# 高考成绩管理系统需求分析

1. **产品分析**

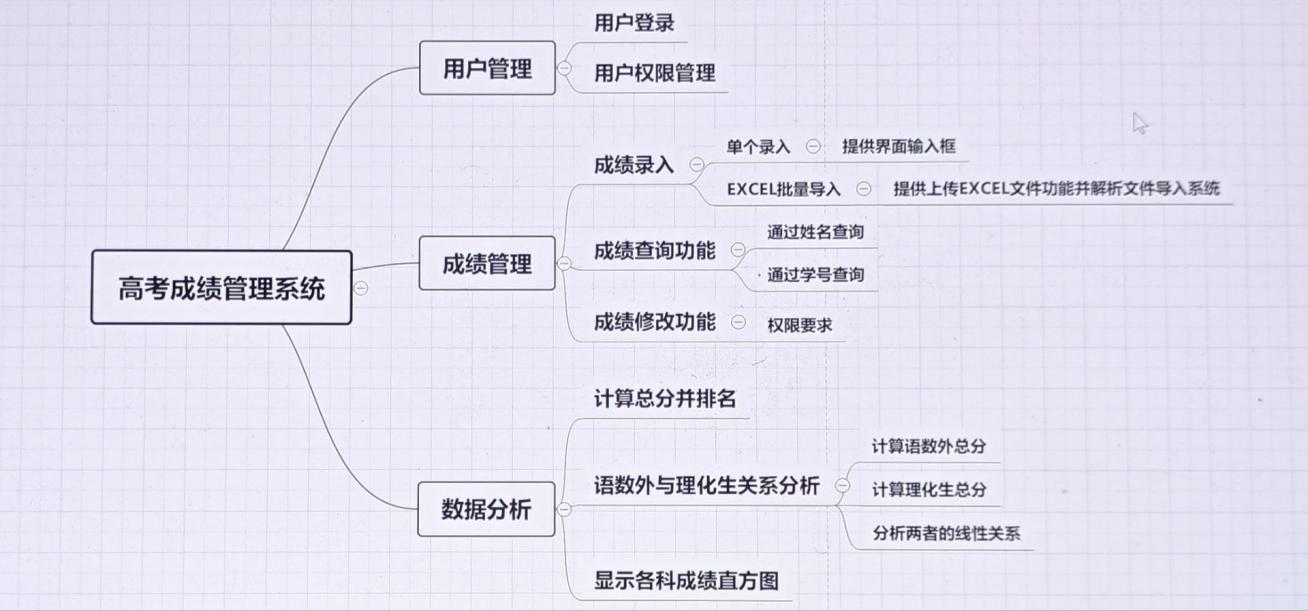
## 1.目标

高考成绩管理系统旨在通过高效、安全、用户友好的方式，帮助学校或教育机构全面管理和分析学生的高考成绩数据。该系统支持用户登录、成绩的便捷录入、成绩的精确查询和灵活修改、总分排序及排名分析、学科间线性关系分析及单科成绩可视化展示，以辅助教学管理、成绩分析和决策支持。

## 2.总体流程

1. 教师、管理员或学生各自登入系统；
2. 教师或管理员录入学生成绩，支持单个成绩录入和批量EXCEL导入；
3. 教师或管理员可对学生的成绩数据进行修改；
4. 教师、管理员、学生可以通过姓名、学号等信息查询到高考成绩；
5. 系统计算所有学生的总分并进行排名；分析语数外和理化生之间的线性关系；
6. 显示该学生每一门考试科目的成绩，显示为直方图。

## 3.功能摘要

3.1产品结构图

3.2功能列表

用户登录；用户权限管理；录入成绩；查询成绩；修改成绩；总分分析；语数外和理化生线性关系分析；可视化显示学科成绩直方图。

# 二、功能需求说明

## 1.数据库设计与管理

1.1模块概述：

本模块负责系统的数据库设计和管理，包括创建数据库模式、实现数据库操作、保障数据安全和备份。

1.2功能需求：

（1）设计数据库模式:创建学生表，包括字段：学生ID、姓名、学号、班级、联系方式等。创建成绩表，包括字段：成绩ID、学生ID、科目、分数、考试日期等。创建科目表，包括字段：科目ID、科目名称、科目类型等。

（2）数据库操作:实现CRUD操作：创建（Create）、读取（Read）、更新（Update）、删除（Delete）。具有数据完整性约束：外键约束、非空约束、唯一约束等。

(3)安全与备份:实现数据加密,对敏感数据进行加密存储。实现用户认证与权限管理,不同用户角色（如管理员、教师）具有不同的权限。实现数据备份与恢复：定期备份数据，提供数据恢复功能。

(4)技术选型:数据库管理系统选择MySQL，开发语言选择使用SQL进行数据库操作，使用C++与MySQL Connector进行数据库交互。

## 2. GUI模块开发设计

2.1模块概述：

GUI开发模块负责构建高考成绩管理系统的用户界面，确保用户能够方便地录入、查询、修改数据，并通过直观的数据可视化了解成绩分布情况。本模块将采用qt开发技术

2.2功能需求：

(1)可访问性：提供清晰的标签、可聚焦的元素,设计清晰直观的网页界面元素，便于用户快速的找到需要使用的功能，提高用户体验感。

(2)总分分析：将总分排序后的排名进行展示。

(3)学科关系分析：设计并实现一个图表，展示语数外总分与物化生总分之间的线性关系，图表应能够清晰得展示出两者的关系，正相关或者负相关。

(4)各科成绩分析：使用图表库为每个科目生成成绩直方图，支持用户选择不同科目查看相应的直方图，直方图展示成绩的分布情况。

(5)反馈机制：在用户执行操作时提供相应的及时反馈，在发现错误时提供明确的错误指示，指出问题的具体位置并提出解决方案及建议。

## 3.后端开发模块

## 3.1 模块概述：

本模块负责开发服务器端逻辑和API，实现系统的核心功能和业务逻辑，确保数据得到安全和高效的处理。

3.2 功能需求：

API开发

（1）开发RESTful API：提供成绩录入、查询、修改、删除等接口。

（2）API文档：详细说明每个API的请求方法、参数、返回值。

例如：成绩录入API： 请求方法：POST URL：/api/scores

请求参数：{ "student\_id": "12345", "subject": "Math", "score": 95 }

返回值：{ "status": "success", "message": "Score recorded successfully" }

业务逻辑实现

（1）成绩计算逻辑：

实现总分计算：根据各科成绩计算总分。科目间的线性关系分析：分析语数外和物化生/史地政成绩之间的关联性。使用合适的线性回归算法建立线性回归模型，对每个对象进行计算，得到散点图和回归线等表征线性关系的结果。

（2）分数直方图：使用相关的Data Visualization技术对每一门考试科目的成绩分布进行可视化展示，生成成绩直方图。

（3）数据验证：确保输入数据的合法性和正确性，例如:学生ID是否存在。成绩是否在合理范围内（0-100）。科目是否在系统支持的科目列表中。

（4）错误处理与日志记录

错误处理机制：处理异常情况，返回友好的错误信息。例如：当学生ID不存在时，返回错误信息：“Student ID not found”。

当输入成绩不合法时，返回错误信息：“Invalid score value”。

日志记录：记录系统操作日志，便于问题追踪和分析。例如：

用户进行成绩录入操作时，记录操作时间、用户ID、输入数据等信息。

3.3技术选型

1. 开发语言：使用C++进行后端开发。
2. API框架：选择合适的C++框架（如<cpprestsdk>）进行RESTful API开发。
3. <cpprestsdk>的优点：支持异步操作、高性能、跨平台。

### 3.4业务逻辑流程图

### 高考成绩管理系统：流程图

3.5 附录

- RESTful API：一种基于REST架构风格的API设计方法。

- OAuth2.0：一种开放标准，用于授权。

- <cpprestsdk>：一个开源C++库，用于创建跨平台的、基于HTTP的网络应用程序和RESTful服务。

## 4.数据分析模块：

4.1模块概述：

对预录入的数据进行预处理，剔除异常值，确保数据分析的准确性。设计并实现高效的统计算法用于统计学生总分并进行排序。

开发算法以计算科目间的线性关系，指学生物理，化学，生物三科总分（简称“物化生”）与语文，数学，外语三科总分（简称“语数外”）的线性关系。

设计合适的算法对各科成绩进行统计与排序。确保算法的高效与稳定性，能够完成大规模数据的统计与分析任务。

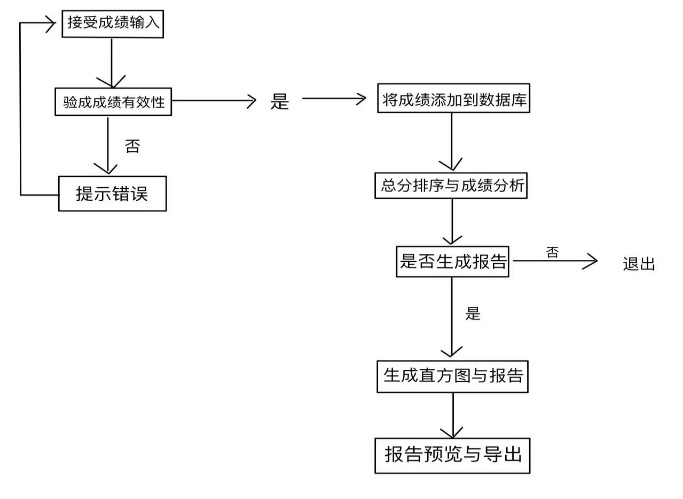
4.2 报告生成

（1）设计各科成绩分布直方图，用于展示科目成绩分布信息。利用图表库实现图表的生成与绘制，确保图表直观清晰。

（2）设计结构化的文本报告模板，包含学生成绩概述，统计分析语数外与物化生分数线性关系的结果，各科成绩分布结果。确保报告格式规范，结果清晰。

（3）根据统计分析结果与生成的可视化图表，实现自动化生成完整报告文件。并同时提供报告预览与导出功能，支持将报告保存为PDF格式，便于分享与存档。

（4）顺序图：



# 三、非功能性需求

## 1.产品需求-效率需求

1.1概述：

系统的效率性决定了系统在各种负载下的响应速度和稳定性，确保用户在各种环境下都能流畅操作。

1.2具体需求：

<1>响应时间：用户在进行成绩查询、录入、修改等操作时，系统应迅速响应。

<2>数据处理能力：系统应能高效处理大批量数据，如批量导入EXCEL文件中的成绩数据，并较快完成处理。

<3>内存空间管理：系统需要尽可能提高对内存空间的利用率，完善内存回收机制，避免占用过多系统资源。

**2.约束性需求**

2.1概述：

约束性需求确保了软件项目计划和进展的有效性、可行性和质量，使软件开发的时间和进程得到有效保障。

2.2具体需求：

（1）交付进度：7月5日完成需求分析和概要设计文档，7月14日之前完成详细设计和编码，7月17日完成迭代和验收。

（2）技术选型：以c++作为实现语言，用qt完成前端设计，数据库选用mysql，并部署在云端。

**3.易用性需求**

3.1概述：

系统界面需简洁直观，操作流程简便，确保用户（如教师、管理员）能轻松上手使用。

3.2具体要求：

(1)GUI要求：GUI各个页面之间功能划分明确，跳转逻辑合理。降低用户的学习成本，提升使用体验。

(2)逻辑要求：各个子功能之间逻辑结构清晰，确保流程简便，便于用户上手操作。各个用户有不同的图形界面，权限控制合理。

# 四、设计概要

## 1.模块分解思路

(1)数据库设计：负责数据存储与管理，确保数据完整性、准确性和安全性；

(2)后端开发：负责服务器端得业务逻辑处理和API接口开发；

(3)前端开发：负责用户界面得设计与实现，确保系统可用性和用户体验；

(4)数据分析：负责对成绩数据得分析和直方图、线性关系的生成，提供分析结果和决策支持。

## 2.设计框图