# 实验7 综合实践

## 7.1实验目的

（1）掌握基于B/S架构的数据库管理系统设计与开发方法。  
（2）熟练应用触发器、存储过程等数据库高级技术实现业务逻辑。  
（3）培养团队协作能力及系统文档编写与项目管理实践技能。

## 7.2实验平台

（1）操作系统：Windows；

（2）编程语言：Java；

（3）脚本语言：JavaScript/Java；

（4）数据库：MongoDB；

（5）框架：SpringBoot

## 7.3 实验任务与基本要求

### 7.3.1 实验内容

设计并实现一个基于B/S或C/S架构的数据库管理系统。DBMS可以从SQL Server、Oracle或MySQL中做选择。开发的管理系统应包含以下功能：

(1) 每个模块必须实现检索、新增、编辑、删除功能；特定模块需支持批量删除操作。

(2) 系统应包含权限控制功能：权限配置需具备动态性，程序应内置用户管理及用户权限管理模块，以便灵活管理用户对各个模块、子模块或具体功能的访问权限。

(3) 至少应运用触发器或存储过程中的一种技术，以增强系统的功能性和效率。

(4) 界面设计应满足基本的人机交互标准，力求简洁易用，并具有一定的美观度。

(5) 数据库中的每张表都必须设置主键，以确保数据的唯一性和完整性。

### 7.3.2. 实验题目

**邦多利大酒店管理系统**

酒店管理系统整合了与酒店入住信息相关的各项管理功能。该系统为两类用户——客户和系统管理员提供服务。每类用户模块包含以下核心功能：

数据库应至少包含以下表格：

1. 入住登记表
2. 入住服务关联表
3. 客户信息表
4. 会员信息表
5. 酒店房间表
6. 酒店服务表
7. 酒店服务请求表
8. 管理员模块

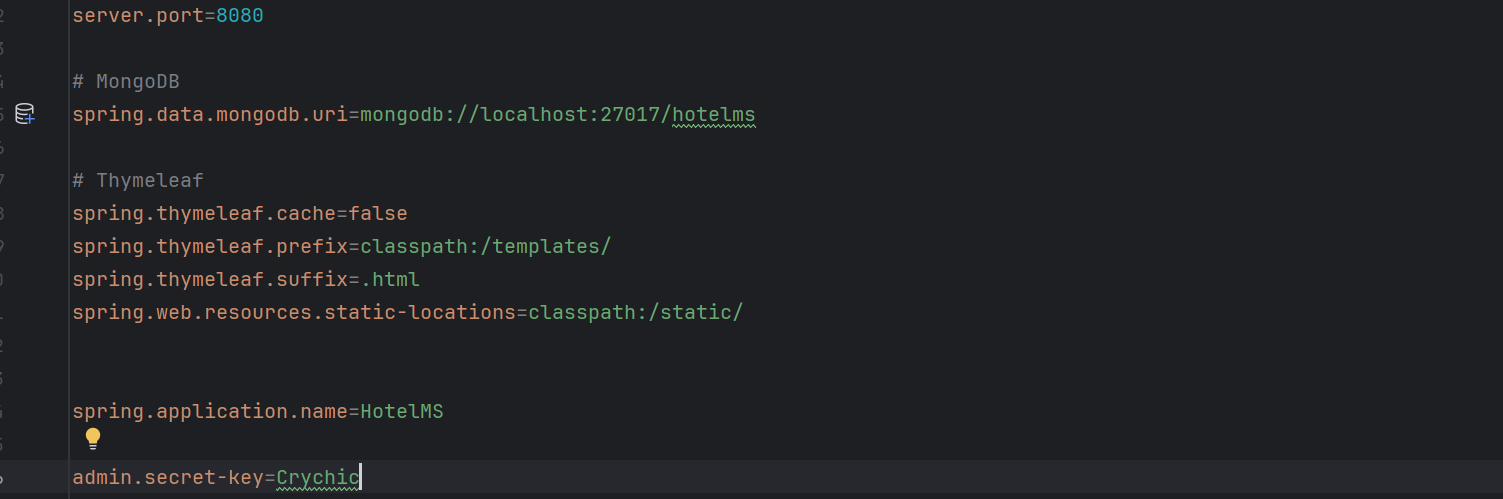
* 负责客户信息的添加、移除、更新及查询操作；
* 负责客房信息的添加、移除、更新及查询操作；
* 负责酒店服务的添加、移除、更新及查询操作；
* 负责会员信息的添加、移除、更新及查询操作；
* 管理客户入住信息；
* 客户历史入住信息；
* 监控酒店客户入住信息；

1. 客户模块

* 酒店入住信息填写；
* 酒店服务申请；

### 7.3.3. 代码实现

**1、环境依赖（application.properties）**

****

网页配置在本地端口8080，同时连接本地mongoDB数据库配置端口27017，使用端口下的数据库hotelms，并设置应用名称为 HotelMS，设置管理员密码为Crychic。

**2、表的实现**

**2.1、定义三个类型**

**2.1.1、性别类型**

public enum Gender {  
 *MALE*("男性"),  
 *FEMALE*("女性"),  
 *OTHER*("其他");  
 private final String description;  
 Gender(String description) {  
 this.description = description;  
 }  
 public String getDescription() {  
 return description;  
 }  
}

**2.1.2、会员类型**

public enum MemberLevel {  
 *NONE*("非会员"),  
 *BRONZE*("铜牌会员"),  
 *SILVER*("银牌会员"),  
 *GOLD*("金牌会员"),  
 *PLATINUM*("白金会员");  
 private final String description;  
 MemberLevel(String description) {  
 this.description = description;  
 }  
 public String getDescription() {  
 return description;  
 }  
}

**2.1.3、房间类型**

public enum RoomType {  
 *SINGLE*("单人间"),  
 *DOUBLE*("双人间"),  
 *SUITE*("套房"),  
 *DELUXE*("豪华间");  
 private final String description;  
 RoomType(String description) {  
 this.description = description;  
 }  
 public String getDescription() {  
 return description;  
 }  
}

**2.2、定义入住信息表**

入住信息表（入住ID，客户ID，客户姓名，客户身份证号，客房ID，客房号，实际入住日期，预计离店日期，实际离店日期，服务项列表，房间费用，服务总费用，总费用，住房状态标示符）

入住ID，主键，在表中插入新项时由MongoDB自动生成

客户ID，外键，与客户表中的客户ID相关联

客户姓名，避免频繁联表查询，记录入住时的客户姓名

客户身份证号，外键记录入住时客户的身份证号，用于公安备案、快速核验

客房ID，外键，与客房表中的客房ID相关联，关联客房信息

客房号，外键，记录入住时的房间号

实际入住日期，标记客户实际入住时间

预计离店日期，客户预订的离店时间，用于生成初始账单和安排客房清洁

实际离店日期，客户真实离店时间，用于计算最终房费和超时费用

服务项列表，存储客户在住期间消费的服务

房间费用，根据 **入住天数 × 房价** 计算的总房费

服务总费用，自动计算服务项列表中所有服务项的小计总和

总费用，代表客户应付的全额费用

住房状态标示符，用于快速筛选在住客户

**2.3、定义入住服务关联表（作为入住信息表的内嵌对象）**

入住服务关联表（服务关联标识，服务名，服务次数，服务单价，服务总费用）

服务关联标识，外键，关联服务信息表，用于追溯服务详细信息

服务名称，外键，记录入住时的服务名称

服务数量，客户消费该服务的次数

单价，记录消费的服务单

服务总费用：表示该服务项的总费用

**2.4、定义客户信息表**

客户信息表（客户ID，客户姓名，性别，身份证号，联系方式，会员标示）

客户ID，主键，MongoDB自动生成的唯一标识。

客户姓名，用于标识客户身份

性别标识，标注客户性别

身份证号，作为核心身份办理入住，为公关备案，确定唯一性

联系方式，存储手机号/邮箱等

会员标识，关联会员系统享受权益

**2.5、定义会员信息表**

会员信息表（会员ID，身份证号，注册日期，会员等级）

会员ID，主键，MongoDB自动生成的唯一标识，用于系统内部管理会员记录

身份证号，确定唯一性，可以关联客户信息表

注册日期，记录会员加入时间

会员等级，确定会员享受的福利

**2.6、定义酒店房间表**

酒店房间表（客房ID，客房号，客房价格，客房状态，客房类型）

客房ID，主键，MongoDB自动生成的唯一标识，用于系统内部管理客房记录。

客房号，作为入住登记、清洁管理的依据，确定唯一性

房间单价，记录房间的价格

客房状态，了解目前酒店入住信息

客房类型，记录酒店房间类型

**2.7、定义酒店服务表**

酒店服务表（服务项目ID，服务名称，服务单价）

服务项目ID，主键，MongoDB自动生成的唯一标识，用于系统内部管理服务项目。

服务名称，确保唯一性，显示服务

服务单价，记录服务的价格

**2.8、定义酒店服务请求表**

酒店服务请求表（ID，入住ID，身份证号，房间号码，服务项目ID，服务名称，服务数量，服务价格，请求时间，请求状态，管理员备注，完成时间）

ID，作为主键，MongoDB自动生成的唯一标识，用于追踪服务请求生命周期。

入住ID，外键，关联当前入住记录

身份证号，外键，确定服务目标身份

房间号码，外键，确定服务地点

服务项目ID，外键，关联服务基础信息

服务名称，外键，存储服务名称

服务数量，记录客户请求的数量

服务单价，记录请求时的服务价格

请求时间，记录客户发起请求时间

请求状态，记录当前请求的完成状况

管理员备注，备注管理员信息

完成时间，记录服务实际完成时间

**3.功能实现**

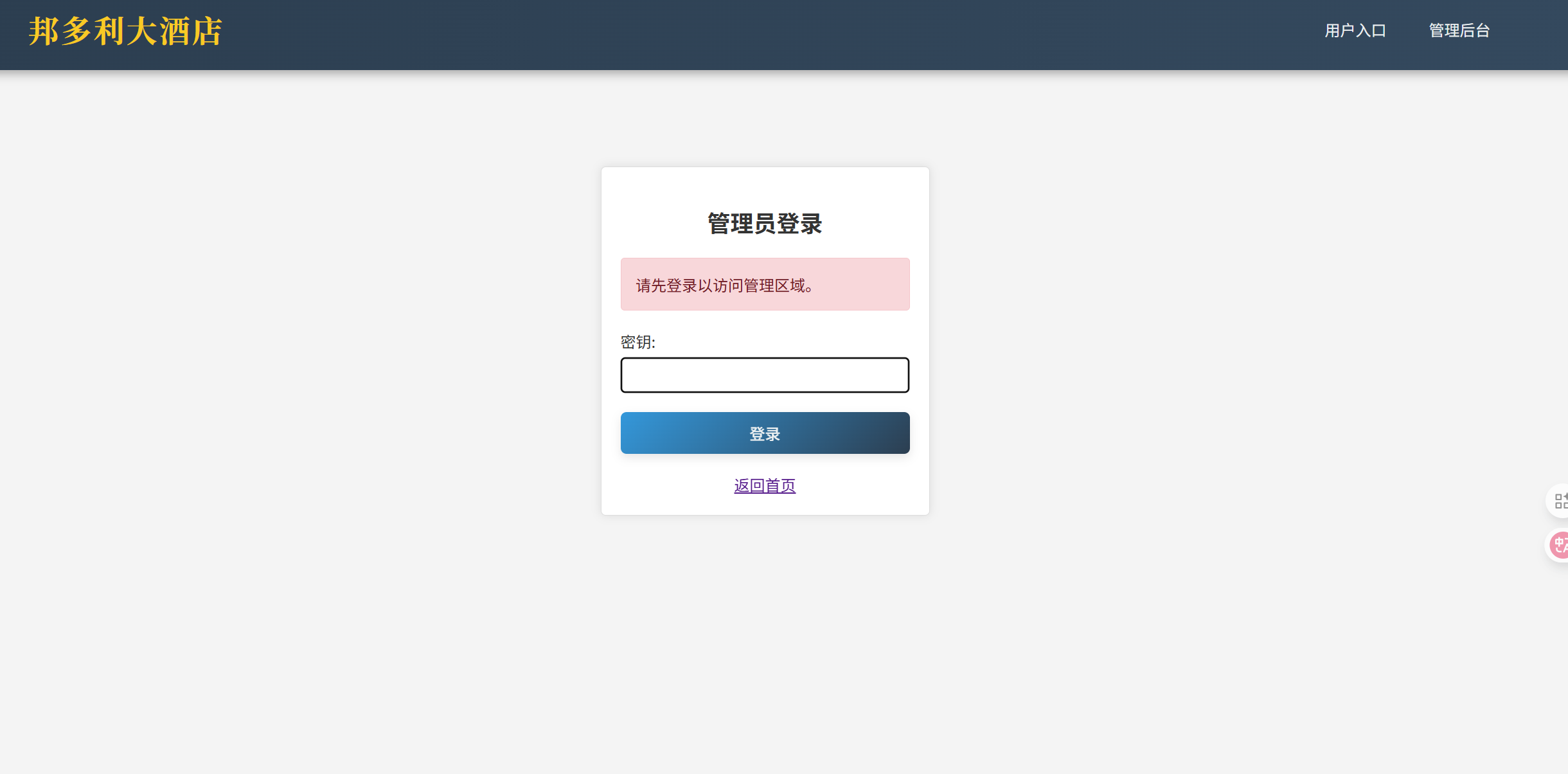
**3.1主界面视图**

****

**3.2用户视图**

****

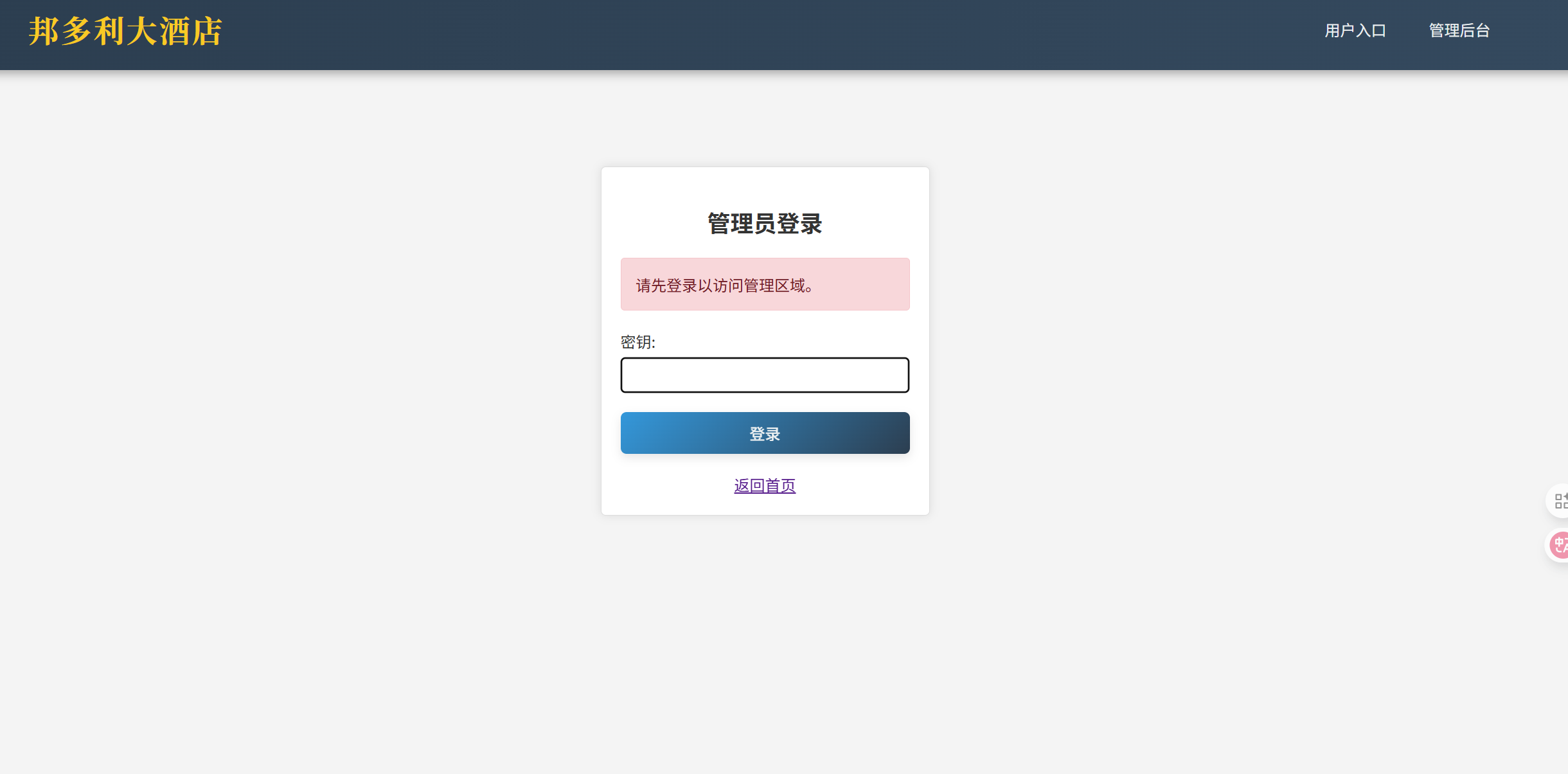
**3.3管理者视图**

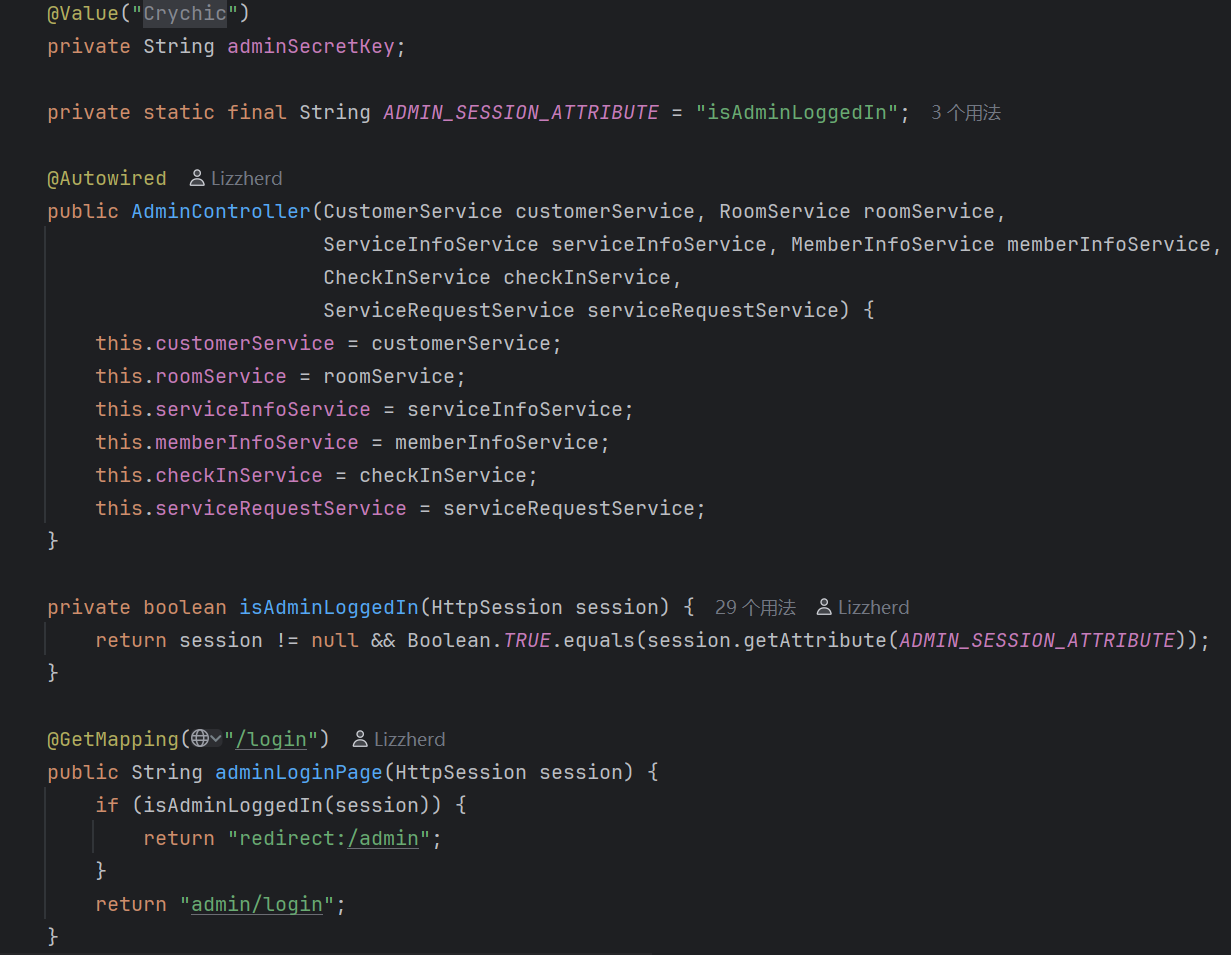
****

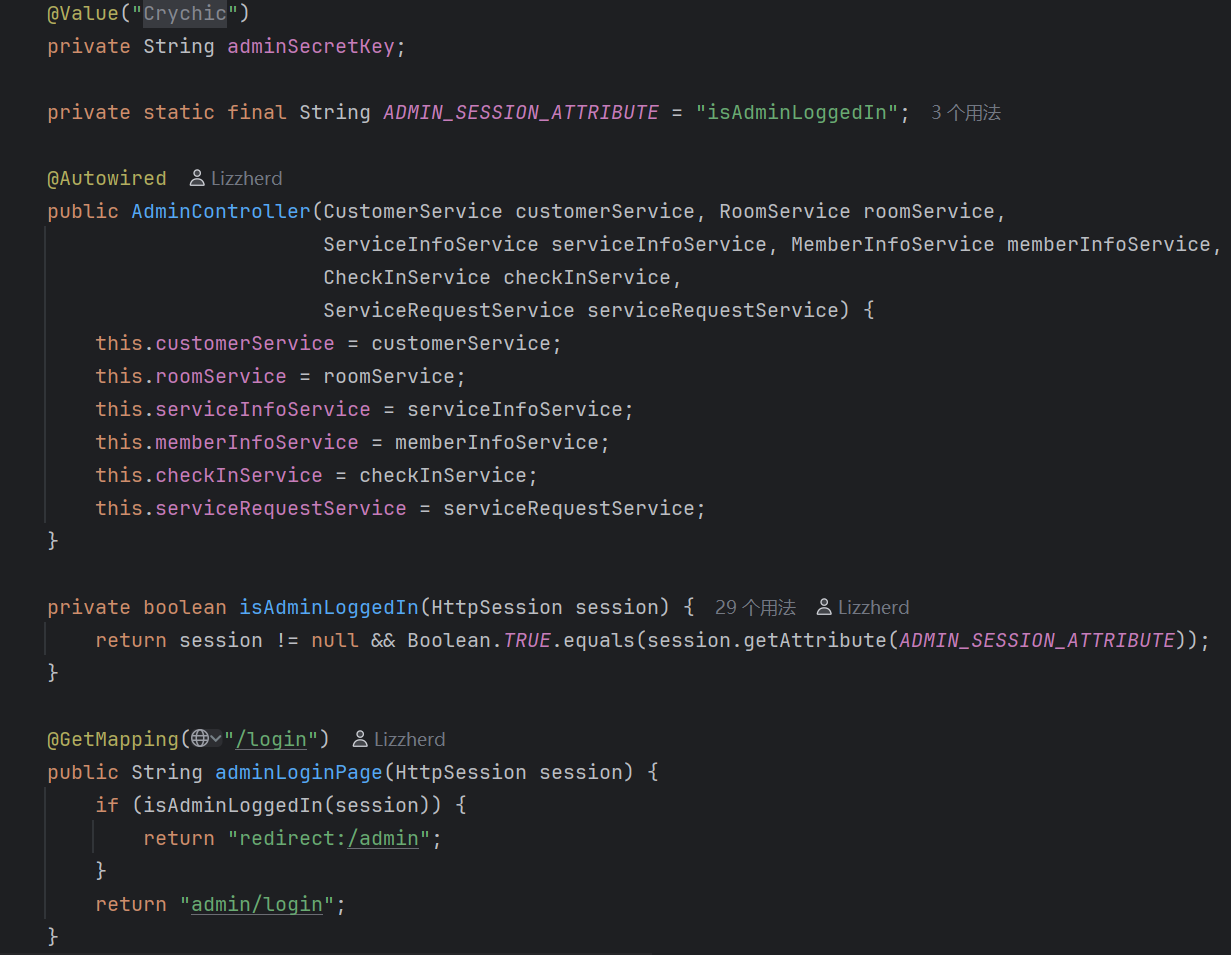
****

进入管理员视图，首先需要输入密码（密码设置为：Crychic），密码通过之后得到后续管理员视图，可以管理酒店信息。

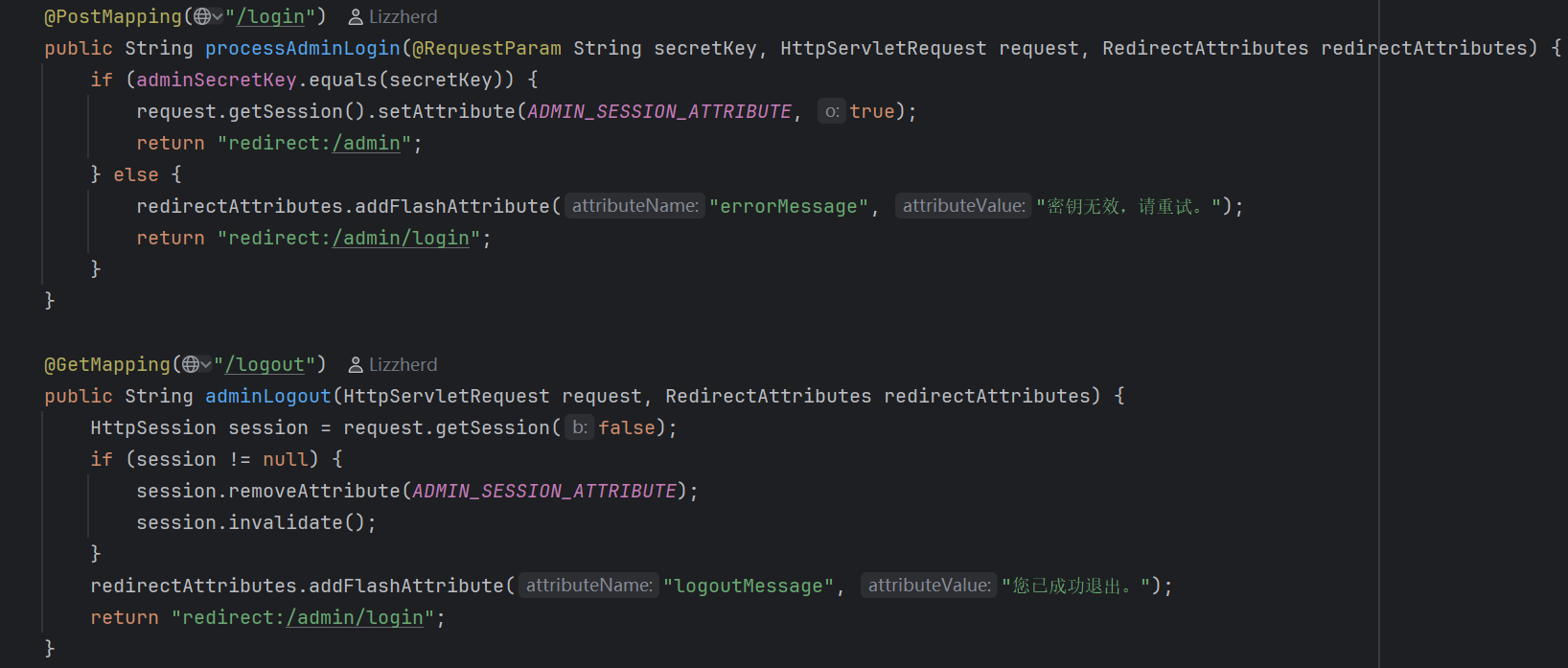
**3.4管理员视图的登录设置**

****





首先定义isAminLoggedIn函数，判断管理员是否在登录状态，若是，则进入admin界面，反之，则重新回到登录界面。



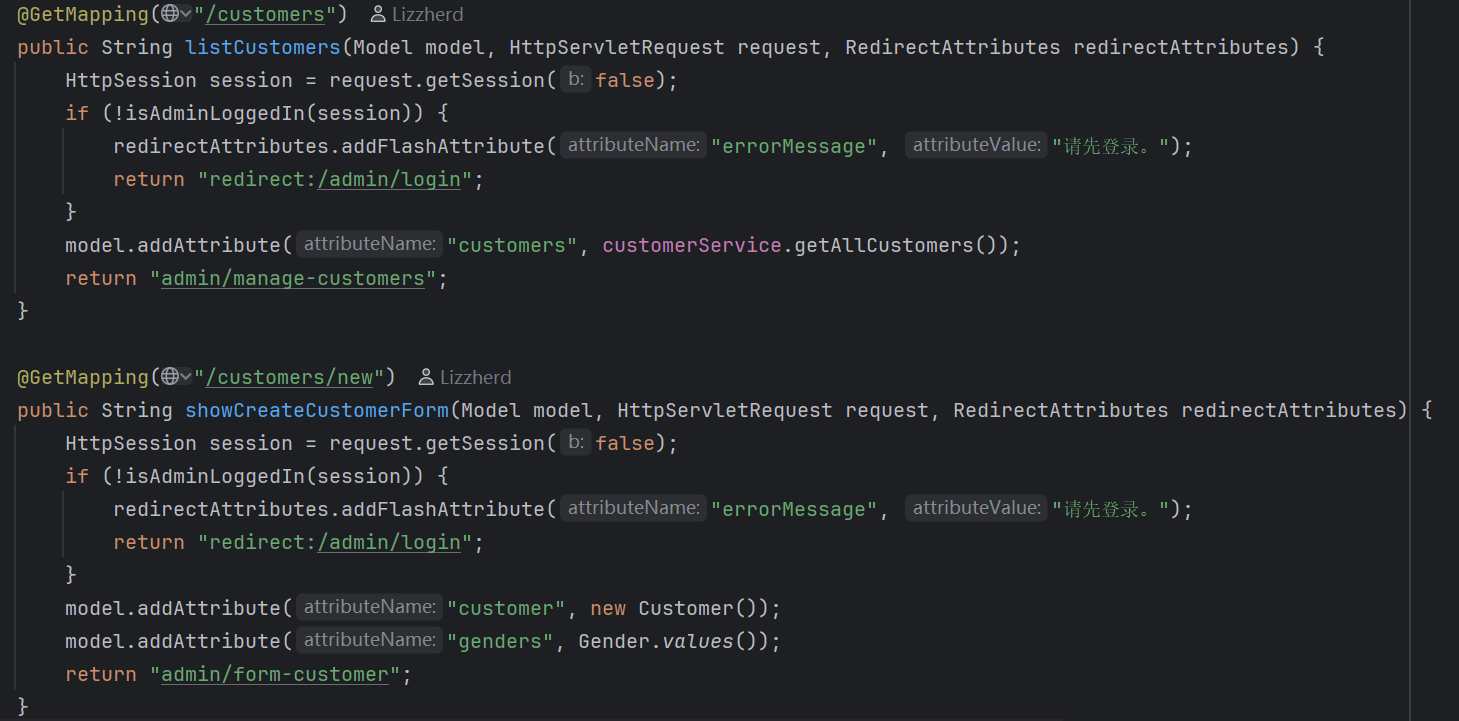
登录过程：用户访问login界面，如果已经登录，则转移到admin界面；否则显示登录页面。在登录界面，用户提交密匙，然后将其与配置的adminSecretKey进行比较。如果匹配，则设置ADMIN\_SESSION\_ATTRIBUTE为true，然后重定向到admin界面；如果不匹配，则回复错误消息（"密钥无效，请重试。"），并重定向回登录页面。

登出过程：用户访问logout界面，如果对话界面存在，移除ADMIN\_SESSION\_ATTRIBUTE，并销毁会话。设置登出成功消息（"您已成功退出。"），然后重定向到登录页面。

**3.5管理员界面的用户管理系统**

****

****



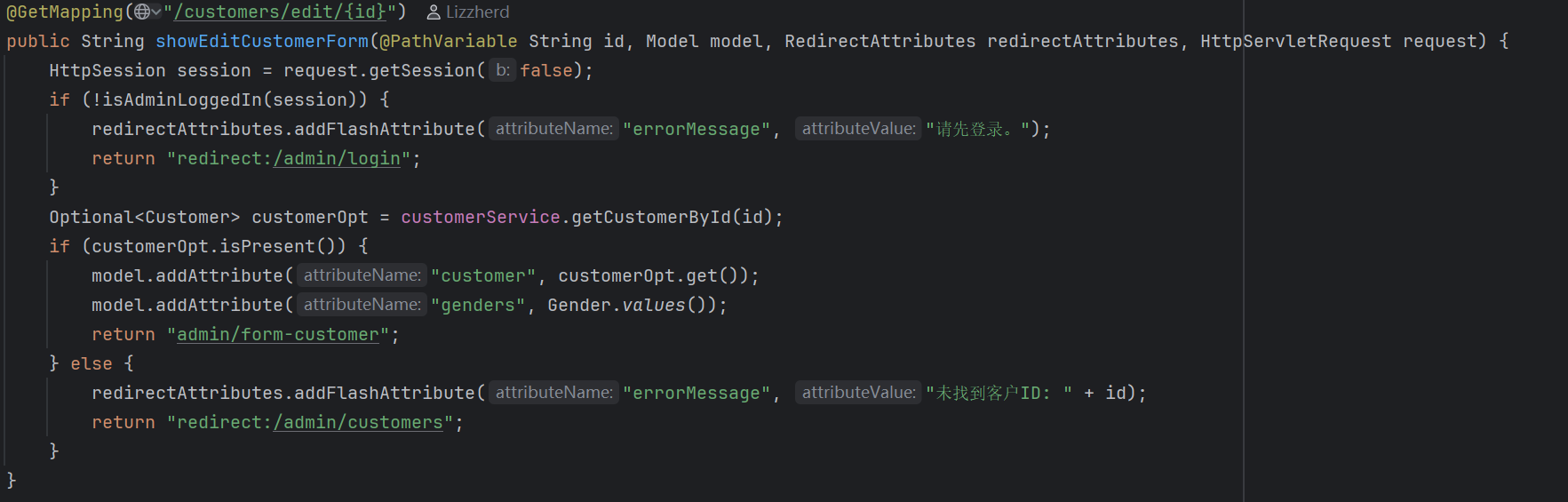
每次切换页面时，检查是否处于登录状态，若没有，则回到登录界面。点击**添加新客户**时，提供form-customer表单供用户输入，当用户输入完成之后，点击**创建客户**。



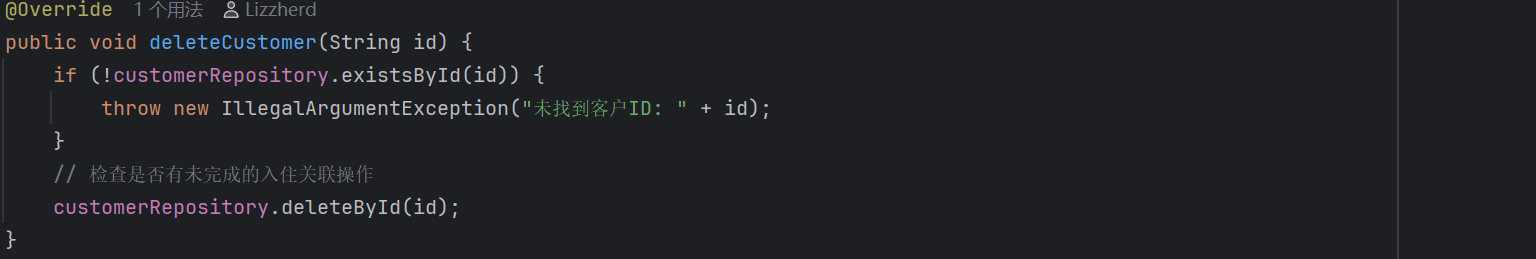
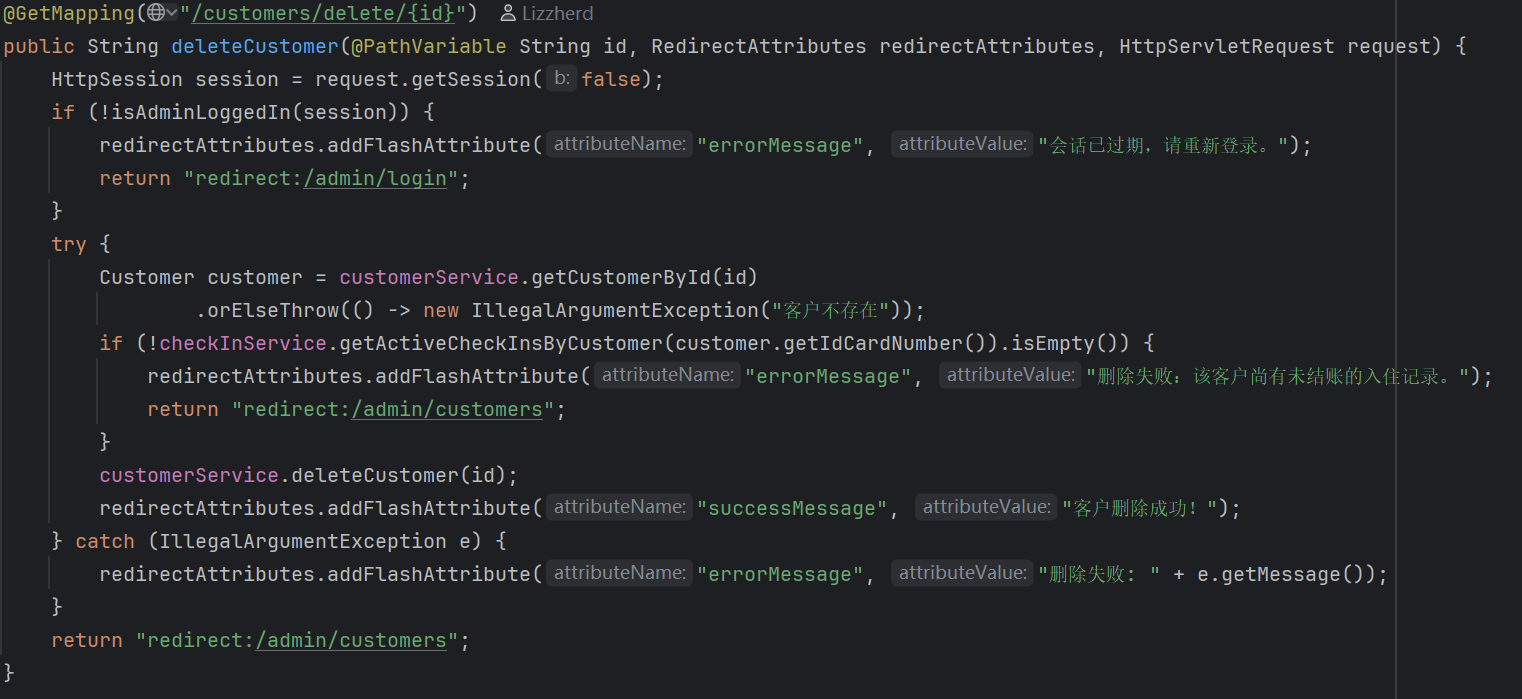
创建触发器，在**创建客户**时启动，首先检测身份证号是否存在，若不存在，则运行service层CustomerServicelmpl.java中的createCustomer函数，在mongoDB中创建实例；若存在，则返回（“创建失败，身份证号XX已存在”）；若姓名重复则不作处理，确保实际生活中存在的重名的人。





点击**编辑**，可以编辑客户消息，首先判断是否处于登录状态，确认登录之后，检测是否存在该用户id，若存在，返回一个id非空的表单，使其id无法更改，然后更改消息，点击**更新用户**，更新函数依赖于saveCustomer，若customer网页存在ID，则显示更新用户，执行service层CustomerServicelmpl.java中的updateCustomer函数，在mongoDB中更新实例；若不存在，则回复（“未找到客户ID”）。



 点击**删除**，首先判断是否处于登录状态，确认登录之后，检测是否存在该用户id，若不存在，删除失败，回复（“客户不存在”）：若存在，则检测其是否存在账单，若存在，则回复（“删除失败：该客户尚有未结账的入住记录。”）；若不存在，执行service层CustomerServicelmpl.java中的deleteCustomer函数，在mongoDB中删除实例。

**3.5管理员界面的客房管理系统**

## 7.4 分组与提交文档要求

以小组形式进行项目开发，每组成员限定为2至3人，组内成员将共享相同的分数。评分标准将依据实验的完成度以及题目的难度。

本实验的最终提交截止日期为\*\*\*\*，逾期提交将被自动判定为0分；同时，抄袭或直接从网络复制的实验内容也将得到0分。

所有小组需以小组为单位提交最终文档。每个小组应提交一份最终源代码和一份实验报告（实验报告中的个人总结部分要求每位成员都需独立完成），所有文件需打包并命名为“组长学号\_姓名\_实验7”的压缩文件（例如2302011220001\_李丽丽\_实验7）。提交内容应包括：

1. 程序源代码
2. 实验报告（文档格式自选，但必须包含以下内容，并以word或pdf格式提交）
   1. 所选开发环境、使用的工具、编程语言、数据库；
   2. 系统需求分析；
   3. 系统详细设计说明：包括整体流程图、总体设计图、各模块设计图等；
   4. 系统数据库设计与SQL脚本：（需包含E-R图和物理数据模型；包括创建表、视图、触发器、存储过程的SQL语句）；
   5. 系统实现细节：程序与数据库的连接说明、界面设计及其操作指南；
   6. 小组成员分工与评分：列出所有小组成员的学号、姓名及分工情况；提供组内评分细节，并解释每位成员获得相应评分的理由；
   7. 个人总结：（每位成员需提交一段100至200字的实验总结，并注明总结人）；

## 7.5参考图

物理数据模型图、系统流程图和模块设计图可参考如下。

