1. **概要设计（10分）（大于300字）**

我设计了一个教务系统，后台使用SSM框架编写，前端页面使用当前主流的Bootstrap和jQuery框架完成页面信息展示功能。系统里面角色分为四个，分别是学生，教师与管理员，并且每个角色有相应的权限控制。例如访客只能访问首页，登陆与注册，但是管理员可以对它们的权限进行管理。管理员本身具有最高权力，对管理员的权限操作只能是对数据库表直接操作，前端不予显示。本系统开放注册功能，但新用户注册后必须由管理员进行审核操作。角色基本功能如下：

**学生基本功能如下**：

·首页

·选课系统

1. 已选课程
2. 全部课程

·我的课程

1. 我的课表
2. 我的成绩单

·个人资料

·注销

**教师基本功能如下**：

·首页

·选课系统

1. 选课情况

·我的课程

1. 我的课表
2. 录入成绩

·个人资料

·注销

**管理员基本功能如下**：

·首页

·选课系统

1. 开启入口
2. 课程信息
3. 关闭入口

·成绩系统

1. 开启入口
2. 关闭入口

·用户管理

1. 注册审核
2. 用户中心

·权限管理

1. 学生权限
2. 教师权限

·个人资料

·注销

1. **详细设计与实现（50分）（大于1000字或者10页以上，要有相应的文字说明）**

**系统架构设计：**

本系统根据功能的不同，项目结构可以划分为以下几个层次。

·持久对象层（也称持久层或持久化层）：该层由若干持久化类（实体类）组成。

·数据访问层（DAO层）：该层由若干DAO接口和 MyBatis 映射文件组成。接口的名称统一以Dao结尾，且 MyBatis 的映射文件名称要与接口的名称相同。

·业务逻辑层（ Service层）：该层由若干 Service接口和实现类组成。在本系统中，业务逻辑层的接口统一使用 Service结尾，其实现类名称统一在接口名后加Impl。该层主要用于实现系统的业务逻辑。

·Web表现层：该层主要包括 Spring MVC中的 Controller类和JSP页面。 Controller类主要负责拦截用户请求，并调用业务逻辑层中相应组件的业务逻辑方法来处理用户请求，然后将相应的结果返回给JSP页面。

**系统开发及运行环境**

GDP教务系统开发环境如下。

·操作系统： Windows

·Web服务器： Tomcat7.0

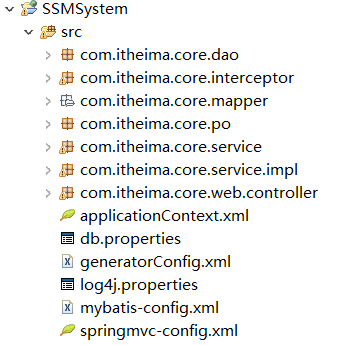
·Java开发包：JDK7

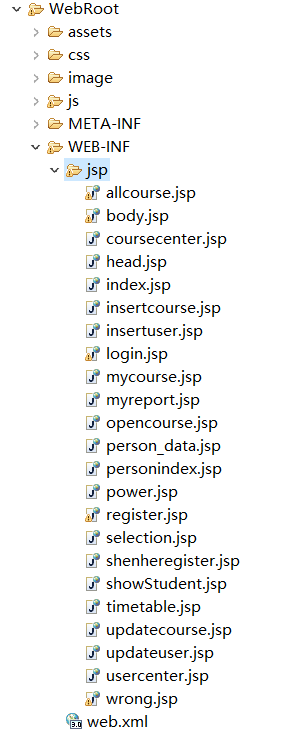
·开发工具： MyEclipse 10

·数据库： MYSQL5．5

·浏览器：谷歌浏览器

**文件组织结构：**

****



**系统环境具体JAR包（35个）如下：**

1. **spring框架所需的JAR包（10个）**  
   主要包括4个核心模块JAR，AOP开发使用的JAR，JDBC和事务的JAR。  
   ·aopalliance-1.0. jar  
   ·aspectjweaver-1.8.10. jar  
   ·spring-aop-4.3.6. RELEASE.jar  
   ·spring-aspects-4.3.6. RELEASE.jar  
   ·spring-beans-4.3.6. RELEASE.jar  
   ·spring-context-4.3.6.RELEASE.jar

·spring-core-4.3. 6. RELEASE.jar

·spring-expression-4.3.6. RELEASE. jar

·spring-jdbc-4.3.6. RELEASE.jar

·spring-tx-4.3.6. RELEASE. jar

**2． Spring MVC框架所需要的JAR包（2个）**

·spring-web-4.3.6. RELEASE.jar

·spring-webmvc-4.3.6. RELEASE. jar

**3.Mybatis框架所需的JAR包（13个）**

主要包括核心包 mybatis-3.4.2jar，以及其解压文件夹中lib目录下的所有JAR。

·ant-1.9.6. jar

·ant-launcher-1.9.6. jar

·asm-5. 1. jar

·cglib-3. 2. 4.jar

·commons-logging-1. 2. jar

·javassist-3. 21.o-GA.jar

·log4j-1.2.17.jar

·log4j-api-2. 3. jar

·log4j-core-2.3. jar

·mybatis-3.4.2. jar

·lognl-3. 1. 12. jar

·slf4i-apl-1. 7. 22.jar

·slf4i-log4j12-1. 7. 22. jar

**4.Mybatis与 Spring整合的中间JAR（1个）**

·mybatis-spring-1. 3. 1. jar

**5．数据库驱动JAR包（1个）**

·mysal-connector-java-5. 1. 40-bin .jar

**6．数据源dbcp所需JAR包（2个）**

·commons-dbcp2-2. 1. 1. jar

·commons-pool2-2. 4. 2. jar

**7．JSTL标签库JAR包（2个）**

·taglibs-standard-impl-1.2. 5. jar

·taglibs-standard-spec-1.2.5. jar

**8．Jackson框架所需JAR包（3个）**

·jackson-annotations-2.8.6. jar

·jackson-core-2.8.6. jar

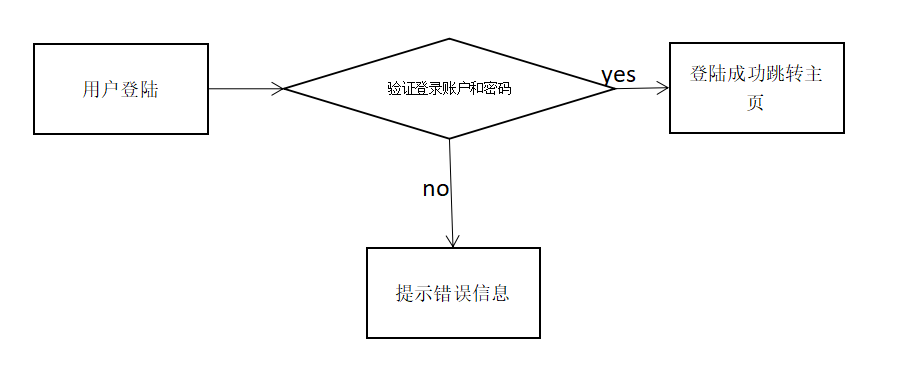
·jackson-databind-2.8.6.jar

1. **Java工具类JAR（1个）**

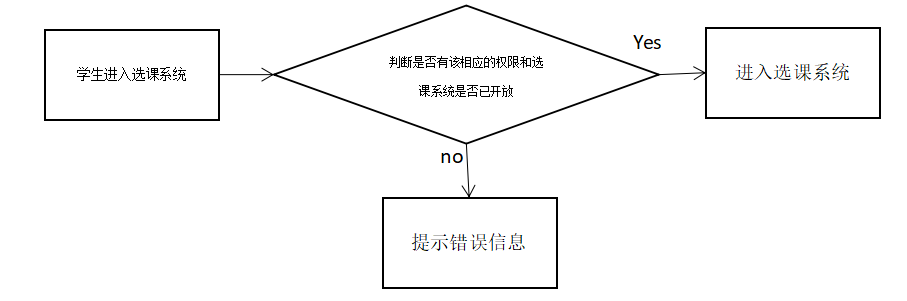
·commons-lang3-3. 4.jar

**流程图模块**

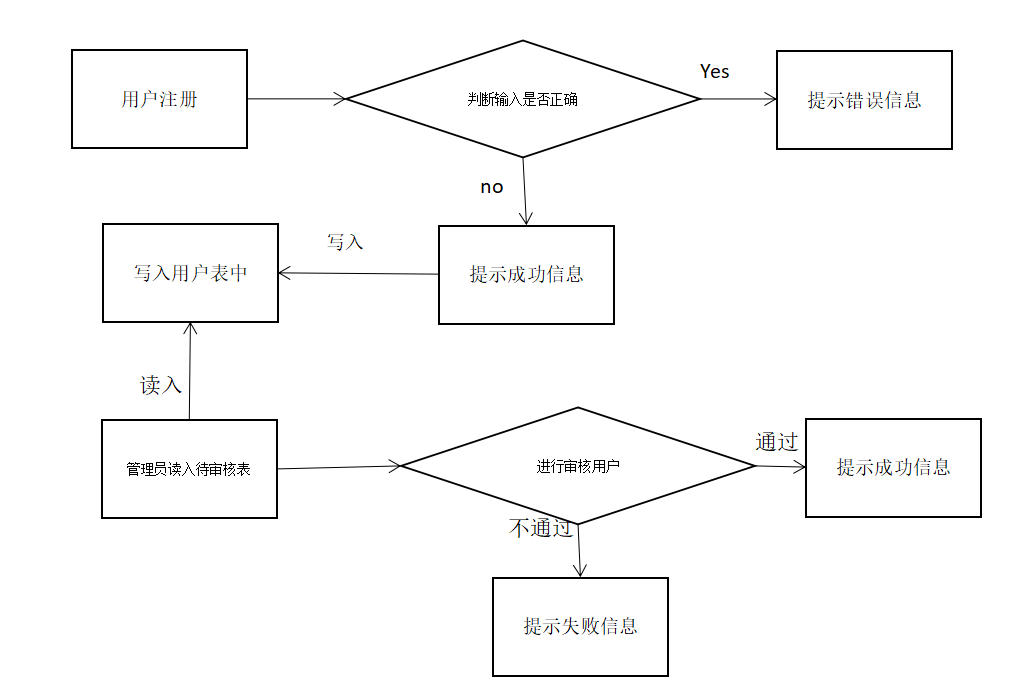
用户登录功能实现流程：



学生选课流程图：



用户注册与管理员审核流程图：



3.1 数据库表结构

数据库名称：SSMSystem

表名：Root\_info(用于存放管理员基本信息表)

User\_info(用于存放用户基本信息表)

Course\_info(用于存放课程基本信息表)

Course\_detail\_info(用于存放选课详细信息表)

Log\_info(用于存放管理员操作日志表)

Role\_info(用于存放角色权限菜单表)

Menu\_info（用于存放权限详细信息菜单表）

Root\_info

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 说明 | 备注 |
| User\_code | Int | 学号 | 主键 |
| User\_password | Varchar(200) | 密码 |  |
| User\_image | Varchar(200) | 头像路径 |  |
| User\_Type | Int | 类别 | 默认2：管理员 |
| User\_name | Varchar | 姓名 |  |
| User\_sex | Int | 性别 | 0：男  1：女 |

User\_info

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 说明 | 备注 |
| User\_code | Int | 学号 | 主键 |
| User\_password | Varchar(200) | 密码 |  |
| User\_Classroom | Varchar(200) | 班级 |  |
| User\_image | Varchar(200) | 头像路径 |  |
| User\_major | Varchar(200) | 专业（学院） |  |
| School\_code | Varchar(200) | 学校代码 |  |
| School\_name | Varchar(200) | 学校名称 |  |
| User\_Type | Int | 类别 | 0：学生  1：老师  2：管理员 |
| User\_name | Varchar | 姓名 |  |
| User\_sex | Int | 性别 | 0：男  1：女 |
| User\_status | Int | 状态 | 0：待审核  1：审核通过  2：审核不通过 |

Course\_info

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 说明 | 备注 |
| Co\_id | Int | 课程编号 | 主键，自增长 |
| Co\_code | Varchar(200) | 课程代码 |  |
| Co\_name | Varchar(200) | 课程名称 |  |
| Co\_status | Int | 课程状态 | 0：未开始  1：正在进行  2：选课结束  3：录入成绩  4：课程结束 |
| Co\_credit | Float | 学分 |  |
| Co\_number | Int | 人数 |  |
| Co\_time | Varchar(200) | 上课时间 |  |
| teacher\_code | Int | 老师编号 |  |
| Teacher\_name | Varchar | 老师姓名 |  |
| Co\_address | Varchar | 上课地点 |  |
| Co\_selected | Int | 已选人数 |  |

Course\_detail\_info

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 说明 | 备注 |
| De\_id | Int | 详细编号 | 主键，自增长 |
| Co\_id | Int | 课程编号 |  |
| Student\_code | Int | 学生编号 |  |
| teacher\_code | Int | 老师编号 |  |
| De\_score | Float | 成绩 | 默认-1 |
| de\_status | Int | 状态 | 0：正在选课  1：选课结束，正在进行  2：学期结束 |
| Student\_name | Varchar | 学生姓名 |  |
| Teacher\_name | Varchar | 老师姓名 |  |

Log\_info

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 说明 | 备注 |
| Log\_id | Int | 编号 | 主键，自增长 |
| Root\_id | Int | 管理员id |  |
| Log\_time | Varchar | 操作时间 |  |
| Log\_content | Varchar | 操作内容 |  |

Menu\_info

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 说明 | 备注 |
| M\_id | int | 菜单编号 | 主键，自增长 |
| M\_url | Varchar | 地址 |  |
| M\_title | Varchar | 菜单名称 |  |
| M\_status | Int | 状态 | 0：禁用  1：启用 |

Role\_info

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 说明 | 备注 |
| R\_id | int |  | 主键，自增长 |
| User\_type | Int | 角色类别 | -1:访客  0：学生  1：教师  2：管理员 |
| M\_id | int | 菜单编号 |  |
| r\_status | Int | 状态 | 0：禁用  1：启用 |

3.2 Mybatis的XML文件、相关文件

由于 Mybatis默认使用log4j输出日志信息，所以如果要查看控制台的输出SQL语句，那么就需要在 classpath路径下配置其日志文件。在项目的src目录下创建log4. properties文件，编辑后的内容如下所示。

**log4.properties**

|  |
| --- |
| # Global logging configuration log4j.rootLogger=ERROR, stdout  # MyBatis logging configuration...  log4j.logger.org.mybatis.example.BlogMapper=TRACE  # Console output...  log4j.appender.stdout=org.apache.log4j.ConsoleAppender  log4j.appender.stdout.layout=org.apache.log4j.PatternLayout  log4j.appender.stdout.layout.ConversionPattern=%5p [%t] - %m%n |

在src目录下，创建Mybatis的核心配置文件mybatis-config.xml，编辑后下所示。

**mybatis-config.xml**

|  |
| --- |
| <?xml version=*"1.0"* encoding=*"UTF-8"* ?>  <!DOCTYPE configuration  PUBLIC "-//mybatis.org//DTD Config 3.0//EN" "http://mybatis.org/dtd/mybatis-3-config.dtd">  <configuration>  <environments default=*"mysql"*>  <environment id=*"mysql"*>  <transactionManager type=*"JDBC"*/>  <dataSource type=*"POOLED"*>  <property name=*"driver"* value=*"com.mysql.jdbc.Driver"*/>  <property name=*"url"* value=*"jdbc:mysql://localhost:3306/ssmsystem?useUnicode=true&amp;characterEncoding=UTF-8"*/>  <property name=*"username"* value=*"root"*/>  <property name=*"password"* value=*""*/>  </dataSource>  </environment>  </environments>  <mappers>  <mapper resource=*"com/itheima/core/mapper/course\_detail\_infoMapper.xml"*/>  <mapper resource=*"com/itheima/core/mapper/course\_infoMapper.xml"*/>  <mapper resource=*"com/itheima/core/mapper/user\_infoMapper.xml"*/>  <mapper resource=*"com/itheima/core/mapper/log\_infoMapper.xml"*/>  <mapper resource=*"com/itheima/core/mapper/root\_infoMapper.xml"*/>  <mapper resource=*"com/itheima/core/mapper/role\_infoMapper.xml"*/>  <mapper resource=*"com/itheima/core/mapper/menu\_infoMapper.xml"*/>  </mappers>  </configuration> |

利用Mybatis Generator自动创建代码

**GeneratorConfig.xml**

|  |
| --- |
| <?xml version=*"1.0"* encoding=*"UTF-8"*?>  <!DOCTYPE generatorConfiguration  PUBLIC "-//mybatis.org//DTD MyBatis Generator Configuration 1.0//EN"  "http://mybatis.org/dtd/mybatis-generator-config\_1\_0.dtd">  <generatorConfiguration>  <classPathEntry location=*"E:/SSM/mysql-connector-java-5.1.25-bin.jar"*/>  <context id=*"DB2Tables"* targetRuntime=*"MyBatis3"*>  <commentGenerator>  <property name=*"suppressDate"* value=*"true"*/>  <property name=*"suppressAllComments"* value=*"true"*/>  </commentGenerator>  <jdbcConnection driverClass=*"com.mysql.jdbc.Driver"* connectionURL=*"jdbc:mysql://localhost/ssmsystem"* userId=*"root"* password=*""*>  </jdbcConnection>  <javaTypeResolver>  <property name=*"forceBigDecimals"* value=*"false"*/>  </javaTypeResolver>  <javaModelGenerator targetPackage=*"com.itheima.core.po"* targetProject=*"SSMSystem"*>  <property name=*"enableSubPackages"* value=*"true"*/>  <property name=*"trimStrings"* value=*"true"*/>  </javaModelGenerator>  <sqlMapGenerator targetPackage=*"com.itheima.core.mapper"* targetProject=*"SSMSystem"*>  <property name=*"enableSubPackages"* value=*"true"*/>  </sqlMapGenerator>  <javaClientGenerator type=*"XMLMAPPER"* targetPackage=*"com.itheima.core.dao"* targetProject=*"SSMSystem"*>  <property name=*"enableSubPackages"* value=*"true"*/>  </javaClientGenerator>  <table tableName=*"user\_info"* domainObjectName=*"user\_info"* enableCountByExample=*"false"* enableUpdateByExample=*"false"* enableDeleteByExample=*"false"* enableSelectByExample=*"false"* selectByExampleQueryId=*"false"*>  <generatedKey column=*"user\_id"* sqlStatement=*"mysql"* identity=*"true"*/>  </table>  <table tableName=*"course\_info"* domainObjectName=*"course\_info"* enableCountByExample=*"false"* enableUpdateByExample=*"false"* enableDeleteByExample=*"false"* enableSelectByExample=*"false"* selectByExampleQueryId=*"false"*>  <generatedKey column=*"co\_id"* sqlStatement=*"mysql"* identity=*"true"*/>  </table>  <table tableName=*"course\_detail\_info"* domainObjectName=*"course\_detail\_info"* enableCountByExample=*"false"* enableUpdateByExample=*"false"* enableDeleteByExample=*"false"* enableSelectByExample=*"false"* selectByExampleQueryId=*"false"*>  <generatedKey column=*"de\_id"* sqlStatement=*"mysql"* identity=*"true"*/>  </table>  <table tableName=*"log\_info"* domainObjectName=*"log\_info"* enableCountByExample=*"false"* enableUpdateByExample=*"false"* enableDeleteByExample=*"false"* enableSelectByExample=*"false"* selectByExampleQueryId=*"false"*>  <generatedKey column=*"log\_id"* sqlStatement=*"mysql"* identity=*"true"*/>  </table>  <table tableName=*"root\_info"* domainObjectName=*"root\_info"* enableCountByExample=*"false"* enableUpdateByExample=*"false"* enableDeleteByExample=*"false"* enableSelectByExample=*"false"* selectByExampleQueryId=*"false"*>  <generatedKey column=*"user\_code"* sqlStatement=*"mysql"* identity=*"true"*/>  </table>  <table tableName=*"role\_info"* domainObjectName=*"role\_info"* enableCountByExample=*"false"* enableUpdateByExample=*"false"* enableDeleteByExample=*"false"* enableSelectByExample=*"false"* selectByExampleQueryId=*"false"*>  <generatedKey column=*"r\_id"* sqlStatement=*"mysql"* identity=*"true"*/>  </table>  <table tableName=*"menu\_info"* domainObjectName=*"menu\_info"* enableCountByExample=*"false"* enableUpdateByExample=*"false"* enableDeleteByExample=*"false"* enableSelectByExample=*"false"* selectByExampleQueryId=*"false"*>  <generatedKey column=*"m\_id"* sqlStatement=*"mysql"* identity=*"true"*/>  </table>  </context>  </generatorConfiguration> |

3.3 SpringMVC部分

**1.配置前端控制器**

在web.xml中，配置Spring MVC 的前端控制器DispatcherServlet,文件如下：

**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"*>  <display-name></display-name>  <!-- 配置加载Spring文件的监听器 -->  <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>    <!-- 编码过滤器 -->  <filter>  <filter-name>CharacterEncodingFilter</filter-name>  <filter-class>  org.springframework.web.filter.CharacterEncodingFilter  </filter-class>  <async-supported>true</async-supported>  <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>\*.action</url-pattern>  </filter-mapping>  <!-- 配置Spring MVC 前端核心控制器 -->  <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-config.xml</param-value>  </init-param>  <load-on-startup>1</load-on-startup>  </servlet>  <servlet-mapping>  <servlet-name>springmvc</servlet-name>  <url-pattern>\*.action</url-pattern>  </servlet-mapping>  </web-app> |

1. **创建Spring MVC的配置文件，配置控制器映射文件**

在src目录下，创建配置文件springmvc-config.xml，并在文件中配置控制器信息，如下所示：

**springmvc-config.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:mvc=*"http://www.springframework.org/schema/mvc"*  xmlns:context=*"http://www.springframework.org/schema/context"*  xsi:schemaLocation=*"http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-4.3.xsd*  *http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc-4.3.xsd*  *http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-4.3.xsd"*>  <context:component-scan base-package=*"com.itheima.core.web.controller"*/>  <mvc:annotation-driven />    <mvc:resources location=*"/js/"* mapping=*"/js/\*\*"*/>  <mvc:resources location=*"/css/"* mapping=*"/css/\*\*"*/>  <mvc:resources location=*"/image/"* mapping=*"/image/\*\*"*/>  <mvc:resources location=*"/assets/"* mapping=*"/assets/\*\*"*/>  <bean class=*"org.springframework.web.servlet.view.InternalResourceViewResolver"*>  <property name=*"prefix"* value=*"/WEB-INF/jsp/"* />  <property name=*"suffix"* value=*".jsp"* />  </bean>  <!-- 配置拦截器 -->  <mvc:interceptors>  <mvc:interceptor>  <mvc:mapping path=*"/\*\*"*/>  <mvc:exclude-mapping path=*"/ssm\_index.action"*/>  <mvc:exclude-mapping path=*"/ssm\_login.action"*/>  <mvc:exclude-mapping path=*"/ssm\_checklogin.action"*/>  <mvc:exclude-mapping path=*"/ssm\_registerinto.action"*/>  <mvc:exclude-mapping path=*"/ssm\_register.action"*/>  <bean class=*"com.itheima.core.interceptor.AllInterceptor"*></bean>  </mvc:interceptor>  </mvc:interceptors>  <!-- 文件上传 -->  <bean id=*"multipartResolver"* class=*"org.springframework.web.multipart.commons.CommonsMultipartResolver"*>  <property name=*"defaultEncoding"* value=*"UTF-8"*/>  </bean>  </beans> |

上述代码配置了需要扫描的包，注解驱动和视图解析器外，还增加了加载属性文件和访问静态资源的配置。

1. **创建总拦截器类**

在src目录下，创建一个com.itheima.core.interceptor包，并在包中创建总拦截器类AllInterceptor,来实现不同角色的拦截功能。代码如下所示：

**AllInterceptor.java**

|  |
| --- |
| package com.itheima.core.interceptor;  import java.util.List;  import javax.servlet.http.\*;  import javax.servlet.http.\*;  import org.springframework.stereotype.\*;  import org.springframework.ui.\*;  import org.springframework.web.bind.annotation.\*;  import com.itheima.core.po.menu\_info;  import com.itheima.core.po.user\_info;  import org.springframework.web.servlet.\*;  public class AllInterceptor implements HandlerInterceptor {  @Override  public boolean preHandle(HttpServletRequest request, HttpServletResponse response,  Object handler) throws Exception {  String url = request.getRequestURI();  HttpSession session = request.getSession();  @SuppressWarnings("unchecked")  List<menu\_info> menu = (List<menu\_info>) session.getAttribute("menu");  if(menu != null){  for(int i=0;i<menu.size();i++){  if(url.indexOf(menu.get(i).getmUrl())>=0){  return true;  }  }  //System.out.println(menu.size());  request.getRequestDispatcher("/WEB-INF/jsp/login.jsp").forward(request, response);  return false;  }  request.getRequestDispatcher("/WEB-INF/jsp/login.jsp").forward(request, response);  return false;  }  @Override  public void postHandle(HttpServletRequest request, HttpServletResponse response,  Object handler, ModelAndView model) throws Exception {  }  @Override  public void afterCompletion(HttpServletRequest request, HttpServletResponse response,  Object handler, Exception ex)  throws Exception {  }  } |

1. **实现Service**
2. 在src目录下，创建一个com.itheima.core.service包，并在包中创建各个与DAO层相应的接口。列表如下所示：

|  |  |
| --- | --- |
| 文件名称 | 功能 |
| UserService.java | 用户Service接口 |
| RootService.java | 管理员Service接口 |
| RoleService.java | 角色表Service接口 |
| MenuService.java | 菜单表Service接口 |
| LogService.java | 管理员日志表Service接口 |
| DetailService.java | 选课情况Service接口 |
| CourseService.java | 课程表Service接口 |

部分代码展示如下：

**RootService.java**

|  |
| --- |
| package com.itheima.core.service;  import java.util.List;  import com.itheima.core.po.\*;  public interface RootService {  public user\_info selectByPrimaryKey(Integer userCode);  } |

1. 在src目录下，创建一个com.itheima.core.serviceImpl包，并在包中创建各个与Service层相应的接口的实现类。列表如下所示：

|  |  |
| --- | --- |
| 文件名称 | 功能 |
| UserServiceImpl.java | UserService的实现类 |
| RootServiceImpl.java | RootService的实现类 |
| RoleServiceImpl.java | RoleService的实现类 |
| MenuServiceImpl.java | MenuService的实现类 |
| LogServiceImpl.java | LogService的实现类 |
| DetailServiceImpl.java | DetailService的实现类 |
| CourseServiceImpl.java | CourseService的实现类 |

部分代码展示如下：

**RootServiceImpl.java**

|  |
| --- |
| package com.itheima.core.service.impl;  import java.util.List;  import org.springframework.beans.factory.annotation.\*;  import org.springframework.stereotype.\*;  import org.springframework.transaction.annotation.\*;  import com.itheima.core.dao.\*;  import com.itheima.core.po.\*;  import com.itheima.core.service.\*;  @Service("rootService")  @Transactional  public class RootServiceImpl implements RootService {  @Autowired  private root\_infoMapper root\_infoMapper;  @Override  public user\_info selectByPrimaryKey(Integer userCode) {  user\_info root\_info = this.root\_infoMapper.selectByPrimaryKey(userCode);  return root\_info;  }  } |

1. **实现Controller**

在src目录下，创建一个com.itheima.core.web.controller包，并在包中创建控制器类SSMController。

部分代码如下所示：

**SSMController.java**

|  |
| --- |
| package com.itheima.core.web.controller;  import java.io.File;  import java.io.UnsupportedEncodingException;  import java.text.SimpleDateFormat;  import java.util.ArrayList;  import java.util.Date;  import java.util.List;  import java.util.UUID;  import javax.servlet.http.HttpServletRequest;  import javax.servlet.http.HttpSession;  import org.springframework.beans.factory.annotation.Autowired;  import org.springframework.stereotype.\*;  import org.springframework.ui.\*;  import org.springframework.web.bind.annotation.\*;  import org.springframework.web.multipart.MultipartFile;  import com.itheima.core.po.\*;  import com.itheima.core.service.\*;  @Controller  public class SSMController {  @Autowired  private UserService userService;  @Autowired  private CourseService courseService;  @Autowired  private DetailService detailService;  @Autowired  private LogService logService;  @Autowired  private RootService rootService;  @Autowired  private RoleService roleService;  @Autowired  private MenuService menuService;  //进入主页请求  @RequestMapping(value="/ssm\_index.action")  public String ssm\_index(HttpServletRequest request,Model model){  HttpSession s=request.getSession();  user\_info user=(user\_info) s.getAttribute("user");  if(user != null){  if(user.getUserType() == 0){  model.addAttribute("username", user.getUserName() + " 同学");  }else if(user.getUserType() == 1){  model.addAttribute("username", user.getUserName() + " 老师");  }else if(user.getUserType() == 2){  model.addAttribute("username", user.getUserName() + " 管理员");  }    }else{  model.addAttribute("username", null);  }  return "index";  }  //进入登陆页面  @RequestMapping(value="/ssm\_login.action")  public String ssm\_login(HttpServletRequest request,Model model){  return "login";  }  //退出登录  @RequestMapping(value="/ssm\_exit.action")  public String ssm\_exit(HttpServletRequest request,Model model){  HttpSession s=request.getSession(true);  s.invalidate();  model.addAttribute("username", null);  return "redirect:/ssm\_index.action";  }  //进入注册请求  @RequestMapping(value="/ssm\_registerinto.action")  public String ssm\_registerinto(HttpServletRequest request,Model model){  return "register";  }  //注册  @RequestMapping(value="/ssm\_register.action")  public String ssm\_register(@RequestParam("name") String name,@RequestParam("uploadfile") List<MultipartFile> uploadfile,  @RequestParam("type") String type,@RequestParam("school\_name") String school\_name,@RequestParam("school\_code") String school\_code,  @RequestParam("major") String major,@RequestParam("classroom") String classroom,@RequestParam("usercode") String usercode,  @RequestParam("password") String password,@RequestParam("sex") String sex,HttpServletRequest request,Model model){  String sign = request.getParameter("sign");  String newFilename = null;  user\_info user = new user\_info();  user\_info A = this.userService.selectByPrimaryKey(Integer.parseInt(usercode), 1);  if(A == null){  if(!uploadfile.isEmpty() && uploadfile.size() > 0)  {  for(MultipartFile file : uploadfile)  {  String originalFilename = file.getOriginalFilename();  String dirpath = "E:\\javaee\\MyEclipse 10\\SSMSystem\\WebRoot\\image\\"+usercode;  File filepath = new File(dirpath);  if(!filepath.exists())  filepath.mkdirs();  newFilename = usercode + "\_" + UUID.randomUUID() + "\_" + originalFilename;  String path ="E:/javaee/MyEclipse 10/SSMSystem/WebRoot/image/"+usercode+"/"+newFilename;  File newFile=new File(path);  try  {  file.transferTo(newFile);  } catch (Exception e) {  e.printStackTrace();  }  }  }  String image = "image/" + usercode + "/" + newFilename;  user.setUserCode(Integer.parseInt(usercode));  user.setSchoolCode(school\_code);  user.setSchoolName(school\_name);  user.setUserClassroom(classroom);  user.setUserImage(image);  user.setUserMajor(major);  user.setUserName(name);  user.setUserPassword(password);  user.setUserSex(Integer.parseInt(sex));  if(sign.equals("1"))  user.setUserStatus(1);  else  user.setUserStatus(0);  user.setUserType(Integer.parseInt(type));  int flag = 0;  flag = userService.insert(user);  if(flag <= 0){  model.addAttribute("msg", "<script type='text/javascript'>alert('注册失败！')</script>");  return "register";  }  if(sign.equals("1")){  model.addAttribute("msg", "<script type='text/javascript'>alert('添加用户成功！')</script>");  return "insertuser";  }  model.addAttribute("msg", "<script type='text/javascript'>alert('注册成功，等待管理员审核！')</script>");  return "register";  }  model.addAttribute("msg", "<script type='text/javascript'>alert('已存在该用户，注册失败！')</script>");  return "register";  }  //处理权限启用或禁用  @RequestMapping(value="/root\_chulipower.action")  public String root\_chulipower(HttpServletRequest request,Model model){  HttpSession s=request.getSession();  int now = Integer.parseInt(request.getParameter("now"));  int rId = Integer.parseInt(request.getParameter("rId"));  int flag = this.roleService.changeRstatus(now, rId);  if(flag <= 0){  s.setAttribute("msg", "<script type='text/javascript'>alert('权限操作失败！')</script>");  return "redirect:/root\_power.action";  }  s.setAttribute("msg", "<script type='text/javascript'>alert('权限操作成功！')</script>");  return "redirect:/root\_power.action?sign="+s.getAttribute("power\_sign");  }  } |

3.4 Spring部分

1. 在src目录下，创建db.properties文件。代码如下所示：

**db.properties**

|  |
| --- |
| jdbc.driver=com.mysql.jdbc.Driver  jdbc.url=jdbc\:mysql\://localhost\:3306/ssmsystem?useUnicode\=true&characterEncoding\=UTF-8  jdbc.username=root  jdbc.password=  jdbc.maxTotal=30  jdbc.maxIdle=10  jdbc.initialSize=5 |

2. 在src目录下，创建Spring的配置文件，命名为applicationContext.xml。代码如下所示：

**applicationContext.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:aop=*"http://www.springframework.org/schema/aop"*  xmlns:tx=*"http://www.springframework.org/schema/tx"*  xmlns:context=*"http://www.springframework.org/schema/context"*  xsi:schemaLocation=*"http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-4.3.xsd*  *http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-4.3.xsd http://www.springframework.org/schema/context*  *http://www.springframework.org/schema/context/spring-context-4.3.xsd http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-4.3.xsd"*>  <!-- 读取db.properties -->  <context:property-placeholder location=*"classpath:db.properties"*/>    <!-- 配置数据源 -->  <bean id=*"dataSource"* class=*"org.apache.commons.dbcp2.BasicDataSource"*>  <property name=*"driverClassName"* value=*"${jdbc.driver}"*/>  <property name=*"url"* value=*"${jdbc.url}"*/>  <property name=*"username"* value=*"${jdbc.username}"*/>  <property name=*"password"* value=*"${jdbc.password}"*/>  <property name=*"maxTotal"* value=*"${jdbc.maxTotal}"*/>  <property name=*"maxIdle"* value=*"${jdbc.maxIdle}"*/>  <property name=*"initialSize"* value=*"${jdbc.initialSize}"*/>  </bean>    <!-- 事务管理器 -->  <bean id=*"transactionManager"* class=*"org.springframework.jdbc.datasource.DataSourceTransactionManager"*>  <property name=*"dataSource"* ref=*"dataSource"*/>  </bean>    <!-- 通知 -->  <tx:advice id=*"txAdvice"* transaction-manager=*"transactionManager"*>  <tx:attributes>  <!-- 传播行为 -->  <tx:method name=*"save\*"* propagation=*"REQUIRED"* />  <tx:method name=*"insert\*"* propagation=*"REQUIRED"* />  <tx:method name=*"add\*"* propagation=*"REQUIRED"* />  <tx:method name=*"create\*"* propagation=*"REQUIRED"* />  <tx:method name=*"update\*"* propagation=*"REQUIRED"* />  <tx:method name=*"delete\*"* propagation=*"REQUIRED"* />  <tx:method name=*"find\*"* propagation=*"SUPPORTS"* read-only=*"true"* />  <tx:method name=*"get\*"* propagation=*"SUPPORTS"* read-only=*"true"* />  <tx:method name=*"select\*"* propagation=*"SUPPORTS"* read-only=*"true"* />  </tx:attributes>  </tx:advice>    <!-- 切面 -->  <aop:config>  <aop:advisor advice-ref=*"txAdvice"* pointcut=*"execution(\* com.itheima.core.service.\*.\*(..))"*/>  </aop:config>    <!-- 配置Mybatis的工厂 -->  <bean class=*"org.mybatis.spring.SqlSessionFactoryBean"*>  <property name=*"dataSource"* ref=*"dataSource"*/>  <property name=*"configLocation"* value=*"classpath:mybatis-config.xml"*/>  </bean>    <!-- 接口开发，扫描com.itheima.core.dao包，写在此包下的接口即可被扫描到 -->  <bean class=*"org.mybatis.spring.mapper.MapperScannerConfigurer"*>  <property name=*"basePackage"* value=*"com.itheima.core.dao"*/>  </bean>  <!-- 配置扫描@Service注解 -->  <context:component-scan base-package=*"com.itheima.core.service"*/>  </beans> |

**3.实现持久化类**

在src目录下，创建一个com.itheima.core.po包，该包用于存放持久化类。列表如下所示：

|  |  |
| --- | --- |
| 文件名称 | 功能 |
| user\_info.java | 用户持久化类 |
| root\_info.java | 管理员持久化类 |
| role\_info.java | 角色表持久化类 |
| Menu\_info.java | 菜单表持久化类 |
| Log\_info.java | 管理员日志表持久化类 |
| Course\_info.java | 选课情况持久化类 |
| Course\_detail\_info.java | 课程表持久化类 |

部分代码展示如下：

**root\_info.java**

|  |
| --- |
| **package** com.itheima.core.po;  **public** **class** root\_info {  **private** Integer userCode;  **private** String userPassword;  **private** String userImage;  **private** Integer userType;  **private** String userName;  **private** Integer userSex;  **public** Integer getUserCode() {  **return** userCode;  }  **public** **void** setUserCode(Integer userCode) {  **this**.userCode = userCode;  }  **public** String getUserPassword() {  **return** userPassword;  }  **public** **void** setUserPassword(String userPassword) {  **this**.userPassword = userPassword == **null** ? **null** : userPassword.trim();  }  **public** String getUserImage() {  **return** userImage;  }  **public** **void** setUserImage(String userImage) {  **this**.userImage = userImage == **null** ? **null** : userImage.trim();  }  **public** Integer getUserType() {  **return** userType;  }  **public** **void** setUserType(Integer userType) {  **this**.userType = userType;  }  **public** String getUserName() {  **return** userName;  }  **public** **void** setUserName(String userName) {  **this**.userName = userName == **null** ? **null** : userName.trim();  }  **public** Integer getUserSex() {  **return** userSex;  }  **public** **void** setUserSex(Integer userSex) {  **this**.userSex = userSex;  }  } |

**4.实现DAO层**

在src目录下，创建一个com.itheima.core.dao包，该包用于存放用户接口。列表如下所示：

|  |  |
| --- | --- |
| 文件名称 | 功能 |
| user\_infoMapper.java | 用户接口DAO |
| root\_infoMapper.java | 管理员接口DAO |
| role\_infoMapper.java | 角色表接口DAO |
| Menu\_infoMapper.java | 菜单表接口DAO |
| Log\_infoMapper.java | 管理员日志表接口DAO |
| Course\_infoMapper.java | 选课情况接口DAO |
| Course\_detail\_infoMapper.java | 课程表接口DAO |

部分代码展示如下：

**root\_infoMapper.java**

|  |
| --- |
| **package** com.itheima.core.dao;  **import** com.itheima.core.po.\*;  **public** **interface** root\_infoMapper {  **int** deleteByPrimaryKey(Integer userCode);  **int** insert(user\_info record);  **int** insertSelective(user\_info record);  user\_info selectByPrimaryKey(Integer userCode);  **int** updateByPrimaryKeySelective(user\_info record);  **int** updateByPrimaryKey(user\_info record);  } |

**5.实现Mapper层**

在src目录下，创建一个com.itheima.core.mapper包，该包用于存放Mybatis映射文件。列表如下所示：

|  |  |
| --- | --- |
| 文件名称 | 功能 |
| user\_infoMapper.xml | 用户表映射文件 |
| root\_infoMapper.xml | 管理员表映射文件 |
| role\_infoMapper.xml | 角色表映射文件 |
| Menu\_infoMapper.xml | 菜单表映射文件 |
| Log\_infoMapper.xml | 管理员日志表映射文件 |
| Course\_infoMapper.xml | 选课情况表映射文件 |
| Course\_detail\_infoMapper.xml | 课程表映射文件 |

部分代码展示如下：

**root\_infoMapper.xml**

|  |
| --- |
| <?xml version=*"1.0"* encoding=*"UTF-8"* ?>  <!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd" >  <mapper namespace=*"com.itheima.core.dao.root\_infoMapper"* >  <resultMap id=*"BaseResultMap"* type=*"com.itheima.core.po.user\_info"* >  <id column=*"user\_code"* property=*"userCode"* jdbcType=*"INTEGER"* />  <result column=*"user\_password"* property=*"userPassword"* jdbcType=*"VARCHAR"* />  <result column=*"user\_image"* property=*"userImage"* jdbcType=*"VARCHAR"* />  <result column=*"user\_type"* property=*"userType"* jdbcType=*"INTEGER"* />  <result column=*"user\_name"* property=*"userName"* jdbcType=*"VARCHAR"* />  <result column=*"user\_sex"* property=*"userSex"* jdbcType=*"INTEGER"* />  </resultMap>  <sql id=*"Base\_Column\_List"* >  user\_code, user\_password, user\_image, user\_type, user\_name, user\_sex  </sql>  <select id=*"selectByPrimaryKey"* resultMap=*"BaseResultMap"* parameterType=*"java.lang.Integer"* >  select  <include refid=*"Base\_Column\_List"* />  from root\_info  where user\_code = #{userCode,jdbcType=INTEGER}  </select>  </mapper> |

3.5 前端页面部分

在webRoot/WEB-INF里创建一个名为jsp的文件夹，用于存放前端jsp文件。

导航栏文件：

|  |  |
| --- | --- |
| 文件名称 | 文件展示的功能 |
| Head.jsp | 首页的导航栏 |
| body.jsp | 教务中心的导航栏 |

公共页面：

|  |  |
| --- | --- |
| 文件名称 | 文件展示的功能 |
| index.jsp | 首页 |
| login.jsp | 登录页面 |
| register.jsp | 注册页面 |
| wrong.jsp | 错误页面 |

学生页面：

|  |  |
| --- | --- |
| 文件名称 | 文件展示的功能 |
| Allcourse.jsp | 所有选课列表 |
| Mycourse.jsp | 已选课程 |
| Myreport.jsp | 我的成绩单 |
| Person\_data.jsp | 个人信息 |
| Personindex.jsp | 教务中心 |
| Timetable.jsp | 我的课表 |

教师页面：

|  |  |
| --- | --- |
| 文件名称 | 文件展示的功能 |
| Selection.jsp | 课程选课情况 |
| ShowStudent.jsp | 展示我的学生 |
| Person\_data.jsp | 个人信息 |
| Personindex.jsp | 教务中心 |
| Timetable.jsp | 我的课表 |

管理员页面：

|  |  |
| --- | --- |
| 文件名称 | 文件展示的功能 |
| Person\_data.jsp | 个人信息 |
| Personindex.jsp | 教务中心 |
| Courseconter.jsp | 课程信息 |
| Insertcourse.jsp | 添加课程 |
| insertuser.jsp | 添加用户 |
| Opencourse.jsp | 开启选课 |
| Power.jsp | 权限管理 |
| Shenheregister.jsp | 审核注册 |
| Updatecourse.jsp | 更新课程信息 |
| Updateuser.jsp | 更新用户信息 |
| Usercenter.jsp | 用户管理 |

因页面过多，只展示部分页面代码。

Head.jsp

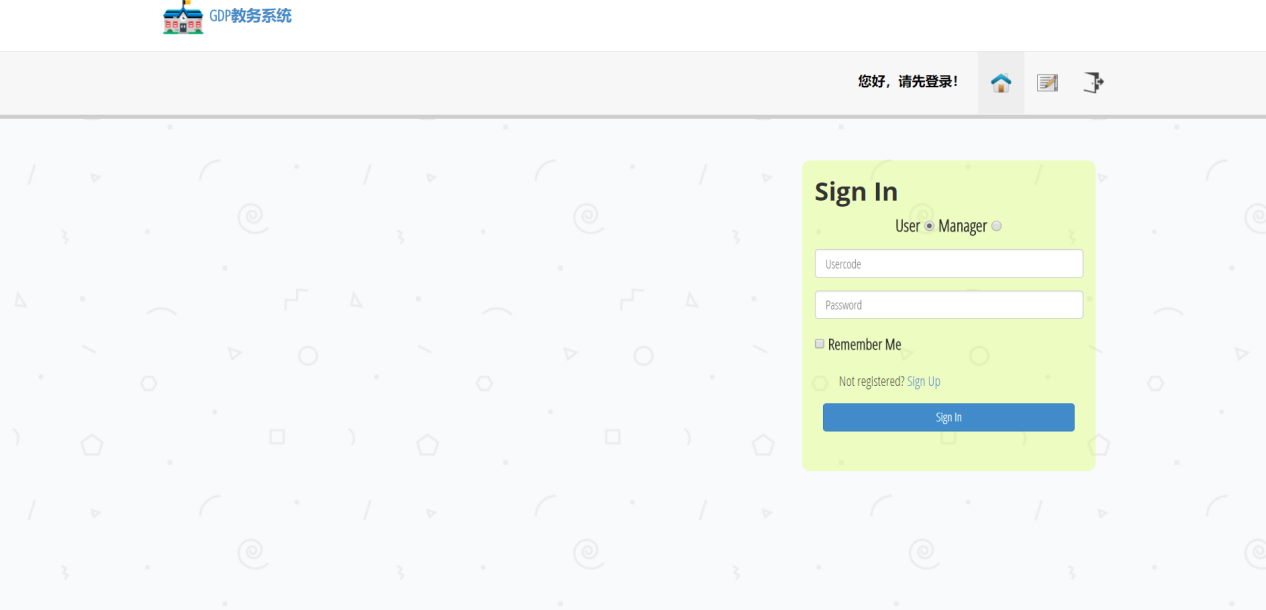
|  |
| --- |
| <%@ page language=*"java"* import=*"java.util.\*"* pageEncoding=*"UTF-8"*%>  <%@ taglib uri=*"http://java.sun.com/jsp/jstl/core"* prefix=*"c"* %>  <!DOCTYPE html>  <html xmlns=*"http://www.w3.org/1999/xhtml"*>  <head>  <meta charset=*"utf-8"* />  <meta name=*"viewport"* content=*"width=device-width, initial-scale=1, maximum-scale=1"* />  <meta name=*"description"* content=*""* />  <meta name=*"author"* content=*""* />  <style type=*"text/css"*>  @import *url("css/bootstrap.css")*;  @import *url("css/font-awesome.css")*;  @import *url("css/animate.css")*;  @import *url("css/flexslider.css")*;  @import *url("css/style1.css")*;  </style>  <link href=*'http://fonts.googleapis.com/css?family=Open+Sans+Condensed:300'* rel=*'stylesheet'* type=*'text/css'* />  <link href=*'http://fonts.googleapis.com/css?family=Open+Sans'* rel=*'stylesheet'* type=*'text/css'* />  <link href=*'http://fonts.googleapis.com/css?family=Lobster'* rel=*'stylesheet'* type=*'text/css'* />  </head>  <body>  <input type=*"hidden"* name=*"name"*>  <div class=*"navbar navbar-inverse set-radius-zero"* >  <div class=*"container"*>  <div class=*"navbar-header"*>  <a href=*"ssm\_index.action"*>  <img src=*"image/head\_school.png"* width=*"55"* height=*"55"*/>  <font size=*"4"*><b>GDP教务系统</b></font>  </a>  </div>  <div class=*"right-div"*>  </div>  </div>  </div>  <section class=*"menu-section"*>  <div class=*"container"*>  <div class=*"row "*>  <div class=*"col-md-12"*>  <div class=*"navbar-collapse collapse "*>  <ul id=*"menu-top"* class=*"nav navbar-nav navbar-right"*>  <c:if test=*"*${username==null }*"*>  <li><a href=*"ssm\_login.action"*>您好，请先登录！</a></li>  </c:if>  <c:if test=*"*${username!=null }*"*>  <li><a href=*"ssm\_personindex.action"*>您好， ${username}</a></li>  </c:if>  <li><a href=*"ssm\_index.action"* class=*"menu-top-active"* ><img src=*"image/index.png"* height=*"25"* width=*"25"*></a></li>  <li><a href=*"ssm\_registerinto.action"* ><img src=*"image/register.png"* height=*"25"* width=*"25"*></a></li>  <li><a href=*"ssm\_exit.action"*><img src=*"image/logoff.png"* height=*"25"* width=*"25"*></a></li>  </ul>  </div>  </div>  </div>  </div>  </section>  </body>  </html> |

1. **成果展示（20分）（大于300字，可图片展示）**

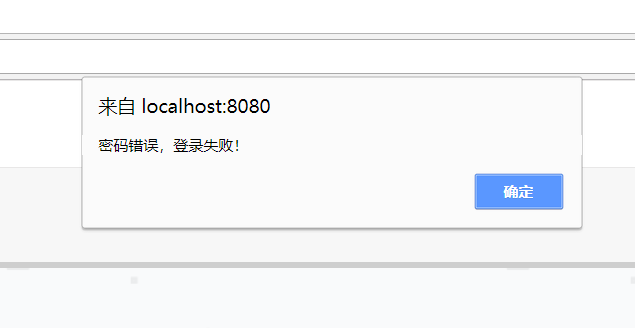
首先进入的是主页：



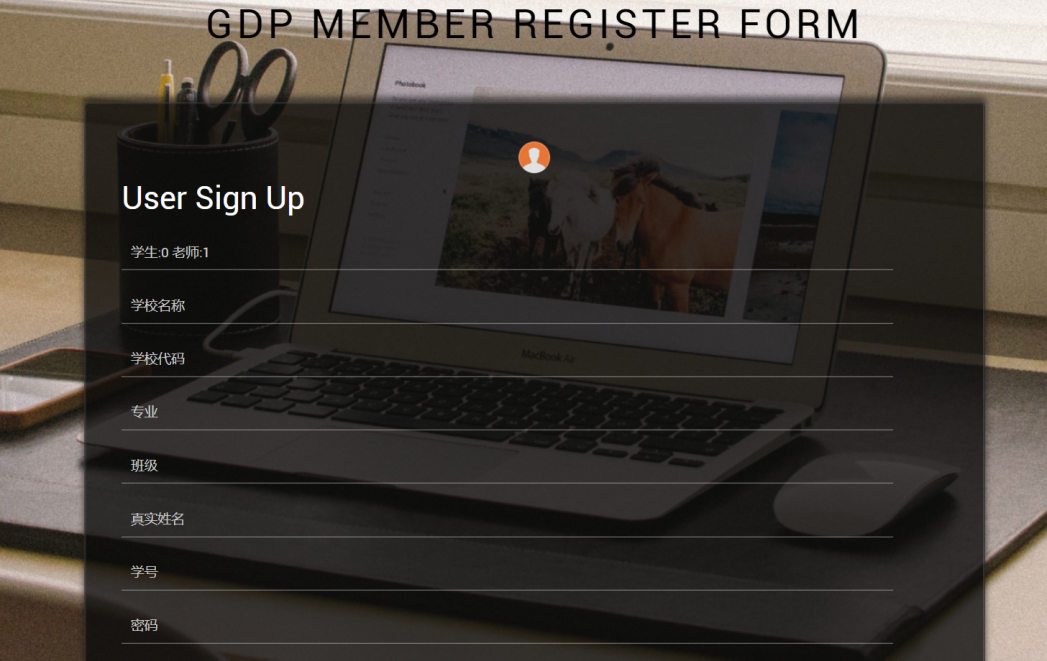
登录页面：



如果登录失败，提示相应的信息：



注册页面：



管理员教务中心



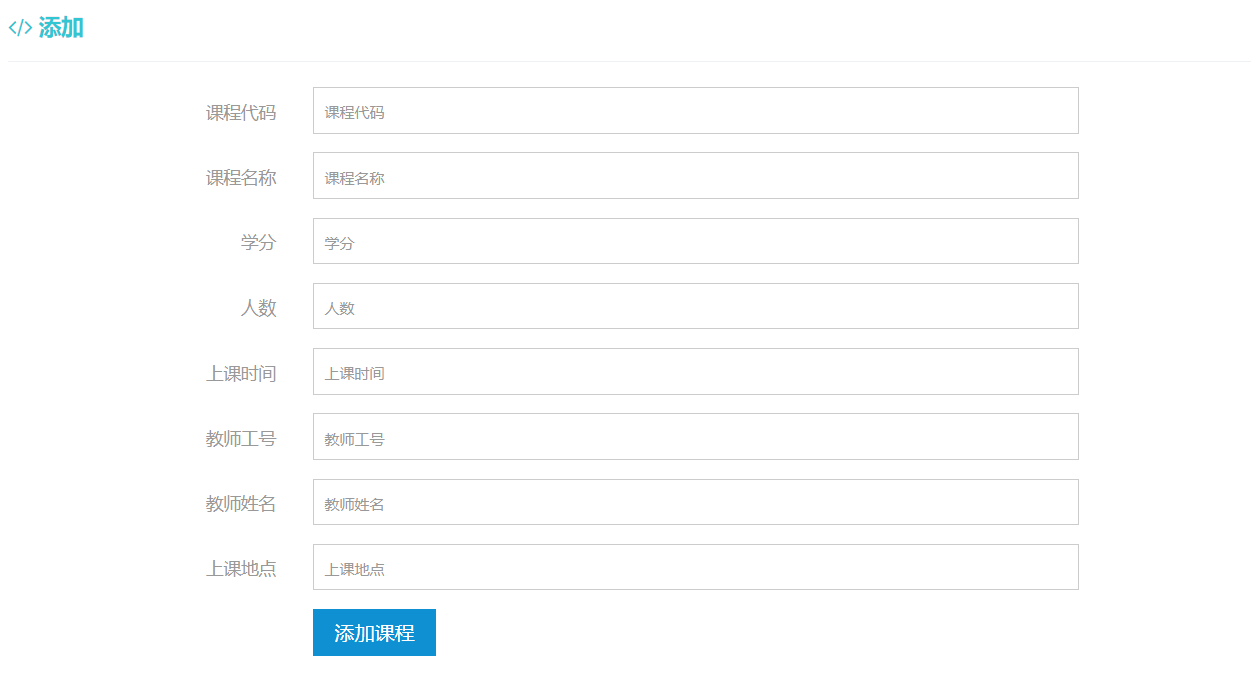
开启选课：



课程信息列表：



添加课程：



编辑课程：



注册审核：



用户中心：



权限管理：



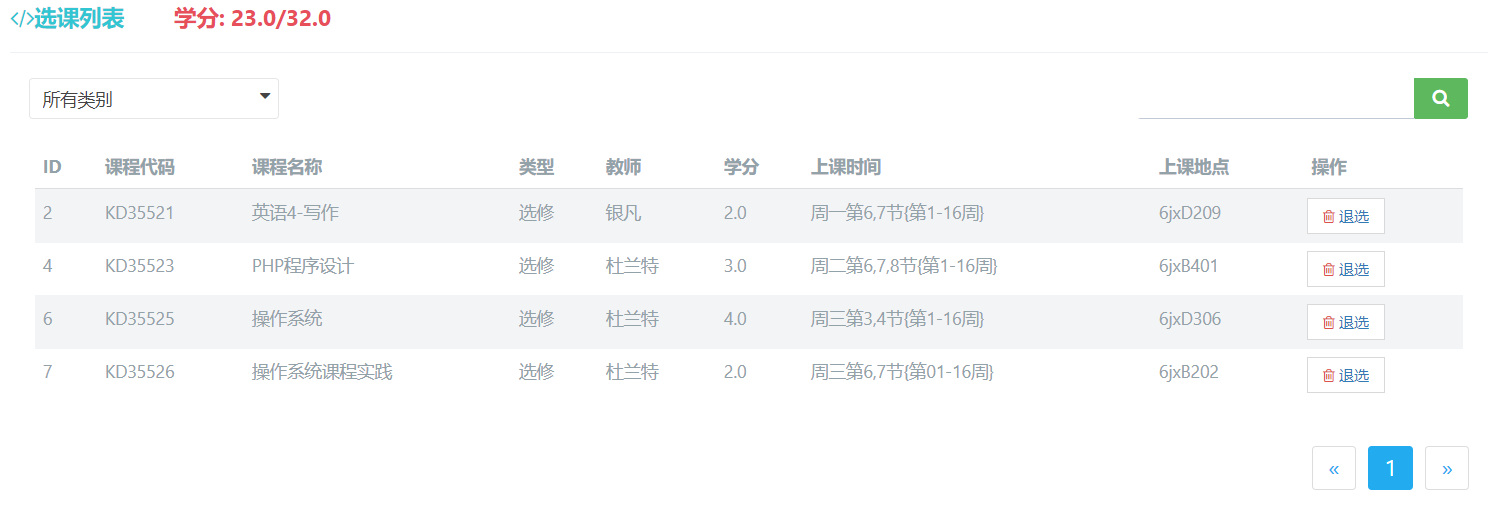
个人资料：



学生查询后的选课列表：



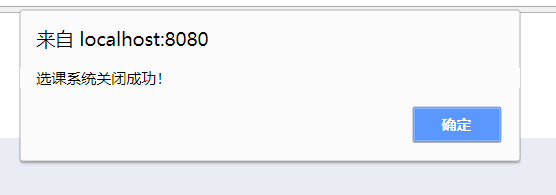
已选课程：



老师查看选课学生名单：



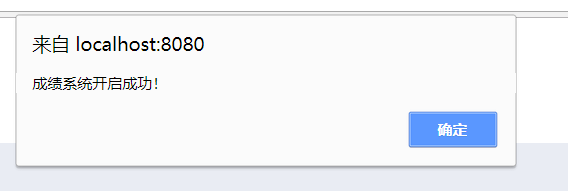
关闭选课入口：



我的课表：



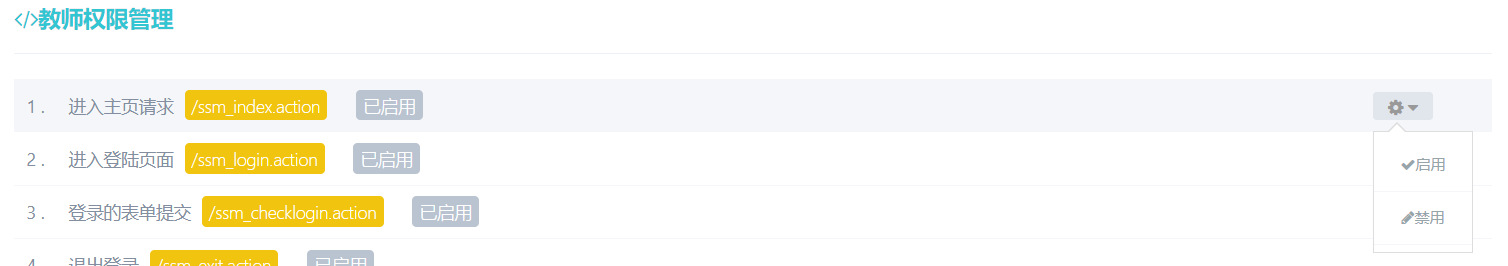
录入成绩系统开启：



老师进行录入成绩操作：



管理员进行权限操作：



我的成绩单：

