Foodie
Final Project

Rahul Kadam and Vinutha Karanth

Table of Contents

Basic Features

Advanced Features

1. Splash Screen

SplashActivity.java

```
SplashActivity
1  package com.connect.foodies.foodies.home;
2
3  import ...
9
10
11 public class SplashActivity extends AppCompatActivity{
12
13     private final int SPLASH_DISPLAY_LENGTH = 3000;
14
15     public void onCreate(Bundle icicle) {
16         super.onCreate(icicle);
17         setContentView(R.layout.activity_splash_screen);
18
19         /* New Handler to start the Menu-Activity
20          * and close this Splash-Screen after some seconds.*/
21         new Handler().postDelayed(() -> {
24                 /* Create an Intent that will start the Menu-Activity. */
25                 Intent mainIntent = new Intent(SplashActivity.this,HomeActivity.class);
26                 SplashActivity.this.startActivity(mainIntent);
27                 SplashActivity.this.finish();
28         }, SPLASH_DISPLAY_LENGTH);
30     }
```

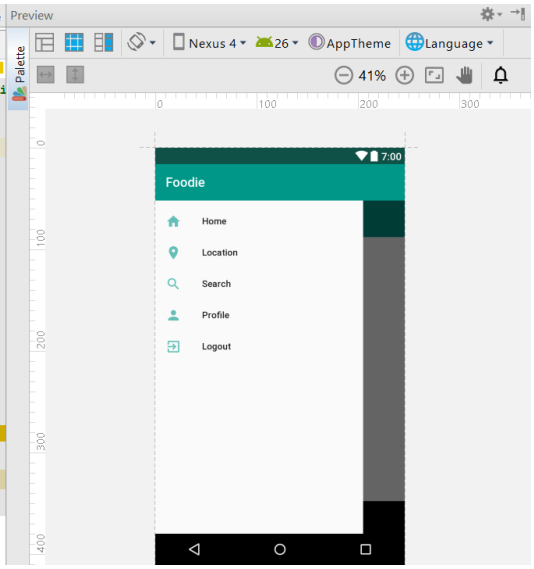
2. Navigation Drawer

HomeFragment.java | HomeActivity.java | activity_home.xml

android.support.v4.widget.DrawerLayout

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.v4.widget.DrawerLayout xmlns:android="http://schemas.androi
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/drawer_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true"
    tools:openDrawer="start">

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

    <android.support.design.widget.NavigationView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/nav_view"
        app:itemIconTint="@color/colorAccent"
        android:layout_gravity="start"
        app:menu="@menu/activity_home_drawer">>
    </android.support.design.widget.NavigationView>

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

Preview



HomeFragment.java | HomeActivity.java | activity_home.xml | MainfeedListAdapter.java | fragment_home.xml

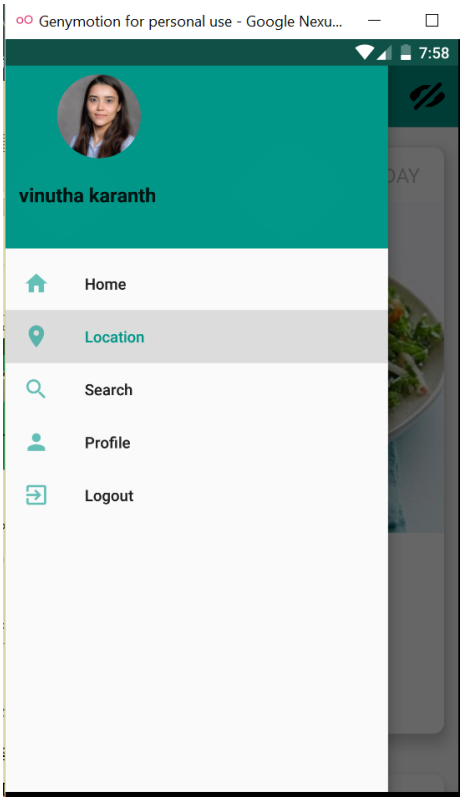
HomeActivity | onCreate()

```
        }
    });

    DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
    ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
            this, drawer, toolbar, "Open navigation drawer", "Close navigation drawer");
    drawer.setDrawerListener(toggle);
    toggle.syncState();
```

HomeActivity.java

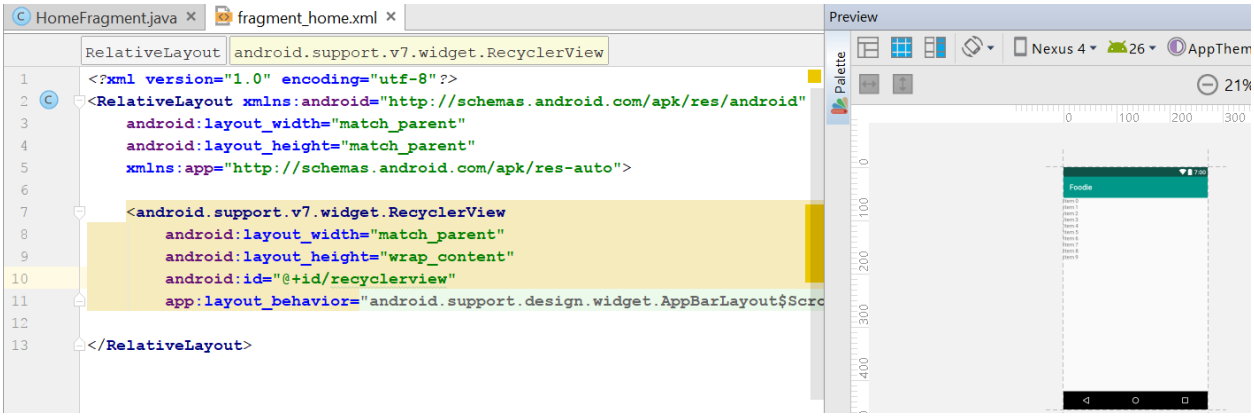
HomeActivity | onNavigationItemSelected()

```
public boolean onNavigationItemSelected(MenuItem item) {
    // Handle navigation view item clicks here.
    int id = item.getItemId();
    if (id == R.id.nav_home) {
        intent = new Intent(HomeActivity.this, HomeActivity.class);
        startActivity(intent);

    } else if (id == R.id.nav_location) {
        intent = new Intent(HomeActivity.this, MapsActivity.class);
        startActivity(intent);

    } else if (id == R.id.nav_search) {
        intent = new Intent(HomeActivity.this, SearchActivity.class);
        startActivity(intent);

    } else if (id == R.id.nav_profile) {
        intent = new Intent(HomeActivity.this, ProfileActivity.class);
        startActivity(intent);

    } else if (id == R.id.nav_logout) {
        Log.d(TAG, "onClick: attempting to sign out.");
        mAuth.signOut();
        finish();
    }
    DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
    drawer.closeDrawer(GravityCompat.START);
    return true;
}
```



3. Recycler View

Recycler View is used in Home Fragment, MainFeedListAdapter file.



HomeFragment.java × fragment_home.xml ×

HomeFragment onCreateView()

```
52      @Override
53      public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {
54          View view = inflater.inflate(R.layout.fragment_home, container, false);
55          mRecyclerView = (RecyclerView) view.findViewById(R.id.recyclerview);
56          mFollowing = new ArrayList<>();
57          mPhotos = new ArrayList<>();
```

HomeFragment.java × fragment_home.xml ×

HomeFragment displayPhotos()

```
141     private void displayPhotos(){
142         mPaginatedPhotos = new ArrayList<>();
143         if(mPhotos != null){
144             try{
145                 Collections.sort(mPhotos, (Comparator) (o1, o2) → {
148                         return o2.getDate_created().compareTo(o1.getDate_created());
149                 });
151                 int iterations = mPhotos.size();
152
153                 if(iterations > 10){
154                     iterations = 10;
155                 }
156                 mResults = 10;
157                 for(int i = 0; i < iterations; i++){
158                     mPaginatedPhotos.add(mPhotos.get(i));
159                 }
160                 mAdapter = new MainfeedListAdapter(getActivity(), mPaginatedPhotos);
161                 mRecyclerView.setAdapter(mAdapter);
162                 mRecyclerView.setNestedScrollingEnabled(false);
163                 mRecyclerView.setHasFixedSize(true);
164
165                 LinearLayoutManager mLayoutManager = new LinearLayoutManager(getContext());
166                 mRecyclerView.setLayoutManager(mLayoutManager);
167
168                 //Animations
169                 AlphaInAnimationAdapter alphaAdapter = new AlphaInAnimationAdapter(mAdapter);
170                 ScaleInAnimationAdapter scaleAdapter = new ScaleInAnimationAdapter(alphaAdapter);
171                 scaleAdapter.setDuration(1000);
172                 mRecyclerView.setAdapter(scaleAdapter);
```

MainFeedlistAdapter

HomeFragment.java × MainfeedListAdapter.java × fragment_home.xml ×

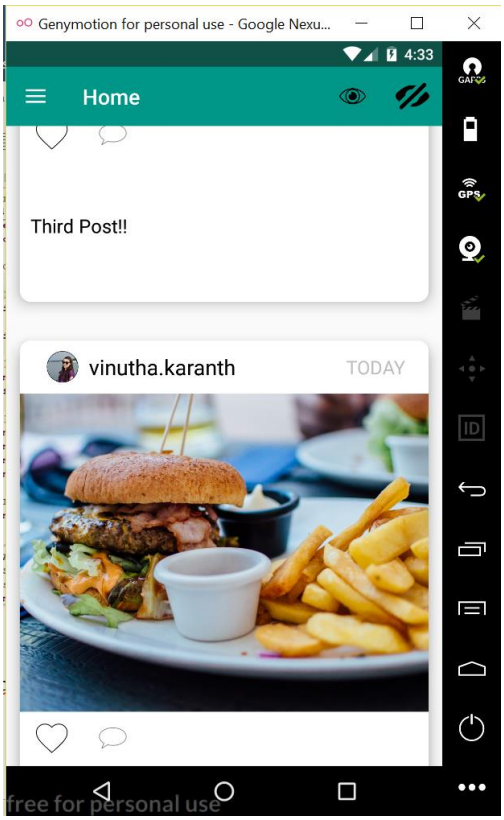
MainfeedListAdapter MyViewHolder

```
43
44  public class MainfeedListAdapter extends RecyclerView.Adapter<MainfeedListAdapter.MyViewHolder> {
45
46      private static final String TAG = "MainfeedListAdapter";
47      OnLoadMoreItemsListener mOnLoadMoreItemsListener;
48      private Context mContext;
49      private DatabaseReference mReference;
50      private String currentUsername = "";
51      private List<Photo> mListPhoto;
```

HomeFragment.java × MainfeedListAdapter.java × fragment_home.xml ×

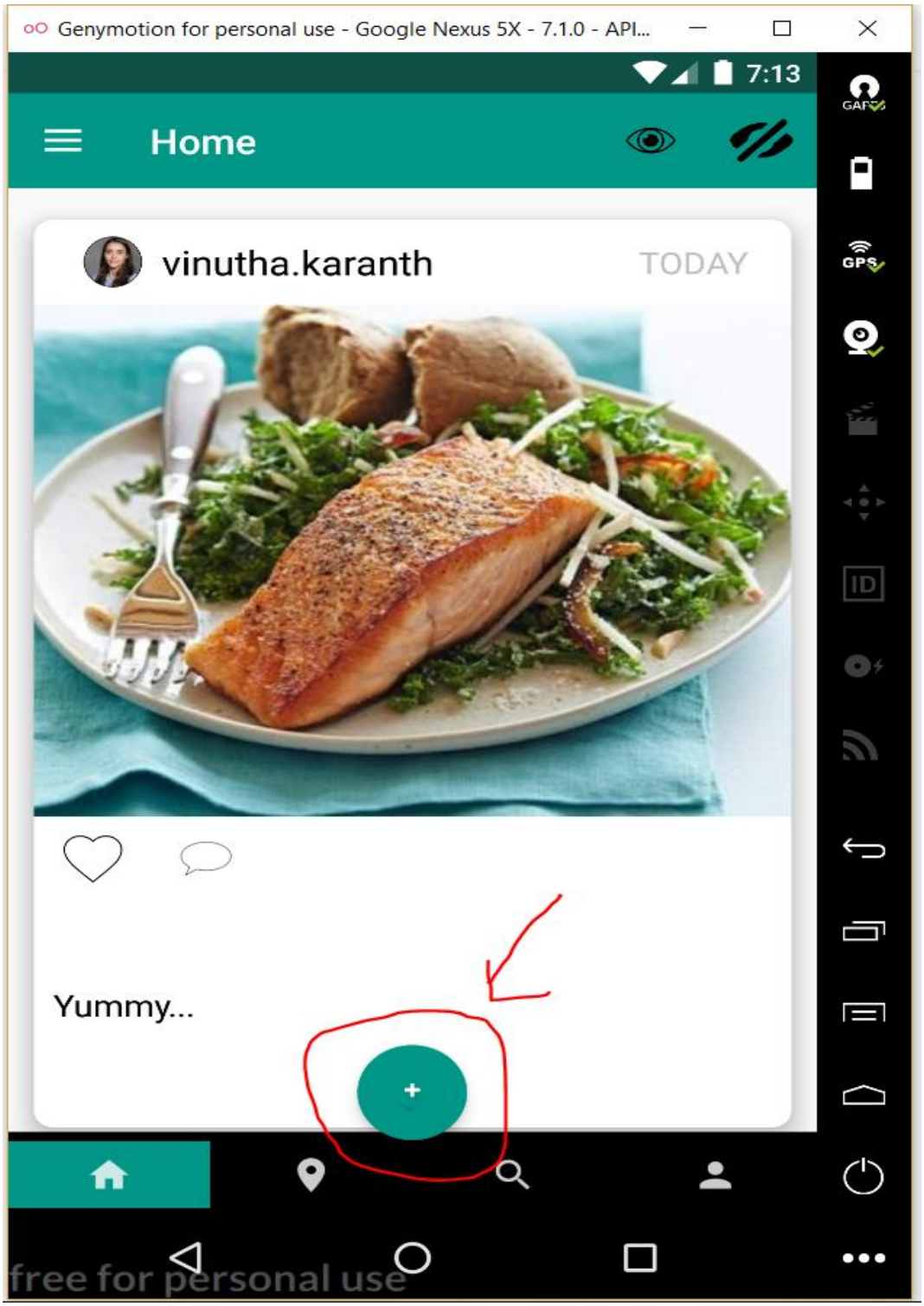
MainfeedListAdapter

```
438     static public class MyViewHolder extends RecyclerView.ViewHolder {
439         CircleImageView mprofileImage;
440         String likesString;
441         TextView username, timeDetla, caption, likes, comments;
442         ImageView image;
443         ImageView heartRed, heartWhite, comment;
444
445         UserAccountSettings settings = new UserAccountSettings();
446         User user = new User();
447         StringBuilder users;
448         boolean likeByCurrentUser;
449         Heart heart;
450         GestureDetector detector;
pter        Photo photo;
452
453         public MyViewHolder(View itemView) {
454             super(itemView);
455
456             username = (TextView) itemView.findViewById(R.id.username);
457             image = (ImageView) itemView.findViewById(R.id.post_image);
458             heartRed = (ImageView) itemView.findViewById(R.id.image_heart_red);
```

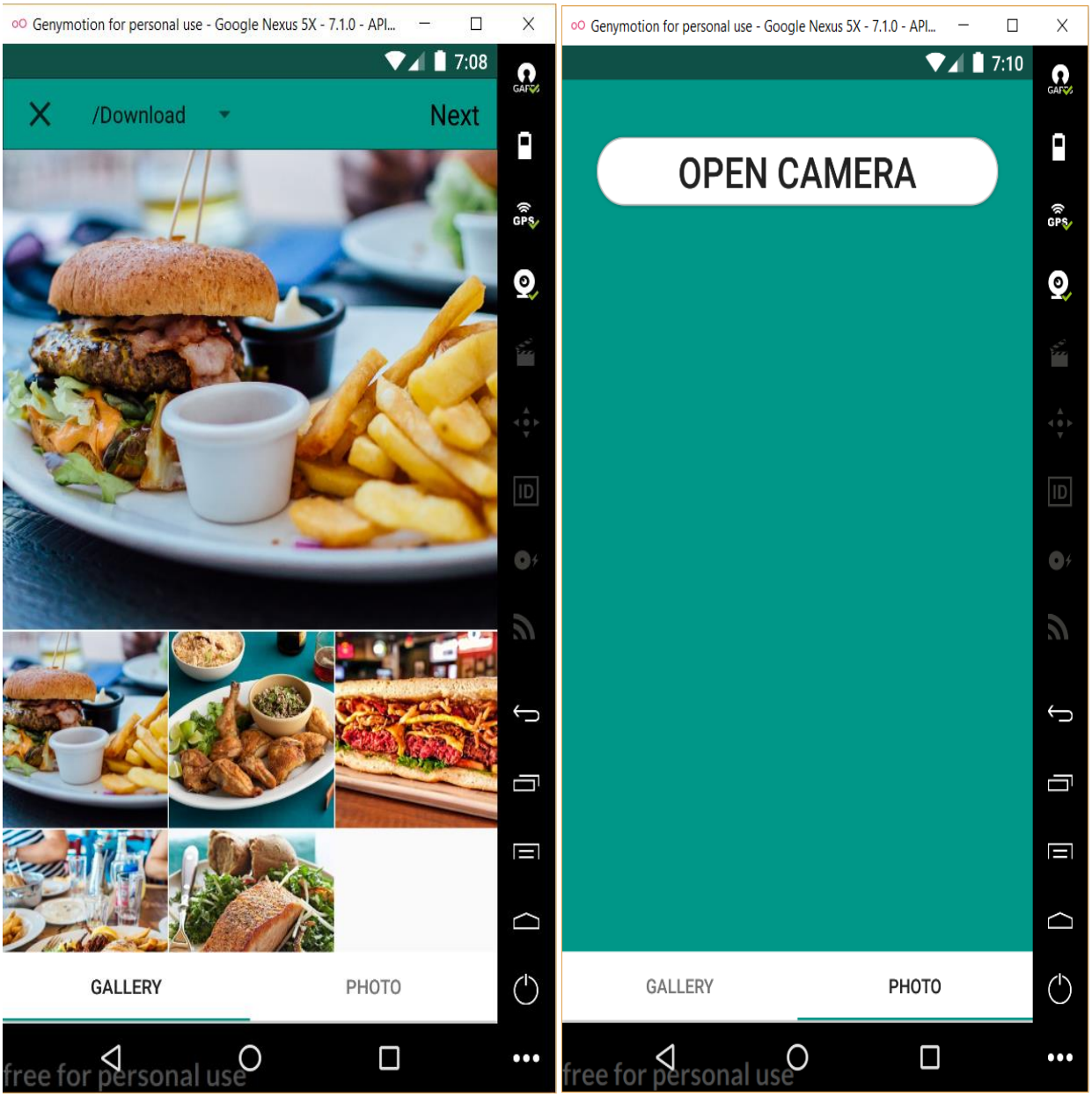


4. View Pager

View Pager is used we click green share floating button highlighted below:



This View Pager has two parts “Gallery” and “Photo” as show below:



Code is shown on next page

ShareActivity.java contains this View Pager:

ShareActivity.java ×

ShareActivity

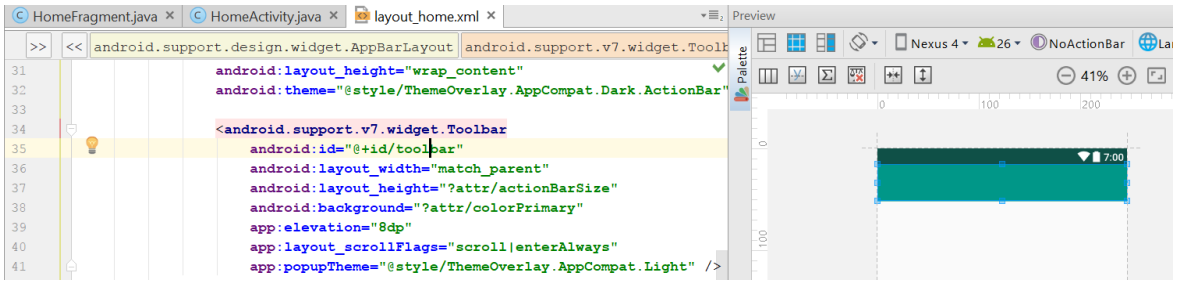
```
1  package com.connect.foodies.foodies.share;
2
3  import ...
28
29 public class ShareActivity extends AppCompatActivity {
30     private static final String TAG = "ShareActivity";
31
32     //constants
33     private static final int ACTIVITY_NUM = 2;
34     private static final int VERIFY_PERMISSIONS_REQUEST = 1;
35
36     private ViewPager mViewPager;
37     private TextView mTextView;
38     private TextView mTextView3;
39     private Context mContext = ShareActivity.this;
40
41     @Override
42     protected void onCreate(@Nullable Bundle savedInstanceState) {
43         super.onCreate(savedInstanceState);
44         setContentView(R.layout.activity_share);
45         Log.d(TAG, "onCreate: started.");
46
47         mTextView = (TextView)  findViewById(R.id.textView);
48         mTextView3 = (TextView) findViewById(R.id.textView3);
49         if (checkPermissionsArray(Permissions.PERMISSIONS)) {
50             setupViewPager();
51         } else {
52             verifyPermissions(Permissions.PERMISSIONS);
53             mTextView.setText("Checking For Permissions...");
54             mTextView3.setText("Once Permissions Granted. Please Go Back and then Share");
55         }
56
57     }
58
```

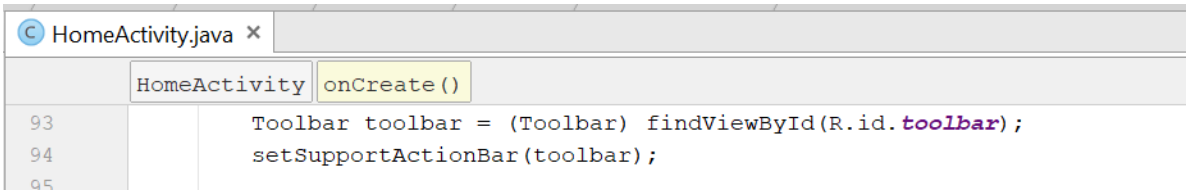
```
/**
 * setup viewpager for manager the tabs
 */
private void setupViewPager() {
    SectionsPagerAdapter adapter = new SectionsPagerAdapter(getSupportFragmentManager());
    adapter.addFragment(new GalleryFragment());
    adapter.addFragment(new PhotoFragment());

    mViewPager = (ViewPager) findViewById(R.id.viewpager_container);
    mViewPager.setAdapter(adapter);
    mViewPager.setPageTransformer(true, new RotateUpTransformer());

    TabLayout tabLayout = (TabLayout) findViewById(R.id.tabsBottom);
    tabLayout.setupWithViewPager(mViewPager);

    tabLayout.getTabAt(0).setText("GALLERY");
    tabLayout.getTabAt(1).setText("PHOTO");

}
```

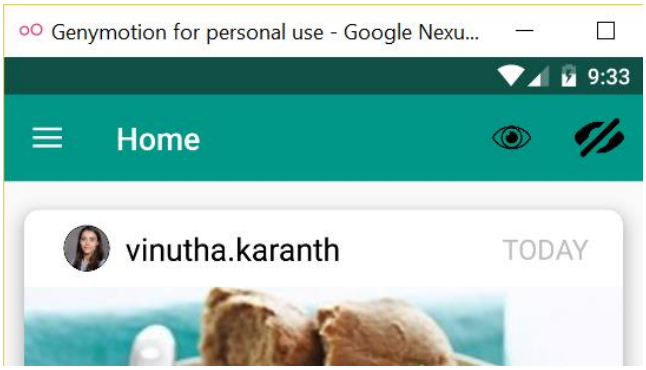
Gallery and Photo Fragments are explained in 9. Media Section.

5. Toolbar, Menus, Floating Action Button, and Coordinator Layout

A. Tools







B. Menus

HomeFragment.java | HomeActivity.java | home.xml | layout_home.xml | Preview

menu > item

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

    <item
        android:id="@+id/action_unhide"
        android:icon="@drawable/ic_unhide_toolbar"
        android:title="Hide"
        app:showAsAction="always" />
    <item
        android:id="@+id/action_hide"
        android:icon="@drawable/ic_hide_bottomtoolbar"
        android:title="Hide"
        app:showAsAction="always" />
</menu>
```

ment.java | HomeActivity.java | home.xml | layout_home.xml | MainfeedListAdapter.java

HomeActivity > onCreateOptionsMenu()

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.home, menu);
    return true;
}
```

ment.java | HomeActivity.java | home.xml | layout_home.xml | Ma

HomeActivity > onOptionsItemSelected()

```
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();
    //noinspection SimplifiableIfStatement
    if (id == R.id.action_hide) {
        bottomNavigationViewEx.setVisibility(View.GONE);
        fab.setVisibility(View.GONE);
        return true;
    }
    else if (id == R.id.action_unhide)
    {
        bottomNavigationViewEx.setVisibility(View.VISIBLE);
        fab.setVisibility(View.VISIBLE);
        return true;
    }
    return super.onOptionsItemSelected(item);
}
```

C. Floating action can be done in 3 activities – Home, profile and search

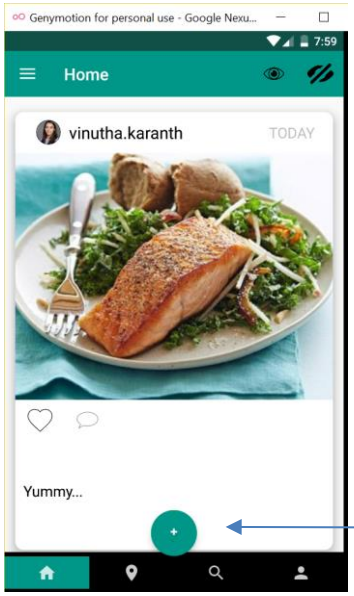
app_bar_home.xml

```
<< RelativeLayout | android.support.design.widget.FloatingActionButton

    <android.support.design.widget.FloatingActionButton
        android:id="@+id/fab"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:layout_gravity="bottom|center"
        android:layout_marginBottom="45dp"
        android:src="@drawable/floating"
        android:transitionName="testAnimation"/>
```

HomeActivity.java | SearchActivity.java | ProfileFragment.java

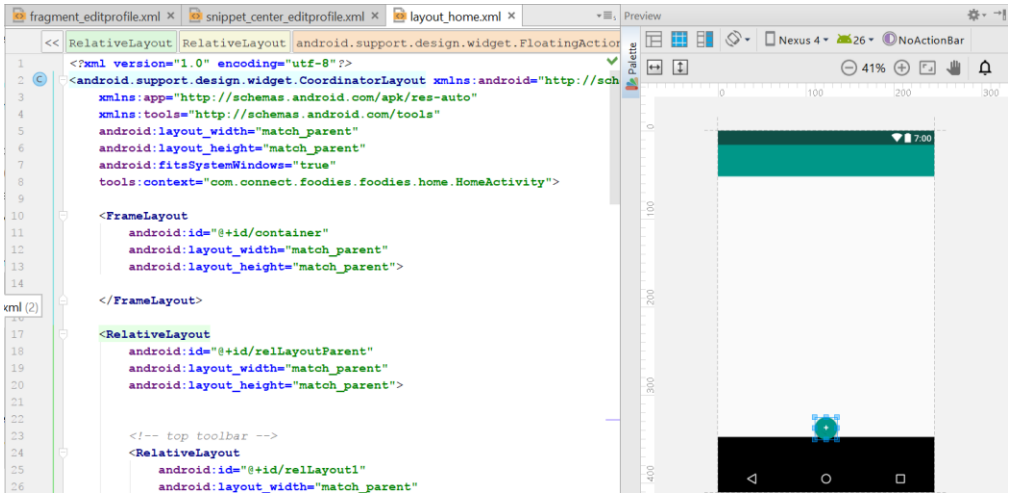
HomeActivity > onCreate()

```
this.fab = (FloatingActionButton) findViewById(R.id.fab);
fab.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(mContext, ShareActivity.class);
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
            ActivityOptionsCompat options = ActivityOptionsCompat.makeSceneTransitionAnimation(HomeActivity.this, view, "testAnimation");
            mContext.startActivity(intent, options.toBundle());
        } else {
            startActivity(intent);
        }
    }
});
```



Floating Action Button

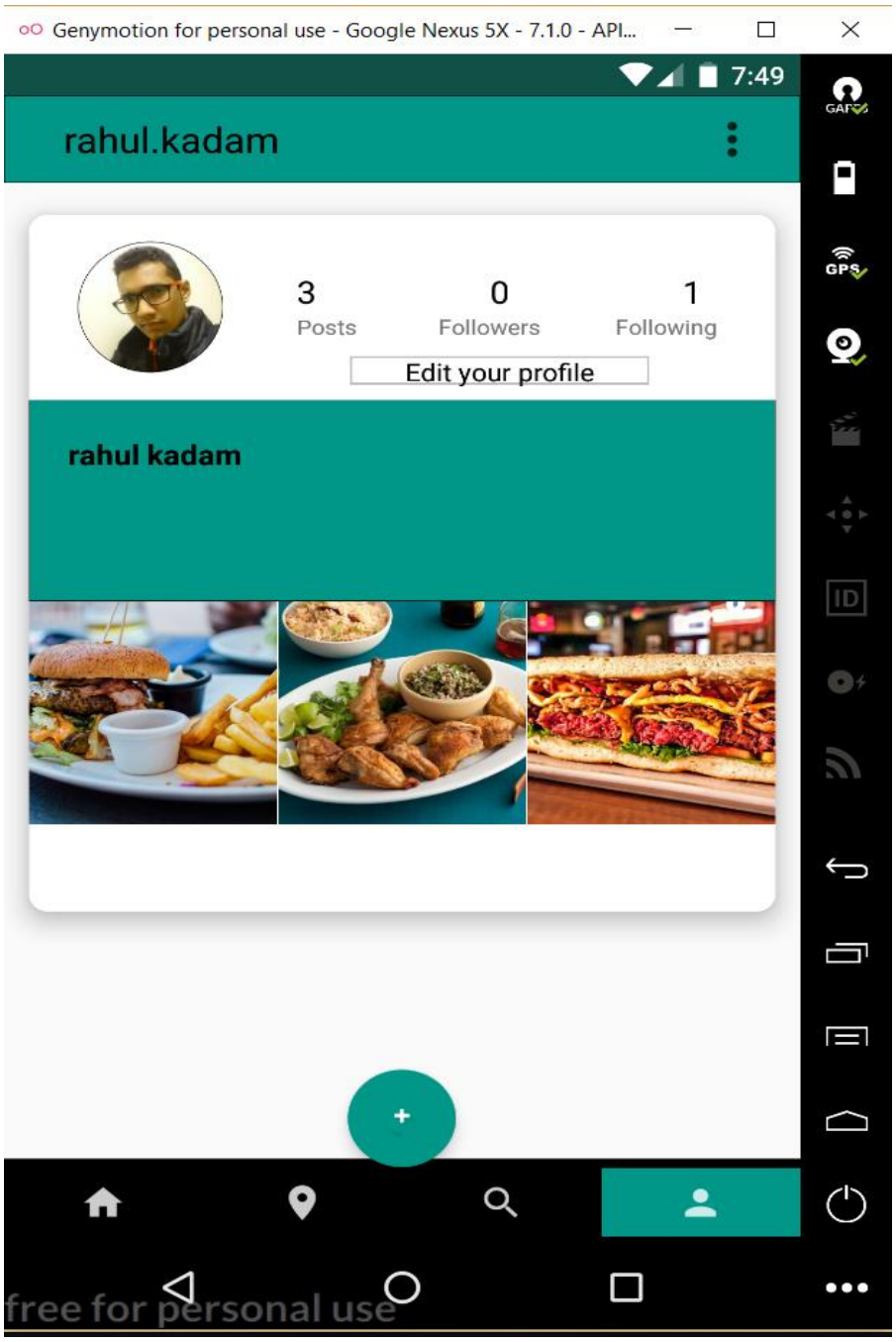
D. Coordinator layout is used in activity_accountsettings, activity_profile, activity_share, activity_bar_home, fragment_editprofile and layout_home. Floating action button and tool bar works perfectly without any disruption.



6. Multiple Fragments and Activities

One of way of using Fragments and Activities is for User Profile feature:

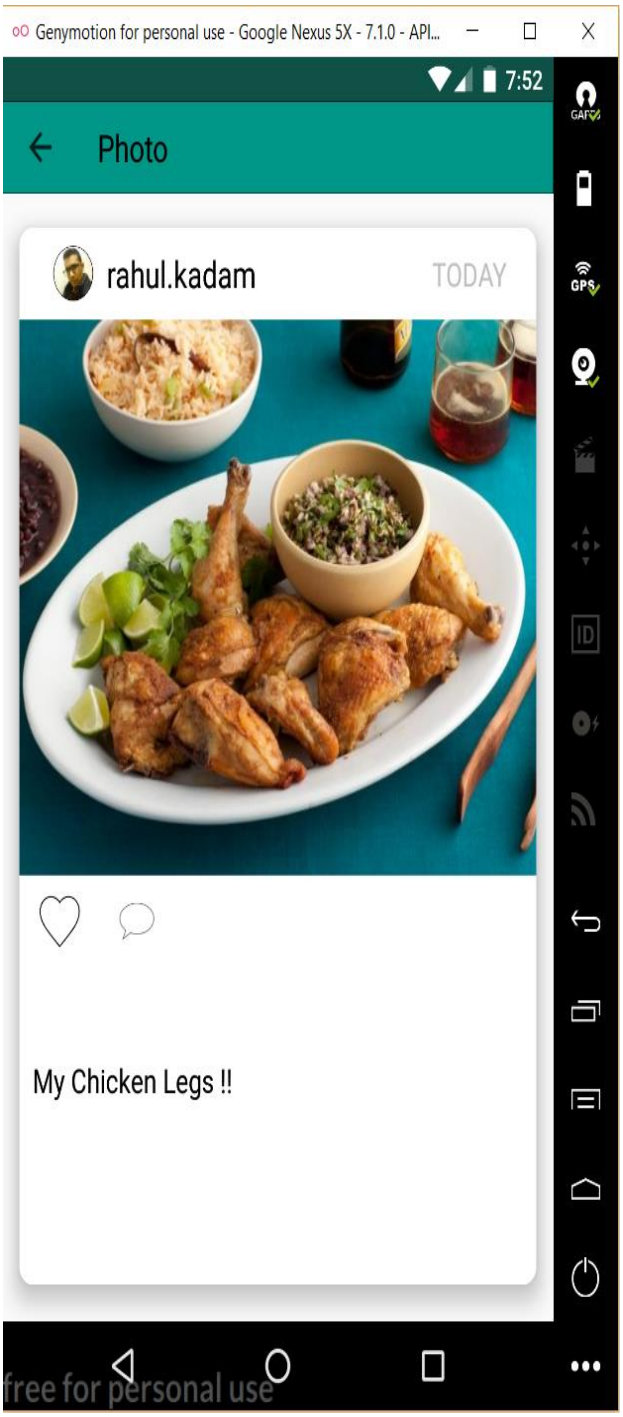
ProfileActivity.java first loads ProfileFragment.java



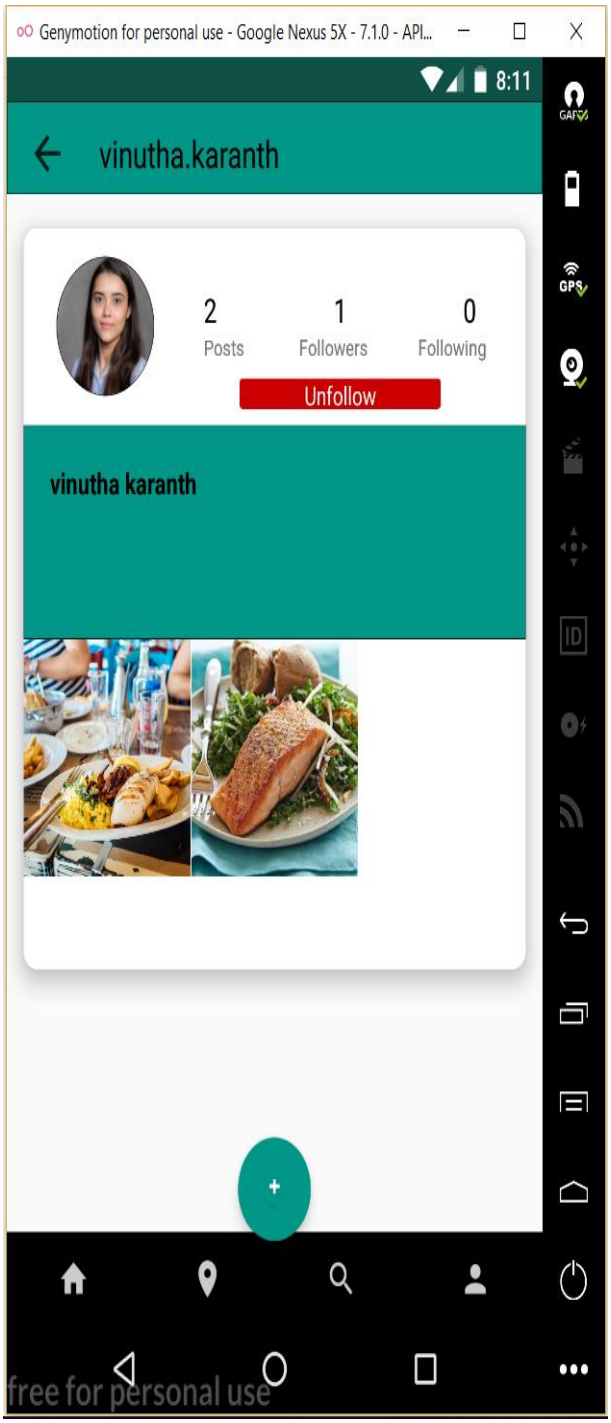
ProfileActivity.java acts as base activity and 3 fragments:

1. ProfileFragment.java (previous page)

2. ViewPostFragment.java



3.ViewProfileFragment.java



Code: ProfileActivity.java

ProfileActivity.java | ProfileFragment.java | ViewPostFragment.java | ViewProfileFragment.java

ProfileActivity

```
1  package com.connect.foodies.foodies.profile;
2
3  import ...
21
22 public class ProfileActivity extends AppCompatActivity implements
23         ProfileFragment.OnGridImageSelectedListener ,
24         ViewPostFragment.OnCommentThreadSelectedListener,
25         ViewProfileFragment.OnGridImageSelectedListener {
26
27     private static final String TAG = "ProfileActivity";
28     private static final int ACTIVITY_NUM = 3;
29     private static final int NUM_GRID_COLUMNS = 3;
30
31     private Context mContext = ProfileActivity.this;
32
```

Code: ProfileFragment.java

ProfileActivity.java | ProfileFragment.java | ViewPostFragment.java | ViewProfileFragment.java

ProfileFragment

```
1  package com.connect.foodies.foodies.profile;
2
3  import ...
54
55 public class ProfileFragment extends Fragment {
56
57     private static final String TAG = "ProfileFragment";
58
59     public interface OnGridImageSelectedListener {
60         void onGridImageSelected(Photo photo, int activityNumber);
61     }
62     OnGridImageSelectedListener mOnGridImageSelectedListener;
```

Code: ViewPostFragment.java

ProfileActivity.java | ProfileFragment.java | ViewPostFragment.java | ViewProfileFragment.java

ViewPostFragment

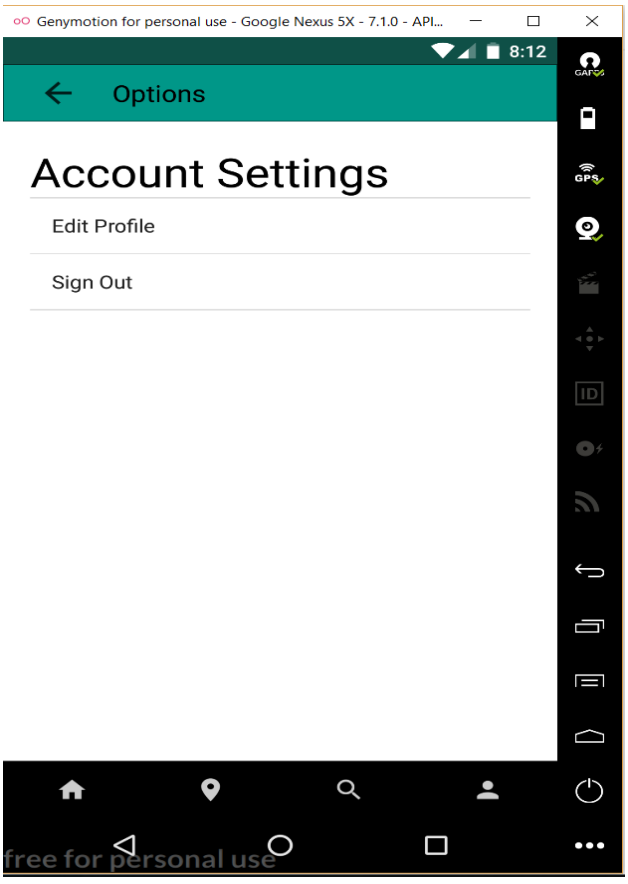
```
1  package com.connect.foodies.foodies.share;
2
3
4  import ...
52
53 public class ViewPostFragment extends Fragment {
54     private static final String TAG = "ViewPostFragment";
55
56
57     public interface OnCommentThreadSelectedListener{
58         void onCommentThreadSelectedListener(Photo photo);
59     }
60     OnCommentThreadSelectedListener mOnCommentThreadSelectedListener;
```

Code: ViewProfileFragment.java

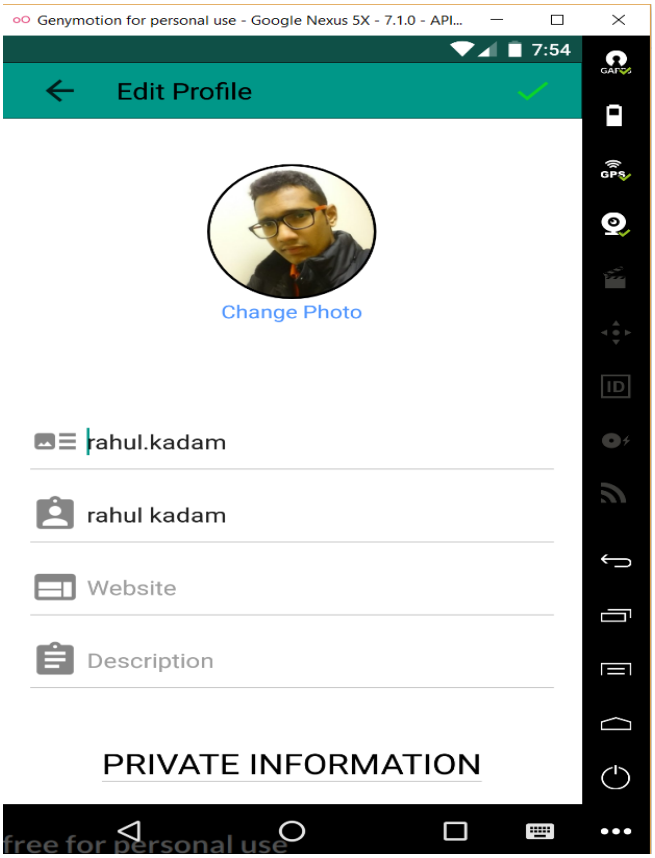
```
ProfileActivity.java × | ProfileFragment.java × | ViewPostFragment.java × | ViewProfileFragment.java ×
ViewProfileFragment
1   package com.connect.foodies.foodies.profile;
2
3   import ...
53
54
55  public class ViewProfileFragment extends Fragment {
56
57      private static final String TAG = "ProfileFragment";
58
59
60      public interface OnGridImageSelectedListener{
61          void onGridImageSelected(Photo photo, int activityNumber);
62      }
63      OnGridImageSelectedListener mOnGridImageSelectedListener;
```

AccountSettingsActivity.java acts as base activity for: EditProfileFragement.java and SignOutFragment.java

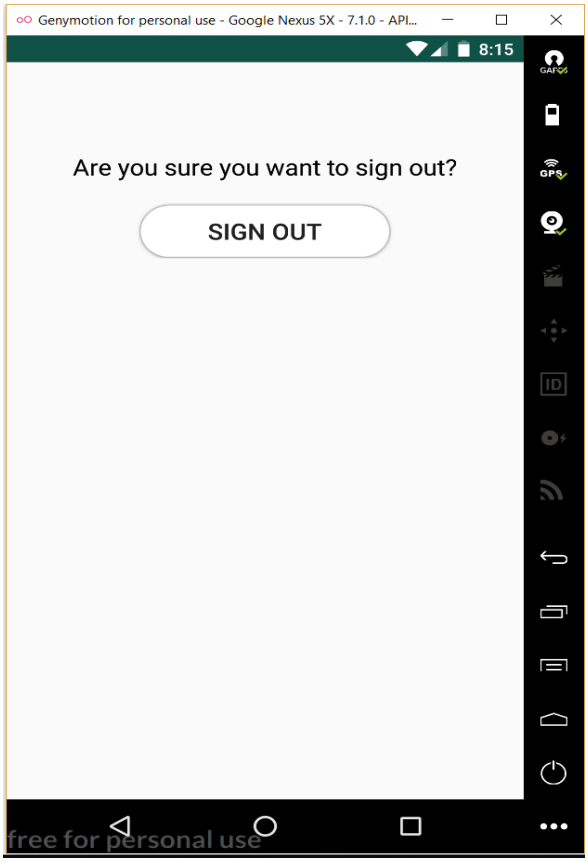
AccountSettingsActivity.java:



EditProfileFragment.java



SignOutFragment.java



Code: AccountSettingsActivity.java

AccountSettingsActivity.java | EditProfileFragment.java | SignOutFragment.java

AccountSettingsActivity › setupFragments()

```
1   package com.connect.foodies.foodies.profile;
2
3   import ...
27
28  public class AccountSettingsActivity extends AppCompatActivity {
29
30      private static final String TAG = "AccountSettingsActivity";
31      private static final int ACTIVITY_NUM = 3;
32
33      private Context mContext;
34      public SectionsStatePagerAdapter pagerAdapter;
35      private ViewPager mViewPager;
36      private RelativeLayout mRelativeLayout;
37
```

Code: EditProfileFragment.java

AccountSettingsActivity.java × EditProfileFragment.java × SignOutFragment.java ×

EditProfileFragment

```
1  package com.connect.foodies.foodies.profile;
2
3  import ...
40
41 public class EditProfileFragment extends Fragment implements
42         ConfirmPasswordDialog.OnConfirmPasswordListener{
43
44
45     @Override
46     public void onConfirmPassword(String password) {
47         Log.d(TAG, "onConfirmPassword: got the password: " + password);
48
49         // Get auth credentials from the user for re-authentication. The example below shows
50         // email and password credentials but there are multiple possible providers,
51         // such as GoogleAuthProvider or FacebookAuthProvider.
52         AuthCredential credential = EmailAuthProvider
53                 .getCredential(mAuth.getCurrentUser().getEmail(), password);
```

Code: SignOutFragment.java

AccountSettingsActivity.java × EditProfileFragment.java × SignOutFragment.java ×

```
1  package com.connect.foodies.foodies.profile;
2
3  import ...
20
21 public class SignOutFragment extends Fragment {
22
23     private static final String TAG = "SignOutFragment";
24
25     //firebase
26     private FirebaseAuth mAuth;
27     private FirebaseAuth.AuthStateListener mAuthListener;
28
29     private ProgressBar mProgressBar;
30     private TextView tvSignout, tvSigningOut;
31
```

There are other fragments used in this project as well, such as Gallery and Photo Fragments are explained in 9. Media Section.

Animations used:

Slide as Gravity Left when clicking on Comments and when clicking on Post

```
public void onCommentThreadSelected(Photo photo, String callingActivity) {
    Log.d(TAG, "onCommentThreadSelected: selected a coemment thread");

    ViewCommentsFragment fragment = new ViewCommentsFragment();
    Bundle args = new Bundle();
    args.putParcelable("PHOTO", photo);
    args.putString("Home Activity", "Home Activity");
    fragment.setArguments(args);
    fragment.setEnterTransition(new Slide(Gravity.LEFT));
    FragmentTransaction transaction = getSupportFragmentManager().beginTransaction();
    transaction.replace(R.id.container, fragment);
    transaction.addToBackStack("View Comments");
    transaction.commit();

}
```

Shared element Activity transition:

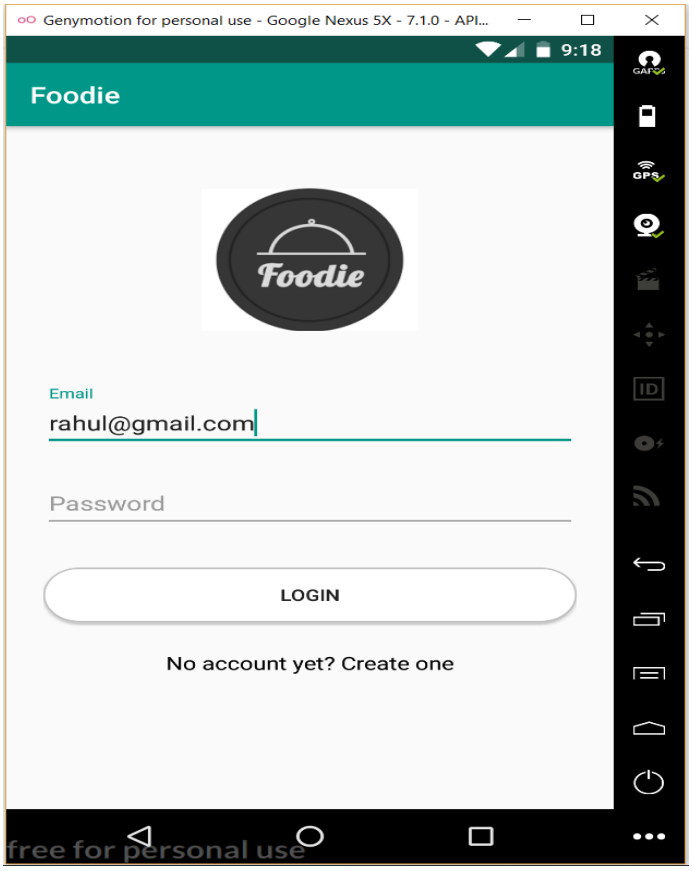
When Clicking on Floating Button Share Activity gets called.

```
<android.support.design.widget.FloatingActionButton
    android:id="@+id/fab"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_centerHorizontal="true"
    android:layout_gravity="bottom|center"
    android:layout_marginBottom="45dp"
    android:src="@drawable/floating"
    android:transitionName="testAnimation" />
```

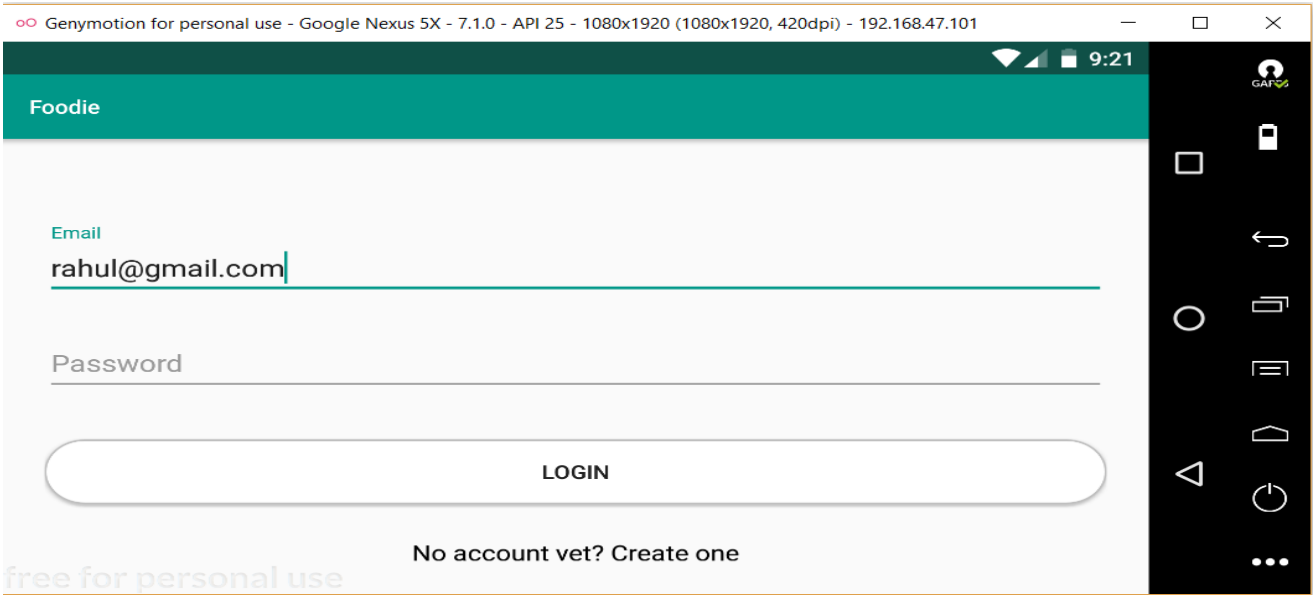
```
this.fab = (FloatingActionButton) findViewById(R.id.fab);
fab.setOnClickListener((view) → {
    Intent intent = new Intent(mContext, ShareActivity.class);
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
        ActivityOptionsCompat options = ActivityOptionsCompat.makeSceneTransitionAnimation(HomeActivity.this, view,
                "testAnimation");
        mContext.startActivity(intent, options.toBundle());
    } else {
        startActivity(intent);
    }
});
```

7. User/App State

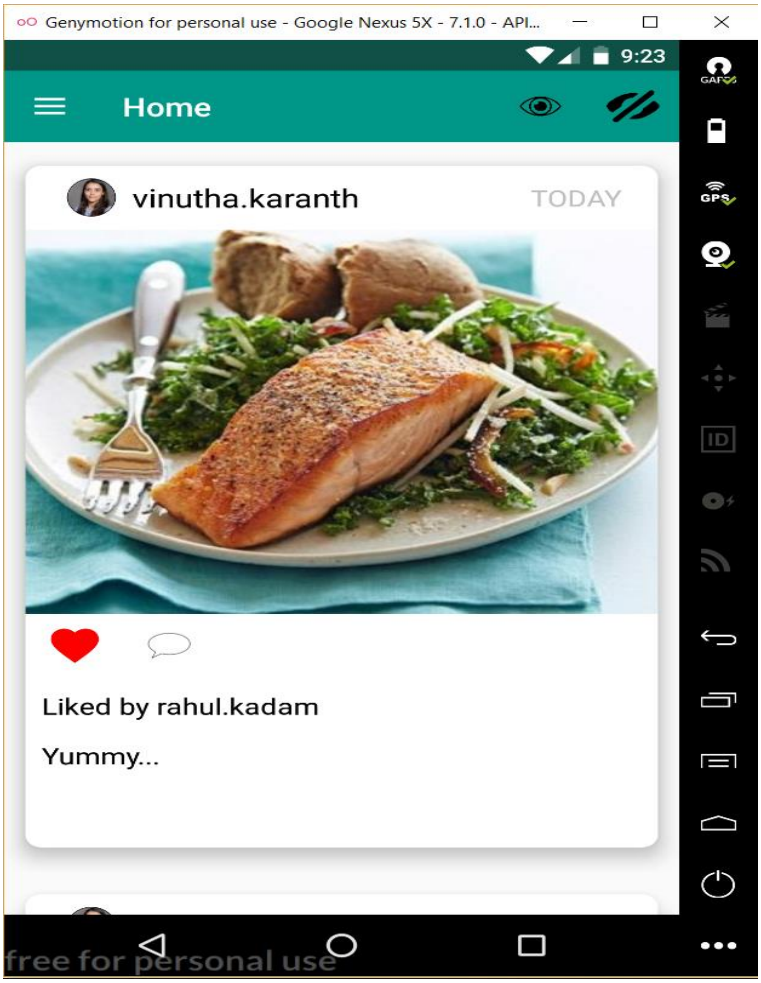
Login Normal Mode:



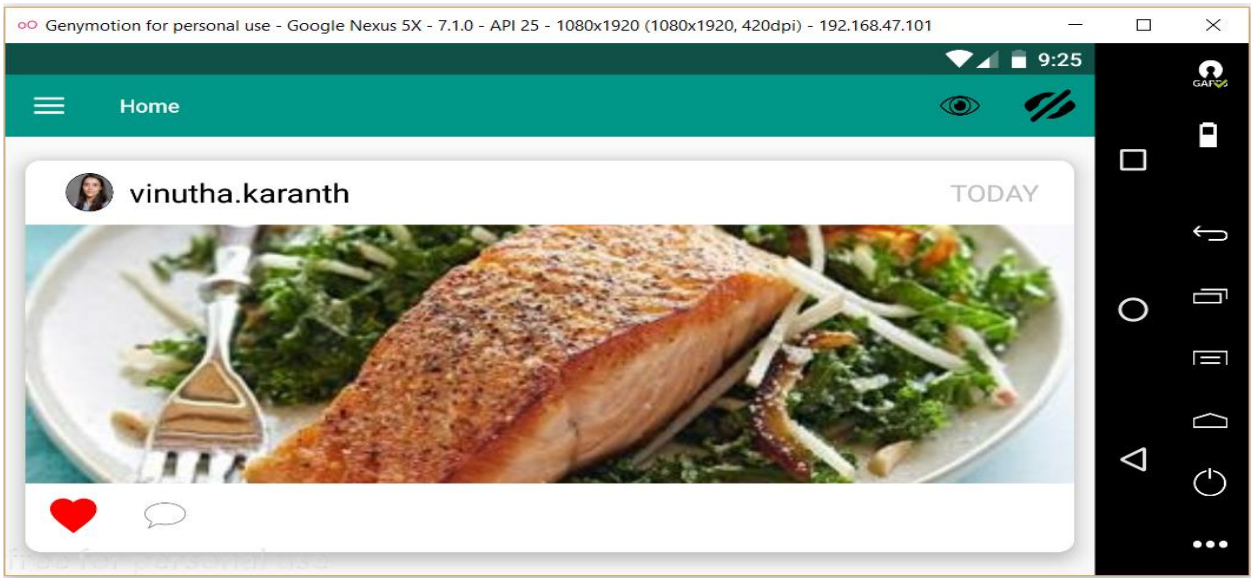
Login on Rotation:



Home Normal Mode:

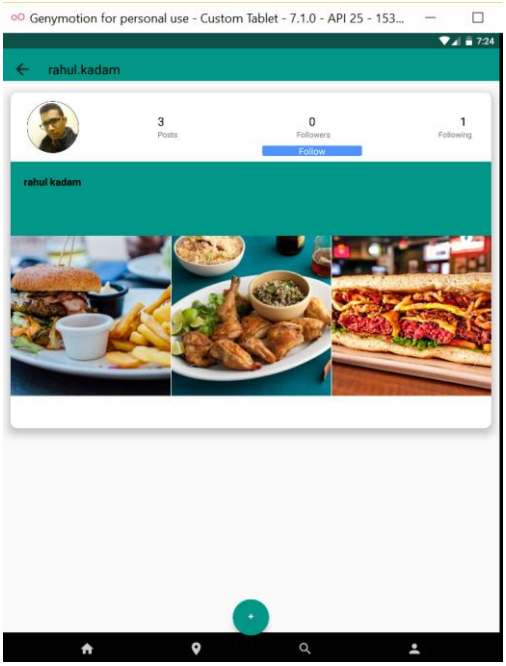


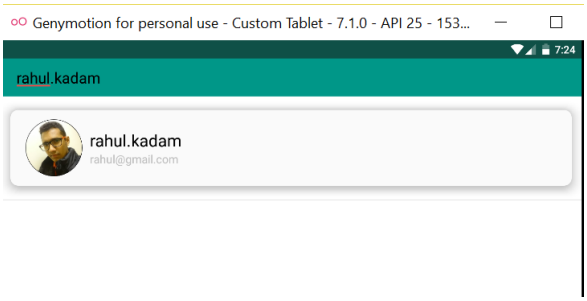
Home on Rotation:

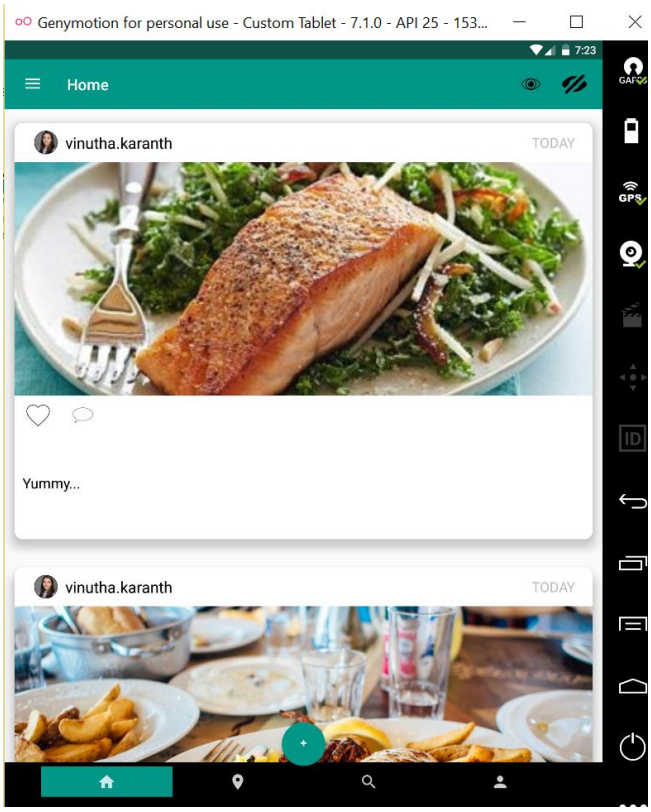


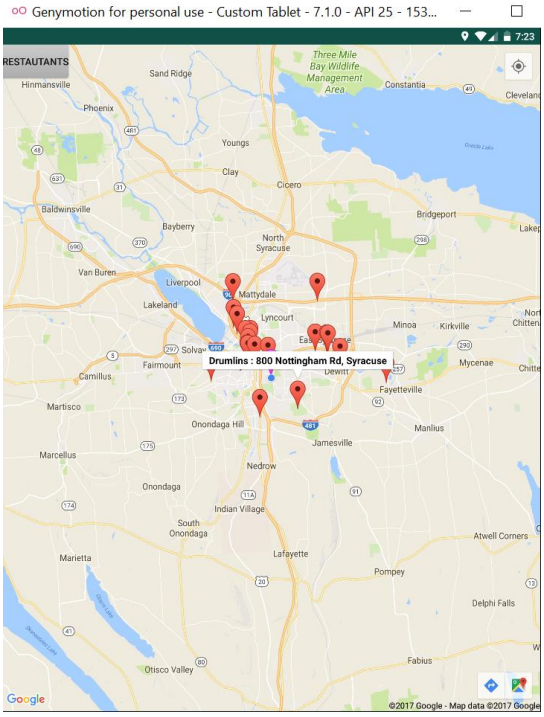
8. Multiple Devices/Screens & Orientation Changes using Constraint

App is tested on other devices like Custom Tablet 7.1.0 API 25 1536*2048 and all the screen are well organized and undisrupted.

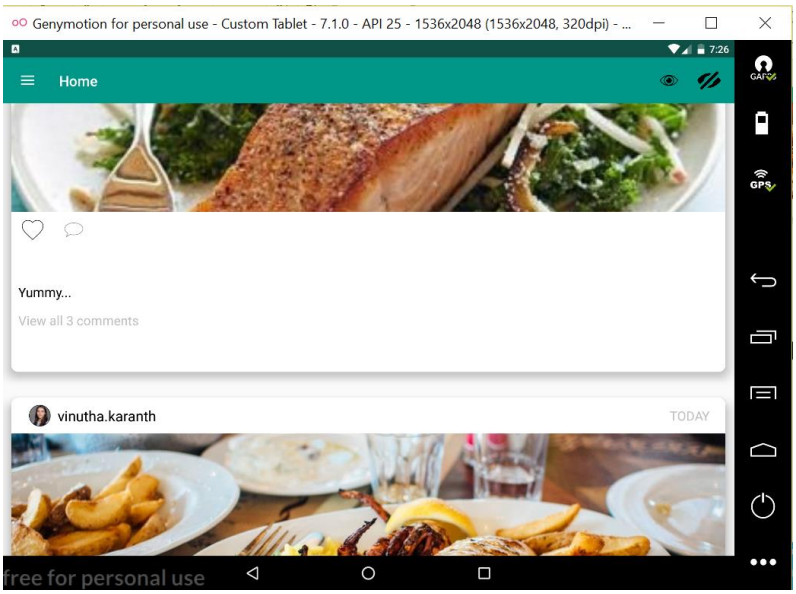






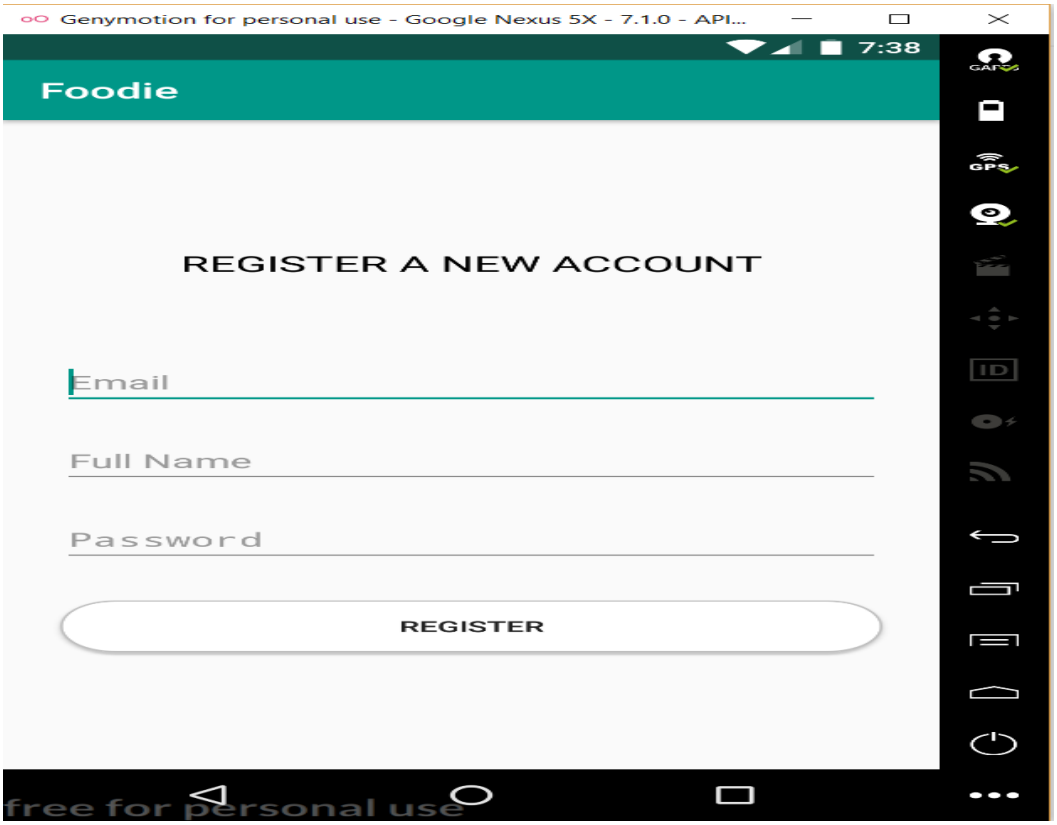


Screen Rotation in Tablet

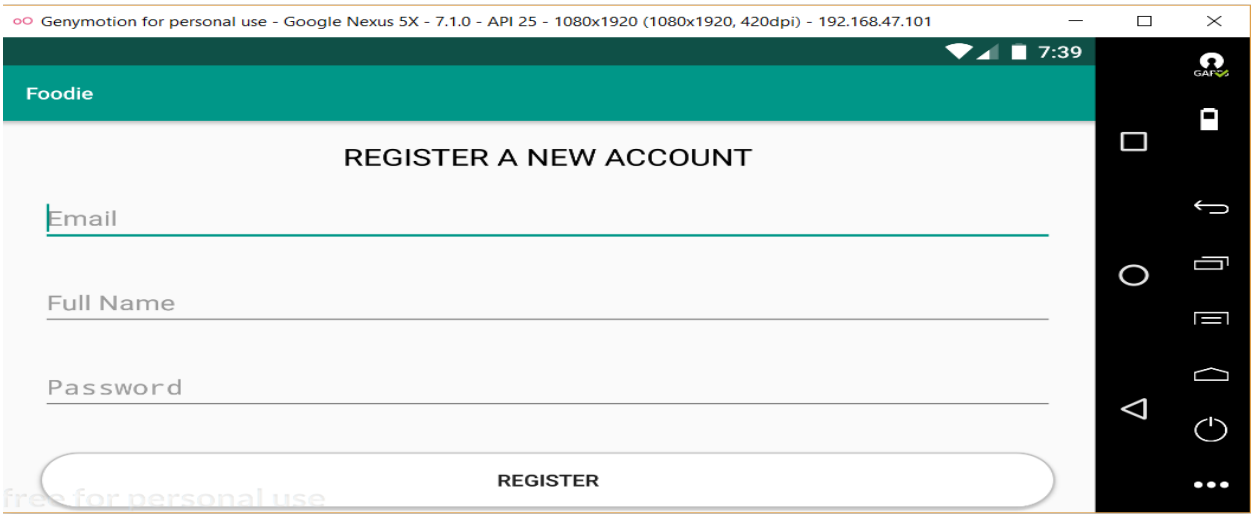


Orientation Changes using Constraint Layout:

Normal layout:



Landscape Layout:



Code: Normal layout:

layout\activity_register.xml × land\activity_register.xml ×

android.support.constraint.ConstraintLayout

```
1   <?xml version="1.0" encoding="utf-8"?>
2   <android.support.constraint.ConstraintLayout
3       xmlns:android="http://schemas.android.com/apk/res/android"
4       xmlns:app="http://schemas.android.com/apk/res-auto"
5       xmlns:tools="http://schemas.android.com/tools"
6       android:layout_width="match_parent"
7       android:layout_height="match_parent">
8
9       <TextView...>
27
28      <EditText...>
47
48      <EditText...>
65
66      <EditText...>
83
84      <android.support.v7.widget.AppCompatButton...>
102
103     <ProgressBar...>
117
118     <TextView...>
131
132 </android.support.constraint.ConstraintLayout>
133
```

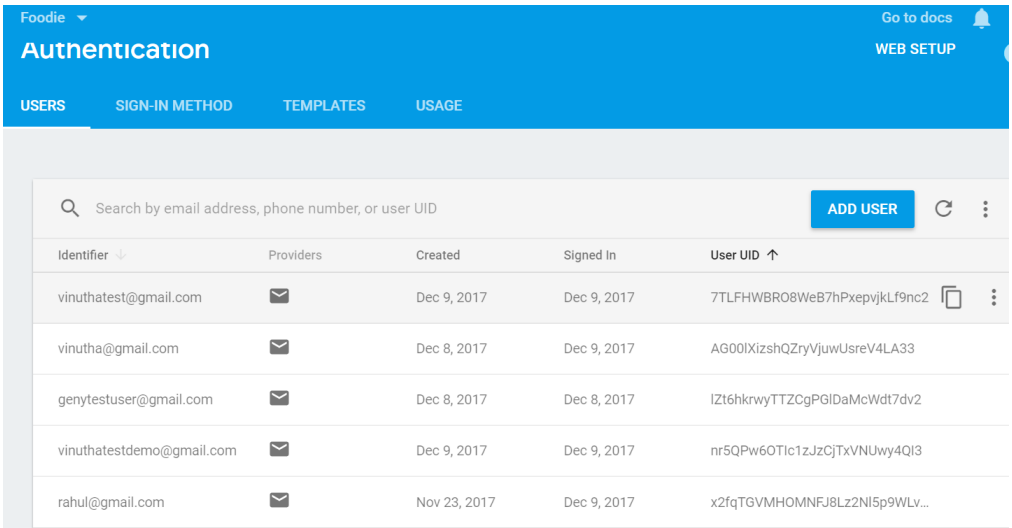
Code: Landscape Layout:

layout\activity_register.xml | land\activity_register.xml

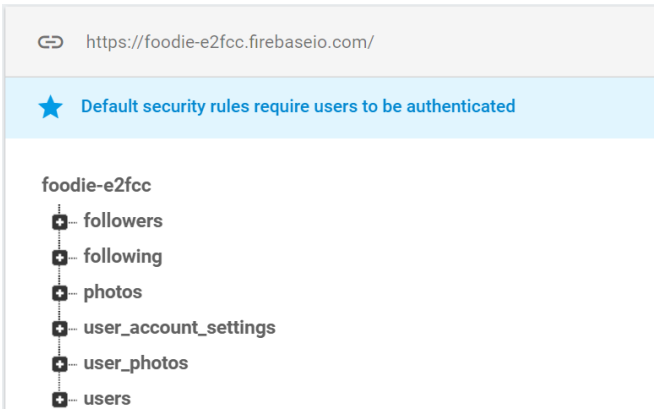
android.support.constraint.ConstraintLayout

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView...>

    <EditText...>

    <EditText...>

    <EditText...>

    <android.support.v7.widget.AppCompatButton...>

    <ProgressBar...>

    <TextView...>

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

9. Host your data in the Cloud

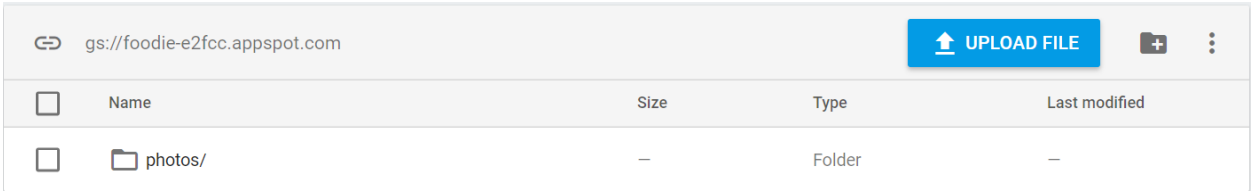


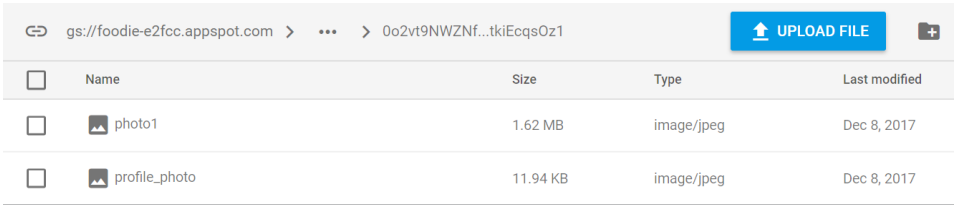
Nodes



UserAccount setting node

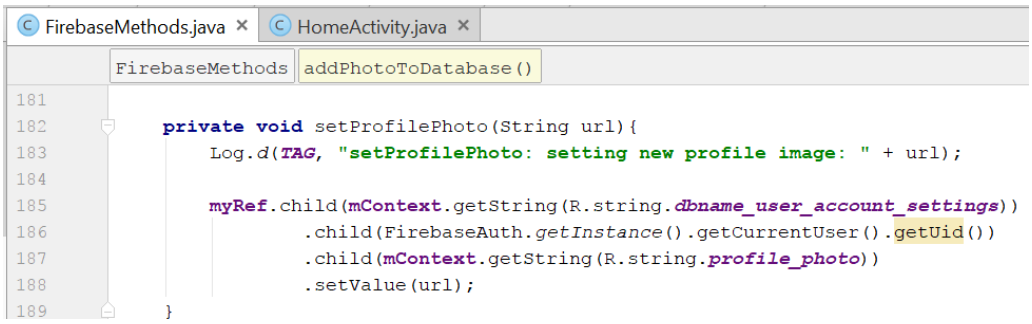
Storage

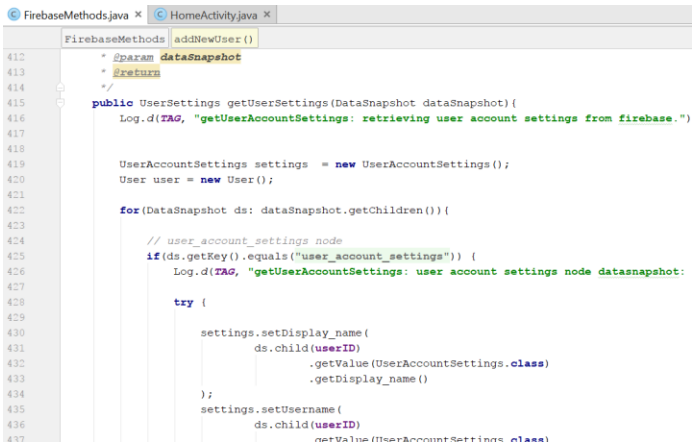




In the Java Code:

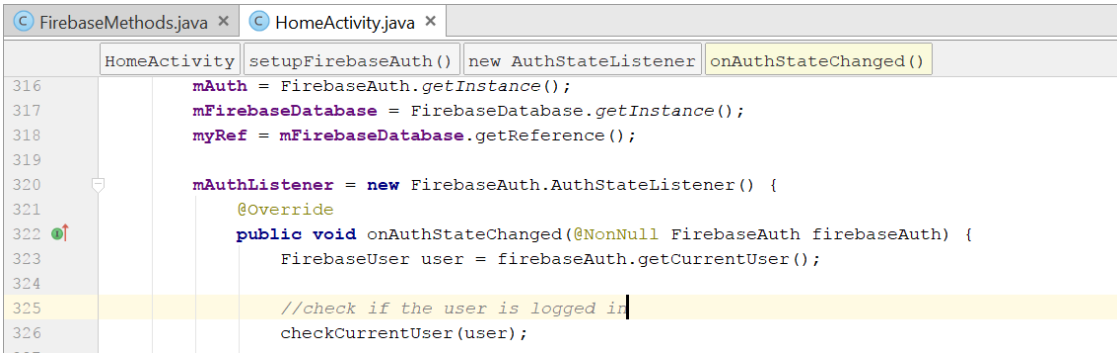
FirebaseMethods.java is the main Database file where all the database get and set activities are done. All the other activity or Fragment files calls the methods or function in the FirebaseMethods.java file by passing required arguments.

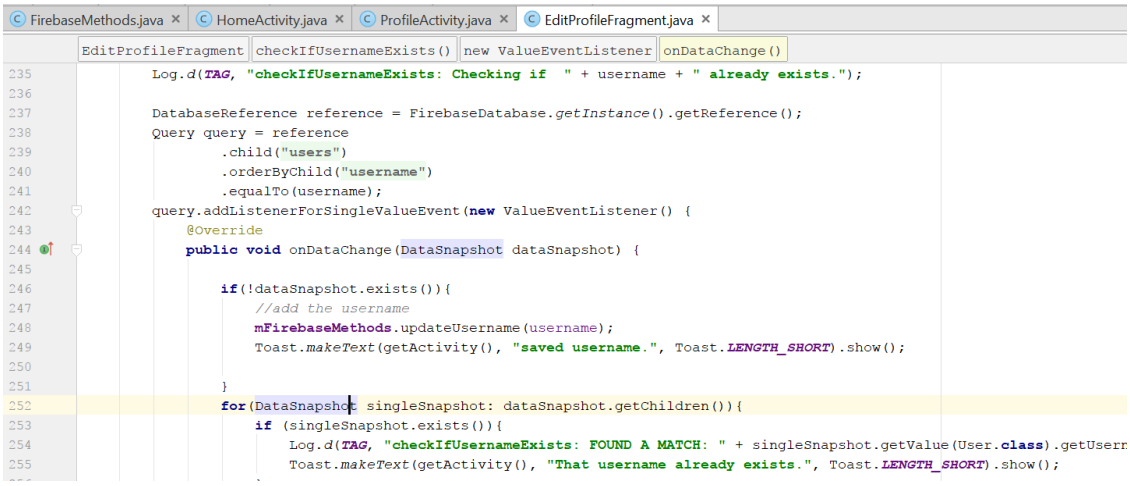




Database is used in almost all the activities and fragments.

Example HomeActivity:





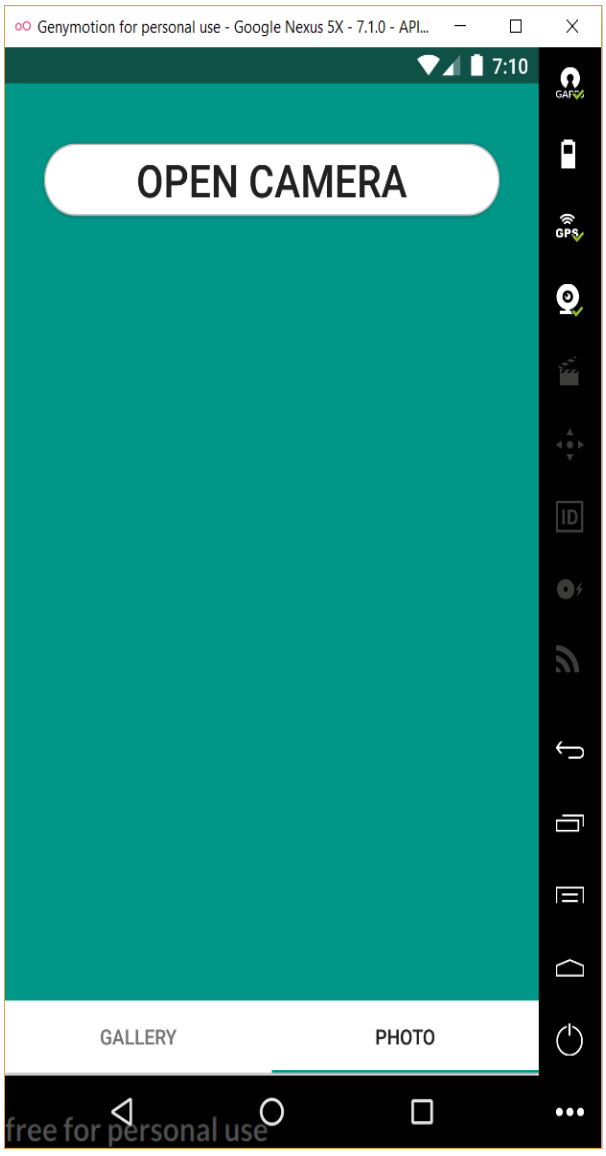
10. Media

Setting Up Permissions in AndroidManifest.xml

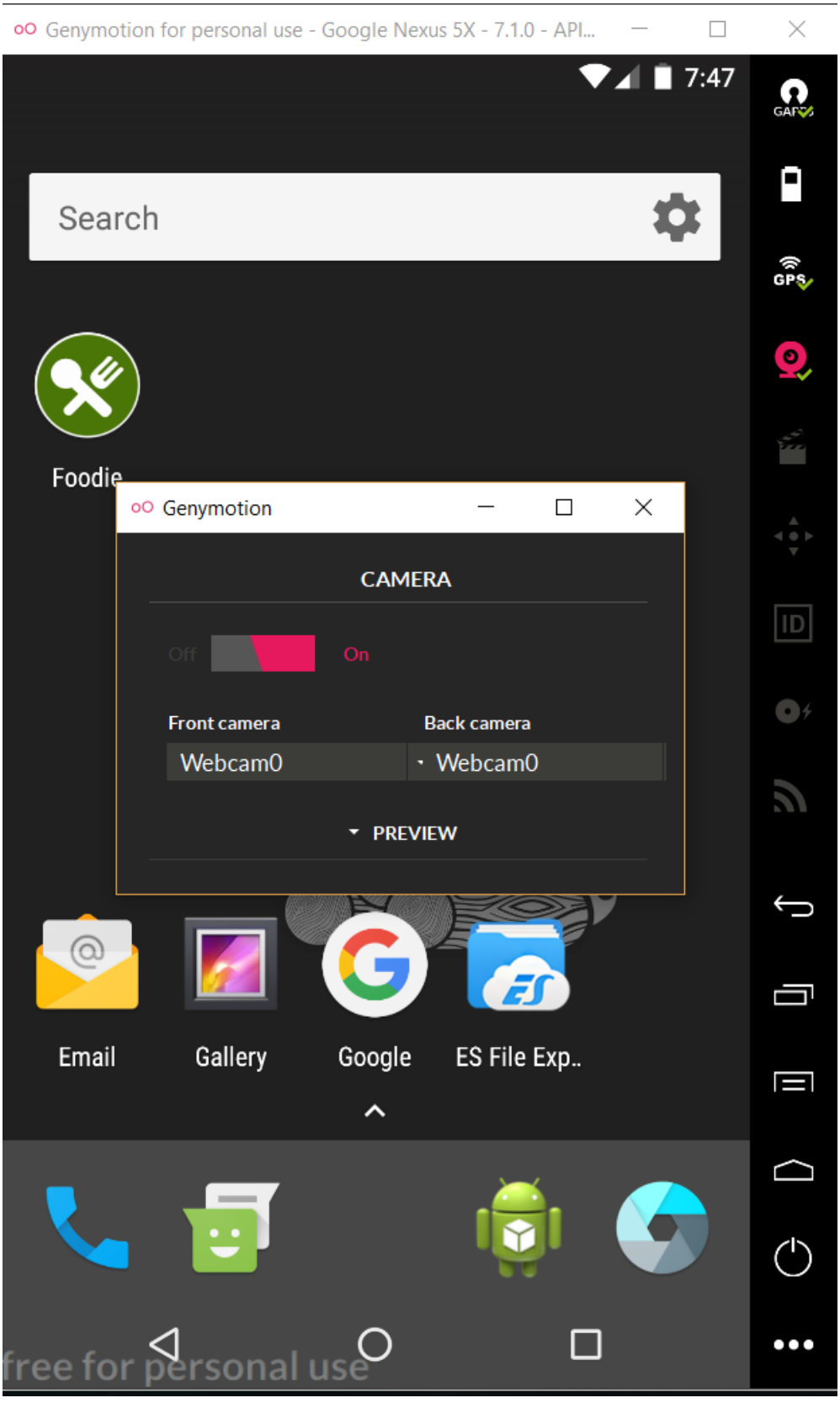
```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.CAMERA"/>
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
<uses-permission android:name="com.google.android.providers.gsf.permission.READ_GSERVICES"/>
<uses-permission android:name="com.connect.foodies.foodies.mapretroapi.mymapsappsdirection.permission.MAPS_RECEIVE" />
```

10.1 Camera

Camera Fragment:



Adding Camera Settings in Geny Motion:



Code: Some essential functions in class PhotoFragment.java

PhotoFragment.java ×

PhotoFragment | onCreateView()

```
1   package com.connect.foodies.foodies.share;
2
3   import ...
18
19  public class PhotoFragment extends Fragment {
20      private static final String TAG = "PhotoFragment";
21      //constant
22      private static final int PHOTO_FRAGMENT_NUM = 1;
23      private static final int GALLERY_FRAGMENT_NUM = 2;
24      private static final int CAMERA_REQUEST_CODE = 5;
25      @Nullable
26      @Override
27      public View onCreateView(final LayoutInflater inflater, @Nullable ViewGroup container,
28                               @Nullable Bundle savedInstanceState) {
29          View view = inflater.inflate(R.layout.fragment_photo, container, false);
30          Log.d(TAG, "onCreateView: started.");
31          Button btnLaunchCamera = (Button) view.findViewById(R.id.btnLaunchCamera);
32          btnLaunchCamera.setOnClickListener((v) -> {
35                  Log.d(TAG, "onClick: launching camera.");
36
37                  if (((ShareActivity) getActivity()).getCurrentTabNumber() == PHOTO_FRAGMENT_NUM) {
38                      if (((ShareActivity) getActivity()).checkPermissions(Permissions.CAMERA_PERMISSION[0])) {
39                          Log.d(TAG, "onClick: starting camera");
40                          Intent cameraIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
41                          startActivityForResult(cameraIntent, CAMERA_REQUEST_CODE);
42                      } else {
43                          Intent intent = new Intent(getActivity(), ShareActivity.class);
44                          intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_CLEAR_TASK);
45                          startActivity(intent);
46                      }
47                  }
48          });
50          return view;
51      }
```

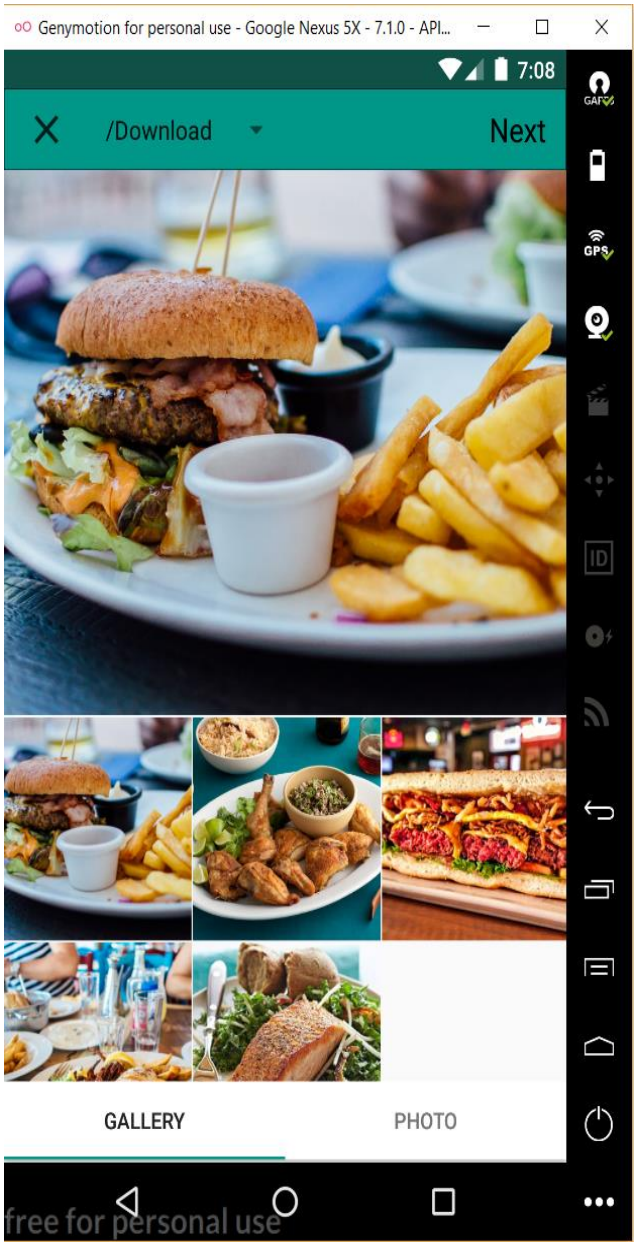
PhotoFragment.java ×

PhotoFragment | onActivityResult()

```
62      public void onActivityResult(int requestCode, int resultCode, Intent data) {
63          super.onActivityResult(requestCode, resultCode, data);
64          if (requestCode == CAMERA_REQUEST_CODE) {
65              if (data != null) {
66                  if (data.getExtras() != null) {
67                      Bitmap bitmap;
68                      bitmap = (Bitmap) data.getExtras().get("data");
69                      if (isRootTask()) {
70                          try {
71                              Log.d(TAG, "onActivityResult: received new bitmap from camera: " + bitmap);
72                              Intent intent = new Intent(getActivity(), NextActivity.class);
73                              intent.putExtra("selected_bitmap", bitmap);
74                              startActivity(intent);
75                          } catch (NullPointerException e) {
76                              Log.d(TAG, "onActivityResult: NullPointerException: " + e.getMessage());
77                          }
78                      } else {
79                          try {
80                              Log.d(TAG, "onActivityResult: received new bitmap from camera: " + bitmap);
81                              Intent intent = new Intent(getActivity(), AccountSettingsActivity.class);
82                              intent.putExtra("selected_bitmap", bitmap);
83                              intent.putExtra("return_to_fragment", "Edit Profile");
84                              startActivity(intent);
85                              getActivity().finish();
86                          } catch (NullPointerException e) {
87                              Log.d(TAG, "onActivityResult: NullPointerException: " + e.getMessage());
88                          }
89                      }
90                  } else {
91                      Log.d(TAG, "onActivityResult: Camera Cancelled");
92                  }
93              }
94          }
95      }
```

10.2 Gallery

Gallery Fragment



Code given on next page

Code: Some essential functions in class GalleryFragment.java

GalleryFragment.java

```
GalleryFragment | onCreateView()
1   package com.connect.foodies.foodies.share;
2   import ...
29  public class GalleryFragment extends Fragment {
30      private static final String TAG = "GalleryFragment";
31      //constants
32      private static final int NUM_GRID_COLUMNS = 3;
33      //widgets
34      private GridView gridView;
35      private ImageView galleryImage;
36      private ProgressBar mProgressBar;
37      private Spinner directorySpinner;
38      //vars
39      private ArrayList<String> directories;
40      private String mAppend = "file:/";
41      private String mSelectedImage;
42      @Nullable
43      @Override
```

GalleryFragment.java

```
GalleryFragment | onCreateView()
44  public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup container,
45                           @Nullable Bundle savedInstanceState) {
46      View view = inflater.inflate(R.layout.fragment_gallery, container, false);
47      galleryImage = (ImageView) view.findViewById(R.id.galleryImageView);
48      gridView = (GridView) view.findViewById(R.id.gridView);
49      directorySpinner = (Spinner) view.findViewById(R.id.spinnerDirectory);
50      mProgressBar = (ProgressBar) view.findViewById(R.id.progressBar);
51      mProgressBar.setVisibility(View.GONE);
52      directories = new ArrayList<>();
53      Log.d(TAG, "onCreateView: started.");
54      ImageView shareClose = (ImageView) view.findViewById(R.id.ivCloseShare);
55      shareClose.setOnClickListener((v) -> {
58              Log.d(TAG, "onClick: closing the gallery fragment.");
59              getActivity().finish();
60      });
62      TextView nextScreen = (TextView) view.findViewById(R.id.tvNext);
63      nextScreen.setOnClickListener((v) -> {
66              Log.d(TAG, "onClick: navigating to the final share screen.");
67
68              if (isRootTask()) {
69                  Intent intent = new Intent(getActivity(), NextActivity.class);
70                  intent.putExtra("selected_image", mSelectedImage);
71                  startActivity(intent);
72              } else {
73                  Intent intent = new Intent(getActivity(), AccountSettingsActivity.class);
74                  intent.putExtra("selected_image", mSelectedImage);
75                  intent.putExtra("return_to_fragment", "Edit Profile");
76                  startActivity(intent);
77                  getActivity().finish();
78              }
79      });
81      init();
82      return view;
83  }
```

GalleryFragment.java

```
GalleryFragment | init() | new OnItemSelectedListener
91      private void init() {
92          FilePaths filePaths = new FilePaths();
93
94          //check for other folders inside "/storage/emulated/0/pictures"
95          if (FileSearch.getDirectoryPaths(filePaths.PICTURES) != null) {
96              directories = FileSearch.getDirectoryPaths(filePaths.PICTURES);
97          }
98          directories.add(filePaths.DOWNLOAD);
99          directories.add(filePaths.CAMERA);
100         ArrayList<String> directoryNames = new ArrayList<>();
101         for (int i = 0; i < directories.size(); i++) {
102             Log.d(TAG, "init: directory: " + directories.get(i));
103             int index = directories.get(i).lastIndexOf("/");
104             String string = directories.get(i).substring(index);
105             directoryNames.add(string);
106         }
107         ArrayAdapter<String> adapter = new ArrayAdapter<~>(getActivity(),
108                 android.R.layout.simple_spinner_item, directoryNames);
109         adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
110         directorySpinner.setAdapter(adapter);
111         directorySpinner.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
112             @Override
113             public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {
114                 Log.d(TAG, "onItemClick: selected: " + directories.get(position));
115
116                 //setup our image grid for the directory chosen
117                 setupGridView(directories.get(position));
118             }
119             @Override
120             public void onNothingSelected(AdapterView<?> parent) {
121
122             }
123         });
124     }
```

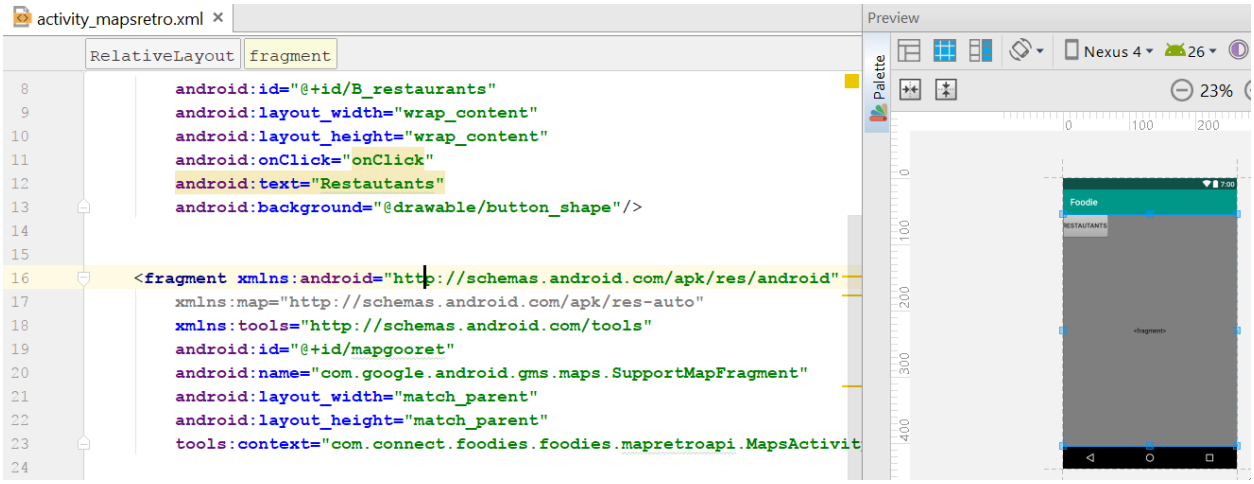
GalleryFragment.java

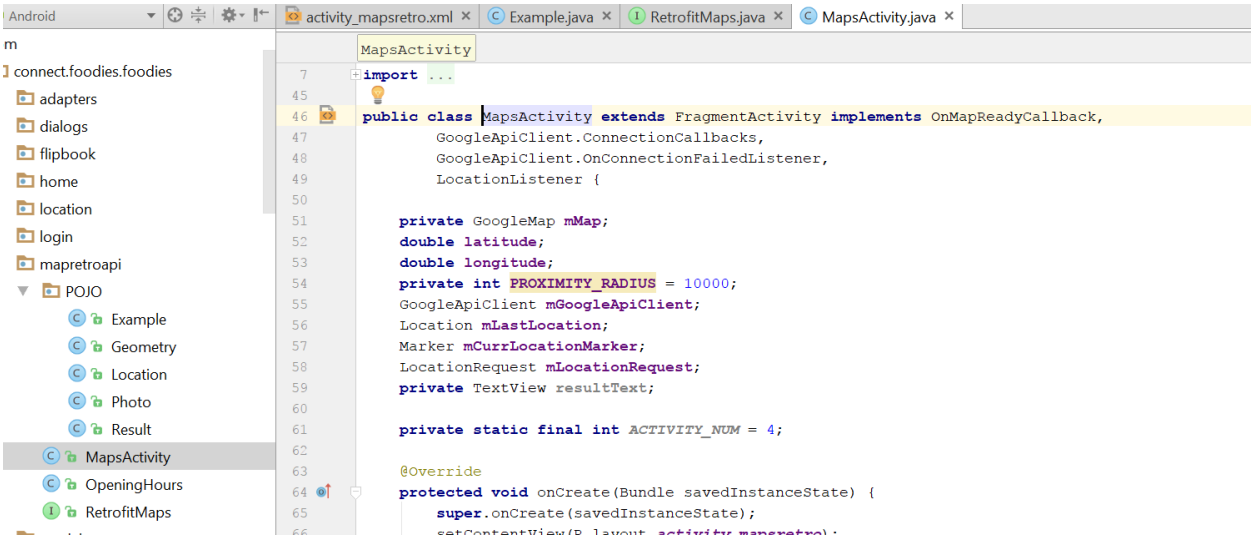
```
GalleryFragment | setupGridView()
125     private void setupGridView(String selectedDirectory) {
126         Log.d(TAG, "setupGridView: directory chosen: " + selectedDirectory);
127         final ArrayList<String> imgURLs = FileSearch.getFilePaths(selectedDirectory);
128         //set the grid column width
129         int gridWidth = getResources().getDisplayMetrics().widthPixels;
130         int imageWidth = gridWidth / NUM_GRID_COLUMNS;
131         gridView.setColumnWidth(imageWidth);
132         if (imgURLs != null) {
133             if (imgURLs.size() != 0) {
134                 //use the grid adapter to adapter the images to gridview
135                 GridImageAdapter adapter = new GridImageAdapter(getActivity(), R.layout.layout_grid_imageview, mAppend, imgURLs);
136                 gridView.setAdapter(adapter);
137                 //set the first image to be displayed when the activity fragment view is inflated
138                 try {
139                     setImage(imgURLs.get(0), galleryImage, mAppend);
140                     mSelectedImage = imgURLs.get(0);
141                 } catch (ArrayIndexOutOfBoundsException e) {
142                     Log.e(TAG, "setupGridView: ArrayIndexOutOfBoundsException: " + e.getMessage());
143                 }
144                 gridView.setOnItemClickListener((parent, view, position, id) -> {
147                         Log.d(TAG, "onItemClick: selected an image: " + imgURLs.get(position));
148                         setImage(imgURLs.get(position), galleryImage, mAppend);
149                         mSelectedImage = imgURLs.get(position);
150                 });
152             } else {
153                 Log.d(TAG, "setupGridView: No Images In Download Folder");
154             }
155         } else {
156             Log.d(TAG, "setupGridView: No Images In Download Folder");
157         }
158     }
```

11. Location

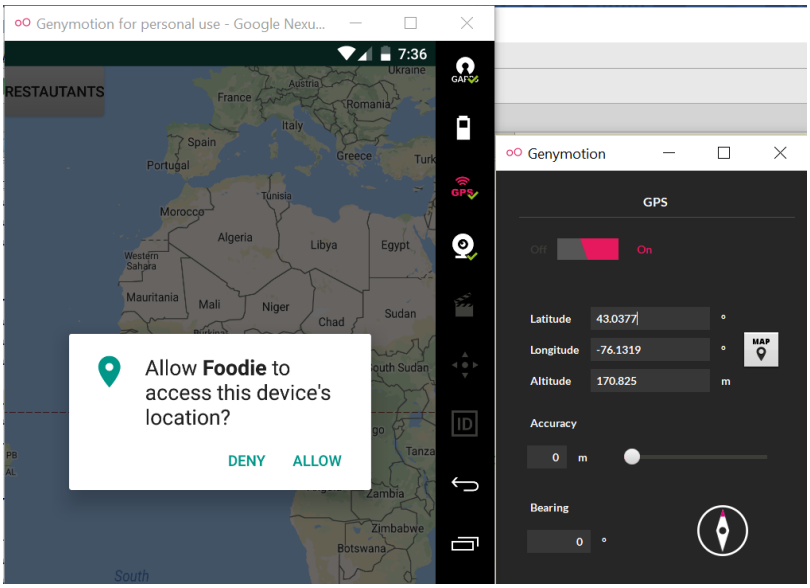
11.1 Map

Longitude and Latitude is received from the retrofit URL in json format. All the acquired data will be set on the map with the markers.

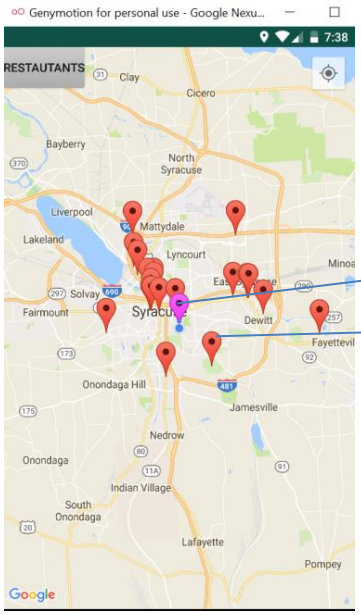




```
MapsActivity.java ×
MapsActivity

138
139     }
140
141     private void build_retrofit_and_get_response(String type) {
142
143         String url = "https://maps.googleapis.com/maps/";
144
145         Retrofit retrofit = new Retrofit.Builder()
146                 .baseUrl(url)
147                 .addConverterFactory(GsonConverterFactory.create())
148                 .build();
149
150         RetrofitMaps service = retrofit.create(RetrofitMaps.class);
151
152         Call<Example> call = service.getNearbyPlaces(type, latitude + "," + longitude, PROXIMITY_RAD
153
154         call.enqueue(new Callback<Example>() {
155             @Override
156             public void onResponse(Response<Example> response, Retrofit retrofit) {
157
158                 try {
159                     mMap.clear();
160                     // This loop will go through all the results and add marker on each location.
161                     for (int i = 0; i < response.body().getResults().size(); i++) {
162                         Double lat = response.body().getResults().get(i).getGeometry().getLocation()
163                         Double lng = response.body().getResults().get(i).getGeometry().getLocation()
164                         String placeName = response.body().getResults().get(i).getName();
```



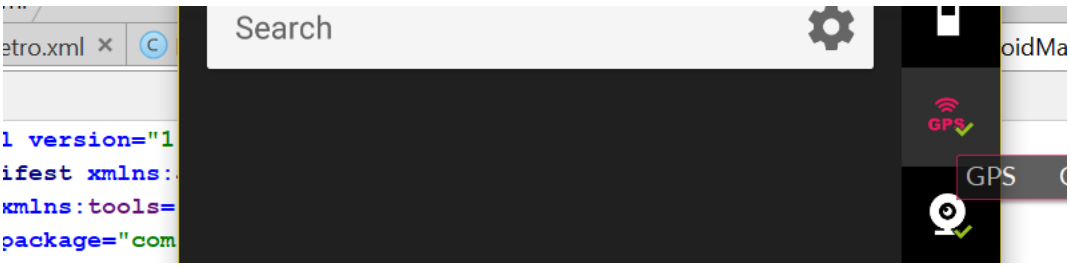
On clicking Allow; Go back and select Location tab again to get current location and click on Restaurant button to view all the nearby Restaurants.



Current Location will be shown in HUE_MAGENTA
All the Nearby Restaurants will be shown in HUE_RED

11.2 GPS

For Current Location we enable GPS location. And the current location will be shown in HUE_MAGENTA




```
 */
@Override
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;
    // mMap.setMapType(GoogleMap.MAP_TYPE_HYBRID);

    //Initialize Google Play Services
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
        if (ContextCompat.checkSelfPermission(this,
                Manifest.permission.ACCESS_FINE_LOCATION)
                == PackageManager.PERMISSION_GRANTED) {
            buildGoogleApiClient();
            mMap.setMyLocationEnabled(true);
```

12. Advanced Animation - Flipbook Style

AnimationFactory.java × FlipAnimation.java ×

FlipAnimation

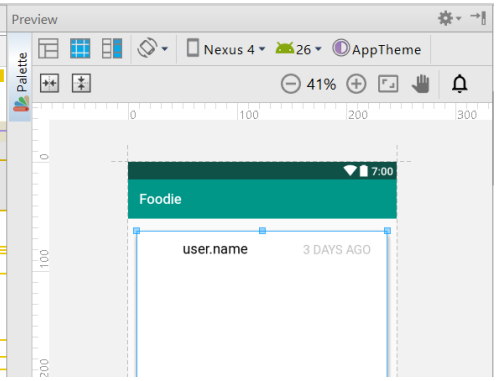
```
37   */
38  public class FlipAnimation extends Animation {
39      private final float mFromDegrees;
40      private final float mToDegrees;
41      private final float mCenterX;
42      private final float mCenterY;
43      private Camera mCamera;
44
45      private final ScaleUpDownEnum scaleType;
46
47      /**
48       * How much to scale up/down. The default scale of 75% of full size seems optimal based on testing
49       */
50      public static final float SCALE_DEFAULT = 0.75f;
51
52      private float scale;
```

AnimationFactory.java × ViewPostFragment.java × layout_view_post.xml ×

RelativeLayout | android.support.v7.widget.CardView | ViewFlipper

```
12      app:cardUseCompatPadding="true">
13
14      <ViewFlipper
15          android:id="@+id/viewFlipper"
16          android:layout_width="match_parent"
17          android:layout_height="match_parent"
18          android:gravity="center"
19          android:layout_gravity="center"
20          android:addStatesFromChildren="true"
21          android:layout_centerInParent="true"
22          android:clickable="true">
23      <RelativeLayout
24          android:layout_width="wrap_content"
25          android:layout_height="wrap_content"
```

Preview



AnimationFactory.java × ViewPostFragment.java ×

ViewPostFragment | onCreateView()

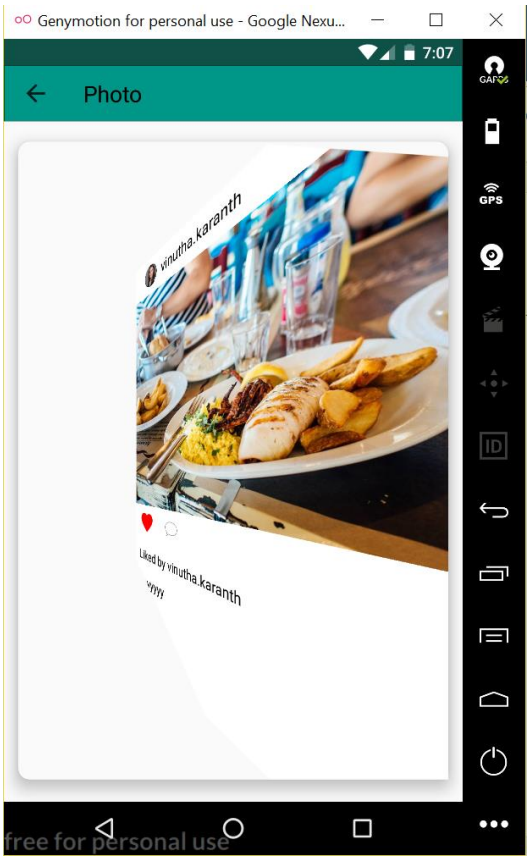
```
117         mComment = (ImageView) view.findViewById(R.id.speech_bubble);
118         mComments = (TextView) view.findViewById(R.id.image_comments_link);
119
120     mViewAnimator = (ViewAnimator) view.findViewById(R.id.viewFlipper);
```

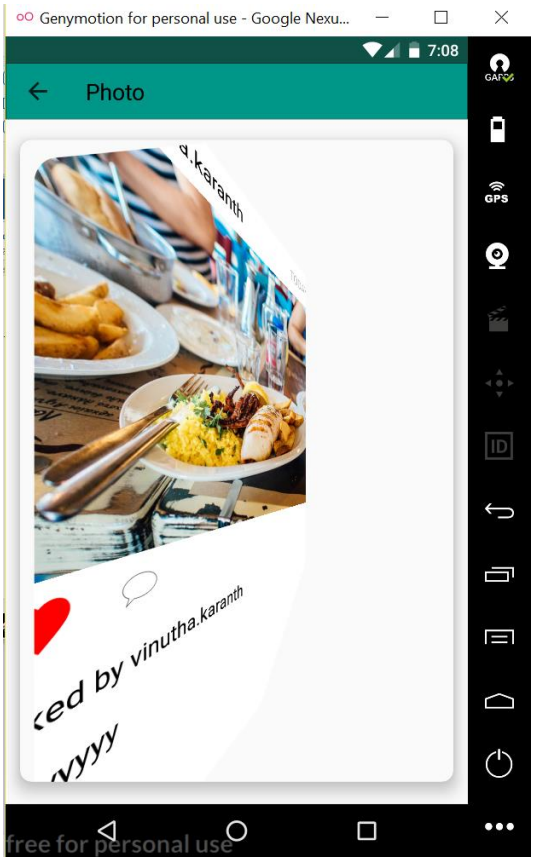
AnimationFactory.java × ViewPostFragment.java ×

ViewPostFragment onCreateView()

```
470        mComment.setOnClickListener((v) -> {
473                Log.d(TAG, "onClick: navigating back");
474                mOnCommentThreadSelectedListener.onCommentThreadSelectedListener(mPhoto);
475
476        });
478
479        mPostImage.setOnClickListener(new View.OnClickListener(){
480            @Override
481            public void onClick(View view) {
482                Log.d(TAG, "onClick: Image clicked");
483                AnimationFactory.flipTransition(mViewAnimator, AnimationFactory.FlipDirection.LEFT_RIGHT);
484
485            }
486        });
```

Animation in APP for ViewPostFragment

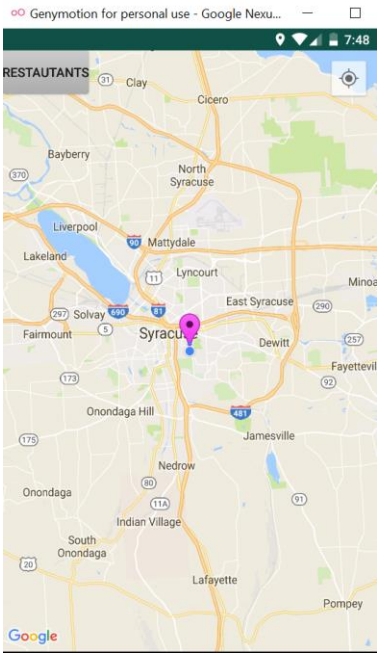


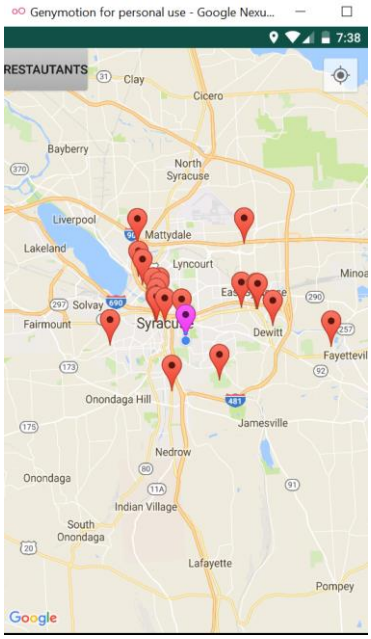


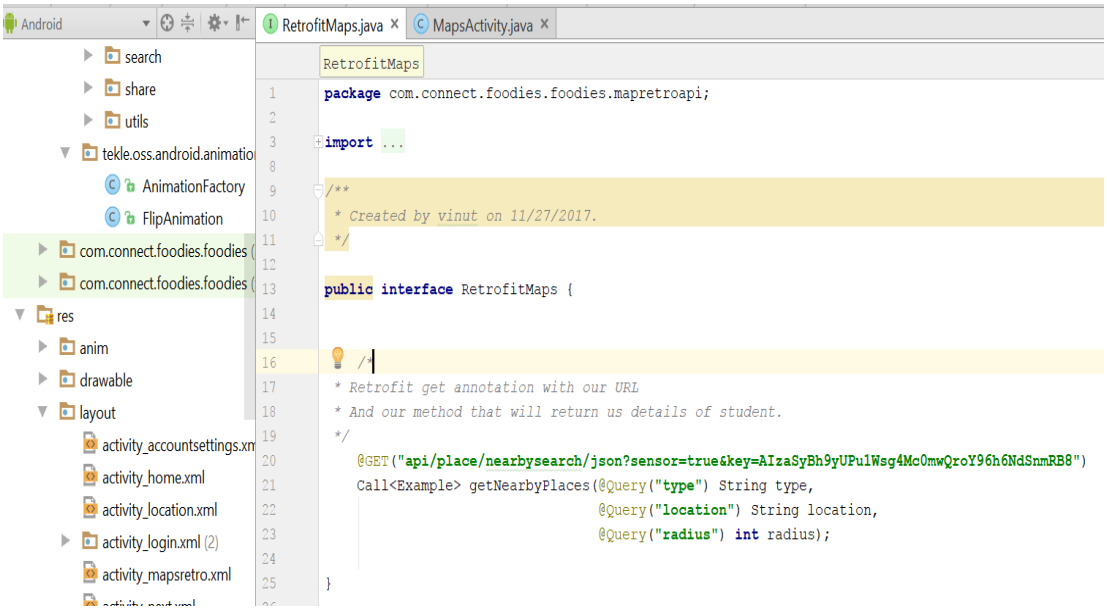
13. Data

13.1 Real API Data and Retrofit Library

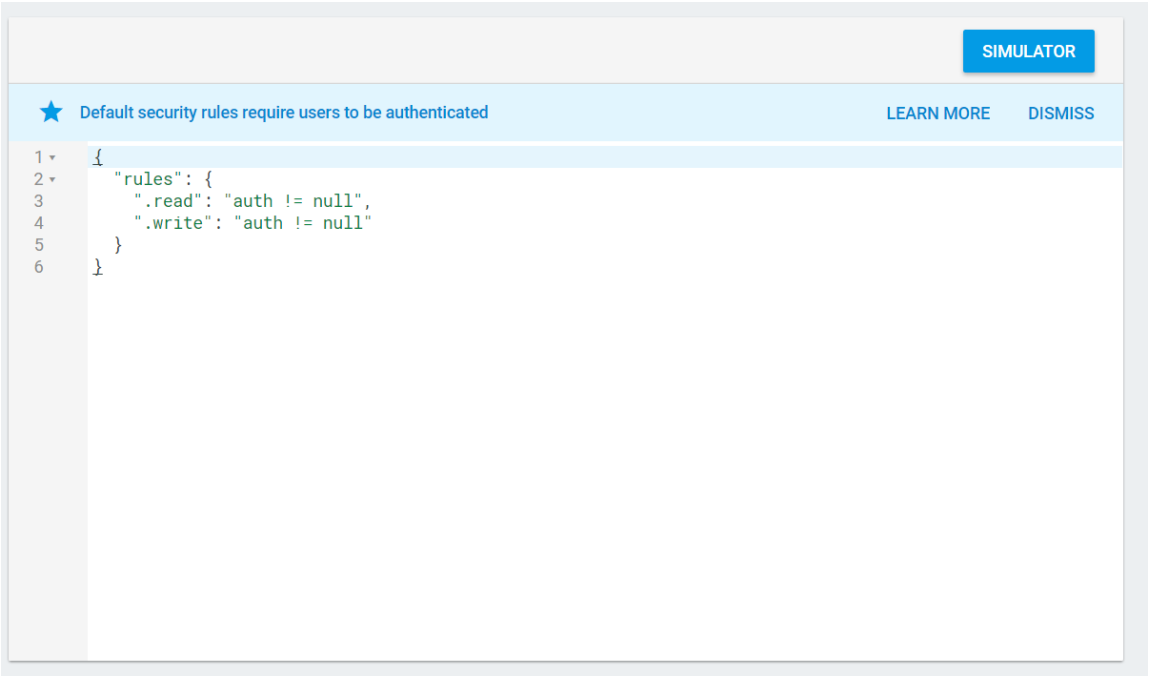
App uses Google Maps API using Retrofit. We have used Retrofit to capture data from URL. Uses POJO classes to retrieve data from the source.







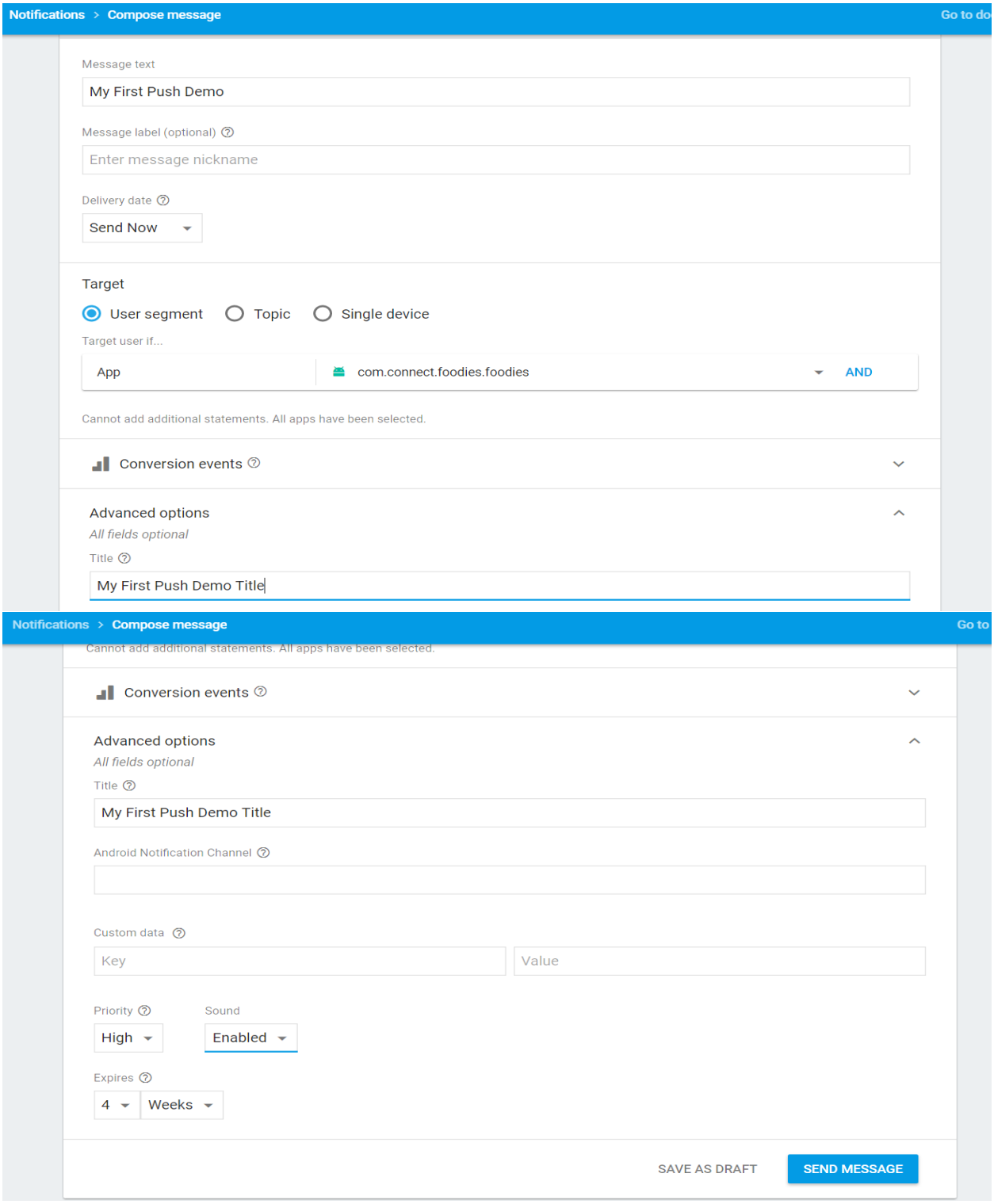
14. Firebase Security



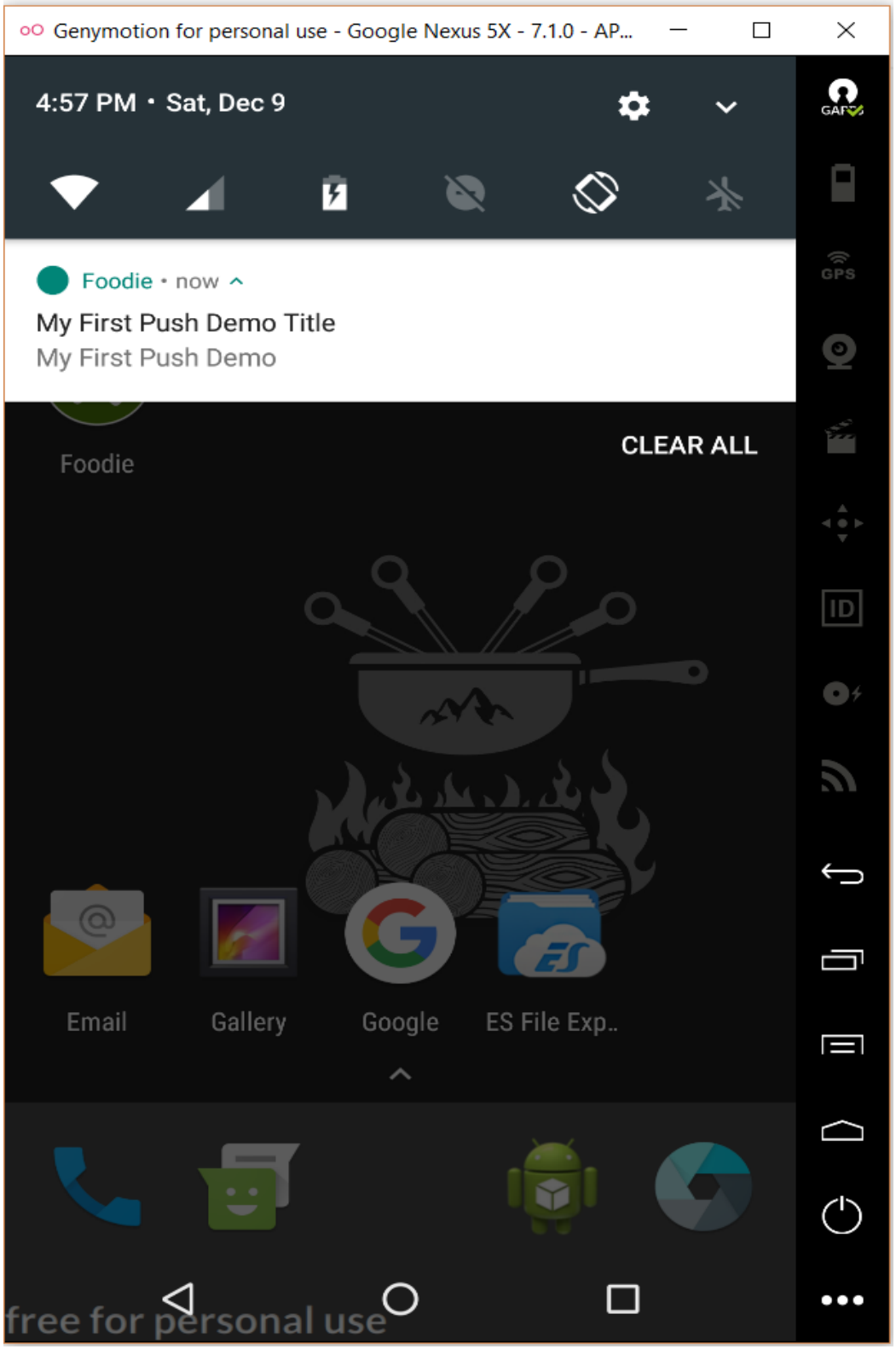
15. System - Service

We have implemented Firebase Push Notifications as part of Service.

1. Creating a Push Notification in Firebase:



2. App Screen:


Genymotion for personal use - Google Nexus 5X - 7.1.0 - AP...
4:57 PM • Sat, Dec 9
Foodie • now
My First Push Demo Title
My First Push Demo
Foodie
CLEAR ALL
Email
Gallery
Google
ES File Exp..
free for personal use

3. AndroidManifest.xml information added

AndroidManifest.xml

```
81              android:name=".share.ShareActivity"
82              android:label="Share"
83              android:theme="@style/AppTheme.NoActionBar"/>
84          <activity
85              android:name=".share.NextActivity"
86              android:label="Next"
87              android:theme="@style/AppTheme.NoActionBar"/>
88
89
90          <activity android:name=".mapretroapi.MapsActivity"
91              android:configChanges="orientation|screenSize"
92              android:theme="@style/AppTheme.NoActionBar"
93              android:label="MapsgooRet"/>
94
95          <service
96              android:name=".utils.FirebaseMessagingService">
97              <intent-filter>
98                  <action android:name="com.google.firebase.MESSAGING_EVENT"/>
99              </intent-filter>
100         </service>
101
102         <meta-data
103             android:name="com.google.firebase.messaging.default_notification_color"
104             android:resource="@color/colorAccent" />
105
106
107     </application>
108
109 </manifest>
```

2. FirebaseMessagingService.java – File used for push Notifications.

FirebaseMessagingService.java

```
package com.connect.foodies.foodies.utils;
import ...
public class FirebaseMessagingService extends com.google.firebase.messaging.FirebaseMessagingService {
    private static final String TAG = "FirebaseMessagingServic";
    public FirebaseMessagingService() {
    }
    @Override
    public void onMessageReceived(RemoteMessage remoteMessage) {
        String title = remoteMessage.getNotification().getTitle();
        String message = remoteMessage.getNotification().getBody();
        Log.d(TAG, "onMessageReceived: Message Received: \n" +
                "Title: " + title + "\n" +
                "Message: " + message);

        sendNotification(title, message);
    }
    @Override
    public void onDeletedMessages() {
    }
    private void sendNotification(String title, String messageBody) {
        Intent intent = new Intent(this, HomeActivity.class);
        intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        PendingIntent pendingIntent = PendingIntent.getActivity(this, 0 /* Request code */, intent,
                PendingIntent.FLAG_ONE_SHOT);
        Uri defaultSoundUri = RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);
        NotificationCompat.Builder notificationBuilder = new NotificationCompat.Builder(this)
                .setSmallIcon(R.mipmap.ic_launcher)
                .setContentTitle(title)
                .setContentText(messageBody)
                .setAutoCancel(true)
                .setSound(defaultSoundUri)
                .setContentIntent(pendingIntent);
        NotificationManager notificationManager =
                (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);
        notificationManager.notify(0 /* ID of notification */, notificationBuilder.build());
    }
}
```

16. Misc.

16.1 Shortcuts

AnimationFactory.java × ViewPostFragment.java × AndroidManifest.xml ×

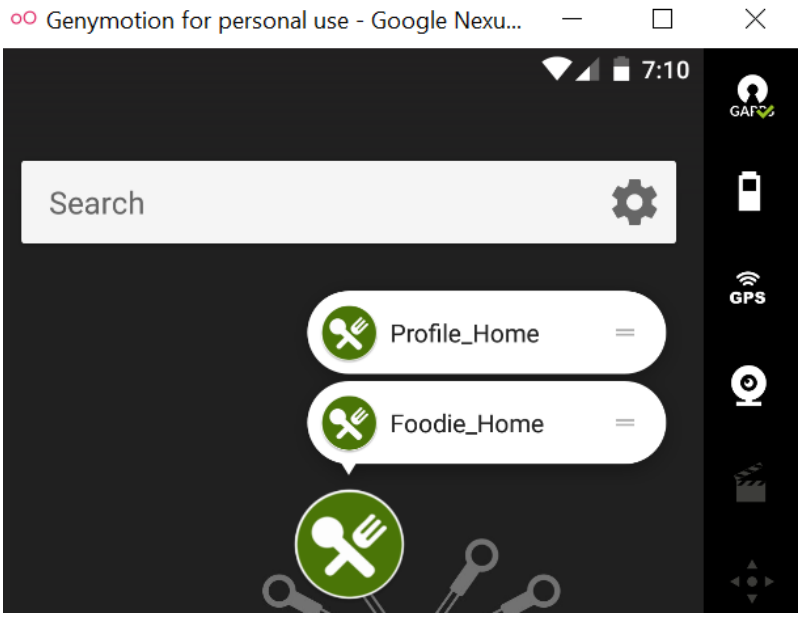
manifest | application | activity | meta-data

```
46          <meta-data
47              android:name="android.app.shortcuts"
48              android:resource="@xml/shortcuts" />
49      </activity>
```

AnimationFactory.java × ViewPostFragment.java × AndroidManifest.xml × shortcuts.xml ×

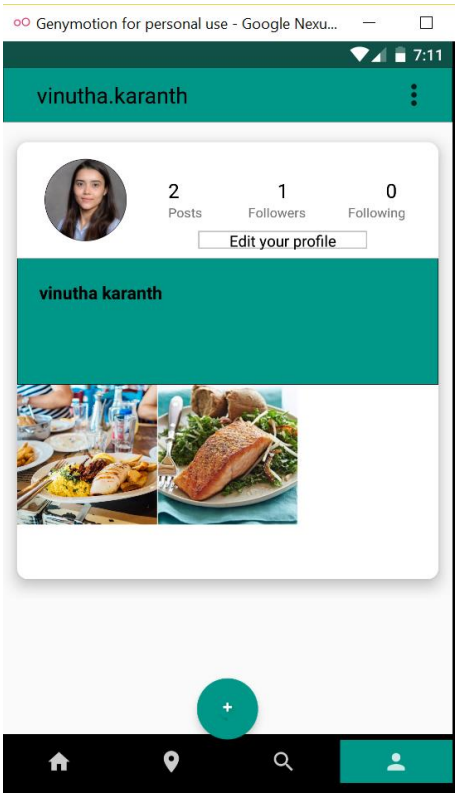
```
1   <?xml version="1.0" encoding="utf-8"?>
2   <shortcuts xmlns:android="http://schemas.android.com/apk/res/android">
3   <shortcut
4       android:shortcutId="Foodie_Home"
5       android:enabled="true"
6       android:icon="@mipmap/ic_launcher"
7       android:shortcutShortLabel="Foodie_Home"
8       android:shortcutLongLabel="Foodie_Home">
9       <intent
10          android:action="android.intent.action.VIEW"
11          android:targetPackage="com.connect.foodies.foodies"
12          android:targetClass="com.connect.foodies.foodies.home.HomeActivity" ></intent>
13      <categories android:name="android.shortcut.conversation"/>
14  </shortcut>
15      <shortcut
16          android:shortcutId="Profile_Home"
17          android:enabled="true"
18          android:icon="@mipmap/ic_launcher"
19          android:shortcutShortLabel="Profile_Home"
20          android:shortcutLongLabel="Profile_Home">
21          <intent
22              android:action="android.intent.action.VIEW"
23              android:targetPackage="com.connect.foodies.foodies"
24              android:targetClass="com.connect.foodies.foodies.profile.ProfileActivity"></intent>
25          <categories android:name="android.shortcut.conversation"/>
26      </shortcut>
27  </shortcuts>
```

Shortcut display on long pressing on app icon

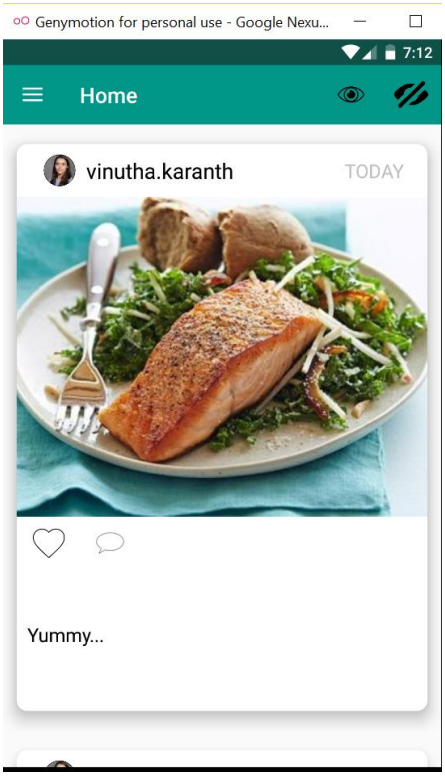


If User is Logged in:

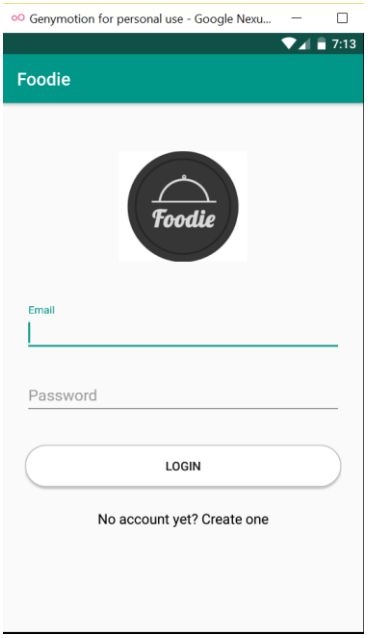
On clicking Profile_Home:



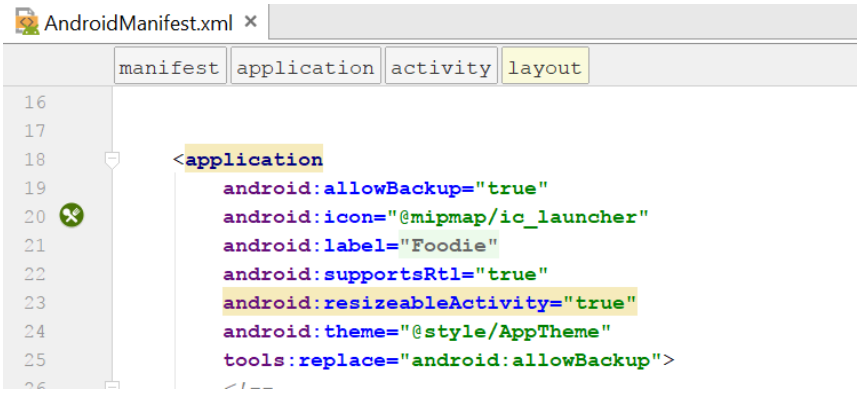
On clicking Foodie_Home:



When User is Logged Out both short cut takes user to Login Screen, App will not have any data backstacked:



16.2 Multi-Window



```
<activity
    android:name=".home.HomeActivity"
    android:label="Home"
    android:theme="@style/AppTheme.NoActionBar">
    <layout android:defaultHeight="500dp"
        android:defaultWidth="600dp"
        android:gravity="top|end"
        android:minHeight="450dp"
        android:minWidth="300dp" />

</activity>
```

MultiWindow Demo in App (Both the apps works simultaneously):