**详细设计说明书**

详细设计说明书

目录

1．引言​1

1.1编写目的​1

1.2项目背景​1

1.3定义​1

1.4参考资料​1

2．总体设计​2

2.1需求概述​2

2.2软件结构​2

3．程序描述​2

3.1功能​2

3.2性能​2

3.3输入项目​2

3.4输出项目​2

3.5算法​3

3.6程序逻辑​3

3.7接口​3

3.8存储分配​3

3.9限制条件​3

3.10测试要点​3

详细设计说明书

1. **引言**

随着互联网技术的不断发展，学生选课系统在高校中得到广泛应用。学生选课系统是一种实用的学习管理方式，其目的在于方便学生进行选课，同时为学校管理部门提供选课管理。该系统将由python，flask和mysql实现。

**1.1编写目的**

详细设计说明书是为了阐述软件项目——学生选课系统的具体设计方案，包括系统的目标、需求、架构、功能模块、业务流程、技术实现等细节。该说明书旨在为系统的开发团队提供一个明确、全面的设计蓝图，以便他们可以按照这个设计进行开发和测试。此外，详细设计说明书也为学生、教师和管理员等用户提供了对系统的深入理解，帮助他们明确系统的功能和操作方式，从而更好地使用和维护系统。

**1.2项目背景**

学生选课系统是一个教育信息化项目，旨在提高学校的教学管理水平，方便学生选课，同时增加学生、教师和管理员之间的互动。来源于学校管理层，教育部门。主管部门通常会涉及到学校的信息技术部门、教务部门或教育技术部门等。

**1.3定义**

1. 系统 (System)：指由一组相互关联的组件组成的整体，用于实现某个特定的目标或功能。
2. 软件 (Software)：指计算机程序及其相关文档、数据和元数据的集合。
3. 学生 (Student)：在教育机构中注册并正在接受教育的人。
4. 选课 (Course Selection)：学生根据个人需求和兴趣选择要参加的课程的过程。
5. 系统设计 (System Design)：确定系统架构、组件、接口、数据结构和算法的过程，以满足用户需求和系统规格说明。
6. 数据库 (Database)：用于存储、检索和管理数据的软件仓库。
7. 用户界面 (User Interface)：人与计算机之间进行交互的界面，包括屏幕、键盘、鼠标等。
8. 模块 (Module)：系统中相对独立的部分，具有特定的功能并可以与其他模块进行交互。
9. 模块化 (Modularization)：将系统划分为相对独立、可重用的模块，以提高可维护性和可重用性。
10. 集成 (Integration)：将各个模块组合在一起，形成一个完整的系统，并确保它们能够协同工作。
11. 测试 (Testing)：验证系统是否符合设计要求、是否能够正常运行的过程，包括单元测试、集成测试和系统测试。
12. 文档 (Documentation)：描述系统设计、实现和使用的文字材料，包括用户手册、技术文档等。
13. 安全性 (Security)：保护系统免受未经授权的访问、修改或破坏的能力。
14. 可维护性 (Maintainability)：指系统在运行过程中能够被修改、扩展和升级的能力。
15. 可学习性 (Learnability)：指新用户能够轻松理解并使用系统的能力。
16. 可测试性 (Testability)：指系统能够被测试并验证其是否正确工作的能力。

**1.4参考资料**

Flask 官方文档:https://flask.palletsprojects.com/en/2.0.x/

Flask-SQLAlchemy 官方文档:https://flask-sqlalchemy.palletsprojects.com/en/2.x/

Flask-SQLAlchemy 是 Flask 中常用的 ORM 库,为数据库操作和模型定义提供了支持。

SQLite 官方文档: https://www.sqlite.org/docs.html

Bootstrap 官方文档:https://getbootstrap.com/

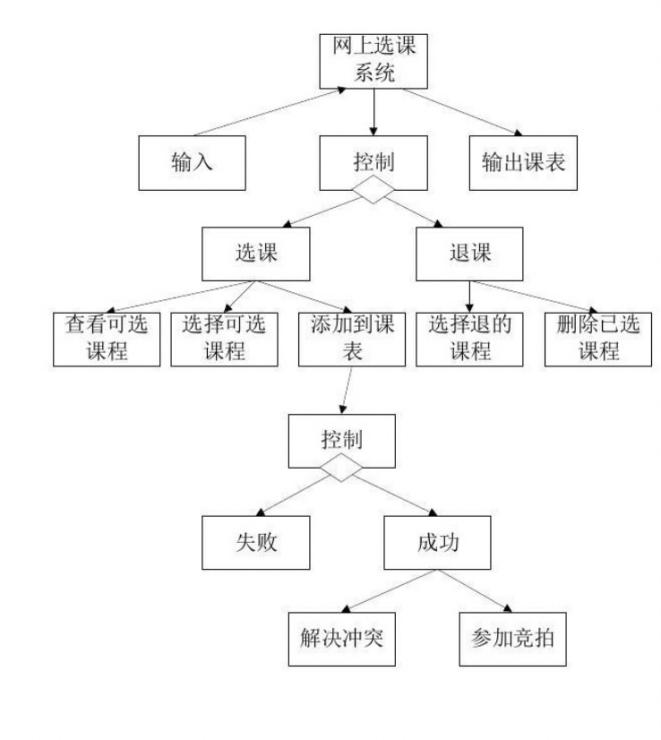
jQuery 文档:https://api.jquery.com/

**2．总体设计**

**2.1需求概述**

1. 确定系统目标和功能：学生选课系统旨在方便学生、教师和管理员进行课程选择、管理、查询和评估。因此，系统的总体设计需要确定这些用户群体的需求，并将其转化为具体的系统目标和功能。
2. 确定系统架构和技术选型：系统的总体设计需要确定系统的架构和技术选型。例如，可以选择基于Web的应用程序架构或客户端/服务器架构，并使用Java、Python、PHP等编程语言进行开发。
3. 确定系统模块和功能划分：学生选课系统主要包括学生选课、退课、查看选课信息，教师评课、填写学生成绩，管理员设置系统参数和权限等功能。因此，系统的总体设计需要将这些功能合理地划分为不同的模块，并确定每个模块之间的关系和操作流程。
4. 确定数据库结构和数据关系：学生选课系统需要存储学生信息、课程信息、选课信息和成绩信息等数据。因此，系统的总体设计需要确定数据库的结构和数据关系，以支持系统的各个模块正常运行。
5. 确定用户界面和交互方式：系统的总体设计需要确定用户界面的风格和交互方式，例如使用表单、按钮、菜单等元素实现用户与系统的交互，以及使用户界面易于使用和操作。
6. 确定系统安全和权限控制：系统的总体设计需要考虑系统的安全性和权限控制。例如，管理员需要设置不同用户的权限级别，以确保系统的数据安全和正常运行。

**2.2软件结构**



**3．程序描述**

1. 登录模块：该模块负责用户登录功能。用户输入用户名和密码后，系统将验证用户身份，并根据用户角色跳转到相应的页面。
2. 学生选课模块：该模块负责学生选课功能。学生可以在该模块中查看可选课程列表，选择感兴趣的课程，并查看已选课程列表。
3. 退课模块：该模块负责学生退课功能。学生可以在该模块中查看已选课程列表，选择需要退出的课程，并更新已选课程列表。
4. 查看选课信息模块：该模块负责学生查看选课信息功能。学生可以在该模块中查看已选课程的详细信息，包括课程名称、任课教师、上课时间、上课地点等信息。
5. 教师评课模块：该模块负责教师评课功能。教师可以在该模块中查看所授课程的选课学生列表，对每个学生进行评价，并填写学生成绩。
6. 管理员设置系统参数和权限模块：该模块负责管理员设置系统参数和权限功能。管理员可以在该模块中设置系统参数，如选课时间、退课时间等，并设置用户角色和权限，以确保系统的安全性和稳定性。
7. 数据库管理模块：该模块负责数据库管理功能。管理员可以在该模块中备份和恢复数据库，以确保数据的安全性和完整性。

**3.1功能**

1. 登录与注册

用户登录:支持学生和管理员登录,验证用户名和密码

用户注册:新用户注册个人账户

2. 个人中心

个人信息管理:查看和修改个人基本信息

我的选课:查看已选择的课程清单

推荐课程:展示系统推荐的匹配课程

3. 课程管理

课程查询:支持多条件查询课程信息

选课:学生选择感兴趣的课程加入课表

退课:学生从课表中删除不再选择的课程

4. 学生信息管理

学生信息维护:添加、编辑、删除学生基本资料

学生考勤信息:查看学生截止考勤情况

5. 课程资源管理

课程开设管理:添加、修改已开设课程信息

教材资源管理:上传与管理课程教材资源

6. 新闻动态

发布公告:管理员发布系统内公告

查看公告:学生可浏览已发布公告

订阅功能:学生选择订阅关注课程的动态

7. 个人计划

培养计划:视图个人四年学习制定计划

计划管理:编辑个人培养计划内容

8. 其他

用户账户:组织用户登录账户信息

系统设置:后台设置系统各项基本信息

**3.2性能**

1. 系统处理的准确性和及时性：学生选课管理系统应该满足高准确性和及时性的性能要求，以确保系统能够长期、安全、稳定、可靠、高效地运行。
2. 系统响应速度：学生选课管理系统应该具有快速响应的能力，以便能够及时响应用户的请求并返回相应的结果。
3. 系统数据精度：学生选课管理系统应该具备高精度的数据管理能力，以确保数据的准确性和完整性。
4. 系统数据存储：学生选课管理系统应该具备大容量的数据存储能力，以存储大量的学生信息、课程信息和选课信息等。

**3.3输入项目**

1. 用户登录输入：用户名/学号、密码

2. 用户注册输入：用户名/学号、姓名、密码、确认密码

3. 个人信息管理输入：姓名、性别、生日、学院、专业、其他个人资料

4. 课程查询输入：课程名称、任课教师、开课学期、学分、课程类别

5. 选择课程输入：课程ID

6. 发布公告输入：公告标题、公告内容

7. 课程管理输入：课程号、课程名称、开课学期、教学计划、教材资源

8. 学生管理输入：学号、姓名、班级、学院、其他学生信息

9. 个人计划输入：年级规划、专业核心课、通识教育、实践环节

**3.4输出项目**

1. 用户登录输出：登录成功与失败提示

2. 个人中心输出：个人资料、已选择课表、推荐课程列表

3. 课程查询输出：课程列表搜索结果

4. 选择课程输出：选课成功提示

5. 发布公告输出：最新公告列表

6. 课程管理输出：课程信息表格、教材资源链接

7. 学生管理输出：学生信息表格、考勤统计表格

8. 个人计划输出：不同年级阶段计划视图、计划项目编辑表单

9. 系统设置输出：系统配置项、版本信息

10. 推荐课程输出：匹配课程卡片视图

11. 详情页面输出：课程/公告详情内容、评论内容列表

**3.5算法**

1. 用户登录算法

获取用户名和密码，调用数据库查询函数,在用户表中搜索用户名，如果搜索到,匹配密码是否一致，一致则登录成功,不一致返回错误

2. 个人中心算法

获取用户id，调用数据库函数,查询用户基本信息表获取个人数据，调用选课表函数,获取已选择课程数据，调用匹配算法函数,匹配推荐课程，将数据渲染个人中心页面展示

3. 课程查询算法

获取查询条件参数，根据条件拼接数据库查询语句，调用数据库查询函数执行查询，将结果集返回

4. 选择课程算法

获取课程id，调用数据库函数插入数据到选课表，返回成功提示

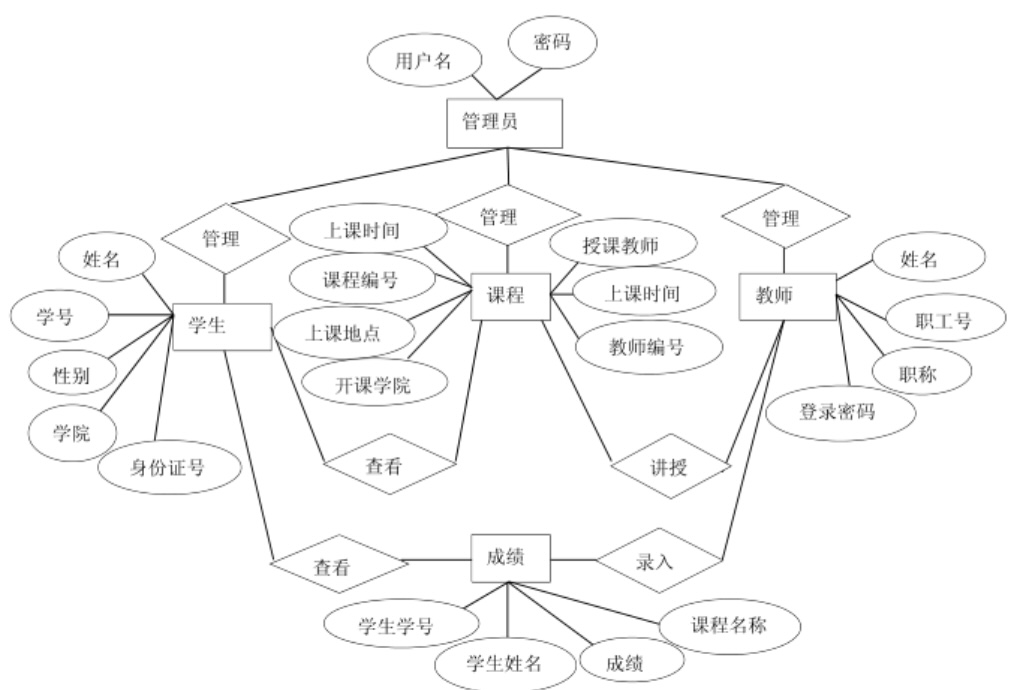
5. 发布公告算法

获取标题和内容数据，调用数据库函数新增一条公告记录，返回最新公告列表

6. 推荐算法

获取用户信息与课程信息，给每个课程计算匹配度分数，通过分数排序后选取前N门课，返回匹配结果

**3.6程序逻辑**



**3.7接口**

1. 用户接口：用户接口包括学生、教师和管理员等用户与系统进行交互的接口。例如，学生可以通过学生端接口进行选课、退课、查看选课信息和成绩等操作；教师可以通过教师端接口进行评课和填写学生成绩等操作；管理员可以通过管理员端接口进行系统设置、用户管理和数据备份等操作。
2. 数据接口：数据接口包括系统内部的数据交互接口，例如学生信息、选课信息、退课信息、成绩信息等数据的读取和写入接口。
3. 外部接口：外部接口包括系统与其他系统进行数据交互的接口。例如，学生选课系统可能需要与学校的其他系统进行数据交互，如学生信息管理系统、课程管理系统等。
4. 硬件接口：硬件接口包括系统与计算机硬件设备进行交互的接口。例如，学生选课系统可能需要与打印机等硬件设备进行交互，以便打印选课表格和成绩单等。
5. 软件接口：软件接口包括系统与其他软件进行交互的接口。例如，学生选课系统可能需要与操作系统、数据库管理系统、Web服务器等软件进行交互。

**3.8存储分配**

1.学生信息表(STUDENT):

表字段:学号、姓名、性别、学院、专业、入学年份、密码、学号(作为主键)用于存储每个学生的基本信息

2.课程信息表(COURSES):

表字段:课程编号、课程名称、学分、任课教师、课程类型、学时、上课时间等，用于存储每个课程的详细信息

3. 选课信息表(SELECTED\_COURSES):

表字段:学号、课程编号，用于记录每个学生选修的具体课程

4.新闻讨论表(NEWS):

表字段:标题、内容、发帖人、新闻ID、是否原帖等用于存储课程讨论区的帖子和回复

**3.9限制条件**

1. 数据库限制：学生选课系统需要使用MySQL数据库进行数据存储和管理，因此需要遵循MySQL数据库的相关规范和标准。同时，为了保证数据的安全性和完整性，需要进行数据库备份和恢复等操作。
2. 网络限制：学生选课系统需要基于互联网进行开发和使用，因此需要考虑网络协议、网络传输、网络安全等因素，以便保障系统的稳定性和安全性。
3. 用户限制：学生选课系统主要面向学生、教师和管理员等用户，因此需要考虑不同用户的需求和权限，以便提供个性化的服务和功能。

**3.10测试要点**

1. 功能测试：测试每个功能模块是否符合设计要求，是否能够正确执行。具体包括但不限于：登录与注册、选课、退课、成绩查询、课程管理、用户管理、系统管理等功能模块的测试。
2. 性能测试：测试系统在不同负载下的响应时间、吞吐量、稳定性等性能指标是否符合要求。
3. 界面测试：测试系统的界面是否美观、易用、符合用户习惯，以及响应速度等。
4. 安全测试：测试系统的安全性，包括但不限于：密码加密、权限控制、防止SQL注入等。
5. 兼容性测试：测试系统在不同浏览器、操作系统、设备等不同平台上的兼容性。