Status Report: MVC Application,

**Name:** Abdullah Bin Asad

**Application Name:** Quick-Flip

**Git Repository:** <https://github.com/Abdullah2Cool/Quick-Flip>

**Functionality:** Flip a Coin or Roll a Dice

**Date:** September 11 – 15

**Specs:**

This program is there to help me get my feet with android studio. It touches on the basics of layouts, loading assets and java code and how they all fit together to create an app.

**Major Challenge/Lessons:**

The hardest part about this was understanding the relationship between fragments, activities and tabs. They require a lot of components to work together.

* Each fragment is a class that extends Fragment and requires its own layout that needs to be inflated

View view = inflater.inflate(R.layout.***flip\_a\_coin\_fragment\_layout***, container, **false**);

* The fragments are held within a container called a ViewPager defined in the xml for the Main Activity:

<**android.support.v4.view.ViewPager  
 android:id="@+id/container"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior"** />

* The ViewPager needs to be setup with a FragmentPagerAdapter. I created a helper class to do that for me:

**public class** SectionsPagerAdapter **extends** FragmentPagerAdapter {  
 */\*  
 This is a helper class for managing fragments  
 FragmentPagerAdapter deals with how fragments are stored in memory (this one keeps them in memory)  
 This adapter is good for tabs or 3-4 fragments  
 \*/* **private final** List<Fragment> **mFragmentList** = **new** ArrayList<>(); *// holds all the fragments* **private final** List<String> **mFragmentTitleList** = **new** ArrayList<>(); *// holds the titles for all the fragments* **public** SectionsPagerAdapter(FragmentManager fm) { *// default constructor (do not change)* **super**(fm);  
 }  
 @Override  
 **public** Fragment getItem(**int** position) { *// returns the fragment at index position* **return mFragmentList**.get(position);  
 }  
  
 @Override  
 **public** CharSequence getPageTitle(**int** position) { *// returns the name of the fragment at index position* **return mFragmentTitleList**.get(position);  
 }  
  
 @Override  
 **public int** getCount() { *// returns the total number of fragments being handled by this FragmentPagerAdapter* **return mFragmentList**.size();  
 }  
  
 **public void** addFragment(Fragment fragment, String title) { *// this adds a new fragment to this Fragment manager* **mFragmentList**.add(fragment);  
 **mFragmentTitleList**.add(title);  
 }  
}

* Each fragment can be added to the ViewPager like this:

**private void** setupViewPager(ViewPager viewPager) {  
 SectionsPagerAdapter adapter = **new** SectionsPagerAdapter(getSupportFragmentManager()); *// declare a new 'fragment holder'* adapter.addFragment(**new** FlipACoin\_Fragment(), **"Flip a Coin"**); *// add a fragment* adapter.addFragment(**new** RollTheDice\_Fragment(), **"Roll the Dice"**); *// add a fragment* viewPager.setAdapter(adapter); *// set the adapter to the viewpager(container for fragments)*}

* The Tabs layout can then be setup with the ViewPager like this to display each fragment on a tab:

tabLayout.setupWithViewPager(**mViewPager**); *// setup the tab with the viewpager*

**Sources:**

<https://www.youtube.com/watch?v=UqtsyhASW74> (tabs + fragments)

<https://developer.android.com/training/implementing-navigation/lateral.html> (tabs)

<https://www.youtube.com/watch?v=00LLd7qr9sA> (tabs + fragments)

Instagram Clone

**Tutorial Series:** <https://www.youtube.com/watch?v=qpJRgr6HzAw&list=PLgCYzUzKIBE9XqkckEJJA0I1wVKbUAOdv>

**Git Repository:**

<https://github.com/Abdullah2Cool/Instagram-Clone>

**Functionality:** there are a few activities with bottom navigation and you can login or register as new user.

**Note:** will be documented soon.