A PROJECT REPORT

ON

ANDROID APPLICATION FOR NEWS APPLICATION

(NEWS INDIA)

SUBMITTED BY

Mr. SALIL RAVINDRA BHAGAT

IN PARTIAL FULFILLMENT FOR THE AWARD OF THE DEGREE

OF

BACHELOR OF SCIENCE

IN

COMPUTER SCIENCE

UNDER THE GUIDANCE OF

PROF. VIPUL SALUJA

DEPARTMENT OF COMPUTER SCIENCE

R. D. & S.H. National College of Arts and Commerce

& W.A. Science College , Bandra, Mumbai – 400050.

(Sem-VI)

(2019– 2020)

R.D. & S.H. NATIONAL COLLEGE & S. W.A. SCIENCE COLLEGE,

Bandra, Mumbai – 400050.

Department of Computer Science

PROJECT CERTIFICATE

This is to certify that **Mr. SALIL RAVINDRA BHAGAT** of **T.Y.B.Sc. (Sem VI)** class has satisfactorily completed the Project entitled "**ANDROID APPLICATION ABOUT NEWS APPLICATION (NEWS INDIA)"**, to be submitted in the partial fulfillment for the award of **Bachelor of Science** in **Computer Science (Semester VI)** during the academic year **2019– 2020**.

It is further certified that the project is an original work and it has not formed the basis for the award of any degree, associateship, fellowship or any other similar titles.

Date of Submission:

Internal Project Guide **Co-ordinator,**

Department of Computer Science

Signature of External Examiner

DECLARATION

I, Mr. SALIL BHAGAT, hereby declare that the project entitled "**ANDROID APPLICATION ABOUT NEWS APPLICATION (NEWS INDIA)"** submitted in the partial fulfilment for the award of **Bachelor of Science** in **Computer Science** during the academics year 2019-2020 is my original work and the project has not formed the basis for the award of any degree, associateship, fellowship or any other similar titles

Signature of the student:

Place:

Date:

ACKNOWLEDGEMENT

This project could not have been completed without **Prof. Vipul Saluja** who not only served as my project guide but also encouraged and challenged me throughout my academic program.

He guided me through the entire project process, never accepting less than my best efforts. I thank them all.

I would also like to thank my family and my fellow coulleagues **,** who helped and supported me throughout the way.

And a great contribution of internet, specifically specifiedcoding, youtube, google, android hub and so on.

INDEX

CHP 1-PRELIMINARY INVESTIGATION

1. Synopsis

2. Organization Overview

3. Working of the current system

4. Limitations of the current system

5. The Proposed System

6. The Advantages of the Proposed System

7. Tools and Technologies to be used

8. Feasibility Study

1. PRELIMINARY INVESTIGATION

1.1: SYNOPSIS

SYNOPSIS

NEWS APPLICATION (NEWS INDIA)

INTRODUCTION:-

- **THIS APPLICATION IS AN ANDROID APPLICATION WHICH IS USED FOR OBTAINING THE LATEST NEWS STORIES**
- **THIS ANDROID APPLICATION CALLED AS "NEWS INDIA"**
- **IT INCLUDES THE LIST OF TRENDING HEADLINES, NEWS REPORT AND LIVE BROADCASTING**
- **THE INFORMATION OF USER IS CALIBRATED WITH "FIREBASE"**
- **DAILY NEWS UPDATE IS AVAILABLE USING THIS APPLICATION**

ADVANTAGES:-

- **USERS CAN ACCESS THE NEWS ANYWHERE WITH THE HELP OF THIS APP**
- **IT CAN PREVENT THE LOSS OF KNOWLEDGE**
- **IT CAN REDUCE THE DEPENDENCY OF PAPERS**

MODULES:-

- **NEWS**
- **LIVE CHANNELS**
- **LOGIN**

SOFTWARE REQUIREMENTS:-

- **ANDROID STUDIO**
- **FIREBASE**

1.2 Organizational Overview:

Few reasons to use "NEWS INDIA"

1) **Quality –** The news present in this applications are from the strong and ideal news sources around the world
2) **Customer Satisfaction –** The customer gets the best quality of news present on internet , thus giving importance to the customer satisfaction.
3) **Simplicity**- THE NEWS INDIA is an application which simply depend upon the simplicity of the UI

1.3 Working of the current system :

1) The customer needs to visit the particular news websites
2) That particular website gives information of particular topic.
3) The particular information may not be present on that particular sources.
4) Thus it difficult to get information from different types of sources
5) So to overcome all these issues there was a need to come up with an Android application that can provide information from various sources.

1.4 Limitations of the current system:

1) The user has to scavange across the internet to get the desired news from various sources.
2) The user may get the fake news form the particular news source unable to realise that
3) The particular source website may not be secure and could be used for malicious intentions
4) Some sources may also contain large amounts of popup ads

1.5 The Proposed System:

1) We have created an application that is not only more interactive but offers quality news for free
2) The application focuses more on Information required by customers rather than unnecessary and annoying advertisements.
3) The application makes the whole process acquiring news very easy.
4) Application had a dedicated feedback section to interact with the developers

1.6 Advantages of the Proposed system:

1) Get latest news from the various sources in one application
2) The sources are verified personally by the developer
3) Complete Cooperation for customers by 24/7 by the contact us function inbuilt in the application with the email.
4) The user can directly interact with the developer using this function.
5) Detection of errors in data entry is easy.
6) As the data entered is not manually it can be easily edited and modified.
7) Less chances of errors.
8) Security and authentication features for protecting sensitive information.

1.7 Tools and Technologies to be used:

1) ANDROID STUDIO 3.5

2) FIREBASE

Android studio:-

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development.It is available for download on Windows, macOS and Linux based operating systems.

The following features are provided in the current stable version:

1) Gradle-based build support

2) Android-specific refactoring and quick fixes

3) Lint tools to catch performance, usability, version compatibility and other problems

4) ProGuard integration and app-signing capabilities

5) Template-based wizards to create common Android designs and components

6) A rich layout editor that allows users to drag-and-drop UI components, option to preview layouts on multiple screen configurations

7) Support for building Android Wear apps

8) Built-in support for Google Cloud Platform, enabling integration with Firebase Cloud Messaging (Earlier 'Google Cloud Messaging') and Google App Engine

9) Android Virtual Device (Emulator) to run and debug apps in the Android studio.

Firebase

Firebase is a mobile and web application development platform developed by Firebase, Inc. in 2011, then acquired by Google in 2014. As of October 2018, the Firebase platform has 18 products which are used by 1.5 million apps.

Firebase evolved from Envolve, a prior startup founded by James Tamplin and Andrew Lee in 2011. Envolve provided developers an API that enables the integration of online chat functionality into their websites. After releasing the chat service, Tamplin and Lee found that it was being used to pass application data that weren't chat messages. Developers were using Envolve to sync application data such as game state in real time across their users. Tamplin and Lee decided to separate the chat system and the real-time architecture that powered it. They founded Firebase as a separate company in September 2011 and it launched to the public in April 2012.

Firebase's first product was the Firebase Real-time Database, an API that synchronizes application data across iOS, Android, and Web devices, and stores it on Firebase's cloud. The product assists software developers in building real-time, collaborative applications.

1.8 Feasibility study:

Introduction:

- A feasibility study is an analysis of how successfully a project can be completed.
- It is the initial design stage of any project, which brings together the elements of knowledge.

Technical feasibility:-

Since THE NEWS INDIA is an android application which developed under android studio and Firebase realtime databases for database connectivity. For feedback data storage access data we used Firebase realtime database and for news ,newsapi.org was used for all the latest news reports.

Hence the decision was taken that Firebase database will store the remote data access and newsapi.org will provide the latest news.

Technical feasibility is concerned with specifying the equipments and the software to satisfy the user requirements.

Operational feasibility:-

The Application has been developed for Android OS and will work on all the Android Device which has Android version 4.4 and above. The database which is used for this application is been hosted using Google Firebase and news are provided by newsapi.org

The overall response of the system will also increase as there will be more number of users affiliated with the system in the near future.

Economic feasibility:-

The Application is been developed for the Third Year College Project.

Economic analysis is the most frequently used method for evaluating the effectiveness of a new system.

The proposed system can be developed at a minimum cost and resource.

The system can assure a good beneficial cost to the organisation.

The savings that would arise from the beneficial cost of the system can be used to improve the system's performance in future.

CHP 2-SYSTEM ANALYSIS

2.1 Event Table

2.2 Entity Relationship Diagram

2.3 Class Diagram

2.4 Object Diagram

2.5 Use Case Diagram

2.6 Activity Diagram

2.7 State Chart Diagram

2.8 Sequence Diagram

2.8 Collaboration Diagram

2.1 Event Table

Events are objects or messages used when a software components wants to notify a state change to other components.

An Event model is a software architecture (a set of classes and interfaces)that determines how components occur.

On the event source side:-

- create and describe events
- trigger (or fire)events
- distribute events to interested components

On the event listener side:-

- subscribe to event sources
- react to events when received
- remove the subscription to event sources when desired

Terminology often used refers to:-

- Event Source or Provider :-the sender of events
- Event :-the object sent
- Event Listener or Event Sink or Consumer :-the receiver of events

FOR ADMIN :

EVENT	SOURCE	TRIGGER	ACTIVITY	DESTINATION
Views Feedback	Registered User	View all the feedback	Feedbacks are view by admin	Admin
Deletes Feedback	Registered User	Deletion of feedback	Feedbacks are Deleted by admin	Admin

FOR REGISTERED USER :

EVENT	SOURCE	TRIGGER	ACTIVITY	DESTINATION
Login	Registered User	Login Page	Display Login Page	Registered User
Visits the Homepage	Registered User	Homepage	Display Homepage	Registered User

View news	Registered User	Viewing of news	Display news	Registered User
View news in Details	Registered User	Details of Products viewed	Display news in Details	Registered User
User Gives Feedback	Registered User	Feedback Page	Feedback Submitted	Admin
Logs Out	Registered User	Log Out	Homepage	Registered User

2.2 Entity Relationship Diagram

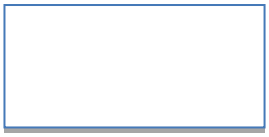
In software engineering, an entity relationship model(ER model)is a data model for describing the data or information aspects of a business domain or its process requirements, in an abstract way that leads itself to ultimately being implemented in a database such as a relational database. The maine components of ER models are entities (things) and the relationships that can exist among them.

An entity- relationship model is a systematic way of describing and defining a business process.The process is modelled as components (entites) that are linked wid each other by relationships that express the dependencies and requirements between them,such as: one building may be divided into zero or more apartments,but one apartment can only be located in one building.Entites may have various properties (attributes) that characterize them.Diagram created to represent these entities, attributes and relationships graphically are called entity -relationship diagrams.

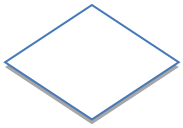
An ER model is typically implemented as a database. In the case of a relational database, which stores data in tables, every row of each table represents one instance of an entity .Some data fields in these tables point to indexes in other tables; such pointers represent the relationship.

Limitations:-

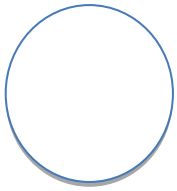
- ER models assume information content that can readily be represented in a relational database. They describe only a relational structure for this information.
- They are inadequate for systems in which the information cannot readily be represented in relational form, such as with semi-structured data



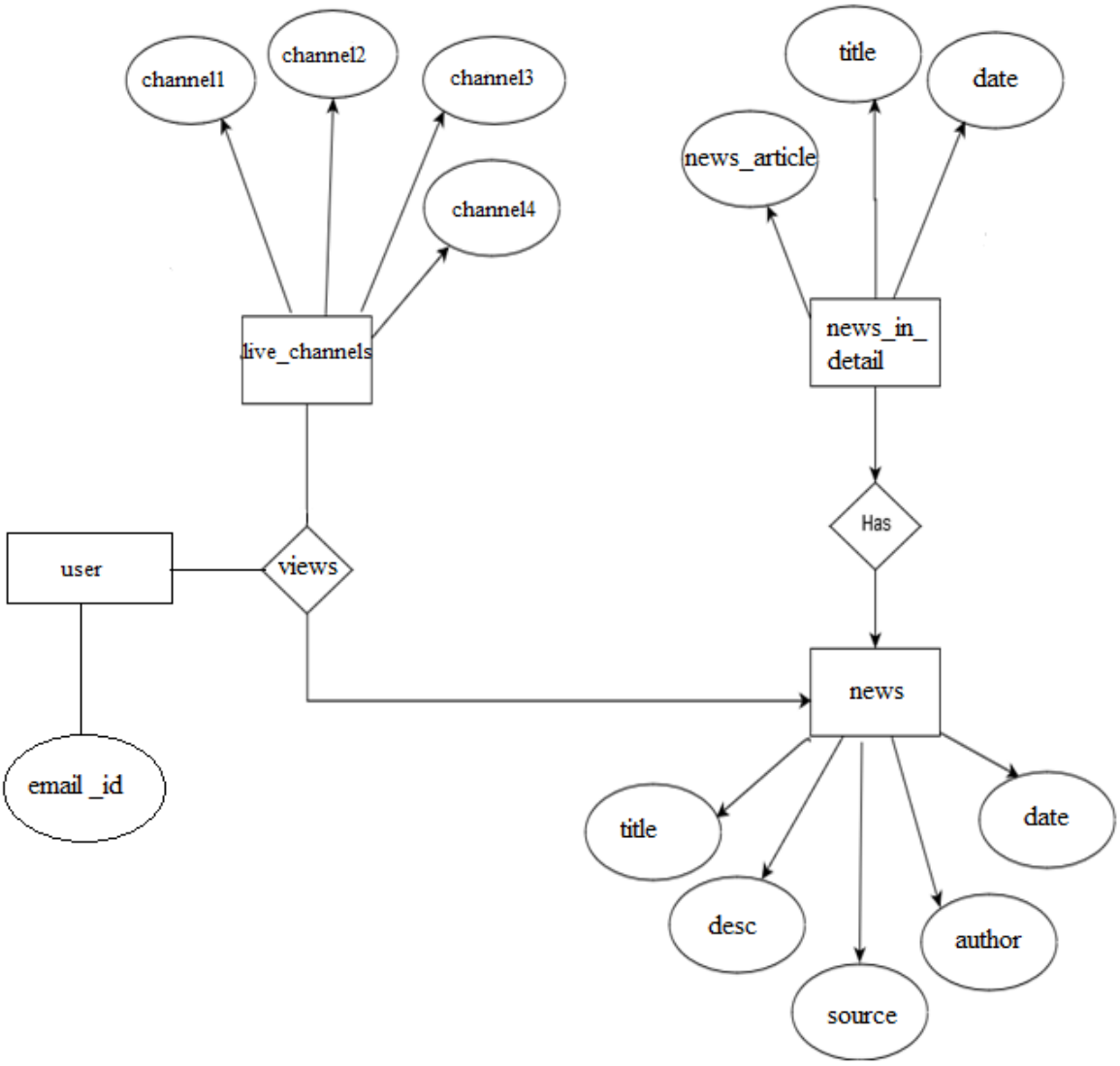
Entity



Relationship



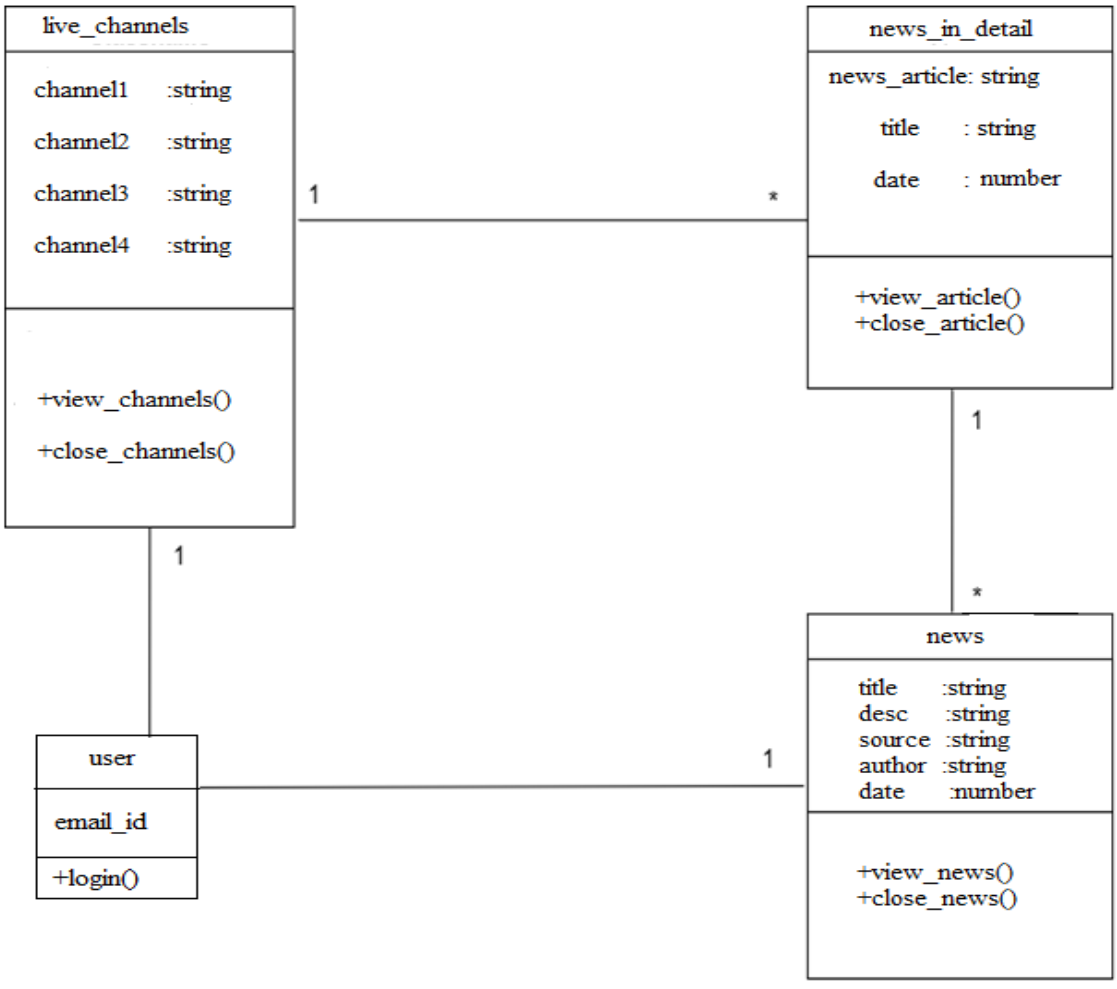
Attribute



2.3 Class Diagram

The class diagram is a static diagram.It represents the static view of an application.Class diagram is not only used for visualising, describing and documenting different aspects of a system but also for constructing executable code of the software application.The class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modelling of object oriented systems because they are the only UML diagrams which can be mapped directly with object oriented languages.The class diagram shows a collection of classes, interfaces, associations, collaborations and constraints .It is also known as a structural diagram.

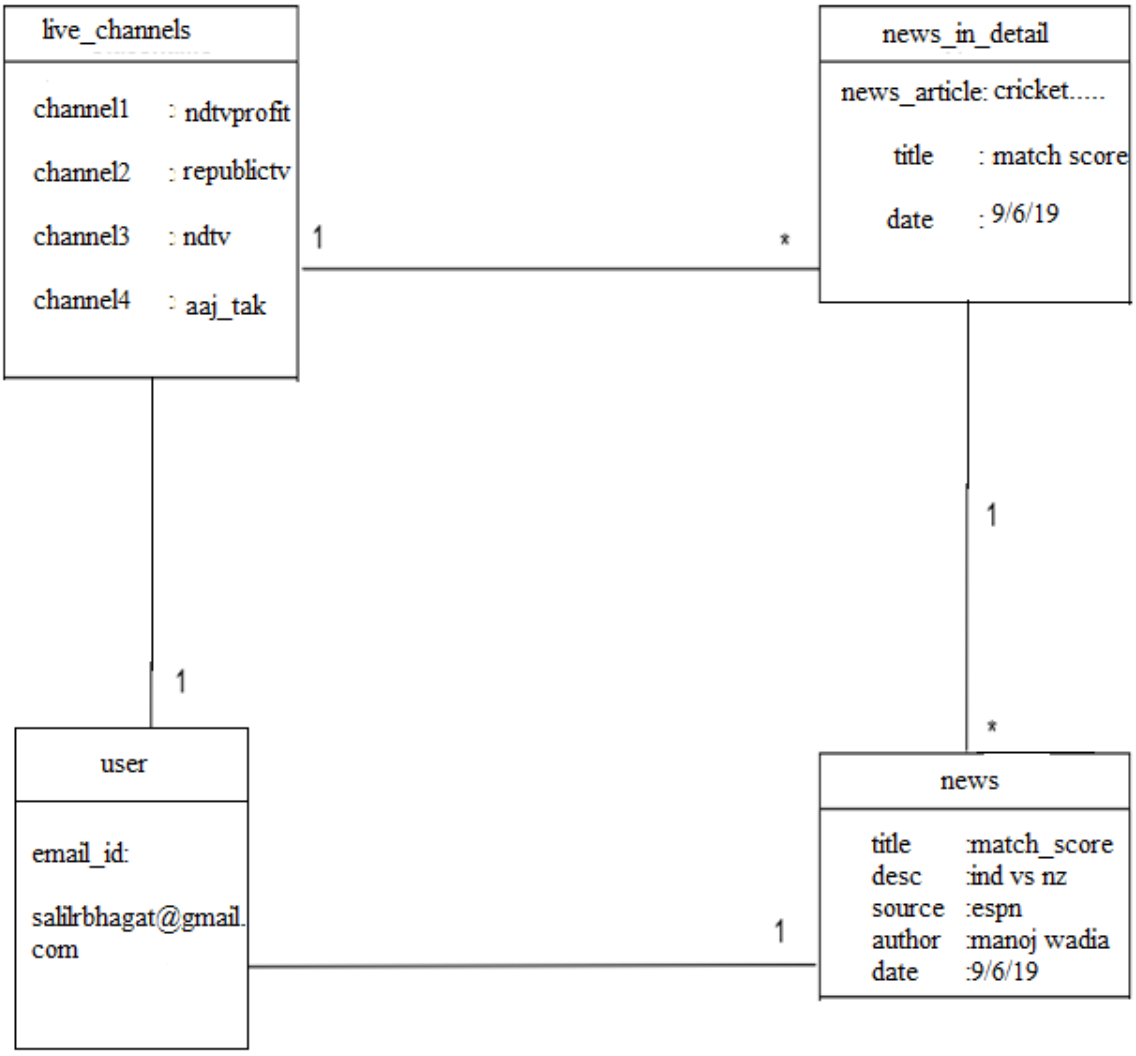
Purpose:-The purpose of the class diagram is to model the static view of an application. The class diagrams are the only diagrams which can be directly mapped with object oriented languages and thus widely used at the time of construction.The UML diagrams like activity diagram, sequence diagram can only give the sequence flow of the application but class diagram is a bit different. So it is the most popular UML diagram in the coder community.



2.4 Object Diagram

Object diagrams are derived from class diagrams so object diagrams are dependent upon class diagrams.Object diagrams represent an instance of class diagram.The basic concepts are similar for class diagrams and object diagrams.Object diagrams also represent the static view of a system but this static view is a snapshot of the system at a particular moment.Object diagrams are used to render a set of objects and their relationships as an instance.

Purpose:-The purpose of a diagram should be understood clearly to implement it practically.The purpose of object diagrams are similar to class diagrams.The difference is that a class diagram represents an abstract model consisting of classes and their relationships.But an object diagram represents an instance at a particular moment which is concrete in nature.It means the object diagram is more close to the actual system behaviour.The purpose is to capture the static view of a system at a particular moment.

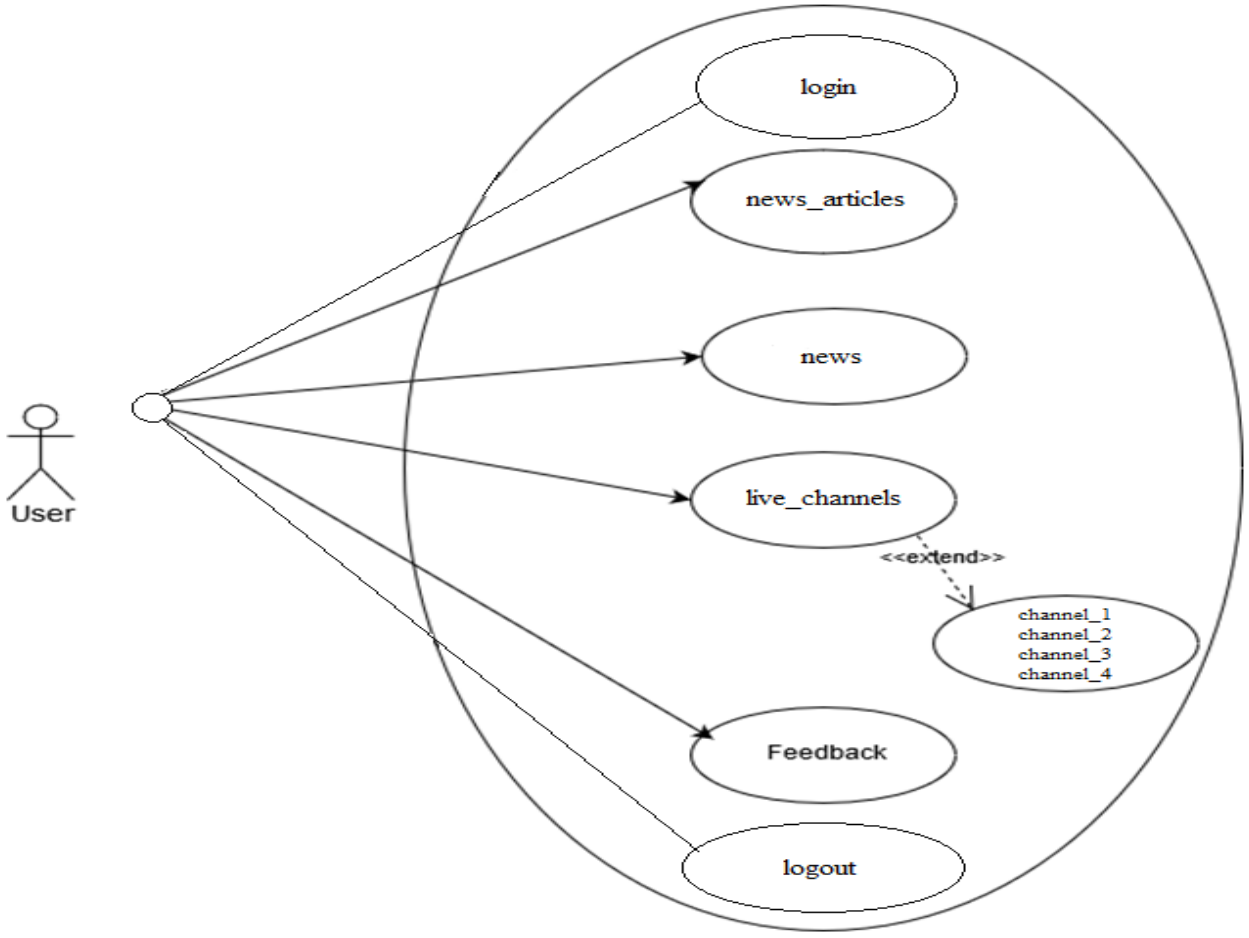


2.5 Use Case Diagram

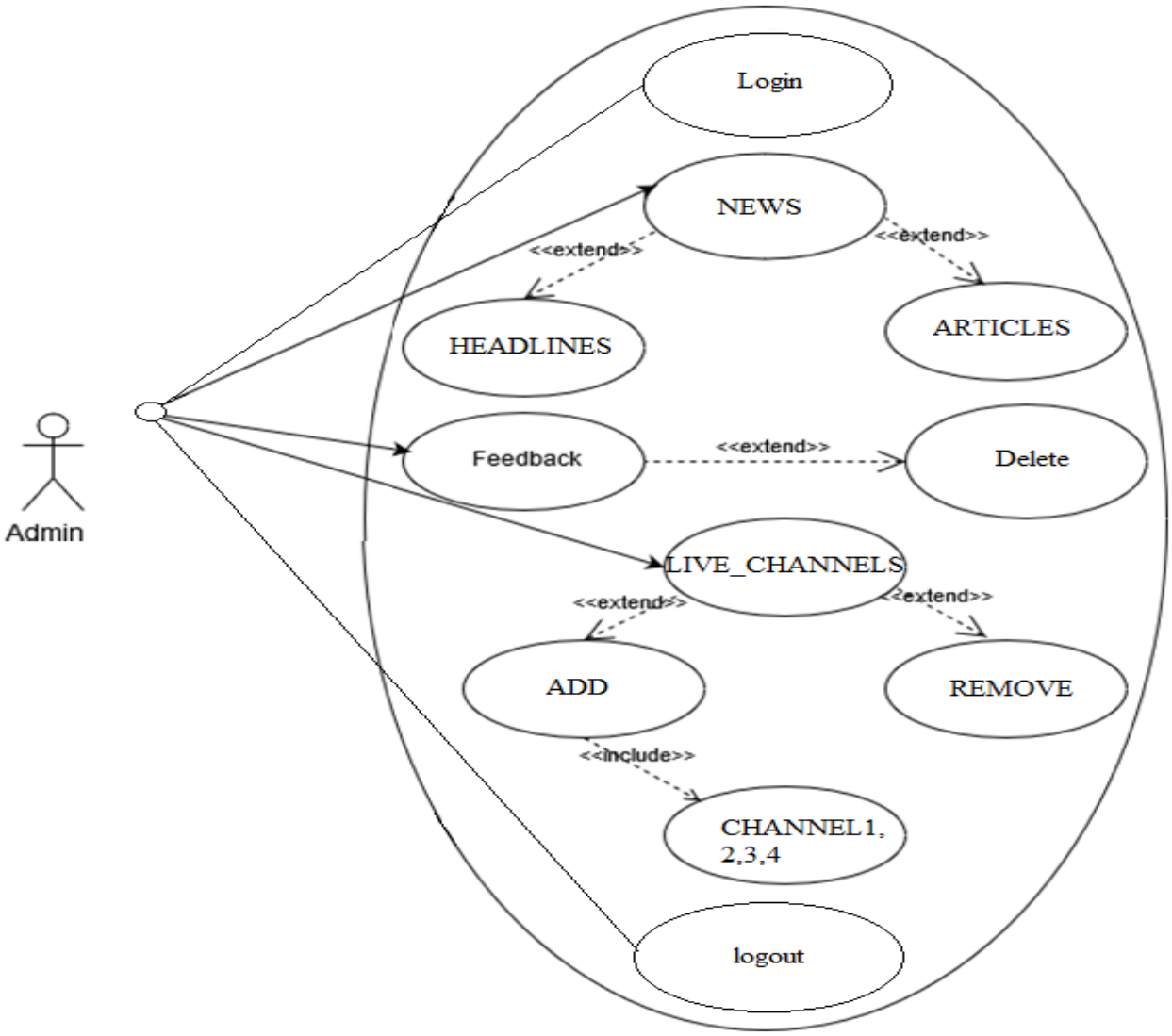
To model a system the most important aspect is to capture the dynamic behaviour.To clarify a bit in details, dynamic behaviour means the behaviour of the system when it is running/operating.So only static behaviour is not sufficient to model a system rather dynamic behaviour is more important than static behaviour.Now as we hve to discuss that the use case diagram is dynamic in nature there should be some internal or external factors for making the interaction.These internal and external agents are known as actors.So use case diagrams consists of actors,use cases and their relationships.The diagram is used to model the system/subsystem of an application.

Purpose:-The purpose of use case diagram is to capture the Dyna aspect of a system.But this definition is too generic to describe the purpose.Use case diagrams are used to gather the requirements of a system including internal and external influences.These requirements are mostly design requirements.So when a system is analysed to gather it's functionalities use cases are prepared and actors are identified.Now when the initial task is complete use case diagrams are modelled to present the outside view.

For User:-



For admin:



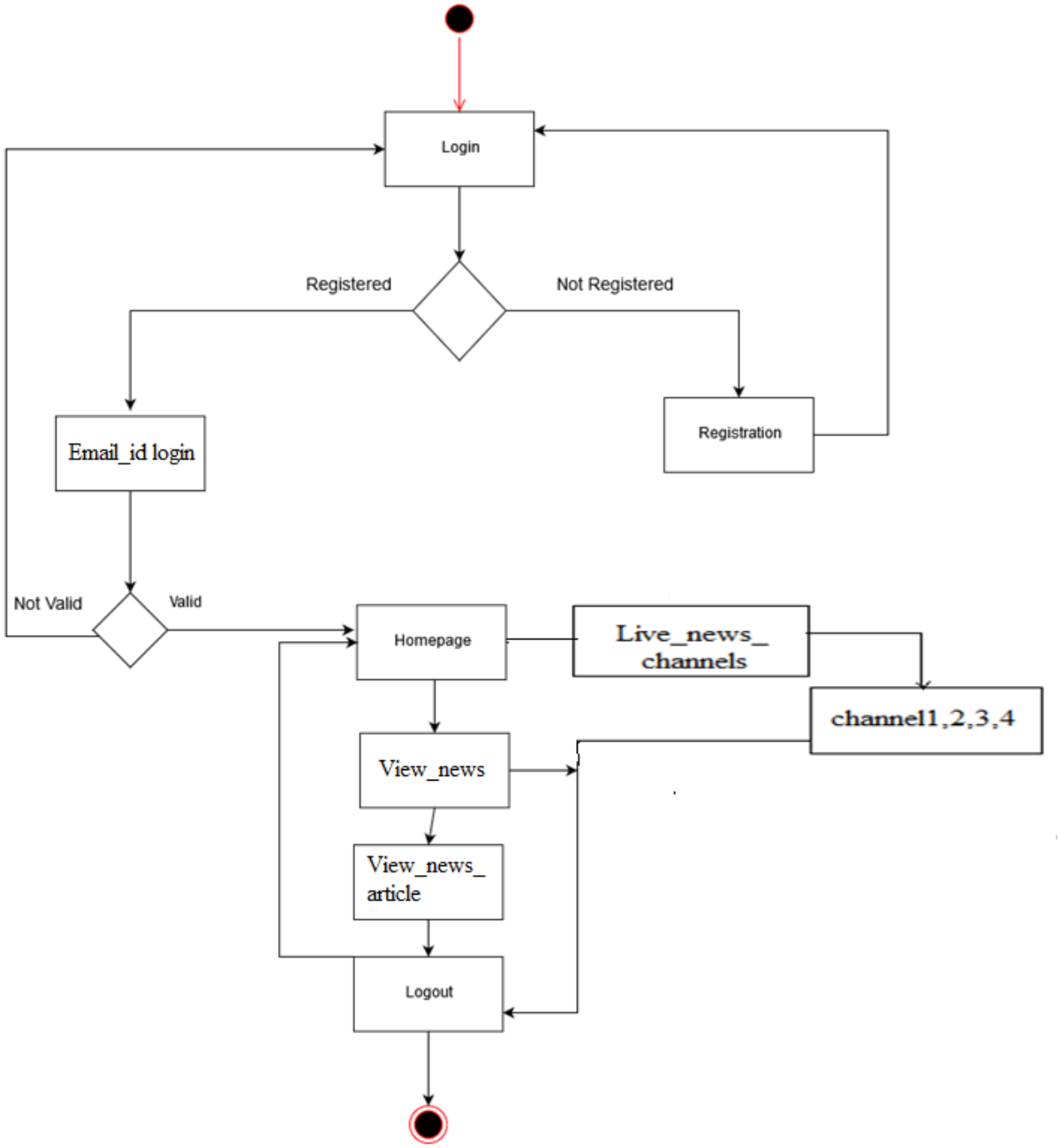
2.6 Activity Diagram

Activity diagram is another important diagram in UML to describe dynamic aspects of the system.Activity diagram is basically a flow chart to represent the flow from one activity to another activity.The activity can be described as an operation of the system.So the control flow is drawn from one operation to another.This flow can be sequential,branched or concurrent.Activity diagram deals with all type of flow control by using different elements like fork,join etc.

Purpose: It captures the dynamic behavior of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another.Activity is a particular operation of the system. Activity diagrams are not only used for visualizing dynamic nature of a system but they are also used to used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in activity diagram is the message part.It does not show any message flow from one

activity to another. Activity diagram is some time considered as the flow chart. Although the diagram looks like a flow chart but it is not. It shows different flow like parallel, branched, concurrent and single.

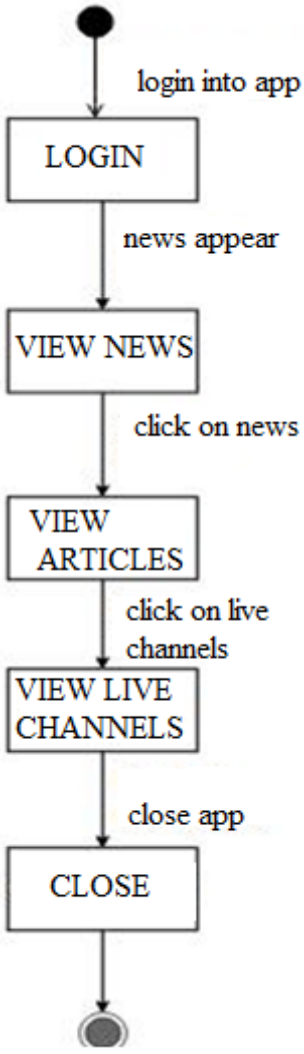
For User:



2.7 State Chart Diagram

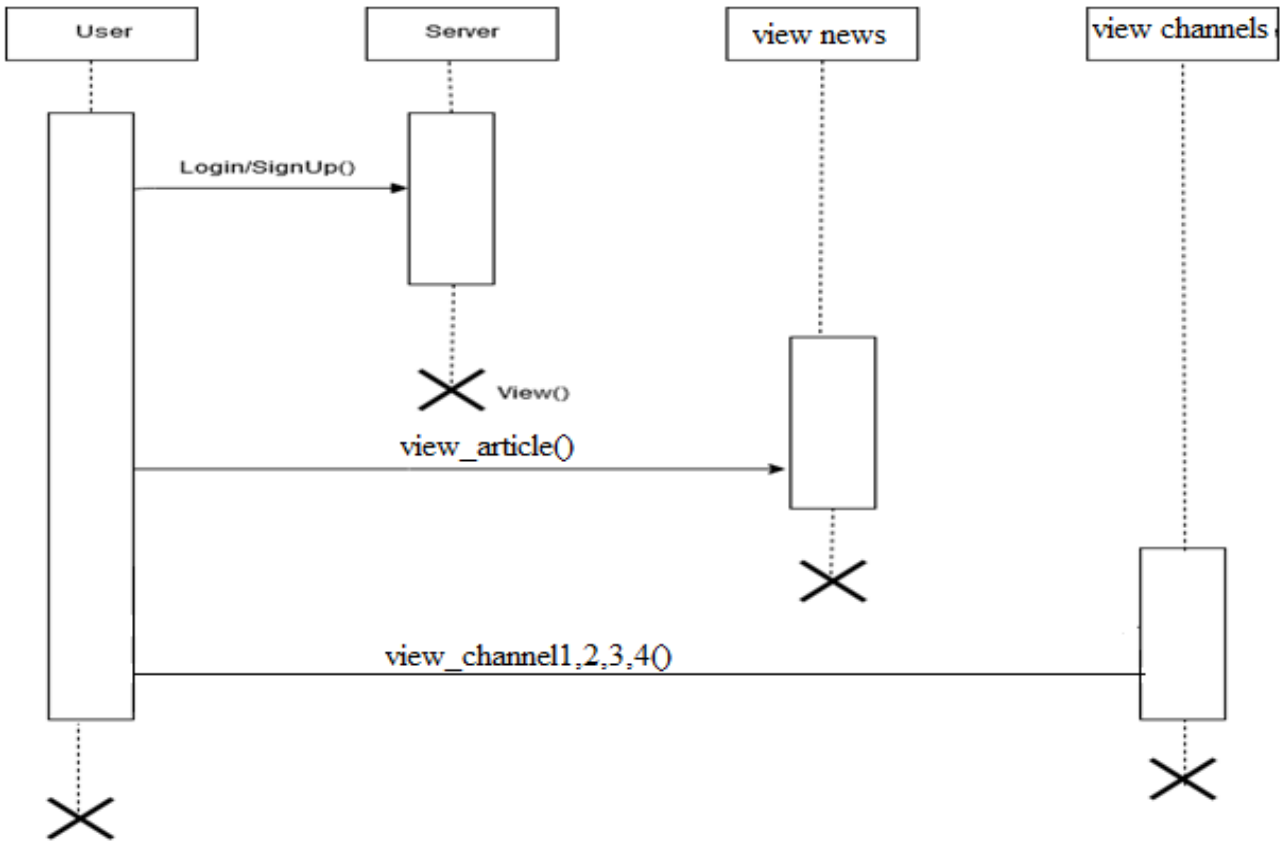
The name of the diagram itself clarifies the purpose of the diagram and other details. It describes different states of a component in a system. The states are specific to a component/object of a system.A State Chart Diagram describes a state machine. Now to clarify it state machine can be defined as a machine which defines different states of an object and these states are controlled by external or internal events.As State Chart Diagram defines states it is used to model lifetime of an object.

Purpose:-State Chart Diagram is one of the five UML diagrams used to model dynamic nature of a system.They define different states of an object during its lifetime. And these states are changed by events. So State Chart Diagrams are useful to model reactive systems. Reactive systems can be defined as a system that responds to external or internal events.State Chart Diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. So the most important purpose of State Chart Diagram is to model life time of an object from creation to termination.State Chart Diagram are also used for forward and reverse engineering of a system. But the main purpose is to model reactive system.



2.8 Sequence Diagram

A Sequence Diagram is an interaction diagram that shows how processes operate with one another and what is their order. It is a construct of a Messsage Sequence Chart. A Sequence Diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence Diagrams are sometimes called event diagrams or event scenarios.A Sequence Diagram shows parallel vertical lines(lifelines),different processes or objects that live simulatenously and as horizontal arrows,the messgaes exchanged between them,in order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.If the lifeline is that of an object,it demonstrates a role.Leaving the instance name blank can represent anonymous and unnamed instances.Messages,written with horizontal arrows with the message name written above them,display interaction.Solid arrow heads represent synchronous calls,open arrow heads represent asynchronous messages, and dashed lines represent reply messages.If a caller sends a synchronous message,it must wait until the message is done,such as invoking a subroutine. If a caller sends an asynchronous message,it can continue processing and doesn't have to wait for a response.Asynchronous calls are present in multithreaded applications and in message-oriented middleware.Activation boxes,or method-call boxes,ar opaque rectangles drawn on top of lifelines to represent that processes are being performed response to the message(Execution Specification in UML).

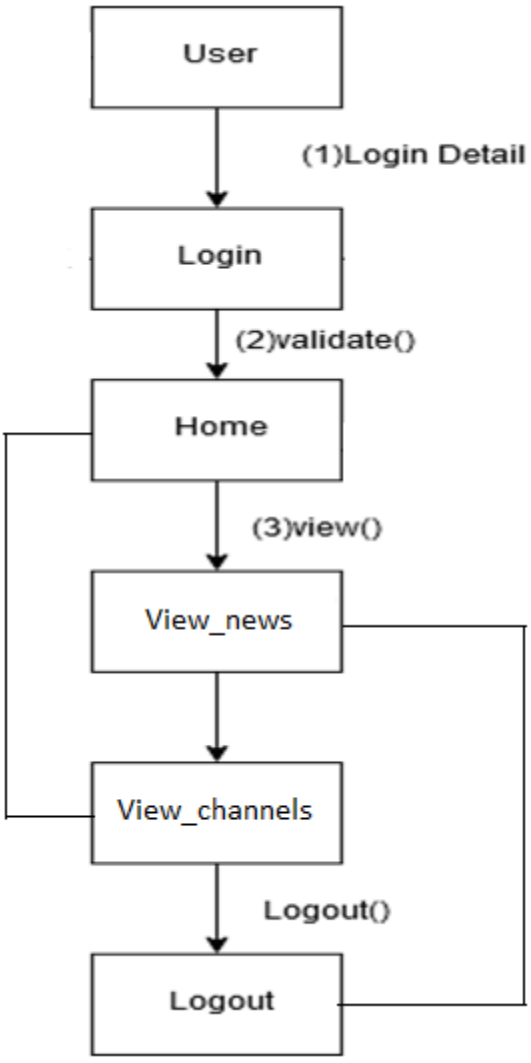


2.9 Collaboration Diagram

A collaboration diagram, also called a communication diagram or interaction diagram, is an illustration of the relationships and interactions among software objects in the UML.

A collaboration diagram resembles a flowchart that portrays the roles, functionality and behavior of individual objects as well as the overall operation of the system in real time.

Collaboration diagrams are best suited to the portrayal of simple interactions among relatively small numbers of objects. As the number of objects and messages grows, a collaboration diagram can become difficult to read.



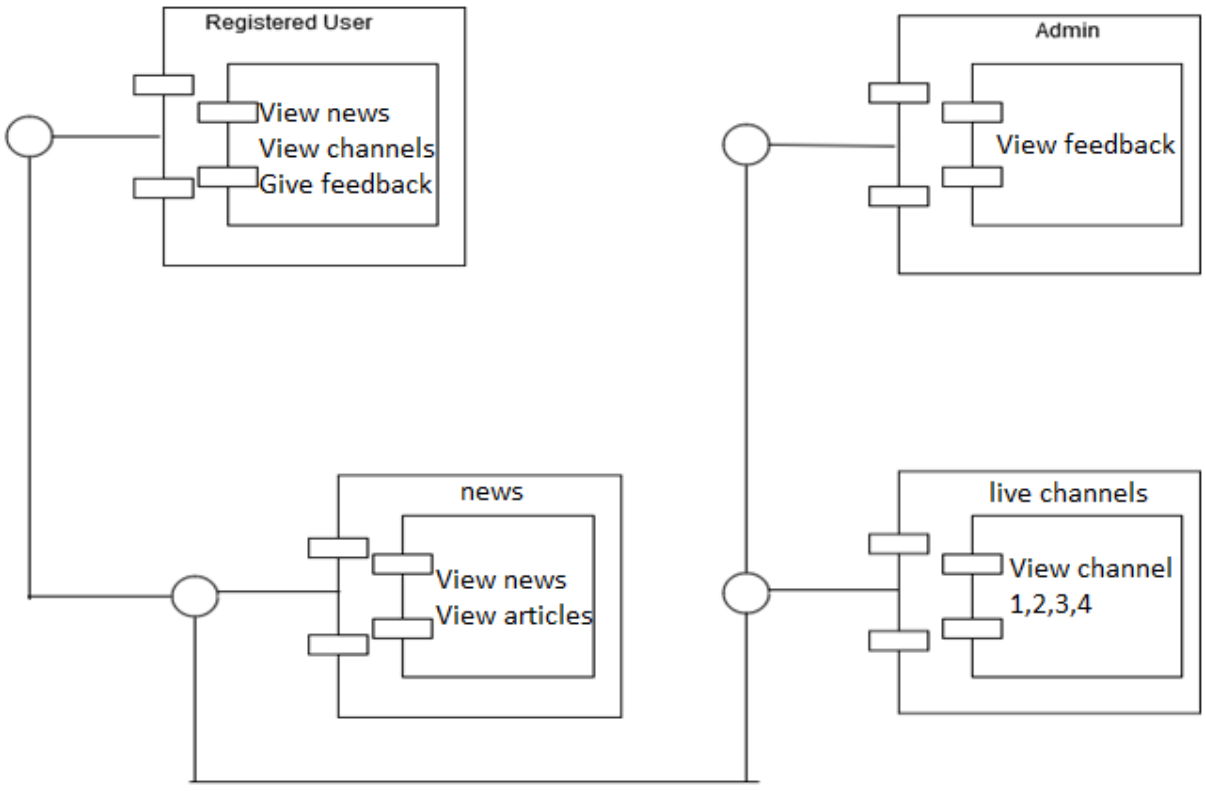
CHP 3 :- System Diagram

3.1 Component Diagram

Component Diagram are different in terms of nature and behaviour. Component diagrams are used to model physical aspects of a system.Component diagrams are used to visualize the organization and relationships among components in a system. These diagrams are also used to make executable systems.

Purpose:-Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far. It does not describe the functionality of the system but it describes the component used to make those functionalities.So from that point component diagrams are used to visualize the physical components in a system. These components are libraries, packages, files, etc.Component diagrams can also be described as a static implementation view of system. Static implementation represents the organization of the components at a particular moment.A single Component Diagram cannot respresent the entire system but a collection of diagrams are used to represent the whole.

Diagram:

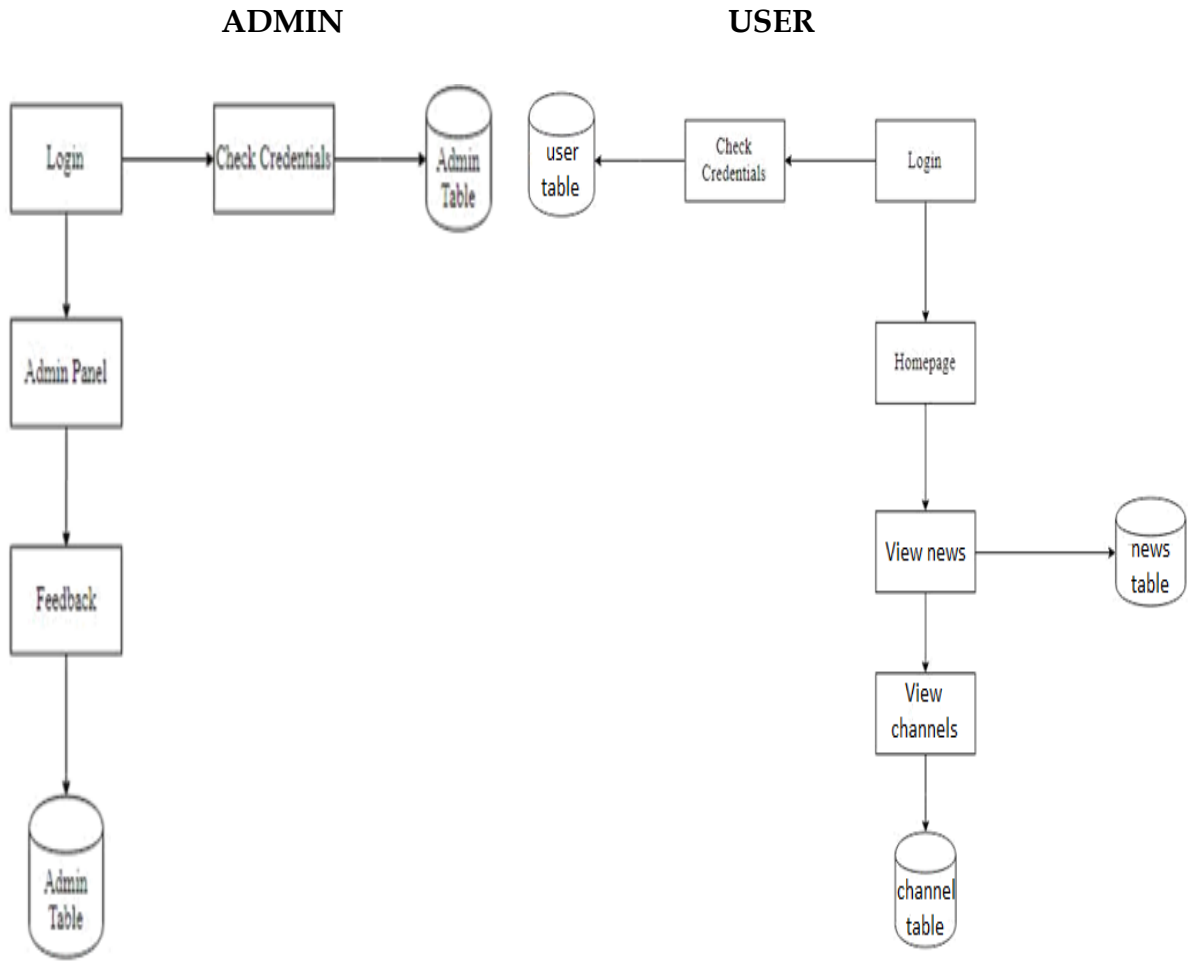


3.2 System Flow Chart Diagram

- System flow charts are a way of displaying how data flows in a system and how decisions are made to control events.
- To illustrate this, symbols are used. They are connected together to show what happens to data and where it goes. The basic ones include: symbols used in flow charts.
- The flow of data generally goes from top to bottom and left to right and depicts the sequence of processing steps along these data lines.
- The following are examples of some of the symbols used in systems flowcharts:

A systems flowchart shows the key input and outputs associated with the program. The shape of the symbols indicates the types of input or output devices.

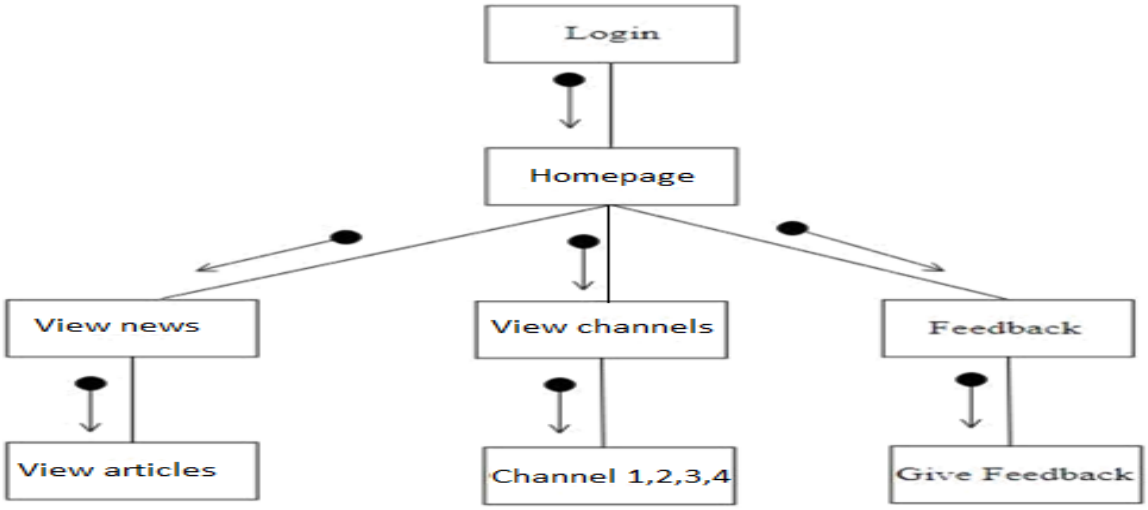
- The type of diagram dictates the flow chart symbols that are used. The terminator symbols marks the starting or ending point of the system.
- A flow chart is a formalized graphic representation of a program logic sequence, work or manufacturing process, organization
- A graphical representation of the sequence of operations in an information system or program. Information system flow charts show how data flows from source.



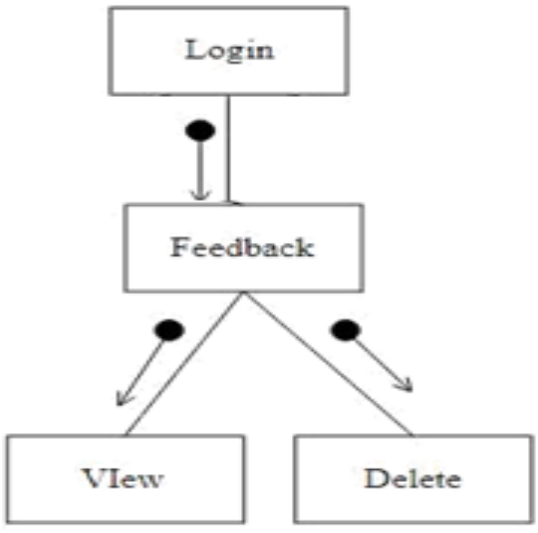
3.3 Structure Chart Diagram

A structure chart (SC) in software engineering and organizational theory, is a chart which shows the breakdown of a system to its lowest manageable levels.They are used in structured programming to arrange program modules into a tree. Each module is represented by a box, which contain the module's name. A structure chart (SC) in software engineering and organizational theory, is a chart which shows the breakdown of a system to its lowest manageable levels.They are used in structured programming to arrange program modules into a tree. Each module is represented by a box, which contains the module's name.Structure diagram is a chart derived from data flow diagram. The system structure chart represents hierarchical structure of modules.A structure chart depicts.The size and complexity of the system, and number of a readily identifiable functions and modules within each function and whether each identifiable function is a manageable entity or should be broken down into smaller components.A structure chart is also used to diagram associated elements that comprise a run stream or thread. It is often developed as a hierarchal diagram, but other representations are allowable.

USER :



ADMIN:

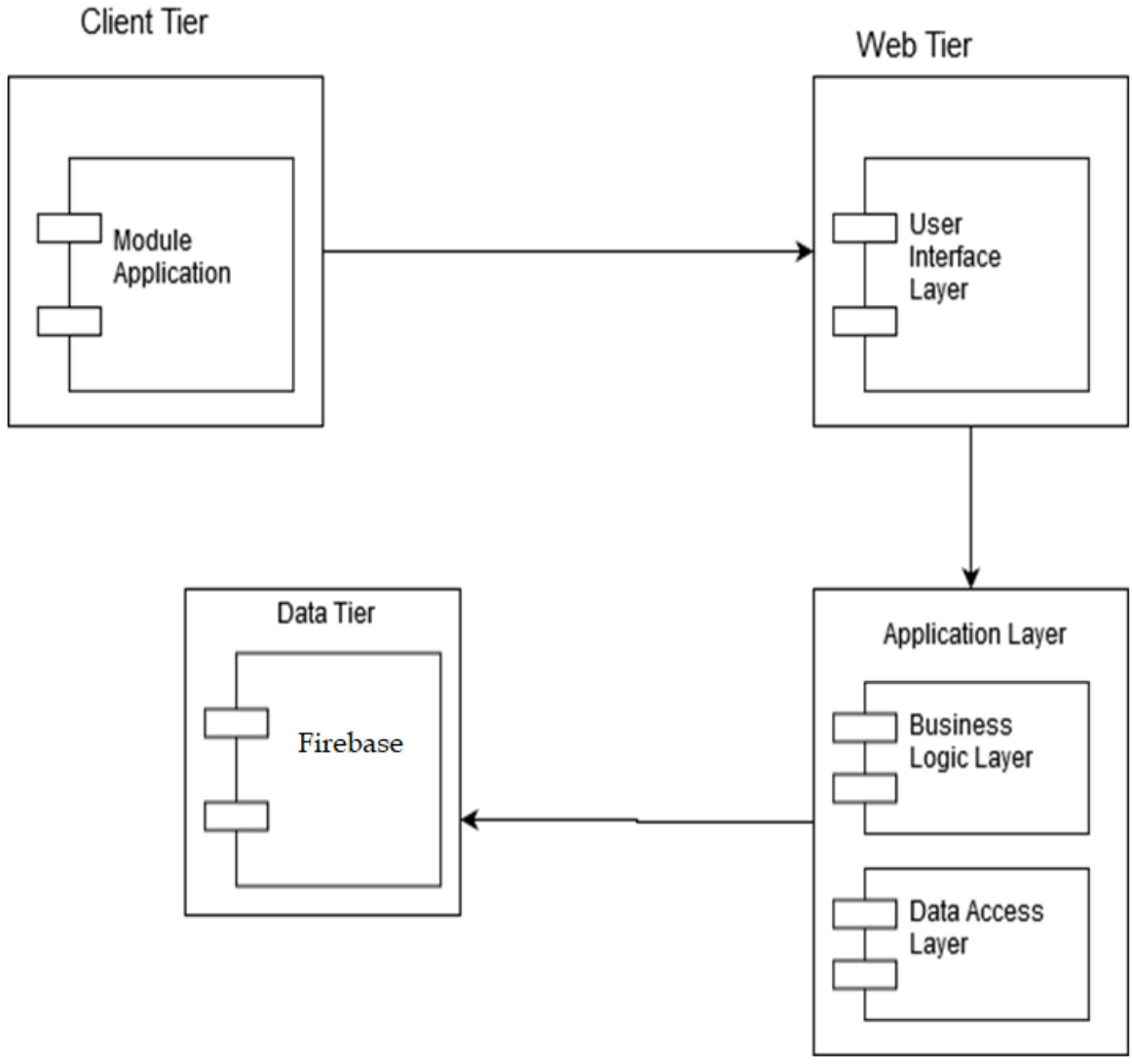


3.4 Deployment Diagram

Deployment diagrams are used to visualize the topology of the physical components of a system where the software components are deployed.So deployment diagrams are used to describe the static deployment view of a system.Deployment diagrams consist of nodes and their relationships.

Purpose:-The name Deployment itself describes the purpose of the diagram.Deployment diagrams are used for describing the hardware components where software components are deployed.Component diagrams and deployment diagrams are closely related.Component diagrams ae used to describe the components and deployment diagrams shows how they are deployed in hardware.UML is mainly designed to focus on software artifacts of a system.But these two diagrams are special diagrams used to focus on software components and hardware components.So most of the UML diagrams are used to handle logical components but deployment are made to focus on hardware topology of a system.Deployment diagrams are used by the system engineers.

Diagram:-



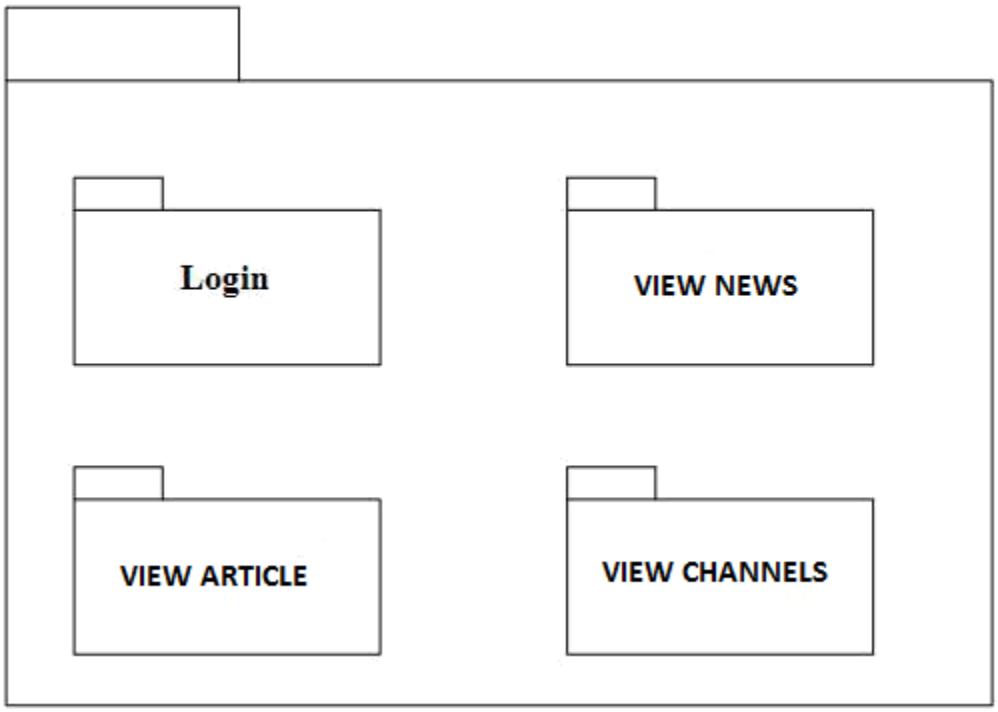
3.5 Package Diagram

1) When modeling a large scale system, you would probably be working with a high volume of model elements. They describe a model from different views and different phases, hence are in different types.

2) UML package helps to organize and arrange model elements and diagrams into logical groups, through which you can manage a chunk of project data together.

3) You can also use packages to represent different views of the systems architecture .In addition, developers can use package to model the physical package or namespace structure of the application to build.

4) Package Diagram visualizes packages and depicts the dependency, Import, access, generalization, realization and merge relationships between them. Package diagram enables you to gain a high level understanding of the collaboration among model elements through analyzing the relationships among their parent package. This also helps explain the systems architecture from a broad view.



CHAPTER 4: SYSTEM CODING

4.1 Site Map

4.2 Data Dictionary

4.3 Source Code

4 .1 SITE MAP

- HOME(VIEW NEWS)
    - ARTICLES
- LIVE CHANNELS
    - CHANNEL 1,2,3,4
- PROFILE
- LOGIN
- ABOUT US
- CONTACT US
    - FEEDBACK

4.2 DATA DICTIONARY

1.TABLES: News

Field	Data type	Length	Constraints
Title	Varchar	20	Primary Key
Desc	Varchar	30	Null
Date	Varchar	100	Null
Author	Interger	10	Null
Source	Varchar	30	Null

2.TABLES: Article

Field	Data type	Length	Constraints
article	Varchar	20	Null
Title	Varchar	100	Primary Key
Date	Interger	50	Null

3. TABLES**: Live channels**

Field	Data type	Length	Constraints
Channel1	Varchar	20	Null
Channel2	Varchar	100	Null
Channel3	Varchar	50	Null
Channel4	Varchar	30	Null

4.3 Source code

Apiclient.java

```
public class ApiClient {
   public static final String BASE_URL = "http://newsapi.org/v2/";
   public static Retrofit retrofit;
   public static Retrofit getApiClient(){
      if(retrofit == null){
         retrofit = new Retrofit.Builder().baseUrl(BASE_URL)
               .client(getUnsafeOkHttpClient().build())
               .addConverterFactory(GsonConverterFactory.create())
               .build();
      }
      return retrofit;
   }
   public static OkHttpClient.Builder getUnsafeOkHttpClient(){
      try {
         // Create a trust manager that does not validate certificate chains
         final TrustManager[] trustAllCerts = new TrustManager[]{
               new X509TrustManager() {
                  @Override
                  public void checkClientTrusted(java.security.cert.X509Certificate[] chain,
String authType) throws CertificateException {}
                  @Override
                  public void checkServerTrusted(java.security.cert.X509Certificate[] chain,
String authType) throws CertificateException {}
                  @Override
                  public java.security.cert.X509Certificate[] getAcceptedIssuers() {
```

```
                return new java.security.cert.X509Certificate[]{};
            }}};
        // Install the all-trusting trust manager
        final SSLContext sslContext = SSLContext.getInstance("SSL");
        sslContext.init(null, trustAllCerts, new java.security.SecureRandom());
        // Create an ssl socket factory with our all-trusting manager
        final SSLSocketFactory sslSocketFactory = sslContext.getSocketFactory();
        OkHttpClient.Builder builder = new OkHttpClient.Builder();
        builder.sslSocketFactory(sslSocketFactory, (X509TrustManager) trustAllCerts[0]);
        builder.hostnameVerifier(new HostnameVerifier() {
            @Override
            public boolean verify(String hostname, SSLSession session) {
                return true; } });
        return builder; }
      catch (Exception e) {
        throw new RuntimeException(e);}}}
```

ApiInterface.java

```
public interface ApiInterface {
    @GET("top-headlines")
    Call<News> getNews(
        @Query("country") String country ,
        @Query("apiKey") String apiKey );
    @GET("everything")
    Call<News> getNewsSearch(
        @Query("q") String keyword,
        @Query("language") String language,
        @Query("sortBy") String sortBy,
```

```
        @Query("apiKey") String apiKey  );}
```

Articles.java

```
public class Articles {
    @SerializedName("source")
    @Expose
    private Source source;
    @SerializedName("author")
    @Expose
    private String author;
    @SerializedName("title")
    @Expose
    private  String title;
    @SerializedName("description")
    @Expose
    private String description;
    @SerializedName("url")
    @Expose
    private String url;
    @SerializedName("urlToImage")
    @Expose
    private String urlToImage;
    @SerializedName("publishedAt")
    @Expose
    private  String publishedAt;
    public Source getSource() {
        return source; }
    public void setSource(Source source) {
```

```
    this.source = source;}
public String getAuthor() {
    return author;  }
public void setAuthor(String author) {
    this.author = author;}
public String getTitle() {
    return title; }

public void setTitle(String title) {
    this.title = title; }
public String getDescription() {
    return description;}
public void setDescription(String description) {
    this.description = description;}
public String getUrl() {
    return url;}
public void setUrl(String url) {
    this.url = url;}
public String getUrlToImage() {
    return urlToImage; }
public void setUrlToImage(String urlToImage) {
    this.urlToImage = urlToImage;}
public String getPublishedAt() {
    return publishedAt;}
public void setPublishedAt(String publishedAt) {
    this.publishedAt = publishedAt;  }
@Override
```

```
    public String toString() {
        return "Articles{" +
                "source=" + source +
                ", author='" + author + '\'' +
                ", title='" + title + '\'' +
                ", description='" + description + '\'' +
                ", url='" + url + '\'' +
                ", urlToImage='" + urlToImage + '\'' +
                ", publishedAt='" + publishedAt + '\'' +  '}';}}
```

News.java

```
public class News {
    @SerializedName("status")
    @Expose
    private String status;
    @SerializedName("totalResults")
    @Expose
    private String totalResults;
    @SerializedName("articles")
    @Expose
    private List<Articles> articles;
    public String getStatus() {
        return status;}
    public void setStatus(String status) {
        this.status = status;}
    public String getTotalResults() {
        return totalResults;}
    public void setTotalResults(String totalResults) {
```

```
    this.totalResults = totalResults;}
public List<Articles> getArticles() {
    return articles;}
public void setArticles(List<Articles> articles) {
    this.articles = articles;}
@Override
public String toString() {
    return "News{" +
            "status='" + status + '\'' +
            ", totalResults='" + totalResults + '\'' +
            ", articles=" + articles +  '}';}}
```

Source.java

```
package com.example.newsapp11.model;
import com.google.gson.annotations.Expose;
import com.google.gson.annotations.SerializedName;
public class Source {
    @SerializedName("id")
    @Expose
    private String id;
    @SerializedName("name")
    @Expose
    private String name;
    public String getId() {
        return id;}
    public void setId(String id) {
        this.id = id;}
    public String getName() {
```

```
    return name;}
  public void setName(String name) {
    this.name = name;}
  @Override
  public String toString() {
    return "Source{" +
        "id='" + id + '\'' +
        ", name='" + name + '\'' +
        '}';  }}
```

Adapter.java

```
public class Adapter extends RecyclerView.Adapter<Adapter.MyViewHolder> {
  private List<Articles> articles;
  private Context context;
  private OnItemClickListener onItemClickListener;
  public Adapter(List<Articles> articles, Context context) {
    this.articles = articles;
    this.context = context;  }
  @NonNull
  @Override
  public MyViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType)
{
    View view = LayoutInflater.from(context).inflate(R.layout.item, parent, false);
    return new MyViewHolder(view, onItemClickListener);  }
  @Override
  public void onBindViewHolder(@NonNull MyViewHolder holders, int position) {
    final MyViewHolder holder = holders;
    Articles model = articles.get(position);
```

```
RequestOptions requestOptions = new RequestOptions();
requestOptions.placeholder(Utils.getRandomDrawbleColor());
requestOptions.error(Utils.getRandomDrawbleColor());
requestOptions.diskCacheStrategy(DiskCacheStrategy.ALL);
requestOptions.centerCrop();
Glide.with(context)
        .load(model.getUrlToImage())
        .apply(requestOptions)
        .listener(new RequestListener<Drawable>() {
           @Override
           public boolean onLoadFailed(@Nullable GlideException e, Object model, Target<Drawable> target, boolean isFirstResource) {
              holder.progressbar.setVisibility(View.GONE);
              return false;}
           @Override
           public boolean onResourceReady(Drawable resource, Object model, Target<Drawable> target, DataSource dataSource, boolean isFirstResource) {
              holder.progressbar.setVisibility(View.GONE);
              return false;}})
        .transition(DrawableTransitionOptions.withCrossFade())
        .into(holder.imageView);
    holder.title.setText(model.getTitle());
    holder.desc.setText(model.getDescription());
    holder.source.setText(model.getSource().getName());
    holder.time.setText("\u2022"+Utils.DateToTimeFormat(model.getPublishedAt()));
    holder.publishedAt.setText(Utils.DateFormat(model.getPublishedAt()));
    holder.author.setText(model.getAuthor());}
  @Override
```

```
    public int getItemCount() {
        return articles.size();}
    public void setOnItemClickListener(OnItemClickListener onItemClickListener){
        this.onItemClickListener = onItemClickListener;}
    public interface OnItemClickListener
    { void onItemClick(View v, int adapterPosition);}
    public class MyViewHolder extends RecyclerView.ViewHolder implements
View.OnClickListener{
        TextView title,desc,author,publishedAt,source,time;
        ImageView imageView;
        ProgressBar progressbar;
        OnItemClickListener onItemClickListener;
        public MyViewHolder(View itemView,OnItemClickListener onItemClickListener){
            super(itemView);
            itemView.setOnClickListener(this);
            title = itemView.findViewById(R.id.title);
            desc = itemView.findViewById(R.id.desc);
            author = itemView.findViewById(R.id.author);
            publishedAt = itemView.findViewById(R.id.publishedAt);
            source = itemView.findViewById(R.id.source);
            time = itemView.findViewById(R.id.time);
            imageView = itemView.findViewById(R.id.img);
            progressbar = itemView.findViewById(R.id.progress_load);
            this.onItemClickListener = onItemClickListener;}
        @Override
        public void onClick(View v) {
 onItemClickListener.onItemClick(v,getAdapterPosition());}}}
```

Utils.java

```
public class Utils {
   public static ColorDrawable[] vibrantLightColorList ={
                new ColorDrawable(Color.parseColor("#ffeead")),
                new ColorDrawable(Color.parseColor("#93cfb3")),
                new ColorDrawable(Color.parseColor("#fd7a7a")),
                new ColorDrawable(Color.parseColor("#faca5f")),
                new ColorDrawable(Color.parseColor("#1ba798")),
                new ColorDrawable(Color.parseColor("#6aa9ae")),
                new ColorDrawable(Color.parseColor("#ffbf27")),
                new ColorDrawable(Color.parseColor("#d93947")) };
   public static ColorDrawable getRandomDrawbleColor() {
      int idx = new Random().nextInt(vibrantLightColorList.length);
      return vibrantLightColorList[idx]; }
   public static String DateToTimeFormat(String oldstringDate) {
      PrettyTime p = new PrettyTime(new Locale(getCountry()));
      String isTime = null;
      try {
         SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd'T'HH:mm:ss'Z'",
               Locale.ENGLISH);
         Date date = sdf.parse(oldstringDate);
         isTime = p.format(date);
      } catch (ParseException e) {
         e.printStackTrace();}
      return isTime;}
   public static String DateFormat(String oldstringDate) {
      String newDate;
```

```
        SimpleDateFormat dateFormat = new SimpleDateFormat("E, d MMM yyyy", new
    Locale(getCountry()));
        try {
            Date date = new SimpleDateFormat("yyyy-MM-
    dd'T'HH:mm:ss'Z'").parse(oldstringDate);
            newDate = dateFormat.format(date);
        } catch (ParseException e) {
            e.printStackTrace();
            newDate = oldstringDate;}
        return newDate;}
    public static String getCountry() {
        Locale locale = Locale.getDefault();
        String country = String.valueOf(locale.getCountry());
        return country.toLowerCase();}
    public static String getLanguage() {
        Locale locale = Locale.getDefault();
        String country = String.valueOf(locale.getLanguage());
        return country.toLowerCase();}}
```

MainActivity,java

```
public class MainActivity extends AppCompatActivity implements
SwipeRefreshLayout.OnRefreshListener{
    public static final String API_KEY = "************************";
private RecyclerView recyclerView;
    private RecyclerView.LayoutManager layoutManager;
    private List<Articles> articles = new ArrayList<>();
    private Adapter adapter;
    private String TAG = MainActivity.class.getSimpleName();
    private SwipeRefreshLayout swipeRefreshLayout;
```

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    recyclerView = findViewById(R.id.recyclerView);
    layoutManager = new LinearLayoutManager(MainActivity.this);
    recyclerView.setLayoutManager(layoutManager);
    recyclerView.setItemAnimator(new DefaultItemAnimator());
    recyclerView.setNestedScrollingEnabled(false);
    swipeRefreshLayout = findViewById(R.id.refresh_layout);
    swipeRefreshLayout.setOnRefreshListener(this);
    swipeRefreshLayout.setColorSchemeResources(R.color.colorAccent);
    onLoadingSwipeRefresh(""); }
public void LoadJson(final String keyword){
    swipeRefreshLayout.setRefreshing(true);
    ApiInterface apiInterface = ApiClient.getApiClient().create(ApiInterface.class);
    String country = Utils.getCountry();
    String language = Utils.getLanguage();
    Call<News> call;
    if (keyword.length() > 0){
        call = apiInterface.getNewsSearch(keyword, language,"publishedAt",API_KEY); }
    else {
        call = apiInterface.getNews(country, API_KEY); }
    call = apiInterface.getNews(country, API_KEY);
    call.enqueue(new Callback<News>() {
        @Override
        public void onResponse(Call<News> call, Response<News> response) {
```

```
            if (response.isSuccessful() && response.body().getArticles() !=null){
                if (!articles.isEmpty()){
                    articles.clear(); }
                articles = response.body().getArticles();
                adapter = new Adapter(articles, MainActivity.this);
                recyclerView.setAdapter(adapter);
                adapter.notifyDataSetChanged();
                initListener();
                swipeRefreshLayout.setRefreshing(false);
            }else {
                swipeRefreshLayout.setRefreshing(false);
                Toast.makeText(MainActivity.this, "No Result",
Toast.LENGTH_SHORT).show();} }
        @Override
        public void onFailure(Call<News> call, Throwable t) {
            swipeRefreshLayout.setRefreshing(false); }}); }
    private void initListener(){
        adapter.setOnItemClickListener(new Adapter.OnItemClickListener() {
        @Override
        public void onItemClick(View v, int position) {
            Intent intent = new Intent(MainActivity.this, MainActivity2.class);
            Articles article = articles.get(position);
            intent.putExtra("url",article.getUrl());
            intent.putExtra("title",article.getTitle());
            intent.putExtra("img",article.getUrlToImage());
            intent.putExtra("date",article.getPublishedAt());
            intent.putExtra("source",article.getSource().getName());
```

```
            intent.putExtra("author",article.getAuthor());
            startActivity(intent); }}); }
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        MenuInflater inflater = getMenuInflater();
        inflater.inflate(main_menu, menu);
        SearchManager searchManager = (SearchManager) getSystemService(Context.SEARCH_SERVICE);
        final SearchView searchView = (SearchView) menu.findItem(R.id.action_search).getActionView();
        MenuItem searchMenuItem = menu.findItem(R.id.action_search);

searchView.setSearchableInfo(searchManager.getSearchableInfo(getComponentName()));
        searchView.setQueryHint("Search Latest News...");
        searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
            @Override
            public boolean onQueryTextSubmit(String query) {
                if (query.length() > 2){
                    onLoadingSwipeRefresh(query); }
                return false; }
            @Override
            public boolean onQueryTextChange(String newText) {
                onLoadingSwipeRefresh(newText);
                return false; }});
        searchMenuItem.getIcon().setVisible(false,false);
        return true; }
    @Override
    public void onRefresh() {
```

```
    onLoadingSwipeRefresh("");
}
private void onLoadingSwipeRefresh(final String keyword){
    swipeRefreshLayout.post(
            new Runnable() {
                @Override
                public void run() {
                    LoadJson(keyword);
                }}); }
public boolean onOptionsItemSelected(MenuItem item){
    int id = item.getItemId();
    if (id==R.id.live){
        Intent intent= new Intent(MainActivity.this,live_channels.class);
        startActivity(intent);
        return true; }
    else if (id==R.id.otherlogin){
        Intent intent= new Intent(MainActivity.this,loginactivity.class);
        startActivity(intent);
        return true; }
    else if (id==R.id.profile){
        Intent intent= new Intent(MainActivity.this,userinfo.class);
        startActivity(intent);
        return true; }
    else if (id==R.id.about){
        Intent intent= new Intent(MainActivity.this,about.class);
        startActivity(intent);
        return true; }
```

```
        else if (id==R.id.contact){
            Intent intent= new Intent(MainActivity.this,contact.class);
            startActivity(intent);
            return true; }
        return super.onOptionsItemSelected(item); }}
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    tools:context=".MainActivity">
    <androidx.swiperefreshlayout.widget.SwipeRefreshLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="@color/colorBackground"
        android:id="@+id/refresh_layout">
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_below="@+id/search_field"
        android:layout_marginTop="10dp">
        <androidx.core.widget.NestedScrollView
            android:layout_width="match_parent"
            android:layout_height="match_parent">
            <androidx.recyclerview.widget.RecyclerView
```

```
            android:id="@+id/recyclerView"
            android:layout_width="match_parent"
            android:layout_height="match_parent">
        </androidx.recyclerview.widget.RecyclerView>
    </androidx.core.widget.NestedScrollView>
</RelativeLayout>
</androidx.swiperefreshlayout.widget.SwipeRefreshLayout></RelativeLayout>
```

MainActivity2.java

```
public class MainActivity2 extends AppCompatActivity implements AppBarLayout.OnOffsetChangedListener {
    private ImageView imageView;
    private TextView appbar_title, appbar_subtitle, date, time, title;
    private boolean isHideToolBarView = false;
    private FrameLayout date_behaviour;
    private LinearLayout titleAppBar;
    private AppBarLayout appBarLayout;
    private Toolbar toolbar;
    private String mUrl, mImg, mDate, mSource, mAuthor,mTitle;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2);
        toolbar =findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        getSupportActionBar().setTitle("");
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
```

```
final CollapsingToolbarLayout collapsingToolbarLayout = findViewById(R.id.collapsing_toolbar);
collapsingToolbarLayout.setTitle("");
appBarLayout = findViewById(R.id.appbar);
appBarLayout.addOnOffsetChangedListener(this);
date_behaviour = findViewById(R.id.date_behavior);
titleAppBar = findViewById(R.id.title_appbar);
imageView = findViewById(R.id.backdrop);
appbar_title = findViewById(R.id.title_on_appbar);
appbar_subtitle = findViewById(R.id.subtitle_on_appbar);
date = findViewById(R.id.date);
time = findViewById(R.id.time);
title = findViewById(R.id.title);
Intent intent=getIntent();
mUrl=intent.getStringExtra("url");
mImg=intent.getStringExtra("img");
mTitle=intent.getStringExtra("title");
mDate=intent.getStringExtra("date");
mSource=intent.getStringExtra("source");
mAuthor=intent.getStringExtra("author");
RequestOptions requestOptions = new RequestOptions();
requestOptions.error(Utils.getRandomDrawbleColor());
Glide.with(this)
        .load(mImg)
        .apply(requestOptions)
        .transition(DrawableTransitionOptions.withCrossFade())
        .into(imageView);
```

```
    appbar_title.setText(mSource);
    appbar_subtitle.setText(mUrl);
    date.setText(Utils.DateFormat(mDate));
    title.setText(mTitle);
    String author = null;
    if(mAuthor != null || mAuthor !=""){
       mAuthor = "\u2022"+mAuthor;
    }else { author=""; }
    time.setText(mSource+author+"\u2022"+Utils.DateToTimeFormat(mDate));
    initWebView(mUrl); }
private void  initWebView(String url){
    WebView webView = findViewById(R.id.webView);
    webView.getSettings().setLoadsImagesAutomatically(true);
    webView.getSettings().setJavaScriptEnabled(true);
    webView.getSettings().setDomStorageEnabled(true);
    webView.getSettings().setSupportZoom(true);
    webView.getSettings().setBuiltInZoomControls(true);
    webView.getSettings().setDisplayZoomControls(false);
    webView.setWebViewClient(new WebViewClient());
    webView.loadUrl(url); }
@Override
public void onBackPressed() {
    super.onBackPressed();
    supportFinishAfterTransition(); }
@Override
public boolean onSupportNavigateUp() {
    onBackPressed();
```

```
        return true;}
    @Override
    public void onOffsetChanged(AppBarLayout appBarLayout, int verticalOffset) {
        int maxScroll = appBarLayout.getTotalScrollRange();
        float percentage = (float) Math.abs(verticalOffset)/(float)maxScroll;

        if (percentage == 1f && isHideToolBarView) {
            date_behaviour.setVisibility(View.GONE);
            titleAppBar.setVisibility(View.VISIBLE);
            isHideToolBarView = !isHideToolBarView;
        } else if (percentage < 1f && isHideToolBarView) {
            date_behaviour.setVisibility(View.VISIBLE);
            titleAppBar.setVisibility(View.GONE);
            isHideToolBarView = !isHideToolBarView; } }}
```

activity_main2.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.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"
    android:fitsSystemWindows="true"
    android:background="@color/colorBackground"
    tools:context=".MainActivity2">
    <com.google.android.material.appbar.AppBarLayout
        android:id="@+id/appbar"
```

```
android:layout_width="match_parent"
android:layout_height="250dp"
android:fitsSystemWindows="true"
android:theme="@style/MyAppBarLayoutTheme">
<com.google.android.material.appbar.CollapsingToolbarLayout
    android:id="@+id/collapsing_toolbar"
    android:layout_width="match_parent"
    app:titleEnabled="false"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true"
    app:contentScrim="?attr/colorPrimary"
    app:layout_scrollFlags="scroll|exitUntilCollapsed">
    <ImageView
        android:id="@+id/backdrop"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:fitsSystemWindows="true"
        android:scaleType="centerCrop"
        app:layout_collapseMode="parallax"
        android:transitionName="img"
        tools:ignore="UnusedAttribute" />
    <RelativeLayout
        android:id="@+id/headerContent"
        app:layout_collapseMode="pin"
        android:fitsSystemWindows="true"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
```

```
    android:layout_centerInParent="true"
    android:orientation="vertical">
    <ImageView
        android:src="@drawable/top_shadow"
        android:scaleType="centerCrop"
        android:id="@+id/img"
        android:layout_width="match_parent"
        android:layout_height="70dp" />
    <ImageView
        android:layout_alignParentBottom="true"
        android:src="@drawable/bottom_shadow"
        android:id="@+id/img2"
        android:layout_alignBottom="@id/img"
        android:scaleType="centerCrop"
        android:layout_width="match_parent"
        android:layout_height="80dp" />
</RelativeLayout>
<androidx.appcompat.widget.Toolbar
    android:id="@+id/toolbar"
    android:layout_width="match_parent"
    android:layout_height="?attr/actionBarSize"
    android:contentInsetStart="0dp"
    android:contentInsetLeft="0dp"
    app:contentInsetLeft="0dp"
    app:contentInsetStart="0dp"
    app:layout_collapseMode="pin"
    app:popupTheme="@style/ThemeOverlay.AppCompat.Light">
```

```
<LinearLayout
    android:id="@+id/title_appbar"
    android:clickable="false"
    android:layout_width="wrap_content"
    android:orientation="vertical"
    android:layout_height="wrap_content">
    <TextView
        android:id="@+id/title_on_appbar"
        style="@style/Base.TextAppearance.AppCompat.Widget.ActionBar.Title"
        android:text="News for you"
        android:textSize="18dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:maxLines="1"
        android:drawablePadding="10dp"
        android:singleLine="true"
        android:ellipsize="end"/>
    <TextView
        android:id="@+id/subtitle_on_appbar"
        style="@style/Base.TextAppearance.AppCompat.Widget.ActionBar.Subtitle"
        android:text="Subtitle"
        android:textSize="12dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:maxLines="1"
        android:drawablePadding="10dp"
        android:singleLine="true"
```

```
                android:ellipsize="end"/>
            </LinearLayout>
        </androidx.appcompat.widget.Toolbar>
    </com.google.android.material.appbar.CollapsingToolbarLayout>
</com.google.android.material.appbar.AppBarLayout>
<androidx.core.widget.NestedScrollView
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    android:fitsSystemWindows="true"
    android:layout_width="match_parent"
    android:background="@color/colorBackground"
    android:layout_height="wrap_content">
    <LinearLayout
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <androidx.cardview.widget.CardView
            app:layout_behavior="@string/appbar_scrolling_view_behavior"
            android:layout_width="match_parent"
            app:cardCornerRadius="0dp"
            app:cardElevation="@dimen/cardview_default_elevation"
            android:layout_height="wrap_content">
            <RelativeLayout
                android:layout_marginBottom="20dp"
                android:layout_marginTop="20dp"
                android:paddingLeft="16dp"
                android:paddingRight="16dp"
                android:layout_width="match_parent"
```

```
        android:layout_height="wrap_content">
        <TextView
            android:id="@+id/title"
            android:textColor="@color/colorTextTitle"
            android:textStyle="bold"
            android:fontFamily="sans-serif-light"
            android:textSize="19sp"
            android:text="Title of News"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"  />
        <TextView
            android:id="@+id/time"
            android:layout_width="wrap_content"
            android:layout_height="20dp"
            android:layout_below="@id/title"
            android:layout_marginTop="10dp"
            android:maxLines="1"
            android:drawablePadding="10dp"
            android:singleLine="true"
            android:ellipsize="end"
            android:text="a time ago" />
    </RelativeLayout>
</androidx.cardview.widget.CardView>
<androidx.cardview.widget.CardView
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    android:layout_marginTop="12dp"
    android:layout_marginBottom="20dp"
```

```
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:cardElevation="@dimen/cardview_default_elevation"
        app:cardCornerRadius="0dp">
        <RelativeLayout
          android:layout_width="match_parent"
          android:layout_height="match_parent">
          <ProgressBar
            app:layout_behavior="@string/appbar_scrolling_view_behavior"
            android:id="@+id/progress_bar"
            android:layout_marginTop="50dp"
            android:layout_marginBottom="70dp"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" />
          <WebView
            app:layout_behavior="@string/appbar_scrolling_view_behavior"
            android:id="@+id/webView"
            android:layout_width="match_parent"
            android:layout_height="match_parent"/>
        </RelativeLayout>
      </androidx.cardview.widget.CardView>
    </LinearLayout>
</androidx.core.widget.NestedScrollView>
<FrameLayout
    android:id="@+id/date_behavior"
    app:layout_anchor="@+id/appbar"
    app:behavior_autoHide="true"
```

```
        android:adjustViewBounds="true"
        app:layout_anchorGravity="right|end|bottom"
        android:clickable="false"
        android:layout_below="@+id/img"
        android:background="@drawable/round_white"
        android:layout_width="wrap_content"
        android:padding="5dp"
        android:layout_alignParentBottom="true"
        android:layout_alignParentRight="true"
        android:layout_marginRight="20dp"
        android:layout_marginBottom="410dp"
        android:layout_height="wrap_content"
        tools:ignore="UnusedAttribute">
        <ImageView
            android:src="@drawable/ic_date"
            android:layout_width="18dp"
            android:layout_height="18dp"
            android:layout_marginLeft="5dp"
            android:layout_marginRight="5dp"/>
        <TextView
            android:id="@+id/date"
            android:layout_marginLeft="27dp"
            android:layout_marginRight="10dp"
            android:text="01 January 1990"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
</FrameLayout></androidx.coordinatorlayout.widget.CoordinatorLayout>
```

Contact.java

```
public class contact extends AppCompatActivity {
    private Firebase Ref;
    private TextView emailTV;
    private EditText feedback;
    GoogleSignInClient mGoogleSignInClient;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_contact);
        emailTV = (TextView) findViewById(R.id.email);
        feedback=(EditText) findViewById(R.id.feedback);
        Firebase.setAndroidContext(this);
        Ref=new Firebase("https://newsapp11-1583994573079.firebaseio.com/");
        // Configure sign-in to request the user's ID, email address, and basic
        // profile. ID and basic profile are included in DEFAULT_SIGN_IN.
        GoogleSignInOptions gso = new
GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
                .requestEmail()
                .build();
        // Build a GoogleSignInClient with the options specified by gso.
        mGoogleSignInClient = GoogleSignIn.getClient(this, gso);
        GoogleSignInAccount acct = GoogleSignIn.getLastSignedInAccount(contact.this);
        if (acct != null) {
            String personEmail = acct.getEmail();
            emailTV.setText("Email: " + personEmail); } }
    public void feedbacksent(View view) {
```

```
        String usernameinput;
        usernameinput = emailTV.getText().toString();
        String feedbackinput=feedback.getText().toString();
        Firebase Reusername=Ref.child("email");
        Reusername.setValue(usernameinput);
        Firebase Reffeedback=Ref.child("Feedback");
        Reffeedback.setValue(feedbackinput);
      Toast.makeText(contact.this,"Feedbackhasbeingsent",Toast.LENGTH_LONG).show();}}
```

Contact.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="vertical"
    tools:context=".contact">
    <TextView
        android:id="@+id/email"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="EMAIL HERE"
        android:textColor="@color/colorPrimary"
        android:textSize="24sp" />
    <EditText
        android:id="@+id/feedback"
```

```
        android:layout_width="match_parent"
        android:layout_height="370dp"
        android:ems="10"
        android:hint="your messege"
        android:inputType="textPersonName"
        android:textColor="@color/colorPrimary"
        tools:layout_editor_absoluteX="12dp"
        tools:layout_editor_absoluteY="61dp" />
    <Button
        android:layout_width="match_parent"
        android:layout_height="100dp"
        android:layout_gravity="center"
        android:onClick="feedbacksent"
        android:text="SEND THE FEEDBACK"
        android:textColor="@color/colorPrimary" />
</LinearLayout>
```

Userinfo.java

```
public class userinfo extends AppCompatActivity {
    private Button Button1;
    GoogleSignInClient mGoogleSignInClient;
    Button sign_out;
    TextView nameTV;
    TextView emailTV;
    TextView idTV;
    ImageView photoIV;
    @Override
```

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_userinfo);
    Button1 = (Button) findViewById(R.id.homebtn);
    Button1.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            openactivity_main();
        }});
    sign_out = findViewById(R.id.log_out);
    nameTV = findViewById(R.id.name);
    emailTV = findViewById(R.id.email);
    idTV = findViewById(R.id.id);
    photoIV = findViewById(R.id.photo);
    // Configure sign-in to request the user's ID, email address, and basic
    // profile. ID and basic profile are included in DEFAULT_SIGN_IN.
    GoogleSignInOptions gso = new
GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
            .requestEmail()
            .build();
    // Build a GoogleSignInClient with the options specified by gso.
    mGoogleSignInClient = GoogleSignIn.getClient(this, gso);
    GoogleSignInAccount acct = GoogleSignIn.getLastSignedInAccount(userinfo.this);
    if (acct != null) {
        String personName = acct.getDisplayName();
        String personGivenName = acct.getGivenName();
        String personFamilyName = acct.getFamilyName();
```

```
            String personEmail = acct.getEmail();
            String personId = acct.getId();
            Uri personPhoto = acct.getPhotoUrl();
            nameTV.setText("Name: "+personName);
            emailTV.setText("Email: "+personEmail);
            idTV.setText("ID: "+personId);
            Glide.with(this).load(personPhoto).into(photoIV); }
        sign_out.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
               signOut();} }); }
    public void openactivity_main(){
        Intent intent = new Intent(this, MainActivity.class);
        startActivity(intent); }
    private void signOut() {
        mGoogleSignInClient.signOut()
                .addOnCompleteListener(this, new OnCompleteListener<Void>() {
                    @Override
                    public void onComplete(@NonNull Task<Void> task) {
                        Toast.makeText(userinfo.this,"Successfully                    signed
out",Toast.LENGTH_SHORT).show();
                        startActivity(new Intent(userinfo.this, loginactivity.class));
                        finish(); }}); }}
```

Userinfo.xml

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

```
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".userinfo">
<Button
   android:id="@+id/homebtn"
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:layout_alignParentBottom="true"
   android:layout_marginBottom="86dp"
   android:background="@color/colorPrimary"
   android:text="GO TO THE HOME PAGE"
   android:textColor="#fff" />
<LinearLayout
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:orientation="vertical"
   android:layout_centerHorizontal="true"
   android:layout_centerVertical="true">
   <ImageView
      android:layout_width="80dp"
      android:layout_height="80dp"
      android:background="@drawable/ic_portraitman"
      android:layout_gravity="center"
      android:id="@+id/photo"/>
```

```
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/name"
        android:text="Name: NAME HERE"
        android:textSize="18sp"
        android:textColor="@color/colorPrimary"
        android:layout_gravity="center"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/email"
        android:text="Email: EMAIL HERE"
        android:textSize="18sp"
        android:textColor="@color/colorPrimary"
        android:layout_gravity="center"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/id"
        android:text="ID: ID HERE"
        android:textSize="18sp"
        android:textColor="@color/colorPrimary"
        android:layout_gravity="center"/>
</LinearLayout>
<Button
    android:layout_width="match_parent"
```

```
        android:layout_height="wrap_content"
        android:id="@+id/log_out"
        android:text="Sign out"
        android:layout_centerHorizontal="true"
        android:layout_alignParentBottom="true"
        android:layout_marginBottom="20dp"
        android:background="@color/colorPrimary"
        android:textColor="#fff"/>
</RelativeLayout>
```

Loginactivity.java

```
public class loginactivity extends AppCompatActivity {
    int RC_SIGN_IN = 0;
    SignInButton signInButton;
    GoogleSignInClient mGoogleSignInClient;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_loginactivity);
        //Initializing Views
        signInButton = findViewById(R.id.sign_in_button);
        // Configure sign-in to request the user's ID, email address, and basic
        // profile. ID and basic profile are included in DEFAULT_SIGN_IN.
        GoogleSignInOptions gso = new
GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
                .requestEmail()
                .build();
```

```
        // Build a GoogleSignInClient with the options specified by gso.
        mGoogleSignInClient = GoogleSignIn.getClient(this, gso);
        signInButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                signIn();}  });}
    private void signIn() {
        Intent signInIntent = mGoogleSignInClient.getSignInIntent();
        startActivityForResult(signInIntent, RC_SIGN_IN);}
    @Override
    public void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        // Result returned from launching the Intent from
GoogleSignInClient.getSignInIntent(...);
        if (requestCode == RC_SIGN_IN) {
            // The Task returned from this call is always completed, no need to attach a listener.
            Task<GoogleSignInAccount>task=
GoogleSignIn.getSignedInAccountFromIntent(data);
            handleSignInResult(task);}}
    private void handleSignInResult(Task<GoogleSignInAccount> completedTask) {
        try {GoogleSignInAccount account = completedTask.getResult(ApiException.class);
            // Signed in successfully, show authenticated UI.
            startActivity(new Intent(loginactivity.this, MainActivity.class));
        } catch (ApiException e) {
            // The ApiException status code indicates the detailed failure reason.
            // Please refer to the GoogleSignInStatusCodes class reference for more information.
            Log.w("Google Sign In Error", "signInResult:failed code=" + e.getStatusCode());
            Toast.makeText(loginactivity.this, "Failed", Toast.LENGTH_LONG).show();}  }
```

```
    @Override
    protected void onStart() {
        // Check for existing Google Sign In account, if the user is already signed in
        // the GoogleSignInAccount will be non-null.
        GoogleSignInAccount account = GoogleSignIn.getLastSignedInAccount(this);
        if(account != null) {
            startActivity(new Intent(loginactivity.this, userinfo.class));
        }super.onStart();}}
```

Ndtv.java

```
public class ndtv extends AppCompatActivity {
    private WebView webView;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_ndtv);
        webView = (WebView) findViewById(R.id.webView);
        webView.setWebViewClient(new WebViewClient());
        webView.loadUrl("http://www.ndtv.com/video/live/channel/ndtv24x7");
        WebSettings webSettings = webView.getSettings();
        webSettings.setJavaScriptEnabled(true);}
    @Override
    public void onBackPressed() {
        if(webView.canGoBack()){
            webView.goBack();
        }else {super.onBackPressed();  }}}
```

Mainmenu.xml

```
<menu 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"
    tools:context="com.example.newsapp11.MainActivity">
    <item
        android:id="@+id/action_search"
        android:orderInCategory="101"
        android:icon="@drawable/ic_search"
        app:showAsAction="always"
        app:actionViewClass="androidx.appcompat.widget.SearchView"
        android:title="Search"
        android:theme="@android:style/Theme.Holo.Light.DarkActionBar"/>
    <item android:id="@+id/live"
        android:title="LIVE CHANNELS"/>
    <item android:id="@+id/profile"
        android:title="PROFILE"/>
    <item android:id="@+id/otherlogin"
        android:title="LOGIN WITH OTHER ACCOUNT"/>
    <item android:id="@+id/about"
        android:title="ABOUT"/>
    <item android:id="@+id/contact"
        android:title="CONTACT US"/></menu>
```

items.xml

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout
    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:orientation="vertical"
android:layout_width="match_parent"
android:layout_height="350dp">
<androidx.cardview.widget.CardView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="10dp"
    android:layout_marginTop="6dp"
    android:layout_marginRight="10dp"
    android:layout_marginBottom="6dp"
    app:cardCornerRadius="10dp"
    app:cardElevation="@dimen/cardview_default_elevation"
    android:focusable="true">
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="?android:attr/selectableItemBackground">
        <ImageView
            android:id="@+id/img"
            android:layout_width="match_parent"
            android:layout_height="200dp"
            android:scaleType="centerCrop"
            android:transitionName="img"
            tools:ignore="UnusedAttribute" />
        <ImageView
            android:id="@+id/shadow_bottom"
            android:layout_width="match_parent"
```

```
    android:layout_height="80dp"
    android:layout_alignBottom="@+id/img"
    android:src="@drawable/bottom_shadow" />
<ProgressBar
    android:id="@+id/progress_load"
    style="@android:style/Widget.ProgressBar.Small"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="70dp" />
<TextView
    android:id="@+id/author"
    android:layout_width="match_parent"
    android:layout_height="30dp"
    android:layout_alignStart="@+id/title"
    android:layout_alignLeft="@+id/title"
    android:layout_alignTop="@+id/layoutDate"
    android:layout_alignEnd="@+id/layoutDate"
    android:layout_alignRight="@+id/layoutDate"
    android:layout_marginRight="160dp"
    android:drawablePadding="10dp"
    android:ellipsize="end"
    android:gravity="bottom"
    android:maxLines="1"
    android:singleLine="true"
    android:text="Author"
    android:textColor="@color/colorBackground" />
<FrameLayout
```

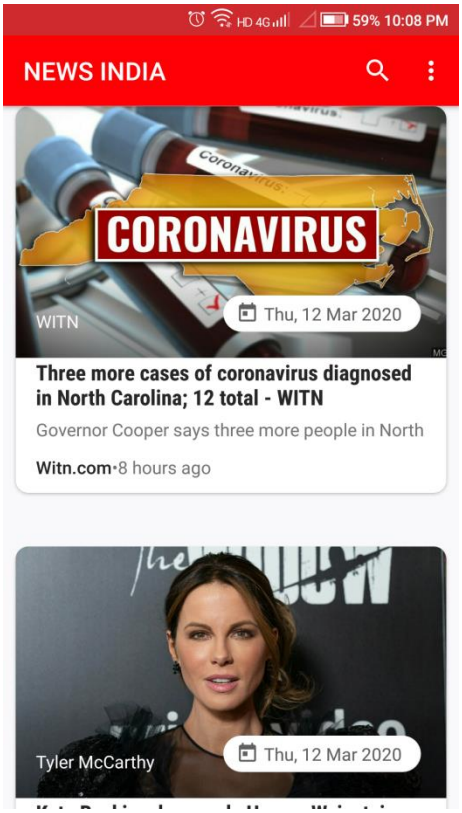
```
    android:id="@+id/layoutDate"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/img"
    android:layout_alignParentRight="true"
    android:layout_marginTop="-50dp"
    android:layout_marginRight="20dp"
    android:background="@drawable/round_white"
    android:padding="5dp">
    <ImageView
        android:layout_width="18dp"
        android:layout_height="18dp"
        android:layout_marginLeft="5dp"
        android:layout_marginRight="5dp"
        android:src="@drawable/ic_date" />
    <TextView
        android:id="@+id/publishedAt"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="27dp"
        android:layout_marginRight="10dp"
        android:text="01 January 1990"
        android:textColor="#606060" />
</FrameLayout>
<TextView
    android:id="@+id/title"
    android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content"
    android:layout_below="@+id/img"
    android:layout_marginLeft="16dp"
    android:layout_marginRight="16dp"
    android:fontFamily="sans-serif-condensed-light"
    android:text="Title"
    android:textColor="@color/colorTextTitle"
    android:textSize="17sp"
    android:textStyle="bold" />
<TextView
    android:id="@+id/desc"
    android:layout_width="wrap_content"
    android:layout_height="20dp"
    android:layout_below="@+id/title"
    android:layout_marginLeft="16dp"
    android:layout_marginTop="5dp"
    android:layout_marginRight="16dp"
    android:text="desc" />
<TextView
    android:id="@+id/source"
    android:layout_width="wrap_content"
    android:layout_height="20dp"
    android:layout_below="@+id/desc"
    android:layout_marginLeft="16dp"
    android:layout_marginTop="10dp"
    android:layout_marginBottom="10dp"
    android:drawablePadding="10dp"
```

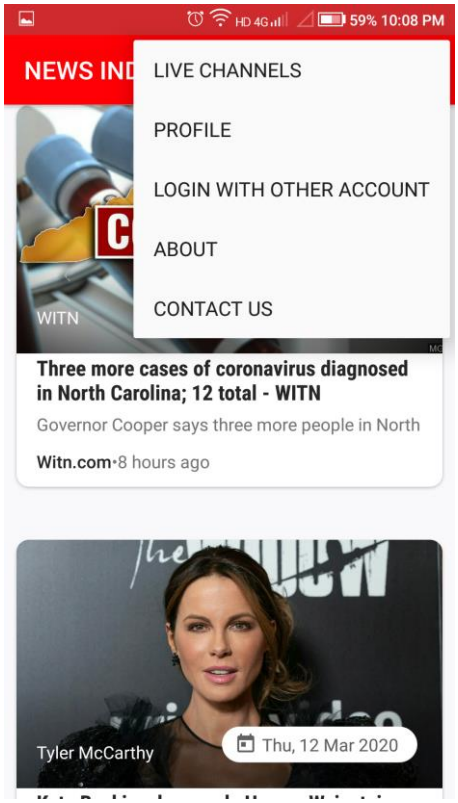
```
            android:ellipsize="end"
            android:fontFamily="sans-serif-light"
            android:maxLines="1"
            android:singleLine="true"
            android:text="Source"
            android:textColor="@color/colorTextTitle"
            android:textStyle="bold" />
        <TextView
            android:id="@+id/time"
            android:layout_width="wrap_content"
            android:layout_height="20dp"
            android:layout_below="@+id/desc"
            android:layout_marginTop="10dp"
            android:layout_marginBottom="10dp"
            android:layout_toRightOf="@+id/source"
            android:drawablePadding="10dp"
            android:ellipsize="end"
            android:maxLines="1"
            android:singleLine="true"
            android:text="Time" />
      </RelativeLayout>
   </androidx.cardview.widget.CardView>
</FrameLayout>
```

5.2 Screen Layouts

1: SPLASH SCREEN

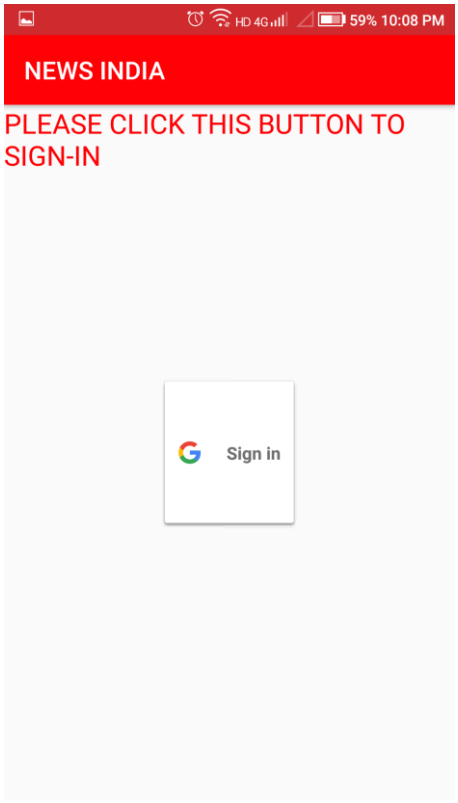


2:NEWS HEADLINES



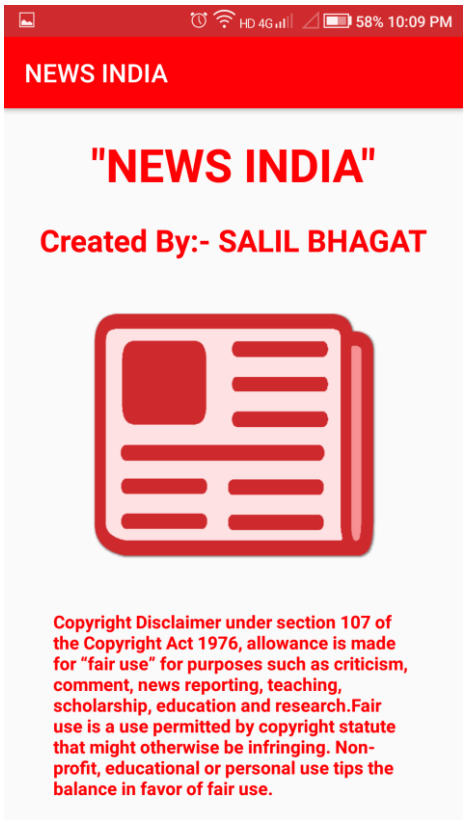
3:OPTIONS

4:LIVE CHANNELS



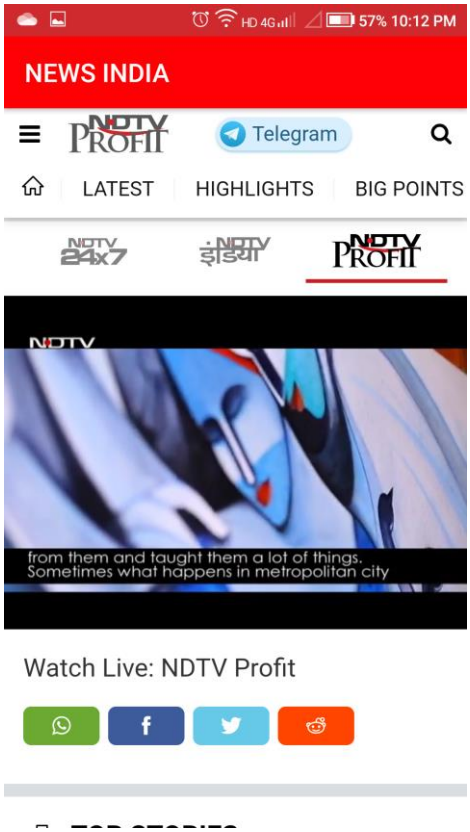
5:SIGN-IN BUTTON

6:PROFILE

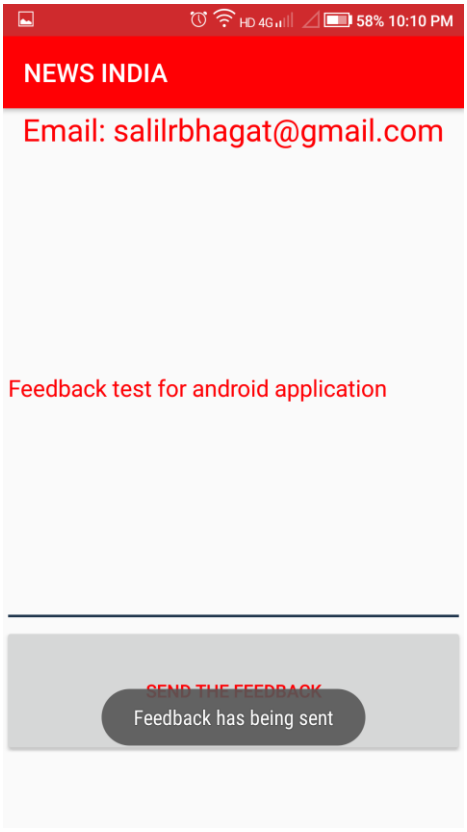


7:ABOUT

8:ARTICLE



9:LIVE NEWS



10:FEEDBACK SYSTEM

CHAP 6: FUTURE ENHANCEMENT

6. FUTURE ENHANCEMENT

If there is any rectification or enhancement proposed by the user, the application will be modified and the desired module will be implemented into current application. Future enhancements include developing the following as stated below:

- Registered user can modify the news according to its use
- Notification function
- Personalise display
- Enhanced ui
- More integration with the user information
- To update the application as per the user's demands.
- To include payment module.
- AI based user behaviour detection

The above mentioned points are the enhancements which can be done increase the applicability and usage of this project.

CHAP 7: REFRENCES AND BIBLIOGRAPHY

7. REFRENCES AND BIBLIOGRAPHY

Website:

- https://www.youtube.com
- https://www.google.com
- https://simplifiedcoding.com
- https://www.tutorialspoint.com

Books:

- Android Programming for beginners