The Design and Implementation of a
Mobile Healthcare System with AppleWatch For Seniors
(MyHealth)

By

Arij Atallh Alfaidi

B.A King Saud University 2010

A thesis submitted to the graduate faculty of the
University of Colorado at Colorado Springs
In partial fulfillment of the
Requirement for the degree of
Master of Science in Computer Science
Department of Computer Science
Spring 2016
This thesis for Master degree by

Arij Alfaidi

Has been approved for

Department of Computer Science

By

_______________________
Advisor: Dr. Edward Chow

_______________________
Committee member: Dr. Rory Lewis

_______________________
Committee member: Dr. Jonathon Ventura

_______________________
Date
Abstract

In this thesis, we developed and implemented a mobile Healthcare app, called MyHealth, on the iPhone with Apple watch to help seniors to manage their health life. MyHealth reminds the users to take their medicine and records the dosage they take. MyHealth measures the heart rate and notifies the user’s emergency contacts when the heart rate becomes above or under a certain threshold. The notification can be a call, SMS, email, or a combination of them. The users choose the contacts when he sets up the MyHealth app. MyHealth reminds the users to move and exercise and it tracks the user movement. MyHealth records the data that collected from the heart rate measurement and sends these data to the user’s physician weekly to monitor their health. Preliminary tests show the MyHealth is achieving all its functionality the watch is measuring the heart rate each 30 minutes if it found that the heart rate is above or under the normal level it notifies the user and then it alerts his emergency contact by calling and texting. Medicine reminder notification shows on iPhone primarily and the watch if the iPhone was locked the user can choose if he took the medicine that this task is complete or should the App reminds him in 30 minutes. There are Notifications to remind the user to move twice a day if he can choose complete the App will start tracking if he moves his legs or not as if he walks or runs or he can delay this for 30 minutes also. By the end of the day the user should press the Email button to send his heart rate measurement reads to his doctor by entering his Email address.
Acknowledgements

I want to thank My Family Father, mother, sisters and brothers for their prayers and support. My Husband Saud for his support and my daughter Laura for inspiring me by giving me a smile each time I feel I couldn’t do it anymore. Her smile was the reason that gave me the power and energy everyday.

I would like to thank Dr. Edward Chow for his supervision and help the whole time. His encouragement, his ideas was a much of a help.

Also, I would like to Thank Dr. Rory Lewis and Dr. Jonathon Ventura for being interesting in my thesis idea and providing useful feedbacks.
Table of Contents

Chapter 1  Introduction ........................................................................................................... 1
  1.1 Goal of the Thesis .......................................................................................................... 4
Chapter 2  Background ......................................................................................................... 6
  2.1 The Apple watch .......................................................................................................... 6
  2.2 History .......................................................................................................................... 8
Chapter 3  MyHealth Design ............................................................................................... 11
  3.1 System Architecture ................................................................................................... 12
  3.2 iOS Development ....................................................................................................... 12
  3.3 Swift 2.2 ....................................................................................................................... 12
  3.4 AppGroup ................................................................................................................... 13
  3.5 Healthkit ..................................................................................................................... 14
  3.6 MyHealth Interface ..................................................................................................... 15
  3.7 MyHealth Design ....................................................................................................... 20
Chapter 4  Implementation ................................................................................................. 24
  4.1 HealthKit Framework ................................................................................................. 26
  4.2 Core Motion Framework ............................................................................................ 35
  4.3 Watch Connectivity Framework ............................................................................... 46
Chapter 5  Performance Evaluation of MyHealth App ...................................................... 48
Chapter 6 Lessons Learned and Challenges ..................................................................... 50
Chapter 7  Future Directions ............................................................................................. 53
Chapter 8  Conclusion ......................................................................................................... 55
References .......................................................................................................................... 58
Appendix A.  Installation and Demo ................................................................................... 60
Table of Figures

FIGURE 3.1. MYHEALTH SYSTEM DIAGRAM ................................................................. 11
FIGURE 3.2. TURN ON APP GROUP. .................................................................................. 14
FIGURE 3.3. TURN ON HEALTHKIT.FRAMEWORK .................................................. 15
FIGURE 3.4. USER INTERACTIONS WITH IPHONE MYHEALTH SYSTEM ...................... 16
FIGURE 3.5. USER INTERACTIONS WITH APPLE WATCH INTERFACE ....................... 18
FIGURE 3.6. DATA FLOW DIAGRAM DIAGRAM OF HEART RATE MEASUREMENT .......... 19
FIGURE 3.7. INITIAL MYHEALTH SCREEN ................................................................. 20
FIGURE 3.8. REMINDER SCREEN ................................................................................. 21
FIGURE 3.9. EMERGENCY CONTACT SCREEN ............................................................. 22
FIGURE 3.10. WATCH APP DISPLAY .............................................................................. 22
FIGURE 3.11. HEART RATE MEASUREMENTS SCREEN .......................................... 23
FIGURE 4.1. MYHEALTH FILES AND TARGETS INSIDE THE XCODE ......................... 25
FIGURE 4.2. IPHONE FILES INSIDE XCODE ............................................................... 25
FIGURE 4.3 ADD MEDICINE SCREEN .......................................................................... 42
FIGURE 4.4 ADD EMERGENCY CONTACT .................................................................... 43
FIGURE 5.1 MYHEALTH PERFORMANCE ON MEMORY .............................................. 48
FIGURE 5.2 MYHEALTH PERFORMANCE ON CPU ..................................................... 49
FIGURE 10.2 HEALTH DATA ACCESS APPROVAL ................................................... 61
FIGURE 10.3 MYHEALTH WATCH ASK TO USE HEALTH DATA ................................. 62
FIGURE 10.4 MYHEALTH ACCESSING HEARTRATE DATA ....................................... 63
FIGURE 10.5 HEART RATE MEASUREMENT ................................................................ 64
FIGURE 10.6 HEART RATE WARNING NOTIFICATION .............................................. 64
FIGURE 10.7 CALLING EMERGENCY CONTACT ........................................................ 65
FIGURE 10.8 TEXTING EMERGENCY CONTACT ......................................................... 66
FIGURE 10.9 MEDICINE DUE NOTIFICATION .......................................................... 66
FIGURE 10.10 HANDLING THE NOTIFICATION ......................................................... 67
FIGURE 10.11 MOVING NOTIFICATION ...................................................................... 68
FIGURE 10.12 TRACKING THE MOVEMENT NOTIFICATION ..................................... 69
FIGURE 10.13 EMAIL HEART RATE VALUES ............................................................ 70
Chapter 1
Introduction

In the contemporary world, a lot of technological inventions are being developed at fast rate. In this regard, many businesses are using the technological development to improve their functions and operations for achieving the required levels of output. Many people can afford to manage their social and private lives using different technologies like personal digital assistants among others [Diva2013]. Technology has improved the manner in which information is developed and shared [ONei2013]. The development of the Smartphone seems to have had a big impact in the social and economic dimensions of life. With these smart devices, people can manage their emails and social networking from their convenience. Developers have focused on creating many essential functions that people often seek. Developers have not stopped at that, thousands of mobile applications are being to influence business practices, governance, and investment. It is important to point out that the beneficiaries of these technological have not been businesses alone, the healthcare industry is also benefiting heavily from technology.

The healthcare is an urgent society problem. With the increase of elderly population, the demand on senior healthcare has been increased. Statistics shows that Between 2012 to 2050 the seniors population will increase considerably. They said by 2050
the elderly population to the people age from 65 and above will be around 83.7 million while it was about 43 million in 2012 [JOVVH2014]. The GreatCall Health agency reported a statistic shows that the mobile healthcare industry will be projected a 26-billion-dollar industry by 2017. There are a lot of health and Fitness applications that have been downloaded daily by users all over the world. To be able to manage your health from your phone is a promising idea.

Especially we would like to remind senior people of their medication and to assist them to contact their health providers easily.

When people become old, the level of activity tends to reduce considerably and need more support from other people. Additionally, [Yoo2013] explains that many elderly people tend to have health issues that need constant attention from medical practitioners. When these elderly people visit hospitals and are given medicines, they often forget taking them as required. This observation is common among many elderly people, who in most instances, prefer to rest for long hours with little physical activities. When these people do not have someone to monitor their progress, they can succumb to the old-age health complications. Perhaps this may explain why many people prefer having their elderly family members taken to nursing homes where they can be taken care of by people specially trained for that.

According to [ONei2013], taking the elderly to nursing homes is not always a better option, because most seniors like to be taken care of by their own relatives. In this regard, taking them to the nursing homes makes them feel lonely and stressed up, something that can
aggravate their health conditions. This understanding has sparked the need of having special ways in which, these elderly people can be monitored by doctors from the comfort of their homes without necessarily having aides. Many developers have been active in the research and development of various applications that can be used to achieve this important goal [Ozda2012].
1.1 Goal of the Thesis

My goal is to design, implement, and evaluate a mobile application that could help seniors to manage their health life in their daily life at home. By reminding them to take their medicine, measuring their heart rates, and remind them to move and track their movement. Also, help the physicians to track the health data of their patients by sending the heart rate data from patients to them daily. Also, the physician will know if their patients are taking their medicines as subscribed. These data will help to improve the patient health issues. Also these data can be used in research if the approval from a patient is obtained.

My thesis question is:

“Can I design an accurate and effective application using the Apple HealthKit with the Apple Watch?”

Since the correctness and timely reporting of health information by the mobile device is critical to the user’s health. It is critical that the design of mobile healthcare app with Apple Watch be conducted with extra care, rigorous process, and thorough experiments. The difficulties are further complicated by the fact that there are few resources for the Apple Watch. The challenge here is to use the Apple Watch sensors in my health application and to send these data back to the iPhone. Also, notifying the emergency contact based on the measured health information in real time is still a new idea to try using Healthkit on the Watch. Related research questions include who to notify and in what order, and what are the information to reveal. This document is organized as follows: Chapter 2 provides a survey of background information and related literature. Chapter 3 presents the design of MyHealth. Chapter 4 shows the implementation details.
of MyHealth. Chapter 5 discusses the performance of MyHealth app. Chapter 6 includes the lessons learnt. Chapter 7 suggests the future directions. Chapter 8 is the conclusion. Chapter 9 the references. Appendix contains the installation guide and demo steps.
Chapter 2

Background

Currently, hundreds of applications have been developed with the aim of managing some important tasks and resources like time and information, access and maintenance of health records, consulting and communications, training, education and decision-making among many others. However, those are not the focus of the proposed research.

Different companies are currently researching on how they can tap into the business opportunity that exists in the healthcare industry. Healthcare practitioners acknowledge the fact that the use of mobile devices has been instrumental in improving the health of patients. The mobile apps also help doctors in their clinical decision-making.

Apple has in the recent past made huge strides in this sector. The development of the Apple watch has been regarded as one of the best innovation in healthcare [Guar2015].

2.1 The Apple watch

The Apple Watch has been described as the big thing in health care [Guar2015], having features that bring the doctor virtually at the patient’s home, especially for the seniors in the society. The Apple Watch comes with an accelerator and heart rate monitor, which are beneficial in calculating the level of general fitness for the patient or person wearing it. The device has the ability to augment the particular information by linking with the iPhone.
Using Wi-Fi networks and GPS satellites, the device can also gather information that is used in determining its specific location at any point and any place.

The data from the Apple watch are provided by the two built-in sensors, accelerators and infra ray sensor. One reports the movement of the body. The other reports the expansion or shrinkage of blood vessel. The data collected by the accelerator sensor will then used by the apps to derive one’s workout and activities. The data collected by the infra ray sensor will be used by the app to decide the heart rate. This device enables the user of the watch to track the intensity of jogging and gym activities as well as walking and other exercises being done in the neighborhood. On the other hand, the sensor devices are quite passive. It tracks light exercise session like movements that the user makes during the day. These exercises may have been done within a minute or 12 hours. The activity and workout apps work alongside the Healthkit framework and health app, which have been announced by the company as having been incorporated in its iOS 8 [Guar2015].

Apple develops the HealthKit to enable sharing data between the new health app and between the apps that provide health and fitness. Also, by using the HealthKit any app that provides health and fitness services can share date among each other. Managing the data from different sources and merging the data from all sources based on the user choice is the HealthKit design functionality. Apps who are using the HealthKit are able to access the data from any source and do their own changes or merging’s on them. The user of the main health app can choose which data to share because he is sure that his health information is safe and secure in the Apple health database. The HealthKit also provides storage service that synchronizes the local health data on apple watch with those in iPhone and iCloud. [Appledeveloper].
Another important feature of the Apple Watch is that it can easily monitor sleep patterns and heart rates in the most convenient way [Kise2011]. Other essential tests that can be done including blood and urine where data can be entered by the user and can be managed through the kits and the resulting information linked to the iPhone. The phone can perform an analysis of the data and store it for future use, or the data can be sent to the doctor for further treatment.

The owner of the iPhone can have a highly comprehensive profile relating to their health. In this regard, he can determine his lifestyle patterns and activities that affect his life on daily basis. This understanding is important in making decisions that can be used to improve the health in different ways [Appledeveloper]. By developing mobile apps with Apple Watch, we hope to contribute to the developer community and enable others to develop app in this area. The current idea is to use the HealthKit to measure and record the heart rate and notify the elderly emergency contact in case of emergency.

2.2 History

Using mobile technology in the health area is not a new idea. The need for using mobile application has been increased with the increase of the cost of health insurance and staying in hospitals and nursery. A lot of studies focused on using mobile health application in different health situation. For monitoring patient health life, diagnose health issues and for research and study areas. Some of these studies has a good results and some of them did not achieve what they looking for. The cost of time and the critical of health experiments extent of the result of using health application. In my search I tried to focus on some of these experiments that made a difference in mobile Healthcare area. The first study was called "The use of personal digital assistants in clinical decision making by health care
professionals: a systematic review.” *Health informatics journal.* In this study they made an experiment in the emergency room at hospital they gave the patient a smart phone with an application to page his doctor in case of pain. They wanted to record how fast the doctor response and Did the patient use the application or not. Over all result was some time the patient does not have the time or energy to use the application in case of pain so I thought how can we do that automatically. in case of emergency we should let the application sense and call for help.

The second paper that I get used of it is called “Apple Watch has designs on health industry – but is it good for doctors?” this paper discusses in detail how Apple watch sensors designed and how the watch is related to Health application. The conclusion of the paper was promising that the Apple watch future will make a difference in mobile healthcare industry. That’s why I choose my application to use the Apple watch even though it’s a new feature and it’s hard to find resources to help implement but maybe this thesis will be the base for future studies that may make a improvement in the mobile health area.

“25 ways to use your smartphone. Physicians share their favorite uses and apps.” This paper shows how smartphone application helped to improve health in the society. They made the users more aware of the disease and their effects on the patients. They gave the patient the knowledge about the medicine that they use and what their side effect. The mobile health application gave the user the aware of infection disease that are spread on a specific time and how can they avoid it. It helped the physician to keep track of their patient health history. It also discussed how the application that used on the clinic’s computers made it easier to know what the patient history but this area is not my thesis interest. So, as a result of this paper the mobile healthcare application can make a difference we can get a useful result and this is my attention from my application.
In this paper “Applying surgical apps: Smartphone and tablet apps prove useful in clinical practice”. They made a study on senior people in a nursery they train them to use some application and they record the result of using these application. They said that these application make the patient more aware of their health issues and they became more careful with their health. Also, they discuss the use of Artificial intelligent inventions on Operation room and this is will be the future if the developers invest on it.

While I am reading in mobile health industry area I found this research paper that called “The smartphone in medicine: a review of current and potential use among physicians and students.” In this paper they made 60 study on the impact of using smartphones on health area. How can we used of the phone feature such as Wi-Fi, GPS to make application that can help users to improve their health life away from hospital?

“The meaning of information technology (IT) mobile devices to me, the infectious disease physician.” In this paper they studied why we should focus on seniors because they have more health issues and the population of elderly people in increase yearly. So, to produce application that can help them improving their health life is sparkle idea. That’s why I wanted to make seniors my target of my MyHealth application.
Chapter 3

MyHealth Design

The basic functions of MyHealth application are:

1. The user can add their medicine by: name, start Date, End Date.
2. The application notifies the user when the time is due to take the medicine.
3. The watch will read the heart rate of the user every thirty minutes.
4. If the heart rate above or below the normal level, the iPhone notifies the emergency contacts.
5. The application reminds the user to move and do some activity and tracks his movement.
6. The application collects the heart rate data and sends them to the user’s physician weekly.

Figure 3.1. MyHealth system diagram
3.1 System Architecture

MyHealth system consists of iPhone and apple watch running MyHealth mobile app as show above in Figure 3.1. It consists of an iOS app running on the iPhone, a WatchKit app running on the apple watch, some data are saved in iCloud. The software is developed using XCode IDE version 7.3 with Swift 2.0 as programming language. HealthKit Framework is the major component of MyHealth app. AppGroup is used to share data.

3.2 iOS Development

In the mobile platforms, a large market share is held by the iPhone. The process of development includes choosing the proper hardware for the applications. Creating the software requirements and the final system requirements are also part of the process. Apple Inc. has made it easier for people who want to become developers with them. All a person needs are to establish an Apple developer account charged $99 annually, a mac computer, Xcode platform and a coding experience with a programming language, such as Swift and Objective-C.

3.3 Swift 2.2

The apple Inc. announced a modern programming language called swift 2015. It is specifically developed to be more precise and flexible than the Objective-C. Swift has LLVM (Low Level Virtual Machine) compiler framework that utilizes the Objective-C
runtime allowing C, C++, Swift code, and Objective-C to run in a single program. The flexibility of Swift is attributed to its capacity to support widespread late binding, dynamic dispatch and extensible programming. It is also considered safe and can effectively manage the common programming errors, for example, the syntactic and the null pointers to prevent occurrence of unwanted results [Appledeveloper].

Swift 2.0 is an open source programming language with better performance than the same software created using the old Objective-C. It improves the error handling, support availability checking, and it’s feel more natural with the enhancements to the Apple SDK [Appledeveloper].

3.4 AppGroup

AppGroup is used to share data between different applications or in different interfaces of the same application. To use AppGroup you have to turn on the Appgroup in your application and on the extension of your application, the extension of MyHealth application here is the Apple Watch extension as a target cause I want the app to be in the iPhone and the Watch. So if you want to share data you have to turn on the Appgroup on the iPhone files and the extension that you want to share data among them. [Tom2014] Figure 3.2 shows how to turn on the App Group on Xcode IDE.
After you create your app go to Capabilities the shift down to turn on the App Groups as the figure above shows.

### 3.5 Healthkit

Apple developed Healthkit to enable developer to develop health and fitness application. By using Healthkit with the customer approval you can share the health Data on Apple health Database to your application. You can use these health data, make your changes and calculation on it, which can make your application a valuable health source. You can use Healthkit in your application by turning on the Healthkit icon on Xcode that will give a lot of options in your code. Figure 3.3. shows how you turn on Healthkit feature.
3.6 MyHealth Interface

In my thesis study I implemented a health application on iPhone and apple watch that has the name “MyHealth”. The application will have two interfaces. The user interacts with each interface differently.
Figure 3.4. User interactions with iPhone MyHealth system.

Figure 3.4 shows how the User will interact with iPhone interface. The user should use the iPhone to Add his medicine, Add emergency contacts and deal with the notifications. As the figure shows, first when the user downloads the App the system will send a request notification to use Apple health data. If approved, then the user should use the app easily. So the first interface the App will provide Add Medicine form when the user add his medicine start and End date the app will send him a notification when the medicine due. The user has two choices with this notification if he took the medicine he can choose Complete Otherwise He will choose the button that will let the App to remind him in 30 minutes. When the user wants to choose or enter his Emergency contacts the Add Emergency contacts form will be shown to the user and when he done entering his emergency contacts he will choose save these data will be saved to Appgroup. So when the
heart rate becomes more or under the normal level the App will call and text these emergency contacts.

There will be Email button by the end of day the when the user clicks this button the heart rate measurements read will be sent to the user physician.

The App will send notification to move to the user so, when the user choose start button the App will track his movement if walks or runs. The App will send the user notification that he is running or walking now.
Figure 3.5. User interactions with Apple Watch interface.

Figure 3.5 shows the User interact with Apple watch interface. It will read the heart rate and show what is the heart pulse. The user can choose either to save data or kill this action. Also, the notification will show on the watch to remind the user of his medicine. If
he took the medicine he can choose Complete or he can choose Remind me in 30 minutes. Movements reminder notification will be shown on the watch if he chooses start the iPhone app will track his movements or he can choose to delay this notification 30 minutes.

Figure 3.6. Data Flow Diagram diagram of heart rate measurement

Figure 3.6 shows the main feature of MyHealth application. To measure the heart, rate the user should wear the watch. The application will send notification to the user that it will measure the heart rate. In case of the heart rate below or up the normal level the App will send the notification to the emergency contacts. Otherwise it will save the read and repeat the process after 30 minutes. When the measurements start the watch sensor green
led light will be shown on the user rest and the pules will be shown on the watch screen with heart animation that moves. I choose 30 minutes between each reads cause my App idea is to monitor the senior heart health so the minimum time that maybe the read will change is 30 minutes. So if there is a change in his heart condition we can discover it early.

3.7 MyHealth Design

Since the target is elderly people in my design I choose to make the interface simple, direct, easy to use. When you first run the app, it will show the first interface screen as shown in Figure 3.7.

![MyHealth App](image)

Figure 3.7. Initial MyHealth Screen.

Then it asks the user for approval to use the health data. If the user approves, The Application can enter the user health data and uses the healthkit as it is supposing to do. If
not the App can not do anything. As it is shown on the bottom there are two tabs. If you click the Reminder button this page in Figure 3.8 will be shown:

![Reminder Screen](image)

**Figure 3.8. Reminder Screen.**

In this page, the user should write his medicine name. The start date and time and the End date and time for his medicine. The Repeat drop down list it will give you the option to repeat the reminder notification for a week, month, year.
Figure 3.9. Emergency Contact screen

As it is shown on Figure 3.9 the contacts button allows the User to choose or enter his emergency contacts. The user can choose two of his contacts to let the App call them in case of emergency. The user can choose three of his contacts to text and five contacts to send email to them of his Health data and in case of emergency.

Figure 3.10. Watch App Display
When the App run on the watch first it will show the message shown in Figure 3.10 to test that the notification will be sent. When the user chooses First button, it will Lunch the heart rate measurement window as shown in Figure 3.11.

When the App is measuring the heart rate this window will be shown on the watch to record the heart rate of the user. At this time the sensor will be green led that’s mean the App is measuring the heart rate.

The design is direct and simple each button can say by reading it what it will do.
Chapter 4

Implementation

To implement MyHealth application, I used:

- Macbook pro with El Capitan Operating System.
- Xcode 7.3 platform.
- Swift 2.0 programming language.
- iPhone 6 simulator.
- Apple Watch 38 mm simulator.

In my Xcode development I used:

- Watchkit as target
- Different swift files.
- AppGroup.
- Healthkit.
- CoreMotion to track the movement.

Inside the Xcode application there are three main folders: MyHealth contain source code, resources, and profile information for the iPhone app;

MyHealthWatch contains the storyboard GUI interface file and the settings for the watch interface;
MyHealthWatchExtension contains the source code, resources, and profile information on the Watch.

![Figure 4.1. MyHealth files and targets inside the Xcode.](image1)

MyHealth which is for the iPhone files
MyHealthWatch which is for the watch interface.
MyHealthWatchExtension which is for the result on the Watch.
The files under MyHealth folder are:

![Figure 4.2. iPhone files inside Xcode.](image2)
These are the files that will be shown on the iPhone to implement the notifications list and the contacts list and Medicine reminder list.

### 4.1 HealthKit Framework

- Apple released Healthkit with iOS 8 in 2014.
- Healthkit can help share and retrieve these data: Calories Burned, Sleep, Blood sugar, Cholesterol, Medications, Nutrition, Heart Rate, Weight, and Fitness.
- You have to take the user approval before you can have an access to healthkit.

**Classes:**

- **HKObject:** is the superclass of all HealthKit sample types.
- **Each object has the following properties:**
  - **UUID.** A unique identifier for that particular entry.
  - **Metadata.** A dictionary containing additional information about the entry.
  - **Source Revision.** The source of the sample. The source can be a device that directly saves data into HealthKit, or an app.
  - **Device.** The hardware device that generated the data stored in this sample.

On the `FirstViewController.swift` we handle the Watch measurement notification of the heart rate each 30 minutes and track the movements.

```swift
func triggerHeartRateWarningAlert()
{
    let currDate = NSDate()
    let notification: UILocalNotification = UILocalNotification()
```
This method will be called before the watch do the measurement to wake the watch and make her do the measurements.

```swift
func triggerHeartRateNotification()
{
    let heartRateCrossedLimit : Bool = fetchHeartRates()
    if heartRateCrossedLimit {
        let currDate = NSDate()
        let notification: UILocalNotification =
            UILocalNotification()
        notification.fireDate = NSDate(timeIntervalSinceNow: 30)
        notification.category = "FIRST_CATEGORY"
        notification.alertBody = "Heart Rate reminder"
        notification.fireDate = currDate
    }
    notification.fireDate = NSDate(timeIntervalSinceNow: 30)
    This will trigger the notification after 30 minutes.
    notification.alertBody = "Heart Rate Warning!!!"
    This is the notification that will be shown to the user.
}
```

When the heart rate is above or below a certain level, a notification will be shown to the user. To inform him that his heart condition is not good and the app will call for help. This is the function that handles that.
```swift
func showAlertMessageOnCrossingHeartRate()
{
    let alert = UIAlertController(title: "Heart Rate Warning", message: "Heart rate doesn't seem good, Sending Text messages to emergency contacts, Reset HeartRate in healthKit", preferredStyle: UIAlertControllerStyle.Alert)
    // add an action (button)
    alert.addAction(UIAlertAction(title: "OK", style: UIAlertActionStyle.Default, handler: {action in
        // Phone Notification
        self.phoneNotificationHandler()
        self.makeACall()
    })))

    // show the alert
    self.presentViewController(alert, animated: true, completion: nil)
}
```

<table>
<thead>
<tr>
<th>let alert = UIAlertController(title: &quot;Heart Rate Warning&quot;, message: &quot;Heart rate doesn't seem good, Sending Text messages to emergency contacts, Reset HeartRate in healthKit&quot;, preferredStyle: UIAlertControllerStyle.Alert)</th>
<th>This will create the alert that will be shown in case of Emergency.</th>
</tr>
</thead>
</table>
| alert.addAction(UIAlertAction(title: "OK", style: UIAlertActionStyle.Default, handler: {action in
        // Phone Notification
        self.phoneNotificationHandler()
        self.makeACall()
    }))) | This will add action button to the notification so when the user clicks Ok the phone should make the call and text |

On MyHealthWatch Extension folder there is a file interfacecontroller who has the function that make the Watch measures the Heart rate:
```swift
func createHeartRateStreamingQuery(wkoutStartDate: NSDate) -> HKQuery? {
    let predicate = HKQuery.predicateForSamplesWithStartDate(workoutStartDate, endDate: nil, options: HKQueryOptions.None)
    guard let qKType = HKObjectType.quantityTypeForIdentifier(HKQuantityTypeIdentifierHeartRate) else { return nil }

    let heartRateQ = HKAnchoredObjectQuery(type: qKType, predicate: predicate, anchor: anchor, limit: Int(HKObjectQueryNoLimit)) {
        (query, sampleObjects, deletedObjects, newAnchor, error) -> Void in
            guard let Anchornew = Anchornew else { return }
            self.anchor = Anchornew
            self.updateHeartRate(sampleObjects)
    }

    heartRateQ.updateHandler = {
        (query, samples, deleteObjects, newAnchor, error) -> Void in
            self.anchor = Anchornew!
            self.updateHeartRate(samples)
    }

    return heartRateQ
}

let heartRateQ = HKAnchoredObjectQuery(type: qKType, predicate: predicate, anchor: anchor, limit: Int(HKObjectQueryNoLimit)) {
    (query, sampleObjects, deletedObjects, newAnchor, error) -> Void in
        guard let Anchornew = Anchornew else { return }
        self.anchor = Anchornew
        self.updateHeartRate(sampleObjects)
}

heartRateQ.updateHandler = {
    (query, samples, deleteObjects, newAnchor, error) -> Void in
        self.anchor = Anchornew!
        self.updateHeartRate(samples)
}
```

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>This will make the watch start measuring the Heart rate</td>
<td></td>
</tr>
<tr>
<td>This will save the value of the heart rate with the value the watch measures it.</td>
<td></td>
</tr>
</tbody>
</table>
This function will save the heart rate measurement from Healthkit.

```
guard HKHealthStore.isHealthDataAvailable() == true else {
    headerLbl.setText("not available")
    return
}

    guard let quantityType = HKQuantityType.quantityTypeForIdentifier(HKQuantityTypeIdentifierHeartRate) else {
        displayNotAllowed()
        return
    }

    let typeToShare = Set(arrayLiteral: quantityType)
    let dataTypes = Set(arrayLiteral: quantityType)
    healthStore.requestAuthorizationToShareTypes(typeToShare, readTypes: dataTypes) {(success, error) -> Void in
        if success == false {
            self.displayNotAllowed()
        }else{

            self.healthStore.startWorkoutSession(self.workoutSession)
        }
    }

    print("will activate ")
```

<table>
<thead>
<tr>
<th>Check if it is available cause it will work only on iOS8 and above.</th>
<th>Accepts the HealthData quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>HKQuantityType.quantityTypeForIdentifier(HKQuantityTypeIdentifierHeartRate) else {</td>
<td></td>
</tr>
<tr>
<td>displayNotAllowed() return</td>
<td>Type of data to share from healthapp</td>
</tr>
<tr>
<td>let typeToShare = Set(arrayLiteral: quantityType)</td>
<td></td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>Set(arrayLiteral: quantityType)</code></td>
<td></td>
</tr>
<tr>
<td><code>healthStore.requestAuthorizationToShareTypes(typeToShare, readTypes: dataTypes) { (success, error) -&gt; Void in</code></td>
<td>Requesting to authorize the HealthStore access</td>
</tr>
<tr>
<td><code>if success == false</code></td>
<td></td>
</tr>
<tr>
<td><code>self.healthStore.startWorkoutSession(self.workoutSession)</code></td>
<td>Triggers as soon as the UI Opens</td>
</tr>
</tbody>
</table>

```swift
func Heartanimated() {
    self.animateWithDuration(0.5) {
        self.heart.setWidth(65)
        self.heart.setHeight(95)
    }

    let when = dispatch_time(DISPATCH_TIME_NOW, Int64(0.5 * double_t(NSEC_PER_SEC)))
    let queue = dispatch_get_global_queue(DISPATCH_QUEUE_PRIORITY_DEFAULT, 0)
    dispatch_after(when, queue) {
        dispatch_async(dispatch_get_main_queue(), {
            self.animateWithDuration(0.5, animations: {
                self.heart.setWidth(55)
                self.heart.setHeight(85)
            })
        })
    }
}
```

The above function will create the interface of the heart rate measurement on the watch.

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>self.animateWithDuration(0.5)</code></td>
<td>This will make the icon moves like the heart beat each .5 seconds.</td>
</tr>
<tr>
<td><code>self.heart.setWidth(55)</code></td>
<td>The width of the image that has been saved to Xcode Assets file as .png file.</td>
</tr>
<tr>
<td><code>self.heart.setHeight(85)</code></td>
<td>The height of the image.</td>
</tr>
</tbody>
</table>
self.animateWithDuration(0.5, animations: {
    self.heart.setWidth(55)
    self.heart.setHeight(85)
})

This will trigger the movement of the icon on the watch.

The heart rate data will be saved from the watch to the iPhone so we can send these data by the end of the day to the user physician, the function that handles that is on the MyHealth iphone files on the ConfigurationTableController.

On the function ViewDidLoad function

```swift
if WCSession.isSupported() {
    session = WCSession.defaultSession()
    session?.delegate = self
    session?.activateSession()
}
```

then we create the function session on the same swift file

```swift
func session(session: WCSession, didReceiveMessage message: [String : AnyObject], replyHandler: ([String : AnyObject]) -> Void) {
    let value = message["Value"] as? String
    dispatch_async(dispatch_get_main_queue()) {
        self.saveHeartValues.addObject(message.description)
    }

    print("array %@", self.saveHeartValues.description)

    var replyValues = Dictionary<String, AnyObject>()
    replyValues["Send"] = "Received from iphone"
    replyHandler(replyValues)
}
```
if WCSession.isSupported() {
    // This class will handle the communication between the Watchkit extension and its iPhone iOS so on the iPhone and the watch we should check if it is working or not.

    func session(session: WCSession, didReceiveMessage message: [String : AnyObject], replyHandler: ([String : AnyObject]) -> Void) {
        // This function will handle the communication of the data and saving it to email it.
        let value = message["Value"] as? String
        // This will handle received message from the watch then save it to value.
        dispatch_async(dispatch_get_main_queue()) {
            // use this to present immediately on the screen
            print("array %@", self.saveHeartValues.description)
            // This will handle saving the values on array to send it by email.
            var replyValues = Dictionary<String, AnyObject>()
            replyValues["Send"] = "Received from iphone"
            // This will handle the data in the email by saving these data in a dictionary
            replyHandler(replyValues)
            // Using the block to send back a message to the Watch
        }
    }
}

override func willActivate() {
    // This method is called when watch view controller is about to be visible to user
    super.willActivate()
    if (WCSession.isSupported()) {
        session = WCSession.defaultSession()
        session.delegate = self
        session.activateSession()
    }

    print("valu ", value, " Time ", NSDate())
}

On the Watch interfaceController file we have also activate WCSession to handle the communication between it and the phone.
let message = ["heartRateValue": value, "heartRateTime": NSDate()]

self.session.sendMessage(message, replyHandler: { ( response ) in
    }, errorHandler: { ( error ) in
        NSLog ( "Error sending message:% @ ", error
    )
}
self.outputLbl.setText(String(UInt16(value)))
self.heartRateSample += heartRateSamples
let name = sample.sourceRevision.source.name
self.updateDeviceName(name)
self.animateHeart()
}
}

if (WCSession.isSupported()) {
    session = WCSession.defaultSession()
    self.session.delegate = self
    session.activateSession()
}

print("valu ",value," Time ",NSDate())

self.session.sendMessage(message, replyHandler: { ( response ) in
    }, errorHandler: { ( error ) in
        NSLog ( "Error sending message:% @ ", error )
    }

let name =
After the notification shows to the user to remind him of moving and doing some activity the phone should track his movement. The movements that will be tracked is walking or running. If the user did not move the App will call his emergency contact. Inside the IPhone folder there is a FirstViewController file whose code is responsible for the movement tracking is:

### 4.2 Core Motion Framework

- To use the core motion framework inside your file you have to import the framework by writing: import CoreMotion
- It let’s the application receive motion data from device hardware and process that data.
- Classes under this framework:
  - CMMotionManager: A CMMotionManager object is the gateway to the motion services provided by iOS.
  - CMPedometer: Use a CMPedometer object to fetch pedestrian-related data.
  - CMStepCounter: The CMStepCounter class provides access to the number of steps the user has taken with the device.[Appledeveloper]

```swift
let pedoMeter = CMPedometer()
let motionManager: CMMotionManager = CMMotionManager()

func callRunningNotification(){
    if(CMMotionActivityManager.isActivityAvailable()){ // if it is available
        self.activityManager.startActivityUpdatesToQueue(NSOperationQueue
```
.currentQueue()!, withHandler: { (data: CMMotionActivity?) -> Void in
    if ((data?.walking) == true || (data?.running) == true) {
        print("walking");
        self.timer.invalidate()
    } else{
        print("not walking");
        dispatch_async(dispatch_get_main_queue(), {
            () -> Void in
                let errorAlert = UIAlertView(title: "You should be Running or walking.", message: "", delegate: self, cancelButtonTitle: "OK")
                errorAlert.show()
                self.timer.invalidate()
                self.timer = NSTimer.scheduledTimerWithTimeInterval(1.0, target: self, selector: "timerEmergencyCall", userInfo: nil, repeats: false)
        })
    }
}

deviceManager, error in
    if CMPedometer.isStepCountingAvailable(){
        self.pedoMeter.startPedometerUpdatesFromDate(NSDate()),
        withHandler: { 
            data, error in 
                guard let data = data else{
                    return
                }
                dispatch_async(dispatch_get_main_queue(),
                { () -> Void in
                if(error == nil){
                    self.timer.invalidate()
                    print("Number of steps = \(data.numberOfSteps)")
                }
                let steps : NSNumber = data.numberOfSteps
                let strConvert = String(steps)
                let errorAlert = UIAlertView(title: "You are Running or walking now", message:strConvert, delegate: self, cancelButtonTitle: "OK")
                errorAlert.show()
    }}}}
let secondController = SecondViewController()

secondController.activateExcercise()

} else{

}

}

}

}

}

}

if (CMMotionActivityManager.isActivityAvailable()) {

This will check if CMMotionActivityManager is available on the system or not because before iOS 7 you can not use it.

self.activityManager.startActivityUpdatesToQueue(NSOperationQueue.currentQueue())!

withHandler: {

(data:CMMotionActivity?) -> Void in

if ((data?.walking) == true || (data?.running) == true) {

print("walking");

} else{

print("not walking");

dispatch_async(dispatch_get_main_queue(),

motionManager.deviceMotionUpdateInterval = 1

This will check if there are update every 1 second. To start counting if the user did not move.
This should count the number of steps and save it. To show it to the user

If the user is moving the phone should show him that it is tracking his movement by showing him this message.

If the user did not move this method will be called to wait for a while, then call his emergency contact.

```swift
let errorAlert = UIAlertView(title: "You are Running or walking now", message: strConvert, delegate: self, cancelButtonTitle: "OK")
```

```swift
func timerEmergencyCall() {
    ++counter
    if counter == 10{
        counter = 0
        timer.invalidate()
        makeAnEmergencyCall()
    }
}
```

```swift
func makeAnEmergencyCall(){
    if let mainphone = defaults.valueForKey("emergencyTF1") {
        let url: NSURL = NSURL(string: "tel://"+(mainphone as! String))!
        UIApplication.sharedApplication().openURL(url)
    }
}
```

After the user enters his/her medicine names and the end and start dates and times and clicks “Save”. The saveMedAlert function will be called:

```swift
@IBAction func saveMedAlert(sender: AnyObject) {
```
let todoItem = TodoMedicineItem(deadline: startDate.date,
title: titleField.text!, UUID: NSUUID().UUIDString)
TodoMedicineList.sharedInstance.addItem(todoItem) // schedule a local notification to persist this item
}

when we save the data we declare the new value by giving it the information of
TodoMedicineItem file that handles the Add medicine form which is:

```swift
struct TodoMedicineItem {
    var title: String
    var deadline: NSDate
    var UUID: String

    init(deadline: NSDate, title: String, UUID: String) {
        self.deadline = deadline
        self.title = title
        self.UUID = UUID
    }

    var isOverdue: Bool {
        return (NSDate().compare(self.deadline) ==
NSComparisonResult.OrderedDescending) // deadline is earlier than current date
    }
}

var isOverdue: Bool {
    return (NSDate().compare(self.deadline) ==
NSComparisonResult.OrderedDescending)
}
```

This function will check each time if the medicine is overdue or not. By comparing the time and the
date that the user enters with current time and date.

addItem function will save our Medicine items to medicine list dictionary.

```swift
func addItem(item: TodoMedicineItem) {
    var MedDictionary =
NSUserDefaults.standardUserDefaults().dictionaryForKey(MEDS_KEY) ?? Dictionary()
    MedDictionary[item.UUID] = [
        "deadline": item.deadline,
```
"title": item.title, "UUID": item.UUID

NSUserDefaults.standardUserDefaults().setObject(todoDictionary, forKey: MEDS_KEY)

    let notify = UILocalNotification()
    notify.alertBody = "Todo Item "\(item.title)" Is Overdue"
    notify.alertAction = "open"
    notify.fireDate = item.deadline
    notify.soundName = UILocalNotificationDefaultSoundName
    notify.userInfo = [
        "title": item.title, "UUID": item.UUID
    ]
    notify.category = "TODO_CATEGORY"

UIApplication.sharedApplication().scheduleLocalNotification(notify)

    self.setBadgeNumbers()

MedDictionary[item.UUID] = [
    "deadline": item.deadline, "title": item.title, "UUID": item.UUID
]  This will store NSData representation of todo item in dictionary with UUID as key

NSUserDefaults.standardUserDefaults().setObject(MedDictionary, forKey: MEDS_KEY)  This will save/overwrite todo item list

let notify = UILocalNotification()
notify.alertBody = "Todo Item "\(item.title)" Is Overdue"

notify.alertAction = "open"  This is the text that is displayed after "slide to..." on the lock screen - defaults to "slide to view"

self.setBadgeNumbers()  This will update the badge on adding new item
So, when Medicine reminder notification shows the user has two choices: one is to press Complete if he took the medicine and the other option is to remind him in 30 minutes and this is the function that do the reminder:

```swift
func scheduleReminderforMedicine(item: TodoMedicineItem) {
    let notify = UILocalNotification()
    notify.alertBody = "Medicine Intake\"\(item.title)\" Is Overdue"
    notify.alertAction = "open"
    notify.fireDate = NSDate().dateByAddingTimeInterval(30 * 60)
    notify.soundName = UILocalNotificationDefaultSoundName // play default sound
    notify.userInfo = ["title": item.title, "UUID": item.UUID]
    notify.category = "MEDS_CATEGORY"

    UIApplication.sharedApplication().scheduleLocalNotification(notify)
}
```

When the user choose remind me in 30 minutes this is will calculate 30 minutes from current time and alert him again.

This will assign a unique identifier to the notification that we can use to retrieve it later.

```swift
@IBAction func startDatePickerValue(sender: UIDatePicker) {
    datePickerChanged()
}
@IBAction func endDatePickerValue(sender: AnyObject) {
    endDateChanged()
}
```
These function will be called when the user picks the Medicine start and End date and time with the drop down list on the interface. It will the startDate and EndDate to what the user choses.

![Figure 4.3 Add Medicine screen](image)

On Add Medicine interface if the user will continue using the same medicine for a while the Code can let him choose from a drop down list.

And below is the code that handles that.

```swift
enum RepeatInterval : String, CustomStringConvertible {
    case Never = "Never"
    case Every_Day = "Every Day"
    case Every_Week = "Every Week"
    case Every_2_Weeks = "Every 2 Weeks"
    case Every_Month = "Every Month"
    case Every_Year = "Every Year"

    var information : String { return rawValue }

    static let Values = [Never, Every_Day, Every_Week, Every_2_Weeks, Every_Month, Every_Year]
}

override func viewDidLoad() {
    super.viewDidLoad()
    intervals = ["Never","Every Day","Every Week","Every 2 Weeks","Every Month","Every Year"]
}```
override func tableView(tableView: UITableView, cellForRowAtIndexPath indexPath: NSIndexPath) -> UITableViewCell {
    let cell = tableView.dequeueReusableCellWithIdentifier("repeatIdentifier", forIndexPath: indexPath)

    // Configure the cell...
    cell.textLabel.text = intervals[indexPath.row]

    return cell
}

Figure 4.4 Add Emergency Contact

As the above figure shows the user should enter his Emergency contacts and this is the code that handles that:

@IBOutlet weak var nameTF: UITextField!
@IBOutlet weak var addNameBtn: UIButton!
@IBOutlet weak var phoneTF: UITextField!
@IBOutlet weak var emergencyTF1: UITextField!
@IBOutlet weak var emergencyTF2: UITextField!
@IBOutlet weak var emergencyTF3: UITextField!
@IBOutlet weak var emailTF1: UITextField!
When the user clicks the save button the function will be called:

```swift
// Save the object values
@IBAction func saveConfig(sender: AnyObject) {
    // sets the key for each field in the configuration
    saveDefaults()
}
```

Inside this function `saveDefaults()` function is called.

```swift
func saveDefaults()
{
    print("phoneTF.text", phoneTF.text)
    defaultsGrp.setValue(nameTF.text, forKey: "name")
    defaultsGrp.setValue(phoneTF.text, forKey: "mainphone")
    defaultsGrp.setValue(emergencyTF1.text, forKey: "emergency1")
    defaultsGrp.setValue(emergencyTF2.text, forKey: "emergency2")
    defaultsGrp.setValue(emergencyTF3.text, forKey: "emergency3")
    defaultsGrp.synchronize()
    updateBasicDefaults()
    print("The value no>>>>>>w is=", self.defaults.objectForKey("phoneTF"))
}
```

```swift
func loadBasicValues() {
    nameTF?.text = defaults.valueForKey("nameTF") as? String
    phoneTF?.text = defaults.valueForKey("phoneTF") as?
    String
    emergencyTF1?.text = defaults.valueForKey("emergencyTF1")
    as? String
    emergencyTF2?.text = defaults.valueForKey("emergencyTF2")
    as? String
    emergencyTF3?.text = defaults.valueForKey("emergencyTF3")
    as? String
}
```

```swift
func contactPicker(picker: CNContactPickerViewController, 
didSelectContactProperty contactProperty: CNContactProperty) {
```
let contact = contactProperty.contact
//set the Phone Number in first column
if let phoneNumber = contactProperty.value as? CNPhoneNumber
{
    if addNameStatus
    {
        print(phoneNumber.stringValue)
        phoneTF?.text = phoneNumber.stringValue
        print(contact.givenName)
        nameTF?.text = contact.givenName
        addNameStatus = false
    }
else if emergencyStatus1{
        emergencyTF1?.text = phoneNumber.stringValue
        emergencyStatus1 = false
    }
else if emergencyStatus2{
        emergencyTF2?.text = phoneNumber.stringValue
        emergencyStatus2 = false
    }
else if emergencyStatus3{
        emergencyTF3?.text = phoneNumber.stringValue
        emergencyStatus3 = false
    }
}
}

This function will save the Emergency contact information that the user enters. When the heart rate above or under the certain level the App will call and text the emergency contact.

Because there were a lot of testing to the App and to save time on editing the code, I create a Constant swift file to change the values to test in all cases.

```swift
import Foundation
struct Constants {
    static let MINIMUM_HEART_RATE: Double = 55.0
    static let MAXIMUM_HEART_RATE: Double = 60.0
    static let HEART_COUNT_FREQUENCY_in_sec: Double = 300.0
    static let EXCERCISE_ALERT_FREQUENCY_in_sec: Double = 300.0
}
```
when I want to test the heart rate in case of emergency I change the Maximum and minimum values. Also, for the Notification time.

4.3 Watch Connectivity Framework

- To use the watch connectivity framework inside your file you have to import the framework by writing: import WatchConnectivity
- Import two way communications between iOS app and a watchkit extension on a paired apple watch.
- Apps use this framework to pass files and data back and forth.
- Classes
  - WCSession: The WCSession class facilitates communication between a WatchKit extension and its companion iOS app.
- Protocols
  - WCSessionDelegate: defines methods for responding to messages sent by a WCSession object.[Appledeveloper]

On the watch FirstViewController File this method to send the heart rate measurement to the IPhone.

```swift
if (WCSession.isSupported()) {
    session = WCSession.defaultSession()
    session.delegate = self
    session.activateSession()

    let message = ["heartRateValue" : value, "heartRateTime" : NSDate()]

    self.session.sendMessage (message,
        replyHandler : { ( response ) in
            }, errorHandler : { ( error ) in
            NSLog ( "Error sending message:% @ " , error
            )
        }
    )
}
self.session.sendMessage({"heartRate": value}, replyHandler: { (response) in
    }, errorHandler: { (error) in
        NSLog("Error sending message:% @ ", error
    )
    })
Chapter 5

Performance Evaluation of MyHealth App

After testing MyHealth application many times XCode enables you to measure your application performance from Memory, CPU, Network and Disk usage.

This is the performance of running the application on the actual devices iPhone 6 and Apple Watch.

Figure 5.1 MyHealth Performance on Memory
so, the application is not heavy on the computer its only take 1% of CPU uses some times to 4% maximum. It is only needs 1.5 MB of memory used.

Even though when we run the App on the actual devices iPhone and Apple Watch it took about 5 minutes to activate on the watch after you open the App main page on the iPhone I think maybe the watch is taking time to process the heart rate measurements not because the App size.
Chapter 6

Lessons Learned and Challenges

Since I started working on my project, quite a few times I felt I couldn’t take it anymore and at the same time, I felt euphoria every time I achieved a challenging task of my application. I was taking a risk of building my application based on Apple’s new software and products. Merely five months after Apple announced its Healthkit, I started working on my application. At the beginning I was working on Xcode 7.2 platform but two months later Apple release Xcode 7.3. I read a lot about it and the decision was not easy should I upgrade and risk that my application not working in new version, or should I stay on the old version even though it is slow and hung a lot. I took my chance and upgrade my Xcode to 7.3, which really made a difference on the time processing the code. Another challenges are the frequent updates of Swift 2.0, which is a new open source programming language that changes everyday and requires modification of my code frequently. A week before the code may have no error, all of a sudden Xcode does not recognize the simple definition of my code.

My application idea was to measure the heart rate every 10 minutes without the user knowing that the watch is measuring the heart rate but that was impossible. I searched a lot and contact Apple developer society for solution. I found that before the watch measurement begins, it should at least send a notification to the user because in real life if
we implement application who do that, it will exhaust the Apple watch battery in minutes. So it is not logical to do so. There supposes to be a notification before the heart rate measurement. As my professor Chow suggested after my proposal that I should use the research kit as an aid to my application. Through research, I found that the research kit is written in Objective-C, so I can’t add it to my application. I tried to record the heart rate data to the user and when the application sends it by email weekly you can do more than research on these data. You can monitor the heart rate health of the user if the physician found something unusual on the data, maybe the physician can help prevent bad things from happening.

Many times the application did not work as it suppose to be and that was frustrating. In these cases, I learned that each problem has a solution if I tried to divide the task to small tasks and try to work on each task individually. That makes me focus on each task and concentrates better.

In summery the lessons I learned working on my application include:

• Implementing mobile health application is not easy. You have to consider all scenarios that may happen and in real life you do not want any mistakes because it has significant impact on human lives. I can see why there is a limitation on mobile health industry specially on critical areas.

• After implementing the application, we need to test on actual devices since the result may changes. The things, that have been worked on simulator, may not work as it is on the real devices.

• My computer had some problems and I took it to be formatted so I was trying to code on my friend’s labtop. I discover that using Xcode on different computers will
revoke your developer certificate and you need to renew it and that took me long
time to realize problem and find the solution. I had to renew my Apple developer
account for one more year.

• HealthKit gave you a lot of choices to work with the health data but sometimes
there are limits you can not exceed and if your goal is to do something beyond these
limits, you have to research more and be smarter than what the Healthkit can offer. My application idea is to make the heart rate measurement without the user know. As a background function but in reality Healthkit have to be active all the time manually. As on the Apple Fitness app on the Watch you have to choose that you are now doing your activity. All the Watch application generally runs in the foreground while the screen is On on the user wrist. So, to achieve my application goal I had to notify the user first each time the App will measure the user heart rate, and when he presses the button when the notification tells him that the App now will make the measurement the Watch will wake up and do the heart rate measurements.
Chapter 7

Future Directions

iOS development is an extensive process that involves mastery of the tools for the development including the WatchKit framework. It is coupled with WatchKit extensions and WatchKit application. The developer then needs to build an understanding of the interface, positioning and formatting. Also, one needs to master the creation of the actions and outlets as well as the lifecycle for the interface. Further, one needs to know App Groups, the storyboard, interface objects, and how to navigate and finally to perform a test on the application. A lot of tests need to be performed on the WatchKit but it is something really important to know.

After implementing MyHealth application I hope that it will contribute to the base for future enhancements and improvements in this area. I had many ideas for my application that I wish in the future if I had the time to implement it, or maybe in my PhD studies I can develop that. One idea is related to the memory problem commonly faced by seniors. If we can implement a bar code reader to read the bar code on the medicine, then we can make sure that the patient is taking the right medicines. Second idea, if we can develop and implement sensor machines that can track the hand and body movements, then in cases that the user need physical therapy for some exercises, we can track if the user are doing these
exercises on the right way. Dr. Lewis gave me a really interesting idea he is doing lot of researches on medical applications. He told me that if we can develop and implement sensor machine to record the ECG, the electrocardiogram of the heart, which can used to assess the electrical and muscular functions of the heart. If we record it in a good and correct way, we can find out the heart attack before it will happen. Managing your health life away from hospitals is the goal of mobile health application. If we can think of the diseases that you have to deal with and taking care of yourself while you are doing your daily routine, we can find an idea for each need.

Mobile health development is considering a work in progress. It depends on the patients needs and demands. That does not mean all the applications will be successful. To be the successful, an application’s performance needs to be dependable and meet the market demand. That’s what the developer are thinking about before develop in health application. Nowadays fitness application proves its quality and its impact on people lives [Miller2014]. The future direction on mobile health application is the application that gathers the information from the user, not the ones that give the user the information. That what Apple did in the ResearchKit that helps develop application that gather information and health data from the user to help clinical research[Miller2014]. So, if we consider this to improve MyHealth application to record the heart rate measurement data and if we improve it to collect other health information using other development tools that will take the application to the next level. I hope to be able to contribute to that in the near future.
Chapter 8

Conclusion

Today people are depending on the smart phones in so many ways: to organize their daily tasks, to navigate to a specific location, and to remember what they are suppose to do in a certain time.

There are a lot of mobile health applications on iPhone and Android that are aspiring to achieve their goals to improve the health and fitness of the user. The requirements for mobile health applications will change with the change of people health issues. Health insurance expense and care limitations on hospitals are main reasons for the continues demands on mobile health application that helps users to adapt to his health issues away from hospitals.

These health applications help increase the awareness of the patient’s parent and relatives of the illness they have. In Autism cases there are a lot of applications that help the parents to be more aware of the needs of their children. There are a lot of exercises that they can do to help their children to adapt with their environment and improve their situations with the help of these apps that give steps and detailed information.

My iOS development using Swift 2.2 is an interesting experience. It gives a lot of options that you can implement, but on the other hand, Apple made some limitations to the developers that you can not cross. You have to do search, dig more to make the
development tools serve your Application goal. Healthkit gives you the access to the user health data and enables you to implement applications that serve health area. Appgroup made it easier to share data between the iPhone and Apple watch. You can save data to Appgroup and retrieve these data using the same group name.

Apple watch serves the mobile health application with it’s built-in features, including sensors to measure the heart rate and sensors to track the movement. While you are sitting on your chair watching your favorite TV show, you can get the result of measuring heart rate as the same as on hospital with only one click on your Apple watch. No doubt that made our life easier and more functional. Using these features of Apple watch to develop mobile health applications will achieve a quantum jump in the field of mobile application development. Instead of developing machines that has sensors to evaluate and measure your health vitals, you can use the Apple watch to save time in implementing and developing with the same results which you want at the end.

Focusing on elderly people is important as their population is increasing. Some of them live alone with no relative to watch them all the time, and some live away from their relatives. To implement mobile applications with ease of use interface will help manage and control senior’s health needs without the need of constant supervision or staying at nursing home. Seniors needs health applications more because when you get older your health life needs more care and attention. That is the reason I chose seniors as a target of
my thesis study. Because I hope some one will use my study and improve my application to help our parents, grandparents and our self when the time comes that we need to focus more on our health issues and needs.

In conclusion, what is the answer to my thesis proposal question: “Can I design an accurate and effective application using HealthKit with the Apple Watch?” After five months of studying and implementing my application on iPhone and Apple watch I can answer “Yes I can design Health Application on both devices using HealthKit but to call my application effective I need years of testing on real people life.” As a developer I can say my application has a potential to do what it is supposed to do. In the future I can see promising ideas realized and mobile health app with significant improvements.

I hope at the end of my study that in the future we can improve mobile health application to help our old people to stay where they want to be while we can make sure that they are fine. They do not need us all the time to be with them. And if they need care we can discover it before it will happen
References


[Yoo2013] Yoo, Jin-Hong. "The meaning of information technology (IT) mobile devices to me, the infectious disease physician." Infection & chemotherapy 45.2 (2013): 244-251.


Appendix A. Installation and Demo

Installation and Configuration of MyHealth

Because my application is for test and study only I did not publish it to Apple store yet. To install MyHealth application you need

• Iphone 6 with 9.2 iOS or more
• Apple watch
• MyHealth xcode with developer account.

When you plug the iPhone paired with Apple watch to the computer open the xcode application and run the App

On the iPhone and the watch, you will get the Application

To download the source code of MyHealth application please use the following link:

http://walrus.uccs.edu/~gsc/pub/master/aalfaidi/src/MyHealth.zip
Figure 10.1 MyHealth main page

This is the iPhone main page when you first run the application it will be shown.

Figure 10.2 Health data access approval
The application will ask you to send Notification that will be shown on the Watch and iPhone.

![Figure 10.3 MyHealth watch ask to use Health data](image)

When the application runs on The watch it will ask you to use Your Health Data The health data that is stored on the Apple Health database.
This will be shown on the iPhone you have to turn on the Heart Rate button and click Allow on the top right.
The Watch will start to measure your heart rate.
In case of emergency when the heart rate above or under the Normal level the application will send a Notification to the user to notify him that his heart doesn’t seem good. When the user press Ok it will start calling and texting.

![Figure 10.7 Calling emergency contact](image)

This is what happens in case of emergency the App will call the Emergency contact.
This is what happens in case of emergency the App will text the Emergency contact.
When the Medicine is overdue the Notification will be shown on both iPhone and watch to remind the user to take his medicine.

![Image of a smartphone screen with a notification]

**Figure 10.10 Handling the notification**

The User has two options either to remind him in 30 minutes or if he took his medicine he can choose complete.
The notification will show on the watch also so, twice daily there will be notification to remind the user to exercise if he chooses complete the phone will track his movement and tell if he walks or run. Also it will calculate the steps number and if he did not move for a while it will call his emergency contact.
Figure 10.12 tracking the movement notification

To make sure that the application is tracking the movement of the user the application when the user start walking or running will send a notification to him that he is moving and the number of steps.

If the user did not move the App will call his emergency contact.
This is the Email that the Physician will receive that shows the heart rate values and the time the measurement done.
Implementation Related Resources:

Movement tracker:
http://stackoverflow.com/questions/24114164/motion-manager-is-not-working-in-swift

NCalendar

Notification

Repeat Notification

Saving Integers

Workout Sessions
https://github.com/schickling/hackrisk-calm/blob/master/Calm/HealthKitHelper.swift

Watch Connectivity.