

Android Application for Ringing and Accessing Device Remotely Using SMS

Yogesh Pawar

Shivaji University, Kolhapur, Maharashtra, India

Email: yogeshpawar223@gmail.com

Abstract---Now a day's Android, Hoog (2011), applications are making our life much easier and happier. We always ensure that device that contains all applications should be always with us and secure anytime anywhere. In this journal I developed an android application to find your mobile phone using another device when you lost it nearby, using text message and unique code generated on both devices. The application is designed in such a way that we can ring phone even when it is in silent mode. For developing this application I proposed a method, I simply divided code modules of Android functions, so that we can able to invoke each module of android device separately to ring, vibrate and other function. Each android module will take device administrator permission to modify android function using application.

Keywords: backend, commands, remote devices, unique code.

Introduction

Text, images, audio, video, social media, documents, even more, we perform on our cell phone. The explosion of Internet and android applications leads us into the digital world and communication via digital data becomes frequent and a device that carries all this personal data is also as much more important than any other thing.

I developed this android application for enabling remote notification or ring, using another device via SMS. This application will provide a unique code protection generated on each device to prevent unauthorized application changes and to get access to getting SMS from any other cell number.

Different commands are used to operate this application. If one user wants to call any person but what if other person's cell phone is on silent mode and our work is too much important, we can forcefully switch cell phone's mode remotely Villan & Jorba (2013) using text message. The unique code should be known to user who is going to activate mode on the remote device. The command must include "rdd" (Remote Destination Device) keyword in command followed by destination code and after that module that we want to activate i.e. ring, silent, switch off etc. After sending this entire command as SMS to the remote user, the remote device will ring continuously until the user who sends this command will send STOP command.

The remainder of this journal is organized as follows. In section 2, the proposed application design and architecture is presented. The detailed description of modules and path is described in section 3. Several experiment results and application code are illustrated in section 4. Finally, concluding remarks, as well as some suggestions for future works, are stated in section 5.

Research Method

Explaining The sender side contain user interface of application through which user interacts with the application and use different services. The user interface consists of the Activation code, Application settings, and Activation logs. The user interacts with a user interface provided via application and activates different input commands.

The receiver side now acts as the backbone of application and perform an action according to commands received at the backend of application. It provides all the services for working of the application. The receiver side uses codes and stored unique code information for verification of received code and authorization of another device to receive

remote commands. It will perform execution of different queries at the backend on the based on commands it received. Queries results will activated appropriately on android application modules and show notification and display it using user interface. As soon as device activated command it sends confirmation of activation of command.

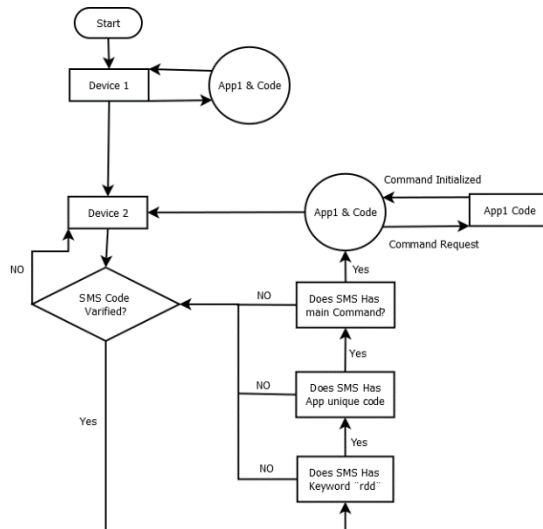


Figure 1. Block Diagram of sender-receiver side communication in-between applications

Results and Analysis

In There are mainly three components in this application as-

A. Homepage B. Application settings and C. Activation log.

Homepage

It includes a basic description of the application and detailed information about how it actually works. It can be used as a manual so that user can understand the use of the application. This page also provides instruction about application unique code for interaction using commands. This also page has two buttons for navigation to other two components.

Application settings

This page gives different types of application module settings using this options we can change default configurations, there are mainly four types of settings options as General settings, Activation code setting, Activation logging and Remote commands settings. Activation log it shows the activities done by the user from application i.e. whatever commands sent by another user all logs like time, command name, date in log page.

Taking about input from a user who wants to activate commands must know the unique code generated at remote device's application. The user should type a text message in this sample format as-"rdd Code Command", rdd is inbuilt keyword and must be specified at the start of every command. The middle part of the command is code that should be known to the sender that is unique code generated at remote device's application. Last part of the code is command itself that is an actual command like a ring, stop, silent, vibrate or switch off the device.

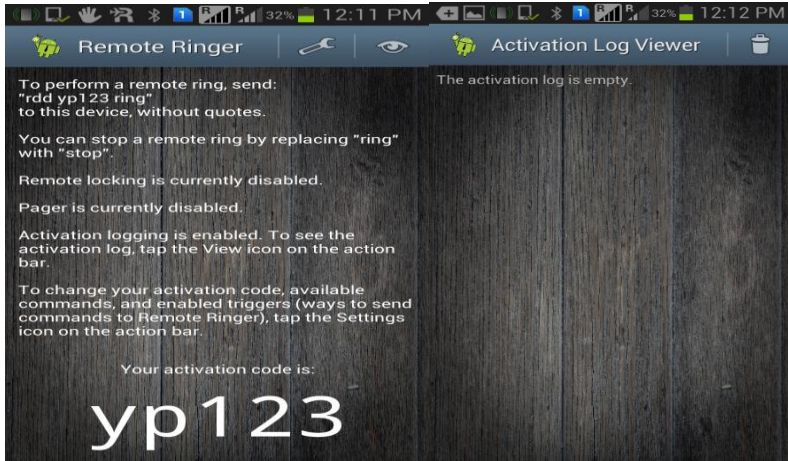


Figure 2. Home Page with instructions

Figure 3. Activation log page

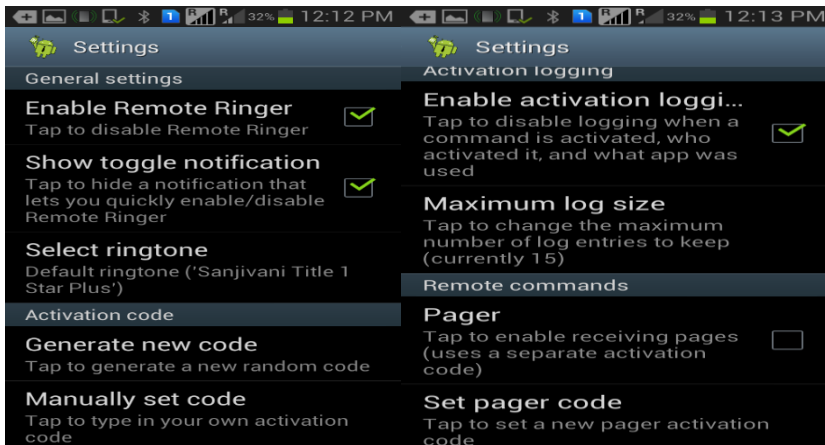


Figure 4. Activation log page

Figure 5. Setting page

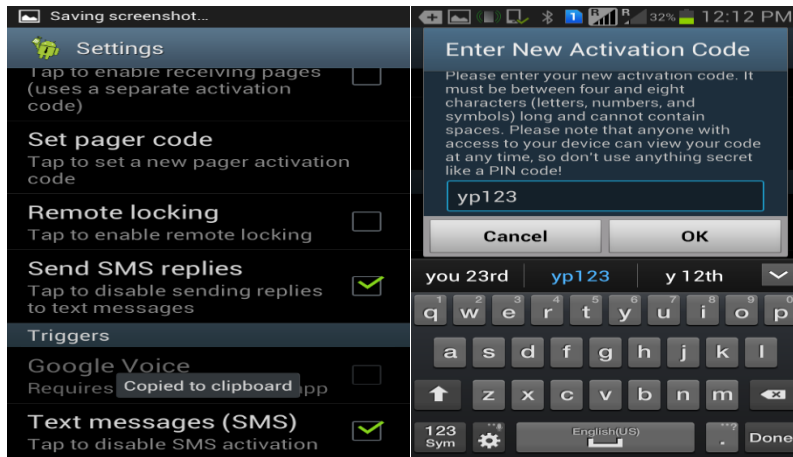


Figure 6. Other Settings page

Figure 7. To change Activation Code

When any command is sent to remote device, it will be received by receipt phone first via SMS and after receiving command as a text message, While installation of application it will get access to read a text message. It will match rdd keyword form text message and recognize it as application command now it will integrate text message into three different parts, first rdd for recognition of application command. The second part will be for unique code and last as a command to activate. Received unique code will be matched with application code for authorization of application code. Last part will invoke appropriate function and module to execute the command.

A flowchart of the proposed method is sketched in Fig.1. The process of sending command and execution are described subsequently.

The process of sending command and execution are described subsequently in the block diagram in fig.2.

- a) To enable or disable application following code is used
- ```
if (!preferencesManager.getEnabled())
 return false;
```

Application will switch on/off, when user checks the above code option.

- a) To get or set activation code following code is used

```
String code = preferencesManager.getCode ();
```

- b) As soon as sender sends the message, at receiver side message is split into the different part to identify that it's application command and perform operation accordingly. For this following code is used.

```
String [] messageTokens=message.split ("\\s+");
```

- c) Now different part of message is assigned to different variables

```
String pagerCode = preferencesManager.getPagerCode ();
```

```
if (messageTokens[0].compareToIgnoreCase(pagerCode) == 0)
```

```
{
 Intent intent = new Intent (ApiHandler.INTENT_RING);
 Intent logIntent = new Intent (LogHandler.INTENT_LOG);
 int prefixLength = pagerCode.length()+ 1;
 if (prefixLength < message.length())
```

```
{
 String pagerMessage = message.substring (prefixLength);
 intent.putExtra ("message", pagerMessage);
```

```
logIntent.putExtra ("argument", pagerMessage);
}
```

```
intent.putExtra ("source", source);
logIntent.putExtra ("command", "ring").putExtra ("app", "RemoteRinger").putExtra ("source", source);
```

```
context.sendBroadcast (logIntent);
```

```
context.sendOrderedBroadcast (intent, ApiHandler.PERMISSION_HANDLE_INTERNAL, resultReceiver,
null, ApiHandler.RESULT_UNKNOWN_COMMAND, null, null);
```

```
return true;
```

```
}
```

- d) For device API device and permission used following code

```
if((messageTokens[0].compareToIgnoreCase("RemoteRinger")==0||messageTokens[0].compareToIgnoreCase("RDD")== 0) && messageTokens[1].compareToIgnoreCase(code)
==0||messageTokens[0].compareToIgnoreCase(code) == 0) {Intent intent = new Intent().putExtra("source",
source);
```

```
String permission = ApiHandler.PERMISSION_HANDLE_INTERNAL;
```

## Conclusion

Using this android application you can locate your lost phone nearby you. This application controls some android modules to activate different commands. The API used to establish a connection between application and device of another system. A new and different android application method for ringing remote device or phone, even when the device is in silent mode is purposed. Improvement and suggestions for this application method are done by adding extra module code for switching off application via SMS and ring phone or give a notification on the phone when it is already switched off.

## Acknowledgements

I thank the group of teacher's from Computer Science and Engineering Department and a group of my department's senior students for their assistance in this application development.

## References

- Ginaya, G. (2018). The Balinese calendar system: From its epistemological perspective to axiological practices. *International Journal of Linguistics, Literature and Culture (IJLLC)*, 4(3), 24-37.
- Hoog, A. (2011). *Android forensics: investigation, analysis and mobile security for Google Android*. Elsevier.
- Jianan, W., Aimin, Z., & Hang, Z. (2015, May). An improved Android based industrial monitoring and locating system. In *Control and Decision Conference (CCDC), 2015 27th Chinese* (pp. 1361-1365). IEEE.
- Kate, P. B., Shinde, A. A., & Babar, P. S. REMOTE ACCESS AND CONTROL MOBILE PHONE USING SMS.
- Lei, Z., Hu, W., Li, H., & Yang, Z. (2013, July). Web-based remote networked control for smart homes. In *Control Conference (CCC), 2013 32nd Chinese* (pp. 6567-6571). IEEE.
- Lu, L., Li, Z., Wu, Z., Lee, W., & Jiang, G. (2012, October). Chex: statically vetting android apps for component hijacking vulnerabilities. In *Proceedings of the 2012 ACM conference on Computer and communications security* (pp. 229-240). ACM.
- Villan, A. G., & Jorba, J. (2013). Remote control of mobile devices in Android platform. *arXiv preprint arXiv:1310.5850*.