

酒店管理系统系统建模报告

成员列表

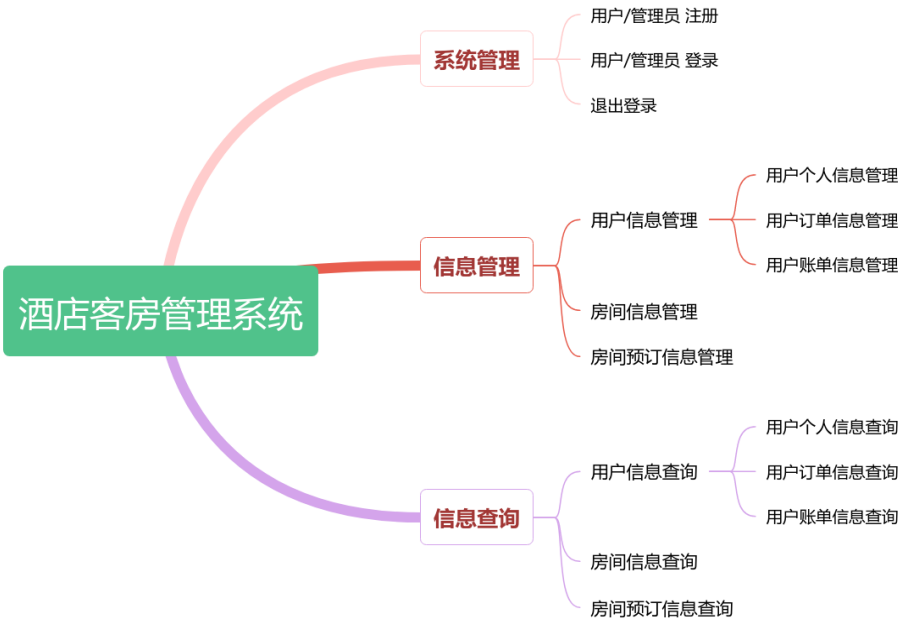
学号	姓名
21307037	金思琪
21307056	钟欣余
21307062	郑越
21307035	邓栩瀛
21307026	何倩盈

系统设计

用例图

酒店客房管理系统的用例图，包含主要的用户和他们的操作。

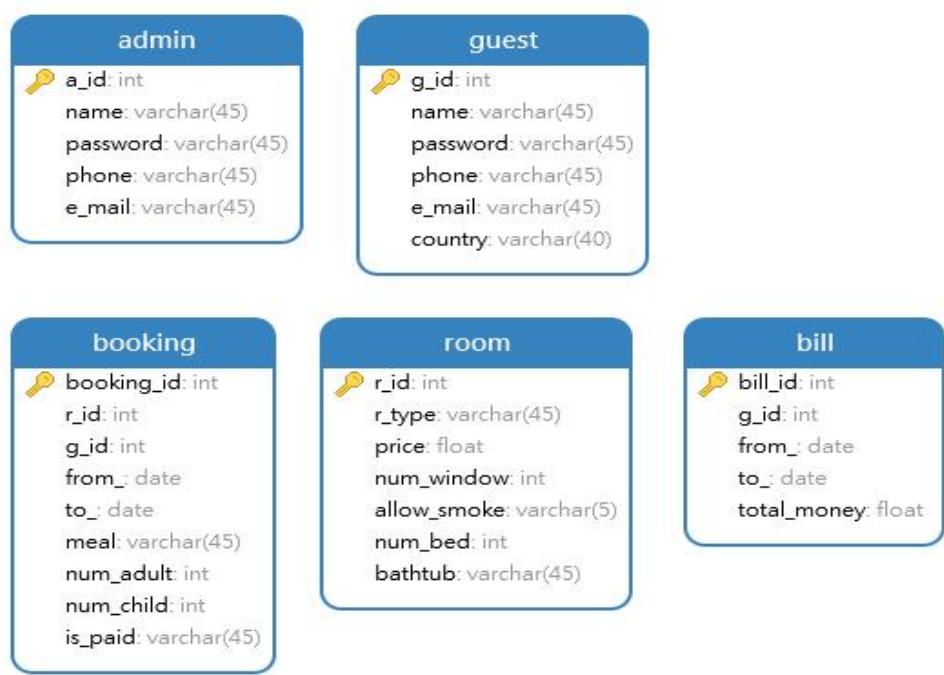
- 用户/管理员
- 用例：注册、登录、查看房间、预订房间、取消预订、查看订单、生成账单、添加房间、编辑房间、删除房间、查看用户信息、编辑用户信息、删除用户信息等。



类图

酒店客房管理系统的类图，包括主要的类及其属性和方法。

- 类：用户、管理员、客房、预订、账单
- 属性：各类的具体属性
- 方法：各类的具体方法

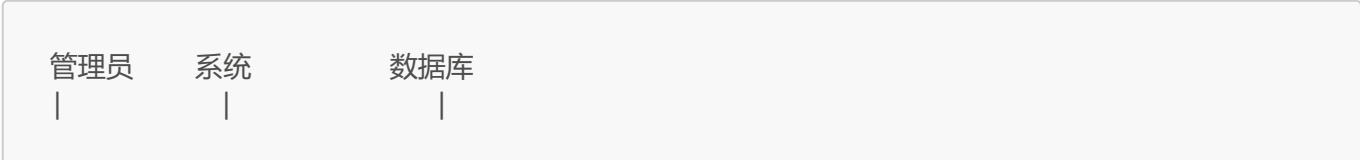


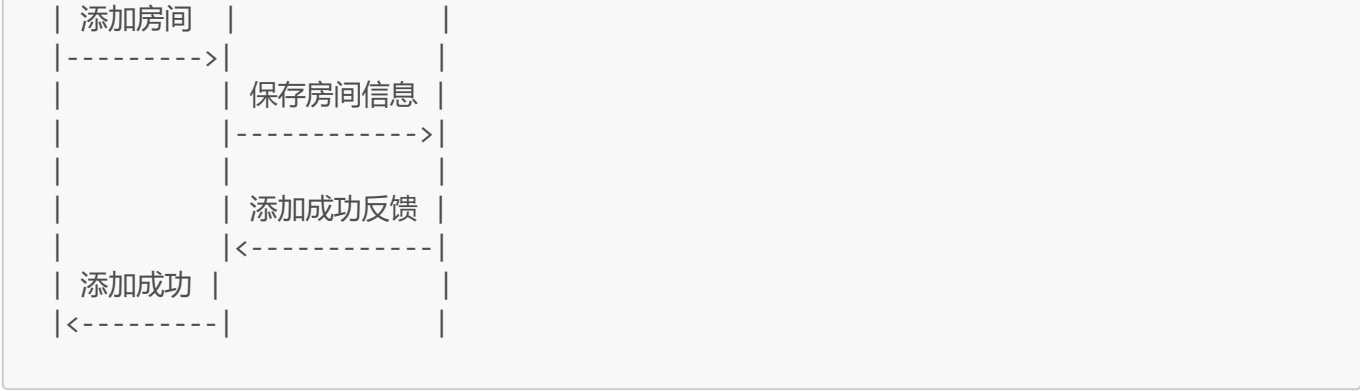
序列图

• 用户预订房间



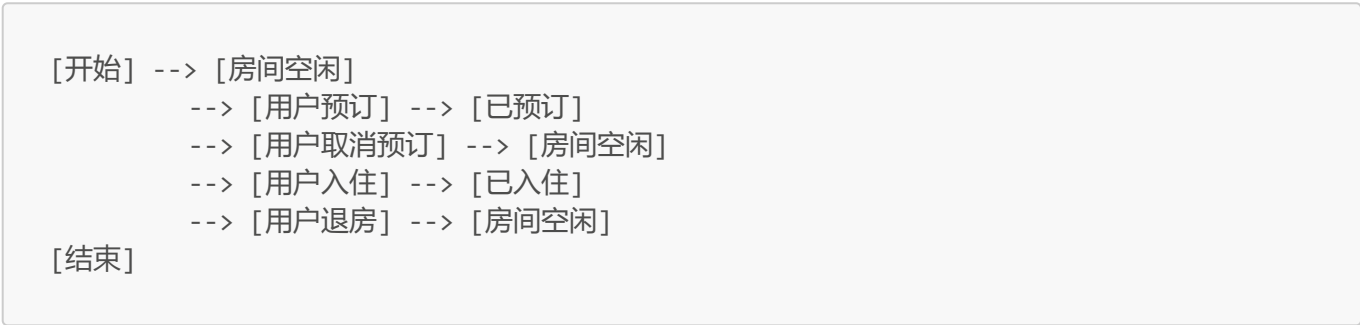
• 管理员添加房间





状态图

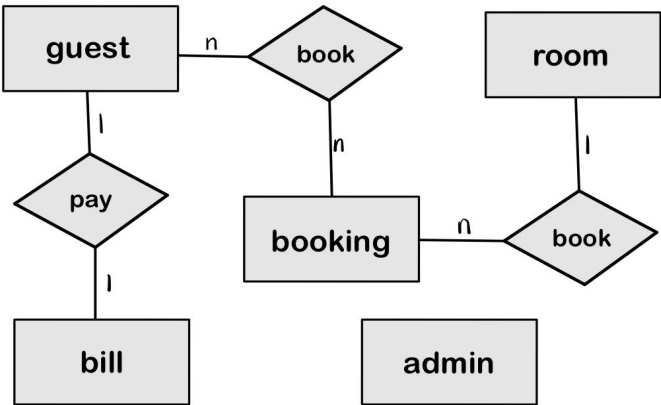
• 预订房间



数据库设计

ER图

酒店客房管理系统的ER图如下：



关系分析

- 1. **一对一关系 (One-to-One Relationship)**：每个客人 (guest) 都有一个对应的账单 (bill)，同时每个账单只属于一个客人。

2. **一对多关系 (One-to-Many Relationship)** : 每个房间 (room) 可以对应多个预订 (booking) , 但是每个预订只能对应一个房间。
3. **多对多关系 (Many-to-Many Relationship)** : 每个客人 (guest) 可以有多个预订 (booking) , 同时每个预订也可以对应多个客人。

数据库创建

```
-- 创建账单表
CREATE TABLE bill(
    bill_id INTEGER PRIMARY KEY, -- 账单ID, 作为主键
    g_id INTEGER REFERENCES guest(g_id), -- 外键, 关联到客人表的g_id
    from_ DATE, -- 账单开始日期
    to_ DATE, -- 账单结束日期
    total_money FLOAT); -- 账单总金额

-- 创建房间表
CREATE TABLE room(
    r_id INTEGER PRIMARY KEY, -- 房间ID, 作为主键
    r_type VARCHAR(45), -- 房间类型
    price FLOAT, -- 房间价格
    num_window INTEGER, -- 窗户数量
    allow_smoke VARCHAR(5), -- 是否允许吸烟
    num_bed INTEGER, -- 床的数量
    bathtub VARCHAR(45)); -- 是否有浴缸

-- 创建客人表
CREATE TABLE guest(
    g_id INTEGER PRIMARY KEY, -- 客人ID, 作为主键
    name VARCHAR(45), -- 客人姓名
    password VARCHAR(45) NOT NULL, -- 客人密码 (不能为空)
    phone VARCHAR(45) NOT NULL, -- 客人电话 (不能为空)
    e_mail VARCHAR(45), -- 客人电子邮件地址
    country VARCHAR(40)); -- 客人所在国家

-- 创建管理员表
CREATE TABLE admin(
    a_id INTEGER PRIMARY KEY, -- 管理员ID, 作为主键
    name VARCHAR(45) NOT NULL, -- 管理员姓名 (不能为空)
    password VARCHAR(45) NOT NULL, -- 管理员密码 (不能为空)
    phone VARCHAR(45) NOT NULL, -- 管理员电话 (不能为空)
    e_mail VARCHAR(45)); -- 管理员电子邮件地址

-- 创建预订表
CREATE TABLE booking(
    booking_id INTEGER PRIMARY KEY, -- 预订ID, 作为主键
    r_id INTEGER REFERENCES room(r_id), -- 外键, 关联到房间表的r_id
    g_id INTEGER REFERENCES guest(g_id), -- 外键, 关联到客人表的g_id
    from_ DATE, -- 预订开始日期
    to_ DATE, -- 预订结束日期
    meal VARCHAR(45), -- 是否包含餐食
    num_adult INTEGER, -- 成人数量
    num_child INTEGER, -- 儿童数量
```

```
is_paid VARCHAR(45
));  -- 是否已支付
```

系统实施

系统实施

系统的实施主要包括以下步骤：

- 1. **需求确认**：与酒店管理层及潜在用户详细沟通，确认所有需求。
- 2. **系统设计**：根据需求进行详细的系统设计，包括架构设计、数据库设计等。
- 3. **编码实现**：根据设计文档进行系统的编码实现。
- 4. **测试阶段**：进行单元测试、集成测试和系统测试，确保系统的功能和性能符合要求。
- 5. **部署上线**：将系统部署到生产环境，并进行上线测试。

系统测试

系统测试主要分为功能测试和性能测试。

- 1. **功能测试**：测试各个功能模块是否能正常工作，包括用户注册、登录、预订房间、取消预订、查看订单、生成账单、管理员添加房间、编辑房间、删除房间、查看用户信息、编辑用户信息、删除用户信息等。
- 2. **性能测试**：测试系统在高并发情况下的性能表现，包括响应时间、吞吐量等指标。

测试用例

1. 用户注册测试

- 测试输入：用户名、密码、电话号码、电子邮件地址等
- 预期输出：用户成功注册，系统返回注册成功信息

2. 用户登录测试

- 测试输入：用户名、密码
- 预期输出：用户成功登录，系统返回登录成功信息并跳转到用户主页

3. 房间预订测试

- 测试输入：房间类型、入住日期、退房日期、成人数量、儿童数量等
- 预期输出：系统返回符合条件的可用房间列表，用户选择房间并确认预订，系统返回预订成功信息

4. 房间取消预订测试

- 测试输入：预订ID
- 预期输出：系统取消对应的预订信息，返回取消预订成功信息

5. 管理员添加房间测试

- 测试输入：房间类型、价格、窗户数量、是否允许吸烟、床的数量、是否有浴缸等

- 预期输出：系统成功添加房间信息，返回添加成功信息

系统维护

维护策略

- 1. **定期备份**：定期备份数据库和系统文件，确保数据安全。
- 2. **监控系统**：实时监控系统运行状态，及时发现并解决问题。
- 3. **用户反馈**：收集用户反馈，持续改进系统功能和性能。
- 4. **安全更新**：定期进行系统安全更新，防止系统被攻击。