



# Team FitByte

## Testing Plan

4/3/19

Jake Farrar

Jacob Lemon

Austin Pederson

Ana Paula Chaves Steinmacher

Dr. Eck Doerry

Dr. Kyle N. Winfree

Dr. Gregory Dominick

### **1. Introduction**

Fitbit was established in 2007. Since then, Fitbit has enjoyed an enormous

amount of success. In 2017 alone, Fitbit sold 15.3 million devices. This is very interesting to both of our sponsors as they are both in fields of study relating to human interaction with technology and its potential effects on fitness. Our sponsors are Dr. Kyle Winfree and Dr. Gregory Dominick. Dr. Winfree conducts his research at Northern Arizona University and focuses on the use of devices engineered for therapies and assessment of health. Dr. Gregory Dominick is a researcher at the University of Delaware whose research focuses on health-related topics. For example, *Modeling Clinically Validated Physical Activity using Commodity Hardware* is the title of a paper recently published by Dr. Dominick. This study details how Dr. Dominick and Dr. Winfree worked together to create a model that reduced Fitbit inaccuracies and made the results be closer to what would be seen if a research-grade device had been worn instead of a Fitbit.

They have been conducting research by giving their study participants Fitbits to wear for a month. This allows them to track all of the data that the wearable technology offers. Currently, they have a piece of software called WearWare to help facilitate their research. WearWare consists of two major parts. A front end application that takes the form of a website and a back end application that deals with all of the data collection. The back end is the piece that grabs the data and exports it to a CSV. The front end is the piece that allows researchers to interface with the program conveniently, easily, and quickly. However, they are running into issues getting participant data, analyzing data for periods of inactivity, tracking progress towards goals, and returning the results to their researchers in a reasonable amount of time.

One of the glaring issues that our sponsors are facing is the sheer amount of time that it currently takes to process the participants' data in order to supply feedback. Researchers do not currently have an easy solution for dynamically analyzing the data in a reasonable amount of time in order to provide useful feedback to all of the participants in the study. Another issue is that researchers do not have an easy way to monitor their studies, the participants that are involved, or a quick way to apply Octave scripts to monitor data to find significant events. There is also currently not an easy way for new researchers to sign up to

use WearWare. As a team, we are looking to enhance WearWare so that all of these issues will be resolved.

We have been tasked with helping Dr. Kyle Winfree and Dr. Gregory Dominick create a solution that will allow them to automatically and dynamically assess their research participants' data in real time. We will be enhancing the capabilities of WearWare, implementing new features into WearWare, and creating a web Application Protocol Interface (API) that allows the team in Delaware (led by Dr. Dominick) to request and receive data in real time that is stored in WearWare's back end. The API will ideally let the Delaware team receive data from WearWare without having to go through the web application and wait for the large CSV to download. The API acts as a gateway that can get data from the database and send it via JSON to whoever requested it. The new features that our team will be implementing are a proper participant management interface, a way for researchers to manage their studies, and a way for data analysis scripts to be run on selected data on our server.

## **2. Unit Testing**

The Wearware project has plenty of room for errors and bugs to occur after many stages of development. The team aims to hit different types of testing in order to solve all the problems. The team plans to focus on unit testing. Unit tests are generally small tests that start with a few inputs that produce an output that provides valuable information to the developer that helps show if a program is working the way it is supposed to. The team will be using unit tests extensively to solve and fix bugs. The small nature of the tests will be very beneficial to the team as time is quite limited. The Django framework provides a large number of documents that show how unit testing and testing in general works within the program. This includes the unit test package and it can be run with a few simple commands.

The biggest problem that Dr. Winfree had with the currently existing WearWare was speed. Our main testing goal that the team will be focusing on is time or speed. Each major section of the project will be running with unit tests that show the speed that a task requires to complete. Other measures will

change for each function requires something a little different.

The project requires a large number of tests and the first test will be on the login page. Since the project is being hosted on an Amazon Web Server, the team will be running tests to make sure that the user is receiving the pages/project at a normal speed. This will be done on most pages but it will start here. The login page will also be running a unit test for checking the login information of the user. The user must enter a username as well as a password. The test will be checking the speed of the database check. It will also make sure that the information is right and an incorrect login will not change to the dashboard. The forgot password will have a test that will check that a link will be sent to a user that is in the database by another database check.

The goals that the team has made for unit testing and the goals are as listed:

1. Perform enough tests to make sure the return speed of the page is adequate for the user
2. Make sure the program is working the way we desire
3. Return results that help us improve the WearWare system

Once the user is logged into the WearWare system more tests will need to be completed. One of the first tests to perform is a unit test that will check if the user is on the correct portal since the system has a user and an admin dashboard. The speed test will also need to be performed to make sure that the system is returning the pages in a quick amount of time. The dashboard should run check that all of the links are clickable so that the right pages can be reached. The dashboard will also have studies that can be view. A test will be performed to check that a study that we know a user has is there and check to find a study that we know a user wouldn't have access to.

After the dashboard, the user heads to the participants' page. Another speed test needs to be here just to make sure that the page is returning in a quick amount of time. The number of participants will be shown and a test will be performed to check if the database is showing participants that the specific user

has access too. This will be checked by trying to find a user that we know the user has and trying to find participants that we know a user doesn't have. A test will also be created to check if a checkbox can be found and clicked on the page.

The dashboard as stated before can access study pages. We will perform a check that returns the speed the page was received and shown to the user. The study pages will have other links that will return back to the dashboard as well as other pages. We will perform a test to make sure all the links can be clicked. The user also sees participants that are apart of the study. We will be checking participants which we know they have but isn't in the study. This will ensure the data is not compromised. Checks will also be performed that make sure that the correct study is being shown again to make sure that the data is not compromised.

The study page will lead to the triggers page. The goal of the triggers page is to perform certain actions on the data set that will be shown on the other page. The first test will be checking that this page is returned to the user in a quick amount of time. We will also perform a test that checks that the user has their triggers and doesn't have other users triggers like the other tests we will check for a trigger that we know the user has for that study and check if it has a trigger that we know the user doesn't have for that study. The team will also have a test that checks if the link to go back to both the dashboard and the study page exists.

Going back to the dashboard allows the user to go to the account page. The account page has all the information related to one specific user. On that page, the team will be testing a few things. First, the team will check that the user's information matches in the database. This will include a check on the name, username, email, and phone number. A test will be done to get the speed that the page was returned to the user.

The dashboard also goes to the scripts page. The first test that we plan to perform is to check the speed that the scripts page is returned to the user. We want to make sure that the button to upload a new script is there and can we upload a script. Currently, we don't check what is being uploaded but that will

also be a different type of test which will be discussed later. A test will be performed so that the user has all of his own scripts and none of the other users' scripts. Similarly, an actions page will be accessed. The same speed test will be performed as well as a test for checking that a user has the right actions and none of the other users' actions.

Some of the other unit testing that the team will be performing is not involving the UI but more of the backend functionality. Dr. Dominick and his team need access to all the data that is in the system to perform their own analysis. As stated early, we will be using an API with the AWS. This will allow them to get the data. The tests will need to complete will be checking the security of the API. The first test will be checking that only certain users can access the API. This will be done by making a user that has this type of access to check if they can retrieve the database. Then a user that doesn't have access to check to see if they can retrieve the data. Based on what is returned it will indicate how safe our data is. The next test will be the speed that the data is returned to the user. Speed is quite important. This will allow us to check if we need to make improvements on how the API is accessed.

Based on all the tests listed above, this is how we will use the collected test data. We will be comparing all the speeds that were returned to us from every page. We will get the average from the data and make a comparison. If we know that our speed is not fast enough this will indicate that changes need to be made to fix this. The speed is one of our goals and it is quite important to us that is running at a quick rate. The speed will also be compared for returning data as well. The tests will also give us valuable data that can be used to help improve our code since it will indicate what the pages are missing/can't access.

Overall, the team believes that the unit testing plan will be useful in improving our code and producing a product that fulfills our clients' needs without bugs and errors and the team has committed multiple weeks to perform these tests.

### **3. Integration Testing**

Integration testing is mainly focused on how the many pieces of one's

software interact with one another. This includes interactions, data exchange and the flow of the system. The unit testing is aimed to check that the correct result is returned to the user while integration testing is focused on that the correct data is passed between the modules. The team wants to make sure that the system is working properly in regards to the data that is passed around as the data passage is vital in showing the correct information to the user. We need to make sure that it is functional no matter who is doing the interactions so this type of testing is vital in making software that can be used in the future.

The team created some goals to create a focus on developing an integration plan and they are as listed as followed:

1. Test each major data passing within WearWare to ensure that the program is working correctly
2. Perform lots of different types of data passage to make sure that the different data sets can be passed through the system
3. Make sure that Fitbit data is passed and usable

These discussed goals helped the team hone in on tests that would fall under integration testing and improve the software as a whole.

The data passage comes up within WearWare quite often. The first major one is the login page where users will pass their information to the system to check if they can log in. The page should pass the username and the password. We need to test that the password is being passed without being grabbed by an outside source since it would lead to a security flaw. The next large data passage will be the Apache Subversion (SVN) to the user. The admin updates the SVN and then the user needs to get the updated one. We need to make sure that the SVN updates correctly when the admin updates it. Fitbyte data is also passed throughout the system. We will be testing that the data is passed directly from the Fitbit API to the database as well as the database to the study pages.

Team FitByte will be tested what was stated above to make sure that the data is being passed correctly and no security leaks. The plan starts with our client, Dr. Kyle Winfree, in order to make sure that the data that will be passing will help him further research on heart health. Since Dr. Winfree is a researcher

he will be able to tell us if what we are passing is helpful to him. The tests above also show if our system is working together properly. This is the main focus on the plan since we want to help our client with a fully functioning system. If the data is being passed correctly the tests will show us which is helpful for this plan. The plan will perform these tests multiple times with different values to ensure that it is working with other sets of data. This will help the team indicate if the system is robust. Our goal is to make the system flow smoothly and correctly.

The team has allotted a week of testing. It will be divided amongst the team and then shown to one another to make sure that the testing is being done properly. First, one team member will contact Dr. Winfree via email and share what data is being passed to make sure that what we are grabbing and passing would be helpful to a user. Then the other team members will start the data passage. One needs to make sure that the data that is passed to the database is correct and not able to be grabbed by others. One needs to check that the Fitbit API is correctly grabbing and sending the data to the database. The team member that contacted Dr. Winfree will be making sure that the study data is being passed correctly and it is able to be used by the user. The goal of the plan is to make sure that the data being passed doesn't occur problems and many different sets can be passed. Our team believes our simple plan will be beneficial in the growth and stability of the software as it will ensure that data is being passed between modules correctly.

## **4. Usability Testing**

Usability testing is centered around the end user to validate that the software is working as expected, is easy to use, and promotes a positive experience for the user. The testing is necessary to ensure that the end user is able to properly navigate through all of the flows of our application. Users should easily understand our interface and intuitively get from place to place without getting lost or confused. Our team wants to be absolutely sure that all of the functionality of the web application works as intended and that users are able to take full advantage of all of the features available in the application. Along with the interface working properly, we need to be absolutely sure that the application



is easy to use no matter the user's familiarity with technology. We want the look and feel to be intuitive, allowing for the best experience for the end user. These are some of the reasons usability testing is so crucial to the development of successful software.

We created some goals to help guide our usability testing so that we would be confident with the integrity of our tests. Our team's goals for usability testing are as listed:

1. Test each major component of our web application to verify that they are working as intended.
2. Rigorously walk through each user flow to check that they are intuitive and would be easy for all users to complete, no matter the skill level.
3. Walk through the design of the entire web application to make sure it is easy to navigate without getting lost or stuck.

With all of the goals outlined, it helped our team to come up with a robust usability test plan for us to follow.

The end users will consist of researchers looking to monitor fitness data from groups of participants wearing FitBits. They are assumed to have a basic background in data analysis techniques. Our web application is able to provide the researchers with a unique tool to passively monitor and contact participants without the need for numerous hours of manually reviewing the data. Our application looks to save researchers lots of time by monitoring their study groups for them while providing a simple, secure, and intuitive interface to do it. Our team understands the implications that could come from a poorly designed application. The end users value security and ease of access for our solution to be viable enough to adopt. Our team wants to be sure that the web application is useful to as many people as possible, hence, the importance of the usability testing plan.

Team FitByte has created a detailed usability testing plan to follow in order to provide the best experience for the end user as possible. The plan starts with Dr. Winfree in order to double check that our solution is meeting his needs. His

feedback is crucial because he is a researcher and a client that is looking to take advantage of our solution. Dr. Winfree is the perfect candidate to check the progress of our application to make sure our interface looks and feels the way it should. The next phase of our test plan is to do rigorous testing on our interface. We will test various “what if” scenarios from the perspective of an inexperienced end user. We will test incorrectly formatting input fields, leaving data fields missing, accessing information that should be hidden to certain users, getting back to the dashboard from every page, and many other-edge-case scenarios. Our goal is to test our application in rainy day scenarios that would not likely occur under perfect conditions.

Team FitByte has allotted one week for usability testing. The first three days will be spent meeting with Dr. Winfree to discuss the look and feel of the website from an end user standpoint. We will then make modifications and review again with Dr. Winfree as many times as possible given the allotted time. We also plan on having our mentor run through various user flows. This will allow us to monitor another mock end user to get a critique on what feels natural and what feels clunky about our design. The four remaining days will be spent tackling as many edge cases as we can. We will go through user flows numerous times while also using different account types. Next, we will check the integrity of each component of our application along with trying to find ways to get stuck on different pages. With this testing, it will allow us to verify our end-user experience from various different perspectives. Our team believes that our usability testing will be effective and lead to a more successful experience for the end user.

## **5. Conclusion**

The goal of our web application is to reduce the amount of time that researchers spend sifting through large amounts of data and contacting participants in studies. For our solution to be effective, we had to create a robust and rigorous test plan to be sure that our claims could be met. The test plan is designed to limit the number of bugs, improve the overall quality, and provide a positive end-user experience while using our application. The plan is broken into three big sections that cover different aspects of our application. The first section

is unit testing which allows us to quickly test small pieces of our application for accuracy. Unit testing is important to reduce the number of bugs present while also double checking that basic functionality is working correctly. The next testing method is integration testing. Integration testing provides a way to check how the software is interacting behind the scenes. This involves checking that information is securely and accurately passed from one page to another, as well as, the data retrieved from the database is correct. The last section is usability testing which is centered on the end user experience. This testing method focuses on the look and feel of the website versus the technical side. Checking user flows, navigating the website, and verifying core functionality is all apart of usability testing. All of these testing methods combine together and allow us to confidently check the integrity of our web application. With this plan in place, Team FitByte can be positive that we are delivering a highly stable, usable, and secure solution to our clients, Dr. Winfree and Dr. Dominick.