|  |  |  |  |
| --- | --- | --- | --- |
| *LET’S measure 1.5 meter* | | | |
| Man |  | | Woman |
| 13-12-2020 | | Adnan Al Jawabra | |

|  |  |
| --- | --- |
|  | FONTYS UNIVERSITY OF APPLIED SCIENCES  HBO-ICT: English Stream |

|  |  |
| --- | --- |
| **Data student:** | |
| Family name, initials: | Al Jawabra, A |
| Student number: | 3209873 |
| Assignment period: (from – till) | 01-09-2020   13-12-2020 |
| **Final Report:** | |
| Title: | Let’s measure 1.5 meter |
| Version: | 2.0 |
| Date: | 13-12-2020 |

Table of Contents

[Summary 4](#_heading=h.gjdgxs)

[Glossary 5](#_heading=h.30j0zll)

[Chapter 1: Introduction 6](#_heading=h.1fob9te)

[Chapter 2: Problem description 7](#_heading=h.3znysh7)

[What is the problem? 7](#_heading=h.2et92p0)

[Current situation 7](#_heading=h.tyjcwt)

[Objectives 7](#_heading=h.3dy6vkm)

[Approach & Activities 7](#_heading=h.1t3h5sf)

[Approach 7](#_heading=h.4d34og8)

[Activities 7](#_heading=h.2s8eyo1)

[How can this approach be helpful? 8](#_heading=h.17dp8vu)

[Chapter 3: Process and results 9](#_heading=h.3rdcrjn)

[What are the already existing technologies that smartphones use to detect objects at a certain distance? 9](#_heading=h.26in1rg)

[Would it be possible to measure distance between smartphones (devices)? 12](#_heading=h.lnxbz9)

[Does all smartphones os’s support the previously founded technologies? 13](#_heading=h.35nkun2)

[How can these technologies be used in mobile apps; what programming languages could be used? 14](#_heading=h.1ksv4uv)

[Chapter 4: Discussion 18](#_heading=h.44sinio)

[Chapter 5: Conclusion(s) and recommendation(s) 20](#_heading=h.2jxsxqh)

[Conclusion: 20](#_heading=h.z337ya)

[Recommendations: 20](#_heading=h.3j2qqm3)

[Chapter 6: Reflection 21](#_heading=h.1y810tw)

[References 22](#_heading=h.4i7ojhp)

[Appendixes 23](#_heading=h.2xcytpi)

[Appendix A: T&H Subject Description 23](#_heading=h.1ci93xb)

[Appendix B: Prototype code & algorithm 26](#_heading=h.3whwml4)

# Summary

Technology has been found and improved to help in simplifying human life. making activities easier to achieve. And it should not be surprising if we say that it helps humans to survive. Surviving from death is something that all humans try to avoid. But in 2020 it seems a bit unavoidable for some of us since that monster called Covid-19 got into our life. It is important that technology helps us in fighting to survive since it was found for this reason.

In this research paper: we would like to make use of technology. Aiming to achieve a good solution that helps us in applying the rules followed in the process of fighting the Covid-19 where not all of us are good at remembering each rule and applying it correctly. One of the most important rules that all of us should follow is keeping 1.5 meters distance according to the Dutch government rules; having a social distance has a big advantage in avoiding Covid-19 from getting separated.

The technology should help us in using our smartphones in applying the above-mentioned rule correctly. Here we come to the point of how smartphones help us in keeping 1.5 meters from other people. Are we able to use our smartphone to get alerted when we break the rule of 1.5-meter distance? The goal of answering those questions is to have a mobile app that measures the distance between two devices and alerts us whenever we get in the range of 1.5 m with another device.

To achieve this goal: having a mobile application that alerts us on a certain distance range. We should get to know what are currently the available technology that we can use to measure the distance between smartphones. Also, it is key important to know if these possibilities are a solution for all mobile platforms since each platform has its own implementations. In the end, we should be able to understand the possible ways to implement those technologies and make use of them to build a functional prototype that measures the distance and alerts us when we get in the distance of 1.5 m with another device.

After months of researching and analyzing the findings. We achieved a way to let a mobile application scan for close devices and calculate the distance between the current and the found device. For that, we made use of the Bluetooth devices available on our smartphones in a natural way nowadays.

The Bluetooth device can discover other Bluetooth devices at a certain distance, but that for sure much bigger than 1.5 meters. Therefore, we used the RSSI (the received signal strength) and this is a part of each device that produces radio waves. When a mobile application Bluetooth device discovers another device, it will be able to read this RSSI. This is a variable that is changing depending on how far the device is and some other factors e.g., battery state. To estimate and calculate the real distance of that RSSI we made use of a well familiar formula that is used in the indoor positioning system. More about this formula in chapter 3.

At the end of this research phase, we were able to develop a prototype in the form of a mobile application that scans for Bluetooth devices and calculates the distance of the discovered device. When the calculation results of having a device in the range of 1.5 meters the mobile application will display a notification and vibrate to alert the user. Having a mobile application that measures the distance between two different devices, is possible. Smartphones do have and support technologies that could help us in achieving estimated real distance measurement.

# Glossary

* **Advertising device**: the device that is broadcasting data to other devices.
* **Bluetooth**: Bluetooth is a wireless technology standard used for exchanging data between fixed and mobile devices over short distances.
* **Covid-19**: Coronavirus disease (COVID-19) is an infectious disease caused by a newly discovered coronavirus.
* **GPS**: Global Positioning System.
* **Instagram**: Social media platform.
* **Measured power**: the measured RSSI value on one-meter distance from the advertising device.
* **NOS**: The Nederlandse Omroep Stichting is one of the broadcasting organisations making up the Netherlands Public Broadcasting system.
* **RVIM**: National Institute for Health and Environment.
* **RSSI**: Received signal strength indication.
* **WIFI**: wireless networking technology.
* **Xamarin forms**: mobile application framework for building user interfaces and it is a cross-platform.

# Chapter 1: Introduction

A few weeks ago, while I was exploring my Instagram account, I came across the NOS stories account. In this news feed, they were presenting the results of the latest research that was made by RVIM. The research covers about 60.000 people. That research is meant to find out if Dutch people are showing full commitments to the Covid-19 rules. The research findings show that people do not fully apply the rules. We are all struggling in this tough time when it comes to remembering all these rules and applying them. On a personal level, I also have trouble with applying the 1.5 m distance rule.

In this report, I want to find out how we can benefit from our smartphones and make them able to measure 1.5 distance and alert us when we fail to keep this distance.

This document contains different chapters. Chapter 2 will describe the problem in depth. Chapter 3 will show the process and list the results of the different phases and activities. The discussion chapter will interpret and discuss the meaning of the found results. Further, the document contains conclusions and recommendations. At the end of this document, you can find the reflection chapter where the reporter will interpret the research process and evaluate it to see if that was enough to produce good results and you will find all related appendices and references.

# Chapter 2: Problem description

## What is the problem?

How do smartphones help in keeping 1.5 m distance?

We are living in a time where getting close to people could be the reason that will be registered on the death certificate. Covid-19 is still spreading, one of the important rules to stay safe is to keep a social distance from other people. That is hard because human nature is exactly the opposite.

## Current situation

The concept of distance measurement exists in the indoor positioning system. Usually, it used to determine an object located inside a building. There is no software that is designed and developed to measure the distance between two devices and alert the users if they enter the distance range that is considered a danger zone.

## Objectives

To help with solving the above-mentioned problem, a research and a report will be conducted to show whether smartphones can help people with keeping a certain distance between each other. The main goal of this paper is to achieve a prototype that will apply the findings of this research to prove the validity of the findings and save people's life of Covid-19.

## Approach & Activities

### Approach

After justifying the main problem, it is convenient to try answering it by dividing the main question into small pieces. Therefore, the following list represents a sub-question, these questions should help in unraveling the problem by breaking it down and start solving it step by step.

* What are the already existing technologies that smartphones use to detect objects at a certain distance?
* Does all smartphones os’s (operating system) support the previously founded technologies?
* Would it be possible to measure distance between smartphones (devices)?
* How can these technologies be used in mobile apps; what programming languages could be used?

### Activities

* What are the already existing technologies that smartphones use to detect objects at a certain distance?
  1. Finding methods that the smartphone is already using to detect objects around it.
     1. **Method**: Library -> Literature study
     2. **Result**: Overview about present methods and ways to detect a physical object using smartphone
  2. Examine if this technology can tell what the object is about (person, tree, etc.)
     1. **Method**: Workshop -> prototyping
     2. **Result**: Knowing if it is possible or not to define the detected object type to ensure that the user will be alerted only about the required ones.
* Would it be possible to measure distance between smartphones (devices)?
  1. Figure out if smartphones recognize each other by sending signals (sound, Bluetooth, location etc.)
     1. **Method**: Library -> Best good and bad practices
     2. **Result**: Knowing the already existing solutions and seeing the best and efficient way that smartphones can recognize each other.
* Does all smartphones os’s support the previously founded technologies?
  1. find out if mobile platforms e.g. (Android and IOS) support this technology?
     1. **Method**: Library -> Literature study
     2. **Result**: Determining the platforms that the implementation of the technology can run on and defining the platforms we should target
  2. Get to know the OS versions that are supported for these technologies.
     1. **Method**: Library -> Literature study
     2. **Result**: To ensure the targeted OS versions that can support the functionality.
* How can these technologies be used in mobile apps; what programming languages could be used?
  1. Find out the programming language(s) or framework(s) that helps in achieving the measurement.
     1. **Method**: Library -> Literature study
     2. **Result**: What are the available coding environments to implement a prototype for this problem
  2. Try to build a mobile app (prototype) that implements the functionality.
     1. **Method**: Workshop -> prototyping
     2. **Result**: Making sure that the function can be implemented and prove the research results.
  3. Testing if the required function is behaving as wanted.
     1. **Method**: Lab -> Component test
     2. **Result**: Testing the implemented solution and making sure parts of the prototype are working as expected.

### How can this approach be helpful?

During the writing of this research paper, there will be three different strategies used (Library, Workshop, and Lab). Each one will serve the research paper in the way it meant to be. The Library will help in collecting general information and spotting good and bad practices. The workshop will be used to develop a prototype that implements the findings of this research paper. After all, the Lab strategy will help us in finding if the implemented solution is identical to the theory of the research.

By answering the main question, we will be able to build and run a mobile application that will alert us if we break the rule, says KEEP 1,5 meters.

# Chapter 3: Process and results

This chapter is about to explain what is done, what the questions were asked, how the Process went, and what are the results.

For simplicity, this chapter will be divided into subsections where each section will cover the activities that previously mentioned in chapter 2. Each subsection will start with the question that needs to be answered and refer to each activity related to that question and describe the process went through and what are the results in addition to the questions asked to achieve the results.

## What are the already existing technologies that smartphones use to detect objects at a certain distance?

#### Activity: Finding methods that the smartphone is already using to detect objects around it

Process: to be able to answer this question, searching the internet was a good idea. Using the following phrase “what are technologies that smartphones offer to detect objects” there was some good research that pointed out some of what are the possibilities to detect objects from a picture or video or using the live cam. However, that was not meant by the previous search phrase. Therefore, a new search phrase was created to find more accurate results that directly related to the search purposes. For example: how to measure the distance between two smartphones, how we can calculate the distance between two devices, how to measure distance using Bluetooth, and what are the best practices to measure a distance between two devices.

Results: this part of the activity resulted in technologies that will help on answering the main question if it will be implied. As a result, there were articles about applying deep learning to detect objects [8,9] and some talked about using the R-CNN technologies [6], but all of that is to detect objects in a picture or video which do not help in solving the current problem.

During this research other articles appear to be helpful and it listed some methods and materials that are available to be used to achieve the distance measurements between two devices:

* Using sounds, it is about clocking the time delay of the emitted sounds waves. This method is already used in the Acoustic Ruler app [1].
* Using Bluetooth: every smartphone nowadays has Bluetooth; it should not be hard to measure the distance once a phone gets closer to another [4]. That could be possible by measuring the RSSI signal strength, but it is not quite accurate since there are many factors that play a role in that measurement e.g. (the type and the condition of the battery) [5].
* Using Wi-Fi: it is possible to achieve the distance measurement using Wi-Fi and the two devices do not have to be on the same network, but a big disadvantage of using Wi-Fi is that two devices should be connected to the internet.
* Using GPS: It is not helpful in measuring the distance between two phones, because it is not good on a security aspect to assure both devices that the location is the good one.

#### Activity: examine if this technology can tell what the object is about (person, tree, etc.)

Process: It was not convenient to continue researching with the above-mentioned question. The result of activity 1 corrected the path of this research. Upon that, I decided to reform the question to look as follows: how do the technologies found in activity 1 work and what does it offer?

For this activity it was hard to apply the research method Workshop -> prototyping since there was not enough information from the previous activity to achieve that. Therefore, I decided to go with the Library -> Literature study to find more general information, guidance, and best practices.

Start by making a search plan and identify relevant keywords.

* How can we measure a distance using the sounds waves?
* Using Bluetooth to measure the distance between phones.
* How can Wi-Fi help in finding the destination between two devices?

After that, I started by finding and judging material. Searching for interesting references and repeating the search process. after finding some relative references, explore and compare them, and finally choosing one to summarize endings.

Results: the expected results of this activity to get familiar with the way of working for the found technologies.

* Using the sounds waves:

Since each smartphone has the two essential devices to send and receive sound (microphone and speaker). The travel distance can be determined using the characteristics of sound waves. There are many ways to achieve that but in general, the time-based is the most used.

The time needed for a sound wave to travel from point A to B could be converted to distance after applying some mathematics. That is only possible if we have enough information about the speed of the wave in a predefined environment [10].

“Phone 1 sends a pulse, phone 2 responds to it with another pulse, everything has been recorded by both phones. The play icon represents what the phones are emitting, the microphone icon that the phones are recording.”: source [10].A picture containing clock, computer, computer, display

Description automatically generated

Disadvantage of this method could be:

1. That the two devices should know about each other.
2. A soundtrack should be played on two phones.

* Using Bluetooth:

Bluetooth is a fantastic part of a smartphone. We can do many things with it. But how can Bluetooth help in solving the main problem?

Bluetooth devices can send low-power signals which do not use much of the battery power as “advantages”. However, this signal has a strength. The signal loses strength when you get away from an electronic device. What we are trying to explain here is: that we can conclude the distance between two phones by using attenuation of the signal. The signal strength should decrease like the square of the distance [11].

source: [11]Diagram

Description automatically generated

But how to measure the strength of a signal? That is happening by using RSSI (Received signal strength indication). After applying the RSSI measurements to the above-mentioned formula, we can calculate an estimated distance e.g., -45 > RSSI ≥ -51 (RSSI’s units are in dBm) is equal to 1 meter in real distance [12]. Also, there are things that make this approach not really accurate and that differ per environment:

* + Emission power.
  + Emitting device antenna path.
  + Fight path.
  + Receiving device antenna path.
  + Receiver sensitivity.
* Using Wi-Fi:

The classic and the most common way for measuring the distance between two phones before apple Before Apple bought BLE was the Wi-Fi. And more on the common approach to achieve that is pretty like what already described above in the Bluetooth section “Received signal strength indication measurements. Therefore, the same weakness points are applied here too.

“Some *companies have developed alternative algorithms, attempting to measure distance more precisely using the time of flight (ToF) or time of arrival (ToA) of Wi-Fi signals, but this can’t be done in a straightforward manner using standard Wi-Fi hardware*” [13]

*Sounds, Wi-Fi, and Bluetooth are the most suitable ways to achieve distance measurements between unconnected devices. After diving into each one of the above-listed technologies, we got to know the pros and cons of each one and the way it is working, and how it is being used to measure a distance.*

## Would it be possible to measure distance between smartphones (devices)?

#### Activity: Figure out if smartphones recognize each other by sending signals?

Process: Activity 2 covered all possibilities that could offer a solution in the field of measuring distance between two devices. Activity 2 results in us with information about each solution and how it does work. To give an answer about if the smartphones can recognize each other, a list of question needs to be answered:

* Do Android Bluetooth devices recognize IOS Bluetooth?
* Do IOS Bluetooth devices recognize Android Bluetooth?
* Is it a must to have two devices paired to measure the RSSI?
* Can a phone stream a Wi-Fi signal? Does that apply to all devices?
* Using WI-FI or Bluetooth: can we measure the RSSI with all available devices?

A research on the internet will be conducted to find out what is the best way to answer the above question.

Results: Bluetooth devices can recognize each other on both platforms (IOS & Android) I was able to prove that by using two phones:

|  |  |
| --- | --- |
| **iPhone Devise** | **Android Device** |
| Graphical user interface, text, application  Description automatically generated | Graphical user interface, text, application, chat or text message  Description automatically generated |

It was possible to pair the two devices but sending files among the two was not possible. To measure the RSSI no connection between the two phones needed. When phone A gets in the range of phone B it would be possible for each one to determine the received signal strength. If the phone detects phones A, B, C, etc. it would be also possible to measure the signal strength for all of them.

Turning the hotspot mode on the phone could make it possible that a smartphone sends a Wi-Fi signal. However, there is no information about if the mentioned approach is a good or bad practice. Nowadays every Android or iOS phone on the market can be turned into a Wi-Fi hotspot. That means it supports the hotspot mode. Turning the hotspot on the phone leads to huge consumption of the battery power comparison to Bluetooth.

## Does all smartphones os’s support the previously founded technologies?

#### Activity: find out if mobile platforms e.g. (Android and IOS) support this technology?

#### Activity: Get to know the OS versions that are supported for these technologies.

Process: Looking back at all performed activity and analyzing them we can end up by saying that the most efficient way found during the research to measure the distance between two devices is to measure the RSSI using Bluetooth or WIFI. Therefore, this activity will be continued by searching the internet or all applicable resources to find out if the RSSI measurements are supported by the two main platforms as mentioned in this activity context. Starting by planning the research with the following questions:

* Is it possible to measure Bluetooth RSSI on IOS?
* Is it possible to measure Bluetooth RSSI on Android?
* Is it possible to measure WIFI RSSI on IOS?
* Is it possible to measure WIFI RSSI on Android?

Results: As an answer to the question was mentioned above: we can measure the signal strength of an IOS Bluetooth device [14], but apparently, there is no way to get any WIFI information on the application that is not developed/built by Apple [15].

On the other hand, the research shows that it is possible to measure the WIFI RSSI on the Android operating system. This was clearly described in the Android documentation [16]. Android made it possible also to measure the signal strength of a Bluetooth device. That is happening by either checking the already connected devices or while the android device is performing a Bluetooth discovery [17].

To round this activity up, it is possible to measure the RSSI on android for the Bluetooth and WIFI devices while on IOS only the Bluetooth RSSI measurement is allowed. By that we determine the platform that could be used for each technology.

The following tables will list the operating systems that support the WIFI and Bluetooth received signal strength measurement. The listed below information is driven from both Android and IOS documentations.

**WIFI RSSI supported operating system**

|  |  |
| --- | --- |
| **Platform** | **OS versions** |
| Android | API level 1 (version 1.0) up to API level 30 (version 11) |
| IOS | Not supported at all |

**Bluetooth RSSI supported operating system**

|  |  |
| --- | --- |
| **Platform** | **OS versions** |
| Android | API level 5 (version 2.0) up to API level 30 (version 11) |
| IOS | iOS 5.0+ |

*It is difficult to access some features on the IOS platform while it is easy on another platform e.g., Android. This features restriction could lead to a less functional application if the application will use WIFI to measure the distance. Bluetooth seems a considerable choice to not lose functionality and give the ability to target more users. From the table above we can conclude that: developing the mobile application using Bluetooth will achieve more success and functional application*

## How can these technologies be used in mobile apps; what programming languages could be used?

#### Activity: Find out the programming language(s) or framework(s) that helps in achieving the measurement.

Process: To give an answer on this question I made use of the internet to do some literature study. The purpose of answering this question is to gain knowledge and form a good overview about the programming languages, frameworks or libraries that could be used to develop a prototype that prove the research results of all above preformed activities.

Results: It appears that there are various ways to develop the prototype. The first way: is to write native code for each platform you are targeting. In our case we are targeting Android & IOS. Therefore, you need to make two separate apps for each platform one is for android and one for IOS. Below the languages and tools needed to achieve the working prototype.

|  |  |
| --- | --- |
| **Platform** | **Requirements** |
| IOS | Language: Swift4, IDE: XCode, Device: Mac |
| Android | Language: Java or Kotlin, IDE: Android studio, Device: Windows or Mac |

The second way: is to use cross-platform mobile application development framework e.g. (Xamarin Forms) in this way only one code needs to be written to achieve a working prototype. The application built with Xamarin can run on the two platforms with the same functionality. The table below illustrates the requirements and tools needed.

|  |  |
| --- | --- |
| **Platform** | **Requirements** |
| IOS | * Language: C#. * Framework: Xamarin Forms * Bluetooth RSSI library: Xamarin-Bluetooth-le. * WIFI RSSI library: developing WIFI RSSI measurement is only valid for android and no library needed. * IDE: Visual studio or Visual studio for Mac. * Device: Windows or Mac. |
| Android |

#### Activity: Try to build a mobile app (prototype) that implements the functionality.

Process: started by collecting all found information from previous activities. Searching for some resources to install and set up the development environment. The purpose of this activity is to make sure that found technologies can be implemented to measure certain distance between two devices.

I decided to build a prototype using Xamarin forms that measure the RSSI for Discovered Bluetooth devices in the distance range of 1.5m. I set up the development environment and downloaded the needed libraries and started with building the prototype.

Results: using the Xamarin Bluetooth-le library was possible to perform Bluetooth device discovery. Also, check the status of the Bluetooth device to determine if it is on or off. After that calculate the distance for each discovered device using the provided RSSI value using the algorithm built through information that was collected previously. To calculate the distance, one of the most efficient and known formulae was used:

Distance = 10 ^ (Measured Power - RSSI) / 10 \* N

Where **Measured Power** is a factory-calibrated, read-only constant which indicates what is the expected RSSI at 1 meter to the Bluetooth device. Combined with RSSI, it allows estimating the distance.

In the worst cases, the measured power value appears to be -56 and the best case is -69. In this prototype it was hard to get the exact value for each device because it needs to establish a connection with the device and that is something we do not need to do in this application. Therefore, a measured power value of -57 was used and that is the result of taking the average of all values between -56 and -69.

**N** is constant depending on the environmental factor. Range 2–4, low to-high. In this prototype, we are expecting it to be low since the application will be used in an open area e.g. (walking on street).

When the distance is calculated the application checks if it is less than 1.5m. in that case, the application will give instructions to the user device to vibrate for 1s and send a notification to the user alerting him/her. It is good to know that the returned distance is not accurate and its approximate, because of the many factors that were mentioned above.

The process of device discovery and user alerting will be repeated each 30s. Here are the user interfaces currently available for this prototype. To view the algorithm implementation, see appendix B.

|  |  |  |
| --- | --- | --- |
| Scanning | Found devices (ready) | Notification |
| Graphical user interface, text, application, chat or text message  Description automatically generated | Graphical user interface, text, application, chat or text message  Description automatically generated | Graphical user interface, text, application, chat or text message  Description automatically generated |

#### Activity: Testing if the required function is behaving as wanted.

Process: to make sure that the required functionality of this prototype is implemented correctly we need to test it. Therefore, a component test will be made to test the implemented solution and make sure parts of the prototype are working as expected.

This type of testing tests the whole component (the prototype from A-to-Z functions). To complete this test tools are needed:

* Two Mobile apps with the app installed on both or one of them.
* Meter, to measure the distance approximately.

Testing will start by setting the meter on 1, 1.5, 2 meters distance each time in order. Enabling Bluetooth on both devices. Then looking at the results of each scan phase and see if the prototype is sending notification and vibrating. If the prototype gives a correct or approximate distance from 1 to 2 meter, then the prototype behaves in the required way.

Results: For the first time testing the prototype it appears to measure wrong values and an adjustment needs to be done. The previous Measured power was assigned based on the average of all possible values. That was not really a good estimation. Therefore, I have followed the rule that says Measured power is the RSSI on a one-meter distance. I have set the two phones on a 1-meter distance and started the application to measure the RSSI. Repeating the process 3 times each time 30 seconds and getting the average of all measurement results. That was -65. I have changed the measured power in the distance calculation and started testing again. The result of that test was kind of good, but it needs some improvement. The N Constance was most probable low. I kept increasing it until I got the expected result which is as following:

Real distance: the distance on the meter.

RSSI value: the measure RSSI for the Bluetooth device.

Calculated distance: the result appears on screen after running the calculation algorithm.

|  |  |  |  |
| --- | --- | --- | --- |
| **Real distance** | **RSSI value** | **Calculated distance** | **App sent Notification** |
| 100 cm | -67 | 1.24 | Yes |
| 100 cm | -63 | 0.83 | yes |
| 100 cm | -66 | 1.1 | yes |
| 100 cm | -65 | 1 meter | yes |
| 150 cm | -68 | 1.33 | Yes |
| 150 cm | -71 | 1.78 | no |
| 150 cm | -70 | 1.62 | no |
| 150 cm | -72 | 1.96 | no |
| 200 cm | -71 | 1.78 | no |
| 200 cm | -70 | 1.68 | no |
| 200 cm | -72 | 1.96 | no |
| 200 cm | -70 | 1.62 | no |

These were the results of the rounds of testing. Each round we measure 4 times for each real distance. The results started to get less accurate each time we got far away from the sender device. Good to know that the application was tested on Galaxy J5, which is a kind of old phone and that might also cause a problem or has a side effect on the measurement process.

*It appears that using a cross-platform framework to develop the prototype will be easier to achieve at this stage. By making use of external libraries, it was possible to read the RSSI value. Using the formula that was previously mentioned, the distance was estimated. The component testing helps here to detect if the implemented functionality is giving a wrong distance estimation or does it give a proximate one. All in all, we achieved the way that mobile applications can make use of the discovered technologies and what is needed to achieve the measurements.*

# Chapter 4: Discussion

In this chapter there will be a discussion held on each activity that was taken and how does it help in answering the main question.

The main problem was: how do smartphones help in keeping a 1.5 m distance? solving this problem needed to do some investigation and research as mentioned in the chapter 3. Start by dividing the main question into sub-questions and for each question assign some activities. In this chapter, a list of sub-question will be listed and followed by their activities. After each sub-question, I will write what value these activities added to answer the main question.

What are the already existing technologies that smartphones use to detect objects at a certain distance?

* Activity 1: Finding methods that the smartphone is already using to detect objects around it.
* Activity 2: Examine if this technology can tell what the object is about (person, tree, etc.)

Answering this question through performing the activities above got us to know all possible technologies that already smartphones support and can help detecting objects. By doing this we could determine the potential ways to achieve the goal. From the first activity, we found that a possible technology to detect objects and measure the distance were: sounds, WIFI, Bluetooth, and GPS. We looked previously more in detail on each detecting method individually. And we saw how useful each one could be. In the second activity, I decided to change the activity and that based on the results I got in activity one. The new activity becomes: how do the technologies found in activity 1 work and what does it offer? In this activity, we could dive more into detail and know exactly how it works and what are the success factors and how does this technology help in measuring the distance. The second activity results within a clearer vision of further steps. From that activity, we knew that Sounds measurement would not be optimal to measure the distance because it needs to make a real sound which will end up annoying the users and no one would like to walk in the street and keep his/her phone running. and using the GPS is also now a considerable case because it is designed for the outside world and cannot function in terms of an indoor positioning system.

Would it be possible to measure distance between smartphones (devices)?

* Figure out if smartphones recognize each other by sending signals.

By answering the first sub-question we end up having the WIFI and the Bluetooth as the potential ways to achieve the distance measurement. After doing some examination and finding out the best and bad practices and previously mentioned in chapter 3. It appears that smartphones can detect each other using Bluetooth and WIFI.

Does all smartphones os’s support the previously founded technologies?

* Find out if mobile platforms e.g. (Android and IOS) support this technology?
* Get to know the OS versions that are supported for these technologies.

By executing this activity, I was able to determine if the WIFI and Bluetooth are the most suitable choices. I wanted to see if those technologies are accessible on all mobile operating systems. Also, which versions of those operating systems support the technology. In that way, I can determine my end users. This activity results in accessing some necessary WIFI details that are not possible on the IOS operating system. That left us with one option which is: Bluetooth.

How can these technologies be used in mobile apps; what programming languages could be used?

* Find out the programming language(s) or framework(s) that helps in achieving the measurement.
* Try to build a mobile app (prototype) that implements the functionality.
* Testing if the required function is behaving as wanted.

After justifying the choice about which technology is the best to achieve 1.5m distance measurement between two devices and discovering the way it works. This is the right time to find out how this technology can be used in a mobile application and how to be implemented. Starting by discovering the possible way to implement the solution into a mobile application and getting to know all possible ways to achieve it. That was described in chapter 3. After gathering all resources, I started by gathering the pieces together and building a prototype. This prototype should implement the findings from the previous activities. For example, it should scan for Bluetooth devices that are in the range. When the device is detected we try to estimate the distance by the formula mentioned above in chapter 3. If the device appears to be in the range of 1.5 meters away from the current device an alert will be sent to the user and the device will vibrate. The testing activity is meant to see if the implemented solution is done well. At an early stage of the test, some bugs appear. That bug was because of the measured power factor, it was not dynamically retrieved from the discovered device that is due to some implementation limitation in the chosen framework.

# Chapter 5: Conclusion(s) and recommendation(s)

### Conclusion:

Having a mobile application that measures the distance between two different devices, is possible. Smartphones do have and support technologies that could help us in achieving estimated real distance measurement. This is done by measuring the RSSI (received signal strength) and the measured power which is the power of RSSI on one-meter distance away from the advertising device. There are various ways as mentioned to measure the RSSI either from Bluetooth or from WIFI devices. Due to some limitations on the IOS platform, the implementation of having a RSSI measured from WIFI was not possible and therefore, an implementation with RSSI measure form Bluetooth device is implemented. After concluding a research on how this solution could be implemented, a prototype was developed to ensure the research results. This prototype is estimating the distance between two devices in a considerable but not perfect way.

### Recommendations:

In the section above, it was mentioned that the prototype could measure estimated distance in a good way but not perfect. That is because of the way the prototype was implemented. To develop the prototype, I had to make a choice using which tools to achieve the implementations. Therefore, I decided to go with Xamarin forms for simplicity and to speed the development process up. There are different issues that result from making that choice. Using Xamarin forms you will not be able to access some native features in the targeted platform. Because of that, it was not possible to retrieve the Measured power value (RSSI value on one-meter distance) from the scanned device using Xamarin forms. If the implementation was done using a native platform then the measurement will be a better more accurate distance.

# Chapter 6: Reflection

This chapter is to make a flash-back on the process that was taken and the way it was performed. The main part here is to cover what went well and could be adapted in further research and what could be done better.

Starting with the approach, it was well structured and in a well-ordered way that makes the research look more like collecting pieces of blokes together. The main question was detailed enough, not too broad, and not too narrow. It was easy to break it into sensible sub-questions. Each sub-question contends a couple of activities. Those activities should have been taken into consideration and evaluated more carefully before starting the research. As seen in chapter 3 some of this activity appears to not be targeting the main problem in a direct way and an adjustment had to occur to maintain the way of working and make the activity adding more valuable input to help answer the mean question.

In chapter 3 while performing each activity a process was defined. I found the way that I did this process is good and gives the reader a clear idea about what I have done or what are the questions and phrases were written and searched about to get the result of that activity. This is something that I might use in other research if it will be applicable.

# References

1. *Acoustic Ruler*. (2016, November 30). Iq-Tainment. https://iqtainment.wordpress.com/acoustic-ruler/
2. *Bluetooth tracking and COVID-19: A tech primer*. (2020, March 31). Privacy International. https://privacyinternational.org/explainer/3536/bluetooth-tracking-and-covid-19-tech-primer
3. Chou, L. P. (2020, September 4). *A New Scalable Scatternet Formation for Bluetooth*. ResearchGate. https://www.researchgate.net/post/How\_to\_detect\_the\_object\_as\_well\_as\_its\_distance\_using\_tringulation
4. *Distance between two mobile devices in Javasacript (Ionic)*. (2017, March 14). Stack Overflow. https://stackoverflow.com/questions/42797813/distance-between-two-mobile-devices-in-javasacript-ionic
5. *How to estimate distance between two android devices?* (2012, December 23). Stack Overflow. https://stackoverflow.com/questions/14015239/how-to-estimate-distance-between-two-android-devices-bluetooth-preferred
6. Jadhav, K. (2020, June 9). *Object Detection via Smartphone Camera using Faster R-CNN*. Medium. https://medium.com/swlh/object-detection-via-smartphone-camera-using-faster-r-cnn-c910471b0a53
7. *Using Bluetooth to Measure distance between android device and objects*. (2015, November 10). Stack Overflow. https://stackoverflow.com/questions/33638476/using-bluetooth-to-measure-distance-between-android-device-and-objects
8. Zhang, Y. (2020a, April 24). *Deep Learning for Detecting Objects in an Image on Mobile Devices*. Towardsdatascience.Com. https://towardsdatascience.com/deep-learning-for-detecting-objects-in-an-image-on-mobile-devices-7d5b2e5621f9
9. Zhang, Y. (2020b, April 24). *Deep Learning for Image Classification on Mobile Devices*. Towardsdatascience.Com. <https://towardsdatascience.com/deep-learning-for-image-classification-on-mobile-devices-f93efac860fd>
10. S.-M.-A.P.D. (n.d.). *Simultaneous Acoustic Localization of Multiple Smartphones with Euclidean Distance Matrices*. Arxiv.Org. Retrieved October 20, 2020, from https://arxiv.org/pdf/1512.05103.pdf
11. Dehaye, P. (2020, October 19). *Inferring distance from Bluetooth signal strength: a deep dive*. Medium. <https://medium.com/personaldata-io/inferring-distance-from-bluetooth-signal-strength-a-deep-dive-fe7badc2bb6d>
12. C.C.C. (2012, January 1). *Development of an Android APK for Bluetooth localization*. Upcommons.Upc.Edu. https://upcommons.upc.edu/bitstream/handle/2099.1/17357/memoria.pdf?sequence=4&isAllowed=y
13. *StackPath*. (n.d.). StackPath. Retrieved October 29, 2020, from <https://www.electronicdesign.com/technologies/communications/article/21800581/whats-the-difference-between-measuring-location-by-uwb-wifi-and-bluetooth>
14. *Reading RSSI of Bluetooth peripheral in background at short interval*. (n.d.). Developer.Apple.Com. Retrieved October 25, 2020, from https://developer.apple.com/forums/thread/95222
15. *Get wifi RSSI on iOS*. (n.d.). Developer.Apple.Com. Retrieved October 25, 2020, from https://developer.apple.com/forums/thread/114437
16. *WifiManager*. (2020, September 30). Developer.Android.Com. https://developer.android.com/reference/android/net/wifi/WifiManager
17. *Get bluetooth signal strength*. (2013, March 9). Stack Overflow. https://stackoverflow.com/questions/15312858/get-bluetooth-signal-strength

# Appendixes

## Appendix A: T&H Subject Description

A picture containing table

Description automatically generated

A screenshot of text

Description automatically generatedTable

Description automatically generated

## Appendix B: Prototype code & algorithm

Source code: <https://github.com/adnanjaw/LetsMeasure1.5.git>

APK for Android devices: <https://github.com/adnanjaw/LetsMeasure1.5m/blob/master/com.companyname.letsmeasure15.apk>

A: The following piece of code shows the status checker for the Bluetooth device.

private void BluetoothStatusChecker()

{

ble.StateChanged += async (s, e) =>

{

if (e.NewState == BluetoothState.Off ||

e.NewState == BluetoothState.Unavailable ||

e.NewState == BluetoothState.Unknown)

{

await DisplayAlert("Bluetooth Status",

$"Blutooth is: {e.NewState}",

"OK");

};

};

}

B: For the Bluetooth device discovery, the following code was written:

private void CheckForAnyNewDevice()

{

Device.StartTimer(new TimeSpan(0, 0, 20), () =>

{

Device.BeginInvokeOnMainThread(async () =>

{

StatusActivityIndicator.IsRunning = true;

StatusActivityIndicator.IsVisible = true;

devices.Clear();

LbStatus.Text = "devices cleard";

adapter.DeviceDiscovered += (s, a) =>

{

IDevice device = a.Device;

VDevice vDevice = new VDevice() {

Device = device,

Distance = 0

};

if (!devices.Contains(vDevice) && !FilterDevices(vDevice))

{

vDevice.Distance = CalculateDistance(vDevice);

devices.Add(vDevice);

}

};

if (!ble.Adapter.IsScanning)

{

adapter.ScanMode = ScanMode.LowLatency;

LbStatus.Text = "Scaning";

await adapter.StartScanningForDevicesAsync();

}

DevicesListView.ItemsSource = devices;

StatusActivityIndicator.IsRunning = false;

StatusActivityIndicator.IsVisible = false;

LbStatus.Text = "Rescaning in 30s";

});

return true; // runs again, or false to stop

});

}

C: To make sure we are scanning only of mobile phones

private bool FilterDevices(VDevice device)

{

if (String.IsNullOrEmpty(device.Device.Name))

return false;

foreach (string type in DeviceType)

{

if (device.Device.Name.Contains(type))

{

return true;

}

}

return false;

}

D: The code for calculation distance

private double CalculateDistance(VDevice device)

{

device.Device.UpdateRssiAsync();

double MeasuredPower = -57;

int RSSI = device.Device.Rssi;

int N = 2;

double one = MeasuredPower - RSSI;

double two = 10 \* N;

double resutltPartOne = one / two;

double distance = Math.Pow(10,resutltPartOne);

distance = Math.Round(distance, 2);

if (distance <= 1.5)

{

SendNotification();

SendVibrateAlert();

}

return distance;

}