HomeWatch

by

Stefan Salzmann, Mark Rankey, Josh Bosse

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

Stefan Salzmann April 28, 2020
Stefan Salzmann Date

Mark Rankey April 28, 2020
Mark Rankey Date

Josh Bosse April 28, 2020
Josh Bosse Date

Ryan Moore April 28, 2020
Ryan Moore Date

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

Table of Contents

Name		Page
Abstract……		...1
1.	**Introduction…**……	2
1.1	Problem……	2
1.2	Project Description……	..2
1.3	Solution……	... 3
1.4	Project Objective……	4
1.5	Overview……	...4
2.	**Discussion**……	..5
2.1	Project Concept……	..5
2.2	Design Objectives……	..6
2.3	Methodology and Goals……	6
2.4	User Profile……	...7
2.5	Use Case Diagram……	..10
2.7	Testing Plan and Report……	..14
2.8	Gantt Chart……	...23
2.7	Problems Encountered……	28
3.	**Conclusion**	30
4.	**References**	32
4.1	Sources	32

List of Illustrations

Tables

No.		Page
Table 1.	Unit Test	15
Table 2.	Budget	18
Table 3.	Gantt Chart	19

Figures

No.		Page
Figure 1.	User Profile	7
Figure 2.	Use Case Diagram	10
Figure 3.	Technical Diagram	12
Figure 4.	Test Text File	13

Acronyms and Abbreviations

IT	Information Technology
UI	User Interface
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
JS	JavaScript
SQL	Structured Query Language

Abstract

According to a study in 2018 from PSYCOM, it was concluded that 50% of teenagers feel they are addicted to their smartphones, 60% of adults say that they believe that their children are having issues with time on their phone and, on average, 60% adults and children state they immediately respond to notifications. HomeWatch is a web-based application that would give everyone ease of mind when it comes to overlogging by tracking the time used on certain websites and applications without limiting time or function. HomeWatch is meant to be a safeguard for expectations set by parents so that children aren't abusing time on their device. With social media, and easy access apps on smartphones, it is easier to spend more time on the phone than it is to do productive work, and HomeWatch is there to make sure that work and play are budgeted effectively.

1. Introduction

1.1 Problem

It was found, in a recent study from PSYCOM, that teenagers, and adults are both experiencing issues when it comes to their device: 50% of teens say they feel that they are addicted to their phones, while 60% of parents believe that their children have issues with overlogging. Both adults and children surveyed portrayed addictive tendencies when it came to their device.

In a related study by Common Sense Media, the issue of overlogging has affected family life in a disruptive manner: a third of both parents and teenagers say that they argue on a near daily basis on their phone/device usage. Observed also was that almost 80% of parents surveyed that their children will use their devices while with family, hindering their family time together. It was also found that technology shouldn't replace human interaction, as it has a detrimental effect on the development with empathy in children.

1.2 Problem Description

The nature of this project stems from the addictive tendencies when it comes to our devices as well as the relationship it has to family life. With the issue of daily arguing among a third of parents surveyed in the aforementioned study, mainly because of device use during family time, it is important to instill expectations of a device prior to getting it and enforcing them once in place. Parental blocks have always been around,

and have allowed parents to block their child from certain materials and media, but it is mostly one-sided: with the parents getting rid of any media deemed harmful in their opinion, and often leaving the child with a sense of frustration and distrust because of the lack of communication. To ensure that child can have a sense of independence, and the parent having the confidence that their child is following their expectations when it comes to browsing the internet, a system of checks and balances on screen time is needed.

1.3 Solution

HomeWatch is a web-based application that would give both parents and children ease of mind when it comes to overlogging. By connecting an at home server with any intended device, a parent can keep track of where, and how long their child is online. HomeWatch will allow parents to create a 'time budget' for their children in order to better space out their work and play time, tracking how long they have been using their device, and notifying both parties when their allotted time is almost or has run out. With games, social media, and easy access media on smartphones, it is easier to spend more time on the phone than it is to do productive work, and HomeWatch is there to make sure that work and play are budgeted effectively. HomeWatch is not intended to be an online filter, or a parental block, but a checks and balances system for trust and independence and meant to start a dialogue between parents and children on expected device usage.

1.4 Project Objective

Develop a web-based application that will give parents the ability to view and monitor network activity. Parents will have the option to enable notifications and set limits on all or specific devices for when they feel a device has been online for too long. HomeWatch will promote healthy online habits and help users be aware of how much time they are spending online. HomeWatch will have a simple and sleek design allowing those without very much technical knowledge to understand the information being presented to them. A Raspberry Pi computer will be used to monitor network activity. This will allow for constant monitoring as it will always be connected to the network. All data will be stored using Microsoft SQL Server database. A minor front end user interface shall be created highlighting all of the major functions within the application, and once the back end functions, server connections, and databases are in place, then they will be improved on and minor aesthetic functions shall be implemented to make HomeWatch feel like an authentic website.

1.5 Overview

The remainder of this final report outlines in detail how the project shall be completed. The report includes the following sections: design objectives, technologies used, methodology, budget, timeline, problems encountered, and future recommendations.

2. Discussion

2.1 Project Concept

This project was conceptualized by the Lead Software Developer of HomeWatch, Mark Rankey, who wanted to see more independence during use of the internet for a child, while also having the parents be sound of mind when letting their child roam the internet, which can be as vital a source of information as it is harmful. HomeWatch is envisioned to be an application that will assist in the tracking of a child's online use on their home network, and relay it to the application to give it a clear overview of the time allotments and websites and activity their child is using. HomeWatch is also meant to be a mediator of information rather than a parental block application, to allow communication between parent and child about expected internet usage. With further research done by the Project Manager, Stefan Salzmann, it was also found that parents and children alike also have the tendency to say they are addicted to their devices, and that it could be detrimental, so the idea to make a notification system for overlogging was also planned for to be a main aspect of HomeWatch, to always allow the parent to have a sense of control. It was also noted than as of recent, many online platforms, including Apple's iPhone, Microsoft's Xbox, and Instagram, have already taken measures to reduce screen time, but always limit the user experience, resulting in distrust with whomever controls the user experience.

HomeWatch's solution to the issue of overlogging on devices, and making sure that a parent is aware that their child is not using potentially harmful websites, is to allow for communication with the child that this system is being setup, it is also allowing a clear line of communication as to what the expectations are for a device. The parent will be able to monitor this activity without limiting physically the child's time on the device, and the parent may also use it to monitor their time as well if they are concerned about their usage.

2.2 Design Objectives

The objectives of this application are to set up an at home system for the use of the parent to relay usage of any devices connected to it in an easy to read format. If any time restrictions are in place, the objective is to have an indicator on the viewable schedule on the application that shows overlogging clearly, so the parent does not have to check every time on a site or activity. A filter that stores keywords that could potentially be deemed as harmful on a website, such as an age restricted website, or distressing media, will be highlighted appropriately according to the age of the child, as to discourage the use of harmful websites. All of these aspects will be relayed to a web based application that will be easy to access and view for a parent, and if there are any concerns as to what their child's device activity is, then they are able to consult the HomeWatch online application to do so.

A main initial goal that had to be abandoned early on was the concept of making this into a mobile application. To create a mobile application to discourage the use of your mobile device seemed counterintuitive, and to create a web based application was to essentially create a 'one-stop-shop' for checking the daily use of a device, and not be an application the user was actively checking.

The HomeWatch team wishes to create an easy to understand interface using HTML, CSS, and JS and having a Raspberry Pi interact with the user's network to get information from a text file, that will have information based on the user, by IP address, URLs based on websites that were pinged during a session, and the times of that session. By storing that information, HomeWatch will be able to access it from an SQL database to be then displayed to the user.

2.3 Methodology and Goals

In order to create a web based application with the function of being able to keep track of screen times, as well as relay it back to the web service in a distinct format, the HomeWatch team will use a combination of Hypertext Markup Language (HTML) and Cascading Style Sheets (CSS)

in order to create an aesthetically pleasing UI. By utilizing Angular framework, HomeWatch can easily be designed with functionality in JavaScript, to allow for the data being presented to be formatted in a way that the user

can easily interact with and read. By creating first the main idea of HomeWatch: the screen tracking system, in the UI, coding for how the data will be stored in the database, and how it will be sent server side to the Raspberry Pi will easily be implemented thereafter.

HomeWatch's development will also be conscious of unit testing while in development so that testing can occur during and after development. The design of HomeWatch requires that it identify a device: HomeWatch will use a device's IP address to communicate the online services that a device is using, and communicate it through the Raspberry Pi to the HomeWatch web service to allow it to be seen, in real time, a device's screen time, and activity.

2.4 User Profile

Potential Users

- ❑ Parents, grandparents, guardians, babysitters, and older siblings who would like to monitor their network for people they are looking over
- ❑ People who would like to monitor themselves on how much time they spend on certain applications

Software, Interface, and Related Experience

All HomeWatch users should have experience with how to use a personal computer and navigating a website. They will also have to be able to interact with their router when the Raspberry Pi that HomeWatch uses is installed.

This project will be primarily targeted towards people who know how to use a web application who are needing to know what activities are happening on their home router, aimed towards parents and the monitoring of their children.

Regardless of the technological experience of the user, they will be able to quickly understand and utilize the features of our application.

Experience with Similar Applications

HomeWatch users may have experience with the below similar applications:

- Qustudio
- Screen Time
- OurPact
- Net Nanny

Task Experience

At initial opening of the application, users will see a selection of users that have been monitored that they can select from. After selecting a user, they can then see how much time is being spent in certain applications and/or websites for that person. From this screen, the person using the

application will be able to select a more detailed display of information, such as charts displaying usage over weeks/months and which application or website is being used the most/least

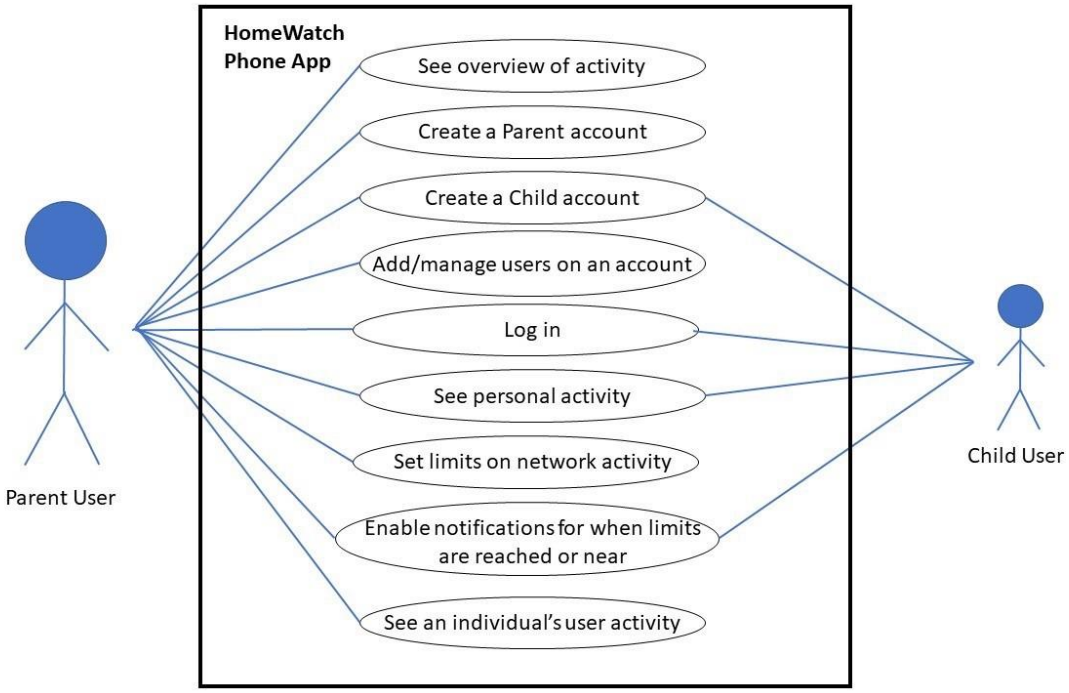
Frequency of Use

This application, in its beta phase, is intended to be used when an individual is trying to monitor their own or their children's usage of certain applications or websites with overall time being the main metric.

Key Interface Design Requirements that the Profile Suggests

- ❑ Simple, intuitive UI
- ❑ Fluid/Responsive design for accessibility
- ❑ Easily understandable display of information that can be accessed quickly

2.5 Use Case Diagram



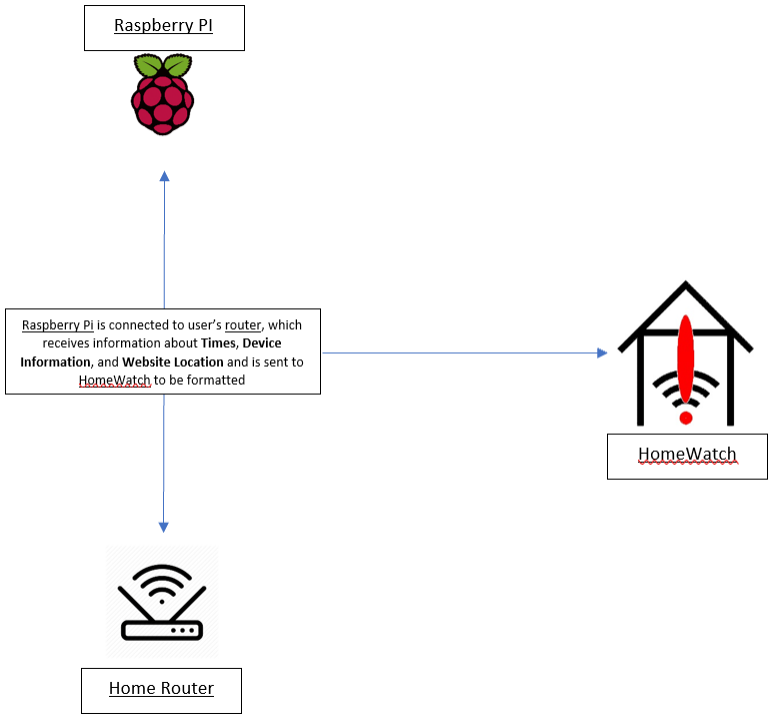
2.6 Technical Discussion

Gathering information about which devices were using certain websites at certain times is vital to our project, so the only way to do this would be to use a network. Since our project would be used by families in a home, gathering network data from a local router would be the route we would need to take. By allowing the router to have an SSH connection with the Raspberry Pi, the Raspberry Pi would have more data to look at when gathering the data we need to send over to our application. After setting SSH, the raspberry pi would be sent log information from the router every 60 seconds, which could then be reformatted to be read more properly using specific commands. Now that the Raspberry Pi has the information needed to be used by our application, it needs to

be retrieved and the best way to do this would be to set up a local network samba share on the Raspberry Pi. As all of the devices needing to access the samba share are all on and using the same Wi-Fi network, then that device can access the log on the Raspberry Pi. After connecting to the share, it is as simple as creating code to open the share and reference the log file needed in our application.

The way this application will function from start is a user will create an account using their specified information based on the parameters provided; for email, name, and other information that is vital to account creation. From creating an account, the user will be able to add devices to monitor based on what their home router is seeing from when devices connect to it. HomeWatch's Raspberry Pi will only monitor devices and what websites they are on for how long, and so only that information will be displayed. The user will then be able to see information unique to the device(s) added on their HomeWatch network and see data pertinent to the websites and URLs they are pinging. Below is Figure 3: Technical Diagram, that shows all aspects of hardware, the Raspberry Pi and user's home router, and how that interacts with the software of HomeWatch. The figure displays how the application functions at a basic level that includes all items as well as how they are involved with one another.

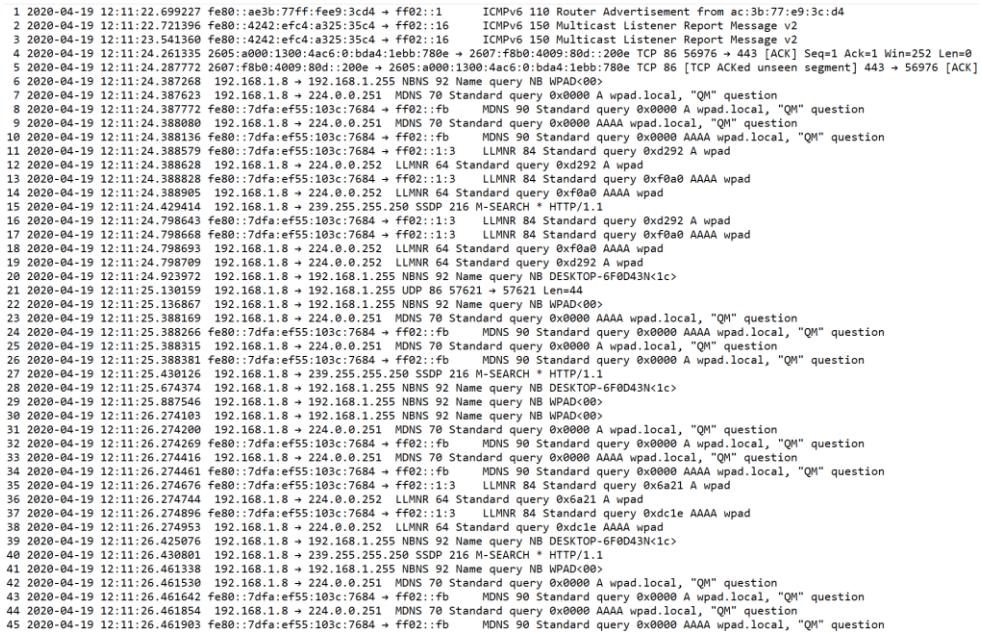
Figure 3: Technical Diagram:



Some of the security implemented in this project would be account credentials and separating where the data is being used. By having a separate account usernames and passwords on the both the Raspberry Pi and the network samba share, we avoid having to worry about users or other bad actors from accessing sensitive data. Also, by having our Raspberry Pi collect the information and having a database that stores that data, we also avoid having everything being hacked in to and manipulated in one spot. This means that if one portion of our application is compromised, someone would not have access to everything and the damage done would be minimized.

One of HomeWatch's key components was the ability to get information from a user's router into the application itself to be formatted and displayed into a way that would be useful to the user, and the way be pursued that was that we were going to have a device, a Raspberry Pi, connect to the user's router, acting as a mirroring port, at the recommendation of our advisor. From there, HomeWatch's cybersecurity analyst, planned to create a way to get a text file from the Raspberry Pi, to be put into a network file that could be accessed by the HomeWatch application to parse for times and IP addresses of devices on the network, as well as the websites the devices were pinging. Below is Figure 4: Test Text File, this figure displays the information, in an unreadable format, and with the help of HomeWatch, will be reformatted into useful information for the user.

Figure 4: Test Text File:



```
 1 2020-04-19 12:11:22.699227 fe80::ae3b:77ff:fee9:3cd4 → ff02::1      ICMPv6 110 Router Advertisement from ac:3b:77:e9:3c:d4
 2 2020-04-19 12:11:22.721396 fe80::4242:efc4:a325:35c4 → ff02::16     ICMPv6 150 Multicast Listener Report Message v2
 3 2020-04-19 12:11:23.541360 fe80::4242:efc4:a325:35c4 → ff02::16     ICMPv6 150 Multicast Listener Report Message v2
 4 2020-04-19 12:11:24.261335 2605:a000:1300:4ac6:0:bda4:1ebb:780e → 2607:f8b0:4009:80d::200e TCP 86 56976 → 443 [ACK] Seq=1 Ack=1 Win=252 Len=0
 5 2020-04-19 12:11:24.287772 2607:f8b0:4009:80d::200e → 2605:a000:1300:4ac6:0:bda4:1ebb:780e TCP 86 [TCP ACKed unseen segment] 443 → 56976 [ACK]
 6 2020-04-19 12:11:24.387268  192.168.1.8 → 192.168.1.255 NBNS 92 Name query NB WPAD<00>
 7 2020-04-19 12:11:24.387623  192.168.1.8 → 224.0.0.251  MDNS 70 Standard query 0x0000 A wpad.local, "QM" question
 8 2020-04-19 12:11:24.387772 fe80::7dfa:ef55:103c:7684 → ff02::fb     MDNS 90 Standard query 0x0000 A wpad.local, "QM" question
 9 2020-04-19 12:11:24.388080  192.168.1.8 → 224.0.0.251  MDNS 70 Standard query 0x0000 AAAA wpad.local, "QM" question
10 2020-04-19 12:11:24.388136 fe80::7dfa:ef55:103c:7684 → ff02::fb     MDNS 90 Standard query 0x0000 AAAA wpad.local, "QM" question
11 2020-04-19 12:11:24.388579 fe80::7dfa:ef55:103c:7684 → ff02::1:3    LLMNR 84 Standard query 0xd292 A wpad
12 2020-04-19 12:11:24.388628  192.168.1.8 → 224.0.0.252  LLMNR 64 Standard query 0xd292 A wpad
13 2020-04-19 12:11:24.388828 fe80::7dfa:ef55:103c:7684 → ff02::1:3    LLMNR 84 Standard query 0xf0a0 AAAA wpad
14 2020-04-19 12:11:24.388905  192.168.1.8 → 224.0.0.252  LLMNR 64 Standard query 0xf0a0 AAAA wpad
15 2020-04-19 12:11:24.429414  192.168.1.8 → 239.255.255.250 SSDP 216 M-SEARCH * HTTP/1.1
16 2020-04-19 12:11:24.798643 fe80::7dfa:ef55:103c:7684 → ff02::1:3    LLMNR 84 Standard query 0xd292 A wpad
17 2020-04-19 12:11:24.798668 fe80::7dfa:ef55:103c:7684 → ff02::1:3    LLMNR 84 Standard query 0xf0a0 AAAA wpad
18 2020-04-19 12:11:24.798693  192.168.1.8 → 224.0.0.252  LLMNR 64 Standard query 0xf0a0 AAAA wpad
19 2020-04-19 12:11:24.798709  192.168.1.8 → 224.0.0.252  LLMNR 64 Standard query 0xd292 A wpad
20 2020-04-19 12:11:24.923972  192.168.1.8 → 192.168.1.255 NBNS 92 Name query NB DESKTOP-6F0D43N<1c>
21 2020-04-19 12:11:25.130159  192.168.1.8 → 192.168.1.255 UDP 86 57621 → 57621 Len=44
22 2020-04-19 12:11:25.136867  192.168.1.8 → 192.168.1.255 NBNS 92 Name query NB WPAD<00>
23 2020-04-19 12:11:25.388169  192.168.1.8 → 224.0.0.251  MDNS 70 Standard query 0x0000 AAAA wpad.local, "QM" question
24 2020-04-19 12:11:25.388266 fe80::7dfa:ef55:103c:7684 → ff02::fb     MDNS 90 Standard query 0x0000 AAAA wpad.local, "QM" question
25 2020-04-19 12:11:25.388315  192.168.1.8 → 224.0.0.251  MDNS 70 Standard query 0x0000 A wpad.local, "QM" question
26 2020-04-19 12:11:25.388381 fe80::7dfa:ef55:103c:7684 → ff02::fb     MDNS 90 Standard query 0x0000 A wpad.local, "QM" question
27 2020-04-19 12:11:25.430126  192.168.1.8 → 239.255.255.250 SSDP 216 M-SEARCH * HTTP/1.1
28 2020-04-19 12:11:25.674374  192.168.1.8 → 192.168.1.255 NBNS 92 Name query NB DESKTOP-6F0D43N<1c>
29 2020-04-19 12:11:25.887546  192.168.1.8 → 192.168.1.255 NBNS 92 Name query NB WPAD<00>
30 2020-04-19 12:11:26.274103  192.168.1.8 → 192.168.1.255 NBNS 92 Name query NB WPAD<00>
31 2020-04-19 12:11:26.274200  192.168.1.8 → 224.0.0.251  MDNS 70 Standard query 0x0000 A wpad.local, "QM" question
32 2020-04-19 12:11:26.274269 fe80::7dfa:ef55:103c:7684 → ff02::fb     MDNS 90 Standard query 0x0000 A wpad.local, "QM" question
33 2020-04-19 12:11:26.274416  192.168.1.8 → 224.0.0.251  MDNS 70 Standard query 0x0000 AAAA wpad.local, "QM" question
34 2020-04-19 12:11:26.274461 fe80::7dfa:ef55:103c:7684 → ff02::fb     MDNS 90 Standard query 0x0000 AAAA wpad.local, "QM" question
35 2020-04-19 12:11:26.274676 fe80::7dfa:ef55:103c:7684 → ff02::1:3    LLMNR 84 Standard query 0x6a21 A wpad
36 2020-04-19 12:11:26.274744  192.168.1.8 → 224.0.0.252  LLMNR 64 Standard query 0x6a21 A wpad
37 2020-04-19 12:11:26.274896 fe80::7dfa:ef55:103c:7684 → ff02::1:3    LLMNR 84 Standard query 0xdc1e AAAA wpad
38 2020-04-19 12:11:26.274953  192.168.1.8 → 224.0.0.252  LLMNR 64 Standard query 0xdc1e AAAA wpad
39 2020-04-19 12:11:26.425076  192.168.1.8 → 192.168.1.255 NBNS 92 Name query NB DESKTOP-6F0D43N<1c>
40 2020-04-19 12:11:26.430801  192.168.1.8 → 239.255.255.250 SSDP 216 M-SEARCH * HTTP/1.1
41 2020-04-19 12:11:26.461338  192.168.1.8 → 192.168.1.255 NBNS 92 Name query NB WPAD<00>
42 2020-04-19 12:11:26.461530  192.168.1.8 → 224.0.0.251  MDNS 70 Standard query 0x0000 A wpad.local, "QM" question
43 2020-04-19 12:11:26.461642 fe80::7dfa:ef55:103c:7684 → ff02::fb     MDNS 90 Standard query 0x0000 A wpad.local, "QM" question
44 2020-04-19 12:11:26.461854  192.168.1.8 → 224.0.0.251  MDNS 70 Standard query 0x0000 AAAA wpad.local, "QM" question
45 2020-04-19 12:11:26.461903 fe80::7dfa:ef55:103c:7684 → ff02::fb     MDNS 90 Standard query 0x0000 AAAA wpad.local, "QM" question
```

By parsing through the file provided by the Raspberry Pi, the software developers can take integers for dates and IP addresses, and turn them into a readable format that shows devices URL access.

2.7 Testing Plan and Report

Overview and Methodology

In order to properly test our application and to see if all necessary features are working correctly, we have developed the following methodology:

1. Testing during newly implemented features as soon as they have been created periodically until completion/final unit test.
2. Both software developers of the group, Mark and Stefan, worked symmetrically on various features of the code to allow for more thorough testing.
3. Agile methodology was used to become more lenient with what features to expand or hinder

In order to keep track of all the aspects of our application, our group created lists of expected interactions from the application in order to have a clear guideline of unit testing once the code was being written. In short, as soon as a feature was thought of or coded into HomeWatch, our group would make a log of an event to then be entered into a Word document for later all-around unit testing. This was to make sure that the unit testing process would be expedited in a way to where nothing was forgotten and everything was accounted for. Testing during the implementation of our application also assisted with finding key areas of weakness that were improved.

The agile methodology was pertinent to the design of HomeWatch, as the framework in the beginning was needing to be changed a few times to allow for a web application to be supported in the way we wanted: with a Raspberry Pi acting to port mirror the web addresses being accessed by computers. We also needed to change how we would format the data being portrayed on each account, which also meant the code needed to be edited on numerous occasions. By allowing for change withing our application, the group was able to give what they thought was the most simples aesthetic way to portray data, while also not limiting the application's features.

Developer Symmetrical Testing:

In order to best test our application and utilize our number of software developers, the two developers of the HomeWatch group, Mark and Stefan, would work together in order to code, and then test code. Once someone had finished a feature, the other would test it out in various scenarios to make sure it was working properly, and to perhaps notice something that the original developer missed. By working in this fashion, features were checked thoroughly to make sure that unit testing would be expedited. In addition to symmetrical working, if a feature needed to be mocked, the developers were able to do so to allow for other features to be tested, with the assumption of an unfinished feature.

Scope of Testing

What will be noted in our Unit Test Scope is all of our major features from the parent user within HomeWatch. In addition to coverage of unit testing in our web application, independent tests of

our hardware, Raspberry Pi, were also done to assure that all aspects of our application were covered in detail.

Our unit testing will cover the adding user devices as a parent user as well as noting the times at which those devices were accessed and used. Both the hardware and the software being used will be monitored to ensure that all unit testing is passing, and if any errors occur, to take note of them, and make edits and keep iterating them until fixed.

Below, in Table 1: Unit Tests, are some example test case scenarios that we thought would be best to include in a Unit Test Case for a website, as well as detailed parameters as to what the key functions of HomeWatch are meant to accomplish:

Table 1: Unit Tests

Test Name	Item Type	Expected Result	Actual Result
Parent Create Account	Test Case	Parent Successfully Creates Account	**Pass**
Parent Info Stored	Test Case	Parent Account Successfully Stores Account Information	**Pass**
Parent Add Device	Test Case	Parent is able to add a device to HomeWatch to be monitored	**Pass**
Page Navigation	Test Case	Parent is able to navigate the pages successfully using the links provided in the website	**Pass**
Parent See Data On Device	Test Case	Parent is able to see website and time used	**Pass**
Filtered Words Apply Altered text	Test Case	If thought to be a malicious website, HomeWatch highlights text based on keywords	**Pass**
Data Retains when exiting app	Test Case	Once exiting the application and reopening, all data from previous session is retained	**Pass**

Unit Test Objectives

1. Any and all features will be unit tested during this process, the group will start will a main feature and work downward to secondary features based on if it branches from the main.
2. All unit tests must be reported, pass or fail, and detailed with behavior of the application during a unit test.
3. Before displaying HomeWatch at the Tech Expo, all bugs must be coded correctly and fixed and al unit tests must be passing

Logging Test and Procedures

To log any and all testing that was done, and what features needed to get tested, the project manager, Stefan, added all features to be tested into a word document. All features listed were features thought of by the group in terms of what necessities should be brought into HomeWatch. From this document, HomeWatch will be able to complete unit testing easily and within a format that details exactly what is needing to get completed so that all tests are passing.

Once all unit test complete preliminary testing, then our lead developer, Mark, will begin final unit testing with Stefan on standby to begin any and all corrections needing to be coded to have all tests passing, once the preliminaries have been completed, Mark will join Stefan in completing the coding to fix the errors, and then final unit testing will begin and reiterate with the method above until completion.

What We Learned During Testing

We learned that programming the application around port mirroring isn't as simple as it seems, and allows for a numerous amounts of iterations to get working without issue. With our

Raspberry Pi acting as a mirroring port to allow us to grab information about what devices are pinging what sites, we had to plan several times, and attempt to execute this feature of our project.

Considering that our group consisted of two software developers and one cybersecurity major, it was important to us that the software and hardware relayed the concentration of all our majors in an equal way so that all of our expertise would be demonstrated.

We also learned that testing and developing can coincide as period testing is happening and there is no falling behind with either development or testing: they have to have a clear balance to maintain. During testing, the HomeWatch group was adamant to make sure that all time spent working on the development of HomeWatch was recorded and that any and all equipment costs were recorded. Below the testing plan, describes the budget for HomeWatch and all finances that were attributed to its completion.

2.8 Budget

Table 2: Budget, illustrates HomeWatch's budget details, including development costs and costs for hardware. It does not include costs for essential items to use HomeWatch such as, personal computer, router, and internet provider. All items that are unique to HomeWatch are detailed and any labor costs are superficial, as HomeWatch was developed as a senior design project and is not receiving a monetary incentive for labor costs.

Table 2: Budget

Project Name:	HomeWatch	**Project Manager:**	Stefan Salzmann	**Team #**	40
Project Members:	Stefan Salzmann Mark Rankey Josh Bosse	**Project Areas:**	Software Development and Cybersecurity	**Project Advisor:**	Ryan Moore

Problem Statement

It was found, in a recent study from PSYCOM, that teenagers, and adults are both experiencing issues when it comes to their device: 50% of teens say they feel that they are addicted to their phones, while 60% of parents believe that their children have issues with overlogging. Both adults and children surveyed portrayed addictive tendencies when it came to their device.

Project Description

HomeWatch is a web-based application that would give both parents and children ease of mind when it comes to overlogging. By connecting an at home server with any intended device, a parent can keep track of where, and how long their child is online. HomeWatch will allow parents to create a 'time budget' for their children in order to better space out their work and play time, tracking how long they have been using their device, and notifying both parties when their allotted time is almost or has run out. With games, social media, and easy access media on smartphones, it is easier to spend more time on the phone than it is to do productive work, and HomeWatch is there to make sure that work and play are budgeted effectively. HomeWatch is not intended to be an online filter, or a parental block, but a checks and balances system for trust and independence and meant to start a dialogue between parents and children on expected device usage.

Project Asset Type		**Funding Source (if applicable)**	
Infrastructure	Comments: HomeWatch is meant to be an addition to a user's router, to be installed and send and receive information pertaining to devices on the user's network, being an addition to the user's infrastructure.	Self	The Raspberry Pi was purchased by Mark Rankey, and was the only monetary expense of this project, detailed below. All softwares were free to download and doesn't require hosting on a third party site.

Risk Identification (See Risk Types tab)

	Risk Rating 1-5 (5 is high)*	*Comments*	*Weight*	*Score*
Work Effort (days)	2		40%	0.80
Complexity	3	Aspects such as the Raspberry Pi and setting up a database might be timely tasks.	60%	1.80
Project Risk Score:				**2.60**

Project Stakeholder(s)

All current stakeholders of this project, being anyone who has interests, concerns, or is affected by the outcome of HomeWatch are the team members who built the project, Stefan Salzmann, Mark Rankey and Josh Bosse, as well as the team and project's advisor, Ryan Moore, and the professor of the Senior Design Management class, Jim Scott.

Estimate of Benefits

If project will generate revenue, estimate 1 year here:	$ 14,000.00

Select other benefits the project may bring a customer or user:

Risk Avoidance	☑
Improved customer satisfaction	☐
Increased system availability	☑
Productivity improvement	☑
Reduced costs	

Estimated Cost Rough Order of Magnitude:

				Ongoing Annual		
	Rate Per/Hr	**Work Effort (Hours)**	**1 X Costs**	**Rate Per/Hr**	**Work Effort (Hours)**	**1 X Support Cost**
Labor - IT	20	700	$ 14,000.00	20	0	$ -
Labor - External	0	0	$ -		0	$ -
Software - External						
Hardware - External						
Misc.						
TOTAL			$ 14,000.00			$ -

Comments: in addition to the labor costs, an external Raspberry Pi device was purchased and was an additional $109.95. HomeWatch is a remote device that is connected to the user's network, so maintenance costs are virtually nothing, and it is not requiring server or network costs.

5-Year ROI Analysis

Description	**5- Year Expected**	**Conservative (1.5)**
Total Costs	**$ 14,000.00**	**$ 21,000.00**

Total Benefit	**$**	**70,000.00**	$35,000
Total Costs/Benefit Differential			
	$	**56,000.00**	
Conservative Costs/Benefit Differential			
	$	**14,000.00**	

Pictured above is Table 2: Budget. The table details, again, the problem statement and description of HomeWatch, and from there begins to delve into detail about the budget situation of HomeWatch. HomeWatch is described as an infrastructure asset, meaning it is meant to improve/be implemented with the user's home router and function off of the software that HomeWatch is built on. From there, the table delves further into the funding for HomeWatch, being it is a senior design project, it is entirely funded by the HomeWatch team, detailing extactly what monetary expenses were debited during the allotment of this project. Identification of risks, as well as potential stakeholders who would be impacted by that risk, are outlined and given a numbered value to show the impact of work hours and complexity. The predicted amount of yearly revenue is predicted as well as the benefits; while benefits of the consumer are very much clear, the amount of 'revenue' that HomeWatch is expected to acquire is very little, in that it doesn't take much to reproduce the hardware, and the main revenue would mainly be because of production costs of the software. Labor costs are an assumed rate for employees in our field and experience level and the hours that the HomeWatch team has put into this project are an accurate estimation based on the team members' accounts, and the time table based on Table 3: Gantt Chart, detailed below. The Gantt chart is to show tasks that the HomeWatch team needs to accomplish as well as a time allotment in days and the days at which the task is planned to be completed.

Table 3: Gantt Chart

2.6 Gantt Chart

Task Name	Duration (Days)	Start Date	End Date
1. 0 Project Management and Deliverables	**7**	**8/26/2019**	**9/2/2019**
1.1 Team Building	**28**	**8/26/2019**	**9/23/2019**
1.2 Ideas and Brainstorming	**5**	**9/17/2019**	**9/23/2019**
1.3 Fall Semester Assignment 0: Team Members & Project Name	**7**	**8/26/2019**	**9/2/2019**
1.3.1 Agree on project idea, name, and branding/logo	**10**	**9/19/2019**	**9/30/2019**
1.4 Fall Semester Assignment 1: Team Contract	**21**	**9/2/2019**	**9/23/2019**
1.4.1 Project Approval	**23**	**8/26/2019**	**9/19/2019**
1.4.2 Work on Team Contract	**7**	**9/16/2019**	**9/23/2019**
1.4.3 Finalize Team Contract and submit	**1**	**9/23/2019**	**9/23/2019**
1.5 Fall Semester Assignment 2: Project Abstract for Tech Expo	**20**	**9/24/2019**	**10/14/2019**

1.6 Fall Semester Assignment 3: Team Contract Resubmission	**20**	**9/24/2019**	**10/14/2019**
1.7 Fall Semester Assignment 4: User Profile	**13**	**10/8/2019**	**10/21/2019**
1.8 Fall Semester Assignment 5: Use Case Diagram	**13**	**10/8/2019**	**10/21/2019**
1.9 Fall Semester Assignment 6: Draft Report	**12**	**10/22/2019**	**11/4/2019**
1.10 Fall Semester Assignment 7: Final Fall Semester Report	**28**	**11/4/2019**	**12/2/2019**
1.11 Fall Semester Oral Presentation	**14**	**11/4/2019**	**11/18/2019**
1.11.1 Presentation Practice	**13**	**11/4/2019**	**11/17/2019**
1.12 Spring Semester Assignment 1. Testing Plan/Report	**28**	**1/13/2020**	**2/10/2020**
1.13 Poster Preparation	**20**	**2/17/2020**	**3/8/2020**
1.13.1 Critique Expo Posters in Class	**1**	**2/24/2020**	**2/24/2020**
1.13.2 Spring Semester Assignment 3. Draft Tech Expo Poster	**1**	**3/2/2020**	**3/2/2020**
1.13.3 Spring Semester Assignment 4. Final Poster Submission	**1**	**3/9/2020**	**3/9/2020**
1.14 Spring Semester Assignment 2.Final Abstract	**1**	**2/17/2020**	**2/17/2020**

1.14.1 Final Abstract Editing	**30**	**1/17/2020**	**2/16/2020**
1.15 Spring Semester Assignment 5. Final Report Submission	**1**	**4/6/2020**	**4/6/2020**
1.15.1 Final Report Preparation & Editing	**80**	**1/17/2020**	**4/5/2020**
1.15.2 Spring Assignment 6. Final Report Safe Assign Submission	**1**	**4/6/2020**	**4/6/2020**
1.16 Tech Expo	**1**	**4/14/2020**	**4/14/2020**
1.16.1 Tech Expo Preparation	**43**	**2/20/2020**	**4/13/2020**
1.17 Spring Semester Assignment 7. Final Library Copy	**1**	**4/20/2020**	**4/20/2020**
2.0 Research	**50**	**9/10/2019**	**10/30/2019**
2.1 Software Requirements	**33**	**9/10/2019**	**10/13/2019**
2.1.1 Determine Front End Development Languages	**20**	**9/10/2019**	**9/30/2019**
2.1.2 Determine Back End Development Languages	**33**	**9/10/2019**	**10/13/2019**
2.2.1 Determine Hosting Environment	**35**	**9/10/2019**	**10/15/2019**

2.2.2 Determine Database Environment	**50**	**9/10/2019**	**10/30/2019**
2.3 Security Requirements	**42**	**9/10/2019**	**10/22/2019**
2.4 Miscellaneous Research	**35**	**9/16/2019**	**10/21/2019**
3.0 System Design	**37**	**9/23/2019**	**10/30/2019**
3.1 Create System Diagrams	**32**	**9/23/2019**	**10/25/2019**
3.1.1 Create Network Diagrams	**12**	**9/23/2019**	**10/5/2019**
3.1.2 Create Database Diagrams	**27**	**9/23/2019**	**10/20/2019**
3.1.3 Create Interaction Diagrams	**32**	**9/23/2019**	**10/25/2019**
3.1.4 Create Wireframe Diagrams	**37**	**9/23/2019**	**10/30/2019**
4.0 Environment Set-Up	**125**	**10/1/2019**	**2/4/2020**
4.1 Create Development Environment	**7**	**10/1/2019**	**10/7/2019**
4.2 Create GitHub Repository	**7**	**10/1/2019**	**10/7/2019**
5.0 Development (Back End and Front End)	**151**	**11/1/2019**	**4/1/2020**

6.0 Testing	**70**	**1/20/2020**	**4/1/2020**
2.2 Network Requirements	**50**	**9/10/2019**	**10/30/2019**
6.1 Set Up Mock Clients	**10**	**1/20/2020**	**1/30/2020**
6.3 Security Test	**26**	**2/15/2020**	**3/12/2020**
6.3.1 Firewall Test	**14**	**2/15/2020**	**2/29/2020**
6.3.2 Penetration Test	**20**	**2/15/2020**	**3/5/2020**
6.3.3 SQL Injection Test	**26**	**2/15/2020**	**3/12/2020**
6.4 User Acceptance Test	**16**	**3/12/2020**	**3/28/2020**
6.4.1 User Acceptance Test for Parent	**16**	**3/12/2020**	**3/28/2020**
6.4.2 User Acceptance Test for Child	**16**	**3/12/2020**	**3/28/2020**
6.5 Final Testing	**12**	**3/29/2020**	**4/10/2020**

2.7 Problems Encounter

An issue that was faced by the team early was whether this application should be a mobile application or a web application. The issue was that it seemed rather counterintuitive to have a device that is used for web browsing be the same device that is checking to track network and website usage. We also found that sending network data from raspberry pi to mobile application would be difficult because of the way that the raspberry pi would have to send text files through a shared folder.

Another problem that came up was getting the raspberry pi to properly send data to HomeWatch. This portable computer was new each member and learning how to operate it was being to become a challenge as this was a new operating system not familiar to anyone. We decide to try another approach of using JavaScript to parse Chrome's history tab, but decided that using the raspberry pi in our project was necessary and could be very useful in getting the correct data that we need, as well as Chrome's history file is limited. By having the history file on a user's computer, it is limiting the websites accessed by only allowing for information pertaining to that device, as opposed to the whole network, an attribute relying solely on the function of the Raspberry Pi, making the Raspberry Pi an essential aspect of our design.

2.8 Problems Solved

Since all of us would be more comfortable, we agreed that HomeWatch would be built as a web application. Another benefit of HomeWatch being a web application would be that it would promote a healthier usage by the parents of how often they would be checking HomeWatch for updates on the devices connected to their home network.

The problem of learning how to operate the raspberry pi was solved by using all of the resources that we had. Some of these resources include many tutorial videos, co-workers who

have used a raspberry pi in the past or are familiar with command based operating systems, and by just continuing to investigate on our own to find out what each setting and command performed to fully understand. Over time, using the raspberry pi became easier and easier to the point that we felt that we were able to retrieve the information that we needed for the project to continue. Another function that was needing to be implemented was parsing the file provided by the Raspberry Pi for only information pertaining to the exact functions of HomeWatch: websites, times, and device identifiers.

2.9 **Recommendations for Improvement**

If we had to do this project all over again, we would have started investigating how to use the Raspberry Pi earlier and had more frequent meetups to see the progress that was being made. After we created the plan and some initial web application time was around the same time that we started to involve the Raspberry Pi in to our work, but in reality it should have been one of the first things that we should have set up as it is vital to the project. This created some headaches as all of the team members were working on different pieces, not leaving the optimal time needed for the raspberry pi. Our group did not meet up in person as we felt that everything could be shared online and that if we had any questions, they could be answered in text or email. This led to some things not being addressed to on time which could have been helped by having a weekly an in person meet up.

Another aspect that we would've done differently again is that before we even started software development, the group was attempting to find a decent framework to work with, rather than begin programming and form the framework as to what was needed. If repeated, the frameworks that the group would have used would be integrated as needed and adjusted as such.

Given more time, we would have created a more aesthetically pleasing web application and an even cleaner design. Many websites are given a unique look that when the first page is loaded, even without seeing the name, the user knows that it is a certain website. Given more time,

HomeWatch would've become a unique website that's aesthetic was one of a kind, and welcoming to new and old users alike.

3. Conclusion

Lessons Learned – Fall Semester 2019

What was learned throughout the development of this application was that in order to have all the devices in the household be connected to this web based application, HomeWatch had to be connected to the router of a user's home, hence the use of the Raspberry Pi as a sort of mediator of information.

The HomeWatch team utilized the Angular framework to support their application, and to get a rough outline of what the end-product shall look like. What is needed now is a connection to a database, and to the Raspberry Pi acting as the server, and then the back-end programming can continue exponentially, with the development team focusing on tying up the loose ends of server-side connection.

The HomeWatch team, in the spring semester of 2020, will focus on the hardware of the HomeWatch, and explain further how that will come into play for the user. The HomeWatch application is web based but used for only devices that are connected in the home, and a database storage of all the information a parent or child user is meaning to input needs to be stored in our database, using MongoDB software.

Lessons Learned – Spring Semester 2020

The main lesson that we learned during the course of the spring semester, is that allowing for the Raspberry Pi to become a mirroring port for information took a lot more research and a lot more time than thought, according to HomeWatch's Josh Bosse. Given more time to work on it, the Raspberry Pi would be able to only display information about what HomeWatch's functions are so that the software developer's of HomeWatch can parse through it more easily for important information.

More time would have also been spent on reviewing other websites of the same function and seeing exactly what aesthetics make their applications more clearly define their functions, such as adding in a timekeeper pop-up for child class users to use when using the internet, so that they do not go over the discussed time limits that their parents set for them. By learning how similar acting applications provide for their user, the HomeWatch group could have more easily seen what aspects of those websites make them unique and provide a unique atmosphere to HomeWatch based on those inspirations.

4. References

Hurley, Katie. “Teenage Cellphone Addiction: Are You Worried About Your Child?”. PSYCOM, LCSW. 06, August, 2019. https://www.psycom.net/cell-phone-internet-addiction

Cohen, Lisa. “New Report Finds Teens Feel Addicted to Their Phones, Causing Tension at Home”. Common Sense Media. 03 May, 2016. https://www.commonsensemedia.org/about-us/news/press-releases/new-report-finds-teens-feel-addicted-to-their-phones-causing-tension-at