Technomania App

A Project Report of Capstone Project – 2
Submitted by
Akanksha Chaudhary
(1613101070/ 16SCSE114063)

In partial fulfillment for the award of the degree
Of
BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING WITH
SPECIALIZATION OF
COMPUTER NETWORKS AND CYBER SECURITY
SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

Under the Supervision of
Dr.Shrddha Sagar,
Associate Professor,
School of CSE,
APRIL / MAY- 2020

GALGOTIAS
UNIVERSITY

SCHOOL OF COMPUTING AND SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

Certified that this project report “Technomania Application” is the bonafide work of “Akanksha Chaudhary (1613101070) who carried out the project work under the supervision.

SIGNATURE OF PANEL

MR.B. MALLIKARJUNA
Associate Professor,
School of Computing Science
& Engineering

SIGNATURE OF GUIDE

DR. SHRDDHA SAGAR
Associate Professor,
School of Computing Science
& Engineering

Table of Contents

1. Title of the Project:

TECHNOMANIA

2. Abstract of the project:

The main objective of the project is to provide people a handy android application through which people can access all type of cyber related news and information. Through this application, any user can gain knowledge of cyber world with just one click ahead. User does not have to visit multiple sites for different cyber related information. Every information is going to be at one place.

Existing Scenario:

In the current scenario, there is no single platform (in application) present right now which provide cyber information at one point. Users have to visit different websites to gather the news related to cyber world. Many people do not have the time to visit different sites to gather the information. Ultimately, this would be the waste of time and efforts. Visiting different websites, user might get the redundancy in the information.

Proposed Scenario:

The proposed scenario is to develop an android application which will eliminate the problems faced in the current scenario. This application will provide all the information and news related to cyber world at one place. So it will save time and efforts of the users by making the proposed scenario more efficient.

Using this application will terminate the possibility of information redundancy.

3. Introduction:

3.1 Overall Description:

The application here is an application which provides users a cross-site platform to access all the information about the Cyber World. This application is made for Android devices that work on Android version 6.0 & above. This application is all about getting people closer to the world of Cyber Security.

This application can be used by anyone who is interested in the field of Cyber Security starting from a kid to highly educated professionals that work in the Cyber World. It not only provides news but also saves the time that a user has to spend on internet searching for news related to the Cyber World.

3.2 Purpose:

The main purpose of proposing this application is:

- To reduce the efforts of the people searching for Cyber news.
- To save their time.
- Increasing efficiency.
- Keeping people updated.
- Reducing redundancy of a particular news.
- Providing a handy application that could work on Android devices.

3.3 Motivations & Scopes:

As we all know news is a very important part of our lives. It gives us the important information of the events happening around us. News now-a-days is presented in various forms available like in hardcopy as newspapers, in audio on Radios or FMs generally available in our homes and cars, in videos on TVs, live channels or mobile applications where we can choose which news we want to know more about through the headlines.

The news around us can belong to a criminal case, politics and many more topics. But as we see around we see that there are very less platforms that provide complete news related to the Cyber world. The main motivation is to provide people a single platform that could provide them all news dedicated to cyber world only.

4. Software Requirement Specification:

4.1 Hardware configuration:

Below mentioned are the minimum requirements by the phone:

CPU	Octa-core 1.3 GHz and above
RAM	2GB and Above
Display	1080*1920 Resolution,16:9 ratio and above
Internal memory	8 GB & above
Chipset	Mediatek, Qualcomm and others
Size	5.5 inches (401 ppi)

4.2 Software Requirements/ Technologies used:

Operating system	Android 6.0(Marshmallow) and above
Coding Language	Java, XML
Java Platform	JRE1.8.0_202 amd64 and JDK 64-bit Server VM by Jetbrains
IDE	Android Studio 3.5.2
Emulator	AVD manager

5. Literature Survey:

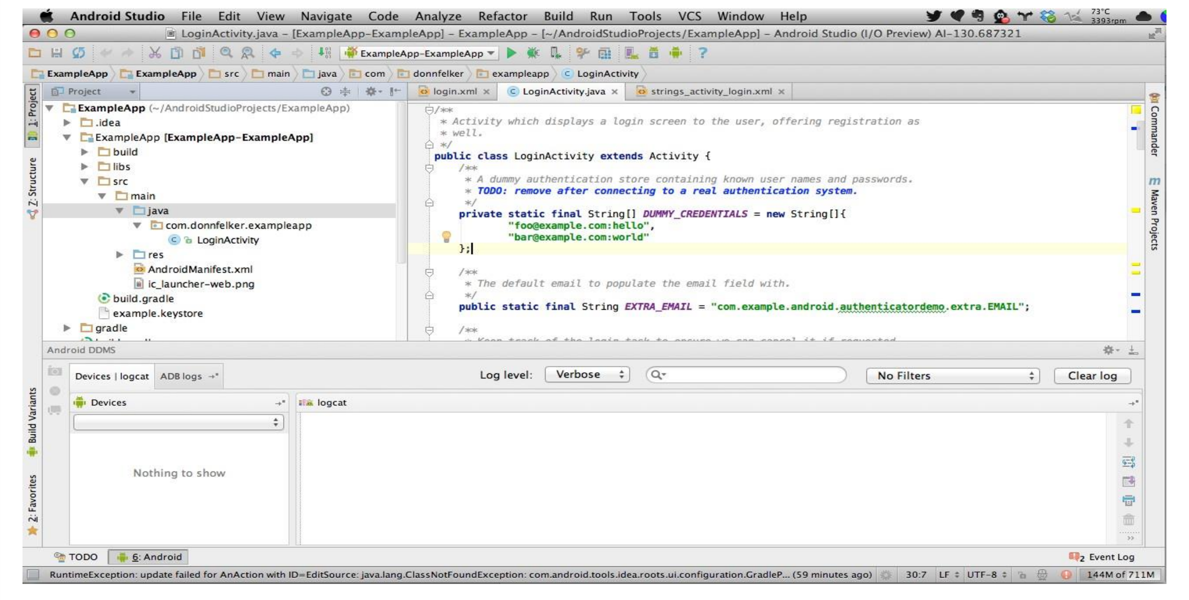
5.1 Introduction to Android Studio:

Android Studio is the IDE for Android that was announced in May 2013 at the Google I/O developers event, and is intended as an alternative to Eclipse. At that time, Android Studio was not ready for full end-to-end Android application development, but should be ready in the coming months. I highly advise you review this chapter, as this is where Android development is migrating to in the future. Android Studio is based on the Java IDE called IntelliJ. If you've worked with other products by JetBrains (developer of IntelliJ), such as RedMine, PyCharm, PhpStorm, WebStorm, or AppCode, you will find yourself at home. All In- telliJ products share the same shell IDE.

Although Android Studio is a brand new IDE, it is important to note that most of your IDE skills from Eclipse apply to Android Studio as well. Most of the tooling in Android Studio is very similar to Eclipse, such as shortcuts, designers, and code editors. You'll still export signed APKs, view logcat, and edit code virtually the same way in Android Studio as if you were in Eclipse. Think of Android Studio like this: if Eclipse were a trusty old power drill used in construction, Android Studio is the new cordless high- powered version of that same drill. Android Studio has some of the same options, and some new ones that you'll need to familiarize yourself with. In the end, you'll still feel comfortable enough to use the tool to get the desired result—an Android app.

5.1.1 Installing Android Studio

Google has made installing Android Studio as simple as possible. Just visit the Android Studio page and download the installer for your platform. Supported platforms include Windows, Mac OS X, and Linux. Follow the installation instructions for your platform to install the application. Installation instructions are not provided in this chapter because installation instructions change often. If you encounter issues, please visit the Android Studio installation page.


Android Studio File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
LoginActivity.java - [ExampleApp-ExampleApp] - ExampleApp - [~/AndroidStudioProjects/ExampleApp] - Android Studio (I/O Preview) AI-130.687321
ExampleApp-ExampleApp
ExampleApp ExampleApp src main java com donnfelker exampleapp LoginActivity
Project
login.xml LoginActivity.java strings_activity_login.xml
ExampleApp (~/AndroidStudioProjects/ExampleApp)
.idea
ExampleApp [ExampleApp-ExampleApp]
build
libs
src
main
java
com.donnfelker.exampleapp
LoginActivity
res
AndroidManifest.xml
ic_launcher-web.png
build.gradle
example.keystore
gradle
1: Project
7: Structure
Build Variants
2: Favorites
Commander
Maven Projects
Android DDMS
Devices | logcat
ADB logs
Log level:
Verbose
No Filters
Clear log
Devices
logcat
Nothing to show
TODO
6: Android
Event Log
30:7 LF UTF-8
144M of 711M

```
/**
 * Activity which displays a login screen to the user, offering registration as
 * well.
 */
public class LoginActivity extends Activity {
    /**
     * A dummy authentication store containing known user names and passwords.
     * TODO: remove after connecting to a real authentication system.
     */
    private static final String[] DUMMY_CREDENTIALS = new String[]{
            "foo@example.com:hello",
            "bar@example.com:world"
    };

    /**
     * The default email to populate the email field with.
     */
    public static final String EXTRA_EMAIL = "com.example.android.authenticatordemo.extra.EMAIL";

    /**
```

```
RuntimeException: update failed for AnAction with ID=EditSource: java.lang.ClassNotFoundException: com.android.tools.idea.roots.ui.configuration.GradleP... (59 minutes ago)
```

Figure 5.1.1: Android Studio with the Editor, Project, and Android panels

5.1.2 Bundled SDK

Android Studio comes bundled with its own version of the Android SDK, which is preconfigured to be used with Android Studio upon installation.

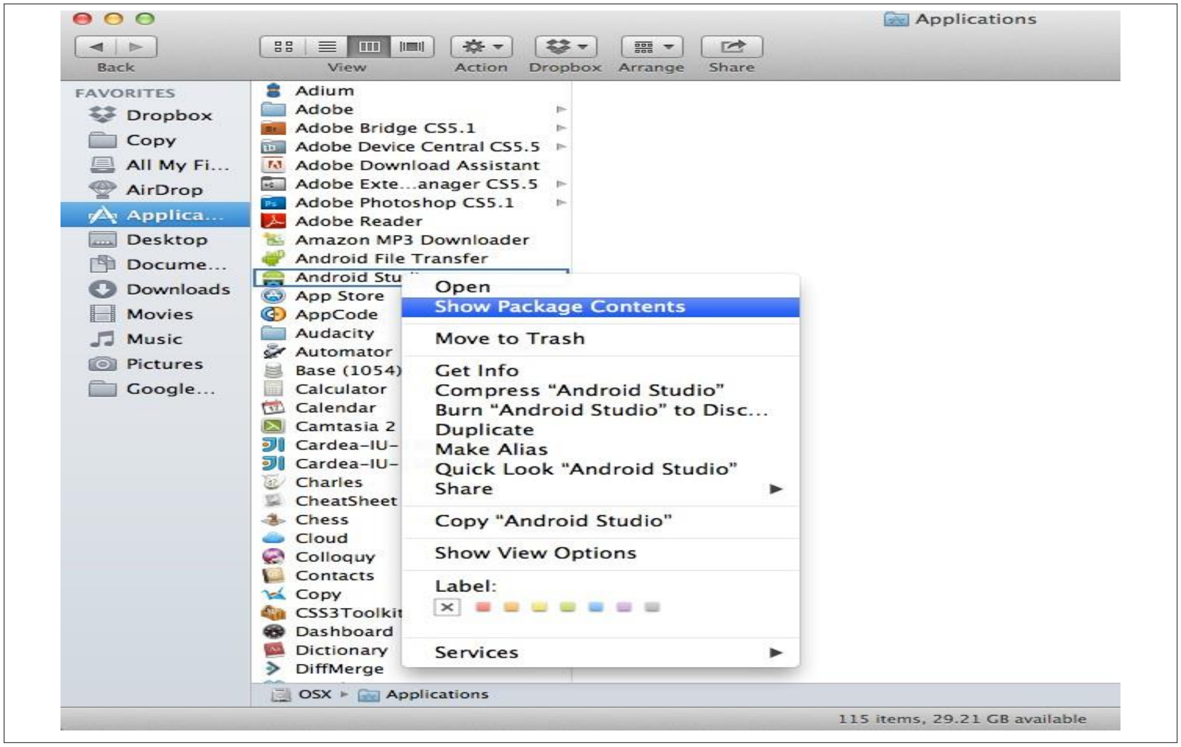


Figure 5.1.2: Showing the package contents of the Android Studio application

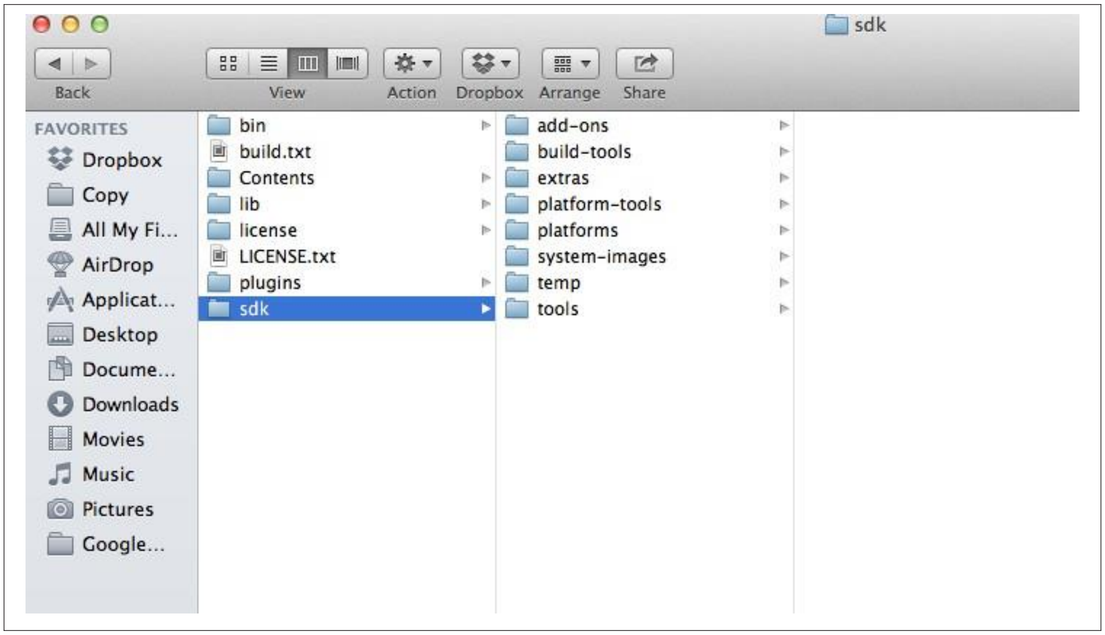


Figure 5.1.2: The SDK folder in the Android Studio package contents

5.1.3 Default Project Location

After installing Android Studio, you can create a new project and define a destination location for the project files. If you don't explicitly define a location for your project, Android Studio will place your files into the *~/AndroidStudioProjects* folder in the cur- rent user's folder on your machine.

5.1.4 Anatomy of the Android Studio IDE

The Android Studio IDE is comprised of a vast array of panels, tools, and functions to help you become as productive as possible at developing Android applications. I'll cover the most common panels, windows, and toolbars with which you'll be interacting.

Panels

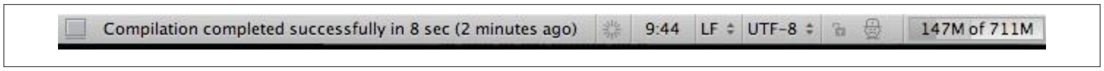
The main panels that you will interact with during your day-to-day development of Android apps appear in Table 5.1.4.

Table 5.1.4. Important panels in Android Studio

Panel	Description
Project Panel	Allows you to navigate through the file hierarchy of your project and select, open, edit, and perform various other actions on your files.
File Editor	The main editing window in Android Studio. This is where you write your code.

Panel	Description
Android Panel	Presents the devices (emulators and physical devices) connected to your system, and allows you to view the logcat output, filter the output, and view ADB logs.
Messages Panel	Here you'll find any important messages that the IDE presents, such as compilation errors.
TODO Panel	Shows all the TODO comments sprinkled throughout your project's code.
Find Results Panel	Here you can examine the results of any find command that you execute. Examples include the Find Results command (Edit ‹ Find ‹ Find) and the Find Usages command (Edit ‹ Find ‹ Find Usages).
Maven Panel	If your project is Maven-based, interact with this panel to perform Maven activities.
Gradle Panel	If you're utilizing the new Gradle build system, you can find the tools necessary to interact with Gradle here.
Event Log Panel	At times, the Android Studio IDE may encounter an unexpected error or have important events that need to be visible to you, the developer. This panel will show you these events.

The final area, which is of utmost importance, is the status bar at the bottom of Android Studio. This is where the majority of status updates will occur when background processes run. Some of these background processes include updating in- dices on the files, Maven or Gradle background processing, and event errors. The right- most box shows the IDE's memory usage.



Compilation completed successfully in 8 sec (2 minutes ago) 9:44 LF ÷ UTF-8 ÷ 147M of 711M

Figure 5.1.4.1. The Android Studio status bar

Toolbars

Android Studio ships with a highly customizable toolbar that is easily accessible from the top of the display. The default toolbar that ships with Android Studio is shown below.

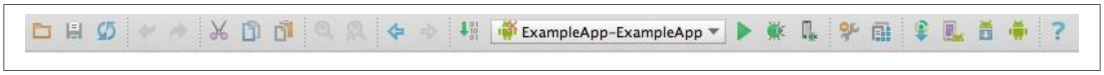


Table 5.1.4.2. Tools in the default toolbar

Tool	Description
File Actions	Actions such as Open, Save, and Synchronize.
Undo/Redo	Undo and redo the previous action.
Cut/Copy/Paste	Quickly cut, copy, and paste from the toolbar.

Tool	Description
Find/Replace	Find and replace values in the project files.
Navigation	Navigate forward and backward in the most recent files that you've accessed or edited recently.
Build/Run/Debug/ Attach	These buttons are some of the most common buttons that you will use in Android Studio, as they allow you to build, run, debug, and attach to a running Android process for debugging.
Settings	These access the IDE Preferences and Project Structure.
Android Actions	The Android Action Group allows you to sync your project with the Gradle files, open the AVD or SDK Manager, and open the Android Monitor application.
Help	Where you can go for help in using Android Studio.

Useful Actions in Android Studio

In addition to the various panels and toolbars, Android Studio has a wide feature set that is accessible via the top menu and various contextual menus. Table below shows a few of the common actions that you'll want to familiarize yourself with.

Table: Common actions

Action	Description
New Module/ Library/Java Library	You can easily add a new Android Module, Android Library, or Java Library to your application by simply choosing the File ‹ New Module or File ‹ Import Module file option and following the wizard through the process.
Preferences	At times, you may want to customize Android Studio. You can do this by accessing the Preferences through the Android Studio ‹ Preferences menu. Some options you can edit are the theme of the IDE, font sizes, keymap, toolbars, and many other options.
Project Structure	An Android project is comprised of modules and libraries, and at times you may need to edit the settings for these modules and libraries. To do so, you'll need to enter the project structure by visiting the File ‹ Project Structure menu.
Showing Additional Windows	Although the default windows that ship with Android Studio are usually sufficient for day-to-day Android development, there may come a time when you need to get into the gritty details of the IDE. Toexplore the various other windows that are available to you (such as file structure, commander, VCS changes, etc.), visit the View ‹Tool Windows menu.
Right-Click to Explore	Anytime you're unaware of the actions you can perform in the IDE, simply right-click the area in which you would like to see the various options. Android Studio will present you with the array of options (if available) that are possible in the given context of the IDE panel in which you're working.

Navigation

Navigation shortcuts are used for navigating around your code base at the speed of light. Master the shortcuts in Table below and you'll increase your productivity immensely.

Table. Keystroke shortcuts in Android Studio

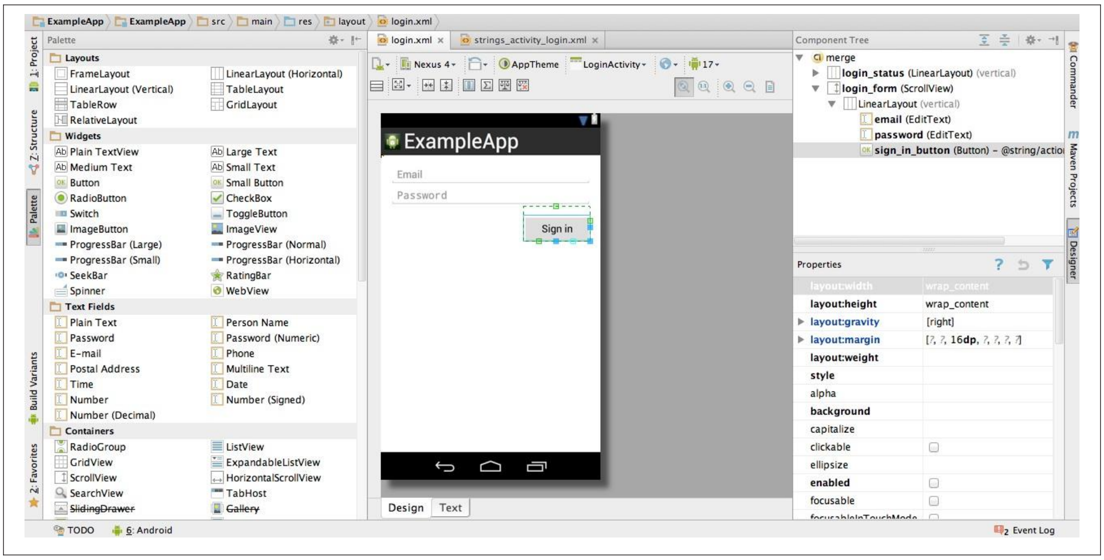
Action	Shortcut on Mac OS X	Shortcut on Windows\Linux
Go to Class	Cmd + O	Ctrl + O
Go to File	Cmd + Shift + O	Ctrl + Shift + O
Go to Definition	Cmd + B	Ctrl + B
Back / Forward	Cmd + [or]	Ctrl + [or]
Code Editor Tab Nav	Cmd+Alt+Left or Right	Ctrl + Alt + Left or Right
File Switcher	Ctrl + Tab	Ctrl + Tab
Find Usages	Alt + F7	Alt + F7
Find	Cmd + F	Ctrl + F
Replace	Cmd + R	Ctrl + R
Find in Path	Cmd + Shift + F	Ctrl + Shift + F
Replace in Path	Cmd + Shift + R	Ctrl + Shift + R

5.1.5 Layout Designer and Layout Preview

Android Studio ships with two graphical tools to help you lay out your user interface: *Layout Designer* and *Layout Preview*. Layout Designer lets you arrange Views on the screen by dragging and dropping, while Layout Preview lets you see how your screen looks while you are editing your XML resources. I'll provide a brief introduction to both tools in this section.

Layout Designer

When you first open an Android layout file, you'll see the Android designer with the Design tab selected. The other tab is Text, which allows you to hop into the XML that defines the layout.



Component Tree:

```
▼ merge
  ▶ login_status (LinearLayout) (vertical)
  ▼ login_form (ScrollView)
    ▼ LinearLayout (vertical)
        email (EditText)
        password (EditText)
        sign_in_button (Button) - @string/actio
```

Properties	
layout:width	wrap_content
layout:height	wrap_content
layout:gravity	[right]
layout:margin	[?, ?, 16dp, ?, ?, ?, ?]
layout:weight	
style	
alpha	
background	
capitalize	
clickable	□
ellipsize	
enabled	□
focusable	□
focusableInTouchMode	□

Android Studio's Layout Designer allows you to easily drag and drop controls onto the layout surface to quickly create a prototype of the layout that you need. Select one of the controls from the palette and drag it to the layout. Once the control is in place, you can edit the various properties of the control by selecting the control and editing the properties on the right. Layout Designer automatically creates the underlying XML code that represents the layout you created. The component tree shows you how the layout is organized in a hierarchical fashion.

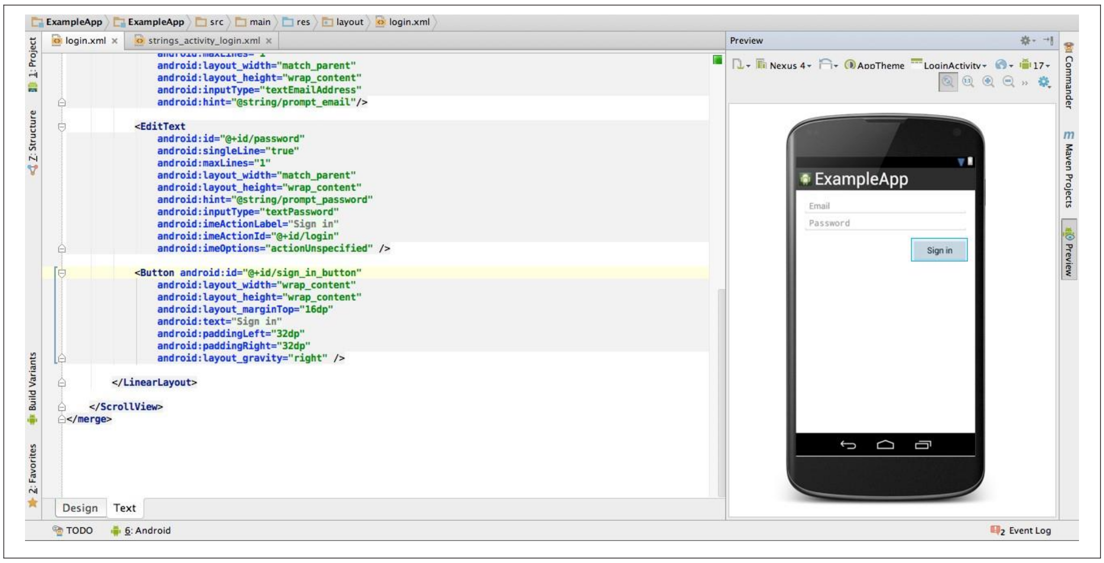
To view the XML of a particular control, simply select it in the designer and click Cmd + B on Mac, or Ctrl + B on Windows\Linux. You can also right-click and choose “Go To Definition.” This will open the Text tab of the layout designer and you are navigated to the XML snippet that defines thatcontrol.

In Layout Designer, you can select various devices to emulate, themes, API levels, and orientations. I highly advise you to peruse the various options in the designer, as it is a very powerful tool.

If you love graphical editors, the Layout Designer is great for whipping up a user interface quickly. However, some of us love to get as close to the metal as possible, and in order to do that you need to edit the XML. To edit the XML, click the Text tab at the bottom of the Layout Designer.

Layout Preview

As soon as you enter the XML layout, you will notice that the control palette, component tree, property editor, and drag-and-drop designer are gone and replaced with a slew of XML code and a layout preview. The preview shown below is the Layout Preview tool. You can turn this panel on and off by selecting the Preview button on the right side of the screen. This panel is shown only when the XML editor is in use.



```
                android:maxLines="1"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:inputType="textEmailAddress"
                android:hint="@string/prompt_email"/>

            <EditText
                android:id="@+id/password"
                android:singleLine="true"
                android:maxLines="1"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:hint="@string/prompt_password"
                android:inputType="textPassword"
                android:imeActionLabel="Sign in"
                android:imeActionId="@+id/login"
                android:imeOptions="actionUnspecified" />

            <Button android:id="@+id/sign_in_button"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginTop="16dp"
                android:text="Sign in"
                android:paddingLeft="32dp"
                android:paddingRight="32dp"
                android:layout_gravity="right" />

        </LinearLayout>

    </ScrollView>
</merge>
```

5.1.6 Generating an APK

Generating an APK in Android Studio is a snap. Follow these steps:

1. Select Generate Signed APK from the Build menu. This will display the Generate Signed APK Wizard.
2. Select your module and click Next.

3. Either supply the path to your keystore that you’re currently using for your Android application, or create a new keystore.
4. (Optional) Once your keystore values are provided, click “Remember Password” and Android Studio will keep track of your entered password in a local password database so you don’t have to enter it again. You will be required to provide a master password for this password database, so be sure you remember this password. Tools like LastPass.com are very useful for keeping track of numerous passwords safely. The remember password feature is very useful if you create or maintain a lot of Android applications.
5. Click Next.
6. At this point you can define the destination for your APK. You can also specify whether you’d like to run ProGuard, and where the ProGuard configuration file is located.
7. Click Finish and your APK will be generated in the destination folder.

5.1.7 Interacting with Maven and Gradle

Maven and Gradle are build systems that are very popular within the Android community. Android Studio ships with support for Maven and Gradle right out of the box. This is great considering that in Eclipse you had to use a plug-in that was often buggy and not entirely reliable. Given that Android Studio ships with support for both tools, you can easily work with projects that use either technology via a panel in Android Studio.

5.2 Activity lifecycle of Android:

As a user navigates through, out of, and back to the app, the Activity instances in the app transition through different states in their lifecycle. The Activity class provides a number of callbacks that allow the activity to know that a state has changed: that the system is creating, stopping, or resuming an activity, or destroying the process in which the activity resides.

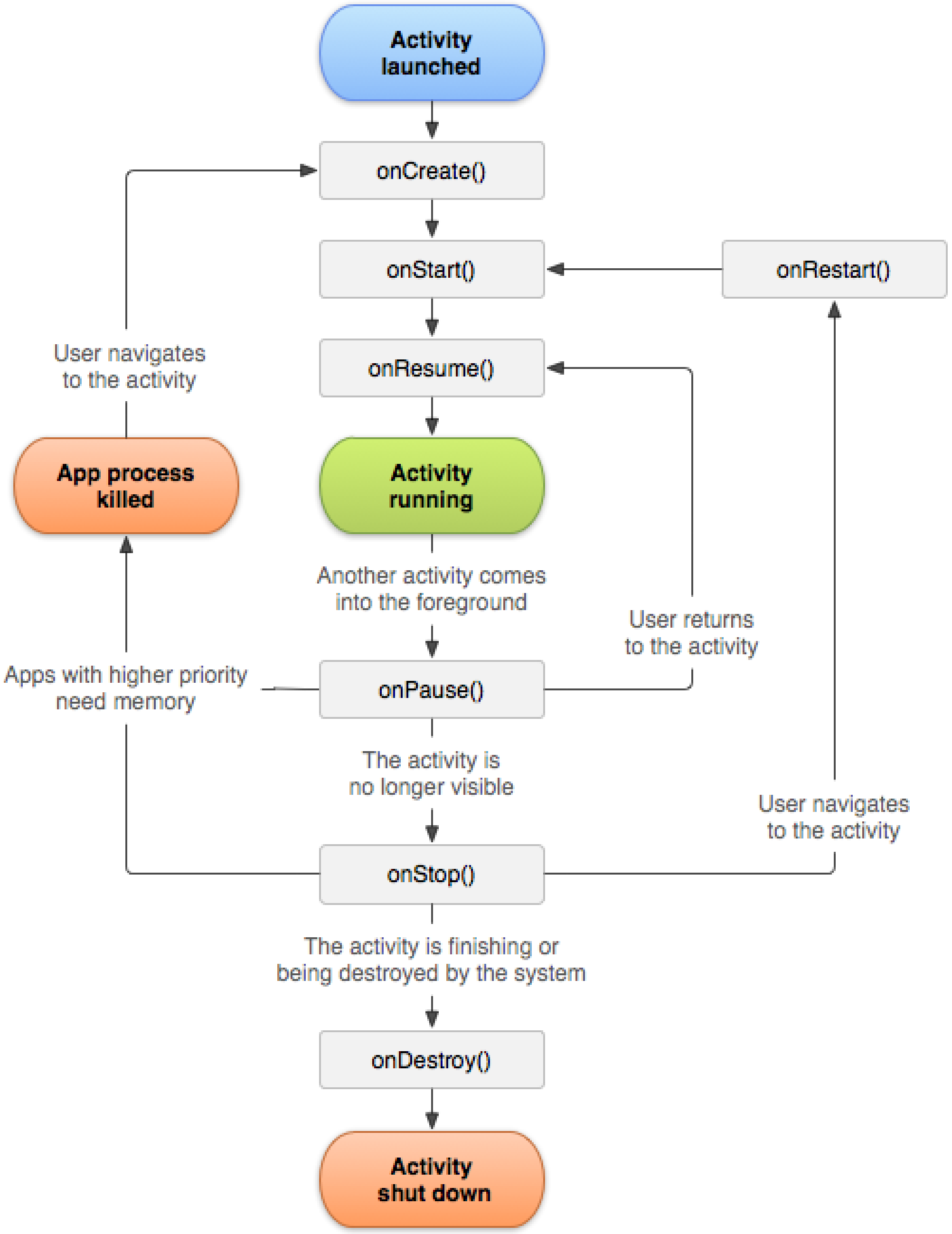
Within the lifecycle callback methods, you can declare how your activity behaves when the user leaves and re-enters the activity. For example, if you're building a streaming video player, you might pause the video and terminate the network connection when the user switches to another app. When the user returns, you can reconnect to the network and allow the user to resume the video from the same spot. In other words, each callback allows you to perform specific work that's appropriate to a given change of state. Doing the right work at the right time and handling transitions properly make your app more robust and performant. For example, good implementation of the lifecycle callbacks can help ensure that your app avoids:

- Crashing if the user receives a phone call or switches to another app while using your app.

- Consuming valuable system resources when the user is not actively using it.
- Losing the user's progress if they leave your app and return to it at a later time.
- Crashing or losing the user's progress when the screen rotates between landscape and portrait orientation.

Activity lifecycle concepts

To navigate transitions between stages of the activity lifecycle, the Activity class provides a core set of six callbacks: onCreate(), onStart(), onResume(), onPause(), onStop(), and onDestroy(). The system invokes each of these callbacks as an activity enters a new state.


Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

As the user begins to leave the activity, the system calls methods to dismantle the activity. In some cases, this dismantlement is only partial; the activity still resides in memory (such as when the user switches to another app), and can still come back to the foreground. If the user returns to that activity, the activity resumes from where the user left off. With a few exceptions, apps are restricted from starting activities when running in the background.

The system's likelihood of killing a given process—along with the activities in it—depends on the state of the activity at the time. Activity state and ejection from memory provides more information on the relationship between state and vulnerability to ejection.

Depending on the complexity of your activity, you probably don't need to implement all the lifecycle methods. However, it's important that you understand each one and implement those that ensure your app behaves the way users expect.

6. Proposed Model:

Modules:

The Proposed software will be having the following modules:

RSS List: RSS stands for Really Simple Syndication. This list will provide the user a way to get all the updates and content available on the website in the android application so that the user does not need to visit the different website daily. This list is a collection of multiple RSS documents. We are going to parse these documents and show it to the user in our application.

News Feed: All the information related to the cyber world will be present in this module. The information here will be provided in the form of the **headlines** or **trending**. The user can scroll through the headlines and click on them to read the full article.

Settings: In this module, the user can decide whether to show the notifications or not, show the notifications with the priority on the top of the notifications list.

About: In this module the user can know the information related to this application like its VERSION, TERMS OF SERVICE, PRIVACY POLICY, LICENCS ETC.

Activities: This app will contain different activities like Scrolling activity and the navigation Drawer activity. The scrolling activity will contain all the information headlines in scrolling format. The navigation Drawer activity contain all the different modules like RSS list, Setting, About.

7. Implementation:

Step 1. Install And Configure your Android Studio with the new version, then afterward create a new android project, application name "News App" with minimum android and tablets API 4.0 jelly Bean configuration with configuring activity "main activity".

Step 2. Now configure Gradle Scripts in build.gradle for customizing dependencies then sync it, now in android manifest give the user permission for INTERNET.

Step 3. Now the main design process starts from res session accompany with drawable, layout, mipmap, and values (color). Change it according to your desire.

Step 4. Now create the package modules in java, afterward create the **News, Article** and **Source class** in that module package as mention in [8]News API

Step 5. Generate your News API key from **www.newsapi.org**, API News Is JSON API for live news and blog articles from the media.

Step 6. Create another package for API in the same java folder then, add a new **API client class** and **API interface** in it.

Step7. API Interface uses retrofit2 as HTTP Client. Retrofit is a REST Client for Java and Android. Its use is very easy.

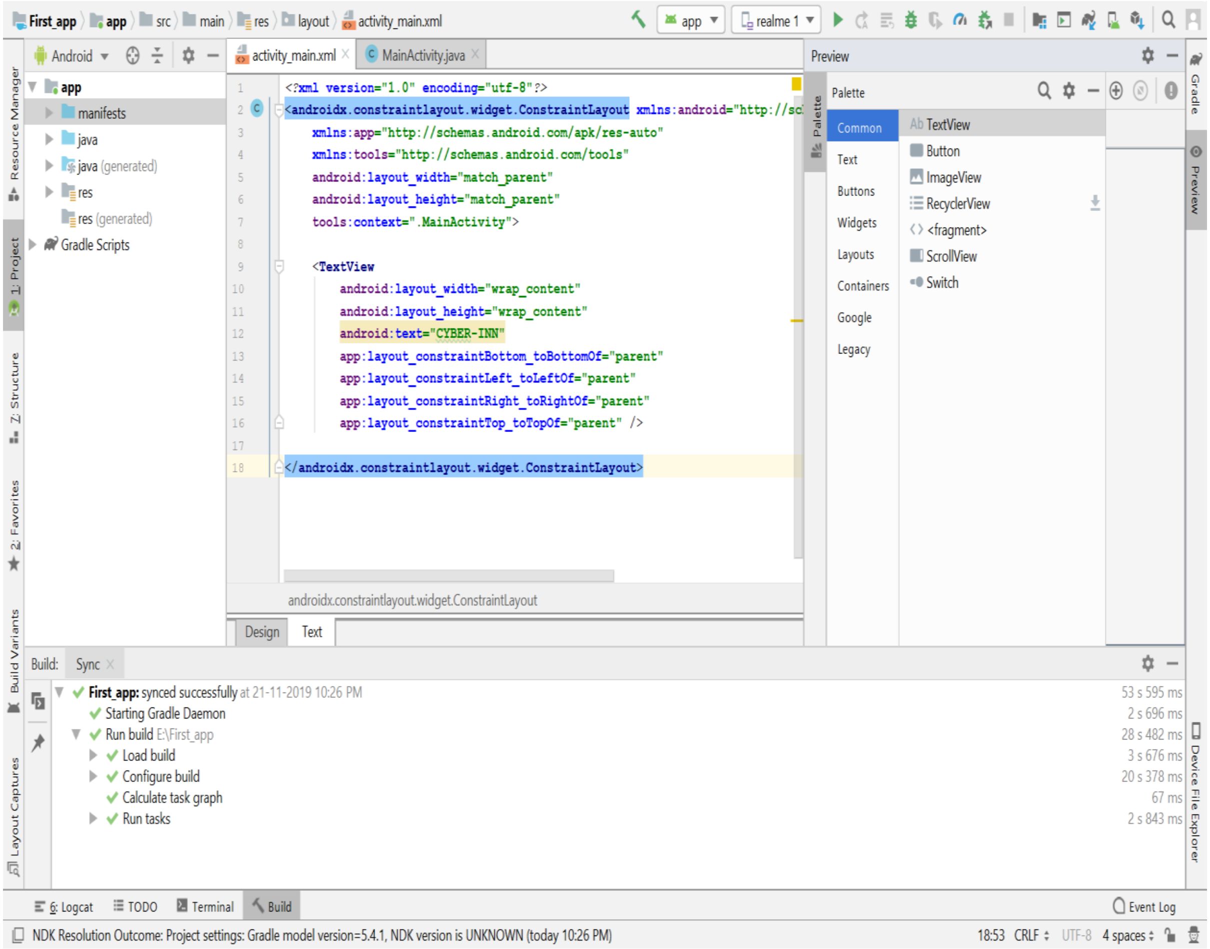
Step 8. Now the overall design process started by adding the new resource file name **item** in the layout folder.

Step 9. Configure your desire layout then, extend all the classes code in the **Adaptor class** And **util class**.

Step 10. Then add your Secret News API key in the **mainActivity class.**

Step 11. Run the program.

Android Studio IDE 3.5.



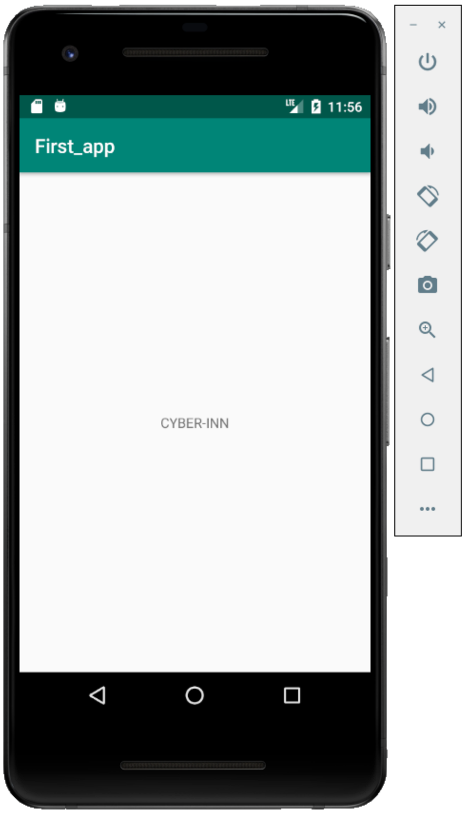
```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://sc
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="CYBER-INN"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

androidx.constraintlayout.widget.ConstraintLayout

```
Build: Sync
▼ ✓ First app: synced successfully at 21-11-2019 10:26 PM        53 s 595 ms
    ✓ Starting Gradle Daemon                                      2 s 696 ms
  ▼ ✓ Run build E:\First app                                     28 s 482 ms
    ▶ ✓ Load build                                                3 s 676 ms
    ▶ ✓ Configure build                                          20 s 378 ms
      ✓ Calculate task graph                                          67 ms
    ▶ ✓ Run tasks                                                 2 s 843 ms
```

```
NDK Resolution Outcome: Project settings: Gradle model version=5.4.1, NDK version is UNKNOWN (today 10:26 PM)    18:53  CRLF  UTF-8  4 spaces
```

Android Emulator

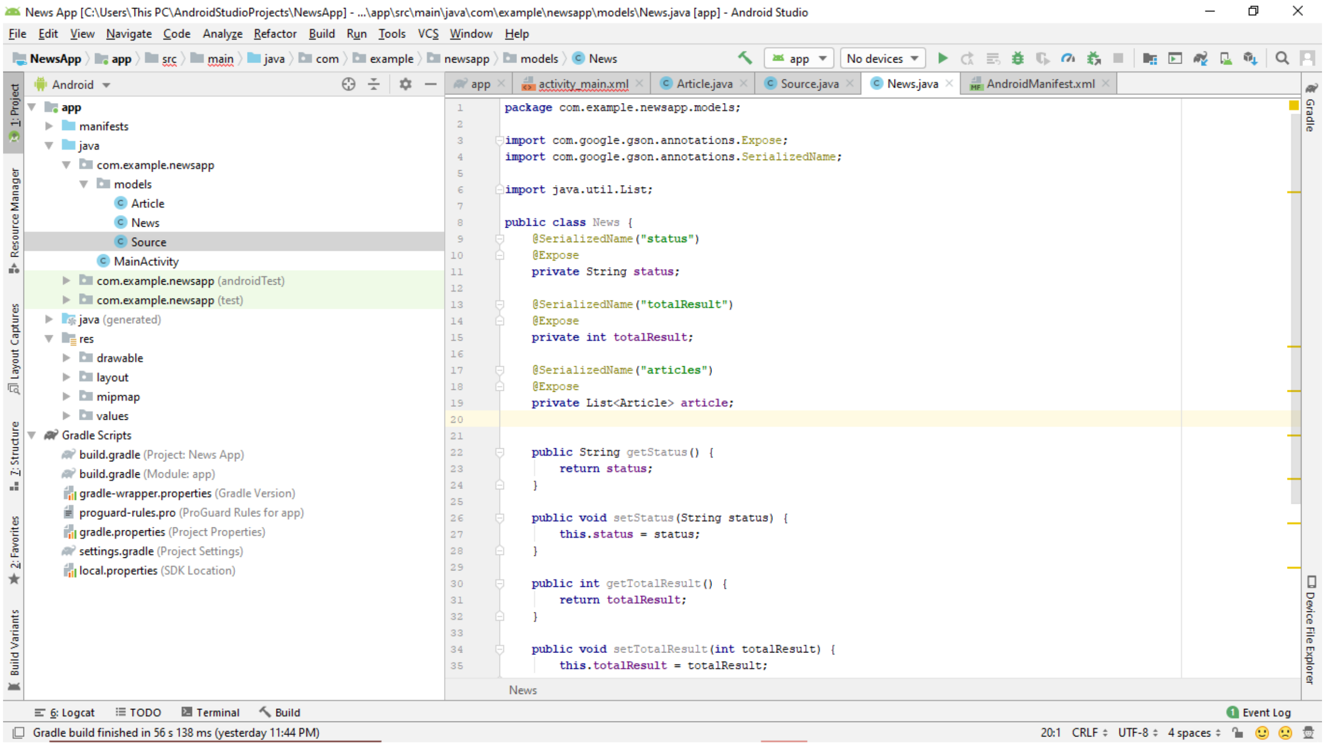


11:56

First_app

CYBER-INN

8. Output / Result / Screenshot:

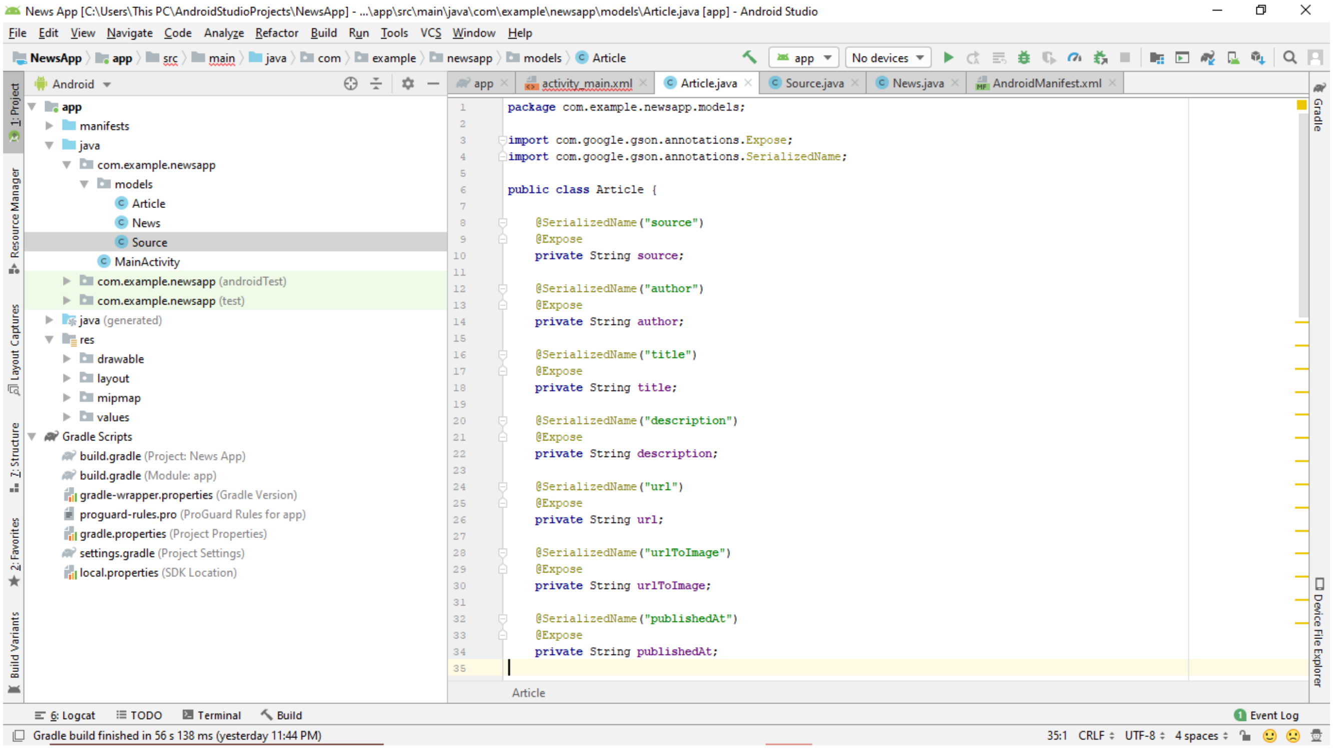


News App [C:\Users\This PC\AndroidStudioProjects\NewsApp] - ...\app\src\main\java\com\example\newsapp\models\News.java [app] - Android Studio

```
package com.example.newsapp.models;

import com.google.gson.annotations.Expose;
import com.google.gson.annotations.SerializedName;

import java.util.List;

public class News {
    @SerializedName("status")
    @Expose
    private String status;

    @SerializedName("totalResult")
    @Expose
    private int totalResult;

    @SerializedName("articles")
    @Expose
    private List<Article> article;


    public String getStatus() {
        return status;
    }

    public void setStatus(String status) {
        this.status = status;
    }

    public int getTotalResult() {
        return totalResult;
    }

    public void setTotalResult(int totalResult) {
        this.totalResult = totalResult;
```

Gradle build finished in 56 s 138 ms (yesterday 11:44 PM)

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
NewsApp app src main java com example newsapp models News
app No devices
Android
activity_main.xml Article.java Source.java News.java AndroidManifest.xml
app
manifests
java
com.example.newsapp
models
Article
News
Source
MainActivity
com.example.newsapp (androidTest)
com.example.newsapp (test)
java (generated)
res
drawable
layout
mipmap
values
Gradle Scripts
build.gradle (Project: News App)
build.gradle (Module: app)
gradle-wrapper.properties (Gradle Version)
proguard-rules.pro (ProGuard Rules for app)
gradle.properties (Project Properties)
settings.gradle (Project Settings)
local.properties (SDK Location)
News
6: Logcat TODO Terminal Build
Event Log
20:1 CRLF UTF-8 4 spaces

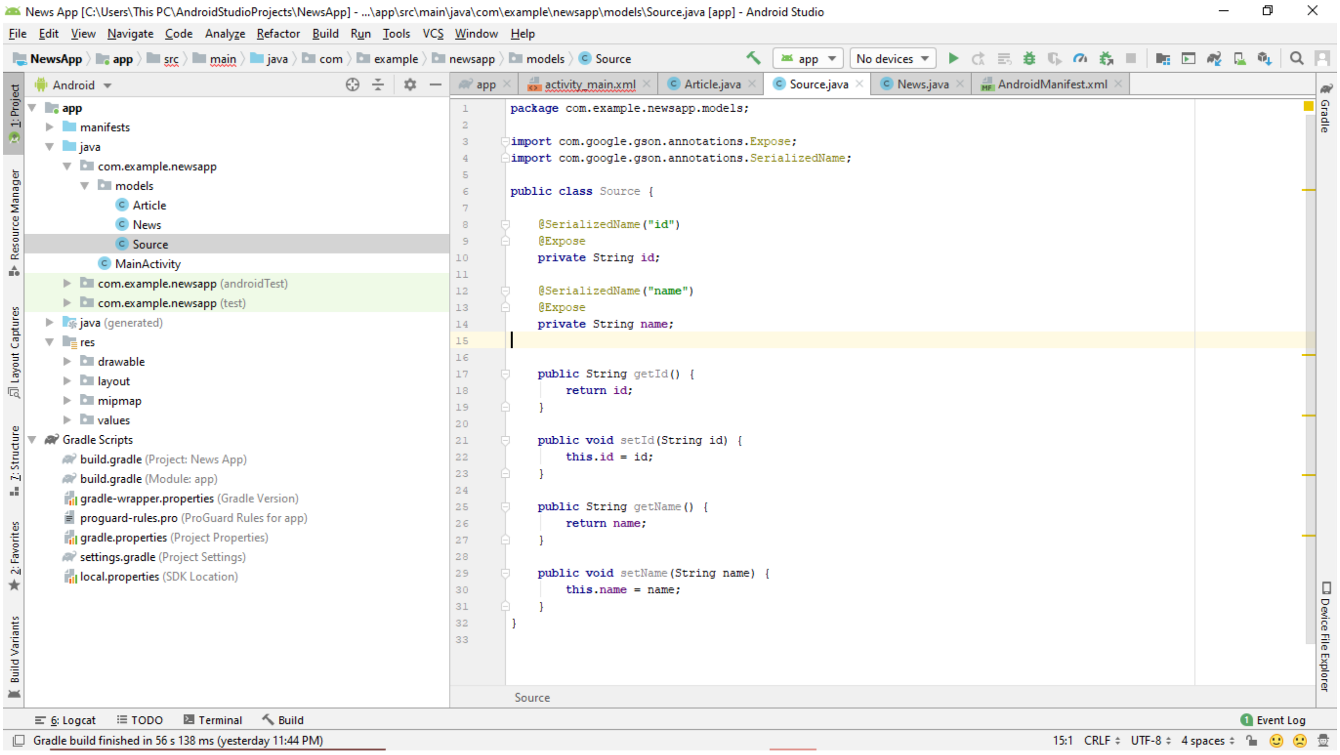


News App [C:\Users\This PC\AndroidStudioProjects\NewsApp] - ...\app\src\main\java\com\example\newsapp\models\Article.java [app] - Android Studio

```
package com.example.newsapp.models;

import com.google.gson.annotations.Expose;
import com.google.gson.annotations.SerializedName;

public class Article {

    @SerializedName("source")
    @Expose
    private String source;

    @SerializedName("author")
    @Expose
    private String author;

    @SerializedName("title")
    @Expose
    private String title;

    @SerializedName("description")
    @Expose
    private String description;

    @SerializedName("url")
    @Expose
    private String url;

    @SerializedName("urlToImage")
    @Expose
    private String urlToImage;

    @SerializedName("publishedAt")
    @Expose
    private String publishedAt;

```

Gradle build finished in 56 s 138 ms (yesterday 11:44 PM)

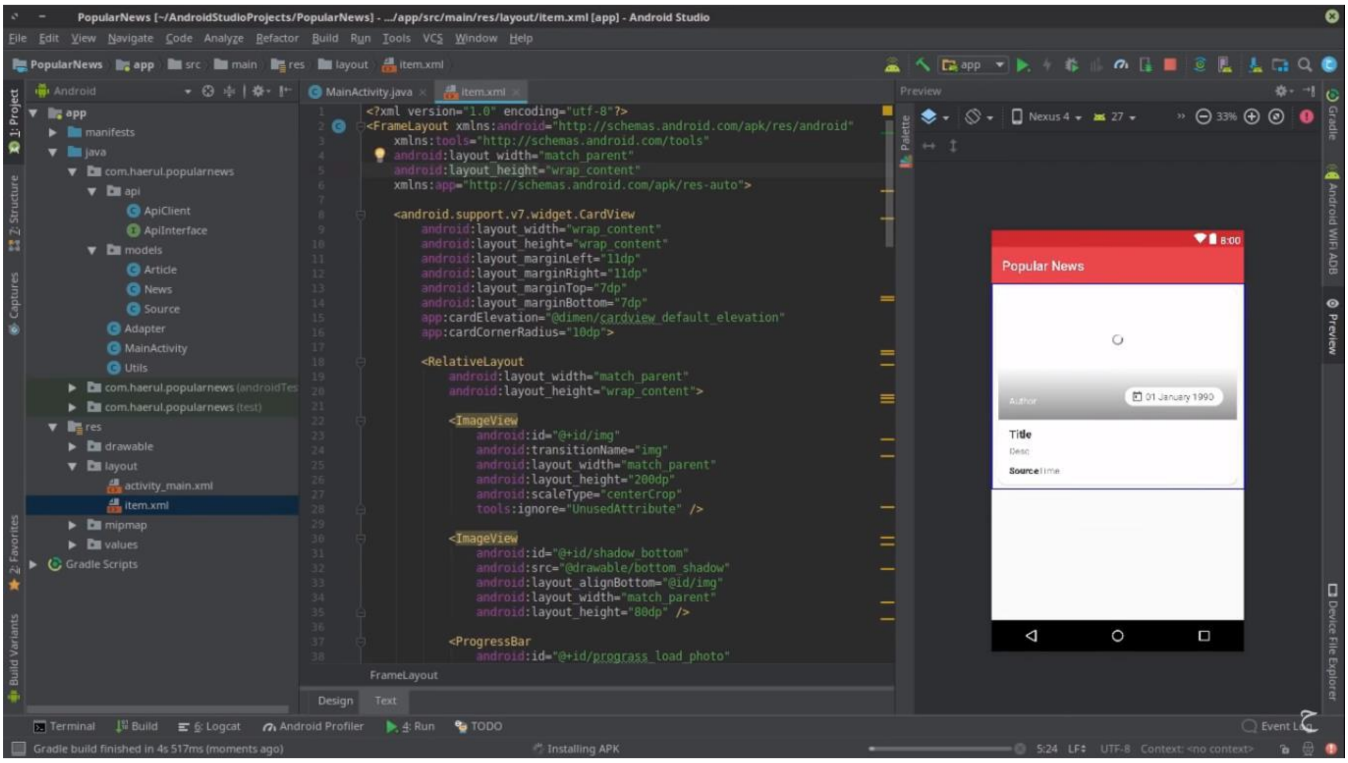
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
NewsApp app src main java com example newsapp models Article
app No devices
Android
activity_main.xml Article.java Source.java News.java AndroidManifest.xml
Article
6: Logcat TODO Terminal Build
Event Log
35:1 CRLF UTF-8 4 spaces



News App [C:\Users\This PC\AndroidStudioProjects\NewsApp] - ...\app\src\main\java\com\example\newsapp\models\Source.java [app] - Android Studio

```java
package com.example.newsapp.models;

import com.google.gson.annotations.Expose;
import com.google.gson.annotations.SerializedName;

public class Source {

    @SerializedName("id")
    @Expose
    private String id;

    @SerializedName("name")
    @Expose
    private String name;


    public String getId() {
        return id;
    }

    public void setId(String id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }
}
```

Gradle build finished in 56 s 138 ms (yesterday 11:44 PM)



PopularNews [~/AndroidStudioProjects/PopularNews] - .../app/src/main/res/layout/item.xml [app] - Android Studio

```xml
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    xmlns:app="http://schemas.android.com/apk/res-auto">

    <android.support.v7.widget.CardView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="11dp"
        android:layout_marginRight="11dp"
        android:layout_marginTop="7dp"
        android:layout_marginBottom="7dp"
        app:cardElevation="@dimen/cardview_default_elevation"
        app:cardCornerRadius="10dp">

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content">

            <ImageView
                android:id="@+id/img"
                android:transitionName="img"
                android:layout_width="match_parent"
                android:layout_height="200dp"
                android:scaleType="centerCrop"
                tools:ignore="UnusedAttribute" />

            <ImageView
                android:id="@+id/shadow_bottom"
                android:src="@drawable/bottom_shadow"
                android:layout_alignBottom="@id/img"
                android:layout_width="match_parent"
                android:layout_height="80dp" />

            <ProgressBar
                android:id="@+id/prograss_load_photo"
```

Gradle build finished in 4s 517ms (moments ago)

9. Conclusions & Future Works:

This project as named "Technomania" is an android application that was proposed to be a game changer for the existing sources of the news industry that produces huge amount of news but very less amount of news related to the Cyber world. Initially the users needed to visit various different websites to gain an information about the Cyber world news and that too wasn't trusted fully. But here this application is supposed to produce trusted news from trusted sources and all of it in just one click. This one click on this app will enable the user to a world of Cyber related things where he will be able to read and analyze a full news article.

10. References:

1.Ribeiro and A. R. D. Silva, "Survey on Cross-Platforms and Languages for Mobile Apps*," Eighth International Conference on the Quality of Information and Communications Technology*, 2012.
2. Pohares, V. C. Kulloli, T. Bhattacharyya, and S. Bhure, "Cross Platform Mobile Application Development," *International Journal of Computer Trends and Technology*, vol. 4, no. 5, pp. 1095–1100 , 2013.
3. S. S. Jagtap and D. B. Hanchate, "Development of Android Based Mobile App for PrestaShop eCommerce Shopping Cart (ALC) ," *International Research Journal of Engineering and Technology (IRJET)* , vol. 4, no. 7, pp. 2248–2254 , Jul. 2017.

4. M. Q. Huynh and P. Ghimire, "Browser App Approach: Can It Be an Answer to the Challenges in Cross-Platform App Development?" *Journal of Information Technology Education: Innovations in Practice*, vol. 16, pp. 047–068, 2017.

5. N. Litayem, B. Dhupia, and S. Rubab, "Review of Cross-Platforms for Mobile Learning Application Development," *International Journal of Advanced Computer Science and Applications*, vol. 6, no. 1, pp. 31–39, 2015.

6. A. Kathuria and A. Gupta, "Challenges in Android Application Development: A Case Study" *International Journal of Computer Science and Mobile Computing*, vol. 4, no. 5, pp. 294–299, May 2015.