimizing weight and cost.

Developing a robust bike lock. Acknowledging that thieves use various lock breaking techniques, bike lock companies employ layered protection to defend against a battery of attacks. Bike lock companies conduct a variety of tests to make sure their locks can withstand these breaking techniques without compromising the security of the lock. ABUS, a bike lock company, currently performs three major tests to check the performance of their bike locks: the freezing and smashing test, the twisting test, and the pulling test (Wolff-Mann, 2014).

The freezing and smashing test is used to emulate an extreme brute force attack. This test emulates a thief using compressed substances such as liquid nitrogen or difluoroethane, a refrigerant, to freeze the lock and then smashing the brittle lock with a hammer (Gray, 2012). The test procedure involves first chilling the lock to a temperature of -40 degrees Fahrenheit and then placing the lock into a unique hammer simulation machine. The hammer simulation machine is an impact machine, similar to the apparatus for a charpy test that provides a sudden impulse to simulate a hammer blow (Wolff-Mann, 2014). Next, the twisting test simulates a thief using a pry bar to twist and break open the lock. This test involves a machine, similar to a torsion test machine, depicted in Figure 5, which allows the bike lock to be twisted in a controlled environment. The machine has a stationary clamp on the bottom, causing the bottom of the bike lock to be fixed, and a horizontal bar that is looped through the upper region of the lock and twists against it (Wolff-Mann, 2014). Finally, the third test utilized by ABUS is the pulling test. This test is used to simulate a thief using a car jack, such as a bottle jack, to lever the lock

and attempt to pull it apart. The lock is first chilled in a similar fashion to the freezing and smashing test to recreate extreme environment conditions. Then, the bike lock is put under tension through a series of fixtures attached to a universal testing machine (Wolff-Mann, 2014).

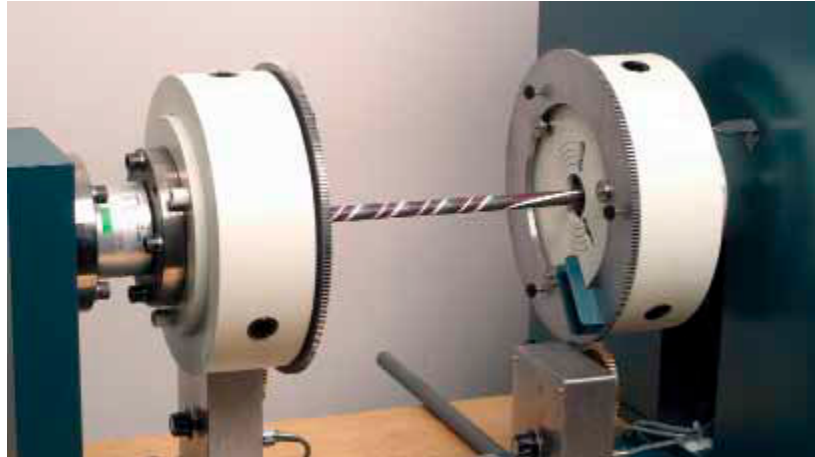


Figure 5. Torsion test machine twisting metal rod.

Although ABUS uses this series of tests to evaluate its own bike locks, many other companies do not have an official standard of tests used for testing. This is due to the difficult nature of emulating some actions, such as cutting a lock using a hacksaw, angle grinder, or bolt cutter, in a repeatable fashion. As a result, bike lock companies, such as TiGr, test their bike locks by asking engineers to break the bike locks using tools and techniques that pertain to the method the bike lock company wishes to test (TiGr, 2016b). Lending credence to these types of tests are independent organizations such as the ART Foundation in the Netherlands. The ART Foundation sets standards for bike, moped, and motorcycle security (TiGr, 2016a). They only certify bike locks that meet

their standards, rating them out of 5 stars based on how well the lock performs in their tests. They consider one or two stars sufficient for bicycles, with more stars required for moped or motorcycle locks (Stichting ART, 2015).

Lock manufacture. Creating locks that can meet the stringent security testing takes a combination of machining methods. The level of difficulty and the type of cut of the machining will vary what tool has to be used. For smoothing a surface, tools include end mills and lathes. For drilling holes, the most widely used tool is the drill bit. All of the above machining operations can be done without a manual control using a Computer Numerical Control (CNC) machine. Team BIKES used all of the aforementioned tools to machine their locks and fixtures.

End mills are used to cut materials in all directions, which is done by their spiral cutting edges called flutes (Company). The number of spiral cutting edges can range from 2-6. Two-flute end mills are used for efficient material removal through high chip ejection and four-flute end mills are used to improve surface finish and reduce ejected chip size (Melin Tool Company, n.d.). Due to its ability to smooth surfaces, Team BIKES used the two and four flute end mills on the locks and fixtures. Unlike end mills, which can cut material in all directions, drill bits are used to cut material axially. The most common drill bit is the twist bit, which has spirals along the shaft to remove material as the front edge cuts the material (DIY Data, 2000). In combination, these bits can create most necessary shapes and allow for the assembly of machined parts.

Lathes are commonly used for cutting, sanding and drilling. They are most known for their process of turning, which is where a work part is rotated around its axis, as a cutting tool shears away unwanted material. The cutting tools are held rigidly on a movable platform that can be moved by hand or power (Hoose, 2002). In order to get a smooth surface on a rounded material, Team BIKES used a lathe because of its ability to cut material, while the work part is being rotated.

CNC machining is known for its high precision machining. This process takes out all human operating error and allows a computer to operate the necessary tools. To do this, the user must upload their m-code to the software package, which controls all of the operating conditions of the CNC machine (ThomasNet, 2015). Team BIKES used the CNC machine to more accurately machine the lock due to the necessary precision that must go into constructing the lock.

There are many different methods of joining metals together. Depending on whether temporary or permanent joints are needed, different techniques must be used. For temporary joints, techniques include soldering, or using external pieces of hardware including rivets, nuts, bolts, or washers. For permanent connections, which are what the team would require, the main technique is welding.

Welding joins two different metals together to make them function as one (Technology, 2004). Welding methods include gas metal arc welding (GMAW), gas tungsten arc welding (GTAW), and flux-cored arc welding (FCAW). GMAW welding is commonly used in high production manufacturing and construction (Society, 2016). This

method is used in a high variety of plate thicknesses. Its advantages include its ease in starting and stopping and its high productivity (Weman, Lindén, & Institute of Materials, 2006). The principle of GMAW welding is that a metallic wire is fed through the welding gun and melted in an arc. A shielding gas protects the arc and pool of the molten material. This shielding gas is commonly argon containing a small proportion of carbon dioxide or oxygen (Weman, Lindén, & Institute of Materials and Mining, 2006). GMAW welding requires the least amount of training and is a good technique for beginners to use.

GTAW welding is mainly used when a high quality of weld is required (Modenesi, Apolinário, & Pereira, 2000). This welding technique requires someone with experience in welding to perform. FCAW welding is the most challenging and requires the most experience out of all the welding techniques described. Welders need advanced training in welding with tubular wires. There is a larger amount of molten material to control in this process, but if done correctly, productivity is very high (Weman, Lindén, & Institute of Materials and Mining, 2006). After researching each welding method, the team determined that while all methods were of sufficient strength for the needs of the team's test fixture, GTAW was ultimately chosen due to its availability at the University of Maryland, its ease in starting and stopping, its high productivity, and its ease of use for beginners.

Access Control

The three access control methods considered for the bikeshare system are Bluetooth, Radio Frequency Identification (RFID), and Near Field Communications (NFC) wireless communication technologies. These technologies are attractive options because of their wide range of current applications in access control. The popularity of these technologies also provides the team with options of a variety of available existing hardware and software for use.

Bluetooth. Bluetooth is a form of wireless communication that allows the exchange of large data through electromagnetic radiation, often viewed as the replacement of physical wires between devices (Timalsina, Bhusal, & Moh, 2012). The physical wire is replaced either by the “pairing” of devices or establishing a distinguished connection between hardware before exchange of data. During pairing, one device takes the role of a master and the other takes the role of a slave, which is left to act according to the input from the master. Although the master node can have infinite slave nodes, the master is limited to communications with only seven of the slave nodes at once. To circumvent this constraint, Bluetooth networks are created in which devices act as both master and slave in relation to other devices. Following the slave and master protocol, the network can be designed to follow a desired logical protocol, enabling the connection of all devices in the network.

Bluetooth technology is currently applied most commonly in wireless phones, computers, and their respective accessories. The Bluetooth connections between wireless

phones and accessories or wireless phones to wireless phones allow for the transfer of relatively large data including music, contact information, images, and videos (Zaruba, Basagni, & Chlamtac, 2001). Recently, Bluetooth technologies have become a more common method of access control with the introduction of Bluetooth locks on the market. These locks allow users to lock and unlock their doors and garages through a mobile phone established as the Bluetooth master (Andersson, 2014). Certain master devices are declared during initial setup and allowed access, making Bluetooth optimal for use with personal items. The aforementioned BitLock is a one of the recent examples of personalized Bluetooth locks introduced in the market.

RFID. RFID is another popular form of wireless communication commonly used in access control. A RFID system is comprised of two parts: a reader and a responder. The reader is the active component of the system that generates a radio frequency. In proximity, a responder will return unique self-identifying information to the reader for authentication. The first use of RFID technology was during World War II, in which the British attached transponders to their fighter planes to easily distinguish incoming planes as friend or foe, known as the Identity Friend or Foe (IFF) system. The IFF system resourcefully took advantage of the fact that multiple responders with one reader made the technology well fitted for inventory purposes (Want, 2006). The next major RFID application was the Electronic Article Surveillance (EAS) theft deterrence system which was introduced commercially to the United States in the 1960's. The EAS system places transponders on store inventory, which must be physically removed upon purchase of the

items. Standing RFID readers are stationed at the store exits and will trigger an alarm if transponders on items are not removed before leaving the store. EAS systems are still common and widely used in shopping establishments (Want, 2006).

Another common use of RFID technology was first introduced in the 1980's in the form of electronic toll systems that tagged unique cars. The toll system operated with RFID readers placed at each toll gate and each car in the system was issued a unique transponder. Each time the readers at the toll gate recognizes a transponder, the account linked to detected transponder is billed. A common example of this application is the EZ-Pass service, which is a combination of stations on toll roads and an EZ-pass transponder placed inside each user's car. This system not only allowed for easier handling of toll fees but contributed to data generation of traffic trends (Domdouzis, Kumar, & Anumba, 2007).

Continued advances in shrinking RFID transponders and readers have launched a recent trend to use RFID systems as inventory methods in many fields, replacing the barcode system. RFID is preferred over the barcode system because radio frequency communications only require proximity, not a direct line of sight. Furthermore, these inventory systems are much more efficient compared to barcode systems since the need to actively read each barcode with an infrared beam is eliminated. The RFID responders could be programmed to hold several pieces of identifying information about the object such as encoded data representing an unique ID that are important to the inventory system (Preradovic, Balbin, Karmakar, & Swiegers, 2008). For large supply chain

operations, standardizing the RFID inventory system through each stage of production enables easy management of items through their production, packaging, transportation, and sales. The RFID log of inventory will also provide the potential for data collection of products and sales (Domdouzis, Kumar, & Anumba, 2007).

RFID systems are also desirable in access control methods and are currently used in a variety of security measures, such as key fobs and public transportation cards. The benefit of using an RFID access control system is that the responder assigned to represent a user is also able to hold other information. Such information possibly includes levels of security clearance or other limitations, which allows for more complex designs in access control system.

NFC. NFC is the final mode of wireless communication for the proposed smartlock. NFC is a specialized form of RFID operating in even closer proximity. The defining quality of NFC is that there is no set distinction between a reader and a transponder in a system (Curran, Millar, & McGarvey, 2012). Any NFC device communicating with another NFC can assume the role of transponder or reader, depending on the desired situation. The ability to act as either the reader or the transponder allows NFC technologies to operate in three different operating modes: reader/writer, card emulation, and peer-to-peer. The Reader/Writer mode is the most common of the three possible operations for NFC technologies. The reader/writer mode is often designed around the NFC in smartphones and allows users to access the system through their mobile device (Timalsina, Bhusal, & Moh, 2012).

Even within the reader/writer mode, a variety of systems can exist. Some will assume the user as a reader while others will assume the user as a transponder in the system. Smart posters, a recent NFC application, are useful in spreading information. They contain NFC transponders with an electronic version of posted information to be read and shared using an individual's NFC devices. Oppositely, recent NFC advances in hospitals treat individual NFC enabled devices as transponders. NFC enabled medical devices can directly update the patient's status through hospital readers. The NFC transponders issued to each patient is an easy method of identification and tracking patients progress throughout large hospital systems. Reader/writer NFC systems are very similar to RFID systems and are fairly simple to design and implement (Ok, Aydin, Coskun, & Ozdenizci, 2011).

NFC can also be used in the card-emulation mode in which NFC devices act as NFC cards meant to interact with designated readers. The card-emulation form of NFC is a recent trend driven by the evolution of smartphones and mobile technology. The greatest benefit of the card-emulation mode is that one NFC enabled device may hold and distinguish the information of multiple NFC cards, an attribute that is mostly used in NFC payments and access control. NFC payment refers to paying with NFC enabled devices, such as smartphones, rather than cash or credit cards. NFC enabled credit cards as well as smartphones linked to credit cards are able to communicate all of the relevant information needed for a purchase with a payment reader instantly. Following the same idea, card-emulation is often used for access control methods. Certain high power NFC

devices, including phones, are able to store access information from other NFC systems to be used in later instances. It is possible for an individual's smartphone to act as a key fob or credit card after initial storage of the NFC transponder information (Timalsina, Bhusal, & Moh, 2012).

The last and least common NFC mode of operation is peer-to-peer mode, which enables two or more NFC devices to freely exchange information. In the peer-to-peer mode, no device is set as a reader or a transponder and the devices are free to exchange data as desired. Such modes allow for easy exchange of contacts, images, or other small data and information between NFC devices.

Security. As commercial uses of RFID have grown, there is increasing concern over the security of RFID systems. There is a distinct tradeoff between security and the efficiency of RFID and other identification systems. RFID systems work between a transponder or tag and a scanner. The transponder holds information that is used to identify the user. Unfortunately, current transponders have little to no security, which means that the information could be read by any reader within close proximity to the transponder (Knospe & Pohl, 2004).

The security of RFID systems has been under scrutiny since the advent of the technology. The wireless nature of RFID systems have enabled the security of such systems to be under constant threat of attack. An attacker could potentially gain access to a wireless channel from any location (Yu, Yiu, & Hui, 2009). Due to the clear security flaw of the system, RFID security has been continually improved (Rieback, Crispo, &

Tanenbaum, 2006). One aspect that has made the most progress is the improvements to the IFF. The IFF allows RFID systems to identify attacks which is a huge step in averting such attacks (Rieback, Crispo, & Tanenbaum, 2006). Through improvements to RFID security, experts have classified five unique types of attacks (Rieback, Crispo, & Tanenbaum, 2006).

The first of these attacks is classified as sniffing in which an attacker is eavesdropping on the wireless channel of a RFID system. This causes privacy concerns especially when the information transferred over the channel includes user information as well as location. Another type of attack is tracking. Similar to sniffing, the attacker can eavesdrop on the channel and find the RFID tag location effectively locating the user. The attacker can use this information to track users. Spoofing is another type of attack in which the attacker creates a fake tag from a blank RFID tag. This is then used to access information. The last type of wireless attack is the replay attack. The replay attacker intercepts RFID signals and then retransmits the signal. This type of attack could be used to relay false information or to steal other tag information wirelessly. These four type of attacks show the importance of user information in RFID systems and non-physical flaws in RFID systems (Rieback, Crispo, & Tanenbaum, 2006).

In addition to wireless attacks, RFID systems are also prone to physical attacks. The attacker can physically remove the RFID tag from the object or place the object in a booster bag that can block the RFID scanner. The attacker could also place RFID tags onto other objects and overload the system with more data than it can take (Rieback).

Once again, this attack would be undertaken with the intent of stealing sensitive user information stored in the RFID system (Rieback, Crispo, & Tanenbaum, 2006).

Thus, the core of the problem for RFID systems lies in the RFID tags that are used to hold user information. With much of the information unprotected, user privacy becomes a vital issue when considering the use of RFID. The simple solution to the problem would be to build innate security onto the RFID tags; however, the tradeoff between security and the cost of individual tags would be too substantial (Knospe & Pohl, 2004). Another solution would be to diminish the proximity at which the reader can scan RFID tags. The fatal flaw of this solution is that it only reduces the range of an attack. If an attacker were to get within the small range the system would be vulnerable. Taking everything into account, one effective security measure would be to enhance the security of the RFID tag by encrypting the information stored on the tag. The encryption must be useable on low cost RFID tags with limited computing in order to be effective with commercial RFID tags (Israsena, 2006). One proposed encryption would be to use the Tiny Encryption Algorithm (TEA) to ensure low cost but efficient protection of user information (Israsena, 2006). TEA only uses bitwise logic commands (XOR, AND, OR and SHIFT operations), which require less hardware components than other algorithms that use more complex operations (Israsena, 2006). Luckily, there are commercially available RFID tags that have innate encryption. The Mifare cards offer triple Data Encryption Standard (3DES) for the Ultralight C series, and Advanced Encryption Standard (AES) for the Mifare Plus card.

The difference between the triple Data Encryption Standard and the Advanced Encryption is that AES outperforms 3DES in terms of software and hardware. The 3DES standard is an improvement over the original DES, which is now considered obsolete. The original DES uses a 56-bit key size, which is too small, being able to be broken by brute force attacks. A brute force attack is when an attacker attempts to break the encryption by using every single combination possible. The 3DES remedies the fatal flaw simply by using 3 separate keys thus achieving the maximum security of a 168-bit key size. Overall, the cypher of the system would stay the same so any system that utilized DES is compatible with 3DES.

AES is an overall improvement over the DES system. It is based on a completely new cipher using different key sizes, and blocks. Standard block size for AES is currently 128 bits, using three different key sizes: 128-bit, 192-bit, and 256-bit. As the best current encryption standard, AES is the U.S. Federal government standard. All in all, AES also offers more protection and works faster in small devices due to longer keys and larger block sizes (Alanazi et al., 2010). In the commercial market, privacy and security of consumer information are vital. The effective tradeoff of security and cost for RFID systems is still being considered and researched.

Power. When RFID modules are viewed as conventional digital systems, then the tradeoffs in terms of power requirements will be very simple when deciding on the operating specifications of the chip. Generally, power consumption of a digital system can be simply calculated as $P = CV^2f$, where P is the system power consumption; C is

the load capacitance of [the] system; V is the voltage of the system; f is the system frequency. In principle, “low voltage, low frequency and low capacitance are chosen at the basis of not affecting the system performance” (Shu-qin, Jin-hui, Lei, li-gang, & Wu-chen, 2008). However, problems begin to arise because the properties of RFID systems vary widely as the frequency, voltage, and capacitance change. For instance, RFID tags and readers are categorized by their operating frequency as low frequency (125 kHz), high frequency (13.56 MHz), ultra-high frequency (850-900 MHz), or microwave (>2.4 GHz) (Journal, 2002). Furthermore, RFID tags are categorized as passive, semi-passive, or active depending on the power scheme used (Weis, 2007). The table below summarizes the specific uses of each configuration of RFID system.

Tag Type	Passive	Semi-Passive	Active
Power Source	Harvesting RF Energy	Battery	Battery
Communication	Respond Only	Respond Only	Respond or Initiate
Max Range	10 M	> 100 M	> 100 M
Relative Cost	Least Expensive	More Expensive	Most Expensive
Example Applications	EPC Proximity Cards	Electronic Tolls Pallet Tracking	Large-asset, Livestock Tracking

Figure 6. Table comparing types of RFID tag and its applications.

As noted in Figure 6 above, passive tags enable access control applications such as ticketing and point of sale use. The basic RF-interface in the chips inside of the passive tags has several different elements including: a modulator/demodulator, a rectifier, clock regenerator, power on reset, and voltage regulator (NXP Semiconductors, 2007). Designs such as these only date back to 2002 when the low voltage/low power circuits necessary to achieve this were created (Villard et al., 2002). Since then, these designs have been refined to improve tag performance and decrease cost. Though a passive tag does not require any constant source of power, there are lower limits on the power the reader must transmit in order to communicate with the tag. This requirement is based on the sensitivity of the modulator/rectifier circuit within the tag. Modulator circuits that perform more effectively, coupled with efficient sense amplifiers, reduce the magnitude of the magnetic field required to trigger a response from the passive card to a reader's query (Chawla & Ha, 2007).

The issue of power versus performance in RFID systems has been well documented from the perspective of not only the tag (i.e. Passive versus active tags), but of the reader. Xu et al pointed out that handheld RFID readers have become very popular, but have challenging power limitations because of their portability (Xu, Gu, Wang, & Xing, 2010). If mobile readers have a high power draw, they will quickly deplete their battery source, which must be relatively small given the compact nature of the device. Conversely, low powered readers have a very limited range (Xu, Gu, Wang, Xing, 2010) since the power of a magnetic field of a dipole falls off as $\frac{1}{r^6}$ in the near field (Chawla & Ha, 2007). Consider an average RFID reader such as the TRF7960/61 by Texas Instruments. The reader has a minimum voltage requirement of 2.7 V and draws 10mA of current (the total power requirement is approximately 30 mW), with an output power between 100 mW and 200 mW (Texas Instruments, 2016). As a small, fully integrated circuit, the TRF7960/61 operates very efficiently with energy saving protocols such as only drawing significant current when in the process of reading a tag. Though, one must remember that this is only the RFID reader circuit of the system as a whole. To actively use the output of an RFID reader, the reader circuit is connected to a processor. According to the manufacture specifications, the power requirements for the processor range from less than 2 V for a microprocessor and up to 12 V for a fully programmable RFID reader. This presents significant challenges when balancing the true power consumption of an RFID system with the required functionality and reliability of the device.

There are many ways to power an RFID system, but the team has chosen to study three specific options for powering the RFID reader and processor. Bicycle dynamos, solar panels, and rechargeable batteries are low cost, simple solutions to the challenge of powering an RFID circuit attached to a bicycle. Ultimately, the best solution may be a combination of these components. Rechargeable batteries provide a steady source of voltage and available current that could be fueled by the bicycle dynamos and/or solar panels. However, for the purpose of the literature review, these three options will be reviewed independently to better reveal their merits and challenges.

While bicycle dynamo generators have proven their use for decades as means of powering headlights and more recently as a source of electricity for rechargeable devices, they have several drawbacks that make their use impractical for a stationless bikeshare system. The basic operation of a bicycle dynamo (be it hub or side wall) is to turn a permanent magnet through coils of wires, thus creating an alternating magnetic field and generating current as a result of Faraday's law. With correct impedance matching, typical bicycle dynamos can generate up to 6 W of power as the user pedals the bicycle at a steady pace (Krygowski & Slanina, 2000), but this comes at the cost of approximately 10% additional metabolic effort by the biker (Langenfeld et al., 2002). One must also consider that by definition the previous power generation only takes place when the bicycle is in motion and the user will only access the RFID system when the bicycle is stationary. That said, commercial devices are available for charging batteries with the excess power generated while in motion, but they can be costly (Cycling About, 2012).

The cost of bicycle dynamos varies based on the type of dynamo, and it can be fairly expensive even without an associated battery. Hub generators are normally more expensive, with typical prices ranging from \$60 to \$250, while sidewall generators typically range between \$30 and \$100.

Another alternative that can be low cost is the use of small solar panels to provide sustainably generated power. Portable, commercially available solar cells online generally provide a range of voltages from 3 to 18 V with current generally less than 100 mA. While these panels are normally inexpensive, they do not include practical circuits for regulating the output voltages. Another challenge for individuals seeking to use such products is that all reported voltages are maximum values. The current vs. voltage output of photovoltaics varies significantly depending upon available sunlight (W. University, 2015). This makes it difficult for individuals without access to reliable solar testing equipment to characterize the behavior of the panels. For instance, one could construct a basic circuit for charging a battery with a solar cell, but to efficiently power an entire system, a full understanding of the interactions among the solar cell, the battery and the circuit being powered is required, and must be incorporated in the initial battery aware design (Raghunathan, Kansal, Hsu, Friedman, & Srivastava, 2005).

Rechargeable batteries provide a convenient and sustainable source of power for portable applications because the mAh to weight ratio of modern rechargeables is fairly high. Additionally, if charged properly, battery lifetimes can extend up to 500 cycles of charging and discharging (B. University, 2013b). Lithium-ion batteries provide the

highest mAh to weight ratio of commercially available batteries, but they are limited by their stringent charging requirements (B. University, 2013a). However, because of the potential that lithium-ion batteries hold for effectively powering a portable RFID system, they will be discussed here in greater detail to better describe how one might take advantage of their properties.

First of all, to charge a lithium-ion battery, current is forced into the battery until a predetermined voltage is reached, and then the voltage must be held steady at that point to charge the battery (Instruments, 2011). The challenge with this charging scheme is that the margin of error for the constant voltage level is very small. For instance, many lithium-ion batteries must be charged at 4.2 V, +/- 50 mV (Simpson, 2011). An advantage, however, is that the initial current used to force the battery to 4.2 V can be extremely low over a long period of time (Simpson, 2011). The overall charging time will vary based upon the size of the battery, but there is some freedom to create a low power, slowly charging circuit for the battery using a variable power source such as a solar panel or bicycle dynamo. That being said, it is of utmost importance that users of lithium-ion batteries charge them correctly, or damage to the battery may ensue. It is important to note that batteries of any kind can overheat and damage themselves or the components situated close to them. Lithium-ion batteries are particularly susceptible to overheating (Bro & Levy, 2013). The process of battery damage can begin at as low as 69°C (156°F). The problem is that the reaction that ensues is exothermic, which further exacerbates thermal runaway (Wang et al., 2012), defined by IEEE as “a

condition that is caused by a battery charging current that produces more internal heat than the battery can dissipate {Wang, 2012, Thermal runaway caused fire and explosion of lithium ion battery}(Wang et al., 2012).” In general, thermal runaway in batteries is typically caused by the misuse of the battery rather than extreme temperatures (Bro & Levy, 2013). This could occur because of high discharge rates, short circuits, or overcharging (Bro & Levy, 2013). High discharge rates are primarily a concern when using lithium-ion batteries that have the ability to discharge large currents. Realistically, most conventional batteries cannot achieve a discharge rate high enough to trigger thermal runaway at room temperature (Bro & Levy, 2013). Lithium ion batteries, which have much lower effective heat capacities than typical electrolyte batteries (Bro & Levy, 2013), are particularly susceptible to thermal runaway when short circuited to a near-zero impedance. In this case, a short circuit refers to any connection that bridges the terminal of the battery with a very low resistance and is generally caused by damage to the battery. Finally, overcharging of batteries is particularly a problem with regards to lithium-ion because they lack the chemical recombination mechanism that prevents heat generation in aqueous electrolyte batteries. Also, overcharging causes irreversible oxidation reactions at the positive electrode.

The Literature Review of popular methods for wireless communication proved to be quite beneficial. The existing literature provided clear applications for each of the researched communication methods, clearly outlining its strengths and shortcomings.

The literature guided further considerations in access control in moving forward with development.

Geolocation

In order to support a campus-wide stationless bikeshare, Team BIKES recognizes the need for a robust geolocation system. The purpose of the system is to provide ease of access to the bikeshare for users and system administrators alike. At a minimum, it must provide a way to measure, store, and communicate positional, usage, and maintenance data for all of the bikes that belong to the bikeshare. Over the duration of the project, Team BIKES has thoroughly researched various existing technologies for designing and building the geolocation system, prioritizing options that offer the best mix of affordability, scalability, and reliability.

The literature review for the geolocation system consists of two parts: a survey of feasible implementations for the hardware infrastructure and a survey of best practices for developing the mobile application.

Hardware. One example of a competing bikeshare design is BitLock, a start-up company that created a smartphone-controlled “smartlock” (Bitlock, 2015). With BitLock, users must pair their smartphone using BlueTooth with their U-lock, and then they can unlock the lock wirelessly as long as they have their paired smartphone on hand. The smartlock not only responds to users’ requests to lock or unlock, but also can track the location of the bike along with other metrics. The smartphone can then send this information to a database through the cellphone’s cellular or Wi-Fi connection, allowing

system administrators to keep track of the bikes in real time (Costa, 2015). Integrating BitLock's concept in a bikeshare is innovative but conceptually straightforward. The team would obtain and program a Bluetooth module, such as a RN42-XV Bluetooth Module, that is approximately \$25 per unit (Sparkfun Electronics, n.d. - a). Since Bluetooth is well-researched and supported by a wide selection of documentation, it brings ease of development to those inexperienced with its technology. Once the Bluetooth module is configured to send and receive relevant signals, the team can mount the module on the bike, which then can be replicated for every bike in the bikeshare. Finally, the team can develop a basic smartphone application to interface the smartphone with the Bluetooth modules and the Internet.

Although promising, the previously mentioned Bluetooth-smartphone approach has a serious disadvantage in the context of a campus bikeshare. Obtaining information on the bikes' status (position, usage, etc.) would rely on the user pairing a smartphone with GPS and Bluetooth capabilities as well as an active subscription to mobile Internet. While it may be possible to leverage campus-wide Wi-Fi networks, many are designed for high populous areas and may have poor to no coverage at critical bike rack locations. Though highly prevalent among the intended user base of on-campus students and faculty, smartphones are far from universal; it is not reasonable to exclude those who do not own a smartphone from using the bikeshare.

In order to develop a more inclusive system that can remotely monitor and locate bikes within a bikeshare without user intervention, Team BIKES considered

implementing a global system for mobile communication (GSM) and general packet radio service (GPRS) based system in combination with a Bluetooth module. The bike lock would effectively emulate a basic smartphone. While this would eliminate the need for the user to own a smartphone, this would incur a substantial subscription fee of up to \$40 per month per bike and would represent a major financial burden on a large-scale bikeshare (Consumer Reports, 2016). Given these considerations, the team decided to move in a different direction by looking into ways to preserve the benefits of the Bluetooth-smartphone approach while addressing its drawbacks.

To create a feasible bikeshare that is more cost effective than a GSM/GPRS-based system, Team BIKES set out to design a small-scale cell system. This can be accomplished by a customized RF-Wi-Fi mesh network, which operates by transmitting data between bikes and users through a number of specialized routing stations. The routing stations, functionally analogous to cell towers, serve as bridges that connect on-bike electronics to the web. They are usually installed in buildings for easy access to power and Internet connectivity, and depending on the hardware used, they can each theoretically cover a radius of up to several miles. The number of routing stations depends on the coverage that the bikeshare intends to achieve. Building the customized RF-Wi-Fi mesh network requires a high upfront cost of around \$130 per bike/station, estimations based on price breakdown discussed later. However, by eliminating the monthly subscription fee, the RF-Wi-Fi mesh network proves more affordable than the aforementioned GSM/GPRS configuration.

Unfortunately, implementing this system is technically challenging. In order to simultaneously handle the large number of bikes that will be communicating with the network, a robust communication protocol must be in place (Masri, Khoukhi, & Gaiti, 2011). A widely recognized technique to accomplish this in similar real-time wireless mesh networks (RT-WMN) is Time Division Multiple Access (TDMA) (Wei et al., 2013). In a TDMA system, essentially all users are allocated small time slots where only that user can transmit or receive data over the RF medium. Developing a custom TDMA technique with limited hardware would cause delays in the system due to a "poll-and-wait" strategy. As the number of bikes in the system increases, the quality of service would plummet to a degree that would render this model infeasible in a real-time bikeshare. A RT-WMN has potential as an effective communication system, but selecting a TDMA system could prove disastrous when attempting to scale the system beyond a proof of concept.

Overcoming the time slot limitation of TDMA, ZigBee is an alternative specification for small, low-power, and low-cost radios based on the IEEE 802.15.4 – 2003 Wireless Personal Area Networks standard (Thaku, 2012). It can support wireless mesh networks that are analogous to the previously mentioned custom RF-Wi-Fi mesh network. ZigBee mesh network consists of three classes of nodes: a “coordinator,” several “routers,” and any number of “end devices.” The coordinator is responsible for initializing the network, specifying the operating frequency channel of the network, and facilitating administrative tasks such as allowing other devices to join the network and

connecting the mesh network to the Internet. The multiple routers relay the messages from one node to another, regardless of the node's type. The end devices are installed on bikes in the form of low-power, battery-powered devices sending and receiving relevant messages through the network while interacting with other on-board electronics (NXP, 2014). A prominent implementation of the ZigBee mesh network is based on using a series of low-power radio modules, branded XBee ZigBee (XBee ZB). XBee ZB modules offer a wide variety of configurations, each featuring a specific operating frequency, range, power consumption, and data rate. Out of all modules, Team BIKES determined that XBee-PRO ZB Series 2B, with the two mile maximum range and at \$28.00 per unit, offered the best match to the needs of the bikeshare. See Appendix B for an example configuration of ZigBee mesh network components.

Overall, ZigBee technology shows promise in the context of a stationless bikeshare, as it preserves the advantages of the custom RF-Wi-Fi mesh network, while circumventing the challenge of building a communication protocol from scratch. It is well-documented in the online community, with a wide variety of tutorials, guides, and references available for new developers (Sparkfun Electronics. n.d. - b).

Software. Every software developer faces similar challenges to address major revisions, modify permissions, transmit data, and support multiple platforms. Traditionally, for an application to be available on multiple devices, the developer would create the same application in multiple coding languages. Each time an application was updated, a developer would repeat the changes across each platform and introduce

platform-specific modifications (Cardoso, Hepp, & Lytras, 2008). Since a company needs to support a number of devices, redundant codebases are often too expensive for startups to maintain. To transcend these problems, future companies can leverage code written and tested by a large user base. A startup can reduce development time and cost by incorporating general code rather than custom-built code. There are several emerging web technology frameworks that simplify the incorporation and sharing of general code. Additionally, some frameworks can support multiple devices, store and exchange data, and ensure user security among other features.

A few of the main frameworks used are discussed including Ruby on Rails, MEAN.io (MongoDB, Ember, Angular, and Node), and Meteor. Ruby on rails is a complex framework that offers a high level of control over development. The software has been around for 7-8 years and has matured with a large number of blog posts, answered support questions, and well documented tutorials (Tate & Hibbs, 2006). MEAN.io is a newer framework that has risen in popularity and is around 5-6 years old. The framework is a collection of smaller frameworks that are added to a core Node.js application. Node.js supports and runs each component of the app and is the most minimal part of a web application. On top of Node.js, the first component of MEAN.io is MongoDB, a highly flexible database for the application back-end. The second component, Express.js, is a framework to manipulate the user-interface. The third component is Angular.js which manipulates the web page content, such as text, links, or images. Along with the MEAN stack, Node offers a very extensive package framework

and is not necessarily limited to a MEAN-only web application (Linnovate, 2015).

However, Meteor is the newest framework in the group. Meteor is built on Node.js, but includes significantly more standard features. Meteor has its own package system and has quickly absorbed many of the Node packages. The software is straightforward to learn and uses JavaScript for every logic-based action (Strack, 2012).

These frameworks change the way development occurs. For multidisciplinary teams of designers, developers, and engineers, development is faster and more affordable. More time can be allocated to consider user feedback and make software improvements. While each framework has certain benefits and drawbacks, Meteor remains a forerunner for development. With a growing user community and an officially supported package management system, Meteor maintains optimal client to server communications and easy extensibility. Additionally, in light of many proven case studies that have highlighted the crucial role of mobile applications in the growth and retention of a business's user base, Meteor proved applicable as a foundation for effective development (Cardoso, Hepp, & Lytras, 2008; Gaziulusoy & Twomey, 2014; Schwartz, 2015).

Development

As discussed during the Literature Review chapter, the development of a stationless bikeshare is a process that draws on technologies from many different industries. The proposed research fields run the gamut from business to wireless communication and no single research methodology exists for gathering all of the data necessary to establish a proof-of-concept of a stationless bikeshare. Additionally, Team BIKES' project deviates from the traditional model of research in that the team's ultimate "experiment" is the iterative development of a novel system, not the validation of an existing system or hypothesis. As such, the team's development is best expressed as a progression from initial design decisions to final integration of the system.

Specifically, the team's Development chapter is separated into two sections: discovery and iteration. Each of these sections is then divided into the concepts that each subteam or subteams worked through during a given stage. The result is a chronological account of the team progression as a whole with each subteam's interrelated results presented sequentially. This organization style also clarifies the influence of each subteam's results on the decisions made by the other subteams throughout the research process.

During the discovery stage, the subteams worked in parallel to finalize the decisions concerning the technology that would be used in the bikeshare system. The subteams worked independently to decide and explore the design spaces established

during the literature review; in general, this took the form of either basic prototyping, or hardware experimentation. This stage was to be completed quickly to link the knowledge acquired during the literature review and the practical implementation of the system.

The subteams then refined the designs during the iteration stage. The approach used during this stage varied across subteams, from additional hardware experimentation and force modeling to focus group interactions. Regardless of the manner each team went about, the ultimate objective was to find optimal solutions to the problems each subteam set out to solve. The Integration chapter following the Development chapter will discuss the approach that the team took to join the separate threads of the project into a functional system.

Discovery

The Discovery section explains the evolution of Team BIKES discovery process through three key phases: decisions, exploration, and experimentation. The decisions phase walks through how the team selected components that were going to be a part of the system. The exploration phase includes the discussion of initial testing and introduction of core concepts. The concepts include exploring XBee operations, building testing fixtures, NFC testing, and IRB approvals. Lastly, the experimentation phase discusses various tests of individual components, such as lock testing, encryption protocol, and GPS experimentation.

Decisions. The first steps the team took were to take the findings in the literature review and make decisions on how to continue the project. The team decided to look into

creating 3-D prototypes to model lock designs and to model various styles of locks. The choice of 3-D printing available on the University of Maryland's campus afforded the team a method to rapidly create the prototypes and quickly test design changes.

Aluminum was chosen as the material to fabricate the prototypes once 3-D printing demonstrated the concept. The team chose the material because of its material properties, affordability, and ease of acquisition.

The team selected NFC technology as the RFID technology best suited for the unlocking and authentication need. Finally, the team decided on which grants to apply for as to provide necessary funding alongside the Gemstone department semesterly-allowance.

3-D prototyping. Rapid prototyping is any technique used to rapidly fabricate a model. One type of rapid prototyping involves the use of 3D CAD software, which operators use to create a 3D model. The model is converted into the appropriate file type for a number of rapid prototype machines including stereolithography (SLA), selective laser sintering (SLS), and fused deposition modeling (FDM). Team BIKES had access to FDM machines, specifically the MakerBot products. All of the printed lock prototypes were printed using MakerBot 5th Generation printers with PLA material. These printers are readily available, use inexpensive material, and allow for rapid testing and design changes without great expense nor great time invested in a metal machine shop.

However, 3D printers present several drawbacks. The MakerBot printers have limited accuracy, making small details impossible to duplicate upon repetition. Circles

smaller than one inch in diameter are often misshapen making it difficult to test part alignment or the effectiveness of pins and rods. To compensate for this, the CAD designs must include extra dimension tolerance. To save on material costs, the models are printed with a honeycomb fill instead of solid fill. The honeycomb structure makes the print lighter, faster, and consumes less material at the cost of structural integrity. During the Gemstone Undergraduate Research day, the weak structural integrity became apparent when the model was dropped and the locking arm broke away from the locking collar. The compromised model was remedied by gluing the locking arm back together with the locking collar. The hollow honeycomb structure also makes it difficult to make physical modifications to the printed model. Any cuts or holes that are drilled into the model will expose the inner cavity. For example, if attempting to modify the 3D printed model to fit an additional pin, the drilled hole would not amply guide the pin because of the now exposed irregular honeycomb structure.

Aluminum manufacturing. Following the rapid prototyping stage, the team began aluminum machining. Aluminum is a lightweight and strong metal, but it is still softer and more ductile than steel. This combination of attributes allowed the team to increase machining speed, reduce material expenses, determine acceptable tolerances for machined parts, and setup automated CNC files for computer controlled machining. The CNC uses CAD files and converts them into machine code, which then controls a mill and machines the stock material into the desired part. In the interest of manufacturability, the lock designs were further refined. In the loop lock pictured below in Figure 7, the

collars would be machined separately from the locking arms, thus preventing the waste of a substantial amount of material. The arms would be individually bent to the appropriate angle, inserted into the matching recesses in the collars, and welded in place to maximize the joint strength. This construction accurately represents an actual machining process for parts of this complexity, which is how the commercialized locks could be manufactured.



Figure 7. Latest iteration of the loop lock.

RFID decision. In the early development stages of the electronic hardware, the Arduino microcontroller was selected to process and operate the necessary electrical components within each smartlock. The Arduino is frequently used in product development for its powerful processing power with a small footprint. Additionally, there is a vast online collection of open-source codes and libraries due to its popularity.

NFC technology was chosen as the access control method for each smartlock. The user will have to present a NFC chip to the reader placed on the smartlock to unlock and use the bike. The decision was made after careful considerations of Bluetooth, RFID, and NFC explored in the literature review. In attempts to streamline the user

experience, the team ruled out Bluetooth technology, which requires pairing of devices. Requiring users to pair personal devices to the smartlock for access control presents a limitation on the scalability and usability of the system. Alternative, electronic payment technologies utilize NFC to communicate between the reader and NFC enabled credit card or mobile payment applications. NFC also requires greater proximity than other RFID systems for communication. The smaller range of operation reduces the chance of attackers to capture or alter the communication, adding to the innate security of the system. These characteristics played an important role in the team's decision pursue NFC technology for user authentication.

Funding. In addition to the semesterly funds of \$300.00 provided through the Gemstone Department, the team sought outside support to supplement the costs of materials and bikes. To do this, the team applied for and received a grant from the University of Maryland's Sustainability Fund. The team received a \$4,000 grant in April of 2014, and the funds were used to purchase raw materials for the XBee network and RFID technology as well as materials to construct, mill, and test locks. The team also submitted applications for the Pepsi Enhancement Fund distributed by the University of Maryland's Stamp Student Union, the VentureWell Grant, and the 2014-15 Sustainability Fund. Unfortunately, for these grants no additional funding was awarded.

Exploration. The exploration subsection follows the primary research of the various technologies by each subteam including initial testing and the ensuing design changes. The Locking subteam started material property testing; created computer aided

design (CAD) models of purchased locks, and completed finite element analysis (FEA) analysis to compare data collected from physical tests to theoretical failure modes. The Access Control subteam acquired a selection of authentication devices and evaluated each technology. Geolocation purchased XBee hardware to test basic operations and to understand XBee product limitations. For the website, the Geolocation subteam tested Ruby, Node, Meteor, and other comparable software frameworks to select the best framework for a stationless bikeshare. Lastly, the team's Business subteam was able to get Institutional Review Board (IRB) approval that would be needed for future focus groups.

Finite element analysis and lock testing. In order to predict and understand failure modes of commercial lock designs, the Locking subteam performed FEA analysis in Solidworks, a computer aided design software. When testing, the machine could exert a maximum force of 25 kN. The FEA analysis provided insight into whether or not the test would break a given lock and the likely method of failure.

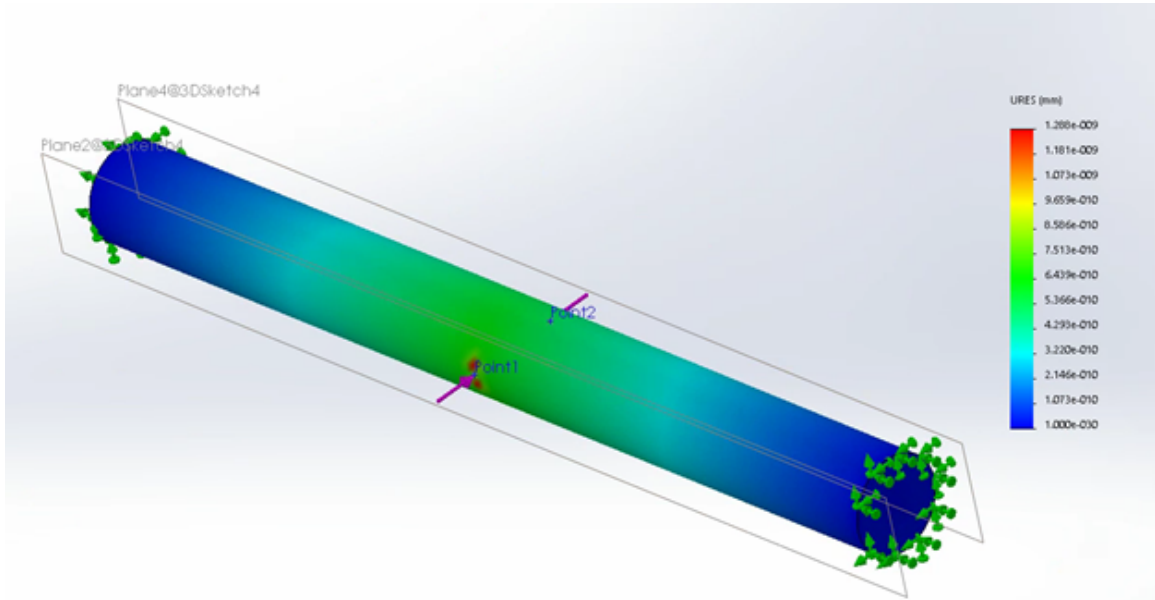


Figure 8. FEA analysis of cutting test on straight portion of U-lock. Each force is 12.5 kN equating to 25 kN.

Figure 8 shows the FEA analysis for the proposed cutting test of the U-lock.

There are fixtures on both ends of the straight portion of the U-lock, like the actual test would incorporate, and two point loads equating to 25 kN of force. This test shows that the U-lock would only displace around 0.002 inches, proving that the test would not break the lock and the cutter would need more than 25 kN to break through the metal. It is also shown in Figure 9 that the OnGuard K9 folding lock would not break with a force of 25 kN.

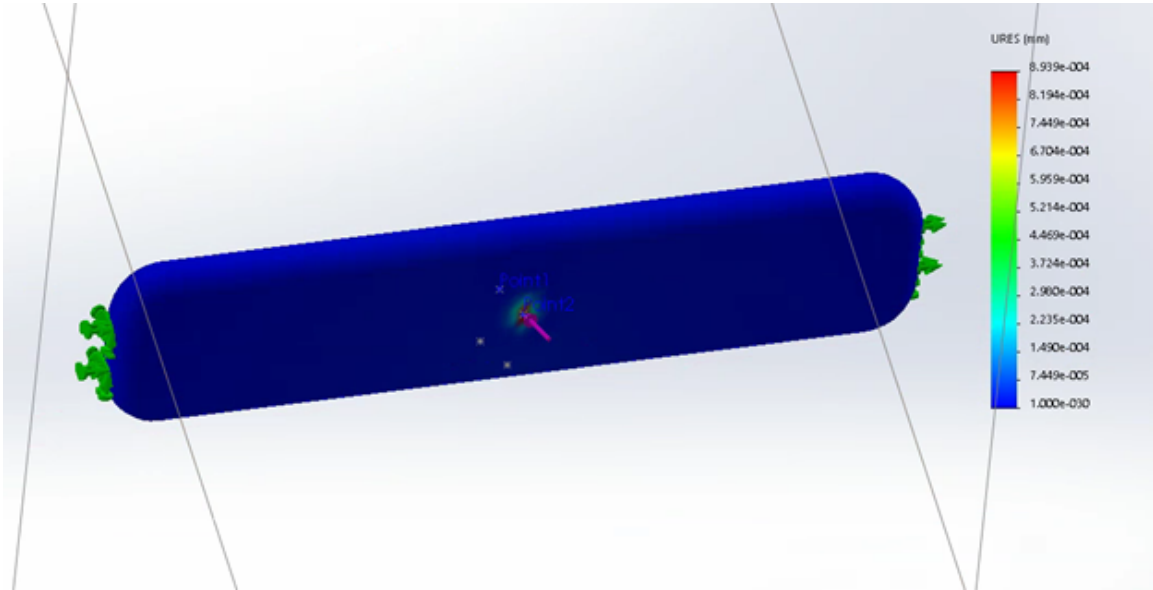


Figure 9. FEA analysis of cutting test on a straight portion of OnGuard K9 folding lock. Each force is 12.5 kN equating to 25 kN.

To test the existing bike locks, Team BIKES needed to design and build fixtures that could interface with the testing machine. To properly interface with this machine, the fixtures had to be able to withstand a load of up to 25 kN and be mounted with a M16 - 2x50 mm Cap Screw depicted in Appendix A. The lock specimens are made of steel and steel composites; therefore, the fixture must be as strong or stronger in order to survive the applied forces without deformation that would influence the test results. To ensure the fixtures have enough strength to handle the tests, the team chose hardened steel with a thickness of $\frac{1}{8}$ in. To accommodate all locks using a single set of fixtures, the team chose to build a pin and plate fixture, which can be seen in Figure 10 below.



Figure 10. Testing fixtures with pins and bolt cutter head piece.

This design uses additive and bonding processes that are far less expensive and time consuming than subtractive processes such as milling through a solid block of steel. Two plates are welded to the inside of each wing in a U-channel of steel. These plates lengthen the wings of the channel, while simultaneously increasing the fixtures strength and stability under load. In order to test the multitude of existing locks, two different sets of plates had to be used. One set of plates was designed for tension testing and were placed on the inside or outside of the U-channel depending on the lock being tested. The other set was designed for cutting the locks with the use of bolt cutter head pieces. The first set of plates, used for tension tests, can be seen assembled with the test fixtures in Figure 11 below.

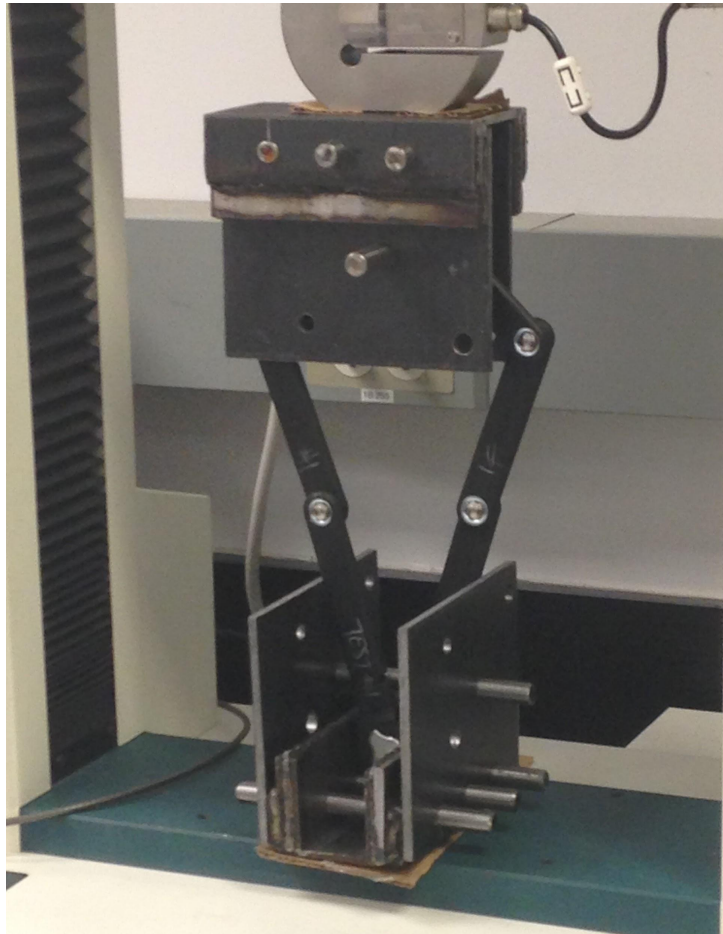


Figure 11. Test fixtures attached to Tinius Olsten model H25K-T benchtop tester during the OnGuard K9 folding lock tension test. The tension plates can be seen on either side of the bottom U-channel. The tension plates are configured on the outside of the U-channel to fit the OnGuard K9 folding lock.

For the tension testing plates, two sets of three $\frac{1}{2}$ in pins were used. One set of pins are shorter in length than the other. The longer set of pins are used to test the OnGuard K9 folding lock, and this allows the plates to be placed on the outside of the U-channel. To pull the locks, an additional $\frac{1}{2}$ in pin is placed in the center of the plates. This pin secures the locks and restricts them from falling out of the fixtures.

Furthermore, the cutting testing plates were designed to allow the bolt cutter blades to fit. To do this, a small rectangular section was cut out on the top left of the plate. Two different sized pins, $\frac{3}{8}$ in and $\frac{1}{4}$ in, were used to secure the bolt cutter onto the plates along with washers. The addition of the washer ensures no moment is induced during testing.

Acquiring NFC materials. Once NFC was selected, the first step towards a secure access system was to select a small, configurable, NFC reader to build the system around. The first model tested was the Adafruit PN532 Reader/Writer available as an Arduino shield that could be used to write as well as read NFC tags. While useful for initial exploration, the dual-mode feature had several drawbacks. First of all, the tag writing feature presents a clear security risk to the system because the writing feature amounts to a broadcast of data sent through the module from the associated microcontroller to any nearby NFC tag. The additional writing circuitry also increased the power draw of the module. The most important shortcoming, however, is that Adafruit module was far too large to fit inside the electronics box for the prototype smartlock. For that reason, the second module tested, the PN532 v3.0 by Seedstudio was selected based on its size- it had a footprint equivalent to that of the Arduino and stacked neatly on top of it. This module had the smallest antenna of any of the modules considered, and the antenna was attached by a short external wire that allowed for more flexibility in the placement of the antenna inside of the electronics box. The Seedstudio PN532 module solved many of the problems of the Adafruit module; it could only read

tags, it had a lower power draw, and it could be more easily configured into a small space. However, the Seedstudio module was ultimately abandoned due to signal integrity concerns that arose when a longer antenna attachment wire was used. In response to this concern, the third NFC module tested was the NFC 2.0 by Elechouse which combined the antenna and reader circuitry on a single small module that is then wired to the Arduino. At approximately a square inch in total size, this reader had a significantly smaller footprint than either of the others, but the wired connection to the Arduino was more robust than the connection between the PN532 v3.0 antenna and body. Before comparing the acquired modules for reliability, ease of use, and power consumption, sizing and packaging concerns alone pointed to the Elechouse NFC 2.0 as the most viable solution.

Every time a new module was acquired, a tag read operation needed to be performed with it in order to assure that the module worked as expected and could be easily integrated into the hardware system by the team in the future. The first step was to write code that would print out the NFC tag number captured by the reader to the serial port of the Arduino. This seemingly simple exercise required team members to gather and install all necessary code libraries and correctly configure any options on the module to operate with the Arduino hardware. After achieving this, the team could move on to a simple “Blink” test where the Arduino checked the tag ID acquired by the reader against a small, internal database of valid ID numbers and blinked a green LED if the tag matched, and a red LED if it did not. Performing both of these tasks with all three

readers ensured that the team had the necessary experience working with each tag to confidently test the properties of the readers.

Though all of the readers had a tag reading distance listed as part of the product specification, the team wanted to confirm that these properties held true under non-ideal conditions. Simple tests were established to determine the distance at which each module could read a tag and the power it consumed. For the distance test, a module was connected to an Arduino with a simple program that read the tag and printed the result to the Arduino's serial port. The module was then placed so that its face was perpendicular to the surface of a table and powered on. Starting at a distance of 20 cm from the reader, the NFC tag was moved closer to the readers at intervals of 5 mm at a time, until the reader was able to recognize the tag. This test was performed for each acquired module and repeated with different materials in close proximity to the NFC reader or tag. For instance, placing a tin sheet within a centimeter of the NFC reader made tag reading distance random and uncontrollable. The closer the metal, the shorter the reading distance has to be. Additionally, the team confirmed that even a thick sheet of plastic separating the NFC tag and reader would not decrease the reading distance or accuracy. The team also measured the operating voltage of the NFC reader modules to determine how far the voltage could be decreased and still ensure correct readings. Additionally, the operating current of each module was measured for consideration in the final system design. Figure 12 summarizes the measured results of the three aforementioned modules.




Module:	Range:	Operating Voltage & Current	Antenna Size	External antenna
PN532 NFC RFID Module 	3.2cm	3.3V or 5V 70mA	42.7mm x 40.4mm	Yes
Adafruit PN532 NFC/RFID Controller Shield 	5.7cm	5V	65.1mm x 53.3mm	No
NFC Shield V2.0 	5 cm	5V 100mA	30.48mm x 27.94mm	Yes

Figure 12. Comparison of tested NFC modules.

XBee. After deciding to pursue the ZigBee protocol, the team researched available modules. The Geolocation subteam referenced the XBee Buying Guide on Sparkfun Electronics website, which aided in deciding which modules to purchase. The system requirements that informed this decision included:

1. Mesh network capability
2. Sufficiently long transmission distances for a large coverage area
3. Minimum per module cost

Other factors considered were the antenna type (duck, whip, or pigtail) and the frequency (2.4 GHz or 915 MHz). Comparison of the antenna types can be seen in Figure 13. The frequency had the most effect on the data rate. The 2.4 GHz band had a higher data rate (250 kbps) than the 915 MHz band (up to 156 kbps).

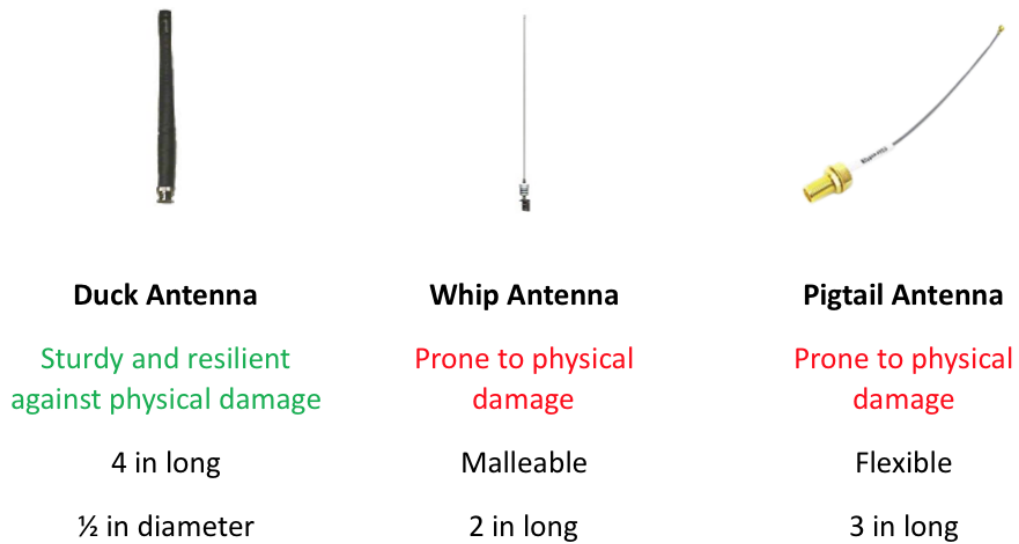


Figure 13. Comparison of antenna types

After reviewing the buying guide, the Geolocation subteam purchased the XBee Module Series 2B with various antenna types and frequency ranges for experimentation. In order to interface with the modules, the Geolocation subteam purchased breakout boards that transitioned the pin spacing to a width that allowed for easier prototyping with a breadboard and added a direct USB interface.

Digi-Key's XCTU XBee desktop application. Once the XBee modules were acquired and configured, the Geolocation subteam began exploring how to interface with the modules. Digi-Key's XCTU application proved to be the best technique for interfacing with the modules for all of the configurations. The subteam found that there were many different configuration parameters with various values and options for each parameter. Some important configurations explored were the PAN ID, which served as the unique ID of the network that all of the modules share, the interfacing options of AT versus API, and the firmware options of the end device, router, or coordinator.

Firmware options. The firmware options include two main components: node type (end device, router, or coordinator) and mode type (AT or API). The Geolocation subteam team explored the differences in types and how they would affect the communications network. The differences in node type are defined above in the Literature Review; however, their relation to the command mode type was explored in detail. AT mode, or transparent mode, acts as a direct serial connection from the TX port of one module to the RX port of the next, much like a physical wire connection. AT mode is useful when a message is being sent through an XBee module as if the XBee module acts as a physical wired connection. API mode, or command mode, is more robust in that a command is sent to the XBee module rather than simply sending a message through it. When a message is sent from an XBee module in API mode, the user must send a ZigBee packet that includes the command to the module representing that a message is to be transmitted, the message payload, and other fields as necessary, such as

destination address. Leveraging the ZigBee protocol in API mode allowed for more capabilities and control over the messages being sent due to the possibilities to add additional fields such as a specific destination address. Additionally, other commands could be utilized to gain further capabilities such as requesting network information or the signal strength of the last received packet.

The different firmware allow for compatibility between variations, meaning an API mode coordinator could send a message to an AT mode end device. For the desired communications system, the transmissions from the end devices simply need to send string messages formatted by an Arduino in the smartlock, as well as receive strings formatted by the website application. Since this transmission requires no advanced features, the end devices were configured in AT mode to reduce complexity. The routers' purposes were simply to relay messages, so they too were configured in AT mode to reduce complexity. However, the coordinator requires more complexity due to its role as the central node in the network and the bridge from the bikes to the website. Because of the increased complexity, the coordinator is configured in API mode.

IRB approvals. In order to carry out a focus group discussion to gather UMD students' feedback regarding interest in bikeshare and progressing smart lock design, the Business subteam sought to acquire IRB approval, which reviews research involving human subjects. The board monitors the human-subject research to protect those tested and to prevent harmful or coercive situations. Once the IRB approved the application, the team organized and carried out the focus group. With guidance provided by the

Gemstone Department, the Business subteam completed two forms: a Human Subject Determination Form (HSD) and a copy of the focus group questions. The HSD form was a way to determine if a full review would be necessary or if the planned focus group discussion could be exempt based on the minimal risk posed by the nature of the research and proposed questions. In February 2015, the IRB found that the planned focus group discussion did not fulfill the requirements of human subject research, and the team could host the focus group.

Experimentation. Once each subteam gained technical proficiency with the selected technology, each component could be further developed to be a reliable component of the future bikeshare system. The Locking subteam used the custom fixtures to conduct materials testing on commercial bike locks. The Access Control subteam tested the viability of solar cells as an energy source and experimented with encryption protocols, NFC tags, and Arduino coding styles. Geolocation experimented with collecting the data needed for each bike using a GPS module and designed their database structure.

Lock testing results. The Locking subteam gained firsthand experience by test fixtures to compare different bike locks' material strengths. The Locking subteam tested the Kryptonite Keeper 12 U-lock and the OnGuard K9 folding lock using Dr. Bonenberger's material testing lab at the University of Maryland, College Park. These selection were based on common lock styles viewed around the University of Maryland campus. The campus Department of Transportation sells the Kryptonite Keeper 12, and

was selected as our first benchmark lock for testing. The OnGuard K9 folding lock is representative of the folding style locks used on campus based on visual inspection. The team's custom-built fixtures interfaced with the Tinius Olsten model H25K-T benchtop tester in Dr. Bonenberger's lab, allowing for tension testing on both locks. The test machine could provide a maximum force of 25 kN, which was enough to break both types of locks. The following graphs represent the relevant data gathered from the Kryptonite Keeper 12 U-lock tests.

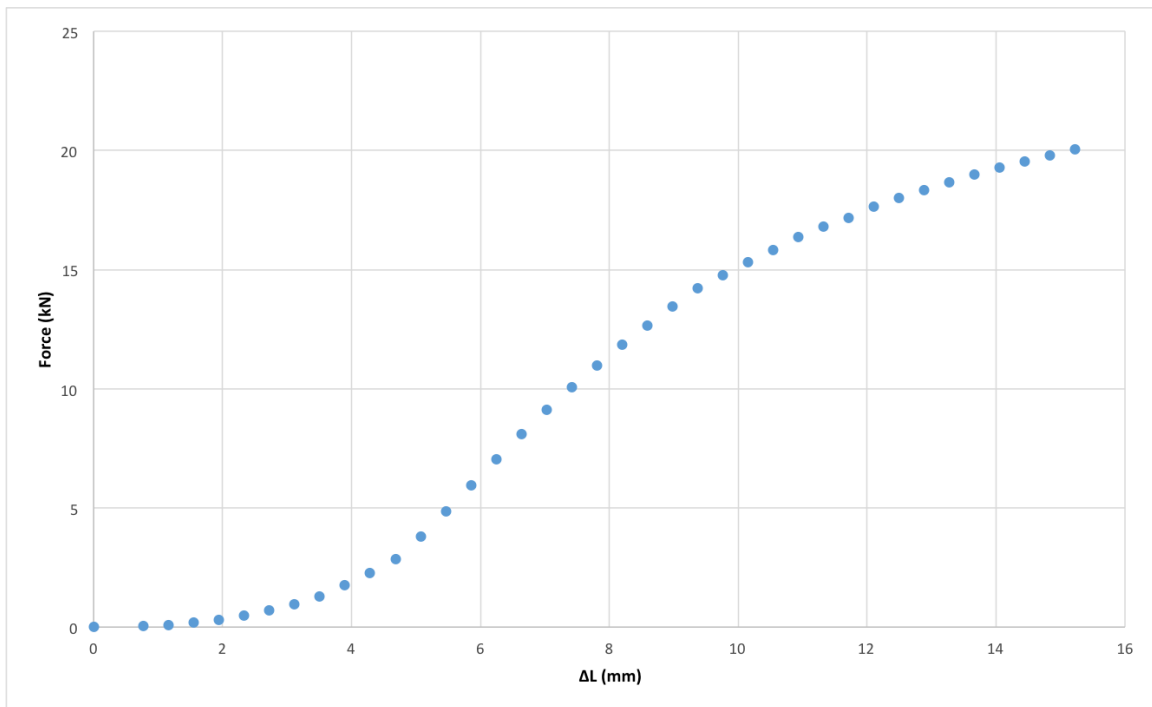


Figure 14. Kryptonite Keeper 12 U-lock tension test one - force vs change in length. This figure shows U-lock tension test 1 results, picturing every 150th data point for ease of reading.

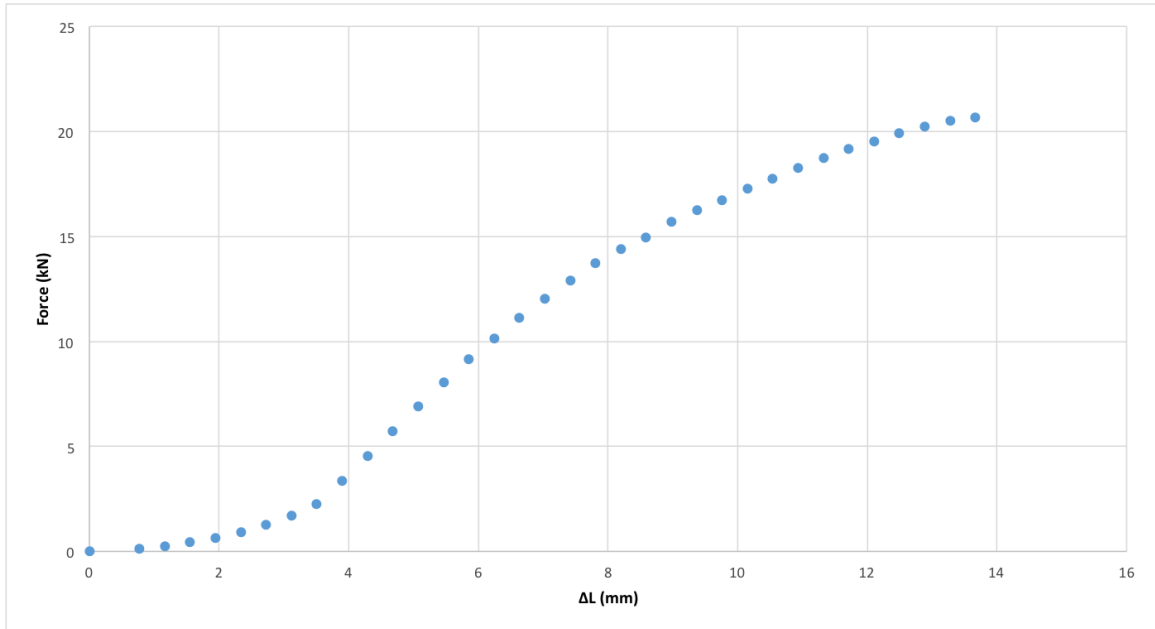


Figure 15. Kryptonite Keeper 12 U-lock tension test two - force vs change in length. This figure shows U-lock tension test 2 results, picturing every 150th data point for ease of reading.

In the U-lock testing graphs, the line begins with a shallow slope. This indicates that a large ΔL is achieved with a relatively small increase in force as the machine begins applying a tensile force on the lock and any gaps between the components of the machine are tightened. When the material begins to deform, the slope of the line increases as more force is required to achieve the same change in length. At the end of the graph, the slope of the line decreases again as the lock begins to fail and the machine needs less force to continue deforming the lock. In the first trial, the lock failed at 20.07 kN of force, snapping apart at the bent foot side of the U-shaped shackle. In the second trial, the lock failed at 20.65 kN of force, also failing at the bent foot. This means that the weakest part of the lock, when subjected to a tensile force, is the interface between the shackle and crossbar on the bent foot side, the side that is not fixed by the U-lock's key-

actuated locking mechanism. If both sides of the shackle were secured by the locking mechanism, as in the case with the Kryptonite Fahgettaboudit U-lock, the lock would be expected to withstand a greater tensile force before failure.

These force vs change in length graphs are comparable to stress-strain curves because stress is equal to force divided by cross-sectional area and strain is equal to change in length over original length. Since the lock dimensions, cross-sectional area, and original length do not change significantly over the course of the test, these graphs will have about the same shape as a stress-strain curve. The line follows the expected form of a stress-strain curve with slight differences caused by the fact that an assembly of multiple pieces is being tested rather than a single piece of material. This difference causes the beginning of the graph to have a smaller slope as the pieces of the assembly shift to allow the maximum possible displacement without undergoing material deformation. Once the slope of the graph becomes larger and linear, as can be seen in the middle section of the above graphs, the curve follows the expected pattern of the elastic region of a stress-strain curve. In this region, the material deforms at a constant rate with respect to the force applied, and the material will regain its original dimensions upon the removal of the force. When the slope of the graph begins to become smaller, curving downward, the material enters the plastic deformation region. In this region, less force is needed to deform the material and the material becomes permanently deformed, meaning that if the force was removed the lock would not return to its original dimensions. The graphs end when the lock failed and the machine stopped reading force and displacement

values. Figure 16 below shows one of the U-locks after testing. Note the bent foot portion has popped out of the crossbar.



Figure 16. Kryptonite Keeper 12 U-lock after tension testing. The lock failed at the interface between the shackle's bent foot and the crossbar.

The team also performed two tension tests on the OnGuard K9 folding lock. The team modified the test fixtures to fit the folding lock, again using the Tinius-Olsten model H25K-T benchtop tester. The graphs below represent the data gathered from these tests.

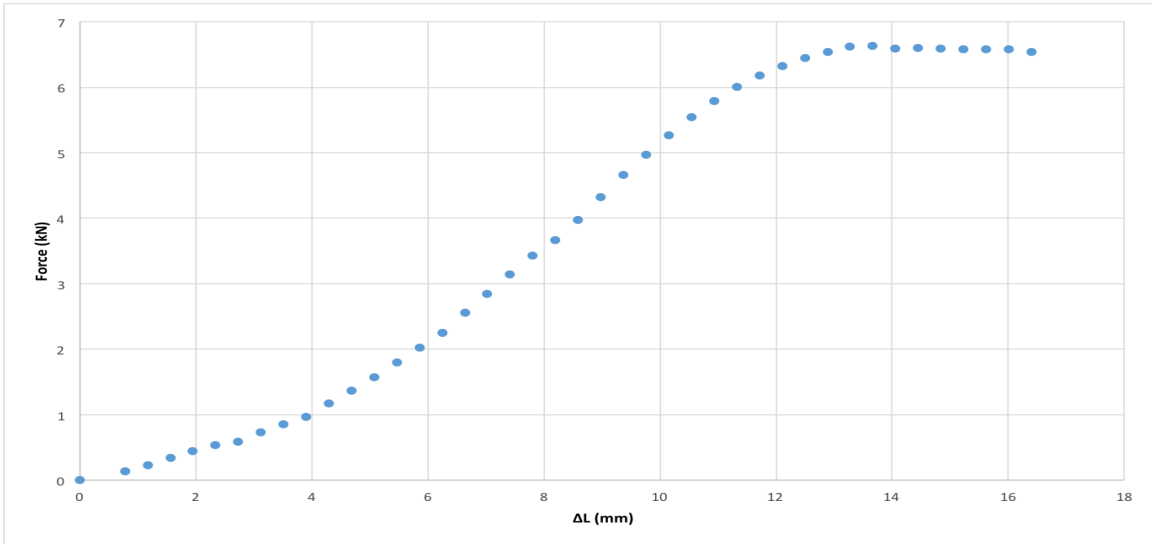


Figure 17. OnGuard K9 folding lock tension test one - force vs change in length. This figure shows folding lock tension test 1 results, picturing every 150th data point for ease of reading.

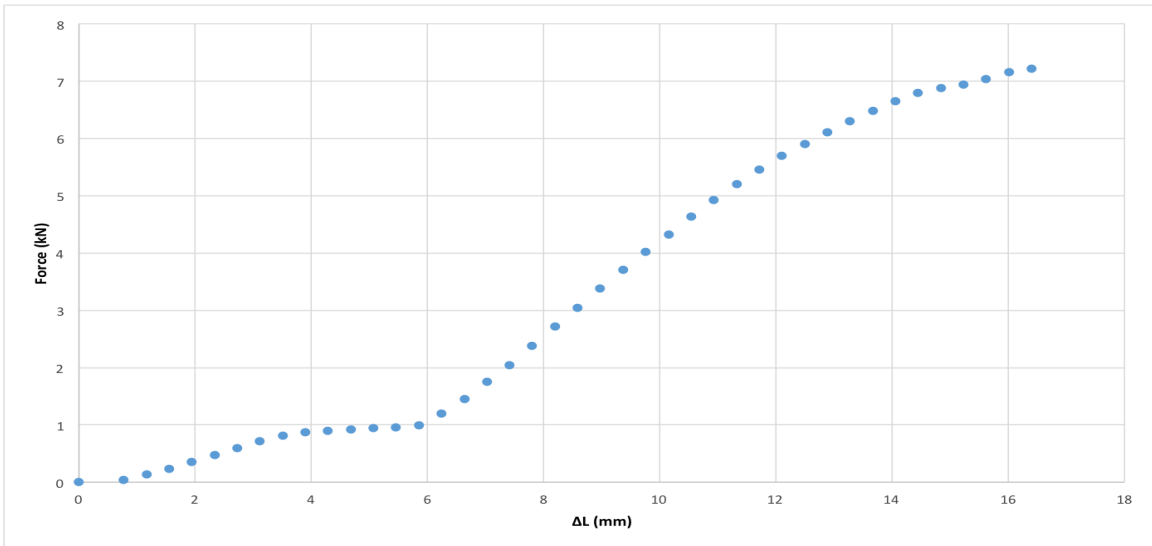


Figure 18. OnGuard K9 folding lock tension test two - force vs change in length. This figure shows folding lock tension test 2 results, picturing every 150th data point for ease of reading.

The folding lock test graphs follow the same general form as the U-lock test graphs, with noticeable variation in the early stages of testing. In the first folding lock

test, the graph started out with a larger slope, quickly entering the elastic deformation region. The second test began by approaching zero slope before jumping to the linear, positively sloped region expected for elastic deformation. This difference was caused by the large amount of moving parts in the OnGuard K9 folding lock. In contrast to the U-lock, which is composed of only two parts, the folding lock features six different sections connected by hinges. This allowed more freedom of movement in the lock in the initial stages of the test before the lock reached maximum elongation and began to deform. Another difference between the U-lock tests and the folding lock tests is pictured in folding lock test one. At the far right end of the graph, the data reaches a peak before sloping downward and reaching failure. This peak indicates the ultimate tensile strength of the material, as can be seen when the applied force decreases but the deformation continues to occur. The first folding lock failed at 6.507 kN after achieving a maximum of 6.661 kN and the second folding lock failed at 7.154 kN. Figure 19 below pictures the folding lock during testing.

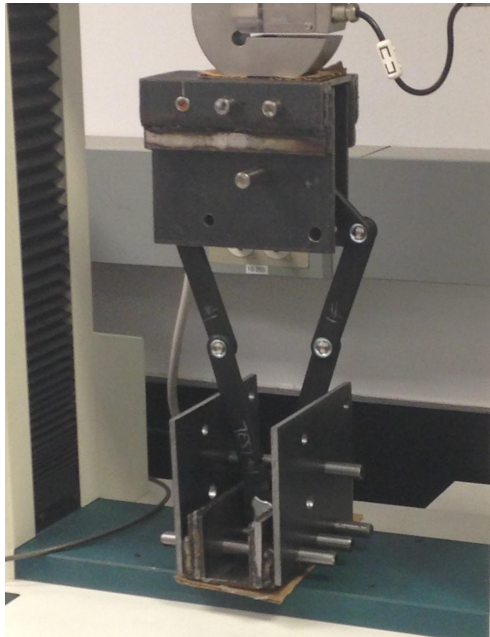


Figure 19. OnGuard K9 folding lock during testing.

Since the U-locks were able to withstand forces of about 20 kN in both trials before failure while the folding locks were only able to withstand forces of about 7 kN, the team learned that the U-lock's design is much more resistant to tensile forces than that of the folding lock. The U-lock's failure at the bent foot portion of the shackle indicates that if that side of the shackle was also secured by the U-lock's key-actuated locking mechanism, it is likely that the lock could have withstood a greater tensile force before failure. The team used this information to develop the Locking subteam's preliminary designs, which will be discussed in the implementation section of the Development chapter.

GPS. In combination with a secure lock, the bikeshare will need real-time location information. The Geolocation subteam began researching and experimenting

with a GPS module. The GPS module was connected with an Arduino and operated using the National Marine Electronics Association (NMEA) protocol. The NMEA protocol uses a comma-separated value (CSV) string to display different strings of information. Luckily, an existing software library allowed for easy parsing of the NMEA data received over the GPS module that allowed for fast-acquisition and accurate GPS readings.

Iteration

The iteration section addresses the evolution of the research from initial discovery into refined concepts based on user feedback. The focus group discussion provided a majority of the feedback on the team's experimentation. The Locking subteam received feedback on the lock design to better accommodate the user. The Business subteam received feedback on the type of payment that college students would find acceptable for the bikeshare. The Access Control subteam learned the optimal placement for the NFC reader on the bike for the best user experience. Lastly, the Geolocation subteam received feedback on the bikeshare website and what components were missing or needed improvement.

Focus group. The team realized that it wanted to obtain constructive information from prospective users of a college bikeshare system. This information would be used to better understand the process of developing and implementing a college bikeshare system. From the participant feedback, the team gained a better grasp of the bikeshare environment on a college campus as well as developed customer requirements to

compare with existing engineering characteristics in order to assist with future product generation.

The focus group discussion took place on April 30, 2015 with ten participants. The team started with a general introduction to bikeshares and the project. Four participants had bicycles on campus and three had used bikeshares in the past. The Locking subteam introduced the participants to the various lock designs and gave a demonstration using a 3D printed lock, locking the bicycle to a bike rack.

The majority of the focus group discussion time was spent asking the participants questions about bicycle and bikeshare usage on campus. The team asked if the participants would use the system on campus; three participants said yes, one said he would use it in emergencies, and another said that if the service was subscription based, yes, but if it was by a pay-per-use system, he would not. This led the discussion into the price of current bikeshares and the price of Team BIKES' proposed system. When the participants were asked to guess what they thought Capital Bikeshare costs per year, they answered \$80 per year and \$200 per year; however, in reality, the subscription is \$75 per year. The team explained that there are many different methods of pricing, such as a flat rate per amount of time (e.g. \$75 per year, \$10 per month, etc.) and charging extra if the bicycle is used over a certain amount of time. Participants expressed concerns about a lack of knowledge by students regarding bicycles in general on the Maryland campus. Few of the participants informed the team that they were unaware of the campus' policy that all bicycles needed to be registered with the university. This suggests that

information regarding riding bicycles on campus may not be reaching all students. In order for the team's bikeshare to be successful, students must first understand how it works. To market the bikeshare, a participant suggested offering students one-time free rides as a trial experience, which could attract prospective users. They suggested to partner with University Recreation and Wellness or Images to set up specified trails for "bike tours" with the bikeshare bicycles, allowing prospective students to try the bikeshare during orientation. The team explained that the current plan is to follow a model similar to Sobi where there is a student rate around \$25-30 per semester as well as the possibility of obtaining daily passes and unlimited riding time on campus.

An issue that was brought up during the focus group discussion was that the current plan for the bikeshare is limited to on-campus use only. The team explained the ZigBee network limitations regarding off-campus bikeshare usage and that the purpose of the bikeshare should be to get around campus instead of getting around off-campus. However, many participants lived off campus and expressed that their need to have a bicycle on campus is because they live off campus.

Finally, the team explained the idea of "bike deserts," sections of the campus that may have a lack of bikeshare bicycles because students will tend to bike to certain areas over other areas. This results in an asymmetric spread of bikeshare bicycles across campus, making it challenging for students to find bicycles if all of the bicycles have been used and left at another location. The team's plan is to involve the students in redistribution of bicycles at certain locations, such as the top of Stamp hill. Participants

suggested a reward system involving discounts off the bikeshare service if students move bicycles to these bike deserts. Another participant suggested making redistribution a competition by rewarding frequent riders or hiring students as dedicated redistributors.

Lock feedback. During the focus group discussion, the team showed the participants both of the lock concepts and demonstrated the locking mechanism and concepts. For the loop lock, the participants thought that it looked very secure and robust. However, they expressed concern that the lock may rotate while riding and present a threat to the rider by impeding leg motion.

For the modified U-lock concept, the participants seemed very open to the design due to its close resemblance to a typical U-lock, which they were used to seeing around campus. They also thought that the locking mechanism was reasonable, and the cable would not impede the locking motion. However, similar to the loop lock design, the participants expressed concern over the lock impeding the pedaling motion of riders and suggested that the width of the lock be adjusted accordingly.

New user onboarding. During the focus group discussion, the team received a variety of feedback related to the website and the system operation. The feedback focused on improving the first web page to be more informative and concise, implementing useful user features, such as a reservation system, and more clearly defining a brand identity. Prior to the focus group, the team had created a brand using the name, RedBar. The name was inspired by the red marks in between a student's

university schedule indicating a difficulty in arriving at a class on time, as seen in Figure 20.

Term: Spring 2015

	Mon	Tue	Wed	Thu	Fri
10am					GVPT289A 0105 Dis 1222 LEF
11am	MATH240 0132 Lec 0131 ARM		MATH240 0132 Lec 0131 ARM		MATH240 0132 Lec 0131 ARM
12pm	CMSC132 0103 Dis 2107 CSI		CMSC132 0103 Dis 2107 CSI		
1pm					
2pm	GVPT289A 0105 Lec 2102 SHM	MATH240 0132 Dis 0307 MTH	GVPT289A 0105 Lec 2102 SHM	MATH240 0132 Dis 0307 MTH	ANTH240 0202 Dis 1130 WDS
3pm	CMSC132 0103 Lec 1115 CSI	ANTH240 0202 Lec 1400 MMH	CMSC132 0103 Lec 1115 CSI	ANTH240 0202 Lec 1400 MMH	CMSC132 0103 Lec 1115 CSI

Figure 20. A picture of a typical student’s schedule.

Many comments from the focus group discussion addressed the “About” page or the home page shown when a user first visits the website, as shown in Figures 21 and 22. The comments focused primarily on color scheme and visual appeal of the website, and the participants also stated that the website failed to convey the team’s unique brand. The users additionally felt that the concept of a stationless bikeshare was not effectively

communicated or explained on the first page, stating that someone outside of the focus group discussion would likely be confused. They suggested improvements such as a visual walkthrough, diagrams, or other more general text to better accomplish this goal. This user feedback demonstrated the user focus on onboarding and highlighted an issue found on many websites.

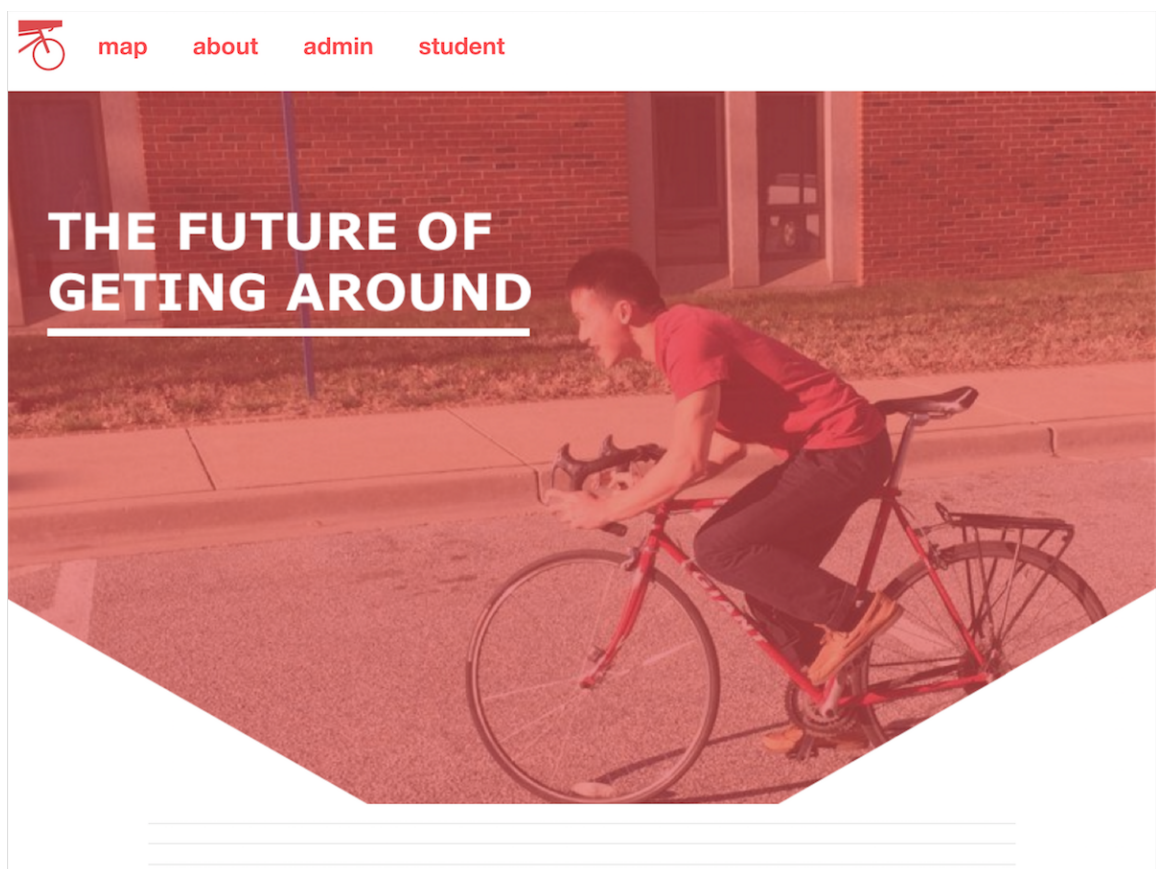
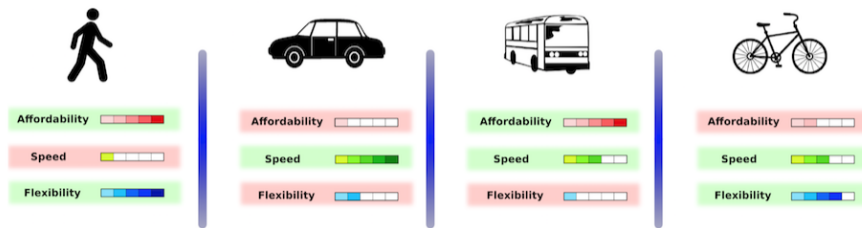


Figure 21. The top portion of the about page with a hero image and nav bar.

Currently Available Modes of Transportation

Each mode has a substantial **tradeoff...**



What if we create a **balanced** mode that is **affordable, fast, and flexible at the same time?**

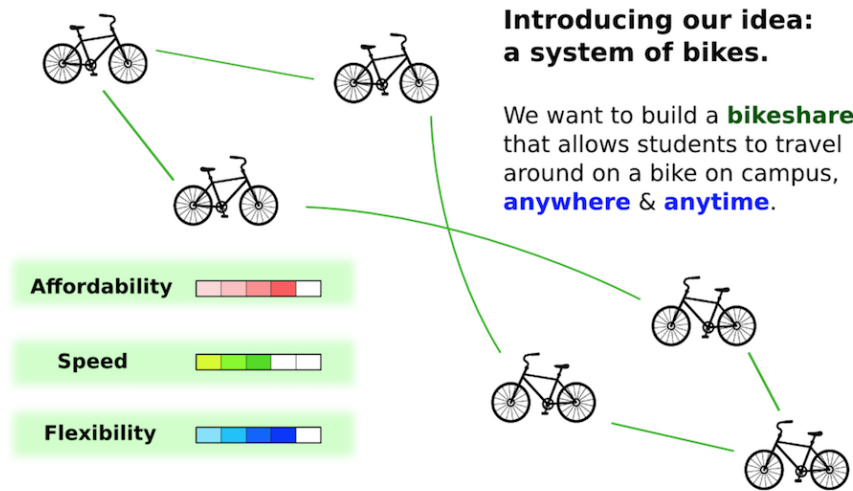


Figure 22. The middle section of the about page including additional details.

Reflecting on the feedback, the team focused on the development of better content focused towards the user needs. First, the users wanted to understand a stationless bikeshare and what makes RedBar unique. The revised layout reflected this change by first focusing on a clear tagline for the company: “bikes on demand for the busy student.” Second, the “About” page incorporated visual content and short, well scripted text blocks to quickly explain what a bikeshare is and to introduce RedBar. Most of the initial content prior to the focus group discussion was removed to focus on the most crucial aspects: describing a bikeshare, explaining the value proposition, and displaying the price structure. One of the minor issues identified by the users was the need for a more cohesive color scheme and brand. While the team originally had a logo in the top left corner and throughout the website, the users did not recognize the brand name, so the full name, “RedBar”, was highlighted, rather than obscured through a logo.

The content removed from the “About” page would still be useful for certain use cases, such as those who wanted to learn more about the system or universities interested in implementing the RedBar system. The information was moved to a separate series of pages focused on different use cases. An example of this useful information is a three section callout that discussed convenience, where users would no longer need to worry about a flat tire; easy access, bikes available anywhere a user may want one; and simple return, to stress the stationless component of the team's bikeshare compared to a station-based systems, such as Capital Bikeshare. The team additionally highlighted how the

wireless communication system works, the smartlock internals, and open-source code released from this project, among other topics.

Interactive map. The focus group discussion provided feedback regarding the map, arguably the most important component of the application. Some comments, “I like how that works” and “it is like ZipCar”, showed positive support for the direction the team was heading. Additional comments brought up potential issues. One comment questioned what the range was and mentioned that the map icons were not immediately intuitive. Several comments focused on the appearance of the map and discussed the popular University of Maryland map application called TerpNav and requested a similar minimalist map tile base. The focus group discussion members brainstormed useful features such as a nearby bike indicator that would show: “nearest bike: 5 mins from current location” or a reservation system to guarantee a bike’s availability. The comments encompassed many components, both visual and technical, but all focused on usability.

To address the comments, the team made simple visual changes to the map tiles and icons but continued the approach to rethink the entire user interface. Adding interactive buttons to the map tile, the team rethought the user’s interaction around a reservation feature. The core use case was for a user to visit the website, browse for a bike in a convenient location and immediately reserve it. Learning from the comments that the original map icons were unintuitive, the team thought about ways to make reservation interaction interactive. Borrowing a feature already used in almost all map

applications, the user can tap an icon to reserve. To inform the user of the functionality, a proposed technique was to add a bounce indicator and notification to direct new users to the reservation process.

The reservation system proposed technical challenges along with the user experience and interface challenges. To successfully implement this feature, there were several desirable characteristics, such as reactivity, database design, and error handling. To accomplish reactivity, the team needed to indicate that a bike had been reserved for the user and quickly update all other users' screens to avoid any duplicate rentals. Additionally, all users need not be required to reserve a bike if they are within eyesight of an available bike; however, a reserved bike cannot be checked out by a user who is in the vicinity. These contingencies helped shape the reservation system.

With the contingencies in mind, the reservation system was made non-binding. A user would only reserve a single bike at a bike rack, but not an exact bike. This way a user would not have to find the bike they reserved and could simply check out any bike they see. The reservation only guarantees that a bike will be at the location specified. So if a user, who does not have a reservation, attempts to check out a bike that would reduce the available bikes at that rack below those reserved, the walk-up user will be denied. To provide more transparent explanation, a notification would be sent to the denied user with an explanation and link to the next nearest bike. If however, there were an excess number of bikes beyond those reserved, the walk-up user could reserve the bike with no interruption.

To handle the error cases, the team would be able to use the non-binding component of the system. If users attempted to reserve bikes at the same location, the application would look for an alternative bike and alert any users with an alternative. This may arise if there is any latency in the system or a user's connection is throttled. While not a likely scenario, error-handling such as this is a critical feature for a robust system. An additional time limit control needed to be placed on rentals to prevent a bike from being reserved for an infinite amount of time. Using a Cron packaged for node.js applications, any time a rental was started the task would fire and clear the bike rental after three minutes of no action. Additionally, all bike rentals are checked on a repeating schedule, so in case of an error, a bike can never be reserved for greater than half an hour.

Critical Feedback. Not all features were received positively. One such feature, a fitness component tied to each user's account would have shown data for the length of a ride, vertical climb, and average speed for each user. The Geolocation subteam imagined this as a fitness tracker and a way for users to receive additional value from the system; however, few focus group discussion members expressed any interest in the feature and most saw it as an unnecessary feature. This was a useful lesson in privacy and the interest in college users to maintain some level of anonymity.

User interface philosophy. The feedback from the focus group helped define a general user interface philosophy to unite and focus the web application's development. Based on feedback from the focus group, it became apparent that building fast was not the only metric for success. Instead, the Geolocation subteam gained a finessed and

refined understanding of user needs and product development for usability and experience. With this new knowledge the constraints changed. First, the team realized that greater focus needed to revolve around product design to deliver a clear user interface that users could easily understand and interact with naturally. Second, the team needed to focus on minimalism by reducing the bloat of pages, the number of scripts, and removing unnecessary content. Finally, the team learned that users would use both desktop and mobile devices, so the website needed to scale responsively on all platforms.

To deliver on the first priority, product design and branding, the Geolocation subteam invested the time to overhaul the about page and rethink the initial user experience. The lessons learned from this overhaul and subsequent custom stylesheet process from Macaw to Stylus, laid the groundwork for subsequent overhauls on the admin and employee pages. For the second priority of minimalism, the Geolocation subteam aspired to reduce the number of tabs, links, and pages across the application. For the about page, the Geolocation subteam reduced much of the content on the page and created additional pages with focused content tailored to specific use cases.

Additionally, based on feedback received from the focus group, the Geolocation subteam began minimizing the number of pages needed to create an administrator's experience. Narrowing down to a single page, the admin can scroll past several graphs of highlighted data before viewing the detailed data of each bike. From the tables, the administrator can sort, search, and open in-depth panels that explore the bike information in greater detail. To accomplish the third priority of responsive styling, the Geolocation subteam focused

on creating unified elements between mobile and desktop, while emphasizing the core features that were expected in each. For example, on mobile, users are accustomed to native applications offering a horizontal navigation bar that can quickly link the user between the minimalist set of three pages (for a user), while on desktop, a top-fixed navigation is more common. The core content of the application is thus consistent between mobile and desktop. The sliding panels that allow for in-depth analysis on the admin pages are identical on mobile.

Visual development pipeline. Prior to the focus group discussion, the Geolocation subteam relied on the Foundation framework, but found the framework too visually constrictive and instead opted for the more mature framework, Bootstrap. To further gain visual freedom and avoid conflicts, the Geolocation subteam selected components from Bootstrap while building custom CSS elements in parallel, rather than writing workarounds to override the Bootstrap enforced styles. To make this possible, the Geolocation subteam adapted open-source code to make a customizable version of Bootstrap available to Meteor as a complete package. The package worked by collecting user preferences through a JSON configuration file, parsed components from the large Bootstrap-Stylus GitHub repo, and selected which code blocks would be written to a file inside the user's local meteor directory. This package proved exceedingly useful for rapid website prototyping and was released under the name: "kyleking:customizable-bootstrap-stylus" and has over 145 users - available open-source at <https://atmospherejs.com/kyleking/customizable-bootstrap-stylus>.

Utilizing the experience gained from developing version of the site in Foundation, Bourbon, and Bootstrap, listening to user feedback on the about page, and researching the design of similar company websites, the Geolocation subteam establish a clear brand identity and create custom stylesheets. To accelerate the development, the Geolocation subteam used Macaw, an application that allows a website to be built with an interactive user interface (UI) and scalable code to be generated on demand rather than developing the code with a text editor. This UI-first process was beneficial in quickly iterating through visual prototypes by mocking the user interface, sample text, images, and output functional code with proper CSS classes. Additionally, breakpoints could be set to demonstrate how the UI elements would cascade on different browser sizes from desktop to mobile. At different browser widths or breakpoints, the elements would have a new style, such as 100% width on mobile screens, vs. 33% width on desktop computer screens.

From this information, Macaw would create CSS code, but needed additional processing to be incorporated into a Meteor application. First, the static assets and pictures must to be moved to the special Meteor, “public” directory inside of an “img” directory as specified by Macaw in the exported HTML files. Second, the HTML code needed to be appended in <template> tags and added to the application’s client folder and routing tool to be incorporated into Meteor’s rendering system. Likewise, any HTML headers needed to be removed to avoid conflict with the default header set elsewhere in the app and the doctype set by meteor. Third, the javascript files exported by Macaw had

to be wrapped in template rendered blocks to prevent interference with other web pages. Upon each new iteration of the design, only the CSS, style files and inner HTML content needed to be updated. This workflow liberated the team from spending countless time manually building the website's component, while allowing the subteam to establish a visual identity and highly accessible web application.

After the completion of the Macaw design phase, the code needed considerable refactoring to both port the code into one of the stylus and jade pre-processors and to refactor the code for meaningful editing. The former is easy to do with an online tool js2coffee or the HTML2Jade Converter, while the latter takes greater manual input. Although the refactoring step is redundant because the code was created in Macaw, it has a faster workflow, allows for a greater number of iterations inside Macaw, is better organized, and is more concise than if written from scratch. This process is faster and allows for a greater number of iterations. Once refactored, the code is ready to be manually edited from text file with ease, with simple variable changes or quick edits to concise components.

Restricting access through roles. To develop security and prevent users from accessing another's data, or worse, the public accessing user data, the Geolocation subteam implemented a role-based authentication system. This system was first applicable to the map application. In order to reserve a bike, a website visitor must be logged in and in the "user" role. If a visitor does not meet the requirements and clicks to reserve a bike, the request is denied and a UI alert appears. The use of the "user" role

simplifies account management. An alternative approach would be to provide individual exceptions for each user. However, the role-system allows for user-base wide changes to be made in a controlled and intuitive fashion to clearly defined roles. Another application of roles serves to provide administrators sufficient access, when necessary. The “admin” role provides access to information including personally identifiable information and would only be used to contact a user, such as in the event a bike is stolen and the user is the last known user. Staff like a mechanic have the “employee” role and have complete access to the inventory of bikes including those available or in need of repair. The role-based system provides a sane way of managing a complex and potentially shifting backend.

Implementation. Incorporating the focus group feedback, each technical aspect of the system was further developed. The lock design was altered to increase user friendliness and reduce manufacturing costs without compromising security. The lock design also incorporated design changes to accommodate the electric systems inside the attached box. The NFC reader deemed best fit for the intended smartlock access control system was determined. Moving forward, the wireless sensor network was further tested in order to find the best method of communicating through the XBee modules. Data was also collected on the range of the individual modules to provide information on how to smartly place the routers in the final system design.

Evolution of the modified U-lock. In parallel with the loop lock, Team BIKES decided to develop a bike lock that was based on a U-lock, the current safety standard

recommended by DOTS at the University of Maryland. Therefore, the design incorporates the structural integrity of the U-lock with a solid loop enclosure. However, the conventional U-lock separates into two parts when opened, which is not desirable in a bikeshare system because detachable parts can be easily taken or lost. As a result, a cable extension connecting the loop enclosure and the lock base is proposed to allow for flexibility during locking while also keeping the whole lock a single system. The lock relies on a reel system to ensure security of the cable and for improved usability.

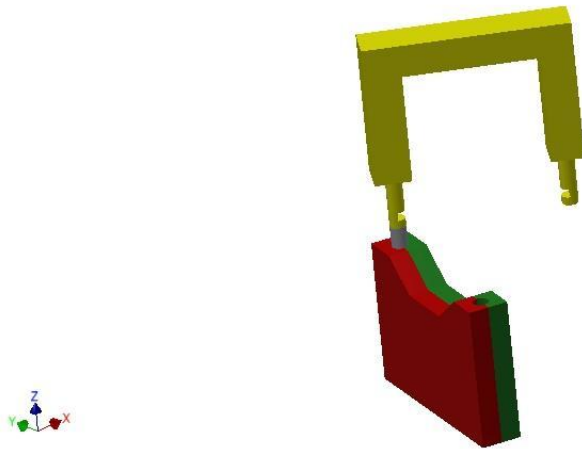


Figure 23. The first version of the modified U-lock.

The first iteration, as shown in Figure 23, was used to gauge the shape of the locking mechanism and whether a retractable pin mechanism would work correctly. Through FEA analysis, the team found that the design would withstand a reasonable range of forces. After rapid prototyping, Team BIKES demonstrated the feasibility of the lock design and novel locking mechanism. The locking mechanism consisted of two pins

that were angled on one side and pushed into a locking position using a spring mechanism. When retracted by a motor, the lock could easily be unlocked, but in the normal, passive state, the pins would prevent unwanted access, when the U-portion of the lock was re-inserted. The pins allowed for minimal electronic interaction, which creates less power draw and longer battery life. This prototype also led to the discovery that the cable connection between the loop enclosure and the lock housing had to be longer in order to provide proper flexibility during locking.

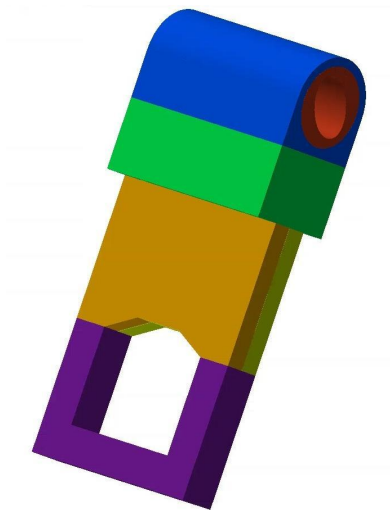


Figure 24. The second version of the modified U-lock.

The second iteration of the modified U-lock, as shown in Figure 24, involved incorporating an electronics housing unit on top of the locking mechanism to allow for close electronics integration. After an additional round of additive manufacturing prototyping, the team learned that the overall size would possibly restrict the cycling motion of the rider. Since the initial placement of the lock was in the crux between the

toptube and downtube of the bike, the location coupled with the size of the lock could result in undesired contact with the biker's legs. The team also concluded that since the top tube of the Fuji bike being used for research is curved, it would be difficult to create a proper mount for the lock. Due to all of these factors, the team realized that it would need to relocate the lock on the bike to allow for optimal performance. The relocation of the lock meant that the mechanism attaching the lock to the bike had to change. Since the shape of the rear stays differ from bike to bike, the team first made a 2D prototype of the curvature of the rear stays of the particular Fuji bike selected. The choice to 3D print a 2D model enabled faster prototyping and minimized excess material use.

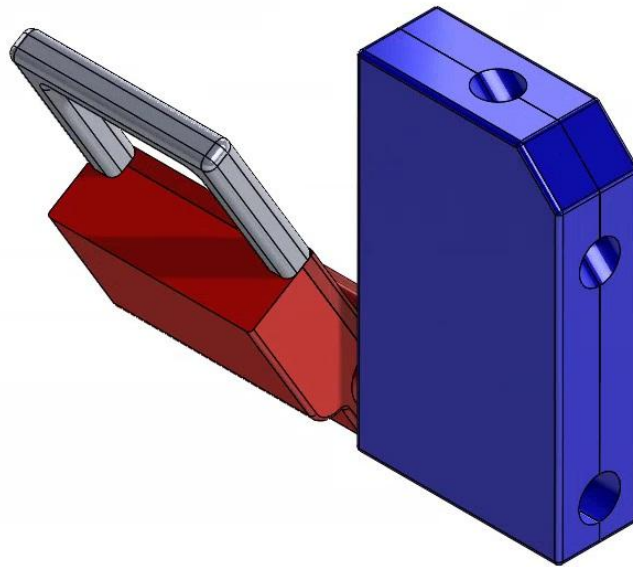


Figure 25. The third version of the modified U-lock.

The third iteration of the lock, as shown in Figure 25, included the relocation to the back of the bike, opposite of the drivetrain. This location was selected to avoid

interfering with the rider, have the least effect on the balance of the bike, and ensure proper locking technique. The electronics housing box would be secured to the rear, non-drive seat stay and had a concealed shaft to the locking component so that electronic components were exposed. Ultimately, the lock had potential as a stationless bikeshare smartlock, but was not selected due to many intricate features that would be hard to manufacture and easily exploited by bike thieves.

Evolution of loop lock. During early stages of research, Team BIKES pursued lock designs that had limited use for individual users, but could be adapted as part of a bikeshare system. The team targeted designs that could not be removed from the bike and locking mechanisms that could not be directly interacted with. During this research stage, the team settled on a design similar to a handcuff. The circular tubes reduce stress concentrations and are notoriously harder to bend than square or other angled shapes.

The first iteration depended heavily on the use of two hinges, as seen in the left print in Figure 26. These hinges were oversized to compensate for their inherent weakness. Through FEA analysis and functional testing by means of additive manufacturing prototyping, this design was deemed non-functional. The hinges were of great concern as they were weak to torsion and required complex machining. The hinged loop also limited the type of bike racks the lock would fit around. In addition, the hinge pins posed a safety concern in that each pin needed to be permanent or a potential thief could simply remove the pins, rendering the lock ineffective. There was little confidence

in the safety that this design could provide, therefore the concept was reworked for the next generation.



Figure 26. First, second, and third iterations of the Loop Lock concept.

The second iteration, the middle print in Figure 26, introduced a rotating collar design. The lock was designed in two halves, each with half of the locking loop. This concept was maintained in future iteration, as seen in the right print in Figure 26. Each half had a large metal cylindrical collar that would fit over a bike seat post and a half loop welded onto the side. The second iteration used thin collar walls with thin vertical pins as the locking mechanism. While the design considerably improved from the first to second iteration, the team recognized the need for stronger pins and collars to meet the stringent security goals set by comparable bike lock manufacturers.

The third iteration of the loop lock concept, the right print in Figure 26, involved a much sturdier design than iteration two. The upper collar has a welded shaft that acts as a center post for the lower collar to rotate around. Two caps are welded on top and bottom to keep both collars together and to allow a release mechanism to be safely contained within the lock housing. This design is the direct precursor to iteration four.

The fourth iteration, depicted in Figure 27 below, is improved from the third iteration with practical design changes and a housing unit for on-board electronics. Each collar thickness is increased to accommodate four steel pins and 4 matched teeth with channels to prevent over-rotation of collars. The teeth on the lower collar (pictured in yellow) mate to channels in the upper collar (pictured in red) to limit the rotation and prevent users from locking the collars in the “open” position. The open position is when the tips of the collars are separated allowing the lock to go around a bike rack. The closed position is when the arm tips are nearly touching and the locking pin has secured the lock shut. Four steel pins are loaded into their respective shafts near the teeth. One end of each pin housing contains a spring to force the pins into the locked position when the system does not actively call it open. The passively locked pin-system conserves system power so that no actuation or sensor is necessary to lock the collars. In the event of system power loss, the system will stay in its locked state and prevents would-be thieves from simply opening the collars. Moreover, the flex in each loop is minimal enough to prevent the lock from being pulled far enough apart to be slid off of a bike rack. The lower collar additionally has drain holes through the entire collar to prevent

moisture buildup, which would interfere with the pin operation. This welded ring on the underside secures the lower collar to the upper one and prevents disassembly. The fourth iteration does not allow access to the internal pins and springs once assembled.

The fourth iteration includes an attached electronics housing. The electronics housing contains two bolts that are inserted from the inside of the housing into the upper collar. There are two holes on the same face of the housing that align with the lower collar that contain spring loaded contacts in order to tell if the lock is open. An inlaid conductive strip in the lower collar will complete a circuit to inform the software if the lock is in the closed position. The top of this housing has access to the top of the upper collars in order to attach actuators that recess the locking pins and allow the collars to rotate and open.

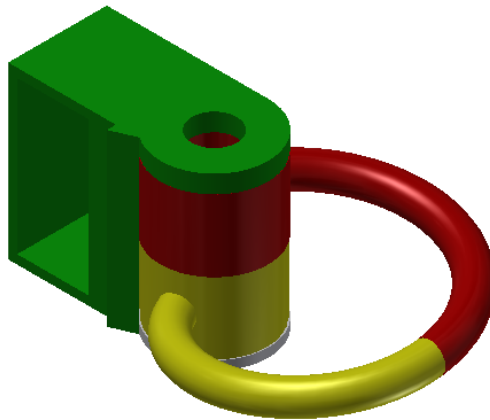


Figure 27. Iteration 4 assembly. Pictured in colors for clarity and reference.

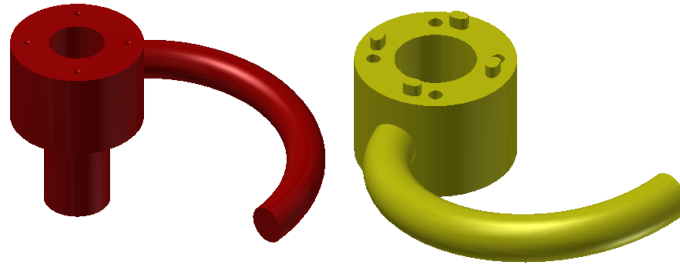


Figure 28. Upper collar (left) and lower collar (right). Lower collar rotates around shaft on upper collar.

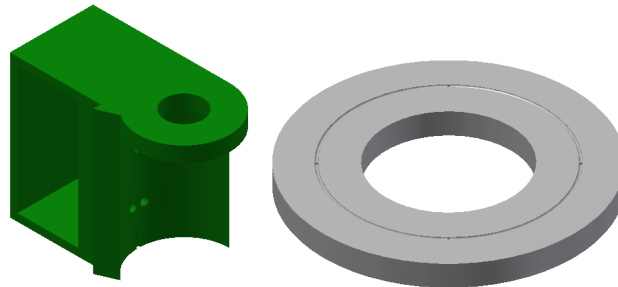


Figure 29. Electronics housing (left). Bolted to upper collar from inside housing. Lower welded ring (right). Drain holes prevent liquid and particulate buildup inside lacking mechanism.

The fourth iteration of the loop lock prototype was also 3D printed, allowing the team to integrate electronics into the device and test the system as a whole. The team corrected bugs in the control program and modified the lock design to improve quality and function. For example, concentric circle tolerances were enlarged to permit smoother rotation and quicker assembly during production. Several edges were chamfered for ease of assembly and user comfort. In future prototypes, the team intends to manufacture aluminum locks for more accurate testing; this model would also be implemented in a scale study to test the user experience of the lock. The team would start with aluminum

due to the cost of material and material softness. Aluminum is softer than steel, therefore simpler to machine. After the aluminum lock is tested and modified, the team would then manufacture prototypes in steel minimizing the needed revisions, thus saving manufacturing time and cost. The steel locks would require professional assistance welds so that the welded bonds are not inherently weak due to poor workmanship. These steel prototypes would be failure tested in order to confirm the FEA failure results and ensure the effectiveness of the lock.

Immediately after each stage of design, the team performed FEA analysis. By performing this analysis, the team could make modifications to maintain strength and predict failure points. Since prototypes are expensive and time consuming, it is ideal to create digital models prior to manufacturing to eliminate as many faults as possible. During failure testing, it is also important to predict where failure will occur in order to focus the study on that area. Studying the failure point can provide insight into the type of break, what types of forces would cause the break, and methods for design improvements to prevent subsequent failures.

To increase manufacturability, the design was adjusted a final time resulting in iteration five. The major differences between this iteration compared to previous versions is the shape of the locking arms, the method of pinning the lock closed, and the method of attaching the electronics box to the rest of the lock. The locking arms are no longer semicircles due to manufacturability issues and have been changed to straight rod arms with a 90-degree bend. This bend has minimal effects on the strength of the arm,

but increases the manufacturability of the lock. Further, the fifth iteration no longer has nubs and pins between the layers of the collars. Instead, the method of locking is a single large pin that extends from within the electronics box, through the support beam that holds the lock together, and into the upper collar. The lower collar is held in place because it is sandwiched between the pinned upper collar and the immobile electronics box and therefore cannot rotate in any direction.

The electronics box will be constructed out of sheet metal and folded into the desired configuration. The folded material will effectively protect the housed electronics from weather and tampering. A structural member of the lock and the locking pin itself pass through the box to maximize difficulty of removing the electronics from the locking mechanism. To prevent the lock from rotating while users are riding the bike, there will be three shafts with ball and spring assemblies. The concept is very similar to the dials in a car console in that the dial stays in a particular position until enough force is applied to move to the next position. The lock will have a similar system so that the lock only moves when the user needs it to move. These shafts will occur in the top and bottom collars to prevent either collar from rotating. Both collars will be constructed identically in every respect so that instead of manufacturing two separate collars and requiring extra infrastructure, only one design will be required. There will be no discernible top or bottom collar during lock assembly.

The sixth and final iteration, pictured in Figure 30, refines some of the finer details from the previous iteration. The locking method with a single pin has stayed the

same. The locking arms are still being bent at 90-degree bend and welded into the collar, and the collars will both be identical to improve manufacturing efficiency. The electronics box is still folded sheet metal that will be welded together at the edges to protect the electronics. The pin housing is welded to the top of the electronics box depicted in Figure 31, which contains the pin, a spring system, and a servo to actuate the pin. The pin is a simple cylinder with a slot through the side as shown in Figure 32; this slot is where the servo arm moves through to open the lock. To conserve power, the servo is only powered when actuating, so a spring is mounted to push the pin to the locked position. To hold the lock in the open position during riding, each collar has three sets of neodymium magnets that hold the lock in the riding position with the locking arms straight back to prevent the rider from hitting the arms.



Figure 30. Final lock prototype.

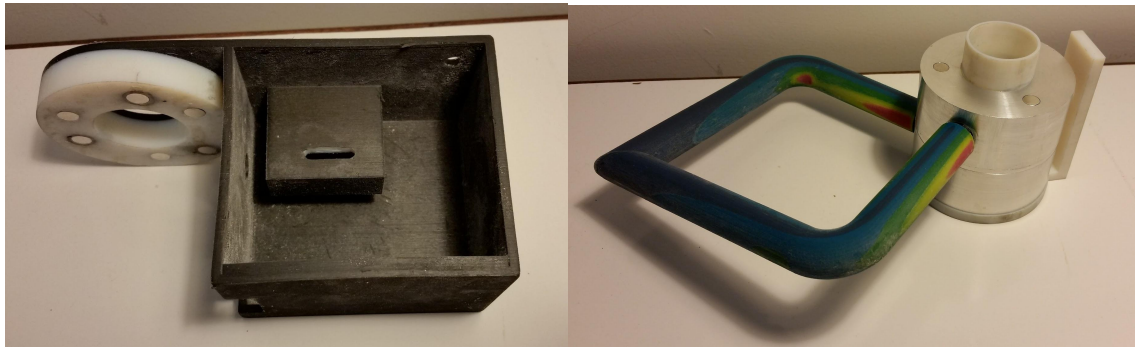


Figure 31. Electronics box (left) with pin housing. Neodymium magnet placement (right).

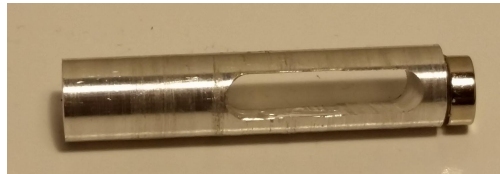


Figure 32. Pin with slot for servo.

NFC reader selection. The Elechouse NFC Module V3 was selected as the NFC reader for the smartlock prototype. The selected model provided similar functionalities to the model tested, but stood out due to the detached antennae that would best fit inside tight smartlock electronics housing. The NFC shields with attached antennas impose the restriction that the NFC shield itself, pinned to the microcontroller, must be in close proximity to the user access control interface. Due to the Elechouse NFC reader's antenna configuration, the user access interface could be physically isolated from the other electronics. Figure 33 illustrates a possible system architecture using the Elechouse NFC Module V3 and the MIFARE smart card.

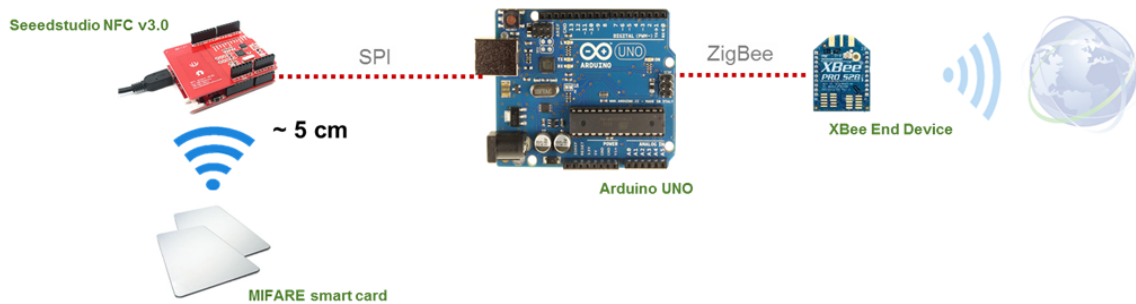


Figure 33. Architecture of the NFC-based access control system.

Hardware-Software co-development. As new hardware modules were added to the system, the code base used to respond to these modules was updated. When system development began, the functionality programmed onto the Arduino was limited to reading a MIFARE smartcard and reporting the card's ID via a serial monitor on connected computer. The next hardware addition was a small push button used to simulate a user activating the lock's electronics by pressing a sensor. The pushbutton was connected to one of the Arduino Uno's interrupt capable pins, and in software a rising-edge interrupt was enabled for this pin. As a result, when the push button is depressed, a connection across the legs of the push button is formed and this results in a logically HIGH voltage level being observed by the interrupt enabled pin and this triggers a specific software function. After adding the interrupt functionality, it became necessary to develop a more formal structure for the code running on the Arduino so that user interaction and required error checking mechanisms could be simulated.

The code was organized into several states that represented potential system conditions. The states and the functionality each performs are specified below in the order in which they would be reached under normal operation. Note, each state contains a specific mechanism for moving to either the subsequent state or the general error checking state.

Not Reading- This is the idle state from which the process of verifying a user begins. The next state is accessed when a user presses the activation button.

Reading- During this state, the reader attempts to read from an NFC tag. The maximum length of time that the Arduino can operate in this state is limited for the sake of power savings. The NFC reader's power draw is significantly higher when it is actively attempting to read a tag.

Read- Upon successfully capturing the ID from an NFC tag during the "Reading" state, this state seeks to verify if it is a valid ID. It broadcasts the tag ID, and waits to receive a signal confirming or denying the tag. If the tag ID is approved, the next state is "Open", otherwise it is "Not Reading."

Open- This is the state that triggers the motor to open the locking mechanism in the final system, so to model this motion, a small servo is rotated 180 degrees once this state is reached. The transition to the "Closed" state occurs after a button press is registered by the same hardware interrupt used to activate the "Reading" stage to simulate the final closing of the lock when a user has re-locked the bike after riding it to a new location.

Closed- In this state a message is broadcast that contains the ending location of the bicycle.

Error Checking- This state is entered if the Arduino cannot connect to one of the attached components, including the NFC reader, GPS or Xbee module. This state forces the Arduino to attempt to reconnect to the missing component until the connection is once again secured.

Once these states were defined, the general structure of the code was left intact, and the only changes made concerned the exact sources of the data that each state is responsible for. The first example of this is the Xbee communication. During initial prototyping stages, the “Read” and “Closed” states simply reported the NFC tag and ‘location’ respectively via a serial monitor and waited for responses from the serial input provided by team members. Once the Zigbee communication and web server back end were developed, these states sent and received information via an Xbee end device connected to the Arduino and the web server responded in real time to the requests sent by these states. Likewise, upon the integration of the Adafruit Ultimate GPS Breakout module, the location information sent to the server was based on actual GPS data. The development of a clear, expected progression of communication between the system and other entities was necessary because in the final design, the entire system was powered using a 600 mAh rechargeable battery in lieu of a wired connection to computer, as used during development. This removes the serial monitor as a debugging tool for the system,

so it is imperative that code progress reliably between the states with appropriate tangible effects, such as the turning of a servo or a record of data received by the servo.

XBee network progress. The techniques for interfacing with the XBee modules went through several iterations. The first iteration involved byte-by-byte communication through multiple XBees configured in the AT mode. The XBee modules' pins were made accessible for prototyping by connecting them to breakout boards and then an Arduino. Initially, the integration between the Arduino and XBee modules was done in the simplest form: using two AT modules with one as the coordinator and another as an end device. Once the XBee modules were configured as such, the non-coordinator XBee was wired to the Arduino, matching the TX pin of one with the RX pin of the other and vice versa. The other module was connected via USB to a computer running the XCTU application with the terminal tab of the application open to allow for serial communications. Within the application, a packet was created as a simple test string: "Hello, world!" Sending the string resulted in the "Hello, world!" message being received by the Arduino's serial monitor.

The second iteration had both modules connected to Arduinos. When messages were sent from one Arduino's serial monitor to the others, the data was transmitted byte-by-byte. This messaging format made receiving the transmissions more difficult to handle due to the possibility of messages from separate XBees becoming interpolated. To solve the byte-by-byte communication shortcomings, the team explored several of the XBee features regarding pin-specific communications. An XBee module was configured

to send data over a specific pin on the board, and then an Arduino digital output pin was wired into the XBee pin configured for data transmission. On the coordinator side, the reading was then received in the XCTU application, proving that this technique would work for one-way, binary data.

A shortcoming of the digital read was that the data size would be limited. In order to include more information from the bike-side of the network, the team explored the analog read feature. The XBee modules were configured to be able to send analog data over a specific pin, and the data was received on the coordinator side in the XCTU application. The transition from digital to analog allowed for more data to be read in a single transmission. However, since there were fewer analog pins than digital pins the size of the data that could be transferred was limited. Furthermore, the analog pins were only capable of one-way communications, so there was still a need to develop a way for the coordinator to send data to the end devices.

To solve the one-way communications issue that existed in the past two experiments, the digital command XBee feature was explored. Using this feature, the coordinator could send a digital HIGH or LOW to a specific pin on the end device. This digital command would suffice for simple yes/no responses, but more pins would need to be used in order to send more than just a 1 or 0. Also, for each command sent down to the end device, an additional command would need to be sent to return the pin to the original state. As a result of the need for multiple messages and the lack of an ability to

send more than a 1 or 0, another technique for data transmission from the coordinator to the end devices was needed.

The solution to the above shortcomings came in the form of mix and matching firmware of the XBees and sending data in a string format. The routers and end devices were configured in AT mode, and the coordinator was configured in API mode. Recall, in AT mode, the Arduinos were receiving the messages byte-by-byte. When the coordinator is set up in API mode, the coordinator end of the system handles this issue by receiving the string within a single packet. This technique allowed the routers and end devices to simply send a string through an AT mode XBee and have the transmission received as a single packet on the coordinator side, which eliminates the interpolation issue described earlier.

To communicate in the other direction, the coordinator wraps a single byte into a transmit request ZigBee packet and sends it back down to the end device. The AT mode end device then receives the message, which was formatted as a single byte intentionally and can then be interpreted however the protocol on the Arduino decides. Note that if the transmission was any more than a single byte, the byte-by-byte issue would arise, thus requiring a smarter receive protocol on the Arduino. With the single byte, the options increase from the previous two options of 1 or 0 to 255 options.

After iterating through multiple techniques for methods of communications with the XBee modules, network testing was performed. To test simple network configurations with various node types, the XBees were configured to transmit empty

packets from various numbers of routers and end devices back to a single coordinator, pictured in Figure 34. The packets were seen with a serial monitor on the coordinator. By analyzing the MAC addresses of each packet, it was confirmed that all of the network configurations below were functional, depicted in Figure 35.

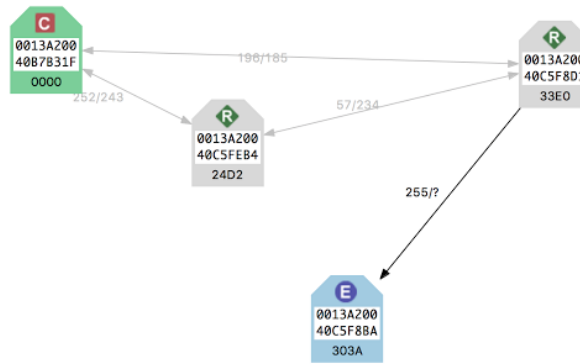


Figure 34. Example network test configuration.

```

7E, 0, 12, 92, 0, 7D, 33, A2, 0, 40, B7, B3, 1F, 39, 60, 1, 1, 0, 0, 1, 2, A, 47,
7E, 0, 12, 92, 0, 7D, 33, A2, 0, 40, B9, B, 95, 55, D8, 1, 1, 0, 0, 1, 2, B, DF,
7E, 0, 12, 92, 0, 7D, 33, A2, 0, 40, B7, B3, 1F, 39, 60, 1, 1, 0, 0, 1, 2, A, 47,
7E, 0, 12, 92, 0, 7D, 33, A2, 0, 40, B9, B, 95, 55, D8, 1, 1, 0, 0, 1, 2, 8, E2,
7E, 0, 12, 92, 0, 7D, 33, A2, 0, 40, B7, B3, 1F, 39, 60, 1, 1, 0, 0, 1, 2, 9, 48,
7E, 0, 12, 92, 0, 7D, 33, A2, 0, 40, B9, B, 95, 55, D8, 1, 1, 0, 0, 1, 2, A, E0,
7E, 0, 12, 92, 0, 7D, 33, A2, 0, 40, B7, B3, 1F, 39, 60, 1, 1, 0, 0, 1, 2, A, 47,
7E, 0, 12, 92, 0, 7D, 33, A2, 0, 40, B9, B, 95, 55, D8, 1, 1, 0, 0, 1, 2, B, DF,
7E, 0, 12, 92, 0, 7D, 33, A2, 0, 40, B7, B3, 1F, 39, 60, 1, 1, 0, 0, 1, 2, A, 47,
7E, 0, 12, 92, 0, 7D, 33, A2, 0, 40, B9, B, 95, 55, D8, 1, 1, 0, 0, 1, 2, 8, E2,
7E, 0, 12, 92, 0, 7D, 33, A2, 0, 40, B7, B3, 1F, 39, 60, 1, 1, 0, 0, 1, 2, 9, 48,
7E, 0, 12, 92, 0, 7D, 33, A2, 0, 40, B9, B, 95, 55, D8, 1, 1, 0, 0, 1, 2, B, DF,
7E, 0, 12, 92, 0, 7D, 33, A2, 0, 40, B7, B3, 1F, 39, 60, 1, 1, 0, 0, 1, 2, B, 46,
7E, 0, 12, 92, 0, 7D, 33, A2, 0, 40, B9, B, 95, 55, D8, 1, 1, 0, 0, 1, 2, 9, E1,
7E, 0, 12, 92, 0, 7D, 33, A2, 0, 40, B7, B3, 1F, 39, 60, 1, 1, 0, 0, 1, 2, A, 47,
7E, 0, 12, 92, 0, 7D, 33, A2, 0, 40, C5, F8, BA, B8, 63, 1, 1, 0, 10, 0, 0, 10, C4,
7E, 0, 12, 92, 0, 7D, 33, A2, 0, 40, B9, B, 95, 55, D8, 1, 1, 0, 0, 1, 2, 9, E1,
7E, 0, 12, 92, 0, 7D, 33, A2, 0, 40, B7, B3, 1F, 39, 60, 1, 1, 0, 0, 1, 2, 9, 48,
7E, 0, 12, 92, 0, 7D, 33, A2, 0, 40, C5, F8, BA, B8, 63, 1, 1, 0, 10, 0, 0, 10, C4,
7E, 0, 12, 92, 0, 7D, 33, A2, 0, 40, B9, B, 95, 55, D8, 1, 1, 0, 0, 1, 2, B, DF,
7E, 0, 12, 92, 0, 7D, 33, A2, 0, 40, B7, B3, 1F, 39, 60, 1, 1, 0, 0, 1, 2, B, 46,
7E, 0, 12, 92, 0, 7D, 33, A2, 0, 40, C5, F8, BA, B8, 63, 1, 1, 0, 10, 0, 0, 10, C4,
7E, 0, 12, 92, 0, 7D, 33, A2, 0, 40, B9, B, 95, 55, D8, 1, 1, 0, 0, 1, 2, 8, E2,
7E, 0, 12, 92, 0, 7D, 33, A2, 0, 40, B7, B3, 1F, 39, 60, 1, 1, 0, 0, 1, 2, A, 47,
7E, 0, 12, 92, 0, 7D, 33, A2, 0, 40, C5, F8, BA, B8, 63, 1, 1, 0, 10, 0, 0, 10, C4,
7E, 0, 12, 92, 0, 7D, 33, A2, 0, 40, B9, B, 95, 55, D8, 1, 1, 0, 0, 1, 2, B, DF,
7E, 0, 12, 92, 0, 7D, 33, A2, 0, 40, B7, B3, 1F, 39, 60, 1, 1, 0, 0, 1, 2, A, 47,
7E, 0, 12, 92, 0, 7D, 33, A2, 0, 40, C5, F8, BA, B8, 63, 1, 1, 0, 10, 0, 0, 10, C4,
7E, 0, 12, 92, 0, 7D, 33, A2, 0, 40, B9, B, 95, 55, D8, 1, 1, 0, 0, 1, 2, 9, E1,
7E, 0, 12, 92, 0, 7D, 33, A2, 0, 40, B7, B3, 1F, 39, 60, 1, 1, 0, 0, 1, 2, B, 46,
7E, 0, 12, 92, 0, 7D, 33, A2, 0, 40, C5, F8, BA, B8, 63, 1, 1, 0, 10, 0, 0, 10, C4,
7E, 0, 12, 92, 0, 7D, 33, A2, 0, 40, B9, B, 95, 55, D8, 1, 1, 0, 0, 1, 2, 9, E1,
7E, 0, 12, 92, 0, 7D, 33, A2, 0, 40, B7, B3, 1F, 39, 60, 1, 1, 0, 0, 1, 2, A, 47,

```

Figure 35. Example XBee Packet Analysis of Various MAC Addresses.

XBee coverage tests. Once a simple network was assembled to send empty packets to the coordinator in close proximity, the range and coverage of the XBee modules were explored. The XBee range capability experiments were performed in three main steps: range, coverage, and network tests. The range tests aimed to find the maximum range of the XBee modules in a clear line-of-sight (LOS) path. The coverage tests were performed to find the quality of the signal in all directions including non-line-of-sight (NLOS) environments. Finally, the network tests were performed to simulate the proposed bikeshare where nodes would be placed accordingly in order to maximize reliable communications with a defined set of bicycle racks on campus.

Range tests. The range tests were conducted using the XCTU's built-in range testing software, depicted in Figure 36. Within the program, two XBee modules can be selected to monitor their received signal strength indicator (RSSI) and the packet success rate over time.

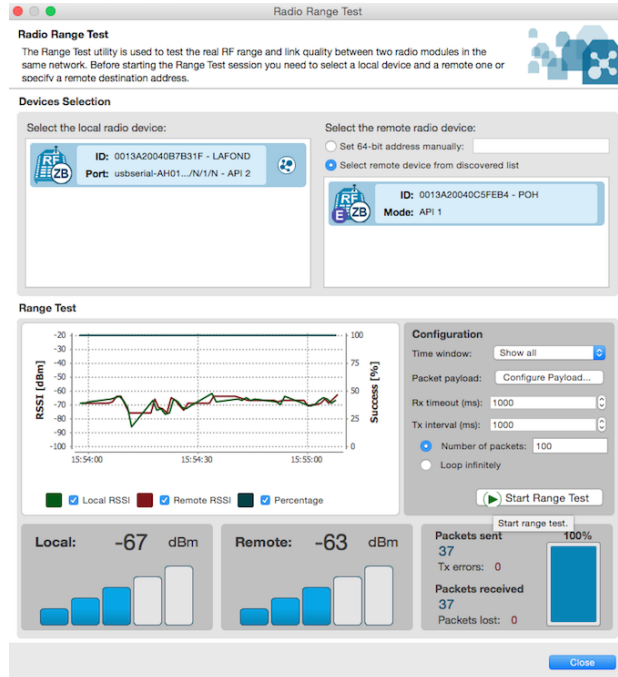


Figure 36. XCTU’s Range Test Program.

The range test was first conducted at McKeldin Mall where a maximum LOS distance of approximately 1,100 ft was measured via Google Maps, depicted in Figure 37. The range test was conducted by leaving the coordinator node stationary in front of McKeldin Library while having the end device node move farther away from crosswalk to crosswalk. The range test was conducted at each intersection of the crosswalk. At the initial crosswalk, the signal strength was recorded at an average value of -50 dB, and as the end device moved away the signal strength expectedly decreased. The documentation for the XBee modules used in the experiment claim to have an acceptable RSSI range of -40 dB down to -102 dB. By incrementing across McKeldin Mall, the modules proved to maintain a strong RSSI reading throughout the entire test. Upon reaching the end of the

Mall, at approximately 1,100 ft, the modules were still maintaining an average RSSI of approximately -70 dB, well within the documented acceptable region.



Figure 37. McKeldin Mall Range Test Site.

Since the initial range test occurred without achieving the goal of finding the maximum distance of connectivity, a new site was selected for an additional range test. The next site was selected since it has the longest LOS path on campus. This makes it the optimal test for the system since the final system will not need to reach LOS distances that cannot be reached on campus. The test was conducted across the Chapel and Fraternity Fields, depicted in Figure 38.



Figure 38. Chapel and Fraternity Fields Test Site.

The new test site was able to leverage the elevation difference between the entrance to Memorial Chapel and the far east edge of the Fraternity Fields to obtain a measured distance of approximately 1,650 ft. The test was conducted in a similar fashion to the initial test with the end device moving in increments away from the coordinator. The maximum distance of 1,650 ft returned a packet success rate of 100% while maintaining an average RSSI value of -75 dB.

The differences in distance and RSSI between the first two tests came out to 540 ft and -5 dB, respectively. Noting the XBee documentation claims of receiver sensitivity down to -102 dB, the XBee units should hold for all possible LOS communications links achievable on campus for the sake of the project. Should the units be located at higher elevations, such as rooftops, it would be possible to extend a clear LOS connection that may fall outside of the tested range.

Coverage tests. The coverage tests were performed in two ways. Initially, the same XCTU range test program was used where a stationary coordinator would run tests against a moving end device. The largest difference between the range and coverage tests had to do with the NLOS conditions. In the coverage tests, a person carrying the mobile node would walk in all directions until the signal was lost, instead of only walking in one direction maintaining LOS, and then the person documented the location of the lost signal on a printed out map. The test was conducted at McKeldin Mall with the coordinator located at the same location as in the initial range test. During the test, the XBee modules appeared to drop the signal at approximately -90 dB, slightly less than the

expected -102 dB claimed in the documentation. Once the test was completed, the boundary points were entered into Google Earth to plot a rough sketch of the coverage region of a single node, depicted in Figure 39. Note that the red dot is the location of the stationary node, and the blue region is the coverage area. It was later discovered that the test experienced issues during the data collection in the most northeast location of McKeldin Mall. The actual coverage information within that region extends slightly farther north and east.

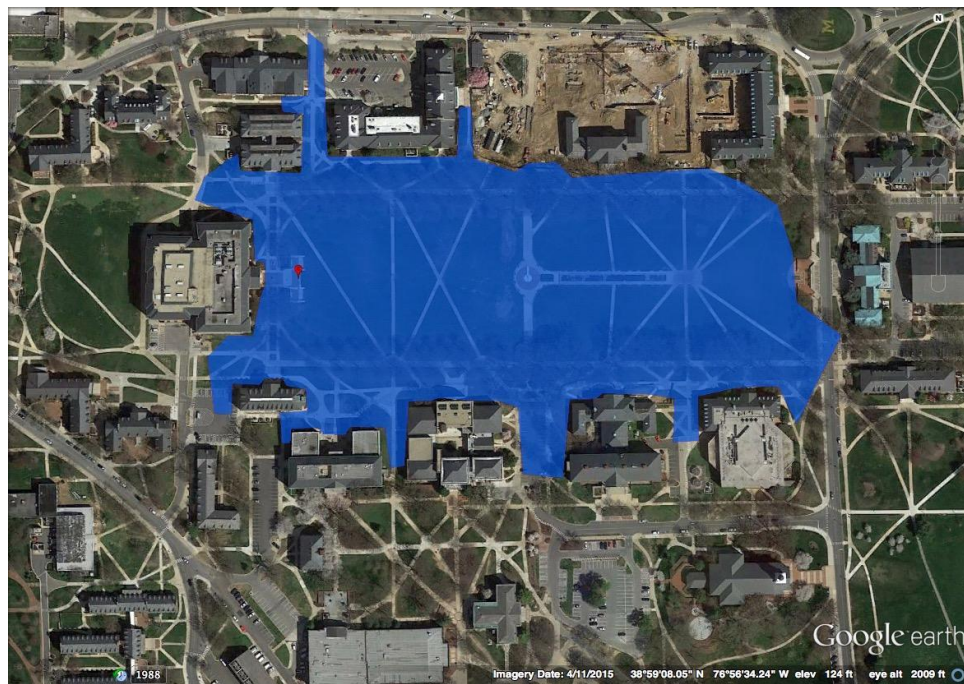


Figure 39. Coverage Region of Single Node.

A flaw with the approach for the coverage test was that the data displayed in Figure 39 above depicts a binary representation of the coverage region, meaning either the location has connectivity to the coordinator or it does not. Given the great deal of

fluctuation in RF systems, any number of conditions such as: population, weather, or device orientation, could affect the exact coverage region. To provide more meaningful data, the signal strength should be measured at each point to provide insight into which regions are more susceptible to potential losses in the connection. In order to more accurately display the coverage region, a heat map approach was proposed. The heat map test would include a device that could log the GPS location and RSSI while the user carries the device around. The device could then save the information and pass it on to a mapping library such as Leaflet or Google Maps.

Building the system necessary for the heat map test presented new problems with logging the RSSI value. Within the ZigBee protocol for the XBee Series 2 modules used, the RSSI is not sent with each packet. Instead, the receiving module measures the RSSI of the last received packet and stores it locally. The value can then be requested locally using the ATDB command. To implement a device that can update the RSSI values at each point, a new packet from the coordinator had to be sent repeatedly at a desired interval. Without the coordinator constantly sending packets, the RSSI value stored on the local XBee module would remain unchanged throughout the polling period. After learning how the RSSI value can be recovered, a device was built using a Raspberry Pi, an Adafruit Ultimate GPS Breakout Board, an Arduino Uno, an XBee end device, and a portable battery as seen in Figure 40.

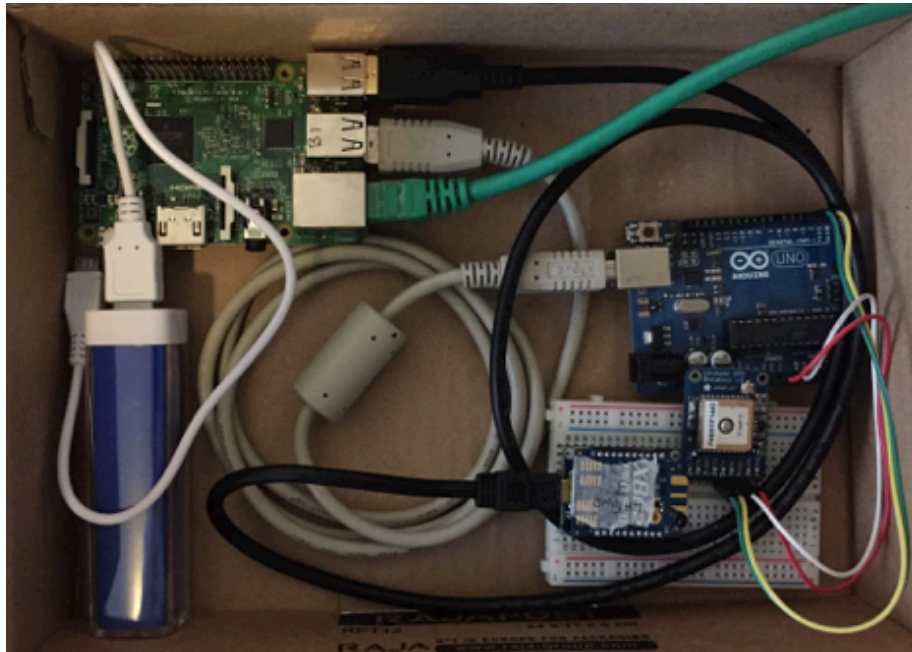


Figure 40. Heat Map Data Collector.

The heat map data collector device used the portable battery to power the Raspberry Pi, which in turn powered the Arduino and the XBee through USB connections. The GPS board was connected to the Arduino where the TinyGPS+ library was used to parse out latitude, longitude, altitude, and time every second from the NMEA data output. The Arduino then printed out the information in CSV format to the serial port. On the Raspberry Pi, a python script read in the serial data from the Arduino to read back the GPS data at any given time. The python script also utilized the XBee-Python library to interact with the XBee in order to assemble the AT command to request the RSSI. With the walking pace set by the user holding the heat map data collector, it was found that polling the RSSI every five seconds sufficiently collected data for use in the

generation of the heat map. An important note to make is that in order to sync the RSSI data polled every five seconds and the GPS data received every second, the serial port connection with the Arduino had to be flushed to receive the latest GPS coordinates immediately following an RSSI request.

After developing the heat map data collector described above, the team found that the Raspberry Pi did not provide enough feedback while the test was being conducted. The system was then simplified to include solely the Arduino, the GPS, and the XBee, where the Arduino and XBee module were connected to a laptop, rather than the Raspberry Pi, as depicted in Figure 41.

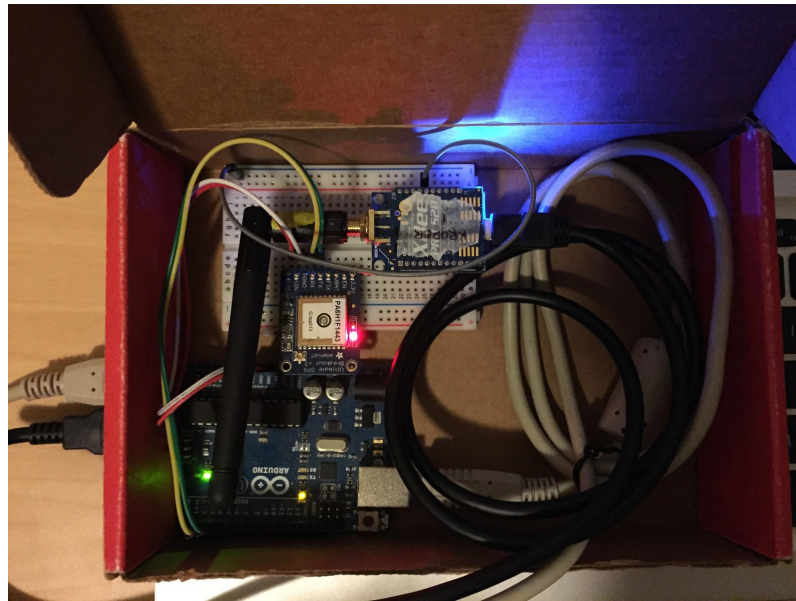


Figure 41. Heat Map Data Collector Version Two.

With the new system assembled and ready to be used, the first heat map test was conducted at McKeldin Mall, with the coordinator located at the same position as in

previous tests. The heat map data collector with the accompanying laptop was then carried around the mall in a vertical zigzag pattern. While walking, the device collected the data needed to generate a heat map and wrote it to a file in the CSV format seen in Figure 42.

```
<RSSI>, <LATITUDE>, <LONGITUDE>, <ALTITUDE>, <TIMESTAMP>
```

Figure 42. CSV format.

At the conclusion of the test, the data was then read into a Leaflet mapping program to turn the GPS coordinates into scaled intensity values based on the RSSI. Since the RSSI is recorded as negative values ranging from approximately -30 dB (strong) to -90 dB (weak), the RSSI was mapped to the intensity factor inversely. The result of the Leaflet map can be seen below in Figure 43. Note that the XBee icon represents the location of the coordinator for the test.

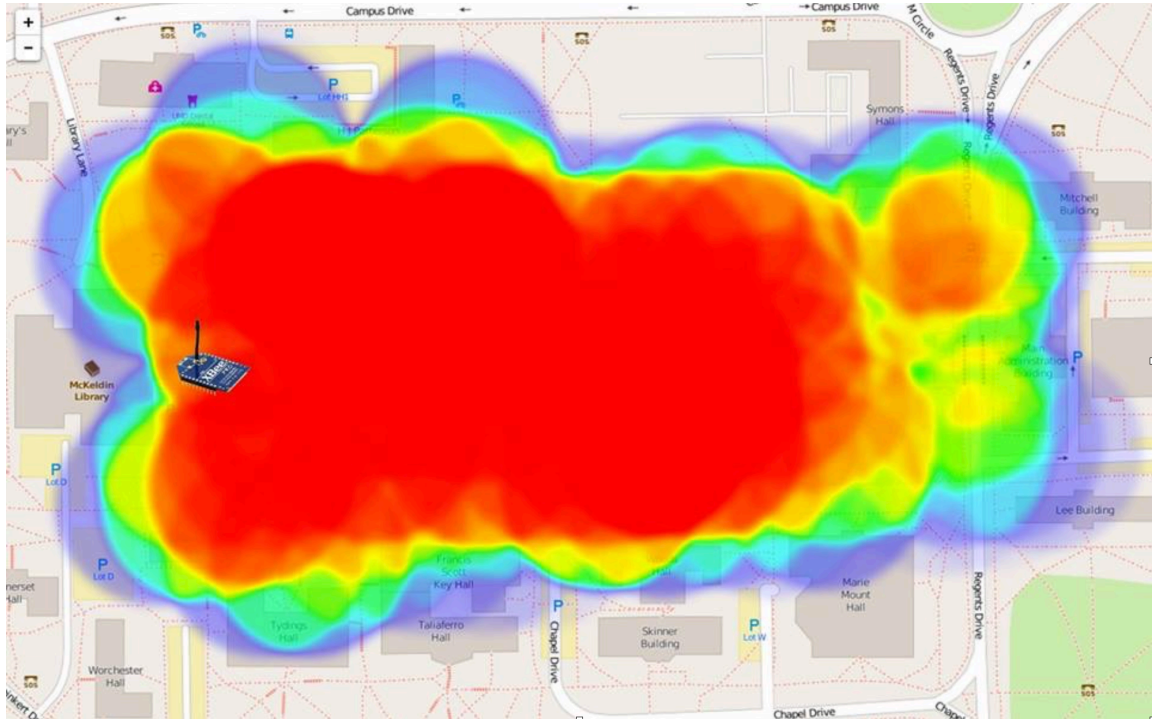


Figure 43. McKeldin Heat Map Test Leaflet Program.

The same heat map with the walking path superimposed can be seen below in Figure 44. With the walking path inserted, it can be seen that the data in the center of the triangle formed at the far right of the heat map alters from reality. Leaflet generated the heat map by taking each individual point and applying a circle with the intensity value determined from the RSSI and then slowly reducing the intensity as the distance from the center increased. Because of this technique, the center of the right triangle appeared to be weaker than the north east point of the triangle even though the center was closer to the coordinator. In reality, the center of the triangle had a higher RSSI value than the north east point.

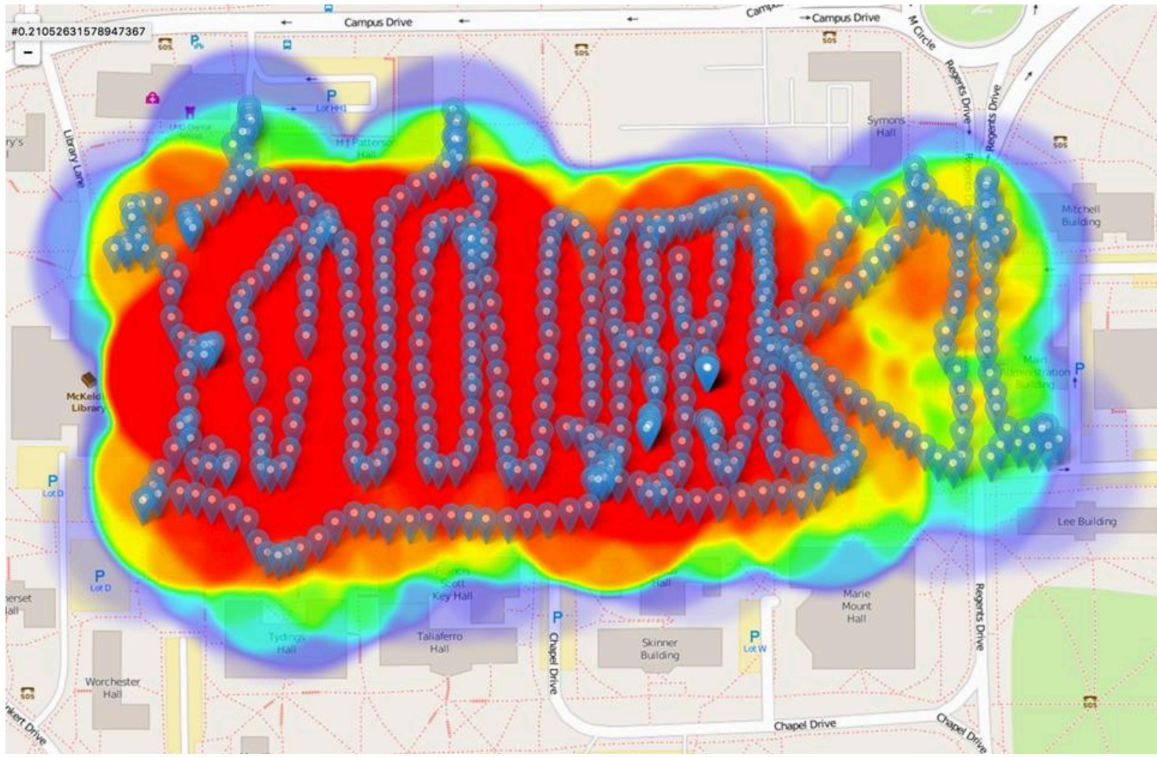


Figure 44. McKeldin Heat Map Test Leaflet with Path.

Other pitfalls with the Leaflet heat map generator involved the same fading circle technique. With the fading circle, weaker points in close proximity to stronger points were masked by the stronger point's computer-generated coverage region. In addition, the boundaries of the coverage area seeped into the adjacent buildings despite the reality being that there was no signal inside the buildings. Viewers of the heat map may also find it difficult to determine where the heat map test took place due to the opacity constraints blocking out the location behind the coverage region. One final issue with the map was that it also failed to distinguish a difference between the strongest signal near

the coordinator at approximately -35 dB from the decent signal located throughout much of the Mall at about -70 dB.

In an attempt to improve the heat map's display, Google Maps' heat mapping tool was explored. Taking the same data but applying a slightly different scaling factor resulted in a much more visually appealing heat map, depicted in Figure 45.

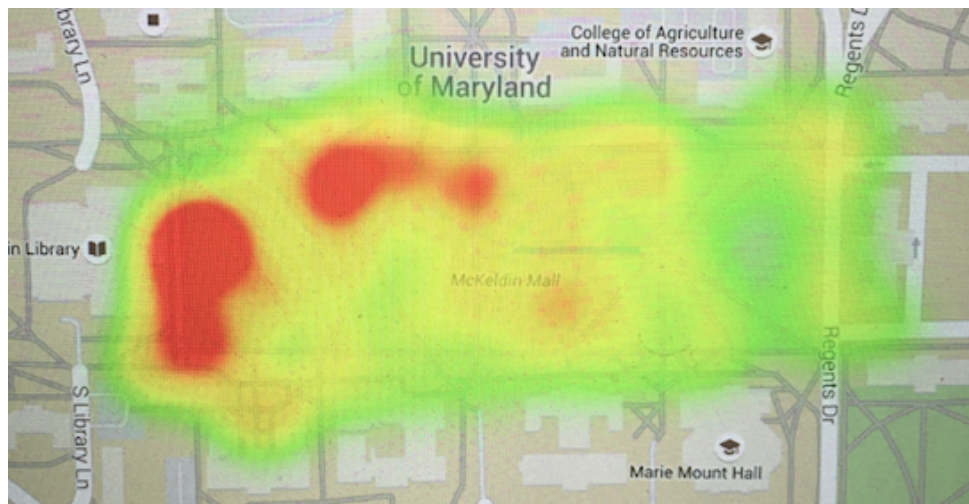


Figure 45. McKeldin Heat Map Google Maps.

The Google Maps heat mapping tool succeeded in distinguishing the difference between signal strength of a decent and a strong signal, and it also succeeded in allowing the viewer to see more of the map detail behind the heat map. Aside from those two improvements, Google Maps still failed to provide the desired heat map image of preserving the reality of the data.

The desired heat mapping tool would allow the user to select a source point where all generated points would tend to decrease from. It would preserve the reality of each

data point and fill in the gaps between the real data points with mesh data to transition from real data point to real data point while tending to decrease away from the source. The heat mapping tool would also need to have an option to insert harsh boundaries where the edges of the buildings could be entered to show that in reality there is no signal inside the classroom buildings.

Network test. The final stage of testing for the XBee units came in the form of network connectivity tests. The goal of the network tests was to take the information learned from the previous range and coverage tests and apply it to a selected area of campus. This would allow the team to intelligently place the stationary nodes of the ZigBee network in order to provide coverage to all of the bicycle racks in the selected region. To obtain the bicycle rack location information, the “UMD Campus Web Map” was used with the bicycle rack location layer turned on. With the bicycle rack location layer on, all of the bike racks on campus appear on a map as red marks as seen in Figure 46.

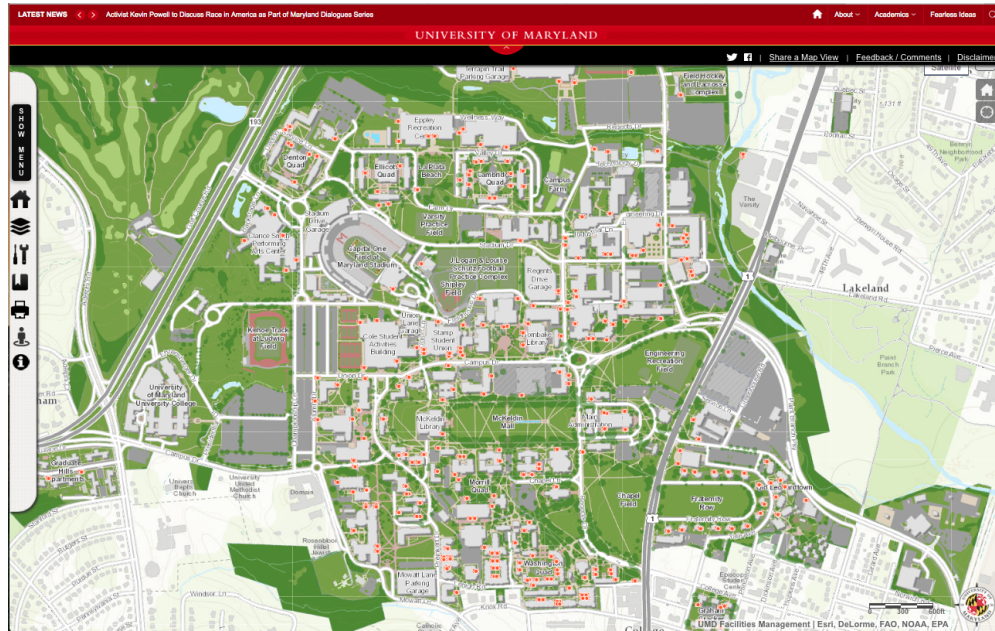


Figure 46. UMD Campus Web Map Displaying Bicycle Rack Locations.

To decide how to run the network tests, the team targeted two specific student types: freshmen and sophomores living in the Ellicott Community studying a humanities field and freshmen and sophomores living in the Cambridge Community studying a science field. With these two subsets of students, the team could target specific areas on campus to implement a scaled down network with the limited hardware available.

The first test conducted was on the first student subset: freshmen and sophomores in the Ellicott Community studying a humanities field. To accommodate this subset of students, all of the bicycle racks in the Ellicott Community will need to be covered by a ZigBee network, as well as all of the bicycle racks near the buildings around McKeldin Mall. Creating a connected network between the two areas posed a problem due to the

distance between the two as well as the lack of any LOS path from one to the other as depicted in Figure 47. The team considered leveraging the altitude of the high-risers in the Ellicott Community, but the inaccessibility of the rooftop of the high-risers prevented a clear test from being conducted. Even if the module could be placed on the roof of Ellicott Hall, it remained doubtful that an elevated module on McKeldin Mall could create a clear LOS path to the Ellicott community to close the link.

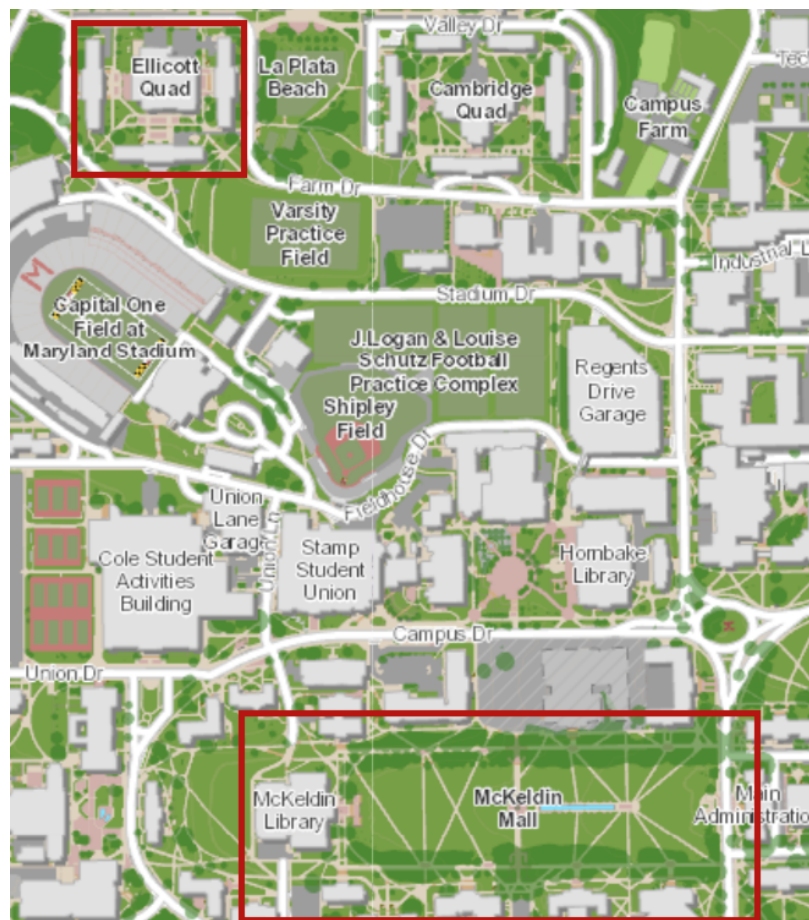


Figure 47. Ellicott Community to McKeldin Mall.

Since the use of a single, connected network between the two areas seemed unlikely without a long stream of routers acting as a backbone, the idea of interoperable, yet separate networks was explored. The proposed idea was that a single coordinator could be set up in the Ellicott Community while an additional single coordinator, operating on the same personal area network identifier (PAN ID), could be set up at McKeldin Mall. Then, in theory, the end device units located on the bicycles could connect to both of the coordinators since they have the same PAN ID. Unfortunately, the multi-coordinator test failed on the initial attempt due to an additional read-only XBee parameter called the 16-bit operating PAN ID. The parameter was specific to each XBee coordinator module such that all of the nodes in the network would be specific to solely that coordinator after the network was established. The routers and end devices on power-up first search for a coordinator on their writable PAN ID. Should the units find a coordinator, they then set their 16-bit operating PAN ID to match that of the coordinator and remember that number so on the next power-up, they can simply attach back to the network without scanning. Since the 16-bit operating PAN ID of both coordinators was different, the end devices that would be located on the bicycles would not be able to establish a connection with the second coordinator.

Fortunately, there exist XBee parameters to fix issues with establishing connections to a coordinator using a separate 16-bit operating PAN ID. In the XBee community, people have had issues after a firmware update where the coordinator obtains a new 16-bit operating PAN ID. Once the parameter is changed because of the firmware

update, all of the previously connected routers and end devices would no longer be able to join unless they too receive a firmware update. Within the XBee module's parameter list, there are three options to go about resolving this issue: forced dissociation, channel verification, and network watchdog timer. Through forced disassociation, the routers or end devices could be sent the AT command, ATDA, to force the module to remove itself from the previously established network and rescan for a new one. Using channel verification, the AT command, ATJV=1, could be sent to the routers or end devices to instruct the modules to verify that there is a coordinator on their operating channel when the device is restarted. If the module finds that there is no existing coordinator on its 16-bit operating PAN ID, it then rescans to attempt to find a new one. The final solution that could be implemented is through use of the network watchdog timer. The AT command, ATNW, instructs the device to wait a set amount of time in integer multiples of three minutes, and if no coordinator was found on its existing 16-bit operating PAN ID, the module then rescans to find a new one.

Since the probability of a bicycle user leaving one community and biking to a distant area exceeds the probability of a short bicycle ride, the minimum time allotment of three minutes for the watchdog timer exceeds the desired delay in the system. The forced dissociation solution provides a faster solution. However, it would require more logic on the smartlock end of the system due to the additional packet type needed, especially considering the fact that the end device on the smartlock is currently configured to operate in AT mode. The channel verification option then proved to be the

best solution for the multi-coordinator problem. With channel verification enabled, the smartlock could reset the XBee end device upon stopping, in turn causing the module to disassociate itself with the previous network and immediately begin scanning for the new coordinator on the same PAN ID. Though these parameters seem to be only used to solve unwanted problems in the community, they can be leveraged to make the network on campus smarter and cheaper. The theory was first tested out by powering on a coordinator and end device pair and sending packets between the two to prove connectivity. The coordinator was then powered off to simulate the end device leaving its coverage area. A new coordinator was then powered on with the same PAN ID but a different 16-bit operating PAN ID to see if the end device would establish a connection. The end device, with channel verification enabled, was then reset to simulate the logic on the smartlock to reset the module upon arriving at its destination. Within seconds, the end device successfully rescanned and established a connection with the new coordinator, proving that the multi-coordinator network on campus was feasible.

Since the Ellicott Community and McKeldin Mall could be connected through separate networks connected through the website rather than a continuous, single mesh network, it was hypothesized that all of the bicycle racks in the two areas could be covered by simply two, intelligently-placed coordinator nodes. To test the hypothesis, an additional heat map test was performed in the Ellicott Community. With the coordinator placed at ground level by the north east corner of Ellicott Hall, the following heat map was created using Leaflet, depicted in Figure 48.



Figure 48. Ellicott Community Heat Map.

Taking note to visit all of the bicycle racks in the community during the test, the single node was proven to have reliable communication to all of the bicycle racks. Again using the UMD Campus Web Map's bicycle rack location layer, the network routing map was drawn, Figure 49.



Figure 49. Ellicott Network Routing Map.

The same network routing map was then applied to the single node covering McKeldin Mall during the earlier heat map tests to provide the following mapping, Figure 50.



Figure 50. McKeldin Network Routing Map.

With the subset of freshmen and sophomores living in the Ellicott Community studying a humanities field, it was proven that with two separate, but connected networks, 21 bicycle racks could be covered with only two XBee modules configured to act as coordinators operating on the same PAN ID.

The next test involved the subset of students who were living in the Cambridge Community studying a science field. The desired region of coverage included the Cambridge Community as well as the buildings along Regents drive. With the close

proximity of these two, it is hypothesized that the targeted coverage region can be covered through the use of a single, multi-node network as depicted in Figure 51.

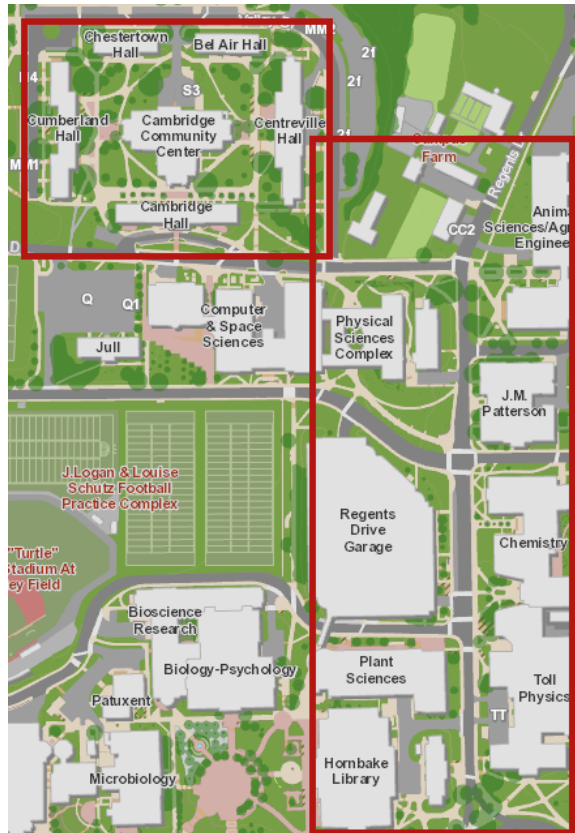


Figure 51. Cambridge Community to Regents Dr. Buildings.

Without being able to leverage the advantages that elevation would provide with LOS, the network was set up with the coordinator just west of the southern tip of Centerville Hall, a router slightly north east of the southern tip of Cumberland Hall (maintaining LOS with the coordinator), and a second router positioned at the intersection of Farm drive and Regents drive. With the nodes positioned as stated, the two routers were powered via laptop USB, and the coordinator was powered by and

interfacing with an additional laptop running the XCTU software. A fourth node was set up as the mobile end device, connected to another laptop running XCTU. The experiment was to walk the end device around to each bicycle rack in the two regions in an attempt to prove connectivity. At each bicycle rack, the end device module was reset, to simulate the actual use in the smartlock design due to the multi-coordinator network requirements. Then a transmission request was sent to the coordinator, to simulate the CSV formatted information to be sent from the bicycle to the website. Upon receiving the packet, the coordinator would then send an acknowledgement packet down to the specific XBee module to simulate the response from the website to the smartlock. Once the transmission was seen to be reliable for both directions, the coordinator would run the routing test to discover the path that it was taking to reach the end device. At each bicycle rack, the path was documented and then the end device would move on to the next bicycle rack.

After conducting the test at the Cambridge Community with the goal of reaching the buildings along Regents Dr., as well, the following network routing maps were drawn to show the connectivity between devices depicted in Figure 52 and Figure 53.



Figure 52. Cambridge to Regents Routing Map One.

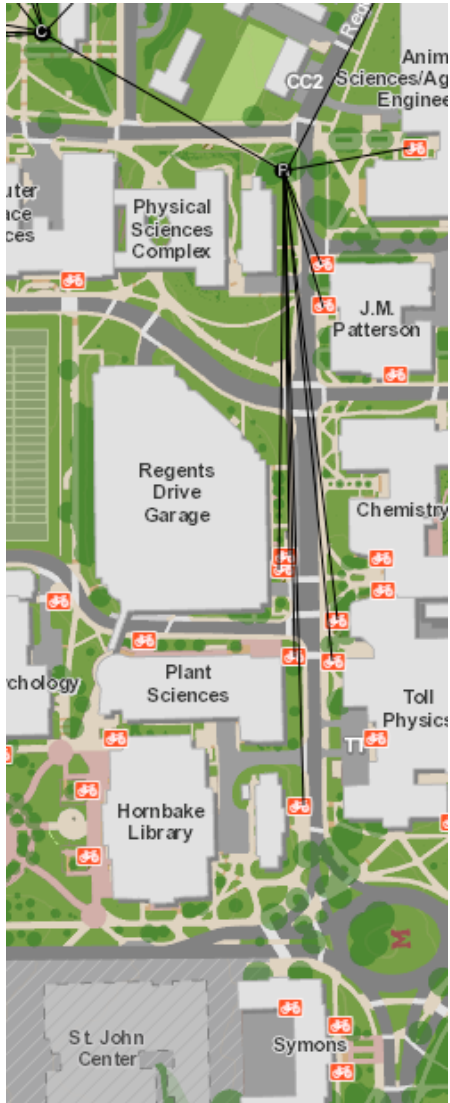


Figure 53. Cambridge to Regents Routing Map Two.

Within the figures, all of the lines represent a stable communications link between the two nodes, whether it be bicycle rack to router, bicycle rack to coordinator, or router to coordinator. Through analyzing these figures, several key aspects of the XBee modules can be observed. First, observing the Cumberland Hall bicycle racks shows the

XBee's inability to propagate through the dormitory to reach the bicycle rack on the west side. However, viewing the bicycle rack on the east side of Centerville Hall shows that the coordinator was successful in propagating through the dormitory. The difference observed may be due to the amount of structure that the RF waves had to pass through due to the difference in angles between the stationary node and the bicycle rack. Additionally, the distance from the building to the bicycle rack was further in the Cumberland case, meaning that once through the building the waves still had to propagate further to reach the bicycle racks. The distance to wrap around the southern side of the building was also shorter in the Centerville case, which lead to the possibility of the wave reaching the destination in another fashion through leveraging multipath, or the RF signal's ability to bounce off of objects. Regardless, it is recommended that further network tests be completed to ensure that the link is reliable in all situations.

An interesting result of the experiment was that bicycle racks that were completely obstructed by buildings, such as the ones located in front of the School of Public Health, consistently had no communications link with the network. Additionally, the bicycle racks that almost had LOS but were far away from the stationary node also failed to have a stable communications link, such as the bicycle racks that were slight back from Regents Dr. These observations are critical to note when optimizing node locations to maximize coverage.

Proposed network system. Taking into consideration all of the information from the tests conducted, the next step was to provide a strategy for developing a campus-wide

network. To make the problem simpler, the objective is not actually to cover the entire campus, but rather to cover every bicycle rack on campus. Then taking into account the fact that there exist bicycle racks that are not used by the target student population and that the initial system may not need to cover the entire campus, a target of 90% of all bicycle racks on campus is selected. The goal of the proposed system is then decided to maximize coverage of bicycle racks with a minimum of 90% of all bicycle racks covered, minimize number of stationary nodes (routers and coordinators), and minimize cost. It is important to note that the requirement to minimize nodes and cost are different due to the coordinator cost exceeding that of the routers because of the Internet capability requirement. In other words, it is preferred that the network be expanded through the use of an additional router, rather than implementing a new network through a new coordinator should the router still provide a sufficient amount of additional coverage to the network.

Before the proposed system problem is tackled, a quick analysis of the system set up through the tests is completed to aid in the process. The coverage and network tests were completed in three different locations: McKeldin Mall, the Ellicott Community, and the Cambridge Community to Regents Dr. At McKeldin, one node configured as a coordinator ran a single network that had connectivity to 14 bicycle racks. At Ellicott, one node configured as a coordinator ran a single network that had connectivity to 7 bicycle racks. And at Cambridge, three nodes in a single network has connectivity to 21 bicycle racks. Leveraging the interoperability of the multiple networks, all three can be

combined to consist of five stationary nodes reaching 42 bicycle racks. Given the cost of each node is about \$40 with an additional cost per coordinator of \$40, a quick cost analysis shows that the three networks combined would have an installation cost of approximately \$320. The total cost per bicycle rack then comes out to be around \$7.62. Granted, the tests were conducted on some of the more dense areas on campus in regards to bicycles racks.

After running the small scale math on the collected data above, the proposed system was then drawn out. Using the data collected from the coverage and range tests, the XBee modules' ability to maintain connectivity with distant points was characterized. In the proposed system, the model assumes that the installer has access to higher elevation areas such as rooftops or edges of buildings to enhance the LOS capabilities of the devices. The UMD Campus Web Map was again used as a reference for the bicycle rack locations, and another note to make is that the bicycle racks located at Fraternity Row and off campus housing complexes were not considered. The campus was then broken into subsections based on how dense the bicycle racks were. Within each section, proposed node locations were listed as GPS coordinates that typically would be elevated to maximize coverage. The result of the analysis is shown in the table below in Figure 54.

	Location	Covered	Bike Racks	Nodes	GPS		
1	Denton Community	8	8	1	38.992859, -76.948861		
2	CSPAC	4	6	1	38.992009, -76.949612		
3	Ellicott Community w/Eppley	9	10	2	38.991871, -76.946189	38.992338, -76.945851	
4	Cambridge Community w/SPH	15	17	2	38.992505, -76.943877	38.992688, -76.942032	
5	Stadium Dr & Regents Dr.	32	33	4	38.990257, -76.937107	38.991066, -76.936496	38.990257, -76.940551
					38.988944, -76.940830		
6	EGR/MTH	6	9	2	38.988189, -76.939296	38.988773, -76.938481	
7	Hornbake/Stamp	24	25	3	38.987814, -76.943169	38.988435, -76.943196	38.987305, -76.945036
8	Tawes	8	8	3	38.986555, -76.948598	38.986342, -76.947563	38.986225, -76.946677
9	McKeldin	11	11	1	38.985983, -76.944671		
10	Armory	7	7	2	38.986196, -76.939457	38.985867, -76.939457	
11	Tydings Area	12	13	2	38.984703, -76.941849	38.984695, -76.944043	
12	Van Munching	19	20	3	38.983786, -76.946924	38.983953, -76.947670	38.983557, -76.945637
13	South Campus	33	34	7	38.982272, -76.944237	38.982539, -76.944253	38.982685, -76.944242
					38.982556, -76.943314	38.982243, -76.943115	38.982539, -76.942069
					38.983511, -76.944226		
14	SCC 3/4	10	11	1	38.981522, -76.940787		
15	Montgomery	7	7	2	38.981921, -76.939612	38.981931, -76.939033	
16	Xfinity	7	11	2	38.992986, -76.939637	38.994937, -76.940516	
	Total	212	230	38			
	Percentage Coverage	0.921739130434783					
	Node Cost	1520	\$40/Node				
	Coordinator Extra Cost	640	\$40/Network				
	Total Cost	2160					

Figure 54. Proposed Campus-Wide Network.

The table shows the subsections of campus, the locations of each network, the number of bicycle racks covered by the network, the total number of bicycle racks within the subsection of campus, the number of nodes in the network, and the GPS locations of each of the nodes in the network. As seen from the table, the proposed system would include 16 interoperable networks with node amounts ranging from a single coordinator up to seven stationary nodes. The system would cover 212 out of the 230 bicycle racks in the defined regions on campus, resulting in 92.2% coverage, exceeding the 90% requirement. At the same estimated costs per node and extra costs per coordinator, the system came out to be approximately \$2,160. The cost per bicycle rack then was calculated to be \$10.19. The value was slightly higher than the real data test, but since

the data was collected at more densely populated bicycle rack areas on campus, this result was expected.

In order for the network to cover all of the bike racks on campus, the bike racks missed in the calculations above would require their own independent networks. This would result in 56 total nodes at a cost of \$3600.

Integration

Following the independent development of each sub-system, the entire team coordinated the integration of the three sub-systems into a bikeshare ready for testing. The final goal of the integration process was to demonstrate that the Arduino on-board the smartlock could process information from the NFC and GPS modules and communicate with the MongoDB online database through the XBee network. The system would thus need to correctly authenticate users and prompt the appropriate actuation of the locking mechanism. The integration process was separated into four demonstrations, progressively improving on the previous to ensure proper operation of key functions through all demonstrations. The first demonstration aimed to establish a channel of communication between the XBee end device on each smartlock and the XBee coordinator connected to the Mongo database. The second demonstration configured authentication by adding the NFC reader communications to the XBee end device and successfully passing messages back and forth with the database. The third demonstration scaled the second demonstration by relaying communications between the XBee end devices to an XBee coordinator through an XBee router acting as a node to introduce the idea of an XBee network. The final demonstration integrated the mechanical components to digitally lock and unlock the smartlock based on a proper authentication message. Each demonstration extended the possibility of each individual component in an incremental fashion until the team achieved a system ready to be scaled

for a potential test implementation. System diagrams visualizing these four demonstrations are included in Appendix B.

Integration Demo 1: Arduino to Website through XBee

The first demonstration established the connection between an XBee network and the Meteor website. The objective of the first demonstration was to prove that the XBee network could successfully pass information from an end device to a web application, which could include GPS information to aid in locating available bikes or lock status among other pertinent information. The demonstration included a laptop running a Node.js parsing application and the web application, two XBee devices, and end device (AT mode) and a coordinator (API mode), and an Arduino to simulate the smartlock.

In the completed bikeshare, information would be regularly passed from each bike, through the XBee network, and then regulated by the server. To simulate this structure, an XBee end device was connected to an Arduino, which repeatedly printed a CSV formatted string with fake data to mimic the way the Smartlock would operate. Upon receipt of the CSV string, the end device would wirelessly send the message to the coordinator, which would relay the message over a serial USB connection to the laptop. During the wireless transmission between the two XBee devices, the message was transformed from a CSV string to an XBee packet. To interpret the message, a parsing application was necessary.

To communicate between the XBee coordinator and the hosted web application, an intermediary was necessary. Connecting directly to the hosted application would be

infeasible because the direct connection would require hosting the application from a local server, which would be cost-prohibitive and more prone to outages. The intermediary would thus aid in scaling the system and provide a cost-effective solution to managing communication. The intermediary was built using the same platform that Meteor was built on top of, Node.js. As a barebones application, it would only need to maintain a serial, USB connection, parse incoming data, and securely connect with the hosted application. To accomplish each of these tasks, the intermediary applications used a series of open-source Node packages managed through the Node Package Manager (NPM). The intermediary application could then be run on any small computer connected to an XBee coordinator. The application would insert the collected CSV information through the secure connection to the hosted web applications database. The data would then be displayed in real time to any user viewing the hosted Meteor application.

Integration Demo 2: NFC User ID Wireless Authentication

The second demonstration aimed to create a communication protocol between the smartlock electrical system and the intermediary Node application for accessing the Mongo database in order to perform user authentication via NFC. The demonstration required the prototype smartlock electrical system, XBee coordinator, Raspberry Pi, and two NFC cards. Of the two NFC cards used for the demonstration, only one card was added to the list of approved users in the web application.

To start the smartlock, the user would first push a button to initiate the NFC reader. Once the reader was active, one of the two NFC cards was placed next to the

reader for communication, which would collect a hexadecimal code identifying the NFC card. Using the protocol specified in demonstration 1, the Arduino would collect available information and create the CSV string seen in Figure 55.

```
<RSSI>, <LATITUDE>, <LONGITUDE>, <ALTITUDE>, <TIMESTAMP>
```

Figure 55. CSV string.

Using the communication process proven in demonstration 1, the information would be relayed to the Meteor database through the XBee wireless connection and an intermediary Node.js application. However, in this demonstration, the ability of the XBee modules to communicate bidirectionally and authenticate a user was tested. Upon receipt of the new data, the Meteor application would determine if the NFC card identified was approved for access and would return a false value accordingly. To monitor this process, the application included a table of received and sent values and the Arduino and NFC device would output serial data to see the data transfer in real time.

The authentication process compares the received NFC code to a server side-only database of registered users. This step could also account for billing or other parameters necessary for the monetary function of the bikeshare. The user information in the database was stored simply as a hex string, rather than the actual user's name, university ID, or any other form of personally identifiable information (PID) to simulate the need to protect the users' information in the final design. Once the user ID string was run through the database comparisons, an "authorized/unauthorized" flag was set

accordingly. If the identification code matched a registered user who was in good standing, the database would return an authorize flag. Any other case would return an unauthorized flag and permit the lock from being opened. To explain this issue to a user, a notification, email, or text message could be sent to alert the user of the issue and suggest an appropriate solution. Errors that might need fixing may be from an improperly linked credit card, a duplicate rental still open, or a more vague error that can be solved by connecting the user to customer support.

Once the authenticity of the user was determined, the system assembled an XBee packet that activated a specific XBee end device using the *<Module_ID>* field from the CSV string. With the packet configured, the AT request was sent to the XBee coordinator from the intermediary application and then to the specified end device. The end device received the message, which specified a specific pin to write as HIGH or LOW accordingly. After initially sending the user information to the coordinator, the Arduino changed logic states to constantly poll the specified digital pin of the XBee module. While the pin was LOW, the authentication flag was set to false. Should the pin transition to HIGH, the authentication flag transitioned to true. The intermediary Node application would then wait a set time delay before sending a LOW message again to reset the system after the pin was read. The authentication flag dictated the change in the Arduino's state when deciding whether or not the smartlock should be unlocked for the user. To demonstrate the functionality of this state change, an LED was added to the

smartlock prototype; a steady LED signal indicated the received signal was for an authenticated user, while a blinking LED indicated that the received signal denies access.

As expected, when the registered NFC card was used, the LED was lit continuously to signal success while using the other the NFC card resulted in the LED flashing. The second demonstration was successful in relaying the NFC authentication information from the XBee end device to the Mongo DB.

Integration Demo 3: Multiple Smartlock Prototype Units

Following the second demonstration, the prototype system needed to be expanded to support multiple end devices and multiple routers. The objective was to prove that the ZigBee protocol implemented through the use of XBees could be easily scaled with the authentication process. In the iteration, two end device systems, two routers, and one coordinator system were used to demonstrate the coordinator's ability to handle multiple transmissions, the interchangeability of users' NFC cards, and the XBee's ability to use intermediary routers to expand the network size.

The prototype smartlock system used in the previous demonstration was duplicated to act as an additional bicycle unit in the proposed bikeshare. With both units functioning independently, the two were connected to the same coordinator node. To simulate two users requesting a bicycle at the same time, an NFC card was presented to each of the prototype smartlocks simultaneously, and as expected, the web application received the CSV strings from both without any loss of data. When alternating the NFC

cards between the smartlocks, the CSV strings arrived with swapped user fields accordingly.

The final step of the demo took the end device system out of range of the coordinator to show how when a bike would lose connection with the XBee coordinator, it could reconnect to another available router using the ZigBee mesh network protocols. With the end device no longer visible by the coordinator XBee, the end device could still see a router and send the CSV formatted data. Even with the packet being routed through an additional node in the network, the website continued to accurately receive live data, demonstrating the scalable routing capabilities of the mesh network.

With the success of a multiunit system, the team sought a way to further scale the system and to circumvent the need for a laptop tethered to the coordinator. Using the cost-effective Raspberry Pi, the intermediary app was installed and ran with a directly connected XBee coordinator. This system was more power efficient, cheaper, and space efficient allowing it to be installed throughout campus as a bikeshare and in strategic locations. By improving the overall efficiency of the XBee-Meteor interface, the cost of developing a multi-coordinator network becomes significantly reduced. With the network proposed in the implementation section, the change affects as many as sixteen nodes, greatly improving the overall bikeshare network.

Integration Demo 4: Locking & Access

The fourth and final demonstration aimed to integrate the electrical and the mechanical components of the smartlock. Using the system prototyped through the first

three demos, the focus shifted to controllably locking or unlocking the bike lock using the entire communication pathway. This demonstration focused on extending the complexity of the electronic system to directly control access to the lock. Similar to the last demonstration, two NFC cards were used but only one was registered to the online database. Either NFC card would trigger a query of the hosted web application database, but only the approved card would return an approved signal and thus a ‘y’ signifying a ‘yes’ message to the XBee. Once triggered, a servo arm would rotate to start the unlocking process. The servo arm provided enough force to compress the springs and move the pin constraining the rotating pieces of the designed smartlock to unlock. As expected, when the NFC card registered with MongoDB communicated with the NFC reader, the servo motor was actuated to move the pin while use of the other NFC card did not result in any motion, sending an ‘n’ signifying a ‘no’ message to the XBee. The fourth demonstration was successful in integrating the mechanical components of the product to the electronics to create a “smart” product.

Bikeshare Design

Implementation of a successful bikeshare requires the integration of various parts such as a working communications network, functioning user interface, and other infrastructure among many others. Any business hopes for a paying and enthusiastic user base to be successful, and a business centered on bikeshares is no different. This section looks to explore why a college campus would want to implement a ZigBee connected smartlock-enabled bikeshare system. Operational responsibilities of the campus will also

be discussed as well as costs associated with system hardware and payment structure that any user would be subject to. Additionally, this section will discuss and propose potential solutions to common problems associated with traditional bikeshares as well as problems specific to a stationless campus bikeshare. Finally, the team looks at the total cost excluding operating costs to create the system.

Operations. Upon implementation on the campus, the university will take on the responsibility for managing the operation of the system. These responsibilities would include but not be limited to: maintaining an appropriately sized fleet of bicycles, maintaining user information, storing data collection if deemed necessary by the individual university, reporting theft and vandalism, supporting users, maintaining and expanding the ZigBee grid if there is demand for a larger coverage area on campus, and managing user access, payment collections, and fee scheduling. Most practically, bikeshare operations would be delegated to the university's transportation department. Specifically for the University of Maryland, the prime responsibility of bikeshare operation would fall upon the Department of Transportation, under which BikeUMD currently operates under.

Payment structure. In order to continue operations, the bikeshare would need to create consistent income from the subscription type service. An example of the payment structure of Capital Bikeshare is shown in Figure 56.

Membership Fee	
24-hour	\$8
3-day	\$17
Daily Key	\$10 initial fee + \$7/day
Monthly	\$28
Annual	\$85
Annual with Monthly Installments	\$96 (\$8/month for 12 months)

Figure 56. Membership and usage fees for users of Capital Bikeshare.

Due to the bikeshare’s target market and anticipated users, the payment structure needs to be favorable to a college student’s budget and schedule. The challenge of the bikeshare system will be to charge enough money to operate with close to no net losses but not enough to discourage already cash-strapped college students. This ideal pricing system will maximize the number of users on campus while satisfying the system’s monetary needs. Furthermore, the structure and term of the subscription should be flexible enough to cater to the traditional semester college schedule. Maryland’s Department of Transportation already offers parking via a semester-by-semester style, and this system would also work well for college students.

Students can subscribe to the system for a semester, month, week, or even a day-by-day basis and add money to their account when they need to once they have purchased the RFID card that gives them user-access to unlock bicycles. The RFID card is the way that students will be able to unlock the bicycles. In addition to students, visitors to the

campus such as prospective students would be able to purchase time to ride around campus; thus they would be able to see more of the campus in less time than the time it takes to walk only a quarter of the campus.

In addition to cost, the bikeshare administrator can also determine how ride time is allotted in each subscription. Two examples are the following. One subscription type would allow riders to essentially use their subscription payment as a debit to use on the bikeshare. Each ride would then deduct from their account based on the duration of the ride. The amount would then be capped to either a daily or weekly limit, which would prevent abuse or monopolization of a single bicycle. If a rider uses the allotted time that their subscription covers each day, they would be able to add money to their account through the website.

Another alternative is that users can have rides allotted so that a rider gets a certain number of rides per week. The rider could space out their rides per week around their schedule. Since the bikeshare is centered on a college campus, each ride should not take any longer than twenty to thirty minutes.

In the end, it would be up to the administrator to experiment with either method of subscription service. Different college campuses may have students with different preferences, and for that reason the term structures should be open to the university and the above are merely suggestions rather than strict rules.

Fee structure. In order to make sure that no one person can abuse the bikeshare by monopolizing a single bike for personal use, fees would be debited to the riders

account when times go over the allotted amount credited to the account. Other bikeshares have these fees set up around in thirty minute to hour-long increments. To determine the best amount, more consumer outreach would need to be conducted. However, as a rough estimate the fee should be set at \$1.00 for every thirty minutes over time. As an example of a traditional day, a student has a semester long package that grants unlimited rides per day where the cumulative length of those rides is less than 45 minutes. If a rider takes one 45-minute ride or 45 one-minute rides, all of that is covered. Once they pass 45 minutes per day, a \$1.00 fee would be charged to their account and would show as an outstanding balance.

Fleet size. Being operated in and around a college campus makes this type of bikeshare unique compared to other bikeshares in metropolitan cities. Unlike large metropolitan cities, which need to maintain a near constant fleet size, a bikeshare targeted to students does not need to maintain a fixed fleet size. Rather, it can adjust the size of the fleet based on student demand, which will fluctuate during the school year. The administrator can cut back the size of the fleet during off cycle times such as summer when the majority of student are home on break and also during the winter months when the weather may impact the demand for biking. One challenge will be wear to store the excess bikes during these times; however, by keeping the size down, the overall cost of maintaining the system lessens.

How to use. First and foremost, the team is under the assumption that the potential rider is familiar with how to operate a bicycle safely, and while not required by

law it is recommended to always wear a helmet or some other form of headgear. After signing up for a subscription, students will be given a bikeshare unlock card. This card may come in a couple forms. It may come as a RFID sticker that can be placed on the student ID card administered by the school or the student may be given an RFID card similar to the metro fare cards used on the Washington, D.C. Metro.

Once registered, the student can download the mobile application, which will highlight available bicycles, areas where bikes can be taken, and the enrollee's subscription status, which features time available and/or rides available before fees are tacked accrued. After locating the closest available bike and a destination, the rider need to unlock the bike to begin the ride. After checking the bike for any visible signs of damage or wear, the user must press the start button located on the electronics box of the lock. This button powers the system up from rest mode, which is used as a power saving measure. Powering the lock on will activate the location feature and open the communication up with the central database. The user will take their card and tap to unlock. Maneuvering the lock off from the post and sliding the lock into the riding position, the bike is now in the hands of the user. He or she is free to ride to the destination. Once the system unlocks the bicycle, the ride has started, which means that the timer for the day officially begins. This clock stops officially once the bike is properly locked at the conclusion of the ride.

Once the rider has gotten to the destined bike rack, they can begin to lock the bicycle. Sliding the lock from the riding position back into the lock position, the user

taps the RFID card again to open the lock and place it around the post of the bike rack. Failure to properly lock the bike around the bike rack or within the coverage area, the timing clock will continue to roll. If this is the case, a notification will be sent to the user's phone giving them a warning about approaching the day's allotted time. This will serve as a reminder if locking was not completed properly. After the user is finished and has locked the bike, the database changes the status back to available. At this point, it becomes visible by all the other users when they open the bikeshare's map.

Common bikeshare problems. Almost all bikeshares, stationed or stationless, encounter common problems including the redistribution of bicycles after each day, the safety of the users, theft of the bicycles, and maintenance. Capital Bikeshare and CitiBike face major problems with the unequal density of their bicycles, specifically at the end of a work day. Users tend to bike to work and then take another form of transportation home, leaving many bikeshare bicycles in the city rather than in the suburbs where they were initially picked up, resulting in empty stations in areas where users would be picking up bicycles in the morning. To combat this and prevent users from giving up on the bikeshare, companies use trucks to manually redistribute the bicycles to empty stations at night. The safety of the users and of the bicycles are also integral in maintaining a working bikeshare. This means ensuring that the bicycles are properly maintained, which will increase the safety of the riders.

Redistribution. The bicycle density problem of Capital Bikeshare and other city bikeshares arises because of complex commuting patterns. The commuting patterns of

users are likely more simple because the bikeshare will be used almost solely for users to get across campus quickly, most likely to go to classes. However, the team understands that users may not want to ride bicycles up inclines, so bikeshare bicycle density at hilltops such as Stamp Student Union may be lower than the bicycle density of the Chemistry building. Bikeshare users will be incentivized and encouraged to move bicycles to hilltops or to other areas where bicycle density is low. For example, if users redistribute bicycles to areas of low density, they can receive benefits such as a free ride or coupons from sponsors of the bikeshare. If they wish to help in redistribution, users may receive push notifications on the mobile application with instructions on where to ride the bicycle to. Once the redistribution act is complete and confirmed, the user will receive the benefit. Naturally, this solution can only be implemented after it is clear where the areas of low bicycle density are on campus and at what times.

Safety. Another issue that could arise with the system is inevitably the safety of the users while riding the team's bikeshare bicycles. Accidents could occur with the user while riding bikeshare bicycles which could include other parties. To minimize these occurrences, users will be required to sign a member agreement and liability waiver prior to their use of the bikeshare. Terms outlined in the waiver will follow Capital Bikeshare's liability waiver and release. The waiver will state that users release any claims they may have against the bikeshare including but not limited to claims, injuries, demands, liabilities, disputes, damages and losses associated with the bikeshare. Essentially, the bikeshare cannot be held responsible for any losses, damages, or injuries

that occur when the user is operating a bikeshare bicycle, including those for the other party of an accident. However, because the safety of the bikeshare users is key, when they sign up for the system, they will be encouraged to practice safe bicycling such as wearing helmets and avoiding using cell phones or bicycling in extreme weather.

In addition to user safety, the safety of bystanders such as pedestrians and automobile drivers was considered. The user would be required to sign a waiver that state that her or she will be liable for any damage to property and life. If an accident were to occur, the responsibility for all liabilities would fall into dispute between the user and the other party. The bikeshare will provide robust methods of identifying bikes on the road including, but not limited to, front and back lights and bells. Furthermore, as stated in the previous paragraph, the user will be encouraged to use best accident preventive bicycling practices that protect both the biker and those on the road.

Theft. To combat the theft of bicycles and equipment, precautions will be built into the bike around the seat post facing the back wheel in an attempt to limit the amount of theft. The lock and electronics box themselves will be welded to the bicycle and unable to be removed. The lock will be built with a pressurized compartment of ink within its metal arms. This simple device will act as a deterrence from cutting the locking arms to steal the bike. If cut or broken, the ink will explode caused from the heat and pressure associated with cutting. The ink will mark the ground as well as anything else around the bicycle. If a bicycle is stolen, the ink mark will differentiate user error from theft. The bicycle will be outfitted unique bolts that allow only bikeshare

administrators to unlock them. This will cut down on the common theft of wheels and other easily removable items on the bike. Although no system is safe from theft or vandalism, these precautions built into the bike will hopefully deter theft and other misuses.

Maintenance. An important problem in the upkeep of bikeshares is that when the bicycles are broken, they need to be fixed right away to ensure the safety of users and to increase the longevity of the bicycles. Since the system is stationless, bicycle maintenance needs to be more than manually checking each station and ensuring the bicycles of each station look well-maintained. The bikeshare bicycles can be found by GPS, but it would be a hassle to search for each bicycle on the entire campus for maintenance checks. Instead, the bikeshare will utilize the user to give an alert when he/she thinks there is a maintenance issue. The bikeshare application will allow users to report problems with the bicycles and include pictures. With these reports, the bicycle can be found and the problem can be fixed before any other users ride the bicycle, which could minimize potential injuries resulting from faulty machinery on the bicycle.

Unique bikeshare problems. Since the bikeshare has features that are novel and not present in current systems, potential unique problems are created. These include: improper locking, protecting the electronic housing and on-bike power system, and the limitation of having an open coverage area. The team looks to extrapolate and mitigate these problems with the suggestions in the following sections.

Improper locking. Unlike traditional bikeshare where the user needs to return a bike to a docking station, this bikeshare allows the user to lock a bike to any bike rack in the system's coverage area. A user may improperly lock a bike by locking it outside this area or they simply may not lock the bike properly to a rack. As described above, the system will alert a user if they are almost up with their allotted time. This may be a signal to the user that they did not properly lock the bike in the coverage area, or failed to properly lock the bike even though it is inside the coverage area. If a user properly sets the lock in a locked position but fails to lock the bike around a rack, the bike will be registered as available in the system since it is in the locked position. Unfortunately, since the bike is not properly locked, someone not registered can easily take the bike. Since the user does not need to unlock the bike to use it, the system still thinks the bike is available at that location since the lock was never powered on or unlocked in the eyes of the system.

This second scenario is caused because of user error whether intentional or not. Education during the on boarding process on how to properly lock the bike around a bike rack can help to lower the number of incidents of this type. Currently, the pressurized ink theft deterrence would give insight into whether a bike was stolen or whether user error resulted in the bike being taken. If no ink is present on the ground or around the last known location of the bike, it could be assumed that the bike was taken as a result of the failure to properly lock the bicycle. If this is deemed the case, the cost of the bicycle,

which include the cost of the customized lock, parts, and hardware stolen would need to be paid by the user who is at fault.

Other precautionary measures to inform users of properly failing to lock a bike around a rack could be built into the system. These could include making the bike immovable while the lock is in the locked position. While in the locked position, the lock itself may prevent the rear wheel from turning. This would cause an improperly locked bike to have to be carried away rather than rode away. Carrying a bike would easily deter the theft of it since this would attract attention to the crime in process. The team is also actively thinking of other ways to build precautionary measures to deter theft from user error.

Electronic housing and battery. Another unique problem of the bikeshare that could arise is the maintenance of the electronics housing and battery on each bicycle. Every bicycle will have weatherproof electronic housing to maximize the lifespan of the electronics within. As the battery powering the electronics on the bicycle is not rechargeable, a replacement is required at the end of the battery's lifespan. To prevent the possibility of a user riding a bicycle with a dead battery, a fail-safe will be built into the code so the bicycle cannot be unlocked and used when the battery is running low. This will minimize the possibility of losing the location of a bicycle or having a bicycle improperly locked due to a dead battery.

Coverage area. Unique to the bikeshare is the limit imposed by the coverage area, which poses multiple challenges to the operation of the system. Unlike traditional

systems where the bounds of use are restricted to locking the bicycle at docking stations around the metropolitan area, this stationless bikeshare has no defined bounds. Users may be unsure of where and how a bicycle can be locked without incurring fees for improper locking or taking a bicycle outside the range of coverage. To minimize this problem, the team proposes the following solutions.

First, to inform students of the coverage area, the bicycle racks in the bikeshare zone can be found on a campus map on the bikeshare's homepage and phone application. A user can quickly check if a bicycle rack is in the bikeshare zone before he or she starts their journey. In addition to students being able to check the bikeshare's campus coverage map, the school may choose to post signs or stickers on the bike racks signifying that the rack is covered in the ZigBee grid.

The next problem that the bikeshare can address is the limitations of the coverage area. When large metropolitan areas need to expand their network it takes time, money, and space. Planning and funding need to be resourced and allocated before the proper steps are taken to construct a docking station. A ZigBee connected system can be expanded quickly wherever there is user demand. Since the bikeshare relies on already available infrastructure in the form of bike racks, the cost is only limited to the monetary amount associated with a coordinator unit. This allows the administrator to quickly expand the coverage area to areas with weak coverage on campus, off campus housing units, and other popular destinations off campus.

By implementing this type of bikeshare, the administrator can create and modify the coverage area to reach all the places that students are likely to frequent. The area covered can be added, removed, transferred, or strengthened relatively quickly, easily, and affordably. All of these factors give the ZigBee connected bikeshare multiple advantages over competitors.

Cost analysis. Team BIKES took on the project as a means of providing a cheaper alternative to bikeshare administrators, especially college campuses. Looking at a number of bikeshare systems around the country, the team wanted to gauge how many bikes would be needed on the University of Maryland’s campus. Figure 57 below shows the number of bikes per 10,000 people in that bikeshare’s target population.

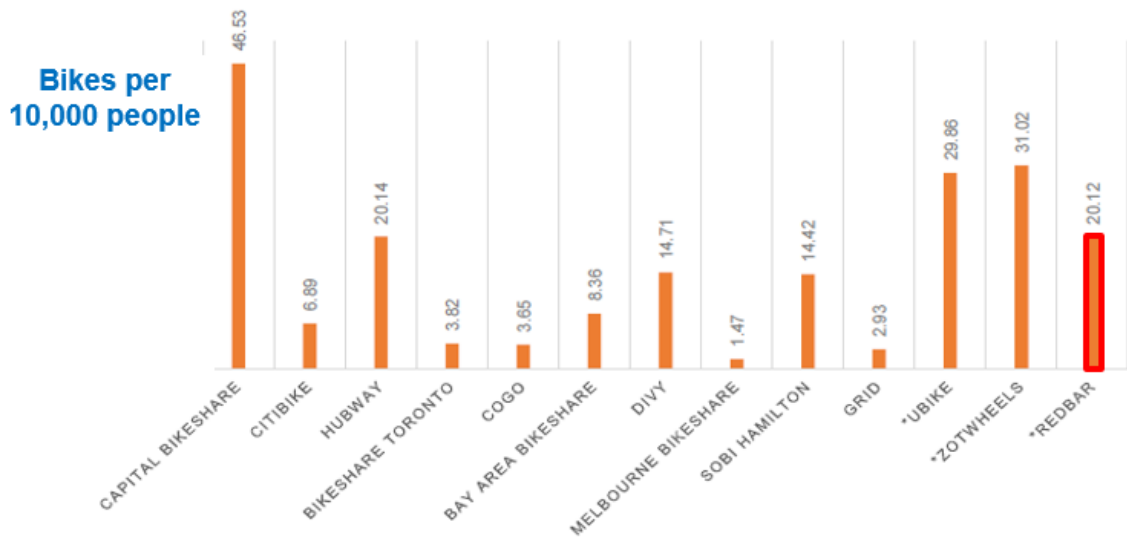


Figure 57. Number of bikes per 10,000 people in bikeshares around the country.

In order to attain a distribution level on par with metropolitan bikeshares, the team would need to purchase 80 bikes. However, to be on par with the other college bikeshares, U Bike and Zotwheels, the team would need to have 120 bikes.

According to the Diamondback, the University currently has plans to welcome a stationed bikeshare to campus in May, 2015. The bikeshare has plans to construct seven stations with 120 bikes, and the school received a grant for \$375,000 to help with the costs of the system (Lang, March 2016). The breakdown of Team BIKES proposed system comes at a much cheaper cost than the grant the school was awarded. The costs that the team used in the estimates are rough and most likely much greater than the actual costs when purchasing materials, bikes, and labor in bulk. The costs are below in Figure 58.

Bikeshare	
Bike	\$ 250
Lock (Materials and Labor)	\$ 660
GPS	\$ 40
Arduino	\$ 35
NFC Reader	\$ 20
Misc.	\$ 20
Total Per Bike	
	\$ 1,025
80 Bike System	\$ 82,000
120 Bike System	\$ 123,000
Mesh Grid	
Routers	\$ 40
Coordinators	\$ 80
22 Router System	\$ 880
16 Coordinator System	\$ 1,280
Total for Mesh Grid	
	\$ 2,160
Total Bikeshare Cost Range:	
	\$ 84,160 - \$ 125,160

Figure 58. Cost Analysis.

Implementing a 120 bike system on Maryland’s campus would cost approximately \$125,160, which is only 33% the cost of the grant that the university was awarded. Without factoring in operating expenses, the costs associated with Team BIKES system is a dramatic reduction in upstart costs, and this as well as all the other benefits of the team’s proposed system make it a very attractive product.

Conclusion

Technological Achievements

Bikeshares have continued to become ingrained into modern cities as an economical, environmental, and enjoyable mode of transportation. College campuses would benefit from similar systems; however college students have different needs than most users in cities. Students will take greater number of shorter trips, where parking location is of utmost importance. To meet the need of students, the bikeshare technology must find a balance between flexibility in bike parking and security. The RedBar smartlock achieves both through the innovative approach of a novel mechanical design and NFC-controlled access, scalable communication system, and real-time web application. With the introduction of the smartlock, biking can become a hyper-efficient and enjoyable transportation method for college students across the country.

While the team relied on personal experience as college students, the team drew most of its inspiration and insight from questioning peers. The team received both formal feedback from the focus group discussion and informal feedback through conversations with peers. Both forms of feedback drove the team's iterative design process and helped us focus on the systems level components that mattered to college students. Necessary components included pricing, quality metrics of a bikeshare smartlock, and user expectations, among other relevant topics.

The mechanical lock and housing design subteam incorporated user feedback, primary research, and an iterative design process to create a novel smartlock mechanism and device. The Locking subteam examined the leading bike lock designs and sought to better understand the material selection and design flaws of the current available options. From the primary research, the subteam developed two unique smartlock designs and selected the loop lock, considered the most intuitive and reliable when tested with users. For the final design, the team was able to manufacture a steel lock to demonstrate the techniques necessary for production and the strength of a proper material. Through the fabrication, material testing, and user feedback, the Locking subteam brought insight into what the next generation of bike locks may offer for a campus bikeshare future.

To digitally secure the bike, the team needed a technology that met the user needs for accessibility, but met the technical needs of a large, multi-user distributed bikeshare. Similar to the approach taken in addressing the mechanical needs, the team chose to begin by researching the state of the art and gaining an understanding of access technologies. Elechouse NFC 2.0 was selected over RFID and Bluetooth because of the low power requirements, protection against man in the middle attacks, activation distance, and other critical features. Integrating the technology into the Arduino controller, the team demonstrated how users could be verified and securely activate a lock or unlock event. Integrating the digital control with the lock proved challenging. Working closely with the Locking subteam and after designing multiple approaches, the

team finalized a mechanical pin system that met the engineering and user constraints of simplicity, reliability, and low power needs.

Creating the technology to coordinate an often haphazard distribution of bikes requires highly effective and reliable communication technology. Losing communication with a bike can be an expensive mishap and out of date bike information can frustrate users. The Geolocation subteam considered GSM/GPRS network or local BlueTooth modules to communicate back to a central database, but found that both technologies would be too expensive. Instead, the subteam utilized XBee modules to prototype a unique, scalable, and campus-wide mesh network able to support hundreds of bikes. The team proved the application of this technology by collecting signal strength readings on McKeldin Mall and other locations on campus. While XBee modules are mostly used in small scale internet of things implementations for a home or small building, the outdoor application of a mesh network for a bikeshare is novel. The RSSI readings and analysis of XBee modules for bikeshare applications will expand the potential application of this technology to meet the need of cost-effective stationless bikeshares or for large scale internet of things technology.

The last component of the system was an effective web portal for users to view the data collected in real time and for administrators to actively manage a large distributed bikeshare. The team evaluated multiple options and chose to pursue a newly developed technology with high potential, Meteor.js. The Geolocation subteam built a real-time web application hosted at <http://redbarbikes.com>, where bikeshare users can see

the location of any available bike and learn more about the system. Incorporating feedback from the focus group, the subteam implemented features such as a reservation system and improved the onboarding experience. To benefit future bikeshare developers and existing ventures, the team chose to open-source the project and released several reusable packages applicable to a larger audience.

The combination of a novel mechanical smartlock design, digital access control, a scalable communication network, and an open-source web application allow for the next generation of bikeshares across the nation. Based on the continuous feedback and input the team received from university students, the technology and user experience are crafted for the unique aspects of a college campus unlike any other bikeshare technology. The stationless component particularly stands out when compared to station-based bikeshares as it can utilize existing infrastructure without changing the architectural character or requiring major construction. The smartlock concept additionally offers flexibility in bike-type selection, improved scalability, cost-effective benefits, and an enjoyable user experience unmatched by current technologies.

Lessons Learned

Developing these novel technologies took overcoming a wide variety of challenges that are applicable to any similar endeavor. The team had to build cohesion and surmount the vast technical challenges posed on several fronts. To breakdown the content, the team split into four subteams, each tackling a portion of the problem. The teams were not only selected by the type of technical challenges, but also represented the

unique skillset of each member. The project was initiated to develop an energy-translation device aimed at making biking safer. Realizing that the team could make a broader impact to bicycling safety by reducing the barriers to entry in cycling with a bikeshare, the team pursued a new idea and reorganized the team accordingly. As ideas continually morphed and new technologies were selected, and a website was added, the team struggled to stay focused and coordinate thirteen members. In order to ensure cohesion, the team created an explicit organization structure with team leaders, subteam leaders, and other specific roles.

An additional future implementation would attempt to reduce the cost of each bike unit by leveraging the signal strength readings from XBee modules to geolocate the bicycles. Taking the signal strength, the known locations of the routers, and the known locations of bicycle racks as the only possible bicycle locations, it may be possible to eliminate the need for a GPS in the smartlock, thus reducing the cost of each smartlock by \$40. Future research could be done to determine if this is a feasible geolocation option.

Each subteam faced unique challenges related to the technical components. The locking subteam struggled with designing both for manufacturing and for a prototype. Often designing for one type of manufacturing can be misleading as there are many ways to produce a product. Instead, effort is better spent building a prototype with some thought of manufacturing. While developing for a mature and well-established product, the XBee modules, the Geolocation subteam struggled with the learning curve and

complexities of the system. Through explorative learning, the subteam experimented with the modules to gain an understanding of potentially useful features and limitations. The unguided learning process aided particularly when troubleshooting errors. After developing a large part of the mechanical design and the electronics in the smartlock, the team realized the need for an additional technology to manage the bike system at a higher level. Lacking a solid foundation in web technologies and other inventory management options, the team sought an alternative that would have a minimal learning curve and could be implemented within the time frame. The team built several “hello world” applications in a variety of backend and frontend languages including Ruby, SQL, Node.js, Meteor, and others. By gaining exposure to many options, the team was able to identify and commit to Meteor at an early stage rather than making the switch well into the development phase. Through building a comparatively simple web application in several languages, the experience helped identify major pain points and needs of working on a full scale web application.

Future Directions

The ultimate goal of this project was to develop and demonstrate a novel smartlock to pioneer the future of college campus bikeshares. If given additional time, the team would have pursued a small-scale implementation of the smartlock system on the University of Maryland, College Park campus. The Geolocation subteam discusses the minimally viable network that could serve a small group of students until the system was ready to expand campus wide. If successful, the technology would be ready to be

incorporated into a company and developed commercially. With the conclusion of this Gemstone research project, there is an incredible opportunity to continue and implement a smartlock-based bikeshare. The team demonstrated the university need by interviewing and working closely with students and has solved many of the technical and design challenges. The next step would be to deploy this technology on a campus and to continue to iterate and improve the design. Ultimately, this technology will serve as the foundation of a successful business.

To prepare for a possible implementation, the team designed a process to scalably deploy the system. The team planned out potential XBee module locations, developed the bike lock for easy manufacturing, and selected a cost-effective bike model for a bikeshare. These considerations taken into account, it is possible to purchase the necessary items and begin scaling a bikeshare from ten bikes or fewer to over several hundred on a large campus. For a future implementation there will likely need to be progress in troubleshooting the XBee system and maintaining a reliable system as unanticipated bugs appear as the system scales. The potential for next generation, sustainable, and profitable bikeshare technology developed in this project can only be realized through implementing the proposed full scale bikeshare system.

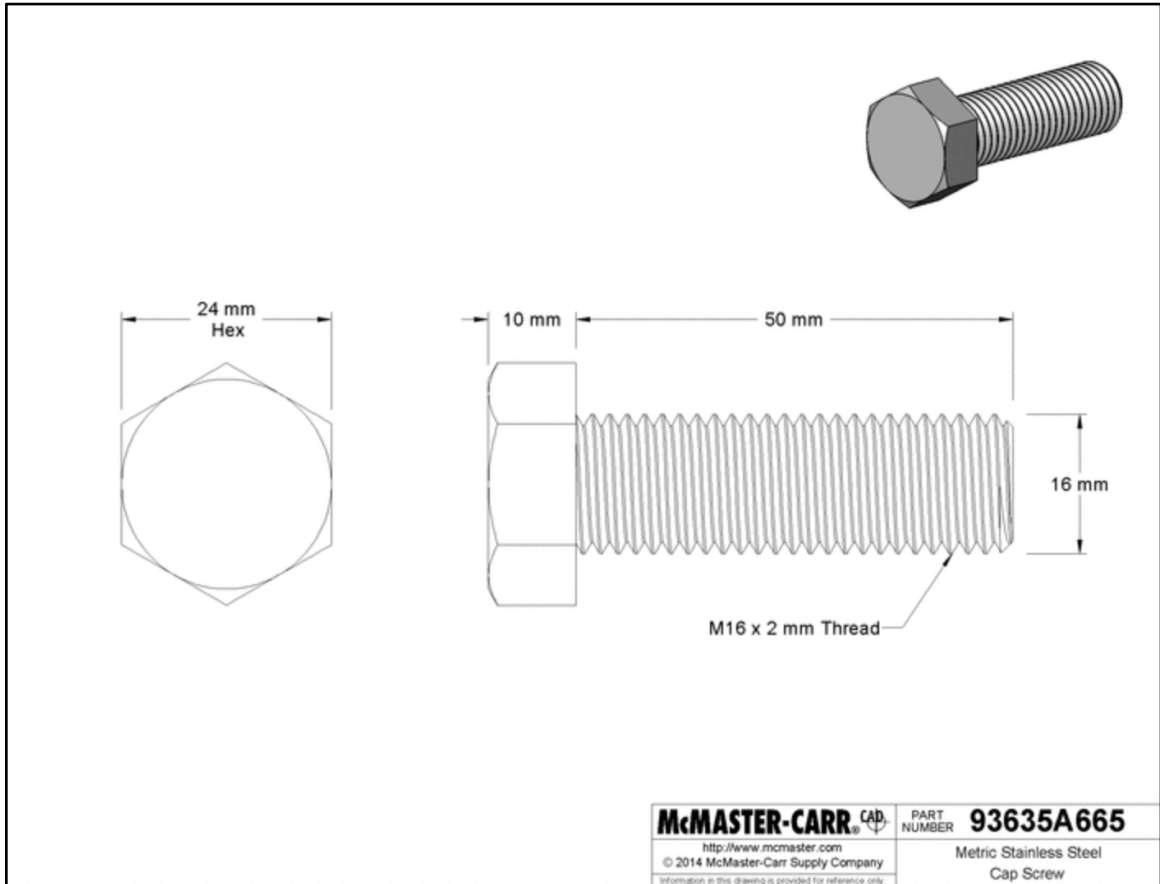
Additionally, the use of the received signal strength could potentially be combined with the known XBee parent nodes and the known bike rack locations in order to geolocate the bikes. Similar strategies have been implemented in various scalable wireless networks (Agarwal et al., 2001), so the use of the signal strength could

accurately locate the bikes and lead to great cost savings and improved geolocation capabilities in areas where GPS modules may not function as well.

Granted more time and financial support, an improved method for powering the electronic system on the smartlock should be further considered and researched. The current iteration uses a lithium-ion battery to power the on board electronics which requires replacing or recharging the battery when exhausted. Future work could focus on alternative methods for harvesting and generating electricity on board the smartlock to reduce maintenance. Research regarding solar panel solutions should be explored to determine the feasibility and cost of installing solar panel system to increase operating time of the smartlock. The battery life and performance for the smartlock is directly related to the proper operation of the smartlock and requires further development for a fully operational system.

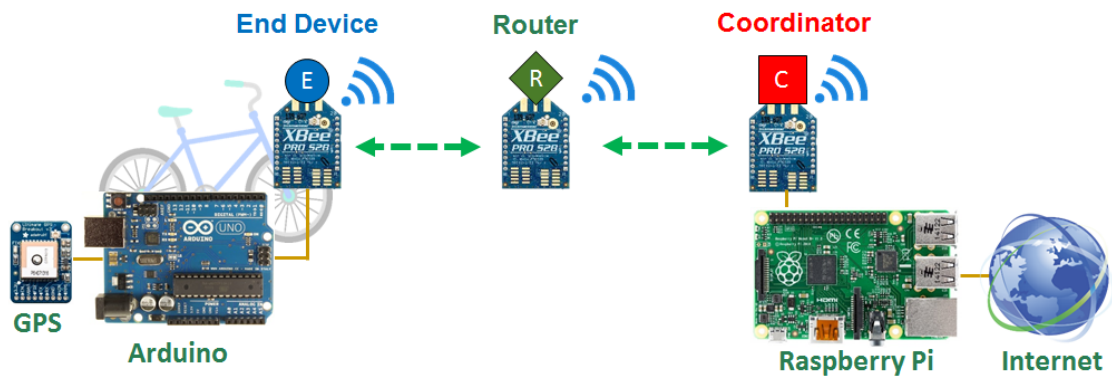
To best protect all the components housed inside, fully metal prototypes of the entire lock should be manufactured out of steel. These models should be failure tested to confirm the predicted FEA results of the design. There could be inconsistencies in the manufactured models that are not represented in an idealized computer model. As the electronics are finalized and the circuit profile is minimized, the shape and size of the electronics housing should be finalized and shaped out of sheet metal then folded and welded in the final form. The team did not previously manufacture the housing out of steel because the electronics profile had not been finalized.

Appendix A – Drawing

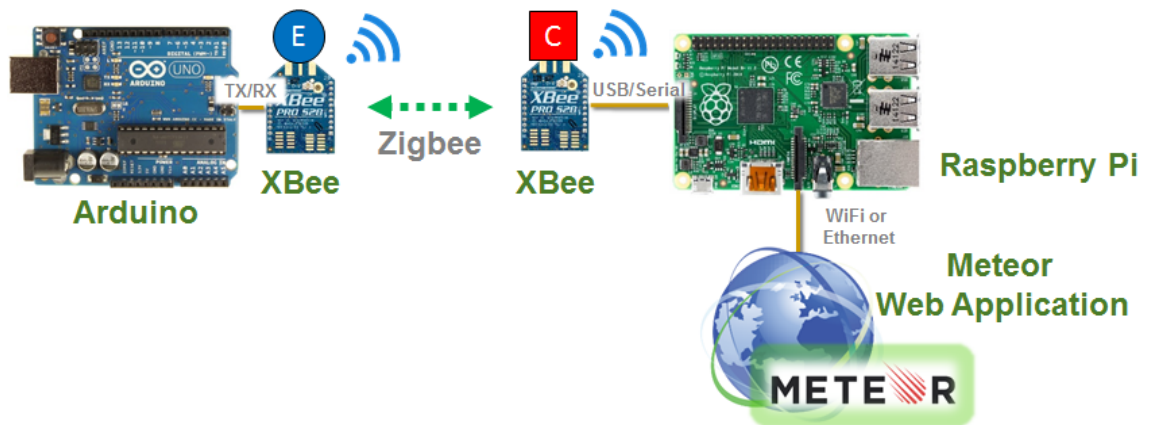


Appendix B – System Integration Diagrams

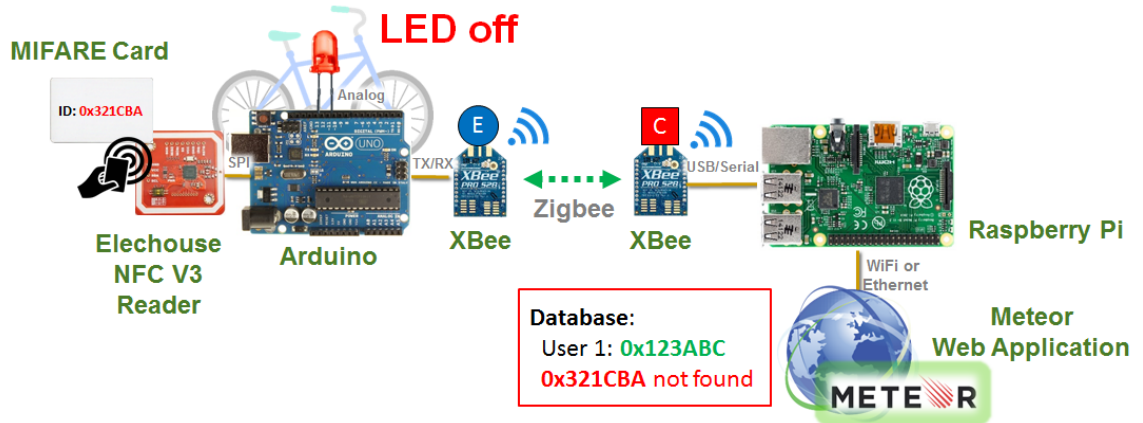
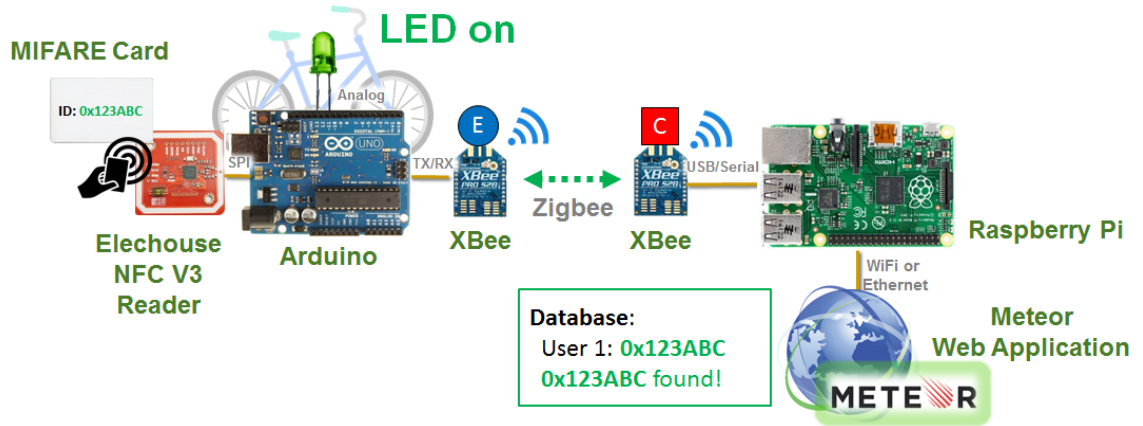
ZigBee Mesh Network Components



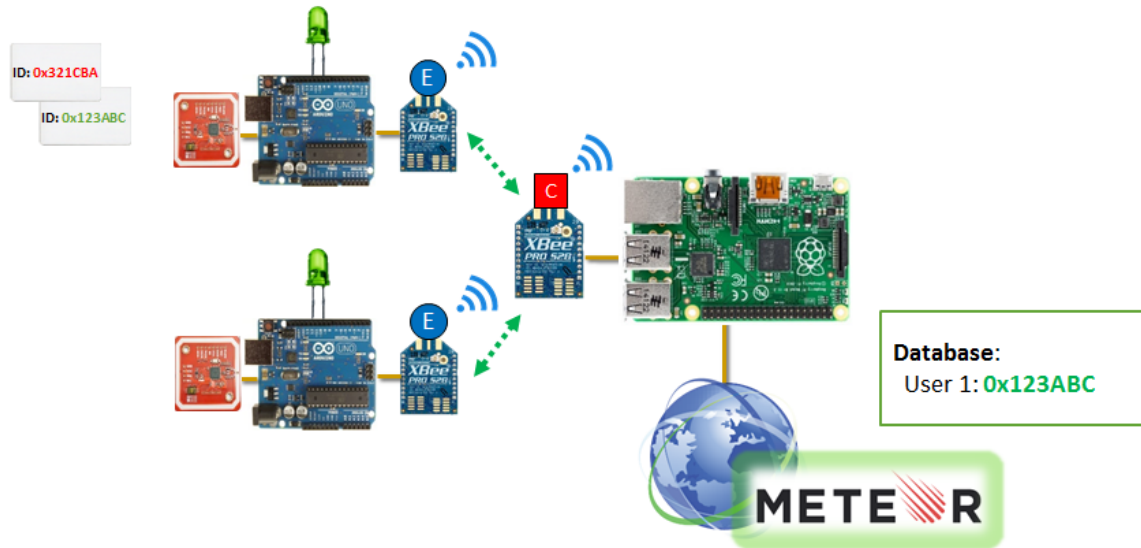
Demonstration 1: Arduino to Website through XBee



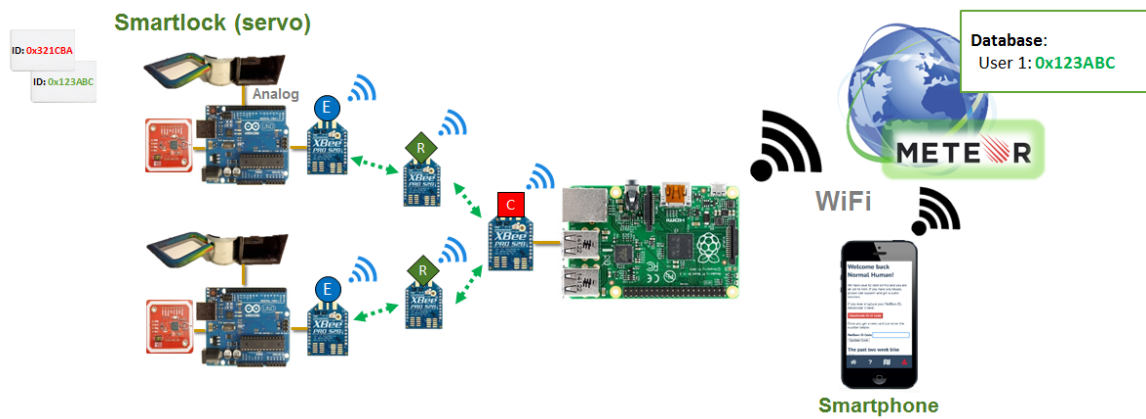
Demonstration 2: NFC User ID Wireless Authentication



Demonstration 3: Multiple Smartlock Prototype Units



Demonstration 4: Locking and Access



Appendix C - Focus Group Questions

- Do you have a bike on campus?
- If not, why don't you have a bike on campus?
- Would you use the system as we have explained it to you?
- Do you believe you know how to properly use a bike lock?
- Which lock locations do you think works the best? (Below the seat, on the seat tube, on the frame, other)
- Would you be opposed to using your student ID as the identification method for use?
- What RFID device would you prefer most? (A small sticker on a card or a plastic tag on a keyring)
- How much do you think Capital Bikeshare costs?
- Would you be willing to use the system for \$25-30 per semester?
- How would you solve the bike desert problem?
- Would you like bike information to be displayed on TerpNav, Google Maps, Bing maps, or another online map service?

References

- Agarwal, S., Agrawala, A., Banerjee, S., Kamel, K., Kochut, A., Kommareddy, C., Larsen, R., Nadeem, T., Shankar, A., Thakkar, P., Trinh, B., Youssef, A. & Youssef, M. (2001, December). *Rover technology: Enabling scalable location-aware computing* (UMIACS-TR 2001-89/CS-TR 4312) (Technical Report of the Computer Science Department). Retrieved from <https://www.cs.umd.edu/users/suman/pubs/cs-tr-4312.pdf>
- Alanazi, H., Zaidan, B., Zaidan, A., Jalab, H. A., Shabbir, M., & Al-Nabhani, Y. (2010). New comparative study between DES, 3DES and AES within nine factors. *Journal of Computing*, 2(3), 152-157.
- Allyn, M. (2013, May 29). Citi Bike 101. *Bicycling*. Retrieved from <http://www.bicycling.com/culture/advocacy/citi-bike-101>
- Andersson, T. (2014, June 8). *Bluetooth low energy and smartphones for proximity-based automatic door locks* (Bachelor's Dissertation). Retrieved from <https://www.diva-portal.org/smash/get/diva2:723899/FULLTEXT01.pdf>
- Balsas, C. J. L. (2003). Sustainable transportation planning on college campuses. *Transport Policy*, 10(1), 35-49. doi:[http://dx.doi.org/10.1016/S0967-070X\(02\)00028-8](http://dx.doi.org/10.1016/S0967-070X(02)00028-8)
- Battery University. (2013a). Advantages & limitations of the lithium-ion battery. Retrieved from http://batteryuniversity.com/learn/article/is_lithium_ion_the_ideal_battery

- Battery University. (2013b). How to prolong lithium-based batteries. Retrieved from http://batteryuniversity.com/learn/article/how_to_prolong_lithium_based_batteries
- Bitlock. (2015). Bitlock: Next generation keyless bike lock. Retrieved from <https://bitlock.co/>
- Bro, P., & Levy, S. C. (2013). *Battery hazards and accident prevention*. Springer Science & Business Media.
- Capital Bikeshare. (2013). 2013 Capital bikeshare member survey report. Retrieved from <http://www.capitalbikeshare.com/assets/pdf/CABI-2013SurveyReport.pdf>
- Cardoso, J., Hepp, M., & Lytras, M. D. (2008). *The semantic web: real-world applications from industry*. Springer Science & Business Media.
- Cervero, R., & Duncan, M. (2003). Walking, Bicycling, and Urban Landscapes: Evidence from the San Francisco Bay Area. *American Journal of Public Health*, 93(9), 1478-1483. doi: 10.2105/AJPH.93.9.1478
- Chavez, N. (2013, August 2). As Capital Bikeshare expands, service feels growing pains. *Washington Post*. Retrieved from https://www.washingtonpost.com/local/trafficandcommuting/as-capital-bikeshare-expands-in-popularity-and-size-it-runs-into-logistical-issues/2013/08/02/c2ee7f80-f925-11e2-afc1-c850c6ee5af8_story.html
- Chawla, V., & Ha, D. S. (2007). An overview of passive RFID. *Communications Magazine, IEEE*, 45(9), 11-17.

- Consumer Reports. (2016). Cell Phone Plan Comparison. Retrieved from <http://www.consumerreports.org/cro/news/2014/01/best-phone-plans-for-your-family-save-money/index.htm>
- Costa, C. (2015, August 2). Bitlock is the lockitron of bike locks, your smartphone is your key. *Gadget Review*. Retrieved from <http://www.gadgetreview.com/bitlock-is-the-lockitron-of-bike-locks>
- Curran, K., Millar, A., & McGarvey, C. (2012). Near Field Communication. *International Journal of Electrical and Computer Engineering*, 2(3), 371.
- Cycling About (2012, March 27). List of hub dynamo power supplies for USB devices. Retrieved from <http://www.cyclingabout.com/list-of-hub-dynamo-power-supplies-for-usb-devices>
- dell'Olio, L., Ibeas, A., & Moura, J. L. (2011). Implementing bike-sharing systems. *Institution of Civil Engineers*, 164(ME2), 89-101.
- DeMaio, P. (2009). Bike-sharing: History, impacts, models of provision, and future. *Journal of Public Transportation*, 12(4), 41-56.
- DeMaio, P., & Gifford, J. (2004). Will smart bikes succeed as public transportation in the United States?. *Journal of Public Transportation*, 7(2), 1-16.
- DIY Data. (2000). Drill bits - the different types explained. Retrieved from <http://www.diydata.com/tool/drillbits/drillbits.php>

- Domdouzis, K., Kumar, B., & Anumba, C. (2007). Radio-Frequency Identification (RFID) applications: A brief introduction. *Advanced Engineering Informatics*, 21(4), 350-355. doi: <http://dx.doi.org/10.1016/j.aei.2006.09.001>
- Erdmanczyk, S., Jr. (2014). XBee API mode tutorial using python and arduino. Retrieved from <http://serdmanczyk.github.io/XBeeAPI-PythonArduino-Tutorial/#why-xbee-api-mode-a-brief-review>
- Fleming, R. & Harris, L. (2010, October). *ZotWheels - UC Irvine's automated bikeshare program*. Case study presented at the Association for the Advancement of Sustainability in Higher Education Conference, Denver. Retrieved from https://www.newpartners.org/2012/docs/presentations/Friday/10%20-%2011.30am/Friday%203rd%20%20%20%20%20%20%20%20%2010%20-%2011.30pm%20%20%20Learning%20to%20Share/NP12_Harris.pdf
- Gaziulusoy, A. I. & Twomey, P. (2014, August). Emerging Approaches in Business Model Innovation Relevant to Sustainability and Low-carbon Transitions. *Visions and Pathway*. Retrieved from http://www.visionsandpathways.com/wp-content/uploads/2014/10/Gaziulusoy_Twomey_NewBusinessModels.pdf
- Gray, T. (2012, September 24). Science of theft: Freeze a bike lock with canned air, then smash it with a hammer. *Popular Science*. Retrieved from <http://www.popsci.com/diy/article/2012-08/gray-matter-how-science-helps-bike-thieves>

- Hoose, F. J. (2002). Mini Lathe Introduction. Retrieved from http://www.mini-lathe.com/Mini_lathe/Introduction/introduction.htm
- Howard, D. (2013, October 25). Smash Lab: Bike Locks Broken. *Bicycling*. Retrieved from <http://www.bicycling.com/bikes-gear/reviews/smash-lab-bike-locks-broken>
- Israsena, P. (2006, January). Securing ubiquitous and low-cost RFID using tiny encryption algorithm. In *Wireless Pervasive Computing, 2006 1st International Symposium on* (pp. 4-pp). IEEE.
- Johnson, S. D., Sidebottom, A., & Thorpe, A. (2008, June). *Bicycle Theft*. Available from <https://www.ncjrs.gov/App/Publications/abstract.aspx?ID=246484>
- Kaplan, S. (1974). *U.S. Patent No. 3,800,570*. Washington, DC: U.S. Patent and Trademark Office.
- Kazis, N. (2010, November 29). Theft and Vandalism Just Not a Problem For American Bike-Sharing. *Streetsblog NYC*. Retrieved from <http://www.streetsblog.org/2010/11/29/theft-and-vandalism-just-not-a-problem-for-american-bike-sharing/>
- Knospe, H., & Pohl, H. (2004). RFID security. *Information security technical report*, 9(4), 39-50.
- Krygowski, F., & Slanina, D. (2000). Generators for bicycle lighting. *Technical Journal of the IHPVA*, 49, 7-10.
- Kumar, U. P. (2013). Implementation Of A Security Protocol For Bluetooth And Wi-Fi. *International Journal of Network Security & Its Applications*, 5(3), 67.

- Kurtzleben, D. (2012, April 17). Bike sharing systems aren't trying to peddle for profit. *U.S. News*. Retrieved from <http://www.usnews.com/news/articles/2012/04/17/bike-sharing-systems-arent-trying-to-peddle-for-profit?page=2>
- Lang, H. (2016, March 24). College-Park bikeshare program to launch in May. *The Diamondback*. Retrieved from <http://www.dbknews.com/2016/03/24/college-park-bike-share-program-umd/>
- Langenfeld, M. E., Pujol, T. J., Moran, M. K., Barnes, J. T., Scheeter, R. L., & Jones, E. J. (2002). The metabolic cost of operating a bicycle generator light. *Ergonomics*, *45*(15), 1117-1120.
- Linnovate (2015). MEAN - Full-Stack JavaScript Using MongoDB, Express, AngularJS, and Node.js. Retrieved from <http://mean.io/#!>
- Masri, A. E., Khoukhi, L., & Gaiti, D. (2011). *A TDMA-based MAC protocol for wireless mesh networks using directional antennas*. Paper presented at the Conference on Communication Theory, Reliability, and Quality of Service, Budapest, Hungary.
- Maynard, M. (2013, July 30). What's Big With The College Kids? Bike Sharing. *Forbes*. Retrieved from <http://www.forbes.com/sites/michelinemaynard/2013/07/30/whats-big-with-the-college-kids-bike-sharing/>
- MD Transp Code § 21-101 (2013)

- Melin Tool Company (n.d.). Types and Characteristics of End Mills. Retrieved from <http://www.endmill.com/pages/training/types.html>
- Midgley, P. (2011). *Bicycle-Sharing Schemes: Enhancing Sustainable Mobility in Urban Areas*. Paper presented at the Commission on Sustainable Development, New York. <http://www.cleanairinstitute.org/cops/bd/file/tnm/13-bicycle-sharing.pdf>
- Modenesi, P. J., Apolinário, E. R., & Pereira, I. M. (2000). TIG welding with single-component fluxes. *Journal of Materials Processing Technology*, 99(1–3), 260–265. doi:[http://dx.doi.org/10.1016/S0924-0136\(99\)00435-5](http://dx.doi.org/10.1016/S0924-0136(99)00435-5)
- Mozer, D. (2016). International Bicycle Fund. Retrieved from <http://www.ibike.org/encouragement/freebike/>
- NXP. (2014). ZigBee PRO Stack User Guide. Retrieved from http://www.nxp.com/documents/user_manual/JN-UG-3048.pdf
- NXP Semiconductors. (2007). MF1 IC S50 functional specification. Retrieved from <https://www.adafruit.com/datasheets/S50.pdf>
- Ok, K., Aydin, M. N., Coskun, V., & Ozdenizci, B. (2011). *Exploring underlying values of NFC applications*. In *Third International Conference on Information and Financial Engineering IPEDR* (Vol. 12). Singapore: IACSIT Press.
- Preradovic, S., Balbin, I., Karmakar, N. C., & Swiegers, G. (2008). *A Novel Chipless RFID System Based on Planar Multiresonators for Barcode Replacement*. 2008 IEEE International Conference on RFID, pp. 289-296.

- Tate, B., & Hibbs, C. (2006). *Ruby on Rails: Up and Running: Up and Running*. " O'Reilly Media, Inc."
- Raghunathan, V., Kansal, A., Hsu, J., Friedman, J., & Srivastava, M. (2005, April). Design considerations for solar energy harvesting wireless embedded systems. In *Proceedings of the 4th international symposium on Information processing in sensor networks* (p. 64). IEEE Press.
- Rieback, M. R., Crispo, B., & Tanenbaum, A. S. (2006). The evolution of RFID security. *IEEE Pervasive Computing*(1), 62-69.
- RFID Journal. (2002). Frequently Asked Questions. Retrieved from <http://www.rfidjournal.com/site/faqs#Anchor-What-28258>
- Rzepecki, S. R. (2009). *U.S. Patent Application No. 12/655,400*. Washington, DC: United States Patent and Trademark Office.
- Schwartz, A. (2015, 2013-08-08). The return of spinlister: how to revive a dead sharing economy startup. *Fast Company*. Retrieved from <http://www.fastcoexist.com/1682784/the-return-of-spinlister-how-to-revive-a-dead-sharing-economy-startup>
- Shaheen, S., Guzman, S., & Zhang, H. (2010). Bikesharing in Europe, the Americas, and Asia: Past, Present, and Future. *Transportation Research Record: Journal of the Transportation Research Board*, 2143, 159-167.
doi:<http://dx.doi.org/10.3141/2143-20>

- Shu-qin, G., Jin-Hui, W., Lei, Z., Li-gang, H., & Wu-chen, W. (2008, April). A Low-power Active RFID Portable Reader System. In *Systems Conference, 2008 2nd Annual IEEE* (pp. 1-4). IEEE.
- Simpson, Chester. (2011). LM2576, LM3420,LP2951,LP2952. Retrieved from <http://www.ti.com/lit/an/snva557/snva557.pdf>
- Smithsonian Institution (2003). *Kryptonite lock company records*. Retrieved from <http://amhistory.si.edu/archives/AC0840.html>
- Sparkfun Electronics. (n.d. - a). RN42-XV Bluetooth module- PCB antenna. Retrieved from <http://cdn.sparkfun.com/datasheets/Wireless/Bluetooth/RN42XV.pdf>
- Sparkfun Electronics. (n.d. - b). XBee buying guide. Retrieved from https://www.sparkfun.com/pages/xbee_guide
- Stichting ART. (2015). *Het keurmerk*. Retrieved from <http://stichtingart.nl/keurmerk/>
- Strack, I. (2012). *Getting Started with Meteor.js JavaScript Framework*. Packt Publishing Ltd.
- Tang, D. (2010, September 22). Bike-sharing programs spin across U.S. campuses. *USA Today*. Retrieved from http://usatoday30.usatoday.com/news/education/2010-09-21-college-bike-sharing_N.htm
- Texas Instruments. (2016). TRF7960 multi-protocol fully integrated 13.6-MHz RFID reader/writer IC. Retrieved from <http://www.ti.com/product/trf7960>
- Thaku, A. (2012). What is ZigBee technology? *Engineers Garage*. Retrieved from <http://www.engineersgarage.com/articles/what-is-zigbee-technology>

- The Best Bike Lock. (2016). The best U-lock. *The Best Bike Lock*. Retrieved from <http://thebestbikelock.com/best-u-lock/>
- ThomasNet. (2015). More about CNC Machining. Retrieved from <http://www.thomasnet.com/about/cnc-machining-45330503.html>
- TiGr (2016a). *ART certification testing*. Retrieved from <https://tigrlock.com/lock-testing/>
- TiGr. (2016b). Strong and Light | TiGr Lock Bike Locks. Retrieved from <https://tigrlock.com/>
- Timalsina, S. K., Bhusal, R., & Moh, S. (2012, June). NFC and its application to mobile payment: Overview and comparison. In *Information Science and Digital Content Technology (ICIDT), 2012 8th International Conference on* (Vol. 1, pp. 203-206). IEEE.
- Toole Design Group. (2012). Bike Sharing in the United States: State of the Practice and Guide to Implementation, 1-68. Retrieved from Toole Design Group website: http://www.pedbikeinfo.org/pdf/Programs_Promote_bikeshareintheus.pdf
- Van Lierop, D., Grimsrud, M., & El-Geneidy, A. (2015). Breaking into bicycle theft: Insights from Montreal, Canada. *International Journal of Sustainable Transportation*, 9(7), 490-501.
- Velolock Germany (2015). Product Page. *Lock8*. Retrieved from <http://lock8.me/product/>
- Villard, P., Bour, C., Dallard, E., Lattard, D., De Pontcharra, J., Robert, G., & Roux, S. (2002, October). A low-voltage mixed-mode CMOS/SOI integrated circuit for

- 13.56 MHz RFID applications. In *SOI Conference, IEEE International 2002* (pp. 163-164). IEEE.
- Vogel, P., Greiser, T., & Mattfeld, D. C. (2011). Understanding Bike-Sharing Systems using Data Mining: Exploring Activity Patterns. *Procedia - Social and Behavioral Sciences, 20*, 514-523. doi:<http://dx.doi.org/10.1016/j.sbspro.2011.08.058>
- Want, R. (2006). An introduction to RFID technology. *IEEE Pervasive Computing, 5*, 25-33. doi: 10.1109/MPRV.2006.2
- Wang, Q., Ping, P., Zhao, X., Chu, G., Sun, J., & Chen, C. (2012). Thermal runaway caused fire and explosion of lithium ion battery. *Journal of power sources, 208*, 210-224.
- Washington University. (2015). Physics of solar cells. Retrieved from http://depts.washington.edu/cmditr/modules/opv/physics_of_solar_cells.html
- Wei, Y. H., Leng, Q., Han, S., Mok, A. K., Zhang, W., & Tomizuka, M. (2013, December). RT-Wi-Fi: Real-time high-speed communication protocol for wireless cyber-physical control applications. In *Real-Time Systems Symposium (RTSS), 2013 IEEE 34th* (pp. 140-149). IEEE.
- Weis, S. A. (2007). RFID (Radio Frequency Identification): Principles and applications. Retrieved from http://www.nfc-off.ch/uploads/4/5/1/2/45128343/rfid-article_mit_usa.pdf

- Welch, G. (2013, October 25). A High-Tech Search for the Best Anti-Theft Bike Lock. *Bicycling*. Retrieved from <http://www.bicycling.com/bikes-gear/reviews/a-high-tech-search-for-the-best-anti-theft-bike-lock>
- Welding Information Center. (2004). Welding Basics. Retrieved from http://www.weldinginfocenter.org/basics/ba_02.html
- Weman, K., Lindén, G., & Institute of Materials and Mining. (2006). *MIG Welding Guide*: CRC Press.
- Wolff-Mann, E. (2014). How ABUS tests its nearly unbreakable bike locks. *Thrillist*. Retrieved from <https://www.thrillist.com/gear/how-abus-tests-its-nearly-unbreakable-bike-locks>
- Xu, X., Gu, L., Wang, J., & Xing, G. (2010, March). Negotiate power and performance in the reality of RFID systems. In *PerCom* (pp. 88-97).
- Yu, K. Y., Yiu, S. M., & Hui, L. C. (2009, March). RFID forward secure authentication protocol: Flaw and solution. In *Complex, Intelligent and Software Intensive Systems, 2009. CISIS'09. International Conference on*(pp. 627-632). IEEE.
- Zane, M. S., & Zane, P. L. (1979). *U.S. Patent No. 4,155,231*. Washington, DC: U.S. Patent and Trademark Office.
- Zaruba, G. V., Basagni, S., & Chlamtac, I. (2001, June). Bluetrees-scatternet formation to enable Bluetooth-based ad hoc networks. In *Communications, 2001. ICC 2001. IEEE International Conference on* (Vol. 1, pp. 273-277). IEEE.