

RESEARCH ARTICLE

A Study on Mobile Hunt App using Android Studio***B. Usharani**

*Assistant Professor, Department of Computer Science and Engineering, KLEF, Andhra Pradesh, India.

Received on: 15/06/2017, Revised on: 01/08/2017, Accepted on: 17/08/2017**ABSTRACT**

Now a day's mobile is the major communication tool. Mobile Apps offers information and purpose. Examples for well known mobile apps are SMS, Contact list, calendar, mobile games, Calculator. Mobile Apps do not need internet Connection. Mobile Apps are either free or paid. This paper gives knowledge of how to start working on Android Studio and to develop an app in android studio and get it run on emulator.

Keywords: Android APP, Android Studio, Android SDK, Mobile Hunt, Smart Phone etc**INTRODUCTION**

Smart phones are become crazy now a day. As Smartphone became popular, operations like listening to music, watching videos, chatting and some can be moved from the computer to a phone now.

HISTORY OF SMARTPHONE'S & APP

Alexander Graham Bell is the inventor of the telephone. In, 1978 he made the first phone call.

The first hand-held mobile phone was demonstrated by John F. Mitchell and Dr. Martin Cooper of Motorola in 1973, using a handset weighing around 4.4 pounds (2 kg). In 1983, the Dyna TAC 8000x was the first commercially sold mobile phone at \$3,995^[2]

The first mobile phone call was made publicly by Martin Cooper of Motorola to Dr. Joel S. Engel of Bell Labs on the 3rd day of April 1973. ^[3] Devices that combined telephony and computing were first conceptualized by Nikola Tesla in 1909 and Theodore Paraskevagos in 1971 and patented in 1974, and were offered for sale beginning in 1993^[5] The short form for the "Application Software is "app". App is used offline. A mobile app is a computer program designed to run on mobile devices such as smart phones and tablet computers ^[1] The IBM Simon Personal Communicator (simply known as IBM Simon) was a handheld, touch -screen cellular phone and PDA designed and engineered by International Business Machines Corp. (IBM) and assembled under contract by Mitsubishi Electric Corp. BellSouth Cellular Corp. distributed the

Simon Personal Communicator in the United States between August 1994 and February 1995. The Simon Personal Communicator was the **first cellular phone** to include telephone and PDA features in one device. ^[4] In the mid-1980s, London-based computer company Psion released the Psion Organizer, largely considered to be the first successful personal digital assistant (PDA) and forefather to mobile computer.^[8] Simon was also able to send and receive faxes, e-mails and cellular pages. Simon featured many applications including an address book, calendar, appointment scheduler, calculator, world time clock, electronic note pad, handwritten annotations and standard and predictive stylus input screen keyboards ^[4] Smartphone's before Android, iOS and BlackBerry, typically ran on Symbian, which was originally developed by Psion ^[5] Radiolinja became the first network to offer a commercial person-to-person SMS text messaging service in 1994^[12].

1993: Mobile phones get SMS: Nokia is the first mobile phone that's able to send texts. But can only be sent between two people on the same network.

1994: SMS as broadcast: Vodafone — one of only two mobile networks in the U.K.

1997: Enter QWERTY: The Nokia 9000i Communicator becomes the first phone to come equipped with a keyboard.

1999: Worlds collide: Text messages finally cross networks for the first time, The first access to the mobile web was

commercially offered in Finland in 1996 on the Nokia 9000 Communicator phone via the Sonera and Radiolinja networks.^[13]

In 2000, The Ericsson R380 was the first device marketed as a “smart phone”.

In 2002 Palm Tero smart phone introduced the features to check the calendar while talking on the phone, dial directly from a list of contacts, and send e-mails.

In 2013 Apple iPhone introduced the features fingerprint recognition system built directly from the home button which can be used to unlock the phone.

ANDROID STUDIO

Android (based on the Linux kernel) is a mobile operating system developed by Google. In. Android was officially introduced via the release of its inaugural device, the HTC Dream (T-Mobile G1) on 20 October 2008^[5]. G1 is the first smart phone run on Linux based OS.

Android Studio is the official integrated development environment (IDE) for Android platform development. It is available for download on Windows, Mac OS X and Linux, and replaced Eclipse Android Development Tools (ADT) as Google's primary IDE for native Android application development.^[9]

Table 1: Version History of Android Studio ^[10]

Version	Downloads	Last updated	Status
Android Studio 2.0.0.20	38	Thursday, May 12, 2016	approved
Android Studio 1.5.1.0	1260	Wednesday, December 30, 2015	approved
Android Studio 1.5.0.0	22	Wednesday, December 30, 2015	approved
Android Studio 1.4.1.0	22	Wednesday, December 30, 2015	approved
Android Studio 1.4.0.10	196	Thursday, October 8, 2015	approved
Android Studio 1.3.1.0	45	Tuesday, August 18, 2015	approved
Android Studio 1.3.0.10	835	Tuesday, August 4, 2015	approved
Android Studio 1.2.1.1	49	Saturday, June 6, 2015	approved
Android Studio 1.1.0	975	Tuesday, February 24, 2015	approved
Android Studio 0.3.2.132893413	1269	Friday, January 3, 2014	
Android Studio 0.2.130.737825	300	Tuesday, July 23, 2013	
Android Studio 0.1.130.677228	217	Monday, May 27, 2013	

Create a Project with Android Studio

1. In Android Studio, create a new project:

- If you don't have a project opened, in the Welcome screen, click New Project.

• If you have a project opened, from the File menu, select New Project. The *Create New Project* screen appears.

2. Fill out the fields on the screen, and click Next.

It is easier to follow these lessons if you use the same values as shown.

- Application Name is the app name that appears to users. For this project, use "My First App."
- Company domain provides a qualifier that will be appended to the package name; Android Studio will remember this qualifier for each new project you create.
- Package name is the fully qualified name for the project (following the same rules as those for naming packages in the Java programming language). Your package name must be unique across all packages installed on the Android system. You can Edit this value independently from the application name or the company domain.
- Project location is the directory on your system that holds the project files.

3. Under Select the form factors your app will run on, check the box for Phone and Tablet.

4. For Minimum SDK, select API 8: Android 2.2 (Froyo).

5. Leave all of the other options (TV, Wear, and Glass) unchecked and click Next.

6. Under Add an activity to *<template>*, select Blank Activity and click Next.

7. Under Customize the Activity, change the Activity Name to *MyActivity*. The Layout Name changes to *activity_my*, and the Title to *MyActivity*. The Menu Resource Name is *menu_my*.

8. Click the Finish button to create the project.

Your Android project is now a basic "Hello World" app that contains some default files.

Review of the most important files and their description:

1. `app/src/main/res/layout/activity_my.xml`
2. This XML layout file is for the activity you added when you created the project with Android Studio. Following the New Project workflow, Android Studio presents this file with both a text view and a preview of the screen UI. The file contains some default interface elements from the material design library, including the `AppBar` and a floating action button. It also

- includes a separate layout file with the main content.
3. app/src/main/res/layout/content_my.xml
 4. This XML layout file resides in activity_my.xml, and contains some settings and a TextView element that displays the message, "Hello world!".
 5. app/src/main/java/com.mycompany.myfirs tapp/MyActivity.java
 6. A tab for this file appears in Android Studio when the New Project workflow finishes. When you select the file you see the class definition for the activity you created. When you build and run the app, the Activity class starts the activity and loads the layout file that says "Hello World!"
 7. app/src/main/AndroidManifest.xml
 8. The manifest file describes the fundamental characteristics of the app and defines each of its components. You'll revisit this file as you follow these lessons and add more components to your app.
 9. app/build.gradle
 10. Android Studio uses Gradle to compile and build your app. There is a build.gradle file for each module of your project, as well as a build.gradle file for the entire project. Usually, you're only interested in the build.gradle file for the module, in this case the app or application module. This is where your app's build dependencies are set, including the default Config settings:
 11. `compileSdkVersion` is the platform version against which you will compile your app. By default, this is set to the latest version of Android available in your SDK. (It should be Android 4.1 or greater; if you don't have such a version available, you must install one using the SDK Manager) You can still build your app to support older versions, but setting this to the latest version allows you to enable new features and optimize your app for a great user experience on the latest devices.
 12. `application Id` is the fully qualified package name for your application that you specified during the New Project workflow.
 13. `minSdkVersion` is the Minimum SDK version you specified during the New Project workflow. This is the earliest version of the Android SDK that your app supports.
 14. `targetSdkVersion` indicates the highest version of Android with which you have tested your application. As new versions of Android become available, you should test your app on the new version and update this value to match the latest API level and thereby take advantage of new platform features.
 15. Note also the `/res` subdirectories that contain the resources for your application:
 16. `drawable-<density>/`
 17. Directories for drawable resources, other than launcher icons, designed for various densities.
 18. `layout/`
 19. Directory for files that define your app's user interface like activity_my.xml, discussed above, which describes a basic layout for the MyActivity class.
 20. `menu/`
 21. Directory for files that define your app's menu items.
 22. `mipmap/`
 23. Launcher icons reside in the `mipmap/` folder rather than the `drawable/` folders. This folder contains the `ic_launcher.png` image that appears when you run the default app.
 24. `values/`
 25. Directory for other XML files that contain a collection of resources, such as string and color definitions.

RUN ON A REAL DEVICE

If you have a device running Android, here's how to install and run your app.

Set up your device

- Plug in your device to your development machine with a USB cable. If you're developing on Windows, you might need to install the appropriate USB driver for your device

- Enable USB debugging on your device. On Android 4.0 and newer, go to Settings > Developer options.
- Run the App From Android Studio
- Select one of your project's files and click Run from the toolbar.
- In the Choose Device window that appears, select the Choose a running device radio button, select your device, and click OK.

Android Studio installs the app on your connected device and starts it.

RUN ON THE EMULATOR

Whether you're using Android Studio or the command line, to run your app on the emulator you need to first create an Android Virtual Device (AVD). An AVD is a device configuration for the Android emulator that allows you to model a specific device.

Create an AVD

1. Launch the Android Virtual Device Manager:
 - In Android Studio, select Tools > Android > AVD Manager, or click the AVD Manager icon in the toolbar. The *AVD Manager* screen appears.
 - Or, from the command line, change directories to `sdk/` and execute: `tools/android avd`
2. On the AVD Manager main screen, click Create Virtual Device.
3. In the Select Hardware window, select a device configuration, such as Nexus 6, and then click Next.
4. Select the desired system version for the AVD and click Next.
5. Verify the configuration settings, and then click Finish.

Run the app from Android Studio

1. In Android Studio, select your project and click Run from the toolbar.
2. In the Choose Device window, click the Launch emulator radio button.
3. From the Android virtual device pull-down menu, select the emulator you created, and click OK.

It can take a few minutes for the emulator to load itself. You may have to unlock the screen. When you do, *My First App* appears on the emulator screen.

GENERATE APKs

When you click **Run 'app'**, Android Studio generates a debug APK and deploys it to your

target device. To locate the generated APK files, click the link in the pop-up dialog, as shown in figure 1.



Figure 1: Click the link to locate the generated APK file ^[14]

EXISTING SYSTEM

In existing system, the traditional methodologies are used to track the mobile like IMEI (International Mobile Station Equipment Identity) is a number, usually unique and it a 15 digit number which is only used by Cybercrime department for the identification of any person who involved in any crimes. But the normal users cannot have the access to IMEI number. Based on cellular network signals you can track their mobile. It is also not helpful to the normal users. The existing system is using GCM push notifications.

GCM push Notifications:

Google Cloud Messaging (GCM) means the lost mobile can be traced using a web interface. It is a free service that enables developers to send messages between servers and client apps.

In GCM, the communication is confined to only web interface. If the user wants to track the mobile he has to use particular web interface. The main disadvantage is the user cannot access website whenever he want.

PROBLEMS WITH EXISTING SYSTEM

- IMEI number are used, cellular network signals to track the lost mobile which is a very time consuming process and have no use for normal users.
- Based on GCM (Google Cloud Messaging) push notifications also you can track their mobile but it is not helpful for freelancers and travelers.
- This conventional method of tracing the mobile is no faster.
- There is no matter of security for user data like ATM pin numbers, bank account numbers, credit card details, passwords in the lost mobile.

PROPOSED SYSTEM

An android application is developing, which shows a slight change from the desktop environment.

This "MOBILE HUNT" application helps in tracing the mobile easily with the help of SMS

commands.

These SMS commands will be useful for the user to trace the device and control the mobile remotely.

“MOBILE Hunt” application has certain features like:

- New SIM Detection
- Locate the device through Google Device Manager
- Lock The Mobile remotely.
- Geocode Response
- Root the application
- SMS Commands: Through SMS Commands, there is the possibility to LOCK the device remotely, WIPE the confidential data, enable or disable the mobile DATA, GPS, WI-FI with the help of SMS commands.
- SMS Command DELSDCARD will clear the external memory.
- CALL XXXXXXXXXXXX will remotely call a specified number.

CONCLUSION

Mobile Apps became an essential part in day to day life .Our objective behind this paper is, to discuss all basic facts to start an android application using android studio and to overcome the technical problems which come as a big constraint to the beginner's. In this paper a study is conducted on Smartphone's and their

applications. In future research, Mobile Hut App is implemented on Android Studio.

REFERENCES

1. https://en.wikipedia.org/wiki/Mobile_app
2. <https://www.ding.com/community/history-of-mobile-apps>
3. <https://www.devsaran.com/blog/mobile-application-development-insight>
4. https://en.wikipedia.org/wiki/IBM_Simon
5. <https://en.wikipedia.org/wiki/Smartphone>
6. https://en.wikipedia.org/wiki/Windows_Phone
7. https://en.wikipedia.org/wiki/Mobile_operating_system
8. <http://www.investopedia.com/articles/personal-finance/062215/who-invented-apps-smart-phones.asp>
9. https://en.wikipedia.org/wiki/Android_Studio
10. <https://chocolatey.org/packages/AndroidStudio>
11. <http://www.androidauthority.com/android-studio-tutorial-beginners-637572>
12. https://en.wikipedia.org/wiki/Text_messaging
13. https://en.wikipedia.org/wiki/Mobile_Web
14. <https://developer.android.com/studio/intro/index.html>