武汉纺织大学

Web应用开发课程设计

**基于SSM的学生选课系统**

**学 院： 数学与计算机学院**

**班 级： 物联网11802**

**姓 名： 喻言**

**学 号： 1804240707**

**指导老师： 聂刚**

**成 绩：**

**完成日期： 2020年12月15日**

目 录

[1 需求分析 1](#_Toc22456)

[1.1用户登陆 1](#_Toc6247)

[1.2主菜单 1](#_Toc15319)

[1.3各项功能 1](#_Toc14458)

[1.3.1 课程查询 1](#_Toc1897)

[1.3.2 后台功能 1](#_Toc7940)

[1.3.3 添加修改课程 2](#_Toc4651)

[1.3.4 菜单管理 2](#_Toc30015)

[1.3.5 课程明细 3](#_Toc5914)

[1.3.6 个人信息 3](#_Toc3637)

[2 开发环境与功能介绍 3](#_Toc21868)

[2.1开发环境 3](#_Toc16712)

[2.2 功能介绍 4](#_Toc13261)

[3 系统实现 10](#_Toc20261)

[3.1 mysql数据库 10](#_Toc17930)

[3.2主要程序 10](#_Toc14003)

[3.2.1 pom文件 10](#_Toc32289)

[3.3 Spring的配置文件 11](#_Toc7838)

[3.3.1 web.xml 10](#_Toc32289)

[3.3.1 spring\_root.xml 1](#_Toc32289)1

[3.3.1 Spring\_web.xml 1](#_Toc32289)1

[4 页面展示 15](#_Toc7918)

[5 课设总结 15](#_Toc18618)

# 1 需求分析

设计一个学生选课系统，实现学生用户以下功能：登陆、查看所有课程、选课、退选、成绩查询统计、修改密码等，具体要求如下：

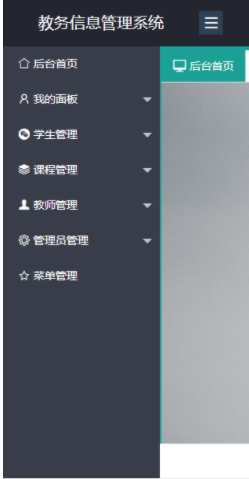
## 1.1用户登陆

可以以“学生”，“教师”，“管理员”三种身份进行登录。学生，教师可以注册登录，也可以由管理员添加。

登陆时需要输入用户类型，用户名，密码与验证码

注册时需要输入用户类型，用户名，密码，确认密码，性别，手机号。

## 1.2主菜单



## 1.3各项功能

### 1.3.1 学生查询

能够根据学生年纪姓名性别等信息查询该学生的信息。该学生的电话，邮箱，以及信息状态，年纪，注册时间。

### 1.3.2 后台功能

能够进行学生管理，课程管理，教师管理，管理员管理以及菜单管理。

### 1.3.3 添加修改课程

在“学生选课菜单”里的“选择课程”的栏目中可以查看所有课程，并进行课程选择。能够看到课程的名称，上课地点，课程类型和该课程的学分，授课老师。

### 1.3.4 菜单管理

可以添加菜单，编辑菜单以及删除菜单。这里有多种顶级菜单和添加子菜单。

### 1.3.5 课程明细

课程明细可以看到不同学生所选择的课程，以及该课程的名称和上课地点。

### 1.3.6 个人信息

能够修改个人信息，修改密码。个人信息根据登录的类别分别为学生信息，教师信息和管理员信息。

# 2 开发环境和功能介绍

## 2.1开发环境

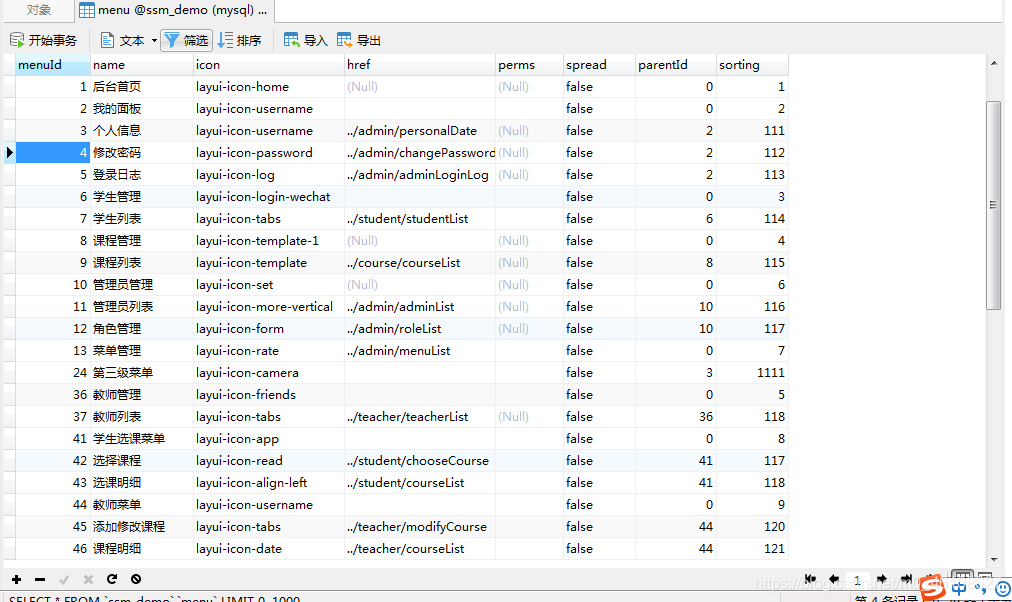
1. 开发软件 idea 也可以用eclipse导入
2. 前端 lyaui
3. 后端 spring springmvc mybatis
4. 分页用的pagehelper
5. maven管理项目
6. tomcat 7
7. 数据库mysql 5.7

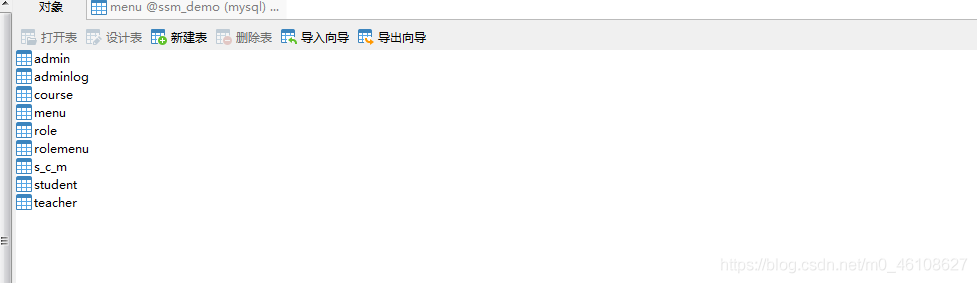
## 2.2 功能介绍

管理员 学生 教师 三种登录  
学生 教师可以注册登录 也可以由管理员添加  
管理员可以添加学生 教师 其他管理员并分配权限 编辑菜单 编辑角色 修改 学生 教师信息 编辑课程 查看学生教师 课程的所有信息  
学生登录 可以查看选课信息 编辑个人信息  
教师登录 可以添加编辑课程信息 查看选课情况

# 3 系统实现

## 3.1 mysql数据库





## 3.2 主要程序

### 3.2.1 pom文件

<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.mecol.demo</groupId>

<artifactId>ssm\_mysql\_hotel</artifactId>

<version>1.0-SNAPSHOT</version>

<packaging>war</packaging>

<name>SSM\_demo Maven Webapp</name>

<!-- FIXME change it to the project's website -->

<url>http://www.example.com</url>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<maven.compiler.source>1.7</maven.compiler.source>

<maven.compiler.target>1.7</maven.compiler.target>

</properties>

<dependencies>

<!-- spring 依赖包 -->

<!-- spring web jar包 -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-web</artifactId>

<version>4.3.8.RELEASE</version>

</dependency>

<!-- spring aop jar包 -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-aop</artifactId>

<version>4.3.8.RELEASE</version>

</dependency>

<!-- spring 驱动jar包-->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-jdbc</artifactId>

<version>4.3.8.RELEASE</version>

</dependency>

<!-- spring mvc 依赖包 -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>4.3.8.RELEASE</version>

</dependency>

<!-- mybatis 依赖包 -->

<!--mybatis jar包 -->

<dependency>

<groupId>org.mybatis</groupId>

<artifactId>mybatis</artifactId>

<version>3.4.6</version>

</dependency>

<!-- mybatis spring-->

<dependency>

<groupId>org.mybatis</groupId>

<artifactId>mybatis-spring</artifactId>

<version>1.3.2</version>

</dependency>

<!-- 连接数据库驱动的jar包 我这里使用的是 jtds链接sqlserver数据库的驱动包 -->

<dependency>

<groupId>net.sourceforge.jtds</groupId>

<artifactId>jtds</artifactId>

<version>1.2.4</version>

</dependency>

<!-- MySql -->

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>8.0.11</version>

</dependency>

<!-- 连接接池 c3p0 -->

<dependency>

<groupId>com.mchange</groupId>

<artifactId>c3p0</artifactId>

<version>0.9.5.2</version>

</dependency>

<!--调试输出的log4j-->

<dependency>

<groupId>log4j</groupId>

<artifactId>log4j</artifactId>

<version>1.2.17</version>

</dependency>

<!-- jstl jar包 -->

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>jstl</artifactId>

<version>1.2</version>

</dependency>

<!-- jackson begin -->

<dependency>

<groupId>com.fasterxml.jackson.core</groupId>

<artifactId>jackson-databind</artifactId>

<version>2.8.8</version>

</dependency>

<dependency>

<groupId>org.apache.commons</groupId>

<artifactId>commons-lang3</artifactId>

<version>3.7</version>

</dependency>

<dependency>

<groupId>org.codehaus.groovy</groupId>

<artifactId>groovy-all</artifactId>

<version>2.2.2</version>

</dependency>

<dependency>

<groupId>org.apache.poi</groupId>

<artifactId>poi</artifactId>

<version>3.7</version>

</dependency>

<!--下面两个包 解决采用@ResponseBody 前台ajax请求 报错406-->

<dependency>

<groupId>org.codehaus.jackson</groupId>

<artifactId>jackson-core-asl</artifactId>

<version>1.8.5</version>

</dependency>

<dependency>

<groupId>org.codehaus.jackson</groupId>

<artifactId>jackson-mapper-asl</artifactId>

<version>1.8.5</version>

</dependency>

<!--分页查询的工具-->

<dependency>

<groupId>com.github.pagehelper</groupId>

<artifactId>pagehelper</artifactId>

<version>4.1.4</version>

</dependency>

<!-- json处理工具包 -->

<dependency>

<groupId>com.google.code.gson</groupId>

<artifactId>gson</artifactId>

<version>2.8.5</version>

</dependency>

<dependency>

<groupId>commons-io</groupId>

<artifactId>commons-io</artifactId>

<version>2.6</version>

</dependency>

<dependency>

<groupId>commons-fileupload</groupId>

<artifactId>commons-fileupload</artifactId>

<version>1.4</version>

</dependency>

<!-- 单元测试 -->

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.11</version>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<finalName>SSM\_demo</finalName>

<pluginManagement><!-- lock down plugins versions to avoid using Maven defaults (may be moved to parent pom) -->

<plugins>

<plugin>

<artifactId>maven-clean-plugin</artifactId>

<version>3.1.0</version>

</plugin>

<!-- see http://maven.apache.org/ref/current/maven-core/default-bindings.html#Plugin\_bindings\_for\_war\_packaging -->

<plugin>

<artifactId>maven-resources-plugin</artifactId>

<version>3.0.2</version>

</plugin>

<plugin>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.8.0</version>

</plugin>

<plugin>

<artifactId>maven-surefire-plugin</artifactId>

<version>2.22.1</version>

</plugin>

<plugin>

<artifactId>maven-war-plugin</artifactId>

<version>3.2.2</version>

</plugin>

<plugin>

<artifactId>maven-install-plugin</artifactId>

<version>2.5.2</version>

</plugin>

<plugin>

<artifactId>maven-deploy-plugin</artifactId>

<version>2.8.2</version>

</plugin>

</plugins>

</pluginManagement>

</build>

</project>

## 3.3 Spring的配置文件

### 3.3.1 web.xml

<?xml version="1.0" encoding="UTF-8"?>

<web-app version="2.4"

xmlns="http://java.sun.com/xml/ns/j2ee"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee

http://java.sun.com/xml/ns/j2ee/web-app\_2\_4.xsd">

<display-name>Archetype Created Web Application</display-name>

<!--配置 Spring 容器-->

<context-param>

<param-name>contextConfigLocation</param-name>

<param-value>classpath:spring\_root.xml</param-value>

</context-param>

<!--中文乱码处理过滤-->

<!-- 编码过滤器 -->

<filter>

<filter-name>characterEncodingFilter</filter-name>

<filter-class>org.springframework.web.filter.CharacterEncodingFilter</filter-class>

<init-param>

<param-name>encoding</param-name>

<param-value>UTF-8</param-value>

</init-param>

</filter>

<filter-mapping>

<filter-name>characterEncodingFilter</filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

<listener>

<listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>

</listener>

<!-- Jasperreport Config Begin -->

<servlet>

<servlet-name>JasperImageServlet</servlet-name>

<servlet-class>net.sf.jasperreports.j2ee.servlets.ImageServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>JasperImageServlet</servlet-name>

<url-pattern>/servlets/image</url-pattern>

</servlet-mapping>

<!-- Jasperreport Config End -->

<!--配置 Spring mvc 容器-->

<!--将所有请求，都交给Spring MVC 处理-->

<servlet>

<servlet-name>webs</servlet-name>

<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

<init-param>

<param-name>contextConfigLocation</param-name>

<param-value>classpath:spring\_web.xml</param-value>

</init-param>

</servlet>

<!--将 请求 截取 转交给 webs -->

<servlet-mapping>

<servlet-name>webs</servlet-name>

<url-pattern>/</url-pattern>

</servlet-mapping>

</web-app>

### 3.3.2 spring\_root.xml

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:contxt="http://www.springframework.org/schema/context" xmlns:p="http://www.springframework.org/schema/p"

xmlns:tx="http://www.springframework.org/schema/tx"

xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd

http://www.springframework.org/schema/context

http://www.springframework.org/schema/context/spring-context.xsd http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx.xsd">

<!-- 扫描 service 包下面使用@Component/@Service/@Repository/@Controller 注解的类 将其实例化，放入容器中 -->

<contxt:component-scan base-package="com.mecol.hotel.service" />

<!--创建数据源-->

<bean id="dataSource" class="com.mchange.v2.c3p0.ComboPooledDataSource">

<!-- 数据库驱动 -->

<property name="driverClass" value="com.mysql.jdbc.Driver" />

<!-- 数据库连接 -->

<property name="jdbcUrl" value="jdbc:mysql://127.0.0.1:3306/hotel?characterEncoding=utf8&amp;useSSL=false&amp;serverTimezone=CTT"/>

<!-- 数据库用户名 -->

<property name="user" value="root"/>

<!-- 数据库密码 -->

<property name="password" value="123489"/>

</bean>

<!-- 配置mybatis -->

<bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">

<!-- 数据源的实例化 -->

<property name="dataSource" ref="dataSource"/>

<!-- 映射 DML 语法 只要文件夹变蓝色 就是classpath 所以要有个mapper文件夹 下面所有的.xml都会映射-->

<property name="mapperLocations" value="classpath:mapper/\*.xml"/>

<!--加载mybatis自身的配置文件-->

<property name="configLocation" value="classpath:mybatis-config.xml"/>

</bean>

<!-- 映射 数据访问层 ( dao ) 中的接口 -->

<bean class="org.mybatis.spring.mapper.MapperScannerConfigurer">

<property name="sqlSessionFactoryBeanName" value="sqlSessionFactory"/>

<property name="basePackage" value="com.mecol.hotel.dao"/>

</bean>

<!-- 文件上传 -->

<bean id="multipartResolver" class="org.springframework.web.multipart.commons.CommonsMultipartResolver">

<!-- 设置上传文件的最大尺寸为10MB -->

<property name="maxUploadSize">

<value>10485760</value>

</property>

<!--设置编码，与jsp页面编码一致-->

<property name="defaultEncoding">

<value>UTF-8</value>

</property>

</bean>

<!-- JDBC事务管理器 -->

<bean id="transactionManager"

class="org.springframework.jdbc.datasource.DataSourceTransactionManager"

p:dataSource-ref="dataSource"/>

<!-- 启用支持annotation注解方式事务管理 -->

<tx:annotation-driven transaction-manager="transactionManager"/>

</beans>

### 3.3.3 Spring\_web.xml

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:contxt="http://www.springframework.org/schema/context" xmlns:p="http://www.springframework.org/schema/p"

xmlns:tx="http://www.springframework.org/schema/tx"

xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd

http://www.springframework.org/schema/context

http://www.springframework.org/schema/context/spring-context.xsd http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx.xsd">

<!-- 扫描 service 包下面使用@Component/@Service/@Repository/@Controller 注解的类 将其实例化，放入容器中 -->

<contxt:component-scan base-package="com.mecol.hotel.service" />

<!--创建数据源-->

<bean id="dataSource" class="com.mchange.v2.c3p0.ComboPooledDataSource">

<!-- 数据库驱动 -->

<property name="driverClass" value="com.mysql.jdbc.Driver" />

<!-- 数据库连接 -->

<property name="jdbcUrl" value="jdbc:mysql://127.0.0.1:3306/hotel?characterEncoding=utf8&amp;useSSL=false&amp;serverTimezone=CTT"/>

<!-- 数据库用户名 -->

<property name="user" value="root"/>

<!-- 数据库密码 -->

<property name="password" value="123489"/>

</bean>

<!-- 配置mybatis -->

<bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">

<!-- 数据源的实例化 -->

<property name="dataSource" ref="dataSource"/>

<!-- 映射 DML 语法 只要文件夹变蓝色 就是classpath 所以要有个mapper文件夹 下面所有的.xml都会映射-->

<property name="mapperLocations" value="classpath:mapper/\*.xml"/>

<!--加载mybatis自身的配置文件-->

<property name="configLocation" value="classpath:mybatis-config.xml"/>

</bean>

<!-- 映射 数据访问层 ( dao ) 中的接口 -->

<bean class="org.mybatis.spring.mapper.MapperScannerConfigurer">

<property name="sqlSessionFactoryBeanName" value="sqlSessionFactory"/>

<property name="basePackage" value="com.mecol.hotel.dao"/>

</bean>

<!-- 文件上传 -->

<bean id="multipartResolver" class="org.springframework.web.multipart.commons.CommonsMultipartResolver">

<!-- 设置上传文件的最大尺寸为10MB -->

<property name="maxUploadSize">

<value>10485760</value>

</property>

<!--设置编码，与jsp页面编码一致-->

<property name="defaultEncoding">

<value>UTF-8</value>

</property>

</bean>

<!-- JDBC事务管理器 -->

<bean id="transactionManager"

class="org.springframework.jdbc.datasource.DataSourceTransactionManager"

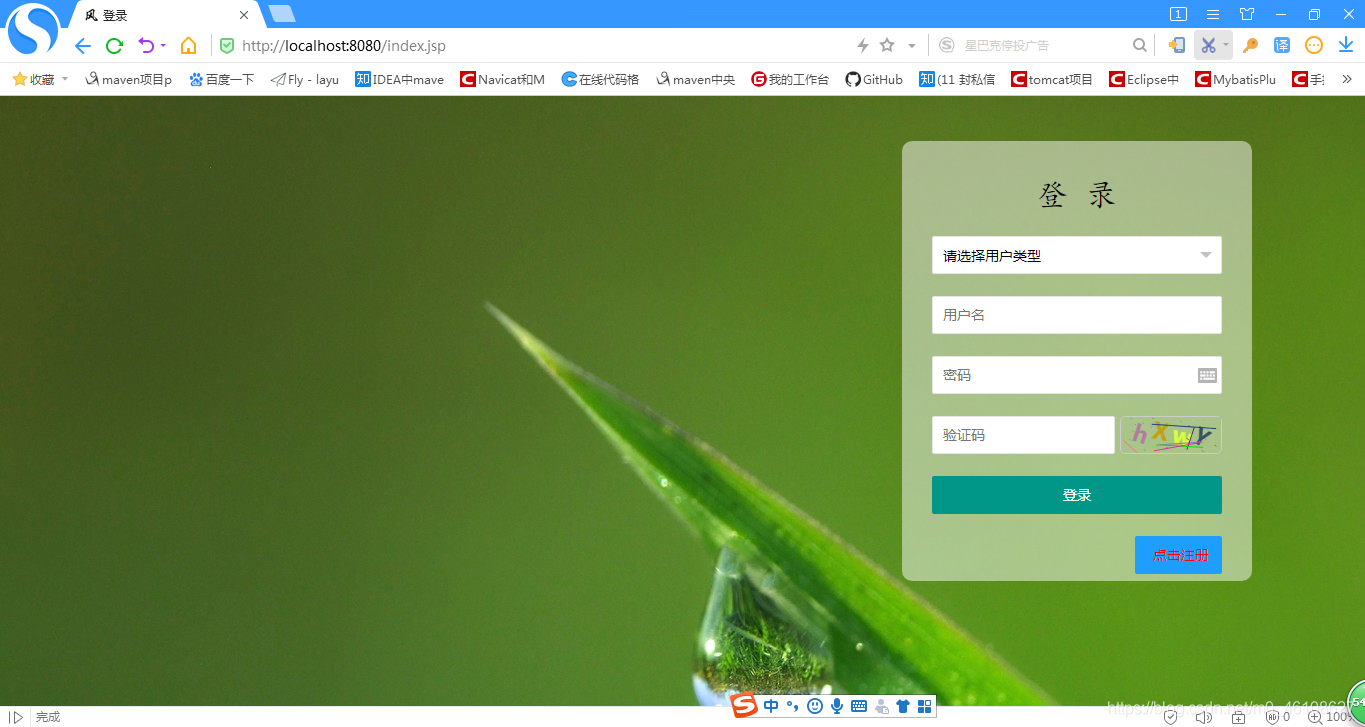
p:dataSource-ref="dataSource"/>

<!-- 启用支持annotation注解方式事务管理 -->

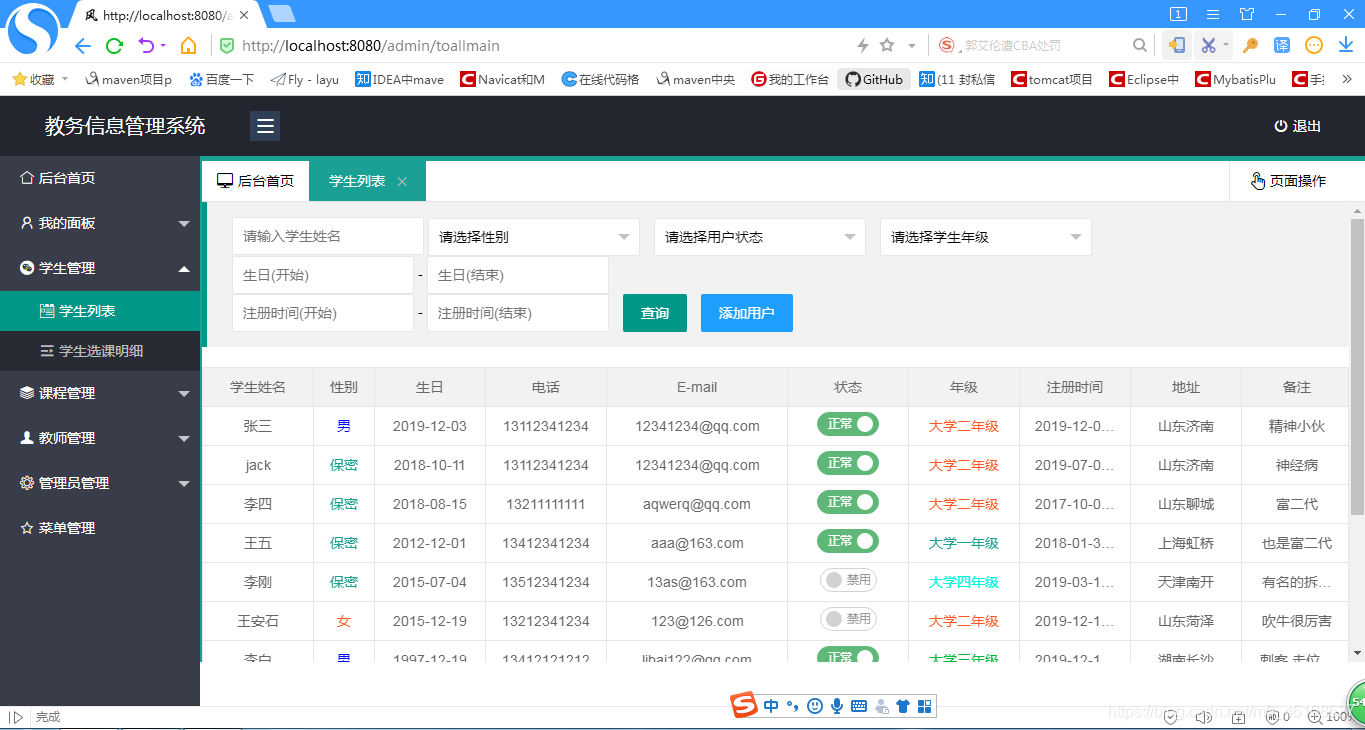
<tx:annotation-driven transaction-manager="transactionManager"/>

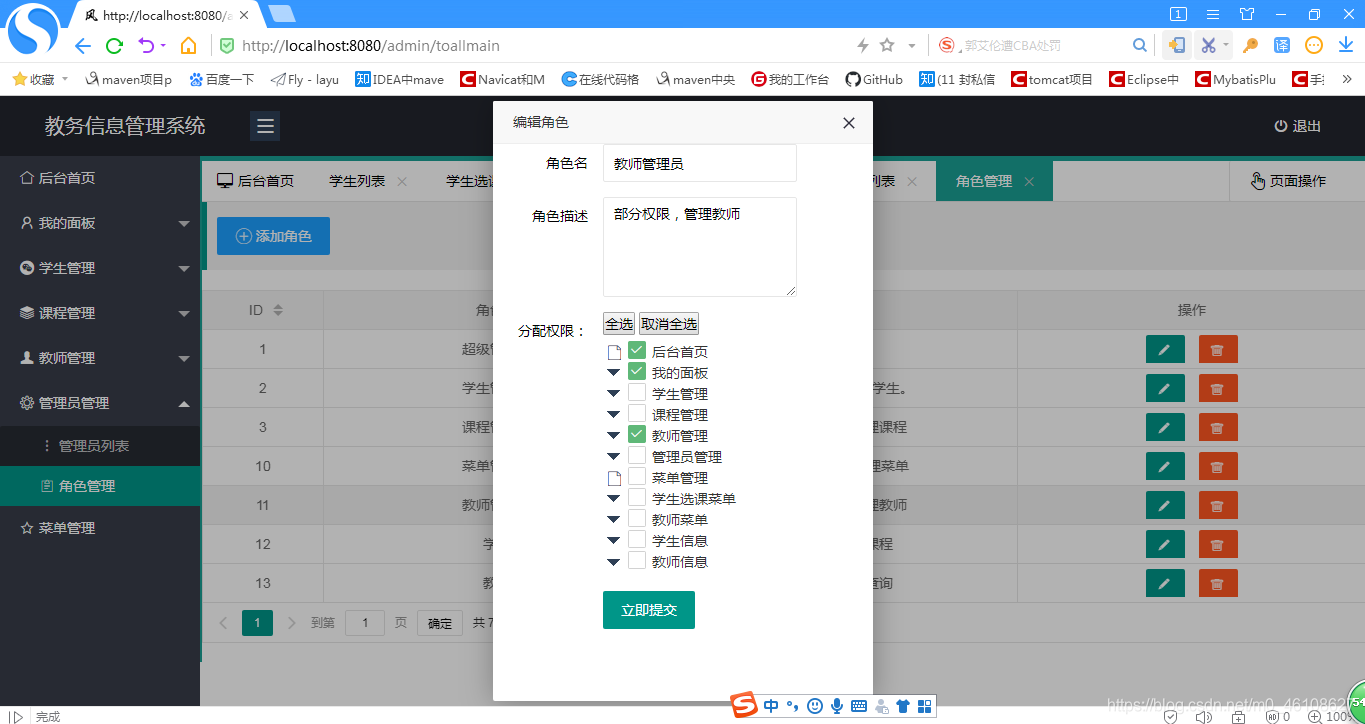
</beans>

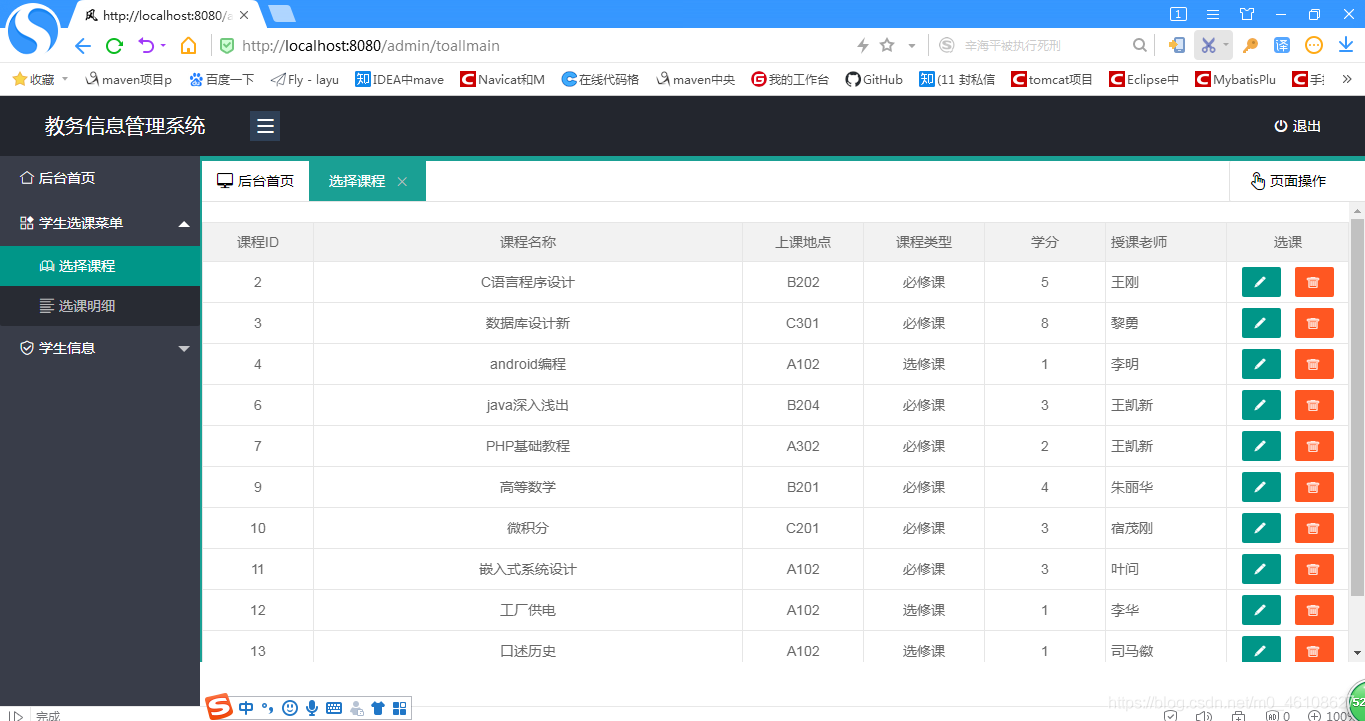
# 4 页面展示

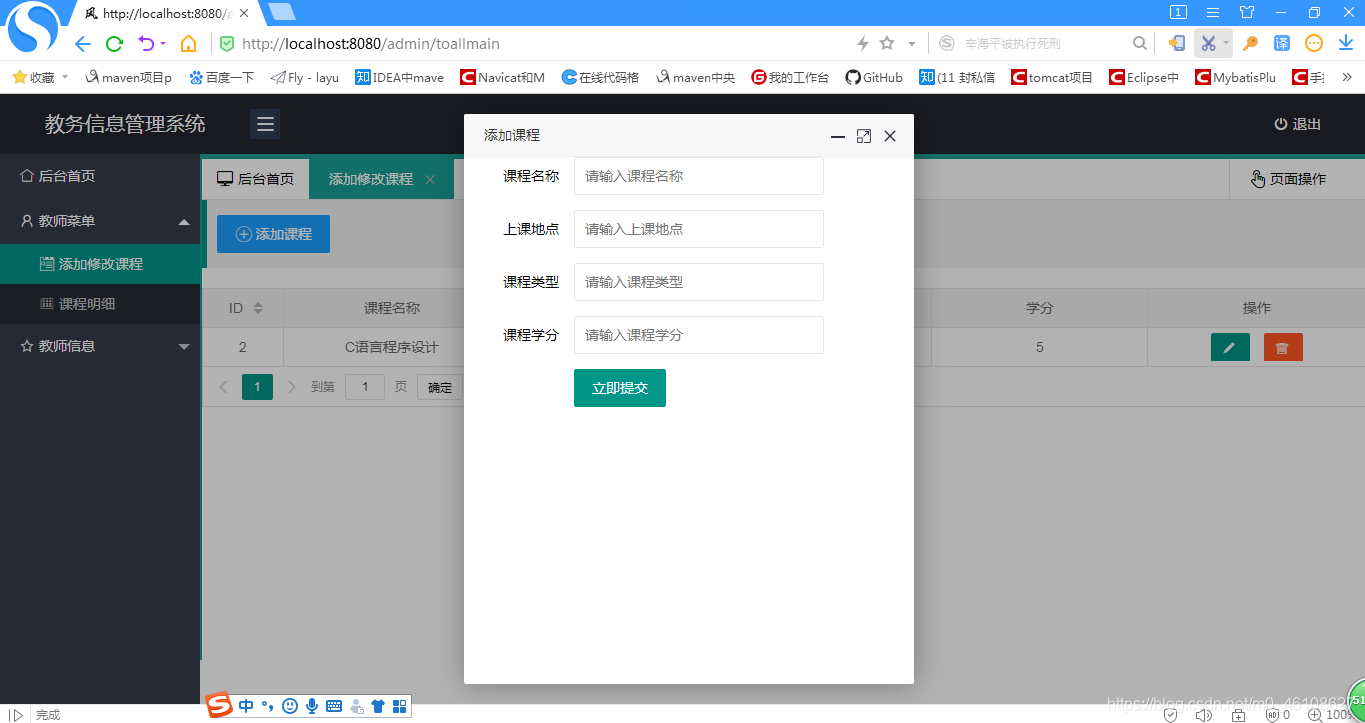


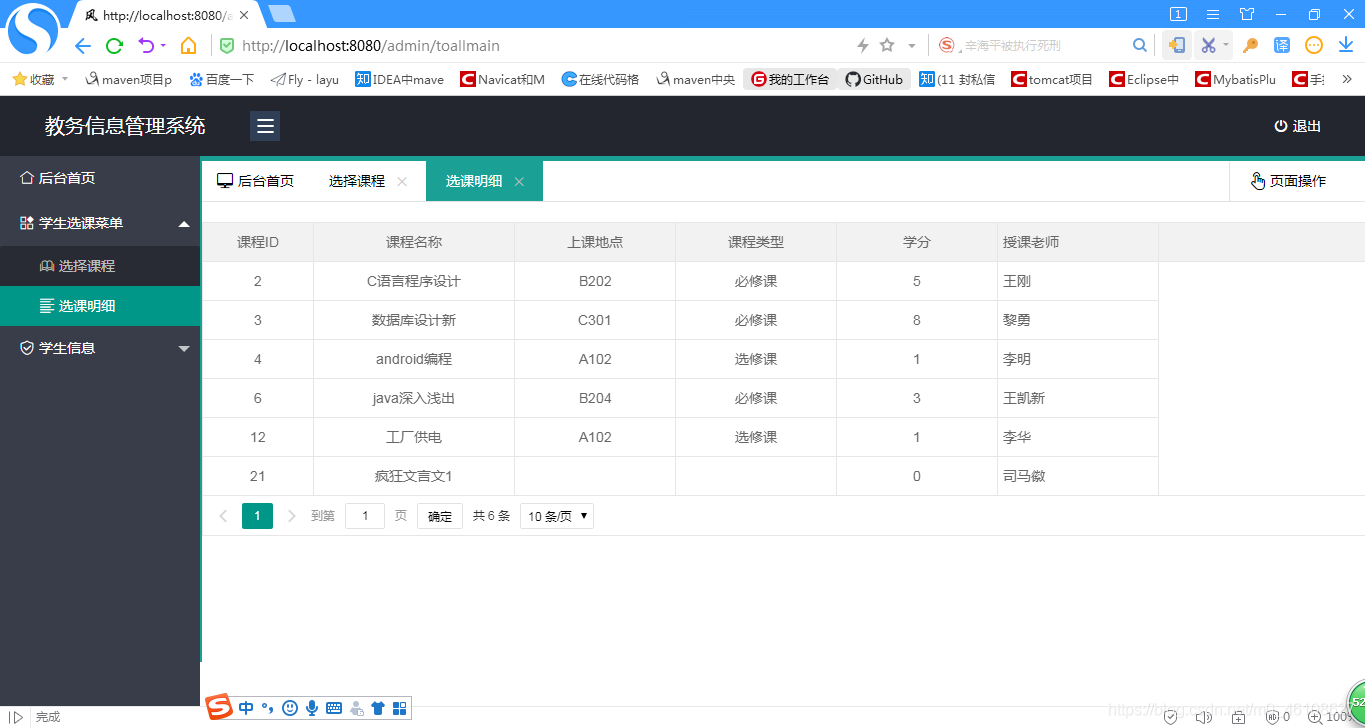


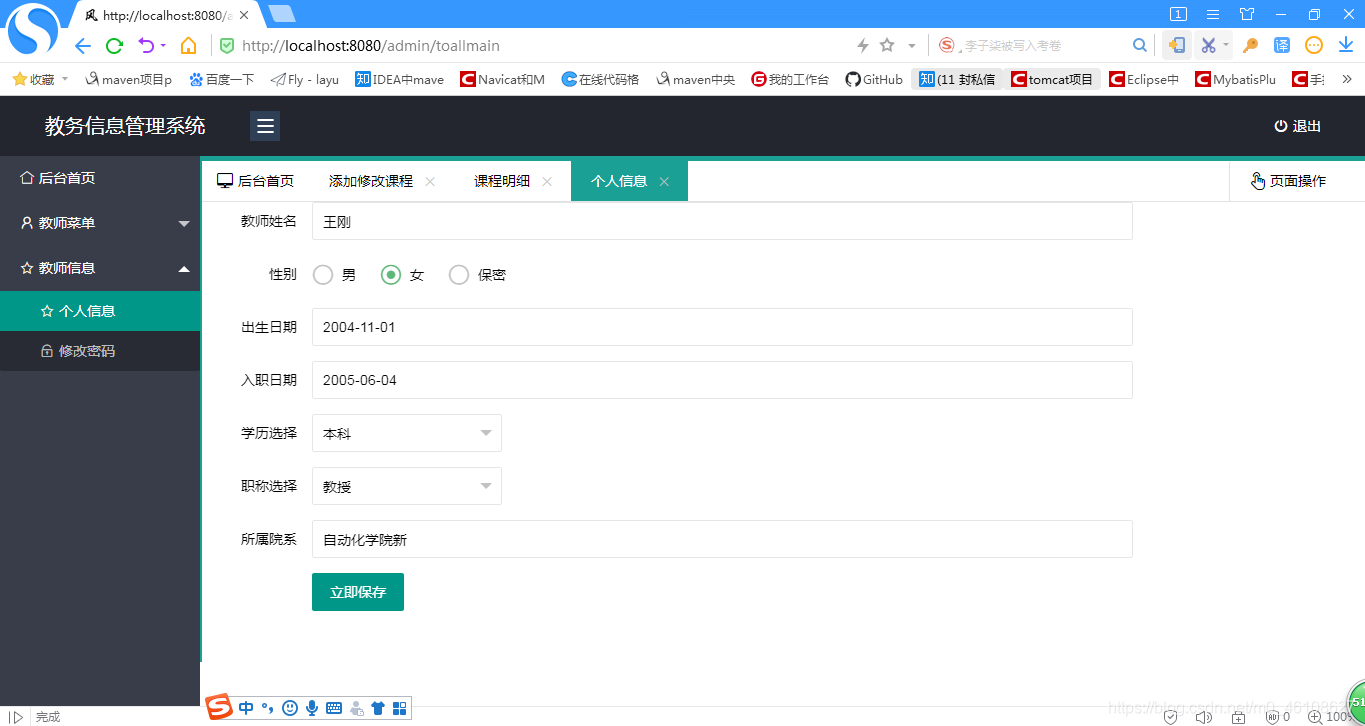












# 5 课设总结

Web的应用开发使我能够掌握和使用本学期学习的web编程的基本内容。在实验过程中，我利用课堂老师的知识和参考教材的知识，通过小组讨论，实现了这个简单的web应用程序。对于我来说，这也是我已经学习了半年的web程序的一项小成就。尽管我制作了这个小应用，但这并不意味着我已经完全掌握了程序设计的知识。如果我想对web有更深入的了解和熟练的应用，或者如果我操纵web编程，我将无法满足我现在所学的知识。我必须自己学习更多。在知识实验的过程中，老师还向我提出了指导性建议。因此，我应该感谢这里的老师。同时，我应该向老师学习更多，并希望将来能学到更多。当然，实验中仍然存在一些缺陷，但是我将在以后的研究中对其进行改进，并有望更加完善。