

LẬP TRÌNH ANDROID CƠ BẢN

BÀI 1 : GIỚI THIỆU ANDROID



T. V
ITC

BÀI 1 : GIỚI THIỆU ANDROID



PHẦN I : TỔNG QUAN VỀ ANDROID

T. V. ITC

Phần I: Giới thiệu về Android

 Các hệ điều hành trên di động

 Tổng quan về android

 Kiến trúc Android

 Môi trường phát triển

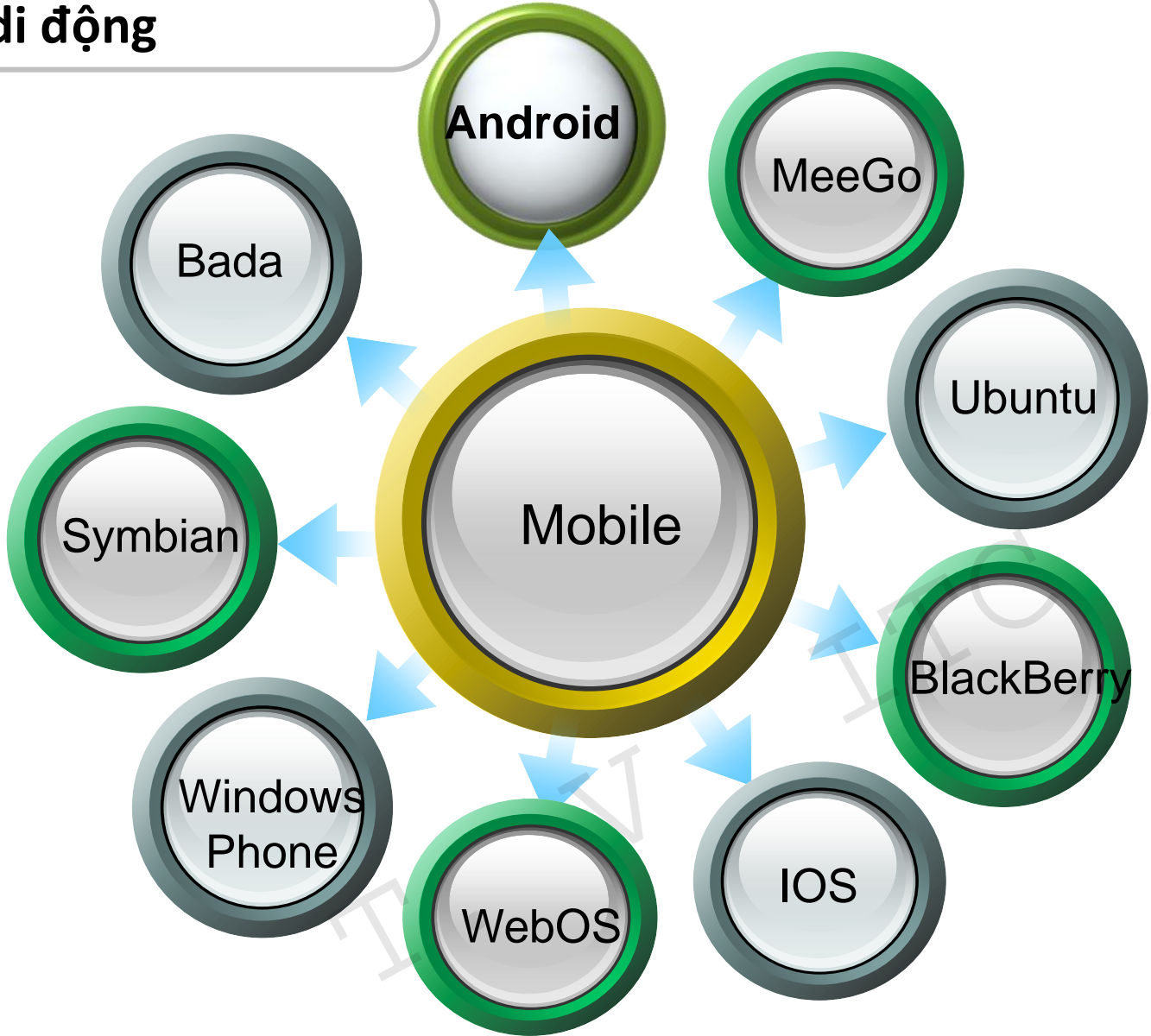
Phần II: Sử dụng Android Studio

 Tạo project Android trên Android Studio

 Tìm hiểu cấu trúc project

 Biên dịch và gỡ lỗi ứng dụng

Các HĐH trên di động





<http://developer.bada.com>



Samsung
Closed source



Samsung **Wave 3**
The Chic Smartphone

bada
**C++ Application
Programming**





<http://www.developer.nokia.com/>

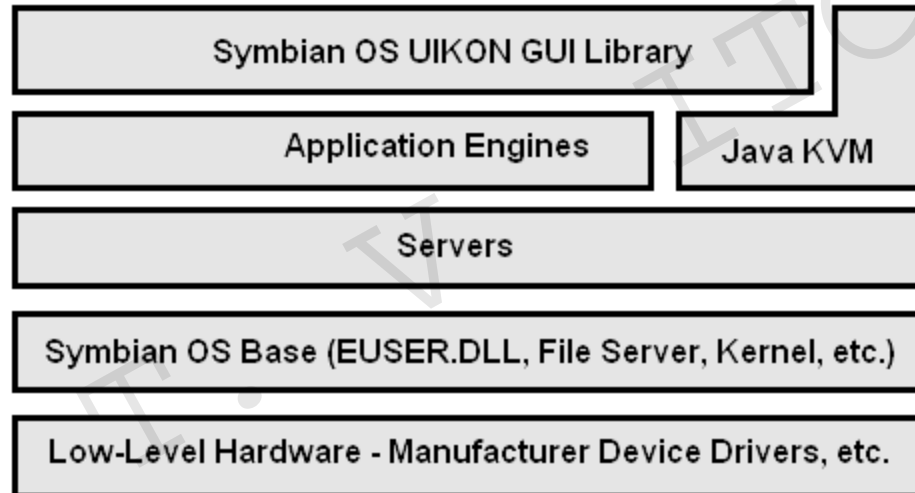
C/C++ J2EM



NOKIA Developer



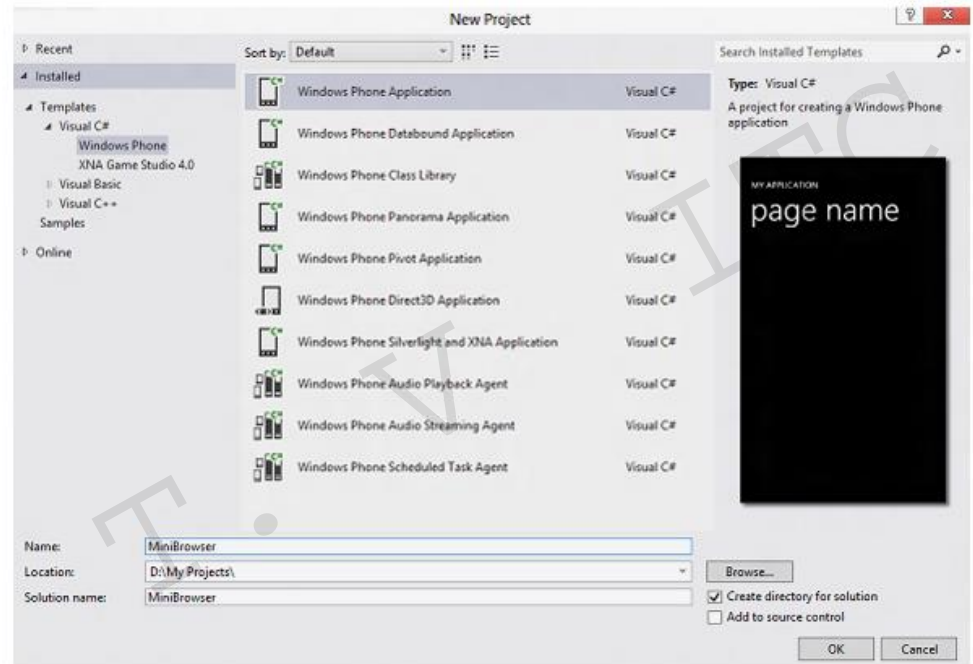
Symbian OS generic technology structure





<http://dev.windowsphone.com>

C#
VB.net
XAML





<https://developer.palm.com/>

from HP

open sourced

running on the Linux kernel

Visual Studio

C/C++

Java



T.V ITC



<https://developer.apple.com>

Apple

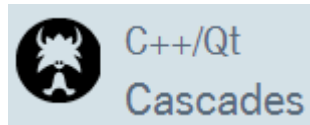
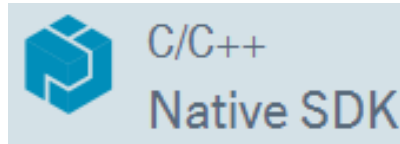
derived from Mac OS X.

Objective C – iPhone SDK





<https://developer.blackberry.com/>



Platform *PlayBook OS*





<http://developer.android.com/>

Google Inc

free and open source

Linux-derived OS

Java (Android SDK)





<http://www.ubuntu.com/devices/android>

Professional Ubuntu Mobile Development



ubuntu^o phone



<https://meego.com/developers>

open source
Nokia + Intel

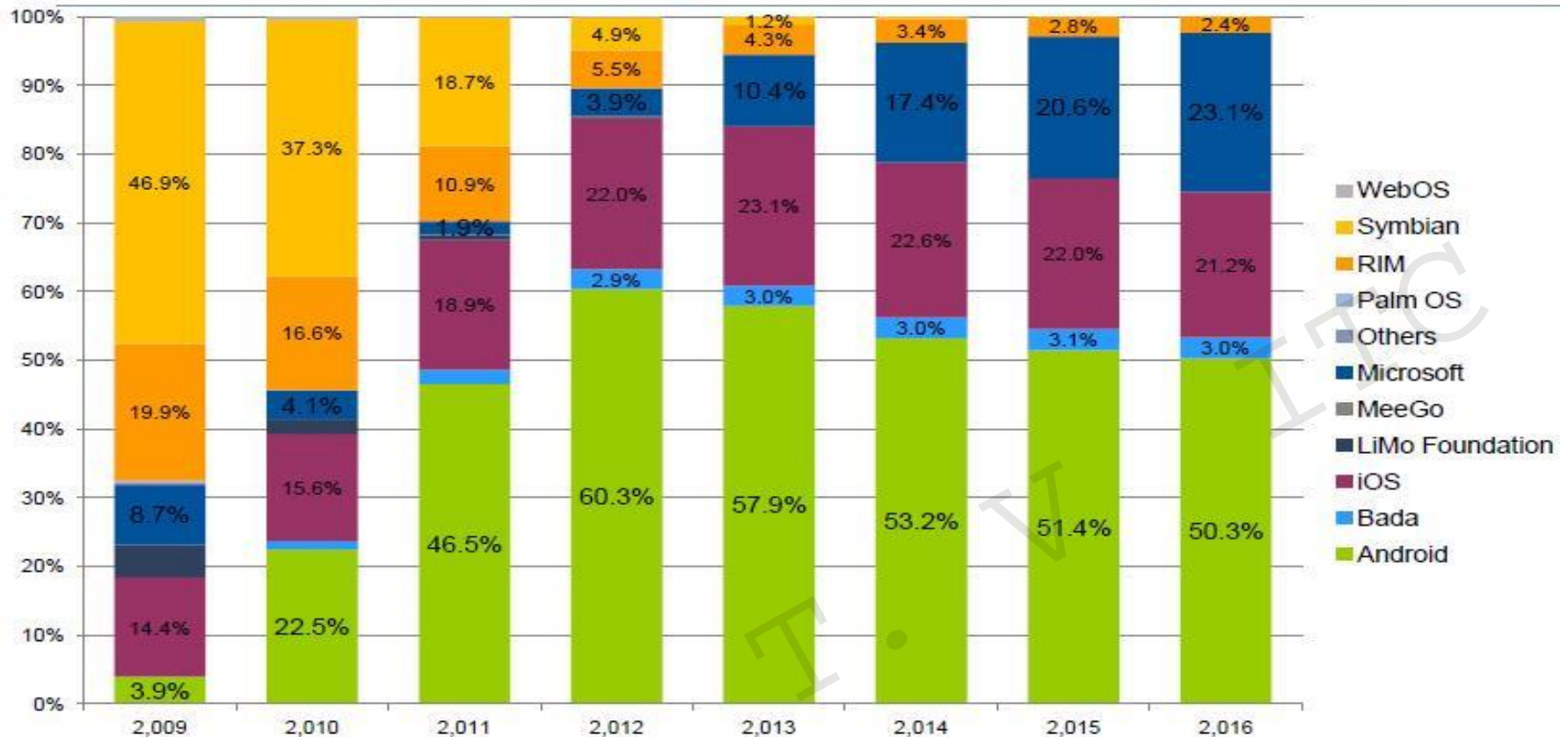
C++



TỔNG QUAN VỀ ANDROID

So sánh các nền tảng mobile platform

Gartner Forecast Estimates Mobile OS Sales by Market Share (2009-2016)



Source: Gartner

Forecast: Mobile Devices by Open Operating System, Worldwide, 2009-2016, 2Q12 Update

TỔNG QUAN VỀ ANDROID



- Android là nền tảng phần mềm mã nguồn mở được Google và Open Handset Alliance tạo ra
- Các dịch vụ hệ thống cơ bản của Android được xây dựng trên Linux 2.6
- Android SDK cung cấp công cụ và API để phát triển ứng dụng Android sử dụng ngôn ngữ Java và Kotlin

TỔNG QUAN VỀ ANDROID

❑ Lịch sử phát triển

- Năm 2003, Android Inc. được thành lập bởi Andy Rubin, Rich Miner, Nick Sears và Chris White tại California.
- Năm 2005, Google sở hữu Android cùng với các vị trí quản lý.
- Năm 2007, OHA (Open Handset Alliance) được thành lập bởi Google cùng với 47 nhà sản xuất thiết bị phần cứng, thiết bị không dây và vi xử lý. Công bố nền tảng phát triển Android.
- Năm 2008, thiết bị HTC Dream là phiên bản thế hệ đầu tiên hoạt động với hệ điều hành Android 1.0.
- Năm 2010, Google khởi đầu dòng thiết bị Nexus với thiết bị đầu tiên của HTC là Nexus One.
- Năm 2013, ra mắt loạt thiết bị phiên bản GPE.
- Năm 2014, Google công báo Android Wear, hệ điều hành dành cho các thiết bị đeo được.

TỔNG QUAN VỀ ANDROID

Open handset alliance

- Liên minh gồm trên 80 công ty cùng xây dựng nền tảng mobile mới (Android)



TỔNG QUAN VỀ ANDROID

Lịch sử phát triển của Android



CÁC PHIÊN BẢN CHÍNH CỦA ANDROID



Giao diện qua các phiên bản Android

Sept.2008



Android 1.0 on the HTC Dream

Feb.2009



Android 1.1

Apr.2009



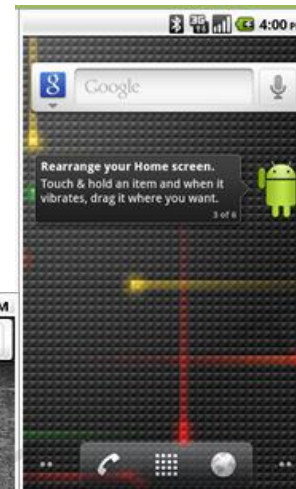
Android 1.5 on the HTC Magic

Oct.2009



Android 2.0 Eclair on the Motorola Droid

May.2010



Android 2.2 Froyo on the Nexus One

Dec.2010



Android 2.3 on Google's Nexus S

24 July 2013

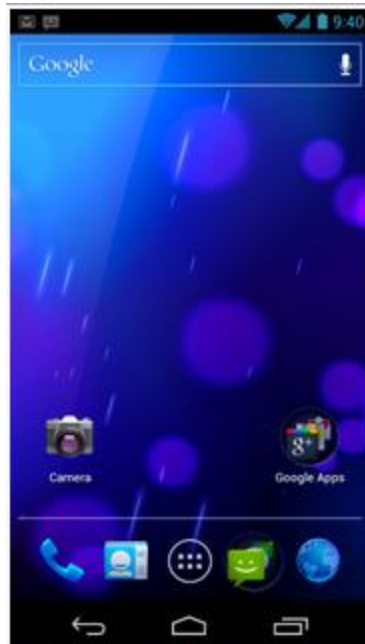
Nov.2012

Oct.2011

Feb.2011



Android 3.0 on the Motorola Xoom tablet



Android 4.0 on the Samsung Galaxy Nexus

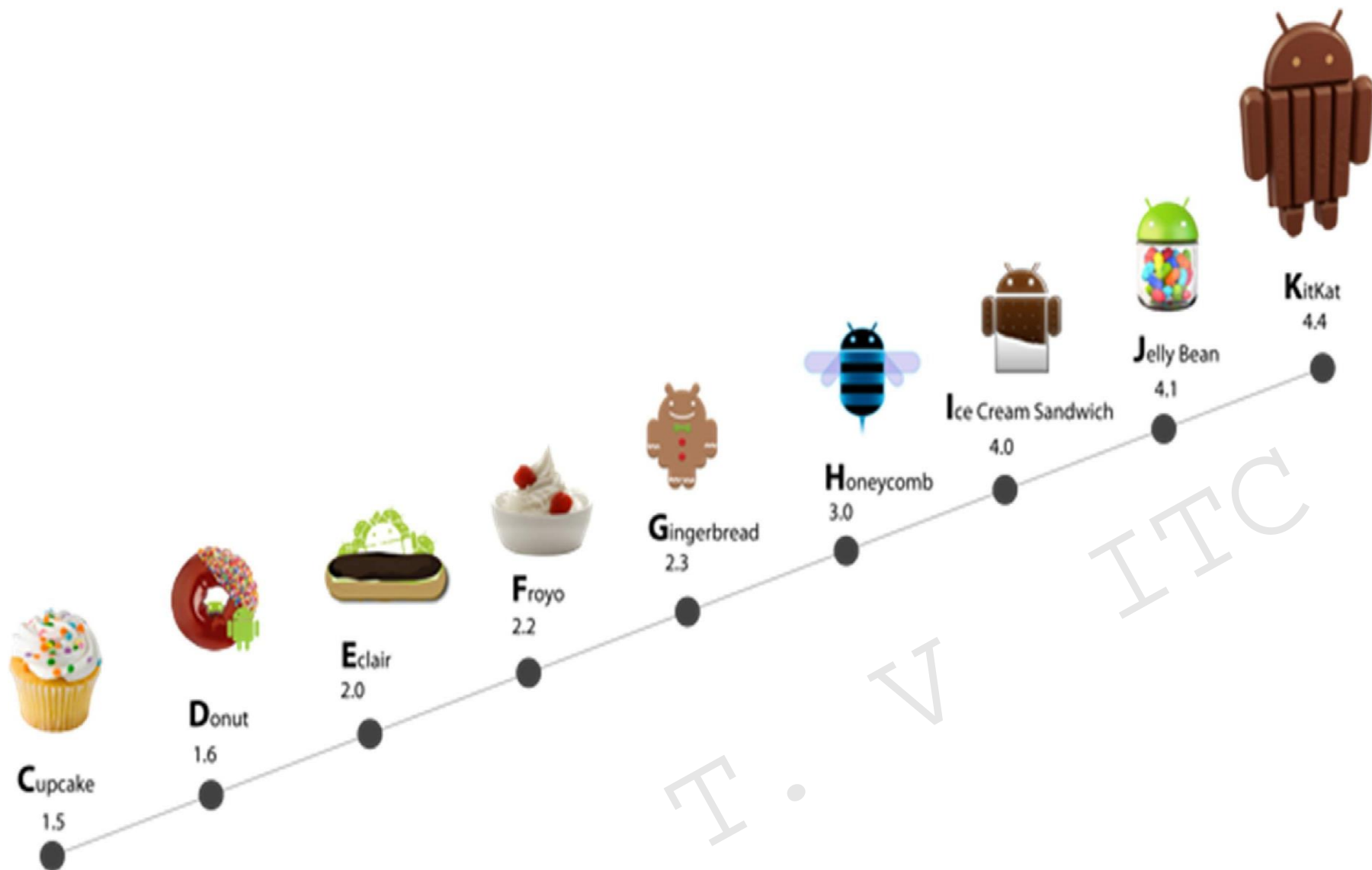


Android 4.2 on the LG Nexus 4



Android 4.3 on the new Nexus 7 (2013 version).

☐ Các phiên bản với tên mã

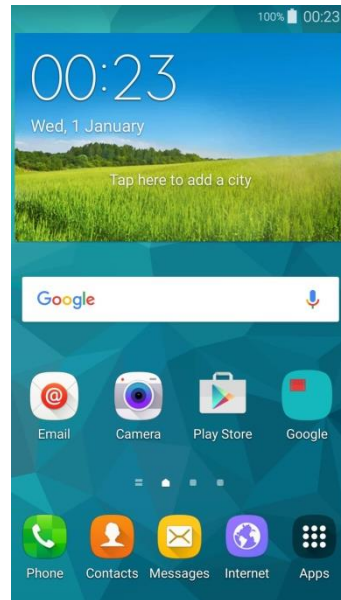




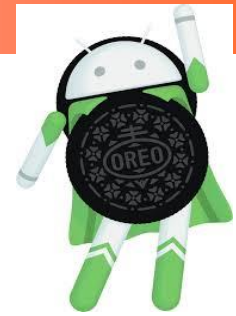
5.0 Lollipop
JUL 2014



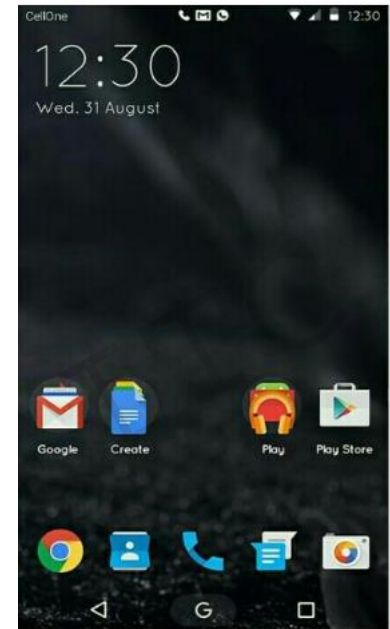
6.0 Marshmallow
2015



7.0 Nougat
2016



Android 8.0 Oreo
2017



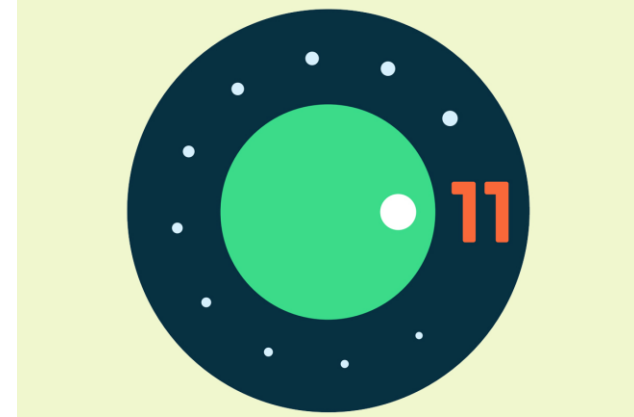
9.0 Pie
AUG 2018



10 Q



Android 11 R



Android 12 S



Android 13 T beta



IITC

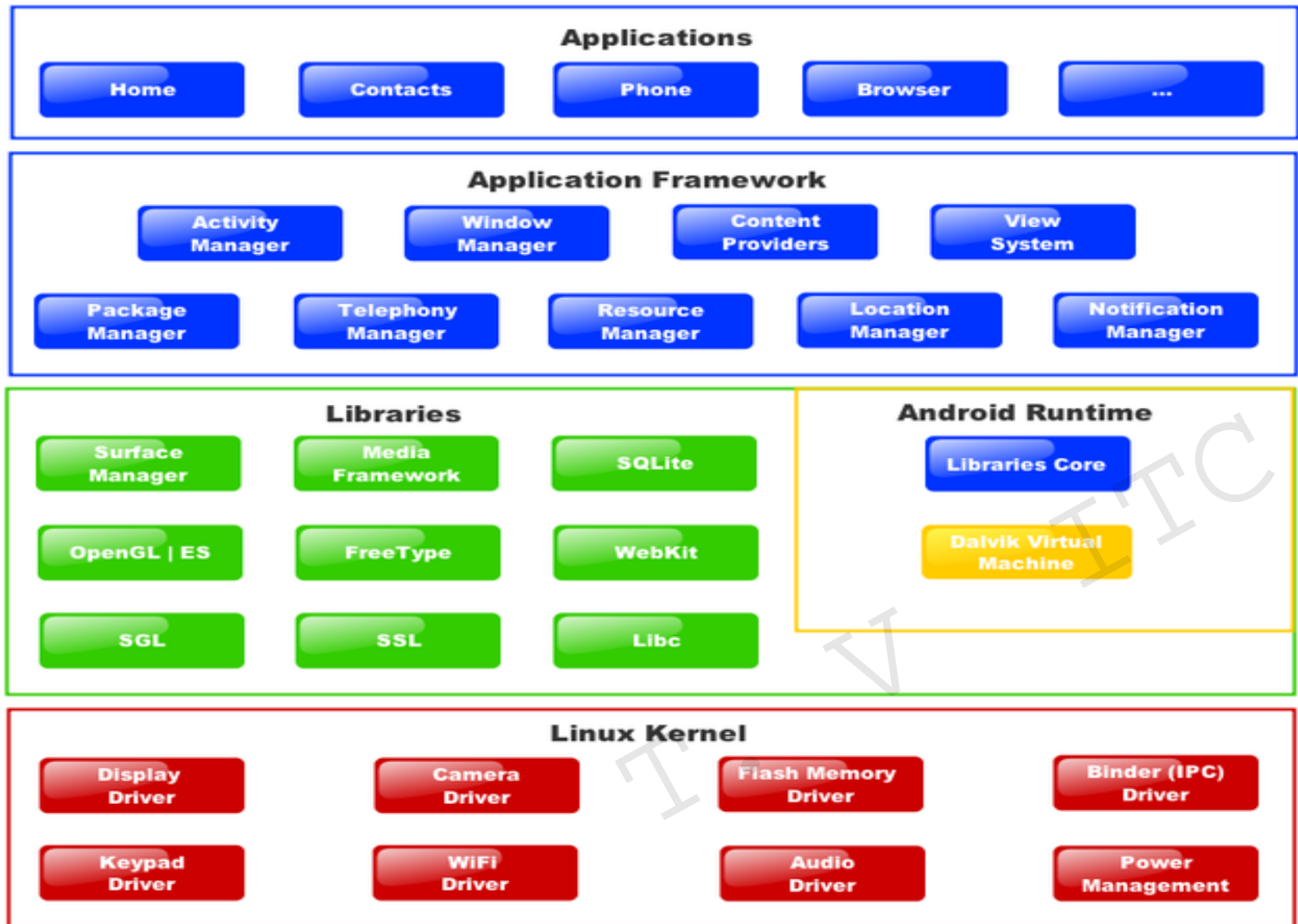
TỔNG QUAN VỀ ANDROID



Android platform

- Là môi trường phần mềm được xây dựng cho thiết bị di động
- Android bao gồm:
 - Hệ điều hành dựa trên nhân Linux
 - Giao diện người dùng
 - Ứng dụng người dùng cuối
 - Thư viện code
 - Application framework

KIẾN TRÚC ANDROID



Nhân Linux:

- Android dựa trên Linux version 2.6 gồm các dịch vụ hệ thống cơ bản như bảo mật, quản lý bộ nhớ, quản lý tiến trình, network và driver
- Nhân linux là tầng trừu tượng giữa phần cứng và software stack

Linux Kernel

Display
Driver

Camera
Driver

Flash Memory
Driver

Binder (IPC)
Driver

Keypad
Driver

WiFi
Driver

Audio
Driver

Power
Management

Thư viện:

- Android bao gồm tập hợp các thư viện C/C++ sử dụng bởi các thành phần khác nhau của Android System
- Nhà phát triển khai thác thư viện này thông qua Android application framework. Một số thư viện cơ bản được liệt kê ở dưới:

Libraries

Surface
Manager

Media
Framework

SQLite

OpenGL | ES

FreeType

WebKit

SGL

SSL

Libc

Android Runtime

Libraries Core

Dalvik Virtual
Machine

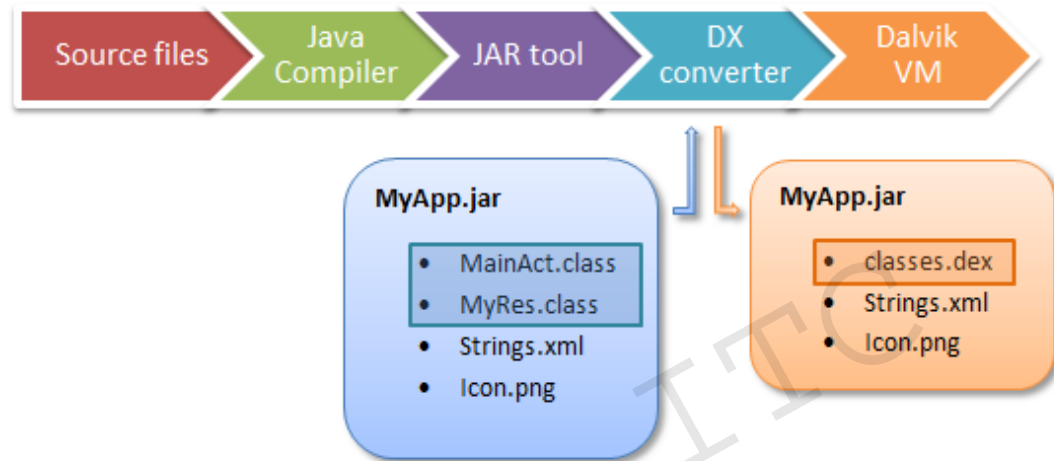
Android runtime

- Dalvik VM:
 - Dex files
 - Gọn nhẹ và linh hoạt hơn file class
 - Tiết kiệm bộ nhớ và pin
- Thư viện cơ bản:
 - Java 5 Std edition
 - Collections, I/O etc...



Dalvik VM

- Là công nghệ VM mã nguồn mở do Google phát triển cho hệ điều hành Android
- Register-based virtual architecture
- Tất cả ứng dụng được viết bằng Java và chuyển đổi thành file dex có thể chạy được trên Dalvik

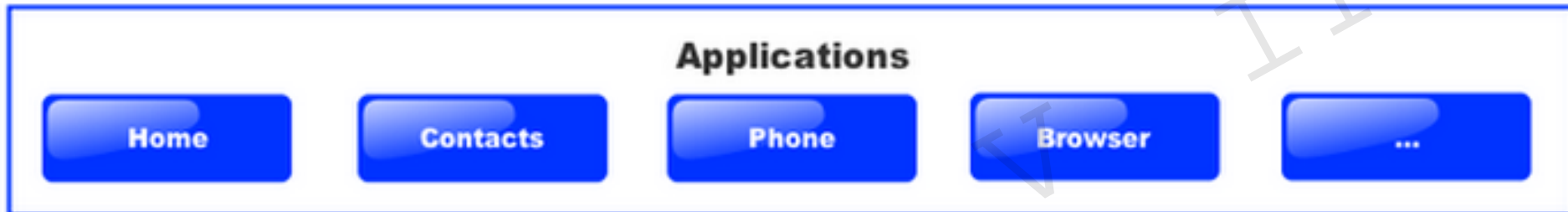


- Application framework



- Nhà phát triển có thể sử dụng framework API được sử dụng bởi core application
- Kiến trúc hệ thống được thiết kế để đơn giản hóa việc tái sử dụng các thành phần

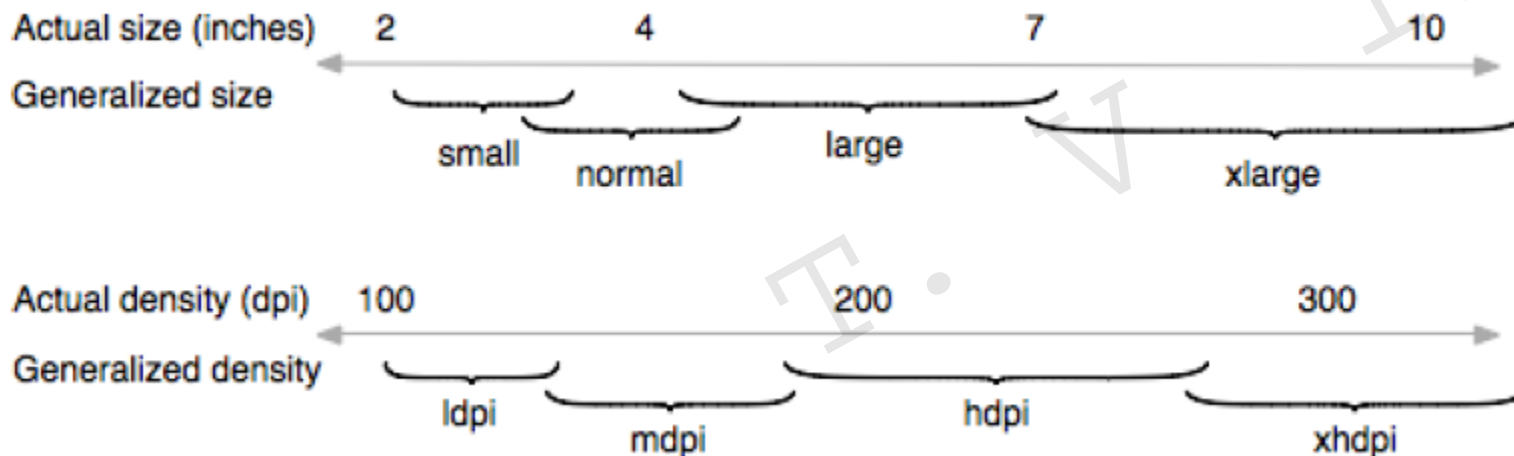
- Application
- Các phiên bản Android bao gồm các ứng dụng cơ bản như email, sms, calendar, map, contact,...
- Tất cả ứng dụng được viết bằng ngôn ngữ lập trình Java



1. Màn hình

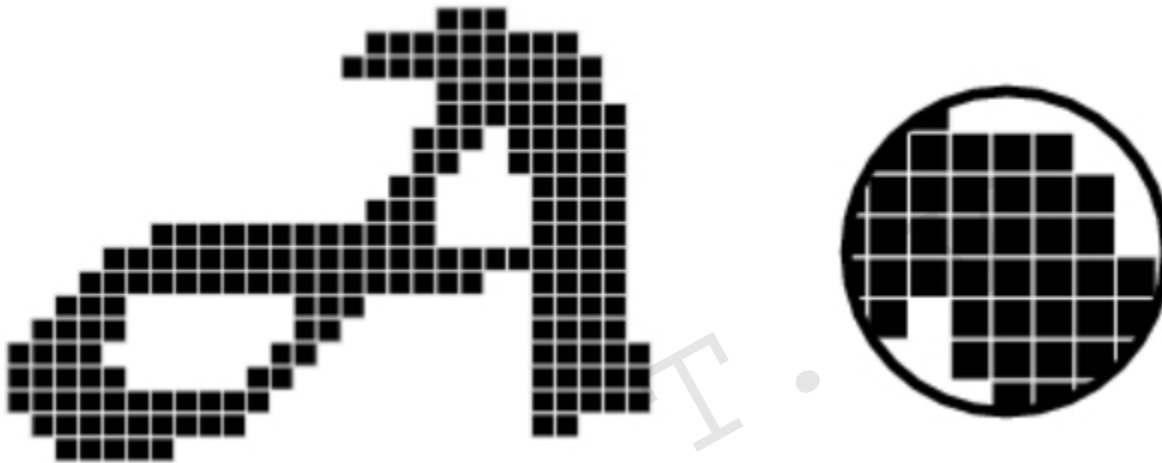
- ❑ Kích thước màn hình: đo theo đường chéo của màn hình, bao gồm: *small, normal, large, extra large, xxl*
- ❑ Screen density: mật độ điểm ảnh trên 1 inch – dpi (dots per inch): *low, medium, high, and extra high, xxh*
- ❑ Orientation: *landscape or portrait*
- ❑ Resolution (Độ phân giải): tổng số pixels trên màn hình

Chú ý: ứng dụng không quan tâm đến độ phân giải, chỉ quan tâm đến kích thước màn hình và mật độ điểm ảnh



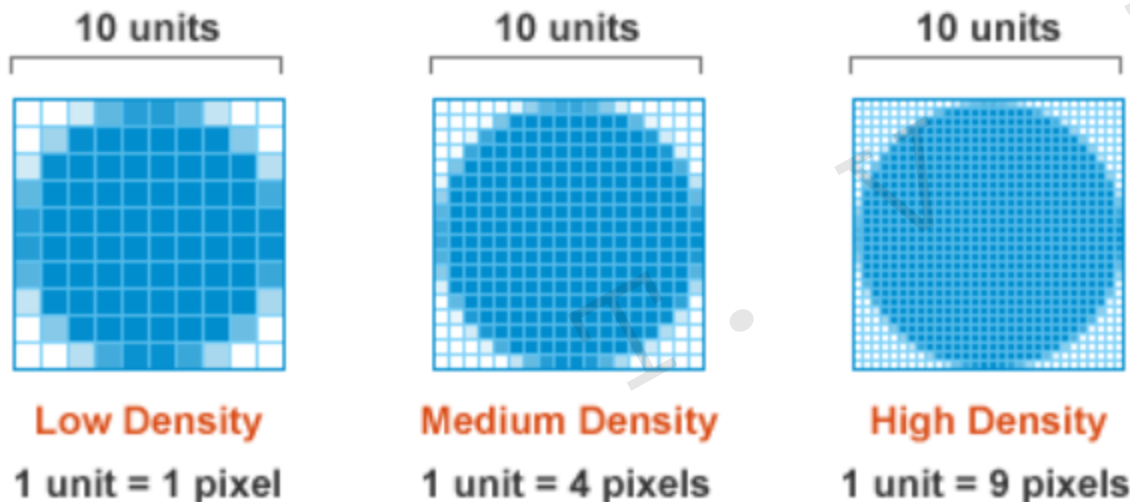
2. Các đơn vị đo lường trong Android

- ▶ **In (Inches)** - dựa vào kích thước vật lý của màn hình
- ▶ **Pt (point)** : $1 \text{ pt} = 1/72 \text{ inch}$ dựa vào kích thước vật lý của màn hình, được APPLE dùng trong IOS.
- ▶ **Px (Pixels)** - tương ứng với điểm ảnh thực tế trên màn hình



2. Các đơn vị đo lường trong Android

- ▶ **Dip / Dp (Density-independent Pixels)** - Đây là một đơn vị đo chiều dài vật lý cũng giống như inch, cm, mm... mà Google thường áp dụng để đo kích thước màn hình của thiết bị
 - ▶ **160 dp = 1 inch** - điều này có nghĩa **1dp = 1/160 = 0.00625 inch**
 - ▶ 1 dp có thể chứa 1 hay nhiều pixel.



2. Các đơn vị đo lường trong Android

- ▶ **DPI – Dots per inch hay PPI – Pixels per inch:**
Dots per inch là số điểm ảnh (pixels) trên 1 inch của màn hình thiết bị, con số này càng lớn thì màn hình thiết bị hiển thị hình ảnh càng mịn và đẹp.
- ▶ Dựa vào dpi người ta chia làm loại màn hình như sau:
 - ▶ small: **ldpi** (120 dpi)
 - ▶ normal: **mdpi** (160 dpi)
 - ▶ large: **hdpi** (240 dpi)
 - ▶ x-large: **xhdpi** (320 dpi)
 - ▶ xx-large: **xxhdpi** (440 dpi)
 - ▶ xxx-large: **xxxhdpi** (560 dpi)

2. Các đơn vị đo lường trong Android

- ▶ Với mỗi loại này thì 1 dp tương ứng với số lượng pixels khác nhau, được tính theo công thức:

$$\text{dp} = \text{px} / (\text{dpi} / 160)$$

- ▶ *Lưu ý dp hay dip không được nhầm lẫn với dpi (dots per inch).*
- ▶ ví dụ: Tính kích thước (rộng và cao) màn hình theo dp với thiết bị có màn hình hdpi(240) kích thước 480 x 800 như sau:
 - + Chiều rộng (dp) = $480 / (240/160) = 320$ dp
 - + Chiều cao (dp) = $800 / (240/160) = 534$ dp

2. Các đơn vị đo lường trong Android

- ▶ **SP - Scale-independent Pixels:** Cũng tương tự như dp, nhưng sp thường được dùng cho font size của văn bản.

Text Size Micro	12sp
Text Size Small	14sp
Text Size Medium	16sp
Text Size Large	18sp

Môi trường phát triển: Eclipse & ADT và Android Studio

- Android Development Tools (ADT) plugin cho Eclipse để phát triển Android
- Cho phép tạo và gỡ lỗi ứng dụng Android dễ dàng và nhanh chóng
- Google đang phát triển IDE mới cho Android có tên là Android studio IDE



Chuẩn bị môi trường

- Yêu cầu hệ thống
- Cài đặt JDK
- Tải và cài đặt Android Studio

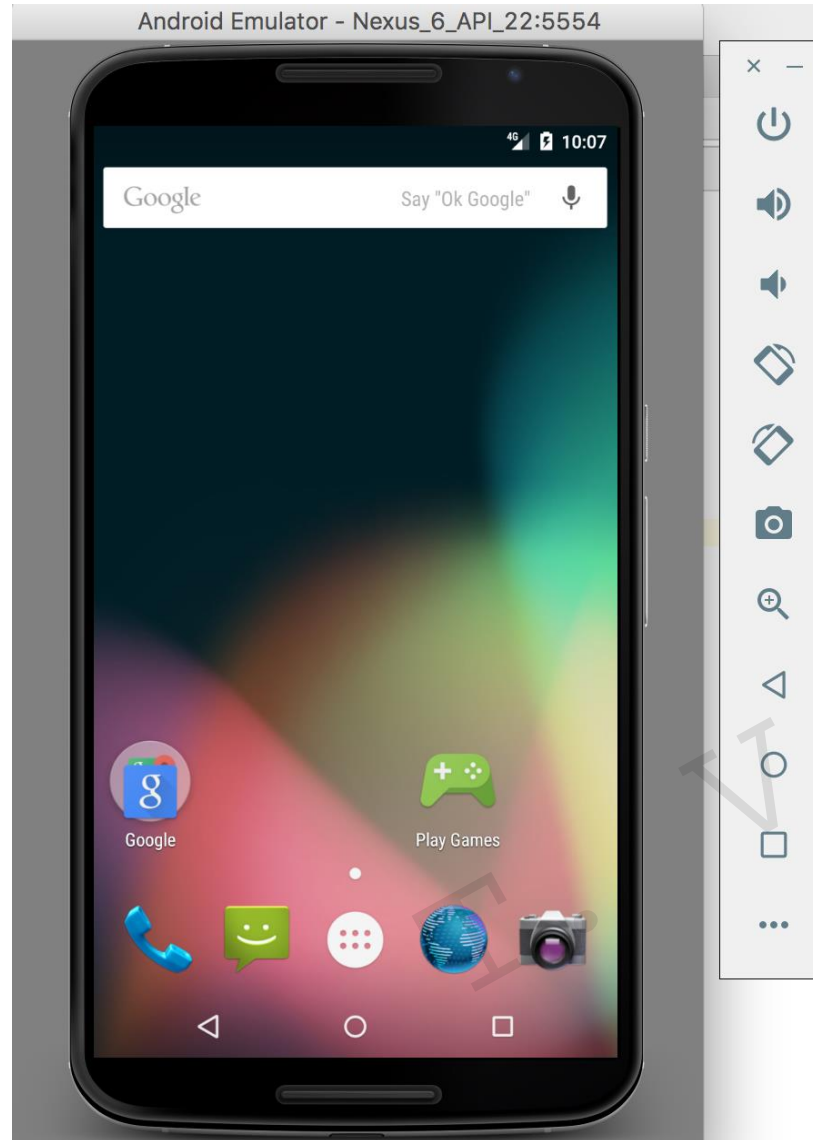


Connecting Internet from Laptop to Android Devices

<https://developer.android.com/studio/install.html>

MÔI TRƯỜNG PHÁT TRIỂN

Android Studio emulator



BÀI 1 : GIỚI THIỆU ANDROID

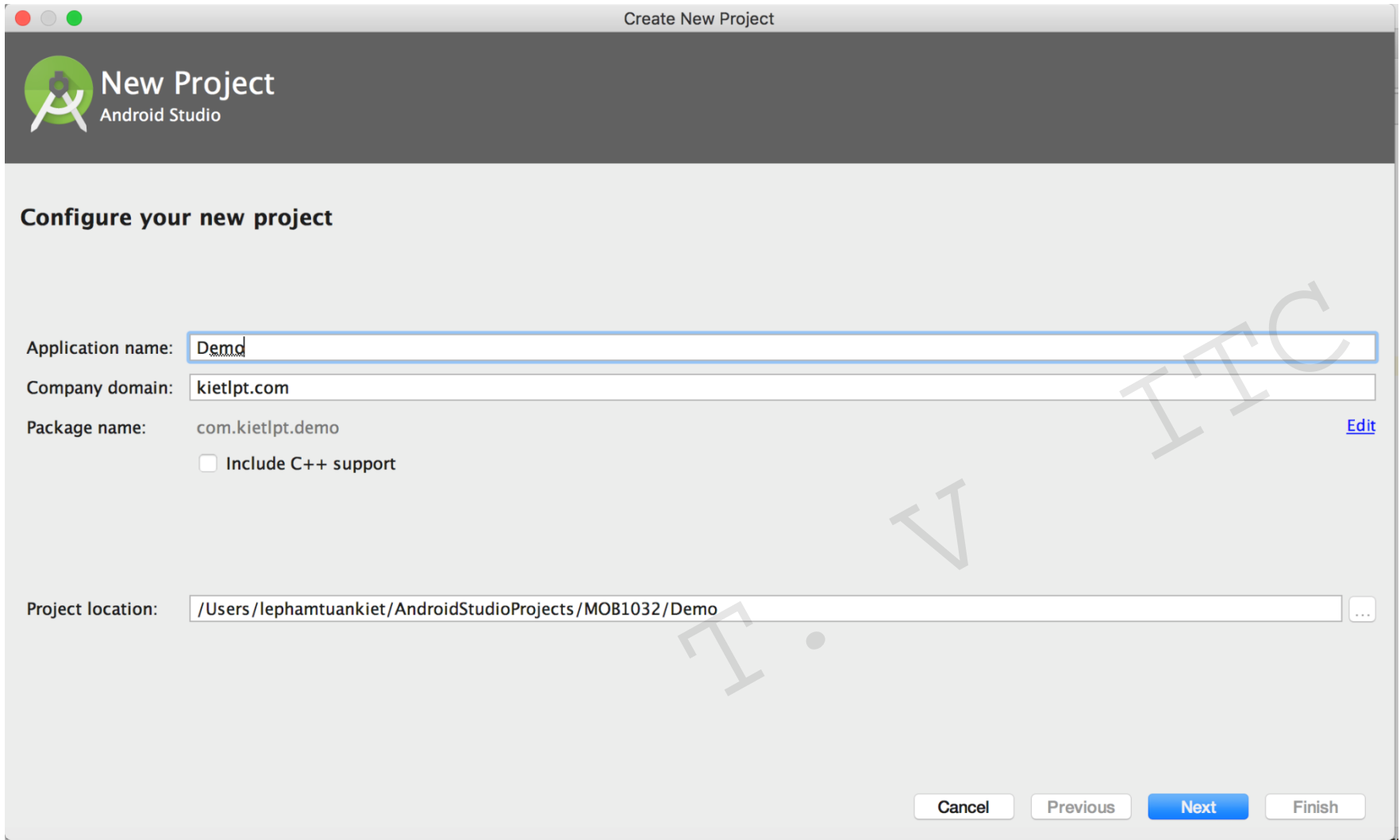


PHẦN II : SỬ DỤNG ANDROID STUDIO

T. V. ITC

TẠO PROJECT ANDROID TRÊN ANDROID STUDIO

- Khởi động Android Studio
- Chọn File > New > New Project



Android Studio window titled "Create New Project".

New Project
Android Studio

Configure your new project

Application name:

Company domain:

Package name: [Edit](#)

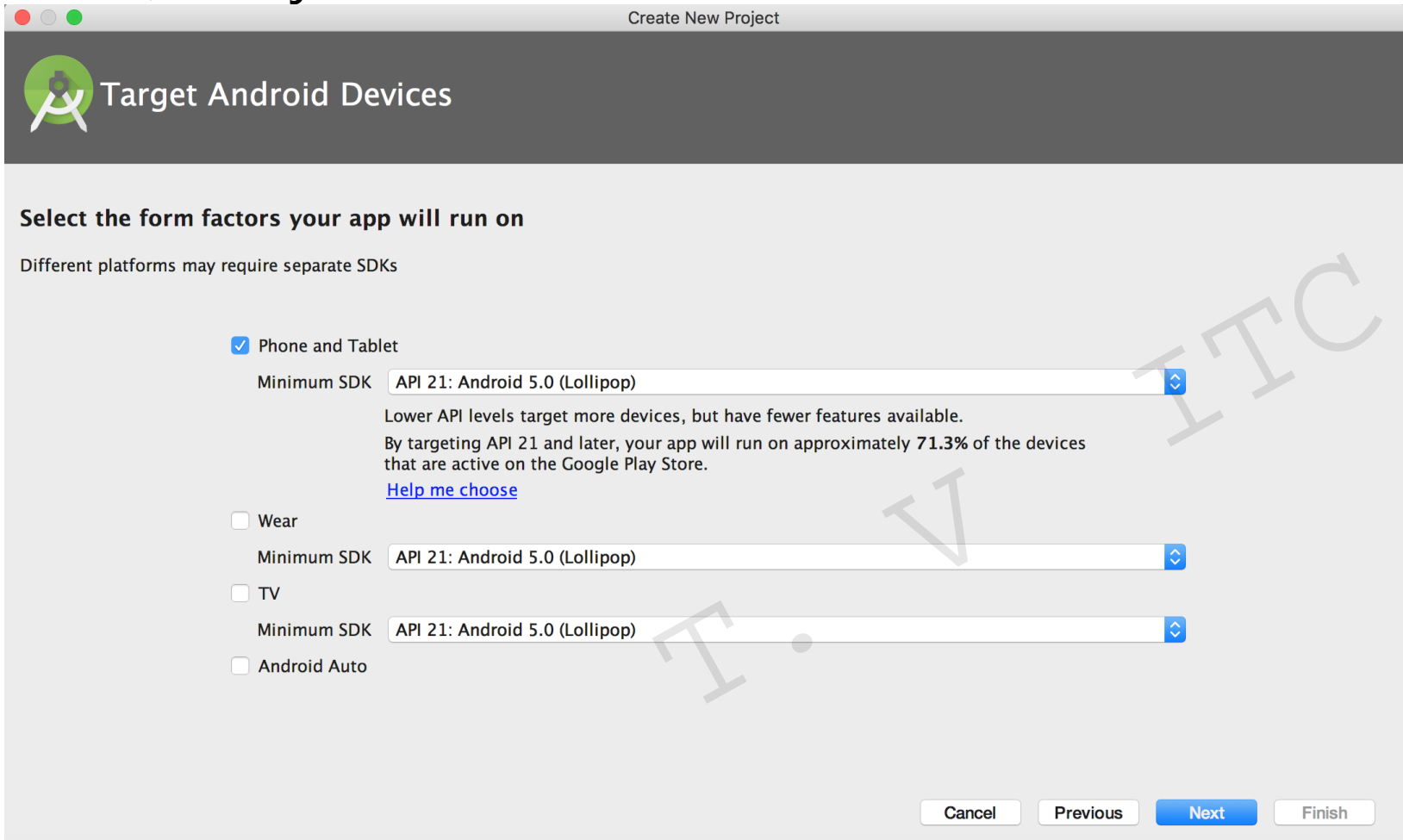
Include C++ support

Project location: ...

Buttons: Cancel, Previous, **Next**, Finish

TẠO PROJECT ANDROID TRÊN ANDROID STUDIO

- Nhập Application Name
- Nhập Company domain => Package name
- Chọn Project location, Next



The screenshot shows the 'Create New Project' dialog in Android Studio. The title bar reads 'Create New Project'. Below the title bar is a dark header with the Android logo and the text 'Target Android Devices'. The main content area is titled 'Select the form factors your app will run on' and includes the subtitle 'Different platforms may require separate SDKs'. There are four radio button options for target devices: 'Phone and Tablet' (checked), 'Wear', 'TV', and 'Android Auto'. Each option has a 'Minimum SDK' dropdown menu set to 'API 21: Android 5.0 (Lollipop)'. Below the 'Phone and Tablet' option, there is explanatory text: 'Lower API levels target more devices, but have fewer features available. By targeting API 21 and later, your app will run on approximately 71.3% of the devices that are active on the Google Play Store.' and a link 'Help me choose'. At the bottom right, there are four buttons: 'Cancel', 'Previous', 'Next', and 'Finish'.

Create New Project

Target Android Devices

Select the form factors your app will run on

Different platforms may require separate SDKs

Phone and Tablet

Minimum SDK API 21: Android 5.0 (Lollipop)

Lower API levels target more devices, but have fewer features available.
By targeting API 21 and later, your app will run on approximately 71.3% of the devices that are active on the Google Play Store.
[Help me choose](#)

Wear

Minimum SDK API 21: Android 5.0 (Lollipop)

TV

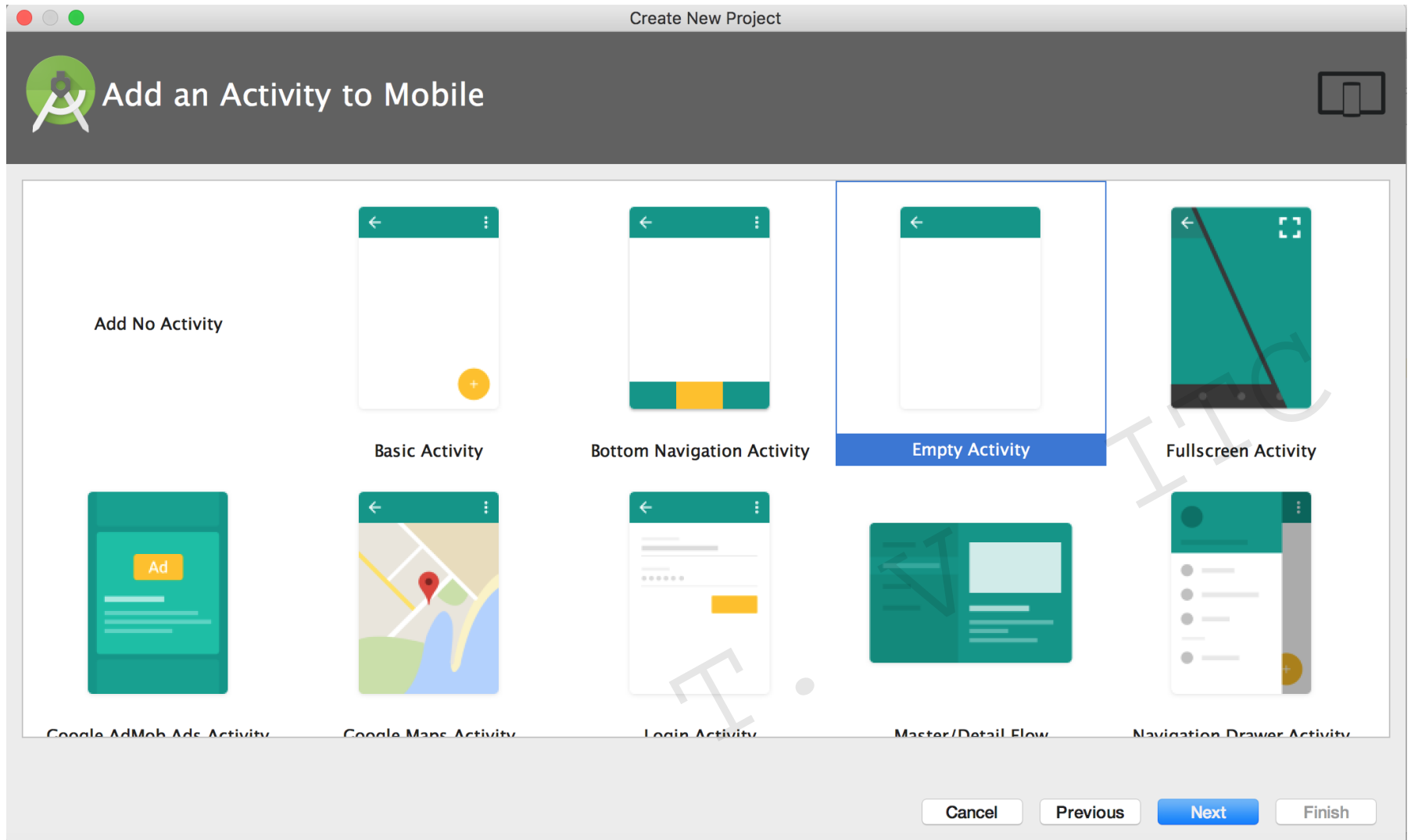
Minimum SDK API 21: Android 5.0 (Lollipop)

Android Auto

Cancel Previous Next Finish

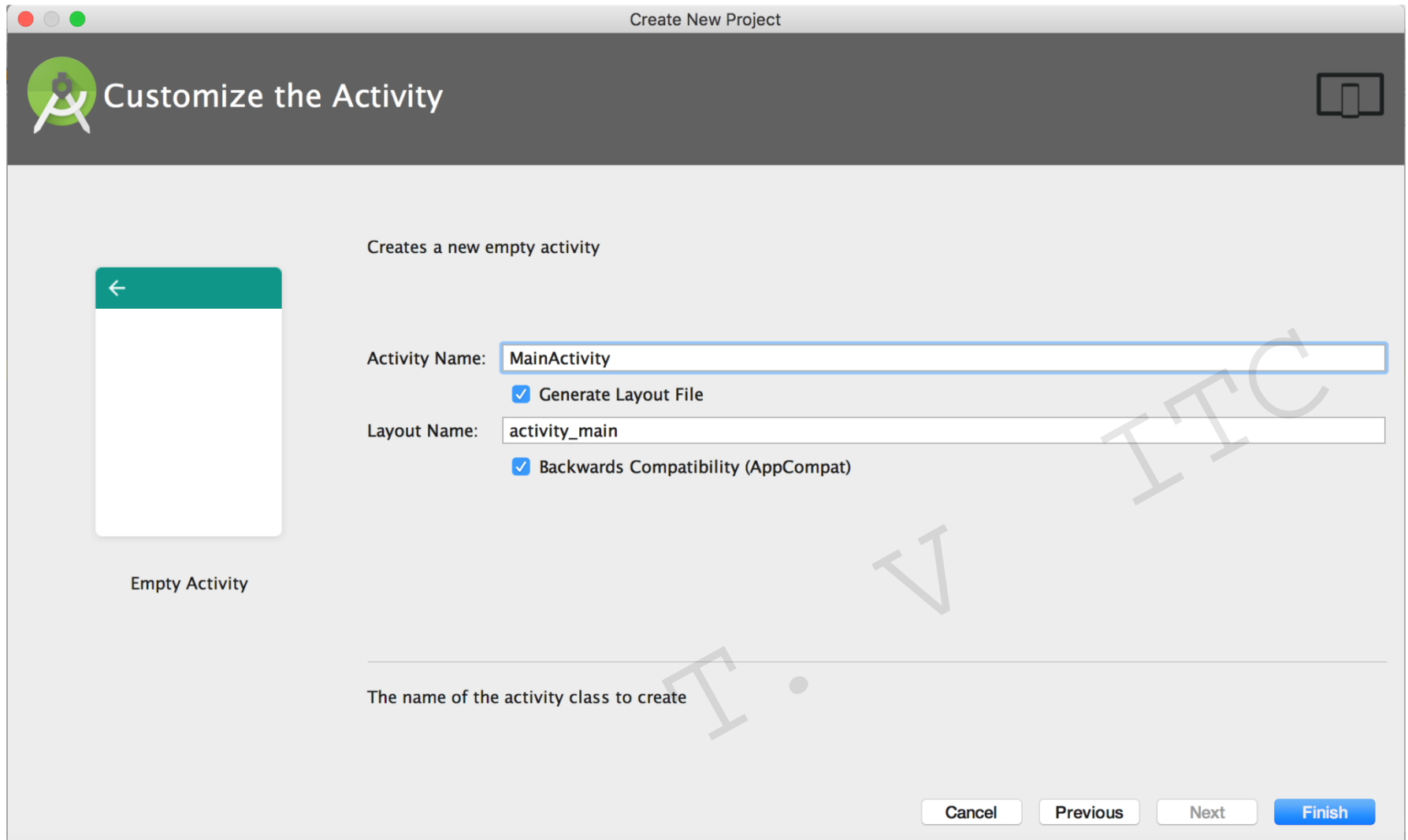
TẠO PROJECT ANDROID TRÊN ANDROID STUDIO

- Chọn Min SDK, Next

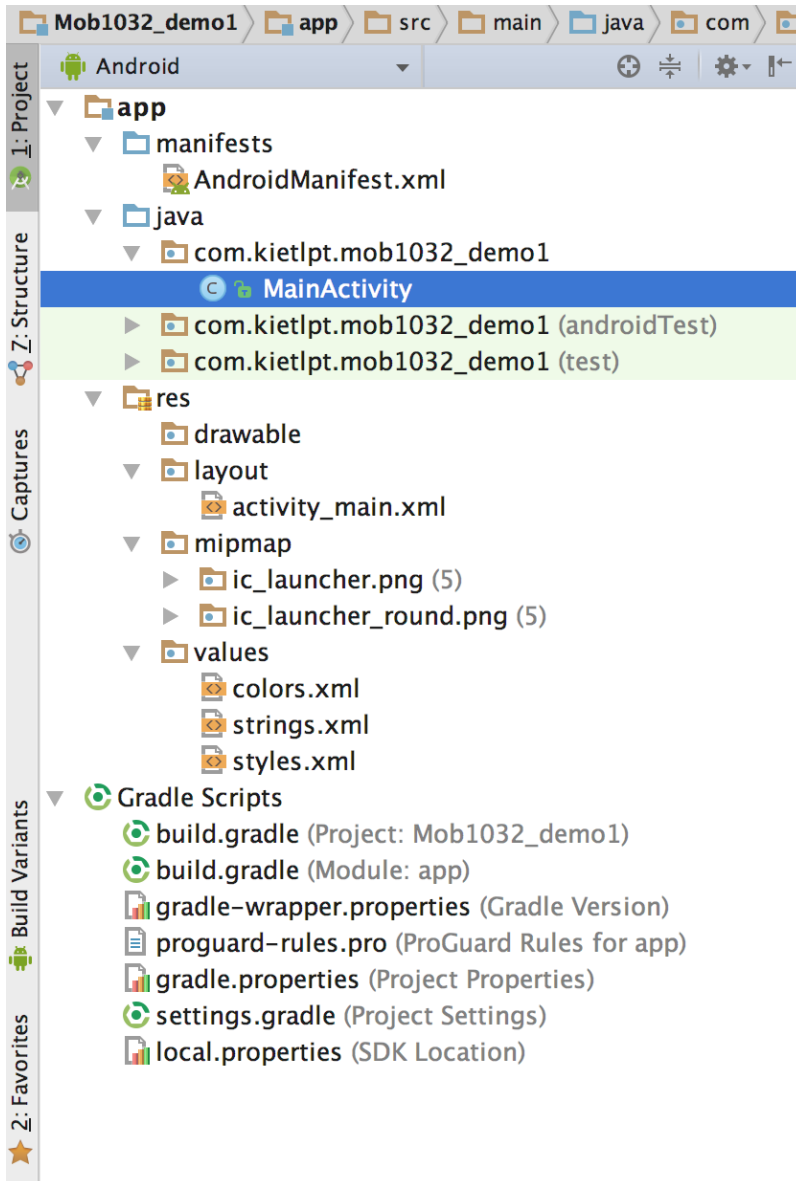


TẠO PROJECT ANDROID TRÊN ANDROID STUDIO

- Chọn tên Activity , Finish

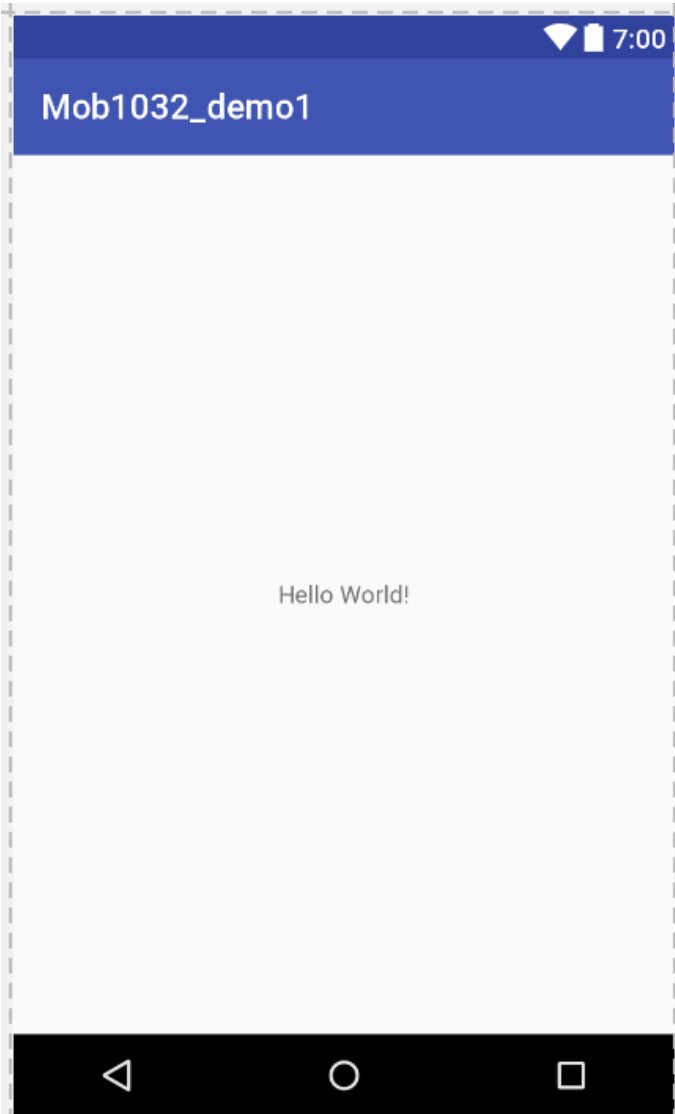


CẤU TRÚC PROJECT ANDROID TRÊN ANDROID STUDIO



- Manifests: thông tin project
- Java: code chính project
- Res: tài nguyên
 - Drawable
 - Layout
 - Mipmap
 - Values:
- Gradle Scripts:

CẤU TRÚC PROJECT ANDROID TRÊN ANDROID STUDIO



Giao diện activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.kietlpt.mob1032_demo1.MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</android.support.constraint.ConstraintLayout>
```

CẤU TRÚC PROJECT ANDROID TRÊN ANDROID STUDIO

Code java MainActivity.java

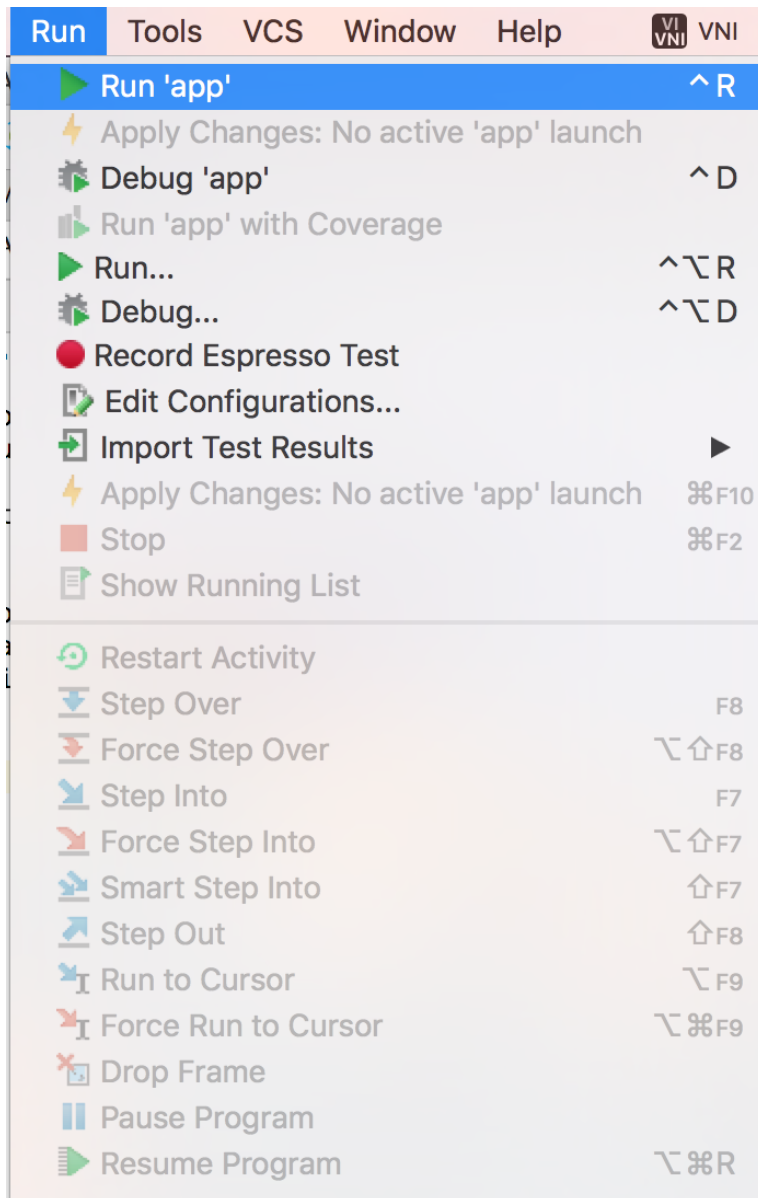
```
package com.kietlpt.mob1032_demo1;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

BIÊN DỊCH VÀ GỠ LỖI ỨNG DỤNG



Android Studio cung cấp khả năng biên dịch và gỡ lỗi ứng dụng rất tốt:

- Run 'app': biên dịch và chạy ứng dụng trên máy ảo hoặc thiết bị android kết nối với máy tính
- Debug: gỡ lỗi ứng dụng.

BIÊN DỊCH VÀ GỠ LỖ ỨNG DỤNG

- Trong tiến trình biên dịch, android project được biên dịch và đóng gói thành file có đuôi apk
- File chứa các thông tin cần thiết cho chạy chương trình:
 - Dex file
 - Binary version của Androidmanifest.xml
 - Compiled resource
 - Các file chưa compile

