

Safety Application in Android

Arun Kumar Sharma

Department of Computer Science & Engineering,
NIT Hamirpur, (H.P.) India.

(Corresponding author: Arun Kumar Sharma)

(Received 12 April 2019, Revised 03 June 2019, Accepted 14 June 2019)

(Published by Research Trend, Website: www.researchtrend.net)

ABSTRACT: Smartphones are revolutionising the field of mobile computing these days. Along with productive suites, Android smartphones also provide many lifestyle-related solutions, like fitness tracker. Such apps are not only free, but in some way also improve the quality of living. So, apart from these apps, we can also make use of the power of smartphones to create such systems that help us in keeping people safe and secure wherever they go. The idea is to create an alert system, through which a user can request help in case he is faced with an emergency. Not to mention the rising cases of untoward incidents that we are facing each day. The Safety Application is an effort in this direction, to provide users a channel through which they can request help when in distress.

Keywords: Android, GPS, Google Maps, Application, Mobile, Computing, Location, Smartphones, Message, Alert System.

I. INTRODUCTION

Motivation of smart phones has always been to improve the comfort of the entire population. With smart phones, a revolution in mobile apps has begun, allowing users to do their jobs and entertainment anywhere. This mobility also enables us to develop some useful software that can improve overall overall well-being. The security application uses this option and provides users with secure coverage wherever it is. The Android app provides a feature that alerts users' contacts in case of any extraordinary event. The current user's location is also transferred to these contacts and helps them track the victim/user. Smartphones enable us with useful utilities on- the-go. So, why not use this to our advantage, and create an application that can keep us secure too. Especially keeping in mind the current scenario, as crime rate is rapidly increasing across the country. This thought is the main motivation behind the project. User on installing the app register themselves, and also enters mobile numbers of people to contact in case of emergency. The application provides a "Help button", which on being pressed sends a distress message to all the pre-entered contacts, along with a Google Maps link to the "victim's" location.

This app is a mobile app for Android that provides users with some security tips. Helps users ask for help so that a user can connect to predefined contacts, along with the Google Maps link for the current location. And all, after pressing a single button [7-11].

II. ANDROID

Today Android is the most efficient and efficient mobile operating system today. From the Linux kernel, it is now Google and supports its range. The use of Linux is largely due to Android being an application that has benefited from Fellows [2-4] a major source of development in recent years which also involves Content-Based Filtering [12].

The many reasons why Android is a preferred platform for mobile applications can be effectively seen from the following figure:

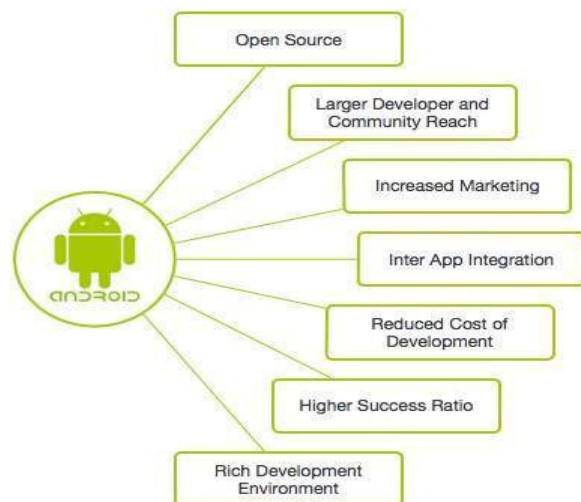


Fig. 1. Why prefer Android.

1. Building blocks for Android application:

Activities and Intents:

Activity is basically any screen that the user views at a given time, like dialling a phone, clicking a picture, etc. As we all know, all C, C++ or Java programs starts execution from the main() function in the program. Analogous to this, an activity is started by executing the on Create() function. Rest of the function are then called as the case may arise. An application basically contains many activities, and user flips back and forth through these activities, we also commonly witness data transferring between one activity to another. This is obvious, as to perform many actions, data is required from a previous execution. Whenever a user switches from an activity, the older activity may be replaced by the screen of the new one, but its state is temporarily stored in the "back-stack". Each application view has a specific task to

perform, depending on the state of the application. The activity life cycle is illustrated in Fig. 2.

Intents are the messages sent between different building blocks. We can visualise them as the message-broadcaster between different activities. An intent initiates an activity to instruct another activity to start-up, terminate or any other message that needs to be passed on. The data-flow talked about above can also be performed using intents.

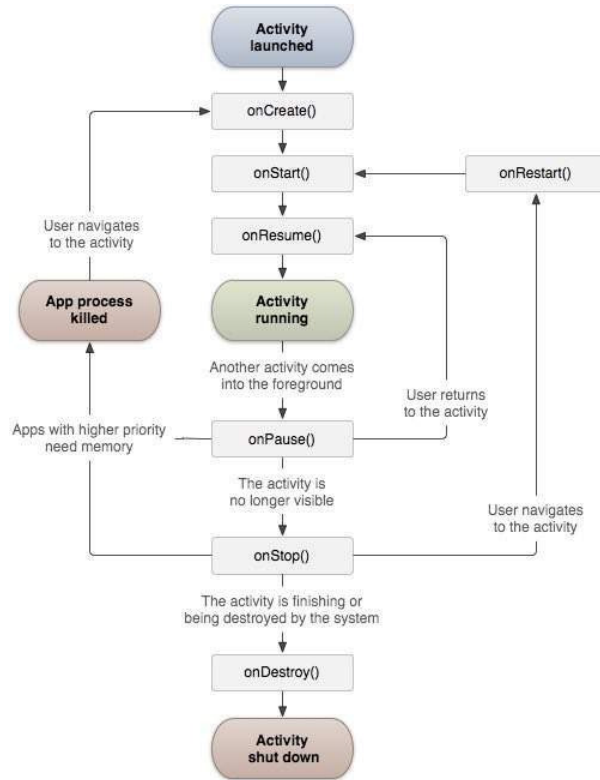


Fig. 2. Activity life cycle.

On executing onCreate() (creating the desired activity), we then see execution of onStart() and onResume() methods sequentially. This is the case for all activities. In case an activity is idle, its visibility is removed by executing the onPause() method. It is worth noting that, no activity executes onStop() directly, first onPause() method needs to be executed.

Databases:

Android provides 3 ways of data storage facility:

- Saving key-value pair in simple Shared Preferences file.
- Saving arbitrary files in Android's file system
- Saving in databases managed by SQLite

As with any application dealing with data, Security app too depends on databases. Databases help store information in a structured manner, based on the type of data and the data field, see [5].

By default, each application creates its own database to work with, over which it has full autonomy. No other application is allowed to access any other application's database.

Adapters:

Adapters are the components that actually link data and views together. Data shown in views comes from databases, and this connection that enables data communication is enabled through adapters. As is evident then, adapters have a huge role to play in the working of any Android application, Android provides many adapters to provide developers' needs, like simpleCursorAdapter, spinnerAdapter, listAdapter etc.

Services:

While activities provide a GUI-based interface to perform tasks, services are more background-running type processes. They perform background or ongoing processes, like playing music, downloading files, etc. Some of their features are:

- Services don't support GUI in any way
- Generally, services may not be started by users, and are instead initiated by some other app in the smartphone
- In case of priorities, services supersede idle activities.

Location Based Services:

Location Based Services are provided by Android through its location framework. This service is a very useful one, and is expressly needed in apps running on location-based functionalities. Basically, the service provides a framework to exploit Google Maps API, to get the exact coordinates of the targeted user.

For this, Location Manager is used, which in turn uses the GPS. Through this, we get the exact geographical coordinates of the smartphone, hence the user too. Those coordinates are then used by LBS to create the Google Maps link to the coordinates obtained from Location Manager.

To use this service, we will also need to provide our application some permissions, which is done in the "manifest" file. The permissions look like:

```
<uses-permission
android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission
android:name="android.permission.ACCESS_INTERNET"/>
<uses-permission
android:name="android.permission.SEND_SMS"/>
```

To improve sensitivity to movement, Location Listener is used. Basically, any changes in coordinates returned by Location Manger is recorded by the listener, and the next link generated accommodates for that.

III. METHODOLOGY

A. Technologies Used

- Android Studio with API 23 (Android 6.0 Marshmallow) compatibility for application development.
- For database architecture (DDL/DML): SQLite.
- Use of cellular network services for implementation of calls/messages using SMSManager.

B. Working

The following figure shows the work-flow of the application:

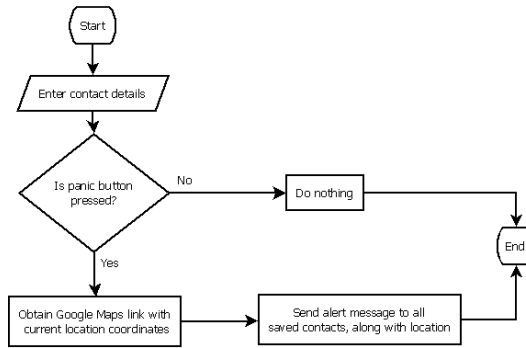


Fig. 3. Flowchart of application.

A new user needs to add contacts which will be contacted during emergency. Thus, as the panic button is pressed, a message is sent to those saved contacts through the messaging service of the smartphone. The message will also contain the Google Maps link, which will give the current location of the victim. As many times the button is pressed, a message gets sent to those contacts [1, 6].

IV. SIMULATION

The following Fig. 4 shows the main interface that will appear while opening the Security App which will have a big red button showing Panic which needs to be pressed in case of emergency.



Fig. 4. Main Interface.

The Fig. 5 denotes the Menu Sidebar which have an icon called as Settings.

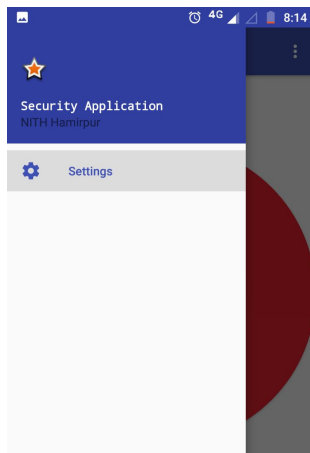


Fig. 5. Menu Sidebar.

The following Fig. 6 asks the user to Enter his Name and Contact Number and an interface to update or delete existing contacts.

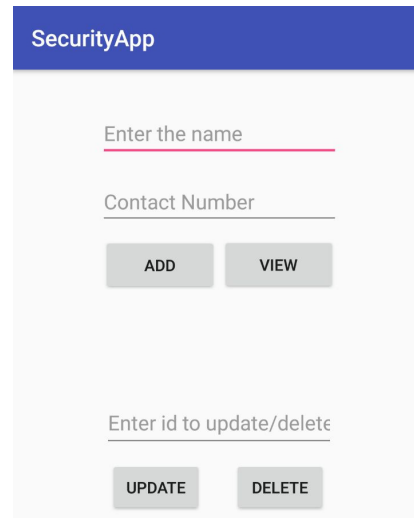


Fig. 6. Adding/Deleting Contacts.

The Data represented are being viewed after it has been entered as shown in Fig. 7.

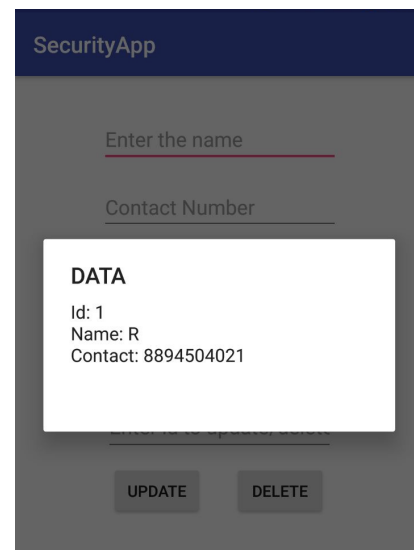


Fig. 7. Viewing contacts entered.

The message which states "I need help. I am in danger. This is my location" is received by the emergency contacts along with the location in which the person is presently as shown in Fig. 8.

The location in which the person is originally present is sent through the Google Maps as illustrated in Fig. 9.

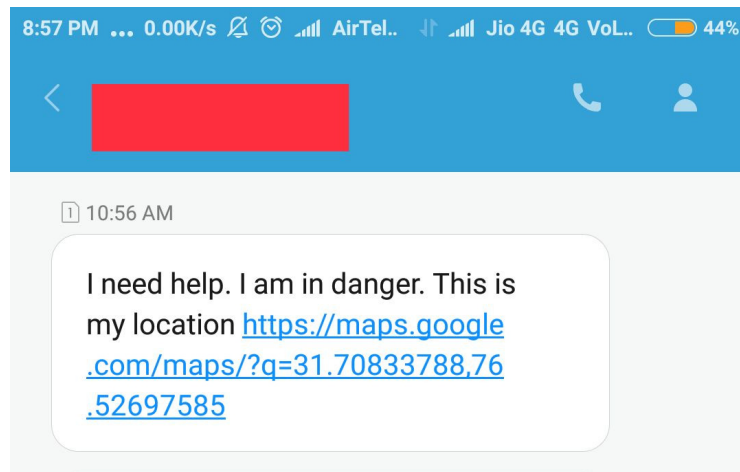


Fig. 8. Message received by the emergency contacts.

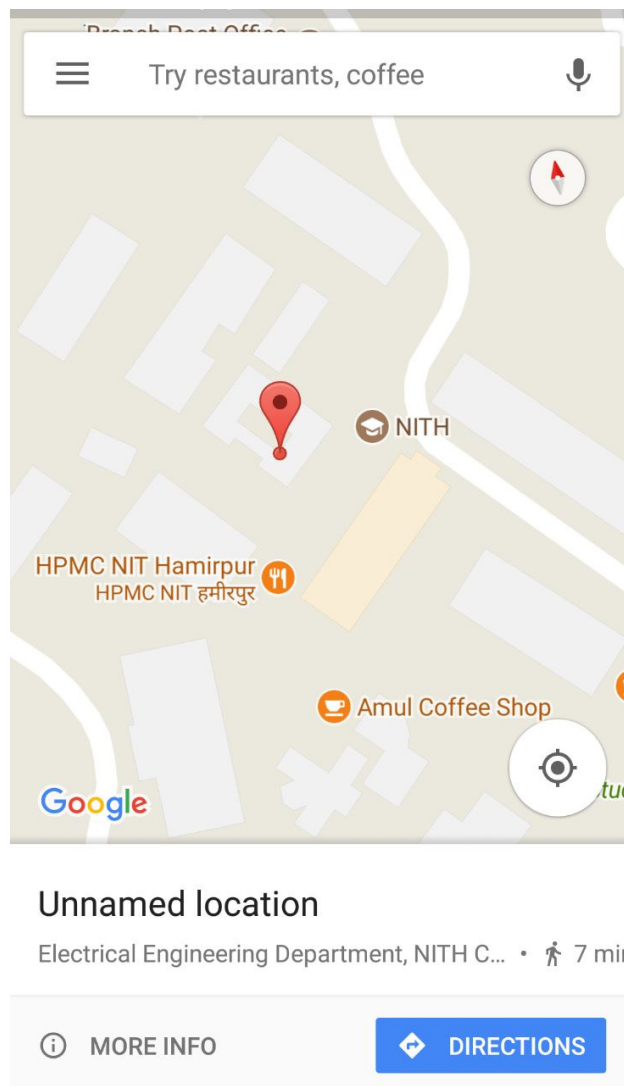


Fig. 9. Location on Google Maps.

V. CONCLUSION

The proposed Android application has been successfully implemented. The Message alert system and Google Map API usage for GPS-based location services have been implemented as proposed.

On installation, user first needs to add contacts he/she wishes, as emergency contacts, to be contacted during emergency. When panic button is pressed by the user, the SMSManager and LocationManager come into action, sending a distress message to those saved contacts, containing a Google Maps link to the user's current location, through his/her GPS coordinates.

VI. FUTURE WORK

Existing application have a huge scope for improvement, like:

- Contact nearby people who are NOT registered with the app, as it is highly unlikely if registered contacts are present at the nearby location
- Improve the location of the victim through IoT sensors.
- A person trapped within a large building will only be shown as a dot, rather than providing a near exact location of floor/room. WPS can be used to provide the exact location.
- By using Sensors, we can narrow down to a near exact location to quickly reach to a person in need.
- Using reference of nearby "objects" than "landmarks", efficiency can be improved.
- The existing Though the safety app is already in place, still the locational GPS services provided may not always be accurate.
- To prevent data theft, the location of users can be fetched from local device itself and not store it in database, and then multicast to targeted devices.

REFERENCES

- [1]. Aggarwal, C. C. (2016). *Recommender systems* (Vol. 1). Cham: Springer International Publishing.
- [2]. Chin, E., Felt, A. P., Greenwood, K., & Wagner, D. (2011, June). Analyzing inter-application communication in Android. In *Proceedings of the 9th international*

conference on Mobile systems, applications, and services (pp. 239-252).

- [3]. Craig, C., & Gerber, A. (2015). *Learn android studio: build android apps quickly and effectively*. Apress.
- [4]. Dunn, G., Wiersema, J., Ham, J., and Aroyo, L. (2009). Evaluating Interface Variants on Personality Acquisition for Recommender Systems. In: Houben, G.J., McCalla, G., Pianesi, F., Zancanaro, M. (eds.) *User Modeling, Adaptation, and Personalization. LNCS, 5535*, pp. 259-270. Springer, Heidelberg.
- [5]. Enck, W., Ocateau, D., McDaniel, P. D., & Chaudhuri, S. (2011, August). A study of android application security. In *USENIX security symposium* (Vol. 2, No. 2).
- [6]. Gao, H., Guo, C., Wu, Y., Dong, N., Hou, X., Xu, S., & Xu, J. (2019). Autoper: Automatic recommender for runtime-permission in android applications. In *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC), 1*, 107-116). IEEE.
- [7]. Hu, R., & Pu, P. (2010). A Study on User Perception of Personality-Based Recommender Systems. In: P.De Bra, A.Kobsa, and D. Chin (Eds.): *UMAP 2010, LNCS 6075*, 291-302.
- [8]. Linden, G., Smith, B, and York, J. (2003). Amazon.com recommendations: Item-to-Item collaborative filtering. *IEEE Internet Computing*, 76- 80.
- [9]. Resnick, P. & Varian, H. R. (1997). Recommender Systems. *Commun. ACM* 40, 56-58.
- [10]. Ricci, F., Rokach, L., & Shapira, B. (2015). Recommender systems: introduction and challenges. In *Recommender systems handbook* (pp. 1-34). Springer, Boston, MA.
- [11]. Rozakis, G., Gratzl, M., Tohida, K., & Yang, J. (2011). *U.S. Patent No. 7, 964,390*. Washington, DC: U.S. Patent and Trademark Office.
- [12]. Sharma, A. K. (2018). Content-Based Filtering in Movie Recommendation. *International Journal of Electrical, Electronics and Computer Engineering*, 7(2): 106-109.

How to cite this article: Sharma A. K. (2019). Safety Application in Android. *International Journal on Emerging Technologies*, 10(1): 234-238.