# 模块2：后端开发

## 1. 模块名称

后端开发

## 2. 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. 功能需求

### 3.1 API开发

### 开发RESTful API：提供成绩录入、查询、修改、删除等接口。

### **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" }

**3.2 业务逻辑实现**

**成绩计算逻辑：**

1. 实现总分计算：根据各科成绩计算总分。
2. 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。
3. 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。

**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。

### 3.3 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。

## 4. 非功能需求

### 4.1 性能要求：API响应时间短，能够处理高并发请求。

例如：在高峰期每秒钟能够处理5000次请求，确保响应时间在200ms以内。

**4.2 安全性要求：**确保API安全，防止未授权访问和数据泄露。例如：

1. 使用OAuth2.0进行用户认证。

2. 对敏感数据进行加密存储。

3. 防止SQL注入和XSS攻击。

**4.3 可靠性要求：**系统稳定可靠，支持7x24小时运行。例如：

1. 使用负载均衡技术，确保服务器高可用。

2. 实现故障自动切换，确保系统在任何时候都能正常运行。

## 5. 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。

## 6. 设计文档

### 6.1 API设计文档

### 包含每个API的详细设计，包括请求方法、URL、参数、返回值、示例等，以便协同开发。

### 6.2 业务逻辑流程图

### 高考成绩管理系统：流程图

## 7. 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。

# 设计概要

## 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。