FitByte

Technological Feasibility

October 17, 2018

Jake Farrar
Jacob Lemon
Austin Pederson

Ana Paula Chaves Steinmacher (Mentor)
Dr. Kyle Winfree (Sponsor)
Dr. Gregory Dominick (Sponsor)

1) Introduction

In 2016, the number one cause of death was ischaemic heart disease (**Fig. 1**). In fact, ischaemic heart disease claimed over four times as many lives as road injuries (**Fig. 1**). The fact that heart disease is such a big killer should motivate people to try and prevent it. The easiest way to help prevent ischaemic heart disease is by completing at least 30 minutes of aerobic exercise five times per week. There are technologies that help motivate people to exercise. These technologies are usually in the form of a wearable. The most prominent example of this, is the Fitbit.

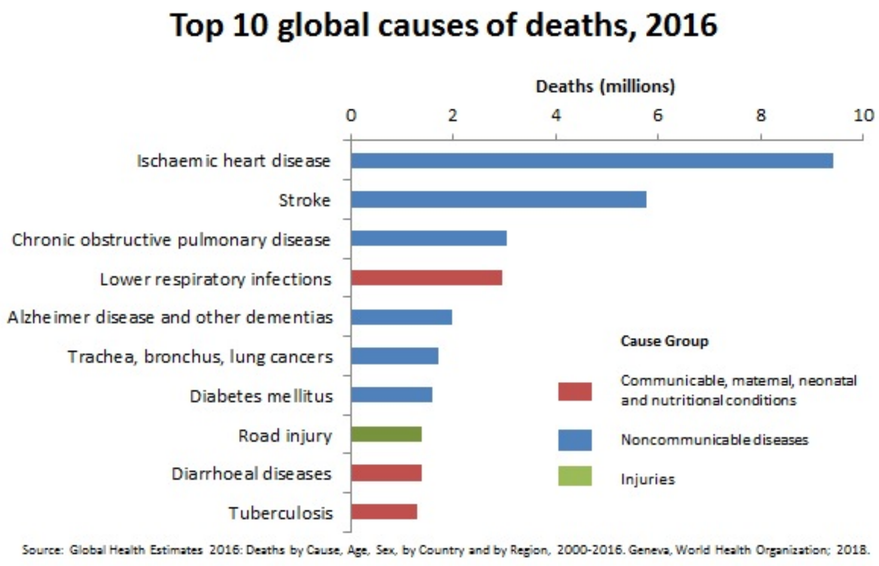

Fig. 1: Graph detailing top 10 causes of death across the globe in 2016.

Fitbit was established in 2007. Since then, Fitbit has enjoyed an enormous amount of success. In 2017 alone, Fitbit sold 15.3 million devices. The success of Fitbit is in part due to its social features. These social features are what help drive users to exercise. For example, users can challenge each other to competitions to see who can get more steps in a day.

This is very interesting to both our sponsors as they are both in the field of study having to do with human interaction with technology and its potential effects on fitness. Our sponsors are Dr. Kyle Winfree and Dr. Gregory Dominick. Dr. Winfree is a researcher at Northern Arizona University. His research centers on the use of devices engineered for therapies and assessment. Dr. Gregory Dominick is a researcher at University of Delaware. Dr. Dominick's research centers on health-related topics.

They have been conducting research by giving their users Fitbits to wear for a month. This allows them to track all of the data that the wearable technology offers. Currently, they have a software called WearWare that grabs the data, and dumps it to a CSV. They are running into issues sifting through the data, sharing it, analyzing it, and giving their users feedback in a reasonable amount of time.

One of the glaring issues that our sponsors are facing is sharing the data with each other. They currently have no easy solution for sharing data. Another large issue is that the data that is acquired from the Fitbit devices can be somewhat unreliable. This means more processing is necessary on the data to ensure correctness. The last big problem Dr. Winfree and Dr. Dominick have is that it is difficult to assess the data in real time in order to determine who needs to be contacted in order get them back on track.

Team Fitbyte consists of Jake Farrar as Team Lead, Jacob Lemon as QA and Enforcer, and Austin Pederson as Web Designer and Code Base Manager. We have been tasked with helping Dr. Kyle Winfree and Dr. Gregory Dominick with their research about the functionality and accuracy of Fitbit data as they are having some troubles sharing data, analyzing data in realtime, and keeping users motivated without having to manually check up on their participants. Their research has proven that Fitbits data has some flaws that need to fixed. They have asked us, Team Fitbyte, to solve their problem by creating a web API that allows the team in Delaware (led by Dr Dominick) to request data that is produced by WearWare. If our team has time, Dr. Winfree would like us implement our own way to dynamically analyze the WearWare data and send users motivational SMS text messages.

Dr. Kyle Winfree did an excellent job outlining the problem and his ideas for a potential solution. He wants us to create a web API that allows the team in Delaware to securely fetch data that has been produced by WearWare. This is the MVP for the project and should be the most important aspect in our research. As soon as we are able to complete the web API, we can move on to implementing a way to analyze the JSON file provided by WearWare. This would include the use of Octave, and if all else fails, Matlab. Octave would allow us to read in a JSON file, analyze the data, and spit out an output. Based on the output that was generated, we would send the user an

SMS update if they are making poor progress to a goal, or have been sitting for long period of time.

In this document, we will explore the various technologies that can be used in order to solve the problems that Dr. Kyle Winfree and Dr. Gregory Dominick are having. We will first address the technological challenges. We will then analyze the solutions that we have found. After the analysis, we will outline the technology integrations. We will then conclude the document and outline which technologies would work best for our sponsors in order to best meet their needs.

2) Technological Challenges

In order to address the problems outlined in the introduction, this section is split into five sub problems: creating a way for servers to make calls to our data server and receive a response, creating a program that will analyze the data collected from WearWare, creating a program that will take the analyzed data and send out SMS text messages when needed, creating a program that will have a UI that researchers can get data from easily, and understanding and interpreting the existing code for WearWare.

Problems we need to solve:

2.1. Create a way for servers to make calls to our data server and receive data. This is a problem because this currently does not exist. Dr. Dominick would like our team to create a way for his team to be able to request data from the servers here at NAU without having to get into contact with Dr. Winfree or anyone on Dr. Winfree's team.

2.2. Create a program that will analyze data. This is an issue because currently there is no program that analyzes the data from WearWare. The data must all be analyzed manually currently and that takes up a lot of time.

2.3. Create a program that will take the analyzed data and create a notification for the end user. This is an issue because it is possible that participants in a study could not be using their FitBit and this would create large inconsistencies in the data. The current solution is to have someone manually check up on any participants that would be participating in a study and then checking on the last time that the FitBit was synced and then manually getting into contact with the participant to try and get them to put the FitBit back on.

2.4. Create a program that will have a UI that researchers can pull their own data from. This is a problem because it is currently a very barebones project and the interface is not super pretty. With a more stylistic, attractive, and intuitive UI, it could become much easier for researchers to pull data from WearWare and for users to see their data that they have contributed to a study.

2.5. Understand a interpret the existing code for the project. This is a minor issue as we need to understand the output of the code so we know what inputs we will be obtaining.

3) Technology Analysis

3.1) Potential Solutions for Problem 2.1:

Creating a solution for servers to talk to each other in order to exchange data requires that we create a REST API. Three possible solutions to this problem that we have identified are Django, Express.js and Laravel. They all could be viable in solving the problem. Below is the analysis between them.

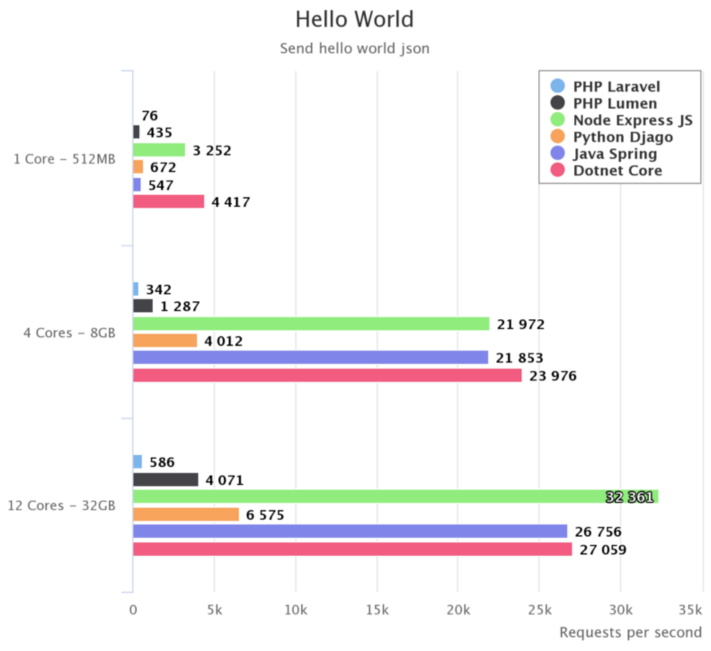

Fig. 2: Requests per second for various web frameworks when asking for a simple json.

a. **Django**

Django is a Python framework for backend web development. The pros and cons for this framework are shown below.

<u>**Pros**</u>	<u>**Cons**</u>
Python	Less frequent updates
Rapid development possible	Performance lacking (**Fig. 2**)
Easy database management	Low flexibility within the framework
Good documentation	Synchronous
Scalable	Not great for small projects
Secure	Large size

One of the most important features here is inherent security. As we are dealing with data related to people's' health, the data is potentially quite sensitive. Another huge advantage for Django is that WearWare is already written in Django, so if the project needs to be altered at a later date, that can easily be done in-house. The lack of performance isn't a huge deal here as requests per second will likely not reach astronomically high numbers.

b. **Express.js**

Express.js is a JavaScript framework for backend web development. It is used in tandem with Node.js. The pros and cons for this framework are shown on the next page.

<u>**Pros**</u>	<u>**Cons**</u>
JavaScript	Bound to single CPU
Excellent performance (**Fig. 2**)	Community examples can be poor

Good documentation	Extremely frequent releases
Asynchronous	Async programming is harder
Extremely extensible	

The main advantages of Express.js are its performance and documentation. Express was the best performer as showcased in (**Fig. 2**). This makes it a strong candidate if this project needs to be grown to a very large user base. The biggest cons against Express.js are that there is always a new version just around the corner that can potentially make your project nearly deprecated and that asynchronous programming can be quite difficult.

c. Laravel

Laravel is a PHP framework for backend web development. The pros and cons for this framework are shown below:

Pros	**Cons**
PHP support is easy to find	PHP is old
Good for creating an API	Very poor performance (**Fig. 2**)
Good documentation	Bloated
Very flexible	Steep learning curve
Easy database management	

The biggest advantage afforded by Laravel is the sheer amount of support for PHP. PHP has been around for so long that you can find almost anything online with one search for any problem you could ever run into. However, that is a double-edged sword as PHP was first designed so long ago and was never planned to become as large as it did. This means that PHP has almost no conventions and a lot of hacky solutions online. PHP is also the worst performer by far, making it very difficult to recommend if the project needs to scale at any point in time.

After analyzing our choices carefully, we have decided upon using Django. This decision was made based on a multitude of different reasons. Since we will not be developing an entire web application, just an API, Django provides us with many important advantages over the competition. Firstly, since this API will be dealing with potentially sensitive data, security is tantamount. Django provides excellent protection without too much effort put in on our part. This makes development timelines more rapid. Another huge advantage of using Django is that WearWare is already written using Django and if the API needs to be updated after this project is completed, it can be done quite easily. Given that this API does not need to serve thousands upon thousands of requests per second, the somewhat lacking performance provided by Django is not a major issue.

3.2) Potential Solutions for Problem 2.2:

Dr. Winfree specified the use of Octave or Matlab when analyzing the data provided by Wearware. Octave is a programming language that is compatible with MATLAB but with a little bit easier syntax. MATLAB also requires a license to use the software which is one of the reasons the Dr. Winfree would prefer to use Octave because it largely has the same functionality as MATLAB. Another possible option would be to R studio to analyze the data. R studio allows data to be loaded in various formats and perform many different statistical functions on the data. From our research, Octave should be able to analyze the data properly and follow the trends of the data to relay the results to the users while satisfying the wishes of our client.

3.3) Potential Solutions for Problem 2.3:

The application needs to send a notification to the user. Our client specifically asked for text messages. The team needed to find a simple solution that the web application would be able to send these texts. The team research and discovered four possible solutions to solve this problem. The solutions include: Twilio,Bandwidth,Nexmo and Sinch. They all could be viable in solving the problem. Below is the analysis between the four.

a. Twilio

Twilio is cloud server application. It allows for developers to make/ recieve texts and calls from users and to user. Some of the features that will help solve our problem of user notification via text include: picture messages, reoccuring number, simple concatenation and many more. The pros and cons are as listed.

Pros	**Cons**
Reliable for sending/receiving	Most expensive option
Easy to set up and use	UI isn't the greatest
Expansive when it comes to phone services	Supports a large number of APIs so potentially overwhelming
Large database of documentation	

b. **Bandwidth**

Bandwidth is communication application that can be used for many things such as voice communication and more. Bandwidth differs from Twilio in a few ways. First since Bandwidth is its own network it allows for potentially quicker sending/receiving. Bandwidth also has a better support portal than Twilio if it is needed. Below is the pros and cons of Bandwidth

Pros	**Cons**
Great Support for users	Lots of logging in
Powerful API for users	Website Portal has many bugs
Own network so quite fast in comparison to other options	Hard to search through the available DID (Direct Inward Dialing)

c. **Nexmo**

Nexmo is a service that is the Voyage API platform. Nexmo is a large competitor to twilio. Nexmo differs from Twilio in a few ways. Nexmo focuses on overseas communication. They have a larger global reach than Twilio. Nexmo also costs less than Twilio. Below are the pros and cons of Nexmo.

Pros	**Cons**
Full Time Support	Difficult to use dashboard
Very Inexpensive	GUI doesn’t look good
Direct to many large carriers	Some countries have problems
Simple REST API	

d. **Sinch**

Sinch is cloud based communication platform that can be used to send and receive messages. It differs from Twilio by adding some new features which include free incoming text messages.Sinch also has an analysis board that can be used. The pros and cons of sinch are listed below.

Pros	**Cons**
Good integration between front and backend	Less online documentation
Supports over twenty thousand users	Notification Bugs are common
Cheap with free incoming options	

After analyzing the options presented above, the team decided to go with Twilio. While it is the most expensive option, the sponsor is willing to fund the solution and the other features seemed more beneficial to us. No one on the team has any experience in regards to sending notifications through a backend. The simple setup was the biggest factor for the team. We also liked the reliability factor that was included with Twilio. We strongly believe that this backend API will be beneficial with the current standing WearWare since the preexisting code will be simple to integrate with Twilio. The team plans to take simple courses and watch youtube videos to help increase our knowledge on the program before the initial setup process.

3.4) Potential Solutions for Problem 2.4:

The next major problem that the team needs to solve was a way to create a user interface that could be used by the researchers. Dr. Winfree was open to really anything since the current software does not have a UI so we decided to look into web applications that could connect to a database that would hold the analysis performed by the other solution. Jake had some experience with Ionic and the team decided to start there. The team did some research and found a few web frameworks that could be viable. These solutions include AngularJS, Meteor, and Django. Below are the comparison between them as viable solutions.

a. AngularJS

AngularJS is an open source website development framework that focuses on one page application. Angular is maintained by Google and has many features such as unit testing, HTML user interface and more. Below are the pros and cons of AngularJS.

Pros	**Cons**
Uses JavaScript	One paged applications
Document Object Model easy to adjust	Difficult to learn about the many features and how they work
Excellent server performance	Uses a Modal Visual Controller which can cause problems
HTML templates that can be used	

b. Meteor

Meteor is another open source website software framework. Meteor can also be used for mobile development on iOS and Android. Meteor can be used with its own engine called Blaze or the React framework. Below are the pros and cons of Meteor.

Pros	**Cons**
Uses Javascript	Not enough of network flexibility
Large number of packages available	No Widget library
Good communication between client and backend	Database selection is limited to MongoDB

c. **Django**

As stated above, Django is a Python framework for backend web development which can also be used to create an intuitive website. This solution would limit our different code as it is being used for API calls. Below are the pros and cons of using Django for a UI.

Pros	**Cons**
Uses Python	Slow so potentially slow websites
Lots of prebuilt in features that make set up easy	Lots of explicit declaration in the code
Security is quite good	

After reviewing the potential solutions to the UI problem, the team decided to go with Django as it is already being used for another part of the solution. The team believed that less technologies could be beneficial in regards to documentation. The team also like the prebuilt features and the security would be beneficial as the data for the UI would be sensitive. A security error would not be good for the project. The team doesn't think speed is the issue and is willing to potentially sacrifice it to us Django. We believe that the project will have a comprehensive website that can be accessed quite easily with Django.

3.5) Potential Solutions for Problem 2.5

The team already mostly solved the last problem. The team met up with Mike Fell. Mike is a employee of NAU and he was assigned to develop and edit WearWare. Mike was responsible for a large majority of the pre-existing code. As a team, we met up one afternoon and he explained the currently existing code. He explained that code already. The program already uses the Django framework and a SQL database to house the data that is mined from the Fitbit api. He also explained that WearWare is hosted on a local machine that we will eventually receive access to. Celery and RabbitMQ are also used for the pre-existing project but we will not be using these for our portion of the project. Mike’s valuable insight helped us improve our decisions for the technologies that we plan to integrate into the project as well as figure out solutions to problems we didn’t know we had which was the location of where to host the program.

4) Technology Integration

The above solutions are the teams tentative plan to tackle all the problems that the sponsor has. The team plans to house all the code that is needed for Django, Octave, and Twilio within a private BitBucket. This repository will also contain the code for the already existing work of WearWare. This centralized location will allow the team to keep tract and properly link all the files to produce the desired result. The pre-existing code will also provide a database that Octave can use to correctly complete the analysis. Django will also need to communicate with this database so that it can properly make the APIs that can be called from the server. This combination of database calls will be paramount in the project as without this being correct the program will have many problems. The team plans to attack this by focusing on the API calls before the analysis so that we can properly call the data. We also need to create a plan that creates the reports for the researchers and sends it to the UI. The UI also needs to be user friendly and that will be refined through the use of user testing which will include surveys as well as hands on usage. Our desired project relationships are shown in a diagram below **(Fig. 3).**

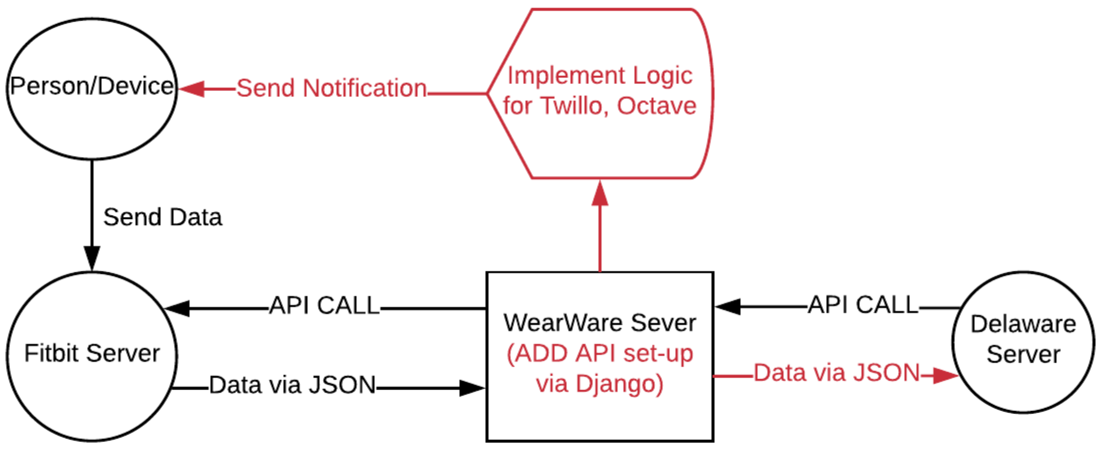

Fig. 3: The diagram displays the intended relationships by the end of the project. The black is representing outside work while the red will be done by the team.

5) Conclusion

Dr. Winfree wants us to create an API that allows Delaware to send a request to our API which will retrieve data from Wearware and send it to them, safely and securely. We plan on using Django because it is a framework that leverages Python and would line up nicely with wearware. Django will give us all of the functionality we need without being too complicated. We will then use Octave to analyze the data to satisfy the request of Dr. Winfree and we feel confident this is our best option. All of our research on the technologies seems to line up very nicely with what Dr. Winfree is expecting. Each of the technologies seem fairly straightforward and easy to use while also being powerful enough to complete our task. Below is a chart that outline all of the technologies we will be using along with a short description and our level of confidence with each one.

Technology	Description	Confidence

Bitbucket	Very similar to Github and will allow us to easily manage code	9/10, we are all familiar with Github.
Django	Python framework to create our web API.	7/10 fairly unfamiliar with creating a Web API but confident working with Python.
Octave	Alternative to MATLAB that is free and will allow us to analyze data.	7/10, We have never used Octave but the syntax seems fairly straightforward.
Twilio	Backend API to send messages via text	5/10, the team has no experience with programs like this but the large supply of documentation is very helpful to the team.

Our team feels fairly confident with all of the technologies that we have researched. Each one that we have found fits nicely into place without us having to try and stretch and accommodate something that will not likely work. Our next steps are to attempt to mock up simple examples that relate to our problem to verify that all of our technologies are the right fit. Our team is very optimistic with the outcome of this project after all of the research we did on all of the various technologies.