“C++程序设计与训练”课程大作业

项目报告

项目名称：餐厅服务与管理系统

姓名： 张嘉玮

学号： 2016011528

班级： 自 61 班

日期： 2017.09.20

目　录

1.系统功能设计

1.1 总体功能描述

1.概述

该系统是一个餐厅管理与服务的软件，可以实现现代化餐厅的常用功能。在记录餐厅已有信息的基础上，满足不同用户对象的使用需求，使得餐厅的管理和服务变得方便快捷。系统具有较好的使用稳定性和对用户友好的设计，使得客户有良好的用户体验。

2.项目具体功能

本系统主要有两大类用户群，一类是作为消费者的顾客，另一类是作为服务人员的对象，对不同的用户提供对应的信息和操作服务。

对于消费者，主要功能有：注册并登录，选座，点菜，下单结账，评价菜品，可实时查询餐桌的剩余情况，在消费期间可实时查询做菜的进度，同时满足加水等常规服务，在消费结束以后还可以对服务员的服务质量作评价，对厨师的厨艺提出评价和建议以及对菜品提出评价和建议，这些信息将储存起来，以备后续顾客和其他人员查看。

对于服务人员又分为对系统管理的管理员，服务员，厨师以及餐厅经理。管理员可以对菜单，餐桌进行分类地增加，删除，改正，查询，同时可以对已经注册过的客户进行管理。服务员和餐桌联系在一起，对对应餐桌的顾客进行服务，接受顾客发来的信息，同时查询账单，查看做菜进度。厨师认领消费者传来的订单，进行做菜，同时可以给服务员和消费者提供做菜的进度信息。餐厅经理可以查看了解厨师和服务员的工作量和工作质量，查看消费这对厨师和服务员的评价。

1.2 功能流程描述

1.2.1 消费者流程图

如果已经注册，直接登录，否则，先注册，后登陆

可以选择多个餐桌，也可以实时查询餐桌状态

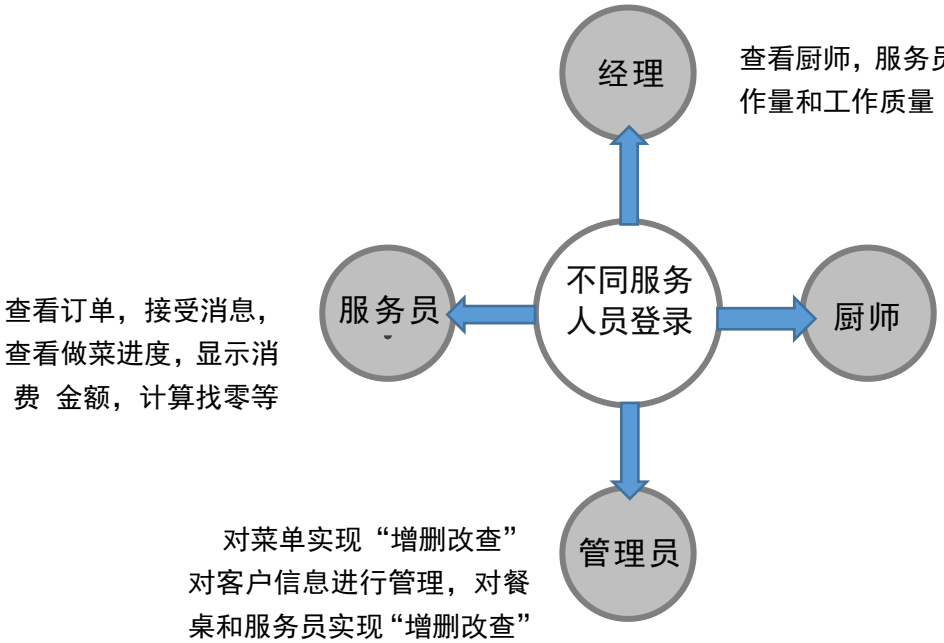
查看本店菜单，可以分类查询，可以选择份数，可以参考评价

查看已经添加的菜品，可以对其进行修改，在确定后提交订单

查看已点菜单，查看进度，呼叫服务员加水，催菜等等

显示消费金额，可对菜品，服务员，厨师等进行评价

1.2.2 服务人员流程图



2.系统结构设计

2.1 主要模块

2.1.1 消费者端

消费者端有四大部分组成，餐桌，菜单，已点菜单，我的清单。

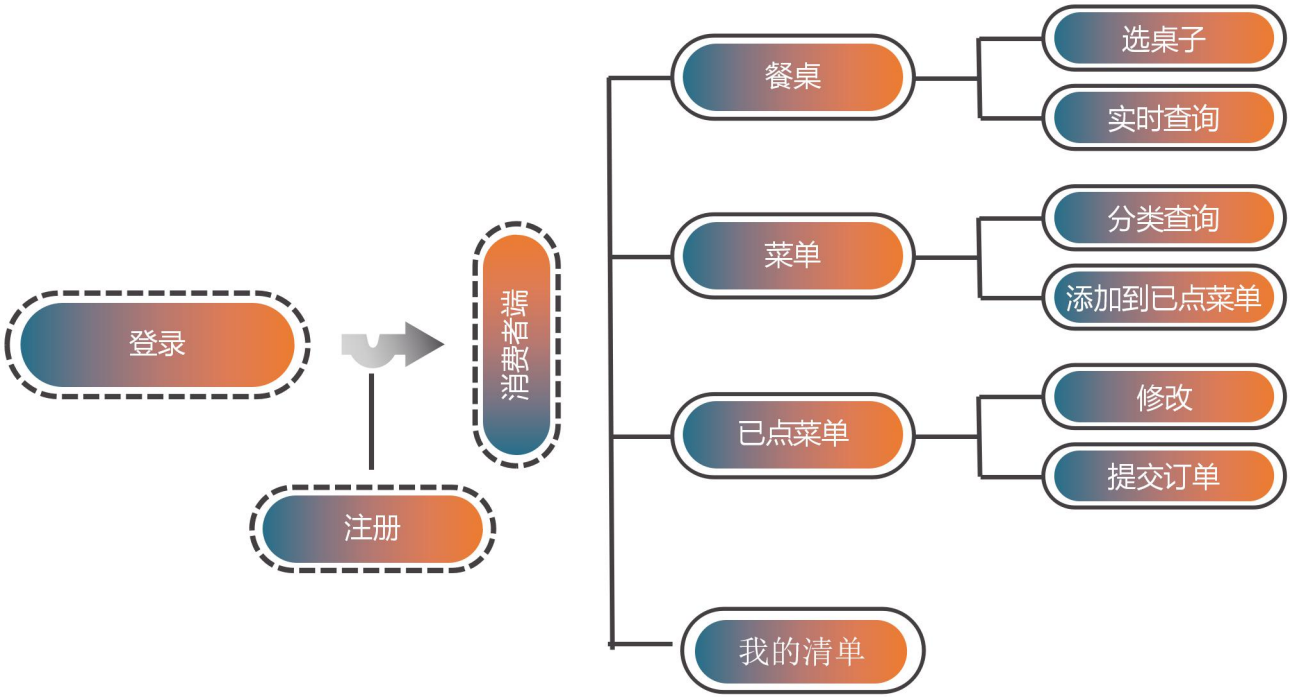
餐桌面可以进行选择餐桌的选择及实时查看餐桌状态。

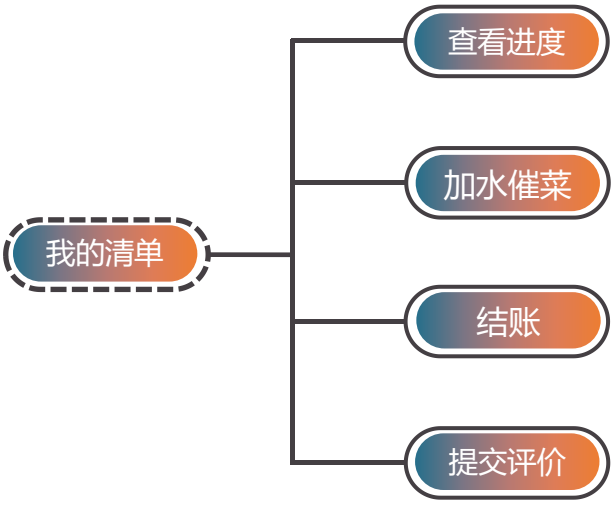
菜单面实现点菜的一系列功能，比如查询，加入到已点菜单里面，操作简单，界面清晰，顾客根据需求，选择想要的菜品。

已点菜单面实现对已点菜单的修改，如果想再添加菜品，可以返回到上一级界面，在确认后提交，生成订单。

我的清单面实现订单的查看，实时查看做菜的进度，呼叫服务员，结账以及评价等一系列功能。

结构图：

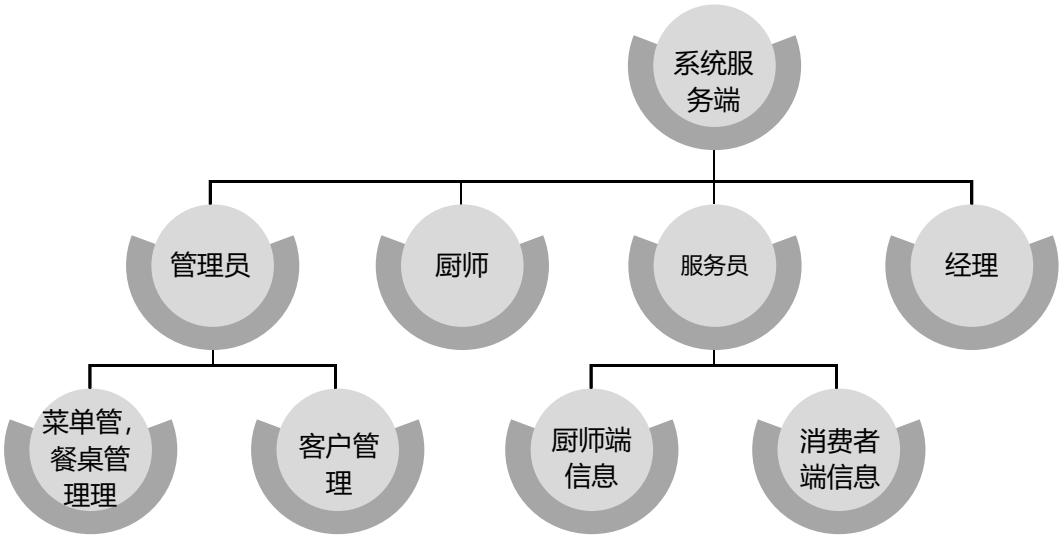




2.1.2 服务端

服务端有四部分组成，有管理员，厨师，服务员以及经理构成。其中管理员可以对数据进行管理，厨师可接受和传输订单信息，服务员接受来自顾客和厨师的信息，经理查看厨师和服务员的工作情况。

结构图：



2.2 类及主要函数

2.2.1 继承自 Dialog 的类：

顾客类	厨师类	服务员类
- 点菜 () :void - 添加到已点菜 () : void - 提交评价 () :void - 结账 () : void - 选桌子 () :void - 提交订单 () : void - 查看进度 () :void	- 接受菜单 () :void - 开始做菜 () : void - 完成做菜 () :void	-接受信息 () :void -得到已点菜单 () :void - 显示做菜进度 () :void .....

收费类（charge）	经理类
- 构建 （）： void	- 显示服务员工作报表 () :void -显示厨师工作报表 () :void

2.2.2 继承自 QWidget 的类

登录类	管理类
- 注册 () :void - 登录（客户，管理员，服务员，厨师经理） () :void	- 修改 () :void - 查询 () :void - 删除 () :void

2.3 主要数据库表

2.3.1 创建菜单表 menu

菜单表用来储存菜单的信息。菜单表的结构:编号（id）,菜名（name）,种类（ctype）,价格（price）,菜品信息（info）。

字段名称	数据类型	字符描述
Id	int	编号
name	varchar	名称
ctype	varchar	类型
price	int	价格
info	varchaar	菜品信息

语句举例:

```
//创建菜单表 menu()
    query.exec(QString("create table mymenu (id int primary key, "
            "name varchar not null,ctype varchar,price int not null,
info varchar)"));
```

2.3.2 创建客户信息表（cguest）

信息表用来存储客户信息。结构：电话号码（gid），密码（gpwd）。

字段名称	数据类型	字符描述
gid	varchar	手机号
gpwd	varchar	密码

2.3.3 创建服务员信息表（waiter）

服务员信息表用来存储服务员信息及对应餐桌。结构：餐桌号（tableid），服务员名称（name），餐桌转台（guest）。

字段名称	数据类型	字符描述
tableid	int	桌号
name	varchar	服务员名称
guest	varcahr	是否有客人

2.3.4 创建密码表（password）

用于储存不同用户类型对应的登录密码。结构：密码（pwd）.

字段名称	数据类型	字符描述
pwd	Varchar	密码

3.系统详细设计与实现

3.1 消费者端界面与类的实现

界面：注册登录->选择餐桌->菜单->已点菜单->我的清单（内含服务界面）。位于后面的界面可以回到前面任何一个界面进行修改，以增加程序的容错性，降低了操作门槛，给予良好的用户体验。

客户类的具体实现：客户类设计和实现一个类，下属四个主要的子界面：

A.在餐桌界面对餐桌进行选择，同时可以实时查看其它餐桌的状态。

转入按钮的槽，根据输入框里的数字，在数据列表里进行查询。查到应的桌号以后，将状态修改为有人，同时更新显示，达到实时查看的目的。

界面：

B.在菜单界面进行点菜，可以选择菜品的份数，同时可以按照类型进行筛选。

点击添加按钮时，槽函数执行添加菜品的操作。首先获取菜的编号，菜名，然后根据输入框里的份数写入对应菜品的份数。在获得这些数据后将信息写入已点菜单。但点击查询时，在其槽函数里执行查询操作。根据类型进行筛选。当点击显示所有菜单时，将菜单遍历一遍，显示在窗口。当点菜结束以后，点击已点菜单，转入已点菜单页面。

界面：

关键代码：

```
//点菜
void guest::on_pushButton_14_clicked()
{
    QString idn,namen;
    int numn;
    double pricen;
    numn=ui->spinBox->value();
    idn=ui->lineEdit_3->text();
    ui->lineEdit_3->clear();
    ui->spinBox->clear();
    QSqlQuery query;
    //得到选中行的数据
    query.exec(QString("select * from mymenu where id='%1'")
               .arg(idn));
    query.next();
    namen=query.value(1).toString();
    qDebug()<<query.value(1).toString();
    pricen=query.value(3).toDouble()*numn;
    qDebug()<<pricen;
    ydmodel->setEditStrategy(QSqlTableModel::OnManualSubmit);
    query.exec("select * from ydcaidan");
    //query.next();
    query.prepare(QString("insert into ydcaidan
(id,name,num,price)values(:id,:name,:num,:price)"));
   query.bindValue(":id",idn);
    query.bindValue(":name",namen);
    query.bindValue(":num",numn);
    query.bindValue(":price",pricen);
    query.exec();
}
```

C.已点菜单界面可以对菜单进行修改，然后提交订单，这时，会将提交订单的信息传到厨师哪里，并且将菜品状态初始化为“没开始”。

当需要删除某菜品时，点击对应的行，然后转入槽函数，使用 qt 的自带函数 removeRow,将其在列表里删除。提交订单的槽函数，利用位置绑定，调用数据库的自带函数，将其写去，同时传到厨师哪里。

关键代码：

```
//提交订单按钮+把已点菜单传到厨师
void guest::on_pushButton_17_clicked()
{
    ui->stackedWidget->setCurrentIndex(3);
```

```
    QString cnamen,cstaten;
    int cidn;
    int cnumn;
    cstaten="没开始";
    QSqlQuery query;
    QSqlQuery cquery;
    query.exec(QString("select * from ydcaidan"));
    while(query.next())
    {
        cidn=query.value(0).toInt();
        qDebug()<<"cidn:";
        qDebug()<<cidn;
        cnamen=query.value(1).toString();
        cnumn=query.value(2).toInt();
        bool q=cquery.exec(QString("select * from cmenu"));
        qDebug()<<q;
        cquery.next();
        bool p=cquery.prepare(QString("insert into cmenu
(cid,cname,cnum,cstate)values(?,?,?,?)"));
        cquery.addBindValue(cidn);
        cquery.addBindValue(cnamen);
        cquery.addBindValue(cnumn);
        cquery.addBindValue(cstaten);
        cquery.exec();


    }
}
```

D.我的清单界面可以查看到订单，桌号，点击服务时可以查看做菜进度，呼叫服务员进行加水催菜等操作。在点击买单时，计算消费金额，同时呼叫服务员进行买单。在这个过程当中，可以对服务员，厨师，每一份菜进行评价。最后退出登录，返回登录界面。

界面当中的桌号由餐桌界面传来，显示对应桌号；刷新可以实现界面数据的更新显示；买单槽函数实现金额的计算（显示框设置为只读，防止修改账单），同时将买单的信息写入，在服务员那里更新便可以显示买单的信息；当点击服务时，转入一个子窗口，里面是已点菜品以及状态，还有催菜加水等功能；除此之外，实现对服务员，厨师以及菜品的评价，

界面：

子界面服务界面：

关键代码：

```
//敲击买单按钮时计算总金额+将信息发给服务员
void guest::on_pushButton_20_clicked()
{
    ui->lineEdit->setEnabled(true);
    ui->lineEdit->setReadOnly(true);
    advice->setTable("ydcaidan");
```

```
    advice->select();
    QSqlQuery query;

   // QSqlQuery query;
    query.exec(QString("select * from ginfo"));
    //query.next();

    query.exec(QString("insert into ginfo(info) values('买单')"));
    query.exec();
    query.exec(QString("select * from ydcaidan"));
    double sum=0;
    while(query.next())
    {
        sum+=query.value(2).toDouble()*query.value(3).toDouble();
        qDebug()<<sum;
    }
    qDebug()<<sum;
    ui->lineEdit->setText(QString::number(sum));
}
```

特别的，在用户退出登录时，会将部分信息进行初始化和归零，比如将该用户的已点菜单清零，把餐桌初始化为无客人状态等等。

3.2 服务端界面与类的实现

界面：管理员，经理，厨师，服务员。从不同的登录入口进入不同的界面，完成不同的功能，同时不同界面的信息可以实现共享。

服务端类的具体实现:服务端设计和实现了四个类,分别对应四类服务人员。

A.管理类（widget）：在这个类里实现对菜单，客户以及餐桌服务员的管理。实现：添加菜品，删除菜品（有增加容错功能的提醒窗口），修改菜品（有具有容错功能的撤销修改），分类查询菜单，按照菜名查询菜单；修改已有客户信息，删除已有客户，查找不同客户。对餐桌实现类操作。

对菜品的管理功能函数大都用 qt 自带的库函数，添加菜品实现在界面里直接输入，同时有提高容错功能的撤销修改函数，调用的库函数比如：commit(),rollback(),revertAll(),removeRow(),submitAll(),setDate(),insertRow() 等等。查询函数与上面的菜单查询函数相似，分别实现两种不同的查询路径，一种是根据菜名，另一种是根据菜品的类型查询（实现查询以后现实全部的

功能）；同时实现桌面之间转换的一些槽函数。

界面：

关键代码举例：

```
//删除菜品
void Widget::on_pushButton_7_clicked()
{
    // 获取选中的行
        int curRow = ui->tableView->currentIndex().row();
        model->removeRow(curRow);
        int ok = QMessageBox::warning(this,tr("删除当前行!"),
                    tr("你确定删除当前行吗？"),QMessageBox::Yes,
QMessageBox::No);
        if(ok == QMessageBox::No)
        { // 如果不删除，则撤销
            model->revertAll();
        } else { // 否则提交，在数据库中删除该行
            model->submitAll();
        }
```

}

B.厨师类（chef）：接受来自客户的订单，有开始和完成个槽函数，分别修改菜品的状态，当对应客户退出登录时，列表初始化为清空状态。

界面：

关键代码：

```
//完成做菜
void chef::on_pushButton_2_clicked()
{
    int i=ui->spinBox->value();
    QSqlQuery query;
    query.exec(QString("select * from cmenu where cid='%1'").arg(i));
    query.next();
    query.exec(QString("update cmenu set cstate = '完成'where
cid=('%1')").arg(i));
    //同时更新菜单
    cmodel=new QSqlTableModel(this);
    cmodel->setTable("cmenu");
    cmodel->select();
    cmodel->setEditStrategy(QSqlTableModel::OnManualSubmit);
    //设置编辑策
    ui->tableView->setModel(cmodel);
}
```

C.服务员类（waiter）：实现两个下属窗口，分别是已点菜单和做菜进度；同时实现来自客户的信息的显示（包括催菜，加水，买单等等），服务员界面实现更新的槽函数，都界面上的数据进行更新；服务员收到信息的槽函数将客户发过来的信息 delete。同时服务员界面上显示已点菜单的总金额，当用户需要结账时，在收到栏里输入金额，点击结算按钮，计算出找零金额，既可以显示找零金额。为了适应电子支付的需求，本系统额外开发了微信和支付宝两个转款的窗口。

界面：

收费界面：

关键代码：

```
//计算找零
void waiter::on_pushButton_6_clicked()
{
    double a,b,c;
    a=ui->lineEdit->text().toDouble();
    b=ui->lineEdit_2->text().toDouble();
    c=b-a;
    ui->lineEdit_3->setText(QString::number(c));
}
```

D.经理类（manager）：查看服务员和厨师的工作情况，信息分别有做的菜的份数，客户评价等等。

界面：

3.3 登录界面类的设计与实现

界面：包含两个下属子窗口，顾客和服务人员。顾客包含注册和登录两个函数；服务人员包含管理员，经理，厨师，服务员四个登录函数。

界面：

登录类（login）的实现：在客户点击登录时，先查询这个手机号是否已经注册，若没有注册，提示先注册；若已经注册，查询密码的准确性。只有手机号与密码对应成功时，才可以实现登录。在服务人员登录时查询对应密码，若密码或者手机号错误提示重新输入，直到数据对应成功。

关键代码：

```
//顾客登录
void login::on_pushButton_3_clicked()
{
    QString s;
    s=ui->lineEdit->text();
    if(ui->lineEdit->text().isEmpty()){
        QMessageBox::information(this,tr("请输入手机号"),
                                 tr("请先输入手机号再登录！如果没有注
册，请先注册！"),QMessageBox::Ok);
        ui->lineEdit->setFocus();
    }
    else{
        if(s.length()!=11){QMessageBox::warning(this,tr("提示"),tr("
请输入已经注册的手机号登陆！"),QMessageBox::Ok);
            ui->lineEdit->clear();
            ui->lineEdit->setFocus();}
        else{
            QString s,p;
```

```
            s=ui->lineEdit->text();
            p=ui->lineEdit_5->text();
            QSqlQuery query;
            query.exec(QString("select * from cguest where gid =
('%1')").arg(s));
            query.next();

            if(query.value(1)==p){
    guest*g=new guest;
    g->show();
   this->close();
            }
            else{
                QMessageBox::warning(this,tr("提示"),tr("密码或账号呢
错误！"),QMessageBox::Ok);
            }
        }
    }
}
```

4. 项目总结

4.1 遇到的问题以及解决方法

问题 1：由于知识的积累有限，导致对开发环境不熟悉，特别对界面的开发一无所知，导致大作业不知从何下手。

解决：为了解决入门的问题，在开始的两三天里，我在网上查询了好多有关 qt 开发环境的资料，但由于网上太零碎，不够系统，不适合入门；于是我找了一本《Qt+Creator 快速入门》，这本书比较系统的介绍了 qt 的一些基础知识，这才使得我对大作业有了初步的想法。从 hello world 到设计界面，从不了解槽与信号到编写按钮的操作，在几天时间里，算是对 qt 有了初步的了解，也了解了它的一些自带函数。这个从零到一的过程算是遇到的最大的困难，花费了五六天的时间。

问题 2：数据库的构造与数据表的设计

解决：大作业另一个难点是对数据库没有概念，不知道数据库是什么，干什么用，怎么用等等概念问题。为了解决这个问题，我找了一本《Qt 及 Qt+Quick 开发实战精解》，这里面有对数据库的讲解，在这本书里我才慢慢建立起了数据库的概

念，但虽然了解了 qt,但对 SQL 语言不是很熟悉，导致在数据库的构造以及调用时出现了许许多多 bug,导致大部分时间花在了调试上。

4.2 设计,开发和调试中的难点

开发最大的难点还是从零到一的过程，概念的建立。由于 qt 开发环境老师没有讲解，全部内容都得靠自学，对好多知识点和概念也是现学现用，导致对问题的考虑不是很全面，使得先做的部分在做到后面时才发现漏洞，修改和补全花费了好多时间。设计上最大的困难时不同界面和类之间的关系，不同模块之间的关系。

4.3 设计，开发和调试中的亮点

我在设计时，特别注重作为一个面向客户的实用系统，其与客户的信息交互作为一个重点，其中包括系统的容错能力，界面的友好等等，使得该系统能提供较高使用性，能够很好的实现人机交互。另外，本系统调用了部分库函数，使得程序相对简单，易于维护。

4.4 心得体会

由于暑假期间有其他事情，在小学期开课之前没有做任何预习工作，导致自己在起步上便落后于其他同学，在小学期的前两周并没有时间准备大作业，时间基本上花费在了 c++的学习上，所以大作业基本上是在课程结束以后开始的。

c++大作业算是上大学的第一份大作业，由于没有任何经验，无论是方法还是知识点，都是从零起步，困难时方方面面的。但也正是因为这个原因，在短短的四周时间里，收获也颇丰，同时也提高了自己的编程水平，把所学的知识用到了实际的项目当中。提高了自己的自学能力，以及获得知识的方法和途径的能力，对信息的选择。特别地，以前的学习是学什么用什么，在做大作业时，是需要用什么，学什么，这对自己的获得知识能力有了显著的提高。

其次，另一个较大的收获是对一个系统开发的步骤。以前的作业都是在短时间能完成的，所以大部分不需要提前的设计和计划，但这次大作业教会我在完成一个较大的任务时，没有顶层的计划与设计，是完不成的。

总之，这次大作业是一次难忘的经历，使我的能力得到了显著的提高。在今后做大作业时，这次经历将发挥重要作用。