

第三届“渝晋鲁·双体杯”

三校大学生计算机应用大赛

开发文档

财务管理系统设计报告

作品名称： 财务管理系统

作 者： 熊 奎 郑明欣 陈 超

版本编号： 1.0.0

填写日期： 2023 年 11 月 15 日

目录

第 1 章 系统定义.....	1
1.1 背景.....	1
1.2 项目意义.....	1
1.3 实现意图.....	2
1.4 目标功能.....	2
第 2 章 系统需求分析.....	3
2.1 系统功能性和非功能性需求.....	3
2.2 可行性分析.....	3
2.3 收益分析.....	3
2.4 技术选型.....	3
第 3 章 系统设计.....	5
3.1 系统功能设计.....	5
3.2 技术架构.....	6
3.3 系统流程图.....	6
3.4 数据库设计.....	7
第 4 章 系统实现.....	8
4.1. 开发环境与工具选择.....	8
4.2. 选择原因与环境要求.....	8
第 5 章 系统测试.....	10
5.1. 项目测试方法介绍.....	10
5.2. 功能测试过程说明.....	10
5.3. 系统运行界面.....	11

第1章 系统定义

1.1 背景

在教育管理中，特别是学校和班级层面，财务管理一直是一项繁重而复杂的任务。传统的手工财务管理方法存在着众多不足，包括但不限于数据记录容易出错、信息不透明、审计难度大等问题。这些问题影响了财务活动的顺畅进行，也使得管理人员难以全面、及时地了解学校或班级的财务状况。

随着科技的迅猛发展，数字化转型成为提高效率和信息透明度的必然趋势。财务管理作为管理的核心之一，迫切需要一个现代化的解决方案。数字化财务管理系统的引入将有助于解决传统财务管理方式的弊端，提高财务工作的效率，降低管理成本，同时为决策提供更加精准的数据支持。

在学校或班级层面，经常需要进行班费的收取、报销、以及其他与财务相关的活动。而这些活动的繁琐性和复杂性使得传统的手工管理难以满足实际需求。因此，一个定制化的财务管理系统将能够更好地满足学校或班级的特定需求，简化财务管理流程，提高管理效率。

考虑到上述的背景和需求，我们决定发起并开发这个财务管理系统。该系统旨在整合先进的数字技术，提供全面、高效、安全的财务管理服务，以满足学校和班级管理者的需求，从而更好地推动教育管理的现代化和智能化。通过数字化财务管理系统的实施，我们期望能够实现财务管理工作的全面优化，提高教育的整体水平，为学校或班级的可持续发展提供有力支持。

1.2 项目意义

(1) 提高管理效率：通过引入自动化财务管理，系统将简化繁琐的手工操作，提高管理效率，减少人为错误。

(2) 增加透明度：系统能够实时记录和展示财务活动，为管理员和用户提供了清晰的财务信息，降低信息不对称的风险。

(3) 提供便捷服务：用户可以通过系统轻松进行账务登记、查询和管理，实现了更加便捷的财务服务。

1.3实现意图

(1) 用户友好设计：系统注重用户体验，通过简洁明了的界面和直观的功能，确保用户能够迅速上手使用。

(2) 数据安全保障：引入系统后，数据的存储和处理将更加安全可靠，减少了手工处理中可能存在的风险。

(3) 统计分析功能：提供对财务数据的统计和分析功能，帮助管理员更深入了解资金状况，从而更好地做出决策。

1.4目标功能

(1) 用户管理

① 账户管理：包括普通用户和管理员账户的登录与登出，确保系统安全性。

② 用户操作：允许管理员对用户进行增加、删除、修改和查询等操作，保障用户信息的完整性和准确性。

(2) 凭证管理

① 凭证录入：提供收款凭证和付款凭证的录入功能，确保财务活动被准确记录。

② 凭证导出与保存：允许用户在线导出凭证并保存，方便后续审计和管理。

③ 凭证查询：提供对凭证的查询功能，以使用户随时查阅。

(3) 收支统计

① 收入统计：对班费余额、收缴情况、收缴次数等进行详细统计，为财务决策提供数据支持。

② 支出统计：统计报销情况，记录报销金额等信息，全面掌握支出状况。

(4) 通知发布与查看

① 实时发布通知：管理员能够实时发布通知，确保信息的及时性和准确性。

② 实时查看通知：用户能够方便地查看到管理员发布的通知，增加信息传递效率。

第2章 系统需求分析

2.1 系统功能性和非功能性需求

(1) 功能性需求

用户管理模块实现普通用户和管理员账户的安全登录与登出，允许管理员对用户进行增加、删除、修改、查询等操作。

凭证管理模块实现收款凭证和付款凭证的在线录入，支持导出和保存功能。提供凭证的查询功能，支持按时间、类型等条件进行筛选。

收支统计模块统计班费余额、收缴情况、收缴次数，统计班费报销情况，记录报销金额等信息。

通知模块管理员能够实时发布通知，用户能够实时查看通知。

(2) 非功能性需求

不同角色的用户有不同的权限，管理员拥有最高权限。对敏感信息进行加密，确保数据的安全性。

2.2 可行性分析

① 可用性：提供直观、易用的界面，使用户能够方便快捷地完成操作。保证系统在合理时间内响应用户请求，提高用户体验。

② 经济可行性：系统采用 **Vue2** 作为前端框架、**Spring Boot** 作为后端框架，结合 **MyBatis** 进行数据库操作。由于使用开源技术，软件开发成本主要包括团队成员的工资、培训成本和服务器成本。

2.3 收益分析

引入财务管理系统后，预期将提高工作效率，自动化的财务管理系统将减少手工操作，提高管理效率，节省人力成本。降低管理成本，管理员可以更轻松地进行财务管理，减少错误发生的可能性，降低管理成本。增强决策支持，数据的准确性和实时性将增强管理者的决策支持，有助于制定更科学的财务策略。

2.4 技术选型

采用 **Vue2+Spring Boot+MyBatis** 的技术栈是现代且成熟的选择，有大量社区支持和丰富

的资源。这有助于确保系统在开发和维护过程中能够获得良好的技术支持。

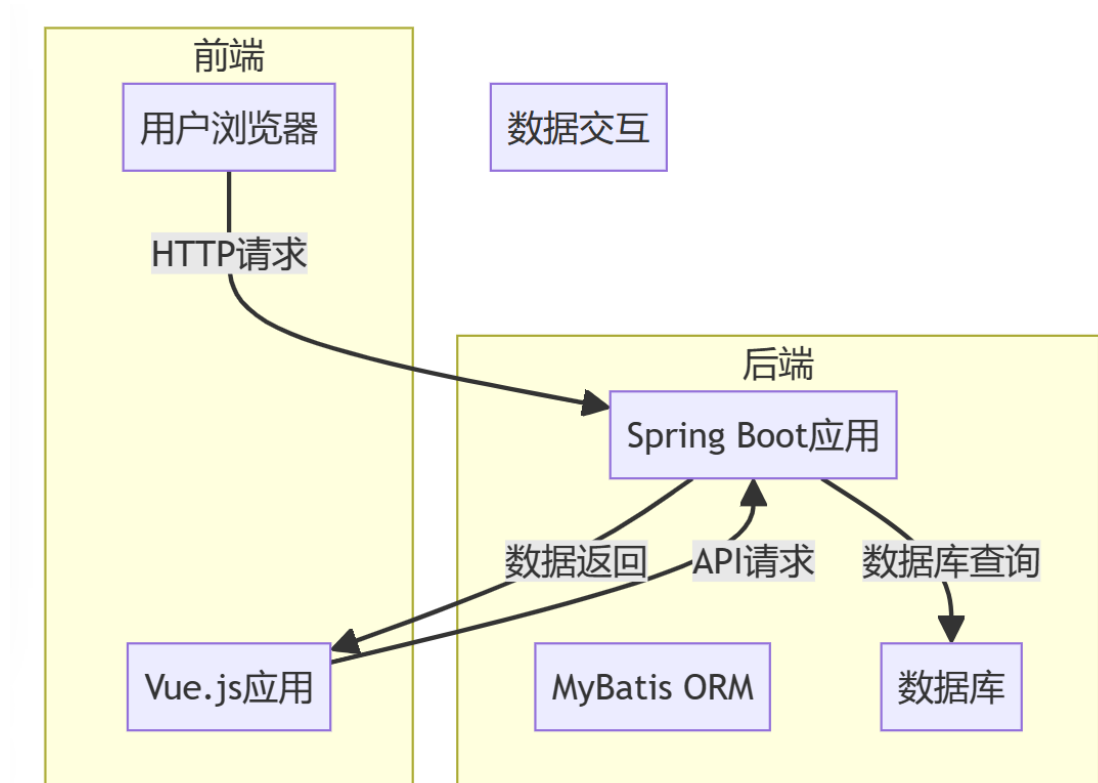
开发小组团队具备 **Vue2**、**Spring Boot** 和 **MyBatis** 等相关技术的知识，可以实现系统的开发。

第3章 系统设计

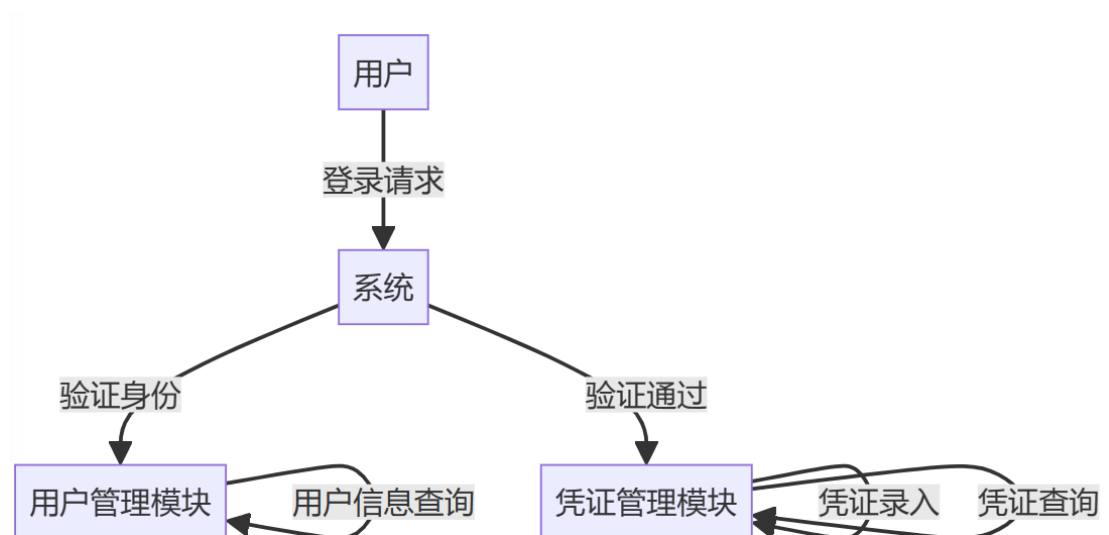
3.1 系统功能设计



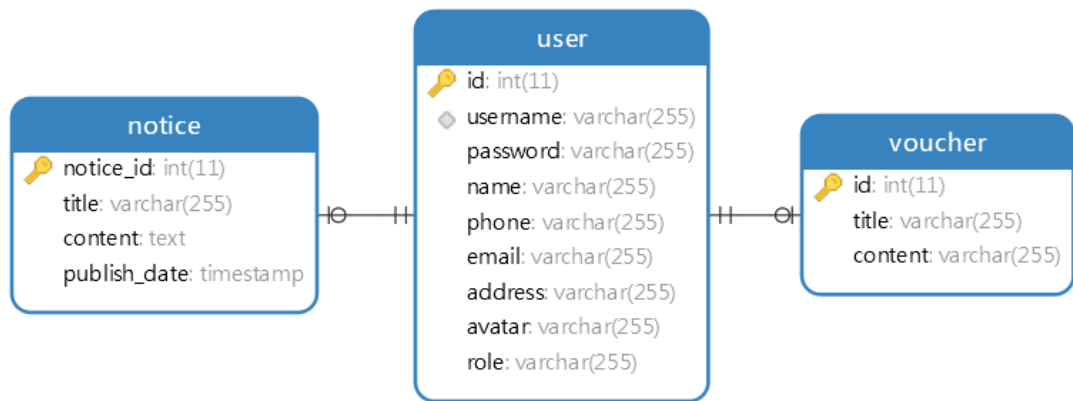
3.2技术架构



3.3系统流程图



3.4数据库设计



第4章 系统实现

4.1. 开发环境与工具选择

(1) 开发平台

前端开发： Vue.js

后端开发： Spring Boot

数据库： MySQL

API 测试： Postman

(2) 集成开发环境（IDE）

Java 开发： IntelliJ IDEA 2021

前端开发： Visual Studio Code

(3) 数据库管理工具

Navicat Premium 16

4.2. 选择原因与环境要求

(1) Vue.js（前端）

Vue.js 是一款轻量级、灵活的 JavaScript 框架，易于学习和使用，适用于构建现代化的单页面应用。支持主流浏览器，可以在开发者本地运行，不需要复杂的配置。

(2) Spring Boot（后端）

Spring Boot 是一款基于 Spring 框架的快速开发框架，简化了 Java 应用的开发过程，提供了强大的功能和良好的扩展性。支持 Java 开发环境，IntelliJ IDEA 是 Spring Boot 的推荐 IDE，可以提高开发效率。

(3) MySQL（数据库）

MySQL 是一款成熟、稳定且开源的关系型数据库，与 Spring Boot 良好集成，并且适用于中小型应用。MySQL 数据库可以在本地或云上搭建，支持多种操作系统，Navicat Premium 作为数据库管理工具，提供了友好的界面和强大的功能。

(4) Postman（API 测试）

Postman 是一款功能强大的 API 测试工具，可以轻松测试、调试和文档化 API。可以在本地运行，支持各种操作系统，通过 Postman 可以模拟 HTTP 请求，测试后端 API 的各种功

能。

(5) IntelliJ IDEA 2021 (Java 开发)

IntelliJ IDEA 是一款强大的 Java 集成开发环境，具有丰富的插件和功能，能够提高开发效率。支持 Java 开发，可以在 Windows、Mac 和 Linux 等操作系统上运行。

(6) Visual Studio Code

Visual Studio Code 是一款轻量级但功能强大的前端开发工具，支持 Vue.js 等前端框架。可以在 Windows、Mac 和 Linux 等操作系统上运行，支持多种前端开发语言。

第5章 系统测试

5.1. 项目测试方法介绍

(1) 单元测试

使用 JUnit 或 TestNG 等单元测试框架，对项目中的各个函数、方法进行测试。通过独立单元的测试，验证其功能是否按照预期工作，提高代码的健壮性。

(2) 集成测试

结合单元测试，测试不同模块之间的协作和集成情况。确保各个模块能够协同工作，检测集成过程中的潜在问题。

(3) 系统测试

对整个项目进行测试，包括功能、性能、安全性等方面。验证系统是否满足需求，发现和修复潜在问题，提高系统质量。

5.2. 功能测试过程说明

(1) 单元测试

- ① 选择一个独立的函数或方法，例如用户登录功能。
- ② 编写单元测试用例，包括正常情况和异常情况。
- ③ 运行单元测试，检查测试结果是否符合预期。
- ④ 修复代码中的问题，重新运行测试。

(2) 集成测试

① 确定集成测试的范围，选择需要测试的模块和其交互情况，例如用户管理模块和凭证管理模块的集成。

- ② 编写集成测试用例，测试各个模块之间的协作。
- ③ 运行集成测试，观察是否有数据传递、接口调用等方面的问题。
- ④ 修复集成测试中发现的问题，重新运行测试。

(3) 系统测试

① 根据需求文档编写系统测试用例，覆盖系统的各个功能点，例如用户管理、凭证管理、收支统计等。

- ② 运行系统测试，测试系统的功能、性能、安全性等方面。
- ③ 收集测试结果，记录和分析测试中发现的问题。
- ④ 修复系统测试中发现的问题，重新运行测试。

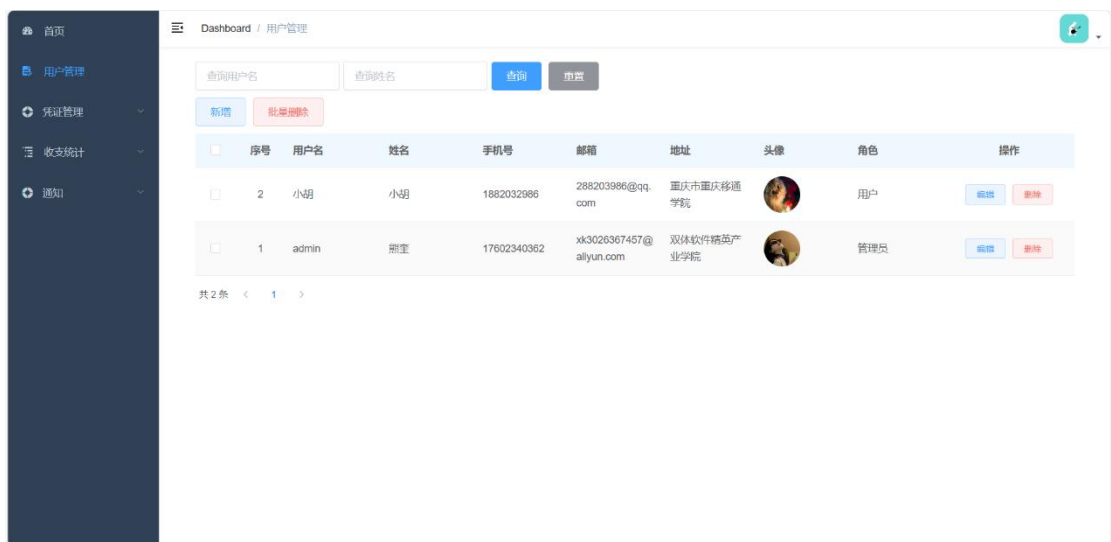
5.3. 系统运行界面

系统代表性运行界面如下：

(1) 主页



(2) 用户管理界面



(3)付款凭证

首页

用户管理

凭证管理

收款凭证

付款凭证

查凭证

收支统计

通知

Dashboard / 凭证管理 / 付款凭证

付款凭证

科目
ACCOUNT

日期:
年 月 日

总号
GEN NO.
第
VOUCHER NO.

摘要 DESCRIPTION	会计科目 ACCOUNT 总账科目 GEN.LED.A/C	明细科目 SUB.LED.A/C	金额 AMOUNT 亿 千 百 十 万 千 百 十 元 角 分	记账 PR
				√
				√
				√
				√
				√
				√
附单据	张 ATTACHMENTS	合计 TOTAL		

会计主管:
SUPERVISOR

复核:
CHECKED

记账:
ENTERED

出纳:
CASHIER

制单:
PREPARED

收款人:
RECEIVER

(4)发布通知

Dashboard / 通知 / 发布通知

标题

请输入标题

通知

请输入

提交时间

选择日期和时间

发布