WHAT IS ANDROID?

**Android** is a mobile OPERATING SYSTEM based on a modified version of the LINUX KERNEL and other OPEN SOURCE software, designed primarily for TOUCHSCREEN mobile devices such as smartphones and tablets. Android is developed by a consortium of developers known as the OPEN HANDSET ALLIANCE, with the main contributor and commercial marketer being GOOGLE.

INTERFACE

Based on direct manipulation, using touch inputs that loosely correspond to real-world actions, like swiping, tapping, pinching, and reverse pinching to manipulate on-screen objects, along with a [virtual keyboard](https://en.wikipedia.org/wiki/Virtual_keyboard). [Game controllers](https://en.wikipedia.org/wiki/Game_controller) and full-size physical [keyboards](https://en.wikipedia.org/wiki/Computer_keyboard) are supported via [Bluetooth](https://en.wikipedia.org/wiki/Bluetooth) or USB The response to user input is designed to be immediate and provides a fluid touch interface, often using the vibration capabilities of the device to provide [haptic feedback](https://en.wikipedia.org/wiki/Haptic_technology) to the user. Internal hardware, such as [accelerometers](https://en.wikipedia.org/wiki/Accelerometer), [gyroscopes](https://en.wikipedia.org/wiki/Gyroscope) and [proximity sensors](https://en.wikipedia.org/wiki/Proximity_sensor) are used by some applications to respond to additional user actions, for example adjusting the screen from portrait to landscape depending on how the device is oriented,[[79]](https://en.wikipedia.org/wiki/Android_(operating_system)#cite_note-79) or allowing the user to steer a vehicle in a [racing game](https://en.wikipedia.org/wiki/Racing_game) by rotating the device, simulating control of a  steering wheels.

APPLICATIONS

Preinstalled google apps comes with android devices, like GMAIL,GOOGLE MAPS,YOUTUBE,GOOGLE CHROME etc..

Apps are written using the software development kit (SDK),Java is used often which may be combined with C/C++.

WHO CREATED IT?

Android Inc. was founded in [Palo Alto, California](https://en.wikipedia.org/wiki/Palo_Alto,_California), in October 2003 by [Andy Rubin](https://en.wikipedia.org/wiki/Andy_Rubin), [Rich Miner](https://en.wikipedia.org/wiki/Rich_Miner), Nick Sears, and Chris White.

In July 2005, [Google](https://en.wikipedia.org/wiki/Google) acquired Android Inc. for at least $50 million.

**WHAT ARE ANDROID APPS?**

Software application running on android platform designed for mobile devices,

Allowing us to perform specific tasks.These are coded in java, c++ and Kotlin languages.

SOFTWARE USED IN THE DEVELOPMENT

Android apps can be written in java, Kotlin and C++ languages, using the SDK.  All non-JVM languages, such as [Go](https://en.wikipedia.org/wiki/Go_(programming_language)), [JavaScript](https://en.wikipedia.org/wiki/JavaScript), [C](https://en.wikipedia.org/wiki/C_(programming_language)), C++ or [assembly](https://en.wikipedia.org/wiki/Assembly_language), need the help of JVM language code, that may be supplied by tools, likely with restricted API support.

ANDROID SDK is a software development kit consisting of debugger ,libraries, emulator based on QEMU(quick emulator) ,documentation, sample code and tutorials.

Android apps can be built using a number of alternative languages and IDEs

Some of the platforms are\_\_\_

* Ruboto (Ruby)
* [Xamarin 2.0](http://www.xamarin.com/) (C#)
* [Basic4android](http://www.basic4ppc.com/) (Basic)
* [Appcelerator Titanium](http://www.appcelerator.com/) (HTML/Javascript)
* [ItelliJ IDEA](http://www.jetbrains.com/idea/) (Altenrative IDE, Java)
* [Scripting Layer For Android](http://code.google.com/p/android-scripting/) (Python, Perl, etc.)
* [AppInventor](http://appinventor.mit.edu/) (Drag and Drop)
* [Adobe Flash](http://www.adobe.com/products/flash.html) (Flash/AIR)

Languages used in the development

Java, the most used language and also the official language for android app development

KOTLIN, cross-platform language can be used as an alternative to java

C++

C#

PYTHON

INSTI APP

API in Django

It uses JAVA

ACTIVITY CYCLE OF A BASIC ANDROID APPLICATION

An activity is a single focused thing which user can do. Almost all activities interact with the user, so activity class creates a window to place UI with  [setContentView(View)](https://developer.android.com/reference/android/app/Activity" \l "setContentView(android.view.View)). While activities are often presented to the user as full-screen windows, they can also be used in other ways: as floating windows (via a theme with [R.attr.windowIsFloating](https://developer.android.com/reference/android/R.attr" \l "windowIsFloating) set), [Multi-Window mode](https://developer.android.com/guide/topics/ui/multi-window) or embedded into other windows. There are two methods almost all subclasses of Activity will implement:

* [onCreate(Bundle)](https://developer.android.com/reference/android/app/Activity#onCreate(android.os.Bundle)) is where you initialize your activity. Most importantly, here you will usually call [setContentView(int)](https://developer.android.com/reference/android/app/Activity" \l "setContentView(int)) with a layout resource defining your UI, and using [findViewById(int)](https://developer.android.com/reference/android/app/Activity" \l "findViewById(int)) to retrieve the widgets in that UI that you need to interact with programmatically.
* [onPause()](https://developer.android.com/reference/android/app/Activity#onPause()) is where you deal with the user pausing active interaction with the activity. Any changes made by the user should at this point be committed (usually to the [ContentProvider](https://developer.android.com/reference/android/content/ContentProvider) holding the data). In this state the activity is still visible on screen.

setContentView(View)

It sets the activity content to an explicit view.  When calling this method, the layout parameters of the specified view are ignored. Both the width and the height of the view are set by default to [ViewGroup.LayoutParams#MATCH\_PARENT](https://developer.android.com/reference/android/view/ViewGroup.LayoutParams" \l "MATCH_PARENT). To use our own layout parameters, invoke [setContentView(android.view.View, android.view.ViewGroup.LayoutParams)](https://developer.android.com/reference/android/app/Activity" \l "setContentView(android.view.View,%20android.view.ViewGroup.LayoutParams)) instead.

ACTIVITY LIFE CYCLE

As the user, navigates through the app Activity instances in the app transition through different stages in their lifecycle. Activity class provides a number of callbacks, so the activity knows that the state has changed. that the system is creating, stopping, or resuming an activity, or destroying the process in which the activity resides.

LIFECYCLE METHODS AND CALLBACKS

In general, activity lifecycle has 7 callbacks

1. onCreate()
2. onStart()
3. onResume()
4. onPause()
5. onStop()
6. onRestart()
7. onDestroy()



Android initiates the program within an activity with a call to  *onCreate method*

1. **onCreate()-activity is created**
2. **onStart()- callback method is called, activity becomes visible to the user.**
3. **onResume()-activity is in the foreground and user can interact with it**
4. **onPause()-activity is partially obscured by another activity. Another activity that’s in the foreground is semi-transparent.**
5. **onStop()-activity completely hidden and is not visible to the user.**
6. **onRestart()-from the stopped state, the activity either stops running and goes away or it comes back to interact with the user. If it comes back, then this method is invoked.**
7. **onDestroy()- activity is destroyed and removed from the memory.**

**DIFFERENT UI ELEMENTS IN AN ANDROID APP**

Following are the commonly used UI or input controls in android applications.

* [TextView](https://www.tutlane.com/tutorial/android/android-textview-with-examples)-In android, **TextView** is a user interface control that is used to display the text to the user.
* [EditText](https://www.tutlane.com/tutorial/android/android-edittext-with-examples)-allows the user to modify the text
* [AutoCompleteTextView](https://www.tutlane.com/tutorial/android/android-autocompletetextview-with-examples)-used to show the list of suggestions based on the user typing text.
* [Button](https://www.tutlane.com/tutorial/android/android-button-with-examples)-interface control used to perform an action when the user clicks or taps on it
* [ImageButton](https://www.tutlane.com/tutorial/android/android-imagebutton-with-examples)- used to display a button with an image to perform an action when the user clicks or tap on it.
* [ToggleButton](https://www.tutlane.com/tutorial/android/android-toggle-button-with-examples)-used to display on or off states as a light indicator
* [CheckBox](https://www.tutlane.com/tutorial/android/android-checkbox-with-examples)-two states button that can either be checked or unchecked
* [RadioButton](https://www.tutlane.com/tutorial/android/android-radiobutton-with-examples)-two-states button that can either be checked or unchecked and cannot be unchecked once it is checked

[RadioGroup](https://www.tutlane.com/tutorial/android/android-radiogroup-with-examples)-used to group one or more radio buttons into separate groups based on our requirements.

In case if we group radio buttons using the radio group, at a time only one item can be selected from the group of radio buttons.

* [ProgressBar](https://www.tutlane.com/tutorial/android/android-progressbar-with-examples)-indicates the progress of an operation
* [Spinner](https://www.tutlane.com/tutorial/android/android-spinner-dropdown-list-with-examples)-drop-drown list which allows the user to select one value from the list
* [TimePicker](https://www.tutlane.com/tutorial/android/android-timepicker-with-examples)-widget for selecting time of the day,24 hour format
* [DatePicker](https://www.tutlane.com/tutorial/android/android-datepicker-with-examples)-widget for selecting the date
* [SeekBar](https://www.tutlane.com/tutorial/android/android-seekbar-with-examples)-
* [AlertDialog](https://www.tutlane.com/tutorial/android/android-alertdialog-with-examples#divaldg)-
* [Switch](https://www.tutlane.com/tutorial/android/android-switch-on-off-button-with-examples)-two state on or off thumb slider
* [RatingBar](https://www.tutlane.com/tutorial/android/android-ratingbar-with-examples)

JAVA

Object oriented programming language-follows the basic principles of oops

ABSTRACTION, ENCAPSULATION, POLYMORPHISM AND INHERITANCE

Robust, secure, supports functional programming ,multi-threaded

Platform independent and portable language

KOTLIN

Main is a top level function i.e. kotlin functions do not need to be nested within a class

Clean, compact system

Single type system

Null safety

Functions and functional programming

Data classes

Extensions

Operator overloading

HOW ARE THEY SIMILAR TO C++?

JAVA

Both java and C++ are high level object oriented programming languages which obeys its principles like encapsulation, data abstraction and polymorphism(doing one thing in more than one ways) and inheritance and they both support the concepts of classes and objects. In both of them,the statements terminate with a semicolon and the basic switch ,if-else constructs remains the same. Both allows us to program using functions. Like C++ , the entry point to a java program is a function named “main”.

KOTLIN

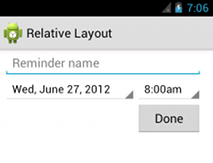
Kotlin also supports the classes and methods of OOP. In addition to it, it also supports procedural programming. It is also a statically typed programming language which means that variables need not be defined before they are used and variables can be initialised anywhere in the program as per the requirement. Like Java and C++ ,the entry point to a kotlin program is through a function named “main”.

**RELATIVE AND LINEAR LAYOUTS**

RELATIVE LAYOUT

A view group that displays views in relative positions. The position of each view can be specified as relative to sibling elements (such as to the left-of or below another view) or in positions relative to the parent RelativeLayout  area (such as aligned to the bottom, left or center).

A RelativeLayout  is a very powerful utility for designing a user interface because it can eliminate nested view groups and keep your layout hierarchy flat, which improves performance. If you find yourself using several nested LinerarLayout  groups, you may be able to replace them with a single RelativeLayout.

****

LINEAR LAYOUTS

Linear Layout is a view group that aligns all children in a single direction , vertically or horizontally . The layout direction can be specified using the android:orientation attribute .

All children of a [LinearLayout](https://developer.android.com/reference/android/widget/LinearLayout) are stacked one after the other, so a vertical list will only have one child per row, no matter how wide they are, and a horizontal list will only be one row high (the height of the tallest child, plus padding). A [LinearLayout](https://developer.android.com/reference/android/widget/LinearLayout) respects *margin*s between children and the *gravity* (right, center, or left alignment) of each child.

LAYOUT WEIGHT

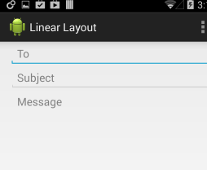
[LinearLayout](https://developer.android.com/reference/android/widget/LinearLayout) also supports assigning a *weight* to individual children with the [android:layout\_weight](https://developer.android.com/reference/android/widget/LinearLayout.LayoutParams" \l "attr_android:layout_weight) attribute. This attribute assigns an "importance" value to a view in terms of how much space it should occupy on the screen. A larger weight value allows it to expand to fill any remaining space in the parent view. Child views can specify a weight value, and then any remaining space in the view group is assigned to children in the proportion of their declared weight. Default weight is zero

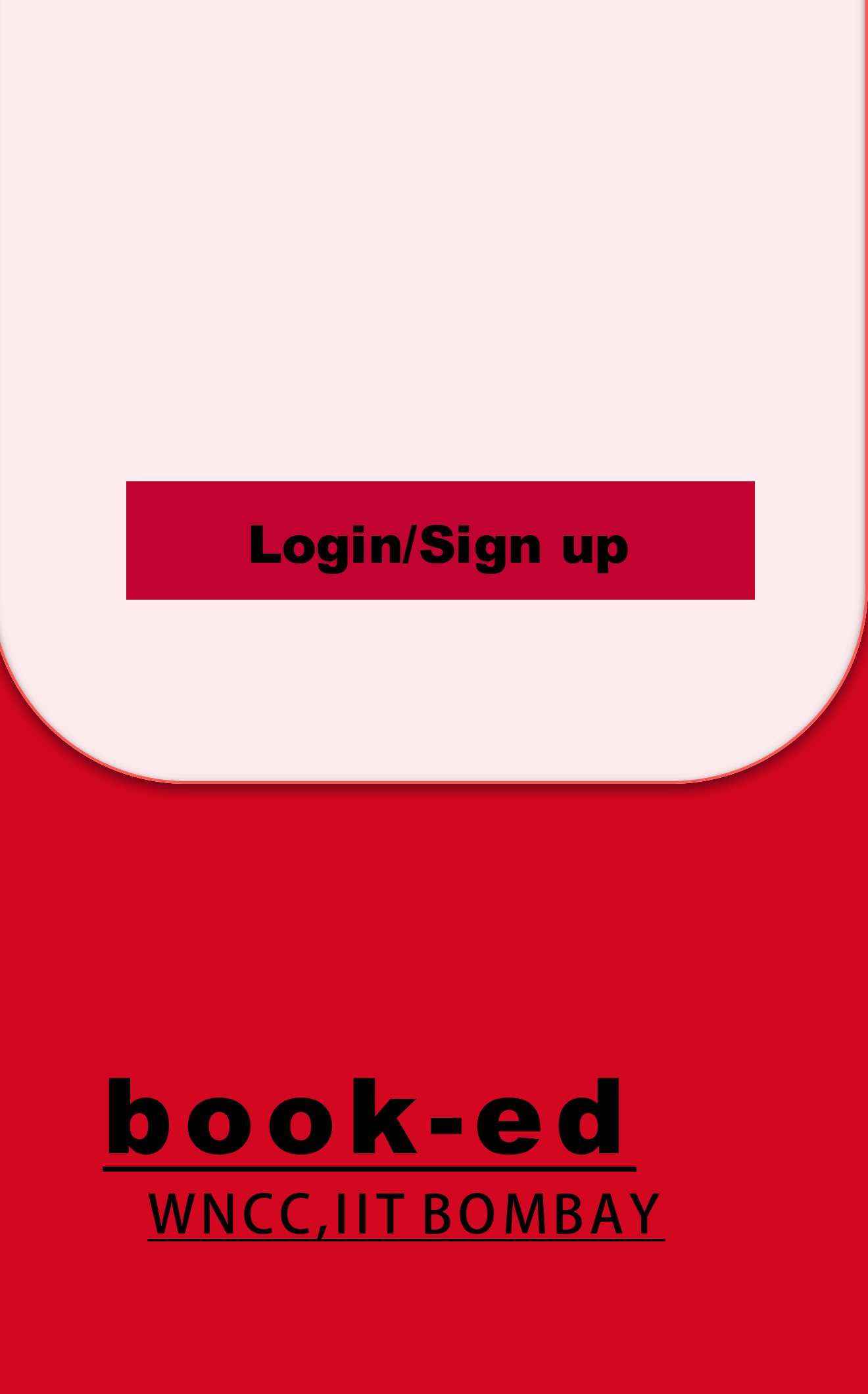
EQUAL DISTRIBUTION

To create a linear layout in which each child uses the same amount of space on the screen, the [android:layout\_height](https://developer.android.com/reference/android/view/ViewGroup.LayoutParams" \l "attr_android:layout_height) of each view is set to "0dp" (for a vertical layout) or the [android:layout\_width](https://developer.android.com/reference/android/view/ViewGroup.LayoutParams" \l "attr_android:layout_width) of each view to "0dp" (for a horizontal layout) and the [android:layout\_weight](https://developer.android.com/reference/android/widget/LinearLayout.LayoutParams" \l "attr_android:layout_weight) of each view is set to "1

UNEQUAL DISTRIBUTION

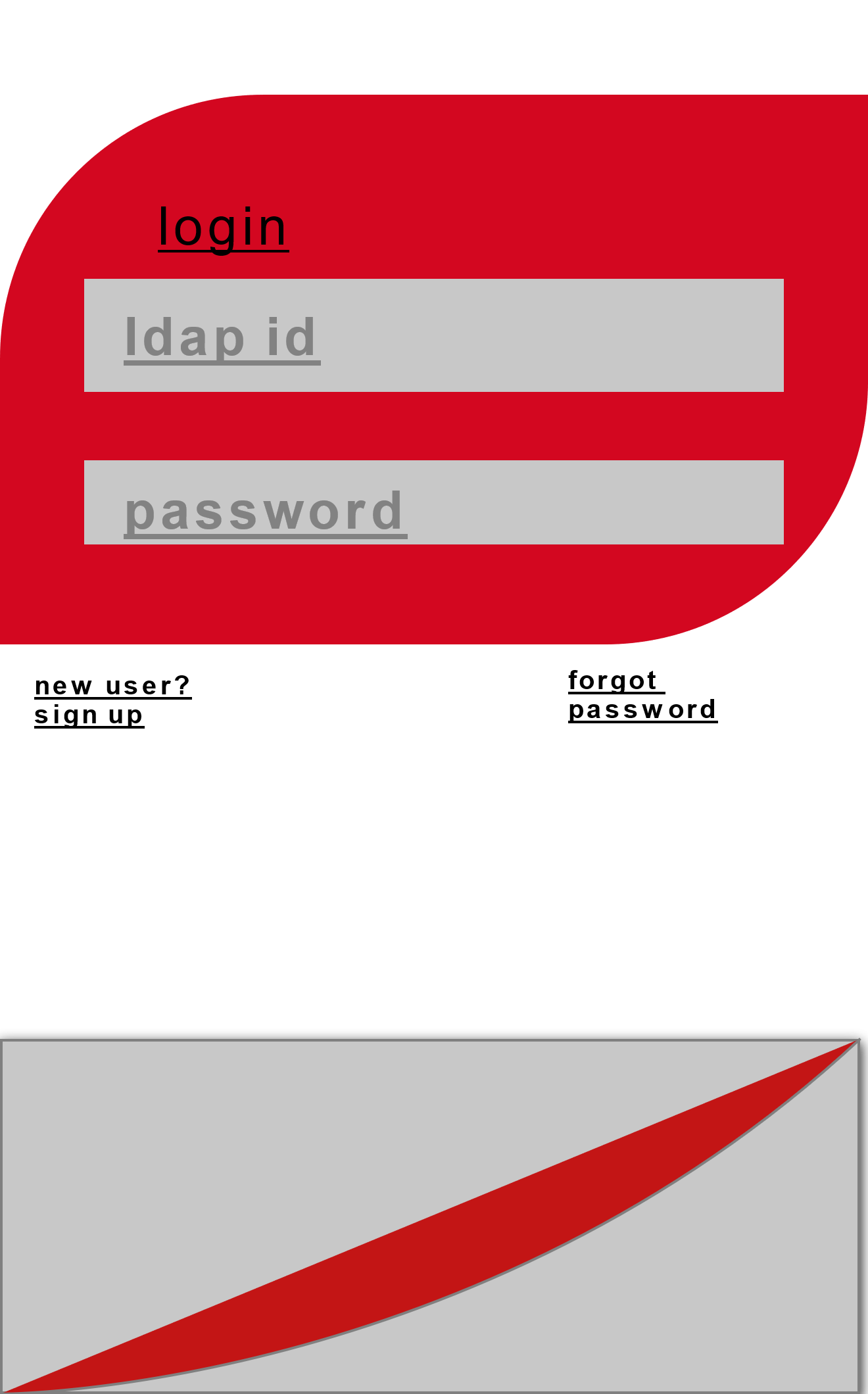
In case of assigning one view more importance over the other,the above mentioned parameters can be varied

****



**OPENING SCREEN**

**LOGIN/SIGN UP-a button which performs the task of taking the user to login or sign up window when user clicks on it.**



**LOGIN SCREEN**

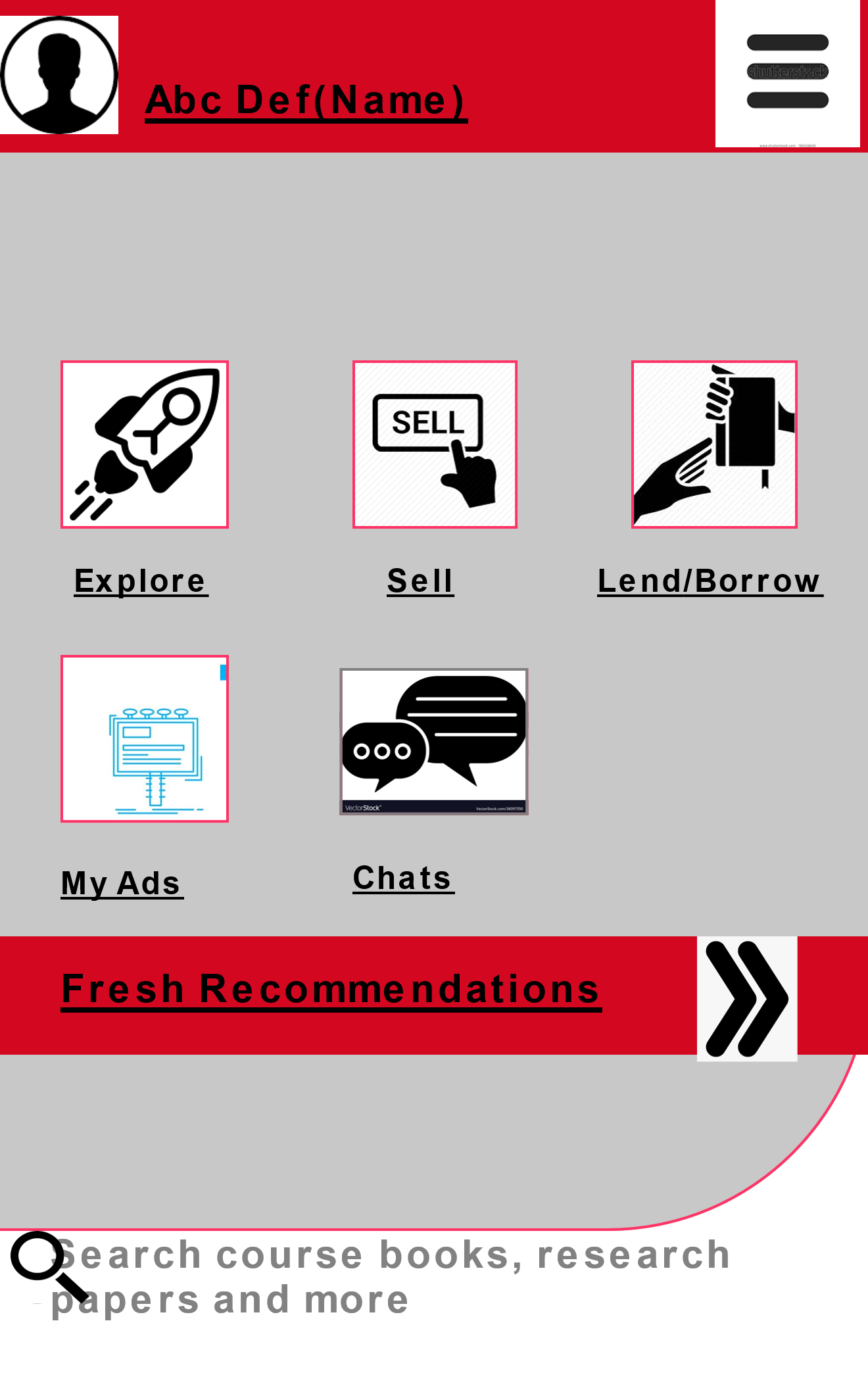
**A linear layout pattern which asks the user to enter his/her LDAP id and password to enable the user to use the app**

**The UI element here is textView that displays text to the user as well as editText which allows the user to modify the text. Then there are two buttons SIGN UP and FORGOT PASSWORD which performs the respective operations of taking the user to sign up window and if it is an existing user and has forgotten the password then progresses towards resetting the password.**



**SIGN UP SCREEN**

**If the user is new then after clicking on the SIGN UP button on the previous screen , user is required to sign up with the essential information required . The UI elements here are textView ,editText and SIGN UP BUTTON**



The UI elements here are IMAGE BUTTON, respectively explore, sell ,lend/borrow, my ads and chats. Fresh recommendations let the user see the recommendations and the search tab allows the user to search their respective interests