

**全栈编程**

**学 生 姓 名： 郭昕睿**

**学 号： 32020210145**

**学 院（系）： 会计学院**

**专业（方向）： 会计学（注册会计师专门化）**

**班 级： 注会一班**

**指 导 教 师： 刘丹**

**完 成 时 间： 2023年3月20日**

摘要

当前时代是一个科学、技术与教育相互融合且高速发展的时代，尤其是在计算机相关的领域。随着越来越多的人投身于计算机行业，相关技术也在不断进步[1]。进而，网络教育也在逐渐普及，成为大众的教育方式。后疫情时代，在线教育与课堂教学相结合的混合教学模式将可能成为高校教学的“新常态。而在计算机相关专业的在线教学与考试中，编程题目处于尚未完全克服的领域，由于其主观性较强，学生的解题思路和想法各不相同，目前尚缺乏统一的评分标准，不同的题目、不同的代码风格都会影响到成绩的评判。因此，在计算机领域的教学中，编程题目的在线自动评测一直是研究热点和难点。

基于以上背景，学者对目前传统的编程题目测评系统进行了深入分析，发现存在一些问题。同时，学者对编程题目的测评特点进行了研究，并结合了人工手动测评的理念，提出了一种编程作业混合式的测评模型。该模型综合了面向结果与面向内容的测评方式，旨在实现对编程题目更全面的测评，以期达到更为合理的测评结果。

本文研究了白盒与黑盒两种方式相结合的测评方式对编程题目进行测评。白盒评分部分主要使用字符串比对方式，通过提取题目要求中的关键代码，与学生完成的代码进行白盒评分比对，根据得分点进行综合测评。黑盒评分方式部分使用测试用例，通过与标准测试用例的运行结果相比较，得到最终动态结果得分。两种评分方式在经过用例大量测试后，表现出令人满意的结果。同时，本系统使用了软件工程的设计思想进行开发，对各层次之间的功能做了详细的区分与隔离，保证了系统的高内聚低耦合性，充分展现了系统的功能。

关键词：自动评分；全栈编程题目编程题目技术；白盒测试；黑盒测试

Abstract

The current era is characterized by the integration and rapId development of science, technology, and education, especially in the field of computer science. As more and more people engage in the computer industry, related technologies are also constantly advancing. Furthermore, online education is gradually becoming popular and a popular way of education for the general public. In the post pandemic era, the hybrId teaching model combining online education and classroom teaching may become the "new normal" of university teaching. In the online teaching and examination of computer related majors, programming questions are still in an area that has not been fully overcome. Due to their strong subjectivity, students have different problem-solving Ideas and Ideas, and there is currently a lack of unified scoring standards. Different questions and code styles can affect the evaluation of grades. Therefore, in the field of computer teaching, online automatic evaluation of programming questions has always been a research hotspot and difficulty.

Based on the above background, scholars have conducted in-depth analysis of the current traditional programming question evaluation system and found some problems. At the same time, scholars have studied the evaluation characteristics of programming questions and proposed a hybrId evaluation model for programming assignments, combining the concept of manual evaluation. This model combines result oriented and content oriented evaluation methods, aiming to achieve a more comprehensive evaluation of programming questions and achieve more reasonable evaluation results.

This article investigates a combination of white box and black box evaluation methods for evaluating programming questions. The white box scoring section mainly uses string comparison to extract key codes from the question requirements, compare them with the code completed by students, and conduct a comprehensive evaluation based on the score points. The black box scoring method uses test cases to obtain the final dynamic result score by comparing the running results with standard test cases. The two scoring methods have shown satisfactory results after extensive testing of test cases. At the same time, this system was developed using the design philosophy of software engineering, and detailed differentiation and isolation of functions between different levels were made, ensuring high cohesion and low coupling of the system and fully demonstrating its functionality.

Key Words: Automatic Scoring; Full stack technology；White box test；Black box test

目录

[第1章 绪论 1](#_Toc161856408)

[1.1 研究背景与意义 1](#_Toc161856409)

[1.2 国内外研究现状 1](#_Toc161856410)

[1.3 论文研究目标与研究内容 2](#_Toc161856411)

[1.3.1 研究目标 2](#_Toc161856412)

[1.3.2 研究内容 2](#_Toc161856413)

[1.4 论文组织结构 4](#_Toc161856414)

[1.5 本章小结 5](#_Toc161856415)

[第2章 相关理论与技术介绍 5](#_Toc161856416)

[2.1 前端网页技术——Vue 5](#_Toc161856417)

[2.2 后端框架技术 6](#_Toc161856418)

[2.3 数据库技术 6](#_Toc161856419)

[2.3.1 Mybatis与Mybatis-plus 6](#_Toc161856420)

[2.3.2 数据库选型 6](#_Toc161856421)

[2.4 服务器部署 7](#_Toc161856422)

[2.5 相关测试理论 7](#_Toc161856423)

[2.6 本章小结 8](#_Toc161856424)

[第3章 系统分析 8](#_Toc161856425)

[3.1 全栈编程题目测评系统整体需求 8](#_Toc161856426)

[3.2 系统可行性分析 8](#_Toc161856427)

[3.2.1 技术可行性 8](#_Toc161856428)

[3.2.2 经济可行性 9](#_Toc161856429)

[3.2.3 社会可行性 9](#_Toc161856430)

[3.3 系统功能性需求分析 9](#_Toc161856431)

[3.3.1 学生模块功能需求分析 9](#_Toc161856432)

[3.3.2 教师功能模块需求分析 10](#_Toc161856433)

[3.4 非功能需求分析 11](#_Toc161856434)

[3.4.1 可扩展性 11](#_Toc161856435)

[3.4.2 规范性 12](#_Toc161856436)

[3.5 本章小结 12](#_Toc161856437)

[第4章 系统设计 12](#_Toc161856438)

[4.1 系统架构设计 12](#_Toc161856439)

[4.2 系统模块设计 13](#_Toc161856440)

[4.3 系统数据库设计模块 13](#_Toc161856441)

[4.3.1 系统数据库总体概述 13](#_Toc161856442)

[4.3.2 各表详细内容说明 14](#_Toc161856443)

[4.4 系统详细设计 16](#_Toc161856444)

[4.4.1 登录流程详细设计 16](#_Toc161856445)

[4.4.2 学生作答功能详细设计 17](#_Toc161856446)

[4.4.3 作业审批功能详细设计 17](#_Toc161856447)

[4.4.4 老师编辑题目详细设计 18](#_Toc161856448)

[4.5 本章小结 19](#_Toc161856449)

[第5章 系统实现 19](#_Toc161856450)

[5.1 登录模块 19](#_Toc161856451)

[5.2 学生上传作业实现 22](#_Toc161856452)

[5.3 教师模块 23](#_Toc161856453)

[5.3.1 新增作业模块 23](#_Toc161856454)

[5.3.2 修改删除作业模块 26](#_Toc161856455)

[5.3.3 教师查看作业情况模块 28](#_Toc161856456)

[5.4 自动化测评模块 28](#_Toc161856457)

[5.4.1 黑盒测评模块 29](#_Toc161856458)

[5.4.2 白盒测评模块 30](#_Toc161856459)

[5.5 本章小结 31](#_Toc161856460)

[第6章 系统测试 31](#_Toc161856461)

[6.1 登录/登出模块测试 32](#_Toc161856462)

[6.2 学生模块测试 32](#_Toc161856463)

[6.3 教师模块 33](#_Toc161856464)

[6.4 本章小结 34](#_Toc161856465)

[第7章 总结与期望 34](#_Toc161856466)

[7.1 总结 34](#_Toc161856467)

[7.2 展望 35](#_Toc161856468)

# 绪论

## 研究背景与意义

互联网技术的发展给企业的改革和发展提供了有力的技术支持，不仅改变了传统的生产模式，也提高了企业的生产效率[1]。这种变革在一定程度上不仅丰富了人们的生活，也改变了其方式，其技术涉及的综合性与现代学科产生了深度的交汇。在社会生产和各企业单位中，计算机技术的应用正变得日益普及，高精尖的计算机领域技术也不断投入到国家的军事领域，深刻影响着一个国家的军事实力。而对于各大院校而言，创新创业人才的培养不仅可以促进学生综合能力的提升，使其更好适应未来工作的需要，也能有效提高院校的就业率。

教育是国家发展的重要基石，信息化教育又是现代教育的重要组成部分，信息化教育不仅包括使用计算机和其他数字化技术，还涉及教法改革、在线学习、在线考试等。目前，在线考试已成为一种新的考试模式。传统纸笔考试不利于教师批改，也不利于学生复习查阅[2]。因此在线考试系统旨在为学生提供一个练习和模拟测试的平台，为教师建立一个了解学生学习情况的途径[3]。由于编程涉及到个人逻辑性和思维习惯，很多问题只能通过运行结果来评价。然而，这种评价方式存在不公平的问题，因为多种因素可能导致运行结果不尽如人意，尤其是在编写程序时很小的差距可能导致效果大不相同。因此，为了客观、全面地评估考生的编程能力，开发一套能够分步评分的系统显得非常必要。

## 国内外研究现状

上世纪60年代，美国率先推出自动化阅卷的考试系统，尽管当时仅适用于单选题、判断题等客观性题目的自动评分。在1970年代，美国提出了如何通过计算机而非传统纸质形式进行考试的问题，并展开相关研究。1990年代初，多个州开始制定相互承认的标准，为计算机考试的广泛发展奠定基础。对于国内，华南农业大学自2009年起上线在线评判系统，在专业竞赛网站和商业化在线评判系统功能基础上，扩展了实验教学管理和评价功能[4]。过去三十年，考试系统经历了不断专业化的演进，包括专门针对英语的在线考试（如托福、雅思）以及专注于计算机等级考试。自动化阅卷在上世纪60年代的基础上迎来了新的发展，尤其是一些算法，例如基于K-Shingling的相似度计算。

现如今，在线评测系统已经广泛应用，发展极其迅猛，功能也日渐完善。一个完整的在线评测系统可以胜任题目展示、用户提交、评测、结果展示和排名等一系列工作[5]。各大企业招聘时，同样不仅仅局限于基于简历内容以及简单计算机基础的提问，更倾向于使用在线编程测评软件现场进行测评。在国内外都有非常多的团队对于评测系统这个题目进行研究和完善[6]，比如国内较为出名的牛客网、ACMCoder、北森测评等，国外也有例如Leetcode、HackTheBox等在线测评系统。这些系统经过优秀的团队花费大量时间研究和开发，可以供一些大型的编程比赛的测评使用，也能够作为学生练习自己的编程能力进行评估来使用。

## 论文研究目标与研究内容

### 研究目标

本系统将传统的用例测试与编程题目代码得分点提取方式相结合，使得整个测试过程中既包含了白盒测试的准确性，同时结合了黑盒评分的广泛兼容性。总体希望测试结果更加合理，反馈给老师同学的得分结果更加符合预期。在减少教师手动下载源码评分的复杂性的同时，也降低了评分周期(即从学生提交作业到得到分数的时长)的长度，提高了老师和学生的学习效率。

### 研究内容

通过文献调研的研究形式，了解了在线测评系统的现状。在系统开发前期，对市面上较为常见的测评系统进行详细的体验，例如牛客网、力扣网等大型测评平台，充分了解到国内在线考试系统的现状。根据网络中现有的资料，掌握目前大多数在线考试系统判分思路、实现技术等。同时询问学生以及教职人员，询问对测评系统的基本预期与开发建议，根据询问结果确定了系统开发方向以及主要功能点，为全栈编程题目编程题目编程题目测评系统的开发打下了坚实的基础。

在明确主要需求后，确定了系统总体采用MVC设计模式。该设计模式旨在分离应用程序的内部表示（Model）、用户界面（View）和应用程序的处理逻辑（Controller）。通过将应用程序分为这三个部分，将请求分布到不同的模块中，利用模块之间的互相服务，使得各部分之间的耦合度降低，提高了应用程序的可维护性和可扩展性。

根据需求、架构设计以及如今常用的技术选型，确定了主要使用SpringBoot +Vue + Redis +Mysql的前后端分离技术作为整体的技术框架。使用前后端分离技术，使得代码更易于维护。前端和后端的代码分离清晰，每个部分都专注于自己的功能，不会影响到另一个部分，也更容易理解和调试。同时可以根据需求独立开发和测试，API接口进行集成，有利于今后的部署与维护。

系统后端数据库相关操作主要使用了Mybatis与Mybatis-plus相结合进行使用。作为后端与数据库常用的框架，MyBatis通过XML或注解的方式将Java对象与数据库中的记录进行映射，使得开发过程中，能够使用面向对象的方式操作数据库，避免了传统JDBC中的繁琐的SQL拼接和结果集处理。Mybatis-plus为Mybatis的扩展，封装了丰富的数据库CRUD操作，增强了Mybatis的实用性，减少了大量非规范代码的入侵。

本系统以编程题目评测的合理性为核心，同时需要兼顾系统的完整性和安全性[7]，针对编程题目测评，本文提出了使用“中间文件”的测评方法。该方法弥补了现有系统中只能使用黑盒测评，即使用传统用例测试的局限性，防止被测对象使用特殊处理，强制输出期望答案的情况。在学生上传作业到服务器后，服务器主动执行Shell脚本，根据老师上传到中间文件，提取出学生作业中的相应答案文件，与标准答案进行字符串比对后，实现打分的目的。根据测试结果，有针对性的帮助学生找到薄弱点，加强相关方面的学习。

在系统编写过程中，对于每个完成的模块，及时进行了模块化的单元测试。在系统完成后，对整体进行了集成测试。从稳定性、兼容性、流畅度、性能、功能各个角度，设计测试用例，保证系统的正确使用。

## 论文组织结构

本文的组织结构如下：

第一章：绪论部分。在该部分中，阐述了如今计算机教育的时代背景，确定了在当今复合型社会中，越来越多的学科将直接或间接的使用到编程来提高工作效率，设计并实现自动评分系统的重要性。查找并收集国内外自动评分系统的研究现状，分析出自动评分系统的进展与不足。简单介绍了论文的研究目标以及具体的研究内容，针对目前全栈编程题目测评系统的薄弱点，本文作者通过学习、研究其他测评系统，结合全栈编程题目语言的语言特点，提出了使用”中间文件“的自动评分方法这一创新点。

第二章：相关技术及理论简介。本系统使用了前后端分离式架构，前端使用的技术为Vue，为目前企业中主流的前端框架，同时结合Element-UI手脚架，实现了快速搭建系统，也兼顾了系统界面的美观。后端使用了SpringBoot，同样也是较为成熟以及常见的后端服务开发框架。服务器以及数据库分别使用了Tomcat和Mysql，免费以及高性能使其成为微小项目的首选。

第三章：系统需求分析。本章对全栈编程题目测评系统进行了大体分析，并对系统的设计，实现过程中的技术，部署的人力，设备租赁的经济开销以及社会需求度等可行性进行具体分析，并着重使用用例图分析了功能技术可行性。

第四章：系统设计。本章基于完成系统可行性分析以及需求分析后，进行系统总体架构方面的的规划，根据常见的软件工程设计模式，梳理出系统的层次结构。对于每个层次，合理规划出教师模块、学生模块、评分模块、通用模块的时序图，形象展示了每一个请求背后的处理过程，也梳理清模块之间的调用情况。使用ER图，简述整个系统实体之间的关系。之后分别对每个数据库表进行具体的设计说明。在前面各项工作顺利完成的基础上，实现概念设计与逻辑结构实现。

第五章：系统实现。本章中大量使用源码，对每段关键的业务逻辑代码进行逐条解释，并通过具体例子说明代码如何处理数据，得到正确的返回结果。

第六章：系统测试。首先对系统进行了集成测试，即保证系统能够正常的运行需求。然后对于系统的主要功能，包括登录、增删改查作业、成绩查询等进行功能性测试，列举测试用例以及对应的测试结果。最后，对非功能性进行测试，为系统稳定运行提供保障。

第七章：总结与展望。对整个毕业设计，从选题到设计，再到实现与测试进行总体的回顾，总结工作中出现的问题，如何发现，如何解决。以及遗留的问题，期望在之后的开发过程总能够对系统进行进一步完善与优化。

## 本章小结

本章首先对研究背景进行了阐述，经过前期调研工作以及对市场中常见的系统测试，外加国内外研究现状，总结出了全栈编程题目测评系统的发展与问题。简单说明了论文的研究目标与解决的实际内容，完成系统使用的技术类型，系统的总体设计等相关内容。最后对论文的框架进行了介绍，方便读者了解论文的大体内容。

# 相关理论与技术介绍

## 前端网页技术——Vue

Vue是一款渐进式JavaScript框架，其设计之初的目的为专注于构建用户界面。 Vue的核心库只有几十KB大小，使得页面加载速度快，结合响应式数据绑定机制，大大减少了手动操作DOM的繁琐工作。同时，市面上含有大量基于Vue框架实现的前端模板，如：Element-Ui、Iview-admin，用户可以简单修改模板，完成项目前端开发。在与后端交互方面，Vue提供了如Vue-resource、Fetch、Axios等类库，通过发送请求到后端，将前后端业务进行分隔，提高效率。Vue总的来说，该框架具有运行高效、易学易用、操作简便及支持组件开发等特点，深受研发者的喜爱，是目前主流前端框架[8]。

## 后端框架技术

SpringBoot是一种基于Spring框架的快速开发、简化配置的框架，它使得开发者可以更加快速地搭建和部署Spring应用程序。所谓Spring，是目前较为流行的后端开发框架，通过使用工厂设计模式，使得开发人员不用自己创建繁琐的类，只用调用工厂类即可。然而，Spring其中包含大量的配置文件，需要用户创建XML进行配置，比较繁琐。SpringBoot通过提供自动配置、起步依赖、嵌入式Web服务器等功能，大大简化了Spring应用程序的开发和部署过程。另外，SpringBoot 集成大量框架，使得依赖包的版本冲突以及引用的不稳定性等问题得到很好的解决 [9]。

## 数据库技术

### Mybatis与Mybatis-plus

My Batis是一个开源的Java持久层框架[10]。它通过简化数据库操作的方式，帮助开发者更轻松地与数据库交互。MyBatis通过XML文件或注解配置SQL语句，并提供了一套映射机制，将数据库记录映射成Java对象，使得开发者可以使用面向对象的方式操作数据库。MyBatis-Plus是在MyBatis基础上的增强工具包，它提供了一系列增强功能，简化了MyBatis的使用。简而言之，Mybatis简化了开发人员访问数据库的复杂性，但还需手动编写SQL，而Mybatis-plus自动生成大量SQL语句，开发人家只需按需调用即可。

### 数据库选型

本系统使用MySQL作为数据库进行数据管理。在系统运行过程中，涉及大量的数据变换，MySQL 作为一款免费开源的关系型数据库，它具备体积小、执行速度快、可移植性高以及相对于大型数据库更易于学习使用等特点，逐渐成为中小型信息系统应用的首选数据库[11]。同时支持事务、以及MVCC的特性使得MySQL在保证数据一致性上具有较好的表现。在缓存方面，本系统使用了Redis作为缓存数据库。由于Redis基于内存的特性，使其处理访问时十分高效。利用Redis缓存Token值，是目前较为常见的解决Session无法跨服务器的方式之一。

## 服务器部署

本系统中，需要对学生上传的全栈编程题目Web应用程序归档文件进行自动部署，以及对学生的源代码进行字符串匹配。因此，选择Tomcat服务器作为文件部署服务器。Tomcat是一个免费的开放源代码应用服务器,提供了作为Web服务器的一些实用功能[12]。同时，Tomcat服务器提供了远程管理发API接口，可以通过URL请求对Webapps下的文件进行自定义部署。而后台服务器端选择使用CentOS7，基于Red Hat Enterprise Linux源代码构建而成，提供了稳定可靠的操作系统环境，并使用Shell脚本对学生源代码进行了过滤，提取关键得分点，实现了白盒评分的方式。

## 相关测试理论

白盒测试也称为逻辑驱动测试或结构测试,其工作原理是对软件系统的内部结构进行测试,对软件系统的运行状态进行评估,根据系统不同区域反馈的测试信息得出最终检测结果[13]。可以发现和解决代码中的逻辑错误、边界条件错误等问题，能够提高软件的质量和稳定性，但是需要深入了解代码，测试过程相对复杂，且可能忽略了用户体验和外部环境等因素。

黑盒测试技术又被称之为数据驱动测试技术,其主要是对已知固有产品功能,通过测试验证其功能是否可顺利实施[14]。测试过程相对简单，可以快速进行测试并发现用户体验和功能方面的问题。相应的无法发现代码中的逻辑错误和边界条件错误，只能验证软件是否符合预期的功能和需求。

通常会结合使用白盒与黑盒测试，以便全面地测试软件的质量和稳定性。

## 本章小结

本章主要介绍了系统实现过程中，使用到的技术。从前端的Vue到后端的SpringBoot，再到主要完成测评功能的服务器，简单阐述了了解本系统应有的技术。

# 系统分析

## 全栈编程题目测评系统整体需求

在当前的教学环境中，教师主要依赖手动评分方法来对学生作业进行评估。这种方式存在着效率和准确性方面的挑战，受到各种因素的影响。相比之下，自动评分系统具有简单稳定的特点，可以有效解决一些评分效率方面的问题。但是，现有的自动评分系统通常采用黑盒测试的方法，即仅依赖于与测试用例进行比对来进行评分，无法涵盖所有可能的情况，对于特殊情况缺乏特殊处理的能力。因此，本系统使用了白盒与黑盒测试相结合的方式，使测评过程中兼顾结果与过程，降低了突发情况的错误率，提升了判分的效率，让老师能更投入到教学工作中。同时，系统界面美观且直接，减少了用户的适应时间，吸引同学使用，方便总结与学习。

## 系统可行性分析

### 技术可行性

本系统使用了两种测试方式，即黑盒与白盒测试，下面对两个测试技术进行分析。首先为黑盒测试，需要解决的主要问题为如何将学生上传的WAR包进行自动部署。Tomcat服务器提供了Manager App接口，允许用户使用请求指定WAR包的位置以及部署的相对路径。充分该接口即可完成WAR包部署并成功访问。其次为白盒测试，涉及到对学生源代码进行审核。因为学生会将压缩包上传到Linux服务器，因此，使用Linux中的Shell脚本即可实现对压缩包的批处理工作。SpringBoot与Vue.js结合创建的在线考试系统是一个高效、灵活的解决方案，完全可以满足本系统的需求[2]。同样Redis+MySQL的数据处理方式也是当今最常见的解决方案，保证了数据的安全性与系统的易用性。本系统具有易操作、易管理、交互性好的优点[15]。

### 经济可行性

经济可行性实质就是设计该方案需要投入的资金数目与其带来的效益是否平衡[16]。本系统所使用的前后端技术框架、数据库、服务器等资源皆是完全开源免费，可以通过官方网站进行下载。而在系统部署运行时，前端利用Node.js中的Npm进行包管理，实现一键安装部署，后端既可以使用本机与虚拟机组合部署，也可以部署到各大厂商提供的云服务器中，可以适用不同需求，具备经济可行性。

### 社会可行性

本系统意在创建一个简单高效的编程测评系统，减少老师的工作压力以及调动学生的学习热情，提升了学生的学习效率，不存在对学生以及老师的人身、精神方面造成威胁的因素。所以该系统具备社会可行性。

## 系统功能性需求分析

本段对系统中两种主要使用人群的功能需求进行了分析，分别为学生模块功能分析，包括对作业的上传修改；老师模块分析，对作业进行白盒黑盒测试，对作业的增删改查，查看作业完成情况。

### 学生模块功能需求分析

在学生进入系统首页后，使用学号与密码进行登录，在登录之后，后端返回时长为30分钟的Token值，该值保证学生在30分钟内无需再次登录，节省时间。登录页面后主要为题目呈现与作答模块。如图3-1所示：

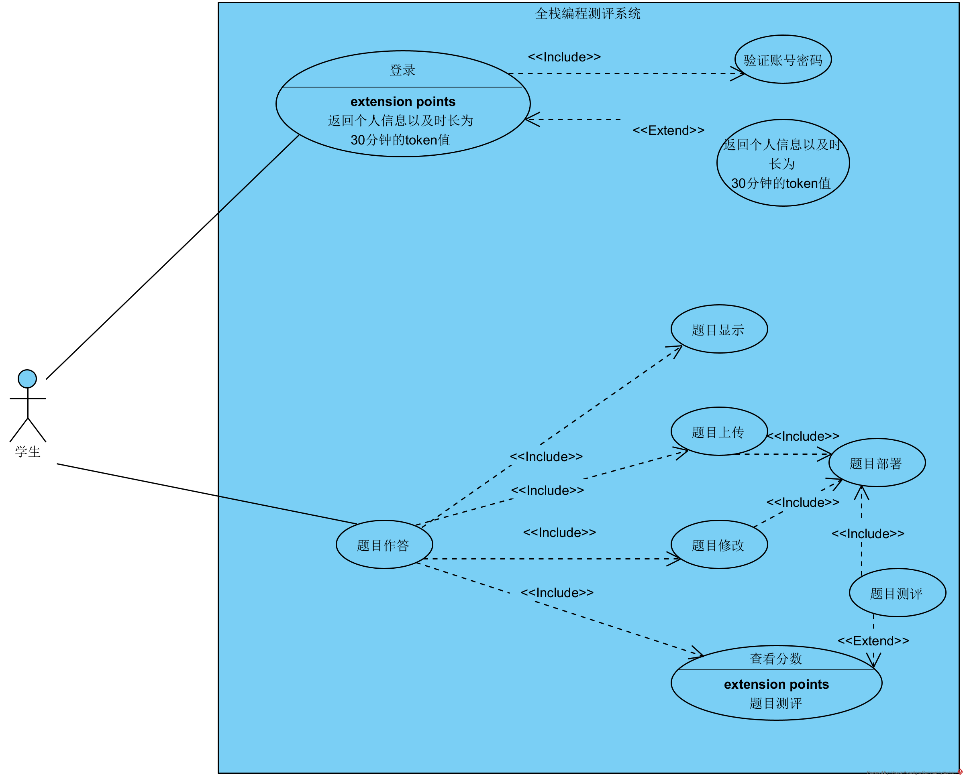


图3-1 学生用例图

在学生登录系统后，系统查询数据库，返回作业列表并显示学生完成情况。学生根据题目要求，提交相应文件。待老师端对作业进行测评后，学生可以得知作答结果并根据成绩规划复习重点。

### 教师功能模块需求分析

教师模块中，同样包含了登录模块，减少重复登录的不必要。在教师模块中，还包含了题目测评、题目检查、题目管理这三个模块，如图3-2所示：



图3-2 教师用例图

题目管理模块：教师角色在进入系统后，可以对系统中题目进行编辑，新增题目，修改题目，删除错误题目等。

题目检查模块：在题目检查模块中，教师可以查询每个题目的完成情况、为完成人员以及每个人的分数情况，及时督促学生完成作业。

题目测评模块：教师可以使用该模块提供的两种测评方法，动态测评与静态测评，对学生所完成的作业进行全方面的评分，评分结果将显示在题目检查模块中。

## 非功能需求分析

### 可扩展性

在系统设计中，采用了分层设计的思想，各个模块之间通过请求服务的形势进行数据传递，在今后的运行过程中，如果需要对系统增加功能点，只需新增接口或服务即可，保证了系统 对扩展开放,对修改关闭。

### 规范性

系统中的变量、方法、以及类名都遵循了开发规范，同时对核心代码进行了注释，方便其他开发人员在后续调试过程中了解系统各模块的实现逻辑。

## 本章小结

本章首先对全栈编程题目测评系统进行了总体分析，描述了系统的主要功能点，之后对从技术、经济、社会等方面，分析了系统实现的难度与价值。在之后，对测评系统中的每个模块做了阐释与说明，保证在实际编码之前，理清系统思路和实现过程。

# 系统设计

## 系统架构设计

根据上述需求分析，系统整体的架构思想如图4-1所示

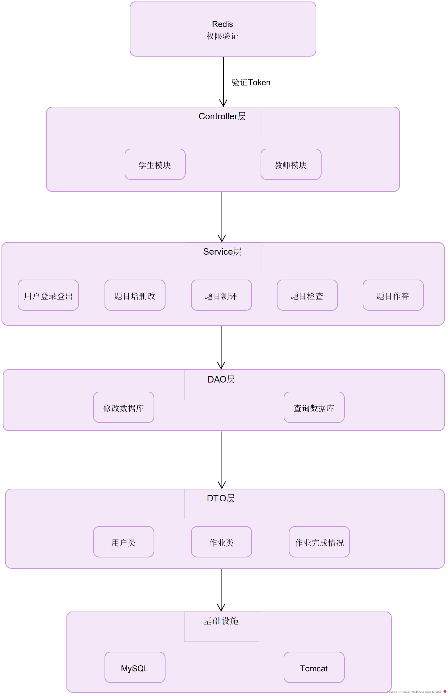


图4-1 系统架构图

## 系统模块设计

在该系统中，主要设计为两个模块，分别为学生与教师模块。在学生模块中，分为登录模块、个人信息模块、作答题目模块、题目呈现模块。教师模块中，分为教师登录模块、题目管理模块、题目白盒测评模块、题目黑盒测评模块、个人信息模块以及题目查看模块。如图4-2所示。

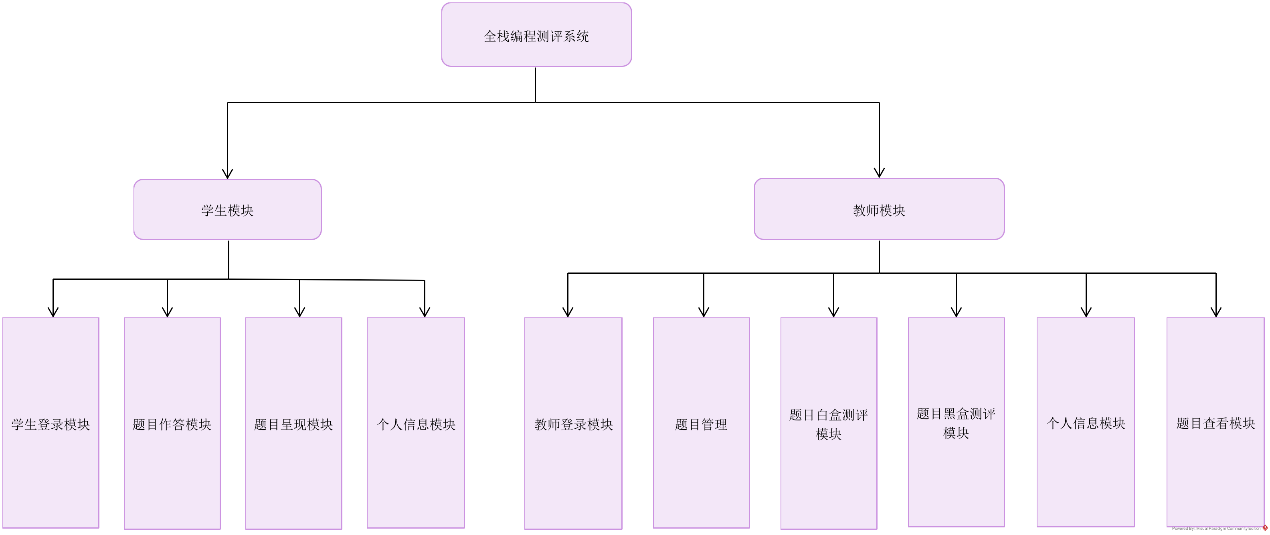


图4-2 系统模块图

## 系统数据库设计模块

### 系统数据库总体概述

系统整体ER图如图4-3所示，其中每个学生对应多个作业完成情况，每个作业对应了多个作业完成情况，通过作业完成情况表，将学生与作业联系起来。老师可以对作业进行增删改查工作，同样可以对作业完成情况进行审批。

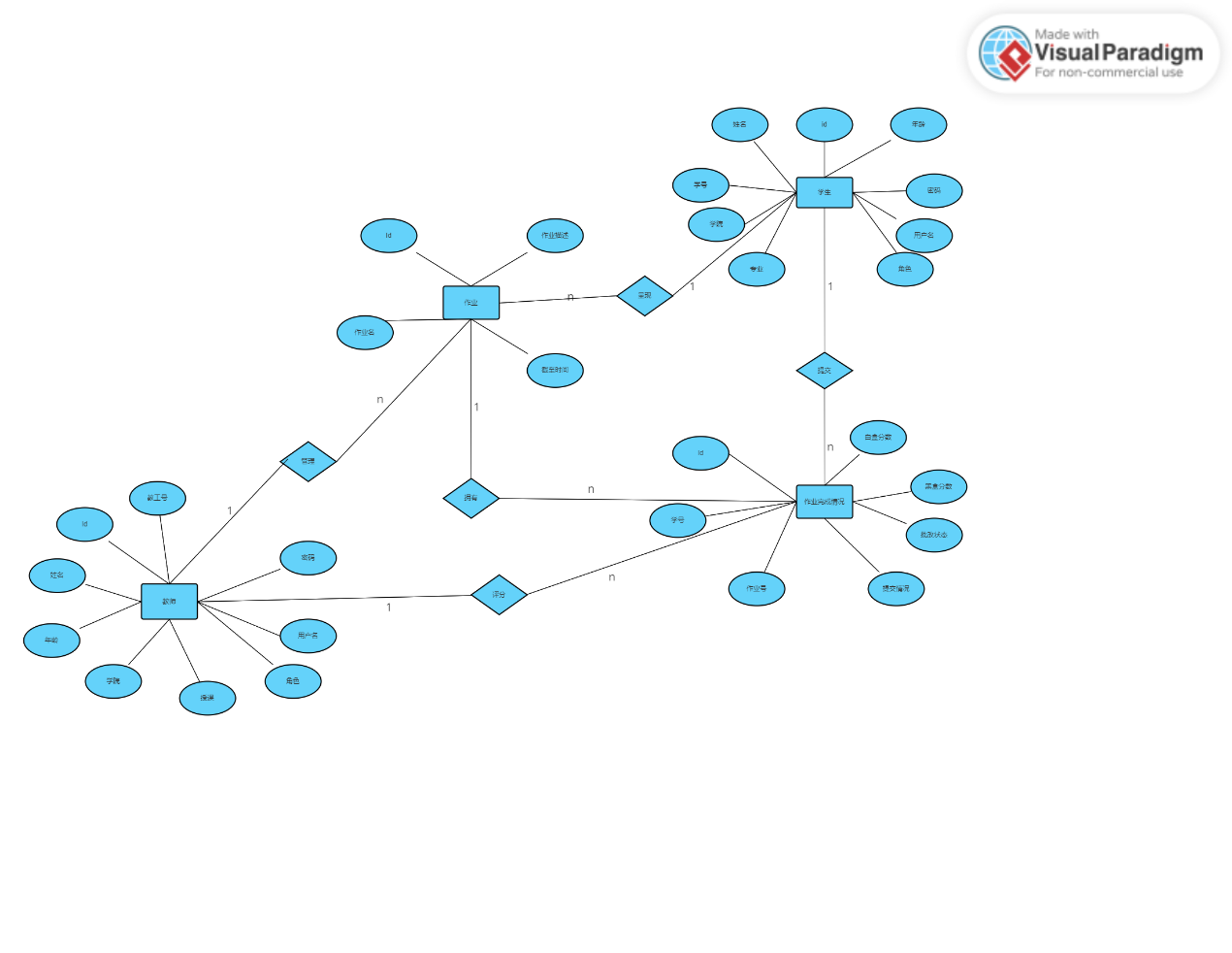


图4-3 系统ER图

### 各表详细内容说明

学生信息表，该表用来存储学生的基本信息，包括学生的学号、姓名、用户名以及密码等信息。通过查询该表，可以得到学生的个人信息。

表4-1 学生信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 解释说明 | 类型 | 允许空值 | 备注 |
| Id | 学生Id | BigInteger | False | 主键 |
| name | 学生姓名 | varchar | False |  |
| student\_Id | 学生学号 | varchar | False |  |
| Age | 学生年龄 | Integer | False |  |
| college | 学生学院 | varchar | False |  |
| major | 学生专业 | varchar | False |  |
| role | 学生角色 | varchar | False |  |
| username | 用户名 | varchar | False |  |
| password | 密码 | varchar | False |  |

教师信息表，该表用来存储教师基本信息，包括教师姓名、学院、授课情况、用户名密码等。

表4-2 教师信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 解释说明 | 类型 | 允许空值 | 备注 |
| Id | 教师Id | BigInteger | False | 主键 |
| name | 教师姓名 | varchar | False |  |
| teacher\_Id | 教工号 | varchar | False |  |
| age | 教师年龄 | Integer | False |  |
| college | 教师学院 | varchar | False |  |
| course | 教课 | varchar | False |  |
| role | 教师角色 | varchar | False |  |
| username | 用户名 | varchar | False |  |
| password | 密码 | varchar | False |  |

作业信息表中，记录了每个题目的信息，包含作业名，作业描述、作业截至时间和作业编号。

表4-3 作业信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 解释说明 | 类型 | 允许空值 | 备注 |
| Id | 记录Id | BigInteger | False | 主键 |
| assignment\_Id | 作业编号 | BigInteger | False |  |
| submission\_status | 提交状态 | varchar | False |  |
| grading\_status | 评分状态 | varchar | False |  |
| black\_box\_score | 黑盒得分 | BigInteger | False |  |
| white\_box\_score | 白盒得分 | BigInteger | False |  |

作业记录表主要用于记录学生作业完成情况，具体包括了记录ID，该记录对应的作业编号、提交状态、评分状态、黑盒测评的得分以及白盒测评的得分。

表4-4 作业记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 解释说明 | 类型 | 允许空值 | 备注 |
| Id | 记录Id | BigInteger | False | 主键 |
| assignment\_Id | 作业编号 | BigInteger | False |  |
| submission\_status | 提交状态 | varchar | False |  |
| grading\_status | 评分状态 | varchar | False |  |
| black\_box\_score | 黑盒得分 | BigInteger | False |  |
| white\_box\_score | 白盒得分 | BigInteger | False |  |

## 系统详细设计

本段将对登录流程、提交作业、白盒测试、黑盒测试、作业增加这五个系统关键功能进行详细说明

### 登录流程详细设计

在登录功能中，用户进入主页面时，如果请求中没有携带的Token值，则会强制进入登录页面进行登录。点击登录按钮，输入用户和密码后，前端页面会发送请求到后端数据库，与数据库数据进行比对验证，如果成功，则会生成一个随机的Token值，将Token值以及用户的个人信息放入redis缓存中，在接下来的访问中，系统会根据用户的Token值以及是否过期来提供服务。时序图如图4-4所示

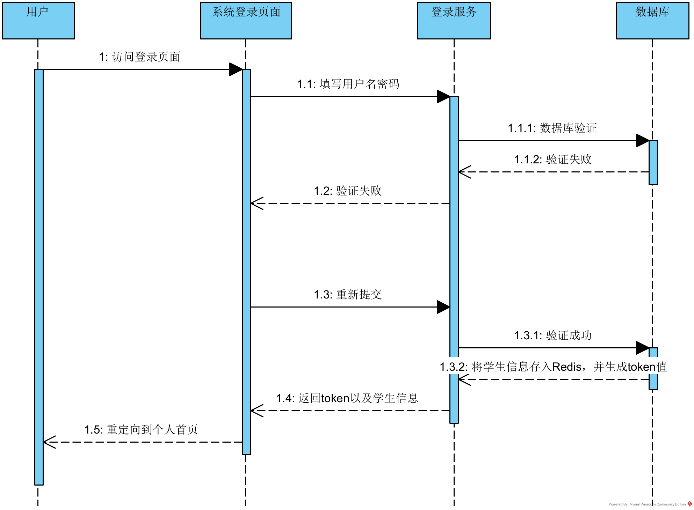


图4-4 登录时序图

### 学生作答功能详细设计

学生点击作业查看后，首先会发送请求到后端，从数据库的作业记录表中利用学生的ID值，查到学生作业完成情况并返回到前端。点击具体题目的提交按钮后，可以进行代码提交。提交白盒代码时，在上传到远端服务器之后，会触发Tomcat的自动部署功能，对新上传的WAR包进行部署后，返回部署情况。提交黑盒代码时，上端zip文件到远程服务器后，返回提交情况，时序图如图4-5所示

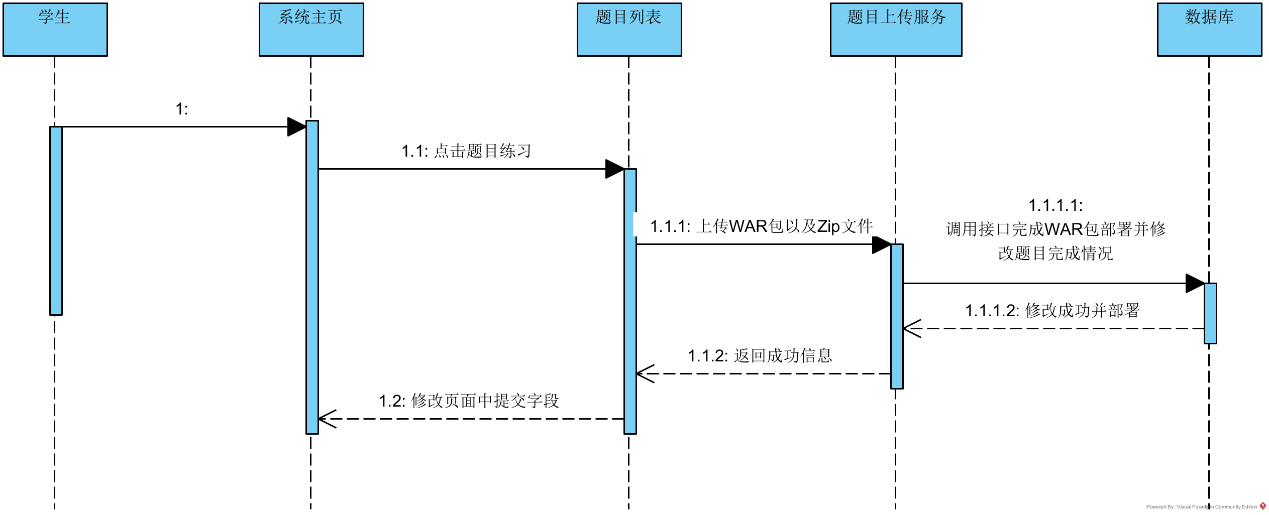


图4-5 作业提交时序图

### 作业审批功能详细设计

在学生提交代码后，教师端利用作业ID，可以在作业检查页面对作业进行检查，查看每个作业的完成情况以及得分情况。当老师使用黑盒测试时，系统会要求老师填写需要发送的请求的类型、请求参数、期望结果，并在后端组装请求路径，发送到学生提交作业时已经部署好的项目中，得到返回值后，后端会将返回值与期望结果比较，进行判分。当老师使用白盒测试时，系统会调用服务器中写好的Shell脚本，该脚本可以解析老师上传的评分文件，取得具体得分点，解压服务器中的zip文件并与评分文件的得分点进行比较判分。最后，将判分结果存入数据库中。时序图如图4-6与图4-7所示

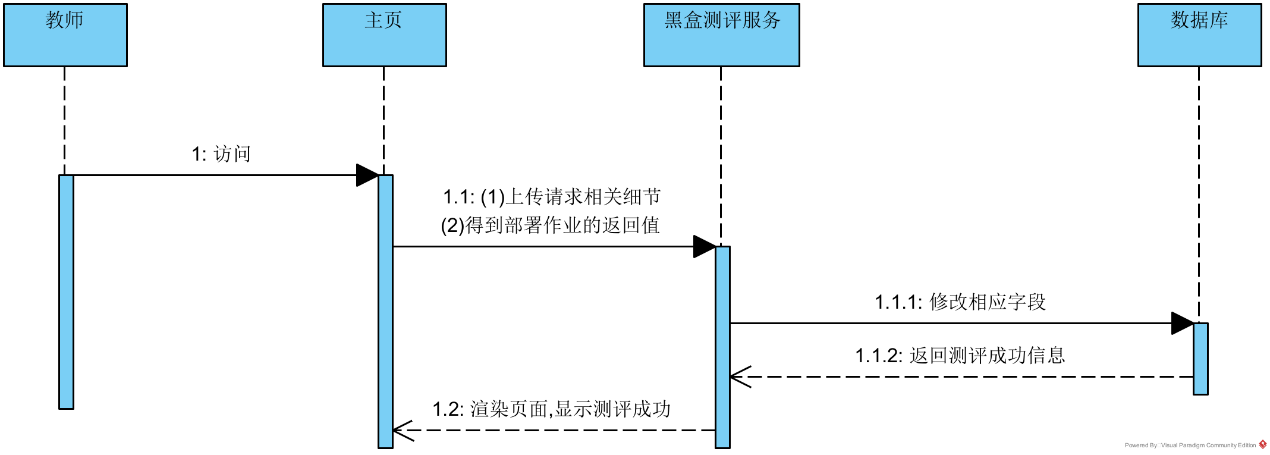


图4-6 黑盒测试时序图

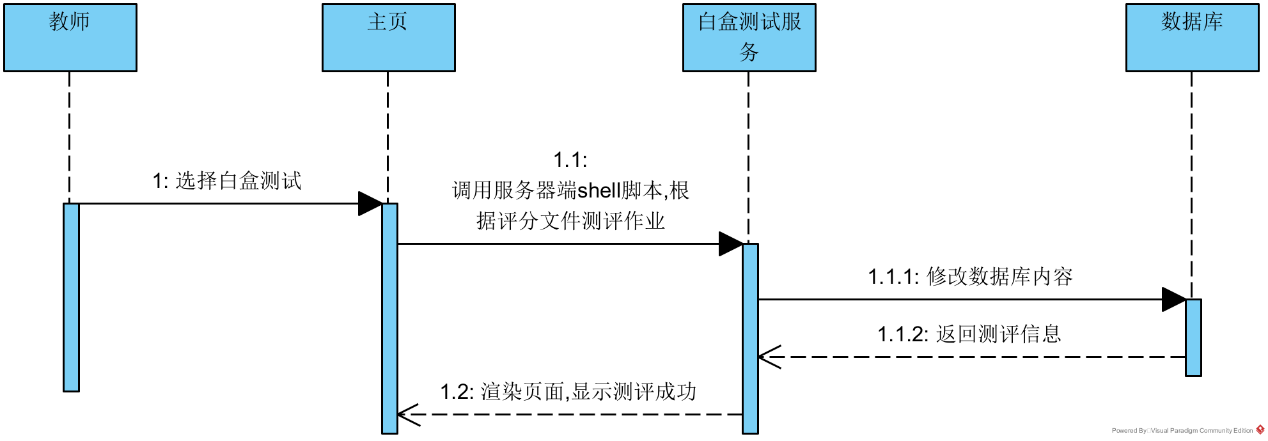


图4-7 白盒测试时序图

### 老师编辑题目详细设计

老师可以选择对某个题目进行修改、删除，在选择修改或删除后，会得到该题目的Id值，使用该Id值，对数据库中的作业表，作业记录表进行修改、删除。在老师增加题目时，需要定义题目名、题目描述、截至时间以及上传评分文件。在点击提交后，会发送请求到后端中，后端会将评分文件上传到服务器中并新增作业表、作业记录表。时序图如图4-8所示

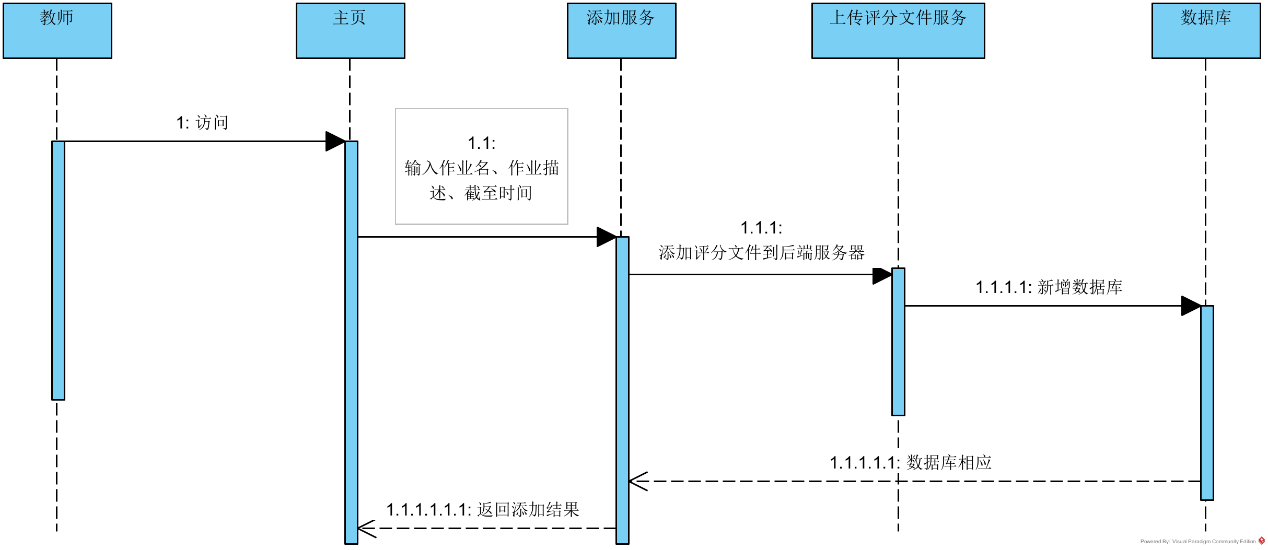


图4-8 教师增加作业时序图

## 本章小结

本章对全栈编程题目测评系统进行了总体加细节的设计，确定了系统使用的架构层次以及系统中各个模块设计，之后采用ER图说明了系统中实体类的关系与组成。随后对系统中的关键模块使用时序图进行了解释说明，从多个角度阐述了系统的设计思路。

# 系统实现

本章中，将对全栈编程题目测评系统划分为三个模块进行实现讲解，分别为登录模块、学生模块以及教师模块，其中，学生模块包括题目上传、学生信息展示，老师模块中包含题目管理、题目测评、成绩查看这三个模块。

## 登录模块

当用户登录项目地址时，首先会访问到系统的登录首页，从该首页中进行登录验证。在前端输入用户名密码后，会访问后端的验证接口，调取数据库中对应信息进行验证。如果验证成长，则会生成Token值并将学生信息放入Token中。在之后，每次访问页面时就会使用Token进行验证登录时效性。如图5-1为前端，后端验证登录如图5-2所示



图5-1 前端页面代码

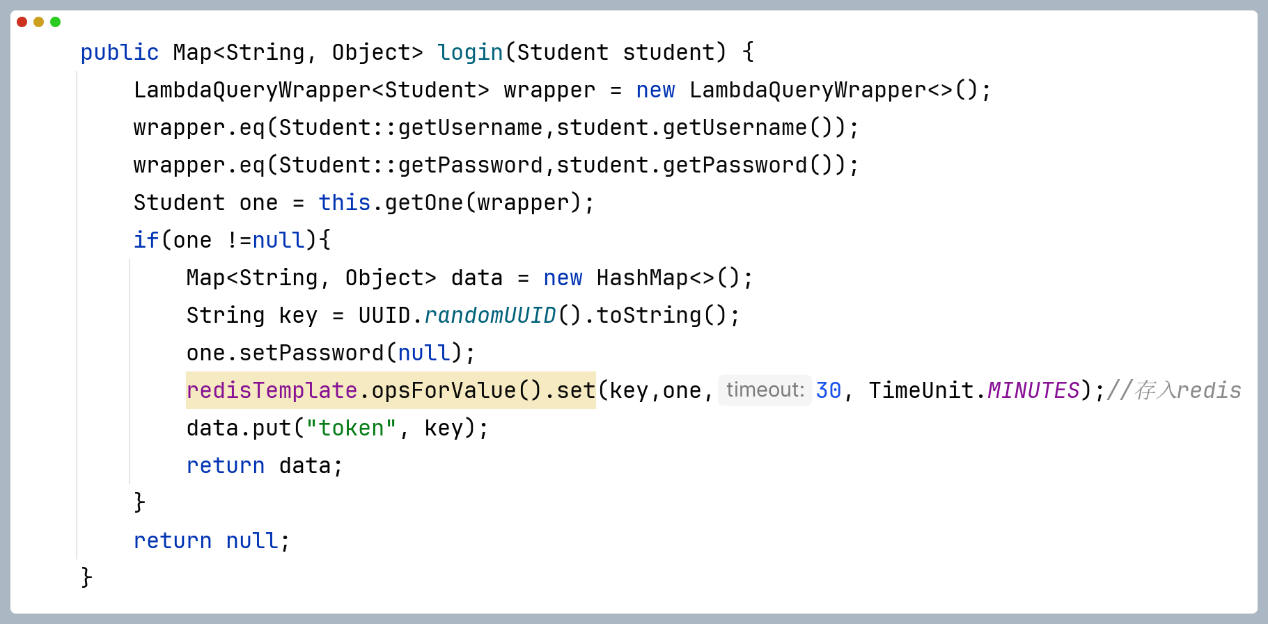


图5-2 后端验证学生信息



图5-3 登录页面展示

图5-3主要展示了在学生登录时的页面效果，分别由两个输入框接受用户名与密码。点击登录按钮之后，会发送请求到后端系统中，进行下一步的验证。

在登出时，学生点击登出后，会发送请求到后端，后端会删除Redis中的Token，再次访问时，会在Redis中找不到Token值并跳转到首页。详细代码如图5-4、图5-4所示

图5-4 前端登出时逻辑代码

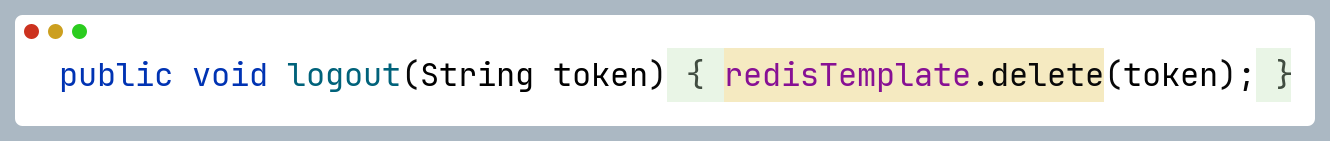


图5-5 后端登出时逻辑代码

## 学生上传作业实现

本模块主要内容为学生点击上传按钮后，需要提交作业的war包以及zip文件。上传后，服务器端部署的接口会接受war包并调用Tomcat的接口进行部署，同时会将zip文件上传到对应作业的文件夹中，等待老师进行测评。页面总览如图5-6所示



图5-6 学生上传作业页面

在学生上传作业后，后端代码得到上传的文件，判断文件名后，如果包含.war字段，则调用Tomcat的部署接口，同时需要在发送的请求头中，添加Authorization验证，通过Tomcat的验证。判断逻辑代码如图5-7所示

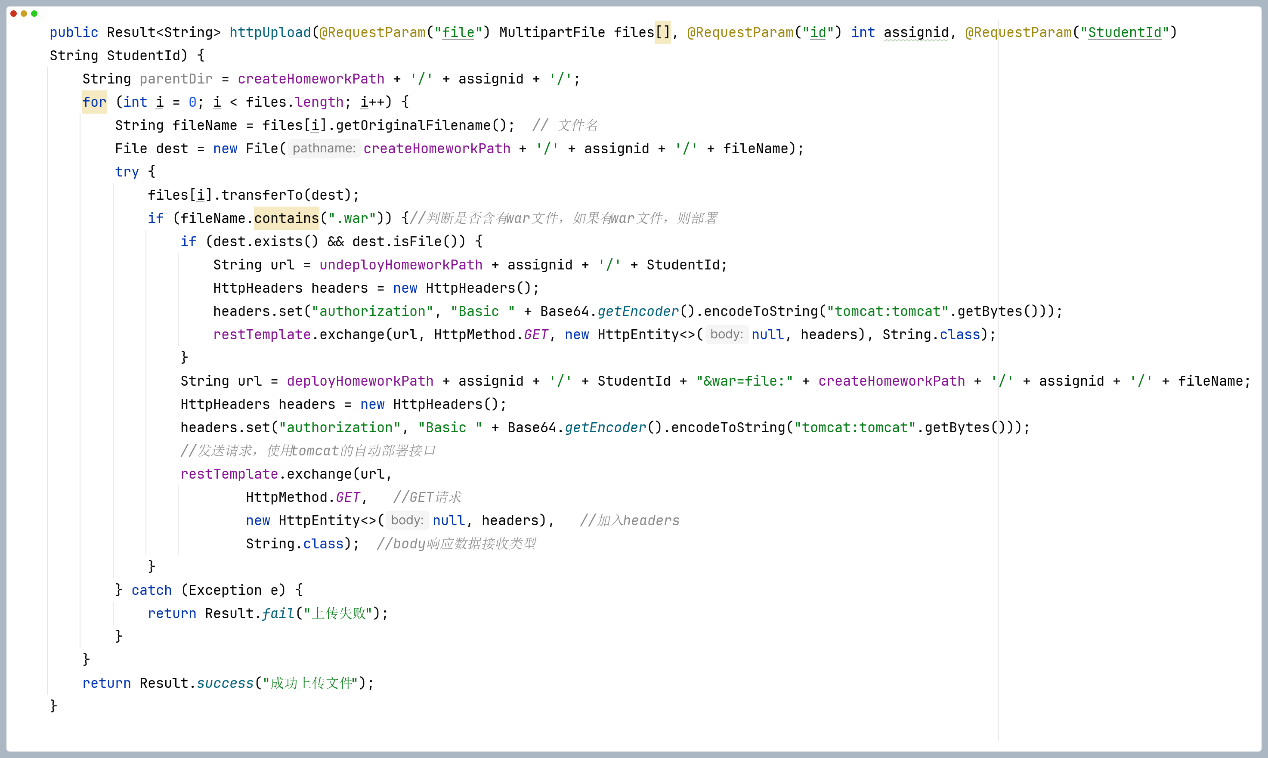


图5-7 上传并部署学生作业代码

通过前端与后端是交互以及互相制约，共同完成了学生上传作业的功能。

## 教师模块

在教师登录后，会显示教师相关的功能，其中包括作业管理，作业删除，作业判分以及作业检查。本段会对上述主要模块进行详细分析，包括代码层面的解释，如何与服务器端进行协作以及Shell脚本的具体实现过程。

### 新增作业模块

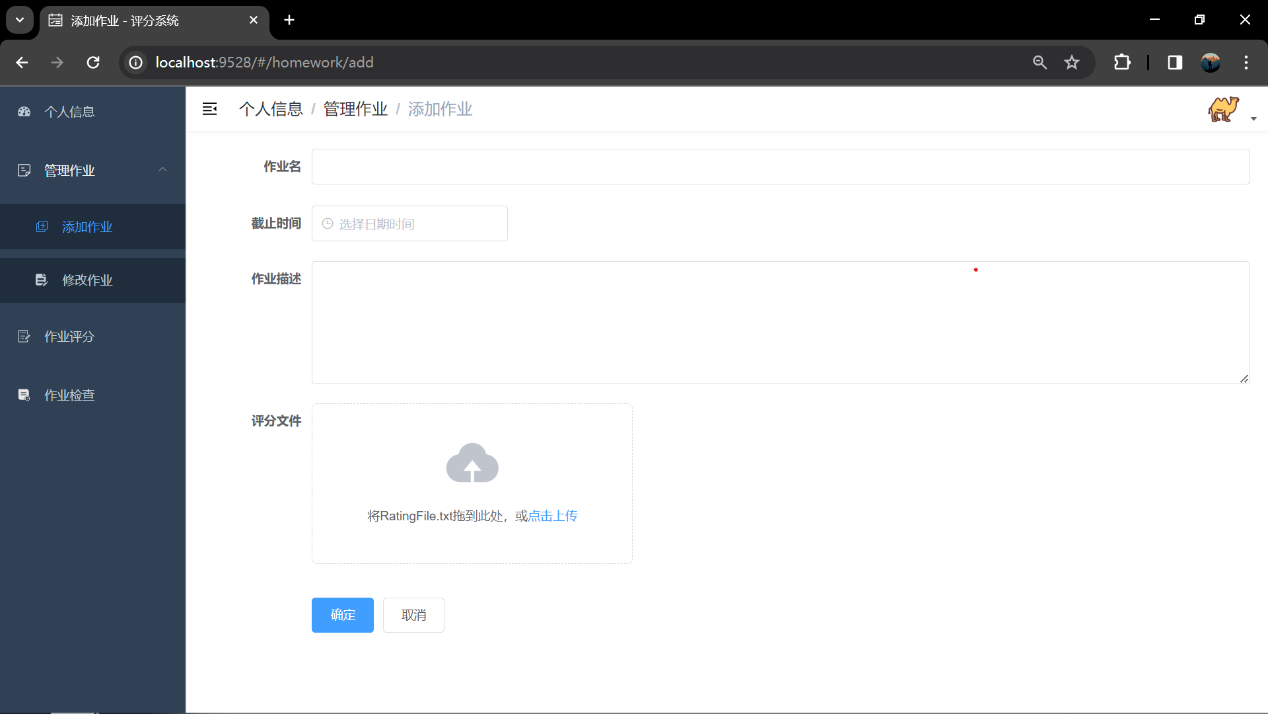
当教师点击新增作业模块后，会提示教师输入作业名、作业描述、截止时间以及需要上传评分文件，教师需要按要求填写并上传所需内容。如图5-8所示

图5-8 教师新增作业页面

当教师填写完毕并点击确定后，会调用前端页面的接口将填写完从表单发送到后端，后端接受后，会访问服务器端接口，在服务器中新增对应作业ID的文件夹，如果该操作成功，那么后端会再次操作数据库中的作业记录表，为每个学生新增作业，并将此次操作的结果封装返回。

图5-9中的代码展示了前端新增作业接口的逻辑代码，其中，data为表单中的数据值，URL为后端接口，使用Post将表单发送到后端中，进行新增过程



图5-9 新增作业前端代码

图5-10中的代码为后端新增代码，首先调用Mybatis-plus封装好的接口，接受到的前端数据封装为作业对象并插入数据库中并返回新增的作业ID值，拿到ID后，访问服务器接口，新建作业ID的文件夹等待上传作业。后端还需使用作业ID为每个学生创建对应的作业



图5-10 后端新增代码

在服务器处理新增作业的流程中，拿到后端发送的ID后，会进入对应的文件夹目录，并创建以作业ID命名的文件夹，并将黑盒测评脚本拷贝到该目录中，等待学生提交作业，具体代码如图5-11所示



图5-11 服务器新增作业处理逻辑

### 修改删除作业模块

当教师点击修改作业模块后，页面显示出所有的作业的详细信息，还提供了修改与删除按钮，教师可以对作业进行修改与删除。页面如图5-12所示



当点击修改按钮后，会弹出对话框，内容包括作业的ID值、作业描述、作业截至时间以及修改中间文件。教师可以对相应的作业内容进行自定义的修改。如图5-13中的代码所示，首先使用了modifyHomework接口，将修改作业的表单进行提交并发送到后端接口中，如果需要修改评分文件，则还需调用下面的submitUpload接口上传评分文件到服务器接口，服务器接口找到对应的作业ID文件夹并重新修改评分文件



图5-12 修改接口

当教师点击删除按钮后，会获取当前的题目ID并请求后端接口。后端接口会根据作业ID值，进入将数据库中作业表、作业记录表中匹配的ID字段进行删除，并访问服务器端。服务器端拿到ID值后，会将webapps下的homework中的ID文件夹进行删除，同时调用Tomcat的undeploy接口，将ID路径下的所有部署的项目全部取消部署，通过上述操作，删除一个作业。具体代码如图5-13，图5-14，图5-15所示



图5-13 前端处理删除逻辑代码



图5-14 后端处理删除逻辑代码



图5-15 服务器端处理逻辑代码

### 教师查看作业情况模块

当教师点击作业检查按钮后，系统跳转到作业检查页面，该页面如下图所示，其中包括了作业名、作业完成情况、查看分数以及未完成人员名单等信息。教师在点击查看成绩按钮后，会获得学生对应作业的白盒评分、黑盒评分分数。具体图示如图5-16所示

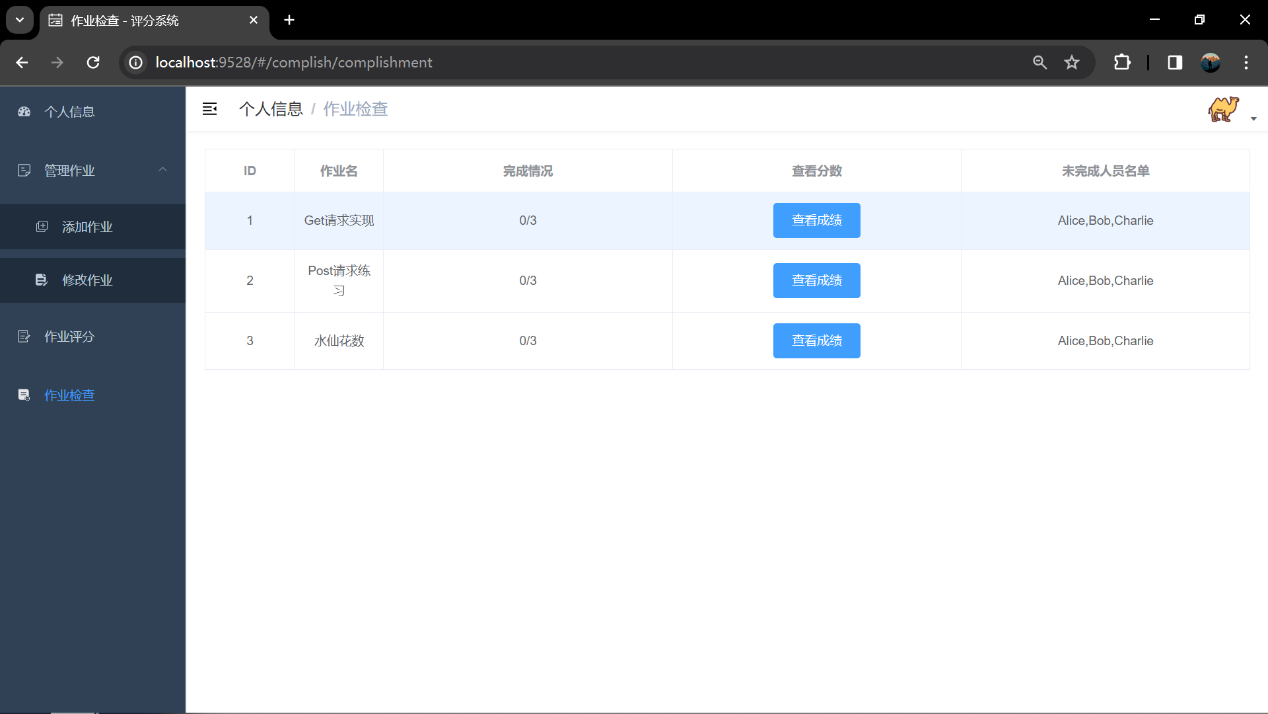


图5-16 教师查看作业情况页面

## 自动化测评模块

本系统主要使用了白盒与黑盒两种测评方式进行测评，教师可以自主选择使用那种测评方式，最后的测评成绩也会展示白盒与黑盒分数

### 黑盒测评模块

黑盒测评中，需要教师确定发送的请求类型、是否携带请求参数、以及最后期望的返回值。具体内容呈现为一个表单，如图5-17所示

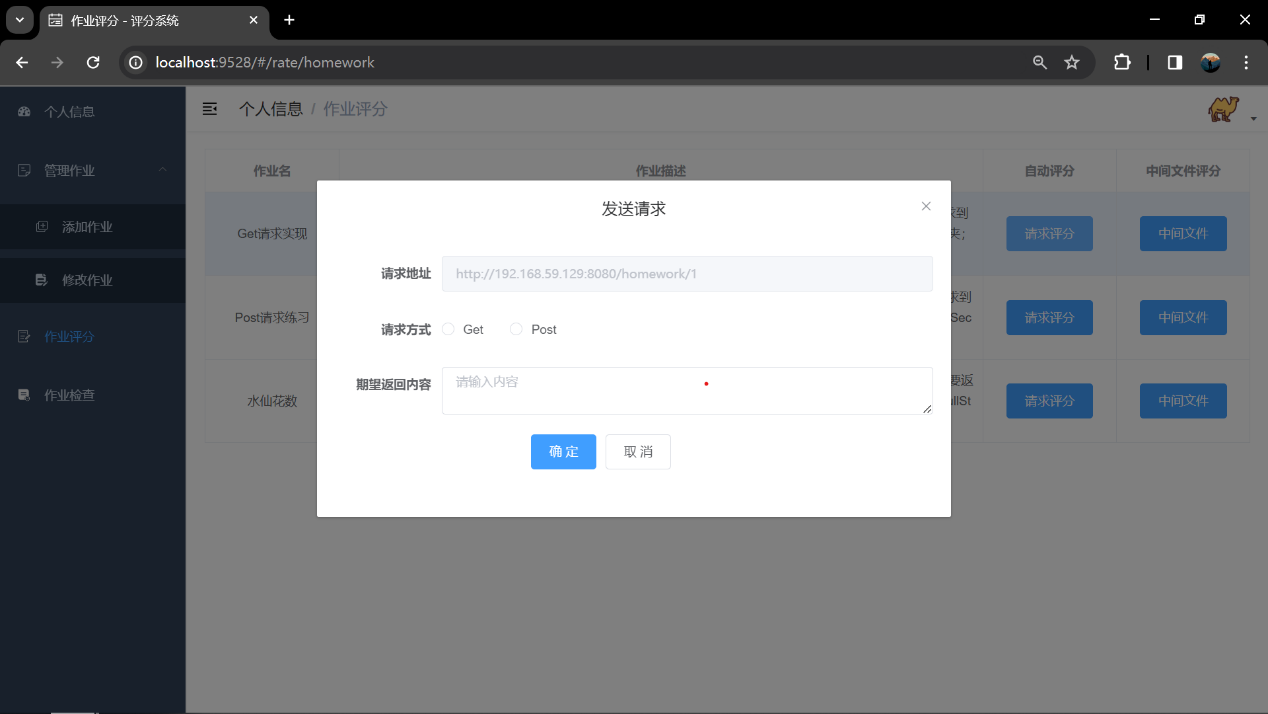


图5-17 黑盒测试发送页面

当教师选择了Get请求后，表单会提示教师输入请求头信息，该信息教师可以不用填写，如果不填写，则进入无参数代码模块。如果填写请求头信息，则在后端会对URL进行拼接，具体代码如图5-18所示



图5-18 黑盒测试Get请求代码

上述代码中，URL参数为具体要访问路径的根地址，即不包括请求头的地址。在下面的for循环中，messages为教师填写的请求头信息，通过for循环进行遍历，得到每一个请求头参数，并使用UriComponentBuilder的queryParam进行拼装。在得到所有的请求参数后，将URL转换为字符串并调用callGetApi函数，该函数用于发送Get请求到对用的URL并得到返回值，将返回值与期望的返回值进行比对，即可得到学生成绩

当教师选择Post请求时，同样表单会提示教师输入请求体信息，该信息教师可以不用填写，如果不填写，则进入无参数代码模块。如果填写请求体信息，则在后端会将请求体封装为Json并发送，具体代码如图5-19所示

URL参数为具体要访问路径的根地址.在下面的for循环中，messages为教师填写的请求体信息，通过for循环进行变量，得到每一个请求体参数，将每个请求体参数放入一个JSONObject中，并调用callPostApi进行发送。得到了返回结果后与期望结果进行比较得出最后成绩，并使用studenthomeworkMapper接口完成对数据库的更新。

对于无参数的请求，后端不再需要对请求进行修改，只需对调用callGetApi或callPostApi，并对最后的返回值进行比较即可完成相关功能。

### 白盒测评模块

白盒测试中，主要通过Shell脚本将对应文件夹目录下的文件进行文本比较。当教师选择白盒测试时，前端页面会访问后端接口，后端接口将访问服务器端，通知服务器端使用Shell脚本进行判分，页面如图5-20所示

在前端发送请求后，会到达服务器端，启动服务器端的Shell脚本进行基本测评，下面开始讲解Shell脚本相关内容。

该段代码主要实现内容为，在获取到前端页面送来的需要测评的作业ID值之后，通过cd命令进入homework下的文件夹中，因为学生上传的为zip文件夹，所以需要对zip文件夹进行解压缩工作，对应代码为unzip -qo "$zip\_file" 。解压后，将文件名保存到数组中，使用for循环对每个学生文件夹进行遍历。

该段代码主要用于将文件夹中的评分文件使用awk指令进行拆分，得到评分文件中的首个信息，即学生作业中必须存在的文件夹路径。

上述代码为白盒测试的核心执行代码，主要由三个for循环组成。第一个for循环用于遍历刚刚解压的所有学生作业文件夹，在访问该学生作业文件夹后，进入第二个for循环。第二个for循环主要用于访问得到的作业中必须存在的文件夹路径，同时使用awk命令，读取出评分文件中的文件名与文件中的内容信息。之后，进入第一个if判断语句。如果对应的文件夹路径可以访问，进入第二个if判断语句，如果不存在该文件夹路径，则终结本次for循环并记录学生成绩。第二个if判断语句中，判定了文件夹路径下是否有包含了对应的文件，如果文件存在，进入下一步，如果不存在，退出for循环并显示学生成绩。在第二个if判断语句之后，会对评分文件中的作业中必须包含的内容进行再次拆分，应对多个给分点的情况。随后，会使用grep命令，将给分点与作业文件进行过滤比较，如果缺少的内容，则得到60分，如果内容对应完全正确，则得到100分，最终将记录学生成绩。

自此，关于系统主要实现的白盒与黑盒测评功能分析结束。

## 本章小结

本章中，首先以学生角色作为切入点，讲述了在学生视角下，在访问了登录页面后，系统如何进行响应，前后端代码如何进行交互以及提交作业的方式与服务器的变换。在介绍完学生相关内容后，转换为教师的角度，讲解了教师如何对作业进行增删改查，如何对作业进行评分，以及最重要的是系统核心测评代码的讲解。通过所有模块之间的合成与协作，最终完成了全栈编程题目自动化测评系统的所有功能。

# 系统测试

系统测试是系统开发的一个重要环节，是验证所设计的系统是否满足功能要求和性能要求的重要手段[17]。在实际生产环境中，对系统的测试主要分为白盒与黑盒测试，其中，白盒测试更偏向于源代码层面，是一种适合程序员的测试方法，与之相对，黑盒测试更关注系统的实用性与易用性，对源代码层面不太关注，更符合用户视角。因此本次测试将采用黑盒测试。通过黑盒测试中的正反用例的方法，编写有效等价类、无效等价类，对比两者之间的结果区别是否符合预期，检测系统功能是否正常运转。本次系统测试的目的在于，在系统正式上线之前，可以及时发现在编写代码中无法察觉到的致命与非致命错误，将测试结果及时反馈于研发人员进行修改，保证系统在所有的回归测试通过后，提供给用户最完美的使用体验

## 登录/登出模块测试

作为系统中最简单也是最重要的模块，登录界面是否能过正常运转对之后的一系列工作有着至关重要的意义。如果登录页面出现了问题，极大可能会影响系统的稳定运行、用户的初体验，甚至是系统数据的安全。因此作为系统的第一关，登录页面测试不容小觑。

表6-1 登录模块测试结果

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用例名称 | 用例所属模块 | 测试步骤 | 期望结果 | 实际结果 | 测试结果 |
| 用户正常登录 | 登录模块 | 用户访问登录页面，输入正确的用户名密码 | 成功跳转相关页面 | 跳转成功且缓存中多出Token值 | 测试通过 |
| 密码有误 | 登录模块 | 用户访问登录页面，输入错误的密码 | 提示密码错误 | 密码错误，未生成Token值 | 测试通过 |
| 用户名有误 | 登录模块 | 用户访问登录页面，输入错误的用户名 | 提示用户名错误 | 用户名错误，未生成Token值 | 测试通过 |
| 密码为空 | 登录模块 | 用户访问登录页面，未输入密码 | 提示输入密码 | 未输入密码，无法生成Token值 | 测试通过 |
| 用户名为空 | 登录模块 | 用户访问登录页面，未输入密码用户名 | 提示输入用户名 | 未输入用户名，无法生成Token值 | 测试通过 |

## 学生模块测试

在学生模块中，需要保证学生在上传作业的顺利进行，检测作业文件的合法性，及时提示作业提交状态，反馈学生作业得分情况，保证学生能过正常完成作业并取得合理的分数。

表6-2 学生模块测试结果

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用例名称 | 用例所属模块 | 测试步骤 | 期望结果 | 实际结果 | 测试结果 |
| 学生上传正确文件 | 学生模块 | 学生按老师要求，提交WAR包以及zip文件 | 提示成功提交作业 | 提示成功提交作业 | 测试通过 |
| 学生上传错误文件 | 学生模块 | 学生未按老师要求，提交WAR包以及zip文件 | 提示系统只能上传相应的文件 | 提示文件有误 | 测试通过 |
| 上传过程中后台有误 | 学生模块 | 在未启动服务器情况下上传文件 | 提示系统有误 | 提示系统有误 | 测试通过 |
| 提交状态正常显示 | 学生模块 | 在学生成功提交后，系统显示成功提交 | 显示提交状态为成功 | 显示提交成功 | 测试通过 |

## 教师模块

教师模块中主要要对作业进行增删改查，至于分数的自动化测评方面，由于各个题目之间没有关联性，因此对该模块进行测试时，需要结合真正的上线环境，需要大量的不同题型与结果进行支撑。本文中不对该内容进行测评。

表6-3 教师模块测试结果

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用例名称 | 用例所属模块 | 测试步骤 | 期望结果 | 实际结果 | 测试结果 |
| 新增题目并上传评分文件 | 教师模块 | 点击新增题目模块并上传评分文件 | 成功新增作业 | 成功新增作业，数据库新增数据，服务器新增文件夹 | 测试通过 |
| 新增题目未上传评分文件 | 教师模块 | 点击新增题目模块但没有上传评分文件 | 提示需要上传评分文件 | 未新增作业 | 测试通过 |
| 修改题目 | 教师模块 | 点击修改题目 | 成功修改题目 | 数据库内容修改，页面内容修改 | 测试通过 |
| 删除题目 | 教师模块 | 点击删除题目 | 成功删除题目 | 数据库内容删除，服务器文件夹删除 | 测试通过 |
| 查询题目完成情况 | 教师模块 | 点击查看学生题目模块 | 正确显示学生完成情况与得分情况 | 成功查询数据库 | 测试通过 |

## 本章小结

在本章中，主要对系统进行了黑盒测试，包括测试了系统登录模块、学生模块、教师模块这三个主要系统功能模块。通过测试，保证了功能的完备性，检测了系统能否实现系统设计与开发的目标。测试结果显示，系统测试结果与预期完全一致，说明本系统具有良好的稳定性与安全性。

# 总结与期望

## 总结

随着社会经济发展水平的提升,对计算机的应用已经在不知不觉的过程中渗透到各个社会行业,这就要求当前社会的从业者必须具备最基础的计算机应用能力,因此各大高校将计算机基础应用作为重点教学内容之一[17]。然而，在全栈教学的进行工作没有想象中的顺利，需要学生对全栈技术中复杂的知识点有清楚的认识，而同样也是因为知识的繁杂，在老师批改作业时，效率低、正确率难以保障的问题日益突出。为了解决这一问题，我开发了一个全栈编程题目测评系统。本文根据这个情况，认为开发一款可以帮助老师进行批改作业的系统至关重要，使老师可以有更多的时间优化课程，帮助学生化繁为简。不仅如此，通过清楚明了的批改过程与结果的展示，还能激发学生的学习热情，了解自身还未掌握哪方面的知识，加速成为全栈技术人才。为了此项目的成功实现，本文主要进行了以下工作：

研发初期，首先进行了广泛的文献调研，并深入分析了老师使用场景，以明确系统需求，知晓市面上常见测评系统中存在的一些痛点问题并对其进行思考。通过前期长时间的调研与体验工作后，本文对测评系统有了更深刻的体会，对之后的设计与实现有重要意义。

确定了系统开发的目的与可行性之后，本文对系统进行了技术选型。目前企业中，逐渐开始流行前后端分离的设计架构，该架构保证了系统的分离度，最大程度上简化了系统各模块之间调用关系。前端页面本文选用VUE作为开发模板，大量了技术文档与优秀的社区文章简化了上手难度，而后端则使用了传统的SpringBoot框架，作为各大厂商的技术首选，SpringBoot的能力不容置疑。至此本文即完成了技术选型任务。

开发过程中，本文注重对系统架构的设计，并确立了系统的总体为分层架构，只有将职责分离，才能保证系统的稳定与可维护性。随后，本文采用了分模块开发的方式，通过模块之间互相请求服务的上市，保证系统的低耦合度也减少了模块内冗余代码。最终，通过在虚拟机中部署服务器并使用Linux系统中的脚本文件，完成了系统最重要的白盒测试模块，同时，还借助服务器的自动部署接口，实现了系统的提交作业模块。对于系统的可移植性方面，利用开发工具的打包功能，可以直接将前端后端代码归档打包，在之后的部署环境中，只需简单几行命令即可完成。

全栈编程题目测评系统的成功开发后，本文对其进行了大量的测试。从最开始的登录页面一直深入到最终的完成测评，每个过程都进行了细致的测试。在测试工程中，也发现了不少的问题，经过不断的调试、抓包一一解决。今后在本系统真正的运用过程中，相信其能经受考验，越来越健壮。

## 展望

在线考试系统的出现为计算机基础考核提供了更可靠的新渠道,但由于其诞生时间尚短,因此还需针对实际教学情况进行设计,并通过不断研发以使其能够实现的功能得到强化与完善,帮助教育工作者完成考核任务[18]。通过持续的改进和优化，相信该系统将成为全栈教学的重要支撑，为教育教学事业带来新的发展机遇。”对于在线评测系统的研究，未来还需要更多地计算机类人才参与进来，对于计算机学科来说，这不仅可以提升计算机类人才的变成水平，也可以更好地完善和提升未来考试系统的更加丰富、灵活的功能.”

参考文献

1. 周旭. 浅析“互联网+”背景下高职院校创新创业型人才培养模式[J]. 四川劳动保障, 2024, (02): 42-43.
2. 姜一波. 基于SpringBoot+Vue的在线考试系统设计与实现[J]. 无线互联科技, 2023, 20 (23): 68-71.
3. 王鹰汉, 明小波. 基于Vue的在线考试系统设计与实现[J]. 无线互联科技, 2023, 20 (06): 52-54+92.
4. 李西明, 梁志才, 刘龙浩,祝胜林.基于微服务架构的在线评判系统设计与现[J].软件导刊,2023,22(08):144-150.
5. 孙奕凡.高中信息技术教学情境下在线评测系统的迭代与融合[J].电脑编程巧与维护,2023,(10):86-89.
6. 李环，基于 ACM 竞赛的程序设计类课程教学改革研究[J].计算机教育,2016(03):115-118.
7. 王腾,姚丹霖.Online Judge 系统的设计开发[J].计算机应用与软,2006(12):129-130+137.
8. Anna, F.; Paulina, D.; Michal, K.; Dominik, M., The Conception of Test Fields for Fast Geometric Calibration of the FLIR VUE PRO Thermal Camera for Low-Cost UAV Applications. Sensors 2022, 22 (7): 2468-2468.
9. Chen, J.; Jian, C.; Hailan, P., Design of Man Hour Management Information System on SpringBoot Framework. Journal of Physics: Conference Series 2020, 1646 (1):012136.
10. 黑马程序员.什么是MyBatis? MyBaits有哪些优点？[EB/OL].[2023/04/26].https://baijiahao.baidu.com/s?id=1761874817126246553&wfr=spider&for=pc.
11. 赵莹 , 胡畅达 , 王国宇 . 三种关系型空间数据库比较 [J]. 科技创新与应用 ,2021,11(19):62-64.
12. 刘卓华, 冉瑞生. 基于JSP模型的客户管理系统设计与实现[J]. 电脑知识与技术, 2019, 15 (18): 64-67.
13. 邓佳. 软件测试技术与发展趋势浅析[J]. 数字通信世界, 2020, (02): 136.
14. 项楠. 软件测试技术与测试管理分析[J]. 科学技术创新, 2018, (31): 75-76.
15. 操心慧, 温智聪. 基于JavaWeb的线上考试系统的设计与实现[J]. 现代计算机, 2023, 29 (24): 90-96.
16. 金圣道. 在线考试及试卷分析系统的设计与实现[J]. 电子技术与软件工程, 2018, (07): 170-171.
17. 邹海蓉. 系统测试的意义及系统可测试性设计[J]. 现代信息科技, 2024, 8 (04): 92-96.
18. 王运成. 计算机基础在线考试系统的设计与实现[J]. 信息与电脑(理论版), 2020, 32 (15): 94-96.