

Name: Dan Cassidy

Class: CSCI-C 490, Mobile Application Development

Assignment: Homework 7 Part 1

Date: 2015-07-30

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3      package="dancassidy.favoritewebsites"
4      android:versionCode="1"
5      android:versionName="1.0" >
6
7      <uses-sdk
8          android:minSdkVersion="18"
9          android:targetSdkVersion="19" />
10     <uses-permission android:name="android.permission.INTERNET"/>
11
12     <application
13         android:allowBackup="true"
14         android:icon="@drawable/ic_launcher"
15         android:label="@string/app_name"
16         android:theme="@style/AppTheme">
17         <activity
18             android:name="dancassidy.favoritewebsites.MainActivity"
19             android:label="@string/app_name" android:windowSoftInputMode="stateAlwaysHidden">
20             <intent-filter>
21                 <action android:name="android.intent.action.MAIN" />
22
23                 <category android:name="android.intent.category.LAUNCHER" />
24             </intent-filter>
25         </activity>
26     </application>
27
28 </manifest>
29
```

```
1  /*-----*/
2  * Author:      Dan Cassidy and Deitel & Associates, Inc.
3  * Date:        2015-07-30
4  * Assignment:  HW7-1
5  * Source File: MainActivity.java
6  * Language:    Java
7  * Course:      CSCI-C 490, Android Programming, MoWe 08:00
8  * Note:        I simply modified the Twitter search application, altering variable names and
9  *              comments to match the new focus of the application, as well making a few (mostly
10 *              cosmetic) fixes/change to the code.  The strings and main layouts were also tweaked.
11 -----*/
12 package dancassidy.favoritewebsites;
13
14 import android.app.AlertDialog;
15 import android.app.ListActivity;
16 import android.content.Context;
17 import android.content.DialogInterface;
18 import android.content.Intent;
19 import android.content.SharedPreferences;
20 import android.net.Uri;
21 import android.os.Bundle;
22 import android.view.View;
23 import android.view.View.OnClickListener;
24 import android.view.inputmethod.InputMethodManager;
25 import android.widget.AdapterView;
26 import android.widget.AdapterView.OnItemClickListener;
27 import android.widget.AdapterView.OnItemLongClickListener;
28 import android.widget.ArrayAdapter;
29 import android.widget.EditText;
30 import android.widget.ImageButton;
31 import android.widget.TextView;
32
33 import java.util.ArrayList;
34 import java.util.Collections;
35
36 // MainActivity.java
37 // Manages your favorite websites for easy access and display in the device's web browser
38 public class MainActivity extends ListActivity {
39     // name of SharedPreferences XML file that stores the saved searches
40     private static final String WEBSITES = "websites";
41
42     private EditText websiteAddressEditText; // EditText where user enters a website address
43     private EditText websiteNameEditText; // EditText where user tags a website name
44     private SharedPreferences websites; // user's favorite websites
45     private ArrayList<String> names; // list of names for saved websites
46     private ArrayAdapter<String> adapter; // binds website names to ListView
47
48     // called when MainActivity is first created
49     @Override
50     protected void onCreate(Bundle savedInstanceState) {
51         super.onCreate(savedInstanceState);
52         setContentView(R.layout.activity_main);
53
54         // get references to the EditTexts
55         websiteAddressEditText = (EditText) findViewById(R.id.websiteAddressEditText);
56         websiteNameEditText = (EditText) findViewById(R.id.websiteNameEditText);
57
58         // get the SharedPreferences containing the user's saved websites
59         websites = getSharedPreferences(WEBSITES, MODE_PRIVATE);
60     }
}
```

```
61         // store the saved website names in an ArrayList then sort them
62         names = new ArrayList<String>(websites.getAll().keySet());
63         Collections.sort(names, String.CASE_INSENSITIVE_ORDER);
64
65         // create ArrayAdapter and use it to bind website names to the ListView
66         adapter = new ArrayAdapter<String>(this, R.layout.list_item, names);
67         setListAdapter(adapter);
68
69         // register listener to save a new or edited website
70         ImageButton saveButton = (ImageButton) findViewById(R.id.saveButton);
71         saveButton.setOnClickListener(saveButtonListener);
72
73         // register listener that opens website when user touches a name
74         getListView().setOnItemClickListener(itemClickListener);
75
76         // set listener that allows user to delete or edit a website
77         getListView().setOnItemLongClickListener(itemLongClickListener);
78     } // end method onCreate
79
80     // saveButtonListener saves a tag-query pair into SharedPreferences
81     public OnClickListener saveButtonListener = new OnClickListener() {
82         @Override public void onClick(View v) {
83             // create website name if neither websiteAddressEditText nor websiteNameEditText is
84             // empty
85             if (websiteAddressEditText.getText().length() > 0 &&
86                 websiteNameEditText.getText().length() > 0) {
87                 addTaggedSearch(websiteAddressEditText.getText().toString(),
88                     websiteNameEditText.getText().toString());
89                 websiteAddressEditText.setText(""); // clear websiteAddressEditText
90                 websiteNameEditText.setText(""); // clear websiteNameEditText
91
92                 ((InputMethodManager) getSystemService(
93                     Context.INPUT_METHOD_SERVICE)).hideSoftInputFromWindow(
94                     websiteNameEditText.getWindowToken(), 0);
95             } else // display message asking user to provide a website and a name
96             {
97                 // create a new AlertDialog Builder
98                 AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity.this);
99
100                 // set dialog's message to display
101                 builder.setMessage(R.string.missingMessage);
102
103                 // provide an OK button that simply dismisses the dialog
104                 builder.setPositiveButton(R.string.OK, null);
105
106                 // create AlertDialog from the AlertDialog.Builder
107                 AlertDialog errorDialog = builder.create();
108                 errorDialog.show(); // display the modal dialog
109             }
110         } // end method onClick
111     }; // end OnClickListener anonymous inner class
112
113     // add new website to the save file, then refresh all Buttons
114     private void addTaggedSearch(String website, String name) {
115         // get a SharedPreferences.Editor to store new name/website pair
116         SharedPreferences.Editor preferencesEditor = websites.edit();
117         preferencesEditor.putString(name, website); // store current search
118         preferencesEditor.apply(); // store the updated preferences
119
120         // if name is new, add to and sort names, then display updated list
```

```
121         if (!names.contains(name)) {
122             names.add(name); // add new name
123             Collections.sort(names, String.CASE_INSENSITIVE_ORDER);
124             adapter.notifyDataSetChanged(); // rebind tags to ListView
125         }
126     }
127
128     // itemClickListener launches a web browser to display website
129     OnItemClickListener itemClickListener = new OnItemClickListener() {
130         @Override
131         public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
132             // get the name of the website and the website itself
133             String name = ((TextView) view).getText().toString();
134             String urlString = websites.getString(name, "");
135
136             // create an Intent to launch a web browser
137             Intent webIntent = new Intent(Intent.ACTION_VIEW, Uri.parse(urlString));
138
139             startActivity(webIntent); // launches web browser to view results
140         }
141     }; // end itemClickListener declaration
142
143     // itemLongClickListener displays a dialog allowing the user to delete
144     // or edit a saved website
145     OnItemLongClickListener itemLongClickListener = new OnItemLongClickListener() {
146         @Override
147         public boolean onItemLongClick(AdapterView<?> parent, View view, int position,
148             long id) {
149             // get the name that the user long touched
150             final String name = ((TextView) view).getText().toString();
151
152             // create a new AlertDialog
153             AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity.this);
154
155             // set the AlertDialog's title
156             builder.setTitle(getString(R.string.shareEditDeleteTitle, name));
157
158             // set list of items to display in dialog
159             builder.setItems(R.array.dialog_items, new DialogInterface.OnClickListener() {
160                 // responds to user touch by sharing, editing or
161                 // deleting a saved website
162                 @Override
163                 public void onClick(DialogInterface dialog, int which) {
164                     switch (which) {
165                         case 0: // share
166                             shareSearch(name);
167                             break;
168                         case 1: // edit
169                             // set EditTexts to match chosen name and website
170                             websiteNameEditText.setText(name);
171                             websiteAddressEditText.setText(websites.getString(name, ""));
172                             break;
173                         case 2: // delete
174                             deleteSearch(name);
175                             break;
176                     }
177                 }
178             }); // end DialogInterface.OnClickListener
179             // end call to builder.setItems
180         }
```

```
181         // set the AlertDialog's negative Button
182         builder.setNegativeButton(getString(R.string.cancel),
183             new DialogInterface.OnClickListener() {
184                 // called when the "Cancel" Button is clicked
185                 public void onClick(DialogInterface dialog, int id) {
186                     dialog.cancel(); // dismiss the AlertDialog
187                 }
188             });
189         // end call to setNegativeButton
190
191         builder.create().show(); // display the AlertDialog
192         return true;
193     } // end method onItemClick
194 }; // end onItemClick declaration
195
196 // allows user to choose an app for sharing a saved website's URL
197 private void shareSearch(String name) {
198     // retrieve the website
199     String urlString = websites.getString(name, "");
200
201     // create Intent to share urlString
202     Intent shareIntent = new Intent();
203     shareIntent.setAction(Intent.ACTION_SEND);
204     shareIntent.putExtra(Intent.EXTRA_SUBJECT, getString(R.string.shareSubject));
205     shareIntent.putExtra(Intent.EXTRA_TEXT, getString(R.string.shareMessage, urlString));
206     shareIntent.setType("text/plain");
207
208     // display apps that can share text
209     startActivity(Intent.createChooser(shareIntent, getString(R.string.shareSearch)));
210 }
211
212 // deletes a website after the user confirms the delete operation
213 private void deleteSearch(final String name) {
214     // create a new AlertDialog
215     AlertDialog.Builder confirmBuilder = new AlertDialog.Builder(this);
216
217     // set the AlertDialog's message
218     confirmBuilder.setMessage(
219         getString(R.string.confirmMessage, name));
220
221     // set the AlertDialog's negative Button
222     confirmBuilder.setNegativeButton(getString(R.string.cancel),
223         new DialogInterface.OnClickListener() {
224             // called when "Cancel" Button is clicked
225             public void onClick(DialogInterface dialog, int id) {
226                 dialog.cancel(); // dismiss dialog
227             }
228         });
229     // end call to setNegativeButton
230
231     // set the AlertDialog's positive Button
232     confirmBuilder.setPositiveButton(getString(R.string.delete),
233         new DialogInterface.OnClickListener() {
234             // called when "Cancel" Button is clicked
235             public void onClick(DialogInterface dialog, int id) {
236                 names.remove(name); // remove name from names
237
238                 // get SharedPreferences.Editor to remove saved website
239                 SharedPreferences.Editor preferencesEditor = websites.edit();
240                 preferencesEditor.remove(name); // remove website
```

```
241             preferencesEditor.apply(); // saves the changes
242
243             // rebind tags ArrayList to ListView to show updated list
244             adapter.notifyDataSetChanged();
245         }
246     } // end OnClickListener
247 }; // end call to setPositiveButton
248
249     confirmBuilder.create().show(); // display AlertDialog
250 } // end method deleteSearch
251 } // end class MainActivity
252
```

```
1  <GridLayout xmlns:android="http://schemas.android.com/apk/res/android"
2      xmlns:tools="http://schemas.android.com/tools"
3      android:id="@+id/gridLayout"
4      android:layout_width="match_parent"
5      android:layout_height="match_parent"
6      android:columnCount="1"
7      android:paddingBottom="@dimen/activity_vertical_margin"
8      android:paddingLeft="@dimen/activity_horizontal_margin"
9      android:paddingRight="@dimen/activity_horizontal_margin"
10     android:paddingTop="@dimen/activity_vertical_margin"
11     tools:context=".MainActivity" >
12
13     <LinearLayout
14         android:layout_gravity="fill_horizontal" >
15
16         <EditText
17             android:id="@+id/websiteNameEditText"
18             android:layout_width="0dp"
19             android:layout_height="wrap_content"
20             android:layout_gravity="bottom|fill_horizontal"
21             android:layout_weight="1"
22             android:hint="@string/websiteNamePrompt"
23             android:imeOptions="actionNext">
24             <requestFocus/>
25         </EditText>
26
27         <ImageButton
28             android:id="@+id/saveButton"
29             android:layout_width="wrap_content"
30             android:layout_height="wrap_content"
31             android:contentDescription="@string/saveDescription"
32             android:src="@android:drawable/ic_menu_save" />
33     </LinearLayout>
34
35     <EditText
36         android:id="@+id/websiteAddressEditText"
37         android:layout_width="wrap_content"
38         android:layout_height="wrap_content"
39         android:layout_gravity="fill_horizontal"
40         android:hint="@string/websiteAddressPrompt"
41         android:imeOptions="actionDone"
42         android:inputType="textUri">
43     </EditText>
44
45     <LinearLayout
46         android:layout_height="0dp"
47         android:layout_gravity="fill"
48         android:layout_marginTop="@dimen/activity_vertical_margin"
49         android:background="@android:color/holo_blue_light"
50         android:orientation="vertical"
51         android:paddingLeft="@dimen/activity_horizontal_margin"
52         android:paddingRight="@dimen/activity_horizontal_margin"
53         android:paddingTop="@dimen/activity_vertical_margin" >
54
55         <TextView
56             android:id="@+id/listTitleTextView"
57             android:layout_width="match_parent"
58             android:layout_height="wrap_content"
59             android:layout_gravity="fill_horizontal"
60             android:gravity="center_horizontal"
```

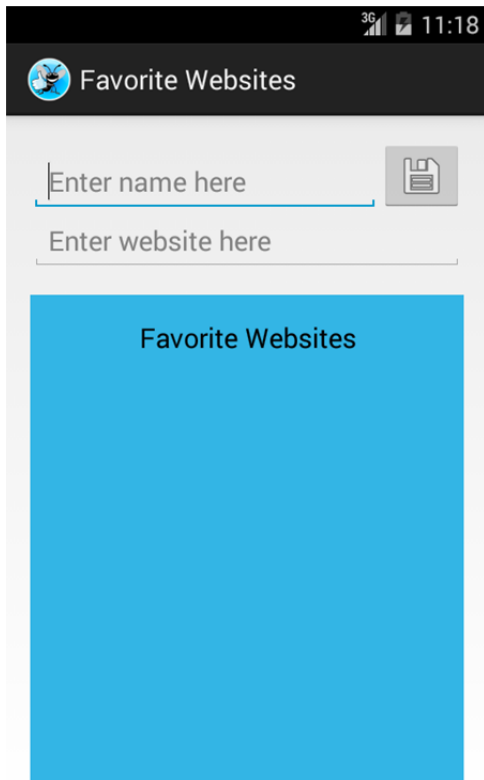


```
61         android:text="@string/favoritedWebsites"
62         android:textAppearance="?android:attr/textAppearanceMedium" />
63
64     <ListView
65         android:id="@android:id/list"
66         android:layout_width="match_parent"
67         android:layout_height="0dp"
68         android:layout_gravity="fill"
69         android:layout_marginBottom="@dimen/tagged_searches_padding"
70         android:layout_marginTop="@dimen/tagged_searches_padding"
71         android:layout_weight="1" />
72
73     </LinearLayout>
74
75 </GridLayout>
```

```
1 <TextView xmlns:android="http://schemas.android.com/apk/res/android"
2     android:id="@+id/textView"
3     android:layout_width="match_parent"
4     android:layout_height="?android:attr/listPreferredItemHeight"
5     android:gravity="center_vertical"
6     android:textAppearance="?android:attr/textAppearanceMedium" >
7
8 </TextView>
9
```

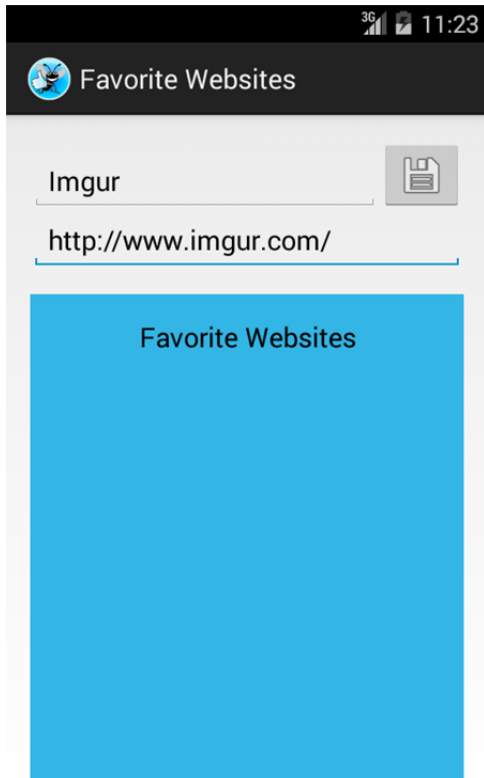
```
1  <?xml version="1.0" encoding="utf-8"?>
2  <resources>
3      <string name="app_name">Favorite Websites</string>
4      <string name="websiteNamePrompt">Enter name here</string>
5      <string name="websiteAddressPrompt">Enter website here</string>
6      <string name="favoritedWebsites">Favorite Websites</string>
7      <string name="saveDescription">Touch this button to save your website</string>
8      <string name="shareEditDeleteTitle">Share, Edit or Delete the website named \"%s\"</string>
9      <string-array name="dialog_items">
10         <item>Share</item>
11         <item>Edit</item>
12         <item>Delete</item>
13     </string-array>
14     <string name="shareSubject">Website that might interest you</string>
15     <string name="shareMessage">Check out this website: %s</string>
16     <string name="shareSearch">Share Search to:</string>
17     <string name="cancel">Cancel</string>
18     <string name="OK">OK</string>
19     <string name="confirmMessage">Are you sure you want to delete the website \"%s\"?</string>
20     <string name="delete">Delete</string>
21     <string name="missingMessage">Enter both a website and a name</string>
22 </resources>
23
```

Basic interface.



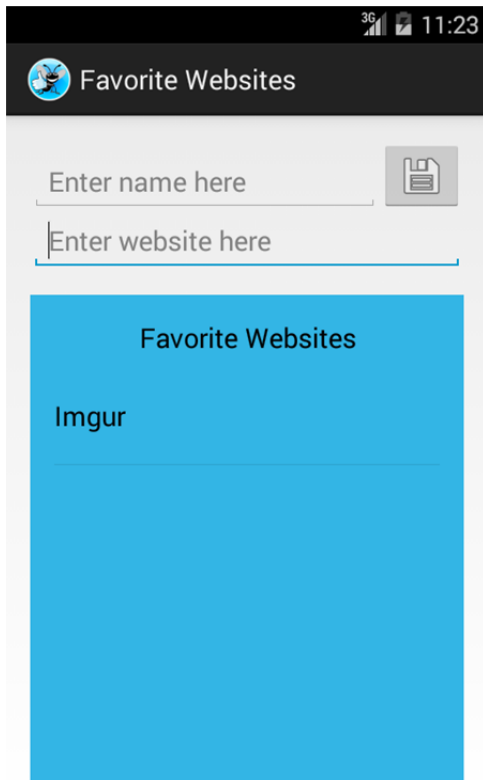
A screenshot of a mobile application titled "Favorite Websites". The app has a dark header bar with a circular icon on the left and the title "Favorite Websites" on the right. Below the header, there are two input fields: the first is labeled "Enter name here" and the second is labeled "Enter website here". To the right of the first input field is a small icon of a document with a plus sign. Below the input fields is a large blue rectangular area with the text "Favorite Websites" centered at the top.

Adding a website.

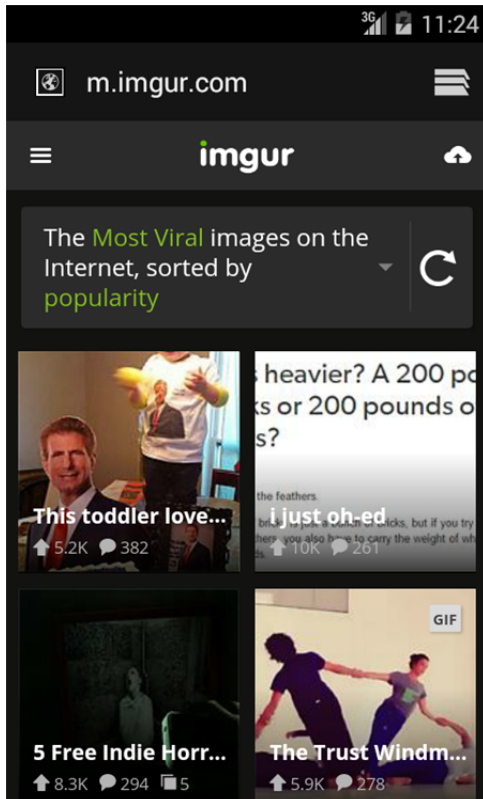


A screenshot of the same mobile application, now with data entered. The first input field contains the text "Imgur" and the second input field contains the URL "http://www.imgur.com/". The blue rectangular area below still contains the text "Favorite Websites".

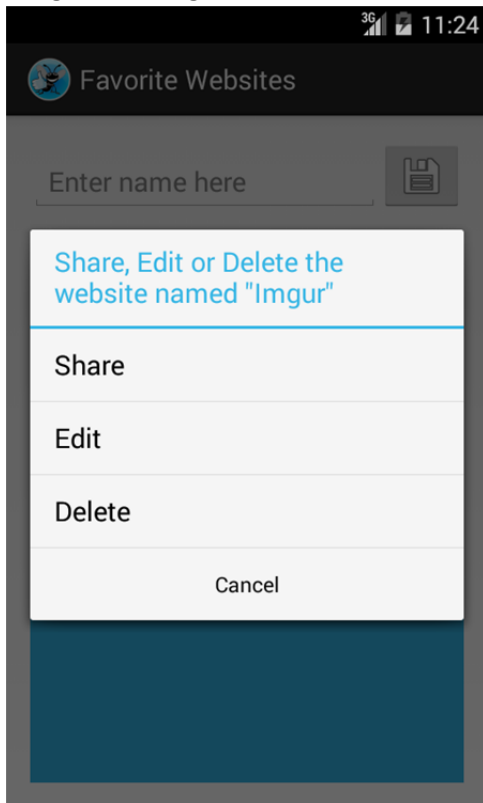
Website added.



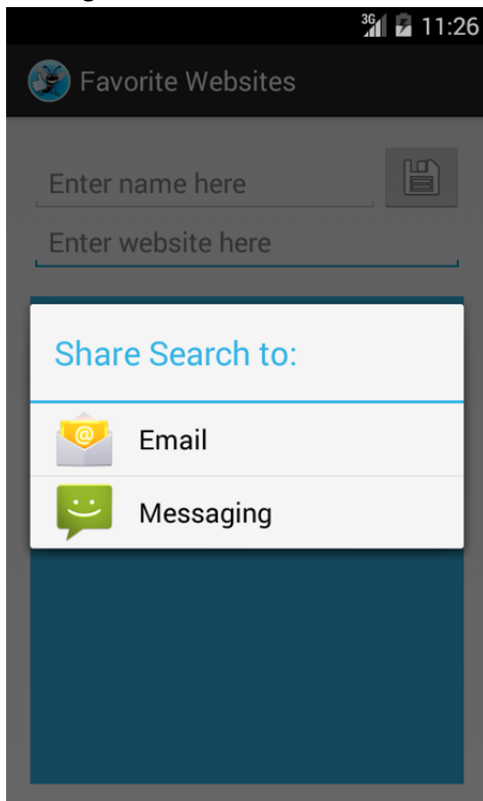
Tapping it, opens up the web browser.



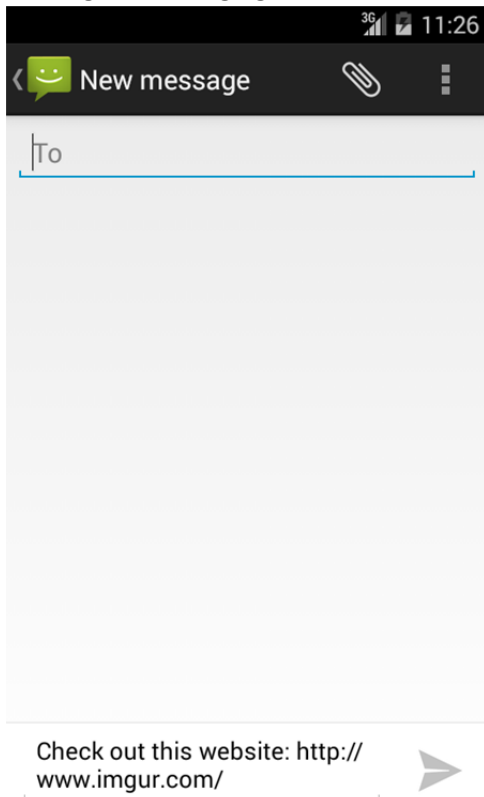
Long click dialog box.



Sharing.



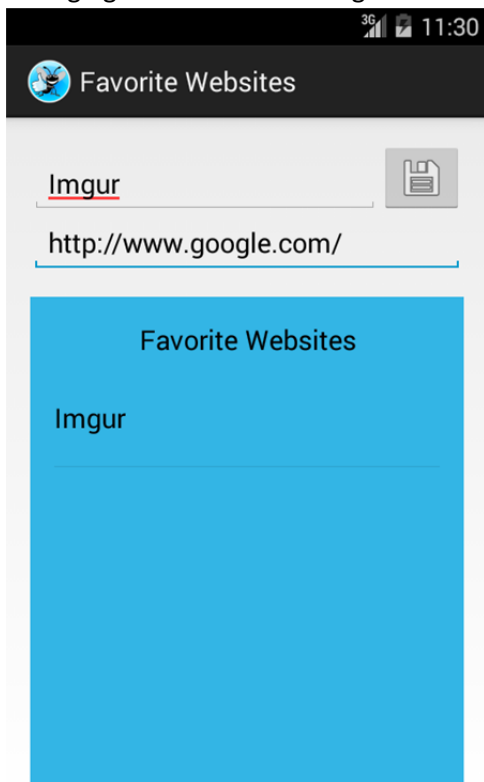
Sharing via messaging.



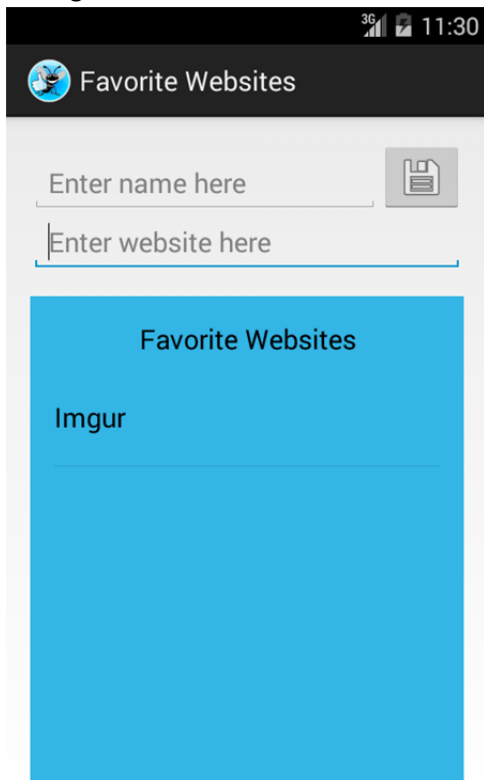
Editing the website.



Changing the website to Google.

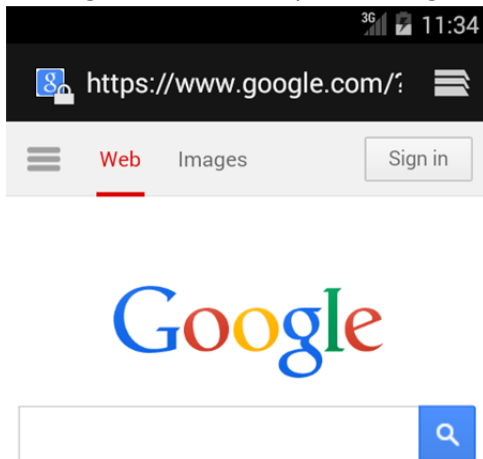


Saving the altered website.

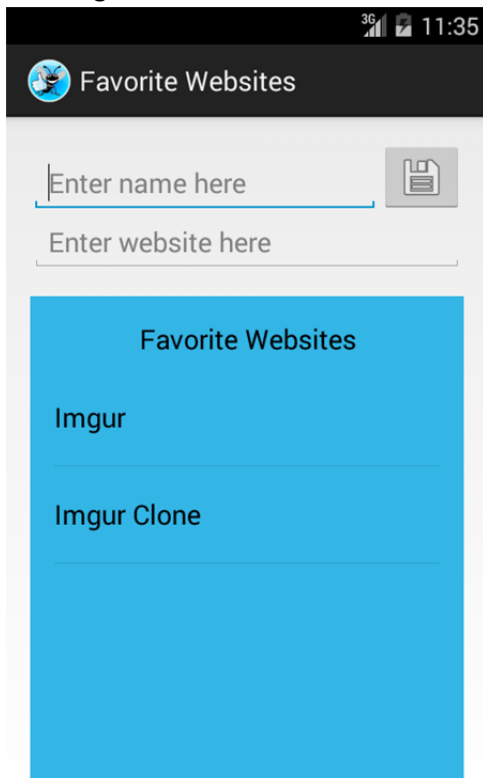




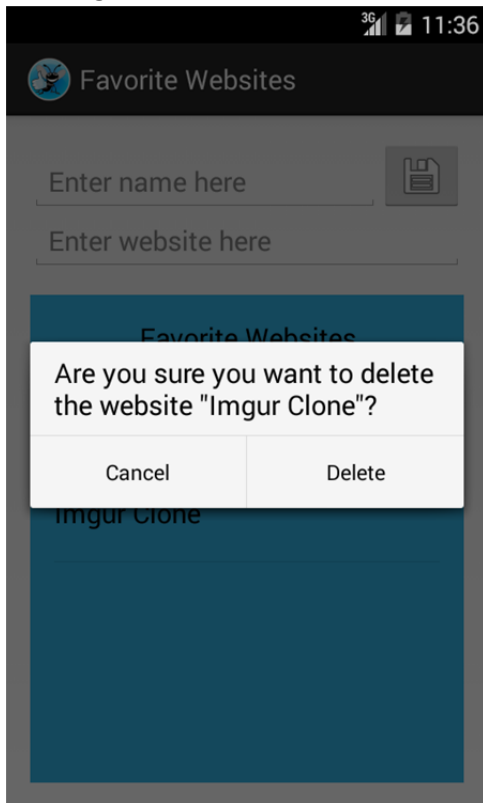
Clicking on it now takes you to Google.



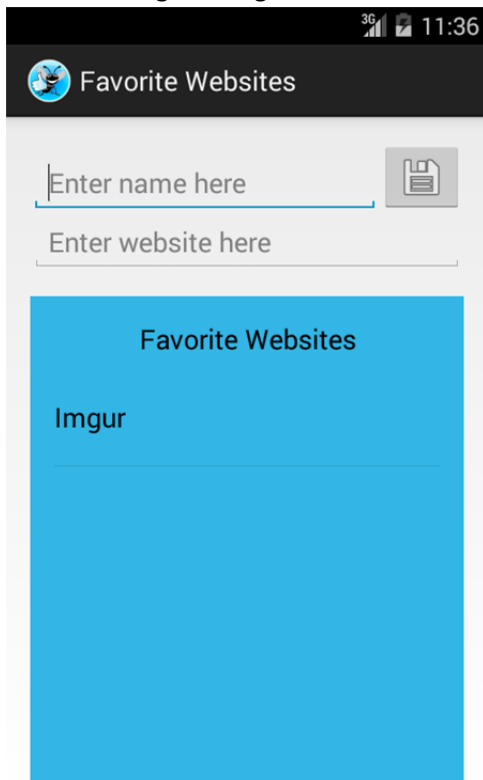
Creating a clone of the website.



Deleting the clone.



Back to a single listing.



Quitting the app and reopening it to make sure the site is saved across sessions.

