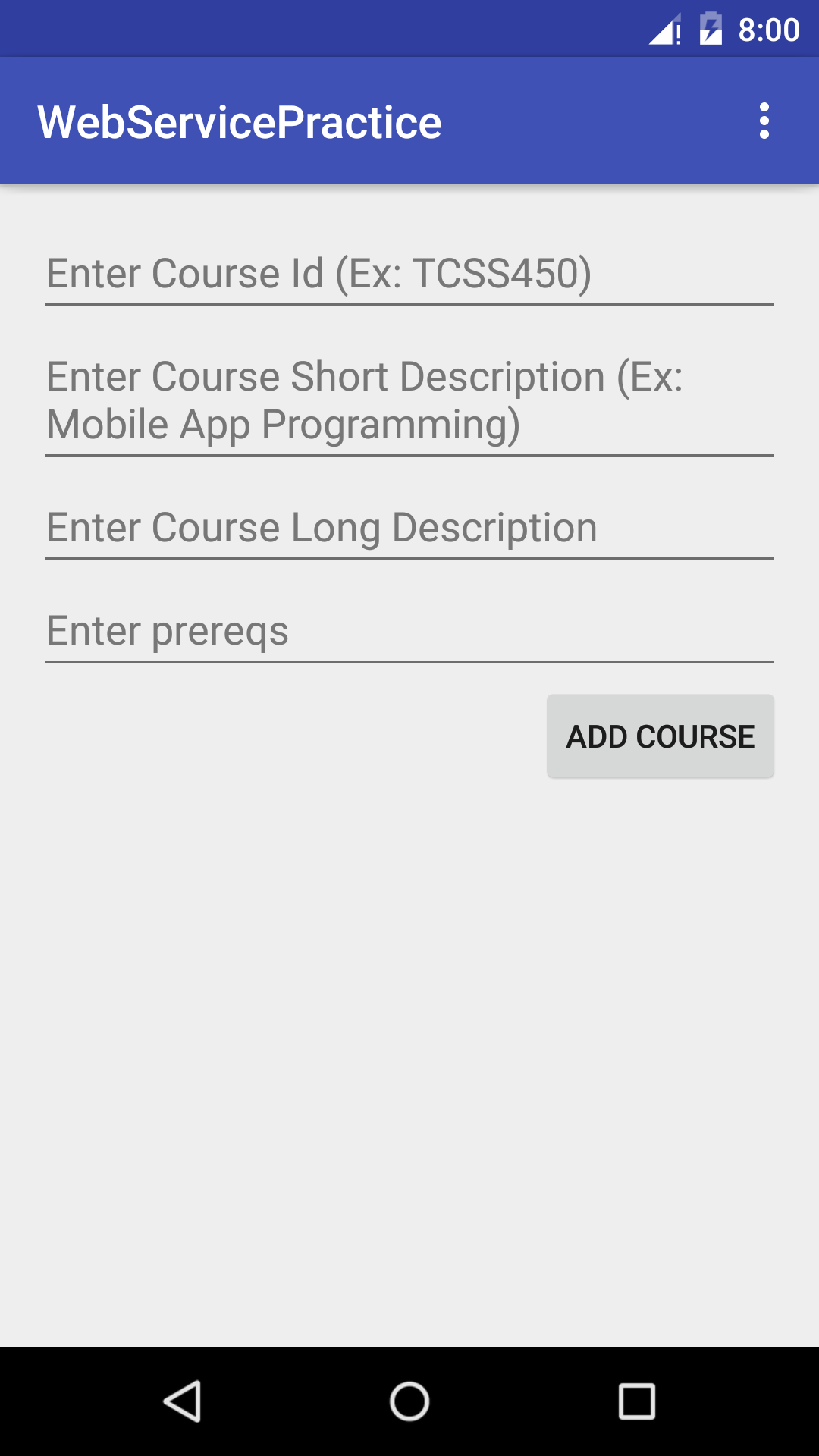
# Web Services Lab – Android Portion Final Part

### Add a new course using web service

1. Modify the layout of the CourseActivity to change the mail icon to an add icon.

<**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"  
 android:src="@android:drawable/ic\_input\_add"**/>

1. Create a new Fragment class called CourseAddFragment using Fragment (blank) template. We will use this class to add a new course using a web service.
2. Make the following changes to course\_add\_fragment.xml. It should look like below.



* 1. Modify the layout of course\_add\_fragment.xml to use LinearLayout with vertical orientation and create 4 EditText views that can be used to input the four different pieces of information we need to create a course – id, short description, long description, prereqs.
  2. Consider using a hint so that the user knows what to enter in each EditText view as shown below. You will have to do this for all of them. Make the hint is a string resource as well.

<**string name="course\_id\_hint"**>Enter Course Id (Ex: TCSS450) </**string**>  
<**string name="course\_short\_desc\_hint"**>Enter Course Short Description (Ex: Mobile App Programming) </**string**>  
<**string name="course\_long\_desc\_hint"**>Enter Course Long Description</**string**>  
<**string name="course\_prereqs\_hint"**>Enter prereqs</**string**>  
<**string name="add\_course\_text"**>Add Course</**string**>

<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:id="@+id/fragment\_add\_course"  
 android:name="edu.uw.tacoma.mmuppa.webservicepractice.CourseAddFragment"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 tools:context=".CourseDetailFragment"**>  
  
 <**EditText  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="@string/course\_id\_hint"  
 android:id="@+id/add\_course\_id"** />  
  
 <**EditText  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="@string/course\_short\_desc\_hint"  
 android:id="@+id/add\_course\_short\_desc"** />  
  
 <**EditText  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="@string/course\_long\_desc\_hint"  
 android:id="@+id/add\_course\_long\_desc"** />  
  
 <**EditText  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="@string/course\_prereqs\_hint"  
 android:id="@+id/add\_course\_prereqs"** />  
  
 <**Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"   
 android:layout\_gravity="right"  
 android:text="@string/add\_course\_text"   
 android:onClick="addCourse"** />  
</**LinearLayout**>

1. Make the following changes to CourseAddFragment
   1. Add a Listener interface and create a member variable to access it.

**private** CourseAddListener **mListener**;  
  
**public interface** CourseAddListener {  
 **public void** addCourse(String url);  
}

* 1. Make sure the onAttach uses this listener in the code.

@Override  
**public void** onAttach(Context context) {  
 **super**.onAttach(context);  
 **if** (context **instanceof** CourseAddListener) {  
 **mListener** = (CourseAddListener) context;  
 } **else** {  
 **throw new** RuntimeException(context.toString()  
 + **" must implement CourseAddListener"**);  
 }  
}

* 1. Modify the onCreateView method of CourseAddFragment to hide the floating action button.

@Override  
**public** View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
 *// Inflate the layout for this fragment* View v = inflater.inflate(R.layout.***fragment\_add\_course***, container, **false**);  
 FloatingActionButton floatingActionButton = (FloatingActionButton)  
 getActivity().findViewById(R.id.***fab***);  
 floatingActionButton.hide();  
 **return** v;  
}

* 1. Create a constant URL that you can use to build the string for adding the course.

**private final static** String ***COURSE\_ADD\_URL*** = **"http://cssgate.insttech.washington.edu/~uwnetid/addCourse.php?"**;

* 1. Create member variables to access the EditText fields in onCreateView.

**private** EditText **mCourseIdEditText**;  
**private** EditText **mCourseShortDescEditText**;  
**private** EditText **mCourseLongDescEditText**;  
**private** EditText **mCoursePrereqsEditText**;

* 1. Change onCreateView to reference the fields above.

@Override  
**public** View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
 *// Inflate the layout for this fragment* View v = inflater.inflate(R.layout.***fragment\_add\_course***, container, **false**);  
  
 **mCourseIdEditText** = (EditText) v.findViewById(R.id.***add\_course\_id***);  
 **mCourseShortDescEditText** = (EditText) v.findViewById(R.id.***add\_course\_short\_desc***);  
 **mCourseLongDescEditText** = (EditText) v.findViewById(R.id.***add\_course\_long\_desc***);  
 **mCoursePrereqsEditText** = (EditText) v.findViewById(R.id.***add\_course\_prereqs***);  
  
  
 FloatingActionButton floatingActionButton = (FloatingActionButton)  
 getActivity().findViewById(R.id.***fab***);  
 floatingActionButton.hide();

* 1. Create a method that will build the url for calling the AsyncTask.

**private** String buildCourseURL(View v) {  
  
 StringBuilder sb = **new** StringBuilder(***COURSE\_ADD\_URL***);  
  
 **try** {  
  
 String courseId = **mCourseIdEditText**.getText().toString();  
 sb.append(**"id="**);  
 sb.append(courseId);  
  
  
 String courseShortDesc = **mCourseShortDescEditText**.getText().toString();  
 sb.append(**"&shortDesc="**);  
 sb.append(URLEncoder.*encode*(courseShortDesc, **"UTF-8"**));  
  
  
 String courseLongDesc = **mCourseLongDescEditText**.getText().toString();  
 sb.append(**"&longDesc="**);  
 sb.append(URLEncoder.*encode*(courseLongDesc, **"UTF-8"**));  
  
 String coursePrereqs = **mCoursePrereqsEditText**.getText().toString();  
 sb.append(**"&prereqs="**);  
 sb.append(URLEncoder.*encode*(coursePrereqs, **"UTF-8"**));  
  
 Log.*i*(**"CourseAddFragment"**, sb.toString());  
  
 }  
 **catch**(Exception e) {  
 Toast.*makeText*(v.getContext(), **"Something wrong with the url"** + e.getMessage(), Toast.***LENGTH\_LONG***)  
 .show();  
 }  
 **return** sb.toString();  
}

* 1. Call the buildURL method in the onClickListener in onCreateView.

@Override  
**public** View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
 *// Inflate the layout for this fragment* View v = inflater.inflate(R.layout.***fragment\_add\_course***, container, **false**);  
  
 **mCourseIdEditText** = (EditText) v.findViewById(R.id.***add\_course\_id***);  
 **mCourseShortDescEditText** = (EditText) v.findViewById(R.id.***add\_course\_short\_desc***);  
 **mCourseLongDescEditText** = (EditText) v.findViewById(R.id.***add\_course\_long\_desc***);  
 **mCoursePrereqsEditText** = (EditText) v.findViewById(R.id.***add\_course\_prereqs***);  
  
  
 FloatingActionButton floatingActionButton = (FloatingActionButton)  
 getActivity().findViewById(R.id.***fab***);  
 floatingActionButton.hide();  
  
 Button addCourseButton = (Button) v.findViewById(R.id.***add\_course\_button***);  
 addCourseButton.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 String url = buildCourseURL(v);  
 **mListener**.addCourse(url);  
 }  
 });  
  
 **return** v;  
}

1. You will have to edit the onCreateView methods of other fragments (CourseListFragment, CourseDetailFragment) to show the floating action button. Add it before the return the statement in each of their onCreateView methods.

FloatingActionButton floatingActionButton = (FloatingActionButton)  
 getActivity().findViewById(R.id.***fab***);  
floatingActionButton.show();

1. Add a web service (addCourse.php in PHP Code zip file on Canvas) to insert the data into the Courses table. Make sure to customize for your account and ftp onto the web server.
2. Test it with your own uwnetid on a browser first.

<http://cssgate.insttech.washington.edu/~uwnetid/addCourse.php?id=TCSS305&shortDesc=Programming%20Practicum&longDesc=Program%20all%20the%20time&prereqs=TCSS143>

1. Make the following changes to CourseActivity:
   1. Implement the CourseAddListener

**public class** CourseActivity **extends** AppCompatActivity  
 **implements** CourseListFragment.OnListFragmentInteractionListener,  
 CourseAddFragment.CourseAddListener {

* 1. Implement the method addCourse(String) and leave it blank for now.
  2. Change the code for Floating Action Button listener to launch the fragment.

FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.***fab***);  
fab.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 CourseAddFragment courseAddFragment = **new** CourseAddFragment();  
 getSupportFragmentManager().beginTransaction()  
 .replace(R.id.***fragment\_container***, courseAddFragment)  
 .addToBackStack(**null**)  
 .commit();  
 }  
});

1. Run the app and see if it launches the CourseAddFragment.
2. Let’s make it work next. Make the following changes to CourseActivity:
   1. Create an AsyncTask class called CourseAddAsyncTask that will allow us to call the service for adding a course.

**private class** AddCourseTask **extends** AsyncTask<String, Void, String> {  
  
  
 @Override  
 **protected void** onPreExecute() {  
 **super**.onPreExecute();  
 }  
  
 @Override  
 **protected** String doInBackground(String... urls) {  
 String response = **""**;  
 HttpURLConnection urlConnection = **null**;  
 **for** (String url : urls) {  
 **try** {  
 URL urlObject = **new** URL(url);  
 urlConnection = (HttpURLConnection) urlObject.openConnection();  
  
 InputStream content = urlConnection.getInputStream();  
  
 BufferedReader buffer = **new** BufferedReader(**new** InputStreamReader(content));  
 String s = **""**;  
 **while** ((s = buffer.readLine()) != **null**) {  
 response += s;  
 }  
  
 } **catch** (Exception e) {  
 response = **"Unable to add course, Reason: "** + e.getMessage();  
 } **finally** {  
 **if** (urlConnection != **null**)  
 urlConnection.disconnect();  
 }  
 }  
 **return** response;  
 }  
  
  
 */\*\*  
 \* It checks to see if there was a problem with the URL(Network) which is when an  
 \* exception is caught. It tries to call the parse Method and checks to see if it was successful.  
 \* If not, it displays the exception.  
 \*  
 \** ***@param result*** *\*/* @Override  
 **protected void** onPostExecute(String result) {  
 *// Something wrong with the network or the URL.* **try** {  
 JSONObject jsonObject = **new** JSONObject(result);  
 String status = (String) jsonObject.get(**"result"**);  
 **if** (status.equals(**"success"**)) {  
 Toast.*makeText*(getApplicationContext(), **"Course successfully added!"** , Toast.***LENGTH\_LONG***)  
 .show();  
 } **else** {  
 Toast.*makeText*(getApplicationContext(), **"Failed to add: "** + jsonObject.get(**"error"**)  
 , Toast.***LENGTH\_LONG***)  
 .show();  
 }  
 } **catch** (JSONException e) {  
 Toast.*makeText*(getApplicationContext(), **"Something wrong with the data"** +  
 e.getMessage(), Toast.***LENGTH\_LONG***).show();  
 }  
 }  
}

* 1. Add the code to implement the listener method to add the course.

@Override  
**public void** addCourse(String url) {  
  
 AddCourseTask task = **new** AddCourseTask();  
 task.execute(**new** String[]{url.toString()});

// Takes you back to the previous fragment by popping the current fragment out.  
 getSupportFragmentManager().popBackStackImmediate();   
}

1. Run the app and test the app to see if a new course gets added.