ITMD 455/555 *Intelligent Device Applications* Lab 8

#### SQLite Books Database (part 3)- 50 points

**Introduction**. This lab will have you continue your work from lab 7. This time around you’ll allow the user, via a Spinner dropdown control which acts as a data bound control, to **return** analytical information retrieved from your database table.

**Objective:** For this lab you will continue working with the BookDB database with some

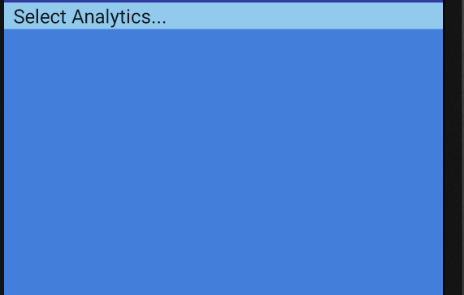
slight modifications to the UI involving adding a Spinner “widget” to allow

for some analytical information provided from your books table.

**STEP 1** Modify the layout or create a new layout for your MainActivity file to allow for only a spinner

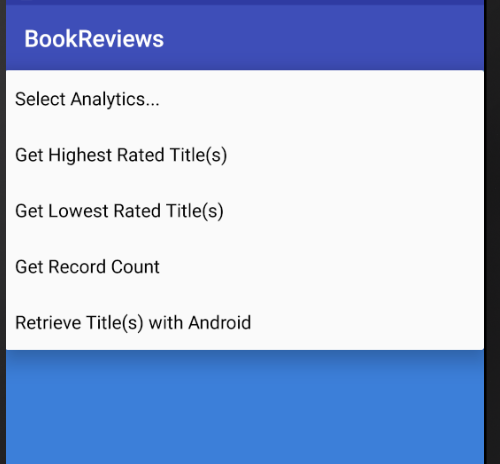
control to be displayed at runtime.

Include the following item to appear first in the containment of your spinner control.



As the user clicks on the first item (shown above) include the following drop down

list of items that will appear for the user to select:



As the user **selects** any valid choice from the spinner drop down list you are to

trigger the analytic result into a toast message.

Ex. Get Highest Rated Titles, will show the highest ranked or “rated” title(s) in a

toast message.

Here is a sample toast message retrieving titles with Android



**STEP 2**  In your MainActivity include the following code tweaks:

1. Include the implementation of a listener for your spinner in your class opening

**public class** MainActivity **extends** AppCompatActivity **implements**

**AdapterView.OnItemSelectedListener**

1. Declare your db object in your class so it is a class member variable.

Your onCreate(…) method should therefore initialize the member variable

as follows:

db = new SqlHelper(this);

This will allow for visibilty of the object within the spinner listener as you will see.

1. In your onCreate(…) method add in the following source code to create your spinner object and list of spinner items as follows:

*// Spinner element* Spinner spinner;  
 *// Spinner element* spinner = (Spinner) findViewById(R.id.***spinner***);  
  
 *//Create spinner item listing* List<String> blist = **new** ArrayList<String>();  
 blist.add(**"Get Highest Rated Title(s)"**);  
 blist.add(**"Get Lowest Rated Title(s)"**);  
 blist.add(**"Retrieve Title(s) with Android"**);  
 blist.add(**"Get Record Count"**);  
  
 *//Sort list in Alphabetical order* Collections.*sort*(blist, **new** Comparator<String>() {  
 @Override  
 **public int** compare(String lhs, String rhs) {  
 **return** lhs.compareTo(rhs);  
 }  
 });  
 blist.add(0, **"Select Analytics..."**);

ArrayAdapter<String> adapter = **new**

ArrayAdapter<String>(MainActivity.**this**,  
 android.R.layout.***simple\_spinner\_item***, blist);

adapter.setDropDownViewResource(android.R.layout.***simple\_spinner\_dropdown\_i***

***tem***);  
 spinner.setAdapter(adapter);  
 spinner.setWillNotDraw(**false**);  
 spinner.setOnItemSelectedListener(**this**);  
  
} *//end onCreate() method*

1. Ok for the fun part. After your onCreate(…) method closes out, include the needed implementations for the OnItemSelectedListener shown below so when the user chooses one of the list items in the pulldown a query will trigger!

@Override  
**public void** onItemSelected(AdapterView<?> arg0, View arg1, **int** position,  
 **long** arg3) {  
  
 **switch** (position) {  
 **case** 1:  
 *//get query result for Highest rated title(s)  
 // display query result(s) in a Toast message*  
 Toast.*makeText*(**this**, **"Title :: "** + **db**.getRatingMax(),  
 Toast.***LENGTH\_LONG***).show();  
 **break**;  
 **case** 2:

*//get query result for lowest rated title(s)  
 // display query result(s) in a Toast message*  
 Toast.*makeText*(**this**, **"Title :: "** + **db**.getRatingMin(),  
 Toast.***LENGTH\_LONG***).show();  
 **break**;  
 **case** 3:  
Toast.*makeText*(**this**, **"Record Count :: "** + **db**.getTotal(),  
 Toast.***LENGTH\_LONG***).show();  
 **break**;  
 **case** 4:  
Toast.*makeText*(**this**, **"Title :: "** + **db**.getBooks(),  
 Toast.***LENGTH\_LONG***).show();  
**break**;  
 }  
  
 }  
@Override  
**public void** onNothingSelected(AdapterView<?> arg0) {   
}

All: include the needed toast message results for Case 1 & Case 2 above by grabbing the query results for each of the method calls in the respective cases namely for getRatingMax() and getRatingMin(). These methods you will need to define in your SQLHelper class.

To code added methods within your SQLHelper follow the same code patterns defined in your other methods listed in the class. Many of your query results should follow a similar pattern as shown next:

For example getRatingMax needs to pull the max rating from the books table and return the title(s).

**public** String getRatingMax() {  
StringBuilder s = **new** StringBuilder();  
String selectQuery = **" your select query statement here "**;

SQLiteDatabase db = **this**.getReadableDatabase();  
Cursor cursor = db.rawQuery(selectQuery,**null**);  
 **if** (cursor.moveToFirst()) {  
 **do** {  
 s.append(cursor.getString(1)); *//get author's value* } **while** (cursor.moveToNext());  
 }  
 cursor.close();  
 db.close();  
 **return** s.toString();  
}

Grads:

Include query results for Cases 3 & 4 above as well. Include also an additional query of your choice! Examples maybe where id > 1 or show overall average rating for all titles, etc.

Include also some nice style effects to your Spinner view such as a nice sized Font for spinner items as well as a differing background and/or text color. You can easily set a theme in your **styles.xml** file, etc.

***Note:* IF there is more than one title retrieved for any query make sure to sort it by title and include nice line spacing as well, ex. new lines for each title.**

**STEP 3 Snapshot results**

Snapshot your Toast message results for *each* of the spinner choices you are responsible for.

**Include in a zip file your project files and seperately, your doc file with your**

**snapshots and all relevant files into BB for credit.**