

CHALMERS



'New Fire' system

The student assignment submission system for Computer
Science Department of Chalmers

*Master's Thesis in the Master Degree Program,
Software Engineering and Technology*

Yi Xu

Department of Computer Science
Division of Software Engineering and Technology
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden, 2010

The Author grants to Chalmers University of Technology and University of Gothenburg the non-exclusive right to publish the Work electronically and in a non-commercial purpose make it accessible on the Internet.

The Author warrants that he/she is the author to the Work, and warrants that the Work does not contain text, pictures or other material that violates copyright law.

The Author shall, when transferring the rights of the Work to a third party (for example a publisher or a company), acknowledge the third party about this agreement. If the Author has signed a copyright agreement with a third party regarding the Work, the Author warrants hereby that he/she has obtained any necessary permission from this third party to let Chalmers University of Technology and University of Gothenburg store the Work electronically and make it accessible on the Internet.

'New Fire' system

Yi Xu

© Yi Xu, June 2010.

Examiner: Sven Arne Andreasson

Chalmers University of Technology
University of Gothenburg
Department of Computer Science and Engineering
SE-412 96 Göteborg
Sweden

[Cover:
an explanatory caption for the (possible) cover picture
with page reference to detailed information in this essay.]

Department of Computer Science and Engineering
Göteborg, Sweden June 2010

PREFACE

This is a thesis within Masters Program “Software Engineering and Technology” at Chalmers University of Technology. The supervisor of this study is Sven Arne Andreasson.

I am very grateful to my supervisor who helped me a lot in better understanding this thesis and writing the report.

Göteborg, Jun 2010

Yi Xu

ABSTRACT

Nowadays 'Fire' system is used to manage the assignments of courses in the department of Computer Science. Though 'Fire' system is quite good and satisfies the needs of requirements, it still has a few drawbacks. In the thesis, 'New Fire' system is implemented to modify these drawbacks by using JSF framework.

'New Fire' system will be used for students from Chalmers University and Gothenburg University due to the students from both universities can choose the same course. When students register in the 'New Fire' system, they choose the corresponding given course code based on the university then they can form the group and submit the assignment.

The report describes the processes of developing 'New Fire' system. First step is to analyze the system requirement and get the domain model. Then 'New Fire' takes use of Oracle as the database and uses Hibernate to get the entity classes. In the front-end of the system, JSF(Java Server Faces) framework is utilized to deal with the business logic and view pages. At the end of the report, the security issues and future works will be discussed.

Content

PREFACE	3 -
ABSTRACT	4 -
1. Introduction	7 -
2. The advantages and drawbacks of the ‘Fire’ system	8 -
3. Requirement Analysis	9 -
3.1. The four roles in the system.....	9 -
4. Analysis for the development technology	14 -
4.1. Technology alternatives consideration.....	14 -
4.1.1. Prefer Java EE to Microsoft .Net.....	14 -
4.1.2. Prefer JSF to Struts2	15 -
4.2. The introduction of JSF.....	16 -
4.2.1. Managed-bean in JSF	16 -
4.2.2. The lifecycle for JSF.....	18 -
4.3. The Hibernate technology	21 -
4.3.1. The introduction of Hibernate	21 -
4.3.2. The architecture of Hibernate	21 -
4.4. The IDE and server used in the system.....	22 -
5. The implementation of ‘New Fire’ system	24 -
5.1. Thin client architecture	24 -
5.2. System Architecture	24 -
5.2.1. The ‘View’ tier	25 -
5.2.2. The ‘Controller’ tier.....	26 -
5.2.3. The ‘Model’ tier.....	26 -
5.3. The domain model.....	26 -
5.4. The implementation of system functions	27 -
5.4.1. The course management	27 -
5.4.2. Teacher assignment for courses.....	30 -
5.4.3. Lab management.....	32 -
5.4.4. Student group management	33 -

5.4.5. Student upload and download file	- 34 -
6. Conclusion	- 38 -
7. Reference	- 39 -
8. Glossary	- 40 -
9. Appendix	- 41 -

1. Introduction

The outcome of Software Engineering and Technology program requires students to be able to systematically develop, operate, maintain and retire large software system.[1] The goal of my master thesis is to enhance my ability to develop software through an entire system.

In this master thesis, the design of a student assignment submission system for Chalmers Computer Science and Engineering department is described; the name of the new system is 'New Fire'. So far, the 'Fire' system is widely used in the department of Computer Science for the assignment submissions. Generally speaking, 'Fire' system satisfies the needs of the labs. But there are still some drawbacks that need to be modified. 'New Fire' System is designed to modify these drawbacks and make it easy to use. The purpose for 'New Fire' system is to facilitate the teachers to grade the assignments and students from Chalmers and Gothenburg University to submit the assignments. Administrator has the right to create the courses and edit the information of the courses. Course responsible creates the labs according to the course plan and grades students' assignments.

The first step of the 'New Fire' system is requirement analysis in order to decide the domain model. Through investigating many students and teachers about the usage of 'Fire' system, the design of the domain model is implemented aiming to make the new system much easier to use. The second step is to carry out the domain model in a database. The 'New Fire' system uses the Oracle to store the data. The third step is to implement the system. Nowadays there are uncountable Java web development frameworks to use, such as Struts1, Struts2, Spring etc. Considering the popularity and usability, JSF1.2+Hibernate3.2.5 framework is used in the implementation of 'New Fire' system. As for the software architecture, the popular architecture Model-View-Controller is used in 'New Fire' system.

2. The advantages and drawbacks of the ‘Fire’ system

Nowadays, ‘Fire’ system is widely used in Computer Science department. The system assists course responsible to perform the lab part for the courses very well. The students can form a group through the system and submit their assignments.

Although ‘Fire’ system has many advantages, it still has the following drawbacks:

1. There is no person at the department that understands the source code of ‘Fire’ system, so it is impossible to add or change functions.
2. The key field for students register is their emails, so that one student can register in the system more than once through different emails. ‘New Fire’ system will take use of personal number of students as the key, so one student can only register once for a course due to every student owns unique personal number.
3. The student cannot change his/her personal information. In ‘New Fire’ system the students are able to change their information through student area function.
4. After course responsible creates a course or a lab, it is impossible for him/her to change the course information or lab information any more.
5. When the students click Logout button, and the navigate will go to the login page, but at this time, if the student click ‘go back’ function in the browser, it will go to student page again. In the ‘New Fire’, when students click Logout button, system will invalidate the session.

3. Requirement Analysis

The 'New Fire' system will be used by both teachers and students in the lab part of courses. First of all, the administrator should create a course. If one university will arrange the course for a certain period then the administrator should create a given course with certain year and period. Next step is to assign teachers and build labs for the given course. After that the students can register for the given course and submit their assignment; teachers can grade the assignment and write the course report. It is critical to analyze the different requirements for different roles and assign proper rights to them.

3.1. The four roles in the system

There are four roles in 'New Fire' system: the student, the course responsible, the regular teacher and the administrator.

● **The student:** is able to register in the system according to the course, log in to the system and upload their submission. The detail authorities of students is as follows:

1. Register in the system
2. Login/logout the system through typing in their personal number and password
3. Change Register Information
4. Create Group according to the requirement of the lab in the course
5. Join the group which is already built in the system
6. Leave the group
7. Upload the submission
8. Download the submission
9. View the submission
10. Delete the submission

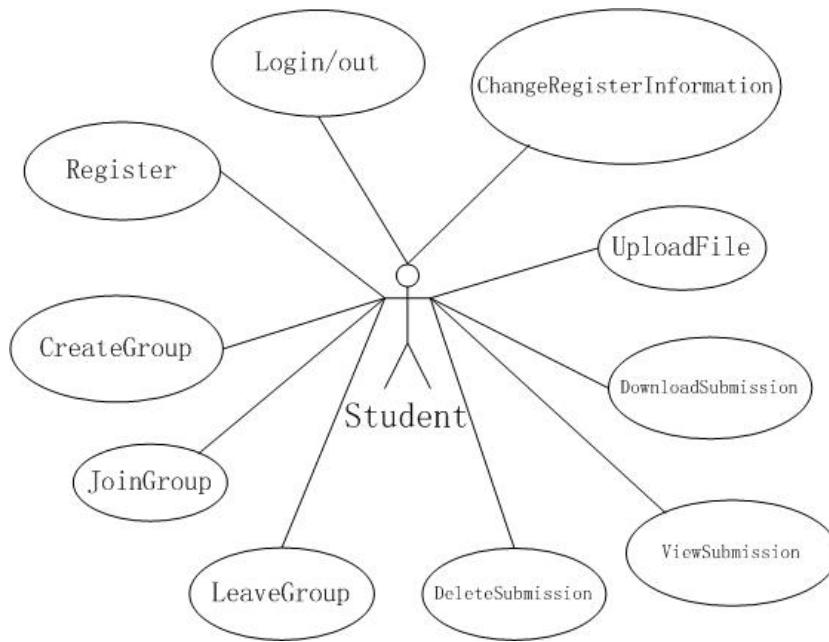


Figure 3-1: the user case: Student

● **The course responsible** is able to create the courses and given courses, edit the courses and given courses, manage the labs and grade the submissions. The detail authorities of course responsible is as follows:

1. Register in the system
2. Log in or log out the system
3. Change his/her register information
4. Create the course/given course
5. Edit the course/given course
6. Add the labs
7. Edit the labs
8. View the labs
9. Grade the labs
10. Respond to the labs
11. Download all submissions
12. Edit groups
13. Get email list
14. Write course report

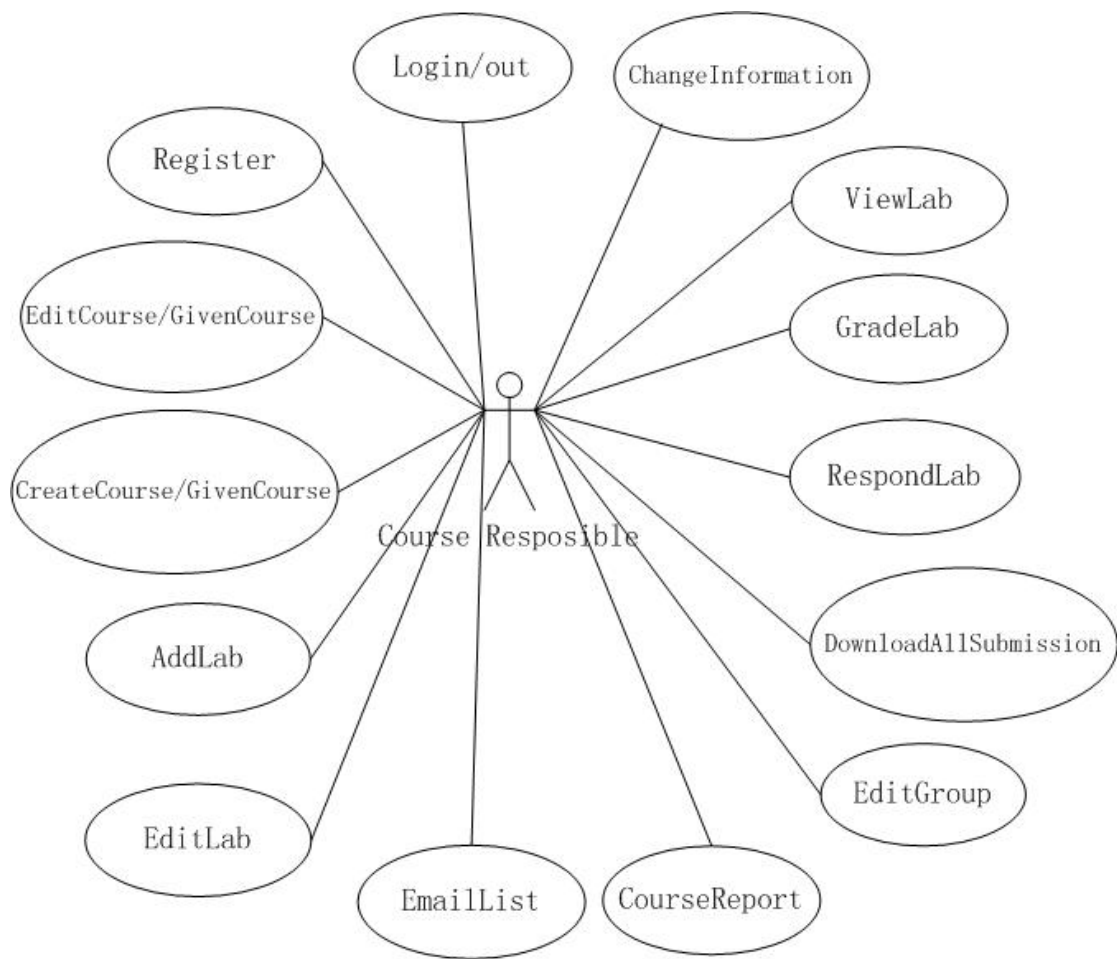


Figure 3-2: the user case: Course Responsible

- **The regular teacher** aims to view and grade the course and write the course report, so this role only contains the part of authorities is as follows:

1. Register in the system
2. Log in or log out the system
3. Change his/her register information
4. View the labs
5. Grade the labs
6. Respond to the labs
7. Download all submissions
8. Get email list
9. Write course report

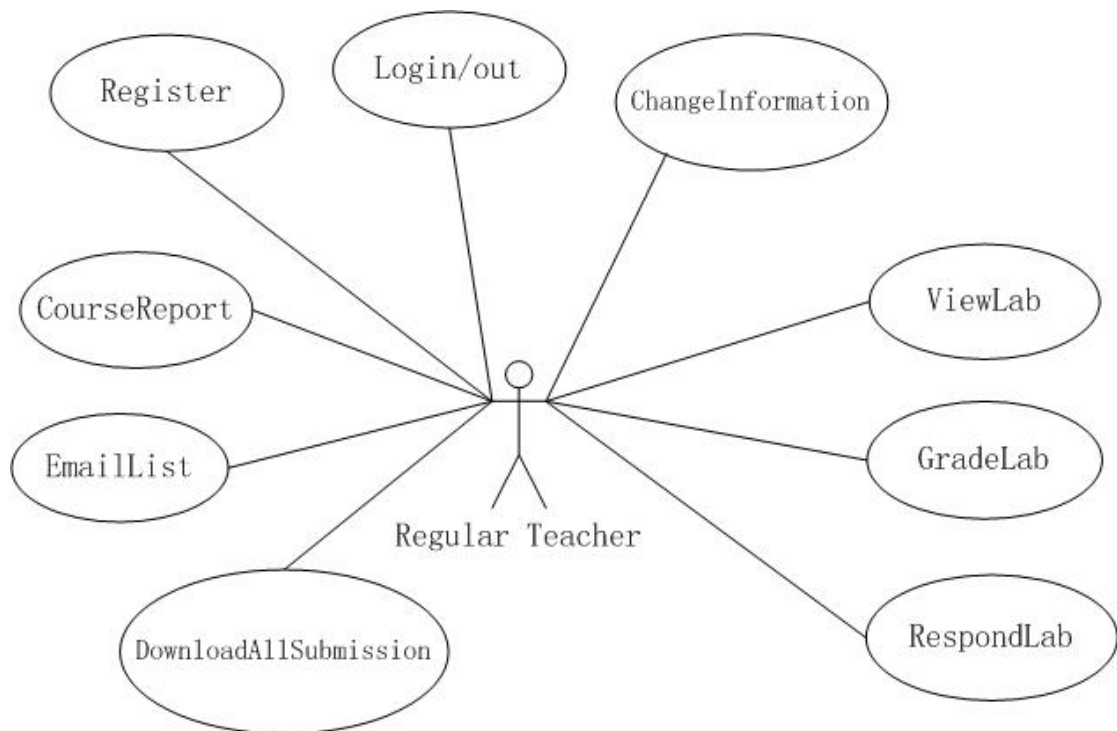


Figure 3-3: the user case: Regular Teacher

- **The system administrator** can do everything the course responsible can do, besides them, he can also add administrator, add the course responsible, regular teacher, take away students and change system password.
1. Functions are the same as course responsible
 2. Add new administrator
 3. Change system password
 4. Add new course responsible
 5. Add new regular teachers
 6. Take away students

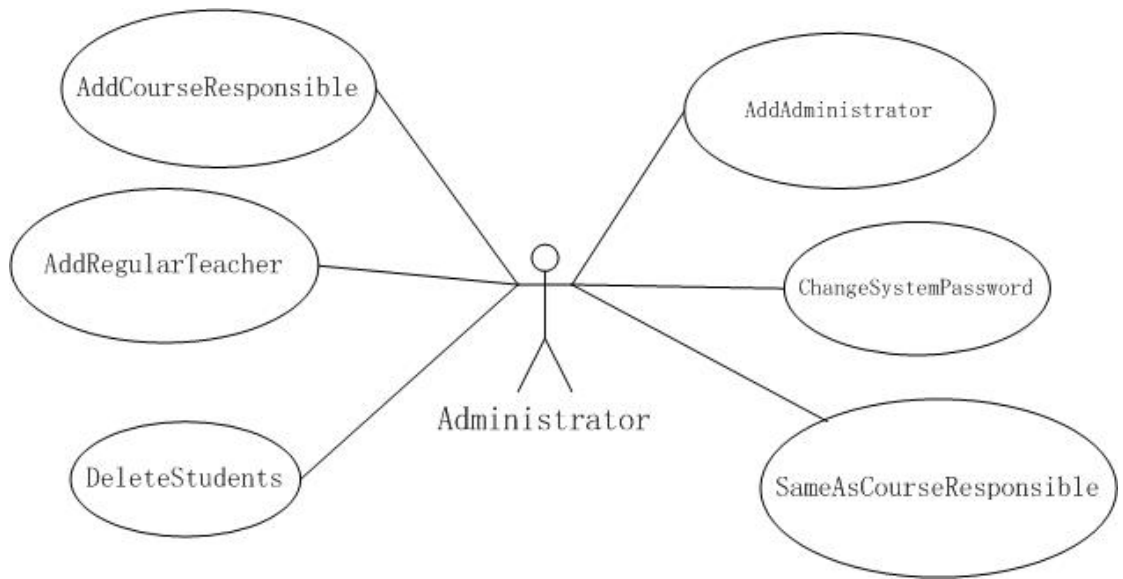


Figure 3-4: the user case: Administrator

4. Analysis for the development technology

The interaction design of web application is developing rapidly in recent years. From the static HTML to JSP to the most popular Ajax which is the basic for Web 2.0 time. The rapid development of Web technology gives us many alternatives to choose. Before implement the program, it is necessary to study different technology and choose the most suitable one.

4.1. Technology alternatives consideration

This section gives reasons for the selected technology over others.

4.1.1. Prefer Java EE to Microsoft .NET

In terms of developing a usable web services system, it should be reliable, highly available, fault-tolerant, scalable and performing at acceptable levels. These needs are no different than the needs of any other enterprise application. Java EE and .NET platforms evaluate from enterprise application and now dominate the web development market. Java EE is an industry standard led by Sun Microsystems, while .NET is a product strategy operated by Microsoft. The main support language for Java EE is Java, while the foundation languages of .NET framework are VB and C#. It is quite difficult to say which one is better. To some extent, they have a few similarities (figure 4-1).

Feature	Java EE	.NET
Type of technology	30+	Microsoft
Interpreter	JRE	CLR
Dynamic Web Pages	JSP	ASP.NET
Middle-Tier Components	EJB, Java Beans	.NET Managed Components
Database Access	JDBC,SQL/J	ADO.NET
SOAP,WSDL,UDDI	Yes	Yes
Implicit middleware (load-balancing, etc)	Yes	Yes

Figure 4-1: Analogies between Java EE and .NET

Though they have some similarities, differences do still widely exist. During the development of the 'New Fire' system, Java EE is used as the following reasons:

- **Portability**

Java EE is platform-agnostic, running on a variety of hardware and operating systems, such as Win32, Linux and Mac. Sun has built a Java EE

compatibility test suite which ensures portability of application and that Java EE platforms comply with the standards.

By the way of comparison, .NET only runs on Windows, its supported hardware, and the .NET environment. It is not portable at all. This makes Java EE to have very splendid future for its portability.

- **System Cost**

Java EE architecture has various implementations which are open source. The users can download the necessary software from Internet free of charge, and a lot of library files as well. The users have the possibility to choose the service according to their budget.

By contrast, taking use of .NET, the user must buy the product of Visual Studio at aggressive price and does not have as many alternatives as Java EE.[2]

In conclusion, while both Java EE and .NET have their own features, more programmers will reap greater wins with J2EE, because its advantages outweigh those offered by Microsoft.NET. That is the preferred architecture for the system.

4.1.2. Prefer JSF to Struts2

After the above comparison, the 'New Fire' system takes use of the open source Java EE frameworks to develop. There are a lot of Java EE frameworks, such as JSF, Struts2, Stripes, Wicket and so on. Among them JSF and Struts2 are widely used. The web programmers using JSF or Struts argue a lot about which one is much better, and the argument last until now. The comparison can be made from the aspects below:

- **Java EE standard**

Java EE 5.0 containers must provide an implementation of JSF, which means that JSF is ubiquitous. With approximately 50 Java-based web app frameworks, tool vendors have been reluctant to support a particular framework, but a standard is something tool vendors prefer to support. What is more, open source projects are rallying around JSF with such dizzying speed that they're stepping over one another to implement the same features . The above advantages of JSF result in much more available tools than Struts2.

- **The different paradigms**

JSF is a component framework and elements in the front-end pages are rendered as individual components. The JSF programs are event driven much like in Java Swing of Java SE. The command button and item list have the ActionEvent and ValueChangeEvent respectively, which listen the users' commands.

Struts2 is an action framework. In essence they give you the ability to map URLs to activities and code on the back end according to the struts.xml file. Struts configuration file defines a Java class and navigator map for every action. Struts2 isolates the binding of the request data to the action implementation classes.

- **Validation**

JSF provides the client-side validation and users could use the validation tags from JSF library for instance: <f:validateLength> to validate the length of user input; <f:validateLongRange> to validate the range of user input. These features make the developers only have to indicate which validation rules to process the validation of users' input.

Struts2 configures a file named “validator-rules.xml” to execute the validation work no matter server-side validation or client-side validation. This requires extra work from developers.[3]

In conclusion, JSF is the more up to date than Struts2 due to its rich infrastructure component feature and easy to use. So JSF framework is used to develop the ‘New Fire’ system.



Figure 4-2: Salvation lies in JSF

4.2. The introduction of JSF

Java Server Faces is the well-established standard for web-development frameworks in Java. The standard is based on the MVC paradigm, but is additionally to most web-frameworks also component-based and event-oriented.[4]

4.2.1. Managed-bean in JSF

In JSF framework, managed-bean which is refer to a Plain Old Java Object(POJO), takes a vital role in the application development such as reading and writing the data from the JSF component in front-end JSP pages; executing the Validation, ActionListener as well as ValueChangeListener;

executing the action method to navigate the system to different page. The managed-beans are defined in the configuration file faces-config.xml with assigning different scope so that managed-beans have their own lifespan.

- **The scope of Managed-bean**

- a) None:** A managed bean with this scope is not stored anywhere, it is created "on demand" whenever it is needed. [5]
- b) Request:** A managed bean with request scope is visible from the start of the request until the end of the request. If the request is forwarded the managed bean to another page, it is still visible in that page because it is still part of the same request/response cycle.
- c) Session:** A managed bean with session scope will be stored onto session that means it is visible for any request/response cycle that belongs to a session until the session is timed out or the user explicitly invalidate the session. Different users have different session objects.
- d) Application:** A managed bean with application scope will exist during the entire lifetime. And all the users share the same application object.

- **Different kinds of Managed-beans**

Because managed-beans deal with a lot of works during the JSF lifecycle so the code in the managed-bean will become cumbersome if all the works are processed in one bean. Thus it is necessary to distinguish between different kinds of managed-beans. [6]

- a) Backing Managed Bean:** This type of managed bean participates in the "View" concern of the MVC design pattern. The backing managed bean has the one to one relationship with JSF form and it typically has JavaBean-style properties with associated getters/setters. Backing managed beans also deal with the ActionListener and valueChangeListener methods. The scope of backing bean is often set to request.
- b) Model Managed Bean:** This type of managed bean takes part in "Model" concern of the MVC design pattern. It is used to fetch the entity data or query result from the database. Backing bean and controller bean often set the model bean as a property and initializes the value through dependency injection. The scope of model bean is often set to session.
- c) Controller Managed Bean:** This type of managed bean is considered as the "Controller" model in MVC design pattern. The controller bean executes some business logic and return the outcome to navigate to next page. It often deals with the action methods in JSF. The scope of controller bean is often set to request.
- d) Support Managed Bean:** This type of managed bean creates the java.util.map object to initialize <h:selectOneMenu> dropdown list in JSP pages. If the data in the dropdown list is particular to the user,

then the bean would be kept in session scope. If the data applies to all users, then the bean would be kept in application scope.

4.2.2. The lifecycle for JSF

Like the JSP page: the client makes an HTTP request for the page, and the server responds with the pages translated to HTML, because there are some extra services JSF support, the lifecycle for JSF is as follows:

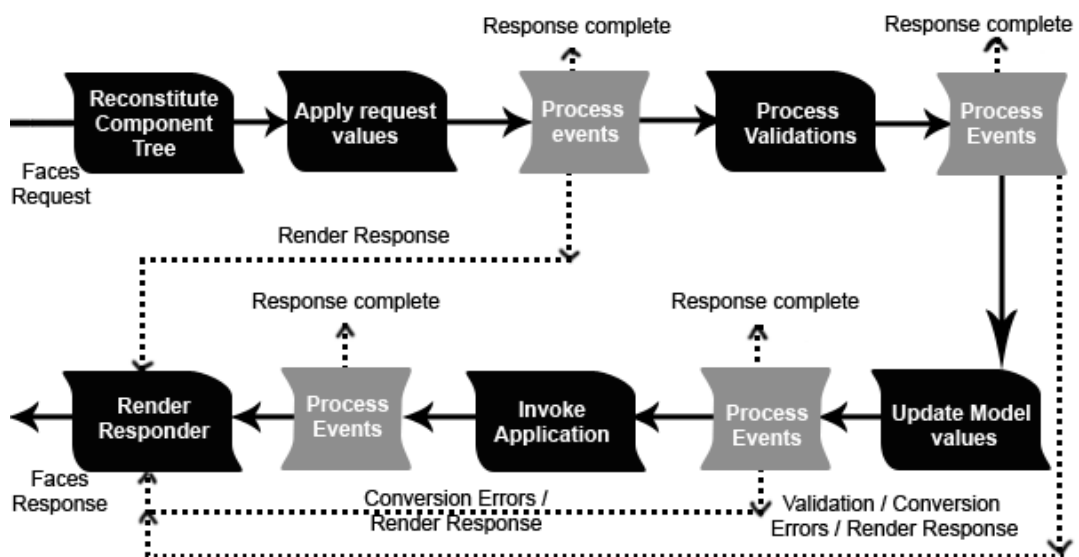


Figure 4-3 :The lifecycle of JSF[7]

There are six steps in the lifecycle:

1. Reconstitute the component tree

At the beginning of the lifecycle, the web application arranges JSF components in tree for defining the page-structure, wires event handlers and validators to components in the view, saves the view in the FacesContext instance.[7] 'New Fire' system uses the JSP as the template language. The snippet is the code from fireLogin.jsp:

```
<f:view><h:form id="loginForm">
<h:outputText value="#{msgsEn.PersonNumber}"/>
<h:inputHidden id="promptMessage" requiredMessage="hello"/>
<h:inputText id="username" size="20"
value="#{userLogin.username}"/></h:inputText>
<h:commandButton value="#{msgsEn.login}" type="submit"
action="#{loginControl.submit}"/>
</h:form></f:view>
```

<f:view/> tag is the root-component of the JSF page, the tag beginning

with the letter <f: >or<h: > is from the JSF tag library. The annotation `#{msgsEn.PersonNumber}` is Expression Language which can access the backing beans and resource files.

2. Apply request values phase

In the second step, the JSF components in the tree extracts its new value from the request parameters by using decode method.

In the JSP page fireLogin.jsp

```
<h:inputText id="username" size="20"
value="#{userLogin.username}"></h:inputText>
```

The `<h:inputText>` component will retrieve its new value from user inputs and store it locally. At the end of this phase, the components are set to their new values, and messages and events have been queued.

3. Process validations phase

During this phase local values stored for the component in the tree are compared to the validation rules registered for the components. Validation is critical to guarantee the user input is valid and meet the requirement of the system.

In the JSF framework, there are four ways to validate the certain data:

- Built-In Validation
- Application Level Validation
- Custom Validation
- Validation Methods in Backing Beans

‘New Fire’ system takes the last method “Validation Methods in Backing Beans” for the following reasons:

- It is easy to manage the validation; the programmer only needs to modify the backing bean code to change the validation messages.
- Programmer does not need to write a new class which implements Validator interface and register custom validator in the configuration file.

In the fireLogin.jsp page, when the user clicks “Login” button, the managed-bean deals with the action:

```
public String submit() {
    if (stuModel==null) {
        new SystemMessage().userLoginError();
        return "loginFail";
    }
}
```

```
}
```

In class `SystemMessage`, JSF implementation adds an error message to `FacesContext` instance:

```
public void userLoginError() {  
    FacesContext context = FacesContext.getCurrentInstance();  
    FacesMessage message = new FacesMessage();  
    message.setSeverity(FacesMessage.SEVERITY_ERROR);  
    message.setSummary("Invalid username or password");  
    message.setDetail("Invalid username or password");  
    context.addMessage("loginForm:username", message);  
}
```

And at this time the life cycle will advance directly to the render response phase so that the page is rendered again with the error messages displayed.

4. Update model values phase

JSF implementation walks the component tree and retrieves current value from JSF component to the backing beans, following is the backing bean for the `fireLogin.jsp`:

```
public class LoginBacking{  
    private String username="";  
    private String password="";  
    public String getPassword() {  
        return password;  
    }  
    public void setPassword(String password) {  
        this.password = password;  
    }  
    public String getUsername() {  
        return username;  
    }  
    public void setUsername(String username) {  
        this.username = username;  
    }  
}
```

In JSF, `faces-config.xml` is used to configure the managed-beans, navigator rules and resource files. The below code is to register a managed-beans in `faces-config.xml` file.

```
<managed-bean>  
<managed-bean-name>userLogin</managed-bean-name>  
<managed-bean-class>LoginBacking</managed-bean-class>
```

```
<managed-bean-scope>request</managed-bean-scope>
</managed-bean>
```

At the end of this phase, the fields username and password in backing bean “userLogin” will be evaluated by the values user input in fireLogin.jsp page.

5. Invoke application phase

In the fifth step, component values which are already stored in the backing beans can execute some business logic. Some application-level code is executed such as submitting a form or linking to another page.

In faces-config.xml the navigator rule is defined as following snippet:

```
<navigation-rule>
  <from-view-id>/fireLogin.jsp</from-view-id>
  <navigation-case>
    <from-outcome>loginFail</from-outcome>
    <to-view-id>/fireLogin.jsp</to-view-id>
  </navigation-case>
  <navigation-case>
    <from-outcome>loginPass</from-outcome>
    <to-view-id>/stuPages/stuArea.jsp</to-view-id>
  </navigation-case>
</navigation-rule>
```

If the submit method returns “loginFail” the system will redirect to page fireLogin.jsp again; while it returns “loginPass” the system will jump to stuArea.jsp.

6. Render response phase

In the last step of JSF lifecycle, JSP container renders the page back to the user with the current state. The state of the response is saved after rendering of the content of the view.

(for more information about lifecycle in JSF, see the appendix)

4.3. The Hibernate technology

‘New Fire’ system takes use of Oracle as the database and utilizes Hibernate to map an object-oriented domain model to the oracle database.

4.3.1. The introduction of Hibernate

Hibernate is a collection of related projects enabling developers to utilize POJO-style domain models in their application in ways extending well beyond object/relational mapping. [8]

4.3.2 The architecture of Hibernate

In the Hibernate architecture, the central part is the configuration file `hibernate.cfg.xml` which contains information about database connection, resource mappings and other connection properties. Before creating the mapping file and POJO persistence class, the programmer also needs to create the reverse engineering file: `hibernate.reveng.xml` which enables the programmer to have better control over database mapping strategy.

The Hibernate creates both mapping file and POJO persistence class for every table in the database. Data access object and service layer can use these persistence classes through the Hibernate API. [9] The architecture of Hibernate following as:

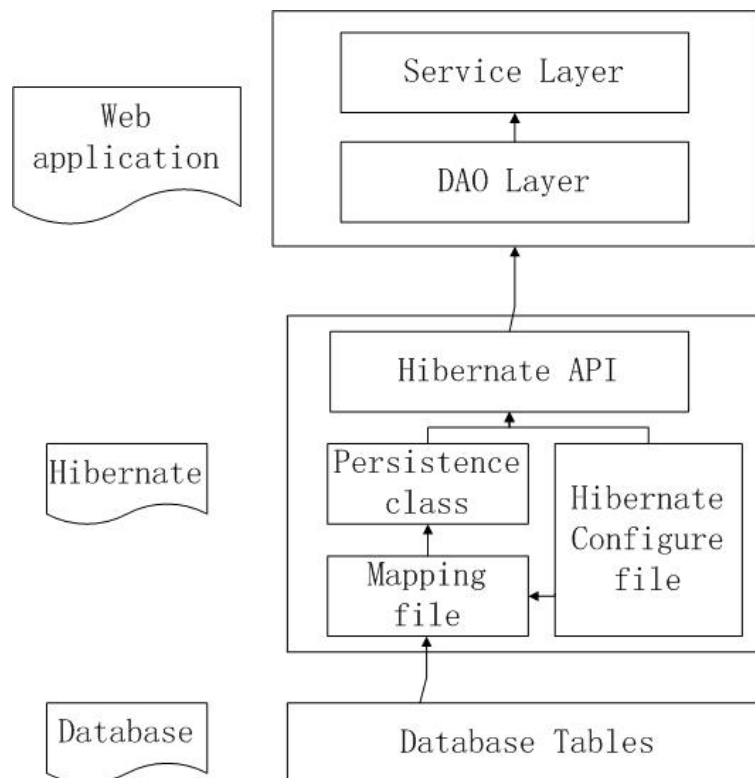


Figure 4-4: The architecture of Hibernate

4.4. The IDE and server used in the system

- **NetBeans 6.8**

The NetBeans6.8 was the latest version when 'New Fire' system began to be developed. It integrates with the web application quite well and imports the necessary libraries automatically.

- **Tomcat 6.0.18**

Tomcat 6.0.18 is the popular and stable free server which performs better than previous versions in terms of security.

- **Free SMTP Server**

Free SMTP Server is the email server to send emails to clients. 'New Fire' system uses it to test Email functions.

5. The implementation of 'New Fire' system

5.1. Thin client architecture

'New Fire' system takes use of the thin client architecture for its implementation. Since this system is a web-based application and the data processing is executed on the server side. The advantages of thin client architecture is as follows:

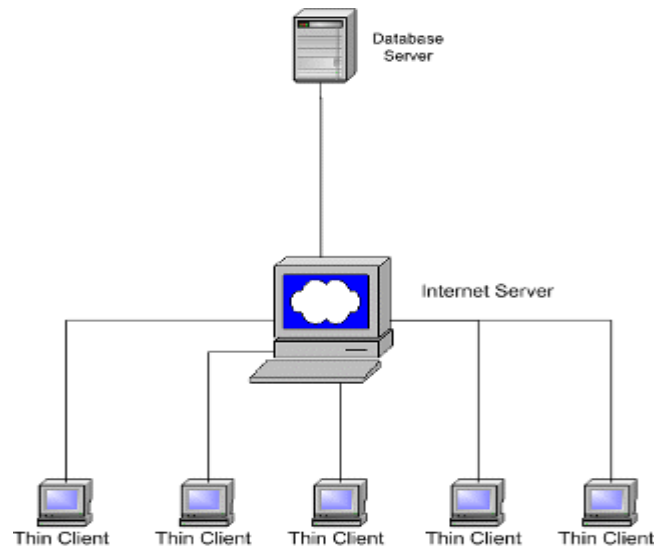


Figure 5-1 : The diagram for thin client architecture

- Security
For New Fire system, security is very important. The security of the thin client system depends on the server side. So as long as we take action to guarantee the safety of server, then the system will run fine.
- Extensibility
The thin client architecture is much easier to update compared with the thick client architecture. When updating, the thin client only needs to update the server side.

5.2. System Architecture

The popular web-based architecture MVC is used in 'New Fire' system, the reason is that MVC separates the web application into different tiers and each part executes the specific functions. MVC makes web application easy to update and keep maintenance.

In 'New Fire' system, View layer focus on the front-end and realizes backing bean for each jsp page; Model layer deals with the business logic, and use model bean to pass data among different layers; Controller layer navigate the system to different pages according to the different return value.

The system architecture is as follows:

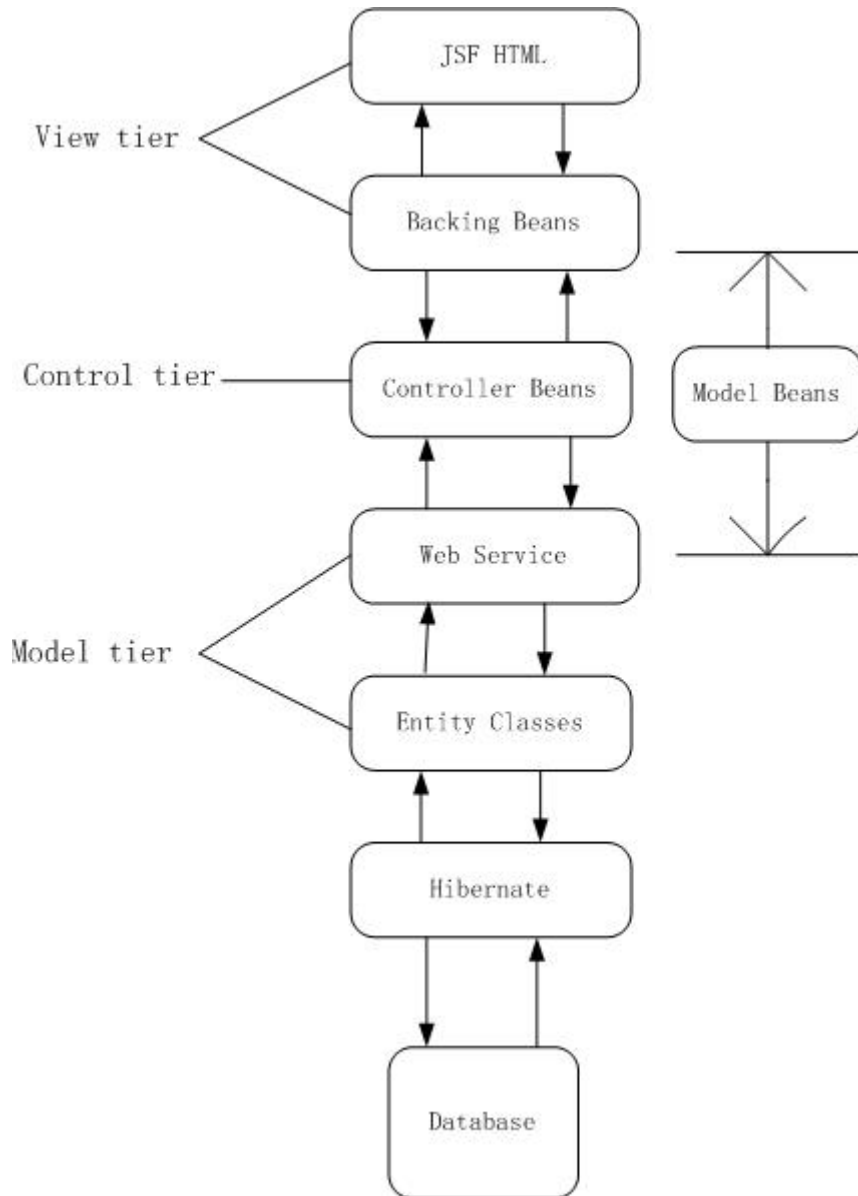


Figure 5-2: The system architecture of 'New Fire' system

5.2.1. The 'View' tier

The 'New Fire' system realizes the front-end with HTML tags and JSF tags(See the appendix). The design of the new system follows the "Fire" system design style. So the CSS file in new system borrows some rules from "Fire" system.

One feature of JSF is to use backing bean to operate the form in the front-end page. This separates the validation code and converter code from the JSP pages in order to make the program clearer.

5.2.2. The 'Controller' tier

The controller bean is responsible for 'Controller' tier. When a user submits a form, corresponding method in controller bean will be invoked and in the end the method returns a String, then system will check the configuration file to jump to next page.

5.2.3. The 'Model' tier

The 'Model' tier manages the business logic in the system. Hibernate extracts the POJO entity classes from the database and then the program operate these class in the web service part. Sometimes wrap the needed data into model bean and then use it in the backing beans or controller beans through dependency injection. Another way to fetch the wanted data is to return the object of entity classes to the backing beans or controller beans.

5.3. The domain model

The design of domain model is according to the system requirement, the goal of the design is to make 'New Fire' system easy to use and maintain, considering the extensibility in future, two reserves fields for each table are created.

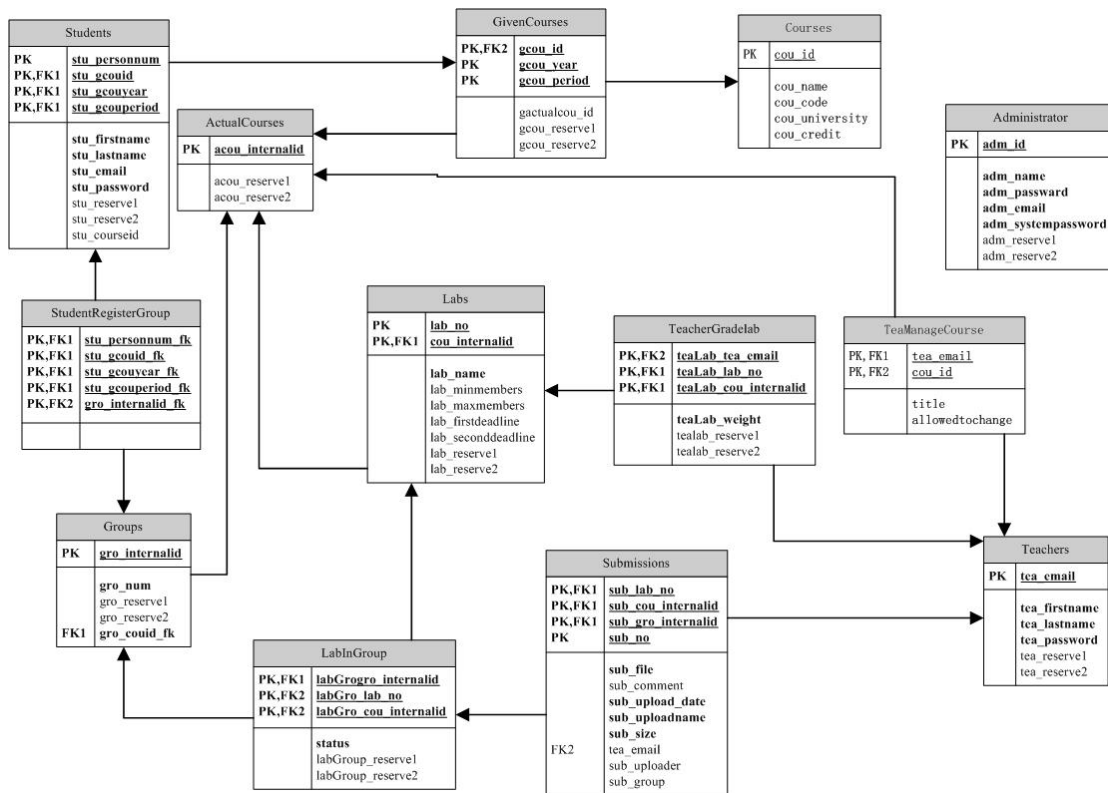


Figure 5-3: The domain model

5.4. The implementation of system functions

During the development of 'New Fire' system, the functions are implemented base on different function modules is as follows:

5.4.1. The course management

The big difference between 'New Fire' system and most course management systems is that most of course management systems are special used in one school but 'New Fire' system is used in both Chalmers and Gothenburg University at the same time. Generally, two courses from both universities have the same ActualCoursesID so that students from both universities can register as the same ActualCoursesID.

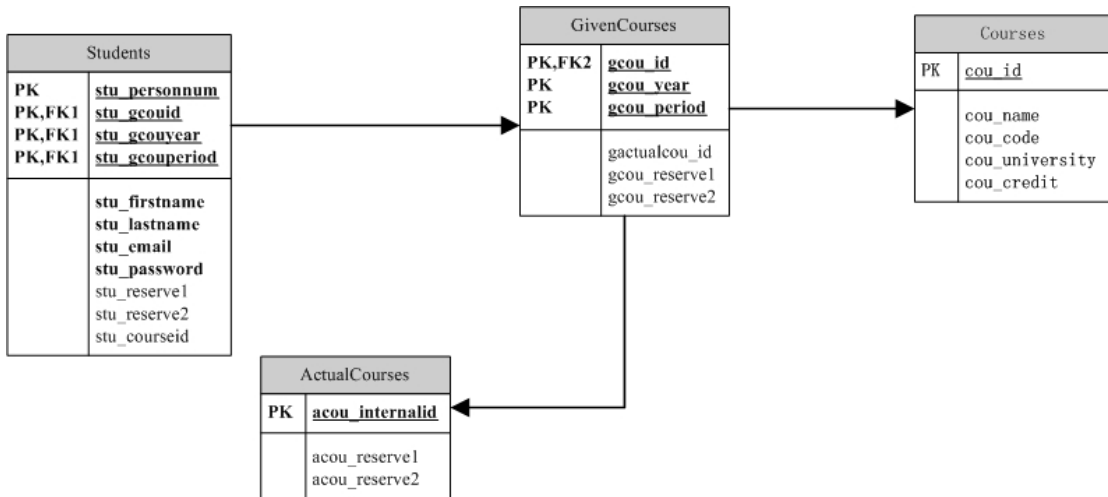


Figure 5-4: The courses management

The steps of creating new courses:

1. If one course determines to use 'New Fire' system to manage the assignment submission, first the administrator or the course responsible creates the course if it does not exist.

Add Courses

CourseName:

CourseCode:

University:

Credit:

Figure 5-5: Add a course

Manager Courses

The list of courses

ID	Name	Code	University	Credit	Edit	Delete	GivenCourses
8	Algorithm	CTH001	CTH	7.5	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>	<input type="button" value="GivenCourses"/>
9	Algorithm	GU001	GU	7.5	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>	<input type="button" value="GivenCourses"/>
10	Distribute System	ADT123	CTH	7.5	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>	<input type="button" value="GivenCourses"/>
11	Concurrent Programing	CTH002	CTH	7.5	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>	<input type="button" value="GivenCourses"/>

Figure 5-6: Course List after adding course

2. After create the course, the administrator needs to create the given course for the course, he should specify the year and the period the course is given. The given course could point to a exist ActualCourseId or generate a new ActualCourseId. If one given course points to an exist ActualCourseId then students registering in the two given course can form the group and submit the assignments together.

Manager Given Courses

CourseName: Concurrent Programing
CourseCode: CTH002
University: CTH
Credit: 7.5

The list of given courses

CourseID	Year	Period	ActualCode	Delete	AssignTeacher	ManageLabs
11	2010	1	25	Delete	AssignTeacher	Managelabs

Create Given Course

Course Year: 2010
Course Period: 1
Actual Course Number: -ActualCourse-
Genetate Actual Code: 25
Generate
Create

Figure 5-7: Create a GivenCourse

Manager Given Courses

CourseName: Concurrent Programing
CourseCode: GU002
University: GU
Credit: 7.5

The list of given courses

CourseID	Year	Period	ActualCode	Delete	AssignTeacher	ManageLabs
12	2010	1	25	Delete	AssignTeacher	Managelabs

Create Given Course

Course Year: 2010
Course Period: 1
Actual Course Number: 25
Create

Figure 5-8: Join an exist GivenCourse

- Until now two given courses from different universities are created and students from different universities can register in different given courses which have the same ActualCourseId.

Enter your personal information

Personal No.	Course Code
<input type="text" value="8512168371"/>	<input type="text" value="- Select Course -"/>
First Name	<input type="text" value="CTH002"/>
<input type="text" value="yi"/>	<input type="text" value="GU002"/>
Password	Password(again)
<input type="password" value="●●●●●●"/>	<input type="password" value="●●●●●●"/>
Email:	
<input type="text" value="yixu@student.chalmers.se"/>	
<input type="button" value="Register"/>	

Figure 5-9: Register the given course

5.4.2. Teacher assignment for courses

When new courses are created, the next step is to assign teachers for the course. The teachers are divided into two roles: course responsible and regular teacher. They have different rights to operate the system; the attribute “title” in table TeaManageCourse distinguishes two kinds of teachers.

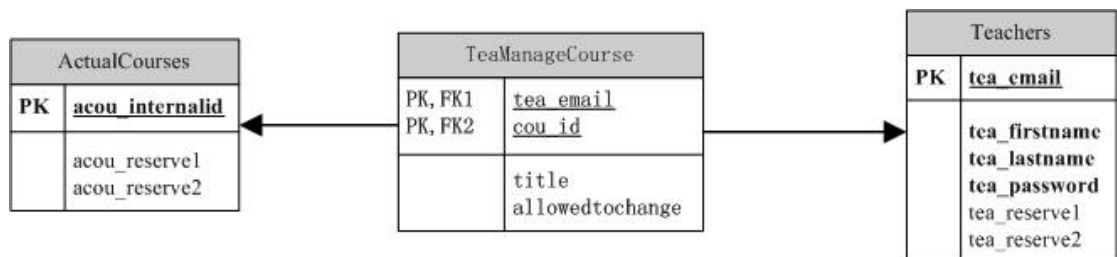


Figure 5-10: Course Management

The steps of add teachers for a course:

1. Choose the given course which needs to add teachers to manage, actually the system will add the teacher for the actual course which the given course point to. That is to say, when assigning a teacher for a given course, the given courses that point to the same actual course will be assigned the teacher as well.

Manager Given Courses

CourseName: Concurrent Programing
CourseCode: CTH002
University: CTH
Credit: 7.5

The list of given courses

CourseID	Year	Period	ActualCode	Delete	AssignTeacher	ManageLabs
11	2010	1	25	Delete	AssignTeacher	Managelabs

Figure 5-11: Assign teachers for one given course

2. In the assignment teachers page, teachers who intend to operate the course can be chosen through add their email. If a teacher leaves the course, he will be deleted from the list. If one teacher is appointed as course responsible for one given course, he cannot be chosen as regular teacher for the given course any more.

Assign Teachers

CourseName: Concurrent Programing
CourseYear: 2010
CoursePeriod: 1
CourseActualCode:25

Add Course Responsible:

- Select Teacher-
- Select Teacher-
- andreass@chalmers.se

The list of Course Responsible

FirstName	LastName	Email Address	Delete
Ali	Shahrokni	nimli@dtek.chalmers.se	Delete

Add Regular Teacher:

The list of Regular Teacher

FirstName	LastName	Email Address	Delete
Phu	Phung	phu.phung@chalmers.se	Delete

Figure 5-12: Assign teachers to given course

5.4.3. Lab management

In the lab management part, the administrator and the course responsible have the right to add and modify the lab information. When creating the new lab, it is necessary to set the workload for the teachers who will grade the lab.

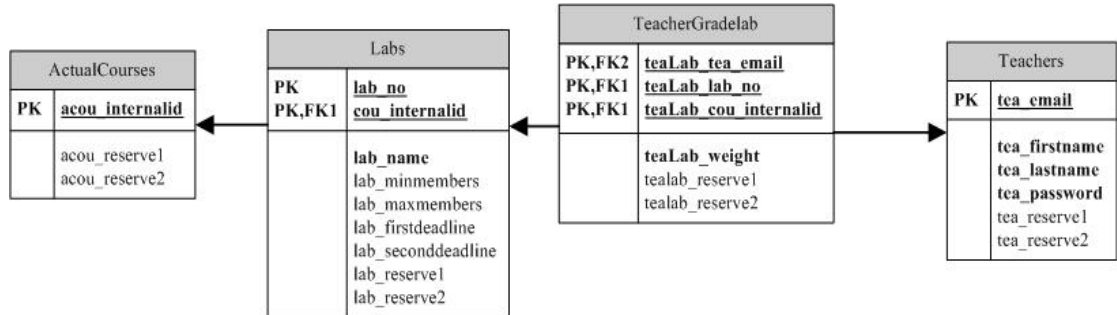


Figure 5-13: Lab management

The steps to add labs and manage labs:

1. Type in the information of the lab, for the first deadline and final deadline of the labs, the format should be the same as the following prompt.

Add Labs

CourseName: Concurrent Programing
CourseYear: 2010
CoursePeriod: 1
CourseActualCode:25

Lab Name:

Minimum members:
Maximum members:

First Deadline: (YYYY-MM-DD hh:mm:ss)
Final Deadline: (YYYY-MM-DD hh:mm:ss)

Grader Submission Distribution:

FirstName	LastName	Email Address	Weight
Ali	Shahrokni	nimli@dtek.chalmers.se	<input type="text" value="1"/>
Phu	Phung	phu.phung@chalmers.se	<input type="text" value="2"/>

Figure 5-14: Add lab

- After the lab is created, the administrator and the course responsible can edit the lab and delete the lab.

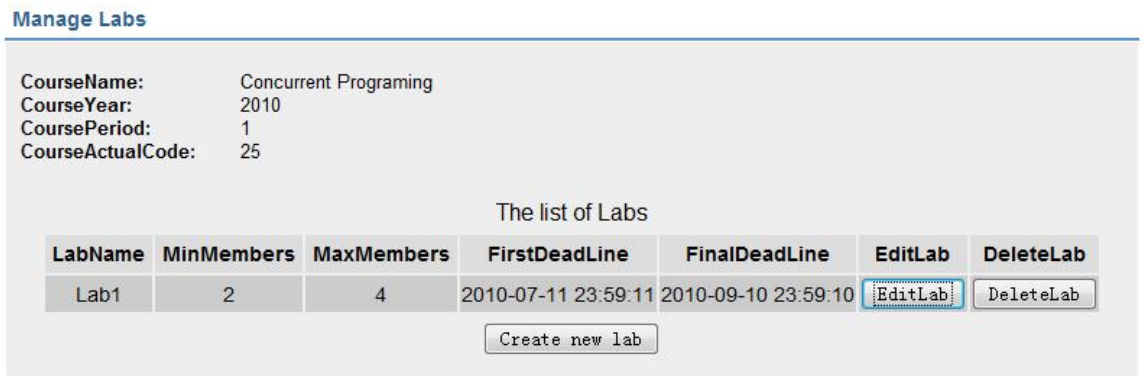


Figure 5-15: Manage lab

5.4.4. Student group management

After students register in a given course, they can join a group or create a group to submit their assignments. Next time when they log in the system again, they will skip the join group page and go to student assignment page directly.

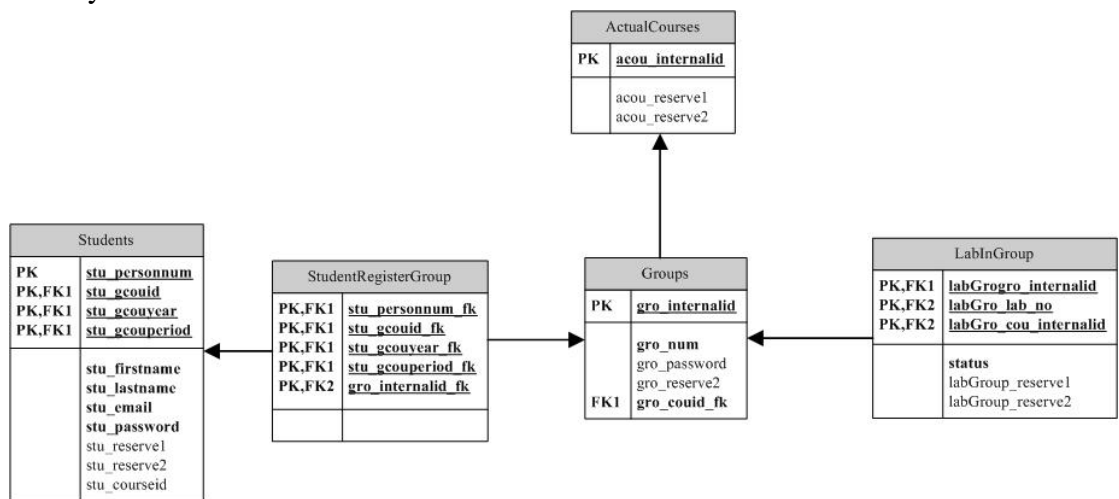


Figure 5-16: Student Group Management

The steps to join the groups:

- Register for one specific course and then login the system by the username and password. Then the join group page will show:

Join a group

Group Number:

Course Password:

Create a new group

Figure 5-17: Join a group

2. If a student clicks the Create Group button, system will create the new group automatically for the student and then send the group number and password to the student’s mailbox, then the lab list will show in next page.

A mail containing group No.17 and password E78fqo has sent to your MailBox

Personal Information

Name: yi xu
 Personal Num: 198512168371
 Course Code: CTH002

Laboration	Status	Next Deadline
Lab1	Unsubmitted	2010-09-10 23:59:59.0
Lab2	Unsubmitted	2010-09-10 23:59:59.0

Figure 5-18: Show lab list

5.4.5. Student upload and download file

After students register in the group, the group can submit their assignments through ‘New Fire’ system. In standard JSF framework, there is no component can be used to upload a file directly. So the MyFaces library which tag begins with letter ‘t’ is added, component `<t:inputFileUpload />` is used to upload file. In terms of security, the form contains file input fields must specify the multipart/form-data encoding and the post method in the `<form>` tag. In ‘New Fire’ system, the file is stored as data type Blob in the database. When executing download operation the system reads the InputStream from database

and uses the OutputStream to download.

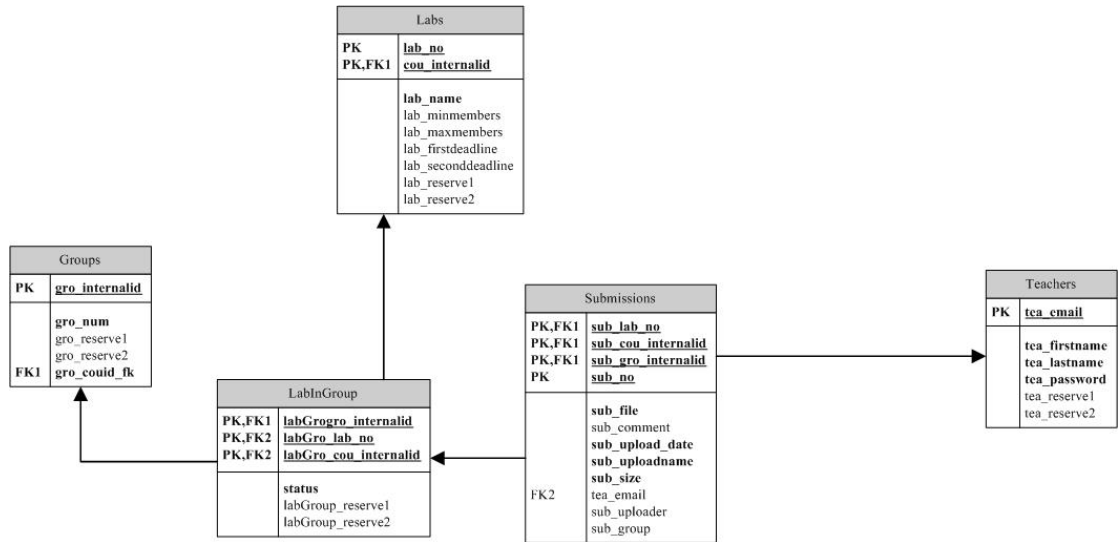


Figure 5-19: Upload and download the file

The steps to upload and download the files

1. In the ‘New Fire’ system, students can upload multi-files for one submission. First students choose the files they want to upload.



Figure 5-20: Upload file

2. After upload the files, the files are categorized as not submitted. The students need to click the “Submit” button to continue submission process.

Lab1

Group 24:
yi xu (black_henry_xu@163.com)

Status: Unsubmitted

Grader:

Final Deadline: 2010-10-20 23:59







Filename	Size	Upload Date	Upload By	
Files not submitted				
Sample.hs	283	2010-06-14 03:12	yi xu	  
Comment for lab3.docx	17286	2010-06-14 03:18	yi xu	  

Figure 5-21: Upload finish before submit

- The submit page records some regulation about the assignments, such as require students to finish the assignments by themselves.

Submit File

- You must finish the assignment by yourself.
- Others regulation

Figure 5-22: Submit the assignment

- After students click “Submit” button, the files will form a submission and system will assign a teacher to grade the submission. At the same time status of the submission will become submitted.







Lab1

Group 24:
yi xu (black_henry_xu@163.com)

Status: Submitted

Grader:

Final Deadline: 2010-10-20 23:59

Filename	Size	Upload Date	Upload By	
Submission 1				
Sample.hs	283	2010-06-14 03:12	yi xu	  
Comment for lab3.docx	17286	2010-06-14 03:18	yi xu	  

Files not submitted

Figure 5-23 Finish submission

5. Students can download the assignment by click the name of file or the download button on the right of each upload item. Students can also view and delete the file.

6. Conclusion

During the half a year master thesis work, I get to know the process of developing web-based software. First of all I learn how to investigate and analyze system requirements in order to get the domain model. Then design the database according to domain model. In the end choose the proper tools and develop the multi-tier system based on the function requirements of the system.

Some experience during the development of 'New Fire' system:

1. Domain model is critical in the process of 'New Fire' development. A good design of domain model can save a lot of work in the future.
2. I use Hibernate to operate the database. If the primary key or foreign key change in the database, it is necessary to change both the XML file and Java file. It always needs a lot of work and it is easy to make errors for the new users. So a good design is very significant for the afterward development.
3. This is the first time I use JSF to develop a system so that there are a lot of new things to learn. The best way to learn the new software knowledge is to write programs.

Until now 'New Fire' system has already implemented most of the functions, there are still some work to do in future:

1. Implement the functions of teacher's part: view and grade the assignment and write the course report.
2. Write the manual for 'New Fire' system.
3. Improve the security of 'New Fire' system.

In summary, through 'New Fire' system development, I learn a lot of knowledge about Web applications; I look forward to finish the rest functions at the end of June.

7. Reference

- [1]. The learning outcome 2009 from Chalmers
- [2]. Java EE vs. Microsoft.NET A comparison of building XML-based web services
by Chad Vawter and Ed Roman
- [3]. Top Ten Reasons to Prefer JSF over Struts by David Geary
<http://softhood.net/blog/2009/07/07/comparing-struts2-with-jsf/>
- [4]. Java Server Faces official website:
<http://myfaces.apache.org/jsfintro.html>
- [5]. The JavaServer Faces Managed Bean Facility
http://www.oracle.com/technology/tech/java/newsletter/articles/jsf_pojo/index.html
- [6]. Making distinctions between different kinds of JSF managed-beans
<http://blog.icefaces.org/blojsom/blog/default/2009/04/23/Making-distinctions-between-different-kinds-of-JSF-managed-beans/>
- [7]. The Life Cycle of a JavaServer Faces Page
<http://www.student.nada.kth.se/~d95-cro/j2eetutorial14/doc/JSFIntro10.html>
- [8]. The official website of Hibernate <http://www.hibernate.org/>
- [9]. Using Hibernate in a Web Application
<http://netbeans.org/kb/docs/web/hibernate-webapp.html#03>
- [10]. http://en.wikipedia.org/wiki/JavaServer_Faces

8. Glossary

‘New Fire’ System	The new student assignment submission system for Computer Science and Department of Chalmers.
Thin Client Architecture	The application is Web-based, and the data processes strictly rely on the server. The users access the server through the client equipments.
JSF(JavaServer Faces)	JSF is a Java-based Web application framework intended to simplify development integration of web-based user interfaces. [10]
Model-View-Controller	MVC is a popular Web-based application architecture, it separates the program into different layers. View layer focus on the front-end; model layer deals with the business logic, and sends the query result to the front end; controller layer implements Servlet function that navigator the system to different pages.
POJO	Plain Old Java Object
Hibernate	Hibernate is a collection of related projects enabling developers to utilize POJO-style domain models in their application in ways extending well beyond object/relational mapping

9. Appendix

Appendix-A:JSF library Tags

1. JSF Core Tags

Tag	Description
f:view	Creates the top-level view
f:subview	Creates a subview of a view
f:facet	Adds a facet to a component
f:attribute	Adds an attribute (key/value) to a component
f:param	Constructs a parameter component
f:actionListener	Adds an action listener to a component
f:valueChangeListener	Adds a valuechange listener to a component
f:setPropertyChangeListener (JSF 1.2)	Adds an action listener to a component that sets a bean property to a given value.
f:converter	Adds an arbitrary converter to a component
f:convertDateTime	Adds a datetime converter to a component
f:convertNumber	Adds a number converter to a component
f:validator	Adds a validator to a component
f:validateDoubleRange	Validates a double range for a component's value
f:validateLength	Validates the length of a component's value
f:validateLongRange	Validates a long range for a component's value
f:loadBundle	Loads a resource bundle, stores properties as a Map
f:selectitems	Specifies items for a select one or select many component
f:selectitem	Specifies an item for a select one or select many component
f:verbatim	Adds markup to a JSF page

2. JSF HTML Tags

Tag	Description
h:form	HTML form
h:inputText	Single-line text input control.
h:inputTextArea	Multiline text input control.
h:inputSecret	Password input control.
h:inputHidden	Hidden field
h:outputLabel	Label for another component for accessibility
h:outputLink	HTML anchor.
h:outputFormat	Like outputText, but formats compound messages
h:outputText	Single-line text output.
h:commandButton	Button: submit, reset, or pushbutton.
h:commandLink	Link that acts like a pushbutton.
h:message	Displays the most recent message for a component
h:messages	Displays all messages

h:graphicImage	Displays an image.
h:selectOneListbox	Single-select listbox.
h:selectOneMenu	Single-select menu.
h:selectOneRadio	Set of radio buttons.
h:selectBooleanCheckbox	Checkbox.
h:selectManyCheckbox	Set of checkboxes.
h:selectManyListbox	Multiselect listbox.
h:selectManyMenu	Multiselect menu.
h:panelGrid	HTML table
h:panelGroup	Two or more components that are laid out as one
h:dataTable	A feature-rich table control
h:column	Column in a dataTable

The two tables are from the book Core JavaServer Faces by David Geary and Cay Horstmann, Sun Microsystems Press 2004.

