CSEE5590/CS490 APS
Programming for Web/Cloud / Mobile Applications

# Part - Mobile Development

# Lab
# ASSIGNMENT - 1

BALACHANDAR KULALA

**(Student ID:12543027)**

# Goal of the Assignment:

To provide good practice on fetching data from APIs, parsing the JSON response, using Async task, updating information regularly, handling errors, creating a customized list and other views and layouts

## Part1: Login, Signup and Social Login Activities

**Aim:**

To understand the creating login, register pages and exploring the feature of social login.

**Description of Work:**

Design a mobile application that contains the login page, registration page and provide the social login.

### Technical Section:

Majorly used technical skills.

- Android technology
- Core Java
- XML
- JSON

### Development Section

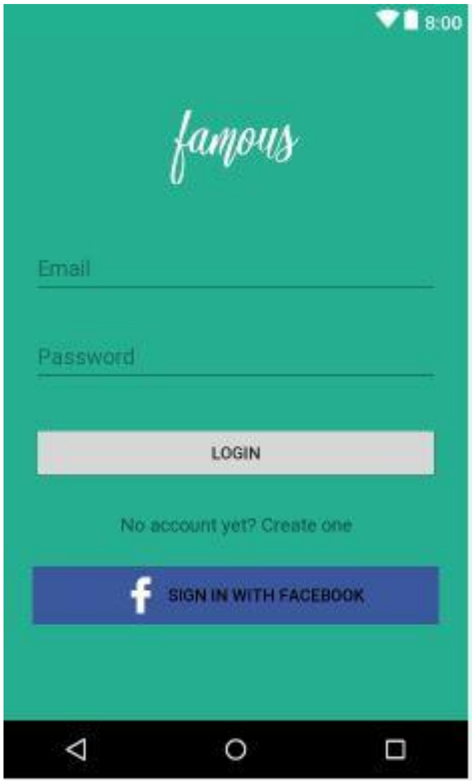
This Application contains mainly 3 pages:

1. LogIn Page:

   This starting page of the application, this page displays the option for user to enter user name and password for login.

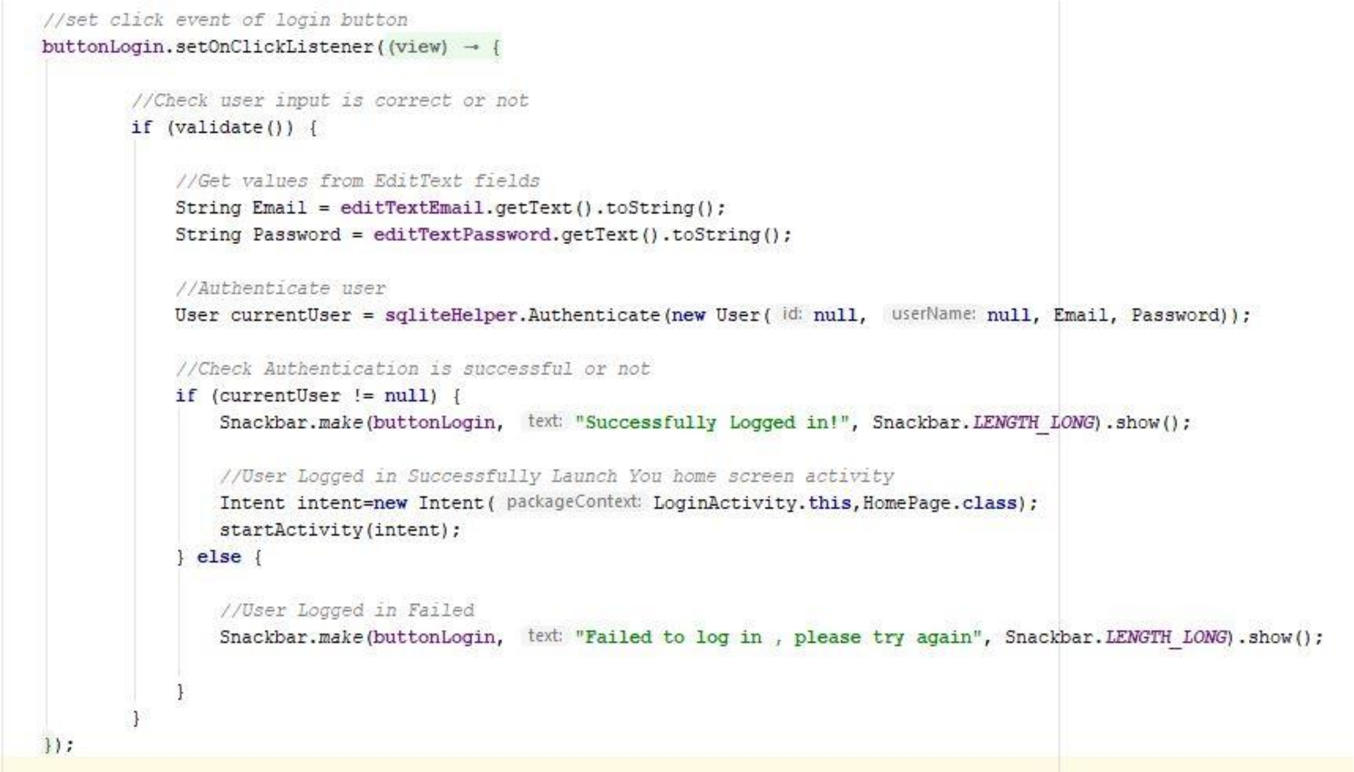
   This page also provides facility to user to login with Facebook account into this application. And Create new user account.

Screenshot:



The user enters the UserName and Password and clicks on LOGIN button.

The corresponding code for this logic is shown in following screen:
It authenticates the user credentials, if it exists allows to the home page else denies.



```
//set click event of login button
buttonLogin.setOnClickListener((view) → {

        //Check user input is correct or not
        if (validate()) {

            //Get values from EditText fields
            String Email = editTextEmail.getText().toString();
            String Password = editTextPassword.getText().toString();

            //Authenticate user
            User currentUser = sqliteHelper.Authenticate(new User( id: null,  userName: null, Email, Password));

            //Check Authentication is successful or not
            if (currentUser != null) {
                Snackbar.make(buttonLogin,  text: "Successfully Logged in!", Snackbar.LENGTH_LONG).show();

                //User Logged in Successfully Launch You home screen activity
                Intent intent=new Intent( packageContext: LoginActivity.this,HomePage.class);
                startActivity(intent);
            } else {

                //User Logged in Failed
                Snackbar.make(buttonLogin,  text: "Failed to log in , please try again", Snackbar.LENGTH_LONG).show();

            }
        }
});
```

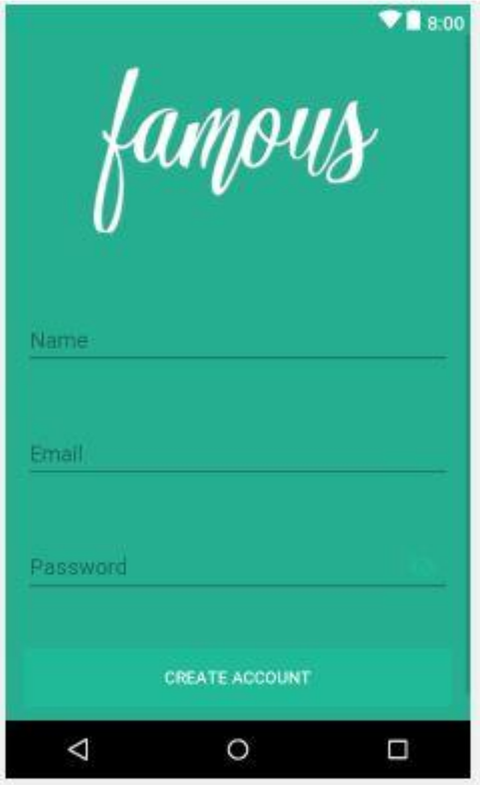
ii) Registration Page:

If user doesn’t have an account, allows to create new account in this page.

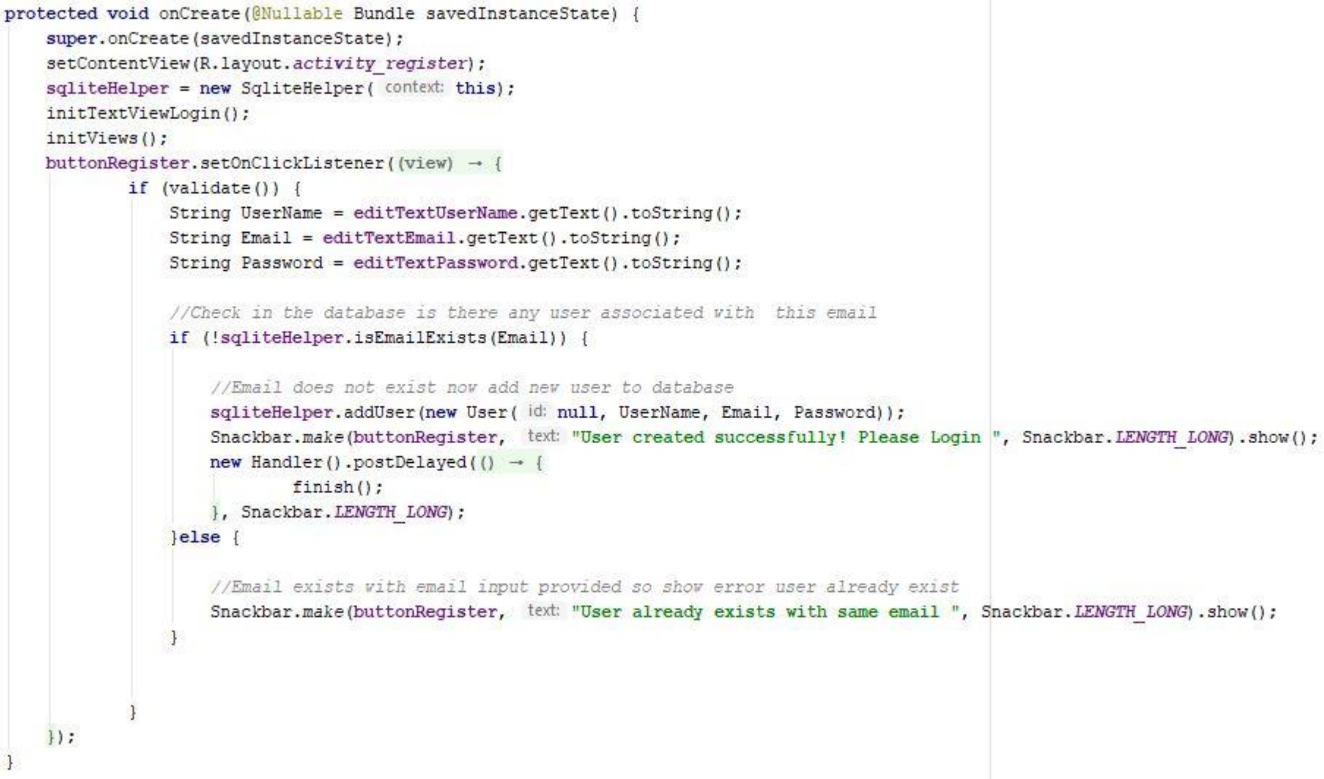
This page asks user to enter name, email and password details.

Once user clicks on the Create Account button, it redirects to the Log in page for logging into the account.

Screenshot:



The code snippet for this registration page is shown below:

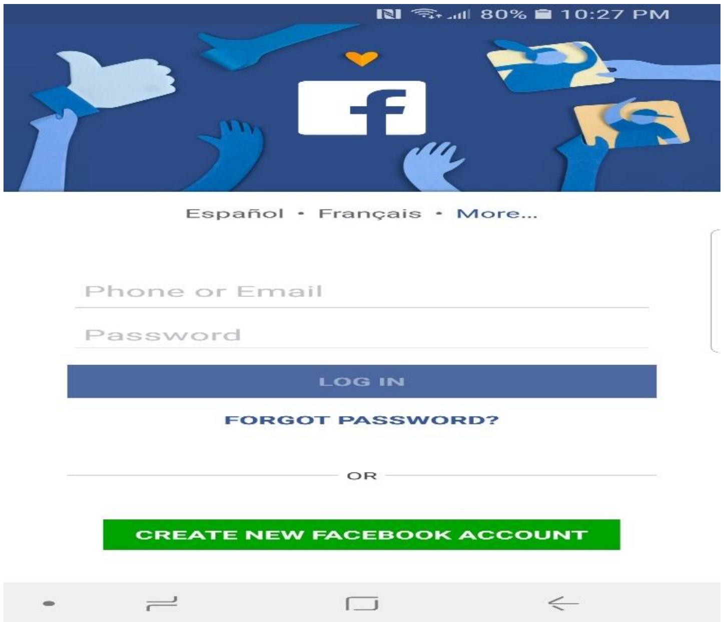


```
protected void onCreate(@Nullable Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_register);
    sqliteHelper = new SqliteHelper( context: this);
    initTextViewLogin();
    initViews();
    buttonRegister.setOnClickListener((view) → {
            if (validate()) {
                String UserName = editTextUserName.getText().toString();
                String Email = editTextEmail.getText().toString();
                String Password = editTextPassword.getText().toString();

                //Check in the database is there any user associated with  this email
                if (!sqliteHelper.isEmailExists(Email)) {

                    //Email does not exist now add new user to database
                    sqliteHelper.addUser(new User( id: null, UserName, Email, Password));
                    Snackbar.make(buttonRegister,  text: "User created successfully! Please Login ", Snackbar.LENGTH_LONG).show();
                    new Handler().postDelayed(() → {
                            finish();
                    }, Snackbar.LENGTH_LONG);
                }else {

                    //Email exists with email input provided so show error user already exist
                    Snackbar.make(buttonRegister,  text: "User already exists with same email ", Snackbar.LENGTH_LONG).show();
                }


            }
    });
}
```

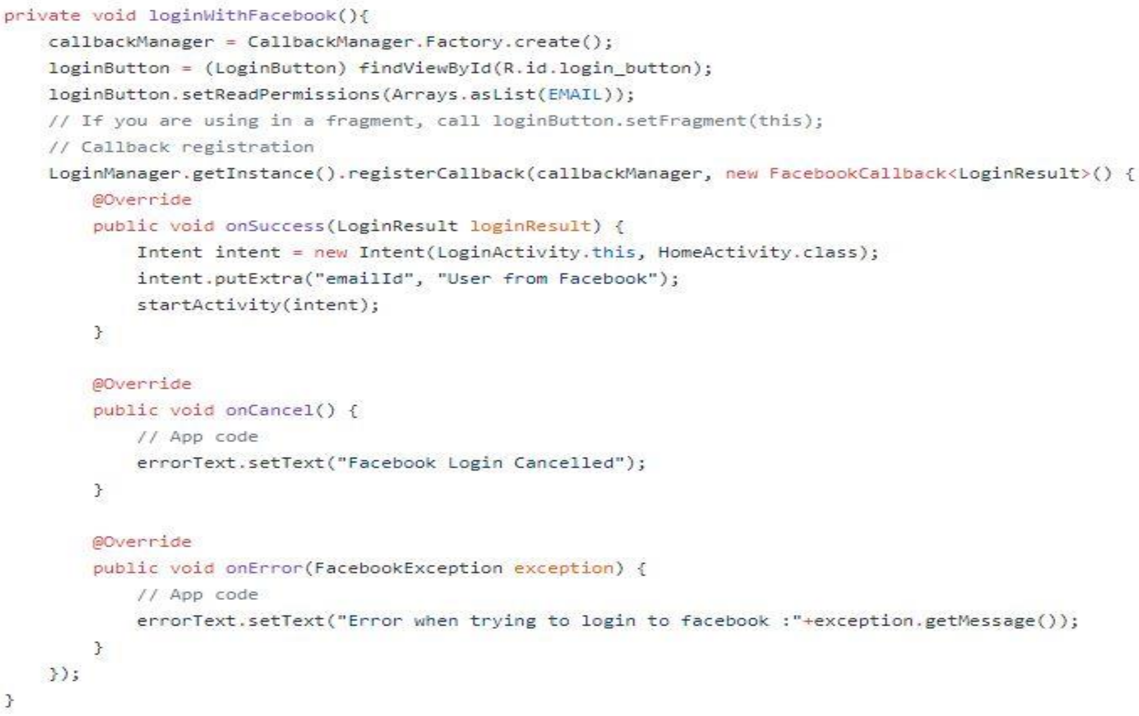
iii) Social LogIn:

It allows user to use the application by using social login feature. This can be accomplished here by using the facebook login.

Screenshot:



The corresponding logic is shown in below snippet:



```
private void loginWithFacebook(){
    callbackManager = CallbackManager.Factory.create();
    loginButton = (LoginButton) findViewById(R.id.login_button);
    loginButton.setReadPermissions(Arrays.asList(EMAIL));
    // If you are using in a fragment, call loginButton.setFragment(this);
    // Callback registration
    LoginManager.getInstance().registerCallback(callbackManager, new FacebookCallback<LoginResult>() {
        @Override
        public void onSuccess(LoginResult loginResult) {
            Intent intent = new Intent(LoginActivity.this, HomeActivity.class);
            intent.putExtra("emailId", "User from Facebook");
            startActivity(intent);
        }

        @Override
        public void onCancel() {
            // App code
            errorText.setText("Facebook Login Cancelled");
        }

        @Override
        public void onError(FacebookException exception) {
            // App code
            errorText.setText("Error when trying to login to facebook :"+exception.getMessage());
        }
    });
}
```

# Part2: Utilizing phone hardware features

**Aim:**

To gain the knowledge in exploring the usage of phone hardware features like sensors, Camera, Maps, scanners etc.

**Description of Work:**

Design a mobile application that utilizes the hardware parts of the mobile phone such as Camera, sensors, Maps. In this case we developed the application using Camera and google maps.

## Technical Section:

Majorly used technical skills.

- Android technology
- Core Java
- XML
- JSON

## Development of Work:

It is developed by using two hardware features of the phone, so that we created the two activities for this task.

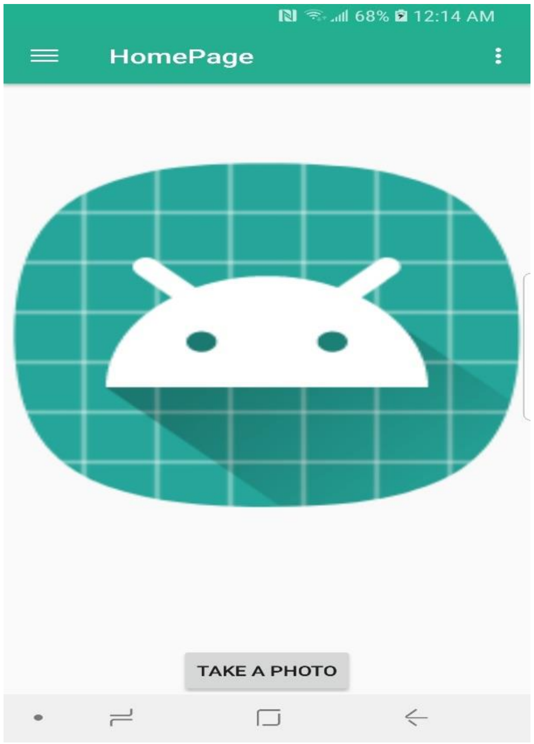
Task1: Camera Usage:

This section uses the basic usage of camera feature of the android phone. It captures the phone and display the image once it is done.

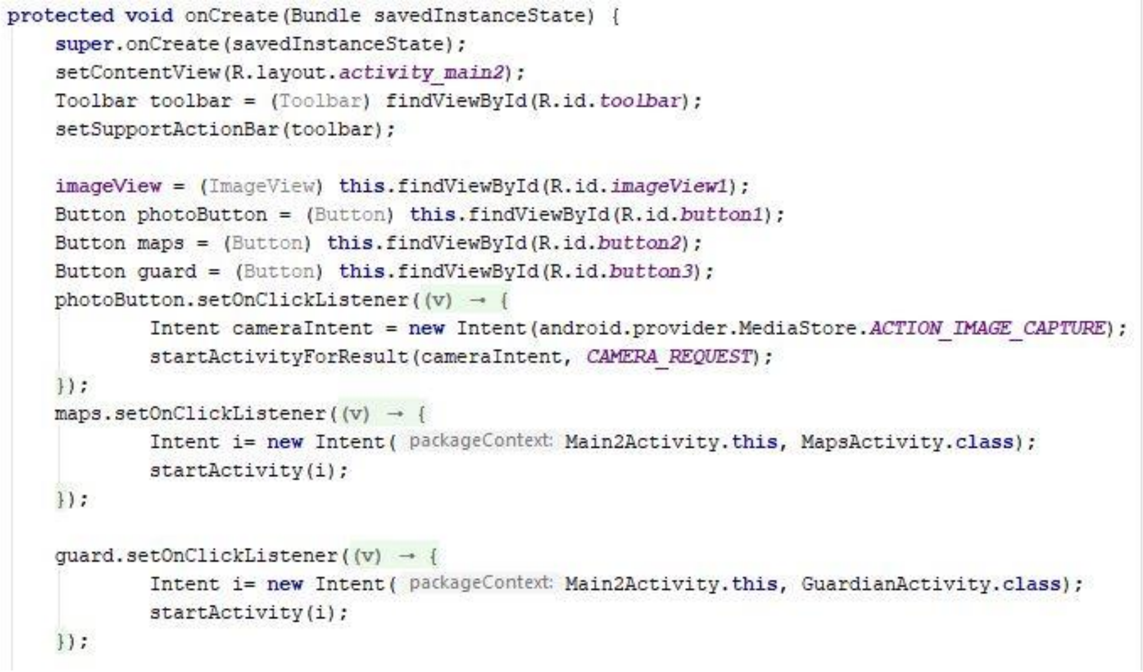
The following two screens shows the design view and Code snippet for that logic.

Design view:

When the user clicks on the “Take A photo” button, camera starts capturing.

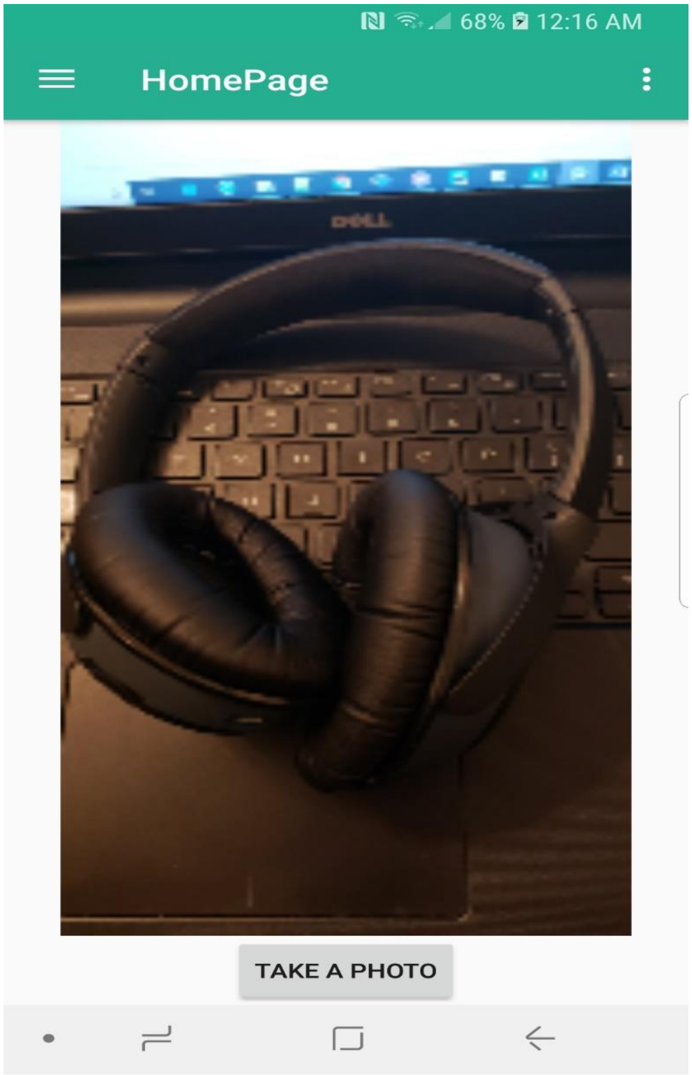


The corresponding logic is shown below.



```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main2);
    Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);

    imageView = (ImageView) this.findViewById(R.id.imageView1);
    Button photoButton = (Button) this.findViewById(R.id.button1);
    Button maps = (Button) this.findViewById(R.id.button2);
    Button guard = (Button) this.findViewById(R.id.button3);
    photoButton.setOnClickListener((v) → {
            Intent cameraIntent = new Intent(android.provider.MediaStore.ACTION_IMAGE_CAPTURE);
            startActivityForResult(cameraIntent, CAMERA_REQUEST);
    });
    maps.setOnClickListener((v) → {
            Intent i= new Intent( packageContext: Main2Activity.this, MapsActivity.class);
            startActivity(i);
    });

    guard.setOnClickListener((v) → {
            Intent i= new Intent( packageContext: Main2Activity.this, GuardianActivity.class);
            startActivity(i);
    });
```

The output screen shown below.

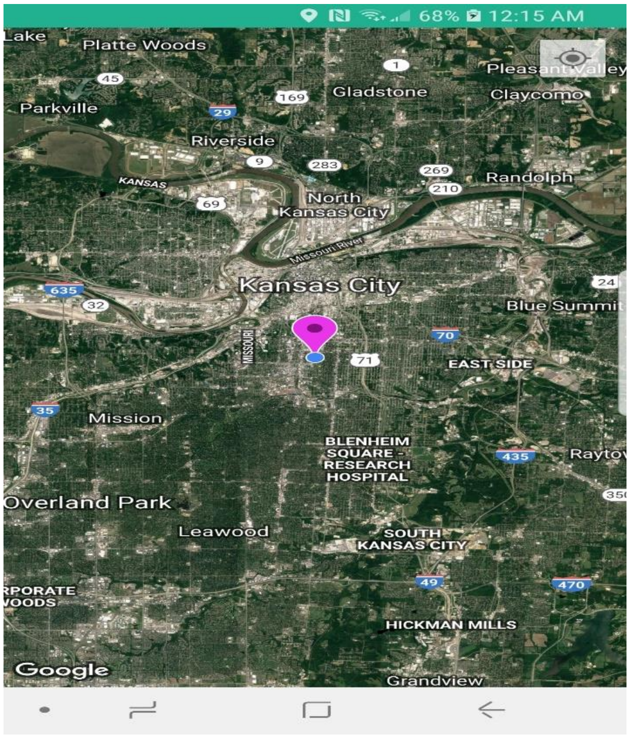


## Task2: Google Maps:

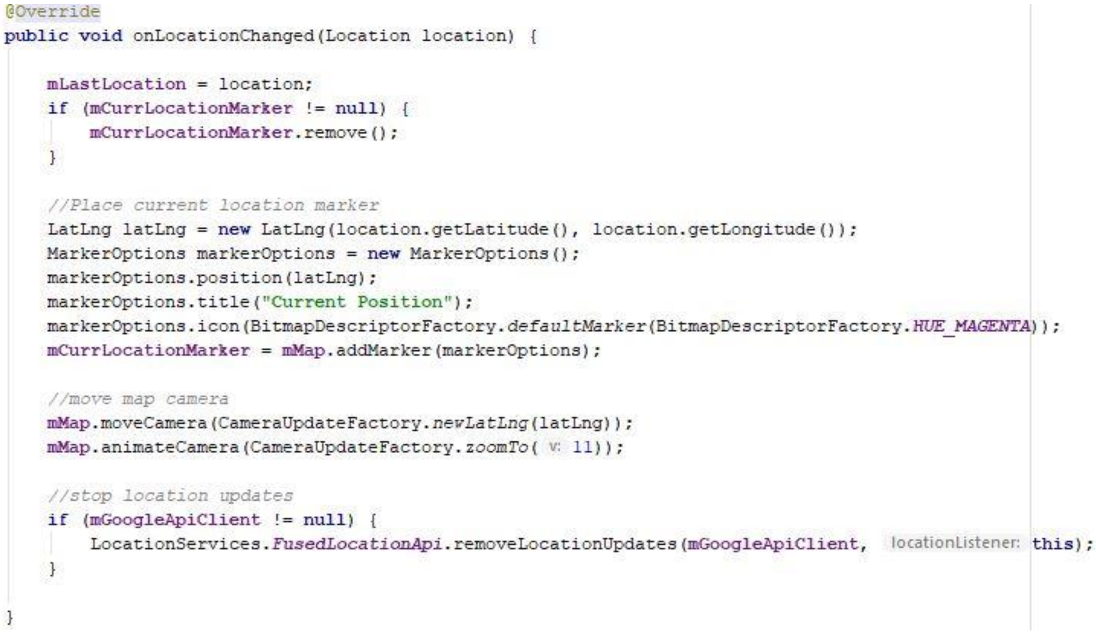
This section uses the google maps feature of the android phone. When the user clicks on the Google Maps button it displays the current location of the user.

The following two screens displays the Output screen and Functional logic of implementation.

Output screen:



The corresponding code is shown below:



```
@Override
public void onLocationChanged(Location location) {

    mLastLocation = location;
    if (mCurrLocationMarker != null) {
        mCurrLocationMarker.remove();
    }

    //Place current location marker
    LatLng latLng = new LatLng(location.getLatitude(), location.getLongitude());
    MarkerOptions markerOptions = new MarkerOptions();
    markerOptions.position(latLng);
    markerOptions.title("Current Position");
    markerOptions.icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_MAGENTA));
    mCurrLocationMarker = mMap.addMarker(markerOptions);

    //move map camera
    mMap.moveCamera(CameraUpdateFactory.newLatLng(latLng));
    mMap.animateCamera(CameraUpdateFactory.zoomTo( v: 11));

    //stop location updates
    if (mGoogleApiClient != null) {
        LocationServices.FusedLocationApi.removeLocationUpdates(mGoogleApiClient,  locationListener: this);
    }

}
```

# Part3: Utilizing Machine Learning Features

**Aim:**

To gain the knowledge in exploring machine learning APIs in android to implement the machine learning activities.

**Description of Work:**

Design a mobile application that utilizes Machine learning APIs such as AWS, Google ML and Watson APIs in android application to develop any Machine learning activity.

## Technical Section:

Majorly used technical skills.

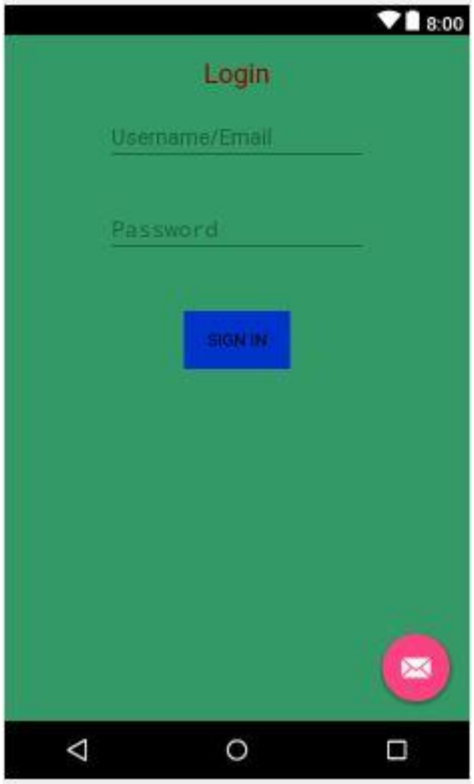
- Android technology
- Core Java
- XML
- JSON
- Using WEB APIs (Watson, AWS, Google ML)

## Development of Work:

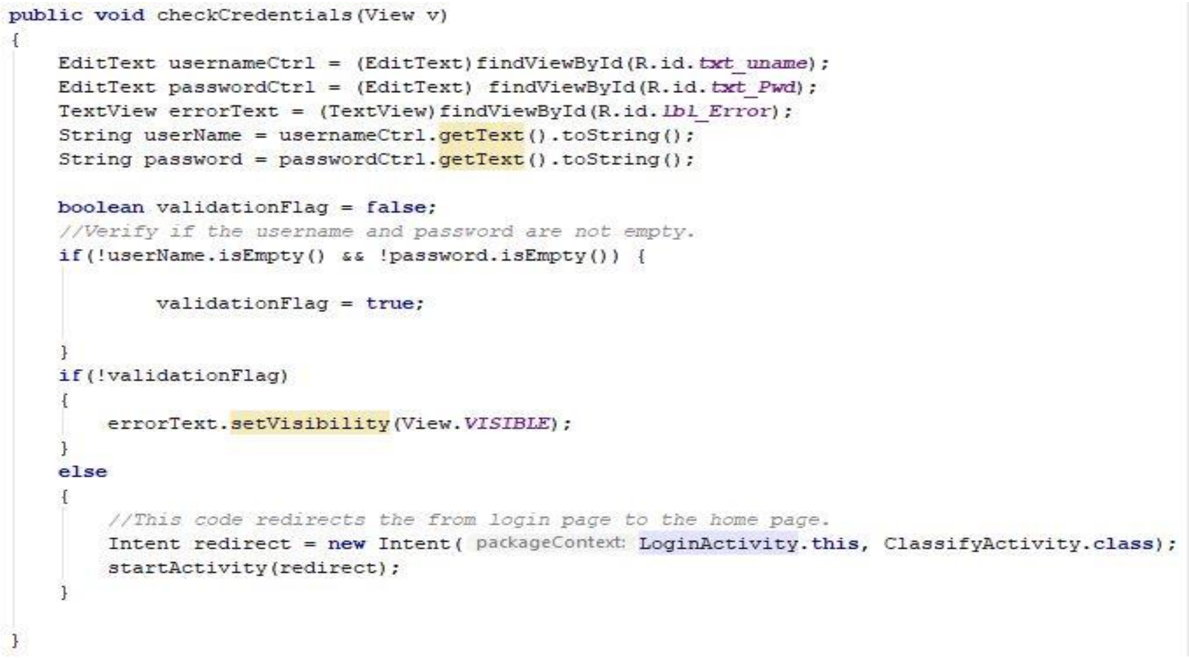
Here we developed an application that checks the entered text is either positive gesture or negative gesture. We have used ML API to detect this activity.

The following page is the login page:

When the user enters User name and password and clicks on login it takes him to the main page of the activity.



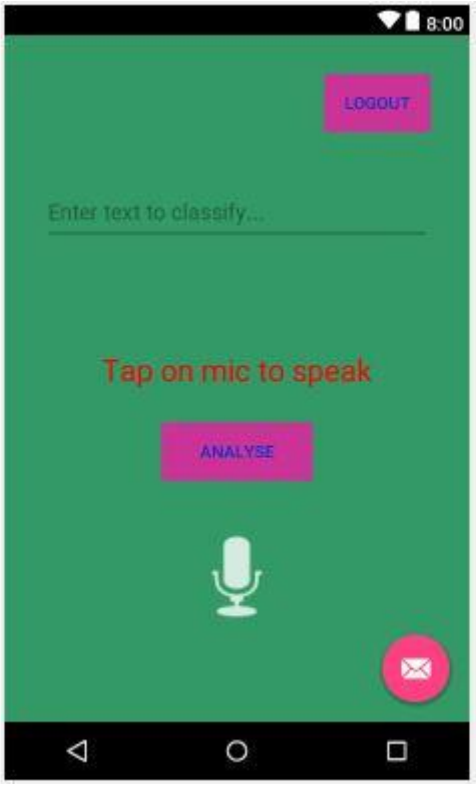
The corresponding Logic for this activity is shown below.



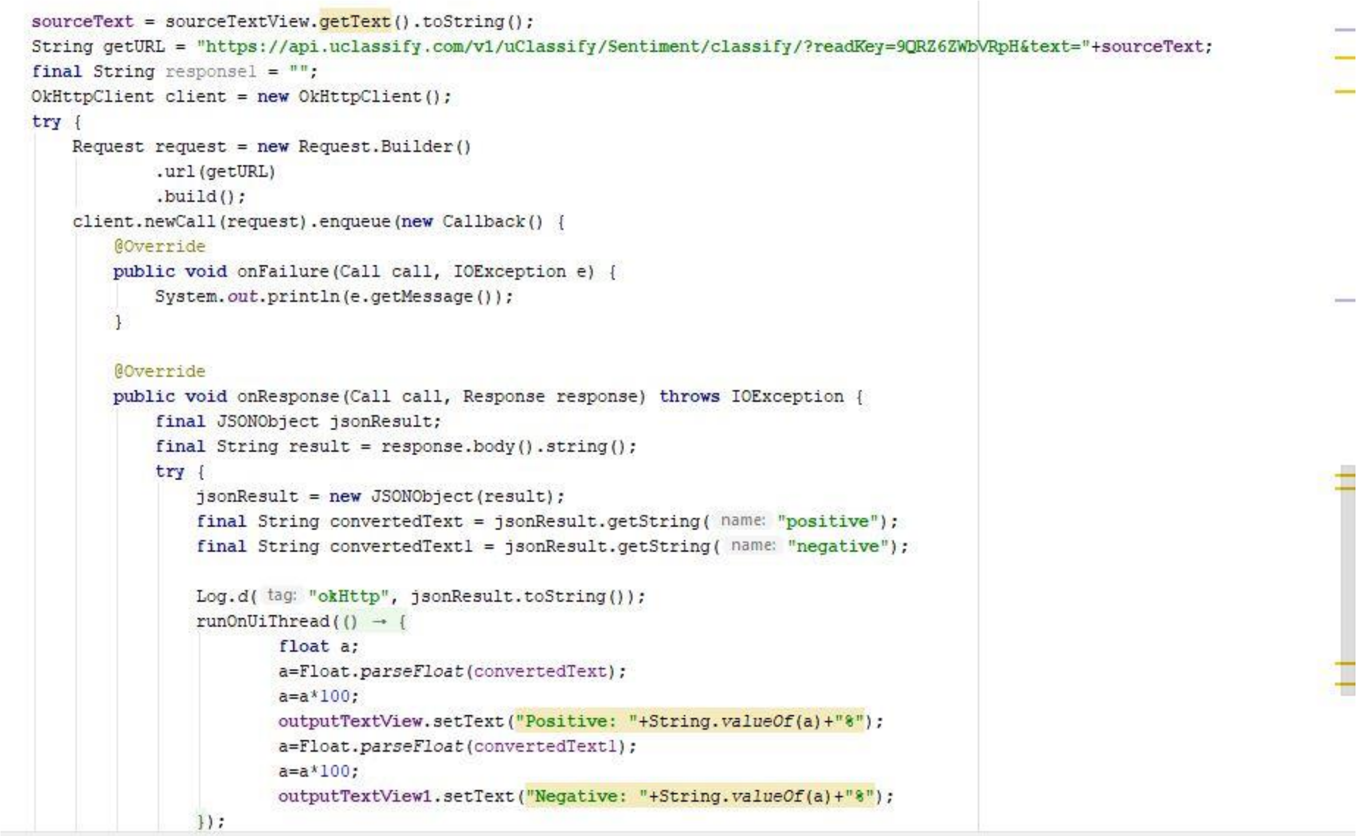
```
public void checkCredentials(View v)
{
    EditText usernameCtrl = (EditText)findViewById(R.id.txt_uname);
    EditText passwordCtrl = (EditText) findViewById(R.id.txt_Pwd);
    TextView errorText = (TextView)findViewById(R.id.lbl_Error);
    String userName = usernameCtrl.getText().toString();
    String password = passwordCtrl.getText().toString();

    boolean validationFlag = false;
    //Verify if the username and password are not empty.
    if(!userName.isEmpty() && !password.isEmpty()) {

            validationFlag = true;

    }
    if(!validationFlag)
    {
        errorText.setVisibility(View.VISIBLE);
    }
    else
    {
        //This code redirects the from login page to the home page.
        Intent redirect = new Intent( packageContext: LoginActivity.this, ClassifyActivity.class);
        startActivity(redirect);
    }

}
```

When the user credentials are verified successfully it allows him to take to the main page of the application.

Main page of the activity is shown below.

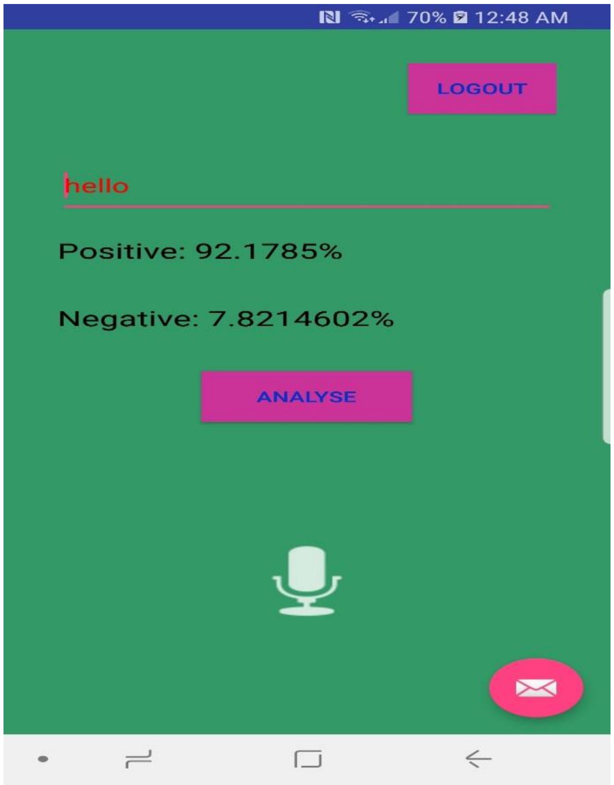


Here user can either enter the text in textbox or can use the mic to speak. The spoken words are displayed in the text box.
By clicking the Analyze button it displays the Positivity or negativity of the code.
When the user wishes to logout from the screen, he clicks on the LOGOUT button.
The corresponding logic is shown below:



```
sourceText = sourceTextView.getText().toString();
String getURL = "https://api.uclassify.com/v1/uClassify/Sentiment/classify/?readKey=9QRZ6ZWbVRpH&text="+sourceText;
final String response1 = "";
OkHttpClient client = new OkHttpClient();
try {
    Request request = new Request.Builder()
            .url(getURL)
            .build();
    client.newCall(request).enqueue(new Callback() {
        @Override
        public void onFailure(Call call, IOException e) {
            System.out.println(e.getMessage());
        }

        @Override
        public void onResponse(Call call, Response response) throws IOException {
            final JSONObject jsonResult;
            final String result = response.body().string();
            try {
                jsonResult = new JSONObject(result);
                final String convertedText = jsonResult.getString( name: "positive");
                final String convertedText1 = jsonResult.getString( name: "negative");

                Log.d( tag: "okHttp", jsonResult.toString());
                runOnUiThread(() → {
                        float a;
                        a=Float.parseFloat(convertedText);
                        a=a*100;
                        outputTextView.setText("Positive:  "+String.valueOf(a)+"%");
                        a=Float.parseFloat(convertedText1);
                        a=a*100;
                        outputTextView1.setText("Negative:  "+String.valueOf(a)+"%");
                });
```

The output screen is displayed below:



Positive: 92.1785%

Negative: 7.8214602%

Here the entered word is “hello”, when user clicks on the Analyze button, it displays the positivity and negativity of the text.

# Part4: Displaying the Latest NEWS using Guardian API

**Aim:**

To gain the knowledge by exploring Guardian API to get the latest new about the entered text.

**Description of Work:**

Design a mobile application that utilizes Guardian API to get the latest new about the entered text.

## Technical Section:

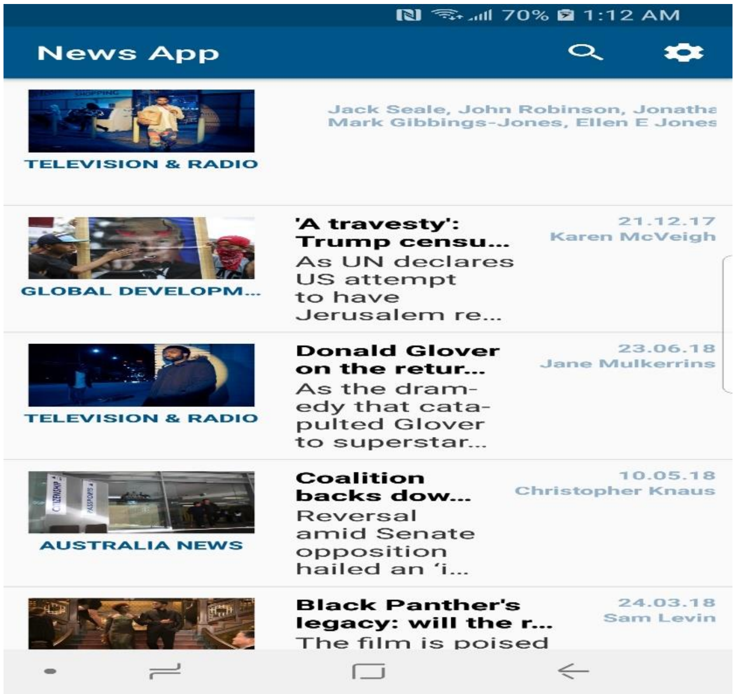
Majorly used technical skills.

- Android technology
- Core Java
- XML
- JSON
- Guardian API

## Development of Work:

This application uses the Guardian API to get the latest news about the entered text.

The app contains one home scree, this screen is shown below:



70% 1:12 AM

News App

TELEVISION & RADIO

Jack Seale, John Robinson, Jonatha
Mark Gibbings-Jones, Ellen E Jones

GLOBAL DEVELOPM...

'A travesty': Trump censu...
As UN declares US attempt to have Jerusalem re...

21.12.17
Karen McVeigh

TELEVISION & RADIO

Donald Glover on the retur...
As the dram-edy that cata-pulted Glover to superstar...

23.06.18
Jane Mulkerrins

AUSTRALIA NEWS

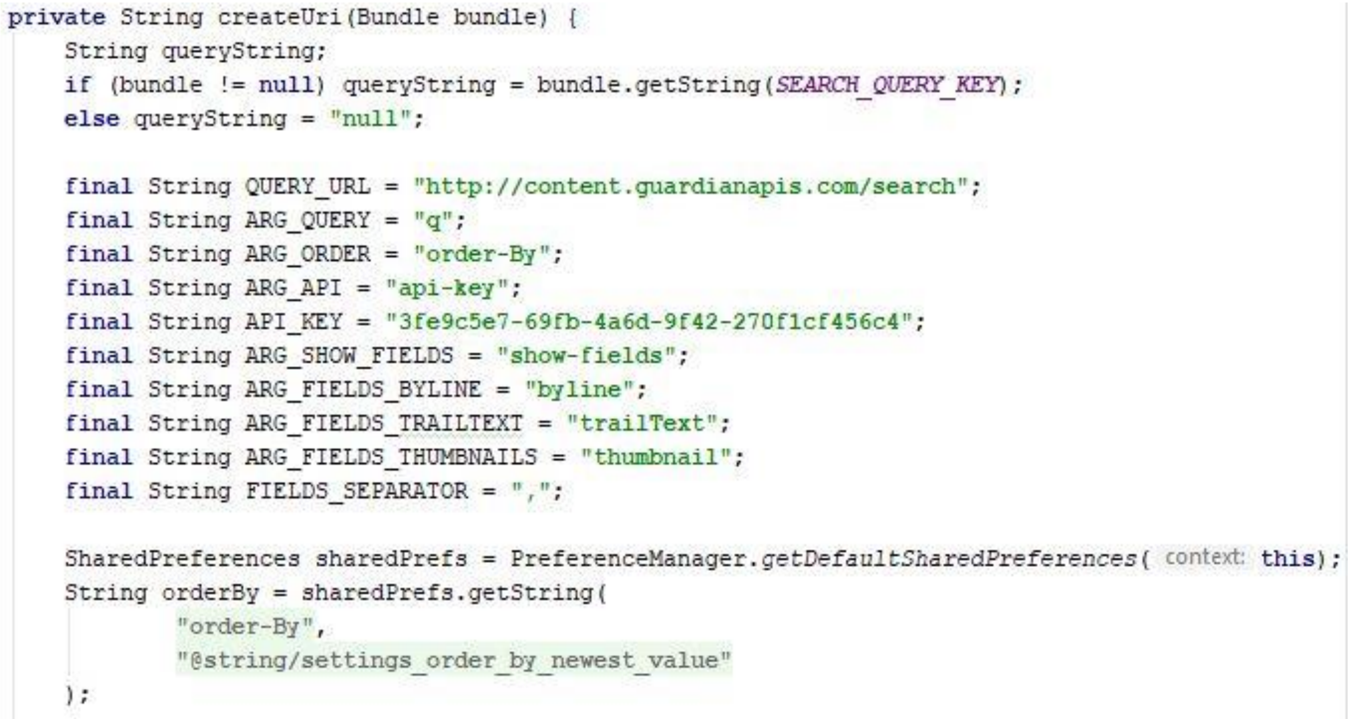
Coalition backs dow...
Reversal amid Senate opposition hailed an 'i...

10.05.18
Christopher Knaus

Black Panther's legacy: will the r...
The film is poised

24.03.18
Sam Levin

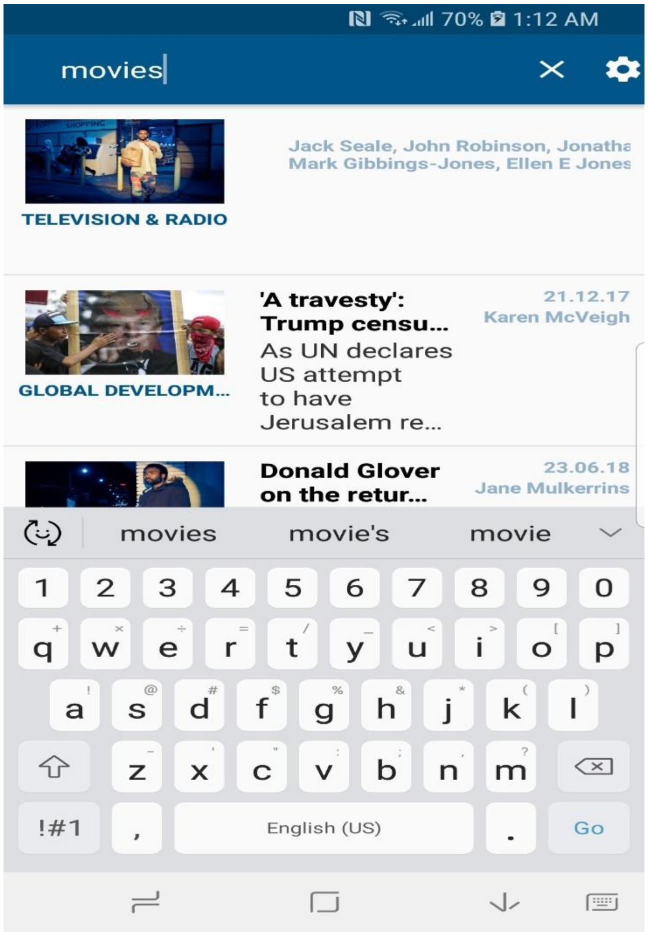
The corresponding Guardian API usage is shown below:



```
private String createUri(Bundle bundle) {
    String queryString;
    if (bundle != null) queryString = bundle.getString(SEARCH_QUERY_KEY);
    else queryString = "null";

    final String QUERY_URL = "http://content.guardianapis.com/search";
    final String ARG_QUERY = "q";
    final String ARG_ORDER = "order-By";
    final String ARG_API = "api-key";
    final String API_KEY = "3fe9c5e7-69fb-4a6d-9f42-270f1cf456c4";
    final String ARG_SHOW_FIELDS = "show-fields";
    final String ARG_FIELDS_BYLINE = "byline";
    final String ARG_FIELDS_TRAILTEXT = "trailText";
    final String ARG_FIELDS_THUMBNAILS = "thumbnail";
    final String FIELDS_SEPARATOR = ",";

    SharedPreferences sharedPrefs = PreferenceManager.getDefaultSharedPreferences( context: this);
    String orderBy = sharedPrefs.getString(
            "order-By",
            "@string/settings_order_by_newest_value"
    );
```

Here the user need to generate the API key to access this API, the complete usage of the guardian API usage is shown in above code.s

The search screen is shown below:



movies

Jack Seale, John Robinson, Jonatha
Mark Gibbings-Jones, Ellen E Jones

TELEVISION & RADIO

'A travesty': Trump censu...
As UN declares US attempt to have Jerusalem re...

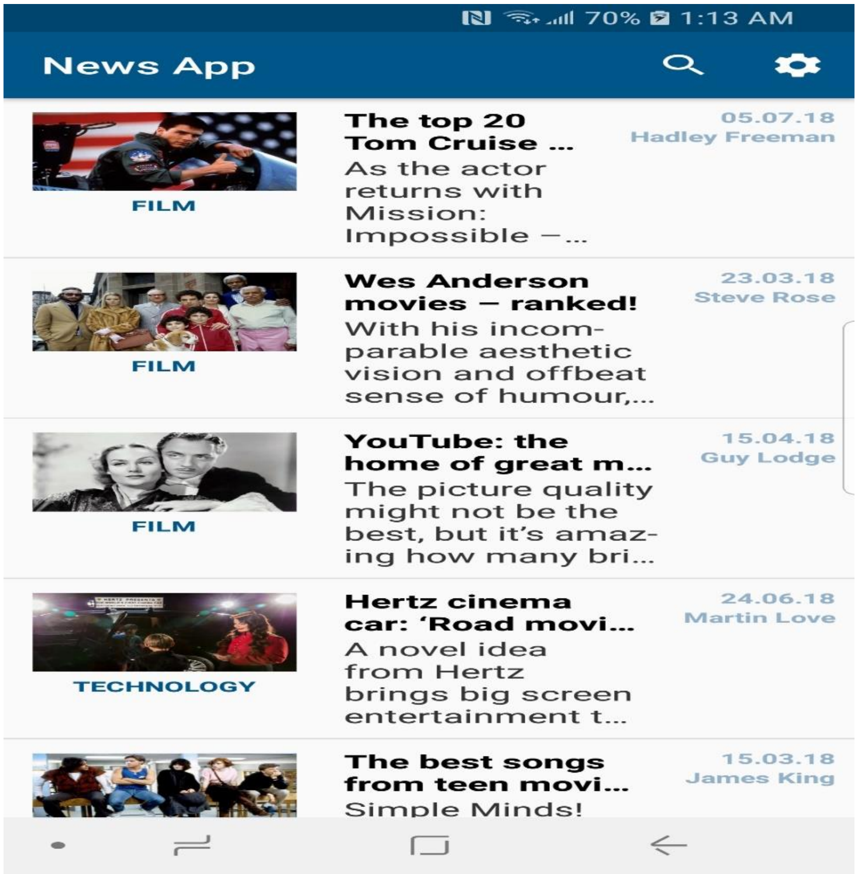
21.12.17
Karen McVeigh

GLOBAL DEVELOPM...

Donald Glover on the retur...

23.06.18
Jane Mulkerrins

Here the user is searching for the news about movies. The search results are shown below.



70% 1:13 AM

News App

The top 20 Tom Cruise ...
As the actor returns with Mission: Impossible –...
05.07.18
Hadley Freeman
FILM

Wes Anderson movies – ranked!
With his incomparable aesthetic vision and offbeat sense of humour,...
23.03.18
Steve Rose
FILM

YouTube: the home of great m...
The picture quality might not be the best, but it's amazing how many bri...
15.04.18
Guy Lodge
FILM

Hertz cinema car: 'Road movi...
A novel idea from Hertz brings big screen entertainment t...
24.06.18
Martin Love
TECHNOLOGY

The best songs from teen movi...
Simple Minds!
15.03.18
James King

## GitHub LINK

https://github.com/bkkhf/Mobilelab1

## DEMO LINK

https://youtu.be/GCPkfDoPVZg

References:

- https://www.w3schools.com
- https://www.udacity.com/
- http://developer.android.com/guide/index.html.

https://www.coursera.org/courses?languages=en&query=android

https://developer.android.com/docs/