ECSE-682 Assignment-2
Pedometer Application in Android

Katyayani Prakash* and Wenqi Liang†
Department of Electrical and Computer Engineering,
McGill University, Montreal, Canada
Email: *katyayani.prakash@mail.mcgill.ca, †wenqi.liang@mail.mcgill.ca,

***Abstract*—We have developed an android application for a pedometer with 2 activities. The main activity counts the number of steps completed, displays the daily step goal of the user, has a button for resetting the step count to zero, and one for entering goals. The second activity is for user to input their step goals. Goals can be set in the forms of steps, calories, or duration of their walk. On completion of these goals, user receives a notification of the same. This application works in the background as well; steps are counted even when the app is minimized, and once the goal is met, user receives a toast message of goal completion. This background activity is achieved with the help of a service.**

***Index Terms*—Android Application, Activity, Service, Step Detector, Pedometer**

I. Introduction

There are many sensors provided by Android platform that let developers monitor the motion of a device. Because the objective of this assignment, we focus on `TYPE_STEP_COUNTER` and `TYPE_STEP_DETECTOR`, which are two motion sensors that can monitor the movements of steps. `TYPE_STEP_COUNTER` returns number of steps taken by user since the last reboot while the sensor is activated. As for `TYPE_STEP_DETECTOR`, it triggers each step motion event by the user. In order to reset the counter without system reboot and to trigger each step event, we choose `TYPE_STEP_DETECTOR` sensor for our application.

A. Step Detector

`SensorManager` allows developer to access device's sensors. We register `SensorManager` in `OnResume()` and unregister it in `OnPause()`. By setting sensor type to `TYPE_STEP_DETECTOR` in `getDefaultSensor(int type)`, sensor type is initiated. The timestamp of this sensor type indicates when a step occurred. A step occurs at a high variation in acceleration by a foot hit the ground. In order to prevent large power consumption and to count the steps when minimizing the application, background service is needed and implemented, which will be introduced in section B.

Sensor event listener is an interface for receiving notifications from `SensorManager` when new sensor event occurs. Main activity of our application implements this interface. Permission is required for movement detection of sensor activities. User permission needs to be set to `android.permission.ACTIVITY_RECOGNITION` in the AndroidManifest.xml file for allowing sensing activity.

B. Service

Service can perform long-running operations in the background. Once a service is started, application component continues to run in the background and even when user switch to another application. Service starts by calling `startService()`, and similarly ends by calling `stopService()`. Once a service starts, `OnStartCommand()` method is called in the service file.

II. Our Application

In this assignment, we have developed an android application for a pedometer with 2 activities. The main activity counts the number of steps completed in a circular progress bar, displays the daily step goal of the user, has a button for resetting the step count to zero, and one for entering goals. The second activity is for user to input their step goals. Goals can be set in the forms of steps, calories, or duration of their walk. Calories and minutes of walking are converted to number of steps which are then set as goals for the user. On completion of these goals, user receives a notification of the same. This application works in the background as well; steps are counted even when the app is minimized, and once the goal is met, user receives a toast message of goal completion. This background activity is achieved with the help of a service. Below we show the app behaviour for different use-cases:

A. Setting Goals

The user has 3 ways of setting the goal. While the step input is a straightforward method, we have internal formulas for converting calories and minutes of walk to steps:

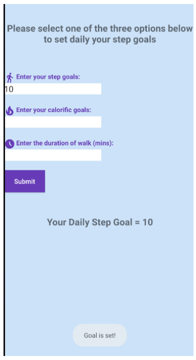
$$Number\ of\ Steps = Calories * 25$$

$$Number\ of\ Steps = Minutes * 80$$[1]

B. Tracking Steps and Goal Progress

User will be notified that goals are set (via a toast notification), and they can track their goal progress via the circular progress bar in the main activity. The progress bar has it's maximum value set to the goal, and thus, which each step, it provides a very visual display of the proportion of goal that is met.

[1] Based on an average healthy individual, who burns 1 calorie in 25 steps, and takes 80 steps per minute.



Please select one of the three options below to set daily your step goals

Enter your step goals: 10

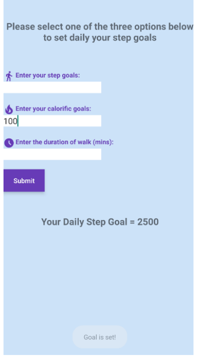
Enter your calorific goals:

Enter the duration of walk (mins):

Your Daily Step Goal = 10

Goal is set!

Fig. 1. Setting goals by entering number of steps



Please select one of the three options below to set daily your step goals

Enter your step goals:

Enter your calorific goals: 100

Enter the duration of walk (mins):

Your Daily Step Goal = 2500

Goal is set!

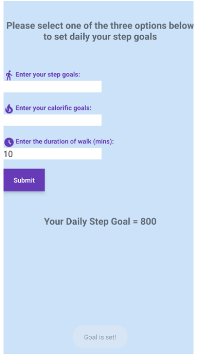
Fig. 2. Setting goals by entering number of calories

C. Background Performance of App

We have implemented a service routine, due to which, the app remains running in background even when it has been paused in the foreground (by going to some other app, or returning to Home Screen). Thus, if the user minimizes the app after setting their goals, the app will continue counting the steps, and notify them of their completion on their present screen.

III. Conclusion

In this assignment, we have built a pedometer android app from scratch, consisting of 2 activities and 1 service. Incorporation of circular progress bar provides the user with real-time visual feedback of their step count and goal progress. Toast messages give the user notification both in foreground and background, when their goal is achieved.



Please select one of the three options below to set daily your step goals

Enter your step goals:

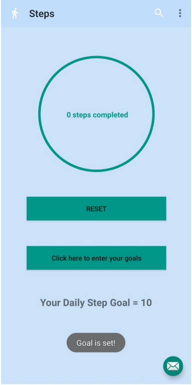
Enter your calorific goals:

Enter the duration of walk (mins): 10

Your Daily Step Goal = 800

Goal is set!

Fig. 3. Setting goals by entering minutes of walking



Steps

0 steps completed

Your Daily Step Goal = 10

Goal is set!

Fig. 4. Goals Set Notification in Main Activity

References

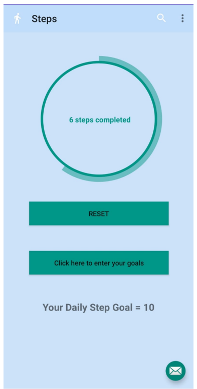
[1] ”Create Step Detector Application : Android Sensor Programming Tutorial”, *Sarthi Technology*, 2021. [Online]. Available: https://www.youtube.com/watch?v=-uYPDgT40eg. [Accessed: 11- Oct- 2021].

[2] ”How to create a Step Counter/Pedometer in Android Studio (Kotlin 2020)”, *Code Palace*, 2021. [Online]. Available: https://www.youtube.com/watch?v=WSx2a99kPY4&t=20s. [Accessed: 11- Oct- 2021].

[3] N. Panigrahy, ”How to Create a Background Service in Android - DZone Mobile”, *dzone.com*, 2015. [Online]. Available: https://dzone.com/articles/how-to-create-a-background-service-in-android. [Accessed: 15- Oct- 2021].

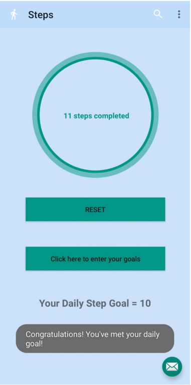
[4] ”Create a background service — Android Developers”, *Android Developers*, 2021. [Online]. Available: https://developer.android.com/training/run-background-service/create-service. [Accessed: 15- Oct- 2021].

[5] Lecture Slides of ECSE-682: Topics in Computers and Circuits by Prof. Zeljko Zilic



Your Daily Step Goal = 10

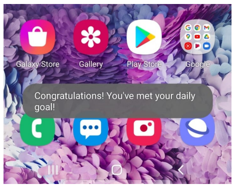
Fig. 5. Goal in Progress (6 out 10 steps completed)



Your Daily Step Goal = 10

Congratulations! You've met your daily goal!

Fig. 6. Goals Met Notification in Main Activity



Congratulations! You've met your daily goal!

Fig. 7. Goals Met Notification while app is in background