

# Mobile Application Development

- **UNIT-I: Introduction to Android OS: Android Description – Open Handset Alliance – Android. Ecosystem – Android versions – Android Activity – Features of Android – Android Architecture – Stack Linux Kernel. Configuration of Android Environment: Operating System – Java JDK – Android SDK – Android Development Tools (ADT) – Android Virtual Devices (AVDs) – Emulators - Dalvik Virtual Machine – Differences between JVM and DVM – Steps to Install and Configure Eclipse and SDK. (Chapters 1 & 2)**
- **UNIT-II: Create the first android application: Directory Structure. Android User Interface: Understanding the Components of a screen– Linear Layout – Absolute Layout – Frame. Layout – Relative Layout – Table Layout. (Chapters 3 & 4)**
- **UNIT-III: Designing User Interface with View – Text View – Button – Image Button – Edit Text – Check Box – Toggle Button – Radio Button and Radio Group – Progress Bar – Auto complete Text View – Spinner – List View – Grid View – Image View - Scroll View – Custom Toast – Alert – Time and Date Picker. (Chapter 5)**
- **UNIT-IV: Activity: Introduction – Intent – Intent filter – Activity life cycle – Broadcast life cycle – Service. Multimedia: Android System Architecture – Play Audio and Video – Text to Speech. (Chapters 6 & 7)**
- **UNIT-V: SQLite Database in Android: SQLite Database – Creation and Connection of the database – Transactions. Case Study: SMS Telephony and Location Based Services. (Chapters 8, 9, & 10)**

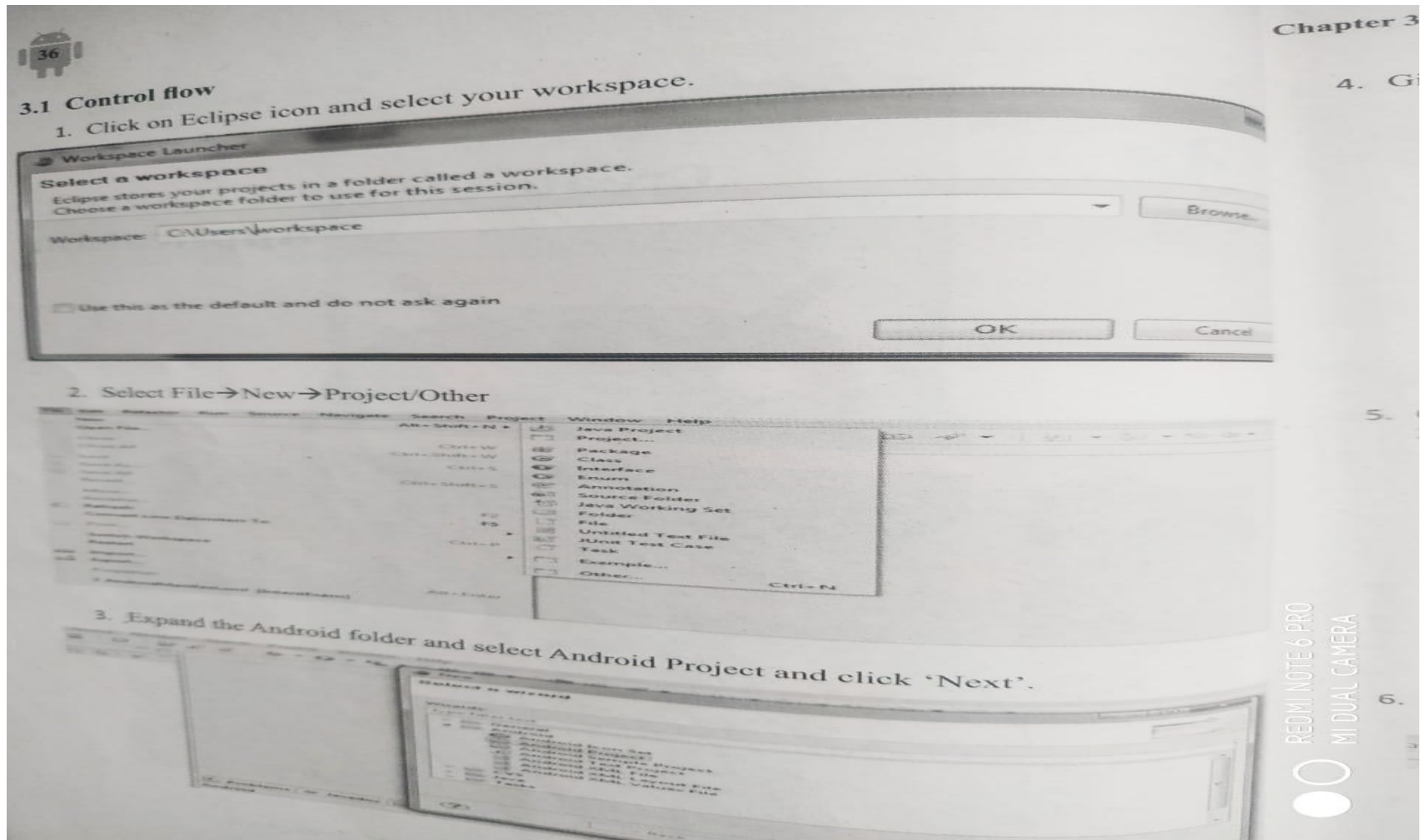
#### **TEXT BOOK**

1. Prasanna Kumar Dixit, "Android", Vikas Publishing House Private Ltd., Noida, 2014.

#### **REFERENCE BOOKS**

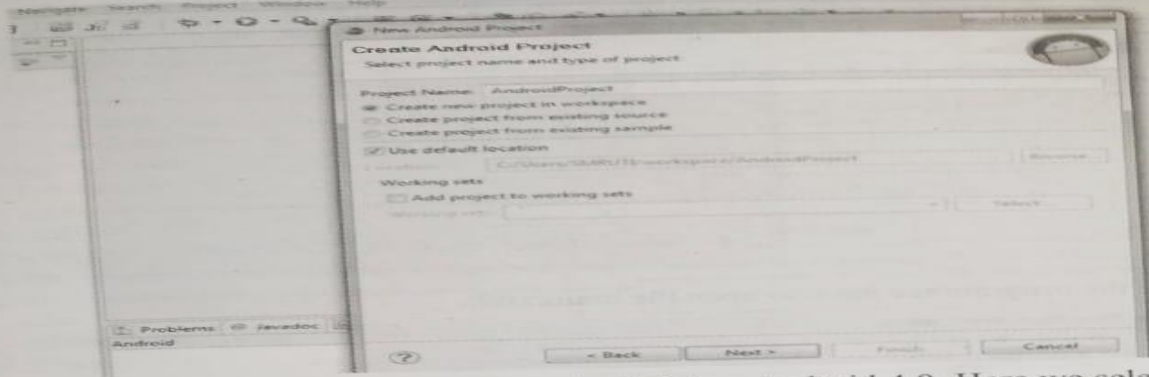
1. Reto Meier and Wrox Wiley, "Professional Android 4 Application Development", 2012.
2. ZiguradMednieks, LaridDornin, G.BlakeMeike, Masumi Nakamura, "Programming Andriod", O'Reilly,2013.
3. Robert Green, Mario Zechner, "Beginning Android 4 Games Development", Apress Media LLC, New York, 2011

# Create the first Android Application

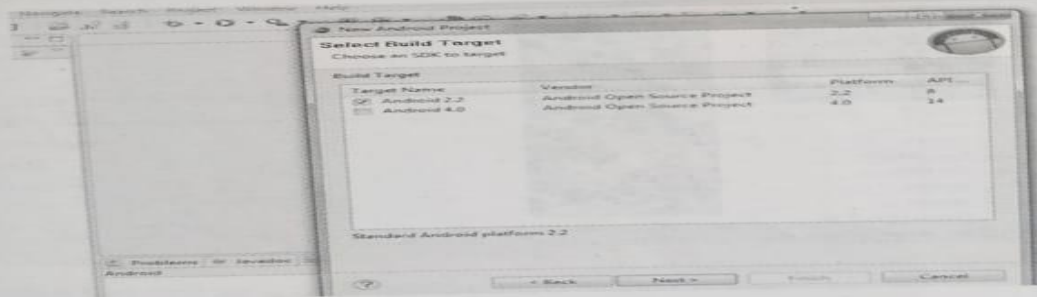


### Chapter 3 : Create the First Android Application

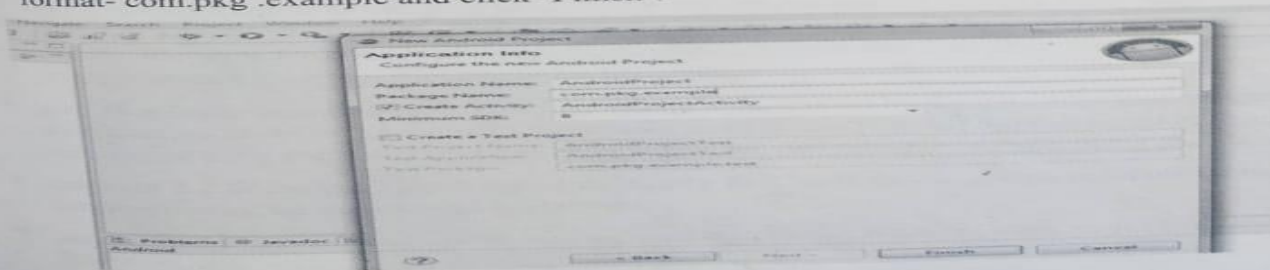
4. Give the Project Name and click 'Next'.



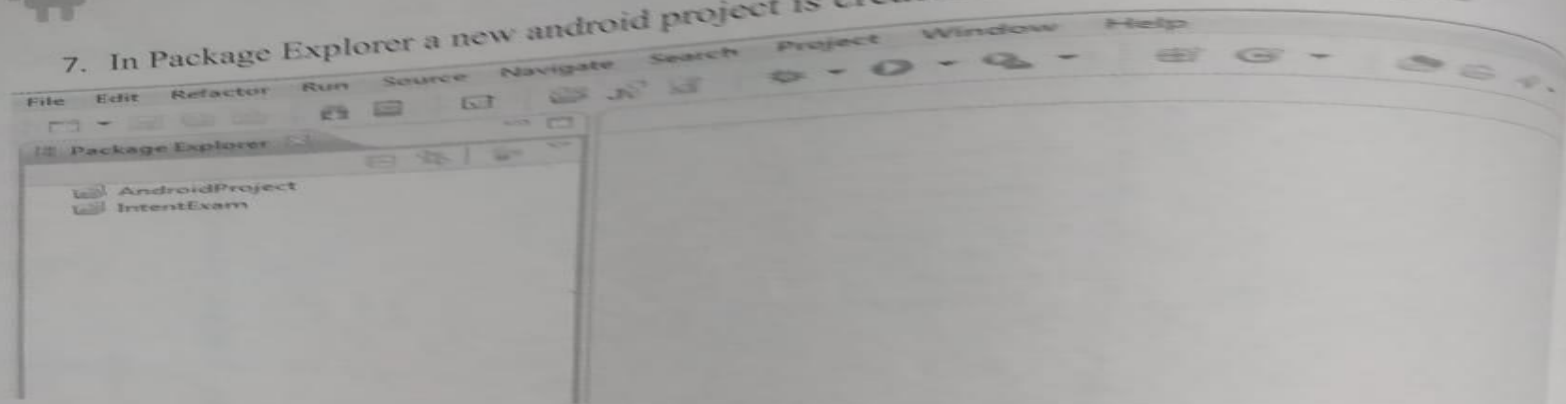
5. Choose your SDK to target whether Android 2.2 or Android 4.0. Here we select the Android 2.2 SDK and click 'Next'.



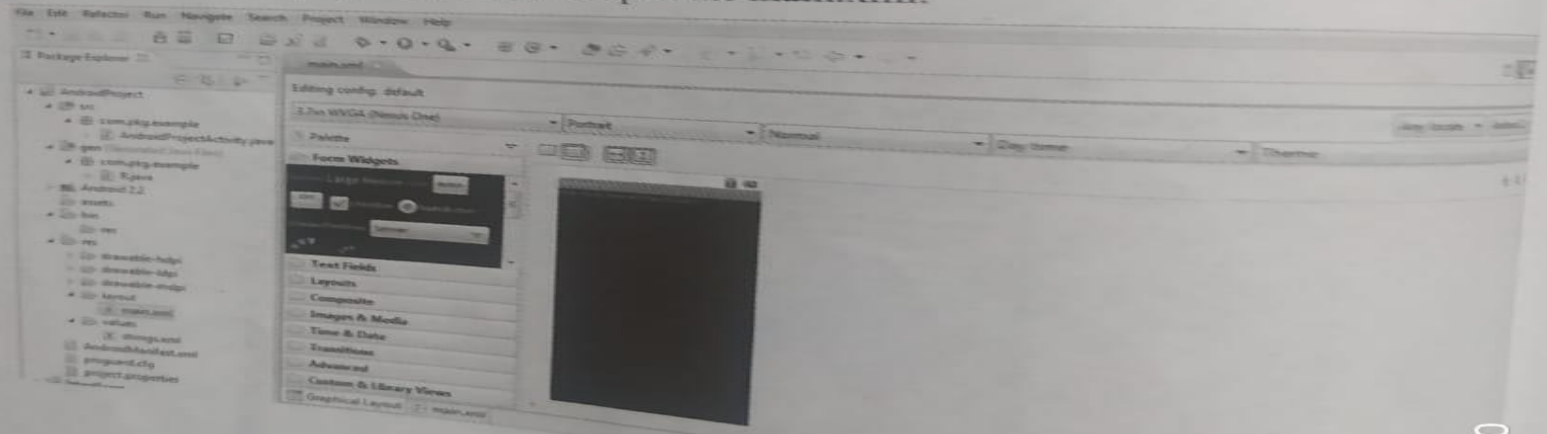
6. To configure the Android Project we have to give the Package Name as your wish in this format- com.pkg .example and click 'Finish'.



7. In Package Explorer a new android project is created named AndroidProject



8. To write the program we have to open the main.xml.



REDMI NOTE 6 PRO  
MI DUAL CAMERA



# Directory Structure

Package Explorer → Project Name / Application Name

\* AndroidProject

\* src → source

\* com.pkg.example → Package Name

\* AndroidProjectActivity.java → class Name

\* gen (Generated Java files)

\* com.pkg.example

\* R.java → Resource class name

\* Android 2.2

\* android.jar → .JAR file

\* assets → Audio, Video, Image etc.

\* bin → Contains .dex file, .apk file etc.

\* res - resource file like to store pictures, XML files etc.

\* AndroidProject.apk

\* classes.dex

\* resources.apk

\* res

\* drawable - hdpi → stores only images  
\* ic\_launcher.png

\* drawable - ldpi

\* drawable - mdpi

\* layout

\* main.xml → Different layout, UI components

\* values

\* AndroidManifest.xml → Contains string.xml file which contains string/color/dimension value etc.

\* proguard.cfg

\* project.properties

↳ free java cl optimized, detects & removes unused classes, fields, methods & attributes.

# Android user interface

- Understanding the components of a screen
  - Fundamental UI design
    - Views
    - View groups
    - Fragments
    - Activities

- Layout manager
  - Types of layout
    - Linear Layout
    - Absolute Layout
    - Table Layout
    - Frame Layout
    - Relative Layout



- Linear layout

- Linear layout is further divided into horizontal and vertical layout. It means it can arrange views in a single column or in a single row. Here is the code of linear layout(vertical) that includes a text view.

- ```
<?xml version="1.0" encoding="utf-8"?> <LinearLayout
xmlns:android="http://schemas.android.com/apk/res/an
droid" android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:orientation="vertical" > <TextView
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:text="@string/hello" /> </LinearLayout>
```

- An Absolute Layout lets you specify exact locations (x/y coordinates) of its children. Absolute layouts are less flexible and harder to maintain than other types of layouts without absolute positioning.

- `<AbsoluteLayout`

```
xmlns:android="http://schemas.android.com/apk/res/  
android" android:layout_width="fill_parent"  
android:layout_height="fill_parent"> <Button  
android:layout_width="100dp"  
android:layout_height="wrap_content"  
android:text="OK" android:layout_x="50px"  
android:layout_y="361px" /> <Button  
android:layout_width="100dp"  
android:layout_height="wrap_content"  
android:text="Cancel" android:layout_x="225px"  
android:layout_y="361px" /> </AbsoluteLayout>
```

- Android TableLayout going to be arranged groups of views into rows and columns. You will use the <TableRow> element to build a row in the table. Each row has zero or more cells; each cell can hold one View object.
- FrameLayout is designed to block out an area on the screen to display a single item. Generally, FrameLayout should be used to hold a single child view, because it can be difficult to organize child views in a way that's scalable to different screen sizes without the children overlapping each other. You can, however, add multiple children to a FrameLayout and control their position within the FrameLayout by assigning gravity to each child, using the [android:layout\\_gravity](#) attribute.

- Android RelativeLayout enables you to specify how child views are positioned relative to each other. The position of each view can be specified as relative to sibling elements or relative to the parent.