2. Starting new Project

1. Create a new project in **Android Studio** from **File ⇒ New Project** by filling the required details. When it prompts you to select the activity, choose **Empty Activity** and continue.

2. Open **build.gradle** located in app level and add below dependencies. I am adding Glide image library dependency. This is not needed for navigation drawer, but to load the profile image from url.

build.gradle

```
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    compile 'com.android.support:appcompat-v7:24.0.0-beta1'
    compile 'com.android.support:design:24.0.0-beta1'
    compile 'com.android.support:support-v4:24.0.0-beta1'

    // Glide image library
    compile 'com.github.bumptech.glide:glide:3.7.0'
}
```

3. Open **strings.xml**, **dimens.xml**, **colors.xml** located under **res ⇒ values** and add the below values.

strings.xml

```
<resources>
    <string name="app_name">Navigation Drawer</string>

    <string name="navigation_drawer_open">Open navigation drawer</string>
    <string name="navigation_drawer_close">Close navigation drawer</string>
    <string name="openDrawer">open_drawer</string>
    <string name="closeDrawer">close_drawer</string>

    <!-- Navigation Drawer menu labels -->
    <string name="nav_home">Home</string>
    <string name="nav_photos">Photos</string>
    <string name="nav_movies">Movies</string>
    <string name="nav_notifications">Notifications</string>
    <string name="nav_settings">Settings</string>
    <string name="nav_about_us">About Us</string>

    <!-- Toolbar titles when navigation menu item is selected -->
    <string-array name="nav_item_activity_titles">
        <item>Home</item>
        <item>Photos</item>
        <item>Movies</item>
        <item>Notifications</item>
        <item>Settings</item>
    </string-array>

    <string name="activity_title_about_us">About Us</string>
    <string name="activity_title_privacy_policy">Privacy Policy</string>
    <string name="action_clear_all">Clear All</string>
```

```
    <string name="action_logout">Logout</string>
    <string name="action_mark_all_read">Mark All as Read</string>
    <string name="privacy_policy">Privacy Policy</string>
</resources>
```

dimens.xml

```
<resources>
    <!-- Default screen margins, per the Android Design guidelines. -->
    <dimen name="nav_header_vertical_spacing">16dp</dimen>
    <dimen name="nav_header_height">160dp</dimen>

    <!-- Default screen margins, per the Android Design guidelines. -->
    <dimen name="activity_horizontal_margin">16dp</dimen>
    <dimen name="activity_vertical_margin">16dp</dimen>
    <dimen name="fab_margin">16dp</dimen>
    <dimen name="profile_width">75dp</dimen>
    <dimen name="profile_height">75dp</dimen>
</resources>
```

colors.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#d32f2f</color>
    <color name="colorPrimaryDark">#c72c2c</color>
    <color name="colorAccent">#FFFFFF</color>
</resources>
```

4. Download Home, Photos, Movies, Notifications, Settings icons from Material Icons and add them to your project's **res** folder. These icons will be used as navigation menu item icons.

5. Under **res ⇒ menu** directory, create two menu xml files named **notifications.xml** and **main.xml**. These menus are used to render different toolbar overflow menus when the user switches between navigation drawer items.

notifications.xml

```
<?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_mark_all_read"
        android:orderInCategory="100"
        android:title="@string/action_mark_all_read"
        app:showAsAction="never" />

    <item
        android:id="@+id/action_clear_notifications"
        android:orderInCategory="101"
        android:title="@string/action_clear_all"
        app:showAsAction="never" />
</menu>
```

main.xml

```
<?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_logout"
        android:orderInCategory="100"
        android:title="@string/action_logout"
        app:showAsAction="never" />
</menu>
```

Creating Fragments & Activities

6. Create three packages named **activity**, **fragment**, **other** and move the **MainActivity.java** to **activity** package.

7. Create all the fragment classes needed for navigation menu. Overall I am creating five fragment classes. **Right click** on **fragment** package, **New ⇒ Fragment ⇒ Fragment (Blank)** and name it as **HomeFragment.java**. This fragment will be loaded always whenever user open the app.

Likewise create other fragment activities named **PhotosFragment**, **MoviesFragment**, **NotificationsFragment** and **SettingsFragment**.

8. In our navigation drawer menu, there are two other menu items, **About Us** and **Privacy Policy**. For these two we are gonna create activities instead of fragments. Create new two **activities** named **AboutUsActivity.java** and **PrivacyPolicyActivity.java**.

9. Inside **other** package, create a class named **CircleTransform.java** This class is used to display the profile image in circular fashion.

CircleTransform.java

```
package com.coderbd.navigationdrawer.other;

import android.content.Context;
import android.graphics.Bitmap;
import android.graphics.BitmapShader;
import android.graphics.Canvas;
import android.graphics.Paint;

import com.bumptech.glide.load.engine.bitmap_recycle.BitmapPool;
import com.bumptech.glide.load.resource.bitmap.BitmapTransformation;

/**
 * Created by Lincoln on 10/03/16.
 */
public class CircleTransform extends BitmapTransformation {
    public CircleTransform(Context context) {
        super(context);
    }

    @Override
    protected Bitmap transform(BitmapPool pool, Bitmap toTransform, int
```

```
outWidth, int outHeight) {
        return circleCrop(pool, toTransform);
    }

    private static Bitmap circleCrop(BitmapPool pool, Bitmap source) {
        if (source == null) return null;

        int size = Math.min(source.getWidth(), source.getHeight());
        int x = (source.getWidth() - size) / 2;
        int y = (source.getHeight() - size) / 2;

        // TODO this could be acquired from the pool too
        Bitmap squared = Bitmap.createBitmap(source, x, y, size, size);

        Bitmap result = pool.get(size, size, Bitmap.Config.ARGB_8888);
        if (result == null) {
            result = Bitmap.createBitmap(size, size,
Bitmap.Config.ARGB_8888);
        }

        Canvas canvas = new Canvas(result);
        Paint paint = new Paint();
        paint.setShader(new BitmapShader(squared, BitmapShader.TileMode.CLAMP,
BitmapShader.TileMode.CLAMP));
        paint.setAntiAlias(true);
        float r = size / 2f;
        canvas.drawCircle(r, r, r, paint);
        return result;
    }

    @Override
    public String getId() {
        return getClass().getName();
    }
}
```

10. Under **res ⇒ drawable**, create a file named **bg_circle.xml**. This provides circular background to view.

bg_circle.xml

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

    <solid
        android:color="@color/colorPrimary"/>

    <size
        android:width="120dp"
        android:height="120dp"/>
</shape>
```

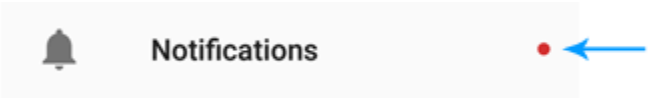
11. Under **res layout** folder, create a layout named **menu_dot.xml**. This layout is used to render dot next to notifications label.

menu_dot.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center_vertical"
    android:orientation="vertical">

    <LinearLayout
        android:layout_width="6dp"
        android:layout_height="6dp"
        android:background="@drawable/bg_circle"></LinearLayout>

</LinearLayout>
```



Adding Navigation Drawer

12. Under **res ⇒ menu** directory, create a menu xml file named **activity_main_drawer.xml**. This menu renders the Navigation Drawer list items. Here we set the **icons** and **labels**. You can also notice here **<group>** is used to combine multiple items under single levels. An **<item>** also can be used group multiple child items with a title. This provides a horizontal separator between the two sets.

activity_main_drawer.xml

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

    <group android:checkableBehavior="single">
        <item
            android:id="@+id/nav_home"
            android:icon="@drawable/ic_home_black_24dp"
            android:title="@string/nav_home" />
        <item
            android:id="@+id/nav_photos"
            android:icon="@drawable/ic_photo_library_black_24dp"
            android:title="@string/nav_photos" />
        <item
            android:id="@+id/nav_movies"
            android:icon="@drawable/ic_local_movies_black_24dp"
            android:title="@string/nav_movies" />

        <item
            android:id="@+id/nav_notifications"
            android:icon="@drawable/ic_notifications_black_24dp"
            android:title="@string/nav_notifications" />
```

```
        <item
            android:id="@+id/nav_settings"
            android:icon="@drawable/ic_settings_black_24dp"
            android:title="@string/nav_settings" />
    </group>

    <item android:title="Other">
        <menu>
            <item
                android:id="@+id/nav_about_us"
                android:title="@string/nav_about_us" />
            <item
                android:id="@+id/nav_privacy_policy"
                android:title="@string/privacy_policy" />
        </menu>
    </item>

</menu>
```

13. Under **res ⇒ layout**, create a file named **nav_header_main.xml** This layout renders the navigation drawer header part with a profile image, name and website.

nav_header_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/view_container"
    android:layout_width="match_parent"
    android:layout_height="@dimen/nav_header_height"
    android:gravity="bottom"
    android:orientation="vertical"
    android:theme="@style/ThemeOverlay.AppCompat.Dark">

    <ImageView
        android:id="@+id/img_header_bg"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:scaleType="fitXY"
        android:src="@mipmap/ic_launcher" />

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerVertical="true"
        android:orientation="vertical"
        android:padding="@dimen/activity_horizontal_margin">

        <ImageView
            android:id="@+id/img_profile"
            android:layout_width="@dimen/profile_width"
            android:layout_height="@dimen/profile_height"
            android:paddingTop="@dimen/nav_header_vertical_spacing"
            app:srcCompat="@android:drawable/sym_def_app_icon" />
```

```
        <TextView
            android:id="@+id/name"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:paddingTop="@dimen/nav_header_vertical_spacing"
            android:textAppearance="@style/TextAppearance.AppCompat.Body1" />

        <TextView
            android:id="@+id/website"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />

    </LinearLayout>
</RelativeLayout>
```

14. Now we'll create another layout file to add required **Toolbar**, **FAB** and **FrameLayout**. FrameLayout is used to load appropriate fragment when an item is selected from nav menu. Create a layout file named **app_bar_main.xml** under **res ⇒ layout**.

app_bar_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true"
    tools:context="com.coderbd.navigationdrawer.activity.MainActivity">

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

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

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

    <FrameLayout
        android:id="@+id/frame"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:layout_behavior="@string/appbar_scrolling_view_behavior"></FrameL
ayout>

    <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="@dimen/fab_margin"
        app:backgroundTint="@color/colorPrimary"
        app:srcCompat="@android:drawable/ic_dialog_email" />

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

15. Open the layout file your main activity **activity_main.xml** and add **NavigationView** element. You can notice that the toolbar layout is added using <**include**> tag.

activity_main.xml

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

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

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

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

16. Now we have all the required elements in place. It's time to open the main activity and do the necessary changes to make the navigation drawer functional. Open **MainActivity.java** and modify as explained below.

setUpNavigationView() – Initializes the Navigation Drawer by creating necessary click listeners and other functions.

loadNavHeader() – Function loads the navigation drawer header information like profile image, name and website. Here we use Glide image library to load the profile image.

getHomeFragment() – Returns the appropriate Fragment depending on the nav menu item user selected. For example if user selects **Photos** from nav menu, it returns **PhotosFragment**. This can be done by using the variable *navItemIndex*.

loadHomeFragment() – Loads the fragment returned from **getHomeFragment()** function into **FrameLayout**. It also takes care of other things like changing the toolbar title, hiding / showing fab, invalidating the options menu so that new menu can be loaded for different fragment.

Other code is very easy to understand and self explanatory. I have also added comments to each code block for better understanding.

MainActivity.java

```
package com.coderbd.navigationdrawer.activity;

import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.NavigationView;
import android.support.design.widget.Snackbar;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentTransaction;
import android.support.v4.view.GravityCompat;
import android.support.v4.widget.DrawerLayout;
import android.support.v7.app.ActionBarDrawerToggle;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import com.bumptech.glide.Glide;
import com.bumptech.glide.load.engine.DiskCacheStrategy;

import com.coderbd.navigationdrawer.R;
import com.coderbd.navigationdrawer.fragment.HomeFragment;
import com.coderbd.navigationdrawer.fragment.MoviesFragment;
import com.coderbd.navigationdrawer.fragment.NotificationsFragment;
import com.coderbd.navigationdrawer.fragment.PhotosFragment;
import com.coderbd.navigationdrawer.fragment.SettingsFragment;
import com.coderbd.navigationdrawer.other.CircleTransform;

public class MainActivity extends AppCompatActivity {

    private NavigationView navigationView;
    private DrawerLayout drawer;
    private View navHeader;
    private ImageView imgNavHeaderBg, imgProfile;
    private TextView txtName, txtWebsite;
    private Toolbar toolbar;
```

```
    private FloatingActionButton fab;

    // urls to load navigation header background image
    // and profile image
    private static final String urlNavHeaderBg =
"https://static1.squarespace.com/static/55c263c0e4b0ad74ccf2c9a9/t/55fadbbde4b01c49490d7b18/1442503622299/Backgrounds+for+Site_UX+BG+Small+Icons.png?format=2500w";
    private static final String urlProfileImg = "https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcScuIluC7jOsZE5DRJi7x3SRPmLJxSKw7GOazGouLqxwCDtRrASmQ";

    // index to identify current nav menu item
    public static int navItemIndex = 0;

    // tags used to attach the fragments
    private static final String TAG_HOME = "home";
    private static final String TAG_PHOTOS = "photos";
    private static final String TAG_MOVIES = "movies";
    private static final String TAG_NOTIFICATIONS = "notifications";
    private static final String TAG_SETTINGS = "settings";
    public static String CURRENT_TAG = TAG_HOME;

    // toolbar titles respected to selected nav menu item
    private String[] activityTitles;

    // flag to load home fragment when user presses back key
    private boolean shouldLoadHomeFragOnBackPress = true;
    private Handler mHandler;


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

        mHandler = new Handler();

        drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
        navigationView = (NavigationView) findViewById(R.id.nav_view);
        fab = (FloatingActionButton) findViewById(R.id.fab);

        // Navigation view header
        navHeader = navigationView.getHeaderView(0);
        txtName = (TextView) navHeader.findViewById(R.id.name);
        txtWebsite = (TextView) navHeader.findViewById(R.id.website);
        imgNavHeaderBg = (ImageView)
navHeader.findViewById(R.id.img_header_bg);
        imgProfile = (ImageView) navHeader.findViewById(R.id.img_profile);
```

```
        // load toolbar titles from string resources
        activityTitles =
getResources().getStringArray(R.array.nav_item_activity_titles);

        fab.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Snackbar.make(view, "Replace with your own action",
Snackbar.LENGTH_LONG)
                        .setAction("Action", null).show();
            }
        });

        // load nav menu header data
        loadNavHeader();

        // initializing navigation menu
        setUpNavigationView();

        if (savedInstanceState == null) {
            navItemIndex = 0;
            CURRENT_TAG = TAG_HOME;
            loadHomeFragment();
        }
    }

    /***
     * Load navigation menu header information
     * like background image, profile image
     * name, website, notifications action view (dot)
     */
    private void loadNavHeader() {
        // name, website
        txtName.setText("Rakibul Hasan");
        txtWebsite.setText("www.coderbd.com");

        // loading header background image
        Glide.with(this).load(urlNavHeaderBg)
                .crossFade()
                .diskCacheStrategy(DiskCacheStrategy.ALL)
                .into(imgNavHeaderBg);

        // Loading profile image
        Glide.with(this).load(urlProfileImg)
                .crossFade()
                .thumbnail(0.5f)
                .bitmapTransform(new CircleTransform(this))
                .diskCacheStrategy(DiskCacheStrategy.ALL)
                .into(imgProfile);

        // showing dot next to notifications label
        navigationView.getMenu().getItem(3).setActionView(R.layout.menu_dot);
    }

    /***
```

```
     * Returns respected fragment that user
     * selected from navigation menu
     */
    private void loadHomeFragment() {
        // selecting appropriate nav menu item
        selectNavMenu();

        // set toolbar title
        setToolbarTitle();

        // if user select the current navigation menu again, don't do
anything
        // just close the navigation drawer
        if (getSupportFragmentManager().findFragmentByTag(CURRENT_TAG) !=
null) {
            drawer.closeDrawers();

            // show or hide the fab button
            toggleFab();
            return;
        }

        // Sometimes, when fragment has huge data, screen seems hanging
        // when switching between navigation menus
        // So using runnable, the fragment is loaded with cross fade effect
        // This effect can be seen in GMail app
        Runnable mPendingRunnable = new Runnable() {
            @Override
            public void run() {
                // update the main content by replacing fragments
                Fragment fragment = getHomeFragment();
                FragmentTransaction fragmentTransaction =
getSupportFragmentManager().beginTransaction();
                fragmentTransaction.setCustomAnimations(android.R.anim.fade_i
n,
                        android.R.anim.fade_out);
                fragmentTransaction.replace(R.id.frame, fragment,
CURRENT_TAG);
                fragmentTransaction.commitAllowingStateLoss();
            }
        };

        // If mPendingRunnable is not null, then add to the message queue
        if (mPendingRunnable != null) {
            mHandler.post(mPendingRunnable);
        }

        // show or hide the fab button
        toggleFab();

        //Closing drawer on item click
        drawer.closeDrawers();

        // refresh toolbar menu
        invalidateOptionsMenu();
```

```
    }

    private Fragment getHomeFragment() {
        switch (navItemIndex) {
            case 0:
                // home
                HomeFragment homeFragment = new HomeFragment();
                return homeFragment;
            case 1:
                // photos
                PhotosFragment photosFragment = new PhotosFragment();
                return photosFragment;
            case 2:
                // movies fragment
                MoviesFragment moviesFragment = new MoviesFragment();
                return moviesFragment;
            case 3:
                // notifications fragment
                NotificationsFragment notificationsFragment = new
NotificationsFragment();
                return notificationsFragment;

            case 4:
                // settings fragment
                SettingsFragment settingsFragment = new SettingsFragment();
                return settingsFragment;
            default:
                return new HomeFragment();
        }
    }

    private void setToolbarTitle() {
        getSupportActionBar().setTitle(activityTitles[navItemIndex]);
    }

    private void selectNavMenu() {
        navigationView.getMenu().getItem(navItemIndex).setChecked(true);
    }

    private void setUpNavigationView() {
        //Setting Navigation View Item Selected Listener to handle the item
click of the navigation menu
        navigationView.setNavigationItemSelectedListener(new
NavigationView.OnNavigationItemSelectedListener() {

            // This method will trigger on item Click of navigation menu
            @Override
            public boolean onNavigationItemSelected(MenuItem menuItem) {

                //Check to see which item was being clicked and perform
appropriate action
                switch (menuItem.getItemId()) {
                    //Replacing the main content with ContentFragment Which
is our Inbox View;
                    case R.id.nav_home:
```

```
                        navItemIndex = 0;
                        CURRENT_TAG = TAG_HOME;
                        break;
                    case R.id.nav_photos:
                        navItemIndex = 1;
                        CURRENT_TAG = TAG_PHOTOS;
                        break;
                    case R.id.nav_movies:
                        navItemIndex = 2;
                        CURRENT_TAG = TAG_MOVIES;
                        break;
                    case R.id.nav_notifications:
                        navItemIndex = 3;
                        CURRENT_TAG = TAG_NOTIFICATIONS;
                        break;
                    case R.id.nav_settings:
                        navItemIndex = 4;
                        CURRENT_TAG = TAG_SETTINGS;
                        break;
                    case R.id.nav_about_us:
                        // launch new intent instead of loading fragment
                        startActivity(new Intent(MainActivity.this,
AboutUsActivity.class));
                        drawer.closeDrawers();
                        return true;
                    case R.id.nav_privacy_policy:
                        // launch new intent instead of loading fragment
                        startActivity(new Intent(MainActivity.this,
PrivacyPolicyActivity.class));
                        drawer.closeDrawers();
                        return true;
                    default:
                        navItemIndex = 0;
                }

                //Checking if the item is in checked state or not, if not
make it in checked state
                if (menuItem.isChecked()) {
                    menuItem.setChecked(false);
                } else {
                    menuItem.setChecked(true);
                }
                menuItem.setChecked(true);

                loadHomeFragment();

                return true;
            }
        });


        ActionBarDrawerToggle actionBarDrawerToggle = new
ActionBarDrawerToggle(this, drawer, toolbar, R.string.openDrawer,
R.string.closeDrawer) {
```

```
            @Override
            public void onDrawerClosed(View drawerView) {
                // Code here will be triggered once the drawer closes as we
dont want anything to happen so we leave this blank
                super.onDrawerClosed(drawerView);
            }

            @Override
            public void onDrawerOpened(View drawerView) {
                // Code here will be triggered once the drawer open as we
dont want anything to happen so we leave this blank
                super.onDrawerOpened(drawerView);
            }
        };

        //Setting the actionbarToggle to drawer layout
        drawer.setDrawerListener(actionBarDrawerToggle);

        //calling sync state is necessary or else your hamburger icon wont
show up
        actionBarDrawerToggle.syncState();
    }

    @Override
    public void onBackPressed() {
        if (drawer.isDrawerOpen(GravityCompat.START)) {
            drawer.closeDrawers();
            return;
        }

        // This code loads home fragment when back key is pressed
        // when user is in other fragment than home
        if (shouldLoadHomeFragOnBackPress) {
            // checking if user is on other navigation menu
            // rather than home
            if (navItemIndex != 0) {
                navItemIndex = 0;
                CURRENT_TAG = TAG_HOME;
                loadHomeFragment();
                return;
            }
        }

        super.onBackPressed();
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is
present.

        // show menu only when home fragment is selected
        if (navItemIndex == 0) {
            getMenuInflater().inflate(R.menu.main, menu);
        }
```

```
        // when fragment is notifications, load the menu created for
notifications
        if (navItemIndex == 3) {
            getMenuInflater().inflate(R.menu.notifications, menu);
        }
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();

        //noinspection SimplifiableIfStatement
        if (id == R.id.action_logout) {
            Toast.makeText(getApplicationContext(), "Logout user!",
Toast.LENGTH_LONG).show();
            return true;
        }

        // user is in notifications fragment
        // and selected 'Mark all as Read'
        if (id == R.id.action_mark_all_read) {
            Toast.makeText(getApplicationContext(), "All notifications marked
as read!", Toast.LENGTH_LONG).show();
        }

        // user is in notifications fragment
        // and selected 'Clear All'
        if (id == R.id.action_clear_notifications) {
            Toast.makeText(getApplicationContext(), "Clear all
notifications!", Toast.LENGTH_LONG).show();
        }

        return super.onOptionsItemSelected(item);
    }

    // show or hide the fab
    private void toggleFab() {
        if (navItemIndex == 0)
            fab.show();
        else
            fab.hide();
    }
}
```

The End