

“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： 高子靖

学号： 2017010917

班级： 自 72

日期： 2018.08.03

目 录

1 系统功能设计.....	3
1.1 总体功能描述	3
1.1.1 顾客操作相应功能	3
1.1.2 酒店管理员操作相应功能	4
1.1.3 平台管理员操作相应功能	5
1.2 功能流程描述	5
2 系统结构设计.....	7
2.1 整体架构设计.....	7
2.2 模块架构设计	7
2.3 类总体设计	7
3 系统详细设计.....	11
3.1 类结构设计	11
3.1.1AbUser 类	11
3.1.2Customer 类.....	11
3.1.3Hotelmanager 类	11
3.1.4Holtel 类.....	12
3.1.5Order 类.....	12
3.1.6Message 类	12
3.1.7Room 类	12
3.1.8myDatabase 类	12
3.1.9Linkedlist 类和 Node 类	12
3.1.10customerlist 等六个链表类	13
3.2 界面结构设计	13
3.2.1 登录/注册界面结构设计	13
3.2.2 顾客主界面结构设计	14
3.2.3 酒店管理员界面设计	16
3.2.4 酒店管理员界面设计	17
3.3 关键设计思路	17
3.3.1 全局变量设计与数据存储.....	17
3.3.2 类的抽象与设计.....	18
3.3.3 用列表形式进行界面设计	18
3.3.4 界面跳转	18
3.3.4 容错设计	18
4 项目总结.....	19
4.1 问题与解决方法.....	19
4.2 项目亮点	20
4.3 开发心得	20
5 相关问题的说明.....	20

1 系统功能设计

酒店预订是旅游出行的重要环节,而酒店预订与管理系统中的管理与信息透明是酒店预订业务的关键问题所在,能够方便地查询酒店信息进行付款退款以及用户之间的交流对于酒店预订行业提高服务质量具有重要的意义。

针对网络预定酒店的实际情况,本项目根据实际需求设计并完成了“紫荆-子靖酒店预订管理系统”。该系统将使用单用户系统模拟多用户系统,使得不同用户登录时能够使用自己的功能并查看其他用户发来的消息。借助此系统,不同的用户之间可实时进行信息消息的交互,顾客,酒店管理员和平台上可进行信息的交互。酒店管理员可实时维护酒店的订单,酒店位置联系电话,房间等基本信息。平台管理员可在平台上看到所有酒店的基本信息,并可删除违规酒店,顾客可查看酒店并查询酒店与房间,下单退款,维护个人信息等。此系统旨在为酒店管理与预订带来便利。

总而言之,本项目是对酒店预定与管理系统除网络通信功能外的模拟,旨在使得酒店的预订与管理流程变得更加方便。

1.1 总体功能描述

本项目利用 OOP 的思想,使用 C++类与对象的知识进行顾客,酒店管理员,订单等程序核心对象与其相应核心功能的实现,借助数据库进行信息的长久储存并实现其相关功能。

除此之外,本项目使用图形用户界面应用程序开发框架——Qt 进行界面开发,在保证系统封装性的同时,用户可通过界面与程序进行交互。系统本质上实现的是不同类、不同对象的信息传递,具体功能如下:

1.1.1 顾客操作相应功能

Form

用户: 123

尊敬的顾客您好

欢迎使用紫荆-子靖酒店预订系统

杨柳荫中新酒店,

葡萄架下小渔船。

我的余额: 300

充值

我的消息

查看酒店

查询酒店

我的订单

退款订单

退出

您可在左查看所有的酒店和房间

所有酒店信息

酒店名称	酒店位置	联系电话	评分
1 友谊宾馆	北京	17717717711	5

所有房间信息

酒店名称	房间类型	房间价格	房间数量	优惠
1 友谊宾馆	大床房	300	4	0
2 友谊宾馆	三人间	500	2	0

退出

登录、注册: 使用账号和密码进行登录,用户同样可以使用身份证号登录,在进入顾客界面之前,必须进行注册。

我的消息：顾客登录后可查看由其他用户所发送的消息，并可进行选择回复，并可通过点击发送消息，通过输入对方的用户名来进行消息的回复。

查看酒店：在此界面顾客可以看到所有酒店信息以及所有房间信息。

查询酒店：在此界面用户可通过输入酒店名称查看该酒店所有房间信息，可通过输入酒店位置查看该位置所有酒店信息，并可通过点击排序按钮进行按照评分排序，可通过输入房间类型查看该类型的所有房间及其所属的酒店，并可通过按照价格排序按钮对房间进行排序。顾客只能看到审核通过的酒店。

发起订单/退款：顾客可通过填写订单来完成付款，并选择入住的天数与房间类型，填写后需进行付款，之后该订单可在我的订单界面查看，并在我的订单中可进行退款操作，将费用返还。我的退款订单可进行查看。

评价酒店：在付款之后，顾客即可对酒店进行打分式评价，评分需要用户输入，在 0-10 分之间的实数都满足要求。

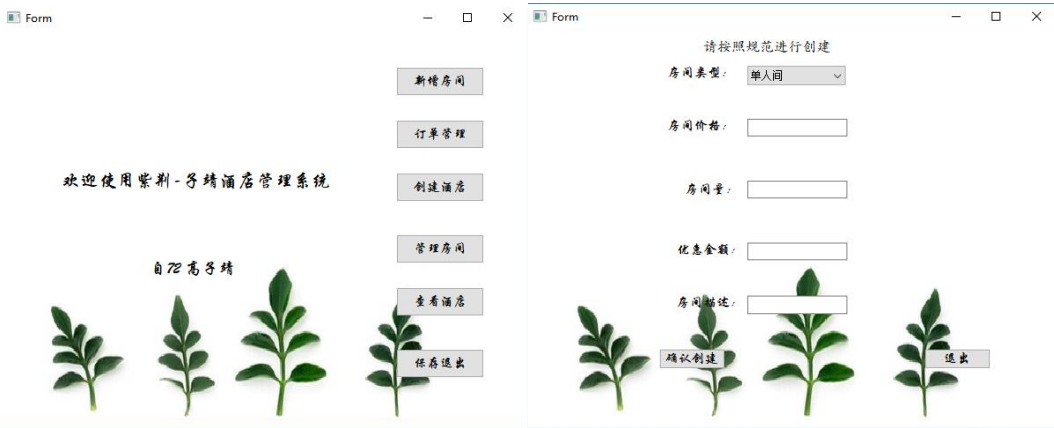
修改密码：在我的订单界面可进行密码的修改，需输入原密码和新密码。

账户充值：可通过输入密码进行账户余额充值。

退出登录：顾客可以暂时离开，其他用户可以登录，再次登录状态不变。

退出系统：程序运行结束，所有数据进行保存。

1.1.2 酒店管理员操作相应功能



登录、注册：同样使用用户名和密码进行登录，用户名具有唯一性不可重复注册，登录后进入管理员界面，如上左图所示。

创建酒店：可以通过填写酒店名称、位置、电话等创建酒店，酒店若无评分，初始化评分为 0，酒店名称具有唯一性，不可重复注册。

查看酒店：此界面显示该酒店的基本信息，酒店管理员可在此查看，以及是否通过审核。

修改酒店：酒店创建之后可进行的操作，仅限于修改酒店的位置与联系电话。

创建房间：酒店管理员可在要求范围内为酒店新增房间，每次可新建不只一个房间，可选择填写房间余量和房间优惠。

修改密码：在我的订单界面可进行密码的修改，需输入原密码和新密码。

查看/删除房间：酒店管理员可在此查看所属于本酒店的所有房间信息，并可进行删除已有房间来达到房间信息更新的效果。

查看已付款/已退款订单：酒店管理员可看到属于本酒店的所有订单信息，包括已付款订单和已退款订单。

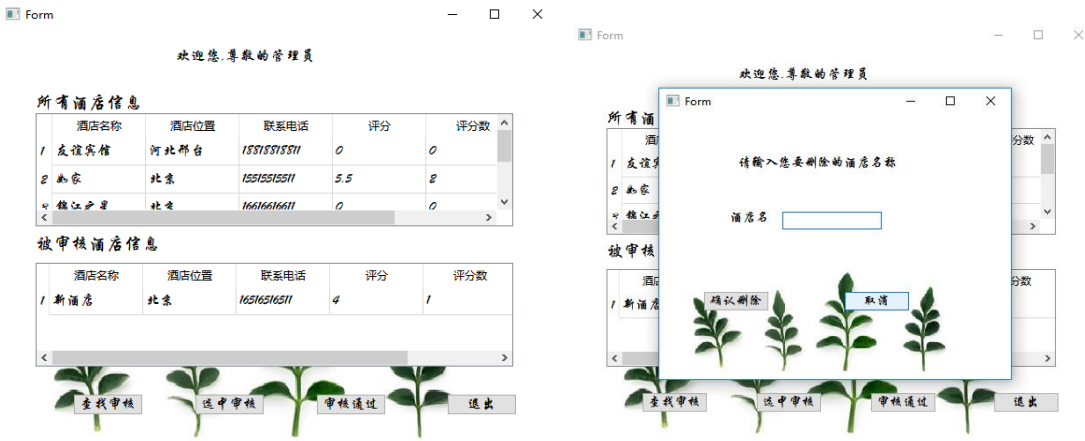
退款：酒店管理员可实时查看已付款订单，如因其他原因该房间不可被占用，可将该订

单退回并全额退款。

退出登录：酒店管理员可以暂时离开，其他酒店管理员可以登录，再次登录状态不变。

退出系统：程序运行结束，所有数据进行保存。

1.1.3 平台管理员操作相应功能



登录：平台管理员已初始化账户为“123456”，密码为“123456”，登陆后可进入到平台界面。

查看酒店：平台管理员登陆后可以查看到所有的酒店基本信息，也包括评分和评分数等信息，并可查看所有未通过审核酒店。

审核酒店：管理员可通过两种方式删除酒店，可从列表选中进行删除，或者可通过查找删除的方式进行删除，删除后酒店状态为未通过审核。

通过酒店：管理员可在已被审核的酒店中选中，重新通过该酒店的审核。

退出登录：平台管理员可以暂时离开，再次登录状态不变。

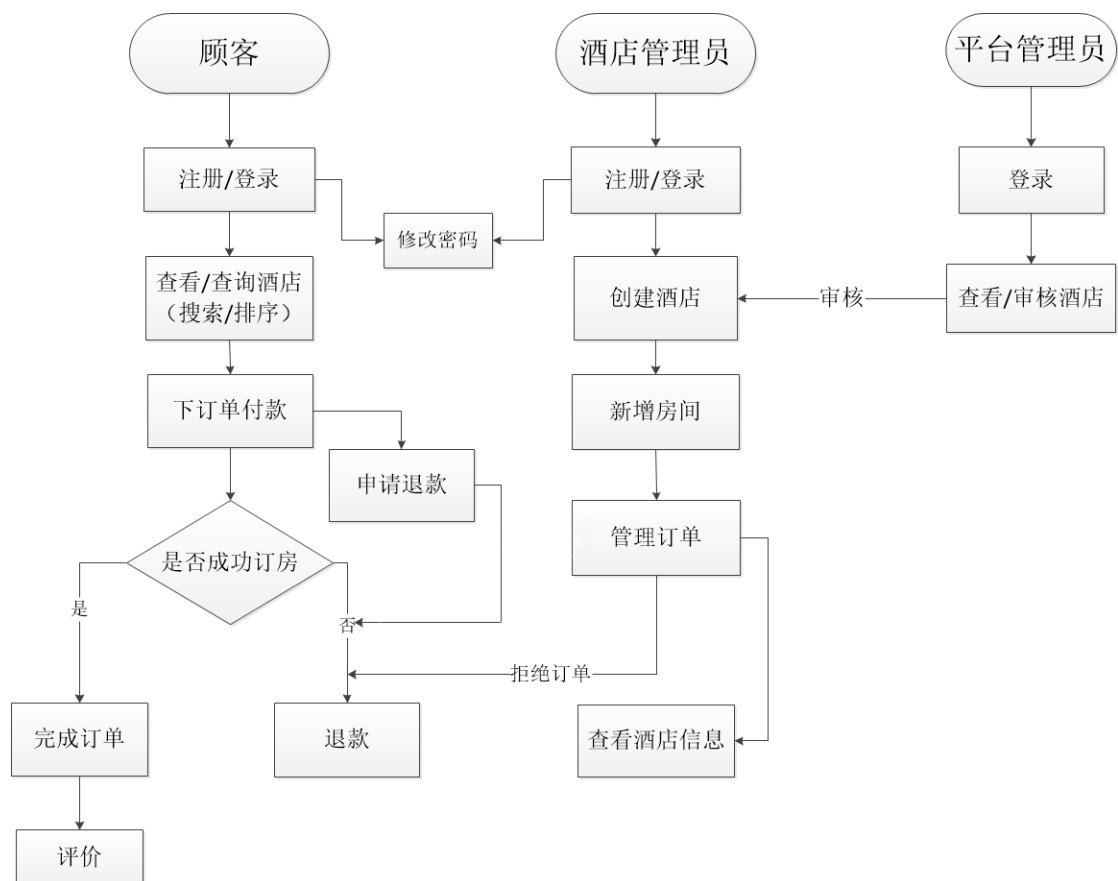
退出系统：程序运行结束，所有数据进行保存。

1.2 功能流程描述

本系统设计时遵循贴合实际的宗旨，在顾客预订过程中将顾客、酒店管理员、平台管理员的行为进行了有机的结合。

其中订单和酒店的状态均采用布尔类型。订单的状态有“已付款”和“未付款”两种，已付款订单可对酒店进行评价。未付款订单则为退款订单，会显示在顾客和酒店的退款订单列表中。酒店的状态为“通过审核”和“未通过审核”两种状态，顾客在查看酒店只能看到通过审核的酒店，而平台管理员可在平台上看到两种状态的酒店，并可实时对酒店进行审核，删除酒店。酒店管理员可在自己的酒店界面查看酒店是否为通过审核状态。

具体的功能流程如下图所示：



功能流程主要分为三方面：**顾客、酒店管理员和平台管理员。**

- 一. 顾客登录后到达主界面，可进行功能的选择，首先是收发消息与其他用户之间的消息交互，并可在登录后修改密码。之后可在查看酒店界面查看所有通过审核的酒店和房间信息。在查询酒店信息界面可完成下单付款，之后该订单即会在我的订单界面显示，顾客可选择退款或评价，若退款，则可在退款订单中查看到已退款的所有订单。
- 二. 酒店管理员登录后同样可修改密码，首先进行的操作是创建酒店，之后可修改酒店的信息，可向该酒店增加房间，并且可删除房间重新创建。酒店管理员可查看自己酒店的信息及是否通过审核。酒店管理员可查看该酒店订单信息，可进行订单的退款和查看退款订单。
- 三. 平台管理员登录后查看并审核酒店，可查看所有已通过审核酒店和未通过审核酒店，可将已通过审核酒店删除或未通过审核酒店通过审核，修改后的信息将会实时的反映在顾客和酒店管理员相应的界面中。

2 系统结构设计

2.1 整体架构设计

由于本项目要求不可频繁访问数据库，因此在没有数据库时，程序依旧能够运行，只是无法保存数据，因此本项目采用了如下的设计思路：程序启动时，数据将从数据库中一次性读出，此后程序将在内存中运行，待程序结束运行时所有数据将被再次存入数据库进行长久保存。

借助这种方法，未连接数据库时，程序依旧可以从网络或程序中获取信息并正常运行，程序将具有较强的可移植性。在内存中运行时选择用链表去保存数据，方便了数据的获取与新数据的插入。

项目的主体部分由自己编写的 C++ 代码和继承于自己编写的“LinkedList”类模板的链表构成。连接数据库时运用 SQL 语句，利用到各类知识点。

2.2 模块架构设计

由于本系统采用单用户模拟多用户，因此每个类至少有多对象，必然要用到合适的数据存储结构。

由功能流程描述可以看出，类与类、对象与对象间的联系十分密切，本项目决定采用全局变量的方法，将核心类的顾客、酒店管理员、房间、酒店、订单、消息采用全局变量的方法，存储在链表中。

本系统分为三个主要模块包含以顾客为核心的包括酒店预订，收发消息等的顾客管理模块，以酒店管理员为核心的酒店、房间、账号管理模块，以及以平台管理员酒店审核模块。

虽然系统设计由三个模块组成，但三个模块间有着密切的联系，如顾客的评价可以改变酒店的评分，酒店管理员可以退顾客订单并退款，平台管理员可以对酒店进行审核等，三个模块共同构成了系统这一整体。

2.3 类总体设计

不包括各种界面类在内，本项目一共设计了 8 个核心基本类：AbUser 类（用户基类）、Customer 类（顾客类）、Hotel 类（酒店类）、Hotelmanager 类（酒店管理员类）、Room 类（房间类）、Order 类（订单类）、Message 类（消息类）、myDatabase 类（数据库类）以及继承于链表类上述除 AbUser 类之外的六个链表类，及 Date 类等非核心类，以及 LinkedList 和 Node 类等用于数据存储功能的基类。UML 类图描述这些类如下：

核心类：

1、AbUser 类：

AbUser
-ID: QString//账号

-Password: QString//密码
+AbUser(QString ID1, QString Password1) //构造函数
+~AbUser () //析构函数
+getID(): QString//获取用户名
+getPassword(): QString//获取密码
+setID(QString ID): void//设置用户名
+setPassword(QString Password): void//设置密码
+changePassword(QString Password): void//修改密码

Customer 类与 Hotelmanager 类均继承自 AbUser 类，实现管理账号编辑密码功能，运用了类型兼容规则的知识，实现精简代码量。一下 AbUser 的子类继承基类的成员不再在 UML 图中进行叙述。

2.Customer 类:

Customer
-IDNumber:QString//身份证号，可用于登录
-treasurer:float//用户余额
+ Customer(QString ID, QString Password, QString IDNumber,float treasure)//构造函数
+~Customer() {}//析构函数
+setIDNumber(QString id):void//设置身份证号
+settreasure(float trea):void//设置余额
+getIDNumber():QString//获取身份证号
+gettreasure():float//获取余额
+paymoney(): bool//判断是否可支付，如可支付，余额减去相应费用，返回 bool
+recharge(): void//充值

本项目将顾客类进行抽象，体现了 OOP 思想，没有采用直接将顾客类设为结构体或者将其数据成员全部设为公有成员的方法，而是为外部提供许接口以实现功能，体现了封装性，酒店管理员类的实现与其类似。

3.Hotel 类

Holtel
-Hotelname: QString//酒店名称
-Hotelplace: QString//酒店地址
-Hoteltel:QString//电话
-ispermitted:bool//是否被审核通过
-score:float//酒店评分
-Numscore:int//酒店拥有评分的数量
+getHotelname():QString&//得到酒店名
+getHotelplace():QString&//得到酒店位置
+getHoteltel():QString&//得到酒店电话
+getscore():float//得到酒店评分
+getnumofscore():int//得到酒店评分数量
+getpermitted():bool//得到是否通过审核
+setscore(float score):void//设置酒店评分
+setHotelname(QString name):void//设置酒店名称


```

+setHotelplace(QString place):void//设置酒店位置
+setHoteltel(QString tel):void//设置酒店联系电话
+setpermitted(bool lpermitted):void//设置是否通过审核
+setnumofscore(int num):void//设置评分数量

```

4.Hotelmanager 类

Hotelmanager
-myHotel:Holtel//房间所属酒店
-HotelName:QString//房间所属酒店名
+setHotelName(QString name):void//设置酒店名
+setRoom(Room &a,QString RoomType,bool avail,float price ,float Discou):void//设置房间
+getHotelName():QString//获取酒店名
+Hotelmanager (QString ID, QString Password) ://构造函数
+~Hotelmanager()://析构函数

5.Room 类

Room
-RoomHol:QString//房间所属酒店
-RoomDiscription:QString//房间描述
-RoomType:QString//房间类型
-Roomprice:float//房间单价
-Discounts: float//优惠
-Roomnum:int//房间量
+setRoom(QString RoomType,float price, float Discou = 0):void//创建房间
+setRoomType(QString type):void//设置房间类型
+setRoomHol(QString Hol):void//设置房间所属酒店名
+setRoomDiscription(QString dis):void//设置房间描述
+setRoomnum(int num):void//设置房间量
+setRoomprice(float price):void//设置金额
+setRoomdiscount(float dis):void//设置优惠
+getRoomHol():QString//获取房间所属酒店名称
+getRoomType():QString//获取房间类型
+getRoomDiscription():QString//获取描述
+getRoomprice():float//获取房间价格
+getDiscounts():float//获取优惠
+getRoomnum():int//获取房间数

6.Order 类

Order
-Hotelname: QString//订单酒店名
-RoomType:QString//订单房间类型
-money:float//订单费用
-Startdate:Date//入住日期

-Customerid:QString//顾客用户名 -Days:int//入住天数 -Ispaid:bool//是否支付 -OrderID:int//订单号区分订单
+setorderid(int id):void//初始化订单号 +setcustomerid(QString id):void //初始化顾客用户名 +setispaid(bool ispaid1):void//初始化是否支付 +setstartdate(Date &a):void//初始化入住日期 +sethotelname(QString name):void//初始化酒店名 +setroomtype(QString type):void//初始化房间类型 +setmoney(float money1):void//初始化费用 +setdaylong(int day):void//初始化入住天数 +getmoney():float//返回费用 +getispaid():bool//返回是否支付 +getstartdate():Date//返回入住日期 +getHotelname():QString//返回酒店名 +getRoomType():QString//返回房间类型 +getcustomerid():QString//返回顾客用户名 +getdaylong():int//返回入住天数 +getorderid():int//获取订单号

7.Meesage 类

Message
-sendID:QString//发送者账号 -receiveID:QString//接受者账号 -message:QString//消息
+setsenID(QString id):void//设置发送者账号 +setreceivID(QString id):void//设置接受者账号 +setmessage(QString message1):void//设置消息 +getsendID():QString//获取发送者账号 +getreceivID():QString//获取接受者账号 +getmessage():QString//获取消息

8.myDatabase

myDatabase
+createConnection(): void//创建与数据库连接并创建表 +initialCustomer(customerlist &L1):void//初始化顾客链表 +saveCustomer(customerlist &L1):void//保存顾客链表数据至数据库 +initialmanager(hotelmanagerlist &L2):void//初始化酒店管理员 +savemanager(hotelmanagerlist &L2):void//保存酒店管理员链表至数据库

```
+initialhotel(hotellist &L3):void//初始化酒店链表
+savehotel(hotellist &L3):void//保存酒店数据至数据库
+initialroom(roomlist &L4):void //初始化房间链表
+saveroom(roomlist &L4):void//保存房间链表至数据库
+initialorder(orderlist &L5):void//初始化订单链表
+saveorder(orderlist &L5):void//保存订单数据至数据库
+initialmessage(messagelist &L6):void//初始化消息链表
+savemessage(messagelist &L6):void//保存消息至数据库
```

除去以上类之外，**LinkedList** 类和 **Node** 类在数据的存储中起到了重要的作用两个类，他们为核心类的链表对于数据的操作提供了基础。

本项目将数据作为单独的一个类，本项目仅在打开登录界面时生成数据库类的一个对象，进行数据初始化，在退出系统时，再依次调用 **save** 函数进行信息的储存，其余情况数据库一直处于关闭状态，而数据库类中的函数全部设为静态函数，方便在调用时切换作用于调用初始化和保存函数。

3 系统详细设计

3.1 类结构设计

3.1.1 AbUser 类

该类为可登录的用户类的基类，具有账号和密码数据，并具有为账号密码初始化的外部接口。设计该类为了方便顾客和酒店管理员继承，减少代码量，用户注册和登录界面需要的 **Customer** 类和 **Hotelmanager** 类均需以此作为基础。

3.1.2 Customer 类

该类为顾客类，为主要核心类之一，继承于 **AbUser** 类，顾客的用户名在顾客的功能之中具有至关重要的作用，而此类作为基本类主要为外部提供接口可调取或初始化类中部分数据成员，体现封装性。类本身不具有具体功能，但为用户基本功能实现提供接口。

3.1.3 Hotelmanager 类

该类为酒店管理员类，同样继承于 **AbUser** 类，设计该类为酒店管理员数据的存储以及外部功能的实现提供接口，两类同样继承于 **AbUser** 类，体现了精简代码的作用。

3.1.4Hotel 类

该类为酒店类，在以顾客为核心的酒店预订模块，和以酒店管理员为核心的酒店管理模块中起到至关重要的作用，该类主要涉及有酒店名，酒店评分等与顾客、订单等进行交互的私有数据，该类设计为订单所属酒店，房间所属酒店的基础，为本项目的基础支撑类之一。

3.1.5Order 类

该类为订单类，同样是顾客与酒店的交互过程中的核心类，订单类中具有所属酒店为酒店查看订单，以及订单付款后对酒店进行评价提供基础，具有订单所属顾客的用户名，为用户查看自己订单提供基础，同样订单具有订单号，在对订单进行付款、查找等提供基础，简而言之，订单类是酒店管理与预定中必不可少的类。

3.1.6Message 类

该类为消息类，主要为用户之间互相发送消息提供基础。该类结构简单，只具有发送者用户名，接受者用户名，和消息三种私有数据，在 `MessageList` 类中，通过查找函数方便对消息信息的分类与整合，为本项目在单用户模拟多用户登录时，提供消息列表的基础。

3.1.7Room 类

该类为房间类，主要功能为酒店创建管理房间信息提供接口，同样在顾客界面，可使顾客查看到房间信息，并进行下单处理。为顾客和酒店管理员功能的实现间接提供基础。

3.1.8myDatabase 类

前文已进行过叙述，本项目将数据库作为单独的一个类存在，为项目中全局变量的数据从数据库中读取和向数据库中存储提供接口。其中函数的设计清晰简单，由于不可频繁访问数据库，因此只设计 `initial` 和 `save`，即初始化链表和将链表数据存入数据库，与 `createconnection` 函数，建立与数据库的连接，并且为六对链表的存储与初始化创建表。

数据表设计添加在附页。

3.1.9Linkedlist 类和 Node 类

此两类为链表类模板和结点类，为链表基本操作的实现提供基础，如表尾插入数据，获取链表结点个数等，为订单、房间、酒店中的查找功能，在遍历时提供方便，同样链表类作为六个全局变量所属类继承的基类，在本项目运行时的数据处理为至关重要的作用。

3.1.10customerlist 等六个链表类

六个链表类各创建一个对象作为本项目的全局变量，例如 customerlist 的 L1 即为顾客类链表，hotelmanagerlist 的 L2 即为酒店管理员链表类，之所以将这些类作为继承 linkedlist 类出现，是由于在个别链表类如 roomlist、messagelist 等类中加入了查找函数与排序函数，为统一起见，本项目统一将六个全局变量所属类继承链表类存在。

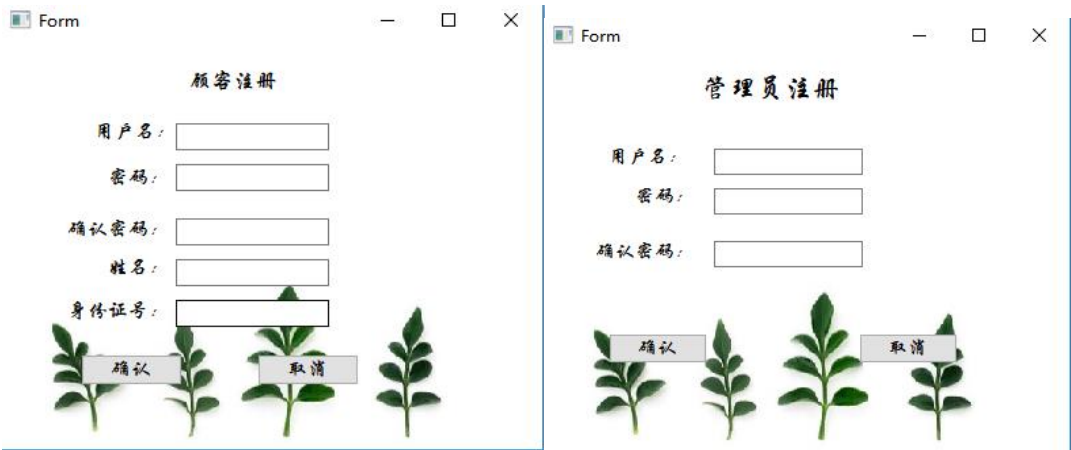
3.2 界面结构设计

3.2.1 登录/注册界面结构设计



登录界面设计为两个注册窗口，和三方登录的选择。首先界面的跳转均在 `button` 的槽函数中进行新界面的 `show`，顾客注册与酒店注册为不同的 Qt 设计师界面，在槽函数中分别新建这些窗口的对象并使其跳出。

其中注册界面设计如下所示：



在用户登录时采用 radiobutton 去选择登录方，在登录按钮的槽函数中使用 switch 去分别执行相应的操作。

用户可使用注册时使用的用户名或身份证号登录，管理员在注册时只进行账号密码的注册，登录窗口如上所示。注册时重复注册，身份证号不为 18 位，登录时用户密码不正确时均会有 QMessageBox 消息框进行提示。

3.2.2 顾客主界面结构设计



如上图所示为顾客登录后的主界面，登录之后右侧为功能按钮，左上角文本会显示登录的用户名，下方直接显示顾客的余额，点击充值会跳入充值界面，需输入密码并输入需要充值的金额。

下面一一介绍右侧 button 的功能，由于界面过多，不再全部显示图片：

我的消息：此界面为消息界面具有点击发送功能，同时可查看我所发送的消息与我收到的消息，在我收到的消息中可选中消息，并在下方回复框中输入回复该顾客的消息，点击回复按钮会发送至对方顾客。

查看酒店：此界面为查看所有酒店按钮，界面中具有两个列表，分别是所有酒店列表和所有房间信息，包括房间的余量和折扣等细节。

查询酒店：如下图所示：

酒店及房间信息：

请您输入必要的信息查找酒店

☒ 酒店位置 ☐ 客房类型☐ 酒店名称

北京

搜索

价格由高到低排序

评分由高到低排序

发起订单

退出

	酒店名称	酒店位置	联系电话
1	紫荆酒店	北京	17676116711
2	如家	北京	15515515511
3	新酒店	北京	16516516511
4	锦江之星	北京	16616616611

可按照酒店位置查询酒店信息，按照客房类型查找房间信息，可按照酒店名称查看该酒店所有房间信息，酒店可进行评分的排序，客房查询可进行价格的排序，在下方发起订单同样跳转到下一 Qt 设计师界面，填写订单，并下单完成金额的扣除。

我的订单：在我的订单中所看到的均为已支付订单，可退款和评价。

退款订单：查看自己退款的所有订单。

退出：返回到登录界面

总体而言，采用不同功能不同界面，在主界面进行功能选择的界面，利用信号槽机制进行界面间的跳转，使用户操作方便，对于功能的选择性增强。

3.2.3 酒店管理员界面设计



和顾客的登录界面相类似，登录后的界面为功能选择的界面，创建酒店新增房间等，每一个功能界面都是新建的一个 Qt 设计师界面，在每个界面的“取消”或者“退出”按钮加上本窗口的 `close()` 函数，因此窗口间的切换是以新窗口的出现与关闭来实现。

查看界面中的 `lineEdit` 均设置只读功能，并从链表中读出并显示其数据。

界面背景设置方法如下：

```
//设置背景图片
QPalette pal;
QPixmap pix = QPixmap(":/new/prefix1/mypicture/_20180801113455.jpg").scaled(this->size());
pal.setBrush(QPalette::Background,QBrush(pix));
this->setPalette(pal);
```

在已添加至资源文件中读取相应图片的路径。

其中数据的列表显示用 `QTableWidget` 进行显示，通过遍历相应的链表初始化行数，再将链表数据域中数据依次放入表中显示。

消息提示框采用 `QMessageBox` 进行容错的消息提示，提醒用户的操作。

3.2.4 酒店管理员界面设计

Form

欢迎您,尊敬的管理员

所有酒店信息

	酒店名称	酒店位置	联系电话	评分	评分数	
1	友谊宾馆	河北邢台	18818818811	0	0	
2	如家	北京	15515515511	5.5	2	
3	锦江之星	北京	16616616611	0	0	

被审核酒店信息

	酒店名称	酒店位置	联系电话	评分	评分数	
1	新酒店	北京	16516516511	4	1	

查找审核

选中审核

审核通过

退出

平台管理员在登录之后可直接进行功能操作，界面由 `tablewidget` 显示出所有审核通过的酒店信息以及审核未通过的酒店信息。

查找审核会跳转到下一个 Qt 设计师界面，可通过查找该酒店名来使酒店处于审核不通过的状态，所有酒店的删除都不会从链表中删除该酒店，而是将该酒店的状态变为未通过审核，顾客只可以看到通过审核的酒店。

选中审核可在上方表格中选中酒店将其变为审核未通过状态。

审核通过是在下方表格中选中酒店，将其恢复为审核通过状态。

退出按钮连接槽函数，使当前界面关闭。

3.3 关键设计思路

3.3.1 全局变量设计与数据存储

本项目仅用数据库实现数据的持久化，主体部分以全局变量为核心。程序运行时，所有的顾客信息，酒店管理员信息，酒店信息，房间信息，订单信息和消息信息从数据库读入内存，放入链表中，程序运行结束时再将所有更新的数据存入数据库，实现了程序脱离数据库亦能运行的要求。

其中六个全局变量的声明如下：

```
//全局变量的声明
customerlist L1;
hotelmanagerlist L2;
hotellist L3;
roomlist L4;
orderlist L5;
messagelist L6;
```

3.3.2 类的抽象与设计

本项目用 C++ 编写，将顾客、酒店管理员、酒店、订单、房间等均抽象为类。不同于直接使用数据库编程，将列表内的 QSqlTableModel 的格与数据库中表中的格相对应，进行删、改、查的做法，本项目严格面向对象编程，所有对象均具有数据抽象与行为抽象。本项目在对酒店预订管理行为进行合理的抽象，并且具有一定的封装性，并通过类的继承达到精简代码的目的。

3.3.3 用列表形式进行界面设计

本项目中在数据的显示与查看和功能操作中，最主要形式是通过 tablewidget 实现，包括对于酒店房间的查看，对于订单的查看，平台管理员的审核，都建立在列表的操作基础上。全局变量链表中存储的均为类的实体，生成列表时从相应链表中对所需要的信息进行检索，符合要求的结果依次放入列表读出。如从列表读取数据，则通过遍历列表的各行，符合要求的结果加入到链表的尾部，可以较为方便地实现增删功能。

3.3.4 界面跳转

本项目中的不同界面均为新建的 Qt 设计师界面，通过界面对象的 show 和 close 函数来掌握界面的显示与关闭，支撑整个程序的界面显示功能。具体跳出界面的函数如下所示：

```
void Loginwindow::on_holregbutton_clicked()
{
    //弹出酒店管理员注册窗口
    managerregwindow *man1 = new managerregwindow;
    man1->show();
}
```

3.3.4 容错设计

本项目进行了必要的容错设计，对用户的不规范操作与误操作有一定的容忍度，不会轻易崩溃，具体如下：

1. 顾客注册，若身份证号不为 18 位，则会弹出 QMessageBox 提示。
2. 顾客注册，若两次输入密码不一致，则会弹出 QMessageBox 提示信息。
3. 顾客注册，若注册已经注册过的用户名或身份证号，同样会弹出消息框提示“该用户已经

注册”。

4.酒店管理员注册时，两次密码不一致则会提示“两次密码不同”，若已注册过会提示“该用户已注册”。

5.顾客与酒店管理员登录时若输入账号密码不正确，则会弹出消息提示框。

6.顾客下单与酒店管理员在创建房间时房间类型从 ComBox 中选择，避免输入时的错误，增强容错性。

7.顾客下单时，会进行金额的计算，若金额不足，则会弹出消息提示框提示“您的余额不足”。

8.顾客下单时，若输入的酒店名不正确，则会弹出消息提示框“您输入的信息有误”。

9.顾客不可在下单时付款余量为零的房间，如果如上操作，则会弹出消息提示框“房间余量不足”。

10.对酒店进行评价时，输入评分必须在 0-10 之间的实数，若超出范围，则会弹出消息提示框“您输入的分数不在范围内”。

11.审核酒店时，若输入不存在的酒店名，则弹出消息提示框“您查找的酒店不存在”。

12.删除房间时，若输入不存在的房间类型，则弹出消息提示框“您查找的房间不存在”。

13.在查询酒店的界面，若选择酒店位置查找，则按照价格排序按钮设置为不可点击；若选择房间类型查找，则按照评分排序按钮不可点击；按照酒店名查找，则两排序按钮不可点击，防止用户的错误操作。

14.对于从列表中选中数据的界面，均设置为不可编辑表中内容，防止操作失误导致表中内容更改。

4 项目总结

4.1 问题与解决方法

项目的开发过程中，作为设计者，我也遇到了很多问题，接受了很多挑战，项目最终的成功也能给人带来喜悦，并伴随着许多思考。

1.数据库

开发项目时，为了数据的长久保存，选择进行数据库的连接，但在创建表与从表中读取或存储数据时，需要用到部分 SQL 语句，在引号引起来的 SQL 语句中，编译器并不会进行报错，如果语句输入存在问题则难以找出错误。在首次写初始化函数和保存函数时，我先为 customer 和 hotelmanager 创建表，由于 sql 语句拼写有单词存在错误，多次尝试都为成功将数据存储，最后在创建表的时候使用 BOOL 类型和 qDebug 去进行判断，以及进行逐句检查才发现问题，为之后其他链表与数据库间的数据交换提供了方便，之后再写 initial 和 save 都是一次成功。

2.酒店删除

在本项目最初的开发过程中，平台管理员是具有删除酒店的功能，当时在删除时最初的想法是在链表中直接将该结点删除，但删除过程中由于直接进行了 delete 操作，导致中间结点出现了野指针的情况，被删除结点前后两结点并没有连接上，因此一度导致程序崩溃，之后采用整数 i 记录位置，再初始化游标的方法删除结点，之后酒店的删除成功，但由于酒店直接删除缺少交互性，因此在酒店中加入是否通过审核的布尔变量，记录酒店状态，因此平台管理员的操作从删除结点变为改变状态量，避免了出现野指针的情况。

3.文件路径

由于一些部件例如按钮等在加入槽函数后删除，程序 debug 时会进行报错，找不到该部件的定义，需要在 moc 文件中进行修改删除原 button 等部件的声明。并且在一次移动文件路径后，添加新文件不可在加入到该项目中，因此我进行了重复添加进行了大量报错，最后同样在报错的文件中进行原重复加入文件的删除才使程序得以运行。

4.2 项目亮点

1. 充分利用所学知识点

本项目体现了 OOP 的抽象、封装、继承、多态等设计思想和关键技术，而且充分运用课程中所学 C++ 的面向对象的知识：所有用户类是对 AbUser 类的继承，并将小作业中的 Linklist 类与 Node 类运用于全局变量链表的基类。包括之后利用的 Qt 的信号槽机制，以及部件文本的读取与写入等，边学边写并将所学新知识充分利用的项目中去。

2. 注意细节

在细节方面，使用强制类型转换，包括余额费用，订单数等具有不同的数据类型，向文本中输入时运用 QString::number 等，以及，评分时 int 型和 float 型之间进行的类型转换。在数据库存储 bool 类型时，为防止出现未知错误，特意将 bool 类型转化成 integer 型储存，在读入读出分别进行判断，来存储 bool 类型，以此防止错误的出现

3. 界面美观

项目制作过程中，修改合适的窗口大小，并且通过添加背景图片、修改字体等来对窗口进行美化，更符合实际要求，做到界面美观舒适。

4.3 开发心得

开发项目时，在首次看到题目真的感到无从下手，大一程设基础非常薄弱，并且在两天内老师就复习完了 c 语言，对于我个人而言，每天新学习的知识量确实是蛮大的，不过在每天对新知识的学习中，与 Qt 知识的自学与网上查询，逐渐对大作业有了头绪，也像范老师课上所说，在完成项目的过程中拥有不少的 Aha time。

当然，在开发中不断的 debug 确实是一个个的挑战，会出现各种各样的问题，但每一种问题在解决之后的确会感到自己的提高。在项目逐渐完成的过程中，越来越能运用到 c++ 抽象、继承与封装的特性。

在过程中遇到很多不知无法解决的问题，也有同学给我的 bug 提供了解决的方法，以及有些功能的实现为我提供了很好的思路，再加上在网络上查询的各种功能的运用，以及从书中所看到的知识，才最终能够使我完成此项目的开发，非常感激自己得到的各种帮助。

总而言之，在大作业完成的过程中感到了自己的很大提高，尤其是在思考下一步功能的实现时所花费的大量时间，在现在看来都是十分有意义的，我也明白自己的知识水平还具有很多的不足，真正提高自己的编程水平，还需在之后的学习生活中继续努力。

5 相关问题的说明

1. 开发测试环境

本项目使用 Qt creator 开发，由 Desktop Qt 5.11.1 MinGW 32 bit 构建运行

测试环境为 Window10

2. 使用说明

1. 平台管理员账号 123456 密码 123456
2. 默认输入身份证号位数为 18 位，否则会进行报错
3. 必须在退出系统时，点击登录界面的“退出”，相应的数据才会被存入数据库，直接关闭窗口则不能讲信息存入数据库中。

附：数据表设计

```
//为Customer类创建表
query.exec("create table if not exists mmyyCustomers(ID varchar(20),"
           "Password varchar(20),IDNumber varchar(20),treasure float);");

//为Hotelmanager类创建表
query.exec("create table if not exists mmyyHotelmanagers(ID varchar(20),"
           "Password varchar(20),HotelName varchar(20));");

//为Holtel类创建表
query.exec("create table if not exists mmyyHoltels(Hotelname varchar(20),"
           "Hotelplace varchar(20),Hoteltel varchar(20),Ispermitted integer,"
           "score float,numofscore integer);");

//为Room类创建表
query.exec("create table if not exists mmyyRooms(RoomType varchar(20),"
           "RoomHol varchar(20),Roomprice float,Discounts float,RoomDiscription varchar(60),Roomnum integer);");

//为Order类创建表
query.exec("create table if not exists mmyyOrderssss(Hotelname varchar(20),Roomtype varchar(20),"
           "customerid varchar(20),money float,Startyear integer,Startmonth integer,Startday integer,"
           "dayslong integer,ispaid integer,orderid integer);");

//为Message类创建表
query.exec("create table if not exists mmyyMessage(sendID varchar(20),"
           "receiveID varchar(20),message varchar(100));");
```