

# 虚拟校园系统设计说明书

09021227 金 桥  
09021218 马晓龙  
09021134 王智东  
09021130 张博雅  
09021202 钟世贵  
09021215 曹江宁

版本：1.0.0-draft

日期：2023 年 8 月 27 日

# 目录

<b>1 引言</b>	<b>2</b>
1.1 编写目的 . . . . .	2
1.2 背景 . . . . .	2
<b>2 程序系统的分析</b>	<b>2</b>
2.1 可行性分析 . . . . .	2
2.2 需求分析 . . . . .	2
2.3 开发设计环境 . . . . .	3
2.3.1 IntelliJ IDEA . . . . .	3
2.3.2 JDK 17 . . . . .	3
2.3.3 MySQL 8.0 . . . . .	3
2.3.4 第三方库 . . . . .	3
<b>3 程序系统的结构</b>	<b>4</b>
<b>4 用户管理模块设计说明</b>	<b>4</b>
4.1 模块背景 . . . . .	4
4.2 需求分析 . . . . .	4
4.3 系统设计 . . . . .	5
4.3.1 界面设计 . . . . .	5
4.3.2 模块流程图 . . . . .	5
4.3.3 类分析 . . . . .	6
<b>5 学籍管理模块设计说明</b>	<b>6</b>
5.1 模块背景 . . . . .	6
5.2 需求分析 . . . . .	6
5.3 系统设计 . . . . .	7
5.3.1 界面设计 . . . . .	7
5.3.2 模块流程图 . . . . .	8
5.3.3 类分析 . . . . .	8
<b>6 教务管理模块设计说明</b>	<b>9</b>
6.1 模块背景 . . . . .	9
6.2 需求分析 . . . . .	9
6.3 系统设计 . . . . .	10
6.3.1 界面设计 . . . . .	10
6.3.2 模块流程图 . . . . .	11
6.3.3 类分析 . . . . .	12
<b>7 图书馆模块设计说明</b>	<b>13</b>
7.1 模块背景 . . . . .	13
7.2 需求分析 . . . . .	13
7.3 系统设计 . . . . .	14

7.3.1 界面设计 . . . . .	14
7.3.2 模块流程图 . . . . .	15
7.3.3 类分析 . . . . .	16
<b>8 商店模块设计说明</b>	<b>17</b>
8.1 模块背景 . . . . .	17
8.2 需求分析 . . . . .	17
8.3 系统设计 . . . . .	18
8.3.1 界面设计 . . . . .	18
8.3.2 模块流程图 . . . . .	19
8.3.3 类分析 . . . . .	20
<b>9 TODO: 其他选做模块</b>	<b>21</b>
<b>10 公共模块设计说明</b>	<b>21</b>
10.1 消息信息模型 . . . . .	21
10.1.1 Request . . . . .	21
10.1.2 Response . . . . .	21
10.2 用户信息模型 . . . . .	21
10.3 工具类 . . . . .	21
10.3.1 数据库工具类 . . . . .	21
10.3.2 密码工具类 . . . . .	22
<b>11 网络模块设计说明</b>	<b>22</b>
11.1 Socket . . . . .	22
11.1.1 客户端 . . . . .	22
11.1.2 服务器端 NettyServer . . . . .	22
11.2 输入输出流 . . . . .	22
11.2.1 读取输入 . . . . .	22
11.2.2 发送输出 . . . . .	24
<b>12 多线程模块设计说明</b>	<b>24</b>
<b>13 数据库设计说明</b>	<b>24</b>
13.1 元数据 . . . . .	24
13.2 表设计 . . . . .	25
13.2.1 user 表 . . . . .	25
13.2.2 student 表 . . . . .	25
13.2.3 major 表 . . . . .	25
13.2.4 school 表 . . . . .	25
13.2.5 accounting 表 . . . . .	26
13.2.6 card_transaction 表 . . . . .	26
13.2.7 course 表 . . . . .	26
13.2.8 class 表 . . . . .	26
13.2.9 selected_class 表 . . . . .	27

13.2.10 library_book 表 . . . . .	27
13.2.11 library_user 表 . . . . .	27
13.2.12 library_transaction 表 . . . . .	27
13.2.13 store_item 表 . . . . .	28
13.2.14 store_transaction 表 . . . . .	28
<b>14 其他</b>	<b>28</b>

# 1 引言

## 1.1 编写目的

本系统为东南大学计算机科学与技术专业 2021 级暑期学校专业技能实训课程内容。为了更清晰的描述本系统的功能以及内部运行逻辑，为代码配套编写此文档，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

本系统为校园系统，采用 Java 语言进行编写。分为服务器端与客户端：

**服务器端** 负责运行数据库管理系统并处理来自客户端的请求并返回信息。

**客户端** 运行在用户的计算机上，提供图形用户界面并将用户的请求发送到服务器端。同时处理服务器返回的信息并进行显示。

# 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，了解到目前学校的办事大厅整体系统结构。本系统开发的主要目的是实现校园系统的网络化，方便学校管理以及自助业务办理。

## 2.2 需求分析

项目必做部分包括以下内容：

**用户管理** 为了实现对于用户的统一管理，我们设计了统一登录验证模块。实现对于用户的管理、登录、鉴权。

**学生学籍管理** 为了实现学生学籍管理，我们设计了学籍管理模块，支持学生修改个人学籍信息以及教务员审核。

**选课系统** 为了实现选课系统，我们设计了教务管理模块，支持学生选课、教师导出名单、教务员修改课程排课等功能。

**图书馆** 参照东南大学图书馆，我们设计了图书馆管理模块，支持预约借还书、查询书籍。并支持工作人员修改书籍信息以及添加新书。

**商店** 为了实现商店功能，我们设计了商店模块，支持用户选购商品并线上结算支付。同时支持工作人员导入货物，修改信息等。

## 2.3 开发设计环境

### 2.3.1 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.

### 2.3.2 JDK 17

Java 17 是 Java 开发平台最新推出的 LTS 版本。Java 17 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 17 作为开发平台。

### 2.3.3 MySQL 8.0

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 8.0 作为数据库管理系统。

### 2.3.4 第三方库

**Netty** Netty 是一个非阻塞 I/O 客户端-服务器框架，主要用于开发 Java 网络应用程序，如协议服务器和客户端。本项目采用 Netty 实现 Socket.

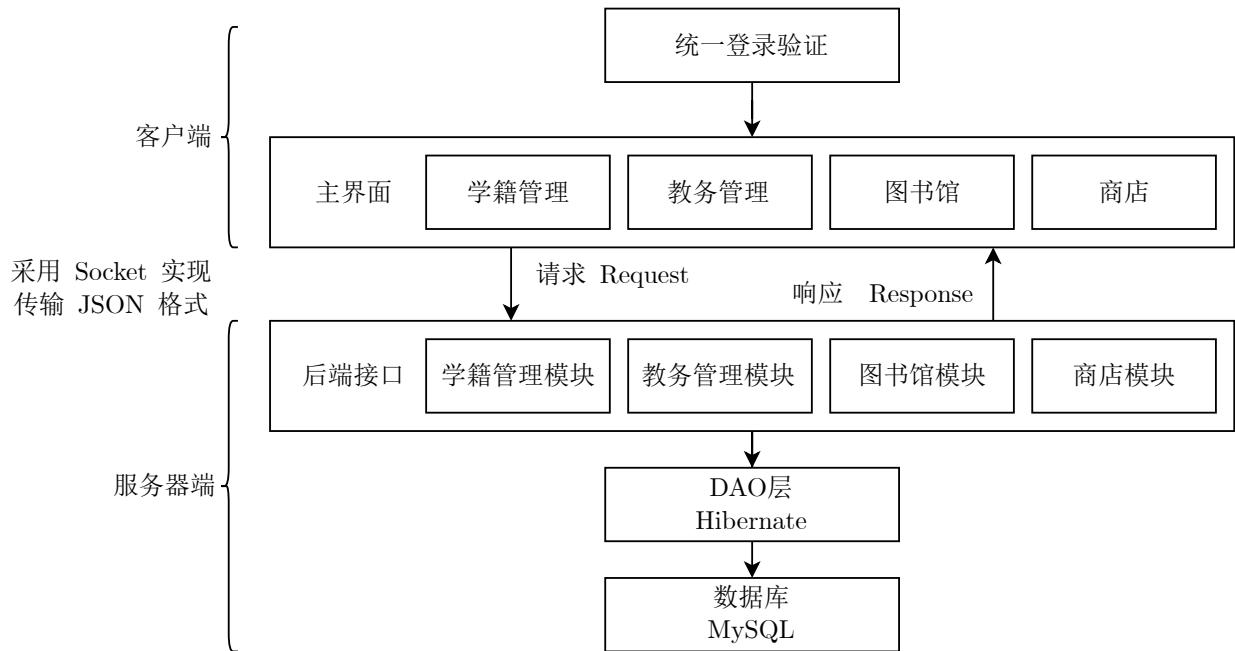
**Gson** Gson 是 Google 公司发布的一个开放源代码的 Java 库，主要用途为序列化 Java 对象为 JSON 字符串，或反序列化 JSON 字符串成 Java 对象。本项目采用 Gson 实现 JSON 格式化。

**Slf4j 与 Logback** Slf4J 是一个用于日志系统的简单 Facade，允许最终用户在部署其应用时使用其所希望的日志系统。Logback 是一个 Java 日志框架，是 log4j 项目的继承者，性能比 log4j 要好。本项目采用这两者作为日志系统。

**Hibernate** Hibernate 是一种 ORM 框架，全称为 Object-Relative DataBase-Mapping，在 Java 对象与关系数据库之间建立某种映射，以实现直接存取 Java 对象。本项目采用 Hibernate 作为数据库映射。

**Jetpack Compose** Jetpack Compose 是 Google 针对 Android 推出的新一代声明式 UI 工具包，完全基于 Kotlin 打造，具备了跨平台的使用基础。本项目采用 Jetpack Compose 实现 UI 界面。

### 3 程序系统的结构



### 4 用户管理模块设计说明

#### 4.1 模块背景

统一的认证、鉴权与会话管理系统。

#### 4.2 需求分析

**登录** 实现基本的登录功能，并存储会话信息，将登录用户信息传递给其他功能模块。

**管理** 实现用户管理功能，允许管理员便捷地管理用户，同时为用户提供部分自助功能。

**鉴权** 配合功能模块对请求进行鉴权，根据用户身份确定其所能访问的功能，阻断非法访问请求。

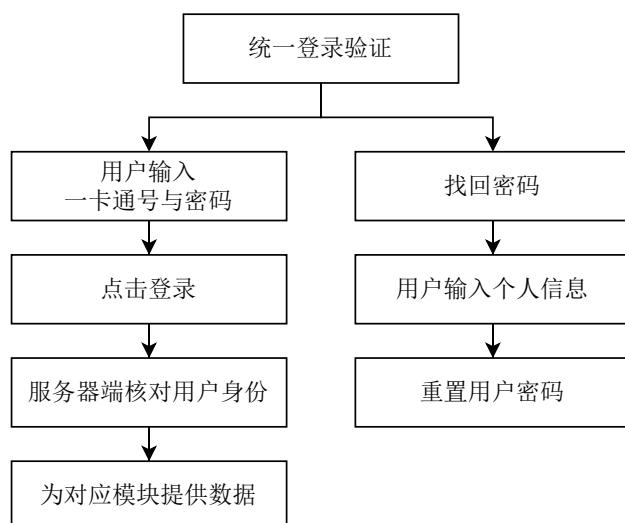
## 4.3 系统设计

### 4.3.1 界面设计



统一登录验证界面

### 4.3.2 模块流程图



### 4.3.3 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		处理后的密码
3	name	Integer		姓名
4	gender	enum		性别
5	phone	String		手机号
6	email	String		邮箱
7	role	String		权限角色

#### 2. 服务类

客户端: AuthClient

序号	名称	方法	备注
1	登录	login	使用密码登录
2	登出	logout	退出登录
3	重置密码	resetPassword	验证后重置密码

服务器端: AuthController

序号	名称	方法	备注
1	登录	login	使用密码登录
2	登出	logout	退出登录
3	重置密码	resetPassword	验证后重置密码

## 5 学籍管理模块设计说明

### 5.1 模块背景

用于管理学生的学籍相关信息。

### 5.2 需求分析

**学生端** 学生可以通过该模块查看、更新、填写自己的学籍信息相关内容，比如一卡通号、学号、专业编号及名称、学院编号及名称、学籍状态、入学时间等。

**教务端** 教务管理人员可以通过该模块审核、管理、修改相关学生的学籍信息。在学生首次填写及后续更改自己的学籍信息后，教务负责审核该信息是否合法、正确。并且可以根据学院相关规章、要求对学生的信息进行后台更改。



## 5.3 系统设计

### 5.3.1 界面设计

个人学籍信息

查看个人学籍信息

姓	王	名	小明	性别	男	出生日期	2000年1月1日
专业	计算机科学与技术			学院	计算机科学与工程学院		
一卡通号				学号			
一卡通号				学号			

+ 提交审核

学生端操作界面

仪表盘

管理中心

学号	待审批	姓名	一卡通号	生日
		王	小明	2000年1月1日
		李	小红	2000年1月1日
		张	小强	2000年1月1日
		赵	小芳	2000年1月1日

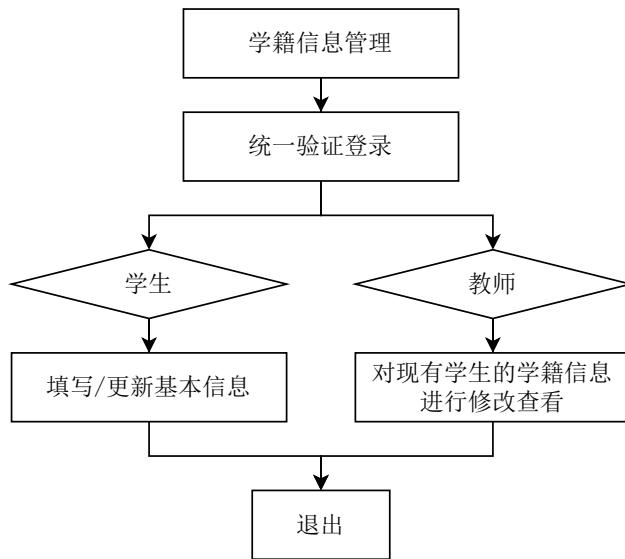
修改学籍信息

添加/修改/删除

点击进入

教务端操作界面

### 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	cardNumber	Integer	9位数字	一卡通号
2	studentNumber	String	8位数字	学号
3	major	Integer		专业编号用于索引
4	school	Integer		学院编号用于索引
5	status	enum		学籍状态(在读/离职等)
6	enrollment	calendar		入学日期(年/月/日)
7	birthPlace	String		籍贯
8	politicalStat	enum		政治面貌

专业类(存放专业编号及名称): Major

序号	名称	类型	约束	备注
1	majorNum	Integer		专业编号
2	majorName	String		专业名称

学院类(存放学院编号及名称): School

序号	名称	类型	约束	备注
1	schoolNum	Integer		学院编号
2	schoolName	String		学院名称

## 2. 服务类

客户端: StudentStatusClient

序号	名称	方法	备注
1	发送	stuInfoFetch	发 request
2	添加	requestInfo	发送个人信息
3	更新	updateInfo	更新个人信息
4	修改	modifyInfo	修改个人信息
5	保存	saveInfo	保存个人信息

服务器端: StudentStatusController

序号	名称	方法	备注
1	添加个人信息	addInfoHandle	添加个人信息
2	删除个人信息	deleteInfoHandle	删除个人信息
3	更新个人信息	updateInfoHandle	更新个人信息
4	暂存个人信息	saveInfoHandle	暂存个人信息
5	修改个人信息	modifyInfoHandle	修改个人信息

## 6 教务管理模块设计说明

### 6.1 模块背景

用于管理教务的相关信息。

### 6.2 需求分析

**课程管理功能** 包括课程创建与编辑、课程计划以及课程信息发布功能。

- **课程创建与编辑:** 教务管理员能够创建新的课程，指定课程名称、课程编号、学分、授课老师、开设学院等信息，并可以编辑课程信息。
- **课程计划:** 管理员可以制定每学期的课程计划，安排各门课程的时间、地点和教师，确保课程表的合理性和平衡性。
- **课程信息发布:** 将课程信息、教师信息、教室信息等发布给学生和教师，以便学生能够及时了解课程安排。

**排课功能** 教务员能够根据不同专业的学生人数及课程相关信息，排出合理课程表，避免时间冲突。

**成绩管理功能** 教师能够录入学生的考试成绩和平时成绩，并提供学生查询自己的成绩。

## 6.3 系统设计

### 6.3.1 界面设计

The screenshot shows the '教务信息' (Academic Information) section of the student interface. It includes a '今日课表' (Today's Class Schedule) table and a '成绩单' (Grade Report) table.

**今日课表**  
下节课上什么？

Class 1	8:00 - 9:00
Teacher 1	
Classroom 1	
Class 2	10:00 - 11:00
Teacher 2	
Classroom 2	
Class 3	14:00 - 15:00
Teacher 3	
Classroom 3	

**成绩单**  
查询成绩

90 分	2 学分
Course 1	
Teacher 1	
100 分	0.5 学分
Course 2	
Teacher 2	
80 分	3 学分
Course 3	
Classroom 3	

学生端操作界面

The screenshot shows the '教务信息' (Academic Information) section of the teacher interface. It includes a '今日教学课表' (Today's Teaching Class Schedule) table and a '我的教学班' (My Teaching Class) table.

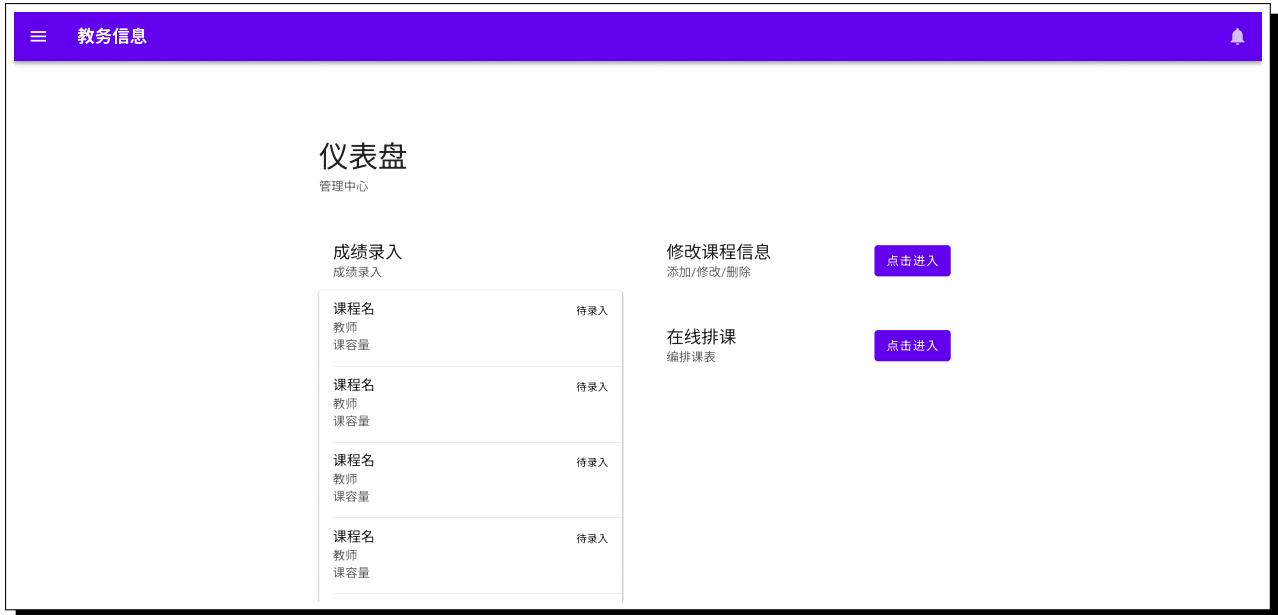
**今日教学课表**  
下节课教什么？

Class 1	8:00 - 9:00
Teacher 1	
Classroom 1	
Class 2	10:00 - 11:00
Teacher 2	
Classroom 2	
Class 3	14:00 - 15:00
Teacher 3	
Classroom 3	

**我的教学班**  
查询学生名单

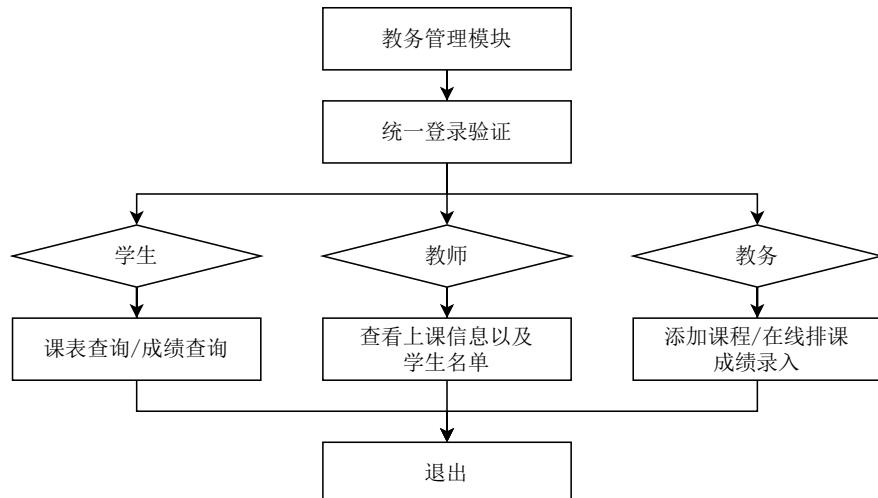
Class 1	2 学分
Classroom 1	
周一 9:00 - 10:00	
Class 2	0.5 学分
Classroom 2	
周一 9:00 - 10:00	
Class 3	3 学分
Classroom 3	
周一 9:00 - 10:00	

教师端操作界面



教务端操作界面

### 6.3.2 模块流程图



### 6.3.3 类分析

#### 1. 实体类

学生类: Student

序号	名称	类型	约束	备注
1	cardNumber	Integer	9位数字	学生一卡通号
2	studentNumber	Integer	8位数字	学号
3	major	Integer		专业编号
4	school	Integer		学院编号
5	status	enum		学籍状态
6	enrollment	calendar		入学日期
7	birthPlace	String		籍贯
8	poliStat	enum		政治面貌

开设课程的基本信息类: Course

序号	名称	类型	约束	备注
1	uuid	UUID		唯一标识符, 用作主键
2	courseId	Integer		课程编号
3	courseName	String		课程名
4	school	String		开设学院
5	credit	float		学分

教学班的基本信息类: Class

序号	名称	类型	约束	备注
1	uuid	UUID		唯一标识符, 用作主键
2	courseId	Integer		课程编号
3	courseName	String		课程名
4	teacherId	Integer		教师编号
5	schedule	String		上课信息
6	place	string		上课地点
7	capacity	String		上课信息

学生所选课程的基本信息类: SelectedClass

序号	名称	类型	约束	备注
1	uuid	UUID		唯一标识符, 用作主键
2	classUuid	UUID		课程 uuid
3	studentId	Integer	9位数字	学生一卡通号
4	selectTime	String		选课时间
5	grade	Integer	不超过三位数字	学生的课程分数

## 2. 服务类

客户端: TeachingAffairsClient

序号	名称	方法	备注
1	获取课程信息	getCourseInfo	获取课程 id, 名称, 上课地点, 老师 id 等
2	获取成绩	getGrade	获取课程成绩
3	添加课程	addCourse	创建新的课程
4	在线排课	scheduleClass	排课
5	录入学生成绩	enterGrades	录入学生成绩
6	教学班信息获取	getClassInfo	查看课程班的学生

服务端: TeachingAffairsController

序号	名称	方法	备注
1	获取课程信息	getCourseInfo	根据学号返回课程 id, 名称, 上课地点等
2	获取成绩	getGrade	根据学号返回课程成绩
3	添加课程	addCourse	创建新的课程
4	在线排课	scheduleClass	排课
5	录入学生成绩	enterGrades	录入学生成绩
6	教学班信息获取	getClassInfo	查看课程班的全部学生

## 7 图书馆模块设计说明

### 7.1 模块背景

用于管理书籍数据信息，为学生提供借阅、查询等服务并记录图书馆书籍借阅状况。

### 7.2 需求分析

**书籍信息管理** 图书馆模块系统需要记录书籍的各种信息，包括书名、ISBN、作者、出版社、简介及借阅状态（是否可借）等。需要提供对书本的基本的增删改查功能。

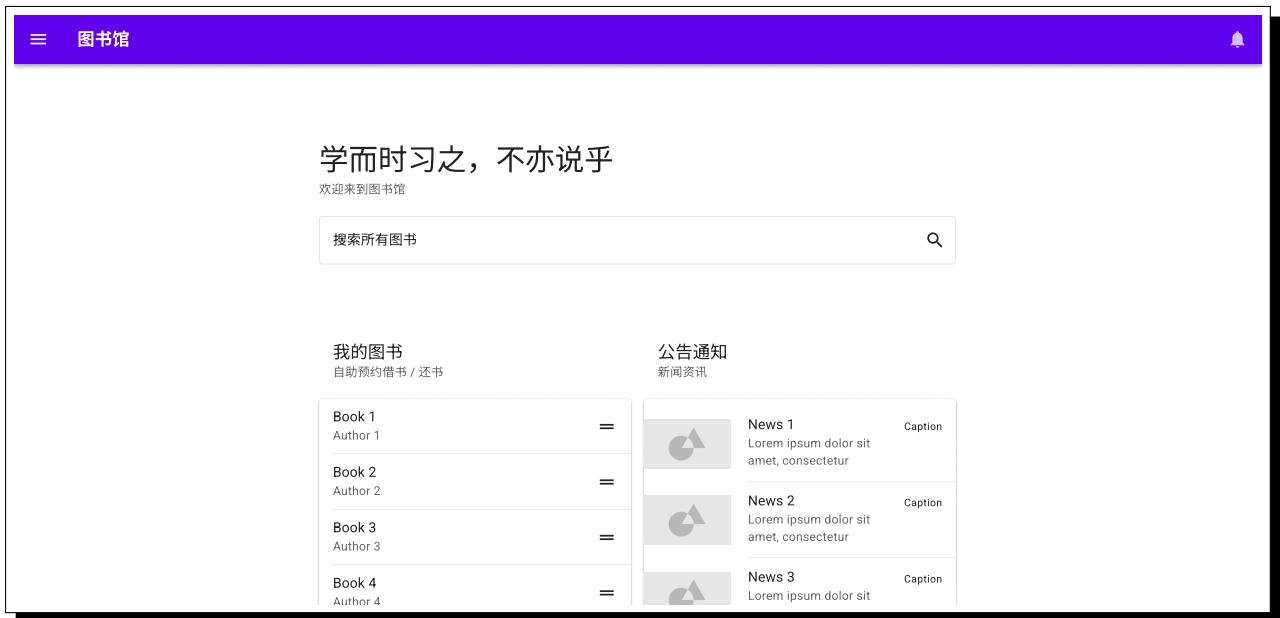
**用户使用** 用户在使用图书馆模块时，应该能够查询图书馆馆藏状态，查找到对应的图书信息，包括借阅状态、藏书地点等，需要提供相关接口实现还书、更改书籍状态，同时用户在图书馆模块的账户信息需要存储在数据库中。

**借阅记录** 图书馆模块需要记录书籍的借阅记录以便查询和追踪，需要记录的信息包括借出书名、借书人、操作类型（借书/还书）以及操作时间等。

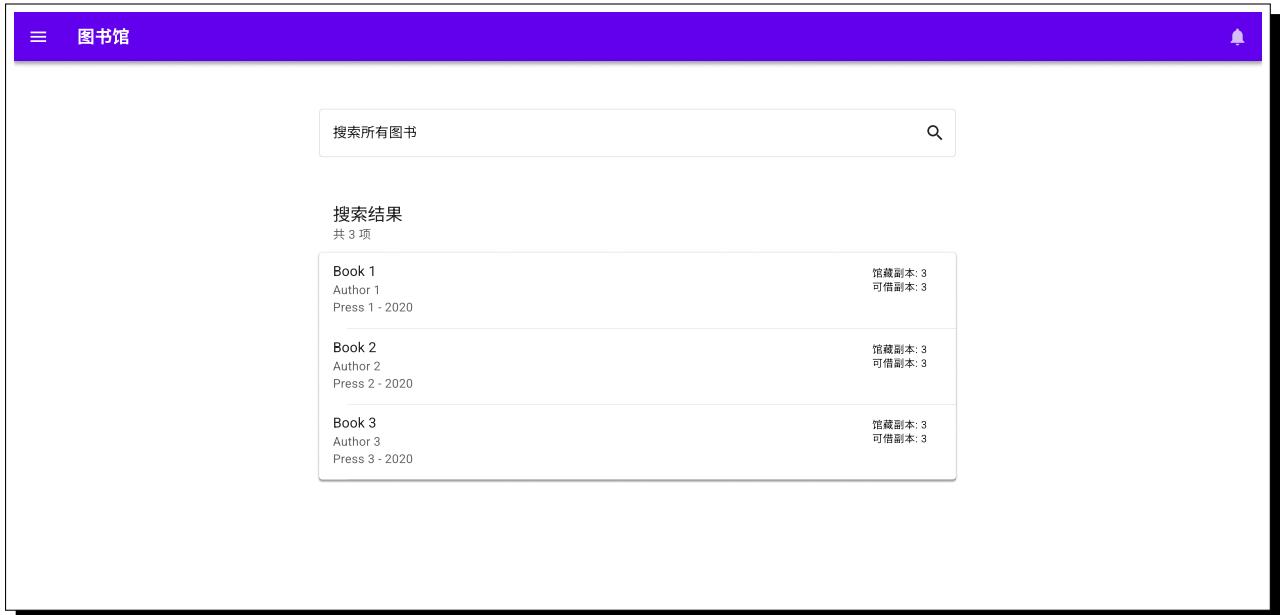
**错误及异常处理** 当遇到用户所借书籍不存在或者已被借阅，用户逾期未还书以及其他异情况时，模块都应给出相应处理。

## 7.3 系统设计

### 7.3.1 界面设计



图书馆检索界面



检索结果页面

## 仪表盘

管理中心

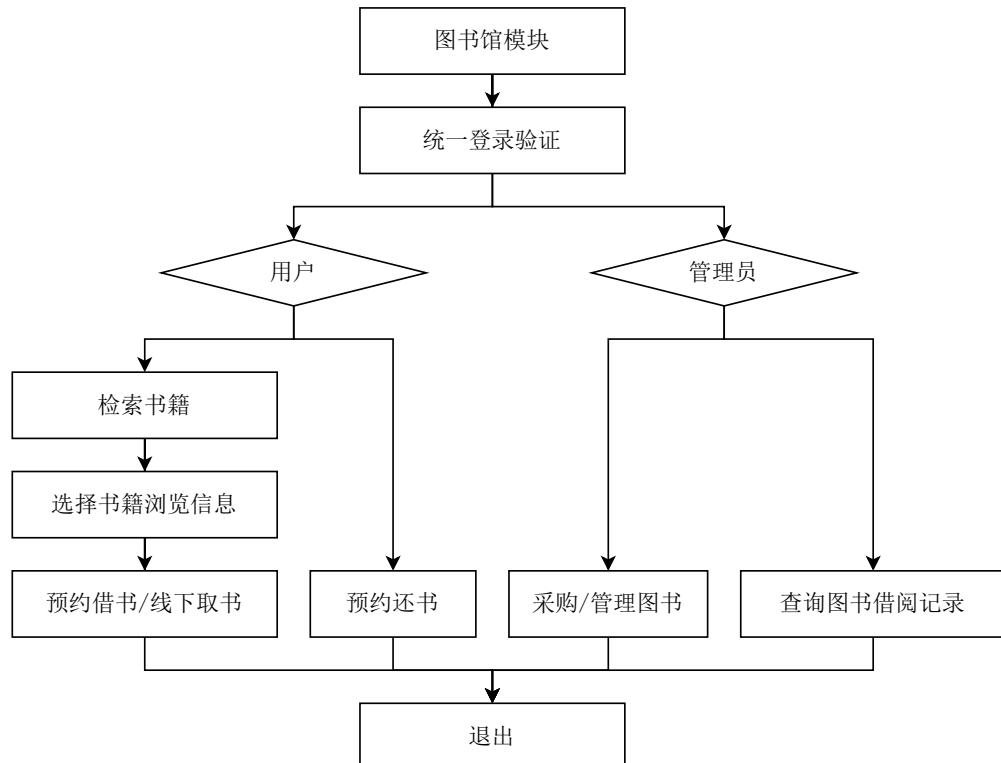
查看借/还书记录  
历史记录修改馆藏图书信息  
添加/修改/删除

点击进入

Book 1 用户 1 2023年1月1日	还书
Book 2 用户 2 2023年1月1日	借书
Book 3 用户 3 2023年1月1日	还书
Book 4 用户 4 2023年1月1日	借书

图书馆管理页面

## 7.3.2 模块流程图



### 7.3.3 类分析

#### 1. 实体类

馆藏图书信息类: LibraryBook

序号	名称	类型	约束	备注
1	uuid	UUID		图书编号
2	name	String		书名
3	isbn	String	ISBN 格式	ISBN
4	author	String		作者
5	press	String		出版社
6	description	String		简介
7	place	String		存放地点
8	status	enum		出借状态

用户信息类: LibraryUser

序号	名称	类型	约束	备注
1	userId	Integer	9 位数字	用户一卡通号
2	userStatus	enum		用户状态 c
3	borrowed	Integer	非负数	已借数量
4	maxBorrow	Integer	非负数	总共可借数量

图书借阅记录类: LibraryTransaction

序号	名称	类型	约束	备注
1	uuid	UUID		表内唯一标识信息
2	bookUuid	UUID		书 UUID
3	userId	Integer	9 位数字	借阅者一卡通
4	action	enum		借入/借出
5	time	LocalDateTime		操作时间

#### 2. 服务类

客户端: LibraryClient

序号	名称	方法	备注
1	检索图书	libFindBooks	用户检索书籍
2	选择书籍	libSelectBook	用户检索界面选择书籍浏览信息
3	预约借书	libBorrowBook	用户点击预约借书
4	预约还书	libReturnBook	用户预约还书
5	图书入库	libAddBook	管理员采购书籍, 图书入库
6	信息修改	libModifyInfo	管理员修改图书信息
7	查询借阅记录	libRecordQuery	管理员查询图书借阅记录

### 服务器端: LibraryController

序号	名称	方法	备注
1	返回检索信息	libGetBooks	依据用户的检索词返回对应的图书信息
2	返回书籍信息	libGetBookInfo	根据用户选择返回对应的书籍具体信息
3	状态修改	libStatusModify	根据用户操作修改图书状态
4	信息修改	libUpdateInfo	根据管理员修改的信息更新数据库
5	记录展示	libRecordShow	展示近期图书借阅记录

## 8 商店模块设计说明

### 8.1 模块背景

用于为学生提供各种商品，并通过在线支付手段进行支付完成交易。

### 8.2 需求分析

**用户角色和权限** 模块应支持不同的用户角色，如学生和商店管理员。不同角色应有不同的权限，例如学生可以浏览商品、下单购买，管理员可以管理用户、商品和订单。

**用户认证和安全性** 用户需要登录账号后才能进入模块，确保信息安全。支付密码应该进行加密存储。同时，应该实施合适的安全措施，防止恶意攻击和数据泄露。

**商品管理** 管理员应该能够添加、编辑和删除商品，并统计商品的销售情况。每个商品应该有名称、价格、库存数量、条形码等属性。商品应该有简洁的描述，以便用户更了解商品信息。

**购物流程** 用户浏览商店，选择感兴趣的商品，并进行结算。结算应该能够显示商品交易金额、交易时间，并支持添加备注，然后进行线上支付。

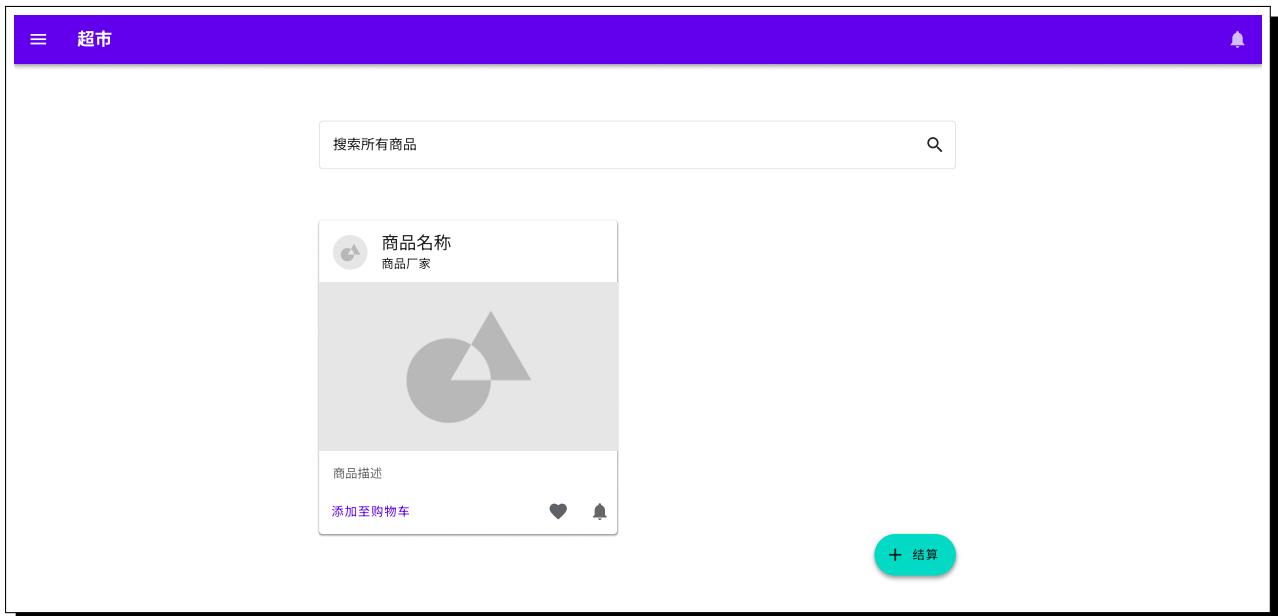
**数据统计和分析** 模块可以提供各种商品销量报告、热门商品统计、用户购买习惯等数据分析功能，帮助管理员做出更好的经营决策。

**用户界面和体验** 用户界面应该友好、易用，并且能够突出商品特性和详细信息。响应式设计和良好的用户体验是至关重要的。

**报错和异常处理** 模块应该能够处理用户可能遇到的各种错误和异常情况，如商品售罄、支付失败等问题，提供友好的错误提示和解决方案。

## 8.3 系统设计

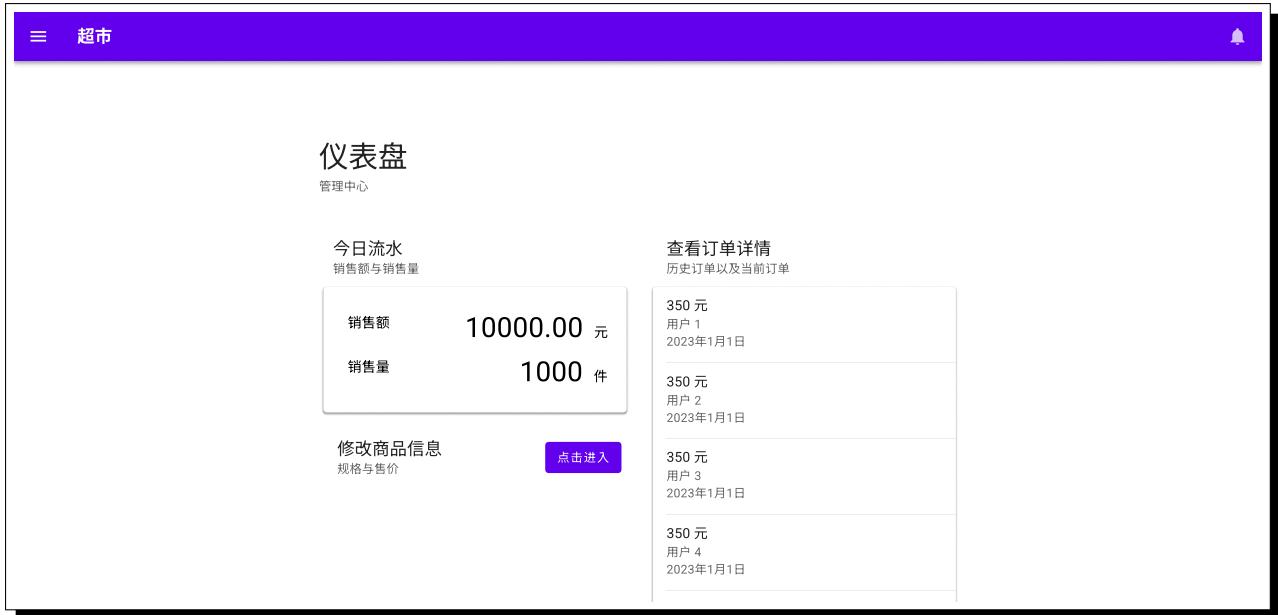
### 8.3.1 界面设计



商店选购页面

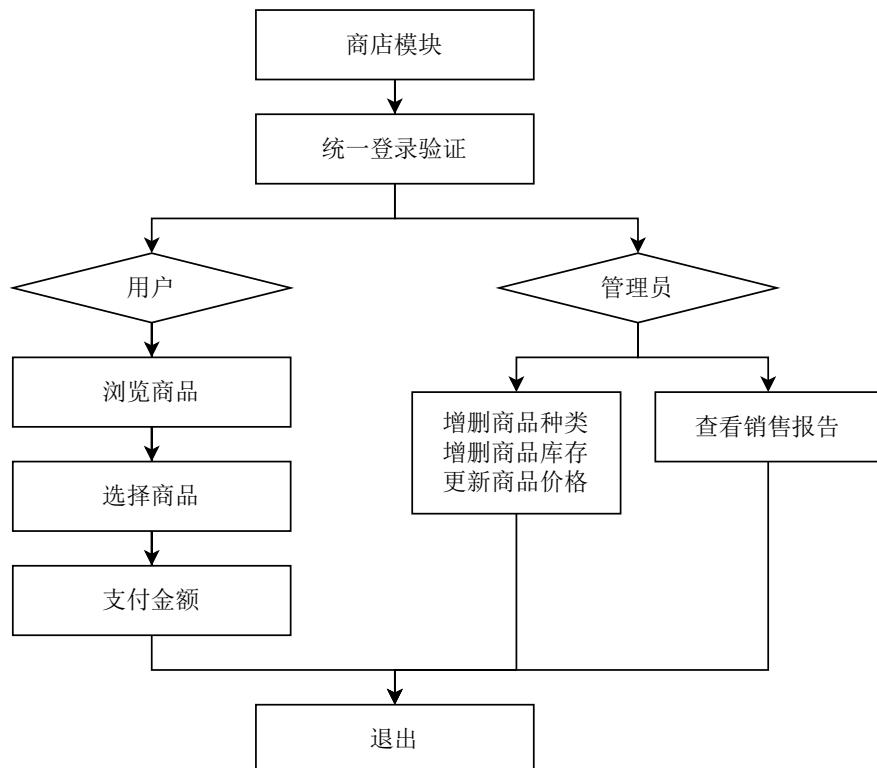


商店结算页面



商店管理页面

### 8.3.2 模块流程图



### 8.3.3 类分析

#### 1. 实体类

商品类: StoreItem

序号	名称	类型	约束	备注
1	uuid	UUID		商品编号
2	name	String	不超过 100 个字符	商品名称
3	price	Integer	非负数	商品金额
4	barcode	String	13 个字符	商品条形码
5	stock	Integer	非负数	商品库存数量
6	description	String		商品信息描述

商品交易记录类: StoreTransaction

序号	名称	类型	约束	备注
1	uuid	UUID		交易编号
2	itemUuid	UUID		交易中的商品编号
3	itemPrice	Integer	非负数	交易中的商品单价
4	amount	Integer	非负数	交易中的商品数量
5	cardNumber	Integer	9 位数字	交易关联一卡通号
6	time	LocalDateTime		交易的时间
7	remark	String		交易的备注信息

#### 2. 服务类

客户端: StoreClient

序号	名称	方法	备注
1	查找	retrieveItem	查找到要购买的商品
2	选择	selectItem	选择要购买的商品
3	移除	removeItem	从购物车中移除商品
4	结算	settle	结算选择的商品总金额
5	支付	pay	输入密码进行支付
6	新增商品	createItem	管理员添加新的商品
7	删除商品	deleteItem	管理员删除商品
8	更新信息	updateInfo	管理员更新商品信息（价格、库存等）
9	查看报告	getReport	管理员查看销售报告

#### 服务器端: StoreController

序号	名称	方法	备注
1	查找	getItem	根据关键词查找商品，并返回匹配的商品
2	记录	selectItem	记录选择的商品信息
3	移除	removeItem	从记录中移除某个商品
4	结算	settle	结算选择的商品总金额
5	支付	pay	根据账号和密码进行支付
6	新增商品	createItem	管理员添加新的商品
7	删除商品	deleteItem	管理员删除商品
8	更新信息	updateInfo	管理员更新商品信息（价格、库存等）
9	查看报告	getReport	生成销售报告

## 9 TODO: 其他选做模块

## 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

### 10.2 用户信息模型

### 10.3 工具类

#### 10.3.1 数据库工具类

名称	说明	备注
init()	完成配置初始化	返回 org.hibernate.Session 对象

### 10.3.2 密码工具类

名称	说明	备注
encoder()	返回 Argon2PasswordEncoder	
hash(password)	返回 hash 后的密码	使用 Argon2 算法
verify(password, hashed)	验证密码	

## 11 网络模块设计说明

为方便开发以及性能，我们采用 Netty 库作为 Socket 实现，客户端与服务端的数据交换封装为 Request 与 Response 实现，通过扩展 ChannelInboundHandlerAdapter 实现数据发送与接收，并使用 Gson 库进行数据序列化。

### 11.1 Socket

#### 11.1.1 客户端

#### 11.1.2 服务器端 NettyServer

名称	说明	备注
NettyServer(port)	初始化相关配置	传入端口
run(router, database)	启动服务器	传入 Router 与数据库 Session

```
1 NettyServer server = new NettyServer(9090);
2 server.run(router);
```

### 11.2 输入输出流

#### 11.2.1 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。

```
1 EventLoopGroup bossGroup = new NioEventLoopGroup();
2 EventLoopGroup workerGroup = new NioEventLoopGroup();
3 try {
4     ServerBootstrap b = new ServerBootstrap();
5     b.group(bossGroup, workerGroup)
6         .channel(NioServerSocketChannel.class)
7         .childHandler(new ChannelInitializer<SocketChannel>() {
8             @Override
9             public void initChannel(@NotNull SocketChannel ch) {
10                 ch.pipeline().addLast(new JsonObjectDecoder())
11                     addLast(new NettyHandler(router));
12             }
13         });
14     b.bind(9090).sync();
15 }
```

```

11             }
12         })
13         .option(ChannelOption.SO_BACKLOG, 128)
14         .childOption(ChannelOption.SO_KEEPALIVE, true);
15     ChannelFuture f = b.bind(port).sync();
16     f.channel().closeFuture().sync();
17 } finally {
18     workerGroup.shutdownGracefully();
19     bossGroup.shutdownGracefully();
20 }

```

```

1 @Override
2 public void channelRead(@NotNull ChannelHandlerContext ctx, @NotNull
3                         Object msg) {
4     ByteBuf in = (ByteBuf) msg;
5     try {
6         log.info("[{}] Received: {}", ctx.channel().id(), in.toString(
7             CharsetUtil.UTF_8));
8         Request request = gson.fromJson(in.toString(CharsetUtil.UTF_8)
9             , Request.class);
10        request.setSession(session);
11        if (!router.hasRoute(request.getUri())) {
12            log.info("[{}] Route not found: {}", ctx.channel().id(),
13                  request.getUri());
14            sendResponse(ctx, Response.Common.notFound());
15            return;
16        } else if (!session.permission(router.getRole(request.getUri())))
17            {
18                log.info("[{}] Permission denied: {}", ctx.channel().id(),
19                      request.getUri());
20                sendResponse(ctx, Response.Common.permissionDenied());
21                return;
22            }
23            Response response = router.invoke(request);
24            if (response.getSession() != null) {
25                log.info("[{}] Session updated: {}", ctx.channel().id(),
26                      response.getSession());
27                session = response.getSession();
28            }
29            sendResponse(ctx, response);
30        } catch (Exception e) {
31            log.error("[{}] Exception: {}", ctx.channel().id(), e.

```

```

        getMessage());
25    } finally {
26        ReferenceCountUtil.release(msg);
27    }
28 }

```

### 11.2.2 发送输出

```

1 private void sendResponse(ChannelHandlerContext ctx, Response response
2 ) {
3     ctx.writeAndFlush(Unpooled.copiedBuffer(gson.toJson(response),
4         CharsetUtil.UTF_8));
5 }

```

## 12 多线程模块设计说明

对于 UI，Compose 库自行管理渲染线程，无需编写相关多线程代码。

对于 Socket，使用 Netty 内置的 NioEventLoopGroup，包括了线程池与 NIO 的使用，在独立线程中处理 Socket 连接。

## 13 数据库设计说明

### 13.1 元数据

共有 14 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
major	存储专业查询表
school	存储学院查询表
accounting	存储财务账户信息
card_transaction	存储财务交易记录
course	存储课程信息
class	存储教学班信息
selected_class	存储选课信息
library_book	存储图书馆馆藏书目信息
library_user	存储图书馆用户信息
library_transaction	存储图书馆借还书记录
store_item	存储商品信息
store_transaction	存储商店交易记录

## 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、教职员等。

字段	类型	约束	说明
card_number	INT	9 位数字	存一卡通号
password	VARCHAR		存储处理过的密码
name	VARCHAR		存储用户名
gender	ENUM		存储用户性别
phone	VARCHAR		存储用户手机号
email	VARCHAR		存储用户邮箱
role	VARCHAR		存储用户角色

### 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
card_number	INT	9 位数字	存一卡通号
student_number	VARCHAR	8 位数字	存学号
major	INT		专业编号
school	INT		学院编号
status	ENUM		学籍状态
enrollment	DATE		入学日期
birth_place	VARCHAR		籍贯
political_status	ENUM		政治面貌

### 13.2.3 major 表

major 表用于存储专业编号对应的专业名称。

字段	类型	约束	说明
major_number	INT		专业编号
major_name	VARCHAR		对应名称

### 13.2.4 school 表

school 表用于存储学院编号对应的学院名称。

字段	类型	约束	说明
school_number	INT		学院编号
school_name	VARCHAR		对应名称

### 13.2.5 accounting 表

accounting 表用于存储所有人一卡通的余额以及状态。

字段	类型	约束	说明
card_number	INT	9 位数字	存一卡通号
balance	INT	非负数	存卡内余额（以分为单位）
card_status	ENUM		存一卡通状态

### 13.2.6 card\_transaction 表

card\_transaction 表用于存储一卡通的交易信息。

字段	类型	约束	说明
uuid	UUID		表内唯一标识信息，用作主键
card_number	INT	9 位数字	存一卡通号
amount	INT	非负数	存交易金额（以分为单位、带符号）
time	DATETIME		存交易时间
remark	TEXT		备注

### 13.2.7 course 表

course 表用于存储课程信息。

字段	类型	约束	说明
uuid	UUID		表内唯一标识信息，用作主键
course_id	VARCHAR		课程编号
course_name	VARCHAR		课程名称
school	INT		开设学院编号
credit	DECIMAL		存课程学分

### 13.2.8 class 表

class 表用于存储教学班信息。

字段	类型	约束	说明
uuid	UUID		表内唯一标识信息，用作主键
course_id	VARCHAR		课程编号
course_name	VARCHAR		课程名称
teacher_id	INT	9 位数字	教师一卡通号
schedule	JSON		上课信息
place	VARCHAR		上课地点
capacity	INT		课容量

### 13.2.9 selected\_class 表

selected\_class 表用于存储学生选课以及成绩信息。

字段	类型	约束	说明
uuid	UUID		表内唯一标识信息, 用作主键
class_uuid	UUID		存储教学班 uuid
student_id	INT	9 位数字	学生一卡通号
select_time	DATETIME		选课时间
grade	INT	不超过三位数字	成绩

### 13.2.10 library\_book 表

library\_book 表用于存储图书馆馆藏书目信息。

字段	类型	约束	说明
uuid	UUID		表内唯一标识信息, 用作主键
name	VARCHAR		书名
isbn	VARCHAR	ISBN 格式	ISBN
author	VARCHAR		作者
press	VARCHAR		出版社
description	VARCHAR		简介
place	VARCHAR		存放地点
status	ENUM		出借状态

### 13.2.11 library\_user 表

library\_user 表用于存储图书馆用户信息。

字段	类型	约束	说明
user_id	INT	9 位数字	用户一卡通号
user_status	ENUM		用户状态
borrowed	INT	非负数	已借数量
max_borrow	INT	非负数	总共可借数量

### 13.2.12 library\_transaction 表

library\_transaction 表用于存储图书馆借还书记录。

字段	类型	约束	说明
uuid	UUID		表内唯一标识信息, 用作主键
book_uuid	UUID		书 uuid
user_id	INT	9 位数字	一卡通
action	ENUM		借入/借出
time	DATETIME		操作时间

### 13.2.13 store\_item 表

store\_item 表用于存储商店商品信息。

字段	类型	约束	说明
uuid	UUID		表内唯一标识信息，用作主键
name	VARCHAR	不超过 100 个字符	商品名称
price	INT	不能为负数	商品价格（以分为单位）
barcode	VARCHAR	13 个字符	商品条形码
stock	INT	不能为负数	库存数量
description	TEXT		商品描述

### 13.2.14 store\_transaction 表

store\_transaction 表用于存储商店交易信息。

字段	类型	约束	说明
uuid	UUID		表内唯一标识信息，用作主键
item_uuid	UUID		商品 uuid
item_price	INT	不能为负数	商品交易时单价
amount	INT	不能为负数	存交易数量
card_number	INT	9 位数字	交易关联一卡通号
time	DATETIME		存交易时间
remark	TEXT		备注

## 14 其他

早期版本，不代表最终成果。