Activity Tracker

CS426: Internet of Things Project

Parampreet Parmar
Department of Computer Science
Binghamton University
Binghamton, NY USA
pparmar2@binghamton.edu

ABSTRACT

Internet of things, IoT for short, has become increasingly popular in the past decade. IoT describes the wireless communication between multiple "things" in order to accomplish a common goal. IoT is made up of devices from simple sensors to smartphones. It has incorporated itself into all parts of human's daily lives and made it easier.

Health is the most important part of one's life. A person's health is at the center of their lives. Today's IoT devices has helped people control and track their health. In this project, I created an activity tracking app that tracks a user's health data. This data includes a user's total walking steps and their heart rate. The app uses Google Firebase to store the user data and credentials. The user may view the data in the app and as well as the database.

KEYWORDS

Internet of Things, Step Counter, Android App, Google Firebase, Health, Heart rate

1 Introduction

The U.S adult obesity rate stands at 42.4 percent and has increased by 26 percent since 2008 [1]. For healthiness and wellness, exercising is one of the key factors. Step counting has long been used as a method of measuring distance. It is an essential role in human activity. Current smart phones come with many different sensors. The built-in step counter sensor in smart phones was used in my application to count the user's steps. Monitoring heart rate during exercise is essential to success in meeting a fitness goal. My app will also allow the user to check their heart rate. The heart rate monitor uses the smartphones camera and flashlight to measure the users heart rate. The user will place their index finger on their android phones camera lens, and within a couple of seconds the app will calibrate and begin to measure the heart rate of the user. An Activity Tracker firebase database will be used to store all of the user login credentials and health data.

This project report is organized as follows. In Section 2, we discuss the overall design of the Android application and the details of our implementation. In Section 3, we evaluate the accuracy and performance of the Activity Tracker app. Then finally in section 4, we conclude.

2 System Design and Implementation

I created an android application for this project. Users first need to log in if they have an account. I used Google firebase as the database to store the user's data. First, user logs into the account on the app. After logging in, the user will be prompted to either start tracking their steps or measure their heart rate. There is also a set goal page where a user is able to set a daily step goal. A user can also view their steps record and heart record. Where the previous steps and heart rate is shown.

In this section, I first describe the overall design and implementation of the Activity Tracker app. First, I follow with a control flow diagram describing the control flow of the app. I then follow with the explanation of the implementation of google firebase. Next, I will give a detailed explanation on the design of the android application. Finally followed by the design the heart rate and step counter sensor implementation.

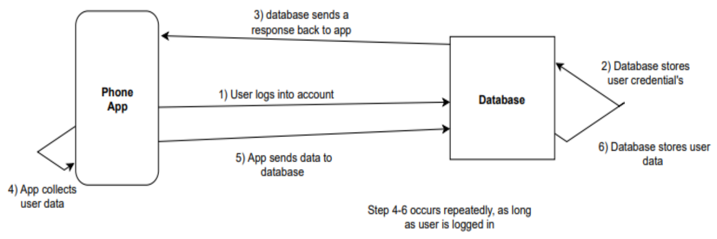

3) database sends a response back to app
Phone App
1) User logs into account
Database
2) Database stores user credential's
6) Database stores user data
5) App sends data to database
4) App collects user data
Step 4-6 occurs repeatedly, as long as user is logged in

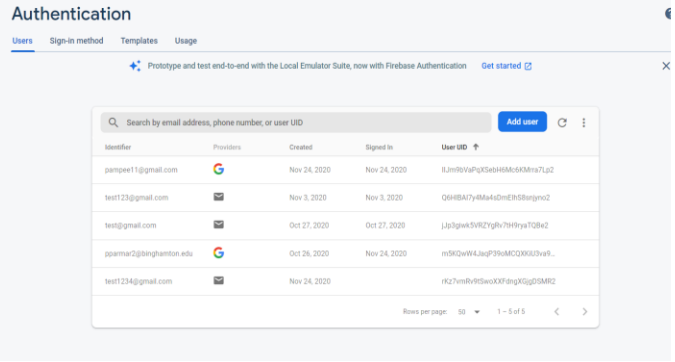
Figure 1: Control Flow diagram of Activity Tracker

2.1 Control Flow Diagram

Figure 1 shows the control-flow diagram of Activity Tracker. The control represents the control flow of the app. First, the user logs into their account on the app, the app sends a response to the activity tracking database. If there is a valid user in database, we move forward to step 2. During step 2, the database stores the user credentials and retries that specific user data and step 3 is the database sending back a response to the app. When the user is logged in, they are able to start tracking their steps or heart rate data. Step 5, the app sends the user data back to the database where it is stored during step 6. Steps 4 to 6 occur repeatedly as long as the user is logged in.

2.2 Google Firebase

To store user credentials and user data, I used Firebase database [2]. The Firebase database is a cloud-hosted NoSQL database that Stores and syncs data between users in real time. The android application has a login page where a user would have to login with an authenticated account on the Activity Tracker firebase. A user who wants to be authenticated will have to be manually added as a user in the Activity Tracker firebase. Figure 2 shows the authentication tab in google firebase. A user may log out on the logout tab in the application, shown in figure 3.



Authentication

Users | Sign-in method | Templates | Usage

Prototype and test end-to-end with the Local Emulator Suite, now with Firebase Authentication Get started

Search by email address, phone number, or user UID | Add user

Identifier	Providers	Created	Signed In	User UID
pampee11@gmail.com	G	Nov 24, 2020	Nov 24, 2020	lIJm9bVaPqXSebH6Mc6KMrra7Lp2
test123@gmail.com	✉	Nov 3, 2020	Nov 3, 2020	Q6HlBAI7y4Ma4sDmElhS8snjyno2
test@gmail.com	✉	Oct 27, 2020	Oct 27, 2020	jJp3giwk5VRZYgRv7tH9ryaTQBe2
pparmar2@binghamton.edu	G	Oct 26, 2020	Nov 24, 2020	m5KQwW4JaqP39oMCQXKiU3va9...
test1234@gmail.com	✉	Nov 24, 2020		rKz7vmRv9tSwoXXFdngXGjgDSMR2

Rows per page: 50 1 – 5 of 5

Figure 2: Authentication Tab in Firebase

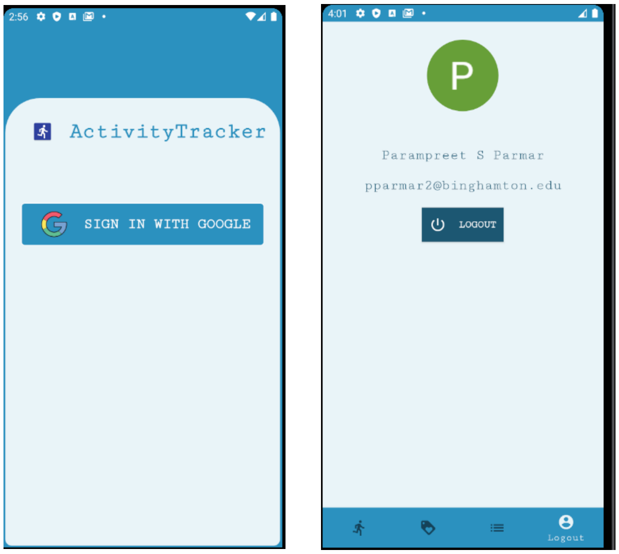
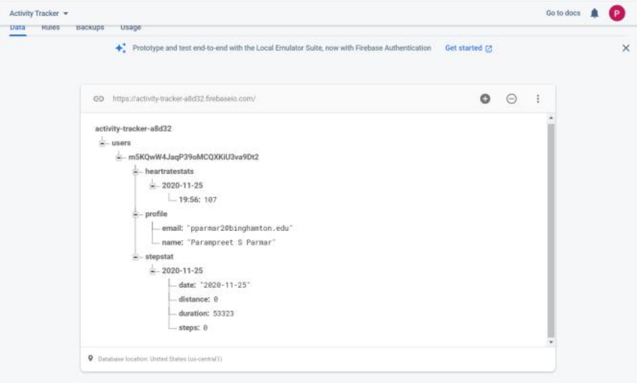


Figure 3: Login and Logout Page of the Activity Tracker App

2.2.1 *Implementation.* I Integrated the Firebase database account to the app using the generated *google-services.json* file that was created while creating the database instance in the Firebase console. I needed to copy the *google-services.json* file in our project under the app/ directory in the Android code.

Once a user is logged in with a validated account, they are able to use the application. In the Firebase database, I created a Realtime Database instance called activity-tracker-a8d32. This instance stores the user's health and login data.



Activity Tracker | Go to docs

Data | Rules | Backups | Usage

Prototype and test end-to-end with the Local Emulator Suite, now with Firebase Authentication Get started

https://activity-tracker-a8d32.firebaseio.com/

```
activity-tracker-a8d32
  users
    m5KQwW4JaqP39oMCQXKiU3va9Dt2
      heartratestats
        2020-11-25
          19:56: 107
      profile
        email: "pparmar2@binghamton.edu"
        name: "Parampreet S Parmar"
      stepstat
        2020-11-25
          date: "2020-11-25"
          distance: 0
          duration: 53323
          steps: 0
```

Database location: United States (us-central1)

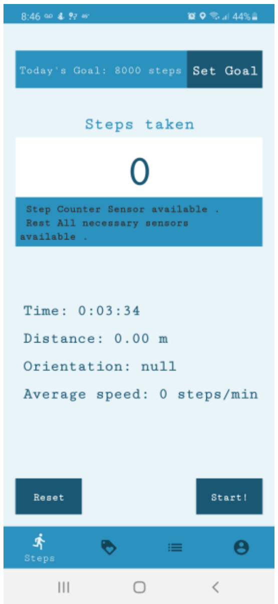
Figure 4: Database Tab in Firebase

Figure 4 shows the Activity Tracker real-time database tab. Users is split into three different children. *Profile*, which store the name and email of the user. *Heartratestat,* which shows the last date which the user measured their heartrate and there beats per minute (bpm) at that given time. Finally, the last child is the *Stepstat*, which contains the date at which the user measured their steps, there total distance, the total duration of how long the user measured their steps, and the total step count. Each user that is authorized on the database and that first logs into the application will get their own unique user-ID, and therefore their own user children.

2.3 Android Application

To prototype my application, I used a Samsung Galaxy S20 5G phone running android version 10.0. The Samsung S20 has a 12GB of RAM and 128GB of internal storage. It has a 3200x1440 display, with a 12-megapixel front camera [3]. Like all modern smart phones, the S20 has a built-in step counter which is a mandatory to run the Activity Tracker application. To develop the android application, I used Android Studio [4].

2.3.1 Activities. The android application is organized into three Activity classes, which are focused tasks that the user can perform on the app [5]. The *LoginActivity* class is the screen that the user sees when the app is first open. This class is responsible for the user google sign and interacting with the firebase database. On a successful login, the *LoginActivity* class opens the *HomeScreenActivity* class, which is the main app screen.



Today's Goal: 8000 steps

Steps taken

0

Step Counter Sensor available .
Rest All necessary sensors available .

Time: 0:03:34

Distance: 0.00 m

Orientation: null

Average speed: 0 steps/min

Figure 5: Main App Screen

Figure 5 shows the main app screen of Activity Tracker. The bottom tab is split up into four different pages. The main steps page, the heartrate, records, and logout page. The top set goal button lets the user enter a targeted step count goal. For a user to start tracking their steps, they must press the 'start' button. When they are ready to stop, they will need to click the 'stop button'.

2.3.2 Adapters. Activity Tracker is organized into two adapters. *RecordsHeartAdapter* contains the implementation for the heart rate record. When a user measures their heartrate, it is stored in the heart rate record tab. While *RecordsStepAdapater* contains the implementation for the step counter record tab.

2.3.3 Fragments. The bottom navigation view, shown in figure 5 is split up into four different fragments. *RecordsFragment* contains the implementation of the Records tab. *HeartRateFragment* contains the implementation for the heart rate tab. This class does all the heartrate measurement, it calls the camaera and flashlight to start receiving frames and sends those 'snapshots' to the imageprocessing class for analyzing and decoding. *ProfileFragment* contains the implementation of the user log out tab. And finally, *StepsFragment* is the implementation of the step tab.

2.3.4 Classes. There are two helper classes. An *ImageProcessing* class that is used for the heartrate measurement. The class has two functions, that decode the frame and returns the amount of red from the given frame data. And a *Constant* class that contains constant variables.

2.4 Step Counter

The main step counter feature is implemented in the *StepService* class. By implementing *SensorEventListener* listener within the *StepService* class and overriding two methods *OnSensorChanged* and OnAccuracyChanged, I was able to start tracking a user step. Next, I initiated the *SensorManager* and Sensor and have the Sensor registered as a listener within the activity, now we need to implement the *onSensorChanged* function that will be triggered by a *SensorEvent* whenever there is a change to the Sensor we registered.

3 EVALUATION

A walk through of my application can be seen in the demo video: [http://y2u.be/uG7tTiBi0Ew]. The step counter has a short delay of about 10 seconds. The implementation is not sophisticated enough to be 100% correct. The sensor checks if there is a spike in the accelerometer data and assumes that the spike is coming from a step. The spike may actually come from a step or from a user simply shaking their device. Even sophisticated fitness trackers, ones that use machine learning, signal processing, and other methods still have difficulties to determine if a step is real or just simply a user playing with there device.

4 CONCLUSION

In this project, I developed an android application that tracks a user's step counts and heart rate. A user will need to login through an authenticated google account once they open the application. Once a user is logged in, they are able to start tracking their steps or measure their heart rate. A user is also able to see their previous measured step and heart rate records.

Living a healthy lifestyle can reduce the risk of developing chronic diseases and long-term illnesses. Activity Tracker approaches this challenge by providing a simple application that tracks a user's step counts and measures their heart rate. Although, the application is not perfect, it is a starting point in providing a simple and accessible tool to help control a user's health.

ACKNOWLEDGMENTS

I would like to thank Professor Mo Sha in Binghamton university, and our teaching assistant Junyang Shi for their helps and for leading and encouraging us to create an IOT project as a solution to a real-life problem.

REFERENCES

[1] Health.2016.https://www.tfah.org/report-details/state-of-obesity-2020/#:~:text=The%20U.S.%20adult%20obesity%20rate%20stands%20at%2042.4%20percent%2C%20the,by%2026%20percent%20since%202008.

[2] Google. 2011. Firebase. https://firebase.google.com/.

[3] Samsung. 2018. Samsung Galaxy. https://www.cnet.com/news/samsungs-galaxy-s20-raises-the-bar-for-phones-in-8-ways/

[4] Android. 2013. Download Android Studio. https://developer.android.com/studio.

[5] Android. 2007. Activity. https://developer.android.com/reference/android/app/Activity.