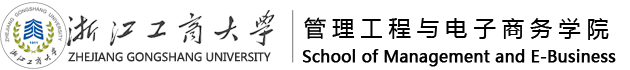
****

**《数据库应用》课程设计报告**

**专 业：信息管理与信息系统**

**班 级：数创2201**

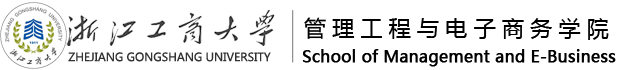
**学 号：2235030130**

**姓 名：曹祥瑞**

**题 目：酒店信息管理系统**

**指导教师：张寅升**

**2024年6月**



**文件编号： 学号**

《酒店信息管理系统》课程设计报告

|  |  |  |
| --- | --- | --- |
| 文件状态：  [ ] 草稿  [√] 正式发布  [ ] 正在修改 | 文件标识： | ZJGSU-SME-DB |
| 当前版本： | 1.0 |
| 作 者： | 曹祥瑞、周星鑫、杜梦铫 |
| 完成日期： | 2024-06-30 |

浙江工商大学管理工程与电子商务学院

**目录**

[一、系统需求分析 4](#_Toc170685031)

[二、概念结构设计 6](#_Toc170685032)

[三、逻辑结构设计 8](#_Toc170685033)

[四、物理结构设计 10](#_Toc170685034)

[五、视图设计 14](#_Toc170685036)

[六、数据保护设计 16](#_Toc170685037)

[6.1数据库密码安全保护 16](#_Toc170685038)

[6.2角色与权限 17](#_Toc170685039)

[七、系统实现 19](#_Toc170685040)

[7.1数据库连接 19](#_Toc170685041)

[7.2功能界面设计 20](#_Toc170685042)

[八、总结 24](#_Toc170685043)

[8.1问题解决 24](#_Toc170685044)

[8.2解决方式 26](#_Toc170685045)

# **一、系统需求分析**

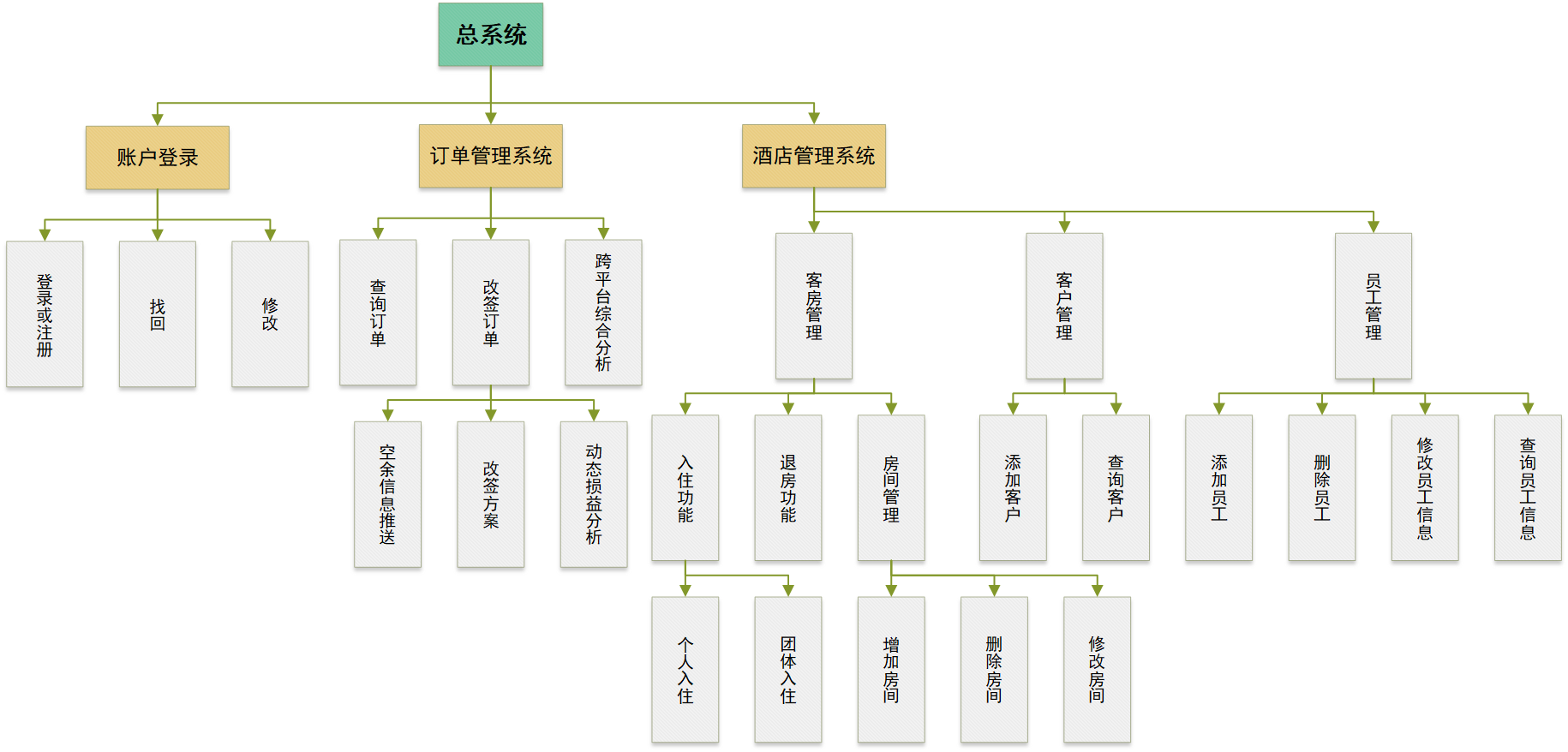
在行业竞争日趋激烈的今天，酒店业急需通过高效的管理手段来提升服务质量和经济效益。针对当前市场痛点，如特种兵式抱团旅游带来的人数众多难以安排的问题，以及大学生等消费群体经济能力有限导致的选择困境，我们提出了基于多用户协同的酒店优化方案。

该方案旨在通过信息资源的深入开发和广泛利用，解决酒店性价比功能缺失的问题，并消除各大平台间的信息差，从而提高酒店管理的效率和水平，实现经济效益的最大化。

我们的方案还能助力酒店实现服务信息化升级，提升顾客在酒店选择查询时的多样化需求体验，高效、快捷地帮助顾客了解酒店详情，提供最优选择方案。同时，这也符合国家对星级酒店计算机管理系统的要求，推动酒店管理向智能化、规范化转变。

此外，我们的系统还将整合各大平台信息，消除信息差，确保数据库操作人员能够轻松比较不同平台的优惠和服务，实时调整价格策略。实现经济效益最大化，并增强酒店的竞争力。

特别是订单管理功能，我们将引入智能改签系统，以应对客户频繁的改签需求。通过优化算法，实现随着时间推移改签难度的动态调整，既满足客户需求，又降低酒店风险。

 图 1-1 酒店信息管理系统功能结构图

通过研究同学们旅游途中遇到的住宿问题以及与酒店之间产生的多样化需求与酒店反馈的差异的案例，我们为系统设计了三个主要模块：账户登录、订单管理系统、酒店管理系统。这些功能将计算机技术与现代酒店服务管理完美结合，旨在提升服务质量，减轻工作人员负担，并满足客户的多样化需求。

以下逐一介绍设计的功能：

**（一）账户登录**

登录或注册、找回和修改功能。用户可以通过注册或登录进入系统，找回功能帮助用户找回忘记的账号或密码，修改功能允许用户更新他们的账户信息。

**（二）订单管理系统**

订单管理系统包含查询订单、改签订单、跨平台综合分析、动态损益分析和空余信息推送功能。查询订单功能允许用户查看当前和过去的订单记录；改签订单功能帮助用户修改已预订的订单，并提供改签方案。跨平台综合分析和动态损益分析功能为管理层提供了有价值的数据分析支持，而空余信息推送功能可以实时通知用户关于空闲客房的信息。

**（三）酒店管理系统**

客房管理：客房管理模块包括入住功能、退房功能和房间管理。入住功能进一步分为个人入住和团体入住，允许个人或团队进行入住登记。退房功能帮助用户完成退房手续。房间管理功能包括增加房间、修改房间和删除房间，方便酒店管理人员管理客房资源。

客户管理：客户管理模块提供查询客户和添加客户功能。查询客户功能允许工作人员查看客户信息，添加客户功能帮助记录新客户的详细信息。

员工管理：员工管理模块包含查询员工信息、添加员工、删除员工和修改员工信息功能。查询员工信息功能帮助管理人员查看员工的基本信息和权限信息。添加员工功能允许录入新员工的信息；删除员工功能确保管理人员可以移除离职员工；修改员工信息功能方便更新员工的个人信息和权限。

这个系统设计通过各个模块的协同工作，实现了对酒店资源、客户信息和员工管理的全面控制和优化，提高了管理效率和服务质量。

# **二、概念结构设计**

通过对一般酒店管理信息系统的需求、内容和流程分析，设计如下实体及其属性：

员工（staff）：{员工编号 sid，员工姓名 sname，员工性别 ssex，登录密码 spassword，权限等级 srole，员工身份证号 sidcard，电话号码 sphone}

客房（room）：{客房编号 rid，客房类型 rtype，客房价格 rprice，客房描述 rdesc}

客户（client）：{客户编号 cid，组长 Leader，客户姓名 cname，客户联系方式 cphone，客户年龄 cage，客户性别 csex}

订单（hotelorder）：{订单编号 id，客户编号 cid，客房编号 rid，开房时间 start\_time，退房时间 end\_time，订单金额 money，订单时间 order\_time，评价 remark}

跨平台比价（compare）：{比价ID co\_id，客房编号 rid，比价日期 co\_date，平台名称 pname，平台价格 pprice，订单编号 id}

其中各实体间关系如下：  
- 员工（staff）与订单（hotelorder）的关系是 m:n。

- 订单（hotelorder）与客房（room）预订的关系是 1:n。

- 客户（client）与订单（hotelorder）的关系是 1:n。

- 跨平台比价（compare）与订单（hotelorder）的关系是 m:n。

根据实体、属性及实体之间的相互联系，我绘制了如下实体关系图（E-R 图）。

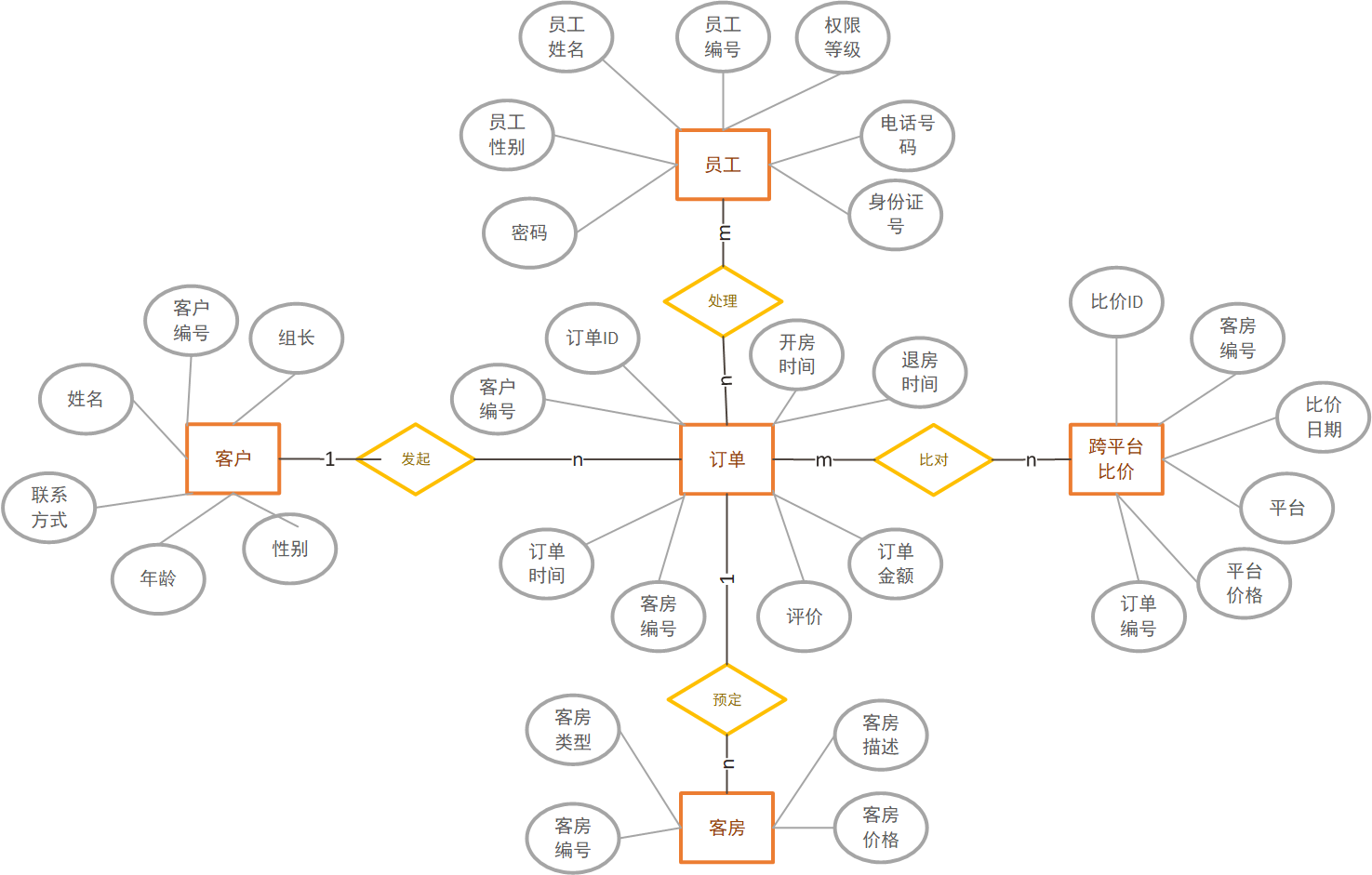


图 2-1 酒店信息管理系统 E-R 图

# **三、逻辑结构设计**

客户和订单的发起联系是 1：n，可以将它们之间的联系与 n 端实体订单合并；订单与客房的预订关系也是 1：n，同样可以将其中的联系与 n 端实体合并。而员工与订单之间的处理和跨平台比价与订单的登记联系方式是 n：m，将其联系转化称独立的关系模式。具体的基本 E-R 图向关系模式的转化如下所述：

**员工表（Employees）**

* 1. 员工编号（EmployeeID）：主键
  2. 员工姓名（EmployeeName）
  3. 员工性别（Gender）
  4. 密码（LoginPassword）
  5. 权限等级（AuthorityLevel）
  6. 员工身份证号（IDNumber）
  7. 电话号码（EmployeeNumber）

**客房表（Rooms）**

* 1. 客房编号（RoomID）：主键
  2. 客房类型（RoomType）
  3. 客房楼层（Floor）
  4. 客房价格（Price）
  5. 客房描述（Description）

**客户表（Customers）**

* 1. 客户编号（CustomerID）：主键
  2. 组长（Leader）：
  3. 客户姓名（CustomerName）
  4. 客户联系方式（ContactInfo）：
  5. 客户年龄（Age）：
  6. 客户性别（Gender）：

**订单表（Bookings）**

* 1. 订单ID（BookingID）：主键
  2. 客户编号（CustomerID）：外键，引用客户表内组长信息
  3. 客房编号（RoomID）：外键，引用客房表
  4. 开房时间（StartTime）
  5. 退房时间（EndTime）
  6. 订单金额（TotalAmount）
  7. 订单时间（BookingTime）
  8. 评价（Review）

**跨平台比价表（CrossPlatformPricing）**

* 1. 比价ID（PricingID）：主键
  2. 客房编号（RoomID）：外键，引用客房表
  3. 比价日期（PricingDate）
  4. 平台名称（PlatformName）
  5. 平台价格（PlatformPrice）
  6. 订单编号（OrderID）：外键，可能为空，引用订单表（如果比价与特定订单相关）

# **四、物理结构设计**

设计表结构。一般地，实体对应于表，实体的属性对应于表的列，实体之间的关系成为表的约束。逻辑设计中的实体大部分可以转换成物理设计中的表，但是它们并不一定是一一对应的。

表 4-1 数据库的表

|  |  |
| --- | --- |
| 表名 | 功能说明 |
| 员工表 | 记录酒店中员工相关信息 |
| 客房表 | 记录酒店客房的相关信息 |
| 客户表 | 记录入住客户的入住人员的信息 |
| 订单表 | 记录酒店客房的订单记录 |
| 跨平台比价表 | 记录酒店客房订单在不同平台的价格比对信息 |

表 4-2 员工表结构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 表名 | 员工表 | | | |  |
| 列名 | 数据类型  （精度范围） | 空/非空 | 约束条件 | 主码 | 外码 |
| 员工编号 | varchar(255) | NOT NULL |  | 是 |  |
| 员工姓名 | varchar(255) | NOT NULL |  |  |  |
| 员工性别 | varchar(255) | DEFAULT NULL |  |  |  |
| 密码 | varchar(255) | NOT NULL |  |  |  |
| 权限等级 | varchar(255) | NOT NULL |  |  |  |
| 员工身份证号 | varchar(255) | DEFAULT NULL |  |  |  |
| 电话号码 | varchar(255) | NOT NULL |  |  |  |
| 补充说明 |  | | | |  |

## 客房表

表 4-3 客房表结构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 表名 |  | 客户表 | |  |  |
| 列名 | 数据类型（精度范围） | 空/非空 | 约束条件 | 主码 | 外码 |
| 客房编号 | varchar(255) | NOT NULL |  | 是 |  |
| 客房类型 | varchar(255) | NOT NULL |  |  |  |
| 客房价格 | varchar(255) | NOT NULL |  |  |  |
| 客房描述 | varchar(255) | DEFAULT NULL |  |  |  |
| 补充说明 |  |  | |  |  |

表 4-4 客户表结构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 表名 |  | 客户表 | |  |  |
| 列名 | 数据类型（精度范围） | 空/非空 | 约束条件 | 主码 | 外码 |
| 客户编号 | varchar(255) | NOT NULL |  | 是 |  |
| 组长 | varchar(255) | NOT NULL |  |  |  |
| 客户姓名 | varchar(255) | NOT NULL |  |  |  |
| 客户联系方式 | varchar(255) | NOT NULL |  |  |  |
| 客户年龄 | varchar(255) | DEFAULT NULL |  |  |  |
| 客户性别 | varchar(255) | DEFAULT NULL |  |  |  |
| 补充说明 | 个人用户默认为组长 |  | |  |  |

表 4-5 订单表结构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 表名 | 订单表 | | | |  |
| 列名 | 数据类型（精度范围） | 空/非空 | 约束条件 | 主码 | 外码 |
| 订单编号 | varchar(255) | NOT NULL |  | 是 |  |
| 客户编号 | varchar(255) | NOT NULL |  |  | 客户.客户编号 |
| 客房编号 | varchar(255) | DEFAULT NULL |  |  | 客房.客房编号 |
| 开房时间 | date | DEFAULT NULL |  |  |  |
| 退房时间 | date | DEFAULT NULL |  |  |  |
| 订单金额 | varchar(255) | DEFAULT NULL |  |  |  |
| 订单时间 | date | DEFAULT NULL |  |  |  |
| 评价 | varchar(255) | DEFAULT NULL |  |  |  |
| 补充说明 |  |  |  | |  |

表 4-6 跨平台比价表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 表名 |  | 跨平台比价表 | |  |  |
| 列名 | 数据类型（精度范围） | 空/非空 | 约束条件 | 主码 | 外码 |
| 比价ID | varchar(255) | NOT NULL |  | 是 |  |
| 客房编号 | varchar(255) | NOT NULL |  |  | 客房.客房编号 |
| 比价日期 | date | NOT NULL |  |  |  |
| 平台名称 | varchar(255) | NOT NULL |  |  |  |
| 平台价格 | varchar(255) | NOT NULL |  |  |  |
| 订单编号 | varchar(255) | DEFAULT NULL |  |  | 订单.订单编号 |
| 补充说明 |  |  | |  |  |

# **五、视图设计**

当用户希望在数据库中快速访问特定字段组合，而不想频繁编写相同的查询语句时，可以创建视图来简化这一过程。视图允许用户定义感兴趣的字段集合，这些字段可能来自一个或多个表中。与实际存储数据的表不同，视图是一个虚拟表，它在数据库中仅保存了视图的查询定义，并不存储数据本身。视图的数据实际存储在它所关联的表中，因此，如果底层表的数据发生变化，视图中的数据也会相应更新。从这个角度来看，视图就像是一个透视镜，允许用户观察并跟踪数据库中他们关心的数据变化。每次访问或使用视图时，都会执行其定义的查询。

用户可以通过简单的 SELECT 语句从视图中检索数据。对于某些具有适当权限的视图，用户还可以使用 INSERT、UPDATE、DELETE 或 MERGE INTO 等语句来修改视图背后表中的数据。视图不仅提供了操作上的便利，还有助于保护数据库数据的安全性和完整性

因此，我们在数据库物理结构设计的基础上设计了如下视图：

**（1）查询员工信息的视图**

Create view EmployeesView (员工编号，员工姓名，员工性别，密码，权限等级，员工身份证号，电话号码)

As select 员工编号，员工姓名，员工性别，密码，权限等级，员工身份证号，电话号码

From 员工表

**（2）查询客房信息的视图**

Create view RoomsView (客房编号，客房类型，客房价格，客房描述)

As select 客房编号，客房类型，客房价格，客房描述

From 客房表

**（3）查询客户信息的视图**

Create view CustomersView (客户编号，是否组长，组长，客户姓名，客户联系方式，客户年龄，客户性别)

As select 客户编号，是否组长，组长，客户姓名，客户联系方式，客户年龄，客户性别

From 客户表

**（4）查询订单信息的视图**

Create view BookingsView (订单ID，客户编号，客房编号，开房时间，退房时间，订单金额，订单时间，评价)

As select 订单ID，客户编号，客房编号，开房时间，退房时间，订单金额，订单时间，评价

From 订单表

**（5）查询跨平台比价信息的视图**

Create view CrossPlatformPricingView (比价ID，客房编号，比价日期，平台名称，平台价格，订单编号)

As select PricingID, RoomID, PricingDate, PlatformName, PlatformPrice, OrderID

From CrossPlatformPricing

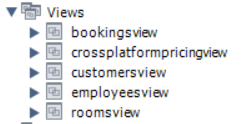


图 5-1 酒店信息管理系统视图

# **六、数据保护设计**

数据库的一大特点是数据可以共享，数据共享必然带来数据库的安全性问题，数据

库系统中的数据共享不能是无条件的共享，所以我们有必要进行数据库安全性设计。

## **6.1数据库密码安全保护**

在设计数据库时，确保数据安全和共享的合理性是至关重要的。以下是对数据库安全性设计的一个重新编写的版本，重点放在用户密码的掩码保护和唯一账号标识机制上：

数据库安全性设计

数据库作为信息共享的核心，其安全性设计是保护数据不被未授权访问、泄露、更改或损坏的关键。以下是我们为确保数据库安全所采取的措施：

（1）用户密码掩码保护：我们采用了密码掩码技术，确保用户在输入密码时，界面上显示的是星号或其他占位符，而非密码本身。这防止了旁观者通过视觉方式获取密码信息。

（2）唯一账号标识机制：每个用户在系统中拥有一个唯一的标识，即员工编号。这个编号在整个系统的生命周期内保持唯一，并且由系统内部记录和维护。系统鉴别机制要求用户在登录时提供员工编号作为登录账号（LoginAccount），并输入密码（LoginPassword）。系统通过这一机制验证用户身份，确保只有合法用户才能获得访问数据库的权限。

（3）登录界面作为唯一入口：我们设计了一个登录界面，作为访问酒店信息管理系统的唯一途径。通过这个界面，用户可以安全地输入他们的登录凭证，而无需担心密码泄露的风险。

（4）数据库角色与员工编号的结合：我们将数据库的角色与员工的员工编号相结合，确保用户只能通过登录应用软件来访问数据库。这种设计避免了用户直接操作数据库，从而减少了数据被非法访问的可能性。通过这些措施，我们不仅提高了数据库的安全性，也确保了数据共享的合理性和有效性。我们的目标是创建一个既安全又高效的数据库管理系统，以满足酒店信息管理的需求。

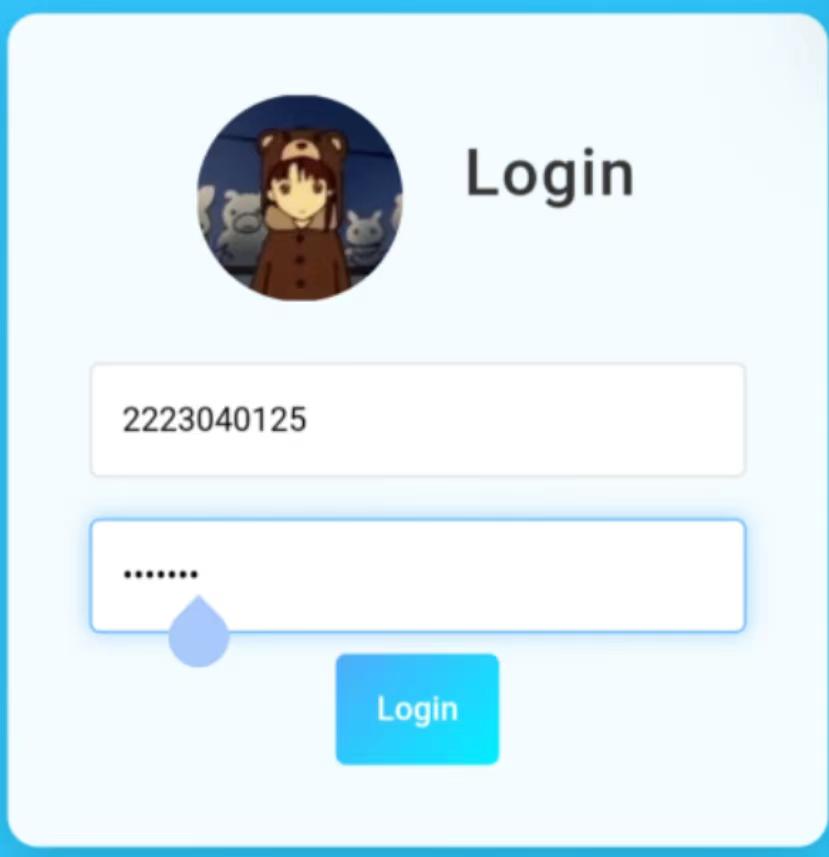


图 6-1 设计账户登录页面（左图为WEB端，右图为PYTHON GUI）

u

## **6.2角色与权限**

在酒店信息管理系统中，数据库角色的设置是权限管理的关键组成部分。通过将权限归类到不同的角色中，可以有效地控制用户对数据库操作的访问，确保数据安全和操作的合理性。每个角色定义了一组特定的权限集合，这些权限决定了角色成员能够对数据库中的表和列执行哪些操作。

角色的设置遵循最小权限原则，即每个角色仅拥有完成其职责所需的最小权限集。这样做可以避免过度授权，减少数据泄露和商业机密被非法获取的风险。同时，确保用户拥有足够的权限来完成其工作，避免因权限不足而影响工作效率。

在酒店信息管理系统中，角色和权限的设置如下：

1级员工：拥有对客房表和订单表的基本操作权限，包括更新（Update）和查询（Select）。此外，1级员工还可以查询跨平台比价表和客户表，但仅限于查询操作。

2级员工：权限更为广泛，可以对员工表、订单表、跨平台比价表和客房表执行更新、删除和查询操作。对于客户表，2级员工同样拥有更新和查询的权限。

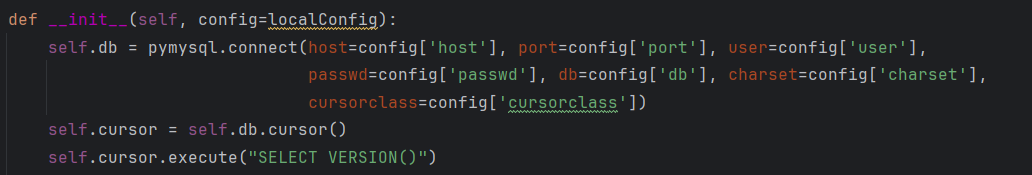
通过为用户分配适当的角色，可以确保他们获得完成工作所需的权限，同时保护数据库中的数据不被未授权访问。这种基于角色的权限管理方法简化了授权过程，提高了系统的安全性和灵活性。

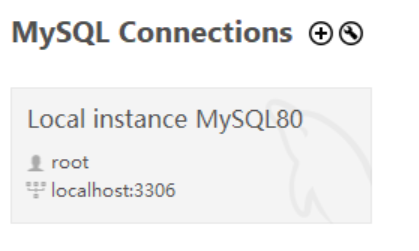
表5 数据库的角色

|  |  |  |
| --- | --- | --- |
| 角色 | 可以访问的表与列 | 操作权限 |
| 1级员工 | 客房表 | Update、Select |
| 订单表 | Update、Select |
| 跨平台比价表 | Select |
| 客户表 | Select |
| 2级员工 | 员工表 | Update、Delete、Select |
| 订单表 | Update、Delete、Select |
| 跨平台比价表 | Update、Select、Delete |
|  | 客房表 | Update、Delete、Select |
|  | 客户表 | Update、Select |

# **七、系统实现**

## **7.1数据库连接**

我采用的编译器为 Pycharm2023.1.6, 采用的 DBMS 为 MySQL 8.0.36。使用pymysql在集成开发环境（IDE）配置连接，实现软件设计和数据库设计一站式完成。 使用在@localhost 3306端口的mysql，正确输入DBA用户名和密码即可连接成功。

图 7-1 在 Pycharm 中连接 MySQL 数据库

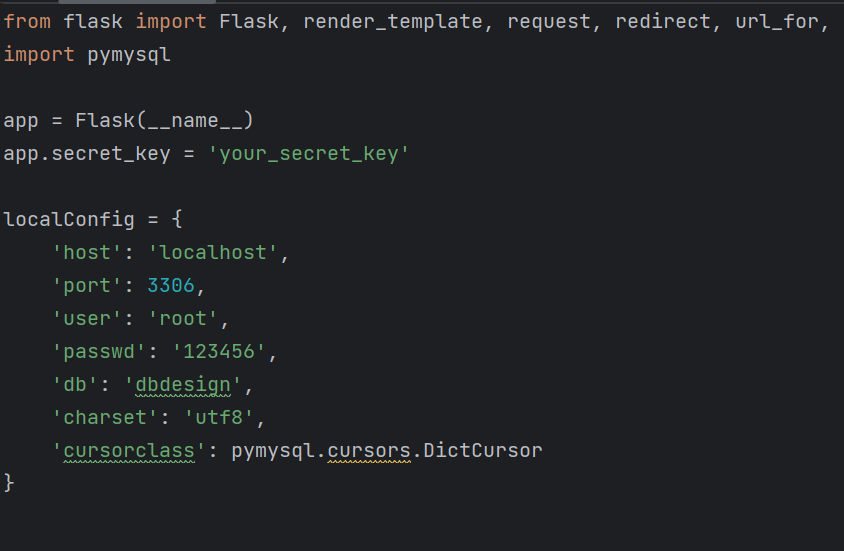
图 7-2 连接 MySQL 数据库及配置数据库编译器

图 7-3 python 数据库配置

## **7.2功能界面设计**

主页提供了一个清晰的概览，用户可以快速访问系统的主要功能区域。它展示了系统的核心价值，即通过一个统一的界面实现对客房、员工和客户管理的综合操作，同时支持订单跟踪和报表生成，为酒店的业务运营和决策提供强有力的支持。

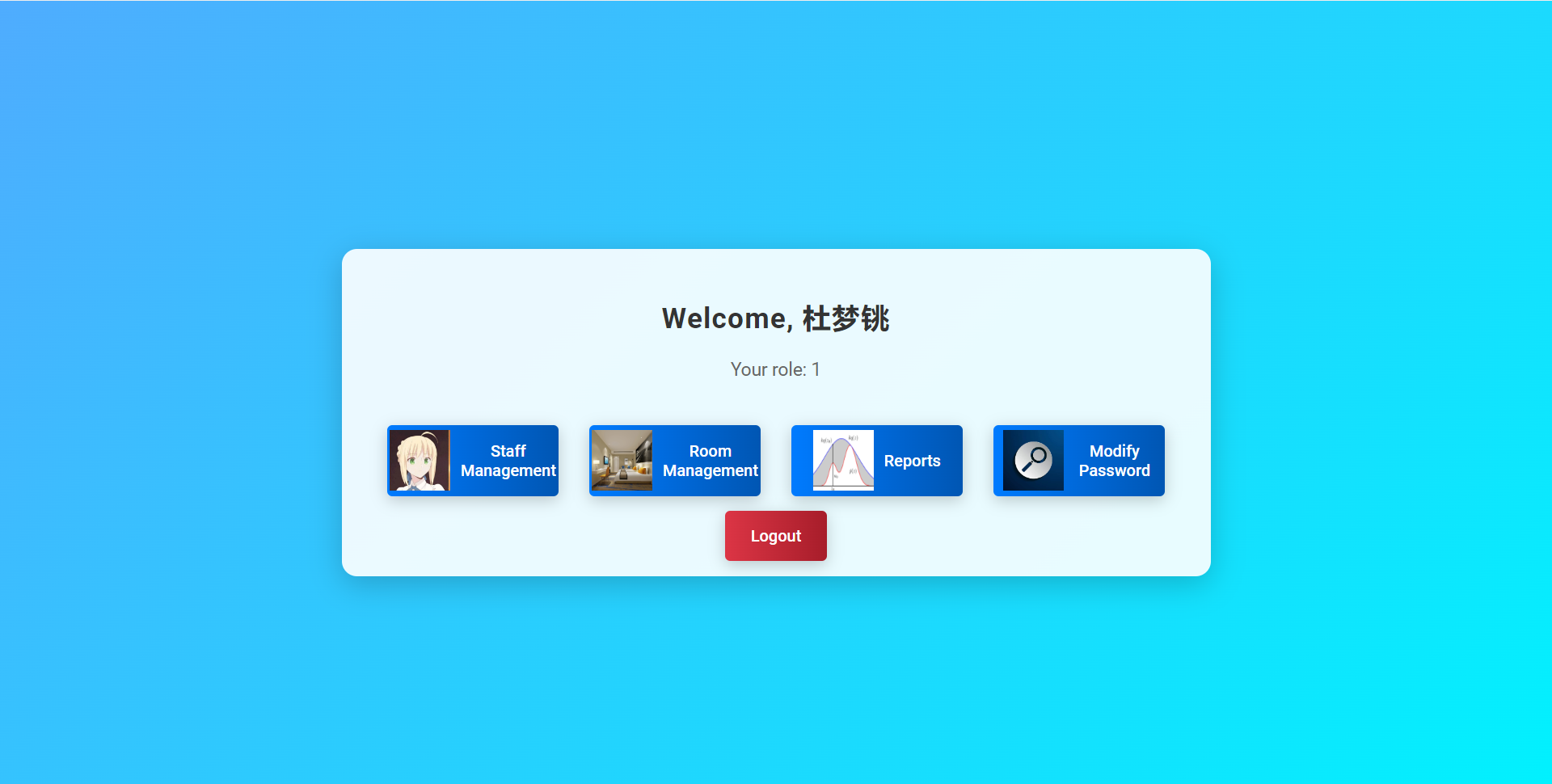
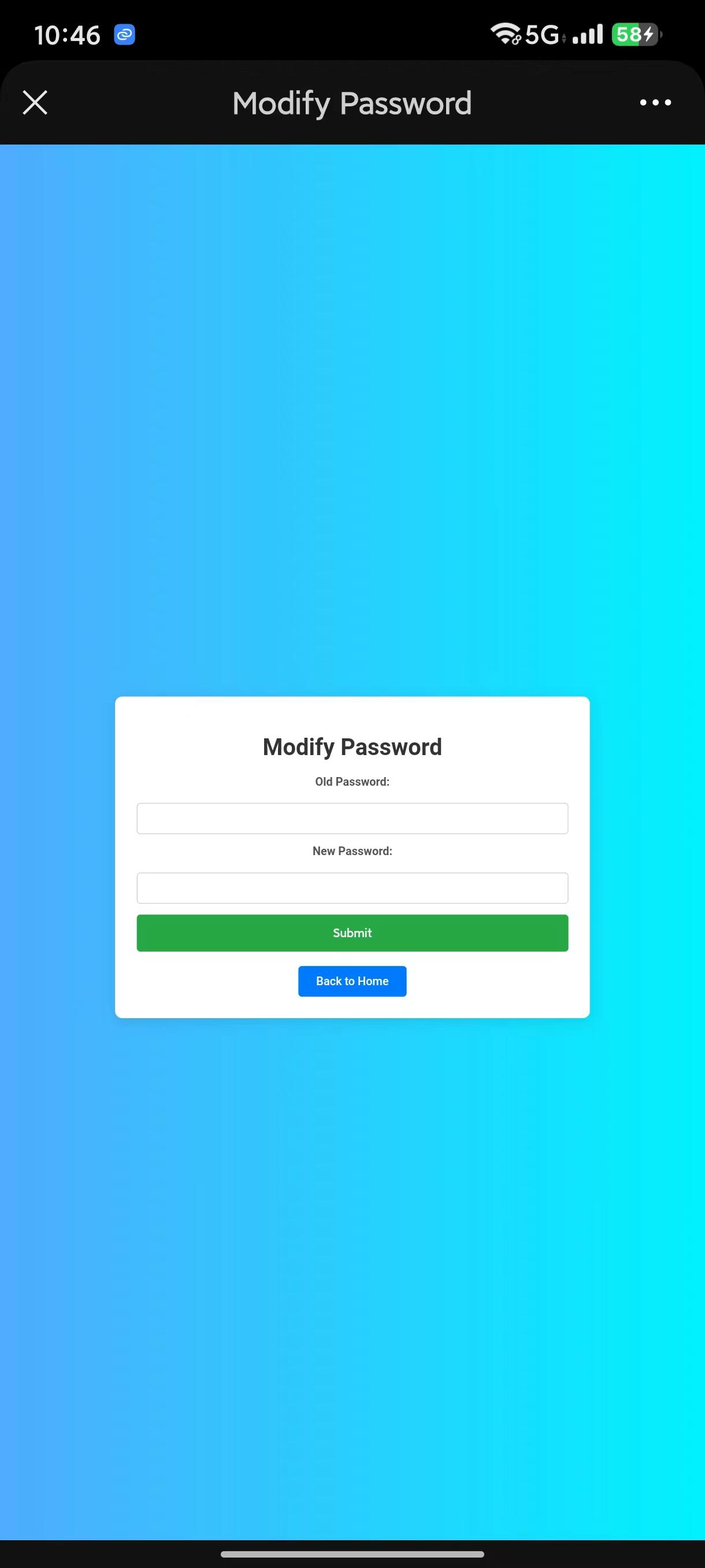
 图 7-4 酒店管理软件主页

图 7-5修改密码功能

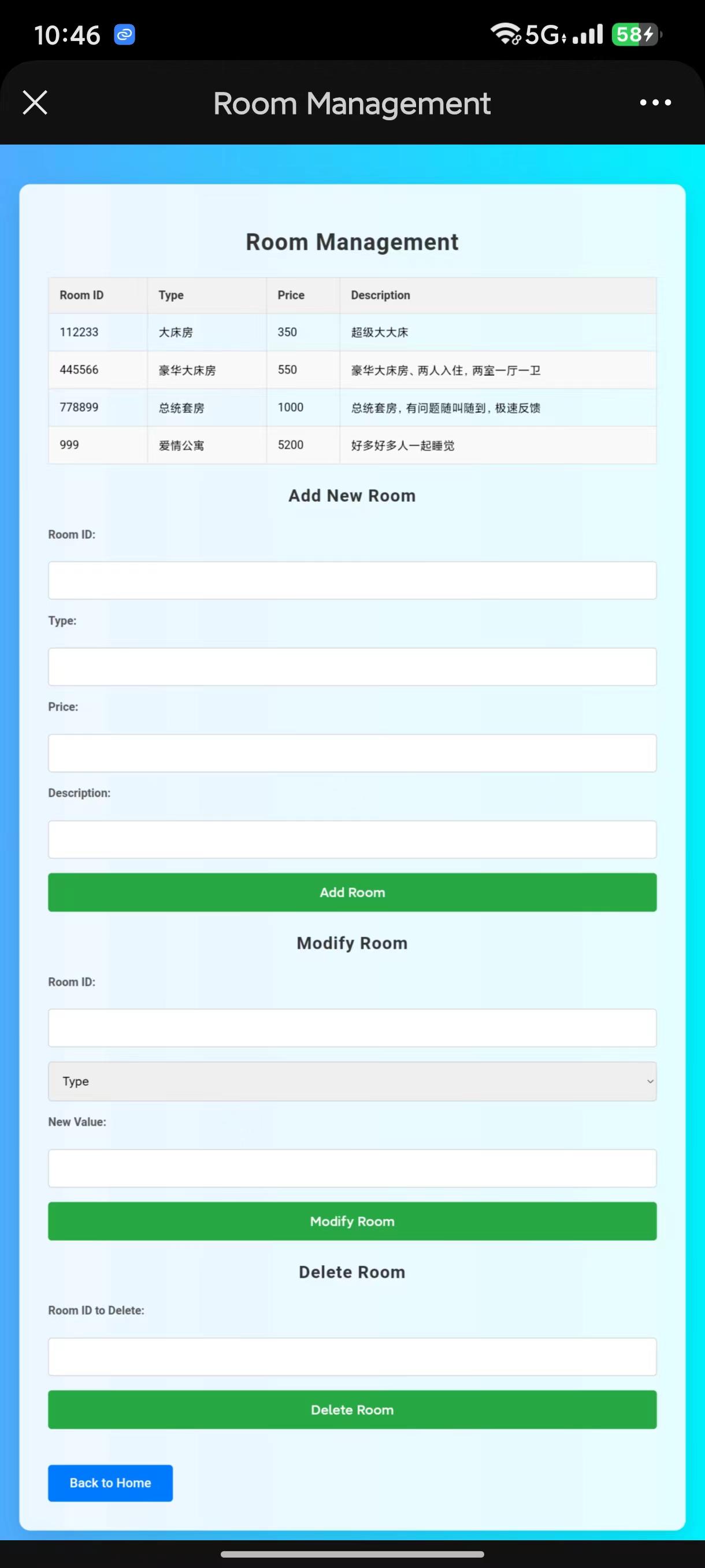
客房管理主页允许用户执行查询、增加、删除和修改客房信息的操作。这一界面的设计旨在简化客房管理流程，确保用户能够高效地处理与客房相关的任务。

图 7-6客房管理主页（查询、增加、删除、修改酒店）

员工管理主页为用户提供了搜索、增加、删除和修改员工信息的功能。这个界面的设计考虑了员工信息管理的便捷性，使用户能够轻松地维护员工资料和工作记录。

图 7-7 员工管理主页（搜索、增加、删除、修改员工）

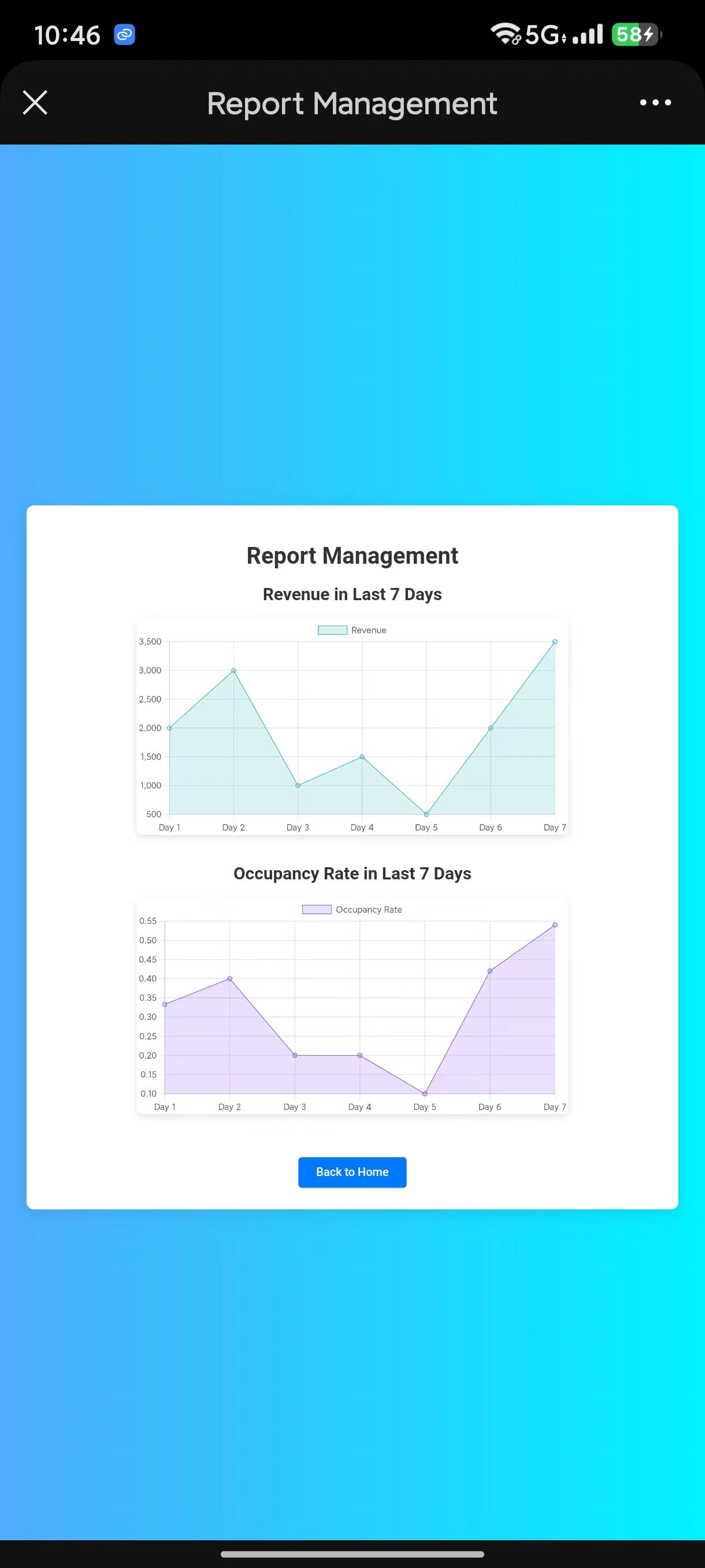
订单报表功能界面展示了一周内的利润和房间入住率，为用户提供了关键的业务指标。这一功能帮助管理者快速了解酒店的运营状况，为制定策略和优化服务提供了数据支持。

图 7-8订单报表功能（一周利润及房间入住率）

# **八、总结**

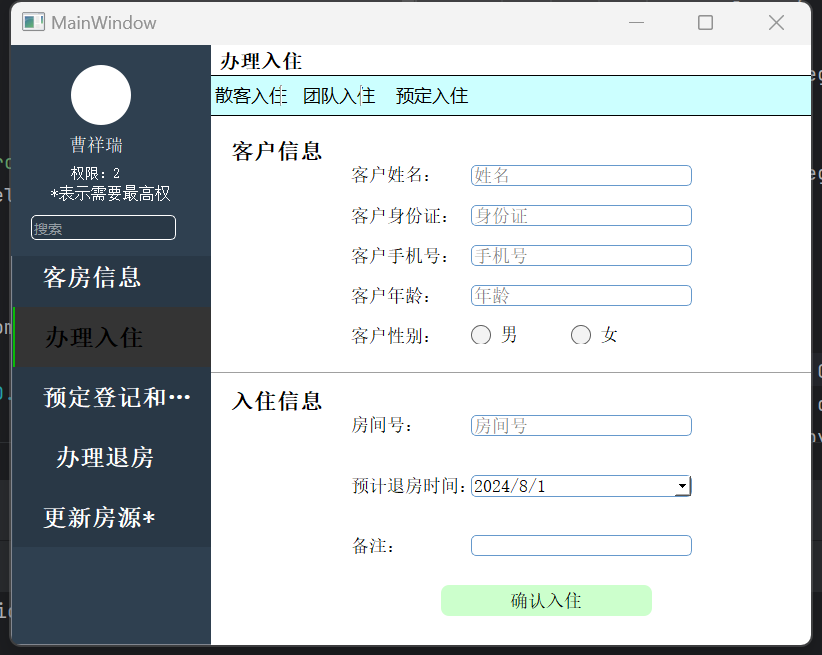
## **8.1问题解决**

课程设计中遇到的问题，解决的方式。

### 要点一：GUI → WEB

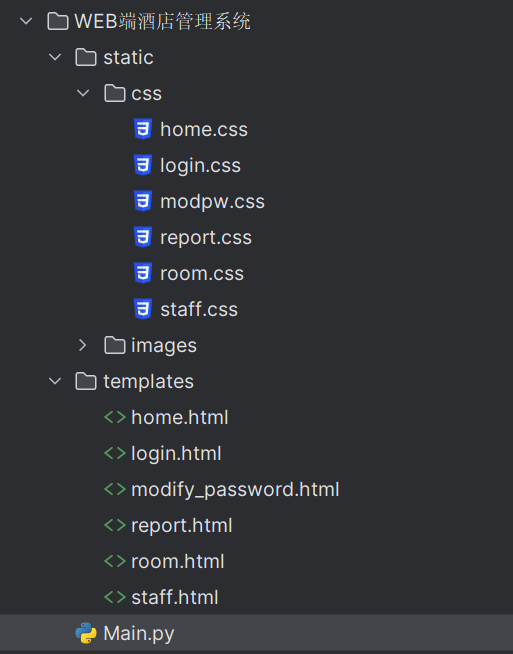
我们刚开始文稿**设计了很多功能**，结合网上的信息系统模板和GPT-4o和文心一言。我们在**GUI端**利用pyqt5产生了**近4000行代码**。虽然说整体架构我们都清楚，但是代码结构不清晰和过多的变量应用导致前端界面实现，但是**交互功能有些不能实现**。

下面仅展示部分页面（共有15个界面）：



## **8.2解决方式**

我们全部重写了代码，以**标准的代码结构**完成了WEB端代码（即正文部分）。



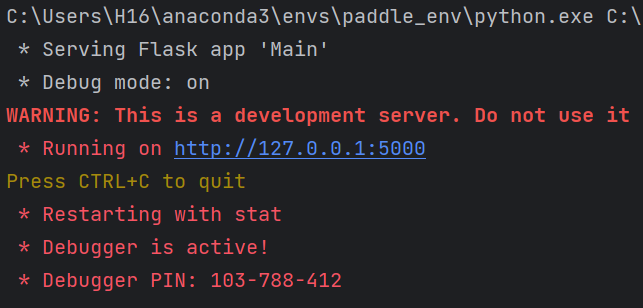
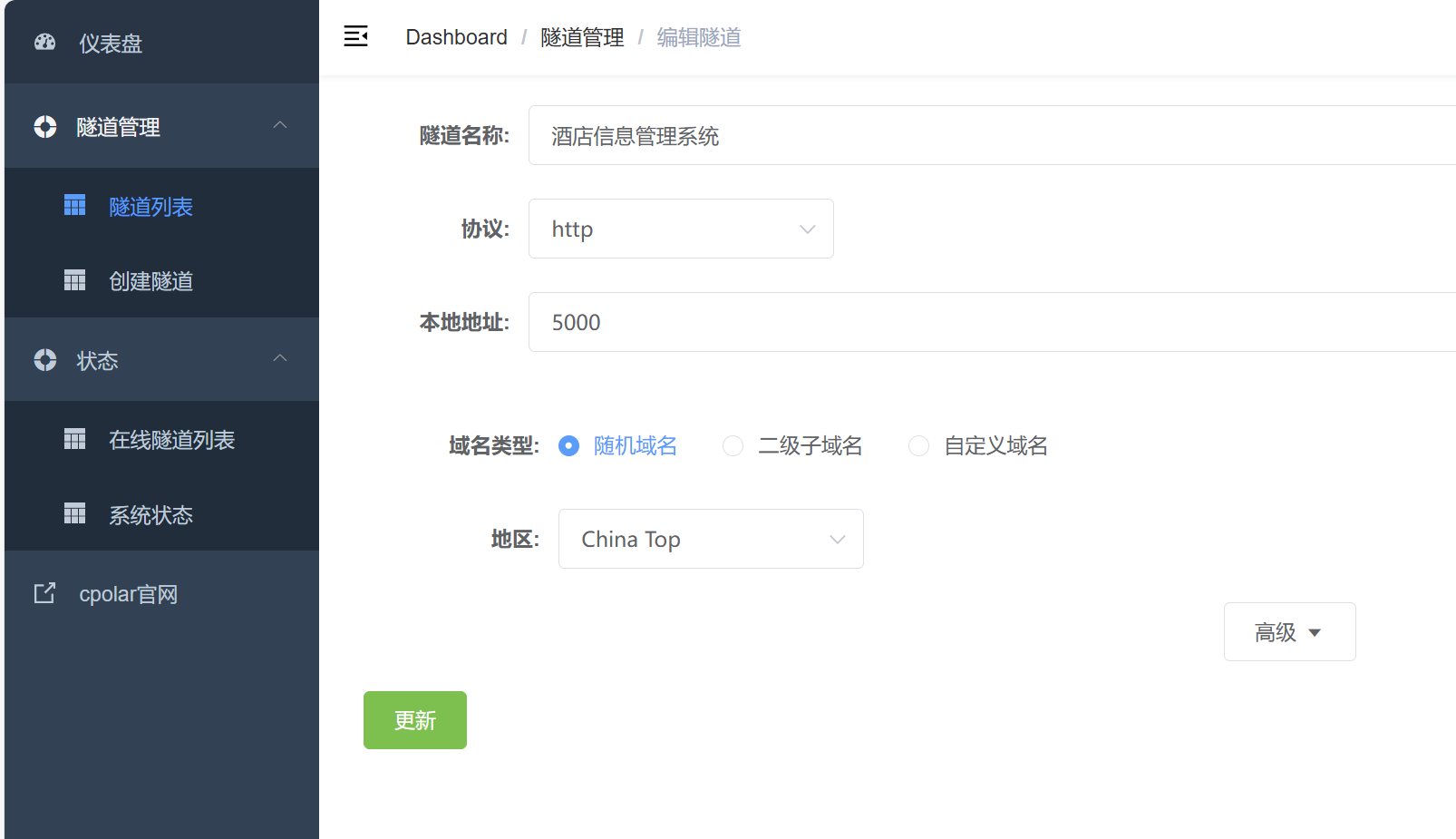
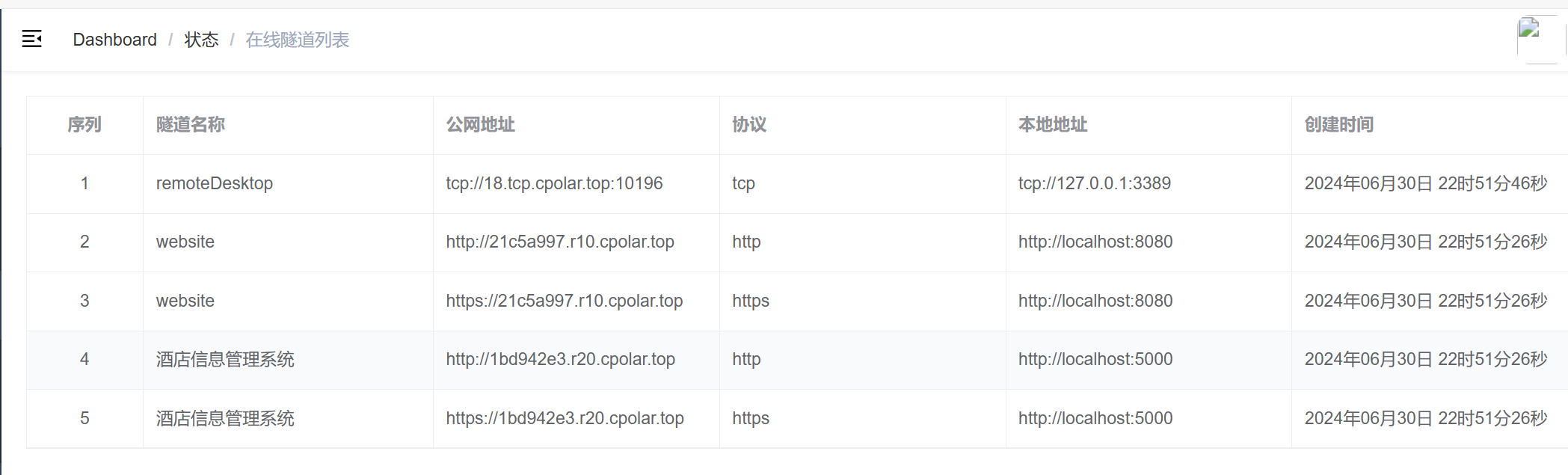
最后保留核心功能，优化到了1500行。其中AI含量大大降低，绝大多数的代码逻辑都是自我完成（不然也实现不出来）。

### 要点二：放弃打包为exe文件

打包为.exe文件失败。我们尝试了使用pyinstaller模块将文件打包为可执行文件，但是发现效果并不好。最终也没有采取这条路。

### 要点三：使用内网穿透实现公网访问

WEB端假如只能在Localhost或者局域网限制下使用，那离全栈开发的本意和实际应用场景就非常远。因此我们实现了把localhost通过Cpolar内网穿透的方法映射到公网上。

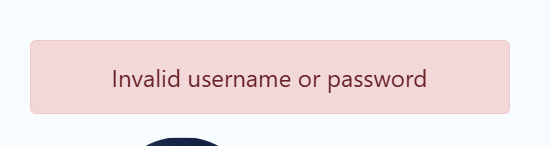


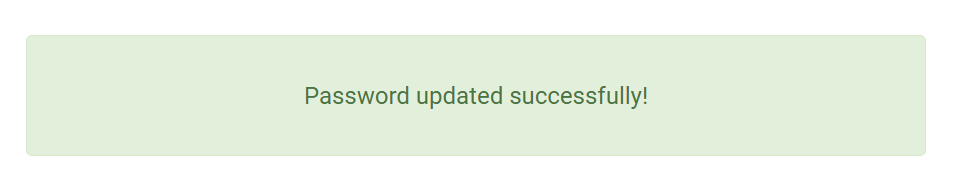
### 要点四：模仿企业端的交互实现

上本项目在**CSS**下了很多功夫，图片见正文。核心点如下：

1. 各container有一个加载的动画
2. user login的图片可以实现自动放缩（变大变小）
3. 背景图片是随时间渐变的
4. 所有子框都是圆角，符合现代企业美学设计
5. 数据表报从matplotlib换成了数据输入html使用chart库实现能和用户鼠标交互的图表
6. 登录界面有密码掩码和空值提醒
7. 每一个html都可以回到home模块，home模块可以退出登录

此外我们有一系列的**操作响应**，对成功和失败都会给用户对应的提示：

失败为红色，且会有对应错误类型：

 成功为绿色，显示当前操作：

 以下是部分代码示例：

