

# “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：           张扬          

学号：           2017011\*\*\*          

班级：           自 72          

日期：           2018 年 8 月 3 日

# 目 录

1 系统功能设计.....	4
1.1 总体功能描述 .....	4
1.2 功能流程描述 .....	5
1.2.1 登录/注册流程 .....	5
1.2.2 用户信息修改功能 .....	5
1.2.3 平台管理员功能 .....	5
1.2.4 顾客功能流程 .....	6
1.2.5 酒店管理员功能 .....	7
2 系统结构设计.....	9
2.1 数据模块 .....	9
2.1.1 用户类 .....	9
2.1.2 酒店类 .....	10
2.1.3 客房类 .....	10
2.1.4 订单类 .....	11
2.1.5 其他数据相关类 .....	11
2.2 界面模块 .....	11
3 系统详细设计.....	12
3.1 类结构设计 .....	12
3.1.1 用户类 .....	12
3.1.2 酒店、客房、订单类 .....	12
3.1.3 消息与评价类 .....	13
3.1.4 DataBase 类 .....	13
3.1.5 Platform 类 .....	14
3.1.6 常用助手类和自定义标题栏类 .....	15
3.1.7 主界面类 .....	15
3.1.8 登陆界面类和注册界面类 .....	15
3.1.9 顾客界面类 .....	16
3.1.10 平台管理员界面类 .....	16
3.1.11 酒店管理员界面类 .....	17
3.2 界面结构设计 .....	17
3.2.1 登陆界面和注册界面设计 .....	17
3.2.2 顾客界面设计 .....	17
3.2.3 平台管理员界面设计 .....	18
3.2.4 酒店管理员界面设计 .....	18
3.3 关键设计思路 .....	18
3.3.1 容错功能设计 .....	18
4 项目总结.....	19
4.1 项目亮点 .....	19
4.1.1 功能齐全，考虑周全 .....	19
4.1.2 美化界面，操作简便 .....	19
4.2 开发难点 .....	20
4.2.1 数据的结构以及成员设置 .....	20

5 心得体会 .....	20
5.1 自学能力 .....	20
5.2 调试技巧的更上一层楼 .....	20
5.3 臻于完美的追求 .....	20

# 1 系统功能设计

## 1.1 总体功能描述

本项目实现了一套单用户的基于三方平台的酒店预订服务与管理系统。利用该系统，酒店的管理人员可以方便快捷地管理自己的酒店（或者一个经理旗下有多个酒店，也可同时管理旗下所有酒店），自由地创建或删除酒店的房型，编辑客房所具有的属性，管理与当下酒店所关联的订单，赋予酒店管理员确认退款的功能，以及一键发送催促付款消息的功能。顾客可以与酒店管理员进行消息通信，可以通过预订服务页面的关键词自动索引搜索，以及选择入住日期、退房日期和房间数量，来快速下单（不需跳转至其他窗体页面），同时，预订服务页面在平常需要具备的基于好评以及价格高低的排序外，还增加了基于首汉字拼音排序的功能，方便顾客寻找想要居住的酒店的房型。平台管理员在这之间起到不可或缺的桥梁作用，平台管理员拥有对在平台上展示的酒店审核通过和重新审核的两大管制功能，在平台上只有被审核通过的酒店才有机会向顾客展示自身的房型。总而言之，本项目的设计目的是满足三方不同的需求。

用户类型	功能名称	功能描述
顾客	修改个人信息	修改密码、用户名、手机号
	消息交流	可与酒店管理员交流
	酒店房型选择	选中所在行的客房
	客房预订	创建成功订单
	支付订单	支付订单金额
	申请退款	申请退还订单金额
	取消订单	取消已创建的订单
	完成订单	相当于退房操作
	完成评价	对已完成订单做出内心满意度评价
	充值余额	增加账户余额
平台管理员	审核酒店	将待审核的酒店置为已核实
	重新审核	将已核实的酒店置为待审核
	修改个人信息	修改密码、用户名、手机号
	查看用户	查看当前平台已注册用户的账户信息
酒店管理员	创建酒店	创建一个新的酒店
	删除酒店	删除一个已创建的酒店
	编辑酒店信息	对已创建的酒店信息进行编辑
	创建房型	创建指定酒店下的新的房型
	删除房型	删除一个已创建的房型
	编辑房型信息	对已创建的房型信息进行编辑

	消息交流	可与顾客交流通信
	确认退款	处理顾客的退款申请，退还指定订单的金额
	催促付款	一键发送提醒付款消息给指定顾客
	修改个人信息	修改密码、用户名、手机号

## 1.2 功能流程描述

### 1.2.1 登录/注册流程

程序开始时默认为顾客登录界面。用户可以通过点击在登陆界面右上方的按钮和选择下拉菜单中的用户类别在三种用户间切换（顾客，平台管理员，酒店管理员）。

在登录界面可以点击最下方的按钮打开注册界面进行新用户注册，本项目使用 11 位手机号登录，用户名和密码只支持大小写英文和数字输入，使用正则表达式规范用户的输入。除此之外，考虑到平台的安全性，在注册界面只给出酒店管理员和顾客身份的注册。平台管理员由系统设计者指定生成（这里笔者在程序之外进入数据库手动生成）。注册时需要确认密码，两次密码一致方可注册成功；已注册用户通过手机号和密码登录，当密码正确时，可以登录到对应用户类别的界面。

### 1.2.2 用户信息修改功能

平台上拥有三大类用户，顾客，平台管理员，酒店管理员，三方各自对应的平台界面都有相对应的个人信息修改功能。

**编辑信息** 顾客、平台管理员和酒店管理员可在个人信息界面点击编辑键，启用编辑个人信息模式，三者的密码、手机号和用户名皆可修改，同时系统自动检索是否使用过相同的用户名和手机号。在顾客界面编辑信息同时也会启用充值余额模式，充值余额输入框可输入需要充值的余额，点击保存即可实现信息保存和余额充值。

### 1.2.3 平台管理员功能

平台管理员承担着审核酒店、查询用户的两大功能。

**审核酒店** 平台管理员可以对酒店管理员创建的酒店的信息进行审核，审核通过，酒店管理员创建的酒店在平台上才拥有权限被顾客看到并订购其中的客房。

审核通过的酒店也可以被平台管理员进行重新审核，回到待审核状态中，取回在平台上得以展示的权限。

**查询用户** 平台管理员可以查看到所有在平台上注册上的非平台管理员类的用户，并可以查看到用户的密码、用户名和手机号信息。

### 1.2.4 顾客功能流程

顾客是预订酒店环节的核心之一，整个流程中，会与预订的酒店进行多次交互，实现订单状态的更新，整个预订流程如图 1 所示。

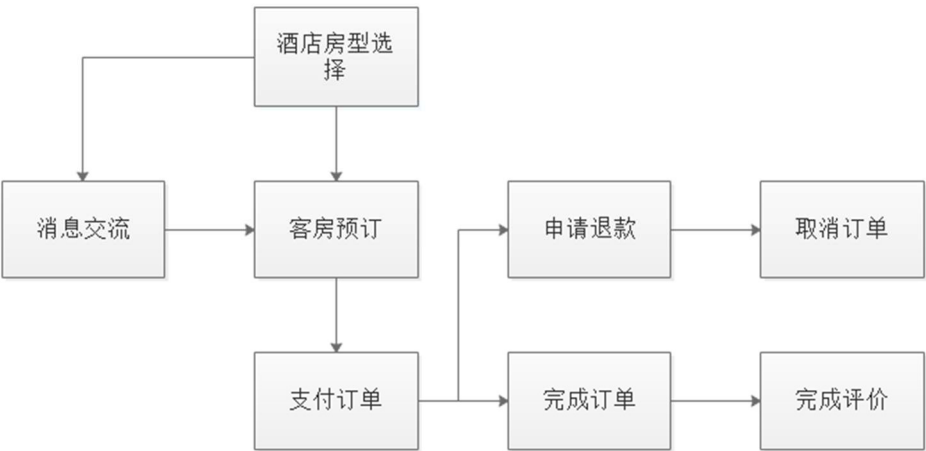


图 1

**酒店房型选择** 单用户模式下，在预订服务页面，顾客可查看所有已经审核通过的酒店的客房信息，在预订服务页面的上方，顾客可通过三个方面（地区名，酒店名，房型名）的自动索引搜索来获取顾客想要的预订的客房。顾客可通过单击表格栏的最后一栏“查看详情”获得当前客房的详细信息。

**消息交流** 顾客若想再进一步通过咨询酒店负责人获得更多信息或者请求更多的优惠，可通过双击当前客房信息所在的行，打开与酒店负责人的消息对话框，可实现与酒店负责人的消息交互。

**客房预订** 顾客确定想要预订的客房后，选择好入住日期和离开日期以及房间数量，即可点击确认预订，跳出信息对话框，再次确认是否下单。点击确认便可生成订单，可回到我的主页查看订单。

**支付订单** 确认订购生成的订单的状态自动置为“未支付”，顾客可通过确认支付的按钮来更改订单状态，将其状态置为“已支付”，若余额不足平台会给出警告提示余额不足，反之，则会给出信息提示支付成功，并成功扣取顾客的余

额。

**完成订单** 顾客在支付完成后，可以点击完成订单按钮，将订单的状态置为“已完成”，表示顾客已完成从入住到退房，即此订单关联的房间不再处于被预订状态。此后订单就作为一份已经结束生命周期的订单。

**申请退款** 顾客在支付订单后，可拥有权限向订单关联的酒店方申请退款，并将订单状态置为“待退款”。在酒店确认退款之前的期间内，顾客没有任何权限修改订单状态。酒店方确认退款后可将订单状态置为“已退款”。

**取消订单** 在酒店确认退款成功后，即订单状态为“已退款”，或者在顾客支付订单之前，顾客可拥有“取消订单”的权限，点击取消订单可将订单状态置为“已取消”，取消后的订单将作为一份已经结束生命周期的订单，不再对顾客和酒店方开放任何权限。

**完成评价** 顾客在完成订单后，订单状态虽然已经置为“已完成”，但此时顾客还拥有一个完成对此订单评论的权限，通过点击订单行的最后一个下拉菜单，选择从“一星好评”到“五星好评”中的其中一个，点击完成评价即可做出评论。同时限制对一份订单只能做出一次评价（详见////）。

### 1.2.5 酒店管理员功能

酒店管理员不仅是预订酒店环节的核心之一，更是整个平台系统中最重要的一环。功能流程如图 2 所示。

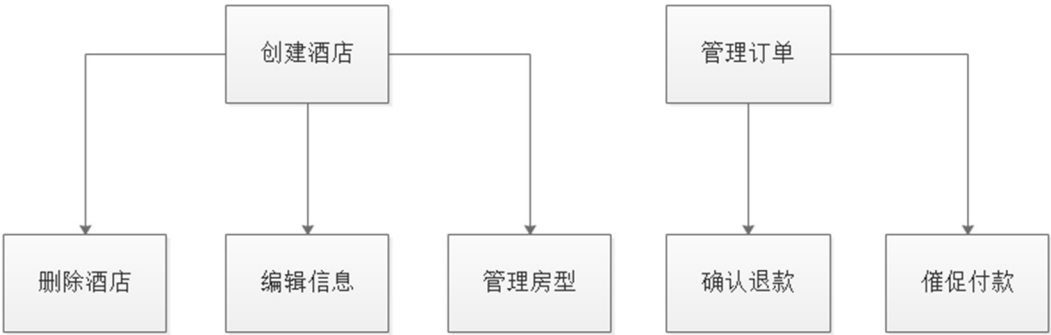


图 2

**创建酒店** 酒店管理员注册账号后，可通过创建酒店来上传自己管理酒店的信息，酒店的信息除了基本的名称和描述之外还需要上传酒店的图片才能创建成功。创建的酒店的审核状态自动置为“待核实”。

**删除酒店** 酒店管理员可对自己曾经创建的酒店进行销毁，销毁一个已创建的酒店会连同删除该酒店名下的所有客房。

**编辑信息** 酒店在创建之后，酒店存在期间，酒店管理员就可以拥有对该酒店信息的编辑权限，但只限于修改酒店名称和酒店描述。

**管理房型** 酒店存在期间,酒店管理员也同样具有对该酒店房型管理的权限。酒店管理员可依据实际情况选择创建相应的房型,或者删除相应的房型,删除房型时会自动把所有关联的订单给全部删除。同时房型创建后,存在期间,酒店管理员就拥有了对该房型信息的编辑权限,可根据实际情况,及时更新客房数量、是否优惠,优惠折数,客房描述,客房图片,房型名称。

**催促付款** 当顾客预订酒店客房的订单长时间未进行支付,酒店管理员可点击催促付款,实现快捷发送该顾客催促付款消息。

**确认退款** 当顾客通过申请退款改变订单状态为“待退款”时,酒店管理员可进行对订单退款的最终确认,并将订单状态置为“已退款”,反馈给顾客,自动将订单金额退回顾客账号余额中,实现与顾客的交互。



## 2 系统结构设计

### 2.1 数据模块

整个系统的数据主要分为四种：用户数据，酒店数据，客房数据，订单数据。每种数据都需要有相关的类（可能多个）进行管理。笔者从项目构思开始就坚定地去使用数据库来完成项目。本项目使用 SQLite 数据库实现数据的持久化。在这里需要声明的是本项目利用 type 值的不同来实现对应不同种类的数据。如表 1 所示。

数据类名	Type(int)
顾客类	0
平台管理员类	1
酒店管理员类	2
订单类	3
客房类	4
酒店类	5
评价类	6
消息类	7

表 1

#### 2.1.1 用户类

用户类型一共分为三种。定义用户类别为 0-2 的整数，从小到大依次为：顾客、平台管理员、酒店管理员。不同类型的用户需要包含的数据也不尽相同。如表 2：每种类型用户的数据 所示。

用户类型	数据
顾客	编号、手机号、用户名、密码、持有订单数、余额
平台管理员	编号、手机号、用户名、密码
酒店管理员	编号、手机号、用户名、密码

表 2：每种类型用户的数据

user.h 中定义了用户类型的基类 MyUser，和继承自 MyUser 的 Customer 类。MyUser 为一般的用户类，包含用户的基本信息，酒店管理员和平台管理员可直接定义为 MyUser 类的对象。顾客类（Customer）从 MyUser 中继承，并添加独有的数据成员。所有用户类的 UML 图如图 3 所示，其中省略了构造函数。

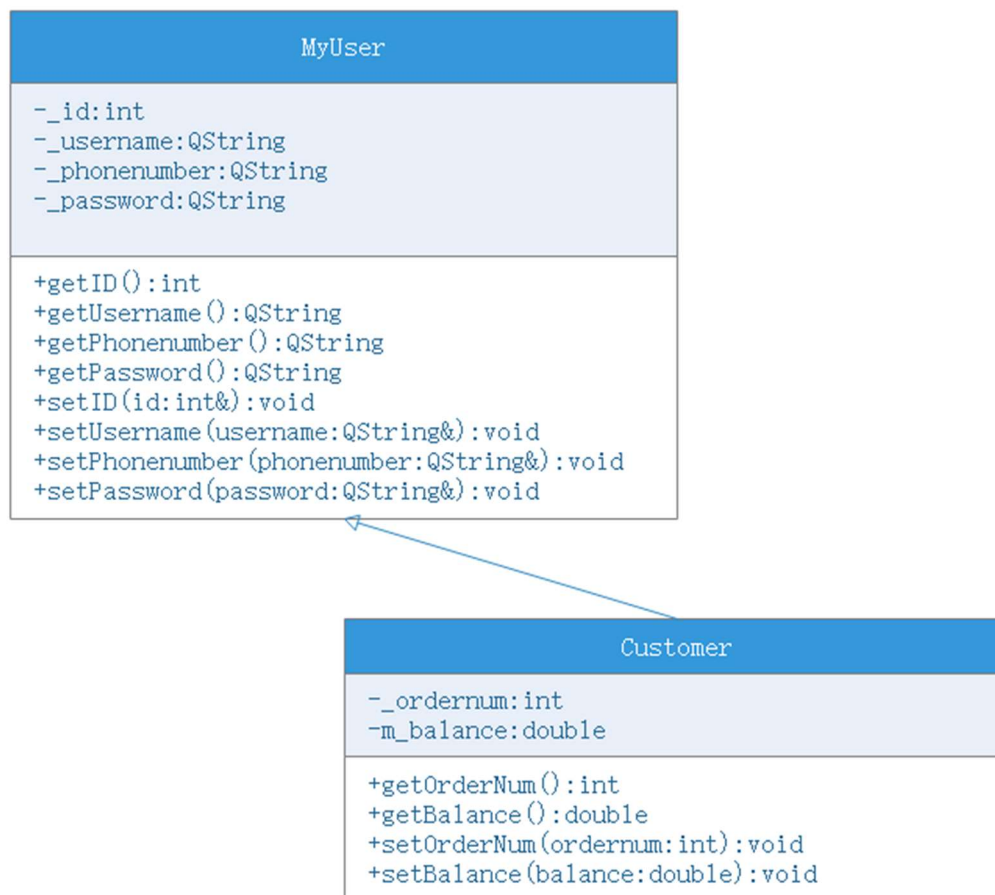


图 3：用户类 UML 图

## 2.1.2 酒店类

酒店类可以用来接载已在数据库中存储的酒店或者在平台上新创建的酒店对象，其中具有的数据：酒店编号、创建者编号（即对应酒店管理员的编号）、酒店名称、酒店所在城市、酒店描述、审核状态以及上传图片的存储地址。酒店类具有和用户类一样的获取成员属性的接口和修改成员属性的接口。

## 2.1.3 客房类

客房类可以用来接载已在数据库中存储的客房或者在平台上新创建的房型对象，其中具有的数据：客房编号、所属酒店编号、客房数量、被预定数量、价格、优惠折数、是否优惠的状态、客房类型（即客房名字）、客房描述以及上传图片的存储地址。其中获取成员属性的接口和修改成员属性的接口也和用户类一样。此外在客房类还加入了两个方便增减被预定数量的修改成员属性接口。（详情见 3.1.2 酒店、客房、订单类）

## 2.1.4 订单类

订单类可以用来接载已在数据库中存储的订单或者在平台上新创建的订单对象，其中具有的数据：订单编号、客房编号、顾客编号、预订房间的数量、入住日期、退房日期、需付金额、订单状态。其中获取成员属性的接口和修改成员属性的接口也和用户类一样。值得一提的是订单状态共有六种对应状态，并且配备有相对应获得对应状态接口。（详情见 3.1.2 酒店、客房、订单类）

## 2.1.5 其他数据相关类

- ◆ 消息类 **Message**：用来封装顾客和酒店管理员相互发送的消息，包含消息编号、发送者名称、接收者名称、发送日期、发送内容、回复消息的编号。顾客界面和酒店管理员界面均会使用消息类来管理消息。
- ◆ 评论类 **Comment**：用来封装顾客对酒店的评分，包含评价编号、订单编号、酒店编号、评价星数。顾客界面会使用封装好的评论类来做出评价。
- ◆ 数据库类 **DataBase**：用来封装 SQLite 数据库的常用操作，简化了平台类（详情见////）中对数据库的使用。
- ◆ 平台类 **Platform**：用来封装对数据库的内容提取到内存和将内存需要改动的地方写入数据库的操作，除此之外还封装了三方界面上需要提供相应功能的操作，包装成接口供界面使用。

## 2.2 界面模块

本项目的界面部分主要由五个界面构成，即开始界面、注册界面和三种类型用户的界面。用一个 **QMainWindow** 类的 **ui** 界面接载五个界面的指针，根据用户的操作调用出不同的 **ui** 界面。在顾客界面类和酒店管理员界面类各自都内嵌有消息细节界面类指针外，在顾客界面类中还内嵌有专门用于查看详情的界面指针，在酒店管理员界面类中内嵌有建立酒店界面类指针、管理房型界面类指针和酒店细节界面类的指针，在管理房型界面类也还内嵌有客房细节界面类的指针，总而言之，界面的相互切换时存在十分严谨的内嵌关系和相互独立关系。

## 3 系统详细设计

### 3.1 类结构设计

#### 3.1.1 用户类

**MyUser** 包含编号、用户名、手机号、密码四个私有类型成员，提供了基本的获取和修改成员属性的公有接口。

**Customer** 继承自 **MyUser** 类，增加了数据成员 `_ordernum` 和 `m_balance` 并提供了相应的接口的来获得和修改相应的成员属性。

#### 3.1.2 酒店、客房、订单类

**Hotel** 包含酒店编号、创建者编号(即对应酒店管理员的编号)、酒店名称、酒店所在城市、酒店描述、审核状态以及上传图片的存储地址，以上所有成员均为私有成员，提供了基本的获取成员属性的公有接口。出于在平台上设计功能的考虑以及实际生活中的经历，酒店一经创建再次可以修改其中信息时之只能更改酒店名称、酒店描述、审核状态以及上传图片的文件，故设计的修改成员属性的公有接口只有以上四个成员的。

**Room** 包含客房编号、所属酒店编号、客房数量、被预定数量、价格、优惠折数、是否优惠的状态、客房类型(即客房名字)、客房描述以及上传图片的存储地址，以上所有成员均为私有成员，提供了基本的获取和修改成员属性的公有接口，除了房间编号以及创建者编号之外。

**Order** 包含订单编号、客房编号、顾客编号、预订房间的数量、入住日期、退房日期、需付金额、订单状态，以上所有成员均为私有成员，提供了基本的获取成员属性的公有接口。订单的创建就只能进行修改其中的订单状态，因此只有一个修改订单状态的公有接口

**Hotel** 中有一个 `bool` 型的成员，为了便于存入数据库的操作，提供了一个 `bool` 转换 `int` 的简单接口：

- `int getIsVerifyInt() const`: 审核状态为真返回 1，否则返回 0。

**Room** 中有一个关于已预定数量的成员，在用户下单时，若通过获取成员属性的接口先获得值，再调用修改的接口会带来语句上的繁琐，于是提供了两个快速增减的简单接口：

- `void declineReserve(int roomnum)`: 已预定数量  $+=$  `roomnum`。
- `void addReserve(int roomnum)`: 已预定数量  $-=$  `roomnum`。

**Order** 中有一个订单状态成员(`int` 型)，在界面中输出时需要转换为相对应状

态的字符串，故设计了一个简单接口：

- `QString toStringBystate()`：利用 `switch` 语句执行相应的 `case` 返回字符串，返回内容如表 3：订单对应状态所示。

state(int)	订单对应状态
0	未支付
1	已支付
2	待退款
3	已退款
4	已完成
5	取消订单

表 3：订单对应状态

### 3.1.3 消息与评价类

**Message** 包含消息编号、发送者名称、接收者名称、发送日期、发送内容、回复消息的编号六个私有类型成员，提供了基本的获取成员属性的公有接口。在修改成员属性的接口只设置了修改回复消息的编号的接口，这是为了实现用户之间的通信而设计的接口。

**Comment** 包含评价编号、订单编号、酒店编号、评价星数四个私有成员，提供了基本的获取成员属性的接口。对外修改属性的接口只有设置评价星数的公有接口。

### 3.1.4 DataBase 类

**DataBase** 包含数据库名称，表格名称，参数名，主键，一个内嵌 `QSqlDatabase` 的对象，数据类型五个私有成员属性，除此之外还包含一个打开数据库操作的私有成员函数。其中并不设置任何获取成员属性的接口，只设置了两个对外的修改成员属性的接口，一个可以修改表格名，一个可以修改参数名。其主要成员函数：

- `void setTableName(QString tbname)`：在执行修改成员属性之后同时根据 `tbname` 的内容修改作为私有成员属性之一的数据类型的值。
- `DataBase(int type)`：专门用于快捷建立起与数据库用户表格的连接。
- 通过对 SQL 语句执行的封装，实现了 `insert`（插入数据），`update`（更新数据），`select`（选择某一列的所有数据），`clear`（清空表）、`count`（返回行数）等数据库的常用操作。

### 3.1.5 Platform 类

**Platform** 按照大作业的要求，不能频繁对数据库进行操作，于是笔者在范老师的指导提醒下，设计出了一个 Platform 类来作为将数据库的内容漏到内存中存储的载体，这是整个程序的核心组成部分。Platform 中根据前文中提到的八种数据类型，分别定义八个私有成员，使用 QMap 容器，利用键值对建立起数据库表格中的 id 项与相应数据对象的指针的映射，同时自定义结构体类型 EditFlag（操作标记），在 Platform 类定义一个私有成员，使用 QVector 容器存储在程序运行中产生 EditFlag，走与大多数人不一样的路，在程序运行末尾存入数据库的操作不采取先清空所有表格，再重新存入的耗时低效率的做法，利用 EditFlag 中元素索引到数据库中相对应的表格指定行进行修改。其主要的成员函数为：

- void ReadDatabase(): 执行该函数可实现将数据库中所有内容读取到内存中。
- void WriteIntoDatabase(): 执行该函数可根据 QVector 容器中 EditFlag 来一一执行对数据库的更新。
- bool checkValid(QString phonenumber, QString password, int type): 根据 type 的不同，检索当前身份登录的有效性。该函数提供给用户登录界面检查登录有效性的功能。
- int getMaxID(int type): 根据 type 即数据类型的值的不同来返回 ID 的最大值+1，用于生成新的对象，并标志着数据库新的一行的插入标志。
- bool registerSearch(QString value, int searchItem): 根据 searchItem 的不同检索用户名或者手机号 value 是否已经被注册过，返回 bool 类型的值。
- bool foundSearch(QString value, QString city): 返回酒店名是否在某一地区被注册过的真假
- void clear(): 释放 Platform 内 QMap 建立起键值对映射的 value（指针）的内存空间
- bool roomFoundSearch(int hotelid, QString roomcate): 返回 hotelid 索引下的酒店旗下的房型名是否有被创建过
- bool belongToHotels(int roomid, int founderid): 检测 roomid 索引指向的房间是否属于 founderid 索引指向的酒店管理员所管理的酒店
- bool refund(QString username, double cost): 提供给酒店管理员界面的功能以实现确认退款操作，并且返回一个 bool 型的值，提示系统是否退款成功

### 3.1.6 常用助手类和自定义标题栏类

**Commonhelper** 该类的设计主要用于实现对调用 QSS 样式的操作的封装，和对调用自定义标题栏的操作的封装以得到简便的操作。这些封装起来的操作都定义成了 static 静态类型，以方便不需创建对象即可调用。主要函数有：

- void setstyle(const QString &style, QApplication &a): 加载 QSS 样式美化界面
- void setTitleBar(QWidget \*a): 调用自定义标题栏

**TitleBar** 该类的设计主要用于实现各个界面使用自定义的标题栏以实现界面美化。

### 3.1.7 主界面类

**MainWindow** 主界面类主要用于接载四大主要界面（三方各自界面和登陆界面）的指针，根据不同的操作跳转至不同的界面，主要的函数有：

- void isLogin(): 登录成功之后根据不同类型进入不同用户身份的界面
- void LogOut(): 点击退出登录后回到登录界面

### 3.1.8 登陆界面类和注册界面类

登陆界面类 Login 利用 m\_type 标识当前用户类别，用 isLogin 来标识是否登录成功，Login 内嵌 Platform 类指针对象 m\_platform，利用 m\_platform 提供的 checkValid 接口来实现输入的手机号和密码是否能够成功登录，构造函数中默认 m\_type=0，即默认一开始用户为顾客身份登录，同时 Login 也内嵌注册界面类指针。主要的函数有：

- void on\_\_signUp\_clicked(): 点击打开注册界面
- void on\_\_signIn\_clicked(): 点击登录按钮就会触发一次该槽函数，槽函数内部调用 m\_platform 上的 checkValid，检测是否登录成功，成功则跳转界面

注册界面类 Register 利用 \_type 来标识用户注册身份，初始化时默认为 0，即顾客注册身份，Register 同样内嵌有 Platform 类指针对象 m\_platform，每次点击注册按钮即可调用 m\_platform 上的 registerSearch 函数来检测用户名和手机号是否已经被使用过。主要的函数有：

- void on\_pushButton\_clicked(): 检测注册有效性，若有效则完成注册操作

### 3.1.9 顾客界面类

顾客界面类 `CustomerWidget`，利用相关按钮可跳转至不同的标签页，主要方法有：

- `void setup()`: 初始化整个顾客界面类窗口
- `void setupMsgPage()`: 初始化我的信息页面
- `void setupOrderPage()`: 初始化我的订单页面
- `void setupRoomPage()`: 初始化预订服务页面
- `void autoSearchForCity(const QString &text)`:
- `void autoSearchForHname(const QString &text)`:
- `void autoSearchForRcate(const QString &text)`: 以上三个是实现自动动态搜索的槽函数
- `void checkOutValid(const QDate &date)`: 该槽函数使得退房日期的编辑框无法调节到入住日期之前
- `void roomNumBoxChangeValue(int row,int column)`: 根据当前客房的预定数量和总数量自动调节可供选择的房间数量编辑框的上限值
- `void clickToDialog(int row,int column)`: 双击客房所在行可打开与当前酒店管理员的消息对话框
- `void on_payButton_clicked()`:
- `void on_refundButton_clicked()`:
- `void on_finishButton_clicked()`:
- `void on_cancelButton_clicked()`:
- `void on_commentButton_clicked()`: 以上五个函数实现了顾客支付、申请退款、完成订单、取消订单、完成评价的五大功能
- `void buttonRespond(int row,int column)`: 该槽函数使得顾客在点击不同的订单，按钮进行不同的响应给予该给予的权限
- `void orderByClement(int logicalIndex)`: 排序
- `void clickToGetDetails(int row,int column)`: 点击查看详情可打开 `GetDetails` 的界面，一睹全部详细信息关于当前客房

### 3.1.10 平台管理员界面类

平台管理员界面类 `AdminWidget` 也同样利用相关按钮跳转至不同的标签页，与顾客界面类相似，一样具有对应的初始化方法，在此不再进行展出，只展出该界面特有功能，主要方法有：

- `void on_passButton_clicked()`: 实现了审核酒店通过的功能
- `void on_reverifyButton_clicked()`: 实现了重新审核酒店的功能



### 3.1.11 酒店管理员界面类

酒店管理员界面类 `HotelManagerWidget` 也同样利用相关按钮跳转至不同的标签页，与顾客界面类相似，一样具有对应的初始化方法，在此不再进行展出，主要方法有：

- `void on_newHotelButton_clicked()`: 打开创建酒店界面
- `void on_getInfoButton_clicked()`: 打开得到酒店详细信息的界面
- `void on_deleteHotelButton_clicked()`: 删除酒店的按钮
- `void on_updateButton_clicked()`: 基于不同页面的 `index` 实现当前页面的刷新
- `void on_manageRoomButton_clicked()`: 打开该酒店的管理房型界面
- `void replyOrCheck(int row,int column)`: 点击回复或查看消息
- `void on_urgeButton_clicked()`: 一键发送催促付款信息
- `void on_refundButton_clicked()`: 实现确认退款
- `void buttonRespond(int row,int column)`: 根据点击的订单给予不同的按钮使用的权限

## 3.2 界面结构设计

### 3.2.1 登陆界面和注册界面设计

酒店预订与管理系统应该以顾客为中心，故系统的初始界面应该为顾客登录界面。且对于顾客来说，界面应该美观整洁，不应该出现很多供其他类别用户出现使用的按钮。基于以上考虑，登录界面在不同类型的用户使用时仅用身份下拉栏的选择不同来标识。

其次，注册界面也应该做得尽量简单，能够让用户注册得快，使用程序得快。

### 3.2.2 顾客界面设计

顾客界面由三个子页面组成，通过点击左边不同的按钮可跳转至不同的子页面。

**我的主页** 点击我的主页按钮可跳转至我的主页页面，在我的主页又分为我的订单和我的消息两个更细小的分支，顾客可在我的消息页面实现查看和回复功能，在我的订单页面可实现支付、申请退款、取消订单、完成订单、完成评价五大功能

**个人信息** 点击个人信息按钮可跳转至个人信息页面，在个人信息页面可进行查看个人信息或者修改个人信息，在开启个人修改信息的同时也会开启充值余

额功能，此时只需输入想要充值的余额，点击保存即可完成充值，方便用户修改信息的同时也能同时充值。

**预订服务** 点击预订服务按钮可跳转至预订服务页面，进入页面就会立即初始化表格，显示所有已通过审核酒店的所有房型，这样方便用户进行查询，在上方的三个关键词搜索框输入内容可实现自动动态搜索，同时考虑到顾客想要与酒店管理员交流，可双击想要的房型所在行打开与对应酒店管理员的消息对话框，可发送消息，又考虑到表格栏的长度限制，顾客想要完全详细的信息，还提供了一个查看详情的功能，在表格的最后一栏，点击即可打开一个新的详细信息的界面。

### 3.2.3 平台管理员界面设计

平台管理员界面同样由三个子页面组成，通过相关按钮跳转至不同页面。其中拥有的**个人信息**页面与顾客界面一样，在此不再展示。

**查看用户** 考虑到现实情况，平台管理员有权看到非平台管理员身份的所有用户，并知晓他们的密码和用户名和手机号，于是设计出了这个页面。

**管理酒店** 平台管理员在设计的过程中应当具有审核通过和重新审核酒店这两大功能，于是就设计出这个子页面，让平台管理员进行方便的操作

### 3.2.4 酒店管理员界面设计

酒店管理员界面由四个子页面组成，通过相关按钮跳转至不同页面。其中拥有的**个人信息**页面和**我的消息**页面与顾客界面一样，在此不再展示。

**管理酒店** 酒店管理员首先必须有管理自己旗下所有酒店的页面，通过点击上方创建酒店可打开创建酒店的新界面，删除酒店在该酒店没有任何下属房型的情况下才能进行，点击管理房型会打开管理房间的新界面，点击查看信息也会打开可查看或编辑酒店信息的界面。

**管理订单** 根据实际情况，应该实现可对订单实现确认退款的功能，在此之外笔者还加入了一个新的功能催促付款，一键发送消息。

## 3.3 关键设计思路

### 3.3.1 容错功能设计

由于本项目与用户之间交互较多，所以要充分设计容错功能。本项目的容错功能主要通过以下手段实现

**检查函数** 例如在 Platform 类中设计的 checkValid 函数即用来检测登录时密码和手机号匹配的错误性。

**正则表达式** 当对输入格式有要求时，采用正则表达式严格控制，例如手机号输入框首位是 1，且长度不超过 11 位的内容。

**禁用按钮** 为了防止乱点乱触产生错误，会禁用相关按钮，等待点击相应地方并且匹配情况正确时，才给予一些按钮的权限，例如顾客对订单操作时，没有选中任何一个订单，将失去支付、申请退款、取消订单、完成订单、完成评价五个按钮的权限，点击后根据订单的状态响应不同的按钮。

**if else 条件语句** 有些按钮不可避免一直保持可使用，为了防止出现错误，在相应槽函数里假如 if else 语句判断，假如当前没有选中是不会有反应的。

## 4 项目总结

### 4.1 项目亮点

#### 4.1.1 功能齐全，考虑周全

本项目在各个界面的逻辑、功能都力求考虑周全，力求完整、合理、便捷。例如一个十分亮眼的地方就是由百度搜索引擎输入关键词自动出现匹配索引结果的启示，巧妙利用了 `QTableWidget` 中提供的函数实现了自动动态搜索，即搜索结果可随着输入框中内容变化而变化，同时笔者不想跟大多数人一样仅限酒店名的搜索，笔者开发出了基于地区名、酒店名、房型名三者同时作用的自动动态搜索，同时不仅限于基于好评高低和价格高低的排序，同时也可以按照列中内容是汉字，则按拼音进行排序的功能，正序反序均可实现，多种排序作用下相信一定有助于顾客以非常快的速度找到称心如意的酒店以及客房。

此外，笔者还实现了数据库的第二范式，每个表格有且仅有一个主键 `id`，或许可以更进一步地说笔者也一定程度上实现第三范式，虽然未设置互为外键，但笔者的订单列表内可通过 `roomid` 查询相应客房，然后通过相应客房列表内的 `hotelid` 查询到相关酒店。

同时提供了预订服务界面点击查看详情的功能，专门设计出全新的界面去让顾客对该房型下所有信息一目了然

#### 4.1.2 美化界面，操作简便

为了美化界面，提升用户体验，大大提高用户使用系统的愉悦性，本项目使用了在网上查询许久的 `QSS` 样式进行修饰，同时为了实现在不同系统中的统一和进一步美化，去除系统自带的边框和标题栏，使用自定义标题栏。

同时本项目连接了许多的信号与槽，方便顾客进行操作，同时也设置有快捷键，例如在登录界面，用户输入手机号和密码之后可按下回车快速登录。

## 4.2 开发难点

### 4.2.1 数据的结构以及成员设置

本项目中涉及到很多种类的数据，这也就意味着数据库表格的数目之多，通过分析不同数据之间的连系，以及范老师上课所提及的数据库第三范式的实现的启示下，数据内包含的成员以及参数经过多次尝试和修改，才得到一个较为合适的结果，现有的结构才能被确定。

## 5 心得体会

### 5.1 自学能力

本次大作业布置的项目大部分知识都需要自己上网有耐心地、有筛选性地查找资料来学习。例如数据库 SQLite，Qt 等。这些东西在网络上拥有许多鱼龙混杂的博客说明。在浩如烟海的网络资料里选择出正确的并且合适自己学习的资料，并将所学所用运用到项目构建中，是一种让人终生受益的能力。

### 5.2 调试技巧的更上一层楼

在面对一种自己之前未曾接触过的软件，在脱离了曾经赖以生存的 VS 后，必须要去适应较为崭新的 debug 界面，曾经在 C 语言学会的调试技巧还远远不足，在不断摸索和总结，懂得合理利用 qDebug() 这一做法把一些作为传参参数的关键变量的值输出到输出框，再进入到函数将相应的参数输出到输出框，或者输出一些字符串提示，可以有效提高 debug 效率，其次是 debug 的收放自如，有时候不必拘于小节，可先暂时搁置，等到过段时间说不定就出了新的灵感找到 bug 所在。

### 5.3 臻于完美的追求

尽管笔者做出的项目功能差强人意，但笔者在过程中一直追求最完美的功能实现和越多越好越高级的功能的添加。这种追求不断激励着笔者去完善项目，去寻找新的知识和新的技巧运用到项目中。这种充实感让笔者认识到了把这些追求推广到学习的各个学科乃至生活，相信都能给自己的学习和自己当前的状况带来可观的改善。