

# Using Android Fragments in A Campus Guide System

Boheng Wei, Xi Yan, Mao Zheng  
Department of Computer Science  
University of Wisconsin-La Crosse  
La Crosse WI, 54601  
[mzheng@uwlax.edu](mailto:mzheng@uwlax.edu)

## Abstract

Smartphones, tablets and wireless data plans are already a trillion dollar a year business. However, this is just the beginning. More and more students are interested in Mobile Computing. The Android application, a campus guide system, is a practice in mobile development.

There are two major platforms in the mobile device community: iOS and Android. This project chose Android development, mainly for the reason of its openness. In addition, all the tools in the Android development are free and no special hardware is required.

Application components are the essential building blocks of an Android application. There are four different types of application components. Each type serves a distinct purpose and has its own lifecycle that defines how the component is created and destroyed. One of the application components is Activity. An *activity* represents a single screen with a user interface. A multi-screen application will consist of a number of activities that work together to form a cohesive user experience. This project uses fragments in the activity to support a dynamic and flexible user interface.

Android introduced fragments in Android 3.0 (API level 11). A Fragment represents a behavior or a portion of a user interface in an Activity. We combine multiple fragments in a single activity to build a multi-pane user interface. The fragment information can be reused in the activity. For example: when a campus map is displayed in the designated panel, the user can move the building of interest to the center of the map. The user can then click to view the detailed information about the building. When the user returns to the previous screen, the selected building still remains in the center of the map. The fragment is able to “remember” previous screen information. Without the use of fragments, the activity has to call for instructions to store and retain the previous actions, to recreate the view in order to have the same screen layout.

The application gives directions, allows a user to view and search campus buildings, and creates a customized tour route from the user’s current location to the selected buildings.

# 1 Introduction

Smartphones, tablets and wireless data plans are already a trillion dollar a year business. However this is just the beginning. More and more students are interested in Mobile Computing. The Android application, a campus guide system, is a practice in mobile development.

There are two major platforms in the mobile device community: iOS and Android. This project chose Android development, mainly for the reason of its openness. In addition, all the tools in the Android development are free and no special hardware is required.

Application components are the essential building blocks of an Android application. There are four different types of application components. Each type serves a distinct purpose and has its own lifecycle that defines how the component is created and destroyed. One of the application components is Activity. An *activity* represents a single screen with a user interface. A multi-screen application will consist of a number of activities that work together to form a cohesive user experience. This project uses fragments in the activity to support a dynamic and flexible user interface. It also applies the navigation-drawer pattern and sliding pane layout from the responsive pattern in different places to handle multi-panes interactions.

The application gives directions, allows a user to view and search campus buildings, and creates a customized tour route from the user's current location to the selected buildings.

## 2 A Campus Guide Application

This mobile application intends to provide information about a university campus to a tourist using his/her current location. When the user opens the app in his/her mobile device, there will be a map showing the overall view of the campus. The user can center the map over an area of interest, zoom in on that area to get detailed information, or select one or more buildings in the campus to visit. The application will be able to navigate the user to a point of interest. In addition, detailed information about a building, such as the name, the services provided, the residing departments, and current events hosted in the building, can be obtained by the user.

Figure 1 is a screen shot of showing a map of the entire University of Wisconsin-LaCrosse campus. The blue arrow is the user's current location. The red circles are the points of the interest.

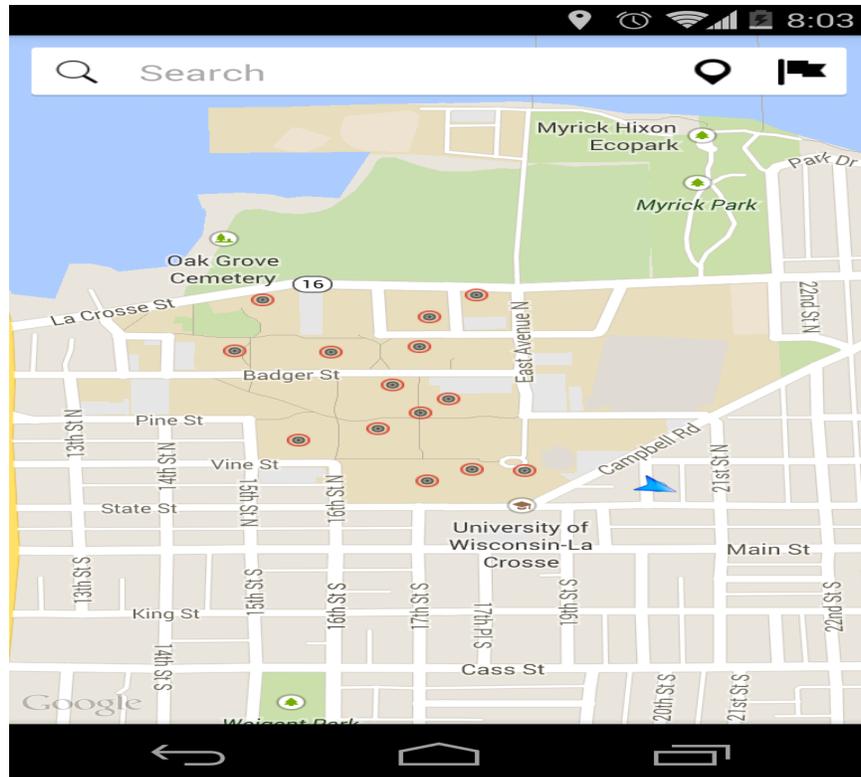


Figure 1 A Campus Map of the University of Wisconsin-LaCrosse

### 3 User Interface Design

This project uses new techniques developed in Google I/O for the interface design to improve the user experience. These techniques include Fragment, Navigation Drawer and SlidingPaneLayout.

#### 3.1 Android Fragment

Android introduced fragments in Android 3.0 (API level 11). A Fragment represents a behavior or a portion of a user interface in an Activity. We combine multiple fragments in a single activity to build a multi-pane user interface. The fragment information can be reused in the activity. For example: when a campus map is displayed in the designated panel, the user can move the building of interest to the center of the map. The user can then click to view the detailed information about the building. When the user returns to the previous screen, the selected building still remains in the center of the map. The fragment is able to “remember” previous screen information. Without the use of fragments, the activity has to call for instructions to store and retain the previous actions, to recreate the view in order to have same screen layout.

The following source code segment is where the project sets the MapFragment in the MainActivity.java.

```
mapFragment = (SupportMapFragment) getSupportFragmentManager()
    .findFragmentByTag(MAP_FRAGMENT_TAG);

// only create a fragment if it doesn't already exist.
if (mapFragment == null) {
    // SupportMapFragment programmatically adds the map
    mapFragment = SupportMapFragment.newInstance();

    FragmentTransaction fragmentTransaction =
        getSupportFragmentManager().beginTransaction();
    fragmentTransaction.add(R.id.main_content, mapFragment,
        MAP_FRAGMENT_TAG);
    fragmentTransaction.commit();
}

if (savedInstanceState == null) {
    // First incarnation of this activity.
    mapFragment.setRetainInstance(true);
    JSONLoader jl = new JSONLoader(this);
    jl.execute("/all");
} else {
    // Reincarnated activity. The obtained map is the same map instance
    // in the previous activity life cycle. There is no need to reinitialize it.
    mMap = mapFragment.getMap();
}

setUpMapIfNeeded();
```

### 3.2 Navigation Drawer

The navigation drawer is a panel that transitions in from the left edge of the screen and displays the app's main navigation options. It is hidden most of the time, but is revealed when the user swipes a finger from the left edge of the screen. A navigation drawer contains of two panes: the master pane and the detail pane. The master pane is the place that you put your navigation items on it, such that user is able to access different screens by just swiping out the drawer and clicking on the item. The detail pane is where you put the contents of different screens. After you click on the item, the navigation drawer collapses and the detail pane will switch to the destination screen. Because they are transient, navigation drawers make views less cluttered. Navigation drawer is the way to access any top level content from anywhere in the app, no matter how deep a level the user has navigated to.

In this project, the navigation drawer is used for switching options between two main modes in the map view. These modes are Normal map view mode and Satellite map view mode. The user is able to choose between these two modes to his/her preferred way of getting to know the campus. Figure 1 is a map of the University of

Wisconsin-LaCrosse in normal map view mode. Figure 2 is the same map in satellite view mode.

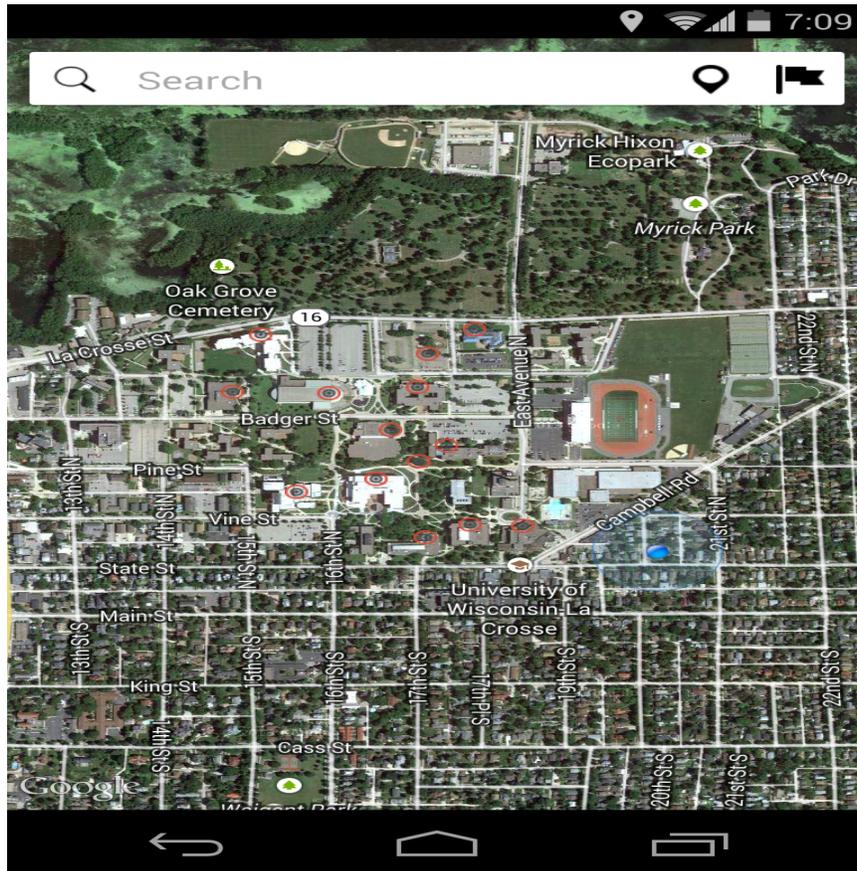


Figure 2 A Map of the University of Wisconsin-LaCrosse in Satellite View Mode

Below are the code segments that initialize the DrawerLayout and related option list.

```
mDrawerLayout = (DrawerLayout) findViewById(R.id.drawer_layout);  
  
mDrawerList = (ListView) findViewById(R.id.left_drawer);  
mDrawerList.setAdapter(new ArrayAdapter<String>(this,  
R.layout.drawer_list_item, titles));  
mDrawerList.setOnItemClickListener(new DrawerItemClickListener());
```

### 3.3 SlidingPaneLayout

The Campus Guide application is intended to run on various mobile devices with different screen sizes. We are certainly interested in creating an application that is scalable to a wide range of Android devices. A powerful tool that can help do this is SlidingPaneLayout. SlidingPaneLayout provides a horizontal, multi-pane layout for use at the top level of a user interface. A left (or first) pane is treated as a content list

or browser, subordinate to a primary detail view for displaying content. A SlidingPaneLayout consists of two panes: the master pane and the detail pane. Selecting an item in the master pane will update the detail pane. The core of SlidingPaneLayout is that the rightmost pane, the detail pane, can be dragged horizontally to uncover the master pane below. On a screen that is large enough to display both panes, the user will see both simultaneously. On a small screen, the user will see a single pane, but the user can slide the detail pane to reveal the master pane.

In this project, we are not using the traditional SlidingPaneLayout. Instead we are using a modified version called sliding up pane. Using this sliding up pane, the detail pane can be dragged vertically, but not horizontally.

When the user selects one of the red circles in Figure 1, information about the building of interest can be found in the detail pane. The detail pane is shown in Figure 3. Once the user slides up the detail pane, the initial map displayed in the master pane will be covered by the detail pane. It can also be uncovered by sliding down the detail pane.

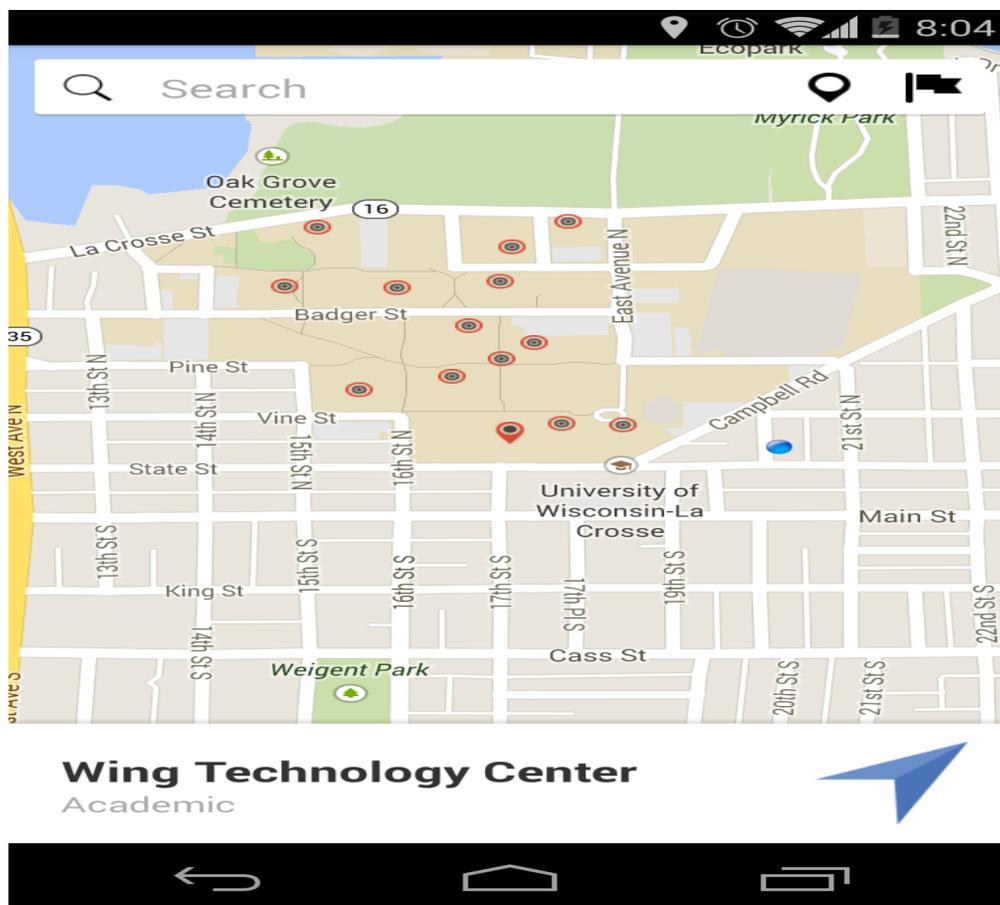


Figure 3 Sliding Up Pane

The SlidingPaneLayout is a tool for handy layout, especially when creating applications that work well on any size device. However, it's certainly not a replacement for the Navigation Drawer paradigm. SlidingPaneLayout is distinct from a navigation drawer, as described in the design guide and should not be used in the same scenarios. SlidingPaneLayout should be thought of only as a way to allow a two-pane layout, normally used on larger screens, to adapt to smaller screens. The interaction pattern expressed by SlidingPaneLayout implies a physicality and a direct information hierarchy between panes, that does not necessarily exist in a scenario, where a navigation drawer should be used instead.

## 4. Conclusions

This project uses a Client-Server architecture. The server is developed in Python with Django framework. A SQLite database is included in the server side.

The current application can be enhanced in two areas. One is augmented reality. This augments the user's view of the real world, with additional virtual information. For example, current event information can be overlaid on a picture of a building. The second area is the ability of the user to leave comments on the interactive map.

Such a guide system can also be developed for museum visitors and exhibition tourists. In this type of systems, the user's location and orientation are used as active environment inputs. Different content is delivered to the mobile device whenever the user relocates.

## References

- [1] Computer Science 2008, An Interim Revision of CS 2001(<http://www.acm.org/education/curricula/ComputerScience2008.pdf>)
- [2] Android Developer's Guide. <http://developer.android.com/guide/index.html>
- [3] Abelson, W.F., Collins, C., Sen, R. *Unlocking Android – A Developer's Guide*. Manning Pub. April 2009.
- [4] Victor Matos, Rebecca Grasser, Building Applications for the Android OS Mobile Platform: A Primer and Course Materials, *Journal of Computing Sciences in Colleges*, Volume 26 Issue 1, pp: 23-29, October 2010
- [5] Derek Riley, Using Mobile Phone Programming to teach Java and Advanced Programming to Computer Scientist, ACM Special Interest Group on Computer Science Education SIGSCE 2012, pp:541-546. Feb. 29-March 3, 2012.

[6] B. N. Schilit, N. Adams, and R. Want. Context-aware Computing Applications. In Proceedings of IEEE Workshop on Mobile Computing Systems and Applications, pp. 85-90, Santa Cruz, CA, Dec. 1994. IEEE Computer Society Press.

[7] A. K. Dey and G. D. Abowd, "Towards a Better Understanding of Context and Context-awareness," in CHI 2000 Workshop on the What, Who, Where, When and How of Context-awareness, 2000.

[8] R. Lowe, P. Mandl, M. Weber. "Context Director: A Context-aware Service for Mobile Context-aware Computing Applications by the Example of Google Android", Tenth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops 2012), Lugano, Switzerland, March 2012.