```
/*-------------------------------------------------------------------------------------------------
 * Author:      Dan Cassidy and Deitel & Associates, Inc.
 * Date:        2015-07-30
 * Assignment:  HW7-1
 * Source File: MainActivity.java
 * Language:    Java
 * Course:      CSCI-C 490, Android Programming, MoWe 08:00
 * Note:        I simply modified the Twitter search application, altering variable names and
 *              comments to match the new focus of the application, as well making a few (mostly
 *              cosmetic) fixes/change to the code.  The strings and main layouts were also tweaked.
-------------------------------------------------------------------------------------------------*/
package dancassidy.favoritewebsites;

import android.app.AlertDialog;
import android.app.ListActivity;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.inputmethod.InputMethodManager;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemLongClickListener;
import android.widget.ArrayAdapter;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.TextView;

import java.util.ArrayList;
import java.util.Collections;

// MainActivity.java
// Manages your favorite websites for easy access and display in the device's web browser
public class MainActivity extends ListActivity {
    // name of SharedPreferences XML file that stores the saved searches
    private static final String WEBSITES = "websites";

    private EditText websiteAddressEditText; // EditText where user enters a website address
    private EditText websiteNameEditText; // EditText where user tags a website name
    private SharedPreferences websites; // user's favorite websites
    private ArrayList<String> names; // list of names for saved websites
    private ArrayAdapter<String> adapter; // binds website names to ListView

    // called when MainActivity is first created
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // get references to the EditTexts
        websiteAddressEditText = (EditText) findViewById(R.id.websiteAddressEditText);
        websiteNameEditText = (EditText) findViewById(R.id.websiteNameEditText);

        // get the SharedPreferences containing the user's saved websites
        websites = getSharedPreferences(WEBSITES, MODE_PRIVATE);

        // store the saved website names in an ArrayList then sort them
        names = new ArrayList<String>(websites.getAll().keySet());
        Collections.sort(names, String.CASE_INSENSITIVE_ORDER);

        // create ArrayAdapter and use it to bind website names to the ListView
        adapter = new ArrayAdapter<String>(this, R.layout.list_item, names);
        setListAdapter(adapter);

        // register listener to save a new or edited website
        ImageButton saveButton = (ImageButton) findViewById(R.id.saveButton);
        saveButton.setOnClickListener(saveButtonListener);

        // register listener that opens website when user touches a name
        getListView().setOnItemClickListener(itemClickListener);

        // set listener that allows user to delete or edit a website
        getListView().setOnItemLongClickListener(itemLongClickListener);
    } // end method onCreate

    // saveButtonListener saves a tag-query pair into SharedPreferences
    public OnClickListener saveButtonListener = new OnClickListener() {
        @Override public void onClick(View v) {
            // create website name if neither websiteAddressEditText nor websiteNameEditText is
            // empty
            if (websiteAddressEditText.getText().length() > 0 &&
                    websiteNameEditText.getText().length() > 0) {
                addTaggedSearch(websiteAddressEditText.getText().toString(),
                        websiteNameEditText.getText().toString());
                websiteAddressEditText.setText(""); // clear websiteAddressEditText
                websiteNameEditText.setText(""); // clear websiteNameEditText

                ((InputMethodManager) getSystemService(
                        Context.INPUT_METHOD_SERVICE)).hideSoftInputFromWindow(
                        websiteNameEditText.getWindowToken(), 0);
            } else // display message asking user to provide a website and a name
            {
                // create a new AlertDialog Builder
                AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity.this);

                // set dialog's message to display
                builder.setMessage(R.string.missingMessage);

                // provide an OK button that simply dismisses the dialog
                builder.setPositiveButton(R.string.OK, null);

                // create AlertDialog from the AlertDialog.Builder
                AlertDialog errorDialog = builder.create();
                errorDialog.show(); // display the modal dialog
            }
        } // end method onClick
    }; // end OnClickListener anonymous inner class

    // add new website to the save file, then refresh all Buttons
    private void addTaggedSearch(String website, String name) {
        // get a SharedPreferences.Editor to store new name/website pair
        SharedPreferences.Editor preferencesEditor = websites.edit();
        preferencesEditor.putString(name, website); // store current search
        preferencesEditor.apply(); // store the updated preferences

        // if name is new, add to and sort names, then display updated list
```

```
         if (!names.contains(name)) {
            names.add(name); // add new name
            Collections.sort(names, String.CASE_INSENSITIVE_ORDER);
            adapter.notifyDataSetChanged(); // rebind tags to ListView
         }
      }

      // itemClickListener launches a web browser to display website
      OnItemClickListener itemClickListener = new OnItemClickListener() {
         @Override
         public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
            // get the name of the website and the website itself
            String name = ((TextView) view).getText().toString();
            String urlString = websites.getString(name, "");

            // create an Intent to launch a web browser
            Intent webIntent = new Intent(Intent.ACTION_VIEW, Uri.parse(urlString));

            startActivity(webIntent); // launches web browser to view results
         }
      }; // end itemClickListener declaration

      // itemLongClickListener displays a dialog allowing the user to delete
      // or edit a saved website
      OnItemLongClickListener itemLongClickListener = new OnItemLongClickListener() {
         @Override
         public boolean onItemLongClick(AdapterView<?> parent, View view, int position,
                                        long id) {
            // get the name that the user long touched
            final String name = ((TextView) view).getText().toString();

            // create a new AlertDialog
            AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity.this);

            // set the AlertDialog's title
            builder.setTitle(getString(R.string.shareEditDeleteTitle, name));

            // set list of items to display in dialog
            builder.setItems(R.array.dialog_items, new DialogInterface.OnClickListener() {
               // responds to user touch by sharing, editing or
               // deleting a saved website
               @Override
               public void onClick(DialogInterface dialog, int which) {
                  switch (which) {
                     case 0: // share
                        shareSearch(name);
                        break;
                     case 1: // edit
                        // set EditTexts to match chosen name and website
                        websiteNameEditText.setText(name);
                        websiteAddressEditText.setText(websites.getString(name, ""));
                        break;
                     case 2: // delete
                        deleteSearch(name);
                        break;
                  }
               }
            } // end DialogInterface.OnClickListener
            ); // end call to builder.setItems

```

```java
            // set the AlertDialog's negative Button
            builder.setNegativeButton(getString(R.string.cancel),
                    new DialogInterface.OnClickListener() {
                        // called when the "Cancel" Button is clicked
                        public void onClick(DialogInterface dialog, int id) {
                            dialog.cancel(); // dismiss the AlertDialog
                        }
                    }
            ); // end call to setNegativeButton

            builder.create().show(); // display the AlertDialog
            return true;
        } // end method onItemLongClick
    }; // end OnItemLongClickListener declaration

    // allows user to choose an app for sharing a saved website's URL
    private void shareSearch(String name) {
        // retrieve the website
        String urlString = websites.getString(name, "");

        // create Intent to share urlString
        Intent shareIntent = new Intent();
        shareIntent.setAction(Intent.ACTION_SEND);
        shareIntent.putExtra(Intent.EXTRA_SUBJECT, getString(R.string.shareSubject));
        shareIntent.putExtra(Intent.EXTRA_TEXT, getString(R.string.shareMessage, urlString));
        shareIntent.setType("text/plain");

        // display apps that can share text
        startActivity(Intent.createChooser(shareIntent, getString(R.string.shareSearch)));
    }

    // deletes a website after the user confirms the delete operation
    private void deleteSearch(final String name) {
        // create a new AlertDialog
        AlertDialog.Builder confirmBuilder = new AlertDialog.Builder(this);

        // set the AlertDialog's message
        confirmBuilder.setMessage(
                getString(R.string.confirmMessage, name));

        // set the AlertDialog's negative Button
        confirmBuilder.setNegativeButton(getString(R.string.cancel),
                new DialogInterface.OnClickListener() {
                    // called when "Cancel" Button is clicked
                    public void onClick(DialogInterface dialog, int id) {
                        dialog.cancel(); // dismiss dialog
                    }
                }
        ); // end call to setNegativeButton

        // set the AlertDialog's positive Button
        confirmBuilder.setPositiveButton(getString(R.string.delete),
                new DialogInterface.OnClickListener() {
                    // called when "Cancel" Button is clicked
                    public void onClick(DialogInterface dialog, int id) {
                        names.remove(name); // remove name from names

                        // get SharedPreferences.Editor to remove saved website
                        SharedPreferences.Editor preferencesEditor = websites.edit();
                        preferencesEditor.remove(name); // remove website
                        preferencesEditor.apply(); // saves the changes

                        // rebind tags ArrayList to ListView to show updated list
                        adapter.notifyDataSetChanged();
                    }
                } // end OnClickListener
        ); // end call to setPositiveButton

        confirmBuilder.create().show(); // display AlertDialog
    } // end method deleteSearch
} // end class MainActivity

```