**河马先生酒店平台**

**《后端技术开发文档》**



**东南大学**

**计算机科学与工程学院、软件学院、人工智能学院**

**指导老师： 王红兵**

**组长： 闫志豪**

**小组成员： 09019226 闫志豪**

**09019212 卢昊飞**

**09019204 曹邹颖**

**09019323 潘昱辰**

二〇二二 年 六 月

版本信息

|  |  |  |  |
| --- | --- | --- | --- |
| 版本号 | 修订人 | 修订日期 | 修订原因 |
| v1.0 | 河马先生全体小组成员 | 2022.06.19 | 新建 |

目录

[第一部分 引言 4](#_Toc106908014)

[1.1 编写目的 4](#_Toc106908015)

[1.2 读者对象 4](#_Toc106908016)

[1.3 项目背景 4](#_Toc106908017)

[第二部分 后端数据库设计 5](#_Toc106908018)

[2.1 概念数据模型设计 5](#_Toc106908019)

[2.1.1 E-R图 5](#_Toc106908020)

[2.1.2 数据流图 5](#_Toc106908021)

[2.2 逻辑数据模型设计 6](#_Toc106908022)

[2.2.1 数据模式（Data schema）设计 6](#_Toc106908023)

[2.2.2 具体数据库设计 7](#_Toc106908024)

[2.3 MYSQL数据库实现 12](#_Toc106908025)

[第三部分 后端开发技术 12](#_Toc106908026)

[3.1 后端技术选型概述 12](#_Toc106908027)

[3.2 后端技术具体阐述 13](#_Toc106908028)

[3.2.1 后端主体框架——Spring Boot框架 13](#_Toc106908029)

[3.2.2 数据库操作框架——Spring Data JPA 13](#_Toc106908030)

[3.2.3 安全权限管理框架——Spring Security 14](#_Toc106908031)

[3.2.4 JSON web 令牌——JWT 15](#_Toc106908032)

[3.2.5 分布式搜索和分析引擎——Elasticsearch 17](#_Toc106908033)

[3.2.6 邮件服务——JavaMailSender 19](#_Toc106908034)

[3.2.7 第三方支付接口——支付宝 alipay easysdk 20](#_Toc106908035)

[3.2.8 统一的异常处理 21](#_Toc106908036)

[3.2.9 统一的日志记录——SLF4J和Logback日志框架 22](#_Toc106908037)

[3.2.10 手机短信服务 23](#_Toc106908038)

[3.2.11 高安全性能 24](#_Toc106908039)

第一部分 引言

## 1.1 编写目的

后端开发技术文档的编写主要是为了给开发《河马先生酒店平台》做主要的规划和整合，在开发过程中起到引导作用，保证项目团队学习使用的技术完全遵照文档规定，便于项目团队成员更好地合作，使项目工作开展的各个过程合理有序，以文件化的形式，作为项目团队成员之间的共识与约定，它又是项目生命周期内的所有项目活动的行动基础、项目团队开展和检查项目工作的依据。

在项目开发的过程中，小组成员对技术开发文档不断更新完善，确保技术开发文档与项目实际情况保持一致。标准规范的项目开发文档也有利于技术人员后期的维护与运营。

## 1.2 读者对象

该文档的读者为《河马先生酒店平台》开发小组后端成员、软件分析人员、开发管理人员和测试人员。

## 1.3 项目背景

据前瞻数据，大的环境下酒店预定市场稳步发展，已经步入万亿级市场。

OTA酒店代理商这个行业的发展前景光明。行业门槛适中，处于蓬勃发展期

时代角度上，无论是疫情时代“酒店隔离”抑或后疫情时代“旅游经济复苏”，酒店预定热潮重现，我国现在正处于旅游业高速发展的时期，越来越多的景区在被开发，一定会有越来越多的酒店在被建设，而且未来酒店的线上预订将逐步取代线下预订，这样一来酒店OTA的市场将进一步扩大。酒店OTA代理基本不存在技术壁垒，操作比较简单，所以未来一定会有大量的竞争者涌入市场。因此，当下是一个绝佳的发展时期。

最后，越来越多的在线平台开始拥抱大数据，大数据量下系统的稳定性保障、如何精准直观地展示数据成为本项目的另一个立足之处。

因此，本项目从以上角度出发，力求通过项目实践更好地接触时代所需的真实开发需求，并借助项目进一步锻炼当代双一流计算机科学与技术专业学生的开发能力与解决现实问题的能力。

第二部分 后端数据库设计

## 2.1 概念数据模型设计

概念数据模型是面向用户、面向现实世界的数据模型，是与DBMS无关的，它主要用来描述一个概念化的结构。针对大学数据库系统分别设计E-R图和数据流图如下：

### 2.1.1 E-R图

根据需求分析，设计河马先生-酒店平台数据库E-R图如下：

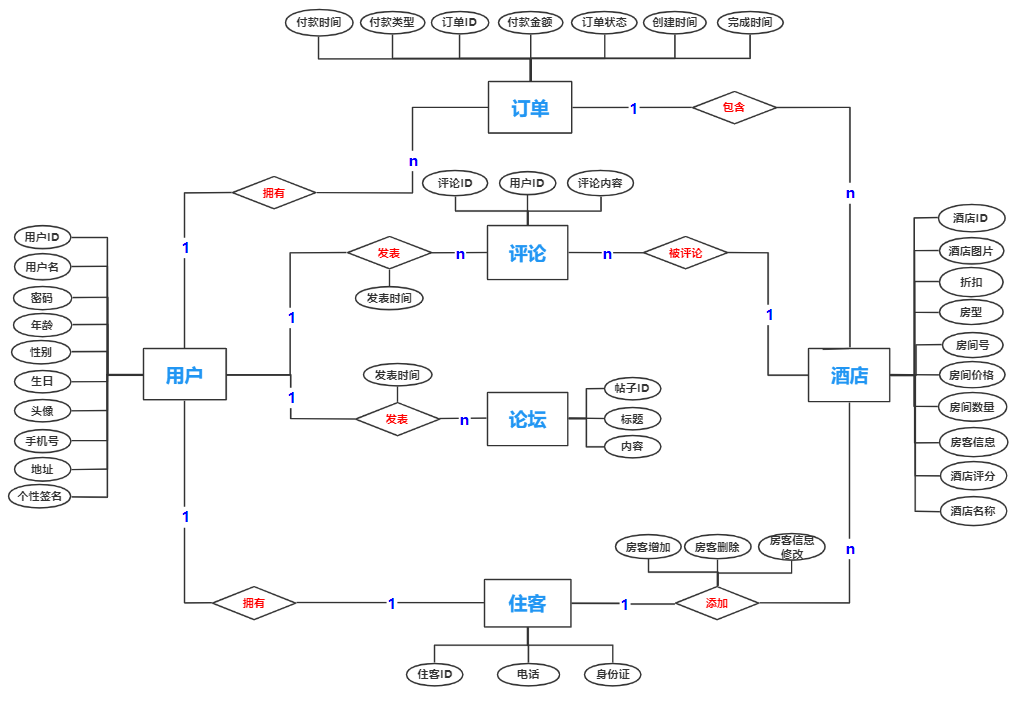


图4.1 E-R图设计

### 2.1.2 数据流图

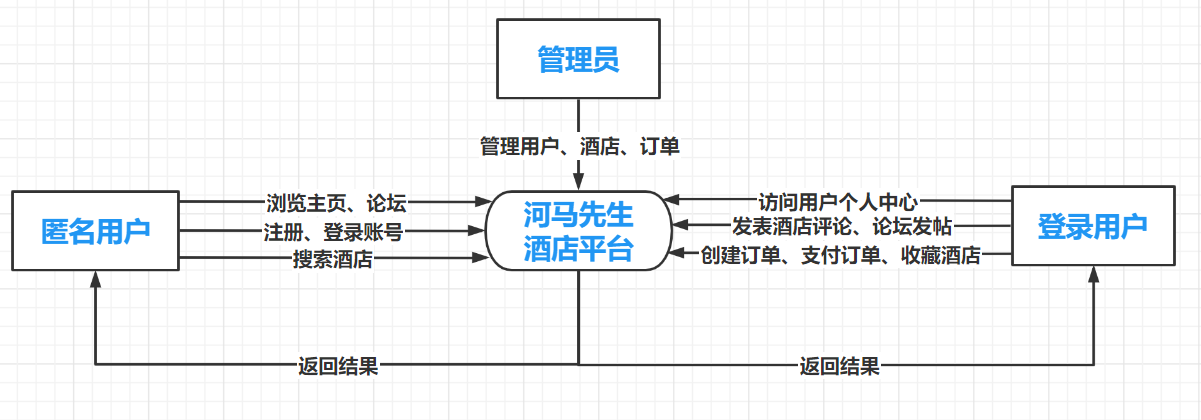


图4.2 数据流图

## 2.2 逻辑数据模型设计

逻辑数据模型是用户从数据库所看到的数据模型，它与DBMS有关，本实验中采用关系数据模型，将上述概念数据模型转化为如下的关系数据模型。

### 2.2.1 数据模式（Data schema）设计

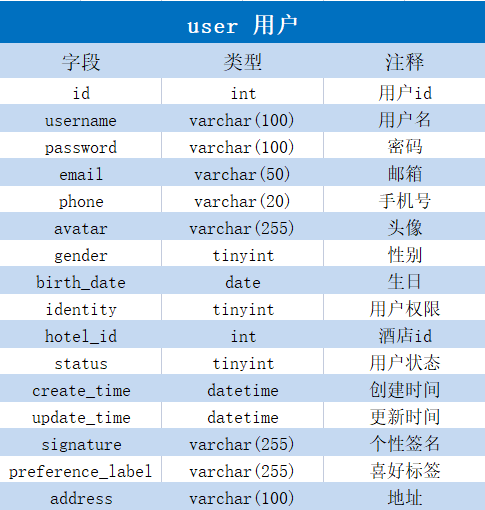
本应用拟设计实现具有上述功能需求分析中所有功能的河马先生-酒店平台，在数据库设计中包括用户、酒店、住客、订单、酒店预定、房间信息、房间种类、收藏夹、酒店评论、论坛、验证码、酒店设施用品、会员共13个关系，具体数据模式如下（每个关系中标红的为主键）：

|  |  |
| --- | --- |
| 关系名 | Data schema |
| 用户 | user**(id**,username,password,email,phone,avatar,  gender,birth\_date,hotel\_id,signature,preference\_label,  address,status,identity,creat\_time,update\_time**)** |
| 酒店 | hotel(**id**,name,location,picture,star,phone,  description,create\_time,update\_time) |
| 住客 | guest(**id**,user\_id,name,phone,id\_number,create\_time,update\_time) |
| 订单 | orders(**id**,user\_id,category\_id,number,status,payment\_money,  payment\_type,create\_time,complete\_time) |
| 酒店预定 | reservation(**id**,user\_id,order\_id,room\_id,order\_ith\_room,  guest\_id1,guest\_id2,guest\_id3,start\_time,end\_time) |
| 房间信息 | room(**id,name**,room\_category\_id,status) |
| 酒店评论 | comment(**id**,user\_id,hotel\_id,content,star,create\_time) |
| 论坛 | forum(**id**,user\_id,title,content,create\_time) |
| 验证码 | verification(**id**,ver\_code,create\_time) |
| 房间种类 | room\_category（**id**,name,hotel\_id,picture,price,description,max\_people,  size,wifi,window,breakfast,create\_time,update\_time） |
| 收藏夹 | favorite（**id**,user\_id,hotel\_id,create\_time） |
| 会员 | vip（**id**，user\_id,vip\_type,discount,total\_score,create\_time） |
| 酒店设施用品 | warehouse（**id**,hotel\_id,item\_name,item\_number） |

### 2.2.2 具体数据库设计

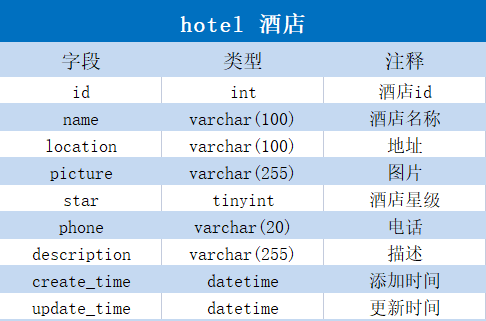
1.user用户表

该表主要用于存储用户个人基本信息、用户状态权限以及创建时间。



2.hotel酒店表

该表主要用于存储酒店基本信息添加时间以及酒店星级。



3.guest住客表

该表主要用于存储入住房客的基本信息、创建时间与更新时间。



4.orders订单表

该表主要用于存储用户订单基本信息，订单相关时间信息。



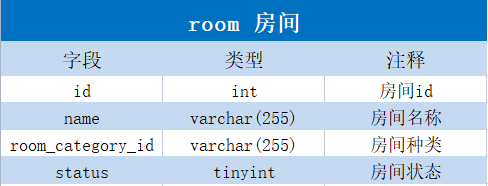
5.reservation酒店预定表

该表主要用于存储酒店房间预定信息。



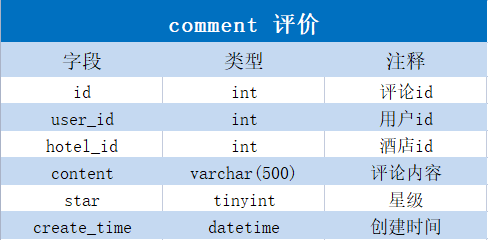
6.room房间信息表

该表主要用于存储酒店每个房间信息。



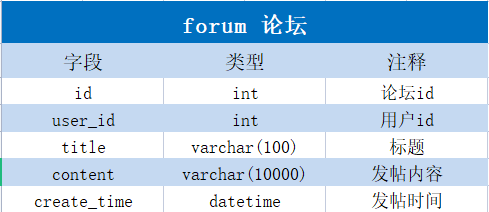
7.comment酒店评论表

该表主要用于存储用户对于酒店的评论基本信息。



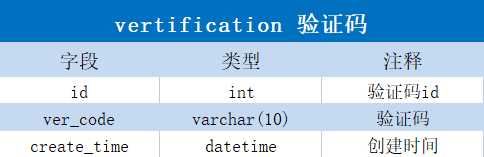
8.forum论坛表

该表主要用于存储用户交流论坛中帖子相关信息。



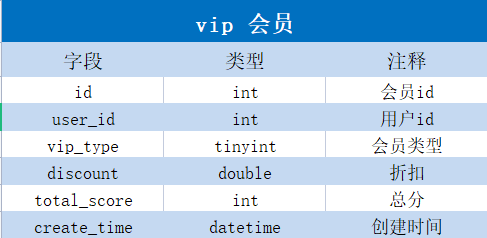
9.verfication验证码表

该表主要用于存储用户验证码与验证码创建时间，便于进行时间失效检测。



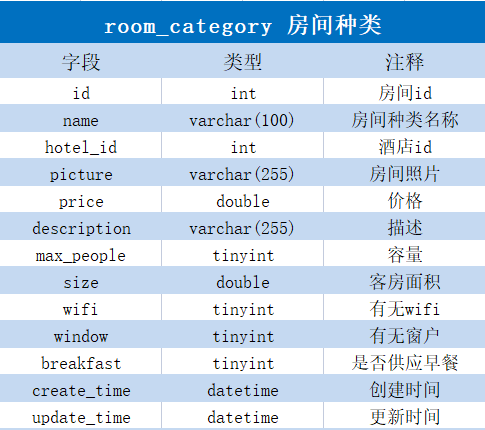
10.vip会员表

该表主要用于存储会员信息。



1. room\_category房间种类表

该表主要用于存储房间种类信息。



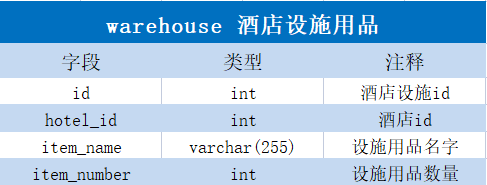
12.favorite收藏夹表

该表主要用于存储用户收藏信息。



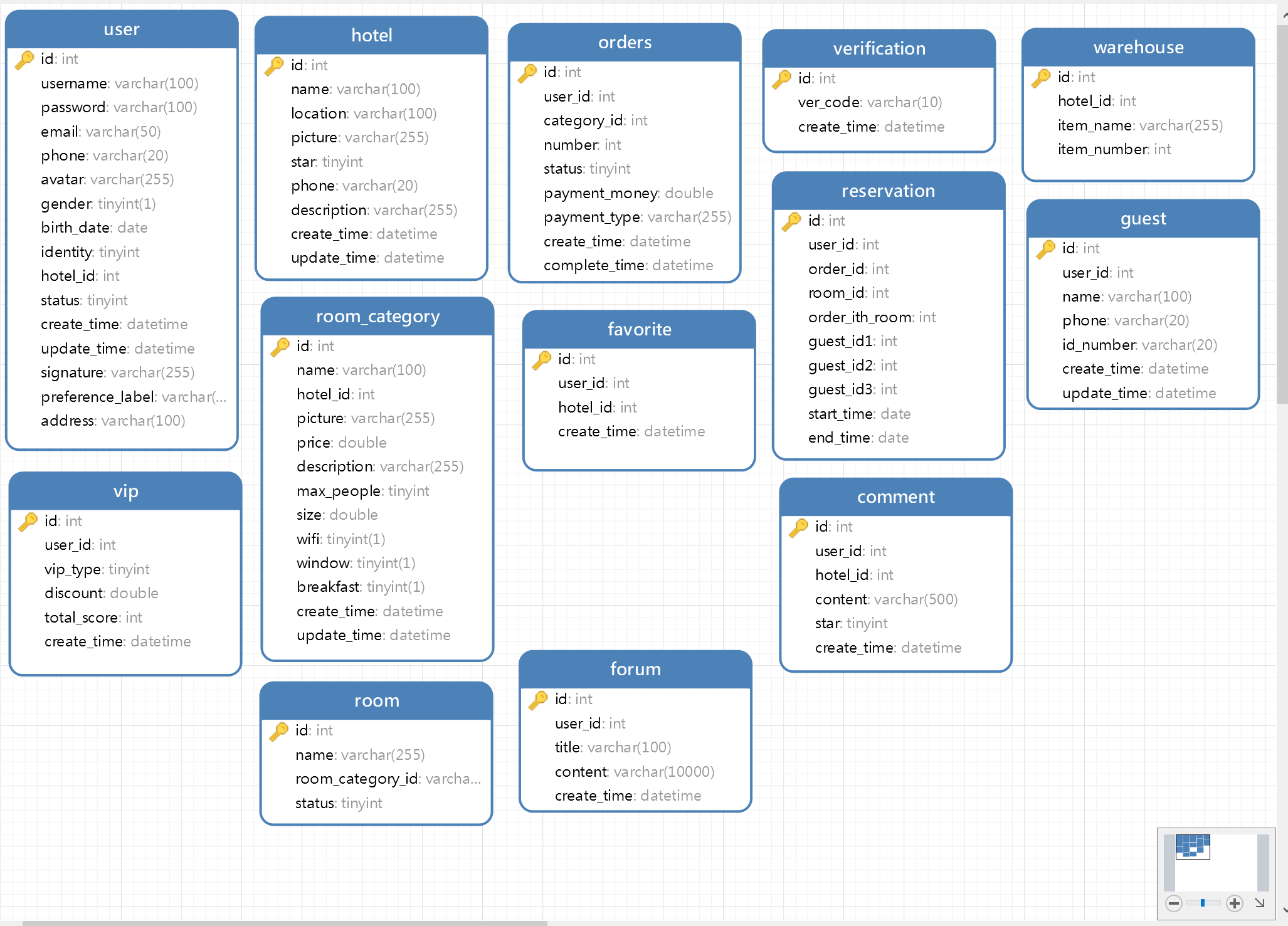
13.warehouse酒店设施用品表

该表主要用于存储酒店设施用品信息。



## 2.3 MYSQL数据库实现

本平台采用MYSQL数据库，将上述逻辑数据模型在MYSQL数据库中实现效果如下：



第三部分 后端开发技术

## 3.1 后端技术选型概述



图3.2.1 后端技术选型总体概述图

## 3.2 后端技术具体阐述

### 3.2.1 后端主体框架——Spring Boot框架

Spring Boot 是由 Pivotal 团队提供的全新框架，其设计目的是用来简化新 Spring 应用的初始搭建以及开发过程。该框架使用了特定的方式来进行配置，从而使开发人员不再需要定义样板化的配置。

Spring Boot最大的优点式自动装配，主要包括以下两点：

* 配置简单，避免使用大量XML方式进行配置；
* 开发者专注于业务逻辑，极大地提高了开发效率；

Spring Boot框架分层设计包括4层，分别是entity层、dao层、service层、controller层。各层功能分别如下：

**（1）entity层**

entity层即数据库实体层，也被称为model层，pojo层。一般数据库一张表对应一个实体类，类属性同表字段一一对应。

**（2）dao层**

dao层即数据持久层，也被称为mapper层。dao层的作用为访问数据库，向数据库发送sql语句，完成数据的增删改查任务。

**（3）service层**

service层即业务逻辑层。service层的作用为完成功能设计与业务逻辑的实现。service层调用dao层接口，接收dao层返回的数据，完成项目的基本功能设计。

**（4）controller层**

controller层即控制层。功能为请求和响应控制。controller层负责前后端交互，接受前端请求，调用service层，接收service层返回的数据，最后返回具体的页面和数据到客户端。

### 3.2.2 数据库操作框架——Spring Data JPA

Spring Data JPA是Spring提供对数据库访问的技术，其中包括关系型数据库、非关系型数据库（No-sql数据库）、以及map-reduce（大数据Hadoop）框架的访问。

Spring Data JPA的优点：

**（1）简化持久层开发工作**

Spring Data JPA可以让我们对于基础功能的实现在持久层不用去书写sql语句，直接继承Repository接口的Dao接口，开启JPA的注解扫描就可以进行持久层开发。对于一些复杂功能还提供了属性表达式、命名查询、Query注解等方式供我们使用。

**（2）简化分页业务处理**

Spring Data JPA的Repository接口实现了PagingAndSortingRepository接口，可以直接传参调用实现分页功能。

在本项目开发过程中，Spring Data JPA的应用主要为以下三点：

* 基础功能：数据库增删改查功能（CRUD）；
* 复杂功能：数据统计、模糊查询；
* 分页与排序功能；

### 3.2.3 安全权限管理框架——Spring Security

Spring Security是一个功能强大且高度可定制的身份验证和访问控制框架。提供了完善的认证机制和方法级的授权功能。是一款非常优秀的权限管理框架。它的核心是一组过滤器链，不同的功能经由不同的过滤器。

网络安全问题至关重要。作为生鲜购物平台开发项目，涉及到用户个人信息、用户与商家金钱交易等诸多用户个人数据，项目开发的安全性能极其关键。我们选择了基于 Spring AOP 和 Servlet 过滤器的Spring Security安全框架，确保项目拥有很好的安全性能。

Spring Security的**核心功能**包括以下三点：

（1）认证（你是谁，用户/设备/系统）

（2）验证（你能干什么，也叫权限控制/授权，允许执行的操作）

（3）攻击防护（防止伪造身份）

Spring Security的**原理技术**：创建一系列的过滤层和拦截器来进行请求的验证和拦截，由认证管理器（AuthenticationManager）和决策管理器（AccessDecisionManager）来放行或拒绝请求（如下图），以此来达到安全的效果。



图3.2.2 Spring Security原理技术图

Spring Security有以下两大优点：

* 免去我们手动配置各种拦截器和过滤器的操作；
* 为项目提供更好的安全性能；

对于河马先生购物商城web应用，基本的用户注册页、用户登录页等页面可以任何用户无权限访问，而对于用户主页、订单等用户个人信息需要用户权限，并且需要设置拦截，我们通过Spring Security 进行了很好的权限管理。

### 3.2.4 JSON web 令牌——JWT

Json web token (JWT), 是为了在网络应用环境间传递声明而执行的一种基于JSON的开放标准（(RFC 7519).该token被设计为紧凑且安全的，特别适用于分布式站点的单点登录（SSO）场景。JWT的声明一般被用来在身份提供者和服务提供者间传递被认证的用户身份信息，以便于从资源服务器获取资源，也可以增加一些额外的其它业务逻辑所必须的声明信息，该token也可直接被用于认证，也可被加密。总体流程图如下：

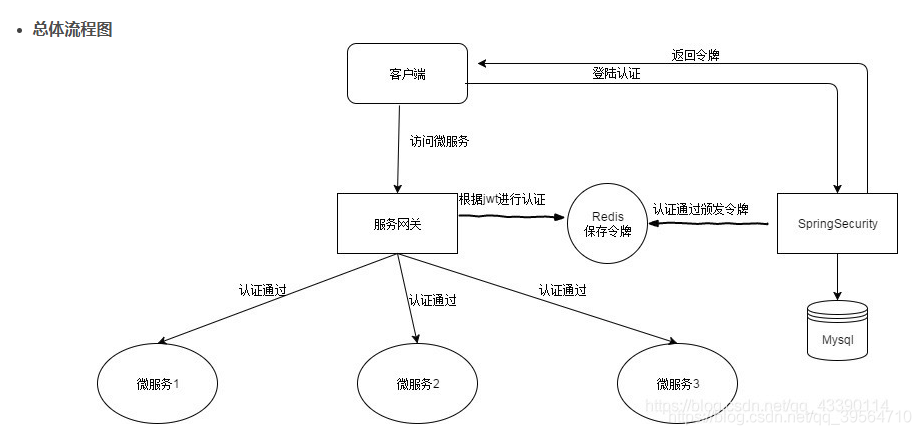


图3.2.3 JWT总体流程图

相比与传统session的用户验证方式，token验证主要具有以下优点：

（1）支持跨域访问，有效预防跨域网站攻击，更加安全；

（2）在网络传输过程中性能更好；

（3）不占用服务器储存资源；

（4）适合移动设备；

**Spring Security 与 JWT结合。**由于本项目采用前后端分离的方法，首先要解决的就是登录及授权的问题。传统的session认证限制了应用的扩展能力，无状态的JWT认证方法应运而生，该认证机制特别适用于分布式站点的单点登录（SSO）场景。

但spring security本身是通过session验证，不原生支持token验证，我小组成员为了使项目实现高安全性能与无状态验证，在spring security框架的基础上，手动添加了jwt token过滤层与相应的处理组件，以实现token的发放与验证。

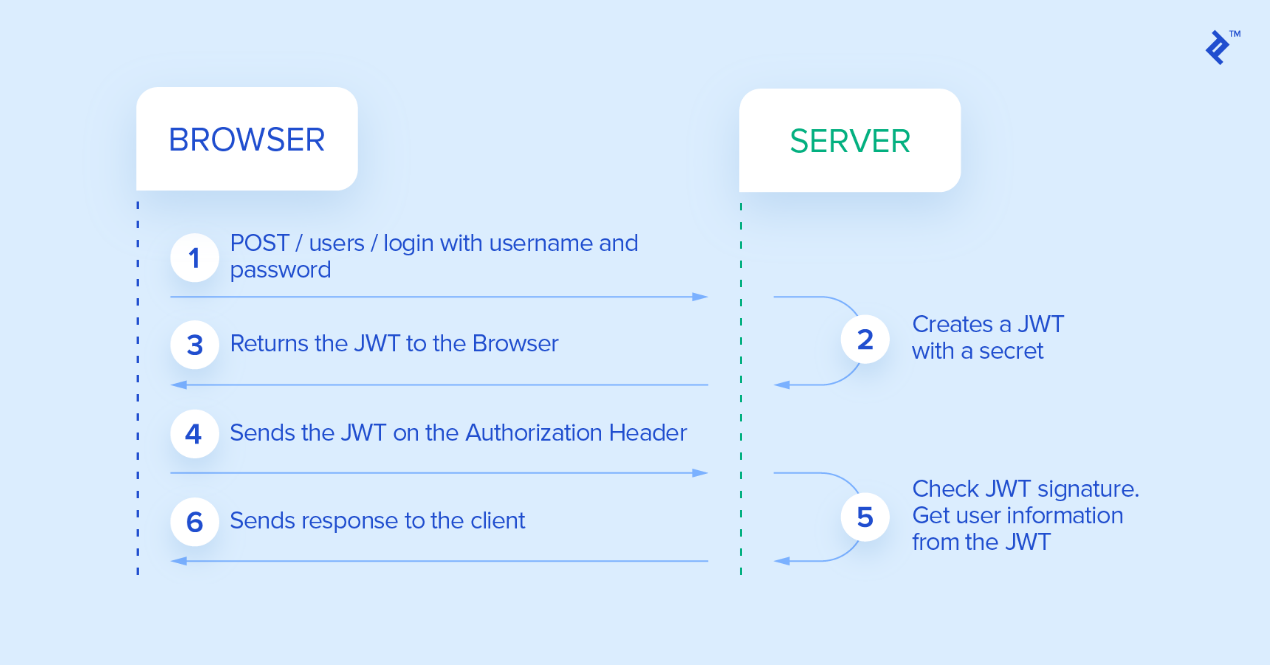


图3.2.4 本项目JWT实现流程图

### 3.2.5 分布式搜索和分析引擎——Elasticsearch

Elasticsearch 是一个分布式、高扩展、高实时的搜索与数据分析引擎。它能很方便的使大量数据具有搜索、分析和探索的能力。



图3.2.5 Elasticsearch

**本项目使用Elasticsearch的原因：**

（1）相比于传统的数据库，由于采用倒排索引，Elasticsearch的搜索响应时间平均较短。

（2）相对于数据库的精准匹配，Elasticsearch的相关性匹配更贴近人的思维方式。在本次项目中，我们小组采用**相关度打分匹配**的方式，在返回结果中就会根据分数由高到低排列，分数越高意味着和查询语句越相关。此外，我们还创新型加入了权重搜索模式，将用户搜索词与商品名称、商品详情、商品类别等不同元素赋予不同的权重，进行综合打分，返回用户一个更加精准的结果。

【举例】当用户搜索“星巴克咖啡”，带有“星巴克咖啡”的信息就要比只包含“咖啡”的信息靠前。那么到底什么是「相关性匹配」，什么才是「人的思维」呢？

比如我搜「莎士比亚」，我要的肯定不只是精确包含「莎士比亚」的文稿，我可能还要搜「莎翁」、「Shakespeare」、「哈姆雷特」、「罗密欧和朱丽叶」、「威尼斯的商人」…

又比如我输错了，输成「莎士笔亚」，「相关性匹配」可以智能的帮我优化为「莎士比亚」，返回对应的搜索结果。

（3）我们小组还在Elasticsearch中加入了**IK分词器**，更能满足中文用户的搜索需求。下图为搜索效果展示：

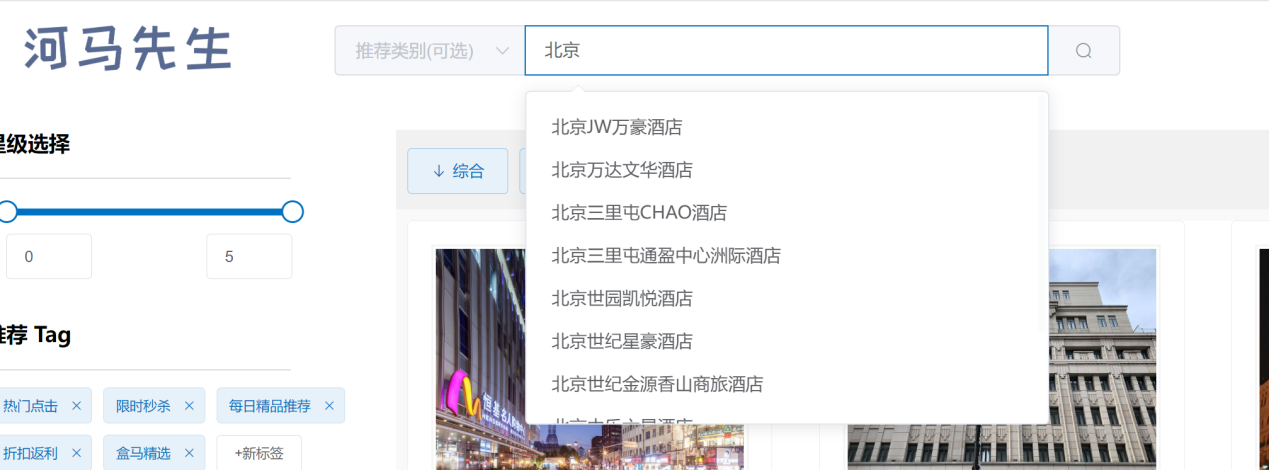


图3.2.6 搜索自动补全页面展示

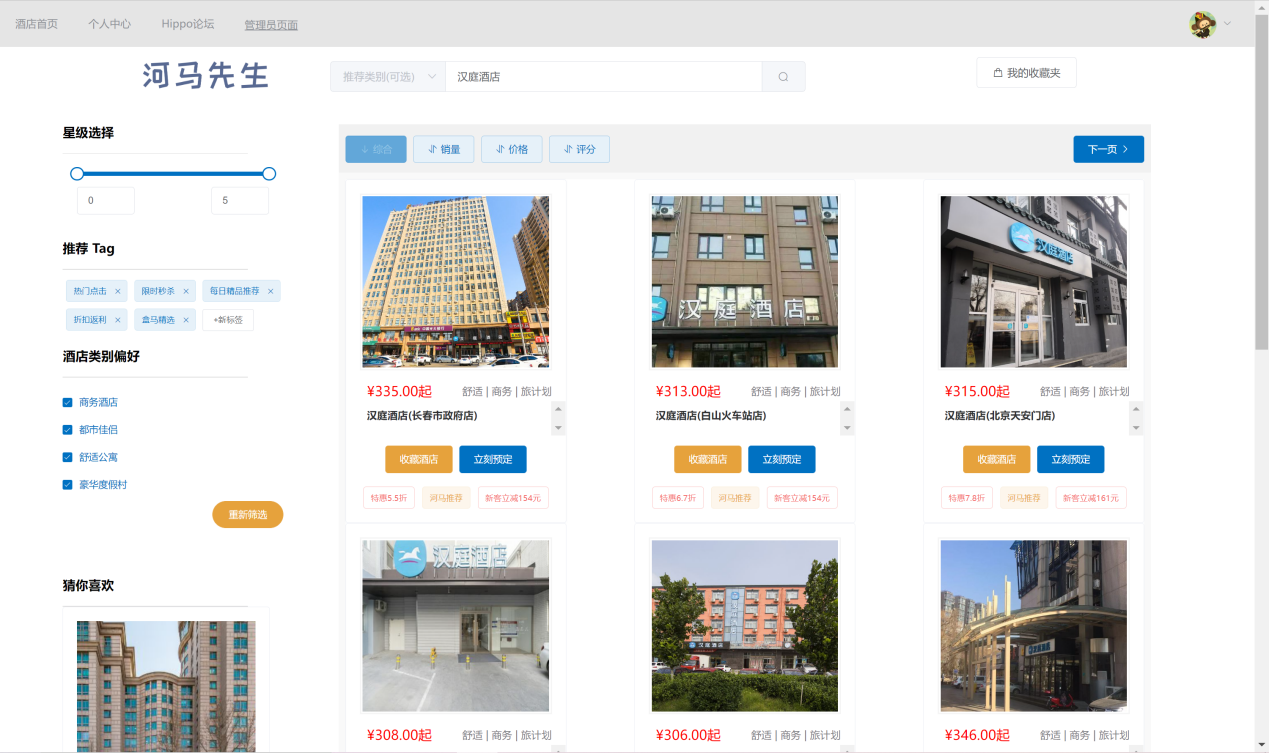


图3.2.7 搜索内容页面展示

### 3.2.6 邮件服务——JavaMailSender

从1969年10月世界上的第一封电子邮件发出，到2021年，已经过去将近半个世纪了。虽然即时通讯和视频会议，甚至全息投影都变得日益普及，但电子邮件依然有着广泛的使用场景和不可撼动的历史地位。

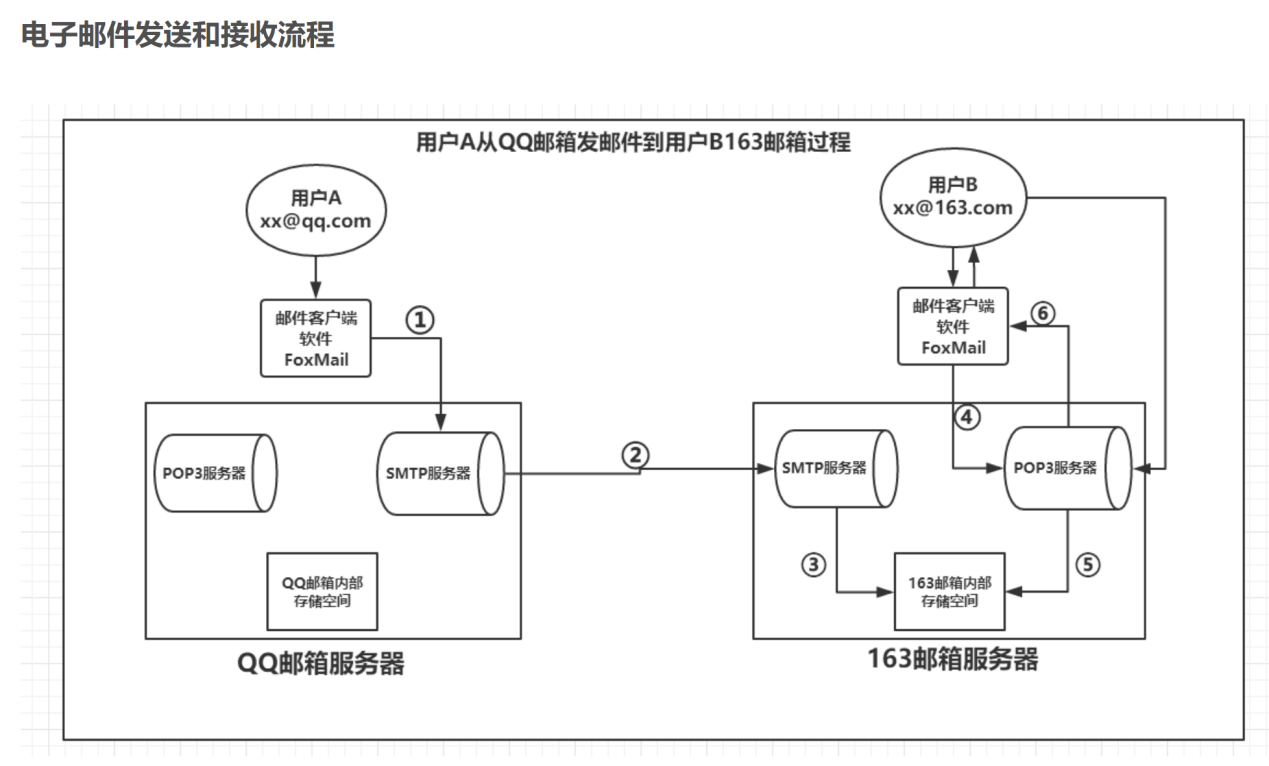


图6.2.8 电子邮件发送和接收流程

在项目中使用邮件功能的配置过程如下：

（1）QQ邮箱开通POP3/SMTP服务。

（2）Spring Boot 引入 maven。

（3）Spring Boot 配置yml文件，添加相关邮箱信息。

（4）Service层实现业务逻辑，Controller层调用接口。

**邮件服务在本项目中的应用：**为了确保用户邮箱的真实性，在用户绑定邮箱时，我们使用邮箱验证码来进行验证。

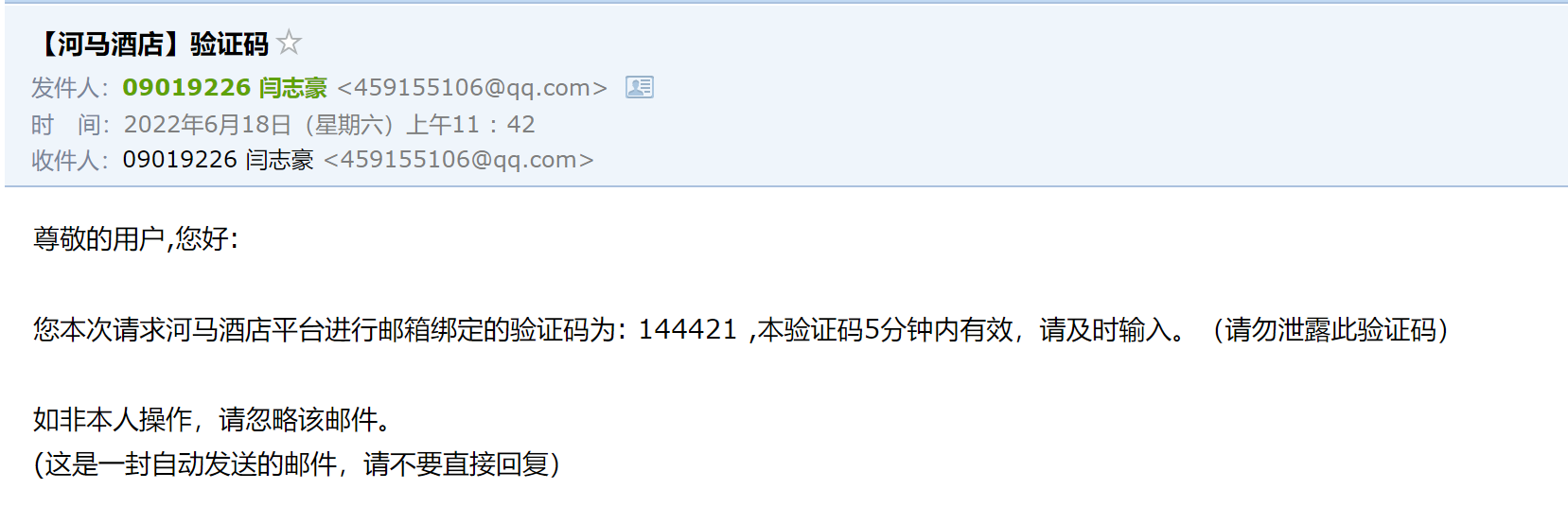


图6.2.9 邮件服务功能实例

### 3.2.7 第三方支付接口——支付宝 alipay easysdk

支付是商品购买中一个必不可少的流程。作为生鲜购物平台，核心业务以卖家与买家之间的交易为主，利用支付宝第三方接口，使项目开发产品功能完备。

在Spring Boot框架中使用支付宝接口的**配置过程**如下：

（1）使用支付宝提供的密钥生成工具生成公钥和私钥；

（2）在支付宝开发者平台开发服务中的研发服务找到沙箱环境，配置公钥；

（3）Spring Boot 中导入 alipay-easysdk maven依赖；

（4）创建config文件进行相关配置；

（5）编写支付和回调业务逻辑层与控制层；

由于支付宝开发平台申请使用正式环境需要商家营业执照，我们选择了全真模拟真实开发环境的支付宝沙箱开发环境，通过支付宝官网提供的开发流程，介入支付宝接口，可以很好的体现出项目的完整性。



图6.2.10 支付宝沙箱环境

### 3.2.8 统一的异常处理

由于我们采用的是前端和后端分离，统一的异常处理显得尤为重要，可以给前端返回一个明确的信息。如果不使用统一的异常处理，当后台出现了异常，直接给前端抛一个500错误，这种错误会被用户看到非常不友好，而我们的统一异常处理就是把所有的异常都经过规范处理，前端可以收到可读性极强的信息，并且符合项目规范。

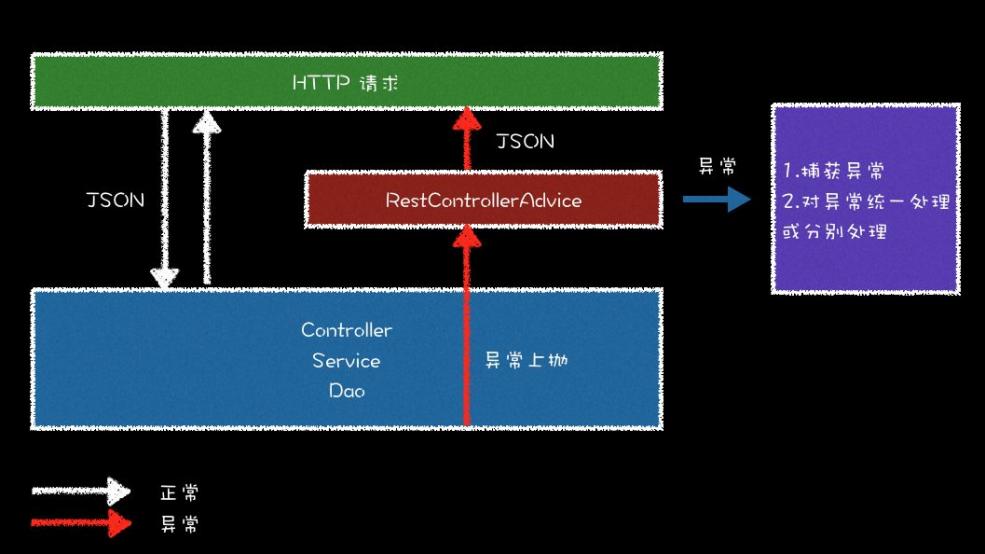


图6.2.11 异常处理步骤

在本项目中，用户注册时异常处理实例如下：



图6.2.12 用户注册异常处理API文档

### 3.2.9 统一的日志记录——SLF4J和Logback日志框架

SLF4J是简单的日志外观模式框架，抽象了各种日志框架例如Logback、Log4j、Commons-logging和JDK自带的logging实现接口。它使得用户可以在部署时使用自己想要的日志框架。SLF4J是轻量级的，在性能方面几乎是零消耗的。

Logback是Log4j的改进版本，而且原生支持SLF4J（因为是同一作者开发的），因此从其它日志框架如Log4j或JDK的logging迁移到Logback是完全可行的。

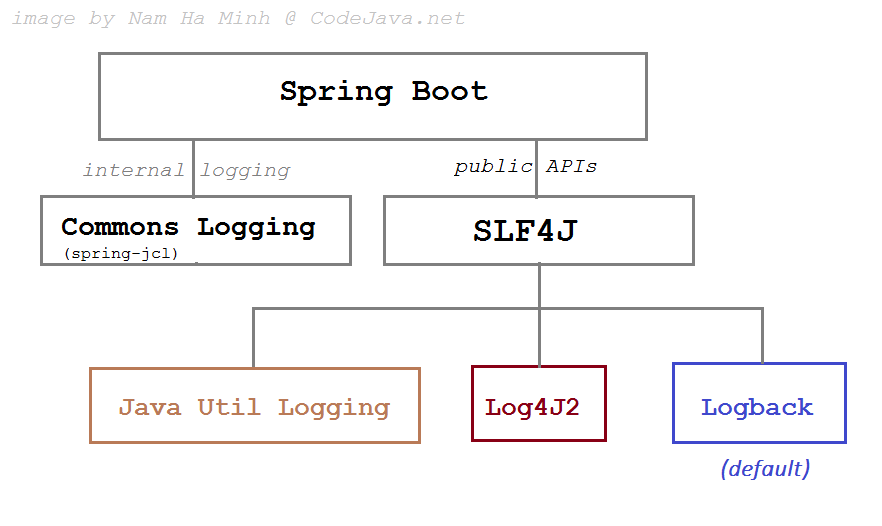


图6.2.13 SLF4j日志框架

### 3.2.10 手机短信服务

短信服务（Short Message Service）是广大企业客户快速触达手机用户所优选使用的通信能力。调用API或用群发助手，即可发送验证码、通知类和营销类短信；本项目选择使用阿里云短信服务，国内验证短信秒级触达，到达率最高可达99%；国际/港澳台短信覆盖200多个国家和地区，安全稳定，广受出海企业选用。

本项目中使用短信服务的场景有：用户注册、用户手机短信验证码登录。



图6.2.14 用户注册页面

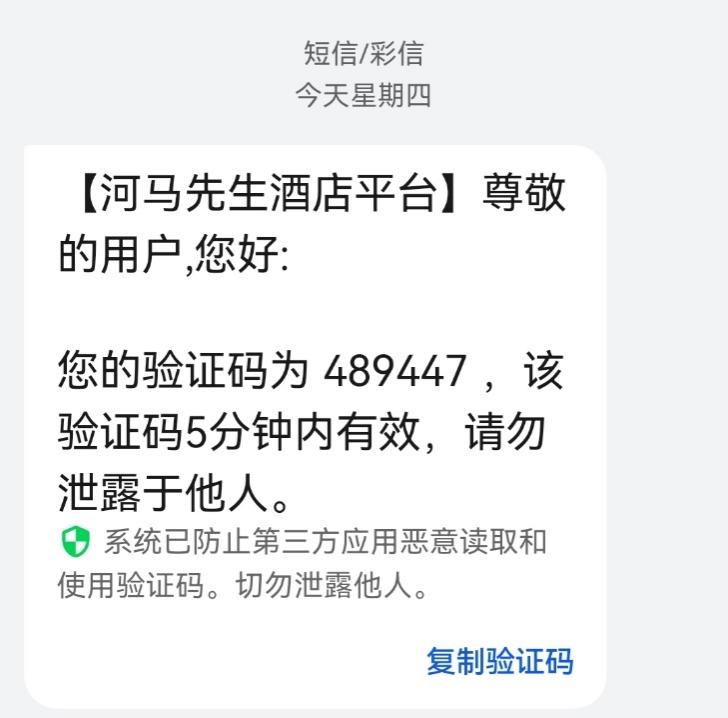


图6.2.15 手机短信验证码

### 3.2.11 高安全性能

**（1）验证码——Secure Random随机数**

通常采取的随机数生成方法是Math.random，它生成的是一般随机数，采用的是类似于统计学的随机数生成规则，其输出结果很容易预测，因此可能导致被攻击者击中。

而SecureRandom是真随机数，采用的是类似于密码学的随机数生成规则，其输出结果较难预测，若想要预防被攻击者攻击，最好做到使攻击者根本无法，或不可能鉴别生成的随机值和真正的随机值。所以我们本项目采用了SecureRandom随机数确保安全性能。

**（2）密码加密——BCryptPasswordEncoder**

spring security中的BCryptPasswordEncoder方法采用SHA-256 +随机盐+密钥对密码进行加密。SHA系列是Hash算法，不是加密算法，使用加密算法意味着可以解密，但是采用Hash处理，其过程是不可逆的。

在本应用中充分考虑安全性能，采用不可逆的SHA Hash算法加密，即使数据库泄露，也不会造成密码泄露，确保用户安全。

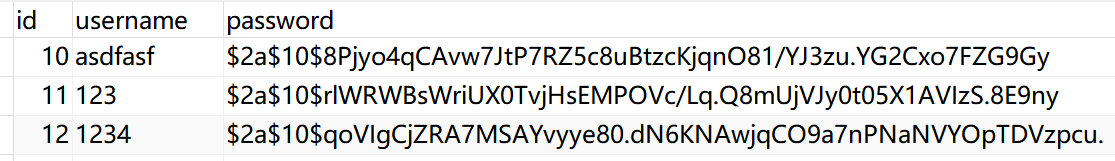


图6.2.16 数据库密码存储