UnityPoint Health Mobile Application

DEC1610
UnityPoint Health
S. Tirthrapura
Ben Kixmiller - Key Concept Holder
Haythem Ebrahem - Key Concept Holder
Brian Nguyen - Team Communications Leader
George Ndemi - Key Concept Holder
Jonathan Campbell - Team Webmaster
Aakash Sheth - Team Leader

dec1610@iastate.edu

Revised: April 5, 2016

Table of Contents

1 Introduction

1.1 Project statement

Explain what the project is about. What are you trying to do?

We are working on sending data from wearable devices to the patient's chart where they can be used by medical personnel towards improving patient's care and wellbeing. The data on the patient's chart should be easy to comprehend to both the patient's and the health care workers on a user friendly UI.

1.2 purpose

Explain what is driving this project. Why is this work of benefit to the society?

Since the passing of the Healthcare bill, hospitals are paid by insurance according to the success of treatment. Any readmissions and patients visiting a provider for the same problem reduces the amount of reimbursement. Health care workers would like to monitor the patients, if they are following Doctor's advice on the number of steps they take everyday, sleep pattern, glucose levels and blood pressure. This information will enable the healthcare workers to know the best treatment with the best outcome for a specific case, thereby decreasing the readmissions and increasing revenue.

1.3 Goals

Explain what you hope to accomplish through this particular senior design project. What would you like to achieve? Enlist as many goals as you can envision.

-We would like the data that is collected by the wearable devices be sent to the server then to the patient's chart where clients can access it.

-The current UI is not user friendly, we would like to upgrade it to a better looking one and easy to scroll through.

-Currently, Facebook is the only authentication used, we would like to to enable clients to log in using Twitter and Google accounts as most clients do not want to link their Facebook accounts with their Medical records.

2 Deliverables

For this project Unity Health Point required us to build on the existing apps to :

- Acquire the data from different wearables such as fitbit, garmin, Apple watch by the IOS apps.
- Setup a server, domain and database to save the data.

- push the data to the server.
- Implement different authentication methods such as Facebook, Google, Twitter and Unity Health Point authentication (Mychart).
- Retrieve the data from the server and display it in different format such as graphs, comparison charts, goals.
- Implement UI that meet the UPH standards.

3 Design

Include any possible methods and/or solutions for approaching the project at hand. You may want to include diagrams such as flowcharts to, block diagrams, or other types to visualize these concepts.

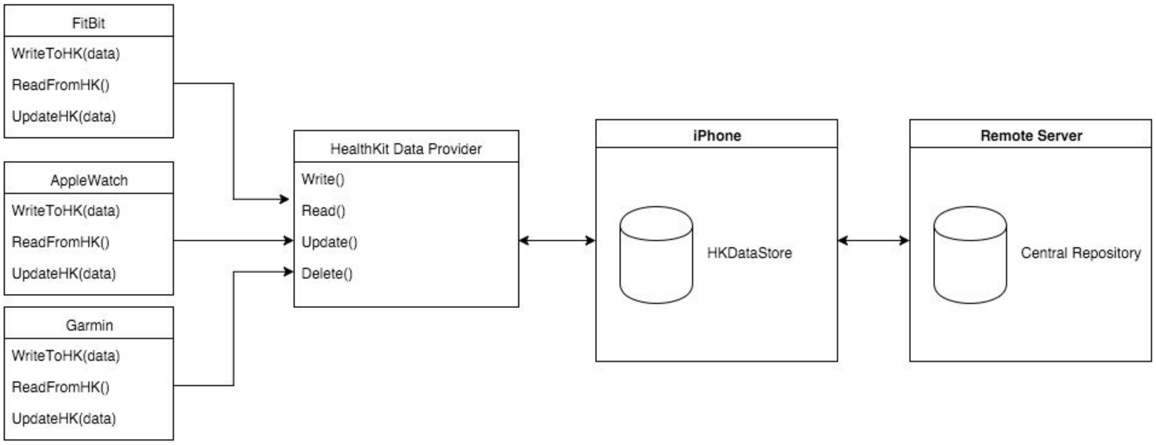

Figure 1.1 – High Level Overview

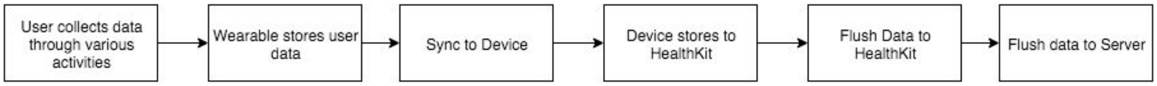
User collects data through various activities
Wearable stores user data
Sync to Device
Device stores to HealthKit
Flush Data to HealthKit
Flush data to Server

Figure 1.2 – Typical Workflow

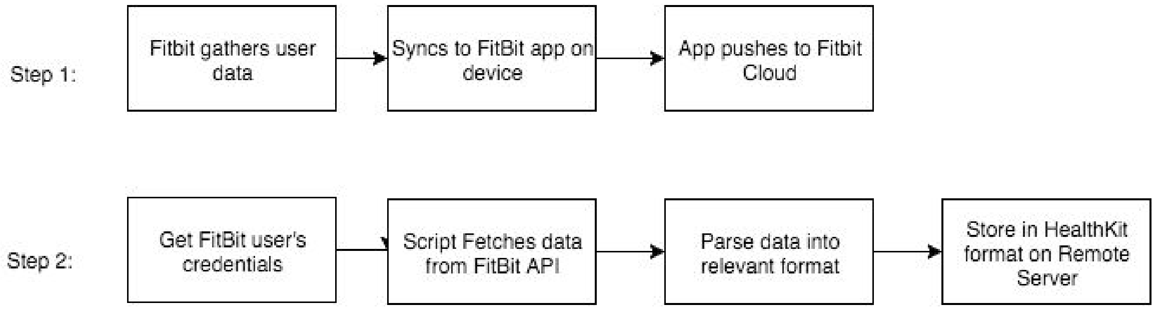
Step 1:
Fitbit gathers user data
Syncs to FitBit app on device
App pushes to Fitbit Cloud
Step 2:
Get FitBit user's credentials
Script Fetches data from FitBit API
Parse data into relevant format
Store in HealthKit format on Remote Server

Figure 1.3 - FitBit Special Case Workflow

3.1 Previous work/literature

Detail any similar products or research done on this topic previously. Please cite your sources and include them in your references. All figures must be captioned and referenced in your text.

Since we will be basing our development from last year's code, we have planned to refactor it so that it allows us maximum flexibility to gather the most data from as many devices as possible. Some devices that we plan to incorporate and pull data from do not provide ready made interfaces that handle interaction with Apple HealthKit, such as FitBit. Instead we will have to create our own custom interface that handles gathering FitBit data from their API, https://dev.fitbit.com/docs/, and then load it into our own HealthKit store. Further research suggests that other wearables will perform in a similiar respect, such as the Garmin devices, who's API can be found at http://developer.garmin.com/wellness-api/overview/.

The most research we have conducted however is from the Apple Developer program documentation. The docs we are most interested in are those for developing iOS applications as well as WatchOS since we plan to gather from Apple Watches, among other devices. In reality, this will be the 'easiest' wearable to implement since AppleWatch interfaces so well with other Apple devices already. However, there are several caveats that need to be addressed, such as permission to access health data, as well as meet Apple's user guidelines to ensure an app that follows tried and tested iPhone user experience standards.

3.2 Proposed System Block diagram

For most groups you can include a flowchart of how the system will work. In case your project is not about putting together some sort of a system, you may describe the process that you will follow to achieve your deliverables.

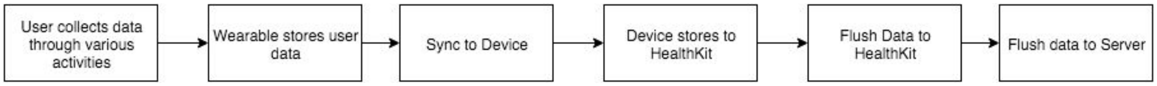
User collects data through various activities
Wearable stores user data
Sync to Device
Device stores to HealthKit
Flush Data to HealthKit
Flush data to Server

Figure 3.1 - High level overview

The project goal, at this point, is to create a mobile application that displays health data based off of various wearables. The data is aggregated on the user's wearable, mobile device, as well as a remote server to be analyzed based on different metrics.

Since many of the deliverables are additions to an existing system, we do not plan to change the existing process until absolutely necessary. The additions include several options that cannot be described in flow charts, and so we will describe them in words.

Firstly, we plan to implement authentication via various single sign on providers such as Google, Facebook and Twitter. Overall, this step should be fairly trivial since it has been implemented already among other popular mobile applications. We may face a challenge in development due to lack of experience in developing in Swift. With that in mind, several team members have already started testing Swift applications in order to get a better understanding of the language.

Developing a more robust interface is next on the agenda. Since we will be working with CyBiz, our goal is to iterate through various styles of user interfaces that align with UPH branding guidelines as well as provide a valuable user experience that makes users attracted to app. As whole, the team has very little experience in graphic design as well as human-mobile interaction. Some of us, however, have mobile development experience that will translate to a more robust user interface. In this case, several iterations as well as testing will the be the only way we gain an idea of the best design to make the app both functional, usable, and aesthetically pleasing.

Data is king, and will the meat of our application. Since we plan to use several wearable devices, we will also be tasked with developing several interfaces for each wearable to keep the project modular and scalable. Through understanding of the DataProvider design pattern, we will create several interfaces with smaller teams working simultaneously on different devices. Again, we will have to overcome the barrier of a new programming language.

Since data is so important, we need to establish a way to store it in a central repository. We plan to implement this feature using a remote server to store health data and be able to push to a website to than display it in a relevant and useful manner. Several team members have experience with handling back end communication, so we feel we are slightly ahead in this area. This step will involve remote server setup and administration, data reports, as well as developing a standard format in which data can be inputted for our analytics, as well as when we UPH decides to integrate them in their Electronic Health Records.

3.3 Assessment of Proposed methods

Provide a short discussion about the different approaches available and the approach you want to follow in your work.

Last years team was able to collect health data and store in the provided HealthKit platform, but they were lacking any useful design that made users want to use the app.

Along the same lines, they did not provide functionality to get data from several sources.

Out iteration of the project aims to improve the user experience by designing an app that is both aesthetically pleasing, as well as efficient. By also providing various authentication protocols, such as Google, Twitter, and Facebook, we open the app to a variety of users.

The other aspect of the app is to develop it in a scalable way. The idea is to create a modular app that breaks down tasks into smaller pieces to allow for code updates that do not break other parts of the app while still providing reliability. For instance, we intend to use the Data Provider design pattern that isolates the database driver used to talk to HealthKit from wearables other than the AppleWatch. On the same note, we intend to create interfaces for each of the variable devices so that we can add new devices without having to rewrite how that device specifically talks to the database since we have a single class that handles all database transactions. By creating multiple interfaces based on each device, we allow the addition of any new device by creating its respective class based on the original interface.

There are several design patterns we could follow such as Factories or State Design, but we found that using the DataProvider pattern best suits our needs. The use of this pattern allows the most flexibility to add and remove devices as well as change portions of code without affecting reliability of other components.

3.4 Validation

How will you confirm that your solutions work?

The main goal of this project is to collect relevant health data for various users using various health driven wearable devices. While collecting this data is ultimately the highest priority, displaying that data back to the user as well as analysts is also very important. The user experience of this app will drive the users to collect more data since it allows easier access to viewing the data in a meaningful way.

We will measure the success of our project by collecting data in a standard format that can be scaled with more users as well as devices and then being able to display it back to the user correctly. For example, displaying heart rate data incorrectly would be utterly useless because it offers no way of improving their health.

Having a standard data storage format will also allow analysts to decipher the data into useful metrics. If the data is analyzed and found to be useful in some metric, it will directly correlate to the success of this project.

4 Project Requirements/Specifications

4.1 FUNCTIONAL

List and explain the functional requirements of the project. This would include all the technical requirements you fulfil during your senior design project.

- iOS app allows user login with several different authentication option such as Google, Twitter Facebook, etc.
- iOS app automatically receives data which collected from multiple types of devices such as: fitbit, apple watch, garmin..
- iOS app analyzes the data and show useful report to user.
- iOS app send the data to server in appropriate format that can be compatible with existing data from Epic.

4.2 NON-FUNCTIONAL

List and explain the non-functional requirements of the project. This is where you would enlist non-technical requirements. This may still be a fundamental deliverable that your client needs at the end of the semester.

- security
    - the data is transferred to sever will be encrypted
    - no user can access any data of other user
    - users can erase their own confidential data which stored in the phone
- reliability
    - the app runs 24/7 without interrupt
    - the app is responsive, quick
- performance
    - the data is compressed with advanced algorithm to help saving storage
    - the app is energy efficient

5 Challenges

Include any concerns or details that may slow or hinder your plan as it is now. These may include anything to do with costs, materials, equipment, knowledge of area, accuracy issues, etc.

- The client doesn't have exact plans on what we should do. We are kind of confused on if our plans are what's best for our app.
- The language for the code is Swift, which nobody in our group has used previously before taking this course. We are learning it but this has led to a lack of progress on the project.

- Our team has to work with a CyBiz team, that gives us feedback into what the market is looking for. We have to wait for them to do their research before we can know what the customers are looking for which can slow our progress down.
- Fitbit is hard to pull data from due to them not being very compatible with Apple HealthKit. We are still researching on how to pull data from them.
- Only a couple of us have Macbooks. Using a Mac is required for us to develop ioS. Members of the group that don't have Mac's use a computer from the senior design lab to work on the project.

6 Timeline

You may want to include a Gantt chart/something similar to help visualize your timeline to complete the project.

6.1 First Semester

Breakdown your timeline into detail of what needs to be done by the end of the first semester. You may want to include division of work amongst the team.

2/20 - 2/29:

- Acquire the data from different wearables such as fitbit, garmin, Apple watch by the IOS apps.
- Get developer licences and apple cloud licences from Unity Point (early this week).

3/1 - 3/31:

First two weeks: (we will have the code by now)

- Start going over previous code, get a feel for what is going on.
- Continue acquiring data.
- Setup a server, domain and database to save the data.
- push the data to the server.

Second two weeks:

- Implement different authentication methods such as facebook, google and Unity Health Point authentication (Mychart).
- Wearable API

4/1 - 4/30:

Detail what needs to be done in the second semester. You may want to include division of work amongst the team.

9/1 - 9/31:

- Look over previous code, get back in touch with group/project lead/professors/Unity Point.
- Anything (minus UI) that hasn't been completed yet last semester must be done now (or the last part of August).

10/1 - 10/30

- Lot's of UI (get the app to use the Unity Point standard style)
- Extra deliverables or additional changes
- Start preparing for presentation

11/1 - 11/30

- Lot's of UI (get the app to use the Unity Point standard style)
- Last minute changes
- Finish presentation

7 Conclusions

Our plan is to take the work from the previous group that worked on this project and improve it to have new features in it. Our goals are aligned with the functional and nonfunctional requirements from above. We are working with a CyBiz team to find out the data that is wanted the most from the current market. The current UI is not very user-friendly so we'd like to make it look better. We also are adding google and possibly Twitter authentication so more people can access MyChart. We are currently implementing the different tabs for the project such as Notes, Devices, and the different metrics to measure health.