**1. 需求分析**

**1.1 目标**

**设计一个学生管理系统，以高效、便捷地管理学生的信息、课程、成绩和其他相关数据。系统应支持学生、教师和管理员的不同需求，提供灵活的功能和直观的操作界面。**

**1.2 功能需求**

**1.2.1 学生模块**

**学生注册：学生可以通过系统注册，提交个人信息，如姓名、学号、联系方式等。**

**个人信息管理：学生可以查看和更新自己的个人信息，包括联系方式、家庭地址等。**

**课程选修：学生可以查看可选课程，并进行选课操作。**

**成绩查询：学生可以查看自己在各个课程中的成绩。**

**学业报告：学生可以生成并查看自己的学业报告单。**

**1.2.2 教师模块**

**教师注册和信息管理：教师可以注册并更新自己的个人信息，包括联系方式、所属部门等。**

**课程管理：教师可以创建和管理课程，设置课程时间、地点及相关信息。**

**学生管理：教师可以查看自己课程的学生信息，包括学生成绩和出勤记录。**

**成绩录入和管理：教师可以录入、修改和删除学生成绩。**

**课程安排：教师可以查看自己课程的安排和学生选课情况。**

**1.2.3 管理员模块**

**用户管理：管理员可以管理系统中的学生、教师和其他管理员账户，包括添加、删除和修改用户信息。**

**课程管理：管理员可以创建和管理课程类型，设置课程规则和选修条件。**

**数据备份与恢复：管理员可以备份和恢复系统数据，确保数据的安全性和完整性。**

**系统设置：管理员可以配置系统设置，如学期安排、课程计划等。**

**1.2.4 通用功能**

**通知系统：系统可以向学生、教师发送重要通知和公告。**

**权限管理：系统应提供细粒度的权限管理功能，确保不同角色的用户只能访问和操作其权限范围内的数据。**

**报表生成：系统应能够生成各种报表，包括学生成绩报表、课程统计报表等。**

**数据导入和导出：系统应支持数据的导入和导出功能，方便与其他系统的数据交互。**

**2. 概要设计**

**2.1 系统架构**

**2.1.1 技术选型**

**前端技术：HTML5、CSS3、JavaScript、React/Vue.js**

**后端技术：Node.js、Express、Spring Boot**

**数据库：MySQL、PostgreSQL**

**服务器：Nginx、Apache**

**其他：Docker（容器化部署）、Git（版本控制）**

**2.1.2 系统组件**

**用户界面：包括学生、教师和管理员的登录、注册和操作界面。**

**业务逻辑层：处理所有的业务逻辑，包括课程管理、成绩录入等。**

**数据访问层：与数据库进行交互，执行数据的增删改查操作。**

**通知系统：实现通知和公告的发送功能。**

**报表生成模块：生成各种报表的功能模块。**

**2.2 数据库设计**

**2.2.1 数据表**

**学生表 (students)**

**student\_id (主键)**

**name**

**gender**

**date\_of\_birth**

**contact\_number**

**address**

**email**

**教师表 (teachers)**

**teacher\_id (主键)**

**name**

**department**

**contact\_number**

**email**

**课程表 (courses)**

**course\_id (主键)**

**course\_name**

**description**

**teacher\_id (外键)**

**schedule**

**学生选课表 (course\_enrollments)**

**enrollment\_id (主键)**

**student\_id (外键)**

**course\_id (外键)**

**enrollment\_date**

**成绩表 (grades)**

**grade\_id (主键)**

**student\_id (外键)**

**course\_id (外键)**

**grade**

**通知表 (notifications)**

**notification\_id (主键)**

**title**

**message**

**date\_sent**

**recipient\_type (学生/教师/所有用户)**

**2.3 用户界面设计**

**2.3.1 学生界面**

**主页：显示欢迎信息、课程列表、个人信息链接。**

**个人信息页面：展示和编辑学生的个人信息。**

**课程选修页面：列出所有可选课程，并提供选课功能。**

**成绩查询页面：显示所有课程的成绩信息。**

**2.3.2 教师界面**

**主页：显示欢迎信息、课程管理链接、学生管理链接。**

**课程管理页面：创建、编辑和删除课程。**

**学生信息页面：查看课程的学生信息及成绩。**

**成绩录入页面：录入和管理学生的成绩。**

**2.3.3 管理员界面**

**主页：显示系统概况、用户管理链接、数据备份链接。**

**用户管理页面：添加、删除和编辑学生、教师和管理员账户。**

**课程管理页面：管理所有课程和课程规则。**

**数据备份页面：执行数据备份和恢复操作。**

**总结**

**学生管理系统应实现学生、教师和管理员之间的高效互动，提供功能丰富、界面友好的操作环境。系统架构需考虑到可扩展性和安全性，以支持未来的功能扩展和数据保护。**