简介

该项目是基于JSP技术的小型银行管理系统，根据《JSP实用教程（第4版）》中的案例“小蜜蜂手机销售网”进行改进，在原有的基础上增加了银行独有的存款、取款等功能，涉及JSP、Servlet、MySQL、JDBC等相关知识。

该项目是《Web开发技术》课程的大作业，其设计初衷，是通过线上模拟银行业务，熟悉现实业务处理流程，并利用所学知识做一次实际应用，巩固编程水平，积累项目经验，为后续的实习工作打下良好基础。

该项目面向客户和交易员两种身份，主要功能包括客户和交易员的注册和登录，以及各自针对账户信息和业务信息的操作。用户可以进行个人信息查询，例如银行卡信息、身份证号码、客户编号等内容，同时还可以进行一系列常见的银行业务操作，如开户、存款、取款和挂失等。交易员则具备对客户信息和业务信息进行增删改查的权限，可以有效管理系统中的数据。

开发流程

|  |  |
| --- | --- |
| 前端 | HTML + CSS |
| 后端 | JavaBean + JavaServlet + JSP + JDBC |
| 数据库 | MySQL 5.7 |
| 服务器 | Tomcat 10.1.7 |
| 开发工具 | IDEA 2023 + Navicat Premium 15 |
| 项目启动 | 2023.06.19 |
| 项目完成 | 2023.06.25 |

特点

页面简洁，方便测试和演示；

系统流程图、数据库结构图完整，专业性较好；

采用前后端分离的架构和MVC设计模式，业务逻辑清晰；

设计了客户编号，身份证号，手机号等多种信息，模拟业务的真实性较好；

实现了注册、登录、查看信息、客户开户、客户存款、客户取款的基本功能；

注册和登录功能都可以对数据进行合法性校验，且登录功能具备密码加密和安全验证机制，安全性较好；

不足

页面简陋，不够美观；

代码杂乱，缺少注释，可读性差；

页面结构紧凑，没有分包，可维护性较差；

注册和登录成功时直接跳转，没有相关提示，页面交互性较差；

客户和管理员的页面分开，而不是共用单个页面，造成代码冗余；

交易员仅具备查看信息功能，编辑功能和删除功能尚未实现，功能不完善；

未添加完整约束，在未登录的情况下，交易员仍然可以对客户信息进行查看，存在逻辑漏洞；

部分页面未对表单信息进行过滤，如开户，存款等页面，即使表单信息非法也可以录入数据库，可靠性较差。

结构

1. 文件结构

CSS结构图

JSP结构图

JavaBean结构图

JavaServlet结构图

1. 数据库结构

表结构图

E-R图

1. 流程图

JSP中的MVC模式

系统功能模块图（客户）

系统流程图（客户）

Java文件作用图（客户）

演示

1. 首页
2. 客户
3. 交易员

总结

这个项目开发于大二第二学期，实际历时约一周，由于是第一次开发项目，缺少思路，所以借鉴了课本案例，但在实现注册登录功能后，剩余功能都是独立开发。加上当时正处于期末周，要准备复习和考试，时间紧凑，不得不天天熬夜，所以导致项目总体质量差。

开发过程并不是一帆风顺，尽管有课本的现成案例，但我在实际模仿中仍然会出现许多报错，主要是代码缺漏这种细节性问题和对业务逻辑不够熟悉的技术性问题。如最初的数据库注册驱动问题，由于MySQL5.8以前版本无需在语句中添加.cj，导致我最初的连接数据库一直不成功，界面显示404。还有后面的Servlet因为web.xml配置有误和form表单提交有误，界面显示500。还有因为CSS代码修改和各种路径跳转问题，让我不得不经常测试，导致开发进度缓慢，这一切也体现出我对JSP、Java等后端知识的不熟悉，不过由于前端知识在大一下学期有学习过，所以不太吃力。

值得一提的是，虽然实际开发很繁琐枯燥，但当真正独立开发出一个自己的管理系统时，看着这灵活跳转的页面和精美的页面样式，心中真的会有一种成就感油然而生。也正是实际操作过，才明白和理论差距有多大，丰富了自己开发项目的经验，掌握了更多的技巧，所以学编程必须多实操，特别是上手一个项目，真的可以很快进步。