CERIAS Tech Report 2023-6
Forensic Insights: Analysis and Visualization of Fitbit Cloud Data
by Poorvi Hegde
Center for Education and Research
Information Assurance and Security
Purdue University, West Lafayette, IN 47907-2086

FORENSIC INSIGHTS: ANALYZING AND VISUALIZING FITBIT CLOUD DATA

by

Poorvi Hegde

A Thesis

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Master of Science

Department of Computer and Information Technology

West Lafayette, Indiana

December 2023

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Marcus Rogers, Chair

Department of Computer and Information Technology

Dr. Smriti Bhatt

Department of Computer and Information Technology

Dr. Umit Karibiyik

Department of Computer and Information Technology

Approved by:

Dr. Stephen Elliott

To my family.

ACKNOWLEDGMENTS

I thank my advisor and mentor, Dr.Marcus Rogers, for providing me the opportunity to pursue this study and for his valuable guidance and support throughout the course of my degree program.

I thank my beloved parents and my entire family for their ever-loving support and faith in my abilities.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

AFib	Atrial Fibrillation
API	Application Programming Interface
CSS	Cascading Stylesheet
EDA	Electrodermal Activity
GPS	Global Positioning System
HTML	Hypertext Markup Language
IOT	Internet of Things
JS	JavaScript
MMS	Multimedia Message Service
OS	Operating System
POC	Proof of Concept
SMS	Short Message Service

ABSTRACT

Wearable devices are ubiquitous. There are over 1.1 billion wearable devices in the market today[ ]. The market is projected to grow at a rate of 14.6% annually till 2030[ ]. These devices collect and store a large amount of data[ ]. A major amount of this collected data is stored in the cloud. For many years now, law enforcement organizations have been continuously encountering cases that involve a wearable device in some capacity. There have also been examples of how these wearable devices have helped in crime investigations and insurance fraud investigations [ ],[ ],[ ],[ ],[ ]. The article [ ] performs an analysis of 5 case studies and 57 news articles and shows how the framing of wearables in the context of the crimes helped those cases. However, there still isn't enough awareness and understanding among law enforcement agencies on leveraging the data collected by these devices to solve crimes. Many of the fitness trackers and smartwatches in the market today have more or less similar functionalities of tracking data on an individual's fitness-related activities, heart rate, sleep, temperature, and stress [ ]. One of the major players in the smartwatch space is Fitbit. Fitbit synchronizes the data that it collects, directly to Fitbit Cloud [ ]. It provides an Android app and a web dashboard for users to access some of these data, but not all. Application developers on the other hand can make use of Fitbit APIs to use user's data. These APIs can also be leveraged by law enforcement agencies to aid in digital forensic investigations. There have been previous studies where they have developed tools that make use of Fitbit Web APIs [ ],[ ], [ ] but for various other purposes, not for forensic research. There are a few studies on the topic of using fitness tracker data for forensic investigations [ ],[ ]. But very few have used the Fitbit developer APIs [ ]. Thus this study aims to propose a proof-of-concept platform that can be leveraged by law enforcement agencies to access and view the data stored on the Fitbit cloud on a person of interest. The results display data on 12 categories - activity, body, sleep, breathing, devices, friends, nutrition, heart rate variability, ECG, temperature, oxygen level, and cardio data, in a tabular format that is easily viewable and searchable. This data can be further utilized for various analyses. The tool developed is Open Source and well documented, thus anyone can reproduce the process.

1. INTRODUCTION

Application Programming Interface or an API is a way for two or more computer programs to communicate with each other, enabling seamless communication and data exchange between them. It essentially acts as a bridge, allowing different software applications to interact and share information efficiently. In the constantly evolving field of digital forensics, where extracting valuable insights from various sources is of great importance, Application Programming Interfaces (APIs) play a vital role in accessing and analyzing data. One such API that could become significant in this field is the Fitbit Developer API. The Fitbit Developer API allows for the smooth integration of Fitbit's comprehensive health metrics into forensic investigations, providing a unique approach to reconstructing a user's profile. This study discusses how the Fitbit Developer API operates and its potential application in leveraging a user's data in forensic analyses.

1.1 Background

In recent years, Law enforcement organizations have been repeatedly encountering cases where either the victim or the perpetrator is found using a wearable device[ ]. According to an article by Pew Research Center, about 1 in every 5 Americans wears a fitness tracking device[ ]. According to another recent report, about 45% of Americans already wear fitness devices [ ]. The data stored on these devices and their respective cloud accounts could possibly contain some crucial information helpful for solving crimes more effectively and efficiently [ ]. Since it is potentially accepted as admissible evidence, law enforcement departments should be able to leverage it well[ ]. There have been many examples of how wearable devices have been crucial to solving crimes[ ],[ ],[ ]. Fitbit is one such Wearable device that continuously collects health and other information related to a user's day-to-day activities. According to a 2016 research by the University of Vido [ ], Fitbit covered more than one-third of the market share in the smartbands sector.

The data recorded by a Fitbit band could be locally stored within the band for up to a week before being transmitted to the Fitbit Cloud environment through a connected device when an internet connection is available[ ]. While Fitbit offers a mobile application for

users to access their data, the data displayed there is not necessarily the complete data that is available on the user. Furthermore, the mobile application lacks a convenient feature to swiftly retrieve data for a specific date, making it a limitation for forensic applications. Additionally, it is challenging to recover deleted data in the case that data is deliberately deleted from devices. Fitbit Cloud is a more comprehensive database for all the data collected and stored over a longer period. The data on the cloud is also hard to tamper with or erase. Fitbit provides a web interface in addition to the phone application to view this data. However, the web dashboard is not designed specifically for forensic purposes. So, it does not necessarily have all the desirable features that a forensic tool offers. Similarly to the mobile application, the web interface also does not provide the functionality to look up all the data for a specific date.

Application Programming Interface (API) is "a simple way for connecting to, integrating with, and extending a software system. Web APIs are web services that deliver data resources via web technology stack"[ ]. The operational mechanism involves web applications exposing a certain endpoint (think of them as doors) through which specific data can be either sent to or retrieved from the application. Two key components characterize most of the API requests - Type of request and Authorization information. "Type of request" outlines the intention of the request. That is, whether the requester wants to read, write, update, or delete the data - denoted as GET, POST, PUT, and DELETE requests respectively[ ]. The authorization segment ensures that the application recognizes the interaction as originating from an authorized user and not a malicious party.

Fitbit provides developer APIs that can be used by software developers to access the data on the user's cloud with the user's permission and build various types of applications. These APIs can also be made use of by law enforcement to extract the cloud data. The focus of this study is to leverage Fitbit developer API to fetch data from Fitbit Cloud and display it in a dashboard that is designed with a focus on forensic investigation.

1.2 Problem Statement

The data on Fitbit devices are retained for a maximum of a week. However, the data is retained for a longer time on Fitbit's cloud servers[ ]. Currently, law enforcement agencies can access the Fitbit cloud data related to a person of interest in one of 3 ways. 1) by accessing their mobile app, 2) through the Fitbit web dashboard, and 3)by performing forensic analysis on the physical device or the connected mobile device. All these methods have their drawbacks. The mobile application shows the information directly collected by the wearable devices and can get additional data stored on Fitbit's servers. The web dashboard has a summary dashboard page and provides searchability options on some of the types of data, but not all. Fitbit stores granular data, which is called intra-day data, for some categories. For example, minute-to-minute heart rate. Both the web application and the mobile app display this granular data in the form of graphs. So if one needs to look for a specific minute/second, as might be necessary for forensic applications, they would need to find it in the graph[ ]. Digital Forensic analysis on the physical devices might not provide older data due to memory capacity restrictions. Additionally, the study by S. Mcnary et al. [ ] states that they could not find any useful information about user activity on the mobile device connected to Fitbit through standard mobile forensic tools. Thus no tool or proof-of-concept is designed to aid law enforcement agencies to access and look up specific data from Fitbit Cloud related to a user. There's also no research done to understand if the web dashboard provided by Fitbit contains all the data that are collected about a user and stored on Fitbit's cloud.

1.3 Research Questions

1. Does Fitbit collect and save more information on its cloud than what's displayed on its web dashboard?

2. Can we have a tool for anybody to fetch and view the information that Fitbit has on a user from Fitbit Cloud?

1.4 Hypotheses

The hypotheses for this study are as follows:

1. H_1: Fitbit stores more information on its cloud than it displays to users on its web dashboard.

2. H_2: It is possible to extract all the information that Fitbit has on a person directly from the Fitbit cloud through Fitbit Developer Web APIs.

3. H_3: It is possible to perform the process of extracting data from Fitbit Cloud and displaying it through a tool.

1.5 Assumptions

The assumptions for this study are as follows:

1. Fitbit cloud Developer API endpoints are accessible and available for use.

2. The user has followed the setup instructions detailed in the tool's README file.

3. Fitbit User Authorization is not an issue as the intended users of the application are law enforcement agencies with authority.

1.6 Limitations

1. Since a Fitbit user would need to authorize access to their information to be able to use this tool, the scope of the tool is limited to the users themselves and law enforcement agencies (with proper authority).

2. The tool is developed using free-to-use open-source technologies and hosted on local-host infrastructure. So the tool is not expected to scale to mass usage unless hosted on a better infrastructure.

3. During the period of the research (August-December 2023), Fitbit did not have a policy against using its developer API to build a forensic tool. However, if such a policy is included in the future, this research might become inapplicable.

1.7 Delimitations

1. The tool is developed using popular technologies that can seamlessly integrate with the APIs provided by Fitbit.

2. It is tested on popular browsers such as Chrome, Edge, and Firefox and is optimized for the best experience in them.

3. This study does not focus on the Fitbit mobile phone Application and artifacts found outside of the app on mobile phones.

4. The study only uses Fitbit API for retrieving the data, traditional mobile forensic tools are out of scope.

1.8 Contribution of the Study

The following are the contributions of this study:

1. The study provides a proof-of-concept tool to collect and display important information that Fitbit has on its cloud, related to a person of interest, in a manner that forensic investigators can leverage to find the exact information that they are looking for and use it in a court of law.

2. The study compares the data from Fitbit Cloud with the data that Fitbit has made available through its web interface to find out if more information can be extracted from Fitbit Cloud through Developer APIs than what Fitbit makes available for the user.

2. LITERATURE REVIEW

With fitness trackers and smartwatches on the boom, several studies have been conducted on this topic over the years. This section highlights the previous work related to wearable devices, crimes, and laws involving wearable devices, forensic analysis of wearables, forensic analysis of Fitbit, and other related work. The section also compares previous work with the current research.

2.1 Wearable witnesses: Deathlogging and framing wearable technology data in Fitbit murders

The paper [ ] conducts a discourse analysis of 5 case studies and 57 news stories involving wearable devices in criminal trials and discusses death logging and datafication by examining how wearables may leave traces of the wearer's death helping in the reconstruction of a timeline of their death. Deathlogging using wearables examines how these devices actively contribute to understanding what happens to a person's body around the time of death by continuously collecting data, especially in forensic use where data collection is seen as constant observation, attesting to facts about human conditions. Taking a communication approach as opposed to a criminological approach, the article majorly focuses on analyzing how Fitbit and Apple smartwatches, being part of courtroom proceedings, are framed in newsroom reporting for five such murder cases - Connie Debate, Karen Navarra, Nicole VanderHeyden, Myrna Nilsson and Maria Ladenburger. The article says that the news around these cases mainly followed three themes: 1)wearable data as objective, 2)wearable technology as subjective witness, and 3)wearable technology as inadmissible (in and of itself) expressing embrace, skepticism, and critique, respectively. While discussing the first theme, the paper talks about the trial of Connie Dabate, which was headlined "Fitbit Murder" in the news. In this case, Fitbit data had shown that Connie had moved a quarter of a mile in her house an hour after the time that her husband had claimed she was killed in a home invasion. Fitbit data reconstruction helped in creating a timeline. While other data besides Fitbit, such as home cameras, alarm system logs, IP addresses, etc., were used to solve the crime, media headlines seemed to have given Fitbit the entire credit, giving it more capabilities

than it had. In exploring the second theme of fitness acting as the witness, the article delves into the murder of Karen Navarra. In this case, Fitbit's heart-rate data was used to show a spike in Navarra's heart rate followed by a rapid slowdown.

The media reports around this case all seem to exaggerate Fitbit's role, calling it the "witness" and giving it the ability to produce stories. Treating Fitbit as a testimonial source in legal cases reinforces datafication as a witnessing method, where the data generated by wearables serve as a reliable account of events. In this theme, the devices are not presented as directly revealing objective truths, but are portrayed as storytellers and interpreters, assuming the role of witnesses by being depicted as entities capable of conveying information. In the third theme, wearable technology is seen as having limitations, requiring new expertise to explain its functionality and interpret data accurately. The paper says that investigators have attempted to replicate data from digital devices, such as health data from an iPhone in Maria Ladenburger's murder, to support their admissibility in court.

In Ladenburger's case, police used health data from an iPhone to imply that the suspect had been "climbing stairs" around the time they suspected the victim's body had been moved. They recreated the scenario with an investigator of a similar build, and the movement data on the app showed him also "climbing stairs". While celebrated as a breakthrough, the method highlighted how certain activities, like climbing stairs, are coded based on slight changes in the device's altimeter when worn on the body. In the case of Myrna Nilsson, Caroline Nilsson was acquitted of murdering her mother-in-law, Myrna Nilsson, even though Apple Watch data contradicted her account, as the court deemed the circumstantial evidence insufficient to establish a clear case of murder.

Defense teams in murder trials, like Richard Dabate's and Caroline Nilsson's, questioned the reliability of Fitbit and Apple Watch data, highlighting the devices' inaccuracies for court testimony and illustrating the complexities and limitations of relying solely on wearable data as evidence. It can be seen from this paper that there is a consistent connection between witnessing and datafication in media coverage of cases involving trackers, suggesting that data from wearables offer new and potentially better ways of producing legal evidence, particularly in reconstructing crime scene elements like the time of death. However, the

study warns against simplifying these cases as "Fitbit murders," emphasizing that wearables are just one part of a diverse array of evidence in criminal investigations.

2.2 Wearable Device Data for Criminal Investigation

The paper [ ] considers the potential of wearable devices being used for crime investigations, primarily focusing on fitness trackers. The paper makes contributions to social network forensics, explicitly considers challenges in using fitness tracker data in forensic investigations, and claims to be the first to conduct an experimental study to try to use a fitness tracker to identify when a violent crime has occurred. The research uses Fitbit Charge 2 as the primary device. This was an experimental research. The experiment was that the participant 1)walked for 30 minutes, 2)Stopped at a specific location, 3)Kneeled on the ground and repeatedly hit the ground in front of him 10 times, 4) walked again for 30 minutes. After forensically examining the phone linked to the Fitbit device using the standard mobile forensic tools, the researchers found that the phone does not store any useful information about the user activity even though Fitbit collects it. They hypothesize that the reason for this is that Fitbit stores all the crucial information on its server, not on the device itself. This makes the current research even more important to law enforcement agencies as this research intends to make it easier to obtain and use the information stored on the Fitbit cloud servers.

2.3 Wearable Devices as Admissible Evidence: Technology Is Killing Our Opportunities To Lie

This is an article published in a law journal [ ]that discusses the balance between the benefits of technology in solving crimes and user privacy. It refers to multiple cases where a wearable device acted as one of the key pieces of evidence in crime investigations and trials. In one of the first cases involving wearables, *Commonwealth v. Risley*, the police questioned a womans rape claim when her Fitbit contradicted her statement to the police. She then faced three counts of a misdemeanor for prompting an emergency response and manhunt in response to her allegations. The paper mentions a litigation case where Fitbit's activity

data was used by a Canadian law firm to demonstrate how their client's physical lifestyle was severely affected by an injury, showing that Fitbit may assist in personal injury cases.

In another case in San Fransico, data from a Strava wearable device was used to show that the defendant was speeding and was responsible for an accident. Based on a detailed analysis of the Risley case, the paper suggests that the tracker data alone should not be a piece of admissible evidence in the court of law but should be combined with expert testimony. It advocates a need for clear legal frameworks, transparent privacy settings, and stringent rules for the admissibility of Fitbit data in criminal cases. It proposes that jurisdictions should update rules to treat Fitbits like cell phones and computers, requiring a warrant for legal searches by police, given their potential for storing sensitive medical information. The suggestion is to categorize wearable devices in line with existing rules in the Federal Rules of Evidence. The article anticipates the need for case law to establish precedent on the use of data from wearable technology in litigation over the next decade. Additionally, it recommends that Fitbit, beyond altering privacy policies, should set default information-sharing settings to private, with users having the option to change it to "anyone." If Fitbit data meets admissibility regulations, the article suggests the involvement of an expert witness to interpret the data in the context of litigation.

2.4 Forensic Analysis of Fitbit Versa 2 Data on Android

This study [ ] delves into the forensic analysis of the Fitbit Versa 2 and its smartphone app, examining the generated artifacts with an emphasis on areas of forensic interest for law enforcement. The study examines in detail the artifacts generated by Fitbit Versa2. While this model differs from the model used in the current research (Fitbit Charge 6), it still contains many of the same features. Thus, the artifacts discovered in this research are relevant to the current study. The researchers started by making a list of the features that they wanted to test, using Fitbit Versa 2's feature list on the Fitbit website. Then, they set up their mobile device to enable testing of these features by installing the required applications. They then made an image of the phone using MSAB XRYs full logical image acquisition option to act as the non-populated image. Following that, the researchers populated the

data by using the specific features needed and took a second image of the device to compare and contrast with the base image. They analyzed these images by opening them in MSAB XACT and Magnet AXIOM Examine. After a thorough investigation, the study found many interesting artifacts. The artifacts found in the /data directory on the devices include - GPS location, heart rate, calories, app ID, Web Cookies Database, empty credit card database, image of credit card, Alexa serial number, and credit card info. The study also found some user data, including oauth refresh tokens and credit card information in plaintext, which raised some security and privacy concerns. But it also found some reassuring security features such as not storing any data related to notifications produced by messaging apps and data related to the integrated Alexa app. The main forensic contribution of the paper is that it identifies where information relevant to law enforcement is stored on mobile devices for Fitbit versa 2. The paper concludes that there is room for improvement in Fitbits mobile application security.

2.5 Using Traces from IoT Devices to Solve Criminal Cases

This paper [ ] talks about using traces from different types of IOT devices to solve criminal cases. Since manual analysis of a large amount of data that is collected through many different types of IOT devices at a crime scene is a time-consuming and difficult process, this paper aims to solve the issue by presenting a data extraction and processing platform. The platform uses Lambda architecture. The proposed pipeline involves importation from various IOT devices, a lambda layer that consists of a batch layer, a speed layer & serving layer, and a storage layer. The lambda layer also has specialized APIs to collect different data traces. The paper also mentions the different types of data traces that are collected by different IOT devices and the forensic implications of that data. Smartwatches can collect information on 1)settings and configurations with the forensic implication of getting access to the configuration of devices, 2)Voice commands with the forensic implication of getting access to voice commands used by the owner, 3)SMS/MMS with the forensic implication of getting access to SMS and MMS belonging to the device, 4) Paired device information with the forensic implication of accessing the information on devices connected to the device and

5)Notifications with the forensic implication of getting access to the notifications received by the device.

2.6 How can data from fitness trackers be obtained and analyzed with a forensic approach?

This paper [ ] looks into three different fitness trackers and provides general guidance on how to forensically analyze these devices. The paper particularly talks about the Xiaomi Mi Band 2, Fitbit Charge 2, and Huawei Band 2 Pro. There are many findings from this research. Some of the important ones that are related to the current research are: 1) The Fitbit Charge 2 device measures heart rate every 5 seconds in default mode, and while working out, it measures every second, 2) The data collected by Fitbit Charge 2 is first synchronized with the cloud, only after which it can be viewed on the connected smartphone app, 3) The connected smartphone acts as a transmitter for the trackers to send a large amount of encrypted data to the servers, 4) The tracker records GPS information only if the phone is connected to the internet, 5) With Fitbit Web API, the time series calls did not work properly for the researchers, 6) The API limits the requests to 74 per hour, 7) There's an API call that is not listed in the official document. The paper describes it as "The only interesting call we found offers the time at which the user was sitting: https://api.fitbit.com/1 /user/[user-id]/sed/date/[date].json" (p.4). This can be used to find the time at which the user was sedentary, 8) One way to evaluate the Forensic soundness of the developed tool is by considering 4 criteria by Rodney McKemmish - Meaning, Errors, Transparency, and Experience. In addition to these points, the paper also suggests a procedure to ensure the forensic soundness of data from Fitness trackers.

The paper also talks about the databases that are found in the linked device and the information they hold. The "exercise_db" database holds various tables related to exercises. For example, a table called "EXERCISE_EVENT" holds timestamps and coordinates. Combining this with the information from another table called "EXERCISE_SEGMENT", it's possible to construct the path of the exercise."fitbit_db" database contains the information about the user's profile and devices. The database "heart_rate_db" contains the data on average heart rate each day and also on different heart zones such as cardio or fat burn. The

"sleep" database holds information on the user's sleep routine - start and duration of the recorded sleep, minutes spent asleep and awake, and also the information on the exact time and frequency that a user enters a particular sleep phase.

The researchers evaluated their tool for forensic soundness by considering Rodney McKemmish's [ ] four criteria - Meaning, Error, Transparency, and Experience. They then checked for the completeness and robustness of the tool. The POC tool is also similarly evaluated.

2.7 Digital Forensic Analysis of Fitbit Wearable Technology: An Investigator's Guide

This paper [ ] proposes a guide on forensically extracting and analyzing data collected by two of the Fitbit devices (Alta tracker and Ionic) using open source tools, Autopsy Sleuth kit and Bulk Extractor Viewer. While the methods they use are much different from the current research, the paper presents some findings that are relevant here. Mainly: 1)Fitbit stores day-to-day data for 30 days and minute-to-minute data for 7 days before syncing them to the cloud, and 2)Every log entry as a tag in the end ("Alta"/ "Inonic") that shows if the entry was made automatically or manually added by the user. The paper also explains that the method of downloading data from the Fitbit Cloud using the Fitbit web application may take up to weeks or a month to process the request and get hold of the archive. The paper also provides some information regarding a forensic investigation: 1) Devices must be turned off at the time of seizure, or else they might accidentally log information, leading to integrity issues; 2) The Investigator should sync the device on the user's computer/smartphone right before seizing the device; 3)It isn't possible to locate user's call logs and voice notes through Fitbit forensic analysis alone. Thus, other methods might need to be employed for this information.

The paper provides a detailed description of the data population procedure that they followed. This is relevant to the current research and has been replicated in the data population phase of the same. The population phase was divided into two steps. The first step involved identifying the features of the alta and ionic devices to understand what features are important to a forensic investigation. The procedure that they followed for this involves:

1) Discovering the available features in the devices by reading the documentation and going through Fitbit's website.

2) Discovering the features personally by manually going through the device and applications.

3) Listing the above-discovered features and description of each of them.

4) Creating a brief criminal scenario and identifying the potential value of each of the features.

5) Recording whether these features are populated manually, automatically, or both.

The second step involved populating the discovered features using Desktop, Web, Mobile applications, and the Device itself. They also modified, added, and deleted data manually and recorded the specific changes for later comparison during forensic investigation.

2.8 Collection and Processing of Data from Wrist Wearable Devices in Heterogeneous and Multiple-User Scenarios

This paper [ ] provides the background needed for the current research on fitness devices and the ecosystem around them. The most common sensor present in over 75% of fitness devices is the accelerometer. Some of the other common sensors are Heart Rate, GPS, Gyroscope, Compass, Microphone, and Ambient Light. Some of the sensors that are present in high-end devices are Barometer, Altimeter, Camera, and Thermometer. The main systems that are involved in data collection are wearables, smartphones, computers, and cloud servers. The data cycle starts with collection using the smartwatch/fitness band. This data is sent to the linked device like s smartphone/tablet/PC. The device then sends the data to a proprietary server environment. This environment provides cloud services. Third-party applications and developers can then request the data from the server environment through REST API endpoints. The paper does a comparison of what vendor provides what option for data access by the third party. While Google Fit, Jawbone, and Microsoft Health provide both SDK and API options, Apple Health and Samsung Health only provide SDK options, and Fitbit only provides API options. The paper also notes some other key information regarding the data. The important details for Fitbit are that the sleep record is noted minute by minute, and every minute is tagged with a sleep state. Fitbit also distinguishes

between sleep states by using the tags "awake" for light brief movements and "really awake" for strong long movements.

2.9 Forensic Analysis of Wearable Devices: Fitbit, Garmin and HETP watches

This paper [ ] presents the forensic analysis of Fitbit Charge HR, Garmin Forerunner 110, and a generic low-cost HETP fitness tracker. Useful to the current study, this paper addresses the questions of data from fitness devices being useful in forensic investigation and also tries to find out the accuracy of evidence generated by fitness bands. The paper uses manual, physical, and logical data extractions as suited. As much of the work related to forensic analysis of wearables until then used the smartphone devices linked to the wearables for the analysis, the researchers present techniques for analyzing the wearables themselves. They use the tools Forensic Toolkit (FTK), FTK Imager, and Autopsy for these devices. Some of the notable findings of this paper are that 1) Fitbit stores only 31 days' worth of data on its device, 2) There is no clear picture of the deleted data in the Garmin device, 3) The paper attempts to sync the devices with an independent unlinked account and is successful in doing so for Garmin, 4) There are some minor differences in the values of some of the metrics between the devices. Still, they are well documented by the manufacturers, and they haven't affected any court cases that involve wearable devices as part of the evidence.

2.10 Human Behavior and Anomaly Detection using Machine Learning and Wearable Sensors

This paper [ ] makes use of smart bands and wearable devices for detecting and analyzing human behavior. The notable contribution of this paper that's useful for the current research is the architecture of the application that they built. The architecture is similar to the architecture of the POC application. The researchers use the Fitbit Inspire 2 wearable. The data collected by this device is sent to a connected mobile device through Bluetooth. The device further sends the data to the Fitbit Cloud. The researchers then make use of the Fitbit developer API by first completing an authorization workflow. After the authorization, the application polls the Fitbit cloud for updated information about a user. Once

the polling system finds new data, the data is requested and then queued using a message queuing tool called RabbitMQ. The data goes to the MySQL database from here, which is later used by the custom application, which consists of Spring-Boot for the back end and React with TypeScript for the front end. They use Redux Saga for the API calls. While the POC application has a similar architecture, it does not make use of the message queues. The technologies/tools used to build the application are also different.

This paper explains step-by-step the process of using Fitbit API. The first step is to register on dev.fitbit.com as a developer. Next, they must register an application to get access to the data. This process requires URLs for the application website, organization, Terms of Conditions, and Privacy Policy. Once these steps are completed, the developer is authorized to request data. The data can be of two types. 1) Summary for some time and 2)Intraday time series. Intraday can only be accessed by a third-party application if it is declared as "private" in the account. This limits the number of users for the application. But Fitbit allows third-party applications to obtain intraday data for multiple users if the application that is being developed is for research purposes.

The paper also describes the steps involved in the authorization process for Fitbit. To get the monitored data registered by the Fitbit bracelet, they needed to first obtain a token from Fitbit. This token is granted by the Fitbit user. For this process of user authorization and API authentication, Fitbit uses OAuth 2.0[ ]. Once the user gives authorization, Fitbit sends an access token to the application. The flow that the researchers here followed involves the following steps:

1) The system displays the authorization page.

2) User consenting to data sharing and providing authorization.

3) Fitbit generates an authorization code to the application server using the redirect URL provided.

4) Server exchanging the authorization code with Fitbit to obtain access and refresh tokens.

5) The obtained tokens are stored in the database together with user information.

2.11 Security Analysis of Wearable Fitness Devices (Fitbit)

This paper [ ] researches the security of the Fitbit ecosystem involving the device, communication between the wearable and connected smartphone, the Android application, and the network traffic between the devices and the Fitbit cloud. The paper also analyses the data collected by Fitbit, and similar to one of the goals of the proposed research, this paper examines if all the data is made available to the customer. However, many of the techniques used here for that purpose are out of the scope of the proposed research. In addition, the paper also demonstrates some of the attacks possible on the devices. Following are some noteworthy findings and insights from this research regarding Fitbit - 1) Fitbit uses Bluetooth Low Energy (BTLE) protocol instead of the standard Bluetooth 4.0 to synchronize data with its smartphone, 2)The smartphone communicates with the cloud through an encryption TLS session over the internet, 3) It was possible to obtain private addresses of any Bluetooth device in the BTLE range of a Fitbit Flex as it responded to broadcasts from any device in the range, 4) The private address of the device does not change. So, it is possible to track a person based on their Fitbit Bluetooth advertisement, and 5) The research found out through logs by inspecting the phone that Fitbit collects more data than it provides to the user. For example, it collects extraneous information about users, including MAC addresses of nearby Fitbits.

2.12 Leveraging the Fitbit API to Share Activity Levels with a Trusted Caregiver

Very similar to the intended use of Fitbit API in the POC application, this project **chenleveraging** leverages Fitbit developer API to develop a mobile application. The goal of this research is to better understand the types of health data that individuals are comfortable sharing with their friends and family and the context in which sharing such sensitive information makes sense. The research uses Fitbit API to authenticate a Fitbit user on another mobile user's phone so that he or she receives access to various Fitbit health-related information. The approach taken by the project is that they built a mobile application that redirects users to Fitbit's authorization page. It then redirects users back to the application's

redirect URL with the access token. The application then stores the access token that it receives on the client side. Fitbit uses a Callback URL that includes the access token as a fragment of the URL itself to redirect authenticated users. The application parses the access token and saves it on the client side to use when making future API calls. Every time that the application needs to retrieve information, it sends a request to the Fitbit API including the users' token. This sends back the data from the Fitbit database in the form of a JSON response. The application parses this response and displays the information.

Current Study

All these previous studies help the current research in different ways. Gilmore et al. [ ],Chauriye[ ] and Mcnary et al. [ ] highlight the importance of this research. Yoon et al. [ ] and J.M do Valle et al. [ ] discuss the process of collecting data for digital forensic research and also help in understanding which data is of forensic importance and where they're located. Similar to the tool developed for the current research, F. Hantke et al. [ ] also provides an open-source tool that uses different methods to extract and report these trackers' data. But the research is specific to a fitness tracker, and smartwatches were out of their scope. This study differs from the current study in its purpose. While F. Hantke et al. is intended to aid digital forensic analysis of the devices, the current study is to make use of cloud data rather than the physical devices themselves. This paper also provides a method to evaluate tools, which is followed in the current study. Many of the findings of the study by Á. MacDermott et al. [ ] are useful for the current research as the paper explains forensic investigation implications of the data collected by Fitbit and other wearables. The paper by Bozdog et al. [ ] outlines the steps to Obtain Fitbit cloud access tokens. The POC application follows a similar flow in the initial steps to obtain Fitbit cloud access tokens. Similar to this paper by B. Cyr et al. [ ], the current research also aims to find out the differences in the data collected by Fitbit and the data made available to the user. However, the scope of the research only applies to the Fitbit cloud and access to the data through developer APIs.B. Cyr et al. only mention that as a possible future research area.

The POC application also follows similar steps as mentioned in the paper by K. Chen **chenleveraging**. The key differences are that the POC application is web-based, not mobile phone-based and it displays all the information as opposed to only caregiver-relevant information that this research displays.

3. METHODOLOGY

The study aims at developing a proof-of-concept tool to fetch data stored on Fitbit Cloud using Fitbit Developer API and present it in a manner that's useful for forensic investigation, and performing a comparison between Fitbit data retrieved through this means and published by Fitbit on the user's web dashboard. This section is divided mainly into five parts. 1)Set up 2) Data; 3) Application Backend; 4)Application Frontend; 5)Data Comparision. Figure depicts how steps 2,3 and 4 work.

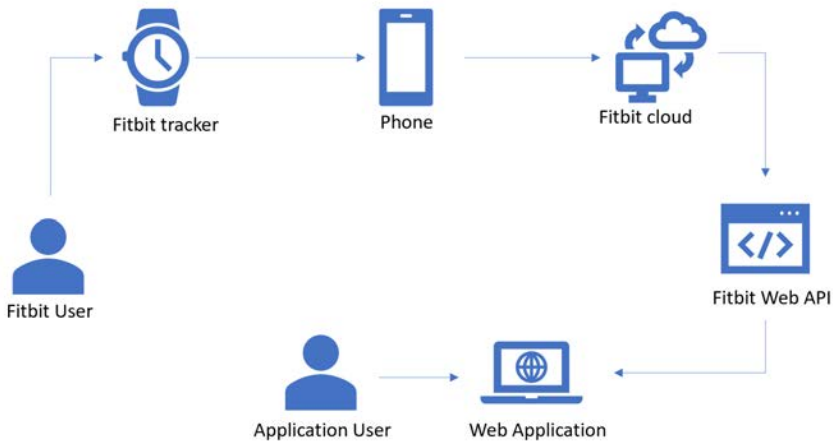


Figure 3.1. Fitbit Process

3.1 Set Up

This section describes all the setup that was needed for this study - device and account, application registration, application backend, and application front end.

3.1.1 Choosing the devices and creating Fitbit account

Fitbit has many models of activity trackers. All the models were compared and contrasted. Fitbit Charge 6 was chosen as it possesses a balance between features and affordability. It is running Fitbit OS 5 with Firmware version 66.20001.202.94. The mobile device used in the study is the researcher's personal Android device - Samsung Galaxy S23 Ultra

running Android version 13. Fitbit requires an account on the Fitbit cloud to be able to use any Fitbit wearable. Fitbit offers two types of accounts, the free and the premium. The premium version offers more features, such as personalized suggestions and advanced health tracking. To leverage all the features provided by Fitbit, the study used a premium account.

3.1.2 Registration of the Application on Fitbit Developer Portal

To be able to use the Fitbit Developer API, an application needs to be first registered on the Fitbit developer site.

1. Create a Fitbit Account: Sign up at https://www.fitbit.com/signup.

2. Go to the Fitbit Developer Site: Visit https://dev.fitbit.com/ and sign in with the Fitbit account.

3. Register a New Application: Click on "Manage" in the top menu, then "Register An App". This will produce a form where the details about the application need to be entered.

4. Application Registration: By clicking on "Manage" in the top menu and then "Register An App", a form is presented where details about the application can be entered.

5. Form Completion: The required information about the application must be entered. This includes the application name and description, application website and organization, OAuth 2.0 application type(server), callback URL(This can be any URL if the application is not hosted anywhere. In this case, it's given as www.purdue.edu), and default access type.

6. Terms of Service Agreement: Read and Agree to the Fitbit API Terms of Service.

7. Application Registration: Click the "Register" button to register the application. Once the application is registered, a client ID and client secret are provided by Fitbit. These can be used to authenticate with the Fitbit API in the application.

3.1.3 Setting up the Application Back End

The application contains a back-end component. More about this will be discussed in the next section. The setup for the back end requires certain tools and packages to be installed and the project created.

The latest Python 3 version is installed, followed by Django. through with the command "pip install Django". The next step is to create a project in Django. The project here is called "FitbitCloudApplicationBackend". It can be created using the command "django-admin startproject FitbitCloudApplicationBackend".This will create a new directory named FitbitCloudApplicationBackend with the basic files and directories needed for a Django project. In order to run the application the server application, the development server must be started. This can be done by navigating into the project directory and starting running the command "python manage.py run server". This should give an output indicating that the server is running, and the site can be accessed at http://127.0.0.1:8000 in a local web browser. The folder structure for the backend can be seen in .

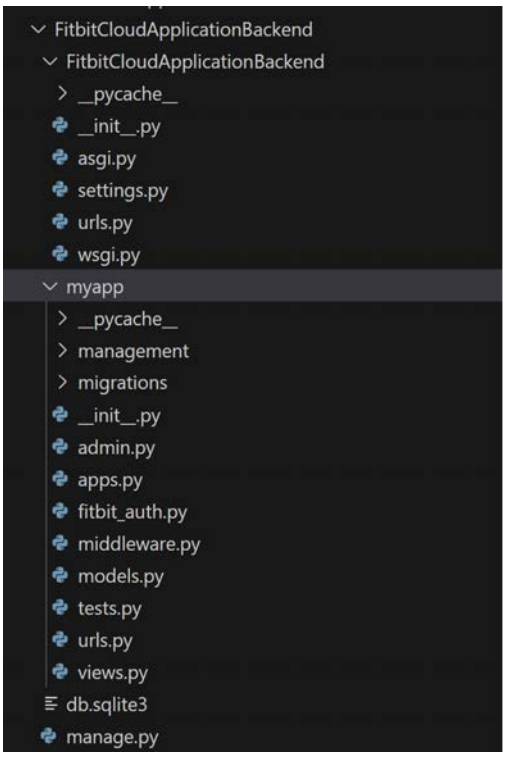


Figure 3.2. Back End Code Structure

3.1.4 Setting up the Application Front End

The application also contains a front-end component consisting of the technologies HTML, CSS, and JS. These technologies don't need any new installations. To ensure distinction between the front end and the back end, a separate folder is created for the front end inside the project directory. Three different subfolders are created each for HTML, CSS, and JS. Inside each of these folders, further subfolders are created for each of the Fitbit features - Activity, Body, Breathing, Cardio, Devices, ECG, Friends, Heartrate, Nutrition, Oxygen, Sleep, and Temperature. The folder structure can be seen in and . The application uses jQuery and specific jQuery UI components. To be able to do that, the project needs a jQuery UI package. This can be downloaded from https://jqueryui.com/download/. This site also allows for customizing the exact widget to include in the package in order to optimize the web page load latency. For this project, only the calendar widget is required. This downloaded package must be added to the JS folder.

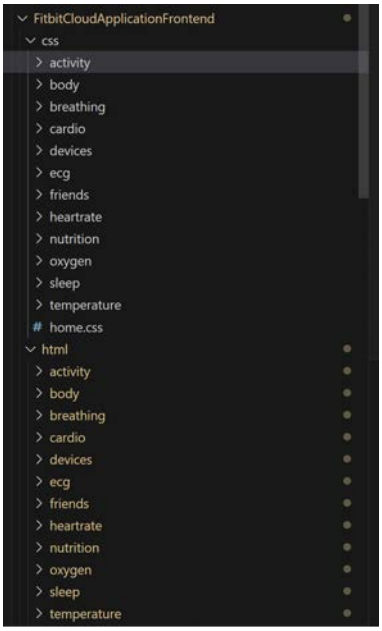


Figure 3.3. Front End Code Structure

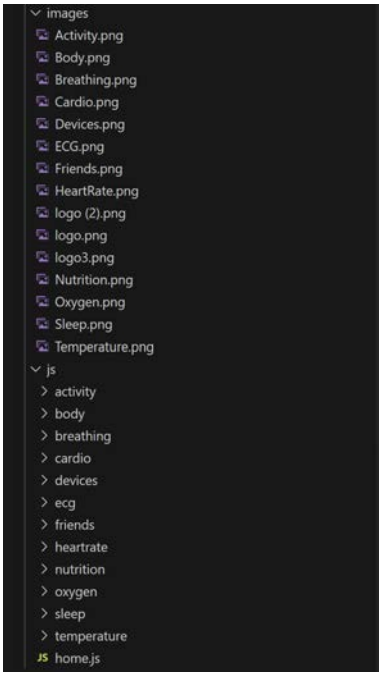


Figure 3.4. Front End Code Structure

3.2 Collection and handling of data

The study requires Fitbit data to be collected in a forensically sound manner where all the data is tracked. It's also necessary to ensure all the features provided by a Fitbit wearable device are used in order to get a complete set of data. There are two steps involved in this process.

The first step is to make a list of all the features that Fitbit Charge 6 offers with the help of Fitbit Documentation and website and understand which are of relevance in the context of a forensic investigation. A preliminary analysis indicates that these are the features that Fitbit Charge 6 offers :

1. Fitness Features such as Heart Rate monitor on gym equipment, Daily Readiness Score, Exercise Modes, Heart Rate Tracking, Activity time in heart rate zones, GPS, Cardio Fitness score, Automatic exercise recognition, All-day activity tracking, Workout intensity map.

2. Health features such as ECG assessment, High-Low heart rate notification, SPO2 tracking, Skin temperature variation, Resting heart rate tracking, Breathing Rate tracking, Irregular heart rhythm notification, Blood Glucose tracking in the app, and Menstrual health tracking.

3. Stress & Sleep features such as Stress Management Score, EDA scan app, Sleep score, Sleep Profile, Sleep tracking & Sleep stages, and Smart wake alarm.

4. Other smart features such as Do-not-disturb and sleep mode, Music controls, Google Maps, Google Wallet, Call, text & App notifications, Find My Phone, and a 7-day long battery life.

3.2.1 Understanding the meaning of the data

To understand which of these are of relevance for a forensic investigation, the meaning of each of these data should be recognized.

1. Heart Rate monitor on gym equipment - Heart rate collected through the Fitbit tracker can be viewed in real-time on the display of selected workout machines.

2. Daily Readiness Score - Tells if the body is ready to work out or needs to rest.

3. Exercise Modes - Ability to track exercises manually and get live status on the tracker.

4. Heart Rate Tracking - Tracking of Heart Rate, which also powers other features like tracking of Calories Burnt, sleep stages, etc.

5. Activity time in heart rate zones - Tracks time in different heart rate zones.

6. GPS - Provides Pace and Distance information.

7. Cardio Fitness score - Estimate Fitness level by seeing how well the body uses oxygen while exercising.

8. Automatic exercise recognition - Automatically detects when exercising and logs it post-workout.

9. All-day activity tracking - 24/7 tracking of steps, distance, calories, and time in the active zone.
10. Workout intensity map - Track workout pace and intensity throughout a map after an outdoor workout.
11. ECG assessment - Ability to do ECG and assess heart rhythm on demand.
12. High-Low heart rate notification - Notification sent on the Fitbit App if the heartbeat is above or below the person's normal range.
13. SPO2 tracking - Tracks blood oxygen saturation level.
14. Skin temperature variation - Tracks variation in skin temperature level from baseline.
15. Resting heart rate tracking - Tracking of heart rate when the user's body is not active.
16. Breathing Rate tracking - Tracks changes in breathing rate.
17. Irregular heart rhythm notification - Sends notification on Fitbit app if Atrial Fibrillation or irregular heartbeat is detected.
18. Blood Glucose tracking -Users can log blood glucose levels to get advanced analysis on it.
19. Menstrual health tracking - Users can log periods, and symptoms, find patterns in the cycle, and estimate ovulation period.
20. Stress Management Score - Shows how well the body is handling stress every day.
21. EDA scan app - Can do EDA mindful sessions on the wrist to help relax.
22. Sleep Score - Tells user's quality of sleep.
23. Sleep Profile - Matches users with a ' sleep animal' and provides personal sleep analysis every month.

24. Automatic tracking of sleep duration each night and breakdown of time spent in light, deep, and REM sleep stages.

25. Smart wake alarm - vibrating alarm feature.

26. Do-not-disturb & sleep mode - silences calls and notifications and turns the display off.

27. Music control - Can control YouTube start, stop, and skip from the watch.

28. Google maps - Get navigation directions on the watch.

29. Google Wallet - Make purchases through the watch.

30. Call, text & App notifications - Get notifications on the tracker when the connected device is nearby.

31. Find My Phone - Use the tracker to make the connected phone ring.

32. Food Log - Users can log the food that they ate and get an analysis of calories and nutrients in it.

3.2.2 Mock scenario that uses different Fitbit data

Following the methodology described by A. Almogbil et al. [ ], To understand which of these features are important, a scenario is created to check what significance these features have in the scenario.

Imagine a scenario where police have found a young, seemingly healthy individual wearing a Fitbit tracker dead on a running track. There don't appear to be any wounds and struggle marks. The investigators need to decide if it was death by natural cause or if there's foul play. The following are the possible insights that can be gathered from Fitbit alone that can aid in the investigation.

Daily readiness score could help in understanding if the person was already ill before he left for his morning run. Exercise Modes and Activity Tracking can give information on the person's exercise routine and physical activity levels on the day of his death and in the

period preceding it. If the person had a rigorous exercise schedule, combining that with heart rate data over a period of time could potentially help in understanding if his routine suited his heart condition. A workout intensity map can be used to understand if there was a pattern of intensity levels that may be related to the death. High-low notification and AFib notification can be used to check if there was something unusual with his heart that had been identified. Glucose Monitoring data could indicate if the person had any history of diabetes. Stress Management Score and any EDA scan results combined with sleep score and sleep stages could give some insight into the person's mental status. Irregular patterns in SPO2 data and skin temperature data could indicate if the person was sick. The food and nutrition information could help in understanding if any food he ate, the ingredients, and the sources they came from could have anything to do with the death. The Activity Time series contains minute-to-minute data of steps. This could tell the exact time that the person stopped moving. The heart rate time series contains hourly data on heart rate. These two together could help in determining the time of death. Thus, we can conclude that all the health metrics data could be useful for law enforcement agencies.

3.2.3 Data population on the Fitbit device

The next step is to populate the data in a way that all the relevant features can be utilized. This is achieved by personally using the device to populate the data for 10 days and maintaining a corresponding activity log for all the features mentioned above.

1. The Fitness data such as exercise modes, heart rate tracking, activity time in heart rate zones, GPS, cardio fitness, auto exercise detections, all-day activity tracking, workout pace & intensity tracking were populated by wearing the device for at least 6 hrs a day while performing various activities including walking and running along a route and other cardio activities such as dancing, and while resting.

2. The health data that are collected automatically, i.e., Skin temperature, irregular heart rhythm, resting heart rate, were populated by using the watch for at least 6 hours a day for 10 days. The health data that is tracked by explicitly activating the tracker, i.e., ECG assessment, was tracked by using these features at least 5 times. The other

health data that needed to be manually entered on the app, i.e., nutrition and body metrics, were populated by creating at least 10 logs.

3. The Stress, Sleep-related data, & breathing rate were populated by wearing the watch while sleeping at night and for a minimum of 6 hours during the day for 10 days.

4. The data related to other features, i.e., Do-not-disturb and sleep mode, music controls, Google maps, Google wallet, call, text & App notifications, and Find My Phone, were populated by using the watch to manage these features for 10 days.

   The log for the data population is shown in the table below in

   and

Date	Time Start	Time End	Activity	Available on API?	Location
2-Nov	7.10am		Sleep end	N	
2-Nov	7.10am	9.10am	walking around in home	Y	Activity Time Series
2-Nov	9.10am	9.30am	walk to campus + veo ride	Y	Activity
2-Nov	11.25am	11.50am	walk back home	Y	Activity
2-Nov	11.50am	2.30pm	some movement at home	Y	Activity Time Series
2-Nov	2.30pm	3.30pm	Nap	Y	Sleep Log
2-Nov	3.30pm	11.59pm	some movement at home	Y	Activity Time Series
3-Nov	12.15am	6.40am	sleep	Y	Sleep Log
3-Nov	6.40am	9.00am	some movement at home	Y	Activity Time Series
3-Nov	9.00am	1.00pm	Sit in car from EST time zone to CT	Partial	Activity Time Series
3-Nov	12.00pm	2.30pm	walk in conference center	Y	Activity Time Series
3-Nov	2.30pm	3.10pm	sit	Y	Activity Time Series
3-Nov	3.10pm	5.30pm	walk in conference center	Y	Activity Time Series
3-Nov	5.30pm	7.00pm	mostly sitting, with some walking here and there	Y	Activity Time Series
3-Nov	7pm	7.30pm	walk in conference center	Y	Activity Time Series
3-Nov	7.30pm	11.59pm	mostly sitting, with some walking here and there	Y	Activity Time Series
4-Nov	12.00am	1.30am	sit	Y	Activity Time Series
4-Nov	1.30am	6.45am	sleep	Y	Sleep Log
4-Nov	8.20am	8.40am	walk	Y	Activity
4-Nov	8.40am	9.00am	some movement at room	Y	Activity Time Series
4-Nov	9.00am	9.50am	walk in conference center	Y	Activity Time Series
4-Nov	9.50am	12.00pm	sit	Y	Activity Time Series
4-Nov	12.00pm	12.10pm	short walk	Y	Activity Time Series
4-Nov	12.10pm	7.00pm	sit	Y	Activity Time Series
4-Nov	7.00pm	7.30pm	walk in conference center	Y	Activity Time Series
4-Nov	7.30pm	8.00pm	sit	Y	Activity Time Series
4-Nov	8.00pm	8.30pm	walk in conference center	Y	Activity Time Series
4-Nov	8.30pm	10.00pm	sit	Y	Activity Time Series
4-Nov	10pm	10.30pm	walk	Y	Activity
4-Nov	10.30pm	11.59pm	Sit in car from CT time zone to EST	Partial	Activity Time Series
4-Nov	11.59pm	1.30am	Sit in car and some walk	Y	Activity Time Series
5-Nov	1.30am	10.00am	Sleep	Y	Sleep Log
5-Nov	10.00am	5.30pm	Some movement inside house	Y	Activity Time Series
5-Nov	5.30pm	5.50pm	Walk	Y	Activity
5-Nov	7.50pm	8.10pm	Walk	Y	Activity
5-Nov	8.10pm	11.59pm	some movement in house	Y	Activity Time Series
6-Nov	12.00am	12.30am	some movement in house	Y	Activity Time Series
6-Nov	12.30am	8.40am	Sleep	Y	Sleep Log
6-Nov	8.40am	2.30pm	some movement in house	Y	Activity Time Series
6-Nov	2.30pm	2.50pm	Nap	N	
6-Nov	2.50pm	3.20pm	sit	Y	Activity Time Series
6-Nov	3.20pm	4.00pm	walk	Y	Activity
6-Nov	4.00pm	8.20pm	some movement on campus	Y	Activity Time Series
6-Nov	8.20pm	9.10pm	walk	Y	Activity
6-Nov	9.10pm	11.59pm	some movement inside house	Y	Activity Time Series
7-Nov	12.00am	12.30am	some movement inside house	Y	Activity Time Series
7-Nov	12.10am	8.30am	sleep	Y	Sleep Log
7-Nov	8.30am	2.30pm	Some movement inside house	Y	Activity Time Series
7-Nov	2.30pm	3.00pm	Nap	N	
7-Nov	3.00pm	4.00pm	Some movement inside house	Y	Activity Time Series
7-Nov	4.00pm	4.40pm	Walk	Y	Activity
7-Nov	4.40pm	5.30pm	Some movement	Y	Activity Time Series
7-Nov	5.30pm	6.00pm	Walk	Y	Activity
7-Nov	6pm	11.59pm	Some movement inside house	Y	Activity Time Series
7-Nov	7pm		ECG	Y	ECG
7-Nov	7pm		EDA	N	
8-Nov	12.00am	12.30am	Sit	Y	Activity Time Series

Figure 3.5. Daily Log

Figure 3.6. Daily Log Continued

Date	Time Start	Time End	Activity	Available on API?	Location
8-Nov	12.30am	8.40am	Sleep	Y	Sleep Log
8-Nov	8.40am	4.45pm	Some movement inside house	Y	Activity Time Series
8-Nov	4.31pm		ECG	Y	ECG
8-Nov	4.45pm	5.15pm	Nap	N	
8-Nov	5.15pm	11.59pm	Some movement inside house	Y	Activity Time Series
9-Nov	12.00am	12.30am	Some movement inside house	Y	Activity Time Series
9-Nov	12.30am	7.40am	Sleep	Y	Sleep Log
9-Nov	7.40am	9.15am	Some movement inside house	Y	Activity Time Series
9-Nov	9.15am	9.30am	Walk	Y	Activity
9-Nov	9.30am	11.30am	Some movement inside classroom	Y	Activity Time Series
9-Nov	11.30am	11.50am	Walk	Y	Activity
9-Nov	11.50am	2.45pm	Some movement inside house	Y	Activity Time Series
9-Nov	2.45pm	3.20pm	Nap	N	
9-Nov	3.30pm	11.00pm	Some movement inside house	Y	Activity Time Series
9-Nov	11.00pm	11.59pm	Sleep	Y	Sleep Log
10-Nov	12.00am	5.45am	Sleep	Y	Sleep Log
10-Nov	5.45am	7.15am	Some movement inside house	Y	Activity Time Series
10-Nov	7.15am	7.30am	Walk + Veo ride	Y	Activity Time Series
10-Nov	7.30am	11.15am	Some movement inside classroom	Y	Activity Time Series
10-Nov	11.15am	11.35am	Walk	Y	Activity
10-Nov	11.35am	2.15pm	Some movement inside house	Y	Activity Time Series
10-Nov	2.15pm	2.55pm	Nap	Y	Activity
10-Nov	2.55pm	3.10pm	Some movement inside house	Y	Activity Time Series
10-Nov	3.10pm	3.30pm	Walk	Y	Activity
10-Nov	3.30pm	5.30pm	Some movement inside classroom	Y	Activity Time Series
10-Nov	5.10pm		ECG	Y	ECG
10-Nov	5.11pm		EDA	N	
10-Nov	5.30pm	5.50pm	Walk	Y	Activity
10-Nov	5.50pm	11.30pm	Some movement inside house	Y	Activity Time Series
10-Nov	11.30am	11.59pm	Sleep	Y	Sleep Log
11-Nov	12.00am	8.30am	Sleep	Y	Sleep Log
11-Nov	8.30am	11.59pm	Some movement inside house	Y	Activity Time Series
12-Nov	12.00am	12.20am	Some movement inside house	Y	Activity Time Series
12-Nov	12.15am		ECG	Y	ECG
12-Nov	9.30am	11.59pm	Some movement inside house	Y	Activity Time Series
13-Nov	12.00am	12.30am	Some movement inside house	Y	Activity Time Series
13-Nov	12.30am	8.30am	Sleep	Y	Sleep Log
13-Nov	12.30am	9.30am	Sleep	Y	Sleep Log
13-Nov	9.30am	11.59pm	Some movement inside house	Y	Activity Time Series
13-Nov	5.30pm		ECG	Y	ECG

3.3 Data verification by comparison with personal log

Before moving on to the process of building the POC application, the accuracy of the data collected had to be verified. This was done with the help of the Postman tool[ ] that can make API calls to the developer endpoints. Postman was used to make the API calls to the Fitbit developer endpoints. Each entry in the log was searched in different API call results. The match results can be seen in columns 5 and 6 in the figures and . The 'Location' column tells which API call gave the result that matched that specific log.

For Activity-related data, it seems like the tracker didn't record activities' correctly when the time zone changed from ET to CT and vice-versa on November 3 and November 4, respectively. It also seems that the tracker does not record naps/smaller sleeps. On the days where there are records for both night sleep and day nap, it only shows the longer one. The night sleep also has a tag called "Is main sleep?". Other than these two mises, the tracker there is a match for every log.

3.4 Building the Application

Knowing that the data was being collected properly by the device, the focus was shifted to building the tool. The tool is a web application that uses the Fitbit developer API to retrieve the collected data from the Fitbit cloud and display it using an intuitive interface for law enforcement to use. The application design has many components, such as the Back End, the Front End, and the Database(potentially). Figure gives an overview of the application design.

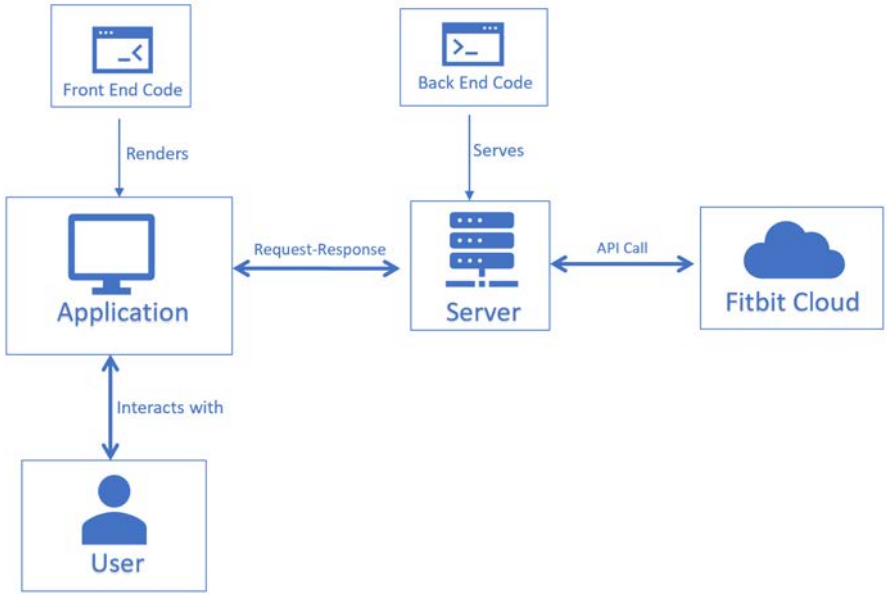


Figure 3.7. Application Design

3.4.1 Back End Components

Back end/server-side code is the code that runs on the server, receives requests from the application front end/client side, and provides the necessary data. It contains the logic to acquire the data from either the database or from a third-party service (in our case, Fitbit cloud). It acts as the layer connecting the front-end Graphical User Interface with the database/cloud that holds the data. There are various technologies that are used to develop back-end services. The POC application uses the Python Django framework because of its ease of usage, speed of processing, and maintenance support.

Figure shows the back-end file structure.

Described below are step-by-step explanations of creating the back-end application.

Authorization Process

The first step is to get the user authorization set up. The authorization process in this application uses the OAuth 2.0 protocol, which is a standard protocol for authorization. Here's a step-by-step explanation:

1. Define Client ID and Secret: The client ID and secret for the registered Fitbit application are defined. These are used to authenticate the application with Fitbit's API. This is available on Fitbit's developer portal

2. Define Authorization and Token URLs: The URLs for Fitbit's authorization endpoint and token endpoint are defined. The authorization endpoint is used to redirect the user to Fitbit's website for authorization, and the token endpoint is used to exchange an authorization code for an access token. authorization URL = 'https://www.fitbit.com/oauth2/authorize' token url = 'https://api.fitbit.com/oauth2/token'.

3. Create OAuth2 Session: An instance of OAuth2Session is created with the client ID and a list of scopes. The scopes represent the types of data that your application wants to access. For this application, access to all the possible data is ideal, so all the scope variables are set. Scope=[activity,heart rate,location,nutrition,oxygen_sat-

uration, profile,respiratory_rate,settings,sleep, social,weight, temperature,cardio_fitness,electrocardiogram].

To get the authorization for a user, the authorization URL should be clicked on the phone having the Fitbit account or opened in a browser while logged into the user's Fitbit account in the same browser. Currently, since the website is not hosted anywhere on a public server, the redirect URL provided to Fitbit is "www.purdue.edu", which is not owned by the researcher. Thus the entire redirect URL needs to be manually pasted back in the application. But in the future, when the application can be hosted on a public server, the redirect URL could be set to a page on the same application, and the process of getting the URL back to the application could also be automated.

Making API request calls to Fitbit Cloud

The API requests to Fitbit cloud API endpoints are sent from different functions inside the "myapp.views.py" file. There is a view function created there for each endpoint that is being called. The data from the endpoints are returned in the form of JSON. This data can now be sent to the front end for display in the necessary format.
A sample snippet for the view related to the activity log is shown in figure

```
def activity_log(request, activity_type=None, date=None):
    print("request: ", request)
    auth_header = request.META.get('HTTP_AUTHORIZATION')
    # print("auth_header: ", auth_header)
    if auth_header is not None:
        access_token = auth_header.split(' ')[1]  # The access token is after the 'Bearer ' part of the header
    else:
        access_token = None
    # The headers for the API request
    headers = {
        'Authorization': f'Bearer {access_token}',
    }
    if request.path == '/api/activity/log/':
        api_url = 'https://api.fitbit.com/1/user/-/activities/list.json?afterDate=2019-01-01&sort=asc&offset=0&limit=100'
    else:
        return JsonResponse({'error': 'Invalid path'}, status=400)
    activity = requests.get(api_url, headers=headers)
    # print("activity: ", activity.json())
    return JsonResponse(activity.json())
```

Figure 3.8. Sample View Function

Map between the front end functions and API calls

For the server to know which API call to make when a front-end webpage asks for a specific piece of information, there should be a map between the functions making the API call and the page displaying the data. This mapping is done in the 'myapp.urls.py' file. This file holds a list of paths for the server, and it maps each path to the respective function inside myapp.views.py.

An example snippet of the URL related to the activity log function is shown in figure

```
urlpatterns = [
    path('api/activity/log/', views.activity_log, name='activity_log')]
```

Figure 3.9. Sample URL Map

3.4.2 Front End Components

The "front end" of an application refers to the user interface (UI) and user experience (UX) components that users interact with directly. It encompasses everything that a user experiences visually and interactively on a website, software, or any digital product. Some of the widely used front-end technologies are HTML-CSS-JS, Jquery, Angular, and React.

The POC application uses HTML, CSS, and JQuery as its front-end technology stack to present a simple and easy-to-use tool.

HTML for GUI layout

The GUI components for all the pages of the website are located in HTML files. The landing page for the site is 'home.html', which is present in the top directory under html inside the project 'FitbitCloudApplicationFrontEnd'. The other pages related to each feature that Fitbit offers are present inside folders of the respective features. The homepage is comprised of a menu to navigate between different types of data - activity, body, breathing, cardio, devices, ECG, friends, heart rate, nutrition, oxygen, sleep, and temperature. The

data for each of these is displayed in a searchable tabular view; there is a search option to search in a specific field and date-related filters that can be applied to the data.

CSS for GUI beautification

The beautification of the web pages is done in the CSS component. The folder structure for CSS follows the same convention and name as its HTML counterpart. Each HTML page has a CSS page linked to it to provide all the display-related settings for the page elements.

JQuery for GUI functionality

The logic to handle different features and functionalities of the web application is present in the javascript files. The application uses JQuery, which is a javascript library, to perform the required actions. The JS files also follow the same folder structure and naming convention as their HTML counterparts. The application uses the JQuery data tables plugin [ ] to directly convert the JSON data sent from serverside to display in the form of tables.

3.5 Evaluation of the Application

3.5.1 Evaluation Criteria

Since the intended purpose of the tool is to aid in crime investigations, the tool is being evaluated for forensic soundness using McKemish's four criteria:

1. Meaning - The tool is checked for accurate representation of the data, without any harm done to its meaning, by comparing the data displayed on the tool with the data obtained by directly calling the API endpoint using Postman.

2. Error - Unit testing is performed on the code, and manual testing on the website. Any errors or exceptions are documented.

3. Transparency - The tool is checked for easy reproducibility and unexpected/hidden behaviors.

4. Experience - The tool is developed with a focus on experience. It is automated where possible.

3.6 Other Application Components

3.6.1 Database to save the data

Database or a cache to hold the retrieved data. This layer is currently not present in the POC application. It is a plan for future work for the application. Some examples of this technology include SQL servers, Redis, MongoDB, etc. This component is present to improve the performance, decrease the wait time for data that can be cached, and ensure smooth use of the application.

The tables and attributes in the database will match the tables and attributes on the Fitbit cloud. Additional tables may be created if needed to handle additional data or business logic.

3.6.2 Server to host the application

Cloud server to host the website. Some popular providers are Azure, AWS, and Google Cloud. The application will be hosted on one of these three. The application is currently hosted on localhost. But, one of these cloud servers will be used to make this application available to the public in the future.

3.7 Data Comparison Process

One of the aims of this study is to understand if Fitbit makes more information available through the developer API compared to the official Fitbit web application. To understand this, a comparison is performed between the two as seen in Table 4.1 and in Figures to present in Appendix A.2. The data from the API is listed in the format that it's available. The equivalent data is searched in the web application. In the case that it is found, it's entered similarly to the API data. In the scenario that it is not found, the row is left empty. Sometimes the data in API is in numerical form, but on the website, a graph is

plotted using those numbers. In those cases, it is entered as a "Graph" in a new row under the "Fitbit Web/App" column.

4. RESULTS

This section discusses the tool execution result and results of the tool evaluation, and also the result of the comparison of data obtained from Fitbit API and Fitbit Web. For the data obtained by using the tool to be of forensic use, it's important to understand and validate the data.

4.1 Tool Execution

The tool can be used in two ways. One is by hosting the application on a server and using it as is. Since the permission granted by Fitbit for this POC application is for research, only 5 user's data can be viewed in this method. The other method is to register a new application as "personal" on the Fitbit developer portal using the Fitbit user's login credentials to get unlimited access to their data.

4.1.1 Step 1: Start the Backened

For the tool to be able to fetch the data from a user's Fitbit cloud, the backend engine needs to start, which is done by running a simple command inside the project's /Fitbit-CloudApplication/FitbitCloud ApplicationBackend directory:
"python manage.py runserver 8080"

4.1.2 Step 2: User Authorization

The tool needs to be authorized to get the user's data. This is done by visiting the URL: "https://www.fitbit.com/oauth2/authorize?response_type=code&client_id=23RKLK&scope=activity+heartrate+location+nutrition+oxygen_saturation+profile+respiratory_rate+settings+sleep+social+weight+temperature+cardio_fitness+electrocardiogram&state=MW3UDeqJATR162QB6ePsgvya7SDT3u" while logged into the user's Fitbit account. Note that the value "23RKLK" This then redirects to Purdue.edu's home page. It can be seen from Figure that the URL contains "?code=[CODE]" after the original www.purdue.edu part. This is the authorization code that the application requires. The entire URL that's

Figure 4.1. Redirect Page

seen in the URL bar should be pasted back into the terminal where the Python server is running, as can be seen from Figure

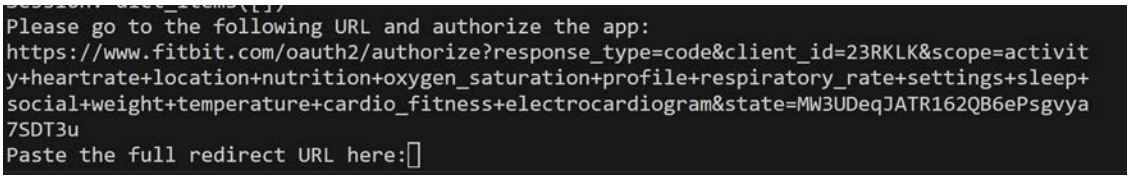


Figure 4.2. Paste URL

Once the URL is given to the application, the application handles the rest of the authorization steps.

4.1.3 Step 3: Landing on Home Page

The home page or landing page, as can be seen in Figure , displays all the possible types of data that can be viewed in the application. This can be accessed by opening the Home.html file that is present in the /FitbitApplicationFrontend directory.

4.1.4 Step 4: Navigating through the Fitbit data categories

There may be multiple kinds of information under a category. One can click on the category name to see the options and decide on what type of data to view. Figure shows the expanded menu for Activity Category. Users can click on one of the options displayed to view the corresponding data.

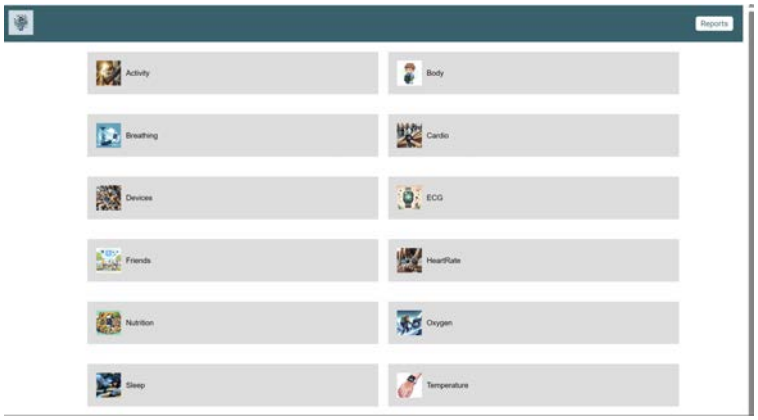


Figure 4.3. Navigation Page

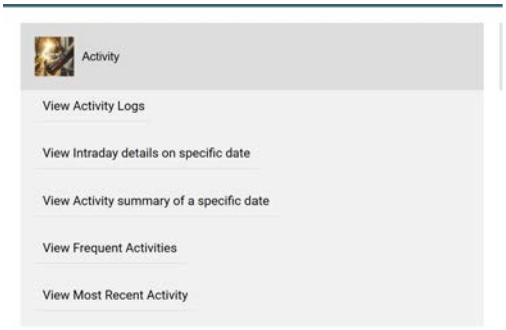


Figure 4.4. Activity Menu

4.1.5 Step 5: (Optional) Selecting the Date for the information request

There are some data types that need a specific date or the upper and lower end of a date range. For those, a date picker panel appears where users can pick their dates and click confirm to view the respective data. For example, activity summary data is fetched for a specific date. As can be seen from Figure , the date picker has options to navigate to a specific date. In addition, it has close and confirm buttons. The specific date or specific start and end dates are locked in when the user hits the confirm button. Until then, the user can change their selection or selections. In the case where two dates are to be selected, and the user clicks on more than two dates, only the last two selections are considered.

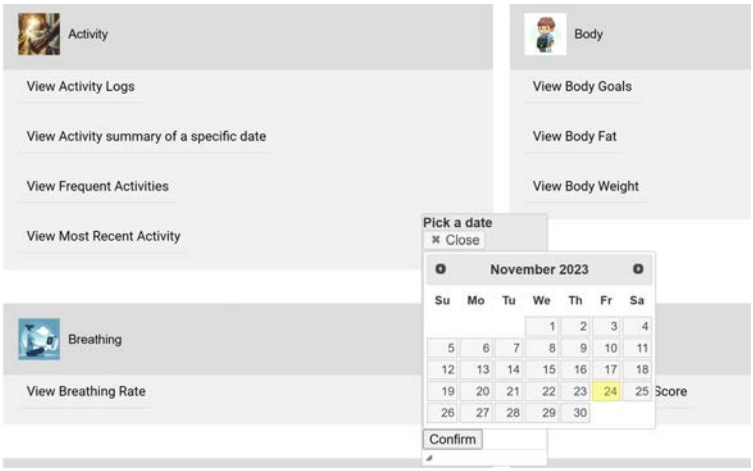


Figure 4.5. Date Selection

4.1.6 Step 6: Viewing the information retrieved

When a user selects the type of data to view on the navigation page, the application fetches the respective data from the Fitbit cloud and displays it in the form of tables. Each column header in the table is sortable. Additionally, there is also a search bar where a user can search for a specific entry in the table. For example, if an investigator wants to get all the activity information for a specific date, they can use the search bar to look for that specific date. As long as it's in the table, the table will be searched, and the related row/rows will be displayed. Figure shows the information page for 'Activity Log'

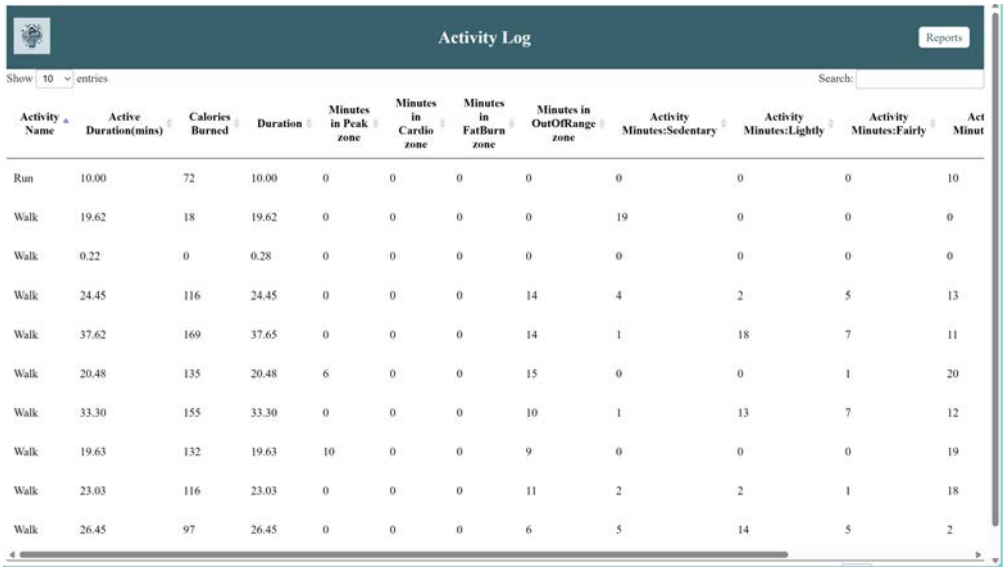
Figure shows the same activity log page when it's filtered on the keyword "run".

4.1.7 Step 7: (Optional) Exporting the data in desired format

The tool also contains the feature to export the data by copying it, printing it, or downloading it as CSV/Excel/PDF, as can be seen in the Figure

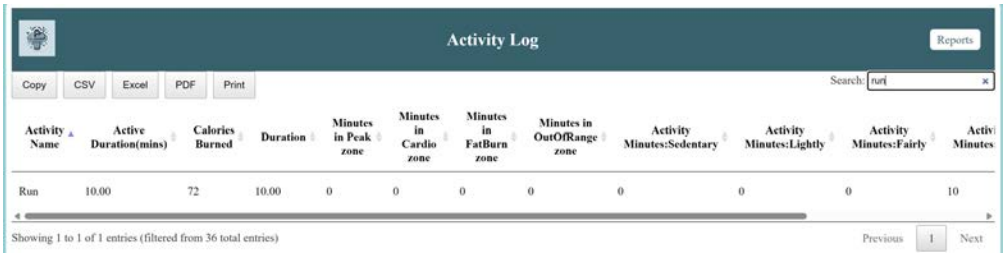
4.2 Tool Evaluation Results

The tool is being evaluated for forensic soundness using McKemish's four criteria



Activity Name	Active Duration(mins)	Calories Burned	Duration	Minutes in Peak zone	Minutes in Cardio zone	Minutes in FatBurn zone	Minutes in OutOfRange zone	Activity Minutes:Sedentary	Activity Minutes:Lightly	Activity Minutes:Fairly	Act Minut
Run	10.00	72	10.00	0	0	0	0	0	0	0	10
Walk	19.62	18	19.62	0	0	0	0	19	0	0	0
Walk	0.22	0	0.28	0	0	0	0	0	0	0	0
Walk	24.45	116	24.45	0	0	0	14	4	2	5	13
Walk	37.62	169	37.65	0	0	0	14	1	18	7	11
Walk	20.48	135	20.48	6	0	0	15	0	0	1	20
Walk	33.30	155	33.30	0	0	0	10	1	13	7	12
Walk	19.63	132	19.63	10	0	0	9	0	0	0	19
Walk	23.03	116	23.03	0	0	0	11	2	2	1	18
Walk	26.45	97	26.45	0	0	0	6	5	14	5	2

Figure 4.6. Information Pages



Activity Name	Active Duration(mins)	Calories Burned	Duration	Minutes in Peak zone	Minutes in Cardio zone	Minutes in FatBurn zone	Minutes in OutOfRange zone	Activity Minutes:Sedentary	Activity Minutes:Lightly	Activity Minutes:Fairly	Activ Minutes
Run	10.00	72	10.00	0	0	0	0	0	0	0	10

Figure 4.7. Search Filter

Figure 4.8. Export Options

1. Meaning - The meaning of the data cannot be manipulated or changed in any way since the application only makes use of the "GET" API calls that are meant to retrieve data. A comparison is done between the data retrieved directly from the API

through Postman and the data shown in the application. The Figures and show the user's activity summary on November 04, 2023, in the app and in Postman, respectively. It can be seen that the values are the same in both.

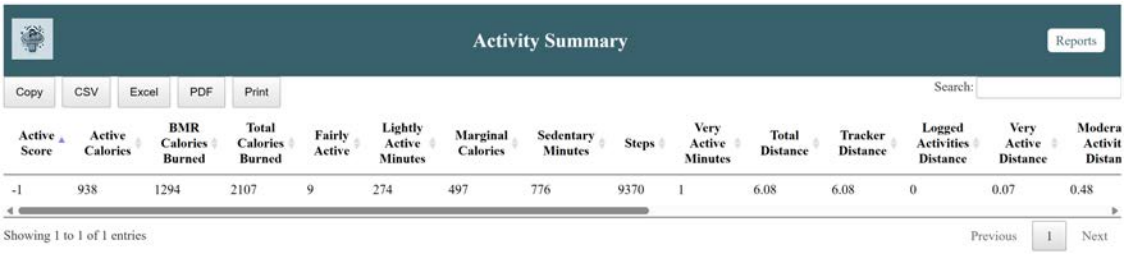


Figure 4.9. Activity Summary in App


```
{
    "activities": [
        {
            "activityId": 90013,
            "activityParentId": 90013,
            "activityParentName": "Walk",
            "calories": 76,
            "description": "Walking less than 2 mph, strolling very slowly",
            "duration": 1125000,
            "hasActiveZoneMinutes": true,
            "hasStartTime": true,
            "isFavorite": false,
            "lastModified": "2023-11-05T02:35:28.000Z",
            "logId": 59669817010,
            "name": "Walk",
            "startDate": "2023-11-04",
            "startTime": "22:05",
            "steps": 1149
        }
    ],
    "summary": {
        "activeScore": -1,
        "activityCalories": 938,
        "caloriesBMR": 1294,
        "caloriesOut": 2107,
        "distances": [
            {
                "activity": "total",
                "distance": 6.08
            ,
```


Figure 4.10. Activity Summary in Postman

2. Error - The tool is well tested, and any exceptions that happen in the tool are well documented and displayed in logs.

   Unit tests have been performed on the tool to get a 100% success on all the views, as can be seen in the Figure . Any other exception that the application throws is logged in the terminal (for serverside) and browser console (for clientside).

Figure 4.11. Unit Test

3. Transparency - The application is open source and documented clearly so as to make it easy to reproduce any step if necessary. The entire codebase for the application can be accessed on Github[ ].

4. Experience - The experience of the investigator cannot be influenced by the tool. However, the tool is simple, intuitive, and automated where possible, making it easy to use for investigators of all experience levels. The technologies for the front end are chosen to optimize the load latency and fast rendering. The information tables are paginated with a page navigation panel at the bottom.

Thus, the tool tests positive for all McKemish's test criteria.

4.3 Data Comparison Results

To verify the hypothesis H_1 of the research that states that "Fitbit collects and stores more information on its cloud than it displays on its web dashboard", the data obtained through the API is listed and compared with the data available on the Fitbit web dashboard as can be seen in table 4.1 below. The complete comparison with all the data is available in Appendix A.2. A plain count comparison shows that there are 304 data headings that can be received from API and 84 from the Fitbit Web App. But this is misleading as there are many headings that are merely present because the data is coming from database tables that need to store additional primary key/ID information. So, removing headings that do not add any value or meaning from API data, there are 280 data headings. That's 3.3 times more data than is displayed on the web application.

Hypothesis H_2 that states that the API can be used to extract all the data on a user from the Fitbit cloud can also be verified with the same comparison. The comparison also shows that there is one particular piece of information that's available on the web application that is not available through the API - the nutrition breakdown information. That is, Fat, Fiber, Carbs, Protein, and Sodium breakdown in food logs.

Table 4.1. Comparision b/w Fitbit API data and Fitbit Web Dashboard data

	Fitbit API	**Fitbit Web Dashboard**
1	food : accessLevel	
2	food : brand	Brand
3	food : calories	Calories
4	food : defaultServingSize	food : defaultServingSize
5	food : defaultUnit : name	food : defaultUnit : name
6	food : defaultUnit : plural	
7	food : isGeneric	
8	food : locale	
9	food : name	
10	food : servings : multiplier	food : servings : multiplier
11	food : servings : servingSize	food : servings : servingSize
12	food : servings : unit : name	food : servings : unit : name
13	food : servings : unit : plural	
14	food : units	
15	goals : calories	
16	foods : isFavorite	
17	foods : logDate	
18	foods : logId	
19	foods : loggedFood : accessLevel	
20	foods : loggedFood : amount	foods : loggedFood : amount

5. DISCUSSION

The comparison of the Fitbit web dashboard data and API data indicated that Fitbit has 3.3 times more information available on the cloud than what's accessible through the official Fitbit web dashboard. In addition to this, the Fitbit developer documentation [ ] allows developers to use many more data types such as Accelerometer, Appbit, Barometer, Body-Pressure, CBOR, Clock, Geolocation, and gyroscope-related data through an Android application. However, it's only possible to access them by creating a mobile application, which is out of the scope of the current research. Thus, we accept the first Hypothesis H_1.

The comparison analysis indicated that there is some data available on the cloud that is not accessible through the API, namely, the nutrition breakdown of food. The study by Hantke et al [ ] also mentions an API call, i.e., https://api.fitbit.com/1 /user/[user-id]/sed/date/[date].json, which isn't listed in the official Fitbit Web API documentation [ ]. There might be more calls like that which are unlisted in the documentation. As mentioned above, the developer documentation also gives a list of data that can be used through an Android application, which indicates that more data can't be accessed through Web API alone and requires other methods to access. Thus, the second Hypothesis, H_2, that it is possible to extract all the information that Fitbit has on a person on the cloud directly through the API, is rejected. However, it is evident that most of the data that are of significance from a forensic standpoint, especially data related to activities, heart rate, breathing rate, and food eaten, can be extracted with this approach. Furthermore, the tool developed provides an excellent interface to view this data and download it in formats like Excel, CSV, and PDF. Being able to download the data in these formats makes it easier to perform any further analysis on it. The possibility to filter on a specific piece of information or get the result on a particular date saves the time spent sifting through a large amount of information. Thus, even if all the data aren't retrievable, the retrieved and displayed data could potentially be of great value to an investigation.

The results obtained successfully validate that extracting data from Fitbit Cloud can be automated through a tool. With the aid of the tool, an investigator does not need to manually perform the steps to authenticate and write API calls to get the d ta. Additionally,

they do not need to manually go through hundreds to thousands of lines of JSON data. All the data is presented in an easy-to-read and search tabular format. Thus the hypothesis H_3 is accepted.

6. CONCLUSION

In recent years, there has been much research with a focus on fitness trackers. Many of them are in the field of Digital Forensics due to the increase in criminal cases involving a smartwatch or a tracker device. However, most studies here focus on retrieving data from physical devices, not the cloud. Cloud Forensics is a budding field that has significant research gaps. Thus, the motivation for this study.

The aims of this study were to 1)propose a proof-of-concept application to automate the process of retrieving user information stored on Fitbit cloud and 2)to understand if the amount of information here differs from the amount of information that is presented to the user in their dashboards. To the author's best knowledge, the methodology used here has not been used before. The methods presented here could also be used for other platforms and devices that store information on the cloud, provided they have developer API endpoints.

There are some limitations to this study. As discussed before, the tool can only be used by the Fitbit user or the authorities with proper permissions because of the authentication process. The data from API and data from the Fitbit Application are compared manually. There might be a more comprehensive way of extracting the Fitbit Web labels using technologies such as web scraping.

The data obtained by the application is trusted to have preserved its integrity and can be used by law enforcement, investigators, and digital forensic researchers for analysis, validation, and any other use. It can effectively reduce the manual effort of going through a large amount of information. Thus, researchers can use the tool to get data from the Fitbit cloud in an easy-to-read format and download them in a desired format to perform analysis. Digital Forensic investigators can use the tool to go through Fitbit data on a person of interest and look up specific information using dates and keywords. Since the tool is open source, other researchers can similarly develop a platform for different devices with developer APIs, such as the Aura ring, Garmin smartwatch, trackers, etc, by making minimal modifications.

7. FUTURE WORK

7.1 Current study

7.1.1 Reports component in the application

The application is proposed to have analytical reports. Figure illustrates the report page's appearance. The page provides options to select the kind of information summarized (For example - Activity summary). Reports regarding the chosen topic are generated using the most up-to-date data available in the Fitbit cloud database.

Figure 7.1. Report

7.1.2 Automation of the user authorization step

Currently, the data authorization steps are printed in the server console. This process can be made more accessible by having a webpage to intuitively guide the user through the steps.

7.2 Other studies

7.2.1 Using other Fitbit APIs

The current study made use of the Fitbit developer Web API to understand what data can be retrieved from the Fitbit cloud. Similarly, Fitbit also provides other APIs for developers, i.e., Device API, Companion API, and Settings API. These APIs can only be accessed by applications that run inside Fitbit. While developing an Android application for the Fitbit tracker was out of scope for this study, it could provide more insight into the data stored on the Fitbit cloud.

REFERENCES

[1] F. Laricchia, "Number of connected wearable devices worldwide from 2019 to 2022," *Statistica.com*, May 15, 2023. [Online]. Available:
.

[2] G. V. Research, "Wearable technology market size, share trends analysis report by product (head eyewear, wristwear), by application (consumer electronics, healthcare), by region (asia pacific, europe), and segment forecasts, 2023 - 2030," *Grandviewresearch.com*, 2023. [Online]. Available:
.

[3] S. R. Department, "Global wearable device data traffic from 2015 to 2020," *Statistica.com*, Feb. 3, 2016. [Online]. Available:
.

[4] J. N. Gilmore and C. Gruber, "Wearable witnesses: Deathlogging and framing wearable technology data in fitbit murders," *Mobile Media & Communication*, vol. 0, no. 0, p. 20 501 579 231 208 139, 0. DOI: . eprint:
. [Online]. Available:
.

[5] BBC, "Apple health data used in murder trial," *bbc.com*, Jan. 12, 2018. [Online]. Available: .

[6] C. Hauser, "Police use fitbit data to charge 90-year-old man in stepdaughters killing," *The New York Times*, Oct. 3, 2018. [Online]. Available:
.

[7] P. Olson, "Fitbit data now being used in the courtroom," *Forbes*, Nov. 16, 2014. [Online]. Available:
.

[8] L. Gardner, "Fitness tracker data used in court cases," *news4jax.com*, Feb. 22, 2022. [Online]. Available:
.

[9] A. Dini Kounoudes, G. M. Kapitsaki, and I. Katakis, "Enhancing user awareness on inferences obtained from fitness trackers data," *User Modeling and User-Adapted Interaction*, pp. 1–48, 2023.

[10] Fitbit, "Fitbit privacy policy," *Fitbit.com*, [Online]. Available: .

[11] G. M. Balbim, I. G. Marques, D. X. Marquez, *et al.*, "Using fitbit as an mhealth intervention tool to promote physical activity: Potential challenges and solutions," *JMIR mHealth and uHealth*, vol. 9, no. 3, e25289, 2021.

[12] A. Salih, "Exploring fitbit smartwatches to detect sleep related disorders," M.S. thesis, 2021.

[13] L. von Niederhäusern and J. Suter, "Fitness data platform," Ph.D. dissertation, OST Ostschweizer Fachhochschule, 2023.

[14] S. Hutchinson, M. M. Mirza, N. West, *et al.*, "Investigating wearable fitness applications: Data privacy and digital forensics analysis on android," *Applied Sciences*, vol. 12, no. 19, p. 9747, 2022.

[15] S. Mcnary and A. Hunter, "Wearable device data for criminal investigation," *Security, Privacy, and Anonymity in Computation, Communication, and Storage: 11th International Conference and Satellite Workshops, SpaCCS 2018, Melbourne, NSW, Australia, December 11-13, 2018, Proceedings 11*, pp. 60–71, 2018.

[16] F. Hantke and A. Dewald, "How can data from fitness trackers be obtained and analyzed with a forensic approach?" In *2020 IEEE European Symposium on Security and Privacy Workshops (EuroSPW)*, 2020, pp. 500–508. DOI: .

[17] A. Watson, "How wearables are being used to solve homicides, missing person and illicit drug cases," Jan. 9, 2019. [Online]. Available: .

[18] E. A. Vogels, "About one-in-five americans use a smartwatch or fitness tracker," Pew Research Center, Tech. Rep., 2020.

[19] D. Jacqualine, "Nearly 70% of americans would wear a fitness trackersmartwatch for discounted health insurance," *Value Pengiun*, 2022. [Online]. Available: .

[20] N. Chauriye, "Wearable devices as admissible evidence: Technology is killing our opportunities to lie," *Cath. UJL & Tech*, vol. 24, p. 495, 2015.

[21] C. Hauser, "In connecticut murder case, a fitbit is a silent witness," *The New York Times*, 2017. [Online]. Available:
.

[22] L. Dawson and A. Akinbi, "Challenges and opportunities for wearable iot forensics: Tomtom spark 3 as a case study," *Forensic Science International: Reports*, vol. 3, p. 100 198, 2021.

[23] K. Pickles, "Police claim woman lied about being raped after her fitbit fitness watch showed she had not been dragged from her bed," *Dailymail*, 2015. [Online]. Available:
.

[24] F. de Arriba-Pérez, M. Caeiro-Rodríguez, and J. M. Santos-Gago, "Collection and processing of data from wrist wearable devices in heterogeneous and multiple-user scenarios," *Sensors*, vol. 16, no. 9, p. 1538, 2016.

[25] M. Biehl, *API Architecture* (API-University Series). CreateSpace Independent Publishing Platform, 2015, ISBN: 9781508676645. [Online]. Available:
.

[26] A. Neumann, N. Laranjeiro, and J. Bernardino, "An analysis of public rest web service apis," *IEEE Transactions on Services Computing*, vol. 14, no. 4, pp. 957–970, 2018.

[27] F. C. support, "How long are activities stored on the dashboard," [Online]. Available:
.

[28] Fitbit, "What should i know about health metrics in the fitbit app?what does each fitbit health metric mean?," [Online]. Available:
.

[29] Y. H. Yoon and U. Karabiyik, "Forensic analysis of fitbit versa 2 data on android," *Electronics*, vol. 9, no. 9, p. 1431, 2020.

[30] J. M. do Valle, G. Souza, N. Cacho, *et al.*, "Using traces from iot devices to solve criminal cases," in *2020 IEEE 6th World Forum on Internet of Things (WF-IoT)*, 2020, pp. 1–6. DOI: .

[31] R. McKemmish, "When is digital evidence forensically sound?," vol. 285, Jan. 2008, ISBN: 978-0-387-84926-3. DOI: .

[32] A. Almogbil, A. Alghofaili, C. Deane, T. Leschke, A. Almogbil, and A. Alghofaili, "Digital forensic analysis of fitbit wearable technology: An investigators guide," in *2020 7th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2020 6th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)*, 2020, pp. 44–49. DOI: .

[33] Á. MacDermott, S. Lea, F. Iqbal, I. Idowu, and B. Shah, *2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*. IEEE, 2019, pp. 1–6.

[34] I. A. Bozdog, T. Daniel-Nicusor, M. Antal, *et al.*, "Human behavior and anomaly detection using machine learning and wearable sensors," in *2021 IEEE 17th International Conference on Intelligent Computer Communication and Processing (ICCP)*, 2021, pp. 383–390. DOI: .

[35] D. Hardt, *The oauth 2.0 authorization framework*, 2012. [Online]. Available: .

[36] B. Cyr, W. Horn, D. Miao, and M. Specter, "Security analysis of wearable fitness devices (fitbit)," *Massachusetts Institute of Technology*, vol. 1, 2014. [Online]. Available: .

[37] Postman, "Postman tool," [Online]. Available: .

[38] Datatables, "Datatables plugin," [Online]. Available: .

[39] P. Hegde, "Poorvi github," [Online]. Available: .

[40] Fitbit, "Fitbit developer documentations- web api," *Fitbit.com*, [Online]. Available: .

A. APPENDIX

A.1 Application Codebase

The entire codebase for the application developed in this study is available on the author's GitHub repository at https://www.github.com/PoorviHegde.

A.2 The complete data comparison between Fitbit API and Fitbit Dashboard

Figures to show the comparison performed between the data shown through Fitbit Web API and on the Fitbit Dashboard.

	Fitbit API	Fitbit Web Dashboard
1	food : accessLevel	
2	food : brand	Brand
3	food : calories	Calories
4	food : defaultServingSize	food : defaultServingSize
6	food : defaultUnit : name	food : defaultUnit : name
7	food : defaultUnit : plural	
9	food : isGeneric	
10	food : locale	
11	food : name	
12	food : servings : multiplier	food : servings : multiplier
13	food : servings : servingSize	food : servings : servingSize
15	food : servings : unit : name	food : servings : unit : name
16	food : servings : unit : plural	
18	food : units	
19	goals : calories	
20	foods : isFavorite	
21	foods : logDate	
22	foods : logId	
23	foods : loggedFood : accessLevel	
24	foods : loggedFood : amount	foods : loggedFood : amount
25	foods : loggedFood : brand	foods : loggedFood : brand
26	foods : loggedFood : calories	foods : loggedFood : calories
27	foods : loggedFood : foodId	
28	foods : loggedFood : locale	
29	foods : loggedFood : mealTypeId	
30	foods : loggedFood : name	foods : loggedFood : name
32	foods : loggedFood : unit : name	foods : loggedFood : unit : name
33	foods : loggedFood : unit : plural	
34	foods : loggedFood : units	
35	foods : nutritionalValues	foods : nutritionalValues
36	goals : calories	goals : calories
37	summary	
39	name	
40	plural	
41	frequent food	frequent food
42	meal : description	
44	meal : mealFoods : accessLevel	
45	meal : mealFoods : amount	
46	meal : mealFoods : brand	
47	meal : mealFoods : calories	
49	meal : mealFoods : locale	
50	meal : mealFoods : mealTypeId	
51	meal : mealFoods : name	
53	meal : mealFoods : unit : name	
54	meal : mealFoods : unit : plural	
55	meal : mealFoods : units	
56	meal : name	

Figure A.1. Comparision between Fitbit API Data and Fitbit Dashboard Data

Figure A.2. Comparision between Fitbit API Data and Fitbit Dashboard Data

56	meal : name	
57	summary : water	summary : water
58	water : amount	water : amount
60		Fat
61		Fiber
62		Carbs
63		Sodium
64		Protein
65	activities : activeDuration	activities : activeDuration
66	activeZoneMinutes : minutesInHeartRateZones : minuteMultiplier	
67	activeZoneMinutes : minutesInHeartRateZones : minutes	activeZoneMinutes : minutesInHeartRateZones : minutes
68	activeZoneMinutes : minutesInHeartRateZones : order	
69	activeZoneMinutes : minutesInHeartRateZones : type	activeZoneMinutes : minutesInHeartRateZones : type
70	activeZoneMinutes : minutesInHeartRateZones : zoneName	activeZoneMinutes : minutesInHeartRateZones : zoneName
71	activeZoneMinutes : totalMinutes	
72	activities: activityLevel : minutes	
73	activities: activityLevel : name	
74	activities : activityName	activities : activityName
76	activities : averageHeartRate	
77	activities : calories	activities : calories
78	activities : caloriesLink	
79	activities : detailsLink	
80	activities : calories	
81	activities : distanceUnit	activities : distanceUnit
82	activities : duration	activities : duration
83	activities : elevationGain	
84	activities : hasActiveZoneMinutes	
85	activities : heartRateLink	
86	activities : heartRateZones : caloriesOut	
87	activities : heartRateZones : max	
88	activities : heartRateZones : min	
89	activities : heartRateZones : minutes	
90	activities : heartRateZones : name	
91	activities : lastModified	activities : lastModified
93	activities : logType	
94	activities : manualValuesSpecified : calories	

Figure A.3. Comparision between Fitbit API Data and Fitbit Dashboard Data

94	activities : manualValuesSpecified : calories	
95	activities : manualValuesSpecified : distance	
96	activities : manualValuesSpecified : steps	
97	activities : originalDuration	
98	activities : originalStartTime	
99	activities : pace	
101	activities : source : name	
102	activities : source : trackerFeatures	
103	activities : source : type	
104	activities : source : url	
105	activities : speed	
106	activities : startTime	
107	activities : steps	
108	activities : tcxLink	
109	goals : activeMinutes	goals : activeMinutes
110	goals : activeZoneMinutes	goals : activeZoneMinutes
111	goals : caloriesOut	goals : caloriesOut
112	goals : distance	goals : distance
113	goals : floors	goals : floors
114	goals : steps	goals : steps
115	Activity : Sport	
117	Lap : StartTime	
118	Lap : TotalTimeSeconds	
119	Lap : DistanceMeters	
120	Lap : Calories	
121	Lap : Intensity	
122	Lap : TriggerMethod	
123	Track : Trackpoint : Time	
124	Track : Trackpoint : Position : LatitudeDegrees	
125	Track : Trackpoint : pPosition : LongitudeDegrees	
126	Track : Trackpoint : AltitudeMeters	
127	Track : Trackpoint : DistanceMeters	
128	Track : Trackpoint : HeartRateBpm : Value	

Figure A.4. Comparision between Fitbit API Data and Fitbit Dashboard Data

129	activityLog : activityId	
130	activityLog : activityParentId	
131	activityLog : activityParentName	
132	activities : calories	
133	activities : description	activities : description
134	activities : detailsLink	
135	activities : distance	activities : distance
136	activities : duration	activities : duration
137	activities : hasActiveZoneMinutes	
138	activities : hasStartTime	
139	activities : isFavorite	
140	activities : lastModified	
142	activities : name	
143	activities : startDate	activities : startDate
144	activities : startTime	activities : startTime
145	activities : steps	activities : steps
146	goals : activeMinutes	goals : activeMinutes
147	goals : caloriesOut	goals : caloriesOut
148	goals : distance	goals : distance
149	goals : floors	goals : floors
150	goals : steps	goals : steps
151	summary : activeScore	
152	summary : activityCalories	
153	summary : caloriesEstimationMu	
154	summary : caloriesBMR	
155	summary : caloriesOut	
156	summary : caloriesOutUnestimated	
157	summary : distances : activity	
158	summary : distances : distance	
159	summary : elevation	
160	summary : fairlyActiveMinutes	
161	summary : floors	
162	summary : heartRateZones : caloriesOut	
163	summary : heartRateZones : max	
164	summary : heartRateZones : min	
165	summary : heartRateZones : minutes	
166	summary : heartRateZones : name	
167	summary : lightlyActiveMinutes	
168	summary : marginalCalories	
169	summary : restingHeartRate	
170	summary : sedentaryMinutes	
171	summary : steps	
172	summary : useEstimation	
173	summary : veryActiveMinutes	
174	Favorite Activity	
175	frequent Activity	

Figure A.5. Comparision between Fitbit API Data and Fitbit Dashboard Data

175	frequent Activity	
176	best : total : distance : date	best : total : distance : date
177	best : total : distance : value	best : total : distance : value
178	best : total : floors : date	
179	best : total : floors : value	
180	best : total : steps : date	best : total : steps : date
181	best : total : steps : value	best : total : steps : value
182	best : tracker : distance : date	
183	best : tracker : distance : value	
184	best : tracker : floors : date	
185	best : tracker : floors : value	
186	best : tracker : steps : date	
187	best : tracker : steps : value	
188	lifetime : total : activeScore	
189	lifetime : total : caloriesOut	
190	lifetime : total : distance	lifetime : total : distance
191	lifetime : total : floors	
192	lifetime : total : steps	lifetime : total : steps
193	lifetime : tracker : activeScore	
194	lifetime : tracker : caloriesOut	
195	lifetime : tracker : distance	
196	lifetime : tracker : floors	
197	lifetime : tracker : steps	
198	Recent Activity	
199	goal : goalType	Graph
200	goal : startDate	
201	goal : startWeight	goal : startWeight
202	goal : weight	goal : weight
203	goal : weightThreshold	
204	goal : fat	goal : fat
205	fat : date	
206	fat : fat	
207	fat : logId	
208	fat : source	
209	fat : time	
210	weight : bmi	weight : bmi
211	weight : date	weight : date
213	weight : source	
214	weight : time	
215	weight : weight	weight : weight
216		Graph
217	consistency : flowId	
218	goal : minDuration	goal : minDuration
219	goal : updatedOn	goal : updatedOn
220	sleep : dateOfSleep	sleep : dateOfSleep
221	sleep : duration	sleep : duration
222	sleep : efficiency	
223	sleep : endTime	sleep : endTime
224	sleep : infoCode	
225	sleep : isMainSleep	

Figure A.6. Comparision between Fitbit API Data and Fitbit Dashboard Data

226	sleep : levels : data : dateTime	
227	sleep : levels : data : level	
228	sleep : levels : data : seconds	
229	sleep : levels : shortData : dateTime	
230	sleep : levels : shortData : level	
231	sleep : levels : shortData : seconds	
232	sleep : levels : summary : [level] : count	sleep : levels : summary : [level] : count
233	sleep : levels : summary : [level] : minutes	sleep : levels : summary : [level] : minutes
234	sleep : levels : summary : [level] : thirtyDayAvgMinutes	
236	sleep : minutesAfterWakeup	
237	sleep : minutesAsleep	sleep : minutesAsleep
238	sleep : minutesAwake	sleep : minutesAwake
239	sleep : minutesToFallAsleep	
240	sleep : logType	
241	sleep : startTime	sleep : startTime
242	sleep : timeInBed	
243	sleep : type	
244	summary : stages : [level]	
245	summary : totalMinutesAsleep	
246	summary : totalSleepRecords	
247	summary : totalTimeInBed	
248		Graph
249	br : dateTime	
250	br : value : breathingRate	
251		Graph
252	cardioscore : dateTime	
253	cardioscore : value : vo2Max	
254		Graph
255	battery	
256	batteryLevel	
257	deviceVersion	
258	features	
260	lastSyncTime	
261	mac	
262	type	
263	trackerAlarms : deleted	
265	trackerAlarms : enabled	
266	trackerAlarms : recurring	
267	trackerAlarms : snoozeCount	
268	trackerAlarms : snoozeLength	
269	trackerAlarms : syncedToDevice	
270	trackerAlarms : time	
271	trackerAlarms : vibe	
272	trackerAlarms : weekDays	
273	ecgReadings : startTime	Graphical Report

Figure A.7. Comparision between Fitbit API Data and Fitbit Dashboard Data

273	ecgReadings : startTime	Graphical Report
274	ecgReadings : averageHeartRate	
275	ecgReadings : resultClassification	
276	ecgReadings : waveformSamples	
277	ecgReadings : samplingFrequencyHz	
278	ecgReadings : scalingFactor	
279	ecgReadings : numberOfWaveformSamples	
280	ecgReadings : leadNumber	
281	ecgReadings : featureVersion	
282	ecgReadings : deviceName	
283	ecgReadings : firmwareVersion	
290	activities-heart : datetime	activities-heart : datetime
291	activities-heart : value : customHeartRateZone : caloriesOut	
292	activities-heart : value : customHeartRateZone : max	
293	activities-heart : value : customHeartRateZone : min	
294	activities-heart : value : customHeartRateZone : minutes	
295	activities-heart : value : customHeartRateZone : name	
296	activities-heart : value : HeartRateZone : caloriesOut	
297	activities-heart : value : HeartRateZone : max	
298	activities-heart : value : HeartRateZone : min	
299	activities-heart : value : HeartRateZone : minutes	activities-heart : value : HeartRateZone : minutes
300	activities-heart : value : HeartRateZone : name	
301	activities-heart : value : restingHeartRate	
302		Graph
303	hrv : dateTime	
304	hrv : value : dailyRmssd	
305	hrv : value : deepRmssd	
306		Graph
307	dateTime	
308	value : avg	
309	value : min	
310	value : max	Graph
311	tempCore : dateTime	tempCore : dateTime
312	tempCore : value	tempCore : value
313	tempSkin : dateTime	Graph
314	tempSkin : value : nightlyRelative	