## 学生选课系统

**要求具备如下知识：**

1. Java基础
2. 前端基础（Html、css、JavaScript、Jquery）
3. mysql数据库
4. Web容器（Tomcat）
5. Jsp/Servlet

**通过该项目可以掌握哪些知识：**

1. 掌握整个项目的流程
2. 掌握开发中的简单权限控制
3. 掌握前端页面如何与后端整合
4. 宏观把握一个项目
5. 掌握echarts图表技术

# 技术选型

Web容器：Apache Tomcat8.5

编程语言：Java1.8

数据库连接池：DBCP

数据库操作：DBUtils

动态网页：JSP/Servlet3.1

静态网页：HTML+CSS+Jquery

数据库：MySQL5.7

操作系统：Window10

# 需求描述

**角色**：学生、老师、管理员

**2.1 学生：基本信息、课程信息**

1. 学生操作流程：
2. 学生登录系统
3. 选择课程并提交确认（确认之后则不能修改了）
4. 考试系统(省略)
5. 各种查询（查询各科成绩、查询总成绩、查询排名等）
6. 学生查询功能
   1. 各科成绩查询
   2. 总成绩查询
   3. 成绩排名、各科成绩排名

**2.2 老师：基本信息、所教课程**

1. 老师操作流程：
2. 登录系统
3. 课程管理（实现对课程的CURD）
4. 考试成绩录入
5. 各种查询（学生的单科成绩、总成绩、排名）
6. 各种统计（统计及格率、统计>80分以上的同学等）

**2.3 管理员：维护学生和老师信息、查询统计**

1. 管理员操作流程：
   1. 登录系统
   2. 实现对学生的CURD
   3. 实现对老师的CURD
   4. 实现各种查询、统计

学生

登录

成功

失败

选课

评分

查询成绩

# 数据库设计

详情查看pdm图

# 项目搭建

1. 创建项目（dynamic Web project）sct
2. 建包
   1. com.hua.entity(model\vo)：student实体、课程实体等
   2. com.hua.dao（Data Access Object 数据访问对象）操作数据库
   3. com.hua.utils:工具包
   4. com.hua.servlet(web\action\controller)：存放servlet项目的文件
   5. com.hua.service(biz)：存放业务逻辑的

举例：甲🡪乙转账业务:甲转出（JDao.out()调用）🡪乙转入（YDao.in()调用），他们之间要有事务。

1. 编码

配置数据库连接文件db.properties

|  |
| --- |
| driverClassName=com.mysql.jdbc.Driver  url=jdbc:mysql://127.0.0.1:3306/sct?useUnicode=true&characterEncoding=UTF-8  username=root  password=123456 |

加载db.properties配置文件

建立数据库连接，返回数据源

# 项目界面

1. 登录页面login.html
2. 首页：index.html
3. 列表页面：list.html
4. 新增或修改页面：add.html/edit.html

# 项目界面与代码整合

详解视频

# 登录

管理员：

老师：

学生：

1、假设：后端设计数据库中账号字段的时候，有规律可循：

管理员：Admin password:123456(数据库md5加密过)

老师：1001开头 1001001、1001002

学生：2001开头 2001001、2001002

2、假设：后端设计数据库中账号字段没有规律，此时需要添加一个登录类型：

# 登录权限过滤

PermissionFilter可以过滤是否登录的权限！

# 权限的功能设置

**管理员：**

1. 学生管理：CURD
2. 老师管理：CURD、（老师课程设置）
3. 课程管理：CURD
4. 统计查询：（及格率、每个科目等等统计）

**老师：**

1. 登录
2. 录入分数
   1. 先查询自己所教的课程(管理员已经设置过了)
   2. 查询所教课程下的学生（学生角色选课的过程，学生登录后做选课）
   3. 最后录入评分,提交即可（老师角色评分即可）
3. 查询统计（统计及格率、所教科目统计等等）

**学生：**

1. 登录
   1. 登录操作
2. 选课
   1. 查询所有的课程
   2. 选课其实就是选择老师（因为课程关联老师）
3. 考试（考试系统…），老师录入分数之后。（6b步骤）
4. 查询考试成绩、查询排名等

# 学生管理

1. 列表查询
2. 新增
3. 修改

修改分两步：1）先查询，跳转到待修改页面，2）修改信息，然后提交完成

1. 组合条件查询
2. 分页

解决乱码问题采用过滤器

# 分页技术实现

分页：数据库分页查询语句+掌握jsp+java等综合技术

**常见第一种**



**常见第二种**



**常用分页技术：**

内存分页:一次性将数据查询出来放入内存，然后进行分页，每页的数据都是在内存中存储，所以查询效率高。适用于小数据量，要求查询效率较高。

数据库分页：每次从数据库中查询出来一次的数据，然后放入内存，如果点击下一页，则重新从数据库查询下一页数据放入内存，上一页内存数据就被GC回收。适用于数据量较大的，查询效率较低。

**分页技术点**

****

页面大小(PageSize)：已知设置的10；

总记录条数（TotalCount）数据库查询而来；

总页数(totalPage)：TotalCount /pageSize = 85;

如果：totalCount%pageSize!=0 +1

如果：totalCount%pageSize==0 +0

页码数（PageNo）：默认为1，

下一页 pageNo+1（尾页的时候，不加1了），

上一页pageNo-1（首页的时候不减1）

尾页：pageNo = totalPage

首页：pageNo = 1

# 组合条件查询

组合条件：一个条件、两个条件、三个及其以上

输入几个条件，按照条件查询结果（满足查询的条件与）。

组合条件查询出来的结果是&&的关系

|  |
| --- |
| -- 按照ID查询  select \* from student where stuId = 1  -- 当前端录入学生姓名的时候  select \* from student where stuName like '%2%'  -- 当前端录入姓名与学号的时候 ，执行的如下SQL  select \* from student where stuName like '%2%' and stuNO like '%1%'  -- 通常情况下：字符串都按照like 模糊查询较多，数字：（=或者范围居多） |

# 老师管理

1. 列表查询
2. 新增
3. 修改

修改分两步：1）先查询，跳转到待修改页面，2）修改信息，然后提交完成

1. 组合条件查询
2. 分页

# 课程管理

1. 列表查询
2. 新增
3. 修改

修改分两步：1）先查询，跳转到待修改页面，2）修改信息，然后提交完成

1. 组合条件查询
2. 分页

**查询条件：课程ID、课程名称、老师姓名**

**列表中的数据来源于course表与teacher表**

|  |  |  |
| --- | --- | --- |
| 课程ID | 课程名称 | 老师姓名 |
| 1 | Java编程 | 张平1 |
| 2 | HTML编程 | 张平1 |

课程与老师的关系

N:1

主外键关系：

主表（Teacher 1）、从表（Course：N）

# 学生角色

1. 登录
2. 学生选课

# 老师角色

1. 登录
2. 评分
   1. 先查询自己所教的课程(管理员已经设置过了)
   2. 查询所教课程下的学生（学生角色选课的过程，学生登录后做选课）
   3. 最后录入评分,提交即可（老师角色评分即可）

# 查询统计

1. 数据录入
2. 查询统计
   1. **管理员-成绩区间统计**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 科目 | <60） | [60-70] | (70-85] | (85-100] |
| Java | 1 | 10 | 5 | 6 |
| Html | 1 | 10 | 5 | 6 |
| Javascript | 1 | 10 | 5 | 6 |

* 1. **管理员-及格率**

|  |  |  |
| --- | --- | --- |
| 科目 | 及格率 | 参加考试人数 |
| Java | 72% | 10 |
| Html | 75% | 10 |
| Javascript | 84% | 10 |

* 1. **老师-成绩区间统计(所教科目)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 科目 | <60 | 60-70 | 70-85 | 85-100 |
| Java | 1 | 10 | 5 | 6 |

* 1. **老师-及格率**

|  |  |  |
| --- | --- | --- |
| 科目 | 及格率 | 参加考试人数 |
| Java | 72% | 10 |

* 1. **老师/学生-排名（共有查询）**
  2. **学生-自己分数**