

# ANDROID AND ANDROID APPLICATIONS



UDAY LINGALA  
CSCI 5448, Fall 2012

# Content

- Introduction to Android system
  - What is android?
  - History
  - Android Market
- Why Android
- Design philosophy
- System Architecture
- Features
- Android applications
- Frame Works
- Using Existing Application in our own Application
- Data Storage System in Android
- Best Application implementation mechanisms

# Introduction

What is Android?

- Android is a software stack for mobile devices that includes an operating system, middleware and key applications.
- Based on Linux Kernel
- Includes java compatible libraries
- Multilanguage support
- Developed by Open Handset Alliance led by Google

# What is android?

- It is basically an operating system for smartphones and tablets which is developed and supported by Google
- The android SDK provides the tools and APIs necessary to begin developing applications on the android platform using the Java Programming language
- Light weight
- open source.
- Highest number of applications in the world's market.

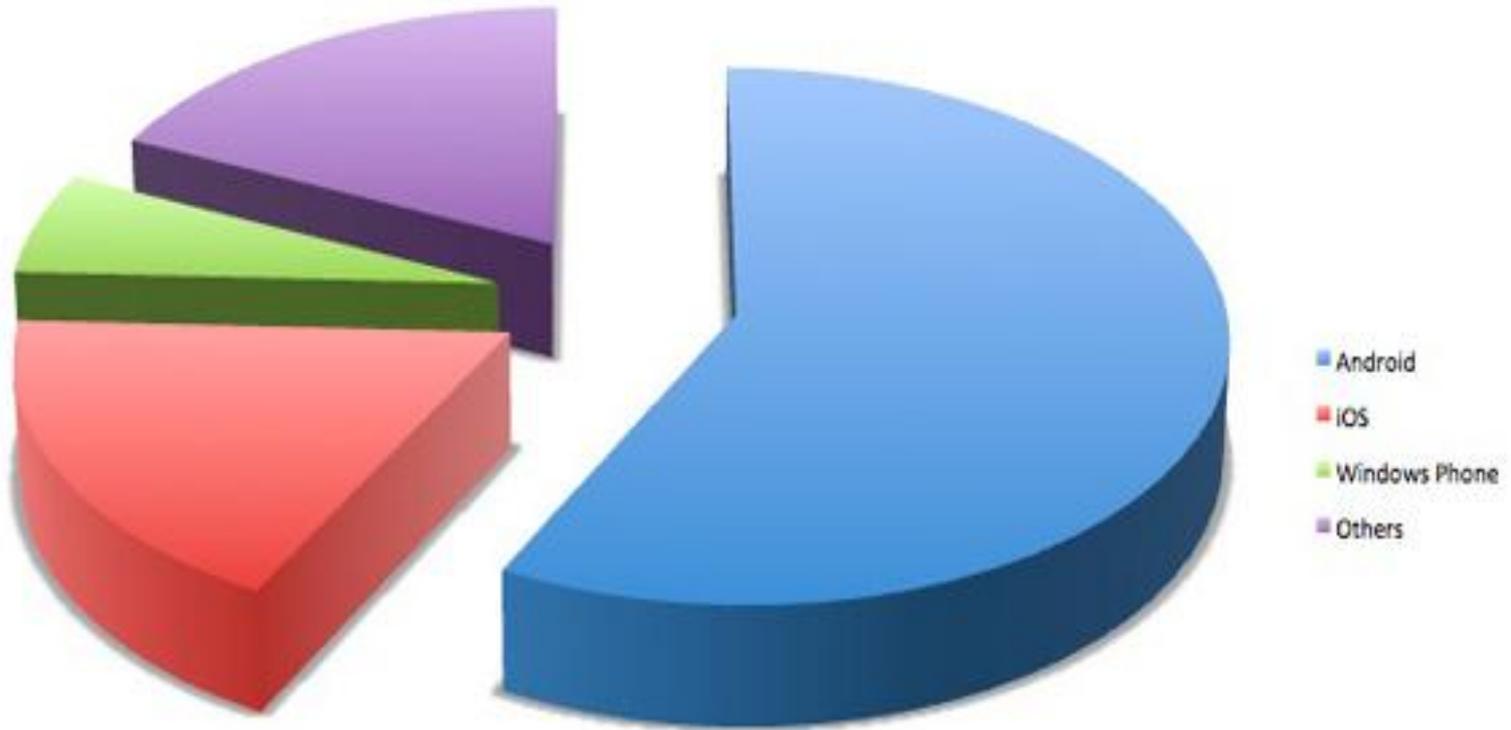
# History

- Initially developed by Android Inc.
- Google Financially backed and later purchased in 2005
- Android was out in 2007 along with the founding of the Open Handset Alliance by Google.
- First phone with android is HTC
- Google releases the Android code as open source, under the Apache License.
- The first Android-powered phone was sold in October 2008
- by the end of 2010 Android had become the world's leading smartphone platform
- Application of the operating system has also moved beyond mobile phones and tablets, currently televisions, netbooks and cameras are some of the types of devices Android is featured in.

# Android Market

- Available in goggle's play store
- largest part in the app market
- Largest number of application downloads per day compare to other operating system applications
- Google monitors every application to provide security for users
- more than 600,000 apps and games available on Google Play
- Many free useful applications.
- had a worldwide smartphone market share of 75% during the third quarter of 2012.

# Market Share in 2012



Source:

<http://www.tecflap.com/2011/12/27/research-expects-android-57-6-ios-18-1-windows-phone-6-2-market-share-2012/>

# Android design Philosophy

All the applications should be:

- **Fast**
  - It should be fast enough to meet the user requirements
- **Responsive**
  - Applications must respond to user actions within no time
- **Seamless**
  - Usability is key, persist data, suspend services
- **Security**
  - Applications declare permissions in manifest(will be discussed in later slides)

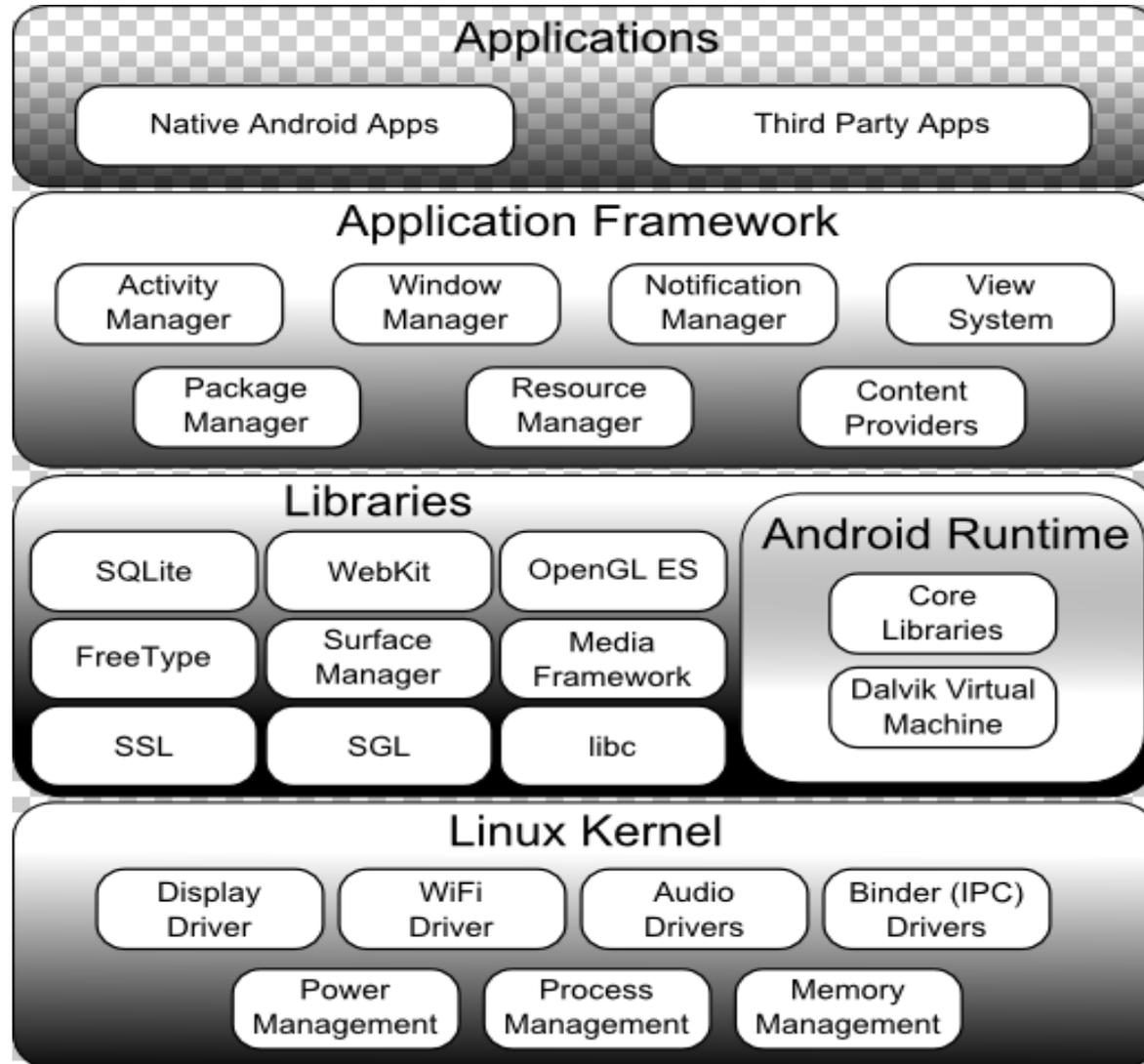
# Why it is so interesting?

- Open source, any one can work on it, any one can develop his own applications using android
- Compatible to the changes in the system
  - Dalvik will provide base for the android, so that it can run on any device which is having Dalvik operating system
- Easy to implement, flexible to modifications
  - Many online resources available to develop applications and use it. Supports for changes because it is purely Object Oriented design

# Why android

- Light weight
  - Works on smart phones and tablets. Uses SQLite database system.
- Fast and reliable
  - Works faster and better on smart phones and tablets.
- Every time new features are provided by the developers
  - Google is working on android to provide better and better services for android users
- User friendly APIs
  - Easy to use applications
- Powerful development framework
  - Android gives us everything we need to build best in class app experiences.

# Android Architecture



Source:

# Android Architecture

The software stack and android is composed of four layers

- **Application Layer**
  - Includes all native and third applications like emails, SMS, calendar, maps, contacts etc.
- **Application Framework**
  - All background services which includes starting application,, status bar updating, location services, sounds, notifications etc.
  - Provides access to non code resources such as localized strings, layout files etc.

# Android Architecture

- Libraries and runtime

Libraries includes

- System C libraries, media libraries, Surface Manager, SGL(2D libraries), 3D libraries, SQLite, FreeType etc.

Rum time includes

- Core libraries of JAVA that provides most of the functionality
- runs on Dalvik Virtual Machine

- Linux Kernel

- It is an abstract layer between hardware and upper layers(application, framework, libraries layers)
- Manages core system services like security, memory management, process management.

# Android SDK

- Android SDK is a software development kit which is used by the developers to create Android Applications.
- Android SDK includes development tools, emulator, sample projects and libraries to assist the developer in creating an application
- Applications are coded in Java and run on a virtual machine called Dalvik.

# Features

- Dalvik Virtual machine is optimized for mobile devices and tablets
- Integrated fast browser based on open source web kit engine
- Supports 2D and 3D graphics, smartphone layouts
- High connectivity
- Rich Media support

# Android Applications

- Android applications are written in the Java programming language.
- A unique aspect of the Android system design is that any application can start another application's component.
- Android applications run in a **SANDBOX** protecting it from the rest of the system resources.

# Components of Android application

Four Building blocks of Android application

- Activity
  - An *activity* represents a single screen with a user interface.
  - An activity is implemented as a subclass of Activity
- Service
  - It is a component that runs in the background to perform long-running operations or to perform work for remote processes.
- Content Provider
  - It manages the application data. We can store data in file system, SQLite database or on web.
- Broadcast Providers
  - It is a component that responds to system-wide broadcast announcements. Examples are when screen turned off, battery low etc.

# Framework API

- The Android platform provides a framework API that applications can use to interact with the underlying Android system. The framework API consists of:
  - A core set of packages and classes
  - A set of XML elements and attributes for declaring a manifest file
  - A set of XML elements and attributes for declaring and accessing resources
  - A set of Intents
  - A set of permissions that applications can request, as well as permission enforcements included in the system

# OOAD aspects of Android Application

- Everything in android application is an object
- We use java and xml to create application
- In XML we declare components as data objects
- These two are purely object oriented language
- In XML we store components in terms of objects.
- Using java we perform the objects operations.
- Special android java frame works for code reusability and it also includes MVC design pattern which is a good object oriented design.

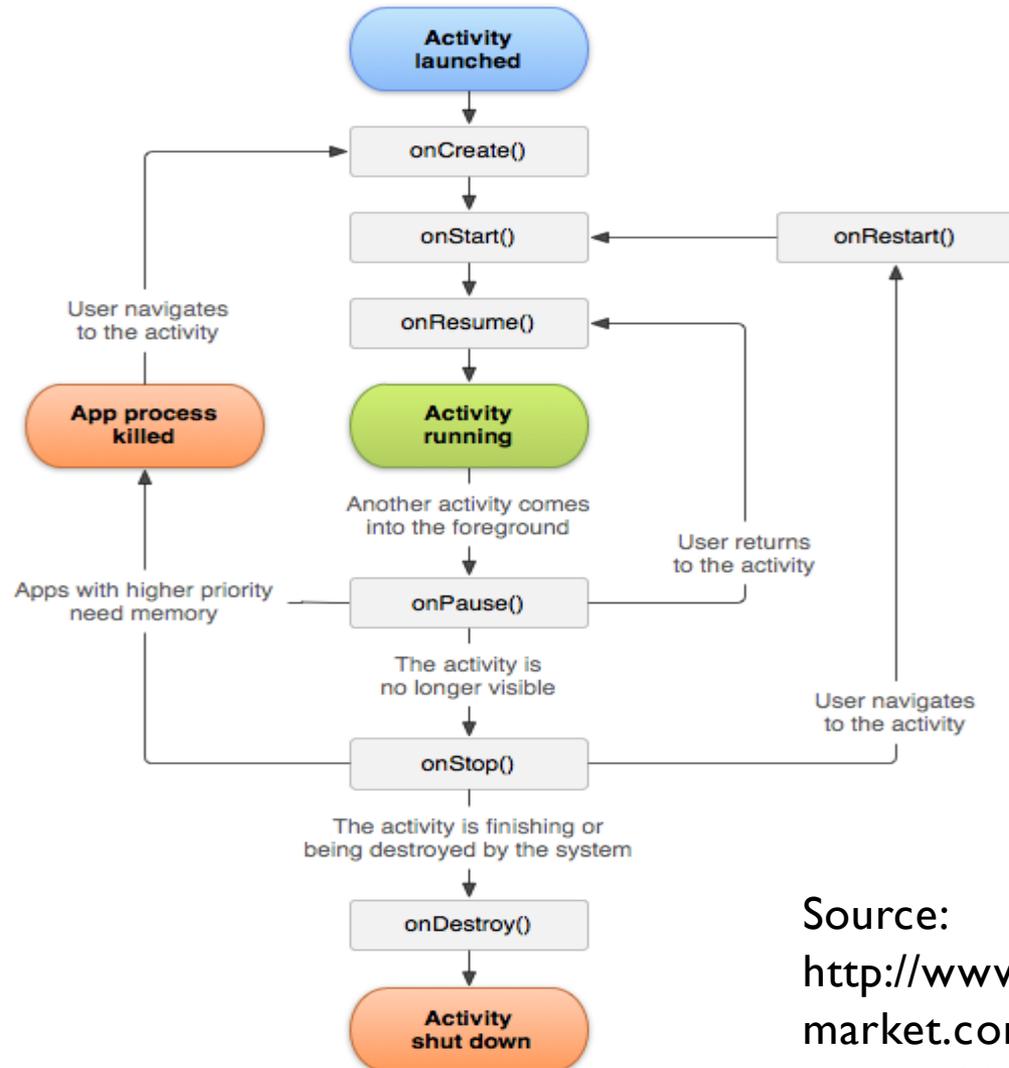
# Manifest File

- Android System will know about the system components by looking into **AndroidManifest.xml** file.
- Other operations by Manifest file:
  - Identify user permissions the application requires
  - Declare hardware and software features used or required by the application
  - API libraries the application needs to be linked against etc.
- Some legal elements in AndroidManifest.xml are
  - <action>
  - <activity>
  - <activity-alias>
  - <application>
  - <category>
  - <data>

# Activities

- An **Activity** is an application component that provides a screen with which users can interact in order to do something.
- Application contains multiple activities (each activity for each user interface)
- Need to be declared in the manifest file
- It is treated as a higher level object where it will have a life cycle.
- Different operations includes onCreate(), onStart(), onResume(), onPause(), onResume(), onStop(), onDestroy().

# Activity Life Cycle



Source:  
<http://www.android-app-market.com/android-activity-lifecycle.html>

# Activity components

## Three components

- **Fragment**
  - Fragments decompose application functionality and UI into reusable modules
- **Loaders**
  - Provides asynchronous loading of data
  - When content changes, delivers new data
  - Provides automatic reconnection so that avoids re-querying the same data.
- **Tasks and Back Stack**
  - task is a collection of activities that users interact with when performing a certain job.
  - The activities are arranged in a stack (the "back stack"), in the order in which each activity is opened

# Permissions

- A *permission* is a restriction limiting access to a part of the code or to data on the device.
- By default no application have permission on other application
- No dynamic permission granting because of design complexity
- We can define permissions for an applications explicitly in manifest file

```
<uses-permission  
  android:name="android.permission.ACCESS_FINE_LOCA  
  TION" />
```

# Two Example concepts

- 1. Access the camera.
  - We are going to look into how to create simple application which uses media or camera of the system
- 2. Data Storage System
  - We are going to look into different type of storage systems in android.

# Standalone Application Example: accessing camera

- In this example, We are going to learn how to create simple basic android application which access the existing camera application using eclipse.
- Prerequisites:
  - Android 4.1 SDK installed on system
  - Eclipse with android software
  - Java JDK7 compatible with Eclipse
  - At least one AVD manager
  - An Emulator

To know how to install all these, please refer the below link:

<http://developer.android.com/tools/sdk/eclipse-adt.html>

# Eclipse overview for android

Under the project name mainly we will find

Below folder which we are going to use frequently.

Src – this folder consists of all java files. These are the main source files

Res – contains folders for app resources. Some important folders are:

drawable-hdpi –

Directory for drawable objects (such as bitmaps) that are designed for high-density (hdpi) screens

Layout ---

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

Values –

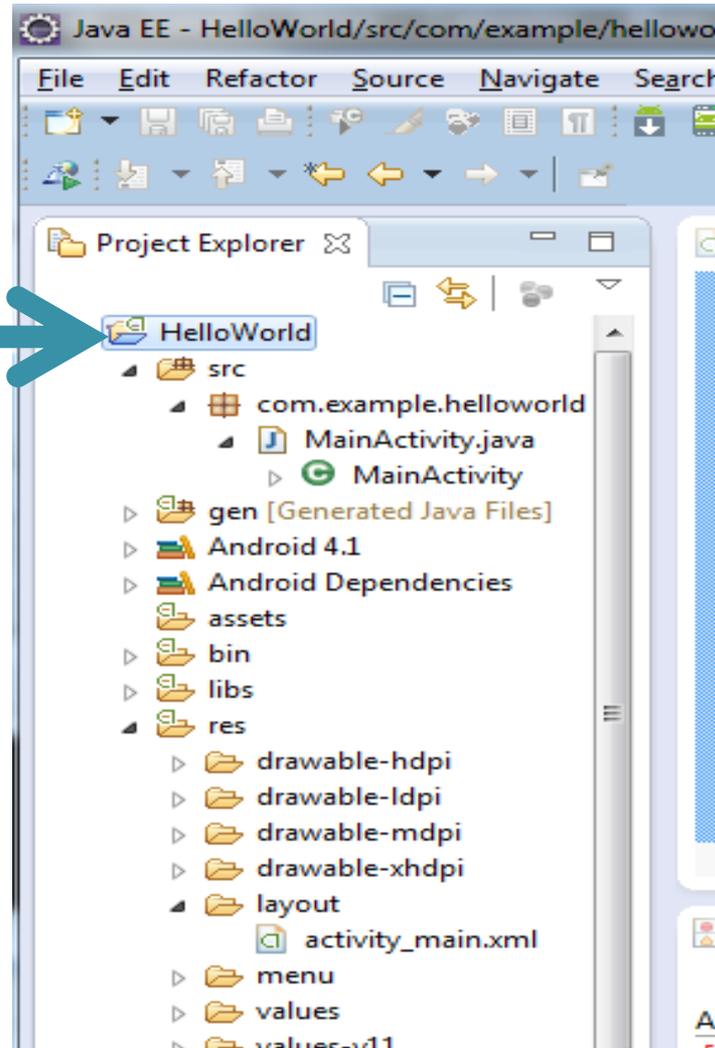
Directory for other various XML files that contain a collection of resources, such as string and color definitions.

AndroidManifest.xml –

We discussed these in previous slides

# Sample Eclipse files structure

HelloWord  
is the  
project  
name



# Key Classes used for Camera application

- **Camera** – This class controls device cameras. It will be used for taking pictures for our application
- **SurfaceView** – provides live camera preview
- **Intent** – This class can be used to capture without accessing the camera object.

# Manifest declaration

- As discussed in the previous slides, manifest file need to be declared before starting application
- We need to specify the camera permissions for accessing camera , storage permissions for storing pictures and some camera features in the AndroidManifest.xml(Manifest file) which is located in project\_name/res .

```
<uses-permission  
android:name="android.permission.CAMERA" />
```

```
<uses-permission  
android:name="android.permission.WRITE_EXTERNAL  
STORAGE" />
```

```
<uses-feature  
android:name="android.hardware.camera" />
```

# Using Intent Class to Access camera

- A good example of reuse or delegation.
- How?
  - We are developing our own application and handling basic operations to the existing application.
- Intent class will allow us to use the existing camera application in android.
- Easier to way of building our applications using existing applications

# Procedure

- In the 'src>MainActivity.java we need to create an camera Intent class object

```
Intent in = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
```

- start the Camera intent

```
startActivityForResult(in, CAPTURE_IMAGE_ACTIVITY_REQUEST_CODE);
```

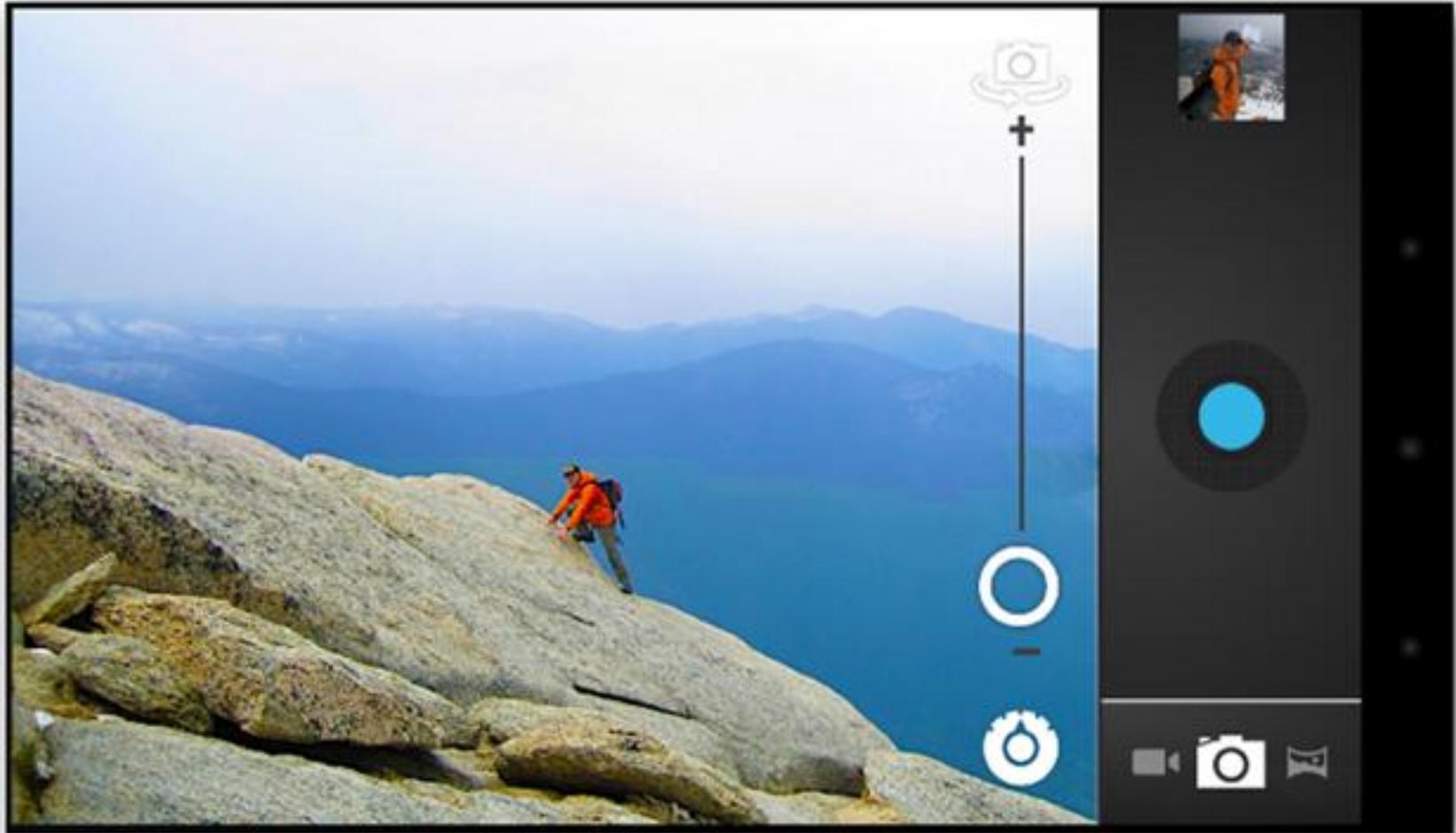
Here, when the above method is executed, the user can see the our own app interface.

- Receive the Intent result

- `private static final int CAPTURE_IMAGE_ACTIVITY_REQUEST_CODE = 100;`
- Whenever the user captures the picture, after receiving the above code , we will get the media data by `getdata()`, so that we can save the file.

(go for coding)

We can create our own user interface and options by using exist camera application



Source: <http://www.vikitech.com/6933/android-4-ice-cream-sandwich-major-features>

# Data Storage: SQLite DataBase

- In android we have several data storage options. They are:
  - SQLite
    - Store structured data in database
  - Shared Preferences
    - Store data in key value pairs
  - Internal Storage
    - Store on device
  - External Storage
    - Store data on shared external device
  - Network Connections
    - Store data on network servers



# SQLite Database

- What is SQLite?
  - SQLite is a software library that implements a self-contained , server less, zero-configuration , transactional SQL database engine
- Android provides full support to SQLite database with class **SQLiteOpenHelper** and **SQLiteDatabase** classes.
- We can create a new SQLite database by inheriting the **SQLiteOpenHelper** class.

# SQLite Database Operations

- `getWritableDatabase()`
  - Which is useful for writing data into DataBase. It will return the `SQLiteDatabase` object, so that we can perform database operations
- `getReadableDatabase()`
  - Which is useful for reading data into DataBase. It will return the `SQLiteDatabase` object, so that we can perform database operations

# Best Application implementation mechanisms

- Good use of design patterns
- Avoiding unnecessary object creation
- Prefer Static over Virtual
- Avoid internal getter or setters
- Prefer Static Final for Constant values
- Avoid using unnecessary libraries
- Use floating points carefully

# Android Versions

- Google releasing new android with new features. The latest version released as of November 2012 is Android 4.1.2.
- Previous versions are:
  - 1.1 Android beta
  - 1.2 Android 1.0
  - 1.3 Android 1.1
  - 1.4 Android 1.5 Cupcake
  - 1.5 Android 1.6 Donut
  - 1.6 Android 2.0/2.1 Eclair
  - 1.7 Android 2.2.x Froyo
  - 1.8 Android 2.3.x Gingerbread
  - 1.9 Android 3.x Honeycomb
  - 1.10 Android 4.0.x Ice Cream Sandwich
  - 1.11 Android 4.1/4.2 Jelly Bean

# Summary

- Android is a powerful, open source operating system for smart phones and tablets.
- Within less than 5 years of its first release, it has taken the larger part of the mobile application project.
- It is easy, light weight, fast and reliable.
- All android applications are written in java, a fully object oriented design language.
- A new virtual machine is designed to support rich media, 2D,3D graphics, high connectivity and much more.
- We can use existing applications to create our own applications in easy and structured way which is shown in camera access application.
- Android mostly uses SQLite which is self contained, server less and zero configured so that all the operations are light weight.
- In the last, we also discussed standards of android application design.



# References

- [1] <http://developer.android.com/guide>
- [2] lectures by prof. Ken Anderson
- [3] <http://developer.android.com/tools/sdk/eclipse-adt.html>
- [4] <http://www.android-app-market.com/android-activity-lifecycle.html>
- [5] <http://www.ign.com/articles/2011/12/23/the-history-of-android>