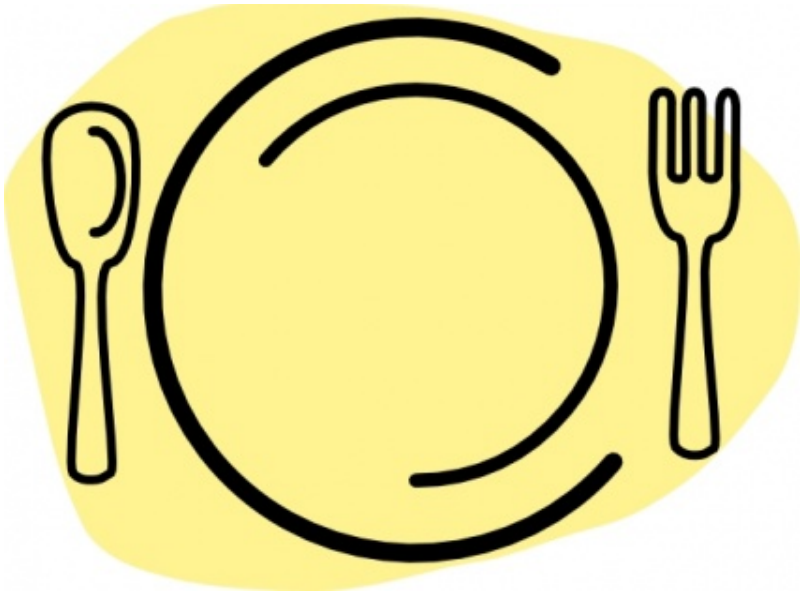
ANDROID PROGRAMMING
CIS 600

FINAL PROJECT REPORT

YUMMY BOT

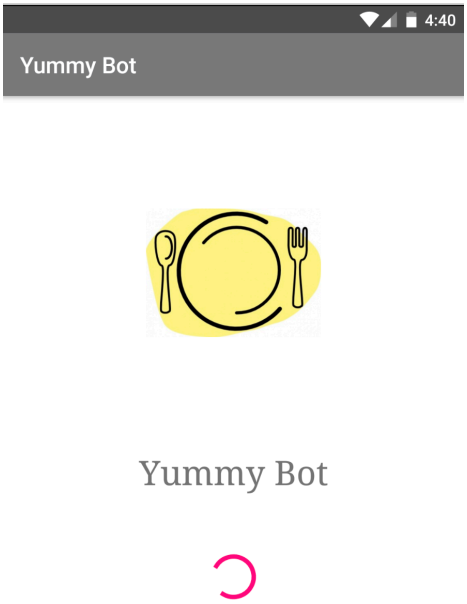


BY

BRAHMACHAITHANYA SADASHIVA
ABHIRAM SRINIVASAN

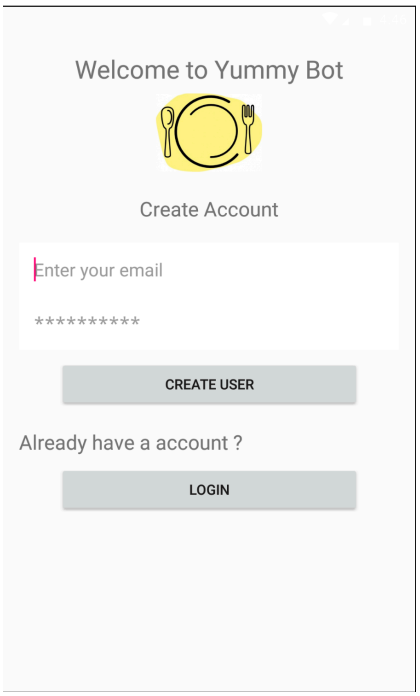
Table of Contents

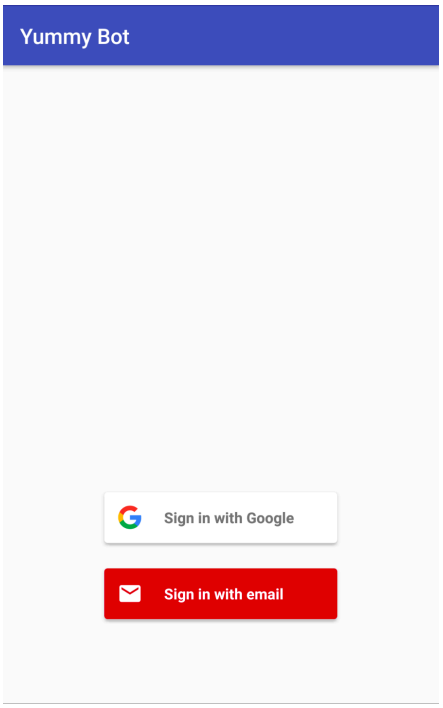
SPLASH SCREEN




```
public class SplashScreen extends AppCompatActivity {

    private final int SPLASH_DISPLAY_LENGTH = 3000;

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle icicle) {
        super.onCreate(icicle);
        setContentView(R.layout.activity_splash_screen);

        /* New Handler to start the Menu-Activity
         * and close this Splash-Screen after some seconds.*/
        new Handler().postDelayed(() -> {
                /* Create an Intent that will start the Menu-Activity. */
                Intent mainIntent = new Intent(SplashScreen.this,LoginActivity.class
                SplashScreen.this.startActivity(mainIntent);
                SplashScreen.this.finish();
        }, SPLASH_DISPLAY_LENGTH);
    }

}
```

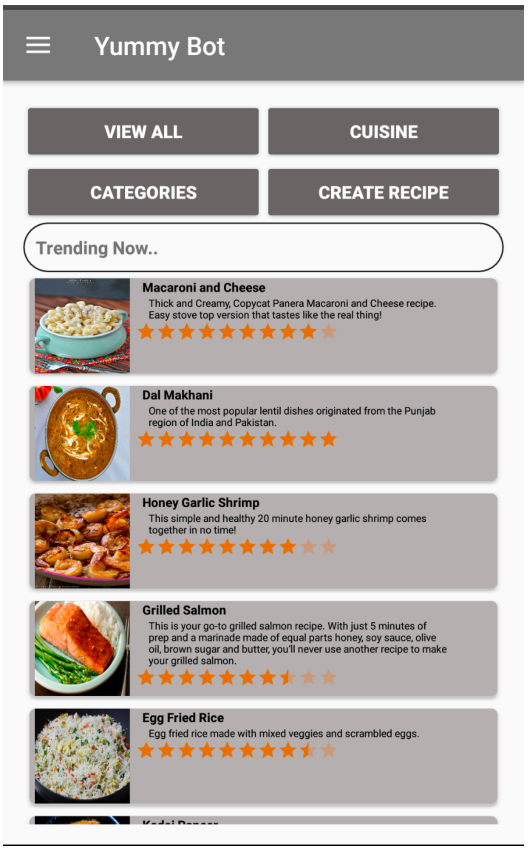
LOGIN ACTIVITY





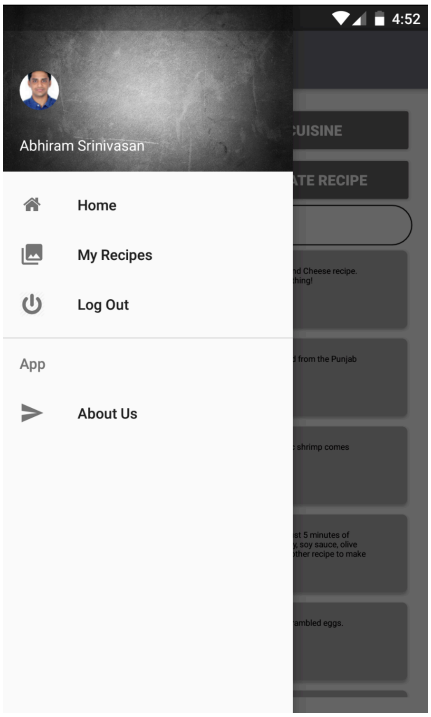

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_login);

    mAuth = FirebaseAuth.getInstance();

    mAuthListener = (AuthStateListener) (firebaseAuth) → {
            FirebaseUser user = firebaseAuth.getCurrentUser();
            if(user!=null){
                Intent myIntent = new Intent(LoginActivity.this, MainActivity.class);  //Replace MainActivity.class wi
                LoginActivity.this.startActivity(myIntent);
            }else{

            }

    };

    userNameET = (EditText)findViewById(R.id.edit_text_email);
    passwordET = (EditText)findViewById(R.id.edit_text_password);
    Button login = (Button) findViewById(R.id.login);
    login.setOnClickListener((v) → {
            startActivityForResult(
                    AuthUI.getInstance()
                            .createSignInIntentBuilder()
                            .setIsSmartLockEnabled(false)
                            .setProviders(Arrays.asList(new AuthUI.IdpConfig.Builder(AuthUI.EMAIL_PROVIDER).build(),
                                    new AuthUI.IdpConfig.Builder(AuthUI.GOOGLE_PROVIDER).build()
                            ))
                            .build(),
                    RC_SIGN_IN);
    });
```

TRENDING RECYCLER VIEW




```
@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
                         Bundle savedInstanceState) {
    // Inflate the layout for this fragment
    View v = inflater.inflate(R.layout.fragment_main_page,container,false);
    Query query = childRef.orderByChild("view_count");
    myFirebaseRecyclerAdapter = new MyFirebaseRecylerAdapter(Recipe.class,
            R.layout.cardviewlayout,MyFirebaseRecylerAdapter.MovieViewHolder.class,query,getApplicationContext()
    mRecyclerView = (RecyclerView) v.findViewById(R.id.recipecards);
    mLinearLayoutManager = new LinearLayoutManager(getActivity());
    mRecyclerView.setLayoutManager(mLinearLayoutManager);
    mRecyclerView.setAdapter(myFirebaseRecyclerAdapter);
    mRecyclerView.setHasFixedSize(true);
    categories = (Button) v.findViewById(R.id.ButtonCatg);
    cuisine = (Button) v.findViewById(R.id.ButtonCuisine);
    createRecipe = (Button) v.findViewById(R.id.ButtonCreate);
    viewAll = (Button) v.findViewById(R.id.ButtonViewAll);
    final HandleClickListerner handleClickListerner;
    handleClickListerner = (HandleClickListerner)v.getContext();
    myFirebaseRecyclerAdapter.SetOnItemClick(new MyFirebaseRecylerAdapter.ClickItem(){
```

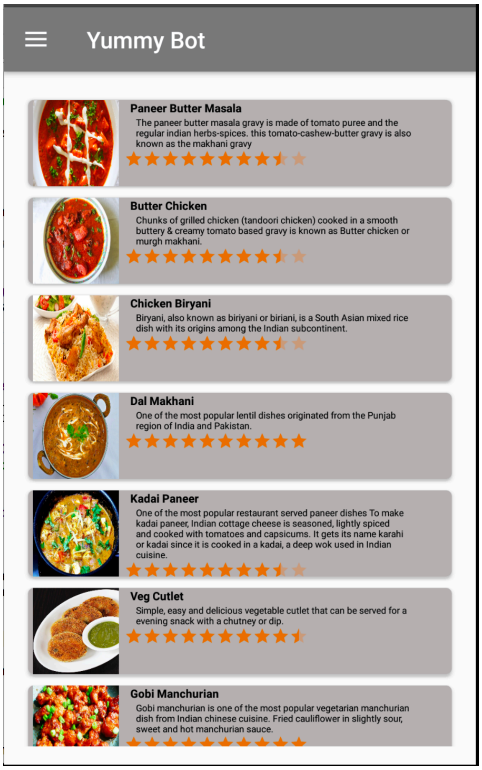
NAVIGATION DRAWER




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

    if (id == R.id.homePage) {
        Intent myIntent = new Intent(getApplicationContext(),MainActivity.class);
        startActivity(myIntent);
        // Handle the camera action


    } else if (id == R.id.nav_manage) {
        FirebaseAuth auth = FirebaseAuth.getInstance();
        auth.signOut();
        Intent intent = new Intent(getApplicationContext(),LoginActivity.class);
        startActivity(intent);

    } else if (id == R.id.nav_share) {
        query = childRef.orderByChild("user").equalTo(name);
        mContent = CardViewFragment.newInstance(query);
        getSupportFragmentManager().beginTransaction()
                .replace(R.id.mainContainer,mContent)
                .addToBackStack("store")
                .commit();

    } else if (id == R.id.nav_about) {

    }

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

RECYCLER VIEW

Connects to Firebase Database where FireBaseRecycler Adapter is used.



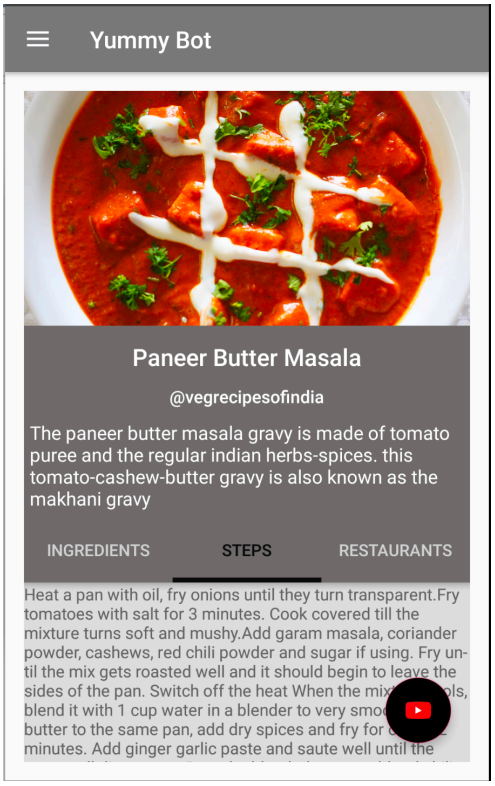

```
@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
                         Bundle savedInstanceState) {
    View v = inflater.inflate(R.layout.fragment_card_view, container, false);
    //Query query = childRef.orderByChild("view_count");
    myFirebaseRecyclerAdapter = new MyFirebaseRecylerAdapter(Recipe.class, R.layout.cardviewlayout,
            MyFirebaseRecylerAdapter.MovieViewHolder.class, query, getApplicationContext());
    mRecyclerView = (RecyclerView) v.findViewById(R.id.recipecards);
    mLinearLayoutManager = new LinearLayoutManager(getActivity());
    mRecyclerView.setLayoutManager(mLinearLayoutManager);
    mRecyclerView.setAdapter(myFirebaseRecyclerAdapter);
    final HandleClickListerner handleClickListerner;
    handleClickListerner = (HandleClickListerner) v.getContext();
    myFirebaseRecyclerAdapter.SetOnItemClick(new MyFirebaseRecylerAdapter.ClickItem() {

        @Override
        public void LongClick(View v, int position) {

        }

        @Override
        public void ShortClick(View v, int position) {
            final DatabaseReference reference = FirebaseDatabase.getInstance().getReference().child("recipes").getRef();
            reference.child(myFirebaseRecyclerAdapter.getRef(position).getKey()).
                    addValueEventListener(new com.google.firebase.database.ValueEventListener() {

                @Override
                public void onDataChange(DataSnapshot dataSnapshot) {
                    HashMap<String, ?> item = (HashMap<String, String>) dataSnapshot.getValue();
                    handleClickListerner.refresh(DetailViewFragment.newInstance(item));
                    Log.d("", "");
                }

                @Override
                public void onCancelled(DatabaseError databaseError) {

                }
            });
```

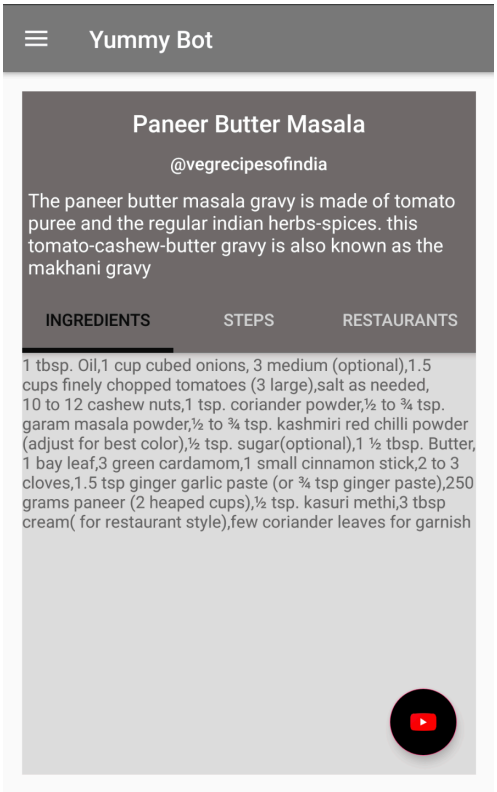
CO-ORDINATOR LAYOUT

Contains a collapsible image, appbar layout and view pager.

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <android.support.design.widget.AppBarLayout
        android:id="@+id/materialup.appbar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="#6e6d6d"
        android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar">
        <android.support.design.widget.CollapsingToolbarLayout
            android:id="@+id/main.collapsing"
            android:layout_width="match_parent"
            android:layout_height="200dp"
            app:expandedTitleMarginStart="48dp"
            app:expandedTitleMarginEnd="64dp"
            app:layout_scrollFlags="scroll|snap">
            <ImageView
                android:id="@+id/RecipeImage"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:scaleType="centerCrop"
                android:src="@drawable/test_image"
                app:layout_collapseMode="parallax"/>
        </android.support.design.widget.CollapsingToolbarLayout>
        <LinearLayout
            android:id="@+id/materialup.title_container"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            android:paddingTop="8dp"
            android:gravity="center"
            app:layout_scrollFlags="scroll|enterAlways|snap">
            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:id="@+id/RecipeName"
                android:layout_margin="5dp"
                android:textAppearance="@style/TextAppearance.AppCompat.Widget.ActionBar.Title"
```

```
            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_margin="5dp"
                android:textAppearance="@style/TextAppearance.AppCompat.Widget.ActionBar.Title"
                android:text="user"
                android:textSize="15dp"
                android:id="@+id/user"/>
            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginTop="4dp"
                android:layout_marginBottom="4dp"
                android:id="@+id/RecipeDescription"
                android:layout_margin="5dp"
                android:textAppearance="@style/TextAppearance.AppCompat.Widget.ActionBar.Subtitle"
                android:text="description"
                android:textColor="@android:color/white"/>
        </LinearLayout>
    <android.support.design.widget.TabLayout
        android:id="@+id/materialup.tabs"
        android:layout_width="fill_parent"
        android:layout_height="?attr/actionBarSize"
        app:tabSelectedTextColor="?android:attr/textColorPrimaryInverse"
        app:tabIndicatorColor="?android:attr/textColorPrimaryInverse"
        app:tabIndicatorHeight="4dp"/>
</android.support.design.widget.AppBarLayout>
<android.support.design.widget.FloatingActionButton
    android:id="@+id/fab"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="bottom|end"
    android:layout_margin="16dp"
    android:src="@drawable/utube"
    android:backgroundTint="#000000"/>
<android.support.v4.view.ViewPager
    android:id="@+id/materialup.viewpager"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layout_behavior="android.support.design.widget.AppBarLayout$ScrollingVie..."/>
</android.support.design.widget.CoordinatorLayout>
```

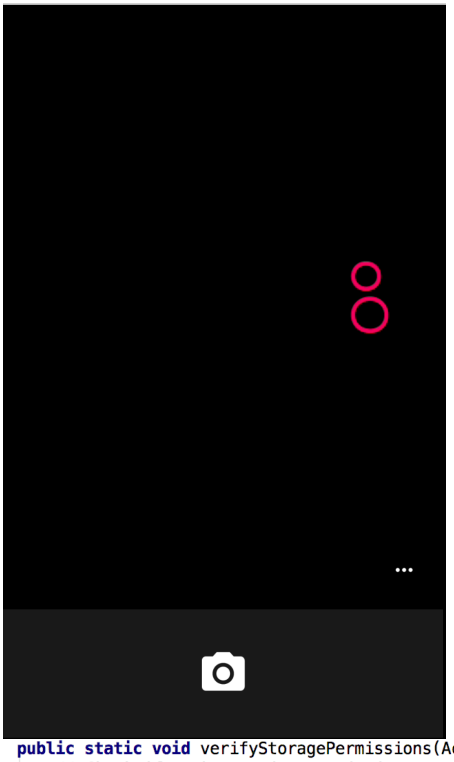

Yummy Bot
Paneer Butter Masala
@vegrecipesofindia
The paneer butter masala gravy is made of tomato puree and the regular indian herbs-spices. this tomato-cashew-butter gravy is also known as the makhani gravy
INGREDIENTS
STEPS
RESTAURANTS
Heat a pan with oil, fry onions until they turn transparent.Fry tomatoes with salt for 3 minutes. Cook covered till the mixture turns soft and mushy.Add garam masala, coriander powder, cashews, red chili powder and sugar if using. Fry until the mix gets roasted well and it should begin to leave the sides of the pan. Switch off the heat When the mix
ols,
blend it with 1 cup water in a blender to very smo
butter to the same pan, add dry spices and fry for
minutes. Add ginger garlic paste and saute well until the


Yummy Bot
Paneer Butter Masala
@vegrecipesofindia
The paneer butter masala gravy is made of tomato puree and the regular indian herbs-spices. this tomato-cashew-butter gravy is also known as the makhani gravy
INGREDIENTS
STEPS
RESTAURANTS
1 tbsp. Oil,1 cup cubed onions, 3 medium (optional),1.5 cups finely chopped tomatoes (3 large),salt as needed, 10 to 12 cashew nuts,1 tsp. coriander powder,½ to ¾ tsp. garam masala powder,½ to ¾ tsp. kashmiri red chilli powder (adjust for best color),½ tsp. sugar(optional),1 ½ tbsp. Butter, 1 bay leaf,3 green cardamom,1 small cinnamon stick,2 to 3 cloves,1.5 tsp ginger garlic paste (or ¾ tsp ginger paste),250 grams paneer (2 heaped cups),½ tsp. kasuri methi,3 tbsp cream(for restaurant style),few coriander leaves for garnish

View Pager

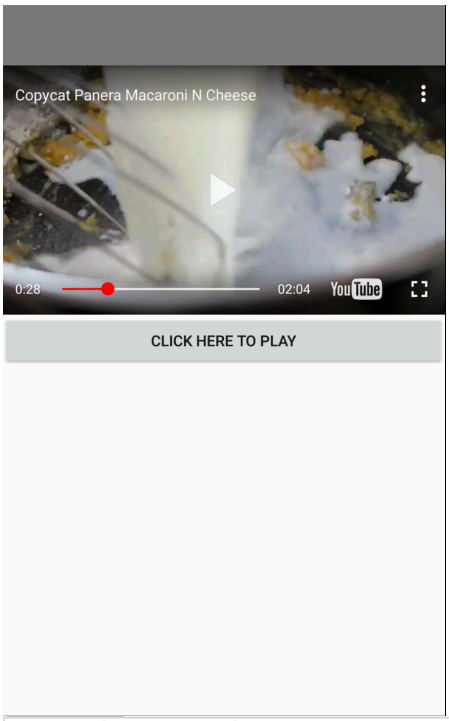
```
        appbarLayout.addOnOffsetChangedListener(this);
        mMaxScrollSize = appbarLayout.getTotalScrollRange();

        myPagerAdapter = new TabsAdapter(getActivity().getSupportFragmentManager(), 3);

        viewPager.setAdapter(myPagerAdapter);
        viewPager.setPageTransformer(true,new RotateDownTransformer());
        tabLayout.setupWithViewPager(viewPager);

        final TextView recipeName = (TextView) v.findViewById(R.id.RecipeName);
        final ImageView recipeImage = (ImageView) v.findViewById(R.id.RecipeImage);
        final TextView recipeUser = (TextView) v.findViewById(R.id.user);
        final TextView recipeDescription = (TextView)v.findViewById(R.id.RecipeDescription);

        recipeName.setText((String)item.get("recipe_name"));
        Picasso.with(getContext()).load((String)item.get("image_URL")).into(recipeImage);
        recipeUser.setText("@"+(String)item.get("user"));
        recipeDescription.setText((String)item.get("description"));

        return v;
    }

    private  class TabsAdapter extends FragmentStatePagerAdapter {
        private int TAB_COUNT;

        TabsAdapter(FragmentManager fm,int size) {
            super(fm);
            TAB_COUNT = size;
        }

        @Override
        public int getCount() { return TAB_COUNT; }

        @Override
        public Fragment getItem(int i)
        {
            Fragment frag = null;
            if(i==0)
            {
                 frag = IngredientsFragment.newInstance(item);
            }
            else if(i==1)
            {
                frag = StepsFragment.newInstance(item);
            }

            else{
                frag = RestaurantFragment.newInstance(item);
            }
            return frag;
        }

        @Override
        public CharSequence getPageTitle(int position)
        {
            if(position==0)
            {
                return "Ingredients";
            }
            if(position==1)
            {
                return "Steps";
            }
```

CAMERA ACCESS AND FIREBASE STORAGE



```
public static void verifyStoragePermissions(Activity activity) {
    // Check if we have write permission
    int permission = ActivityCompat.checkSelfPermission(activity, android.Manifest.permission.WRITE_EXTERNAL_STORAGE);

    if (permission != PackageManager.PERMISSION_GRANTED) {
        // We don't have permission so prompt the user
        ActivityCompat.requestPermissions(
                activity,
                PERMISSIONS_STORAGE,
                REQUEST_EXTERNAL_STORAGE
        );
    }
}
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    Context context = this;

    if (requestCode == 20 && resultCode == Activity.RESULT_OK) {
        ImageView imageview = (ImageView) findViewById(R.id.imageView);
        Bitmap bitmap;
        try {
            //bitmap = MediaStore.Images.Media.getBitmap(getContentResolver(), photoURI);
            bitmap = BitmapFactory.decodeFile(Userphoto.getAbsolutePath());
            imageview.setImageBitmap(bitmap);

            String imageURL = UUID.randomUUID().toString() + ".jpg";
            InputStream stream = new FileInputStream(Userphoto);
            FirebaseStorage storage = FirebaseStorage.getInstance();
            StorageReference storageRef = storage.getReferenceFromUrl("gs://yummy-bot-7b256.appspot.com/images");
            StorageReference storageReference = storageRef.child("images/" + imageURL );
           // StorageReference storageReference = FirebaseStorage.getInstance("images").getReference();
            UploadTask uploadTask = storageReference.putStream(stream);



            uploadTask.addOnFailureListener((exception) → {
                    // Handle unsuccessful uploads
            }).addOnSuccessListener((OnSuccessListener) (taskSnapshot) → {
                    // taskSnapshot.getMetadata() contains file metadata such as size, content-type, and download URL.
                    /VisibleForTests/ Uri downloadUrl = taskSnapshot.getDownloadUrl();
                    uploadedImageUrl = downloadUrl.toString();
            });
```

YOUTUBE




```
<?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"
    app:layout_behavior="android.support.design.widget.AppBarLayout$ScrollingVie..."
    tools:context="com.example.abhiram.yummybot.YouTubeActivity"
    tools:showIn="@layout/activity_you_tube">
    <com.google.android.youtube.player.YouTubePlayerView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/YoutubePlayer"/>
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Click here to Play"
        android:id="@+id/playButton"/>
</LinearLayout>
```

```
public class YouTubeActivity extends YouTubeBaseActivity {
    YouTubePlayerView mYouTubePlayerView;
    Button mButton;
    YouTubePlayer.OnInitializedListener mOnInitializedListener;
    String newString;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_you_tube);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        mButton = (Button) findViewById(R.id.playButton);
        Bundle extras = this.getIntent().getExtras();
        if(extras==null){
            newString =null;
        }
        else{
            newString = extras.getString("videoURL");
        }
        mButton.setOnClickListener((v) -> {
                mYouTubePlayerView.initialize(PlayerConfig.API_KEY,mOnInitializedListener);
        });
        mYouTubePlayerView = (YouTubePlayerView) findViewById(R.id.YoutubePlayer);
        mOnInitializedListener = new YouTubePlayer.OnInitializedListener() {
            @Override
            public void onInitializationSuccess(YouTubePlayer.Provider provider, YouTubePlayer youTubePlayer, boolean b) {
                youTubePlayer.loadVideo(newString);
            }
            @Override
            public void onInitializationFailure(YouTubePlayer.Provider provider,
                                                YouTubeInitializationResult youTubeInitializationResult) {

            }
        };
```

REAL API DATA

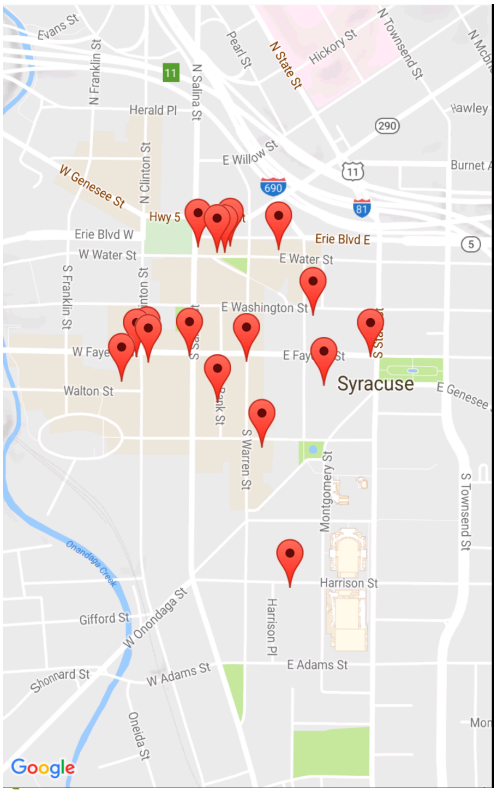
Here we use Zomato API to fetch nearby restaurants by giving current location and cuisine as input.

```
    @Override
    public void onMapReady(GoogleMap googleMap)
    {
        mgoogleMap=googleMap;
        String apiurl="https://developers.zomato.com/api/v2.1/search?entity_type=city&cuisines=";
        new RestaurantDownloadJsonAsyncTask(this).execute( new String[]{apiurl+newString});
    }
    @Override
    public void onFailure() {
    }
    @Override
    public void onStartlisten() {
    }
    @Override
    public void onSuccess(Object obj)
    {
        List<Marker> markers = new ArrayList<>();
        LatLng newloc=null;
        RestaurantDataJson threadMovieData = (RestaurantDataJson) obj;
        mgoogleMap.setMapType(GoogleMap.MAP_TYPE_TERRAIN);
        for (int i = 0; i < threadMovieData.getSize(); i++)
        {
            newloc = new LatLng(Double.parseDouble(threadMovieData.getItem(i).
                    get("latitude").toString()), Double.parseDouble(threadMovieData.getItem(i).get("longitude").toString()));

            Marker marker = mgoogleMap.addMarker(new MarkerOptions().position(newloc)
                    .title(threadMovieData.getItem(i).get("name").toString()).visible(true));
            markers.add(marker);
        }
        LatLngBounds.Builder builder = new LatLngBounds.Builder();
        for (Marker marker : markers)
        {
            builder.include(marker.getPosition());
        }
        LatLngBounds bounds = builder.build();
        int padding = 0; // offset from edges of the map in pixels
        CameraUpdate cu = CameraUpdateFactory.newLatLngBounds(bounds, padding);
        mgoogleMap.animateCamera(cu);
    }
}
```

```
public static String downloadJSONusingHTTPGetRequest(String urlString)
{
    String jsonString=null;

    try {
        URL url = new URL(urlString);
        HttpsURLConnection httpConnection = (HttpsURLConnection) url.openConnection();
        httpConnection.setRequestProperty("Accept", " application/json");
        httpConnection.setRequestProperty("user-key", " "+"0faa556e15490bddd4ac5fb5b97cb555");
        if (httpConnection.getResponseCode() == HttpsURLConnection.HTTP_OK)
        {
            InputStream stream = httpConnection.getInputStream();
            jsonString = getStringfromStream(stream);
        }
        httpConnection.disconnect();
    }  catch (UnknownHostException e1)
    {
        Log.d("MyDebugMsg", "UnknownHostexception in downloadJSONusingHTTPGetRequest");
        e1.printStackTrace();
    } catch (Exception ex) {
        Log.d("MyDebugMsg", "Exception in downloadJSONusingHTTPGetRequest");
        ex.printStackTrace();
    }
     return jsonString;
}
private static String getStringfromStream(InputStream stream){
    String line, jsonString=null;
    if (stream != null) {
        BufferedReader reader = new BufferedReader(new InputStreamReader(stream));
        StringBuilder out = new StringBuilder();
        try {
            while ((line = reader.readLine()) != null) {
                out.append(line);
            }
            reader.close();
            jsonString = out.toString();
        } catch (IOException ex) {
            Log.d("MyDebugMsg", "IOException in downloadJSON()");
            ex.printStackTrace();
        }
    }
    return jsonString;
```

MAPS

Latitude and longitude of restaurants from Zomato API are marked on the map using marker.




```
public void onSuccess(Object obj)
{
    List<Marker> markers = new ArrayList<~>();
    LatLng newloc=null;
    RestaurantDataJson RecipeData = (RestaurantDataJson) obj;
    mgoogleMap.setMapType(GoogleMap.MAP_TYPE_TERRAIN);
    for (int i = 0; i < RecipeData.getSize(); i++)
    {
        newloc = new LatLng(Double.parseDouble(RecipeData.getItem(i).
                get("latitude").toString()), Double.parseDouble(RecipeData.getItem(i).get("longitude").toString()));

        Marker marker = mgoogleMap.addMarker(new MarkerOptions().position(newloc)
                .title(RecipeData.getItem(i).get("name").toString()).visible(true));
        markers.add(marker);
    }
    LatLngBounds.Builder builder = new LatLngBounds.Builder();
    for (Marker marker : markers)
    {
        builder.include(marker.getPosition());
    }
    LatLngBounds bounds = builder.build();
    int padding = 0; // offset from edges of the map in pixels
    CameraUpdate cu = CameraUpdateFactory.newLatLngBounds(bounds, padding);
    mgoogleMap.animateCamera(cu);
}
```

ANIMATION

```xml
<?xml version="1.0" encoding="utf-8"?>
<set>
    <translate xmlns:android="http://schemas.android.com/apk/res/android"
               android:fromXDelta="0"
               android:toXDelta="-100%"
               android:interpolator="@android:anim/decelerate_interpolator"
               android:duration="300"/>
</set>
```

```xml
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
     android:fillAfter="true" >

    <alpha
        android:duration="1000"
        android:fromAlpha="1.0"
        android:interpolator="@android:anim/accelerate_interpolator"
        android:toAlpha="0.0" />

</set>
```

```xml
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
     android:fillAfter="true" >

    <alpha
        android:duration="1000"
        android:fromAlpha="0.0"
        android:interpolator="@android:anim/accelerate_interpolator"
        android:toAlpha="1.0" />

</set>
```

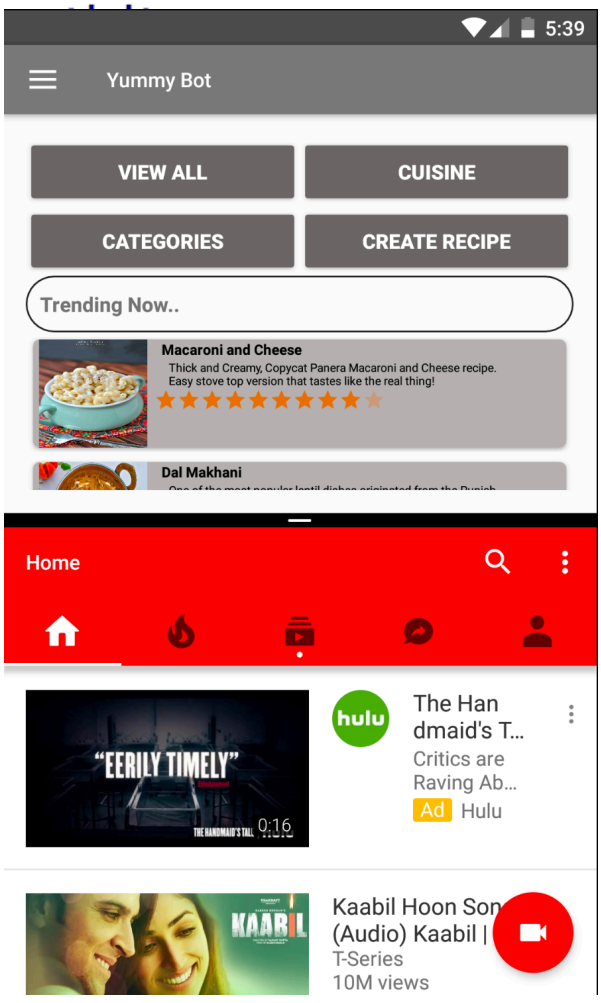
```xml
<?xml version="1.0" encoding="utf-8"?>
<set>
    <translate xmlns:android="http://schemas.android.com/apk/res/android"
               android:fromXDelta="-100%"
               android:toXDelta="0"
               android:interpolator="@android:anim/decelerate_interpolator"
               android:duration="500"/>
</set>
```

SHORTCUTS

```xml
<?xml version="1.0" encoding="utf-8"?>
<shortcuts xmlns:android="http://schemas.android.com/apk/res/android">
    <shortcut
        android:shortcutId="Trending Now"
        android:enabled="true"
        android:icon="@mipmap/ic_launcher"
        android:shortcutShortLabel="Trending Now"
        android:shortcutLongLabel="Trending Now"
        >
        <intent
            android:action="android.intent.action.VIEW"
            android:targetPackage="com.example.abhiram.yummybot"
            android:targetClass="com.example.abhiram.yummybot.MainActivity" />

    </shortcut>

    <shortcut
        android:shortcutId="Create Recipe"
        android:enabled="true"
        android:icon="@mipmap/ic_launcher"
        android:shortcutShortLabel="Create Recipe"
        android:shortcutLongLabel="@string/label_long_create"
        >
        <intent
            android:action="android.intent.action.VIEW"
            android:targetPackage="com.example.abhiram.yummybot"
            android:targetClass="com.example.abhiram.yummybot.CreateRecipe" />

    </shortcut>
    <!-- Specify more shortcuts here. -->
</shortcuts>
```

MULTI WINDOW

```
<activity
    android:name=".MainActivity"
    android:label="Yummy Bot"
    android:resizeableActivity="true"
    android:supportsPictureInPicture="true"
android:theme="@style/AppTheme.NoActionBar">
    <layout android:defaultHeight="500dp"
            android:defaultWidth="600dp"
            android:gravity="top|end"
            android:minHeight="450dp"
            android:minWidth="300dp" />
</activity>
```



FIREBASE CLOUD STORAGE