COMP08076
Programming Native App Interaction

- Module coordinator: John Nixon
  - john.nixon@uws.ac.uk, room E259, x3617

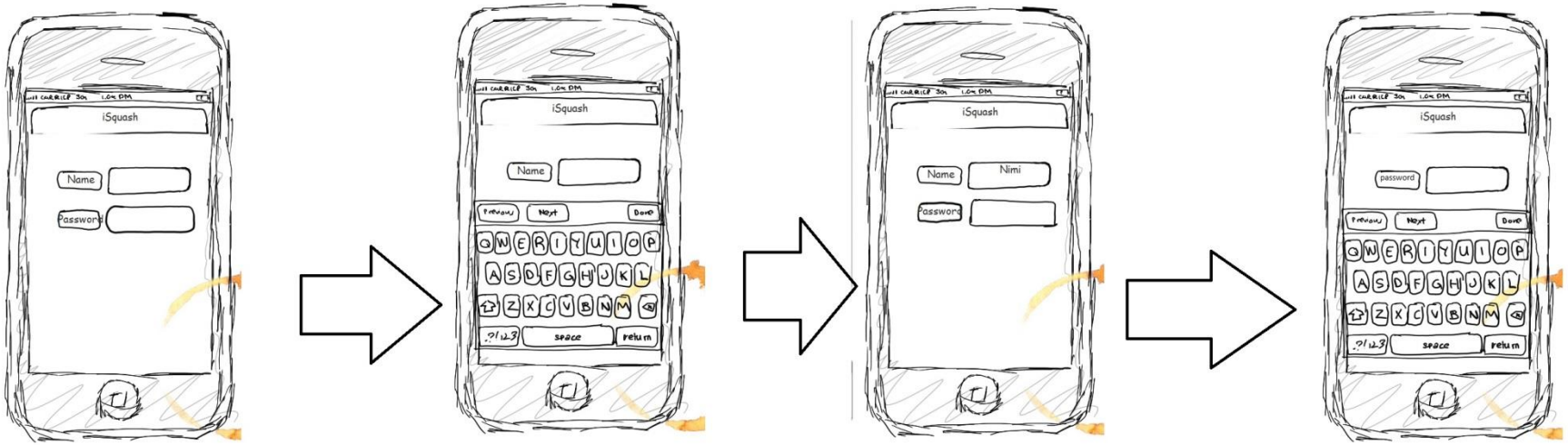
- Lecture 3
  - Wireframes and design
  - Some java
  - The Activity – the building block of Android apps
  - Activity life-cycle
  - Views

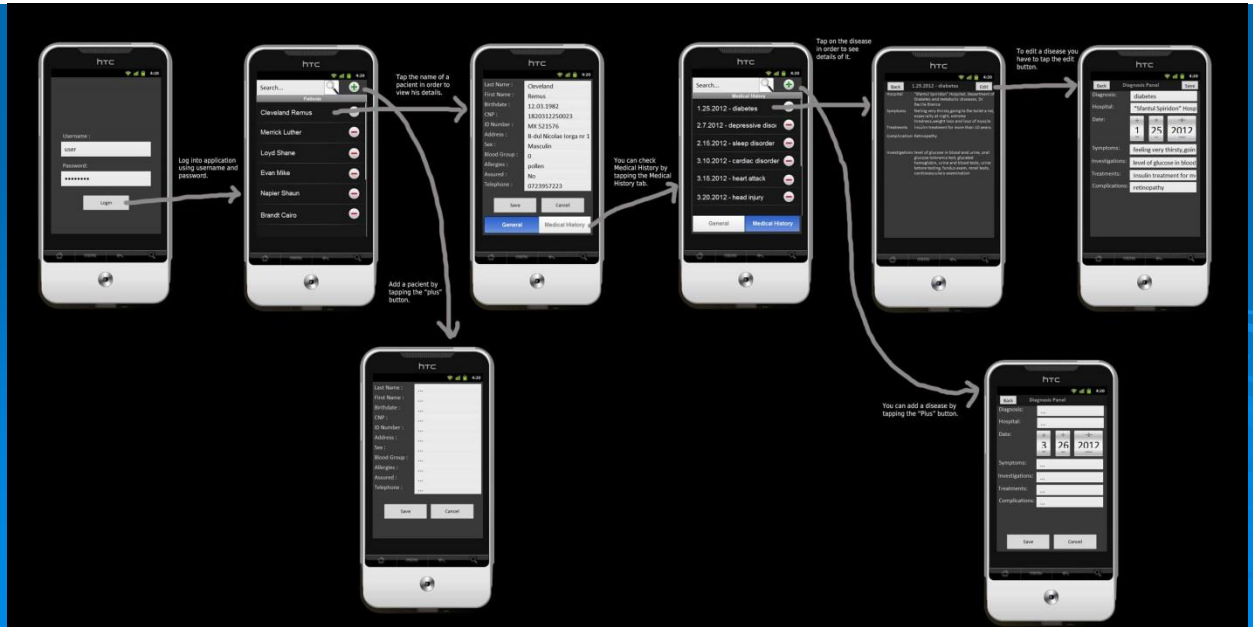
Wireframe

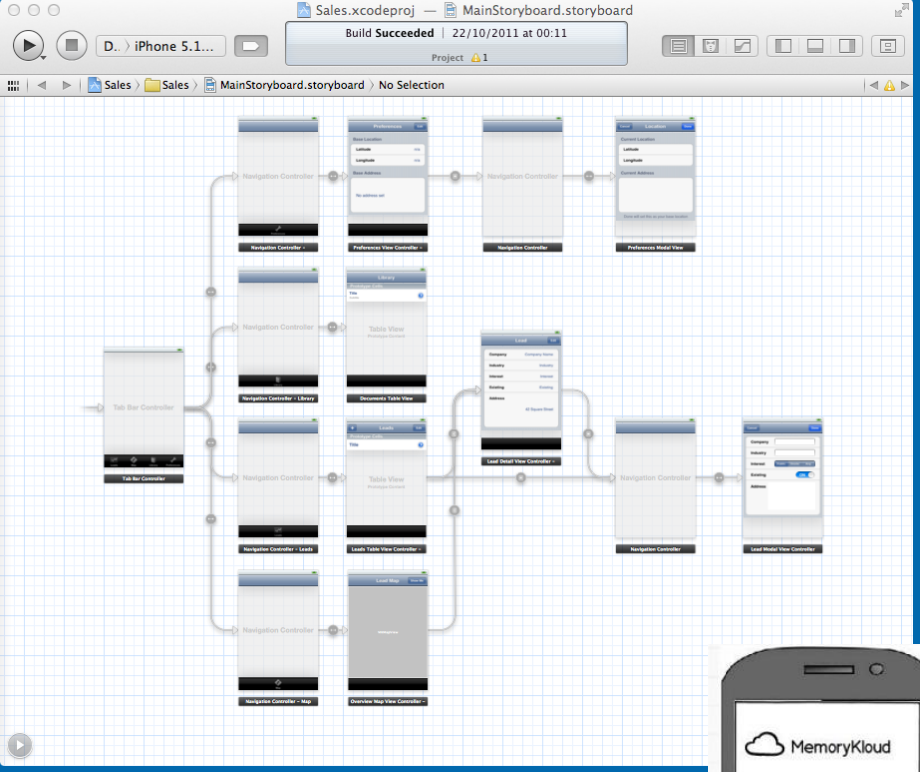
- A **wireframe** is a visual representation of the skeletal structure of a website or mobile application
  - Information design (information architecture)
  - Navigation design
  - Interface design

- http://webdesign.tutsplus.com/tutorials/workflow-tutorials/a-beginners-guide-to-wireframing/

- Numerous software packages


iSquash
Name
Password
Nimi
password
Previous
Next
Done
space
return


Log into application using username and password.
Tap the name of a pacient in order to view his details.
Add a pacient by tapping the "plus" button.
You can check Medical History by tapping the Medical History tab.
Tap on the disease in order to see details of it.
To edit a disease you have to tap the edit button.
You can add a disease by tapping the "Plus" button.


Sales.xcodeproj — MainStoryboard.storyboard
Build Succeeded | 22/10/2011 at 00:11
Sales › Sales › MainStoryboard.storyboard › No Selection

Balsamiq


MemoryKloud
Username
Password
Sign In
New User?
Kloudlets
Barbados
Mary's wedding
Graduation
TEDxSeattle
Awesome talk!
New Image
Add to
TedxSeattle
Description
The yummy food they had
Geolocatio
ON
Publish on
Add
New Kloudlet
Name
Las Vegas Trip
Co-Workers Trip to Vegas on Jan 23rd, 2011
Create!
Pin
Edit
Delete
Settings
The Sky's the limit!
Jennie Marcus
Berkeley, CA
Networks
Notifications
Friend Invites
Kloudlet Requests
Save
About Us
Version
A
B
C
D
E
F
G
H
I
J

Variables

- Are for holding data
- Declaring a variable
  - boolean flag; or boolean flag = false;
    - true or false
  - double myHeight = 4.67;
  - float myHeight = 4.67f;
  - int numberOfPages = 2;
  - String hotelName = “Imperial”;
- Scope of a variable
  - Class variables, local variables

If statement

- if (condition is true) {
  - do something;
  }

- if (condition is true) {
  - do something;
  } else {
  - do something different
  }

Things to test in an if statement

- Is a boolean true or false
  - if(isMoving){…}
- Comparison of values
  - if (x > y) {…}
- Component properties
  - if (myCheckbox.ischecked()) {…}
  - if (myRadioButton.ischecked()) {…}
  - Use android reference to find out more about components
  - E.g. http://developer.android.com/reference/android/widget/CheckBox.html

```
public class MainActivity extends Activity {

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

// other stuff

}
```

What is an Activity?

- **A class**
- **Activity**
  - A application component that has a **screen** with which users can interact
    - To make a phone call, take a photo etc.
  - Each screen of an application is a different activity (i.e. an application may have several activities)
  - Activity has a life cycle
  - Sub class the activity class to make one
- Discussion on activities
  - http://developer.android.com/guide/components/activities.html

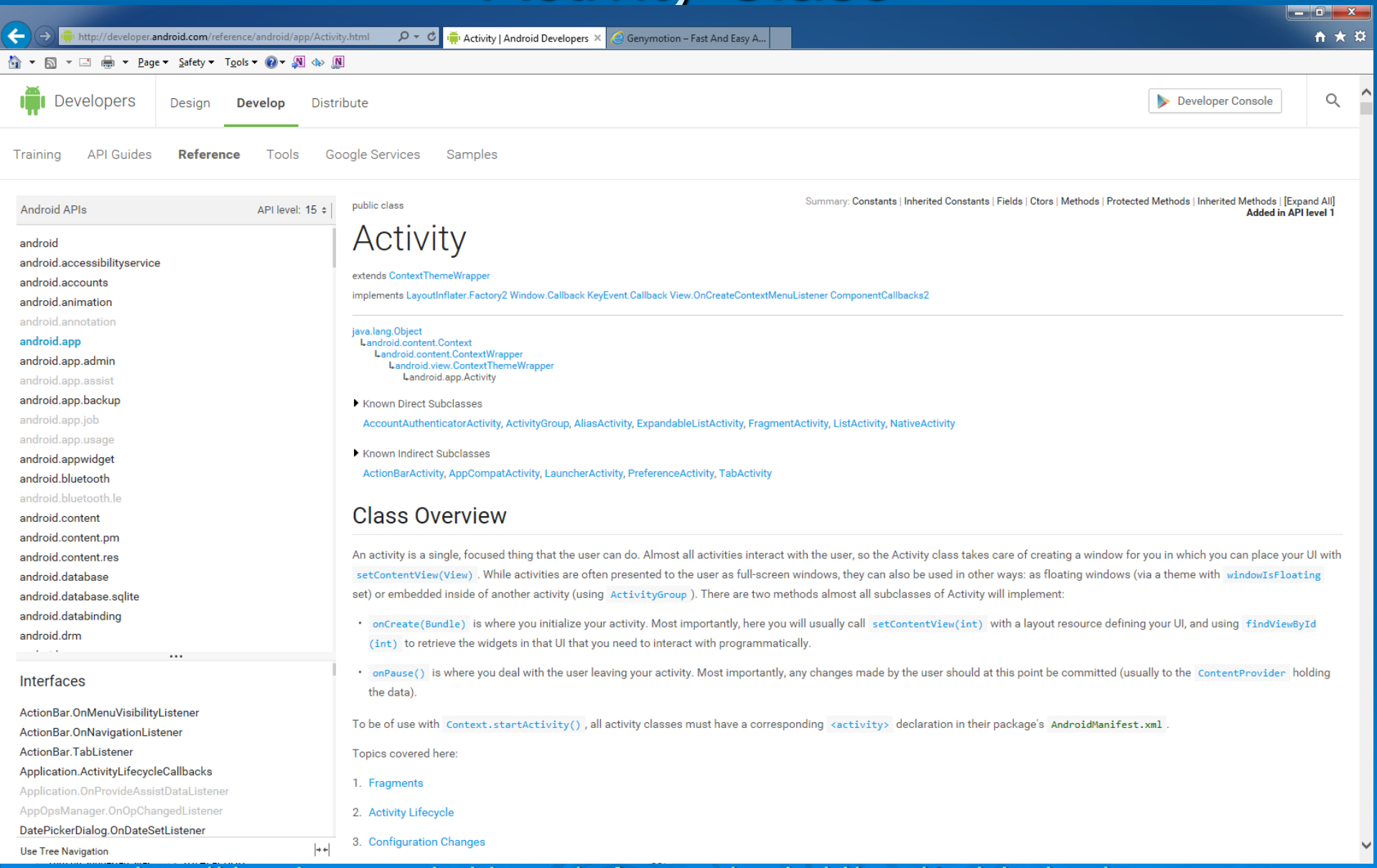
Objects and Classes

- A class is a template for a type of object
  - All objects are defined by a class
  - All objects of the same class have the same attributes
  - All objects of the same class have a defined set of methods
- Inheritance
- An existing class can be used as the starting point for a new one
  - The new class **extends** the existing one
  - Methods in the new class **override** existing methods

```
public class MainActivity extends Activity {

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

// other stuff

}
```

- Create a subclass of activity
  - public class MainActivity extends Activity {
- Implement the onCreate method – the **system** calls this when creating your activity
  - protected void onCreate(Bundle savedInstanceState) {
  - *protected* – accessible only from within class and any subclass – inherited by subclass
  - savedInstanceState – stores information about activity to recreate it if required
- @Override – a compiler directive

Activity Class



- http://developer.android.com/reference/android/app/Activity.html

public class

Activity

Summary: **Constants** | **Inherited Cons**

extends ContextThemeWrapper

implements LayoutInflater.Factory2 Window.Callback KeyEvent.Callback View.OnCreateContextMenuListener ComponentCallbacks2

java.lang.Object
  ↳android.content.Context
    ↳android.content.ContextWrapper
      ↳android.view.ContextThemeWrapper
        ↳android.app.Activity

▶ Known Direct Subclasses

AccountAuthenticatorActivity, ActivityGroup, AliasActivity, ExpandableListActivity, FragmentActivity, ListActivity, NativeActivity

▶ Known Indirect Subclasses

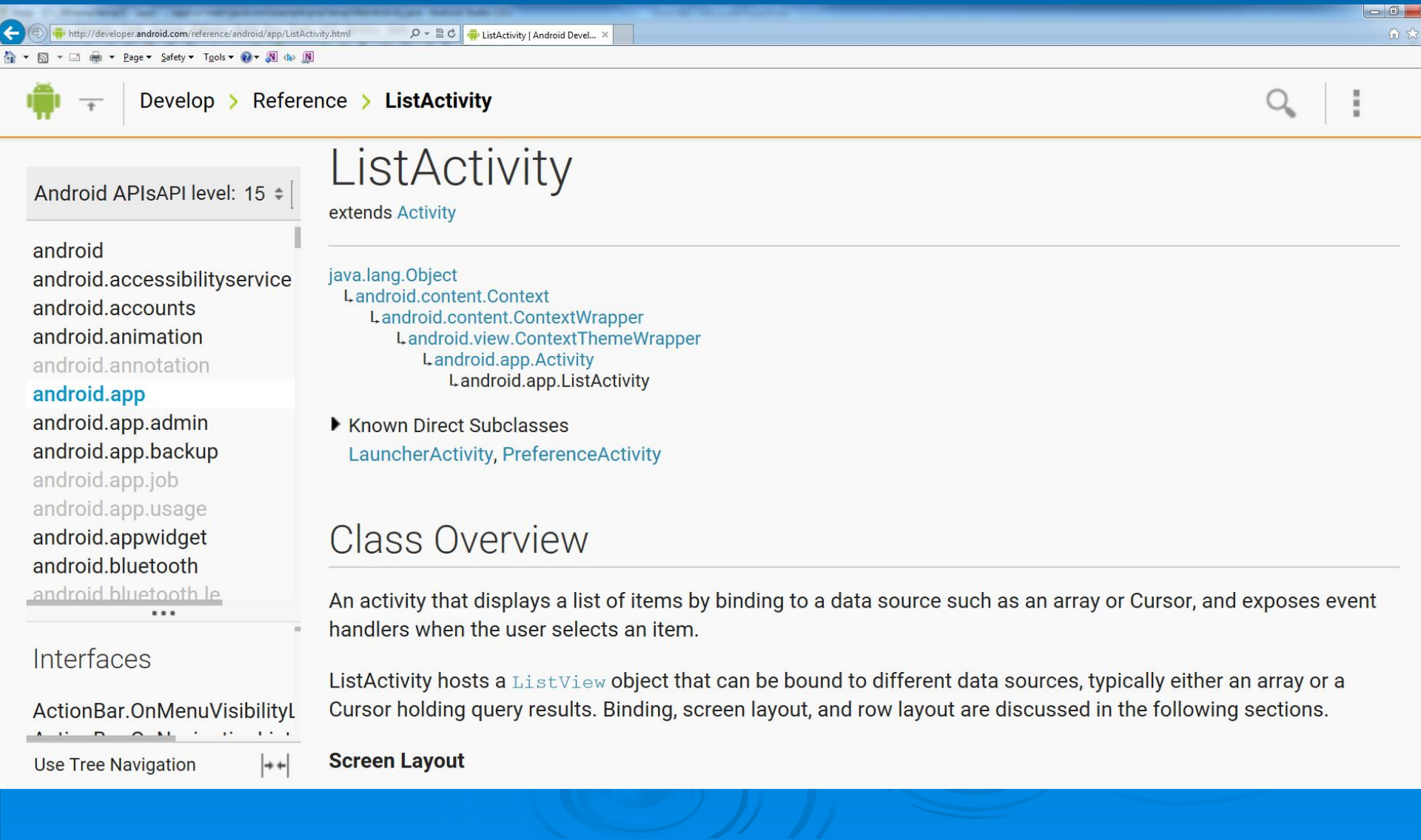
ActionBarActivity, AppCompatActivity, LauncherActivity, PreferenceActivity, TabActivity

Class Overview

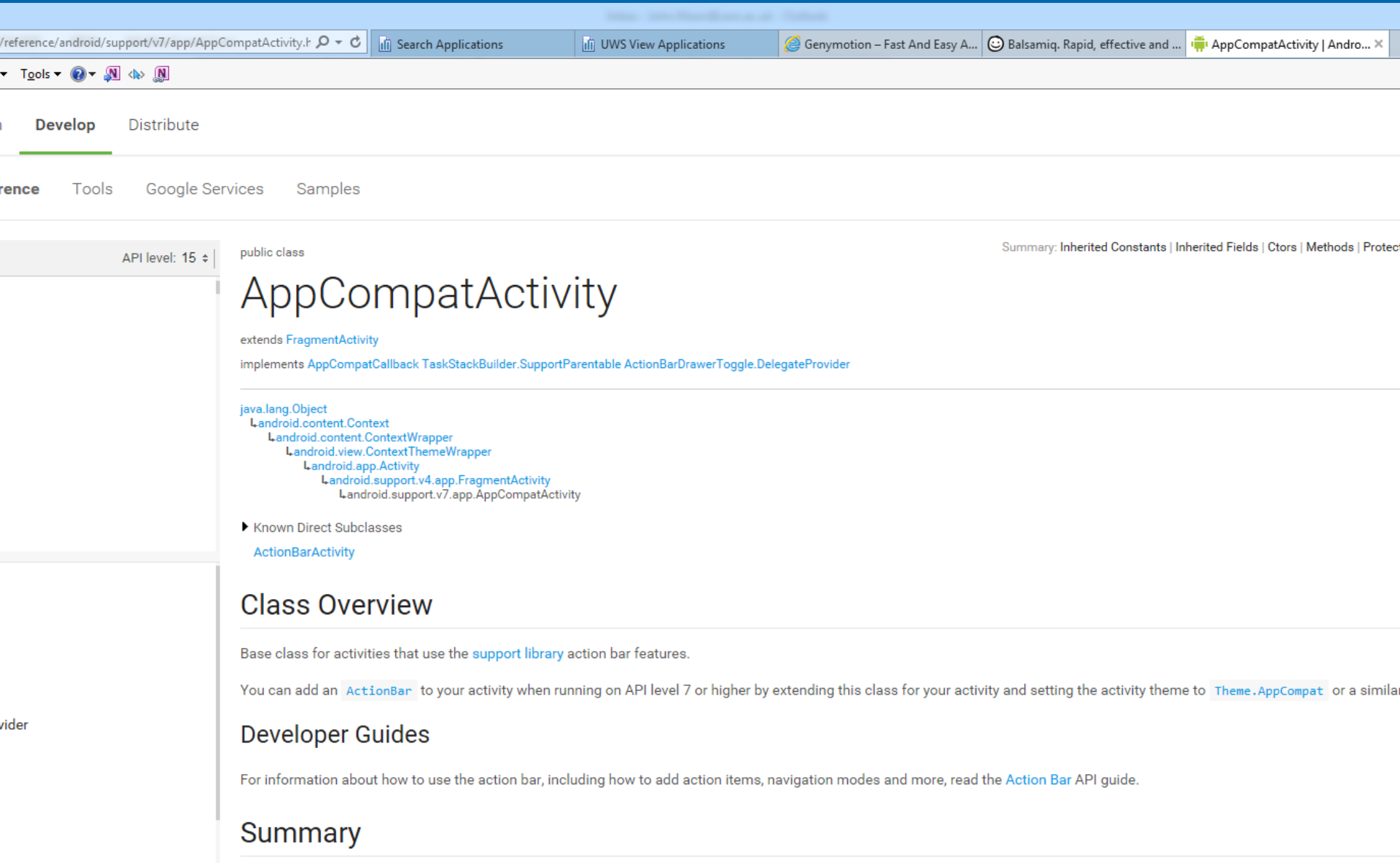
An activity is a single, focused thing that the user can do. Almost all activities interact with the user, so the Activity class t
`setContentView(View)` . While activities are often presented to the user as full-screen windows, they can also be used in
set) or embedded inside of another activity (using `ActivityGroup`). There are two methods almost all subclasses of Act

- `onCreate(Bundle)` is where you initialize your activity. Most importantly, here you will usually call `setContentView(i`
  `(int)` to retrieve the widgets in that UI that you need to interact with programmatically.

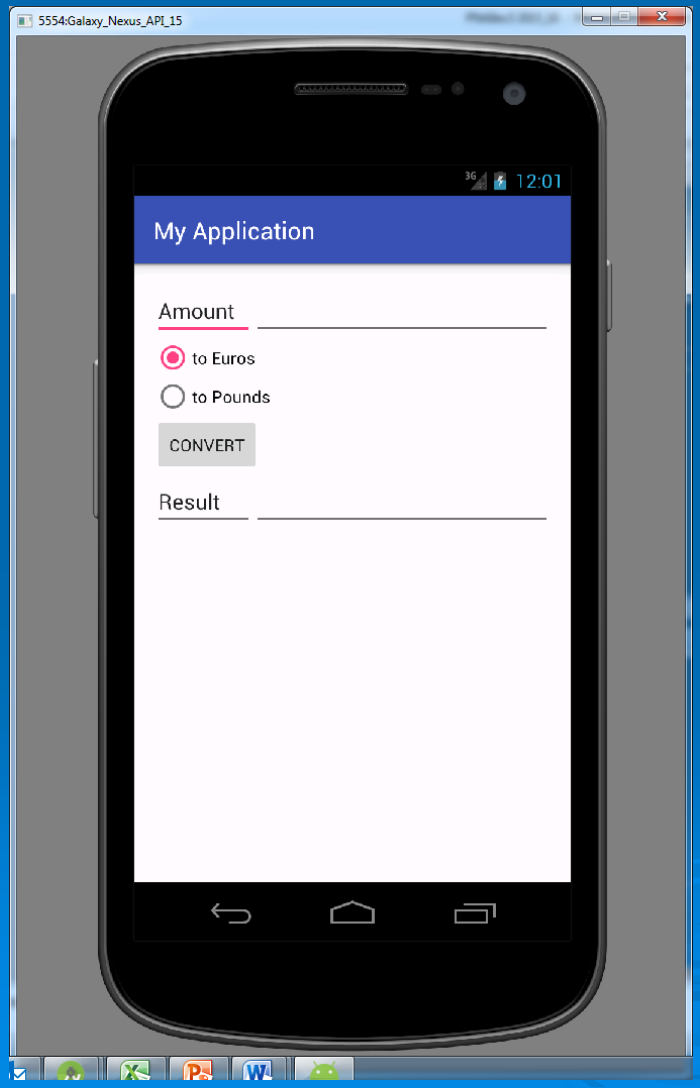
ListActivity class, extends Activity



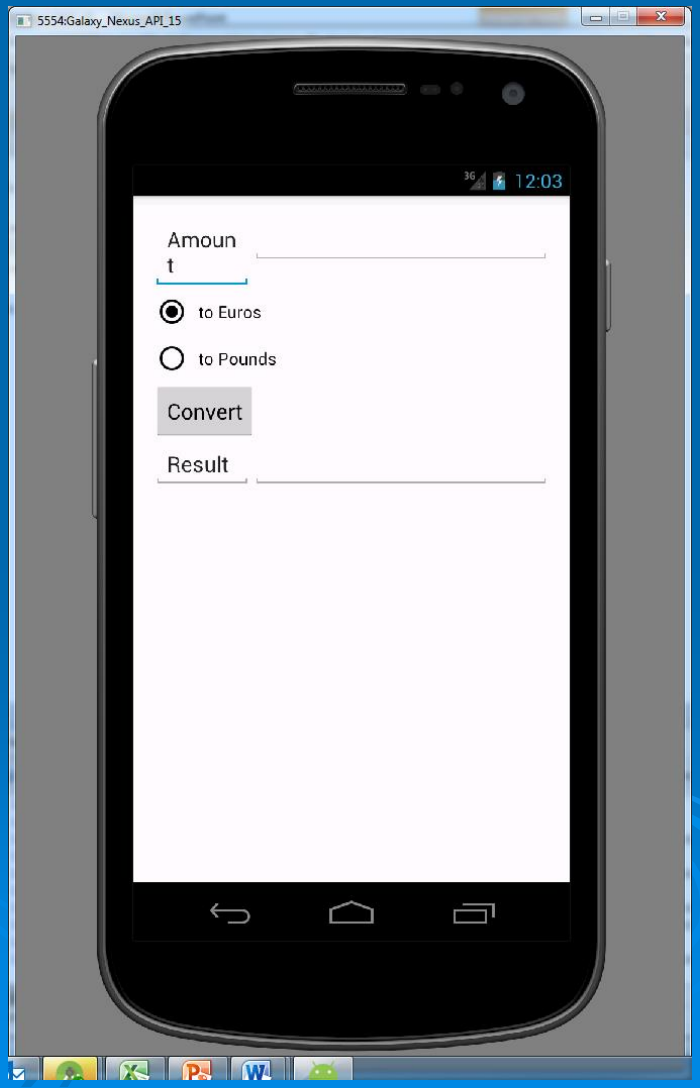
AppCompatActivity



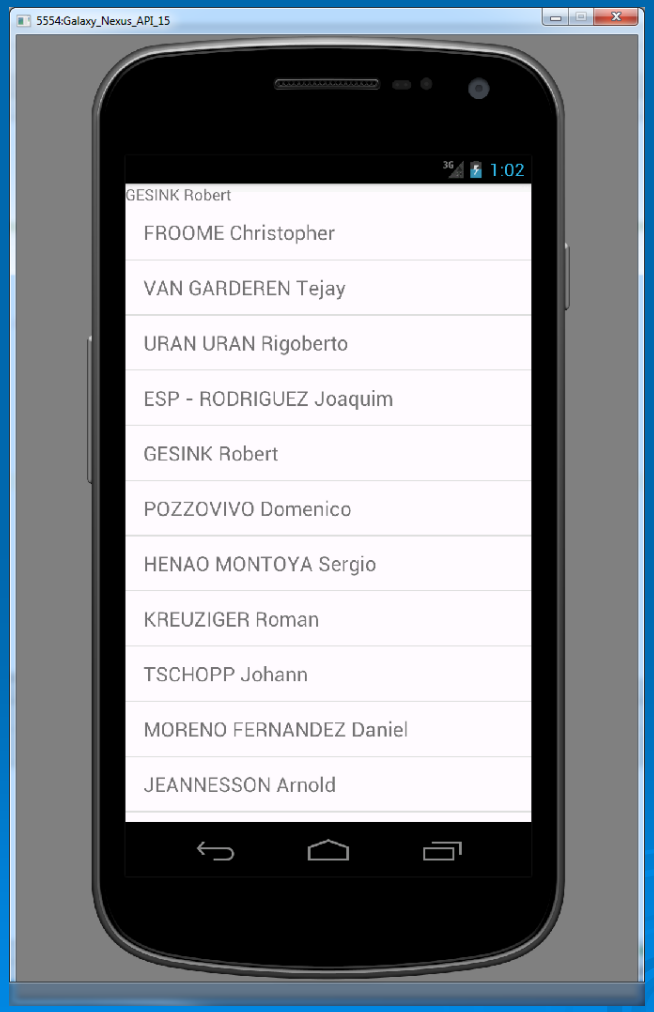
AppCompatActivity

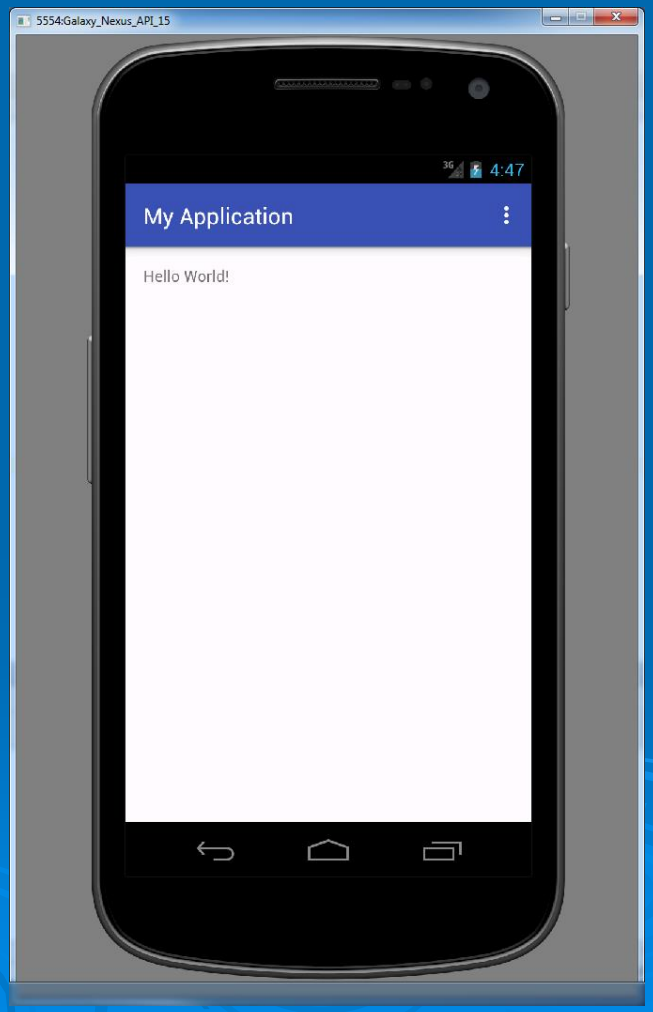


Activity



List Activity AppCompatActivity with menu





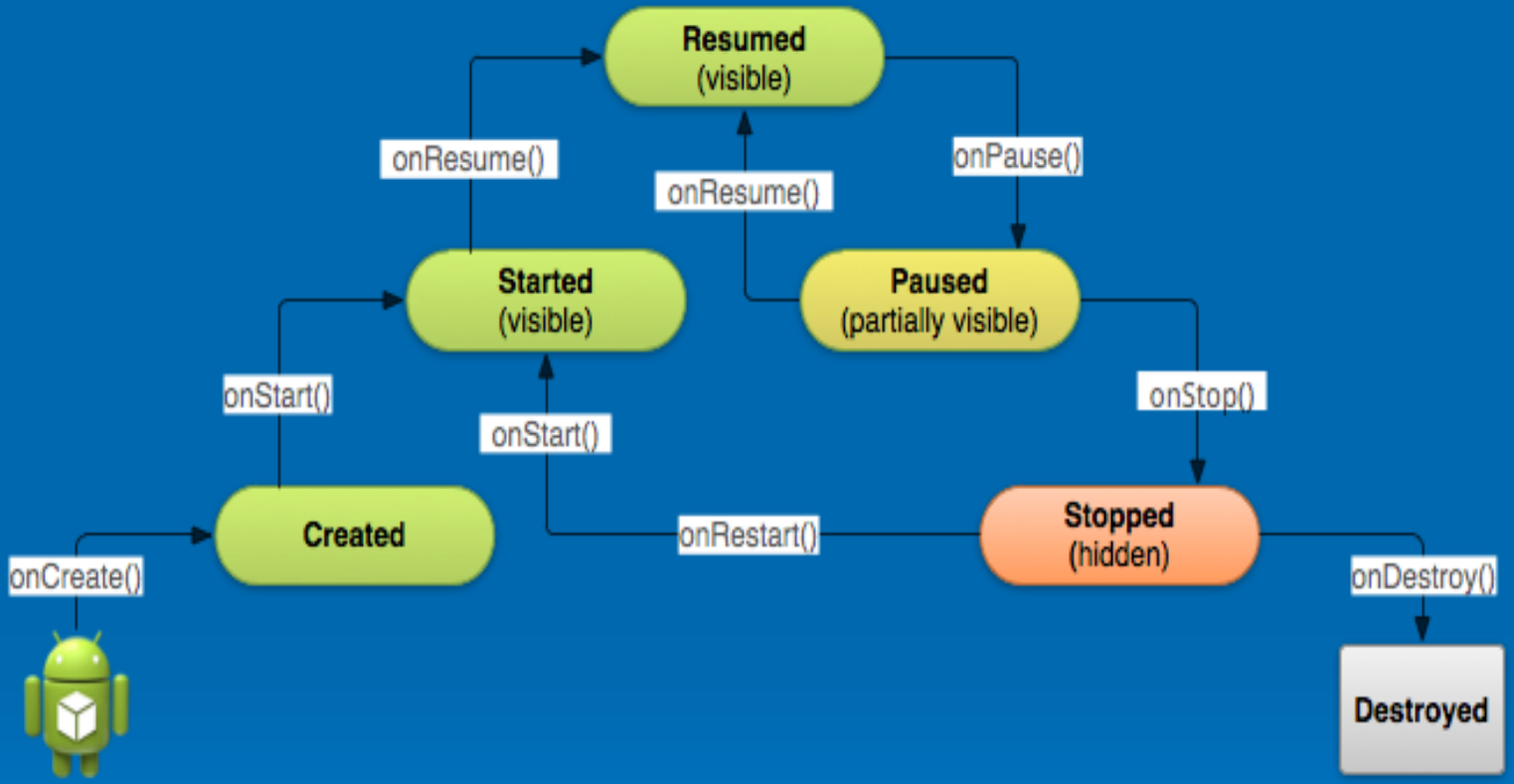
Activity Life Cycle Events

- ***Resumed (Active)*:** The activity was started by the user, is running, and is in the foreground.

- ***Paused***: The activity was started by the user, is running, and is visible,
- but a notification or something is overlaying part of the screen.
  - During this time, the user can see your activity but may not be able to interact with it (e.g. due to incoming phone call, battery low alert).

- ***Stopped*:** The activity was started by the user, is running, but is hidden
- by other activities that have been launched or switched to.

- ***Dead*:** Either the activity was never started (e.g., just after a phone
- reset) or the activity was terminated, perhaps due to lack of available
- memory.



- http://developer.android.com/training/basics/activity-lifecycle/starting.html

onCreate() and onDestroy()

- We have been implementing onCreate() in all of our Activity subclasses in all the examples. This method will be called in three situations:
  - When the activity is first started (e.g., since a system restart)
  - If the activity had been running, then sometime later was killed off, onCreate() will be invoked with the Bundle from onSaveInstanceState() as a parameter
  - If the activity had been running and you have set up your activity to have different resources based on different device states

- onDestroy() may be called when the activity is
  - shutting down, either because the activity called finish()
  - or because Android needs RAM and is closing the activity prematurely.
  - onDestroy() may not be called if the need for RAM is urgent (e.g., an incoming phone

View Class

- This class represents the basic building block for user interface components. A View occupies a rectangular area on the screen and is responsible for drawing and event handling. View is the base class for *widgets*, which are used to create interactive UI components (buttons, text fields, etc.).
- http://developer.android.com/reference/android/view/View.html

- Extends object

- Has subclasses
  - ImageView
  - SurfaceView
  - TextView
  - Etc.

- ImageView has subclass ImageButton
- TextView has subclasses Button, EditText etc
- Button has subclasses CheckBox, RadioButton etc

Numerical Input from an EditText

- In the onClick() method
- EditText weightText = (EditText)findViewById(R.id.inWeight);
- Text property
- weightText.getText()
- Convert text to String to floating point number
- float myWeight = Float.parseFloat(weightText.getText().toString());

Output in EditText

- bmiResult a floating point number
- EditText bmiResult = (EditText)findViewById(R.id.bmiResult);
- Take value of float, convert to string and put in text property of EditText
-
- bmiResult.setText(String.valueOf(bmi));