Chapter -1

INTRODUCTION

1.1 Project Summary

The evolution of getting everything done through mobile phones has drastically changed our way of life. as to fulfill the need of getting everything done though our mobile the news reading app helps keeping the user informed about the lastest updates around the world. The news reading app has different categories to choose from ranging from sports to finance. In the home page all the latest news is displayed each news is displayed with the headline, if the user wants to read the whole news, he can click on the particular news which leads to the whole story. The different categories display category specific news in an ascending order while the latest news being on the top. News is updated to the app as soon as the news makes it on the internet and this happens add free, without any interruption.

1.2 Project Purpose

The purpose 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 Business, Entertainment, Sports, Science, and Technology or that are in trend at one place. So, it will save time and efforts of the users by making it more efficient. Using, this application will terminate the possibility of information redundancy.

1.3 Project Scope

The Motivations and scope behind this project are to connect people through this application and provide a medium to share their views on the topic/news/information. Then, People with the same interest can interact with each other. However, they can even share more information on the topic. Later on, we can publish this application on the Play Store.

1.4 Technology and Literature Review

1. Android Studio: Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA . On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps, such as:

- A flexible Gradle-based build system
- A fast and feature-rich emulator
- A unified environment where you can develop for all Android devices
- Apply Changes to push code and resource changes to your running app without restarting your app.
- Code templates and GitHub integration to help you build common app features and import sample code.
- Extensive testing tools and frameworks.
- Built-in support for Google Cloud Platform, making it easy to integrate Google.
- Cloud Messaging and App Engine.

2. Node: npm is the world's largest Software Registry. The registry contains over 800,000 code packages. Open-source developers use npm to share software. Many organizations also use npm to manage private development.

3. Blender: Blender is the free and open source 3D creation suite. It supports the entirety of the 3D pipeline—modeling, rigging, animation, simulation, rendering, compositing and motion tracking, even video editing and game creation. Advanced users employ Blender’s API for Python scripting to customize the application and write specialized tools; often these are included in Blender’s future releases. Blender is well suited to individuals and small studios who benefit from its unified pipeline and responsive development process.

4. Python: Python is a computer programming language often used to build websites and software, automate tasks, and conduct data analysis. Python is a general-purpose language, meaning it can be used to create a variety of different programs and isn't specialized for any specific problems.

5. XML: XML (Extensible Markup Language) is a markup language similar to HTML, but without predefined tags to use. Instead, you define your own tags designed specifically for your needs. This is a powerful way to store data in a format that can be stored, searched, and shared. Most importantly, since the fundamental format of XML is standardized, if you share or transmit XML across systems or platforms, either locally or over the internet, the recipient can still parse the data due to the standardized XML syntax.

6. Java: Java is a programming language and computing platform first released by Sun Microsystems in 1995. It has evolved from humble beginnings to power a large share of today's digital world, by providing the reliable platform upon which many services and applications are built. New, innovative products and digital services designed for the future continue to rely on Java, as well.

7. Kotlin: Kotlin is a general purpose, free, open source, statically typed "pragmatic" programming language initially designed for the JVM (Java Virtual Machine) and Android that combines object-oriented and functional programming features. It is focused on interoperability, safety, clarity, and tooling support. Kotlin originated at JetBrains, the company behind IntelliJ IDEA, in 2010, and has been open source since 2012. The Kotlin team currently has more than 90 full-time members from JetBrains, and the Kotlin project on GitHub has more than 300 contributors. JetBrains uses Kotlin in many of its products including its flagship IntelliJ IDEA.

In this app, we are using **newsapi.org API** to get news and get JSON data back. Then it is parsed using Google's **gson library** and the requests are sent by **Retrofit** and is displayed to the user. One should have basic knowledge of **Java, OPPs, Activity, Fragment,** anyone **networking library**, **Layouts**, and **Recycler view** to implement this app**.**

1.5 Problem Statement

"Design and develop a news app which feeds user with news articles from various publications and categories."

1.6 Objective of The Project

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

Many people generally get the redundancy in the information. Sometimes, people even spread fake news, which circulates and spread more like a disease of false information in WhatsApp and other social media. Various myths are also likely to spread as soon as possible which gives more harm than good to the people. This app while cross-checks the redundancy in the information along with the false and misleading information, which later results in panic in the people.

Chapter -2

SYSTEM REQUIREMENTS

2.1 User Characteristics

The various probable characteristics of an end user of this application may include:

- User visiting the app can view list of news from various genres.
- User can read articles from different news publications.

2.2 Hardware and Software Requirements

Hardware Requirements

1. Processor: Intel CORE i5 Octa-core1.3GHz and above
2. Ram: 8GB and above
3. Display 1080*1920 Resolution, 16:9 ratio and above
4. Internal memory 8GB and above

Software Requirements

1. Operating system: Android4.1(jelly beam) and above
2. Coding Language: Java
3. Java Platform JRE 11.0.11 amd64 and Open JDK 64-Bit Server VM by Oracle Corporation
4. Android Studio

2.3 Project Operation Constraints

Libraries used along with their versions:

2.3.1. implementation ‘com.android.support:cardview-v7:26.1.0’

This library is used for cardView design. Its looks like cards.

2.3.2. implementation ‘com.google.code.gson:gson:2.9.0’

This library is used to convert Java Objects into JSON representation. It can also be used to convert JSON string to Java Object.

2.3.3. implementation ‘com.squareup.retrofit2:converter-gson:2.9.0’
This library is convertor. A Convertor which uses Gson for serialization to and from JSON.

2.3.4 implementation ‘com.android.support:design:26.1.0’
This library is used to support the design. Like Recycler view, Floating Action button, Navigation View etc.

2.3.5 implementation 'com.squareup.picasso:picasso:2.71828'

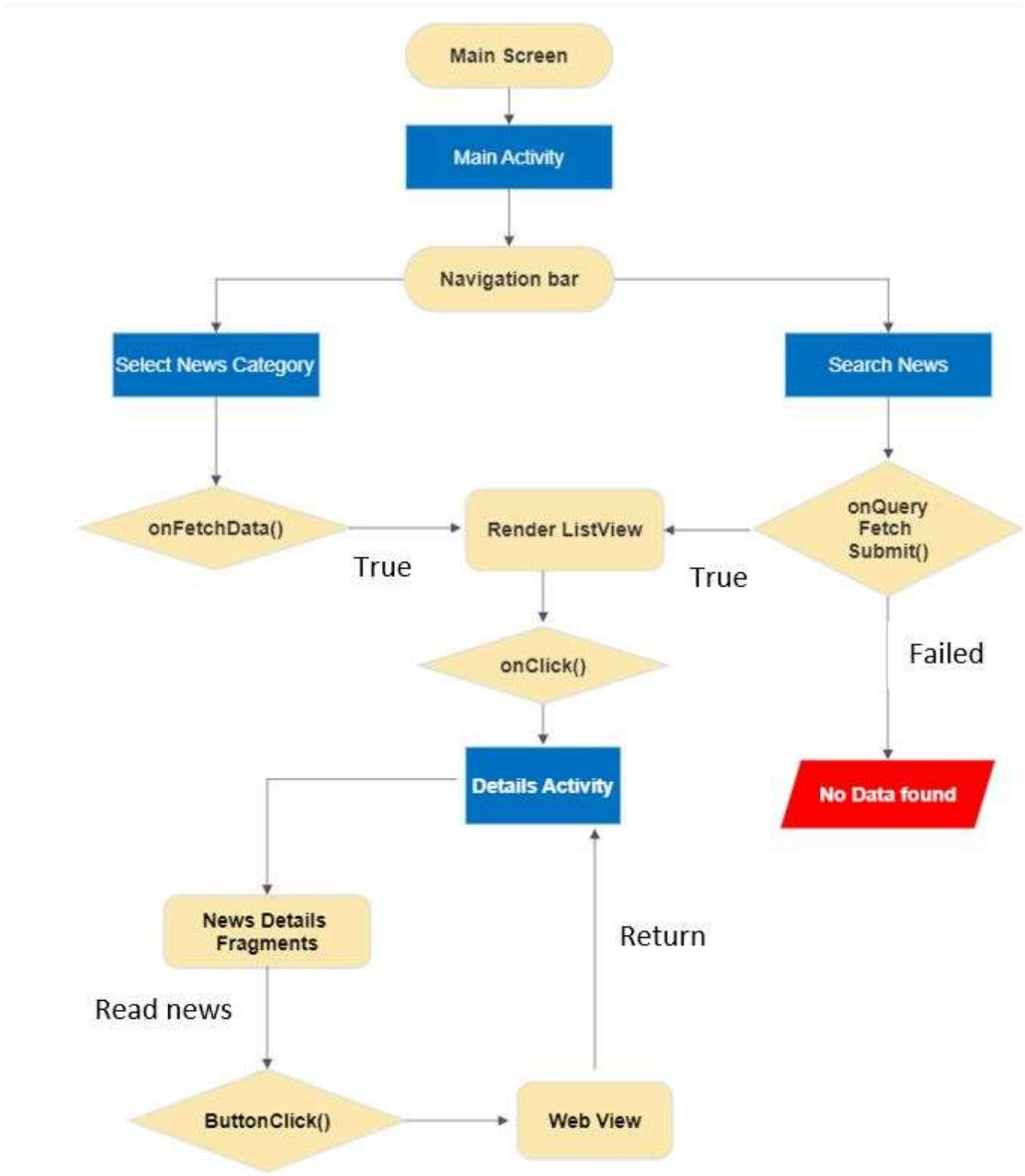
A powerful image downloading and caching library for Android.

One should have basic knowledge of Java, OOPs, Activity, fragments, anyone networking library, Layouts and and Recycler view to implement this app.

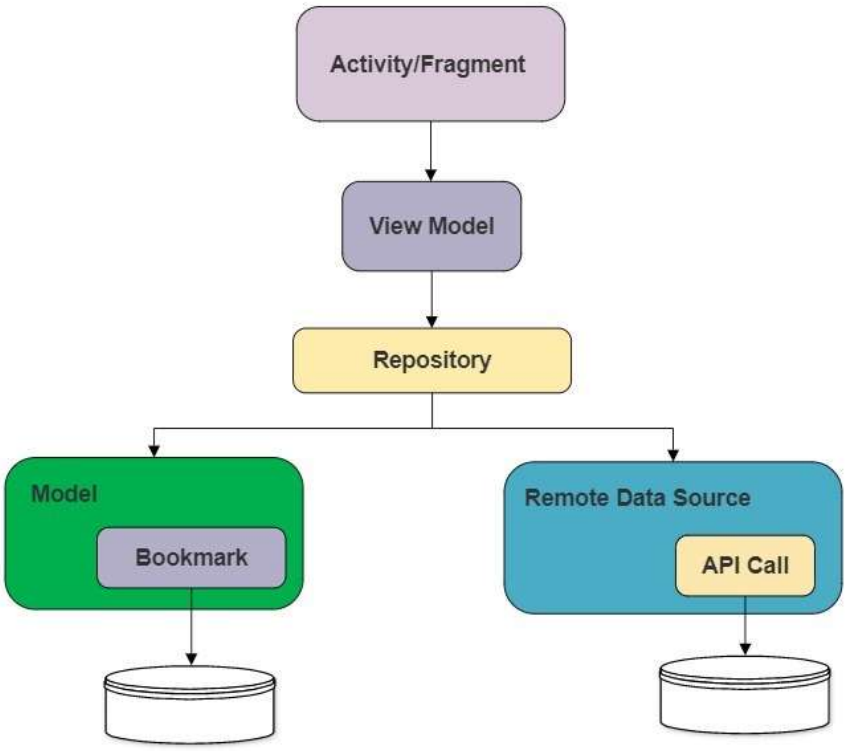
Chapter -3

DESIGN

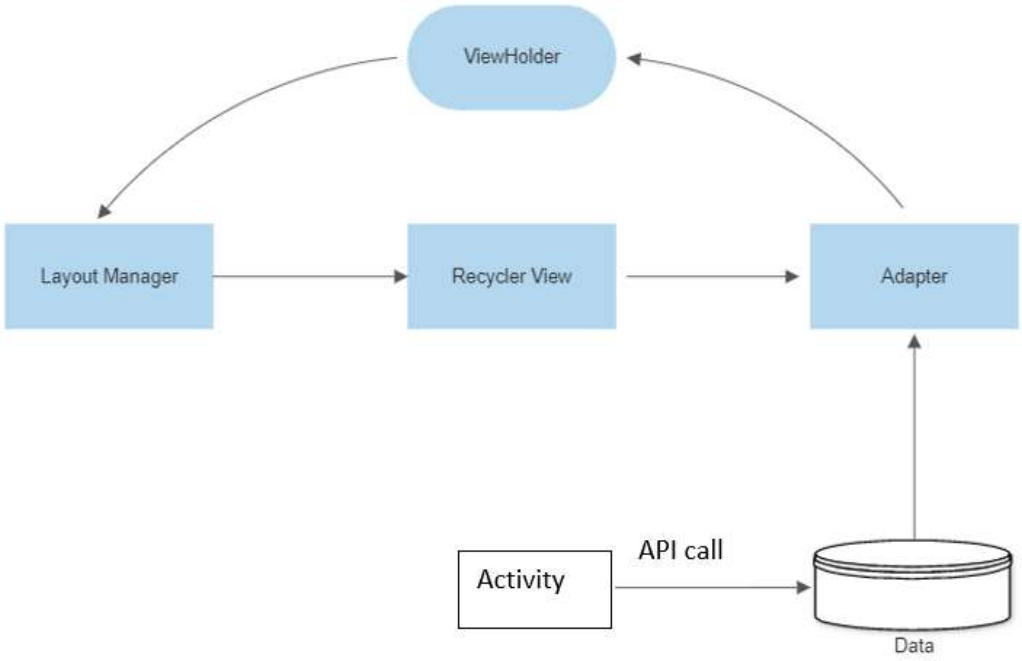
3.1 Control Flow



3.2 Architecture



3.3 Data Flow



3.4 Module Description

The proposed app will be having the following modules:

1. Activities:

This app will contain different activities like Scrolling activity and the navigation bar activity. The scrolling activity will contain all the information headlines in a scrolling format. The navigation bar activity contains all the different genres like Entertainment, Business, Health, Science, Sports and Technology.

2. Views:

View is the basic building block of UI (User Interface) in android. View refers to the android.view.View class, which is the super class for all the GUI components. The app contains TextView, ImageView, Button, EditText, ListView,ProgressBar, CardView which helps in achieving the flow of Activities and design the layout responsively. Image Views are used to display the icon of the app and List Views are used to display the daily news according to the category

3. Layouts:

Android Layout is used to define the user interface that holds the UI controls or widgets that will appear on the screen of an android application or activity screen.
The app is designed using Constraint Layout and Linear Layout.

4. Intents, Resources:

The app communicates or transfers the control and information from one Activity to other Activity with the help of intents. The ID's, String values, Colors, Styles, XML file for every Activity, Drawable such as icons are stored in the resource folder.

Chapter -4

IMPLEMENTATION

4.1Built-In Functions

startActivity (Intent): Enables the program to send information from one activity to another.

findViewById(R.id._id): Find a view that was identified by the Id attribute from XML Resource.

ArrayList: Finds a view that was identified by the id attribute from the XML layout resource.

Fragment: A Fragment represents a reusable portion of your app's UI. A fragment defines and manages its own layout, has its own lifecycle, and can handle its own input events. Fragments cannot live on their own--they must be hosted by an activity or another fragment.

Gson: Gson is a Java library that can be used to convert Java Objects into their JSON representation. It can also be used to convert a JSON string to an equivalent Java object. Gson is an open-source project hosted at code.google.com.

newsapi: Locate articles and breaking news headlines from news sources and blogs across the web with JSON API.

CardView: Card View is the view that can display views on top of each other. The main usage of Card View is that it helps to give a rich feel and look to the UI design.

RecyclerView: A Recycler View is a container used to display a list or grid of data, like text, photos.

Retrofit: It is a simple network library that used for network transactions. By using this library we can seamlessly capture JSON response from web service/web API.

Picasso: Picasso is an image library for Android. It's created and maintained by Square, and caters to image loading and processing. It simplifies the process of displaying images from external locations.

4.2Android Manifest and Special Permissions

We are working with network operations etc. so we need Internet connection. So we need to add Internet permission in **AndroidManifest.xml**

<uses-permission android:name=”android.permission.INTERNET”/>

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.newsx">

    <uses-permission android:name="android.permission.INTERNET" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.NewsX">
        <activity android:name=".DetailsActivity"></activity>
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

4.3XML and Java Code

XML

acitivity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="275dp"
   xmlns:app="http://schemas.android.com/apk/res-auto">
   <androidx.cardview.widget.CardView
      android:layout_width="match_parent"
      android:layout_height="match_parent"
      android:padding="10dp"
      android:layout_margin="2dp"
      app:cardCornerRadius="10dp"
      android:id="@+id/cardview"
      android:elevation="5dp">
      <LinearLayout
         android:layout_width="match_parent"
         android:layout_height="match_parent"
         android:orientation="vertical">
         <TextView
            android:layout_width="match_parent"
            android:layout_height="60dp"
            android:id="@+id/mainheading"
            android:padding="5dp"
            android:ellipsize="end"
            android:maxLines="2"
            android:textColor="@color/black"
            android:text="Main heading display here hello Main heading "
            android:textSize="20sp"
            android:textStyle="bold">
         </TextView>
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="130dp"
    android:layout_marginTop="10dp"
    android:orientation="horizontal">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:id="@+id/content"
        android:padding="5dp"
        android:ellipsize="end"
        android:maxLines="6"
        android:text="Main heading display here hello Main heading "
        android:textSize="15sp">
    </TextView>
    <ImageView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:padding="5dp"
        android:layout_weight="1"
        android:scaleType="centerCrop"
        android:id="@+id/imageview"
        android:src="@drawable/ic_launcher_background">
    </ImageView>
</LinearLayout>
<TextView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:text="By Sandeep"
    android:textSize="13sp"
    android:textStyle="italic"
    android:id="@+id/author"
```

```
            android:textColor="@color/purple_200"
            android:padding="5dp">
        </TextView>
        <TextView
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:text="yuagwefuggefw"
            android:id="@+id/time"
            android:padding="5dp"
            android:textColor="@color/black"
            android:textStyle="bold"
            android:textSize="12sp">
        </TextView>
    </LinearLayout>
  </androidx.cardview.widget.CardView>
</RelativeLayout>
```

layout_item.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="match_parent"
  android:layout_height="275dp"
  xmlns:app="http://schemas.android.com/apk/res-auto">
  <androidx.cardview.widget.CardView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="10dp"
    android:layout_margin="2dp"
    app:cardCornerRadius="10dp"
    android:id="@+id/cardview"
    android:elevation="5dp">
    <LinearLayout
```

```
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical">
<TextView
   android:layout_width="match_parent"
   android:layout_height="60dp"
   android:id="@+id/mainheading"
   android:padding="5dp"
   android:ellipsize="end"
   android:maxLines="2"
   android:textColor="@color/black"
   android:text="Main heading display here hello Main heading "
   android:textSize="20sp"
   android:textStyle="bold">
</TextView>
<LinearLayout
   android:layout_width="match_parent"
   android:layout_height="130dp"
   android:layout_marginTop="10dp"
   android:orientation="horizontal">
   <TextView
      android:layout_width="match_parent"
      android:layout_height="match_parent"
      android:layout_weight="1"
      android:id="@+id/content"
      android:padding="5dp"
      android:ellipsize="end"
      android:maxLines="6"
      android:text="Main heading display here hello Main heading "
      android:textSize="15sp">
   </TextView>
   <ImageView
      android:layout_width="match_parent"
```

```
            android:layout_height="match_parent"
            android:padding="5dp"
            android:layout_weight="1"
            android:scaleType="centerCrop"
            android:id="@+id/imageview"
           android:src="@drawable/ic_launcher_background">
        </ImageView>
    </LinearLayout>
    <TextView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:text="By Sandeep"
        android:textSize="13sp"
        android:textStyle="italic"
        android:id="@+id/author"
        android:textColor="@color/purple_200"
        android:padding="5dp">
    </TextView>
    <TextView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:text="yuagwefuggefw"
        android:id="@+id/time"
        android:padding="5dp"
        android:textColor="@color/black"
        android:textStyle="bold"
        android:textSize="12sp">
    </TextView>
  </LinearLayout>
 </androidx.cardview.widget.CardView>
</RelativeLayout>
```

activity_web_view.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    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=".webView">
    <androidx.appcompat.widget.Toolbar
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:layout_marginTop="0dp"
        android:background="@color/white"
        android:id="@+id/toolbar">
        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:gravity="center_vertical">
            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Specific News"
                android:textStyle="bold"
                android:textSize="20sp"
                android:textColor="#03A9F4">
            </TextView>
        </RelativeLayout>
    </androidx.appcompat.widget.Toolbar>
    <WebView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/webview"
        android:layout_below="@+id/toolbar">
```

```
    </WebView>
</RelativeLayout>
```

Healthfragment.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:background="#E6E6E6"
    android:layout_height="match_parent">

    <androidx.recyclerview.widget.RecyclerView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/recyclerviewofhealth"
        android:layout_margin="2dp"
        android:fitsSystemWindows="true"
        android:padding="20dp">

    </androidx.recyclerview.widget.RecyclerView>

</RelativeLayout>
```

Java

Mainactivity.java

```
package com.example.newz5;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;
import androidx.viewpager.widget.ViewPager;
import android.os.Bundle
import com.google.android.material.tabs.TabItem;
import com.google.android.material.tabs.TabLayout;
public class MainActivity extends AppCompatActivity {
    TabLayout tabLayout;
    TabItem mhome,mscience,msports,mhealth,mtechnology,mentertainment;
    PagerAdapter pagerAdapter;
    Toolbar mtoolbar;
    String api="842427b7f1194fc292ecff37228863a9";
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mtoolbar=findViewById(R.id.toolbar);
        setSupportActionBar(mtoolbar);
        mhome=findViewById(R.id.home);
        mscience=findViewById(R.id.science);
        msports=findViewById(R.id.sports);
        mtechnology=findViewById(R.id.technology);
        mhealth=findViewById(R.id.health);
        mentertainment=findViewById(R.id.entertainment);
        ViewPager viewPager=findViewById(R.id.fragmentContainer);
        tabLayout=findViewById(R.id.include);

        pagerAdapter=new PagerAdapter(getSupportFragmentManager(),6);
        viewPager.setAdapter(pagerAdapter);
```

```
        tabLayout.addOnTabSelectedListener(new TabLayout.OnTabSelectedListener()
{
            @Override
            public void onTabSelected(TabLayout.Tab tab) {
                viewPager.setCurrentItem(tab.getPosition());
if(tab.getPosition()==0||tab.getPosition()==1||tab.getPosition()==2||tab.getPosition()=
=3||tab.getPosition()==4||tab.getPosition()==5)
                {
                    pagerAdapter.notifyDataSetChanged();
                }
            }

            @Override
            public void onTabUnselected(TabLayout.Tab tab) {
            }
            @Override
            public void onTabReselected(TabLayout.Tab tab) {
            }
        });
        viewPager.addOnPageChangeListener(new
TabLayout.TabLayoutOnPageChangeListener(tabLayout));
    }
}
```

MainNews.java

```
package com.example.newz5;
import java.util.ArrayList;
public class mainNews {
    private String status;
    private String totalResults;
    private ArrayList<ModelClass> articles;
    public mainNews(String status, String totalResults, ArrayList<ModelClass>
articles) {
        this.status = status;
        this.totalResults = totalResults;
        this.articles = articles;
    }
    public String getStatus() {
        return status;
    }
    public void setStatus(String status) {
        this.status = status;
    }
    public String getTotalResults() {
        return totalResults;
    }
    public void setTotalResults(String totalResults) {
        this.totalResults = totalResults;
    }
    public ArrayList<ModelClass> getArticles() {
        return articles;
    }
    public void setArticles(ArrayList<ModelClass> articles) {
        this.articles = articles;
    }
}
```

ModelClass.java

```
package com.example.newz5;
public class ModelClass {
    private String author,title,description,url,urlToImage,publishedAt;
    public ModelClass(String author, String title, String description, String url, String
urlToImage, String publishedAt) {
        this.author = author;
        this.title = title;
        this.description = description;
        this.url = url;
        this.urlToImage = urlToImage;
        this.publishedAt = publishedAt;
    }
    public String getAuthor() {
        return author;
    }
    public void setAuthor(String author) {
        this.author = author;
    }
    public String getTitle() {
        return title;
    }
    public void setTitle(String title) {
        this.title = title;
    }
    public String getDescription() {
        return description;
    }
    public void setDescription(String description) {
        this.description = description;
    }
    public String getUrl() {
        return url;
```

```
    }
    public void setUrl(String url) {
        this.url = url;
    }
    public String getUrlToImage() {
        return urlToImage;
    }
    public void setUrlToImage(String urlToImage) {
        this.urlToImage = urlToImage;
    }
    public String getPublishedAt() {
        return publishedAt;
    }
    public void setPublishedAt(String publishedAt) {
        this.publishedAt = publishedAt;
    }
}
```

WebView.java

```
package com.example.newz5;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.webkit.WebView;
import android.webkit.WebViewClient;
import androidx.appcompat.widget.Toolbar;
public class webView extends AppCompatActivity {
    Toolbar toolbar;
    WebView webView;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_web_view);
```

```
        toolbar=findViewById(R.id.toolbar);
        webView=findViewById(R.id.webview);
        setSupportActionBar(toolbar);
        Intent intent=getIntent();
        String url = intent.getStringExtra("url");
        webView.setWebViewClient(new WebViewClient());
        webView.loadUrl(url);
    }
}
```

PageAdaptor.java

```
package com.example.newz5;
import androidx.annotation.NonNull;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.FragmentManager;
import androidx.fragment.app.FragmentPagerAdapter;
public class PagerAdapter extends FragmentPagerAdapter {
    int tabcount;
    public PagerAdapter(@NonNull FragmentManager fm, int behavior) {
        super(fm, behavior);
        tabcount=behavior;
    }
    @NonNull
    @Override
    public Fragment getItem(int position) {
        switch (position)
        {
            case 0:
                return new HomeFragment();
            case 1:
                return new SportsFragment();
            case 2:
                return new HealthFragment();
```

```
            case 3:
                return new ScienceFragment();
            case 4:
                return new EntertainmentFragment();
            case 5:
                return new TechnologyFragment();
            default:return null;
        }
    }
    @Override
    public int getCount() {
        return tabcount;
    }
}
```

HealthFragement.java

```
package com.example.newz5;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import java.util.ArrayList;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;
public class HealthFragment extends Fragment {
    String api="842427b7f1194fc292ecff37228863a9";
```

```
    ArrayList<ModelClass> modelClassArrayList;
    Adapter adapter;
    String country="in";
    private RecyclerView recyclerViewofhealth;
    private String category = "health"
    @Nullable
    @Override
    public View onCreateView(@NonNull  LayoutInflater inflater, @Nullable
ViewGroup container, @Nullable  Bundle savedInstanceState) {
        View v = inflater.inflate(R.layout.healthfragment,null);
        recyclerViewofhealth = v.findViewById(R.id.recyclerviewofhealth);
        modelClassArrayList = new ArrayList<>();
        recyclerViewofhealth.setLayoutManager(new
LinearLayoutManager(getContext()));
        adapter=new Adapter(getContext(),modelClassArrayList);
        recyclerViewofhealth.setAdapter(adapter);
        findNews();
        return v;
    }
    private void findNews() {
ApiUtilities.getApiInterface().getCategoryNews(country,category,100,api).enqueue(n
ew Callback<mainNews>() {
            @Override
            public void onResponse(Call<mainNews> call, Response<mainNews>
response) {
                if(response.isSuccessful())
                {
                    modelClassArrayList.addAll(response.body().getArticles());
                    adapter.notifyDataSetChanged();
                }
            }
            @Override
            public void onFailure(Call<mainNews> call, Throwable t) {
```

```
        }
    });
  }
}
```

Adaptor.java

```
package com.example.newz5;
import android.content.Context;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;
import com.bumptech.glide.Glide;
import java.util.ArrayList;
import androidx.annotation.NonNull;
import androidx.cardview.widget.CardView;
import androidx.recyclerview.widget.RecyclerView;
public class Adapter extends RecyclerView.Adapter<Adapter.ViewHolder>{
  Context context;
  ArrayList<ModelClass>modelClassArrayList;
  public Adapter(Context context, ArrayList<ModelClass> modelClassArrayList) {
    this.context = context;
    this.modelClassArrayList = modelClassArrayList;
  }
  @NonNull
  @Override
  public Adapter.ViewHolder onCreateViewHolder(@NonNull ViewGroup parent,
int viewType) {
    View view =
LayoutInflater.from(context).inflate(R.layout.layout_item,null,false);
    return new ViewHolder(view);
```

```
    }
    @Override
    public void onBindViewHolder(@NonNull Adapter.ViewHolder holder, int
position) {
        holder.cardView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(context,webView.class);
                intent.putExtra("url",modelClassArrayList.get(position).getUrl());
                context.startActivity(intent);
            }
        });
        holder.mtime.setText("Published At:-
"+modelClassArrayList.get(position).getPublishedAt());
        holder.mauthor.setText(modelClassArrayList.get(position).getAuthor());
        holder.mheading.setText(modelClassArrayList.get(position).getTitle());
        holder.mcontent.setText(modelClassArrayList.get(position).getDescription());
Glide.with(context).load(modelClassArrayList.get(position).getUrlToImage()).into(h
older.imageView);
    }
    @Override
    public int getItemCount() {
        return modelClassArrayList.size();
    }
    public class ViewHolder extends RecyclerView.ViewHolder {
        TextView mheading,mcontent,mauthor,mtime;
        CardView cardView;
        ImageView imageView;
        public ViewHolder(@NonNull View itemView) {
            super(itemView);
            mheading=itemView.findViewById(R.id.mainheading);
            mcontent=itemView.findViewById(R.id.content);
            mauthor=itemView.findViewById(R.id.author);
```

```
            mtime=itemView.findViewById(R.id.time);
            imageView=itemView.findViewById(R.id.imageview);
            cardView=itemView.findViewById(R.id.cardview);
        }
    }
}
```

ApiInterface.java

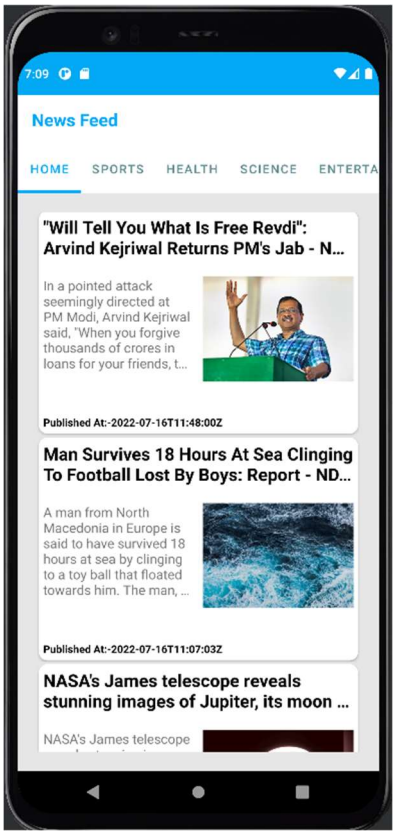
```
package com.example.newz5;
import retrofit2.Call;
import retrofit2.http.GET;
import retrofit2.http.Query
public interface ApiInterface {
    String BASE_URL="https://newsapi.org/v2/";
    @GET("top-headlines")
    Call<mainNews> getNews(
            @Query("country") String country,
            @Query("pageSize") int pagesize,
            @Query("apiKey") String apikey
    );
    @GET("top-headlines")
    Call<mainNews> getCategoryNews(
            @Query("country") String country,
            @Query("category") String category,
            @Query("pageSize") int pagesize,
            @Query("apiKey") String apikey
    );
}
```

ApiUtilities.java

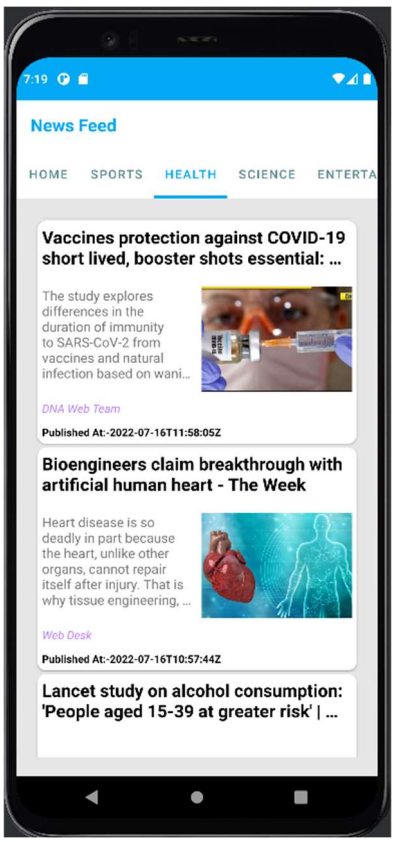
```
package com.example.newz5;
import retrofit2.Retrofit;
import retrofit2.converter.gson.GsonConverterFactory;
public class ApiUtilities {
   private static Retrofit retrofit = null;
   public static ApiInterface getApiInterface()
   {
      if(retrofit==null)
      {
         retrofit=new
Retrofit.Builder().baseUrl(ApiInterface.BASE_URL).addConverterFactory(GsonCon
verterFactory.create())
            .build();
      }
      return retrofit.create(ApiInterface.class);
   }
}
```

Chapter -5

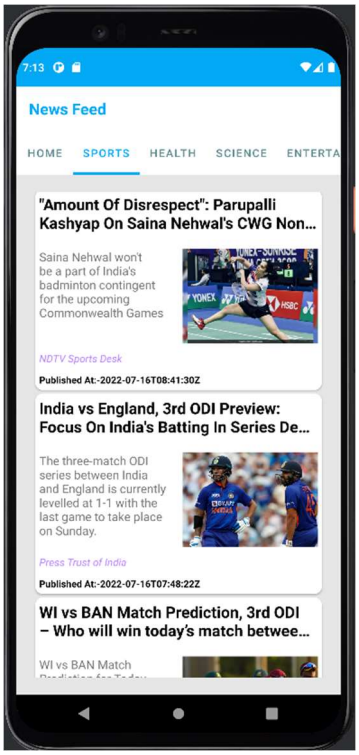
SNAPSHOTS



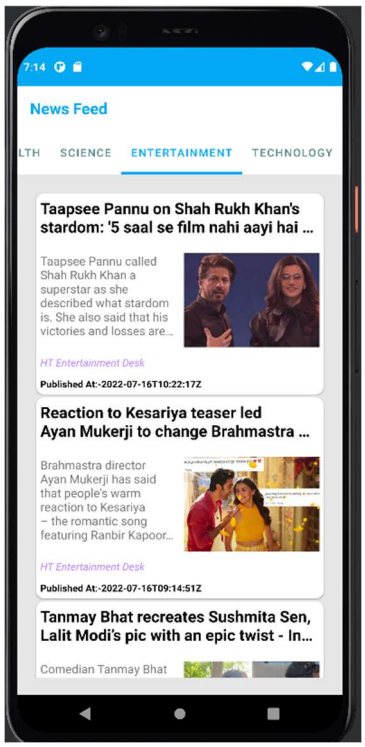
5.1 Home



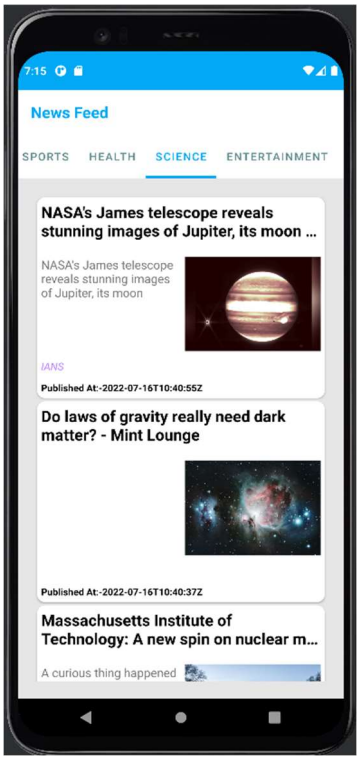
5.2 Health News



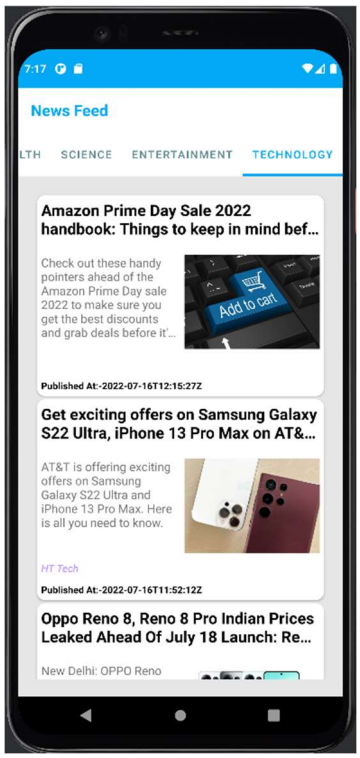
5.3 Sports news



5.4 Entertainment news

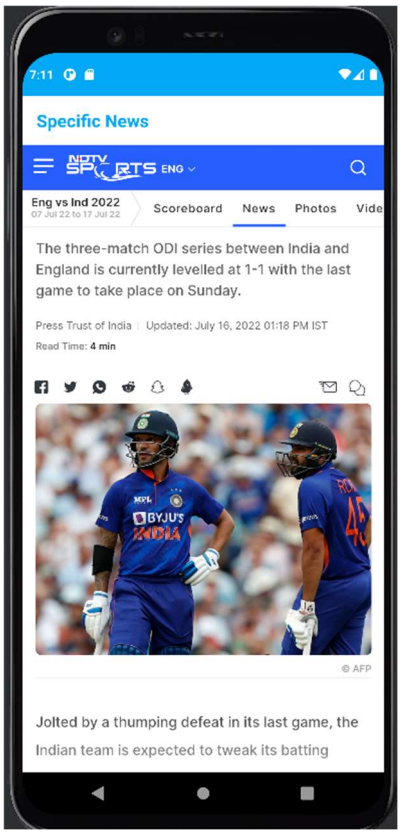


5.5 Science news



5.6 Technology news

5.9 WebView of the specific news



5.10 WebView of the specific news

Chapter -6

CONCLUSION

News reader app is a good application for the users who require instant news updates. The application is designed straight forward with all the detailed categories to be found easily in the app. This makes user experience effortless and also saves time for searching a particular news type. We have designed the app to be as simple as possible, so the user with no experience can also use the app without having to worry about anything. All the news is delivered advertisement free so the user has uninterrupted experience.

6.1 Future Enhancement

Location feature with automation can be implemented which means as user move from one city to other local news will change as per it. Data quality check needed. If API can't reach to certain article source it gives null value which can cause problem in JSON parsing. Offline can be enhanced by adding caching feature, bookmarking your favorite news article.

6.2 Bibliography

https://newsapi.org/
https://dzone.com/articles/how-to-parse-json-datafrom-a-rest-api-using-simpl
https://material.io/
https://developer.android.com/guide