虚拟校园系统使用说明书

09022217 汪毅书
09022230 韩潇康
09022214 吴培俊
09022236 周方昱
09022310 李　婷

版本：`1.0.0-release`

日期：2024 年 9 月 15 日

目录

1 引言 **5**
1.1 编写目的 5
1.2 背景 5

2 程序系统的分析 **5**
2.1 可行性分析 5
2.2 需求分析 5
2.3 额外扩展功能 6
2.4 开发设计环境 7
2.4.1 IntelliJ IDEA 7
2.4.2 JDK 21 7
2.4.3 MySQL 8.4.2 7
2.4.4 第三方库 7

3 程序系统的结构 **9**

4 用户管理模块设计说明 **9**
4.1 模块背景 9
4.2 需求分析 9
4.3 系统设计 10
4.3.1 界面设计 10
4.3.2 类分析 11

5 学籍管理模块设计说明 **13**
5.1 模块背景 13
5.2 需求分析 13
5.3 系统设计 13
5.3.1 界面设计 13
5.3.2 模块流程图 15
5.3.3 类分析 16

6 教务管理模块设计说明 **17**
6.1 模块背景 17

6.2 需求分析 17
6.3 系统设计 18
6.3.1 界面设计 18
6.3.2 模块流程图 22
6.3.3 类分析 23

7 图书馆模块设计说明 27
7.1 模块背景 27
7.2 需求分析 27
7.3 系统设计 28
7.3.1 界面设计 28
7.3.2 模块流程图 31
7.3.3 类分析 32

8 商店模块设计说明 34
8.1 模块背景 34
8.2 需求分析 34
8.3 系统设计 35
8.3.1 界面设计 35
8.3.2 模块流程图 38
8.3.3 类分析 38

9 其他选做模块 40
9.1 银行模块说明 40
9.1.1 界面设计 40
9.1.2 模块流程图 41
9.1.3 类分析 41
9.2 GPT 内联 42

10 公共模块设计说明 42
10.1 消息信息模型 42
10.1.1 `Request` 42
10.1.2 `Response` 42

11 网络模块设计说明 **42**
11.0.1 服务器端 ServerSocket 43

12 多线程模块设计说明 **44**
12.1 Socket 44
12.1.1 客户端 44
12.1.2 服务器端 Server 44
12.2 输入输出流 44
12.2.1 读取输入 44
12.2.2 发送输出 47

13 数据库设计说明 **47**
13.1 元数据 47
13.2 表设计 48
13.2.1 user 表 48
13.2.2 student 表 49
13.2.3 image_from_client 表 49
13.2.4 image_table 表 49
13.2.5 course 表 50
13.2.6 evaluationofteaching 表 50
13.2.7 grade 表 50
13.2.8 studentselectclass 表 51
13.2.9 teachingclass 表 51
13.2.10 teachingclassenter 表 52
13.2.11 librarybook 表 53
13.2.12 librarycom 表 54
13.2.13 librarytran 表 54
13.2.14 libraryuser 表 54
13.2.15 product 表 55
13.2.16 producttransactionrecord 表 55

目录

1 引言

1.1 编写目的

本系统为东南大学计算机科学与工程专业 2022 级暑期学校专业技能实训课程项目。为了便于项目初期的代码编写和项目后期的代码维护，将代码功能和代码内部框架描述在此文档中，也方便后续查阅。

本文档面向系统开发、维护以及验收人员编写。

1.2 背景

本系统为参考东南大学网上办事大厅开发的虚拟校园系统，采用 Java 语言进行编写。为 C/S 模式，分服务器端和客户端：

服务器端 数据处理端，运行数据库管理系统，处理来自客户端的请求并返回信息。

客户端 面向用户的图形端，运行于用户的计算机上，提供图形化可交互界面给用户，将用户的请求发送到服务器端，对服务器端返回的信息进行显示。

2 程序系统的分析

2.1 可行性分析

通过对学校办事大厅的调查，了解到目前学校办事大厅的整体系统结构，可知本系统已经有类似的实际运用。本系统是参考办事大厅后实现的虚拟校园系统，可便于学校管理以及用户自助业务办理。

2.2 需求分析

考虑到学校中学生和老师常用的内容，项目中包括以下内容：

用户管理 为了实现对于用户的管理，我们设计了登录模块用于认证与登录，方便用户安全登录。

学籍管理 为了实现对学生学籍的管理，我们在学籍管理模块设置了学生端和教务端，方便学生对自己学籍的查询以及教务端可以录入学籍信息以及审核修改信息。

教务管理 为了方便教务进行管理，主要设置了课程管理（选课、退课、查看课表、成绩查询、成绩录入）

图书馆 为了方便学生访问图书馆以及图书馆管理员进行管理，对于学生设置了借书、还书、检索、查看当前图书情况（例如剩余多少、有哪些本）；对于图书管理员设置了添加书、删除修改、查看借阅记录。

商店 由于是面向广大师生群众的商店，分别对购买者和管理者进行功能设置，购买者主要包括：购物车、浏览商店、选择商品添加至购物车、结账，管理者主要包括：添加商品，删除修改商品（包括定价、数量、名称等）。

2.3 额外扩展功能

人脸识别与比对 在用户忘记密码时，为方便用户进行密码修改，决定不采用传统的发送电子邮件或信息的方式进行验证修改，而是采用新式的人脸识别功能进行密码修改与确认。

密码加密环节 为防止密码在传输过程被破译获取，团队对密码进行 RSA 加密传输。

爬虫 为了使展示信息更加真实具体，团队利用爬虫功能爬取了学校讲座、文娱活动等内部信息并展示，使得用户能够获取更加准确的资料。

GPT 对话学习机器人 大模型到来的时代，系统配置一个可以进行聊天学习的机器人是十分有必要的，除去可以进行日常聊天外，还能进行学习资料搜查，辅助学习的功能。

商店智能推荐算法 为了让商店能有更加智能的展示功能，团队采用了 LRU 推荐算法（最近最少使用）对部分商品进行淘汰，而把获胜的商品进行优先展示，这样更能迎合用户的喜好。

2.4 开发设计环境

2.4.1 IntelliJ IDEA

IntelliJ IDEA 是一种商业化销售的 Java IDE 工具软件，由 JetBrains 软件公司开发，提供 Apache 2.0 开放式授权的社区版本以及专有软件的商业版本，开发者可选择其所需来下载使用。本系统采用 IntelliJ IDEA 作为开发用 IDE.

2.4.2 JDK 21

Java 21 是 Java 开发平台最新推出的 LTS 版本。Java 21 提供了数千种性能、稳定性和安全性更新，来进一步优化 Java 语言和平台，从而帮助开发人员提高工作效率。本系统采用 JDK 21 作为开发平台。

2.4.3 MySQL 8.4.2

MySQL 是一个关系型数据库管理系统，由瑞典 MySQL AB 公司开发，目前属于 Oracle 公司。MySQL 是一种关联数据库管理系统，关联数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。本系统采用 MySQL 8.4.2 作为数据库管理系统。

2.4.4 第三方库

1. **OpenCV**

   OpenCV (Open Source Computer Vision Library) 是一个开源的计算机视觉和机器学习软件库，提供了数百个高效的函数用于实时图像处理、视频捕捉和分析。它由 Intel 于 1999 年开发，后来成为跨平台的工具，支持 Windows、Linux、macOS 以及 Android 和 iOS。本系统采用 OpenCV 库处理人脸识别相关功能。

2. **Jsoup**

   Jsoup 是一个用于解析、处理和清理 HTML 的 Java 库。它可以让开发者方便地从网页中提取数据，并对 HTML 进行操作。Jsoup 提供了类似 jQuery 的 DOM 操作 API，因此对前端开发人员来说非常容易上手。本系统采用 Jsoup 库进行网站提取操作。

3. **JetPack Compose（Compose for desktop）**

   Jetpack Compose 是 Google 为 Android 开发推出的现代 UI 工具包，它基于声明式编程，简化了用户界面的构建。Compose for Desktop 是 Jetpack Compose 的桌面扩展，允许开发者用同样的声明式方式为 Windows、macOS 和 Linux 创建跨平台桌面应用程序。本系统采用 JetPack Compose（Compose for desktop）库进行用户界面构建。

4. **GPT-4 API**

   GPT-4 API 是由 OpenAI 提供的强大自然语言处理接口，允许开发者通过调用 OpenAI 的 GPT-4 模型来处理各种语言任务。它能够生成、理解和操作文本内容，支持聊天机器人、文本摘要、内容生成、代码解释等广泛的应用场景。本系统采用 GPT-4 API 库实现 GPT robot 功能。

5. **科大讯飞人脸比对-API**

   科大讯飞的人脸比对 API 是科大讯飞提供的一项人脸识别技术服务，允许开发者通过 API 接口实现高效、精准的人脸比对。该 API 广泛应用于身份验证、安全监控、门禁系统、金融支付等需要身份识别的场景。本系统采用科大讯飞人脸比对-API 库实现忘记密码时的人脸识别比对。

3 程序系统的结构

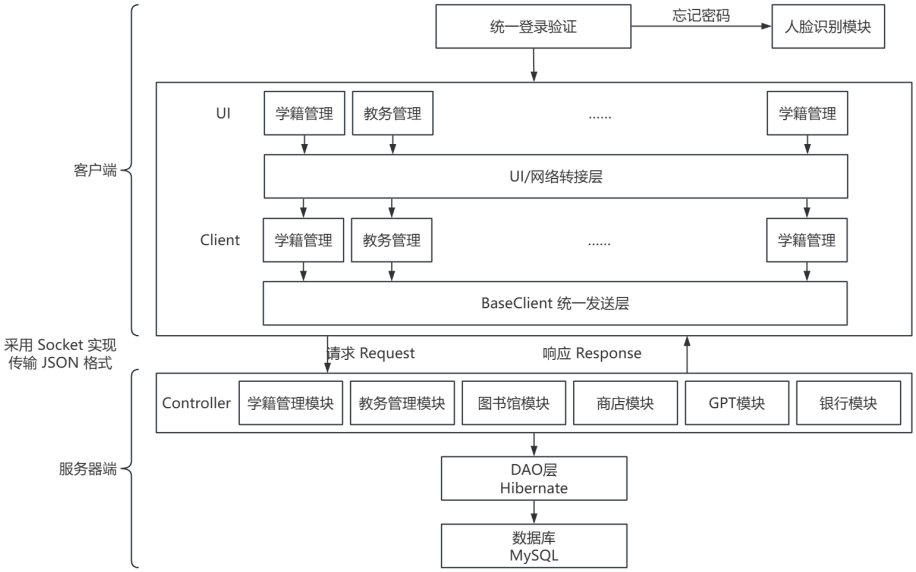


图 1: 程序系统的结构

4 用户管理模块设计说明

4.1 模块背景

统一认证系统，并根据用户身份提供其所能使用的功能

4.2 需求分析

登录：根据一卡通号和密码实行登录，并将登录信息保存传递至其它模块加以确定

管理：利用数据库对新用户的信息进行管理与设置，以及密码修改。

验证：通过用户输入的信息进行验证是否具有访问资格

忘记密码：忘记密码时，可以通过输入一卡通号，并进行人脸识别与人脸比对，确认身份后方可进行密码修改。

4.3 系统设计

4.3.1 界面设计

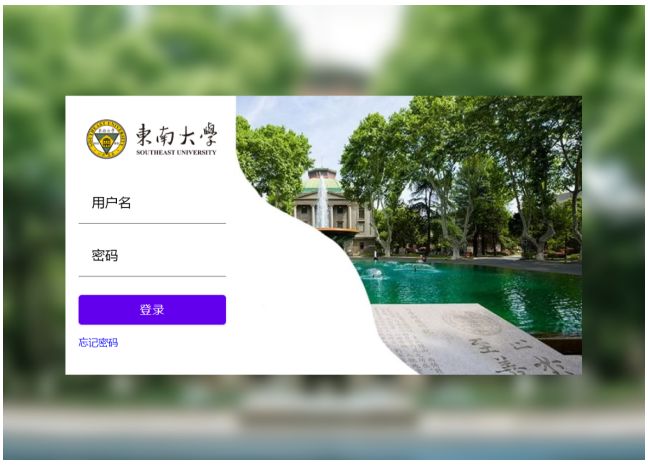


图 2: 登录页面

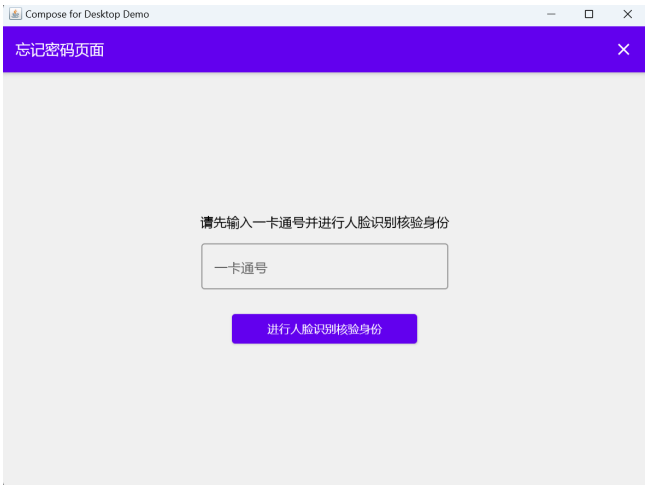


图 3: 忘记密码页面

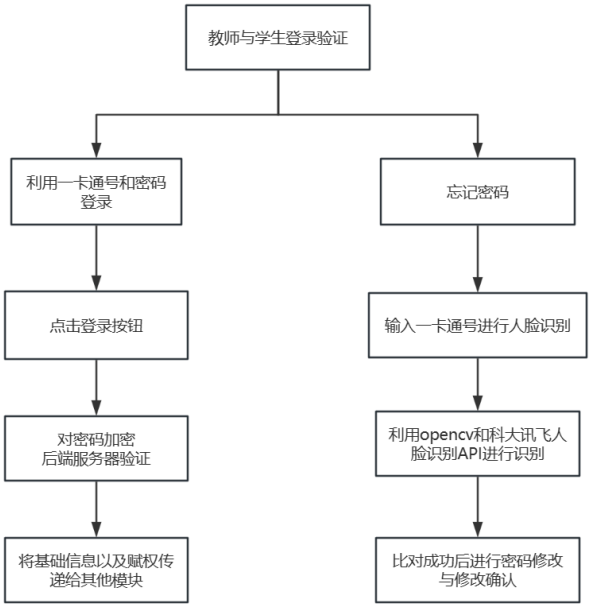


图 4: 登录页面

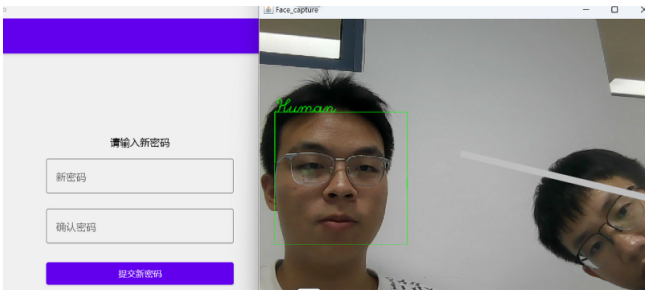


图 5: 登录页面

4.3.2 类分析

1. **实体类**

   User 类:

序号	名称	类型	约束	备注
1	cardNumber	Integer	9 位数字	一卡通号
2	password	String		登录密码
3	name	String		姓名
4	gender	enum	枚举类型	性别
5	phone	String		手机号
6	email	String		邮箱状态
7	role	String		登录角色

2. **服务类**

客户端：StudentClient:

序号	名称	方法	备注
1	登录	log_in	使用密码登录
2	登出	log_out	退出登录
3	忘记密码	ForgetPassword	修改密码

服务器端：server:

序号	名称	方法	备注
1	登录	log_in	使用密码登录
2	登出	log_out	退出登录
3	修改密码	change_password	确定修改密码

3. **人脸识别**

服务端：ImageToDatabase:

序号	名称	方法	备注
1	人脸比对	faceImageCompare	人脸存入数据库并返回比对结果

服务端：FaceCompare:

序号	名称	方法	备注
1	人脸识别	compareFace	调用科大讯飞人脸识别 API 对数据库中人脸图片进行比对

5 学籍管理模块设计说明

5.1 模块背景

用于管理学生、老师、管理员账号的相关信息。

5.2 需求分析

- **学生端：**学生可以通过该模块对个人学籍相关内容进行访问，比如姓名，籍贯，政治面貌，学籍状态，专业等。
- **教师端：**教师可以通过该模块对其个人信息相关内容进行访问，比如姓名，籍贯，政治面貌等
- **教务端：**教务管理人员可以通过该模块对全校学生的学籍信息进行管理，允许管理员通过搜索对每位学生学籍的相关内容进行访问修改与录入

5.3 系统设计

5.3.1 界面设计

图 6: 管理员教务页面

图 7: 办理入学页面

图 8: 办理退学页面

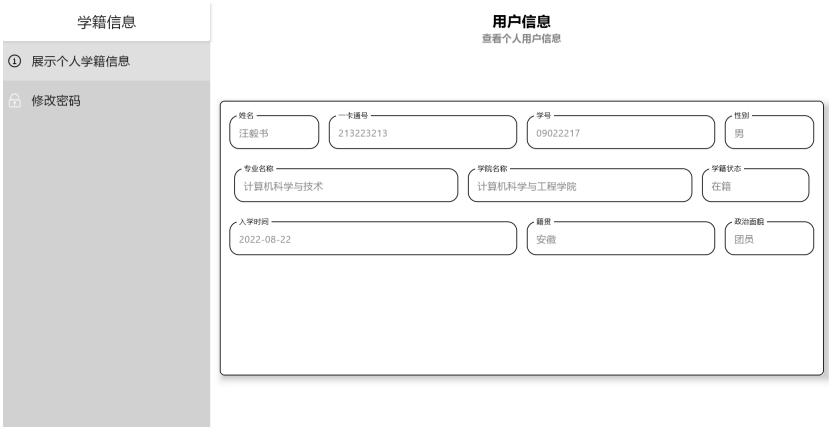
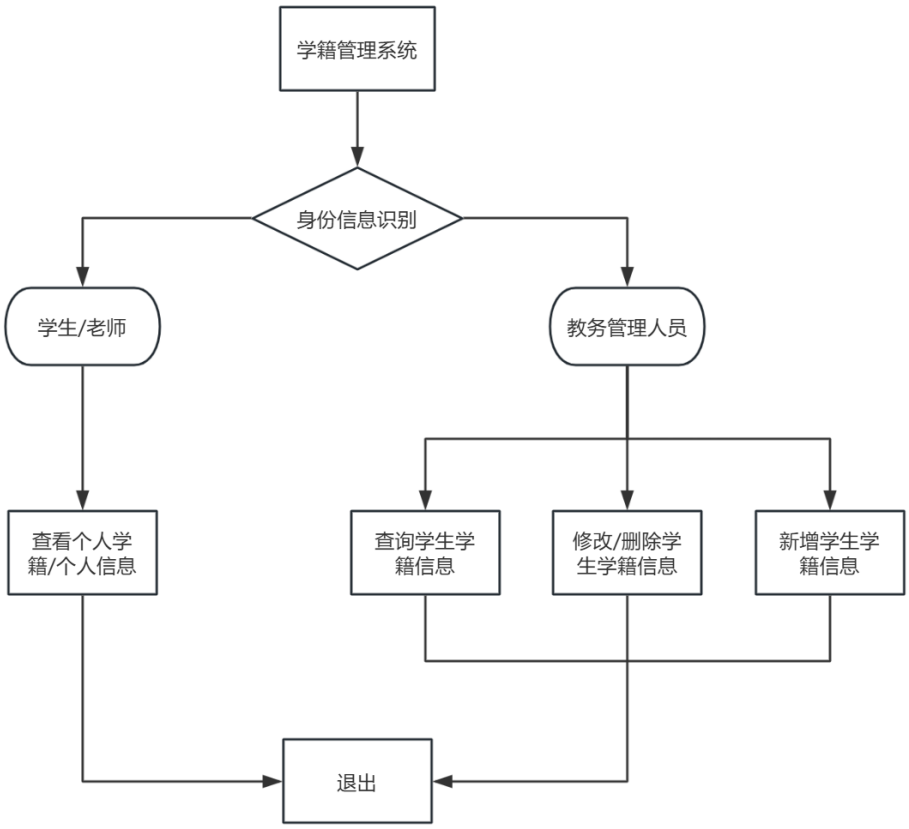


图 9: 学籍信息页面

5.3.2 模块流程图



5.3.3 类分析

1. **实体类**

   学生类：Student

序号	名称	类型	约束	备注
1	`name`	`String`		姓名
2	`cardNumber`	`Integer`	9 位数字	一卡通号
3	`studentNumber`	`String`	8 位数字	学号
4	`gender`	`enum`	枚举类型	性别
5	`major`	`String`		专业名称
6	`school`	`String`		学院名称
7	`studentStat`	`enum`	枚举类型	学籍状态
8	`enrollment`	`calendar`	YYYY-MM-DD	入学日期
9	`birthPlace`	`String`		籍贯
10	`politicalStat`	`enum`	枚举类型	政治面貌

2. **服务类**

   客户端：StudentStatusClient

序号	名称	方法	备注
1	搜索学籍信息	SearchInformation	搜索并展示所有学生信息
2	更新学籍信息	ChangeInformation	修改个人信息
3	办理入学	addInfomation	新增个人信息
4	办理退学	deleteInformation	删除学生信息
5	修改密码	changePassword	修改个人账号密码
6	查看个人信息	showInformation	查看个人的学习信息

   服务器端：StudentStatusController

序号	名称	方法	备注
1	接收请求查看学生信息	studentStatus_view	查看学生信息
2	接收请求修改学生信息	studentStatus_change	修改学生信息
3	接收请求搜索学生信息	studentStatus_search	搜索学生信息
4	接收请求添加学生信息	studentStatus_add	添加学生信息
5	接收请求删除学生信息	studentStatus_delete	删除学生信息
6	接收请求修改学生密码	studentStatus_changePassword	修改学生密码

6 教务管理模块设计说明

6.1 模块背景

此模块为实现教务管理相关功能。

6.2 需求分析

经过分析，该模块需要包括以下基本功能：

- **成绩查询：**学生用户可以通过该功能查询自己的考试成绩；
- **课程查询：**学生用户可以通过该功能查询自己已选课程；
- **课表查询：**学生用户可以通过该功能查询自己的课表；
- **评教管理：**学生用户可以通过该功能对课程以及上课教师进行评价；
- **排课功能：**管理员用户可以通过该功能增加课程以供选课；
- **成绩录入：**教务处管理者可以通过该功能上传成绩的 Excel 表格；
- **选课与退课：**学生用户可以通过该功能进行选课或者退课，根据容量限制人数；
- **查看评教：**管理员用户可以通过该功能查看各个老师的评教结果；
- **教学班信息：**教师用户可以通过该功能查看自己负责的课程与学生；

6.3 系统设计

6.3.1 界面设计

图 10: 教务主页面

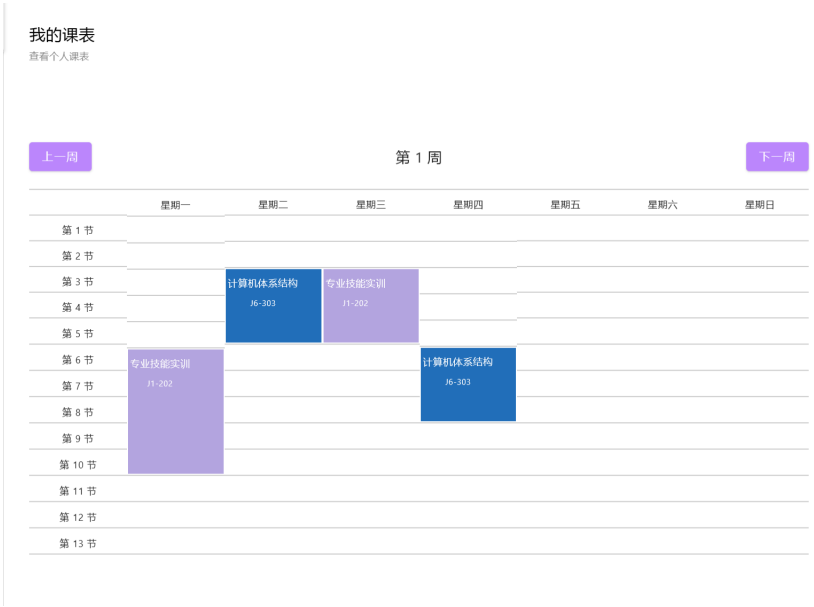


图 11: 我的课表页面

图 12: 我的成绩页面

图 13: 我的课堂页面

图 14: 评教页面

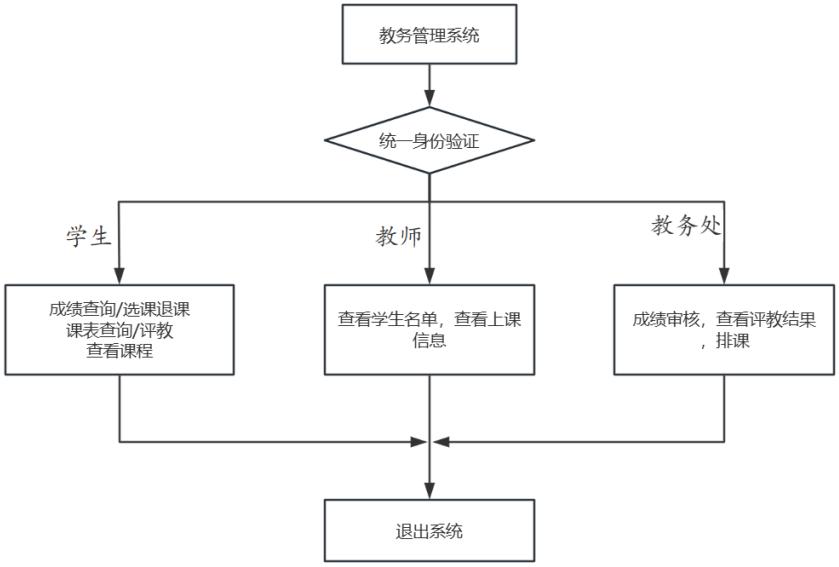
图 15: 选课页面

图 16: 评教页面

图 17: 导入成绩页面

图 18: 排入课程页面

6.3.2 模块流程图



6.3.3 类分析

1. **实体类**

   课程类：Course

序号	名称	类型	约束	备注
1	id	Integer		标识符
2	cardNumber	Integer	9 位数字	一卡通号
3	courseName	String		课程名
4	classroomName	String		教室
5	classDate	Integer		上课星期
6	classBegin	Integer		开始节次
7	classEnd	Integer		结束节次

教学班类：TeachingClass

序号	名称	类型	约束	备注
1	id	Integer		标识符
2	cardNumber	Integer		一卡通号
3	courseId	String		课程编号
4	courseName	String		课程名
5	credit	Double		学分
6	teacherName	String		任课教师
7	classroomName	String		上课地点
8	capacity	Integer		课容量
9	selectCount	Integer		选课人数
10	isSelected	Boolen		是否选课
11	weekRange	String		上课周数
12	dayOfWeek	List<String>		上课星期
13	courseBegin	List<Integer>		开始节次
14	courseEnd	List<Integer>		结束节次

排课教学班类：TeachingClassEnter

序号	名称	类型	约束	备注
1	id	Integer		标识符
2	courseId	String		课程编号
3	courseName	String		课程名
4	credit	Double		学分
5	teacherName	String		任课教师
6	classroomName	String		上课地点
7	capacity	Integer		课容量
8	weekRange	String		上课周数
9	dayOfWeek	List<String>		上课星期
10	courseBegin	List<Integer>		开始节次
11	courseEnd	List<Integer>		结束节次
11	selectCount	Integer		选课人数

成绩类：Grade

序号	名称	类型	约束	备注
1	id	Integer		标识符
2	cardNumber	Integer		一卡通号
3	courseName	String		课程名
4	credit	Double		学分
5	grade	Integer		成绩
6	regulargrade	Integer		平时分
7	midtermgrade	Integer		期中成绩
8	finalgrade	Integer		期末成绩

学生选课类：StudentSelectClass

序号	名称	类型	约束	备注
1	id	Integer		标识符
2	cardNumber	Integer		一卡通号
3	courseName	String		课程名
4	teacherName	String		任课教师
5	qqGroup	Integer		QQ 群
6	classroomName	String		教室
7	duration	String		持续时间
8	week	List<Integer>		上课星期
9	time	List<String>		具体时间
10	introduction	String		课程简介

评教类：EvaluationOfTeaching

序号	名称	类型	约束	备注
1	id	Integer		标识符
2	cardNumber	Integer		一卡通号
3	courseName	String		课程名
4	teacherName	String		任课教师
4	scores	List<Integer>		评教分数
4	isEvaluated	Boolean		是否评教

2. **服务类**

客户端：TeachingAffairsClient

序号	名称	方法	备注
1	查看我的课程	getClassInfo	获取教学班名称，上课地点，时间等
2	查成绩	getCoursesGrade	查看自己每一门课的成绩
3	查课表	send_dateInfo	查看自己的课表
3	发送评教信息	sendpingjiao_info	获取需要评教的课程
3	学生评教	sendInfo	获取需要评教的课程
4	选课	send_select_page	获取学生需要选择的课
4	选课	send_select_enter	学生选课
5	退课	dropClass	学生退课
6	教学班信息获取	getMyTeachingClasses	用于老师查看学生
7	学生信息获取	exportStudentList	导出学生名单
8	导出成绩信息	exportGradeTemplate	批量导出学生成绩
9	导入成绩	send_Entergrade	导入学生成绩
10	排课	send_schedule_Info	排入课程

服务器端：TeachingAffairsController

序号	名称	方法	备注
1	查看课程	getSelectedClasses	返回教学班名称，上课地点，时间等
2	查成绩	getGrade	返回学生成绩
3	查课表	classSchedule	返回课表
4	发送评教信息	sendEvaluation	返回需要评教的课程
5	学生评教	sendEvaluation	获取发回的评教分数
6	查看评教结果	teacherEvaluation	教务处获取评教结果
7	选课	studentSelection	获取学生选课信息
8	退课	studentDeseclection	获取学生退课信息
9	发送选课内容	sendSelection	用于显示学生已经选课页面
10	教务添加选课	addCourse	获取教务处的排课信息
11	发送选课信息	sendSelectionBefore	用于显示全新选课界面
12	发送待接收成绩课程	sendGradeCourse	用于显示需要导入成绩的课程
13	教师查看课程	teacherForCourse	返回老师的教学班信息

服务器端：ExcelGrade

序号	名称	方法	备注
1	导入成绩	`handleClientInput`	处理客户输入，包括操作和文件上传
2	保存文件	`saveBase64File`	
3	导入数据库	`processExcelFile`	

7 图书馆模块设计说明

7.1 模块背景

图书馆中图书信息的流通数据过于复杂庞大，依靠人工统计很难保证管理信息的效率和正确性。我们设计了一套图书信息管理系统，旨在帮助师生和管理员更加高效的实现图书的借阅、归还以及信息的获取与管理。

7.2 需求分析

我们需要实现以下功能，并提供调用对应功能的接口:

1. **书籍信息管理**
   图书馆模块需记录书籍的信息，包括书名、ISBN、作者、出版社、简介、借阅状态、数量、可借阅数量以及对应的书评信息，并存储在数据库中。

2. **用户信息管理**
   图书馆模块需记录师生用户的各种信息，包括用户名、一卡通号、借阅书籍信息及状态、剩余借阅数量、最大借阅数量，并存储在数据库中。

3. **借阅信息追踪**
   图书馆模块需要记录借阅信息，包括借出书名、借出书 ISBN 编号、借出人、借出人一卡通号、借阅时间、操作类型（借书、还书）、应还时间等，并存储在数据库中。

4. **讲座文体活动信息展示功能**
   图书馆模块需要展示实时展示最近一段时间的讲座以及文体活动，后端通过爬虫实时获取最新信息

5. **图书借阅功能**
   图书馆模块需分别为师生用户和管理员提供不同功能：

   - **对于师生用户：** 应能够查询书籍信息、发表书评；查看个人信息；提供相关接口实现借书、还书；续借已有书籍等功能并同步数据库状态。
   - **对于管理员：** 能够查看并管理书籍信息、查询借阅信息、获取所有用户信息并为用户提供借书、续借、归还服务。

6. **异常信息处理**
   当输入错误信息时应给予提示或解决方案，如信息格式错误、所借书籍不存在或无库存、用户逾期未换等等。

7.3 系统设计

7.3.1 界面设计

图 19: 图书馆主页页面

图 20: 我的图书页面

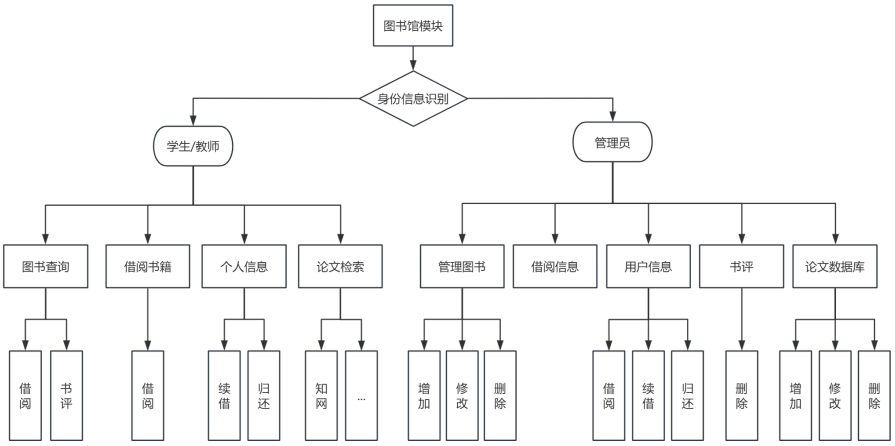
图 21: 修改图书页面

图 22: 添加图书页面

图 23: 查看借阅信息页面

图 24: 图书馆额外页面

7.3.2 模块流程图



7.3.3 类分析

1. **实体类**

用户信息类：LibraryUser

序号	名称	类型	约束	备注
1	userId	Integer	9 位数字	一卡通号
2	userName	String		姓名
3	userBorrowedNum	Integer	正整数	已借数量
4	userMaxBorrowNum	Integer	正整数	总可借数量
5	userStatu	enum		用户状态
6	userBorrowed	LinkedList	存放 LibraryTran	已借图书列表

图书信息类：LibraryBook

序号	名称	类型	约束	备注
1	bookId	UUID		书籍唯一标识符
2	bookName	String		图书名
3	bookIsbn	String	ISBN 格式	ISBN 编号
4	bookAuthor	String		作者
5	bookPress	String		出版社
6	bookDescription	String		书籍简介
7	bookPlace	String		藏书地点
8	bookStatus	enum		图书状态
9	bookCommand	LinkedList	存放 LibraryCom	书评列表

书评类：LibraryCom

序号	名称	类型	约束	备注
1	comId	UUID		书评唯一标识符
2	comTime	Date		评价时间
3	comUserId	Integer	9 位数字	评价人一卡通号
4	comUserName	String		评价人姓名
5	comText	String		评价内容
6	comBookId	UUID		评价图书唯一标识符

借阅信息类：LibraryTran

序号	名称	类型	约束	备注
1	tranId	UUID		借阅信息编号
2	tranBorrowTime	Date		借阅时间
3	tranDueTime	Date		到期时间
4	tranBookId	UUID		所借书籍的唯一标识符
5	tranBook	LibraryBook		所借书籍对象
6	tranUserId	Integer	9 位数字	借阅人一卡通号

2. **服务类**

客户端：LibraryClient

序号	名称	方法	备注
1	搜索图书	userSearch	用户搜索图书
2	评论	userCom	用户发表评价
3	借书	userBorrow	用户借书
4	续借	userRenew	用户续借
5	还书	userReturn	用户还书
6	添加书籍	adminAddBook	管理员添加书籍
7	修改书籍	adminChangeBook	管理员修改书籍
8	删除书籍	adminDeleteBook	管理员删除书籍
9	获取借阅列表	adminGetTran	管理员获取借阅列表
10	借书	adminBorrow	管理员借书
11	续借	adminRenew	管理员续借
12	还书	adminReturn	管理员还书
13	获得书评列表	adminGetCom	管理员获取书评列表
14	删除书评	adminDeleteCom	管理员删除书评
14	获得活动列表	getMessage	用户管理员获取活动列表

服务器端：LibraryController

序号	名称	方法	备注
1	搜索书籍	searchBook	响应用户传入信息进行模糊或精确搜索
2	返回用户信息	getUser	响应用户获取个人信息的请求
3	返回书籍信息	getBook	响应管理员获取书籍列表的请求
4	添加书籍	addBook	响应管理员添加书籍的请求
5	修改书籍	updateBook	响应管理员修改书籍的请求
6	删除书籍	removeBook	响应管理员删除书籍的请求
7	借书	borrowBooks	响应管理员或用户的借书请求
8	续借	renewBooks	响应管理员或用户的续借请求
9	还书	returnBooks	响应管理员或用户的还书请求
10	删除评论	remCom	响应管理员删除评论的请求
11	添加评论	addCom	响应用户发表书评的请求
12	爬取最新活动列表	getMes	响应用户获得最新活动列表的请求

8 商店模块设计说明

8.1 模块背景

如今线上交易平台非常成熟，为了便于学生进行线上购物，本系统模仿线上交易平台提供商品信息并可通过在线支付进行交易。

8.2 需求分析

用户类别 根据登录的用户角色（学生、老师或管理员），提供不同的功能访问权限。学生拥有浏览、购买商品的权限，管理员拥有修改商品信息、查看销售额等权限。

商品管理 商品信息由管理员录入，商品信息有名称、价格、库存、描述等。管理员可添加、编辑或删除商品。

商品推荐 通过用户的购物习惯推测用户可能想要购买的商品并将其推荐于主页。

购物操作 学生或老师在商店界面选择需要的商品，结算时显示所选商品和交易金额，之后进行线上支付。

交易统计　给管理员提供交易额统计，便于管理员为将来的销售做出修改。

界面设计　应参考现有电商平台，为用户提供一个简约、实用的线上交易平台界面。

安全性　涉及交易，应保证交易过程的安全，防止用户信息泄露和外部攻击。

异常处理　在用户执行非法操作时给出提醒，如试图购买已售罄的商品等，并提供可参考的解决方案。

8.3　系统设计

8.3.1　界面设计

选购界面

图 25: 商店主页页面

订单界面

图 26: 我的订单页面

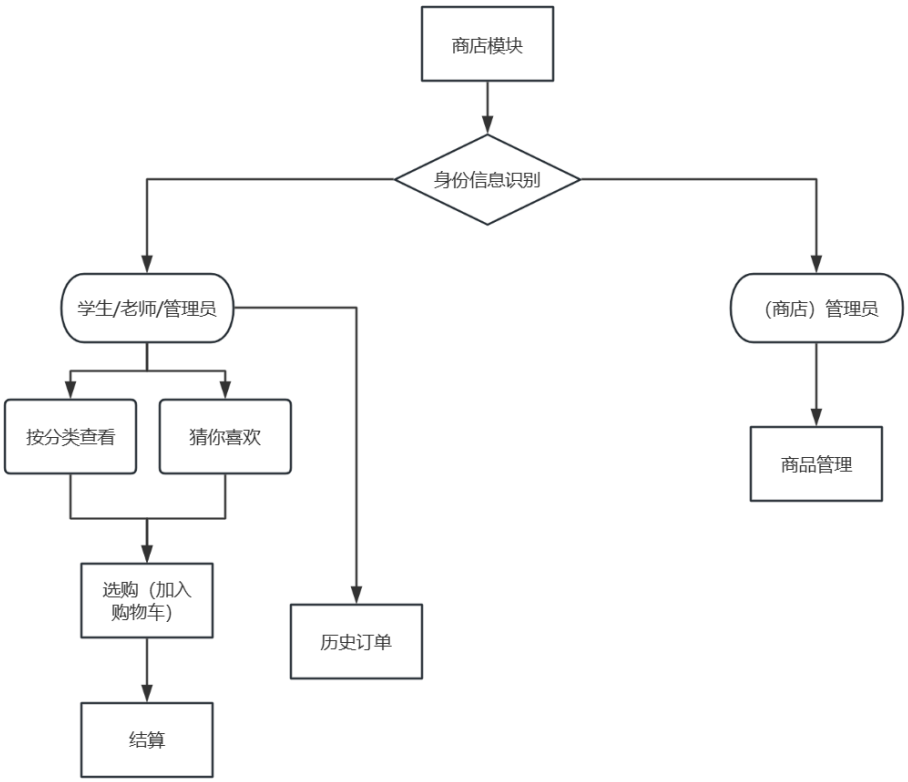
管理界面

图 27: 上架商品页面

图 28: 修改商品页面

图 29: 商品交易信息页面

8.3.2 模块流程图



8.3.3 类分析

1. **实体类**

   商品类：Product

序号	名称	类型	约束	备注
1	`productID`	`UUID`		商品编号
2	`name`	`String`		商品名
3	`price`	`Double`	≥ 0	商品价格
4	`pictureLink`	`String`		商品图片链接
5	`stock`	`Integer`	≥ 0	商品库存数量
6	`sales`	`Integer`	≥ 0	商品销量
7	`description`	`String`		商品信息描述
8	`select`	`String`		商品类别

商品交易记录类：ProductTransactionRecord

序号	名称	类型	约束	备注
1	transactionID	UUID		交易编号
2	productID	UUID		交易中的商品编号
3	name	String		交易中的商品名称
4	productPrice	Integer	≥ 0	交易中的商品单价
5	productAmount	Integer	≥ 0	交易中的商品数量
6	cardNumber	integer		交易中的商品编号
7	time	LocalDateTime		交易时间
8	cardNumber	Integer	9 位数字	交易关联一卡通号
9	product	Product		交易中的商品

2. **服务类**

客户端：ShopClient

序号	名称	方法	备注
1	查找商品	searchProduct	查找商品
2	支付	pay	支付结算金额
3	获取商品信息	getAllProduct	获取数据库中所有商品信息
4	获取交易记录	getAllTransaction	获取数据库中所有商品交易信息
5	获取销量	getSales	获取商品累计销量
6	根据 ID 查找	searchID	查找某 ID 的商品
7	查找交易	serachTransaction	查找含有关键词的交易记录
8	获取交易记录	getTransaction	获取按照日期排序的交易记录
9	创建交易	createTransaction	创建新的交易记录
10	新增商品	addProduct	（管理员）添加新的商品
11	删除商品	deleteProduct	（管理员）删除商品
12	修改商品信息	modifyProduct	（管理员）修改商品信息（价格、库存等）
13	查看报告	getReport	管理员查看销售情况

服务器端：ShopController

序号	名称	方法	备注
1	猜你喜欢	store_guess	根据历史购买记录推荐商品
2	商店展示商品	store_show	根据选择分类不同进行对应的商品展示
3	获取我的订单	store_getMyTransaction	获取并返回数据库中对应用户的订单
4	购买商品	store_buygoods	进行商品购买
5	增加商品	store_addProduct	管理员增加商品
6	查找商品	store_search	查找并返回所搜索的商品（模糊搜索）
7	修改商品	store_change	管理员修改商品
8	删除商品	store_delete	管理员删除商品
9	获取所有订单	store_getAllTransaction	管理员获取所有订单

9 其他选做模块

9.1 银行模块说明

在虚拟校园系统中加入银行系统，该系统作为商店模块的补充，为账户余额进行模块化管理，并留出充值和提现接口，使得商店系统能够更加真实完善。

9.1.1 界面设计

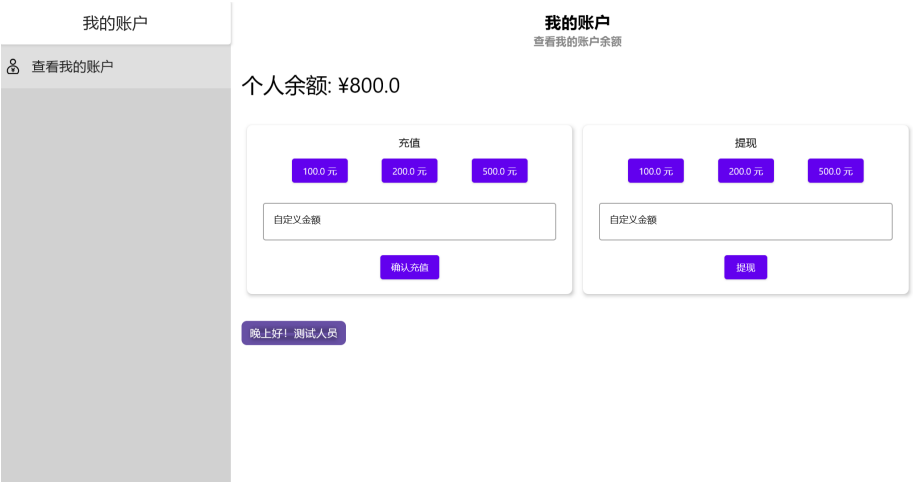
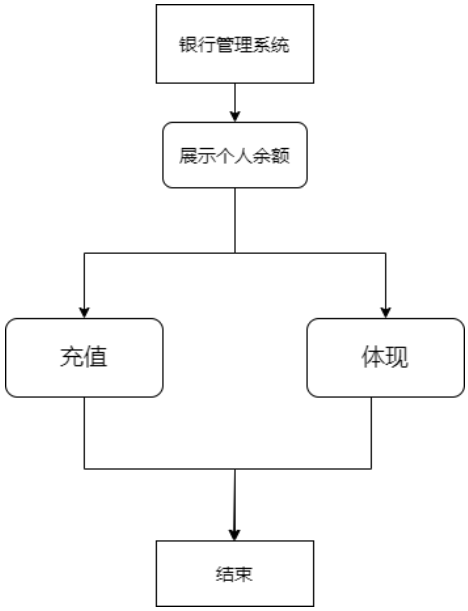


图 30: 银行页面

9.1.2 模块流程图



9.1.3 类分析

1. **实体类**

   银行类：Balance

序号	名称	类型	约束	备注
1	user	String		显示欢迎界面的用户名称
2	balance	Double	不能为负数	个人账户余额

2. **服务类**

   客户端：BalanceClient

序号	名称	方法	备注
1	展示个人余额	ViewBalance	
2	充值	Recharge	充值接口
3	提现	withdraw	提现接口

9.2 GPT 内联

在虚拟校园系统中加入 GPT-4 API 接口可以为学生、教师和管理人员提供智能化的互动和支持。通过加入 GPT，虚拟校园系统能够更加智能化、个性化，并为全体师生提供更高效的学习体验。

10 公共模块设计说明

10.1 消息信息模型

10.1.1 Request

名称	类型	备注
uri	String	请求资源唯一标识符
params	Map<String, String>	请求参数
session	Session	由服务端维护的 Session 状态

10.1.2 Response

名称	类型	备注
status	String	响应状态
message	String	响应信息
data	Object	响应体
session	Session	对服务端 Session 状态更新

11 网络模块设计说明

在本项目中，我们采用了 Socket 通信技术实现了服务器与客户端的连接。通过使用 Socket 创建 TCP 连接，客户端能够成功连接到服务器，实现了数据的稳定传输。为了确保传输过程中数据的安全性，我们对用户密码进行了 MD5 加密处理。这一措施有效保护了密码信息，防止未经授权的访问和数据泄露，提高了系统的安全性。

同时，我们使用了 JSON 格式来实现变量的序列化，并以 Json 文件的格式实现服务器和客户端的数据传输，这使得数据在传输过程中保持了结构

化和可读性。在客户端，我们采用了对 Kotlin 更加友好的 moshi 框架进行的序列化和反序列化；在服务器端，我们使用了基于 Java 语言的 JsonObject 实现相同的操作。通过 JSON 序列化，变量的转换与解析变得更加高效和灵活，提升了数据交换的便利性和系统的整体性能。

11.0.1 服务器端 ServerSocket

名称	说明	备注
`ServerSocket(port)`	初始化相关配置	传入端口
`serverSocket.accept()`	接收客户端相应	

```
try (ServerSocket serverSocket = new ServerSocket(4444)) {
            System.out.println("服务器已启动，等待连接...");

            while (true) {  // 循环等待客户端连接
                Socket socket = serverSocket.accept();
                System.out.println("客户端已连接");

                // 为每个客户端创建一个新的线程来处理请求
                new ClientHandler(socket, functionMap).start();
            }

        } catch (IOException e) {
            e.printStackTrace();
        }
```

12 多线程模块设计说明

在 Socket 中使用内置的线程池进行自动的多线程处理，以应对多个客户同时发出请求的情况。

12.1 Socket

12.1.1 客户端

12.1.2 服务器端 Server

12.2 输入输出流

12.2.1 读取输入

借助 Netty 提供的 ChannelPipeline 和 ChannelInboundHandlerAdapter 等工具，可方便地定义流的处理方式。

```
EventLoopGroup bossGroup = new NioEventLoopGroup();
EventLoopGroup workerGroup = new NioEventLoopGroup();
try {
    ServerBootstrap b = new ServerBootstrap();
    b.group(bossGroup, workerGroup)
            .channel(NioServerSocketChannel.class)
            .childHandler(new ChannelInitializer<
               SocketChannel>() {
                @Override
                public void initChannel(@NonNull
                   SocketChannel ch) {
                    ch.pipeline().addLast(new
                       JsonObjectDecoder()).addLast(
                       new NettyHandler(router));
                }
            })
            .option(ChannelOption.SO_BACKLOG, 128)
            .childOption(ChannelOption.SO_KEEPALIVE,
               true);
```

```
    ChannelFuture f = b.bind(port).sync();
    f.channel().closeFuture().sync();
} finally {
    workerGroup.shutdownGracefully();
    bossGroup.shutdownGracefully();
}
```

```
@Override
public void channelRead(@NonNull ChannelHandlerContext ctx, @NonNull Object msg) {
    ByteBuf in = (ByteBuf) msg;
    try {
        log.info("[{}] Received: {}", ctx.channel().id(), in.toString(CharsetUtil.UTF_8));
        Request request = gson.fromJson(in.toString(CharsetUtil.UTF_8), Request.class);
        request.setSession(session);
        if (!router.hasRoute(request.getUri())) {
            log.info("[{}] Route not found: {}", ctx.channel().id(), request.getUri());
            sendResponse(ctx, Response.Common.notFound());
            return;
        } else if (!session.permission(router.getRole(request.getUri()))) {
            log.info("[{}] Permission denied: {}", ctx.channel().id(), request.getUri());
            sendResponse(ctx, Response.Common.permissionDenied());
            return;
        }
        Response response = router.invoke(request);
        if (response.getSession() != null) {
            log.info("[{}] Session updated: {}", ctx.channel().id(), response.getSession());
            session = response.getSession();
        }
        sendResponse(ctx, response);
    } catch (Exception e) {
        log.error("[{}] Exception: {}", ctx.channel().
```

```
            id(), e.getMessage());
    } finally {
        ReferenceCountUtil.release(msg);
    }
}
```

12.2.2 发送输出

```
private void sendResponse(ChannelHandlerContext ctx,
    Response response) {
     ctx.writeAndFlush(Unpooled.copiedBuffer(gson.
        toJson(response), CharsetUtil.UTF_8));
}
```

13 数据库设计说明

13.1 元数据

共有 16 个数据库，名称以及说明如下表：

名称	说明
user	存储用户信息
student	存储学生学籍信息
course	存储课程信息
evaluationofteaching	存储评教信息
grade	存储成绩信息
studentselectclass	存储学生选择教学班信息
teachingclass	存储教师授课信息
teachingclassenter	存储学生选课信息
librarybook	存储图书馆馆藏书目信息
librarycom	存储图书馆图书评论
librarytran	存储图书馆借还书记录
libraryuser	存储图书馆使用人员信息
product	存储商店中商品信息
producttransactionrecord	存储商店交易记录
image_from_client	存储用户人脸图像（用于图像识别）
images_table	存储用户人脸原始图像（用于作图像识别的基准图像）

13.2 表设计

13.2.1 user 表

user 表用于存储所有用户的信息，包括学生、教职员工等。

字段	类型	约束	说明
cardNumber	INT	9 位数字	一卡通号
password	VARCHAR		密码
name	VARCHAR		姓名
gender	ENUM		性别
phone	VARCHAR		手机号
email	VARCHAR		邮箱
balance	DOUBLE		账户余额
role	ENUM		权限角色

13.2.2 student 表

student 表用于存储学生学籍信息。

字段	类型	约束	说明
cardNumber	INT	9 位数字	一卡通号
password	VARCHAR		密码
name	VARCHAR		姓名
studentNumber	VARCHAR		学号
gender	ENUM		性别
major	VARCHAR		专业
school	VARCHAR		学院
studentStat	ENUM		学籍状态
enrollment	DATA		入学日期
birthPlace	VARCHAR		生日
politicalStat	ENUM		政治面貌

13.2.3 image_from_client 表

image_from_client 表用于存储用户人脸图片信息。

字段	类型	约束	说明
id	INT		表内唯一标识信息，用作主键
cardNumber	INT		一卡通号
image	LONGBLOB		图片

13.2.4 image_table 表

image_table 表用于提前存储用户人脸图片信息以对比。

字段	类型	约束	说明
id	INT		表内唯一标识信息，用作主键
cardNumber	INT		一卡通号
image_data	LONGBLOB		图片

13.2.5 course 表

course 表用于存储课程信息，在课表显示

字段	类型	约束	说明
id	INT		标识符
cardNumber	INT		一卡通号
courseName	VARCHAR		课程名
classroomName	VARCHAR		教室
classDate	INT		上课星期
classBegin	INT		开始节次
classEnd	INT		结束节次

13.2.6 evaluationofteaching 表

evaluationofteaching 表用于存储评教信息。

字段	类型	约束	说明
id	INT		标识符
cardNumber	INT		一卡通号
courseName	VARCHAR		课程名
teacherName	VARCHAR		任课教师
scores	TEXT		评教分数
isEvaluated	Boolean		是否评教

13.2.7 grade 表

grade 表用于存储成绩信息。

字段	类型	约束	说明
id	INT		标识符
cardNumber	INT		一卡通号
courseName	VARCHAR		课程名
credit	Double		学分
grade	INT		成绩
regulargrade	INT		平时分
midtermgrade	INT		期中成绩
finalgrade	INT		期末成绩

13.2.8 studentselectclass 表

studentselectclass 表用于存储学生评教信息。

字段	类型	约束	说明
id	INT		标识符
cardNumber	INT		一卡通号
courseName	VARCHAR		课程名
teacherName	VARCHAR		任课教师
qqGroup	INT		QQ 群
classroomName	VARCHAR		教室
duration	VARCHAR		持续时间
week	TEXT		上课星期
time	TEXT		具体时间
introduction	TEXT		课程简介

13.2.9 teachingclass 表

teachingclass 表用于存储教学班信息。

字段	类型	约束	说明
id	INT		标识符
cardNumber	INT		一卡通号
courseId	VARCHAR		课程编号
courseName	VARCHAR		课程名
credit	Double		学分
teacherName	VARCHAR		任课教师
classroomName	VARCHAR		上课地点
capacity	INT		课容量
selectCount	INT		选课人数
isSelected	Boolen		是否选课
weekRange	VARCHAR		上课周数
dayOfWeek	TEXT		上课星期
courseBegin	TEXT		开始节次
courseEnd	TEXT		结束节次

13.2.10 teachingclassenter 表

teachingclassenter 表用于存储教学班排课信息。

字段	类型	约束	说明
id	INT		标识符
courseId	VARCHAR		课程编号
courseName	VARCHAR		课程名
credit	Double		学分
teacherName	VARCHAR		任课教师
classroomName	VARCHAR		上课地点
capacity	INT		课容量
weekRange	VARCHAR		上课周数
dayOfWeek	TEXT		上课星期
courseBegin	TEXT		开始节次
courseEnd	TEXT		结束节次
selectCount	INT		选课人数

13.2.11 librarybook 表

librarybook 表用于存储图书馆图书的基本信息。

字段	类型	约束	说明
bookId	VARCHAR		表内唯一标识信息，用作主键
bookName	VARCHAR		书名
bookIsbn	VARCHAR		书籍 ISBN 码
bookAuthor	VARCHAR		作者
bookPress	VARCHAR		出版社
bookDescription	LONGBLOB		简介
bookPlace	VARCHAR		位置
bookStatus	enum		借阅状态
bookPhoto	VARCHAR		图片

13.2.12 librarycom 表

librarycom 表用于存储图书馆图书的书评信息。

字段	类型	约束	说明
comId	VARCHAR		表内唯一标识信息，用作主键
comTime	DATE		书评发布时间
comUserId	INT		书评发布人的 id
comUserName	VARCHAR		书评发布人的姓名
comText	VARCHAR		书评内容
comIsbn	LONGBLOB		评价书籍的 Isbn

13.2.13 librarytran 表

librarytran 表用于存储图书馆图书的借阅信息。

字段	类型	约束	说明
tranId	VARCHAR		表内唯一标识信息，用作主键
tranBorrowedTime	DATE		借阅时间
tranDueTime	DATE		应还时间
tranBookId	VARCHAR		借阅书籍的 id
tranUserId	VARCHAR		借阅人的 id
tranIsbn	VARCHAR		借阅书籍的 Isbn

13.2.14 libraryuser 表

libraryuser 表用于存储图书馆的用户信息。

字段	类型	约束	说明
userId	VARCHAR		表内唯一标识信息，用作主键
userName	DATE		用户名称
userBorrowedNum	DATE		已借阅数量
userMaxBorrorwNum	VARCHAR		最大借阅数量
userStatus	VARCHAR		用户状态

13.2.15 product 表

product 表用于存储商店交易信息。

字段	类型	约束	说明
productID	INT		表内唯一标识信息，用作主键
name	VARCHAR		商品名称
price	DOUBLE		商品交易时单价
pictureLink	VARCHAR		商品图片链接
stock	INT		商品库存
sales	INT		商品销量
description	VARCHAR		商品描述
select	VARCHAR		商品分类

13.2.16 producttransactionrecord 表

producttransactionrecord 表用于存储商店交易信息。

字段	类型	约束	说明
transactionID	INT		表内唯一标识信息，用作主键
productID	INT		交易中商品 ID
name	VARCHAR		交易中商品名称
productPrice	DOUBLE		商品交易时单价
productAmount	DOUBLE		交易中商品数量
cardNumber	INT	9 位数字	交易关联一卡通号
time	DATE		交易时间
remark	VARCHAR		交易中商品描述
select	VARCHAR		交易中商品分类