Leveraging the FitBit API to Share Activity Levels with a Trusted Caregiver

Kim Chen
Department of Computer Science
Stetson University
kchen@stetson.edu

Dr. Pamela Wisniewski and Dr. Damla Turgut
Department of Computer Science
University of Central Florida
pamela.wisniewski@ucf.edu and *turgut@eecs.ucf.edu*

Introduction

In the context of wearable Internet of Things (IoT) technology, the goal of this research is to better understand:

➢ The types of health data individuals feel comfortable sharing with others

➢ The types of people they feel this information is appropriate to share with

➢ The context in which sharing such sensitive data makes sense

To accomplish this, we apply the theory of contextual integrity and created a caregiving mobile application called "**Carebitter**" that uses the Fitbit API to authenticate a Fitbit user on another mobile users' phone so that he or she would have access to various levels of the Fitbit user's health information.

Related Work

With our research, we propose a noninvasive, unobtrusive health monitoring system using the Fitbit.

Previous use of the Fitbit in research includes a study conducted by Prasad et al, which collects the health data of real users and measures their willingness to share the information with family, friends, third parties and the public [1]. Sizable studies such as this are heavily resource intensive, take a decent amount of time, and are tough to reproduce. Our approach is intended to be just as effective, whilst also being simple and reproducible.

We are also interested in studying the sharing behaviors of individuals when it comes to their health data and whom they are sharing it with, under the specific context of Caregiving. In research conducted by Khorakhun and Bhatti, a similar study was designed using a wellbeing scenario as a proxy of mHealth monitoring [2]. In this study, they created remote monitoring application using an open-source online social media platform alongside a Fitbit to measure the sharing behaviors of individuals over a social media platform. Our approach, however, is intended to be more intimate and personal, as a Fitbit user is going to be sharing information with one Caregiver at a time, as opposed to a social media platform.

References

[1] A. Prasad, J. Sorber, T. Stablein, D. Anthony, and D. Kotz, "Understanding sharing preferences and behavior for mHealth devices," in *Proc. ACM Wrkshp. Privacy in the electronic society*, 2012.

[2] C. Khorakhun, and S. N. Bhatti. "MHealth through Quantified-Self: A User Study". *2015 17th International Conference on E-health Networking, Application & Services (HealthCom)*. Web.

Technical Approach

Fitbit API
Enables third-party applications to access and write data on behalf of users

OAuth 2.0 Implicit Grant Flow
| The application redirect the user to Fitbit's authorization page | Redirects user back to application's redirect URL with access token | The application stores the access token client side |

Callback URL
| Where Fitbit redirects the user after authentication is complete | Includes an access token as a URL fragment |

Access Token
Parse the token and save it client side to use when making requests to the Fitbit API

Making Requests
| Send a request to the Fitbit API with the users' token | Receive a JSON response from the database | Parse the response for the information to be displayed |

```
https://dev.fitbit.com/#access_token=eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI0
TllZWUwiLCJhdWQiOiIyMjdUTVkiLCJpc3MiOiJGaXRiaXQiLCJ0eXAiOiJhY2Nlc3NfdG
9rZW4iLCJzY29wZXMiOiJ3aHIgd251dCB3cHVIHdzbGUgd3dlaSB3c29jIHdzZXQgd2Fj
dCB3bG9jIiwiZXhwIjoxNDY5NzI1ODQ3LCJpYXQiOjE0NjkxMjEwNDg9.Olm5IM38ddKvx
96XWQ8CIjvYxzZ0gz5BCt85Jb7HpKA&user_id=4NYYYL&scope=sleep+settings+nut
rition+activity+social+heartrate+profile+weight+location&token_type=Be
arer&expires_in=604800
```
Example Callback

```
GET https://api.fitbit.com/1/user/-
/activities/heart/date/today/1d/1sec/time/00:0
0/00:01.json Authorization: Bearer
eyJhbGciOiJIUzI1NiJ9.eyJleHAiOjE0MzAz
NDM3MzUsInNjb3BlcyI6Indwcm8gd2xvYyB3bnV0IHdzbG
Ugd3NldCB3aHIgd3dlaSB3YWN0IHdzb2MiLCJzdWIiOiJB
QkNERUYiLCJhdWQiOiJJSktMTU4iLCJpc3MiOiJGaXRiaX
QiLCJ0eXAiOiJhY2Nlc3NfdG9rZW4iLCJpYXQiOjE0MzAz
NDAxMzV9.z0VHrIEzjsBnjiNMBey6ywtu26yHTnSWz_qlqo
EpUlpc
```
Example Request

```java
private void loadHRData() {
    String heartRateURL = this.fullURL("activities/heart/date/today/1d/1min/time/00:00/23:59.json");
    JsonObjectRequest heartRateRequest = new JsonObjectRequest(heartRateURL, null,
        new Response.Listener<JSONObject>() {
            @Override
            public void onResponse(JSONObject response) {
                try {
                    JSONObject heartRateIntradayData = response.getJSONObject("activities-heart-intraday");
                    JSONArray dataset = heartRateIntradayData.getJSONArray("dataset");
                    int lastArrayIndex = dataset.length() - 1;
                    JSONObject currentHeartRate = dataset.getJSONObject(lastArrayIndex);
                    String bpm = currentHeartRate.getString("value");
                    String time = currentHeartRate.getString("time");
                    updateHRData(bpm, time);
                } catch (Exception e) {
                    Toast.makeText(getApplicationContext(), "Could not load Detailed Heart Rate for an unknown reason.",
                        Toast.LENGTH_SHORT).show();
                }
            }
        }, new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error){
                Toast.makeText(getApplicationContext(), "Error retrieving Detailed Heart Rate.", Toast.LENGTH_SHORT).show();
            }
    }) {
        @Override
        public Map<String, String> getHeaders() throws AuthFailureError {
            Map<String, String> headers = new HashMap<>();
            headers.put("Authorization", "Bearer " + accessToken);
            return headers;
        }
    };
    FitbitAPI.getInstance().addToQueue(heartRateRequest);
}
```
Code for making a Request

```json
{
    "activities-heart": [
    ],
    "activities-heart-
intraday": {
        "dataset": [
            {
                "time":
"00:00:00",
                "value": 50
            },
            {
                "time":
"00:01:00",
                "value": 46
            },
            {
                "time":
"00:02:00",
                "value": 47
            }
        ],
        "datasetInterval": 1,
        "datasetType":
"minute"
    }
}
```
Example Response

Interface Design

Title Screen displaying application logo and slogan

Explains to the Fitbit user what the application does, and how they can revoke access whenever they feel necessary

Chrome tab for Fitbit login and authorization

Screen that allows the Fitbit user to select the type of information they want to share

Dashboard which displays the information and shows when it was last updated

Future Work

➢ Create alerts within the application for when either heart rate or steps fall outside of a certain threshold

➢ Conduct an in-depth, user-study to gauge perceptions and privacy behaviors

➢ Suggest new features and pinpoint potential opportunities for interface redesign

Acknowledgements