Expanded Tele-heatlh Platform for Android

Adarsh Mani
Phillip Azar
Jochem Van Gaalen
Quan Peng

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2015-110
http://www.eecs.berkeley.edu/Pubs/TechRpts/2015/EECS-2015-110.html

May 14, 2015

Acknowledgement

Dr. Ruzena Bajcsy
Dr. Ali Javey
Daniel Aranki

University of California, Berkeley College of Engineering

MASTER OF ENGINEERING - SPRING 2015

Electrical Engineering and Computer Sciences

Robotics and Embedded Software

EXPANDED TELE-HEALTH PLATFORM FOR ANDROID

ADARSH KUMAR MANI

This **Masters Project Paper** fulfills the Master of Engineering degree requirement.

Approved by:

1. Capstone Project Advisor:

Signature: ___________________________ Date ____________

Print Name/Department: RUZENA BAJCSY/ELECTRICAL ENGINEERING AND COMPUTER SCIENCES

2. Faculty Committee Member #2:

Signature: ___________________________ Date ____________

Print Name/Department: ALI JAVEY/ ELECTRICAL ENGINEERING AND COMPUTER SCIENCES

Expanded Tele-health Platform For Android
M.Eng EECS - Robotics and Embedded Software - 2015

ABSTRACT

In 2013, of 26 billion dollars spent on patient readmissions in Medicare, an estimated 17 billion of this was generated by readmissions caused by avoidable factors. The most preventable of these factors - and arguably the most severe - are improper administration of therapy and medication. To combat this, we developed a lean, smartphone-based telemedical system that not only provides personal health statistics to the patient, but relays this information to the doctor allowing for a more tailored approach to medicine. By leveraging the latest open source health monitoring algorithms and libraries, we have created a reliable, low-cost and easily implementable solution that has the potential to simultaneously improve patient care and dramatically reduce healthcare expenditure.

University of California, Berkeley College of Engineering

MASTER OF ENGINEERING - SPRING 2015

Electrical Engineering and Computer Sciences

Robotics and Embedded Software

EXPANDED TELE-HEALTH PLATFORM FOR ANDROID

ADARSH KUMAR MANI

This **Masters Project Paper** fulfills the Master of Engineering degree requirement.

Approved by:

1. Capstone Project Advisor:

Signature: ____________________________ Date _____________

Print Name/Department: RUZENA BAJCSY/ELECTRICAL ENGINEERING AND COMPUTER SCIENCES

2. Faculty Committee Member #2:

Signature: ____________________________ Date _____________

Print Name/Department: ALI JAVEY/ ELECTRICAL ENGINEERING AND COMPUTER SCIENCES

ABSTRACT

In 2013, of 26 billion dollars spent on patient readmissions in Medicare, an estimated 17 billion of this was generated by readmissions caused by avoidable factors. The most preventable of these factors - and arguably the most severe - are improper administration of therapy and medication. To combat this, we developed a lean, smartphone-based telemedical system that not only provides personal health statistics to the patient, but relays this information to the doctor allowing for a more tailored approach to medicine. By leveraging the latest open source health monitoring algorithms and libraries, we have created a reliable, low-cost and easily implementable solution that has the potential to simultaneously improve patient care and dramatically reduce healthcare expenditure.

2015

Expanded Tele-Health Platform for Android

CAPSTONE REPORT

ADARSH MANI, PHILLIP AZAR, JOCHEM VAN GAALEN, QUAN PENG

Table of Contents

1. Introduction

In 2010, $17 billion dollars was spent on patient readmissions in Medicare. (Centers for Medicare & Medicaid Services: 2012). The most preventable readmission factors - and likely the most severe - are improper administration of therapy and medication. To combat this, we are developing a lean, smartphone-based telemedical system that not only provides personal health statistics to the patient, but relays this information to the doctor allowing for a more tailored approach to medicine. By leveraging the latest open source health monitoring algorithms and libraries, we have created a reliable, low-cost and easily implementable solution that has the potential to simultaneously improve patient care and dramatically reduce healthcare expenditure.

The team, advised by Dr. Ruzena Bajcsy and led by Ph.D. student Daniel Aranki, consisted of several members, including four Master of Engineering students (Phillip Azar, Adarsh Mani, Quan Peng, and Jochem van Gaalen) specializing in Robotics & Embedded Software concentration in the EECS department, and engineers Arjun Chopra, Priyanka Nigam, Sneha Sankavaram, Maya Reddy and Qiyin Wu.

The aim of this project is to develop and test a highly configurable, open-sourced Android powered framework that will enable doctors and hospitals to remotely monitor outpatients with chronic health conditions and determine their risk of re-admittance. This platform will transform the ubiquitous smartphone into the backbone of a tele monitoring system which uses various sensors to track different vital signs. These sensors can either be off-the-shelf products (e.g. temperature sensors) or sensors embedded in the smartphone itself (such as gyroscopes, accelerometers, cameras etc.). Once the data is extracted, the smartphone will be used to relay the information, in a fault-tolerant manner[1], to servers where various diagnostic algorithms are run to calculate re-admittance risk factor and other recuperation metrics. Security and privacy procedures will be embedded into these data structures prior to storage.

The team's specific contribution to this project is to develop:

1. Fault-tolerant communication between various off-the-shelf vital sign extractors and the smartphone via Bluetooth.

[1] Fault-tolerance can be thought of as a personal guarantee to the user that the process will be handled despite the presence of faults such as unexpected power downs, disconnection etc.

2. Novel vital sign extraction algorithms for heart rate and blood pressure using the smartphone's camera to serve as a proof of concept to demonstrate the fault-tolerant communication and provide a simpler way to conduct health and vital monitoring.

Over the course of the academic year, an application in Android was built and tested through laboratory validation of individual components, and a pilot study conducted near the end of the project which combined these. The results of this work is outlined in this report, along with challenges faced and potential next steps for this project.

This project was also analyzed from a business perspective, with potential competitors highlighted, and a thorough analysis of the industry and market trends to aid in identifying potential business opportunities.

2. Market Trends Analysis

2.1.1 Motivation & Outline

This section of the paper is dedicated to analyzing the telehealth industry and deriving a potential business strategy for a startup centered about our capstone project. It begins by discussing the broad industry trends that show which direction would yield the best go-to-market results. Next, the importance of adopting a well-defined strategy in terms of maximizing business potential is highlighted. From analyzing the industry as a whole in terms of its value chain, the most optimal place within this market a business should occupy will be determined. This is done by identifying the full chain and finding the strongest link. This is followed by an analysis of Porter's Five Forces, which is used to understand strengths and weaknesses in the chosen position within the value chain. Knowing these strengths and weaknesses is a key factor in shaping the business plan. The conclusions drawn from these forces will be presented along with suggestions for ameliorating any weaknesses identified.

After extensive analysis using these models, the section of the market that was identified to be targeted is the "mediator" segment - a segment whose products would form the platform that will serve as common gateway for devices measuring vital signs to communicate with health analytics services and vice versa. The platform the team is developing will allow for a much easier facilitation of data between entities specializing in either the analysis or measurement of health data. It essentially abstracts away the communication layer of the entire telehealth chain into an easy to use, robust and secure service. The strategy adopted to conquer this segment is to pursue an open-source double licensing model wherein royalty will be collected at a pre-determined rate only when the service or code is used for commercial purposes. Personal use cases will be free of charge.

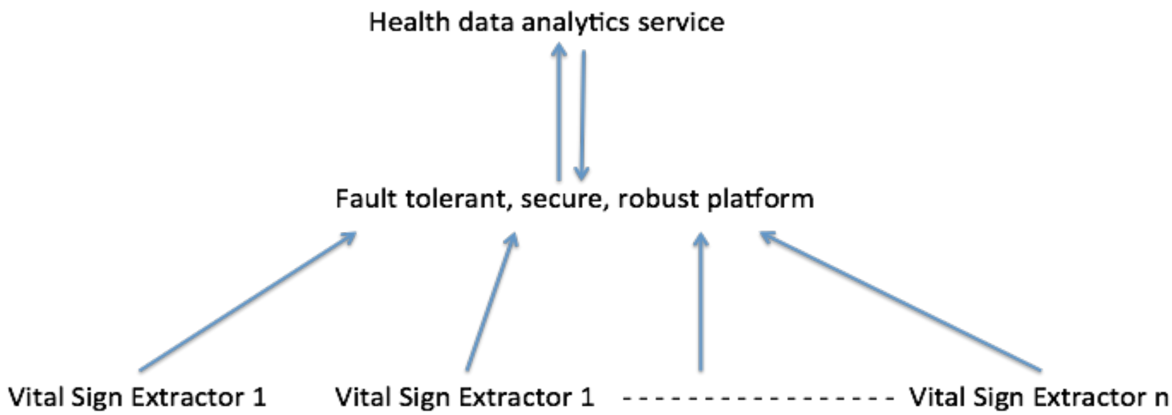

Figure 1.1: Telehealth Platform Hierarchy

We live in a highly interconnected world. Everything from common household appliances, such as refrigerators, to complex control systems, such as those found in vehicles are connected to information relaying services designed to expand their ordinary functionality. Our lives are becoming more and more dependent on being "connected," and as a result, many major social, technological, and economic trends in the last decade have been fueled by connectivity. Among these trends is the push for telehealth, which since the 1990s, has been a principal force in providing healthcare to underserved communities. (Miller 2007: 133-141) By leveraging major social trends towards connected privacy, availability, and transparency, it is possible to flesh out a telehealth strategy that addresses key impacts and vantage points associated with what will be collectively known as the "connectivity trends." These connectivity trends are driven by a new, socially aware consumer that can greatly influence a product's adoption rate and popularity.

To provide context for these consumers, it is important to examine contemporary cases of social awareness with respect to connectivity. In recent years, the online health industry has been abuzz with issues of privacy, data storage, and patient protection. So much so, in fact, that entire companies have been spawned out of the necessity for these core three attributes (e.g. Veeva Systems.) Chief among these concerns is that of patient protection. Where as many medical devices remain "analog" with respect to connectivity (i.e. no internet/wireless access,) manufacturers are pushing to create connected equivalents of these products. With this comes the potential for new features, but also the introduction of many risks. In a more extreme (and relevant) example, data from an insulin pump that now communicates via Bluetooth low energy may find itself in the hands of third party advertisers. (Hall & McGraw 2014: 216-221) While this data is protected under the Health Insurance Portability and Accountability Act (HIPAA) passed in 1996, it is very likely that the patient will be unaware of what is occurring, since they are often surrounded by Bluetooth enabled devices with internet access. This creates a very real, and potentially dangerous situation: data can be intercepted and collected from medical devices, which can then be used for almost any purpose. And in effect, what you find is similar in many ways with the motivation behind the Digital Millennium Copyright Act (DMCA,) in which illegally obtained data is so abundant that liability is no longer held against companies who store that data.

From a strategy perspective, the contemporary social awareness of patient privacy, data storage, and protection significantly detracts from attempts at commercialization. The FDA Center for Devices and Radiological Health (CDRH) is the major regulatory arm that governs the development, implementation, and evaluation of products with telehealth features. Since the FDA is a member of the federal government, policymakers, who in turn are influenced by those who they represent (and in many cases, special interest groups,) have the ability to directly impact the regulation of telehealth systems. Negative social acknowledgement can therefore directly harm the commercialization of a new telehealth product by influencing the implementation of stricter regulation. So the question remains, how can telehealth products hope to compete against an increasingly aware consumer base?

It is not feasible to simply promote ignorance of a very real and tangible problem: the digital age has long since begun, and next to nothing is primarily stored physically. If you cannot simply "wish" the knowledge away, then the next best thing is to approach the problem with the contrapositive: education. And this kind of education, one that targets the socially aware consumers (and in this case, patients,) is not strictly limited to the benefits of a product. In fact, it is critical to convey both sides of the technology. Net neutrality is one modern example where conveying only the benefits of a technology is more subversive than it is helpful. Major internet service providers attempted to only convey the benefits of discarding net neutrality in an attempt to mask its true purpose as a means for generating additional revenue. This had the effect of creating a nearly unanimous voice against the proposition. In conveying both the benefits and downsides to telehealth, we can best leverage the trend of social awareness in connected health care, and in turn, create a more positive public image.

The importance of influencing the socially aware consumer cannot be understated: coordinating a strategy of education rather than marketing can mean the difference between revolutionizing and downplaying the entire telehealth industry. With the advent of free online educational resources, it is simply not good enough to assume that a market segment will be ignorant about a product or feature. A good strategy will leverage this education and in some ways seek to exploit it, feeding a growing demand for high tech while at the same time appealing to the scientific side of a market segment. There are, however, many more sides to this “good” strategy that cannot be addressed by analyzing the consumer alone.

2.2.1 Strategy: What it is and why it is required

Strategy is both the art and science involved in the formulation and evaluation of key-decisions that will give an entity an edge over its competitors, attain a sustainable advantage and help achieve its long-term objectives. Delineating a strategy also helps the organization formalize objectives, and hence give it concrete direction. It helps one decide how to position a product and what decisions to make to maximize growth and profitability, as it is just as important when trying to enter a saturated market as it is when trying to enter a nascent one. It is just as useful when trying to consolidate one’s lead in a market as when one is trying to establish it.

The telehealth industry is still relatively nascent, but is rapidly expanding. Industry revenue was projected to grow $320.2 million in the five years leading to 2014, including revenue growth of 23.1% in 2014. From 2014 through 2019, the industry stands to reap the benefits of demographic, structural and legal factors. These include a growing aging population, advances in telecommunication and wearable technology, and statutes like the Affordable Care Act. As a result, industry revenue is expected to increase at an annualized 49.7% to $2.4 billion in the five years to 2019 (Morea 2014:9).

Although the top two companies, GlobalMed and InTouch Technologies, occupy 22% of the market share (Morea, 2014), the industry itself is rapidly expanding, and thus there is ample opportunity for this technology to enter the market and seize a sizable share.

However, given that the project is a potential spin off from an academic endeavor, and doesn't have (at least presently) the financial clout or business savviness of companies already in the market, adopting the right strategy becomes crucial to conquer and succeed in this market. Where the team temporarily lacks in business authority, it excels in technical skill and prowess. While it is acknowledged that this will have to change in the future, the team believes it can use this technical superiority to get a foothold in market.

2.3.1 The Strategy

As mentioned in the previous section, the team plans to leverage its technical prowess to its advantage. This involves building a robust easy to use, easy to integrate platform, and then treading the open-source route. In the current market, existing solutions are mostly "walled-gardens" with players providing proprietary solutions to all fields in telehealth (vital-sign extraction, collection, logging, analysis and result distribution). However, with the impending explosion of the internet of things, several third-party developers of medical hardware used for constantly monitoring vital sign data are expected (Lars 2014). The number of connected devices is expected to reach 50 billion mark by 2020 (Swan 2012). This data emphasizes a need for an open platform to handle the software/communication part of the system. It is this opportunity that the strategy is designed to seize.

2.4.1 Industry Value Chain Analysis

The first requirement one must consider when creating a new business is how it should be positioned within the industry. For example, there are many levels involved in delivering medical devices to hospitals. A company wanting to produce a device must first buy supplies, and then usually contract out the labor to a manufacturer. Once the devices have been manufactured, the company will sell these medical devices to a distributor who carries the device in a catalog. Hospitals partnered with these re-sellers will then purchase these devices, process them, and finally hand them over to doctors who will use them as a part of the care they provide.

We can see that the value chain is long, and in many cases, not quite linear. Various market forces also make the chain asymmetric in that the value that is realized in each part of the chain is not evenly distributed. Many markets often have a particularly 'fat', or highly profitable part of the chain while the other sections fight for the leftovers. An example of this can be seen in the technology industry. In delivering cell phones to consumers, Apple, the company that designs and markets its products, clearly makes the most money, even though there are many other businesses involved in the value chain such as the phone manufacturers (e.g. Foxconn), distributors (e.g. FedEx) and carriers (e.g. AT&T). Unlike the consumer technology industry which arguably has started to reach maturation, telehealth is still a fledgling industry, which is still being shaped by early startups, new government regulations and many new technological innovations. Despite this, the telehealth industry value chain is no different in that the 'fat' part of the chain can clearly be identified.

From the IBISWorld report on telehealth, the industry value chain is shaped with several 'demand' players such as data processing and hosting services, hospitals, physical therapists and health & welfare funds to name a few (Morea 2014:14) . On the other side of the chain, there are several 'supply' players providing infrastructure within which telehealth services can operate. These include businesses such as mobile carriers, software publishers and hardware manufacturers.

Because of the wide variety of players in the industry, both supplying and pulling technologies to and from telehealth services, a simplified linear model was developed to illustrate the supply chain. This is shown below in Figure 2.1.

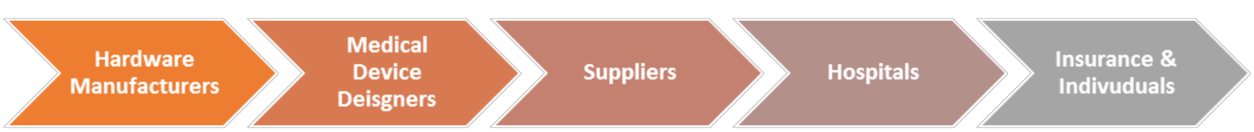

Figure 2.1: Industry Value Chain

From here, we see that the chain is longer than one might expect in that there are several intermediate steps involved with getting a device into a hospital. It is important to note that this value chain is a simplification of the actual chain, which is not linear and has many more steps involved within each major stop in the chain presented.

Due to the established nature of similar industries requiring the same suppliers and hardware manufacturers as the medical devices industry, suppliers and hardware manufacturers are well developed and highly competitive (Kahn 2015:24). Thus, the fat part of the value chain is located with the medical device designers. A good example of the power of this part of the chain are medical devices like Medtronic and Siemens AG - companies whose revenues exceed $17 billion USD**,** which is where we would like to position a potential startup. It is important to note however that there remain several large pitfalls in which a business situated as a device designer could become wildly unprofitable. Many of these have to do with regulatory approvals (Morea 2014:21) such as meeting FDA requirements (Medina, Kremer & Wysk 2012). However, once met, these businesses will be well situated within the chain. The team's particular business would circumvent many of these pitfalls by taking on the 'mediator' role within the medical devices industry. As alluded to earlier, this would allow participating medical devices companies to safely gather data from patients and provide additional services derived from this.

By identifying the best location within the industry chain, we can now analyze which parties in the medical industry would have interests driving the adoption of new medical devices. These parties would be ideal candidates to which a startup business could offer services. Performing this analysis also has the added benefit of being able to predict the demand of the industry, which is a function of the size and strength of desire of its customers.

2.5.1 Potential Stakeholders

By its very nature, there are a plethora of potential stakeholders with key interests in the success of this platform. First, insurance companies would have an interest in the rise of tele-health monitoring systems, as having the ability to better assess the risk of patients could drastically change the way that insurance is calculated. From this, it follows that insurance payers and patients would see substantial benefits in the way that medicine is applied in that increased information about the patient will allow for more accurate diagnoses and more targeted treatments. This will ultimately help in lengthening and improving the quality of life.

Furthermore, the ability to take measurements of a patient's vital signs opens the door for alternative options to physically visiting the doctor – a practice continued for centuries. This is an enormous market currently valued at approximately $1.151 billion USD and growing at approximately 2.2% yearly (IBISWorld Report: Number of Physician Visits 2014:1) which presents a huge financial opportunity for the telehealth medicine. While certainly not a complete substitute, the telemedicine industry could replace large swathes of routine checkups and minor cases. This could serve as a boon not only for businesses looking to save on costs associated with granting time off to see a doctor but also for understaffed hospitals to relieve labor-related costs.

Governments and regulatory bodies such as the World Health Organization (WHO) would benefit from enhanced data collection in order to present new legislation or seek out cost cutting opportunities in relation to providing health care, and to help model and eventually mitigate the effects of contagious new diseases. With recent legislative moves seeking to expand health coverage in the US such as the Affordable Care Act, an increasing amount of financial support will flow into the telehealth industry contributing to the estimated 49.7% growth in industry revenue (Morea 2014:5) as a reaction to making care affordable to a wider swath of the population.

From this analysis, we can see that there are many large and powerful stakeholders with keen interests to either increase profits, or reduce costs with the aid of telehealth technology. This further validates the team's judgment in the Porter's power of buyer's analysis. With so many parties interested, and relatively weak internal industry rivalry, starting a business in this sector of the industry should have the highest chance of success.

Now that the stakeholders have been identified, it is crucial to understand the role that the team's technology would play in commercial applications. Investors in a startup are investing primarily in the technology, and the team, so without a clear vision of how the technology will function, it would become impossible to market it correctly or draw up a business model.

2.6.1 Relation between Technology and Commercial Product

The ability to monitor vital sign information in real time has direct applications to commercial technology. The team aims at providing a common platform to developers with some sample vital sign extraction methods embedded with the idea that other companies build on top of this platform. This platform will enable developers to work on producing technology more focused on data extraction and/or analysis without having to spend inordinate effort establishing fault tolerant protocols, server security, and data privacy. The idea here is to position the platform similarly to how the Android operating system is marketed. Android is an open-source cellular operating platform that comes with several basic functionalities, but is meant as a platform where third-party developers can extend its functionality.

Different vital sign extraction techniques are analogous to new apps developed specifically for this platform. Revenue would be collected from licensing agreements between device makers seeking to utilize the platform for profitable ventures. Such a strategy would have a benefit of rapid growth due to its flexibility, but could have the potential downside of being slow to grow revenue. In all likelihood, due to the nature of the technology industry to rapidly grow in new sectors, this would be the best approach as opposed to a strategy that would involve keeping the platform closed and competing rather than collaborating with medical devices and technology companies.

With a defined position within the value chain, and an idea of how the technology in this capstone project will interact with the final product, we can now analyze the remainder of Porter's Five Forces to assess potential threats in the industry.

2.7.1. Porter's Five Forces

In his seminal article in the Harvard Business Review, Michael E. Porter outlines five major factors that should influence the approach one should espouse to be successful (Porter 2008). These are:

1. Bargaining power of buyers
2. Bargaining power of suppliers
3. Threat of new entrants
4. Threat of substitutes
5. Rivalry amongst existing competitors

Each of these forces will be analyzed in detail in how they pertain to our chosen industry in the following sections.

2.7.2 Bargaining power of Suppliers

The bargaining power of suppliers refers to the ability of business providing materials, data, IP etc. to a business further down the chain to ask for increased compensation or to switch to supplying a direct competitor. Due to the fact that this platform is largely based off the heavily subsidized, and highly competitive mobile phone market, the suppliers in this regard have little power to demand higher prices were the telemedicine industry to grow and increase their demand. Furthermore, the ubiquity of Android-based phones (International Data Corporation 2014) in the consumer market further weakens dependence on suppliers as our platform does not require a proprietary phones. Rather, a patient can just use a personal phone to use the monitoring platform.

2.7.3 Bargaining power of Buyers

The bargaining power of buyers refers to the ability of customers of the business to demand extra features, lower prices, or switch to a competitor. Due to the suggested licensing strategy of this product, the buyers in this case would be device manufacturers and potentially research laboratories and regulatory bodies looking to do research on the aggregate dataset of patient data. It is without doubt that the sale or licensing of data would come with some sort of public resistance, but if done in an identity sensitive way (and this is already ingrained into the platform itself from its inception) this could open the door to many licensing opportunities. Since there are few alternatives to this system, the power of buyers to simply switch to another platform would be weak.

2.7.4 Threat of New Entrants

New entrants bring with them fresh ideas and a tenacity ideas to gain a share of the market from existing players. Ease of entrance, characterized by barriers to entry, will encourage more new comers especially if the industry is nascent and has potential for being lucrative. This precisely characterizes the telehealth market. We live in a connected world and with the Internet of Things verging on explosion, telehealth is one of the many industries that will receive a significant boost in market size (Lars, 2014.)

In such a scenario, it is anticipated that many players will prop up to get a piece of the pie. It is, therefore, of utmost importance to increase the barriers to entry to dissuade potential entrants. The team has done this by increasing barriers to entry in the following ways:

1. As the team is building a platform that will be leveraged by third-party manufacturers of health monitoring devices, the platform has to be robust, complete and easily joinable. Developing a solution that offers all three features requires a lot of developmental effort and time, which in turn raises the barrier of entry.
2. Once adopted, it'll be very difficult for our customers/partners to switch to a new platform.

3. The open source model works in the team's favor. Open source provides for collaboration and cooperation as opposed to competition. As the platform gains more adoption, a larger community would organically grow, further fuelling adoption.

By raising the barrier to entry for new entrants, the team has ensured a sustainable "economic moat" can be built.

2.7.5 Rivalry

Rivalry is greatest in an industry if competitors are roughly of equal size, industry growth is slow or if exit barriers are high (Porter, 2008). It can become particularly entrenched if the rivals are competing on price. Size and financial might of the competitors play a major role in determining who comes out victorious in market share wars. The current major players in the telehealth market are GlobalMed and InTouch Technologies who together own 22% of the market (Morea 2014:25). However, the industry itself is rapidly growing and hence there is abundant opportunity for entry.

In the telehealth industry, the basis of competition is primarily based on product quality, which is determined by factors like product functionality, features, speed and ease of use. Other media of competition include breadth of services, marketing, customer services and training offered to clients. (Morea 2014:24)

Given this, the biggest threat for the team comes from the threat of rivals. Being a potential startup spun off from an academic endeavor, the team is definitely at a disadvantage in terms of the amount of money that can be spent on marketing and customer service.

However, the open-source strategy being adopted will help us overcome this hurdle. The team aims at revolutionizing the telehealth industry the way Linux revolutionized the mainframe operating system industry or the Android revolutionized smartphone operating system industry. By providing a free to use platform that can be leveraged by third party developers, the growth of a community of support and innovation is encouraged. This will be particularly successful because:

1) The proliferation of cheaper hardware and sensors is spurring the growth of independent original equipment manufacturers who can use the team's platform to build a complete solution.

2) With cheaper networking solutions and the growth of internet of things, several startups are developing devices each with their own proprietary communication protocols. Currently, there isn't a single platform that would allow these disparate devices to talk to each other. This reduces interoperability, which can stall growth of the industry a whole (Ghosh et al., 2011). As adoption increases, the open source platform the team is developing has the potential to allow for greater interoperability, thus further spurring the industry growth.
3) Existing companies have their proprietary platforms which can be licensed by third party developers at a hefty fee. By providing an open source solution, the team can win over more developers, thus gaining market share.

2.7.6 Threat of Substitutes

A substitute product is one that that offers similar benefits to customers of a company from other industries. The threat of substitutes describes the possibility that substitutes can replace the company's products. If the company is faced with a high-level threat of substitutes, it loses the control over the price that it can set to sell to customers, since customers can easily switch to its substitutes if the price is set too high. Therefore, it is important to analyze the threat of substitutes given it can affect profitability of the company and the industry. In this section, we will discuss the problem our technology tries to solve, what substitutes there are that solve the same problem, and what the level of threat the substitutes pose on our technology.

Health care is by far one of the most expensive industries in the United States. Readmissions lead to a huge hospital costs. According to Agency for Healthcare Research and Quality, there were 3.3 million readmissions in the United States, which cost hospitals about $41.3 billion USDs (Heins et al 2014). By developing a system that can reduce trips to the hospital, helping people get in touch with their own health, and mediate the link between patient and physician, we can cut down the waste in health care. Specifically, the problem that our technology tires to solve is to reduce hospital readmissions by reporting patients' vital signs (including heart rate, blood pressure, energy expenditure and body temperature) to physicians, who will in return give patients' health advice remotely to keep healthy.

Except our technology, traditional medical devices and modern technologies may achieve the same purpose and become our substitutes. Traditional medical devices are single functional devices such as thermometer, sphygmomanometer, EE monitors and etc. Patients can use those devices to record their vital signs by themselves and remotely consult physicians for feedback with recorded data.

Traditional medical devices have advantage on price over our product. They are relatively cheaper to Bluetooth-enabled medical devices required by our technology, which may lead patients to adopt traditional medical devices instead of our technology. Another weak point of commercializing our technology is switching cost. Since our technology is open-source, we should obey open source philosophy, which states that open-source software should be freely used, changed and shared (Open Source Initiative 2015:1). As a result, there is no reason to charge termination fees to patients when they decide switching to substitutes. However, our technology differentiates from these substitutes in three aspects which can stick patients to our product:

1. Instant data transmission.

   Our technology helps patients to transfer health data to physicians. If the patients use traditional medical devices to monitor their vital signs, they have to record the data and call or email physicians for advice. This procedure is both time consuming and boring. The patients may give up tracking their health conditions after several trials. With our technology, the health data gathered by Bluetooth-enabled devices will transmit to smartphones immediately and then transfer to physicians by cellular network. The instant health data transmission saves time for patients and physicians.

2. Easy access to health data.

   Our technology provides convenience for patients to check their vital signs record. Traditional medical devices are separate. To check vital signs history, patients need to check on individual devices. Our technology is based on smartphone. All measured vital signs will store on the phone. As a result, patients can check their health data anywhere anytime with their phone.

3. Expandable functionalities.

Our technology is expendable because we built an open-source platform for telemedicine. Developers can write more applications using our platform. Therefore, in the future, our technology has potential to provide more useful functionalities as well as benefits to patients and physicians by the contributions of third-party developers.

In addition to traditional medical devices, modern technologies could also pose a threat to the platform we are developing. Following are two existing modern technologies serve as substitutes of our product.

1. Motorola - Moto 360™

Motorola has recently released a smartwatch called the Moto 360™. Other than being an accessory to a smartphone, it can give basic biomedical data such as skin temperature, heart rate and energy expenditure. This platform could be developed further to come closer to the product we envision, but this would require substantial modifications to the current product.

2. Microsoft – Kinect

Microsoft's Kinect system has cameras which can be used to detect heart rate, energy expenditure and even mood by sensing the changes in flood flow in the face (Zhang 2012:4-10). Microsoft is currently focused on integrating the Kinect system with its Xbox One gaming console. However, this does not stop third-party developers from using this system to provide in-house medical diagnostics such as respiratory surface motion tracking (Alnowami et al 2012), 4D thermal imaging system (Skala 2011 et al:407-416), and otoneurological examinations (Dolinay et al 2014).

With considerations about price, switching cost, and differentiators with respect to traditional medical devices and discussion about modern technologies, the threat of substitutes of our product is moderate.

2.8.1 Major Conclusions

In this report, it was concluded that the best part of the industry for a startup centered about the technology being developed in our capstone is as a mediator between medical device companies and data services. This is the fat part of the chain. A Porter's Five Forces analysis was also conducted, and it was determined that with small tweaks to the planned dual-licensing strategy, all of the forces can be reduced to weak or moderate in strength. It is important to note however that this favorable analysis in no way guarantees success. As will be discussed in papers to come, the largest hurdles lie not within the placement of the startup, but with issues like avoiding IP prosecution, gaining the necessary industry approvals, and building up a reputation for being protective about sensitive health information. These issues are not easy, and some will be addressed specifically in future papers.

3. Intellectual Property Strategy

3.1.1 Introduction.

This section of the report is dedicated to explaining the strategy the team would like to adopt with respect to intellectual property. While the final goal is to create an open source product, the strategy is to attempt to patent the system architecture in order to safeguard the platform against patent trolls. We will first provide a brief outline of the technology, and then dive into the strategic reasoning behind patenting this particular part of the project in terms the competitive and strategic advantages it would give the team. We will then talk about what the risks of not patenting are and how these hurdles can be overcome.

3.2.1 The Aspect: System Architecture

Being the platform that forms the intermediary between data collection and analysis mandates robustness, fault tolerance and ease of use. For widespread adoption, not only is it necessary to be first to market, it is also necessary to provide an easy to integrate solution. The application programming interfaces that would be provided should be seamless enough for people in the adjoining industries to build upon. The product's system architecture caters to all these requirements. We believe the innovation that went into meeting the constraints imposed by the requirements puts the team in good stead to warrant a patentable system architecture. This architecture allows for an instant state recall, clever storage and persistent reconnection as broad means to implement fault tolerance and robustness - aspects that are quite novel and developed from needs specific to handling sensitive health data.

3.3.1 Problems with Software Patents

Software patents are notoriously difficult to obtain due to the difficulty involved in proving the novelty and obviousness of them, unless they are extremely specific. Common problems include

1. Deeming software is math, and math is not patentable (Jones, 2009).
2. Software programs being different than electromechanical devices because they are designed solely in terms of their function. While the inventor of a typical electromechanical device must design new physical features to qualify for a patent, a software developer need only design new functions to create a working embodiment of the program (Plotkin, 2002).
3. Computers are deemed to be the "designers" and “builders” of the structure of executable software. Software developers do not “design” the executable software's physical structure, but merely provide the functional terms (Plotkin, 2002).
4. The large number of micro-niches in software and relative few examiners mean patent examiners seldom have sufficient know how to recognize if the technology disclosed is innovative or novel enough (Bessen, Meurer, 2008).

For the reasons stated above, it may not be possible for a small team of engineers to mount a serious patent application. In order to overcome this, we plan on leveraging U.C. Berkeley’s Office of transfer office, the Office of Technology Transfer, to help design and formulate a strong application to provide the best chance of being awarded a patent.

3.4.1 The Strategy

Applying for a patent is best in this case as it would protect the business from the threat of patent trolls and also would set a legal basis by which the business can protect its work. Forgoing patents could spawn copycat businesses that would patent this architecture, and could ultimately lead to the business being coerced into paying licensing fees for technology it originally developed. Applying for the patent protects the product irrespective of the outcome of the application for two reasons:

1. Approval of the patent will ensure that other competitors do not develop similar technologies, and thus raise barriers to entry. While it is true that the business may ultimately pursue an open-source

strategy to encourage an organic growth of a strong third party community (which in itself raises barrier to entry), being awarded the patent ensures that the economic moat the team aims at building is deeper and wider.

2. Failure to get approval of the patent reassures the business that the same technology likely will not be awarded to a patent troll, giving peace of mind.

It is true that established competitors are much larger and financially capable than a startup this capstone group can produce, but the playing field can be leveled by leveraging the resources of the university to apply for a patent. This will both ease the financial burden, and cast a much larger enforcement net over the patent then could be done alone.

3.5.1 Risks of Not Patenting

If the backend architecture of the platform is not patented during the development phase, the team would leave itself open to the threat of having the architecture stolen and patented by another entity. As a consequence, the team could lose revenues from licensing, or even have to pay for technology the team invented in the first place. This could happen if some other group invented similar back-end architecture and patents it before the group's project is finished and released as an open source software. In addition, due to the highly collaborative nature of these projects, there is a risk that external developers would file a patent for the technology themselves. This would distract the team with crippling legal proceedings rather than focusing on further innovations.

Because of the risks of not patenting, it is advisable that patentable parts of the project are quickly pursued. To achieve this, an internal search utilizing university resources such as the Office of Technology Licensing (OTL) must be conducted to discover any prior art that may stand in the way of granting a patent. Legal experts should be then involved as quickly as possible to explore the potential patentability of the system architecture. Upon a thorough literature review by these experts, their advice will be taken and put into action.

In the case that it is found that a patent would be unlikely to be granted, the group would focus on making the platform open source as soon as possible in order to reduce the risk of infringement of patent rights. This establishes a record of prior art which can be used as a legal defense against patent trolls. A dual-licensing model will then be adopted to produce revenue. With dual-licensing, clients can use the technology under the condition that they redistribute their technologies as open source as well. If these entities wish to commercialize technologies utilizing this technology however, they will be required to pay a licensing fee on a per-use basis.

In conclusion, patenting the backend architecture provides a shield from legal problems and enables safe cooperation with external developers to speed up the development of the platform. In the unfortunate case that the patent is not granted, the only way to reduce legal risk is to release the platform as quickly as possible to establish prior art against potential litigation.

4. Individual Technical Contribution

Introduction

This paper is a report on my technical contributions for the Capstone Project titled **Expanded Telehealth Platform for Android.** In the first section of the paper, I will walk the reader through an overview of the project, and my specific contribution to it. In the second section, titled "Knowledge Domains", I will talk about the various literature and standards I used to help me make the decisions I took during the course of the project. In the third section, I will go into detail about the methods and materials used during the project. I will then walk the reader through the analysis of the pilot study conducted and the results we inferred from the study, followed by an overview of the next steps that should be taken.

4.1 - Project Overview and Context

A crucial component in any telemonitoring system is the transmission of the vital sign data from the source, i.e. the different vital sign monitoring devices, to the data analytics servers. Over the past decade the smartphone has become a vital part of our lives. The ubiquitous and versatile nature of the smartphone makes it a good candidate to act as the gateway of the communication channel that facilitates transmission of data from the vital sign-monitoring device to the cloud. It does this by collating the data from different vital sign monitoring devices and streaming the information via cellular data or WiFi to the cloud and to the server. At the server, the data is processed, and the inferences drawn are sent back to the smartphone for the user's consumption. This paradigm, therefore, splits the communication involved in telemedicine into three parts -

a. transfer of the data from the vital sign collecting devices to the smartphone
b. transfer of the data from the smartphone to the cloud.
c. transfer of analyzed data from the cloud back to the smartphone for user's consumption.

The first leg of communication is between the client device (i.e. smartphone) and the vital sign-monitoring device (i.e. remote device), whereas the second and third legs are between the server and client device. This is illustrated in Fig 4.1.1. In this part of the paper, I will walk the reader through my particular focus in our project - developing the communication between the remote device and the client device.

At the outset, there were two ways to perform this communication. One could adopt the wired connection approach where the vital sign monitoring devices are connected to the phone via wires, much like how we use earphones. The main advantages of using wires are robustness and reliability, i.e., guarantee of completion of transmission (Ashar, 2006) and data security. The only way to steal data from a wired link is to physically tap into the wire (Arbaugh, 2003:99-101) and this is not very feasible when the wires are strapped to you. However, wired communication, especially for the purposes of telemedicine, has many disadvantages.

First, it can be extremely cumbersome for the patient to continue daily activity while having wires attached to his body. Also, smartphones are not equipped with enough ports to support many vital sign monitoring devices working in tandem.

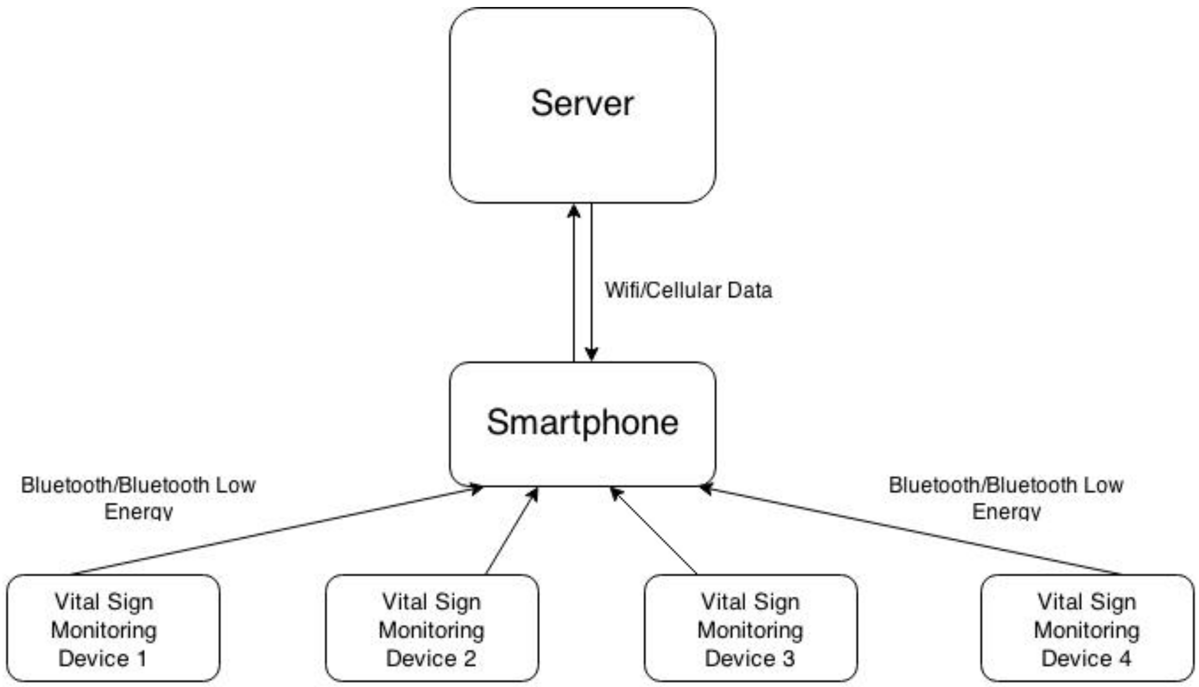

Fig 4.1.1 Telemonitoring System Architecture

These hurdles convinced us to adopt the wireless approach, and specifically, use Bluetooth and Bluetooth Low Energy to establish communication between the vital sign monitoring devices and the smartphone. Bluetooth was specifically chosen because it offers the following advantages:

a. Prevalence of the technology in the smartphone industry - Most, if not all, smartphones are Bluetooth enabled today (Bluetooth SIG, 2015).
b. Ability to establish a secure communication channel.
c. Support for simultaneous connection of several devices.

My task in this project was to develop Bluetooth and Bluetooth Low Energy support to establish transfer of data from the vital-sign monitoring devices to the smartphone. The team is building the telemonitoring platform on Android. The aim was to abstract the intricacies of Bluetooth and Bluetooth Low Energy development on Android to a more generic library that would make development for both technologies more consistent with each other. In addition to this, a second layer of abstraction was also developed in the form of Bluetooth Health Devices that would allow the application developer to seamlessly build applications for devices that monitor various vital signs and which adhere to the IEEE 11703 standard. This is discussed at length in section 4.3.3.3. Along with abstraction, there was a need to add fault tolerance to increase the robustness of the communication. Care was also taken to ensure the application programming interfaces (APIs) were generic enough to support communication between any type of Bluetooth enabled device and the smartphone.

This task forms the core of the telemonitoring system on the client side. It will be used, in the future, to help connect devices to the telemonitoring platform and extract data from them. Currently, my colleagues, Jochem van Gaalen and Quan Peng, are developing two independent algorithms to detect heart rate using different technologies currently embedded in the phone. Their algorithms will then be tested against readings extracted from a Bluetooth enabled Blood Pressure and heart rate monitor, which will connect to our telemonitoring platform through the Bluetooth stack we have developed.

4.2 Knowledge Domain

My technical contribution to this project involves abstracting the Bluetooth Library in Android to allow developers easily integrate their vital-sign monitoring devices into a tele-health platform. Thus, in this section I will first walk the reader through the motivation of using tele-health and then introduce Bluetooth as a technology.

High re-hospitalization rates amongst patients with chronic life-threatening conditions presents a severe financial burden to the health care system. For example, it has been estimated that approximately 50% of patients with chronic heart failure are readmitted to the hospital within 6 months of discharge (Giamouzis et. al., 2010: 922-930). The benefits of employing a tele-monitoring system for medical intervention has recently been studied and includes reduction in re-admission (Chaudhry et. al., 2010:2301-2309; Clark et. al., 2007; Giamouzis et. al., 2012; Inglis, 2010:228-228). Telemonitoring systems can be categorized based on

a) Type(s) of collected data
b) Modality and frequency of data collection
c) Means of data transmission
d) Type and number of devices used
e) Level of real time operation (Aranki et al., 2014: 135-141).

This project will be integrated into a telemonitoring system developed in University of California, Berkeley. This system currently delivers real time information on a patient's overall activity levels via energy expenditure estimates and self reported vital signs. A pilot study was conducted to evaluate the feasibility of the system in the real world and to identify privacy implications, usability and other challenges that healthcare providers encounter (Aranki et al., 2014: 135-141).

Bluetooth technology was introduced as a standard by Ericsson, IBM, Intel, Toshiba and Nokia. Through this technology, low-cost and power frugal ad-hoc networks can be formed where information can be transferred via short-range radio frequency waves (Bluetooth ISG, 2001).

Being a ubiquitously adopted technology, the infrastructure cost to use Bluetooth in telemedicine will be low. When dealing with communication between devices separated by small distances, Bluetooth seems to be the best technology. This is due to low power consumption, robustness and security inherent to the technology (Tafa, Stojanovic, 2005: 75-80). In addition to the above, Bluetooth also gives the added advantage of being able to coexist with other communication technologies, like ZigBee (ZigBee Alliance, 2007).

Using Bluetooth as the primary means of communication does have its disadvantages. These include lower throughput and transmission range compared to Wireless Large Area Networks (Tafa, Stojanovic, 2005: 75-80). There is also the limitation of only being able to use 7 slave devices or nodes (Kroc and Delic, 2003) simultaneously. However, given the nature of telemedicine, higher throughput is not a necessity. The data being collected from the vital sign monitoring devices is characterized by small size. Although they are frequently collected, each packet of data is small and hence the throughput required for reliable communication is low. Also, having more than seven devices constantly attached to the subjects body brings with it usability and comfort issues, and hence the cap on the maximum number of devices that can be connected simultaneously using Bluetooth is not a drawback.

To get comprehensive information about a person's health, it is important to monitor their vital signs continuously. Due to memory constraints, these monitoring devices cannot store a lot of data on board, and hence have to frequently transmit the data they collect to another device that has more memory (like a smartphone). Although, the classic Bluetooth technology is a low power technology, it can still be quite power hungry if used continuously.

Technologies like Bluetooth Low Energy (BLE), ZigBee, 6LoWPAN or Z-wave provide a good solution to the problem of power consumption. Bluetooth Low Energy is an emerging wireless technology developed by the Bluetooth Special Interest Group (Gomez, Oller and Paradells, 2012: 11734-11753) that is extremely frugal in terms of power consumption. In fact, Bluetooth Low Energy has been shown to be to be more power efficient than ZigBee in terms of number of bytes transferred per Joule spent (Siekinnen, Matti et al., 2012). It also integrates easily into classic Bluetooth circuitry and is more likely to be deployed in smartphones than other competing low power technologies (Gomez, Oller and Paradells, 2012: 11734-11753). Furthermore, also allows for secure data transmission, interoperability across devices and direct communication with Internet and cellular infrastructure, which are all prime requirements for medical applications (Omre, Helge and Keeping, 2010: 457-463).

There are three main advances in technology whose conception and adoption has fuelled the rise of telemedicine systems, and the importance of Bluetooth in it. These are

1. ISO/IEEE 11073 standard that defines the transportation-medium agnostic communication protocol to be adopted by medical, healthcare and wellness devices (IEEE, 2008). This standard set the precedent for interoperability of communicating devices in healthcare, thus spurring growth of telemedicine systems.
2. Conception of Bluetooth Health Profile by the Bluetooth Special Interest Group (which oversees the development Bluetooth standards). This profile defines the requirements that must be met by health-monitoring devices using Bluetooth communication to be considered an implementation of Bluetooth Healthcare and Fitness (Bluetooth SIG, 2012).
3. Extension of support for Bluetooth Health Profile in Android 4.0(Open Handset Alliance, Google, 2011).

The conception of Bluetooth Health Device Profile (HDP) by the Bluetooth SIG helped in growth of telemedicine because it:

1. Guaranteed interoperability between vital sign monitoring devices using Bluetooth communication that implement these requirements.

2. Increased interoperability at the application level through adherence to ISO/IEEE 11073 standard that defines the transportation-medium-agnostic communication protocol to be adopted by medical, healthcare and wellness devices (IEEE, 2008).
3. Increased connection fault tolerance. This profile improves robustness against connection failures owing to communicating devices moving out of range.

Google's support for Bluetooth Health Profile Devices, in its 4th major release of the Android Operating System, also increased the adoption of Bluetooth in telemedicine systems because

1. Developers could now create applications that would use the smartphone as the "sink" to collect data from monitoring devices using the Bluetooth HDP standard.
2. New devices that run Android as their operating system could now leverage the APIs provided by Android to act as a Bluetooth Health Device themselves. The explosion of the Internet of things (Amyx, 2014) will further increase the role of Bluetooth in telemedicine.

These advances all point to the widespread adoption of Bluetooth in telemedicine. Abstracting the Bluetooth Stack in Android to a library that can be easily utilized will definitely help manufacturers of vital sign monitoring devices and healthcare professionals develop systems and applications for telemedicine, despite lacking intricate knowledge of Android. Thus, I believe that my technical contribution to the project will have far reaching impacts on telemedicine as a whole.

4.3 Methods and Materials

Development of the library that would abstract Bluetooth communication was done in two broad phases – abstraction of classic Bluetooth and then abstraction of Bluetooth Low Energy.

The first step in development of the library to abstract Bluetooth communication was to understand the intricacies present in Android Bluetooth Development. And to do this, the first logical step was to develop a sample Bluetooth application that would allow two phones to send messages over Bluetooth to each other. I pursued this step to get acquainted with the APIs Android provides to developers and to understand the workflow one has to follow to establish communication. The end goal for this phase of the capstone project was to develop an application that would:

a. Turn on Bluetooth on the host device
b. Scan for nearby Bluetooth devices.
c. Pair with a "visible" device.
d. Connect to this newly paired device.
e. Send and receive text messages from connected device.

This step helped me understand all the subtleties of Android development, and helped me gauge which parts of the code were to be abstracted away into a library, whose APIs could be used by developers of a tele-health system. Once this step was completed, I was in a position to begin the implementation of our library. The library was developed on Android, with Eclipse IDE being the development environment.

For development, I adopted the asynchronous, object oriented programming paradigm. In this section, I will first discuss this paradigm, while simultaneously giving the reasons for adoption of this paradigm. Next, I will illustrate each of the concepts central to this method of programming using a simple example. After that, I will demonstrate the general workflow of Bluetooth and BLE communication. Finally, I will conclude with the different components of the library I have developed.

4.3.1) Object Oriented, Asynchronous, Programming Paradigm

Bluetooth communication, as a system, is largely event-driven. Be it scanning for available devices, connecting with them or sending/receiving messages or information to these devices, all the phases in Bluetooth communication are event driven. While it is possible for one to spawn different threads (for accepting connections, requesting a connection to be made or managing connections) and constantly monitor them, managing these threads can be difficult for the application developer. Thus, it was important to be able to hide the difficulties of managing a multi-threaded system and the associated thread safety problems in the library, and expose to the application developer only the occurrence of certain events – like discovering a new device, receiving a request for a new connection, receiving data etc.

It is also important to ensure that the application does not stall when processing a particular task. Once it has initiated a task on a thread, it should move onto processing other tasks. When the results of the first task are available, the thread that was processing it will inform the application of the same, and the application can now use the results of the task. This is particularly important when the application has a user-facing component (namely the user interface) to ensure a lag-free experience.

In order to cater to both of the above requirements, we adopted an asynchronous event based paradigm. Central to the concept of asynchronous, event-based, communication is the concept of callbacks. Essentially, a callback is a means of being notified of the completion of a certain task. It is often implemented as a function or method that is invoked on the occurrence of a particular event. This method of programming is also, arguably, simpler for programmers to develop bug-free, robust code (Dabek, et. al., 2002).

Further, keeping in line with Android, and the fact that the end goal of my technical contribution to the project was to develop a robust, fault tolerant library that would abstract away the complicated details of Android, I adopted object oriented programming during development. Object Oriented Programming(OOP) is a programming paradigm based on constructs called objects. Objects are data structures that contain data or properties that describe the object, and functions or methods that describe the behavior of the object and its functionality. Thus, these objects can be used to model a real world entity.

Object Oriented Programming encourages modular programming and promotes reuse of code efficiently. It also supports encapsulation wherein interdependencies between separately written modules are reduced by defining strict external interfaces. These interfaces serve as a strict contract between the modules and the clients, and thus analogously, between the designers of one module (in this case, the Bluetooth library) and the designers of another module (in this case, the application making use of the library) (Snyder, 1986). It also promotes data abstraction wherein the behavior of an object is fully determined by the set of abstract operations performed on the object, with the user not having to worry about the underlying implementations or representations of this object (Snyder, 1986). Last, but not the least, it supports inheritance, wherein an object can be created based on another (its parent), by "inheriting" the parent's attributes and behaviors, while adding its own.

4.3.1.1) An Example

To illustrate the concepts of asynchronous, Object Oriented Programming, lets use the example of a lawyer. We can define an object to be a human being and also create another object to be a lawyer. All lawyers are human beings, but not all humans are lawyers. This is because all the behaviors and attributes of a human are behaviors and attributes of lawyers. An example of a behavior of a human is the ability to talk or think. An example of an attribute of a human is age and sex.

Clearly these behaviors and attributes apply to all lawyers. However, certain attributes, like the discipline of law practiced and certain behaviors like arguing in court, are specific to lawyers and do not apply to all humans in general. Thus, we can say that lawyers can *inherit* from humans. Also, when a client and lawyer decide to pursue a particular strategy or tactic, the client does not need to know the nitty-gritties of how the lawyer is going to argue in court to represent him. That is the function of the lawyer, and has been *abstracted* away from the client. Furthermore, the concept of *encapsulation* can be illustrated by how the opposition views the lawyer and his client. As far as the opposition is concerned, the lawyer and his client are one unit, namely, their opponents. Thus, both the client and the lawyer have been *encapsulated* into one unit.

Also, let's assume that the lawyer has just defended his client against the opposition, and has submitted his terms to the them. The ball is now in the opposition's court, and the lawyer leaves the phone number with the opposition. The phone, now, functions as a *listener*. It constantly waits for a call from the opposition, while letting the lawyer carry on with other work. An *event* can be thought of as the opposition as either accepting the terms proposed, or rejecting the same. Once the *event* occurs, the opposition would get in touch with lawyer, informing him about their decision, through the *listener*.

This entire interaction typifies asynchronous communication as the lawyer is free to carry on with other work while opposition evaluates the terms (i.e. they are performing their own task). He is not "stalled", but will attend to an event as and when it occurs.

These advantages provided by asynchronous and Object Oriented Programming naturally led me to embrace the combination as the primary programming paradigm for development. In the next section, I will talk about the various design decisions I adopted, while explaining the different components of the library in parallel.

4.3.2) Workflow

4.3.2.1) Classic Bluetooth

The general workflow for Bluetooth communication consists of the following:

1. Enabling Bluetooth on host device.
2. Scanning for nearby devices
3. Pairing with an unpaired devices

4. Connecting with paired device.
5. Transmitting information to paired device, and receiving information from the same.

The host device in case of telemedicine will be the smartphone. During the setup phase, it would scan for nearby devices to see which vital sign monitoring devices (also referred to as node) are accessible (i.e. both publicly visible, and in range). These devices, initially unpaired, will be contacted, thus initiating a pairing process. This often includes typing a special password/key to authenticate the connection. Once this initial pairing has been done, the nodes will show up as paired devices to the host. Now, a connection has to be established between the two devices, and once this is done, the node and the host can freely transmit information to and from each other. Pairing gives the two devices engaged in the conversation the privilege of directly initiating a connection, without having to enter a password each time.

4.3.2.2) Bluetooth Low Energy:

In Bluetooth Low Energy, the remote device does not have to pair with the host device to establish communication. Rather, the remote device broadcasts data in the form of *Advertising Payload* and *Scanning Payload*.

The Advertising Payload is the mandatory data that the remote device (also known as the peripheral) transmits at definite intervals to alert host devices, like the smartphone, of its presence. The host device can then request more information from the remote device through a *Scan Response Request*. The remote device then supplies this information using the *Scanning Payload*. The diagram in Fig 4.3.1. illustrates this communication.

Once the host device has sufficient information, it can request a connection for direct communication to be formed with the remote device. After connection is established, the remote device stops broadcasting its data, and only transmits information to the host device it has established connection with.

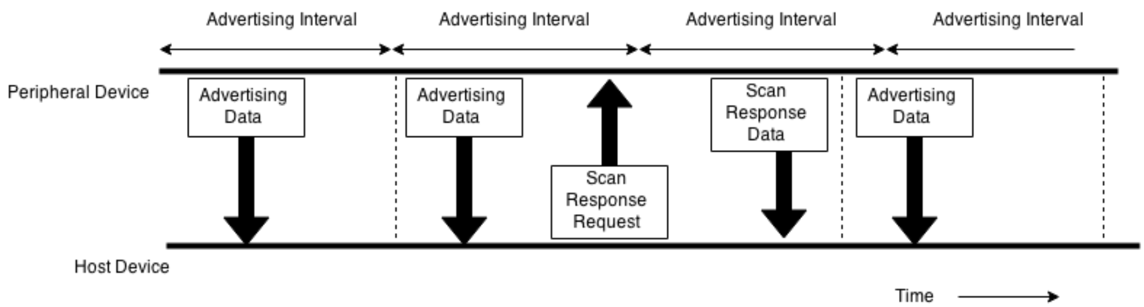

Fig 4.3.1: Bluetooth Low Energy Broadcasts

Direct communication is established through the exchange of packets of information known as Generic ATTribute (GATT) transactions. These are high level, hierarchical objects consisting of *Profiles*, *Services* and *Characteristics*. This is illustrated in Figure 4.3.2

Characteristic

The most granular chunk of data in the GATT Transaction, a Characteristic represents a single data point. For example, Blood Pressure Measurement Characteristic contains the values of the Systolic, Diastolic and Mean Arterial Blood pressure in one reading.

Service

A Service is represents a chunk of data corresponding to a logical entity. It consists of one or more Characteristics. For example, the Blood Pressure Service consists of Blood Pressure Measurement Characteristic (which contains the actual blood pressure reading), and auxiliary data about the features supported by the device (like if the monitor sends information about if it can detect body movement, or cuff fit etc.)

Profile

A Profile is a collection of Services that has been defined by Bluetooth SIG. For Example, the Blood Pressure Profile consists of Blood Pressure Service (that contains information about Blood Pressure) and Device Information Service (that contains information about the device, like the model number, firmware revision number etc.).

Once the host device receives a GATT packet, it parses it to understand the different Services supported by the remote device, and then further parses these Services to understand the different Characteristics available. It's within these Characteristics that the host device can finally extract the readings.

2

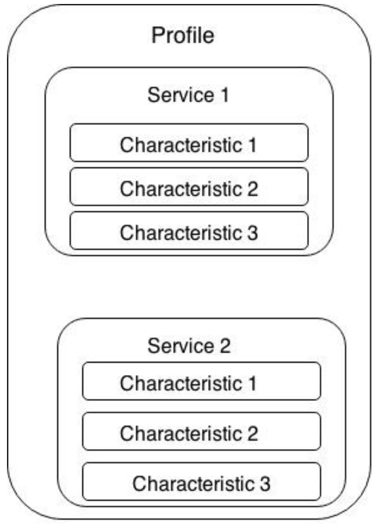

Fig 4.3.2: GATT Transaction

4.3.3) Components of Communication:

4.3.3.1) Bluetooth

There are six basic components involved in Bluetooth communication, namely:

1. Bluetooth Service - the service that runs on the host device and facilitates various requests like switching on Bluetooth, listening to incoming connections etc.
2. Unpaired Bluetooth Device - Any node device that has not paired with the host previously.
3. Paired Bluetooth Device - Any remote device that has previously paired with the host.
4. Bluetooth Sendable - an encapsulated message that is represents the information being transferred between the application and the library.
5. Bluetooth Converter - A converter interface that converts the data being transferred from its original form to the Bluetooth Sendable form and vice-versa.
6. Bluetooth Listeners – The library interacts with the application through these listeners. They act as the callbacks from the library, one for each event. Listeners have to be registered by the application and serve as the means of communication from the library to the application informing it of the completion of a task or the occurrence of a particular event.

In keeping with the spirit of OOP, we have designed individual classes to represent each of the above components.

4.3.3.1.1) `BluetoothService`

The Bluetooth Service is represented in our OOP design using the `BluetoothService` class.

This class represents the host device. It is the portal of entry, for the application engineer, into the Bluetooth library. It is responsible for switching on Bluetooth, ensuring host device discoverability, initiating a scan for nearby Bluetooth devices and pairing with unpaired Bluetooth devices.

On construction, it populates a list of all devices the host has paired with in the past. The user can then establish a new connection to any of these devices. In case he wants to establish connection with a new device, he will have to initiate a scan. On discovering every new device, the `BluetoothService` provides a call back to the application with the details of this device (as an object of `UnpairedBluetoothDevice` class). This includes the remote device alias and MAC address. The `BluetoothService` also allows the application programmer to initiate a pair request with the unpaired device. Once the pairing is successful, the `BluetoothService` constructs an object of `PairedBluetoothDevice` class representing the newly paired device.

This class also allows the application programmer to spawn the Accept Thread. The Accept Thread essentially listens for an incoming connection. On receiving a request for a connection, the thread examines if the connection is from a paired (and hence trusted) device, in which case it invokes a callback to the application to get permission from the user to accept the incoming connection. If the permission is granted, it returns a socket for communication with this device. Once it does this, it continues listening for other incoming connections. Requests for connection from unpaired devices are ignored, thus adding security to the telemonitoring system.

4.3.3.1.2) Paired and Unpaired Bluetooth Devices

Each remote/node device is represented in our OOP design though the `BluetoothDeviceWrapper` class.

Paired and unpaired Bluetooth devices can be thought of as extensions Bluetooth devices. They share a set of attributes and behaviors, common to all Bluetooth Devices. In keeping with the philosophy of OOP, these devices represented as `PairedBluetoothDevice` and `UnpairedBluetoothDevice` respectively, were designed to inherit from `BluetoothDeviceWrapper`. The class inheritance diagram is shown below.

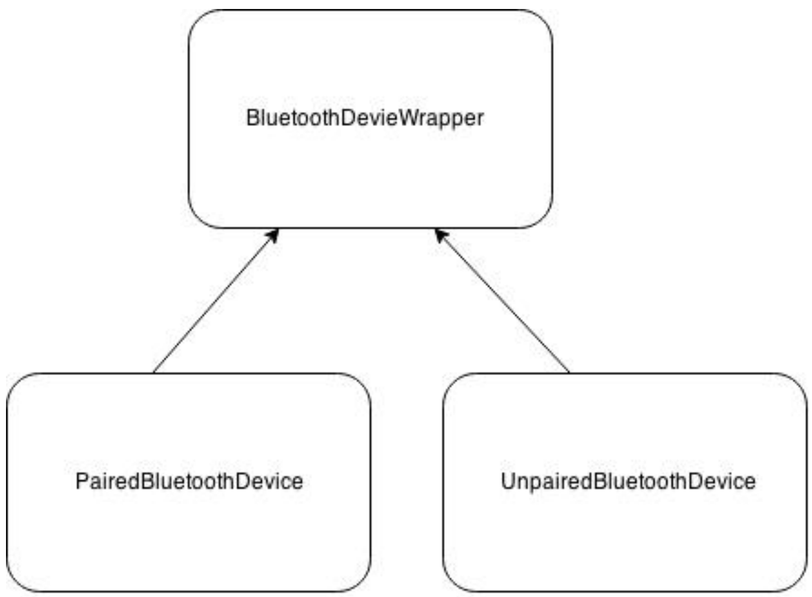

Fig 4.3.3 - Class Diagram for `PairedBluetoothDevice` and `UnpairedBluetoothDevice`

4.3.3.1.2.1) `PairedBluetoothDevice`

Each remote device the host has paired with will be an object of this class. The behaviors that set this device apart from an object of `BluetoothDeviceWrapper` are the abilities to initiate connection, the ability to transmit information to a remote device, and receive information from a remote device. For security purposes, establishing connection and communication can only happen with paired devices. This class provides the methods for writing (sending information) to a paired device. It provides a call back to the application on receiving information/data from the remote device the host device has established a connection with. If the connection with this device breaks, it will automatically try establishing the connection again. If remote device cannot be connected to after three attempts, this class provides a callback to the application indicating that the connection has been lost.

4.3.3.1.2.2) `UnpairedBluetoothDevice`

Each unknown device (a Bluetooth device the host has not paired with in the past) will be an object of this class. When the host device starts scanning for nearby Bluetooth devices, a callback is provided by `BluetoothService` returning an object of this class. The application programmer can then pair with this device using the methods (behaviors) provided in this class. Depending on whether the pairing is successful or not, an appropriate callback is provided to the application.

4.3.3.1.3) `UnpairedBluetoothDevice`

This class represents the kind of information that can be sent using the Bluetooth library. It is a generic serializable class. Its a custom format of our Bluetooth library, and is used by all the modules within this library.

4.3.3.1.4) BluetoothEventListeners

Each of the classes - `BluetoothService`, `PairedBluetoothDevice` and `UnpairedBluetoothDevice` - have their own listeners called `BluetoothServiceEventListenerInterface`, `BluetoothPairedDeviceEventListenerInterface`, and `BluetoothUnpairedDeviceEventListenersInterface` respectively. The inheritance tree for the listeners is shown in Fig. 4.3.4

The library exposes these listeners to the application as *interfaces*. These are binding contracts between the library and the application, and allow the library to send information to the application or notify the application of the occurrence of an event. The application is responsible for using this data, and has the privilege of using it in whatever manner it wishes. For example, assume the `BluetoothService` has spawned off an Accept thread, allowing the host device to listen to incoming connections. If any remote device wants to connect to the host, it would send a connection request. The library receives this request, and relays the same to the application through the callback, with details about the device that has requested the connection. The application can now decide if or not the library should form the communication channel to allow the remote device to transfer data with the host device.

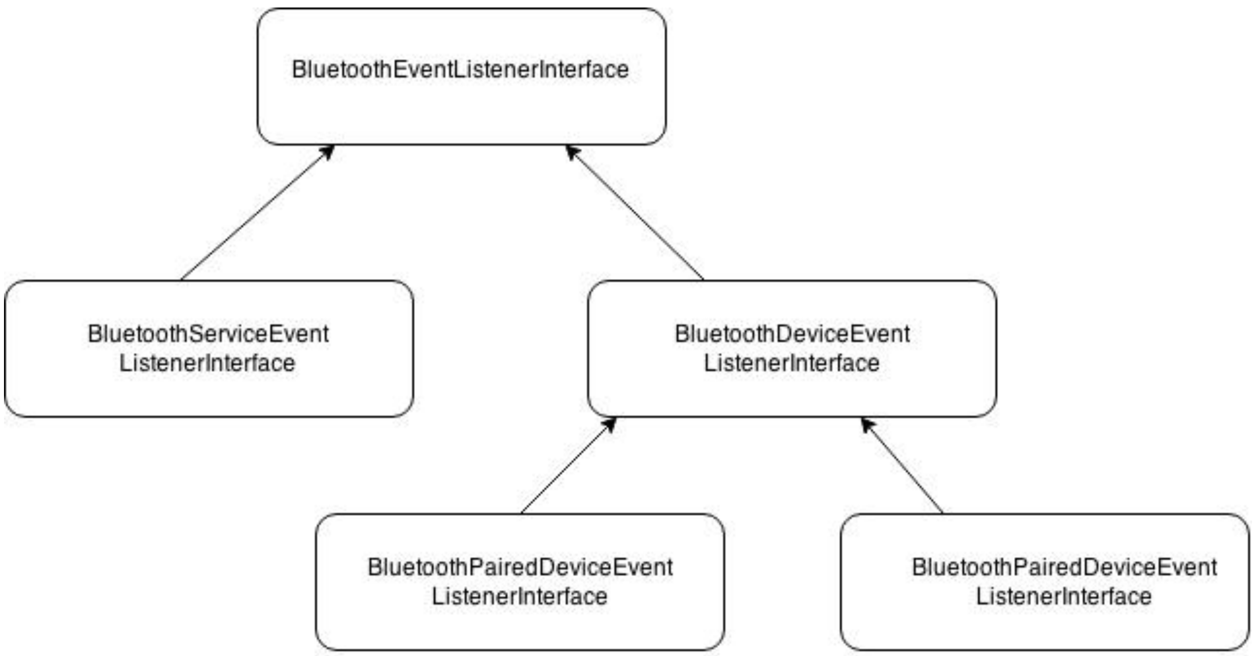

Fig 4.3.4 Class Inheritance Diagram for the Event Listeners

This inheritance design decision was a conscious one taken to adhere to OOP paradigms and mimic the real world. We have one parent `BluetoothEventListenerInterface` class. This is inherited by the `BluetoothServiceEventListenerInterface` and the `BluetoothDeviceEventListenerInterface` class. The `BluetoothDeviceEventListenerInterface` class now is inherited by `BluetoothPairedDeviceEventListenerInterface` and

`BluetoothUnpairedDeviceEventListenerInterface` classes. This is in keeping with the philosophy that all instances of `PairedBluetoothDevice` and `UnpairedBluetoothDevice` are still children of the `BluetoothDeviceWrapper` class. Also, because they share some attributes, this design allows us to define certain interface elements that are common to both `BluetoothPairedDeviceEventListenerInterface` and `BluetoothUnpairedDeviceEventListenerInterface` in `BluetoothDeviceEventListenerInterface`.

4.3.3.1.4.1) `BluetoothServiceEventListenerInterface`

This interface contains the callbacks that will be made from the `BluetoothService`. These include:

1. `onFinishedScanning` – Callback to indicate that the service has finished scanning for Bluetooth Devices.
2. `onObtainedOneUnpairedDevice` – Callback to indicate a new device was discovered.
3. `onSwitchingOnBluetooth` – Callback to indicate Bluetooth was successfully switched on.
4. `onSwitchingOffBluetooth` – Callback to indicate Bluetooth was successfully switched off.
5. `onIncomingConnection` – Callback to indicate a remote device is attempting to make a connection with the host.

4.3.3.1.4.2) `BluetoothUnpairedDeviceEventListenerInterface`

This interface contains the callbacks that would be made from an object of the `UnpairedBluetoothDevice` class. These include:

1. `onSuccessfulPair` – Callback to indicate that the remote device was successfully paired with.

2. `onFailurePair` – Callback to indicate that the remote device was could not be paired with.

4.3.3.1.4.3) `BluetoothPairedDeviceEventListenerInterface`

This interface contains the callbacks that would be made from an object of the `PairedBluetoothDevice` class. These include:

1. `onConnect` – Callback to indicate successful connection with the `PairedBluetoothDevice` device.

2. `onReceive` – Callback to indicate that a packet of information has been received from a `PairedBluetoothDevice` with which connection was established.

3. `onFailureRead` – Callback to indicate that the data from the `PairedBluetoothDevice` with which connection was established was not read completely

4. `onFailureWrite` – Callback to indicate that the data meant to be written to a connected device was not written successfully.

5. `onConnectionFailure` – Callback to indicate that the connection established with the `PairedBluetoothDevice` failed. This callback has further information to inform the application if the connection failed after a communication channel was established, or during the initial connection request itself.

4.3.3.1.5) `BluetoothConverters`

The `UnpairedBluetoothDevice`, being a custom format that is understandable only by the modules of the library, necessitates the need for a Bluetooth converter module that converts the incoming and outgoing data from the proprietary format of the node device to the `UnpairedBluetoothDevice` format. Since the format of the data provided by node depends only on the node, the implementation of this conversion is left to the application developer/medical device manufacturer. Therefore, the `BluetoothConverters` are provided by the library as interfaces that have to be developed by those using this library.

For example, if one was developing for a Bluetooth-enabled Blood Pressure Monitor that implements the IEEE-11703 protocol, he or she should design this `BluetoothConverter` extract the raw data from the incoming packet, using the protocol, and create a new `UnpairedBluetoothDevice` with this raw data.

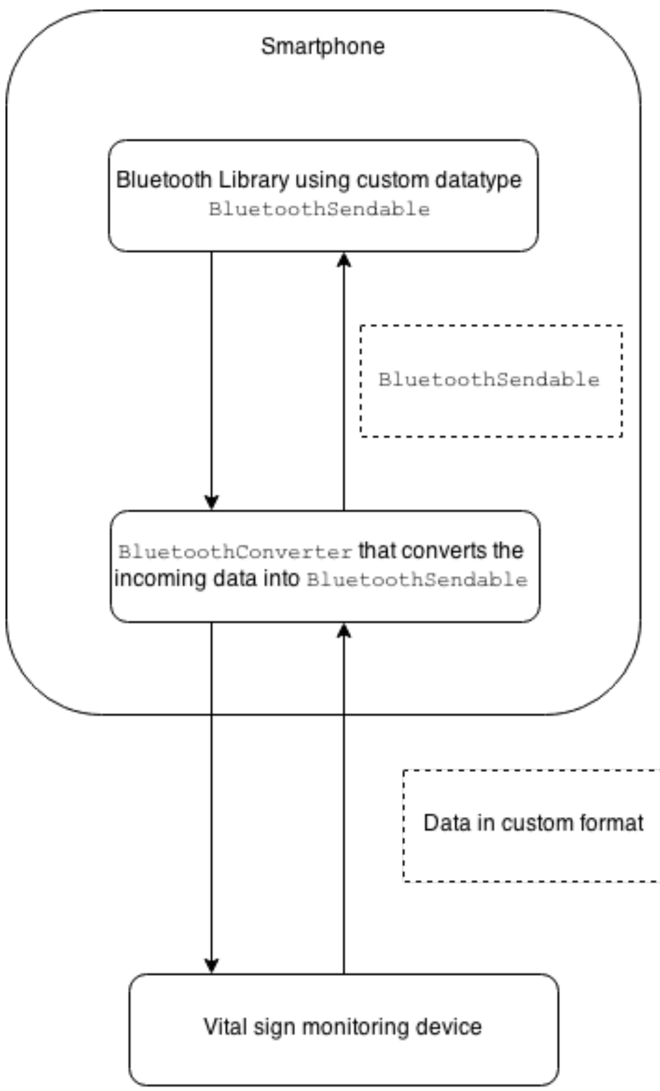

Fig 4.3.5 `BluetoothConverter`

4.3.3.2) Bluetooth Low Energy

Because Bluetooth Low Energy does not have the concept of pairing, and because the data being transmitted is already packaged into logical entities, the communication is simpler to implement. Consequently, the number of components required to establish this communication is lesser. The main components are:

1. Bluetooth Low Energy Service: Responsible for scanning for nearby remote devices, by listening to the Advertisement Payload that is being broadcast by the remote devices.
2. Bluetooth Low Energy Device: The OOP class modeling the remote device that allows you to read from and write to it.
3. Bluetooth Low Energy Listeners: The library interacts with the application through these listeners. They act as the callbacks from the library, one for each event. Listeners have to be registered by the application and serve as the means of communication from the library to the application, informing it of the completion of a task or the occurrence of a particular event.

4.3.3.2.1) Bluetooth Low Energy Service

The Bluetooth Low Energy Service is represented in our OOP design using the following class: `BluetoothLEService`.

This class represents the host device. It is the portal of entry into the Bluetooth Low Energy library for the application engineer. It provides application-programming interfaces to discover remote devices by scanning for the Advertising Payload broadcast by them. For each such discovery, this service calls back the application with information about the remote device that was discovered.

4.3.3.2.2) Bluetooth Low Energy Device:

The Bluetooth Low Energy device is represented in our OOP design using the following class: `BluetoothLEDevice`.

This class represents the remote Bluetooth Low Energy device, and provides APIs to connect to the remote device, read the Services and Characteristics supported by it, instruct the remote device to send notifications to the host device when the value of any Characteristic has changed, and read a particular Characteristic.

4.3.3.2.3) Listeners:

There are two interfaces corresponding to the callbacks invoked by `BluetoothLEService` and `BluetoothLEDevice`.

4.3.3.2.3.1) BluetoothLEServiceEventListenersInterface:

The interface includes callbacks for:

1. `onBeginScanning` – Callback to indicate that the scan to discover remote Bluetooth Low Energy devices has begun.
2. `onFoundLEDevice` – Callback to indicate that a remote Bluetooth Low Energy device was discovered. It provides the application with information about this remote device.
3. `onFinishedScanning` – Callback to indicate that the scan to discover remote Bluetooth Low Energy devices has completed.

4.3.3.2.3.2) BluetoothLEDeviceEventListenerInterface:

The interface includes callbacks for:

1. `onDeviceConnected` - Callback to indicate that the communication channel with the remote device was successfully established.
2. `onFinishedDiscoveringServices` - Callback to indicate that a list of all Services and Characteristics supported by the remote Bluetooth Low Energy device was obtained.
3. `onReadCharacteristicValue`: Callback to indicate reception of a packet of information from the remote Bluetooth Low Energy device. This callback also contains information about whether this packet was sent because a Characteristic value was altered and hence the remote device is notifying the host device, or if the packet was sent in response to a request from the host device to read a Characteristic value.
4. `onDeviceDisconnected` – Callback to indicate that the communication channel with the remote device was severed.

4.3.3.3) Bluetooth Health Devices

The above implementations of Bluetooth and Bluetooth Low Energy provide simple APIs to scan for, connect with, and transfer information to and from remote devices serving any purpose. Different devices follow their own protocol built on top of this Bluetooth and Bluetooth Low Energy technology to communicate information. These protocols package the data transmitted by the device in a proprietary manner. Therefore, the onus lies on the application developer to decipher this proprietary packetizing mechanism to extract the underlying data.

However, a special class of devices exists in the market that employs Bluetooth or Bluetooth Low Energy technology to monitor various health and fitness parameters. These include Bluetooth-enabled blood pressure monitors, heart rate monitors, thermometers, weighing scales and oximeters. Health monitoring devices have inherent requirement interoperability, and the guidelines provided by IEEE 11703 standard allow devices that adhere to this standard to freely communicate with each other. These guidelines delineate how the data transmitted from the health device should be packaged.

We were able to leverage this uniformity amongst Bluetooth-enabled vital sign monitoring devices to design an abstract layer representing this class of devices called the Bluetooth Health Device. We baked the ability to decipher the packaging of the packets and extract the raw data into the library. This layer contains the following components:

1. `BluetoothHealthDevice` – the model that represents the remote health-monitoring device.
2. `BluetoothHealthDeviceListenerInterface` – the interface that library provides to the application through which it communicates the extracted vital sign data.

4.3.3.3.1) `BluetoothHealthDevice`

This is the OOP model representing the class of devices that monitor vital sign and adhere to the IEEE 11703 standard.

Through this class, the application developer can connect to the remote device, even when it is not in the vicinity of the host device. The library guarantees that the connection will be established when the remote device becomes available. It does so by continuously scanning for it, and establishes communication when the device comes in range. The library then configures the remote device to notify the host when the value of the vital sign the device is monitoring changes. Once it receives this information, the library parses the it, extracts the underlying value of the vital sign, and invokes the appropriate callback to the application with extracted value.

4.3.3.3.2) `BluetoothHealthDeviceListenerInterface`:

It consists of the interfaces provided by the library through which it sends the extracted data points to the application. The interfaces include:

1. `BluetoothBloodPressureListenerInterface`: Interface that contains callback to indicate that a new Blood Pressure reading has been received by the host device.
2. `BluetoothHeartRateListenerInterface`: Interface that contains callback to indicate that a new Heart rate measurement has been received by the host device.
3. `BluetoothTemperatureListenerInterace`: Interface that contains callback to indicate that a new temperature data point has been received by the host device

As can be seen, the Bluetooth Health Device abstraction makes it very simple for the application developer to create an application to read data from the vital sign monitoring device. The application first has to indicate to the library that it would like to connect with a particular health device. Additionally, the application may have many components that have to be notified when a particular vital sign reading is received. For this, it has to register different listeners for each of the components that have to be notified, for each of the vital signs. This is done through the interface provided by the library. For instance, if there are two components of the library that are interested in being notified when the Blood Pressure changes, the application will register two listeners which implement the

BluetoothBloodPressureListenerInterface. When the blood pressure monitoring device becomes visible, the library will automatically establish communication and listen for any information sent by this device. On receiving said information, the library then informs each of the components interested in the information through the list of listeners registered with it by the application.

4.4 Pilot Study and Results

This library was integrated into a test application that was used in a pilot study. The purposes of the pilot study were three-fold:

1. To verify robustness and scalability of the Bluetooth Library
2. To verify the accuracy of the algorithms implemented by my teammates Jochem van Gaalen and Quan Peng.
3. To test usability and acceptability of the telemonitoring platform.

The pilot study was conducted for a duration of 24 hours. In the pilot study we requested the participants to first fill a questionnaire before using the telemonitoring system (henceforth referred to as pre-study questionnaire). The details of this questionnaire are discussed in the Section 4.4.1. Next, the test application was installed on their individual phones. This application would log their caloric expenditure automatically. The participants were encouraged to measure their heart rate several times using the two algorithms implemented. They were also requested to come into a controlled environment and measure their Blood Pressure using the A&D UA-651 BLE Blood Pressure Monitor[3] at least once during the study. This monitor communicated with the pilot application through the BluetoothHealthDevice stack I developed. The measurement from this monitor would serve as the ground truth to how accurate the implemented algorithms were. At the end of the 24 hours, they were asked to fill the another questionnaire (henceforth referred to as post-study questionnaire) with the same questions as those in the pre-study questionnaire. All along, the application would transmit their vital-sign information in a secure manner to a server where the data would be logged.

[3] http://www.aandd.jp/products/manual/medical/ua651ble_en.pdf

While my colleagues will discuss the results of the accuracy measurements in their individual reports, I will discuss the results of the robustness and scalability of the library and the usability and acceptability of the platform in subsequent sections.

4.4.1) Scalability:

To test scalability, we deployed the application on many different smartphones. We noticed that in all smartphones, the application was able to establish connection with the Bluetooth Low Energy enabled blood pressure monitor. While we were able to extract the heart rate measurements using the application on some phones, on a few phones, the data could not be extracted. On initial inspection and research from the online community, we suspect this anomalous behavior could be because of the differences in the Hardware Abstraction Layers (for Bluetooth) provided by different Original Equipment Manufacturers(OEMs) of the different smartphones. The Bluetooth Low Energy technology is still nascent in Android and has gone through a few iterations. Its inconsistency has been documented in the online community, where the same code works for many devices, but not for all. Since Android caters to a large number of devices from different OEMs with different Hardware Abstraction Layers, testing the same technology on each of these devices would take time and consistency will be established as the technology matures.

4.4.2) Robustness:

To test robustness, we deployed the application on one smartphone and used it to extract multiple blood pressure data points from the UA 651-BLE. We were successfully able to read the data from the device on all attempts which testifies to the robustness of the solution.

4.4.2) Usability and Acceptability:

For the telemonitoring system to be widely adopted, it was important to measure how easy it would be to use the system (i.e. usability), and also how comfortable people would be with divulging different types of information like their vital-sign data (i.e. acceptability). We measured this by studying the responses of the pre-study and post-study questionnaires. To study usability, the questionnaire had subjective questions inquiring about the participants' comfort levels with smartphones, Bluetooth and health-monitoring devices. It also inquired about how easy the participants thought using the telemonitoring system was. To study acceptability, the questionnaire had subjective questions about how comfortable they would be with sharing different vital sign data with different entities (family, friends, social network, insurance companies, healthcare professionals, government agencies, and other private companies). By comparing the responses for each of the questions between the pre-study questionnaire and post-study questionnaire, we were able to see if the telemonitoring system changed the participants perception of telemonitoring.

On examining the two sets of responses, there wasn't a major shift in the perception of telemonitoring amongst the participants. In both questionnaires, all participants, almost unanimously, agreed that they would change their lifestyle if more personal health data was available to them. They also indicated that they prefer avoiding answering subjective questionnaires about their health on their phone. Although most of them hadn't used Bluetooth enabled product to collect and monitor their health stats, they were open to doing so in the future. The participants did not mention any difficulties in using the sample application.

With respect to acceptability, we saw the same trends in both the pre-study questionnaire and the post-study questionnaire. The participants were willing to share information about their vital signs with family and healthcare professionals but unanimously said they would not want to share this information with private companies, insurance providers and government agencies.

4.4.3) Ease of Extension of the `BluetoothHealthDevice` library:

In addition to the above results, it must also be noted that using the Abstracted Bluetooth Library considerably shortened development time to extend support for a new vital-sign measuring device. Using the framework of the `BluetoothHealthDevice`, I found the development time to extend support to integrate a Bluetooth Enabled Thermometer (Pyle Health PHTM60BT[4]) was considerably shorter than the development time to extend support for the A&D UA651BLE Blood Pressure monitor.

[4] http://www.pyleaudio.com/sku/PHTM60BTBL/Bluetooth-Non-Contact-Infrared-Handheld-Thermometer-with-Digital-LCD-Display-Readout

4.5 Conclusion

At the onset, my technical contribution to this project was to build a library that would abstract away the nuances of Android Bluetooth and Bluetooth Low Energy Programming. I have successfully achieved this goal. The library I have developed has a simple API that an application developer can easily build on. The library also provides an abstraction layer in the form of Bluetooth Health Devices for all vital sign monitoring devices that adhere to the IEEE 11703 format. This further simplifies companion application development for these devices.

Fault tolerance and reliability are important requirements for any telemonitoring application. All communication systems encounter occasional errors caused due to channel disturbances, interference, hardware failures etc. In such a scenario, it is of utmost importance that the data that is to be communicated is not lost.

To protect against these errors when communicating data from the smartphone to the server, our application stores data locally, and securely, till successful communication is complete. Additionally, the connection between the smartphone (host device) and the vital sign monitoring device (remote device) might break due to interference or because the devices temporarily move out of range. In this case, the library tries to automatically reestablish communication with the remote device. If the reconnection does not succeed in three successive tries, the communication channel is severed, and the devices have to explicitly reestablish a connection to resume communication. This gives the application the liberty to decide what is to be done when the remote device has moved out of range.

However, in telemonitoring applications, it is important that the communication channel between the remote device and the smartphone be established as soon as possible to prevent the loss of data (these remote devices have limited memory, and hence cannot store all measurements). Therefore, a conscious decision was made while designing the Bluetooth Health Device to constantly seek to establish connection with the remote device to re-facilitate transmission of data.

In terms of security when using Bluetooth, the library ensures that only devices paired in the past will be able to connect and transmit data to the host device. Connections from other devices are ignored.

All these efforts put together make it much easier for an application developer to build a telemonitoring system. He or she can now use the easy to understand APIs provided by the library to construct the system without needing prior knowledge in the workings of Bluetooth on Android. I believe this would also help in the migration of various current wired medical applications to wireless transmission, or migration of traditional radio frequency transmission ((S. Rhee, B.H. Yang and H.H. Asada, 2001: 795-805), (J. Soi et al., 2005 : 485-497) to Bluetooth enabled transmission.

From the questionnaires in the pilot study, it can be seen that the participants did not like to answer subjective questions about their health on a daily basis. This is an important observation, as it can be a leading cause in the decrease of willingness of the patients to use the telemonitoring system over time. Automated, objective measurements using vital-sign monitoring devices that do not need patient intervention would ensure this "usage fatigue" does not grow. Although most participants hadn't used a Bluetooth-enabled vital sign monitoring device to monitor their health, they were open to the idea of using one. This, coupled with the fact that most participants said they would change their lifestyle if more personal health data was available to them, shows that there is a willingness to adopt this technology in the near future.

With regards to our capstone project, the library I have developed forms the core of the communication between the smartphone and the vital sign monitoring devices.

4.6 Next Steps

Scalability needs further investigation, for which it becomes necessary to necessary to test the library on a wide array of smartphones running Android and Bluetooth enabled health-monitoring devices. A more objective and extensive study has to be conducted to determine how much easier our library makes it for application developers to build applications for vital-sign monitoring devices.

In addition to this, the `BluetoothHealthDevice` currently supports measurement of Blood Pressure and Temperature. It can be further extended to be able to extract data from more vital-sign monitoring devices like pulse oximeters, activity trackers, weighing scales etc.

Furthermore, the `BluetoothHealthDevice` currently has extensive support for Bluetooth Low Energy. The same amount of support is yet to be extended for Bluetooth.

While the focus of this project has been to enhance the Bluetooth development experience for the purpose of telehealth, the same Bluetooth stack and platform can also be used to build many applications in the field of the Internet of Things. For example, using the same Bluetooth stack one can build a “Connected Home” where several appliances, from the refrigerator, to the television to the coffee maker, can be controlled from just your smartphone. Home security can also be further enhanced. The field of fitness technology has already received a significant through Bluetooth and Bluetooth Low Energy, with companies like Fitbit, Jawbone, Apple, Motorola etc. making several fitness trackers and smartwatches that keep track of ones’ health. The same kind of boost can also be expected in retail, where Bluetooth Low Energy devices can be used to enhance the shopping experience, through faster checkout, tailor-made discounts etc. All application developers who develop applications that provide Internet of Things solutions to these fields can use our Bluetooth Library to more seamlessly integrate Bluetooth connectivity into their products

5. References

Ahmed, Ashar, et al.
Wired vs wireless deployment support for wireless sensor networks."
TENCON 2006. 2006 IEEE Region 10 Conference. IEEE, 2006.
Accessed: April 14, 2015

Alnowami, M., et al.
A quantitative assessment of using the Kinect for Xbox360 for respiratory surface motion tracking.
SPIE Medical Imaging. International Society for Optics and Photonics, 2012.

Amyx, Scott
2014, Why the Internet of Things Will Disrupt Everything.
http://www.wired.com/2014/07/internet-things-will-disrupt-everything/
Accessed: April 14, 2015

Arbaugh, William A.
"Wireless security is different."
Computer 36.8 (2003): 99-101.
Accessed: April 14, 2015

Aranki, Daniel et. al.
Continuous, real-time, tele-monitoring of patients with chronic heart-failure: lessons learned from a pilot study.
Proceedings of the 9th International Conference on Body Area Networks. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2014.
Accessed: May 10, 2015

Banet, Matthew J., et al.
Body-worn sensor featuring a low-power processor and multi-sensor array for measuring blood pressure. U.S. Patent Application 13/346,408.

Banet, Matthew J., et al.
Vital sign monitor for cufflessly measuring blood pressure using a pulse transit time corrected for vascular index. U.S. Patent Application 14/072,305.

Bessen, James and Meurer, Michael J.
2008, "Patent Failure: How Judges, Bureaucrats, and Lawyers Put Innovators at Risk, Princeton University Press, http://press.princeton.edu/titles/8634.html, Accessed March 02, 2015.

Bluetooth Special Interest Group
Specification of the Bluetooth System, 2001
Available at http://www.bluetooth.org/specs
Accessed: April 14, 2015

Bluetooth Special Interest Group
Health Device Profile, 2012
Available at
https://www.bluetooth.org/en-us/specification/assigned-numbers/health-device-profile
Accessed: April 14, 2015

Bluetooth Special Interest Group
Skyrocketing demand for Bluetooth appcessories for latest phones, 2015
Available at
http://www.bluetooth.com/Pages/Mobile-Telephony-Market.aspx
Accessed: April 14, 2015

Chaudhry, Sarwat I., et al.
Telemonitoring in patients with heart failure.
New England Journal of Medicine 363.24 (2010): 2301-2309.
Accessed: May 10, 2015

Centers for Medicare & Medicaid Services.
"National Medicare Readmission Findings: Recent Data and Trends." *Office of Information Product and Data Analytics* (2012).

Clark, Robyn A., et al.
Telemonitoring or structured telephone support programmes for patients with chronic heart failure: systematic review and meta-analysis.
Bmj, 2007.

Dabek, Frank, et al.
Event-driven programming for robust software.
Proceedings of the 10th workshop on ACM SIGOPS European workshop. ACM, 2002.
Accessed: April 14, 2015

Dolinay, Viliam, Lucie Pivnickova, and Vladimir Vasek
Objectivization of traditional otoneurological examinations based on Kinect sensor Hautant's test based on Kinect. *Control Conference (ICCC), 2014 15th International Carpathian*. IEEE, 2014.

Ghosh, Rajib; Heit, Juergen; Srinivasan, Soundararajan
Telehealth at scale: The need for interoperability and analytics
International Conference on Information and Knowledge Management, Proceedings, p 63-66, 2011, *CIKM 2011 Glasgow: MIXHS'11 - Proceedings of the 1st International Workshop on Managing Interoperability and Complexity in Health Systems*;
Accessed February 16, 2015

Giamouzis, Gregory, et al.
Impact of dopamine infusion on renal function in hospitalized heart failure patients: results of the Dopamine in Acute Decompensated Heart Failure (DAD-HF) Trial.
Journal of cardiac failure 16.12 (2010): 922-930.
Accessed: May 10, 2015

Giamouzis, Gregory, et al.
Telemonitoring in chronic heart failure: a systematic review.
Cardiology research and practice 2012 (2012).
Accessed: May 10, 2015

Gomez, Carles, Joaquim Oller, and Josep Paradells.
"Overview and evaluation of bluetooth low energy: An emerging low-power wireless Technology."
*Sensors*12.9 (2012): 11734-11753.
Accessed: April 14, 2015

Hall, Joseph L., and Deven McGraw.
For telehealth to succeed, privacy and security risks must be identified and addressed. *Health Affairs* 33.2 (2014): 216-221.

Hines, A. L., Barrett, M. L., Jiang, H. J., & Steiner, C. A.
Conditions with the largest number of adult hospital readmissions by payer, 2011 (2014).
http://www.hcup-us.ahrq.gov/reports/statbriefs/sb172-Conditions-Readmissions-Payer.pdf, accessed February 16, 2015

IEEE
IEEE Std 11073-20601, 2008 Health Informatics - Personal Health Device Communication - Application Profile - Optimized Exchange Protocol
Accessed: April 14, 2015

Inglis, Sally.
Structured telephone support or telemonitoring programmes for patients with chronic heart failure.
Journal of Evidence−Based Medicine 3.4 (2010): 228-228.
Accessed: May 10, 2015

Jones, Pamela
2009, "An Explanation of Computation Theory for Lawyers"
http://www.groklaw.net/pdf2/ComputationalTheoryforLawyers.pdf Accessed March 02, 2015.

Kroc, Srdjan, and Delic, Vlado.
Personal wireless sensor network for mobile health care monitoring.
2003, Telecommunications in Modern Satellite, Cable and Broadcasting Service TELSIKS 2003. 6th International Conference on.
Vol. 2. IEEE, 2003.
Accessed: April 14, 2015

Lars, Nile
2014 Connected Medical Devices, Apps: Are They Leading the IoT Revolution — or Vice Versa?
Wired
http://www.wired.com/2014/06/connected-medical-devices-apps-leading-iot-revolution-vice-vers
Accessed Feb 27, 2015

Miller, Edward Alan.
"Solving the disjuncture between research and practice: telehealth trends in the 21st century." Health Policy 82.2 (2007): 133-141.

Morea, Stephan
2014 IBISWorld Industry Report OD5775: Telehealth Services in the US.
http://clients1.ibisworld.com/reports/us/industry/default.aspx?entid=5775, Accessed February 16, 2015

Omre, Alf Helge, and Steven Keeping.
"Bluetooth low energy: wireless connectivity for medical monitoring."
Journal of diabetes science and technology 4.2 (2010): 457-463.
Accessed: April 14, 2015

Open Handset Alliance, Google
Android introduces Bluetooth Health Profile, 2011
http://developer.android.com/about/versions/android-4.0.html
http://developer.android.com/reference/android/Bluetooth/BluetoothHealth.html
Accessed: April 14, 2015

Open Source Initiative
Welcome to The Open Source Initiative. http://opensource.org , accessed February 16, 2015

Plotkin, Robert
Intellectual Property and the Process of Invention: Why Software is Different, 2002, International Symposium on Technology and Society,
http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1013821, Accessed March 02, 2015

Porter E., Michael
2008 The Five Competitive Forces That Shape Strategy. Harvard Business School Publishing Corporation. https://hbr.org/2008/01/the-five-competitive-forces-that-shape-strategy. Accessed February 16, 2015

Skala, Karolj, et al.
4D thermal imaging system for medical applications. *Periodicum biologorum* 113.4 (2011): 407-416.

Snyder, Alan.
Encapsulation and inheritance in object-oriented programming languages. *ACM Sigplan Notices*.
Vol. 21. No. 11. ACM, 1986.
Accessed: April 14, 2015

Seo, Jinwoo, et al.
The development of a nonintrusive home-based physiologic signal measurement system.
Telemedicine Journal & e-Health 11.4 (2005): 487-495.
Accessed: April 14, 2015

Siekkinen, Matti, et al.
"How low energy is bluetooth low energy? Comparative measurements with Zigbee/802.15.4."
Wireless Communications and Networking Conference Workshops (WCNCW), 2012 IEEE.
Accessed: April 14, 2015

Sokwoo Rhee, Boo-Ho Yang, and Haruhiko Harry Asada,
Artifact- Resistant Power-Efficient Design of Finger-Ring Plethysmographic Sensors
IEEE Transactions on Biomedical Engineering
Vol. 48, No. 7, July 2001, pp. 795-805.
Accessed: April 14, 2015

Swan, Melanie
2012 Sensor Mania! The Internet of Things, Wearable Computing, Objective Metrics, and the Quantified Self 2.0, Journal of Sensors and Actuators, Page 217-25, http://www.mdpi.com/2224-2708/1/3/217/htm#sthash.e0RU9ljv.dpuf, Accessed Feb 27,2015

Tafa, Zilbert, and Radovan Stojanovic.
Bluetooth-based approach to monitoring biomedical signals.
WSEAS Transactions on Business and Economics 2.3 (2006): 75-80.

Zhang, Zhengyou.
Microsoft kinect sensor and its effect. *MultiMedia, IEEE* 19, no. 2 (2012): 4-10.

Zigbee Alliance
White paper on ZigBee and Wireless Radio Frequency Coexistence, June 2007.