

Requerid Components

This is the list of points in blue, and my fix are explains in **RED COLOR**.

Standard Design Does Not Meet Specifications

[~] App does not redefine or misuse Android UI patterns, such that icons or behaviors could be

misleading or confusing to users.

○ I noticed in SlidingMenu.java, the code looks like it implements the ExpandableListView to build out something similar to a Navigation drawer shown in this Android Developer Doc. Is there a particular reason you chose to implement ExpandableListView.OnChildClickListener rather than use the Navigation Drawer? The app's current structure looks like it would work well with a Navigation Drawer (and it might be easier for you to build and maintain).

○ Also, I think because the code currently uses ExpandableListView, slide_menu_button_home appears in all the actionbars even when there is no additional Navigation Drawer. For example, when I've navigated to the Images Activity, the slide menu button is there even though there are no additional Activities to navigate to. The Android Developer doc linked above gives a good explanation of when a user should expect to see the Navigation Drawer icon.

SOLUTION: SlidingMenu is an Open Source Android library that allows developers to easily create applications with sliding menus like those made popular in the Google+, YouTube, and Facebook apps.

In my app I used to create the Slide Menu in both side (left and right) .

With the NavigationDrawer from here -->

<https://developer.android.com/training/implementing-navigation/nav-drawer.html> I cannot create the slide menu in the rigth side for the user.

I need that 2nd menu for the User Experience, that I made in my mockups papers.

I meet the NavigationDrawer the last year, when I start to learn Android Development and news libraries. But here, I need a different think, I need the second menu.

That the reason for the SlideMenu library.

Standard Design Does Not Meet Specifications

Also, I think because the code currently uses ExpandableListView, slide_menu_button_home appears in all the actionbars even when there is no additional Navigation Drawer. For example, when I've navigated to the Images Activity, the slide menu button is there even though there are no additional Activities to navigate to. The Android Developer doc linked above gives a good explanation of when a user should

expect to see the Navigation Drawer icon.

SOLUTION: Using this two lines into the onCreate method for each Activity:

```
getSupportActionBar().setDisplayHomeAsUpEnabled(true);  
getSupportActionBar().setIcon(android.R.color.transparent);
```

I removed the Navigation Drawer Icon and only put the “back arrow” , so the use knows that with one tap in that icon, is back to the previous screen. I have used this non-traditional method, because we are using the library "SlideMenu" to take ANY TIME 2 menu's side, both the right and the left, so that the user can quickly access the results of matches.

Navigation Does Not Meet Specifications

- [-] App supports standard system Back button navigation and does not make use of any custom, on-screen “Back button” prompts.
 - When I click the Back button from the Home Page, I get a toast that says “Por favor, haga clic en atras para salir”. Was this toast put in for testing purposes? Users may find this extra tapping excessive and frustrating, so I would suggest removing it.

SOLUTION: I comments the lines 66 to 84 into Home.java , so I removed the feature “doble tap on back to exit” , that I implements with this piece of code:

```
@Override  
public void onBackPressed() {  
    if (doubleBackToExitPressedOnce) {  
        super.onBackPressed();  
        return;  
    }  
  
    this.doubleBackToExitPressedOnce = true;  
    Toast.makeText(this, "Por favor, haga clic en Atrás para salir",  
        Toast.LENGTH_SHORT).show();  
  
    new Handler().postDelayed(new Runnable() {  
        @Override  
        public void run() {  
            doubleBackToExitPressedOnce = false;  
        }  
    }, 2000);
```

}

Optional Components

Custom Views Does Not Meet Specifications

- App creates and uses a custom View.
- App uses a novel View that couldn't sufficiently be satisfied by the core Views in Android.
- We cannot count the custom text towards this criteria because there are methods of modifying text that are already built in and rather straightforward to implement. For example, Roboto is the default font of Android and if you want to set it to bold, can use textStyle in the layout.xml as described in this [StackOverflow discussion](#).

SOLUTION: You have the right response. So looking into my app and the feature that we need to for the user I have create a new ImageView, I worked over TouchImageView that extends ImageView and supports all of ImageView's functionality.

I added into the project this three files:

- ar.com.sccradiomobile.utils.TouchImageView.java implements all of ImageView's functionality
- ar.com.sccradiomobile.utils.DetaillImageFullScreenActivity.java Activity that help to load the imagen in a new screen.
- res\layout\activity_imagenes.xml the layout that contains the custom view for the image.

In addition, TouchImageView adds pinch zoom, dragging, fling, double tap zoom functionality and other animation polish. The intention is for TouchImageView to mirror as closely as possible the functionality of zoomable images in Gallery apps.

This functionality it's based on this repo <https://github.com/MikeOrtiz/TouchImageView> I read how it's works and I do in our final project.

TouchImageView does not yet support pinch image rotation. Also, FIT_START and FIT_END scaleTypes are not yet supported.

Suggested Modifications

- 1) into DynamicPhotoListActivity.java I comments the lines
from 127 to 129 , to remove the toast for testing porpouse.
- 2) Master-Detail flow for the the news on the HomePage:

I don't have time for the moment.