



学籍管理系统需求分析

1. 引言

1.1 项目背景

随着教育信息化的发展，传统的人工管理学籍信息的方式已难以满足高效、准确、便捷的需求。开发一套学籍管理系统，旨在利用计算机技术，实现对学生学籍信息的系统化、规范化和自动化管理，提高管理效率和信息处理能力。

1.2 项目目标

- 建立一个集中的学生学籍信息数据库，确保信息的安全、完整和一致。
- 实现对学生基本信息、课程信息及课程成绩的有效管理。
- 提供便捷的信息查询、统计和报表功能。
- 根据不同用户角色设置相应权限，保证系统的安全性和可操作性。

1.3 适用范围

本系统适用于各类大中专院校的学籍管理工作。

2. 用户分析

2.1 用户角色

系统主要涉及以下用户角色：

- **学生 (Student):**
 - 查看个人基本信息。
 - 查看已选课程信息。
 - 查看个人课程成绩。
 - 注册账户。
 - 查看下学期课程列表。
 - 选课。

- 完成学费在线充值。
- 取消已选课程。
- **教师 (Teacher):**
 - 管理所授课程的学生成绩录入与修改。
 - 查看所授课程的学生信息和成绩。
 - 注册账户并开设课程。
 - 查看所授课程的选课学生名单。
- **教务管理员 (Academic Administrator):**
 - 管理学生基本信息（增、删、改、查）。
 - 管理专业信息（增、删、改、查）。
 - 管理课程信息（增、删、改、查）。
 - 管理学生课程成绩（查询、统计、批量导入、修改审批等）。
 - 录入或删除课程信息。

2.2 用户特点

- **学生:** 数量庞大，计算机操作水平不一，需要简洁易懂的界面和操作流程。
 - **教师:** 熟悉教学业务，对信息准确性要求高，需要便捷的成绩录入和学生信息查询功能。
 - **教务管理员:** 核心用户，需掌握系统各项功能，对数据管理和流程审批有较高要求。
-

3. 功能需求

3.1 学生模块

3.1.1 注册功能:

- **描述:** 学生可以注册账户，注册需满足特定条件才能成功（例如，新生信息已由教务管理员预先导入）。
- **输入:** 学生学号、姓名、证件信息等（用于验证）。
- **输出:** 成功注册的提示，账户创建。

3.1.2 课程查询:

- **描述:** 学生可以查看下学期的课程列表。
- **输入:** 无。
- **输出:** 下学期课程的详细列表。

3.1.3 课程查询 (按条件) :

- **描述:** 学生可以按学分和课程性质等条件查询下学期课程。
- **输入:** 学分范围、课程性质 (如必修、选修) 等。
- **输出:** 符合条件的课程列表。

3.1.4 成绩查询:

- **描述:** 学生在老师给分并经管理员审核后, 查询课程成绩。
- **输入:** 无。
- **输出:** 学生的课程成绩列表。

3.1.5 学费缴纳:

- **描述:** 学生完成学费的在线充值。
- **输入:** 充值金额, 支付方式。
- **输出:** 充值成功的提示, 更新缴费状态。

3.1.6 取消选课:

- **描述:** 学生在正式上课前可以取消已选课程。
- **输入:** 需取消的课程。
- **输出:** 成功取消选课的提示。

3.2 教师模块

3.2.1 注册功能:

- **描述:** 教师注册账户和开设课程, 需要得到管理员 (教务管理员) 监管。
- **输入:** 教师工号、姓名、课程信息等。
- **输出:** 注册成功/待审核提示, 课程开设状态。
- **约束:** 需经教务管理员审批。

3.2.2 查看名单与对应成绩:

- **描述:** 教师可以查看所授课程的选课学生名单与成绩。
- **输入:** 课程名称/编号。
- **输出:** 选课学生列表与成绩。

3.2.3 成绩修改:

- **描述:** 教师可以查看并修改学生成绩。
- **输入:** 学生学号, 课程, 新成绩。
- **输出:** 成绩修改成功的提示。

3.3 管理员模块 (教务管理员)

3.3.1 信息管理:

- **描述:** 教务管理员对学生和教师的部分信息做出修改。
- **输入:** 学生或教师的待修改信息。
- **输出:** 信息修改成功的提示。

3.3.2 课程管理:

- **描述:** 教务管理员录入或删除课程信息。
- **输入:** 课程详细信息或待删除课程编号。
- **输出:** 课程信息操作成功提示。

3.3.3 注册审批:

- **描述:** 教务管理员审核教师和管理员的注册申请。
- **输入:** 待审批的注册申请, 审批结果。
- **输出:** 审批结果通知。

4. 数据需求

4.1 实体及其主要属性

- **学生 (Student):** 学号 (主键), 姓名, 性别, 出生日期, 身份证号, 籍贯, 入学日期, **专业ID (外键)**, **班级ID (外键)**, 联系电话, 邮箱, 家庭住址, 照片路径, 当前状态 (在读/毕业/

休学/退学等)。

- **专业 (Major):** 专业ID (主键), 专业代码, 专业名称, 所属学院名称, 学制。
- **班级 (Class):** 班级ID (主键), 班级名称, 所属专业ID (外键), 年级, 辅导员ID (外键, 指向教师表)。
- **课程 (Course):** 课程ID (主键), 课程编号, 课程名称, 课程类型 (必修/选修), 学分, 学时, 开课学院名称, 描述。
- **开课信息 (OfferedCourse):** 开课ID (主键), 课程ID (外键), 学年, 学期, 教师ID (外键), 上课时间, 上课地点。
- **成绩 (Grade):** 成绩ID (主键), 学生学号 (外键), 开课ID (外键), 平时成绩, 期末成绩, 总评成绩, 绩点, 录入时间, 录入教师ID (外键)。
- **用户 (User):** 用户ID (主键), 用户名, 密码 (加密存储), 真实姓名, **角色ID (外键)**, 联系方式, 状态 (启用/禁用)。
- **角色 (Role):** 角色ID (主键), 角色名称。
- **教师 (Teacher):** 教师ID (主键), 教师工号, 姓名, 性别, 所属学院名称, 职称, 联系方式。

4.2 数据流图 (高层次)

- **学生** -> 系统: 注册、查看信息、选课、缴费。
- **教师** -> 系统: 注册、开设课程、录入/修改成绩、查看学生信息。
- **教务管理员** -> 系统: 维护学生、专业、班级、课程信息, 管理成绩, 生成报表, 信息管理, 注册审批。
- **系统** -> 各用户: 展示信息、反馈操作结果、生成报表。

5. 界面需求 (初步设想)

- 系统应采用B/S架构, 通过浏览器访问。
 - 界面风格统一、简洁、美观, 符合用户操作习惯。
 - 导航清晰, 功能模块易于查找。
 - 表单设计合理, 提供必要的输入校验和提示。
 - 查询结果以列表形式清晰展示, 支持分页和排序。
 - 重要的操作应有确认提示, 防止误操作。
-

6. 部署需求

- **服务器操作系统:** Windows Server。
- **数据库管理系统:** MySQL
- **Web服务器:** Nginx
- **开发语言与框架:** Python (Django, Flask), (html,css,javascript)