**“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：**马天云

**学号：**2016011417

**班级：** 自62

**日期：**2017.9.16

目 录

[目 录 2](#_Toc891237713)

[1系统功能设计 3](#_Toc947534933)

[1.1总体功能描述 3](#_Toc380989104)

[1.2功能流程描述 5](#_Toc1788710343)

[1.3初始内置信息说明 10](#_Toc59908557)

[2系统结构设计 10](#_Toc1961236868)

[3系统详细设计 14](#_Toc1456324806)

[3.1 类结构设计 14](#_Toc2094606078)

[3.2 界面结构设计 15](#_Toc1189663935)

[3.3关键设计思路 16](#_Toc467606891)

[4项目总结 18](#_Toc1119124413)

[5相关问题的说明 19](#_Toc1764817119)

# 1系统功能设计

YRMS餐厅管理系统旨在实现餐厅管理系统化,使顾客用餐更加方便快捷,同时,实现了,服务员-厨师-顾客实时信息沟通,解决服务员-顾客-厨师的在现实餐厅中信息交流出错的问题,而且也实现了经理对员工的工作情况查看,以及系统管理员对餐厅信息的管理,从而使得餐厅管理一体化,更加高效的运行。基于数据库的实现,使得整个餐厅的数据可以有效存储下来。

本系统模拟设想了未来的一种用餐模式，即在每一个餐桌上会有一个显示屏，并配备一个机器人服务员，二者共用一个应用程序，有一名顾客入座后，相应的机器人服务员会启动并对该桌顾客服务，同时，所有餐桌启动的程序在厨房都会各有一个显示屏，每个餐桌可登入一名厨师，厨师会接收该桌顾客的订单，与该桌顾客进行信息交流，而该显示屏上登录的厨师也可以与相应的机器人服务员进行信息交流。

## 1.1总体功能描述

具体实现的功能罗列如下:

* 登录界面

- 不同身份用户登录

- 注册成为顾客

* 系统管理员界面

- 菜单管理(增加、删除、修改)

- 用户信息查看(修改员工信息、增加员工[经理,服务生,厨师])

**此系统不实现员工删除功能，用意将在下文描述**

* 经理界面

- 查看顾客对所有服务员和厨师的工作质量，工作效率的评价

- 查看 顾客的评语

- 查看服务员的历史服务次数和厨师的历史做菜数量

* 顾客界面

- 选择座位入座

- 查看菜单

- 点菜

- 查看已点菜

- 加菜

- 结账

- 呼叫服务员

- 查看做菜进度

- 查看餐桌剩余情况

- 评价厨师和服务员

* 服务生界面

- 实时接收顾客请求

- 实时接收厨师上菜请求

- 实时统计服务次数

* 厨师界面

- 实时接收顾客order

- 确认做菜

- 确认菜品已完成

- 实时统计做菜数

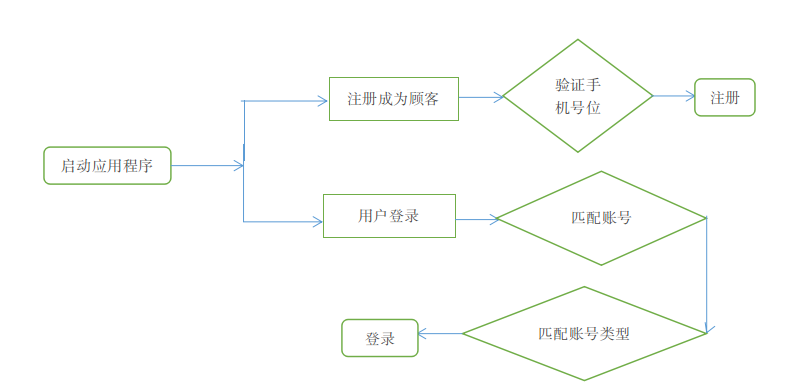
**本系统最主要的功能是实现了多用户系统，即在一次程序启动后，可以登录进入一名服务生，一名厨师，一名顾客，实现顾客，厨师，服务生实时信息沟通。**

## 1.2功能流程描述

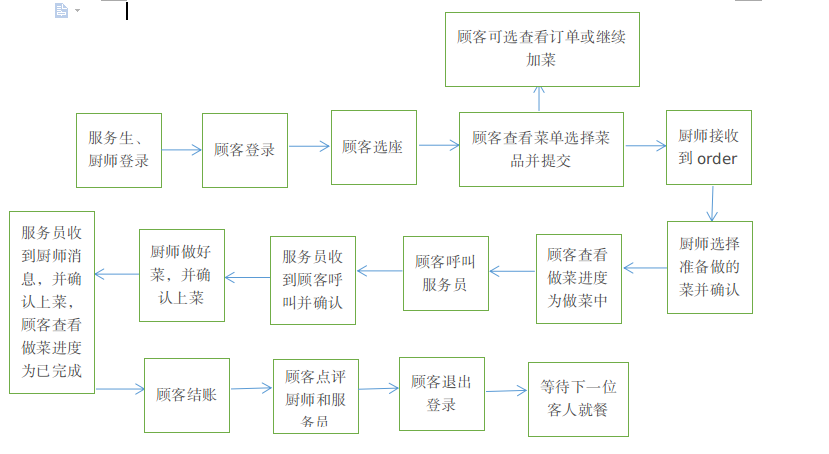
登录主界面如图:



* 登录界面操作流程



* 模拟标准用餐过程



**以上为一次标准的用餐过程,可以体现在本次用餐过程中的所有的实现的功能,当然,在实际使用中不一定会全部体现出。**

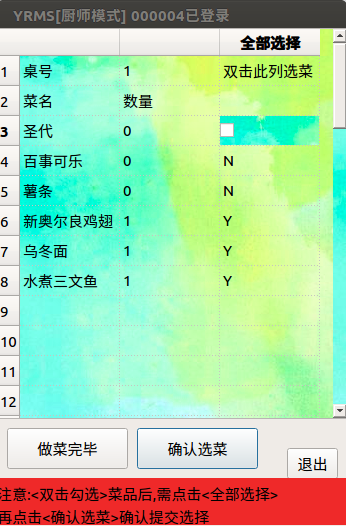
* **服务员设计**

服务生可以实时接收所服务桌号的顾客发送过来的请求,和厨师已做好菜需要上菜的请求,并确认消息.

**此图展示同时有厨师和顾客发送消息**



* **厨师设计**



厨师界面实时刷新显示顾客的已点菜品,并勾选要做的菜品,点击确认选菜可认领已选菜品,做菜完毕后点击做菜完毕即可通知服务员端菜

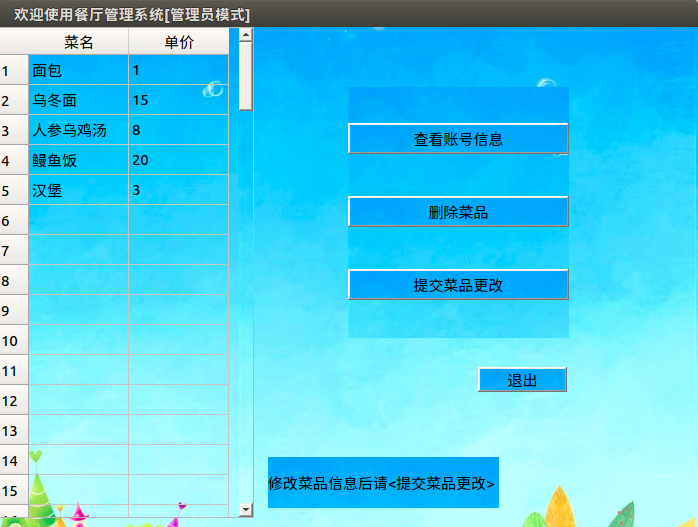
* **餐厅经理设计**

餐厅经理登录后可选择查询服务员评价或查询厨师评价,点击后会弹出相应的员工列表,选择一个即可查看本人的工作质量折线图,工作效率折线图,以及历史评价,历史服务(做菜)总数

**如图片展示的界面，厨师可以双击order第三列，会弹出勾选框，选择后会显示Y表示选中，未选择则显示N，未双击则无反应最后提交后，即确认一次做菜任务。**

* **系统管理员设计**

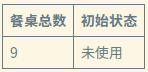
**系统管理员界面如图:**



可以对在菜单中直接进行菜品的添加修改,若删除则选中菜品点击删除菜品,但每次操作完都要提交菜品更改以保存,可点击查看账号信息以进行对用户账号的管理,其中顾客信息是不可修改的,员工信息可以直接进行修改或选择添加员工信息,但每次操作后都要提交修改以保存,此次不设计删除员工信息的功能,旨在保存历史员工信息,方面餐厅其他的事务管理。

## 1.3初始内置信息说明





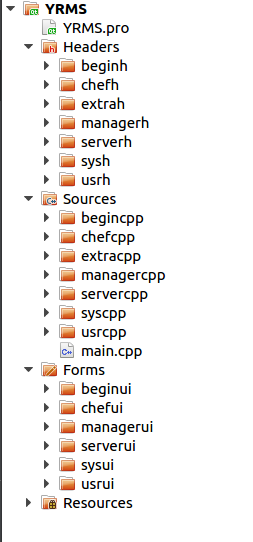




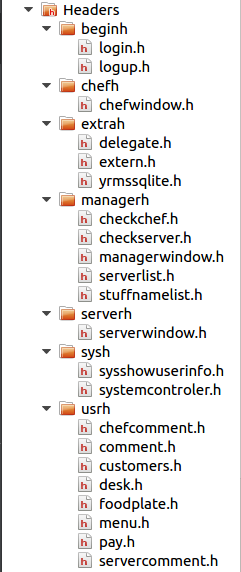
[点击查看数据库结构具体说明](#sqlitle3数据库的使用)

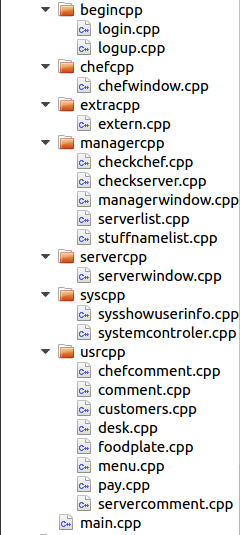
# 2系统结构设计

代码结构图如下:



**其中,每种用户界面需要的cpp,h,ui文件都存放在相应的文件夹之中,extrah头文件夹和extracpp文件夹存放了系统所需要的额外编写的头文件和cpp文件**





**在extrah和extracpp文件夹中,extern文件是我声明并定义的一些程序中使用到的全局变量,delegate文件中定义在使用model/view时的一些委托类,yrmssqliter文件中定义一些数据库操作的函数。在使用到的文件中都会引用相应的头文件,其他每一个界面文件都包含一个类,且继承了Qdialog类,用作界面显示以及功能实现**

# 3系统详细设计

## 3.1 类结构设计

除extrah外中的头文件外,每个头文件都是一个Qdialog类,以实现相应的界面的显示,同时在相应的类的私有成员变量中,声明了要跳转的界面的类的指针,在cpp文件中会实现存储空间的分配。代码举例如下:



在delegate.h文件中声明了model/view模式中需要的spinbox委托类,readonly委托类,combobox委托类,checkbox委托类,代码举例如下:





**其他三类设计类似,在此不再累赘**

## 3.2 界面结构设计

系统界面设计采用以ui界面设计为主,代码信号槽实现为辅,界面的跳转都会有相应的部件操作,同时在部分操作中有QMessagebox类窗口提醒。

## 3.3关键设计思路

本系统设计的亮点有如下：

1.extern全局变量的使用



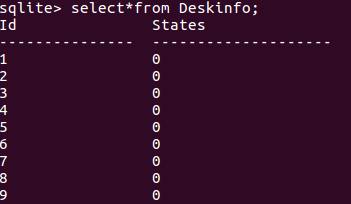
1. 

通过声明extern类来保持在**一次程序运行中**需要的状态信息,如本次用餐的顾客信息,经理查看的员工账号等,方便了在程序设计中的数据保持与交流。

2.sqlitle3数据库的使用

usrinfo 中Users表存放用户信息,serverlist中存放所有服务生账号,cheflist中存放所有厨师账号,每一位厨师有一个以"自己账号"命名的表,存放历史做菜数,有一个以"自己账号+c"命名的table存放顾客的所有评价,每一位服务生也有一个以"自己名字命名的表,存放历史服务总数,另一个以"自己账号+c"的表存放顾客对本人的所有评价

dishes中Deshinfo存放餐桌编号以及餐桌状态,如下图,states=0表示未使用

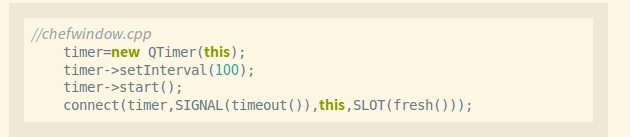


Dishinfo表存储所有菜品的名称和单价

[[返回查看表的结构]](#_1.3初始内置信息说明)

3.Timer库的使用

使用Timer类库,可以实现对界面的实时刷新,实现顾客-服务生-初始实时通讯,代码举例如下



这段代码可以实现界面每隔0.1秒刷新一次。

通过以上方法,实现了实时的三角信息交流。

# 4项目总结

自己感觉qt的上手还是需要一个比较长的过程,需要在实践中慢慢理解,比如窗口的类实现,不同窗口类之间的区别,不同model类的继承关系等,在熟悉后做做起来就会很快,同时熟练掌握几种库类的使用,在很多地方就可以多次利用了。本次大作业收获很多,很好地巩固了c++的基础知识,对面向对象设计有了更深的体会,对一个软件项目的实施也有了基本的认识,我认为这次作业最大的收获,就是极大的提高了自学的能力,大作业的完成很大一部分都是边自学边实践,在实践中熟悉,并一次次修改大作业中不满意的地方,直到做到自己认为比较完美。我希望这只是一个开始,以后会自学更多的东西,并多锻炼自己文字表达的能力,以更好的提高自己的技术。

# 5相关问题的说明

由于我是使用Ubuntu16.04的系统，在软件安装过程中比较方便，没有遇到很多在window中出现的问题，整个数据库操作也方便很多，节省了不少时间。