|  |  |
| --- | --- |
| **Logo_FPT_University_doc** | **MINISTRY OF EDUCATION AND TRAINING** |

|  |
| --- |
| **FPT UNIVERSITY** |
| Capstone Project Document |
| E-learning website |
|  |
| |  |  | | --- | --- | | **Unicorn** | | | **Group Members** | |  |  |  | | --- | --- | --- | | -NguyễnTuấn Minh | 01135 | Leader | | -NguyễnHồng Nam | 00903 | Member | | -NguyễnTiếnHuy | 01016 | Member | | - Nguyễn Minh Hiếu | 00684 | Member | | -NguyễnTrườngHưng | 00838 | Member | | | **Supervisor** | NguyễnTấtTrung | | **Ext Supervisor** | HuỳnhAnhDũng | | **Capstone Project code** | E-learning | |
|  |

|  |
| --- |
| Hanoi,29 July 2012 |

Record of change

\*A - Added M - Modified D - Deleted

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Effective Date | Changed Items | A\* M, D | Change Description | New Version |
| 15/5 | Create new | A |  | V1.0 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

TABLE OF CONTENTS

[Definitions and Acronyms 8](#_Toc332418874)

[1.1. Background 9](#_Toc332418875)

[1.2. Purpose 9](#_Toc332418876)

[1.3. Literature review 9](#_Toc332418877)

[1.3.1. Overview of existing system 9](#_Toc332418878)

[1.3.2. Achievements of the existing system 12](#_Toc332418879)

[1.3.3. Limitations of existing system 13](#_Toc332418880)

[1.4. Idea 13](#_Toc332418881)

[1.4.1. Home page “Trang chủ” 13](#_Toc332418882)

[1.4.2. News “Tin tức” 14](#_Toc332418883)

[1.4.3. Theories “Lý thuyết” 14](#_Toc332418884)

[1.4.4. Exercise “Thư viện bài tập” 15](#_Toc332418885)

[1.4.5. Self-study exams “Đề tự luyện” 15](#_Toc332418886)

[1.4.6. Exam online “Thi trực tuyến” 16](#_Toc332418887)

[1.4.7. Forum “Diễn đàn” 16](#_Toc332418888)

[1.4.8. Member “Thành viên” 16](#_Toc332418889)

[1.5. Proposal 17](#_Toc332418890)

[2.1 . Problem Definition 19](#_Toc332418891)

[2.1.1. Name of this Capstone Project 19](#_Toc332418892)

[2.1.2. Problem Abstract 19](#_Toc332418893)

[2.1.3. Project Overview 20](#_Toc332418894)

[2.2. Project organization 26](#_Toc332418895)

[2.2.1. Software Process Model 26](#_Toc332418896)

[2.2.2. Roles and Responsibilities 28](#_Toc332418897)

[2.2.3. Tools and Techniques 34](#_Toc332418898)

[2.3. Project management plan 35](#_Toc332418899)

[2.3.1. Tasks: 35](#_Toc332418900)

[2.3.2. Task Sheet: Assignments and Timetable 38](#_Toc332418901)

[2.3.3. Risk management plan 39](#_Toc332418902)

[2.3.4. All Meeting Minutes 40](#_Toc332418903)

[2.4. Coding Convention 41](#_Toc332418904)

[- General 41](#_Toc332418905)

[- Indentation 41](#_Toc332418906)

[- Maximum Line Length 42](#_Toc332418907)

[- Line Termination 42](#_Toc332418908)

[- Character encoding 42](#_Toc332418909)

[- Spelling 42](#_Toc332418910)

[- Classes 42](#_Toc332418911)

[- Filenames 43](#_Toc332418912)

[- Functions and Methods 43](#_Toc332418913)

[Project Success Criteria 46](#_Toc332418914)

[3.1. User Requirement Specification 46](#_Toc332418915)

[3.2. System Requirement Specification 48](#_Toc332418916)

[3.2.1. External Interface Requirements 48](#_Toc332418917)

[3.2.2. System Features 50](#_Toc332418918)

[3.2.3. Software System Attributes 84](#_Toc332418919)

[3.3. Entity Relationship Diagram or Data Structures 84](#_Toc332418920)

[4.1. Design Overview 85](#_Toc332418921)

[4.2. System Architectural Design 85](#_Toc332418922)

[4.2.1. Choice of System Architecture 85](#_Toc332418923)

[4.2.2. Description of System Interface 93](#_Toc332418924)

[4.3. Component Diagram 94](#_Toc332418925)

[4.4. Detailed Description of Components 95](#_Toc332418926)

[4.4.1. Theory library 95](#_Toc332418927)

[4.4.2. Exercise library 107](#_Toc332418928)

[4.4.3. Open sources integration 115](#_Toc332418929)

[4.5. Database Design or Data Structures 126](#_Toc332418930)

[4.5.1. Detailed database design for “E-Learning” system: 126](#_Toc332418931)

[4.5.2. Table and columns description and explanation 127](#_Toc332418932)

[5.1. Introduction 132](#_Toc332418933)

[5.1.1. Purpose 132](#_Toc332418934)

[5.1.2. Reference 132](#_Toc332418935)

[5.2. Test plan 132](#_Toc332418936)

[5.2.1. Feature to be tested 132](#_Toc332418937)

[5.2.2. Feature not to be tested 136](#_Toc332418938)

[5.3. Test strategy 137](#_Toc332418939)

[5.3.1. Test types 137](#_Toc332418940)

[5.3.1.1. Function Testing 137](#_Toc332418941)

[5.3.1.2. User Interface Testing 138](#_Toc332418942)

[5.3.1.3. Performance testing 139](#_Toc332418943)

[5.3.1.4 Security and Access Control Testing 140](#_Toc332418944)

[5.3.2. Test environment 142](#_Toc332418945)

[5.3.3. Risk list 142](#_Toc332418946)

[5.4. Deliverables 143](#_Toc332418947)

[5.5. Test cases 143](#_Toc332418948)

[5.5.1. Home site 143](#_Toc332418949)

[5.5.2. Forum site 155](#_Toc332418950)

[5.6. Test Summary Report - 157 -](#_Toc332418951)

[5.6.1. Text execution summary - 157 -](#_Toc332418952)

[5.6.2. Text results - 158 -](#_Toc332418953)

[5.6.2.1. Summary of Incident Reports Raised During Project - 158 -](#_Toc332418954)

[5.6.2.2. Action Plan for Outstanding Incidents - 158 -](#_Toc332418955)

[5.6.3. Conclusion - 158 -](#_Toc332418956)

[5.6.3.1. System Test Success Criteria - 158 -](#_Toc332418957)

[5.6.3.2. Conclusion - 159 -](#_Toc332418958)

[6.1. Installation guide - 159 -](#_Toc332418959)

[6.1.1. System requirements - 159 -](#_Toc332418960)

[6.1.2. Installing “E-Learning-Website” - 159 -](#_Toc332418961)

[6.1.2.1. Installing Wamp Server - 159 -](#_Toc332418962)

[6.1.2.2. Uploading “E-Learning-Website” files to web host - 159 -](#_Toc332418963)

[6.1.2.3. Creating database and a user for “E-Learning-Website” - 159 -](#_Toc332418964)

[6.1.2.4. Browsing to “E-Learning-Website” administrator homepage - 159 -](#_Toc332418965)

[6.1.2.5. Browsing to “E-Learning-Website” homepage - 159 -](#_Toc332418966)

[6.2. User’s guide - 159 -](#_Toc332418967)

[6.3. Other - 159 -](#_Toc332418968)

[Reference - 160 -](#_Toc332418969)

# Definitions and Acronyms

|  |  |  |
| --- | --- | --- |
| Acronyms | Definition | Note |
| E-Learning | Electronic Learning |  |
| ID | Intrusion Detection |  |
| PDF | Portable Document Format |  |
| E-mail | Electronic mail |  |
| WAMP | Name of web development environment |  |
| PHP | Programing language |  |
| MySQL | Name of the database server |  |
| RAM | Random Access Memory |  |
| CPU | Central Processing Unit |  |
| SQL | Structured Query Language |  |
| Apache | Web server for development of web application |  |
| UC | Use case |  |
| UI | User Interface |  |
| URL | Universal Resource Locator |  |
| WWW | World Wide Web |  |
| IDE | Integrated Development Environment |  |
| HTML | Hypertext Markup Language |  |
| CSS | Cascading Style Sheets |  |
| JS | Javascript |  |
| HTTP | Hypertext Transfer Protocol |  |
| phpBB | PHP Bulletin Board |  |
| AT | Acceptance test |  |
| IT | Integration test |  |
| PM | Project Manager |  |
| PTL | Project Technical Leader |  |
| QA | Quality Assurance |  |
| SRS | Software Requirement Specification |  |
| ST | System test |  |
| TP | Test Plan |  |
| TC | Test Case |  |
| TR | Test Report |  |
| UAT | User Acceptance test |  |
| UT | Unit test |  |
| AT | Acceptance test |  |
| IT | Integration test |  |

I> Report No.1: Introduction

# 1.1. Background

Nowadays, e-learning has become commonly in many countries around the world. In Vietnam, with the development of Internet, e-learning is really necessary to students to get more flexible, convenient… in receiving knowledge.

Simpler than going to school or somewhere in which students must pay fee, they only need to have a computer which connected to internet, they don’t need to go anywhere and learning cost will be cheaper if they study online. They can get more information, materials about special subject. For example, a student wants to learn trigonometry in grade 11, beside some theory from text book, he can get some interesting information about history of trigonometry from Wikipedia. He also finds out many forms of exercises (with solutions or not) to train himself.

# 1.2. Purpose

This website is created to meet two goals:

* Complete the capstone project of FPT-University.
* Create an e-learning website that can support Vietnamese students in learning, reviewing…

# 1.3. Literature review

## 1.3.1. Overview of existing system

In this chapter, we will show some e-learning websites that are available in Vietnam. Each of them has some advantages and disadvantages. So you can understand clearly about the development of e-learning in Vietnam. They Include:

* *hocmai.vn*
* *moon.vn*
* Some other websites: *truongvietnam.vn*, *onthionline.net*, *truongtructuyen.vn,thaytro.vn*…

We will analyze two websites: *hocmai.vn* and *moon.vn* because they are websites which are familiar and popular.

* **hocmai.vn**

This website was established in 2007. Nowadays it is the biggest e-learning forum in Vietnam. It makes profit by selling more than 2000 videos, courses which are built and taught by famous experience teachers.



Figure 1.1: Hocmai’sWebsite[1]

* **Advantages:**
* Essence of hocmai.vn is a forum that help student share and exchange knowledge about subjects (math, literature, English…).
* *hocmai.vn* has more than 2000 videos about courses. Student can learn online by watching these video and theory files with .pdf extension.

[1] **hocmai.vn**

* With *hocmai.vn*, not only students but also their parents can log in and watch for learning of students.
* **Disadvantages:**
* The databases of hocmai.vn focus on videos of courses to make profit. So its exercises, theories are updated rarely.
* Forum of *hocmai.vn* is very large with many subjects of many grades. So it is difficult to use, manage, update.
* *hocmai.vn* is used for advertising of offline learning operation. So it does not focus on e-learning completely.
* **moon.vn**

This website is established in 2008. It makes profit by focusing on selling examination products. Most of exercises on moon.vn are multiple choices.



Figure 1.2 Moon’s Website [2]

[2] **moon.vn**

* **Advantages:**
* *moon.vn* focus on examination. So that, its functions support taking exam online.
* Site “Hỏi đáp” is a small forum. It helps students give questions and take answers easier and fast.
* User ranking function “hệ thống level” creates competition between students.
* User (teachers) can give questions, exercises, exams to *moon.vn* to get money in return.
* The function “Đấu trường” is a special attraction of *moon.vn*. Students can make groups and compete together in “Đấu trường”.
* **Disadvantages:**
* The databases of *moon.vn* have been taken from many sources in internet at beginning. So that it is not updated regularly.
* *moon.vn* does not have own good teacher. It also does not have a function to guarantee quality of exercises.
* *moon.vn* does not have many subjects. It just focuses on some important subjects (math, English, chemistry…)

## 1.3.2. Achievements of the existing system

* Big e-learning websites focus on making profit. So they have functions for selling online education products.
* Students have many choices for e-learning:
  + Sharing, exchanging knowledge through forums with support of teachers or senior members.
  + Join in special operation like “Đấu trường” of *moon.vn*.
  + Learning through watching videos like truongtructuyen.vn or *hocmai.vn*.

## 1.3.3. Limitations of existing system

* Small e-learning websites can support free users. But these websites is not developed carefully. These sites are places for some groups of students sharing knowledge, not for all kinds of student.
* Most of databases depend on administrators of websites. So that it is difficult to update or updated slowly.
* There is not a website which supports requirements from many kinds of user. Example: members of *moon.vn* easy to do exams but difficult to learn theories structurally; members of *hocmai.vn* usually learn through videos by paying fee, so free users rarely join in *hocmai.vn*.

# 1.4. Idea

The goal of this project is an e-learning website that supports many kinds of user with many kinds of requirements. It is expected to overcome disadvantages of available e-learning websites. This chapter shows some features that can be developed in this capstone project.

This website has following sites:

## 1.4.1. Home page “Trang chủ”

This site is designed friendly, simple and easy to use. It has some beginning divisions: Logo, banner and slogan, menu bar, hot topics, news, login form, search bar, subjects (divide by grades or by topics – this will be described later in this chapter) and some other information.

Menu bar include links to other sites. This menu is common to other sites of this website:

* Home page “Trang chủ”
* News “Tin tức”
* Theories “Lý thuyết”
* Exercises “Thư viện bài tập”
* Self-study exams “Đề tự luyện”
* Exam online “Thi trực tuyến”
* Forum “Diễn đàn”
* Member “Thànhviên”

***Main functions*:** Login, link to other sites, show some main information: hot topics, subjects…

## 1.4.2. News “Tin tức”

This site includes some news about education: Information of schools, Information of entrance exam, university ranking…

***Main function:*** Show news.

## 1.4.3. Theories “Lý thuyết”

This site provides theories of subject with two kinds: grade and topic.

* **Grade theory:**
* When a student chooses this, he can learn basic theories depend on which grade he is in. Example, student Nguyen Tuan Minh (grade 11) wants to learn geometry, he can chooses “Lý thuyết” site => Grade 11 “Lớp 11” =>Subject math “Toán” => Geometry “Hình học”. After that, he can learn theories about geometry of grade 11 and get some refer exercises below.
* Grade theory is developed to focus on supporting students who want to learn new knowledge.
* **Topic theory:**
* When a student chooses this, he can learn collective theories about concrete topic. Example, student Nguyen Hong Nam (grade 12) wants to learn knowledge about organic chemistry that collected from three grades (10,11,12), he can chooses “Lý thuyết” site => Topic “Chuyên đề” => Subject chemistry “Môn hóa” => Organic chemistry “Hóa hữu cơ”. After that, he can learn theories about organic chemistry collectively and get some refer exercises below.
* Topic theory is developed to focus on supporting students who want to take university exams.

***Main function:*** Show theories and knowledge of subjects and refer exercises.

## 1.4.4. Exercise “Thư viện bài tập”

This site provides many kinds of exercises, questions and answers for students. Exercise can be divided by grades, subjects or students can do collective knowledge exercise here. It is different from theories site in which, students just can do basic and simple exercises.

***Main functions:***

* Show exercises from databases. Students can do multiple choice exercises by tick answers.
* Students can choose level of exercises (from simple to difficult) and estimate level of an exercise.

## 1.4.5. Self-study exams “Đề tự luyện”

This site provides many kinds of exams for students.

To do an exam, students can choose exam from library of the system. There is a built examination library that includes many exams of many subjects. Students can choose one of them to self-study. An exam has some important attributes: time, number of question, level of questions.

***Main function:*** Allow student choose exam’s attributes, show exam’s questions and time, count down and finish exam when user agree to finish or time up, show mark after user finish the exam.

## 1.4.6. Exam online “Thi trực tuyến”

Functions of this site are different from self-study exams.

In this site, administrators or people with authority can give special exams. These exams are informed before about start time, end time, number of questions. And when an exam is started, students can take the exam in beginning 15 minutes and like self-study exam. After exam finishes, mark of examinees will be sent to each student and some of them can take the next one (round 2, 3, 4…) if they have good mark. Exam can also be taken by group of users.

***Main functions:*** Show questions, time about exam; filter users to take next exam; support taking exam by group of users.

## 1.4.7. Forum “Diễn đàn”

This site is a forum in which, members can share and exchange knowledge about subjects, education information.

***Main function:*** Show topics, boxes like a normal forum.

## 1.4.8. Member “Thành viên”

This site helps a student manage individual information: e-mail, name, age, marks, birthday…

# 1.5. Proposal

We will list some functions that can be developed if we have enough time. There following are functions which are different from other e-learning websites here.

**1.5.1. Comment in comment**

In common forums, a comment can have a quotation of other comments. This is often inconvenient if a topic has many comments. This function is developed to help a user track his comment easily when every answers of his comment can be displayed in one form without quotation.

Following picture is an example of this function:



Figure 1.3 comment example

**1.5.2. Log in and register**

This website provides two ways for new member to register:

* Use open ID (account Facebook)
* Register directly in this website

**1.5.3. Exercise ranking**

Student can estimate the level (from simple to difficult)

**1.5.4. Databases updated by user**

This function is developed to allow users who have authority to input new exercises or knowledge into databases. This function help the website have databases that are updated regularly.

**1.5.5. Formula**

This function helps the website operate with storing, displaying formula of math, chemistry, and physics.

**1.5.6. Intelligent user identification**

This function helps students learn more easily in the website.

When a student login to the website, by some information, we can provide some suggestion to that user.

Example: When a student login to the website, he is grade 11. We get grade information and display subjects, topics relate to knowledge of grade 11 to him. So he can be easy to get what he wants. Besides, he also can get other options, he can do exercises of grade 10 if he wants, just click on grade 10 in “Exercise” site.

II> Report No.2: Software Project Management Plan

# . Problem Definition

## 2.1.1. Name of this Capstone Project

This Capstone Project’s name is E-learning.

## 2.1.2. Problem Abstract

Nowadays online services are developing quickly in Viet Nam; one of them is online studying. It has been common in development countries, there are many famous schools and training centers used this service. For example, granting Java’s certificates to trainees by Oracle, grating Animation Mentor to trainees by Animation Mentor School… Trying to imagining in the next few years, form of online studying and taking an exam will become commonly, changing form of normal studying as going to school, training centers, and students can sit in front of a computer which connects to internet and study like sitting in class. The tool will provide lectures as video or text, after each of period, students which participate at the period will be provided exercises concern to the period content. The benefit of the form bring out: lectures which they registered will be saved in their account, they can watch again many time if having one problem isn’t clear yet. They can send mail to teacher which taught the period to have better supporting. Moreover, we create favorable conditions for students can take an online exam on our website .We will make a forum is creative and self-motivated to help members are able to exchange information and learn in groups effectively. The website will emphasize on bring comfortable and user-friendly, users will be familiar at the first time accessing the website, simple and elegance in design but effective in functions. We want to guide Vietnamese to use internet such as a tool which serves effectively their study, making an individual environment to practice and really self-motivated.

## 2.1.3. Project Overview

**2.1.3.1. The Existing System**

Most of existing systems are concentrating on paying fees users. Below are the existing methods that these websites about studying online:

- Collecting data:

Nowadays, most of websites focus on studying and preparing for a university entrance examination’s objects. These websites collect data under 2 ways:

+ Collecting information about theories and exercises of subjects on internet. They usually concentrate at learning forums.

+ Working with large organizations, training centers to create courses. Since then, members can register to learn.

+ Working with illustrious teachers: They will make lectures as video and provide exercises concern.

- Online Course:

Most of courses are video. Members must purchase to watch the courses. These courses are divided into theories.

- Preparing exam:

Most of them are preparing for a university entrance examination. They are divided into many courses, each of course talk about a part of knowledge in exam question. After a course finished, students will be provided forms of exercises concern the learning.

- Exercises Library:

Using data which is collected by many sources: schoolbook, learning forums, teachers…They are saved by .pdf files. Students can download and do on their computer.

Below are the limitations of the existing systems:

- Data: exercises Libraries are old, content of lectures aren’t visual, most of lectures are video, they don’t update frequently.

- Forums: the operations aren’t effective, don’t be cared yet, there are a few members.

**2.1.3.2. The Proposed System**

Because there are restrictions on capstone project completion time and resources involved, we will choose open sources to develop the system. We suggest popular open sources:

* **Joomla** is a [free](http://en.wikipedia.org/wiki/Free_software) and [open source](http://en.wikipedia.org/wiki/Open_source) [content management framework](http://en.wikipedia.org/wiki/Content_management_framework) (CMF) for publishing content on the [World Wide Web](http://en.wikipedia.org/wiki/World_Wide_Web) and [intranets](http://en.wikipedia.org/wiki/Intranet) and a [model–view–controller](http://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller) (MVC) [Web application framework](http://en.wikipedia.org/wiki/Web_application_framework) that can also be used independently.
* **Phpbb** is a popular [Internet forum](http://en.wikipedia.org/wiki/Internet_forum) package written in the [PHP](http://en.wikipedia.org/wiki/PHP) [scripting language](http://en.wikipedia.org/wiki/Scripting_language). Available under the [GNU General Public License](http://en.wikipedia.org/wiki/GNU_General_Public_License), phpBB is [free and open source software](http://en.wikipedia.org/wiki/Free_and_open_source_software).
* **iGiveTest** is a comprehensive solution for creating, administering, and providing thorough analysis of tests on the Internet and Intranet.

By using the open sources, we can develop the website with main functions. The website will include sites below:

|  |  |
| --- | --- |
| Priority | Site |
| 1 | Home page “Trang chủ” |
| 2 | Theories “Lý thuyết” |
| 3 | Exercise “Thư viện bài tập” |
| 4 | Self-study exams “Đề tự luyện” |
| 5 | Forum “Diễn đàn” |
| 6 | Member “Thành viên” |
| 7 | News “Tin tức” |
| 8 | Exam online “Thi trực tuyến |

* **Home page “Trang chủ”**

This site has main functions: login, link to other sites, show some main information: hot topics, subject and news during the day…

The page will include some components: logo, banner and slogan, menu bar, hot topics (it shows some outstanding news which we choose according to each of day), news, login form, and search bar. At the front page, we put subjects’ module which is divided by grade or by topics, and news about education, training, and examinations… Some functions will be added to the page after finishing Capstone Project, such as ranking schools, mark of university’s exam…

Menu bar include links to other sites. They are common sites of the website:

* Home page “Trang chủ”
* News “Tin tức”
* Theories “Lý thuyết”
* Exercise “Thư viện bài tập”
* Self-study exams “Đề tự luyện”
* Exam online “Thi trực tuyến”
* Forum “Diễn đàn”
* Member “Thành viên”
* **Theories “Lý thuyết”**

The site has main function: show theories and knowledge of subjects and refer exercises.

This site provides theories of subject with two kinds: grade and topic. They are particular features:

* Grade theory:
  + When a student chooses this, he can learn basic theories depend on which grade he is in.
  + Grade theory is developed to focus on supporting students who want to learn new knowledge.
* Topic theory:
* When a student chooses this, he can learn collective theories about concrete topic.
* Topic theory is developed to focus on supporting students who want to take university exams.
* **Exercise “Thư viện bài tập”**

The site has two main functions:

+ Showing exercises from databases. Students can do multiple choice exercises as the system provide by ticking answers.

+ Students can choose level of exercises (from simple to difficult) and estimate level of an exercise.

The site provides many kinds of exercises, questions and answers for students. Exercises can be divided by grades, subjects or students can do collective knowledge exercise here. It’s different from theories site in which, students just can do basic and simple exercises.

* **Self-study exams “Đề tự luyện”**

The site allows student choose exam’s attributes, show exam’s questions and time, count down and finish exam when user agree to finish or time up, show mark after user finish the exam.

The site provides many kinds of exams for students. To do an exam, students can choose exam from library of the system. There is a built examination library that includes many exams of many subjects. Students can choose one of them to self-study. An exam has some important attributes: time, number of question, level of questions.

* **Forum “Diễn đàn”**

This site is a place which members can share experience, discuss, exchange some problems need care. It provides helpful information for members. We will create an environment which is creative and self-motivated for students.

* **Member “Thành viên”**

This site helps a student manage individual information: e-mail, name, age, marks, birthday…

* **News “Tin tức”**

This site includes some news about education: Information of schools, Information of entrance exam, university ranking… News will be updated during the day.

* **Exam online “Thi trực tuyến”**

At certain hours, we will create some exams about subjects and add authority to members which are appropriate to do.

Functions of this site are different from self-study exams. In this site, administrators or people with authority can give special exams. These exams are informed before about start time, end time, number of questions. And when an exam is started, students can take the exam in beginning 15 minutes and like self-study exam. After exam finishes, mark of examinees will be sent to each student and some of them can take the next one (round 2, 3, 4…) if they have good mark. Exam can also be taken by group of users.

**2.1.3.3. Boundaries of the System**

To finish the project on time and successfully, we present our goals will focus on these core functions:

* Developing theories site and exercises site.
* Exam online site.

Firstly, we choose the highest priority function that needs to be developed is “Theories site and exercises site”, the next will exam online site.

Later, when these core functions are successfully developed, we will continue to develop, integrate the rest functions to complete the idea of the whole group which includes:

* Administrator site
* Forum site
* News site
* Functions which are different with other e-learning website here.

*The future vision for our system:*

We have the ambition to develop an e-learning website which can satisfy most of Vietnamese students in learning. Making good weakness of e-learning websites now. Students don’t need to go anywhere and learning cost will be cheaper. We wish to bring highest benefits for not only students but also teachers

**2.1.3.4. Development Environment**

**2.1.3.4.1 Software environment**

* Operating system: Windows XP/Windows 7
* WAMP server
* Apache Tomcat version 2.2.21
* PHP version 5.3.10
* MySQL version 5.5.20
* Open source Frameworks

**2.1.3.4.2 Hardware environment**

Required (minimum) Specifics

+ CPU: Intel Pentium(R) 4 2.4 GHz or better supported

+ RAM: 512MB

+ Hard Drive: 2GB or free space

Recommended Specifics

+ CPU: Dual core 2.4GBz or better supported

+ RAM: 1GB for Windows XP, 2GB for Windows 7

+ Hard Drive: 2GB of free space

# 2.2. Project organization

## 2.2.1. Software Process Model

In the Capstone Project, we choose Iterative Model because an iterative lifecycle model does not attempt to start with a full specification of requirements. Instead, development begins by specifying and implementing just part of the software, which can then be reviewed in order to identify further requirements. This process is then repeated, producing a new version of the software for each cycle of the model. Consider an iterative lifecycle model which consists of repeating the following four phases in sequence:



**Figure 2.1**: iterative model [3]

- A ***Requirements*** phase, in which the requirements for the software are gathered and analyzed. Iteration should eventually result in a requirements phase that produces a complete and final specification of requirements.

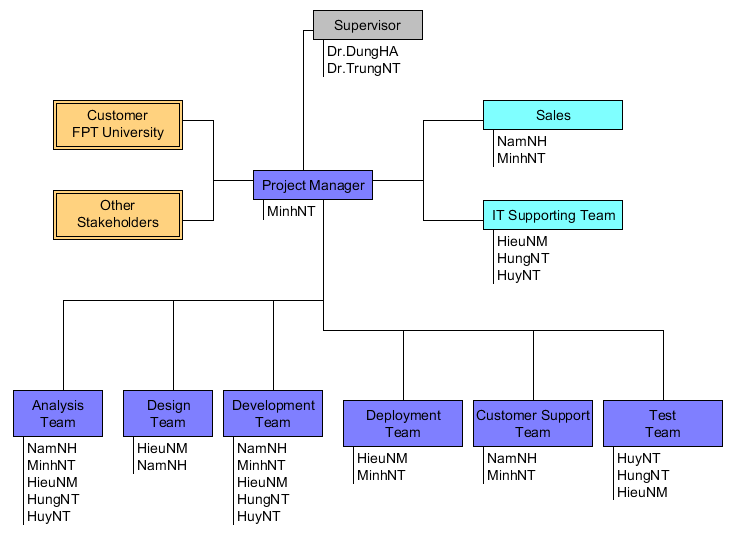
- A ***Design*** phase, in which a software solution to meet the requirements is designed. This may be a new design, or an extension of an earlier design.  
- An ***Implementation and Test*** phase, when the software is coded, integrated and tested.

[3] <http://www.arctern.com/uploadedimages/iterative-model.jpg>

- A ***Review*** phase, in which the software is evaluated, the current requirements are reviewed, and changes and additions to requirements proposed.  
For each cycle of the model, a decision has to be made as to whether the software produced by the cycle will be discarded, or kept as a starting point for the next cycle (sometimes referred to as incremental prototyping). Eventually a point will be reached where the requirements are complete and the software can be delivered, or it becomes impossible to enhance the software as required, and a fresh start has to be made.  
The iterative lifecycle model can be likened to producing software by successive approximation.

Drawing an analogy with mathematical methods that use successive approximation to arrive at a final solution, the benefit of such methods depends on how rapidly they converge on a solution.

## 2.2.2. Roles and Responsibilities



**Figure 2.2**: Project organization in “E-Learning” project

|  |  |
| --- | --- |
| Title | Roles |
| Project Manager | Responsible for developing, in conjunction with the Project Sponsor, a definition of the project. The Project Manager then ensures that the project is delivered on time, to budget and to the required quality standard (within agreed specifications). He/she ensures the project is effectively resourced and manages relationships with a wide range of groups (including all project contributors).  The Project Manager is also responsible for managing the work of consultants, allocating and utilizing resources in an efficient manner and maintaining a co-operative, motivated and successful team. |
| Responsibilities | |
| * Managing and leading the project team. * Recruiting project staff and consultants. * Managing co-ordination of the partners and working groups engaged in project work. * Developing and maintaining a detailed project plan. * Managing project deliverables in line with the project plan. * Recording and managing project issues and escalating where necessary. * Resolving cross-functional issues at project level. * Managing project scope and change control and escalating issues where necessary. * Monitoring project progress and performance. * Providing status reports to the Project Sponsor. * Managing project training within the defined budget. * Liaison with, and updates on progress to, Project Steering Board/Senior Management. * Managing project evaluation and dissemination activities. * Managing consultancy input within the defined budget. * Final approval of the design specification. * Working closely with users to ensure the project meets business needs. * Definition and management of the User Acceptance Testing programme. * Identifying user training needs and devising and managing user training programmes. | |

|  |  |
| --- | --- |
| Title | Roles |
| Supervisor | The person who commissions others to deliver the project and champions the cause throughout the project. They will normally be a senior member of staff with a relevant area of responsibility that will be affected by the outcome of the project. They are involved from the start of the project, including defining the project in conjunction with the Project Manager. Once the project has been launched they should ensure that it is actively reviewed. |
| Responsibilities | |
| * Acts as champion of the project. * Is accountable for the delivery of planned benefits associated with the project. * Ensures resolution of issues escalated by the Project Manager or the Project Board. * Sponsors the communications programme; communicates the programme’s goals to the organization as a whole. * Makes key organization/commercial decisions for the project. * Assures availability of essential project resources. * Approves the budget and decides tolerances. * Leads the Project Steering Board. * Ultimate authority and responsibility for the project. | |

|  |  |
| --- | --- |
| Title | Roles |
| Project Team Member | The staff who actively work on the project, at some stage, during the lifetime of the project. This could be further broken down into specific roles as required – such as Project Administrator, etc. |
| Responsibilities | |
| Team member roles will vary depending on the type of project. Typically they might be to:   * Provide functional expertise in an administrative process. * Work with users to ensure the project meets business needs. * Documentation and analysis of current and future processes/systems. * Identification and mapping of information needs. * Defining requirements for reporting and interfacing. * User training. | |

|  |  |
| --- | --- |
| Title | Roles |
| Customer | These are the people who will actually use the deliverables of the project. These people are also involved heavily in the project in activities such as defining business requirements. In other cases, they may not get involved until the testing process. Sometimes you want to specifically identify the user organization or the specific users of the solution and assign a formal set of responsibilities to them, like developing use cases or user scenarios based on the needs of the business requirements. |

|  |  |
| --- | --- |
| Title | Roles |
| Designer | The Designer is responsible for understanding the business requirements and designing a solution that will meet the business needs. There are many potential solutions that will meet the client's needs. The Designer determines the best approach. A Designer typically needs to understand how technology can be used to create this optimum solution for the client. The  Designer determines the overall model and framework for the solution, down to the level of designing screens, reports, programs and other components. He or she also determines the data needs. The work of the Designer is then handed off to the programmers and other people who will construct the solution based on the design specifications. |
| Responsibilities | |
| Typically Designer roles might be to:   * Have a basic understanding of technology in order to know what is or is not possible given certain technology realities. * Quickly and accurately recognize performance/knowledge gaps. * Creativity tempered with an understanding of the intended audience, client culture, and learning preferences. * Understanding of human computer factors and interface design. * Ability and willingness to adapt to a dynamic set of standards and tools. | |

|  |  |
| --- | --- |
| Title | Roles |
| Analysis | The Analyst is responsible for ensuring that the requirements of the business clients are captured and documented correctly before a solution is developed and implemented.  In some companies, this person might be called a Business Analyst, Business Systems Analyst, Systems Analyst or Requirements Analyst. |
| Responsibilities | |
| * Analyzing and understanding the current state processes to ensure that the context and implications of change are understood by the clients and the project team * Developing an understanding of how present and future business needs will impact the solution * Identifying the sources of requirements and understanding how roles help determine the relative validity of requirements * Developing a Requirements Management Plan and disseminating the Plan to all stakeholders * Identifying and documenting all business, technical, product and process requirements * Working with the client to prioritize and rationalize the requirements * Helping to define acceptance criteria for completion of the solution | |

|  |  |
| --- | --- |
| Title | Roles |
| Tester | The Tester role is responsible for the core activities of the test effort, which involves conducting the necessary tests and logging the outcomes of that testing. |
| Responsibilities | |
| * Identifying the most appropriate implementation approach for a given test * Implementing individual tests * Setting up and executing the tests * Logging outcomes and verifying test execution * Analyzing and recovering from execution errors | |

## 2.2.3. Tools and Techniques

- App Server: WAMP server v2.2

- UML Tools: StarUML and Umlet

- IDE: Dreamwaver 8.0

- Design tool: Photoshop CS5

- Microsoft Word 2007

- Microsoft Project 2007

# 2.3. Project management plan

## 2.3.1. Tasks:

**2.3.1.1 Task-1: Planning**

* Description:

Planning to manage the project include human, designing and confirming the project goals and objectives, identifying tasks and how goals will be achieved, quantifying the resources needed and determining budgets and timelines for completion. It also includes managing the implementation of the project plan.

* Deliverables:

Software Project Management Plan (SPMP)

* Resources Needed:

MinhNT

**2.3.1.2 Task-2: System Design**

* Description:

Design functional and non-functional requirement in the top-level diagram.

* Deliverables:
* Software Requirement Specification (SRS)
* Resources Needed:

- NamNH

- MinhNT

- HieuNM

**2.3.1.3 Task-3: Detailed System Design**

* Description:

Design detailed architecture of system, component design, diagrams…

* Deliverables:

Software Design Description (SDD)

* Resources Needed:

- NamNH

- MinhNT

- HieuNM

- HuyNT

- HungNT

* Dependencies and Constraints: SRS
* Risks

**2.3.1.4 Task-4: Code**

* Description:

Using Joomla framework to build the website, creating some of new modules about: exam online and exercises library…

* Deliverables: source code, fully implemented system.
* Resources Needed: NamNH, HieuNM, HuyNT, HungNT
* Dependencies and Constraints:

SRS, System Design

* Risks: time

**2.3.1.5 Task-5: Test Plan**

* Description:

Create test plan for the system including Test Design

* Deliverables:

Software Test Document

* Resources Needed:

MinhNT

* Dependencies and Constraints:

SRS, implemented system

* Risks: time

**2.3.1.6 Task-6: System Testing**

* Description:

Executing test following the test design document and preparing test report

* Deliverables:

Software Testing Report

* Resources Needed:

MinhNT

**2.3.1.7 Task-7: Deliver Software**

* Description:

After fixing all the defects, beautiful source code, Project Manager need build a full package to deliver

* Deliverables:

Software’s package and user’s guideline.

* Resources Needed:

MinhNT

NamNH

HieuNM

## 2.3.2. Task Sheet: Assignments and Timetable

Assignments and Timetable is described as the following:



## 2.3.3. Risk management plan

|  |  |  |  |
| --- | --- | --- | --- |
| No | Description | Avoidance Plan | Contingency Plan |
| 1 | People risk: Team members are ill at critical times in the project. | Training technique for team member from the start and assigning tasks conformably. | Changing other member to work the tasks. |
| 2 | People risk: Team members contradict each other | Through open talk, communication team understands each other better. | Finding out the root cause of the conflict and resolve them. |
| 3 | People risk: Team member lack the skills required: technical skills, soft skills. | Set up training technique for team in appropriate time. Organizing team building, open talk. | Arranging meets to support to team. |
| 4 | Technology risk: Software that used to develop the system is not free, beta version… | Researching information about the software on sources. Buying full version out the supplier. | Using open source software. |
| 5 | Requirement risk: Requirements are not defined clearly, inconsistency in defined SRS. | Make sure that the team goals are totally clear. And finishing SRS as soon as possible to operate other tasks. | Set up meeting to discuss and give the final requirements. |
| 6 | Schedule risks: Overestimated time requires doing tasks. | Consult supervisor to build appropriate schedule, keep track the rate of process in tasks of each member. | If there are critical tasks, leader must add more human resources to finish. If there are normal tasks, team should cancel them to achieve rate of process. |
| 7 | Estimation risks: The size of the project is not estimated. | Divide the project into phrases, arrange priority phrases and develop the first | Consult supervisor to estimate appropriately. |

## 2.3.4. All Meeting Minutes

Each week, we always have a meeting about 2h on Thursday to track rate of progress and to discuss any new issues. All problems will be discussed as brainstorming work and we bring out suggestions and comments. After that, we will report the next works and deadline.

|  |  |  |  |
| --- | --- | --- | --- |
| Name of Organization: Unicorn Team  Purpose of Meeting: Kick-off project and choose team leader for project  Data/Time: Wednesday May 2, 2012  Location: FPT University | | | |
| Topic | **Discussion** | **Action** | **Person Responsible** |
| Kick-off project | Discuss content of project and learn about capstone project | - Background of project  - Discuss project organizational structure  - Discuss of project scope, time and list of action items from meeting | All team members (MinhNT, NamNH, HuyNT, HungNT, HieuNM) |
| Team leader | Choose team leader for project | MinhNT is the leader of project | All team members |
| Team name | Create team name | “Unicorn” is the name of team | All team members |

|  |  |  |  |
| --- | --- | --- | --- |
| Name of Organization: Unicorn Team  Purpose of Meeting: divide works, define roles and responsibilities for team, choose tools and techniques for developing on, and configure environment.  Data/Time: Saturday May 19, 2012  Location: FPT University | | | |
| Topic | **Discussion** | **Action** | **Person Responsible** |
| Divide works | Divide works for each of member: | - MinhNT: analyze system and write documents.  - NamNH: create and design template of website. Design and insert database.  - HieuNM, HungNT, HuyNT: code modules and extensions of website. | All team members |
| Define roles and responsibilities for team | - Leader  - Developer  - Designer | All members of project will have specific roles in each phase of development process. | All team members |
| Tools and techniques | Choose tools and techniques:  - PHP, MySQL  - WAMP server  - Joomla framework  - Phpbb or Vbullettin | The tools and techniques choosed:  - PHP, MySQL  - Joomla framework  - Phpbb | All team members |
| Configure environment | Using Windows 7-32 bit, install tools which need for project | - Install WAMP server and Joomla framework  - Install Dreamwaver 8.0 to code. | All team members |

# 2.4. Coding Convention

**2.4.1. PHP File Formatting**

### General

For files that contain only PHP code, the closing tag ("?>") is never permitted. It is not required by PHP, and omitting it prevents the accidental injection of trailing whitespace into the response.

### Indentation

Indentation should consist of 4 spaces. Tabs are not allowed.

### Maximum Line Length

The target line length is 120 characters. That is to say, developers should strive keep each line of their code under 120 characters where possible and practical. This improves the readability of code.

### Line Termination

Line termination follows the Unix text file convention. Lines must end with a single linefeed (LF) character. Linefeed characters are represented as ordinal 10, or hexadecimal 0x0A.

### Character encoding

Use UTF-8 character encoding for PHP files.

**2.4.2. Naming Conventions**

### Spelling

All names should be spelled correctly. It is easy to spell check a name if you are not sure.

### Classes

Classes should be given descriptive names. Avoid using abbreviations where possible. Class names may only contain alphanumeric characters. Numbers are permitted in class names but are discouraged in most cases. Underscores are not permitted in class names. Class names start with a capital letter. If a class name is comprised of more than one word, the first letter of each new word must be capitalized.

|  |
| --- |
| CentralController  LeaveQuote |

Class Names are usually singular nouns. Use LeaveRequest, not LeaveRequests

### Filenames

PHP class files should have the same name as the class. eg: Class CentralController will be in CentralController.php For all other files, including template files, only alphanumeric characters, underscores, and the dash character ("-") are permitted. Spaces are strictly prohibited.

#### File Extensions

* PHP files: ".php"
* HTML files: ".html"
* Javascript files: ".js"
* CSS files: ".css"

### 

### Functions and Methods

Function names may only contain alphanumeric characters. Underscores are not permitted. Numbers are permitted in function names but are discouraged in most cases.

Function names must always start with a lowercase letter. When a function name consists of more than one word, the first letter of each new word must be capitalized. This is commonly called "camelCase" formatting.

Function and method names should be verbs.

These are examples of acceptable names for functions:

|  |
| --- |
| filterInput()  getElementById()  widgetFactory() |

For object-oriented programming, accessors for instance or static variables should always be prefixed with "get" or "set". Accessors for boolean instance variables should usually be prefixed with "is" and "set".

|  |
| --- |
| getName()  setName($name)  isApproved()  setApproved($approved) |

In implementing design patterns, such as the singleton or factory patterns, the name of the method should contain the pattern name where practical to more thoroughly describe behavior.

For methods on objects that are declared as follow:

|  |
| --- |
| function privateFunction(){  // code goes here  } |

**2.4.3. Coding Style**

#### - Variable Substitution

Variable substitution is permitted using either of these forms:

|  |
| --- |
| $greeting = "Hello $name, welcome back!";  $greeting = "Hello {$name}, welcome back!"; |

For consistency, this form is not permitted:

|  |
| --- |
| $greeting = "Hello ${name}, welcome back!"; |

#### 

#### - String Concatenation

Strings must be concatenated using the "." operator. A space must always be added before and after the "." operator to improve readability:

|  |
| --- |
| $company = 'Zend' . ' ' . 'Technologies'; |

When concatenating strings with the "." operator, it is encouraged to break the statement into multiple lines to improve readability. In these cases, each successive line should be padded with whitespace such that the "."; operator is aligned under the "=" operator:

|  |
| --- |
| $sql = "SELECT `id`, `name` FROM `people` “  . "WHERE `name` = 'Susan' “  . "ORDER BY `name` ASC "; |

#### - Function and Method Declaration

Functions must be named according to naming conventions described in [#Functions and Methods].

Methods inside classes must always declare their visibility by using one of the private, protected, or public modifiers.

The following is an example of an acceptable function declaration in a class:

|  |
| --- |
| /\*\*  \* Documentation Block Here  \*/  class Foo {  /\*\*  \* Documentation Block Here  \*/  public function bar() {  // all contents of function  // must be indented four spaces  }  } |

III>Report No.3: Software Requirements Specifications

# Project Success Criteria

|  |
| --- |
| Project Success Criteria |
| * The project that finishes in scheduled time, within the allocated budgets and which satisfies the customer requirements. * All high-priority functionality defined in the requirements specification is delivered in the first release. * Open-sources are optimized closely, less complexity, improving quality of the system. * The estimated number of residual defects does not exceed 3 per function point. |

# 3.1. User Requirement Specification

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Code | Name | Description |
| *Common User Requirement* | EL\_CUR\_UR1 | Interface | * The system shall provide Vietnamese interface. * The system need have the friendly graphical user interface, visual, lifelike, logical layout. |
|  | EL\_CUR\_UR2 | Announcement | The system will release an announcement if the result was not found. |
|  | EL\_CUR\_UR3 | Register/ login | Users can register and login easily and quickly. |
|  | EL\_CUR\_UR4 | Categorize | Data will be categorized efficiently to help user to search conveniently. |
|  | EL\_CUR\_UR5 | Question bank | The system will provide many kinds of exercises, questions and answers for students. |
|  | EL\_CUR\_UR6 | News providing | The system will provide news about education and university entrance exam… |
|  | EL\_CUR\_UR7 | Search | The system allows users to search for goods quickly, efficiently. |
|  | EL\_CUR\_UR8 | Testing online | Admin will create tests and user can choose a test to do. |
| *Business User Requirement* | EL\_BUR\_UR1 | Performance | The system should optimize to operate stably, improve quality to create competitive advantage. |
|  | EL\_BUR\_UR2 | Decentralization | The system need divide authority for each group: admin, mod, user, teacher… |
|  | EL\_BUR\_UR3 | Security | Commit that the system operate stably. It is very bad when users visit the website or manipulate on it while server die or other crash. |
|  | EL\_BUR\_UR4 | Statistic | The system will provide statistic about number of users visit the website. |
|  | EL\_BUR\_UR5 | Administrator | The system provide administrator site: admin can configuration for system and update database on the website. |
| *Single User Requirement* | EL\_SUR\_UR1 | Studying online | The system will provide theory site: showing theories and knowledge of subjects and refer exercises, and objectives. User can study and practice together. |
|  | EL\_SUR\_UR2 | Report | After testing, the system will send result for user. There are 2 kinds of report: basic and detail. The basic report will show point and time to use to user. The detail report will provide more information: detail about answers which user chooses. |
|  | EL\_SUR\_UR3 | Forum | The system will have forum site: this site is a place which members can share experience, discuss, and exchange some problems need care. |
|  | EL\_SUR\_UR4 | Case study | The system will provide case study for each question. Users can self-study better. |

# 3.2. System Requirement Specification

## 3.2.1. External Interface Requirements

**3.2.1.1. User Interfaces**

A user behavior pane will be available in all workflow scenarios to allow easy access to online help and next-step options. The navigation options in every screen will be similar to lower or eliminate the system learning curve. Input confirmation and error notification will be consistent throughout the application.

**3.2.1.2. Hardware Interfaces**

- Common peripheral devices:

* ADSL (or wireless) modem and cable internet
* Mouse
* Keyboard

- Required (minimum) specifications:

* CPU: Intel® Pentium® 4 2.4 GHz or better supported
* RAM: 512MB (768MB for Windows 7)
* Hard Drive: 2GB of free space

- Recommended specifications:

* CPU: 2.4 GHz dual core or better supported
* RAM: 1GB for XP, 2GB for Windows 7
* Hard Drive: 2GB of free space

**3.2.1.3. Software Interfaces**

“E-Learning” system can run on any platform/operating system (includes Windows, Mac OS, and Linux). That support browsers listed below:

* Firefox (3.0 or higher)
* Google chrome (9.0 or higher)
* Internet Explorer (7.0 or higher)

**3.2.1.4. Communications Protocol**

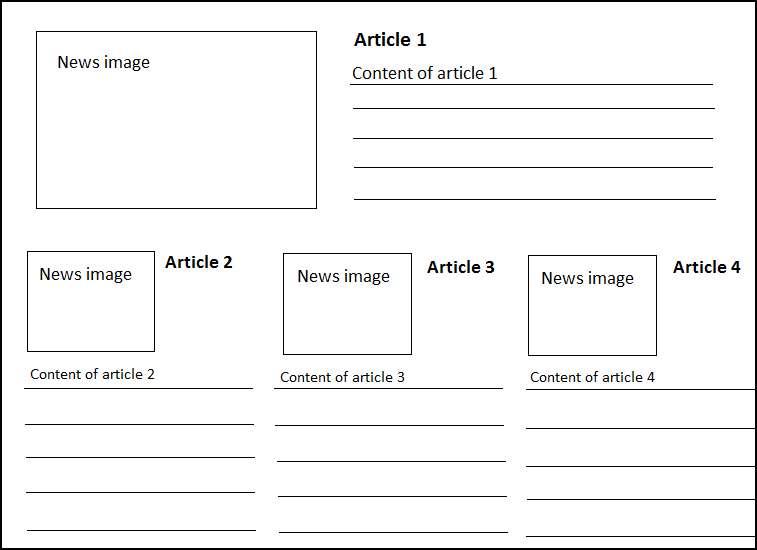
The system will require communication via the Hyper Text Transfer Protocol (HTTP) to complete interactions based services with client computers. “E-Learning” system also requires a web browser to function. The Web Browser must comply with standards for HTTP version 1.0 or 1.1 HTTP version 1.0 is a well-founded and highly supported protocol. Now considering legacy by some organizations, we believe this is a safe foundation for the system.

The system also indirectly requires some network connection to the internet, over which it may communicate in HTTP. This network connection assumes a physical or wireless connection from the client computer to a consumer Internet Service Provider (ISP) or enterprise environment Local Area Network (LAN).

## 3.2.2. System Features

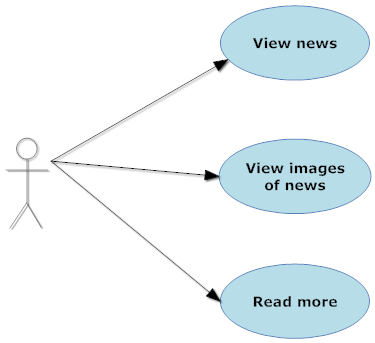
**3.2.2.1. Home**

The page is the “E-Learning” system home page, and will be showing whenever user load “E-Learning” website. This page looks like news site.

****

**Figure 3.1:** Home page

**3.2.2.1.1 Use Case Diagram**



**Figure 3.2:** Home page use case diagram

**3.2.2.1.2 Use Case Specification**

**Use case 1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE- SPECIFICATION** | | | | |
| **Use-case No.** | UC001 | **Use-case Version** | | 1.0 |
| **Use-case Name** | View news | | | |
| **Author** | Huynt | | | |
| **Date** | 26/05/2012 | **Priority** | Normal | |
| **Actor:**  Visitor, Register Member, Admin  **Summary:**  After main page load, user can view news in the front page.  **Goal:**  Display news that user chose.  **Triggers**   * When user click on one of news link in the front page.   **Preconditions:**   * Main page loaded successfully * User select news in front page   **Post Conditions:**  Display news page  **Main Success Scenario:**  1. Main page is loaded with links news  2. User clicks on one of news link in the front page.  **Alternative Scenario:**  None  **Exceptions:**   1. Page is not loaded successfully. 2. News that selected is displayed wrongly.   **Relationships:**  This use case related to News page.  **Business Rules:**  - News must available for anyone visit website  - News must display in the front page of website  - News display content after user click on a link of its | | | | |

**Use case 2**

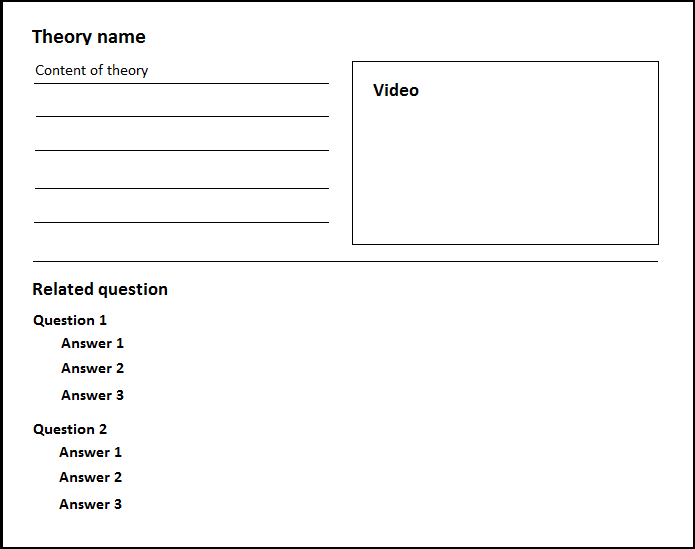
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE- SPECIFICATION** | | | | |
| **Use-case No.** | UC002 | **Use-case Version** | | 1.0 |
| **Use-case Name** | View images | | | |
| **Author** | Huynt | | | |
| **Date** | 26/05/2012 | **Priority** | Low | |
| **Actor:**  Visitor, Register Member, Admin  **Summary:**  After main page load, user can view images in the front page.  **Goal:**  Display images in the front page.  **Triggers**  None  **Preconditions:**   * Main page loaded successfully   **Post Conditions:**  Display images  **Main Success Scenario:**  1. Main page is loaded with collection of images.  **Alternative Scenario:**  None  **Exceptions:**  1. Page is not loaded successfully.  **Relationships:**  This use case related to Home page.  **Business Rules:**  - Images must available for anyone visit website  - Images must display in the front page of website | | | | |

**Use case 3**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE- SPECIFICATION** | | | | |
| **Use-case No.** | UC003 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Read more | | | |
| **Author** | Huynt | | | |
| **Date** | 26/05/2012 | **Priority** | Low | |
| **Actor:**  Visitor, Register Member, Admin  **Summary:**  After main page load, user can read more news in the front page.  **Goal:**  Display full content of news that user chose to read more in front page  **Triggers**  When user click on “read more” link  **Preconditions:**   * Main page load successfully   **Post Conditions:**  Display content of a topic in news.  **Main Success Scenario:**  1. Main page is loaded with an example of news.  **Alternative Scenario:**  None  **Exceptions:**  None  **Relationships:**  This use case related to Home page.  **Business Rules:**  - Content of News must available for anyone visit website  - Content of News must display in the front page of website | | | | |

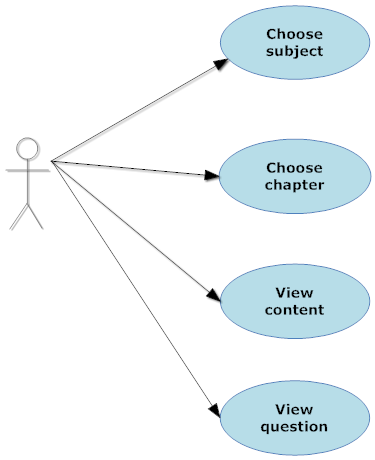
**3.2.2.2. Theory libraries**

The features will allow user to choose theories about subjects to study. User need choose subject -> chapter -> theory. If a theory has learning’s video, the system will display the video below content of theory.



**Figure 3.3:** View content of theory page

**3.2.2.2.1 Use Case Diagram**

****

**Figure 3.4:** Theory libraries use case diagram

**3.2.2.2.2 Use Case Specification**

**Use case 4**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE- SPECIFICATION** | | | | |
| **Use-case No.** | UC004 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Chose subject | | | |
| **Author** | Huynt | | | |
| **Date** | 27/05/2012 | **Priority** | High | |
| **Actor:**  Visitor, Register Member, Admin  **Summary:**  After main page load, user can use theory library component to choose subject to view.  **Goal:**  Display all chapter that subject (the one that user chose) contain  **Triggers**   * When user click on tab theory library on the main menu or menu bar. * User click one of the list subject that display in front page or left menu   **Preconditions:**   * Theory Library page loaded successfully. * User select subject for display.   **Post Conditions:**  Show chapter of a subject in theory library page  **Main Success Scenario:**  1. User chose Theory Library page and the page loaded successfully  2. User clicks on a subjects in front page.  **Alternative Scenario:**  1. User chose Theory Library page and the page loaded successfully  2. User clicks on a subject in left menu.  **Exceptions:**   1. Page is not loaded successfully. 2. Chapter of subject display wrongly.   **Relationships:**  This use case related to Theory Library page and first step of view Theory process  **Business Rules:**  - Theory Library component must always available for anyone visits the website.  - System will display results which show the list chapters of subject they chose. | | | | |

**Use case 5**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE- SPECIFICATION** | | | | |
| **Use-case No.** | UC005 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Chose chapter | | | |
| **Author** | Huynt | | | |
| **Date** | 27/05/2012 | **Priority** | High | |
| **Actor:**  Visitor, Register Member, Admin  **Summary:**  After main page load, user can use theory library component to choose chapter to view.  **Goal:**  Display all theory that chapter (the one that user chose) contain  **Triggers**   * User click one of the list chapter that display in front page   **Preconditions:**   * Theory Library page loaded successfully. * User select subject * User select chapter for display.   **Post Conditions:**  Show theory of a chapter in theory library page  **Main Success Scenario:**  1. User clicks on a chapter in front page.  **Alternative Scenario:**  1. User clicks on a chapter in front page. (chapters that are example of each subject)  **Exceptions:**   1. Page is not loaded successfully. 2. Theory of chapter display wrongly.   **Relationships:**  This use case related to Theory Library page and second step of view Theory process  **Business Rules:**  - Theory Library component must always available for anyone visits the website.  - System will display results which show list theories of chapter they chose. | | | | |

**Use case 6**

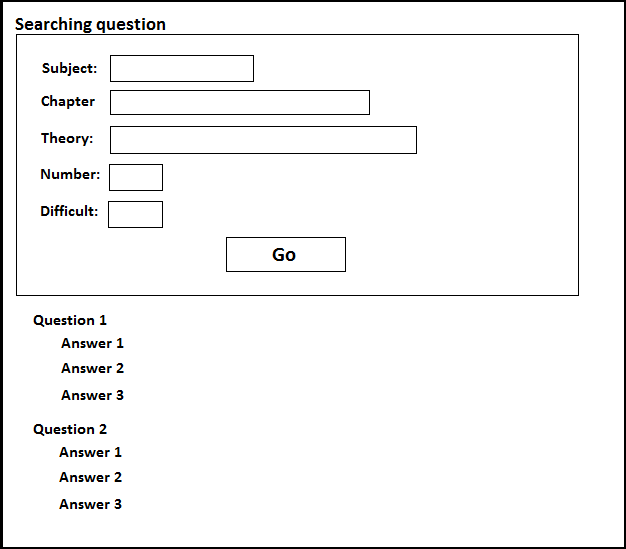
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE- SPECIFICATION** | | | | |
| **Use-case No.** | UC006 | **Use-case Version** | | 1.0 |
| **Use-case Name** | View content | | | |
| **Author** | Huynt | | | |
| **Date** | 27/05/2012 | **Priority** | High | |
| **Actor:**  Visitor, Register Member, Admin  **Summary:**  After main page load, user can use theory library component to choose theory to view content.  **Goal:**  Display content and play video of theory that user chose to display  **Triggers**   * User click one of the list chapter that display in front page * User click play icon in video area   **Preconditions:**   * Theory Library page loaded successfully. * User select subject * User select chapter for display. * User select theory for display.   **Post Conditions:**  Show chapter of a subject in theory library page  **Main Success Scenario:**  1. User clicks on a theory in front page.  **Alternative Scenario:**  None  **Exceptions:**   1. Page is not loaded successfully. 2. Content or video of theory display wrongly.   **Relationships:**  This use case related to Theory Library page.  **Business Rules:**  - Theory Library component must always available for anyone visits the website.  - System will display results which show the content and video (if available) of theory they chose. | | | | |

**Use case 7**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE- SPECIFICATION** | | | | |
| **Use-case No.** | UC006 | **Use-case Version** | | 1.0 |
| **Use-case Name** | View question | | | |
| **Author** | Huynt | | | |
| **Date** | 27/05/2012 | **Priority** | High | |
| **Actor:**  Visitor, Register Member, Admin  **Summary:**  After main page load, user can use theory library component to choose chapter to view.  **Goal:**  Display, allow user answer and display the answer of question in theory that user chose to display.  **Triggers**   * User click one of the list chapter that display in front page * User click to link “Answer” bellow each question   **Preconditions:**   * Theory Library page loaded successfully. * User select subject * User select chapter for display. * User select theory for display.   **Post Conditions:**  Show chapter of a subject in theory library page  **Main Success Scenario:**  1. User clicks on a theory in front page.  **Alternative Scenario:**  None  **Exceptions:**   1. Page is not loaded successfully. 2. Question of theory display wrongly. 3. Answer of question display wrongly.   **Relationships:**  This use case related to Theory Library page.  **Business Rules:**  - Theory Library component must always available for anyone visits the website.  - System will display results which show the question of theory they chose and the answer. | | | | |

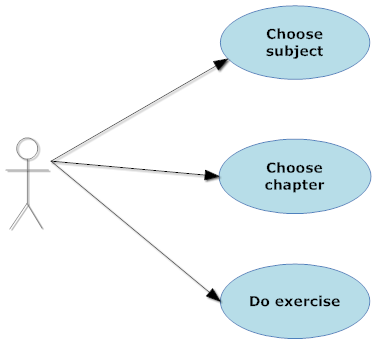
**3.2.2.3. Exercise libraries**

The features allow user to choose exercise to practice following subject, theory and difficult. User will choose subject -> chapter -> theory-> number of question -> difficult.



**Figure 3.5:** Exercise libraries page

**3.2.2.3.1 Use Case Diagram**

****

**Figure 3.6:** Exercise libraries use case diagram

**3.2.2.3.2 Use Case Specification**

**Use case 8**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE- SPECIFICATION** | | | | |
| **Use-case No.** | UC008 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Chose subject | | | |
| **Author** | Huynt | | | |
| **Date** | 27/05/2012 | **Priority** | High | |
| **Actor:**  Visitor, Register Member, Admin  **Summary:**  After main page load, user can use exercise library component to choose subject to do questions.  **Goal:**  Display all chapter that subject contain (the one that user chose) in combo box  **Triggers**   * When user click on tab exercise library on the menu bar. * User click one of the list subject that display in combo box in the front page   **Preconditions:**   * Exercise Library page loaded successfully. * User select subject for display.   **Post Conditions:**  Show list chapter of a subject in combo box of theory library page  **Main Success Scenario:**  1. User chose Exercise Library page and the page loaded successfully  2. User clicks on a subject in combo box of front page.  **Alternative Scenario:**  None  **Exceptions:**   1. Page is not loaded successfully. 2. List chapter of subject display wrongly.   **Relationships:**  This use case related to Exercise Library page  **Business Rules:**  - Exercise Library component must always available for anyone visits the website.  - System display results which show the list chapters in combo box of subject they chose. | | | | |

**Use case 9**

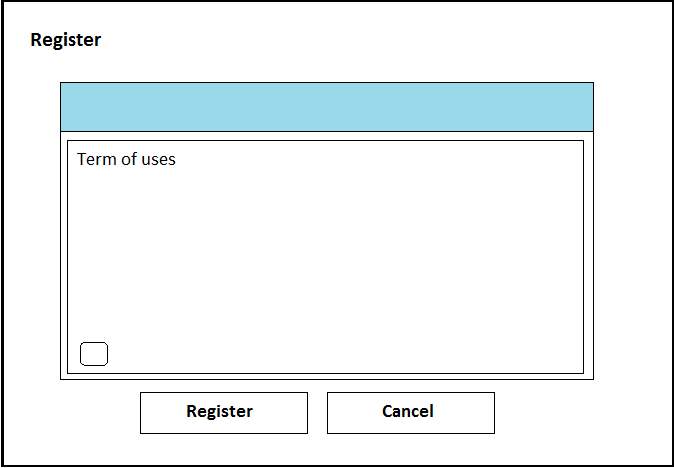
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE- SPECIFICATION** | | | | |
| **Use-case No.** | UC009 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Chose chapter | | | |
| **Author** | Huynt | | | |
| **Date** | 27/05/2012 | **Priority** | High | |
| **Actor:**  Visitor, Register Member, Admin  **Summary:**  After main page load, user can use exercise library component to choose chapter to do questions.  **Goal:**  Display all theories that chapter contain (the one that user chose) in combo box  **Triggers**   * When user click on tab exercise library on the menu bar. * User click one of the list subject and then chapter that display in combo box in the front page   **Preconditions:**   * Exercise Library page loaded successfully. * User select subject, chapter for display.   **Post Conditions:**  Show list theory of a subject in combo box of theory library page  **Main Success Scenario:**  1. User chose Exercise Library page and the page loaded successfully  2. User clicks on a subject in combo box of front page.  3. User clicks on a chapter in combo box of front page.  **Alternative Scenario:**  None  **Exceptions:**   1. Page is not loaded successfully. 2. List chapter of subject display wrongly.   **Relationships:**  This use case related to Exercise Library page.  **Business Rules:**  - Exercise Library component must always available for anyone visits the website.  - System display results which show the list theory in combo box of subject they chose. | | | | |

**Use case 10**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE- SPECIFICATION** | | | | |
| **Use-case No.** | UC010 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Do exercise | | | |
| **Author** | Huynt | | | |
| **Date** | 27/05/2012 | **Priority** | High | |
| **Actor:**  Visitor, Register Member, Admin  **Summary:**  After main page load, user can use exercise library component to do questions.  **Goal:**  Display all question in condition that user provide and answer (after user click on answer link)  **Triggers**   * When user click on tab exercise library on the menu bar. * User click one of the list subject, chapter, theory, number of question and difficulty (optional) that display in the front page   **Preconditions:**   * Exercise Library page loaded successfully. * User select subject, chapter, theory, number of question and difficulty (optional) to do questions.   **Post Conditions:**  Display list of question in the previous condition that user have provide.  **Main Success Scenario:**  1. User chose Exercise Library page and the page loaded successfully  2. User clicks on a subject in combo box of front page.  3. User clicks on a chapter in combo box of front page.  4. User clicks on a theory in combo box of front page.  5. User clicks on a number of questions in combo box of front page.  6. User clicks on a choice of difficulty in combo box of front page.(optional)  **Alternative Scenario:**  None  **Exceptions:**   1. Page is not loaded successfully. 2. List question display wrongly. 3. Questions answer display wrongly.   **Relationships:**  This use case related to Exercise Library page.  **Business Rules:**  - Exercise Library component must always available for anyone visits the website.  - System display results which show the list theory in combo box of subject they chose. | | | | |

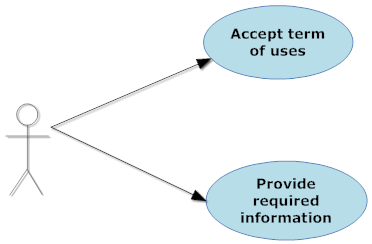
**3.2.2.4. Register**

The page allow visitor to register as a member of “E-Learning” in order to use more of its features and receive member’s services.



**Figure 3.7:** Register page

**3.2.2.4.1 Use Case Diagram**

****

**Figure 3.8:** Register use case diagram

**3.2.2.4.2 Use Case Specification**

**Use case 11**

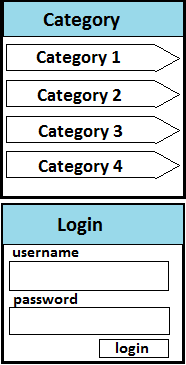
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE- SPECIFICATION** | | | | |
| **Use-case No.** | UC011 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Accept term of use | | | |
| **Author** | Huynt | | | |
| **Date** | 28/05/2012 | **Priority** | High | |
| **Actor:**  Visitor  **Summary:**  Before registered as a member of “E-learning-website”, visitor must accept the website term of use, as well as they should read what they will receive once registered.  **Goal:**  Help visitor understand the term of use to avoid any unfortunate violation of website policies and to encourage to them willfully become a member of “E-learning website”  **Triggers**   * When user click on link “Create an account” in left menu of Home Page.   **Preconditions:**   * User doesn’t accept the term of use yet. * User chose to register.   **Post Conditions:**  Allow user to create an account.  **Main Success Scenario:**  1. User clicks on link “Create an account” in left menu of Home Page.  2. The term of use will show up.  3. User clicks on button “I agree with the term”.  **Alternative Scenario:**   1. When user clicks on button “I don’t agree with the term” then go to forum page.   **Exceptions:**   1. Page is not loaded successfully. 2. List question display wrongly. 3. Questions answer display wrongly.   **Relationships:**  This case is the first step in register process.  **Business Rules:**  - Register function must always available for anyone visits the website.  - System display results which show the list theory in combo box of subject they chose. | | | | |

**Use case 12**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE- SPECIFICATION** | | | | |
| **Use-case No.** | UC012 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Provide required information | | | |
| **Author** | Huynt | | | |
| **Date** | 28/05/2012 | **Priority** | High | |
| **Actor:**  Visitor  **Summary:**  When register, user must provide their information, so that system can recognized them when they are login and also provide a way to communicate between user and website.  **Goal:**  For system to recognize user when they are login.  **Triggers**   * After user accept the Term of use.   **Preconditions:**   * User chose to register. * User accepts the term of use.   **Post Conditions:**  Allow user to complete register after they fill in all mandatory field.  **Main Success Scenario:**  1. User accepts the Term of use.  2. Prompt user to fill in mandatory fields.  + User name  + Email  + Verify Email  + Password  + Verify password  . + Language optional).  + Time zone (optional).  + Confirmation code  3. After user fills all mandatory fields, allow user to complete register after click on accept “Accept” button.  **Alternative Scenario:**  None  **Exceptions:**   1. Page is not loaded successfully. 2. Page is not loaded successfully.   **Relationships:**  This case is the second step in register process.  **Business Rules:**  - In order to complete the registration, user must fill in all mandatory fields except time zone and language.  - The password should be at least 6 characters long and at most 100 characters. | | | | |

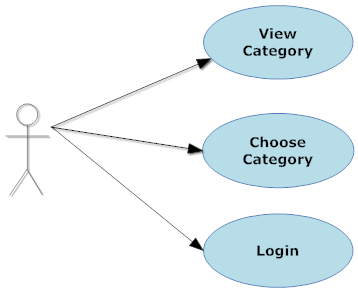
**3.2.2.5. Left menu**

This menu will be located on the left hand of website. It’s displayed on most of sites.



**Figure 3.9:** Left menu

**3.2.2.5.1 Use Case Diagram**

****

**Figure 3.10:** Left menu use case diagram

**3.2.2.5.2 Use Case Specification**

**Use case 13**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE- SPECIFICATION** | | | | |
| **Use-case No.** | UC013 | **Use-case Version** | | 1.0 |
| **Use-case Name** | View category | | | |
| **Author** | Huynt | | | |
| **Date** | 29/05/2012 | **Priority** | Normal | |
| **Actor:**  Visitor, Register Member, Admin  **Summary:**  Each of categories is a link to a site of “E-Learning”. The categories can automatically cross out more of child categories. User can click on it to go to the site.  **Goal:**  Redirect to sites  **Triggers**   * When page is loaded. * User click on category hyperlink in left menu.   **Preconditions:**   * Page is loaded successfully.   **Post Conditions:**  The system will redirect the site to Home Page.  **Main Success Scenario:**   1. User click on category hyperlink in left menu.   **Alternative Scenario:**  None  **Exceptions:**   1. Page is not loaded successfully. 2. Cannot redirect to a site.   **Relationships:**  This user case is related to Home Page.  **Business Rules:**  - This link is always displayed. | | | | |

**Use case 14**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE- SPECIFICATION** | | | | |
| **Use-case No.** | UC014 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Choose category | | | |
| **Author** | Huynt | | | |
| **Date** | 29/05/2012 | **Priority** | Normal | |
| **Actor:**  Visitor, Register Member, Admin  **Summary:**  After user click on the category, the system will display the site according to the category.  **Goal:**  Display a site following category.  **Triggers**   * When page is loaded and user click on category hyperlink in left menu   **Preconditions:**   * Page is loaded successfully.   **Post Conditions:**  The system will redirect the site to category page.  List of subjects display in left menu  **Main Success Scenario:**   1. User click on category hyperlink in left menu.   **Alternative Scenario:**  None  **Exceptions:**  1. Page is not loaded successfully.  2. Cannot redirect to category page  **Relationships:**  This user case is related to category page.  **Business Rules:**  - This link is always displayed. | | | | |

**Use case 15**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE- SPECIFICATION** | | | | |
| **Use-case No.** | UC015 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Login | | | |
| **Author** | Huynt | | | |
| **Date** | 29/05/2012 | **Priority** | Normal | |
| **Actor:**  Register Member, Admin  **Summary:**  User has to input user name, and password to login to the system. If username or password is not valid, user cannot login to the system, and has to try again.  **Goal:**  Allow user login to the system.  **Triggers**   * When user click on “Login” hyperlink.   **Preconditions:**   * Page is loaded successfully. * User has valid user name and valid password to the system.   **Post Conditions:**   * In case of fail:   + User cannot login to the system.  + The system must give an error message.  + User has to try again to login.   * In case of success:   + The system gives a successful message.  + User login to the system successfully.  **Main Success Scenario:**  The use case start when:   1. Page is load successfully. 2. User click on “Login” hyperlink. 3. The system display a popup allow user to input user name and password. 4. User input username and password. 5. User press on “Login” button.   **Alternative Scenario:**  None  **Exceptions:**   1. Page is not loaded successfully. 2. The system does not load login panel.  * Connection too slow. * The system will allow user to reload page via “F5” button.  1. User input invalid user name  * The system will tell user about this error via message. * Return to step 4.  1. User input invalid password  * The system will tell user about this error via message. * Return to step 4   **Relationships:**  This user case is related to Login page.  **Business Rules:**  - After click on “Login” button, the system must display a message that allow user to input user name and password. In case of login fail, the system must display error message to inform user know the reason, invalid username, invalid password…  - User has to create account before login to the system. | | | | |

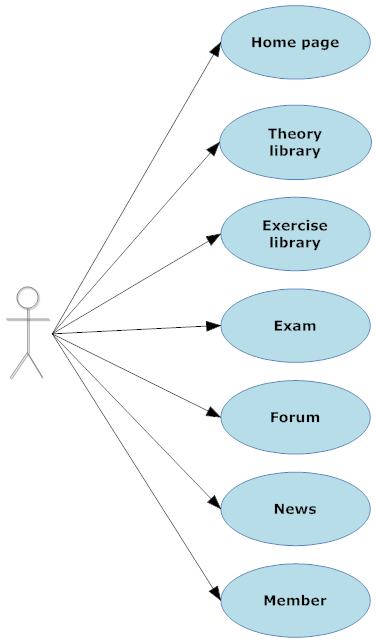
**3.2.2.6. Menu bar**

This menu includes links to main sites of the system. It is located under logo and banner.



**Figure 3.11:** Menu bar

**3.2.2.6.1 Use Case Diagram**

****

**Figure 3.12:** Menu bar use case diagram

**3.2.2.6.2 Use Case Specification**

**Use case 16**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE- SPECIFICATION** | | | | |
| **Use-case No.** | UC016 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Home page | | | |
| **Author** | Huynt | | | |
| **Date** | 29/05/2012 | **Priority** | Normal | |
| **Actor:**  Visitor, Register Member, Admin  **Summary:**  On each page, “Home” is a tab in menu bar. When user click on this tab, system will redirect the site to Home Page  **Goal:**  Redirect to Home Page  **Triggers**   * When page is loaded and user click on ‘Home” tab in menu bar   **Preconditions:**   * Page is loaded successfully.   **Post Conditions:**  The system will redirect the site to Home Page.  **Main Success Scenario:**   1. User click on ‘Home” tab in menu bar.   **Alternative Scenario:**  None  **Exceptions:**   1. Page is not loaded successfully. 2. Cannot redirect to Home Page   **Relationships:**  This user case is related to Home Page.  **Business Rules:**  - This tab is always displayed. | | | | |

**Use case 17**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE- SPECIFICATION** | | | | |
| **Use-case No.** | UC017 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Theory Library | | | |
| **Author** | Huynt | | | |
| **Date** | 29/05/2012 | **Priority** | Normal | |
| **Actor:**  Visitor, Register Member, Admin  **Summary:**  On each page, “Theory” is a tab. When user click on this tab, system will redirect the site to Theory Library Page  **Goal:**  Redirect to Theory Library Page  **Triggers**   * When page is loaded and user click on ‘Theory” tab in menu bar   **Preconditions:**   * Page is loaded successfully.   **Post Conditions:**  The system will redirect the site to Theory Library Page.  **Main Success Scenario:**   1. User click on ‘Theory” tab in menu bar.   **Alternative Scenario:**  None  **Exceptions:**  1. Page is not loaded successfully.  2. Cannot redirect to Theory Library Page  **Relationships:**  This user case is related to Theory Library Page.  **Business Rules:**  - This tab is always displayed. | | | | |

**Use case 18**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE- SPECIFICATION** | | | | |
| **Use-case No.** | UC018 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Exercise Library | | | |
| **Author** | Huynt | | | |
| **Date** | 29/05/2012 | **Priority** | Normal | |
| **Actor:**  Visitor, Register Member, Admin  **Summary:**  On each page, “Exercise” is a tab. When user click on this tab, system will redirect the site to Exercise Library Page  **Goal:**  Redirect to Exercise Library Page  **Triggers**   * When page is loaded and user click on ‘Exercise” tab in menu bar   **Preconditions:**   * Page is loaded successfully.   **Post Conditions:**  The system will redirect the site to Exercise Library Page.  **Main Success Scenario:**   1. User click on ‘Exercise” tab in menu bar.   **Alternative Scenario:**  None  **Exceptions:**  1. Page is not loaded successfully.  2. Cannot redirect to Exercise Library Page  **Relationships:**  This user case is related to Exercise Library Page.  **Business Rules:**  - This tab is always displayed. | | | | |

**Use case 19**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE- SPECIFICATION** | | | | |
| **Use-case No.** | UC019 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Exam | | | |
| **Author** | Huynt | | | |
| **Date** | 29/05/2012 | **Priority** | Normal | |
| **Actor:**  Visitor, Register Member, Admin  **Summary:**  On each page, “Exam” is a tab. When user click on this tab, system will redirect the site to Exam Page  **Goal:**  Redirect to Exam Page  **Triggers**   * When page is loaded and user click on ‘Exam” tab in menu bar   **Preconditions:**   * Page is loaded successfully.   **Post Conditions:**  The system will redirect the site to Exam Page.  **Main Success Scenario:**   1. User click on ‘Exam” tab in menu bar.   **Alternative Scenario:**  None  **Exceptions:**  1. Page is not loaded successfully.  2. Cannot redirect to Exam Page  **Relationships:**  This user case is related to Exam Page.  **Business Rules:**  - This tab is always displayed. | | | | |

**Use case 20**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE- SPECIFICATION** | | | | |
| **Use-case No.** | UC020 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Forum | | | |
| **Author** | Huynt | | | |
| **Date** | 29/05/2012 | **Priority** | Normal | |
| **Actor:**  Visitor, Register Member, Admin  **Summary:**  On each page, “Forum” is a tab. When user click on this tab, system will redirect the site to Theory Library Page  **Goal:**  Redirect to Forum Page  **Triggers**   * When page is loaded and user click on ‘Forum” tab in menu bar   **Preconditions:**   * Page is loaded successfully.   **Post Conditions:**  The system will redirect the site to Forum Page.  **Main Success Scenario:**   1. User click on ‘Forum” tab in menu bar.   **Alternative Scenario:**  None  **Exceptions:**  1. Page is not loaded successfully.  2. Cannot redirect to Forum Page  **Relationships:**  This user case is related to Forum Page.  **Business Rules:**  - This tab is always displayed. | | | | |

**Use case 21**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE- SPECIFICATION** | | | | |
| **Use-case No.** | UC021 | **Use-case Version** | | 1.0 |
| **Use-case Name** | News | | | |
| **Author** | Huynt | | | |
| **Date** | 29/05/2012 | **Priority** | Normal | |
| **Actor:**  Visitor, Register Member, Admin  **Summary:**  On each page, “News” is a tab. When user click on this tab, system will redirect the site to News Page  **Goal:**  Redirect to News Page  **Triggers**   * When page is loaded and user click on ‘News” tab in menu bar   **Preconditions:**   * Page is loaded successfully.   **Post Conditions:**  The system will redirect the site to News Page.  **Main Success Scenario:**   1. User click on ‘News” tab in menu bar.   **Alternative Scenario:**  None  **Exceptions:**  1. Page is not loaded successfully.  2. Cannot redirect to News Page  **Relationships:**  This user case is related to News Page.  **Business Rules:**  - This tab is always displayed. | | | | |

**Use case 22**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE- SPECIFICATION** | | | | |
| **Use-case No.** | UC022 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Members | | | |
| **Author** | Huynt | | | |
| **Date** | 29/05/2012 | **Priority** | Normal | |
| **Actor:**  Visitor, Register Member, Admin  **Summary:**  On each page, “member” is a tab. When user click on this tab, system will redirect the site to Members Page  **Goal:**  Redirect to Members Page  **Triggers**   * When page is loaded and user click on “member” tab in menu bar   **Preconditions:**   * Page is loaded successfully.   **Post Conditions:**  The system will redirect the site to Members Page.  **Main Success Scenario:**   1. User click on ‘member” tab in menu bar.   **Alternative Scenario:**  None  **Exceptions:**  1. Page is not loaded successfully.  2. Cannot redirect to Members Page  **Relationships:**  This user case is related to Members Page.  **Business Rules:**  - This tab is always displayed. | | | | |

## 3.2.3. Software System Attributes

**3.2.3.1 Reliability**

**3.2.3.2 Availability**

**3.2.3.3 Security**

**3.2.3.4 Maintainability**

**3.2.3.5 Portability**

**3.2.3.6 Performance**

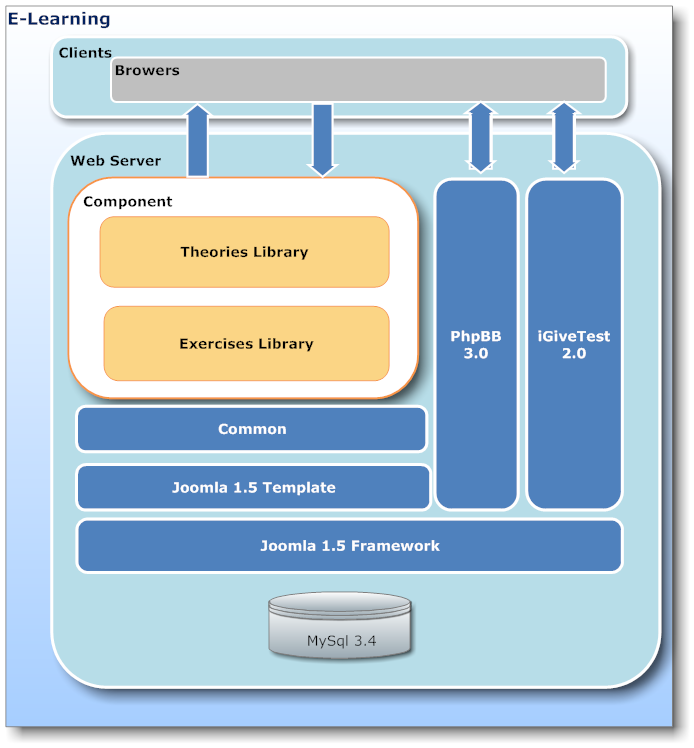
# 3.3. Entity Relationship Diagram or Data Structures

IV> Report No.4: Software Design Description

# 4.1. Design Overview

# 4.2. System Architectural Design

## 4.2.1. Choice of System Architecture

To approach the project, we choose the system architecture which is built as the diagram as below. This diagram will help us visually abstract the system and understand the key modules with their interaction in our E-Learning system.

**Figure 4.1:** E-Learning system architecture design

**Client:** the client layer is user, user will access to web server by browser. The layer will send requests on the server. Then, the server will receive and return request for user.

**Web server:** the layer is where process and returns request to the client. After receiving request, the controller on web server will control the model to get data from MySQL server. Then, the controller sends methods of model to update the view and return the browser on client. The layer includes 7 modules:

**+ Component:**

The layer includes 2 components: theories library and exercises library. *Theories library* will help users to study on the website. Content of theories are video, text which is compiled easy to understand and impression; users will study better and not boring. We based on **Bloom model** about learning method, each of theory have objective and concern questions. The questions usually are easy and have case study to help users to understand problems clearly. The next module is *exercises library*. The module brings out a large number of exercises to users to choose. Users can filter by subject, chapter, theory, and difficult.

**+ Common:**

The layer includes common definition, configuration and model objects. Components are developed will have structure look like the tree diagram:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | e-learning-website | | | |  |
|  | |-- | style | | |  |
|  | | | `-- style.css | | |  |
|  | |-- | | images | | |
|  | |-- | | components | | |
|  | | | | |-- com\_componentname | | |
|  | | | | | | models | |
|  | | | | | | `-- componentname.php | |
|  | | | | | | `-- index.html | |
|  | | | | |-- | views | |
|  | | | | | | `-- index.html | |
|  | | | | | | `-- componentname | |
|  | | | | | | |-- tmpl | |
|  | | | | | | | `-- default.php | |
|  | | | | | | | `-- index.html | |
|  | | | | | | |-- index.html | |
|  | | | | | | |-- view.html.php | |
|  | | | | |-- | controller.php | |
|  | | | | |-- | index.php | |
|  | | | | |-- | componentname.php | |
|  | |-- | | administrator | | |
|  | | | | |-- | components | |
|  | | | | | | |-- com\_componentname | |
|  | | | | | | `-- componentname.php | |
|  | | | | | | `-- componentname.xml | |
|  | | | | | | `-- install.sql | |
|  | | | | | | `-- uninstall.sql | |
|  | | | | | | `-- index.html | |

We will create “*index.html*” files in folders contain our code files. The file is forced in all of folders “**E-learning-website**” to increase security, avoid accessing administrator files. The file doesn’t have content, it only display white page.

Code *.css* to design components which we develop is placed in “**e-learning-website/style**” and image files will be put into Images folder.

Folder “**e-learning-website/components**”contain all of components of *Joomla* and our component. The components are categorized using prefix “com\_” and name of component. We choose Model-View-Controller model to develop the system. The schema of MVC code will be described below.

+ Model:

The classes that connect to database will be in “*componentname.php*” file. The file is placed in “**components/com\_componentname**”.The default “*componentname.php”* is as follow:

|  |  |  |  |
| --- | --- | --- | --- |
|  | *// application/models/applicationmodel.php*  jimport( ‘joomla.application.component.model’ );  class ComponentNameModelComponentName extends JModel | | |
|  | { |  | |
|  |  | function getModel() |  |
|  |  | { |  |
|  |  | $db =& JFactory::getDBO();  $query = “SELECT column\_name FROM table\_name”  $db ->setQuery( $query );  $result = $db ->loadResult();  return $result; | |
|  |  | } |  |
|  | } |  |  |

The class is extended *JModel* is an abstract class of *Joomla* framework. The class provides the basic functionality for concrete model objects in conjunction with *Joomla’s* MVC pattern.

+ View:

Files that concern display will be in Views folder. HTML code of component on front page is in *default.php* file. The file is placed in “**view/componentname/tmpl/**”. Besides, in “**Views**” folder, we use file “*view.html.php”* involves PHP code about display on the website.

The default “*view.html.php”* is as follow:

|  |  |  |  |
| --- | --- | --- | --- |
|  | *// application/views/* *applicationview/view.html.php*  jimport( ‘joomla.application.component.view’ );  class ComponentNameViewComponentName extends JView | | |
|  | { |  | |
|  |  | function display($tpl = null) | |
|  |  | { | |
|  |  | $model = & $this->getModel(); | |
|  |  | *// action body* |  |
|  |  | } |  |
|  | } |  |  |

The class is extended *JView* is an abstract class of *Joomla* framework*.* Creating tasks of view is very simple: It retrieves the data to be displayed and pushes it into the template. Data is pushed into the template using the *JView::assignRef* method. (Note: The key (the first argument) passed to the assignRef method cannot be preceded by an underscore i.e. *$this->assignRef('\_greeting',$greeting).* Doing so will cause the *assignRef* method to return false and your variable will not be pushed into the template.)

Our template is very simple: we only want to display the greeting that was passed in from the view.

This file is “**componentname/views/componentname/tmpl/default.php**”:

|  |
| --- |
| <?php defined('\_JEXEC') or die('Restricted access'); ?>  <h1><?php //echo $this->chapterName;?></h1> |

+ Controller:

Our component's action controllers contain our action workflow, and do the work of mapping our requests to the appropriate models and views. No data manipulation is required. All that needs to be done is the appropriate view loaded. We will have only one method in our controller: *display().* Most of the required functionality is built into the *JController* class, so all that we need to do is invoke the *JController::display()* method.

The code for the base controller “*componentname/controller.php*” is:

|  |
| --- |
| <?php  // No direct access  defined( '\_JEXEC' ) or die( 'Restricted access' );  jimport('joomla.application.component.controller');  class ComponentNameController extends JController  {  function display()  {  parent::display();  }  } |

The *JController* constructor will always register a display() task and unless otherwise specified (using the *registerDefaultTask()* method), it will set it as the default task.

This barebones *display()* method isn't really even necessary since all it does is invoke the parent constructor.

The *JController::display()* method will determine the name of the view and layout from the request and load that view and set the layout. When we create a menu item for our component, the menu manager will allow the administrator to select the view that they would like the menu link to display and to specify the layout. A view usually refers to a view of a certain set of data (i.e. a list of cars, a list of events, a single car, a single event). A layout is a way that that view is organized.

In “**administrator/components/com\_componentname**”folder, we create “*componentname.xml”* file to store information of component’s installation: folder tree structure, version, author… “*install.sql”* an “*uninstall.sql”* are files to store command statements into database when installing and uninstalling the component.

**+ PhpBB 3.0:**

We will use open sources: *iGiveTest* and *PhpBB3*. *PhpBB* is a free flat-forum bulletin board software solution that can be used to stay in touch with a group of people or can power your entire website. No other bulletin board software offers a greater complement of features, while maintaining efficiency and ease of use. Best of all, *phpBB* is **completely** **free**. We use *phpBB* version 3.0 to develop forum’s module.

**+ iGiveTest 2.0:**

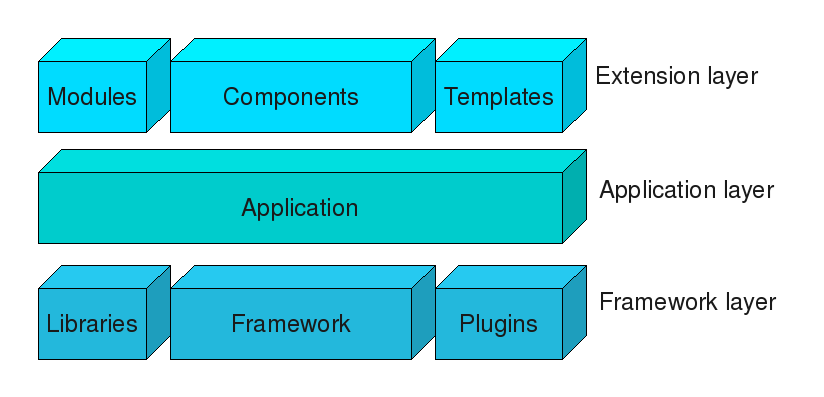
This open source is a comprehensive solution for creating, administering, and providing thorough analysis of tests on the Internet and Intranet. We use the software to develop test’s module and some other functions. The software will support us to create any tests, quizzes and assessments, ask any kind of question. Each test can present questions in random order. You can display one question per page or all questions on one page. Besides grading tests by points or by the number of correct answers, you can define your own grading scale (for example, the 5-grade system A, B, C, D, F, or any other system). Besides it allows you to view testing results for each user or group of users, at a glance and in detail (when the test was taken, how long it took, the answer to each question, points earned for each question, points for the entire test, and so on). Also, you can read and give or change marks for the essay answers.

**+ Joomla Framework:**

We use the framework to develop our components. We will build the framework uponto show the full power and versatility of the MVC design pattern in *Joomla*.

This is an important part of the *Joomla* architecture. It's based on modern object-oriented design patterns that make the *Joomla* core highly maintainable and easily extendable.

Third party developers benefit from the rich and easily accessible functionality that the *Joomla* Framework provides. On this page we'd like to provide you a reference of all classes and respective methods. The links will take you to further information about each class including, where possible, examples of use.

****

**Figure 4.2:** Architecture diagram showing 3-tier architecture of Joomla

*Joomla* is a three tiered system:-

* The top, Extensions layer, consists of [extensions](http://docs.joomla.org/Extension) to the *Joomla* [framework](http://docs.joomla.org/Framework) and its applications:
  + - [Modules](http://docs.joomla.org/Module)
    - [Components](http://docs.joomla.org/Component)
    - [Templates](http://docs.joomla.org/Template)
* The middle, Application layer, consists of applications that extend the Framework [*JApplication*](http://docs.joomla.org/JApplication) class. There are three applications included in the Joomla distribution:
  + - [*JInstallation*](http://docs.joomla.org/JInstallation) is responsible for installing *Joomla* on a web server and is deleted after the installation procedure has been completed.
    - [*JAdministrator*](http://docs.joomla.org/index.php?title=JAdministrator&action=edit&redlink=1) is responsible for the back-end Administrator.
    - [*JSite*](http://docs.joomla.org/JSite) is responsible for the front-end of the website.
* The bottom, Framework layer, consists of:
  + - The Joomla [Framework](http://docs.joomla.org/Framework) itself, whose classes are listed below.
    - [Libraries](http://docs.joomla.org/Library) that are required by the [Framework](http://docs.joomla.org/Framework) or are installed for use by third-party developers.
    - [Plugins](http://docs.joomla.org/Plugin) extend the functionality available in the [Framework](http://docs.joomla.org/Framework).

**+ MySQL:**

We choose *MySql* is our database server because it is completely free. Besides, softwares which we choose as *Joomla*, *PhpBB* and *iGivetest* also use *MySQL*.

*MySQL* is the world's most popular open source database. Whether you are a fast growing web property, technology ISV or large enterprise, *MySQL* can cost-effectively help you deliver high performance, scalable database applications. *MySQL* is a popular choice of database for use in web applications, and is a central component of the widely used [LAMP](http://en.wikipedia.org/wiki/LAMP_(software_bundle)) open source web application software stack—LAMP is an acronym for "[Linux](http://en.wikipedia.org/wiki/Linux), [Apache](http://en.wikipedia.org/wiki/Apache_HTTP_Server), MySQL, [Perl](http://en.wikipedia.org/wiki/Perl)/[PHP](http://en.wikipedia.org/wiki/PHP)/[Python](http://en.wikipedia.org/wiki/Python_(programming_language))".

*MySQL* is an open source database management system and is used in some of the most frequently visited websites on the Internet, including [Flickr](http://en.wikipedia.org/wiki/Flickr) [Nokia.com](http://en.wikipedia.org/wiki/Nokia), [YouTube](http://en.wikipedia.org/wiki/YouTube).

[Free-software](http://en.wikipedia.org/wiki/Free_software)-open source projects that require a full-featured database management system often use *MySQL*. For commercial use, several paid editions are available, and offer additional functionality. Applications which use *MySQL* databases include: [TYPO3](http://en.wikipedia.org/wiki/TYPO3), [Joomla](http://en.wikipedia.org/wiki/Joomla), [WordPress](http://en.wikipedia.org/wiki/WordPress), [phpBB](http://en.wikipedia.org/wiki/PhpBB), [MyBB](http://en.wikipedia.org/wiki/MyBB), [Drupal](http://en.wikipedia.org/wiki/Drupal).

## 4.2.2. Description of System Interface

All screens on the “E-Learning” system use the same format. Since the header, footer, navigation, and toolbar are consistent, the only place where the content will change is in the front page. The area is used to display content available only to the specific page. It will also be used to display a general application error message if the website is unavailable.

*Page Header*: It is displayed at the top of all pages. It contains logo, slogan and menu bar include a set of links about “Thư viện bài tập”, “Thư viện lý thuyết”, “Diễn đàn”, “Đề thi”, “Thành viên”. The header is standard on all “E-Learning” website.

*Navigation:* It displays relevant links to the categories of theory and question that exist on “E-Learning” system, and the affiliate links that are meant to add value to users.

*Page Footer:* It displays the following links:

* About Us
* Contact
* Policies
* Service
* Help

# 4.3. Component Diagram

# 4.4. Detailed Description of Components

## 4.4.1. Theory library

**4.4.1.1. Theory – User Interface Design**

**4.4.1.1.1. Layout**

**4.4.1.1.1.1. Layout – Screen Images**



**Figure 4.3:** “E-Learning” layout page

**4.4.1.1.1.2. Layout – Description of the User Interface**

|  |  |  |
| --- | --- | --- |
| No | Name | Description |
| 1 | Menu bar | Display menu bar which include links to other sites of the website |
| 2 | Body content | Display the content for specific request of user. |
| 3 | “Home page” panel | The panel contains list of categories in “E-Learning” system. |
| 4 | Specify category | Click on the category to access to other sites of the system. |
| 5 | Login text boxes | Allow user to login to the system by entering username and password to text boxes. |
| 6 | Login button | Click on the button to send request to the system. |
| 7 | Forgot password button | Click on the button to send request to get new password. |
| 8 | Forgot username button | Click on the button to send request to find username. |
| 9 | Create account button | Click on the button to register to the system. |
| 10 | New topic area | The area includes links to new topic. |

**4.4.1.1.2. Concern question**

**4.4.1.1.2.1. Concern question – Screen Images**

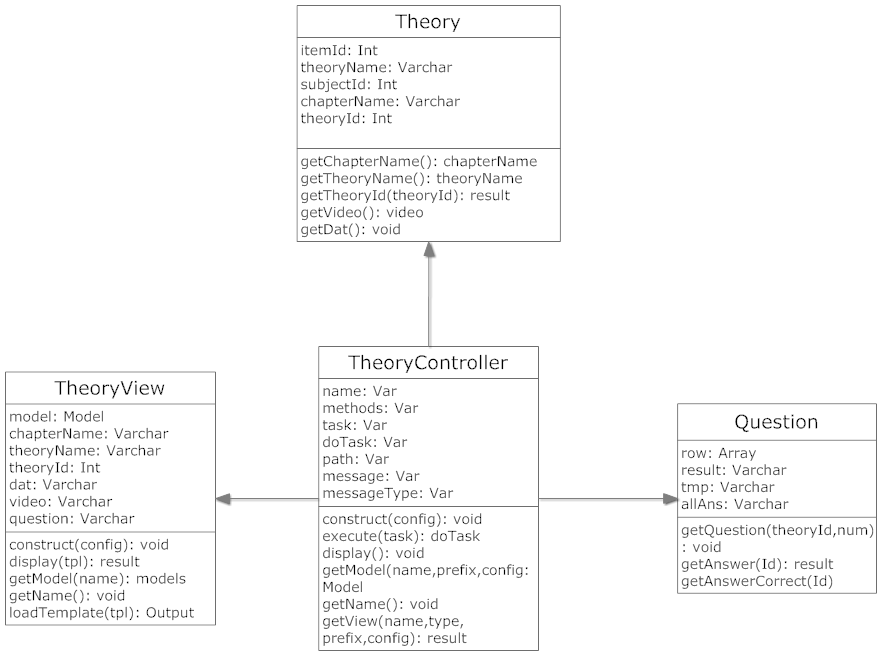


**Figure 4.4:** Question concern of theory panel

**4.4.1.1.2.2. Concern question – Description of the User Interface**

|  |  |  |
| --- | --- | --- |
| No | Name | Description |
| 1 | Question label | Display description of question concern of theory. |
| 2 | Question content | Show content of question. |
| 3 | Question answer | Show answers of the question |
| 4 | Case study | Show case study of the question |

**4.4.1.2. Theory – Class Diagram**



**Figure 4.5:** Theory’s class diagram

**4.4.1.3. Theory – Class Diagram Explanation**

**4.4.1.3.1. Theory class:**

**4.4.1.3.1.1 Attributes:**

|  |  |  |  |
| --- | --- | --- | --- |
| No | Parameter | Type | Description |
| 1 | Itemid | Int | ID of theory, it is set to be unique. |
| 2 | TheoryName | Varchar | Theory name. |
| 3 | Subjectid | Int | ID of subject which has the theory. |
| 4 | Chapter\_name | Varchar | Name of chapter which has the theory. |
| 5 | FileVideoPath | Varchar | Path of video file (if has) of theory |
| 6 | FileDatPath | Varchar | File keeps content of theory. |

**4.4.1.3.1.2. Methods:**

* **Method Get chapter**:

Purpose: get chapter of subject which user selected.

Parameters & return:

|  |  |  |  |
| --- | --- | --- | --- |
| No | Parameter | Type | Description |
| 1 | Subjectid | Int | ID of subject |
|  | <return> | None | None |



**Figure 4.6:** Sequence diagram for get chapter

* **Method Get theory**:

Purpose: get theory which user selected.

Parameters & return:

|  |  |  |  |
| --- | --- | --- | --- |
| No | Parameter | Type | Description |
| 1 | Chapter\_name | Varchar | Name of chapter |
|  | <return> | None | None |

 **Figure 4.7:** Sequence diagram for get theory

**4.4.1.3.2. Question class:**

**4.4.1.3.2.1 Attributes:**

|  |  |  |  |
| --- | --- | --- | --- |
| No | Parameter | Type | Description |
| 1 | Row | Array | Array of row to show question and answers. |
| 2 | Result | Var | Description of question which concern with theory. |
| 3 | Tmp | Var | Keep content of question. |
| 4 | AllAns | Var | Content of answers. |

**4.4.1.3.2.2 Operations:**

* **Method Get question**:
* Purpose: get question which concern of the theory user selected.
* Parameters & return:

|  |  |  |  |
| --- | --- | --- | --- |
| No | Parameter | Type | Description |
| 1 | QuestionID | Int | ID of question |
| 2 | num | Int | Number of question |
|  | <return> | None | None |



**Figure 4.8:** Sequence diagram for get question

* **Method Get answer**:
  + Purpose: get answers which correspond with question.
  + Parameters & return:

|  |  |  |  |
| --- | --- | --- | --- |
| No | Parameter | Type | Description |
| 1 | QuestionID | Int | ID of question |
|  | <return> | Result |  |



**Figure 4.9:** Sequence diagram for get answer

* **Method Get answer correct**:
  + Purpose: get answers is correct.
  + Parameters & return:

|  |  |  |  |
| --- | --- | --- | --- |
| No | Parameter | Type | Description |
| 1 | Id | Int | ID of answer. |
|  | <return> | Result |  |

**4.4.1.3.3. TheoryView class:**

**4.4.1.3.3.1 Attributes:**

|  |  |  |  |
| --- | --- | --- | --- |
| No | Parameter | Type | Description |
| 1 | model | Model | Instance of Model class. |
| 2 | chapterName | Varchar | Name of chapter which includes the theory. |
| 3 | theoryName | Varchar | Name of theory. |
| 4 | theoryId | Int | Id of theory. |
| 5 | Dat | Varchar | Content of theory. |
| 6 | Video | Varchar | Path of video file. |
| 7 | Question | Varchar | Content of question. |

**4.4.1.3.3.2. Methods:**

* **Method Construct:**
  + Purpose: set view name, layout, and charset used by the variable escaping functions.
  + Parameters & return:

|  |  |  |  |
| --- | --- | --- | --- |
| No | Parameter | Type | Description |
| 1 | Config | Array | Array includes configuration of the system. |
|  | <return> | None | None |

* **Method Display** :
  + Purpose: Execute and display a template script, show chapter and content of theory.
  + Parameters & return:

|  |  |  |  |
| --- | --- | --- | --- |
| No | Parameter | Type | Description |
| 1 | Tpl | Var | The name of the template files to parse. |
|  | <return> | Result |  |

* **Method GetModel**:
  + Purpose: method to get the model object.
  + Parameters & return:

|  |  |  |  |
| --- | --- | --- | --- |
| No | Parameter | Type | Description |
| 1 | Name | Var | The name of the model (optional) |
|  | <return> | Methods |  |

* **Method GetName**:
  + Purpose: method to get the name of the model.
  + Parameters & Return:

|  |  |  |  |
| --- | --- | --- | --- |
| No | Parameter | Type | Description |
|  | <return> | Name |  |

* **Method LoadTemplate**:
  + Purpose: Load a template file -- first look in the templates folder for an override.
  + Parameters & Return:

|  |  |  |  |
| --- | --- | --- | --- |
| No | Parameter | Type | Description |
| 1 | Tpl | Var | The name of the template source file. |
|  | <return> | Output | The output of the the template script. |

**4.4.1.3.4. TheoryController class:**

**4.4.1.3.4.1 Attributes:**

|  |  |  |  |
| --- | --- | --- | --- |
| No | Parameter | Type | Description |
| 1 | Name | Var | Name of controller |
| 2 | Methods | Var | Array of class methods |
| 3 | Task | Var | Mapped task that was performed |
| 4 | DoTask | Var | Set of search directories for resources (views) |
| 5 | Path | Var | URL for redirection |
| 6 | Message | Var | Redirect message type |
| 7 | MessageType | Var | Section for the controller |

**4.4.1.3.4.2. Methods:**

* **Method construct:**
  + Purpose: recognized key values include “name”, “default task”, “model path” and “view path” (this list is not meant to be comprehensive).
  + Parameter & Return:

|  |  |  |  |
| --- | --- | --- | --- |
| No | Parameter | Type | Description |
| 1 | Config | Array | Optional associative array of configuration setting |
|  | <return> | Void |  |

* **Method execute**:
  + Purpose: execute a task by triggering a method in the derived class.
  + Parameter & Return:

|  |  |  |  |
| --- | --- | --- | --- |
| No | Parameter | Type | Description |
| 1 | Task | Var | Task to perform. If no matching task is found, the “default” task is executed, if defined. |
|  | <return> | retval |  |

* **Method display**:
  + Purpose: typical view method for MVC based architecture. This method is provided as a default implementation, in most cases we will need to override it in our own controllers.
  + Parameter & Return:

|  |  |  |  |
| --- | --- | --- | --- |
| No | Parameter | Type | Description |
|  | <return> | Void |  |

* **Method getModel**:
  + Purpose: to get model to implement.
  + Parameter & Return:

|  |  |  |  |
| --- | --- | --- | --- |
| No | Parameter | Type | Description |
| 1 | Name | Var | Model name |
| 2 | Prefix | Var | Class prefix |
| 3 | Config | Array | Configuration array for model |
|  | <return> | Model | Object model |

* **Method getName**:
  + Purpose: method to get the controller name.
  + Parameter & Return:

|  |  |  |  |
| --- | --- | --- | --- |
| No | Parameter | Type | Description |
|  | <return> | name | The name of the dispatcher |

* **Method getView**:
  + Purpose: method to get a reference to the current view and load it if necessary.
  + Parameter & Return:

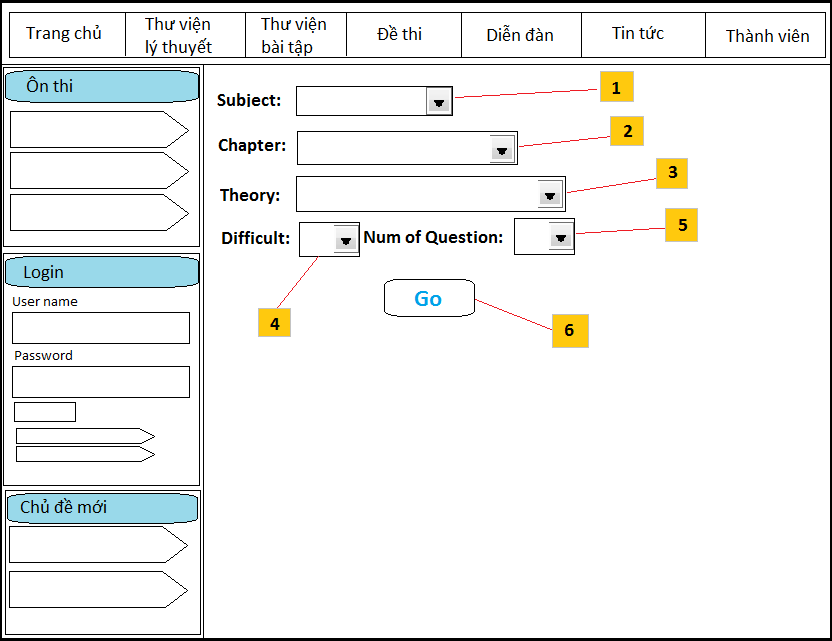
|  |  |  |  |
| --- | --- | --- | --- |
| No | Parameter | Type | Description |
| 1 | Name | Var | The view name, optional, defaults to the controller name. |
| 2 | Type | Var | The view type |
| 3 | Prefix | Var | The class prefix |
| 4 | Config | Array | Configuration array for view |
|  | <return> | Result | Reference to the view or an error |

## 4.4.2. Exercise library

**4.4.2.1. Exercise – User Interface**

**4.4.2.1.1. Choice Question**

**4.4.2.1.1.1. Choice Question – Screen Images**



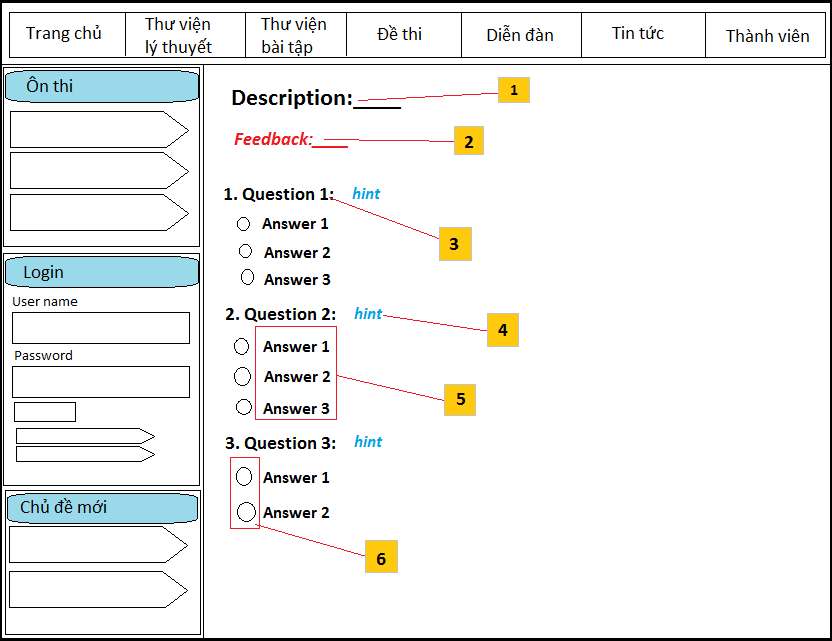
**Figure 4.10:** Choice question page

**4.4.2.1.1.1. Choice Question – Description of the User Interface**

|  |  |  |
| --- | --- | --- |
| No | Name | Description |
| 1 | List box subject | Allow user to choose subject. |
| 2 | List box chapter | Allow user to choose chapter of subject. |
| 3 | List box theory | Allow user to choose theory of chapter. |
| 4 | List box difficult | Allow user to choose difficult to do. |
| 5 | List box number of question | Allow user to choose number of question to do |
| 6 | Button Go | Click on button to start doing exercises. |

**4.4.2.1.2. Do exercise**

**4.4.2.1.2.1. Do exercise – Screen Images**

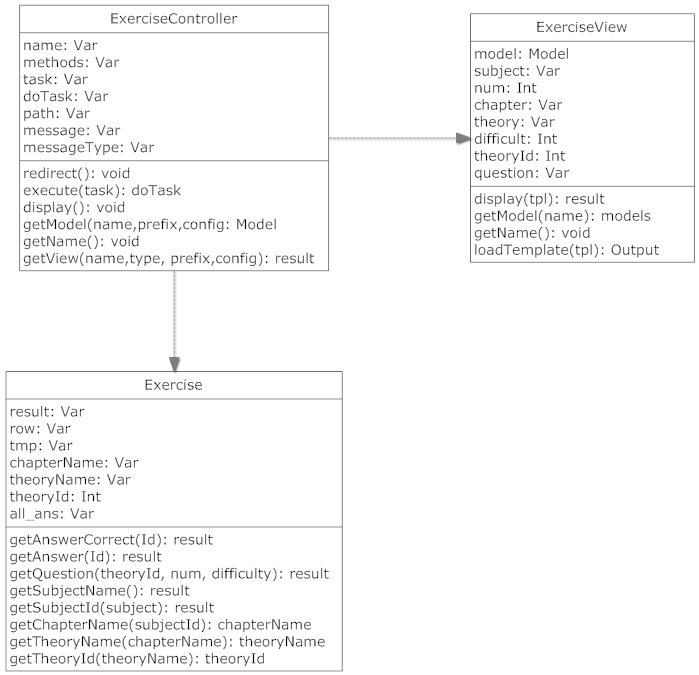


**Figure 4.11:** Do exercise page

**4.4.2.1.2.1. Do exercise – Description of the User Interface**

|  |  |  |
| --- | --- | --- |
| No | Name | Description |
| 1 | Description label | Show description of exercise includes: subject, chapter, theory, difficult which user choose. |
| 2 | Feedback label | Show feedback of the system about the exercise. |
| 3 | Content of question | Content of question |
| 4 | Button hint | Click on button to show case study following to the question. |
| 5 | Content of answers | Content of answers of question. |
| 6 | Check box answer | Click on check box to choose answers of user. |

**4.4.2.2. Exercise – Class Diagram**

****

**Figure 4.12:** Exercise’s class diagram

**4.4.2.3. Exercise – Class Diagram Explanation**

**4.4.2.3.1. Exercise class:**

**4.4.2.3.1.1 Attributes:**

|  |  |  |  |
| --- | --- | --- | --- |
| No | Parameter | Type | Description |
| 1 | Result | Var | Store content of questions and answers which are displayed on front page. |
| 2 | Row | Array | Array of record concern with each of question and answer. |
| 3 | Tmp | Var | Record of question and answers. |
| 4 | ChapterName | Var | Name of chapter |
| 5 | TheoryName | Var | Name of theory |
| 6 | TheoryId | Int | Id of theory |
| 7 | AllAns | Var | Content of answers of question. |

**4.4.2.3.1.2. Methods:**

* **Method getAnswerCorrect:**
* Purpose: get answer which is correct.
* Parameters & return:

|  |  |  |  |
| --- | --- | --- | --- |
| No | Parameter | Type | Description |
| 1 | Id | Int | ID of question |
|  | <return> | Result |  |

* **Method getAnswer:**
* Purpose: get content of answer
* Parameters & return:

|  |  |  |  |
| --- | --- | --- | --- |
| No | Parameter | Type | Description |
| 1 | Id | Int | ID of question |
|  | <return> | Result |  |

* **Method getQuestion:**
* Purpose: get question by theory, number of questions and difficult.
* Parameters & return:

|  |  |  |  |
| --- | --- | --- | --- |
| No | Parameter | Type | Description |
| 1 | theoryId | Int | ID of theory |
| 2 | num | Int | Number of question |
| 3 | difficulty | Int | Difficulty of question. |
|  | <return> | Result |  |

* **Method getSubjectName**:
* Purpose: get name of subject.
* Parameters & return:

|  |  |  |  |
| --- | --- | --- | --- |
| No | Parameter | Type | Description |
| 1 |  |  |  |
|  | <return> | Result |  |

* **Method getSubjectId**:
* Purpose: get ID of subject
* Parameters & return:

|  |  |  |  |
| --- | --- | --- | --- |
| No | Parameter | Type | Description |
| 1 | subject | Var | Name of subject |
|  | <return> | result |  |

* **Method getChapterName**:
* Purpose: get name of chapter
* Parameters & return:

|  |  |  |  |
| --- | --- | --- | --- |
| No | Parameter | Type | Description |
| 1 | subjectId | Int | ID of subject |
|  | <return> | chapterName |  |

* **Method getTheoryName**:
* Purpose: get name of theory
* Parameters & return:

|  |  |  |  |
| --- | --- | --- | --- |
| No | Parameter | Type | Description |
| 1 | chapterName | Var | Name of chapter |
|  | <return> | theoryName |  |

* **Method getTheoryId**:
* Purpose: get ID of theory
* Parameters & return:

|  |  |  |  |
| --- | --- | --- | --- |
| No | Parameter | Type | Description |
| 1 | theoryName | Var | Name of theory |
|  | <return> | theoryId |  |

**4.4.2.3.2. ExerciseController class:**

**4.4.2.3.2.1 Attributes:**

|  |  |  |  |
| --- | --- | --- | --- |
| No | Parameter | Type | Description |
| 1 | Name | Var | Name of controller |
| 2 | Method | Var | Array of class methods |
| 3 | Task | Var | Mapped task that was performed |
| 4 | doTask | Var | Set of search directories for resources (views) |
| 5 | Path | Var | URL for redirection |
| 6 | Message | Var | Redirect message type |
| 7 | messageType | Var | Section for the controller |

**4.4.2.3.2.2. Methods:**

* **Method redirect:**
  + - Purpose: Redirects the browser or returns false if no redirect is set.
    - Parameters & return:

|  |  |  |  |
| --- | --- | --- | --- |
| No | Parameter | Type | Description |
| 1 |  |  |  |
|  | <return> | Boolean |  |

* **Method execute:**
* Purpose: Execute a task by triggering a method in the derived class.
* Parameters & return:

|  |  |  |  |
| --- | --- | --- | --- |
| No | Parameter | Type | Description |
| 1 | Task | Var | The task to perform. If no matching task is found, the default' task is executed, if defined. |
|  | <return> | retval |  |

* **Method display** :
  + - * Purpose: display view of content in front page.
      * Parameters & return:

|  |  |  |  |
| --- | --- | --- | --- |
| No | Parameter | Type | Description |
| 1 |  |  |  |
|  | <return> | None |  |

* **Method getModel**:
* Purpose: to get model to implement.
* Parameters & return:

|  |  |  |  |
| --- | --- | --- | --- |
| No | Parameter | Type | Description |
| 1 | Name | Var | Model name |
| 2 | Prefix | Var | Class prefix |
| 3 | Config | Array | Configuration array for model |
|  | <return> | Model | Object model |

* **Method getName:**
* Purpose: method to get the controller name.
* Parameters & return:

|  |  |  |  |
| --- | --- | --- | --- |
| No | Parameter | Type | Description |
| 1 |  |  |  |
|  | <return> | name |  |

* **Method getView:**
* Purpose: method to get a reference to the current view and load it if necessary.
* Parameters & return:

|  |  |  |  |
| --- | --- | --- | --- |
| No | Parameter | Type | Description |
| 1 | Name | Var | The view name, optional, defaults to the controller name. |
| 2 | Type | Var | The view type |
| 3 | Prefix | Var | The class prefix |
| 4 | Config | Array | Configuration array for view |
|  | <return> | Result | Reference to the view or an error |

**4.4.2.3.3. ExerciseView class:**

**4.4.2.3.3.1 Attributes:**

|  |  |  |  |
| --- | --- | --- | --- |
| No | Parameter | Type | Description |
| 1 | Model | Model | The base path of the view |
| 2 | Subject | Var | Name of subject |
| 3 | Num | Int | Number of question |
| 4 | Chapter | Var | Name of chapter |
| 5 | Theory | Var | Name of theory |
| 6 | Difficult | Int | Difficulty of question |
| 7 | TheoryId | Int | ID of theory |
| 8 | Question | Var | Content of question |

4.4.2.3.3.2. **Methods**:

* **Method display**:
  + Purpose: display content of exercise site in fornt page
  + Parameters & return:

|  |  |  |  |
| --- | --- | --- | --- |
| No | Parameter | Type | Description |
| 1 | tpl | Var | Content of exercise |
|  | <return> | Result | Content of exercise site. |

* **Method getModel**:
  + Purpose: Method to get the model object
  + Parameters & return:

|  |  |  |  |
| --- | --- | --- | --- |
| No | Parameter | Type | Description |
| 1 | name | Var | The name of the model |
|  | <return> | name |  |

* **Method getName**:
  + Purpose: Method to get the view name
  + Parameters & return:

|  |  |  |  |
| --- | --- | --- | --- |
| No | Parameter | Type | Description |
| 1 |  |  |  |
|  | <return> | name | The name of the model |

* **Method loadTemplate**:
  + Purpose: Load a template file -- first look in the templates folder for an override
  + Parameters & return:

|  |  |  |  |
| --- | --- | --- | --- |
| No | Parameter | Type | Description |
| 1 | tpl | Var | The name of the template source file |
|  | <return> | output | The output of the template script. |

## 4.4.3. Open sources integration



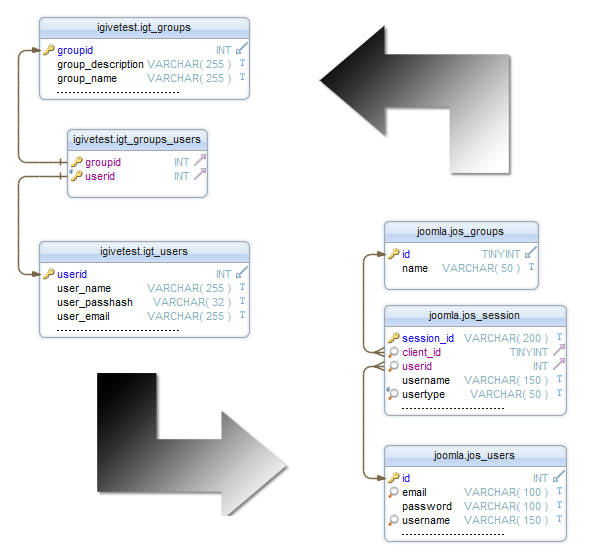
**Figure 4.13:** Synchronization Joomla, Phpbb and iGiveTest

**4.4.3.1. Synchronization databases**

**4.4.3.1.1 Joomla vs iGiveTest**

After analyzing database of *iGiveTest* and *Joomla*, we see they have same tables: users and groups. When installing *Joomla*, tables will have prefix “jos\_”, but *iGiveTest* we can choose any prefix for the open source. We need synchronize 2 open sources which use a common database with database of *iGiveTest* has prefix is “igt\_” and Database of *Joomla* has prefix is “jos\_”.

The pattern below shows table users and table groups of 2 open sources and the relationship of the tables:



**Figure 4.14:** Synchronization table users, groups of iGiveTest and Joomla

**Description of tables and columns and explanation:**

1. Table igivetest.igt\_users:

The table stores information of users. It has more column but we only describe some important column which we want to use.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Column name | Data type | Nullable | Description |
| 1 | Userid | INT | No | Unique identifier of user |
| 2 | User\_name | VARCHAR(255) | No | Username |
| 3 | User\_passhash | VARCHAR(32) | No | Password of user which is encrypted MD5 |
| 4 | User\_email | VARCHAR(255) | No | Email of user |

1. Table igivetest.igt\_groups\_users:

The table is joined by table users and table groups with 2 important columns: userid and groupid to set authority for users.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Column name | Data type | Nullable | Description |
| 1 | Userid | INT | No | Unique identifier of user |
| 2 | Groupid | INT | No | Unique identifier of group |

1. Table igivetest.igt\_groups:

The table stores information of user’s groups. For example: admin, user, guest, teacher…. The table has more columns but we only use the below column.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Column name | Data type | Nullable | Description |
| 1 | Groupid | INT | No | Unique identifier of user’s group |
| 2 | Group\_description | VARCHAR(255) | No | Description of group |
| 3 | Group\_name | VARCHAR(255) | No | Name of group |

1. Table joomla.jos\_groups:

The table stores access level of articles or sites. For example: all of users can view sites if they are at “public”, users have login to the website will view sites which are at “registered”.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Column name | Data type | Nullable | Description |
| 1 | Id | TINYINT | No | Unique of access level’s group |
| 2 | Name | VARCHAR(50) | No | Name of access level’s group |

1. Table joomla.jos\_session:

The table stores session of users when access to the website.

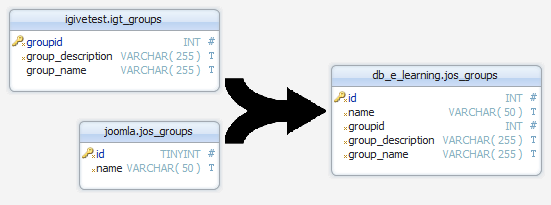
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Column name | Data type | Nullable | Description |
| 1 | Session\_id | VARCHAR(200) | No | Session when user access the website |
| 2 | Client\_id | TINYINT | No | Unique identifier of client correspond with groupid |
| 3 | Userid | INT | No | Unique identifier of user |
| 4 | Username | VARCHAR(150) | No | Username of user |
| 5 | Usertype | VARCHAR(50) | No | Type of user |

1. Table joomla.jos\_users:

The table stores information of user. It has more columns but we only describe some important column which we use.

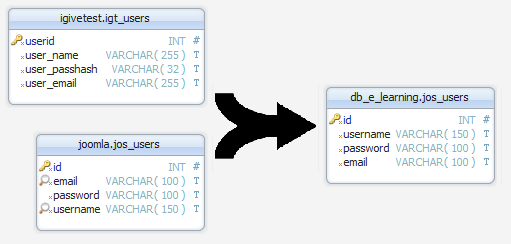
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Column name | Data type | Nullable | Description |
| 1 | Id | INT | No | Unique identifier of user |
| 2 | Email | VARCHAR(100) | No | Email of user |
| 3 | Password | VARCHAR(100) | No | Password |
| 4 | Username | VARCHAR(100) | No | Username |

We will merge 2 databases with a prefix “jos” to correspond with Joomla framework. There are common tables: groups and users. We will process as follow:



**Figure 4.15:** Merge table iGiveTest’s groups and table Joomla’s groups

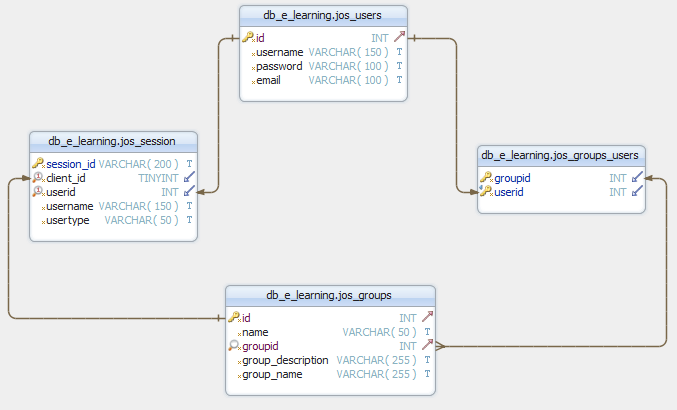
Because two tables don’t have columns which have the same name. We can merge them as above.



**Figure 4.16:** Merge table iGiveTest’s users and table Joomla’s users

The tables have columns which store the same information. The system only need use one of them. Because the system develops on Joomla framework, name of the columns should follow Joomla’s definition. In iGiveTest’s source code, we will edit variables: “*userid*” to “*id*”, “*user\_name*” to “*username*”, “*user\_passhash*” to “*password*” and “*user\_email*” to “*email*”. Other columns don’t change. Then, we merge the tables.

There is the result after merging the common tables of the database system about user:



**Figure 4.17:** Database schema of users and groups after merging

**4.4.3.1.2 Joomla vs Phpbb**

We will synchronize login/logout of *Joomla* and *Pbpbb* by an extension of *Joomla*. We will talk about it below.

**4.4.3.2. Implement tools.**

To synchronize *Joomla* and *Phpbb*, we will use extensions of *Joomla*: ***RokBridge***.

**What is RokBridge?**

*RokBridge* is a non-invasive, bi-directional bridge for *Joomla* and the forum platform *phpBB3*. By following a few simple sets, you can easily integrate the two platforms, providing registration, user syncing and syncing of login sessions.

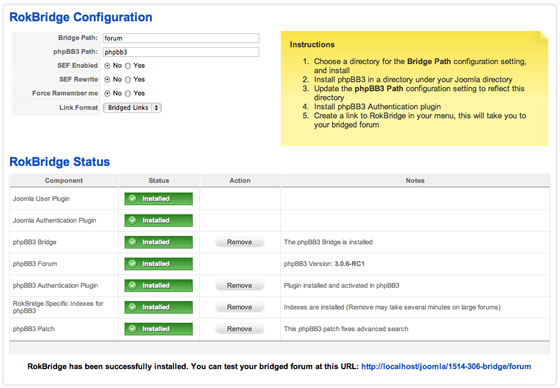
*RokBridge* has been developed with the help of key members of the *phpBB3* team and 100% GPL and Free.

**How to use?**

We use *RokBridge* to take care of the user synchronization login/logout synchronization from *Joomla* to *phpBB3*.

When a user creates an account via *phpBB3*, the user's password is stored in *phpBB3*. If they create an account in *Joomla*, the password is stored in *Joomla*. If they created an account in phpBB3 and try to change it in *Joomla*, it will update the password on *phpBB3* for them, however phpBB3 does not have a 'change password event', so if they create an account in *Joomla* and change the password in *phpBB3*, it doesn't know to update their *Joomla* password. Things can get a bit screwy. We suggest to only allowing users to change passwords in *Joomla*. To ensure this, we will disable the ability in *phpBB3* via the user profile by going to the ACP -> System -> Module Management -> User Control Panel -> Profile, and then disabling Account Settings.

The interface of the extension which install successfully will display below:



**Figure 4.18:** RokBridge interface

**4.4.3.3. Integration modules**

**4.4.3.3.1. Get password in Phpbb**

Because three open sources use different hashing algorithms, users can’t login in test site with their password. For example, a user registers to the website with username: “guest”, password: “guest”. The password will be hashed by *Phpbb* as: “$H$92gtO3cr4hj5gNY5jlvhJhJjqjT9Eq1”. With *iGiveTest*, it will be hashed: “df6f58808ebfd3e609c234cf2283a989”. So, the user only login to *Phpbb* but he can’t login to test site (*iGiveTest*).

So we need modify source code of *Phpbb* and *iGiveTest* to login to three open sources. We do tasks below:

+ Editing phpbb\_hash function:

The function is in “**e-learning-website/phpBB3/includes/functions.php**”. It’s used to hash password as a string. The string will be compared with user’s password stored in database. The algorithms of *Phpbb* as follow:

|  |
| --- |
| \*  \* *Hash the password in PhpBB*  \*  function phpbb\_hash($password)  {  for ($i = 0; $i < $count; $i += 16)  {  $random\_state = md5(unique\_id() . $random\_state);  $random .= pack('H\*', md5($random\_state));  }  $random = substr($random, 0, $count);  }    $hash = \_hash\_crypt\_private($password, \_hash\_gensalt\_private($random, $itoa64),  $itoa64);    if (strlen($hash) == 34)  {  return $hash;  } |
| } |

We will modify the function below:

|  |
| --- |
| \*  \* *Hash the password in PhpBB after modifying*  \*  function phpbb\_hash($password)  {  return md5($password);  } |

+ Editing signinUser function:

The file includes the function is placed in “**e-learning-website/exam/inc/functions.inc.php**”. We will add a query to get password in table *phpbb\_users* when users login to test. The query is below:

|  |
| --- |
| /\*  \* *To get password in table phpbb\_users*  \*/  $sql\_str = "SELECT \* FROM ".$srv\_settings['table\_prefix']."users  LEFT JOIN `phpbb\_users`  ON jos\_users.`username` = phpbb\_users.`username`  WHERE jos\_users.`username`=".$i\_username."  AND phpbb\_users.`user\_password` =".$i\_pass\_hash."  AND jos\_users.`user\_enabled`=1  AND (jos\_users.`user\_expiredate`=0  OR jos\_users.`user\_expiredate`>".$i\_time.")"; |

**4.4.3.3.2. Add authority for user.**

The system use iGiveTest’s open source to develop test site. When synchronizing open sources, we removed register’s module of *iGiveTest*. The module will auto set authority for user. The system only allow user to register by *Phpbb.*

When a user want to do a test, he need login to test site. At the first time to login, if successfully, the system will add authority for user to do a test and view result.

The system will check ID of user into table *jos\_groups\_users.* The table includes 2 columns: userid and groupid. Inside, groupid stores ID of authority of user. In test site, each of group will have different features which the system supports. *iGiveTest* defines authority of user below:

|  |  |
| --- | --- |
| Groupid | Description |
| 1 | Administrator |
| 2 | Teacher |
| 19 | User |
| 20 | Guest |

**Figure 4.19:** Description of groupid in iGiveTest

We will do tasks: getting ID of username and checking if ID exists in table jos\_groups\_users in singinUser function. The function is placed in “**e-learning-website/exam/inc/function.inc.php**”. We will develop code for the tasks as follow:

|  |
| --- |
| *\**  *\*get userid of username*  *\**  $sql\_str1 = "SELECT id FROM jos\_users WHERE username=".$i\_username;  $result1 = mysql\_query($sql\_str1);  while($row = mysql\_fetch\_array($result1))  {  $id = $row['id'];  } |

|  |
| --- |
| \*  \**check userid exist in jos\_group\_user*  *\*if not, adding groupid = ‘19’ for userid*  \*  $sql\_str = "SELECT \* FROM jos\_groups\_users WHERE id=".$id;  $result2 = mysql\_query($sql\_str);  while($row = mysql\_fetch\_array($result2))  {  $groupid = $row['groupid'];  }  $groupid = $i\_rSet1->fields['groupid'];  if ($groupid == "")  {    mysql\_query("INSERT INTO jos\_groups\_users (groupid, id)  VALUES ('19', $id)");  } |

When a user logins at the first time to test, the system will automatically set the value of *groupid* is ‘19’. It is equivalent of user.

**4.4.3.3.3. Logout the system.**

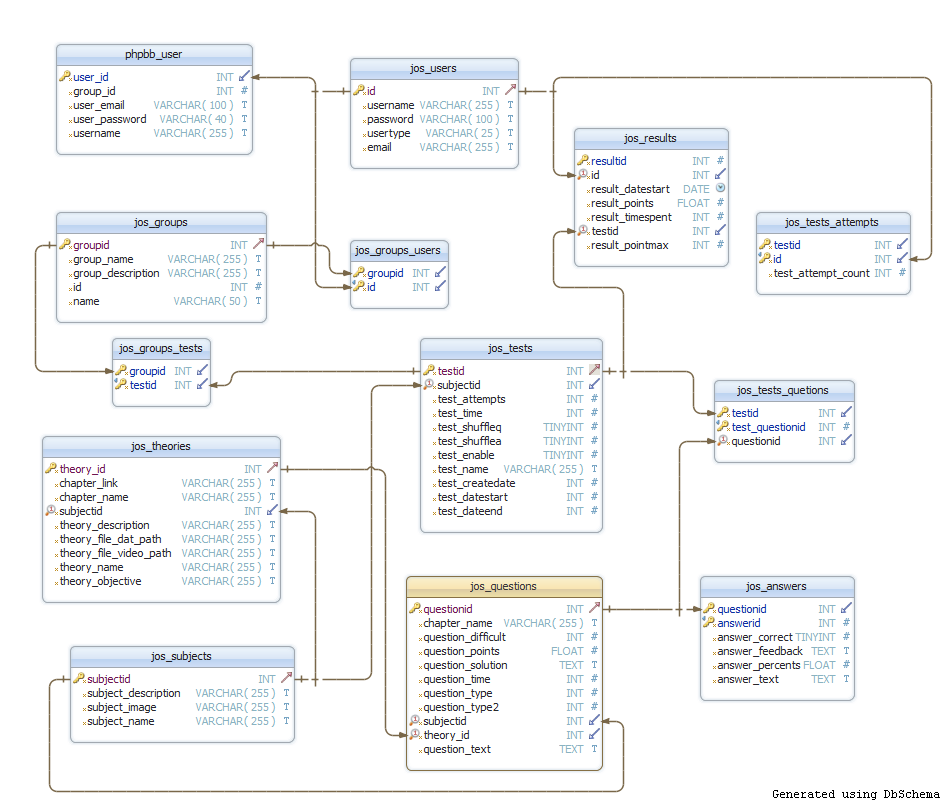
Because the system uses different open sources, each of them must require user to click on button “Sign Out” to logout the website. We develop a function to help user to click once button “Sign Out” to logout the system.

In file “*default.php*” is placed in “**e-learning-website/modules/mod\_login/tmpl/**”, we put the code below:

|  |
| --- |
| \*  \**Call to signout function of iGiveTest*  \*  echo "<script type=\"text/javascript\" src=\"exam/signout.php\"></script>"; |

# 4.5. Database Design or Data Structures

## 4.5.1. Detailed database design for “E-Learning” system:



**Figure 4.20:** Detailed database design for “E-Learning” system

## 4.5.2. Table and columns description and explanation

1. Table phpbb\_users:

This table is built by *Phpbb*. We use the table to store information of user. When user register or edit privacy information, data will be stored in the table. It has 60 column but we only use 5 important columns below to develop the system:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Column name | Data type | Nullable | Description |
| 1 | User\_id | INT | No | Unique identifier of user |
| 2 | Group\_id | INT | No | Unique identifier of user’s group |
| 3 | User\_email | VARCHAR(100) | No | Email of user. |
| 4 | User\_password | VARCHAR(40) | No | User’s password to login to the website. |
| 5 | username | VARCHAR(255) | No | Username of user to login to the website. |

1. Table jos\_users:

Because *Joomla* and *iGiveTest* also have database about user, we synchronized data from user’s table of *Joomla* with user’s table of *iGiveTest* to create the table. It has more column but we only describe some important column.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Column name | Data type | Nullable | Description |
| 1 | Id | INT | No | Unique identifier of user |
| 2 | Username | VARCHAR(255) | No | Username of user to login to the website. |
| 3 | Password | VARCHAR(100) | No | User’s password to login to the website. |
| 4 | Usertype | VARCHAR(25) | No | User’s group |
| 5 | Email | VARCHAR(255) | No | Email of user. |

1. Table jos\_results:

This table stores result about test of user. We use the table to count.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Column name | Data type | Nullable | Description |
| 1 | Resultid | INT | No | Unique identifier of result |
| 2 | Testid | INT | No | Unique identifier of test |
| 3 | Id | INT | No | Unique identifier of user which have report of test result |
| 4 | Result\_datestart | INT | No | Date which result of test is started |
| 5 | Result\_timespend | INT | No | Time which user spend to test |
| 6 | Result\_points | FLOAT | No | Point of test of user |
| 7 | Result\_pointmax | FLOAT | No | Max point of test |

1. Table jos\_tests\_attempts:

This table stores number of attempts to take a test.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Column name | Data type | Nullable | Description |
| 1 | Testid | INT | No | Unique identifier of test |
| 2 | Userid | INT | No | Unique identifier of user which takes the test |
| 3 | Test\_attempt\_count | INT | No | Number of attempts to take the test |

1. Table jos\_groups:

This table stores information of authorities which correspond with user’s groups.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Column name | Data type | Nullable | Description |
| 1 | Groupid | INT | No | Unique identifier of iGiveTest’s group. |
| 2 | Group\_name | VARCHAR(255) | No | Name of iGiveTest’s group |
| 3 | Group\_description | VARCHAR(255) | No | Description of iGiveTest’s group |
| 4 | Id | INT | No | Unique identifier of Joomla’s user group |
| 5 | Name | VARCHAR(50) | No | Name of Joomla’s user group |

1. Table jos\_groups\_users:

This table is joined with groups table, users table by groupid and userid.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Column name | Data type | Nullable | Description |
| 1 | Groupid | INT | No | Unique identifier of iGiveTest’s user group. |
| 2 | Userid | INT | No | Unique identifier of user |

1. Table jos\_groups\_tests:

This table is joined with groups table, tests table by groupid and testid.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Column name | Data type | Nullable | Description |
| 1 | Groupid | INT | No | Unique identifier of iGiveTest’s user group. |
| 2 | Testid | INT | No | Unique identifier of test |

1. Table jos\_tests:

This table stores information of tests: test name, test time, date which test is started, shuffling question, number of user’s attempts…

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Column name | Data type | Nullable | Description |
| 1 | Testid | INT | No | Unique identifier of test |
| 2 | Subjectid | INT | No | Unique identifier of subject |
| 3 | Test\_name | VARCHAR(255) | No | Name of test |
| 4 | Test\_time | INT | No | Time which test enable |
| 5 | Test\_datestart | INT | No | Date which test is started |
| 6 | Test\_dateend | INT | No | Date which test ends |
| 7 | Test\_shuffleq | TINYINT | No | Shuffling questions |
| 8 | Test\_shufflea | TINYINT | No | Shuffling answers |
| 9 | Test\_createdate | INT | No | Date which test are created |
| 10 | Test\_enable | TINYINT | No | State of test |
| 11 | Test\_attempts | INT | No | Number of attempts are allowed |

1. Table jos\_tests\_questions:

This table stores questions of tests.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Column name | Data type | Nullable | Description |
| 1 | Test\_questionid | INT | No | Position of questions in a test |
| 2 | Testid | INT | No | Unique identifier of test |
| 3 | Test\_sectionid | INT | No | Section of questions in a test |
| 4 | questionid | INT | No | Unique identifier of question which is taken in question bank |

1. Table jos\_theories:

This table stores information of theories. It includes: theory name, theory description, objective, reference questions…

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Column name | Data type | Nullable | Description |
| 1 | Theory\_id | INT | No | Unique identifier of theory. |
| 2 | Theory\_description | VARCHAR(255) | No | Description of theory |
| 3 | Theory\_name | VARCHAR(255) | No | Name of theory |
| 4 | Theory\_Objective | VARCHAR(255) | No | Objectives which theory bring out |
| 5 | Theory\_file\_dat\_path | VARCHAR(255) | No | Content of theory |
| 6 | Theory\_file\_video\_path | VARCHAR(255) | No | Path of theory source (video, text…) |
| 7 | Subjectid | INT | No | Unique identifier of subject of theory |
| 8 | Chapter\_link | VARCHAR(255) | No | Link to site which includes chapters |
| 9 | Chapter\_name | VARCHAR(255) | No | Name of chapter which includes the theory. |

1. Table jos\_subjects:

This table stores information of subjects.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Column name | Data type | Nullable | Description |
| 1 | Subjectid | INT | No | Unique identifier of subject |
| 2 | Subject\_description | VARCHAR(255) | No | Description of subject |
| 3 | Subject\_image | VARCHAR(255) | No | Link to subject image file |
| 4 | Subject\_name | VARCHAR(255) | No | Name of subject |

1. Table jos\_questions:

This table is considered as question bank. It includes information of questions: question time, content of question, point of question…

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Column name | Data type | Nullable | Description |
| 1 | Questionid | INT | No | Unique identifier of question |
| 2 | Subjectid | INT | No | Unique identifier of subject about test |
| 3 | Question\_time | INT | No | Time which question is finished |
| 4 | Question\_text | TEXT | No | Content of question |
| 5 | Question\_points | TEXT | No | Point of question |
| 6 | Question\_solution | TEXT | No | Solution of question |
| 7 | Question\_type | INT | No | Type of question |
| 8 | Question\_type2 | TINYINT | No | Allowing partially correct answers if Question\_Type is “Multiple Answer” |
| 9 | Theory\_id | INT | No | Unique identifier of theory which correspond with question. |
| 10 | Chapter\_name | VARCHAR(255) | No | Name of chapter |
| 11 | Question\_difficult | INT | No | Difficulty of question |

1. Table jos\_nswers:

This table stores answers of each question.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Column name | Data type | Nullable | Description |
| 1 | Answerid | INT | No | Unique identifier of answer concern with each question |
| 2 | Questionid | INT | No | Unique identifier of question of answer |
| 3 | Answer\_text | TEXT | No | Content of answer |
| 4 | Answer\_feedback | TEXT | No | Feedbacks of answer (if have) |
| 5 | Answer\_correct | TINYINT | No | Answer is correct |
| 6 | Answer\_percents | FLOAT | No | Percentage correct of answer |

V> Report No.5: Software Test Document

# 5.1. Introduction

## 5.1.1. Purpose

This document is prepared as the test documentation for project , in scope of Capstone Project at FPT University

## 5.1.2. Reference

|  |  |  |  |
| --- | --- | --- | --- |
| # | Title | Version | Description |
| 1 | Report #1 – Project Description | 1.0 | Description of E-learning Project |
| 2 | Report #2 – Software Project Management Plan | 1.0 | Software Project Management Plan of E-learning Project |
| 3 | Report #3 – Software Requirement Specification | 1.0 | Software Requirement Specification of E-learning Project |
| 4 | Report #4 – Software Detailed Design | 1.0 | Software Detailed Design of E-learning Project |

# 5.2. Test plan

## 5.2.1. Feature to be tested

|  |  |  |
| --- | --- | --- |
| Component Name | Function | Description |
| **Theory library** | View subject name | User views all subject name on the main menu |
| View chapter name | User views all chapter names after select subject by header link on content area. |
| View theory name | User views all theory names after click header by header link on content area. |
| View theory content | User view content of theory on content area after click theory name. |
| View video of theory | User view video in theory content on content area. |
| View questions of theory | User does exercise. |
| **Exercise library** | Select subject name | User select subject name by combo box. |
| Select chapter name | User select chapter name by combo box. |
| Select theory name | User select theory name by combo box. |
| Select number question | User select number question by combo box. |
| Select difficulty of question | User select difficulty of question by combo box. |
| View questions follow selections | User view questions content on content area. |
| View answer of question | User view answer when click header link”Hint”. |
| **Exam** | Login on the home site | Login to show tab “Đề thi” |
| Authenticate account user before do exam. | Authenticate account to do exam. |
| Do exam in site “Làm bài thi” | Show all exams and do exam. |
| Calculate result of exam | Calculate result of exam |
| Do exam by account Admin | Do exam by account admin like user. |
| Add new question for a new exam | Admin can make question for exam. |
| Add new subject | Admin add subject for exam. |
| Manage account in exam. | Admin manage account in exam |
| Make new exam | Admin make new exam. |
| Manage Report | Admin manage report. |
| **Forum** | Login forum | Login forum and auto login in home site |
| Logout forum | Logout forum and auto login in home site |
| Post new thread in forum | Make new thread in forum |
| Change profile account user | Change profile account user |
| Component Name | Function | Description |
| **Theory library** | View subject name | User view all subject name on the main menu |
| View chapter name | User views all chapter names after select subject by header link on content area. |
| View theory name | User views all theory names after click header by header link on content area. |
| View theory content | User view content of theory on content area after click theory name. |
| View video of theory | User view video in theory content on content area. |
| View questions of theory | User does exercise. |
| **Exercise library** | Select subject name | User select subject name by combo box. |
| Select chapter name | User select chapter name by combo box. |
| Select theory name | User select theory name by combo box. |
| Select number question | User select number question by combo box. |
| Select difficulty of question | User select difficulty of question by combo box. |
| View questions follow selections | User view questions content on content area. |
| View answer of question | User view answer when click header link”Hint”. |

## 5.2.2. Feature not to be tested

|  |  |  |
| --- | --- | --- |
| Component Name | Function | Description |
| Exam site | Manage profile account user | Change and show all profile of account user. |
| Read repost account | Read repost from site exam. |
| Change user group | Change user group to delegate user |
| Show member online | Show member online if login by admin account |
| Forum site | Create new subject | Create new subject |
| Forget password “Quên mật khẩu” | Send password to mail of account |
| View your post ”Xem bài viết của bạn” | View post of account |
| View thread unanswered ”Xem bài viết chưa trả lời” | View thread unanswered |
|  | View thread unread post “Xem bài viết chưa xem” | View thread unread post |
|  | View new post ”Xem bài viết mới” | View new post |
|  | Mark all forum as read “Đánh dấu tất cả các chuyên mục ” | Mark all forum as read |
|  | View active topic “Xem chủ đề đang hoạt động” | View active topic |
|  | Delete cookies of forum | Delete cookies of forum |
|  | Help “Trợ giúp” | All help to member |
| Home site | News in front page | Show news in content area |
| New thread in forum | Show subject of new thread in left menu |
| Show member online | Show member online in left menu |

# 5.3. Test strategy

## 5.3.1. Test types

### 5.3.1.1. Function Testing

The objective of function test is to measure the quality of the functional components of the system like verify proper data acceptance, processing, and retrieval, and the appropriate implementation of the business rules.  This type of testing is based upon black box testing that is verifying the application and its internal processes by interacting with the application

Table identified below is an outline of the testing recommended for each component of e-learning website:

|  |  |
| --- | --- |
| Test Objective | Verify proper target-of-test functionality, including navigation, data entry, processing, and retrieval. |
| Technique | Execute each use case, use-case flow, or function, using valid and invalid data, to verify the following:   * The expected results occur when valid data is used. * The appropriate error or warning messages are displayed when invalid data is used. * Each business rule is properly applied. * Some case for exception occur when special data is used |
| Completion Criteria | * All planned tests have been executed. * All identified defects have been addressed and closed |
| Special Considerations | Identify or describe those items or issues (internal or external) that impact the implementation and execution of function test |

### 

### 5.3.1.2. User Interface Testing

User Interface (UI) testing verifies a user’s interaction with the website.  The objective of UI testing is to ensure that the User Interface provides the user with the appropriate access and navigation through the functions of the target-of-test.  In addition, UI testing ensures that the objects within the UI function as expected and conform to corporate or industry standards.

|  |  |
| --- | --- |
| Test Objective | Verify the following:   * Navigation through the target-of-test properly reflects business       functions and requirements, including window-to-window,, and use of access methods (mouse movements, accelerator keys) * Website window objects and characteristics, such as menus, size, position, state, and focus conform to standards. |
| Technique | Create or modify tests for each screen to verify proper navigation and object states for each application window and objects. |
| Completion Criteria | Each window successfully verified to remain consistent with benchmark version or within acceptable standard |
| Special Considerations | Not all properties for custom and third party objects can be accessed. |

### 

### 5.3.1.3. Performance testing

Performance profiling is a performance test in which response times, transaction rates, and other time-sensitive requirements are measured and evaluated.  The objective of Performance Test is to verify performance requirements have been achieved.

|  |  |
| --- | --- |
| Test Objective | Verify performance behaviors for designated transactions or business functions under the following conditions:   * normal anticipated workload * anticipated worst case workload |
| Technique | * Use Test Procedures developed for Function or Business Cycle Testing. * Modify data files to increase the number of transactions or the scripts to increase the number of iterations each transaction occurs. * Scripts should be run on one machine (best case to benchmark single user, single transaction) and be repeated with multiple clients (virtual or actual, see Special Considerations below). |
| Completion Criteria | * Single Transaction or single user:  Successful completion of the test scripts without any failures and within the expected or required time allocation per transaction. * Multiple transactions or multiple users:  Successful completion of the test scripts without any failures and within acceptable time allocation. |
| Special Considerations | Comprehensive performance testing includes having a background workload on the server.  There are several methods that can be used to perform this, including:   * “Drive transactions” directly to the server, usually in the form of Structured Query Language (SQL) calls. * Create “virtual” user load to simulate many clients, usually several hundred.  Remote Terminal Emulation tools are used to accomplish this load. This technique can also be used to load the network with “traffic”. * Use multiple physical clients, each running test scripts to place a load on the system.   Performance testing should be performed on a dedicated machine or at a dedicated time.  This permits full control and accurate measurement.  The databases used for Performance Testing should be either actual size or scaled equally. |

### 5.3.1.4 Security and Access Control Testing

Security and Access Control Testing focus on two key areas of security:

* Application-level security, including access to the Data or Business Functions
* System-level Security, including logging into or remote access from the system.

Application-level security ensures that, based upon the desired security, actors are restricted to specific functions or use cases, or are limited in the data that is available to them.  For example, everyone may be permitted to enter data and create new accounts, but only managers can delete them. If there is security at the data level, testing ensures that” user type one” can see all customer information, including financial data, however,” user two” only sees the demographic data for the same client.

System-level security ensures that only those users granted access to the system are capable of accessing the applications and only through the appropriate gateways.

|  |  |
| --- | --- |
| Test Objective | * Application-level Security:  Verify that an actor can access only those functions or data for which their user type is provided permissions. * System-level Security:  Verify that only those actors with access to the system and applications are permitted to access them. |
| Technique | * Function-level Security: Identify and list each user type and the functions or data each type has permissions for. * Create tests for each user type and verify all permission by creating transactions specific to each user type. * Modify user type and re-run tests for same users.  In each case, verify those additional functions or data are correctly available or denied. * System-level Access: See Special Considerations below |
| Completion Criteria | For each known actor type the appropriate function or data are available, and all transactions function as expected and run in prior Application Function tests. |
| Special Considerations | Access to the system must be reviewed or discussed with the appropriate network or systems administrator.  This testing may not be required as it may be a function of network or systems administration. |

## 5.3.2. Test environment

Test server configuration:

1. **Hardware:**

* CPU: core i7 2630
* RAM: 4GB
* HDD: 320GB
* OS :window 7 professional 64-bit

1. **Software:**

* Macromedia Dreamweaver 8
* Wamp Server 2.2
* Notepad++

1. **Test client configuration:**

* Windows 7
* Chrome 21; Mozilla Firefox 14.0

## 5.3.3. Risk list

* Web-based test experience
* Compatibility test
* Performance test

# 5.4. Deliverables

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Deliverables | Delivered Date | Delivered by | Delivered to |
| 1 | Test Plan | 18-06-2012 | Hungnt | Unicorn team |
| 2 | System Test case | 20-07-2012 | Hungnt | Unicorn team |
| 3 | Defect | Update daily | Huynt  Hungnt | Unicorn team |
| 4 | Test report | After finish testing | Hungnt | Unicorn team |

# 5.5. Test cases

## 5.5.1. Home site

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Test Case Description | | Test Case Procedure | Expected Output | Result |
|  | **1. Theory Library** | | |  |  |
| [Home-1] | View subject name | | Click “Thư viện lý thuyết” link in the left menu or tab “Thư viện lý thuyết” in the menu bar | Show list of subject in exist left menu. And show Subject and all chapter names of each subject. |  |
| [Home -2] | View chapter name | | 1. Click “Thư viện lý thuyết” link in the left menu or tab “Thư viện lý thuyết” in the menu bar 2. If click “Thư viện lý thuyết” link in the left menu then click on a subject in the left menu or link of subject in content area. If click or tab “Thư viện lý thuyết” in the menu bar user can only click link of subject in content area. | Show lists all chapters below subject name in the content area. If click on a subject, show list all chapter in that subject. |  |
| [Home -3] | View theory name | | 1. Click “Thư viện lý thuyết” link in the left menu or tab “Thư viện lý thuyết” in the menu bar 2. If click “Thư viện lý thuyết” link in the left menu then click on a subject in the left menu or link of subject in content area. If click or tab “Thư viện lý thuyết” in the menu bar user can only click link of subject in content area. 3. Click on the chapter name. | Show lists of all theory names below chapter name in the content area. |  |
| [Home -4] | View theory content | | 1. Click “Thư viện lý thuyết” link in the left menu or tab “Thư viện lý thuyết” in the menu bar 2. If click “Thư viện lý thuyết” link in the left menu then click on a subject in the left menu or link of subject in content area. If click or tab “Thư viện lý thuyết” in the menu bar user can only click link of subject in content area. 3. Click on the chapter name.  4. Click on the theory name. | Show content of theory in the content area and related exercises below content theory. |  |
| [Home -5] | View video of theory | | 1. Click “Thư viện lý thuyết” link in the left menu or tab “Thư viện lý thuyết” in the menu bar 2. If click “Thư viện lý thuyết” link in the left menu then click on a subject in the left menu or link of subject in content area. If click or tab “Thư viện lý thuyết” in the menu bar user can only click link of subject in content area. 3. Click on the chapter name. 4. Click on the theory name. | Show related video of theory content in the theory content. |  |
| [Home -6] | View questions of theory | | 1. Click “Thư viện lý thuyết” link in the left menu or tab “Thư viện lý thuyết” in the menu bar 2. If click “Thư viện lý thuyết” link in the left menu then click on a subject in the left menu or link of subject in content area. If click or tab “Thư viện lý thuyết” in the menu bar user can only click link of subject in content area. 3. Click on the chapter name. 4. Click on the theory name. | Show related questions below theory content. |  |
|  | **2. Exam** | | |  |  |
| [Home -7] | Login on the home site | Log in by user who having access right in the tab exam. | | 1. Log in successfully 2. Show tab exam in the menu bar. |  |
| [Home -8] | Authenticate account user before do exam. | 1. Log in by user who having access right in the tab exam. 2. Click tab exam in the menu bar. 3. Show site “Xác thực tài khoản” request user authenticate account before do exam. | | Show site “Xác thực tài khoản” and authenticate account successful. |  |
| [Home -9] | Do exam in site “Làm bài thi” | 1. Log in by user who having access right in the tab exam. 2. Click tab exam in the menu bar. 3. Show site “Xác thực tài khoản ” request user authenticate account before do exam. 4. Click “Làm bài”. 5. Click link name of exam. | | 1. Log in successful in the home site. 2. Authenticate account user before do exam in the site “Xác thực tài khoản ”. 3. After click link “Làm bài” 4. Do exam after click link name of exam. |  |
| [Home-10] | Calculate result of exam | 1. Log in by user who having access right in the tab exam. 2. Click tab exam in the menu bar. 3. Show site “Xác thực tài khoản ” request user authenticate account before do exam. 4. Click “Làm bài”. 5. Click link name of exam. 6. Answer all question in exam | | 1. Log in successful in the home site. 2. Authenticate account user before do exam in the site “Xác thực tài khoản ”. 3. After click link “Làm bài” 4. Do exam after click link name of exam. 5. Show result of exam. |  |
| [Home -11] | Do exam by account Admin | 1. Log in by admin who having access right in the tab exam. 2. Click tab exam in the menu bar. 3. Show site “Xác thực tài khoản ” request user authenticate account before do exam. 4. Click “Làm bài”. 5. Click link name of exam. | | 1. Log in successful in the home site. 2. Authenticate account user before do exam in the site “Xác thực tài khoản ”. 3. After click link “Làm bài” 4. Do exam after click link name of exam. |  |
| [Home -12] | Add new question for a new exam | 1. Log in by admin who having access right in the tab exam. 2. Click tab exam in the menu bar. 3. Show site “Xác thực tài khoản ” request user authenticate account before do exam. 4. Click “Ngân hang câu hỏi”. 5. Click image C:\Users\hungnt00838\Desktop\question-bank.gif . 6. Select “Loại cau hỏi” by combobox. 7. Select “Đề tài cầu hỏi” by combobox. 8. Select “Số những câu trả lời” by combo box. 9. Enter content of question in text area. 10. Enter answer content in text area. 11. Tick in check box “Chọn đáp án đúng” and enter percent right if it is correct answer. | | 1. Log in successful in the home site. 2. Authenticate account admin in the site “Xác thực tài khoản ”. 3. After click link “Ngân hang câu hỏi” and click link image in that site, go to site “Tạo câu hỏi”.  4. Make new question success. |  |
| [Home -12] | Add new subject | 1. Log in by admin who having access right in the tab exam. 2. Click tab exam in the menu bar. 3. Show site “Xác thực tài khoản ” request user authenticate account before do exam. 4. Click “Ngân hang câu hỏi”. 5. Click link “Môn học”. 6. Click image C:\Users\hungnt00838\Desktop\question-bank.gif . 7. Select “Loại cau hỏi” by combo box. 8. Select “Đề tài cầu hỏi” by combo box. 9. Select “Số những câu trả lời” by combo box. 10. Enter content of question in text area. 11. Enter answer content in text area. 12. Tick in check box “Chọn đáp án đúng” and enter percent right if it is correct answer. | | 1. Log in successful in the home site. 2. Authenticate account admin in the site “Xác thực tài khoản ”. 3. After click link “Ngân hang câu hỏi” and click link image in that site, go to site “Môn học”.  4. Make new subject success. |  |
| [Home -13] | Manage account in exam. | 1. Log in by admin who having access right in the tab exam. 2. Click tab exam in the menu bar. 3. Show site “Xác thực tài khoản ” request user authenticate account before do exam. 4. Click link “Administration (Quản lý)”.  5. Click link “Quản lý tài khoản” | | Show all account in the table |  |
| [Home -14] | Make new exam | 1. Log in by admin who having access right in the tab exam. 2. Click tab exam in the menu bar. 3. Show site “Xác thực tài khoản ” request user authenticate account before do exam. 4. Click link “Quản lý đề thi”. 5. Click image C:\Users\hungnt00838\Desktop\question-bank.gif . 6. Enter information, click “Cập nhật” | | Go to site “Quản lý đề thi”, enter information then click “Cập nhật” and add question in question bank. |  |
| [Home -15] | Manage Report | 1. Log in by admin who having access right in the tab exam. 2. Click tab exam in the menu bar. 3. Show site “Xác thực tài khoản ” request user authenticate account before do exam. 4. Click link “Quản lý báo cáo”. | | Enter information to search report show in table. |  |
|  | **3. Exercise library** | | | | |
| [Home -16] | Select exercise in chapter and theory | 1. Click tab “Thư viện bài tập”. 2. Select Subject 3. Select chapter. 4. Select theory. 5. Select difficulty (or not)  6. Number questions. | | Select question selections and can do it. |  |
| [Home -17] | Show result when do exercise. | 1. Click tab “Thư viện bài tập”. 2. Select Subject 3. Select chapter. 4. Select theory. 5. Select difficulty ( optional ) 6. Number questions. 7. Click answer | | Show answer of questions below question. |  |
| [Cent-18] | Login home site | 1. Login by account user or admin in the left menu | | Login successful. |  |
| [Cent-19] | Test show site “Đề thi” | 1. Login by account user or admin in the left menu. 2. Click tab “Đề thi” | | Show tab “Đề thi” after login successful. If not login, don’t show tab “Đề thi” |  |

## 5.5.2. Forum site

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Test Case Description | | Test Case Procedure | Expected Output | Result |
|  | | 1. **Log in, Log out, post new thread** | | | |
| [Foru-1] | | Login forum | 1. Log in by user who having access right in the home site, or login in forum site. 2. Click on “Diễn đàn”. | In forum has been login by account in home site. |  |
| [Foru-2] | | Logout forum | 1. Log in by user who having access right in the home site, or login in forum site. 2. Click on “Thoát”. | In forum and home site have been logout account |  |
| [Foru-3] | | Post new thread in forum | 1. Log in by user or admin in site forum 2. Click on one topic in “Chuyên mục”. 3. Click on “Tạo chủ đề mới” | Make new thread in that topic |  |
| [Foru-4] | | Change profile account user | 1. Log in by user or admin in site forum 2. Click on one topic in “Thiết lập cá nhân”. | Change profile account successful |  |

# 5.6. Test Summary Report

## 5.6.1. Text execution summary

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Execution Summary** | | | | | | |
|  | **Pass** | **Fail** | **Untested** | **Accepted** | **N/A** | **Number of Test cases** |
|  | *0* | *0* | *0* | *0* | *0* | *0* |
| **Function** | *0* | *0* | *0* | *0* | *0* | *0* |
| **Workflow** | *0* | *0* | *0* | *0* | *0* | *0* |
| **Permission** | *0* | *0* | *0* | *0* | *0* | *0* |
| **GUI** | *0* | *0* | *0* | *0* | *0* | *0* |
| **External Interface** | *0* | *0* | *0* | *0* | *0* | *0* |
| **Performance** | *0* | *0* | *0* | *0* | *0* | *0* |
| **Data migration** | *0* | *0* | *0* | *0* | *0* | *0* |
| **Test coverage** |  |  |  |  |  |  |
| Test coverage |  |  |  |  |  | ***100%*** |
| Test successful coverage |  |  |  |  |  | ***100%*** |

## 5.6.2. Text results

### 5.6.2.1. Summary of Incident Reports Raised During Project

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Severity | Total Defect | Open Defect | | | |
| **Assigned** | **Corrected** | **Accepted** | **Total Open defects** |
| Fatal | **0** | 0 | 0 | 0 | **0** |
| Serious | **0** | 0 | 0 | 0 | **0** |
| Medium | **0** | 0 | 0 | 8 | **8** |
| Cosmetic | **0** | 0 | 0 | 4 | **4** |
| **Total (Status)** | **0** | **0** | **0** | **12** | **12** |
| **Total (W.def)** | **0** | **0** | **0** | **28** | **28** |

Total defects above exclude “Cancelled” defects.

### 5.6.2.2. Action Plan for Outstanding Incidents

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | Defect ID | Defect Title | Severity | Next action | Confirmed By | Notes |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

## 5.6.3. Conclusion

### 5.6.3.1. System Test Success Criteria

|  |  |  |
| --- | --- | --- |
| # | Criteria | Status |
| 1 | All the inputs of system testing are baselined and approved by relevant stakeholders (Common requirement, ADD, SRS, Design,Software package, System test cases). |  |
| 2 | System testing is over and all known issues (if any) are aware by relevant stakeholders (Team lead, Test lead, PM, Tech lead) and documented in Test summary report. |  |
| 3 | All planned Test Cases have been executed (100% test execution coverage) |  |
| 4 | At least 95% executed system test cases have to be passed. (More than 95% test successful coverage) |  |
| 5 | 100% high priority test cases in scope are passed. |  |
| 6 | Only Cosmetic defects can be accepted |  |

### 5.6.3.2. Conclusion

V> Report No.6: Software User’s Manual

# 6.1. Installation guide

## 6.1.1. System requirements

## 6.1.2. Installing “E-Learning-Website”

### 6.1.2.1. Installing Wamp Server

### 6.1.2.2. Uploading “E-Learning-Website” files to web host

### 6.1.2.3. Creating database and a user for “E-Learning-Website”

### 6.1.2.4. Browsing to “E-Learning-Website” administrator homepage

### 6.1.2.5. Browsing to “E-Learning-Website” homepage

# 6.2. User’s guide

# 6.3. Other

# **Reference**

<http://hocmai.vn/>

<http://www.moon.vn/Home1/>

<http://truongtructuyen.vn/TrangChu.aspx>

<http://thaytro.vn>

<http://onthionline.net>

<http://kites.vn/thread/-kbs-2008-you-are-my-destiny-im-yoona-park-jaejung-vietsub-ep-103-105-sd-hd-completed--775-20-1.html>