Software Design Specification

I. Table of Contents

Table of Contents

1.0 Introduction

This section describes the design for the **E-Learning Management System(ELMS)**.

1.1 Goals and Objectives

The main purpose of ELMS is to create a online study platform.By managing both students and teachers, ELMS can easily deliver every user's work. Here are ELMS's targets:

- To record each user's information in a database.
- To minimize user's actions and keep it simple to use.
- To deliver each user's work for learning, training, teaching or managing(decided by role).
- To provide a charging system for administer to get profit.

1.2 System Statement of Scope

1.2.1 General Requirements

The following general requirements were laid out for our project named ELMS:

- A way in which student(S), teacher(T), admin(A) could login with unique ID and password.
- A way in which A could deliver both S and T's work.
- A way in which A could collect money from S&T and commit the charge.
- A way in which S and T's information could be check and change to correct mistakes.
- A way in which S and T's work could be checked by their own.
- All details from login to system working progress should be stored into database and could be called easily with security.

- **Interface**
  System request lots of interfaces to make sure the usability and coupling of whole system.

- **Database**
  An available database model should be designed to make sure every single function with related table(s) will run perfect.

- **Training**
  No one needs a complex training to capture all functions in ELMS. By simple and interactive UI, all users should get it started easily.

If there is anything you don't understand in ELMS, please find us.

1.3 System Context

Eventually, multiple users will be using the product simultaneously. Therefore, concurrent connection will be an issue for implementation. In addition, this is a pilot product that hopefully, if successful, can be used in other locations as well. This leads to issues about future support for a larger user base.

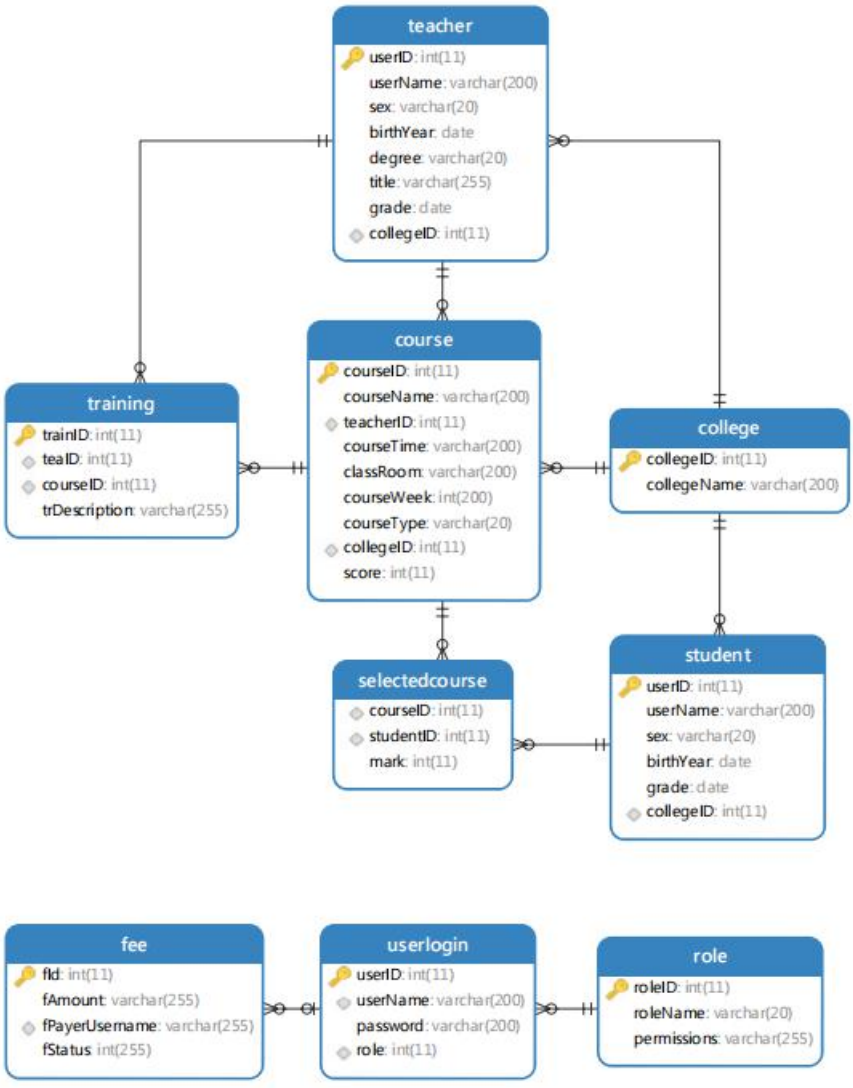
1.4 Major Constraints

Time

We only have about three weeks to finish all documentation, application creation and enhancements. We have a lot of ideas but cannot implement them due to time constraint. One of the major ones is schedule for every user which can be create, change, remove and save in our database.

2.0 Data Design

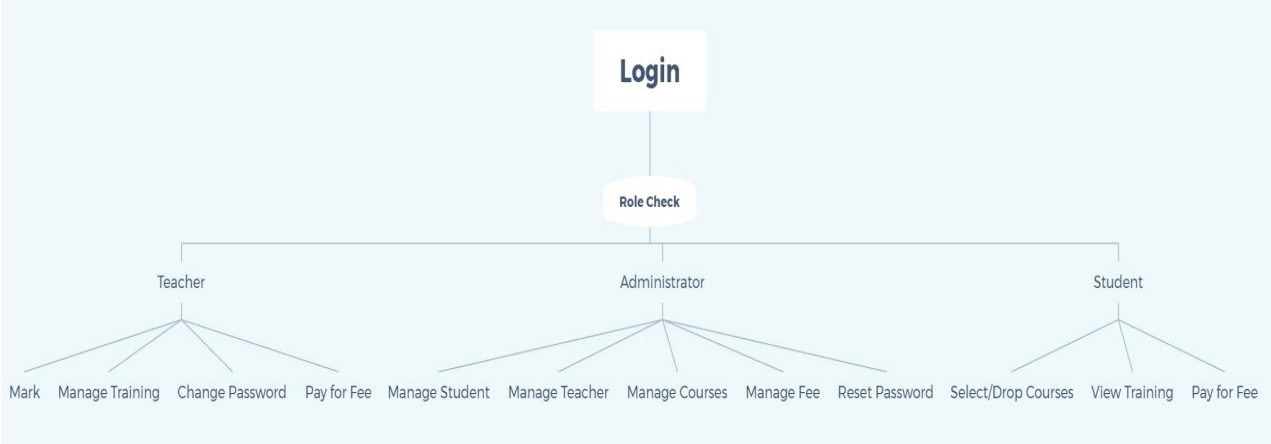
2.1 Database Description



3.0 Architectural and Component-Level Design

3.1 Program Structure

3.1.1 Overall



Menu Items

The following shows the architecture of the main menu:

- **Course**
  - Create/Modify/Delete
  - Search
  - Set other element

- **Student**
  - Create/Modify/Delete
  - Search
  - Set other element

- **Teacher**
  - Create/Modify/Delete
  - Search
  - Set other element

- **Training**
  - Create/Modify/Delete
  - Search
  - Set other element

- **Fee**
  - Create/Modify/Delete
  - Admit

3.2 Description for Components

3.2.1User Login

At the first page of log in, user should input its own id and password to log in the ELMS. At the same time our Userlogin table will record its log in action and give support for other action.

3.2.2 Admin

About student

User can click “student” button in the menu of navigation area to check/create/modify/delete every student information. Student information will infect most part of ELMS.

About teacher

User can click teacher button on the navigation bar and go to view teacher list. Then teacher information can be added/modified/deleted.

About fee

User can click “fee” button in the menu of navigation area to check/create/modify/delete fee information. Fee system is the biggest part of ELMS to get profit. Student and teacher is responsible to pay.

3.2.3 Student

About course

Student can click “course” button to see courses need to join.

Student also needs to enter the Facility ID. The browse and search functions are identical to the ones in the Facility module.

Student can select courses and also drop courses.

About training

Student can view the list of training of his/her selected courses and view the details of each course.

Students don’t have the right to edit each training.

3.2.4 Teacher

View Courses

To get the courses, the front will send the id of the teacher and access the database and search the courses of the teacher.

Giving marks

The teacher will give the marks according to the specific student. The front will send the student ID and his mark to the back which will be stored in the database.

Edit Training

The teacher has right to add/edit/delete a training which will be delivered to the students who have selected this course

3.3 Technology Applied

3.3.1 Spring MVC

SpringMVC is a Web MVC framework. It is with the Struts framework is similar, is one of the current mainstream Web MVC framework.

A key design principle in Spring Web MVC and in Spring in general is the "Open for extension, closed for modification" principle.

The Spring Web model-view-controller (MVC) framework is designed around a DispatcherServlet that dispatches requests to handlers, with configurable handler mappings, view resolution, locale and theme resolution as well as support for uploading files. The default handler is based on the @Controller and @RequestMapping annotations, offering a wide range of flexible handling methods.

In our application, the Spring MVC is the fundamental framework of the whole design in order to handle the front end, the back end and the database.

3.3.2 MyBatis

MyBatis is a first class persistence framework with support for custom SQL, stored procedures and advanced mappings. MyBatis eliminates almost all of the JDBC code and manual setting of parameters and retrieval of results. MyBatis can use simple XML or Annotations for configuration and map primitives, Map interfaces and Java POJOs (Plain Old Java Objects) to database records.

In our application, we used it to operate the database and generated mapper files using mybatis.

3.3.3 JSP

JSP stands for JavaServer Pages. JSP is one of the most powerful, easy-to-use and fundamental technology for building Java web applications. JSP combines HTML, XML, Java Servlet and JavaBeans technologies into one highly productive technology that allows web developers to develop reliable, high performance and platform independent web applications and dynamic websites.

In our application, the front page applied the JSP which made the front end stable and secure.

3.3.4 Security: Shiro

Shiro is a powerful and flexible open-source security framework that cleanly handles authentication, authorization, enterprise session management and cryptography.

Apache Shiro’s first and foremost goal is to be easy to use and understand. Security can be very complex at times, even painful, but it doesn’t have to be. A framework should mask complexities where possible and expose a clean and intuitive API that simplifies the developer’s effort to make their application(s) secure.

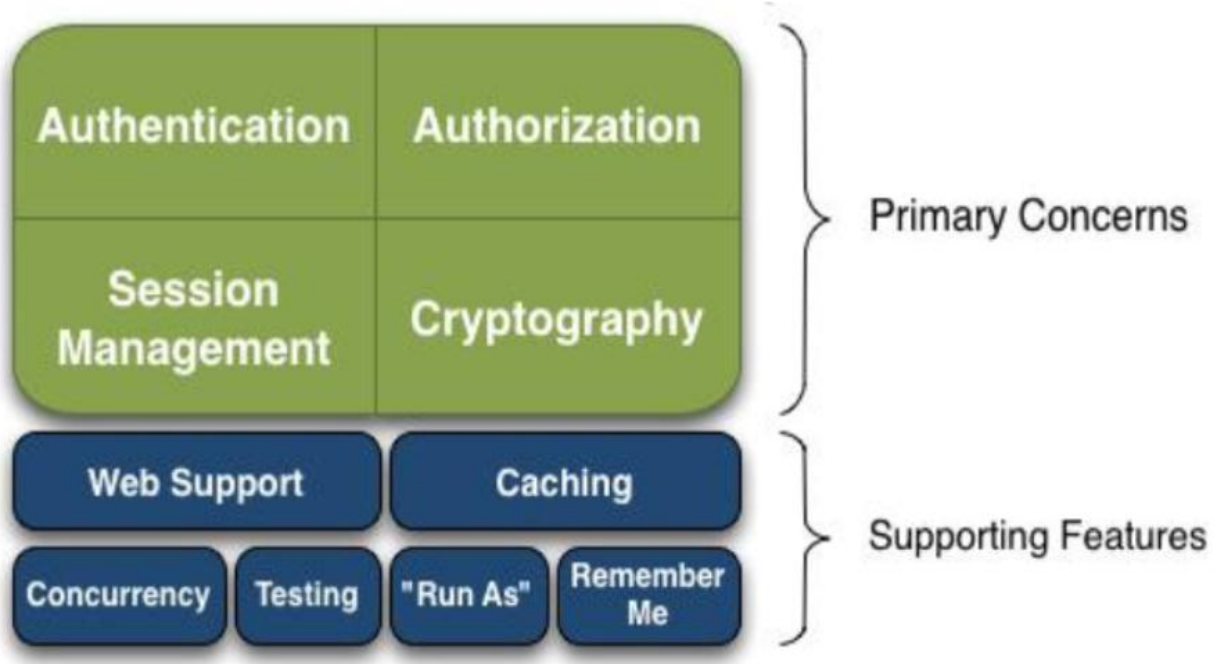
Shiro targets what the Shiro development team calls “the four cornerstones of application security” - Authentication, Authorization, Session Management, and Cryptography:

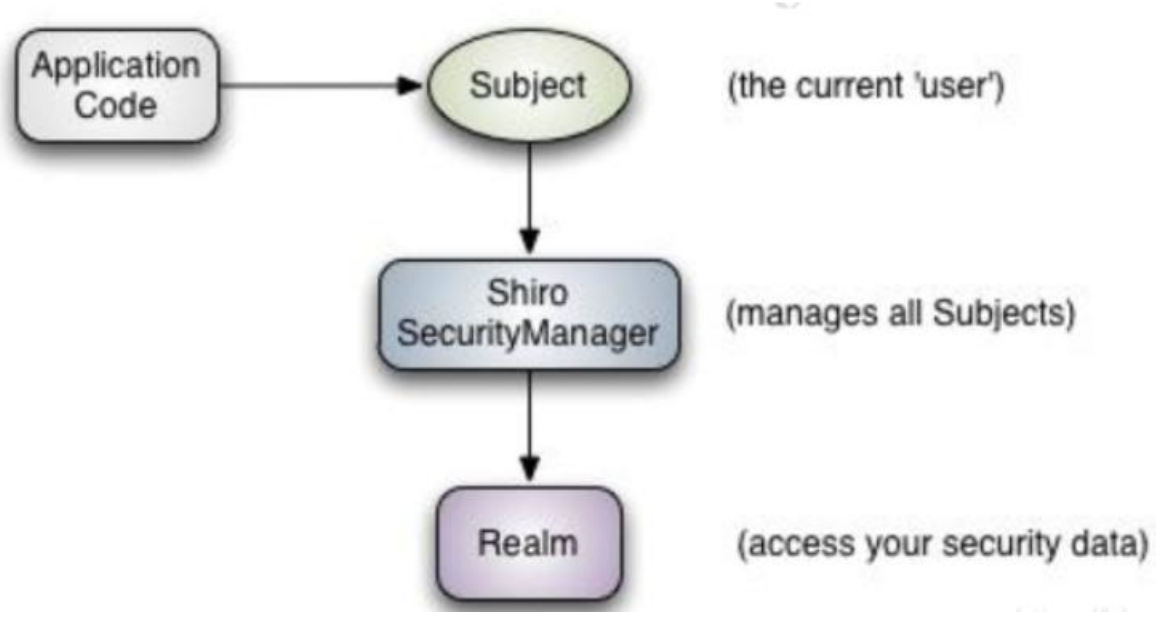
Authentication: Sometimes referred to as ‘login’, this is the act of proving a user is who they say they are.

Authorization: The process of access control, i.e. determining ‘who’ has access to ‘what’.

Session Management: Managing user-specific sessions, even in non-web or EJB applications.

Cryptography: Keeping data secure using cryptographic algorithms while still being easy to use.





Principles of Shiro

In order to apply some simple security claim, the light frame Shiro is applied to fulfill our demands. And according to the four cornerstones of Shiro, **session** is also applied in the application which used the session manager of Shiro.

4.0 User Interface Design

There will be about 30 interfaces in the program. We can't design on the exact number of it yet. Because the clients still have to think over on several interfaces, to see rather they can be combined some of the forms or put some them in separated forms.

4.1 Description of the User Interface

Below are some introductions of interfaces of the system (which are stored in the directory).

4.1.1 Objects and actions

1. Login Screen

User Name
User name is the ID of each account. The username is the ID of each account.

Password
Password is stored in the database and each login will access the database and check if the password is correct

Login
If the users enter the right user name with the matching password, it will immediately take them to the main interface.

2. Main Interface

- **Switch User**

Logout: to switch the user, one user should firstly logout or quit the system. And the system will be redirect to the login page to login.

- **Browse**

Each role has different rights to view different information. A table of the specific information is formed.

- **Search**

In some pages of the tables of information, the information can be searched using a search box. The front will send the request to the back end and search the record in the database.

- **Add**

Administrator has the right to add courses, fee, students and teachers.
Teacher has the right to add new training for certain courses.
The post request is generated and the information will be sent to the back end to insert new records. New accounts will then be stored in the userlogin table and the original password is set to a certain simple password.

- **Delete**

The Teacher has the authority to delete the training.
Administrator has the right to delete courses, students and teachers.

- **Modify**

Administrator and teacher also have the right to edit/modify the information they have created/added.
And teachers can mark students' scores which is also a modification way.
Students can select courses or drop courses which is also modification.

4.2 Interface Design Rules

Interface design focuses on three areas of concern:
1. The design of interfaces between software modules;
2. The design of interfaces between the software and other nonhuman producers and consumers of information (i.e., other external entities);
3. The design of the interface between a human (i.e., the user) and the computer.

Easy to Learn
Readability
Easy navigate between interfaces

4.3 Components Available

Since we are using Visual Basic as our front-end development language, there are a lot of ready-made components that are available for us to use already. The following is a list of controls that we will be using for this software.

4.3.1 Intrinsic Controls

Input

A Input control, sometimes called an edit field or edit control, displays information entered at design time, entered by the user, or assigned to the control in code at run time.

Label

A Label control is a graphical control you can use to display text that a user can't change directly.

Image

Use the Image control to display a graphic. An Image control can display a graphic from a bitmap, icon, or metafile, as well as enhanced metafile, JPEG, or GIF files.

Menu

A Menu control displays a custom menu for the application. Each role has different menus.

RadioButton

Radio buttons are generated to select some information. And the value can be sent to the back end.

Shape

The Shape control is a graphical control displayed as a rectangle, square, oval, circle, rounded rectangle, or rounded square.

5.0 Testing Issues

To validate the system we need to test the system. During the testing we will be concerned about the inputs and their expected outputs. We emphasize on the testing where we will input the data and will compare the output with the expected results. At this stage, we are not concerned about the process; we are only looking for the correct outputs.

6.0 Summary

In this application development phase, we have learned the brand new technologies of JAVAEE and database. The new technologies of this era made us challenged in designing and developing something new. We strongly believe that these new technologies will be the promotion of enterprise development in our future.