概要设计说明书

1. **引言**
   1. 编写目的

本文档旨在说明学生成绩管理系统的设计考虑，为后续详细设计和开发实现提供基础。

**1.2** 背景

随着教育信息化的发展,传统的人工管理方式已经无法满足现代教育管理的需求。本系统旨在为教育机构提供一个全面的成绩管理解决方案

**1.3**定义

1.用户类型：管理员、教师、学生

2.RBAC：基于角色的访问控制

**1.4**参考资料

1.Vue 3 官方文档

2.Element Plus 组件库文档

3.Spring Boot 官方文档

**1.5**系统目标

1.实现教务管理信息化、规范化

2.提高成绩管理效率

3.为教学决策提供数据支持

4.提升师生使用体验

**2.** 系统架构设计

**2.1** 技术架构

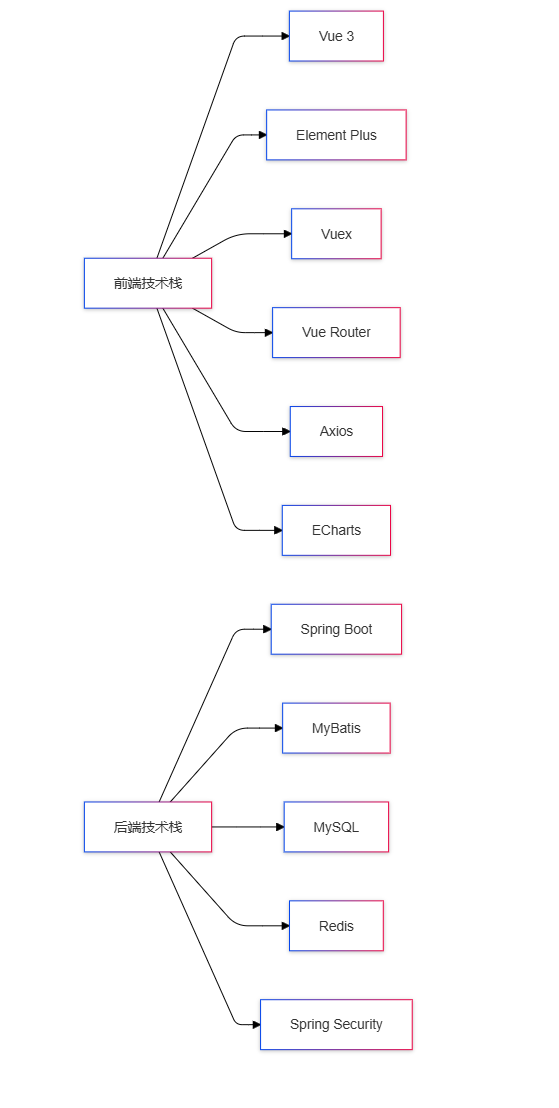


图2.1 技术架构图

**2.2** 系统架构

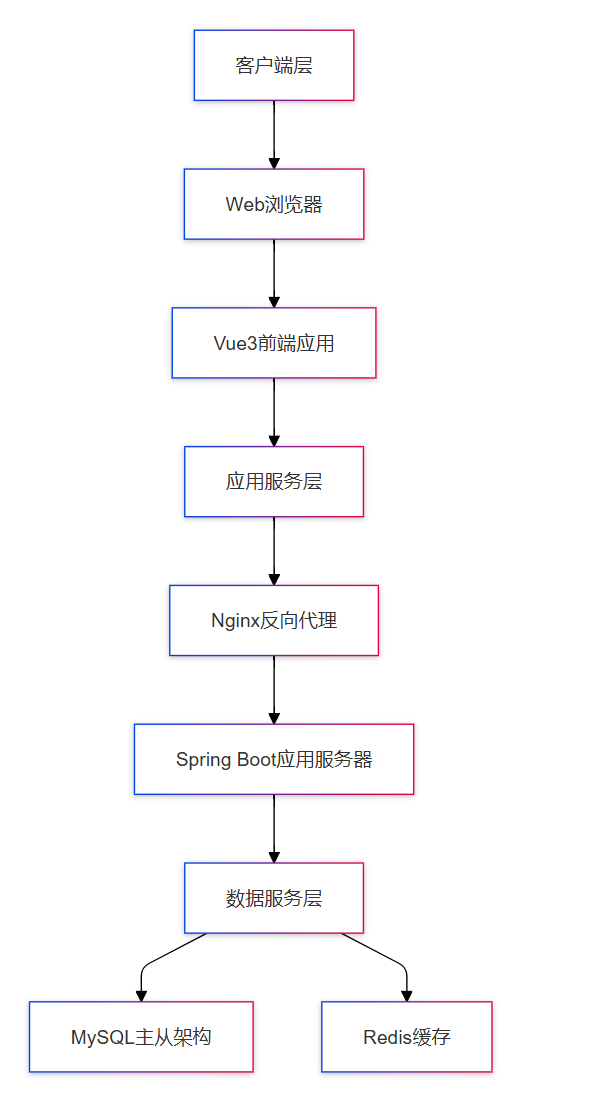


图2.2.1系统架构图

图示

描述已自动生成

图2.2.2数据层

图形用户界面, 应用程序

描述已自动生成

图2.2.3业余层

图形用户界面, 文本, 应用程序

描述已自动生成图2.2.4表示层

图示

描述已自动生成

图2.2.5用户层

图表, 箱线图

描述已自动生成

图2.2.6执行逻辑

**3**. 功能模块设计

**3.1** 用户认证模块

**3.1.1** 登录功能

1.多角色登录(管理员/教师/学生)

2.验证码校验

3.记住密码

4.自动登出

**3.1.2** 权限控制

1.基于RBAC的权限模型

2.动态路由加载

3.按钮级权限控制

4.数据权限控制

**3.2** 教务管理模块

**3.2.1** 课程管理

1.课程信息维护

2.课程分类管理

3.教学计划管理

4.选课管理

**3.2.2** 班级管理

1.班级信息维护

2.班级学生管理

3.班主任管理

4.班级课表管理

**3.3** 成绩管理模块

**3.3.1** 成绩录入

1.平时成绩录入

2.期末成绩录入

3.成绩导入导出

4.成绩修改审核

**3.3.2** 成绩统计分析

1.成绩分布统计

2.成绩趋势分析

3.班级排名

4.学科分析报告

3.4 **功能需求与模块关系**

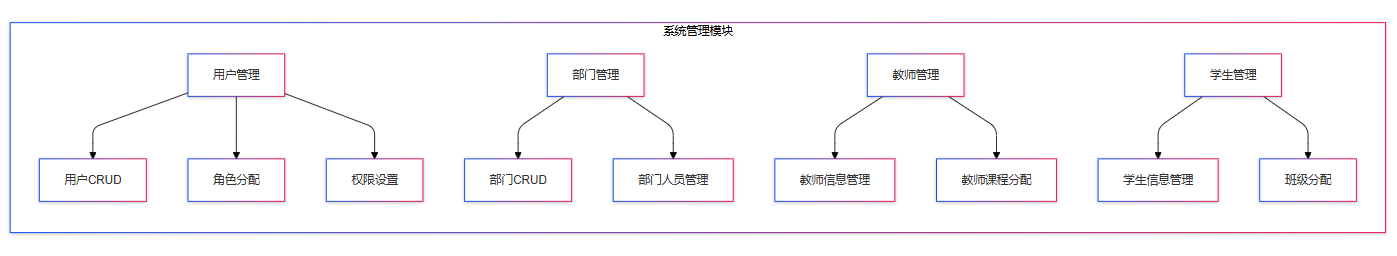


图3.4.1功能模块映射图

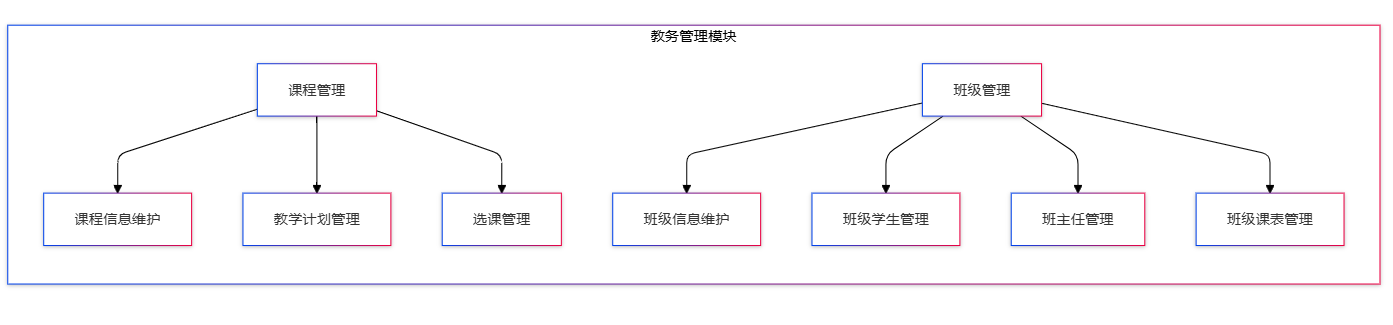


图3.4.2教务管理功能关系图

图示

描述已自动生成

图3.4.3成绩管理功能关系图

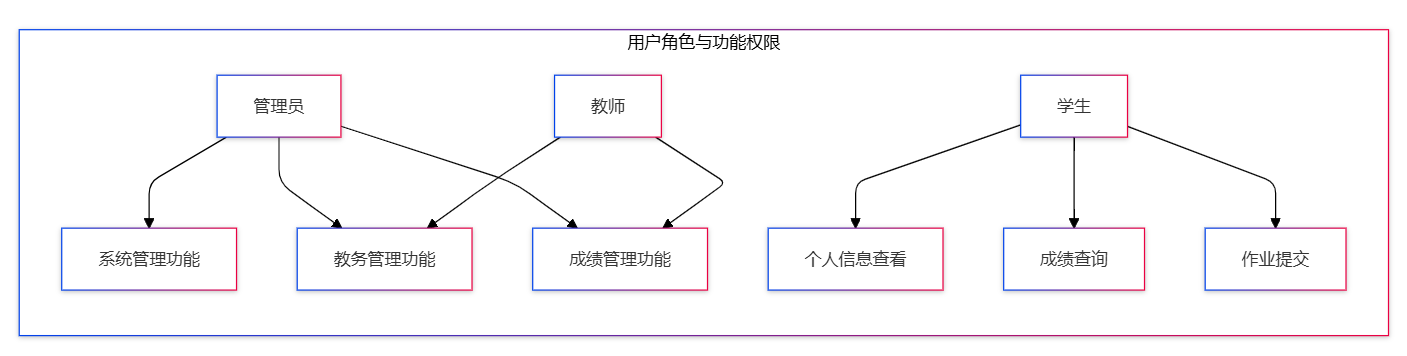


图3.4.4功能与用户角色对应关系图

图示

描述已自动生成

图3.4.5功能依赖关系图

**4**数据库设计

**4.1** 数据库表结构

1. 用户表(user)

2. 部门表(department)

3. 课程表(course)

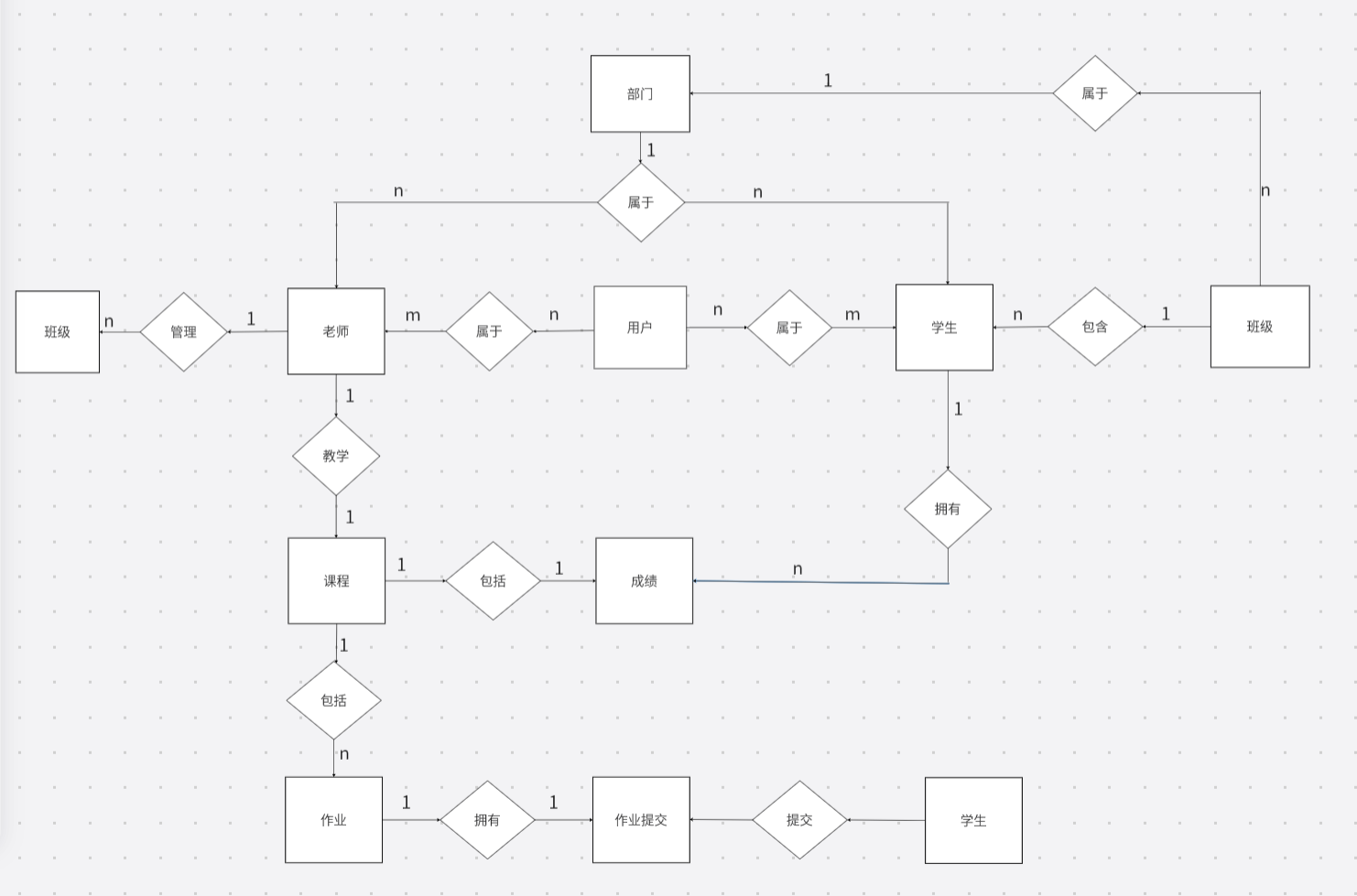
4. 班级表(class)

5. 学生表(student)

6. 教师表(teacher)

7. 成绩表(score)

8. 作业表(homework)



**4.2** 数据库优化

1.索引优化

2.分表策略

3.读写分离

4.缓存设计

**5.**权限设计

**5.1** 权限模型

角色(Role) ←→ 权限(Permission)

↑ ↑

用户(User) → 资源(Resource)

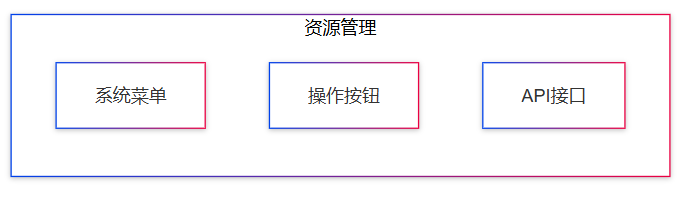


图5.1.1资源管理

文本

描述已自动生成

图5.1.2权限管理

图片包含 文本

描述已自动生成

图5.1.3角色管理

图示

中度可信度描述已自动生成

图5.1.4用户管理

**5.2**权限控制

系统采用基于角色的访问控制(RBAC):

1. 管理员

拥有所有模块的操作权限

可以管理用户、部门等系统配置

2. 教师

可以管理课程信息

可以录入和管理成绩

可以布置和批改作业

3. 学生

可以查看个人课程信息

可以查看个人成绩

可以提交作业

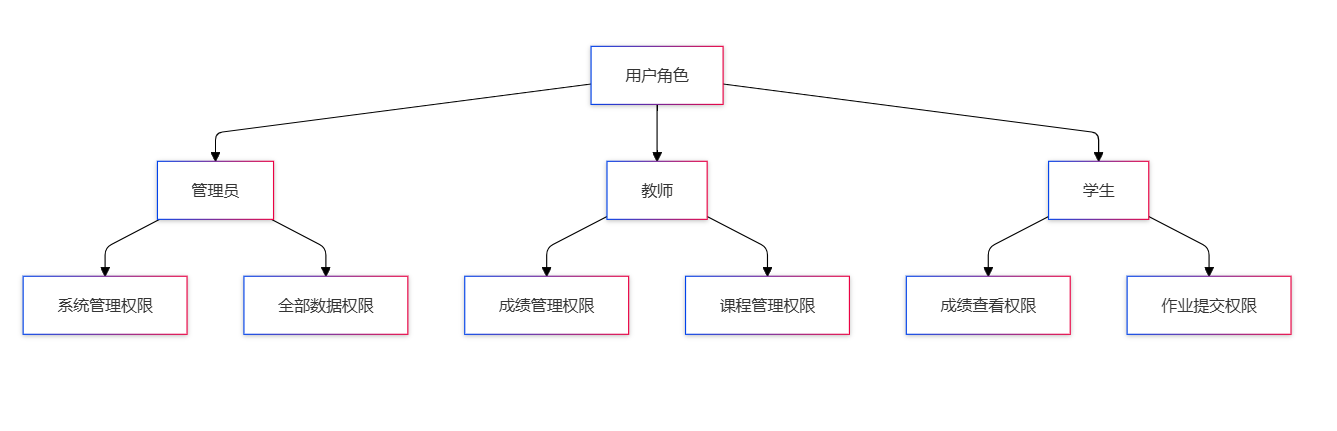


图5.2用户角色权限图

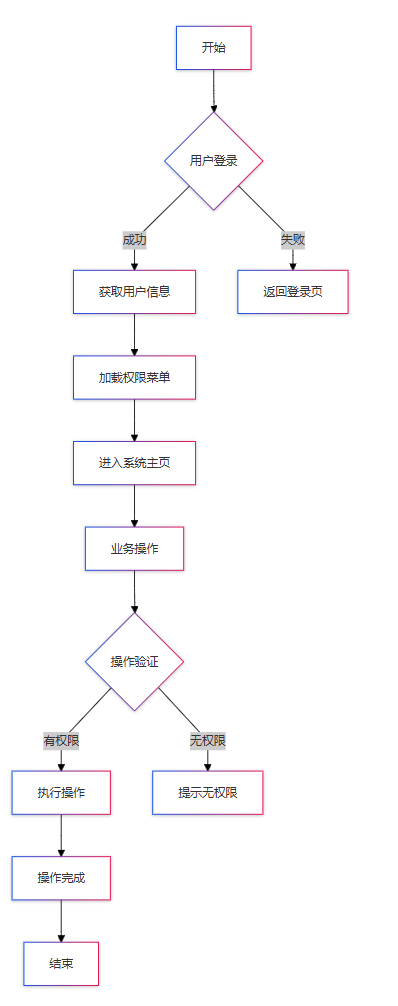


图5.3核心业务流程图

**6** 界面与接口设计

**6.1**界面设计

系统采用 Element Plus 组件库,整体风格简洁现代。主要包括:

1. 登录界面

2. 系统主界面

顶部导航栏

左侧菜单栏

内容区域

3. 各功能模块界面

列表页

表单页

详情页

**6.2**接口规范

统一响应格式

接口版本控制

接口文档规范

**6.3** 核心接口

**6.3.1**内部接口

1.用户认证接口

POST /api/auth/login

POST /api/auth/logout

2.成绩管理接口

GET /api/score/list

POST /api/score/save

GET /api/score/statistics

**6.3.2**外部接口

1.文件导入导出接口

2.数据库连接接口

3. 缓存服务接口

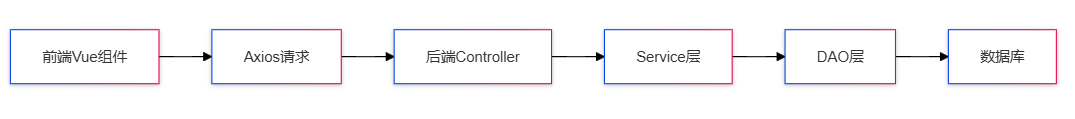


图6.3前后端接口关系图

**7** 安全设计

**7.1**安全认证

1. 身份认证

基于 Token 的用户认证

密码加密存储

2. 权限控制

基于角色的菜单权限

基于角色的接口权限

3. 数据安全

敏感数据加密

SQL注入防护

XSS防护

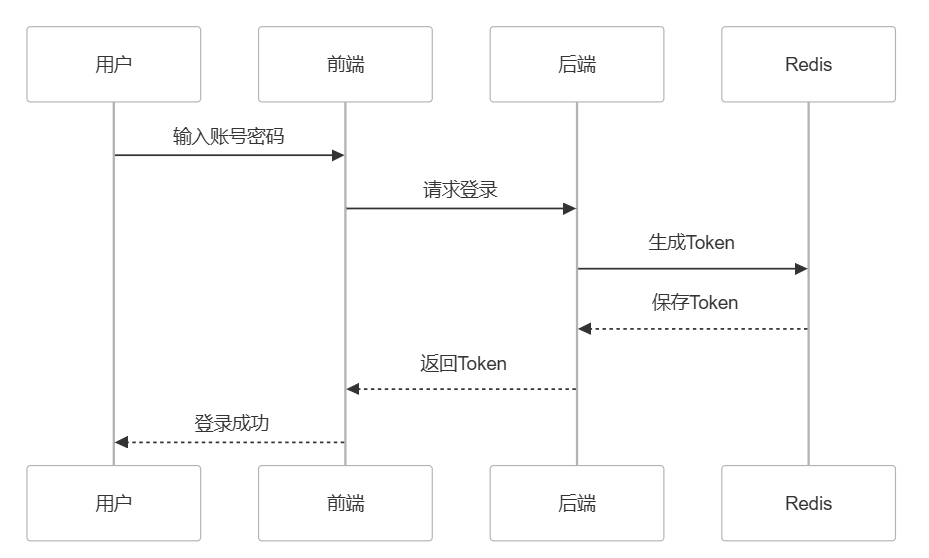


图7.1认证授权流程图

**7.2**系统出错处理设计

**7.2.1**出错信息

系统统一的错误处理机制，包括：

表单验证错误

接口调用错误

权限验证错误

**7.2.2**补救措施

数据备份机制

操作日志记录

事务回滚

**7.2.3**系统维护设计

系统监控

性能优化

定期备份

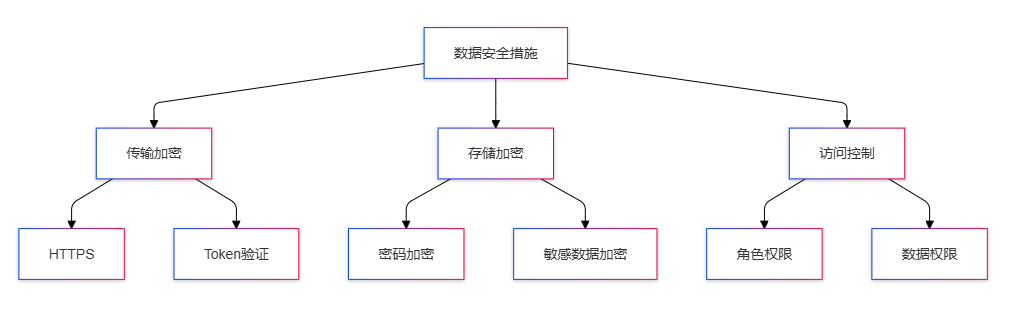


图7.3数据安全保护图

8 部署方案

8.1. 开发环境与生产环境

- Node.js 16+

- Vue 3

- Vite

- Git

-Springboot

-Mybaitis

-Maven

- Nginx

- Node.js

- MySQL

- Redis

**8.2**部署架构

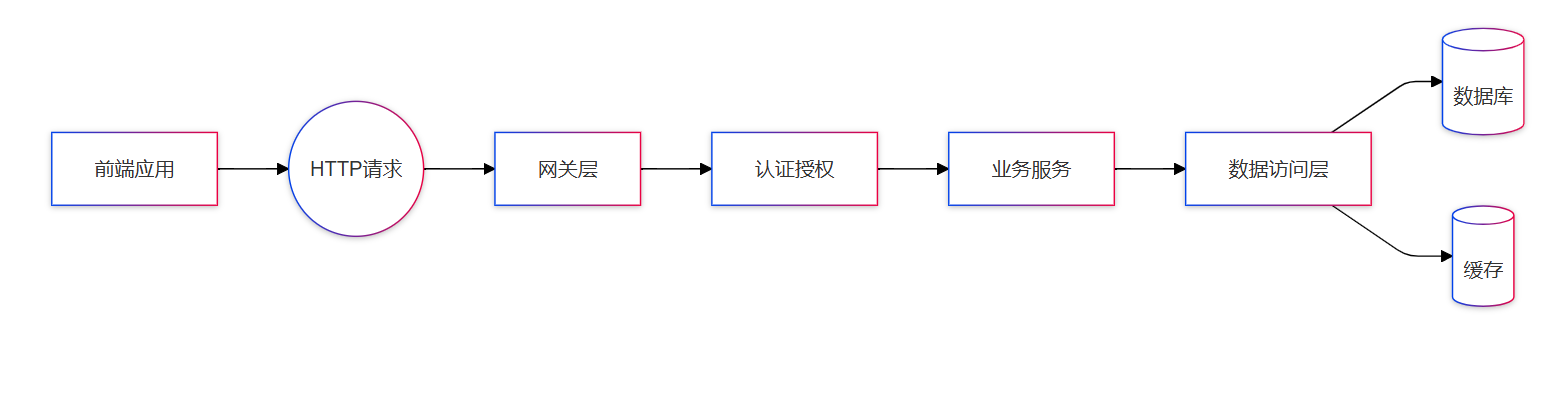


图8.2部署架构图

8.3 性能优化

1.前端优化

- 路由懒加载

- 组件按需加载

- 静态资源CDN

2.后端优化

- 接口缓存

- 数据库优化

- JVM调优

9. 项目进度规划

9.1 开发周期

需求分析: 1周

概要设计: 1周

详细设计: 1周

测试部署: 1周

9.2 里程碑计划

1. 完成基础框架搭建

2. 完成核心功能开发

3. 完成系统联调测试

4. 完成系统上线部署