Varun Chilukuri

Mobile/Web Senior Lab

Period 7 Tra

8 November 2020

**A Second Pair of Eyes: Social Accountability for Fitness Goals**

**Abstract**

*It is common to experience sudden bouts of motivation and set ambitious fitness goals. However, motivation rapidly declines and goals become difficult to pursue after a few days. If you are only accountable to yourself, it is easier to let those commitments slide. It has been proven that notifying others about your goals makes it more likely that you will complete them. This project will help users achieve their fitness goals by holding them accountable through social means via various exercising and messaging platforms.*

**Problem/Purpose/Engineering Goals**

The purpose of this project is to help users achieve their exercise goals by holding them accountable. To do this, the application integrates with Strava and Telegram, which are fitness and messaging platforms. To provide the social accountability aspect, it will check if the user has exercised that day and message a friend if the user has not. Social features that are already in place (like virtual competitions and “kudos” you can earn from your followers) are all positive reinforcement that come *after* you post a workout. They pose no consequence if the user doesn’t exercise, which my project will provide.

**Background:**

With the start of the new year, gyms see increased activity as people around the world become more motivated to improve their fitness and become active in the new year. But by February 1st, activity returns to normal as resolutions fall apart (Bachman, 2015). This happens year after year. Commitments that are undertaken as an individual effort are more likely to fail. However, it has been shown that when other people are holding you accountable, you are more likely to achieve your goals (Stieg, 2019).

After a run or session at the gym, many users post their workout to a platform like Strava. This is a social media that is aimed towards runners/cyclists but can also include other types of activity. Users can participate in virtual competitions, become the leader of local routes, and earn “kudos” when followers like their workout (Features for Athletes, n.d.). A typical post for a GPS-based activity on the platform includes distance, pace, time, elevation, calories, and heart rate data. A typical post for a non-GPS activity includes time, calories, and heart rate data (Features for Athletes, n.d.). This information (and more) can be accessed via the Strava developer API, which is a REST API that also uses OAuth 2.0. A REST (**RE**presentational **S**tate **T**ransfer) API is the industry standard that allows for two computers to communicate over HTTP using a series of requests and responses (Fielding, 2000, p. 3). OAuth 2.0 is an open standard for authorization commonly used in login-flows and single sign-on. It is also used by Okta, a company that offers authorization and verification services through its API (API Overview, n.d.).

To connect their activity with a messaging platform, my project will use the corresponding Telegram API (also a REST API). Since some functionality is missing from the official API, my project will also be using Pyrogram, a Telegram wrapper that is written in Python and C.

Due to the dynamic nature of the site’s content, the website will use a Python-based microframework like Flask. A microframework is a minimal web application framework that does not have the “bells-and-whistles” of a full-stack framework. Specifically, it lacks form validation, a database abstraction layer, upload handling, various open authentication technologies, and more (Burch, 2010, p. 1). A micro-framework does less than a full-stack framework but offers more freedom to the developer. To handle the issue of storing data with Flask, my project will utilize built-in support for sqlite3 that comes with Python (Define and Access, n.d.).

**Research Techniques / Methods:**

There are three major parts to my project that must work together. When the user first begins using the service, they must create an account on my website so that all data is centralized and stored in one location. All data tied to user accounts will be saved in a database using sqlite3. The log-in authentication will be done through Okta since they provide a secure and thorough API specifically for login flow.

Once the user creates their account, they will need to allow access to their Strava account. This will be handled by the Strava API and the user will just need to authorize my application to access their data. When the user clicks the “Authorize” button, they will be redirected to a page on my site with the following format:

http://example.com/?state=&code=[ALPHANUMERIC\_CODE]&scope=read,activity:read\_all

Using Flask, this code will be saved and tied to the user’s account. The code is needed to generate the refresh and access tokens to access the user’s Strava activities in the future. Getting user data and handling the Strava API will be done with Python.

The user will then connect a messaging platform to their account. With Telegram, they must sign in and also provide the phone number of a friend. Since the Telegram API is very limited, the use of a third party API means that the user will have to grant temporary access to their account. Since all logins on Telegram require 2FA, this is guaranteed to be one-time access. Once in the account, the application will create a group chat between my registered Telegram bot, the user, and their friend. This is also done with Python. The Telegram bot will then handle all future work and my application will log out of the account.

After the “setup” process is complete, the program will run at 11 PM and message the user’s friend if there is no activity for that day. After a bare-bones implementation, I plan to improve website design, add graphs/statistics, add other messaging and fitness platforms, and allow for more customization by the user.

**Materials**

* Strava API
* Telegram API (official)
* Pyrogram (Unofficial Telegram API)
* Okta Authentication
* Flask
* Postman
* Personal laptop

**References**

*API Overview*. (n.d.). Okta. Retrieved November 8, 2020, from https://developer.okta.com/docs/reference/api-overview/

Bachman, R. (2015, January 19). *The week your New Year's resolution to exercise dies*. The Wall Street Journal. Retrieved November 8, 2020, from https://www.wsj.com/articles/the-week-your-new-years-resolution-to-exercise-dies-1421703411

Burch, C. (2010). Django, a web framework using Python: tutorial presentation. *Journal of Computing Sciences in Colleges*, *25*(5).

*Define and access the database*. (n.d.). Flask. Retrieved November 8, 2020, from https://flask.palletsprojects.com/en/1.1.x/tutorial/database/

*Features for athletes, made by athletes*. (n.d.). Strava. Retrieved November 8, 2020, from https://www.strava.com/features

Fielding, R. T. (2000). *Architectural styles and the design of network-based software architectures* [Doctoral dissertation, University of California Irvine]. ACM. https://dl.acm.org/doi/book/10.5555/932295

*Foreword*. (n.d.). Flask. Retrieved November 2, 2020, from https://flask.palletsprojects.com/en/1.1.x/foreword/#what-does-micro-mean

*Getting started with the Strava API*. (n.d.). Strava. Retrieved November 2, 2020, from https://developers.strava.com/docs/getting-started/

Stieg, C. (2019, September 5). *How to stay committed to your goals: Tell someone more successful than you, says new study*. CNBC Make It. Retrieved November 8, 2020, from https://www.cnbc.com/2019/09/05/why-sharing-goals-with-someone-helps-you-achieve-them.html

Each Section will be awarded a percentage of points according to the following scale:

* Superior …………………….…………………………………………… 100 %
* Thorough …………………….………………………………………….. 80 %
* Somewhat …………………….………………..……………………..…. 50 %
* Insufficient …………………….…………………..…………………….. 30 %

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Item | Description (these are brief descriptions of what you must address from the above areas, see above for more complete Description.) | Teacher Notes and Scale Factor from Above | Points Possible x Above Scale | Your Points |
| General Format | Professional font, normal margins. Professional spacing | 85% | 10 | 8.5 |
| Abstract | Abstract is concise but not choppy, summarizes the work without adding new information. Can be understood by those with minimal knowledge of the discipline. Structurally distinct from rest of paper | 93% | 10 | 9.3 |
| Problem/Purpose | 3-5 sentence description of goals and purpose | 90% | 10 | 9.0 |
| Background and Introduction | Provides complete explanation of the background to scientific topic. | 97% | 40 | 38.8 |
| Research Techniques / Methods | Reality-based strategy for implementation | 97% | 40 | 38.8 |
| Materials | Can be in list form | 100% | 10 | 10 |
| References | Adhere to proper guidelines | 90% | 5 | 4.5 |
| Total: |  |  | 125 | 118.9 |