420-603-AB Assignment II February 19th, 2013
Winter 2013 Worth 7%

Assignment II: Android Twitter App

(Due on March 1st, 2013)

In this assignment, you will create and Android App which does the following:

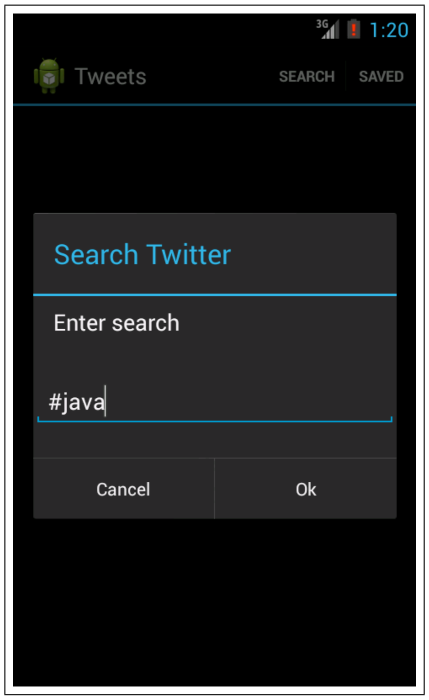
- given a search string, uses the Twitter API to retrieve tweets matching the search,
- presents the user the tweets as a list, using the Android ListView, and
- allows the user to store a tweet to a local database, residing on the device.

The App will consist of one activity containing a ListView and a menu for switching between Tweets drawn from two *sources:* from twitter.com and from a local database.

1 User Interface

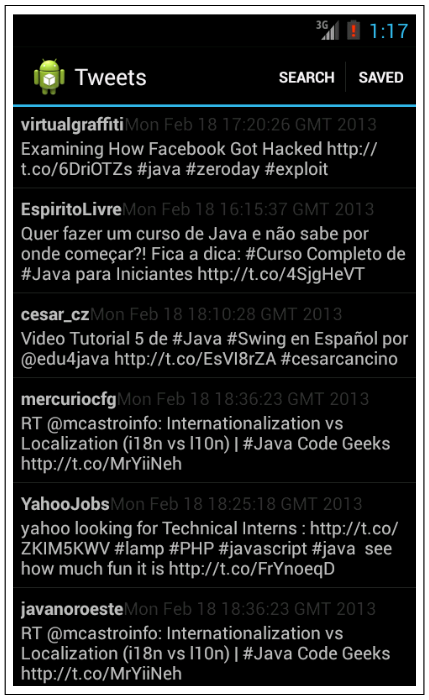
1.1 Search dialog

User is presented a dialog to obtain a search string.



1.2 List of tweets from twitter.com

The list view source is the search results.



virtualgraffitiMon Feb 18 17:20:26 GMT 2013
Examining How Facebook Got Hacked http://t.co/6DriOTZs #java #zeroday #exploit

EspiritoLivreMon Feb 18 16:15:37 GMT 2013
Quer fazer um curso de Java e não sabe por onde começar?! Fica a dica: #Curso Completo de #Java para Iniciantes http://t.co/4SjgHeVT

cesar_czMon Feb 18 18:10:28 GMT 2013
Video Tutorial 5 de #Java #Swing en Español por @edu4java http://t.co/EsVI8rZA #cesarcancino

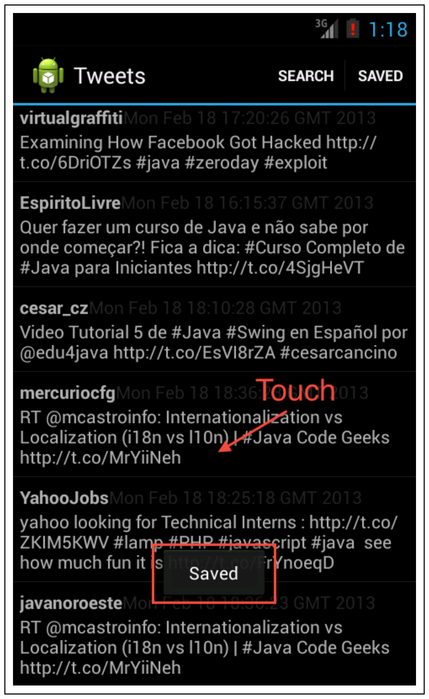
mercuriocfgMon Feb 18 18:36:23 GMT 2013
RT @mcastroinfo: Internationalization vs Localization (i18n vs l10n) | #Java Code Geeks http://t.co/MrYiiNeh

YahooJobsMon Feb 18 18:25:18 GMT 2013
yahoo looking for Technical Interns : http://t.co/ZKIM5KWV #lamp #PHP #javascript #java see how much fun it is http://t.co/FrYnoeqD

javanoroesteMon Feb 18 18:36:23 GMT 2013
RT @mcastroinfo: Internationalization vs Localization (i18n vs l10n) | #Java Code Geeks http://t.co/MrYiiNeh

1.3 Saving

When the user select an item from the list, the App will save the tweet to the local database and toast a message “Saved”.



virtualgraffitiMon Feb 18 17:20:26 GMT 2013
Examining How Facebook Got Hacked http://t.co/6DriOTZs #java #zeroday #exploit

EspiritoLivreMon Feb 18 16:15:37 GMT 2013
Quer fazer um curso de Java e não sabe por onde começar?! Fica a dica: #Curso Completo de #Java para Iniciantes http://t.co/4SjgHeVT

cesar_czMon Feb 18 18:10:28 GMT 2013
Video Tutorial 5 de #Java #Swing en Español por @edu4java http://t.co/EsVI8rZA #cesarcancino

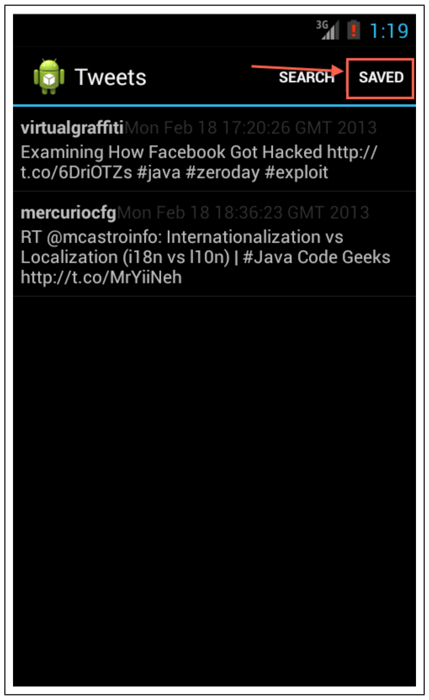
mercuriocfgMon Feb 18 18:36:23 GMT 2013
RT @mcastroinfo: Internationalization vs Localization (i18n vs l10n) | #Java Code Geeks http://t.co/MrYiiNeh

YahooJobsMon Feb 18 18:25:18 GMT 2013
yahoo looking for Technical Interns : http://t.co/ZKIM5KWV #lamp #PHP #javascript #java see how much fun it is http://t.co/FrYnoeqD

javanoroesteMon Feb 18 18:36:23 GMT 2013
RT @mcastroinfo: Internationalization vs Localization (i18n vs l10n) | #Java Code Geeks http://t.co/MrYiiNeh

1.4 List of tweets from local database.

Selecting the “Saved” menu action causes the list view to use the tweet database as a source.

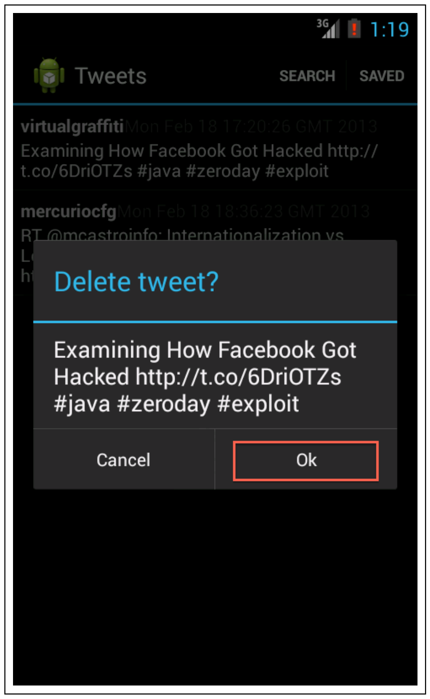


virtualgraffiti Mon Feb 18 17:20:26 GMT 2013
Examining How Facebook Got Hacked http://t.co/6DriOTZs #java #zeroday #exploit

mercuriocfg Mon Feb 18 18:36:23 GMT 2013
RT @mcastroinfo: Internationalization vs Localization (i18n vs l10n) | #Java Code Geeks http://t.co/MrYiiNeh

1.5 Deleting

When a tweet is selected from the list, a dialog prompts confirmation before deleting from the database.

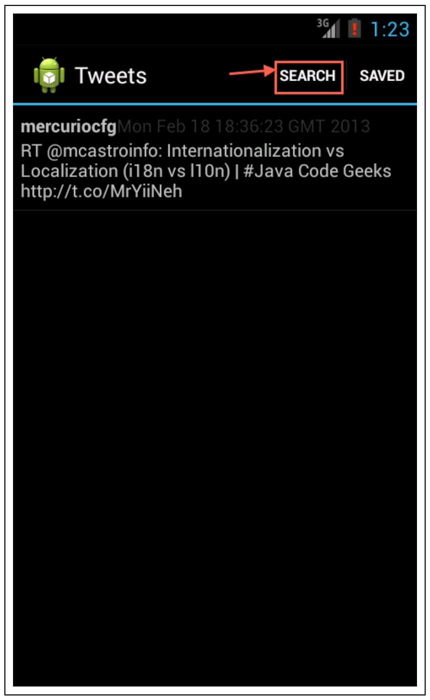


Delete tweet?

Examining How Facebook Got Hacked http://t.co/6DriOTZs #java #zeroday #exploit

1.6 Switching between lists

Selecting "Search" from the action menu causes the list view to use return to the search results.



mercuriocfg Mon Feb 18 18:36:23 GMT 2013
RT @mcastroinfo: Internationalization vs Localization (i18n vs l10n) | #Java Code Geeks http://t.co/MrYiiNeh

2 Provided classes, etc...

Use the starter application with the class MainActivity provided. The project contains the following layouts

1. XML layouts for main activity and a "tweet" view.
2. XML layout for an Android menu. In MainActivity, the method `onOptionsItemSelected(MenuItem item)` is called when these items are selected from the "action" menu.

2.1 Tweet "source" classes

Two classes are given to facilitate the above: TwitterSource and TweetDbSource. The latter is partially implemented. Both of these use the class Tweet to store a tweet. Read the provided documentation for these classes.

Implement methods in TweetDbSource to create a "saved tweets" database and table, and perform insert, delete and select operations. Use the SQLiteDatabase object returned from the following methods:

```
SQLiteDatabase db = this.getWritableDatabase();
```

or, for read-only operations

```
SQLiteDatabase db = this.getReadableDatabase();
```

See the reference page [1] and a tutorial[2].

3 Interfaces

To accomplish the above, there are three interfaces which must be implemented. You might have to implement these interfaces multiple times to accomplish different behaviours.

3.1 android.widget.ListAdapter

A list adapter will be used by Android to populate a ListView with data, it acts a "the bridge between a ListView and the data that backs the list"[3]. Write two classes which implements this interface: one for each tweet source.

Set the adapter using ListView's `setAdapter(ListAdapter)` method.

Notes:

1. For methods:

   ```
   registerDataSetObserver(DataSetObserver observer)
   unregisterDataSetObserver(DataSetObserver observer)
   ```

   create a DataSetObservable field and call `registerObserver()` or `unregisterObserver()` as needed. A TwitterSource will notify any observers after retrieving new tweets, so it's important to pass the DataSetObservable to this object.

2. For method:

   ```
   getView(int position, View convertView, ViewGroup parent)
   ```

   create a new view to hold an individual tweet. Use the layout "inflater" to create an view from the XML layout file. Ex:

   ```
   LayoutInflater inflater = (LayoutInflater)context.getSystemService(
                                                  Context.LAYOUT_INFLATER_SERVICE
                                              );
   View tweetView = inflater.inflate(R.layout.tweet, null);
   ```

[1] http://developer.android.com/reference/android/database/sqlite/SQLiteDatabase.html
[2] http://www.androidhive.info/2011/11/android-sqlite-database-tutorial/
[3] http://developer.android.com/reference/android/widget/ListAdapter.html

3.2 OnItemClickListener

A click listener interface registers click event handlers for UI components. For the ListView, set the *item* click listener using `setOnItemClickListener()`[4]. Implement this interface for both list sources, following the behaviours outlined above.

Notes:

1. If you need to use UI features (ex: toasting), be careful to pass the current *context* to the resulting listener class. See your class notes for an example.
2. (Optional) You may use anonymous classes to implement this interface. If you do so, you can obtain the current context directly from the encompassing class.

3.3 Comparator<Tweet>.

A *comparator* is used to compare objects of the specified type (here Tweet). It's only method,

```
int compare(Tweet lhs, Tweet rhs)
```

should return a negative number if `lhs` is less than `rhs`, a positive number if `lhs` is greater than `rhs` and 0 if they are the same. The TwitterSource uses a comparator to sort the retrieved tweets.

Notes:

1. Many data types in the Java standard library contain a method `compareTo()` with similar return values.
2. (Optional) You may use anonymous classes to implement this interface.

4 Observers

Android will use an *observer* mechanism to refresh the listview. An observer will be "notified" when data changes. For example, when new tweets are retrieved from twitter.com, the TwitterSource notifies all its observers of the update, letting them know to perform a refresh.

To keep a ListView up to date, Android will register data observers through your ListAdapter's `register` and `unregister` methods. Make sure that you implement these methods and store any observers in a DataSetObserable object. Sources that use this feature (i.e.: TwitterSource) should also have access to this set of observers.

[4]http://developer.android.com/reference/android/widget/AdapterView.OnItemClickListener.html

5 Requirements

1. Build on the MainActivity class.
2. Implement the missing features of the TweetDbSource.
3. Using the classes TwitterSource and TweetDbSource, write two classes which implement the ListAdapter interface.
4. Write a tweet comparator to supply to the TwitterSource for ordering.
5. Using the classes above, set the source for the main activity's list view.
6. Write item click handlers for the list view, one for each source.

6 Deliverables

- ☐ The minimum target framework should be Android 4.0 (Froyo), the maximum target framework should be Android 4.2 (IceCreamSandwich).
- ☐ Code should be commented and variable names chosen appropriately. Insufficient commenting will be penalized.
- ☐ Submit the entire project directory (from Eclipse workspace) compressed, on Léa.