PotPi

by

Joseph Niemiller and Ryan Brad

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

© Copyright 2020 Joseph Niemiller, Ryan Brad

The author grants to the School of Information Technology permission
to reproduce and distribute copies of this document in whole or in part.

Joseph Niemiller 4/27/2020
Date

Ryan Brad 4/27/2020
Date

Bo Vykhovanyuk 4/27/2020
Insert Faculty Advisor's Name Here, Faculty Advisor Date

Technical Advisor: Bo Vykhovanyuk

University of Cincinnati
College of
Education, Criminal Justice, and Human Services
April 2020

TABLE OF CONTENTS

ABSTRACT 1

INTRODUCTION 2

1.1 Problem 2
2.1 Solution 2
2.1 Overview 2

DISCUSSION 3

2.1 Project Concept 3
2.2 Design Objectives 4
2.3 User Profile 5
2.3.1 Project Title 5
2.3.2 Potential Users 5
2.3.3 Software and Interface Experience 5
2.3.4 Task Experience 6
2.3.5 Frequency of Use 6
2.3.6 Key Interface Design Requirements 6
2.4 Use Case 7
2.5 Technical Discussion 7
2.5.1 Network Overview 8
2.5.2 Application Overview 8
2.5.3 Architecture Diagram 9
2.5.4 User Interface Overview 10
2.6 Testing 10
2.6.1 Methodology 11
2.6.2 Test Plan 13
2.6.3 Test Results 16
2.7 Budget 17
2.8 Gantt Chart 18
2.9 Problems Encountered 19
2.10 Future Project Plans 20
3.1 Lessons Learned 20
3.1.1 Skills Developed 21
4.1 Conclusion 20

APPENDIX 23

A.1 Spring presentation 23
A.2 PotPi Poster 24
B.1 References 20

LIST OF ILLUSTRATIONS

FIGURES

Figure 1. Use Case Diagram 7
Figure 2. Architecture Diagram 9
Figure 3. Budget Diagram 17
Figure 4. Gantt Chart 18

TABLES

Table 1 UAT Scope 11
Table 2 UAT Assumptions of Constraints 12
Table 3 UAT Constraints 12
Table 4 UAT Risks 13
Table 5 UAT Team Roles and Responsibilities 14
Table 6 UAT Entry Criteria 14
Table 7 UAT Requirements-Bases Test Cases 15
Table 8 UAT Test Results 16

Abstract

The Internet of Things is an emerging market that is expected to grow to 43 billion devices by 2023.[1] This exponential growth comes with security concerns, as 80% of devices including cloud and mobile applications fail to require complex passwords, and 70% of devices did not encrypt traffic to and from its connected network.[2] Identifying the security risks can be difficult, since every network has a different combination of devices. The use of PotPi's testing ground for emerging IoT devices provides the necessary foresight to combat the increasing number of security threats in homes and workplaces worldwide. PotPi provides this service to a growing industry and creates targeted objectives to reduce security concerns by allowing the community to discover vulnerabilities and be rewarded for doing so.

Creating a game-like platform is not a new concept. In fact, there are many companies that have adopted this concept due to its increased user interaction and ability to relate to the audience it is trying to reach. Giving the community a way to show their progress and accomplishments gives companies data on what is working and what needs to be changed. More incentive to complete objectives leads to more solutions and a better market.

1.1 Problem

Within the IoT device industry, manufacturers have very little monetary incentive to increase security in their devices. Average consumers who do not know IoT technologies inside and out are risking their personal information by placing flawed devices on their networks which house personally identifiable and other sensitive information. According to Hewlett Packard's 2014 report, "70% of 26 billion current IoT devices are vulnerable to an attack. It is expected that the number of IoT Devices will grow to 43 Billion by 2023." [3]

1.2 Solution

PotPi helps to improve security posture in the industry by bridging the gap between their products and consumer security standards. PotPi provides a boost in manpower and means for manufacturers to test devices while gaining valuable insight regarding consumer perspectives. Being a member of the PotPi community means contribution towards testing and resolution of issues found in IoT devices, while progressing through a game like platform to showcase your skills and major achievements within the application and community.

1.3 Overview

With our Senior Design project PotPi, our team hopes to provide an education platform to those learning about information security concepts, while opening a direct avenue for collaborative IoT product testing and planning with end users. While other platforms offer a more in-depth approach, PotPi aims to create a sense of trust in the IoT device

industry by setting a standard for security expectations in a simpler manner. The remainder of this report covers with detail how the project was completed. The report includes the following sections: Project Concept, Design Objectives, User Profile, Technology Discussion, budget, problems encountered, and future recommendations for PotPi.

2.1 Project Concept

PotPi provides an incentive for manufacturers to take a hard look at the security of their IoT devices and how they may be exposing sensitive consumer data. The product bridges the gap between IoT manufacturers and consumers by providing a space where security vulnerabilities can be identified, documented, and prevented in future updates and before products are released. PotPi provides a platform for IoT manufacturers to host their products for end users, consumers, and security professionals alike to test the security controls on their products, attaining achievements and even monetary rewards in some cases for identifying security flaws within IoT products. PotPi is complete with a knowledge base full of discussion and known vulnerabilities which provide both direction and motivation for users to uncover new security flaws on their own, all while progressing through a game like achievement system and netting some serious rewards!

The idea for PotPi originally stemmed from a “Plug and Play Honeypot” product idea aimed for small to medium sized businesses that may not have the vast resources required to put towards a proactive security posture within their organization. This original idea mutated along the way into more of a security vulnerability identification /

bounty program that allows consumers to learn and achieve rewards while facilitating a positive influence on security posture in the IoT device industry.

2.2 Design Objectives

When planning for PotPi as a product some of the goals we identified are as follows:

- Bettering the community through education and opportunity.
- Creating a sense of partnership and trust between consumer and manufacturer.
- Creating a more secure standard for the IoT device industry.
- Providing a public facing opportunity for manufacturers to test their products.
- Achieving the above utilizing a fun and interactive, game-like platform.
- Offering monetary rewards for finding major security flaws.

Some of the original goals pertaining to the “Plug and Play Honeypot” project approach included:

- Providing a quick and easy plug and play honeypot deployment.
- Providing real time monitoring and alerting in the event the honeypot was compromised.
- Facilitating a proactive approach to Information Security at a low cost, low resource, low labor-intensive bottom line for small business.

These goals did not fit within the scope and were thrown out as we shifted our project’s direction towards a collaborative effort towards improving security posture in the IoT device industry.

2.3 User Profile

2.3.1 Project Title

PotPi

2.3.2 Potential Users

Potential users for PotPi can be separated into three groups: IoT Manufacturers, Administrators, Normal Users. IoT Manufacturers will have a direct communication line to the Administrators since they will be supplying information about their devices. The Manufacturers will have minimal access to the Administrator console which has permissions to add and remove device tests. Completion data is sent to the Administrator, who will manage PotPi containers, goals, rewards and other day-to-day issues within scope. Administrators will directly communicate the progress to the Manufacturers. This creates a personal connection between the Manufacturer and PotPi, which builds upon the goal of bridging the gap between the Manufacturer and the consumer. Normal users are tasked with assorted goals depending on the requests of the other two user types. Users have non-administrative access to PotPi containers, as well as forums, profiles, knowledgebases, and other community portals.

2.3.3 Software and Interface Experience

The user facing interface has a modern design that is pleasing to the eyes, smooth transitions and easy navigation. Multiple application pages are separated into different sections depending on what type of challenge is presented. Familiar login system like that of other forum website models.

2.3.4 Task Experience

Working on a challenge can vary depending on the type. The original test application shows two standard widgets, one of which shows the local IP address of the Raspberry Pi test device. The second widget checks for changes on the Raspberry Pi within a specific file designated prior to beginning the challenge. The use of the first widget will not be necessary when the service becomes public since the IP will not be changing constantly. Plans for the second widget include:

- Expanding requirements for objectives. (Not just a simple file change)
- Addition of more sub-routines for user to complete.
- In-browser view of device.

2.3.5 Frequency of Use

Future implementations include a daily challenge or a forum challenge. Devoted members may use this service everyday depending on the availability of a task. Normal users may take part in using PotPi weekly. With the full support of a manufacturer, a constant flow of support would be needed to fulfill the needs of the customer.

2.3.6 Key Interface Design Requirements

- A key interface design requirement is the login functionality must be personable, generally for the average user.
- Quick and intuitive way to access trials and tasks.
- A forum style knowledge base where users can learn about challenges or Cybersecurity as a whole.

2.4 Use Case Diagram

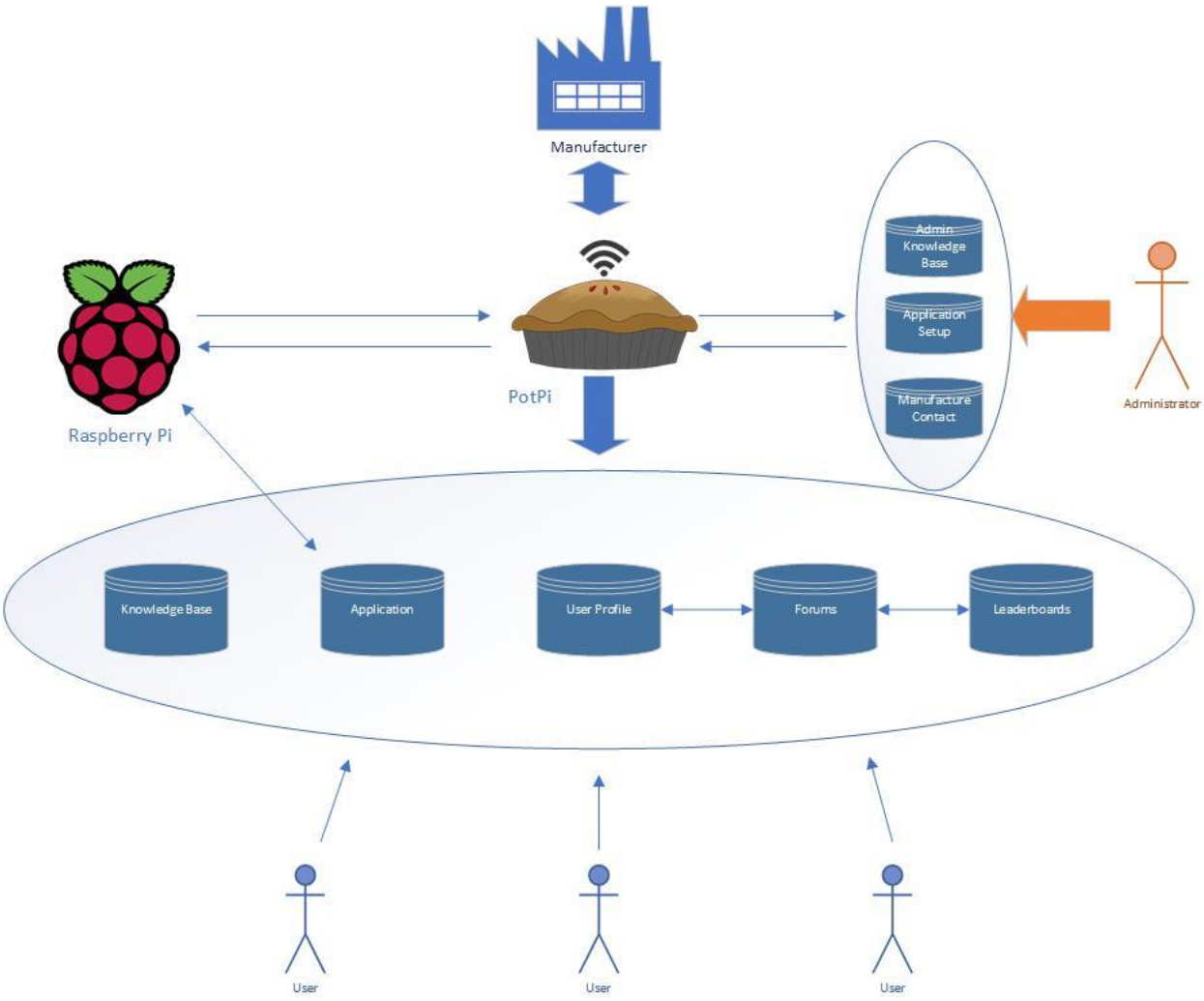


Figure 1: *The Use Case diagram* shows the basic architecture and workflow of the different sections of PotPi.

2.5 Technology Discussion

We wanted to create a clean, user friendly interface for the PotPi application that motivates users to tinker with various IoT devices and security technologies. To

accomplish this, we used HTML and CSS to build the front end of our application that users interact with. We utilized JavaScript functionality for the back end in order to bring user data and input to interface with the PotPi application. Additionally, one of the most critical technical aspects of our project was providing the ability for users to login, create specific profiles, and track their achievements and progression within the PotPi application. To do this we needed to implement database technologies to track specific user by user information and actions. We accomplished this objective by utilizing Firebase's real time database features. Lastly, to provide a demonstration of the product, we utilized a locally hosted Raspberry Pi in order to provide examples and demonstrations of PotPi's security vulnerability testing, tracking, and documenting features pertaining to IoT devices.

2.5.1 Network Overview

The project currently relies on a locally hosted Raspberry Pi 4 and a Windows 10 Machine. Original setup required SSH access, however now we use VNC viewer to access the Raspberry Pi machine.

2.5.2 Application Overview

On the Application layer of this project, the backend runs off JavaScript, while running HTML, CSS, Handlebars and Bootstrap for the front end. JavaScript made the most sense to us because of how fast it can work when written properly, and the compatibility with the other systems we are using. The two main widgets of the application involve a

Test function to make sure the Raspberry pi has not changed IP addresses. The other widget looks at a specific file on the Raspberry Pi and compares the last 5 changes.

2.5.3 Architecture Diagram

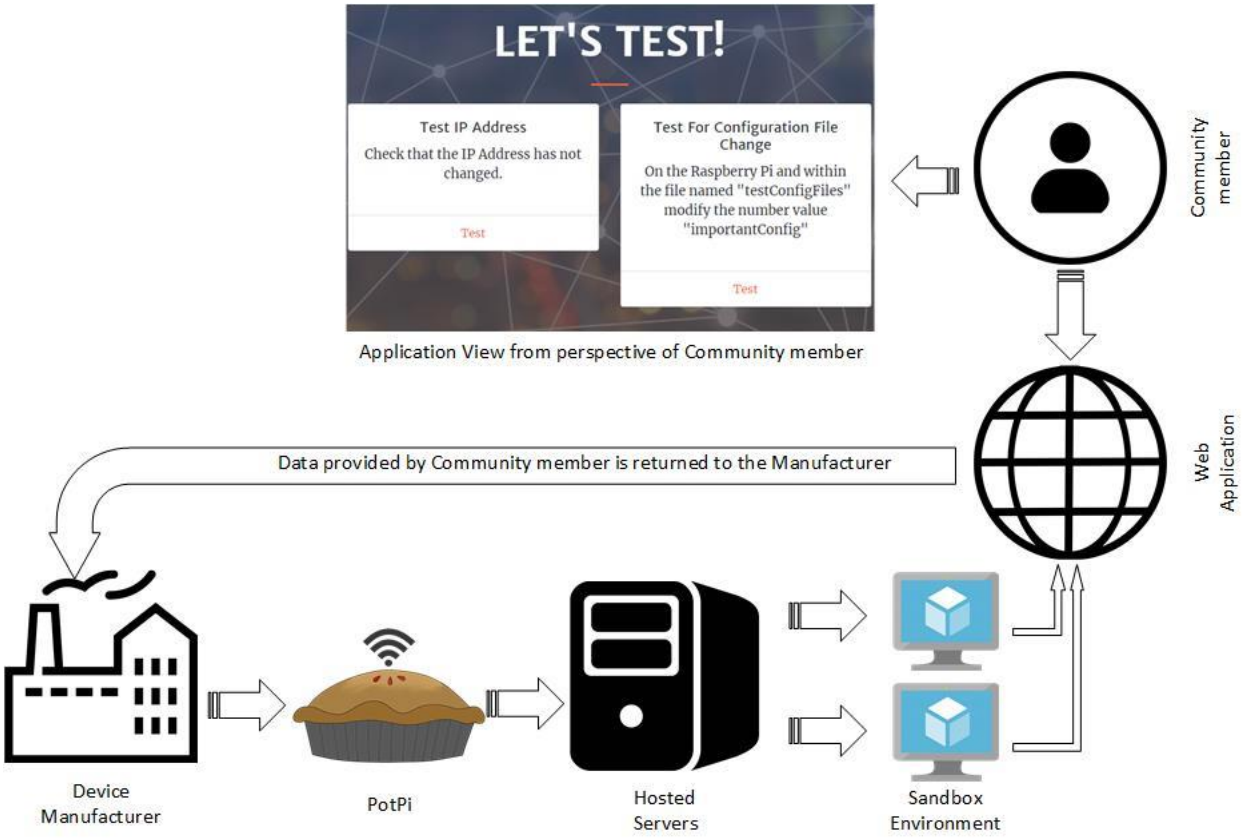


Figure 2: The Architecture Diagram shows the architecture of PotPi as well as a quick look at the widgets available.

Our architecture is moderately simplistic, as shown above. The purpose of this simplicity is to make the experience less overwhelming to new users, as well as make development more streamlined. Manufacturers contact PotPi, we host what needs to be tested, and from there community members access the sandbox through PotPi's hosted servers.

2.5.4 User Interface Overview

Staying on the topic of simplistic design, the interface is intended to direct the user's eyes to important parts of the page. We simply wanted to allow the user to get wherever they needed to navigate to within three mouse clicks. With the current setup, we have accomplished this and will continue to follow that guideline to keep the user experience fluid. The navigation bar contains relevant information and directs you to the other application pages.

2.6 Testing

2.6.1 Methodology

We were able to get 3 IT students and 1 Business student from different Universities. Getting a wide range of experience is needed due to the nature of this project. Since this is designed so that someone with minimal experience can pick this application up, we needed to make sure that this goal was achievable. Testing multiple browsers is important because we want to be sure that we can reach the widest demographic. Testing involved: making an account, testing their new login account, attempting to breach the Raspberry Pi given the IP address, and editing a single text file on the Raspberry Pi.

2.6.2 Test Plans

1. UAT Scope (In Scope – Out of Scope)	
UAT – In Scope	**UAT – Out of Scope**
In Scope *List of features that are tested* - **Account Creation services** - **Login services** - **Granting Token for breach attempt** - **Using token to begin breach attempt** - **Simple fail task scenario** - **Simple pass task scenario** - **Account points**	**Out of Scope** *List of features that are not tested* - **Creation of breach attempts** - **Creation of Corporate accounts** - **Creation of Sandbox** - **Corporate environment** - **Community task creation**

Table 1: *UAT Scope illustration shows the scope of PotPi and the list of items that are out of scope.*

2. UAT Assumptions of Constraints

UAT Assumptions
Assumption *List of UAT assumptions* - **Test environment:** The test cases will be conducted by Joseph Niemiller and Ryan Brad in the Lindner College of business testing facility. Hardware and software are provided by PotPi representatives. Screen capture software will be used to view user mouse-application interactions. - **Test documentation:** All UAT test cases will be held in a shared folder to be viewed only by PotPi staff. - **Error reporting:** Errors, issues and flaws will be reported by both the user and PotPi staff.

Table 2: *UAT Assumptions of Constraints shows the list of assumptions made regarding the User Testing.*

3. UAT Constraints

Constraint *List of UAT constraints* - **Time Frames:** Test results will be gathered and provided by April 15th 2020 - **Resources:** - Testers: Total of 15 sessions of 30 minutes based on user experience - Hardware: Lenovo Workstation (Desktop), Lenovo Workstation (Laptop) - Tested OS: Mac OS, Windows, Linux - Browsers tested: Chrome, Mozilla Firefox, Internet Explorer, Edge, Safari, Opera

Table 3: *UAT Constraints shows the list of constraints within the testing of PotPi, giving insight on the things tested.*

4. UAT Risks

Description	Probability High | Med | Low	Impact High | Med| Low	Mitigation
Improper test training	Med	Low	Ongoing testing
Incomplete testing due to time	Med	Med	Realistic time management
Testers are not sure how to report bugs	Low	Med	Bug reporting viewed by user and PotPi Staff
UAT test failure	Low	High	Development completion before testing

Table 4: *UAT Risks illustration shows the risk factors associated with each testing scenario.*

5. UAT Team Roles and Responsibilities

Name	Roles	Responsibility
Joseph Niemiller	**Project Manager | Tester**	**Testing on:** **Windows: Chrome, Mozilla, Edge**
Ryan Brad	**Project Worker | Tester**	**Testing on:** **Linux: Chrome, Mozilla** **Mac OS: Safari, Chrome**

Table 5: *UAT Team Roles and Responsibilities table shows each member of the PotPi team, and their respective roles within the testing process.*

6. UAT Entry Criteria

ID	Criteria
6.1	**Development of target services are completed and operational**
6.2	**Testing integration has been complete**
6.3	**Minimal bugs found | Bugs that have been found that do not interrupt testing are being patched**
6.4	**UAT Test Environment prepared including: Hardware, Software, Location**
6.5	**Brief testers and explain what the end goal is.**
6.6	**UAT test plan is available to testers**

Table 6: *UAT Entry Criteria illustration shows the testing plan and criteria involved in the user testing process.*

7. UAT Requirements-Bases Test Cases	
ID	**Criteria**
7.1	**Testing Login:** **- Please visit PotPi.org/Testing** **- Login with the following credentials** **Tester1** **TesterPassword** **- On the dashboard, view "Community"** **o Open Community** **o Find the test environment named "UAT Test"** **- Expected Result** **o Near-Blank screen while environment is spooled** **o Message overlay stating objectives**
7.2	**Testing** **- Follow First objective** **o View designated file system** **o Edit a YAML file** **o Save YAML file to file system** **o Open command prompt** **§ Netstat** **§ Take note of device names** **- Follow Secondary objectives** **o Use IOT IP address to get into user portal** **o Once on user portal (changes based on device) view settings** **o Verify you can alter device settings**

Table 7: *The Requirements-Bases Test Cases illustration shows the two different tests performed by testers, and the involved tasks within each one.*

2.6.3 Test Results

8. UAT Test Results				
ID	**Test Cases**	**Pass/Fail**	**Tested By**	**Date Tested**
8.1	**Testing 6.1**	**Pass**	**Anonymous_NKU**	**3/23**
8.2	**Testing 6.1**	**Pass**	**Anonymous_UD**	**3/23**
8.3	**Testing 6.2**	**Pass**	**Anonymous_UD2**	**3/30**
8.4	**Testing 6.2**	**Pass**	**Anonymous_UC**	**3/31**

Table 8: *The Test Results illustration shows the results per tester, along with a pass or fail, who the tester is, and the date tested.*

We were very excited to see that all our testers had completed their attempts at our product. The application is designed with new users in mind, and the fact that the amount of time taken to test between the users was minimal, confirmed that this could be a successful product.

2.7 Budget

PotPi		Project Budget	
Hardware		Team Cost	Enterprise Cost
Compute:			
Raspberry Pi 4		$75	$75
Items:			
Ethernet Cable		$11	
Keyboard		$10	
Mouse		$8	
	Totals:	$104	$75
Hosting			
Service:			
AWS			$.15/hr
Bluehost Hosting			$13.95/Month
Self-Hosted		On-Hand (Local)	
	Totals:	$0	$121.95/Month
Our Total Cost:		$104	
Enterprise Cost (Minus Salary):			$121.95

Figure 3: *Project Budget* shows the budget for the PotPi project. It provides an overview of real market costs for our project team and enterprise PotPi users.

The main reason why the cost is so much cheaper on PotPi's end as opposed to the Enterprise level cost, is because of the number of man-hours required to complete a project like this. Simply put, salaries and hardware are the most expensive pieces. Cutting out the middleman for businesses means that they do not need to spend the money to host their own product or testing ground. On top of this, a whole division would be needed to monitor and maintain the web application, leading to a higher associated cost. These are the main reasons companies do not worry about device security, and PotPi is cutting into that market by giving a sustainable, cheaper way to improve security.

2.8 Gantt Chart

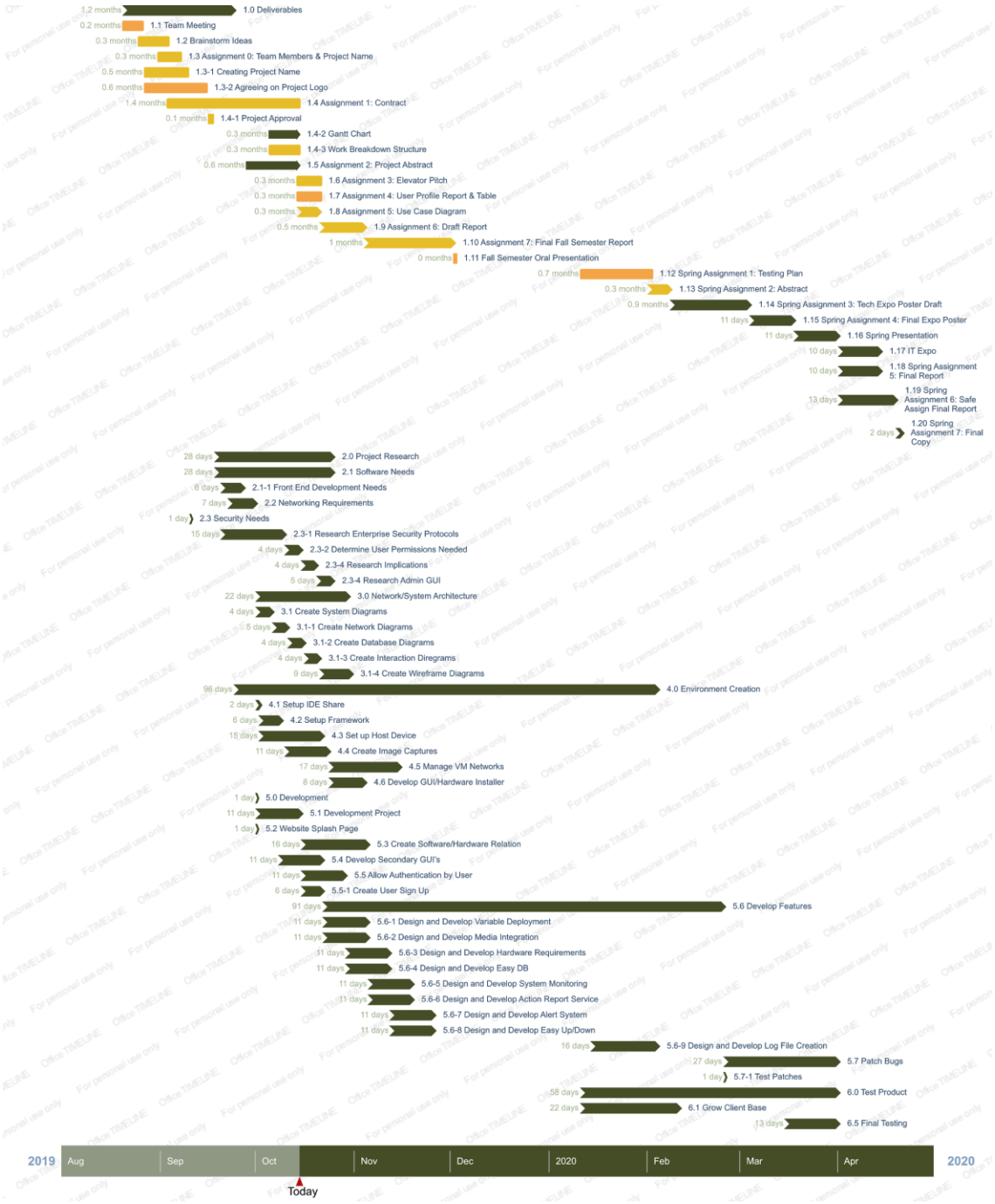


Figure 4: The figure shown above is the project schedule with assignment due dates and project milestones.

2.9 Problems Encountered

- New to programming

  One of the things that presented a real challenge for us during the creation of PotPi is the learning curve that comes with programming. Neither of us had much prior experience with app development, and learning the necessary languages was challenging. After spending the past few years studying the subjects, understanding the technical aspects of networking and cyber security is well within our wheelhouse. Taking our vision for PotPi to applying actual development techniques and seeing the product through to completion put a lot of new learning onto our plates.

- User Databases

  Another issue that we had involved the User Database system. Using Firebase, we were able to create a proper login server. However, the issue arose when we attempted to connect relevant user data to a user login profile. What we did not know at the time, is that the Google service Firestore, could have solved these issues. Again, neither of us had much Database experience, and this caused a large amount of tinkering to make this work.

- Local vs. Hosted Web App

  Something that came up early in our planning, was the discrepancy between running a hosted Web Application, or keeping it locally hosted. The decision came to keep the app locally hosted, because it kept the

scope of the project more manageable, and kept the price down as opposed to AWS or Azure pricing.

2.10 Future Project Plans

In order to further this project and get PotPi in a position where it is marketable to manufacturers, some features need to be added. Below is a list of planned additions:

- Secure, online payment method for a (pay-per-attempt) type web format. Not only for the common user, but also for manufacturer payment to PotPi.
- Readable and editable knowledge base section created both by administrators and active members.
- Fields that will allow the user profile page to be customizable.
- More pages for applications, and a section to organize multiple trials.

3.1 Lessons Learned

In terms of lessons learned, there are a slew of things we learned throughout our senior design coursework that would be suitable to list. Firstly, recruiting one more team member would have gone a long way towards ensuring we had the technical development experience critical to taking our application the extra mile. Having a clearer vision for a project which complimented our strengths and talents from the beginning would have been huge for the planning stages of our project. Additionally, the next time either of us are working towards a project with an application that interacts with a database element, I know we will have learned from this experience by better mapping out the plans for different aspects of relational databases required to support the application

3.1.1 Skills Developed

Project management skills learned from previous classes such as Human Computer Interaction at UC were crucial to keeping this project progressing, staying on time and within our project's scope and delivering on the requirements we identified for the PotPi application. Additionally there were a number of technical skills we had to develop along the way to drive the front and back ends of our application This included JavaScript, Handlebars, and Bootstrap to interface between HTML elements on our web application and other backend services like our database application. Lastly database design for the PotPi application was a large undertaking for our skill sets. Learning how to use database technologies to track user accounts and progress within our application was a major skill developed when building the PotPi application. Since changing the direction of our project and planning in fall semester, we were able to get up to speed on the technologies above and build a demo-ready web application capable of testing IoT device security and documenting progress towards challenges completed by specific PotPi users.

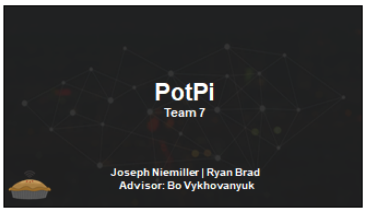
4.1 Conclusion

The two semesters that we worked to make this project a reality has taught both of us a lot about what it means to develop something from the ground up. The small technical things that seem insignificant can create a lot of issues for you down the road if not taken seriously or could end up being a roadblock. This is what happened regarding the use of a real-time database for us. Looking at how to operate new services like Google Firebase was daunting, but that was nothing compared to learning everything needed to

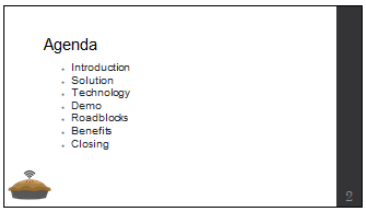
create an application using JavaScript. The original concept of creating a quickly deploying Honeypot solution was difficult to create a clear path for, and even more difficult attempting to explain why it was even needed. PotPi however did not cause this issue. We were able to mark down exactly what needed to be done and how, giving us a better way to work and explain progress and workflow to our class. Once we sought out our advisor for help, we decided to make that move towards working on PotPi; Otherwise we would have had an incomplete project due to the original scope. This project was a wonderful learning experience, and we suggest that anyone going through the IT program attempt to reach out for a difficult project and go for it. You will learn a lot and have wonderful support from UC faculty.

Appendix A Presentation Information

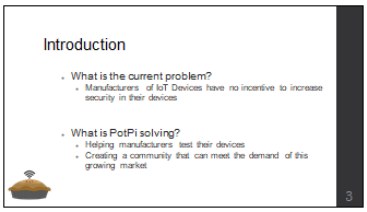
A.1 Spring presentation



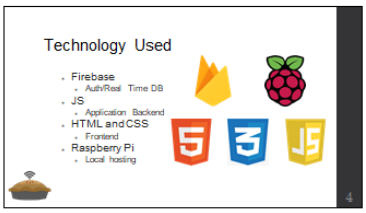
1



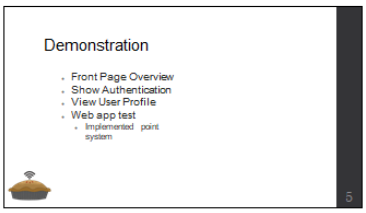
2



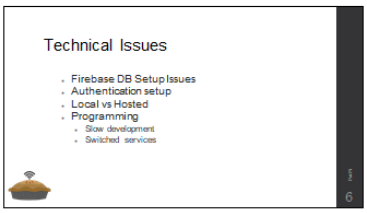
3



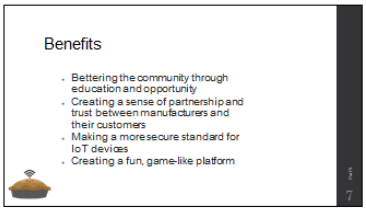
4



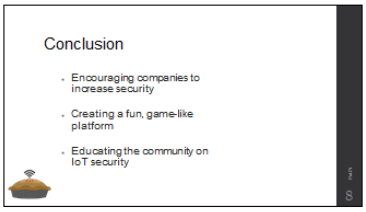
5



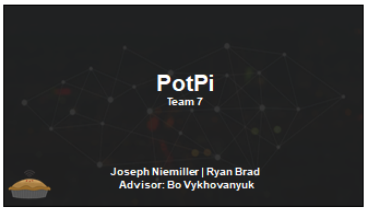
6



7

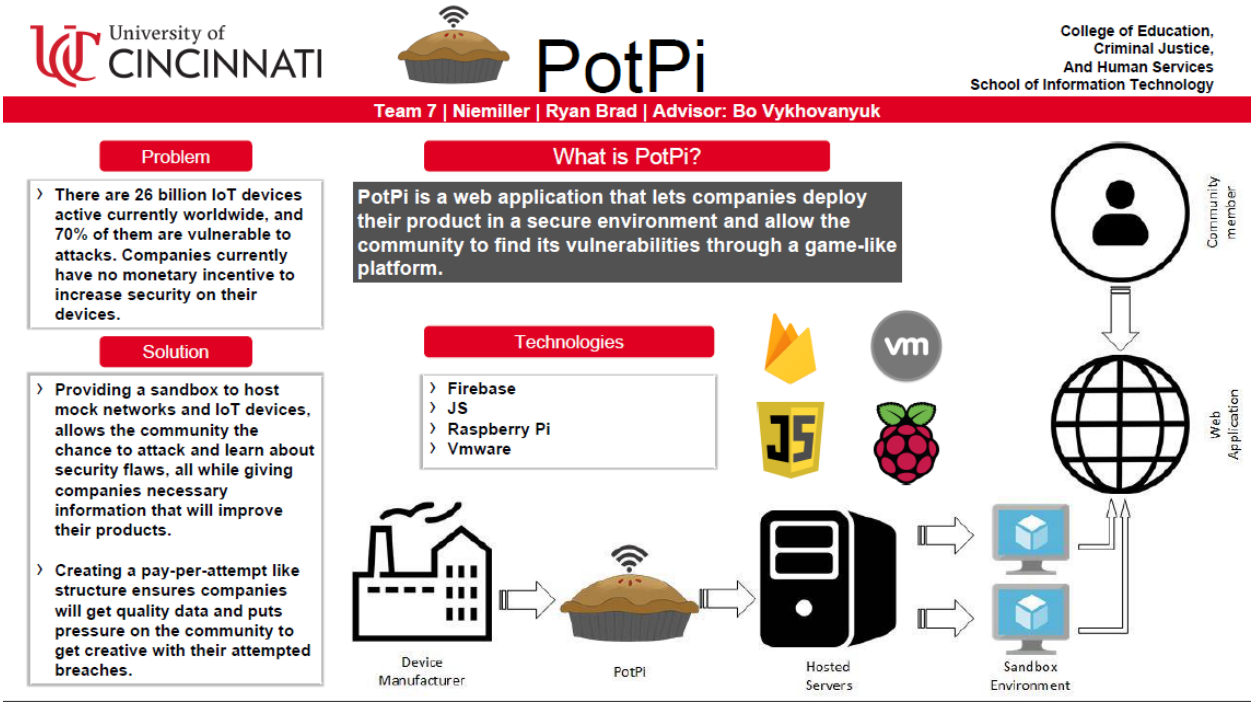


8



9

A.2 PotPi Poster



Appendix B References

[1] Dahlqvist, Fredrik, et al. “Growing Opportunities in the Internet of Things.” *McKinsey & Company*, www.mckinsey.com/industries/private-equity-and-principal-investors/our-insights/growing-opportunities-in-the-internet-of-things.

[2][3] Rawlinson, Kristi. “HP Study Finds Alarming Vulnerabilities with Internet of Things (IoT) Home Security Systems.” *HP News - HP Study Finds Alarming Vulnerabilities with Internet of Things (IoT) Home Security Systems*, 10 Feb. 2015, www8.hp.com/us/en/hp-news/press-release.html?id=1909050.