3. Features of Android

Android is a powerful Operating System supporting a large number of powerful applications. Hardware that supports Android is mainly based on ARM Architecture. The main feature of Android application includes:

3.1 Interface

The default interface of Android is based on direct manipulation. It uses the real world touches like tapping, swiping, pinching and reverse pinching and a virtual keyboard. It uses the user touches as an input to perform many functions or different tasks and provides a fluid touch interface. The hardware contained in the devices like accelerometers, gyroscopes and proximity sensor are used by the applications to perform various actions like to adjust the screen from portrait to landscape, turning off the screen during phone call and steering of the car during games.

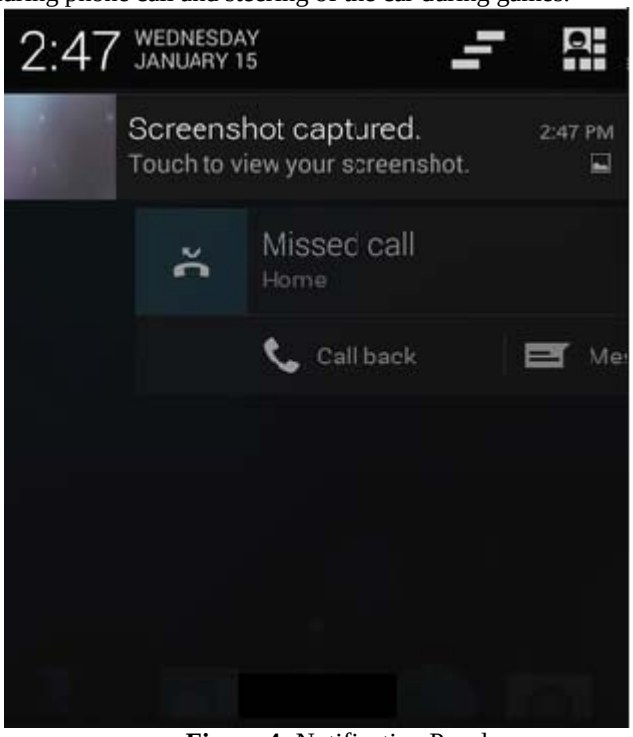

Figure 4: Notification Panel

The homescreen of Android are usually made up of widgets and app icons. App icons launch the application whereas widget shows live, auto updating features of the homescreen such as weather forecast, clock, currency converter, etc. A homescreen also comprises of several pages and the android's homescreen highly customisable allowing the user to texture the face of the homescreen. Third-party call from Google Play can enhance the homescreen to great extent and can also mimic the look of other operating system. Most mobile manufacturers make their own custom rom and differentiate the look for competition in the globe. Each mobile company has its own style of the user interface.

The most common look that almost every manufacturer has copied is the status bar which indicates the battery level, the network, operator name, etc. When the status bar is pulled down it reveals the notification screen which has newly updates, emails, SMS, toggles buttons, etc. It sees to that the notification does not immediately interrupt or inconvenience the user. After Android 4.1, the notification has expanded its features with adding music player in the notification screen and other toggle buttons.

3.2 Applications

Applications now a days called as "Apps" are made for android and are mainly based in JAVA programming language using the Android Software Development Kit (SDK). Software Development Kit (SDK) includes main contains of Android which includes a debugger, software libraries, a handset emulator that is based on QEMU, sample codes, documentation and tutorials. Eclipse is the official Integrated Development Environment (IDE) using Android Development Tools (ADT) plugin. Other developments tools include, Native Development Kit for applications, Google App Inventor, etc. In January 2014, Google came up with an Apache Cordova – based framework for porting Chrome HTML 5 applications to Android, covered in a native application shell.

The android has been famous for its third party application which can be installed by using APK's file and installing it apart from Google Play store. Or else downloading it through a system which allows installing, updating and removing applications from their devices. Google Play is the primary place to install the application for Android and it also consist third party applications too. It was recorded in July 2013 that there more than one million applications available in Google Play for Android and 50 billion applications have been installed from Google Play.

Due to the open nature of Android, a number of third-party application marketplace also exist for Android, either to provide a substitute for devices that are not allowed to ship with Google Play Store, provide applications that cannot be offered on Google Play Store due to policy violations, or for other reasons. Examples of these third-party stores have included the Amazon Appstore, GetJar, and SlideMe. F-Droid, another alternative marketplace, seeks to only provide applications that are distributed under free and open source licenses.

3.3 Memory Management

Android operating works on high performance system
hence it is designed to maintain and manage RAM to keep power consumption at minimum in contrast to desktop operating system which is always connected to electricity. Whenever any app is no longer in any use, the system will automatically cleans it in memory while technically is still open. These suspended apps do not require sources like memory and power rather they stay in the background in the idle state until required again. This has new feature that the apps need not to be closed and can be opened again and thus opens the gate to multitasking and it all ensures power efficiency and memory management.

Figure 5: RAM Management in Android (Third Party)

Android manages the apps stored in memory automatically: when memory is low, the system will begin killing apps and processes that have been inactive for a while, in reverse order since they were last used (oldest first). This process is designed to be invisible to the user, such that users do not need to manage memory or the killing of apps themselves. However, confusion over Android memory management has resulted in third-party task killers becoming popular on Google Play store; these third-party task killers are generally regarded as doing more harm than good.

4. Android Architecture

Android is an open-source software architecture provided by the Open Handset Alliance , a group of 71 technology and mobile companies whose objective is to provide a mobile software platform. The Android platform includes an operating system, middleware and applications. As for the features, Android incorporates the common features found nowadays in any mobile device platform, such as: application framework reusing, integrated browser, optimised graphics, media support, network technologies, etc.

The Android architecture, depicted in Figure 1, is composed by five layers: Applications, Application Framework, Libraries, Android Runtime and finally the Linux kernel.
The uppermost layer, the Applications layer, provides the core set of applications that are commonly offered out of the box with any mobile device.

The Application Framework layer provides the framework Application Programming Interfaces (APIs) used by the applications running on the uppermost layer. Besides the APIs, there is a set of services that enable the access to the Android's core features such as graphical components, information exchange managers, event managers and activity managers, as examples.

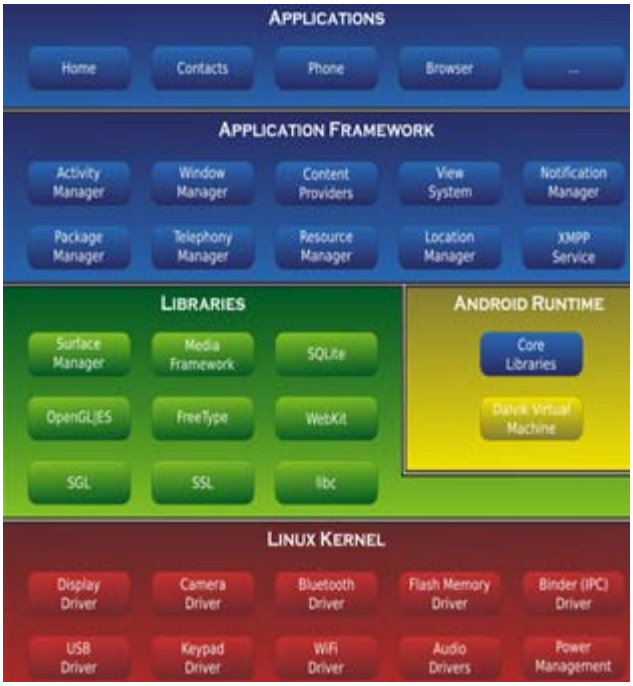

Figure 6: Android Architecture

Below the Application Framework layer, there is another layer containing two important parts: Libraries and the Android Runtime. The libraries provide core features to Android Architecture the applications. Among all the libraries provided, the most important are libc, the standard C system library tuned for embedded Linux-based devices; the Media Libraries, which support playback and recording of several audio and video formats; Graphics Engines, Fonts, a lightweight relational database engine and 3D libraries based on OpenGL ES.

Regarding the Android Runtime, besides the internal core libraries, Android provides its own VM, as previously stated, named Dalvik. Dalvik was designed from scratch and it is specifically targeted for memory-constrained and CPU constrained devices. It runs Java applications on top of it and unlike the standard Java VMs, which are stack-based; Dalvik is an infinite register-based machine. Being a register machine, it presents two advantages when compared to stack-based machines. Namely, it requires 30% less instructions to perform the same computation as a typical stack machine, causing the reduction of instruction dispatch and memory access; and less computation time, which is also derived from the elimination of common expressions from the instructions. Nevertheless, Dalvik presents 35% more bytes in the instruction stream than a typical stack-machine.

This drawback is compensated by the consumption of two bytes at a time when consuming the instructions. Dalvik uses its own byte-code format name Dalvik Executable (.dex), with the ability to include multiple classes in a single file. It is also able to perform several optimizations during dex generation when concerning the internal storage of types and constants by using principles such as minimal repetition; per-type pools; and implicit labeling. By applying these principles, it is possible to have dex files smaller than a typical Java archive (jar) file. During install time, each dex file is verified and optimizations such as byte swapping and padding, static-linking and method in-lining are performed in

order to minimize the runtime evaluations and at the same time to avoid code security violations.

The Linux kernel, version 2.6, is the bottom most layer and is also a hardware abstraction layer that enables the interaction of the upper layers with the hardware layer via device drivers. Furthermore, it also provides the most fundamental system services such as security, memory management, process management and network stack.

5. Basic Hardware

Considering the hardware requirements main hardware platform for Android is the 32-bit ARMv7 architecture. But in 2012, Intel processors began to appear on more mainstream Android platforms, such as phones. In 2013, Freescale announced support for Android on its i.MX processor, specifically the i.MX5X and i.MX6X series.

As the Android 5.0 Lollipop is released officially with the Nexus 7 and new Motorola X, so as far as the official requirement stated for Android 4.4 is at least 512 MB of RAM, while for "low RAM" devices 340 MB is the required minimum and standard android OS requirement that does not include memory dedicated to various hardware components such as the baseband processor. Android 4.4 requires a 32-bit ARMv7, MIPS or x86 architecture processor (latter two through unofficial ports),together with an OpenGL ES 2.0 compatible graphics processing unit (GPU).Android supports OpenGL ES 1.1, 2.0 and 3.0. Some applications explicitly require a certain version of the OpenGL ES, thus suitable GPU hardware is required to run such applications.
In addition to running on smartphones and tablets, it is possible to run Android natively on regular PC hardware with versions of Android, including 4.4, downloadable from the Android-x86 project. Chinese companies are building a PC and mobile operating system, based on Android, to "compete directly with Microsoft Windows and Google Android". The Chinese Academy of Engineering noted that many companies are now customizing Android following a Chinese ban on the use of Windows 8 on government PCs.

6. Official Integrated Development Environment

6.1 Eclipse Software

In Computer Programming Eclips is one of the software used widely. It has mainly base workspace and extensible plug-in system for customizing requirement. Eclipse can be used to developed java application with the help of various plugins. Eclips also suppoerts different programming languages like Ada, ABAP, C, C++, COBOL, Fortran, Haskell, JavaScript, Lasso,Natural, Perl, PHP, Prolog, Python, R, Ruby Scala, Clojure, Groovy, Scheme, and Erlang.

Eclipse software development kit (SDK) which includes the java development tools, it developed for Java developer. Users can extend its abilities by installing plug-ins written for the Eclipse Platform, such as development toolkits for other programming languages, and can write and contribute their own plug-in modules.

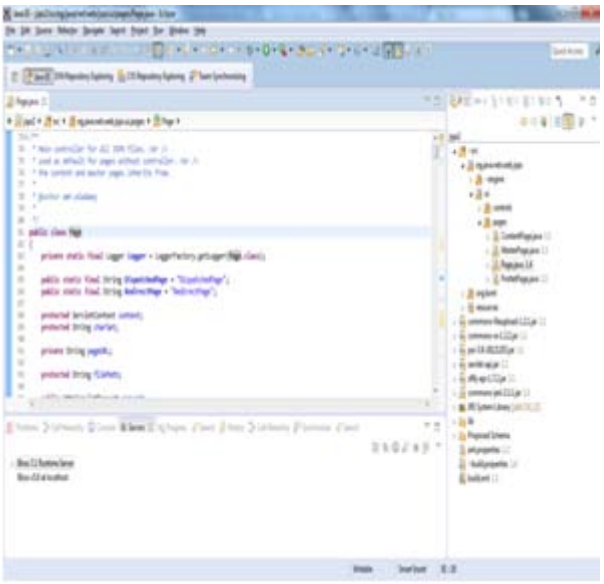

Figure 7: Eclipse Software

Eclipse uses plugging that give all the functionality on top of runtime system. Its runtime system is based on Equinox, an implementation of the OSGi core framework specification. The eclipse SDK includes the Eclipse JAVA Development tools (JDT) this offers an IDE features like built in incremental java compiler and full mode of JAVA source files.

Eclipse implements uses the graphical control elements of the Java toolkit called SWT, whereas most Java applications use the Java standard Abstract Window Toolkit (AWT) or Swing. Eclipse's user interface also uses an intermediate graphical user interface layer called JFace, which simplifies the construction of applications based on SWT. Eclipse was made to run on Wayland during a GSoC-Project in 2014.

New Android projects, create an application UI, add packages based on the Android Framework API, debug their applications using the Android SDK tools, and export signed (or unsigned) .apk files in order to distribute their applications. It is a freeware available to download.

As shown in the figure, we can create an APP with the help of Eclipse environment. All programming features are made available with the help of installing required plug-in as well as libraries. Final output will be application file which uses data abstraction to hide information from user interface only showing what is using by user.

1. Eclipse PDT (PHP Development Tools)
The PHP (PHP hypertext Preprocessor) Development Tools project provides a PHP Development Tools framework for the Eclipse platform. The project encompasses all development components, including code-completion, develop PHP and facilitate extensibility. It leverages the existing Eclipse Web Tools Platform (WTP) and Dynamic Languages Toolkit (DLTK).

2. Eclipse ADT (Android Development Tools)
Android Development Tools (ADT) is a plug-in for the Eclipse IDE that is designed to provide an integrated

environment in which to build Android applications. ADT extends the capabilities of Eclipse to let developers set up.

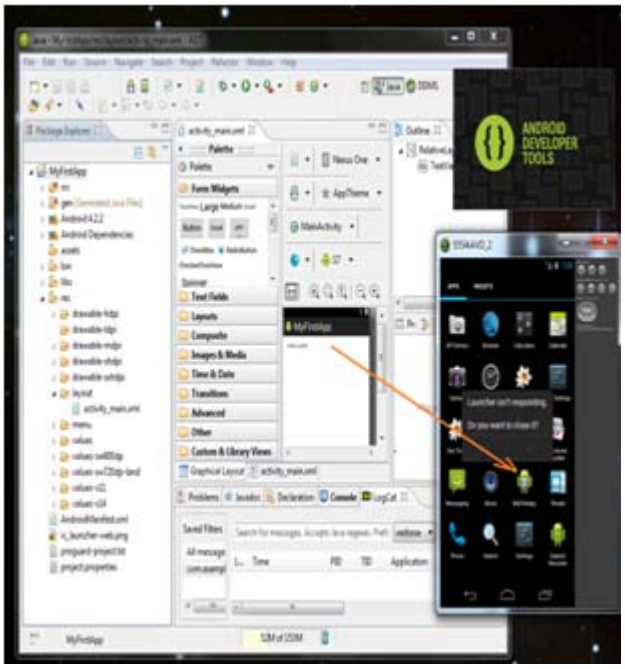

Figure8. Eclipse Software with Emulator

7. Conclusion

Thus, through this paper we can understand the Android operating system in deep detail. We understood the origin of Android and how the Android came into existence. Its open features for all mobile manufacturers to use it and third party software developers to develop millions of software for the Android platform. The open source is the key for its development in the mobile and other industries. The architecture of Android is explained in detail with explanation all layers. The paper also successfully pin points the features of Android in terms of memory management. The operating system forms the base for the billions of mobile devices in today's market with millions of Android applications.

References

[1] *G. Marcario, M. Torchiano, and M. Violante*, **"An in-vehicle infotainment software architecture based on google android"** in SIES. Lausanne, Switzerland: IEEE, July 2009, pp.257-260.

[2] *L.Nogueira and L.M. Pinho*, **"Time-bounded distributed qos-aware service configuration in heterogeneous cooperative en-vironments,"** Journal of Parallel and Distributed Computing vol.69, no. 6, pp. 491-507, June 2009.

[3] *M.T. Higuera –Toledano and V. Issarny*, **"Java embedded real time systems: An overview of existing solutions,"** in Proceedings of the Third International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC 2000). Washington, D.C, USA: IEEE Computer Society, 2000,pp. 392-399.

[4] *R. Guerra, S. Schorr, and G. Fohler*, **"Adaptive resource management for mobile terminals – the actors approach,"** in Proceedings of 1st Workshop on Adaptive Resource Management (WARM10), Stockholm, Sweden, April 2010.

[5] *C.Maia, L. Nogueira, and L. M. Pinho*, **"Experiences on the implementation of a cooperative embedded system framework,"** CISTER Research Centre, Porto, Portugal, Tech. Rep., Jun '10.

[6] *C. L. Liu and J. W. Layland*, **"Scheduling algorithms for multiprogramming in a hard-real time environment,"** J. ACM, vol. 20, no. 1, pp. 46-61, 1973.

[7] *D. Faggioli, M. Trimarchi, and F. Checconi*, **"An implementation of the earliest deadline first algorithm in linux,"** in Proceedings of the 2009 ACM Symposium on Applied Computing (SAC09). New York, NY, USA: ACM, 2009, pp. 1984- 1989.

[8] *Mohd Shahdi Ahmad, Nur Emyra Musa, Rathidevi Nadarajah, Rosilah Hassan and Nor Effendy Othman*," **Comparison Between Android and iOS Operating System in terms of Security"** School of Computer Science,Faculty of Information Science & Technology, Universiti Kebangsaan Malaysia, 43600 Bangi, Selangor, Malaysia 2013 8th International Conference on Information Technology in Asia (CITA)

Author Profile

Tehseen Bano Mohd Anis was born in Mumbai, Maharashtra district, India. She received B.E. degree in electronics from Rizvi College of Engineering (RCOE) affiliated to Mumbai University in May-2012 and currently doing ME in Electronics from Shah & Anchor Kutchhi Engineering College affiliated to Mumbai University.

Subha Subramaniam was born in Nagercoil, Kanyakumari district, TamilNadu, India. She received B.E degree in Electronics and Communication Engineering from Govt. College of Engineering (GCE), Tirunelveli affiliated to Anna University in July 1998 and M.E degree in Electronics Engineering from Thadomal Shahani Engineering College (TSEC), Mumbai affiliated to Mumbai University in Nov 2003. She is currently pursuing Ph.D in Electronics Engineering from Veeramata Jijabai Technological Institute (VJTI), Mumbai. Presently she is working as Associate professor, Department of Electronics at Shah & Anchor Kutchhi Engineering College (SAKEC), Mumbai. Her research interest is in the area of Nanoelctronics.