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11801**

**姓 名： 赵子航**

**学 号： 1804240127**

**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)

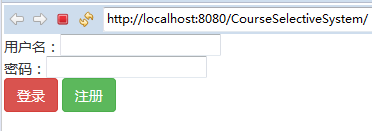
[5 课设总结 15](#_Toc18618)

# 1 需求分析

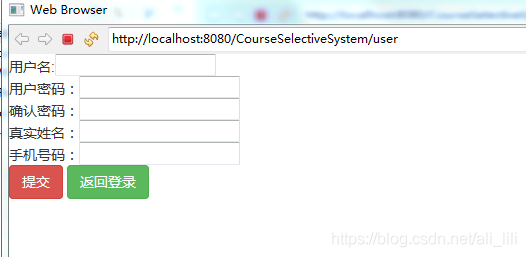
设计一个学生选课系统，实现学生用户以下功能：登陆、查看所有课程、选课、退选、成绩查询统计、修改密码等，具体要求如下：

## 1.1用户登陆

登录页面：



注册页面：



## 1.2各项功能

### 1.2.1 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。

### 1.2.2 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。

### 1.2.3 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。

### 1.2.4 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。

### 1.2.5 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。

### 1.2.6 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

之前用的是h2数据库，但是感觉h2的使用人数好像比较少，后来就下载了mysql，直接在里面建立三个比较简单的数据库

create schema courseselectionsystem;

create table courseselectionsystem.user(

username varchar(100),

password varchar(200),

realname varchar(100),

phone int(11)

);

create table courseselectionsystem.course(

coursename varchar(20),

teacher varchar(20),

number int(100)

);

create table courseselectionsystem.user\_course(

username varchar(100) references user(username),

coursename varchar(20) references course(coursename)

)

## 3.2 主要程序

### 3.2.1 pom文件

<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/maven-v4\_0\_0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>lixiaoli</groupId>

<artifactId>CourseSelectiveSystem</artifactId>

<packaging>war</packaging>

<version>0.0.1-SNAPSHOT</version>

<name>CourseSelectiveSystem Maven Webapp</name>

<url>http://maven.apache.org</url>

<dependencies><dependency>

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

<artifactId>pagehelper</artifactId>

<version>5.0.0</version>

</dependency>

<!-- 1.springmvc -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-web</artifactId>

<version>5.1.3.RELEASE</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>5.1.3.RELEASE</version>

</dependency>

<!-- 返回json字符串支持 -->

<!-- https://mvnrepository.com/artifact/com.fasterxml.jackson.core/jackson-databind -->

<dependency>

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

<artifactId>jackson-databind</artifactId>

<version>2.9.7</version>

</dependency>

<!-- hibernate-validator jsr303校验 -->

<!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-validator -->

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-validator</artifactId>

<version>5.3.5.Final</version>

</dependency>

<!-- 2.spring -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-beans</artifactId>

<version>5.1.3.RELEASE</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>5.1.3.RELEASE</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-core</artifactId>

<version>5.1.3.RELEASE</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-expression</artifactId>

<version>5.1.3.RELEASE</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-jdbc -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-jdbc</artifactId>

<version>5.1.3.RELEASE</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-aspects -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-aspects</artifactId>

<version>5.1.3.RELEASE</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-test -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-test</artifactId>

<version>5.1.3.RELEASE</version>

<scope>test</scope>

</dependency>

<!-- https://mvnrepository.com/artifact/commons-logging/commons-logging -->

<dependency>

<groupId>commons-logging</groupId>

<artifactId>commons-logging</artifactId>

<version>1.2</version>

</dependency>

<!-- 3.mybatis -->

<!-- https://mvnrepository.com/artifact/org.mybatis/mybatis -->

<dependency>

<groupId>org.mybatis</groupId>

<artifactId>mybatis</artifactId>

<version>3.4.5</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.mybatis/mybatis-spring -->

<dependency>

<groupId>org.mybatis</groupId>

<artifactId>mybatis-spring</artifactId>

<version>1.3.2</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.mybatis.generator/mybatis-generator-core -->

<dependency>

<groupId>org.mybatis.generator</groupId>

<artifactId>mybatis-generator-core</artifactId>

<version>1.3.5</version>

</dependency>

<!-- https://mvnrepository.com/artifact/com.github.pagehelper/pagehelper -->

<!-- <dependency>

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

<artifactId>pagehelper</artifactId>

<version>5.1.3</version>

</dependency> -->

<!-- 日志 -->

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-api</artifactId>

<version>1.7.12</version>

</dependency>

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-log4j12</artifactId>

<version>1.7.12</version>

</dependency>

<dependency>

<groupId>log4j</groupId>

<artifactId>log4j</artifactId>

<version>1.2.17</version>

</dependency>

<!-- 4.数据库连接池和驱动 -->

<!-- https://mvnrepository.com/artifact/c3p0/c3p0 -->

<dependency>

<groupId>c3p0</groupId>

<artifactId>c3p0</artifactId>

<version>0.9.1.2</version>

</dependency>

<dependency>

<groupId>mysql</groupId>

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

<version>5.1.38</version>

</dependency>

<!-- 5.其他(jstl servlet-api junit) -->

<!-- https://mvnrepository.com/artifact/jstl/jstl -->

<dependency>

<groupId>jstl</groupId>

<artifactId>jstl</artifactId>

<version>1.2</version>

</dependency>

<!-- https://mvnrepository.com/artifact/javax.servlet/javax.servlet-api -->

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>javax.servlet-api</artifactId>

<version>3.1.0</version>

<scope>provided</scope>

</dependency>

<!-- https://mvnrepository.com/artifact/junit/junit -->

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.12</version>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<finalName>sss-test</finalName>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

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

<version>3.0</version>

<configuration>

<!-- 指定source和target的版本 -->

<source>1.8</source>

<target>1.8</target>

</configuration>

</plugin>

</plugins>

</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="3.0" xmlns="http://java.sun.com/xml/ns/javaee"

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

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

http://java.sun.com/xml/ns/javaee/web-app\_3\_0.xsd">

<!-- 启动spring的容器 -->

<context-param>

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

<param-value>classpath:applicationContext.xml</param-value>

</context-param>

<listener>

<listener-class>

org.springframework.web.context.ContextLoaderListener

</listener-class>

</listener>

<!-- Spring MVC的前端控制器 -->

<servlet>

<servlet-name>springMVC</servlet-name>

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

<init-param>

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

<param-value>classpath:SpringMVC-servlet.xml</param-value>

</init-param>

<load-on-startup>1</load-on-startup>

</servlet>

<servlet-mapping>

<servlet-name>springMVC</servlet-name>

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

</servlet-mapping>

<!-- 字符编码过滤器 -->

<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>

<init-param>

<param-name>forceRequestEncoding</param-name>

<param-value>true</param-value>

</init-param>

<init-param>

<param-name>forceResposeEncoding</param-name>

<param-value>true</param-value>

</init-param>

</filter>

<filter-mapping>

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

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

</filter-mapping>

<!-- 使用rest风格的URI -->

<filter>

<filter-name>HiddenHttpMethodFilter</filter-name>

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

</filter>

<filter-mapping>

<filter-name>HiddenHttpMethodFilter</filter-name>

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

</filter-mapping>

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

</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。