ASSIGNMENT 1: DESIGN DOCUMENT

Andrew Thompson *CSC-421 Mobile Development*

PURPOSE

This document details the layout and class design of Mobile Development Assignment 1, Twitter Tag Search.

LAYOUT DESIGN

The application layout consists of two XML files. The primary layout, activity_main.xml, is a TableLayout with 5 TableRows. TableRow4 contains a ScrollView and another TableLayout. This second TableLayout, TableLayout2, is used to inflate other TableRow views. These inflated views are designed in the second layout file, tagged_search.xml. This layout consists of 3 buttons, the button with the tag name, and edit and delete buttons.

The primary layout also contains EditText fields, used for input of Twitter search query and tag name, and buttons to save the tag or clear all tags.

The following diagram details the layout items:

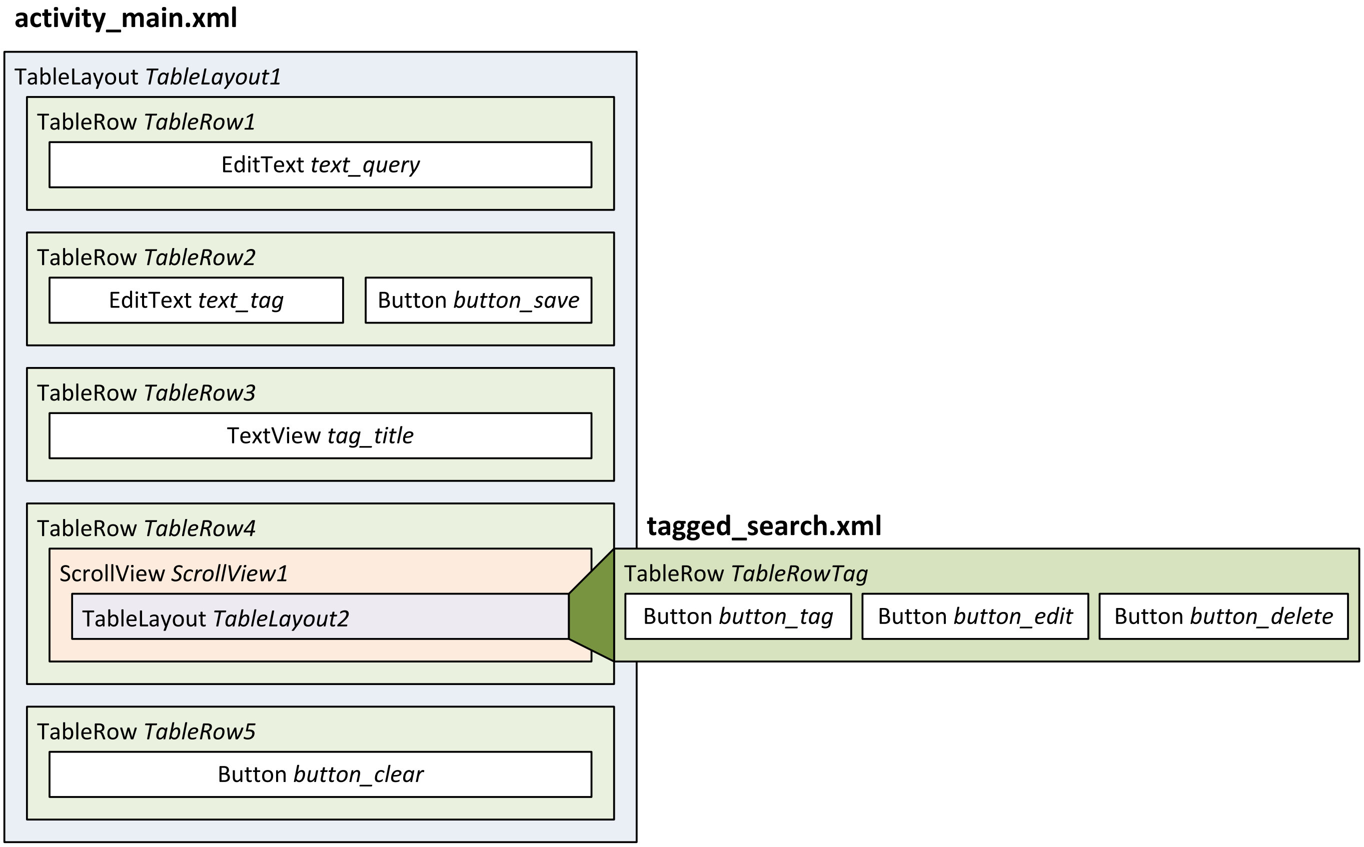

Figure 1: Application Layout

CLASS DESIGN

This application only uses one class called **MainActivity**. This class extends Android's Activity base class. We elected to use anonymous inner classes for button listeners instead of implementing it directly on the class.

We used several class members for tracking tag searches and layout objects. Specifically, the **SharedPreference** object called **_savedSearches** is used to store the Twitter query and tag data. The query string is stored in a key/value pair with the tag used as the key.

The following methods are included in the class. Detailed information can be found on the documentation located at GitHub - Assignment1.pdf.

onCreate

Sets the content view for the application and defines button objects and the table layout object that tag views will be inflated into. The two buttons defined here, save and clear, use anonymous inner classes to handle the click listener events. These events fire the **saveTag** and **sendClearTagAlert** methods. This method then calls **getTaggedSearches** which refreshes the tagged search view.

saveTag

Saves the Twitter query string into the shared preferences object with the tag string as the key. If the query or tag field is left empty, an error window is displayed by calling **sendCheckInputAlert**. If the class member **_editTag** is set, the item is removed from shared preferences first. This method then clears the fields and calls **getTaggedSearches** which refreshes the tagged search view.

getTaggedSearches

First, any existing views are removed from the **TableLayout2** object located in **TableRow4**. Next, the shared preference array is sorted and looped through to add new views back into the previous **TableLayout2**. This method also instantiates the new row's tag, edit and delete buttons and their click listeners. Again, this uses via anonymous inner classes. Clicking the tag button fires **sendQueryIntent**, the edit button calls **editTag** and the delete button calls a confirmation window via **sendDeleteTagAlert.**

sendQueryIntent

Searches the shared preference with the tag as a key. An intent is fired that sends the Twitter search URL and query.

editTag

Updates the query and tag fields with the selected tagged search. Also sets the **_editTag** class member.

deleteTag

Removes a key/value pair from the shared preferences based on the tag as a key. This method then calls **getTaggedSearches** to refresh the tagged search view.

deleteAllTags

Removes all tagged searches and any inflated views in **TableLayout2**. This is called when the button to clear all tags is clicked.

sendCheckInputAlert, sendDeleteTagAlert, sendClearTagsAlert

These methods display confirmation or error windows. When a tag's delete button is clicked, the **sendDeleteTagAlert** is fired and if a positive confirmation is entered, the **deleteTag** method is executed. When the clear all tags button is clicked, the **sendClearTagsAlert** is fired and a positive confirmation executes the **deleteAllTags** method.