



南開大學
Nankai University

计算机学院和网络空间安全学院
软件工程实验课

《英语六级考试报考系统》
项目开发文档

队长：贾宇航 2013628

队员：沈冠翔 2012805 队员：高森森 2010764

队员：戴廷钧 2011258 队员：姬世成 2012648

专业：计算机科学与技术 信息安全

2023 年 6 月 20 日

目录

1 开发背景	3
2 项目目标	3
2.1 总体目标	3
2.2 功能实现目标	4
3 项目开发环境	5
4 可行性分析	6
4.1 技术可行性	6
4.2 经济可行性	6
4.3 时间可行性	7
4.4 法律和合规可行性	7
4.5 用户可行性	7
5 项目计划	7
6 需求分析	9
6.1 编写目的	9
6.2 编写背景	9
6.3 任务概述	9
6.3.1 任务目标	9
6.3.2 用户特点	10
6.3.3 假定与约束	11
7 业务描述	12
7.1 系统总业务流程图及其描述	12
7.2 各个子业务流程图及其描述	13
7.2.1 用户信息管理子业务	13
7.2.2 试题管理子业务	13
7.2.3 考生答题子业务	14
7.2.4 阅卷子业务	15
8 数据需求	16
8.1 数据需求描述	16
8.2 数据流图	17
8.3 数据字典	17
9 功能需求	19
9.1 功能划分	19
9.2 功能描述	19
9.2.1 管理员端	19
9.2.2 教师端	19

9.2.3 学生端	19
10 非功能性需求	20
10.1 用户体验需求	20
10.2 可靠性需求	20
10.3 性能需求	20
10.4 可用性需求	20
11 系统运行要求	20
11.1 硬件配置要求	20
11.2 软件配置要求	20
12 系统设计	21
12.1 模块划分	21
12.2 系统架构	22
12.3 数据流程	22
12.4 数据库设计	23
12.5 UI 设计	25
12.5.1 登录/注册界面	25
12.5.2 学生端	26
12.5.3 教师端	29
12.5.4 教务管理人员端	34
13 系统测试	35
13.1 测试环境	35
13.2 功能测试	35
13.3 性能测试	37
13.4 编程规范	37
13.5 性能优化	38
14 项目管理	40
14.1 参与人员及分工	40
14.1.1 代码实现部分	40
14.1.2 文档内容部分	40
14.2 项目进展记录	40
14.3 项目管理工具	41
15 总结	42

1 开发背景

随着全球化的发展和国际交流的日益频繁，英语已经成为一种全球通用的语言。在中国，英语已经成为大学英语教育的核心内容，尤其是大学生必须通过英语六级考试来证明自己的英语水平。然而，传统的六级考试方式存在一些问题，如考试报名流程繁琐、试题生成和阅卷效率低下、成绩查询不便等。

基于这些问题，我们决定开发一个全新的英语六级在线考试系统，以便提供更便捷、高效和公平的考试体验。该系统旨在利用现代技术手段，改善传统六级考试的各个环节，为学生和教师提供更好的考试管理和学习辅助工具，主要的改进点有：

- 首先，我们将通过在线报名功能简化考试报名流程。传统的报名方式需要学生亲自前往指定地点填写表格并缴纳报名费用，这耗费了大量时间和精力。而在线报名功能可以让学生在家中通过系统完成报名、选择考试科目和缴纳费用，大大提高了报名的便利性和效率。
- 其次，系统将实现自动化的试题生成和阅卷功能。目前的六级考试试题都是人工编写和批改的，这个过程既费时又容易出现人为差错。通过系统的试题生成功能，我们可以根据预设的题库和题目属性，自动生成符合规定要求的试卷，确保每次考试试题的随机性和公平性。而自动化的阅卷功能可以对客观题进行自动评分，大大减轻了改卷教师的负担并提高了评分的准确性和一致性。
- 此外，系统还将提供便捷的成绩查询功能。传统的成绩查询方式需要学生亲自到指定地点查询成绩，这不仅浪费了学生的时间，也给学校和教师带来了不必要的工作压力。通过在线成绩查询功能，学生可以随时随地通过系统查询自己的考试成绩和答题情况，方便快捷。
- 最后，系统在后续还将推出数据统计和分析功能，以帮助学校和教师对考试结果进行全面的统计和分析。通过系统生成的统计数据，学校和教师可以了解整体的考试情况，包括通过率、平均分、题目难度分析等，为进一步的教学和考试改进提供参考依据。

《英语六级考试报考系统》的开发旨在改善传统考试方式存在的问题，提供便捷、高效和公平的考试体验。通过引入现代化技术手段，该系统将极大地简化报名流程，提高试题生成和阅卷效率，并提供便捷的成绩查询和数据统计功能。相信该系统的实施将为学生、教师和学校带来诸多便利和效益，提升英语六级考试管理的水平 and 质量。

2 项目目标

2.1 总体目标

《英语六级考试报考系统》要实现的项目目标总体目标：

- 实现可上线的系统：项目的最终目标是开发一个稳定、可靠的在线考试系统，具备上线和商用的能力。系统应经过充分的测试和调试，确保在实际运行环境中的稳定性和性能表现。同时，系统应具备高可用性和可扩展性，以支持大规模的用户访问和并发考试。
- 提供高效的用户界面和操作体验：系统应具备直观、简洁的用户界面，使用户能够轻松地进行操作和导航。考虑到不同用户的技术水平和习惯，系统应提供友好的交互设计和指导，确保用户能够快速上手和高效使用系统。

- 支持大规模用户同时使用：考虑到英语六级考试的广泛参与度，系统应具备支持大规模用户同时使用的能力。通过优化系统架构和扩展服务器资源，保证系统的性能和响应时间，确保用户能够顺利访问和使用系统，避免因用户数量增加而导致的系统瓶颈。
- 保障数据安全和隐私保护：系统应具备严格的数据安全措施，包括数据加密、访问权限控制和漏洞修复等。用户的个人信息和考试数据应得到妥善保护，避免数据泄露和不当使用。系统还应遵守相关的隐私法规和数据保护要求，以确保用户的权益和信任。
- 实现系统的可扩展性和易维护性：系统应具备良好的可扩展性和易维护性，以满足未来的功能扩展和需求变化。采用模块化设计和可重用的代码结构，确保系统的灵活性和可持续性。此外，系统的维护和更新应方便快捷，减少对业务的影响和停机时间。

通过实现以上目标，我们将建立一个具备商业价值的英语六级考试报考系统，支持大规模用户同时访问和使用。系统的稳定性、性能、安全性和用户体验将成为项目开发的重点，以确保系统能够满足用户的期望，并提供可靠的在线考试服务。

2.2 功能实现目标

对于项目的功能实现目标简要描述如下：

- 考试信息发布：管理员可以发布考试通知、考试时间等相关信息，供考生和其他用户查看。
- 报名管理：提供在线报名功能，包括学生信息录入、报考科目选择、支付报名费用等。
- 题库管理：出题组可以创建、编辑和管理题库，包括选择题、阅读理解、写作等各类题型，并设置题目属性和难度等级。
- 试卷生成：根据考试科目和题库，系统能够自动生成符合规定要求的试卷，确保每次考试试题的随机性和公平性。
- 答卷评分：改卷教师可以查看考生的答卷，进行评分和备注，并生成成绩报告。
- 成绩查询：大学生可以通过系统查询考试成绩，答题情况，系统应提供便捷的成绩查询界面。
- 权限管理：系统应支持不同用户角色的权限管理，确保各用户只能访问其权限范围内的功能和数据。
- 考试监控：提供在线考试环境，监控考生的答卷过程，防止作弊行为的发生，并记录考试过程中的异常情况。
- 数据统计与分析：系统能够对考试数据进行统计和分析，如通过率、平均分、题目难度分析等，以提供给管理员和教师有关考试结果的参考。
- 系统维护和日志记录：系统应具备良好的维护性和可扩展性，同时记录用户操作日志、异常情况和系统运行状态，以便管理员进行故障排查和系统优化。

通过实现以上目标，我们的目标是建立一个完整的、功能齐全的英语六级在线考试系统，为学校、学生和教师提供高效、便捷和可靠的考试管理平台，提升考试的质量和管理水平。

3 项目开发环境

《英语六级考试报考系统》的项目开发主要采用 Vue + Springboot + mybatis 的分层设计，使用这种前后端分离的模式可以更好地实现团队的分工。下图为软件开发环境架构图：

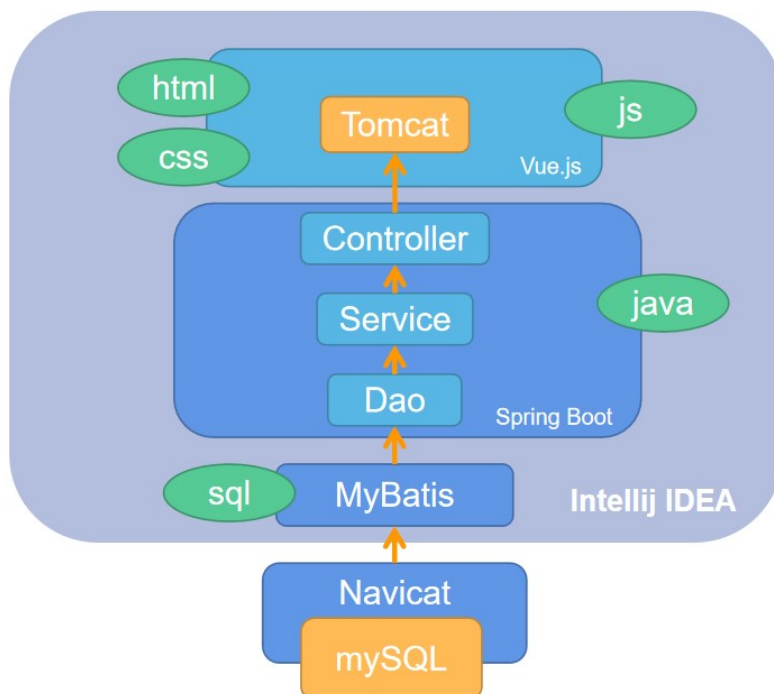


图 3.1: 软件开发环境架构图

以下是详细项目开发环境介绍：

- 集成开发环境（IDE）：本次项目开发使用 IntelliJ IDEA 作为开发工具。IntelliJ IDEA 是一款由 JetBrains 开发的集成开发环境，专注于 Java 开发，并提供了丰富的功能和工具。它具有智能代码编辑器、强大的代码导航和自动补全功能，以及调试和测试工具，可以提高我们开发的效率。
- 后端开发语言：本次项目后端开发使用 Java 语言进行后端开发。Java 是一门通用的、面向对象的编程语言，具有跨平台性和广泛的应用领域。其次，Java 还拥有庞大的开发者社区和成熟的生态系统，提供了丰富的类库和框架，便于我们编写可靠、高效的应用程序。
- 前端开发语言：本次项目前端开发使用 HTML、CSS 和 JavaScript 进行前端开发：
 - HTML 用于定义网页的结构和内容。它使用标签（tag）来标识不同的网页元素，如标题、段落、图像、链接等。通过合理使用 HTML 标签，可以创建出清晰、有层次感的网页结构。
 - CSS 用于控制网页的样式和布局。它通过选择器和样式规则来选择特定的网页元素，并定义其外观和排版方式。CSS 提供了丰富的样式属性，如颜色、字体、边距、背景等，使得开发人员可以对网页进行精确的样式控制。
 - JavaScript 是用于网页交互和动态效果。通过操作网页元素、处理用户输入和响应事件等方式，实现与用户的实时交互。
- 后端框架：本次项目选择使用 Spring Boot 作为后端框架。Spring Boot 是一个快速开发的 Java 框架，简化了 Spring 应用程序的配置和部署过程。它提供了自动化配置和约定优于配置的原则，

减少了我们的工作量，提高了开发效率。Spring Boot 还集成了许多常用的 Spring 组件和第三方库，方便我们构建现代化的 Web 应用程序。

- 前端开发框架：本次项目选择使用 Vue.js 作为前端开发框架。Vue.js 是一款流行的 JavaScript 前端框架，它通过采用组件化的开发方式，使得前端开发更加模块化、可维护性更强。此外它还具有响应式的数据绑定和虚拟 DOM 技术，可以快速构建交互性强、用户体验优秀的前端界面。
- 数据访问层框架：本次项目选择使用 MyBatis 作为数据访问层框架。MyBatis 是一款优秀的持久层框架，与 Java 应用程序无缝集成。它采用 XML 或注解的方式进行 SQL 映射，提供了灵活的数据库访问能力。MyBatis 还支持事务管理、缓存机制和动态 SQL 等特性，方便开发人员进行数据操作和管理。
- 前端构建工具：本次项目使用 Maven 作为前端项目的构建工具。Maven 是一个强大的项目管理工作，可以帮助您管理项目的依赖关系、构建和部署过程。通过在项目的 pom.xml 文件中定义依赖和插件，Maven 能够自动下载所需的库和工具，并根据配置进行项目的构建和打包。
- 前端服务器：本次项目使用 Maven 项目中的 Tomcat 作为前端服务器。Tomcat 是一个常用的 Java Web 服务器，它能够运行和部署 Java Web 应用程序。通过将前端代码部署到 Tomcat 服务器上，我们可以通过 HTTP 协议访问和展示前端页面，与后端进行交互。
- 数据库管理工具：本次项目选择使用 MySQL 数据库和 Navicat 作为数据库管理工具。MySQL 是一个流行的关系型数据库管理系统，具有高性能、稳定可靠的特点。Navicat 是一款强大的数据库管理工具，它提供了可视化的界面和丰富的功能，方便我们进行数据库的设计、开发和管理。

4 可行性分析

4.1 技术可行性

技术可行性分析如下：

- 技术能力：开发团队成员具备一定的编程能力、环境配置能力和项目经验，能够使用 IDEA、Navicat 等开发工具进行项目开发，并能够利用个人笔记本进行开发。
- 开发工具和框架：在技术可行性方面，团队选择使用 IDEA 作为主要开发工具，这是一款功能强大的集成开发环境，能够提供丰富的功能和工具支持，有助于提高开发效率和代码质量。同时，Java 作为后端开发语言具有广泛的应用领域和强大的生态系统，可以满足项目的需求。前端采用 HTML、CSS 和 JavaScript 作为开发语言，这些技术被广泛应用于 Web 开发，并具备良好的跨平台兼容性和丰富的 UI 交互效果，也能完全满足此次项目的需求。
- 网络和互联网访问：团队成员能够连接校园网并访问互联网，这对于获取必要的资源、调试和部署应用程序是必要的。

4.2 经济可行性

经济可行性分析如下：

- 资金预算：虽然项目没有明确的经费预算，但团队成员已经拥有了个人高性能笔记本电脑，不需要额外的硬件投入。同时，开发工具和框架的使用也是免费的。
- 学生身份：作为学生，我们能够充分利用免费或学生优惠的开发工具、学术资源和技术支持。

4.3 时间可行性

时间可行性分析如下：

- 开发周期：项目的开发周期为 3 个月，根据教学时间安排，基本框架需要在 6 月 5 日前完成，完整项目需要在 7 月 2 日前交付。
- 开发效率：由于团队成员具备一定的编程能力、环境配置能力和项目经验，他们可以快速适应项目，并以高效的方式进行协作开发。

4.4 法律和合规可行性

法律和合规可行性分析如下：

- 抄袭问题：本项目是完全由开发人员自行构建的，不存在抄袭等法律问题。确保项目的原创性和合法性。
- 数据隐私和安全：在设计和开发过程中，团队应遵守相关的隐私和数据安全法律法规，确保用户数据的安全和保密。

4.5 用户可行性

用户可行性分析如下：

- 管理员和教师用户：管理员和教师用户在信息化办公、电脑操作和网站访问方面具有一定的经验。他们可以通过一定的培训快速掌握系统的使用和管理。
- 学生用户：当前学生普遍具备较强的电脑操作和网站访问能力，他们可以在简单的探索后迅速熟悉并掌握系统的基本操作。系统设计和界面设计应该简单易懂，结合操作教程，使学生用户能够快速上手使用。
- UI 友好性：系统设计和界面设计将注重简洁易懂，提供清晰的操作指导和友好的用户界面，以便管理员和教师用户在繁忙的工作环境中能够轻松上手和使用。

5 项目计划

项目计划按照时间顺序进行安排：

1. 项目启动阶段 (3 月 1 日 - 3 月 15 日)：

- 进行项目启动会议，明确项目目标和范围。
- 确定项目团队成员和各自的职责，推选团队负责人。
- 确定项目大致的开发环境，进行工具的注册和学习等准备工作。
- 开启项目开发书的撰写。

2. 前期准备阶段 (3 月 15 日 - 5 月 15 日)：

- 进行《软件工程》课程的学习，为后续开发提供理论支撑和经验积累。
- 进行需求分析和功能规划，明确系统的基本功能模块。

- 编写项目计划和需求文档。
- 进行技术选型，确定使用的开发框架和技术工具。
- 使用当前准备的技术制作一个 DEMO，确定技术可行性。

3. 系统开发阶段 (5 月 15 日 - 5 月 25 日):

- 设计数据库结构，创建数据库并进行初始化配置。
- 开发用户登录和权限管理功能。
- 开发考试信息发布和报名管理功能。
- 开发题库管理和试卷生成功能。
- 实现在线考试环境和答卷评分功能。
- 开发成绩查询和数据统计与分析功能。

4. 前端设计和开发阶段 (5 月 25 日 - 5 月 31 日):

- 进行前端界面设计，包括页面布局和样式设计。
- 使用 HTML、CSS 和 JavaScript 等前端开发语言实现界面交互和动态效果。
- 集成前后端功能，确保系统的完整性和一致性。

5. 中期展示和验收 (5 月 31 日 - 6 月 5 日):

- 整合当期已完成的代码，形成《英语六级考试报考系统》的初代版本。
- 总结当前已完成的文档。
- 录制当前已完成的项目功能演示视频。
- 制作 PowerPoint、展示和答辩。

6. 系统测试和优化阶段 (6 月 5 日 - 6 月 20 日):

- 进行系统功能测试，包括单元测试和集成测试。
- 进行性能测试和安全测试，确保系统的稳定性和安全性。
- 修复和优化系统中发现的问题和漏洞。
- 进行 UI 的深度优化。

7. 系统部署和上线阶段 (6 月 20 日 - 6 月 25 日):

- 进行系统的性能调优和优化，确保系统能够承受并发访问和高负载。
- 进行最终的功能测试和验收测试。
- 考虑到成本限制和该项目的定位，我们仅仅论证将系统部署到生产环境，配置服务器和数据库的可行性。
- 上线系统，发布给管理员、教师和学生使用（待定）。

8. 项目收尾和总结阶段 (6 月 25 日 - 7 月 2 日):

- 编写项目总结报告，总结项目的开发过程和经验教训。

- 归档项目文档和代码，确保项目的可维护性和后续支持。
- 录制项目的完整功能演示并提交所有代码，完成项目交付。
- 进行项目的评估和回顾，收集反馈意见并进行改进。

以上是项目的大致开发计划，具体时间安排和任务分配可根据实际情况进行调整和细化。在项目开发过程中，需密切沟通和协作，及时解决问题和调整计划，确保项目按时交付并达到预期目标。

6 需求分析

6.1 编写目的

编写本部分的目的在于详细、准确和全面的定义英语六级报考系统的功能和用户需求，用以指导系统顺利地进行开发，提高开发效率，使系统开发人员在开发之前能周密地思考系统的全部需求从而减少后期重新设计、编码、测试等活动。本文档所描述的需求信息将作为该项目最终验收的标准和依据。

6.2 编写背景

由于参加六级考试的大学生众多，包含的信息量大且复杂，需要建立一个英语六级报考系统，使教师、管理者的工作规范化，系统化，程序化，提高信息处理的速度和准确性，能够及时、准确、有效的查询试卷信息和设计、批改试卷。同时设计数据库以支持题库系统，增添自动批改客观题的功能，为教师学生的工作提供遍历。

6.3 任务概述

6.3.1 任务目标

- **详细、准确和全面地定义该系统的功能和用户需求**
 - 系统应该能够提供报名信息，包括考试时间、地点、费用等。
 - 系统应该能够提供考试成绩查询功能。
 - 系统应该能够提供考试报名和缴费功能。
 - 系统应增设自动批改客观题、快速设计试卷的功能。
- **用以指导系统顺利地进行开发，提高开发效率**
 - 系统需求分析应该在开发之前完成，以便开发人员能够更好地理解系统需求并进行开发。
 - 系统需求分析应该包括详细的功能和用户需求，以便开发人员能够更好地理解系统需求并进行开发。
- **使系统开发人员在开发之前能周密地思考系统的全部需求从而减少后期编码、测试等活动**
 - 系统需求分析应该包括详细的功能和用户需求，以便开发人员能够更好地理解系统需求并进行开发。
 - 系统需求分析应该在开发之前完成，以便开发人员能够更好地理解系统需求并进行开发。

6.3.2 用户特点

此系统用户由管理员、出题组、改卷教师和大学生构成，三者特点如下：

- **管理员：**

- 管理员在系统中拥有最高权限，熟悉系统的整体管理和配置，包括用户管理、权限设置、数据维护等。他们可以添加、删除和管理其他用户的权限。
- 需求上，管理员需要掌握系统的各项操作和功能，对系统进行设置和配置，包括学生、考试信息、教师等相关信息的录入、修改、删除等操作；同时需要监测和分析考试和成绩数据，包括学生报考情况、考试成绩分布情况等，以便于调整考试安排和难度等。
- 管理员一般是教育系统工作人员，对信息化办公、电脑操作以及网站访问有较多的使用经验，可以在一定时间的培训后熟练掌握系统的使用和管理。
- 管理人员业务十分繁忙，设计的系统应该能够提供快捷、高效、批量化的辅助服务。

- **出题组：**

- 出题组负责制定考试题目，包括选择题、阅读理解、写作等。他们可以在系统中创建和编辑题目，并设置题目的相关属性和难度等级。
- 出题组人员大多为大学授课教师或者教育系统工作人员，需要使用系统来管理和发布英语六级的试题，但对整个系统的运行流程缺乏足够的了解，仅对考试内容、题目负责，拥有部分的管理权限。
- 需求上，出题组需要及时发布和更新考题，确保考试的正常进行；同时需要监测考生的答题情况和成绩分布，以便于调整考试难度。
- 教师信息化办公、电脑操作以及网站访问的能力由于专业、年龄的不同存在差异，同时教师、教育系统工作人员工作比较繁忙，长时间的培训并不可行，因此系统设计、界面设计应该简单易懂，在阅读一定的操作教程后就能很快上手。同时，系统应该提供快捷、批量化的辅助服务，避免重复处理大量的工作。

- **改卷教师：**

- 改卷教师负责批阅考试答卷，并给予评分和反馈。他们可以在系统中查看考生答卷，进行评分和备注，并生成成绩报告。
- 改卷老师不熟悉整个系统的运作流程，只对自己负责的某次考试中的某道题目负责，仅具有部分管理权限。
- 需求上，改卷教师需要及时完成试卷的批阅，确保成绩的及时发布。同时，由于改卷老师具有评分的权限，因此系统应该设计相应的机制（如多人阅卷取平均分）以保证试题批阅的公平性。
- 改卷老师的信息化办公、电脑操作以及网站访问的能力与出题组人员情况类似。

- **大学生：**

- 大学生是考试的参与者，他们使用系统进行报名、查看考试信息、参加考试和查询成绩等操作。

- 需求上，学生需要及时获取发布的考试信息；同时需要及时进行考试的报名、缴费、准考证信息的获取等；最后考生还应按照流程和考试规则完成考试，并获取成绩信息。
- 当前学生的电脑操作以及网站访问的能力普遍较强，在简单探索后能够快速熟练掌握某个网站系统的基本操作。
- 学生只需要处理自己的信息，因此不需要批量处理等功能，因此界面设计应该遵从简单和方便的原则。

6.3.3 假定与约束

《在线选课管理系统》中的假定和约束条件:

• 假定条件

- 硬件资源假定：本系统需要运行在一定的硬件资源上，包括服务器、存储设备和网络设备等。假定这些硬件资源能够满足系统的性能和可靠性要求。
- 软件平台假定：本系统需要基于特定的软件平台进行开发和实施，如操作系统、数据库、Web服务器等。假定这些软件平台能够满足系统的功能和性能要求，并且具有良好的兼容性。
- 安全性假定：本系统需要保证数据的安全性和保密性，以防止未经授权的用户访问和篡改数据。假定系统的安全措施能够满足数据安全的要求。
- 人力资源假定：本系统的开发和实施需要一定的人力资源支持，包括项目经理、系统分析师、开发人员、测试人员等。假定这些人力资源能够满足系统的质量和进度要求。
- 财务资源假定：本系统的开发和实施需要一定的财务资源支持，包括硬件和软件采购、人员培训等方面的支出。假定这些财务资源能够满足系统的开发。
- 用户特点假定：该系统的用户为管理员、出题组、改卷教师和大学生的，他们需要具备一定的计算机操作技能和网络应用基础，包括浏览器使用、电子邮件使用等。
- 其他假定：该系统会进行定期的维护和更新，以保证系统的正常运行和数据的安全性；用户可以通过教育系统提供的用户名和密码登录系统等。

• 约束条件

- 硬件资源约束：本系统在使用硬件资源时，需要考虑硬件资源的成本、可靠性、容量等因素，以保证系统的性能和可靠性要求。
- 软件平台约束：系统在选择软件平台时，也需要考虑软件平台的成本、兼容性、可维护性等因素，以确保系统能够正常运行。
- 安全性、法律约束：系统需要具有高度的稳定性和安全性，保证用户的信息和隐私不被泄露或遭受损失；在保证数据安全性的同时，需要遵守相关的法律法规和行业标准，如个人信息保护法、网络安全法等。
- 时间约束：系统的开发和测试时间受到一定的限制，需要在规定的时间内完成（具体的时间可以根据实际情况来定）。
- 总成本约束：系统的开发和测试成本受到一定的限制，需要在可接受的成本范围内完成（具体的成本可以根据实际情况来定）。
- 用户需求约束：系统需要满足学校和机构的管理需求，例如考试管理、学生管理等，还要具有良好的用户体验，保证用户能够方便、快捷地完成选课和管理操作

7 业务描述

7.1 系统总业务流程图及其描述

业务总流程图如下：

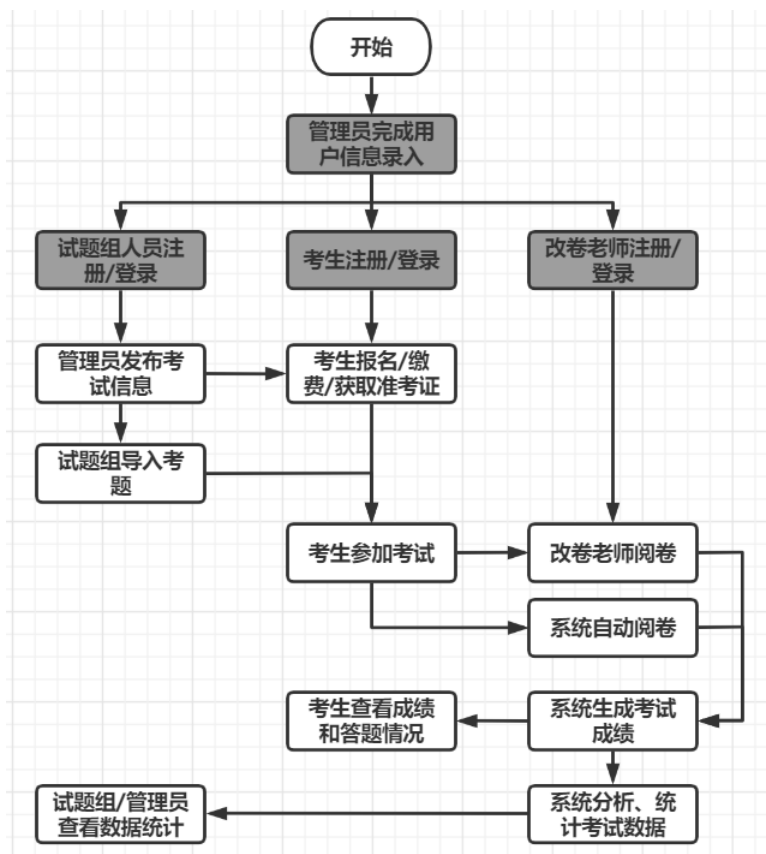


图 7.2: 系统总业务流程图

业务总流程描述：

1. 管理员登录后，对用户认证信息进行录入，用户可凭自己的身份信息进行身份认证并注册账号；
2. 试题组人员、考生和改卷老师可以凭借自己的身份信息进行注册并登陆系统；
3. 当有考试要进行时，管理员将考试信息录入并发布，包括考试的名称、考试的时间、缴费等信息；
4. 当考试信息发布后，考生可以进入系统进行报名、缴费操作，最后获取准考证信息，用于在考试时认证身份；
5. 当考试信息发布后，试题组人员可以进入系统进行进行考题的导入，考题导入后由系统管理，在考试时展示给考生进行作答；
6. 考题准备就绪后，到达考试时间，考试进入系统完成考试；
7. 考生作答结束后，改卷老师进入系统对部分题目进行阅卷；
8. 系统会对部分客观题进行自动阅卷；
9. 当系统阅卷和教师阅卷完成后，系统会统计每位考生的得分情况；

10. 分数统计结束后，考生再次登录查看自己的分数和答题情况；
11. 最后，系统会对本次考试的数据进行分析和统计，并生成相应的统计数据；
12. 管理员和试题组会收到相应的统计数据，并根据该数据对考试安排和试题难度进行分析和调整。

7.2 各个子业务流程图及其描述

7.2.1 用户信息管理子业务

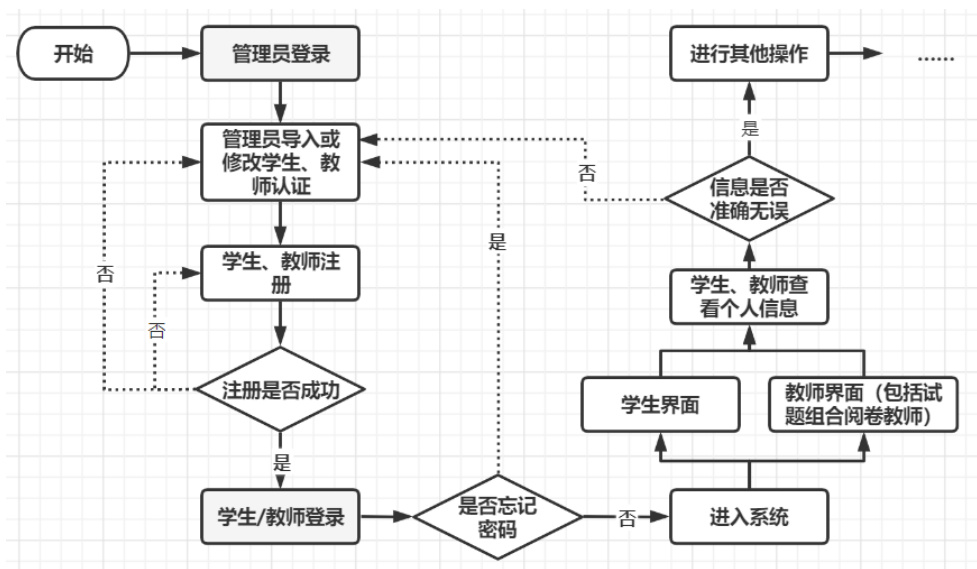


图 7.3: 用户信息、登录权限管理子业务流程图

用户信息管理子业务流程描述:

1. 管理员登录后，导入老师（包括试题组和阅卷老师）和学生的身份信息；
2. 在信息导入后，教师和学生可以凭借其身份信息注册账号，如果无法注册则尝试重新注册或联系管理员修改相关个人信息；
3. 注册账号后，教师和学生登录系统，如果忘记密码则联系管理员修改个人信息，如果密码正确则进入到各自身份的对应系统界面；
4. 进入到对应的系统界面后，学生或老师需要查看自己的个人信息是否有误，部分信息可以在系统上自行修改和完善，重要信息需要联系管理员进行修改；
5. 信息无误后，教师和学生等用户便可以执行其他相关操作了；

7.2.2 试题管理子业务

试题管理子业务流程描述:

1. 管理人员发布考试信息后，试题组找到对应的考试进行试题管理；
2. 对试题的管理主要分为：增、删、改三种操作；
3. 当试题组对试题进行导入或者修改时，需要对操作的试题类型进行判断；

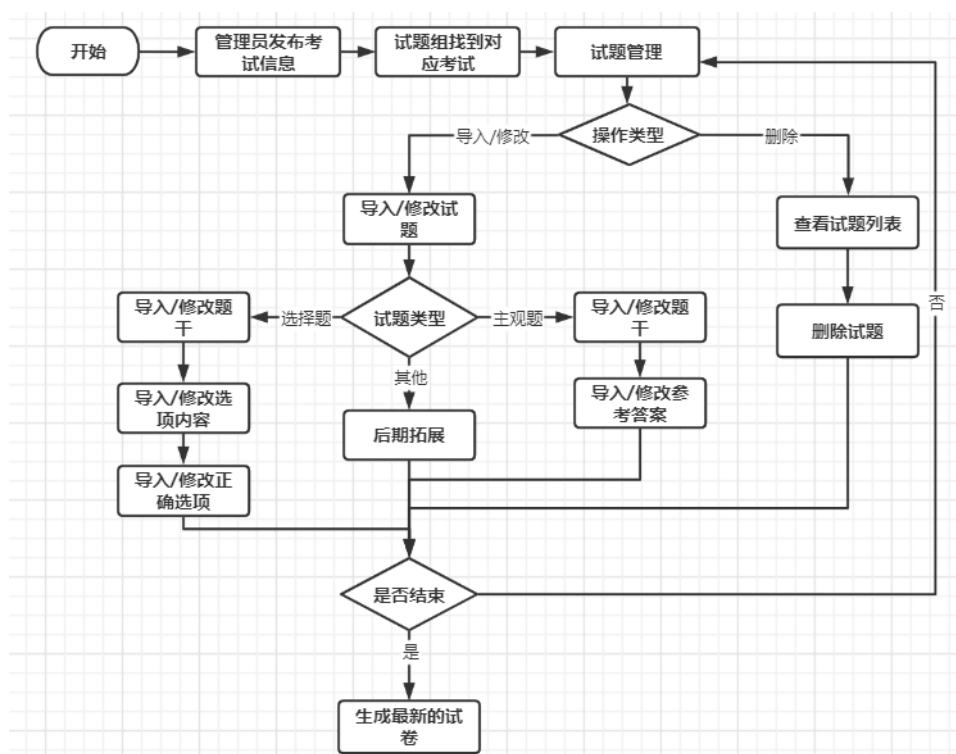


图 7.4: 试题管理子业务流程图

4. 当试题组对选择题进行导入或者修改时，需要先导入或修改选择题的题干，接着导入或修改选项的内容，再导入或修改正确答案；
5. 当试题组对主观题进行导入或者修改时，需要先导入或修改选择题的题干，接着导入或修改参考答案即可；
6. 如果试题组进行的操作是删除试题，则需要在系统中看到试题列表，接着将想要删除的试题进行删除即可；
7. 完成上述操作后，用户可以选择继续下一个操作或者保存退出；
8. 当完成所有的操作后，系统会生成最新的试卷样式；

7.2.3 考生答题子业务

考生答题子业务流程描述：

1. 考生进入系统后等待考试开始；
2. 考试开始后，考生可以看到试卷上的所有题目，考生可以选择作答或提前交卷两种操作；
3. 选择提前交卷后，考生会退出考试，考生退出考试后不能重新参加考试，也不能继续作答，之前的作答内容会被自动保存；
4. 如果考生选择作答某一考题，则会对试题类型进行判断；
5. 对于主观题，考生根据自己的理解填写回答的内容或者对之前的作答内容进行修改；

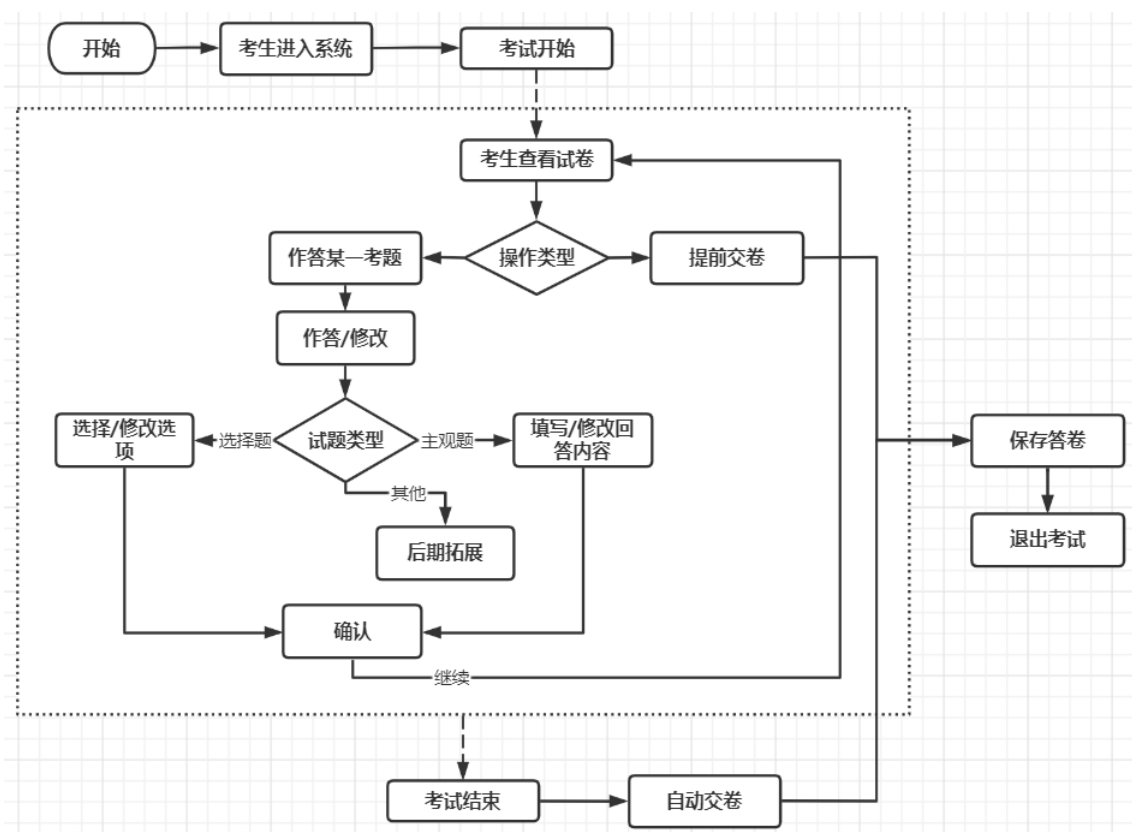


图 7.5: 考生答题子业务流程图

6. 对于选择题，考生需根据自己的理解选择正确的选项或者对之前的选择进行修改；
7. 完成作答或修改后，需要点击确认，系统会覆盖之前的答案；
8. 接着考生可以选择下一步的操作；
9. 当考试结束时，考生将不能继续答题，系统将会自动交卷，考生退出考试，之前的作答内容会被自动保存；

7.2.4 阅卷子业务

阅卷子业务流程描述：

1. 在考试结束后，学生的试卷会被自动保存；
2. 阅卷时，系统和阅卷老师会分别对选择题和主观题进行打分；
3. 对于客观题，系统会对比每一道客观题与其标准答案是否完全一致，完全一致则记满分，否则记零分，分数会被写回数据库；
4. 对于主观题，老师会根据参考答案以及自己的判断给出分数，分数也会被写回数据库；
5. 当所有试题都完成评分之后，系统会自动统计出每张试卷的得分；

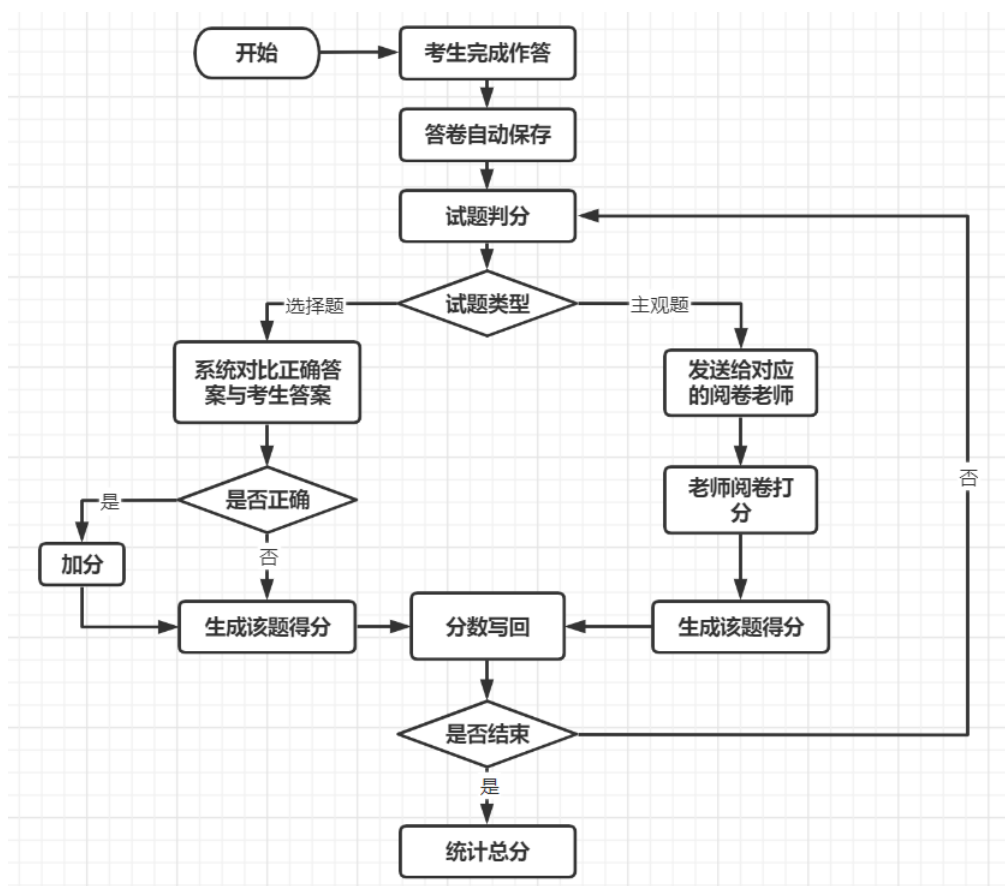


图 7.6: 阅卷子业务流程图

8 数据需求

8.1 数据需求描述

用户信息数据：用户信息数据包括学生、教师和管理人员的基本信息，包括姓名、性别、编号（或工号）、联系方式等。

试卷信息数据：试卷信息数据包括每个试卷的试卷编号、试卷名称、试卷题量、试题编号、考试时间地点、试题分数等信息。

报考记录数据：报考记录数据包括每个学生报名的考试、考试时间、缴费状态等信息。

缴费记录数据：缴费记录数据包括每个学生缴费的金额、时间、缴费方式等信息。

题库数据：题库数据包括所有考试客观题与主观题信息、由教师录入管理。

用户登录信息数据：用户登录信息数据包括用户的登录名和密码等信息。

系统日志数据：系统日志数据包括系统操作日志和异常日志等信息，用于系统管理和监控。

8.2 数据流图

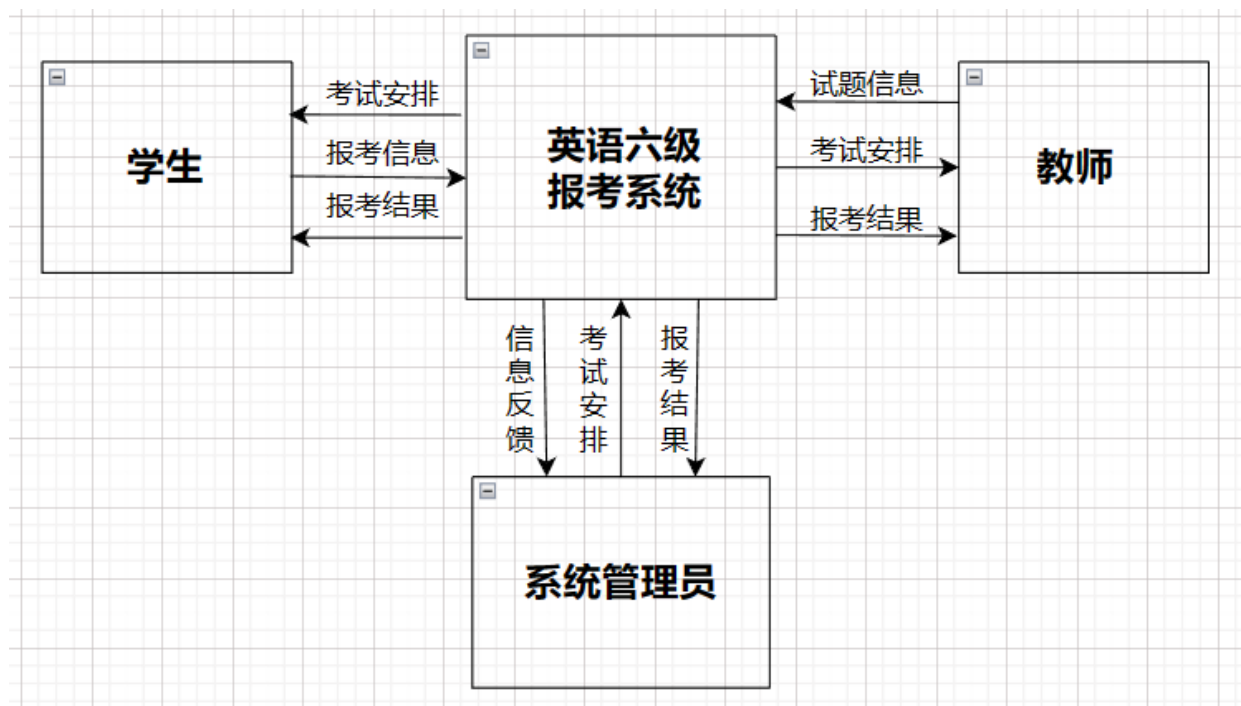


图 8.7: 数据流图

8.3 数据字典

• 答卷信息表 (answer):

- Examiner_ID: 出题人 ID, int 类型, 非空。
- ExamPaper_ID: 试卷 ID, int 类型, 非空。
- Examinee_ID: 考生 ID, int 类型, 可空。
- Choice_A: 选择题答案, varchar 类型, 长度为 1024, 可空。
- Subjective_A: 主观题答案, varchar 类型, 长度为 4096, 可空。
- Seperate_S: 分数明细, varchar 类型, 长度为 512, 可空。
- Gross_S: 总分数, varchar 类型, 长度为 20, 可空。
- Whe_Take_Test: 是否参加考试, int 类型, 可空。
- Whe_Pay_Fees: 是否缴纳考试费用, int 类型, 可空。

• 选择题表 (choice):

- Choice_ID: 选择题 ID, int 类型, 非空。
- Examiner_ID: 出题人 ID, varchar 类型, 长度为 20, 非空。
- Choice_Stem: 选择题题干, varchar 类型, 长度为 1024, 可空。
- Choice_Options: 选择题选项, varchar 类型, 长度为 1024, 可空。
- Choice_Answer: 选择题答案, varchar 类型, 长度为 20, 可空。

- **主观题表 (subjective):**

- Subjective_ID: 主观题 ID, int 类型, 非空。
- Examinee_ID: 考生 ID, varchar 类型, 长度为 20, 非空。
- Subjective_Stem: 主观题题干, varchar 类型, 长度为 1024, 可空。
- Subjective_Answer: 主观题答案, varchar 类型, 长度为 1024, 可空。

- **管理员信息表 (manager):**

- ID: 管理员 ID, int 类型, 非空。
- Name: 管理员姓名, varchar 类型, 长度为 10, 非空。
- Gender: 管理员性别, varchar 类型, 长度为 10, 非空。
- Email: 管理员邮箱, varchar 类型, 长度为 40, 非空。
- Tel: 管理员电话号码, varchar 类型, 长度为 20, 非空。
- Password: 管理员密码, varchar 类型, 长度为 20, 非空。

- **教师信息表 (examiner):**

- ID: 出题人 ID, int 类型, 非空。
- Name: 出题人姓名, varchar 类型, 长度为 10, 非空。
- Gender: 出题人性别, varchar 类型, 长度为 10, 非空。
- Email: 出题人邮箱, varchar 类型, 长度为 40, 非空。
- Tel: 出题人电话号码, varchar 类型, 长度为 20, 非空。
- Password: 出题人密码, varchar 类型, 长度为 20, 非空。

- **学生信息表 (examinee):**

- ID: 考生 ID, int 类型, 非空。
- Name: 考生姓名, varchar 类型, 长度为 10, 非空。
- Gender: 考生性别, varchar 类型, 长度为 10, 非空。
- Email: 考生邮箱, varchar 类型, 长度为 40, 非空。
- Tel: 考生电话号码, varchar 类型, 长度为 20, 非空。
- Password: 考生密码, varchar 类型, 长度为 20, 非空。

- **试卷信息表 (exampaper):**

- ExamPaper_ID: 试卷 ID, int 类型, 非空。
- ExamPaper_name: 试卷名称, varchar 类型, 长度为 255, 可空。
- ExamDate: 考试日期, varchar 类型, 长度为 20, 可空。
- Choice_Num: 选择题数量, varchar 类型, 长度为 20, 可空。
- Choice_IDs: 选择题 ID 列表, varchar 类型, 长度为 255, 可空。
- Subjective_Num: 主观题数量, varchar 类型, 长度为 20, 可空。
- Subjective_IDs: 主观题 ID 列表, varchar 类型, 长度为 255, 可空。

9 功能需求

9.1 功能划分

在该《英语六级考试报考系统》中，主要包括以下功能模块：

- 用户管理功能：包括用户注册、登录、密码找回等功能，还包括用户基本信息的增删改查，例如学生和教师的个人信息。
- 试卷管理功能：包括试卷信息的增删改查，例如试卷名称、试题信息、考试时间和地点等。
- 题库管理功能：包括录入和修改功能，同时还包括不同种类题目结果的汇总。
- 信息展示功能：包括对试卷信息的展示、对报考结果的展示。
- 系统管理功能：包括对系统整体的管理，例如数据备份、数据恢复、权限管理等。

9.2 功能描述

9.2.1 管理员端

- 考试安排：输入被安排考试的名称、考试地点和时间信息。系统对提交的试卷信息进行验证，验证成功后，将试卷信息录入数据库并更新数据库。
- 基本资料管理：管理学生、教师和课程的试卷属性信息。系统对提交的学生、教师和试卷信息进行检查和验证，验证通过后将相关信息录入数据库并更新数据库。输出学生、教师和试卷的基本信息表。
- 设置各级用户的权限：对各级用户进行权限设置，系统对这些权限进行验证，通过验证后更新系统。给出所设置的用户的权限说明。

9.2.2 教师端

- 试题录入：可以在教师端主页进行试题的录入，包括主观题与客观题，录入的信息将会自动提交到数据库。
- 在线组卷：可以在教师端主页进行组卷，包括客观题与主观题，录入的信息将会自动提交到数据库。
- 试卷批改：对于已经结束的考试，教师可对学生答卷的主观题进行批改，批改完成后将由系统自动统计考生的主观题与客观题分数，并给出总分。

9.2.3 学生端

- 考试报名：在学生端主页面可以查看当前可报名的考试，若符合报考条件即可进行报名。
- 考试缴费：对已经报名考试可以进行缴费操作。
- 在线答题：考生可在规定的时间内在学生端主页进入对应考试的在线答题页面，可自主切换题目进行作答，系统提供考试的计时的功能，考生须在规定时间内作答，超时自动交卷。

10 非功能性需求

10.1 用户体验需求

- 页面内容：页面内容突出主题，栏目、菜单布置合理，内容丰富，文字准确，行文格式统一规范。
- 导航结构：具有明确的导航指示，便于理解，便于用户使用。
- 艺术风格：界面、版面形象清新悦目，布局合理，字号大小适宜，字体选择合适，动静效果好，色彩和谐，与主题相吻合。

10.2 可靠性需求

- 具有高可靠性，确保数据的完整性和准确性。具有数据备份和恢复机制，确保数据的安全性和可恢复性。

10.3 性能需求

- 能够快速响应用户请求，保证系统的高并发性。能够支持大规模的数据存储和处理，保证系统的高可扩展性。

10.4 可用性需求

- 易于使用，不需要用户特殊的技术背景。能够提供良好的用户反馈，帮助用户快速解决问题。能够提供适当的帮助文档和指导，帮助用户快速了解系统的功能和操作方法。

11 系统运行要求

11.1 硬件配置要求

下面是关于英语六级考试报考系统的硬件配置要求：

- 服务器：至少 1 台 24 小时运行的具有 2GHz 或更高处理器速度的服务器，16GB 或更高内存，200GB 或更高硬盘容量，支持 RAID 磁盘阵列，支持冗余电源，以确保系统能够稳定、流畅地运行；
- 数据库服务器：至少 1 台 24 小时运行的具有 2GHz 或更高处理器速度的数据库服务器，8GB 或更高内存，500GB 或更高硬盘容量，支持 RAID 磁盘阵列，支持冗余电源，以保证数据存储的可靠性；
- 客户端电脑：至少具有 1GB 内存，128MB 硬盘空间，能够连接到互联网，安装有谷歌、火狐等较新版本的常用浏览器，同时支持 1280×800 分辨率的显示器；

11.2 软件配置要求

下面是关于英语六级考试报考系统的软件配置要求：

- 操作系统：Windows 7 及以上版本、macOS 10.12 及以上版本、Linux；

- 数据库管理系统：MySQL5.0 及以上版本、Oracle Database 11 及以上版本、SQL Server 2019 及以上版本等主流数据库系统；
- Web 服务器：Apache Tomcat 8.0 或更高版本；
- 前端框架：Bootstrap、jQuery 等主流框架；
- 后端框架：JavaEE、Spring、Spring MVC 等主流框架；
- 版本控制工具：Git 等主流版本控制工具；
- 开发工具：Eclipse、IntelliJ IDEA 等 Java 开发工具，Visual Studio Code、Sublime Text 等前端开发工具；
- 其他辅助工具：Maven、Gradle 等构建工具，Jenkins、Travis CI 等持续集成工具；

12 系统设计

12.1 模块划分

下面是关于英语六级考试报名系统的模块划分：

- 学生模块：主要包括学生注册、登录、报名、缴费等功能。
- 管理员模块：主要包括信息管理、审核学生报名信息、发布考试通知等功能。
- 教师模块：主要包括试题录入、题库管理、在线组卷、在线批改等功能。
- 考试模块：主要包括考试时间安排、考场安排、在线考试等功能。

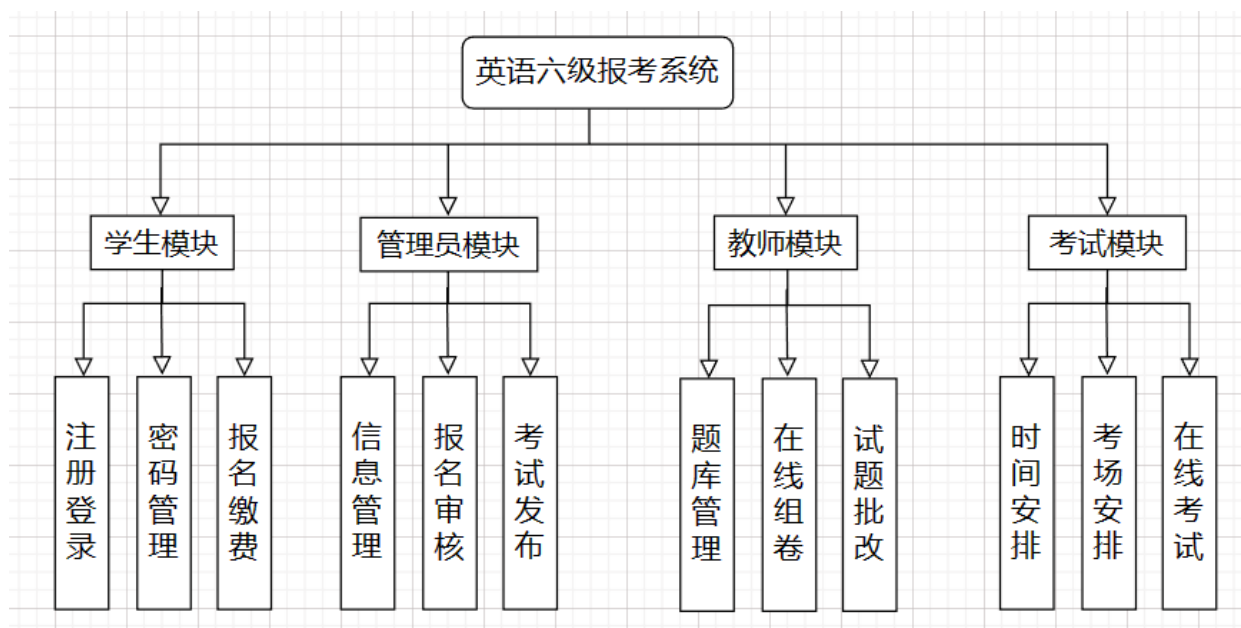


图 12.8: 模块划分图

12.2 系统架构

下面是关于英语六级考试报考系统的系统架构图：

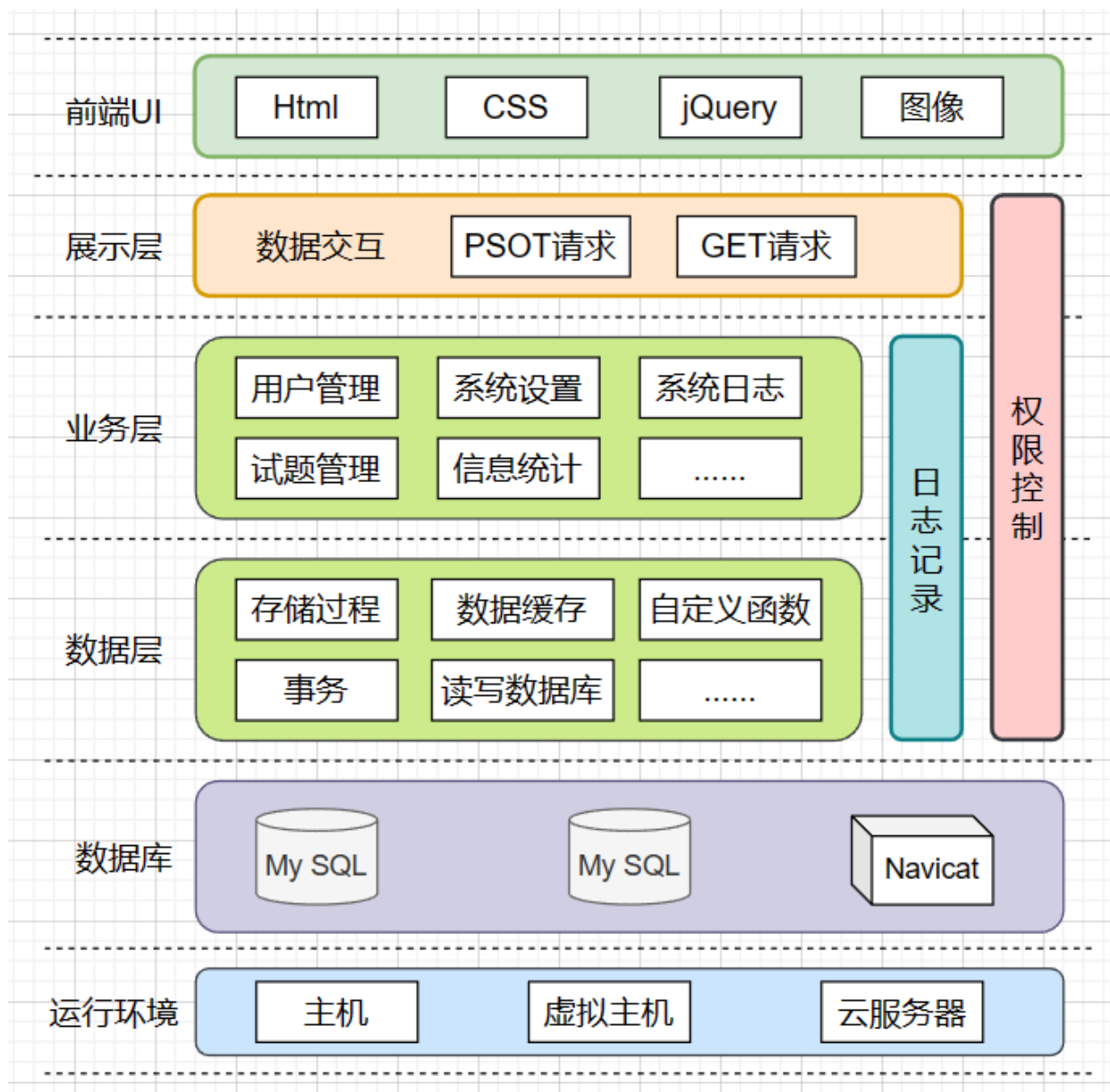


图 12.9: 系统架构图

12.3 数据流程

英语六级报考系统的数据流程如下：

• 报考信息流程：

- 考生注册：考生在系统中注册账号，输入个人信息并提交。
- 考生登录：考生使用注册时填写的账号和密码登录系统。
- 报名信息录入：考生登录后，在系统中填写报名信息并提交。
- 缴费信息录入：考生缴费后，系统自动收集信息并提交。

- 信息存储：系统将考生提交的报名信息、缴费信息存储到数据库中。
- 试卷信息发布：规定时间内，考生在系统中查看考卷并作答提交。
- 成绩信息录入：考试结束后，管理员将考试成绩录入系统。
- 成绩信息存储：系统将管理员录入的成绩信息存储到数据库中。
- 成绩查询：用户登录后，在系统中查询自己的成绩信息。

• **组卷信息流程：**

- 教师注册：教师在系统中注册账号，输入个人信息并提交。
- 主观题录入：教师在系统中录入主观题信息并提交。
- 客观题录入：教师在系统中录入客观题信息并提交。
- 信息存储：系统将教师提交的题目信息存储到数据库中。
- 在线组卷：教师浏览数据中的题库，选择题目组卷并提交。
- 信息存储：系统将教师提交的试卷信息存储到数据库中。

12.4 数据库设计

为了实现需求分析中要求的各类功能，我们需要建立一个强大的，能够为各项功能提供支持，且简单易懂的数据库系统，经过前期小组成员的不断讨论，以及后续的项目实践过程，我们最终确定了由七个数据表构建的数据库系统，其类图如下图所示：

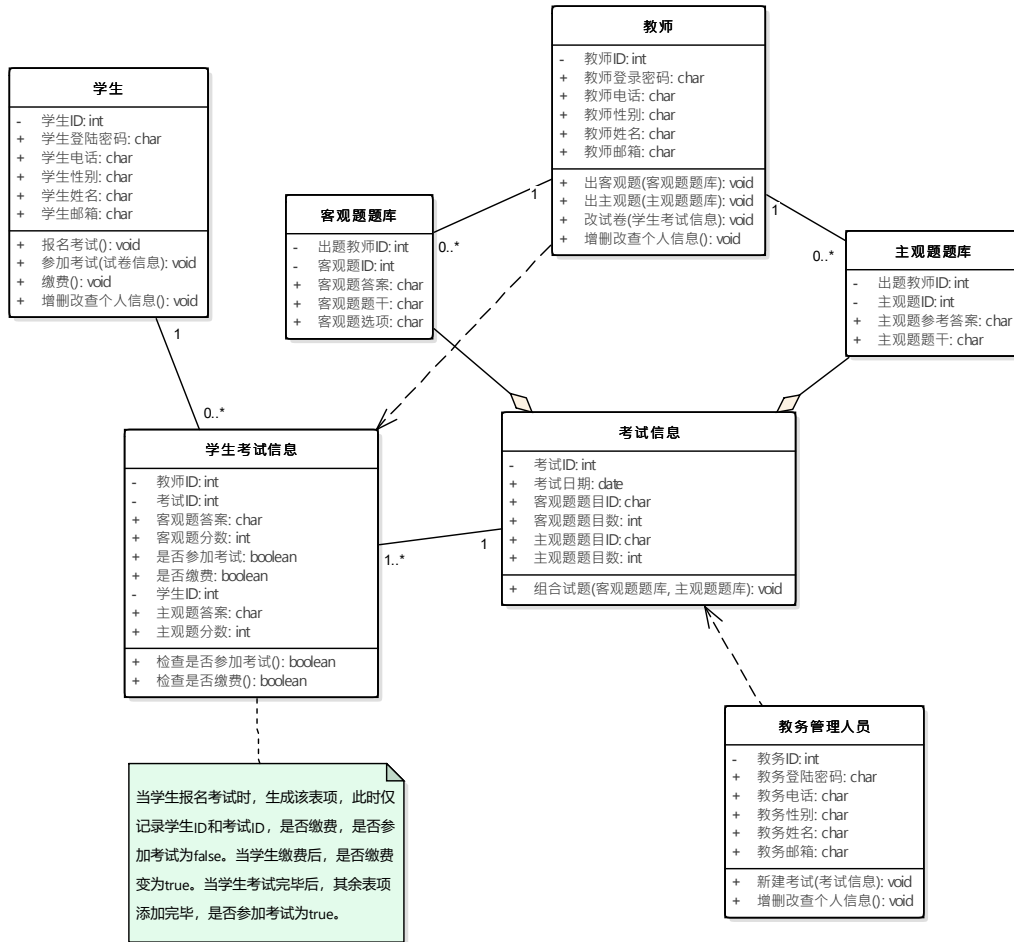


图 12.10: 类图

其中，**学生表**，**教师表**，**教务管理人员表**用于分别存储三种用户的个人信息，并支持个人信息的增删改查，这三张表分别存储了三种用户的用户 ID 及其账户密码，为系统的登录和注册功能提供了数据支持。

客观题题库表以及**主观题题库表**储存了考试需要用到的所有题目信息，分别包括客观题题干，客观题选项，客观题答案；以及主观题题目，和主观题参考答案。这两张表由教师用户负责维护，教师用户可以对这两张表的信息进行增删改查操作。这两张表为系统所需要的题库管理功能，组卷功能，以及客观题自动批改功能提供了数据库支持。在组卷的过程中，教师能够使用这两张表中存储的题目进行试卷的组成。又因为客观题题库表中存储了客观题的答案，故当学生提交试卷后，系统即可查询对应答案，对学生试卷进行自动批阅。

考试信息表对一次考试所需包含的信息提供了数据库支持。该表由教务管理人员负责进行添加操作，其负责维护该表项的考试时间信息。当教务管理人员添加表项后，教师需要为本次考试进行组卷操作。体现在数据库表项中，即教师对该表的客观题题目 ID 和主观题题目 ID 两个属性进行增删改查操作，而系统自动根据表项中题目的数量修改主观题题目数和客观题题目数。当考试信息表中的所有内容填写完毕后，即新的一场考试发布后，学生用户即可对该表进行查询操作，查看可以参加的考试。

学生考试信息表对学生的一次考试所需要进行的操作提供了数据库支持。当学生报名考试时，添加该表项，此时仅记录学生 ID 和考试 ID，是否缴费，是否参加考试为 false。当学生缴费后，是否缴费变为 true。当学生考试完毕后，其余表项添加完毕，是否参加考试为 true。根据以上逻辑可以发现，

学生能否参加考试取决于是否缴费表项的值。当学生参加完毕考试后，该表项将继续发挥其作用，在教师 and 系统批阅完毕学生的试题后，该表项将会储存学生的各项考试分数，学生可以在系统中对该表的相关表项进行查询操作，获取自己的成绩信息。

综上所述，系统的数据库使用简单的七个表完全提供了系统各项功能所需的数据库资源，为系统功能的实现打下了坚实的基础。

12.5 UI 设计

在六月二十一日展示前，我们完成了初版的 UI 设计，由于时间较为仓促，并且我们小组成员一致认为后端的实现相较于 UI 实现更为重要且迫切，故 UI 的设计更加侧重于简单和易于实现，而忽略了 UI 美观的重要性，且并没有重点突出功能。在小组展示后，惊叹于其他小组堪比毕业设计级别的 UI 设计，我们重新学习了 HTML，CSS 语言的编写，完成了第二版 UI 设计。下面我们将分别从登录/注册界面，学生端，教师端，以及教务管理人员端分别介绍 UI 设计。

12.5.1 登录/注册界面



图 12.11: 旧版登录界面



图 12.12: 新版登录界面

登录界面的初版 UI 设计及新版 UI 设计差别较小，其主要区别为新版 UI 设计更加向用户强调了页面的具体功能，在页面上部提示用户该界面用于登录和注册操作，且在左上方标识了该系统服务于全国大学生四、六级考试。在该页面中，三种用户可以输入自己的账户和密码，登录至对应的系统中。而对于学生用户，其可以点击注册按钮，进入注册界面进行学生端账户的注册业务中。

The image shows the old version of the student registration interface. It features a light blue background. In the center, there is a white rectangular form titled "注册信息" (Registration Information). The form contains several input fields: "用户名:" (Username), "密码:" (Password), "邮箱:" (Email), and "电话:" (Phone). Below these are two dropdown menus: "性别:" (Gender) with "男" (Male) selected, and "选择身份:" (Select Identity) with "学生" (Student) selected. At the bottom of the form is a green button labeled "注册" (Register).

图 12.13: 旧版学生注册界面

The image shows the new version of the student registration interface. It has a light blue background. At the top, there is a dark blue header bar containing the "CET" logo, the text "全国大学英语四、六级考试" (National University English Four and Six Level Exam), and a "注册" (Register) button. Below the header, the registration form is centered. It is titled "注册信息" (Registration Information) and includes the same input fields as the old version: "用户名:" (Username), "密码:" (Password), "邮箱:" (Email), and "电话:" (Phone). It also has the same dropdown menus: "性别:" (Gender) with "男" (Male) selected, and "选择身份:" (Select Identity) with "学生" (Student) selected. A green "注册" (Register) button is at the bottom of the form.

图 12.14: 新版学生注册界面

在学生注册页面中，学生需要填写自己的用户名，密码，邮箱，电话，性别等信息完成自身账户的注册。新版和旧版 UI 设计均体现了学生需要填写的信息内容，而新版 UI 相较于旧版 UI 更在网页上部凸显了该页面的功能为注册。

12.5.2 学生端

在初版学生端 UI 设计中，当学生通过登录页面登录系统后，其首先看到的页面如下所示：

试卷列表			
考试ID	考试名称	考试日期	操作
1	CET6-2002-06	2002-06-14	报名考试 考试缴费 进入考试
2	CET6-2023-03	2023-03-05	报名考试 考试缴费 进入考试
3	CET6-2023-06	2023-06-15	报名考试 考试缴费 进入考试
查看历史分数			

图 12.15: 旧版学生登录后界面

可以发现，该页面直接跳转至了学生报名参加考试的页面，在该页面中，学生可以查看自己能够报名参加的考试，在报名考试后，可以进一步进行缴费操作，最后进入到正式的考试中。在完成考试后，学生可以点击页面最下角的查看分数按钮，跳转至查看历史分数的页面，如下所示：

试卷列表						
学生ID	学生姓名	考试ID	考试日期	主观题总分	客观题总分	总分
2010764	gaosensen	1	2002-6-14	75.5	413.5	489
2010764	gaosensen	2	2002-12-15	68	503	571

图 12.16: 旧版学生查看成绩界面

可以非常直观的看出，这样的 UI 设计极为的不美观，学生虽然能够理解系统的使用方法，但存在诸多如按钮过小，功能不明确不具体，学生难以在短时间内了解系统的所有功能等问题。故在新版 UI 的设计上，当学生登录系统后，其首先会进入其业务大厅，如下图所示：



图 12.17: 新版学生登陆后界面

在该页面下，学生能够清晰的了解系统所具有的全部功能，且功能下方均有具体的功能描述，使得学生能够在第一时间就掌握系统的使用方法。当学生点击考试报名后，其会跳转至原先的登陆后界面的新版，如下图所示：



图 12.18: 新版学生考试报名界面

在该页面下，学生能够进行报名参加考试的相关操作，而原先位于该页面的查询成绩功能被移动至业务大厅中。其在业务大厅点击对应按钮跳转后的界面如下所示：



图 12.19: 新版学生查询成绩界面

这样的 UI 布局和跳转逻辑，极大的减轻了学生的系统使用的学习成本，并且更加凸显了系统的各项功能，并使得 UI 更加的美观，是对系统的成功升级。

此外，当学生点击进入考试后，会进入到答题界面，初版答题界面如下所示：

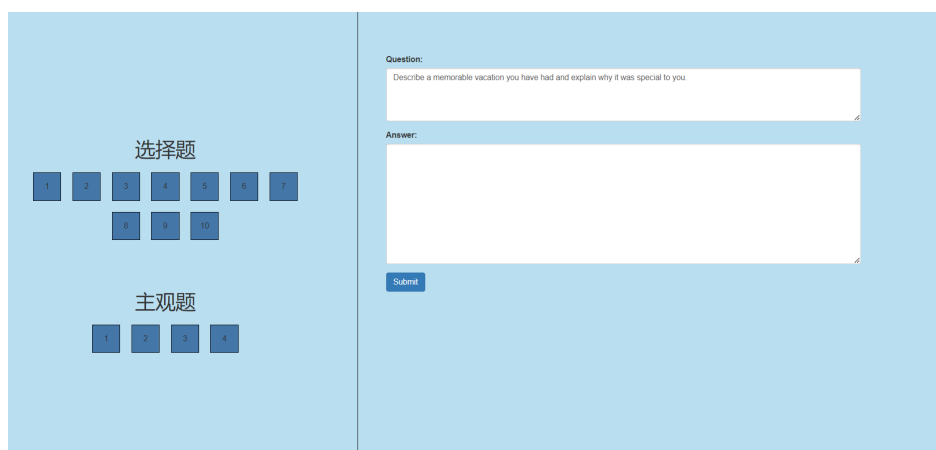


图 12.20: 旧版学生答题界面

学生通过左侧选择题目序号，而在右侧进行答题操作。但这样的 UI 设计同样存在问题，一方面，该页面并不显示考试剩余时间，学生难以掌握考试剩余时间，继而将影响其答题；另一方面，题目的字体过小，且并不突出，学生答题过程中的审题环节将耗费许多不必要的精力。明确修改方向后，我们对 UI 进行了如下改进：

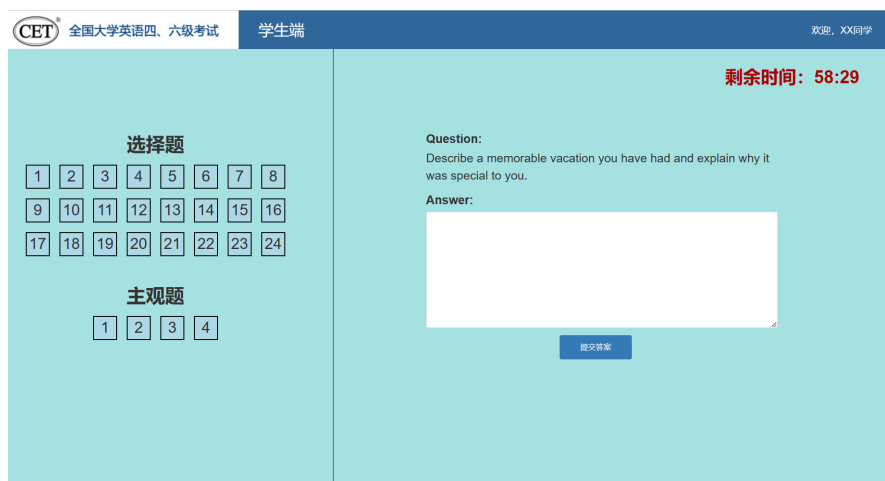


图 12.21: 新版学生答题界面

在新版界面中，我们添加了考试剩余时间的显示，且优化了答题题目 UI，现在题目的字体较旧版页面放大许多，学生能够清晰的阅读题目，并进行作答。

以上是学生界面的 UI 设计，通过使用如上的 UI，学生能够轻松使用系统中的各项功能。

12.5.3 教师端

当教师通过登录页面登录系统后，其在旧版 UI 设计中将会进入如下页面：



图 12.22: 旧版教师登录后页面

该页面显示了教师系统所具备的四个功能，分别为添加客观题题库，添加主观题题库，组卷和批阅主观题。但非常的显然，旧版 UI 的界面过于紧缩在一个极小的范围内，而浪费了极大的页面空间。较小的字体也不利于教师阅读系统的功能，故我们设计了学生端相似的教师端业务大厅，如下所示：



图 12.23: 新版教师登录后页面

该页面添加了各个功能的简要描述，并充分利用了页面的大小，且更为美观，方便教师用户使用系统提供的各项功能。

当教师点击添加客观题或者添加主观题按钮时，其实际是向题库中添加题目，其旧版页面如下所示：

Objective Questions

Question:

Option A:

Option B:

Option C:

Option D:

Correct Answer:

☒ Option A

☐ Option B

☐ Option C

☐ Option D

Submit

图 12.24: 旧版教师录入客观题页面

Subjective Questions

Question:

Answer:

Submit

图 12.25: 旧版教师录入主观题页面

教师通过填写对应输入框中的内容，来分别添加客观题的题干，选项和正确答案；主观题的题干和参考答案。新版 UI 延续了旧版 UI 的设计，并将所有的英文标题全部改为中文，并进一步美化且清晰化了页面 UI，方便教师理解每一个输入框的含义，新版 UI 如下所示：

CET 全国大学英语四、六级考试 教师端 欢迎, jiaoyuhang老师

主观题录入

题干录入:

参考答案录入:

提交录入信息

图 12.26: 新版教师录入客观题页面



图 12.27: 新版教师录入主观题页面

当教师在业务大厅点击查看题库/在线组题时，教师能够对特定试卷进行组卷操作，并能够对题库中的表项进行查询和删除操作，旧版组卷界面如下所示：

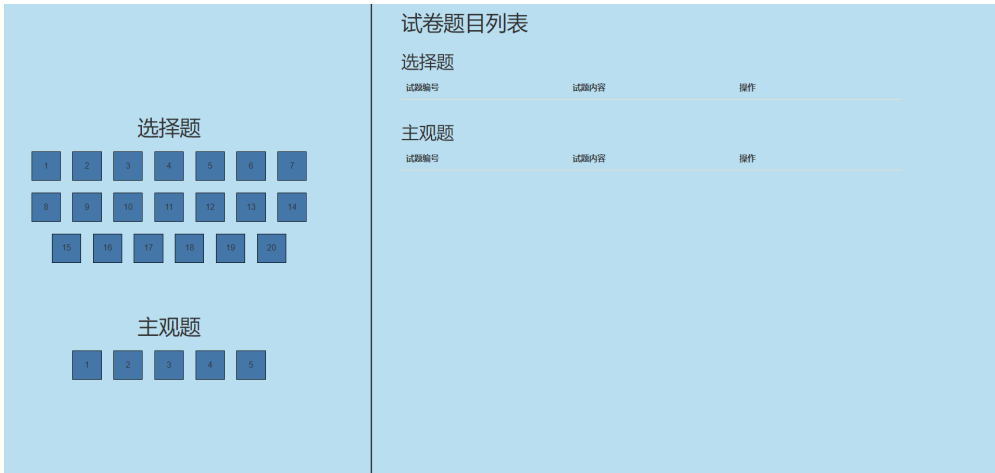


图 12.28: 旧版教师组卷页面

教师能够在左侧页面查看题库中已有的主观题和客观题题目详情，并选择合适的题目加入到右侧的题目列表中。若教师认为题库中的题目有错误或已经过时，也可通过左侧页面对题目进行删除操作。新版 UI 界面延续了旧版 UI 界面的布局 and 实现逻辑，并在此基础上进一步美化了 UI，如下图所示：



图 12.29: 新版教师组卷页面

最后，教师能够通过业务大厅点击批改主观题按钮进入批改主观题页面，在该页面中，教师通过选择左侧学生的答卷，并在右侧看到学生答卷的内容，并在其答案下方录入学生对应的分数，其旧版 UI 界面如下所示：

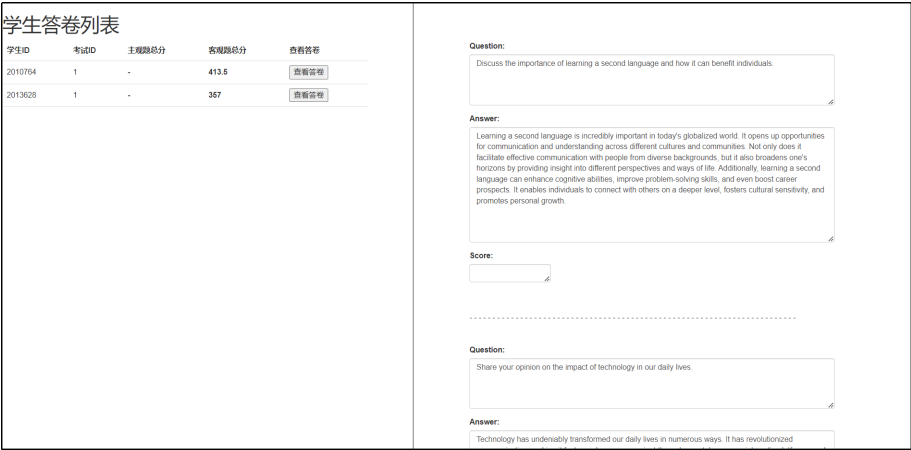


图 12.30: 旧版教师批阅主观题页面

而新版 UI 界面同样在此基础上对组件布局进行了微调，并对系统的整体 UI 风格进行了统一，包括更改背景颜色，增加上方显示栏等，进一步美化 UI 设计，新版界面如下所示：



图 12.31: 新版教师批阅主观题页面

12.5.4 教务管理人员端

教务管理人员能够管理考试信息，教师信息以及学生信息。在初版的 UI 设计中，考虑到教务管理人员较为熟悉系统的各项操作，相较于学生和教师需要较多的功能使用指引，其对于指引的需求不高，而更加需要更加方便快捷的操作系统，故我们为其设计的 UI 极为突出了功能性和便捷性，教务管理人员能够在一个页面中实现对考试信息，学生信息和教师信息的增删改查操作。



图 12.32: 旧版教务管理人员界面

这样的 UI 设计虽然使得教务管理人员能够极为快捷的操作系统，但过于铺陈的 UI 布局容易导致教务管理人员长时间使用系统而增加疲劳感，误操作的可能性也较高。故在新版 UI 中，我们在延续便捷性，功能性的 UI 设计总体思路不变的前提下，减少了一个页面的内容量。在教务管理人员页面中，教务管理人员能够通过点击上方的三个按钮，快速切换至考试，学生和教师的管理页面中，并实现增删改查的操作。同时，页面上方添加了欢迎管理员进入系统的字样，增加了管理人员使用系统的愉悦感。轻量扁平化的 UI 设计也能够有效缓解教务管理人员长时间工作可能导致的视觉疲劳。



图 12.33: 新版教务管理人员界面

13 系统测试

13.1 测试环境

1. 硬件配置: Intel Core i7-8700K 处理器, 16GB DDR4 内存, 1TB SATA 硬盘
2. 软件环境: 操作系统: Windows 10 Pro 版本 21H1, 数据库管理系统: MySQL 8.0.26, Web 服务器: Apache Tomcat 9.0.52, 浏览器: Google Chrome 91.0.4472.124。
3. 网络配置: localhost, 防火墙关闭, 允许内部网络通信
4. 测试数据: 数据库各张表都填充若干行表项
5. 配置和部署: 见“项目开发环境”部分。
6. 安全性和权限: 用户角色: 系统管理员、普通用户、测试用户。
访问权限: 系统管理员具有完全访问权限, 教师和学生有限制的访问权限。
7. 限制和假设: 并发用户数限制: 最大支持 100 个并发用户; 数据容量限制: 数据库最大支持 1TB 的数据存储。
8. 预期结果: 系统行为: 登录成功后显示用户主页, 数据输入验证应显示相应的错误提示, 功能测试应符合预期的业务逻辑。性能指标: 登录响应。

13.2 功能测试

- 用户登录

测试步骤:

打开浏览器, 输入系统的 URL, 即 localhost:65530//login.do。在登录页面, 输入有效的用户名和密码。点击登录按钮。

验证登录结果: 预期结果: 成功登录后跳转到用户主页。实际结果: 确认是否跳转到用户主页。如果成功登录, 记录通过; 如果登录失败, 记录失败原因。

- 注册

在登录页面点击注册。

输入有效的用户信息，如用户名、密码、电子邮件等。

点击保存按钮。

验证新用户创建结果：

预期结果：新用户应成功创建，并在用户列表中显示。实际结果：确认新用户是否成功创建，并在用户列表中找到相应的用户记录。如果新用户成功创建，记录通过；如果创建失败或未在列表中找到用户记录，记录失败原因。

- 管理员功能：

登录到管理员界面，并验证增删改查功能。

- 分别点击考试信息、学生信息、教师信息管理按钮，应能看到相关的信息，分别呈表状显示。
- 点击三张表下方的添加按钮，填写新记录的信息，然后应该能在数据库相应表中查看到新增的记录，并显示在前端。
- 点击每张表每一行右侧的删除按钮，相应记录应该在数据库相应表中被删除，并且前端不再显示这条记录。
- 鼠标双击任意表中的某一格，可以编辑（修改）相应信息，在数据库相应表中可以看到更新后的信息，并且显示在前端。

- 教师相关功能

登录到教师界面，并验证教师功能。

- 点击录入主观题/录入客观题，可以填写给出的几条信息，在题库中增加新的题目，并且在数据库试卷表中体现出来。
- 点击查看题库/在线组题，输入考试号之后可以题库中的所有题目，并且可以查看相应试卷中的所有题目，还可以对题库进行添加或删除、对试卷的题目进行添加或删除，进行添加或删除之后应能在数据库中体现出变化。
- 点击批改主观题，可以查看每场考试的作答试卷，并手动批改其中的主观题、自动批改其中的客观题，在进行打分并提交后会体现在数据库中试卷表。

- 学生相关功能

登录学生界面，并验证学生功能。

- 点击修改个人信息，可以进行相应修改并同时体现在前端网页和后端数据库。
- 点击成绩查询，可以看到已经完成的试卷以及得分。
- 学生还可以报名考试、缴费，且数据库相应字段发生变化。
- 进入考试后，学生可以进行题目作答，点击提交试卷后，数据库的试卷表发生变化。

- 上述测试可以全部通过并且代码覆盖率达到 98.17%。

13.3 性能测试

- 更新、添加、删除用户数据一次用时约 0.039 秒。
- 登录、注册用户一次用时 0.5 秒（手动设置的延时）。

13.4 编程规范

后端使用 JAVA 进行编程时，我们制定并遵循了以下编程规范：

- 代码格式化：
 - 使用一致的缩进，通常是使用 4 个空格进行缩进；
 - 对代码块使用大括号，即使它只包含一条语句；
 - 在二元操作符周围使用空格，例如赋值运算符、比较运算符等；
- 命名规范：
 - 使用有意义的变量、方法和类名，遵循驼峰命名法；
 - 类名使用名词或名词短语，方法名使用动词或动词短语；
- 类和方法设计：
 - 遵循单一职责原则（SRP），每个类和方法应该只负责一项任务；
 - 使用面向接口编程，将接口和实现分离，以便于扩展和替换实现；
 - 尽量使用封装，将类的内部实现细节隐藏起来，通过公共方法来访问和操作数据；
- 性能考虑：
 - 避免在循环体中进行耗时的操作，尽量将其移到循环外部；
 - 避免频繁的对象创建和销毁，尽量重用对象；
 - 避免频繁地进行 IO 操作，尽量将数据先缓存到内存中，最后再一并写到数据库内；
 - 避免频繁的对象创建和销毁，尽量重用对象；

以管理员页面的控制层代码为例，分析本项目的编程规范：

```
1  RequestMapping("admin.do")
2  public String admin(Model model, ..... String studentPassword) {
3      System.out.println("admin.do"); //打印一定的日志信息；
4
5      // 使用 switch 代替 if-else, 提升代码可读性：
6      switch (action) {
7          case "addExam": {
8              userService.addExam(examName, examDate); // 变量均使用驼峰命名法命名方法
9              break;
10         }
```

```

11     case "updateExam": {
12         if (!examName.isEmpty()) { //调用相应的函数接口进行功能实现
13             userService.updateExam(examId, examName, "ExamPaper_name");
14         }
15         if (!examDate.isEmpty()) {
16             userService.updateExam(examId, examDate, "ExamDate");
17         }
18         break;
19     }
20     case "deleteExam": {
21         userService.deleteExam(examId);
22         break;
23     }
24     .....
25     .....
26     default: { //对代码块使用大括号，即使它只包含一条语句；
27         .....
28     }
29 } //缩进符合规范；
30
31 // 必要的空行，提升代码的美观性：
32 List<ExamInfo> examList = userService.getExam(); //调用服务层封装的接口进行处理
33 model.addAttribute("examList", examList);
34
35 List<ExameeInfo> studentList = userService.getStudent(); // 使用驼峰命名法命名变量
36 model.addAttribute("studentList", studentList);
37
38 List<ExamerInfo> teacherList = userService.getTeacher();
39 model.addAttribute("teacherList", teacherList);
40
41 return "admin";
42 }

```

13.5 性能优化

为了应对较多用户使用的情形，我们需要对程序的性能进行优化。在选题功能实现的过程中，我们为了避免频繁地进行 IO 操作导致性能浪费，我们尽量将数据先缓存到内存中，最后再一并写到数据库内：

```

1  RequestMapping("viewBankAndSelectQuestion.do")
2  public String viewBankAndSelectQuestion(Model model, .....){
3      List<SubjectiveQuestion> subSrclist = userService.getSubQuestions();

```

```

4      model.addAttribute("subQuestionList", subSrcclist);
5
6      if (examId.length() >= 1) {
7          List<ExamPaper> subSequence = userService.getExamSub(examId);
8          if (subSequence.isEmpty()) {
9              List<SubjectiveQuestion> examSublist = new ArrayList<>();
10             model.addAttribute("examSublist", examSublist);
11         } else {
12             if (newSub.length() >= 1) { //插入新的主观题;
13                 subSequence.get(0).setSubjective_IDs(subSequence.get(0).getSubjective_IDs()
14                     + "@@" + newSub);
15             }
16             List<Integer> Ids = new ArrayList<>();
17             String[] tokens = subSequence.get(0).getSubjective_IDs().split("@@");
18             for (String token : tokens) {
19                 try {
20                     int number = Integer.parseInt(token.trim());
21                     Ids.add(number);
22                 } catch (NumberFormatException e) { //对异常情形进行处理
23                     System.err.println(" 无法解析为数字: " + token);
24                 }
25             }
26             .....
27             .....
28             //插入新题后, 先将插入的结果传给前端显示, 后续再写入数据库;
29             model.addAttribute("examId", examId);
30             model.addAttribute("examSublist", examSublist);
31             .....
32             .....
33
34             if (newSub.length() >= 1) {
35                 //插入数据库:
36                 userService.changeSubofExam(examId, subSequence.get(0).getSubjective_IDs());
37             }
38         }
39     }
40     return "viewBankAndSelectQuestion";
41 }

```


14 项目管理

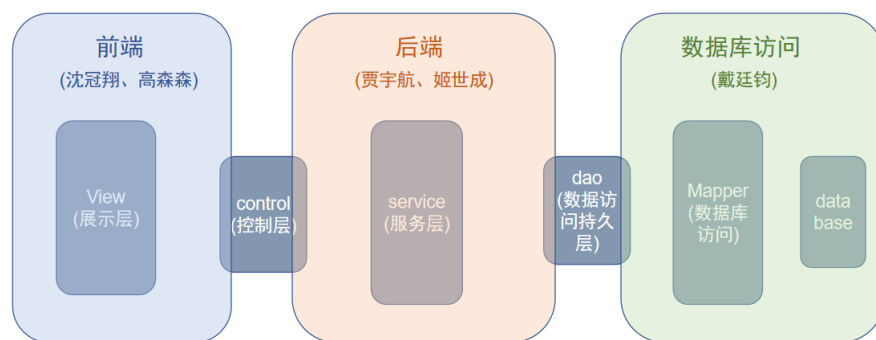
14.1 参与人员及分工

14.1.1 代码实现部分

- 贾宇航: 教师端、学生端后端实现, 前端部分页面后期美化
- 沈冠翔: 管理员端、登录界面前端实现, 前端部分页面后期美化
- 姬世成: 注册功能实现, 管理员端后端实现, 前端部分页面后期美化
- 戴廷钧: 数据库设计实现, SQL 语句实现, 前端部分页面后期美化
- 高森森: 教师端、学生端前端实现, 前端部分页面后期美化

14.1.2 文档内容部分

- 贾宇航: 项目概述、业务流程图、总结
- 沈冠翔: 系统测试
- 姬世成: 需求分析与系统设计
- 戴廷钧: 数据库设计、UI 设计
- 高森森: 项目管理、用户手册



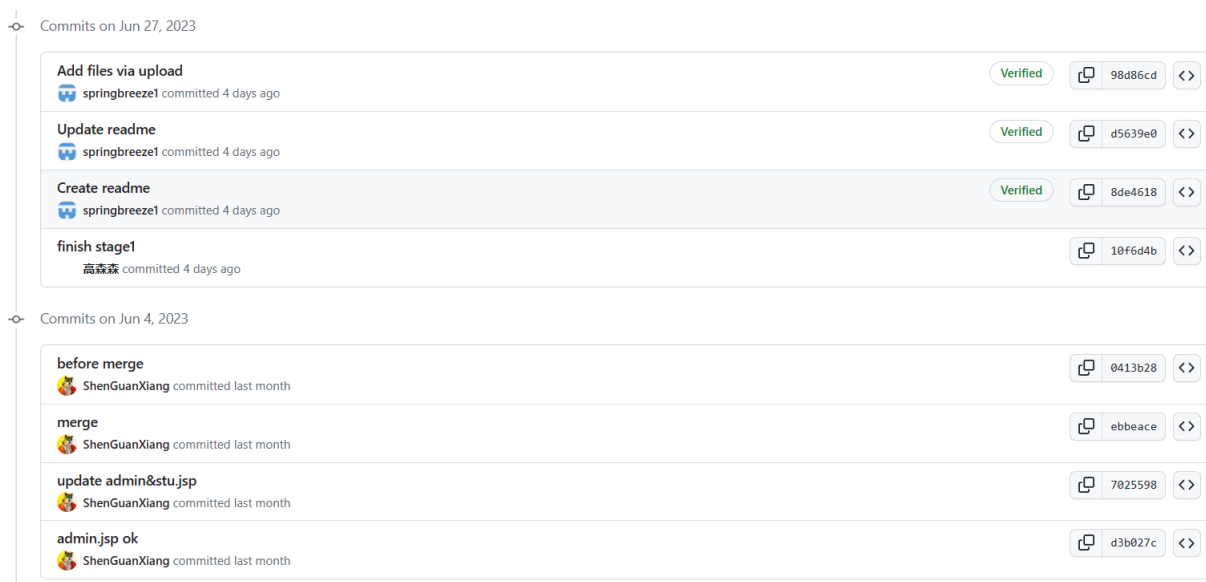
14.2 项目进展记录

- 5月5日, 确定所有的小组成员
- 5月9日, 确定项目使用 SpringBoot 框架
- 5月13日, 小组成员均完成 SpringBoot 框架的学习
- 5月14日, 确定数据库需求
- 5月17日, 戴廷钧同学完成数据库设计
- 5月18日, 进行前后端和 SQL 查询实现的分工

- 5 月 31 日，学生端、管理员端所有功能实现
- 6 月 2 日，教师端所有功能实现
- 6 月 3 日，准备展示 PPT 和录制视频
- 6 月 5 日，项目展示
- 6 月 7 日，确定要扩展的功能，页面美化的分工
- 6 月 27 日，页面美化完成、扩展功能完成
- 6 月 30 日，所需文档完成

14.3 项目管理工具

在本次实验中，我们使用 git 进行代码版本的管理，我们的远程仓库为CET6-Online，部分提交记录如下图所示：



对于数据库部分，我们使用 navicat 进行数据库的相关设计：



15 总结

本次项目是一个有意义且具有挑战性的开发项目。首先，在需求分析阶段，我们充分理解了教务管理系统的功能需求和用户需求。通过调查和分析现有的在线考试系统的特点，我们明确了其存在的不足，并将其转化为具体的功能模块和设计要求。这个阶段的工作让我们深入思考用户需求、业务流程，并将其转化为系统设计的基础。

其次，在系统设计和架构阶段，我们选择了 Vue + Springboot + mybatis 的技术栈。这种前后端分离的架构使得我们能够更好地进行分工协作，提高开发效率和代码质量。我们合理划分了系统的模块和组件，采用了面向接口编程和封装的设计原则，使得系统的可维护性和扩展性得到了提高。

在具体的开发过程中，我们按照项目计划进行了迭代开发，充分发挥团队成员的优势，分工合作，高效地完成了各个功能模块的开发和集成。在开发过程中，我们严格遵守了编程规范，保持了代码的一致性和可读性。同时，我们也注重代码的性能优化和异常处理，确保系统的稳定性和用户体验。

最后，在项目测试和部署阶段，我们进行了全面的功能测试和性能测试。通过对系统的各个功能模块进行测试，我们发现了一些潜在的问题并及时修复。

在整个项目开发的过程中，我们充分运用了《软件工程》课程中学到的理论知识，例如：

- 开闭原则的应用：我们采用了 Vue + Springboot + mybatis 框架实现前后端解耦，符合开闭原则。前端使用 Vue 作为 UI 框架，实现了良好的界面交互和用户体验；后端使用 Springboot 框架搭建业务逻辑和数据处理的框架；而 mybatis 作为持久层框架实现了与数据库的交互。这样的架构使得系统各个模块之间的解耦性更高，方便分工合作，并能够支持后续的功能扩展和维护。
- 业务流程图在需求分析中的应用：在项目的需求分析阶段，我们绘制了各种业务流程图，例如用

户注册流程、考试报名流程等。这些流程图帮助我们深入理解用户需求，识别各个业务环节，明确功能模块和数据流动，从而更好地进行系统设计和开发。

- 项目的开发流程和版本迭代：在项目中，我们采用了版本控制和迭代开发的方法。通过前后开发两个版本，我们能够逐步完善功能，修复问题，并进行改进。这样的开发流程使得我们能够更好地管理项目，快速响应需求变化，并逐步提升产品质量。
- 其他软件开发原则的应用：在项目中，我们还遵循了其他软件开发原则，如单一职责原则、接口隔离原则、依赖倒置原则等。这些原则帮助我们设计出可维护、可扩展的代码结构，提高代码的可读性和可重用性。

总的来说，通过本项目的开发，我们不仅学习到了软件工程课程中的知识和理论，更重要的是将其应用于实际项目中，锻炼了我们的项目开发能力和团队合作能力。我们在需求分析、系统设计、编码实现和测试验证等方面都得到了实践和提升。这些经验将对我们今后的软件开发工作产生积极的影响，并为我们的职业发展打下坚实的基础。