

Example 2 A:

Develop an application that uses Layout Managers

Layout Manager (ViewGroup)

ViewGroup is a subclass of View. One or more Views can be grouped together into a ViewGroup. A ViewGroup provides the android layout in which we can order the appearance and sequence of views.

Examples of ViewGroup are LinearLayout, FrameLayout, RelativeLayout etc.

Layout Managers Types:

- **LinearLayout:** is a ViewGroup that aligns all children in a single direction, vertically or horizontally.
- **RelativeLayout:** is a ViewGroup that displays child views in relative positions
- **AbsoluteLayout:** allows us to specify the exact location of the child views and widgets
- **TableLayout:** is a view that groups its child views into rows and columns
- **FrameLayout:** is a placeholder on screen that is used to display a single view

In this example we will see the detailed explanations about LinearLayout and RelativeLayout below

LinearLayout:

LinearLayout organizes elements along a single line. We can specify whether that line is vertical or horizontal using “android: orientation”. The orientation is horizontal by default.

A vertical LinearLayout will only have one child per row (so it is a column of single elements), and a horizontal LinearLayout will only have one single row of elements on the screen.

android:layout_weight attribute depicts the importance of the element. An element with larger weight occupies more screen space.

RelativeLayout:

Android RelativeLayout lays out elements based on their relationships with one another, and with the parent container. This is one of the most complicated layout and we need several properties to actually get the layout we desire.

Using RelativeLayout we can position a view to be **toLeftOf**, **toRightOf**, **below** or **above** its siblings.

We can also position a view with respect to its parent such as **centered**, **horizontally and vertically** or both, or aligned with any of the edges of the parent RelativeLayout. If none of these attributes are specified on a child view then the view is by default rendered to the top left position.