Mobile Operating System

Beulah A.
AP/CSE

Introduction

- Smartphones
  - Phone calls, Video calls, Multimedia messages, take pictures, play media files, browse www, run web applications etc.
- This requires multiple tasks run on the device.
- Therefore a powerful operating system is necessary for every smartphone.
- The OS performs 2 main responsibilities:
  - Managing resources
  - Providing interfaces

Managing Resources

- Important feature → efficient utilization of the resources of the device by performing multiple tasks.
- The resources managed by OS
  - Processor, Memory files,
  - Various devices such as camera, speaker, keyboard, screen.
- A person listening to music, same time he can send sms, view email, search contacts etc...
- Even multiple tasks contend to use the same set of resources.

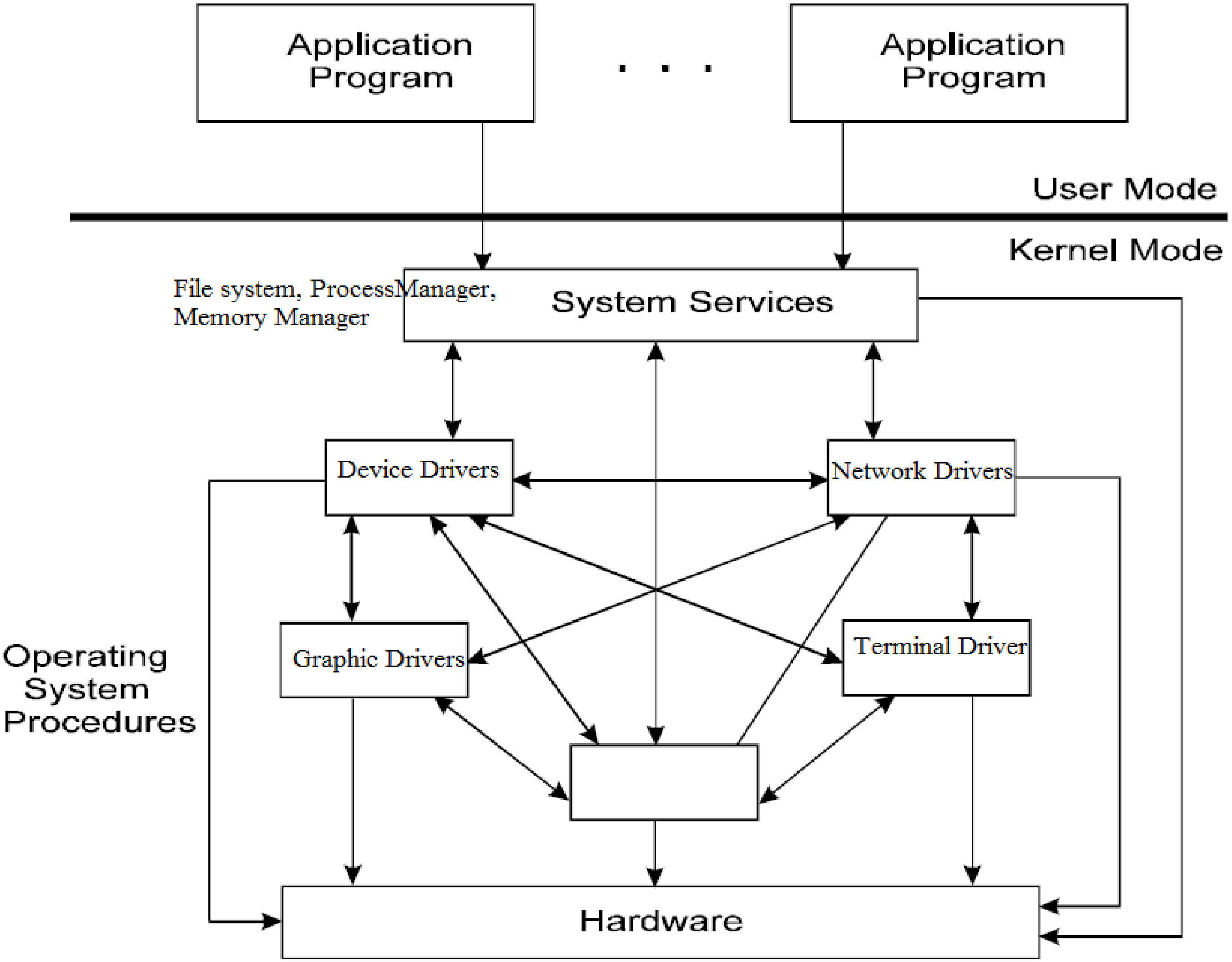
Providing Interfaces

- Mobile Os provides
  - A high interactive interface to the mobile user
  - Interfaces with several devices and networks
- Important interface are control, data, voice communications.
- Apart from it OS also takes care of recognizing inputs from the keyboard, sending outputs to the screen, interfacing with peripheral devices such as other mobile devices, computers, printers, etc.
- Keyboard in touch screens.
- Variations in screen resolution and size of mobile device.
- Underlying Os should handle all these interfaces.

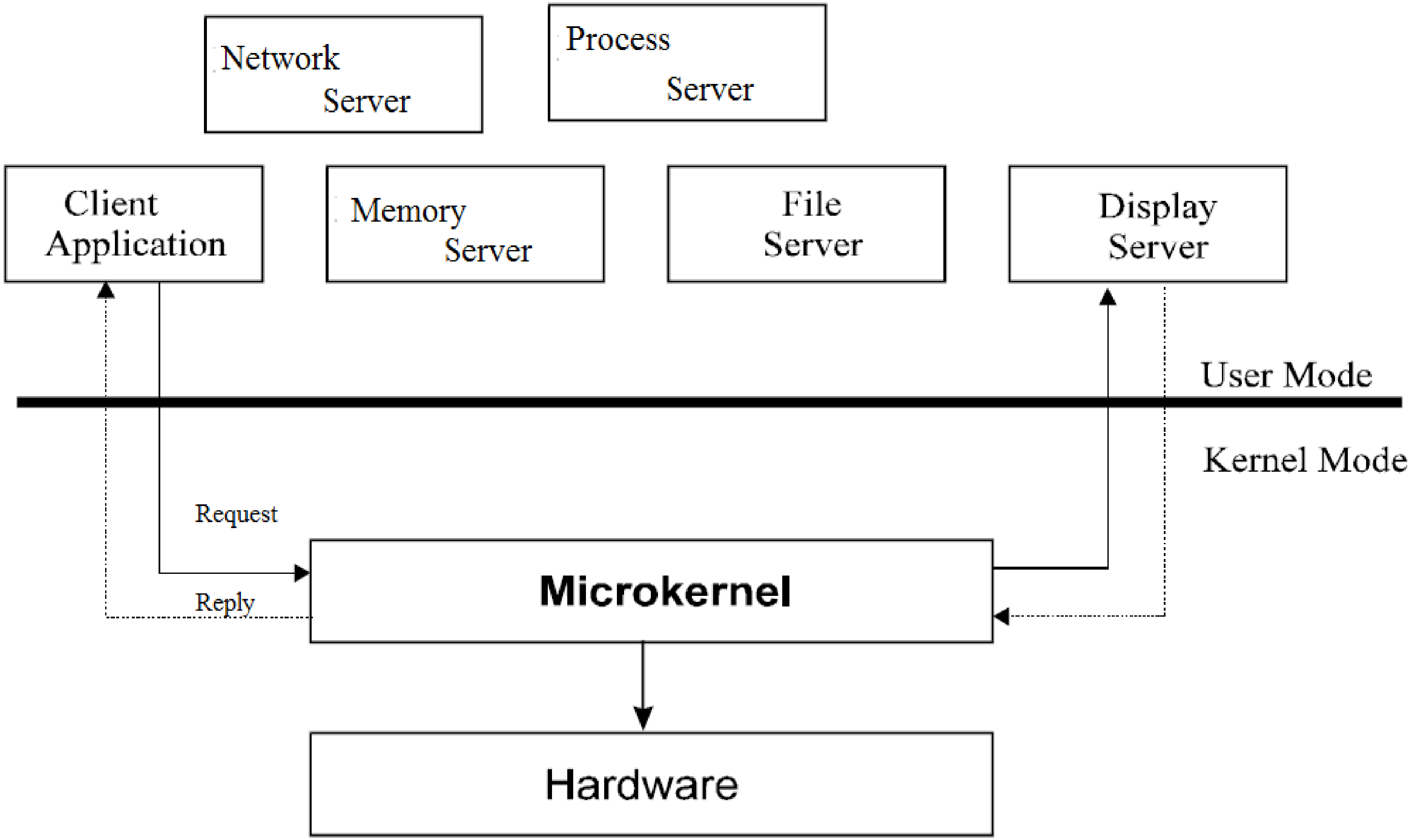
Mobile OS available in Market

- Symbian
- Android
- Windows
- Palm Os
- iOS
- Blackberry Os

Monolithic Kernel



Microkernel



Special Constraints and Requirements of Mobile OS

Beulah A.
AP/CSE

Special Constraints

- A mobile device needs to function in the presence of many types of constraints which are not present in a traditional computer.
- Ex: Mobile device is powered by limited energy stored in tiny battery. Therefore complex computations should be avoided. Also enter into a low power sleep mode as soon as power gets drained. (Such constraint not in traditional OS)
- To handle such constraints, is an important reason why the MOS needs to differ significantly from a traditional OS.

Special Constraints

- Limited Memory
  - Mobile device’s permanent and volatile memory is very less compared to laptop or desktop.
  - To handle this M OS should be small and it should provide rich functionalities to meet user demands.
  - Therefore size of kernel is an important feature to be considered.
- Limited Size Screen
  - Size of MD is small which limits the size of display screen.
  - M OS should provide easy interface to suit individual preferences, switching between menu and iconic interfaces, etc.

Special Constraints

- Miniature Keyboard
  - Small keypad or small sized display with touch screen mode.
  - Typing documents is difficult.
  - Therefore new facilities like word completion prompts, handwriting recognition, iconic commands etc.
- Limited Processing Power
  - ARM – Based processor.
  - Energy efficient, Powerful and cheaper when compared with laptop/desktop processors.
  - With the restricted processing power, memory, the M OS is made to provide only limited number of functionalities that are useful in the actual operation of the mobile device.

Special Constraints

- Limited Battery Power
  - Mobile device needs to be lightweight.
  - So have a small batter with recharging capacity.
  - Small battery should support long talk time without frequent recharge
- Limited and fluctuating bandwidth
  - Wireless medium – more noise – high bit error rate.
  - Variable bandwidth leads to fluctuation in speed of communication.
  - Movement of mobile device - handoff
  - Above leads to data loss.
  - Data caching, Pre fetching, Integration.
- Real time data streaming
  - Beyond 3G OS, real time data streaming such as mobile TV.

Special Service Requirements

- Support for Specific Communication Protocols
  - 1G, 2G 3G etc. Uses different communication protocol
  - Mobile device should support 2 or 3 generations.
  - To communicate with other devices (computers, printers etc) use specific protocols.(TCP/IP, Wireless LAN)
  - Other devices like headphones, USB drives etc.
- Support for Variety of Input Mechanism
  - Miniature keyboard, smart keyboard, stylus based input mechanism, touch screen.
  - Mobile OS needs to support these variety of input mechanism.

Special Service Requirements

- Compliance with Open Standards
  - Should provide open standard facility to develop innovative applications by third party developers.
- Extensive Library Support
  - Third party applications requires library support.
  - M OS should provide libraries to be called for email, SMS, MMS, Bluetooth, multimedia, user interface primitives, GSM/GPRS, etc.
- Support for Integrated Development Environment (IDE)
  - General purpose IDE such as Eclipse can be used to develop applications.
  - M OS can have their own IDE for effective Software development and good performance

Commercial Mobile OS

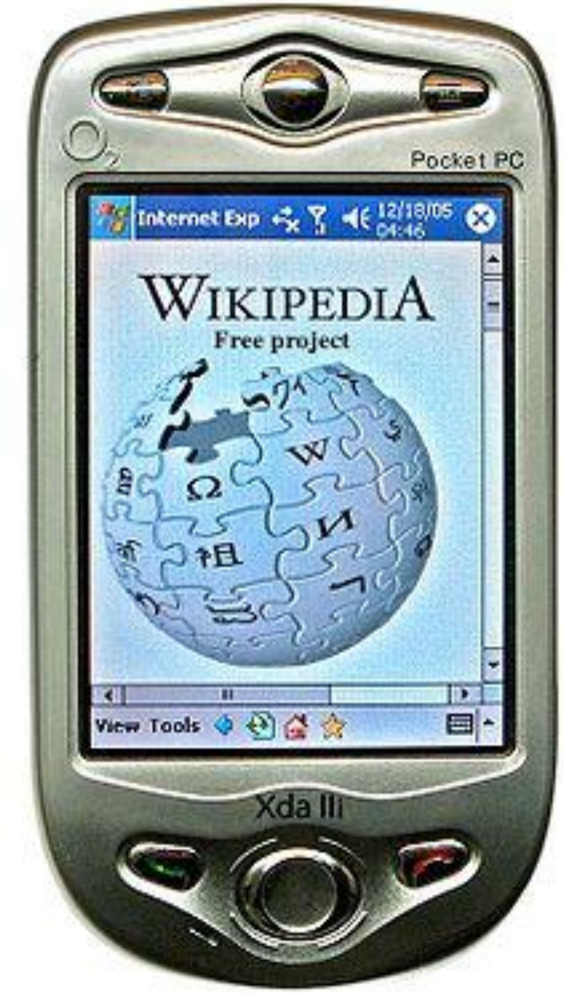
Beulah A.
AP/CSE

Popular Mobile OS

- Windows Mobile
- Palm OS
- Symbian OS
- iOS
- Android
- Blackberry OS

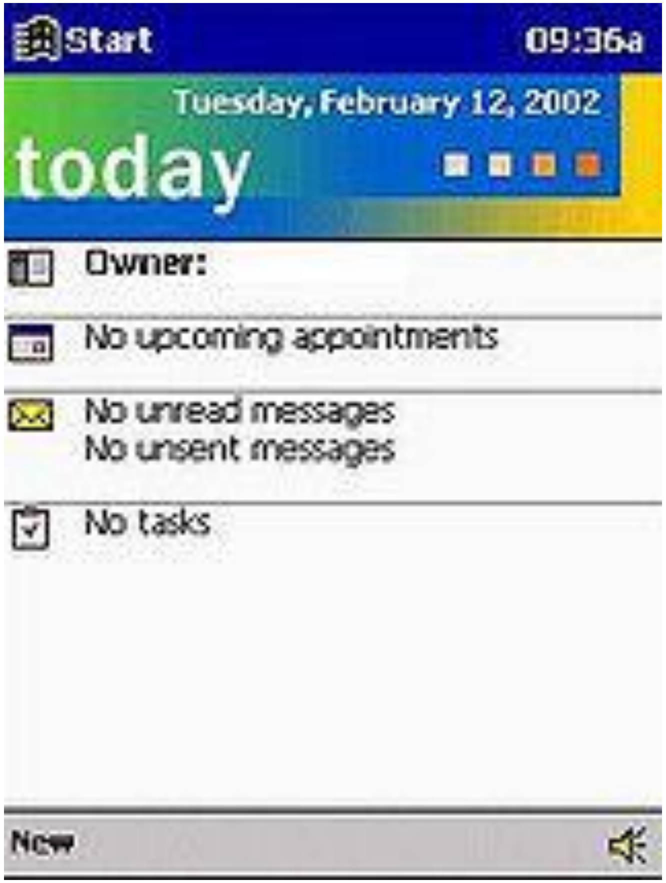
Windows Mobile

- Windows CE (Compact Edtion) - designed specifically for handheld devices, based on Win32 API
- For devices without mobile phone capabilities, and those that included mobile phone capabilities



Windows Mobile

- 1996 – Windows CE 1.0
- 1997 – Windows CE 2.0 (ATM, games consoles, Handheld PC's, kitchen utensils)
- 2000 - Windows CE 3.0 - Pocket PC 2000 - (became the OS of choice on many Pocket PCs, looked and worked like Windows 98, no phone feature)



Start 09:36a

Tuesday, February 12, 2002

today

Owner:

No upcoming appointments

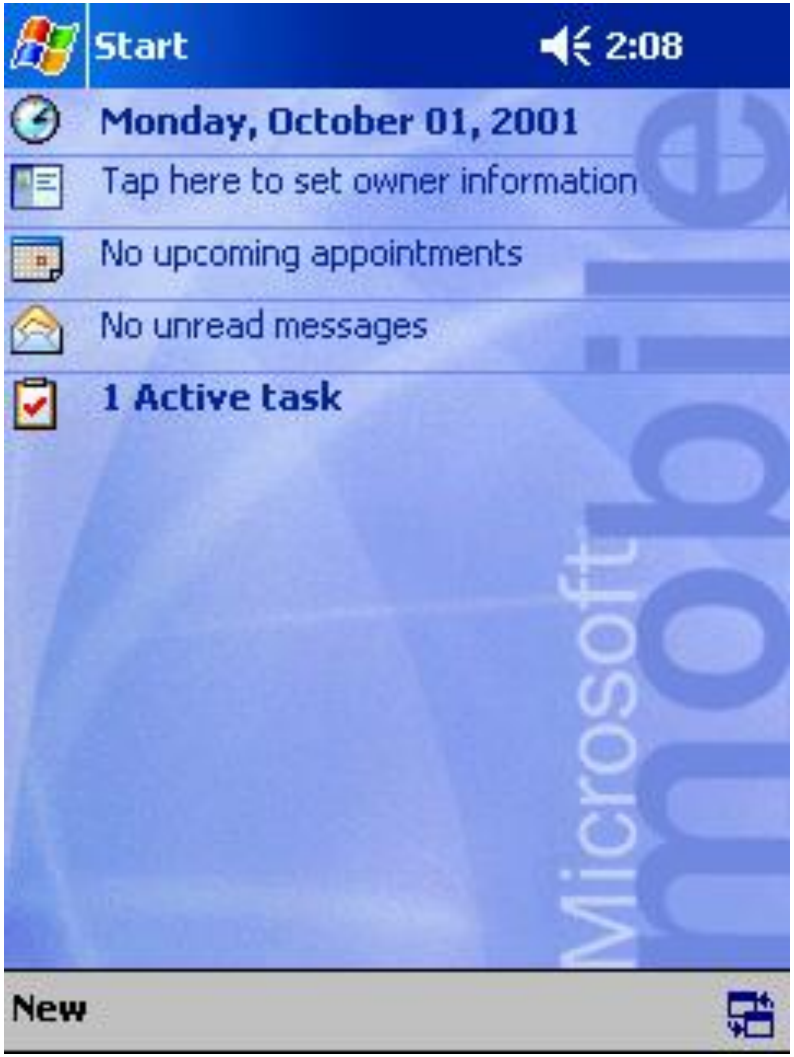
No unread messages
No unsent messages

No tasks

PocketPC 2000

Windows Mobile

- 2001 - CE 3.0 - Smartphone 2002– used for Pocket PC phones and Smartphones, UI reflect the new Windows XP



Start 2:08

Monday, October 01, 2001

Tap here to set owner information

No upcoming appointments

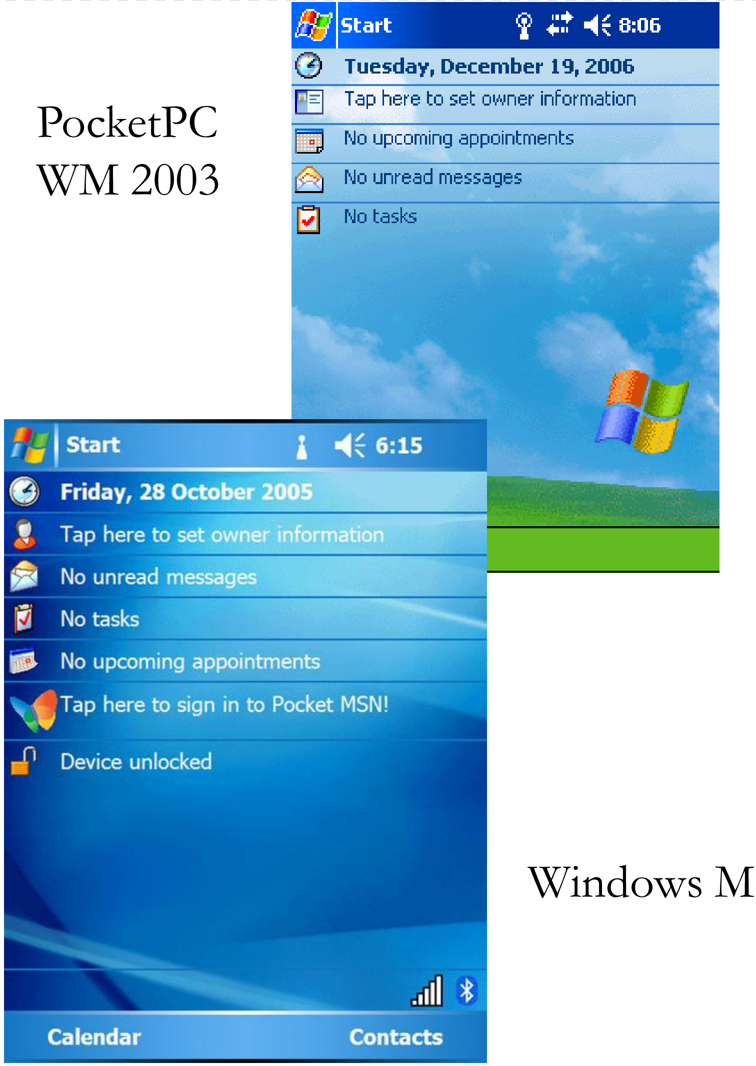
No unread messages

1 Active task

Windows Mobile

- 2003 – Windws Mobile 2003 (Windows CE 4.2) - first release under the Windows Mobile banner - name changed form PocketPC to Windows Mobile
- 2005 - WM5 (CE5.0) - new standard API created for a simplified programming of 3D apps and games with Direct3Dmobile. It use .Net Compact Framework environment

PocketPC
WM 2003



Start 8:06
Tuesday, December 19, 2006
Tap here to set owner information
No upcoming appointments
No unread messages
No tasks

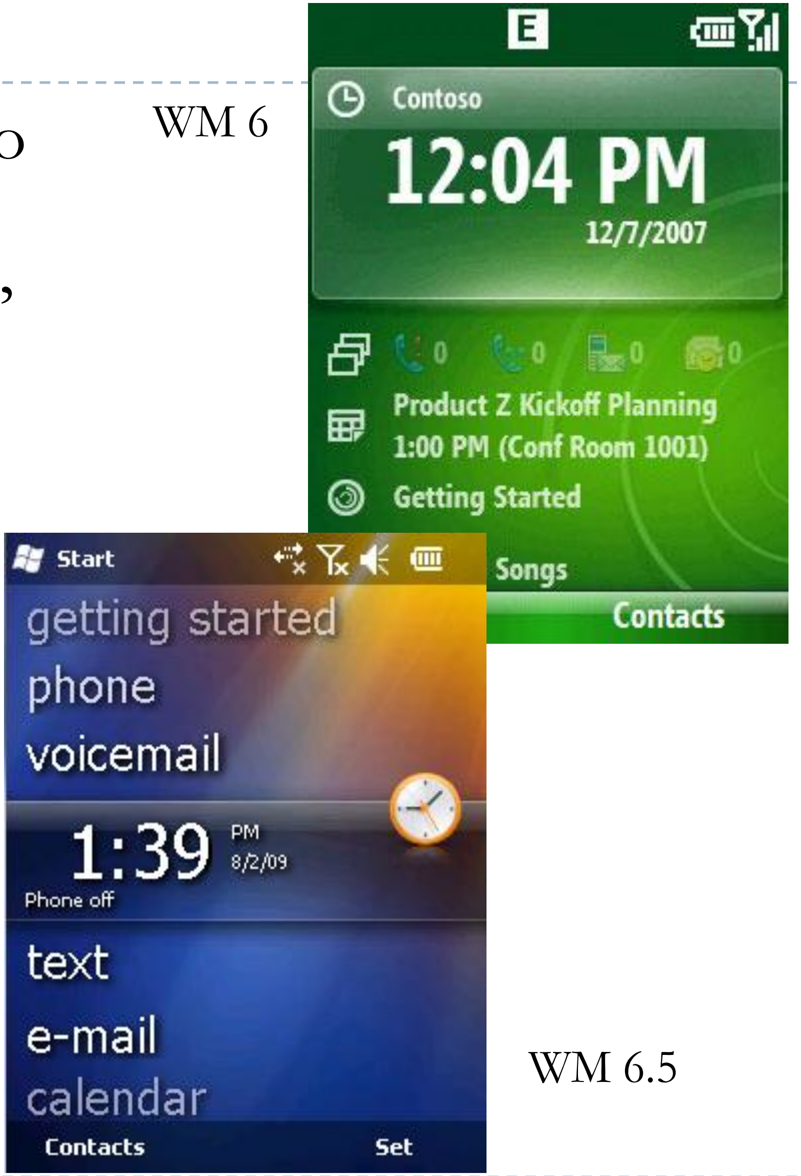
Start 6:15
Friday, 28 October 2005
Tap here to set owner information
No unread messages
No tasks
No upcoming appointments
Tap here to sign in to Pocket MSN!
Device unlocked
Calendar
Contacts

Windows Moblie 5

Windows Mobile

- 2007 – WM6 (CE 5.2) – (also year of introducing iPhone) similar in design to the Vista, works much like WM5, but with much better stability
- 2008 – WM 6.1 – (year of releasing Android)
- 2009 – WM6.5, vertically scrollable labels, Windows Marketplace announced
- Feb 2010 – WM6.5.3, was officially announced as first Windows Phone 6.5.3 smartphone

WM 6



E

Contoso

12:04 PM

12/7/2007

0 0 0 0

Product Z Kickoff Planning
1:00 PM (Conf Room 1001)

Getting Started

Songs

Contacts

Start

getting started

phone

voicemail

1:39 PM 8/2/09

Phone off

text

e-mail

calendar

Contacts Set

WM 6.5

Windows Mobile

- 2010- windows phone 7
- 10 devices operating Windows Phone 7, made by HTC, Dell, Samsung, and LG
- 2012- windows phone 8
- 2015- windows phone 10

Windows Mobile

- Important features of Windows Mobile OS
  - Graphics/Window/Event Manager component handles all input and output
  - Provides a virtual memory management
  - Security through cryptographic library.
  - Win-32 based applications
  - Not true multitasking. Application in background goes into hibernation and gets active only when it comes to foreground.

Palm OS

- Palm OS/ Garnet OS was developed by Palm Computing.
- Designed for the ease of use with the provision of a touch screen based graphical user interface.
- Deployed in smartphones, wrist watches, hand held gaming consoles, bar code readers and GPS devices.
- Nokia n810, Aceeca

Palm OS

- Features of Palm OS
  - Single tasking OS
  - Palm supplies Palm emulator (develop apps)
  - Handwriting recognition
  - Data synchronization
  - Playback and recording capabilities
  - Simple Security – lock device with password
  - Supports Interfaces - USB, Infrared, Bluetooth, Wifi
  - Proprietary format to store calendar, address task and note entries and yet are accessible by third party applications

Symbian OS

- Symbian OS was developed through a collaboration among a few prominent mobile manufactures including Ericsson, Nokia, Panasonic, Samsung, Siemens, Sony Ericsson, Psion.
- Symbian OS is a standard operating system for data-enabled mobile devices
- Symbian OS is 32 bit, little-endian operating system, running on different flavors of ARM architecture.
- It is a multitasking operating system and very less dependence on peripherals.

Symbian OS

- User libraries include networking, communication, I/O interfaces and etc.
- Access to these services and resources is coordinated through a client-server framework.
- The inherent design of Symbian OS is microkernel based.
- 2 types of Symbion OS
  - Series 60
  - UIQ Interface

Symbian OS

- Series 60
  - Large sized colour screen, easy to use interface, rich content downloading, MMS.
  - Mainly used in Nokia and Samsung
- UIQ Interface
  - User Interface Quartz technology
  - GUI with third party application developers to develop new applications.

Symbian OS

- Features of Symbian OS
  - Supports networking protocols such as TCP, UDP, PPP, DNS, FTP, WAP etc.
  - For PDA it supports Bluetooth, Infrared, USB etc.
  - Pre-emptive multitasking scheduling
  - CPU switched to low power mode when app is not responding.
  - Object oriented paradigm
  - IDE toolkit for C++ application on Symbian OS.

iOS

- Apples mobile operating system considered the foundation of the iPhone
- Originally designed for the iPhone but now supports iPod touch, iPad, and Apple TV
- iPhone OS was first unveiled in Jan 2007 at the Macworld Conference and Expo
- Released June 2007
- In June 2010 licensed the trademark iOS (From Cisco IOS)
- Now goes all the way up to iOS 9.3 Beta 6
- Originally did not allow third party applications but after Feb 2008 this changed
  - With either 30% profit to apple, or free with membership fee

iOS

- It is a closed and proprietary operating system fully owned and controlled by Apple and not designed to be used by various mobile phone vendors on their system.
- Several innovations
  - Swipe, tap, pinch, reverse pinch
  - Shake the device to undo (accelerometer sensor)
  - Rotate the device to switch from portrait to landscape

Android

- Google’s income is based on searches performed (no. of hits)
- Computer/ Laptop – preferred search engine – Google.
- Mobile phone - preferred search engine – ?
- Ex: Verizon mobile phone uses their own search engine (as Verizon don’t want to miss the extra revenue)
- Now Google took a serious decision, as mobile searches out pace the conventional searches.
- So in 2005 Google acquired a small startup ‘Android’ - which develops OS for mobile phones on Linux.

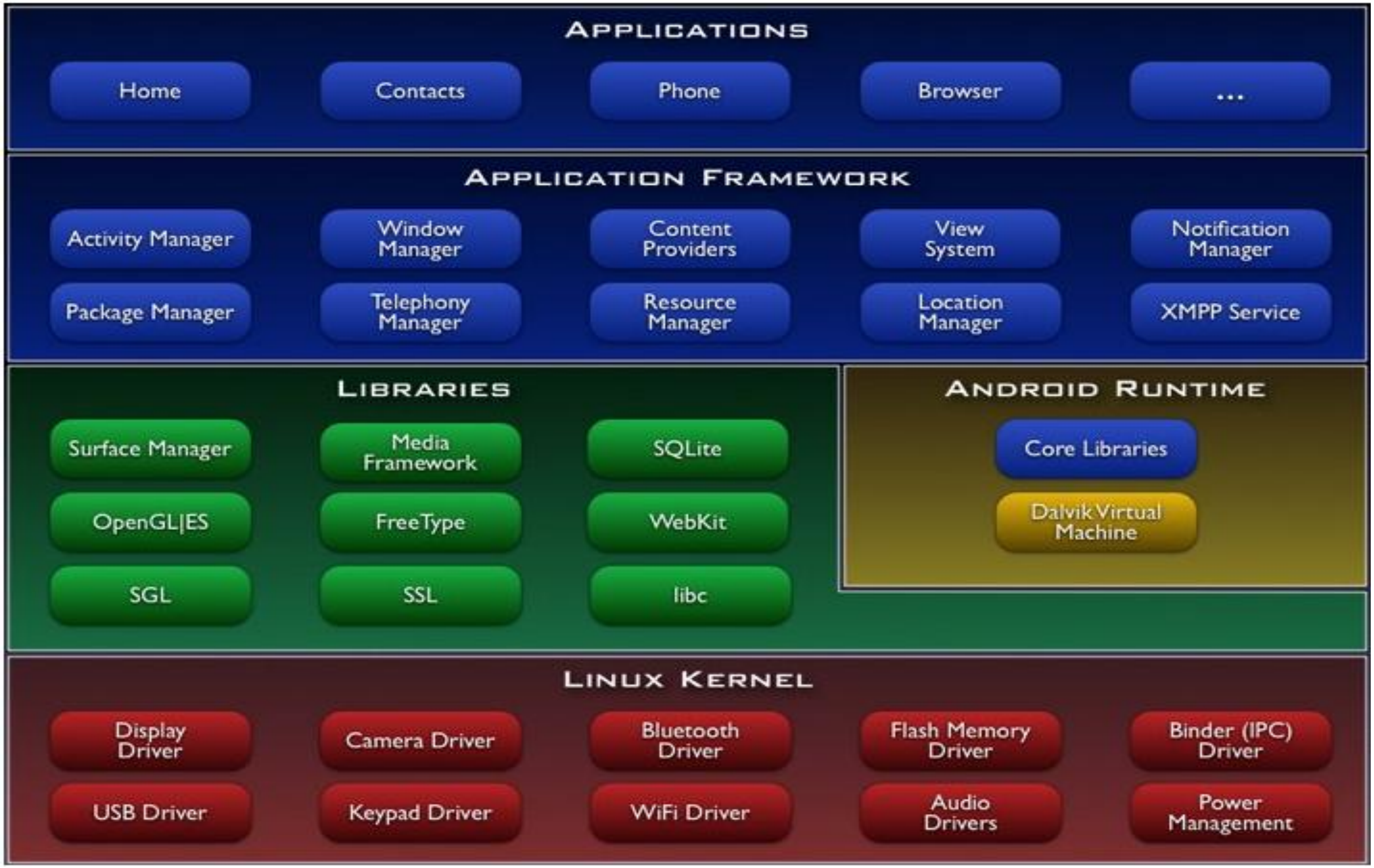
Android

- 2007 Google setup ‘Open Handset Alliance’ with 82 technology and mobile companies to develop Android OS.
- This facilitates third party developers to develop applications to android.
- Now Google could embed its search engine into Android to get extra revenue.
- Starting form 0% market in 2008 (MP with Android announced) it has shown a remarkable rate of growth in market share and user acceptance.

Android

- To understand the success of Android, it is important to understand the difficulties that users were experiencing with other OS:
  - Different user interfaces and interaction styles.
    - Ability to use either phone based keyboard or a touch screen
  - To browse real web pages and not the simplified version.
    - Ability to provide a built-in full web browser capable of rendering full web pages and not just mobile versions.
  - No third party applications
    - Provides 3rd party applications.
  - Android SDK works in eclipse
  - RDBMS SQLite
  - Preinstalled applications – Gmail, Maps, Voice search, Translate etc.

Android Software Stack



APPLICATIONS

Home
Contacts
Phone
Browser
...

APPLICATION FRAMEWORK

Activity Manager
Window Manager
Content Providers
View System
Notification Manager
Package Manager
Telephony Manager
Resource Manager
Location Manager
XMPP Service

LIBRARIES

Surface Manager
Media Framework
SQLite
OpenGL|ES
FreeType
WebKit
SGL
SSL
libc

ANDROID RUNTIME

Core Libraries
Dalvik Virtual Machine

LINUX KERNEL

Display Driver
Camera Driver
Bluetooth Driver
Flash Memory Driver
Binder (IPC) Driver
USB Driver
Keypad Driver
WiFi Driver
Audio Drivers
Power Management

Android Software Stack

- Kernel
  - It is the heart of android architecture that exists at the root of android architecture.
  - Linux kernel is responsible for device drivers, power management, memory management, device management and resource access.
- Libraries
  - On the top of linux kernel, their are Native libraries such as WebKit, OpenGL, FreeType, SQLite, Media, C runtime library (libc) etc.
  - The WebKit library is responsible for browser support, SQLite is for database, FreeType for font support, Media for playing and recording audio and video formats.

Android Software Stack

- Runtime
  - In android runtime, there are core libraries and DVM (Dalvik Virtual Machine) which is responsible to run android application.
  - DVM is like JVM but it is optimized for mobile devices.
  - It consumes less memory and provides fast performance.
- Application Framework
  - On the top of Native libraries and android runtime, there is android framework.
  - Application framework includes Android API's such as UI (User Interface), telephony, resources, locations, Content Providers (data) and package managers.
  - It provides a lot of classes and interfaces for android application development.

Android Software Stack

- Applications
  - On the top of android framework, there are applications.
  - All applications such as home, contact, settings, games, browsers are using android framework that uses android runtime and libraries.
  - Android runtime and native libraries are using linux kernal.

Blackberry OS

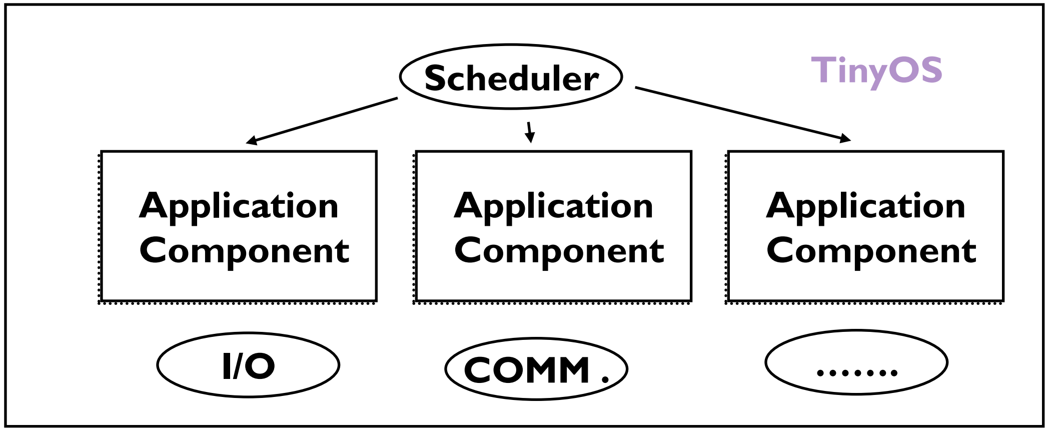
- This is a Proprietary OS designed for Blackberry smartphones.
- Features of Blackberry Os
  - Ability to send and receive internet e- mail using the “push” method of delivery
  - Phone and texting functionality
  - Supports Internet faxing and Web browsing
  - Supports the viewing of Office applications
  - Ability to support numerous other wireless information services

Comparison of Mobile OS

https://en.wikipedia.org/wiki/Comparison_of_mobile_operating_systems

OS for Sensor Networks

- NO Kernel - Direct hardware manipulation
- NO Process management - Only one process on the fly.
- NO Virtual memory - Single linear physical address space
- NO Dynamic memory allocation -Assigned at compile time
- NO Software signal or exception - Function Call instead
- Tiny Os, Contiki OS, Lite OS, MANTIS

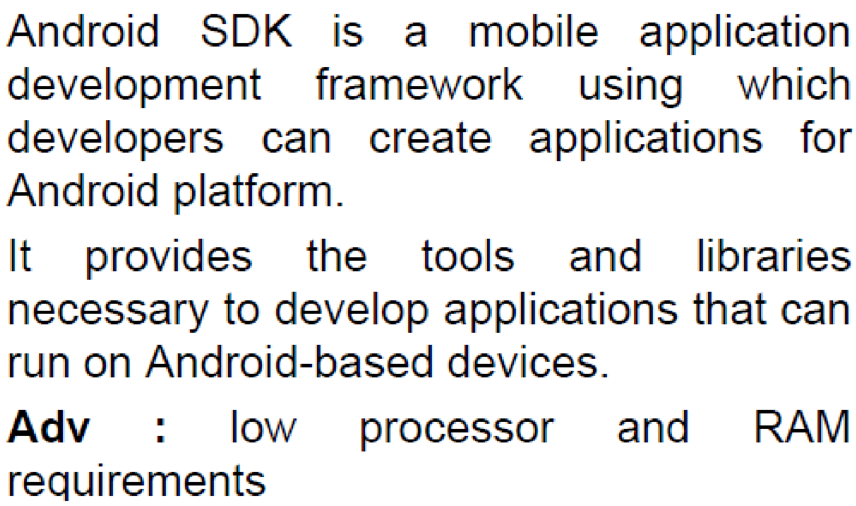


SOFTWARE DEVELOPMENT KIT: ANDROID

Android:

- Android is an Open software platform for mobile development. It has a complete stack of OS, Middleware and Applications.
- Android is an Open Handset Alliance (OHA) project, powered by Linux operating system. It is open source under the Apache 2 license.
- Android Applications are written in Java language. An easily and freely available SDK is generally used for developing applications. The development environment provides several API to help quickly develop sophisticated applications.

Android SDK:



Android SDK is a mobile application development framework using which developers can create applications for Android platform.

It provides the tools and libraries necessary to develop applications that can run on Android-based devices.

Adv : low processor and RAM requirements

Besides, Android SDK can be installed on almost all popular operating systems such as Windows, Mac OS and Linux.

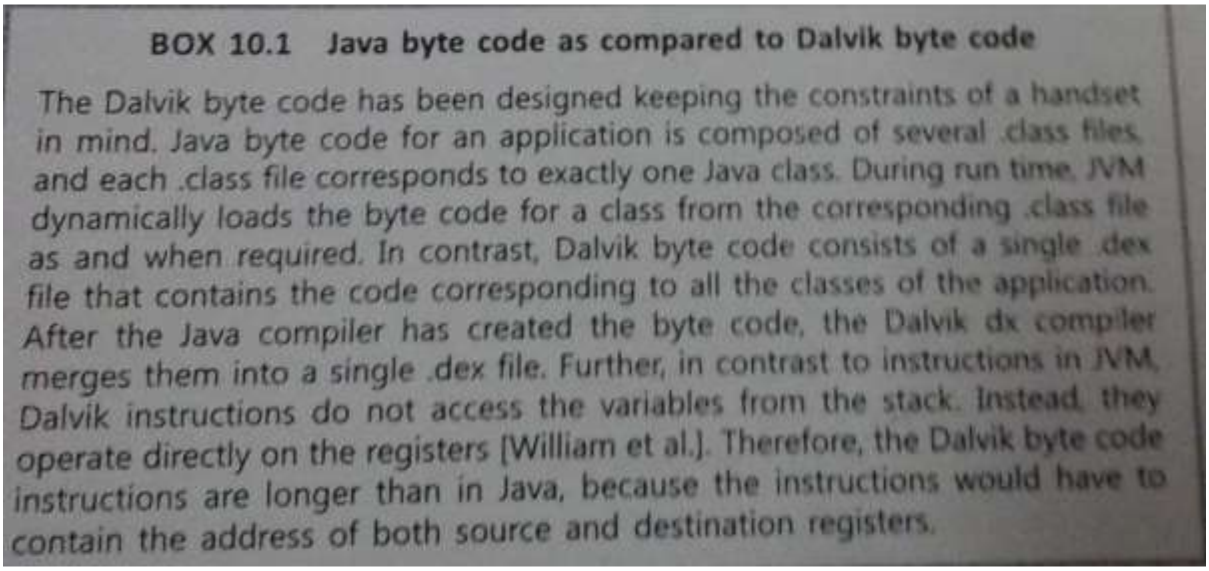
SDK comes with an IDE (Integrated Development Environment) that has a set of seamlessly integrated tools such as

- compiler
- Code editor
- Debugger
- Execution support

Dalvik VM Byte Code:

- While developing Android based applications, the developer codes the applications using Java.

- ➔ But the mobile device running Android does not finally run the Java byte Code. It runs Dalvik VM byte code.
- ➔ This is why, applications developed using J2ME cannot run directly on Android mobile phones.

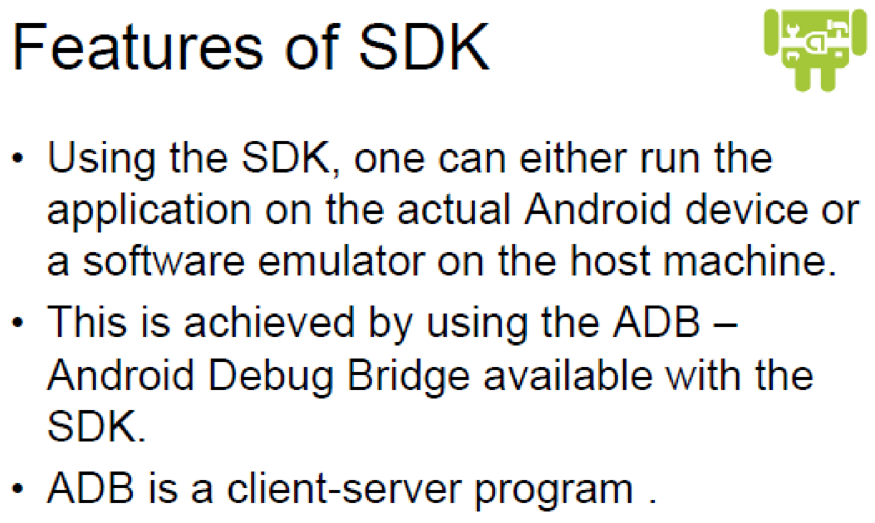


BOX 10.1 Java byte code as compared to Dalvik byte code

The Dalvik byte code has been designed keeping the constraints of a handset in mind. Java byte code for an application is composed of several .class files, and each .class file corresponds to exactly one Java class. During run time, JVM dynamically loads the byte code for a class from the corresponding .class file as and when required. In contrast, Dalvik byte code consists of a single .dex file that contains the code corresponding to all the classes of the application. After the Java compiler has created the byte code, the Dalvik dx compiler merges them into a single .dex file. Further, in contrast to instructions in JVM, Dalvik instructions do not access the variables from the stack. Instead, they operate directly on the registers [William et al.]. Therefore, the Dalvik byte code instructions are longer than in Java, because the instructions would have to contain the address of both source and destination registers.

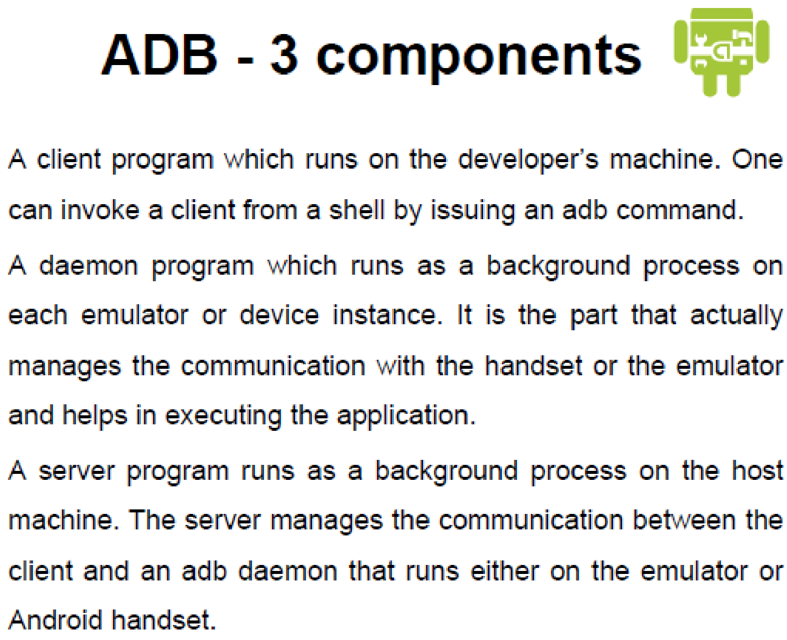
<u>Setting up the IDE:</u>

1. The environment to develop applications for Android consists of Android SDK, IDE Eclipse and Java Development Kit (JDK)
2. After installing these SDK, the path to the SDK has to be set in the path environment variable.
3. Eclipse can be used as IDE, which also automatically installs the Android SDK as a plug-in.
4. After this, new Android Project can be created.



Features of SDK

- Using the SDK, one can either run the application on the actual Android device or a software emulator on the host machine.
- This is achieved by using the ADB – Android Debug Bridge available with the SDK.
- ADB is a client-server program .



ADB - 3 components

A client program which runs on the developer’s machine. One can invoke a client from a shell by issuing an adb command.

A daemon program which runs as a background process on each emulator or device instance. It is the part that actually manages the communication with the handset or the emulator and helps in executing the application.

A server program runs as a background process on the host machine. The server manages the communication between the client and an adb daemon that runs either on the emulator or Android handset.

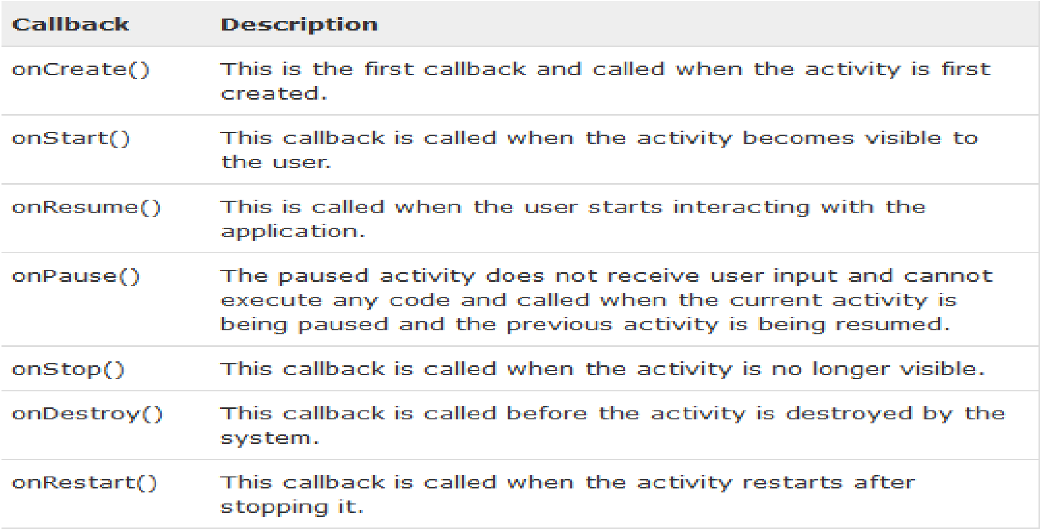
Application Building blocks of Android SDK:

Application Components are the building blocks of an Android application. They are:

- Activity
- Content Provider
- Service
- Broadcast Receiver

1) Activity:

Each activity presents a single GUI screen of an application. Almost all activities interact with the user, so the Activity class takes care of creating a window for you in which you can place your UI with **setContentView(View).**

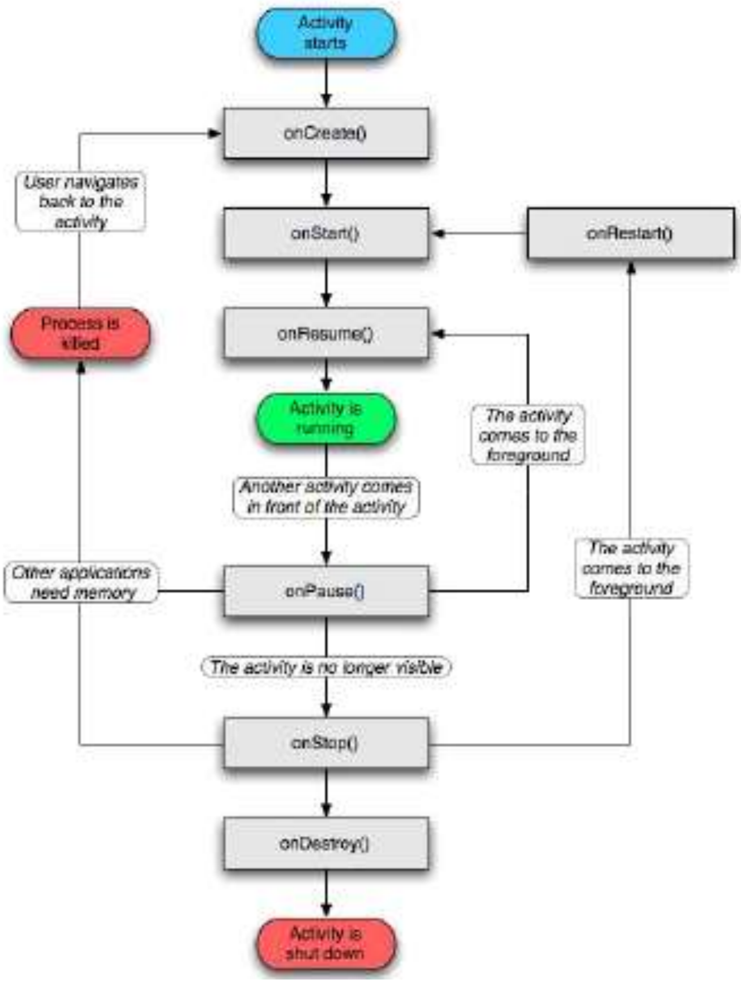


Callback	Description
onCreate()	This is the first callback and called when the activity is first created.
onStart()	This callback is called when the activity becomes visible to the user.
onResume()	This is called when the user starts interacting with the application.
onPause()	The paused activity does not receive user input and cannot execute any code and called when the current activity is being paused and the previous activity is being resumed.
onStop()	This callback is called when the activity is no longer visible.
onDestroy()	This callback is called before the activity is destroyed by the system.
onRestart()	This callback is called when the activity restarts after stopping it.

Android Activity Lifecycle

An activity has essentially four states:

- If an activity is in the foreground of the screen (at the top of the stack), it is **active** or running.
- If an activity has lost focus but is still visible, it is **paused**. A paused activity is completely alive (it maintains all state and member information and remains attached to the window manager), but can be killed by the system in extreme low memory situations.
- If an activity is completely obscured by another activity, it is **stopped**. It still retains all state and member information and it will often be killed by the system when memory is needed elsewhere.
- If an activity is paused or stopped, the system can drop the activity from memory by either asking it to **finish**, or simply killing its process.


Activity starts
onCreate()
User navigates back to the activity
onStart()
onRestart()
Process is killed
onResume()
Activity is running
The activity comes to the foreground
Another activity comes in front of the activity
The activity comes to the foreground
Other applications need memory
onPause()
The activity is no longer visible
onStop()
onDestroy()
Activity is shut down

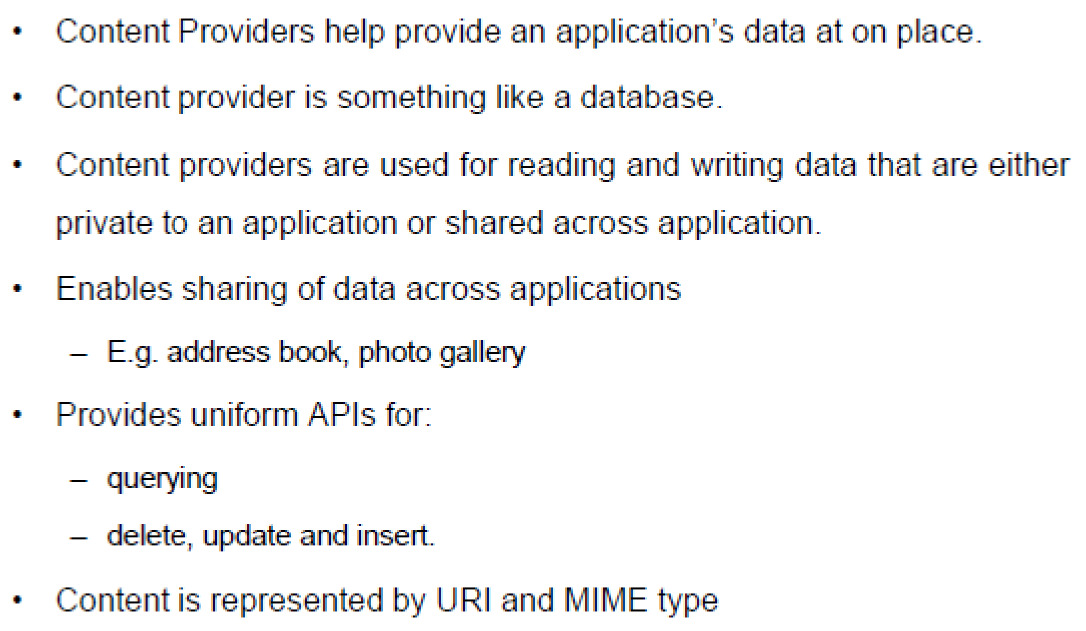
Moving from First Activity to Second Activity:

This is done using Intent class. If the source Activity is named MainActivity and destination activity is called SecondActivity, the Intent is written as:

Intent intent= new Intent(MainActivity.this, SecondActivity.class);

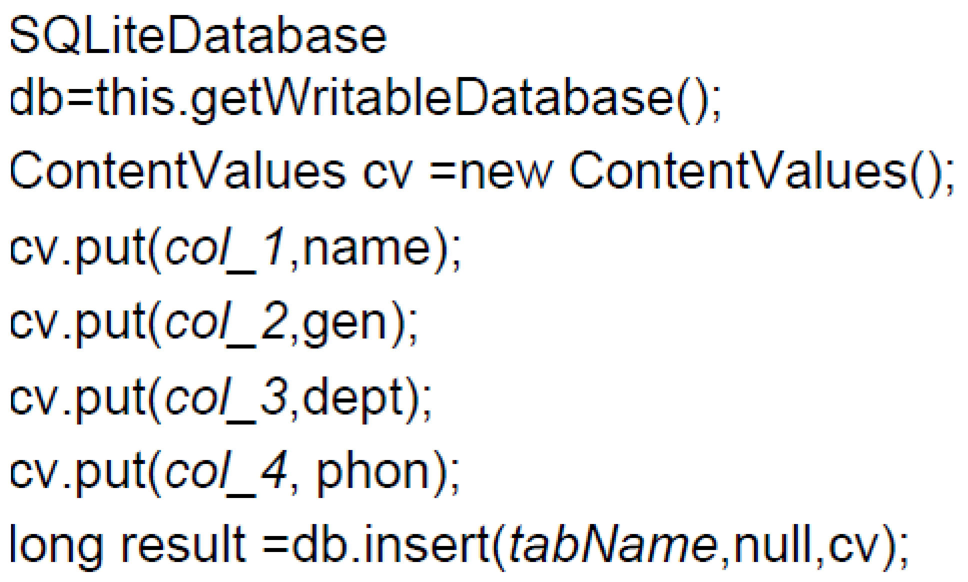
startActivity(intent);

2) Content Providers:



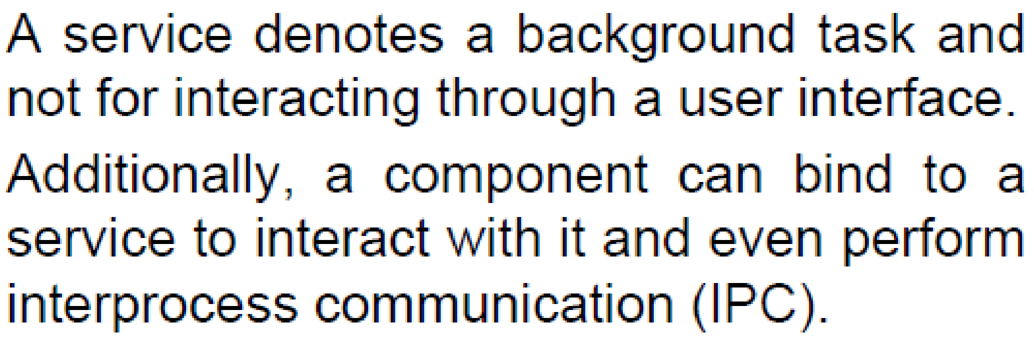
- Content Providers help provide an application’s data at on place.
- Content provider is something like a database.
- Content providers are used for reading and writing data that are either private to an application or shared across application.
- Enables sharing of data across applications
  - E.g. address book, photo gallery
- Provides uniform APIs for:
  - querying
  - delete, update and insert.
- Content is represented by URI and MIME type

Code to insert values into Database using Content Providers:



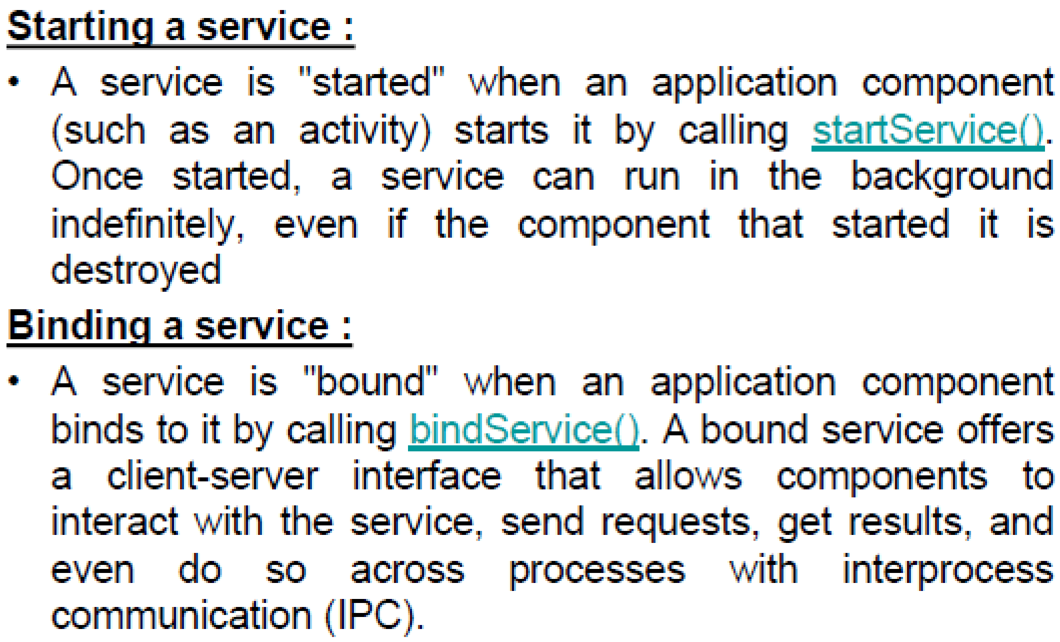
```
SQLiteDatabase
db=this.getWritableDatabase();
ContentValues cv =new ContentValues();
cv.put(col_1,name);
cv.put(col_2,gen);
cv.put(col_3,dept);
cv.put(col_4, phon);
long result =db.insert(tabName,null,cv);
```

3) **Services:**



A service denotes a background task and not for interacting through a user interface.

Additionally, a component can bind to a service to interact with it and even perform interprocess communication (IPC).



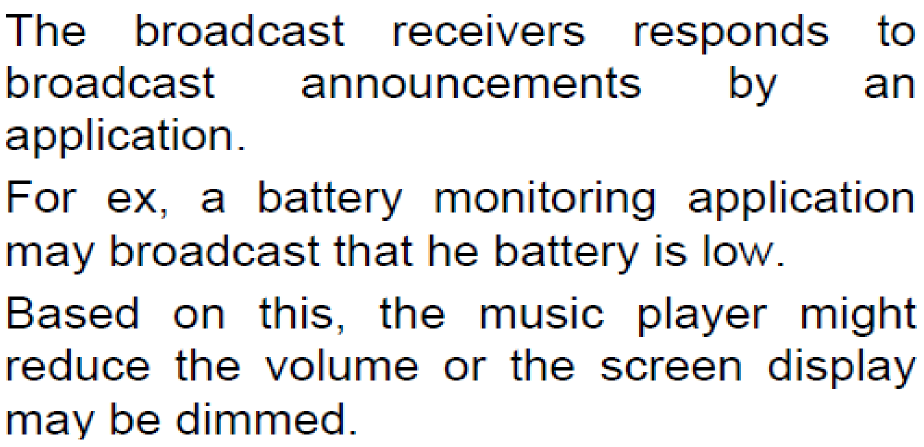
Starting a service :

- A service is "started" when an application component (such as an activity) starts it by calling startService(). Once started, a service can run in the background indefinitely, even if the component that started it is destroyed

Binding a service :

- A service is "bound" when an application component binds to it by calling bindService(). A bound service offers a client-server interface that allows components to interact with the service, send requests, get results, and even do so across processes with interprocess communication (IPC).

4) **Broadcast Receivers**



The broadcast receivers responds to broadcast announcements by an application.

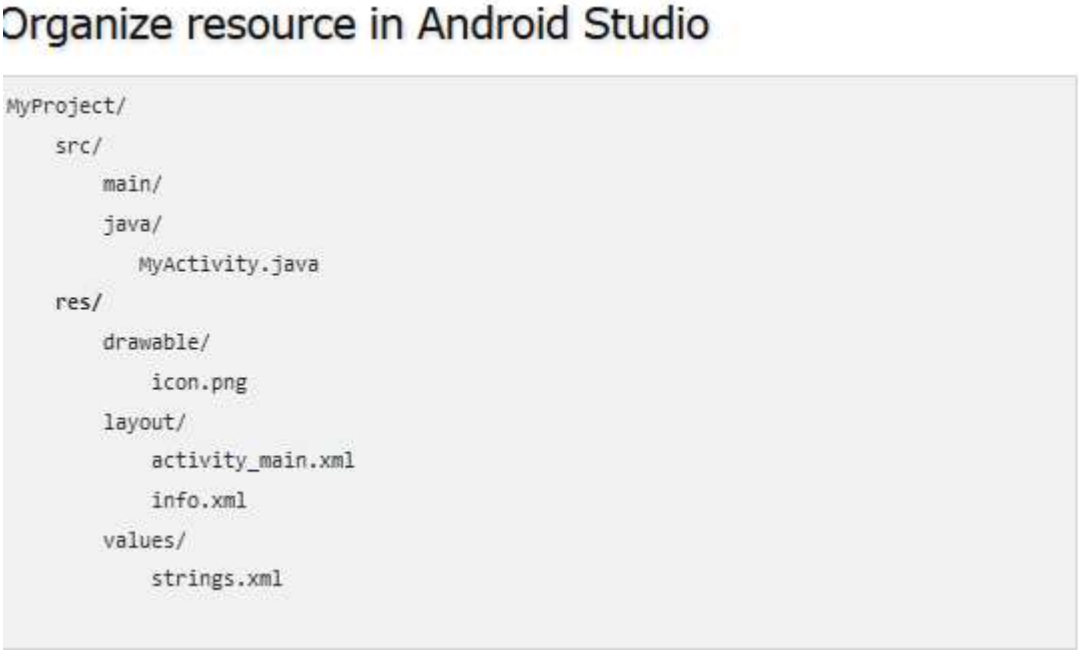
For ex, a battery monitoring application may broadcast that he battery is low.

Based on this, the music player might reduce the volume or the screen display may be dimmed.

Code to create a broadcast Receiver:

```
public class MyReceiver extends BroadcastReceiver {
   @Override
```

```
    public void onReceive(Context context, Intent intent) {
        Toast.makeText(context, "Intent Detected.",
Toast.LENGTH_LONG).show();
    }
}
```



Organize resource in Android Studio

```
MyProject/
    src/
        main/
        java/
           MyActivity.java
    res/
        drawable/
            icon.png
        layout/
            activity_main.xml
            info.xml
        values/
            strings.xml
```

SOFTWARE DEVELOPMENT KIT: WINDOWS

Definition of SDK: (write under any SDK asked)

SDK is a set of software development tools that allows the creation of applications for a certain software package, software framework, device drivers, hardware platform, computer system, video game console, enterprise, operating system, mobility, analytics or similar development platform.

Windows SDK:

Microsoft Windows SDK (and its predecessors Platform SDK, and.NET Framework SDK) are software development kits(SDKs) from Microsoft that contain documentations, header files, libraries, samples and tools required to develop applications for Microsoft Windowsand.NET Framework.

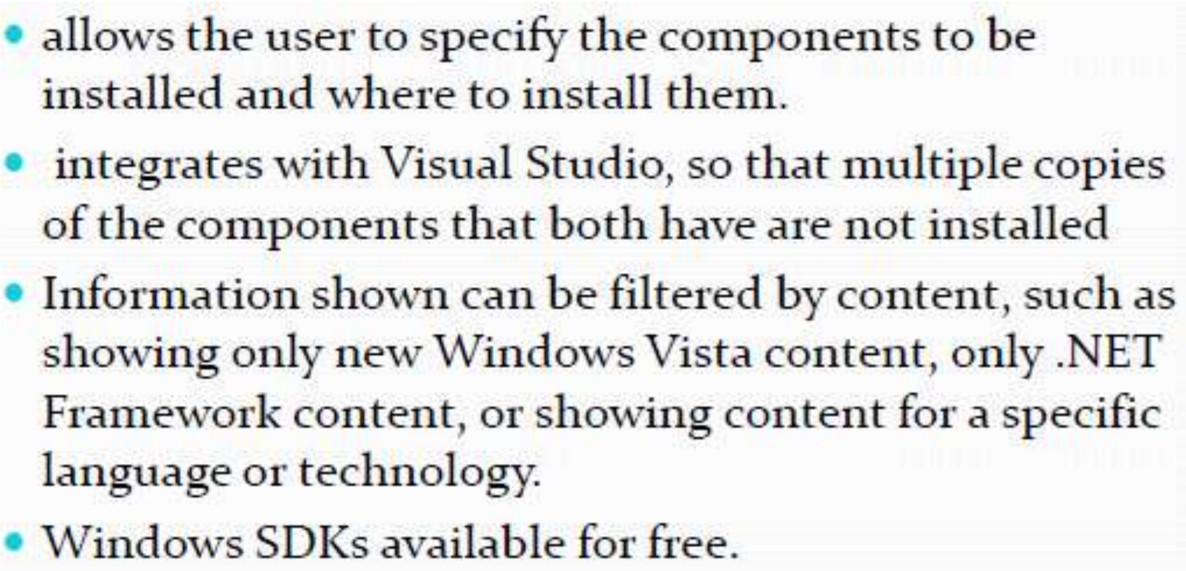
Installing Development environment for Windows SDK:

The Windows SDK no longer ships with a complete command-line build environment.

Case 1: You must install a compiler and build environment separately.

Case 2: If you require a complete development environment that includes compilers and a build environment, you can download Visual Studio 2012 Express, which includes the appropriate components of the Windows SDK.

Features of Windows SDK:



- allows the user to specify the components to be installed and where to install them.
- integrates with Visual Studio, so that multiple copies of the components that both have are not installed
- Information shown can be filtered by content, such as showing only new Windows Vista content, only .NET Framework content, or showing content for a specific language or technology.
- Windows SDKs available for free.

Development Resources of Windows SDK:

- By using the Windows Software Development Kit for Windows 8 (Windows SDK), you can develop both Windows Store apps and Windows desktop apps.
- The Windows SDK supports the following models for developing Windows Store apps:
    - Web (HTML5, CSS3, and JavaScript)
    - Managed XAML, with code-behind in C# or Visual Basic
    - Native XAML, with code-behind in C++
    - Native (Win32/COM)

Windows Library for JavaScript:

- The Windows SDK includes the Windows Library for JavaScript, which provides a **set of new controls** that are designed for developing Windows Store apps by using JavaScript.

- These controls include **WinJS.UI.DatePicker, WinJS.UI.FlipView, WinjS.UI.ListView, and WinJS.UI.Rating.**
- The Windows SDK also includes style sheets to give your app the look and feel of Windows 8.

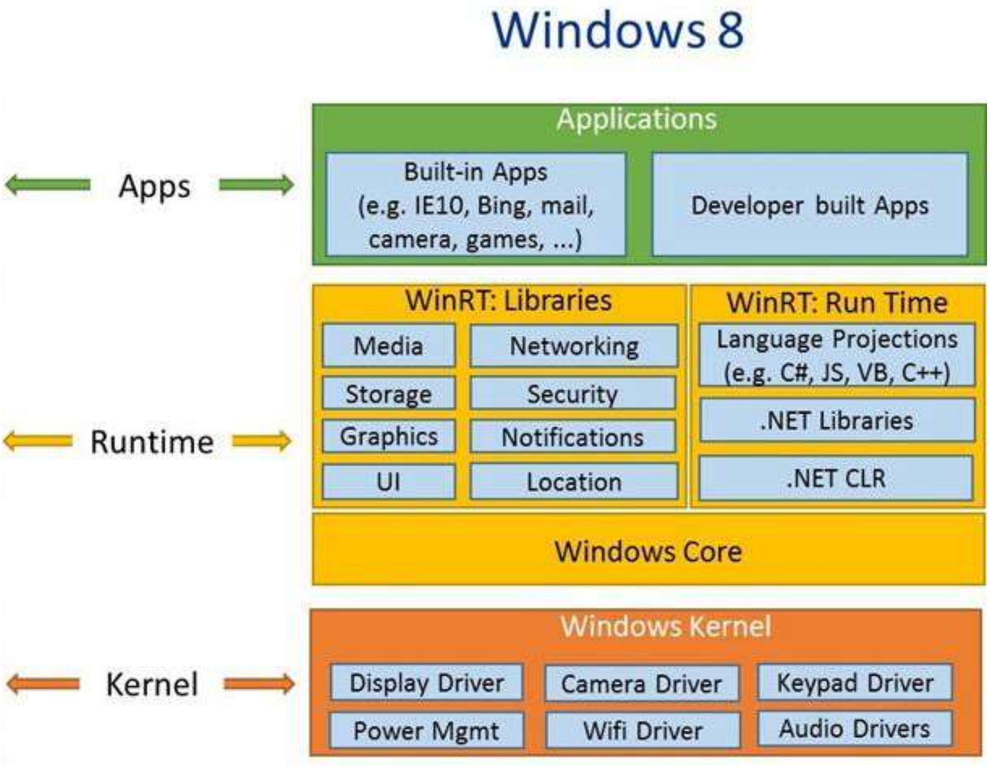
Windows Store Applications vs. Desktop Applications:

Windows Store applications run inside a security sand box which tightly controls access to system resources.

Traditional desktop applications run with privileges implicitly inherited from the user principal that started the application.

Architecture and working of Windows 8 SDK:

Windows 8 supports a driver architecture that allows user mode as well as kernel mode drivers.



1. User mode drivers (e.g. camera, printer) run in a low privileged environment (in user mode) which prevents system crashes resulting from driver quality issues.
2. **WinRT** is a sand box for running Windows Store applications.
3. Each process in Windows Store application creates an instance of .NET CLR (Common Language Runtime) for running application code written in .NET languages like C# and VB.net.

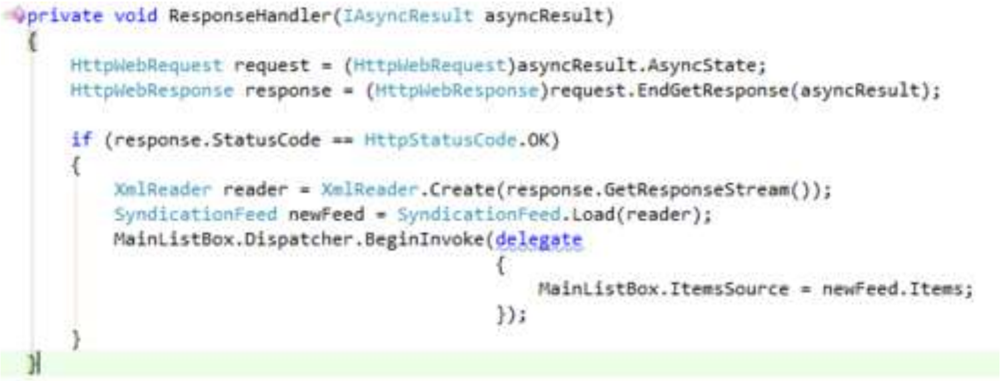
4. .NET code gets deployed as **MSIL (Microsoft Intermediate Language) byte code** that gets compiled into X86 or ARM instruction set at run time.
5. **WinRT API** is a collection of native components implemented in COM (Component Object Model).
6. COM is an RPC based technology that creates and manages object life cycle through class factories and reference counting.

Code written in Windows SDK for:

1. **Downloading a HTML Document:**

```
HtmlDocument doc = new HtmlDocument();
doc.Load("file.htm");
foreach(HtmlNode link in doc.DocumentElement.SelectNodes("//a[@href"])
{
   HtmlAttribute att = link["href"];
   att.Value = FixLink(att);
}
doc.Save("file.htm");
```

2. **WCF RSS:** Parses (or generates) RSS and ATOM into strongly-typed classes and support LINQ querying.



```
private void ResponseHandler(IAsyncResult asyncResult)
{
    HttpWebRequest request = (HttpWebRequest)asyncResult.AsyncState;
    HttpWebResponse response = (HttpWebResponse)request.EndGetResponse(asyncResult);

    if (response.StatusCode == HttpStatusCode.OK)
    {
        XmlReader reader = XmlReader.Create(response.GetResponseStream());
        SyndicationFeed newFeed = SyndicationFeed.Load(reader);
        MainListBox.Dispatcher.BeginInvoke(delegate
                                           {
                                               MainListBox.ItemsSource = newFeed.Items;
                                           });
    }
}
```

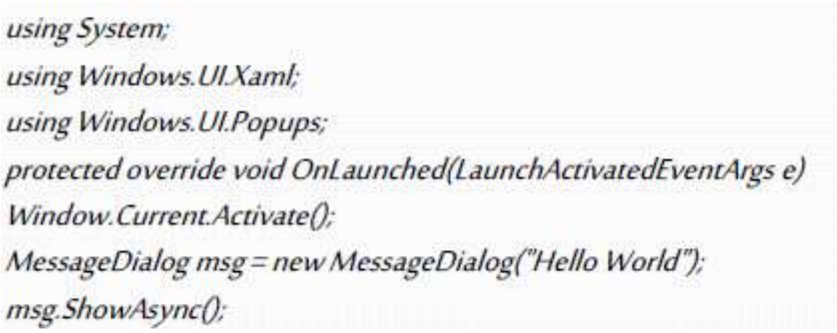
3. Loading images using ImageTools:

```
ImageTools.IO.Decoders.AddDecoder<BmpDecoder>();
ImageTools.IO.Decoders.AddDecoder<PngDecoder>();
ImageTools.IO.Decoders.AddDecoder<JpegDecoder>();
ImageTools.IO.Decoders.AddDecoder<GifDecoder>();
```

Steps in running the application:

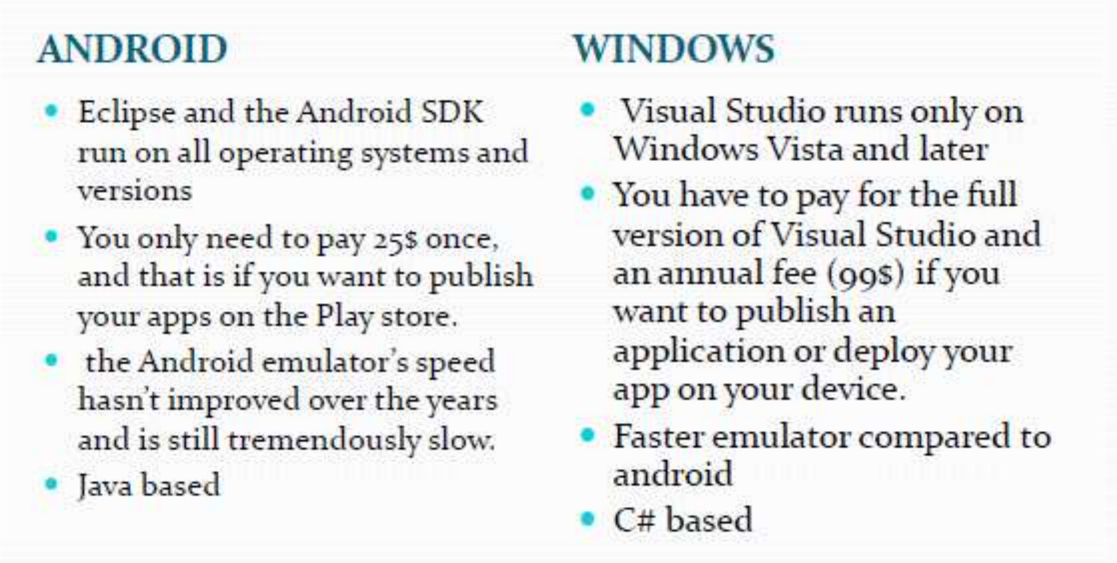
1) Create a universal app project
2) Add C# code to a file in the Solution Explorer: The **solution** is a container for all the code and files in our app.
3) Set the startup project
4) Run the app.

Sample Hello World Code:



```
using System;
using Windows.UI.Xaml;
using Windows.UI.Popups;
protected override void OnLaunched(LaunchActivatedEventArgs e)
Window.Current.Activate();
MessageDialog msg = new MessageDialog("Hello World");
msg.ShowAsync();
```

Difference between Windows SDK and Android SDK:



ANDROID

- Eclipse and the Android SDK run on all operating systems and versions
- You only need to pay 25$ once, and that is if you want to publish your apps on the Play store.
- the Android emulator's speed hasn't improved over the years and is still tremendously slow.
- Java based

WINDOWS

- Visual Studio runs only on Windows Vista and later
- You have to pay for the full version of Visual Studio and an annual fee (99$) if you want to publish an application or deploy your app on your device.
- Faster emulator compared to android
- C# based

SOFTWARE DEVELOPMENT KIT: iOS

iOS SDK:

- The iOS SDK (Software Development Kit) (formerly iPhone SDK) is a software development kit developed by Apple Inc.,
- All applications are developed in Xcode.
- The current version being iOS 9.3 , Xcode version 7.3
- The iOS SDK contains the tools and resources to develop **native** iOS applications, which mean that the SDK allows developers to develop, install, run, and test applications in the iOS Simulator and on physical devices.
- The two driving forces powering native iOS applications are **Objective-C and the native iOS system frameworks.**

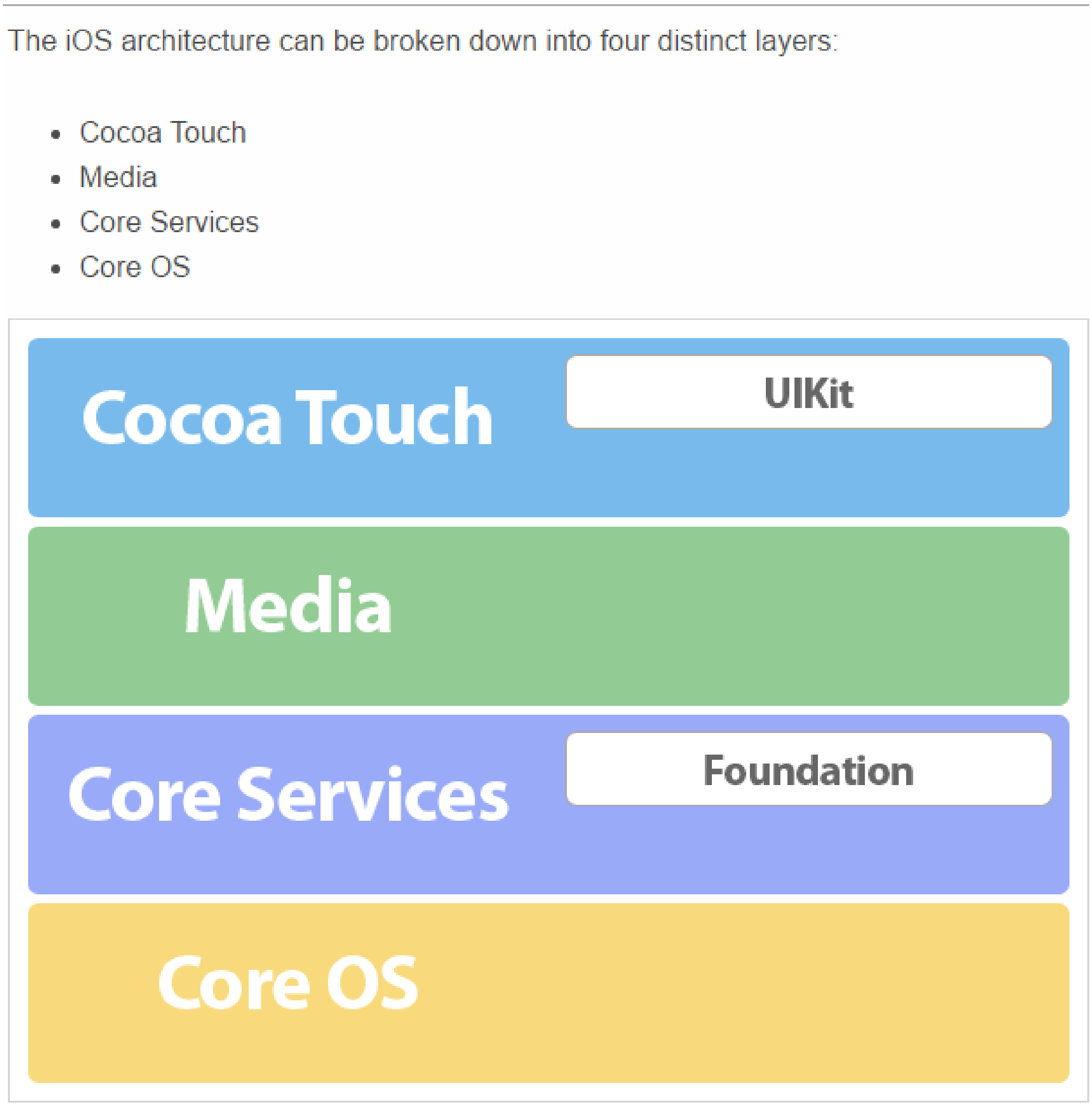
What is a Cocoa application?

Apple defines a Cocoa application as an application that's composed of objects that ultimately inherit from **NSObject,** a root class declared in the Foundation framework, and that's based upon the Objective-C runtime.

Framework:

- → A framework is a directory that contains a dynamic shared library and the resources associated with it, such as header files, images, etc.
- → Frameworks are the access points to various system interfaces, such as the iOS address book, the device's camera roll, and the music library

iOS Architecture:



The iOS architecture can be broken down into four distinct layers:

- Cocoa Touch
- Media
- Core Services
- Core OS

Fig: Layered architecture of iOS

1) **Cocoa Touch layer:**

- The Cocoa Touch layer is the topmost layer of the iOS architecture.
- It contains some of the key frameworks which native iOS applications rely upon, with the most prominent being the UIKit framework.
- The Cocoa Touch layer defines the basic application infrastructure and provides a number of vital technologies, such as multitasking and touch-based input.
- iOS applications rely heavily on the UIKit framework.
- In addition to UIKit, it contains the Map Kit, Event Kit, and Message UI frameworks, among others.

1b) UIKit framework:

- The UIKit framework or UIKit is tailored to the iOS platform.
- UIKit provides the infrastructure for graphical, event driven iOS applications. It also takes care of other core aspects that are specific to the iOS platform like multitasking, push notifications, and accessibility.

2) **Media layer:**

- Graphics, audio, and video are handled by the Media layer.
- This layer contains a number of key technologies, such as Core Graphics, OpenGL ES and OpenAL, AV Foundation, and Core Media.
- The Media layer contains a large number of frameworks including the **Assets Library framework** to access a device's photos and videos, the **Core Image framework** for image manipulation through filters, and the **Core Graphics framework** for 2D drawing.

3) **Core Services Layer**

- The Core Services layer is in charge of managing the fundamental system services that native iOS applications use.
- The Cocoa Touch layer relies heavily on the Core Services layer for some of its functionality.
- The Core Services layer also provides a number of indispensable features, such as block objects, Grand Central Dispatch, In-App Purchase, and iCloud Storage.
- One of the most welcomed additions to the Core Services layer is ARC or **A**utomatic **R**eference **C**ounting.
- ARC simplifies the process of memory management in Swift (Objective C).
- It sends two messages retain and release messages to prevent and allow deallocation of memory.

3b) Foundation framework:

- ❒ The Foundation framework or Foundation is another essential framework for iOS and OS X applications.
- ❒ The Foundation framework is more than a collection of useful classes, such as NSArray, NSDictionary, and NSDate.
- ❒ Foundation makes it possible for the different libraries and frameworks to easily share data and code.

3c) Core Foundation Framework:

Core Foundation has a feature referred to as **toll-free bridging**, which enables the substitution of Cocoa objects for Core Foundation objects in function parameters and vice versa.

4) Core OS layer:

- ► Most of the functionality provided by the three higher level layers is built upon the Core OS layer.
- ► The Core OS layer provides a handful of frameworks that your application can use directly, such as the Accelerate and the Security frameworks.
- ► The Core OS layer also encapsulates the kernel environment and low level UNIX interfaces.
- ► Through the **libSystem** library, which is C-based, many low level features can be accessed directly, such as BSD sockets, POSIX threads, and DNS services.

Objective-C:

- ❒ Objective-C is a strict superset of C and, in contrast to C, it is a high level programming language.
- ❒ The key difference between C and Objective-C is that the latter is an object oriented programming language, whereas C is a procedural programming language.
- ❒ Getter and Setter methods are essential features of Objective C in iOS SDK .

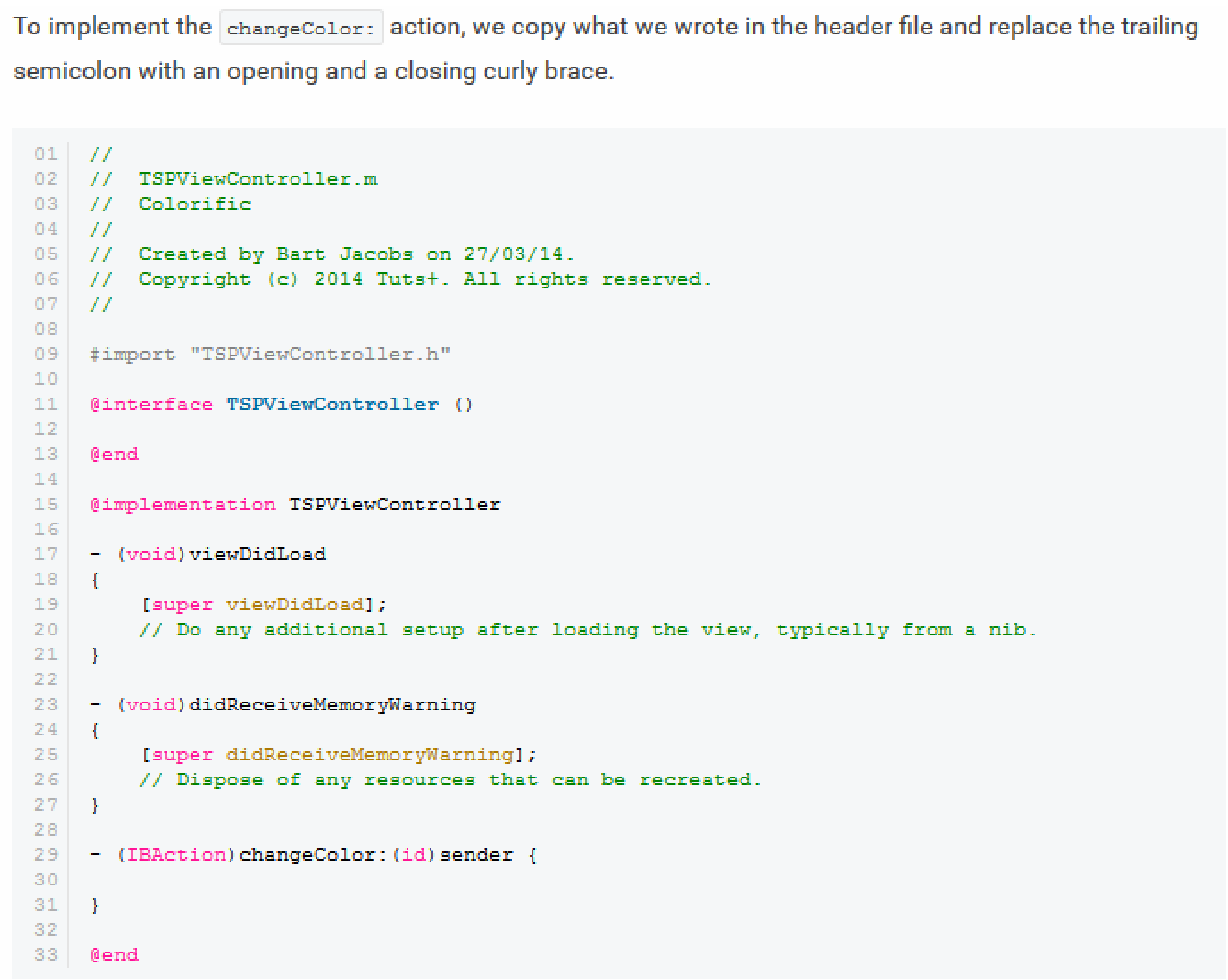
Setting up Development environment:

Before we can start creating iOS applications, we need to set up the development environment. Xcode is an outstanding Integrated Development Environment (IDE) developed by Apple.

1. Download and Install Xcode from iOS Dev Center or Mac App Store.
2. To create the first iOS application, create an **Xcode project- > Single Window Application**. An Xcode project is a folder or package that contains all the necessary files and components to manage and build your application.

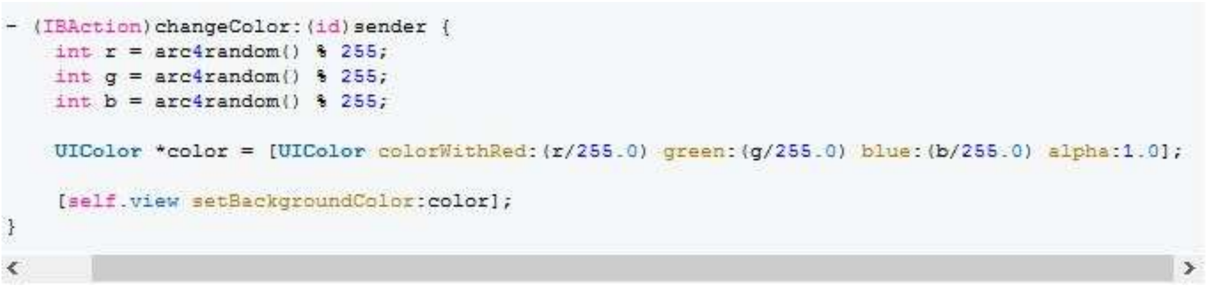
3. Configure the Xcode project by filling in various details about the project.
4. Build and run the Xcode project on the simulator.

Sample code to change background color of the application:



To implement the `changeColor:` action, we copy what we wrote in the header file and replace the trailing semicolon with an opening and a closing curly brace.

```
//
//  TSPViewController.m
//  Colorific
//
//  Created by Bart Jacobs on 27/03/14.
//  Copyright (c) 2014 Tuts+. All rights reserved.
//

#import "TSPViewController.h"

@interface TSPViewController ()

@end

@implementation TSPViewController

- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.
}

- (void)didReceiveMemoryWarning
{
    [super didReceiveMemoryWarning];
    // Dispose of any resources that can be recreated.
}

- (IBAction)changeColor:(id)sender {

}

@end
```



```
- (IBAction)changeColor:(id)sender {
    int r = arc4random() % 255;
    int g = arc4random() % 255;
    int b = arc4random() % 255;

    UIColor *color = [UIColor colorWithRed:(r/255.0) green:(g/255.0) blue:(b/255.0) alpha:1.0];

    [self.view setBackgroundColor:color];
}
```

APPLICATIONS OF M-COMMERCE

M-COMMERCE

- ***B2C***

- ***B2B***

B2C

- Business-to-consumer(B2C) is a form of commerce in which products or services are sold by a business firm to consumer directly

ADVERTISING

- Good targeted advertising can be done by using demographic information collected by network service providers and based on current location of a user.

- Network service provider may also keep track of the history of the purchases made by customers by directing advertisements to mobile phones.

- Customers may also solicit specific advertisements.

COMPARISON SHOPPING

- Consumers can use their mobile phones to get a comparative pricing analysis of a product at different stores and also the prices of the related products.

- By scanning the *bar code* on a product , the customers can see the prices of this products at different shops in the adjacent area.

- Similarly consumers can also access product reviews from consumer organizations or customers

INFORMATION ABOUT A PRODUCT

- Consumers can access additional information about products through their mobile phones.
- EXAMPLE : Consumer buys some medicine in a pharmacy shop, but the dosage instruction on the carton is given in German and Spanish languages.
- But customer can scan bar code on the pack using the mobile device and after requests information find that he is able to read the dosage instructions in the English language.

MOBILE TICKETING

- Mobile phones can be used to purchase movie tickets and train tickets using credit cards.

- After payment is received , a unique bar code is sent to the purchaser’s mobile phone by an SMS.

LOYALTY AND PAYMENT SERVICES

- Mobile phones can replace the physical loyalty cards.
- When consumers sign up for a supermarket loyalty scheme, a unique bar code is sent to a consumer’s mobile phone.
- After shopping the customers accumulate points based on total amount spent.
- Mobile phones can also be used to make payment.
- Payment is made through the person’s mobile payment balance.

INTERACTIVE ADVERTISEMENTS

- Customers can scan a bar code in an advertisement for a product appearing on a TV screen using their mobile phones.

- By scanning the bar code , the consumer can order the product by invoking an internet application.

CATALOGUE SHOPPING

- Mobile phones can be used to place orders for products listed in a catalogue
- EXAMPLE: a consumer might receive a catalogue by SMS from a catalogue shopping company.
- The SMS could briefly describe the product on sale and would be accompanied by a unique bar code .
- By scanning the bar codes, the consumer can buy products directly from the catalogue shopping company.

B2B

- Business-to-business(B2B) is a form of commerce in which products or services are sold by company to its dealers

ORDERING AND DELIVERING CONFIRMATION

- Mobile phones can be used by dealers to order products.

- The orders can be sent to the supplier in standard format. By scanning the bar code on a product by using the camera of a mobile phone and specifying the quantity required through a simple application, a dealer can automatically re-order goods.

- Mobile phones can be used to gather information about the status of consignment during the transport and delivery process.

STOCK TRACKING AND CONTROL

- Mobile phones can be used to keep track of the stock in a distributed inventory system and send updates to a central database.

- By using a mobile phone to scan bar codes or RFID tags on products in the inventory, employees can update the stock in real time and that too effortlessly.

- EXAMPLE: Stock control of apparel items warehoused in the various department stores.

SUPPLY CHAIN MANAGEMENT

- Information about the supply chain processes can be made available via mobile devices.
- By scanning an RFID tag using a mobile phone, it is possible for a manager or anyone in the supply chain to check information about a product’s state in the supply chain.
- This kind of accurate information can help manage the business efficiently.

MOBILE INVENTORY MANAGEMENT

- Whenever a store needs certain goods, it locates the nearest truck to take delivery of the required goods.
- This reduces amount of inventory and cost for both the producers and the retailers .
- It also has the potential to drastically reduce the delivery times and help in just-in-time delivery goods.

Uses of Mobile Commerce in India

The areas of operation largely affected by m-commerce technologies include:

M-pesa

Vodafone has teamed up with ICICI Bank for M-pesa. The users can send money through mobile phones to anybody. The system is fast and secure. The user has the option to send money either to a mobile number or to a bank account.

Financial Services

Those include mobile banking whereby customers may use their mobile devices, such as, smart phone to access their accounts and transfer of fund to other account or to pay their bills. Funds can be transferred through Mobile Phones without the need to go to ATM.

Mobile Recharges

Mobile Recharge is the widely used M-Commerce transaction in India.

Tele Communications

Those include service bill payment, and account reviews can all be conducted from the same hand-held device.

Online Purchases

The consumers are given the ability to place orders and make the payment from mobile devices.

Information on Mobile Web

Those include the delivery of financial news, movie downloads, entertainment, m-training etc. through a single mobile service.

Payment of Utility Bills

M-banking is also used to pay insurance premium payments, electricity and other bills, internet bills etc.

Entertainment

Many restaurants and also allow to reserve seat from Mobile Devices.

List of M-Commerce Companies in India

Online Stores

- m.ebay.in
- m.flipkart.com
- m.jabong.com
- m.yepme.com

Mobile Banking

- m.hdfcbank.com
- m.icicibank.com
- m.onlinesbi.com
- m.axisbank.com

Online Travel

- m.makemytrip.com

STRUCTURE OF MOBILE COMMERCE

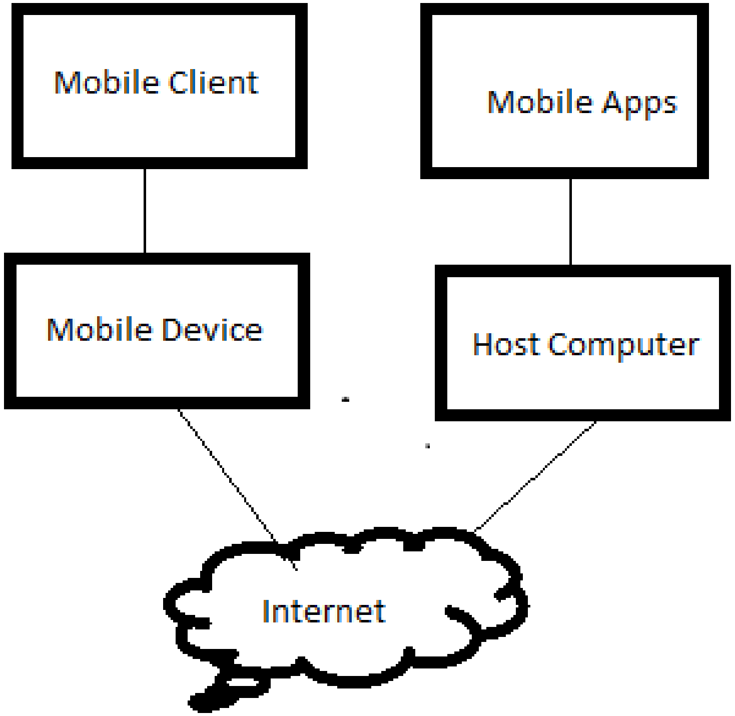
- In mobile commerce, a content provider implements an application by providing two sets of program

  a) Client side program

  b) Server side program

- Client side programs– runs on the microbrowsers installed in the user’s mobile
- Server side programs – performs database access , computations and it reside on the host computer.

Architecture of mobile commerce FRAMEWORK



1. **Mobile Devices:**

   Handheld devices- provides user interface part of mobile commerce.

   Users specify the requests through appropriate interfaces, which are then transmitted through the internet.

   Results obtained from application are displayed in a suitable format.

2. **Mobile Middleware:**

   purpose- seamlessly and transparently

Map the internet content to mobile phones that may have a variety of OSs, markup lang, protocols.

Also handle encrypting and decrypting communication – to provide secure transactions.

3. **Network:**

core of every form of mobile commerce.

- User requests are delivered either to the closest wireless access points or to the base station.
- Servers are connected to the internet(wired network).
- So, user requests are routed to these servers using the standard transport and secured mechanism provided by the network.

- **4.Host Computers:**

powerful servers – process and store all the info needed for the apps.

3 major components:

1.web server-help connect with the client

2. database server – store data

3. application program – implements the logic of the application.

- **Pros and Cons of M-Commerce:**

Advantages:

- Business organization – provide customer convenience, cost savings, new business opportunities.
- For customers- provides the flexibility of anywhere and anytime.
- Mobile devices – highly personalizable
- Can store many customer specific info.

- **Disadvantages:**

* Mobile devices – do not offer graphics or processing power of pc.

- Small screen of devices – limit the complexity of applications.
- Underlying network – impose several type of restrictions . So, ubiquity of E-commerce in true sense can be hard to achieve.
- The biggest disadvantage – security. If a customer is not careful , he may get billed for items he did not purchase.