

# 旅店管理软件

## 设计文档

小组成员：---、---、---

### 一、设计任务描述

一个功能完善的旅馆管理系统能够为旅店前台提供高效的工作环境。本次数据结构课程设计旨在设计一个实用旅店管理软件，完成入住登记、离店收费、统计查询等功能。

### 二、功能需求说明及分析

1. 旅馆客房标准类型不少于 4 个，并且有不同标准的房间，即有标准价位和优惠价位；
2. 建立房间分布平面图；
3. 满足旅客的要求，包括房间类型以及房间位置等；
4. 每天十二点后退房按半天计算，十八点后退房按一天计算；
5. 旅客入住时开始登记，离店时结算房价（含食品消耗费、通话费及损坏赔偿费等）；
6. 旅客入住时须填写旅客信息表：姓名、性别、年龄、身份证号、工作单位、所在城市等；
7. 用图形显示旅店某层房间分布图，用不同颜色标注该房间是否已有旅客入住；
8. 统计当前哪些房间已有旅客入住，入住旅客人数，今日离店的旅客数，已接收到的预定订单数，今日预定旅客的入住人数，今日收入等；
9. 实时查询旅客所在房间号，以及查询某房间的旅客姓名等；
10. 建立日志文件，对旅客信息、入住离店等进行记录。

### 三、总体方案设计说明

#### 1) 实现任务的方法

由于要设计出良好的人机界面，我们采用跨平台的 C++图形用户界面应用程序框架 Qt 所提供的功能与 C++知识组合起来创建了我们的旅店图形界面应用程序。

- 1、“顾客信息”选项卡中显示了顾客信息列表，可根据姓名等旅客的个人信息模糊查找已入住旅客信息(在设置中提供了模糊查找/精确查找的设置)。在查找完成后可以点击“清除所有”将搜索重置。



图 1 旅店管理系统顾客信息模块示意图

- 2、“客房信息”选项卡提供了一目了然的房间分布图，方便旅店管理者全面的掌握旅店入住、预约等情况，在点击按钮弹出的信息对话框中可以了解房间的类型、价格等信息以及顾客的相关信息；同时如果旅客对特定房间有需求，允许前台指定房间入住/退房。当有住户应当离店时会弹出窗口提醒前台催促，并在“客房信息”中用红色显著标明，提醒前台及时处理。



图 2 旅店管理系统客房信息模块示意图

- 3、该旅店管理系统提供了预约信息列表，在列表中列出了已经预约房间的客人的信息，可根据姓名、手机号码等个人信息快捷查找相关旅客的预约登记信息。当一个客人预定多间房间时将显示为多个条目，通过搜索名字即可进行定向筛选。

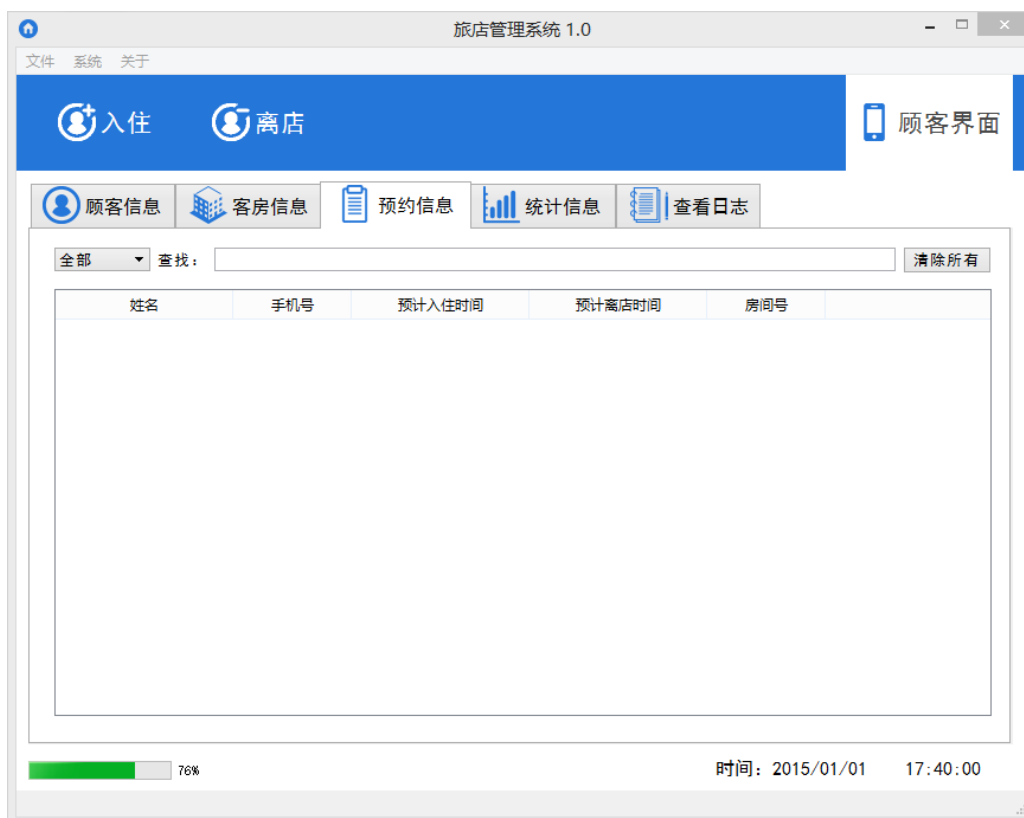


图 3 旅店管理系统预约信息模块示意图

- 4、“统计信息”选项卡提供了统计信息列表，使用户能够清楚了解当前旅店房间状态，当前旅店房间入住率，当前旅店顾客状态等统计信息，方便对旅馆进行管理。



图 4 旅店管理系统统计信息模块示意图

- 5、“查看日志”模块将日志按照时间顺序罗列，使我们清楚的了解旅客入住、预约、离店等相关动作的状态。通过以时间或内容为条件进行模糊搜索可以立即得到想要的信息（例如搜索“入住”即可查询所有入住日志条目）。



图 5 旅店管理系统查看日志模块示意图

- 6、顾客界面面向顾客而设计，通过“我要预约”“取消预约”和“客房消费”三个模块方便顾客的预约、取消预约、客房服务等操作。顾客可以在这里自助预约客房、取消自己的预约或者（针对于住户）可以进行饮用水等商品的消费。

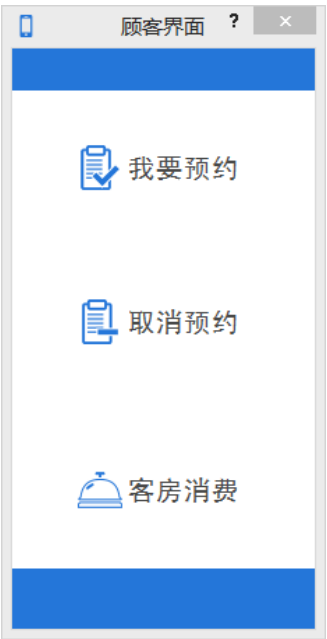


图 6 旅店管理系统顾客界面示意图


- 7、该旅店管理系统提供了登记入住界面，登记信息条目清晰，便于操作。如果是已经预约过的顾客，在登记时选择“是”已经预约，系统会根据顾客姓名自动查找顾客的预约并进行登记入住。如果没有预约，将在符合条件的可住房间内自动选择一间给该顾客入住。



该界面为“旅店客户登记系统”的“登记入住”窗口。窗口顶部有标题“登记入住”和关闭按钮。主标题为“旅店客户登记系统”。下方有一个复选框“是否已经预约过”，旁边有“是”和“否”两个选项。接着是“用户信息”区域，包含“姓名”（文本框）、“性别”（男/女复选框）、“工作单位”（文本框）、“身份证号”（文本框）和“联系电话”（文本框）。然后是“房间信息”区域，包含“入住天数”（数字输入框，当前值为1）和“房间类型”（下拉菜单，当前选中“标准间”）。底部有两个按钮：“登记”和“取消”。

图 7 旅店管理系统登记入住界面示意图

- 8、该旅店管理系统提供了离店结算界面，方便旅客离店登记。顾客在退房时只要提供收据，通过收据上的房间号即可对任一房间进行退房结算处理。系统会自动计算顾客的押金余额并提醒前台退费(或对欠费的客人收费)。



该界面为“旅店客户离店系统”的“离店结算”窗口。窗口顶部有标题“离店结算”和关闭按钮。主标题为“旅店客户离店系统”。下方有一个文本框“请输入房间号：”。文本框下方有一行红色的提示文字：“提示：房间号为201-218...601-618”。底部有两个按钮：“离店”和“取消”。

图 8 旅店管理系统离店结算界面示意图

- 9、该旅店管理系统提供了系统设置界面，方便旅店管理者设置相关房间、物品的价位。为开发测试方便，在系统设置中提供了“时间间隔”的调整，通过调整这一参数可以快速测试各种情况，方便测试。而在系统设置中还可选择是否进行模糊搜索。当“模糊搜索”选中时只要输入部分

信息即可查找到想要查找的条目。而需要精确查找时将此项勾掉即可进行精确查找，输入完整信息找到特定条目。

图 9 旅店管理系统系统设置界面示意图

## 2) 总体结构

在老师提供的程序参考结构的基础上，我们将整个管理系统总体上分为了两大部分：前端使用和后端管理，在这两部分的基础上又对系统进行了深层次的细分，如图 10 所示。

本系统不仅在功能上分为前段使用和后端管理，在开发上还分为前端针对用户的界面和后端的系统内部处理。前端通过 QtCreator 绘制界面并在构造函数中设置相关参数使界面更加清晰美观。而用户点击前端界面的按钮即会触发后端的处理函数，对用户输入的信息进行处理。对于一些特殊操作（例如入住、预定等）会将其记录在日志中方便管理员查看。而系统底端则是时间函数，不仅推动时间轴前进，在特定时刻（如中午 12 点退房、午夜 0 点取消过期预约等）对系统数据进行自动处理。

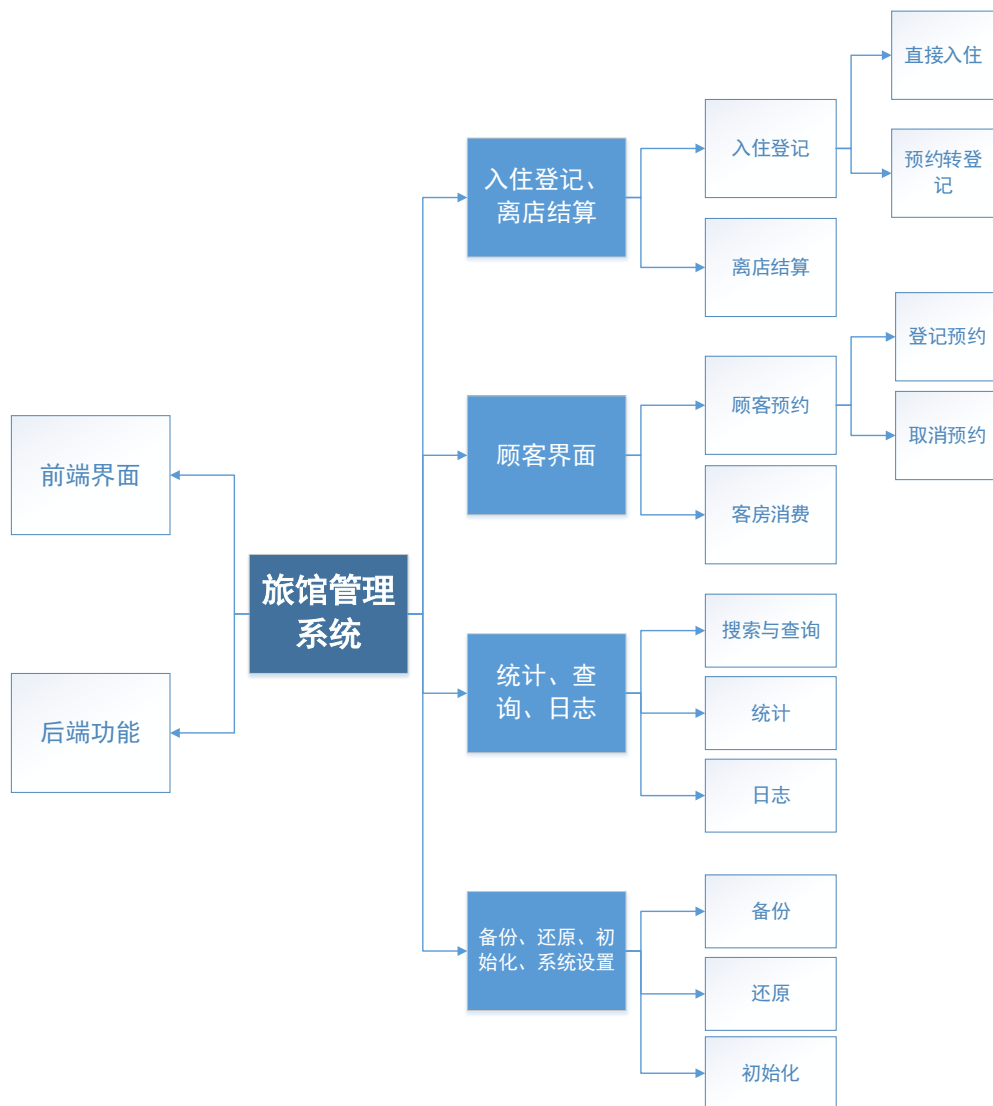


图 10 旅馆管理系统的模块划分

### 3) 模块划分

由上图可知我们将整个旅店管理系统细分成了 6 个主要的模块：主模块、入住登记及预约模块、离店收费模块、统计与查询模块、运行日志模块、系统管理模块。同时，在开发过程中我们进一步对各个模块功能进行了深入的讨论与设计。对其中涉及到的算法进行不断修正改进。

#### ① 主模块

主模块是整个系统的核心，主要实现对键盘、鼠标等外设输入事件的响应，整个系统时间轴的不断推移，以及对其他各个子模块的调用，起到了连接整个程序的作用。在代码中就体现在 `mainwindow.cpp` 和 `main.cpp` 中。在 `mainwindow.cpp` 中，`MainWindow` 的构造函数将各模块的触发按钮、菜单等与功能函数连接起来，其他类内函数执行模块功能。



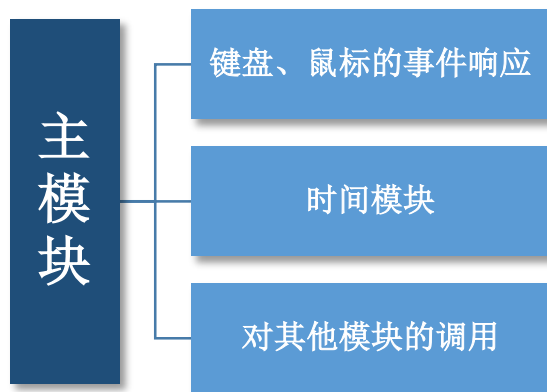


图 11 旅馆管理系统的主模块划分

在本模块中主要涉及：

```
explicit MainWindow(QWidget *parent = 0);
```

将所有后端的功能模块与前端界面连接起来。

```
void createConnection(bool isExist);
```

```
void sqlInitialize();
```

创建数据库，将数据库中的“房间信息”进行初始化，将房间信息写入。

```
void progressUpdate();
```

时间函数，通过进度条来更新时间，并在中午 12 点提示退房和午夜 0 点将过期预约删除

## ② 入住登记及预约模块

此模块主要分为入住以及预约两部分。

**对于入住：**

- 1、对入住旅客进行入住信息登记，其中包括姓名、性别、工作单位、身份证号、联系电话、入住天数和入住时间等信息。
- 2、通过调用统计与查询模块为旅客随机分配可入住房间，如果旅客对房间的位置有特殊要求，则可通过自选的方式入住特定位置的房间；
- 3、将旅客入住的房间信息登记到当前已入住房间表之中；
- 4、调用日志模块将入住信息记录到日志中。

**对于预约：**

- 1、对预约旅客进行预约信息登记，其中包括姓名和联系方式以及入住天数和入住时间等信息；
- 2、通过统计与查询模块为旅客随机分配可预订的房间；
- 3、将旅客的预约信息登记入预约表中；
- 4、若系统时间到达了对应的预约时间，将相应预约房间转变为已入住房间。如果到达预定时间时旅客并未到达旅店入住，并且在第二天 0 点之前还没有成功入住，则系统将自动删除预约信息。

5、调用日志模块将预约信息记录到日志中。

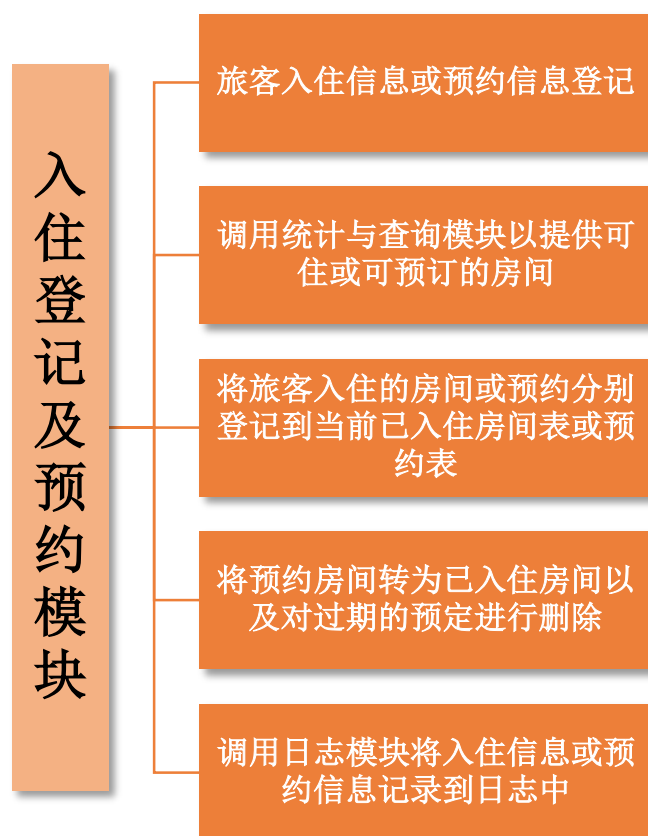


图 11 旅馆管理系统的入住及预约模块划分

在代码中主要体现为：

checkin.cpp 中的 `void checkinDialog::checkinto()`

checkin1dialog.cpp 中的 `void checkin1Dialog::checkinto()`

roominfodialog.cpp 中的 `void roominfodialog::checkin1()`

三个函数，分别对应不同对话框中触发的入住功能模块。

而 reservedialog.cpp 中的 `reserveDialog::reserveDialog(QWidget *parent)` 和 `void reserveDialog::on_okButton_clicked()` 负责预约模块

cancelreservedialog.cpp 中的 `void cancelreservedialog::on_okButton_clicked()` 对用户取消预约进行相应。

### ③ 离店收费模块

离店收费模块主要记录旅客离店结算等相关信息。

首先统计包括食品消费等额外消费的住房总费用，将顾客的押金减去消费金额后进行退费（如果欠费则收款）。成功进行结算后，将旅客所入住房间从当前已入住房间删除，并将相应旅客信息删除，之后调用日志模块将离店信息记录到日志中。

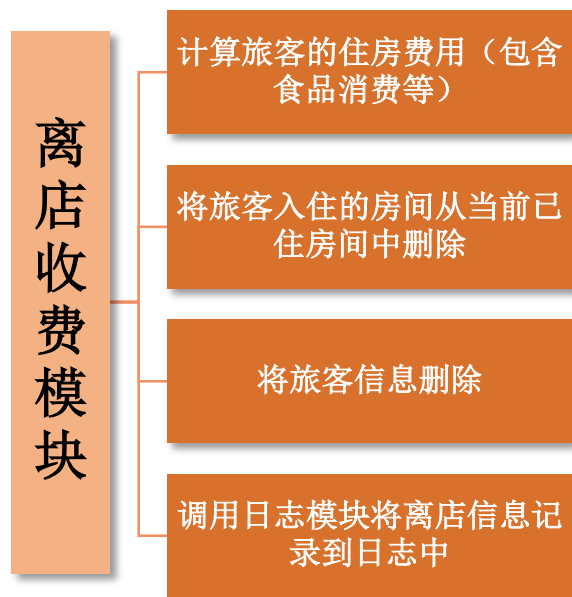


图 11 旅馆管理系统的离店收费模块功能划分

在代码中主要体现为：

checkout.cpp 中的 `void checkoutDialog::checkoutto()`

roominfodialog.cpp 中的 `void roominfodialog::checkout1()`

分别对应两种界面中的离店函数

#### ④ 统计与查询模块

该模块主要分为统计、查询两个方面。

统计主要是统计当前预约及入住情况，入住率、收入等旅店总和信息, 而查询主要查找相应房间入住及预约状态，旅客入住情况，相应旅客信息等, 同时统计与查询的行为将通过调用日志模块写入日志。

在代码中主要体现为：

mainwindow.cpp 中的 `void MainWindow::search()`

`void MainWindow::searchReserve()`、`void MainWindow::searchLog()`

为列表提供搜索查找功能。

`void MainWindow::roomButtonChange()`

绘制房间平面图中的按钮状态以显示房间状态。

roominfodialog.cpp 中的 `void roominfodialog::updateRoomInfoDialog(int num)`

在对话框中显示房间信息和入住客人信息。

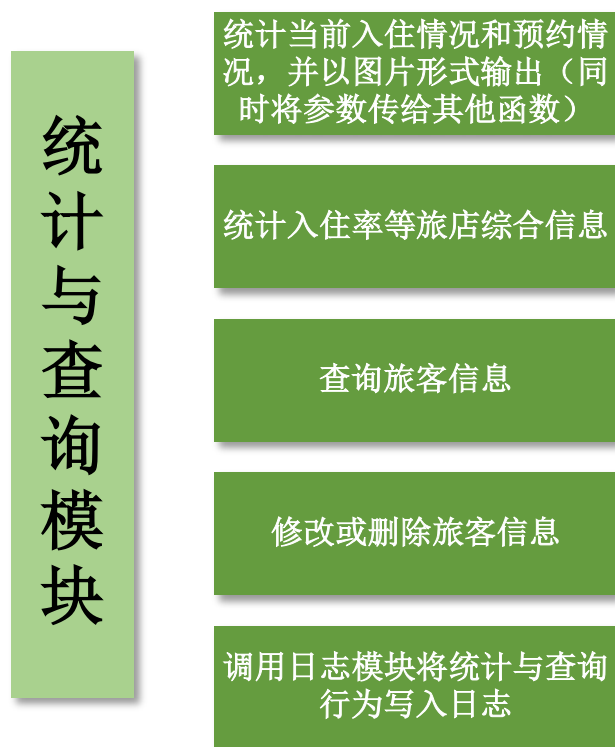


图 11 旅馆管理系统的统计查询模块功能划分

#### ⑤ 系统管理模块

系统管理模块主要是对旅店系统的参数进行修改，并负责备份和还原功能。在打开软件时会询问是否要从已有备份中恢复，然后根据选择执行备份或初始化函数。在关闭软件时会询问是否需要备份，如果需要则对软件进行备份。在软件运行过程中也有备份和还原功能可以直接调用。

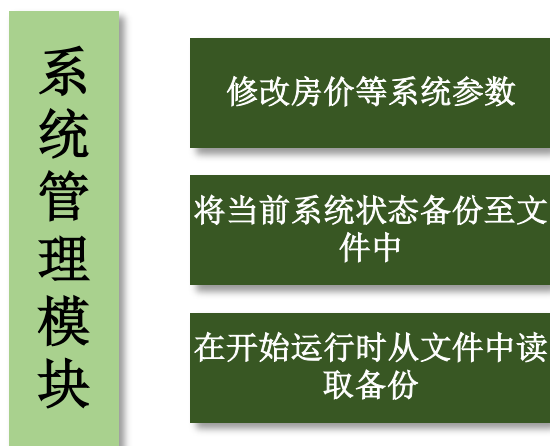


图 12 旅馆管理系统的系统管理模块功能划分

在代码中体现为：

mainwindow.cpp 中的 `void MainWindow::restoreFunction()`

负责从备份中还原系统状态。

mainwindow.cpp 中的 `void MainWindow::backupFunction()`

mainwindow.cpp 中的 `int MainWindow::startRestoreFunction()`

分别负责软件开始时和日常运作时的备份。

mainwindow.cpp 中的 `void MainWindow::initialFunction()`

负责软件开始时的初始化。

setting.cpp 中的 `void setting::pricechange()`和 `void setting::pricerreset()`

负责软件参数设置。

## ⑥ 运行日志模块

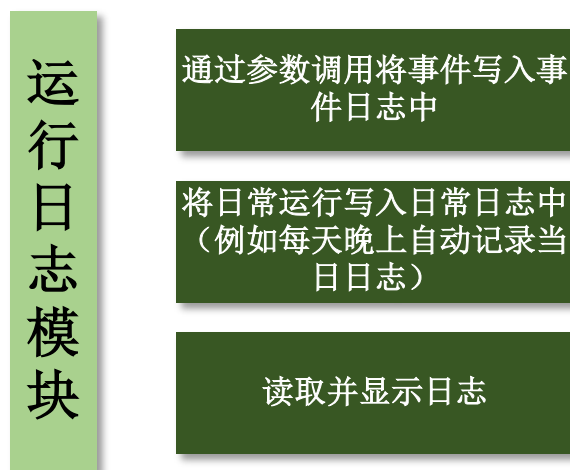


图 12 旅馆管理系统的日志模块功能划分

运行日志模块主要涉及对每次预约，取消预约，入住，离店，查询，统计等动作的记录，并及时显示出来。

在代码中表现为：

main.cpp 中的 `void logPrint(QString str)`

将事件和时间记录至日志中。

## 四、数据结构说明和数据字典

### 1. 数据结构说明

- 1) `enum roomStatus{`  
`empty,checked,booked,checkandbook`

```
};
```

功能：表示房间状态，分别对应：空房，已入住，已预定，入住且预定

```
2) enum roomType{
```

```
    standard,doublebed,largebed,business,vip
```

```
};
```

功能：表示房间类型，分别对应：标准间，双人间，大床间，商务间，VIP 贵宾间 VIP

```
3) query.exec("CREATE TABLE roomInfo ("
```

```
    "id INTEGER PRIMARY KEY AUTOINCREMENT,"
```

```
    "num INT," //房间序号
```

```
    "type VARCHAR," //房间类型
```

```
    "status VARCHAR," //房间状态
```

```
    "checkInTime INT," //入住时间
```

```
    "checkOutTime INT," //离店时间
```

```
    "checkInTimeText NVARCHAR," //入住时间文字  
                                (方便输出)
```

```
    "checkOutTimeText NVARCHAR," //离店时间文字
```

```
    "cost INT)" //押金余额
```

```
);
```

与之对应的链表节点的定义：

```
struct roomInfo{
```

```
    int number; //房间号
```

```
    enum roomType type; //房间类型
```

```
    enum roomStatus status; //房间状态
```

```
    struct datetime checkInTime; //入住时间
```

```
    struct datetime checkOutTime; //离店时间
```

```
    struct guestNode * person; //旅客信息
```

```
};
```

功能：记录房间信息

```
struct roomNode{
```

```
    struct roomNode * prev;
```

```
    struct roomInfo data; //房间信息
```

```
    struct roomNode * next;
```

```
};
```

功能：建立房间表的节点

```
4) query.exec("CREATE TABLE guestInfo ("
```

```
    "id INTEGER PRIMARY KEY AUTOINCREMENT,"
```

```
    "name NVARCHAR," //顾客姓名
```

```
    "gender NVARCHAR," //性别
```

```
    "idNum NVARCHAR," //身份证号
```

```
    "workplace NVARCHAR," //工作地点
```

```
    "phoneNum NVARCHAR," //手机号
```

```

        "consume INT," //客房消费金额
        "checkInTime INT," //入住时间
        "checkOutTime INT," //离店时间
        "checkInTimeText NVARCHAR," //入住时间的文字
        "checkOutTimeText NVARCHAR," //离店时间的文字
        "room INT)" //所住房间号
    );

```

与之对应的链表节点的定义：

```

struct guestInfo{
    QString name; //姓名
    QString id; //身份证号
    bool gender; //性别
    QString workplace; //工作单位 ***
    QString phoneNum; //电话号码
    int consume; //房间内消费
    int age;
    struct roomNode * room; //对应的房间信息
};

```

功能：记录宾客信息

```

struct guestNode{
    struct guestNode * prev;
    struct guestInfo data; //宾客信息
    struct guestNode * next;
};

```

功能：建立宾客信息表的节点

```

5) query.exec("CREATE TABLE bookInfo ("
        "id INTEGER PRIMARY KEY AUTOINCREMENT,"
        "name VARCHAR," //预约顾客的姓名
        "phoneNum VARCHAR," //手机号
        "checkInTime INT," //预计入住时间
        "checkOutTime INT," //预计离店时间
        "checkInTimeText NVARCHAR," //入住时间文字
        "checkOutTimeText NVARCHAR," //离店时间文字
        "room INT)" //房间
    );

```

```

struct bookNode{
    struct bookNode * prev;
    QString name; //预定信息
    QString phoneNum; //手机号
    struct datetime checkInTime; //预计入住时间
    struct datetime checkOutTime; //预计离店时间
    struct roomNode * roomNodePtr; //指向房间表节点的地址
    struct bookNode * next;
};

```

```
};
```

功能：建立预定表的节点

```
6) query.exec("CREATE TABLE log ("
               "id INTEGER PRIMARY KEY AUTOINCREMENT,"
               "time NVARCHAR,"//事件时间
               "string NVARCHAR)" //事件内容
               );
```

功能：向房间表里写入初始的房间信息

我们在 1.0 版本使用链表作为数据结构,在 2.0 版本时将链表转换为数据库,数据库中的数据结构可以与链表的一一对应。

## 2. 数据字典

int layerShow;	//客房信息中当前显示的层数
int price[5];	//房间价格
int itemPrice[5];	//房间内消费的价格
int interval;	//时间推进的时间间隔, 每 n 秒 20 分钟
bool searchmodel;	//是否进行模糊搜索
int now;	//当前时间

## 五、各模块设计说明

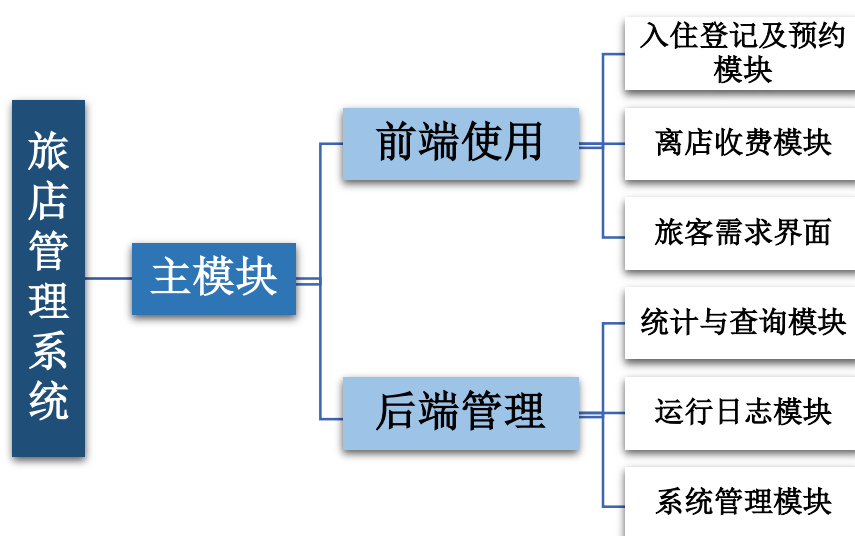


图 13 旅馆管理系统的主要模块划分



## 1. 算法思想

### 1) 入住算法

- a) 登记入住信息包括：个人信息，是否预约过，房间类型，入住天数；
- b) 获取当前时间为入住时间，根据入住天数算出离店时间；
- c) 若信息未输入完整，则提示输入信息不完整；
- d) 输入完整信息后：
  - if 顾客输入已经预约，遍历预约表：
    - {
    - if 预约表中没有该顾客信息，提示没有预约信息请检查是否预约过；
    - else if 预约表中有该顾客信息，判断入住时间与预约时间的关系：
      - {
      - if 入住时间小于预约时间则提示预约房间暂时不能入住，请仔细核对预约信息；
      - else if 入住时间不小于预约时间，则将顾客信息插入顾客信息表中，并成功将预约信息转入登记入住信息中；
      - }
    - }
  - }
  - else if 顾客输入未预约，则判断顾客所选房间类型判断该房间类型的房间是否有空房及是否有已被预约但可以入住的房间。循环检索房间信息表中此类型的房间：
    - {
    - if 有空房则跳出循环，并将空房状态改为已入住，将顾客信息插入顾客信息表中，提示入住成功；
    - else if 没有可用空房，存在已被预约但可入住的房间，跳出循环，将该已被预约房间状态改为入住且被预定，将顾客信息插入顾客信息表中，提示入住成功；
    - else if 没有可以入住的房间，提示所选房间类型房间已满，请选择其他类型房间；
    - }

### 2) 预约算法：

- a) 登记预约信息包括个人信息，预定房间类型，预定房间数，入住时间，入住天数；
- b) 根据入住时间、入住天数遍历所选类型房间：
- c) {
  - ① 优先安排空房，查询到可用空房直接加入可用房间列表；

- ② 其次安排预订过的房间，查询到预订过的房间以后确定该房间此时间段没有被预定，加入可用房间列表；
- ③ 再者安排登记入住过的房间，只要预约日期在离店日期之后就加入可用房间列表；
- ④ 最后安排登记并预约过的房间，要求预约日期在离店日期之后且与其他预约冲突就可加入可用房间列表；

}

- d) 如果找到的房间数不足，返回提示信息；
- e) 如果找到的房间数够，则按照优先顺序将旅客的预约信息登记，并
- f) 更新房间状态，返回预约成功信息，将事件记录到日志中；

### 3) 离店收费算法：

输入要离店的房间号码；

if 输入房间号为错误房间号，则提示输入信息有误；

else if 输入正确的房间号码：

{

    获取该房间的状态：

    {

        if 需要离店的房间为空房，则提示所选房间为空房；

        else if 房间不空，获取客户消费信息，获取客户入住时间，计算出客户入住的天数：

        {

            if 天数不正确提示输入信息有误；

            else if 天数正常：

            {

                if 房间状态为 checked 则将其变为 empty；

                else if 房间状态为 checkandbook 则将其变为 booked；

                同时更新房间信息表中各个状态；

            }

        if 顾客当天入住且当天 12 点前退房则入住天数按半天计；

        else if 顾客当天入住且隔天或之后几天 12 点前退房则入住天数不变；

        else if 顾客 12 点后 18 点前退房则入住天数加半天；

        else if 顾客 18 点后退房则入住天数加一天；

        根据各个房间类型的房价以及物品消费，计算顾客的总消费；

        提示已成功退房，并将事件写入日志文件中；

    }

  }

}

## 2. 各模块的逻辑对应关系

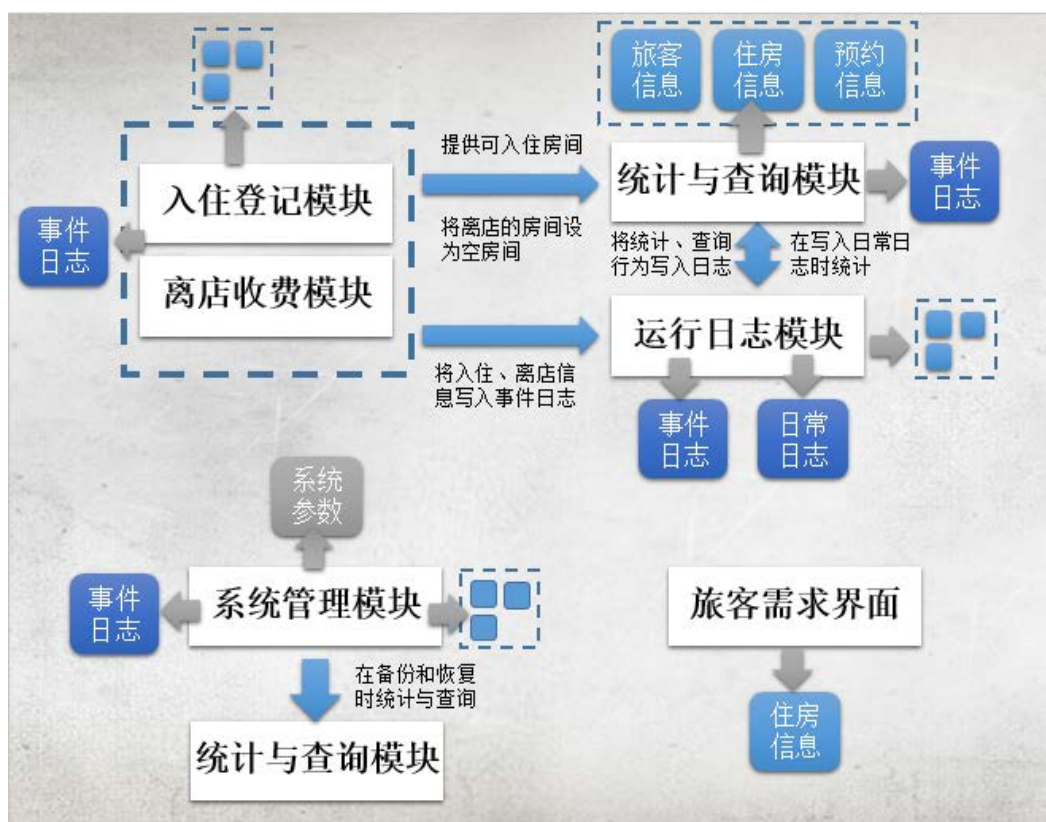


图 12 旅馆管理系统各逻辑关系对应图

软件中入住登记、离店收费和预约都会对房间信息、顾客信息和预约信息的数据库进行操作，而系统管理模块则会对以上三个模块的价格等参数进行调整。统计与查询模块将会对所有数据库进行查询操作从而列出统计信息并提供搜索功能。

我们将预定、取消预定和客房消费放入顾客界面中，面向顾客开放，而其他功能都是面向旅馆前台和旅店管理者开放。

所有模块都将调用日志模块将事件写入日志以便后期管理。

# 六、测试情况说明

对于该旅店管理系统，我们进行了多次的测试。

- 1、 对于一个房间可以进行不同时间的多次预订，运行结果：  
如下图，房间 202、203 在不同的时刻存在多次预订。

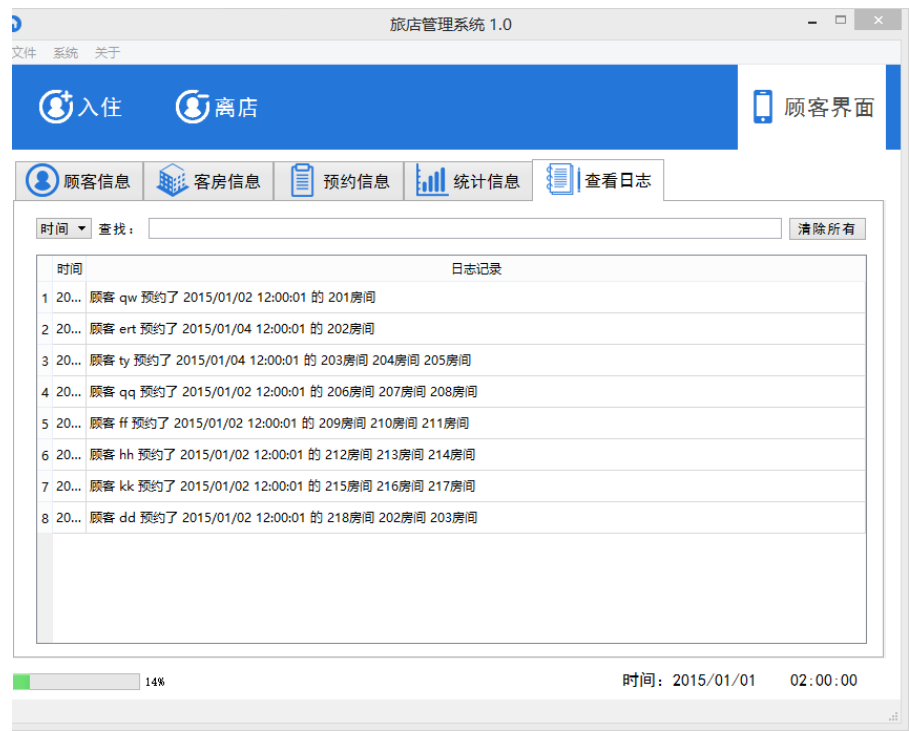


图 13

- 2、 对于一个房间在已有预约的情况下若当前房间还是空房也可进行入住  
运行结果：

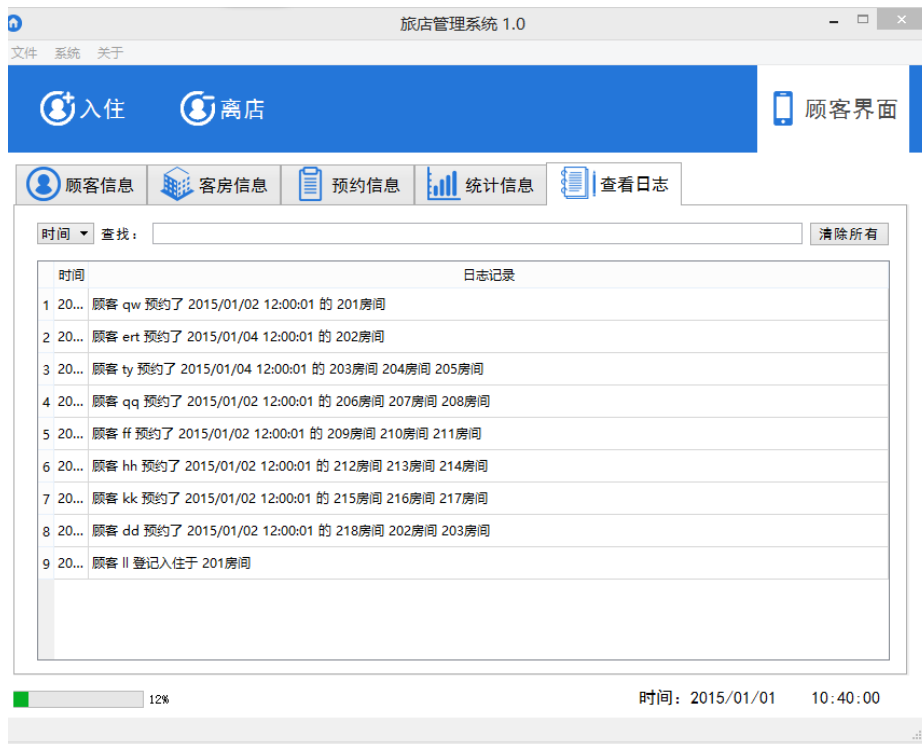


图 14

如图 13 所示，在 201 号房间有预约的情况下，由于当前时间 201 是空房，所以还可入住。

- 3、若预约的旅客在预定时间之前入住，则系统也可以做出正确的提示  
运行结果如下：

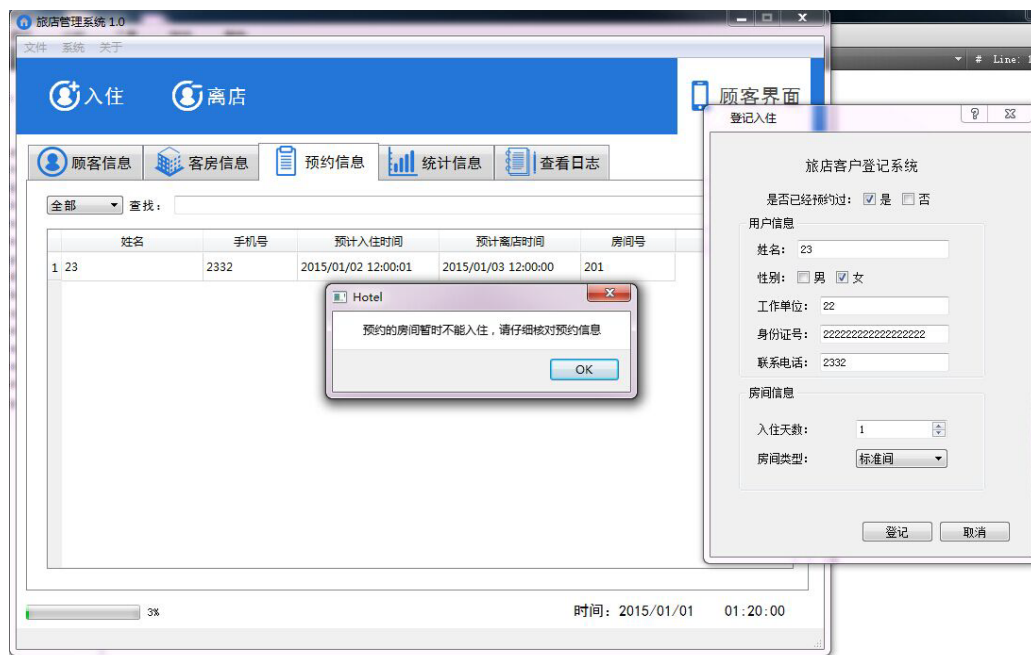


图 15

如图所示，及系统提示预约的房间暂时不能入住，请仔细核对预约信息

## 七、评价和改进意见

本次旅店管理课程设计，小组分工明确，合作顺利，成功实现了一个能完成预约、登记、离店结算、统计查询、日志记录等多功能的旅店管理系统，巩固了C++使用方法，初步掌握了Qt使用手册，使编程能力有了进一步的提高。

本旅馆管理系统界面简洁大方，功能丰富，能够帮助旅店管理者处理日常生活中的旅馆入住、离店、预约等情况。本程序健壮性强，在最终测试中没有出现程序运行错误，且在用户使用时给予使用提示，同时通过正则式来帮助用户规范输入，能够满足日常旅店管理的需要。

当然本旅馆管理系统也存在需要改进之处，目前我们仍未加入数据库加密功能，在安全方面还可以改进；出于房间分布图这一要求限制，此系统不能动态添加新房间，只能在开发时根据需求进行修改。