



East West University

Faculty of

Electronics and Communication Engineering

SSC ENGLISH SOLUTION - An Android Application

B.Sc Project on Electronics & Telecommunication Engineering

Submitted By

MD.FAKHRUL ISLAM

&

MD.MAHAFUZER RAHMAN

ID: 2010-3-55-016

ID: 2010-3-55-011

Supervisor

Mustafa Mahmud Hussain

Assistant Professor

Electronics & Communication Engineering Department



Supervisor's Statement

Hereby I confirm that the present project was prepared under my supervision and that is fulfills all the requirements for the Bachelor degree of Electronics & Telecommunication Engineering.

.....
.....

Date

Supervisor's Signature

Author's Statement

Hereby We Declare that the present project was prepared by us and none of its contents was obtained by means that are against the law.

The project has never before been a subject of any procedure of obtaining an academic degree.

Moreover, We declared that the present version of the project is identical.

.....

MD.FAKHRUL ISLAM

.....

MD.MAHAFUZER RAHMAN

Acknowledgement

First we would like to express our sincere gratitude to Allah that we could complete our project. Then we are externally grateful to parents for their constant support through put my academic work.

We are extremely fortunate towards our supervisor Mustafa Mahmud Hussain sir for helping us throughout the project. He helped us in every possible way for completing our project.

We are also thankful to our classmates for inspiring us and helping us to choose mobile application development as a project.

Abstract

This paper describes the design and implementation of android BCS QUIZ application. The main objective of the thesis was to discover development in the android Platform and produce a prototype BCS QUIZ application. The theoretical part of this paper discusses the android platform, its architecture and the development process. The thesis can be used as a reference to carry out further developments in android.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

08

CHAPTER 2

2.1.1 HISTORY

09

2.1.2 ANDROID BOOTUP PROCESS

13

2.1.3 LINUX KERNEL

14

2.1.4 INIT PROCESS

14

2.1.5 ZYGOTE

14

2.1.6 RUNTIME PROCESS

15

2.2.1 INTERFACE

16

2.2.2 ANDROID SOFTWARE DEVELOPMENT

17

2.2.3 ANDROID ADAPTION FOR MOBILE

18

CHAPTER 3

3.1 WHY ANDROID

20

3.1.1 MARKET GROWTH

21

3.1.2 TECHNICAL CAUSES

22

CHAPTER 4

4.1 ANDROID ARCHITECTURE

24

CHAPTER 5

5.1 APPLICATION FRAME WORK

28

5.1.1 ANDROID LIBRARY

28

5.1.2 C/C++ LIBRARY

29

5.1.3 ACTIVITY

30

5.1.4 VIEW

33

5.1.5 XML

33

5.1.6 INTENT

34

5.2 ANDROID MANIFEST

34

CHAPTER 6

6.1 SNAPSHOT AND OVERVIEW OF SSC ENGLISH SOLUTION APP

35

CHAPTER 7

7.1 SSC ENGLISH SOLUTION JAVA CODE FOR APP

43

7.2 ANDROIDMANIFEST.XML & HTML

44

7.3 BIBLIOGRAPHY

50

Chapter 1

Introduction:

In recent years mankind has witnessed a revolution in the smart phone industry and emerging growth in the usage of mobile applications that range from entertainment and educational apps to simple games, health care apps and more. The concept “mobile applications” has become very central lately and has indeed turned into a great impact on our everyday life. Almost everyone has some kind of mobile device like a mobile phone or smart phone and to use it in the everyday life is understood. The way we are using our mobile devices have changed and the expectations on those devices have increased. To summarize one can say that the

needs and requests from the users’ are quite different compared to the situation a few years ago. Today when using a mobile device we more or less are taking for granted, that such a device is able to both enter the Internet and run applications like e-mail programs or text editors. One can assert that the mobile phone has become a mobile multimedia device, letting the user to do almost everything possible on a desktop. Not only have the devices been changing regarding the hardware, but also the development of software and applications for mobile

devices have gone through a change. Therefore the concept mobile applications including development and the possibilities of using mobile devices in everyday life situations, is very interesting and topical. Android has been with us in one form or another for more than six years. During that time, we've seen an absolutely breathtaking rate of change unlike any other development cycle that has ever existed. When it came time for Google to dive in to the smartphone wars, the company took its rapid-iteration, Web-style update cycle and applied it to an operating system, and the result has been an onslaught of continual improvement. Lately, Android has even been running on a previously unheard of six-month development cycle, and that's slower than it used to be. For the first year of Android’s commercial existence, Google was putting out a new version every two-and-a-half months.

Chapter 2

2.1.1 History

Android is the most widely used mobile operating system and, as of 2014, the highest selling OS overall. Android devices sell more than Windows, iOS, and Mac OS X devices combined with sales in 2012, 2013 and 2014 close to the installed base of all PCs. As of July 2013 the Google Play store has had over 1 million Android apps published, and over 50 billion apps downloaded. A developer survey conducted in April–May 2013 found that 71% of mobile developers develop for Android. At Google I/O 2014, the company revealed that there were over 1 billion active monthly Android users, up from 538 million in June 2013.

Android's source code is released by Google under open source licenses, although most Android devices ultimately ship with a combination of open source and proprietary software. Initially developed by Android, Inc., which Google backed financially and later bought in 2005, Android was unveiled in 2007 along with the founding of the Open Handset Alliance—a consortium of hardware, software, and telecommunication companies devoted to advancing open standards for mobile devices.

Android is popular with technology companies which require a ready-made, low-cost and customizable operating system for high-tech devices. Android's open nature has encouraged a large community of developers and enthusiasts to use the open-source code as a foundation for community-driven projects, which add new features for advanced users or bring Android to devices which were officially released running other operating systems. The operating system's success has made it a target for patent litigation as part of the so-called "smartphone wars" between technology companies.

Android Version History

a Visual Timeline

Since 2008, Android has released 39 version updates and 18 API levels, commercially. Here we've captured the major milestones.



The Beginning

1.0

- Android Market
- Camera support
- Gmail app & sync
- Contacts sync w/ People app
- Calendar sync w/ Calendar app
- Google Maps, Sync, Search, & Talk
- IM, SMS & MMS
- Media Player
- Custom notifications & wallpapers
- Voice Dialer
- YouTube player
- Wi-Fi & Bluetooth support

2008

1.5 - CUPCAKE

- Support for widgets
- Soft-keyboard w/ text-prediction
- Bluetooth A2DP, AVRCP support
- Record/watch videos
- Copy & paste in browser
- Animated screen transitions
- Auto-rotation
- New stock boot animation
- Upload videos to YouTube
- Upload photos to Picasa

SEP 23

2009

1.6 - DONUT

- Quick Search Box
- Gallery, camera, camcorder integration
- CDMA/EVDO, 802.1x & VPNs support
- Battery usage indicator
- Text-to-speech engine
- Multilingual speech synthesis engine
- Easier search, app screenshots in Market
- WVGA screen resolution support
- New Gestures Builder dev tool
- New framework APIs

APR 30

SEP 15

2.0 - ÉCLAIR

- Multiple accounts
- Turn-by-turn navigation for Google Maps
- Email & contact sync
- Exchange email support
- Quick Contact
- Bluetooth 2.1 support
- Search for saved SMS & MMS
- Live wallpapers
- New camera features
- Browser support for HTML5
- New framework APIs
- Optimized hardware speed
- Multiple screen sizes & resolutions support

OCT 26

2010



Android 5.0 "Lollipop" is the latest version of the Android mobile operating system



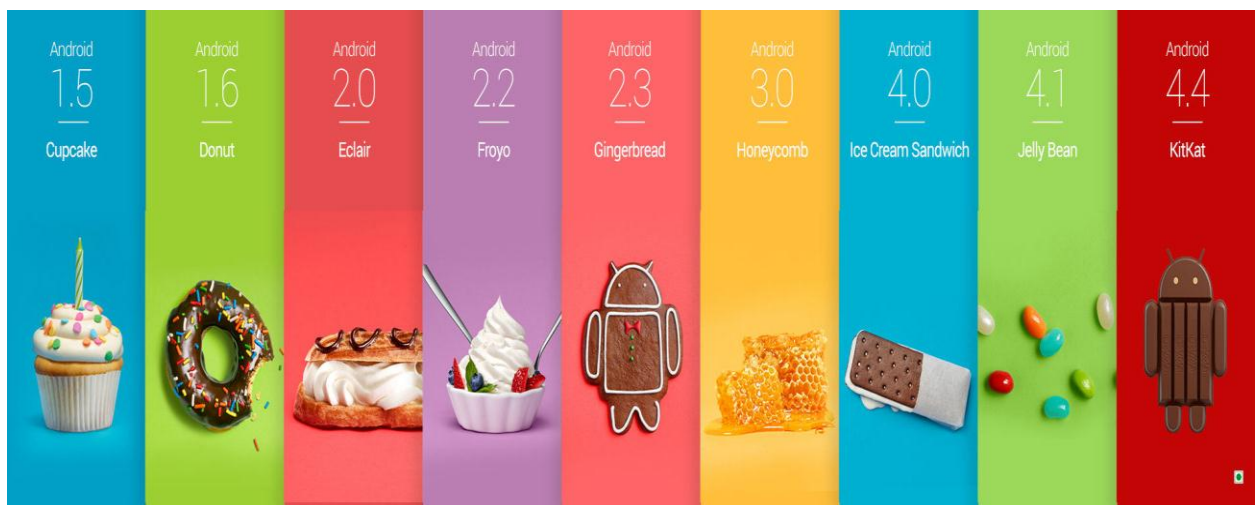
Lollipop: Android 5.0 introduces a refreshed notification system. Individual notifications are now displayed on cards to adhere to the material design language, and batches of notifications can be grouped by the app that produced them. Notifications are now displayed on the lock screen as cards, and "heads up" notifications can also be displayed as large banners across the top of the screen, along with their respective action buttons. A do-not-disturb feature is also added for notifications. The recent apps menu was redesigned to use a three-dimensional stack of cards to represent open apps. Individual apps can also display multiple cards in the recent menu, rather than only one entry per app; for example, a web browser can show all of its open tabs as individual cards.

Lollipop also contains major new platform features for developers, with over 5,000 new APIs added for use by applications. Additionally, the Dalvik virtual machine was officially replaced by Android Runtime (ART), which is a new runtime environment that was introduced as a technology preview in KitKat. ART is a cross-platform runtime which supports the x86, ARM, and MIPS architectures in both 32-bit and 64-bit environments. Unlike Dalvik, which uses just-in-time compilation (JIT), ART compiles apps upon installation, which are then run exclusively from the compiled version from then on. This technique removes the processing overhead associated with the JIT process, improving system performance.

Lollipop also aims to improve battery consumption through a series of optimizations known as "Project Volta". Among its changes are a new battery saver mode, job scheduling APIs which can restrict certain tasks to only occur over Wi-Fi, and batching of tasks to reduce the overall amount of time that internal radios are active.

The new developer tool called "Battery Historian" can be used for tracking battery consumption by apps while in use. The Android Extension Pack APIs also provide graphics functions such as new shaders, aiming to provide PC-level graphics for 3D games on Android devices.

A number of system-level, enterprise-oriented features were also introduced under the banner "Android for Work": Samsung contributed its Knox security framework for segregating personal and work-oriented data from each other on a device, along with accompanying APIs for managing the environment. Devices can also be configured so that users do not have to unlock their device with a PIN or pattern if it is within a trusted, physical location, or is in proximity to a user's Android Wear device. Device encryption will be enabled by default.



2.1.2 ANDROID BOOTUP PROCESS:

Power On: Master boot record (MBR) is a boot sector which contains partition table which has the information about how the device is partitioned in a structure. There is no MBR or partition when the device is started for the first time. When the phone is switched on, CPU will be in a no initialization state. Internal RAM is available and no internal clocks are set up. The device starts executing code located in the ROM and finds a specific block which has first Stage boot loader. The first boot loader points to a second stage boot loader, which is located in a known block. This "pointing" process is called raw partition table

Boot loader: Boot loader is a code which is executed before android operating system runs. It loads kernel to the RAM and sets up the initial memories.

Manufacturers use existing boot loaders or they create their own boot loaders. The First stage boot loader will find and setup the external RAM. Now Main boot loader is loaded and placed in external RAM as the RAM is available. The First important

program is in the second boot loader stage which contains code for file systems, additional memory and network support etc. When the boot loader is done it goes to the linux kernel [6].

2.1.3 Linux kernel

A kernel acts as a bridge between hardware and software. It setups cache protected memory, scheduling and loads drivers. After initializing Memory management units and caches, virtual memory can be used and user space processes can be launched by the system. After finishing the setup Kernel looks for init process which can found under system/core/init and launch it.

2.1.4 Init process

This process is the root process. Every process will be launched from this process. Init process mounts directories like /sys, /dev, /proc. It will run init.rc script and system service processes. This script is located in system/core/rootdir in the Android open source project and describes system services, file system and other parameters .

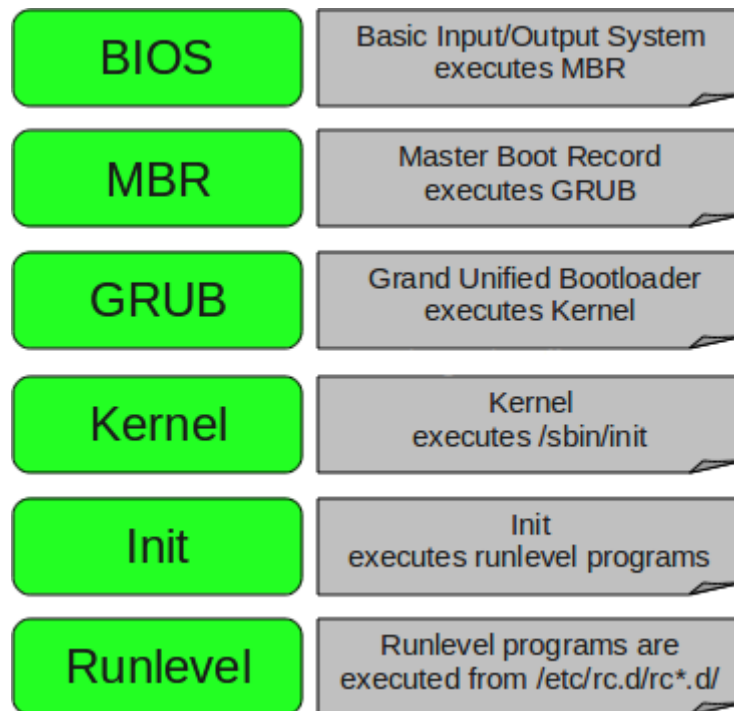
2.1.5 Zygote

After starting various daemons like Android Debug Bridge (adb), Radio Interface Layer Daemon (rild), etc, Init process initiates a process called Zygote. In java there is a separate instance of a Virtual Machine for each application. In android Dalvik, virtual machine is used as VM. So there is high consumption of memory and time because of different instances of dalvik VM for every application. Now Zygotes comes into play. It enables shared code across Dalvik VM, lower memory footprint and minimal startup time. Zygote process starts at system boot up and it preloads and initializes core library classes. After initialization, zygote process waits for socket request coming from the runtime process. If any request comes then it forks starts processes with VM instances.

2.1.6 Runtime process

The next init initiates the Runtime process and this process starts the service manager. All the services should be registered with the service manager and it provides local lookup service and binds services given their name. The runtime requests zygote to start system server process. Zygote splits and starts up a new dalvik vm instance and

starts the service. To control display device and audio output device the system server starts surface flinger and audio flinger. These services get registered with the service manager so that other applications can use display and audio. Now the system server will start all the core platform services and hardware services like activity manager, window manager and power manager etc. All of these services will get registered with the service manager



2.2.1 Interface

Android's default user interface is based on direct manipulation , using touch inputs, that loosely correspond to real-world actions, like swiping, tapping, pinching, and reverse pinching to manipulate on-screen objects, and a virtual keyboard. The response to user input is designed to be immediate and provides a fluid touch interface, often using the vibration capabilities of the device to provide haptic feedback to the user. Internal hardware such as accelerometers, gyroscopes and proximity sensors are used by some applications to respond to additional user actions, for example adjusting the screen from portrait to landscape depending on how the

device is oriented, or allowing the user to steer a vehicle in a racing game by rotating the device, simulating control of a steering wheel.

Android devices boot to the homescreen, the primary navigation and information point on the device, which is similar to the desktop found on PCs. Android homescreens are typically made up of app icons and widgets; app icons launch the associated app, whereas widgets display live, auto-updating content such as the weather forecast, the user's email inbox, or a news ticker directly on the homescreen. A homescreen may be made up of several pages that the user can swipe back and forth between, though Android's homescreen interface is heavily customisable, allowing the user to adjust the look and feel of the device to their tastes. Third-party apps available on Google Play and other app stores can extensively re-theme the homescreen, and even mimic the look of other operating systems, such as Windows Phone. Most manufacturers, and some wireless carriers, customise the look and feel of their Android devices to differentiate themselves from their competitors.

Present along the top of the screen is a status bar, showing information about the device and its connectivity. This status bar can be "pulled" down to reveal a notification screen where apps display important information or updates, such as a newly received email or SMS text, in a way that does not immediately interrupt or inconvenience the user. Notifications are persistent until read (by tapping, which opens the relevant app) or dismissed by sliding it off the screen. Beginning on Android 4.1, "expanded notifications" can display expanded details or additional functionality; for instance, a music player can display playback controls, and a "missed call" notification provides buttons for calling back or sending the caller an SMS message.

Android provides the ability to run applications which change the default launcher and hence the appearance and externally visible behaviour of Android. These appearance changes include a multi-page dock or no dock, and many more changes to fundamental features of the user interface.

2.2.2 Android software development

The Android software development kit (SDK) includes a comprehensive set of development tools. These include a debugger, libraries, a handset emulator based on QEMU, documentation, sample code, and tutorials. Currently supported development platforms include computers running Linux (any modern desktop Linux distribution), Mac OS X 10.5.8 or later, and Windows XP or later. For the moment one can also develop Android software on Android itself by using the AIDE - Android IDE - Java, C++ app and the Java editor app. The officially supported integrated development environment (IDE) is Eclipse using the Android Development Tools (ADT) Plugin, though IntelliJ IDEA IDE (all editions) fully supports Android development out of the box, and NetBeans IDE also supports Android development via a plugin. Additionally, developers may use any text editor to edit Java and XML files, then use

command line tools (Java Development Kit and Apache Ant are required) to create, build and debug Android applications as well as control attached Android devices (e.g., triggering a reboot, installing software package(s) remotely).

Enhancements to Android's SDK go hand in hand with the overall Android platform development. The SDK also supports older versions of the Android platform in case developers wish to target their applications at older devices. Development tools are downloadable components, so after one has downloaded the latest version and platform, older platforms and tools can also be downloaded for compatibility testing.

Android applications are packaged in .apk format and stored under /data/app folder on the Android OS (the folder is accessible only to the root user for security reasons). APK package contains .dex files (compiled byte code files called Dalvik executables), resource files, etc.

Android apps are written in the Java programming language. The Android SDK tools compile code—along with any data and resource files—into an APK: an Android package, which is an archive file with an .apk suffix. One APK file contains all the contents of an Android app and is the file that Android-powered devices use to install the app.

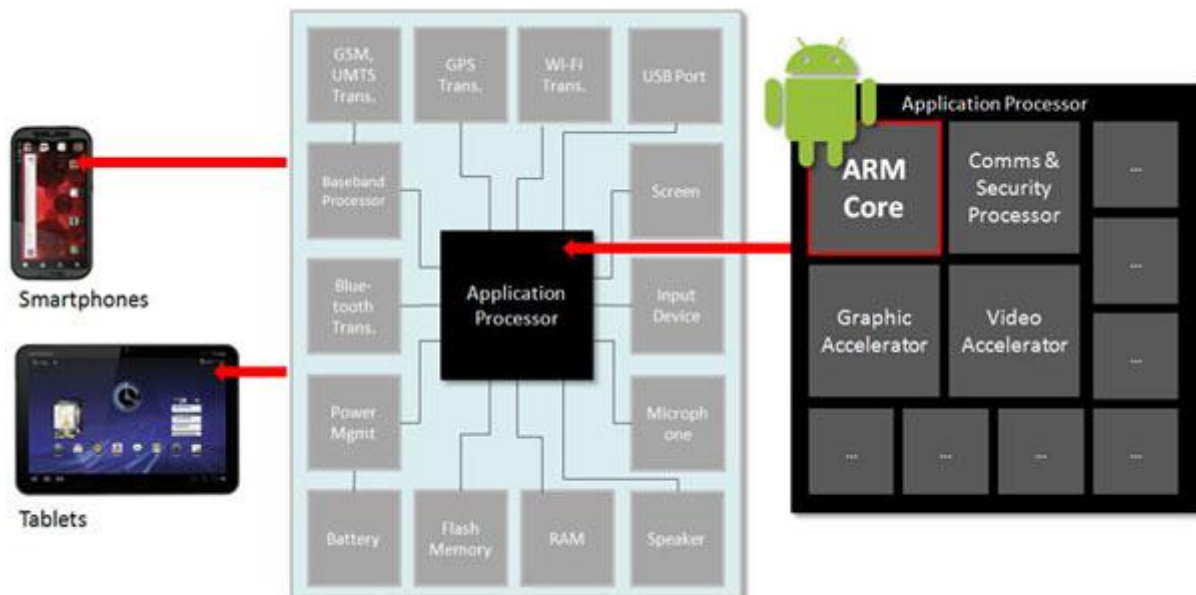
Once installed on a device, each Android app lives in its own security sandbox:

- The Android operating system is a multi-user Linux system in which each app is a different user.
- By default, the system assigns each app a unique Linux user ID (the ID is used only by the system and is unknown to the app). The system sets permissions for all the files in an app so that only the user ID assigned to that app can access them.
- Each process has its own virtual machine (VM), so an app's code runs in isolation from other apps.
- By default, every app runs in its own Linux process. Android starts the process when any of the app's components need to be executed, then shuts down the process when it's no longer needed or when the system must recover memory for other apps.

2.2.3 Android adaption for mobile

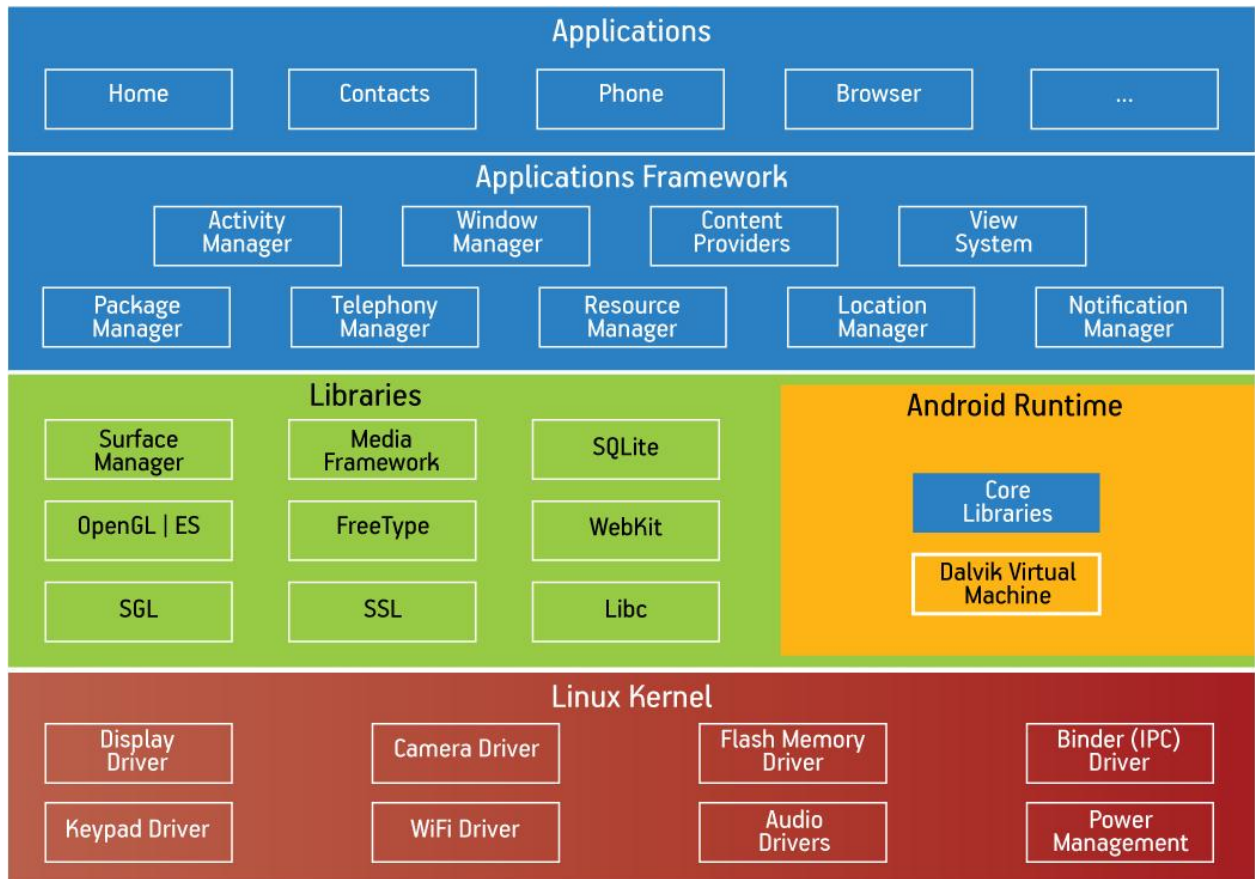
Technology analysts claim android will support nearly half of all smartphones and tablets in just a few short years. Although each new Android release enhances the mobile user experience exponentially, it also creates unique hardware-OS compatibility challenges for mobile device manufacturers and system-on-a-chip (SoC) providers.

A single generation of a mobile application processor can span through multiple versions of an Android operating system. As such, mobile device manufacturers and SoC providers must frequently adapt new Android OS versions to run on their existing hardware. These ongoing adaptation efforts can quickly become overwhelming and pull focus away from core R&D initiatives.



Based on the Android platform software system structure. Android phone is a standard Smartphone. The so-called Smartphone, refers to "the same as personal computers, with separate operating system, installed by the user software, games and other third-party service provider process, through such programs to continue to expand the functionality of the phone, and can be achieved through the mobile communication network to access the wireless network, so that the general term for a class of mobile phone.

Android is designed specifically for mobile terminal software platform and its software architecture from the bottom is divided into four levels, as shown in the figure.



Chapter 3

3.1 Why android

3.1.1 Market Groth:

Android is continuing its onward march. The latest market figures from analyst ABI Research peg the platform at 80% marketshare of smartphone OSes, with just under 300 million smartphones running Android or an Android fork shipped in the quarter.

The analyst notes that basic mobile phone operating systems lost 5% share in the quarter with Android hovering up most of those users as they upgrade to smartphones.

That suggests Google's mobile platform is set to be the biggest winner as the "billions" of remaining mobile subscribers upgrade to smartphones helping to keep what is already the dominant smartphone platform ahead of the competition.

That's bad news for Microsoft which now owns Nokia's mobile making business, and will be hoping to on-ramp users of basic Nokia mobile phones to its Windows Phone-based Lumia smartphones. Currently, Android continues to exert more pulling power for first time smartphone buyers.

Taking the mobile market as a whole so looking collectively at both smartphones and basic mobiles. Android took a 44% share of the market in Q1 (up 24% year-on-year), according to ABI's data, with the Android Open Source Project (AOSP) which companies including Amazon and Nokia have used to fork Android taking 13%.

Microsoft's Windows Phone also grew its share of the market in Q1, with 16% sequential growth and a 3% share overall but the platform remains very much the underdog vs the dominance of Android.

Still, Microsoft can console itself that it's the top underdog in mobile. The analyst described Windows Phone as "the only viable third ecosystem" noting the complete collapse of BlackBerry's OSes, and adding that low-cost Firefox OS remains a potential challenger but has yet to make any significant dent.

Apple's iOS grew 17% year on year but ABI said its growth is "undoubtedly flattening", noting that the iPhone 5c has done little to boost sales volumes.

Apple’s decision to limit itself to selling premium devices continues to constrain its marketshare and as the lower-end portion of the smartphone market continues to expand, fueled by users of basic mobiles upgrading to their first smartphone, that share will inevitably be diluted further.

**Worldwide Smartphone Sales to End Users by Operating System in 2013
(Thousands of Units)**

Operating System	2015 Units	2015 Market Share (%)	2014 Units	2014 Market Share (%)
Android	1,004,675	78.9	227,549	80.8
iOS	191,426	17.9	43,062	15.3
Microsoft	8,271	2.5	7,580	2.7
BlackBerry	1,325	0.4	1,714	0.6
Other OS	1,268.7	0.4	1,731.0	0.6
Total	336,054.4	100.0	281,636.9	100.0

Table 1.1 worldwide Smartphone sales by operating system, Source: Gartner (May 2015) [<http://www.gartner.com/newsroom/id/3061917>]

3.1.2 Technical causes:

After so many years, one would think that the mobile market is now saturated with every app imaginable to man – but that’s not the case. There are plenty of niches that still need to be filled and even current markets with several leading apps can still fall behind in quality. All that to say: there’s always room for new Android apps, and with a little bit of know-how, you could be the next developer to create a smash hit.

Before you begin on your Android development journey, there are a few prerequisites that you need to fulfill.

Java Experience: Android development is mostly done in Java. It’s not a difficult language to learn, per se, but knowing how it applies to Android development can be confusing without a firm foundation in the language. Android Java is not exactly standard Java, but learning standard Java will make it much easier for you to pick up Android development.

XML Experience: Knowing XML can be useful in a number of tech-related fields and Android is no exception. In particular, XML files make it easier to declare UI elements in the apps you create. Past experience is a big plus.

Development Environment: Most Android developers recommend using Eclipse as the development environment for your coding sessions. You’ll also need to install the Android SDK before you can begin making apps. Fortunately, Android provides a single package that includes Eclipse, Android SDK, and related tools. Be aware that in the near future, most Android development will move away from Eclipse toward Android Studio, so you may want to look into that as well.

Perseverance: Lastly, learning the ins and outs of the Android development toolkit is going to take time. Learning how to create a *quality* app is going to take even more time. Don’t rush. Keep at it and focus your mind on the long term. If you don’t, you’ll be frustrated and want to give up.

The operating system is the basis of Smartphone platform, which provides process scheduling, task manager, file systems, network protocols and the phone peripherals such as keyboards, LCD, Buletooth, WiFi and other equipment management. The Choice of operating systems has a direct relationship between product stability, cost, and development prospects. The mainstream of the current Smartphone market, the operating system are the windows Mobile, symbian, iPhone, Android. Windows Mobile is a famous Microsoft designed for intelligent Mobile

terminal development in the operating system, Due to the Windows operating system used for, the user easy to get started, and compatible PC swing many file format. The defect is that the system is relatively high share of resources, easy to collapse, and Microsoft needs to develop products commission, while the operating system's user interface is relatively obscure, bad user experience.

Symbian is currently the largest market share of Smartphone operating system, mainly due to the amount of NOKIA large market share of products, NOKIA symbian Smartphone operating system mainly used. The operating system applies to the operation with the keyboard. The defect is that the mainstream media format support is poor; development requires authorization; With the prevalence of large-screen touch keyboard, the system is still in the keyboard operation mode of operation.

Apple's iPhone leads the current trend of Smartphones: cool visual effects, convenient user has and the faster speed of response. But unfortunately not opening to the outside world is a private system. Android system achieved the completely open to the user, anyone, any units can download the free code for development and design of mobile devices with its own mobile terminal. At the same time, this system of UI design also accords with current trends. Motorola, Samsung, LG Sony Ericsson and other large manufacturers have followed suit, using the system to develop their own intelligence products. Based on Android development, no matter from the technical (developers have the system all the code) or from the development costs on the (no royalties), we will not be controlled by others, it has been more and more manufacturers and developers welcome.

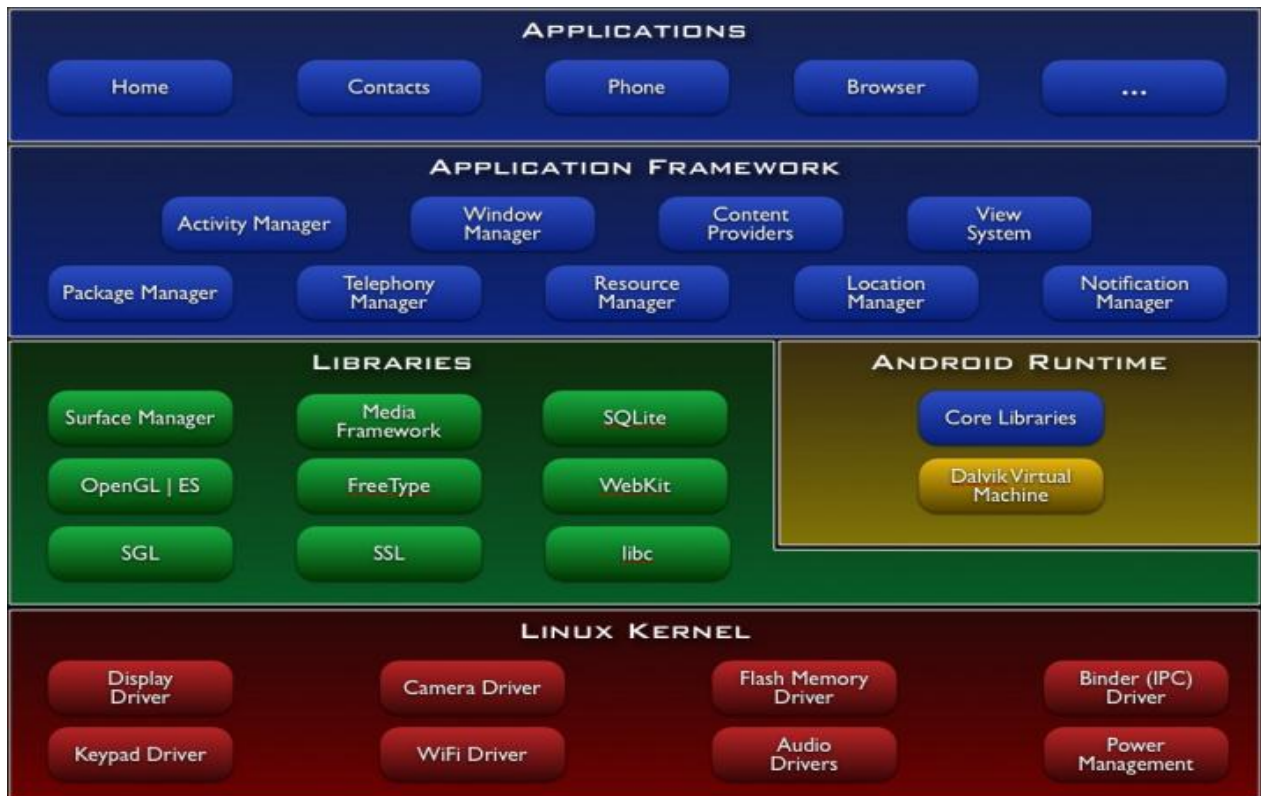
Chapter 4

4.1 Android Architecture

At the top of the stack are the applications themselves. This is where you find the basic functions of the device such as making phone calls, accessing the Web browser and accessing your contacts list. If you're an average user, this is the layer you'll use most. You do that with the user interface. Only Google programmers, application developers and hardware manufacturers access the other layers further down the stack.

The next layer is the application framework. This includes the programs that manage the phone's basic functions like resource allocation, telephone applications, switching between processes or programs and keeping track of the phone's physical location. Application developers have full access to Android's application framework. This allows them to take advantage of Android's processing capabilities and support features when building an Android application. Think of the application framework as a set of basic tools with which a developer can build much more complex tools.

The next level of software includes Android's libraries. You can think of libraries as a set of instructions that tell the device how to handle different kinds of data. For example, the media framework library supports playback and recording of various audio, video and picture formats. Other libraries include a three-dimensional acceleration library (for devices with accelerometers) and a Web browser library.



Android application programmers build their apps using the Java programming language. It also includes the Dalvik Virtual Machine. Libraries provides a set of C/C++ libraries which are used by various components of the Android system. The Media Framework libraries are used to store and playback video as well as images. SQLite is a lightweight relationship database available for all applications. SGL are the underlying 2D graphics engine, while the 3D libraries are based on an OpenGL ES implementation. LibWebCore is a web browser engine. All of these libraries and their capabilities are available for the developer through the Android application framework. Android Runtime includes core libraries and the Dalvik VM. Dalvik is the process virtual machine (VM) in Google's Android operating system, which, specifically, executes applications written for Android. stack-based machines must use instructions to load data on the stack and manipulate that data, and, thus, require more instructions than register machines to implement the same high level code, but the instructions in a register machine must encode the source and destination registers and, therefore, tend to be larger. This difference is primarily of importance to VM interpreters for which opcode dispatch tends to be expensive along with other factors similarly relevant to just-in-time compilation. The core libraries provide most of the basic functionality available in the core libraries of the Java programming language. The Dalvik VM is written as to reduce memory footprint and enable a device to run multiple VM's effectively.

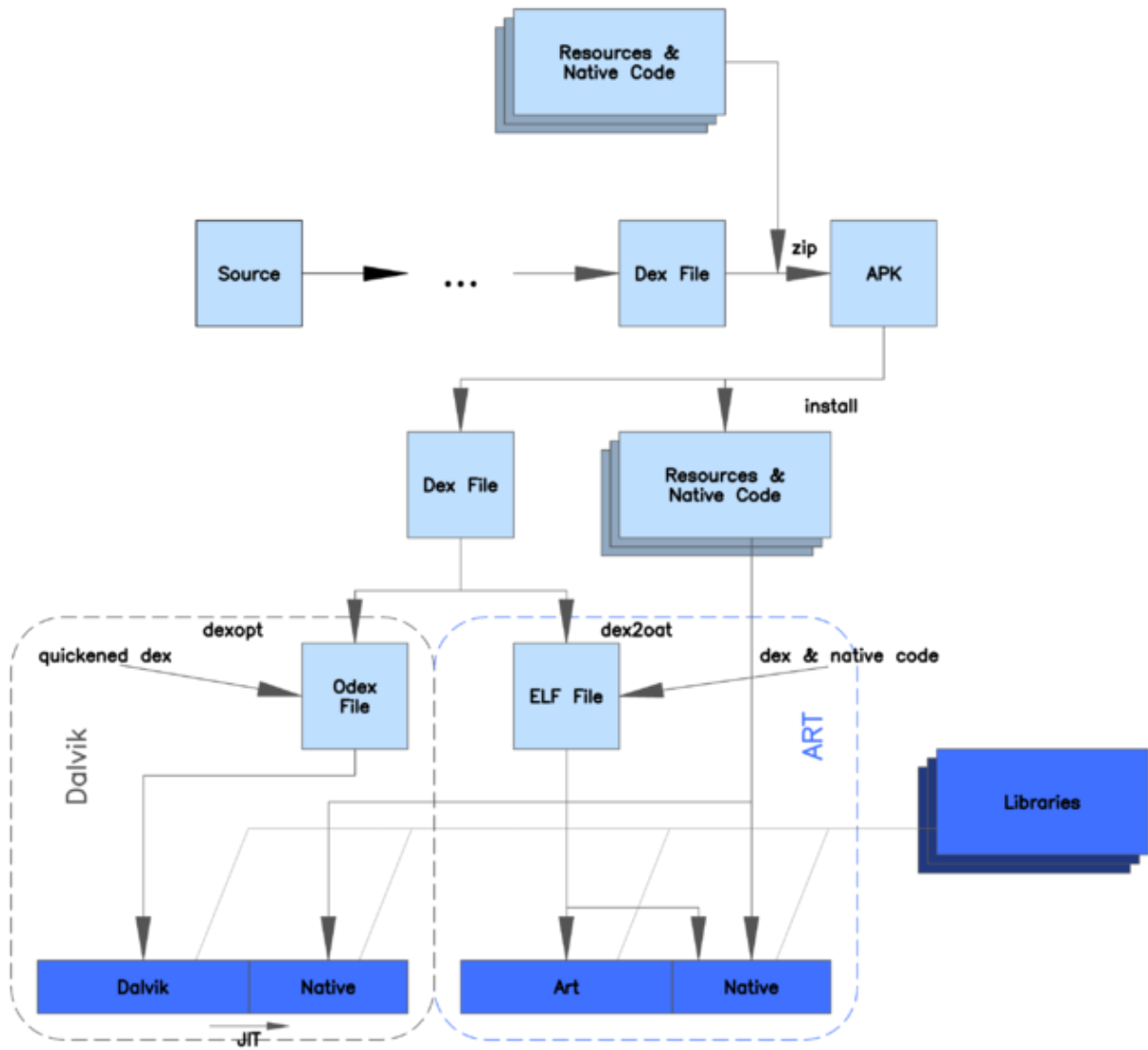


Figure: Depicts the Android architecture

Android Device Manager The Application Framework architecture allows for reuse of components. Any application can make use of the capabilities of a component and also publish its own capabilities. Every application has underlying components, including:

- **Views** consisting of i.e buttons, lists, text boxes and a web browser, all used to build an application.
- **An Activity Manager** that controls navigation and manages the life cycle of an application.
- **A Notification Manager** that enables all applications to have notifications displayed as alerts in the status bar.
- **A Resource Manager** providing access to non-code resources such as localized strings, graphics, and layout files.

Chapter 5

5.1 Application Frame Work

The Application Framework is a set of services that collectively form the environment in which Android applications run and are managed. This framework implements the concept that Android applications are constructed from reusable, interchangeable and replaceable components. This concept is taken a step further in that an application is also able to publish its capabilities along with any corresponding data so that they can be found and reused by other applications.

The Android framework includes the following key services:

- **Activity Manager** – Controls all aspects of the application lifecycle and activity stack.
- **Content Providers** – **Allows applications to publish and share data with other applications.**
- **Resource Manager** – Provides access to non-code embedded resources such as strings, color settings and user interface layouts.
- **Notifications Manager** – Allows applications to display alerts and notifications to the user.
- **View System** – An extensible set of views used to create application user interfaces.
- **Package Manager** – The system by which applications are able to find out information about other applications currently installed on the device.
- **Telephony Manager** – Provides information to the application about the telephony services available on the device such as status and subscriber information.
- **Location Manager** – Provides access to the location services allowing an application to receive updates about location changes.

5.1.1 Android Libraries

This category encompasses those Java-based libraries that are specific to Android development. Examples of libraries in this category include the application framework libraries in addition to those that facilitate user interface building, graphics drawing

and database access. A summary of some key core Android libraries available to the Android developer is as follows:

- `android.app` – Provides access to the application model and is the cornerstone of all Android applications.
- `android.content` – Facilitates content access, publishing and messaging between applications and application components.
- `android.database` – Used to access data published by content providers and includes SQLite database management classes.
- `android.graphics` – A low-level 2D graphics drawing API including colors, points, filters, rectangles and canvases.
- `android.hardware` – Presents an API providing access to hardware such as the accelerometer and light sensor.
- `android.opengl` – A Java interface to the OpenGL ES 3D graphics rendering API.
- `android.os` – Provides applications with access to standard operating system services including messages, system services and inter-process communication.
- `android.media` – Provides classes to enable playback of audio and video.
- `android.net` – A set of APIs providing access to the network stack. Includes `android.net.wifi`, which provides access to the device’s wireless stack.
- `android.provider` – A set of convenience classes that provide access to standard Android content provider databases such as those maintained by the calendar and contact applications.
- `android.text` – Used to render and manipulate text on a device display.
- `android.util` – A set of utility classes for performing tasks such as string and number conversion, XML handling and date and time manipulation.
- `android.view` – The fundamental building blocks of application user interfaces.
- `android.widget` – A rich collection of pre-built user interface components such as buttons, labels, list views, layout managers, radio buttons etc.
- `android.webkit` – A set of classes intended to allow web-browsing capabilities to be built into applications.

Having covered the Java-based core libraries in the Android runtime, it is now time to turn our attention to the C/C++ based libraries contained in this layer of the Android software stack.

5.1.2C/C++ Libraries

The Android runtime core libraries outlined in the preceding section are Java-based and provide the primary APIs for developers writing Android applications. It is important to note, however, that the core libraries do not actually perform much of the actual work and are, in fact, essentially Java “wrappers” around a set of C/C++ based libraries. When making calls, for example, to the `android.opengl` library to draw 3D graphics on the device display, the library actually ultimately makes calls to the

OpenGL ES C++ library which, in turn, works with the underlying Linux kernel to perform the drawing tasks.

C/C++ libraries are included to fulfill a wide and diverse range of functions including 2D and 3D graphics drawing, Secure Sockets Layer (SSL) communication, SQLite database management, audio and video playback, bitmap and vector font rendering, display subsystem and graphic layer management and an implementation of the standard C system library (libc).

In practice, the typical Android application developer will access these libraries solely through the Java based Android core library APIs. In the event that direct access to these libraries is needed, this can be achieved using the Android Native Development Kit (NDK), the purpose of which is to call the native methods of non-Java programming languages (such as C and C++) from within Java code using the Java Native Interface (JNI).

Main application components

To begin to program for android we need some basic, because some elements are very different, even if programming an application in Android uses the java language, therefore, an object oriented language.

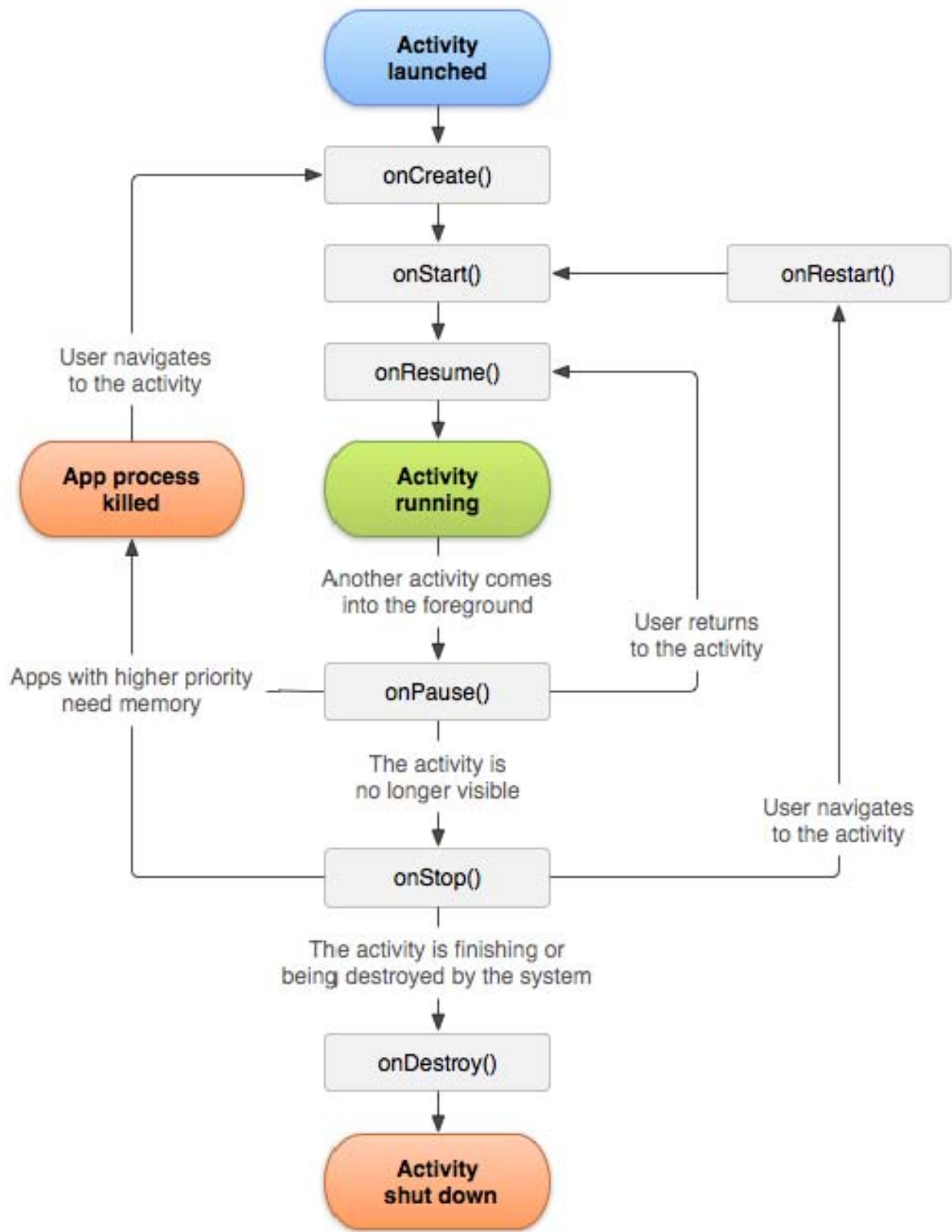
All Android application contain more of the following components

5.1.3 Activity

An activity is an user interference that allows the user to interact with the screen, to perform actions. For example, a text messaging application could have an activity that displays a list of contacts to send messages. Once the contact is selected ,activity could send information to a second activity that could serve to send the message to the contact.

When an application is launched, what is displays is the result of an activity. At the code level, for create an activity; we must create a class that extend the activity class. An activity has a required on create() method. To interact with the program, through the activity, there must be something displayed, that is why the activity, contains what is called views.

Process status	Description	Priority
Foreground	An application in which the user is interacting with an activity, or which has an service which is bound to such an activity. Also if a service is executing one of its lifecycle methods or a broadcast receiver which runs its onReceive() method.	1
Visible	User is not interacting with the activity, but the activity is still (partially) visible or the application has a service which is used by a inactive but visible activity.	2
Service	Application with a running service which does not qualify for 1 or 2.	3
Background	Application with only stopped activities and without a service or executing receiver. Android keeps them in a least recent used (LRU) list and if requires terminates the one which was least used.	4
Empty	Application without any active components.	



Activity lifecycle

The Android system is also allowed to recycle Android components to free up resources. This part explains which for activities, the lifecycle of other components is described in the respective part of these components.

An activity can be in different states which are described by the following table.

State	Description
Running	Activity is visible and interacts with the user.
Paused	Activity is still visible but partially obscured, instance is running but might be killed by the system.
Stopped	Activity is not visible, instance is running but might be killed by the system.
Killed	Activity has been terminated by the system or by a call to its finish() method.

5.1.4 View

A view is the basic building block for user interface components. A view occupies a rectangular area on the screen. View is the base class for the widget, which are used to create interactive UI components. There are different kinds of view, for example a ListView is able to display only an interactive list of what we want to display, while a WebView allows us to display a webpage. As said before, a view occupies a rectangular area on the screen. To organize these rectangles on the screen, there is a text file written in the XML for every different screen.

5.1.5 XML

XML means Extensible Markup Language. Android provides a straightforward XML vocabulary that corresponds to the view classes and subclasses. The goal of using Android XML vocabulary is to quickly design UI layouts and the screen elements.

they contain, in the same way that creating web pages in HTML: with a series of nested elements.

5.1.6 Intent

An activity can of course start another one, even if it but to do this, it will need special object called intent. Intent is basic description of an operation to be performed. It can launch an Activity, send a broadcastintent to any interested Broadcast receiver components, and communicate with a background service. Intent performs binding between the codes indifferent applications. It can be thought of as the link between activities.

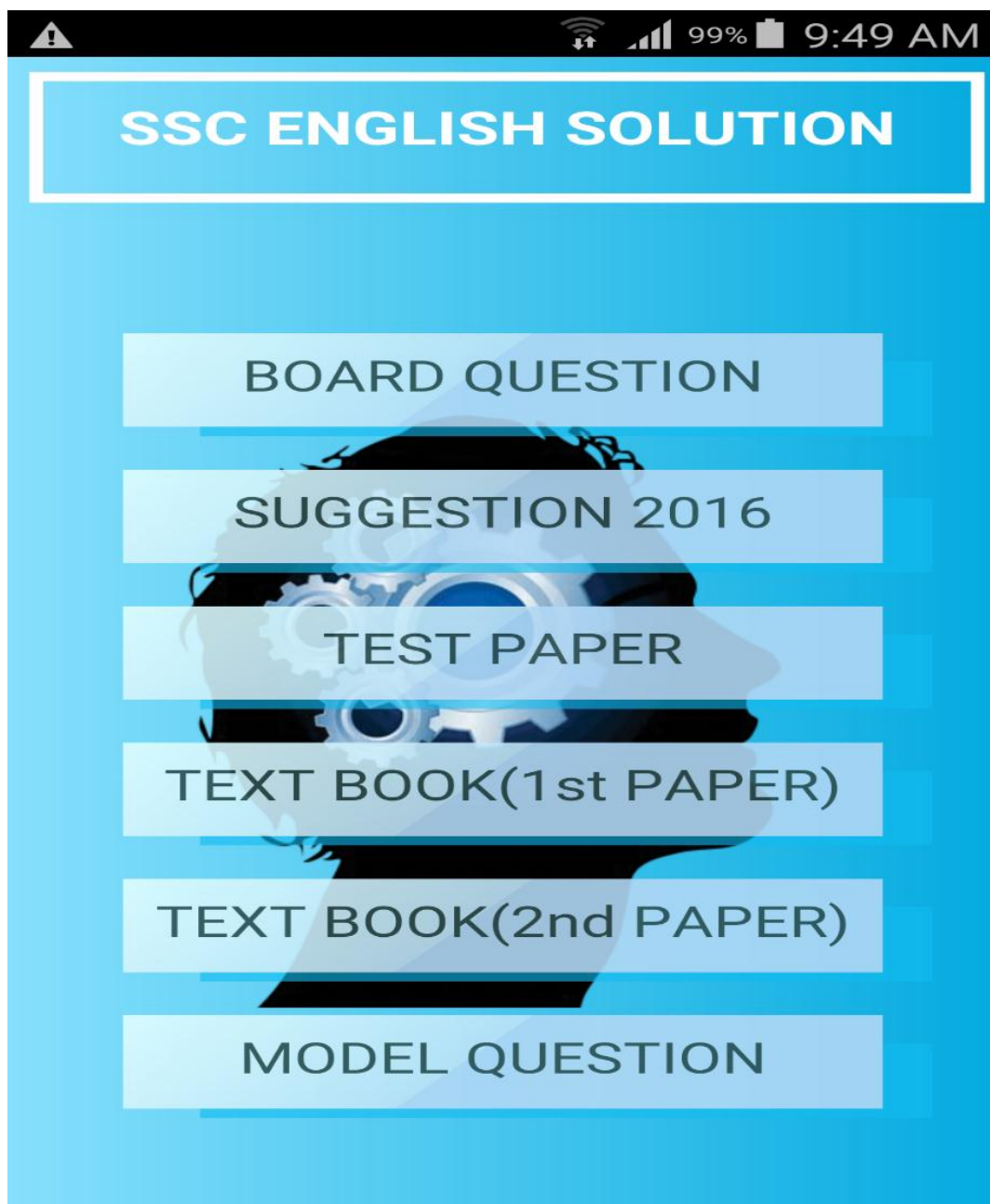
5.2 Android Manifest

AndroidManifest.xml file is necessary for all android applications and must have this name in itsroot directory. In the manifest we can find essential information about the application for the androidsystem, information that the system must have before it can run any of the application's code. Here is what we can find in the Android manifest:

- The name of the Java package for the application. The package name serves as a unique identifier for the application.
 - The description of the components of the application: the activities, services, broadcast receivers, and content providers that the application is composed of and under what conditions they can be launched.
 - The process that will host application components.
 - The permissions the application must have in order to access protected parts of the API andinteract with other applications.
 - The permissions that others are required to have in order to interact with the application's components.
 - The list of tee instruments classes that provide profiling and other information as the application is running. These declarations are present in the manifest only while the application is published.
 - The minimum level of the Android API that the application requires.
 - The list of the libraries that the application must be linked against.
- With all these elements, an application can be created

Chapter 6

6.1 Snapshot and overview of SSC ENGLISH SOLUTION APP



ENGLISH SECOND PAPER
DHAKA BOARD—2013

Part A: Grammar (40 Marks)

1. Complete the passage with suitable verbs from the list. Put them in the correct tense. Use the negatives where necessary. 1×5=5

provide leave turn be cut stop prevent

Trees (a) _____ useful to us in many ways. They (b) _____ us with food, fruit, shade and shelter. They prevent a region from (c) _____ into a desert. But it is a pity that the people of our country (d) _____ trees indiscriminately. It is high time we (e) _____ such an act of human cruelty towards trees.

2. Read the following letter and fill in the blanks with suitable prepositions. 1×5=5

Dear Helal,

Yesterday I was informed (a) _____ your bad habit. You are addicted (b) _____ smoking! This piece of information has upset me very much. You should bear in mind that smoking is very harmful and it tells (c) _____ health. So, you should abstain (d) _____ smoking. You should not indulge (e) _____ any bad habit.

Your loving elder brother

Habib

3. Use articles where necessary. Put a cross (×) mark where an article is not needed. .5×10=5

Though newspaper is not always (a) _____ unmixed blessing, it is (b) _____ most useful thing in our life. We rarely think of (c) _____ day without reading (d) _____ newspaper. Reading newspaper is one of (e) _____ good habits of a man. We shall be like (f) _____ frog in a dark well if we neglect developing this habit. However, some people prefer (g) _____ electronic media to reading (h) newspaper. But (i) _____ newspaper offers more to its readers than (j) _____ electronic media.

4. Make meaningful sentences with the following idioms (any five). 1×5=5

keep pace with, win over, accountable to, read between the lines, skim through, Achilles' heel, slow coach.

5. Rewrite the following passage in the reported speech. 5

The teacher said to the boy, "Do you think that honesty is the best policy?" The boy said, "Yes, sir, I think so." "Then, learn to be honest from your boyhood," said the teacher. "Thank you, sir," said the boy. "May Allah grant you a long life," said the teacher to the boy.

6. Read the following passage and transform the underlined sentences as directed. 1×5=5

Success will not come to one's life automatically. (a) Do you know it? (Passive). (b) An industrious boy will shine in life. (Complex). (c) A life with an assignment is an actual life. (Negative). (d) None can receive any reward unless he works hard. (Simple) (e) It is evident that industry is one of the greatest virtues. (Positive).

7. Fill in the blanks with tag questions. 1×5=5

(a) A student should read the texts again and again, —?

(b) Nobody pays due respect to a dishonest man, —?

(c) "I" is a pronoun, —?

(d) Thousands of people admired Rumi in his life time, —?

(e) Deeba as well as her friends enjoyed the short film, —?

8. Complete the following sentences. 1×5=5

(a) Read attentively or _____.

(b) Had I been there _____?

(c) There goes the proverb that _____.

(d) It was long since _____.

(e) Do what _____.

Part B : Composition (60 Marks)

9. Suppose, you are a reporter of a national daily. You are assigned to cover a massive fire in a factory.

Now, write a report on it. 10

10. Write a composition on any one of the following. 15

(a) Patriotism; (b) Duties and Responsibilities of Students.

Subject Code 1 0 7

HSC Model Question of 2016

Class XI

Subject : English 1st Paper

Time : 3 Hours

Full Marks — 100

Read the passage below and answer the Questions 1– 4.

Waste or rubbish is what we throw away everyday. It includes everything from unwanted old cars to cigarette packets. As the amount of waste has grown rapidly all over the world and has started polluting the environment, time has come to think about it very seriously.

We cannot altogether get rid of our waste but a proper management of it can certainly reduce its quantity. If we think of burning, burying, recycling and thus reduce our waste, we can to a large extent save our environment. Our garbage consists of various kinds of things – some are burnable and some are non-burnable and some garbage can be melted and refabricated. Some rubbish is not rubbish at all. For example, most textiles, glass metal, paper and have been using vegetable waste to make compost to improve our soil.

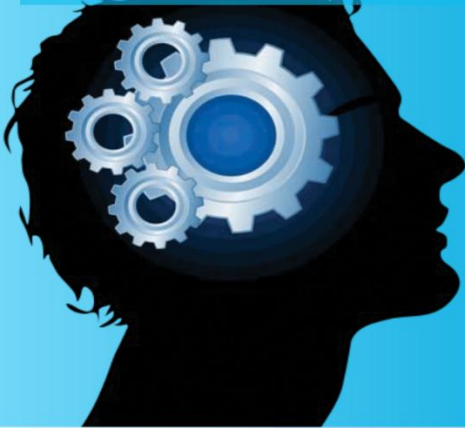
The only problem then is to find authorities to make it easy to reuse what we throw away. This is because we cannot recycle paper or metal ourselves. It is a big proceduse. In many countries it is now quite normal to collect old bottles and reuse them. Again many Companies are now designing their containers and packaging in such a way that they are easy to recycle them. Some Companies are using biodegradable wrappers. More and more organizations should came forward to promote greater recycling and changing its consumption patterns to reduce the amount of rubbish we produce and help people save the environment.

1. Choose the right word/expression to complete each sentence : $1 \times 5 = 5$
 - a) In some countries, incinerators destroy/destroying/destroyed burnable waste.
 - b) We can reduce waste by proper law/decision/management.
 - c) Biodegradable wrappers are suitable/harmful/destructive for the environment.
 - d) What we throw away everyday includes/included/including everything from old cars to cigarette packets is waste.
 - e) Some kinds of waste can be used as fuel/food/clothes.
2. True/False? If false, give the correct information – $1 \times 5 = 5$
 - a) It is not proper to reuse the old bottles.
 - b) We can recycle paper or metal ourselves.
 - c) Vegetable waste is used for making fertilizer.
 - d) Most textiles, glass and metal are not rubbish at all.
 - e) All kinds of garbage are burnable.

<http://teachingbd.com>

TEST PAPER

1st PAPER



2nd PAPER



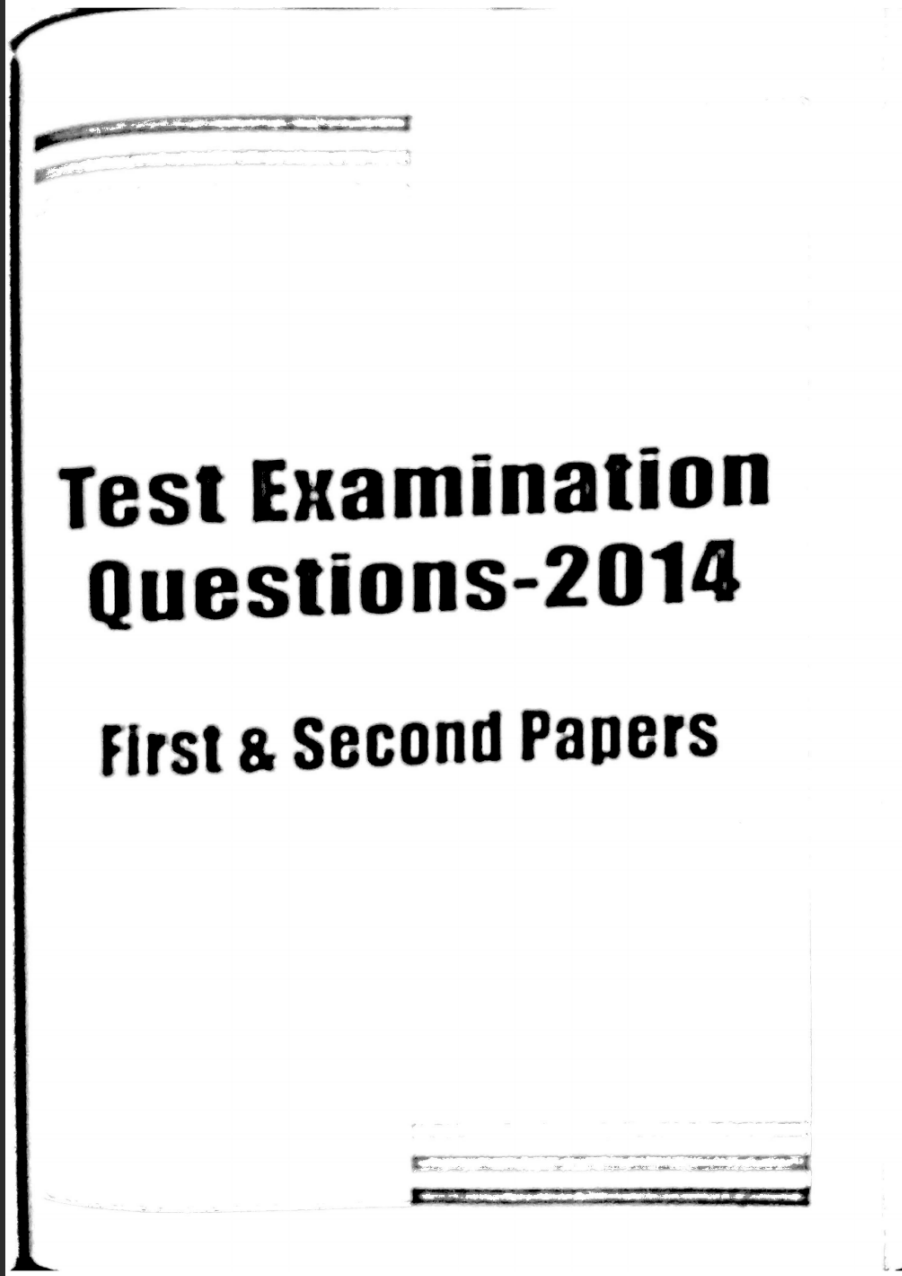
69% 10:24



englis ... er.pdf



সাইন ইন করুন



Test Examination Questions -2014 Second Paper

I Rajuk Uttara Model College, Dhaka
Test Examination-2014
English Second Paper
Part A: Grammar (60 Marks)

1. Fill in the blanks with the words from the box. You may need to change the forms of some of the words. You may need to use one word more than once. 0.5×10=5

dominate	an	a	over	after
from	free	on	to	before

Liberty does not descend (a) ----- people automatically. People must raise themselves (b) ----- it. It is (c) ----- fruit that must be earned (d) ----- it can be enjoyed. That (e) ----- means freedom only from foreign (f) ----- is (g) ----- -outworn idea. No (h) ----- has any real value unless it also means freedom (i) ----- want, disease and ignorance, not merely freedom from foreign (j) -----.

2. Fill in the blanks with suitable words: 0.5×10=5

Smoking (a) ----- dangerous (b) ----- health as tobacco leaf contains (c) ----- large quantity of (d) ----- highly poisonous alcoholic, nicotine which is one of (e) ----- most known fatal poisons. Thousands of people of Bangladesh suffer (f) ----- cancerous diseases and (g) ----- 60.00 of them (h) ----- of these diseases every year. Most of these cancerous diseases are (i) ----- directly or indirectly due (j) ----- smoking of tobacco.

3. Make five sentences using the parts of sentences from each column of the table below: 1×5=5

Education in our country	Bangladesh is already	a daily affair.
Campus Violence	are getting concerned about	only to be engaged in violence.
It seems that	has become almost	stage of crisis
It is true that	students are in educational institutions	educationally backward.
The teachers and the guardians	is passing through a	the welfare of the students.

4. Complete the following text with right forms of the verb given in the box. 0.5×10=5

contribute	kill	come	show	flee
involve	achieve	have	be	take

Today we (a) ----- free nation. We (b) ----- to take great pains to gain freedom. Our War of Liberation (c) ----- place in 1971. People of all walks of life (d) ----- forward and got (e) ----- in the war directly and indirectly. Many a man (f) ----- in the battlefields. They (g) ----- from the battlefields (h) ----- their back. Rather all of them (i) ----- much to (j) ----- dependence.

5. Change the narrative style of the following text. 5

"Sir," he told, "May I ask you a question?" "Of course." "I was absent in your class yesterday and could not understand the lesson." "Ok, if you are free after this class, meet me in the teachers' room. I will help you." "Thank you, sir." "You're welcome, son."

6. Change the sentences according to directions. 1×10=10

- (a) Wherever we may cast our eyes, we notice the achievements of science. (simple)
- (b) There electricity lights the streets and houses. (passive)
- (c) The radio broadcasts news, songs and lectures. (passive)
- (d) There is cinema to delight you in the evening. (complex)
- (e) Computer is the most important invention of modern science. (positive)
- (f) We cannot go without it. (interrogative)
- (g) Internet is not less important than computer. (positive)
- (h) It helps us to know about the world. (compound)
- (i) It is needed in every office. (active)
- (j) Having a good effect, it helps us. (compound)

ADV. GRA. 2ND. IX-6. A



68%



10:25



englis ... ay.pdf



সাইন ইন করুন

ENGLISH FOR TODAY

CLASSES 9-10



NATIONAL CURRICULUM & TEXTBOOK BOARD, DHAKA

Prescribed by the National Curriculum & Textbook Board
as a Textbook from the academic session 2013

MODEL QUESTION

1st PAPER



2nd PAPER

Chapter 7

7.1 JAVA Code For SSC ENGLISH SOLUTION APP

```
package nextapps.sscenglishsolution;

import android.os.Bundle;
import android.view.KeyEvent;
import android.widget.LinearLayout;
import android.widget.Toast;

import org.apache.cordova.DroidGap;

public class MainActivity extends DroidGap {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        super.loadUrl("file:///android_asset/www/index.html")
        ;

    }
}
```

7.2 AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="nextapps.sscenglishsolution"
    android:installLocation="preferExternal">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >

        <activity
            android:name=".MainActivity"
            android:label="@string/app_name"
            android:configChanges="orientation|screenSize|keyboardHidden">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

    </application>

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission
        android:name="android.permission.ACCESS_NETWORK_STATE"/>
</manifest>
```

HTML CODE FOR INDEX:

```
<!DOCTYPE HTML>
```

```
<HTML>
```

```
  <HEAD>
```

```
    <LINK HREF="STYLE.CSS" REL="STYLESHEET">
```

```
  </HEAD>
```

```
  <BODY>
```

```
    <DIV CLASS="MAIN">
```

```
      <DIV CLASS="HEADER">
```

```
        <H1>SSC ENGLISH SOLUTION</H1>
```

```
      </DIV>
```

```
    <DIV CLASS="MAIN2">
```

```
      <DIV CLASS="BUTTON">
```

```
        <A
```

```
        HREF="HTTPS://DRIVE.GOOGLE.COM/FILE/D/0B-
```

```
        TXFWWX1QJcWXF6Z01OUUHMR3M/VIEW?USP=SHARING">
```

```
        <P>BOARD QUESTION</P>
```

```
        </A>
```

```
      </DIV>
```

```
      <DIV CLASS="BUTTON">
```

```
        <A HREF="HTTPS://DRIVE.GOOGLE.COM/FILE/D/0B-
```

```
        TXFWWX1QJcAMHFRJIYDHTCFk/VIEW?USP=SHARING">
```

```
        <P>SUGGESTION 2016</P>
```

```
        </A>
```

```
      </DIV>
```

```
      <DIV CLASS="BUTTON">
```

```
        <A HREF="PAGE1.HTML">
```

```
        <P>TEST PAPER</P>
    </DIV>
</A>

<DIV CLASS="BUTTON">
    <A HREF="HTTPS://DRIVE.GOOGLE.COM/FILE/D/0B-
TXFWWX1QJCDUFMVVLTVKQXNXC/VIEW?USP=SHARING">
        <P>TEXT BOOK(1ST PAPER)</P>
    </DIV>
</A>

<DIV CLASS="BUTTON">
    <A HREF="HTTPS://DRIVE.GOOGLE.COM/FILE/D/0B-
TXFWWX1QJcVDC4VLBWR3J6ELE/VIEW?USP=SHARING">
        <P>TEXT BOOK(2ND PAPER)</P>
    </DIV>
</A>

<DIV CLASS="BUTTON">
    <A HREF="PAGE2.HTML">
        <P>MODEL QUESTION</P>
    </DIV>
</A>

</DIV>
</DIV>
</BODY>
</HTML>
```

HTML CODE FOR PAGE 1:

```
<!DOCTYPE HTML>
<HTML>
  <HEAD>
    <LINK HREF="STYLE1.CSS" REL="STYLESHEET">
  </HEAD>

  <BODY>
    <DIV CLASS="MAIN">
      <DIV CLASS="HEADER">
        <H1>TEST PAPER</H1>
      </DIV>

      <DIV CLASS="MAIN2">

        <DIV CLASS="BUTTON">
          <A HREF="HTTPS://DRIVE.GOOGLE.COM/FILE/D/0B-
TXFWWX1QJcRNjMwK5xV2JLSEU/VIEW?USP=SHARING">
            <P>1ST PAPER</P>
          </A>
        </DIV>

        <DIV CLASS="BUTTON">
          <A HREF="HTTPS://DRIVE.GOOGLE.COM/FILE/D/0B-
TXFWWX1QJcBTVWQ3o5VNBKYWs/VIEW?USP=SHARING">
            <P>2ND PAPER</P>
          </A>
        </DIV>

      </DIV>
    </DIV>
```

</BODY>

</HTML>

HTML CODE FOR PAGE 2:

<!DOCTYPE HTML>

<HTML>

<HEAD>

<LINK HREF="STYLE1.CSS" REL="STYLESHEET">

</HEAD>

<BODY>

<DIV CLASS="MAIN">

<DIV CLASS="HEADER">

<H1>MODEL QUESTION</H1>

</DIV>

<DIV CLASS="MAIN2">

<DIV CLASS="BUTTON">

<P>1ST PAPER</P>

</DIV>

<DIV CLASS="BUTTON">

<P>2ND PAPER</P>

</DIV>

</DIV>
</DIV>

</BODY>

</HTML>

BIBLIOGRAPHY

[ADB] Android Developers Blog

<http://android-developers.blogspot.com/>

[ADG] Android Developers Group

[http:// groups.google.com/group/android-developers](http://groups.google.com/group/android-developers)

[AND] Android Developers

<http://developers.android.com>

<http://w3schools.com>

<http://epubl.ltu.se/1404-5494/2010/006/LTU-HIP-EX-10006-SE.pdf>

<http://arstechnica.com/gadgets/2014/06/building-android-a-40000-word-history-of-googles-mobile-os/>

<http://startapp.com/>

<http://www.btd.com.bd/>

<http://www.gartner.com/newsroom/id/2665715>