# **学生管理系统详细设计**

## **一、引言**

本详细设计是在学生管理系统概要设计的基础上，对系统的原型图、接口设计以及数据库表设计进行更深入、细致的规划和描述，为系统的开发提供具体的技术指导和实现方案，确保系统能够满足用户的功能需求、性能要求以及易用性标准。

## **二、原型图设计**

### **（一）登录页面**

****布局****：页面简洁，顶部为系统名称 “学生管理系统”，中间是用户名和密码输入框，下方有 “登录” 和 “忘记密码” 按钮，底部显示版权信息和技术支持联系方式。

****交互****：当用户输入用户名和密码后，点击 “登录” 按钮，系统进行验证，若验证通过则跳转到系统主页面；若点击 “忘记密码”，则弹出密码重置页面，用户可通过邮箱或手机验证码重置密码。

### **（二）系统主页面（管理员视角）**

****布局****：采用侧边栏导航菜单，包括 “学生信息管理”、“课程管理”、“成绩管理”、“考勤管理”、“用户管理” 等模块。页面主体区域根据所选菜单模块显示相应的信息列表或操作界面，如在 “学生信息管理” 模块下，展示学生信息表格，包含学号、姓名、性别、专业、班级等字段，表格上方有搜索框、添加按钮和导出按钮。

****交互****：点击侧边栏菜单选项，主体区域内容动态切换；在学生信息表格中，点击某一行可查看该学生的详细信息，点击 “添加” 按钮弹出学生信息录入弹窗，输入信息后点击 “保存” 可将新学生信息存入数据库，点击 “导出” 按钮可将学生信息表格导出为 Excel 文件。

### **（三）学生信息录入页面**

****布局****：分为基本信息、学籍信息、联系方式等几个部分，每个部分使用表单形式布局，如基本信息部分包含姓名、性别、出生日期、民族等字段，每个字段前有清晰的标签，字段后有必要的提示信息（如日期格式要求），页面底部有 “保存” 和 “取消” 按钮。

****交互****：用户在各字段输入相应信息，系统实时进行格式校验，如身份证号位数校验、日期格式校验等，当用户点击 “保存” 按钮时，系统将信息发送到后台进行数据存储，若数据合法且存储成功，则弹出提示框告知用户 “学生信息保存成功”，并自动清空表单，方便下一次录入；若点击 “取消”，则关闭当前页面，不保存任何数据。

### **（四）课程管理页面（教师视角）**

****布局****：上方为课程搜索框和 “添加课程” 按钮，下方是课程列表表格，包含课程编号、课程名称、学分、授课教师、课程类型等字段，表格右侧有 “编辑” 和 “删除” 按钮，对应每行课程信息。

****交互****：教师在搜索框输入课程相关信息可快速查找特定课程；点击 “添加课程” 按钮弹出课程信息录入弹窗，录入完成后点击 “保存” 将课程信息存入数据库；在课程列表中，点击 “编辑” 按钮可对课程信息进行修改，点击 “删除” 按钮则弹出确认框，确认后删除该课程信息，并刷新课程列表。

### **（五）成绩录入页面（教师视角）**

****布局****：页面顶部显示当前课程名称和授课班级，中间是成绩录入表格，每行对应一个学生，列分别为学号、姓名、平时成绩、期中成绩、期末成绩、总评成绩，总评成绩列自动根据预设权重计算（如平时成绩占 30%、期中成绩占 30%、期末成绩占 40%），不可手动编辑，表格下方有 “保存成绩” 和 “取消” 按钮。

****交互****：教师在平时成绩、期中成绩、期末成绩列输入相应分数后，总评成绩自动更新；输入过程中，系统对成绩进行范围校验（如 0 - 100 分），超出范围则给出提示；全部成绩录入完成后，点击 “保存成绩” 按钮，系统将成绩数据发送到后台进行存储，若存储成功则提示 “成绩录入成功”，否则提示错误信息；点击 “取消” 按钮则放弃本次成绩录入操作，关闭页面。

## **三、接口设计**

### **（一）学生信息管理接口**

****获取学生信息列表接口（GET /api/students）****：

****请求参数****：可选参数包括学号（studentId）、姓名（name）、班级（class）等，用于筛选特定学生信息。

****响应数据****：返回一个 JSON 数组，包含符合条件的学生信息对象，每个对象包括学号、姓名、性别、出生日期、专业、班级等字段。

****示例****：

收起

json

[

{

"studentId": "2021001",

"name": "张三",

"gender": "男",

"birthDate": "2003-05-10",

"major": "计算机科学与技术",

"class": "2021 级 1 班"

},

{

"studentId": "2021002",

"name": "李四",

"gender": "女",

"birthDate": "2003-08-20",

"major": "软件工程",

"class": "2021 级 2 班"

}]

****添加学生信息接口（POST /api/students）****：

****请求参数****：一个 JSON 对象，包含学生的所有必填信息，如学号、姓名、性别、出生日期、身份证号、民族、家庭住址、联系方式、入学时间、专业、班级等。

****响应数据****：返回一个 JSON 对象，表示添加成功后的学生信息，包含服务器生成的唯一学生 ID 等。

****示例****：

收起

json

{

"studentId": "2021003",

"name": "王五",

"gender": "男",

"birthDate": "2003-10-15",

"major": "网络工程",

"class": "2021 级 3 班"}

****修改学生信息接口（PUT /api/students/{studentId}）****：

****请求参数****：一个 JSON 对象，包含需要修改的学生信息字段，如姓名、性别、专业等，studentId 为路径参数，表示要修改的学生的唯一标识。

****响应数据****：返回一个 JSON 对象，表示修改后的学生信息。

****示例****：

收起

json

{

"studentId": "2021001",

"name": "张三（修改后）",

"gender": "男",

"major": "计算机科学与技术（修改后）"}

****删除学生信息接口（DELETE /api/students/{studentId}）****：

****请求参数****：studentId 为路径参数，表示要删除的学生的唯一标识。

****响应数据****：返回一个 JSON 对象，包含操作结果信息，如 {"message": "学生信息删除成功"}。

### **（二）课程管理接口**

****获取课程信息列表接口（GET /api/courses）****：

****请求参数****：可选参数包括课程编号（courseId）、课程名称（courseName）、授课教师（teacherId）等，用于筛选特定课程信息。

****响应数据****：返回一个 JSON 数组，包含符合条件的课程信息对象，每个对象包括课程编号、课程名称、学分、课程类型、授课教师等字段。

****示例****：

收起

json

[

{

"courseId": "C001",

"courseName": "高等数学",

"credit": 5,

"courseType": "必修课",

"teacherId": "T001"

},

{

"courseId": "C002",

"courseName": "大学英语",

"credit": 4,

"courseType": "必修课",

"teacherId": "T002"

}]

****添加课程接口（POST /api/courses）****：

****请求参数****：一个 JSON 对象，包含课程的所有必填信息，如课程编号、课程名称、学分、课程类型、授课教师等。

****响应数据****：返回一个 JSON 对象，表示添加成功后的课程信息，包含服务器生成的唯一课程 ID 等。

****示例****：

收起

json

{

"courseId": "C003",

"courseName": "数据结构",

"credit": 4,

"courseType": "必修课",

"teacherId": "T003"}

****修改课程接口（PUT /api/courses/{courseId}）****：

****请求参数****：一个 JSON 对象，包含需要修改的课程信息字段，如课程名称、学分、授课教师等，courseId 为路径参数，表示要修改的课程的唯一标识。

****响应数据****：返回一个 JSON 对象，表示修改后的课程信息。

****示例****：

收起

json

{

"courseId": "C001",

"courseName": "高等数学（修订版）",

"credit": 6,

"teacherId": "T004"}

****删除课程接口（DELETE /api/courses/{courseId}）****：

****请求参数****：courseId 为路径参数，表示要删除的课程的唯一标识。

****响应数据****：返回一个 JSON 对象，包含操作结果信息，如 {"message": "课程信息删除成功"}。

### **（三）成绩管理接口**

****获取学生成绩列表接口（GET /api/scores/{courseId}）****：

****请求参数****：courseId 为路径参数，表示要获取成绩的课程的唯一标识。

****响应数据****：返回一个 JSON 数组，包含该课程下所有学生的成绩信息对象，每个对象包括学号、姓名、平时成绩、期中成绩、期末成绩、总评成绩等字段。

****示例****：

收起

json

[

{

"studentId": "2021001",

"name": "张三",

"dailyGrade": 80,

"midtermGrade": 85,

"finalGrade": 90,

"overallGrade": 86.5

},

{

"studentId": "2021002",

"name": "李四",

"dailyGrade": 75,

"midtermGrade": 70,

"finalGrade": 80,

"overallGrade": 75.5

}]

****录入学生成绩接口（POST /api/scores）****：

****请求参数****：一个 JSON 对象，包含学生成绩的相关信息，如学号、课程编号、平时成绩、期中成绩、期末成绩等，总评成绩由系统根据预设权重自动计算。

****响应数据****：返回一个 JSON 对象，表示录入成功后的成绩信息，包含服务器生成的唯一成绩 ID 等。

****示例****：

收起

json

{

"scoreId": "S001",

"studentId": "2021003",

"courseId": "C001",

"dailyGrade": 85,

"midtermGrade": 90,

"finalGrade": 92,

"overallGrade": 89.6}

****修改学生成绩接口（PUT /api/scores/{scoreId}）****：

****请求参数****：一个 JSON 对象，包含需要修改的成绩信息字段，如平时成绩、期中成绩、期末成绩等，scoreId 为路径参数，表示要修改的成绩的唯一标识。

****响应数据****：返回一个 JSON 对象，表示修改后的成绩信息。

****示例****：

收起

json

{

"scoreId": "S001",

"studentId": "2021003",

"courseId": "C001",

"dailyGrade": 88,

"midtermGrade": 92,

"finalGrade": 95,

"overallGrade": 91.4}

## **四、数据库表设计**

### **（一）学生表（students）**

****字段****：

student\_id（主键，VARCHAR (20)）：学号，唯一标识每个学生。

name（VARCHAR (50)）：学生姓名。

gender（CHAR (1)）：性别，取值为 “男” 或 “女”。

birth\_date（DATE）：出生日期。

id\_card\_number（VARCHAR (18)）：身份证号，唯一且用于身份验证。

ethnicity（VARCHAR (20)）：民族。

home\_address（VARCHAR (200)）：家庭住址。

contact\_number（VARCHAR (20)）：联系方式，如手机号码。

enrollment\_date（DATE）：入学时间。

major（VARCHAR (50)）：专业名称。

class\_id（VARCHAR (20)）：班级 ID，关联班级表的外键。

****索引****：

在 student\_id 字段上创建主键索引，确保学号的唯一性和快速查询。

在 name 字段上创建普通索引，提高按姓名查询学生信息的效率。

### **（二）课程表（courses）**

****字段****：

course\_id（主键，VARCHAR (20)）：课程编号，唯一标识每门课程。

course\_name（VARCHAR (100)）：课程名称。

credit（INT）：学分。

course\_type（VARCHAR (20)）：课程类型，如 “必修课”、“选修课” 等。

teacher\_id（VARCHAR (20)）：授课教师 ID，关联教师表的外键。

****索引****：

在 course\_id 字段上创建主键索引，保证课程编号的唯一性和快速查询。

在 course\_name 字段上创建普通索引，方便按课程名称搜索课程信息。

### **（三）成绩表（scores）**

****字段****：

score\_id（主键，VARCHAR (20)）：成绩 ID，唯一标识每条成绩记录。

student\_id（VARCHAR (20)）：学号，关联学生表的外键，标识该成绩所属的学生。

course\_id（VARCHAR (20)）：课程编号，关联课程表的外键，标识该成绩所属的课程。

daily\_grade（INT）：平时成绩。

midterm\_grade（INT）：期中成绩。

final\_grade（INT）：期末成绩。

overall\_grade（DECIMAL (5, 2)）：总评成绩，根据平时、期中、期末成绩按照一定权重计算得出，保留两位小数。

****索引****：

在 score\_id 字段上创建主键索引，确保成绩记录的唯一性和快速查询。

在 student\_id 和 course\_id 字段上创建联合索引，提高根据学生和课程查询成绩的效率。

### **（四）考勤表（attendances）**

****字段****：

attendance\_id（主键，VARCHAR (20)）：考勤 ID，唯一标识每条考勤记录。

student\_id（VARCHAR (20)）：学号，关联学生表的外键，标识该考勤记录所属的学生。

attendance\_date（DATE）：考勤日期。

attendance\_status（VARCHAR (20)）：考勤状态，取值为 “出勤”、“迟到”、“早退”、“旷课” 等。

attendance\_method（VARCHAR (20)）：考勤方式，如 “点名”、“扫码” 等。

****索引****：

在 attendance\_id 字段上创建主键索引，保证考勤记录的唯一性和快速查询。

在 student\_id 和 attendance\_date 字段上创建联合索引，便于按学生和日期查询考勤情况。

### **（五）教师表（teachers）**

****字段****：

teacher\_id（主键，VARCHAR (20)）：教师 ID，唯一标识每位教师。

name（VARCHAR (50)）：教师姓名。

gender（CHAR (1)）：性别，取值为 “男” 或 “女”。

birth\_date（DATE）：出生日期。

title（VARCHAR (20)）：职称，如 “教授”、“副教授”、“讲师” 等。

contact\_number（VARCHAR (20)）：联系方式，如手机号码。

department（VARCHAR (50)）：所在系部。

****索引****：

在 teacher\_id 字段上创建主键索引，确保教师 ID 的唯一性和快速查询。

在 name 字段上创建普通索引，方便按教师姓名查找教师信息。

### **（六）班级表（classes）**

****字段****：

class\_id（主键，VARCHAR (20)）：班级 ID，唯一标识每个班级。

class\_name（VARCHAR (50)）：班级名称，如 “2021 级 1 班”。

major\_id（VARCHAR (20)）：专业 ID，关联专业表的外键，标识该班级所属的专业。

****索引****：

在 class\_id 字段上创建主键索引，保证班级 ID 的唯一性和快速查询。

在 class\_name 字段上创建普通索引