Lecture Notes: Android UI Basics

1. Model View Separation Principle
    a. Don't mix UI code with model code and vice versa
2. Activity UIs are defind as a tree of widgets
    a. Leaf nodes in the tree are simple widgets like buttons and extend the view class: TextView, EditText, Button, Switch, ImageView, Many others
    b. Internal nodes extend the ViewGroup class: LinearLayout, RelativeLayout, ConstraintLayout, TableLayout, FrameLayout, ScrollView, others
3. Can build the tree of widgets by hand (in code) but the reasonable way to do it is to specify it in XML and have the activity "inflate" it's view
4. Show GeoGuiz as an example of a combination of linear layouts
5. UI draws and waits for events to respond to
    a. Similar to a server waiting for requests (event driven programming)
    b. You write code to respond to events in event handlers (usually in anonymous inner classes)
6. TableLayoutExample
    a. Can add a ScrollView to handle landscape
    b. Can also make a landscape specific version by creating a new resource in the layout folder
        i. Create the layout file with the same name as the portrait (original) (right-click in the 'layout' folder and select 'New -> Layout Resource File'
        ii. Specify 'orientation' as the qualifier and specify 'landscape'
        iii. Copy the portrait layout into the new landscape one as a starting point and make whatever changes you want for the landscape layout
7. Three xml namespaces
    a. android: Use what's on the device the app will run on
    b. app: Use support library components (for backward compatibility)

    c. tools: For attributes you want to appear in the design view only (to help you with building your layout)
8. Getting started on Family Map Client
    a. Read the entire spec
    b. Think about how to organize your client model
        i. Client model is more complex than the server one because of the need to support filters (see the FilterActivity in the spec)
    c. Can layout your login activity
        i. The spec says to create your login as a fragment in your main activity. We haven't talked about fragments in class yet, so for now, you can just create a LoginActivity with a corresponding layout matching what you see in the spec
        ii. **Hint:** Use a TableLayout (see the code example discussed in class and posted in the lecture notes)
        iii. It will be easy to use your layout in a fragment later