

DESIGNING AN M-LEARNING APPLICATION FOR A UBIQUITOUS LEARNING ENVIRONMENT IN THE ANDROID BASED MOBILE DEVICES USING WEB SERVICES

SHANMUGAPRIYA M

Research Scholar

Mother Teresa Women's University, Kodaikanal, Tamilnadu, India

Dr TAMILARASI A

Professor & Head

Department of Computer Applications

Kongu Engineering College, Perundurai, Tamilnadu, India

Abstract

Educational Technology is constantly evolving and growing, and this progression will continually offer new and interesting advances in our learning environment. Traditional E-Learning systems developed for laptop and desktop computers were based on stand-alone software application or through websites and lack the ability to provide a comprehensive ubiquitous learning environment. A ubiquitous learning environment based on early days mobile phones lack the processing power of notebooks or desktop computers, low data transfer speeds and capacity. However, the ability to provide a comprehensive ubiquitous learning environment on the 3G (3rd Generation) mobile device will offer powerful collaborative and interactive learning opportunities. Thus the main objective of the research work is to develop an interactive mobile learning application based on Web Services in Android base mobile devices to facilitate the ubiquitous learning. This paper deals with the prototype development of an M-Learning application for mobile phones running with Android platform using Web services.

Keywords: Ubiquitous Learning Environment, Web Services; Mobile learning, Android OS, 3G Mobile

1.0 Introduction

A ubiquitous learning environment is any setting in which students can become totally immersed in the learning process. Since the mobile devices support the anytime, anywhere learning, m-learning can foster the growth of the ubiquitous learning (u-learning). M-learning application framework enables the learner to access the learning object and interact with the instructor and other learner seamlessly from the tablet PCs while in class, from his mobile phone during travelling or from his laptop when at home. U-learning provides the learner the freedom from learning environments, learning devices and learning content format and rather emphasize on the constructivist learning process and cognitive development among learners. Using portable computing devices (such as laptops, tablet PCs, PDAs, and smart phones) with wireless networks enables mobility and mobile learning, allowing teaching and learning to extend to spaces beyond the traditional classroom. The evolution of today's mobile devices increases the number of mobile applications developed, and among them the mobile learning applications. Recent 3G mobile hardware and software platforms allow running of faster and richer applications. Web services facilitate the ubiquitous learning environment by providing the unified interaction irrespective of the request originates from a Portable computer, a mobile phone or from a tablet PCs.

2.0 M-Learning Application Framework for Ubiquitous Learning Environment (ULE)

Mobile learning (m-learning) applications which supports the u-learning are developed using wide variety of languages, platforms and technologies. Each application developed has their own implementation strategies and specific features in terms of user interface, targeted mobile devices and interoperability features. The mobile application development depends mainly on these characteristics and influenced by their underlying platform for running these applications. Typically, a mobile learning system requires a Mobile Learning device, an m-learning application and m-learning content. The m-learning application framework helps to run diverse mobile-learning content. The m-learning content may be of simple text type to complex multi-media content with image, audio and video file formats. The m-learning application that runs on mobile device may be a dedicated stand-alone application, a client – server model or mobile Web browser based application. Android, an open-source mobile operating system and application framework supported by Google is a recent addition in the mobile application platform. By providing an open development platform, Android offers developers the ability to build extremely rich and innovative applications with a rich set of User Interfaces, support for broad range audio and video file formats. Developers and mobile handset manufacturers around the world has embraced this new platform due its opens-source and diverse application development and running capabilities with programming language as Java.

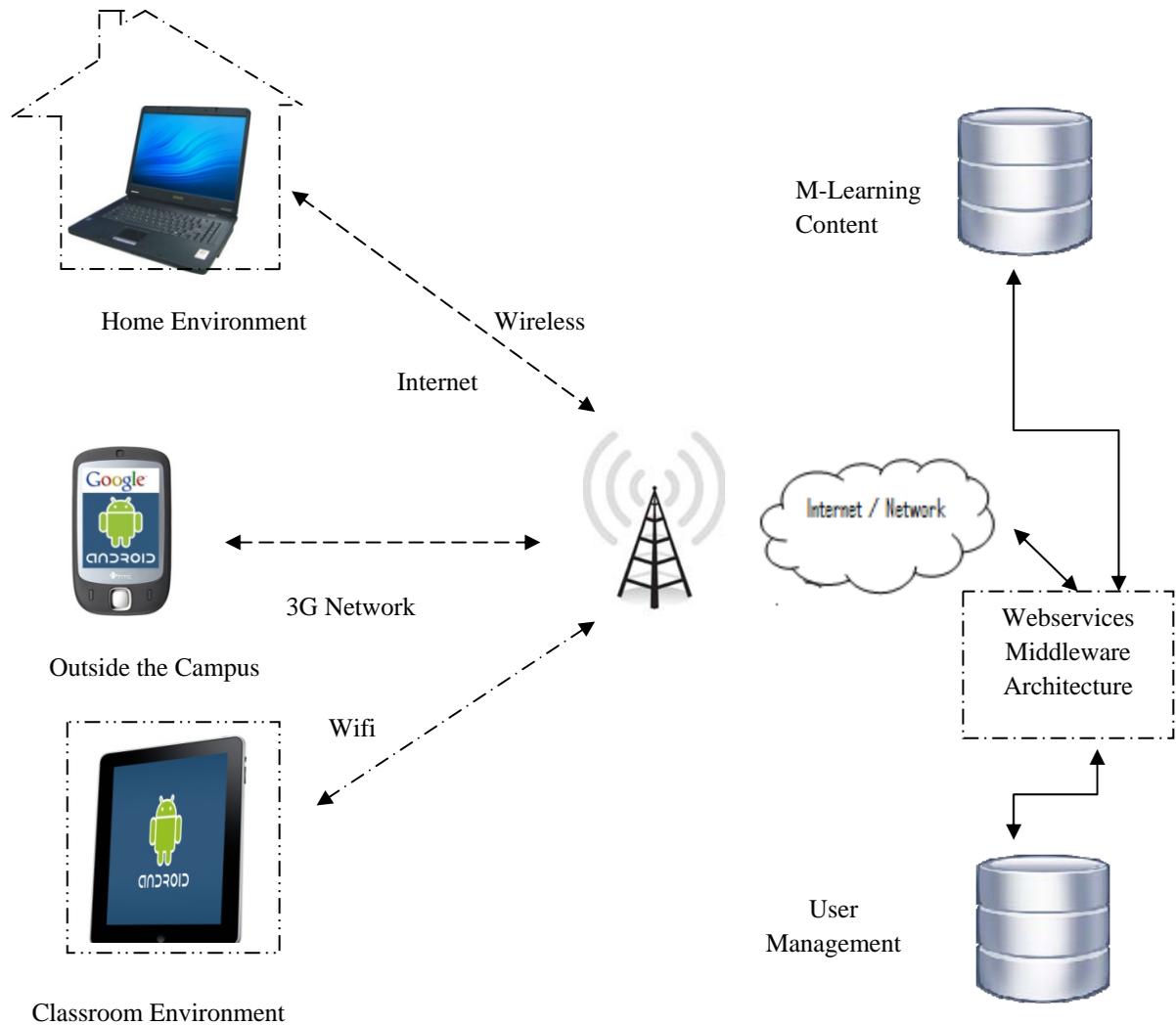


Fig 1. M-Learning Application for Ubiquitous Learning Environment

The Ubiquitous Mobile Learning Architecture is depicted in the figure 1. The three main factors in this design are the 'where', 'what' and the 'how'. The 'where' is the learning environment which signifies the anytime, anywhere learning be in the traditional classroom setting or outside the classroom like travelling in a bus or being at home and uses a wireless network with both 3G and WiFi to connect with the University server at all time using Web Services architecture. The 'what' is the inclusion of pedagogical information which is based on constructivist theory, allowing students to create knowledge from what they see, hear, read and perceive. The 'how' denotes an interactive learning application framework which have both client module running on mobile devices with different form factor suitable for varying learning environments like Tablet PCs for classroom learning, Mobile Phones for anywhere learning outside the classroom and Notebook PCs to be used at home environment and linked with the server module which runs on the University Campus Server through Web services middleware architecture. Students using the ULE will intuitively interpret their surroundings and construct their own knowledge.

3.0 M-learning Application

An m-learning application can be designed as a dedicated stand-alone application that can run on individual mobile device, a client-server model which requires a client application running on mobile device and a server application runs in a remote-server or on mobile web browser that requires back-end application-server connectivity while sending requests from the mobile device [Metcalf II D. S et.al(2006)]. The roles of different users in the m-learning applications designed for ubiquitous learning environment based on the Web Services architecture discussed as follows:

The roles of a *trainer/instructor* in the m-learning are to

- Design and develop the content for a course
- Prepare tests, Quizzes and assignments for students/learners evaluation
- Assess the tests, assignments and projects submitted by the students/learners
- Send Feedback on the assignments and projects submitted by the students/learners
- Interacting with students/learners using communication tool provided in the learning management system

The roles of the *learner/students* in the m-learning are to

- Take the course
- Take Tests on the learnt content
- Work on Assignments and Projects related to the course
- Send Feedback and Queries Related to the course
- Interact with the trainer and fellow learner/student concerned with the course

The roles of the *Application Developer* in the m-learning are to

- Develop a framework where the content designed by the trainer can be presented in a pre-formatted manner to the learner when they launch the application
- Provide Pre-defined templates for framing the tests and assignments
- Design the Pervasive User Interfaces for accessing the content and tests related to that course
- Provide the communication platform for trainer – learner interaction messaging from the application itself

Android Platform provides the necessary tools and packages for designing the desired m-learning application for mobile devices in the u-learning environment.

4.0 Android Platform

4.1 Android Operating System

Android operating system is a project initiated by Google through the Open Handset Alliance, which includes over 30 companies in ICT. Android platform is an open source project, allowing its amendment by any manufacturer of mobile devices [Meier R.(2010)].

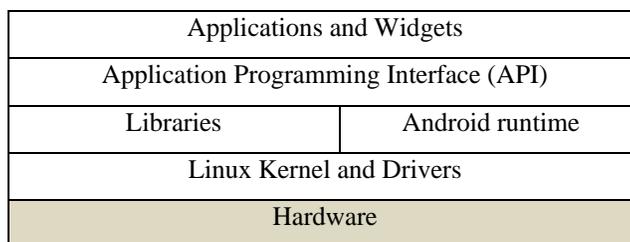


Fig 2. Android Architecture

Figure 2 depicts the architecture of Android operating system.

The operating system is based on *Linux kernel* version 2.6.x, that is a monolithic kernel. The kernel includes *drivers* for the mobile device hardware: screen, keyboard, camera, USB, Bluetooth etc. Kernel provides inter-face hardware and memory management, processes and other resources.

Native *libraries* on the next level are dependent on hardware architecture of the mobile device. These libraries include support for 2D and 3D graphics (Single, OpenGL ES), multimedia, security, storage, browsing (WebKit) and standard C library.

Android applications are developed using Java programming language. Applications require an *environment* to manage their life-cycle. This includes a Java virtual machine (called Dalvik virtual machine) and Java class

libraries that provide basic support for applications (collections, input/output operations management etc.). Android applications are not compatible with Java ME or Java SE. The applications are optimized for mobile devices constraints.

The *application programming interface* allows accessing a framework that includes components used by all Android applications. The application framework includes components for Android application management (installation, execution), windows management and user interface graphical, event handling etc.

Application level includes pre-installed applications (contact management, phone, calendar, Internet browser) and user applications. Applications are based on Java technologies and use classes provided through application programming interface. In addition, there is the possibility to use native functions in programs written in C/C++ programming language through NDK (Native Development Kit).

Widgets unlike applications, occupies only part of the main display screen and associated (Home). Android operating system is multitasking, each application running in a separate thread.

4.2 Android Application Development

Android applications are developed using one or more basic components [Burnette, E. (2009)]:

- activities (**Activity** base class),
- services (base class **Services**),
- content providers (**ContentProvider** base class)
- components that receive and act on messages sent to all applications (the base class **BroadcastReceiver**)
- messages (class **Intent**)

A particular importance in application development is the resources that enable separation of interface code. Activities represent the screen associated to an application. An application can have one or more activities.

Services are routines that run in parallel with the main thread and do not have GUI. They allow the development of actions in the background without blocking the main thread execution and interaction with such application.

Content providers are used for sharing data between applications. Data sharing is done through files, databases or other means. An alternative to content providers is the use of communication between processes.

Applications can respond to the occurrence of events in the system by using classes derived from **BroadcastReceiver**. They do not GUI and an application can have several components of this type.

In order to activate components like activities, asynchronous messages encapsulated in objects of **Intent** type are used.

Android applications are developed mainly using Eclipse IDE with Android Development Tools (ADT) plug-in. Android SDK and emulators are necessary for application development.

4.3 Graphical User Interface

The prototype m-learning application developed consists of three activities:

- Module Selection;
- Course Selection;
- Quiz Question & Answer Choices

The user/learner interface is simple and intuitive, without much graphics and drawings. It will reduce the amount of memory required by the application and reduces the time of development. The original version has been designed with interactive and user-friendly icons and menus. The application targets the mobile devices with touch screen, but can be accessible by any kind of device running Android platform.

User Logs in to the application environment and has not been shown in fig. The user is authenticated by the ULE server which stores the unique profile for each learner. Upon logging to the application the user profile is synced with the type of device that he uses. After successful login the main screen for module selection is presented in Fig 3. The associated activity is *Mlearn* and is declared in *Androidmanifest.xml* configuration file. The main screen shows the list of choices for the learner to choose his learning activity like Course Selection, Test Selection or Assignment Selection. The list is an object of **ListView** type.

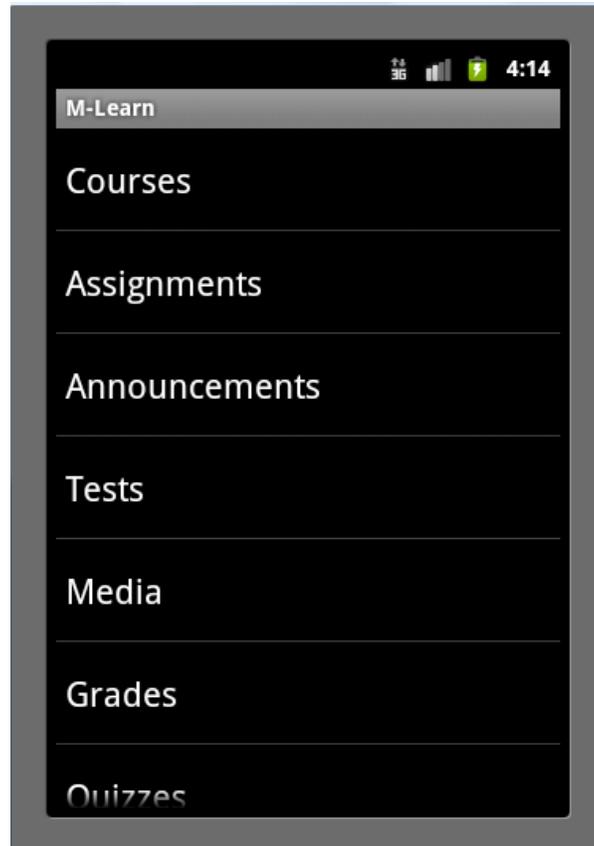


Fig 3. Module Selection Screen displayed on an Android Device

```
public void onItemClick(ListView parent, View v, int position, long id)
{
    .....
    .....
    //launch selected module activity list
    startActivity(intent);
}
}
```

When the user chooses the appropriate choice the next screen based on another activity is launched. In this case the user has chosen the Course List and the following Activity is launched.



Fig 4. Course List Screen Shown on an Android Device

The figure shown above displays the list of courses that the user can select for learning.

The description of list elements is defined in a resource file based on XML. The following structure shows the content of the XML file.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content" android:layout_height="wrap_content">
    <ImageView android:id="@+id/icon" android:layout_height="wrap_content"
    android:src="@drawable/icon" android:layout_width="22px" android:layout_marginTop="4px"
    android:layout_marginRight="4px" android:layout_marginLeft="4px">
        </ImageView>
    <TextView android:text="@+id/TextView01" android:layout_width="wrap_content"
    android:layout_height="wrap_content" android:id="@+id/label" android:textSize="30px"></TextView>
</LinearLayout>
```

In order to start a new activity, an Intent instance is used. All activities need to be declared in *AndroidManifest.xml* file:

```
<activity android:name=".Mlearn" />
<activity android:name=".CourseList"/>
<activity android:name=".QuizChoice"/>
```

If the activities are not declared in *Android-Manifest.xml*, a runtime exception will occur when the code is executed.

All the available courses / tests / assignments etc are loaded from a database stored on the server side. A dedicated Web service method is used for this initial action.

The main menu includes items that allow to:

- Login/Logout the application;
- Browse the Learning Content Modules;
- Syncing with the Server;
- Exit the application;

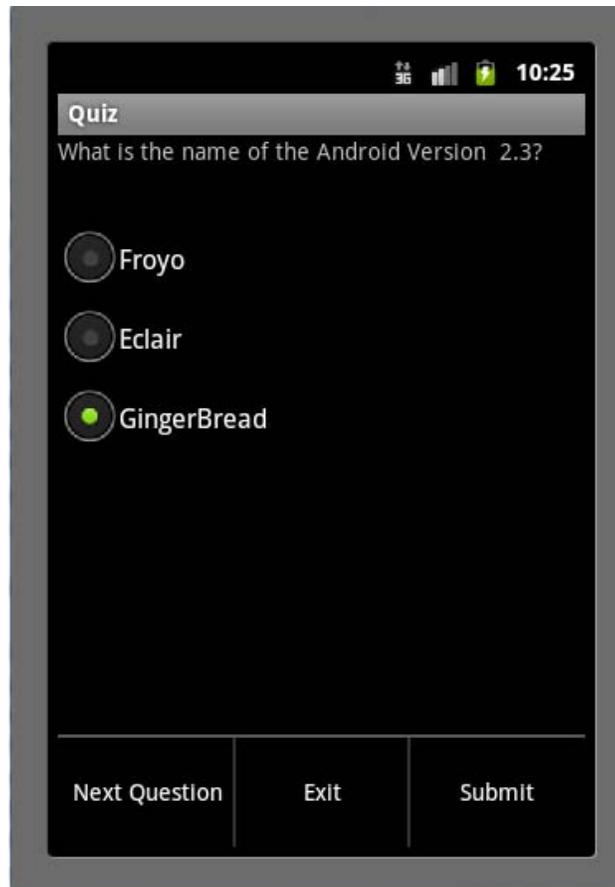


Fig 5 Quiz Question and Answer Choice Screen

Fig 5 depicts the Quiz Question and Answer format. On Each submission menu will be provided relevant to the screen to navigate to the next question, quit the application or submit the answer. The answer submission will be stored in the database server under each user profile for evaluation by the instructor. The submission of answer to the server is achieved through Web Services.

The Web service used by this application has been developed using Microsoft .NET technologies. The Web service provides several Web methods that are used by this mobile learning application. The use of Web services provide a very flexible way for client applications that can be developed on almost any mobile or desktop platform.

The **QuizImpl** method returns the number of questions of a given quiz identified through a code. The number of questions is taken from the database. If an exception occurs, the method will return a negative value.

The Web services are consumed using a third-party library distributed as free source kSOAP2 (<http://ksoap2.sourceforge.net/>), optimized for Android.

The libraries need to be added to the project in order to be used. This library is based on the SOAP architecture and there is no need to generate a proxy/stub to call Web services methods.

A SOAP envelope is created using the **Soap-SerializationEnvelope** class (specifying the SOAP version) and the request details are added to the envelope body (using **SoapObject** class).

The **HttpTransportSE** class is used to make the actual call of the Web service method, the envelope being passed as parameter. The result is retrieving from the response part of the envelope.

The following code is an example of a method used in this Web service:

```
import android.app.Activity;
import android.os.Bundle
.....

public class Quiz extends Activity {

    private static final String NAMESPACE = "..... "
    private static final String URL = "http://..... "
    private static final String SOAP_ACTION = "QuizImpl";
    private static final String METHOD_NAME = "QuizResponse";
    .....
    .....
};

@Override
Public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
    .....

    SoapObject request = new SoapObject(NAMESPACE, METHOD_NAME);
    SoapSerializationEnvelope envelope =
    new SoapSerializationEnvelope(SoapEnvelope.VER11);

    envelope.setOutputSoapObject(request);
    HttpTransportSE androidHttpTransport = new HttpTransportSE(URL);

    try {
        androidHttpTransport.call(SOAP_ACTION, envelope);
        SoapObject resultsRequestSOAP = (SoapObject) envelope.bodyIn;
        .....
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

It is very important to send the right parameters to the methods. In this example, the parameters used in these calls are initialized as follows:

- **NAMESPACE** = "http://tempuri.org"; – Web service namespace; in this example the default namespace is used
- **URL_WS** = "http://servername/WebService.asmx"; – Web service URL
- **GET_QUIZIMPL** = "GetQuizImpl"; – Web service method name

In order to use Web services the application needs special permission. Permissions are stored in *AndroidManifest.xml* file. For this mobile application, the following line has to be added:

```
<uses-permission android:name="android.permission.INTERNET">
</uses-permission>
```

Each activity in our m-learning application has been designed to invoke a web service method which returns the uniform object request accessed in a different learning environment. This architecture allows the scalability of the application and supports wide variety of display formats. The user profile is always synchronized with the server and eases the learner to concentrate on learning than with the device settings etc.

5.0 Conclusions and Future Work

The ubiquitous learning environment (ULE) evolves more on context data than eLearning. Besides the domains of eLearning, uLearning may use more context awareness to provide most adaptive contents for learners. M-learning application based on 3G mobile devices nicely fits into uLearning paradigm. Education is happening all around the student but the student may not even be conscious of the learning process in the ULE. A wide range of platforms are available to choose to develop m-learning applications. The Ubiquitous Learning environment emphasizes the learner-centric curriculum and content. Among the wide range of choices the open-source nature and rich user interface that even supports sensor hardware and multitasking makes Android a desired platform for designing m-learning application.

Regarding the presented mobile application, the following development directions are considered in order to provide a basis for a working m-learning application.

- new functionalities,
- interface improvement,
- server interaction,
- user management,
- modules integration,

This prototype is a part of an m-learning system that will also be developed for other mobile platforms in order to cover a wide area of mobile devices and users.

References

- [1] Alier, M., Casany, M., & Casado, P. (2007). A Mobile extension to a web based Moodle virtual classroom. In P.Cunningham & M. Cunningham (Ed.), *Expanding the Knowledge Economy: Issues, Applications, Case Studies*:Vol. 4 (pp. 1169-1176). IOS Press
- [2] Boja, C. L Batagan, (2009) "Analysis of M-Learning Applications Quality," *WSEAS TRANSACTIONS on COMPUTERS*, Issue 4, Vol. 8, ISSN 1109-2750, pp. 767-777.
- [3] Burnette, E. (2009) *Hello, Android: Introducing Google's Mobile Development Platform, 2nd Edition*, The Pragmatic Bookshelf.
- [4]Goole Inc. ANDROID developers User Interface. Last checked on January 10, 2011, from <http://developer.android.com/guide/topics/ui/index.html>.
- [5]Hansen, F. and Bouvin, N., (2009). "Mobile Learning in Context — Context aware Hypermedia in the Wild", *International Journal of Interactive Mobile Technologies (IJIM)*, Vol 3, No 1.
- [6] Meier R.(2010), *Professional Android 2 Application Development*, Wiley Publishing, Inc.,
- [7] Metcalf II D. S. and De Marco J. M., (2006) *mLearning: Mobile Learning and Performance in the Palm of Your Hand*, HRD Press, Inc.,