

ABSTRACT

In this present age, the internet and new technologies are changing the information behavior of news reader. Instead of reading a copy of the local newspaper or watching the scheduled evening news, people increasingly turn to the internet for daily news updates. A news feed application is aimed at developing a mobile application named News Bay. This Application deals with the user who wants to read news from their mobile phones. User can select different categories in which a user is interested. the latest news will be fetched from the selected categories. The news will be fetched and displayed based on the country & the news is categorized into different categories.

Contents

Introduction.....	1
Chapter 2	2
2.1 Literature Review	2
2.1.1 News Sharing in Social Media: A Review of Current Research on News Sharing Users, Content, and Networks	4
2.1.2 Exploring mobile news reading interactions for news app personalization.....	5
2.1.3 A review and model of journalism in an age of mobile media	6
2.1.4 Impact of Smartphone News Apps on Print Media	7
2.2 Problem Statement.....	8
2.3 Objectives.....	9
Chapter 3	10
Design Methodology.....	10
3.1 Use Case Diagram.....	10
3.2 Class diagram.....	11
3.3 Sequence Diagram	12
3.4 Gant Chart.....	13
Chapter4	14
Coding and Implementation	14
4.1 Main Activity	14
4.2 DetailActivity.....	16
4.3 NewsApiResponse	17
4.4 News Headlines	18
4.5 Source	19
4.6 Request Manager	20
4.7 CustomViewHolder	21
4.8 CustomAdapter	22
4.9 Home Screen.....	23
4.10 Sports Category.....	23
4.11 Health Category	24
4.12 Entertainment Category	24
4.13 Science Category	25
4.14 Technology Category	25
4.15 Detailed News Tab.....	26
4.16 Search News Bar.....	26

Chapter 5	27
Conclusion and Future Scope	27
5.1 Conclusion	27
5.2 Future Scope	27
References.....	28

List of figures

Figure 1: Use Case Diagram of News Feed Application	10
---	----

4.11 Health Category	24
4.12 Entertainment Category	24
4.13 Science Category	25
4.14 Technology Category.....	25
4.15 Detailed News Tab.....	26
4.16 Search News Bar.....	26

Chapter 5	27
Conclusion and Future Scope	27
5.1 Conclusion	27
5.2 Future Scope	27
References.....	28

List of figures

Figure 1: Use Case Diagram of News Feed Application	10
Figure 2: Class Diagram of News Feed Application	11
Figure 3: Sequence Diagram of News Feed Application.....	12
Figure 4: Gant Chart	13

Chapter 1

Introduction

In this project we created a mobile News feed application. With the use of API called newsAPI, we will be bringing news from all around the world into the app. The news will be fetched and displayed based on the country selected & the news is categorized into different categories. A user can select any category which they are looking for. When the user is done selecting the category, then the page will automatically refresh and the news will be displayed on the news feed application.

Chapter 1

Introduction

In this project we created a mobile News feed application. With the use of API called newsAPI, we will be bringing news from all around the world into the app. The news will be fetched and displayed based on the country selected & the news is categorized into different categories. A user can select any category which they are looking for. When the user is done selecting the category, then the page will automatically refresh and the news will be displayed on the news feed application.

Chapter 2

2.1 Literature Review

This part of study literature review includes available research related to News Feed Application

Sl No	Title / Author	Year	Methodology	Result	Research Gap
1	News Sharing in Social Media: A Review of Current Research on News Sharing Users, Content, and Networks - Anna Sophie Kumpel, Veronika Karnowski, and Till Keyling	2015	The articles discussed in this literature review have been obtained by searching the Communication & Mass Media Complete (CMMC) and the ACM Digital Library databases.	The results shows that with a few exceptions it was not until 2010 that the relationship between news sharing and social media really started to gain academic attention	The general lack of context can be attributed to the small amount of qualitative and situation-related research
2	Exploring mobile news reading interactions for news app personalization - MariosConstantinides , John Dowell	2015	User's news reading preferences and behaviors; issues in the development of adaptive news app interfaces and alternative, adaptive user interfaces for each reader type.	Demonstrated a method for monitoring users' news reading behaviour and inferring news reader type from it.	The design of adaptive interfaces was unexplored.

3	Impact of Smartphone News Apps on Print Media - Dr. Elangovan N , Harshit Gupta	2015	A twin TAM was framed with Cognitive dissonance connecting attitude towards News Apps and attitude towards Printmedia and leading towards	Personalisation of news apps, and readers behaviour can be used for developing a smarter app that is able to recognise reading	TAM gives an opportunity to test only a particular or rather one new technology at a time
---	---	------	---	--	---

2	Exploring mobile news reading interactions for news app personalization - MariosConstantinides , John Dowell	2015	User's news reading preferences and behaviors; issues in the development of adaptive news app interfaces and alternative, adaptive user interfaces for each reader type.	Demonstrated a method for monitoring users' news reading behaviour and inferring news reader type from it.	The design of adaptive interfaces was unexplored.
---	--	------	--	--	---

2

11

3	Impact of Smartphone News Apps on Print Media - Dr. Elangovan N , Harshit Gupta	2015	A twin TAM was framed with Cognitive dissonance connecting attitude towards News Apps and attitude towards Printmedia and leading towards intention to use News Apps	Personalisation of news apps, and readers behaviour can be used for developing a smarter app that is able to recognise reading behaviour and adapt its display.	TAM gives an opportunity to test only a particular or rather one new technology at a time
4	A review and model of journalism in an age of mobile media - Oscar Westlund	2012	Reconfigurations of legacy media such as newspapers as well as the ways in which information and communication technologies (ICTs) are developed and used.	The production of mobile journalism has generally travelled from the human-led customisation dimension towards the technology-led customisation dimension	Using of Social Media as a source of information might not be a reliable source.

3

12

12

2.1.1 News Sharing in Social Media: A Review of Current Research on News Sharing Users, Content, and Networks

This article provides a review of scientific, peer-reviewed articles that examine the relationship between news sharing and social media in the period from 2004 to 2014. A

Methodology:

The articles discussed in this literature review have been obtained by searching the Communication & Mass Media Complete (CMMC) and the ACM Digital Library databases.

By: Anna Sophie Kumpel ,Veronika Karnowski and Till Keyling

Published On: 2015

Results: The results shows that with a few exceptions it was not until 2010 that the relationship between news sharing and social media really started to gain academic attention. News content that elicits positive or pleasant feelings is more likely to spread than negative and neutral content. However, valence is not the sole driver as it interacts with arousal. More arousing content regardless if positive or negative is shared more often. Furthermore, content that is deemed interesting originates from a trusted source or contains high informational value news factors like controversy, relevance, or unexpectedness shares a positive relation with sharing news content.

Research Gap: The general lack of context can be attributed to the small amount of qualitative and situation-related research.

13

2.1.2 Exploring mobile news reading interactions for news app personalization

This article talks about the surveyed users' news reading preferences and Behaviours and benefits of adaption for different users.

Methodology:

User's news reading preferences and behaviors; issues in the development of adaptive news app interfaces and alternative, adaptive user interfaces for each reader type.

By:MariosConstantinides , John Dowell

Published On: 2015

2.1.3 A review and model of journalism in an age of mobile media

This article explores the production of mobile news, by discussing and synthesising the findings of the contemporary literature found in the nexus of journalism and mobile media.

Methodology:

This study has looked at Reconfigurations of legacy media such as newspapers as well as the ways in which information and communication technologies (ICTs) are developed and used

By: - Oscar Westlund

Published on: 2012

Results: The production of mobile journalism has generally travelled from the human-led customisation dimension towards the technology-led customisation dimension.

Research Gap: Using of Social Media as a source of information might not be a reliable source.

2.1.4 Impact of Smartphone News Apps on Print Media

The idea of this research was to study the on-going recent changes which the print media magazine is witnessing over the past few years or rather will witness in the near future due to advancement in the technology.

Methodology:

A twin TAM was framed with Cognitive dissonance connecting attitude towards News Apps and attitude towards Printmedia and leading towards intention to use News Apps.

By: - Dr. Elangovan N , Harshit Gupta

Published on: 2015

Results: Personalisation of news apps, and readers behaviour can be used for developing a smarter app that is able to recognise reading behaviour and adapt its display. The study found that the intention to use news apps is influenced by the attitude towards news apps and attitude towards print magazine through dissonance. Hence, dissonance acts as a deciding factor for the selection of a variable depending upon the mental conflict, which each individual undergoes for deciding to use it. Attitude towards print magazine creates more dissonance, whereas attitude towards news app reduces dissonance.

Research Gap: TAM gives an opportunity to test only a particular or rather one new technology at a time.

factor for the selection of a variable depending upon the mental conflict, which each individual undergoes for deciding to use it. Attitude towards print magazine creates more dissonance, whereas attitude towards news app reduces dissonance.

Research Gap: TAM gives an opportunity to test only a particular or rather one new technology at a time.

7

16

2.2 Problem Statement

Some of the similar applications we reviewed had complaints about their user interface being clustered with ad banners, offer walls, which made navigation difficult. While some news applications only provided news from only a specific source which was very limited and some applications even provided only a limited viewing before asking the users for a subscription to view more content.

8

17

2.3 Objectives

The objective of our project is:

- To create a mobile application that displays recent news.
- Provide news on the go
- Easy to use

News Feed Application Development

Figure 4: Gant Chart

13

22

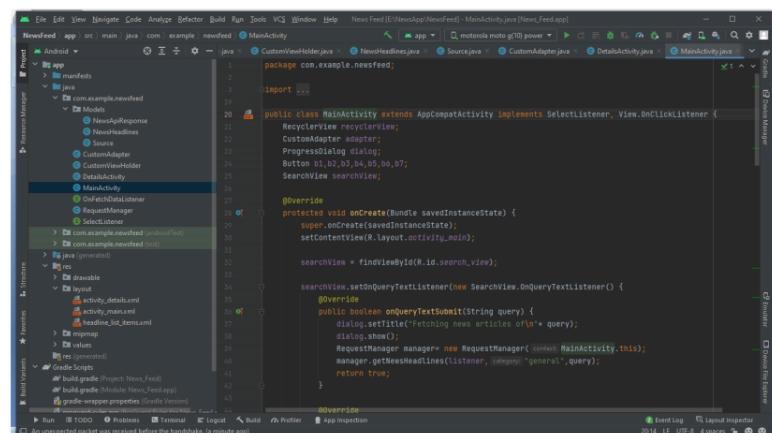
Chapter4

Coding and Implementation

The outcome of our project is we created an application with following codes and activities.

4.1 Main Activity

The Main Activity is where we initialize the activity .Here we create an object for RequestManager to call to get the news headlines and contents and a listener to fetch the data to be displayed. We also create and initialize a few buttons for a selection of categories and a Searchview for searching the news according to the specified query.



```

package com.example.newsfeed;
import ...;

public class MainActivity extends AppCompatActivity implements SelectListener, View.OnClickListener {
    ...
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        searchView = findViewById(R.id.search_view);
        searchView.setQueryTextListener(new SearchView.OnQueryTextListener() {
            ...
            public boolean onQueryTextSubmit(String query) {
                progressDialog.setTitle("Fetching news articles on "+query);
                progressDialog.show();
                RequestManager manager = new RequestManager((Context) MainActivity.this);
                manager.getNewsHeadlines(listener, query);
                return true;
            }
        });
    }
}

```

14

23

```
* res (generated)
> java
> build.gradle (Project: News_Feed)
> build.gradle (Module: News_FeedApp)
> gradle-wrapper.properties (Gradle Version)
Build Events
Build Scripts
Build Variants
Run
TODO
Problems
Terminal
Logcat
Build
App Inspection
2014 LF UTF-8 4 spaces
An unexpected packet was received before the handshake. (2 minutes ago)
```

```
recyclerView.setHasFixedSize(true);
recyclerView.setLayoutManager(new GridLayoutManager(context, spanCount));
adapter = new CustomAdapter(context, this_list, listener);
recyclerView.setAdapter(adapter);
```

15

24

```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Windows Help News Feed [E:\NewsAppl\NewsFeed] - MainActivity.java [News_Feed app]
Newsfeed app inc main java com example newsfeed MainActivity
Project Android
> app
> manifests
> java
> res
> com.example.newsfeed
> Models
> NewsApiResponse
> NewsHeadlines
> Source
> CustomAdapter
> CustomViewHolder
> DetailsActivity
> MainActivity
> UnsplashDataListener
> UnsplashImage
> SelectListener
> com.example.newsfeed (src-test)
> com.example.newsfeed (test)
> java (generated)
> layout
> drawable
> activity_details.xml
> activity_main.xml
> activity_main_items.xml
> anim
> res (generated)
> build.gradle (Project: News_Feed)
> build.gradle (Module: News_FeedApp)
> gradle-wrapper.properties (Gradle Version)
Build Events
Build Scripts
Build Variants
Run
TODO
Problems
Terminal
Logcat
Build
App Inspection
2014 LF UTF-8 4 spaces
An unexpected packet was received before the handshake. (2 minutes ago)
```

```
recyclerView.setHasFixedSize(true);
recyclerView.setLayoutManager(new GridLayoutManager(context, spanCount));
adapter = new CustomAdapter(context, this_list, listener);
recyclerView.setAdapter(adapter);
```

```
@Override
public void OnnewsClicked(NewsHeadlines headlines) {
    startActivity(new Intent(getApplicationContext(), MainActivity.this, DetailsActivity.class)
        .putExtra("name", "data", headlines));
}
```

```
@Override
public void onClick(View view) {
    Button button = (Button) view;
    String category = button.getText().toString();
    dialog.setTitle("Fetching news articles of " + category);
    dialog.show();
    RequestManager manager = new RequestManager(context);
    manager.getHeadlines(listener, category, query);
}
```

4.2 DetailActivity

The DetailActivity class displays the news in a detailed format when the specific news on the cardview from the MainActivity is selected. Here we capture the data from the intent we have passed from the MainActivity.

```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Windows Help News Feed [E:\NewsAppl\NewsFeed] - DetailActivity.java [News_Feed app]
Newsfeed app inc main java com example newsfeed DetailActivity
Project Android
> app
> manifests
> java
> com.example.newsfeed
> Models
> NewsApiResponse
> NewsHeadlines
> Source
> CustomAdapter
> CustomViewHolder
> DetailsActivity
> MainActivity
> UnsplashDataListener
> UnsplashImage
> SelectListener
> com.example.newsfeed (src-test)
> com.example.newsfeed (test)
> java (generated)
> layout
> drawable
> activity_main.xml
> activity_main_items.xml
> anim
> res
> build.gradle (Project: News_Feed)
> build.gradle (Module: News_FeedApp)
> gradle-wrapper.properties (Gradle Version)
Build Events
Build Scripts
Build Variants
Run
TODO
Problems
Terminal
Logcat
Build
App Inspection
15:24 LF UTF-8 4 spaces
An unexpected packet was received before the handshake. (10 minutes ago)
```

```
public class DetailActivity extends AppCompatActivity {
    NewsHeadlines headlines;
    TextView txt_title, txt_author, txt_time, txt_detail, txt_content;
    ImageView img_news;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_detail);
        txt_title=findViewById(R.id.text_detail_title);
        txt_author=findViewById(R.id.text_detail_author);
        txt_time=findViewById(R.id.text_detail_time);
        txt_detail=findViewById(R.id.text_detail_detail);
        txt_content=findViewById(R.id.text_detail_content);
        img_news=findViewById(R.id.img_detail_news);
        headlines = (NewsHeadlines) getIntent().getSerializableExtra("name: "data");
        txt_title.setText(headlines.getTitle());
        txt_author.setText(headlines.getAuthor());
        txt_time.setText(headlines.getPublishedAt());
        txt_detail.setText(headlines.getDescription());
        txt_content.setText(headlines.getContent());
        Picasso.get().load(headlines.getThumbnail()).into(img_news);
    }
}
```

16

25

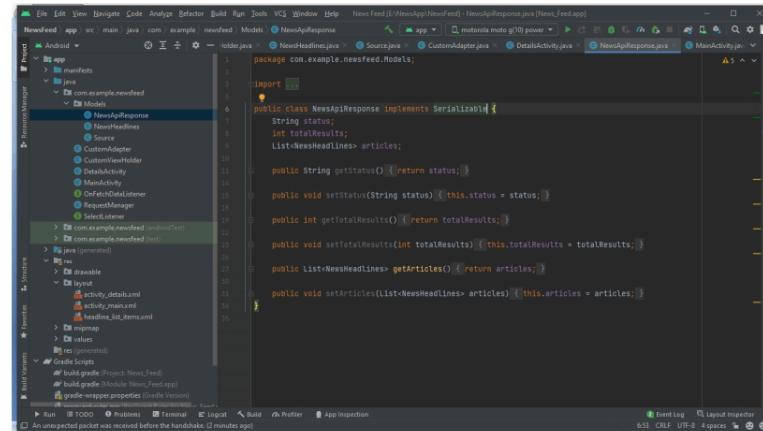
4.3 NewsApiResponse

The NewsApiResponse class has two response objects the status and totalResults. The status

25

4.3 NewsApiResponse

The NewsApiResponse class has two response objects the status and totalResults. The status shows if the request was successful or not. And the totalResults show the total number of results available for our request.



```

package com.example.newsfeed.Models;

import java.util.List;

public class NewsApiResponse implements Serializable {
    String status;
    int totalResults;
    List<NewsHeadlines> articles;

    public String getStatus() { return status; }

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

    public int getTotalResults() { return totalResults; }

    public void setTotalResults(int totalResults) { this.totalResults = totalResults; }

    public List<NewsHeadlines> getArticles() { return articles; }

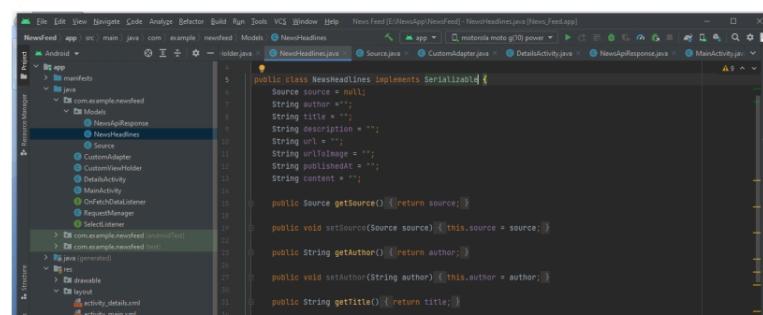
    public void setArticles(List<NewsHeadlines> articles) { this.articles = articles; }
}

```

26

4.4 News Headlines

The NewsHeadlines Class fetches the objects id and a display name from the source class. It also contains response objects which fetches the author, title, description, url, the date and time of publishing and the unformatted content of the article, where available which is truncated to 200 char.



```

public class NewsHeadlines implements Serializable {
    String id;
    String displayTitle;
    String author = "";
    String title = "";
    String description = "";
    String url = "";
    String urlToImage = "";
    String publishedAt = "";
    String content = "";

    public Source getSource() { return source; }

    public void setSource(Source source) { this.source = source; }

    public String getAuthor() { return author; }

    public void setAuthor(String author) { this.author = author; }

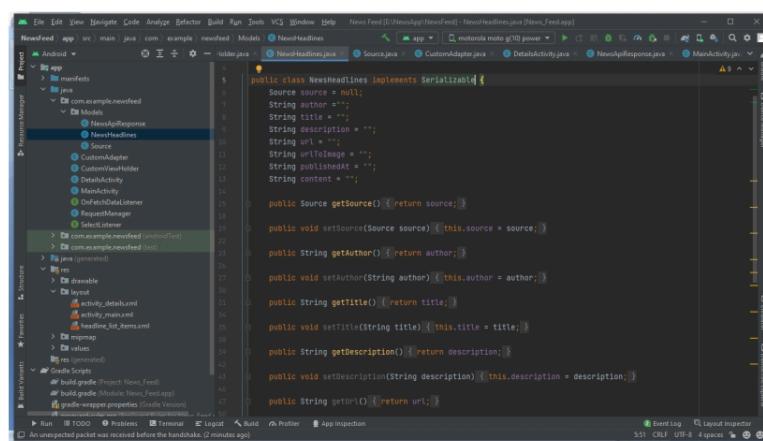
    public String getTitle() { return title; }
}

```

26

4.4 News Headlines

The NewsHeadlines Class fetches the objects id and a display name from the source class. It also contains response objects which fetches the author, title, description, url, the date and time of publishing and the unformatted content of the article, where available which is truncated to 200 char .



```

public class NewsHeadlines implements Serializable {
    Source source = null;
    String author = "";
    String title = "";
    String description = "";
    String url = "";
    String urlToImage = "";
    String publishedAt = "";
    String content = "";

    public Source getSource() { return source; }

    public void setSource(Source source) { this.source = source; }

    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; }

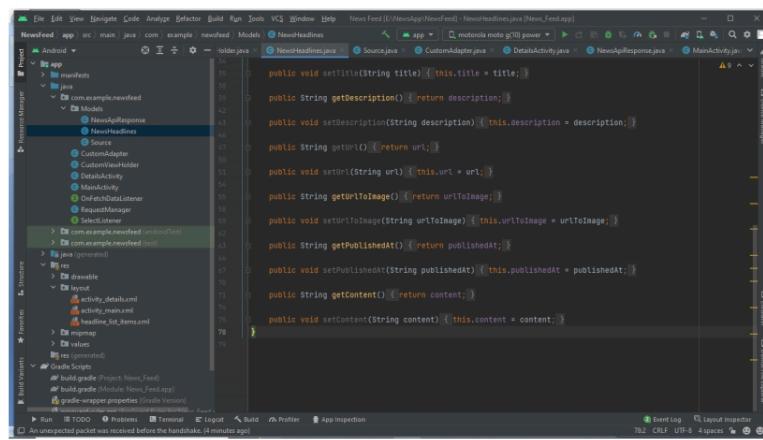
    public String getPublishedAt() { return publishedAt; }

    public void setPublishedAt(String publishedAt) { this.publishedAt = publishedAt; }

    public String getContent() { return content; }

    public void setContent(String content) { this.content = content; }
}

```



```

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; }

public String getContent() { return content; }

public void setContent(String content) { this.content = content; }

```

27

4.5 Source

The Source class has two response objects id and name which fetches the news sources or blogs we want headlines from.

27

4.5 Source

The Source class has two response objects id and name which fetches the news sources or blogs we want headlines from.

```

package com.example.newsfeed.Models;

import java.io.Serializable;

public class Source implements Serializable {
    String id = "";
    String name = "";

    public String getId() { return id; }

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

    public String getName() { return name; }

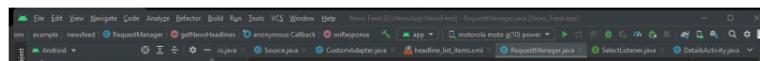
    public void setName(String name) { this.name = name; }
}

```

28

4.6 Request Manager

The RequestManager class manages all the Api calls. We use a retrofit function here for authenticating and interacting with APIs and sending network requests with OkHttp. Using Retrofit makes it easy to parse API response and use it in your application. It has built-in GSON converter that can automatically parse HTTP response into an Object or any other types in Java.



4.6 Request Manager

The RequestManager class manages all the Api calls. We use a retrofit function here for authenticating and interacting with APIs and sending network requests with OkHttp. Using Retrofit makes it easy to parse API response and use it in your application. It has built-in GSON converter that can automatically parse HTTP response into an Object or any other types in Java.

```

1 package com.example.newsfeed;
2
3 import android.os.AsyncTask;
4 import android.util.Log;
5 import android.widget.ArrayAdapter;
6 import android.widget.ListView;
7 import android.widget.TextView;
8
9 import java.util.ArrayList;
10 import java.util.List;
11
12 import retrofit.Call;
13 import retrofit.http.GET;
14 import retrofit.http.Query;
15
16 public interface CallNewsApi {
17     @GET("top-headlines")
18     Call<NewsApiResponse> callHeadlines(
19             @Query("country") String country,
20             @Query("category") String category,
21             @Query("q") String query,
22             @Query("apiKey") String apiKey);
23 }

```

```

1 package com.example.newsfeed;
2
3 import android.os.AsyncTask;
4 import android.util.Log;
5 import android.widget.ArrayAdapter;
6 import android.widget.ListView;
7 import android.widget.TextView;
8
9 import java.util.ArrayList;
10 import java.util.List;
11
12 import retrofit.Call;
13 import retrofit.http.GET;
14 import retrofit.http.Query;
15
16 public class RequestManager {
17     Context context;
18     Retrofit retrofit = new Retrofit.Builder()
19             .baseUrl("https://newsapi.org/v2/")
20             .addConverterFactory(GsonConverterFactory.create())
21             .build();
22
23     public void getNewsHeadlines(OnFetchDataListener listener, String category, String query) {
24         try {
25             Call<NewsApiResponse> callNewsApi = retrofit.create(CallNewsApi.class);
26             Call<NewsApiResponse> call = callNewsApi.callHeadlines(country, category, query, "7e73538440a094c8bf9");
27
28             try {
29                 call.enqueue(new Callback<NewsApiResponse>() {
30                     @Override
31                     public void onResponse(Call<NewsApiResponse> call, Response<NewsApiResponse> response) {
32                         if(response.isSuccessful()){
33                             Toast.makeText(context, "Success!!", Toast.LENGTH_SHORT).show();
34
35                             listener.onFetchData(response.body().getArticles(), response.message());
36                         }
37                     }
38
39                     @Override
40                     public void onFailure(Call<NewsApiResponse> call, Throwable e) {
41                         listener.onError("Request Failed !!");
42                     }
43                 });
44             } catch (Exception e) {
45                 e.printStackTrace();
46             }
47         } catch (Exception e) {
48             Log.e("Error", "Error while fetching news");
49         }
50     }
51
52     public RequestManager(Context context) { this.context = context; }
53
54     public interface CallNewsApi {
55         @GET("top-headlines")
56         Call<NewsApiResponse> callHeadlines(
57             @Query("country") String country,
58             @Query("category") String category,
59             @Query("q") String query,
60             @Query("apiKey") String apiKey);
61     }
62 }

```

4.7 CustomViewHolder

The CustomViewHolder is a class that extends to a RecyclerView's ViewHolder to store the headlines inside the cardview format .

29

4.7 CustomViewHolder

The CustomViewHolder is a class that extends to a RecyclerView's ViewHolder to store the headlines inside the cardview format .

```

package com.example.newsfeed;

import ...

public class CustomViewHolder extends RecyclerView.ViewHolder {
    TextView text_title, text_source;
    ImageView img_headline;
    CardView cardView;
    public CustomViewHolder(@NonNull View itemView) {
        super(itemView);
        text_title=itemView.findViewById(R.id.text_title);
        text_source=itemView.findViewById(R.id.text_source);
        img_headline=itemView.findViewById(R.id.img_headline);
        cardView=itemView.findViewById(R.id.main_container);
    }
}

```

30

4.8 CustomAdapter

The CustomAdapter is a class that extends to a RecyclerView's adapter where we pass the CustomViewHolder we created. We create objects and list of headlines and implement the methods and constructor.

```

package com.example.newsfeed;

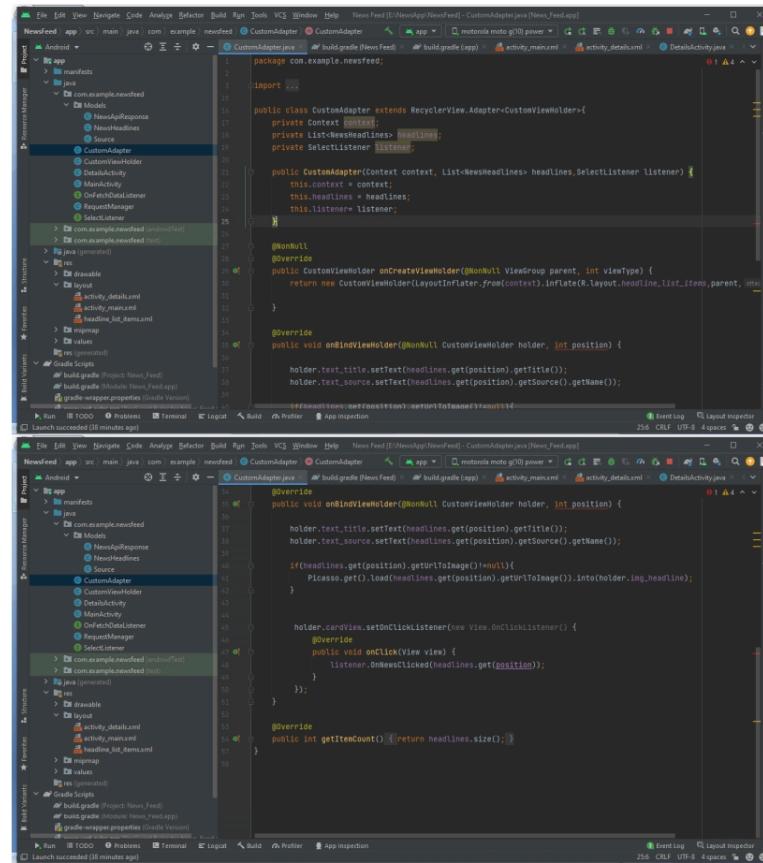
import ...

public class CustomAdapter extends RecyclerView.Adapter<CustomViewHolder>{
    private Context context;
    private List<NewsHeadlines> headlines;
    private SelectListener listener;
    public CustomAdapter(Context context, List<NewsHeadlines> headlines,SelectListener listener) {
        this.context = context;
        this.headlines = headlines;
        this.listener= listener;
    }
    @Override
    public CustomViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        return new CustomViewHolder(LayoutInflater.from(context).inflate(R.layout.list_items,parent, false));
    }
}

```

4.8 CustomAdapter

The CustomAdapter is a class that extends to a RecyclerView's adapter where we pass the CustomViewHolder we created. We create objects and list of headlines and implement the methods and constructor.



```
package com.example.newsfeed;

import ...

public class CustomAdapter extends RecyclerView.Adapter<CustomViewHolder> {
    private Context context;
    private List<NewsHeadlines> headlines;
    private SelectListener listener;

    public CustomAdapter(Context context, List<NewsHeadlines> headlines, SelectListener listener) {
        this.context = context;
        this.headlines = headlines;
        this.listener = listener;
    }

    @Override
    public CustomViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        return new CustomViewHolder(LayoutInflater.from(context).inflate(R.layout.headline_list_items,parent, false));
    }

    @Override
    public void onBindViewHolder(@NonNull CustomViewHolder holder, int position) {
        holder.text_title.setText(headlines.get(position).getTitle());
        holder.text_source.setText(headlines.get(position).getSource().getName());

        if(headlines.get(position).getImageUrl()!=null){
            Picasso.get().load(headlines.get(position).getImageUrl()).into(holder.img_headline);
        }

        holder.cardView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                listener.NewsClicked(headlines.get(position));
            }
        });
    }

    @Override
    public int getItemCount() {
        return headlines.size();
    }
}
```

22

31

The implementation of the coding produces the following results/outcomes

4.9 Home Screen

The Home Screen is the main screen of the app from which user can navigate to other activities.



4.10 Sports Category

The Sports Category displays news headlines related to sports.



The implementation of the coding produces the following results/outcomes

4.9 Home Screen

The Home Screen is the main screen of the app from which user can navigate to other activities.



4.10 Sports Category

The Sports Category displays news headlines related to sports.



23

32

4.11 Health Category

The Health Category displays news headlines related to health.



4.12 Entertainment Category

The Entertainment Category displays news headlines related to entertainments.

