

T.C.
Bahçeşehir Üniversitesi

**MOBILE LEARNING BASED ON ADAPTIVE WEB
INTERFACE**

Musa KARAKELLE

Master Thesis

Istanbul, 2010

T.C.
Bahçeşehir Üniversitesi

Institute Sciences
Computer Engineering Graduate Program

**MOBILE LEARNING BASED ON ADAPTIVE WEB
INTERFACE**

Musa KARAKELLE

Master Thesis

Supervisor
Assoc.Prof. Adem KARAHOCA

Istanbul, 2010

T.C
BAHÇEŞEHİR ÜNİVERSİTESİ
The Graduate School of Natural and Applied Science
Computer Engineering Graduate Program

Title of the Master's Thesis : **Mobile Learning Based on Adaptive Web Interface**

Name/Last Name of the Student : Musa KARAKELLE

Date of Thesis Defense : 14.06.2010

The thesis has been approved by the Institute of Science.

Signature

Assist. Prof. F. Tunç BOZBURA

Director

This is to certify that we have read this thesis and that we find it fully adequate in scope, quality and content, as a thesis for the degree of Master of Science.

Examining Committee Members:

Assoc.Prof. Adem KARAHOCA (Supervisor)

Assist. Prof. Yalçın ÇEKİÇ

Assist. Prof. Mehmet Alper TUNGA

ABSTRACT

MOBILE LEARNING BASED ON ADAPTIVE WEB INTERFACE

KARAKELLE, Musa

Computer Engineering Graduate Program

Supervisor: Assoc.Prof. Adem KARAHOCA

05 June 2010

Web applications automatically adapted for different clients and user preferences gain more importance. Still, there are barely technologies to compensate the additional effort of creating, maintaining and publishing such Web content. To address this problem, this thesis introduces a declarative, Model View Controller approach for adaptive, dynamic view on the basis of Browser Definition File Schema (XML-technology). Adaptive Web interface on different abstraction levels are defined in order to support effective Web view authoring and generation. Most of the learning content existing today is designed for high resolution screen. Only a fraction of contents can be utilized on a mobile device. Although some content may be specifically designed for mobile devices, content providers cannot assure that this content can be accessed by all mobile devices, because of the wide variety of technical characteristics, and existing standards. In this thesis, we propose a device-independent architecture for mobile learning, which provides contents based on characteristics of mobile devices and mobile learners. Delivering contents tended to adapt to not only learner's needs and preferences, but also to mobile device used.

CONTENTS

ABSTRACT	I
CONTENTS	II
LIST OF FIGURES.....	IV
1. INTRODUCTION	1
1.1. INTRODUCTION.....	1
2. METHODOLOGIES	3
2.1. MODEL VIEW CONTROLLER	3
2.2. ORTHOGONALITY	4
2.3. THE DRY PRINCIPLE.....	4
2.4. AGILE SOFTWARE DEVELOPMENT	5
2.5. AGILE MANIFESTO (HTTP://WWW.AGILEMANIFESTO.ORG/.....	5
3. OPEN API'S AND MODELS FOR WEB 2.0 TECHNOLOGIES.....	5
3.1. EXTENSIBLE MARKUP LANGUAGE.....	5
3.2. REPRESENTATIONAL STATE TRANSFER	6
3.3. RESTFUL WEB SERVICES CHARACTERISTICS.....	8
3.4. SOAP	8
3.5. WEB SERVICE DESCRIPTION LANGUAGE	9
4. PRESENTATION TIER (UI).....	HATA! YER İŞARETİ TANIMLANMAMIŞ.
4.1. EXAMPLE AJAX PRESENTATION LAYER DATAFLOW.....	HATA! YER İŞARETİ TANIMLANMAMIŞ.
4.2. AJAX	11
5. RUBY ON RAILS	12
5.1. EXAMPLE APPLICATION LAYER DATA FLOW	12
5.2. RUBY ON RAILS.....	14
6. MOBILE AND ASP.NET MVC	15

6.1. MOBILE BROWSER DEFINITION FILE	15
6.1.1. Here's a simple example to start.....	17
6.2. IPHONE USER INTERFACE FRAMEWORK	17
6.2.1. Link in iUI javascript and css files.....	20
6.2.2. Define toolbar and main menu	20
6.2.3. Define any sub-menus	21
6.2.4. Define any panels/dialogs	Hata! Yer işareti tanımlanmamış.
6.2.5. Complex panels	Hata! Yer işareti tanımlanmamış.
6.2.6. Striped tables	Hata! Yer işareti tanımlanmamış.
7. ENTITY FRAMEWORK	25
7.1. ACCESSING AND CHANGING ENTITY DATA	25
8. MOBILE LEARNING ADAPTIVE WEBSITE OUTLINE	27
8.1. PROJECT MODULES	27
8.1.1. Admin Module	27
8.1.2. User Module.....	27
8.2. WINDOWS MOBILE AND IPHONE.....	29
9. M-LEARNING ADAPTIVE WEB INTERFACES	33
9.1. INSTRUCTOR UI.....	33
9.2. STUDENT UI ON WINDOWS MOBILE, IPHONE AND PC.....	44
CONCLUSION	52
REFERENCES.....	HATA! YER İŞARETİ TANIMLANMAMIŞ.

LIST OF FIGURES

Figure 2.2.1.1 Model View Controller	3
Figure 4.1.1 Ajax presentation layer	10
Figure 5.1.1 Application layer data flow	13
Figure 6.1.1 Mobile device browser file	15
Figure 6.1.2 Views containing folders	16
Figure 6.1.3 Mobile views	17
Figure 6.2.1 Toolbar main menu	Hata! Yer işareti tanımlanmamış.
Figure 6.2.2 Sub-menus	Hata! Yer işareti tanımlanmamış.
Figure 6.2.3 Sub-menus 2	Hata! Yer işareti tanımlanmamış.
Figure 6.2.4 Sub-menus 3	Hata! Yer işareti tanımlanmamış.
Figure 6.2.5 Panel	Hata! Yer işareti tanımlanmamış.
Figure 6.2.6 Search panel	Hata! Yer işareti tanımlanmamış.
Figure 6.2.7 Login panel	Hata! Yer işareti tanımlanmamış.
Figure 6.2.8 Complex panel	Hata! Yer işareti tanımlanmamış.
Figure 6.2.9 Striped panel	Hata! Yer işareti tanımlanmamış.
Figure 7.3.1 Entity framework	26
Figure 8.1.1 Mobile learning ER-diagram	29
Figure 8.2.1 Computer view	30
Figure 8.2.2 Windows mobile view	31
Figure 8.2.3 iPhone mobile view	32
Figure 9.1.1 Instructor Home	33
Figure 9.1.2 Instructor Administrator Tools	33
Figure 9.1.3 Student List for Enrolment	34
Figure 9.1.4 Enroll & Drop Course	34
Figure 9.1.5 Course List & Operations	35

Figure 9.1.6 Exam Adding Form.....	35
Figure 9.1.1 Exam Listing & Operations	36
Figure 9.1.2 Exams's eveluating status.....	36
Figure 9.1.3 Eveluating exams	37
Figure 9.1.4 Create clasical questions	38
Figure 9.1.5 Create true / false questions	39
Figure 9.1.6 Create multiple chooses questions	40
Figure 9.1.7 User list & operations	41
Figure 9.1.8 Account details.....	41
Figure 9.1.9 Account roles	42
Figure 9.1.10 Content upload panel	42
Figure 9.1.11 Content Category	43
Figure 9.1.12 Content list	43
Figure 9.2.1 Student's home page on pc.....	44
Figure 9.2.2 Student's home page on windows mobile	44
Figure 9.2.3 Student's home page on iPhone.....	44
Figure 9.2.4 Log On screen on pc	45
Figure 9.2.5 Log On screen on Windows Mobile	45
Figure 9.2.6 Log On screen on iPhone	45
Figure 9.2.7 Wellcome screen on pc	46
Figure 9.2.8 Wellcome screen on Windows Mobile	46
Figure 9.2.9 Wellcome screen on iPhone	46
Figure 9.2.10 Course list on pc.....	47
Figure 9.2.11 Course list on Windows Mobile	47
Figure 9.2.12 Course list on iPhone	47
Figure 9.2.13 Course list on Windows Mobile	48

Figure 9.2.14 Course list on iPhone	48
Figure 9.2.15 Finished exam on pc	49
Figure 9.2.16 Finished exam on Windows Mobile	49
Figure 9.2.17 Finished exam on iPhone	49
Figure 9.2.18 Exam list on pc.....	50
Figure 9.2.19 Exam list on Windows Mobile	50
Figure 9.2.20 Exam list on iPhone	50
Figure 9.2.21 Exam on pc	51
Figure 9.2.22 Exam on Windows Mobile	51
Figure 9.2.23 Exam on iPhone	51

1. INTRODUCTION

1.1. INTRODUCTION

Economies at the present day, we meet new generation education and development model in our country as being in the world. The unavoidable chance at our education culture continues bringing a lot of modernity day by day. In recent years, through e-learning demand growth and evaluation zone of influence bring to minds the question 'What is happening at e-learning community?' doubtless.

M-learning is such as e-learning which is coming into existence evaluation of e-learning domains as together and accessing ability into e-learning contents environment independent. Also m-learning provide to benefit dynamically created services and to keep in communication with each other m-learning clients. At this point of view, m-learning is e-learning with 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. Mobile learning gives instructors and learners increased flexibility and new opportunities for interaction. Mobile technologies provide to learning experiences being collaborative, accessible, and integrated with the world beyond the classroom. Notwithstanding, m-learning can be defined as "... any service or facility that supplies a learner with general electronic information and educational content that aids in acquisition of knowledge regardless of location and time ..." (Lehner & Nosekabel, 2002).

According to Sharples (2000), palmtop computers could be useful lifelong learning tools. He claimed that mobile devices could accompany learners throughout their lives and be used to input data and access information whenever the learner need it at anytime and anywhere. In this context, there are apperent varieties of mobile device. Smart media-based cognitive trait models are being developed that preserve the state of learner's cognitive attributes (such as working memory capacity, inductive reasoning ability, information processing speed and so on) and allow learners to plug-in these modules to any learning environment to receive instant adaptation (Kinshuk, & Patel, 2003).

Although, there a lot of adaptive web interface definitions, in actual fact adaptive web interface's intent is to provide device specific view according to device capabilities and

browser specifications. Hence, we need adaptive web interface development framework to handle these browser requests from various devices (both mobile and desktop platforms) and to route client's request to the concerned with view automatically. Firstly, we need to know device information to provide an adaptive web interface to clients and secondly application sends views concerned with device specification. Therefore, we used mvc framework in this project.

MVC design pattern will split your application in 3 different parts (model, view, controller). Separation of concern is one of its main benefits. Besides, one of most important feature of MVC you can write your own route engine. This feature provides us to route browser request according to device specification. In this project, we also used Ajax web controller for interactive and user friendly web interfaces.

Above quoted all over topics formed our project and in light of the foregoing we developed an adaptive web interface for m-learning by using following technologies.

2. METHODOLOGIES

2.1. MODEL VIEW CONTROLLER

Software Architects have been arguing a lot of debates about various approaches and architectures. The most highlights of examples are object relational mapping, store procedures, REST (Representational State Transfer) and SOAP (Simple Object Access Protocol) etc. There is a debate happening inside the Microsoft community about ASP.net web form Vs ASP.net MVC. Especially, nowadays a major part of developers thinking that ASP.net MVC will be replace webforms at least eventually and others are thinking that ASP.net MVC will not be replace webforms. By reason of routing flexibility and development professionalism we decide to develop our thesis project with MVC framework.

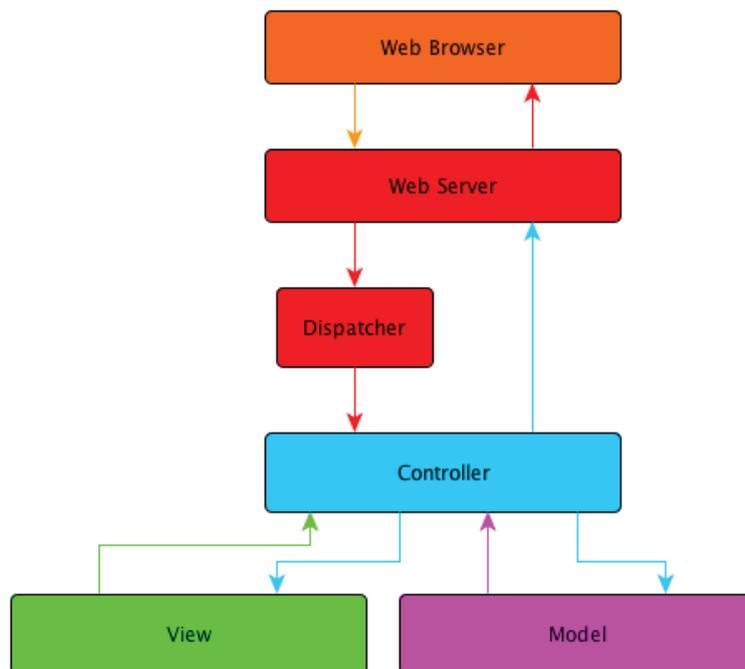


Figure 2.2.1.1 Model View Controller

Reference:(Bacon, 2008)

Model-View-Controller (MVC) is a classic design pattern often used by applications that need the ability to maintain multiple views of the same data. The MVC pattern hinges on a clean separation of objects into one of three categories — models for maintaining data, views for

displaying all or a portion of the data, and controllers for handling events that affect the model or view(s) (Bacon 2008).

Because of this separation, multiple views and controllers can interface with the same model. Even new types of views and controllers that never existed before can interface with a model without forcing a change in the model design (eNode, 2002).

2.2. ORTHOGONALITY

The basic idea of orthogonality is that things that are not related conceptually should not be related in the system. Parts of the architecture that really have nothing to do with the other, such as the database and the UI, should not need to be changed together. A change to one should not cause a change to the other (Venners, 2003).

Successful use of the pattern isolates business logic from user interface considerations, resulting in an application where it is easier to modify either the visual appearance of the application or the underlying business rules without affecting the other.

It is common to split an application into separate layers that run on different computers: presentation (UI), domain logic, and data access. In MVC the presentation layer is further separated into view and controller (Bacon 2008).

2.3. THE DRY PRINCIPLE

DRY (Don't repeat yourself) says that every piece of system knowledge should have one authoritative, unambiguous representation. Every piece of knowledge in the development of something should have a single representation. A system's knowledge is far broader than just its code. It refers to database schemas, test plans, the build system, even documentation.

Given all this knowledge, why should you find one way to represent each feature? The obvious answer is, if you have more than one way to express the same thing, at some point the two or three different representations will most likely fall out of step with each other. Even if they don't, you're guaranteeing yourself the headache of maintaining them in parallel whenever a change occurs. And change will occur. DRY is important if you want flexible and maintainable software (Venners, 2003).

2.4. AGILE SOFTWARE DEVELOPMENT

Agile Software Development refers to a group of software development methodologies that are based on similar principles. Agile methodologies generally promote: A project management process that encourages frequent inspection and adaptation; a leadership philosophy that encourages team work, self-organization and accountability; a set of engineering best practices that allow for rapid delivery of high-quality software; and a business approach that aligns development with customer needs and company goals (Agile Manifesto, www.agilemanifesto.org/).

2.5. AGILE MANIFESTO

- Individuals and interactions over processes and tools.
- Working software over comprehensive documentation.
- Customer collaboration over contract negotiation.
- Responding to change over following a plan, that is, while there is value in the items on the right, we value the items on the left more.
- Customer satisfaction by rapid, continuous delivery of useful software.
- Working software is delivered frequently (weeks rather than months).
- Working software is the principal measure of progress.
- Even late changes in requirements are welcomed.
- Close, daily cooperation between business people and developers.
- Face-to-face conversation is the best form of communication (Co-location).
- Projects are built around motivated individuals, who should be trusted.
- Continuous attention to technical excellence and good design.
- Simplicity.
- Self-organizing teams.
- Regular adaptation to changing circumstances.

3. OPEN API'S AND MODELS FOR WEB 2.0 TECHNOLOGIES

Professional developers care to write codes more readable, reuseable and maintainable. This sensibility provide to a project developer easier entegration. Especially, big project has big 'know-how' and surely must be used at any projects by any developers.If We want to our project to gain to Open api's and models profits ,we should use development principle.

According to Bacon (2008), web 2.0 applications grow and scale it is important that certain development principles are followed and maintained using architectural software models such as Representational State Transfer (REST) for distributed media and resources, Rapid Development Methodology (RDM) where changes to the system are made available for use without building and deployment cycles, restarts or reloading of key components, Model View Controller (MVC) which isolates business logic from the user interface allowing for (code) changes in on layer without the need to (rewrite code) change other layers and reduction of coupling through the DRY principle and orthogonality which eliminates propagation; modified code or structure in one layer should not cause changes in other layers of the application.

3.1. EXTENSIBLE MARKUP LANGUAGE

According to W3Shool Extensible Markup Language (XML) provides a foundation for creating documents and document systems. XML operates on two main levels: first, it provides syntax for document markup; and second, it provides syntax for declaring the structures of documents. XML is clearly targeted at the Web, though it certainly has applications beyond it. Users who have worked with HTML before should be able to learn the basics of XML without too much difficulty. XML's simplicity is its key selling point, perhaps even its strongest feature.

In this m-learning project XML's set of tools allows us to create web pages and much more. XML allows us to set standards defining the information that should appear in a document, and in what sequence. XML, in combination with other standards, makes it possible to define the content of a document separately from its formatting, making it easy to reuse that content in other applications or for other presentation environments. Most important, XML provides a basic syntax that can be used to share information between different kinds of computers, different applications, and different organizations without needing to pass through many layers of conversion.

3.2. REPRESENTATIONAL STATE TRANSFER

REST (Representational State Transfer) is a style of software architecture for distributed hypermedia systems such as the World Wide Web. REST refers to a collection of network architecture principles, which outline how resources are defined and addressed (MSDN, ADO.NET Data Services Terminology).

Açıklama [AK1]: Tek bir kaynaktan bu kadar çok alıntı olmaz. Kendi cümlelerin ile düzenle.

In this m-learning project REST is one of most important development strategy according to me. Because, in the default page controller based design in ASP.NET, we don't follow a strict REST-based architecture. If we use a pure REST-based architecture, then all of the information required to access a particular resource would be in the URI. This means that we don't need to check if a postback happened or not, because each request is unique in itself and will be treated differently (via unique URLs). Whereas in ASP.NET, we can use the postback technique to make the same requests using the same URLs and do different processing based on whether it is a postback or not. Many a times, in numerous projects, we come across the following coding style in ASP.NET code-behind files:

```
btnSave_Click()
{
    Response.Redirect("~/MyPage.aspx");
}
```

Here, on the postback (button click), we are redirecting the user to another resource (mypage.aspx), and this approach goes against the REST principle as we are delegating the responsibility to load a resource to another page based controller's postback event. This is not REST-like behavior. Now, we will see how MVC compliments the REST approach.

REST provides improved response time and reduced server load due to its support for the caching of representations. It improves server scalability by reducing the need to maintain session state. This means that different servers can be used to handle different requests in a session. It requires less clientside software to be written than other approaches, because a single browser can access any application and any resource. REST depends less on vendor software and mechanisms, which layer additional messaging frameworks on top of HTTP and provides equivalent functionality when compared to alternative approaches to communication. REST does not require a separate resource discovery mechanism, due to the use of hyperlinks in representations.

REST provides better long-term compatibility and evolvability characteristics than RPC. This is due to: The capability of document types such as HTML to evolve without breaking backwards- or forwardscompatibility. The ability of resources to add support for new content types as they are defined without dropping or reducing support for older content types (Kay, 2007).

3.3. RESTFUL WEB SERVICES CHARACTERISTICS

We have built m-learning web sites that can be used by desktop computer. But can learners also build m-learning web sites that are usable by machines? We developed a m-learning web site which shows you how to do.

Today's web service technologies have lost sight of the simplicity that made the Web successful. This thesis explains how to put the "Web" back into web services with REST, the architectural style that drives the Web.

- Stateless: each request from client to server must contain all the information necessary to understand the request, and cannot take advantage of any stored context on the server.
- Cache: to improve network efficiency responses must be capable of being labeled as cacheable or non-cacheable.
- Uniform interface: all resources are accessed with a generic interface (e.g., HTTP GET, POST, PUT, and DELETE).
- Named resources - the system is comprised of resources, which are named using a URL.
- Interconnected resource representations - the representations of the resources are interconnected using URLs, thereby enabling a client to progress from one state to another.
- Layered components - intermediaries, such as proxy servers, cache servers, gateways, etc, can be inserted between clients and resources to support performance, security, etc (Costello, 2002).

3.4. SOAP

SOAP is a protocol for exchanging XML-based messages over computer networks, normally using HTTP/HTTPS. SOAP forms the foundation layer of the web services protocol stack providing a basic messaging framework upon which abstract layers can be built. Using SOAP

over HTTP allows for easier communication through proxies and firewalls than previous remote execution technology. SOAP is versatile enough to allow for the use of different transport protocols. The standard stacks use HTTP as a transport protocol, but other protocols are also usable (e.g., SMTP). SOAP is platform independent and language independent (W3C, 2000).

3.5. WEB SERVICE DESCRIPTION LANGUAGE

WSDL is an XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information. The operations and messages are described abstractly, and then bound to a concrete network protocol and message format to define an endpoint. Related concrete endpoints are combined into abstract endpoints (services). WSDL is extensible to allow description of endpoints and their messages regardless of what message formats or network protocols are used to communicate, however, the only bindings described in this document describe how to use WSDL in conjunction with SOAP 1.1, HTTP GET/POST, and MIME (W3C, 2000). WSDL is often used in combination with SOAP and XML Schema to provide web services over the Internet. A client program connecting to a web service can read the WSDL to determine what functions are available on the server. Any special data types used are embedded in the WSDL file in the form of XML Schema. The client can then use SOAP to actually call one of the functions listed in the WSDL (Christensen¹, Curbera², Meredith³, Weerawarana, 2001).

4. PRESENTATION TIER (UI)

4.1. EXAMPLE AJAX PRESENTATION LAYER DATAFLOW

We used MVC framework in this m-learning project and separation concern is most important issue to our project. So we have presentation tier and presentation tier contains a lot of dynamic Ajax controller. According MSDN Ajax provides us the following features:

- A typical AJAX utilizes the Presentation Layer (UI) to create dynamic interaction within a Internet Browser without the need to reload elements.
- AJAX utilizes JavaScript and other programming languages to create a stateful and dynamic application
- AJAX allows for the display of Rich Media and interaction with the content without having to reload the browser
- Content is requested and delivered using HTTP requests
- All the processing of data is done on the Application Level leaving the Presentation Layer to just display content.

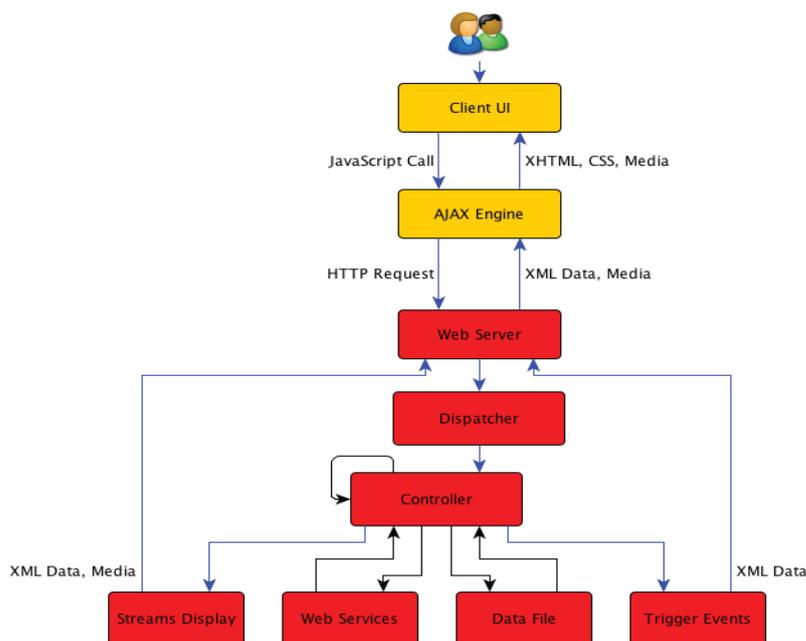


Figure 4.1.1 Ajax presentation layer

Reference: (Bacon, 2008)

4.2. AJAX

AJAX (Asynchronous JavaScript and XML) provide web pages dynamic interaction with users. AJAX also reduce connecitons to server, since web page make connection based on partial web page. We use jquery framework for AJAX compenent at our m-learning project.

However, some processing at the server side may be required to handle requests, such as finding and storing the data. This is accomplished more easily with the use of a framework dedicated to process Ajax requests. The goal of the framework is to provide the Ajax engine described below and associated server and client-side functions (Garrett, 2005, Merrill, 2006).

5. RUBY ON RAILS

5.1. EXAMPLE APPLICATION LAYER DATA FLOW

In this thesis m-learning web application utilizing AJAX and Ruby on Rails, a user enters a record through the Client UI.

- After the record is complete the user clicks submit.
- This action sends a HTTP request though the AJAX interface engine to the web server.
- The content of the note is formatted to the proper XML specification and is passed along.
- The web server dispatches the request to the Ruby on Rails application server.
- The Rails controller processes the request and a standard routine is executed for updating a record.
- The Rails controller then updates the record in the Local Cache Storage and a notification of the CRUD is returned to the controller.
- The Rails controller also executes a routine to the streams display, which updates the record via the web server and the browser AJAX engine notifying the user that the record has been updated and saved.
- A secondary background process is executed in the Rails controller that uses REST / SOAP to connect to other external web services to update the master record in a master database.
- Updating the Local Cache Store's record first and displaying the record to the browser reduces latency as the secondary update to the external database can be processed as needed in scheduled batches or over a slower Internet connection.

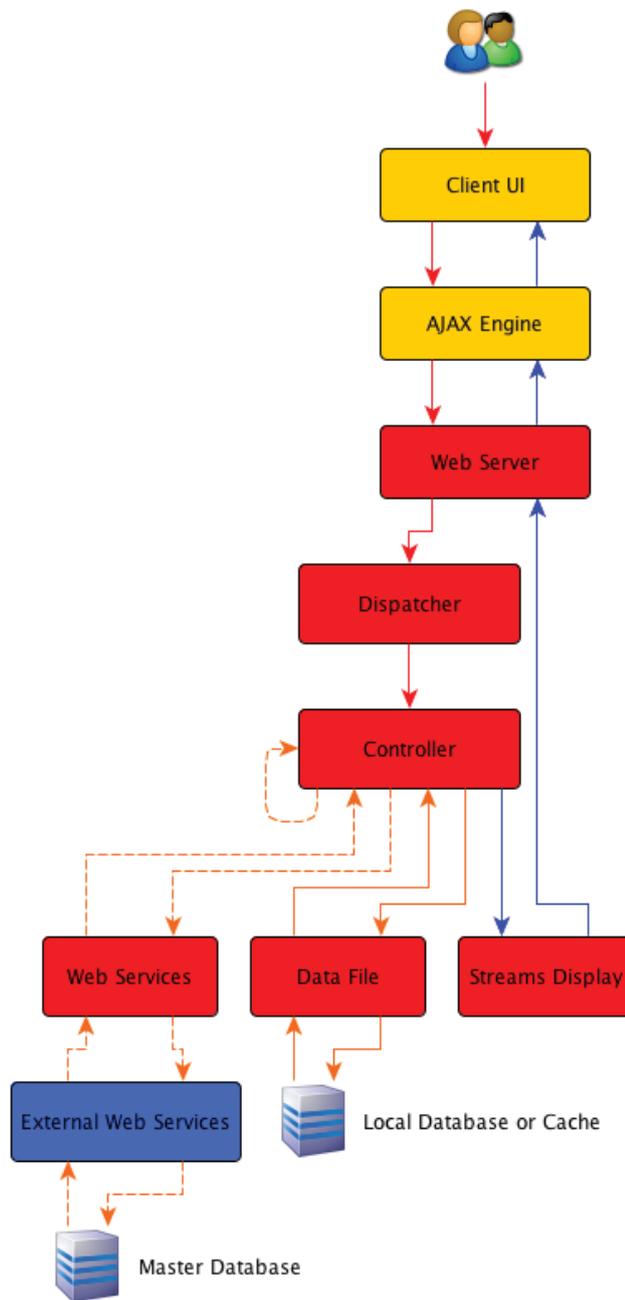


Figure 5.1.1 Application layer data flow

Reference: (Bacon, 2008)

5.2. RUBY ON RAILS

Firstly we decided to use model view controller at our project. Additionally we thought that use DRY principles and orthogonality at this project. At this context, Ruby on Rails (RoR) provide to implement DRY principles and orthogonality.

Additionally we need mock object testing for user interface, at this point Rails provide a framework that give us flexibility a lot of testing issues. One of the most important features of the Rails SOAP, REST and JavaScript.

Ruby provides a reduced code footprint over .NET and Java applications. Some developers have noted a reduction of code that is 6 to 10 times less than the standard Java application. This decrease in code allows for rapid prototyping and development without being held back by thousands of lines of code. This also reduces the time needed for new team members to understand the code base allowing for a reduced learning curve of the software system (Bradley, 2006, Hunt/Thomas, 1999, Raymond, 2003).

6. MOBILE AND ASP.NET MVC

6.1. MOBILE BROWSER DEFINITION FILE

The Mobile Browser Definition File contains definitions for individual mobile devices and browsers. At run time, ASP.NET uses the information in the request header to determine what type of device/browser has made the request (MSDN, Browser Definition File Schema).

This project provides a data file that when used with ASP.NET will detect the incoming mobile device and present you as the web developer with a set of 67 capabilities or properties describing the requesting device. These capabilities range from screen size to cookie support and provide all the information you need to adaptively render content for mobile phones and devices.

The Mobile Device Browser Definition File contains capability definitions for individual mobile devices and browsers. At run time, ASP.NET uses this .browser file, along with the information in the HTTP request header, to determine what type of device/browser has made the request and what the capabilities of that device are. This information is exposed to the developer through the Request.Browser property and allows them to tailor the presentation of their web page to suit the capabilities of the target device.

The format of the .browser file is XML and the schema is defined here: Browser Definition File Schema. The Browser Definition File was introduced in the .NET Framework version 2.0. In earlier versions of the .NET Framework, the browserCaps element was used to define browser definitions in configuration files.

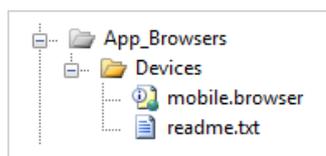


Figure 6.1.1 Mobile device browser file

First, I downloaded the new **Mobile Device Browser File** and put it in \App_Browsers\Devices in my ASP.NET project. Just by dropping this file in the App_Browser\Devices folder, I automatically get all these 67 capabilities populated in the Request.Browser option. I can address them like

Request.Browser["AjaxSupportsGetElementById"]. Just open up the mobile.browser file to see all your choices.

For example, ordinarily if you're in the Home Controller and the Index action, the ViewEngine's FindView() method gets called and will look in /Views/Home for Index.aspx.

The Mobile ViewEngine needs to detect the device and change where we look. I could just make a "Mobile" version and call it a day. However, since we've got a nice tree structure here, why not set up:

/Views/Home

/View/Home/Mobile

/View/Home/Mobile/iPhone

/View/Home/Mobile/MobileIE

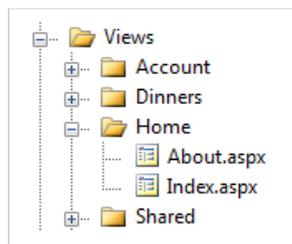


Figure 6.1.2 Views containing folders

etc...so, now we can have many different files, any combination of them.

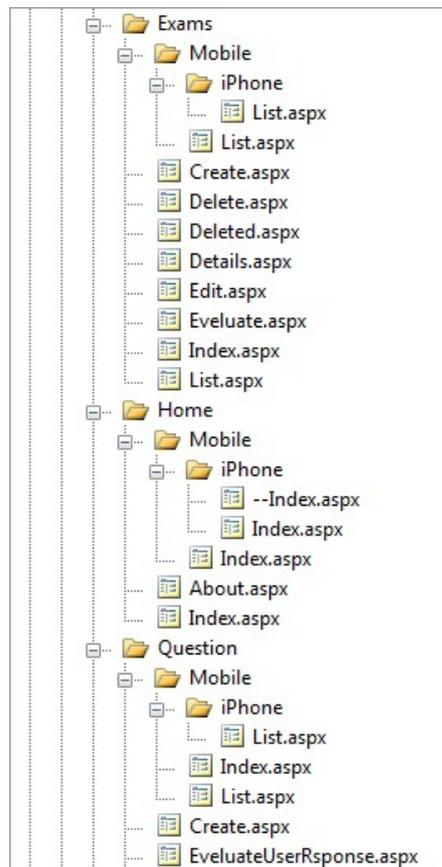


Figure 6.1.3 Mobile views

We can also have things fallback neatly. For example, we can create a really specific View, say, for a Nokia "Whatever" when it needs the details, but that Nokia will use the standard Mobile view in all other situations. This is accomplished by just the folder convention.

6.1.1. A simple example to start

We want to redirect the request from user client device to the view according to device specific. We aimed to provide client to see correct view and interact easily. So we wrote a view engine module like following.

```
public class MobileCapableWebFormViewEngine : WebFormViewEngine
{
    public override ViewEngineResult FindView(ControllerContext controllerContext, string
    viewName, string masterName, bool useCache)
    {
```

```

ViewEngineResult result = null;
var request = controllerContext.HttpContext.Request;

// Avoid unnecessary checks if this device isn't suspected to be a mobile device
if (request.Browser.IsMobileDevice)
{
    result = base.FindView(controllerContext, "Mobile/" + viewName, masterName,
useCache);
}

//Fall back to desktop view if no other view has been selected
if (result == null || result.View == null)
{
    result = base.FindView(controllerContext, viewName, masterName, useCache);
}

return result;
}
}

```

This can be expanded on like this (and in a number of other ways...switch statements, tables, conventions, it's up to you to map the Views as you like.

```

public class MobileCapableWebFormViewEngine : WebFormViewEngine
{
    public override ViewEngineResult FindView(ControllerContext controllerContext, string
viewName, string masterName, bool useCache)
    {
        ViewEngineResult result = null;
        var request = controllerContext.HttpContext.Request;

        //This could be replaced with a switch statement as other advanced / device specific
views are created
        if (UserAgentIs(controllerContext, "iPhone"))
        {
            result = base.FindView(controllerContext, "Mobile/iPhone/" + viewName,
masterName, useCache);
        }

        // Avoid unnecessary checks if this device isn't suspected to be a mobile device
        if (request.Browser.IsMobileDevice)
        {
            if (UserAgentIs(controllerContext, "MSIEMobile 6"))
            {
                result = base.FindView(controllerContext, "Mobile/MobileIE6/" + viewName,
masterName, useCache);
            }
            else if (UserAgentIs(controllerContext, "PocketIE") &&
request.Browser.MajorVersion >= 4)
            {

```

```

        result = base.FindView(controllerContext, "Mobile/PocketIE/" + viewName,
masterName, useCache);
    }

    //Fall back to default mobile view if no other mobile view has already been set
    if ((result == null || result.View == null) &&
        request.Browser.IsMobileDevice)
    {
        result = base.FindView(controllerContext, "Mobile/" + viewName, masterName,
useCache);
    }
}

//Fall back to desktop view if no other view has been selected
if (result == null || result.View == null)
{
    result = base.FindView(controllerContext, viewName, masterName, useCache);
}

return result;
}

private bool UserAgentIs(ControllerContext controllerContext, string userAgentToTest)
{
    return (controllerContext.HttpContext.Request.UserAgent.IndexOf(userAgentToTest,
StringComparison.OrdinalIgnoreCase) > 0);
}
}

```

Remember when I said there were 67 capabilities for each of the devices? I might need a custom view for a device, but in many instances, I might just want to modify some small aspect of a few based on the browser's characteristics like screen width or height. I could control the Mobile ViewPort like this for phones that support it.

```

1.<meta name="viewport" content="width=<%= Request.Browser["ScreenPixelsWidth"] %>,
height=<%=Request.Browser["ScreenPixelsHeight"]%>"/>

```

A more advanced example might be this. Say I store all my images in PNG format, but I come upon a phone that doesn't support PNG. I might automatically transcode my PNGs into JPGs. Same with a phone that supports WAV but not MP3.

6.2. IPHONE USER INTERFACE FRAMEWORK

iUI is a framework consisting of a JavaScript library, CSS, and images for developing advanced mobile webapps for iPhone and comparable/compatible devices. We use iUI to develop m-learning project iPhone interfaces.

Create WebApps with an iPhone-like Look and Feel iUI has the following features:

- Create Navigational Menus and iPhone-style interfaces from standard HTML
- Use or knowledge of JavaScript is not required to create modern mobile web pages
- Ability to handle phone orientation changes
- Provide a more "iPhone-like" experience in your Web apps

6.2.1. Link in iUI javascript and css files

Açıklama [AK2]: İlk paragrafa Açıklama ekle.

M-Learning project provide iPhone user to use portal iPhone specific user interface. We use a iPhone user interface framework by adding like the following style-sheet.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>iUI Music Demo</title>
<meta name="viewport" content="width=device-width; initial-scale=1.0; maximum-scale=1.0; user-scalable=0;"/>
<style type="text/css" media="screen">@import "../iui/iui.css";</style>
<script type="application/x-javascript" src="../iui/iui.js"></script>
</head>
<body>
```

6.2.2. Define toolbar and main menu

We need a menu bar to access and navigate m-learning portal easily on iPhone device. iPhone user interface framework provide to us to develop user menu and form contents. iUI has some rules to do this like the following. According iUI:

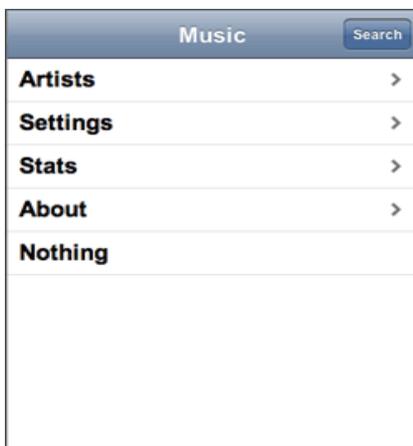
- The toolbar is a div with the class of "toolbar". The definition of the right button is optional.
- The main List or Div should have *selected="true"* as an attribute to make it the main panel
- All Lists should have Ids, which are then referenced in hrefs
- All Lists and Divs should have Title attributes, which are displayed at the top of the panels
- Hrefs that are preceded with the hash symbol (#) reference other "panels" (lists, divs, etc.) in the same html document.
- To reference an external document, reference the page as normal - the page will be loaded via AJAX and inserted.

- Pages loaded via AJAX should be snippets of html, not full html pages (i.e., no html or body tags). Also, note: scripts will not run when loaded in this manner.
- To force a complete page replacement, make the target equal to "_self" on an anchor.
- To replace an anchor with content loaded via AJAX, make the target equal to "_replace"

Açıklama [AK3]: Alıntı ise ref ver. Ayrıca başlığın hemen altından item list başlamaz..

6.2.3. Define any sub-menus

A class of "group" on a list item makes the background the bluegreen color.



```
<div class="toolbar">
  <h1 id="pageTitle"></h1>
  <a id="backButton" class="button"
href="#"></a>
  <a class="button"
href="#searchForm">Search</a>
</div>
<ul id="home" title="Music"
selected="true">
  <li><a href="#artists">Artists</a></li>
  <li><a
href="#settings">Settings</a></li>
  <li><a href="stats.php">Stats</a></li>
  <li><a
href="http://code.google.com/p/iui/"
target="_self">About</a></li>
  <li>Nothing</li>
</ul>
```

Figure 6.2.1 Toolbar main menu

Reference: iUI(iPhone User Interface Framework)

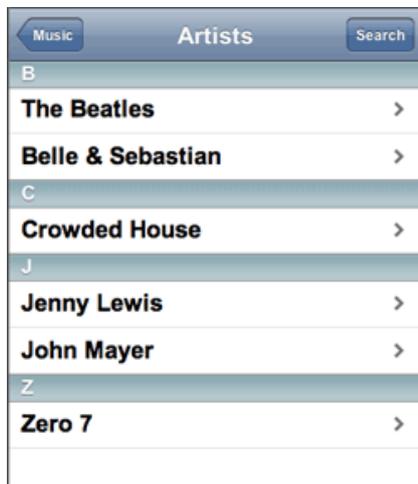


Figure 6.2.2 Sub-menus

Reference: iUI(iPhone User Interface Framework)

```

<ul id="artists" title="Artists">
  <li class="group">B</li>
  <li><a href="#TheBeatles">The
eatles</a></li>
  <li><a href="#BelleSebastian">Belle
& Sebastian</a></li>
  <li class="group">C</li>
  <li><a
href="#CrowdedHouse">Crowded
House</a></li>
  <li class="group">J</li>
  <li><a href="#JennyLewis">Jenny
Lewis</a></li>
  <li><a href="#JohnMayer">John
Mayer</a></li>
  <li class="group">Z</li>
  <li><a href="#Zero7">Zero
7</a></li>
</ul>

```

6.2.4. Define any panels/dialogs

Dialog panel works like messagebox as windows form.

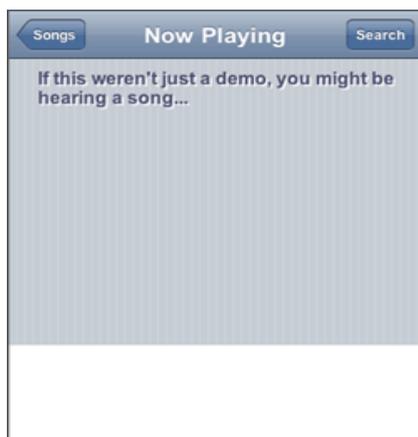


Figure 6.2.3 Panel

Reference: iUI(iPhone User Interface Framework)

```

<div id="player" class="panel"
title="Now Playing">
  <h2>If this weren't just a demo, you
might be hearing a song...</h2>
</div>

```

Note: this is a Div with a class of "panel", which gives the Div a blue pinstripe background

6.2.5. Complex panels

Here is an example of a complex panel, containing On/Off switches (replacements for checkboxes) and input text boxes. Also note the use of Fieldset and Divs with class "row" to create the rounded border style.

Note: the labels are of a width-length, as defined in the stylesheet. Because the label will not wrap properly, you may have to override the default width for the labels on particular panels.



Figure 6.2.4 Complex panel

Reference: iUI(iPhone User Interface Framework)

```
<div id="settings" title="Settings"
class="panel">
  <h2>Playback</h2>
  <fieldset> <div class="row">
    <label>Repeat</label>
    <div class="toggle" onclick=""><span
class="thumb"></span><span class="toggleOn"
>ON </span> <span
class="toggleOff">OFF</span></div>
  </div><div class="row" > <label> Shuffle
</label>
    <div class="toggle" onclick=""
toggle="true" ><span class="thumb"> </span>
<span class="toggleOn"> ON</span> <span
class="toggleOff">OFF</span></div>
  </div></fieldset><h2>User</h2> <fieldset>
  <div class="row">
    <label>Name</label>
    <input type="text" name="userName"
value="johnappleseed"/></div>
  <div class="row">
    <label>Password</label>
    <input type="password"
name="password" value="delicious"/>
  </div>
  <div class="row"> <label>Confirm</label>
  <input type="password"
name="password" value="delicious"/>
  </div>
</fieldset>
</div>
```

6.2.6. Striped tables

I have hobbled together some css and images to pseudo-replicate the iTunes-styled table:

Here's the CSS:

```
01..itable { border: 1px solid gray; }
```

```
02..itable tr.header th { text-align: left; }
```

```
03..itable tr.alt { background-color: #eff7ff; }
}
```

```
04..itable tr.reg { background-color: #fff; }
```

```
05..itable th { background:
url(/iui/blue_hd_bg.png) top left repeat-x;
border-right: 1px solid gray; }
```

```
06..itable th:last-child { border-right: none; }
```

```
07..itable td { border-right: 1px solid
gray; }
```

```
08..itable tr:first-child { white-space: nowrap; }
```

```
09..itable tr:last-child { border-right: none; }
```

Per	Days	Room	Course Name
HR	AB	140	Homeroom 5-2
01-03	AB	140	Math 5
04-06	AB	138	Soc St 5
07-09	AB	139	Science 5
10-11	AB	Cafe162	Lunch 5
12-14	AB	138	Reading 5
15-17	B	AUDC	Chorus 5
15-17	A	139A	AIS ELA 5
18-20	B	GYM	Phys Ed 5
18-20	A	125	Music 5
21-22	AB	223	French 5
23	AB	138	Math Support
24-26	AB	138	ELA 5

Figure 6.2.5 Striped panel

7. ENTITY FRAMEWORK

Entity Framework is a set of technologies in ADO.NET that support the development of data-oriented software applications. Architects and developers of data-oriented applications have struggled with the need to achieve two very different objectives. They must model the entities, relationships, and logic of the business problems they are solving, and they must also work with the data engines used to store and retrieve the data. The data may span multiple storage systems, each with its own protocols; even applications that work with a single storage system must balance the requirements of the storage system against the requirements of writing efficient and maintainable application code(MSDN, Entity Framework Overview).

While we were developing m-learning project the Entity Framework enables us to work with data in the form of domain-specific objects and properties, such as customers and customer addresses, without having to concern themselves with the underlying database tables and columns where this data is stored. With the Entity Framework, developers can work at a higher level of abstraction when they deal with data, and can create and maintain data-oriented applications with less code than in traditional applications. Because the Entity Framework is a component of the .NET Framework, Entity Framework applications can run on any computer on which the .NET Framework starting with version 3.5 SP1 is installed.

7.1. ACCESSING AND CHANGING ENTITY DATA

More than just another object-relational mapping solution, the Entity Framework is fundamentally about enabling applications to access and change data that is represented as entities and relationships in the conceptual model. The Entity Framework uses information in the model and mapping files to translate object queries against entity types represented in the conceptual model into data source-specific queries. Query results are materialized into objects that the Entity Framework manages. The Entity Framework provides the following ways to query a conceptual model and return objects (MSDN,Entity Framework Overwiev):

- LINQ to Entities. Provides Language-Integrated Query (LINQ) support for querying entity types that are defined in a conceptual model. For more information, see LINQ to Entities.
- Entity SQL. A storage-independent dialect of SQL that works directly with entities in the conceptual model and that supports Entity Data Model concepts. Entity SQL is

used both with object queries and queries that are executed by using the EntityClient provider. For more information, see Entity SQL Overview.

- Query builder methods. These methods enable you to construct Entity SQL queries using LINQ-style query methods. For more information, see Query Builder Methods (Entity Framework).

The following diagram illustrates the Entity Framework architecture for accessing data:

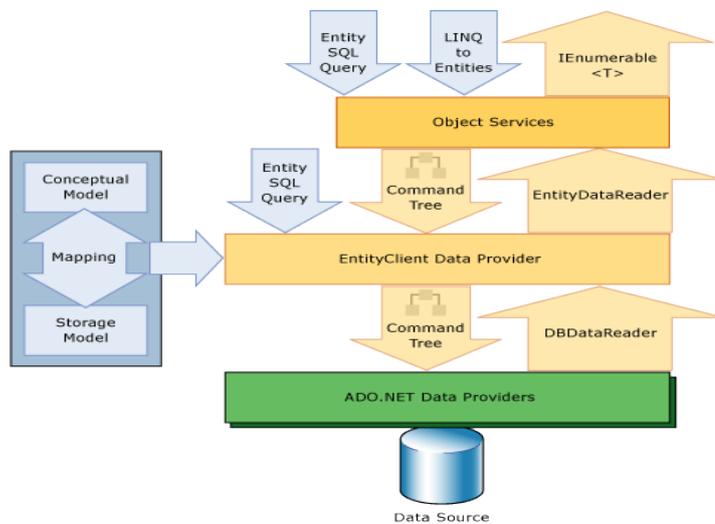


Figure 7.1.1 Entity framework

Reference: MSDN, Entity Framework Overview

This object context provides the facilities for tracking changes and managing identities, concurrency, and relationships. This class also exposes a SaveChanges method that writes inserts, updates, and deletes to the data source. Like queries, these changes are either made by commands automatically generated by the system or by stored procedures that are specified by the developer.

8. MOBILE LEARNING ADAPTIVE WEBSITE OUTLINE

8.1. PROJECT MODULES

In this project, you will design and implement a Web based application that allows multiple choice exams to be created as well as administered. The system should be implemented in Asp.Net 4.0 and MVC Framework. For the database you will use MsSql 2005.

The data requirements for this project have already been analyzed and a relational database schema has been designed. The database stores information about multiple-choice exams, their questions and possible options for answers along with the correct answer.

The project should be implemented in two separate modules:

8.1.1. Admin Module

The admin module should allow administrators to create multiple-choice exams. This is basically a data input/update module that allows admin user to

- **Create Exam:** After collecting top level exam details such as exam title etc, the user should be allowed to add questions one at a time. The add question screen should contain input boxes and text areas for top level question data and a pull down list for number of options (2 through 5). Using Javascript, you should create the right number of answer option data input text areas for answer option text along with a check box that can be checked for "correct answer" option. Once all information is given, the user can submit the question to be added to the database. The user should be presented with the add question screen in case they want to add the next question.
- **Delete Exam:** Given a list of exams, the user chooses exam to be deleted. Only exams in which no one has signed up should be presented. A "confirm delete" screen should be presented before the exam is deleted.
- **Edit Exam:** The user should be able to add new questions at a particular position and delete a question.

8.1.2. User Module

The user module should allow ordinary users to register, sign in, update profile, sign up for exams, take exams, and see their results.

- Register, Change Password, Sign In, and Sign Out: A standard login page (with email and password text boxes) along with a "If you do not have an account, register here" link. Once logged in successfully, the user should be presented with several options including "Update Profile" in which they can change some of the data about themselves such as password, address etc.
- Enroll in Exam(s): A menu option for the user - used to enroll in a particular exam (you may present a select list of all available exams and ask the user to choose one). Note: If the student is already enrolled in the exam and has finished taking the exam or is currently taking the exam (i.e. has started taking the exam but not yet finished), a warning should be issued stating that his answers will be reset. You may confirm that the user wishes to reset the old exam. Once enrolled, you should present the user with a confirmation which includes details about the exam he or she has just signed up for.
- Take an Exam: The user should be presented questions from where they left off the last time they signed on to take the exam. Questions should be presented in order using the pre-defined number of questions per page. Once answers are submitted, they cannot be revisited. The user is then presented the next set of questions until time runs out or there are no more questions..
- View their Grade Report(s): The user chooses the exam for which they like to view results. Only list of exams that have been completed should be presented. The format of the grade report is up to you, but must include number of questions answered correctly, total number of questions, percentage correct, and a detailed listing of user responses and correct answers.

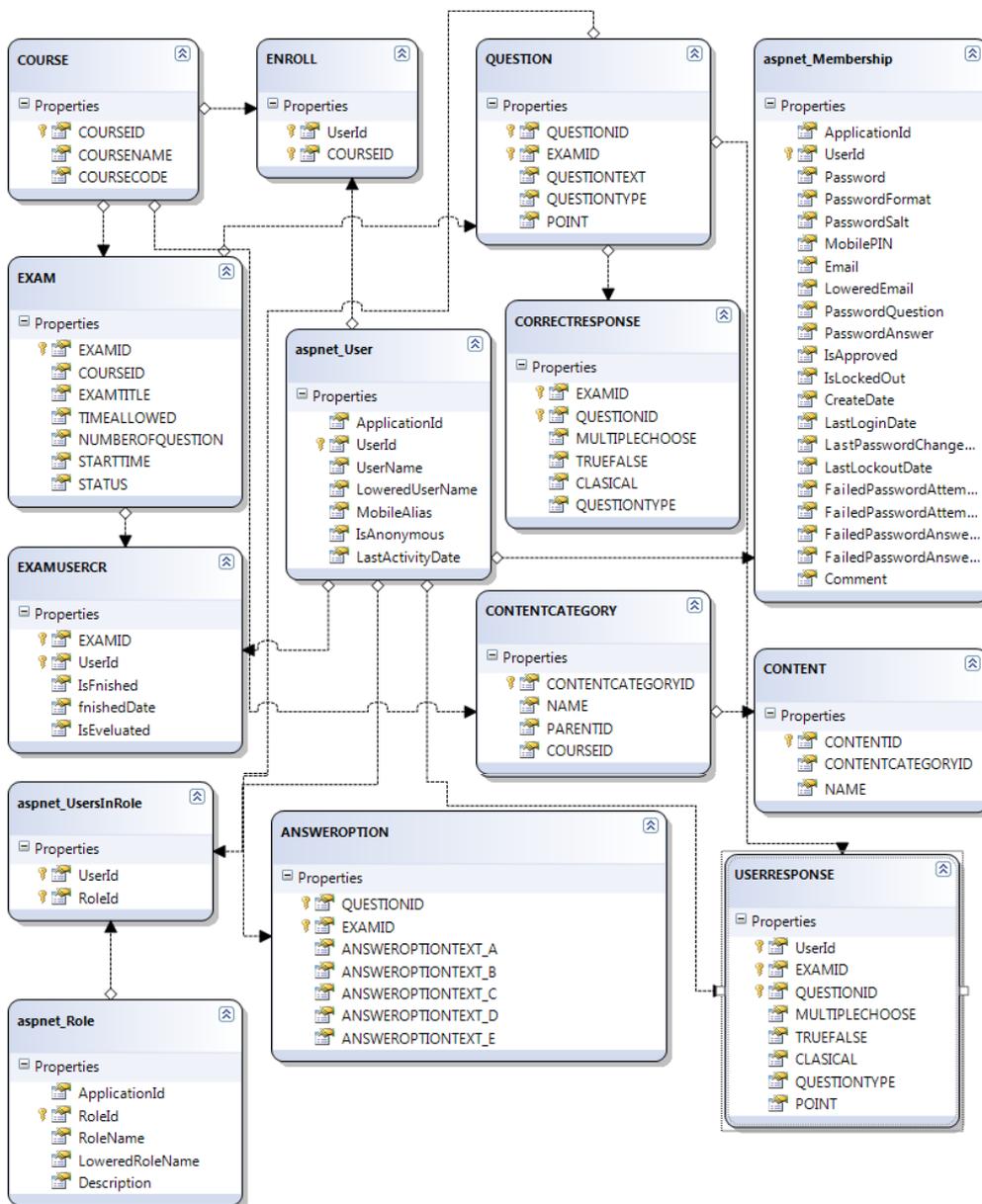


Figure 8.1.1 Mobile learning ER-diagram

8.2. WINDOWS MOBILE AND IPHONE

Microsoft Device Emulator 3.0 is the third publicly released version of the Device Emulator for ARM-based devices. This emulator will also release with Visual Studio 2008. Device Emulator 3.0 will replace the executables and DLLs from previous releases (1.0 and 2.0). The

main addition with Device Emulator 3.0 is the new automation API for the device emulator manager, which allows developers programmatic access to perform the same operations on installed emulator images that can be performed interactively on the Device Emulator Manager drop-down and context menus.

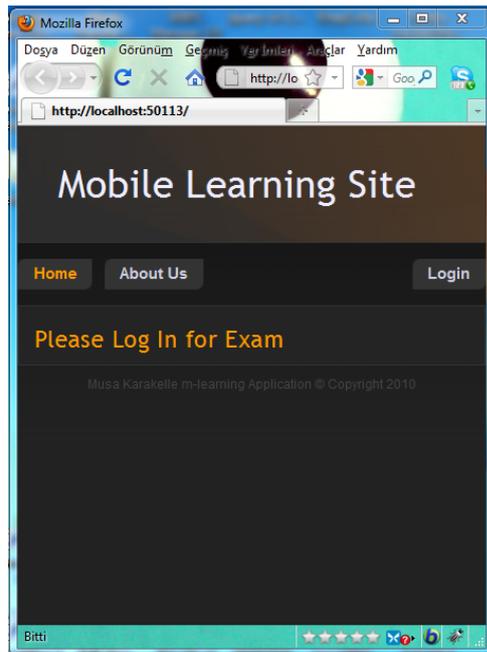


Figure 8.2.1 Computer view

The Windows Mobile 6.1.4 Emulator Images package adds emulator images to Visual Studio 2005 or Visual Studio 2008 that let you test applications for Windows Mobile 6.1.4, including Internet Explorer Mobile 6. For general information about writing software for Windows Mobile, please see the Windows Mobile Developer Center.



Figure 8.2.2 Windows mobile view

The emulator images can also be used as standalone without Visual Studio. They can be accessed through the Start Menu shortcuts.

For iPhone I used Joe Hewitt's IUI JavaScript and CSS Framework (and the Blackbaud quasi-iPhone Simulator). The iPhone includes a lot of Webkit CSS extensions so the view did need be custom. It just doesn't work at all on other browsers, but it looks great on the iPhone. For Windows Mobile, I downloaded the free Windows Mobile 6.1.4 Emulator Images and the Device Emulator.



Figure 8.2.3 iPhone mobile view

9. M-LEARNING ADAPTIVE WEB INTERFACES

9.1. INSTRUCTOR UI

Instructors can manage administrative tool accessing admin panel.



Figure 9.1.1 Instructor Home

Administrative has some m-learning functionality like manage, enrollment, courses, users, roles and account.

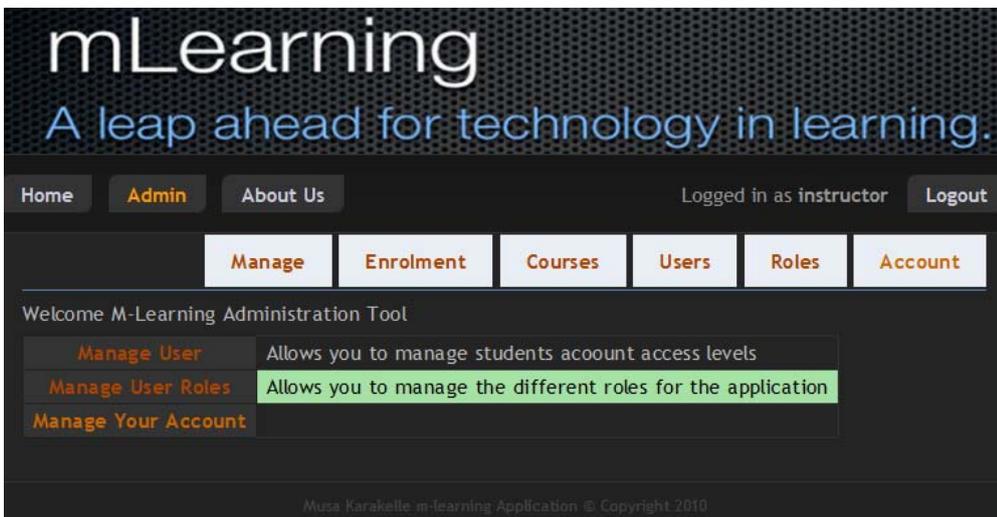


Figure 9.1.2 Instructor Administrator Tools

Instructor who had accessed m-learning system as 'admin' can see student is registered and applied.

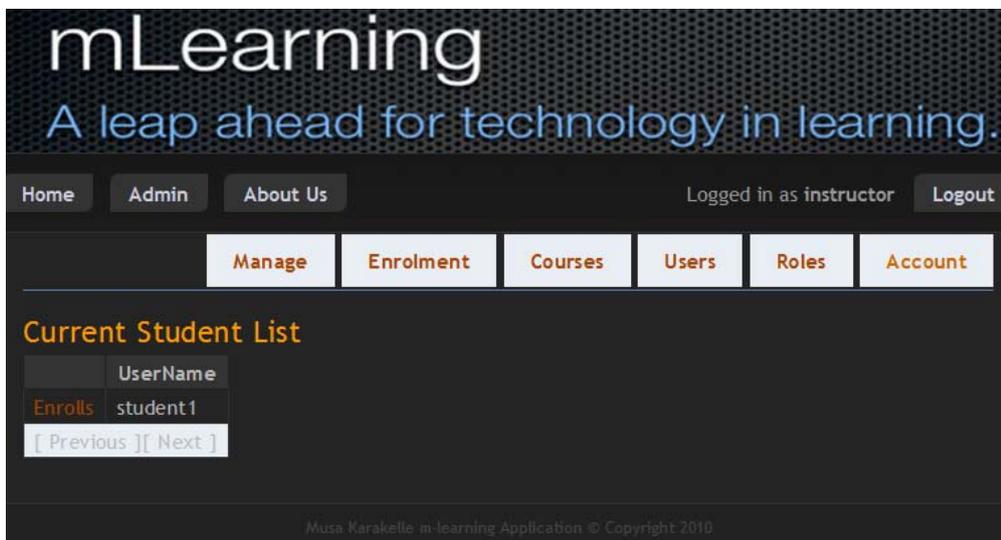


Figure 9.1.3 Student List for Enrolment

Instructor can enroll and drop course on student in student list. Consequently, students can see all course content that is enrolled on them by instructor.

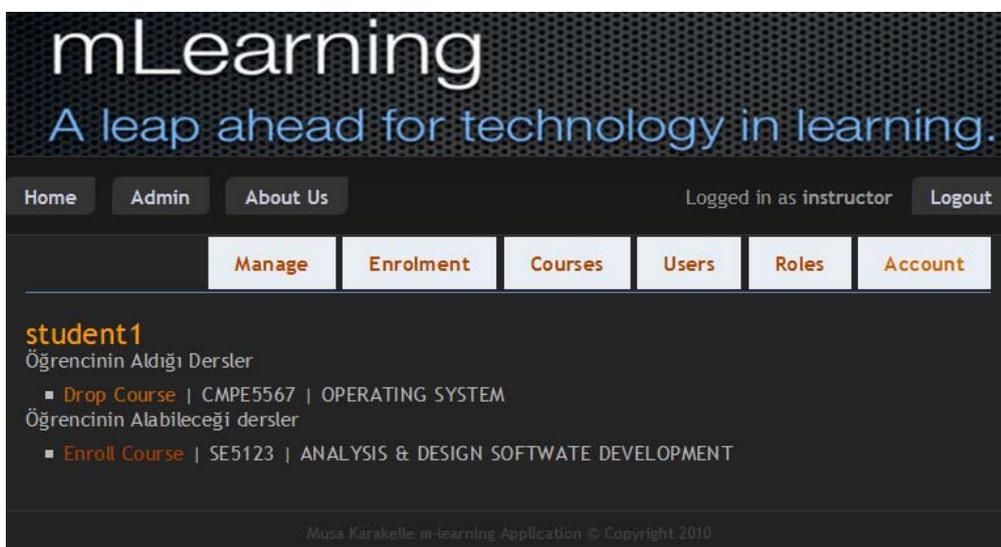


Figure 9.1.4 Enroll & Drop Course

Instructor has some privileges on course operation. These are listing courses, adding course, listing exams and adding exams.

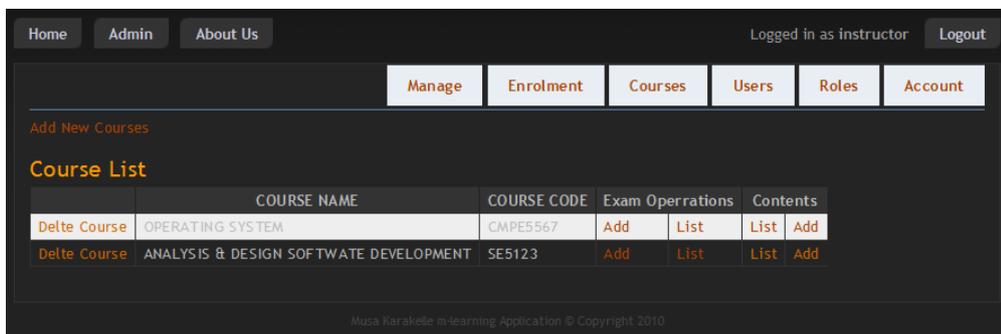


Figure 9.1.5 Course List & Operations

Instructor can add an exam by completing concerning all exam details (Exam title, Time allowed, Number of question, Status, Start date).

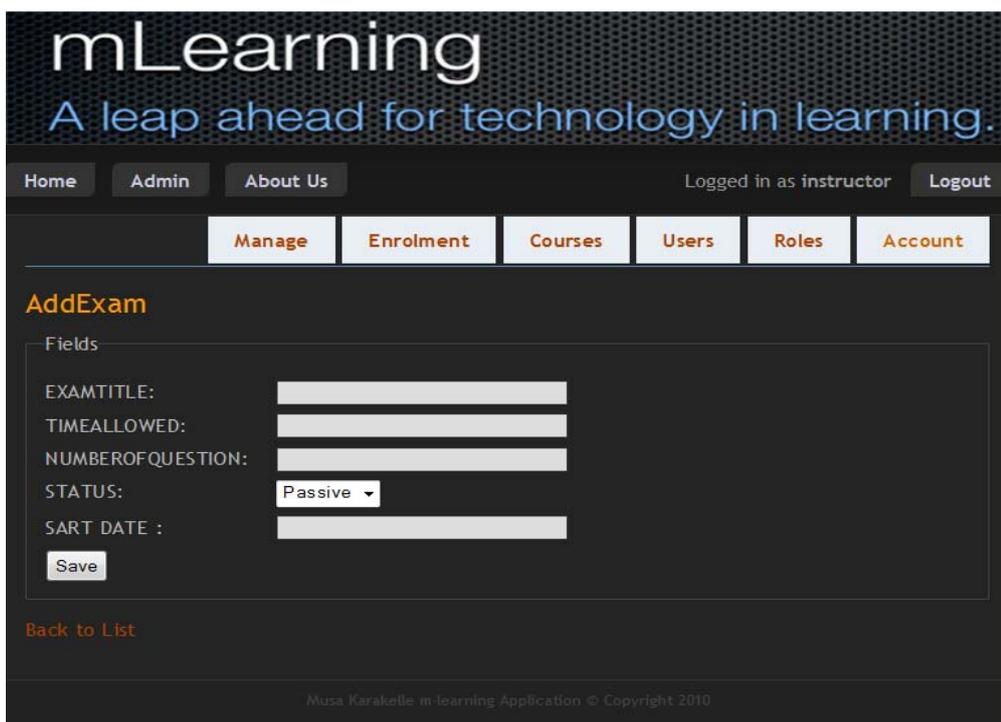


Figure 9.1.6 Exam Adding Form

Instructor can see list of exams and list of exam operations (Add question, Evaluate exam, Edit exam, and List questions) are added by him / her before.

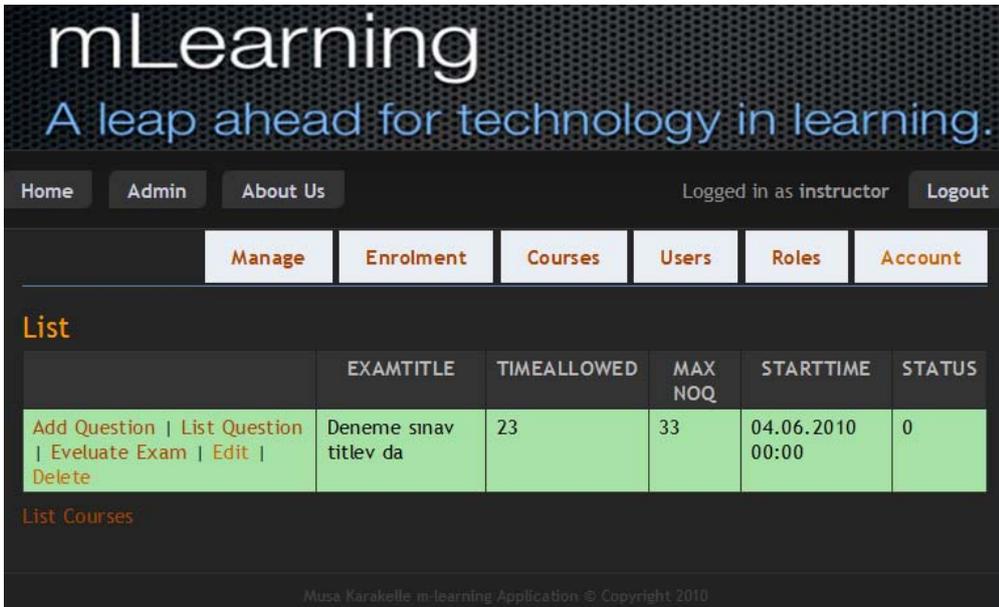


Figure 9.1.1 Exam Listing & Operations

If instructor click evaluate exam link, instructor can see student evaluating list with status if student is not been evaluated before, instructor



Figure 9.1.2 Exams's evaluating status

Instructor can see all student response based on specific question, in there multiple choose questions and true false questions are evaluated by m-learning system according points of questions are defined while questions are preparing.

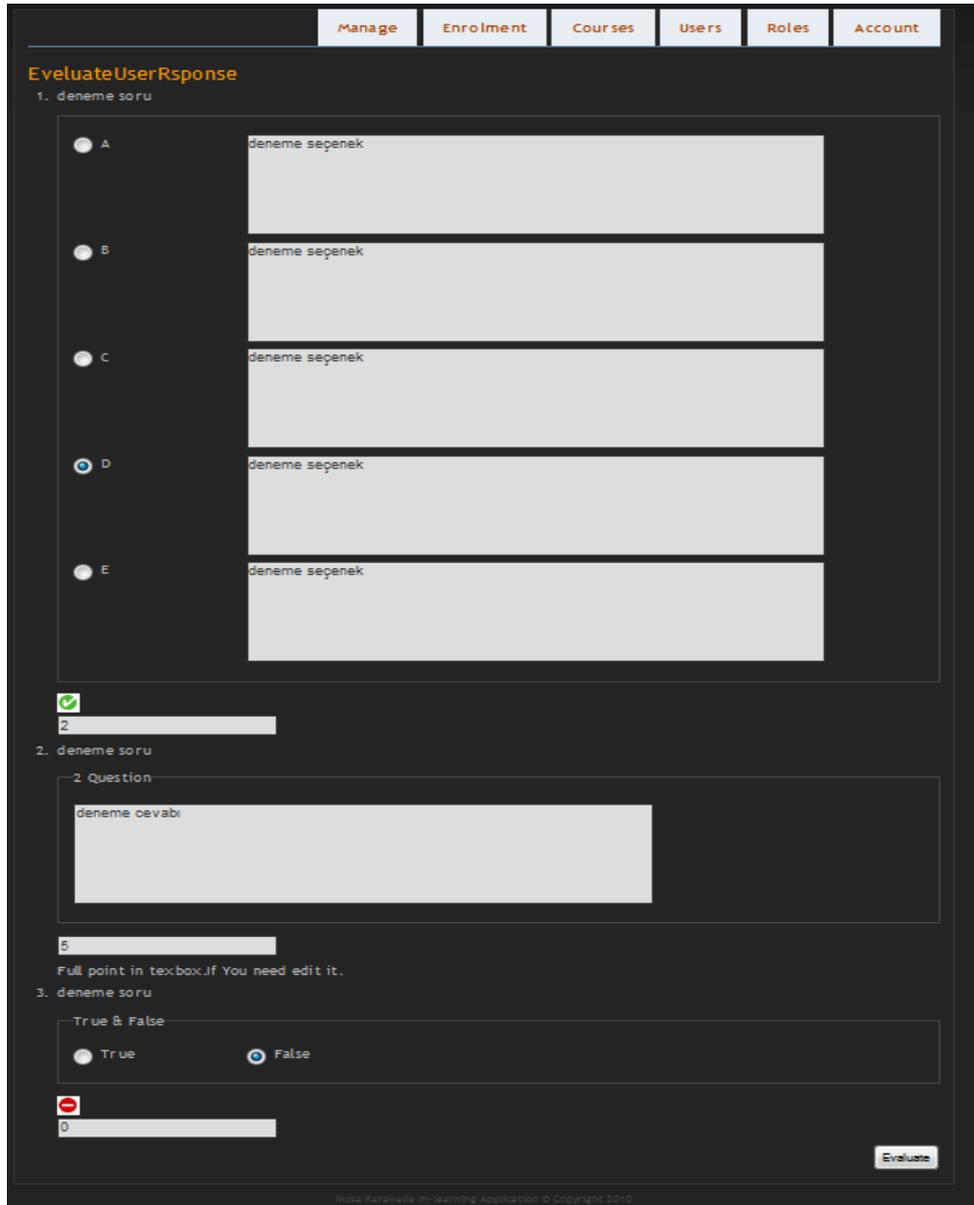


Figure 9.1.3 Evaluating exams

Instructor can prepare questions by choosing question type and completing concerning questions details. If instructor chooses question type as classical, then instructor enters question text, point and correct response.

The screenshot shows the mLearning application interface. At the top, the logo "mLearning" is displayed with the tagline "A leap ahead for technology in learning." Below the logo, there are navigation links for "Home", "Admin", and "About Us". On the right side, it indicates "Logged in as instructor" with a "Logout" button. A horizontal menu contains several options: "Manage", "Enrolment", "Courses", "Users", "Roles", and "Account". The main content area is titled "Create" and contains a form with the following fields:

- Fields:** A section header for the form fields.
- QuestionText:** A large text input field.
- Question Point:** A small text input field.
- Question Type:** A dropdown menu currently set to "Classical".
- Classical:** A large text input field for the correct response.
- Create:** A button to submit the form.
- Back to List:** A link to return to the question list.

At the bottom of the page, there is a small copyright notice: "Nusa Fanielle m-learning Application © Copyright 2010".

Figure 9.1.4 Create classical questions

If instructor chooses question type as True / False, then instructor enter question text, point and correct response.

The screenshot shows the mLearning application interface. At the top, the logo "mLearning" is displayed with the tagline "A leap ahead for technology in learning." Below the logo, there are navigation links: "Home", "Admin", and "About Us". On the right side, it says "Logged in as instructor" and "Logout". A horizontal menu contains "Manage", "Enrolment", "Courses", "Users", "Roles", and "Account". The main content area is titled "Create" and contains a form with the following fields:

- Fields:** A section header for the form fields.
- QuestionText:** A large text input field.
- Question Point:** A text input field.
- Question Type:** A dropdown menu with "TrueFalse" selected.
- True & False:** A section with two radio buttons: "True" (selected) and "False".
- Create:** A button to submit the form.
- Back to List:** A link to return to the list of questions.

At the bottom of the page, there is a small copyright notice: "Musa Karakelle m-learning Application © Copyright 2010".

Figure 9.1.5 Create true / false questions

If instructor chooses question type as multiple choose, then instructor enter question text, point, one by one multiple choose options and correct response.

The screenshot shows a web interface for creating a question. At the top, there are navigation links: Home, Admin, About Us, and a user status bar indicating 'Logged in as instructor' with a Logout button. Below this is a menu with Manage, Enrolment, Courses, Users, Roles, and Account. The main section is titled 'Create' and contains a 'Fields' section. It includes a 'QuestionText:' label followed by a large text input field. Below that is a 'Question Point:' label with a smaller text input field. The 'Question Type:' is set to 'MultipleChose' in a dropdown menu. A sub-section titled 'Multiple Choose Please Enter Options & Correct Chose' contains five radio buttons labeled 'For Choose A' through 'For Choose E', each followed by a text input field. At the bottom left of the form is a 'Create' button, and below the form is a 'Back to List' link. A small copyright notice 'Maza Karakelle m-learning Application © Copyright 2010' is visible at the very bottom.

Figure 9.1.6 Create multiple chooses questions

Instructor can manage all student and guest on user list by entering user details. Instructor can see abstract of user details on user list panel.

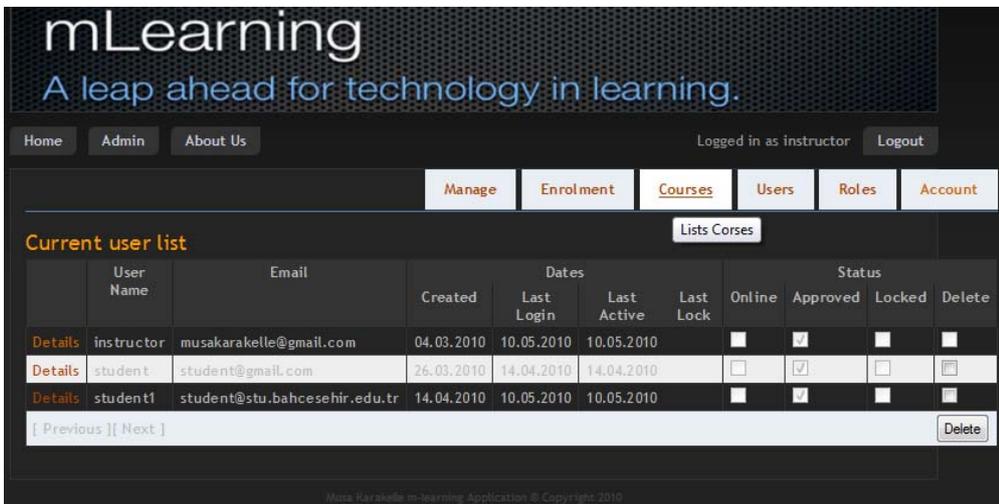


Figure 9.1.7 User list & operations

Instructor can manipulate user information and assign role user. Especially, instructor assigns role to guest user as 'student' is need from guest to be able to see courses details. Because only student has sufficient privileges.

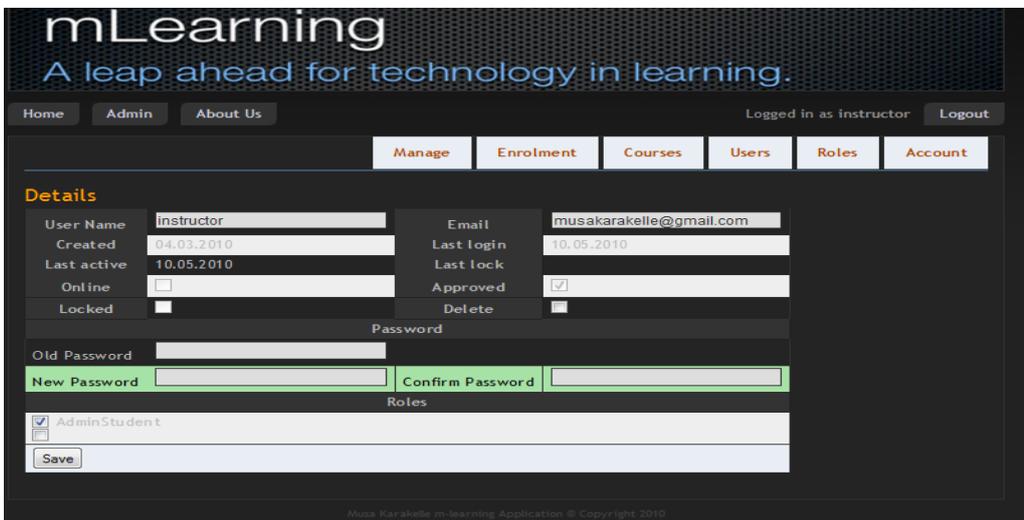


Figure 9.1.8 Account details

Instructor also can add roles for m-learnin system to provide accessing different level privileges.



Figure 9.1.9 Account roles

If instructor needs to provide to students course metarials on m-learing system, instructor can upload metarials on content panel based on couser code and course name.

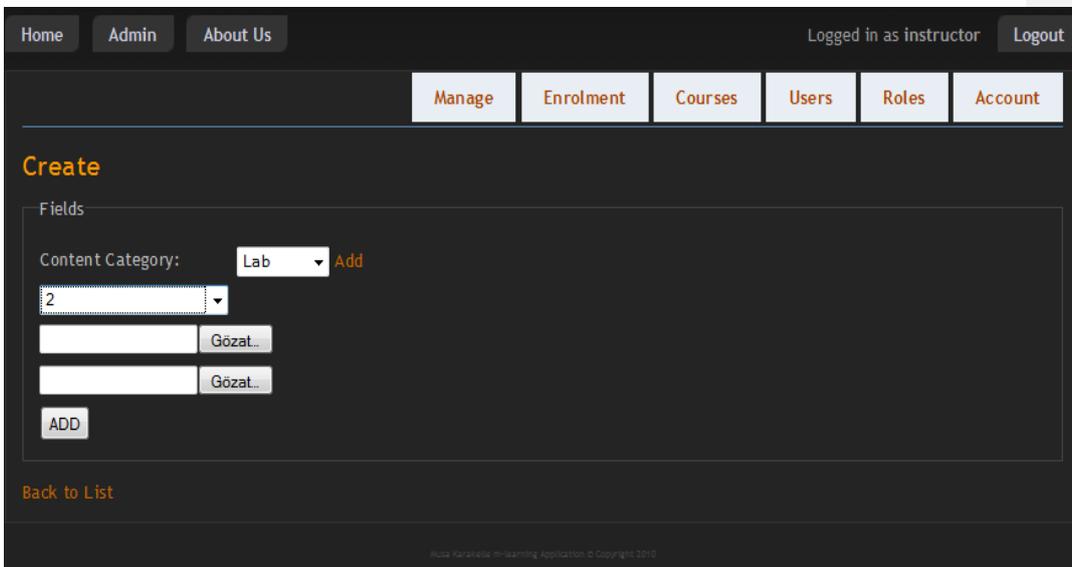


Figure 9.1.10 Content upload panel

If instructor needs a category to upload content into a root title, then instructor can category or sub category from content category panel.

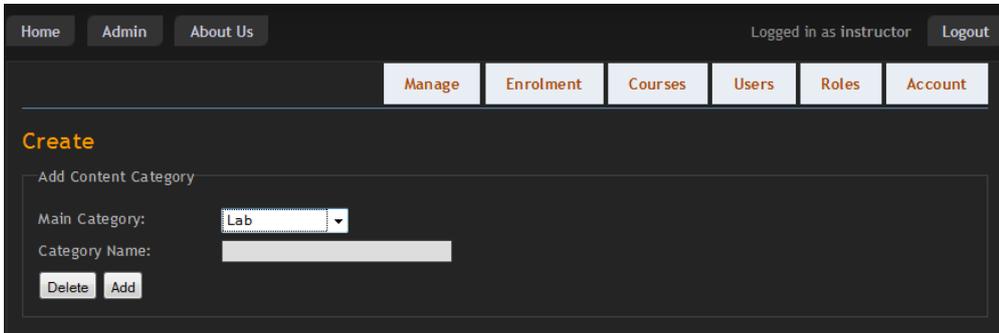


Figure 9.1.11 Content Category

Also, instructor can see contents based on course code and course name by clicking list content link near course operations links.

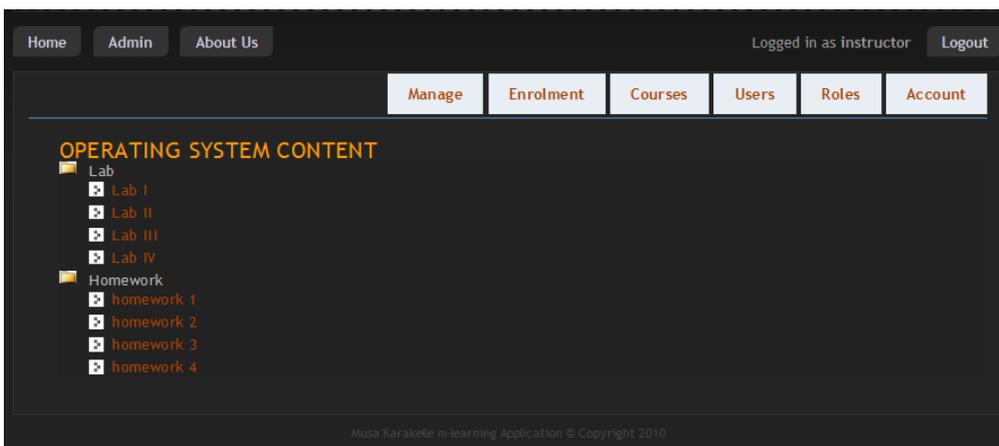


Figure 9.1.12 Content list

Finally, we can say admin panel is made to provide effective m-learning panel to instructor as you see. So instructor manage and manipulate learning stuff quickly and effectively and admin panel make up a good connection between student and instructor.

9.2. STUDENT UI ON WINDOWS MOBILE, IPHONE AND PC



Figure 9.2.1 Student's home page on pc



Figure 9.2.2 Student's home page on windows mobile



Figure 9.2.3 Student's home page on iPhone

Students see different Wellcome screen same URL request based on their devices. Because, m-learning system detect your browser capabilities and device specific and route request concerning view.



Figure 9.2.4 Log On screen on pc



Figure 9.2.5 Log On screen on Windows Mobile

Figure 9.2.6 Log On screen on iPhone

Firstly, student log on m-learning system or register an account and wait for confirmation from instructor to have role as student.



Figure 9.2.7 Wellcome screen on pc



Figure 9.2.8 Wellcome screen on Windows Mobile



Figure 9.2.9 Wellcome screen on iPhone

If student logs on system as student, student can see course list link on menu. Student can terminate session by clicking loo off link.

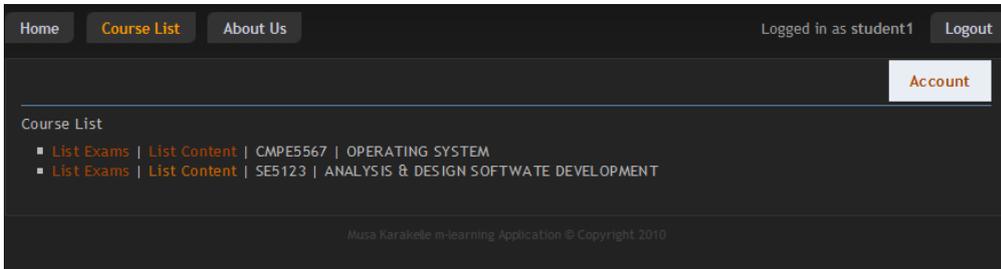


Figure 9.2.10 Course list on pc



Figure 9.2.11 Course list on Windows Mobile



Figure 9.2.12 Course list on iPhone

If student logs on system as student, student see courses and their content links and navigate by clicking they are enrolled by instructors.

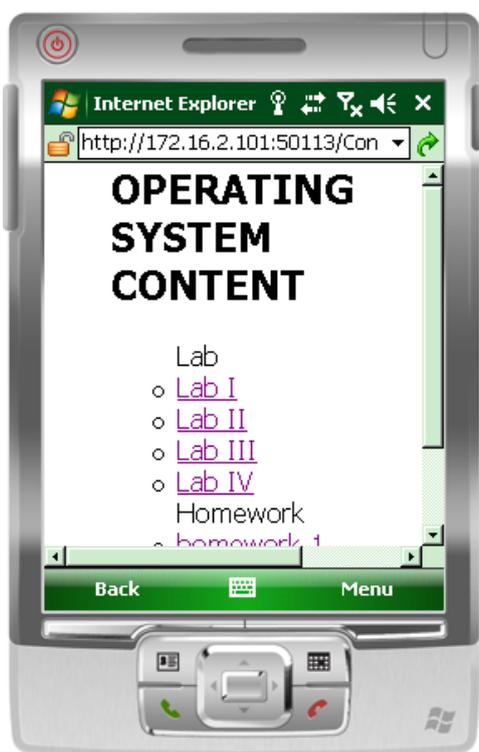
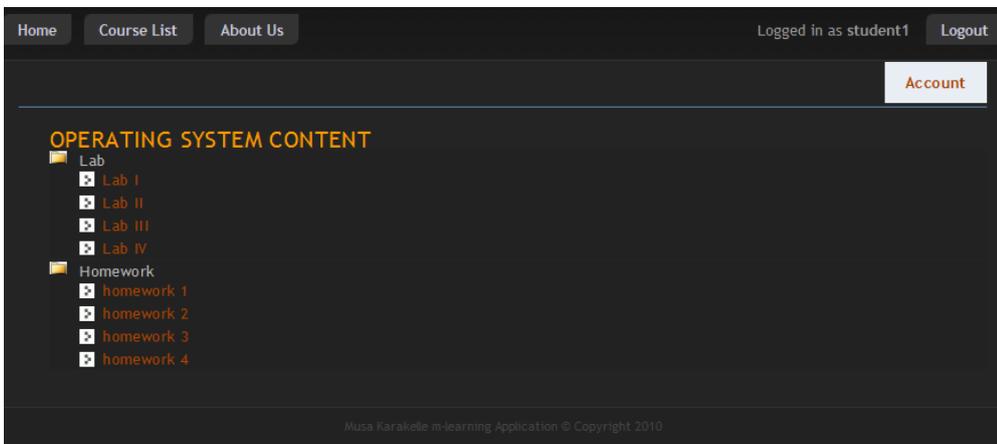


Figure 9.2.13 Course list on Windows Mobile

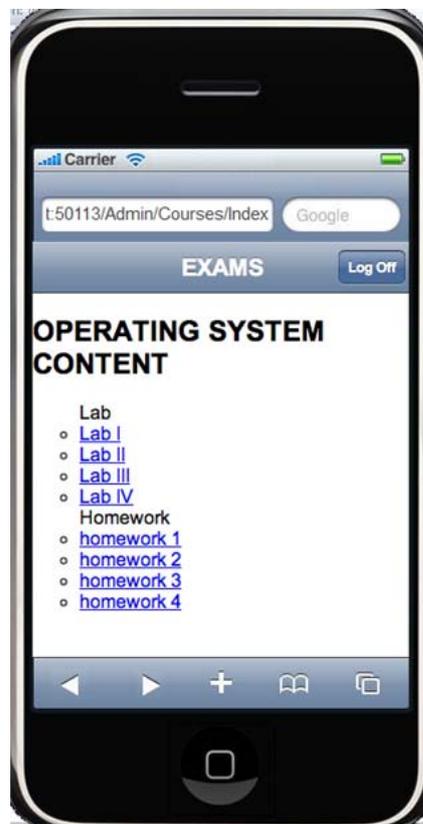


Figure 9.2.14 Course list on iPhone

Student can see course contents by clicking list content link based on category according to course code and course name.

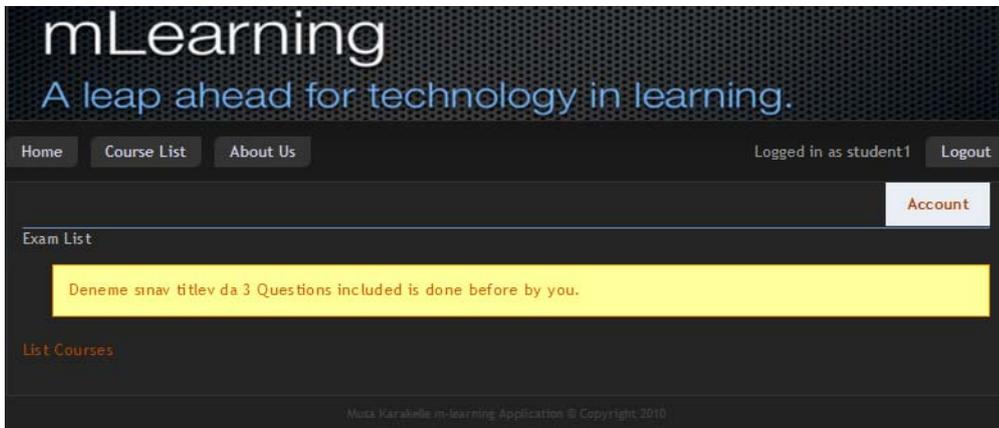


Figure 9.2.15 Finished exam on pc



Figure 9.2.16 Finished exam on Windows Mobile

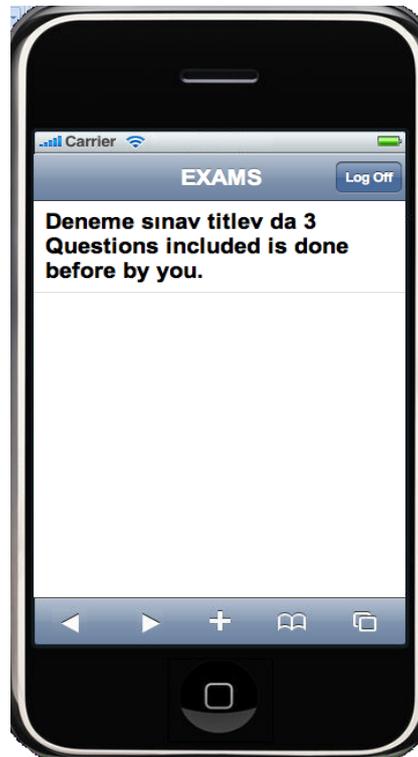


Figure 9.2.17 Finished exam on iPhone

If student clicks list exams, student sees exams. If student completes exam before, m-learning system warn it 'exam is completed before by you'.



Figure 9.2.18 Exam list on pc



Figure 9.2.19 Exam list on Windows Mobile

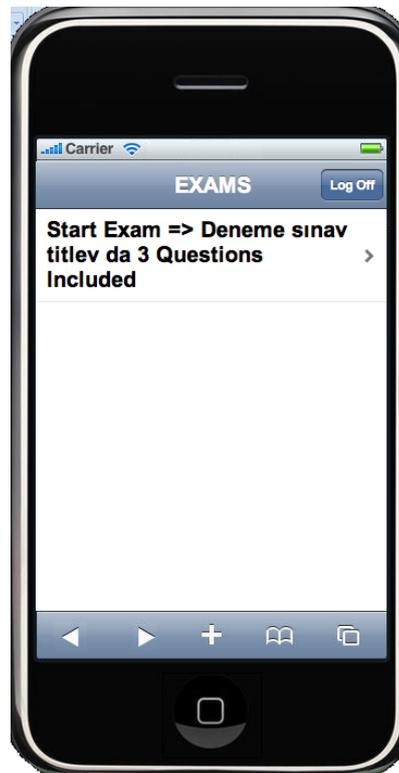


Figure 9.2.20 Exam list on iPhone

If exam is available, student can start the exam for particular time.

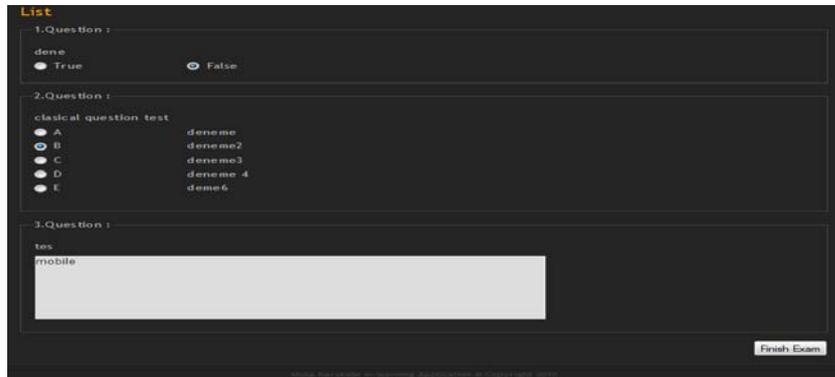


Figure 9.2.21 Exam on pc

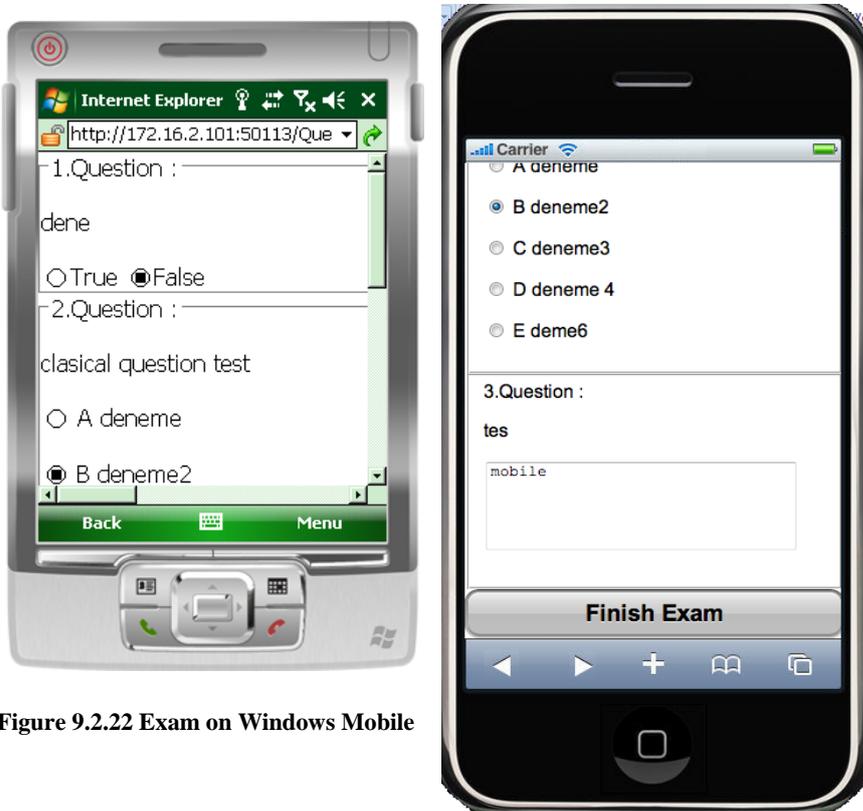


Figure 9.2.22 Exam on Windows Mobile

Figure 9.2.23 Exam on iPhone

Student sees question by starting exam, signs questions with own responses and sends responses to m-learning system by clicking 'Finish Exam' button.

CONCLUSION

Adaptive mobile learning technologies are going to be particularly useful for those who have traditionally pursued distance education methods. Although their inherent portable and mobile nature makes them highly useful for various interactive and dynamic activities, not everyone is going to fancy the limitations of small screen, limited bandwidth and difficult input methods. One has to be careful not to expect too much from mobile learning technologies as has been the case in the past with many other newly-emerged technologies. However, if used sensibly, adaptive mobile learning technologies have the potential to revolutionise distance education by bringing the concept of anytime and anywhere to reality.

REFERENCES

- Agile Manifesto. Agile Software Development Principles, 2001
<http://www.agilemanifesto.org/principles.html>
- Bacon, Benjamin. PIIM Research. Web 2.0 System Architecture Guidelines. PIIM The New School October 30, 2008
- Bradley, Rick. Evaluation: moving from Java to Ruby on Rails for the CenterNet rewrite. 2006. http://rewrite.rickbradley.com/pages/moving_to_rails.html
- Burbeck, Steve. Applications Programming in Smalltalk-80: How to use Model-View-Controller, 1992. <http://st-www.cs.uiuc.edu/users/smarch/st-docs/mvc.html>
- Chen, Y.S., Kao, T.C., Sheu, J.P. and Chiang, C.Y. (2002). A Mobile Scaffolding-Aid-Based Bird-Watching Learning System. In M. Milrad, H. U. Hoppe and Kinshuk (Eds.), IEEE International Workshop on Wireless and Mobile Technologies in Education (pp. 15-22). Los Alamitos, USA: IEEE Computer Society.
- Christensen, Erik. Curbera, Francisco. Meredith, Greg. Weerawarana, Sanjiva. Web Services Description Language (WSDL) 1.1. 15 March 2001 <http://www.w3.org/TR/wsdl>
- Kay, Russell. Computerworld, Representational State Transfer (REST), August 6, 2007, http://www.computerworld.com/s/article/297424/Representational_State_Transfer_REST_
- Costello, Roger L.. Building Web Services the REST Way. 2002. <http://www.xfront.com/REST-Web-Services.html>
- eNode, Inc. Markup Language. 2002. <http://www.enode.com/x/markup/tutorial/mvc.html>
- Evri . Virtual Database EMR. <http://www.evri.com/product/virtual-database-emr-0xcd0ab>
- Fielding, Roy Thomas. Architectural Styles and the Design of Network-based Software Architectures. 2000. <http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>
- Fitzpatrick, Brad. Distributed Caching with Memcached. 1 August 2004. <http://www.linuxjournal.com/article/7451>
- Fowler, Martin. Inversion of Control Containers and the Dependency Injection pattern. 23 January 2004. <http://www.martinfowler.com/articles/injection.html>
- Fowler, Martin. Using an Agile Software Process with Offshore Development. 08 July 2006. <http://www.martinfowler.com/articles/agileOffshore.html>
<http://www.martinfowler.com/articles/injection.html>
- Fowler, Martin. The New Methodology (Agile). 13 December 2005. <http://martinfowler.com/articles/newMethodology.html>
- Garrett, Jesse James. Ajax: A New Approach to Web Applications. 18 February 2005. <http://www.adaptivepath.com/ideas/essays/archives/000385.php>

Hunt, Andrew, Thomas, Andrew. *The Pragmatic Programmer: From Journeyman to Master*. Addison- Wesley, October 1999

iUI, User Interface (UI) Framework for WebApp development on iPhone-class devices,2009 August. <http://code.google.com/p/iui/>

Kinshuk & Goh T. T. (2003). Mobile Adaptation with Multiple Representation Approach as Educational Pedagogy. In Uhr W., Esswein W. & Schoop E. (Eds.), *Wirtschaftsinformatik 2003 - Medien - Märkte - Mobilität* (pp. 747-763), Heidelberg, Germany: Physica-Verlag.

Lehner, F. and Nosekabel, H. (2002). The Role of Mobile Devices in E-learning -First Experience with a E-learning Environment. In M. Milrad, H. U. Hoppe and Kinshuk (Eds.), *IEEE International Workshop on Wireless and Mobile Technologies in Education* (pp. 103-106). Los Alamitos, USA: IEEE Computer Society.

Lin T., Kinshuk, & Patel A. (2003). Cognitive Trait Model for Persistent Student Modelling. In D. Lassner & C. McNaught (Eds.), *EdMedia 2003 Conference Proceedings* (pp. 2144-2147), Norfolk, USA: AACE.

Luchini, K., Curtis, M., Quintana, C., & Soloway, E. (2002). Supporting Learning in Context: Extending Learner-Centered Design to the Development of Handheld Educational Software. In M. Milrad, H. U. Hoppe and Kinshuk (Eds.), *IEEE International Workshop on Wireless and Mobile Technologies in Education* (pp. 107-111). Los Alamitos, USA: IEEE Computer Society.

Merrill, Christopher. Performance Impacts on AJAX Development. Using AJAX to Improve Bandwidth Performance of Web Applications. 15 January 2006. <http://www.webperformanceinc.com/library/reports/AjaxBandwidth/>

Microsoft ASP.net . AJAX. <http://www.asp.net/ajax>

MSDN. Model-View-Controller. <http://msdn.microsoft.com/en-us/library/ff649643.aspx>

MSDN ,ADO.NET Data Services Terminology.<http://msdn.microsoft.com/en-us/library/cc668803%28VS.90%29.aspx>

MSDN ,Browser Definition File Schema. <http://msdn.microsoft.com/en-us/library/ms228122.aspx>

MSDN , Entity Framework Overview. <http://msdn.microsoft.com/en-us/library/bb399567.aspx>

Pautasso, Cesare. Zimmermann, Olaf. Leymann, Frank. *RESTful Web Services vs. "Big" Web Services: Making the Right Architectural Decision*. 2008.

<http://www.jopera.org/files/www2008-restws-pautassozimmermann-leymann.pdf>;

<http://www.jopera.org/docs/publications/2008/restws>

Raymond, Eric. *The Art of Unix Programming*. Addison-Wesley, October 2003

Riehle, Dirk. *A Comparison of the Value Systems of Adaptive Software Development and Extreme Programming: How Methodologies May Learn From Each Other Appeared in Extreme Programming Explained*. G. Succi and M. Marchesi, ed. Boston. Addison-Wesley. 2001

Reenskaug, Trygve. *Models-Views-Controllers*. 10 December 1979.

<http://heim.ifi.uio.no/~trygver/1979/mvc-2/1979-12-MVC.pdf>

Ruby on Rails Community. *Ruby on Rails*. <http://www.rubyonrails.org/>

Sharples, M. (2000). The design of personal mobile technologies for lifelong learning. *Computers & Education*, 34, 177-193.

Smith, J., Mohan, R., & Li, C. (1999). Scalable multimedia delivery for pervasive computing. *ACM Multimedia*. www.ee.columbia.edu/~jrsmith/html/pubs/acmmm99.pdf.

Stephens, Matt. Rosenberg, Doug. *Extreme Programming Refactored. The case against XP*. Apress. September 2008.

Steve Smith. (DRY) Don't Repeat Yourself. <http://stevesmithblog.com/blog/don-repeat-yourself/>

Tilkov, Stefan. *A Brief Introduction to REST*. 10 December 2007.

<http://www.infoq.com/articles/restintroduction>

Tilkov, Stefan. *Addressing Doubts about REST*. 13 March 2008.

<http://www.infoq.com/articles/tilkovrest-doubts>

Venners, Bill. *Don't Live with Broken Windows: A Conversation with Andy Hunt and Dave Thomas, Part I*. 3 March 2003. <http://www.artima.com/intv/fixit.html>

Vavoula, G. N. and Sharples, M. (2002). KleOS: A personal, mobile, Knowledge and Learning Organization System. In M. Milrad, H. U. Hoppe and Kinshuk (Eds.), *IEEE International Workshop on Wireless and Mobile Technologies in Education* (pp. 152-156). Los Alamitos, USA: IEEE Computer Society.

Venners, Bill. *Orthogonality and the DRY Principle: A Conversation with Andy Hunt and Dave Thomas, Part II*. 10 March 2003.

Wan, X. D. (2003). *Statistics Of China Mobile Phone Sales & Production 2002* (in Chinese). www.mii.gov.cn/mii/hyzw/tongji/2003-03-0702.htm

W3C. Simple Object Access Protocol (SOAP) 1.1. <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>

W3shool:XML tutorial <http://www.w3schools.com/xml/default.asp>