SINHGAD INSTITUTE OF TECHNOLOGY LONAVALA

SAVITRIBAI PHULE PUNE UNIVERSITY

A

PROJECT REPORT

ON

KHABAR

SUBMITTED IN FULFILLMENT FOR SUBMISSION

OF

Skill Development Lab

SUBMITTED BY

Shashwat Kumar	TEC51
Shyam Kachru Bodke	TEC67
Prateek Singh	TEC69

DEPARTMENT OF COMPUTER ENGINEERING

Sinhgad Institute of Technology, Kusgaon, Lonavala

Savitribai Phule Pune University

SINHGAD INSTITUTE OF TECHNOLOGYLONAVALA

SAVITRIBAI PHULE PUNE UNIVERSITY

CERTIFICATE

This is certified that the Mini Project Entitled

KHABAR

SUBMITTED BY

Prateek Singh	(T150424339)
Shashwat Kumar	(T150424358)
Shyam Kachru Bodke	(T150424406)

Prof. S.N. Wandre
Subject Teacher

Dr. S. D. Babar	**Dr. M. S. Gaikwad**
HOD Computer Engineering	Principal

Table of Contents

Introduction

Project Title: KHABAR

We are a Team of Three Third Year Computer Engineering Student From Sinhgad Institute Of Technology Affiliated to Savitribai Phule Pune University (SPPU) Formely Known As University Of Pune. As a part of our course we have been working on Software Develop - ment project in Teams. The project aim is to introduce software Engineering Concept and Development to the students by matching their skills and assigning the Project to each member of the team,through this each student can contribute their skills in Project. This section gives a scope description and overview of everything included in this Project Report.Also,the purpose for this document is described and system overview along with Goal and Vision are listed.

Purpose :

The purpose of this Document is to give detailed Description of the project that we had made On a news application . The name of our News app is KHABAR . It will provide us all the Information regarding the Current affairs , news like Sports, Entertainment, World, Business , Society , Fashion , Science news etc. It will also explain the interface and Interaction with the API that we have use to fetch all the news from their Server. This Document is primarily Intended to anyone how want to get an overview of current news that is happening the current Scenario with a immediate information of the data available on article, By seeing the headline Of the news the user can get a idea of what's happening in our country or world right now.

Objective

“Design and produce a mobile Android application that uses one or more open public environmental datasets to illustrate how open data applications can benefit from the capabilities of mobile devices.”

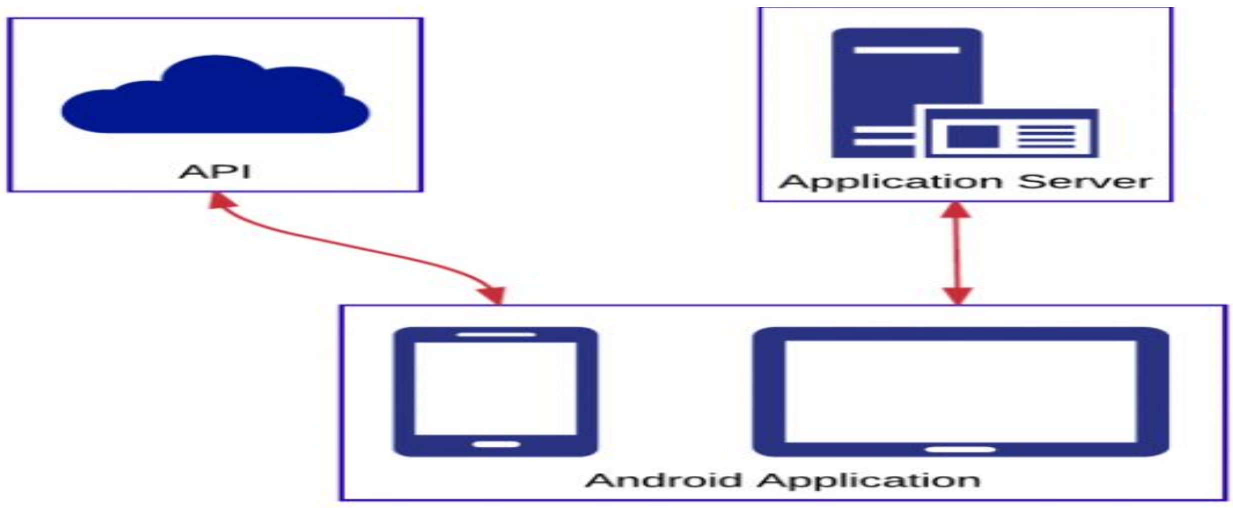
Before deciding on what the application was going to be, we isolated features unique to mobile applications that we would like to include in our app including API fetching, use of the Button and Status bar, and push notifications. It was really interesting to see a range of different ideas resulting from brainstorming.

The discussion and evaluation was quite constructive, leaving us Many different ideas.

The Project has been split into Front-end and back-end teams. The front-end team is responsible for designing and Implementation of the UI and data visualization using Android Studio.The Back -end team, of which I am a part,is querying and filtering the data from the guardian news data API Which will be displayed in the app.

When working with new technologies it is criticalto do a lot of research and get familiar with what is available. We have spends hours of time in collecting information and would to implement this new technologies.

System Architecture Layout



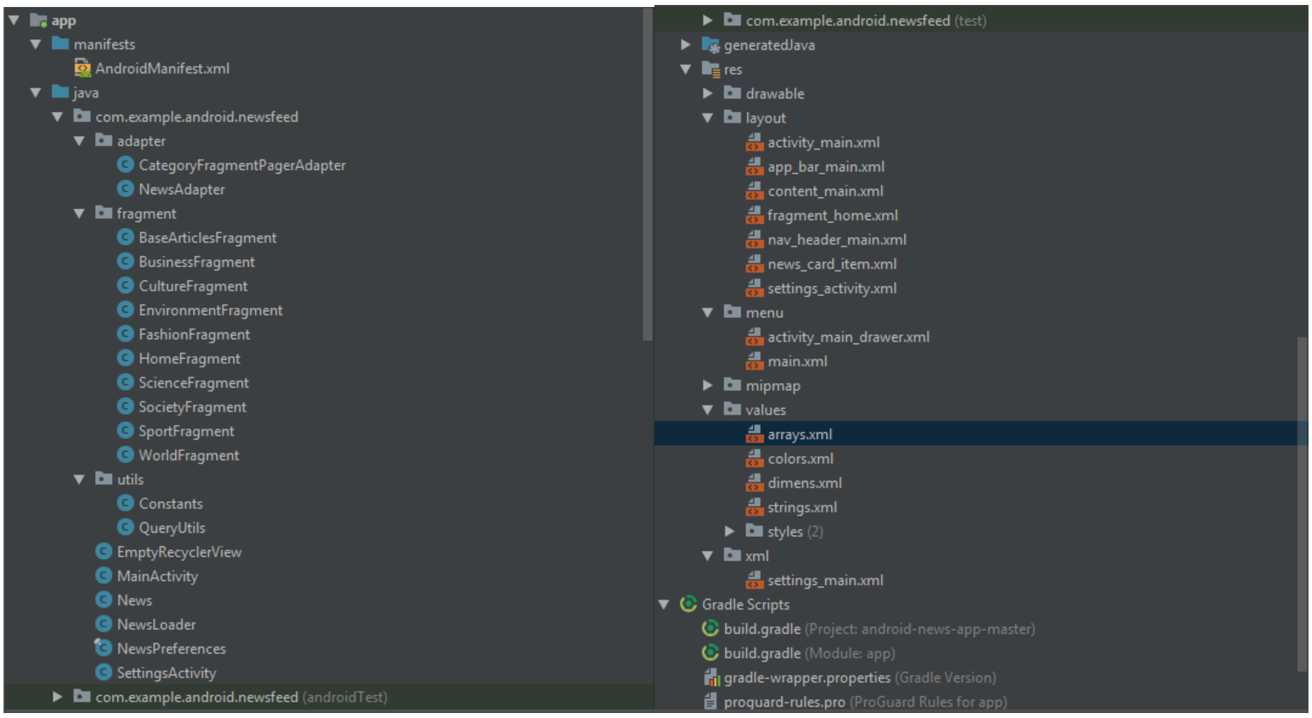
Motivation

Tablet devices gain greater and greater importance around the world. The portable design attracts users for reading information and watching videos. Although there are still limitations of tablet devices and it is not possible to replace the traditional personal computer, the tablets can be further utilized in many aspects.

Seeing the advantages of the portability and the ease of reading books, the tablets/ mobile devices should be promoted to perform academic objectives. We would like to introduce it to research activities. However, the design of Android devices aims at the simplicity. As mentioned previously, the tablet is different from personal computer. Too complex tasks should not be implemented. So what can be done without a substantial keyboard? Yes, the task should be performed without large amount of input data. We noticed that the large screen of tablet eases the difficulties of reading books. Papers are important components of research activity, and that is what can be easily done without a substantial keyboard.

Will researching on various application available on internet we decided to make an application that will be easy for user to interact and understand the application in an easy way . We have seen other Application like news hunt , daily hunt , TV today news etc we have got ideas that we can make an Application that can be used to implement in the application this news app is made for immediate Information of news in a short period of time further information can we fetch from the news article; That can be fetch by our api we have used is the Guardian news Api ,Guardian is a firm for news and Article they pulish on their website and their forum.

Project Code



```
▼ app
  ▼ manifests
      AndroidManifest.xml
  ▼ java
    ▼ com.example.android.newsfeed
      ▼ adapter
          CategoryFragmentPagerAdapter
          NewsAdapter
      ▼ fragment
          BaseArticlesFragment
          BusinessFragment
          CultureFragment
          EnvironmentFragment
          FashionFragment
          HomeFragment
          ScienceFragment
          SocietyFragment
          SportFragment
          WorldFragment
      ▼ utils
          Constants
          QueryUtils
        EmptyRecyclerView
        MainActivity
        News
        NewsLoader
        NewsPreferences
        SettingsActivity
    ▶ com.example.android.newsfeed (androidTest)
    ▶ com.example.android.newsfeed (test)
  ▶ generatedJava
  ▼ res
    ▶ drawable
    ▼ layout
        activity_main.xml
        app_bar_main.xml
        content_main.xml
        fragment_home.xml
        nav_header_main.xml
        news_card_item.xml
        settings_activity.xml
    ▼ menu
        activity_main_drawer.xml
        main.xml
    ▶ mipmap
    ▼ values
        arrays.xml
        colors.xml
        dimens.xml
        strings.xml
      ▶ styles (2)
    ▼ xml
        settings_main.xml
▼ Gradle Scripts
    build.gradle (Project: android-news-app-master)
    build.gradle (Module: app)
    gradle-wrapper.properties (Gradle Version)
    proguard-rules.pro (ProGuard Rules for app)
```

AndroidManifest.xml

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

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

<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/AppTheme"
tools:ignore="GoogleAppIndexingWarning">
<activity
android:name=".MainActivity"
android:label="@string/app_name"
android:theme="@style/AppTheme.NoActionBar">
<intent-filter>
```

```
<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>
<activity
android:name=".SettingsActivity"
android:theme="@style/SettingsTheme"
android:label="@string/action_settings"
android:screenOrientation="portrait">
<meta-data
android:name="android.support.PARENT_ACTIVITY"
android:value="com.example.android.newsfeed.MainActivity" />
</activity>
</application>

</manifest>
```

Adaptor:

CategoryFragmentPagerAdapter.java

```
package com.example.android.newsfeed.adapter;

import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentManager;
import android.content.Context;
import android.support.v4.app.FragmentPagerAdapter;

import com.example.android.newsfeed.R;
import com.example.android.newsfeed.fragment.BusinessFragment;
import com.example.android.newsfeed.fragment.CultureFragment;
import com.example.android.newsfeed.fragment.EnvironmentFragment;
import com.example.android.newsfeed.fragment.FashionFragment;
import com.example.android.newsfeed.fragment.HomeFragment;
import com.example.android.newsfeed.fragment.ScienceFragment;
import com.example.android.newsfeed.fragment.SocietyFragment;
import com.example.android.newsfeed.fragment.SportFragment;
import com.example.android.newsfeed.fragment.WorldFragment;
import com.example.android.newsfeed.utils.Constants;

/**
 * Provides the appropriate {@link Fragment} for a view pager.
 */

public class CategoryFragmentPagerAdapter extends FragmentPagerAdapter {

/** Context of the app */
private Context mContext;

/**
     * Create a new {@link CategoryFragmentPagerAdapter} object.
     *
     * @param context is the context of the app
     * @param fm is the fragment manager that will keep each fragment's
```

```
state in the adapter
      * across swipes.
      */
public CategoryFragmentPagerAdapter(Context context, FragmentManager fm)
{
super(fm);
mContext = context;
}

/**
      * Return the {@link Fragment} that should be displayed for the
given page number.
      */
@Override
public Fragment getItem(int position) {
switch (position) {
case Constants.HOME:
return new HomeFragment();
            case Constants.WORLD:
return new WorldFragment();
            case Constants.SCIENCE:
return new ScienceFragment();
            case Constants.SPORT:
return new SportFragment();
            case Constants.ENVIRONMENT:
return new EnvironmentFragment();
            case Constants.SOCIETY:
return new SocietyFragment();
            case Constants.FASHION:
return new FashionFragment();
            case Constants.BUSINESS:
return new BusinessFragment();
            case Constants.CULTURE:
return new CultureFragment();
            default:
return null;
}
    }

/**
      * Return the total number of pages.
      */
@Override
public int getCount() {
return 9;
}

/**
      * Return page title of the tap
      */
@Override
public CharSequence getPageTitle(int position) {
int titleResId;
        switch (position) {
case Constants.HOME:
                titleResId = R.string.ic_title_home;
                break;
            case Constants.WORLD:
                titleResId = R.string.ic_title_world;
                break;
            case Constants.SCIENCE:
                titleResId = R.string.ic_title_science;
                break;
            case Constants.SPORT:
                titleResId = R.string.ic_title_sport;
                break;
            case Constants.ENVIRONMENT:
                titleResId = R.string.ic_title_environment;
                break;
            case Constants.SOCIETY:
                titleResId = R.string.ic_title_society;
                break;
            case Constants.FASHION:
                titleResId = R.string.ic_title_fashion;
                break;
            case Constants.BUSINESS:
                titleResId = R.string.ic_title_business;
                break;
            default:
                titleResId = R.string.ic_title_culture;
                break;
}
return mContext.getString(titleResId);
}
}
```

NewsAdapter.java

```
package com.example.android.newsfeed.adapter;

import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.Color;
import android.net.Uri;
import android.preference.PreferenceManager;
import android.support.v7.widget.CardView;
import android.support.v7.widget.RecyclerView;
import android.text.Html;
import android.text.format.DateUtils;
import android.util.Log;
import android.util.TypedValue;
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 com.example.android.newsfeed.News;
import com.example.android.newsfeed.R;
```

```java
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.List;
import java.util.Locale;
import java.util.TimeZone;

/**
 * A {@link NewsAdapter} can provide a card item layout for each news in
the data source
 * ( a list of {@link News} objects).
 */

public class NewsAdapter extends
RecyclerView.Adapter<NewsAdapter.ViewHolder> {
private Context mContext;
    private List<News>mNewsList;
    private SharedPreferences sharedPrefs;

/**
     * Constructs a new {@link NewsAdapter}
     * @param context of the app
     * @param newsList is the list of news, which is the data source of
the adapter
     */
public NewsAdapter(Context context, List<News> newsList) {
mContext = context;
mNewsList = newsList;
}

@Override
public NewsAdapter.ViewHolder onCreateViewHolder(ViewGroup parent, int
viewType) {
        View v =
LayoutInflater.from(parent.getContext()).inflate(R.layout.news_card_item
, parent, false);
        return new ViewHolder(v);
}

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

class ViewHolder extends RecyclerView.ViewHolder {
private TextView titleTextView;
        private TextView sectionTextView;
        private TextView authorTextView;
        private TextView dateTextView;
        private ImageView thumbnailImageView;
        private ImageView shareImageView;
        private TextView trailTextView;
        private CardView cardView;

ViewHolder(View itemView) {
```

```
super(itemView);
titleTextView = itemView.findViewById(R.id.title_card);
sectionTextView = itemView.findViewById(R.id.section_card);
authorTextView = itemView.findViewById(R.id.author_card);
dateTextView = itemView.findViewById(R.id.date_card);
thumbnailImageView = itemView.findViewById(R.id.thumbnail_image_card);
shareImageView = itemView.findViewById(R.id.share_image_card);
trailTextView = itemView.findViewById(R.id.trail_text_card);
cardView = itemView.findViewById(R.id.card_view);
}
    }

@Override
public void onBindViewHolder(ViewHolder holder, int position) {
sharedPrefs = PreferenceManager.getDefaultSharedPreferences(mContext);

// Change the color theme of Title TextView by using the user's stored
preferences
setColorTheme(holder);

// Change text size of TextView by using the user's stored preferences
setTextSize(holder);

// Find the current news that was clicked on
final News currentNews = mNewsList.get(position);

holder.titleTextView.setText(currentNews.getTitle());
holder.sectionTextView.setText(currentNews.getSection());
// If the author does not exist, hide the authorTextView
if (currentNews.getAuthor() == null) {
            holder.authorTextView.setVisibility(View.GONE);
} else {
            holder.authorTextView.setVisibility(View.VISIBLE);
holder.authorTextView.setText(currentNews.getAuthor());
}

// Get time difference between the current date and web publication date
and
        // set the time difference on the textView
holder.dateTextView.setText(getTimeDifference(formatDate(currentNews.get
Date())));

// Get string of the trailTextHTML and convert Html text to plain text
        // and set the plain text on the textView
String trailTextHTML = currentNews.getTrailTextHtml();
holder.trailTextView.setText(Html.fromHtml(Html.fromHtml(trailTextHTML).
toString()));

// Set an OnClickListener to open a website with more information about
the selected article
holder.cardView.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
// Convert the String URL into a URI object (to pass into the Intent
constructor)
Uri newsUri = Uri.parse(currentNews.getUrl());
// Create a new intent to view the news URI
Intent websiteIntent = new Intent(Intent.ACTION_VIEW, newsUri);

// Send the intent to launch a new activity
mContext.startActivity(websiteIntent);
}
        });

        if (currentNews.getThumbnail() == null) {
            holder.thumbnailImageView.setVisibility(View.GONE);
} else {
            holder.thumbnailImageView.setVisibility(View.VISIBLE);
// Load thumbnail with glide
Glide.with(mContext.getApplicationContext())
                    .load(currentNews.getThumbnail())
                    .into(holder.thumbnailImageView);
}
// Set an OnClickListener to share the data with friends via email or
social networking
holder.shareImageView.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
                shareData(currentNews);
}
        });
}

/**
     * Set the user preferred color theme
     */
private void setColorTheme(ViewHolder holder) {
// Get the color theme string from SharedPreferences and check for the
value associated with the key
String colorTheme = sharedPrefs.getString(
mContext.getString(R.string.settings_color_key),
mContext.getString(R.string.settings_color_default));

// Change the background color of titleTextView by using the user's
stored preferences
if
(colorTheme.equals(mContext.getString(R.string.settings_color_white_valu
e))) {
            holder.titleTextView.setBackgroundResource(R.color.white);
holder.titleTextView.setTextColor(Color.BLACK);
}else if
(colorTheme.equals(mContext.getString(R.string.settings_color_sky_blue_v
alue))) {

holder.titleTextView.setBackgroundResource(R.color.nav_bar_start);
holder.titleTextView.setTextColor(Color.WHITE);
} else if
(colorTheme.equals(mContext.getString(R.string.settings_color_dark_blue_
value))) {

holder.titleTextView.setBackgroundResource(R.color.color_app_bar_text);
holder.titleTextView.setTextColor(Color.WHITE);
} else if
(colorTheme.equals(mContext.getString(R.string.settings_color_violet_val
ue))) {
            holder.titleTextView.setBackgroundResource(R.color.violet);
holder.titleTextView.setTextColor(Color.WHITE);
} else if
(colorTheme.equals(mContext.getString(R.string.settings_color_light_gree
n_value))) {

holder.titleTextView.setBackgroundResource(R.color.light_green);
holder.titleTextView.setTextColor(Color.WHITE);
} else if
(colorTheme.equals(mContext.getString(R.string.settings_color_green_valu
e))) {

holder.titleTextView.setBackgroundResource(R.color.color_section);
holder.titleTextView.setTextColor(Color.WHITE);
}
    }

/**
     * Set the text size to the text size the user choose.
     */
private void setTextSize(ViewHolder holder) {
// Get the text size string from SharedPreferences and check for the
value associated with the key
String textSize = sharedPrefs.getString(
mContext.getString(R.string.settings_text_size_key),
mContext.getString(R.string.settings_text_size_default));

// Change text size of TextView by using the user's stored preferences
if(textSize.equals(mContext.getString(R.string.settings_text_size_medium
_value))) {
            holder.titleTextView.setTextSize(TypedValue.COMPLEX_UNIT_PX,
mContext.getResources().getDimension(R.dimen.sp22));
holder.sectionTextView.setTextSize(TypedValue.COMPLEX_UNIT_PX,
mContext.getResources().getDimension(R.dimen.sp14));
holder.trailTextView.setTextSize(TypedValue.COMPLEX_UNIT_PX,
mContext.getResources().getDimension(R.dimen.sp16));
holder.authorTextView.setTextSize(TypedValue.COMPLEX_UNIT_PX,
mContext.getResources().getDimension(R.dimen.sp14));
holder.dateTextView.setTextSize(TypedValue.COMPLEX_UNIT_PX,
mContext.getResources().getDimension(R.dimen.sp14));
} else
if(textSize.equals(mContext.getString(R.string.settings_text_size_small_
value))) {
            holder.titleTextView.setTextSize(TypedValue.COMPLEX_UNIT_PX,
mContext.getResources().getDimension(R.dimen.sp20));
holder.sectionTextView.setTextSize(TypedValue.COMPLEX_UNIT_PX,
mContext.getResources().getDimension(R.dimen.sp12));
holder.trailTextView.setTextSize(TypedValue.COMPLEX_UNIT_PX,
mContext.getResources().getDimension(R.dimen.sp14));
holder.authorTextView.setTextSize(TypedValue.COMPLEX_UNIT_PX,
mContext.getResources().getDimension(R.dimen.sp12));
holder.dateTextView.setTextSize(TypedValue.COMPLEX_UNIT_PX,
mContext.getResources().getDimension(R.dimen.sp12));
} else
if(textSize.equals(mContext.getString(R.string.settings_text_size_large_
value))) {
            holder.titleTextView.setTextSize(TypedValue.COMPLEX_UNIT_PX,
mContext.getResources().getDimension(R.dimen.sp24));
holder.sectionTextView.setTextSize(TypedValue.COMPLEX_UNIT_PX,
mContext.getResources().getDimension(R.dimen.sp16));
holder.trailTextView.setTextSize(TypedValue.COMPLEX_UNIT_PX,
mContext.getResources().getDimension(R.dimen.sp18));
holder.authorTextView.setTextSize(TypedValue.COMPLEX_UNIT_PX,
mContext.getResources().getDimension(R.dimen.sp16));
holder.dateTextView.setTextSize(TypedValue.COMPLEX_UNIT_PX,
mContext.getResources().getDimension(R.dimen.sp16));
}
    }

/**
     * Share the article with friends in social network
     * @param news {@link News} object
     */
private void shareData(News news) {
        Intent sharingIntent = new Intent(Intent.ACTION_SEND);
sharingIntent.setType("text/plain");
sharingIntent.putExtra(android.content.Intent.EXTRA_TEXT,
news.getTitle() + " : " + news.getUrl());
mContext.startActivity(Intent.createChooser(sharingIntent,
mContext.getString(R.string.share_article)));
}

/**
     *  Clear all data (a list of {@link News} objects)
     */
public void clearAll() {
mNewsList.clear();
notifyDataSetChanged();
}

/**
     * Add  a list of {@link News}
     * @param newsList is the list of news, which is the data source of
the adapter
     */
public void addAll(List<News> newsList) {
mNewsList.clear();
mNewsList.addAll(newsList);
notifyDataSetChanged();
}

/**
     * Convert date and time in UTC (webPublicationDate) into a more
readable representation
     * in Local time
     *
     * @param dateStringUTC is the web publication date of the article
(i.e. 2014-02-04T08:00:00Z)
```

```
     * @return the formatted date string in Local time(i.e "Jan 1, 2000
2:15 AM")
     * from a date and time in UTC
     */
private String formatDate(String dateStringUTC) {
// Parse the dateString into a Date object
SimpleDateFormat simpleDateFormat =
new SimpleDateFormat("yyyy-MM-dd'T'kk:mm:ss'Z'");
Date dateObject = null;
        try {
            dateObject = simpleDateFormat.parse(dateStringUTC);
} catch (ParseException e) {
            e.printStackTrace();
}
// Initialize a SimpleDateFormat instance and configure it to provide a
more readable
        // representation according to the given format, but still in
UTC
SimpleDateFormat df = new SimpleDateFormat("MMM d, yyyy  h:mm a",
Locale.ENGLISH);
String formattedDateUTC = df.format(dateObject);
// Convert UTC into Local time
df.setTimeZone(TimeZone.getTimeZone("UTC"));
Date date = null;
        try {
            date = df.parse(formattedDateUTC);
df.setTimeZone(TimeZone.getDefault());
} catch (ParseException e) {
            e.printStackTrace();
}
return df.format(date);
}

/**
     * Get the formatted web publication date string in milliseconds
     * @param formattedDate the formatted web publication date string
     * @return the formatted web publication date in milliseconds
     */
private static long getDateInMillis(String formattedDate) {
        SimpleDateFormat simpleDateFormat =
new SimpleDateFormat("MMM d, yyyy  h:mm a");
        long dateInMillis;
Date dateObject;
        try {
            dateObject = simpleDateFormat.parse(formattedDate);
dateInMillis = dateObject.getTime();
            return dateInMillis;
} catch (ParseException e) {
            Log.e("Problem parsing date", e.getMessage());
e.printStackTrace();
}
return 0;
}

/**
     * Get the time difference between the current date and web
```

```
publication date
     * @param formattedDate the formatted web publication date string
     * @return time difference (i.e "9 hours ago")
     */
private CharSequence getTimeDifference(String formattedDate) {
long currentTime = System.currentTimeMillis();
        long publicationTime = getDateInMillis(formattedDate);
        return DateUtils.getRelativeTimeSpanString(publicationTime,
currentTime,
DateUtils.SECOND_IN_MILLIS);
}
}
```

Fragment:

BaseArticlesFragment.java

```
package com.example.android.newsfeed.fragment;

import android.content.Context;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.net.Uri;
import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.v4.app.Fragment;
import android.support.v4.app.LoaderManager;
import android.support.v4.content.Loader;
import android.support.v4.widget.SwipeRefreshLayout;
import android.support.v7.widget.LinearLayoutManager;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
import android.widget.Toast;

import com.example.android.newsfeed.EmptyRecyclerView;
import com.example.android.newsfeed.News;
import com.example.android.newsfeed.NewsLoader;
import com.example.android.newsfeed.NewsPreferences;
import com.example.android.newsfeed.R;
import com.example.android.newsfeed.adapter.NewsAdapter;
import com.example.android.newsfeed.utils.Constants;

import java.util.ArrayList;
import java.util.List;

/**
 * The BaseArticlesFragment is a {@link Fragment} subclass that
implements the LoaderManager.LoaderCallbacks
 * interface in order for Fragment to be a client that interacts with
the LoaderManager. It is
 * base class that is responsible for displaying a set of articles,
regardless of type.
```

```
 */
public class BaseArticlesFragment extends Fragment
implements LoaderManager.LoaderCallbacks<List<News>>{

private static final String LOG_TAG =
BaseArticlesFragment.class.getName();

/** Constant value for the news loader ID. */
private static final int NEWS_LOADER_ID = 1;

/** Adapter for the list of news */
private NewsAdapter mAdapter;

/** TextView that is displayed when the recycler view is empty */
private TextView mEmptyStateTextView;

/** Loading indicator that is displayed before the first load is
completed */
private View mLoadingIndicator;

/** The {@link android.support.v4.widget.SwipeRefreshLayout} that
detects swipe gestures and
     * triggers callbacks in the app.
     */
private SwipeRefreshLayout mSwipeRefreshLayout;

@Override
public View onCreateView(@NonNull LayoutInflater inflater, ViewGroup
container, Bundle savedInstanceState) {
        View rootView = inflater.inflate(R.layout.fragment_home,
container, false);

// Find a reference to the {@link RecyclerView} in the layout
        // Replaced RecyclerView with EmptyRecyclerView
EmptyRecyclerView mRecyclerView =
rootView.findViewById(R.id.recycler_view);
LinearLayoutManager layoutManager = new
LinearLayoutManager(getActivity());
mRecyclerView.setHasFixedSize(true);

// Set the layoutManager on the {@link RecyclerView}
mRecyclerView.setLayoutManager(layoutManager);

// Find the SwipeRefreshLayout
mSwipeRefreshLayout = rootView.findViewById(R.id.swipe_refresh);
// Set the color scheme of the SwipeRefreshLayout
mSwipeRefreshLayout.setColorSchemeColors(getResources().getColor(R.color
.swipe_color_1),
getResources().getColor(R.color.swipe_color_2),
getResources().getColor(R.color.swipe_color_3),
getResources().getColor(R.color.swipe_color_4));

// Set up OnRefreshListener that is invoked when the user performs a
swipe-to-refresh gesture.
mSwipeRefreshLayout.setOnRefreshListener(new
SwipeRefreshLayout.OnRefreshListener() {
```

```
@Override
public void onRefresh() {
                Log.i(LOG_TAG, "onRefresh called from
SwipeRefreshLayout");
// restart the loader
initiateRefresh();
Toast.makeText(getActivity(), getString(R.string.updated_just_now),
Toast.LENGTH_SHORT).show();
}
        });

// Find the loading indicator from the layout
mLoadingIndicator = rootView.findViewById(R.id.loading_indicator);

// Find the empty view from the layout and set it on the new recycler
view
mEmptyStateTextView = rootView.findViewById(R.id.empty_view);
mRecyclerView.setEmptyView(mEmptyStateTextView);

// Create a new adapter that takes an empty list of news as input
mAdapter = new NewsAdapter(getActivity(), new ArrayList<News>());

// Set the adapter on the {@link recyclerView}
mRecyclerView.setAdapter(mAdapter);

// Check for network connectivity and initialize the loader
initializeLoader(isConnected());

        return rootView;
}

@NonNull
    @Override
public Loader<List<News>>onCreateLoader(int i, Bundle bundle) {

        Uri.Builder uriBuilder =
NewsPreferences.getPreferredUri(getContext());

Log.e(LOG_TAG,uriBuilder.toString());

// Create a new loader for the given URL
return new NewsLoader(getActivity(), uriBuilder.toString());
}

@Override
public void onLoadFinished(@NonNull Loader<List<News>> loader,
List<News> newsData) {
// Hide loading indicator because the data has been loaded
mLoadingIndicator.setVisibility(View.GONE);

// Set empty state text to display "No news found."
mEmptyStateTextView.setText(R.string.no_news);

// Clear the adapter of previous news data
mAdapter.clearAll();
```

```
// If there is a valid list of {@link News}, then add them to the
adapter's
        // data set. This will trigger the recyclerView to update.
if(newsData != null && !newsData.isEmpty()) {
mAdapter.addAll(newsData);
}

// Hide the swipe icon animation when the loader is done refreshing the
data
mSwipeRefreshLayout.setRefreshing(false);
}

@Override
public void onLoaderReset(@NonNull Loader<List<News>> loader) {
// Loader reset, so we can clear out our existing data.
mAdapter.clearAll();
}

/**
     * When the user returns to the previous screen by pressing the up
button in the SettingsActivity,
     * restart the Loader to reflect the current value of the
preference.
     */
@Override
public void onResume() {
super.onResume();
restartLoader(isConnected());
}

/**
     *  Check for network connectivity.
     */
private boolean isConnected() {
// Get a reference to the ConnectivityManager to check state of network
connectivity
ConnectivityManager connectivityManager = (ConnectivityManager)

getActivity().getSystemService(Context.CONNECTIVITY_SERVICE);

// Get details on the currently active default data network
NetworkInfo networkInfo = connectivityManager.getActiveNetworkInfo();

        return (networkInfo != null && networkInfo.isConnected());
}

/**
     * If there is internet connectivity, initialize the loader as
     * usual. Otherwise, hide loading indicator and set empty state
TextView to display
     * "No internet connection."
     *
     * @param isConnected internet connection is available or not
     */
private void initializeLoader(boolean isConnected) {
if (isConnected) {
```

```
// Get a reference to the LoaderManager, in order to interact with
loaders.
LoaderManager loaderManager = getLoaderManager();
// Initialize the loader with the NEWS_LOADER_ID
loaderManager.initLoader(NEWS_LOADER_ID, null, this);
} else {
// Otherwise, display error
            // First, hide loading indicator so error message will be
visible
mLoadingIndicator.setVisibility(View.GONE);
// Update empty state with no connection error message and image
mEmptyStateTextView.setText(R.string.no_internet_connection);
mEmptyStateTextView.setCompoundDrawablesWithIntrinsicBounds(Constants.DE
FAULT_NUMBER,
R.drawable.ic_network_check,Constants.DEFAULT_NUMBER,Constants.DEFAULT_N
UMBER);
}
    }

/**
     * Restart the loader if there is internet connectivity.
     * @param isConnected internet connection is available or not
     */
private void restartLoader(boolean isConnected) {
if (isConnected) {
// Get a reference to the LoaderManager, in order to interact with
loaders.
LoaderManager loaderManager = getLoaderManager();
// Restart the loader with the NEWS_LOADER_ID
loaderManager.restartLoader(NEWS_LOADER_ID, null, this);
} else {
// Otherwise, display error
            // First, hide loading indicator so error message will be
visible
mLoadingIndicator.setVisibility(View.GONE);
// Update empty state with no connection error message and image
mEmptyStateTextView.setText(R.string.no_internet_connection);
mEmptyStateTextView.setCompoundDrawablesWithIntrinsicBounds(Constants.DE
FAULT_NUMBER,
R.drawable.ic_network_check,Constants.DEFAULT_NUMBER,Constants.DEFAULT_N
UMBER);

// Hide SwipeRefreshLayout
mSwipeRefreshLayout.setVisibility(View.GONE);
}
    }

/**
     * When the user performs a swipe-to-refresh gesture, restart the
loader.
     */
private void initiateRefresh() {
        restartLoader(isConnected());
}
}
```

BusinessFragment.java

```java
package com.example.android.newsfeed.fragment;

import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.v4.content.Loader;
import android.util.Log;

import com.example.android.newsfeed.News;
import com.example.android.newsfeed.NewsLoader;
import com.example.android.newsfeed.NewsPreferences;
import com.example.android.newsfeed.R;

import java.util.List;

/**
 * The BusinessFragment is a {@link BaseArticlesFragment} subclass that
 * reuses methods of the parent class by passing the specific type of
article to be fetched.
 */
public class BusinessFragment extends BaseArticlesFragment {

private static final String LOG_TAG = BusinessFragment.class.getName();

@NonNull
    @Override
public Loader<List<News>>onCreateLoader(int i, Bundle bundle) {
        String businessUrl =
NewsPreferences.getPreferredUrl(getContext(),
getString(R.string.business));
Log.e(LOG_TAG, businessUrl);

// Create a new loader for the given URL
return new NewsLoader(getActivity(),businessUrl);
}
}
```

CultureFragment.java

```java
package com.example.android.newsfeed.fragment;

import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.v4.content.Loader;
import android.util.Log;

import com.example.android.newsfeed.News;
import com.example.android.newsfeed.NewsLoader;
import com.example.android.newsfeed.NewsPreferences;
import com.example.android.newsfeed.R;
import java.util.List;

/**
 * The CultureFragment is a {@link BaseArticlesFragment} subclass that
 * reuses methods of the parent class by passing the specific type of
article to be fetched.
 */
public class CultureFragment extends BaseArticlesFragment {

private static final String LOG_TAG = CultureFragment.class.getName();

@NonNull
    @Override
public Loader<List<News>>onCreateLoader(int i, Bundle bundle) {
        String cultureUrl =
NewsPreferences.getPreferredUrl(getContext(),
getString(R.string.culture));
Log.e(LOG_TAG, cultureUrl);

// Create a new loader for the given URL
return new NewsLoader(getActivity(), cultureUrl);
}
}
```

EnvironmentFragment.java

```
package com.example.android.newsfeed.fragment;

import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.v4.content.Loader;
import android.util.Log;

import com.example.android.newsfeed.News;
import com.example.android.newsfeed.NewsLoader;
import com.example.android.newsfeed.NewsPreferences;
import com.example.android.newsfeed.R;

import java.util.List;

/**
 * The EnvironmentFragment is a {@link BaseArticlesFragment} subclass
that
 * reuses methods of the parent class by passing the specific type of
article to be fetched.
 */
public class EnvironmentFragment extends BaseArticlesFragment {

private static final String LOG_TAG =
EnvironmentFragment.class.getName();

@NonNull
    @Override
public Loader<List<News>>onCreateLoader(int i, Bundle bundle) {
        String environmentUrl =
NewsPreferences.getPreferredUrl(getContext(),
getString(R.string.environment));
Log.e(LOG_TAG, environmentUrl);

// Create a new loader for the given URL
return new NewsLoader(getActivity(), environmentUrl);
}
}
```

FashionFragment.java

```
package com.example.android.newsfeed.fragment;

import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.v4.content.Loader;
import android.util.Log;

import com.example.android.newsfeed.News;
import com.example.android.newsfeed.NewsLoader;
import com.example.android.newsfeed.NewsPreferences;
import com.example.android.newsfeed.R;

import java.util.List;

/**
 * The FashionFragment is a {@link BaseArticlesFragment} subclass that
 * reuses methods of the parent class by passing the specific type of
article to be fetched.
 */
public class FashionFragment extends BaseArticlesFragment {

private static final String LOG_TAG = FashionFragment.class.getName();

@NonNull
    @Override
public Loader<List<News>>onCreateLoader(int i, Bundle bundle) {
        String fashionUrl =
NewsPreferences.getPreferredUrl(getContext(),
getString(R.string.fashion));
Log.e(LOG_TAG, fashionUrl);

// Create a new loader for the given URL
return new NewsLoader(getActivity(), fashionUrl);
}
}
```

HomeFragment.java

```
packagecom.example.android.newsfeed.fragment;

/**
 * The HomeFragment is a {@link BaseArticlesFragment} subclass that
 * reuses methods of the parent class
 */
public class HomeFragment extends BaseArticlesFragment {

public static final String LOG_TAG = HomeFragment.class.getName();

}
```

ScienceFragment.java

```
package com.example.android.newsfeed.fragment;

import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.v4.content.Loader;
import android.util.Log;

import com.example.android.newsfeed.News;
import com.example.android.newsfeed.NewsLoader;
import com.example.android.newsfeed.NewsPreferences;
import com.example.android.newsfeed.R;

import java.util.List;

/**
 * The ScienceFragment is a {@link BaseArticlesFragment} subclass that
 * reuses methods of the parent class by passing the specific type of
article to be fetched.
 */
public class ScienceFragment extends BaseArticlesFragment {

private static final String LOG_TAG = ScienceFragment.class.getName();

@NonNull
    @Override
public Loader<List<News>>onCreateLoader(int i, Bundle bundle) {
        String scienceUrl =
NewsPreferences.getPreferredUrl(getContext(),
getString(R.string.science));
Log.e(LOG_TAG, scienceUrl);

// Create a new loader for the given URL
return new NewsLoader(getActivity(), scienceUrl);
}
}
```

SocietyFragment.java

```
package com.example.android.newsfeed.fragment;

import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.v4.content.Loader;
import android.util.Log;

import com.example.android.newsfeed.News;
import com.example.android.newsfeed.NewsLoader;
import com.example.android.newsfeed.NewsPreferences;
import com.example.android.newsfeed.R;

import java.util.List;

/**
 * The SocietyFragment is a {@link BaseArticlesFragment} subclass that
 * reuses methods of the parent class by passing the specific type of
article to be fetched.
 */
public class SocietyFragment extends BaseArticlesFragment {

private static final String LOG_TAG = SocietyFragment.class.getName();

@NonNull
    @Override
public Loader<List<News>>onCreateLoader(int i, Bundle bundle) {
        String societyUrl =
NewsPreferences.getPreferredUrl(getContext(),
getString(R.string.society));
Log.e(LOG_TAG, societyUrl);

// Create a new loader for the given URL
return new NewsLoader(getActivity(), societyUrl);
}
}
```

SportFragment.java

```
package com.example.android.newsfeed.fragment;

import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.v4.content.Loader;
import android.util.Log;

import com.example.android.newsfeed.News;
import com.example.android.newsfeed.NewsLoader;
import com.example.android.newsfeed.NewsPreferences;
```

```java
import com.example.android.newsfeed.R;

import java.util.List;

/**
 * The SportFragment is a {@link BaseArticlesFragment} subclass that
 * reuses methods of the parent class by passing the specific type of
article to be fetched.
 */
public class SportFragment extends BaseArticlesFragment {

private static final String LOG_TAG = SportFragment.class.getName();

@NonNull
    @Override
public Loader<List<News>>onCreateLoader(int i, Bundle bundle) {
        String sportUrl = NewsPreferences.getPreferredUrl(getContext(),
getString(R.string.sport));
Log.e(LOG_TAG, sportUrl);

// Create a new loader for the given URL
return new NewsLoader(getActivity(), sportUrl);
}
}
```

WorldFragment.java

```java
package com.example.android.newsfeed.fragment;

import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.v4.content.Loader;
import android.util.Log;

import com.example.android.newsfeed.News;
import com.example.android.newsfeed.NewsLoader;
import com.example.android.newsfeed.NewsPreferences;
import com.example.android.newsfeed.R;

import java.util.List;

/**
 * The WorldFragment is a {@link BaseArticlesFragment} subclass that
 * reuses methods of the parent class by passing the specific type of
article to be fetched.
 */
public class WorldFragment extends BaseArticlesFragment {

private static final String LOG_TAG = WorldFragment.class.getName();

@NonNull
    @Override
public Loader<List<News>>onCreateLoader(int i, Bundle bundle) {
        String worldUrl = NewsPreferences.getPreferredUrl(getContext(),
getString(R.string.world));
Log.e(LOG_TAG, worldUrl);

// Create a new loader for the given URL
return new NewsLoader(getActivity(), worldUrl);
}
}
```

Utils:

Constants.java

```
packagecom.example.android.newsfeed.utils;

/**
 * Store Constants for the NewsFeed app.
 */

public class Constants {

/**
     * Create a private constructor because no one should ever create a
{@link Constants} object.
     */
private Constants() {
    }

/**  Extract the key associated with the JSONObject */
static final String JSON_KEY_RESPONSE = "response";
    static final String JSON_KEY_RESULTS = "results";
    static final String JSON_KEY_WEB_TITLE = "webTitle";
    static final String JSON_KEY_SECTION_NAME = "sectionName";
    static final String JSON_KEY_WEB_PUBLICATION_DATE =
"webPublicationDate";
    static final String JSON_KEY_WEB_URL = "webUrl";
    static final String JSON_KEY_TAGS = "tags";
    static final String JSON_KEY_FIELDS = "fields";
    static final String JSON_KEY_THUMBNAIL = "thumbnail";
    static final String JSON_KEY_TRAIL_TEXT = "trailText";

/** Read timeout for setting up the HTTP request */
static final int READ_TIMEOUT = 10000; /* milliseconds */

/** Connect timeout for setting up the HTTP request */
static final int CONNECT_TIMEOUT = 15000; /* milliseconds */

/** HTTP response code when the request is successful */
static final int SUCCESS_RESPONSE_CODE = 200;

/** Request method type "GET" for reading information from the server */
static final String REQUEST_METHOD_GET = "GET";

/** URL for news data from the guardian data set */
public static final String NEWS_REQUEST_URL =
"https://content.guardianapis.com/search";
```

```java
/** Parameters */
public static final String QUERY_PARAM = "q";
    public static final String ORDER_BY_PARAM = "order-by";
    public static final String PAGE_SIZE_PARAM = "page-size";
    public static final String ORDER_DATE_PARAM = "order-date";
    public static final String FROM_DATE_PARAM = "from-date";
    public static final String SHOW_FIELDS_PARAM = "show-fields";
    public static final String FORMAT_PARAM = "format";
    public static final String SHOW_TAGS_PARAM = "show-tags";
    public static final String API_KEY_PARAM = "api-key";
    public static final String SECTION_PARAM = "section";

/** The show fields we want our API to return */
public static final String SHOW_FIELDS = "thumbnail,trailText";

/** The format we want our API to return */
public static final String FORMAT = "json";

/** The show tags we want our API to return */
public static final String SHOW_TAGS = "contributor";

/** API Key */
public static final String API_KEY = "test"; // Use your API Key when
API rate limit exceeded

/** Default number to set the image on the top of the textView */
public static final int DEFAULT_NUMBER = 0;

/** Constants value for each fragment */
public static final int HOME = 0;
    public static final int WORLD = 1;
    public static final int SCIENCE = 2;
    public static final int SPORT = 3;
    public static final int ENVIRONMENT = 4;
    public static final int SOCIETY = 5;
    public static final int FASHION = 6;
    public static final int BUSINESS = 7;
    public static final int CULTURE = 8;

}
```

QueryUtils.java

```java
package com.example.android.newsfeed.utils;

import android.text.TextUtils;
import android.util.Log;

import com.example.android.newsfeed.News;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;
import java.nio.charset.Charset;
import java.util.ArrayList;
import java.util.List;

/**
 * Helper methods related to requesting and receiving news data from
Guardian.
 */
public class QueryUtils {

/** Tag for the log messages */
private static final String LOG_TAG = QueryUtils.class.getSimpleName();

/**
     * Create a private constructor because no one should ever create a
{@link QueryUtils} object.
     */
private QueryUtils() {
    }

/**
     * Query the Guardian data set and return a list of {@link News}
objects.
     */
public static List<News>fetchNewsData(String requestUrl) {
// Create URL object
URL url = createUrl(requestUrl);

// Perform HTTP request to the URL and receive a JSON response back
String jsonResponse = null;
        try {
            jsonResponse = makeHttpRequest(url);
} catch (IOException e) {
            Log.e(LOG_TAG, "Problem making the HTTP request.", e);
}

// Extract relevant fields from the JSON response and create a list of
{@link News}
List<News> newsList = extractFeatureFromJSON(jsonResponse);

// Return the list of {@link News}
return newsList;
}

/**
     * Returns new URL object from the given string URL.
     */
private static URL createUrl(String stringUrl) {
        URL url = null;
        try {
            url = new URL(stringUrl);
} catch (MalformedURLException e) {
            Log.e(LOG_TAG, "Problem building the URL.", e);
}
return url;
}

/**
     * Make an HTTP request to the given URL and return a String as the
response.
     */
private static String makeHttpRequest(URL url) throws IOException {
        String jsonResponse = "";

// If the URL is null, then return early.
if (url == null) {
return jsonResponse;
}

        HttpURLConnection urlConnection = null;
InputStream inputStream = null;
        try {
            urlConnection = (HttpURLConnection) url.openConnection();
urlConnection.setReadTimeout(Constants.READ_TIMEOUT /* milliseconds */);
urlConnection.setConnectTimeout(Constants.CONNECT_TIMEOUT /*
milliseconds */);
urlConnection.setRequestMethod(Constants.REQUEST_METHOD_GET);
urlConnection.connect();

// If the request was successful (response code 200),
            // then read the input stream and parse the response.
if (urlConnection.getResponseCode() == Constants.SUCCESS_RESPONSE_CODE)
{
                inputStream = urlConnection.getInputStream();
jsonResponse = readFromStream(inputStream);
} else {
                Log.e(LOG_TAG, "Error response code: " +
urlConnection.getResponseCode());
}
        } catch (IOException e) {
            Log.e(LOG_TAG, "Problem retrieving the news JSON results.",
e);
} finally {
if (urlConnection != null) {
                urlConnection.disconnect();
}
if (inputStream != null) {
// Closing the input stream could throw an IOException, which is why
                // the makeHttpRequest(URL url) method signature
specifies that an IOException
                // could be thrown.
inputStream.close();
}
        }
return jsonResponse;
}
```

```
/**
     * Convert the {@link InputStream} into a String which contains the
     * whole JSON response from the server.
     */
private static String readFromStream(InputStream inputStream) throws
IOException {
        StringBuilder output = new StringBuilder();
        if (inputStream != null) {
            InputStreamReader inputStreamReader = new
InputStreamReader(inputStream, Charset.forName("UTF-8"));
BufferedReader reader = new BufferedReader(inputStreamReader);
String line = reader.readLine();
            while (line != null) {
                output.append(line);
line = reader.readLine();
}
        }
return output.toString();
}

/**
     * Return a list of {@link News} objects that has been built up from
     * parsing the given JSON response.
     */
private static List<News>extractFeatureFromJSON(String newsJSON) {
// If the JSON string is empty or null, then return early.
if(TextUtils.isEmpty(newsJSON)) {
return null;
}
// Create an empty ArrayList that we can start adding news to
List<News> newsList = new ArrayList<>();

// Try to parse the JSON response string. If there's a problem with the
way the JSON
        // is formatted, a JSONException exception object will be
thrown.
try {
// Create a JSONObject from the JSON response string
JSONObject baseJsonResponse = new JSONObject(newsJSON);

// Extract the JSONObject associated with the key called "response"
JSONObject responseJsonObject =
baseJsonResponse.getJSONObject(Constants.JSON_KEY_RESPONSE);

// Extract the JSONArray associated with the key called "results"
JSONArray resultsArray =
responseJsonObject.getJSONArray(Constants.JSON_KEY_RESULTS);

// For each element in the resultsArray, create a {@link News} object
for (int i = 0; i < resultsArray.length(); i++) {

// Get a single news at position i within the list of news
JSONObject currentNews = resultsArray.getJSONObject(i);
// For a given news, extract the value for the key called "webTitle"
String webTitle = currentNews.getString(Constants.JSON_KEY_WEB_TITLE);
// For a given news, extract the value for the key called "sectionName"
String sectionName =
currentNews.getString(Constants.JSON_KEY_SECTION_NAME);
// For a given news, extract the value for the key called
"webPublicationDate"
String webPublicationDate =
currentNews.getString(Constants.JSON_KEY_WEB_PUBLICATION_DATE);
// For a given news, extract the value for the key called "webUrl"
String webUrl = currentNews.getString(Constants.JSON_KEY_WEB_URL);

// For a given news, if it contains the key called "tags", extract
JSONArray
                // associated with the key "tags"
String author = null;
                if (currentNews.has(Constants.JSON_KEY_TAGS)) {
// Extract the JSONArray associated with the key called "tags"
JSONArray tagsArray = currentNews.getJSONArray(Constants.JSON_KEY_TAGS);
                    if (tagsArray.length() != 0) {
// Extract the first JSONObject in the tagsArray
JSONObject firstTagsItem = tagsArray.getJSONObject(0);
// Extract the value for the key called "webTitle"
author = firstTagsItem.getString(Constants.JSON_KEY_WEB_TITLE);
}
                }

// For a given news, if it contains the key called "fields", extract
JSONObject
                // associated with the key "fields"
String thumbnail = null;
String trailText = null;
                if (currentNews.has(Constants.JSON_KEY_FIELDS)) {
// Extract the JSONObject associated with the key called "fields"
JSONObject fieldsObject =
currentNews.getJSONObject(Constants.JSON_KEY_FIELDS);
// If there is the key called "thumbnail", extract the value for the key
called "thumbnail"
if (fieldsObject.has(Constants.JSON_KEY_THUMBNAIL)) {
                        thumbnail =
fieldsObject.getString(Constants.JSON_KEY_THUMBNAIL);
}
// If there is the key called "trailText", extract the value for the key
called "trailText"
if (fieldsObject.has(Constants.JSON_KEY_TRAIL_TEXT)) {
                        trailText =
fieldsObject.getString(Constants.JSON_KEY_TRAIL_TEXT);
}
                }

// Create a new {@link News} object with the title and url from the JSON
response.
News news = new News(webTitle, sectionName, author, webPublicationDate,
webUrl, thumbnail, trailText);

// Add the new {@link News} to list of newsList.
newsList.add(news);
}
```

```
        } catch (JSONException e) {
// If an error is thrown when executing any of the above statements in
the "try" block,
            // catch the exception here, so the app doesn't crash. Print
a log message
            // with the message from the exception.
Log.e(LOG_TAG, "Problem parsing the news JSON results", e);
}

// Return the list of news
return newsList;
}
}
```

EmptyRecyclerView.java

```
packagecom.example.android.newsfeed;

import android.content.Context;
import android.support.v7.widget.RecyclerView;
import android.util.AttributeSet;
import android.view.View;

/**
 * EmptyRecyclerView is RecyclerView subclass that provides empty view
support for RecyclerView
 * to show or hide an empty view based on whether the adapter provided
to the RecyclerView has
 * data or not.
 */

public class EmptyRecyclerView extends RecyclerView {

private View mEmptyView;

/**
     * The AdapterDataObserver calls checkIfEmpty() method every time,
and it observes
     * an event that changes the content of the adapter
     */
final private AdapterDataObserver observer = new AdapterDataObserver() {
@Override
public void onChanged() {
            checkIfEmpty();
}

@Override
public void onItemRangeInserted(int positionStart, int itemCount) {
            checkIfEmpty();
}

@Override
public void onItemRangeRemoved(int positionStart, int itemCount) {
            checkIfEmpty();
}
```

```
    };

    public EmptyRecyclerView(Context context) {
super(context);
}

public EmptyRecyclerView(Context context, AttributeSet attrs) {
super(context, attrs);
}

public EmptyRecyclerView(Context context, AttributeSet attrs,
                             int defStyle) {
super(context, attrs, defStyle);
}

/**
     * Checks if both mEmptyView and adapter are not null.
     * Hide or show mEmptyView depending on the size of the data(item
count) in the adapter.
     */
private void checkIfEmpty() {
// If the item count provided by the adapter is equal to zero, make the
empty View visible
        // and hide the EmptyRecyclerView.
        // Otherwise, hide the empty View and make the EmptyRecyclerView
visible.
if (mEmptyView != null && getAdapter() != null) {
final boolean emptyViewVisible =
                    getAdapter().getItemCount() == 0;
mEmptyView.setVisibility(emptyViewVisible ? VISIBLE :GONE);
setVisibility(emptyViewVisible ? GONE : VISIBLE);
}
    }

@Override
public void setAdapter(Adapter adapter) {
final Adapter oldAdapter = getAdapter();
        if (oldAdapter != null) {
            oldAdapter.unregisterAdapterDataObserver(observer);
}
super.setAdapter(adapter);
        if (adapter != null) {
            adapter.registerAdapterDataObserver(observer);
}

        checkIfEmpty();
}

/**
     * Set an empty view on the EmptyRecyclerView
     * @param emptyView refers to the empty state of the view
     */
public void setEmptyView(View emptyView) {
mEmptyView = emptyView;
checkIfEmpty();
```

```
}
}
```

MainActivity.java

```
package com.example.android.newsfeed;

import android.content.Intent;
import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.design.widget.NavigationView;
import android.support.design.widget.TabLayout;
import android.support.v4.view.GravityCompat;
import android.support.v4.view.ViewPager;
import android.support.v4.widget.DrawerLayout;
import android.support.v7.app.ActionBarDrawerToggle;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.Menu;
import android.view.MenuItem;

import
com.example.android.newsfeed.adapter.CategoryFragmentPagerAdapter;
import com.example.android.newsfeed.utils.Constants;

public class MainActivity extends AppCompatActivity
implements NavigationView.OnNavigationItemSelectedListener {

private ViewPager viewPager;

@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
Toolbar toolbar = findViewById(R.id.toolbar);
setSupportActionBar(toolbar);

DrawerLayout drawer = findViewById(R.id.drawer_layout);
ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
this, drawer, toolbar, R.string.navigation_drawer_open,
R.string.navigation_drawer_close);
drawer.addDrawerListener(toggle);
toggle.syncState();

// Find the view pager that will allow the user to swipe between
fragments
viewPager = findViewById(R.id.viewpager);

// Give the TabLayout the ViewPager
TabLayout tabLayout = findViewById(R.id.sliding_tabs);
tabLayout.setupWithViewPager(viewPager);
// Set gravity for tab bar
tabLayout.setTabGravity(TabLayout.GRAVITY_FILL);
```

```
NavigationView navigationView = findViewById(R.id.nav_view);
        assert navigationView != null;
navigationView.setNavigationItemSelectedListener(this);

// Set the default fragment when starting the app
onNavigationItemSelected(navigationView.getMenu().getItem(0).setChecked(
true));

// Set category fragment pager adapter
CategoryFragmentPagerAdapter pagerAdapter =
new CategoryFragmentPagerAdapter(this, getSupportFragmentManager());
// Set the pager adapter onto the view pager
viewPager.setAdapter(pagerAdapter);
}

@Override
public void onBackPressed() {
        DrawerLayout drawer = findViewById(R.id.drawer_layout);
        if (drawer.isDrawerOpen(GravityCompat.START)) {
            drawer.closeDrawer(GravityCompat.START);
} else {
super.onBackPressed();
}
    }

@SuppressWarnings("StatementWithEmptyBody")
@Override
public boolean onNavigationItemSelected(@NonNull MenuItem item) {
// Handle navigation view item clicks here.
int id = item.getItemId();

// Switch Fragments in a ViewPager on clicking items in Navigation
Drawer
if (id == R.id.nav_home) {
viewPager.setCurrentItem(Constants.HOME);
} else if (id == R.id.nav_world) {
viewPager.setCurrentItem(Constants.WORLD);
} else if (id == R.id.nav_science) {
viewPager.setCurrentItem(Constants.SCIENCE);
} else if (id == R.id.nav_sport) {
viewPager.setCurrentItem(Constants.SPORT);
} else if (id == R.id.nav_environment) {
viewPager.setCurrentItem(Constants.ENVIRONMENT);
} else if (id == R.id.nav_society) {
viewPager.setCurrentItem(Constants.SOCIETY);
} else if (id == R.id.nav_fashion) {
viewPager.setCurrentItem(Constants.FASHION);
} else if (id == R.id.nav_business) {
viewPager.setCurrentItem(Constants.BUSINESS);
} else if (id == R.id.nav_culture) {
viewPager.setCurrentItem(Constants.CULTURE);
}

        DrawerLayout drawer = findViewById(R.id.drawer_layout);
drawer.closeDrawer(GravityCompat.START);
```

```
        return true;
}

@Override
// Initialize the contents of the Activity's options menu
public boolean onCreateOptionsMenu(Menu menu) {
// Inflate the Options Menu we specified in XML
getMenuInflater().inflate(R.menu.main, menu);
        return true;
}

@Override
// This method is called whenever an item in the options menu is
selected.
public boolean onOptionsItemSelected(MenuItem item) {
int id = item.getItemId();
        if (id == R.id.action_settings) {
            Intent settingsIntent = new Intent(this,
SettingsActivity.class);
startActivity(settingsIntent);
            return true;
}
return super.onOptionsItemSelected(item);
}

}
```

News.java

```
packagecom.example.android.newsfeed;

/**
 * An {@link News} object contains information related to a single news.
 */

public class News {

/** Title of the article */
private String mTitle;

/** Section name of the article*/
private String mSection;

/** Author name in the article */
private String mAuthor;

/** Web publication date of the article */
private String mDate;

/** Website URL of the article */
private String mUrl;

/** Thumbnail of the article */
private String mThumbnail;
```

```java
/** TrailText of the article with string type Html */
private String mTrailTextHtml;

/**
      * Constructs a new {@link News} object.
      *
      * @param title is the title of the article
      * @param section is the section name of the article
      * @param author is author name in article
      * @param date is the web publication date of the article
      * @param url is the website URL to find more details about the
article
      * @param thumbnail is the thumbnail of the article
      * @param trailText is trail text of the article with string type
Html
      */
public News(String title, String section, String author, String date,
String url, String thumbnail, String trailText) {
mTitle = title;
mSection = section;
mAuthor = author;
mDate = date;
mUrl = url;
mThumbnail = thumbnail;
mTrailTextHtml = trailText;
}

/**
      * Returns the title of the article
      */
public String getTitle() {
return mTitle;
}

/**
      * Returns the section name of the article.
      */
public String getSection() {
return mSection;
}

/**
      * Returns the author name of the article.
      */
public String getAuthor() {
return mAuthor;
}
/**
      * Returns the web publication date of the article.
      */
public String getDate() {
return mDate;
}

/**
      * Returns the website URL to find more information about the news.
```

```
     */
public String getUrl() {
return mUrl;
}

/**
     * Returns the thumbnail of the article
     */
public String getThumbnail() {
return mThumbnail;
}

/**
     * Returns the TrailText of the article with string type Html
     */
public String getTrailTextHtml() {
return mTrailTextHtml;
}
}
```

NewsLoader.java

```
packagecom.example.android.newsfeed;

import android.support.v4.content.AsyncTaskLoader;
import android.content.Context;

import com.example.android.newsfeed.utils.QueryUtils;

import java.util.List;

/**
 * Loads a list of news by using an AsyncTask to perform the network
request to the given URL.
 */
public class NewsLoader extends AsyncTaskLoader<List<News>> {

/** Tag for log messages */
private static final String LOG_TAG = NewsLoader.class.getName();

/** Query URL */
private String mUrl;

/**
     * Constructs a new {@link NewsLoader}.
     *
     * @param context of the activity
     * @param url to load data from
     */
public NewsLoader(Context context, String url) {
super(context);
mUrl = url;
}

@Override
protected void onStartLoading() {
// Trigger the loadInBackground() method to execute.
forceLoad();
}

/**
      * This is on a background thread.
      */
@Override
public List<News>loadInBackground() {
if (mUrl == null) {
return null;
}

// Perform the network request, parse the response, and extract a list
of news.
List<News> newsData = QueryUtils.fetchNewsData(mUrl);
        return newsData;
}
}
```

NewsPreferences.java

```
package com.example.android.newsfeed;

import android.content.Context;
import android.content.SharedPreferences;
import android.net.Uri;
import android.preference.PreferenceManager;

import com.example.android.newsfeed.utils.Constants;

import static com.example.android.newsfeed.utils.Constants.API_KEY;
import static
com.example.android.newsfeed.utils.Constants.API_KEY_PARAM;
import static com.example.android.newsfeed.utils.Constants.FORMAT;
import static com.example.android.newsfeed.utils.Constants.FORMAT_PARAM;
import static
com.example.android.newsfeed.utils.Constants.FROM_DATE_PARAM;
import static
com.example.android.newsfeed.utils.Constants.ORDER_BY_PARAM;
import static
com.example.android.newsfeed.utils.Constants.ORDER_DATE_PARAM;
import static
com.example.android.newsfeed.utils.Constants.PAGE_SIZE_PARAM;
import static com.example.android.newsfeed.utils.Constants.QUERY_PARAM;
import static
com.example.android.newsfeed.utils.Constants.SECTION_PARAM;
import static com.example.android.newsfeed.utils.Constants.SHOW_FIELDS;
import static
com.example.android.newsfeed.utils.Constants.SHOW_FIELDS_PARAM;
import static com.example.android.newsfeed.utils.Constants.SHOW_TAGS;
import static
com.example.android.newsfeed.utils.Constants.SHOW_TAGS_PARAM;
```

```
public final class NewsPreferences {

/**
     * Get Uri.Builder based on stored SharedPreferences.
     * @param context Context used to access SharedPreferences
     * @return Uri.Builder
     */
public static Uri.Builder getPreferredUri(Context context) {
        SharedPreferences sharedPrefs =
PreferenceManager.getDefaultSharedPreferences(context);

// getString retrieves a String value from the preferences. The second
parameter is the
        // default value for this preference.
String numOfItems = sharedPrefs.getString(

context.getString(R.string.settings_number_of_items_key),
context.getString(R.string.settings_number_of_items_default));

// Get the information from SharedPreferences and check for the value
associated with the key
String orderBy = sharedPrefs.getString(
                context.getString(R.string.settings_order_by_key),
context.getString(R.string.settings_order_by_default));

// Get the orderDate information from SharedPreferences and check for
the value associated with the key
String orderDate = sharedPrefs.getString(
                context.getString(R.string.settings_order_date_key),
context.getString(R.string.settings_order_date_default));

// Get the fromDate information from SharedPreferences and check for the
value associated with the key
String fromDate = sharedPrefs.getString(
                context.getString(R.string.settings_from_date_key),
context.getString(R.string.settings_from_date_default));

// Parse breaks apart the URI string that is passed into its parameter
Uri baseUri = Uri.parse(Constants.NEWS_REQUEST_URL);

// buildUpon prepares the baseUri that we just parsed so we can add
query parameters to it
Uri.Builder uriBuilder = baseUri.buildUpon();

// Append query parameter and its value. (e.g. the 'show-
tag=contributor')
uriBuilder.appendQueryParameter(QUERY_PARAM, "");
uriBuilder.appendQueryParameter(ORDER_BY_PARAM, orderBy);
uriBuilder.appendQueryParameter(PAGE_SIZE_PARAM, numOfItems);
uriBuilder.appendQueryParameter(ORDER_DATE_PARAM, orderDate);
uriBuilder.appendQueryParameter(FROM_DATE_PARAM, fromDate);
uriBuilder.appendQueryParameter(SHOW_FIELDS_PARAM, SHOW_FIELDS);
uriBuilder.appendQueryParameter(FORMAT_PARAM, FORMAT);
uriBuilder.appendQueryParameter(SHOW_TAGS_PARAM, SHOW_TAGS);
uriBuilder.appendQueryParameter(API_KEY_PARAM, API_KEY); // Use your API
key when API rate limit exceeded
```

```java
return uriBuilder;
}

/**
     * Returns String Url for query
     * @param context Context used to access getPreferredUri method
     * @param section News section
     */
public static String getPreferredUrl(Context context, String section) {
        Uri.Builder uriBuilder = getPreferredUri(context);
        return uriBuilder.appendQueryParameter(SECTION_PARAM,
section).toString();
}
}
```

SettingsActivity.java

```java
package com.example.android.newsfeed;

import android.app.DatePickerDialog;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.preference.ListPreference;
import android.preference.Preference;
import android.preference.PreferenceFragment;
import android.preference.PreferenceManager;
import android.support.v7.app.AppCompatActivity;
import android.view.MenuItem;
import android.widget.DatePicker;

import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;

/**
 * The SettingsActivity is the activity that appears when a settings
icon is clicked on.
 */

public class SettingsActivity extends AppCompatActivity {
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.settings_activity);

// Navigate with the app icon in the action bar
getSupportActionBar().setDisplayHomeAsUpEnabled(true);
getSupportActionBar().setDisplayShowHomeEnabled(true);
}

/**
     * The NewsPreferenceFragment implements the
Preference.OnPreferenceChangeListener interface
```

```
     * to set up to listen for any Preference changes made by the user.
     * And the NewsPreferenceFragment also implements the
DatePickerDialog.OnDateSetListener to
     * receive a callback when the user has finished selecting a date.
     */
public static class NewsPreferenceFragment extends PreferenceFragment
implements Preference.OnPreferenceChangeListener,
DatePickerDialog.OnDateSetListener {

@Override
public void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
addPreferencesFromResource(R.xml.settings_main);

// Find the preference for number of items
Preference numOfItems =
findPreference(getString(R.string.settings_number_of_items_key));
// bind the current preference value to be displayed
bindPreferenceSummaryToValue(numOfItems);

// Find the "order by" Preference object according to its key
Preference orderBy =
findPreference(getString(R.string.settings_order_by_key));
// Update the summary so that it displays the current value stored in
SharedPreferences
bindPreferenceSummaryToValue(orderBy);

// Find the "order date" Preference object according to its key
Preference orderDate =
findPreference(getString(R.string.settings_order_date_key));
// Update the summary so that it displays the current value stored in
SharedPreferences
bindPreferenceSummaryToValue(orderDate);

// Find the "color theme" Preference object according to its key
Preference colorTheme =
findPreference(getString(R.string.settings_color_key));
// Update the summary so that it displays the current value stored in
SharedPreferences
bindPreferenceSummaryToValue(colorTheme);

// Find the "text size" Preference object according to its key
Preference textSize =
findPreference(getString(R.string.settings_text_size_key));
// Update the summary so that it displays the current value stored in
SharedPreferences
bindPreferenceSummaryToValue(textSize);

// Find the "from date" Preference object according to its key
Preference fromDate =
findPreference(getString(R.string.settings_from_date_key));
setOnPreferenceClick(fromDate);
// bind the current preference value to be displayed
bindPreferenceSummaryToValue(fromDate);
}
```

```
/**
         * This method is called when the user has clicked a Preference.
         */
private void setOnPreferenceClick(Preference preference) {
            preference.setOnPreferenceClickListener(new
Preference.OnPreferenceClickListener() {
@Override
public boolean onPreferenceClick(Preference preference) {
                    String key = preference.getKey();
                    if
(key.equalsIgnoreCase(getString(R.string.settings_from_date_key))) {
                        showDatePicker();
}
return false;
}
            });
}

/**
         * Show the current date as the default date in the picker
         */
private void showDatePicker() {
            Calendar calendar = Calendar.getInstance();
            int year = calendar.get(Calendar.YEAR);
            int month = calendar.get(Calendar.MONTH);
            int dayOfMonth = calendar.get(Calendar.DAY_OF_MONTH);
            new DatePickerDialog(getActivity(),
                   this, year, month, dayOfMonth).show();
}

@Override
public void onDateSet(DatePicker datePicker, int year, int month, int
dayOfMonth) {
// Since it starts counting the months from 0, add one to the month
value.
month = month + 1;
// Date the user selected
String selectedDate = year + "-" + month + "-" + dayOfMonth;
// Convert selected date string(i.e. "2017-2-1" into formatted date
string(i.e. "2017-02-01")
String formattedDate = formatDate(selectedDate);

// Storing selected date
SharedPreferences prefs =
PreferenceManager.getDefaultSharedPreferences(getActivity());
SharedPreferences.Editor editor = prefs.edit();
editor.putString(getString(R.string.settings_from_date_key),
formattedDate).apply();

// Update the displayed preference summary after it has been changed
Preference fromDatePreference =
findPreference(getString(R.string.settings_from_date_key));
bindPreferenceSummaryToValue(fromDatePreference);
}

/**
```

```
         * This method is called when the user has changed a Preference.
         * Update the displayed preference summary (the UI) after it has
been changed.
         * @param preference the changed Preference
         * @param value the new value of the Preference
         * @return True to update the state of the Preference with the
new value
         */
@Override
public boolean onPreferenceChange(Preference preference, Object value) {
            String stringValue = value.toString();
// Update the summary of a ListPreference using the label
if (preference instanceof ListPreference) {
                ListPreference listPreference = (ListPreference)
preference;
                int prefIndex =
listPreference.findIndexOfValue(stringValue);
                if (prefIndex >= 0) {
                    CharSequence[] labels = listPreference.getEntries();
preference.setSummary(labels[prefIndex]);
}
            } else {
                preference.setSummary(stringValue);
}
return true;
}

/**
         * Set this fragment as the OnPreferenceChangeListener and
         * bind the value that is in SharedPreferences to what will show
up in the preference summary
         */
private void bindPreferenceSummaryToValue(Preference preference) {
// Set the current NewsPreferenceFragment instance to listen for changes
to the preference
            // we pass in using
preference.setOnPreferenceChangeListener(this);

// Read the current value of the preference stored in the
SharedPreferences on the device,
            // and display that in the preference summary
SharedPreferences preferences =

PreferenceManager.getDefaultSharedPreferences(preference.getContext());
String preferenceString = preferences.getString(preference.getKey(),
"");
onPreferenceChange(preference, preferenceString);
}

/**
         * Convert selected date string(i.e. "2017-2-1" into formatted
date string(i.e. "2017-02-01")
         *
         * @param dateString is the selected date from the DatePicker
         * @return the formatted date string
         */
private String formatDate(String dateString) {
            SimpleDateFormat simpleDateFormat = new
SimpleDateFormat("yyyy-M-d");
Date dateObject = null;
            try {
                dateObject = simpleDateFormat.parse(dateString);
} catch (ParseException e) {
                e.printStackTrace();
}
            SimpleDateFormat df= new SimpleDateFormat("yyyy-MM-dd");
            return df.format(dateObject);
}
    }

// Go back to the MainActivity when up button in action bar is clicked
on.
@Override
public boolean onOptionsItemSelected(MenuItem item) {
switch (item.getItemId()) {
case android.R.id.home:
                finish();
                return true;
}
return super.onOptionsItemSelected(item);
}
}
```

Layout:

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.v4.widget.DrawerLayout
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:id="@+id/drawer_layout"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:fitsSystemWindows="true"
tools:openDrawer="start">

<include
layout="@layout/app_bar_main"
android:layout_width="match_parent"
android:layout_height="match_parent" />

<android.support.design.widget.NavigationView
android:id="@+id/nav_view"
android:layout_width="wrap_content"
android:layout_height="match_parent"
android:layout_gravity="start"
android:fitsSystemWindows="true"
app:headerLayout="@layout/nav_header_main"
app:menu="@menu/activity_main_drawer" />
```

```
</android.support.v4.widget.DrawerLayout>
```

app_bar_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout
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="com.example.android.newsfeed.MainActivity">

<android.support.design.widget.AppBarLayout
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:theme="@style/AppTheme.AppBarOverlay">

<android.support.v7.widget.Toolbar
android:id="@+id/toolbar"
android:layout_width="match_parent"
android:layout_height="?attr/actionBarSize"
android:background="@android:color/white"
app:popupTheme="@style/AppTheme.PopupOverlay" />

<!-- TabLayout that provides a horizontal layout to display tabs -->
<android.support.design.widget.TabLayout
android:id="@+id/sliding_tabs"
style="@style/CategoryTab"
android:background="@android:color/white"
android:layout_width="match_parent"
android:layout_height="wrap_content"
app:tabMode="scrollable" />

</android.support.design.widget.AppBarLayout>

<include layout="@layout/content_main" />

</android.support.design.widget.CoordinatorLayout>
```

Content_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
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"
app:layout_behavior="@string/appbar_scrolling_view_behavior"
```

```
tools:context="com.example.android.newsfeed.MainActivity"
tools:showIn="@layout/app_bar_main">

<android.support.v4.view.ViewPager
android:id="@+id/viewpager"
android:layout_width="match_parent"
android:layout_height="match_parent">

<FrameLayout
android:id="@+id/content_frame"
android:layout_width="match_parent"
android:layout_height="match_parent">
</FrameLayout>
</android.support.v4.view.ViewPager>

</android.support.constraint.ConstraintLayout>
```

fragment_home.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="match_parent">

<android.support.v4.widget.SwipeRefreshLayout
android:id="@+id/swipe_refresh"
android:layout_width="match_parent"
android:layout_height="match_parent">

<!-- Replaced android.support.v7.widget.RecyclerView with the new
EmptyRecyclerView -->
<com.example.android.newsfeed.EmptyRecyclerView
android:id="@+id/recycler_view"
android:layout_width="match_parent"
android:layout_height="match_parent" />

</android.support.v4.widget.SwipeRefreshLayout>

<!-- Empty view is only visible when the list has no items. -->
<TextView
android:id="@+id/empty_view"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_centerInParent="true"
android:gravity="center"
android:textAppearance="?android:textAppearanceMedium" />

<!-- Loading indicator is only shown before the first load -->
<ProgressBar
android:id="@+id/loading_indicator"
style="@style/Widget.AppCompat.ProgressBar"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_centerInParent="true" />
```

```
</RelativeLayout>
```

nav_header_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent"
android:layout_height="@dimen/nav_header_height"
android:background="@drawable/side_nav_bar"
android:gravity="bottom"
android:orientation="vertical"
android:paddingBottom="@dimen/activity_vertical_margin"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"
android:theme="@style/ThemeOverlay.AppCompat.Dark">

<TextView
android:id="@+id/textView"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:fontFamily="sans-serif"
android:text="@string/Khabar"
android:textSize="45sp"
android:textStyle="bold" />

</LinearLayout>
```

news_card_item.xml

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

<android.support.v7.widget.CardView
android:id="@+id/card_view"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_margin="@dimen/layout_margin_5"
android:clickable="true"
android:focusable="true"
android:foreground="?android:attr/selectableItemBackground"
card_view:cardCornerRadius="@dimen/card_corner_radius">

<LinearLayout
```

```xml
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginBottom="@dimen/layout_margin"
android:orientation="vertical">

<TextView
android:id="@+id/title_card"
style="@style/TitleTextViewStyle"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:textAppearance="?android:textAppearanceLarge"
android:textStyle="bold"
tools:text="title" />

<RelativeLayout
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginBottom="@dimen/layout_margin_8"
android:layout_marginTop="@dimen/layout_margin_4">

<TextView
android:id="@+id/section_card"
style="@style/SectionTextViewStyle"
android:layout_alignParentStart="true"
android:layout_toStartOf="@id/thumbnail_image_card"
tools:text="section" />

<TextView
android:id="@+id/trail_text_card"
style="@style/TrailTextViewStyle"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentStart="true"
android:layout_below="@+id/section_card"
android:layout_toStartOf="@+id/thumbnail_image_card"
tools:text="trailText" />

<ImageView
android:id="@+id/thumbnail_image_card"
android:layout_width="@dimen/thumbnail_image_width"
android:layout_height="@dimen/thumbnail_image_height"
android:layout_alignParentEnd="true"
android:contentDescription="@string/image_des"
android:scaleType="centerCrop" />
</RelativeLayout>

<TextView
android:id="@+id/author_card"
style="@style/AuthorTextViewStyle"
tools:text="author" />

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

<TextView
```

```
android:id="@+id/date_card"
style="@style/DateTextViewStyle"
android:layout_alignParentStart="true"
android:layout_toStartOf="@+id/share_image_card"
tools:text="date" />

<ImageView
android:id="@+id/share_image_card"
android:layout_width="@dimen/image_share"
android:layout_height="@dimen/image_share"
android:layout_alignParentEnd="true"
android:layout_marginEnd="@dimen/layout_margin"
android:background="@drawable/image_button_style"
android:contentDescription="@string/image_des_ic_share"
android:src="@drawable/ic_share_black_18dp" />

</RelativeLayout>

</LinearLayout>

</android.support.v7.widget.CardView>

</RelativeLayout>
```

settings_activity.xml

```
<?xml version="1.0" encoding="utf-8"?>
<fragment xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/fragment_settings"
android:name="com.example.android.newsfeed.SettingsActivity$NewsPreferen
ceFragment"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context="com.example.android.newsfeed.SettingsActivity">

</fragment>
```

activity_main_drawer.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
tools:showIn="navigation_view">

<group android:checkableBehavior="single">
<item
android:id="@+id/nav_home"
android:icon="@drawable/ic_menu_home"
android:title="@string/ic_title_home" />
<item
android:id="@+id/nav_world"
```

```
android:icon="@drawable/ic_menu_world"
android:title="@string/ic_title_world" />
<item
android:id="@+id/nav_science"
android:icon="@drawable/ic_menu_science"
android:title="@string/ic_title_science" />
<item
android:id="@+id/nav_sport"
android:icon="@drawable/ic_menu_sport"
android:title="@string/ic_title_sport" />
<item
android:id="@+id/nav_environment"
android:icon="@drawable/ic_menu_environment"
android:title="@string/ic_title_environment" />
<item
android:id="@+id/nav_society"
android:icon="@drawable/ic_menu_society"
android:title="@string/ic_title_society" />
<item
android:id="@+id/nav_fashion"
android:icon="@drawable/ic_menu_fashion"
android:title="@string/ic_title_fashion" />
<item
android:id="@+id/nav_business"
android:icon="@drawable/ic_menu_business"
android:title="@string/ic_title_business" />
<item
android:id="@+id/nav_culture"
android:icon="@drawable/ic_menu_culture"
android:title="@string/ic_title_culture" />
</group>
</menu>
```

Main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
tools:context="com.example.android.newsfeed.MainActivity">
<item
android:id="@+id/action_settings"
android:title="@string/action_settings" />
</menu>
```

arrays.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
<!--  for the human-readable labels -->
<string-array name="settings_order_by_labels">
<item>@string/settings_order_by_newest_label</item>
<item>@string/settings_order_by_oldest_label</item>
<item>@string/settings_order_by_relevance_label</item>
</string-array>

<!-- for the value to save for a preference when an entry from entries
is selected -->
<string-array name="settings_order_by_values">
```

```xml
<item>@string/settings_order_by_newest_value</item>
<item>@string/settings_order_by_oldest_value</item>
<item>@string/settings_order_by_relevance_value</item>
</string-array>

<string-array name="settings_order_date_labels">
<item>@string/settings_order_date_published_label</item>
<item>@string/settings_order_date_newspaper_edition_label</item>
<item>@string/settings_order_date_last_modified_label</item>
</string-array>

<string-array name="settings_order_date_values">
<item>@string/settings_order_date_published_value</item>
<item>@string/settings_order_date_newspaper_edition_value</item>
<item>@string/settings_order_date_last_modified_value</item>
</string-array>

<!--  for the human-readable labels -->
<string-array name="settings_color_labels">
<item>@string/settings_color_white_label</item>
<item>@string/settings_color_sky_blue_label</item>
<item>@string/settings_color_dark_blue_label</item>
<item>@string/settings_color_violet_label</item>
<item>@string/settings_color_light_green_label</item>
<item>@string/settings_color_green_label</item>
</string-array>

<!-- for the value to save for a preference when an entry from entries
is selected -->
<string-array name="settings_color_values">
<item>@string/settings_color_white_value</item>
<item>@string/settings_color_sky_blue_value</item>
<item>@string/settings_color_dark_blue_value</item>
<item>@string/settings_color_violet_value</item>
<item>@string/settings_color_light_green_value</item>
<item>@string/settings_color_green_value</item>
</string-array>

<!--String array for the text size -->
<string-array name="settings_text_size_labels">
<item>@string/settings_text_size_small_label</item>
<item>@string/settings_text_size_medium_label</item>
<item>@string/settings_text_size_large_label</item>
</string-array>
<string-array name="settings_text_size_values">
<item>@string/settings_text_size_small_value</item>
<item>@string/settings_text_size_medium_value</item>
<item>@string/settings_text_size_large_value</item>
</string-array>
</resources>
```

colors.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<resources>
```

```xml
<color name="colorPrimary">#005689</color>
<color name="colorPrimaryDark">#005689</color>
<color name="colorAccent">#FF4081</color>

<color name="swipe_color_1">#4CAF50</color>
<color name="swipe_color_2">#03A9F4</color>
<color name="swipe_color_3">#FFC107</color>
<color name="swipe_color_4">#FF5722</color>

<color name="color_section">#4CAF50</color>
<color name="color_author">#8E24AA</color>

<!-- Colors for settings of color theme-->
<color name="white">#ffffff</color>
<color name="violet">#9c27b0</color>
<color name="light_green">#8bc34a</color>

<!-- Colors for gradient.xml file -->
<color name="gradient_center_color">#880d0d0f</color>
<color name="gradient_end_color">#885d5d5e</color>
<color name="gradient_start_color">#880f0f10</color>

<color name="color_trail_text">#102027</color>
<color name="color_date_text">#a4a4a4</color>

<color name="color_app_bar_text">#005689</color>
<color name="color_tab_text">#a7c0cd</color>

<color name="nav_bar_start">#7db1ec</color>
<color name="nav_bar_end">#002e5b</color>
</resources>
```

dimen.xml

```xml
<resources>
<!-- Default screen margins, per the Android Design guidelines. -->
<dimen name="activity_horizontal_margin">16dp</dimen>
<dimen name="activity_vertical_margin">16dp</dimen>
<dimen name="nav_header_height">176dp</dimen>

<dimen name="padding">16dp</dimen>
<dimen name="layout_margin">16dp</dimen>
<dimen name="image_share">18dp</dimen>
<dimen name="padding_bottom_4">4dp</dimen>
<dimen name="layout_margin_4">4dp</dimen>
<dimen name="layout_margin_8">8dp</dimen>
<dimen name="layout_margin_5">5dp</dimen>
<dimen name="card_corner_radius">2dp</dimen>
<dimen name="thumbnail_image_width">140dp</dimen>
<dimen name="thumbnail_image_height">120dp</dimen>

<dimen name="sp22">22sp</dimen>
<dimen name="sp16">16sp</dimen>
<dimen name="sp14">14sp</dimen>
<dimen name="sp20">20sp</dimen>
```

```xml
<dimen name="sp12">12sp</dimen>
<dimen name="sp24">24sp</dimen>
<dimen name="sp18">18sp</dimen>
</resources>
```

strings.xml

```xml
<resources>
<string name="app_name">Khabar</string>

<string name="navigation_drawer_open">Open navigation drawer</string>
<string name="navigation_drawer_close">Close navigation drawer</string>

<string name="action_settings">Settings</string>

<!-- Strings for items in activity_main_drawer.xml -->
<string name="ic_title_home">Home</string>
<string name="ic_title_world">World</string>
<string name="ic_title_science">Science</string>
<string name="ic_title_sport">Sport</string>
<string name="ic_title_environment">Environment</string>
<string name="ic_title_society">Society</string>
<string name="ic_title_fashion">Fashion</string>
<string name="ic_title_business">Business</string>
<string name="ic_title_culture">Culture</string>

<!-- Text to display in the list when there are no news [CHAR
LIMIT=NONE] -->
<string name="no_news">No news found.</string>

<!-- Error message when there is no internet connectivity [CHAR
LIMIT=NONE] -->
<string name="no_internet_connection">You are offline. \nPlease check
your Internet
        connection.</string>

<!-- Content description of the image -->
<string name="image_des">Image</string>
<string name="image_des_ic_share">Share</string>

<string name="updated_just_now">Updated just now</string>
<string name="share_article">Share Article</string>

<string name="Khabar">Khabar</string>

<!-- Strings for Section -->
<string name="world">world</string>
<string name="science">science</string>
<string name="sport">sport</string>
<string name="environment">environment</string>
<string name="society">society</string>
<string name="fashion">fashion</string>
<string name="business">business</string>
```

```
<string name="culture">culture</string>

<string name="hint_num_of_items">Between 1 and 50</string>

<!--Strings to be used for the Preference Labels -->
<string name="settings_number_of_items_label">Number of Items</string>
<string name="settings_number_of_items_key"
translatable="false">number_of_items</string>
<string name="settings_number_of_items_default"
translatable="false">10</string>

<!-- Strings for Order-By preference [CHAR LIMIT=30] -->
<string name="settings_order_by_label">Order By</string>
<string name="settings_order_by_key"
translatable="false">order_by</string>
<string name="settings_order_by_default"
translatable="false">@string/settings_order_by_newest_value</string>

<!-- Label for order-by newest option [CHAR LIMIT=20] -->
<string name="settings_order_by_newest_label">Newest</string>
<string name="settings_order_by_newest_value"
translatable="false">newest</string>

<!-- Label for order-by oldest option [CHAR LIMIT=20] -->
<string name="settings_order_by_oldest_label">Oldest</string>
<string name="settings_order_by_oldest_value"
translatable="false">oldest</string>

<!-- Label for order-by relevance option [CHAR LIMIT=20]-->
<string name="settings_order_by_relevance_label">Relevance</string>
<string name="settings_order_by_relevance_value"
translatable="false">relevance</string>

<!-- Strings for From Date preference [CHAR LIMIT=30] -->
<string name="settings_from_date_label">From Date</string>
<string name="settings_from_date_key" translatable="false">from
date</string>
<string name="settings_from_date_default" translatable="false">2010-01-
01</string>

<!-- Strings for Order Date preference [CHAR LIMIT=30] -->
<string name="settings_order_date_label">Order Date</string>
<string name="settings_order_date_key" translatable="false">order
date</string>
<string name="settings_order_date_default"
translatable="false">@string/settings_order_date_published_value</string
>

<!-- Label for order-date published option [CHAR LIMIT=20]-->
<string name="settings_order_date_published_label">Published</string>
<string name="settings_order_date_published_value"
translatable="false">published</string>

<!-- Label for order-date newspaper-edition option [CHAR LIMIT=30]-->
<string name="settings_order_date_newspaper_edition_label">Newspaper
Edition</string>
```

```
<string name="settings_order_date_newspaper_edition_value"
translatable="false">newspaper-edition</string>

<!-- Label for order-date last-modified option [CHAR LIMIT=30]-->
<string name="settings_order_date_last_modified_label">Last
Modified</string>
<string name="settings_order_date_last_modified_value"
translatable="false">last-modified</string>

<!-- Strings for Color-Theme preference [CHAR LIMIT=30] -->
<string name="settings_color_label">Color Theme</string>
<string name="settings_color_key">color theme</string>
<string name="settings_color_default">white</string>

<!-- Label for color theme white preference [CHAR LIMIT=20] -->
<string name="settings_color_white_label">White</string>
<string name="settings_color_white_value">white</string>

<!-- Label for color theme sky blue preference [CHAR LIMIT=20] -->
<string name="settings_color_sky_blue_label">Sky Blue</string>
<string name="settings_color_sky_blue_value">sky blue</string>

<!-- Label for color theme dark blue preference [CHAR LIMIT=20] -->
<string name="settings_color_dark_blue_label">Dark Blue</string>
<string name="settings_color_dark_blue_value">dark blue</string>

<!-- Label for color theme violet preference [CHAR LIMIT=20] -->
<string name="settings_color_violet_label">Violet</string>
<string name="settings_color_violet_value">violet</string>

<!-- Label for color theme light green preference [CHAR LIMIT=20] -->
<string name="settings_color_light_green_label">Light Green</string>
<string name="settings_color_light_green_value">light green</string>

<!-- Label for color theme green preference [CHAR LIMIT=20] -->
<string name="settings_color_green_label">Green</string>
<string name="settings_color_green_value">green</string>

<!-- Strings for Text Size preference [CHAR LIMIT=30] -->
<string name="settings_text_size_label">Text Size</string>
<string name="settings_text_size_key" translatable="false">text
size</string>
<string name="settings_text_size_default"
translatable="false">@string/settings_text_size_medium_value</string>

<!-- Label for text size small preference [CHAR LIMIT=20] -->
<string name="settings_text_size_small_label">Small</string>
<string name="settings_text_size_small_value"
translatable="false">small</string>

<!-- Label for text size medium preference [CHAR LIMIT=20] -->
<string name="settings_text_size_medium_label">Medium</string>
<string name="settings_text_size_medium_value"
translatable="false">medium</string>

<!-- Label for text size large preference [CHAR LIMIT=20] -->
```

```
<string name="settings_text_size_large_label">Large</string>
<string name="settings_text_size_large_value"
translatable="false">large</string>

</resources>
```

styles.xml

```
<resources>

<!-- Base application theme. -->
<style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
<!-- Customize your theme here. -->
<item name="colorPrimary">@color/colorPrimary</item>
<item name="colorPrimaryDark">@color/colorPrimaryDark</item>
<item name="colorAccent">@color/colorAccent</item>

<!-- Change Action Bar title text color  -->
<item name="titleTextColor">@color/color_app_bar_text</item>
<item name="subtitleTextColor">@color/color_app_bar_text</item>

<!-- Set Navigation drawer icon color-->
<item name="drawerArrowStyle">@style/DrawerArrowStyle</item>

<!-- Change Action Bar overflow icon (Options Menu button) color -->
<item
name="actionOverflowButtonStyle">@style/MyOverflowButtonStyle</item>
</style>

<style name="AppTheme.NoActionBar">
<item name="windowActionBar">false</item>
<item name="windowNoTitle">true</item>
</style>

<style name="AppTheme.AppBarOverlay"
parent="ThemeOverlay.AppCompat.Dark.ActionBar" />

<style name="AppTheme.PopupOverlay"
parent="ThemeOverlay.AppCompat.Light" />

<!-- Theme of the settings -->
<style name="SettingsTheme"
parent="Theme.AppCompat.Light.DarkActionBar">
<item name="colorPrimary">@color/colorPrimary</item>
<item name="colorPrimaryDark">@color/colorPrimaryDark</item>
<item name="colorAccent">@color/colorAccent</item>
</style>

<!-- Set Navigation drawer icon color -->
<style name="DrawerArrowStyle"
parent="Widget.AppCompat.DrawerArrowToggle">
<item name="color">@color/color_app_bar_text</item>
</style>
```

```xml
<!-- Change Action Bar overflow icon (Options Menu button) color -->
<style name="MyOverflowButtonStyle"
parent="Widget.AppCompat.ActionButton.Overflow">
<item name="android:tint">@color/color_app_bar_text</item>
</style>

<!-- Style for a tab that displays a category name -->
<style name="CategoryTab" parent="Widget.Design.TabLayout">
<item name="tabIndicatorColor">@color/color_app_bar_text</item>
<item name="tabSelectedTextColor">@color/color_app_bar_text</item>
<item name="tabTextAppearance">@style/CategoryTabTextAppearance</item>
</style>

<!-- Text appearance style for a category tab -->
<style name="CategoryTabTextAppearance"
parent="TextAppearance.Design.Tab">
<item name="android:textColor">@color/color_tab_text</item>
</style>

<!-- Style for title TextView-->
<style name="TitleTextViewStyle">
<item name="android:paddingBottom">@dimen/padding_bottom_4</item>
<item name="android:paddingTop">@dimen/padding</item>
<item name="android:fontFamily">sans-serif-condensed</item>
<item name="android:paddingLeft">@dimen/padding</item>
<item name="android:paddingRight">@dimen/padding</item>
</style>

<!-- Style for Section, Author, and Date TextView -->
<style name="SectionAuthorDateStyle">
<item name="android:layout_width">wrap_content</item>
<item name="android:layout_height">wrap_content</item>
<item name="android:layout_marginLeft">@dimen/layout_margin</item>
<item name="android:layout_marginRight">@dimen/layout_margin</item>
<item name="android:fontFamily">sans-serif</item>
</style>

<!-- Style for section TextView -->
<style name="SectionTextViewStyle"
parent="@style/SectionAuthorDateStyle">
<item name="android:ellipsize">end</item>
<item name="android:maxLines">1</item>
<item name="android:textColor">@color/color_section</item>
</style>

<!-- Style for trailText TextView -->
<style name="TrailTextViewStyle">
<item name="android:layout_marginBottom">@dimen/layout_margin_8</item>
<item name="android:layout_marginLeft">@dimen/layout_margin</item>
<item name="android:layout_marginRight">@dimen/layout_margin</item>
<item name="android:ellipsize">end</item>
<item name="android:fontFamily">sans-serif-light</item>
<item name="android:maxLines">6</item>
<item name="android:textColor">@color/color_trail_text</item>
</style>
```

```xml
<!-- Style for Author TextView -->
<style name="AuthorTextViewStyle"
parent="@style/SectionAuthorDateStyle">
<item name="android:textColor">@color/color_author</item>
<item name="android:textStyle">italic</item>
</style>

<!-- Style for Date TextView -->
<style name="DateTextViewStyle" parent="@style/SectionAuthorDateStyle">
<item name="android:textColor">@color/color_date_text</item>
</style>

</resources>
```

settings_main.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<PreferenceScreen
xmlns:android="http://schemas.android.com/apk/res/android"
android:title="@string/action_settings">

<EditTextPreference
android:defaultValue="@string/settings_number_of_items_default"
android:hint="@string/hint_num_of_items"
android:inputType="numberDecimal"
android:key="@string/settings_number_of_items_key"
android:selectAllOnFocus="true"
android:title="@string/settings_number_of_items_label" />

<ListPreference
android:defaultValue="@string/settings_order_by_default"
android:entries="@array/settings_order_by_labels"
android:entryValues="@array/settings_order_by_values"
android:key="@string/settings_order_by_key"
android:title="@string/settings_order_by_label" />

<Preference
android:defaultValue="@string/settings_from_date_default"
android:key="@string/settings_from_date_key"
android:title="@string/settings_from_date_label" />

<ListPreference
android:defaultValue="@string/settings_order_date_default"
android:entries="@array/settings_order_date_labels"
android:entryValues="@array/settings_order_date_values"
android:key="@string/settings_order_date_key"
android:title="@string/settings_order_date_label" />

<PreferenceCategory android:title="Customize Styles">
<ListPreference
android:defaultValue="@string/settings_color_default"
android:entries="@array/settings_color_labels"
android:entryValues="@array/settings_color_values"
android:key="@string/settings_color_key"
```

```
android:title="@string/settings_color_label" />

<ListPreference
android:defaultValue="@string/settings_text_size_default"
android:entries="@array/settings_text_size_labels"
android:entryValues="@array/settings_text_size_values"
android:key="@string/settings_text_size_key"
android:title="@string/settings_text_size_label" />

</PreferenceCategory>
</PreferenceScreen>
```

build.gradle

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 28
defaultConfig {
applicationId "com.example.android.newsfeed"
minSdkVersion 17
targetSdkVersion 28
versionCode 1
versionName "1.0"
testInstrumentationRunner
"android.support.test.runner.AndroidJUnitRunner"
}
    buildTypes {
        release {
minifyEnabled false
proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-
rules.pro'
}
    }
}

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:28.0.0'
implementation 'com.android.support:design:28.0.0'
implementation 'com.android.support.constraint:constraint-layout:1.1.3'
implementation 'com.android.support:support-v4:28.0.0'
testImplementation 'junit:junit:4.12'
androidTestImplementation 'com.android.support.test:runner:1.0.2'
androidTestImplementation'com.android.support.test.espresso:espresso-
core:3.0.2'

implementation 'com.android.support:recyclerview-v7:28.0.0'
implementation 'com.android.support:cardview-v7:28.0.0'

implementation 'com.github.bumptech.glide:glide:4.7.1'
annotationProcessor 'com.github.bumptech.glide:compiler:4.7.1'
}
```

Icons Used:

Steps to Run The Project

By GUI:

1.Import Project in Android Studio

2.Connect Android Device or Android Emulator

3.Click on run▶ button it will install apk on device or emulator

By Terminal:

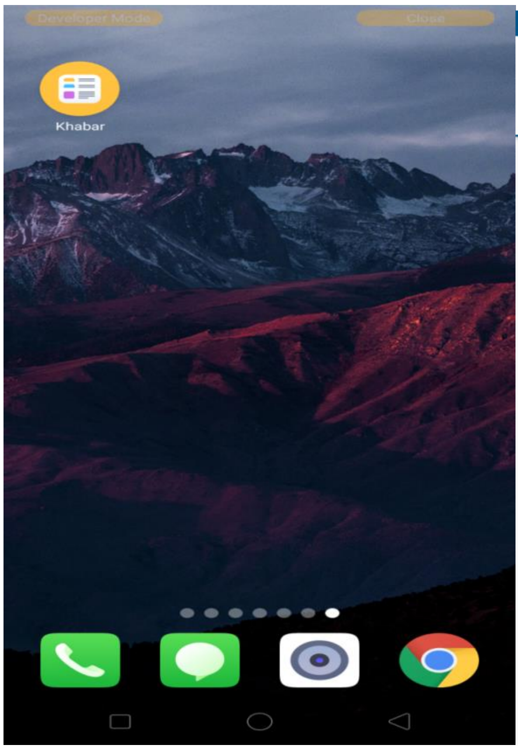
Emulator-

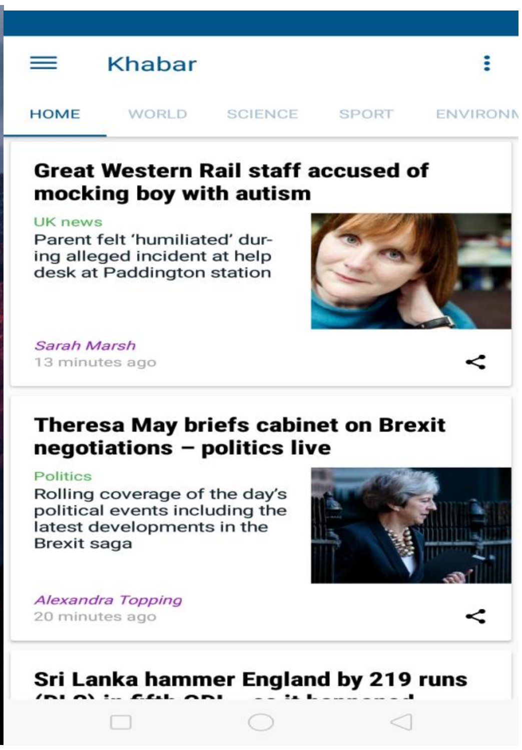
```
Emulator –avd avd_name
adb install path\to\your_app.apk
```

Physical Device:

```
adb – d install path/to/your_app.apk
```

Screenshots





Great Western Rail staff accused of mocking boy with autism

UK news

Parent felt 'humiliated' during alleged incident at help desk at Paddington station

Sarah Marsh

13 minutes ago

Theresa May briefs cabinet on Brexit negotiations – politics live

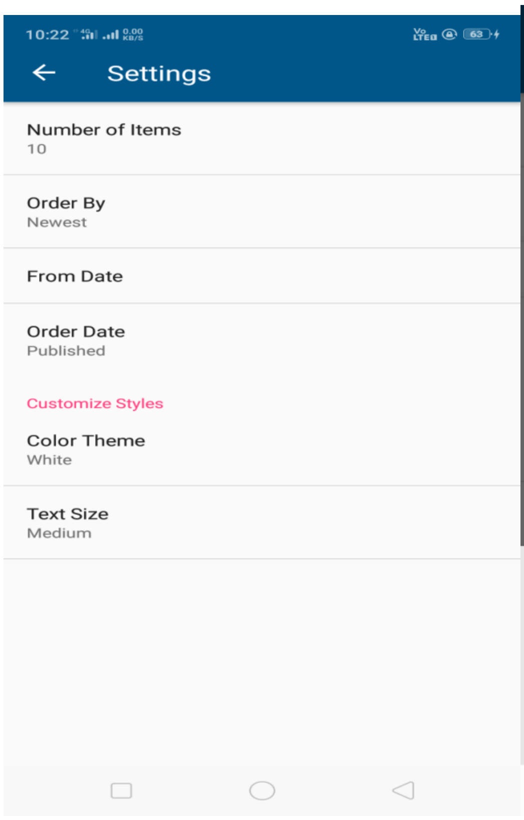
Politics

Rolling coverage of the day's political events including the latest developments in the Brexit saga

Alexandra Topping

20 minutes ago

Sri Lanka hammer England by 219 runs



Number of Items
10

Order By
Newest

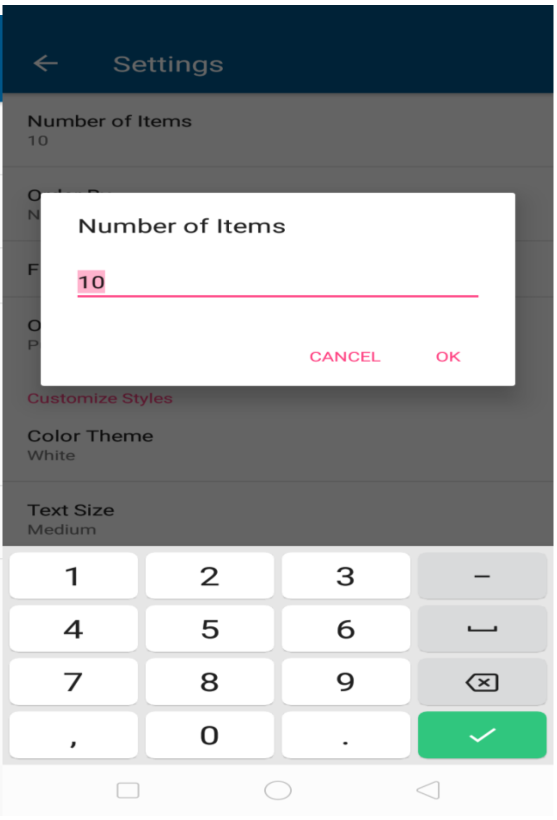
From Date

Order Date
Published

Customize Styles

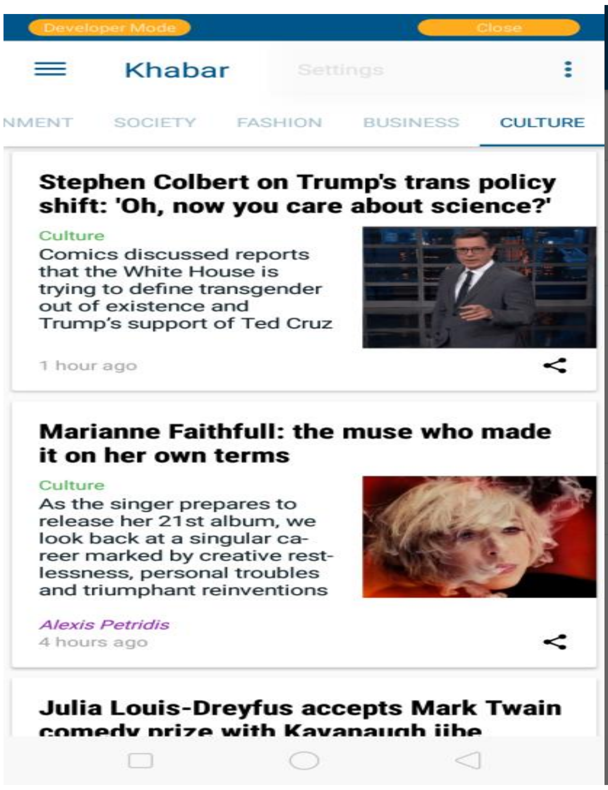
Color Theme
White

Text Size
Medium



Number of Items

10



Stephen Colbert on Trump's trans policy shift: 'Oh, now you care about science?'

Culture

Comics discussed reports that the White House is trying to define transgender out of existence and Trump's support of Ted Cruz

1 hour ago

Marianne Faithfull: the muse who made it on her own terms

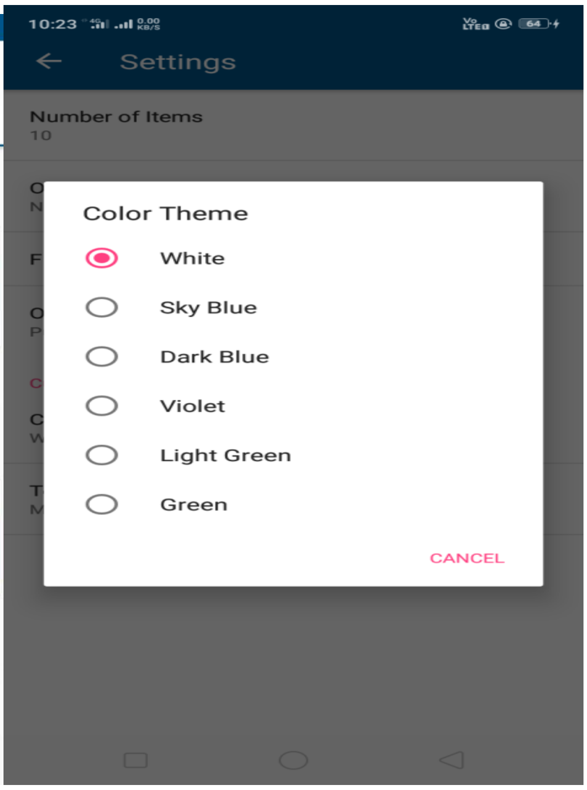
Culture

As the singer prepares to release her 21st album, we look back at a singular career marked by creative restlessness, personal troubles and triumphant reinventions

Alexis Petridis

4 hours ago

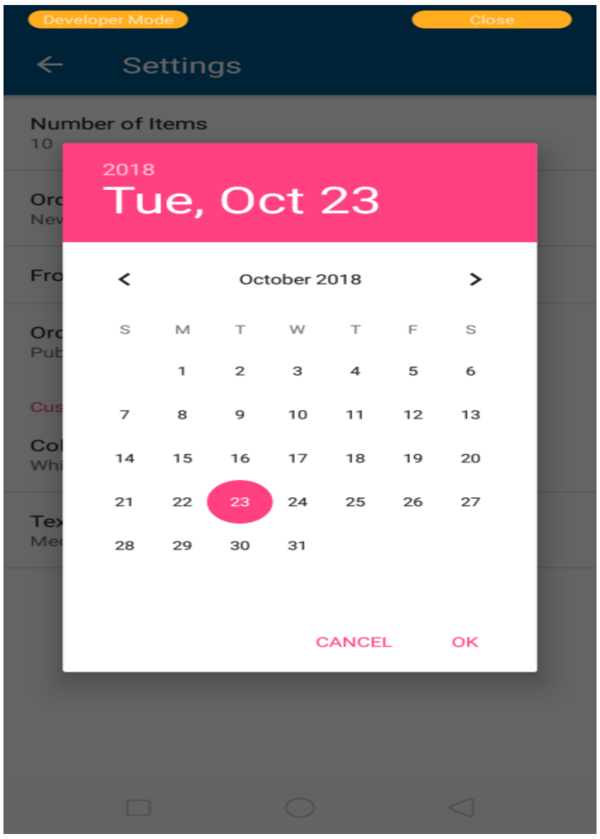
Julia Louis-Dreyfus accepts Mark Twain comedy prize with Kavanaugh jibe



Number of Items
10

Color Theme

- [✓ White]
- Sky Blue
- Dark Blue
- Violet
- Light Green
- Green

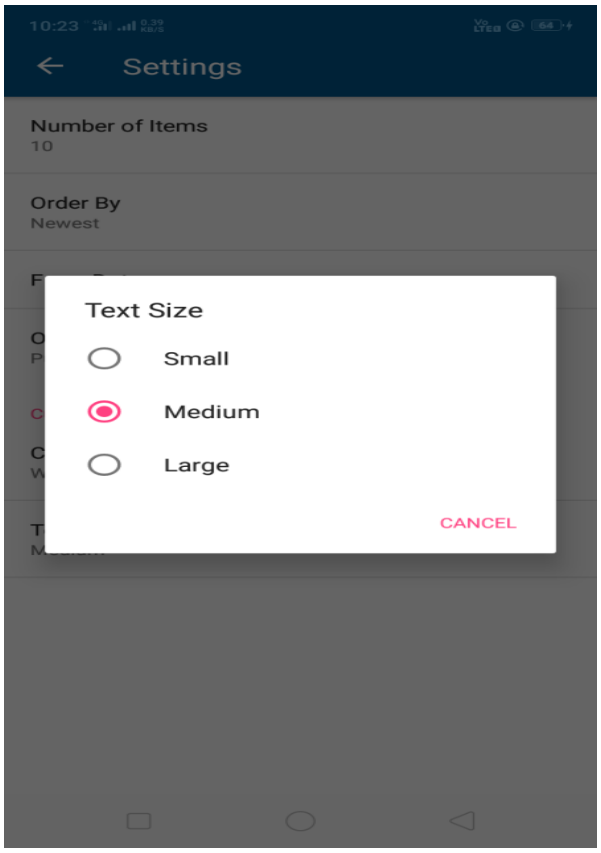


Number of Items
10

2018
Tue, Oct 23

October 2018

S	M	T	W	T	F	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	[✓ 23]	24	25	26	27
28	29	30	31			

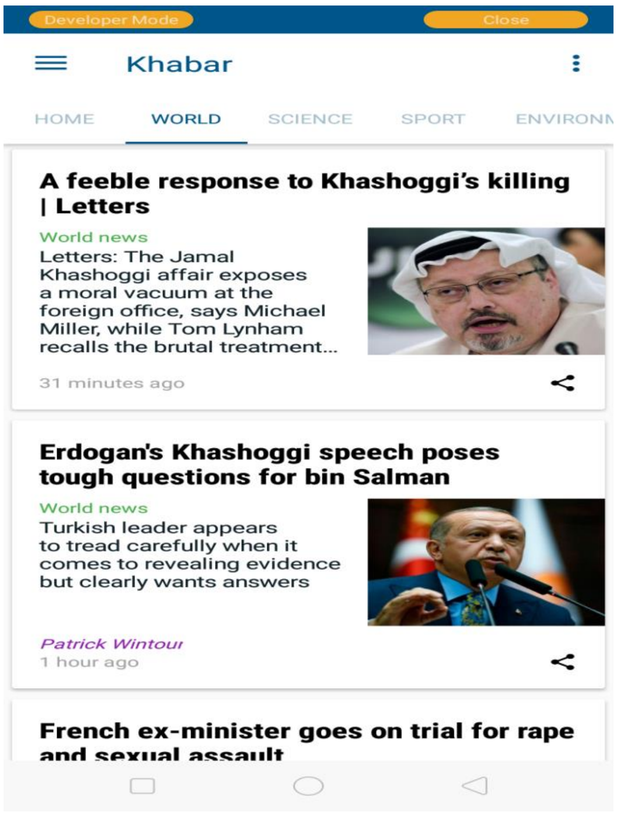


Number of Items
10

Order By
Newest

Text Size

- Small
- [✓ Medium]
- Large


Developer Mode Close

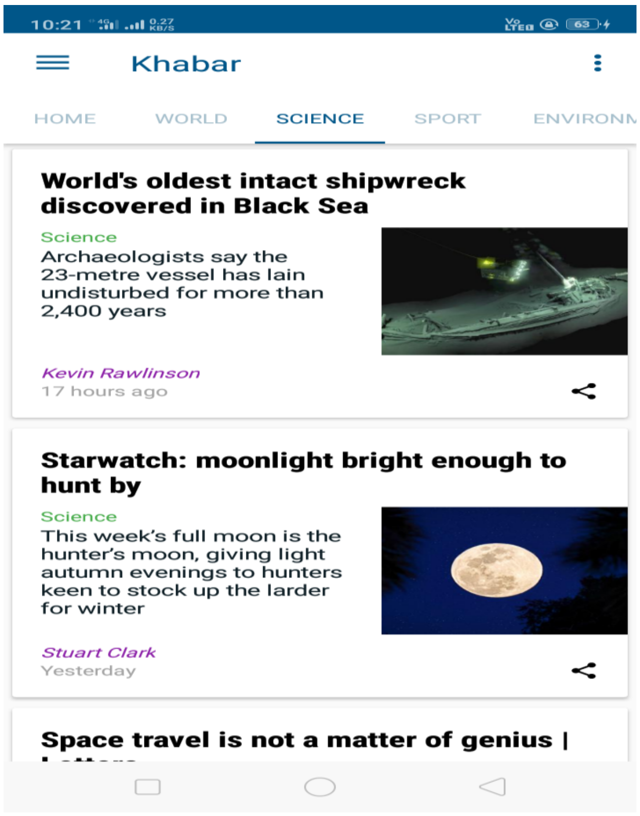
Khabar

HOME WORLD SCIENCE SPORT ENVIRONM

A feeble response to Khashoggi's killing | Letters
World news
Letters: The Jamal Khashoggi affair exposes a moral vacuum at the foreign office, says Michael Miller, while Tom Lynham recalls the brutal treatment...
31 minutes ago

Erdogan's Khashoggi speech poses tough questions for bin Salman
World news
Turkish leader appears to tread carefully when it comes to revealing evidence but clearly wants answers
Patrick Wintour
1 hour ago

French ex-minister goes on trial for rape and sexual assault


10:21 0.27 KB/S 63

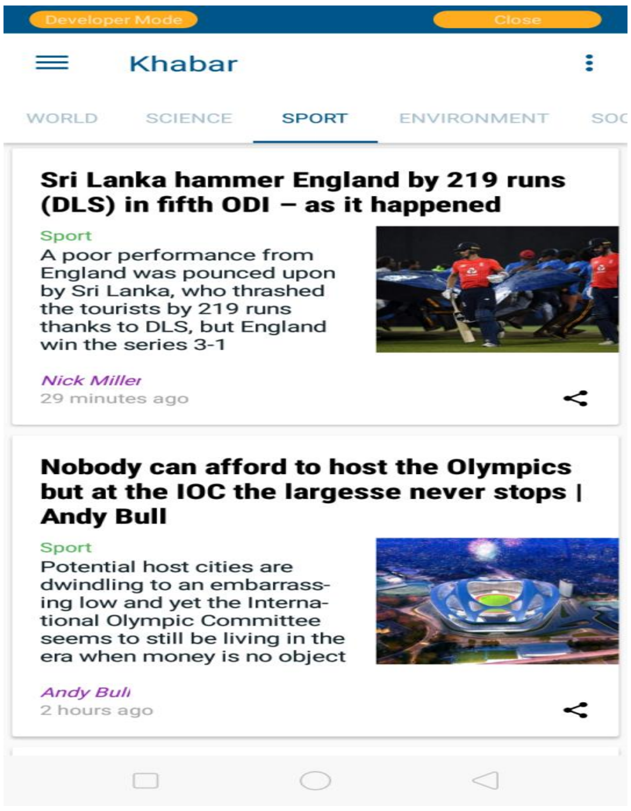
Khabar

HOME WORLD SCIENCE SPORT ENVIRONM

World's oldest intact shipwreck discovered in Black Sea
Science
Archaeologists say the 23-metre vessel has lain undisturbed for more than 2,400 years
Kevin Rawlinson
17 hours ago

Starwatch: moonlight bright enough to hunt by
Science
This week's full moon is the hunter's moon, giving light autumn evenings to hunters keen to stock up the larder for winter
Stuart Clark
Yesterday

Space travel is not a matter of genius |

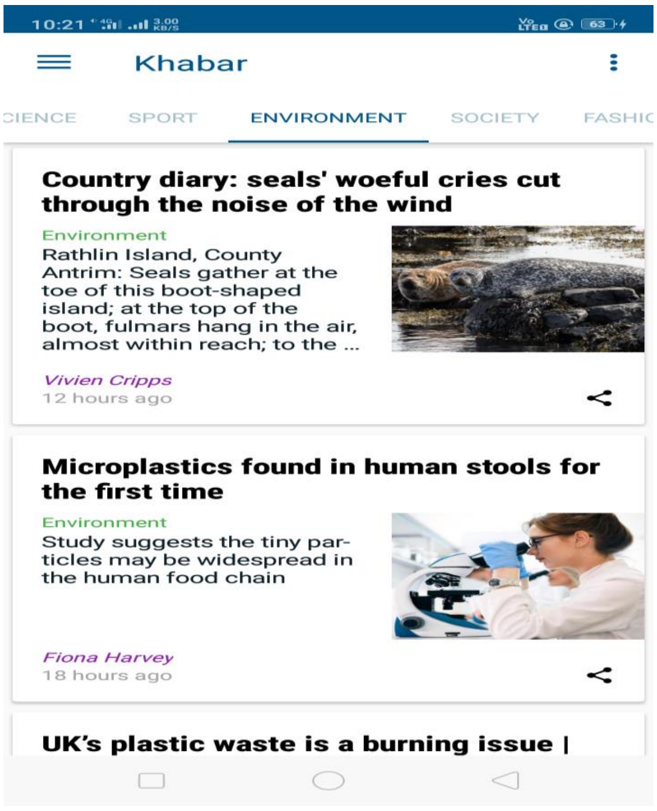

Developer Mode Close

Khabar

WORLD SCIENCE SPORT ENVIRONMENT SOC

Sri Lanka hammer England by 219 runs (DLS) in fifth ODI – as it happened
Sport
A poor performance from England was pounced upon by Sri Lanka, who thrashed the tourists by 219 runs thanks to DLS, but England win the series 3-1
Nick Miller
29 minutes ago

Nobody can afford to host the Olympics but at the IOC the largesse never stops | Andy Bull
Sport
Potential host cities are dwindling to an embarrassing low and yet the International Olympic Committee seems to still be living in the era when money is no object
Andy Bull
2 hours ago


10:21 3.00 KB/S 63

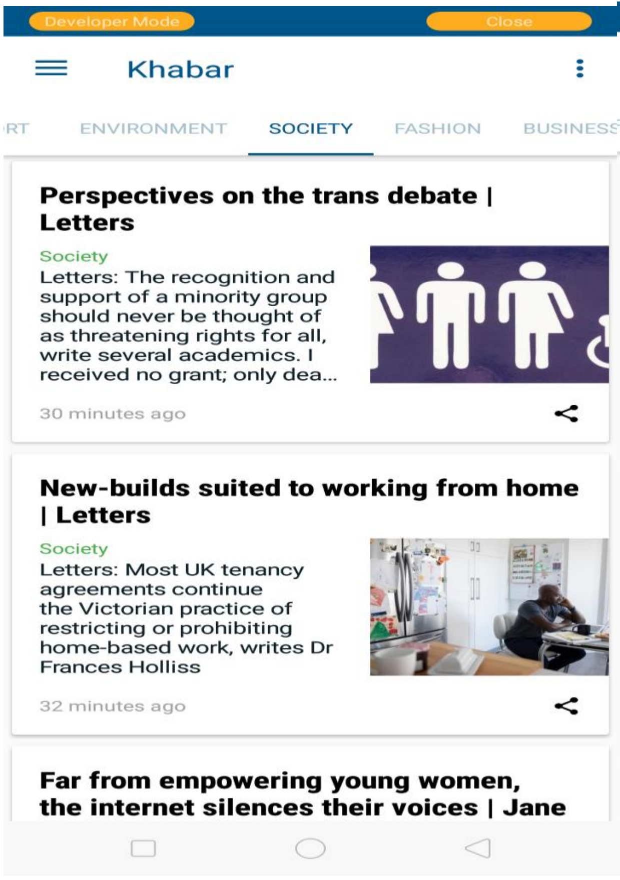
Khabar

CIENCE SPORT ENVIRONMENT SOCIETY FASHIC

Country diary: seals' woeful cries cut through the noise of the wind
Environment
Rathlin Island, County Antrim: Seals gather at the toe of this boot-shaped island; at the top of the boot, fulmars hang in the air, almost within reach; to the ...
Vivien Cripps
12 hours ago

Microplastics found in human stools for the first time
Environment
Study suggests the tiny particles may be widespread in the human food chain
Fiona Harvey
18 hours ago

UK's plastic waste is a burning issue |


Developer Mode Close

Khabar

RT ENVIRONMENT SOCIETY FASHION BUSINESS

Perspectives on the trans debate | Letters

Society

Letters: The recognition and support of a minority group should never be thought of as threatening rights for all, write several academics. I received no grant; only dea...

30 minutes ago

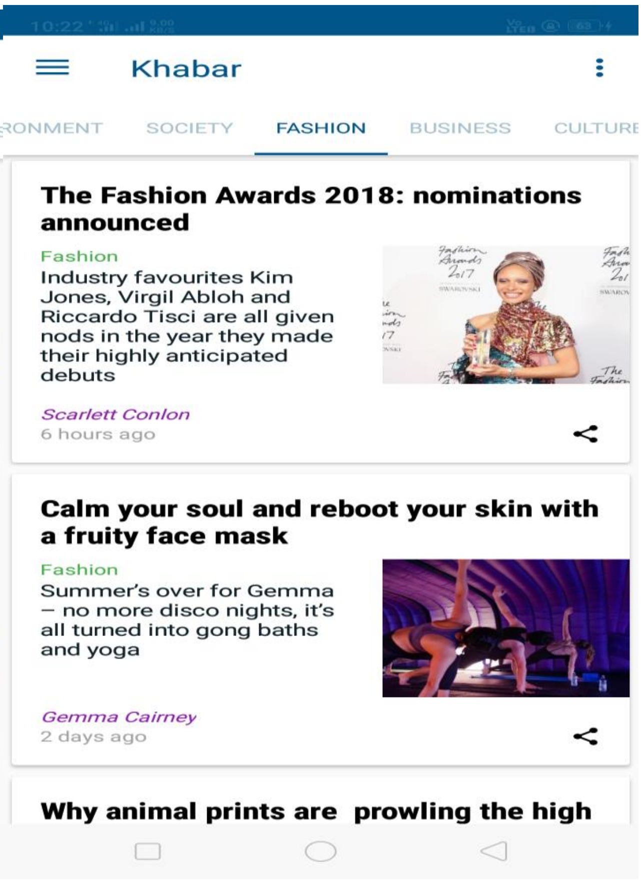
New-builds suited to working from home | Letters

Society

Letters: Most UK tenancy agreements continue the Victorian practice of restricting or prohibiting home-based work, writes Dr Frances Holliss

32 minutes ago

Far from empowering young women, the internet silences their voices | Jane


10:22

Khabar

RONMENT SOCIETY FASHION BUSINESS CULTURE

The Fashion Awards 2018: nominations announced

Fashion

Industry favourites Kim Jones, Virgil Abloh and Riccardo Tisci are all given nods in the year they made their highly anticipated debuts

Scarlett Conlon

6 hours ago

Calm your soul and reboot your skin with a fruity face mask

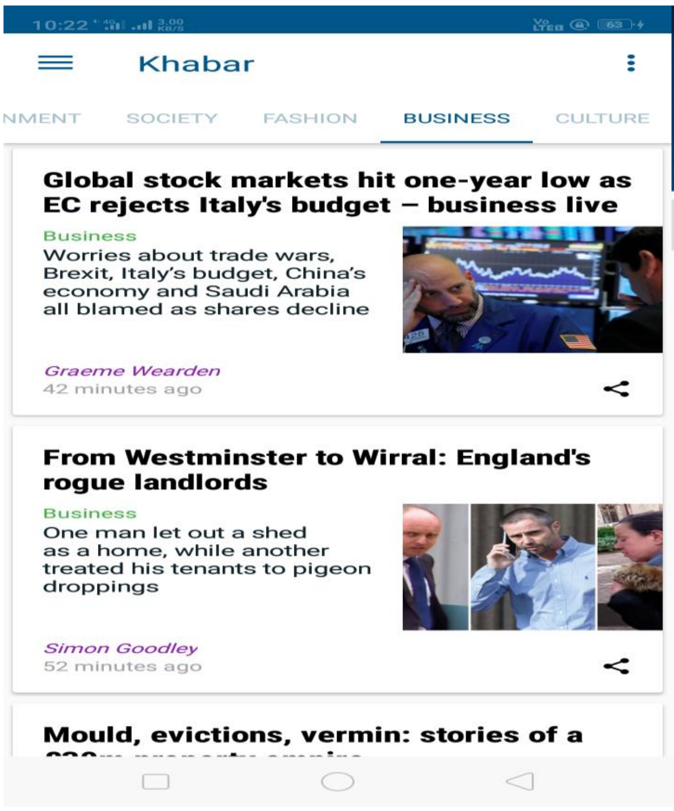
Fashion

Summer's over for Gemma – no more disco nights, it's all turned into gong baths and yoga

Gemma Cairney

2 days ago

Why animal prints are prowling the high


10:22

Khabar

NMENT SOCIETY FASHION BUSINESS CULTURE

Global stock markets hit one-year low as EC rejects Italy's budget – business live

Business

Worries about trade wars, Brexit, Italy's budget, China's economy and Saudi Arabia all blamed as shares decline

Graeme Wearden

42 minutes ago

From Westminster to Wirral: England's rogue landlords

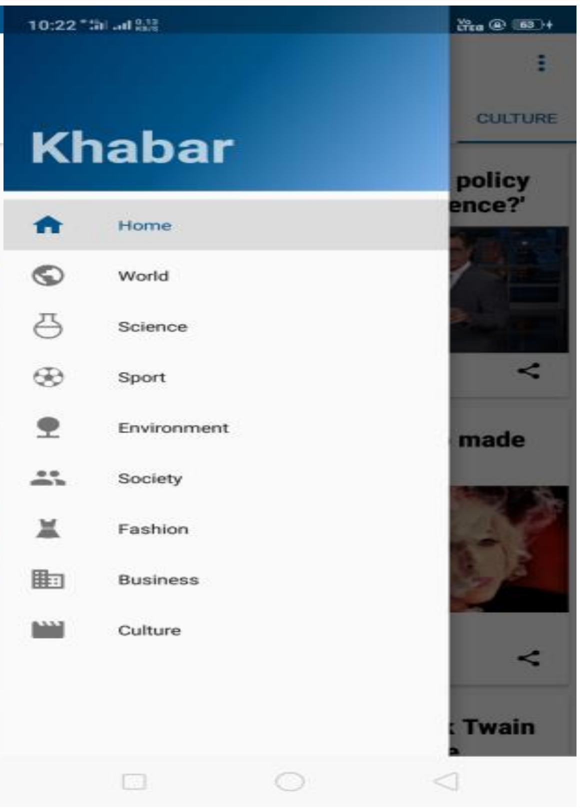
Business

One man let out a shed as a home, while another treated his tenants to pigeon droppings

Simon Goodley

52 minutes ago

Mould, evictions, vermin: stories of a


10:22

Khabar

Home

World

Science

Sport

Environment

Society

Fashion

Business

Culture

CULTURE

policy

ence?'

made

Twain

Advantages and disadvantages

Advantages

1. **To Access the news and articles in digital format.**
2. **More features, more fun!**
3. **A more convenient user experience.**
4. **Offline reads & comments.**
5. **Wide range News.**

Disadvantages

1. **Currently User is only able to access only one news Firm that can be upgraded in future this was due to limitation of time.**
2. **Splash screen not available currently.**
3. **Battery draining is more.**
4. **Database not used.**

Conclusion

We would like to thank the supervisor of our project, Professor SN.Wandre who gives us valuable advices for our project and remainders to present well. Professor SN.Wandre arranges a meeting with us every week for 1 hour so as to keep our progress up. We are not sure whether the final product can be achieved perfectly, but we feel confident to overcome the obstacles with the help of Professor SN.Wandre. Besides, we would like to thank Dr.S.D. Babar HOD of Compter Department. He provides us with facilities and technical support when we are setting up the Project and gives us many suggestions on how to make the application more feasible.

Thank You