University of Anbar
College of Computer Science and Information Technology
Computer Science Department

Mobile Application Programming

Lab Three
Third Stage

First Course 2023 - 2024

Rihab Hazim Qasim

MSc Computer Science

rehz1991@uoanbar.edu.iq

Mobile Design Principles – 1. Who?

- User requirements

- ○ **UI Layouts**

- Basic building block for UI is a View object which is created from View class and occupies a rectangular area on screen and is responsible for drawing and event handling.

- View is base class for widgets, which are used to create interactive UI components like buttons, text fields, etc.

- ViewGroup is a subclass of View and provides invisible container that hold other Views or other ViewGroups and define their layout properties.

- At third level we have different layouts which are subclasses of ViewGroup class and a typical layout defines visual structure for an Android UI and can be created either at run time using View/ViewGroup objects or you can declare your layout using simple XML file main_layout.xml which is located in the res/layout folder of your project.

o   UI Layouts – Types

S.N.	Layout & Description
1	**Linear Layout** **a view group that aligns all children in a single direction, vertically or horizontally.**
2	**Relative Layout** **a view group that displays child views in relative positions.**
3	**Table Layout** **a view that groups views into rows and columns.**
4	**Absolute Layout** **enables you to specify exact location of its children.**
5	**Frame Layout** **a placeholder on screen that you can use to display a single view.**
6	**List View** **a view group that displays a list of scrollable items.**
7	**Grid View** **a ViewGroup that displays items in a two-dimensional, scrollable grid.**

UI Layouts - View Input Events

screen layout (made up of View objects, such as text and button)

- A view object may have a unique ID assigned to it which will identify the View uniquely within the tree. The syntax for an ID, inside an XML tag is:

  android:id="@+id/my_button"

- The at-symbol (@) at the beginning of the string indicates that the XML parser should parse and expand the rest of the ID string and identify it as an ID resource.

The plus-symbol (+) means that this is a new resource name that must be created and added to our resources. To create an instance of the view object and capture it from the layout, use the following:

Button myButton = (Button) findViewById(R.id.my_button);