# 详细设计报告

课题名称：在线考试系统

班级：软件工程2144

成员： 龚圆康 21040147

# 目录

目录

[设计报告 1](#_Toc155093830)

[目录 2](#_Toc155093831)

[一、 引言 3](#_Toc155093832)

[1．项目背景和目的 3](#_Toc155093833)

[2 . 简要描述系统的主要功能和特性 3](#_Toc155093834)

[(1) 用户管理系统 3](#_Toc155093835)

[(2) 考试创建与管理 4](#_Toc155093836)

[(3) 在线考试功能 4](#_Toc155093837)

[二、系统架构设计 4](#_Toc155093838)

[1．系统总体架构 4](#_Toc155093839)

[2．系统的部署方式 5](#_Toc155093840)

[三、系统模块设计 5](#_Toc155093841)

[1．系统的各个模块（组员写自己负责的模块） 5](#_Toc155093842)

[2．模块之间的接口和通信方式 5](#_Toc155093843)

[3．各模块的详细设计 5](#_Toc155093844)

[四、数据库设计 6](#_Toc155093845)

[1．定义数据库模式（包括表结构、字段类型、关系） 6](#_Toc155093846)

[教师表 (teacher) 6](#_Toc155093847)

[考试表 (exams) 6](#_Toc155093848)

[问题表 (questions) 7](#_Toc155093849)

[考试记录表 (records\_test) 7](#_Toc155093850)

[学生考试表 (student\_exam) 7](#_Toc155093851)

[五、用户界面设计 8](#_Toc155093852)

[1．系统界面截图 8](#_Toc155093853)

[六、安全性考虑 9](#_Toc155093854)

[1．身份验证 9](#_Toc155093855)

[2．数据安全 9](#_Toc155093856)

[3．sql注入 9](#_Toc155093857)

[七、性能优化 10](#_Toc155093858)

[1．PageHelp分页查询插件 10](#_Toc155093859)

[2．Spring Task优化考试状态 10](#_Toc155093860)

[八、测试策略 11](#_Toc155093861)

[1．单元测试、集成测试、系统测试 11](#_Toc155093862)

[1． 测试用例，预期结果 11](#_Toc155093863)

[九、总结 11](#_Toc155093864)

# 引言

## 1．项目背景和目的

（1）教育行业的数字化转型：随着信息技术的快速发展，教育行业也在不断进行数字化转型。传统的纸质考试方式存在着许多问题，如试卷管理不便、阅卷耗时、成绩统计困难等。因此，开发一个在线考试系统可以提供更高效、便捷和准确的考试方式，满足现代教育的需求。

（2）提高考试效率和准确性：在线考试系统可以自动化和智能化地处理试卷的生成、考生的答卷、阅卷和成绩统计等过程。相比传统的手工操作，它可以大大提高考试的效率和准确性，减少人为错误和漏洞。

（3）方便灵活的考试安排：在线考试系统可以灵活地安排考试时间和地点，不再受限于传统的固定时间和地点。考生可以根据自己的时间和地点选择进行考试，提高了考试的便捷性和灵活性。

（4）提供反馈：在线考试系统可以及时给考生提供考试结果和反馈，帮助他们了解自己的考试表现，并进行针对性的学习和提高。

## . 简要描述系统的主要功能和特性

### 用户管理系统

学生账户管理: 允许学生注册、登录、修改个人信息，以及查看考试记录。

教师账户管理: 提供教师账户，使其能够创建、编辑和管理考试，监控学生的考试进度，并查看成绩考试创建与管理

### 考试创建与管理

试题库: 允许教师创建和维护试题库，包括单选题、多选题、填空题等不同类型

考试安排:支持教师安排考试的时间和持续时间，以及设置考试规则和限制。

### 在线考试功能

考试界面: 提供直观、用户友好的考试界面，支持多种题型的展示和作答

自动评分：提供自动评分功能，特别是对选择题的自动评分，以减轻教师的工作负

担

成绩报告:生成详细的成绩报告，展示学生在每个考试中的表现，并提供必要的反

馈

# 二、系统架构设计

## 1．系统总体架构

技术选型：



## 2．系统的部署方式

本系统采用前后端分离式开发，前端使用的是VsCode开发工具进行开发，后端使用IntelliJ IDEA 进行开发。前端的主机端口为localhost:4399，后端的主机端口为8080，数据端口为3306.前端通过在vue.config.js中配置server的端口属性可以发送请求到8080端口，后端在.yaml文件中配置数据库属性，以此访问数据库的信息。

# 三、系统功能设计

## 1．系统的各个功能

用例图：



主要分为两个主要功能区：教师端与考生端。

教师端主要负责：管理学生，管理题目，管理考试，查询考试信息等一些基础的功能

学生端主要部分：参与考试，以及查询自己的考试情况或者记录

## 2．各功能的详细设计

（1）题目新增：



sql语句：

@Insert("insert into exams set name=#{examName},className=#{className},start\_time=#{startTime},end\_time=#{endTime},teacher\_id=#{teacherId}")  
@Options(useGeneratedKeys = true, keyProperty = "id")  
int InsertExam(TeacherExamPlus teacherExamPlus);

@Insert("insert into question\_exams set question\_id=#{questionId},exam\_id=#{examId}")  
int InsertTeacherExam(Integer questionId, Integer examId);  
  
@Insert("INSERT INTO student\_exam (student\_id, exam\_id, state, name) " +  
 "VALUES (#{studentId}, #{examId}, '未开始', (SELECT MAX(name) FROM exams WHERE exams.id = #{examId}))")  
int InsertStudentExam(Integer studentId, Integer examId);

本功能共用三个sql语句实现：其中InsertExam用于插入考试的信息（考试时间或班级），插入成功后返回在数据库中的主键。并根据考试主键id与在前端选择的题目，遍历插入题目与考试关系表question\_exam，在根据班级查询学生ids，遍历插入到学生与考试student\_exam的关系表中。

（2）题目（考试）修改



@Update("update questions set content=#{content},type=#{type}," +  
 "opa=#{opa},opb=#{opb},opc=#{opc},opd=#{opd},correntAnswer=#{currentAnswer} where id=#{id}")  
int updateQuestion(Question question);

修改题目与修改考试功能业务代码相似共关系一个sql，点击题目后，在表单中回显数据，对题目进行修改后，进入mapper层,利用update语句修改数据。

1. 单个考试明细查询



@Select("select id, name,student\_name, teacher\_id, end\_time, start\_time, className, id, name, time, core, student\_id" +  
 " from exams left join records\_test join student\_exam se on exams.id = se.exam\_id and exam\_id=#{examId}")  
List<Records> getRecordByExamId(Integer examId);

本功能共涉及到一个sql,跟据前端传入的考试id查询到该考试下所有的考生信息并进行返回

1. 查询单个考试的信息



@Select("select id, name,student\_name, teacher\_id, end\_time, start\_time, className, id, name, time, core, student\_id" +  
 " from exams left join records\_test join student\_exam se on exams.id = se.exam\_id and student\_id=#{studentId}")  
List<Records> getRecordByStudentId(Integer studentId);

根据学生的id查询该学生所有的考试信息记录，并进行返回给前端渲染

# 四、数据库设计

## 1．定义数据库模式（包括表结构、字段类型、关系）

学生表 (students)

- `id` (主键)

- `name` (学生姓名)

- `password` (登录密码)

- `class\_name` (班级名称)

- `student\_id` (学生ID，外键关联其他表)

- `image` (学生头像)

### 教师表 (teacher)

- `id` (主键)

- `name` (教师姓名)

- `teacher\_id` (教师ID，外键关联其他表)

- `password` (登录密码)

### 考试表 (exams)

- `id` (主键)

- `name` (考试名称)

- `teacher\_id` (负责教师ID，外键关联教师表)

- `end\_time` (结束时间)

- `start\_time` (开始时间)

- `className` (班级名称)

### 问题表 (questions)

- `id` (主键)

- `content` (问题内容)

- `type` (问题类型)

- `opa` (选项A)

- `opb` (选项B)

- `opc` (选项C)

- `opd` (选项D)

- `correntAnswer` (正确答案)

### 考试记录表 (records\_test)

- `id` (主键)

- `name` (考试记录名称)

- `time` (考试时间)

- `core` (考试成绩)

- `student\_id` (学生ID，外键关联学生表)

### 学生考试表 (student\_exam)

- `id` (主键)

- `student\_id` (学生ID，外键关联学生表)

- `exam\_id` (考试ID，外键关联考试表)

- `state` (考试状态)

- `name` (考试名称)

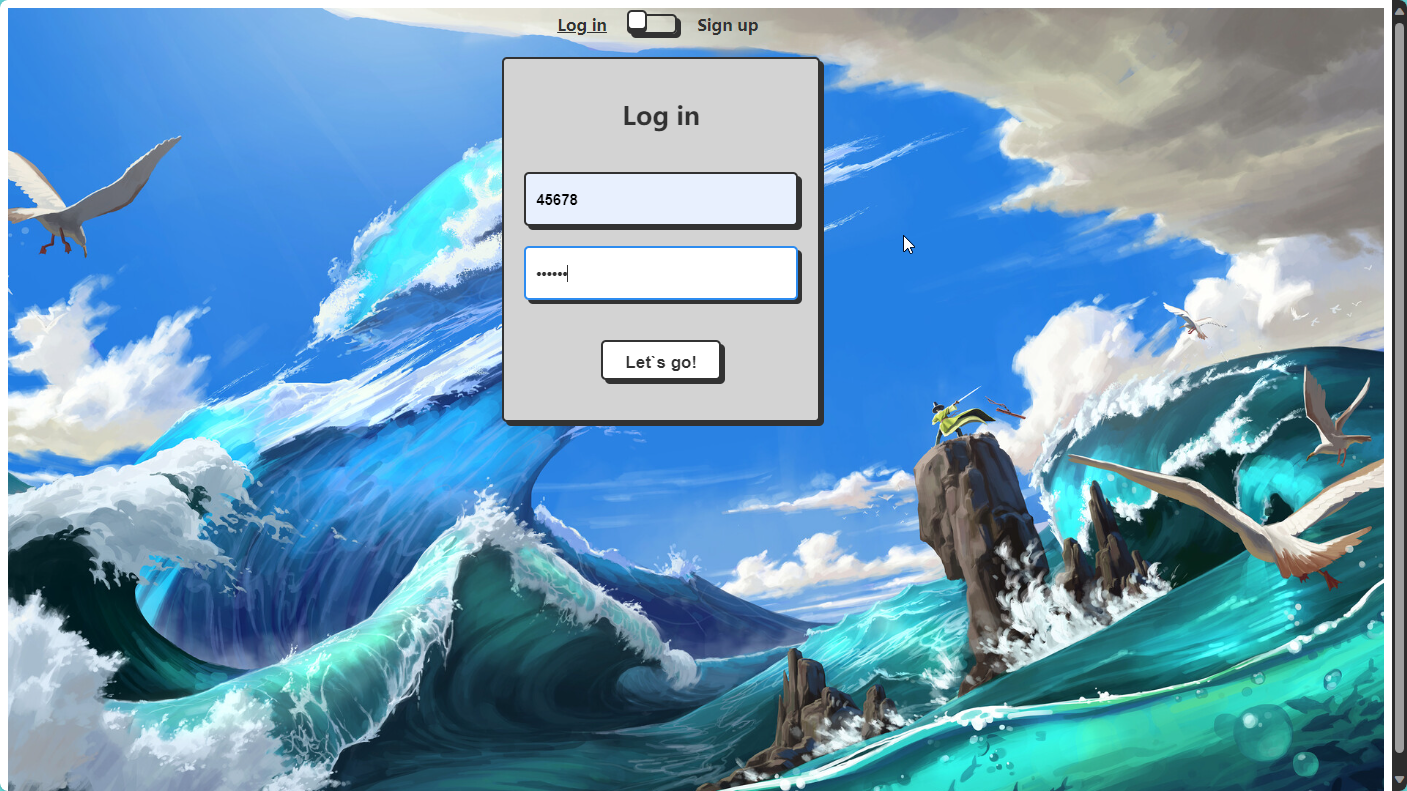
图示

描述已自动生成

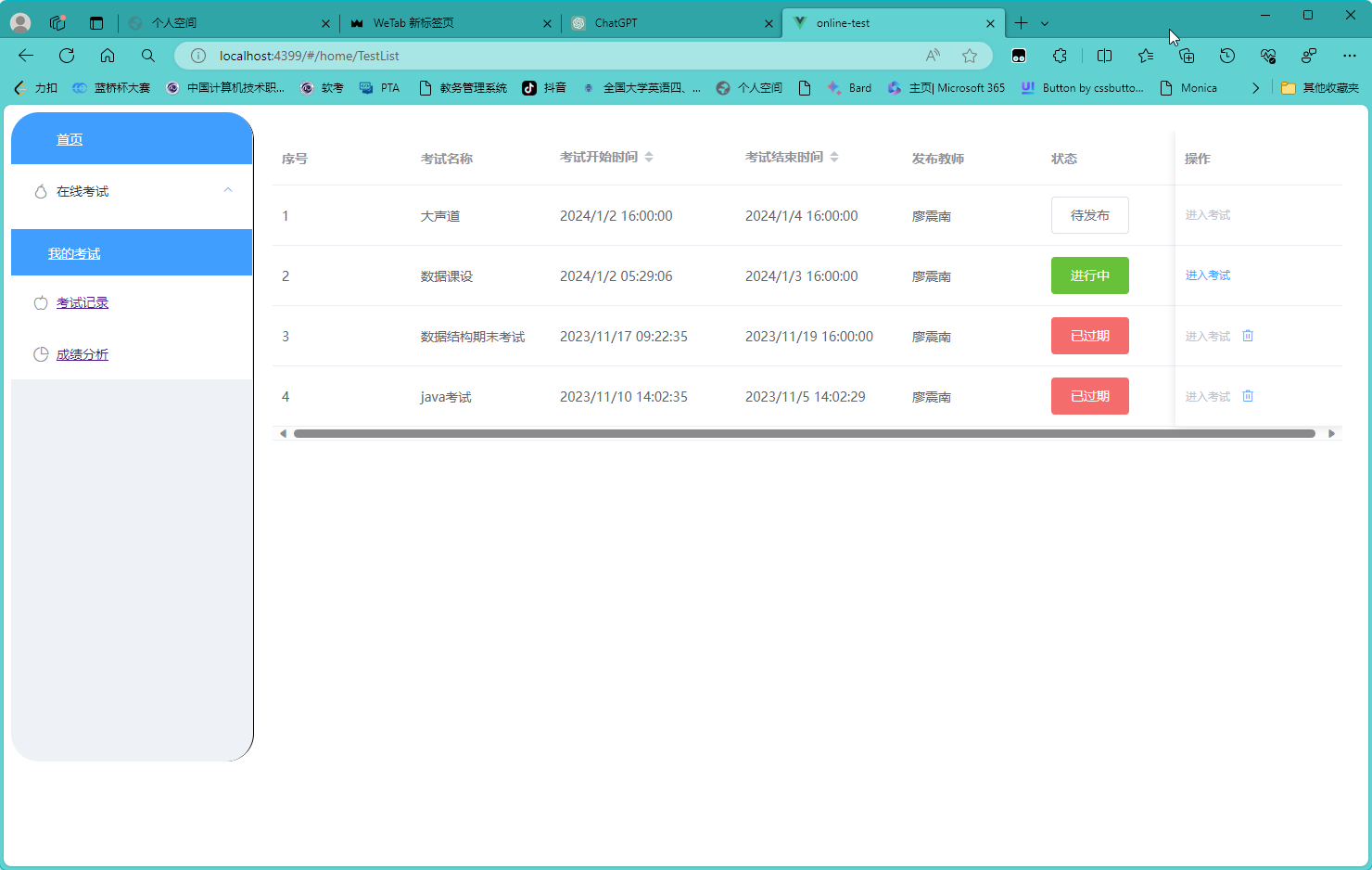
# 五、用户界面设计

## 1．考生端

（1）登录页面



(2)学生考试详细

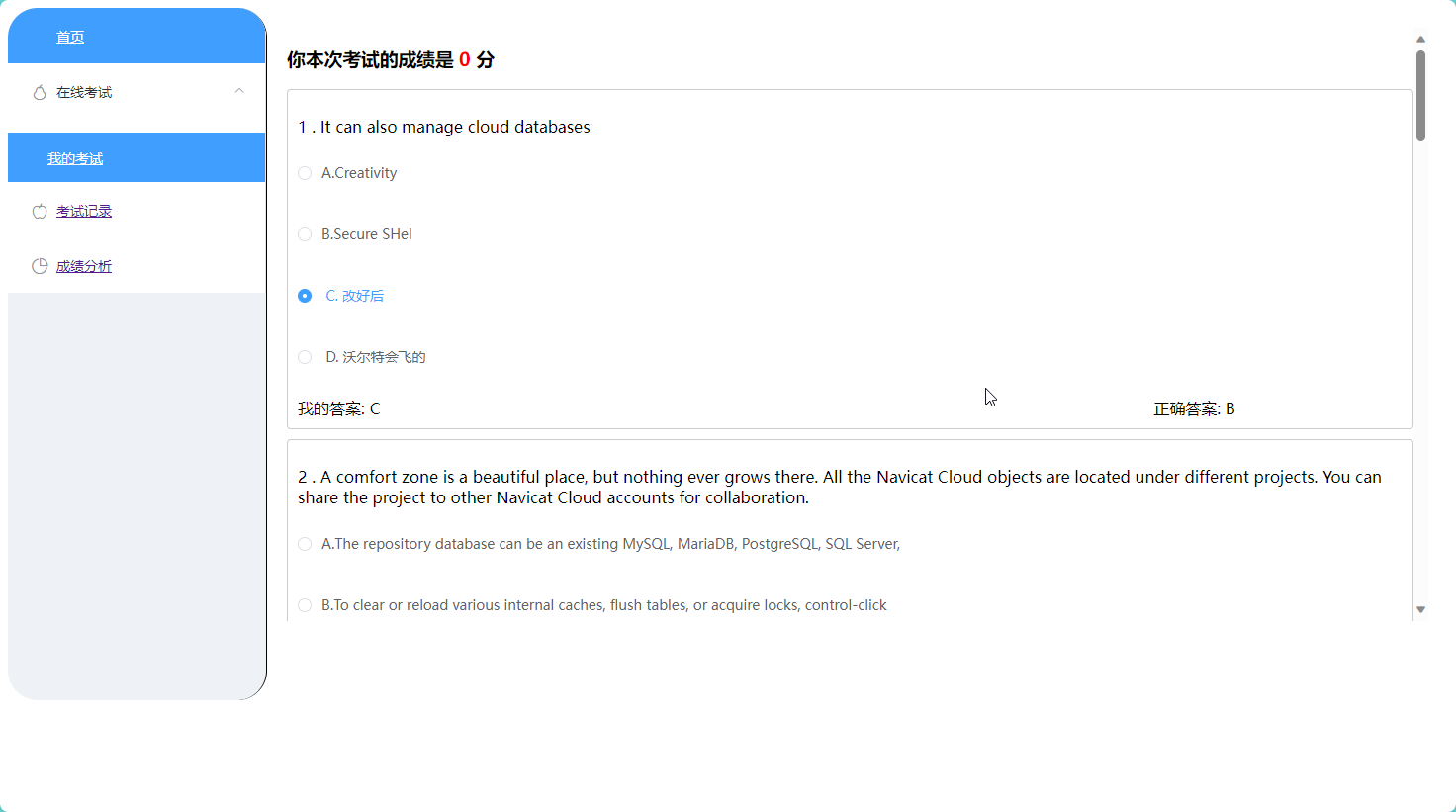


（3）考试中界面

图形用户界面, 应用程序, 电子邮件

描述已自动生成

（4）提交考试试卷后



（5）考试记录

图形用户界面, 应用程序, 表格

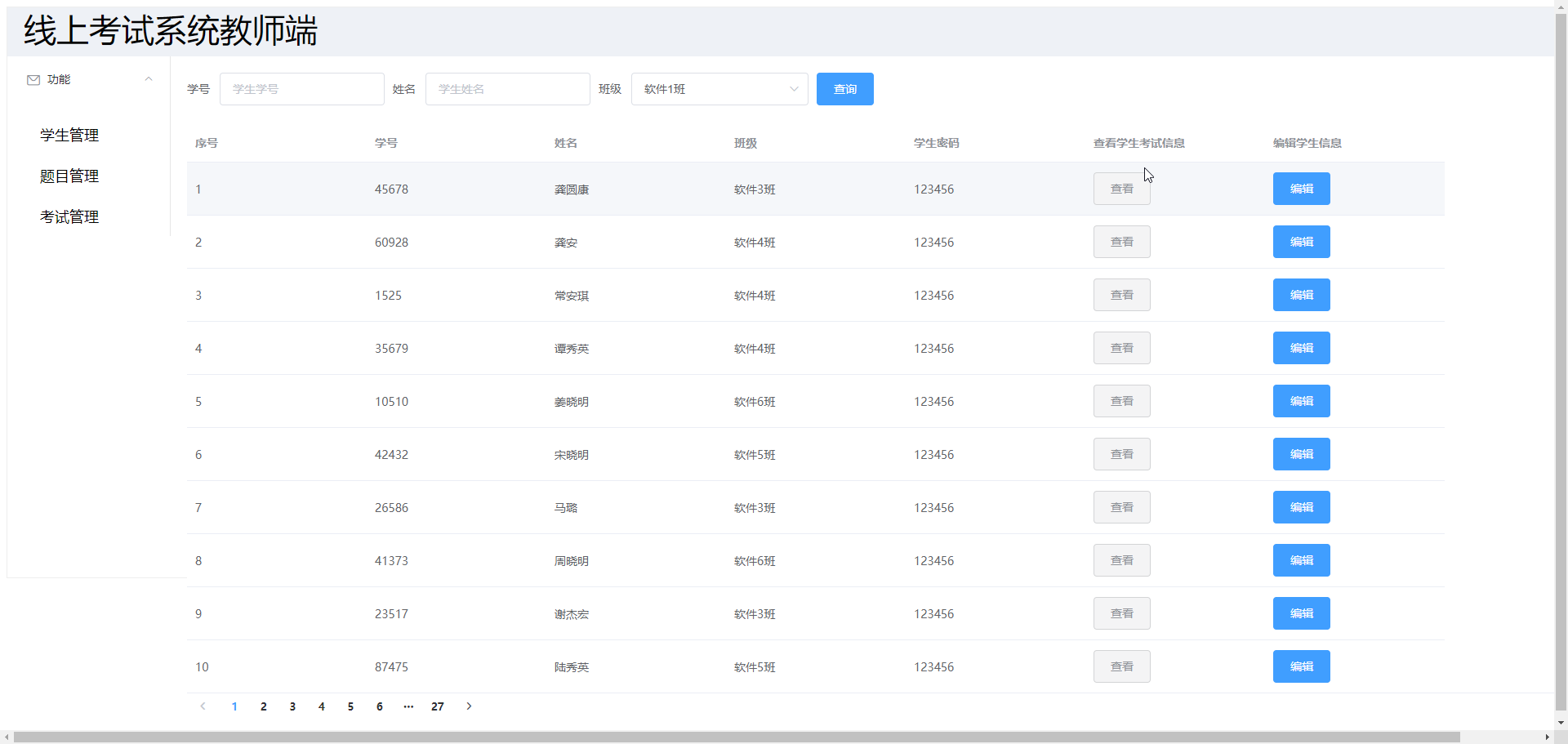
描述已自动生成

（6）考试成绩分析

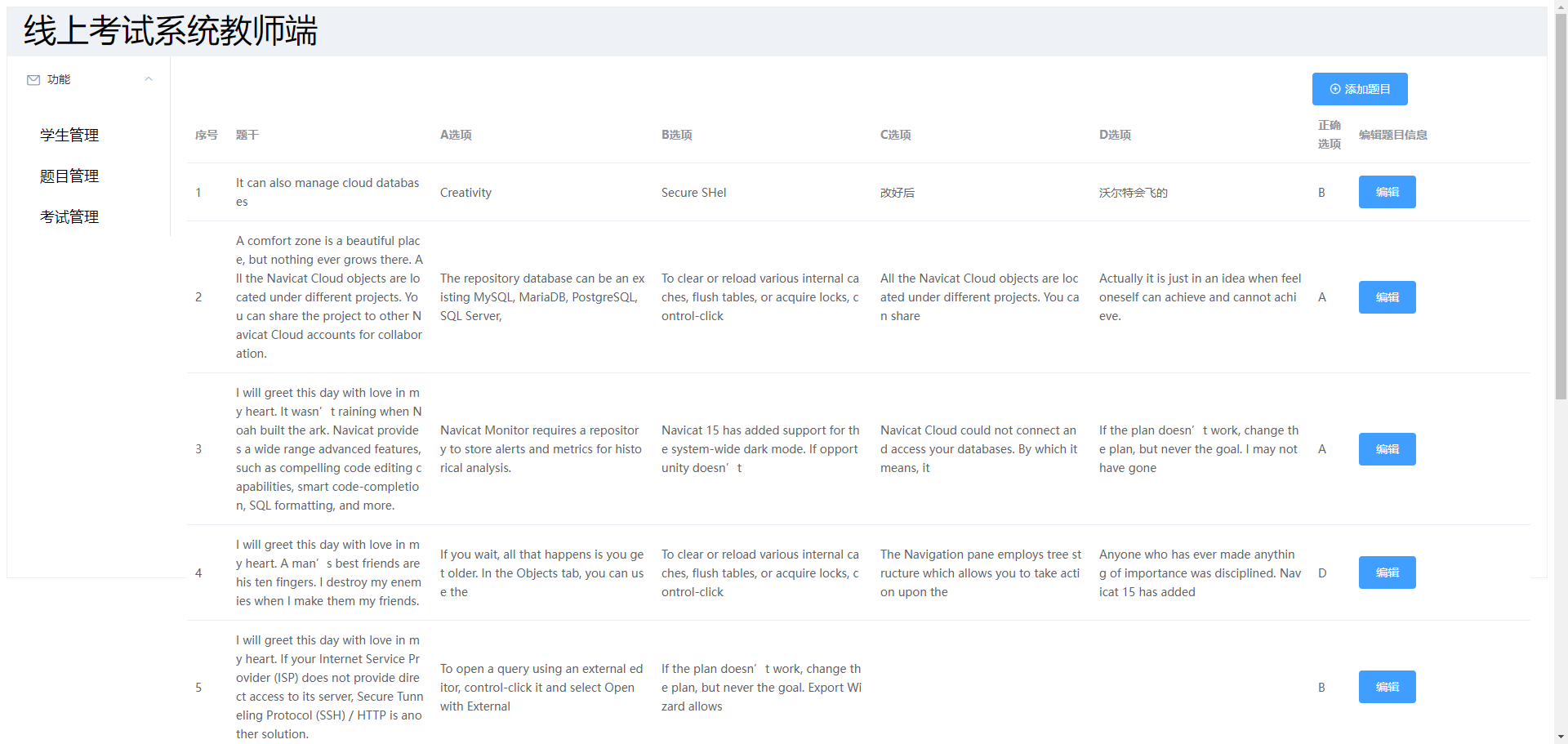


## 2. 教师端

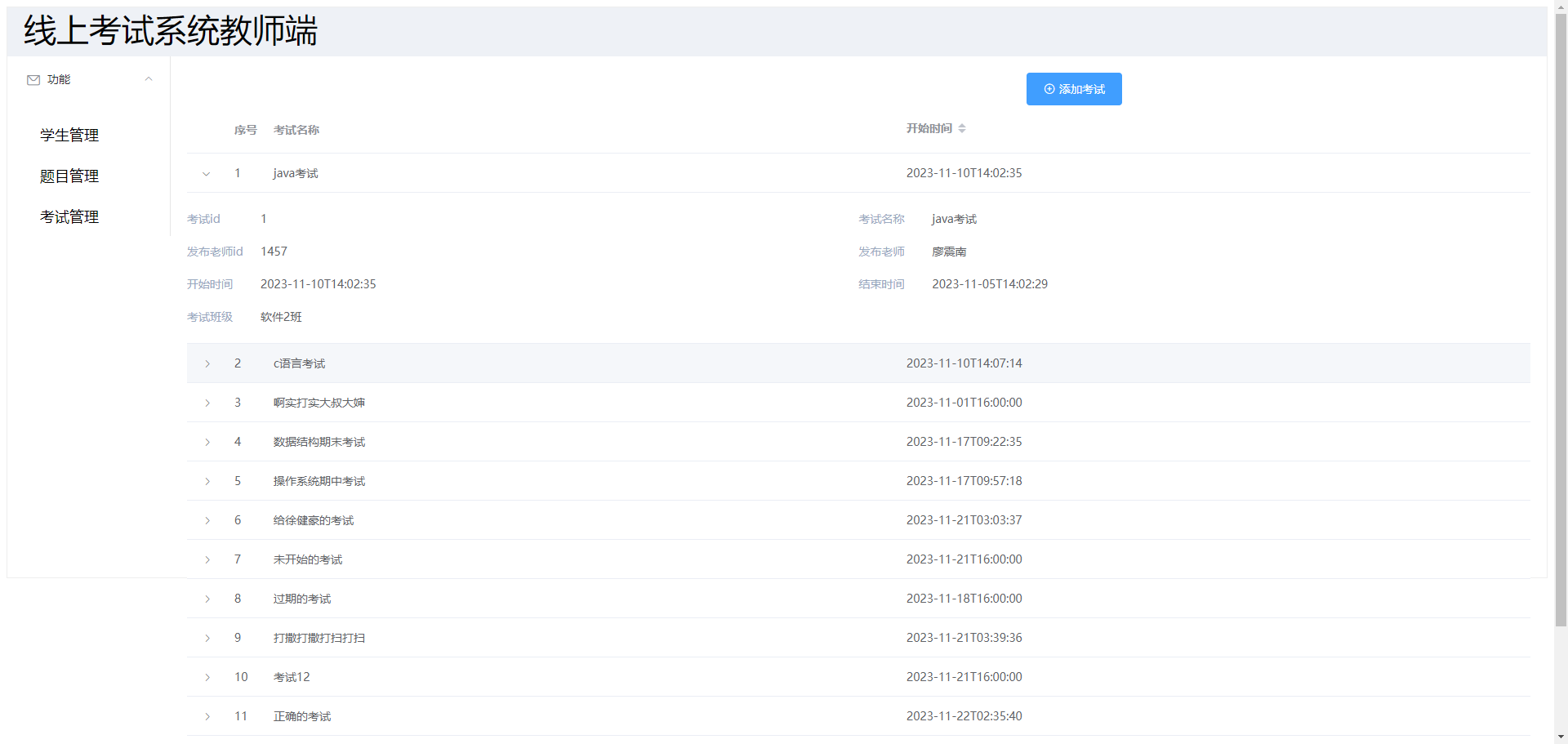
(1)学生管理页面



（2）题目管理页面



（3）考试管理页面



（4）添加考试

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

# 六、安全性考虑

## 1．身份验证

1.1 用户登录认证

系统采用JWT（JSON Web Token）来进行用户身份验证。以下是相关的安全性考虑：

JWT安全性：使用HS256或更高级别的加密算法来签名JWT，确保令牌的完整性和保密性。

令牌过期时间：设置合理的令牌过期时间，以减少令牌被滥用的可能性。客户端在过期前需重新获取令牌。

HTTP传输：所有用户认证和令牌传输均通过安全的HTTP协议，防止中间人攻击。

防重放攻击：使用随机生成的令牌以及合理的过期时间，减少令牌被截获后重放的风险。

1.2 用户密码安全

密码哈希存储：用户密码在数据库中以哈希形式存储，使用适当的哈希算法（如bcrypt）以增加破解难度。

密码策略：强制用户采用强密码，包括数字、字母和特殊字符，并定期提示或要求更新密码。

## 2．数据安全

2.1 数据加密

敏感数据加密：对于敏感数据，如用户个人信息，采用适当的加密算法进行加密，确保数据在存储和传输时的安全性。

数据库连接加密：使用SSL/TLS协议加密数据库连接，防止数据在传输过程中被窃听。

## 3．sql注入

3.1 使用参数化查询

参数化查询： 使用参数化查询方式而不是拼接字符串，以防止SQL注入攻击。

mybatis框架:使用mybatis框架来处理数据库查询，框架通常能够有效地防范SQL注入。

3.2 输入验证和过滤

输入验证：对所有用户输入进行验证和过滤，确保用户提供的数据不包含恶意内容。

白名单过滤：使用白名单过滤输入数据，只允许符合规定格式的数据通过。

转义字符：对用户输入中的特殊字符进行转义，防止其被误解为SQL命令。

# 七、性能优化

## 1．PageHelp分页查询插件

1. 优点

PageHelp 是系统中用于实现分页查询的插件，它允许在数据库查询中轻松地实现分页功能。为了确保系统性能最佳化，我们将在以下方面对 PageHelp 进行性能优化。

2. 目标

优化 PageHelp 插件，以提高分页查询的效率，减少数据库负载，并确保用户在执行大型数据集查询时获得更快的响应时间。

3. 性能优化策略

3.1 分页查询时的字段选择

在进行分页查询时，仅选择必要的字段。避免不必要的字段将降低查询的复杂性，提高查询速度。

3.2 数据量大时的延迟加载

对于大型数据集，考虑使用延迟加载的方式，即只在需要时加载数据。这可以通过 PageHelp 插件的配置参数进行控制。

3.4 查询缓存

PageHelp 插件应支持查询缓存。对于相同的查询，避免重复查询数据库，而是从缓存中获取结果，以提高查询速度。

3.5 异步查询支持

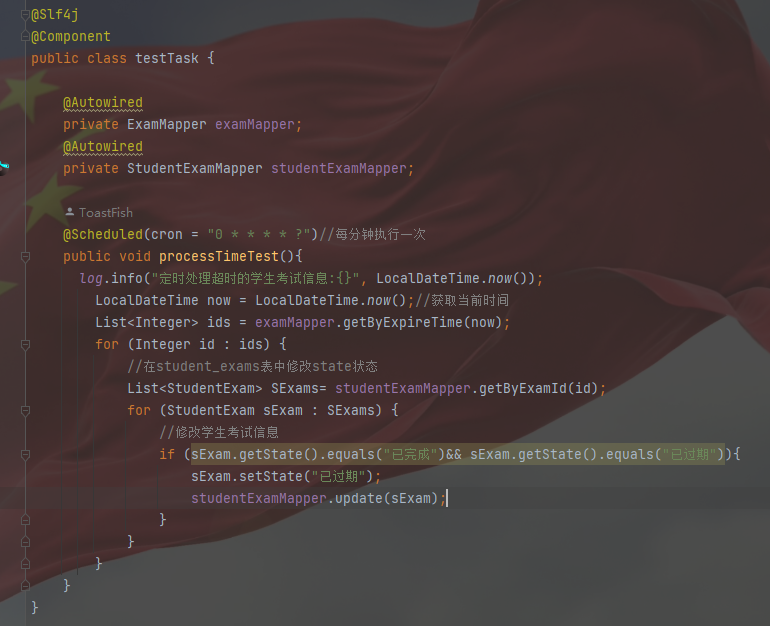
考虑在 PageHelp 插件中引入异步查询的支持。对于大数据集，异步查询可以提高系统的响应速度，确保用户体验。

## 2．Spring Task优化考试状态

项目问题：因为数据库中的考试是以开始时间start\_time,与结束时间end\_time来划定考试时间的。数据库是静止的，导致当当前时间过了结束时间（end\_time）之后，学生与考试关系表中的状态还是为**(未开始)。**考生仍然可以参加考试

解决思路：定期地检查数据库，为已经超过end\_time的考试，并且没有参加此次考试的学生的考试信息的状态改为**（已过期）**

解决方案：利用Spring Task技术，使用（@Scheduled）注解与cron=”0 \* \* \* \* ?”搭配，生成每分钟执行一次的事件。检查数据库中是否有过期未修改状态的学生考试信息，并进行修改。



# 八、总结