

**课 程 设 计**

**课程设计名称：** 软件测试课程设计

**专 业 班 级 ：** 软件2102

**学 生 姓 名 ：** 朱明宇

**学 号 ：** 211070100227

**指 导 教 师 ：** 段爱玲

**课程设计时间：** 2023.12.25—2024.1.5

**软件工程专业课程设计任务书**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **学生姓名** | **朱明宇** | **专业班级** | | 软件2102 | **学号** | | **211070100227** |
| **题 目** | **银行账户管理系统** | | | | | | |
| **课题性质** | A．工程设计 | | **课题来源** | | | D．自拟课题 | |
| **指导教师** | **段爱玲** | | **同组姓名** | | | **无** | |
| **主要内容** | 根据所学的软件测试的基本理论，测试方法以及测试工具，对银行账户管理系统进行功能测试和单元测试。主要内容包括：  （1）测试需求  （2）测试计划  （3）测试用例设计  （4）测试的执行过程  （5）测试总结和评价 | | | | | | |
| **任务要求** | （1）能够了解测试计划的作用和内容组成；  （2）能够根据功能需求派生测试需求；  （3）能够设计规范、准确的测试用例；  （4）撰写、提交课程设计报告，能够对完成的工作清晰描述分析、设计、并能做出评价，能够回答老师提出的问题。 | | | | | | |
| **参考文献** | [1]《软件测试方法与技术》（第三版）. 朱少民.清华大学出版社，2015年.  [2]《软件测试开发理论项目实战教程》. 霍格沃兹开发学社. 人民邮电出版社, 2022年.  [3]《软件测试实战教程》高科华.清华大学出版社.2021年.  [4]《软件故障注入关键技术研究》潘庆和.哈尔滨工业大学出版社.2021年  [5]《基于测试性设计的软件综合研究》计算机测量与控制.2022年 | | | | | | |
| **审查意见** | **同意**  **教研室主任签字：8087f98da7960912759f1007e9f24d3**  **2023年 12 月25日** | | | | | | |

目 录

[1.测试需求分析 1](#_Toc154175599)

[1.1系统概述 1](#_Toc154175600)

[1.2测试需求 2](#_Toc154175601)

[2. 测试计划 3](#_Toc154175602)

[3. 用例设计 4](#_Toc154175603)

[4. 功能测试 7](#_Toc154175604)

[4.1](#_Toc154175600)开户注销 7

[4.2存储交易 9](#_Toc154175600)

4.3业务查询 11

[5. 单元测试 12](#_Toc154175605)

[6. 集成测试 14](#_Toc154175606)

[7. 测试总结 16](#_Toc154175607)

[8.心得体会 17](#_Toc154175607)

[9.参考文献 18](#_Toc154175607)

# 测试需求分析

## 1.1系统概述

为保证某银行账户管理系统的可用性，对其进行软件测试，以下是一个对该银行账户管理系统的软件测试的测试概述：

测试目标：

1.确保银行账户管理系统的功能性、性能、安全性、和可用性等方面符合预期要求。

2.发现和修复潜在的缺陷、错误和漏洞，确保系统的稳定性和可靠性。

3.验证系统的用户界面和交互设计是否符合用户期望，提供良好的用户体验。

测试范围：

1.银行账户管理系统的所有核心功能，包括账户创建、登录、存款、取款、转账、交易历史记录等。

2.系统的性能和负载容量，以及与外部系统（如数据库等）的集成。

3.系统的安全性，包括密码安全、权限管理、防护措施等。

4.用户界面的可用性和易用性，以及系统的错误处理和可访问性。

测试策略：

1.结合黑盒测试和白盒测试方法，对系统进行功能性和安全性等方面的测试。

2.使用适当的测试技术，如边界值分析、等价类划分等，针对不同的测试需求进行测试。

3.对系统编码具体某个模块进行单元测试，包括使用覆盖率方法运行，检测代码的复用性与高效性。

4.进行系统集成测试，确保与其他系统的正确集成和交互。

5.进行用户界面测试，关注用户体验和可用性方面的问题。

## 1.2测试需求

测试需求分析是软件测试过程中的重要一步，它有助于明确测试的目标和范围，确保测试团队了解应用程序的功能和预期行为。以下是一个针对银行账户管理系统的详细测试需求分析：

功能性测试需求：验证账户创建功能，确保能够成功创建新账户，并生成唯一的账户号码；测试账户登录功能，验证账户持有人能够正确登录，并且登录失败时能够给出适当的错误提示；验证账户余额计算的准确性，包括存款、取款和转账等操作后的余额变化；测试账户交易历史记录功能，确保所有交易都被正确记录，并能够正确显示和检索历史记录。

单元测试需求：针对该银行账户管理系统的源代码进行测试。对其中某一个具体的函数进行基于Junit的单元测试，并且使用覆盖率的方法运行，提高代码局部的高效性。

集成测试需求：该银行账户管理系统的数据存储主要存储于Mysql数据库中，在实际测试中使用到了外检约束、视图、存储过程以及触发器等多种数据库模块代码。因此在系统实际运行时用必要对其整体进行集成测试，以防止出现不可兼容的问题。

可用性测试需求：测试用户界面的友好性和易用性，确保用户能够轻松地进行操作并找到所需的功能；验证系统的错误处理和提示信息，确保用户在出现错误时能够获得清晰的错误提示和帮助信息（如用户在进行取款操作时，如果取款金额大于银行卡里的存款，则会弹出提示：“余额不足”等）。

# 测试计划

本次银行账户管理系统的测试主要包括对运行时各种情况系统反应的功能测试、对银行用户开户时的代码单元测试以及系统与数据库结合时相关存储过程体现的集成测试。以下是详细的测试计划：

首先编写用于功能测试的测试用例，每个测试用例要具体包括测试标题、测试模块、预置条件、输入数据、预期结果、实际结果以及最后的测试结果。根据这些测试用例，对系统进行功能测试，包括开户注销模块、存储交易模块和业务查询三个模块。

之后使用IDEA中的相关插件，对银行账户管理系统的开户注销模块中BankUser类进行Junit单元测试。这是一种白盒测试，测试的目的在于检验系统代码的复用性与高效性。为此编写了测试类BankUserTest代码，对其进行覆盖率运行。对结果中元素和其对应的类、方法以及行的相关覆盖率进行统计和分析，以不断提高代码高效性。

对于银行账户管理系统而言，相关信息数据的存储至关重要。因此最后进行了与数据库集成的系统集成测试。包括对数据的增删查改等，通过mysql数据库的视图、存储过程以及触发器等功能，从多个角度对系统整体进行与数据库相关的集成测试.

# 用例设计

本银行账户管理系统的概要设计主要分为开户注销。存储交易以及业务查询三个模块，总体功能模块图如下图3.1所示：

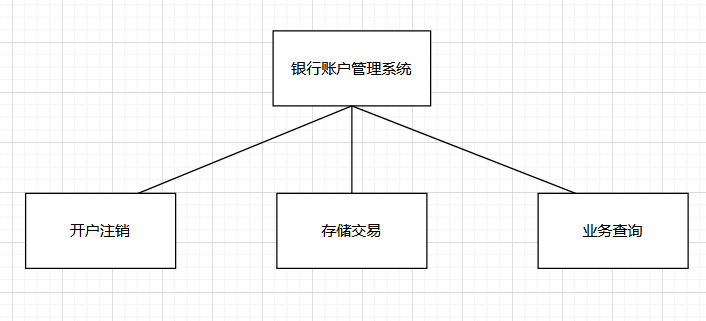


图3.1 总体功能模块图

对于开户注销模块，功能由细分为“用户注册”、“忘记密码”、“修改密码”、“注销账户”以及“信息修改”四个功能模块。开户注销的功能模块图如下图3.2所示：

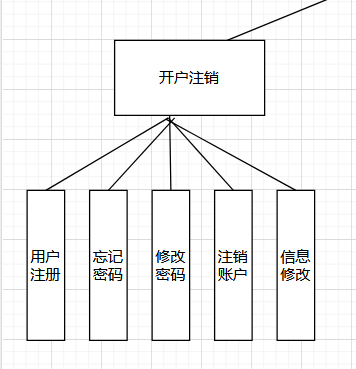


图3.2 开户注销功能模块图

对于存储交易模块，功能细分为“存款”、“取款”以及“转账”三个功能模块。存储交易的功能模块图如下图3.3所示：

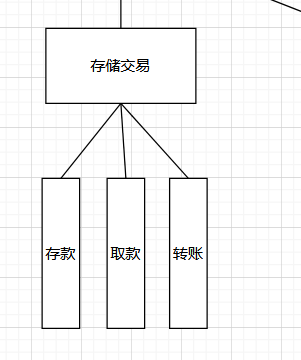


图3.3 存储交易功能模块图

对于业务查询模块，功能细分为“信息核对”和“流水查询两个功能模块。其中借助Mysql视图和存储过程可以实现根据储户银行卡号等个人信息进行精确查找。业务查询的功能模块图如下图3.4所示：

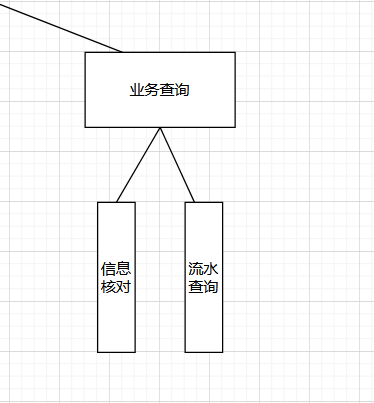


图3.4 业务查询功能板块图

下面对以上“开户注销”、“存储交易”和“业务查询”三个模块分别设计不同的测试用例。如下表3.1所示：

表3.1 功能运行测试测试用例

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试用例 | | | | | | |
| 测试标题 | 测试模块 | 预制条件 | 输入数据 | 预期结果 | 实际结果 | 测试结果 |
| 用户注册银行账户 | 开户注销 | 用户成功进入系统 | 开户用户个人信息 | 弹出提示注册成功 | 弹出提示注册成功 | PASS |
| 用户注销某银行卡 | 开户注销 | 用户成功开户并有相关权限 | 持卡人姓名卡号交易密码等 | 弹出提示注销成功 | 弹出提示注销成功 | PASS |
| 某用户取钱数大于账户余额 | 存储交易 | 用户输入正确交易密码进入 | 取钱金额 | 弹出提示余额不足 | 弹出提示余额不足 | PASS |
| 用户A向用户B提出转账 | 存储交易 | 用户进入转账界面 | 对方交易账号交易量和密码 | 弹出提示转账成功 | 弹出提示转账成功 | PASS |
| 系统管理员查看存款交易 | 业务查询 | 管理员进入数据库管理视图 | 点击查看存款视图 | 视图展示存款交易 | 视图展示存款交易 | PASS |
| 用户忘记登录密码 | 开户注销 | 用户成功进入系统 | 开户用户个人信息 | 弹出提示登录密码 | 弹出提示登录密码 | PASS |

# 功能测试

下面根据表3.1列出的功能运行测试测试用例进行相关的功能测试：

## 4.1开户注销

对于用户的注册，如下图4.1所示，需要输入用户个人的相关信息，输入成功后即可产生如下图4.2所示的“注册成功”界面。经检验，测试用例通过。



图4.1 用户注册

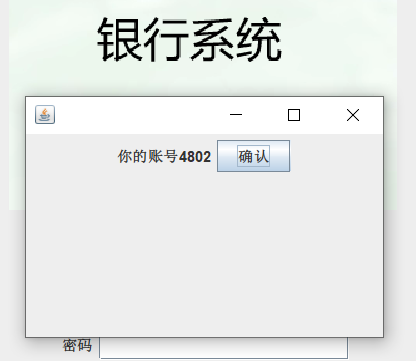


图4.2 注册成功

如果用户忘记密码，需要在用户登录界面点击“忘记密码”，然后在弹出界面中输入姓名和手机号，如下图4.3所示。输入成功后，点击“忘记密码”，即可弹出当前用户的账号和密码，如下图4.4所示。经检验，测试用例通过。

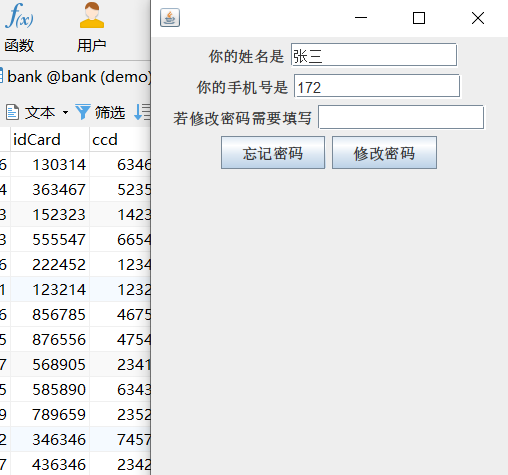


图4.3 忘记密码登记

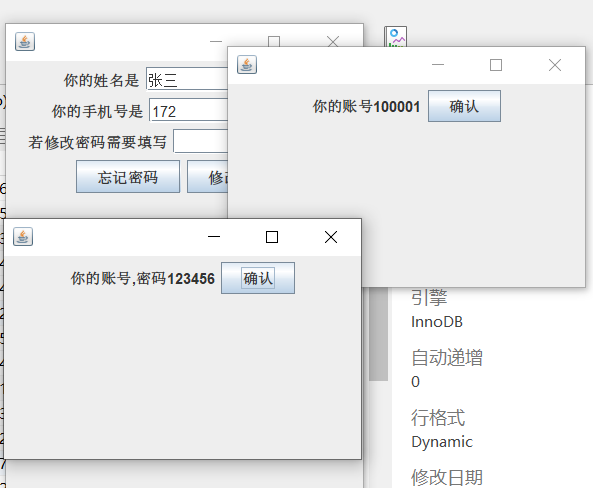


图4.4 忘记密码成功

## 4.2存储交易

当用户A需要取款，但其账户里的余额不够这些取款时，系统会对其进行处理。弹出提示“余额不足”。详细见下图4.5所示。经过这种极端情况的检验，测试用例通过。

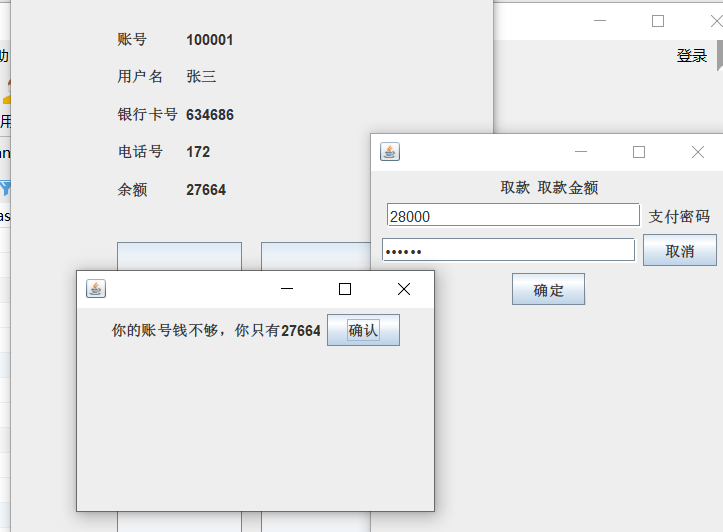


图4.5 取款超额处理

当用户A需要向用户B转账时，需要用户A输入转账金额、支付密码以及目标账号，如下图4.6所示，输入成功后即可弹出如下图4.7所示的“交易已被记录弹窗”。

经检验，测试用例通过。



图4.6 转账信息处理

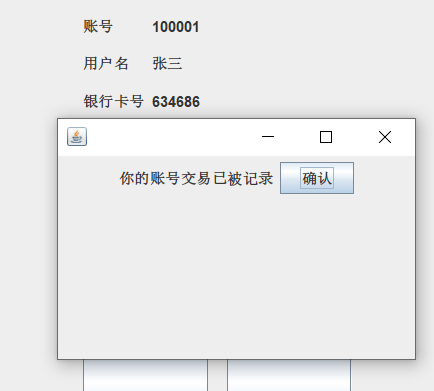


图4.7 转账成功处理

## 4.3业务查询

下面进行交易记录查询的相关测试。对某一用户而言，需要可以随时查到该账户的历史交易记录。如下图4.8所示，对于用户“张三”，点击“查看流水按钮”，即可显示该用户的历史交易记录，交易记录如下图4.9所示。经检验，测试用例通过。



图4.8 用户账户信息



图4.9 用户交易记录

# 单元测试

JUnit是一个流行的Java单元测试框架，用于编写和执行单元测试。它提供了一组注解和断言方法，使得编写测试用例变得简单和可维护。

单元测试用于验证代码的最小可测试单元（通常是函数、方法或类）是否按照预期工作。它的目标是在代码编写过程中尽早地发现和修复缺陷，确保被测试的代码模块的正确性和可靠性。

下面以Bean中的BankUser类为例，编写测试类BankUserTest对银行账户的创建类进行单元测试。

import org.junit.Test;

public class BankUserTest {

BankUser t1 = new BankUser();

BankUser t2 = new BankUser(123456789, "John Doe", 123456, 1234567890, 987654321, 789456, 1234567890);

@Test

public void test() {

t1.getBankid();

t1.getCcd();

t1.getCcdpassword();

t1.getId();

t1.getName();

t2.getBankid();

t2.setBankid(123456789);

}

private void assertEquals(String regular, String type) {

}

}

public BankUser(int bankid, String name, int password, int idCard, int ccd, int ccdpassword, int phone) {

this.bankid = bankid;

this.name = name;

this.password = password;

this.idCard = idCard;

this.ccd = ccd;

this.ccdpassword = ccdpassword;

this.phone = phone;

}

使用覆盖率运行后BankUser和BankUserTest中重复行、方法以及类的覆盖率总结如下图5.1所示：

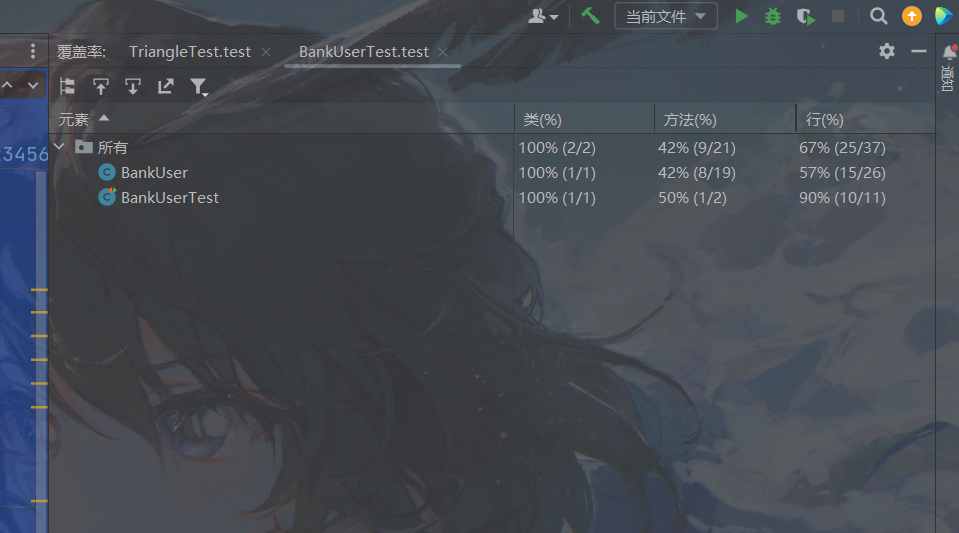


图5.1 覆盖率结果

其中，对测试类BankUserTest而言，空的实体t1和以及含有如用户姓名、银行卡号等信息的实体t2均已经被全部覆盖，如下图5.2所示。说明该BankUser类的可复用性和高效性以得到一定程度的保障。即针对该模块的Junit单元测试测试通过。

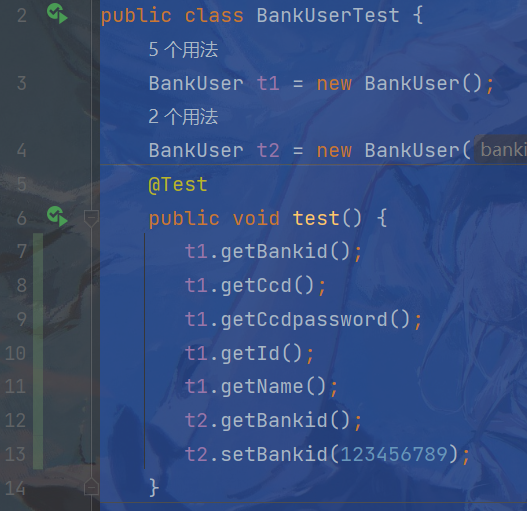


图5.2 覆盖运行结果

# 集成测试

在上述单元测试的成功基础上，我们需要把多个单元整合起来。最终再把JAVA可视化前端和mysql数据库后端集成，重点针对数据的存储和查阅进行集成测试。

系统使用JDBC的方式在JAVA代码中嵌入部分mysql源码进行增删查改操作。为了方便管理员更好地管理整个银行账户管理系统，利用数据库的视图功能完成以下操作：

统计所有银行卡号以某某数字结尾的用户信息；统计所有银行账户存款大于某一数额的用户信息；统计某短时间内操作类型为“取款”的交易信息；统计某一用户的所有操作信息。

部分集成演示如下图6.1所示：

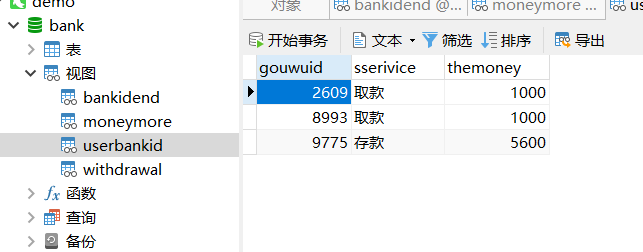


图6.1 视图集成测试

为了更好地体现储户余额的变动，以做到金额变动的实时更新，确保系统的安全性，为储户余额的变化设计了两个函数处理存储过程，如下图6.2所示：

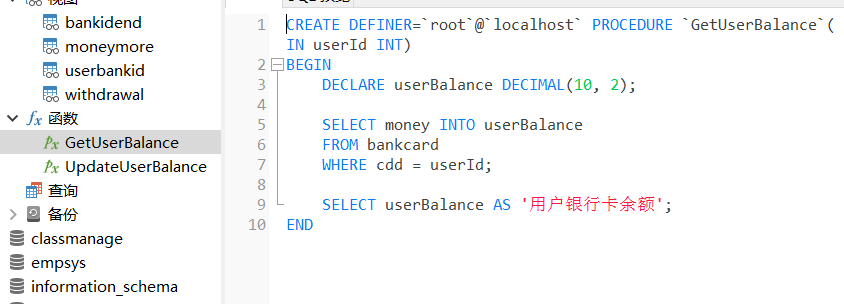


图6.2 存储函数实例

然后对该存储过程进行集成测试，如下图6.3所示，输入参数即用户的存储ID，测试中输入“123456”。

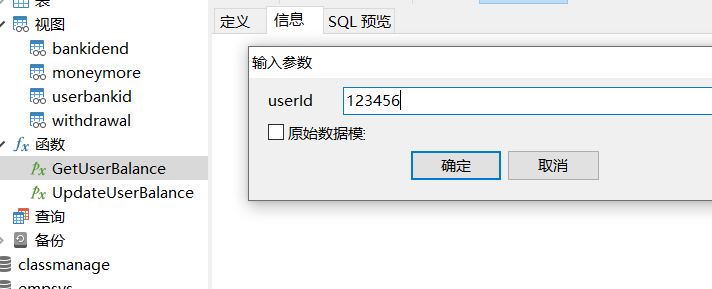


图6.3 测试参数输入

参数成功输入后，机会出现如下图6.4所示的界面，打印显示该测试用户的银行卡余额。

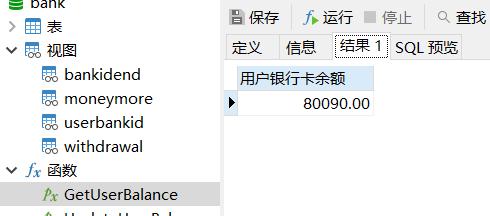


图6.4 数据库函数集成测试

综上，对该银行账户管理系统的的数据库集成测试测试通过。

# 测试总结

在对银行账户管理系统进行测试时，采用了不同的测试方法，包括功能测试、Junit单元测试和集成测试，并结合数据库进行了全面的测试。以下是对每个测试阶段的总结：

在功能测试阶段，对整个银行账户管理系统的功能进行了验证。编写了一系列测试用例，覆盖了系统的各个功能点，包括账户创建、存款、取款、转账等。通过这些测试用例，确认系统在各种不同场景下的功能是否正常运行。还验证了系统的边界条件和异常情况处理，以确保系统能够正确地处理各种情况。功能测试确保系统的各项功能符合预期，并且能够满足用户需求。

在针对某一特定模块的测试中，使用了Junit框架进行单元测试。通过编写一系列针对该模块的测试用例，验证了该模块的各个方法和函数是否按照预期进行操作。测试了模块的各种输入和输出组合，以及各种边界条件和异常情况。通过Junit单元测试，能够快速发现和修复模块中的问题，并确保模块的功能正确性。  
 在集成测试阶段，将银行账户管理系统与数据库进行了集成测试。验证了系统与数据库之间的数据交互是否正常，包括数据的读取、写入和更新等操作。还测试了系统在高并发情况下的性能和稳定性。通过集成测试，能够确保系统与数据库的协作正常，并且能够处理大量的数据和用户请求。

通过以上三个测试阶段，我们对银行账户管理系统进行了全面的测试，并确保系统的功能正常、模块的单元测试通过，以及与数据库的集成测试成功。在测试过程中，发现并修复了一些问题，包括功能缺陷、模块错误和性能瓶颈等。还对测试用例进行了优化，增加了对边界条件和异常情况的覆盖率。

总体而言，测试工作对银行账户管理系统的稳定性、正确性和性能提供了有效的保障。通过测试，确认系统能够满足用户需求，并且在各种情况下都能够正常工作。建议在发布系统之前进行更多的测试，以进一步提高系统的质量和可靠性。

# 心得体会

经过软件测试的过程，对软件开发和质量保证有了一些深刻的体会。以下是在测试之后的心得体会：

测试是必不可少的：软件测试是确保软件质量的重要环节。通过测试，可以发现和修复软件中的问题和缺陷，提高软件的可靠性和稳定性。测试不仅有助于发现功能上的问题，还能验证软件是否满足用户需求，并确保软件在各种环境和使用场景下都能正常工作。因此，测试是软件开发过程中不可或缺的一部分。

早期测试节省成本：将测试早期纳入软件开发过程中可以大大节省成本和时间。在需求分析和设计阶段时，就应该开始思考和编写测试用例，以确保软件在开发完成后能够满足预期。早期的测试可以帮助发现和解决问题的成本较低，并且减少了在后期修复问题所需的时间和精力。

自动化测试提高效率：自动化测试是提高测试效率和准确性的重要手段。通过使用自动化测试工具和框架，可以快速执行大量的测试用例，并及时发现问题。自动化测试还可以降低人为错误的风险，并提供可重复性和可扩展性。因此，在适当的情况下，应该考虑引入自动化测试来提高测试效率。

测试需求和场景的全面性：在进行测试时，需要确保测试用例覆盖到软件的各个功能点、边界条件和异常情况。仅仅验证主要功能是不够的，还需要对边界情况、异常输入和不同用户角色等进行测试。只有全面覆盖，才能保证软件在各种情况下的稳定性和正确性。

持续改进和学习：软件测试是一个不断改进和学习的过程。通过每一轮测试，可以从中吸取经验教训，并不断改进测试策略和方法。应该关注新的测试技术和工具，保持学习的态度，并与团队成员分享和交流经验。

总而言之，软件测试是确保软件质量的关键步骤。通过测试，可以提高软件的可靠性和稳定性，满足用户需求，并确保软件在各种情况下都能正常工作。通过持续改进和学习，可以不断提高测试效率和质量，为用户提供更好的软件体验。

# 参考文献

[1]《软件测试方法与技术》（第三版）. 朱少民.清华大学出版社，2015年.

[2]《软件测试开发理论项目实战教程》. 霍格沃兹开发学社. 人民邮电出版社, 2022年.

[3]《软件测试实战教程》高科华.清华大学出版社.2021年.

[4]《软件故障注入关键技术研究》潘庆和.哈尔滨工业大学出版社.2021年

[5]《基于测试性设计的软件综合研究》计算机测量与控制.2022年