

Developing Smartphone App to Protect Personal Data

Delilah Florea, Andrei Vasiu
Informatics and Computer Science Department
Samuel von Brukenthal National College of Sibiu, Huet Square, No. 5
Sibiu, Romania
{vasiu_andrei2000@yahoo.com} {delilah_florea@yahoo.com}



ABSTRACT: *The technology is all around us, but how sure are we that it is safe for us? How can we know that our privacy is being protected? Many smartphone users are constantly connected to the Internet and to many social media platform and most of them share their location on the Internet, which makes them visible to everyone. This application is based on sensors and it replaces a GPS that works only with internet and location. It helps the user to don't get lost in an unknown place, starting at a reference point. The user is able to walk anywhere he desires and with the help of this app he can turn back to his reference point whenever he wants. This works regardless the user's position in the world as it does not use the Internet or the location.*

Keywords: Mobile Apps, Graphic Users Interface, Software Design, Reminders

Received: 13 May 2019, Revised 4 September 2019, Accepted 17 September 2019

DOI: 10.6025/jet/2019/10/2/31-37

© 2019 DLINE. All Rights Reserved

1. Introduction

Any form of mobile computing systems contains huge personal and professional information of their owners [1]. In the digital communications era these data become vulnerable. Many security breaches are generated by human actions due to various factors like: people are not trained and have poor security awareness or people are malicious and sometimes expose deliberately to risk [2]. This mobile application helps keep safe personal data and contributes to keep the phone's battery by saving space and not using the Internet.

“Tracker” is an application designed for all smartphone user, that have implemented sensors in their phones, which aims to help those who moved in a new city or are adventuring themselves in a forest, basically, it helps those who want to discover unknown places.

The young generation has always enjoyed having mobile applications that helped us during the day, such as alarms, timers, time tables, Google Maps and others. This made us to have a passion towards application and made us want to know how they work. We have been analyzing and testing various apps until we decided to work on our own application that shall be useful.

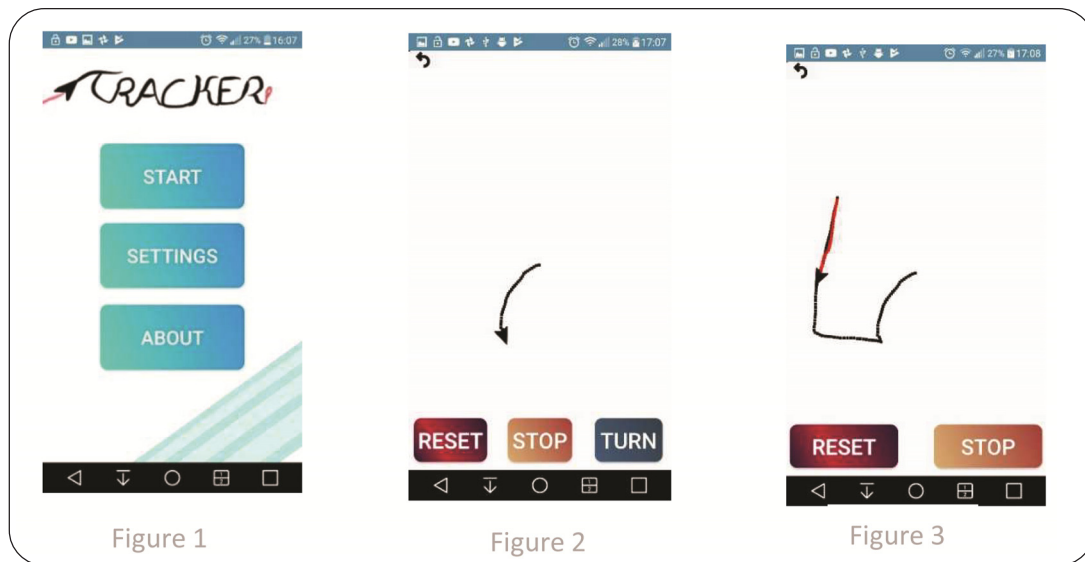
This application is based on an old story, and that is “*Hänsel and Gretel*”. In the story, Hansel and Gretel are siblings that are cast out of their parental house. The night the parents decided to get rid of them, the father walked them deep into the forest. But because the children overheard their conversation, they were prepared for this, so they took bread and on the whole way into the forest, they kept leaving breadcrumbs on the way to be able to find the way back home from wherever the father would have left them. “*Tracker*” works on the same principle, it starts drawing a line the moment the user presses on “*start*”. Furthermore there are 3 other buttons:

- “*reset*”, that resets the whole way and bring the “*navigator*” back in the middle of the screen
- “*stop*”, in case the user occurred a problem and has to rest for a while. After it is pressed it turns to “*resume*” and the user can go on with his “*journey*”
- “*turn*”, a button lets the application know that the user has reached its destination and wants to turn back, so the way’s color turns to red.

This paper is structured as follows. Section 2 presents the user interface and application functionality. Section 3 describes the application software design emphasizing on the code sections and the programming environment. Finally, Section 4 highlights the main contributions, advantages of developed application and proposes some possible further work.

2. User Interface and Application Functionality

The first activity shown in Figure 1 is the menu that has 3 different buttons. I gave the application one specific color matching palette, that makes the application easy to follow and it does not tire the eyes.



The other two pictures show how “*Tracker*” looks. In Figure 2 there are three buttons, but none of them is pressed, so that the application works normally, the user walks in real life and the application leaves a trail starting from the center of the screen.

3. Application Software Design

3.1 Code sections

The concept is simple, as almost all smartphone that are owned nowadays have these basic sensors: *StepCounter* and *Orientation*. From the sensor *Orientation* I have taken the values of a compass and it indicates me where north is.

```
int degree = Math.round(sensorEvent.values[0]);
```

The *sensorEvent.values* can take more values. We used 0 because it indicates the north and it detects movements on the x Axis, while with value 1 it would detect movements based on the y Axis. Other values take all directions, but only the 0 has a reference point, the north.

Based on this, each time the smartphone detects a step, the small *navigator* from the screen moves a few pixels in the directions the user is heading:

```
if ( (degree / 90) == 0) {
    dirX = (degree / 90f - (degree / 90));
    dirY = -1 + (degree / 90f - (degree / 90)); // x,00
}

if ( (degree / 90) == 1) {
    dirX = 1-(degree / 90f - (degree / 90));
    dirY = (degree / 90f - (degree / 90)); // x,00
}

if ( (degree / 90) == 2) {
    dirX = -(degree / 90f - (degree / 90));
    dirY = 1 - (degree / 90f - (degree / 90)); // x,00
}

if ( (degree / 90) == 3) {
    dirX = -1 +(degree / 90f - (degree / 90));
    dirY = - (degree / 90f - (degree / 90)); // x,00
}
```

This is an algorithm that works based on the 4 main parts of a coordinate system. 0 indicates the north.

- Between 0 – 90 degrees, right from north, the *navigator* moves on the x Axis, on the positive side of it, meaning *dirX* is positive and it moves up the screen. This means that *dirY* has to be negative in order to move up.
- Between 90 – 180 degrees, the *navigator* moves in the positive side on the x Axis, but still on the negative side of the y Axis.
- Between 180 – 270 degree, the *navigator* moves in the negatives sides of both Axis.
- Between 270 – 360 degrees, the *navigator* moves on the negative side of the x Axis but on the positive side of the y Axis.

In order for the image *navigator* to move, it only changes its position on the screen:

```
posY = posY + (int) (speed * dirY);
posX = posX + (int) (speed * dirX);

navigator.setX(posX);
navigator.setY(posY);
```

An advantage of “Tracker” is that it tells the user if he does not have one of the sensors implemented. It also says which one is missing:

```
@Override
protected void onResume() {
    super.onResume();
    running = true;

    if(StepSensor != null)
        sensorManager.registerListener(listener: this, StepSensor, SensorManager.SENSOR_DELAY_UI);
    else
        Toast.makeText(context this, text "No StepCounter sensor detected", Toast.LENGTH_LONG).show();

    if(OrientationSensor != null)
        sensorManager.registerListener(listener: this, OrientationSensor, SensorManager.SENSOR_DELAY_UI);
    else
        Toast.makeText(context this, text "No Orientation sensor detected", Toast.LENGTH_LONG).show();
}
```

3.2 Inside the Canvas class

The part described in the previous pages is the main class, which has implemented the sensors. Another class, that is important, is a class that extends a view. This class creates a canvas on which the path is drawn.

```
mPaint = new Paint();
mPaint.setAntiAlias(true);
mPaint.setStyle(Paint.Style.STROKE);
mPaint.setStrokeJoin(Paint.Join.ROUND);
mPaint.setStrokeWidth(16f);
```

This is the brush used to draw the path. In order for the color to be switched when the *Turn* button is clicked, I have added the following code:

```
switch (MainActivity.turn){
    case 1:
        mPaint.setColor(Color.RED);
        break;
    case 0:
        mPaint.setColor(Color.BLACK);
        break;
}
```

For having the opportunity to save a route, the actual route has to be saved in an array of points:

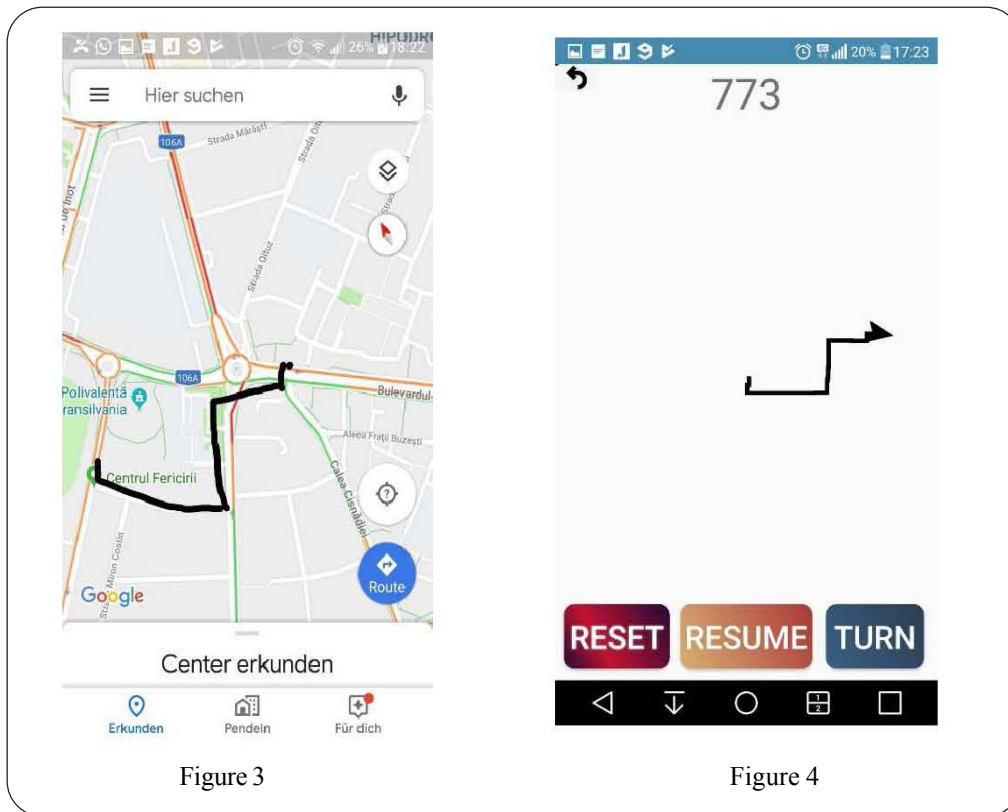
```
canvas.drawPoint(MainActivity.pointsX[i], MainActivity.pointsY[i], mPaint);
```

This *Offline Tracker* is the same for any width and height:

```
DisplayMetrics metrics = new DisplayMetrics();  
((Activity) getContext()).getWindowManager().getDefaultDisplay().getMetrics(metrics);  
height = metrics.heightPixels;  
width = metrics.widthPixels;
```

3.3 The Programming Environment

This app was created with the in Java using Android Studio 3.1.2 as a programming environment. It has been tested both on phones with no sensors and also on new smartphones [3-4]. Our choice is motivated by the fact that, according to market research 49.2% of mobile devices sold worldwide in 2012 have used Android System Architecture [5]. To test our app and the sensors to see if it draws an accurate path, we turned *Tracker* on during some of our walks through the city and when we wanted to turn back from where we started, we took our phones out from my pocket and followed the black path.



Here are two images that show how effective this app is:

Figure 3 shows a preview of the way from a certain point to another. This picture has been taken from Google Maps.

Figure 4 is what the app has detected during my walk from the same point to my destination.

This is a comparison between what the application has detected and the same route drawn in Google Maps.

4. Conclusion and Further Work

Overall, the application we have developed can be used in many domains and it can be helpful for many sportive people, such as mountain climbers.

We believe that *Tracker* is a very efficient application, we can tell this from our own experience, as our phone's battery dries like seconds when we have our phones location turned on and the mobile data. This may not apply to all smartphones, of course, but it is still an efficient way to save battery and space.

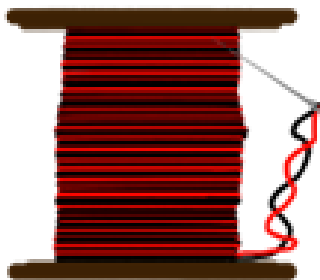
Why use this application when almost everyone has Internet connexion?

Tracker can be called an "offline Tracker". It does not show the direction between 2 points, it does not lead you to a destination **you want to go** and instead it helps you go everywhere. It is useful, because where is no Internet connexion or signal, you can still be assured that you can turn back to your reference point without worrying that you might get lost. This place can be a forest, a desert, a jungle, a new city in a new country. It also lets the user explore, an aspect that is healthy and interesting. Many do not know what their own surroundings have to offer. Using Google Maps, you go straight from point A to point B. Like this, you do not know where you adventure and you can discover a lot more.

What about Google Maps offline?

Google maps offline is not yet fully developed and besides that, it occupies much of the phones storage. This application only takes a few MB and yet again, it is not designed to be like a map. Its efficiency and unique part consists in the fact that it leads the user to discover more, protect his location and personal data. A feature will be that *Tracker* will use Google Maps offline and draw the route you walked on, on the maps.

In future updates, "*Tracker*" will be an application to help those who suffer of Alzheimer. The user can leave an email in the app saying that the user has Alzheimer, he writes the email of the person, who is in charge of him. With this done, the user can go outside and the person in charge will receive emails every 5 minutes with the activity of the user, this activity can be shown with captures of the screen or time intervals in which the user might have moved or not. While the user uses his phone normally, this application will record his activity. This is useful, because, in case the user loses his keys, he is then able to walk on the way he has already walked that day and have a chance to find the keys. Another idea is that the user will be able to save a route and whenever he wants to walk the route again, he can choose it. Let us say that we live in another country, where we cannot use our mobile data and we have to go to a shop. We start this application and we go outside and search a store. When we find one, we can save the route till there and name it "*Best Juice Store*" and the next time we are looking for a store we can already follow a route. And maybe the next time we can follow the same direction but find a shorter way.



References

- [1] Amato, F., Cozzolino, G., Mazzeo, A., Moscato, F. (2018). An Advanced Methodology to Analyse Data Stored on Mobile Devices. *In: Castiglione A., Pop F., Ficco M., Palmieri F. (eds) Cyberspace Safety and Security. CSS 2018. Lecture Notes in Computer Science, vol 11161. Springer, Cham.*
- [2] Govender, S. G., Looock, M., Kritzinger, E. (2018). Enhancing Information Security Culture to Reduce Information Security Cost: A Proposed Framework. *In: Castiglione A., Pop F., Ficco M., Palmieri F. (eds) Cyberspace Safety and Security. CSS 2018. Lecture*

Notes in Computer Science, vol 11161. Springer, Cham.

[3] Android Developer Tools, 2014, <https://developer.android.com/studio/intro/index.html>, accessed 05 May 2019.

[4] Udemy Online Courses, The Complete Android N Developer Course, <https://www.udemy.com/complete-android-n-developer-course/>, accessed 28 February 2019.

[5] Ivanovic, M., Jain, L. C. (2014). E-Learning Paradigms and Applications, Agent-based Approach, Eds. Springer.