Unit 14 Application for Mobile Devices

Structure:

14.1 Introduction
Objectives
14.2 Exploring Android Operating System
Features of Android Operating System
14.3 Platforms to Design Mobile Applications
App Inventor for Android
Android SDK
14.4 Components to Enable Android Function
14.5 Summary
14.6 Terminal Questions
14.7 Answers

14.1 Introduction

Mobile applications supports the user to access various enterprise applications in an efficient and timely manner**.** Many organizations provided simple choices of email options to use on small devices. With such Interactive mobile devices and applications, the usage of 3G data services are growing with new application stores and mobile platforms leads to rapid growth in wireless industry. Through mobile application we are able to deliver short and accurate information easily and fast.

Android Operating System has become the most common Operating System (OS) used mainly in the Mobile devices. It has many benefits and It is the software to run a mobile phone but without the proprietary obstacles. However, the vast majority of Android devices ship with additional proprietary software, primarily Google Mobile Services (GMS), which includes essential apps like Google Chrome, the digital distribution platform Google Play, and the associated Google Play Services development platform. The intent of Android technology is to accelerate innovation in mobile devices and offer consumers a richer, less expensive, and better mobile experience. In this unit we are discussing the features of Android mobile operating system. Also we will explore the App inventor visual programming platform its components and to develop mobile applications in App inventor and Android SDK environment. We will also tend to discussthe list of components that enable the android functions.

Objectives:

After studying this unit, you will be able to:

- discuss the features of Android operating system
- brief the App inventor Android visual programming platform
- develop mobile application on App Inventor IDE
- list and discuss the components that enable Android function

Mobile Devices and its features:

Let us understand the basic definition of mobile devices and some of their features. A portable device is a piece of technology that can be utilised on the go. Any portable virtual machine will have a touchscreen LCD or OLED display, digital buttons and keyboard, or physical buttons and keyboard. Android is a software stack for mobile devices that includes an operating system, middleware and key applications. The Android Software Development Kit (SDK) provides the tools and Application Programming Interfaces (APIs) necessary to begin developing applications on the Android platform using the Java programming language. A long-lasting battery, Warp-speed processing, Crystal-clear display and a great camera these are all the basic features of a mobile devices

Platform used for mobile devices: Explain the foundation on which mobile devices are built. For the purposes of delivering voice, data, and AV services, a "Mobile Platform" refers to any network or service that employs mobile communications standards and protocols. When we talk about "mobile platforms," we're referring to things like construction platforms, floating service platforms, and mobile drilling units. So the basic platform for mobile devices are google playstore and App store.

1. Google playstore: The Play Store is Google's official pre-installed app store on Android-certified smartphones. It allows users to access the Google Play Store content including sports, books, magazines, music, movies and TV shows. The Play Store limits the number of apps available to those that work with the user's device. Google's digital distribution service is called Google Play, but it's also known as the Google Play Store and before that as the Android Market.
2. App Store: The App Store is a digital distribution platform for mobile apps for Apple's iOS and iPadOS operating systems that was developed and is maintained by Apple Inc. The store allows users to browse for and download apps developed with Apple's iOS Software Development Kit. Apps for the iPhone, iPod Touch and

iPad tablet computers are all available for free

3.

Mobile Phone Applications:

Let us understand the areas where mobile phone applications work. You can use them to send and receive calls, texts, and emails. We can also use our phones to access the web. Most significantly, we use the camera on our phones to take pictures and film videos. Smartphones are the modern-day equivalent of mobile phones. They can be used in small business, professional life, learning purpose , as a travel guide , for connectivity , news , email or social media and also can be used for entrainment purpose

Types of Mobile Apps: let us understand the various types of mobile apps that we use in daily life. Internet browser Android has the open source Web browser like WebKit layout engine, coupled with Chrome's V8 JavaScript engine. Here in terms of performance web browser in Android scored 100 percentage on the Acid3 test (checks for web browser's compliance of various elements with web standards) on the Android version 4.0. weather check . It can be used in weather channel , social media , google map, uber apps , youtube and many more.

let us know some of the top free apps like whats app, applock , VLC media player , viva video , hill climbing racing , google chrome and much more in various filed.

Fundamentals of mobile operating System:

A mobile operating system is a type of computer software that allows mobile devices to execute additional applications. It's a marvel of both computer and hand-held technology. Common features include a SIM card slot and a built-in modem for cellular data and voice calls. The mobile device's operating system (OS) is determined by the manufacturer. The software is the same as that found in well-known OSes like Linux and Windows, but with a focus on portability and ease of use. An operating system is a software that allows other programmes to communicate with the hardware and run, Windows, OS X and Linux are all popular desktop operating systems, Android and iOS are two common mobile operating systems

14.2 Exploring Android Operating System

Nowadays, you can find mobile devices with various technologies and Mobile Operating System (OS) running on it. The types of Mobile OS that you may find are: Symbian, iPhone, RIM's BlackBerry, Window mobile, Linux, Palm web OS, Android etc. It's open-source and free for anyone to use. The Android Open Source Project (AOSP) is responsible for its development and distribution.

Although all the mobile OS has its own merits and demerits, Android is the first truly open and comprehensive platform for mobile devices. It is the software to run a mobile phone but without the proprietary obstacles. Android is the world's mainly accepted mobile platform. Android also provides a platform to the developers to develop new applications and games and sell or distribute those applications and games on the Android marketplace.

Android is a software stack for mobile devices that includes an operating system, middleware and key applications. The Android Software Development Kit (SDK) provides the tools and Application Programming Interfaces (APIs) necessary to begin developing applications on the Android platform using the Java programming language. Android Inc was founded in Palo Alto of California, U.S. by Andy Rubin, Rich miner, Nick sears and Chris White in 2003. Later Android Inc. was acquired by Google in 2005. After original release, there have been number of updates in the original version of Android. Android is now developed by Open Handset Alliance (OHA), a group of 84 technology and mobile companies who have come together to accelerate innovation in mobile and offer consumers a richer, less expensive, and better mobile experience.

You must be wondering why we need to go for Android in comparison to other OSs. It is so because it provides a simple and powerful SDK and no licensing, distribution or development fees is required to obtain it. With Android, you can go for development over many platforms like Linux, Mac

OS, Windows. You can also get excellent documentation with the support of active developer community. Android is built on the open Linux Kernel. Furthermore, it utilizes a custom virtual machine that was designed to optimize memory and hardware resources in a mobile environment. Android is an open source and hence it can be liberally be extended to incorporate new cutting edge technologies as they emerge. You can incorporate Java programming for the development of mobile device applications with the support of open source libraries like SQLite, WebKit, OpenGL. You can import 3rd party Java library for the enhancement of the applications. The platform will continue to evolve as the developer community works together to build innovative mobile applications.

14.2.1 Features of Android Operating system

Handset layouts: The platform is adaptable to higher, VGA (Video Graphics Array), 2D graphics library, 3D graphics library based on OpenGL ES 2.0 specifications, and traditional smartphone layouts.

Storage: SQLite is one of the lightweight relational databases, used in Android platform that makes the data storage more easy and comfortable. Android uses a file system that's similar to disk-based file systems on other platforms. The system provides several options for you to save your app data:

Connectivity: Android supports a connectivity technology that includes IDEN NFC, GSM/EDGE, Bluetooth, CDMA, EV-DO, UMTS, WiMAX., Wi-Fi, LTE, etc.

Messaging: The two major forms of message availability are in the forms of SMS (Shot Message Service) and MMS (Multimedia Messaging Service). It includes the text messaging and Android Cloud To Device Messaging (C2DM) is threaded text messaging. Now enriched version of C2DM, Android GCM (Google Cloud Messaging) are also a part of Android Push Messaging service.

Multiple language support: Android Operating system supports multiple languages.

Web browser: Android has the open source Web browser like WebKit layout engine, coupled with Chrome's V8 JavaScript engine. Here in terms of performance web browser in Android scored 100 percentage on the Acid3 test (checks for web browser's compliance of various elements with web standards) on the Android version 4.0.

Java support: Most of the applications of Android is written in Java language but there is no JVM or byte code to execute in the platform. Dalvik is the process VM in Google’s Android OS supporting apps on Android OS. All the Java class files are compiled and executed using this Dalvik support. Dalvik is an integral part of Android designed to optimize the battery powers in devices with limited CPU and memory. With the Support of third party application J2ME can be executed in Android OS.

Media support: Android Operating system supports the following audio, video and still media formats: WebM, H.263, H.264, AAC, HE-AAC (in 3GP or MP4 container), AMR, MPEG-4 SP, AMR-WB (in 3GP container), MIDI,MP3, Ogg Vorbis, WAV, FLAC, GIF, JPEG, PNG, BMP, and WebP.

Streaming media support: RTP/RTSP streaming (3GPP PSS, ISMA), HTML progressive download (HTML5 <video> tag). Adobe Flash Streaming (RTMP) and HTTP Dynamic streaming are supported by the Flash plugin.

Additional hardware support: Android can use video/still cameras, touchscreens, GPS, accelerometers, gyroscopes, barometers, magnetometers, dedicated gaming controls, proximity and pressure sensors, thermometers, accelerated 2D bit blits (with hardware orientation, scaling, pixel format conversion) and accelerated 3D graphics.

Multi-touch: Android has native support for multi-touch which was initially made available in handsets such as the HTC Hero. Multiple fingers touching the display at once constitutes a multi-touch gesture. How to identify multi-pointer motions is covered in this tutorial.

Bluetooth: Supports A2DP, AVRCP, sending files (OPP), accessing the phone book (PBAP), voice dialing and sending contacts between phones. Keyboard, mouse and joystick (HID) support is available from Android 3.1, and in earlier versions done through manufacturer customizations and third-party applications.

Video calling: Android does not support native video calling, but some handsets have a customized version of the operating system that supports it, either via the UMTS network (like the Samsung Galaxy S) or over IP. Video calling through Google Talk is supported in Android 2.3.4 and in later versions.

Multitasking: Android OS supports the multitasking of applications by allocating unique memory.

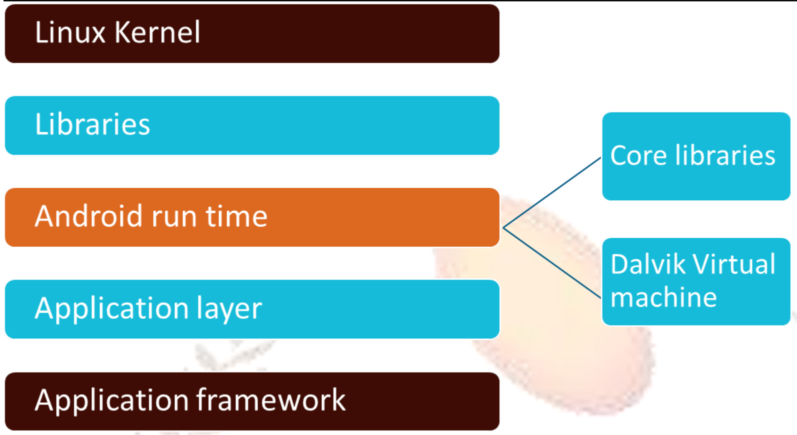
Accessibility: Other aids of supports available for the people those are having hearing disabilities and talk back solutions for non-vision people. Voice actions support for calling, texting, and navigation controls are available in Android 2.2 onwards. As of Android 4.1, Google has expanded Voice Actions with the ability to talk back and read answers from Google's Knowledge Graph when queried with specific commands.

Tethering: Android supports tethering that allows a phone to act as a wireless/wired Wi-Fi hotspot. This was earlier supported by third-party applications before Android 2.2. The Tethering module shares an Android device's internet connection with other connected client devices, which can connect to tethering devices over Wi-Fi, USB, Bluetooth, or Ethernet

Screen capture: If you press the volume and power button together it start capturing the screenshot. Earlier to version 4.0 this was done with the help of third party software.

External storage: Most Android devices include microSD slot can read microSD cards formatted with FAT32, Ext3 or Ext4 file system. To allow use of high-capacity storage media such as USB flash drives and USB HDDs, many Android tablets also include USB 'A' receptacle. Storage formatted with FAT32 is handled by Linux Kernel VFAT driver, while 3rd party solutions are required to handle other popular file systems such as NTFS, HFS Plus and exFAT.

Architecture of Android OS:



In order to meet the requirements of any Android device, the Android architecture incorporates a wide range of components. Android is built on top of an open-source Linux kernel that includes a large number of C++ libraries that are made available via an application framework's services. The Linux Kernel is the core of the smartphone's operating system, and the Dalvik Virtual Machine (DVM) is the platform on which Android apps operate. The components of android os are linux kernel, libraries, android run time , application layer and application frame work.

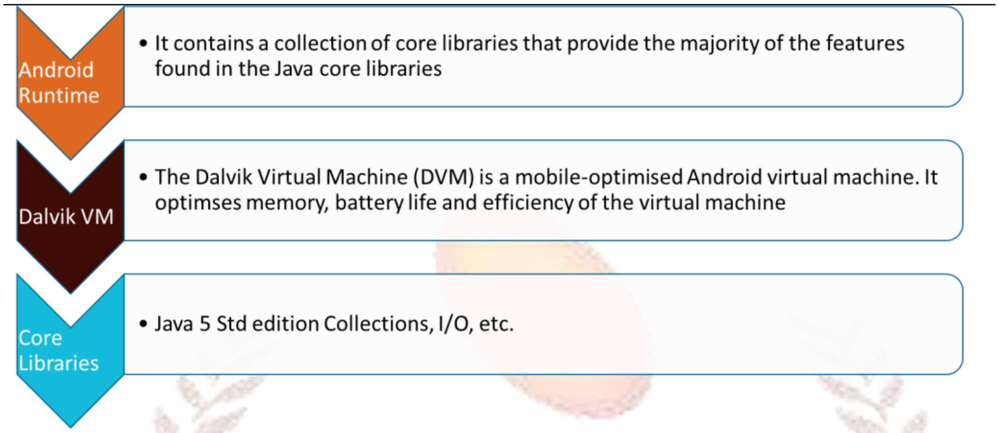
Application of framework architecture:

The framework architectures are created to make the reuse of components as simple as possible, the different layers of application framework and its description is as follows: Activity Manger : which manges application life cycle, android provides a set of core applications like email client, sms programme , calender, maps, contacts and much more. All the mobile applications are written using the java language lie Netflix, uber , amazon and so on.

Android Libraries:

The libraries used in Android operating system. Dependency injection is one of the most popular apps in Android development since it facilitates the creation of a solid library architecture for Android apps. Dagger is a well-known dependency injection library. To aid in android development, the Platform Libraries provide a variety of C/C++ core libraries and Java based libraries including Media, Graphics, Surface Manager, OpenGL, etc. A wide variety of audio and video file types can be played and recorded with the help of the media library. Both SGL and OpenGL are used for 2D and 3D computer graphics, and they are cross-language and cross-platform application programme interfaces (APIs).

Android Runtime:



Android Runtime

- It contains a collection of core libraries that provide the majority of the features found in the Java core libraries

Dalvik VM

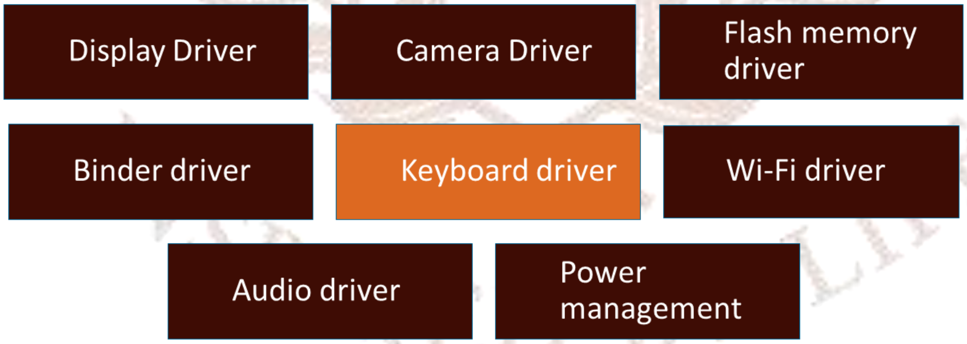
- The Dalvik Virtual Machine (DVM) is a mobile-optimised Android virtual machine. It optimses memory, battery life and efficiency of the virtual machine

Core Libraries

- Java 5 Std edition Collections, I/O, etc.

Linux –Kernel:

Linux kernel and the drivers present in it. The Linux 2.6 kernel is used in Android architectures. It aids in the management of security, memory, processes, the network stack and other critical issues. The Linux kernel is the backbone of the Android platform. The Linux kernel is used by the Android Runtime (ART), for instance, to perform essential tasks like threading and low-level memory management.Android's adoption of the Linux kernel enables it to make use of essential security features and enables device manufacturers to create hardware drivers for a widely used kernel



Limitation of Android OS:

- In comparison to standard mobile phones, Android OS consumes a greater amount of battery,
- Making source code open to the public immediately attracts the attention of hackers ans also it has been discovered that there are security issues.
- Google is quite lenient and may delete your developer profile if it finds certain problems with your program.
- One of the drawbacks of an Android phone is the software update. You cannot update your smartphone to a newer version of Android if it is running an outdated version of the operating system.
- Numerous smartphones with older Android versions are available on the market. You must purchase one of the new Android smartphone flagships if you want to use the most recent version of Android.

Self Assessment Questions

1. ___________________provides the tools for Android platform.
2. Android uses the___________________lightweight relational database for its storage.
3. IDEN NFC, GSM/EDGE connectivity technology supported by the Android OS. State [True/False].
4. In Android multitasking of application done through_______________.

<u>Development of mobile Application:</u>

After knowing the definition of mobile application development , let us understand the importance of mobile application and its features. A mobile application development platform (MADP) is a piece of software that helps businesses rapidly prototype, test, and release apps for mobile devices like smartphones and tablets. The following are some of the important features :

- The Ionic Framework is an open source software that is fully free to use.
- Develops a progressive online interface and a native smartphone app for every big app store.

- It offers excellent tools and services for developing interactive apps .
- Web developers will create apps for all major app stores with a single code base

The various types of platforms are as follows:

- Knoy App platform which is used as a multi-channel app development platform. You may create an Omni-channel, cross-platform app with the help of AppPlatform.
- Android studio allows developers to create apps quickly on any Android device. A smart editor for computer code. Several previews in one API. Put on OS Pairing. Adaptable structure.
- Apple Xcode –all registered users of the Apple Developer Program can now access Xcode Cloud. The Xcode Cloud is an integrated solution for continuous integration and delivery.
- Create native, hybrid, or mobile web apps that run on several platforms with the help of open-source framework Appcelerator Titanium. This includes operating systems like iOS and Android.
- Kinvey With the help of a cloud-based backend and front-end software development kits (SDKs), Kinvey is a highly productive serverless application development platform.
- Appzillon is an omnichannel digital banking platform that requires little to no coding and is hosted in the cloud. These solutions, which are based on Appzillon, improve the onboarding process and app delivery for financial institutions without sacrificing quality of service.
- Xamarin is a free and open-source framework for creating high-performance mobile, desktop, and server apps in. NET. Xamarin is an abstraction layer that controls how common code interacts with platform-specific code.

14.3 Platform to Design Mobile Application

To develop mobile applications we need the development tools and environment. In this section we are going to discuss two different platforms which support mobile application

- App Inventor for Android
- Android SDK

14.3.1 App Inventor for Android

App Inventor for Android is an application launched by Google, now the maintenance is under the control of MIT (Massachusetts Institute of Technology). It is a visual programming language for developing applications for the Android mobile computing platform. It is easy to develop an application for those who are familiar with any other programming language. It allows the programmer to design the application visually, it supports with drag and drop actions like .NET visual and Tersus platform as we discussed earlier. Here the developer can easily drag and drop the existing objects to design the application that are run on Android OS and in turn runs on many mobile devices. In recent years, an increasing number of individuals all around the world have gained access to the internet. In turn, this has prompted business owners and anybody else hoping to reach a wide audience to embrace mobile and desktop apps. It's easier for customers to navigate the supplier's information with these apps, and the provider has more options for presenting their material than they had with websites.

If you look at the history of this application, it was made available through request on July 12, 2010, and released publicly on December 15, 2010. Google terminated App Inventor on December 31, 2011. The MIT Center for Mobile Learning is now supporting it under the name MIT App Inventor. There was thorough research done on educational computing and Google online development environment before developing this App Inventor for Android.

Now we are going to discuss the key areas that help you design the simple applications. In this section we are going to discuss the key elements of App inventor, the component designer and the block editor.

App Inventor Environment

App inventor application generally run on the browser but few software needs to be downloaded to your desktop and few settings needs to be done in your mobile devices also. As shown in figure 14.1 the App Inventor environment will have three key areas they are

- ➢ Component Designer
- ➢ Block Editor

- Android Emulator

Component Designer is going to act as a container where you can select your components that are require for your application and can set the properties accordingly.

Block Editor runs in a separate window through which you can set the behavior of the component.

The application developed through the App inventor can be tested using your Android device some time due to device driver setups it may create an issue. In that situation you can use the Android emulator that comes along with the App Inventor package.

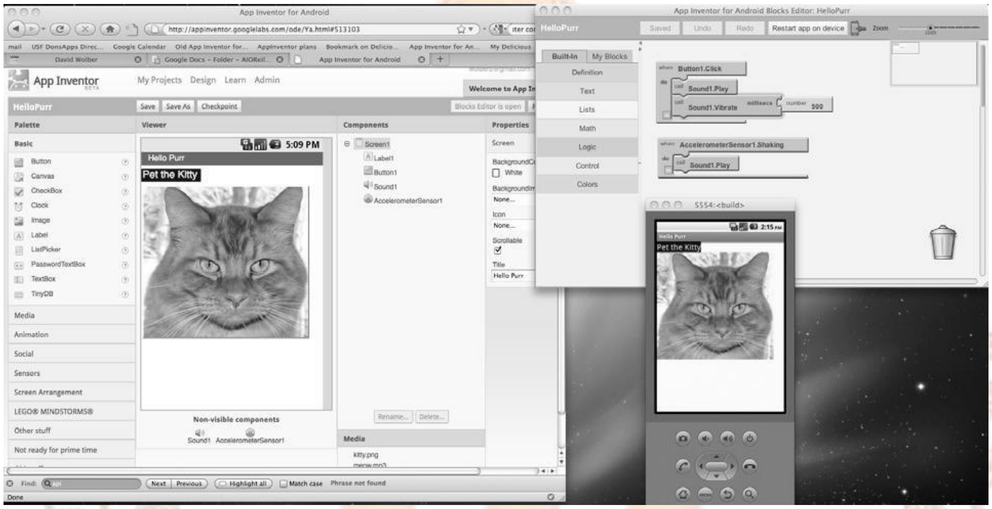


Fig. 14.1: App Inventor IDE

If you are using the App Inventor first time the project page will be empty, to create a new project click New at left top of the page and enter the project name and click ok. Now the component designer window will be opened, you click on the top right to open the blocks editor. As we discussed earlier the blocks editor opens separately with the support of Java web start. Now the block editor will appear on the screen as shown on the figure 14.2 through which you can connect your android device.

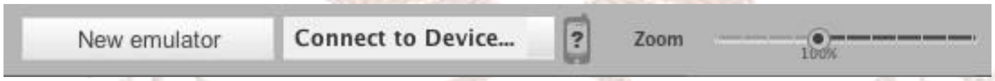


Fig. 14.2: Connecting Android device

Through USB you can connect your android device with the computer and click on “connect to device”. Otherwise if you want to use the Android emulators click on “New emulator” and wait for few seconds to load the Android emulator application then click on “connect to Device”. Now your App Inventor IDE (Integrate Developemnt Environment) is ready for developing application.

Developing mobile application using App Inventor

Now we are going to see how to design a mobile application using App Inventor. Create a new project using the New project button and name the project tile as “Project 1”. Before proceeding with the application design let us have an idea about the App inventor projects, it has a collection of building blocks called components. These components includes the basic controls like text label to display the title, text box to accept and display the calculated result buttons to trigger events, images, camera component to capture photos etc. The specialized components include the components that support the media and animation displays and the components that are acting as interface to device sensors.

Now in this example, we are going to design a mobile application that invoke camera to capture the image and display the same on the screen. For this we are going to use three different components like Camera, Image and Button. The expected behavior of the application is, when the user clicks on the Button it start the camera on your mobile device through which the use can capture the image and the same can be displayed through the Image component.

Following are the steps to design the component designer viewer.

1. Drag the camera component to the viewer, available under the Non-visible component below the viewer in the palette. It comes with the default name “camera1” if it requires the name can be changed by the user through its property.
2. Drags the Button component to the viewer and change its text as “Click”.
3. Drag the Image controller to the viewer.
4. Also you can design the back ground of the screen by changing its Title, background color, and orientation etc. through the property of the screen component.

After this the screen looks as shown in figure 14.3.

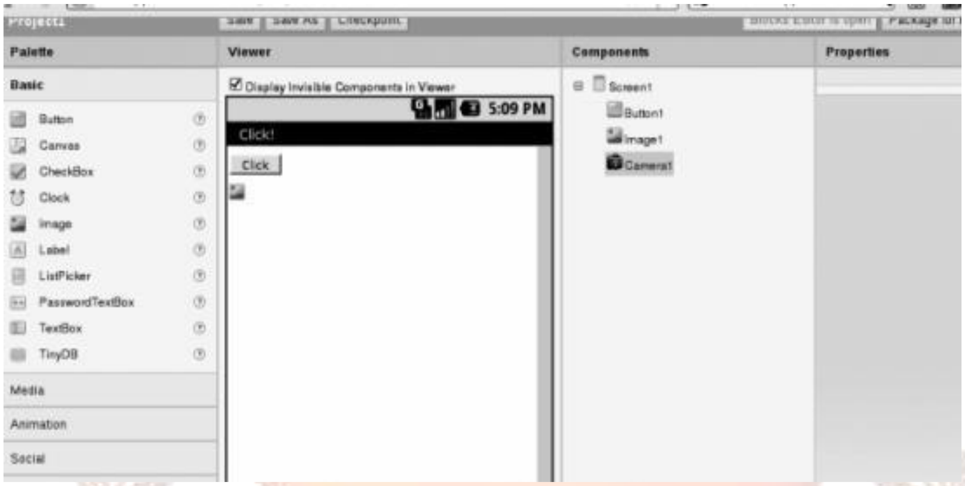


Fig. 14.3: Screen with image, Button and camera components

Once the design part is over we can now move to the coding part of the application. The goal of this project is to activate the camera when the button is clicked by the user. The block editor for our sample application will appear as shown in figure 14.4.

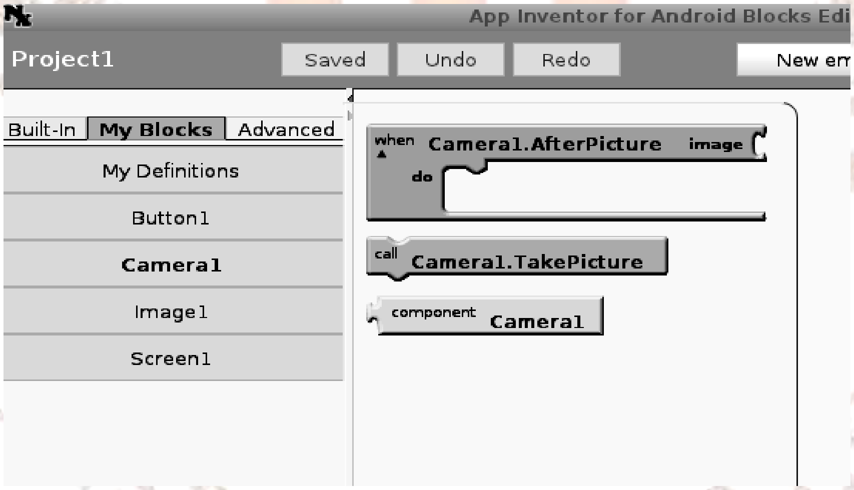


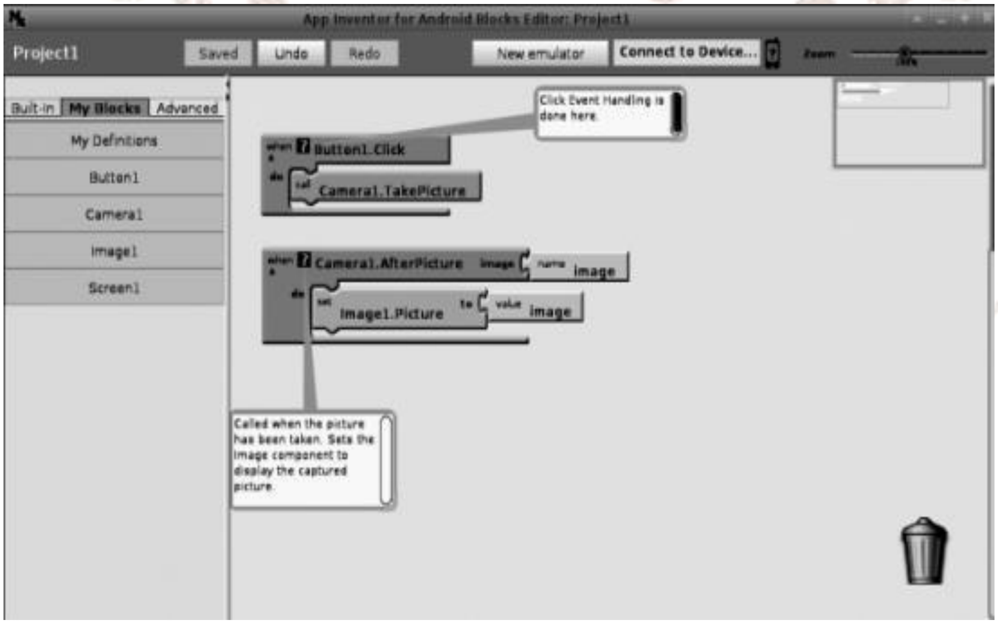
Fig. 14.4: Block Editor for sample application

We will start with the button click event require to trigger the camera component to activate. About event we are already familiar with the visual programming here the same concept will be achieved with the help of “When Button1.Click”. This needs to be dragged from the Blocks pane on the left side of the window. Next the action is, when the button is clicked you wanted to activate the camera this can be achieved through “call

Camera1.Takepicture" block inside the previous block. Next step in the application is to display the picture in the picture component to do this we need to insert the "When camera1.AfterPciture" in to the editor. Then set the "Image1.Picture" to the location of the saved image.

Now the user interface is ready with the application logic we can go for testing. Go to right side of the Designer window and click on "Package for Phone-> Download to this computer". Now it will initiate the down load the Android package ".apk file" for our sample project. Now the application is ready to install in the Android device, you can install and test the application.

So we completed one simple mobile application that are designed and deployed in the Android application. From this example we must understand that the App inventor is the event driven programming these event can be the user click or the text message that the application may receive. In our sample when the user clicks the button the event occurred thus camera activated again when the camera finishes its work, the response code uses the image location to display it using image component. Thus it is proved that the entire component has associated behavior, properties and methods. This can be seen by clicking the component in the Block editor Figure 14.5 shows the methods and behavior of the camera component.



Click Event Handling is done here.

Called when the picture has been taken. Sets the image component to display the captured picture.

Fig. 14.5: Methods and behavior of camera component

1. **Knoy App Platform:**

It is an excellent method for creating native and cross-platform applications. The drag-and-drop functionality provided by Kony Marketplace or their API libraries is the most appealing feature of Kony. API connectivity should be tested twice by developers. Kinvey is a development platform that includes a data store, data integration, single sign-on integration, and file storage, among other features. Kinvey frees up developers to concentrate on what really matters: the app's user experience (UX) and its business logic. This method takes advantage of Kinvey's pre-built components to boost development productivity and allow for higher quality apps.

Pros and cons about the platform :

- Let us discuss about the pros and cons of the Kony app development platform. The advantage part of kony app is that it contains prototype applications to assist new developers in getting started and the backend layout is diverse so can be used in vast area

- Similarly it also has some disadvantage also like On the internet, there is a scarcity of Kony documents and support and As a result, it is less likely to be used by newcomers than other pages

2. **Android Studio Platform:**

Android studio is one of the platform , When creating an app for the Android platform, you must use Android Studio, the official IDE. Code editing and other developer tools from the popular Java IDE IntelliJ IDEA are built right in.according to the platform on which we want to run the app, Android Studio has a variety of shortcuts and SDK. API networking is also simplified and Android Studio adapts to SDK functions and their use and Android Studio provides the fastest software on creating apps for any Android device

Pros and cons about the platform :

- we will understand some benefits and disadvantages of the android app development platform. Some Pros of android studio It has simple shortcuts and working with xml is simple
- Every function is explained in great detail so that we can put it to good use
-

Similarly it also has few cons like to run, it necessitates the use of incredibly powerful hardware

- It is likely that your computer's slowness is due to hardware issues

- The Java code in the project gets more complex as it increases in size

3. Apple Xcode Platform:

This platform is basicaaly used for creating the IOS apps. This smartphone software development tool for iOS developers uses Swift, Objective-C, C and C++ as well as other languages

XCode is a fantastic platform and all IOS applications run on iPhones and iPads. There are not many codes in it and they are all short.

Pros and cons of Xcode:

- Advantage part of Xcode are The drag-and-drop feature is widely used in it and it is XCode's most useful feature in terms of software development.It contains a small number of codes that are not very long.
- The disadvantage part of Xcode is The developers must build the licenses for the utilities that they want to use in the application.
  They have to apply all utilities to it and wait for Apple's permission once more.

4. Appcelerator platfrom:

Let us understand the Appcelerator mobile app development platform and its pros and cons. The Appcelerator framework simplifies the integration of JavaScript and backend programming for the production of native apps that can run on several platforms. The percentage of our code that can run on other systems ranges between 60-90%. JavaScript application programming interfaces (APIs) are used for iOS, Android, Windows, BlackBerry, and HTML5.

Pros and cons of Appcelerator:

- Advantage is We can build a full app in only a few hours and with just a few lines of code due to significant prototyping and JSON is the JavaScript encoder and Appcelerator fully supports JavaScript APIs

- Were as the disadvantage is that Documentation is not necessarily flawless and we frequently run into problems with it and There is a great deal of trial and error about it. Special coding is needed for both IOS and Android

5. Kinvey Platform:

Using a cloud-based backend and front-end software development kits (SDKs), Kinvey is a highly productive serverless application development platform. Kinvey is a development platform that includes a data store, data integration, single sign-on integration, and file storage, among many other features. Kinvey frees up developers to concentrate on what really matters: the app's user experience (UX) and its business logic. Using Kinvey's pre-built components, this method boosts developer productivity and strives to enable higher-quality projects. We will discuss about its pros and cons.

Pros and cons of Kinvey:

- Pros are as follows It is easy to connect an app to BaaS, which is a cloud backend provider that can be counted on for software development projects. It also has a package that helps 100 users to store 2GB of data and get 5 million updates.
- The cons are considered to be Since you wrote some code that could be exploited in the future, you run the risk of coding damage.

It is not ideal for bigger projects because we will run into issues and our results may be lost during testing.

6. Appzillion platform:

Appzillon is an open source, low-code platform for omnichannel digital banking in the cloud. These solutions, developed on top of Appzillon, expedite the onboarding process and software deployment for financial institutions, all while improving the overall user experience. Join us in our mission to improve banking institutions' bottom lines by attracting and retaining customers through superior service.

Pros and cons of the appzillion:

The pros are i

The cons are as follows

Using just a web browser, an Android device, or an emulator, you can create apps for Android phones with App Inventor. The servers of App Inventor save your work and make it easy to keep track of your many projects. The App Inventor Designer is where you'll be doing most of your work when making an app. It is considered to be as a stepping stone in standard android development , single screen use , low functionality, its can be used for educational purpose and also for personal use

Steps for creating an App:

1. First step is Using your Google Account, sign up for App Inventor,
2. Then download additional app inventor tools, third step make a new project on the website and
3. next step is to drag and drop UI elements and components into the designer. We will continue further in the next slide.
4. With the continuation of the previous steps , we have to activate the block editor , then drag and drop the capabilities ,
5. once this is done next compilation , analyzation and debugging is carried out finally upload the APK to the internet or save it to your phone .

These are the detailed step by step creation of an application using App inventor.

14.3.2 Android SDK

A software development kit, that enables developers to create applications for the Android platform**.** Whenever the Google releases the Android OS, corresponding Android SDK will also get released.

The Android SDK provides the tools and the APIs (Application Program Interface) that are necessary to develop application on the Android platform using Java programming language. Even though the Android application written in the Java language, there is no Java Virtual Machine in the platform, and thus the Java byte code is not executed. Java classes are recompiled into Dalvik executable and run on a Dalvik virtual machine; it is the modified VM for Android.

For developers, Android SDK provides a rich set of tools, including debugger, libraries, handset emulator, documentation, sample code, and tutorials. Android applications can be easily developed using Eclipse (Android's official development platform) with the help of a plug-in called Android Development Tools (ADT).

Eclipses SDK, Android SDK, Android Development Tool (ADT), needs to be installed before you start with the application development.

Languages usedin android : The Android software development kit (SDK) is a set of resources for programmers working on Android-based software. The Android Software Development Kit (SDK) is a collection of resources for creating Android-based software. This software development kit (SDK) provides access to a set of resources vital to the creation of Android apps and guarantees a streamlined development experience. The

languages used in SDK are listed aas follows
1.Kotlin
2.Java
3.C++

Advantage and disadvantage of Android SDK:

1. let us discuss Let us discuss the disadvantage of the disadvantages of Android SDK. Slower installation, system slowness, greater hardware requirements, and increased RAM usage are some of the more frequently mentioned drawbacks of utilising Android Studio. Android consumers spending activities, we have Security issues , we also face Disparity in operating system acceptance, then we will also face copy right related issues.
2. The advantages of using Android SDK. It is the easiest and most dependable software development kit on the market. It is free to use and assists in the development of wonderful Android apps, It has remarkable flexibility and provides consumers with a cost-effective solution, It enables programmers to access information quickly, and It also provides you with the precise information you need. The SDK tools are simple to use.

Task performed by android:

You may use the Android tool to manage AVDs, activities and SDK components. The following are some of the activities that this programming tool can perform: The Android application allows you to control AVDs, activities, and SDK parts. The following are some of the things that can be done with this programming instrument: Digital Android devices can be controlled (AVD), Make and maintain Android projects and To your SDK, add new platform add-ons and documents

DDMS: this tool may be used to debug android apps , the basics of DDMS. If you're having trouble with your Android device, you can use the Dalvik Debug Monitor Service (DDMS). When installing the Android SDK, you'll also need to install the Dalvik Debug Monitor Service. Port forwarding, device screen capture, thread and heap monitoring, and radio state data are only some of the features offered by the DDMS. Message development, call spoofing, screenshot capturing and discovery of internal threads and file systems may all be part of the service. Whether on an emulator or a real Android device, developers can use the Dalvik Debug Monitor Service to find and fix software flaws.

Running and working of DDMS:

Running: From android studio Tools>Android>Android device Monitor

- Running From android studio has to follow these steps like Tools, select Android then Android device Monitor, then for working with DDMS As DDMS starts up, it connects to the Android Debug Bridge (adb).

- When a computer is attached, a VM monitoring service between adb and DDMS is created, which notifies DDMS when a VM on the system begins or ends

Some fo the features of DDMS are some of the features offered by DDMS. Like it is used in making call , making sms and capturing screen shots and much more

1. **SqLite3:**

Sqlite3 is a command-line programme that lets you manage SQLite databases created by Android applications. The method can also be used to execute SQL statements on the fly. There are two methods for accessing SQLite. SQLite databases built by Android apps can be managed with the help of Sqlite3, a command-line software. SQL statements can also be run dynamically with this approach. SQLite can be accessed in two different ways. SQLite holds information in a text file on a device and is available as open source software. SQLite database support is built into Android.All the functionalities of a relational database are available in SQLite. No special connections (JDBC, ODBC, etc.) are required to access this database.

2. Proguard : Proguard reduces the size of the application, it eliminates the unused classes and methods that add to an Android app's 64K method limit and also It is impossible to reverse engineer the software since the implementation is abstracted

The different features of proguard:

- The code is pre-verified for Java Micro Edition as well as Java 6 and higher For smart phones, Blu-ray players, set-top boxes and other small platforms, Java applications are optimized and abstracted
- Reduces the time it takes to download and start Android applications and improves their performance on mobile devices

Android Debug Bridge:

- Communication between a computer and an Android device is facilitated via the Android Debug Bridge (adb), a powerful command-line utility.
- The adb command allows you to install and debug applications on your device,

among other things.

- To execute various commands on a device, adb grants entry to a Unix shell.
- The ADB is a common protocol for connecting with an Android-enabled device, such as a smartphone, tablet, smart watch or set-top box We can do stuff like update applications outside the Play Store, debug apps and unlock secret features on an Android smartphone

Components of Debug Bridge: There are three parts to this client-server application: who issues orders, or a client

1. Client
2. Deamon
3. Servere

Your development machine serves as the client. From a terminal, you can launch a client by using the adb command.

A background process (adbd) that executes other programmes. The daemon is constantly active in the background on all connected devices. One that handles daemon-client interactions as a server.

On your development PC, the server operates invisibly as a background process.The Android Debug Bridge (adb) is a command-line interface that allows you to communicate with Android devices

3.

Android virtual Device: Android phone emulator or Android Virtual device needs to installed. It is used to run and test the application.

1. In Eclipse, choose *Window > Android SDK and AVD Manager.*
2. Select *Virtual Devices*.
3. Click *New*. The *Create New AVD* dialog appears.
4. Type the name of the AVD
5. Choose a target. The target is the platform to be run on the emulator.
6. Click *Create AVD*

Once the AVD is launched the emulator appears shown in figure 14 .6 It gives the real feel of android mobile phone with keyboard and multi-touch support.

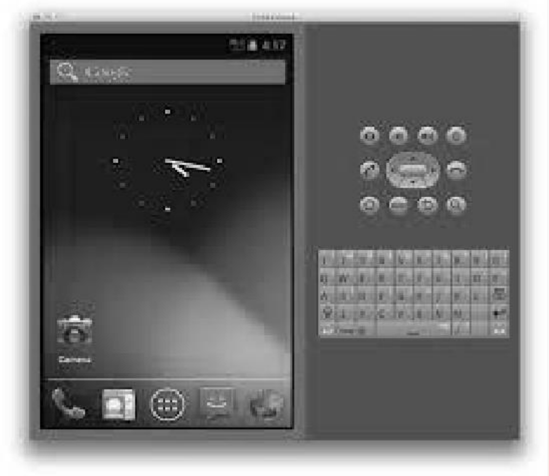
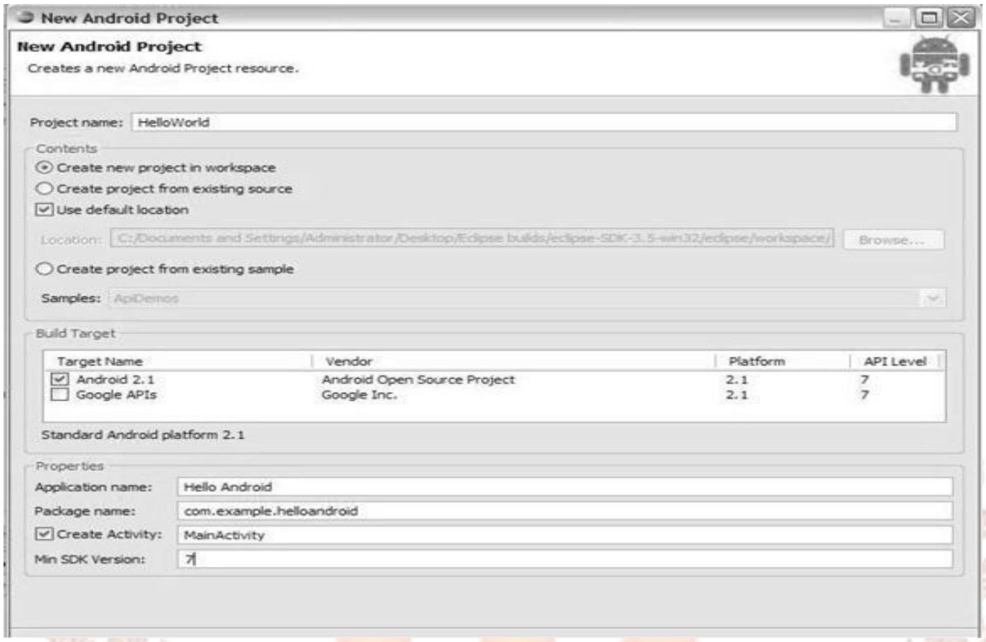


Fig. 14.6: Android virtual device(emulator)

1. From Eclipse, select *File > New > Project*. You can see a folder called *Android* if you expand *Android project*.-> *Next.*
2. You will be prompted to fill in the following details in a form as shown in fig 14.7.
   - *Project name:* Name of the project can be "HelloWorld."
   - *Application name:* This is the name that will appear everywhere in your Android device, can be something like "Hello Android."
   - *Package name: "*com.example.helloandroid" or your own private namespace. Naming convention should follow the java rule
   - *Create activity* :Is the name for the class stub that will be generated by ADT. An activity is simply a class that can run and do some work. An application moves from one activity to another by calling a method known as startActivity() or startSubActivity().
   - *Min SDK version* – This specifies the minimum API level required by your application.



New Android Project

New Android Project

Creates a new Android Project resource.

Project name: HelloWorld

Contents

● Create new project in workspace

○ Create project from existing source

☑ Use default location

Location: C:/Documents and Settings/Administrator/Desktop/Eclipse builds/eclipse-SDK-3.5-win32/eclipse/workspace/ Browse...

○ Create project from existing sample

Samples: ApiDemos

Build Target

Target Name	Vendor	Platform	API Level
☑ Android 2.1	Android Open Source Project	2.1	7
□ Google APIs	Google Inc.	2.1	7

Standard Android platform 2.1

Properties

Application name: Hello Android

Package name: com.example.helloandroid

☑ Create Activity: MainActivity

Min SDK Version: 7

Fig. 14.7: New Android project window

Before you start executing your application, you should know few files and directories that are in Android project.

AndroidManifest.xml: The manifest file describes the fundamental characteristics of the app and defines each of its components. Important element in manifest is, it should include is the <uses-sdk>element. This declares your app's compatibility with different Android versions using the *android:minSdkVersion* and *android:targetSdkVersion* attributes. For your first app the used-sdk code is

```
<manifest
xmlns:android="http://schemas.android.com/apk/res/android" ... >
   <uses-sdk android:minSdkVersion="7"
android:targetSdkVersion="17" />
   ...
</manifest>
```

It is better to set the android:targetSdkVersion as high as possible and test your app on the corresponding platform version.

src/ : Directory for your app's main source files. By default, it includes an Activity class that runs when your app is launched using the app icon.

res/: Contains several sub-directories for app resources. Few are listed below

drawable-hdpi/ : Directory for drawable objects (such as bitmaps) that are designed for high-density (hdpi) screens. Other drawable directories contain assets designed for other screen densities.

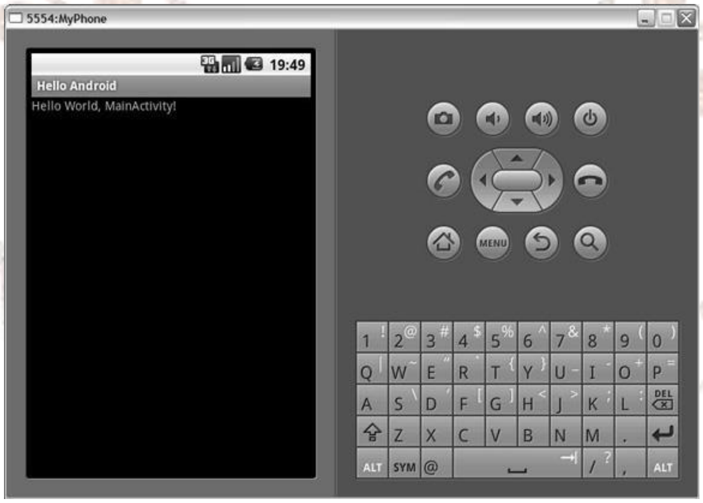
layout/: Directory for files that define your app's user interface.

values/: Directory for other various XML files that contain a collection of resources, such as string and color definitions.

Now, if you build and run the default Android app, Since no other coding part is done in our sample application the default Activityclass starts and loads a layout file that says "Hello World."

Running the app on the emulator

For deploying and running the app, you need to define a run configuration. Select *Open* > *Run* > *Dialog* or select *Android application* on the toolbar within Eclipse, and. Click *New* and fill the name of the run configuration. Specify the name of your project and default activity. Then in the target tab, choose some desired emulator settings and specify an AVD you want to start. Click *Run*. The out will be shown as shown in figure 14.8.



5554:MyPhone

19:49

Hello Android

Hello World, MainActivity!

Fig. 14.8: output Android application on emulator

Self Assessment Questions

5. App inventor for Android initially launched by__________________.
6. ____________________ window is used to develop the logic code of an application
   a. component designer b. block editor
   c. Android emulator d. Development Tool Kit
7. App Inventor projects has the collection of building blocks called __________________.
8. Write event that require to invoke when the Button is clicked.
9. __________________is the files directory defines the app interface.

14.4 Components to Enable Android Function

The basic blocks consist of various components that are necessary to design Android application. Different components at various locations help to design and allow your application to execute. Majorly it act as entry points for the user but not all the components does this, remaining components will depend on each other to do the specific assigned task. Each component that is defined in the building blocks decides the overall behavior of the application. The components are majorly divided in to four types.

- Activities
- Services
- Content providers
- Broadcast receivers

Each type of components are assigned with specific purpose with unique lifestyle that gives an idea about the creation and destroys of those components. Now we are going to discuss those four components in detail.

Activities

An activity can represent a single screen with a user interface. For example, an email application may have multiple activity one among may be listing the new emails, another activity may be composing an email, and another activity for reading emails etc. Though all the modules work together to form a consistent user experience in the email application, each one is independent in its work nature. As such, a different application can start any one of these activities (if the email application allows it). For example, a camera application can start the activity in the email application that composes new mail, the user may want to share a picture. Each activity is implemented as a subclass of Activity.

Services

A service is a component that runs in the background to perform long-running operations or provide support to perform remote processes. A service does not provide a user interface. For example, a service might play music in the background while the user is in a different application, or it might fetch data over the network without blocking user interaction with an activity. Another component, such as an activity, can start the service and let it run or bind to it in order to interact with it. A service can be implemented as a subclass of Service.

Content providers

A content provider manages a shared set of application data. You can store data in the file system; say for example in SQLite database, on the web, or any other persistent storage location where your application can access. Through the content provider, other applications can query or even modify the data provided the content provider allows for updation. For example, the Android system provides a content provider that manages the user's contact information. As such, any application with the proper permissions can query part of the content provider to read and write information about a particular person. Content providers are also useful for reading and writing data that is private to your application and not shared. For example, the Note Pad sample application uses a content provider to save notes. A content provider is implemented as a subclass of ContentProvider and must implement a standard set of APIs that enable other applications to perform transactions.

Broadcast receivers

A broadcast receiver is a component that responds to system-wide broadcast announcements. Broadcasts generally originate from the system, for example, a broadcast announcing that the screen has turned off, the battery is low, or a picture was captured. Similarly applications can also initiate broadcasts for example, to let other applications know that some data has been downloaded to the device and is available for them to use. Although broadcast receivers don't display a user interface, they may create a status bar notification to alert the user when a broadcast event occurs. More commonly, though, a broadcast receiver is just a "gateway" to other components and is intended to do a very minimal amount of work. For instance, it might initiate a service to perform some work based on the event. When an Android smartphone boots up, when a message or call comes in, when the user toggles aeroplane mode, etc., these are all examples of broadcast events. To deal with such widespread occurrences, the system employs Broadcast Receivers. By using Broadcast Receivers, we may subscribe to system and application events and be notified when they occur. Broadcast receivers can be roughly divided into two categories:

Receivers declared in the manifest file continue to function even when the app is closed; they are called "static" broadcast receivers.

In order for dynamic broadcast receivers to function, the app must either be running or minimised.

A unique aspect of the Android system design is that one application can start another application’s component. For example, if you want the user to capture a photo with the device camera, there's probably another application that does that and your application can use it, instead of developing an activity to capture a photo yourself. You don't need to incorporate or even link to the code from the camera application. Instead, you can simply start the activity in the camera application that captures a photo. When complete, the photo is even returned to your application so you can use it. To the user, it seems as if the camera is actually a part of your application.

When the system starts a component, it starts the process for that application (if it's not already running) and instantiates the classes needed for the component. For example, if your application starts the activity in the camera application that captures a photo, that activity runs in the process that belongs to the camera application, not in your application's process. Therefore, unlike applications on most other systems, Android applications don't have a single entry point. For example there's no main () function to start the process like C or C++ language.

Because the system runs each application in a separate process with file permissions that restrict access to other applications, your application cannot directly activate a component from another application. The Android system, however, can. So, to activate a component in another application, you must deliver a message to the system that specifies your intent to start a particular component. The system then activates the component for you.

Self Assessment Questions

10. ______________________ can represent a single screen with a user interface.
11. An Activity is a component that runs in the background to perform operations. State [True/False].

14.5 Summary

- Mobile applications increase the efficiency of the users and Android OS become the most popular OS for mobile devices.
- The Android Software Development Kit (SDK) provides the tools and Application Programming Interfaces (APIs) necessary to begin developing applications.
- Storage, messaging, web browser, multiple language support, multi-touch, Bluetooth and screen capturing are some of features of Android OS.
- App Inventor for Android is a visual programming platform to develop mobile application.
- Android SDK is the Google product supports in developing mobile application
- Component Designer, Block editor and Android emulators are the three major components available in the App inventor IDE.
- The application developed through the App inventor visual platform can be test using Android mobile device or the Android emulator that comes with the App inventor package.
- Building block is one of the component in App inventor helps to define the logic codes of an application.
- The entire component in the component designer has associated with its behavior, properties and methods.
- Activities, services content providers and broadcast receivers are the major components that enable the Android functions.
- Location aware programming is the new dimension to the mobile application helps to deliver the online information based on their physical locations.

14.6 Terminal Questions

1. Explain about Android operating system in detail.
2. List and explain the features of Android operating system.
3. Describe the App inventor application platform.
4. Design a small mobile application using 2 text and button control.
5. Discuss the various components that enable the Android function.

14.7 Answers

Self Assessment Questions

1. Software development Kit
2. SQLite.
3. True
4. Allocating unique memory
5. Google.
6. b. block editor
7. Components.
8. When Button1.Click.
9. layout/
10. Activity.
11. False

Terminal Questions

1. Android is considered as the first truly open and comprehensive platform for mobile devices. For more details refer section 14.2.
2. Storage, messaging, web browser, multiple language support, multi-touch, screen capturing etc. are some of features of Android OS. For more details refer sub section 14.2.1.
3. App Inventor for Android is an application launched by Google, now maintained by MIT is a platform to design mobile application. For more details refer sub section 14.3.1.
4. Components can be picked from the palette, block editor can be used to develop the code and using android device you can test your application. For more details refer sub section 14.3.1.
5. Activities, services content providers and broadcast receivers are the major components. For more details refer section 14.4.

E-Reference:

- http://en.wikipedia.org/wiki/List_of_features_in_Android
- http://www.appinventor.org/apps/hellopurr/hellopurr.pdf
- http://www.slideshare.net/BrockButler/location-aware-mobile-applications
- http://www.developermarch.com/Open%20Source/exclusives/google-ide-for-the-non-coder-android-developer.html