

# **“C++程序设计与训练”课程大作业**

## **项目报告**

**项目名称：酒店预订与管理系统**

姓名： 常成

学号： 2017010252

班级： 自 75

日期： 2018 年 8 月 4 日

# 目 录

1 系统功能设计.....	3
1.1 总体功能描述 .....	3
1.2 功能流程描述 .....	4
2 系统结构设计.....	5
2.1 概要设计 .....	5
2.2 模块功能设计.....	5
2.3 数据表设计.....	5
2.4 类的设计.....	6
3 系统详细设计.....	6
3.1 类重点方法的实现 .....	8
3.1.1 搜索功能的实现.....	8
3.1.2 排序功能的实现.....	9
3.1.3 链表功能的实现.....	9
3.1.4 评分功能的实现.....	10
3.1.5 数据库功能的实现.....	10
3.2 界面结构及容错设计.....	10
3.2.1 登录界面.....	10
3.2.2 注册界面.....	11
3.2.3 顾客主界面.....	11
3.2.4 顾客预订界面.....	12
3.2.5 酒店创建及审核界面.....	12
3.2.6 酒店管理及修改信息界面.....	13
3.2.7 顾客及酒店查看订单界面.....	13
4 项目总结.....	14
5 相关问题的说明.....	14

# 1 系统功能设计

酒店预定是出门旅行、办公的重要环节，尤其处在旅游旺季如暑假、五一小长假、十一黄金周等时间段，各大景区的酒店往往会出现一房难求的情况，而当下的酒店预订 APP 如携程、米途、途家等，均有界面繁乱、操作复杂的问题。基于这种考虑，我开发了一款酒店预订与管理系统的软件，既为顾客预订酒店提供便利，又方便酒店方管理酒店信息，同时加入平台系统管理员审核功能，保障酒店信息的真实性，维护消费者的权益。

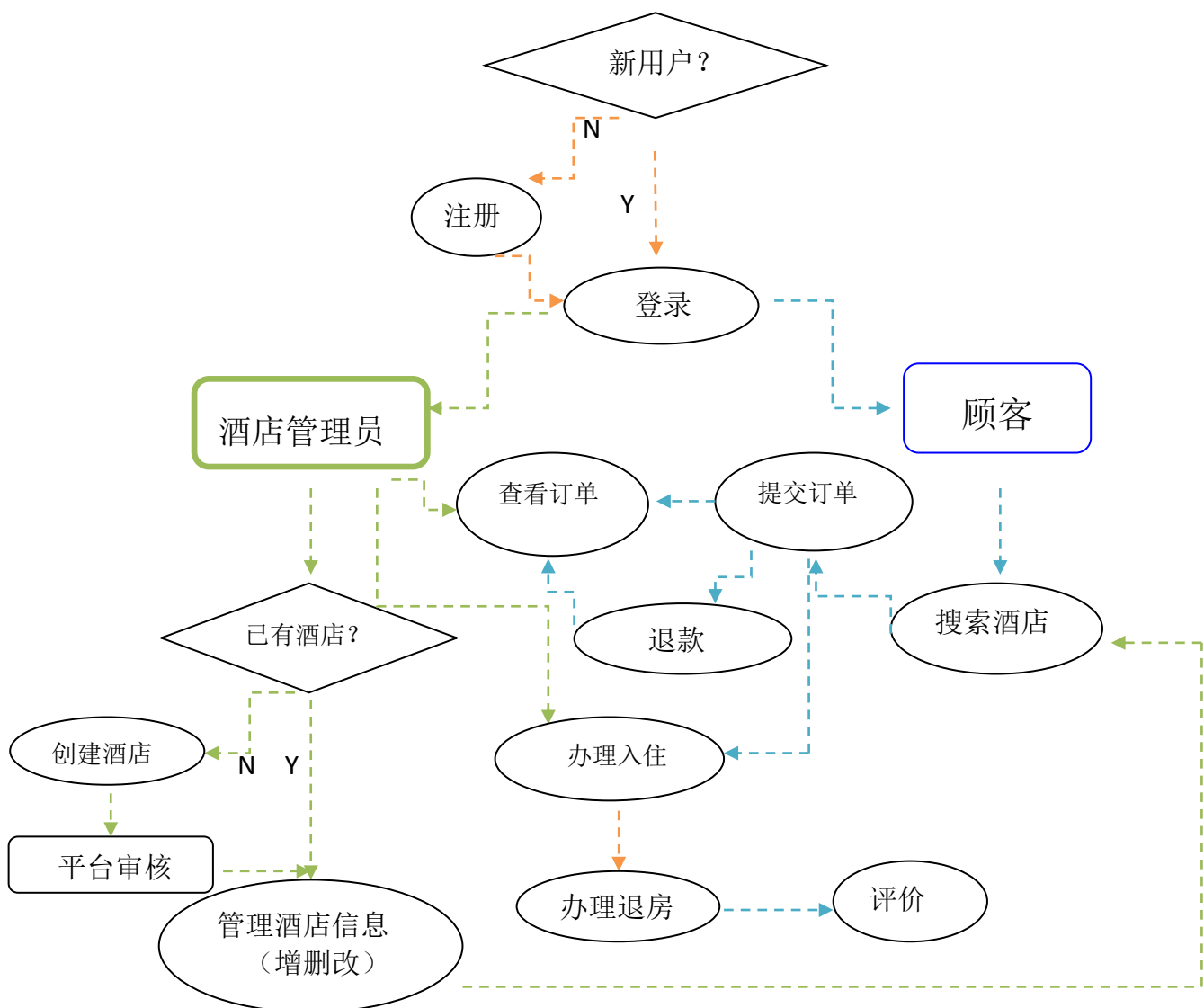
## 1.1 总体功能描述

用户类型	功能名称	功能描述
顾客	注册	新用户点击登录界面的“新用户注册”填写相关信息即可在系统完成注册
	查找酒店	顾客可在主界面获取到酒店信息，可按照城市、酒店星级、酒店所在区域（街道）等信息进行检索，可按照酒店星级或酒店评分对酒店信息进行排序，选中酒店后还可获取酒店具体客房信息（价格、房型、预订状态等），可预览酒店对应客房，实现高效搜索
	预订	顾客填写订单完成预订
	订单查询	在“我的订单”模块顾客可实时获取订单状态（已预订、申请退款中、已入住、已退房等）
	付款	顾客在填写订单后，确认并付款
	退款	顾客在入住之前，可申请退款手续（需相应酒店管理员审核）
	删除订单	顾客可随时删除自己的订单信息（主要为清除长时间冗余的订单），如顾客想删除已预订尚未入住的订单，系统会提示建议申请退款
	评价	顾客在入住或退房后可对酒店的服务进行评价，该评价会影响到酒店的综合评分
平台系统管理员	审核	在酒店管理员创建酒店后，对酒店信息进行审核确认，无误后加入系统
	创建酒店	酒店管理员在创建界面填写酒店相关信息即可向平台发出创建申请

酒店管理员	增删客房	酒店管理员可随时更改未被占用（包括已入住、已预订）的客房信息，增添或删除客房，更改酒店房型或价格
	办理退款	顾客发出退款申请后，酒店管理员需审核是否通过
	办理退房	为顾客办理退房业务，同时使该客房恢复可预订状态
	办理入住	为顾客办理入住业务，同时使该客房变为已入住状态

## 1.2 功能流程描述

流程图：



## 2 系统结构设计

### 2.1 概要设计

考虑到顾客、酒店、平台三方的需求，酒店预订及管理系统设计为三个模块。酒店管理员创建酒店并向平台提出申请，平台审核通过后，酒店管理员便可在酒店管理界面增、删、改、查酒店信息，并由平台整合显示在顾客界面，供顾客筛选查看。顾客提交订单后，可查看自己的订单状态，还可在未入住之前申请退款；酒店管理员也可随时观察订单状态，为顾客办理退款、入住、退房等业务，订单结束后，由顾客对酒店进行评价。这样就形成了一套完整的酒店预订及管理体系。

### 2.2 模块功能设计

Qt 安装目录下的 `sqldrivers` 文件夹含有数据库的驱动插件文件，QtSql 模块提供了对数据库的支持，SQLite 数据库又是 Qt 提供默认支持的一款轻型的文件型数据库，所以我选择采用 Qt + SQLite 的方式完成系统的开发。

为丰富代码的数据结构，避免数据库的大量使用，我选择用链表+Qlist 遍历容器实现大部分功能，少部分搜索功能由数据库来实现。并且建立 `data_object.h` 和 `data_object.cpp` 文件专门存储数据。系统可在应用程序启动时读取数据库的数据，并将数据存储到链表或 Qlist 容器内，进行数据初始化。在用户操作过程中，通过修改链表节点或 list 指针数组列表更改数据，并将更改的数据信息保存到数据库。需要说明的是，当用户在一次操作过程中再次进入到同一界面刷新时，数据便会得到初始化，这样可实现实时更新；但本系统还存在部分单向不可返回的界面，由于要等到线程结束后数据才会保存到数据库，因此用户要等到下次登录时链表或 Qlist 重新初始化，才可查看到修改信息。

因此，系统功能可主要归结为三方面：从数据库中读取数据并初始化；实现查找、修改信息的功能；将修改信息保存到数据库。

### 2.3 数据表设计

为了建立冗余较小、结构合理的数据库，建立数据库时要遵循一定的范式。在开发过程中，根据需求我建立了五类信息表，分别为全部酒店信息表、对应酒店客房信息表、订单信息表、平台审核信息表及注册用户名与密码信息表。注册信息表仅有两列，分别是用户名（主键）和密码。考虑到不同城市甚至是同一城市都可能有相同的酒店名，所以不能以酒店名为主键，因此我为每家酒店都配置一个 ID 号，在酒店信息表中以此作为主键。同时在表列中插入星级、评分等数据方便顾客进行酒店排序筛选。以下是酒店信息表的结构：

ID 号	酒店名	星级	城市	所在区域	评分
------	-----	----	----	------	----

为方便酒店与客房的对应,我将对应酒店的客房信息表表名设置为酒店的 ID 号,考虑到中文字符或数字命名数据库表格可能出现编码错误等种种问题,我将酒店的 ID 号统一设定为四位英文字母,如数据库样例中的“北京祥达酒店”,ID 号即为“aaae”,以房间号作为主键,同时向表列中加入房型、价格、房间状态,方便顾客选择,以下是酒店客房信息表:

样例: aaaa (北京祥达酒店)

房间号	房型	价格	房间状态
-----	----	----	------

在订单信息表中,我将顾客的 id 号(简化为六位数字)作为主键,并配有姓名、酒店名、房间号、入住时间、订单状态等信息,方便顾客和酒店管理员进行查看和修改,以下是订单信息表:

id	姓名	电话号码	酒店名	房间号	入住时间	订单
----	----	------	-----	-----	------	----

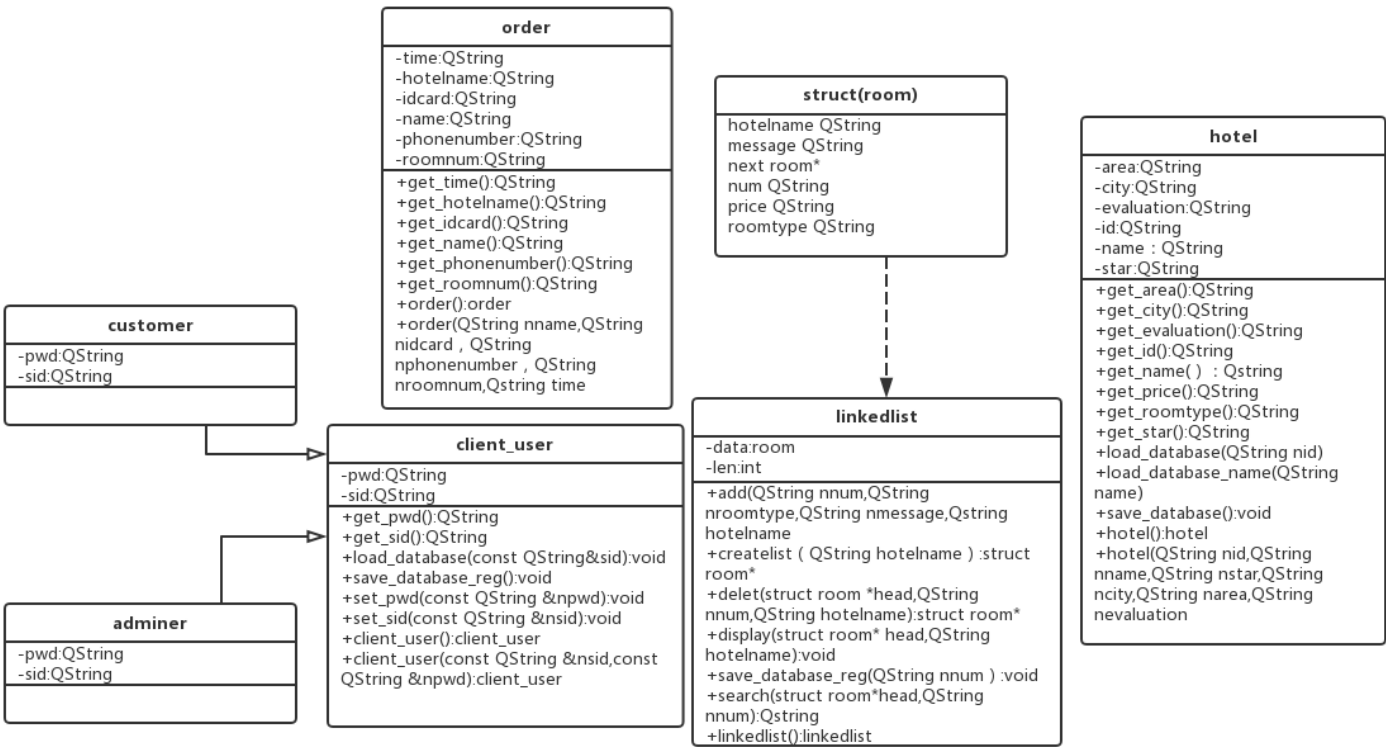
在平台审核信息表中,我以酒店的联系方式作为主键(电话号码不可重复)建立:

联系方式	酒店名	星级	城市	区域	审核状态
------	-----	----	----	----	------

## 2.4 类的设计

在这部分,我将采用 UML 类图的方式说明:

后端类:



建立客房结构体,并建立链表 linkedlist 类,用于存储酒店的全部客房信息,

类含有对客房结点的增、删、改、查函数。

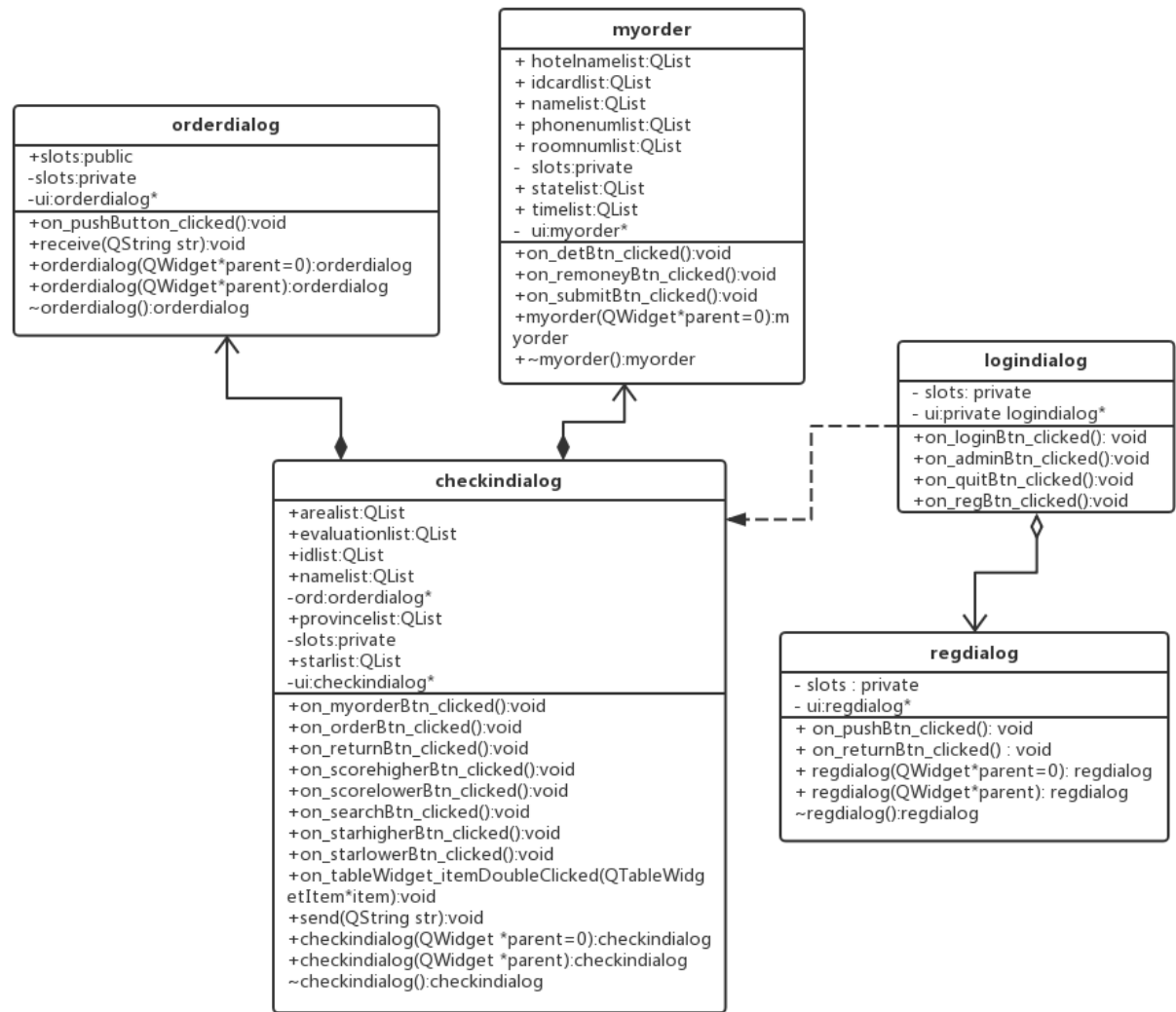
建立酒店类，含有城市、区域、评价、id 号、酒店名、星级等私有变量，并提供外部接口，通过 `get` 系列函数可以得到具体信息，`save_database` 函数可以保存数据，`load_database` 函数可以按照酒店名或 id 号的查找。

订单类含有日期时间、酒店名、顾客 id 号及名字、电话号码、房间号等私有数据成员，同样提供外部接口供获取。

顾客类和酒店管理员类继承于 `client_user` 类，并将 `client_user` 类的函数声明为虚函数。

以上是底层开发的类，均在 `data_object.h` 文件声明，在 `data_odject.cpp` 文件中具体实现。

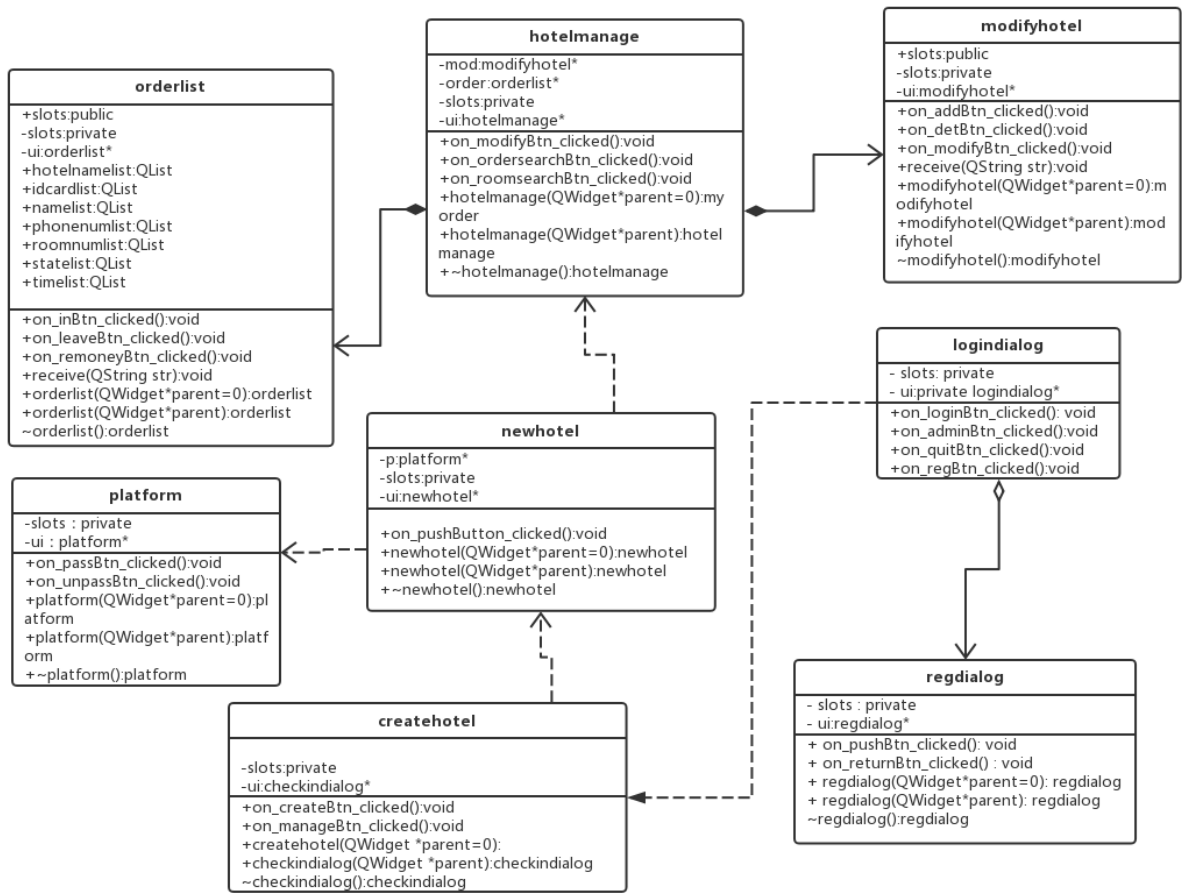
前端类：



注册界面类和登录界面类可双向访问，当用户初次使用系统时，可以通过登录界面进入注册系统，注册成功后可返回登录界面。

顾客主界面类含有我的订单界面和预订界面两个子窗口，可以实现双向访问，同时顾客主界面类将指向 `orderdiaog` 类的指针定义为自己的私有成员，在此处引

入信号槽机制，将顾客选择的酒店作为信号，从而在 `orderdialog` 中直接显示该酒店名，提高顾客预订效率。



相应地，酒店管理界面同修改酒店信息界面类及酒店订单界面类也存在组合关系，采用同顾客界面类同样的信号槽机制，在修改信息界面和酒店订单界面接收信号，显示酒店名。

### 3 系统详细设计

#### 3.1 类重点方法的实现

##### 3.1.1 搜索功能的实现

在顾客主界面，采用 `QtableWidget` 建立表格显示酒店信息，顾客可根据酒店星级及所在城市，地区进行搜索。关于搜索，我采用的是 `Qlist<QString>` 容器遍历的方法：

```
if(province!="&&star!="&&area=="")
{
    for(i=0;i!=idlist.size();i++)
    {
        if(provincelist.at(i)==province&&starlist.at(i)==star)
        {
            rowIndex++;
            ui->tableWidget->setRowCount(rowIndex);
            ui->tableWidget->setItem(j,k,new QTableWidgetItem(idlist.at(i)));
            ui->tableWidget->setItem(j,++k,new QTableWidgetItem(namelist.at(i)));
            ui->tableWidget->setItem(j,++k,new QTableWidgetItem(starlist.at(i)));
            ui->tableWidget->setItem(j,++k,new QTableWidgetItem(provincelist.at(i)));
            ui->tableWidget->setItem(j,++k,new QTableWidgetItem(arealist.at(i)));
            ui->tableWidget->setItem(j,++k,new QTableWidgetItem(evaluationlist.at(i)));
            j++;
            k=0;
        }
    }
}
```



关于 Qlist 模板类，是基于索引的访问，但是对于只读的访问，用 at() 比用 [] 操作符要更快，因此采用 at() 函数来查找。

### 3.1.2 排序功能的实现

在顾客主界面，顾客可对列表中的酒店根据星级或评分排序，从而实现更高效的检索。关于排序，我采用的是冒泡排序的算法：

```
for(j=0;j<=idlist.size()-2;j++)    //冒泡排序
{
    for(i=idlist.size()-1;i>=j+1;i--)
    {
        if(starlist.at(i)>starlist.at(i-1))
        {
            idlist.swap(i,i-1);
            namelist.swap(i,i-1);
            starlist.swap(i,i-1);
            provincelist.swap(i,i-1);
            arealist.swap(i,i-1);
            evaluationlist.swap(i,i-1);
        }
    }
}
```

其他位置的搜索、查找和排序功能与此相似。

### 3.1.3 链表功能的实现

为丰富程序的数据结构，在酒店查看订单和修改订单的过程，我没有采用 Qlist 模板类模拟链表，而是采用真正的链表数据结构，下面为链表提取数据库初始化和搜索、添加、删除节点的算法：

```
room *head;
room *p,*q;
head=p=q=NULL;
 QSqlQuery query;
query.exec(QString("select * from myHotelInFR where name='%1'" ).arg(hotelname));
query.next();
QString cmd=query.value(0).toString();
QString str=" select * from "+cmd;
query.exec(str);
while(query.next())
{
    p=new room;
    p->num=query.value(0).toString();
    p->roomtype=query.value(1).toString();
    p->price=query.value(2).toString();
    p->message=query.value(3).toString(); //初始化已经创建的数据
    if(head==NULL)
        head=p;
    else
        q->next=p;
    q=p;
}
if(head!=NULL) q->next=NULL;
return head;
```

```
room *p;
p=new room;
p=head;
while(p!=NULL)
{
    if(p->num==nnum)
    {
        return (p->message);
    }
    else
        p=p->next;
}
```

```

pnode=new room;
pnode->num = nnum;
pnode->roomtype=nroomtype;
pnode->price=nprice;
pnode->message=nmessage;
if (head ==NULL)
{
    head= pnode;
    return head;
}
if (nnum < head->num) //将nnum插入, 成为新的头节点
{
    pnode->next = head; //将pnode结点和原来的头结点联系起来
    head=pnode;        //pnode成为了新的头结点
    return head;
}
q=head;
p=head->next;
while (p != NULL)
{
    if (nnum < p->num)
        break;
    else
    {
        p=p->next;
        q=q->next;
    }
}
pnode->next = p;
q->next = pnode;
return head;

```

```

room *p,*q;
p= new room;
q= new room;
if(head==NULL) return head;
p=head;
while((p!=NULL)&&(p->num!=nnum))
{
    q=p;
    p=p->next;
}
if((p!=NULL)&(p->num==nnum))
{
    if(p==head)
    {
        head=p->next;
    }
    else
        q->next=p->next;
    free(p);
}
return head;

```

### 3.1.4 评分功能的实现

考虑到保障酒店的利益, 在酒店创建之初, 系统会自动为酒店分配 4.5 的评分, 在顾客评分后, 将采用 加权评分= ((现有评分\*3)+此次评分)/4 的算法, 既使评分具有一定的反映酒店时效状况的作用, 又保障了酒店的利益, 避免了评分的剧烈浮动。

```

float nscore=(score.toFloat()*3+evaluation.toFloat())/4;
QString neuevaluation=QString::number(nscore,'f',1);

```

### 3.1.5 数据库功能的实现

为避免应用程序中的数据丢失, 本系统所有数据都是基于数据库存储的, 在程序运行时, 读取数据库, 将数据库初始化保存到容器类或链表中, 在类模板或链表中进行数据的操作, 并同步修改数据库, 最终将数据保存在数据库中, 在程序中运用了 SQL 语法的知识。

## 3.2 界面结构及容错设计

### 3.2.1 登录界面

欢迎来到成成酒店预订系统!

用户名:  ☐ 顾客

密码:  ☐ 管理员

[点此注册](#)

登录界面采用中国风山水图作为背景，给用户以美感和享受性的体验，用户需输入用户名和密码并选择身份才可登录；新用户可通过右下角新用户注册通道进入注册界面；管理员可通过左下角管理员通道进入审核系统。实现多用户共用同一登录界面，体现简约性。

容错设计：（1）用户未选择身份或全部勾选时，警告信息；  
（2）用户密码或管理员密钥输入错误时，提示信息。

### 3.2.2 注册界面



The registration interface features a traditional Chinese landscape painting background. At the top, there is a navigation bar with a '注册页' (Registration Page) icon, a question mark, and a close button. The main heading is '欢迎注册' (Welcome to Register). Below this, there are three input fields: '用户名:' (Username) with a placeholder '请输入用户名' (Please enter username), '密码:' (Password) with a placeholder '请输入密码' (Please enter password), and '确认密码:' (Confirm Password) with a placeholder '请再次输入密码' (Please enter password again). There are two checkboxes: '顾客注册' (Customer Registration) and '管理员注册' (Administrator Registration). At the bottom, there are two buttons: '注册' (Register) and '返回' (Return).

注册界面用户需输入两次相同的密码完成注册，并且要选择相应身份，还可以通过返回键与登录界面实现双向访问。

容错设计：（1）用户注册用户名已存在时，提示信息；  
（2）用户两次密码输入不一致时，警告信息；

### 3.2.3 顾客主界面



The customer main interface features a traditional Chinese landscape painting background. At the top, there is a navigation bar with a '酒店信息' (Hotel Information) icon, a question mark, and a close button. The main heading is '酒店信息' (Hotel Information). Below this, there are search filters: '搜索酒店:' (Search Hotel), '目的地:' (Destination) with a placeholder '请填写城市名' (Please enter city name), '酒店类别:' (Hotel Category) with a dropdown menu, and a '搜索' (Search) button. There are also buttons for '酒店照片预览' (Hotel Photo Preview), '我要预订' (I want to book), and '返回酒店页' (Return to hotel page). A dropdown menu for '酒店地区\商业街' (Hotel Area/Commercial Street) is also present. On the left side, there are two sorting options: '按照级别排序:' (Sort by Rating) with buttons '由高到低' (From high to low) and '由低到高' (From low to high), and '按照评分排序:' (Sort by Score) with buttons '由高到低' (From high to low) and '由低到高' (From low to high). The main content area is a table with columns: ID, 酒店名 (Hotel Name), 星级 (Star Rating), 城市 (City), and 区 (District). The table contains 12 rows of hotel data.

ID	酒店名	星级	城市	区
aaab	北京亚朵酒店	5	北京	朝阳区
aaac	北京欢寓酒店	3	北京	海淀区
aaad	北京囊特酒店	4	北京	朝阳区
aaae	北京祥达酒店	3	北京	海淀区
aaaf	北京云逸酒店	4	北京	顺义区
aaag	北京时光漫步酒...	4	北京	顺义区
aaah	北京京伦酒店	5	北京	海淀区
aaaq	北京丽景湾酒店	4	北京	朝阳区
aaai	上海宝安酒店	5	上海	浦东新区
aaaj	上海零点酒店	4	上海	浦东新区

顾客主界面是功能点最多的界面。顾客可通过填写相关信息定向检索酒店，还可按照级别或评分对酒店进行排序，通过“我要预订”进入预订界面，通过“我的订单”进入顾客订单界面，实现功能一体化。TableWidget 含有工具提示（双击酒店 ID 获取酒店详细客房信息），在这里加入了信号一槽机制，获取顾客双击的单元格内容，从而在 TableWidget 上显示对应酒店的客房信息。

### 3.2.4 顾客预订界面

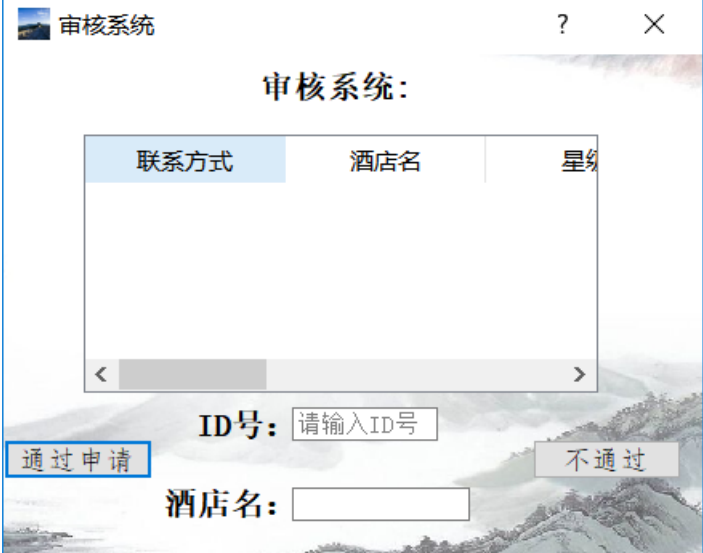
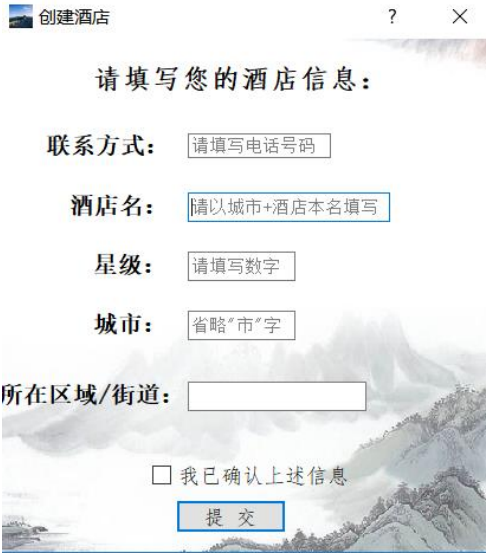


num	feature	price	message
101	单人间	300	已预订
102	单人间	300	已入住
103	双人间	500	已预订
201	单人间	300	未预订
202	多人间	600	未预订
301	双人间	500	已预订
302	多人间	600	已入住

顾客双击 ID 号并点击我要预订进入预订界面，填写相关信息，确认订单点击付款即可完成预订。

**容错设计：**（1）身份证号（简化为 6 位）及电话号位数出现错误时，警告信息；  
（2）确认订单未勾选时，提示信息（目的在于提示顾客检查信息）

### 3.2.5 酒店创建及审核界面

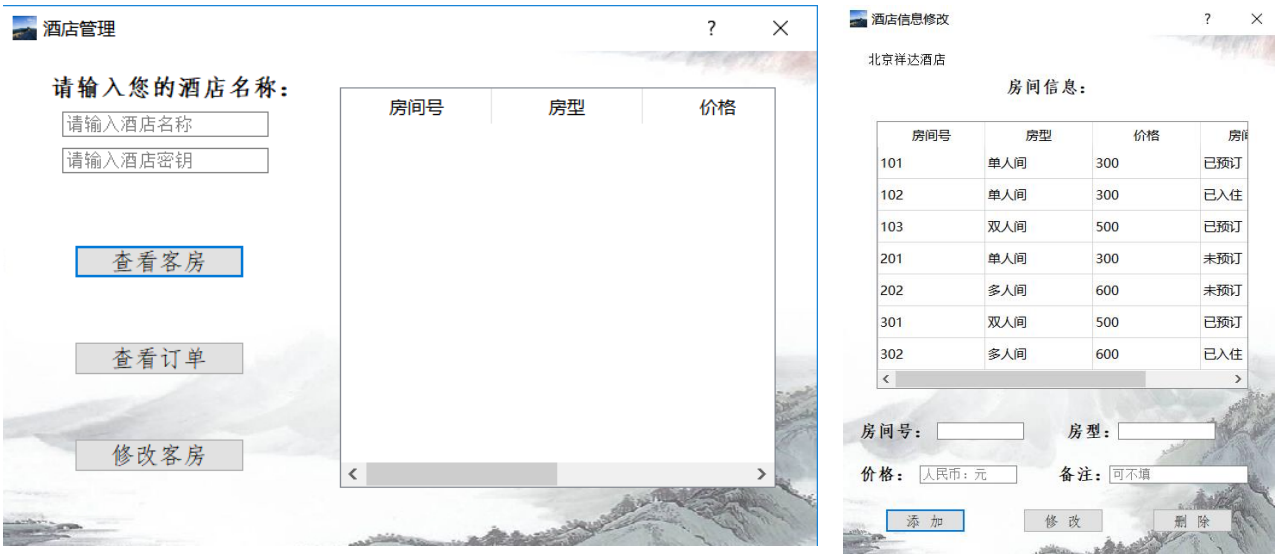


联系方式	酒店名	星级
------	-----	----

酒店管理员在填写相关信息后向平台提出申请，平台管理员可进入审核系统查看申请信息（为避免数据堆积，系统仅显示未核验的酒店信息），给予通过或不通过。

- 容错设计：**
- （1）当申请者填写信息有一项为空时，警告信息；
  - （2）当申请者未确认信息时，警告信息；
  - （3）当平台管理员审核未填写必要信息时，提示信息；

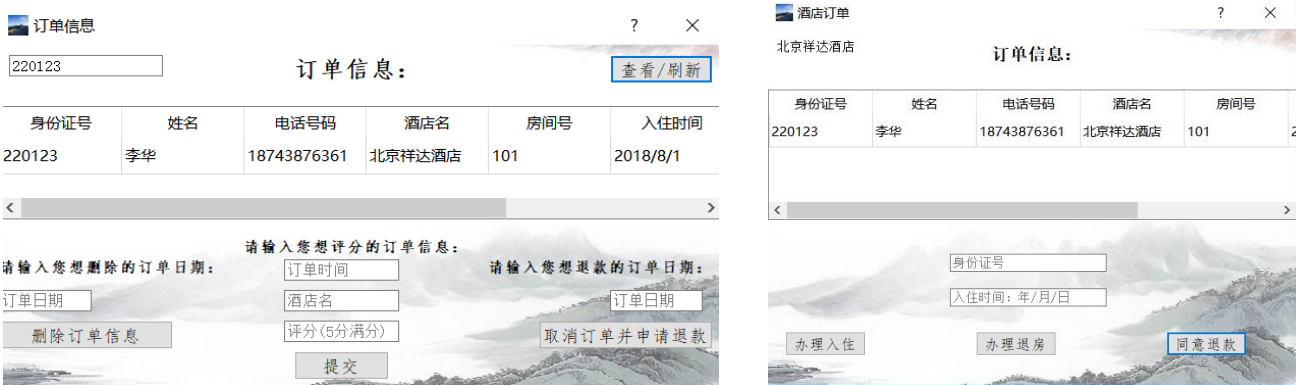
3.2.6 酒店管理及修改信息界面



酒店管理员在输入自己酒店名称和密钥（防止修改其他酒店的信息）后，可查看酒店客房信息及状态，并可在修改酒店界面增、删、改酒店信息。这里再次采用信号-槽机制，将用户输入的酒店名作为信号传递到信息修改界面，使修改界面的 TableWidget 显示同主界面同样的信息。

- 容错设计：**
- （1）当酒店名称或密钥栏为空时，警告信息；
  - （2）当欲删除房间为已预订或已入住状态时，意味着房间已被占用，管理员不可修改该客房有关信息；

3.2.7 顾客及酒店订单界面



顾客在输入自己的 ID 号后即可查看订单信息，还可在入住前申请退款；酒店可为顾客办理退款（需审核）、入住、退房业务，在入住后顾客还可对酒店的服务进行评价。

- 容错设计：**
- （1）顾客已入住后申请退款，警告信息；
  - （2）顾客在预定后选择删除订单，提示顾客是否入住，如未入住可申请退款，减少顾客不必要的损失
  - （3）当顾客未入住即评分时，系统提示信息；
  - （4）当顾客或酒店必要信息未填即进行下一步操作时，警告信息；

除顾客主界面和酒店主界面外，所有的界面均为模态对话框，继承于 `QDialog` 类，并且所有的功能均有独立的界面，软件的运行基于界面而不是数据，并且按钮点击与界面切换形成信号与槽，此项经过反复测试，防止出现应用程序意外崩溃的情况，有效地增强了系统的鲁棒性。

## 4 项目总结

在这次大作业项目中，我主要有以下几方面的收获：

（1）信号-槽机制的运用。信号和槽是 Qt 的核心机制，是一种高级接口，应用于对象之间的通信，减弱了 Qt 对象间的耦合度，并且保证了信号和槽参数的类型安全，使应用程序不致于轻易崩溃；与 MFC 的消息映射机制和 Java 中的监听机制相比是不同的风格，我们可以根据需要选用不同的方法；

（2）面向对象思想的运用。C++ 与 C 语言一个很大的区别就在于 C++ 是面向对象编程的语言，而 C 是面向过程的。在解决问题时，站在全局考虑，将问题的各个部分封装为不同的类，使用恰当的类，应用类的继承与派生特性来实现程序的功能，会使编程更有效率；

（3）调试方法的认识。虽然 `printf` 或 `cout` 输出调试的方法不算专业，并且略显俗套，但有时并不方便在程序中打断点，同时刚刚接触 Qt Creator 编译器的我，对其调试功能不算熟悉，所以本次开发过程中，我采用了 `qDebug()` 应用程序输出的方法完成了部分调试工作。

（4）Qt 和数据库知识的学习。作为一名转系生，刚刚从土木系来到自动化系的大家庭，就经历了 C++ 程序设计大作业的考验。大作业刚发布时，对 C 语言基础尚且薄弱的我来讲，根本找不到头绪和可开展的方向，当时一度陷入迷茫，后来，随着课程讲解的深入和不断地编程练习，我渐渐体会到编程的魅力和乐趣，编程能真正让一个人沉下心，忽略时间的流逝和周围的环境，完全沉浸于代码的海洋中。虽然 debug 的过程很艰辛，但是发现 bug 后的成就感也是满满的。通过一星期多的 Qt 和数据库知识的学习，也最终成功完成了这项大作业。软件功能还有很多可以改进的方面，代码的可复用性也尚有待提高，但是通过这三周，我增强了学习能力，为以后攻坚克难、继续学习设计性课程增强了信心！最后，我还要感谢老师、助教们在这门课上给我的帮助！

## 5 相关问题的说明

本项目采用 Qt 5.10.0 进行开发，连接数据库为 Qt 自带驱动的 SQLite 数据库，无需另行配置。需要的动态链接库为 Qt5Core.dll, Qt5Gui.dll, Qt5Sql.dll, Qt5Widgets.dll, qwindows.dll, qwindowsd.dll, 并将 iconengines, imageformats, platforms 及 sqldrivers 文件夹复制到 exe 文件同级目录下，否则会出现图片加载失败或数据库连接失败的问题。