Bryan Nunez, Neil Himes, Juan Uscanga, Kai T. Sleight,
Wargen Guittap, Sedachael Zeleke, and John Inductivo
CS 472 – Section 1001
Andreas Stefik
08 December 2020

Design Document III

Introduction

Abstract

Network Visualizer is intended to be an android application that will make it easier and more user friendly to understand the network the user is connected to. This is done by first categorizing, then displaying devices on the network in a digestible Graphics User Interface (GUI) with helpful information to explain important parts of a device(s) such as: defining what an IP is, what ports are open on a device mean. Apart from being educational, the device would have practical uses such as being able to perform Nmap scans on specific devices. The app will also be customizable allowing experienced users to use it as a practical tool.

Executive summary

In 2019, the Hiscox Cyber Readiness Report (This report gauges how well businesses are prepared for handling breaches and cyber incidents) [1] surveyed 5,400 firms and reported that 3,300 or about 61% reported an attack costing these businesses an average of $369,000. Now this number is huge, but obviously there are going to be some outliers - big businesses with huge losses that will skew the average. But when we take out those outliers, we still get an average loss of $200,000 per business which is almost a 6x increase from 2018's number of $34,000.

In this same report, the number of breaches for small businesses went up from last year's 33% to 47% this year. With this increase in the number of breaches you would expect that these small businesses would be much better prepared for data breaches. In 2018, Ponemon Institute [2] (institute who conducts research that advances the privacy management practices within business') surveyed 1,040 people from a variety of industries (Finance, Retailing, Services, etc.) and only 28 percent of the companies represented rated their ability to mitigate threats as "highly effective".

Cyber-attacks are getting more widespread and costing businesses more and more financially. According to Accenture [3] (global professional services company who specializes in security), if businesses do not do more to prevent these attacks the total value at risk is $5.2 trillion globally between the years of 2019 and 2023. What do we do to prevent it? One approach is for small business owners to routinely monitor and scan any device that is on the network.

Norbert Nthala (MSU researcher who specializes in information security) [4] et al. found that "prioritizing efforts on critical points in the home network infrastructure and developing appropriate tools to help home users with assessment, protection, monitoring, and incident response is a viable option." This quote shows the importance of small business owners having an application which will observe the devices on the network and detect any that are not recognizable or have open ports.

The issue is this has been done before and done in several different ways (good place to do the device comparison). Wiem Tounsi (PhD from Telecom Bretagne who specializes in cyber security science) et. al. emphasizes the importance of having the organization as a whole be well informed about vulnerabilities in computer networks [5]. The problem with these small businesses is that they could easily download a network manager, but in order for them to understand the magnitude of these cyber-attacks we need to educate them and help the small businesses understand what these vulnerabilities look like. This is what our app is about.

Motivation:

Competitive Analysis:

Network Manager (Android)

Network Manager is an android app which serves as a toolkit for checking multiple things on one's network. It has open-source tools such as Nmap incorporated into it, has a LAN scanner which lists devices on the network by their IP, and manufacturer ID, such as "Nvidia" or "Xerox". The app serves as a great toolkit, however, if a user does not know what they are looking for or has no knowledge of networks, they're berated with all this information and options which is a great thing for experienced users. However, users new to networks or who aren't knowledgeable on networks may be intimidated and turned away by the overwhelming number of options. (Screenshot shows 1/3rd of them).
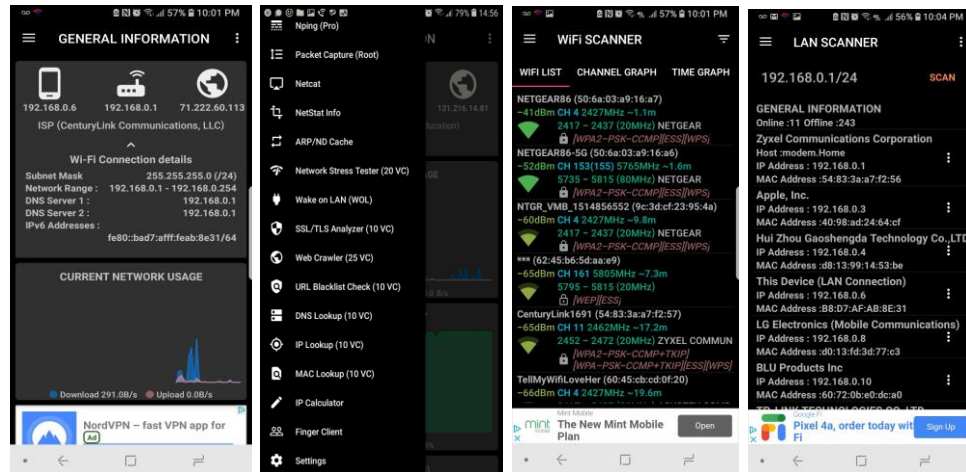
I believe the product's strength lies in the vast number of options it has and how easy it is for an experienced user to utilize them. You can create categories for specific devices and easily run commands on those devices with the click of a button.

The weakness is in how difficult it is to use those tools as an unknowledgeable user. Our app would simplify/compact these features into something more digestible, instead of cluttering the screen with information new users may not understand.

Wi-Fi Analyzer (Android)

Wi-Fi Analyzer is an open-source software designed to portray Network information using four main pages displaying Access Points, Channel Rating, Channel Graph and Time Graph. Furthermore, there is a Menu Icon allowing its user to navigate to the four main pages mentioned previously, Export Data, Show Available Channels, and detailed information regarding sources to the actual repository. It also allows the user to start and stop the scanning feature alongside the filter menu to search for specific SSID, choose Wi-Fi Band, signal strength and security information.
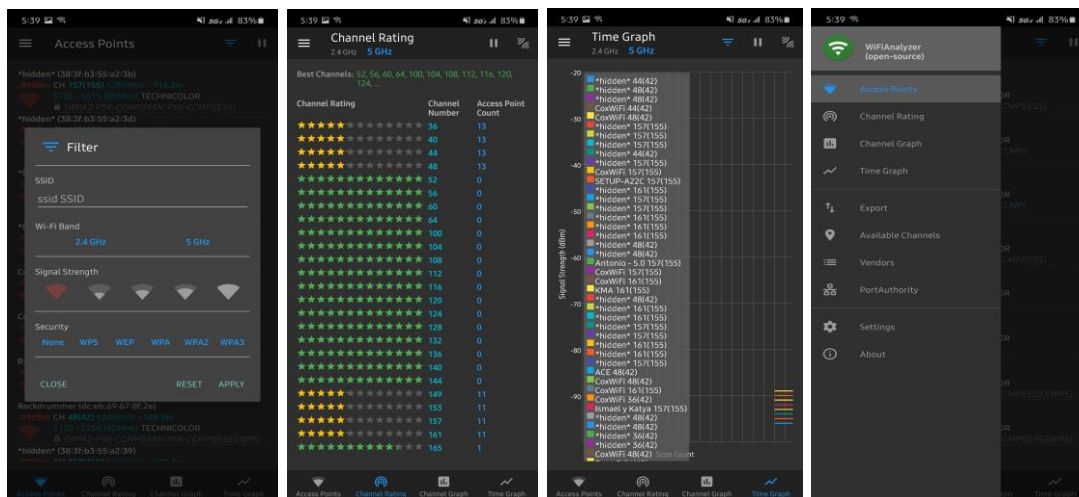
The application excels in portraying Network information into categories easy enough for a user knowledgeable about the subject. The application allows the user great control over the features and portrays data information into graphs and tables for visualization.

The detailed information this application produces is great but maybe a bit confusing for the non-tech savvy user. Our goal for our application is to use Wi-Fi Analyzer as our base and expand on its use as this is open source. We will further give our users the visualization. For every network that appears on scan, devices connected to the network will be visible by relevant icons. Alongside, we will provide a help screen to try and describe the details with less technicality. If time permits, we would like to add advanced features such as removing a device on a network or bandwidth restriction for users wanting a bit more control with their network.

ASUS Router (Apple iOS | Android)

ASUS Router is an app for users who have purchased an ASUS router to monitor and manage traffic, users, and many other features directly via the GUI on Apple devices. Features included within this application allow users to monitor traffic of individual devices, allowing for bandwidth limiting and complete blocking and blacklisting. Users can visualize traffic per device and switch devices to different channels to reduce overloading channel traffic. The QoS feature provides the option to prioritize specific packets such as gaming, media streaming, and file transferring. Parental controls allow the administrator to block all traffic or set a bandwidth limiting schedule for devices that are flagged for limiting. The system monitor provides visual real-time information flow of CPU usage and a metered reading of the percentage of CPU memory being used to inform the network administrator of memory capacity.
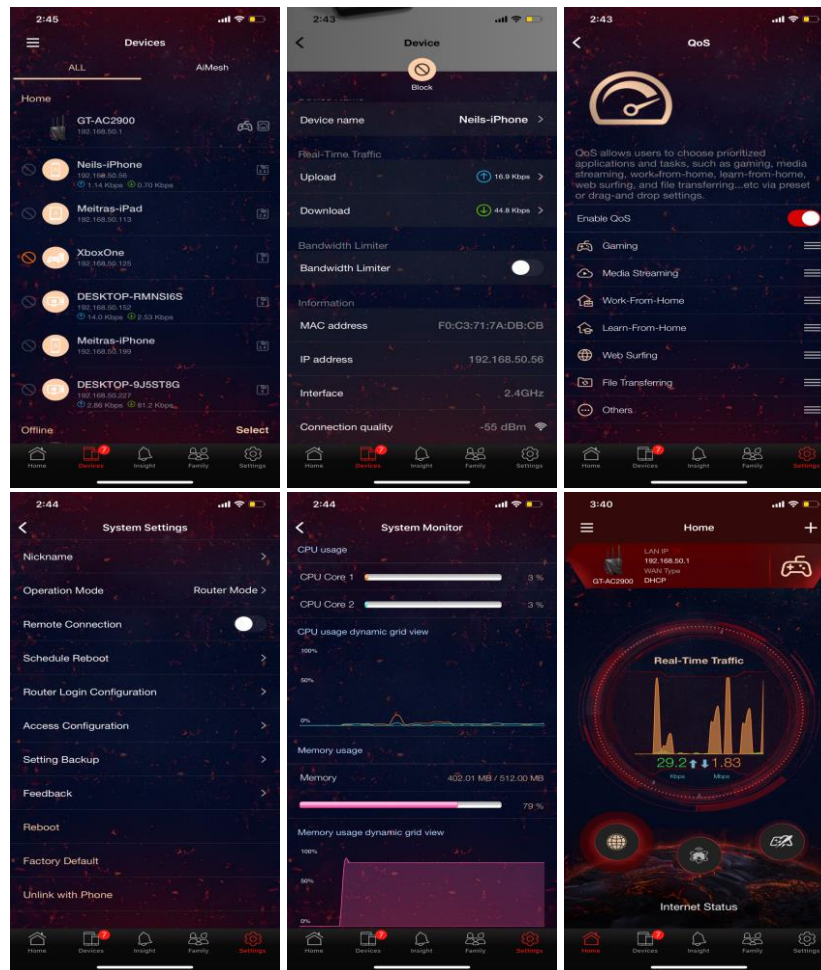
      The versatility of this application provides the network administrator and other users with detailed information and thorough explanations to ensure ease of use and understanding while providing full capability to protect and monitor the private network from beginners to experts in networking.

      Our application, if time permits, will implement various features accessible to the user on their private network such as monitoring and limiting bandwidth usage along with blacklisting unwanted devices and provide real-time traffic analysis as a visual aid of the network load being handled by their router.

Wi-fi Connection Manager (Android)

      Wi-Fi Connection Manager is an open-source Android application developed by "roamingsoft" (all lower case) for Android OS. This app works as a network manager and offers various useful network tools. On initialization, this app starts with a "Dark Theme", meaning it has a black background with contrasting

white text. For techy people, this theme is much preferred as it helps reduce strain on the eyes. However, to the average user, this theme can be off-putting and give the impression the application functionality may be beyond their grasp. From the main screen the user can perform a LAN scan, after scan each network is displayed with its name alongside much of its critical information (such as MAC address, network security options, connection speed, frequency, channel, etc.), all within a tightly packed list of other networks. If a user clicks on one of these networks, they will be introduced to a popup screen displaying even more specifications and statistics of that network. To someone who may know all these different readings and their functions, the wealth of information listed is appreciated, but to someone with little knowledge of networking or what the information signifies will be critically bamboozled. This is just the main screen, the extensive list of other tools and options available has yet to be mentioned.

Some of the features and tools this app offers are a ping tester (selectable web host server), LAN scan (with various scan filters), IP location, backup and restore, network connection priority, diagnose connection, fix Wi-Fi problems, connection history, and a Wi-Fi connection strength chart/spectrum. Besides the fact that there is no real explanation or tooltips for all the information, functions, settings, and tools offered throughout the app. All these options can be great once the user understands what they are looking at and how they can take advantage of it all. Another pro of this app is that it has options to display customized information onto the notification tray. This may be useful in showing users what new devices have been connected to their home networks, what the current signal strength is, and how much data has been uploaded or downloaded in a given time frame.
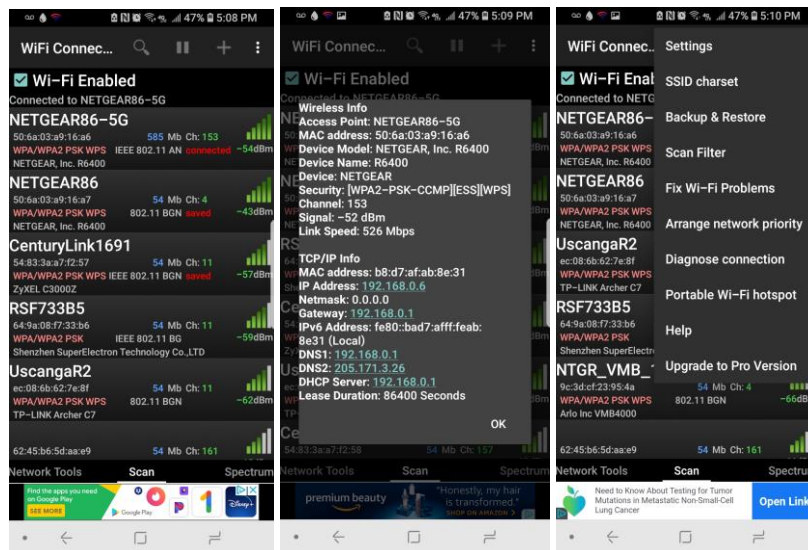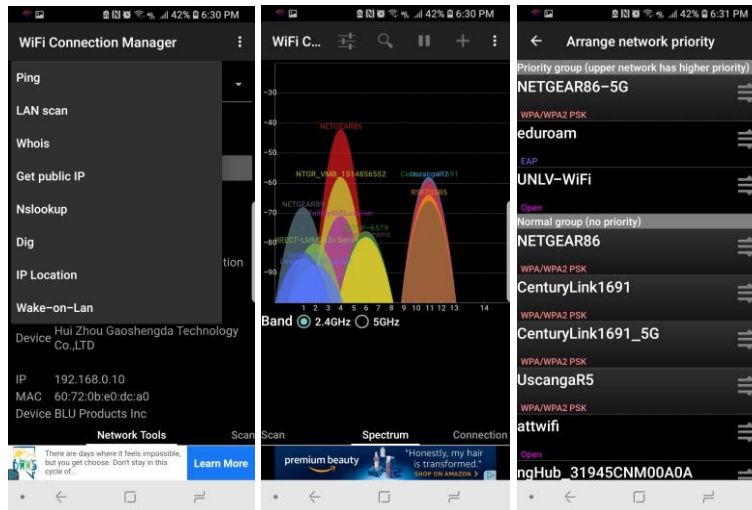
This app is a great example of something we could strive for in terms of features and user options given. This does not go without saying that in our attempt, we will be focusing on ensuring users can understand what all of these things are and how they can use them to improve and secure their private networks at home or in their businesses.

Journal Articles

"Rethinking Home Network Security" by Norbert Nthala and Ivan Flechais

This peer reviewed journal article dives into the importance of home network security and the practices that home users take in securing their network. The researchers illustrate this by conducting interviews to home users and internet service providers (ISPs) to better understand what precautions, if any, users take when dealing with their home network. The questions in the interview aimed at asking the home user their data/security concerns and if they have had any experience with monitoring or dealing with any security incident. Similarly, the researchers questioned the ISPs by asking them what services they provide to keep their customers safe. The results of these interviews showed that the users assumed that their routers and network devices were already secured from the ISP and companies where they purchased it, which led to many users not taking security measures for their devices.

This journal article impacts our product because it provides detailed information on what measures users are taking for network security. For example, this article has taught us the importance of network monitoring tools. We understand that managing your own network can be a daunting task for the average user, which is why we are building our Network Visualizer. Additionally, the

researchers said that responsibility and competence is important in-home network security. They gave an example saying that mobile applications can be a tool used for ISPs to transfer responsibility to the home user and that competence can be addressed through digital assistants, AI agents, and machine learning algorithms. When reading this, our group gained further motivation to build a mobile application that users have full control over. Additionally, we plan on making our users more competent by including resources, definitions, and security tips in our application that will allow them to gain more knowledge and experience in their home network. Overall, this article contains a lot of direct user information via the interviews. It is a resource that we will be revisiting throughout the development of our application.

"SpyGlass: A Wireless Sensor Network Visualizer" by Carsten Buschmann, Dennis Pfisterer, Stefan Fischer, Sándor P. Fekete, and Alexander Kröller

This peer review journal emphasizes the importance of visualizing wireless sensor networks. A wireless sensor network is made up of multiple sensors that collect data from the network environment and send it through the network to the receiver. Visualizing this data is important for further understanding and operating network sensors. The journal further explains a possible GUI for such a software and a possible canvas made up of three layers. The first layer is for painting the background, the second layer is for showing the nodes and their relationship, and the third is for displaying the actual nodes. The journal also explains possible plug-ins available to help support the application's goal in visualization.

This journal illustrates the importance of understanding a network through visualizing them. SpyGlass, the application talked about in this journal, is similar to what we are trying to accomplish. It is known that not many people know how networks work. Having a GUI, like Spyglass, can make it more user friendly for network owners to visualize their networks for home or business. Finally, the in-depth explanation of how the GUI and canvas are set up will help us gather ideas for our project.

This peer review journal goes into detail about the importance of keeping the users in mind when creating an efficient UI/UX. The journal begins to discuss how continuous updates need to be done on the app in order to prevent the feeling of having an inconvenient application. The journal then goes into detail on a method for making a good UI/UX by analyzing the user's cognitive and behavioral information. For the cognitive test, there were a series of vision tests (font size), tap tasks, word span tasks, etc. given to the user and after completing the tasks the user was given a satisfaction test to see what UI model worked best. In the user-behavioral model, the researchers observed the activity profile of various users. The results from the observations helped the researchers modify the application and the overall satisfaction of the users was extremely positive.

This journal will be useful for us when deciding a good UI/UX for our target audience. We want to target people who are not tech savvy and know very little about internet networks. After creating our first prototype, it would be ideal to give the app to several different users and observe their behavioral habits and ask them if there could be any improvements to how the existing UI/UX looks and feels. It would also be beneficial for us to first test all of the users' cognitive abilities so we know what changes will help a specific subset of users and what changes could potentially harm the experience of other users.

Requirements document

High-Level Description

The purpose of Network Visualizer is to develop a beginner friendly application to network management. As such, the functionality and features listed below are aimed at fulfilling all the basic features a novice would need to navigate their own network. Some of these features will include: a scan button to detect devices on a network, a "show devices" feature that will display all the scanned devices within that category, and a more info button that will display detailed information of that device's settings such as the IP address and mac address. Finally, one of the main marketing themes our application is trying to accomplish is the idea of user friendliness towards beginners. As such, we are including information buttons that will give the user a definition of what certain concepts are (an example can also be seen in the prototype below with the IP address definition).

Functional Requirements Table

ID	Title	Requirement Description	Date
F1	Scan button	The Scan Button will allow the user to hold down the scan button to perform a scan of the LAN.	10/23/20
F2	Navigation Menu Button	The Navigation Menu Button will allow users to navigate through the app's other screens such as for the settings, about, and help pages.	10/28/20
F3	About	The About Button will display details of the developer team and application implementations. Links to contact information, GitHub, social media, etc.	10/09/20
F4	Help	The Help Button will display a page that will give users more information about the application's purpose and functionality.	10/31/20
F5	Search	The Search button will search for specific devices.	11/5/20
F6	Nav bar	The Navigation bar on the bottom of the screen will display more options that will allow the user to navigate the GUI.	10/18/20
F7	Signal strength	The Signal Strength icon will show each device's internet connection signal strength.	10/23/20
F8	Graphical User Interface	The GUI will display all devices and helpful tips in a topology design.	11/10/20
F9	Rescan Button	Rescan will do the same as F1, but will be in the Devices Screen	10/23/20
F10	Tutorial	The Tutorial will teach users how to use and navigate the application.	11/6/20
F11	Filter	The Filter options buttons will help users search for devices.	10/30/20
F12	Device bandwidth	This feature will show the bandwidth of devices.	11/6/20
F13	Exit Button	The Exit Button closes the application.	10/2/20
F14	Show Device Name	Displays name of scanned device. If the device is a network, it will show its SSID. For other	10/2/20

		devices such as phones and computers, it will show its edited name or its default device model name.	
F15	Show MAC Address	Displays MAC addresses for devices on the network.	10/30/20
F16	Show IP Address	Display IP address for devices on the network.	10/30/20
F17	Show Device Type	Display whether the device is a router, phone, tablet, computer, etc.	11/15/20
F18	Sort Devices	Sort devices by type, name, signal strength, etc.	10/30/20
F19	Internet Speed Test	Checks internet speed in Mb/s.	11/15/20
F20	Enable or Disable Device	The enable or disable button will connect or disconnect a device on the network.	11//15/20
F21	Information or Description	Button next to a feature that explains a feature.	11/10/20
F22	Categorize Devices	Should be passive when a general scan is done. Needs to detect categories based on IDs and correctly identify them.	11/15/20
F23	Select Color Theme	Allows the user to select between light/dark themes as well as to select different color themes for either one. Also allows for customizing font sizes and types.	11/05/20
F24	Traceroute	Portray the traffic path from a local machine to a set destination	11/25/20

Functional Requirement Descriptions

ID	Title	Requirement Description
F1	Scan button	The scan button feature will detect all connected devices on a local area network. This feature will be implemented using NMAP: The Network Mapper - Free Security Scanner (found on the main screen).
F2	Navigation	The Navigation Menu Button will allow navigation through the app's

	Menu Button	home, about, help, settings, and user profiles' pages. It will open up a side screen which will hold references to all these other screens via a list.
F3	About	The about button open a summary of what the app is for, available documentation, rate us, open-source libraries, and version information. (found in navbar through settings).
F4	Help	The Help button will show frequently asked questions, contact support/submit ticket (found in nav bar through setting).
F5	Search	The search feature will allow the user to fill in a filters page to search every device on the network for what matches it without going through the GUI.
F6	Nav bar	The Navigation bar will help users navigate through the GUI. Ideally it will be at the bottom of the screen.
F7	Signal Strength	The Signal Strength icon will show each device's internet connection signal strength depending on how far away it is to the router.
F8	GUI	The Graphical User Interface (GUI) will help show how networks visualize a network. The GUI displays all devices, helpful tips in a topology design, navigation, etc. The GUI will help people who are not tech friendly.
F9	Rescan	The rescan button feature will detect all connected devices on a local area network. This feature will be implemented using NMAP: The Network Mapper - Free Security Scanner (found on main screen).
F10	Tutorial	The app will start normally, but all the features will be grayed out and text boxes will show up describing what all of the boxes and the app in general does.
F11	Filter	The filter will be used to search for specific networks by entering the SSID name, provided images of signal strength, security type, with an apply, reset, and close button. (found above in section header)
F12	Device Bandwidth	The Bandwidth of devices will show the amount of data that can be sent from point A to point B in a selected period of time. Will be displayed as bit/s, kbit/s, Mbit/s, Gbit/s, etc. (Found in Network description box)
F13	Exit Button	The Exit Button closes the application. It will be in the settings.

F14	Show Device Name	The Device Name will show the name of the scanned device. If the device is a Network Name, it will be displayed as "Network SSID". For other devices such as phones and computers, this will show the edited device name if it has one, otherwise it will show the device model name.
F15	Show MAC Address	The MAC address will show the unique identifier assigned to a network interface controller, which is used as a network address in communications within a network. (Found in Network description box)
F16	Show IP Address	The show IP address is an essential feature of our application. The IP address will display the IP address of each device connected to the network. The IP address will be displayed when they click on the device to view its information.
F17	Show Device Type	The Show Device Type will display whether the device is a router, phone, tablet, computer, etc. This will help with network visualization by showing which devices are connected to the router and see any suspicious devices that may be connected to the router.
F18	Sort Devices	The Sort Devices will sort devices by type, name, signal strength, etc. This way each device will show in order by a specific type and not randomly.
F19	Internet Speed Test	The Internet Speed Test will show the internet speed of the device. Usually give a value in MB/s.
F20	Enable or Disable Device	The Enable or Disable Device will provide administrative capabilities to allow or restrict devices on the network access to bandwidth usage.
F21	Information or Description	The Information or Description will help explain what that feature does. The point is that users can understand these features and teach them about it. Ideally it will be a button on the side of a feature.
F22	Categorize Devices	The Categorize Devices will detect categories based on IDs and correctly identify them when a general scan is done. This should be passive.
F23	Select Color Theme	Allows the user to select between light/dark themes as well as to select different color themes for either one. Some of the options include changing the color of text fonts for titles and headers, hyperlinks, backgrounds, etc. There will also be the option to change font types and font sizes.

ID	Title	Requirement Description
F24	Trace Route	The Trace Route feature will show in detail the path taken for internet traffic originating from the local machine to the local route then to the Internet Service Provider then finally to its set destination

Non-functional requirements table

ID	Title	Requirement Description
NF1	Java	The back end will be using Java (JDK 8) programming language
NF2	Android Studio	The application will be built on Android Studio (4.1)
NF3	XML	The front end will be using xml to design the application (1.0)
NF4	Android	Android operating system SDK 21 (Lollipop - API level 21)

Non-functional requirements description

We will be using the Android operating system since it's more friendly. Android studio will be used for the IDE and building the project on android SDK 21 (lollipop). Java JDK 8 will be used for the back end with dealing with the networks and XML will be used in the front end for visualizing the networks.

Design Document

System

Tools
+ scan(); void static
+ check_alive(Computer)void static
+ resetLastOctet(String)String static
+ UploadDownload():void static
+ getMacAddress(String)String static
+ getHostName(String)String static
+ checkPingConnection(String)int static

Scan
+ counter:int
+ step:int
+ computersArrayList<String>
+ computerlistArrayList<String>
+ incrementIP(String):String
+ listip(String): void
+ resetLastOctet(String)String
+ scan: void
+ wscan(): void
+ printc(): void

Category
- size: int static
- amount_of_computers:int static
+ devices:Computer[]static
+ all_categories:Category[]static
+ indexFront:int
+ indexBack:int
+ sizeSub(): int
+ Category(int):Constructor
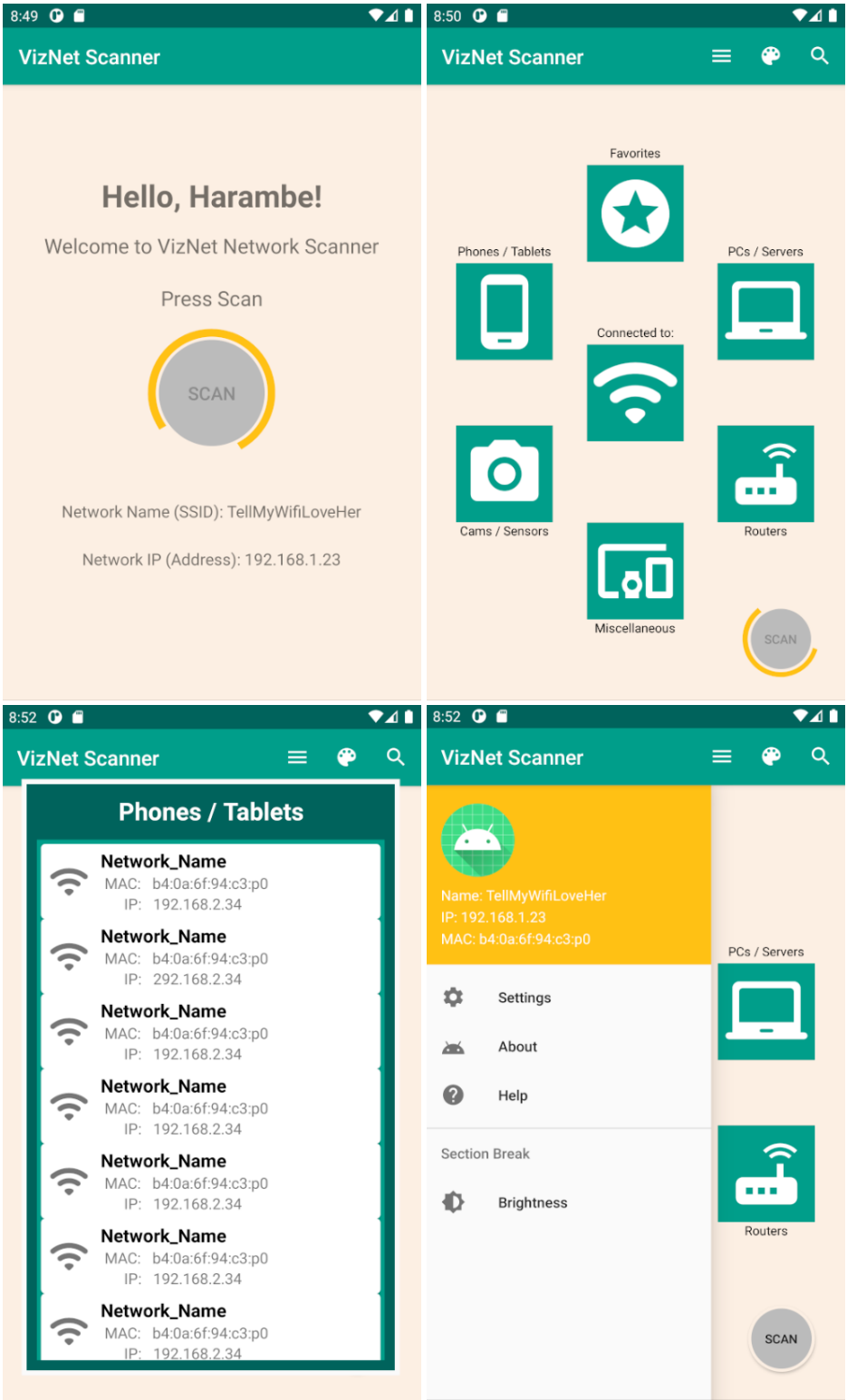+ createSubCategory()void static
+ sortDevices():void static
+ createComputer(String)Computer:static
+ insertDevices(String)void static
+ displayDevices():void
+ deleteDevices():void

Computer
+ NUM_OF_PORTS: int staticfinal
- favorite:boolean
+ alive: boolean
- Localmac: String static
- nickname:String
- hostname:String
- manufacterer_ID:String
- ipv4: String
- mac: String
- tutorial:boolean
+ ports[]:boolean
+ Computer(String):Constructor
+ getConnectionQuality():int
+ Change_name(String):String

Main_Scan
+ newIntent:Intent
+ scanProgressBar:ProgressBar
+ scanButton:Button
- progressVisibility:boolean
onCreate(Bundle)void
+ ButtonScan_OnClick(View):void

Recycle_Adpater
- card_javas:ArrayList<Card_Java>
+ mViewHolder:static
+ mImageView: ImageView
+ mTextView1: TextView
+ mTextView2: TextView
+ mTextView3: TextView
+ mViewHolder(View)
+ Recycle_Adapter(ArrayList<Card_Java>)
+ onCreateViewHolder(ViewGroup,int): mViewHolder
+ onBindViewHolder(mViewHolder,int): void
+ getItemCount():int
+ filterList(ArrayList): void

Popup_device_list
+ ratioW: double
+ ratioH: double
+ newIntent:Intent
+ bundle:Bundle
- mRecyclerView:RecyclerView
- mAdapter:Recycle_Adapter
- mLayoutManagerRecyclerView.LayoutManager
- card_javas: ArrayList<>
- editText: EditText
onCreate(Bundle)void
- filter(String): void
- createList(): void
- buildRecycler(): void
+ Button_DeviceInfo(View):void

Popup_device_info
+ ratioW: double
+ ratioH: double
+ newIntent:Intent
+ bundle:Bundle
# onCreate(Bundle)void
+ Button_Tooltip_IP(View):void

Scanned_Devices
+ button_rescan:Button
+ rescanProgressBar:ProgressBar
- rescanProgressVisibilty:boolean
- navMenuVis:boolean
onCreate(Bundle)void
+ onNavigationItemSelected(MenuItem)boolean
+ openAboutPage()void
+ openHelpPage()void
+ openSettingsPage()void
+ onCreateOptionsMenu(Menu)boolean
+ onOptionsItemSelected(MenuItem)boolean
+ ButtonRescan_OnClick(View)void
+ Button_ConnectedTo(View)void
+ Button_PhonesTablets(View)void
+ Button_PCsServers(View)void
+ Button_CamSensors(View)void
+ Button_Miscellaneous(View)void
+ Button_Routers(View)void

Card_Java
- mImageResource:int
- mText1: String
- mText2: String
- mText3: String
+ Card_java(int, String, String, String)
+ getImageResource():int
+ getText1(): String
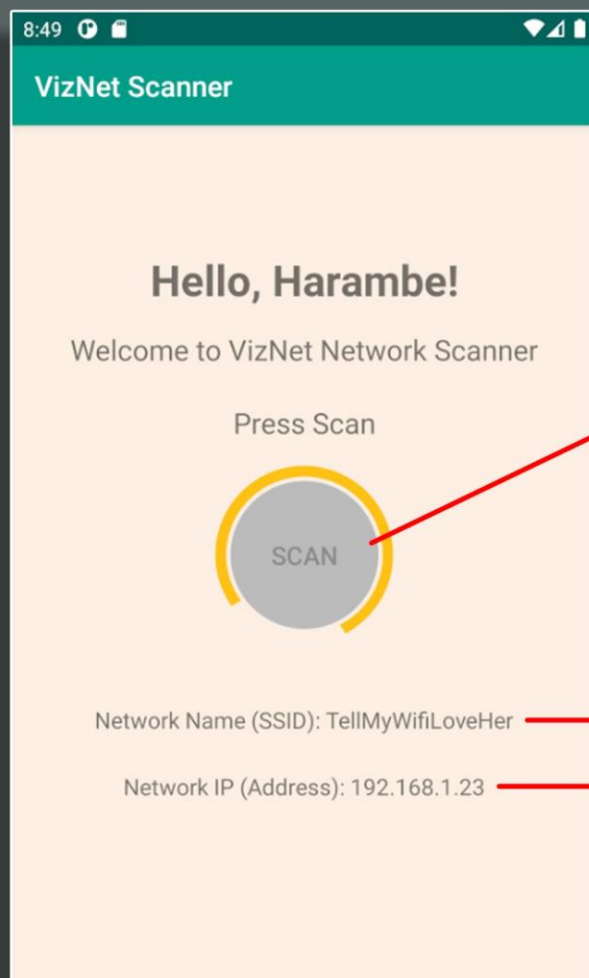+ getText2(): String
+ getText3(): String

Main Scan calls on Tools which then performs a threaded scan using the Scan object. Results from the threads are stored in Category which creates Computer objects which get organized within Category. The Popup_device_list calls upon Category to request for the array of Computers that it needs to display. The Popup_device_list array is generated from inheriting individual Card_Java classes per each device in the respective category and categorized by passing the Card_Java classes to the Recycle_Adapter to create the scrollable array. Each Card_Java is clickable, that information is passed to the Popup_device_info class.

Prototype

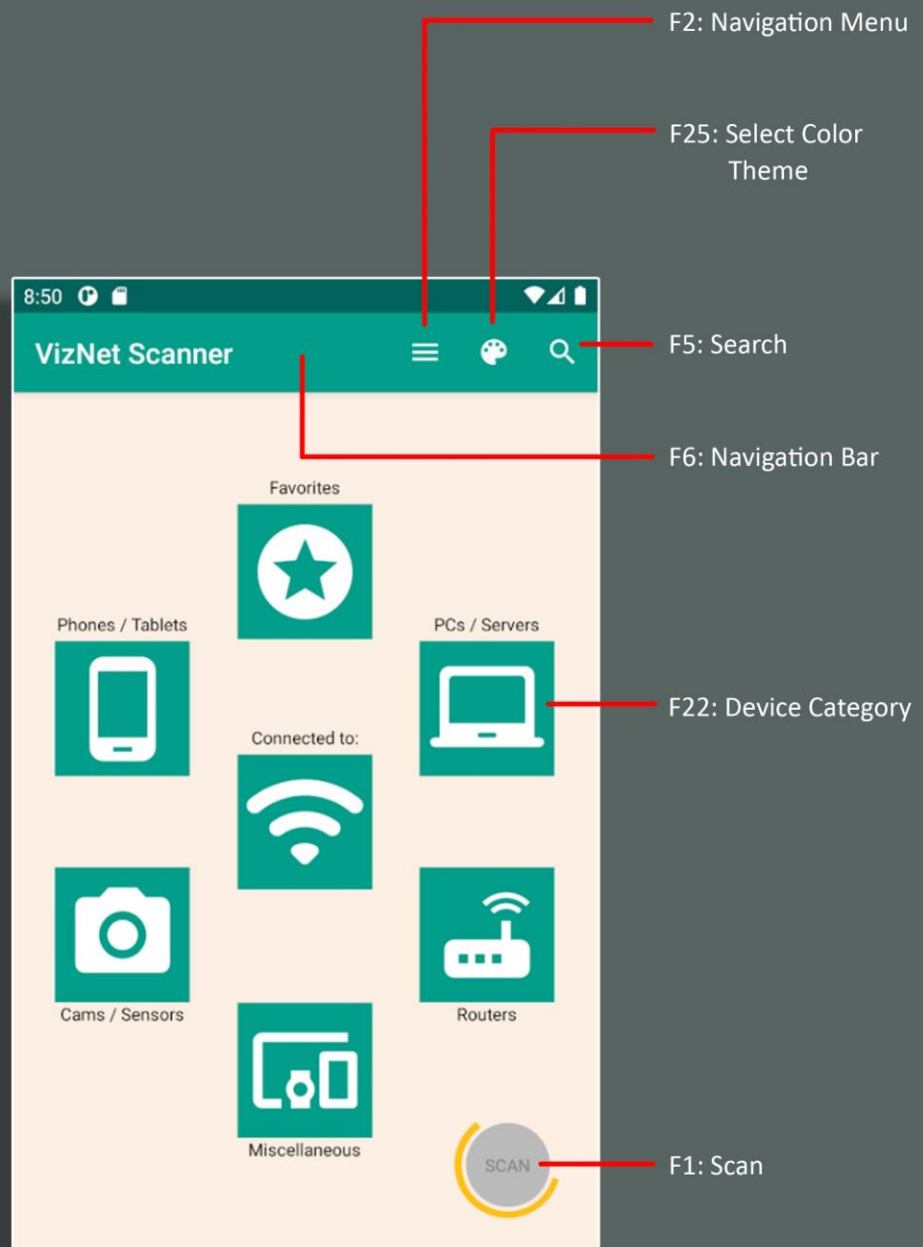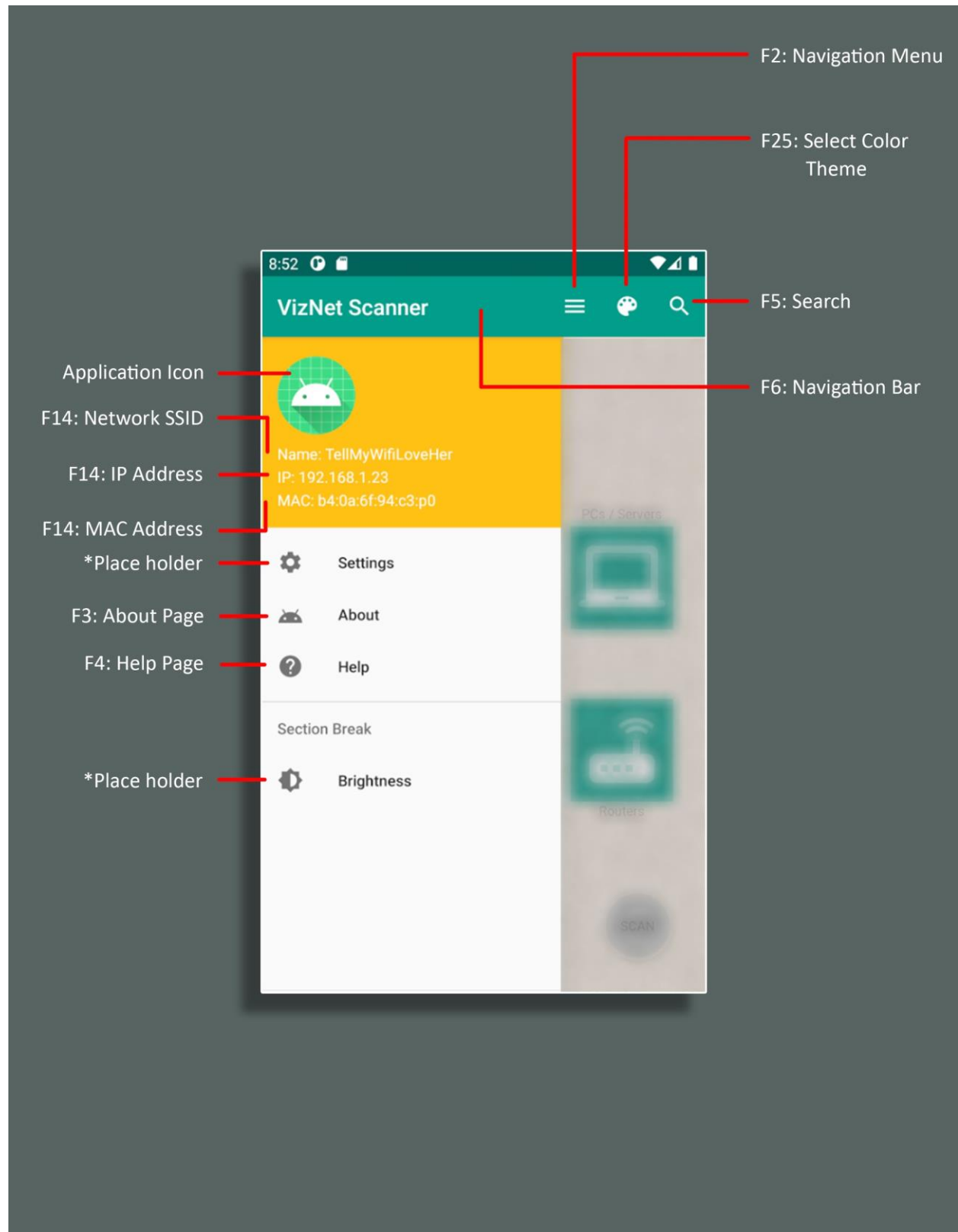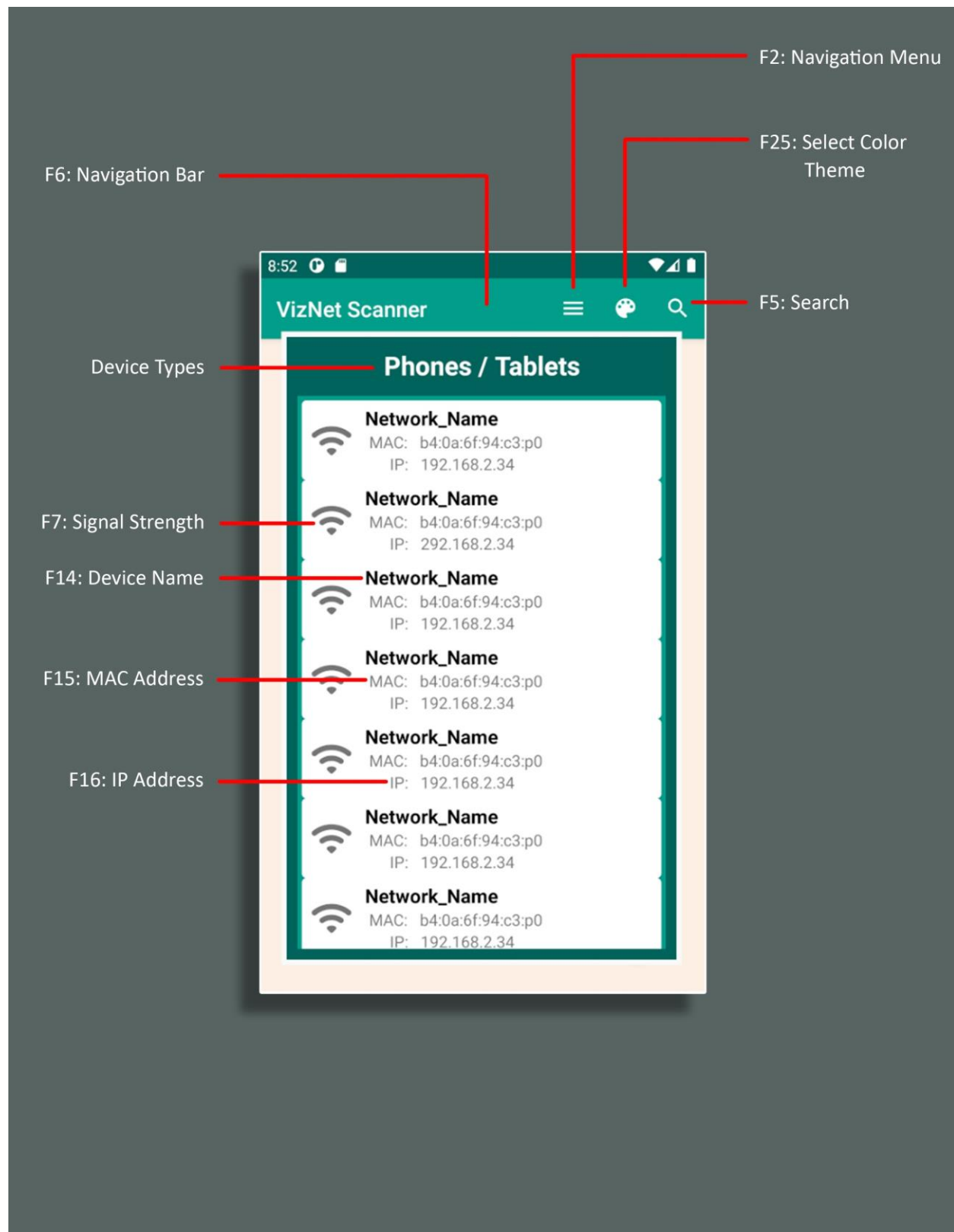8:49

VizNet Scanner

Hello, Harambe!

Welcome to VizNet Network Scanner

Press Scan

SCAN

— F1: Scan Button

Network Name (SSID): TellMyWifiLoveHer —— F14: Network SSID

Network IP (Address): 192.168.1.23 —— F15: Network IP
Address

F2: Navigation Menu

F25: Select Color Theme

F5: Search

F6: Navigation Bar

F22: Device Category

F1: Scan

8:50

VizNet Scanner

Favorites

Phones / Tablets

PCs / Servers

Connected to:

Cams / Sensors

Routers

Miscellaneous

SCAN

F2: Navigation Menu

F25: Select Color Theme

F5: Search

F6: Navigation Bar

Application Icon

F14: Network SSID

F14: IP Address

F14: MAC Address

*Place holder

F3: About Page

F4: Help Page

*Place holder

8:52

VizNet Scanner

Name: TellMyWifiLoveHer
IP: 192.168.1.23
MAC: b4:0a:6f:94:c3:p0

Settings

About

Help

Section Break

Brightness

PCs / Servers

Routers

SCAN

F2: Navigation Menu

F25: Select Color Theme

F6: Navigation Bar

F5: Search

Device Types

F7: Signal Strength

F14: Device Name

F15: MAC Address

F16: IP Address

8:52

VizNet Scanner

Phones / Tablets

Network_Name
MAC: b4:0a:6f:94:c3:p0
IP: 192.168.2.34

Network_Name
MAC: b4:0a:6f:94:c3:p0
IP: 292.168.2.34

Network_Name
MAC: b4:0a:6f:94:c3:p0
IP: 192.168.2.34

Network_Name
MAC: b4:0a:6f:94:c3:p0
IP: 192.168.2.34

Network_Name
MAC: b4:0a:6f:94:c3:p0
IP: 192.168.2.34

Network_Name
MAC: b4:0a:6f:94:c3:p0
IP: 192.168.2.34

Network_Name
MAC: b4:0a:6f:94:c3:p0
IP: 192.168.2.34

The first screenshot is the introductory screen featuring minimal information to ensure users do not feel overwhelmed from the start. This screen will include a greeting for the user including their profile name along with a simple instruction to press the scan button to begin the network scanning process. The user's connected network name and network IP address provide a visual introductory example of the information that will be displayed throughout the application. No toolbar is included on the main activity screen as it is not needed until later use and aids in presenting a simple and clutter-free opening screen to the VizNet Scanner app.

The second screenshot features the bulk of the functions the user will access during the usage of the network scanner. The screen categorizes devices from the original scan based on similarities the user can easily depict. Each category has a visual icon meant to provide quick recognition of the types of device included in the listing. Each category is a clickable image button that will start a pop-up activity displaying that category's device listing sorted based on signal strength which is displayed in screenshot 3. A minimized version of the home screen scan button is placed in the lower right-hand corner to allow users to rescan without having to backtrack to the home screen or completely restarting the application. This screen includes a toolbar with options from right to left including a search button, a color palette button, and a navigation menu button that will open the navigation drawer shown in screenshot 4. The search button will provide the user with a function to search for specific devices that have been detected after a successful scan. The color palette button allows the user to personalize the themes of the application using predefined color schemes.

The third screenshot displays the Phones / Tablets devices that have been scanned after a successful scan and sorted into the scrollable device array ordered by signal strength. Each network device listed has a transparent button sized to the card-view the information is displayed in providing the user with the options to view more detailed information about that specific device. To reduce clutter, network name, network IP, and MAC address are the only information displayed to the user in this device listing.

The fourth screenshot displays the navigation drawer overlaying the network screen upon pressing the menu button on the toolbar. From the navigation menu this user can access application settings to adjust certain features within the app. The about button displays a pop-up about page that features version information, app design information, and a general description of the purpose of VizNet Scanner. The help button will display a common FAQs page to assist the user with any difficulties or questions pertaining to functionality and/or informational issues. The brightness button is a setting we will attempt to implement, that being a version of light and dark mode for users of either light or dark screen preference.

User Study

Interview Questions

Before Seeing Application

- How familiar are you with computer networks?
- Are you the primary person, in your household, that handles internet issues?
- What is your profession or major?

After Seeing Application

- Does the user interface look intimidating to navigate?
- What did you think about how the information is displayed?
- Are the buttons too large? Too small?
- Are all the text sizes visible?
- Did you like the color scheme?
- Closing remarks?

Demographic

Our demographic consisted of English speakers between the ages of 18 and 50 who have varying ranges of network knowledge. We chose English only speakers because the "Help" button (F4) is vital for users who are not well acquainted with networks and the Help screen is only translated in English. We chose the age range of 18 to 50 because we believe that all people entering adulthood should understand networks whereas we put 50 at the limit because once their child turns 18 then he/she can take responsibility for the network issues. The average age for an individual to have a child is 31 according to an article from Stanford medicine (https://med.stanford.edu/news/all-news/2017/08/fathers-of-american-newborns-keep-getting-older.html). We added 18 to that number and rounded it up to 50.

Results

NOTE: all participants in the study gave us permission to disclose their age and gender.

Participant 1 or P1 (age: 24; gender: male)

Participant 1 has semi-familiarity with computer networks. He considers himself a causal but understands the connections of devices to a router. Participant 1 is the primary person who handles internet issues at his house. Participant 1 described the user interface as "clean and easy to understand". He thought the information was displayed in a clear and organized manner and the buttons were of good size but

that the font sizes could potentially be made bigger for clarity. Finally, he described the color scheme as dated. It reminded him of the 1990's. In terms of functionality, P1 said he would like to see his devices signal strength and bandwidth. He said it would be nice to be able to kick devices off the network that use too much bandwidth. Lastly, P1 mentioned that switching between networks would be a cool feature as well.

Participant 2 or P2 (age: 34; gender: male)

P2 is not too familiar with computer networks. He understands what a router is but does not know about IP addresses. He is the primary person who handles network issues, but he only really resets the router or calls his ISP when there are any network issues. He works as a Trader, so he has a lot of familiarity with technology, but not notably familiar.

P2 said that the user interface is very "simple" to understand. He thinks that the information displayed is perfect, but he thinks that it would be more helpful if there was a bit of a tutorial in the beginning to explain things. He says that even though the user interface is easy to use and easy to navigate, he still gets confused by what everything means and what everything does. P2 says that the buttons are a good size, but the design of the Device Screen buttons could stand out a little more. P2 says that the font on the device screen could be a little bigger and that the text in the navigation screen is barely readable because of the white on yellow. P2 really likes the color scheme. P2 says that he would want to see more of a help/explanation for what an IP/MAC address is so that he could use the app to its fullest potential. P2 says that he could see himself using this app in the future.

Participant P3 (age: 40; gender: female)

P3 is somewhat familiar with computer networks. She works as a Business Process Analyst, so she has had a lot of encounters with networks. She is also the primary/co-primary person who takes care of network problems. She says her and her boyfriend have about the same amount of knowledge when it comes to networks.
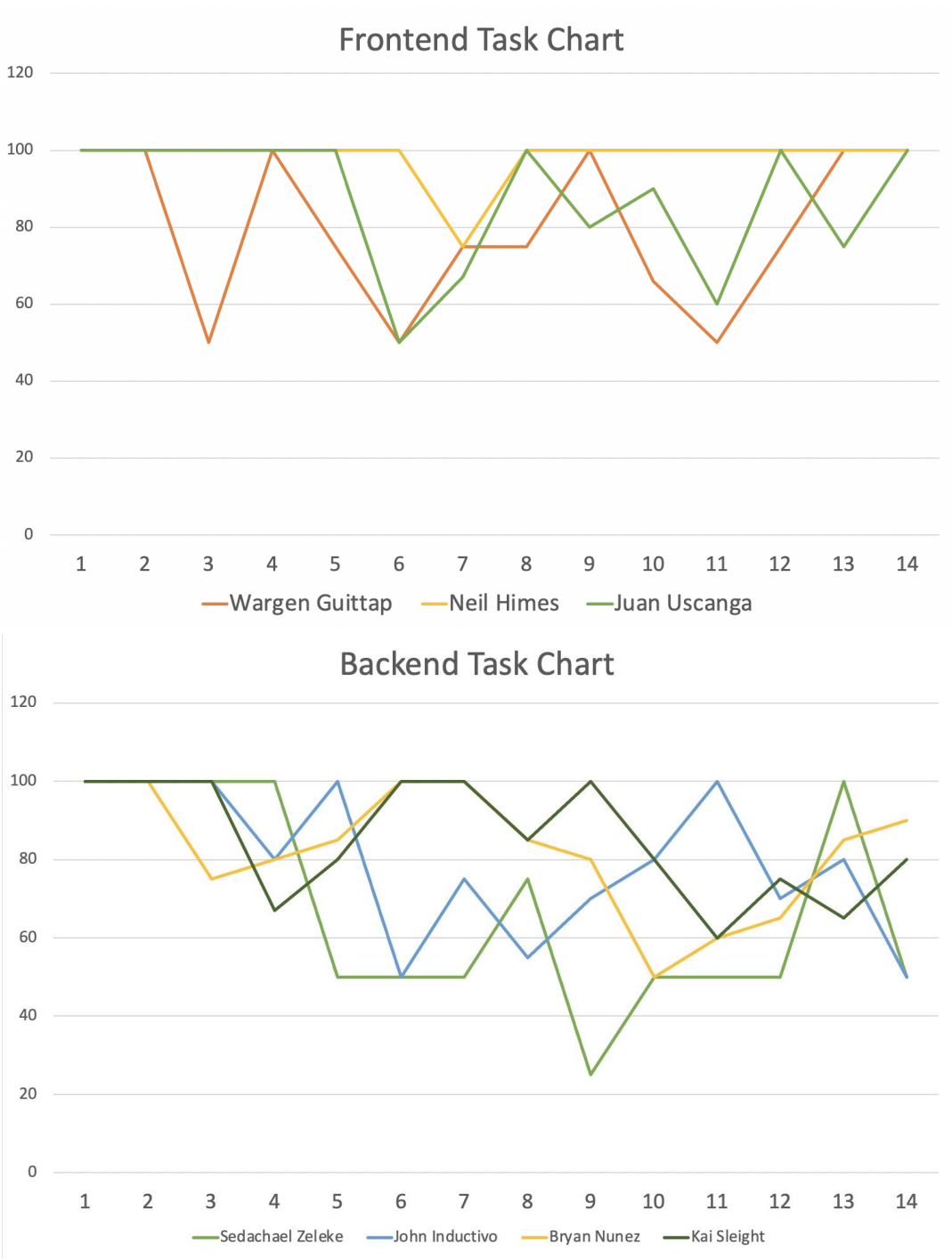
P3 says that the interface is a little intimidating. The scan and network screens are good, but the popup for the devices on the network was a little jarring. She said there was a lot of information that she did not really understand. She said that it would be helpful to have a Help section that will explain what IP and MAC

addresses are. P3 said that the buttons are a good size, but the buttons on the top bar were a tad too small. She liked the color scheme and said it makes it easy for her to see what is going on. She said that it would be nice to have a simple way to see what devices are on her network without having too much extra information. She said that it would be useful to know what her internet speed is and how much bandwidth each device takes up.

Based on all three interviews, a major theme that we got out of it was that our screens were simple and easy to understand. This was good feedback for us since we are aiming towards making the navigation screen as simple as possible. Additionally, another major theme we got from our interviews was that we could improve the font sizes and buttons of the application. All three interviewees seem to agree that we could improve our application by increasing the font size and increasing the size of the buttons on the top right side of the screen. The sizes of these fonts and buttons are definitely something that the front-end team will have to experiment more with. In terms of color schemes, we noticed that our interviewees had different opinions. Two of them liked the color scheme while the other one thought it felt dated and dull. A solution we discussed was to implement a custom color scheme to allow the user to choose which color background they would like. Another theme we noticed was that users would like to have more functionality. This functionality included having speed tests and monitoring their devices bandwidth. Being able to hear this feedback gives us high priority in making sure we can implement these functionalities. Lastly, we had two users describe that we had too much information overload. We will need to work on improving the way we display device information so that we do not surprise our users with a screen full of information.

Team Report

Team Tasks Charts

Frontend Task Chart

Wargen Guittap — Neil Himes — Juan Uscanga

Backend Task Chart

Sedachael Zeleke — John Inductivo — Bryan Nunez — Kai Sleight

Wargen Guittap

Week 1 [100%]

- Research journal articles related to our project and write up the journal article portion of the motivation section.

Week 2 [100%]

- Research user interface design methods for the front end of the application
- Work on the functional requirements table

Week 3 [50%]

- Android Studio Tutorial Course
- Assigning tasks to group members

Week 4 [100%]

- Android Studio Tutorial Course

Week 5 [75%]

- Creating Navigation Menu
- Assigning tasks to group members

Week 6 [50%]

- Created list of questions to ask demographic
- Connecting Navigation Menu items to pages

Week 7 [75%]

- Connecting Navigation Menu items to pages
- Retrieved responses for user study

Week 8 [75%]

- Revise Design Document II
- Taking the New Prototype Pictures
- Manage Team Reporting
- Finish About Page

Week 9 [100%]

- Create Branding for Application
- Work on Presentation
- Finish Help Page

- Add branding onto the application
- Work on presentation
- Fix presentation based on Stefik Criticisms

Week 11 [50%]

- Decide on color palette for the application
- Create a Better Motivation based on Stefik Criticisms

Week 12 [75%]

- Stylize the About and Help pages
- Create a Better Motivation based on Stefik Criticisms
- Fix design document 3 based on feedback

Week 13 [100%]

- Create Design for Project Poster
- Add motivation section to Project Poster
- Fix design document 3 based on feedback

Week 14 [100%]

- Add new motivation to Design Document 3
- Add new motivation to the presentation

Neil Himes

Week 1 [100%]

- Research existing applications and write up the competitive analysis portion in the motivation section of design document 1. (complete)
- Work on the initial low fidelity prototype of our application (complete)

Week 2 [100%]

- Draw up initial low-fidelity prototype. (complete)
- Team member time reporting chart. (complete)
- Upload final low-fidelity prototype. (complete)

Week 3 [100%]

- Watch Android Studio introduction tutorial video - 5 hours. (complete)
- Create projects following the tutorial for syntax learning. (complete)
- Research android studio activity videos. (complete)

Week 4 [100%]

- Create network screen topology with image buttons. (complete)
- Research android studio navigation view functionality. (complete)
- Add drawables for app icons. (complete)

Week 5 [100%]

- Integrate network screen into front-end project files. (complete)
- Implement a down-sized scan button into network activity. (complete)
- Research implementation of navigation drawer into app. (complete)

Week 6 [100%]

- Implement a navigation drawer into the network screen. (scrapped)
- Create a navigation pane pop-up menu. (complete)
- Add icons onto the toolbar. (complete)
- Research array list creation and projection. (complete)

Week 7 [75%]

- Create array list and project to pop-up window. (complete)
- Add transparent clickable button for array list. (complete)
- Create functionality for color palette changing. (scrapped)
- Research search feature functionality. (completed)

Week 8 [100%]

- Implemented search functionality in array list. (complete)
- Remove place-holder icons. (complete)

Week 9 [100%]

- Research implementing light/dark mode functionality. (complete)
- Redesign pop-up color scheme. (complete)

Week 10 [100%]

- Implement light/dark mode. (scrapped)
- Remove place-holders in the navigation menu. (complete)

Week 11 [100%]

- Research network terminology for education. (complete)

Week 12 [100%]

- Create Java classes for each category array pop-up. (complete)
- Research style change functionality. (complete)

- Reformat text view alignment of pop-up xmls. (complete)
- Add educational information into navigation menu tabs. (complete)

Week 14 [100%]

- Added multiple colors into colors xml file. (complete)
- Recolor the entire application. (complete)

John Inductivo

Week 1 [100%]

- Write up the executive summary and abstract for design document 1. (complete)
- Research existing applications and write up the competitive analysis portion in the motivation section of design document 1. (complete)

Week 2 [100%]

- Research how to scan for devices on a network (complete)
- Work on the UML diagram for the application (complete)

Week 3 [100%]

- Research and learn about Android Studio
- Implement projects in order to become familiar with Android Studio Environment

Week 4 [80%]

- Research how to retrieve information from android devices (ongoing)
- Research and implement libraries to be useful for application (ongoing)

Week 5 [100%]

- Determine how to acquire machine information
- Research into implementing libraries for scan button
- Decided on using sockets for communication with devices

Week 6 [75%]

- Research in trying to pass information between front and back end (ongoing)

Week 7 [75%]

- Implement method to remove devices on a desired list (ongoing)

- Implement method to change nicknames (ongoing, need to implement onto other classes)
- Create related questions to ask end users (complete)

- Worked on changing our UML Design
- Continue to work on Category and Tools Class for application (ongoing)

- Research for Concurrency in Java

- Research for Concurrency in Java
- Test Concurrency for scanning IP

- Fix errors for Concurrency and Threading
- Initial work for connecting frontend and backend

- Create test Environment for Features
- Overview of frontend and backend

- Research and test for TraceRoute [ongoing]

- Research and test for TraceRoute [ongoing]
- Implement TraceRoute in test environment
- Research of new future features to be implemented

Bryan Nunez

- Research journal articles related to our project and write up the journal article portion of the motivation section. (complete)

- Research how to scan for devices on a network (complete)
- Work on the functional requirements table (complete)

Week 3 [75%]

- Android Studio course tutorial
- Research and refresh memory on the basics of networking

Week 4 [80%]

- Learn about networking fundamentals and the important concepts needed for our application.
- Start/Continue research on how to find network information in Java. This included importing nmap to android studio and finding what libraries we can use.

Week 5 [85%]

- Attempted alternative methods for WIFI scanning using the libraries in Java.
- Discussed the pros and cons between these alternatives and using sockets with the rest of the group.

Week 6 [100%]

- Added in the port iteration code so that every IP address can attempt to establish a connection to the specified ports in the ports array
- Implemented a method to increase the IP address based on the value of the last octet.
- Verified that our scan was iterating properly through the ip addresses and ports by recording the output to our logcat files.

Week 7 [100%]

- Added in the class - Filters - to the project file from our UML
- Brainstorm questions for the user study and conducted an interview with a participant
- Contributed towards adding in the relationship and multiplicity between our classes in the UML based on the feedback from the TA.

Week 8 [85%]

- Implement code needed for the user's upload/download phone speed.
- Research which method is best to upload/download an image from a server.

Week 9 [80%]

- Completed the code needed for the user's upload/download speed
- Added the code from the background thread as a method in the main thread.

Week 10 [50%]

● Begin researching alternative motivations based on the presentation feedback.

Week 11 [60%]

● Research potential vulnerabilities that small businesses go through.
● Continue researching key design features based on our new motivation.

Week 12 [65%]

● Research solutions towards resolving the manufacturer ID.
● Tested which manufacturer ID code works best for our application.

Week 13 [85%]

● Merge mac vendor code to the front end.

Week 14 [90%]

● Resolved manufacturer ID for a device by using macvendor's api.
● Work on integrating the mac vendor api code to the actual scan.

Kai Sleight

Week 1 [100%]

● Write up the executive summary and abstract for design document 1. (complete)

Week 2 [100%]

● Research how to scan for devices on a network (complete)
● Research existing applications and write up the competitive analysis portion in the motivation section of design document 1. (complete)
● Work on the UML Diagram for the application (complete)

Week 3 [100%]

● Research and familiarize self with Android studio and how the different components work (Complete)

Week 4 [67%]

● Attempt to integrate and use an Android's terminal for many of the backend (incomplete / later abandoned)
● Research about permissions and making requests for permissions with Android (complete)

- Create classes and refine structure from previous UML (ongoing)

Week 5 [80%]

- Attempt to integrate and use an Android's terminal for many of the backend (abandoned)
- Work on alternatives for mimicking nmap abilities with libraries for Java (ongoing)
- Create and use Socket library to build a Proof of Concept for being able to scan an IP and determine which port is opened or closed (complete)
- Refine structure from previous UML (ongoing)

Week 6 [100%]

- Added in parts of Category to be able to create new devices for results found from Tools broad Scan (complete)
- Rework Computer and Category specifically to be able to interact and keep track of every object easier (complete/ongoing)
- Implement two functions in Category to handle creating and managing new Computers picked up by the Scan Tool (complete)
- Refine structure from previous UML (ongoing)

Week 7 [100%]

- Refine two functions in Category to handle creating and managing new Computers picked up by the Scan Tool according to new format (complete)
- Generate questions for survey (complete)
- Refine structure from previous UML (ongoing)

Week 8 [85%]

- Implement use of main list with Sub-Categories interacting with the main Category (in-progress)
- Begin populating the Filters class and changing Tools to take Filters into consideration (ongoing)
- Research how we can grab Mac Address and Hostname of remote devices through Java Libraries or alternative (ongoing)

Week 9 [100%]

- Merge backend into the front end

Week 10 [80%]

- Add functionality to get hostname and Mac address
- Change scan to work with Ping instead of sockets [ongoing]

Week 11 [60%]

- Add functionality to get hostname and Mac address
- Extend Category to begin working with front end [ongoing]
- Try to get scan to work synchronously through threading [ongoing]

Week 12 [75%]

- Extend Category to begin working with front end [ongoing]
- Try to get scan to work synchronously through threading [abandoned]
- Recreate UML for backend

Week 13 [65%]

- Extend Category to begin working with front end [ongoing]
- Reintroduced socket scanning to detect ports
- Setup environment to run demo on
- Recreate UML for backend [complete]

Week 14 [80%]

- Extend Category to begin working with front end [ongoing]
- Reintroduced socket scanning to detect ports
- Setup environment to run demo on

Juan Uscanga

Week 1 [100%]

- Research existing applications and write up the competitive analysis portion in the motivation section of Design Document I [complete]

Week 2 [100%]

- Draw up initial low-fidelity prototype [complete]
- Draw up final low-fidelity prototype [complete]

Week 3 [100%]

- Watch Android Studio introduction video - up until 1hr 30min [complete]
- Follow video to create tutorial application [complete]

Week 4 [100%]

- Continue watching Android Studio introduction video [complete]
- Continue to add to tutorial application based on video [complete]

Week 5 [100%]

- Create "Scan Screen" for prototype [complete]

Week 6 [50%]

- Integrate "Scan Screen" into front end project files [in progress]
- Create "Device List" pop up screen [complete]

Week 7 [67%]

- Add to list of interview questions [complete]
- Create "Device Info" pop up screen [complete]
- Refine app design (colors, fonts, layout, etc) [complete]

Week 8 [100%]

- Finish "Device Info" pop up screen [complete]
- Add application screen shots to Design Document II and revise document [complete]

Week 9 [80%]

- Research how to implement multiple color themes into application [complete]

Week 10 [90%]

- Implement support for multiple color themes for the app [complete]
- Work on mock up presentation PowerPoint [complete]

Week 11 [60%]

- Create icons for information tool tips [complete]
- Implement button for information tool tips on device information screen [complete]

Week 12 [100%]

- Implement information tooltip pop up screen for IP addresses [complete]
- Refine information tooltip pop up screen [complete]

Week 13 [75%]

- Implement information tooltip pop up screen for other information [complete]
- Polish up information tooltip pop up screens [complete]

Week 14 [100%]

- Work on the final theme/color of our application [complete]

- Work on Design Document III [complete]

Sedachael Zeleke

Week 1 [100%]

- Research journal articles related to our project and write up the journal article portion of the motivation section. (complete)

Week 2 [100%]

- Research how to scan for devices on a network (complete)
- Work on the functional and nonfunctional requirements table (complete)

Week 3 [100%]

- Android Studio Tutorial Course (complete)
- Deciding tasks to group members (complete)
- Research and refresh memory on the basics of networking (complete)

Week 4 [100%]

- Finishing Android Studio Tutorial Course (complete)

Week 5 [50%]

- Learning about networks (complete)

Week 6 [50%]

- Trying to find IP addresses of certain devices (complete)

Week 7 [50%]

- Found the local MAC address of an Android device. (complete)
- Find MAC address of devices on the network. (ongoing)

Week 8 [75%]

- Was able to learn more about networking.

Week 9 [25%]

- Implementing on how to categorize devices.

Week 10 [50%]

- Continue to work on Category and Tools Class for application (ongoing)

Week 11 [50%]

- Try to implement traceroute. (ongoing)

Week 12 [50%]
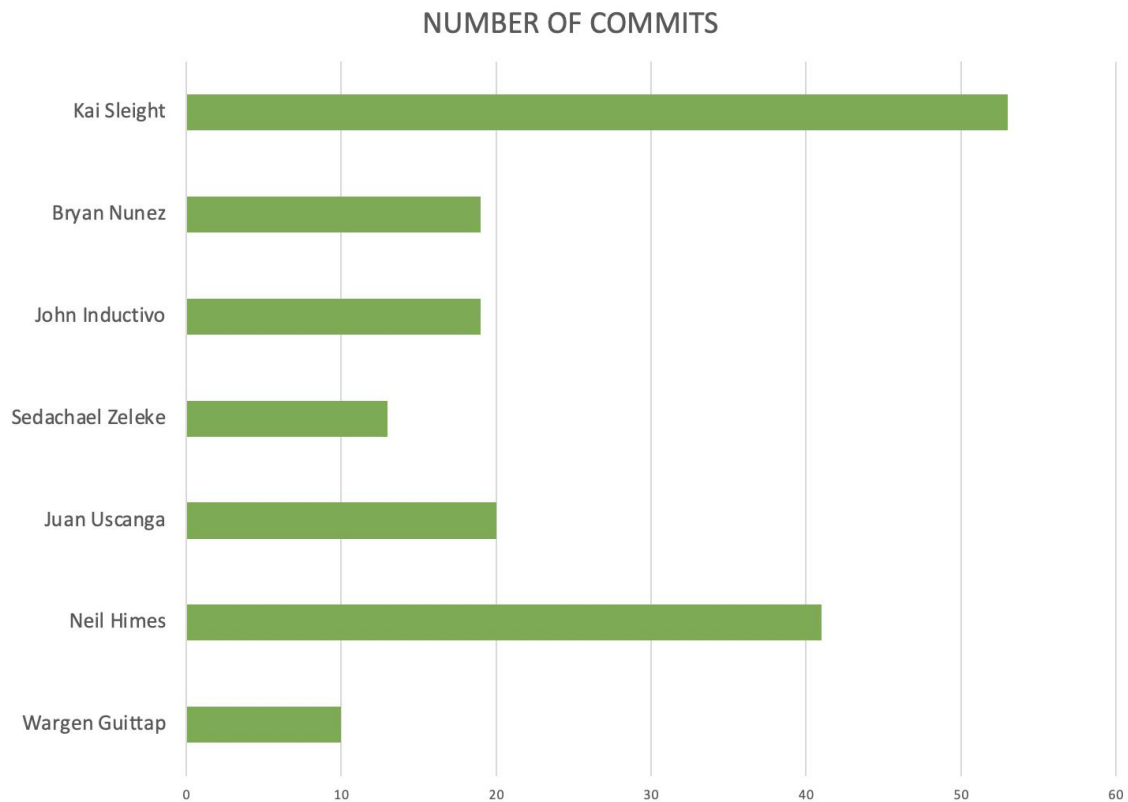
- Try to implement traceroute. (ongoing)

Week 13 [100%]

- Able to get threading working on project making the scan 4x faster (complete)

Week 14 [50%]

- Working on the finishing touches on the project. (ongoing)

Team Commit Chart

NUMBER OF COMMITS

Team Member	Number of Commits (approx.)
Kai Sleight	53
Bryan Nunez	19
John Inductivo	19
Sedachael Zeleke	13
Juan Uscanga	20
Neil Himes	41
Wargen Guittap	10

Success Report

Wargen Guittap: Created Navigation Menu, Made Navigation Menu items operable, Designed and created the main logo of the application, Helped Front End Team with deciding on the color palette for the application.

Neil Himes: Created Network Screen with functional scan button and image buttons for each category of devices. Created scrollable recycler view with card-view to display the generated array of network devices. Created toolbar menu with functional search, color palette, and navigation menu buttons. Designed the navigation menu drawer.

John Inductivo: Implement the function to change device nicknames used for personal preference. Successfully test Traceroute in a test environment to be implemented as a feature and be able to determine the path of the traffic.

Bryan Nunez: Put the scan button in a loop to iterate through each ip address and port number in the network. Wrote the method needed to increase the ip address and a helper method to reset the last octet of the ip address. Started adding the main attributes and

operations needed for the Filter class. Wrote the methods needed to resolve the user's upload/download speed. Implemented the manufacturer ID for devices on the network.

Kai Sleight: Implement and refine each class to be able to interact with each other and keep track of every device without losing any objects. Implemented insertion into the main list in Category and the creation of the Computer objects within Category.

Juan Uscanga: Created initial scan screen with a functional scan button which will scan the network which the device is currently connected to. This page will also display key information about that network. Once the following screens had been created, added functionality to transition between a current screen and the correct follow up screen for a given action. Implemented a pop-up window to display a list of selected devices for a chosen category with key information for each device (i.e. device name, signal strength, IP and MAC addresses). Refined app design colors, fonts, and organization for a smoother user experience.

Sedachael Zeleke: Was able to find the local Mac address of the Android device. Then was able to add concurrency to the project. This was very important since each ping took 1 second so scanning 100 IPs would take over 100 seconds, but with 4 threads this takes 25 seconds now.

Conclusion

A network visualizer must portray detailed network information using technical representation while also being easy to read and use. Many applications have advanced features which give their users customization through its platform, however not many cater to users that are not tech savvy. Thus, the goal for our application is to relay technical network information to our users in as easily and digestible a way as possible. In the long run, our app will grow with our users by having various options, features, and customization options implemented and readily available to users, so that they can further manage their networks and keep them secure.

References

[1] https://www.hiscox.com/documents/2019-Hiscox-Cyber-Readiness-Report.pdf
[2] https://www.keepersecurity.com/assets/pdf/Keeper-2018-Ponemon-Report.pdf
[3] https://www.accenture.com/_acnmedia/PDF-96/Accenture-2019-Cost-of-Cybercrime-Study-Final.pdf#zoom=50
[4] https://www.researchgate.net/publication/324918373_Rethinking_Home_Network_Security
[5] https://www-sciencedirect-com.ezproxy.library.unlv.edu/science/article/pii/S0167404817301839#s0065