



山东大学
SHANDONG UNIVERSITY

酒店综合系统面向对象设计

目录

第一部分：面向对象的需求分析

| | |
|------------------------|----|
| 一：系统功能描述..... | 5 |
| 1.1 人事管理子系统功能需求..... | 5 |
| 1.2 财务管理子系统功能需求..... | 5 |
| 1.3 酒店物资管理子系统功能需求..... | 6 |
| 1.4 餐饮住宿管理子系统功能需求..... | 6 |
| 1.5 网上预约子系统功能需求..... | 7 |
| 1.6 后台管理子系统功能需求..... | 8 |
| 二：用例模型..... | 8 |
| 2.1 顶层用例图..... | 8 |
| 2.2 财务管理子系统..... | 9 |
| 2.3 人事管理子系统..... | 10 |
| 2.4 物资管理子系统..... | 11 |
| 2.5 网上预约子系统..... | 11 |
| 2.6 酒店住宿餐饮管理子系统..... | 11 |
| 2.7 后台管理子系统..... | 12 |
| 三、对象模型-初始类图..... | 13 |
| 四、Actor 与用例描述..... | 15 |
| 4.1 主要 Actor 描述..... | 15 |
| 4.2 主要用例描述..... | 15 |
| 五、动态建模..... | 18 |

| | |
|-----------------|----|
| 5.1 顺序图..... | 18 |
| 5.2 状态特征提取..... | 20 |
| 5.3 状态图..... | 21 |

第二部分：面向对象的设计

| | |
|--------------------|----|
| 六、体系结构设计..... | 24 |
| 6.1 体系结构总图..... | 24 |
| 6.2 包的组织与包图..... | 26 |
| 6.3 部署图..... | 26 |
| 七、对象建模-细化类图..... | 27 |
| 八、接口设计..... | 29 |
| 8.1 人机交互接口设计..... | 29 |
| ①管理员登录界面..... | 29 |
| ②可用房间信息查询界面..... | 29 |
| ③前台办理线下入住界面..... | 30 |
| ④前台查询餐厅余位信息界面..... | 30 |
| ⑤前台安排服务事务界面..... | 31 |
| ⑥订单查询修改界面..... | 31 |
| ⑦客户登录/注册界面..... | 32 |
| ⑧客户选择客房界面..... | 32 |
| ⑨客户预订客房/餐饮界面..... | 33 |

| | |
|-------------------------|----|
| 8.2 模块间接口设计..... | 33 |
| ①查询员工与删除/修改员工之间的接口..... | 33 |
| ②查询房间/餐桌与删改房间/餐桌..... | 34 |
| 8.3 应用服务器与数据库通信接口..... | 34 |
| 九、数据设计..... | 35 |
| 9.1 ER 关系图..... | 35 |
| 9.2 关系模式的建立..... | 35 |
| 9.3 数据库中表的实现..... | 36 |
| 9.4 建表语句（MySQL）..... | 40 |
| 十、过程设计..... | 44 |
| 10.1 活动图..... | 44 |

一：系统功能描述

1.1 人事管理子系统功能需求

- 1 人事管理员登录：人事管理员通过账号密码登录人事管理子系统
- 2 员工信息管理：人事管理员在该系统处于高层管理员身份，将自己、财务管理员、服务员、采购员、前台人员的信息存入员工数据库中，并进行增删改查功能。
- 3 员工信息查询：人事管理员可以通过特定信息查询员工信息

1.2 财务管理子系统功能需求

- 1 财务管理员登录：财务管理员通过账号密码登录账务管理子系统
- 2 工资发放：财务管理员点击发放工资按钮，通过员工数据库里的工资信息，为员工发放工资。该功能设有时间权限，即只有每月 1 日发放工资，其余时间财务管理员无权限发放工资，发放工资以后，自动生成账务记录并存入财物表。
- 3 住宿、餐饮订单导入：订单、退单信息导入到财务管理子系统中
- 4 物资资金发放、物资信息导入：点击购买对应物资按钮，即采购清单编号加上采购总金额记录在财务管理子系统中。
- 5 财务信息查询：财务管理可选择相应的查询条件查询账目明细并生成报表。

1.3 酒店物资管理子系统功能需求

- 1 采购员登录：采购员通过账号密码登录酒店物资管理子系统。
- 2 物资清点：采购员选择上报已采购物资，正确输入 ID 及密码后输入采购物资详情，将购入物资数据整理同步入系统。
- 3 打印清单：采购员正确输入 ID 及密码后，查询物资仓库中各项物资的剩余量信息，打印待采购物资清单，依照清单列表到市场采购物资。

1.4 餐饮住宿管理子系统功能需求

- 1 前台查看房间和餐厅信息：前台根据顾客提出的需求查看相应类型的房间和餐厅还有多少间或多少位置，然后为顾客办理相应业务
- 2 前台查看客户订单：前台根据顾客提供的身份信息获取顾客已经完成的订单，并根据订单为顾客提供相应的服务。
- 3 前台为没有预定的顾客办理相应业务：前台根据顾客的需求与提供的身份信息以及剩余的符合需求的房间为客户建立新的订单，并直接在前台完成支付，支付完成后顾客可直接办理入住及就餐。
- 4 前台为有预约的顾客办理相应业务：前台根据顾客的身份信息查询相匹配的订单信息，完成匹配后顾客可以直接办理入住及就餐
- 5 前台处理线上的顾客订单变动业务：前台根据顾客的身份信息与订单编号及需求生成新的退单订单，并将新的订单返回给住宿餐饮管理子系统，执行订单的内容，并将退单改单成功信息线上返回给顾客

○6 前台处理线下的顾客订单变动业务：前台根据顾客的身份信息及需求生成新的退单订单，并将新的订单返回给住宿餐饮管理子系统，执行订单的内容。

○7 完成物资请求：系统根据房间和餐厅的详细信息生成需求分析，并把需求分析作为物资请求向数据库物资表发送，然后让采购员根据生成的采购清单去进行采购

⑧新事务接收：获取前台发送的订单，订单信息录入到综合事务表中，综合事务表依据传入的订单细节触发生成待服务信息，服务员确认待服务事项后接收并开始工作，综合事务表中的对应订单打上“正在完成”的标签，对应的服务员打上“正在工作”标签。

⑨反馈工作情况：服务员在完成待服务事项后，通过账号密码登录，在综合事务表中自己刚刚完成的对应订单打上“已完成”标签，同时自己打上“空闲”标签。

1.5 网上预约子系统功能需求

○1 用户注册与登录：用户使用软件时需要登录，如果没有注册过帐号则先使用自己的基本信息进行账号注册，注册完毕后使用该账号进行登录并预约房间。

○2 用户进行可用客房的查询：用户可以通过软件进行对可用的客房进行查询，查看自己想要预约的房间是否空闲

○3 用户进行可用餐桌的查询：用户可以通过软件进行对可用的餐桌进行查询，查看自己想要预约的餐桌是否空闲

○4 用户进行预约：用户使用软件对房间或者餐桌进行预约，对餐桌进行预约的时候不需要付订金，而对房间进行预约的时候则需要付全款

○5 用户取消预约：如果用户在北京时间 18：00 前取消预约则退还押金，如果超过当天 18：00 则预约单自动变为订单，用户取消预定也不会退还押金，用户可随时取消对餐桌的预约。

○6 用户查询预约记录：用户可以通过软件查询自己的预约记录，考虑自己是否退单或者再进行预约、改时。

⑦预约单转变订单：如果时间到预约时间当天的 18：00，则自动将预约记录转化为订单记录，用户取消预约也无法退还押金。

⑧对预约进行取消：如果用户在规定时间之前取消预约，则删除预约表中的记录，并将钱退还。

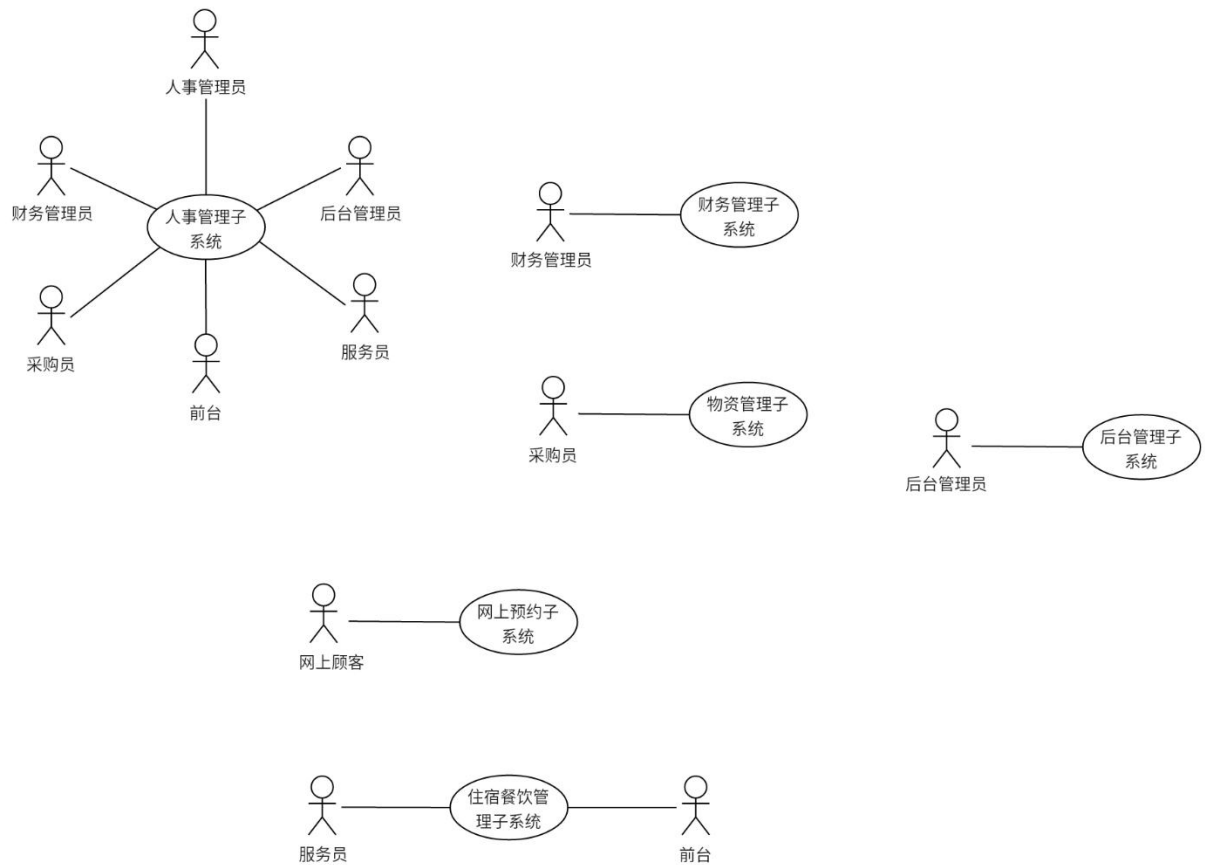
1.6 后台管理子系统功能需求

○1 管理员查看房间信息：后台管理员查看目前的房间类型，数量，价格等具体信息。

○2 管理员查看餐厅信息：后台管理员查看目前的餐厅座位，数量，价格等具体信息。

○3 房间信息管理：管理员对房间各种进行增删改查的操作。

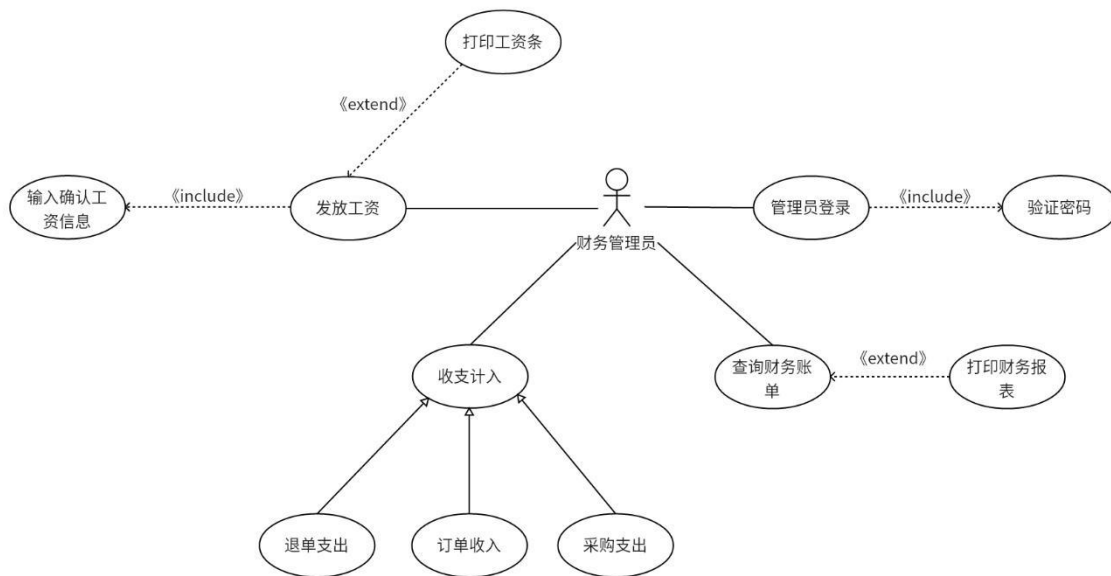
○4 餐厅信息管理：管理员对餐厅信息进行增删改查的操作。



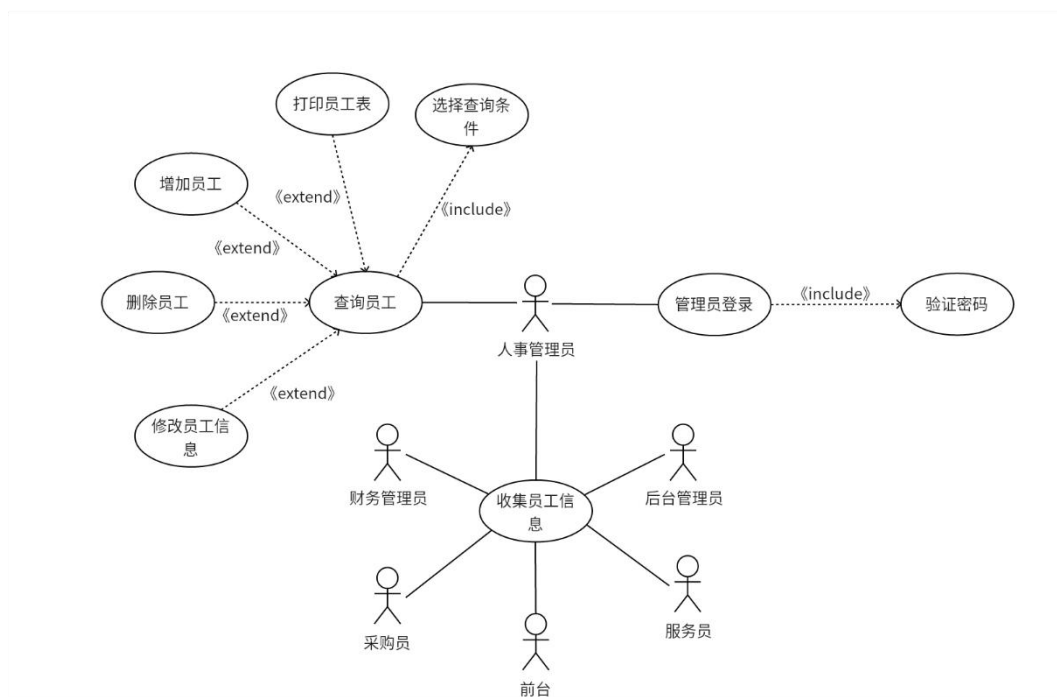
二：用例模型

2.1 顶层用例图

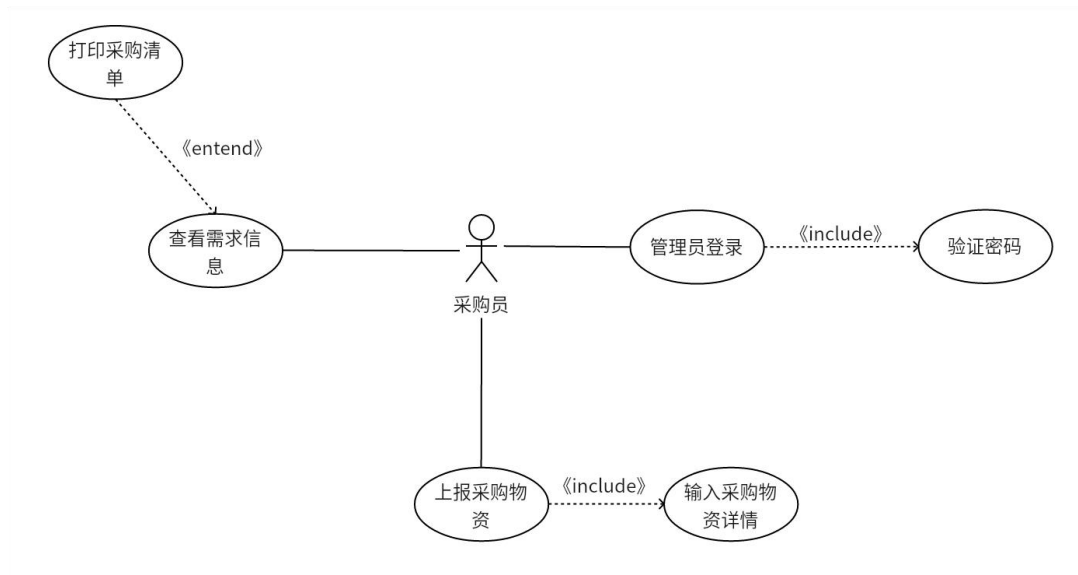
2.2 财务管理子系统



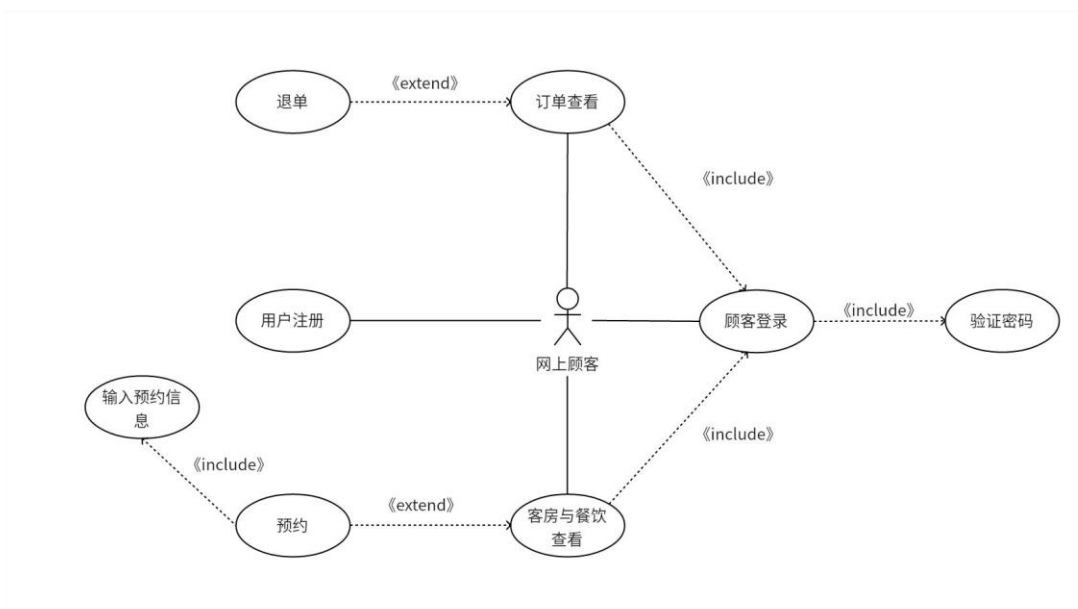
2.3 人事管理子系统



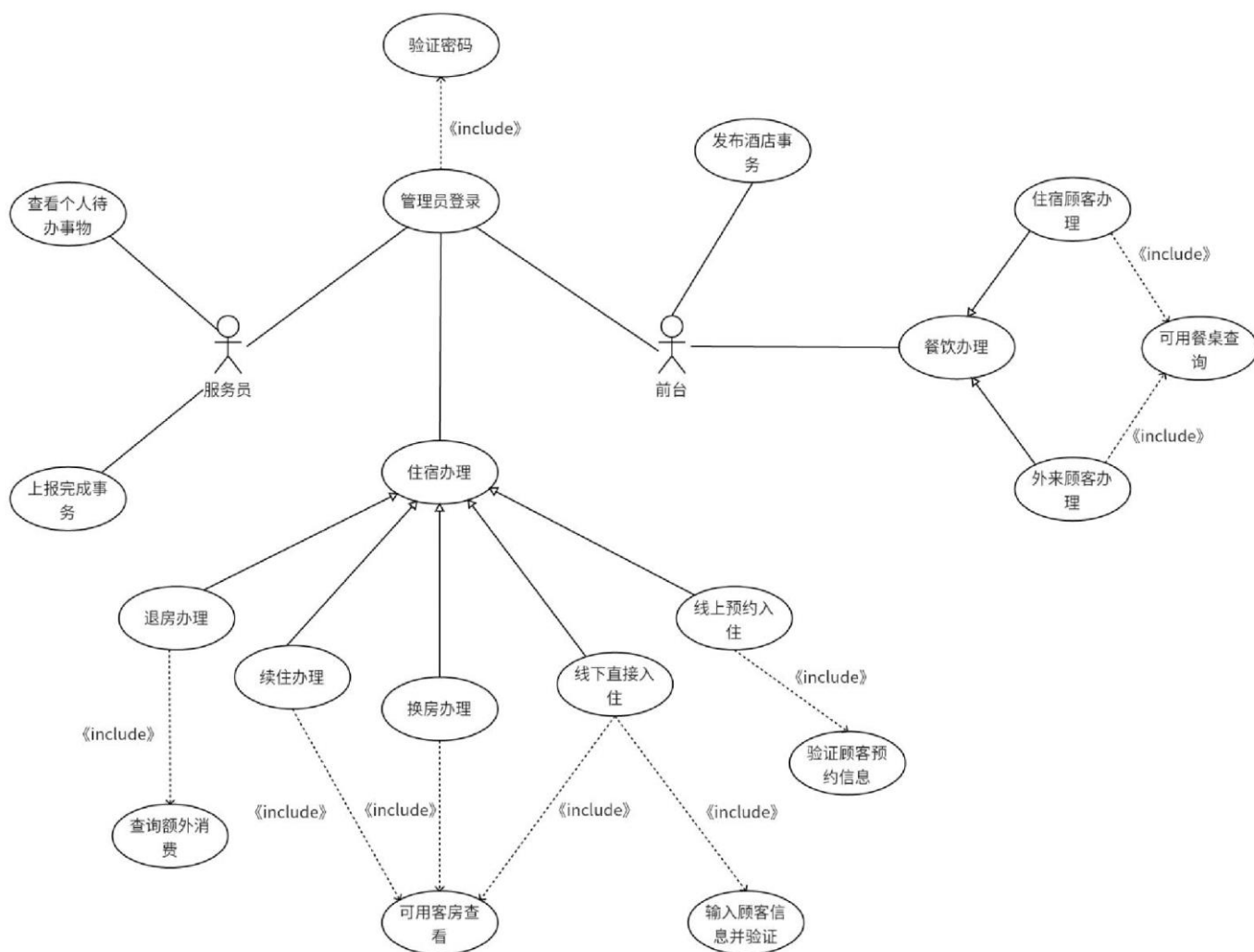
2.4 物资管理子系统



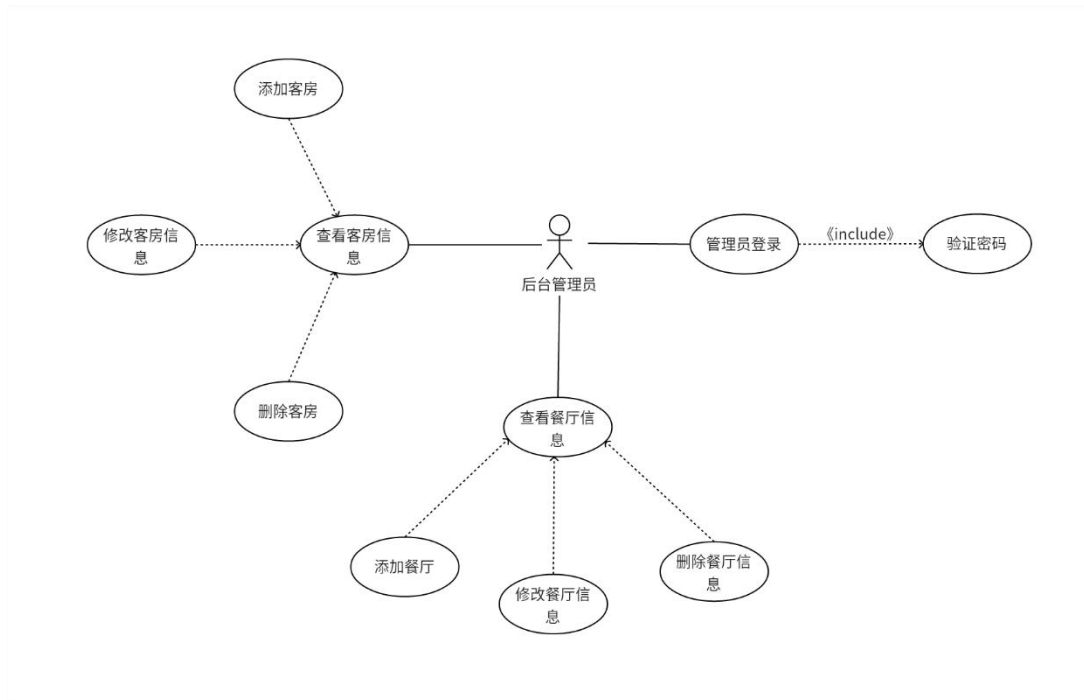
2.5 网上预约子系统



2.6 酒店住宿餐饮管理子系统

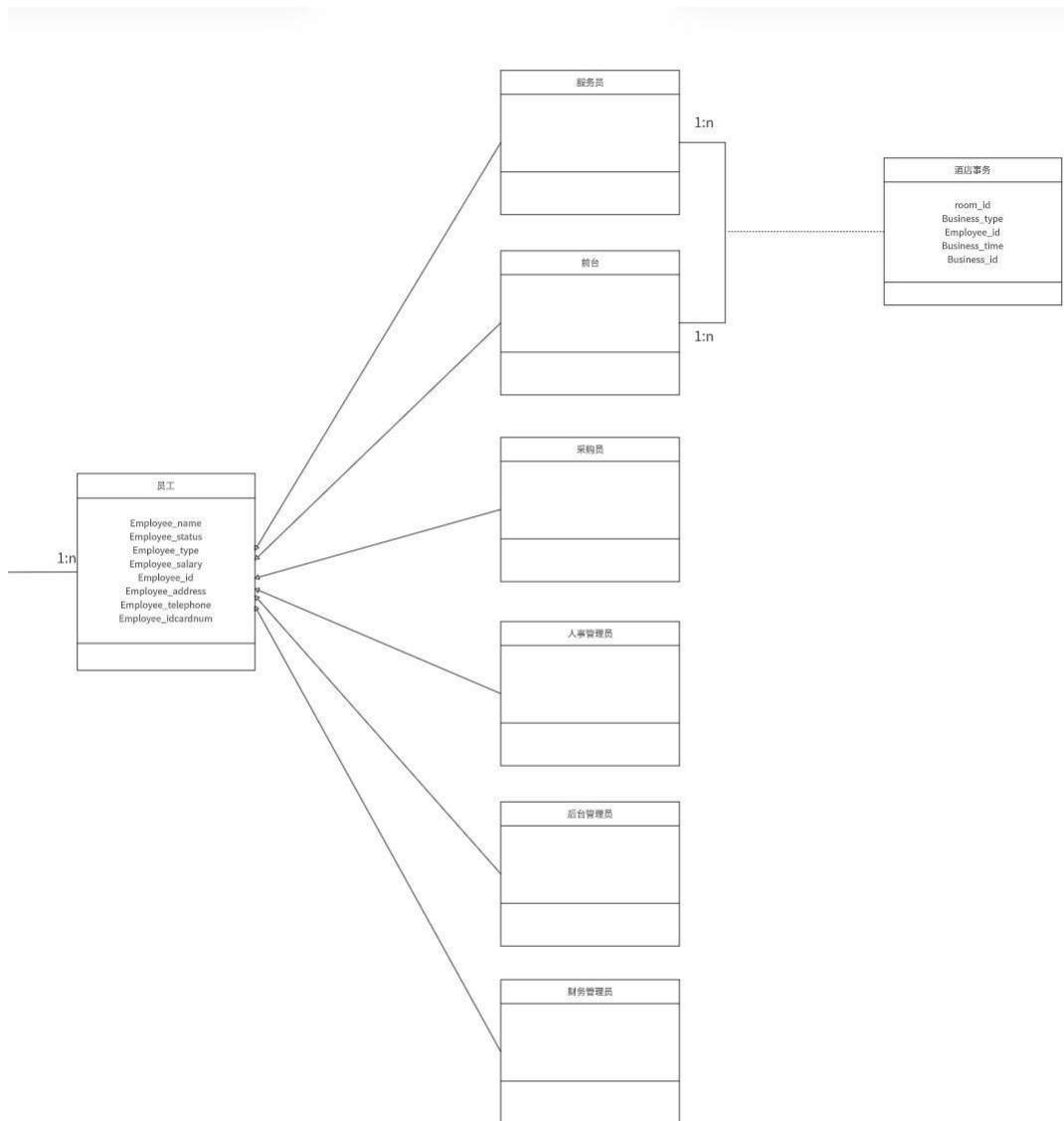
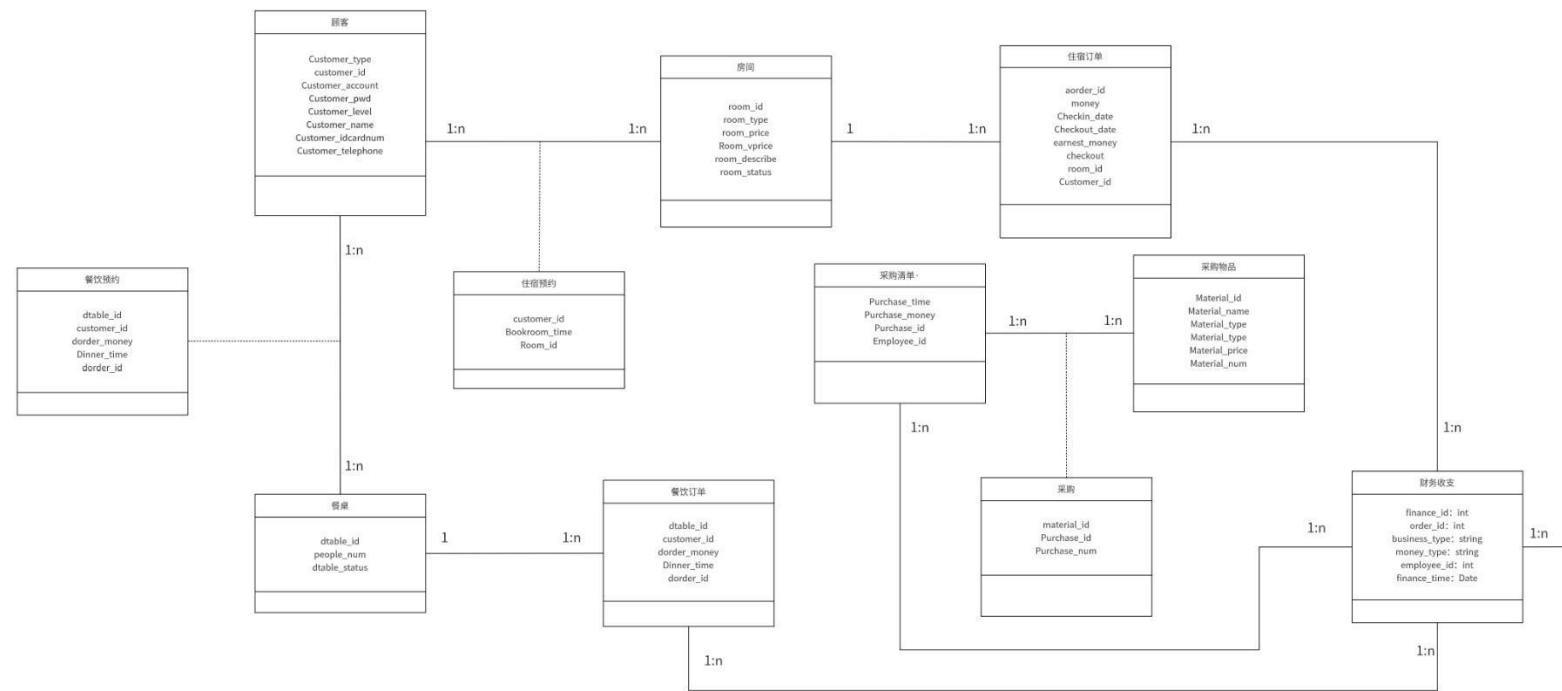


2.7 后台管理子系统



三、对象模型-初始类图

在初始类图中，我们加入系统所具有的实体类，共 13 个类。顾客与餐桌或房间、采购清单与采购物品、房间与住宿订单、餐桌与餐饮订单、各类订单与财务收支之间构成多对多的关联，并产生住宿预约、餐饮预约、采购等关联类，每个关联类包含多个预约细节。除此之外，各类管理员与员工组成泛化关系，员工可能是服务员、前台、采购员、人事管理员、后台管理员和财务管理员中的其中一种，其中前台与服务员产生多对多的关系，生成酒店事务的关联类，每个酒店事务包含若干事务细节。初始类图如下——



四、Actor 与用例描述

4.1 主要 Actor 描述

- ①角色：人事管理员角色职责：进行员工信息的增删改查操作
- ②角色：财务管理员角色职责：统计查询财务信息，包括采购物资信息、订单信息和员工工资信息，生成报表
- ③角色：采购员角色职责：统计需采购物资信息，上报物资采购情况
- ④角色：服务员角色职责：接收并完成待服务事项
- ⑤角色：前台角色职责：办理直接到店顾客的入住、预约顾客的入住、退房等业务
- ⑥角色：后台管理员角色职责：进行客房和餐厅信息的增删改查操作
- ⑦角色：客户角色职责：使用 APP 查看可用客房和可用餐桌信息并预约

4.2 主要用例描述

①用例：可用客房查看

| | |
|------------------------|---------------|
| 用例名：可用客房查看 | 执行者：客户、前后台管理员 |
| 目标：查看当前可以在指定时间内入住的客房信息 | |

功能描述：

1. 对于网上预约的客户，客户可以在 APP 中输入想要预约的房间类型以及入住和离开时间，后端可以生成算法将当前可用房间一览表呈现给客户方便查看和预约
2. 对于前台人员，为到点顾客办理直接入住时，可以查看当前可用客房信息

主要步骤：

1. 根据客房类型筛选出符合类型要求的房间
2. 根据住宿起始时间和结束时间筛选出符合时间要求的房间

相关用例：预约、直接入住

②用例：可用餐桌查看

用例名：可用餐桌查看

执行者：客户、前后台管理员

目标：查看当前可以在指定时间内就餐的餐桌信息

功能描述：

1. 对于网上预约的客户，客户可以在 APP 中输入想要预约的餐桌类型以及入住和离开时间，后端可以生成算法将当前可用餐桌一览表呈现给客户方便查看和预约
2. 对于前台人员，为到点顾客办理直接就餐时，可以查看当前可用餐

桌信息

| |
|--|
| <p>主要步骤：</p> <ol style="list-style-type: none"> 1. 根据餐桌类型筛选出符合类型要求的房间 2. 根据就餐起始时间和结束时间筛选出符合时间要求的餐桌 |
| <p>相关用例：预约、直接就餐</p> |

③用例：采购

| | |
|---|---------|
| 用例名：采购 | 执行者：采购员 |
| <p>目标：采购员可以查询当前物资剩余情况，也可以提交所采购物资信息</p> | |
| <p>功能描述：</p> <ol style="list-style-type: none"> 1. 采购员接收所需采购的物资信息，并将需采购物资信息录入采购单 2. 采购员采购完毕后，将所采购物资信息输入到物资管理子系统 | |
| <p>相关用例：采购</p> | |

④用例：发工资

| | |
|-------------------------|-----------|
| 用例名：发工资 | 执行者：财务管理员 |
| <p>目标：财务管理员为全体员工发工资</p> | |
| <p>功能描述：</p> | |

财务管理员一般在每月的某一天为全体员工发工资，但员工可以提出即时结算的要求，财务管理员应根据距离上次发工资的时间即时结算工资

相关用例：指定日期发工资、即时结算工资

⑤用例：网上预约

用例名：网上预约

执行者：客户

目标：客户可以上网在 APP 中预约某段时间内的任意多间客房或餐桌

功能描述：在订单记录表中插入一条预约信息

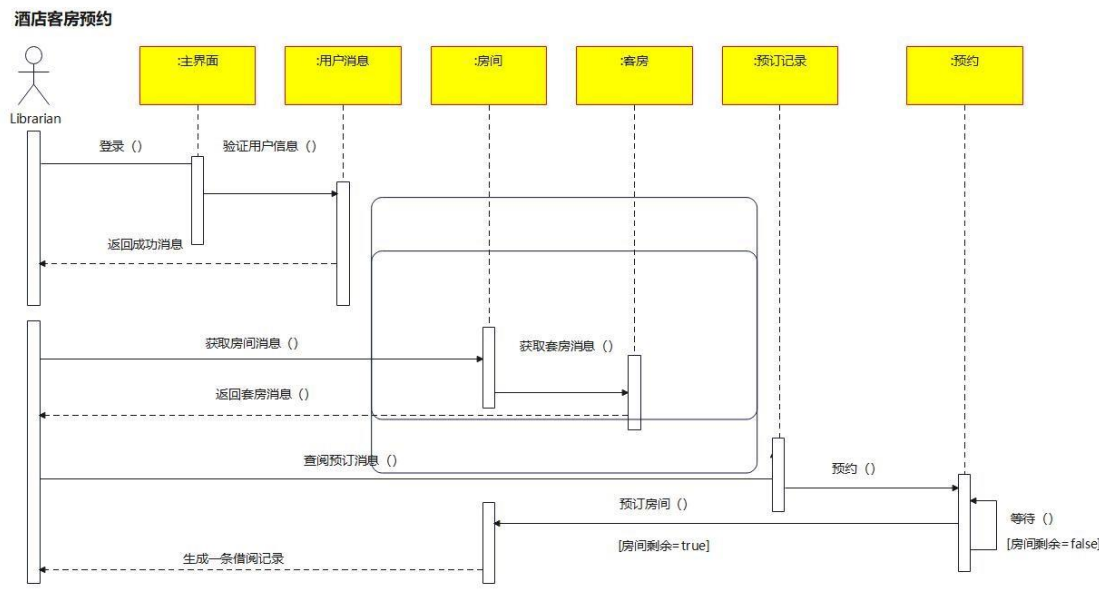
相关用例：可用客房查看、查看已预约信息

五、动态建模

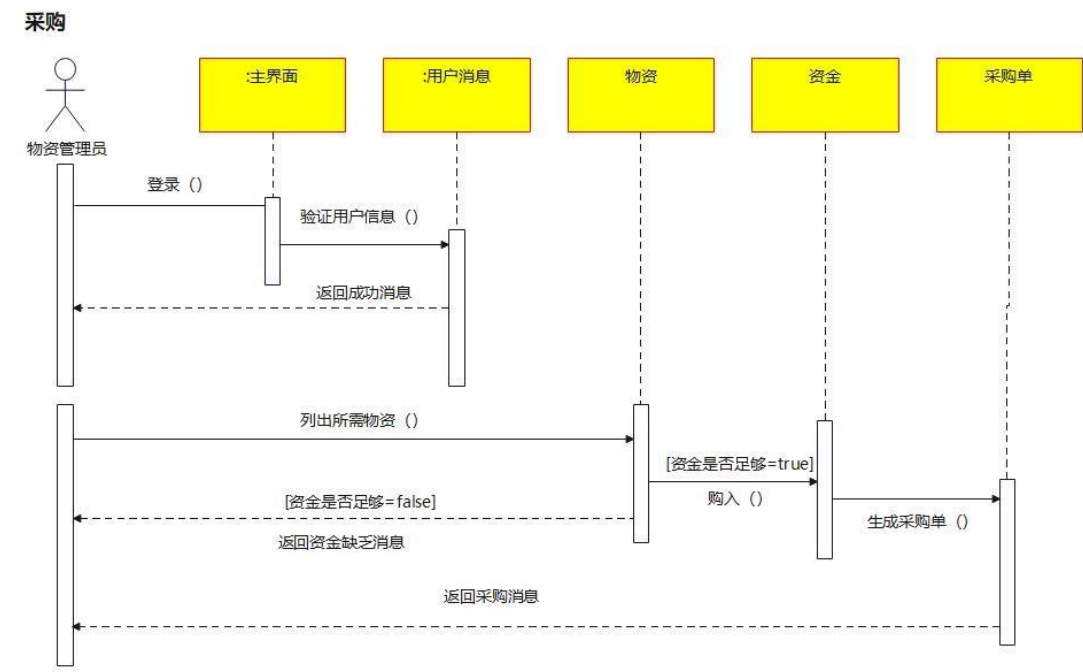
5.1 顺序图

在顺序图中描述了 Actor 与界面类的交互以及界面类再与其他类的交互以达成客户需求的具体流程。

○ 1 酒店客房预约

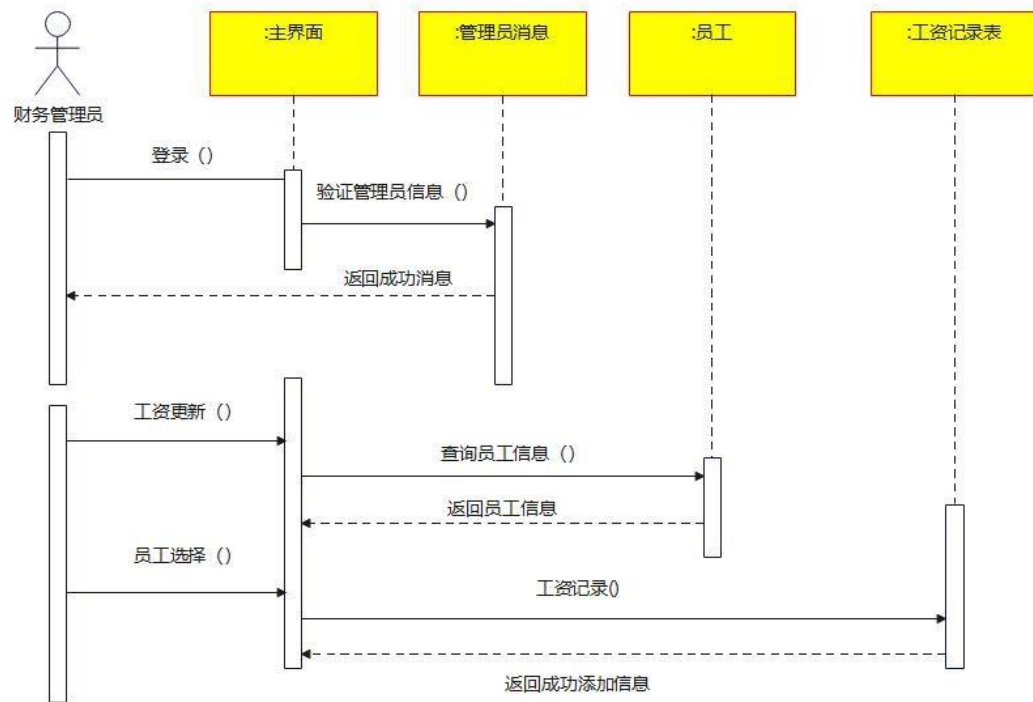


○ 2 采购



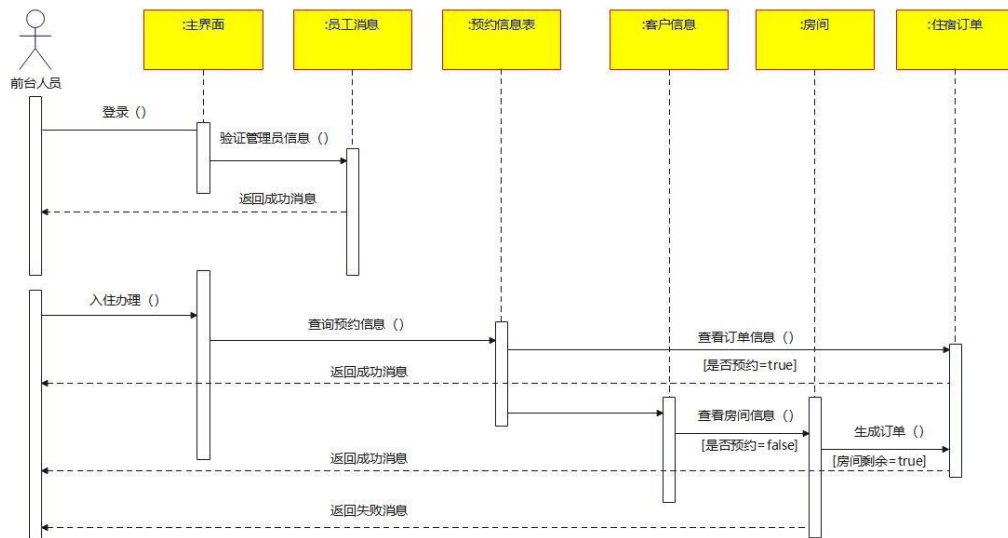
○ 3 工资操作

工资



○ 4 入住

入住



5.2 状态特征提取

在系统中，有明显状态变化信息的有以下几个类：

○1 用户账号
用户账号状态有：已注册普通状态、已登录普通状态、已注册

VIP 状态、已登录 VIP 状态、已预约房间状态、已入住状态

○2 预约订单

预约订单状态有：初始状态、预约房间状态、可用状态、存入订单表状态、完成状态、不可用状态、取消状态、预约餐桌状态

○3 住宿订单

住宿订单状态有：初始状态、可用状态、不可用状态、预约后初始状态、已支付状态、已入住状态、已完成状态

○4 餐桌订单

餐桌订单状态有：初始状态、可用状态、已支付状态、已使用状态、已完成状态、预约后初始状态、不可用状态

○5 采购单

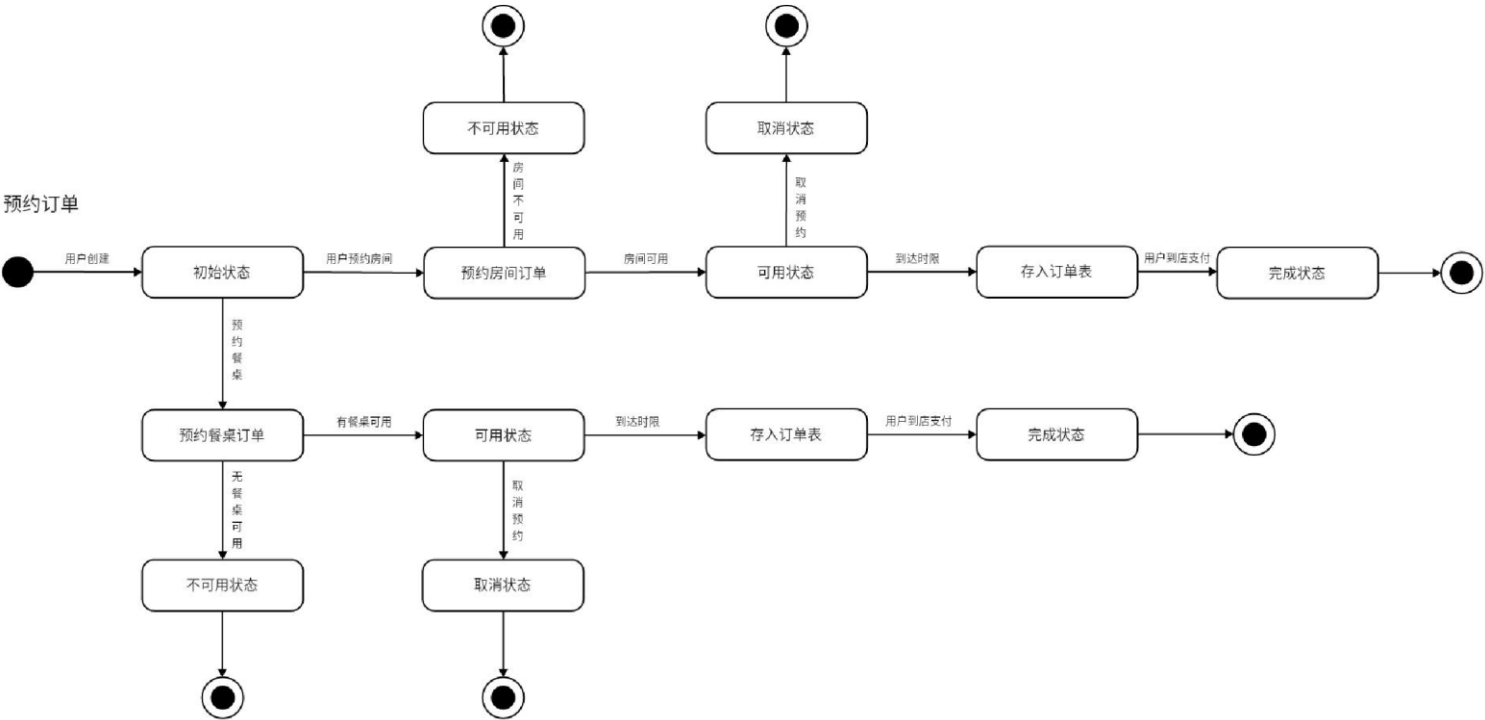
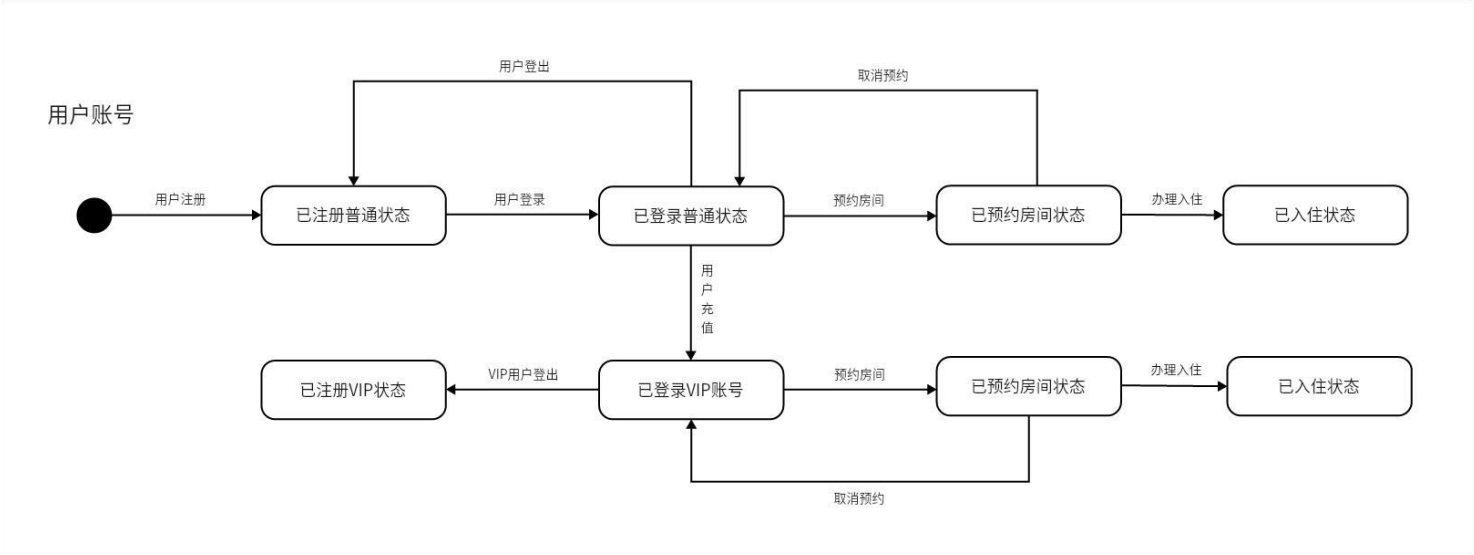
采购单状态有：初始状态、被查看状态、完成采购状态、写入采购数据库状态

○6 酒店事务记录

酒店事务记录状态有：初始状态、被接受状态、已完成状态、未被接收状态

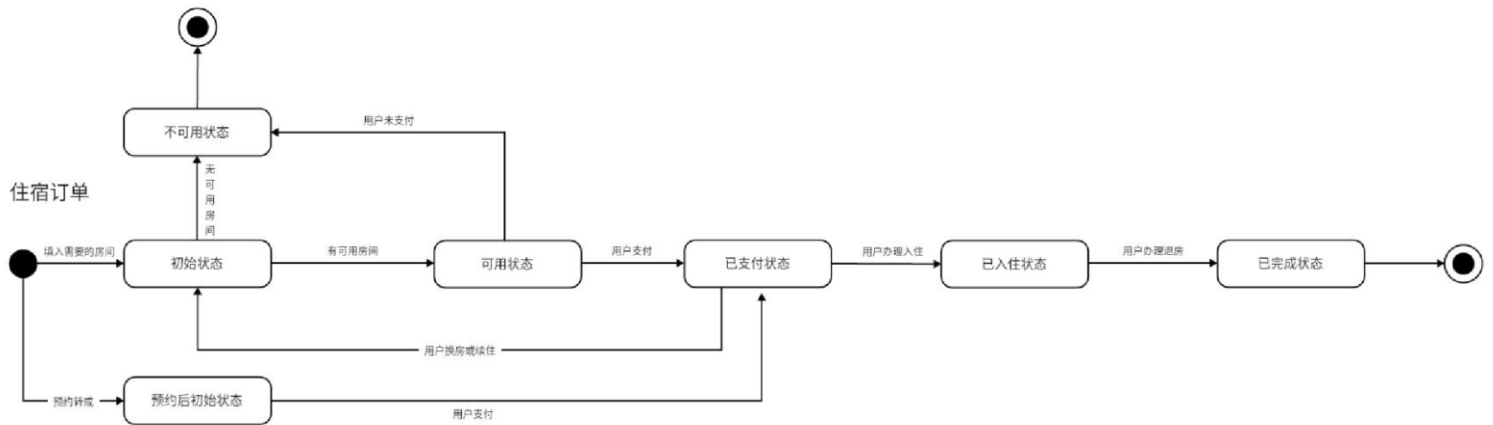
5.3 状态图

○1 用户账号

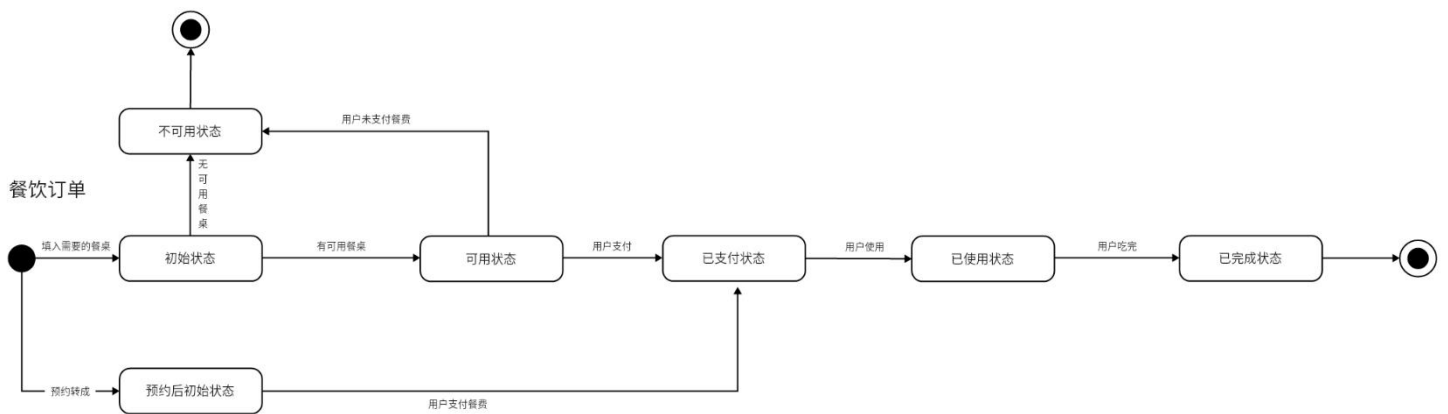


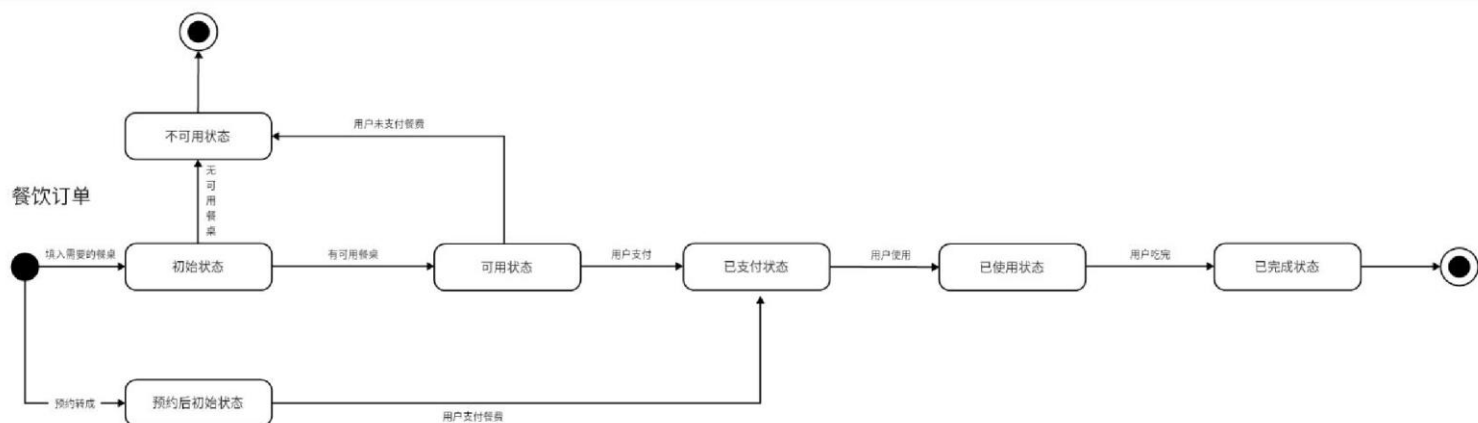
○2 预约订单

○ 3 住宿订单



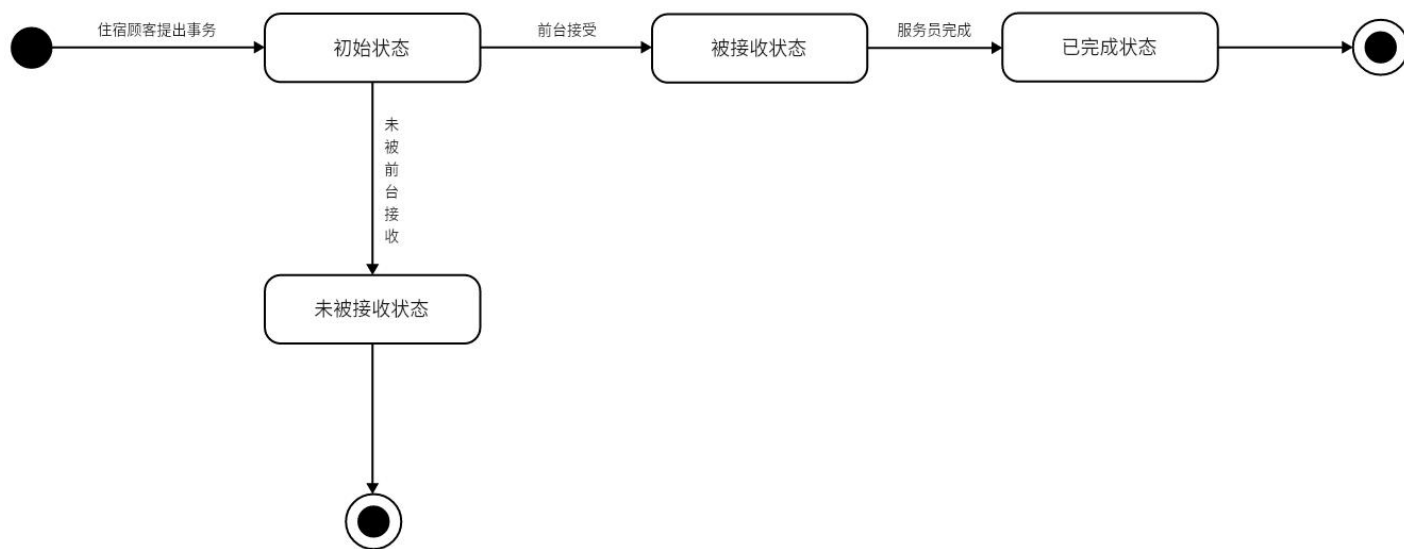
○ 4 餐饮订单





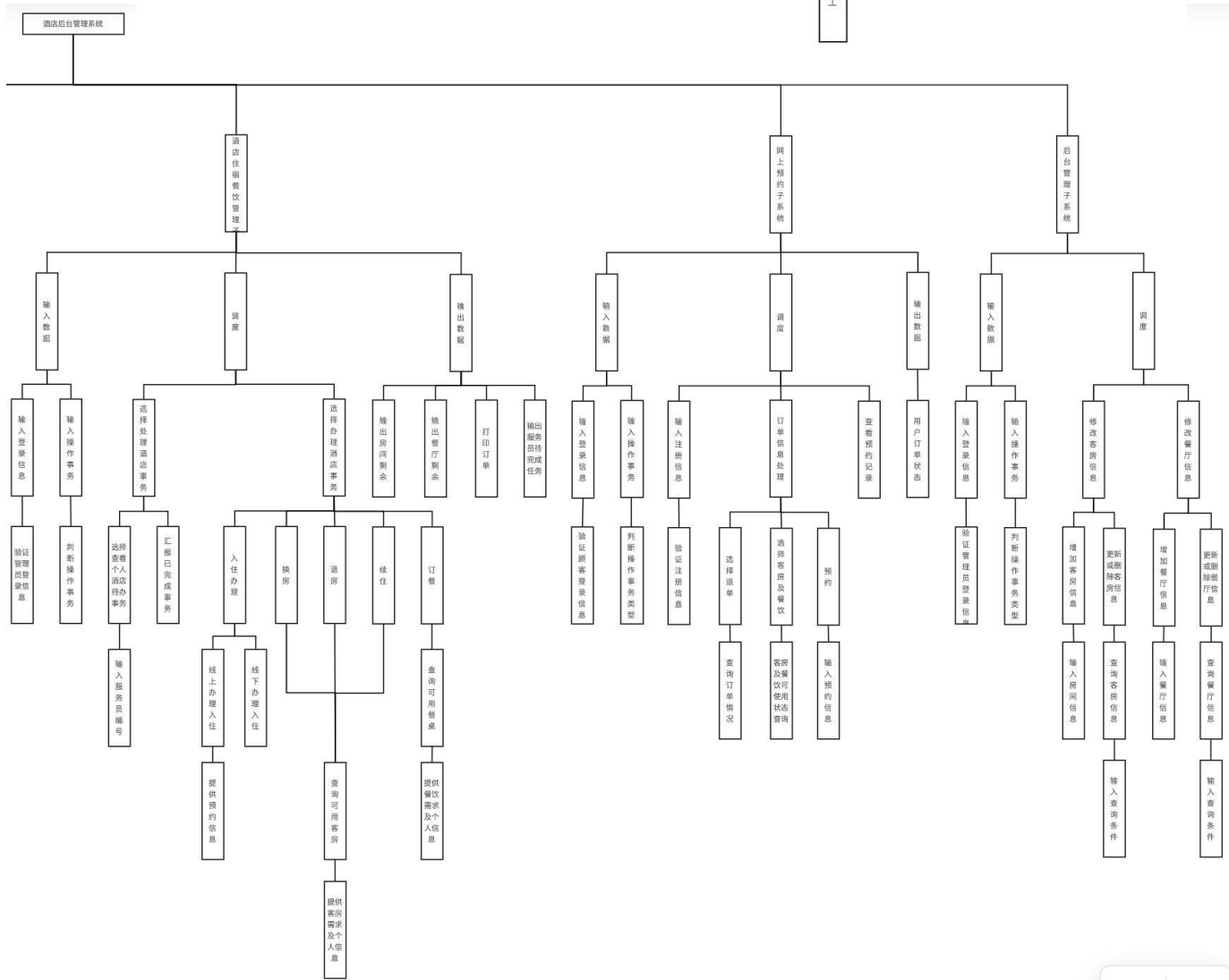
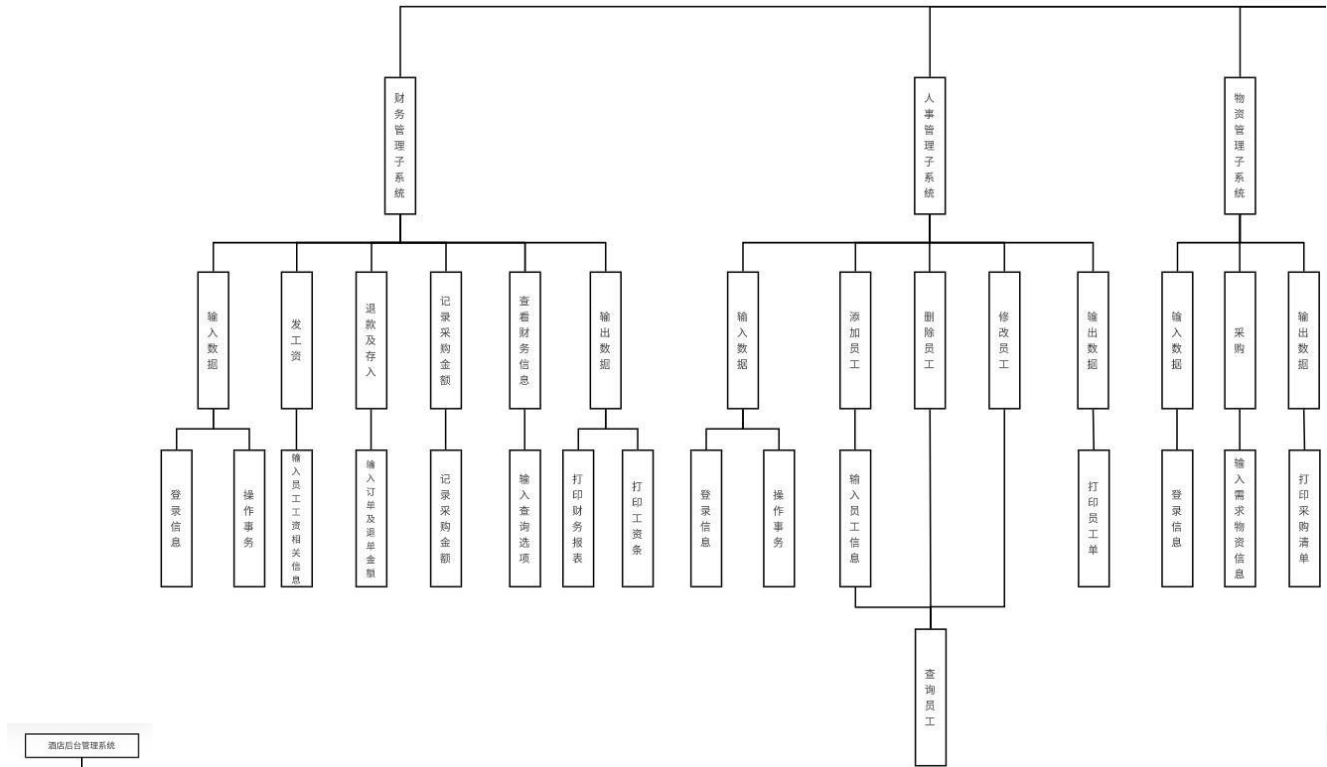
○ 5 采购单 ○ 6 酒店事务记录

酒店事务记录



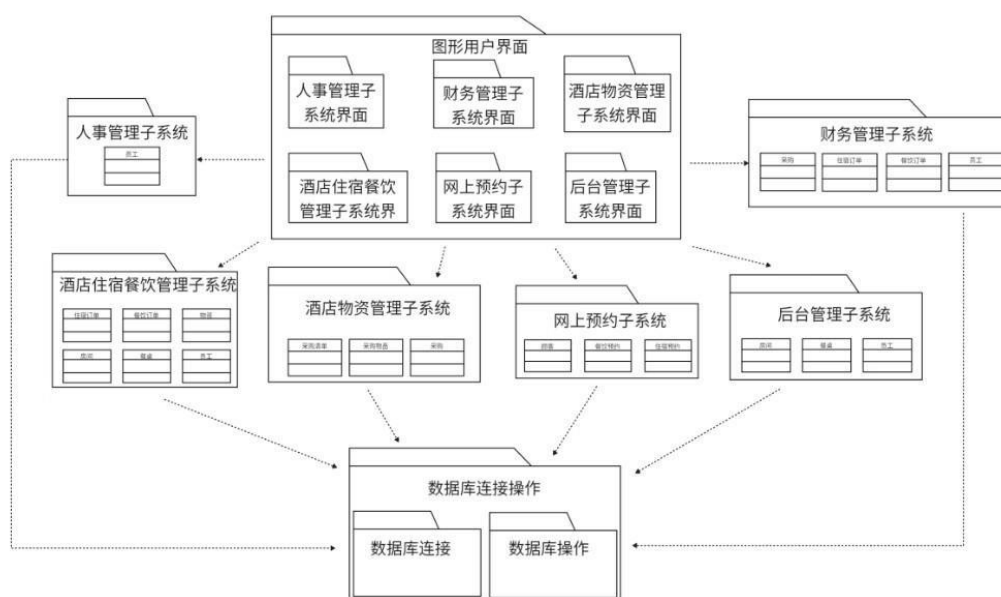
六、体系结构设计

6.1 体系结构总图



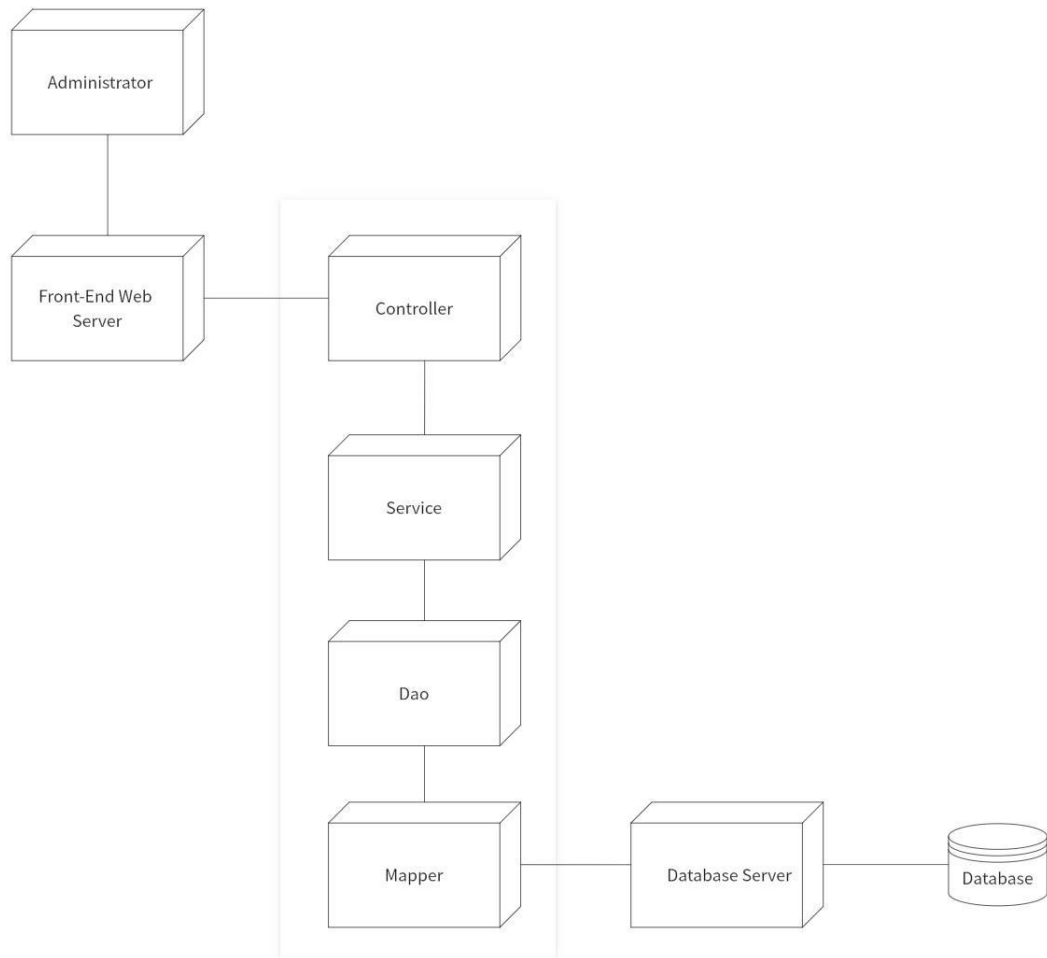
6.2 包的组织与包图

本系统按照子系统的功能以及架构内部的逻辑对各包进行了划分。存放实体类的包中，共分为六个子系统，每个子系统内存放与其密切相关的类。最底层的包是数据库操作包，其中封装了各种数据库连接和数据库操作接口。包图如下：



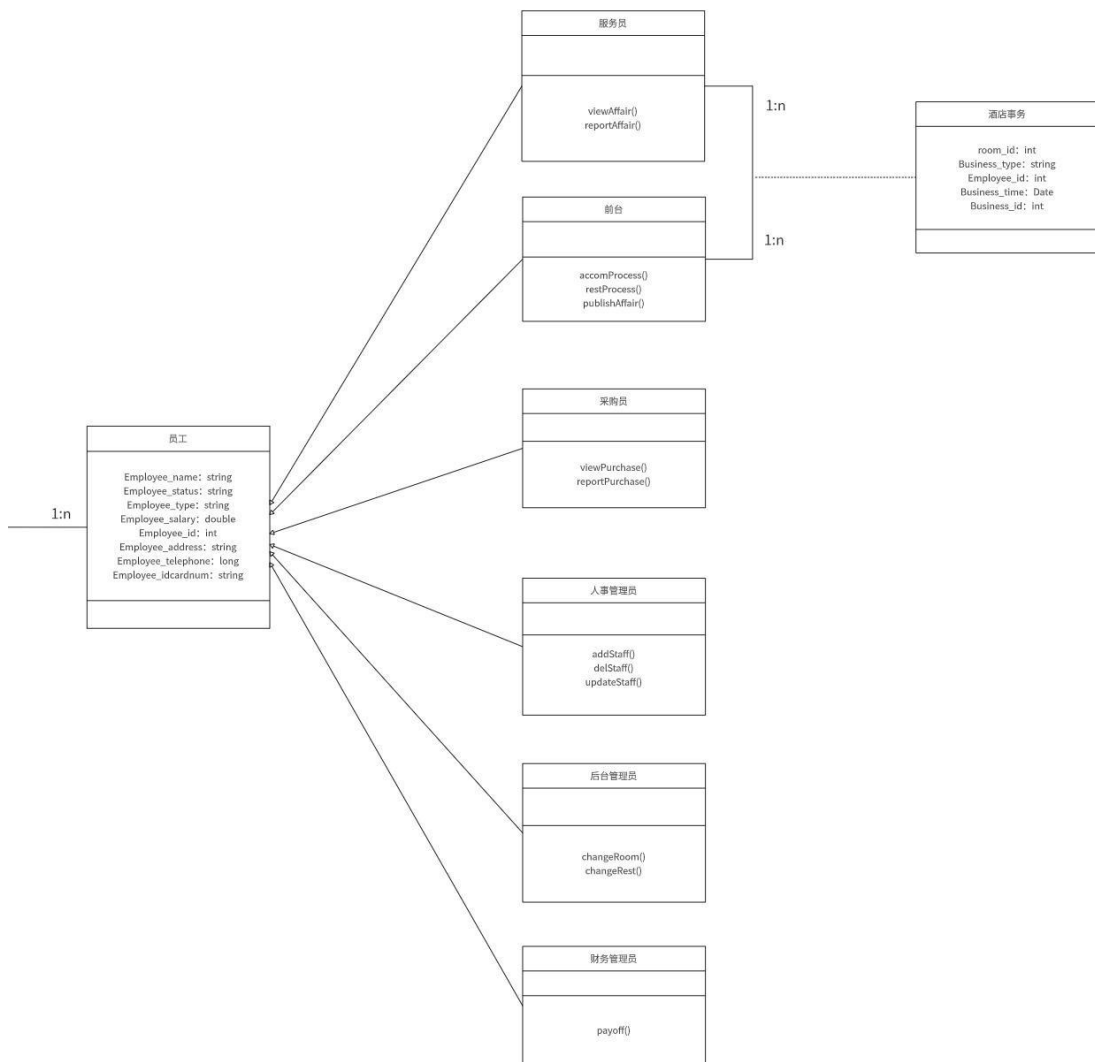
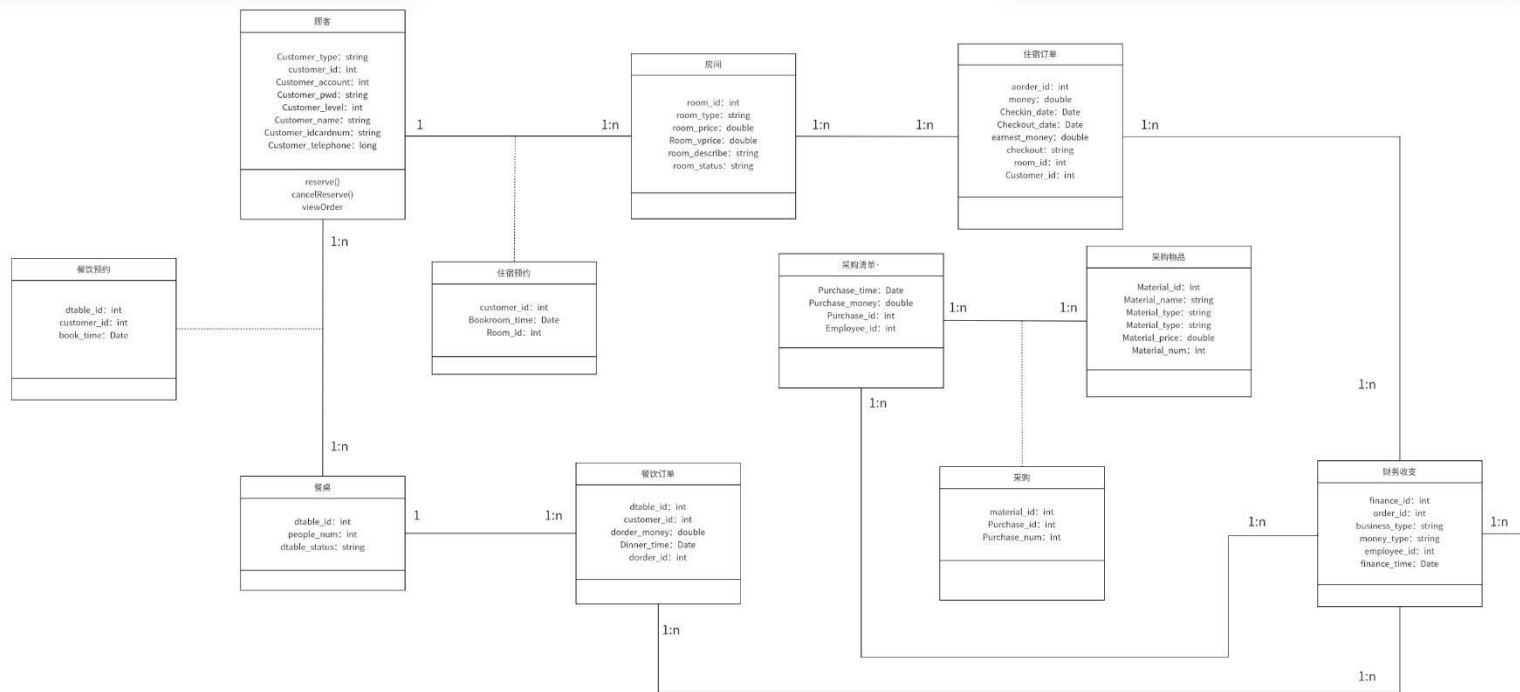
6.3 部署图

整个系统的部署图如下。对于数据库，与后端的 SpringBoot 项目中相应的 mapper 包通过 mybatis 框架进行通信；对于前端网页，其与后端服务器通过 http 请求与响应进行网络间通信。



七、对象建模-细化类图

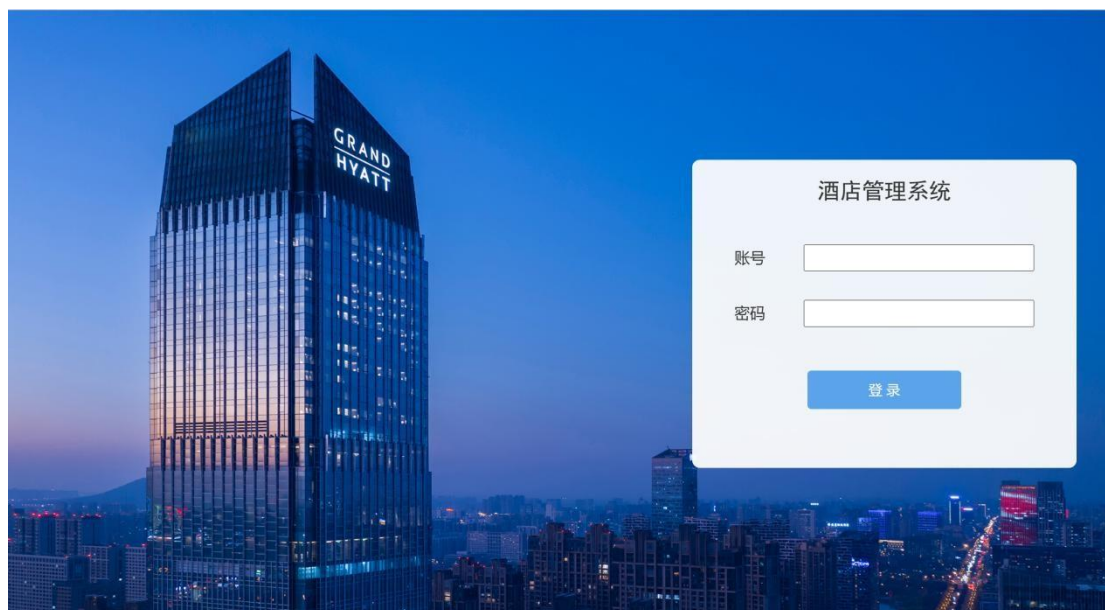
在细化类图中，每个类的属性都添加了数据类型，同时也为部分实体类添加了操作



八、接口设计

8.1 人机交互接口设计

①管理员登录界面



②可用房间信息查询界面



③前台办理线下入住界面

酒店住宿餐饮管理系统

房间剩余

办理业务

订单查改

综合事务系统

你好,

线下入住办理

* 姓名

李小明

* 身份证号

410425200001010010

* 性别

男

* 手机号

13523422333

* 房间类型

大床房

* 住宿时间

2023-04-18 至 2023-04-19

需要早餐

☐

生成订单

重置

订单详情

姓名: 李小明

身份证号: 410425200001010010

性别: 男

手机号: 13523422333

房间类型: 大床房

住宿时间: 2023-04-18 至 2023-04-19

房间号: 1001

是否含早餐: 是

订单总金额: 280

交易完成确认订单

清空订单

④前台查询餐厅余位信息界面

酒店住宿餐饮管理系统

房间剩余

办理业务

订单查改

综合事务系统

你好,

餐厅余位查询

线下办理入住

预订快速入住

退房办理

换房办理

续住办理

餐饮业务

餐厅余位查询

餐桌办理

当前供餐:

当前空座:

选择查询日期

开始日期 至 结束日期

选择餐饮类型

请选择餐饮类型

查询

重置

⑤前台安排服务事务界面

酒店住宿餐饮管理系统

房间剩余

办理业务

订单查改

综合事务系统

你好,

服务需求发布

完成任务汇报

发布

顾客姓名

房间号

事务详述

发布

⑥订单查询修改界面

酒店住宿餐饮管理系统

房间剩余

办理业务

订单查改

综合事务系统

你好,

订单查改

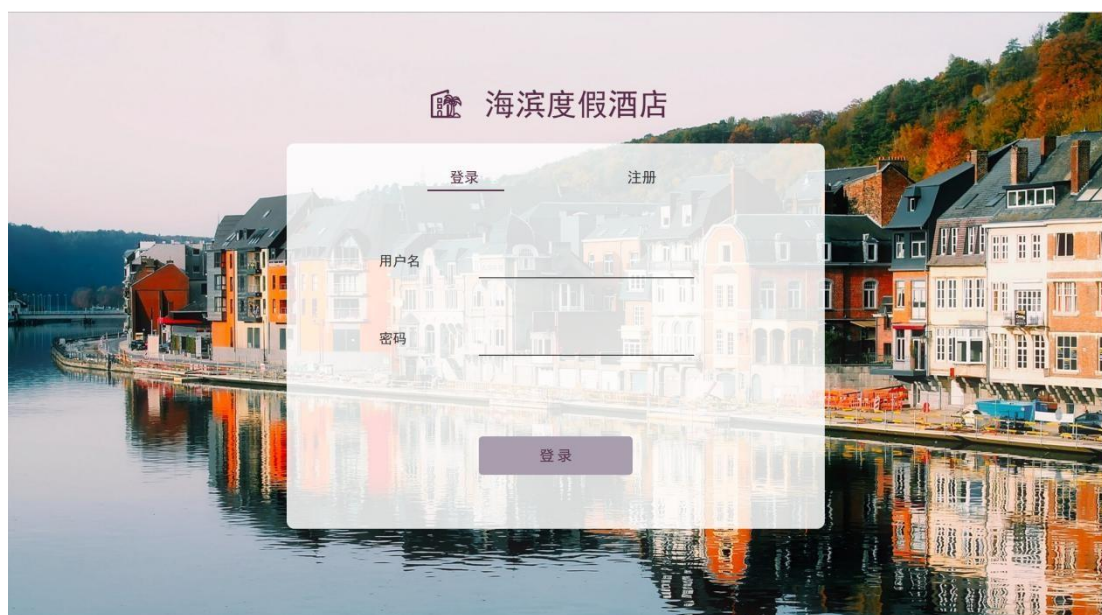
身份证号

410425200000000000

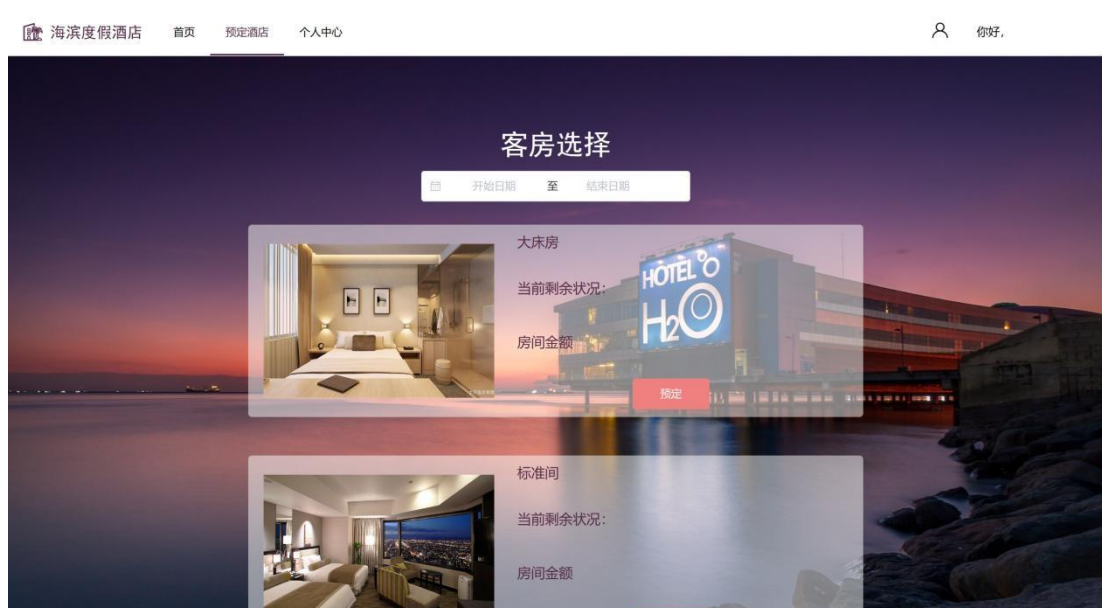
查询

| 身份证号 | 订单编号 | 日期 | 姓名 | 订单类型 | 订单途径 | 操作 |
|--------------------|----------------|------------|----|--------|------|-------|
| 410425200000000000 | 20230404010001 | 2023-04-04 | 张三 | 住宿订单 | 线上订单 | 查看 编辑 |
| 410425200000000000 | 20230404020001 | 2023-04-04 | 张三 | 餐饮订单 | 线上订单 | 查看 编辑 |
| 410425200000000000 | 20221206020004 | 2022-12-06 | 张三 | 过期餐饮订单 | 线上订单 | 查看 编辑 |
| 410425200000000000 | 20221206010004 | 2022-12-06 | 张三 | 过期住宿订单 | 线上订单 | 查看 编辑 |

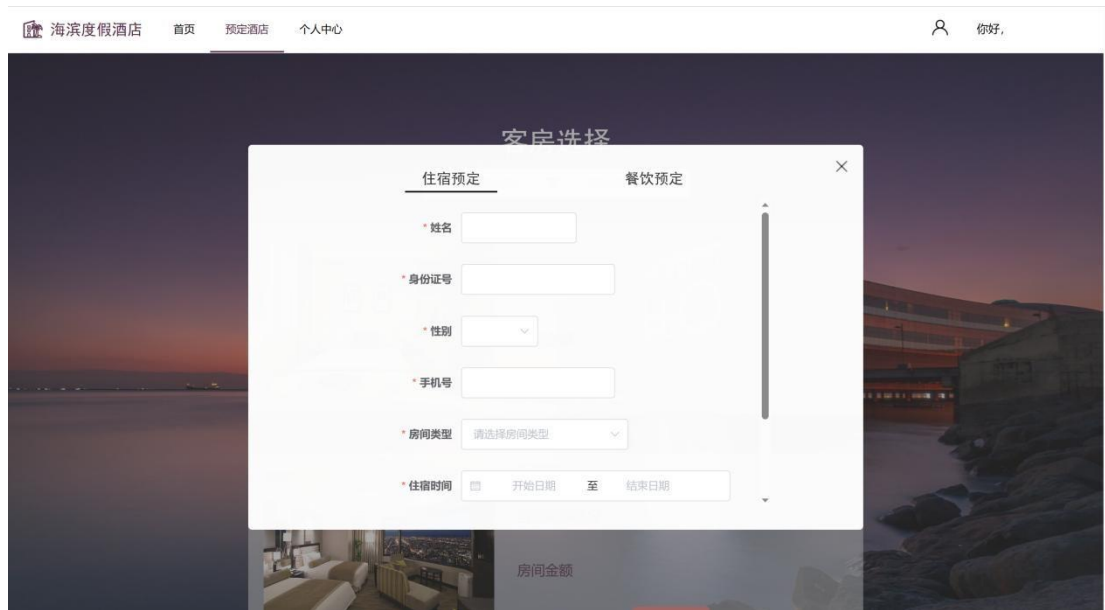
⑦客户登录/注册界面



⑧客户选择客房界面



⑨客户预订客房/餐饮界面



8.2 模块间接口设计

①查询员工与删除/修改员工之间的接口

描述：人事管理员进入人事管理子系统后可以对员工进行查询删除与修改

实现：构造方法的参数传递例：删除数据库中对应的员工

```
public void deleteDish() {  
    //删除数据库中对应的员工  
    delete from Employee where id=Employeeid;  
}
```

例：修改数据库中对应的员工

```
public void updateDish() {  
    //修改数据库中对应的员工  
    update Employee set name="aaa" where id=Employeeid;  
}
```

②查询房间/餐桌与删改房间/餐桌

描述：删除/修改某一个房间或餐桌需要查询出一个特定的位置。此处需要接口传递房间号或餐桌号。

实现：成员变量 inputId 用来保存被点击房间/餐桌的 id

```
private Integer inputId=null;//若不选中默认为 null
```

```
everyid.isSelecteddefaulttofalse;
```

```
If(inputOperation) {
```

```
    getInputId//通过输入获取到餐桌的 id
```

```
    Foreach(id) {
```

```
        If(thosefitids==inputId) { //房间号/餐桌号契合 id 值的
```

```
        thisspecificid.isSelected=true; //选中
```

```
        //删除数据库中对应的房间或餐桌
```

```
        deletefromdatabasewhereid.isSelected=true;
```

```
        //修改数据库中对应的房间或餐桌
```

```
        Updatedatabaseset... (newDatas)whereid.isSelected=true; }
```

```
    }
```

```
}
```

8.3 应用服务器与数据库通信接口

①mysql-connector-j-8.0.32.jar，内含 Java 程序与 MySQL 数据库

的通信协议与方法，位于 External Libraries 文件夹下 ②mybatis-

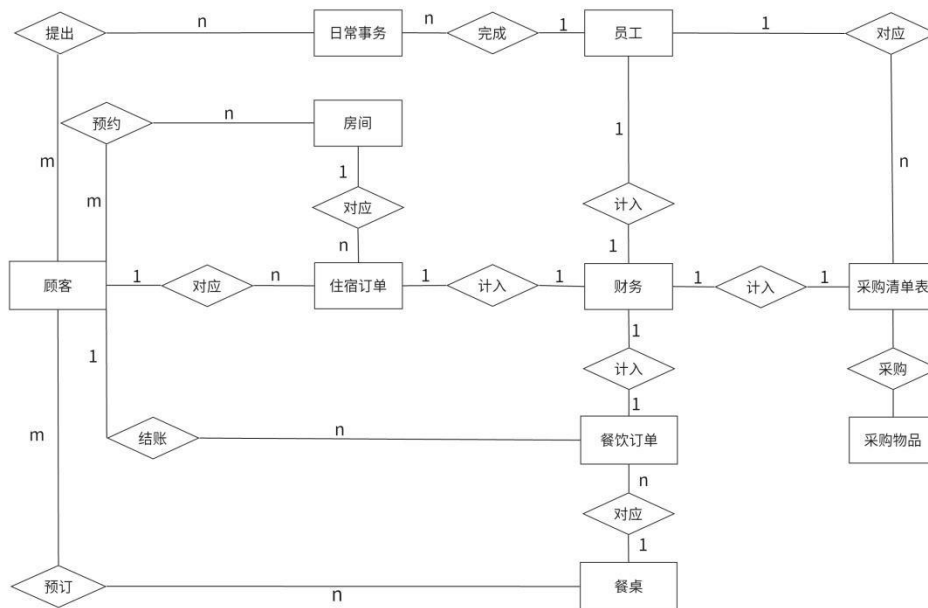
3.5.11.jar，支持自定义 SQL、存储过程以及高级映射。

MyBatis 可以通过简单的 XML 或注解来配置和映射原始类型、接口

和 Java POJO 为数据库中的记录。

九、数据设计

9.1 ER 关系图



9.2 关系模式的建立

实体表

1. 财务表（财务编号，订单表号，事务类型，金额，金额类型，员工编号）
2. 采购清单表（采购清单编号，员工编号，采购时间，采购总金额）
3. 餐饮订单表（餐饮订单编号，顾客编号，金额，点餐时间，餐桌表）
4. 餐桌表（桌号人数，是否空闲）
5. 房间表（房间号，正常房间，价格，会员房间价格，房间类型，是否空闲，房间描述）
6. 顾客表（顾客编号，账号，身份证号，顾客类型，密码，顾客等级，姓名，卡余额，电话）
7. 日常事务表（事务编号，事务内容，顾客编号）
8. 物资表（物资编号，物资名称，物资类型，物资价格，到期时间）
9. 员工表（员工编号，员工姓名，员工身份证号，员工地址，员工类型，具体工资，是否空闲）
10. 住宿订单表（订单编号，顾客编号，房间号，总金额，入住时间，到期时间，押金，是否退房）

多对多关系

11. 预定房间表（房间号，顾客编号，预定时间）
12. 采购表（物资编号，采购清单编号，采购数目）
13. 顾客事务（事务编号，顾客编号，事务内容）
14. 预定餐桌表（桌号，顾客编号，预定时间）

9.3 数据库中表的实现

| 财务表 | 酒店的财务信息 | |
|------------------------|-------------|---|
| 列名 | 数据类型 | 约束 |
| 财务编号 fianance_id | int | Primary key |
| 订单编号 order_id | Int | |
| 事务类型 business_type | Varchar(20) | business_type varchar(20) check(business_type = '餐饮订单收入' or business_type = '住宿订单收入' or business_type = '员工共工资支出' or business_type = '采购支出') |
| 金额 money_type | Varchar(4) | |
| 员工编号 employee_id | int | |
| 财务事务时间 finance_time | date | |
| 房间表 | | |
| 列名 | 数据类型 | 约束 |
| 房间号 room_id | int | Primary key |
| 房间类型 | varchar(10) | check(room_type = '单人间' or room_type = '大床房') |

| 顾客表 | | |
|--------------------------|-------------|-------------------------------|
| 列名 | 数据类型 | 约束 |
| 顾客类型 Customer_type | Varchar(10) | |
| 顾客编号 customer_id | int | Primary key |
| 顾客账号 Customer_account | Int | |
| 顾客密码 Customer_pwd | Varchar(40) | check(length(customer_pwd)>6) |

| | | |
|------------------------|-------------|---|
| 顾客等级 Customer_level | Int | |
| 顾客姓名 | Varchar(10) | |
| room_type | | or room_type = '豪华间' or room_type = '三人间' or room_type = '双人间') |
| 正常房间价格 room_price | Double | |
| 会员房间价格 Room_vprice | double | |
| 房间描述 room_describe | Varchar(40) | |
| 是否空闲 room_status | Varchar(2) | check(room_status = '是' or room_status = '否') |

| 住宿订单表 | | |
|-----------------------|------------|---|
| 列名 | 数据类型 | 约束 |
| 订单编号 aorder_id | Int | primary key |
| 金额 money | Double | |
| 入住时间 Checkin_date | Datetime | |
| 到期时间 Checkout_date | Datetime | |
| 押金 earnest_money | double | |
| 是否退房 checkout | varchar(2) | check(checkout = '是' or checkout = '否') |
| 房间号 room_id | Int | foreign key (room_id) references tb_room(room_id) |

| | | |
|---------------------|-----|---|
| 顾客编号 Customer_id | Int | foreign key (customer_id) references tb_customer(customer_id) |
|---------------------|-----|---|

[illegible]

| | | |
|-------------------------|-------------|---|
| 采购总金额 Purchase_money | Double | |
| 采购清单编号 Purchase_id | Int | Primary key |
| 员工编号 Employee_id | int | foreign key (employee_id) references tb_employee(employee_id) |
| 物资表 | | |
| 列名 | 数据类型 | 约束 |
| 物资编号 Material_id | Int | Primary key |
| 物资名称 | Varchar(20) | |

| | | |
|------------------------|-------------|--|
| Material_name | | |
| 物资类型 Material_type | Varchar(20) | |
| 物资价格 Material_price | Double | |
| 物资数量 Material_num | Int | |

| | | |
|-------------------------|------------|---|
| 酒店事务表 | | |
| 列名 | 数据类型 | 约束 |
| 房间号 room_id | int | foreign key (room_id) references tb_room(room_id) |
| 是否完成 Business_type | Varchar(2) | check(business_status = '是' or business_status = '否') |
| 完成员工编号 Employee_id | Int | foreign key (employee_id) references tb_employee(employee_id) |
| 事务发起时间 Business_time | Datetime | |
| 事务编号 Business_id | Int | Primary key |

| | | |
|--------------------|------|-------------|
| 餐桌号 | | |
| 列名 | 数据类型 | 约束 |
| 桌号 dtable_id | Int | Primary key |
| 就餐人数 people_num | Int | |

| | | |
|--------------------------|-------------|--|
| 是否空闲 dtable_status | Varchar(2) | check(dtable_status = '是' or dtable_status = '否') |
| 餐饮订单表 | | |
| 列名 | 数据类型 | 约束 |
| 餐桌号 dtable_id | int | foreign key (dtable_id) references tb_dtable(dtable_id) |
| 顾客编号 customer_id | Int | foreign key (customer_id) references tb_customer(customer_id) |
| 金额 dorder_money | Double | |
| 点餐时间 Dinner_time | Datetime | |
| 订单编号 dorder_id | int | Primary key |
| 预定房间表 | | |
| 列名 | 数据类型 | 约束 |
| 顾客编号 customer_id | int | foreign key (customer_id) references tb_customer(customer_id) |
| 预定时间 Bookroom_time | Datetime | |
| 房间号 Room_id | int | foreign key (room_id) references tb_room(room_id) |
| 采购表 | | |
| 列名 | 数据类型 | 约束 |
| 物资编号 material_id | Int | foreign key (material_id) references tb_material(material_id) |
| 采购清单编号 Purchase_id | int | foreign key (purchase_id) references tb_purchase(purchase_id) |
| 采购数目 Purchase_num | int | |
| 顾客事务细节表 | | |
| 列名 | 数据类型 | 约束 |
| 事务编号 business_id | Int | foreign key (business_id) references tb_business(business_id) |
| 事务内容 business_content | Varchar(40) | |

| | | |
|---------------------|----------|--|
| 顾客编号 customer_id | int | foreign key (customer_id) references tb_customer(customer_id) |
| 预订餐桌表 | | |
| 列名 | 数据类型 | 约束 |
| 桌号 dtable_id | Int | foreign key (dtable_id) references tb_dinnertable(dtable_id) |
| 预订时间 book_time | Datetime | |
| 顾客编号 customer_id | Int | foreign key (customer_id) references tb_customer(customer_id) |

9.4 建表语句 (MySQL)

#财务表

```
create table tb_finance(
finance_id int primary key, #财务编号 finance_time date, #财务事务时间 money_type
varchar(4), #金额类型 order_id int, #订单编号 employee_id int, #员工编号
business_type varchar(20)
check(business_type = '餐饮订单收入' or business_type = '住宿订单收入' or
business_type = '员工共工资支出' or business_type = '采购支出') #事务类型
);
```

#房间表

```
create table tb_room( room_id int primary key, #房间号 room_type varchar(10)
check(room_type = '单人间' or room_type = '大床房' or room_type = '豪华间' or
room_type = '三人间' or room_type = '双人间'), #房间类型 room_price double,
room_vprice double, room_status varchar(2) check(room_status = '是' or room_status =
'否'), room_describe varchar(40)
);
```

#住宿订单表

```
create table tb_aorder( room_id int, customer_id int, aorder_id int
primary key, money double, checkin_date datetime, checkout_date
datetime, earnest_money double, #押金 checkout varchar(2)
check(checkout = '是' or checkout = '否'), #是否退房 foreign key
(room_id) references tb_room(room_id), foreign key (customer_id)
references tb_customer(customer_id)
);
```

#顾客表

```
create table tb_customer( customer_id int primary key,
customer_pwd varchar(40) check(length(customer_pwd)>6),
customer_level int, customer_name varchar(10),
customer_idcardnum          varchar(36)          check(customer_idcardnum          like
'[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9X]')
, #身份证号
customer_cardbalance double, customer_telephone long
check(length(customer_telephone) <= 11), customer_account int,
customer_type varchar(10)
);
```

#员工表

```
create table
tb_employee( employee_id int primary
key, employee_name varchar(10),
employee_idcardnum          varchar(36)          check(employee_idcardnum          like
'[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9X]')
,
employee_status varchar(2) check(employee_status = '是'or employee_status = '否'), #是否空
闲
employee_type varchar(20) check(employee_type = '人事管理员'or employee_type = '财务管
理员'or employee_type = '前台管理员'or employee_type = '采购人员'or employee_type = '后台
管理员'or employee_type = '服务员'), employee_salary double, employee_address varchar(60),
employee_telephone long check(length(employee_telephone) <= 11)
);
```

#采购清单表

```
create table tb_purchase( employee_id int, purchase_id int
primary key, purchase_time datetime, purchase_money double ,
foreign key (employee_id) references tb_employee(employee_id)
);
```

#物资表

```
create table tb_material( #物资表*增加了一个物资数量
material_name varchar(20),
material_type varchar(20),
```

material_id int primary key,

material_price double,

material_num int

);

#酒店事务表

create table tb_business(#酒店事务表 business_id int primary key,

business_time datetime, room_id int, employee_id int, business_status

varchar(2) check(business_status = '是' or business_status = '否'), foreign key

(room_id) references tb_room(room_id), foreign key (employee_id) references

tb_employee(employee_id)

);

#餐桌表

create table tb_dinnertable(dtable_id int primary key, people_num int,

dtable_status varchar(2) check(dtable_status = '是' or dtable_status = '否')

);

#餐饮订单表

create table tb_dorder(dorder_id int primary key, customer_id

int, dorder_money double, dinner_time datetime, dtable_id int,

foreign key (customer_id) references tb_customer(customer_id),

foreign key (dtable_id) references tb_dtable(dtable_id)

);

#预定房间关系表

create table tb_bookroom(bookroom_time datetime,

customer_id int, room_id int, foreign key (customer_id)

references tb_customer(customer_id), foreign key (room_id)

references tb_room(room_id)

);

#采购关系表

create table tb_buymaterial(purchase_id int, material_id int,

purchase_num int, foreign key (purchase_id) references

tb_purchase(purchase_id), foreign key (material_id) references

tb_material(material_id)

);

#事务顾客对应关系表

```
create table tb_businessdetail( business_id int, customer_id int,  
business_content varchar(40), foreign key (customer_id)  
references tb_customer(customer_id), foreign key (business_id)  
references tb_business(business_id)  
);
```

#预定餐桌表

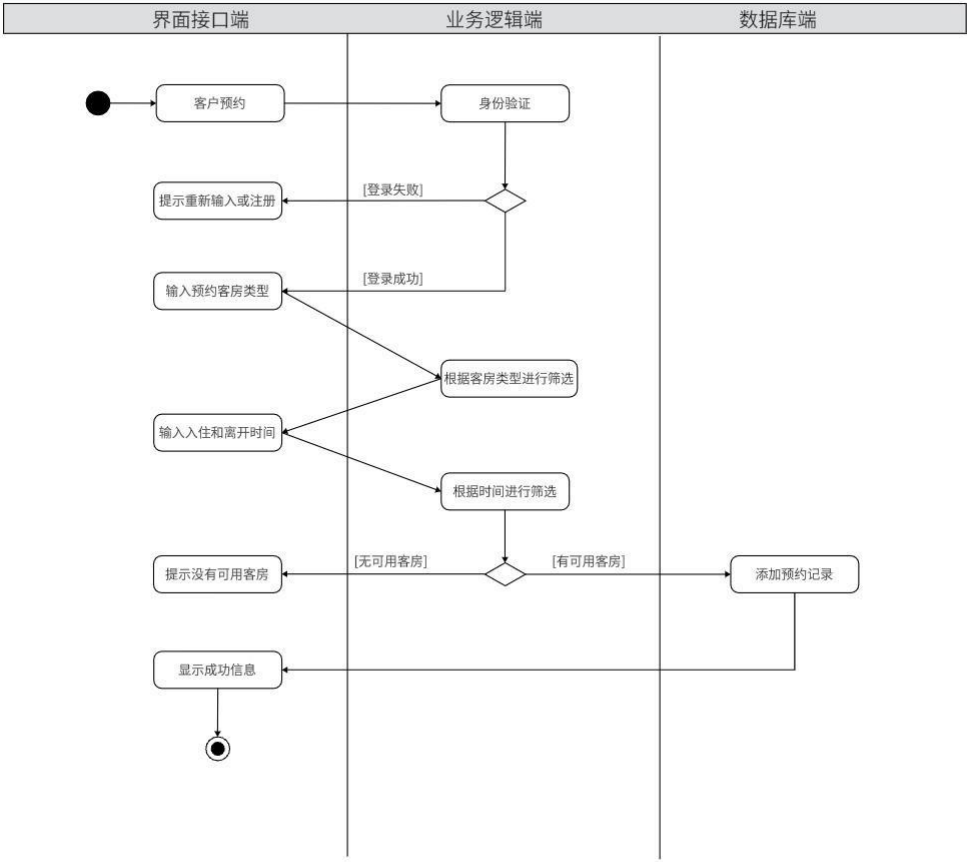
```
create table  
tb_booktable( customer_i  
d int, dtable_id int,  
book_time datetime,  
foreign key (customer_id)  
references  
tb_customer(customer_id)  
, foreign key (dtable_id)  
references  
tb_dinnertable(dtable_id)  
);
```

十、过程设计

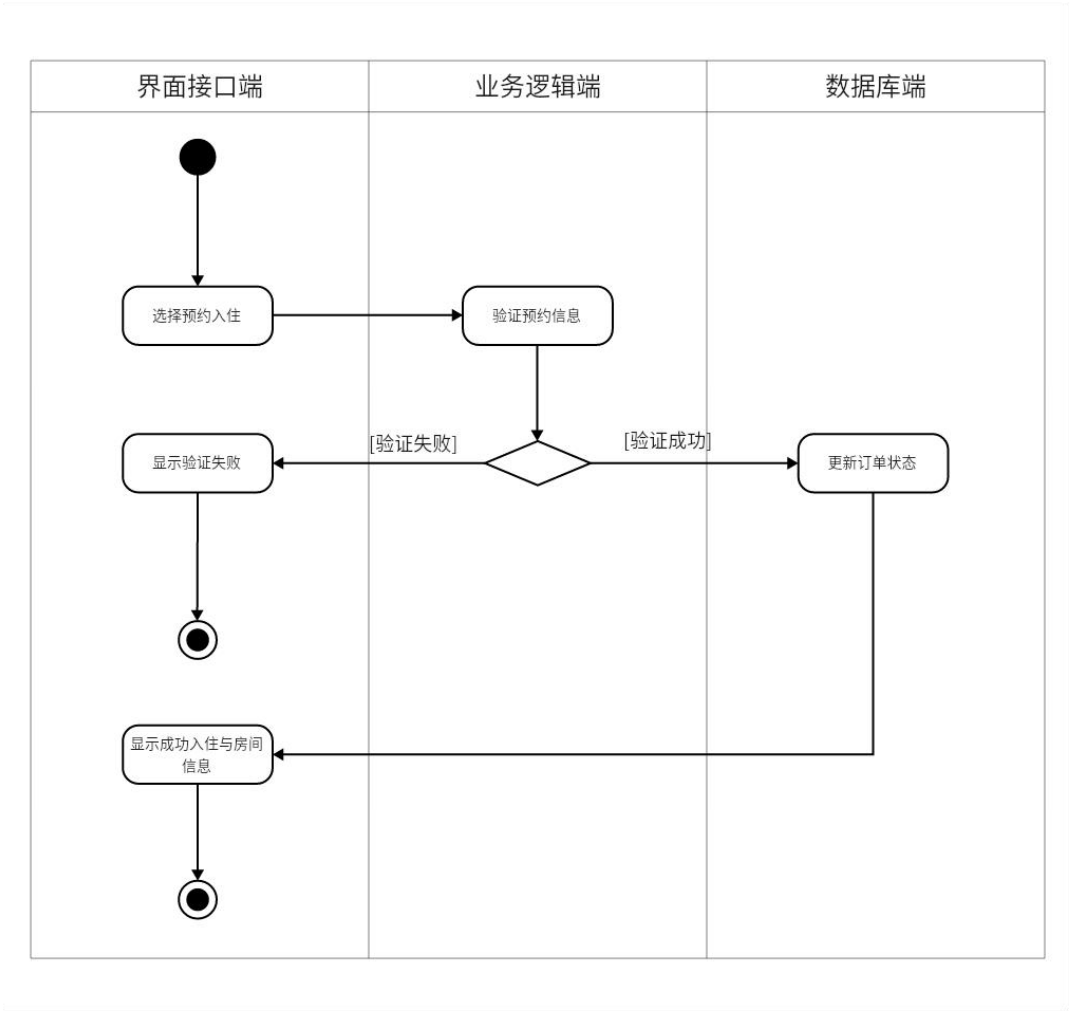
10.1 活动图

活动图具体描述了事务执行过程中的执行顺序，在下列活动图中，我们划分出 3 个泳道，依次为界面接口端，业务逻辑端和数据库端。

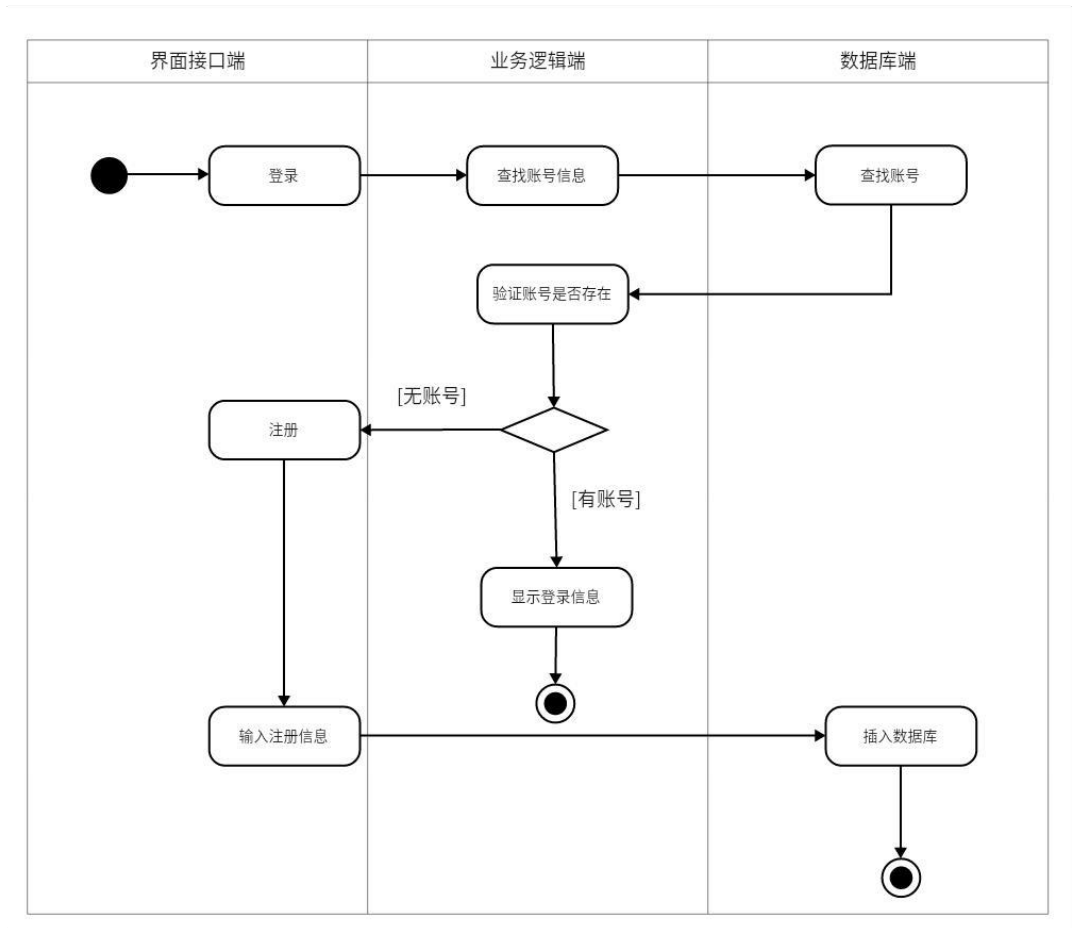
○ 1 预约活动图



○ 2 预约入住



○ 3 用户登录注册



○ 4 用户升级

