P P SAVANI
UNIVERSITY

Mobile Application Development

Bannishikha Banerjee

What is Android?

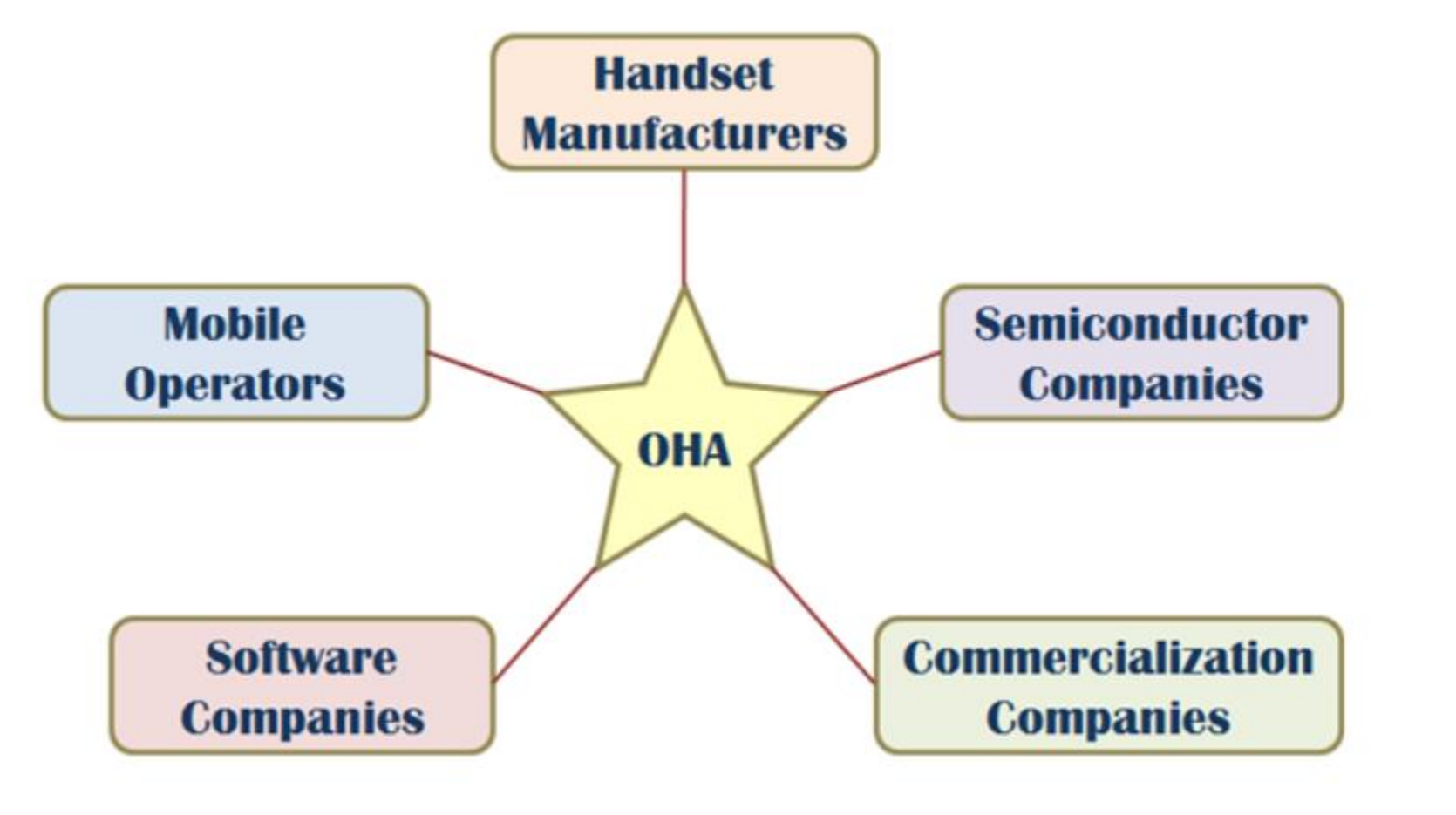
- Android is a Linux based mobile operating system
- Initially Android was available for mobile devices only.
- Later on android is also available for tablets, wear, big screen(TV), automobile etc.
- Android was unveiled during 2007 along with the founding of the Open Handset Alliance

What is “Open Handset Alliance”?

P P SAVANI UNIVERSITY

- Open Handset Alliance(OHA) was formed in November 2007.
- The OHA is the group that is in charge of the Android Smartphone Operating System. It was created by Google.
- The OHA is a business alliance that consists of 47 companies for developing open standard platform for mobile devices.
- The members of OHA includes handset manufactures, chip makers, commercialization companies, software companies and mobile operators.

OHA


Handset Manufacturers
Mobile Operators
OHA
Semiconductor Companies
Software Companies
Commercialization Companies

Features of Android

P P SAVANI UNIVERSITY

1. **Open Source:** the original source code of android is available for modifications.
2. Android OS is device independent.
3. Android OS supports NFC(Near Field Communication)
   - Transfer any amount of data by simply touching the two devices.
   - Maintain less than 1 cm distance in between two devices.
4. Android OS supports IPC(Inter Process Communication)
5. By default Android OS is available with SQLite.
   - This database is open source & supports RDBMS

Features of Android

P P SAVANI UNIVERSITY

6. Android OS supports for SSL(Secure Socket Layer)
   - TLS(Transport Layer Security)
   - In SSL client & server interaction done in encrypted mode.
7. Android OS will support Text to Speech and Speech to Text conversion.
8. Android device can be controlled through voice commands.
9. Android supports all types of images, audio, video, different types of languages, different types of fonts.

Features of Android

P P SAVANI UNIVERSITY

10. Android OS supports basic graphics and 2-D,3-D animation.
11. Supports Video calling.
12. External storage
    - Most Android devices include microSD slot and can read microSD cards formatted with FAT32,Ext3 or Ext4 file system.

Developer

P P SAVANI UNIVERSITY

- Android OS is designed and developed by Andi Rubin.
- Handover to the Google in Corporation in the year of 2007.

Android Versions

P P SAVANI UNIVERSITY

Version	Nickname	Release Date
1.0	Android	September 23,2008
1.1	Beta Android	February 9,2009
1.5	Cupcake	April 27,2009
1.6	Donut	September 15,2009
Android 1.x Mobiles only		
Android 2.0/2.1	Éclair	October 26,2009
2.2 – 2.2.3	Froyo	May 20,2009
2.3 – 2.3.7	Gingerbred	December 6,2010
Android 2.x is also designed for mobiles but from 2.x android started supporting API. - By using Google API android application can interact with Google products such as Gmail, YouTube, google maps, navigation, google search engine, google Clouds		

Android Versions

P P SAVANI UNIVERSITY

Version	Nickname	Release Date
Android 3.0 – 3.2.6	Honey Comb	February 22,2011
Android 3.x is specially designed for tablets. Started supporting Fragments.		
Android 4.0 – 4.0.4	Ice-cream Sandwich	October 18,2011
4.1 – 4.3.1	Jellybean	July 9,2012
4.4 – 4.4.4	KitKat	October 31,2013
4.4.w supports for wearable devices like wrist-watch. From 4.x android started supporting mobiles & tablets application. It means single app can run in mobiles & tablets		

Android Versions

P P SAVANI UNIVERSITY

Version	Nickname	Release Date
Version	Nickname	Release Date
5.0 – 5.1.1	Lollypop	November 12,2014
Android 5.x designed for Big Screens ie. TV		
Android 6.0 – 6.0.1	Marshmallow	October 5,2015
Android 6.0 designed for Automobiles(speed of car, km)		
Android 7.0 – 7.1.2	Nougat	August 22,2016
8.0 – 8.1	Oreo	August 21,2017
9.0	Pie	August 6,2018
10.0		September 3,2019
11.0		September 8,2020

ANDROID Version History

Android 1.0 to 1.1(September 23, 2008)

Features:

- Google Maps.
- Camera.
- Gmail, Contacts, and Google Synchronization.
- Web Browser.
- Wireless supports – Wi-Fi and Bluetooth.

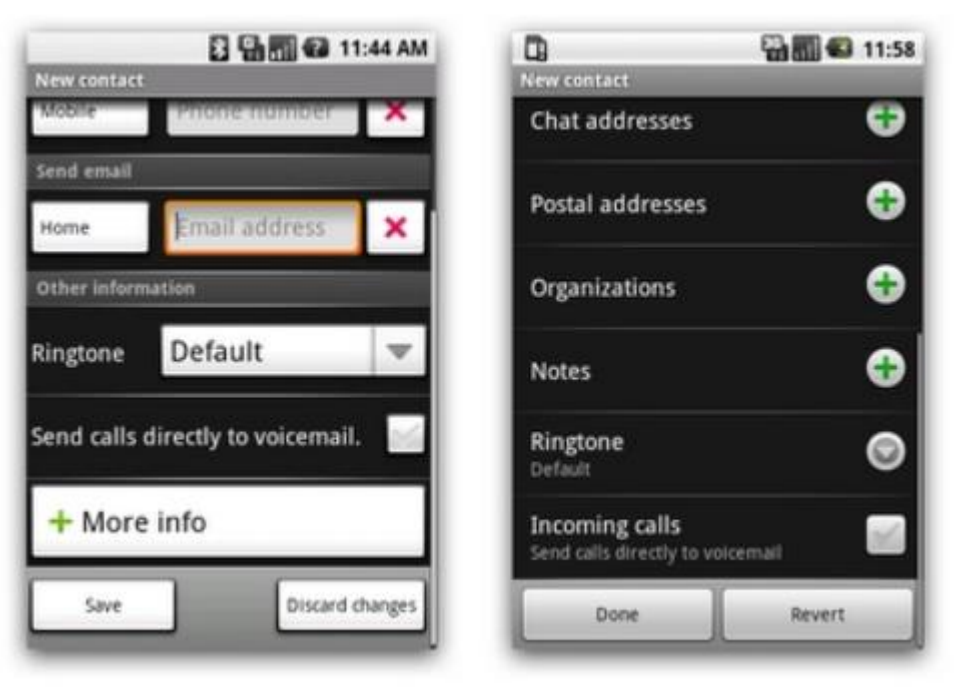
Beta 1.1(February 9, 2009)

P P SAVANI UNIVERSITY

- released on February 9, 2009.
- Features:
  - Add Save attachment in the message.
  - Provides reviews and details when the user search business on maps.

Android version 1.5: Cupcake

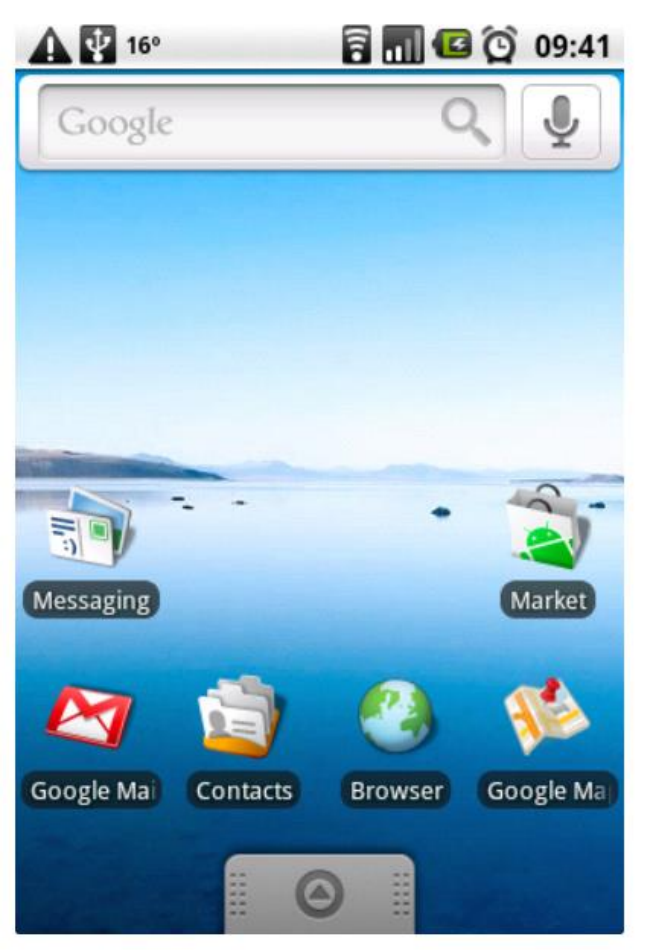
April 30, 2009



Features:

- New upload service on YouTube and Picasa like Uploading Videos and Photos.
- Supporting in MPEG-4, Video recording.
- Improving Web Browser-Copy and Paste facility.

Android version 1.6: Donut
September 15, 2009.



Features:

- The main enhancement was a Power Control widget for managing Wi-Fi, Bluetooth, GPS, etc.
- Provided Gallery and Camera features with quick toggling features.
- Improve the speed in system apps.
- Introduction of the Quick Search Box.

Android versions 2.0 to 2.1: Eclair

December 3, 2009.

Features:

- Update UI.
- Support Live Wallpaper.
- Support Bluetooth 2.1.
- Improve Google map.
- Minor API Changes.

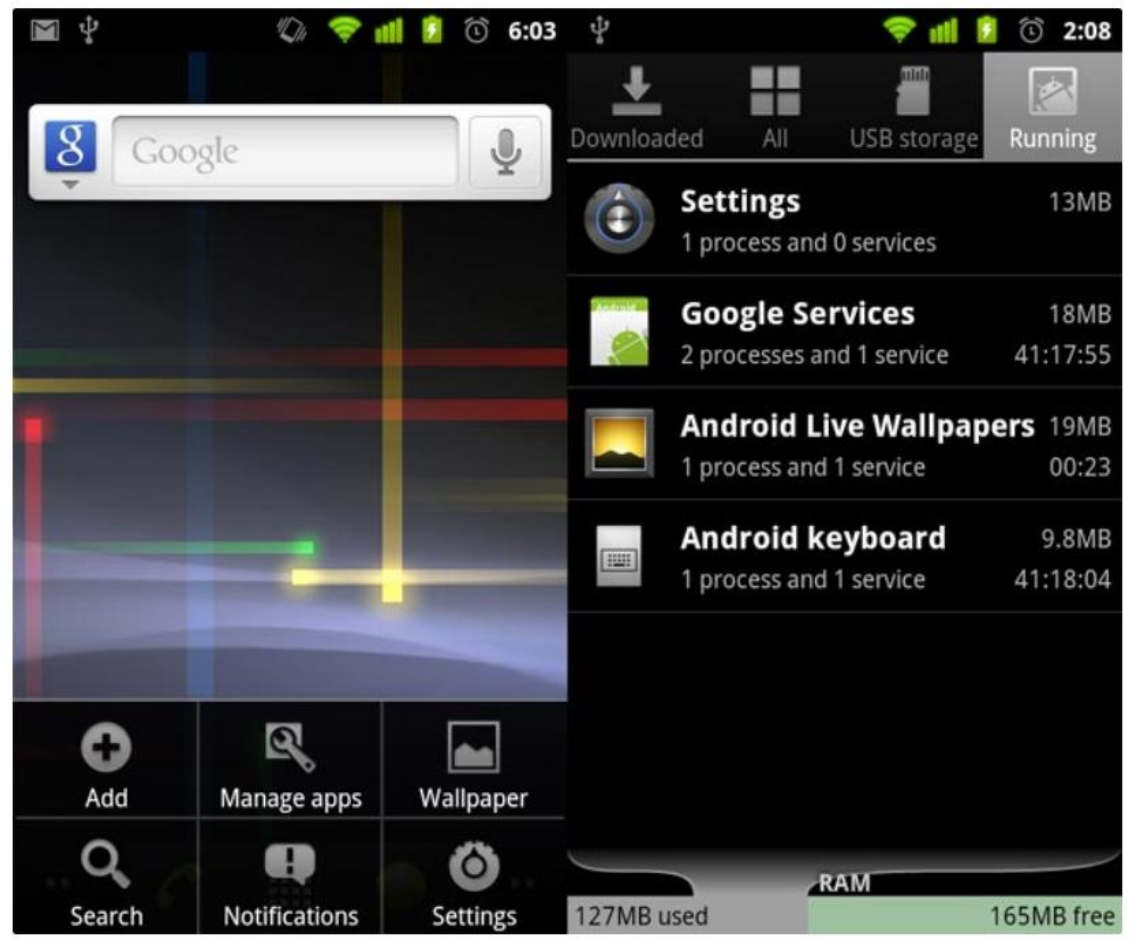
Android version 2.2: Froyo

May 20, 2010.

Features:

- Support for Animated GIF.
- Wi-Fi Support Hotspot functionality.
- Speed improvements.
- Upload file support in the browser.
- Support numeric and alphanumeric passwords.

Android version 2.3: Gingerbread
December 6, 2010

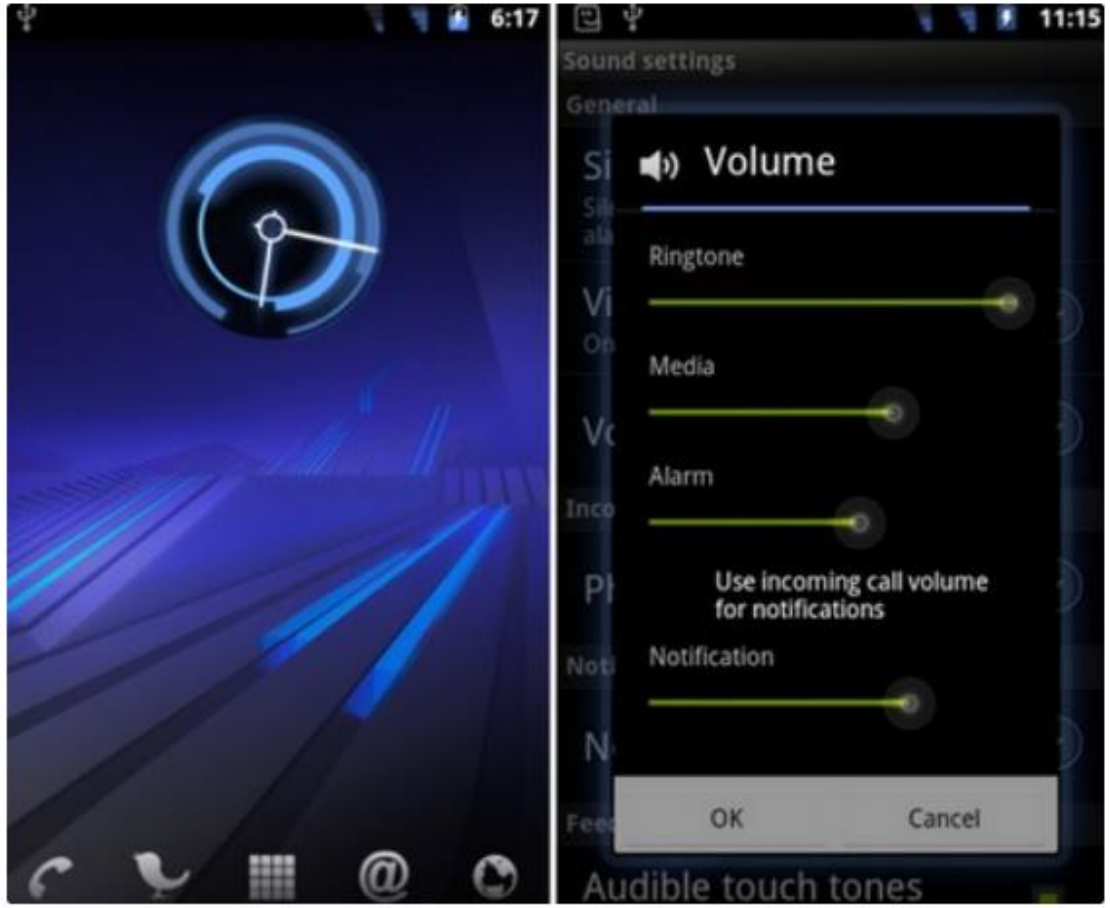


Features:

- Improve Copy-Paste Facility.
- Updated UI design.
- VP8 and WebM video format support.
- Social Networking Supports.
- Easy use of the keyboard.
- Multiple camera support (usually known as a selfie camera nowadays).

Android 3.0/3.1/3.2 – Honeycomb

P P SAVANI UNIVERSITY



Features:

- Gmail App improvements.
- Updated 3D UI.
- Supports multiprocessors and hardware acceleration for graphics.
- Media Sync from SD Card.
- Google eBooks.
- Google Talk Video Chat.
- Support Adobe Flash in Browser.
- High-performance Wi-Fi Connections and Lock.
- Chinese handwriting.

Android version 4.0: Ice Cream Sandwich

October 19, 2011

Features:

- Improved text input and spelling check.
- Wi-Fi direct (Sharing information using NFC).
- Photo Decor facility.
- Improved keyboard correction.
- Unlocking with face-fixing.
- Improved video recording resolution.
- Camera performance.
- Up to 16 tabs in the web browser.

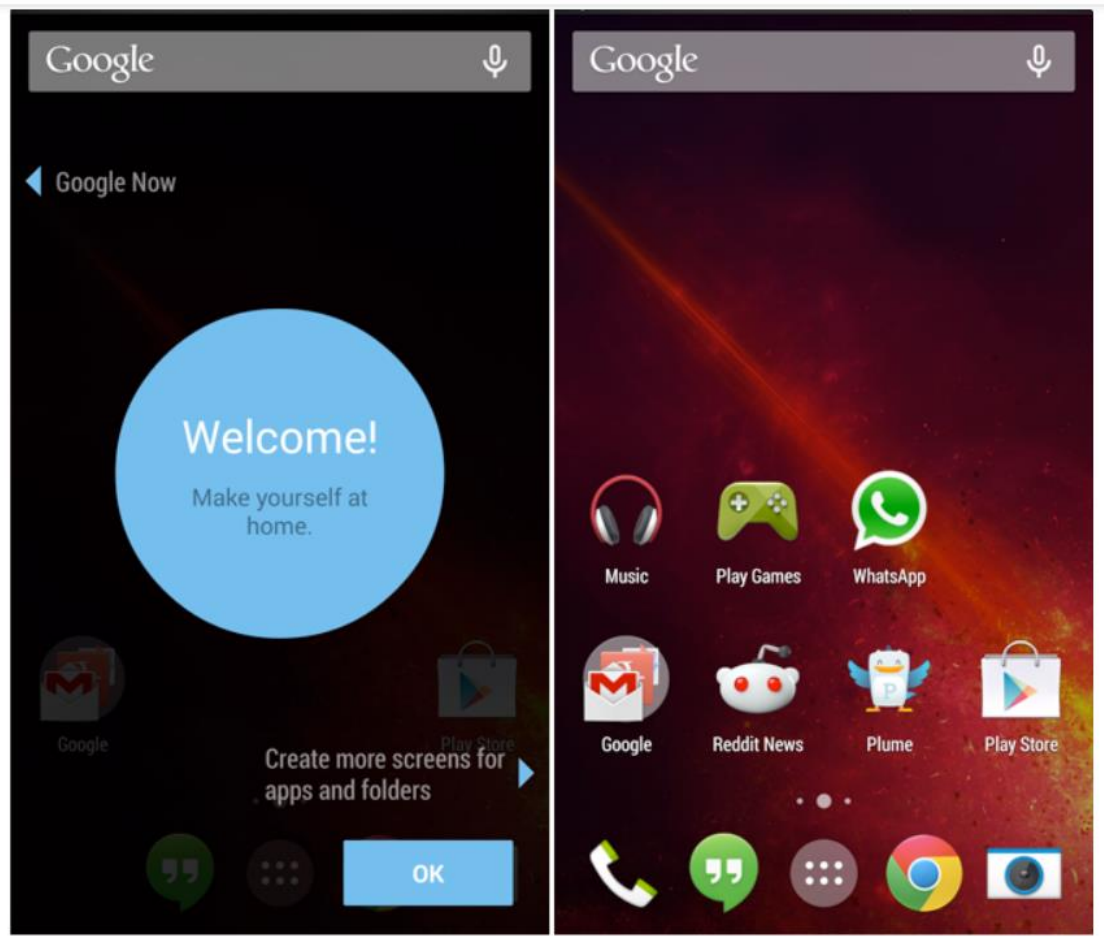
Android versions 4.1 to 4.3: Jelly Bean

Features:

- Voice search.
- Panorama.
- Daydream as a screensaver.
- Power control.
- Improve camera application.
- Security enhancement.
- Voice typing.
- Multiple user accounts on tablets only.
- 4k resolution support.
- Supporting Bluetooth Low Energy.
- Bi-directional text and other language support.
- Support USB audio.
- Set the volume of incoming calls and show a message alert.
- Native emoji support.

Android version 4.4: KitKat

September 3, 2013.



Features:

- Screen Recording.
- KitKat adds a feature in 'Google now'. Its name is 'OK Google'. "OK, Google" allows access to Google to the users without touching your mobile phone.
- GPS Support.
- Offline music support.
- UI updates for google map navigation and alarm.
- Introduction of 'Emoji' to the google keyboard.

Android versions 5.0 and 5.1: Lollipop

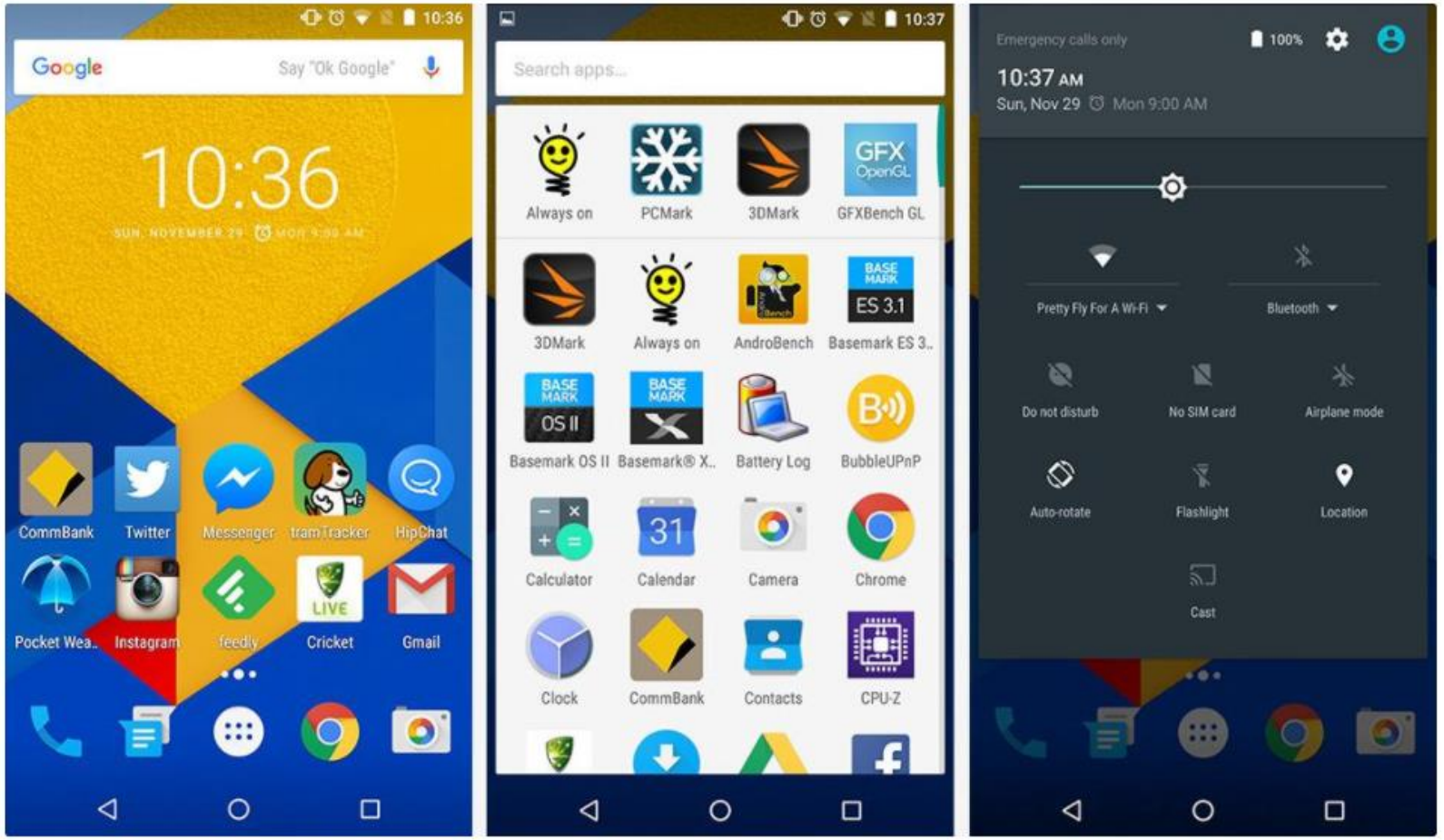
November 12, 2014

Features:

- Support ART(Android Runtime).
- Improvement in UI.
- New material design.
- Notifications on the Lock screen.
- Revamped navigation bar.
- Multiple sim card support.
- High definition voice call.

Android version 6.0: Marshmallow

P P SAVANI UNIVERSITY

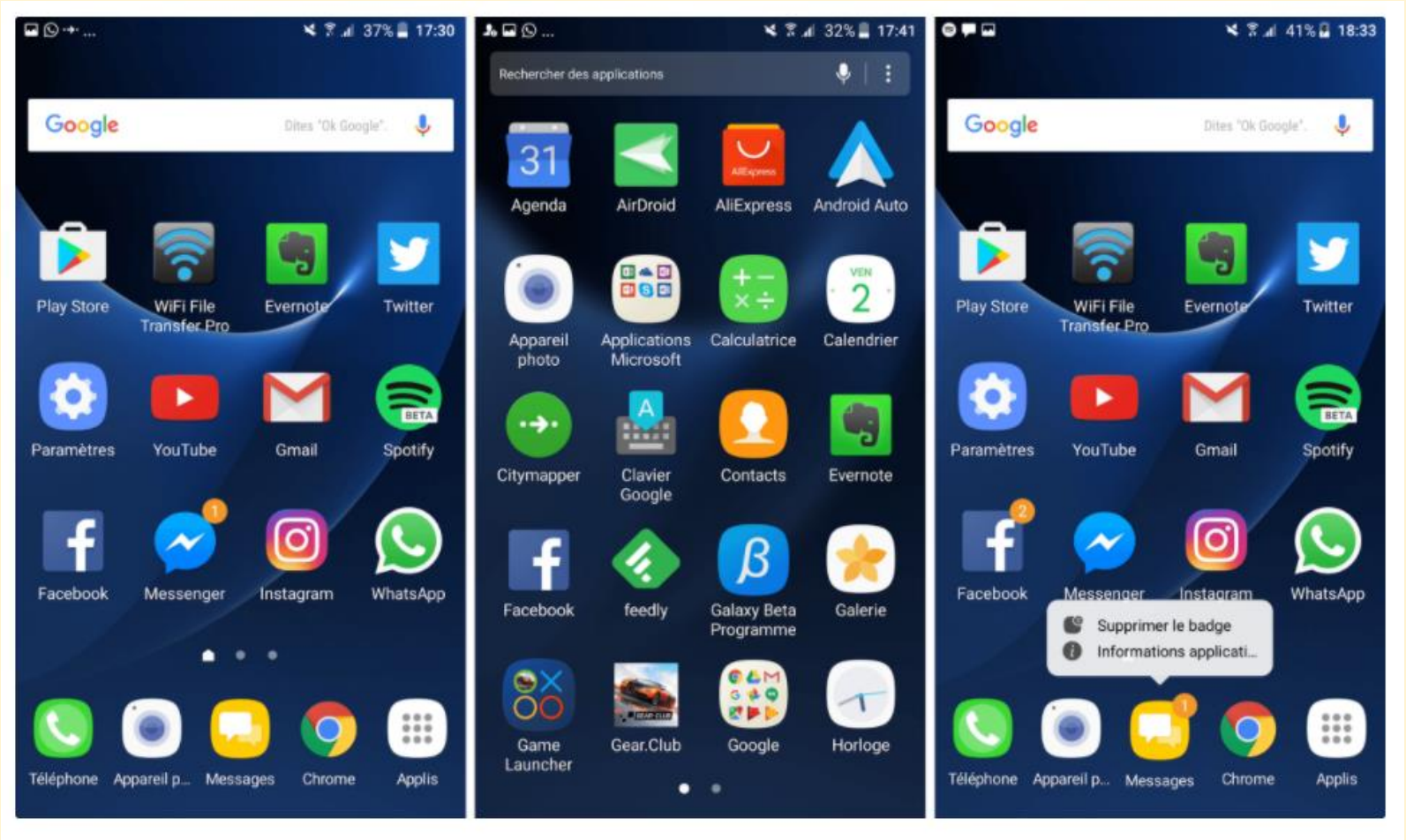


Features:

- **Fingerprint authentication to unlock the screen.**
- **USB Type C support.**
- **Multi-window experiments (user can use two different apps in one screen).**
- **Save battery-'Sleep Mode'.**
- **App permission model-OPT(send a request for permission).**

Android versions 7.0 and 7.1: Nougat

March 2016

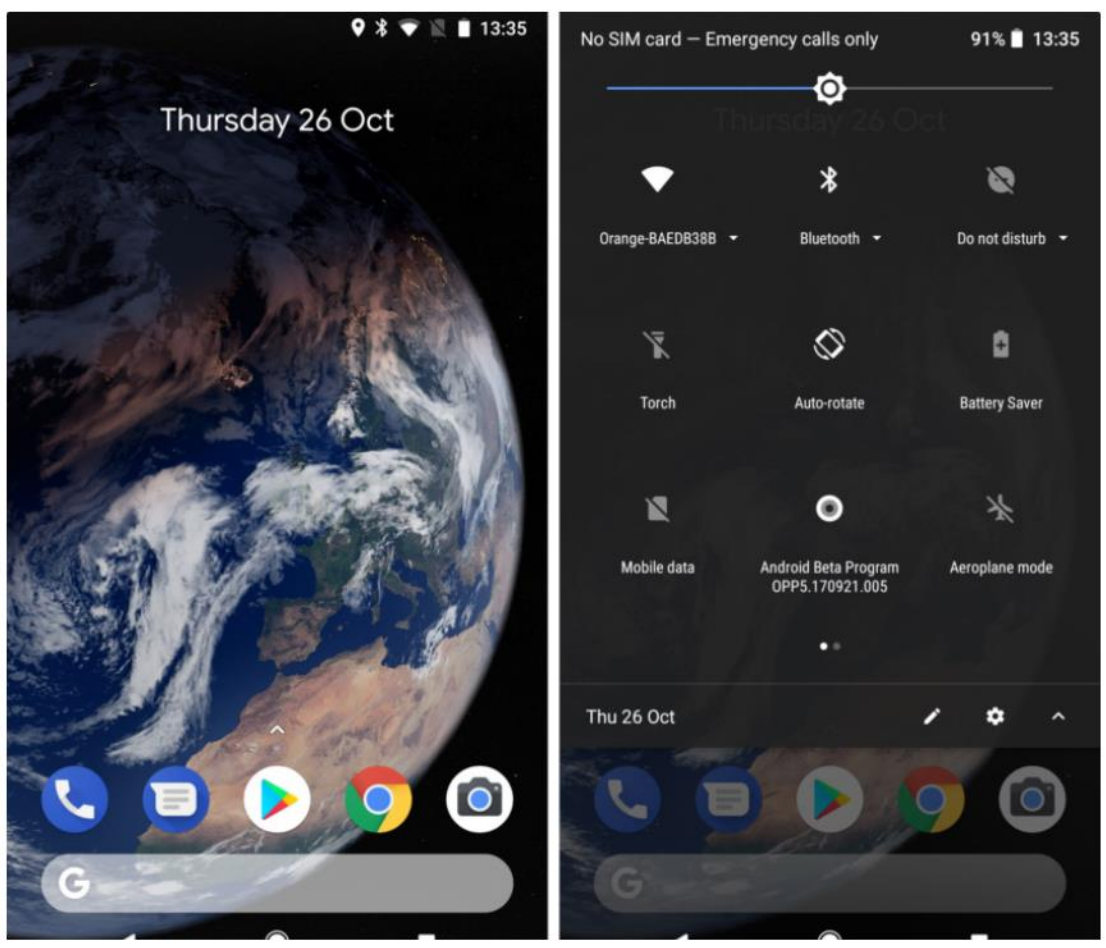


Features:

- Provide multitasking.
- Inline reply to messages and notifications so you won't have to open up your Messenger application for quick replies.
- Providing multi-window mode.
- Improvements in storage manager.
- Display touch improvement.

Android version 8.0 and 8.1: Oreo

P P SAVANI UNIVERSITY

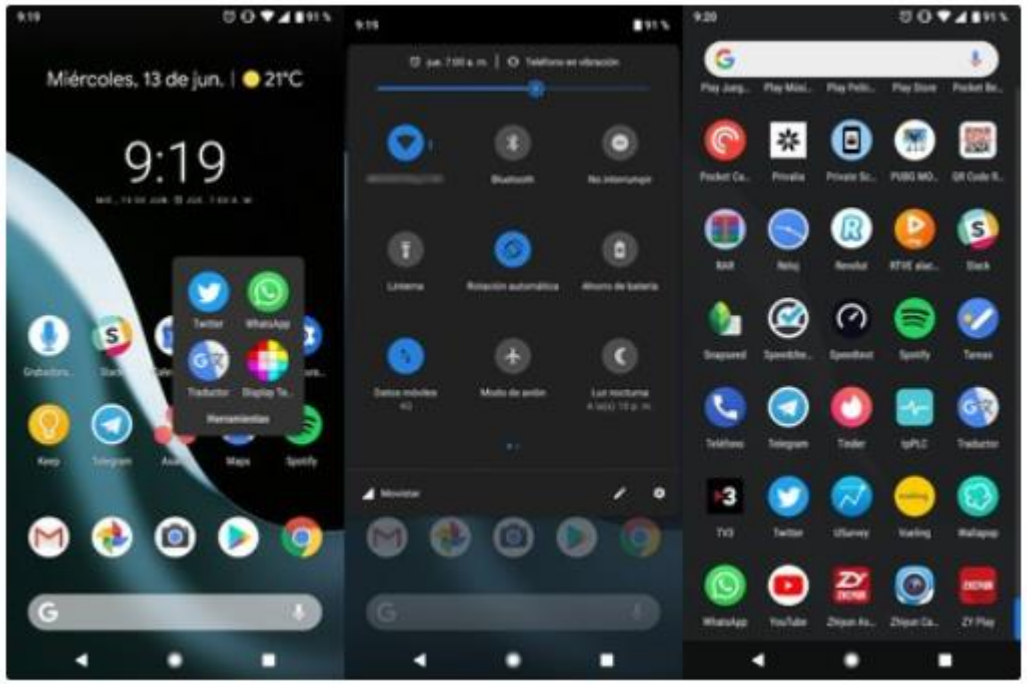


Features:

- Support PIP(Picture-in-Picture).
- Multi-display support.
- Google Play support.
- Adaptive icons.
- Revamped notification section(Users can set which notifications you want to show).

Android version 9: Pie

August 2018.



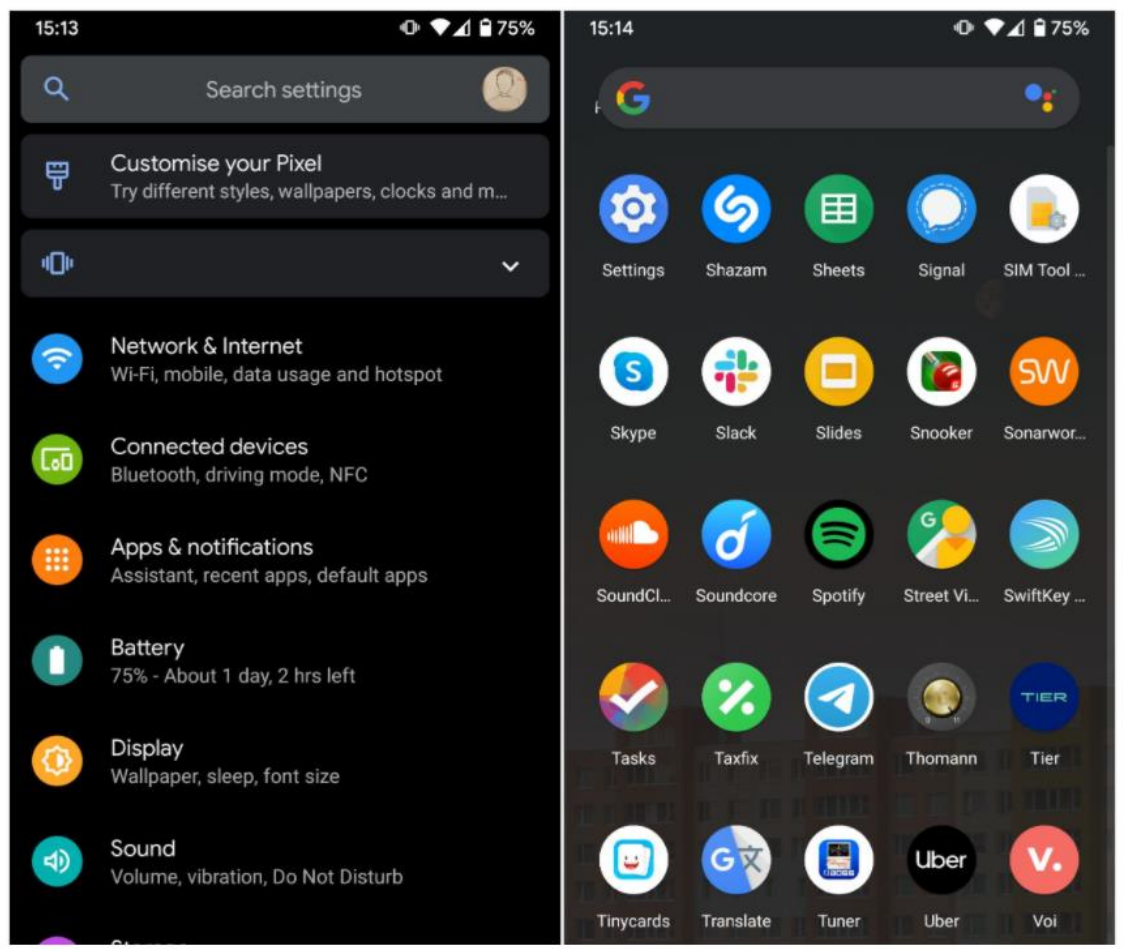
Features:

- New Gesture Navigation.
- Artificial intelligence (AI) Compatible.
- Adaptive Battery and Brightness.
- App Actions.
- New Screenshot Shortcuts.
- Easier Screen Rotation.
- Volume and Sound Improvement.
- Selectable Dark Mode.
- Slices.
- Improved Security Features.
- Digital Wellbeing.
- New Accessibility Menu.
- Easier Text Selection.
- More Notification Information.

Android version 10: Android Q

September 3, 2019

P P SAVANI UNIVERSITY



Features:

- Support for the upcoming foldable smartphones with flexible displays which is an upcoming rush.
- System-wide dark mode.
- Navigation control over gesture.
- Smart reply for all messaging apps.
- Support for Live caption.
- Better notification control.

Android version 11 (Developer Preview) preview on February 19 of year 2021.

- **Features (announced):**
  - **New Support for 5G.**
  - **Privacy and Security; A new privacy choice for apps is the "Only This Time" option when you're allowing the app access to your location, microphone, or camera.**
  - **Support new screen types (pinhole and waterfall).**
  - **Low Latency Options; adds low latency support in new MediaCodec APIs and HDMI which is very useful for use on external displays and TVs.**

Android Architecture

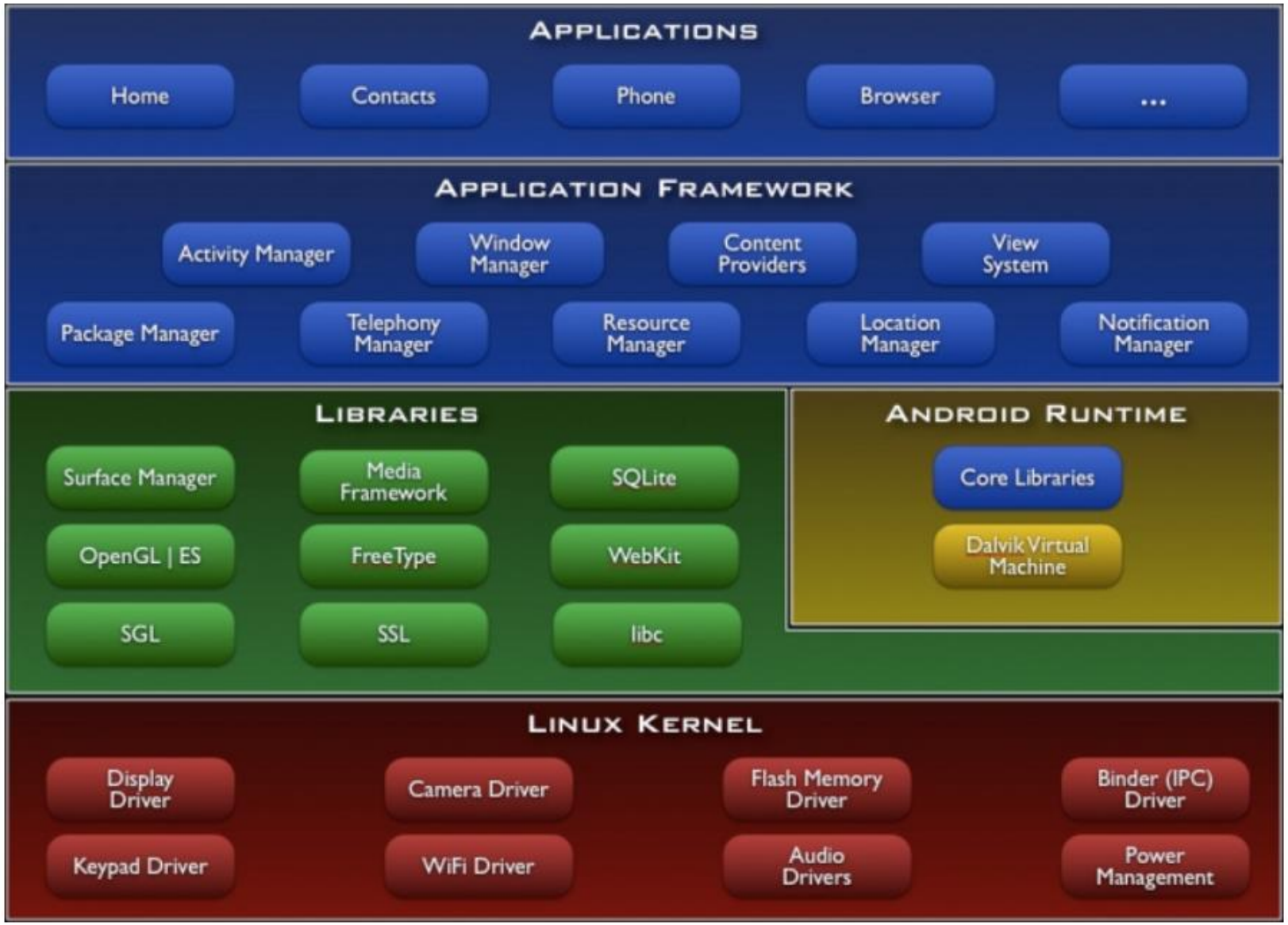
P P SAVANI UNIVERSITY

- Android OS is a software stack of different layers, where each layer is a group of several program components.
- Android has the following layers:

1. Applications
2. Application Framework
3. Libraries
4. Android Runtime
5. Linux Kernel

Android Architecture

P P SAVANI UNIVERSITY



Applications

- Written using Java Language
  - Email Client
  - SMS Program
  - Maps
  - Browser
  - Calendar
  - Contacts
- Supports Parallel running
- No compulsory applications

Application Framework

- The Application Framework layer provides many higher-level services to applications in the form of Java classes.
- Application developers are allowed to make use of these services in their applications.

Application Framework

P P SAVANI UNIVERSITY

- The Android framework includes the following key services –
- Activity Manager – Controls all aspects of the application lifecycle and activity stack.
- Content Providers – Allows applications to publish and share data with other applications.
- Resource Manager – Provides access to non-code embedded resources such as strings, color settings and user interface layouts.
- Notifications Manager – Allows applications to display alerts and notifications to the user.
- View System – An extensible set of views used to create application user interfaces.

Libraries

- Useful to develop any third-party application.
- Native libraries are written in a language that compiles to native code for the platform it run.

Libraries

P P SAVANI UNIVERSITY

1. SQLite - Responsible for Database Operation
2. Free Type - Font Support
3. Media F/W - Audio,Video Format
4. Open GL(Open Google Library) - 2D,3D Graphics Support
5. SSL - Encrypted Communication between Client and Server
6. SGL (Scalable Graphics Libraries)- For Basic Graphics Support

Android Runtime

- Android Runtime consists of Core Libraries and Dalvik Virtual Machine.
- Core Libraries are written in C/C++ languages. This libraries are helpful for runtime environment. Some of the core libraries are Data Structure, File Access, Network Access, Utilities and Graphics.

DVM(Dalvik Virtual Machine)

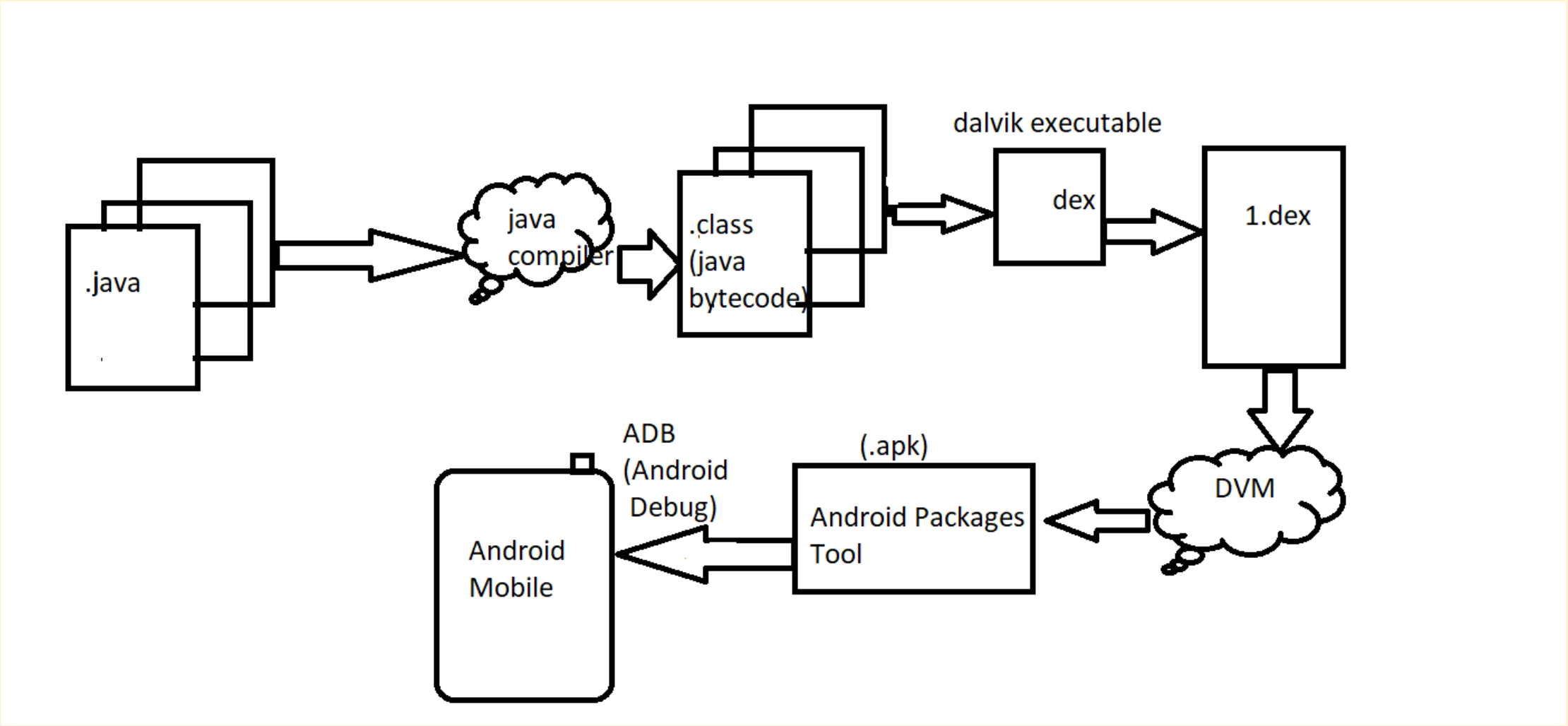
P P SAVANI UNIVERSITY

- Runtime environment for running android application.
- JVM is used to run high-end applications while DVM is used for small-end applications.
- DVM was first written by "Dan Bornstein“
- Unlike JVM, the DVM does not run .class files but it runs .dex files.
- .dex files are built from .class file at the time of compilation and provide higher efficiency in row resource environments.

Linux Kernel

- Linux Kernel is a Root layer of android Architecture which is responsible for device drivers(display, camera, Bluetooth, flash, web driver, usb, keypad,wifi etc.)
- power management, memory management and resource management.

How Android Works?



Exploring Android SDK

P P SAVANI UNIVERSITY

- Android SDK is a software development kit developed by Google for the Android platform.
- Android SDK comes bundled with Android Studio, Google's official integrated development environment (IDE) for the Android operating system

What Is the Android SDK?

P P SAVANI UNIVERSITY

- The Android SDK is a collection of software development tools and libraries required to develop Android applications.
- Every time Google releases a new version of Android or an update, a corresponding SDK is also released which developers must download and install.
- The Android SDK comprises all the tools necessary to code programs from scratch and even test them. These tools provide a smooth flow of the development process from developing and debugging, through to packaging.
- The Android SDK is compatible with Windows, macOS, and Linux, so you can develop on any of those platforms.

Components of the Android SDK

P P SAVANI UNIVERSITY

- The Android SDK consists of an emulator, development tools, sample projects with source code, Google API, and the required libraries to build Android applications

Android SDK Platform

P P SAVANI UNIVERSITY

- **Provided in the android.jar file that made up of several important packages.**

Top-Level Package Name	Description
android.*	Android application fundamentals
dalvik.*	Dalvik Virtual machine support fundamentals
java.*	Core java classes and generic utilities for networking, security, math and so on
javax.*	Encryption support
junit.*	Unit-Testing support
org.apache.http.*	HTTP Protocol support
org.json	JavaScript Object Notation Support

Few Popular Third-Party Android APIs

- Com.google.android.gms.ads.* = Google Mobile Ads SDK Package.
- Com.google.android.gms.analytics.* = Google Analytics SDK for Android Package.
- Com.google.android.gms.gcm = Google Cloud Messaging for Android.
- Com.google.android.gms.appindexing = Google App Indexing Package.
- Com.google.android.gms.appinvite = Google App Invites Package
- Com.google.android.gms.games = Google Play Games Services Package
- Com.google.android.gms.fitness = Google Fit Package

Android SDK Tools

P P SAVANI UNIVERSITY

- **Android SDK provides many tools to design, develop, debug and deploy your applications.**
  - **Android Studio**
  - **Android SDK and AVD Managers**
  - **Android Emulator**

Android Studio

P P SAVANI UNIVERSITY

- Launch the Android Virtual Device Manager.
- Launch the Android SDK Manager.
- Launch the Android Device Monitor

Android SDK and AVD Managers

P P SAVANI UNIVERSITY

- Different Android Devices run different version of the Android Operating System.
- Developers need to be able to target different Android SDK versions with their applications.
- The Android Virtual Device Manager organizes and provides tools to create and edit AVDs.

SDK Platform-Tools

P P SAVANI UNIVERSITY

- Android Platform-Tools are used to support the features for the current Android platform and are necessary for Android app development.
- Android Debug Bridge (adb): This is a handy command-line tool that lets you communicate with a device. The adb command allows you to perform device actions, such as installing and debugging apps. It also provides access to a Unix shell that you can use to run a variety of commands on a device.
- fastboot: This lets you flash a device with a new system image.
- systrace: This tool helps collect and inspect timing information across all processes running on your device at the system level. It's crucial for debugging app performance.

Android Emulator

P P SAVANI UNIVERSITY

- The Android Emulator is a device-emulation tool that simulates Android devices on your computer, allowing developers to test applications on different devices and Android API levels, without needing to have physical devices for each.
- The emulator comes with configurations for various Android phones, tablets, Wear OS, and Android TV devices.

Android Emulator

P P SAVANI UNIVERSITY

- The Android emulator provides almost all of the capabilities of a real Android device. You can perform the following activities:
- simulate phone calls and text messages
- simulate different network speeds
- specify the location of the device
- simulate hardware sensors such as rotation
- access Google Play Store and much more
- Often it is faster and easier to test your app with an emulator instead of using a physical device.

Emulators vs Simulators

Emulators	v/s	Simulators
An emulator is an application that emulates real mobile device software, hardware and operating systems, allowing us to test and debug our applications.		A simulator is a less complex application that simulates internal behavior of a device, but does not emulate hardware and does not work over the real operating system.
Emulator is usually provided by device manufacturer.		A simulator may be created by the device manufacturer or by some other company.
Emulators are written in machine-level assembly languages.		Simulators are written in high level languages.
Emulators are more suitable for debugging.		Simulators can be difficult for debugging purpose.
Often an emulator comes as a complete re-implementation of the original software .		Simulator is just a partial re-implementation of the original software .
e.g. - Android (SDK) Emulator 		e.g. - iOS Simulator 

Android Terminologies

P P SAVANI UNIVERSITY

- **XML**
  - In Android, XML is used for designing the application's UI like creating layouts, views, buttons, text fields etc. and also used in parsing data feeds from the internet.
- **View**
  - A view is an UI which occupies rectangular area on the screen to draw and handle user events.
- **Layout**
  - Layout is the parent of view. It arranges all the views in a proper manner on the screen.
- **Activity**
  - An activity can be referred as your **device's screen** which you see. User can place UI elements in any order in the created window of user's choice.
  - **android.app.Activity** class

Android Terminologies

P P SAVANI UNIVERSITY

- **Emulator**
  - An emulator is an Android virtual device through which you can select the target Android version or platform to run and test your developed application.
- **Manifest file**
  - Manifest file acts as a metadata for every application. This file contains all the essential information about the application like app icon, app name, launcher activity, and required permissions etc.

Android Terminologies

P P SAVANI UNIVERSITY

- **Service**
  - Service is an application component that can be used for long-running background processes. It is not bounded with any activity as there is no UI. Any other application component can start a service and this service will continue to run even when the user switches from one application to another.
  - **android.app.service** class
- **Broadcast Receiver**
  - Broadcast Receiver is another building block of Android application development which allows you to register for system and application events. It works in such a way that, when the event triggers for the first time all the registered receivers through this broadcast receiver will get notified for all the events by Android Runtime

Android Terminologies

P P SAVANI UNIVERSITY

- **Fragment**
  - Used to hold the code and screen logic for placing the same user interface components in multiple screens, which are represented by multiple Activity classes.
  - android.app.Fragment
- **Intent**
  - Intent is a messaging object which can be used to communicate between two or more components like activities, services, broadcast receiver etc. Intent can also be used to start an activity or service or to deliver a broadcast messages.

Android Terminologies

P P SAVANI UNIVERSITY

- **Context:**
  - **The Context class(android.content.context) is a fundamental building block of any android application and provides access to application-wide features such as the application's private files and device resources, as well as system-wide services.**
  - **The application wide Context object is instantiated as an Application object(android.app.Application).**

Performing Application Tasks with Activities

P P SAVANI UNIVERSITY

- **Activity** class is the core of any Android Application.
- You define and implement Activity class for each screen in your application.
- For example a simple game application might have the following five activities:
  - **A starting screen**: displays the application name and version
  - **A main menu screen**: users choose what they want to do within the application.
  - **A game play screen**
  - **A high score screen**
  - **A help/about screen**

Method stubs of Activity class

P P SAVANI UNIVERSITY

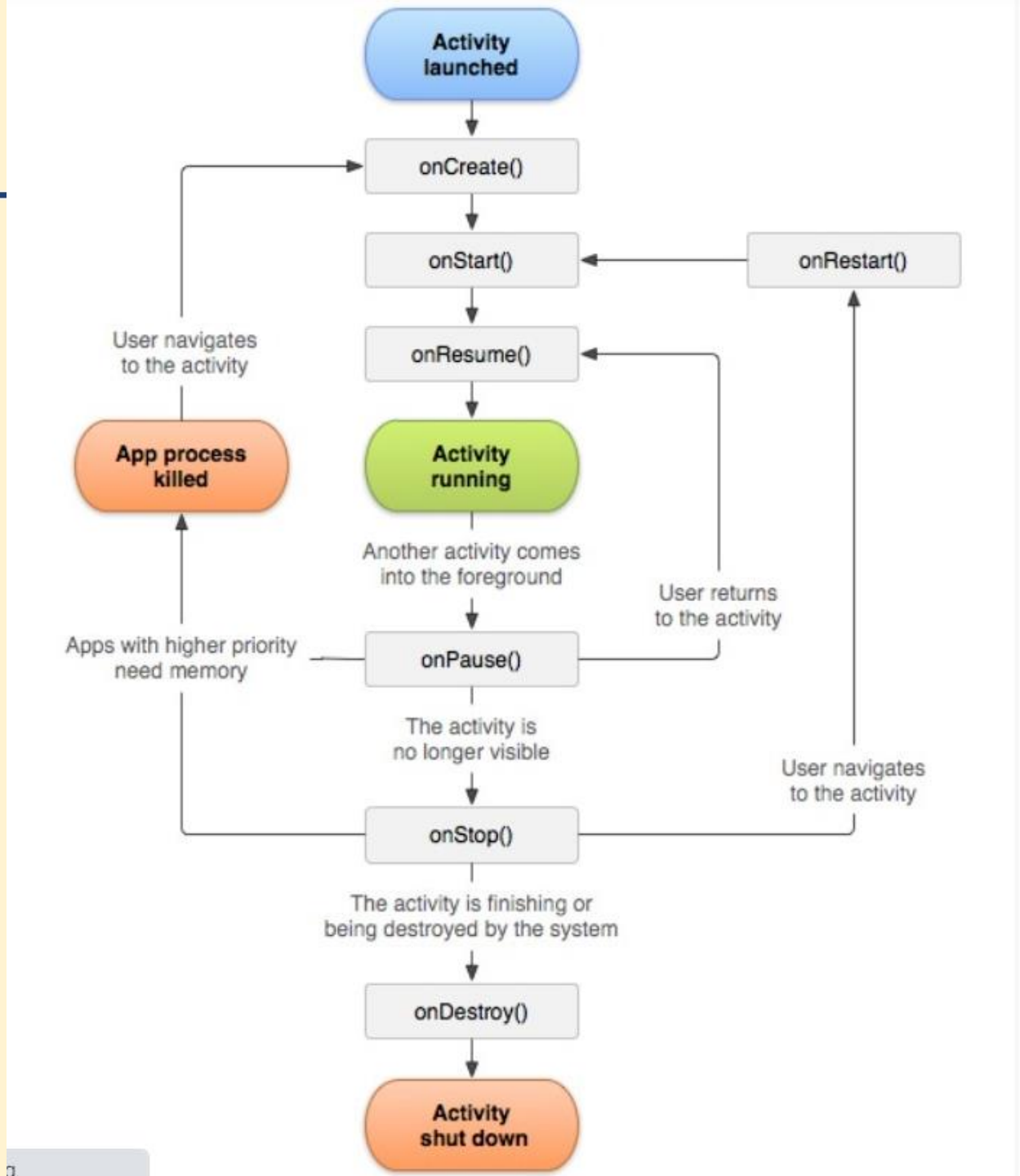
```
public class MyActivity extends Activity
{
   protected void onCreate(Bundle savedInstanceState);
   protected void onStart();
   protected void onRestart();
   protected void onResume();
   protected void onPause();
   protected void onStop();
   protected void onDestroy();
}
```

Android Activity Lifecycle methods

P P SAVANI UNIVERSITY

Method	Description
onCreate	called when activity is first created.
onStart	called when activity is becoming visible to the user.
onResume	called when activity will start interacting with the user.
onPause	called when activity is not visible to the user.
onStop	called when activity is no longer visible to the user.
onRestart	called after your activity is stopped, prior to start.
onDestroy	called before the activity is destroyed.

Activity Lifecycle



Managing Activity Transitions with Intents

P P SAVANI UNIVERSITY

- Launching the new Activity by Class Name

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

- Launching an Activity Belonging to Another Application

```
Uri number = Uri.parse("tel: 9998580268");
Intent dial = new Intent(Intent.ACTION_DIAL,number);
startActivity(dial);
```

Android Services

- Android Services are the application components that run in the background.
- We can understand it as a process that doesn't need any direct user interaction.
- As they perform long-running processes without user intervention, they have no User Interface(connection to server).
- They can be connected to other components and do **inter-process communication (IPC)**.
- Services must be registered in the Android manifest file.

Android Services

P P SAVANI UNIVERSITY

1. A weather, email or social network app might implement a service to routinely check for updates on the network.
2. A game might create a service to download and process the content for the next level before the user actually needs it.
3. A news application might implement a service to “preload” content by downloading news stories before the user launches the application, to improve performance and responsiveness.

Receiving and Broadcasting Intents

P P SAVANI UNIVERSITY

- You can broadcast an Intent(via a call to the sendBroadcast() method of the Context class) to the Android system at large,allowing any interested application(called a BroadcastReceiver) to receive that broadcast and act upon it.
- Broadcasts are generally used to inform the system that something interesting has happened.
- For example, a commonly listened-for broadcast Intent is ACTION_BATTERY_LOW, which is broadcast as a warning when the battery is low.

Context in Android

P P SAVANI UNIVERSITY

- context of the current state of our application.
- We can break the context and its use into three major points:
  - It allows us to access resources.
  - It allows us to interact with other Android components by sending messages.
  - It gives you information about your app environment.

Context in Android

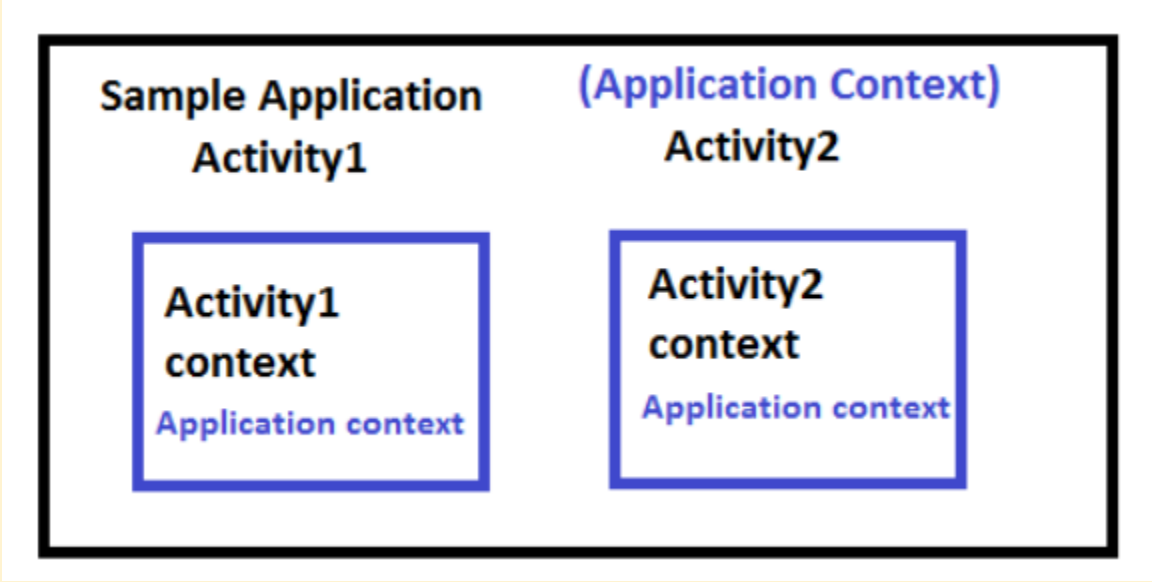
P P SAVANI UNIVERSITY

- **current state of the application.**
  - Usually, the app got multiple screens like display/inquiry/add/delete screens. So when the user is searching for something, the context is an inquiry screen in this case**.**
- **It can be used to get information regarding the activity and application.**
  - The inquiry screen's context specifies that the user is in inquiry activity, and he/she can submit queries related to the app
- **It allows us to interact with other Android components by sending messages.**
- **Both the Activity and Application classes extend the Context class.**

Understanding Context by a Real World Example

- Let’s a person visit a hotel. He needs breakfast, lunch, and dinner at a suitable time. Except for these things there are also many other things, he wants to do during the time of stay. So how does he get these things? He will ask the room-service person to bring these things for him. Right? So here the **room-service person is the context** considering **you are the single activity** and **the hotel to be your app**, finally, the **breakfast, lunch & dinner have to be the resources**.

Types of Context in Android



getApplicationContext()

P P SAVANI UNIVERSITY

- It is used to return the context which is linked to the Application which holds all activities running inside it. When we call a method or a constructor, we often have to pass a context and often we use **“this”** to pass the activity context or **“getApplicationContext”** to pass the application context. This method is generally used for the application level and can be used to refer to all the activities. For example, if we want to access a variable throughout the android app, one has to use it via **getApplicationContext()**

```java
import android.app.Application;

public class GlobalExampleClass extends Application
{
  private String globalName;
  private String globalEmail;

  public String getName()
  {
      return globalName;
  }

  public void setName(String aName)
  {
      globalName = aName;
  }

  public String getEmail()
  {
      return globalEmail;
  }

  public void setEmail(String aEmail)
  {
      globalEmail = aEmail;
  }
}
```

```java
// Activity 1
public class <your activity1> extends Activity {
  ........
  ........
  private <yourapplicationname> globarVar;
  ........
  @Override
  public void onCreate(Bundle savedInstanceState) {
    .......
    final GlobalExampleClass globalExampleVariable = (GlobalExampleClass) getApplicationContext

    // In this activity set name and email and can reuse in other activities
    globalExampleVariable.setName("getApplicationContext example");
    globalExampleVariable.setEmail("xxxxxx@gmail.com");

    .......
}
```

```java
// Activity 2
public class <your activity2> extends Activity {
  ........
  ........
  private <yourapplicationname> globarVar;
  .......
  @Override
  public void onCreate(Bundle savedInstanceState) {
    .......
    final GlobalExampleClass globalExampleVariable = (GlobalExampleClass) getApplicationContext

    // As in activity1, name and email is set, we can retrieve it here
    final String globalName  = globalExampleVariable.getName();
    final String globalEmail = globalExampleVariable.getEmail();

    .......
}
```

Activity Context

P P SAVANI UNIVERSITY

- It is the activity context meaning each and every screen got an activity.
- For example, EnquiryActivity refers to EnquiryActivity only and AddActivity refers to AddActivity only.
- The method of invoking the Activity Context is getContext().
- Some use cases of Activity Context are:
  - The user is creating an object whose lifecycle is attached to an activity.
  - Whenever inside an activity for UI related kind of operations like toast, dialogue, etc

Example

P P SAVANI UNIVERSITY

```
if(inpuName.isEmpty() || inputPassword.isEmpty())
{
    Toast.makeText( context: MainActivity.this, text: "Please Enter ALL the details correctly",Toast.LENGTH_SHORT).show();
}
```

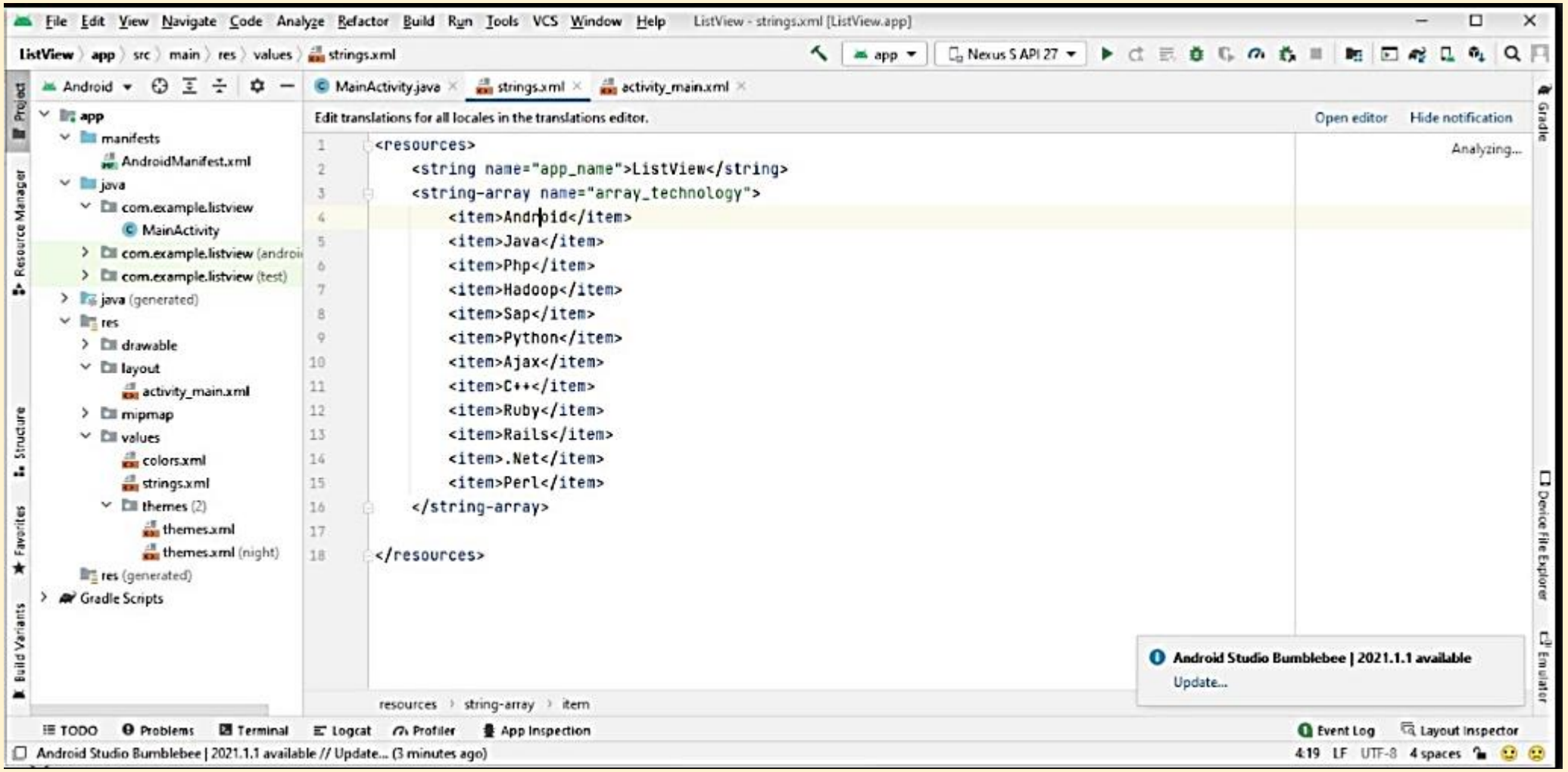
```
Toast.makeText( context: MainActivity.this, text: "Login Succeful",Toast.LENGTH_SHORT).show();
Intent intent=new Intent( packageContext: MainActivity.this,ActivityHomePage.class);
startActivity(intent);
```

P P SAVANI
UNIVERSITY

Some IMP Programs for 1st internal exam

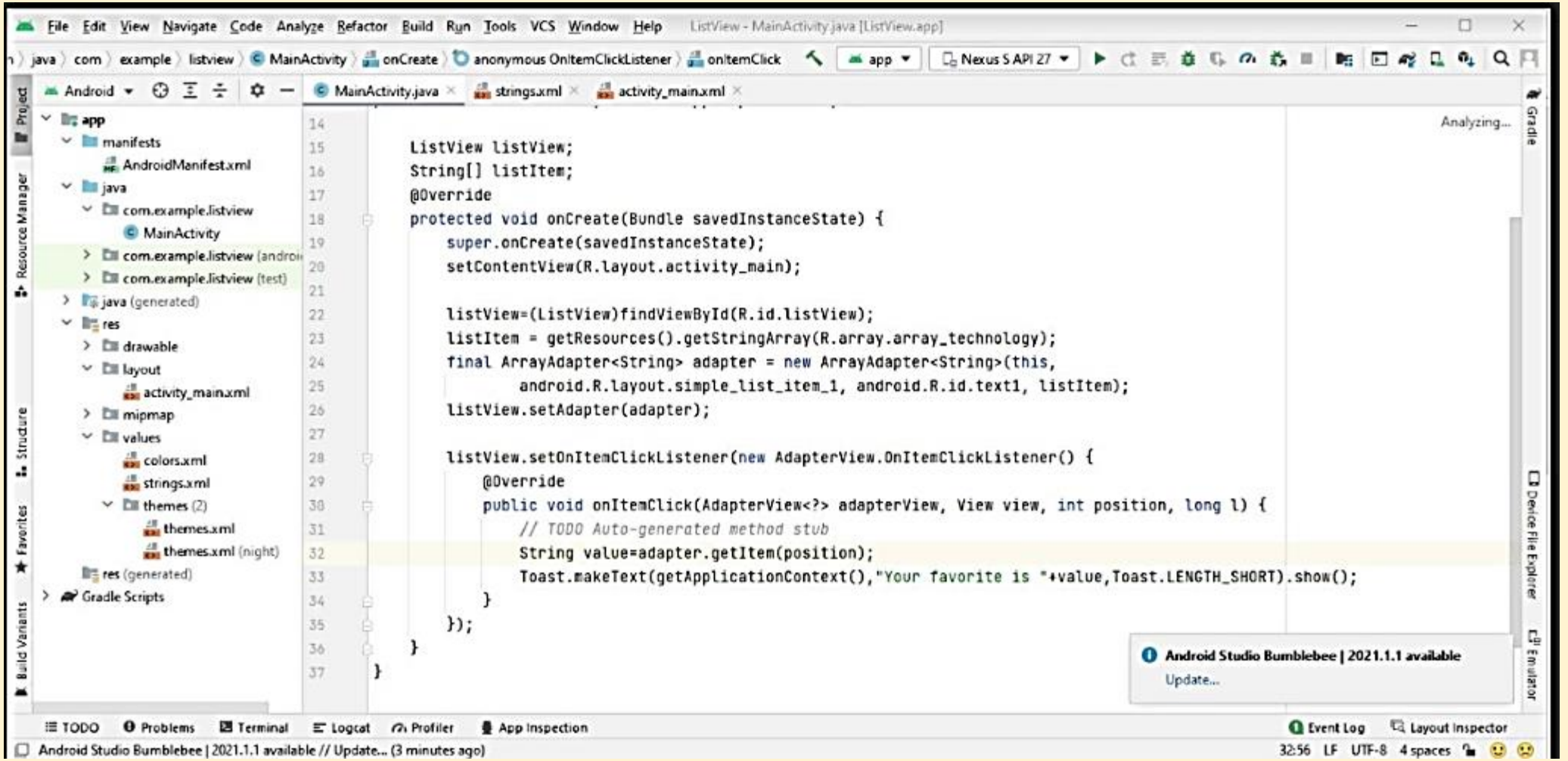
List view

P P SAVANI UNIVERSITY



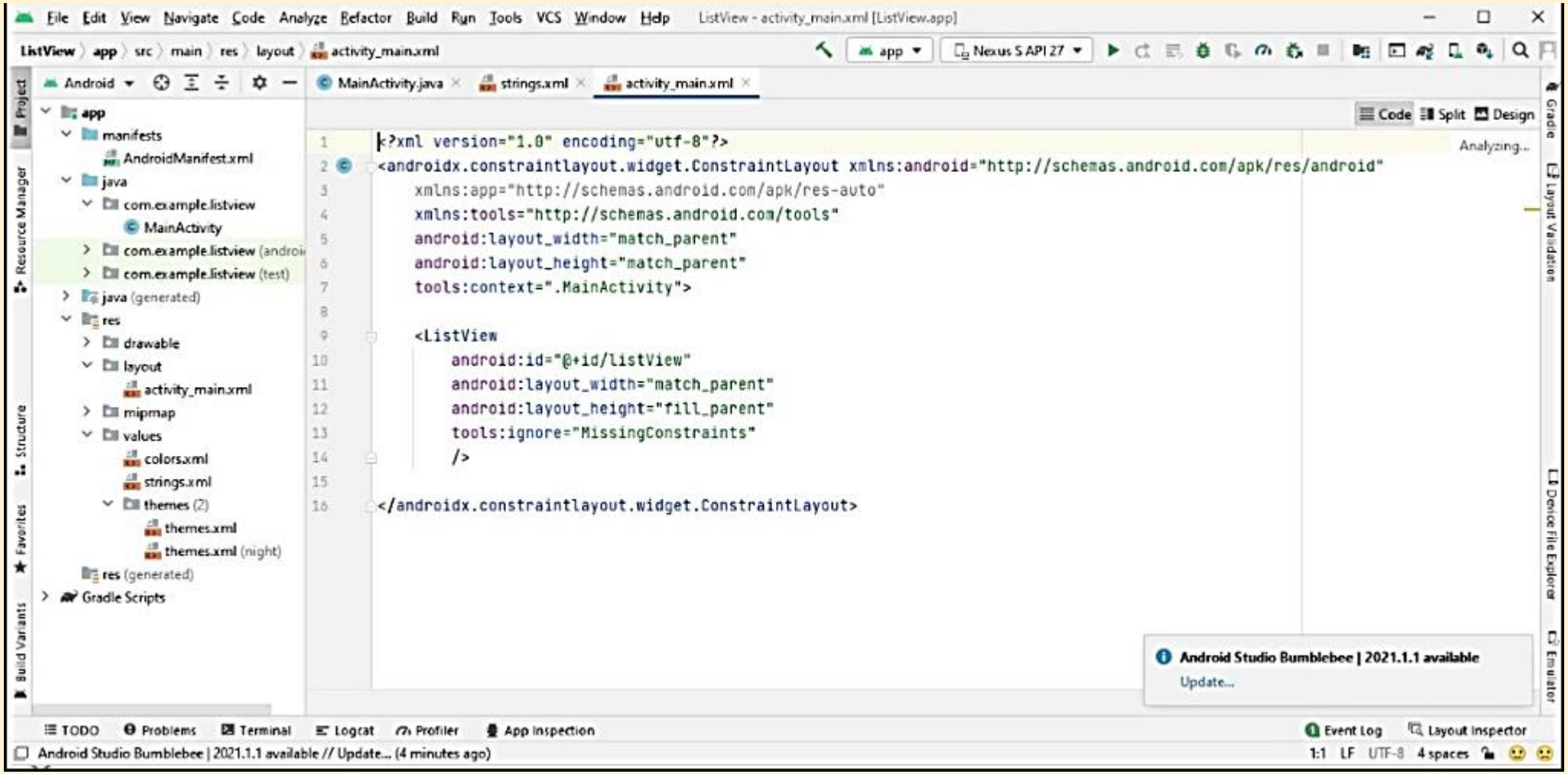
List view

P P SAVANI UNIVERSITY



List view

P P SAVANI UNIVERSITY



Login page with verification

P P SAVANI UNIVERSITY

```
public class MainActivity extends Activity  {
    Button b1,b2;
    EditText ed1,ed2;

    TextView tx1;
    int counter = 3;

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

        b1 = (Button)findViewById(R.id.button);
        ed1 = (EditText)findViewById(R.id.editText);
        ed2 = (EditText)findViewById(R.id.editText2);

        b2 = (Button)findViewById(R.id.button2);
        tx1 = (TextView)findViewById(R.id.textView3);
        tx1.setVisibility(View.GONE);

        b1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if(ed1.getText().toString().equals("admin") &&
                        ed2.getText().toString().equals("admin")) {

                    Toast.makeText(getApplicationContext(),
                            text: "Login successful",Toast.LENGTH_SHORT).show();
```

Login page with verification

P P SAVANI UNIVERSITY

```
        public void onClick(View v) {
            if(ed1.getText().toString().equals("admin") &&
                    ed2.getText().toString().equals("admin")) {

                Toast.makeText(getApplicationContext(),
                        text: "Login successful",Toast.LENGTH_SHORT).show();
            }else{
                Toast.makeText(getApplicationContext(),  text: "Wrong Credentials",Toast.LENGTH_SHORT).show();

                    tx1.setVisibility(View.VISIBLE);
                tx1.setBackgroundColor(Color.RED);
                counter--;
                tx1.setText(Integer.toString(counter));

                if (counter == 0) {
                    b1.setEnabled(false);
                }
            }
        }
    });

    b2.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) { finish(); }
    });
  }
}
```

Calculator

P P SAVANI UNIVERSITY

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help Calculator - MainActivity.java [Calculator.app]

Calculator app Nexus S API 27

Android activity_main.xml MainActivity.java

- app
  - manifests
  - java
    - com.example.calculator
      - MainActivity
    - com.example.calculator (androidTes
    - com.example.calculator (test)
  - java (generated)
  - res
  - res (generated)
- Gradle Scripts

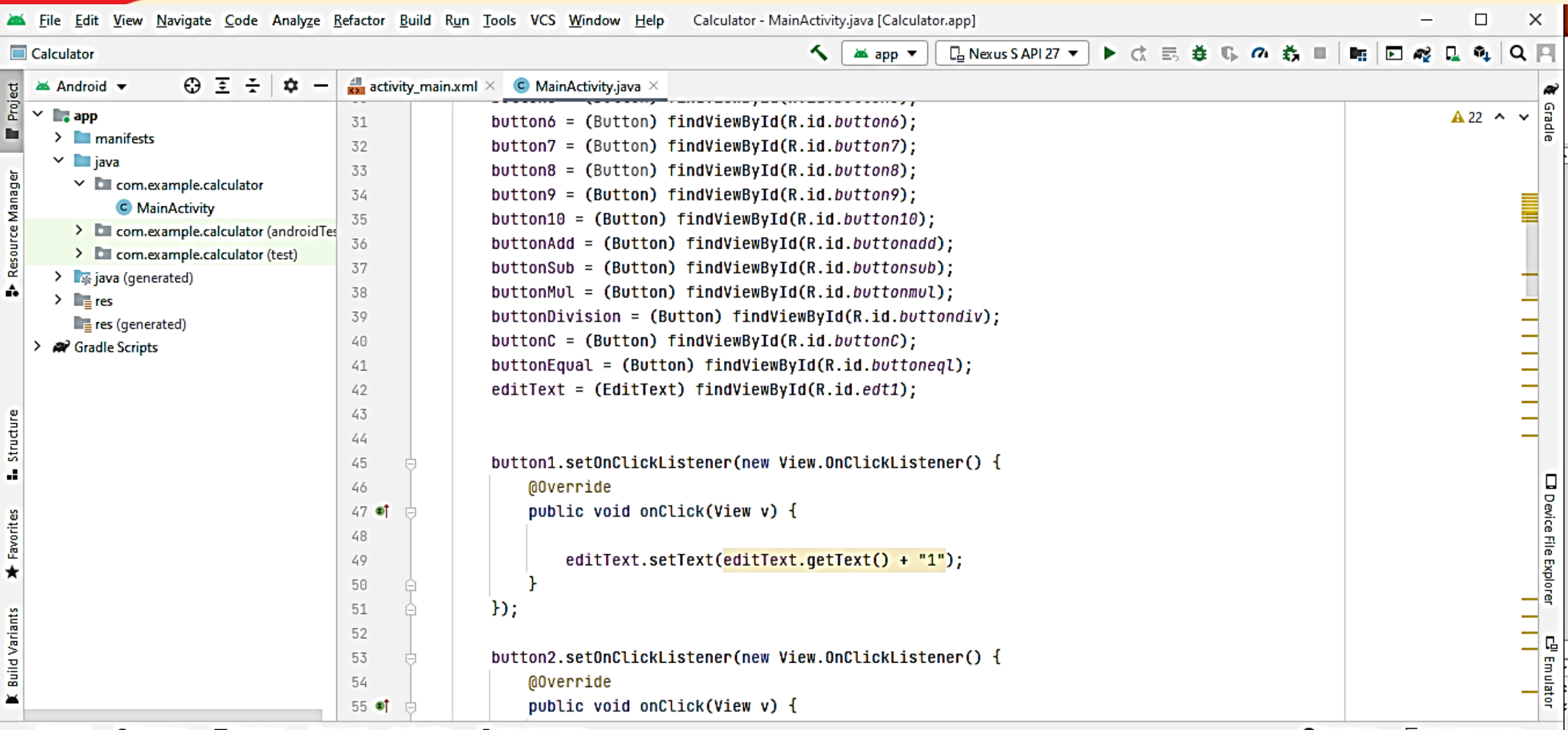
```
1  package com.example.calculator;
2
3  import ...
9
10 public class MainActivity extends AppCompatActivity {
11     Button button0, button1, button2, button3, button4, button5, button6,
12             button7, button8, button9, buttonAdd, buttonSub, buttonDivision,
13             buttonMul, button10, buttonC, buttonEqual;
14     EditText editText;
15
16     float mValueOne, mValueTwo;
17
18     boolean Addition, Subtract, Multiplication, Division;
19
20     @Override
21     protected void onCreate(Bundle savedInstanceState) {
22         super.onCreate(savedInstanceState);
23         setContentView(R.layout.activity_main);
24
25         button0 = (Button) findViewById(R.id.button0);
26         button1 = (Button) findViewById(R.id.button1);
27         button2 = (Button) findViewById(R.id.button2);
28         button3 = (Button) findViewById(R.id.button3);
29         button4 = (Button) findViewById(R.id.button4);
30         button5 = (Button) findViewById(R.id.button5);
```

Analyzing...

Project Resource Manager Structure Favorites Build Variants Gradle Device File Explorer Emulator

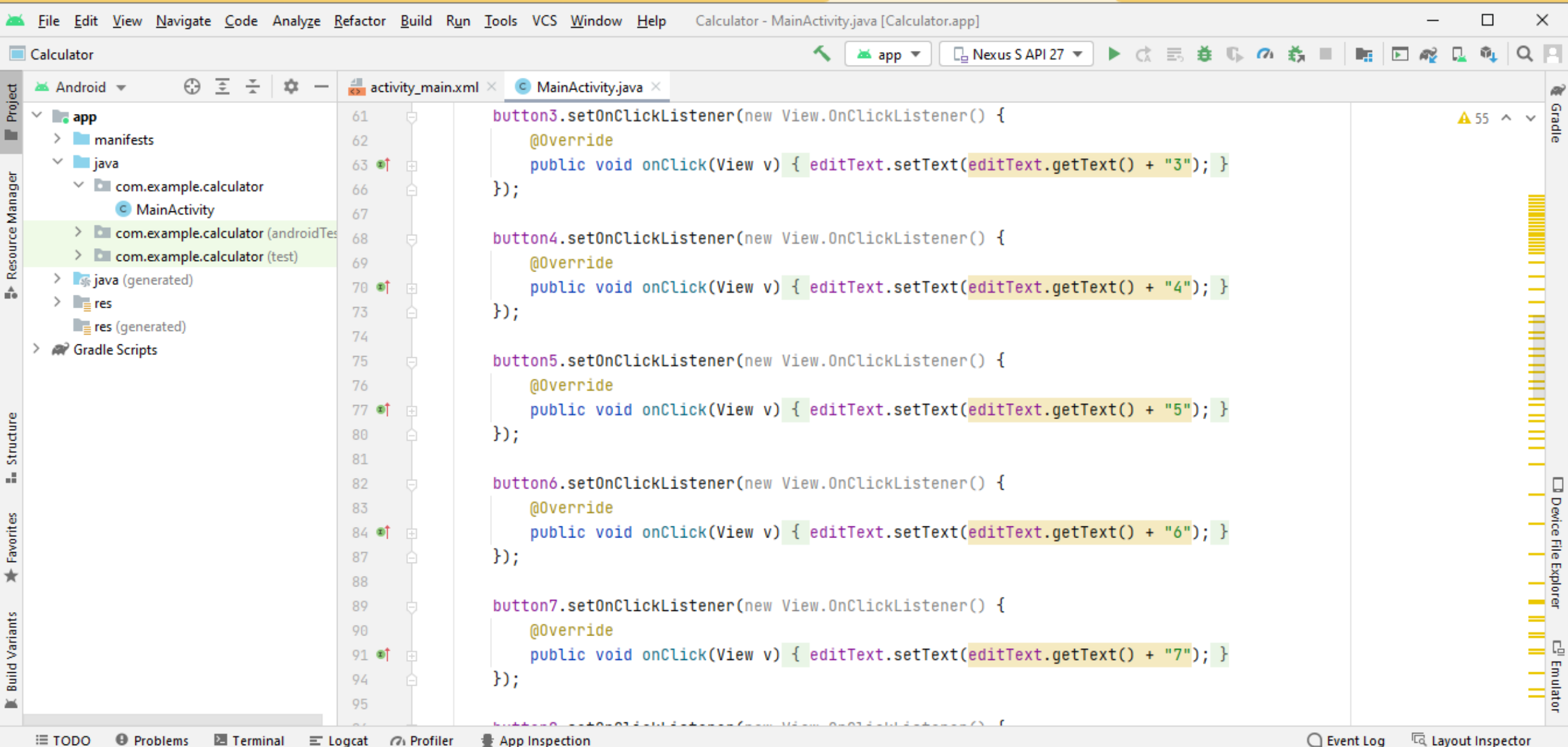
Calculator

P P SAVANI UNIVERSITY



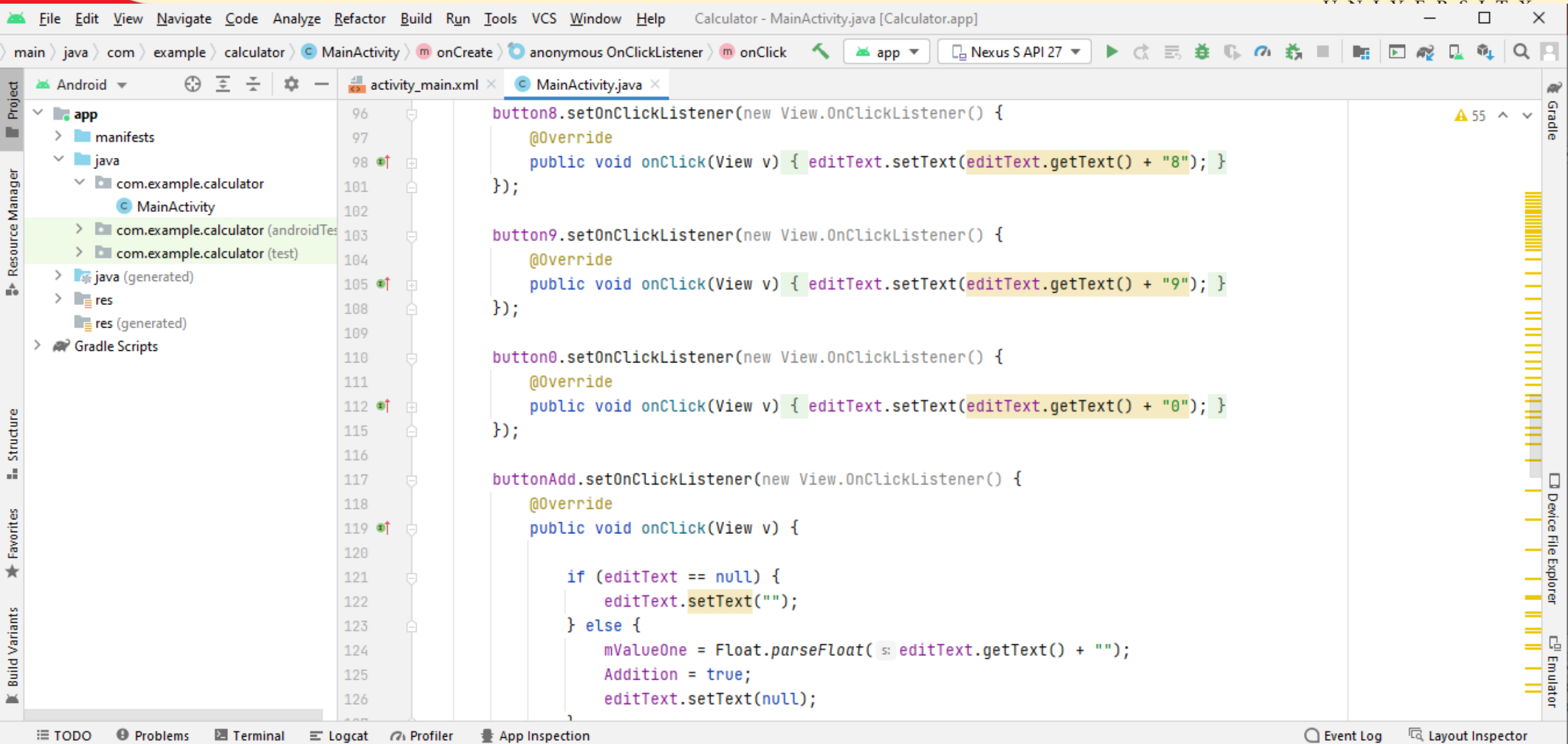
Calculator

P P SAVANI



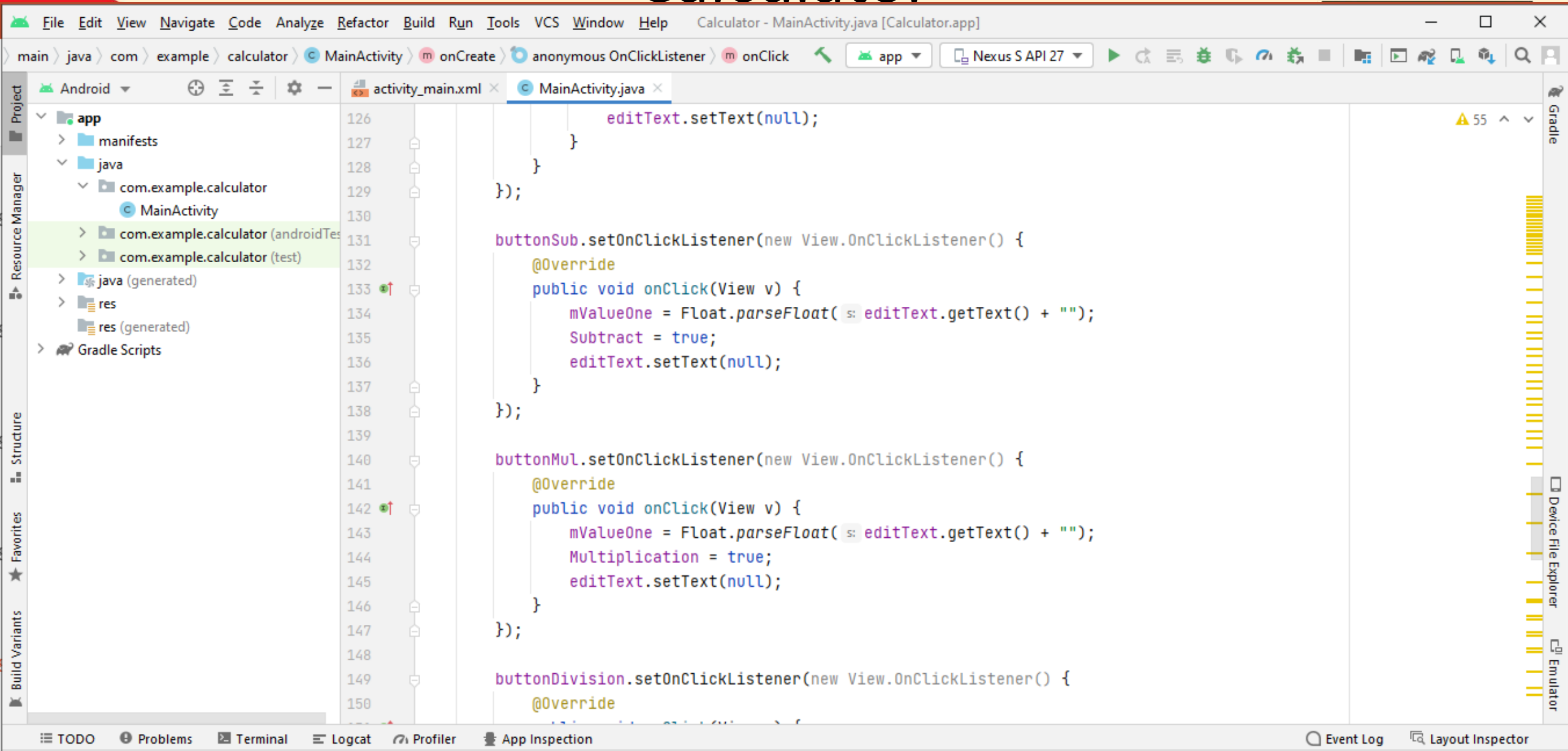
Calculator

P P SAVANI



Calculator

P P SAVANI



Calculator

P P SAVANI UNIVERSITY

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help Calculator - MainActivity.java [Calculator.app]

main > java > com > example > calculator > MainActivity > onCreate > anonymous OnClickListener > onClick app Nexus S API 27

Android

app
- manifests
- java
  - com.example.calculator
    - MainActivity
  - com.example.calculator (androidTes
  - com.example.calculator (test)
- java (generated)
- res
- res (generated)
- Gradle Scripts

activity_main.xml | MainActivity.java

```
152             mValueOne = Float.parseFloat( s: editText.getText() + "");
153             Division = true;
154             editText.setText(null);
155         }
156     });
157
158     buttonEqual.setOnClickListener(new View.OnClickListener() {
159         @Override
160         public void onClick(View v) {
161             mValueTwo = Float.parseFloat( s: editText.getText() + "");
162
163             if (Addition == true) {
164                 editText.setText(mValueOne + mValueTwo + "");
165                 Addition = false;
166             }
167
168             if (Subtract == true) {
169                 editText.setText(mValueOne - mValueTwo + "");
170                 Subtract = false;
171             }
172
173             if (Multiplication == true) {
174                 editText.setText(mValueOne * mValueTwo + "");
175                 Multiplication = false;
176             }
```

55

Gradle | Device File Explorer | Emulator | Project | Resource Manager | Structure | Favorites | Build Variants

Calculator

P P SAVANI UNIVERSITY

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help Calculator - MainActivity.java [Calculator.app]

main > java > com > example > calculator > MainActivity > onCreate > anonymous OnClickListener > onClick app Nexus S API 27

Android

app
- manifests
- java
  - com.example.calculator
    - MainActivity
  - com.example.calculator (androidTes
  - com.example.calculator (test)
- java (generated)
- res
- res (generated)
- Gradle Scripts

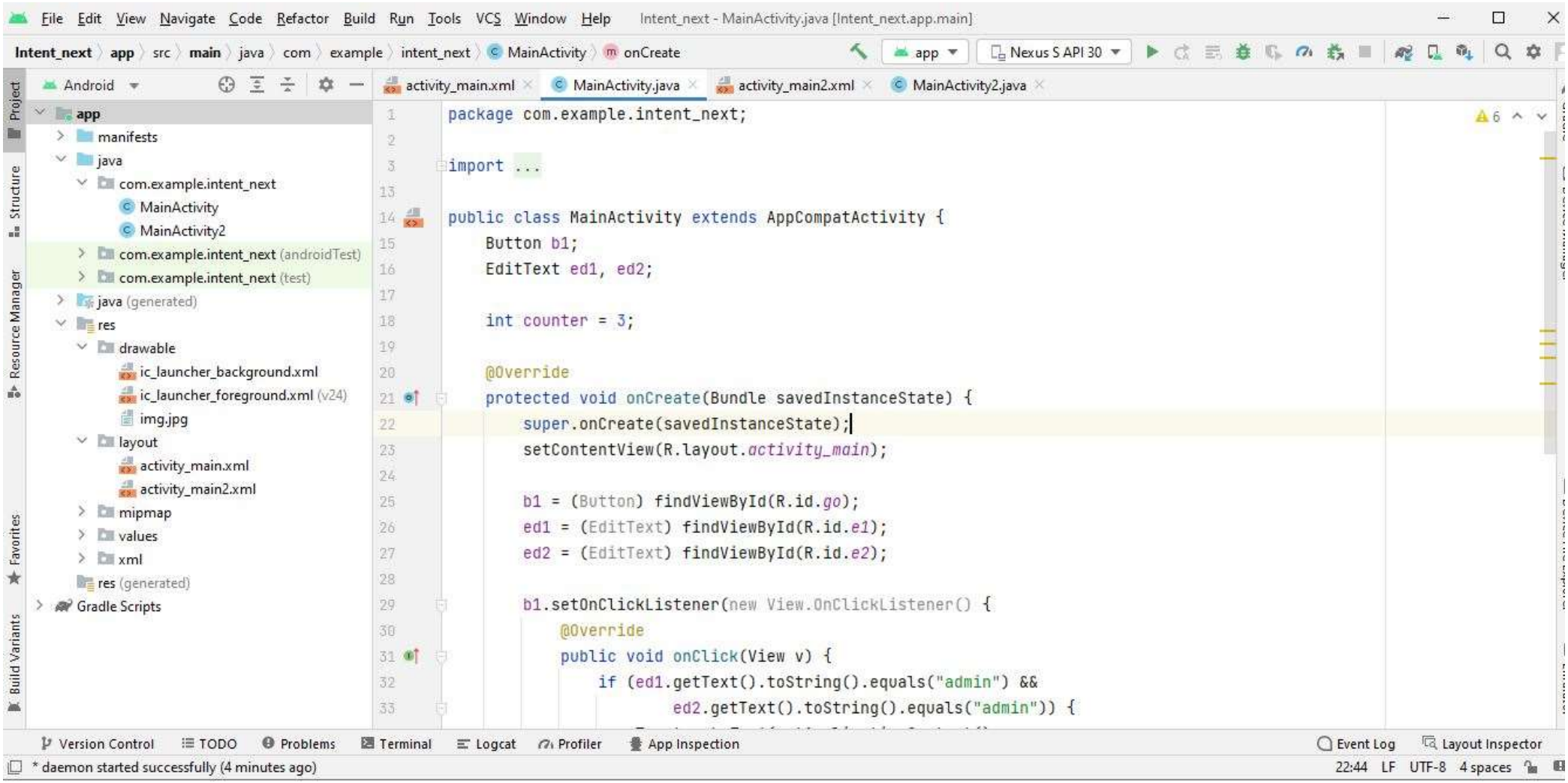
activity_main.xml | MainActivity.java

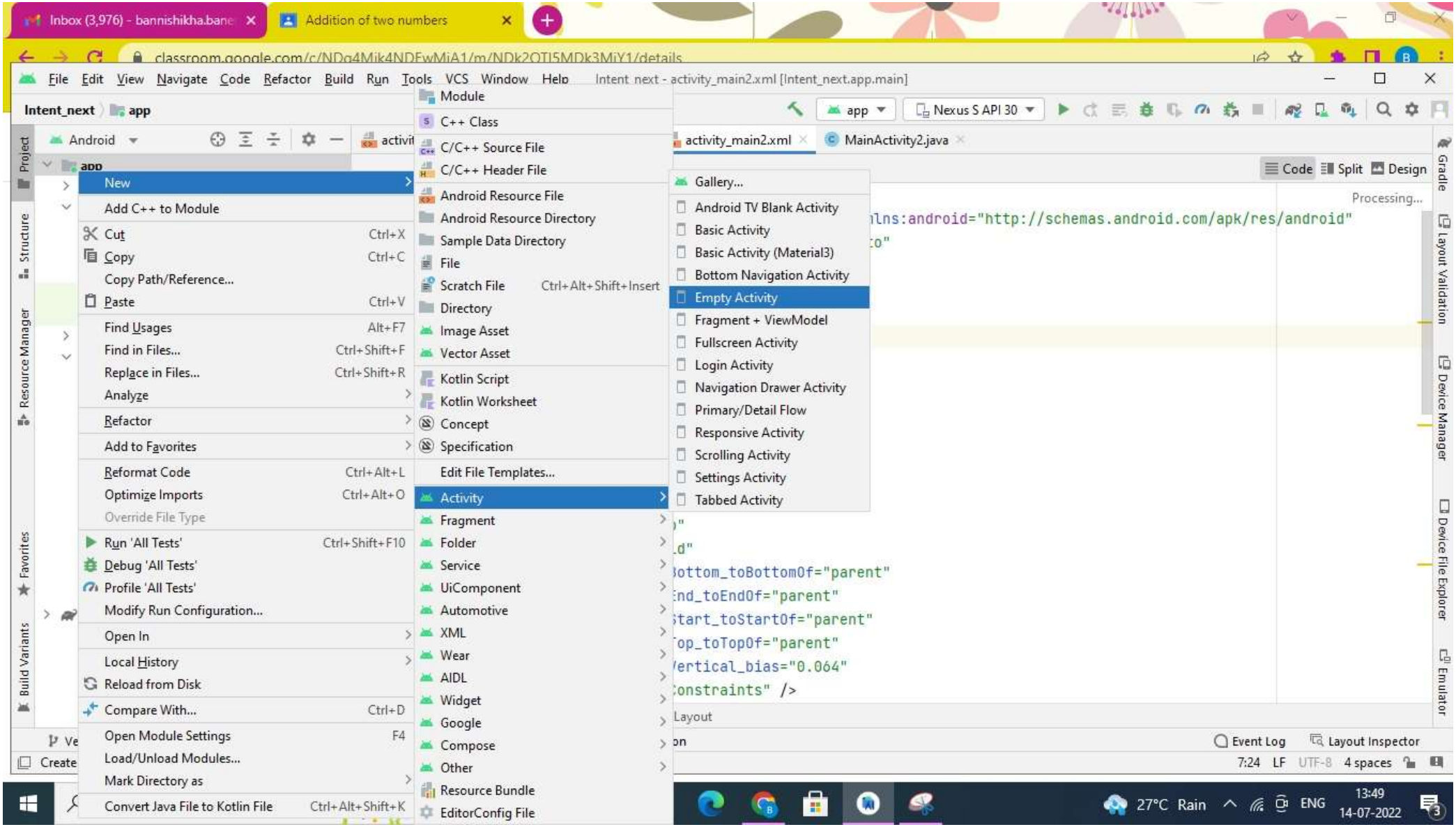
```
173                 if (Multiplication == true) {
174                     editText.setText(mValueOne * mValueTwo + "");
175                     Multiplication = false;
176                 }
177
178                 if (Division == true) {
179                     editText.setText(mValueOne / mValueTwo + "");
180                     Division = false;
181                 }
182             }
183         });
184
185         buttonC.setOnClickListener(new View.OnClickListener() {
186             @Override
187             public void onClick(View v) { editText.setText(""); }
190         });
191
192         button10.setOnClickListener(new View.OnClickListener() {
193             @Override
194             public void onClick(View v) { editText.setText(editText.getText() + "."); }
197         });
198     }
199 }
```

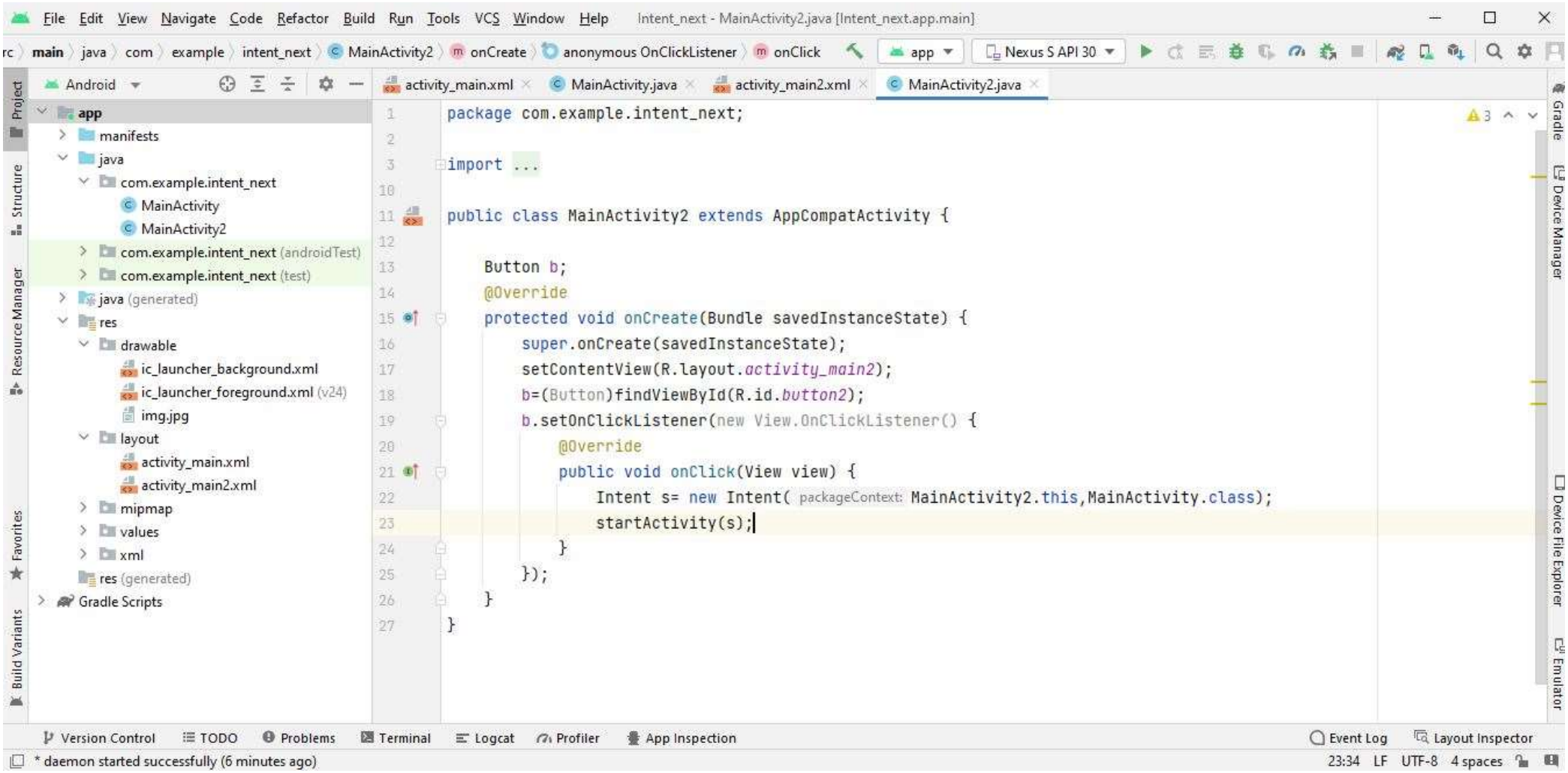
55

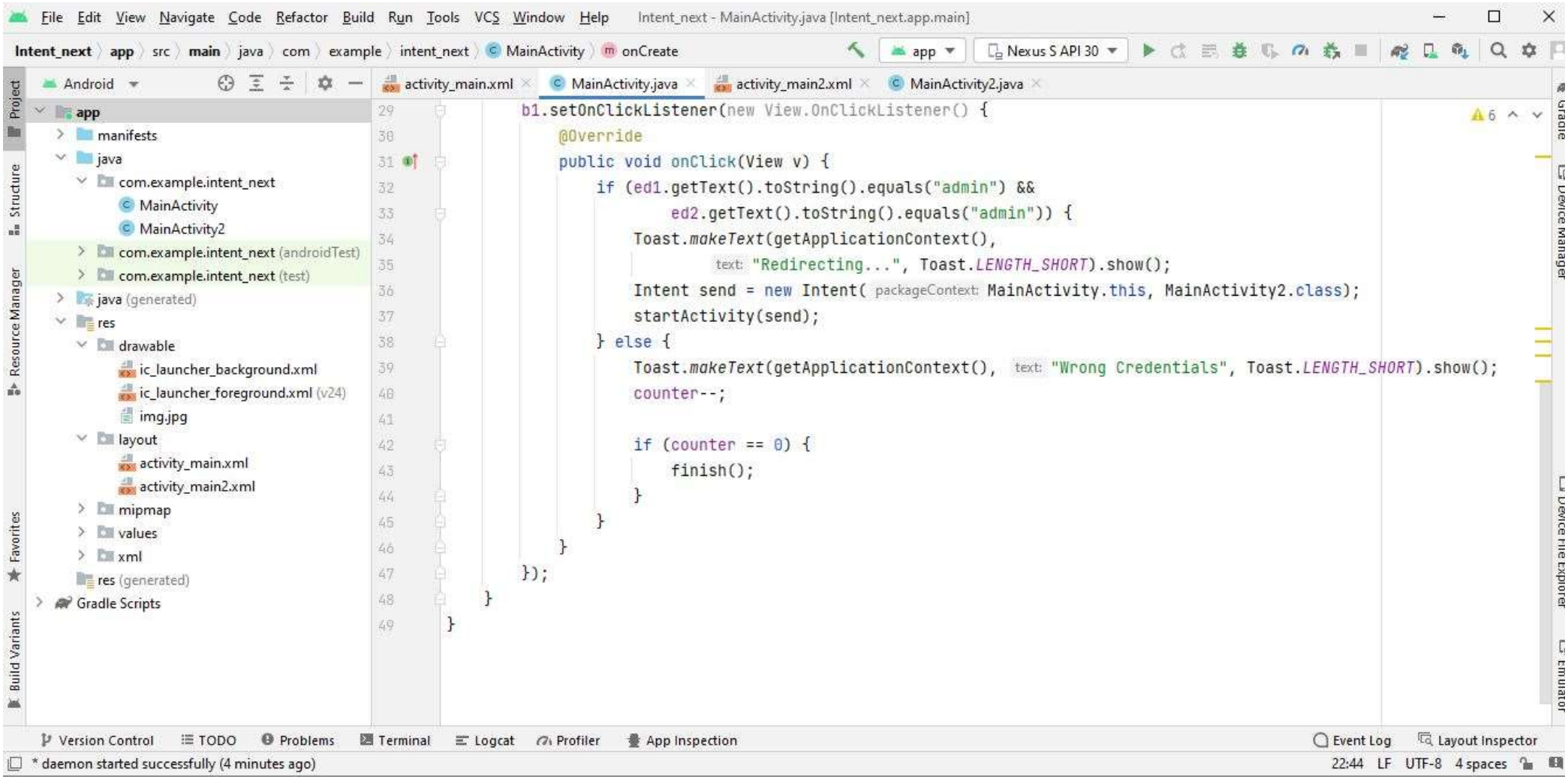
Project | Resource Manager | Structure | Favorites | Build Variants | Gradle | Device File Explorer | Emulator

Intent


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
Intent_next - MainActivity.java [Intent_next.app.main]
Intent_next app src main java com example intent_next MainActivity onCreate
app
Nexus S API 30
Android
activity_main.xml
MainActivity.java
activity_main2.xml
MainActivity2.java
app
manifests
java
com.example.intent_next
MainActivity
MainActivity2
com.example.intent_next (androidTest)
com.example.intent_next (test)
java (generated)
res
drawable
ic_launcher_background.xml
ic_launcher_foreground.xml (v24)
img.jpg
layout
activity_main.xml
activity_main2.xml
mipmap
values
xml
res (generated)
Gradle Scripts
package com.example.intent_next;
import ...
public class MainActivity extends AppCompatActivity {
Button b1;
EditText ed1, ed2;
int counter = 3;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
b1 = (Button) findViewById(R.id.go);
ed1 = (EditText) findViewById(R.id.e1);
ed2 = (EditText) findViewById(R.id.e2);
b1.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
if (ed1.getText().toString().equals("admin") &&
ed2.getText().toString().equals("admin")) {
Version Control TODO Problems Terminal Logcat Profiler App Inspection
Event Log Layout Inspector
* daemon started successfully (4 minutes ago)
22:44 LF UTF-8 4 spaces


Inbox (3,976) - bannishikha.bane
Addition of two numbers
classroom.google.com/c/NDg4Mik4NDEwMiA1/m/NDk2OTI5MDk3MiY1/details
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
Intent_next - activity_main2.xml [Intent_next.app.main]
Intent_next app
app
Nexus S API 30
activity_main2.xml
MainActivity2.java
Code Split Design
Android
Project
Structure
Resource Manager
Favorites
Build Variants
Gradle
Layout Validation
Device Manager
Device File Explorer
Emulator
New
Add C++ to Module
Cut Ctrl+X
Copy Ctrl+C
Copy Path/Reference...
Paste Ctrl+V
Find Usages Alt+F7
Find in Files... Ctrl+Shift+F
Replace in Files... Ctrl+Shift+R
Analyze
Refactor
Add to Favorites
Reformat Code Ctrl+Alt+L
Optimize Imports Ctrl+Alt+O
Override File Type
Run 'All Tests' Ctrl+Shift+F10
Debug 'All Tests'
Profile 'All Tests'
Modify Run Configuration...
Open In
Local History
Reload from Disk
Compare With... Ctrl+D
Open Module Settings F4
Load/Unload Modules...
Mark Directory as
Convert Java File to Kotlin File Ctrl+Alt+Shift+K
Module
C++ Class
C/C++ Source File
C/C++ Header File
Android Resource File
Android Resource Directory
Sample Data Directory
File
Scratch File Ctrl+Alt+Shift+Insert
Directory
Image Asset
Vector Asset
Kotlin Script
Kotlin Worksheet
Concept
Specification
Edit File Templates...
Activity
Fragment
Folder
Service
UiComponent
Automotive
XML
Wear
AIDL
Widget
Google
Compose
Other
Resource Bundle
EditorConfig File
Gallery...
Android TV Blank Activity
Basic Activity
Basic Activity (Material3)
Bottom Navigation Activity
Empty Activity
Fragment + ViewModel
Fullscreen Activity
Login Activity
Navigation Drawer Activity
Primary/Detail Flow
Responsive Activity
Scrolling Activity
Settings Activity
Tabbed Activity
Processing...
lns:android="http://schemas.android.com/apk/res/android"
Layout
Event Log
Layout Inspector
7:24 LF UTF-8 4 spaces
27°C Rain
ENG
13:49
14-07-2022


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
Intent_next - MainActivity2.java [Intent_next.app.main]
main java com example intent_next MainActivity2 onCreate anonymous OnClickListener onClick
app
Nexus S API 30
Android
app
manifests
java
com.example.intent_next
MainActivity
MainActivity2
com.example.intent_next (androidTest)
com.example.intent_next (test)
java (generated)
res
drawable
ic_launcher_background.xml
ic_launcher_foreground.xml (v24)
img.jpg
layout
activity_main.xml
activity_main2.xml
mipmap
values
xml
res (generated)
Gradle Scripts
activity_main.xml
MainActivity.java
activity_main2.xml
MainActivity2.java
package com.example.intent_next;
import ...
public class MainActivity2 extends AppCompatActivity {
Button b;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main2);
b=(Button)findViewById(R.id.button2);
b.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
Intent s= new Intent(packageContext: MainActivity2.this,MainActivity.class);
startActivity(s);
}
});
}
}
Version Control TODO Problems Terminal Logcat Profiler App Inspection
Event Log Layout Inspector
* daemon started successfully (6 minutes ago)
23:34 LF UTF-8 4 spaces


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
Intent_next - MainActivity.java [Intent_next.app.main]
Intent_next app src main java com example intent_next MainActivity onCreate
app
Nexus S API 30
Android
activity_main.xml
MainActivity.java
activity_main2.xml
MainActivity2.java
app
manifests
java
com.example.intent_next
MainActivity
MainActivity2
com.example.intent_next (androidTest)
com.example.intent_next (test)
java (generated)
res
drawable
ic_launcher_background.xml
ic_launcher_foreground.xml (v24)
img.jpg
layout
activity_main.xml
activity_main2.xml
mipmap
values
xml
res (generated)
Gradle Scripts
b1.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
if (ed1.getText().toString().equals("admin") &&
ed2.getText().toString().equals("admin")) {
Toast.makeText(getApplicationContext(),
text: "Redirecting...", Toast.LENGTH_SHORT).show();
Intent send = new Intent(packageContext: MainActivity.this, MainActivity2.class);
startActivity(send);
} else {
Toast.makeText(getApplicationContext(), text: "Wrong Credentials", Toast.LENGTH_SHORT).show();
counter--;
if (counter == 0) {
finish();
}
}
}
});
}
}
Version Control
TODO
Problems
Terminal
Logcat
Profiler
App Inspection
Event Log
Layout Inspector
* daemon started successfully (4 minutes ago)
22:44 LF UTF-8 4 spaces