Bulletin of the *Transilvania* University of Braşov • Vol 4(53), No. 2 - 2011
Series III: Mathematics, Informatics, Physics, 123-128

MOBILE CLIENT FOR ONLINE DICTIONARY

Constantin Lucian ALDEA[1]

Abstract

The *Lexica dictionary* is an online bilingual dictionary [5]. It provides contexts, translations, paradigms and structures for a large number of English words used in the terminology associated with European Community legal texts [4].

In this paper a software architecture that extends the web application to the mobile phones such that a user can also access the online web application using its mobile device is presented. For the proposed architecture a client program is developed with the Google's Android software. *Mobile Lexica* is a software project that uses the same database as the website, but removes the need of a browser. Using the http protocol and the JSON (Javascript Object Notation) serialization format, it transfers data from the web service to the mobile, and backwards.

2000 *Mathematics Subject Classification:* 94A05, 94C30.
Key words: mobile devices, online dictionary, web access.

1 Indroduction

The *Lexica dictionary* is an online vocabulary [5]. It provides contexts, translations, paradigms and structures for a large number of English words. The user accesses the web site, enters a word and presses different search buttons to start a type of search (entire data, only paradigms, only translation etc.). After pressing the search button a request is sent to the web server that looks into the database for the proper data. The dictionary data was already added [4] and the purpose of this project is to present an Android client application project for the existing online dictionary.

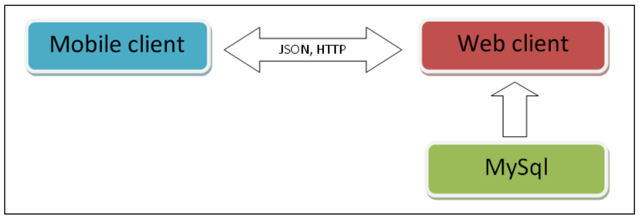


Figure 1: Lexica Web Architecture

[1]Faculty of Mathematics and Computer Science, *Transilvania* University of Braşov, Romania, e-mail: costel.aldea@unitbv.ro

The project extends the application to mobile phones and it results in an application architecture which is further implemented using Google's Android software. *Mobile Lexica* is software that uses the same database as the website, but removes the need of a browser. Using the http protocol and the JSON serialization format, it transfers data from the web service to the mobile, and backwards. As presented in figure 1 for the current project two components for the application were developed:

1. the first component which is server side and it queries the database and intermediates the request formulated on the mobile device interface. This component was developed using php programming and is hosted on the Apache server auxiliary to the old web application. The old application part keeps its own functionality.

2. the second component which is installed on the mobile device and assures the interface between the user and the vocabulary data through the first component. This project part was implemented on the mobile device using the Android platform.

2 Android Framework

Android programming is basically Java programming [3] and the developing platform uses a special type of application life cycle for the developed application [1].

Android is a software stack for mobile devices that includes an operating system, middleware and key applications. The Android SDK provides the tools and APIs necessary to begin developing applications on the Android platform using the Java programming language [6]. The main components of the android framework are the following:

1. View - Views are user interface (UI) elements that form the basic building blocks of a user interface. Views are hierarchical and they know how to draw themselves. A view could be a button, a label ,a text field or other kind of UI elements.

2. Activity - An activity is a user interface concept. An activity usually represents a single screen in the android application. It generally contains one or more views, but not necessarily. Moreover, other concepts in Android could better represent a viewless activity.

3. Intent - An intent generically defines an *intention* to do some work. Intents encapsulate several concepts. Intents can be used to perform the following tasks:

   (a) broadcast a message,

   (b) start a service,

   (c) launch an activity,

   (d) display a web page or a list of contacts,

   (e) dial a phone number or answer a phone call.

4. Content Provider - Data sharing among mobile applications on a device is common. Therefore, Android defines a standard mechanism for applications to share

data (such as a list of contacts) without exposing the underlying storage, structure, and implementation. Through content providers, data can be exposed to other applications.

5. Services in Android resemble services in Windows or other platformstheyre background processes that can potentially run for a long time. Android defines two types of services: local services and remote services.

   Local services are components that are only accessible through the application that is hosting the service. Conversely, remote services are services that are meant to be accessed remotely by other applications running on the device.

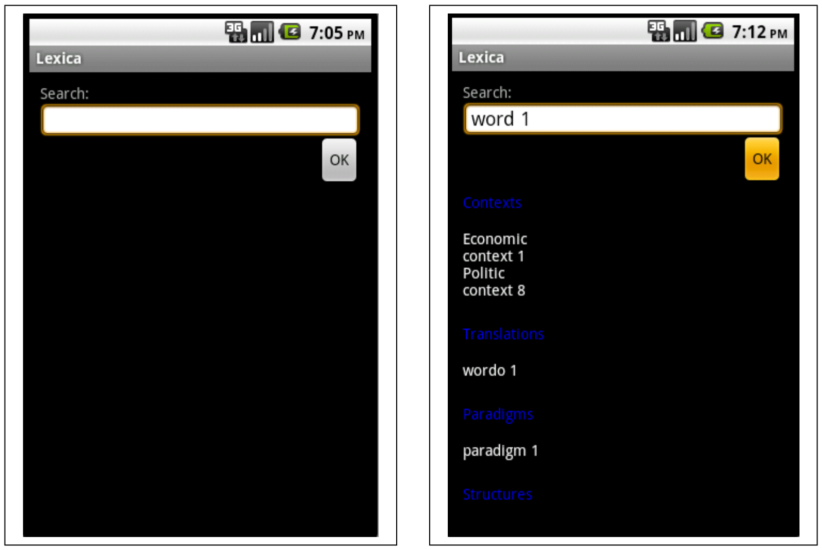
6. AndroidManifest.xml - is similar to the web.xml file in the J2EE world and defines the contents and behavior of the application. For example, it lists the applications activities and services, along with the permissions the application needs to run.

7. Android Virtual Devices - An Android Virtual Device (AVD) allows developers to test their applications without hooking up an actual Android phone. AVDs can be created in various configurations to emulate different types of real phones.

3 Mobile Lexica application use case

When the application is opened, the screen from figure 2 will appear. It contains a field, where the user can type the word he wants to search for, then by pressing the **OK** button, a request is made to the server.

The server will search in the database, and return the result, which will be displayed on the screen of the application.



word 1

Contexts

Economic
context 1
Politic
context 8

Translations

wordo 1

Paradigms

paradigm 1

Structures

Figure 2: Mobile application screens

It displays Contexts, Translations, Paradigms and Structures for the given word.

3.1 Application classes

As stated in the previous sections the proposed project consists of two parts which collaborate for solving the users' requests. In the following table the classes used by the project by keeping the correspondences between them with regards to the mobile dictionary functionality are presented.

Mobile client	Web Server	Description
Context.java	Context.php	A class that represents Context data.
Paradigm.java	Paradigm.php	A class that represents Paradigm data.
Structure.java	Structure.php	A class that represents Structure data.
Translation.java	Translation.php	A class that represents Translation data.
WordContext.java	WordContext.php	A class that represents Word Context data.
HttpAccess.java	-	A class that generates a http request.
Utils.java	-	Class used to convert data from JSON to Java Objects. It also creates requests to the Web Server.
Lexica.java	-	It is the main class of the mobile application. It creates the layout components.
-	Mysql.php	Class used by the web server to communicate to database.
-	Lexica.php	Receives requests from mobile clients and returns a result.

3.2 Application implementation

The main part of the application is the Lexica class:

```java
public class Lexica extends Activity {
  /** Called when the activity is first created. */
  @Override
  public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
  }
}
```

It extends the Activity class, which means that it is the component that runs when the application is started. Here, by overriding the onCreate method, we gain access to the `setContentView` method, with which we define the layout of the application.

The file *main.xml* contains the definition of the applications layout.

```xml
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:padding="10px"
    android:id="@+id/relativeLayout">
        <TextView android:id="@+id/label"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/search" />
    <EditText android:id="@+id/entry"
      android:layout_width="fill_parent"
      android:layout_height="wrap_content"
      android:background="@android:drawable/editbox_background"
      android:layout_below="@id/label" />
          <Button android:id="@+id/ok"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_below="@id/entry"
            android:layout_alignParentRight="true"
            android:layout_marginLeft="10px"
            android:text="OK" />
          <ScrollView android:id="@+id/scroll"
      android:layout_width="fill_parent"
     android:layout_height="wrap_content"
     android:layout_below="@id/ok">
    <LinearLayout android:id="@+id/description"
      android:orientation="vertical"
      android:layout_width="fill_parent"
      android:layout_height="wrap_content" />
        </ScrollView>
  </RelativeLayout>
```

The root element of the visual hierarchy is an instance of the RelativeLayout class, which lets child views specify their position relative to parent view or to each other (specified by ID). Elements are rendered in the order given, so if the first element is centered in the screen, other elements aligning themselves to that element will be aligned relative to screen center.

The TextView is used to display a label with the *Search* content.

EditText will take the user input, and after clicking the ok button, the LinearLayout component will be filled with the data retrieved from the Lexica server.

For each entity, there is a separate class: Context, Paradigm, Structure, Translation, and WordContext. All of them contain a method, which will convert the JSON string to the actual java object.

```
public static ArrayList<Context> getContextsFromJsonObject(String json) ↩
    throws JSONException {
    ArrayList<Context> result = new ArrayList<Context>();
    JSONArray jsonObj = null;

    try {
      jsonObj = (JSONArray) new JSONTokener(json).nextValue();
    }
    catch (Exception ex) {
      System.out.println(ex.getMessage());
      return null;
    }
    int size = jsonObj.length();
    int pos = 0;
    while (pos < size) {
      JSONObject o = (JSONObject) jsonObj.get(pos);
      Context c = new Context(o);
      result.add(c);
      pos ++;
    }
    return result;
  }
```

Which uses the following constructor:

```
public Context(JSONObject obj) throws JSONException
  {
    setIdContext(obj.getInt("idContext"));
    setContext(obj.getString("context"));
    setFile(obj.getString("file"));
  }
```

Communication with the Lexica service is done by using the Utils class, which makes a http post/get request.

```
HttpGet httpGet = new HttpGet(url);
  HttpClient httpclient = new DefaultHttpClient();
  // Execute HTTP Get Request
  HttpResponse response = httpclient.execute(httpGet);
  content = response.getEntity().getContent();
```

4 Conclusion

The electronic society introduces new forms of communication [2]. As regards mobile devices culture imposes new models for web applications.

This paper presents a model which can be further used for transferring the information and the computability power on to mobile devices and at the same time keeping the existing functionalities for the legacy applications.

References

[1] Wei-Meng, L., *Beginning Android Application Development*, Wiley Publishing, Inc., Indianapolis, 2011.

[2] Aldea, C., L., *Elemente de securitate în reţele de calculatoare*, *Transilvania* University Publishing House, Braşov, 2010.

[3] Sângeorzan L., Aldea, C., Dumitru, M., *Java. Aplicaţii*, Infomarket Publishing House, Braşov, 2001.

[4] Sangeorzan, L., Burada, M., Kiss-Iakab, K., *Designing a Text Parsing Programme for a Specialized Bilingual Online Dictionary*, Recent Advances in Computer Engineering, Proc. of the 8^{TH} WSEAS International conf. on applied informatics and communications, Pts I and II, Greece, Aug. 20-22, 2008, 110-114.

[5] http://cerex.unitbv.ro/lexica/index.php

[6] http://developer.android.com/sdk/index.html