|  |
| --- |
| UESTC |
| 软件需求分析 |
| 软件工程大作业 |

|  |
| --- |
| 王哲宇 2023090912005  2024-11-30 |

软件需求分析报告文档

[1. 引言 2](#_Toc182829759)

[1.1 编写目的 2](#_Toc182829760)

[1.2 项目风险 2](#_Toc182829761)

[1.3 文档约定 3](#_Toc182829762)

[1.4 预期读者和阅读建议 3](#_Toc182829763)

[1.5 产品范围 4](#_Toc182829764)

[2. 综合描述 5](#_Toc182829765)

[2.1 产品的状况 6](#_Toc182829766)

[2.2 产品的功能 7](#_Toc182829767)

[2.3 用户类和特性 8](#_Toc182829768)

[2.4 运行环境 9](#_Toc182829769)

[2.5 设计和实现上的限制 10](#_Toc182829770)

[2.6 假设和约束(依赖) 11](#_Toc182829771)

[3. 外部接口需求 13](#_Toc182829772)

[3.1 用户界面 13](#_Toc182829773)

[3.2 硬件接口 14](#_Toc182829774)

[3.3 软件接口 16](#_Toc182829775)

[4. 系统用例设计 17](#_Toc182829776)

[5. 系统功能需求 18](#_Toc182829777)

[5.1 输入／输出数据 18](#_Toc182829778)

[5.2 用例描述 19](#_Toc182829779)

[6. 其它非功能需求 21](#_Toc182829780)

[6.1 性能需求 21](#_Toc182829781)

[6.2 安全措施需求 21](#_Toc182829782)

[6.3 安全性需求 21](#_Toc182829783)

[6.4 软件质量属性 22](#_Toc182829784)

[6.5 业务规则 22](#_Toc182829785)

[6.6 用户文档 22](#_Toc182829786)

[7. 词汇表 23](#_Toc182829787)

[8. 数据定义 24](#_Toc182829788)

[9. 分析模型 25](#_Toc182829789)

[10. 待定问题列表 26](#_Toc182829790)

# 引言

本引言部分是成绩管理系统项目软件产品需求分析报告的概览，阅读者可以通过本引言了解本文档编写的思路，以及了解应当如何阅读、理解和解释这份文档。

## 编写目的

通过此软件产品，学校的信息管理将更加便捷，使得学生查阅成绩、教师登记的操作更加高效化，维护信息的操作便捷化，管理员对信息的管理更加统一化、规范化与合理化。

具体来说，学生可以对个人的身份信息进行记录并存储，并且可以根据学生身份信息对个人信息、个人成绩以及个人课程进行即时的查询，同时，系统为同学们提供维护个人信息的开放渠道。

同理，教师可以通过此软件产品即时查询个人信息，修改个人信息，并且在该系统中实现开课操作，查阅自己的全部课程以及对学生的成绩信息进行登录。

对于管理员，本系统也为维护个人信息，增加学生信息和删除学生信息提供了相应的渠道。

总而言之，学生成绩管理系统最终能够使得高校对成绩和课程的信息管理与查询更加高效且智能，避免繁杂的人工操作，给高校管理者、教师与同学带来极大便利。

本软件需求分析报告文档意在详细说明本软件产品的需求规格，针对文档描述的学生成绩管理系统进行对于以上描述内容的准确的定义。

## 项目风险

* **任务提出者（项目发起人或客户）**：
  + **需求风险**：提出的需求可能不明确或不完整，导致项目目标不清晰。
  + **预算风险**：预算可能不足以覆盖项目的实际成本，导致项目延期或质量下降。
  + **时间风险**：项目时间表可能过于紧张，导致项目延期。
  + **变更风险**：项目进行中需求变更可能导致成本增加和时间延长。
  + **市场风险**：产品可能不符合市场预期，导致销售不佳。
* **软件开发者（开发团队）**：
  + **技术风险**：技术选型不当或技术实现难度大，可能导致项目延期或失败。
  + **人力资源风险**：团队成员技能不足或流动性大，影响项目进度和质量。
  + **管理风险**：项目管理不当，如沟通不畅，协调不力，可能导致项目目标偏离。
  + **质量风险**：代码质量不高，可能导致后期维护成本增加。
  + **交付风险**：项目可能无法按期交付，影响客户满意度。
* **产品使用者（用户）**：
  + **操作风险**：用户可能对新系统不熟悉，导致操作错误或效率低下。
  + **接受度风险**：用户可能对新系统不满意，导致系统使用率低。
  + **数据风险**：数据迁移或整合过程中可能丢失或损坏数据。
  + **安全风险**：系统可能存在安全漏洞，导致数据泄露或其他安全问题。
  + **兼容性风险**：新系统可能与现有系统不兼容，导致额外的集成成本。

在本阶段，即项目的开发阶段，首要风险承担者是软件开发者，因为他们直接负责实现项目的技术目标。他们需要确保技术选型正确、代码质量高、项目按时交付，并且能够满足用户的需求。同时，任务提出者需要确保需求的明确性和合理性，以及提供足够的资源和支持。产品使用者的风险主要在于系统的接受度和使用效果，他们需要参与到系统的测试和反馈中，以确保系统的可用性和有效性。

## 文档约定

1. **正文风格**：
   * + 使用清晰、简洁的语言。
     + 保持一致的时态（通常是现在时）。
     + 使用主动语态而非被动语态。
     + 避免使用缩写和俚语。
   1. **提示方式**：
      * **注意（Note）**：提供额外信息或解释，对理解正文内容有帮助。
      * **警告（Warning）**：指出可能引起问题的情况，需要特别注意。
      * **错误（Error）**：指出文档中的错误或不一致之处。
      * **提示（Tip）**：提供有用的技巧或最佳实践。
   2. **重要符号**：
      * **粗体（Bold）**：用于强调关键词或重要的术语。
      * **斜体（Italic）**：用于书籍、文档、电影等的标题，或用于强调新引入的术语。
      * **下划线（Underline）**：用于强调网址或文件路径。

在需求管理中，高层次需求通常可以被其所有细化的需求所继承。这意味着细化的需求应该与高层次需求保持一致，并且细化需求的实现应该支持高层次需求的实现。这种继承关系有助于确保项目的目标和方向在各个层面上保持一致。

本文档中，每个需求具有自己的优先级，有助于项目团队确定工作的优先顺序。优先级可以根据需求的重要性、紧迫性或对项目成功的影响来确定。优先级别包括：

1. **高（High）**：必须实现的需求，对项目的成功至关重要。
2. **中（Medium）**：重要的需求，但不是项目成功的决定性因素。
3. **低（Low）**：可以推迟或在资源允许的情况下实现的需求。

## 预期读者和阅读建议

本软件产品需求分析报告预期面向的读者群体包括但不限于以下几类：

1. **用户（End Users）**：
   * 包括学生、教师和教务管理员，他们将直接使用该系统进行日常的教学和管理工作。
   * **阅读建议**：关注“综合描述”与“外部接口需求”部分，以了解系统的基本功能和操作流程，同时可以阅读“非功能需求”中的“用户文档”部分。用户应重点阅读与自己角色相关的功能描述，如学生用户应关注“学生用户”部分，教师用户应关注“教师”部分。
2. **开发人员（Developers）**：
   * 负责实现软件产品，需要详细了解系统架构、技术细节和接口规范。
   * **阅读建议**：深入阅读“综合描述”和“外部接口需求”部分，了解系统架构、类图设计和包结构。同时，关注“系统功能需求”部分，以理解功能的实现细节。
3. **项目经理（Project Managers）**：
   * 负责项目的规划、执行和监控，需要了解项目的整体需求和进度。
   * **阅读建议**：重点关注“综合描述”部分，了解项目的结构和主要组件。同时，查看“外部接口需求”和“其他非功能需求”部分，以掌握项目的关键数据和新特性。
4. **营销人员（Marketing Personnel）**：
   * 负责产品的推广和销售，需要了解产品的特点和卖点。
   * **阅读建议**：阅读“综合描述”和“其他非功能需求”部分，以了解系统的核心功能和新特性，这些信息有助于制定营销策略和推广材料。
5. **测试人员（Testers）**：
   * 负责验证软件产品的质量，需要了解需求的详细描述以设计测试用例。
   * **阅读建议**：仔细阅读整个文档，特别是“综合描述”和“外部接口需求”部分，以确保测试覆盖所有功能点。同时，关注“系统功能需求”部分，以了解测试环境的配置。
6. **文档编写人员（Documentation Writers）**：
   * 负责编写和维护项目文档，需要了解项目的所有细节，以便编写用户手册和在线帮助文档。
   * **阅读建议**：全面阅读整个文档，特别是“综合描述”和“外部接口需求”部分，以获取编写用户手册所需的详细信息。同时，参考“新增分段统计功能”部分，以包含新特性的文档中。

## 产品范围

* **产品名称**：学生成绩管理系统
* **开发目的**：

学生成绩管理系统旨在提供一个集成化平台，用于管理学生成绩、课程信息以及相关教学活动。该系统通过提供一个用户友好的界面，使得学生、教师和教务管理员能够高效地访问和处理成绩数据，从而提高教育管理的效率和透明度。

* **利益**：
* **学生**：能够方便地查询个人成绩和课程信息，更好地监控自己的学术进展。
* **教师**：可以轻松管理课程内容、录入和查询学生成绩，以及开设新课程。
* **教务管理员**：可维护用户账号和学生、教师信息，确保数据的准确性和安全性。
* **目标**：
  + 实现一个集中的成绩管理解决方案，以减少手动处理成绩的工作量。
* 提高成绩管理的准确性和及时性，确保成绩信息的一致性和可靠性。
* 增强学生、教师和教务管理员之间的信息共享和沟通。
* **与企业目标和业务策略的联系：**

学生成绩管理系统的开发与企业的教育信息化目标紧密相连。在当前数字化转型的背景下，教育行业越来越依赖于技术来提高教学质量和管理效率。以下是该系统如何与企业目标和业务策略相联系：

通过自动化成绩管理流程，减少手动输入错误，提高数据处理速度，从而提升整体运营效率；系统收集和分析的成绩数据可以为教育机构提供洞察，帮助制定更有效的教学策略和资源分配；通过提供一个直观、易用的平台，增强学生和教师的用户体验，提高用户满意度和忠诚度；系统的设计允许灵活地添加新功能，如成绩分段统计，支持教育创新和个性化教学需求；系统的设计考虑到了数据安全和隐私保护，符合教育行业的合规性要求；系统采用模块化设计，便于未来的扩展和维护，确保长期的可持续发展。

一言以蔽之，通过实现这些目标，学生成绩管理系统不仅能够提升教育机构内部的工作效率，还能够增强其在市场上的竞争力，支持教育机构实现长期的业务增长和成功。

# 综合描述

这一部分概述了正在定义的软件产品的作用范围以及该软件产品所运行的环境、使用该软件产品的用户、对该软件产品己知的限制、有关该软件产品的假设和依赖。

## 产品的状况

**学生成绩管理系统**是一个专为教育机构设计，用以管理学生成绩、课程信息和教学活动的软件产品。以下是该软件产品的不同情况说明：

1. **产品系列中的下一成员**：
   * 该系统不是某个产品系列中的下一成员。它是作为一个独立的解决方案来开发的，旨在满足特定教育机构的需求。
2. **成熟产品所改进的下一代产品**：
   * 该系统不属于成熟产品的下一代改进。它是根据当前教育管理的需求全新设计的，以适应现代教育环境中对成绩管理系统的新要求。
3. **现有应用软件的替代品（升级产品）**：
   * 该系统可以被视为现有成绩管理应用软件的替代品。它旨在提供更高效、更用户友好的解决方案，以替代那些过时或功能不足的系统。
4. **一个新型的、自主型的产品**：
   * 该系统是一个新型的、自主型的产品。它结合了最新的技术趋势和教育实践，提供了一个创新的解决方案，以满足现代教育机构的需求。

**学生成绩管理系统** 可以是大系统的一个组成部分，也可以与其它系统和机构存在基本的相互关系。以下是具体的说明：

1. **大系统的一个组成部分**：
   * 如果该系统是大系统的一个组成部分，比如一个全面的教育管理平台，那么它将与平台内的其他模块（如学生信息管理系统、课程管理系统、教师评估系统等）紧密集成。
   * 该系统将通过共享数据库、API接口和用户界面集成，以确保数据的一致性和流程的连贯性。
2. **与其它系统和机构之间的相互关系**：
   * 如果该系统与其它系统和机构存在基本的相互关系，比如需要与国家教育管理部门的系统进行数据交换，那么它将通过标准化的数据接口和协议来实现这一点。
   * 这些接口可能包括数据导入/导出功能、Web服务、RESTful API等，以确保与外部系统的兼容性和数据同步。

**接口定义**

为了确保软件产品与大系统或其它系统之间的有效集成，以下是接口定义：

1. **API接口**：
   * 定义了系统提供的API端点，以及这些端点如何被外部系统调用，包括请求和响应的数据格式。
2. **用户界面集成**：
   * 定义了系统如何与其他系统的用户界面集成，包括单点登录（SSO）和用户界面框架的共享。
3. **数据交换格式**：
   * 定义了系统与其他系统交换数据时使用的数据格式，如JSON、XML等。
4. **安全协议**：
   * 定义了系统在与外部系统交互时使用的加密和认证协议，以确保数据传输的安全性。

## 产品的功能

本软件旨在为用户提供高效、便捷的学生成绩管理功能，覆盖学生、教师、管理员三类用户的核心需求。以下是软件的主要功能描述：

* **主要功能**

1. **学生功能**
   * **信息维护**：学生可以修改个人基本信息，如联系方式、地址等。
   * **成绩查询**：根据课程号查询已修课程成绩，支持按学期分类查询。
2. **教师功能**
   * **个人信息管理**：维护教师的基本信息，如邮箱、研究方向等。
   * **课程管理**：
     + 新增课程：根据课程要求创建新的课程条目。
     + 课程查询：查看自己教授课程的基本信息（如课程编号、学生名单等）。
   * **成绩录入与管理**：为学生录入课程成绩，支持批量录入和修改。
3. **管理员功能**
   * **用户管理**：新增、删除或修改学生和教师账户，重置密码。
   * **信息管理**：维护学校的课程信息、教师信息和学生基本信息。
   * **系统设置**：设置登录错误次数限制、调整系统默认参数等。
4. **系统功能**
   * **登录与权限控制**：提供基于用户角色的权限分级，确保数据访问安全性。
   * **数据备份与恢复**：定期备份成绩和用户数据，支持数据恢复功能。

* **非作用领域**
* 本系统**不支持**以下功能：
  + 高级教学分析功能（如数据挖掘或课程优化建议）。
  + 校外系统的集成（如与外部考勤系统或第三方学习管理系统的联动）。
  + 实时通讯功能（如即时消息或在线协作）。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **用户 \ 功能** | 功能1 | 功能2 | 功能3 | 功能4 | 功能5 |
| 管理员 | 修改信息 | 增加用户 | 删除用户 | 无 | 无 |
| 教师 | 信息查询 | 修改信息 | 开课 | 全部课程 | 成绩登录 |
| 学生 | 信息查询 | 成绩查询 | 课程查询 | 修改信息 | 无 |

表2-1 用户功能表

## 用户类和特性

根据本软件的使用场景与业务需求，将可能使用本软件的用户划分为以下类别，并描述其主要特征和需求：

* **重要用户类**

1. **学生**
   * **特征**：学生为软件的主要用户之一，他们以个人身份使用系统，关注与自身学业相关的信息。
     + 使用频率：中等，通常在学期内较为活跃。
     + 技术能力：一般，不具备专业的技术知识。
   * **需求**：
     + 查询个人信息与成绩数据。
     + 修改个人基础信息。
     + 获取课程相关成绩反馈。
2. **教师**
   * **特征**：教师是课程信息和学生成绩的主要维护者，对数据的完整性和准确性负责。
     + 使用频率：高，特别是在课程开设与成绩录入期间。
     + 技术能力：中等，对系统操作较为熟悉。
   * **需求**：
     + 查看和维护所教授课程的详细信息。
     + 添加新课程，录入和管理学生成绩。
     + 统计分析学生成绩（如分段统计）。
3. **管理员**
   * **特征**：管理员是系统的核心管理者，负责用户账户和数据的全局维护。
     + 使用频率：较低，主要在系统维护、用户权限管理期间使用。
     + 技术能力：较高，熟悉系统和数据库操作。
   * **需求**：
     + 新增、删除或修改学生、教师账户及相关信息。
     + 管理课程信息及系统配置参数。
     + 确保系统数据安全性，执行数据备份与恢复。

* **重要团队类**

1. **学校管理层**
   * **特征**：学校领导通过分析学生成绩数据和课程管理数据，做出学术评估和资源分配的决策。
     + 使用频率：低，主要依赖报表和分析数据。
     + 技术能力：一般，通过报表工具间接获取数据。
   * **需求**：
     + 定期生成学生成绩和教师教学绩效的统计报表。
     + 获取课程完成率、学生平均成绩等高层次数据视图。
2. **IT技术支持团队**
   * **特征**：为软件提供部署和维护支持，关注系统的技术细节和稳定性。
     + 使用频率：低，仅在出现问题时介入。
     + 技术能力：专业，熟悉软件架构和服务器管理。
   * **需求**：
     + 系统的日志记录、异常报告与接口信息。
     + 系统的扩展性与可维护性文档。
3. **外部关联系统用户**
   * **特征**：通过API或报表与系统交互的其他组织或子系统，如财务系统、数据中心等。
     + 使用频率：间歇性，取决于具体业务。
     + 技术能力：高，通常由专业开发者或系统管理员操作。
   * **需求**：
     + 清晰定义的接口文档和数据格式规范。
     + 提供历史数据的批量导入或导出功能。

## 运行环境

* **硬件平台**
* **客户端硬件需求**：
  + CPU：双核2.0GHz以上处理器。
  + 内存：4GB以上。
  + 硬盘：至少2GB可用存储空间。
  + 显示器分辨率：1024x768或更高。
* **服务器硬件需求**：
  + CPU：四核3.0GHz以上处理器。
  + 内存：8GB以上（支持并发用户数视需求增加）。
  + 硬盘：至少20GB可用存储空间（用于存储数据库和日志文件）。
  + 网络：100Mbps及以上带宽。
* **操作系统和版本**
* **客户端操作系统**：
  + Windows 7/8/10或更高版本。
  + Linux桌面环境（如Ubuntu 20.04及以上）。
* **服务器操作系统**：
  + Windows Server 2016及以上。
  + Linux服务器环境（推荐CentOS 7或Ubuntu Server 20.04）。
* **支撑环境**
* **数据库（可选）**：
  + MySQL 5.7及以上，或MariaDB 10.5及以上。
  + 可选：支持其他关系型数据库如PostgreSQL。
* **开发与运行环境**：
  + JDK 1.8（Java开发环境）。
  + Apache Tomcat 9.0或Jetty作为应用服务器。
  + Eclipse或其他IDE（用于开发阶段）。
* **其他软件组件（可选）**
* **日志管理**：Log4j 2.0，用于记录系统运行日志。
* **用户界面库**：Swing或JavaFX（用于实现客户端图形界面）。
* **数据传输**：基于HTTP的RESTful API，支持JSON格式的数据交互。
* **与软件共存的应用程序**
* **办公应用**：
  + Microsoft Office或LibreOffice，用于导出或查看报表。
* **网络应用**：
  + Web浏览器（如Google Chrome、Mozilla Firefox）可通过访问API调试系统服务。
* **其他系统**：
  + 可选对接的校内系统（如校园财务系统、教务系统等）。

## 设计和实现上的限制

在开发学生成绩管理系统时，受多方面因素的影响，开发人员的自由选择受到一定限制。以下列出了主要限制内容及其原因：

**1. 必须使用的特定技术、工具、编程语言和数据库**

* **编程语言**：必须使用Java作为开发语言。
* **原因**：客户要求基于Java开发，确保系统的跨平台兼容性，同时符合现有团队的技术栈。
* **数据库**：必须使用MySQL或MariaDB。
  + - * **原因**：学校现有的信息系统已采用MySQL作为数据库，便于数据迁移与系统集成。
* **开发工具**：必须使用Eclipse IDE或类似支持Java开发的工具。
  + - * **原因**：开发团队已熟悉Eclipse环境，便于快速开发和调试。

**2. 避免使用的特定技术、工具、编程语言和数据库**

* **技术与工具**：避免使用未成熟或实验性的技术（如beta版的框架或工具）。
  + **原因**：系统稳定性需求较高，不适合采用不稳定或缺乏长期支持的技术。
* **编程语言**：避免使用Python、Ruby等非团队主流语言。
  + **原因**：团队对这些语言不够熟悉，会增加开发成本和维护难度。

**3. 开发规范和标准**

* **编码标准**：必须遵循Java代码规范（如Google Java Style Guide）。
  + **原因**：确保代码可读性，提高后续维护和二次开发效率。
* **设计符号表示**：使用UML（统一建模语言）进行系统设计，绘制用例图、类图、数据流图等。
  + **原因**：客户要求统一的设计文档格式，便于第三方审核和维护。

## 假设和约束(依赖)

* **假设因素**

1. **计划使用的商业组件或其他软件部件**
   * + **假设**：本软件将使用成熟的MySQL数据库管理系统，并假定其性能能够满足需求。
     + **影响**：如果数据库性能不达标或功能不符合实际需求，可能需要更换数据库，导致开发延误和额外成本。
     + **假设**：采用现有的Java Swing框架作为用户界面实现工具，假定其易用性和性能能够满足需求。
     + **影响**：如果用户对界面的性能或设计风格不满意，可能需要调整UI框架或投入更多资源进行优化。
2. **用户界面的特殊设计约定**
   * + **假设**：用户界面遵循学校现有信息化系统的设计规范（如配色、布局）。
     + **影响**：如果学校的设计规范发生变更，将导致用户界面需要重新设计和实现。
3. **有关本软件用户的若干假定**
   * + **假设**：假定用户（学生、教师、管理员）具备基础计算机操作能力，能够理解并操作软件提供的功能模块。
     + **影响**：如果用户实际操作能力低于预期，可能需要开发更详细的帮助文档或增加培训成本。
     + **假设**：假定管理员用户熟悉基本的数据库操作，能够理解并应用系统的高级功能。
     + **影响**：如果管理员的技术水平不足，可能需要提供额外的技术支持或简化高级功能。
4. **有关本软件开发工作的若干假定**
   * + **假设**：假定学校承诺提供相关的硬件设备（如服务器）和开发工具的许可支持。
     + **影响**：如果设备或工具的供应出现延误，将直接影响开发进度。
     + **假设**：上级部门将对系统开发工作提供政策支持，包括数据访问权限和跨部门协作的协调。
     + **影响**：如果政策支持不足，数据对接和用户需求收集可能受阻，导致需求范围缩小或工期延长。
5. **有关本软件运行环境的若干假定**
   * + **假设**：系统将运行在一个稳定且高性能的校内网络环境中。
     + **影响**：如果网络环境不稳定，可能导致系统性能下降或某些功能无法正常使用。
     + **假设**：客户端用户设备将符合推荐的最低硬件配置。
     + **影响**：如果实际用户设备性能低于推荐配置，可能需要调整软件的兼容性和性能优化策略。

* **外部约束因素**

1. **工期约束**
   * + **限制**：项目必须在学期开始前完成并投入使用（如6个月内完成）。
     + **影响**：工期紧张可能导致功能范围的缩减或降低测试的全面性，影响系统质量。
2. **经费约束**
   * + **限制**：开发预算有限，仅能覆盖基本的开发成本和部分硬件设备采购费用。
     + **影响**：经费不足可能导致无法引入高级功能或外部专业技术支持，影响项目整体质量。
3. **人员约束**
   * + **限制**：开发团队成员有限，且开发人员的专业技能水平参差不齐。
     + **影响**：可能延长开发周期或需要增加培训时间，影响项目进度。
4. **设备约束**
   * + **限制**：校方提供的服务器和用户设备性能有限。
     + **影响**：如果硬件资源不足，将影响系统性能，可能需要优化软件运行效率或另行采购设备。
5. **地理位置约束**
   * + **限制**：开发团队可能分布在不同地区，需依赖远程协作工具进行沟通和开发。
     + **影响**：远程协作可能导致沟通效率降低，进而影响需求确认和开发进度。
6. **其他项目约束**
   * + **限制**：学校学术政策或教学计划的调整可能会影响需求范围（如课程信息的变化）。
     + **影响**：如果政策变化频繁，将增加需求分析和功能调整的难度，延误项目交付时间。

# 外部接口需求

通过本节描述可以确定，保证软件产品能和外部组件正确连接的需求。关联图仅能表示高层抽象的外部接口，必须对接口数据和外部组件进行详细描述，并且写入数据定义中。如果产品的不同部分有不同的外部接口，那么应该把这些外部接口的全部详细需求并入到这一部分实例中。

注意：必须将附加用户类的特征与外部接口需求加以区分，附加用户类的特征描述的是通过接口取得软件产品的数据和服务的人的需求；而外部接口需求描述的是接口本身的需求。

## 用户界面

**1. 图形用户界面(GUI)标准与风格**

* 系统将采用标准的 **Java Swing** 框架实现用户界面，并遵循现代教育信息化产品的简洁风格。
* 所有界面均采用一致的导航和布局规则，确保用户操作的直观性和一致性。

**2. 屏幕布局与解决方案限制**

* **屏幕分辨率要求**：支持 1024×768 及以上的分辨率。
* **布局规则**：采用标准的面板分区设计，包括顶部导航栏、左侧功能菜单、中间主要显示区域、底部状态栏。
* **屏幕切换**：在切换模块时，保持数据操作状态并实时更新显示。

**3. 用户界面组件**

**3.1 选单**

* 提供全局菜单，分为主菜单和上下文菜单。
  + 主菜单包括 **“首页”**、**“学生管理”**、**“课程管理”**、**“成绩管理”**、**“系统设置”** 等。
  + 上下文菜单根据用户权限动态生成。

**3.2 标准按钮**

* 所有操作按钮使用清晰的标签（如 **“提交”**、**“取消”**、**“返回”**）。
* 常用按钮（如“保存”）采用统一颜色（如蓝色）。

**3.3 导航链接**

* 在页面顶部或侧边提供导航链接，指向常用模块。
* 链接在鼠标悬停时有视觉提示（如颜色变化）。

**3.4 功能组件**

* 搜索框：支持模糊搜索，实时显示结果。
* 数据表格：用于展示成绩数据，支持分页、排序和筛选功能。
* 输入控件：包括文本框、下拉菜单、日期选择器等，确保易用性和容错性。

**3.5 消息栏**

* 页面顶部或弹窗形式显示操作结果。
  + 成功消息：绿色背景，明确表述成功内容。
  + 错误消息：红色背景，附带可操作的错误建议。

**4. 快捷键**

* 支持常用快捷键，提升操作效率：
  + **Ctrl + S**：保存当前编辑内容。
  + **Esc**：退出当前界面或取消操作。
  + **Ctrl + F**：激活搜索功能。

**5. 显示格式规定**

**5.1 文字对齐方式**

* 左对齐：主要用于标签、输入框说明。
* 居中对齐：用于标题、表格数据。
* 右对齐：主要用于数值或金额类数据。

**5.2 数字格式**

* 成绩显示：保留1位小数（如 89.5）。
* 学分显示：保留2位小数（如 3.00）。

**5.3 日期格式**

* 采用 **YYYY-MM-DD** 格式（如 2024-11-18）。

**5.4 时间格式**

* 采用24小时制，格式为 **HH**

（如 14:30）。

**6. 错误信息显示标准**

* 错误信息以弹窗或消息栏显示，内容包括：
  + 错误原因（如“成绩输入超出范围”）。
  + 建议解决方案（如“请输入0到100之间的分数”）。
  + 错误编号（便于技术支持跟踪）。

## 硬件接口

本节描述学生成绩管理系统与系统硬件之间的接口特征，明确支持的硬件类型、数据交互和通讯协议，以确保系统能够正确运行并与硬件设备兼容。

**1. 支持的硬件类型**

1. **客户端设备**：
   * **类型**：
     + 桌面计算机（Windows、Linux）。
     + 笔记本电脑（Windows、Linux）。
   * 支持最低硬件配置：
     + CPU：双核2.0GHz及以上。
     + 内存：4GB及以上。
     + 存储：2GB可用磁盘空间。
     + 显示器分辨率：1024×768或更高。
2. **服务器设备（若使用服务器）**：
   * **类型**：
     + 专用服务器硬件或虚拟化环境（支持Windows Server或Linux Server）。
   * 推荐最低配置：
     + CPU：四核3.0GHz及以上。
     + 内存：8GB及以上（根据并发用户数扩展）。
     + 存储：20GB及以上，用于数据库和日志存储。
     + 网络：100Mbps或更高带宽。

**2. 软、硬件之间交流的数据**

1. **客户端与硬件交互**：
   * **数据类型**：
     + 键盘和鼠标输入：用于用户登录、数据录入与查询。
     + 显示输出：包括成绩表格、图表、系统消息等，呈现在客户端界面上。
2. **服务器与硬件交互（若使用服务器）**：
   * **数据类型**：
     + 请求数据：从客户端发送的用户请求（如查询成绩、提交录入数据）。
     + 响应数据：服务器处理后返回给客户端的查询结果或操作反馈。
     + 系统日志：生成并存储在服务器硬盘上的运行记录。

**3. 控制信息的性质**

* **客户端设备控制**：
  + - 使用鼠标和键盘作为主要输入设备，无需额外的硬件控制。
    - 支持通过快捷键（如Ctrl+S、Esc）发送操作指令。
* **服务器硬件控制（若使用服务器）**：
  + - 系统管理员可通过服务器的管理工具（如SSH、远程桌面）远程控制和维护系统。
    - 支持通过控制命令（如启动、停止服务）调整系统状态。

**4. 使用的通讯协议**

1. **客户端与服务器之间的通讯协议**：
   * **协议**：基于HTTP/HTTPS协议进行数据传输，确保数据的安全性和兼容性。
   * **加密**：采用SSL/TLS加密，保护敏感数据（如登录信息、成绩数据）。
   * **数据格式**：使用JSON格式在客户端和服务器之间进行请求和响应的数据交换。
2. **服务器与数据库之间的通讯协议（若使用数据库与服务器）**：
   * **协议**：采用MySQL数据库的标准SQL协议，通过JDBC接口与数据库交互。
   * **数据安全**：支持基于角色的访问控制和数据加密存储，防止未经授权的访问。

本节的硬件接口需求旨在确保系统与所支持硬件的兼容性和高效交互，同时为开发和部署提供明确的规范。若后续需求发生变更，将对接口进行适当调整并更新文档。

## 软件接口

**1. 操作系统**

* **支持的操作系统**：
  + **客户端**：
    - Windows 7、Windows 10 及以上版本。
    - Linux 桌面环境（如 Ubuntu 20.04 及以上）。
  + **服务器**：
    - Windows Server 2016 及以上版本。
    - Linux 服务器环境（如 CentOS 7 或 Ubuntu Server 20.04）。
* **数据交换**：操作系统为系统运行提供硬件抽象层，负责文件系统访问、网络通信和进程管理。

**2. 数据库**

* **数据库组件**：
  + MySQL 5.7 或 MariaDB 10.5。
* **交互方式**：
  + 使用 JDBC（Java Database Connectivity）接口与数据库交互，支持标准 SQL 语法。
* **数据交换目的**：
  + 数据存储：存储用户信息、课程信息、成绩数据、系统日志等。
  + 数据查询：支持按条件筛选数据，如按学号查询学生成绩。
  + 数据更新：对用户操作的结果（如成绩录入）实时写入数据库。

**3. 工具**

* **开发工具**：
  + Eclipse IDE 2023 或 IntelliJ IDEA。
  + 数据库管理工具：MySQL Workbench 8.0，用于调试和管理数据库。
* **使用目的**：
  + 提供开发环境，支持代码编写、调试与测试。
  + 便捷的数据库结构可视化和数据管理。

**4. 函数库**

* **标准函数库**：
  + **Java Standard Library**（JDK 1.8）：提供基础功能支持（如 I/O 操作、集合框架）。
* **外部函数库**：
  + **Log4j 2.0**：记录系统运行日志，用于调试和错误跟踪。
  + **JSON.simple**：用于解析和生成 JSON 数据，便于客户端和服务器之间的数据交换。
* **数据交换目的**：
  + 提供稳定的编程接口，提升开发效率。
  + 支持数据格式转换、日志记录等功能。

**5. 集成的商业组件**

* **中间件**：
  + **Apache Tomcat 9.0**：作为应用服务器，用于运行系统的后端逻辑。
* **消息服务**：
  + 暂无独立消息队列服务，如需扩展支持可考虑 RabbitMQ 或 Kafka（根据后续需求）。
* **数据交换方式**：
  + HTTP/HTTPS 协议，采用 RESTful API 数据交互。

**6. 数据共享机制**

* **数据共享方式**：
  + 客户端与服务器之间的数据交互通过 JSON 格式进行，便于解析和传输。
  + 多用户访问时，通过数据库事务管理机制确保数据一致性。
  + 系统日志和错误信息通过 Log4j 实现统一的文件写入和存储，便于后续审计和排查。

本节明确了学生成绩管理系统与外部组件的接口需求，确保系统能够高效、稳定地与各软件组件集成，满足功能和性能要求。后续若发生需求变更，将及时更新接口描述。

# 系统用例设计

系统用例设计图如图4-1所示，在系统中，用户模型主要有学生用户、教师用户和管理员用户。此外，课程模型类主要存储课程基本信息。

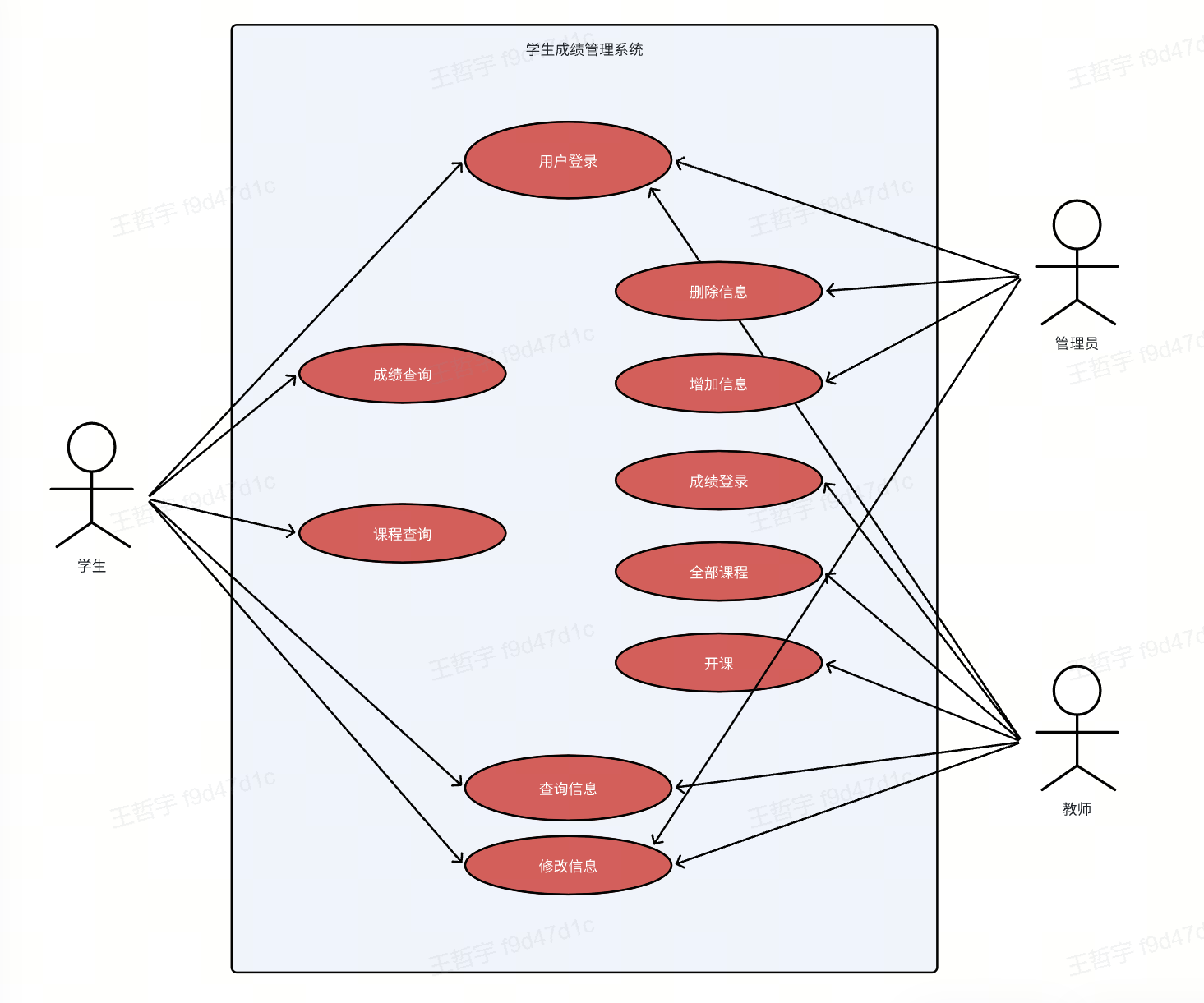


图4-1 系统用例设计UML图

# 系统功能需求

以下描述学生成绩管理系统的功能需求，确保每一项功能有明确的目标、输入/输出定义、用例描述等，以便后续开发、测试和验证。

## 输入／输出数据

* **学生用户**
* **输入数据**:
  + 数据名称：课程号  
    实际含义：查询某门课程的成绩  
    数据类型：字符串  
    数据格式：String，例如 1（代表课程号）  
    数据约束：必须为已存在的课程号
* **输出数据**:
  + 数据名称：课程成绩  
    实际含义：显示查询课程的成绩  
    数据类型：浮点型  
    数据格式：float，例如 85.0  
    数据约束：范围 [0, 100]
* **教师用户**
* **输入数据**:
  + 数据名称：学生成绩  
    实际含义：录入某课程的学生成绩  
    数据类型：浮点型  
    数据格式：float，例如 90.5  
    数据约束：范围 [0, 100]，必须对应已有学生和课程
* **输出数据**:
  + 数据名称：录入结果  
    实际含义：提示是否成功录入成绩  
    数据类型：布尔值  
    数据格式：boolean，例如 true 或 false  
    数据约束：无
* **管理员用户**
  + **输入数据**:
* 数据名称：用户信息

实际含义：添加新用户或修改现有用户信息  
数据类型：结构化数据（包含用户名、密码、角色等）  
数据格式：JSON，例如 { "id": "a3", "pwd": "123456", "role": "student" }  
数据约束：id 必须唯一

* + **输出数据**:
* 数据名称：操作结果

实际含义：提示用户操作是否成功

数据类型：布尔值

数据格式：boolean，例如 true 或 false  
数据约束：无

## 用例描述

* **用例 1：学生查询成绩**
* **用例名**: 查询成绩
* **简述**: 学生通过输入课程号查询相应课程的成绩
* **主要参与者**: 学生用户
* **情景目标**: 获取课程成绩
* **前提条件**: 学生已登录系统
* **触发事件**: 学生选择“查询成绩”功能并输入课程号
* **场景描述**:
  1. 系统显示输入框，提示学生输入课程号
  2. 学生输入课程号并提交
  3. 系统验证课程号的合法性
  4. 系统显示该课程的成绩或提示“未找到相关信息”
* **异常处理**:
  1. 如果输入为空或不合法，系统提示“请输入有效的课程号”
  2. 如果课程号不存在，系统提示“课程信息不存在”
* **优先级**: 高
* **使用频率**: 高
* **次要参与者**: 无
* **未明确的问题**: 是否需要支持模糊查询
* **用例 2：教师录入成绩**
* **用例名**: 录入成绩
* **简述**: 教师为指定课程的学生录入成绩
* **主要参与者**: 教师用户
* **情景目标**: 成功录入学生成绩
* **前提条件**: 教师已登录系统
* **触发事件**: 教师选择“录入成绩”功能并输入数据
* **场景描述**:
  1. 系统显示课程和学生信息
  2. 教师选择某学生并输入成绩
  3. 系统验证成绩数据的合法性
  4. 系统提示成绩录入成功或失败
* **异常处理**:
  1. 如果输入成绩不在 [0,100]，提示“成绩范围不合法”
  2. 如果学生或课程不存在，提示“无效的学生或课程信息”
* **优先级**: 高
* **使用频率**: 中
* **次要参与者**: 无
* **未明确的问题**: 是否需要批量录入功能
* **用例 3：管理员添加用户**
* **用例名**: 添加用户
* **简述**: 管理员为系统添加新用户
* **主要参与者**: 管理员用户
* **情景目标**: 成功添加新用户
* **前提条件**: 管理员已登录系统
* **触发事件**: 管理员选择“添加用户”功能并输入数据
* **场景描述**:
  1. 系统显示添加用户表单
  2. 管理员填写用户信息并提交
  3. 系统验证数据合法性
  4. 系统提示操作结果
* **异常处理**:
  1. 如果输入的用户名已存在，提示“用户已存在”
  2. 如果字段为空，提示“请输入完整信息”
* **优先级**: 高
* **使用频率**: 低
* **次要参与者**: 无
* **未明确的问题**: 是否需要支持批量添加用户
* **用例 4：学生查询课程**
* **用例名**: 查询课程
* **简述**: 学生通过系统查询自己选修的课程信息。
* **主要参与者**: 学生用户
* **情景目标**: 获取学生当前选修的课程信息。
* **前提条件**: 学生已登录系统。
* **触发事件**: 学生选择“课程查询”功能。
* **场景描述**:
  1. 系统显示课程查询界面。
  2. 学生点击“查询”按钮。
  3. 系统获取学生的课程信息并进行展示。
  4. 系统显示课程编号、课程名称、学分、学时等信息。
* **异常处理**:
  1. 如果学生没有选修课程，系统提示“未找到相关课程信息”。
* **优先级**: 高
* **使用频率**: 高
* **次要参与者**: 无
* **未明确的问题**: 是否需要支持自定义筛选条件（如按学期、课程类型查询）。
* **用例 5：教师开设课程**
* **用例名**: 开设课程
* **简述**: 教师通过系统添加新课程，并填写课程相关信息。
* **主要参与者**: 教师用户
* **情景目标**: 成功开设新课程。
* **前提条件**: 教师已登录系统。
* **触发事件**: 教师选择“开课”功能并输入课程信息。
* **场景描述**:
  1. 系统显示开课信息表单。
  2. 教师填写课程编号、课程名称、学分、学时、课程描述等信息。
  3. 教师提交表单。
  4. 系统验证课程信息的合法性。
  5. 系统提示课程开设成功或失败。
* **异常处理**:
  1. 如果课程编号已存在，提示“课程已存在”。
  2. 如果字段为空，提示“请输入完整信息”。
* **优先级**: 中
* **使用频率**: 中
* **次要参与者**: 无
* **未明确的问题**: 是否需要支持批量开设课程功能。
* **用例 6：管理员删除用户信息**
* **用例名**: 删除信息
* **简述**: 管理员通过系统删除指定用户的信息。
* **主要参与者**: 管理员用户
* **情景目标**: 成功删除用户信息。
* **前提条件**: 管理员已登录系统。
* **触发事件**: 管理员选择“删除信息”功能并输入用户编号。
* **场景描述**:
  + 1. 系统显示删除用户界面。
    2. 管理员输入用户编号并提交。
    3. 系统验证用户编号的合法性。
    4. 系统提示删除成功或失败。
* **异常处理**:
  + - 如果用户编号不存在，提示“该用户不存在”。
    - 如果输入为空或不合法，提示“请输入有效的用户编号”。
* **优先级**: 中
* **使用频率**: 低
* **次要参与者**: 无
* **未明确的问题**: 是否需要支持批量删除用户功能。
* **用例 7：用户修改个人信息**
* **用例名**: 修改信息
* **简述**: 学生、教师或管理员通过系统修改个人信息。
* **主要参与者**: 学生用户、教师用户、管理员用户
* **情景目标**: 更新个人信息并保存修改。
* **前提条件**: 用户已登录系统。
* **触发事件**: 用户选择“修改信息”功能并编辑数据。
* **场景描述**:
  + 1. 系统显示用户当前信息。
    2. 用户修改需要更改的字段（如姓名、性别、生日、学院、专业等）。
    3. 用户提交修改请求。
    4. 系统验证修改数据的合法性。
    5. 系统提示修改成功或失败。
* **异常处理**:
  + - 如果输入数据不合法，提示“修改信息失败，请检查输入数据”。
    - 如果某些字段为空，提示“请填写完整信息”。
* **优先级**: 高
* **使用频率**: 高
* **次要参与者**: 无
* **未明确的问题**: 是否需要支持历史记录追溯功能。
* **用例 8：管理员查看全部课程**
* **用例名**: 全部课程
* **简述**: 管理员通过系统查看所有课程的信息。
* **主要参与者**: 管理员用户
* **情景目标**: 获取所有课程的详细信息。
* **前提条件**: 管理员已登录系统。
* **触发事件**: 管理员选择“全部课程”功能。
* **场景描述**:
  + 1. 系统显示所有课程的信息列表。
    2. 管理员可查看课程编号、课程名称、学分、学时、教师信息等。
* **异常处理**:
  + - 如果没有课程信息，提示“暂无课程信息”。
* **优先级**: 中
* **使用频率**: 中
* **次要参与者**: 无
* **未明确的问题**: 是否需要支持按筛选条件查询课程。

# 其它非功能需求

在这里列举出所有非功能需求，主要包括可靠性、安全性、可维护性、可扩展性、可测试性等。

## 性能需求

* **用户数量**

系统应支持同时在线的用户数量：学生 500 人，教师 100 人，管理员 20 人。

* **并发操作**  
  系统应支持至少 50 个并发操作而不会显著影响响应时间。
* **响应时间**
* 学生成绩查询：响应时间 ≤ 2 秒。
* 教师成绩录入：响应时间 ≤ 3 秒。
* 管理员用户管理操作：响应时间 ≤ 5 秒。
* **实时系统关系**  
  系统非严格实时系统，但需保证操作完成后实时反馈操作结果。
* **容量需求**
* **存储器**：服务器需具备至少 8GB 内存，支持高效处理多个请求。
* **磁盘空间**：初始阶段磁盘需求 ≥ 100GB，可扩展至 500GB。
* **数据库容量**：表设计需支持以下最大容量：
  + 学生表：100,000 行。
  + 教师表：10,000 行。
  + 成绩表：1,000,000 行。

## 安全措施需求

* 系统应限制单用户连续错误登录次数不超过 5 次，超过后锁定账户 15 分钟。
* 重要操作（如新增用户、修改用户信息）需管理员二次认证。
* 学生成绩等敏感数据需加密存储，采用 AES256 算法。
* 系统需对登录、操作等关键事件进行日志记录，保存期限为 6 个月。

## 安全性需求

* **用户身份认证**

每个用户在第一次登录后，必须更改默认密码，且默认密码不能重用。密码应符合以下策略：长度 ≥ 8，包含字母、数字和特殊字符。

* **授权机制**

不同角色用户只能访问特定模块：

* 学生：查询个人信息和成绩。
* 教师：管理课程、录入成绩。
* 管理员：用户管理、课程管理。
* **数据完整性**

操作必须验证合法性（如，学生成绩录入前需确认学生、课程的存在）。

* **隐私保护**

学生和教师的个人信息仅限特定角色访问，防止数据泄露。

## 软件质量属性

* **可靠性**

系统正常运行时间应达到 99.9%，平均故障恢复时间 ≤ 10 分钟。

* **可维护性**  
  系统需模块化设计，支持后续功能扩展和错误修复，平均修复时间（MTTR）≤ 2 小时。
* **可扩展性**
* 数据库设计需支持动态扩展。
* 前端和后端模块设计需允许后续功能的插件式扩展。
* **可测试性**  
  每个功能需求需至少对应一个测试用例，系统测试覆盖率 ≥ 95%。

## 业务规则

* 查询学生成绩时，仅允许学生本人或相关教师查询，不得跨角色访问。
* 教师录入成绩时，需核对学生身份和课程信息，录入后成绩不可直接修改，仅可通过管理员审核更改。
* 每位管理员每日最多新增 50 个用户账号。

## 用户文档

将与系统一同交付以下用户文档：

* **安装指南**  
  格式：纸质文档，16 开本。内容包括系统部署的详细步骤、硬件和软件要求。
* **用户手册**  
  格式：纸质文档，16 开本。内容包括学生、教师和管理员的系统操作说明。
* **在线帮助**  
  格式：嵌入式 HTML 文档，与系统 UI 集成，提供关键功能操作的实时指导。
* **使用教程**  
  格式：电子文档（PDF），与软件一同分发，包含图文并茂的操作案例和 FAQ。

# 词汇表

**业务层面专业术语**

1. **学生用户**  
   指在校学生，通过系统登录后可查询个人信息和成绩，并维护其个人基本信息。
2. **教师用户**  
   指任课教师，通过系统登录后可管理所教授课程的信息，录入和查询学生成绩。
3. **管理员用户**  
   指教务管理员，负责管理系统中的用户信息，包括增加、修改和删除用户账号。
4. **课程信息**  
   包括课程编号、课程名称、学分、学时数、任课教师编号及姓名等课程相关的基本数据。
5. **成绩信息**  
   每位学生在某门课程中的最终得分，范围一般为 0 到 100。
6. **角色**  
   系统中用户分为三种角色：学生、教师和管理员，不同角色具有不同的权限。
7. **分段统计**  
   根据成绩划分为多个等级（如优秀、良好、及格、不及格）的统计方式。
8. **用户认证**  
   确认登录用户身份的过程，包括用户名和密码验证。
9. **权限管理**  
   系统对不同用户角色的功能和数据访问范围的控制机制。

**软件与计算机专业术语**

1. **并发操作**  
   系统同时处理多个用户请求的能力，通常表示为同时执行的事务或操作数。
2. **响应时间**  
   用户请求到系统反馈结果的时间间隔，用于衡量系统性能。
3. **C/S 架构**  
   英文原词：Client/Server Architecture  
   客户端与服务器之间以请求-响应方式通信的一种系统架构模式。
4. **MVC 架构**  
   英文原词：Model-View-Controller  
   一种常见的软件设计模式，分为模型（Model）、视图（View）和控制器（Controller）三部分，各部分职责明确，提高系统的可维护性和可扩展性。
5. **AES**  
   英文原词：Advanced Encryption Standard  
   一种对称加密算法，用于数据的高安全性加密。
6. **日志记录**  
   系统自动保存操作事件的行为，如用户登录、数据变更等，用于安全审计和问题排查。
7. **测试覆盖率**  
   衡量测试用例对系统代码的覆盖程度，通常以百分比表示。
8. **MTTR**  
   英文原词：Mean Time to Repair  
   平均修复时间，用于描述系统从故障到恢复的时间。

**缩写及定义**

1. **UI**  
   英文原词：User Interface  
   用户界面，系统与用户交互的界面部分。
2. **GBK**  
   英文原词：GuoBiao Kuozhan  
   一种汉字编码标准，扩展了国标码以支持更多字符。
3. **PDF**  
   英文原词：Portable Document Format  
   一种用于电子文档发布的格式，支持跨平台阅读。
4. **FAQ**  
   英文原词：Frequently Asked Questions  
   常见问题解答，提供用户快速获取问题答案的方式。
5. **AST**  
   英文原词：Abstract Syntax Tree  
   抽象语法树，用于表示代码的结构化表示（仅适用于扩展开发需求的用户）。

# 数据定义

以下为系统使用的主要数据项及其定义，按照原数据元素、选择项、组合项和重复项分类描述：

**8.1 原数据元素**

* **学号**（Student\_ID） = *学生的唯一标识符，用于区分不同学生。类型：字符串，大小：10，格式：XXXXXXX（数字），范围：[000000001, 999999999]*
* **姓名**（Name） = *用户的姓名，类型：字符串，大小：20，格式：UTF-8 编码，无固定范围*
* **性别**（Gender） = *用户性别，类型：字符串，大小：1，格式："男"/"女"*
* **出生日期**（DOB） = *用户的出生年月，类型：日期，大小：10，格式：YYYY-MM-DD，范围：[1900-01-01, 当前日期]*
* **学院**（Department） = *所属学院的名称，类型：字符串，大小：50，无固定范围*
* **专业**（Major） = *学生或教师的专业名称，类型：字符串，大小：50，无固定范围*
* **课程编号**（Course\_ID） = *课程的唯一标识符，用于区分不同课程，类型：字符串，大小：5，格式：XXXXX，范围：[00001, 99999]*
* **课程名称**（Course\_Name） = *课程的名称，类型：字符串，大小：50，无固定范围*
* **学分**（Credit） = *课程的学分值，类型：浮点型，大小：4，格式：#.0，范围：[0.0, 10.0]*
* **成绩**（Grade） = *学生某门课程的成绩，类型：浮点型，大小：4，格式：##.#，范围：[0.0, 100.0]*
* **用户角色**（User\_Role） = *用户在系统中的角色，类型：字符串，大小：10，范围：["学生", "教师", "管理员"]*
* **密码**（Password） = *用户登录密码，类型：字符串，大小：16，格式：任意字符，必须符合复杂性规则*
* **登录尝试次数**（Login\_Attempts） = *用户连续登录失败的次数，类型：整数，大小：1，范围：[0, 5]*

**8.2 选择项**

* **性别**（Gender） = ["男" | "女"]
* **用户角色**（User\_Role） = ["学生" | "教师" | "管理员"]
* **成绩等级**（Grade\_Level） = ["优秀" | "良好" | "及格" | "不及格"]

**8.3 组合项**

* **学生信息**（Student\_Info） = 学号 + 姓名 + 性别 + 出生日期 + 学院 + 专业
* **教师信息**（Teacher\_Info） = 职工编号 + 姓名 + 性别 + 出生日期 + 学院 + 专业
* **课程信息**（Course\_Info） = 课程编号 + 课程名称 + 学分 + 任课教师编号
* **成绩信息**（Grade\_Info） = 课程编号 + 学生学号 + 成绩

**8.4 重复项**

* **学生成绩列表**（Student\_Grades\_List） = {成绩信息}，最小值：1，最大值：所有所修课程的成绩记录
* **课程学生列表**（Course\_Students\_List） = {学生学号}，最小值：0，最大值：课程容量
* **学院列表**（Department\_List） = {学院}，无范围限制

**数据定义说明**

* 数据名称统一采用中英文命名，英文名称符合常见命名规范，便于代码开发和维护。
* 每个数据项的取值范围和格式定义旨在减少数据冗余和输入错误，同时提高数据的可读性和一致性。
* 以上数据定义为系统核心数据模型的基础，后续开发过程中，可根据实际需求动态扩展或细化。

# 分析模型

整个系统的类图设计如图9-1所示。



图9-1 项目类图

# 待定问题列表

以下是软件产品需求分析报告中尚未确定的问题列表，每项问题均附带编号以便后续跟踪与调查：

| **编号** | **问题描述** | **涉及功能模块** | **当前状态** | **解决负责人** | **预计解决时间** |
| --- | --- | --- | --- | --- | --- |
| 1 | 是否需要支持学生查询所有历史成绩，或仅查询当前学期成绩？ | 学生成绩查询模块 | 待确认 | 产品需求负责人 | YYYY-MM-DD |
| 2 | 教师是否需要批量录入学生成绩的功能？ | 教师成绩录入模块 | 待确认 | 系统设计负责人 | YYYY-MM-DD |
| 3 | 管理员是否需要提供批量导入用户数据的功能（如 CSV 文件格式）？ | 管理员用户管理模块 | 待确认 | 产品需求负责人 | YYYY-MM-DD |
| 4 | 是否需要对学生成绩设定分类阈值（如优秀分数线由用户自定义）？ | 分段统计功能模块 | 待确认 | 产品需求负责人 | YYYY-MM-DD |
| 5 | 日志保存的默认周期是多久？可否由管理员配置？ | 系统日志管理模块 | 待确认 | 系统设计负责人 | YYYY-MM-DD |
| 6 | 密码复杂性规则的具体要求如何设定？（如必须包含哪些字符类型） | 用户身份认证模块 | 待确认 | 安全设计负责人 | YYYY-MM-DD |
| 7 | 用户登录失败超过限制后的解锁方式（自动解锁还是管理员解锁）？ | 用户身份认证模块 | 待确认 | 产品需求负责人 | YYYY-MM-DD |
| 8 | 是否需要支持手机或电子邮件作为辅助验证方式？ | 用户身份认证模块 | 待确认 | 系统设计负责人 | YYYY-MM-DD |
| 9 | 成绩录入后是否允许教师修改？修改权限和审批流程如何定义？ | 教师成绩录入模块 | 待确认 | 产品需求负责人 | YYYY-MM-DD |
| 10 | 是否需要支持按课程或学期对学生成绩的导出（如 PDF 格式）？ | 学生成绩查询模块 | 待确认 | 产品需求负责人 | YYYY-MM-DD |
| 11 | 系统是否需要为操作密集时段（如期末成绩录入）优化性能？ | 系统性能模块 | 待确认 | 系统架构负责人 | YYYY-MM-DD |
| 12 | 是否需要多语言支持（如英文界面）？ | 用户界面模块 | 待确认 | 产品需求负责人 | YYYY-MM-DD |
| 13 | 数据存储是否需要支持多种数据库（如 MySQL 和 PostgreSQL）？ | 数据库模块 | 待确认 | 数据库设计负责人 | YYYY-MM-DD |
| 14 | 是否需要记录用户操作历史（如修改记录）供审计使用？ | 系统日志管理模块 | 待确认 | 安全设计负责人 | YYYY-MM-DD |
| 15 | 是否对管理员操作的频率限制做出特别要求？ | 管理员用户管理模块 | 待确认 | 产品需求负责人 | YYYY-MM-DD |

**跟踪与更新**

* **状态说明**：
  + - * **待确认**：问题尚未解决，需要进一步讨论或确认。
      * **处理中**：问题已进入解决流程。
      * **已解决**：问题已完全解决，并记录解决方案。
* 每次更新此列表时，应注明变更时间和修改者，确保版本控制。