# 实验二 总体设计

1. **实验目的**
2. 通过完成一个完整的软件项目，掌握软件项目管理的方法和技术。针对小组项目进行软件总体设计，并撰写相应的软件总体设计说明书。
3. 通过总体分析掌握软件项目的沟通管理等领域知识很方法。
4. 回顾结构化或面向对象的设计方法。
5. **实验内容与步骤**

## 2.1需求规定

1、操作简单，对于大多数网络用户都可以轻松地实现在网上订餐的功能，界面清晰，简单易懂，易于操作。

2、具有类似网站的管理的功能，增加，删除，更改，填写订餐客户信息。

3、系统运行快速稳定，高效。

4、在结构上具有很好的可扩展性，一边与将来在功能上的增加。

## 2.2运行环境

操作系统: Windows 10及其以上版本

数据库:MySQL 8.0.5及其以上版本

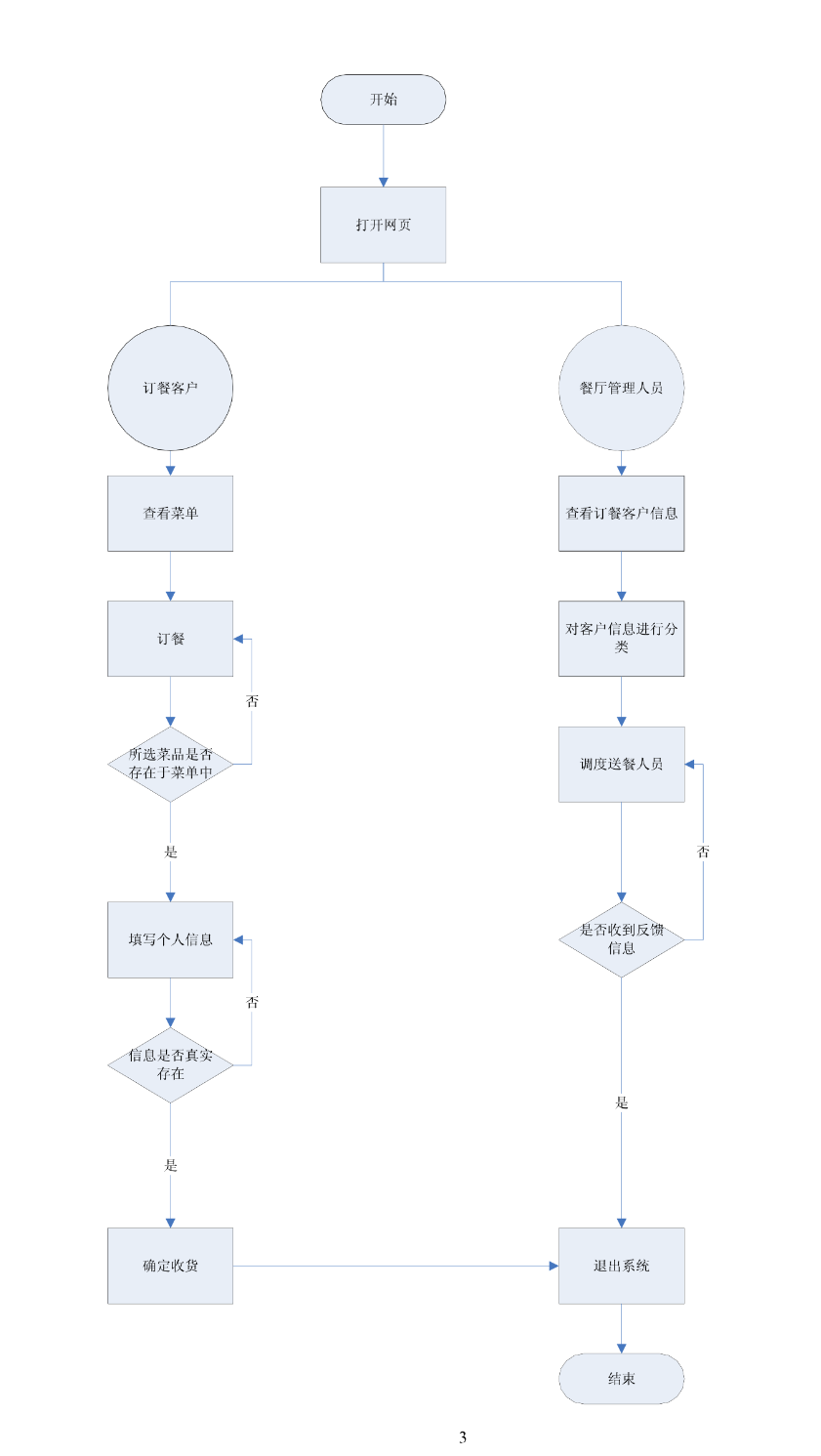
## 2.3基本设计概念和处理流程

说明本系统的基本设计概念和处理流程，尽量使用图表的形式。

本系统的基本设计概念:订餐客户通过浏览网页上的菜品信息,通过填写信息进行订餐。餐厅通过收集订餐信息进行送餐，并对客户信息进行管理。

处理流程:餐厅将菜品的基本信息，包括图片，价格，配料发布到网上订餐主页上->订餐客户浏览网页->打开订餐菜单填写信息->验证所填写信息是否正确->提示客户订餐成功->餐厅整理订餐信息->送餐。

其流程图如下所示:



# 3 接口设计

## 3.1用户接口

窗口化，索引式，在操作时相应热键

## 3.2外部接口

硬件接口：无；

软件接口：通过桥接器与MYsql8.0数据库连接；

## 3.3内部接口

通过面向对象语言设计类，采用MVC响应式架构，在前端窗口页面是实现对类的属性和功能的调用

# 4运行设计

## 4.1运行模块组合

主要是以窗口为模块，一个窗口完成一项功能，通过每一个窗口最上面的索引实现模块间不同功能的连接和组合。各模块之间相互独立。

## 4.2运行控制

不限制用户的输入，对于每次输入，只要符合要求，可继续执行，若超出规定范围则会出现对话框提示用户信息错误

## 4.3运行时间

每次相应时的上限应该控制在1s内

# 5系统数据结构设计

5.1逻辑结构设计要点

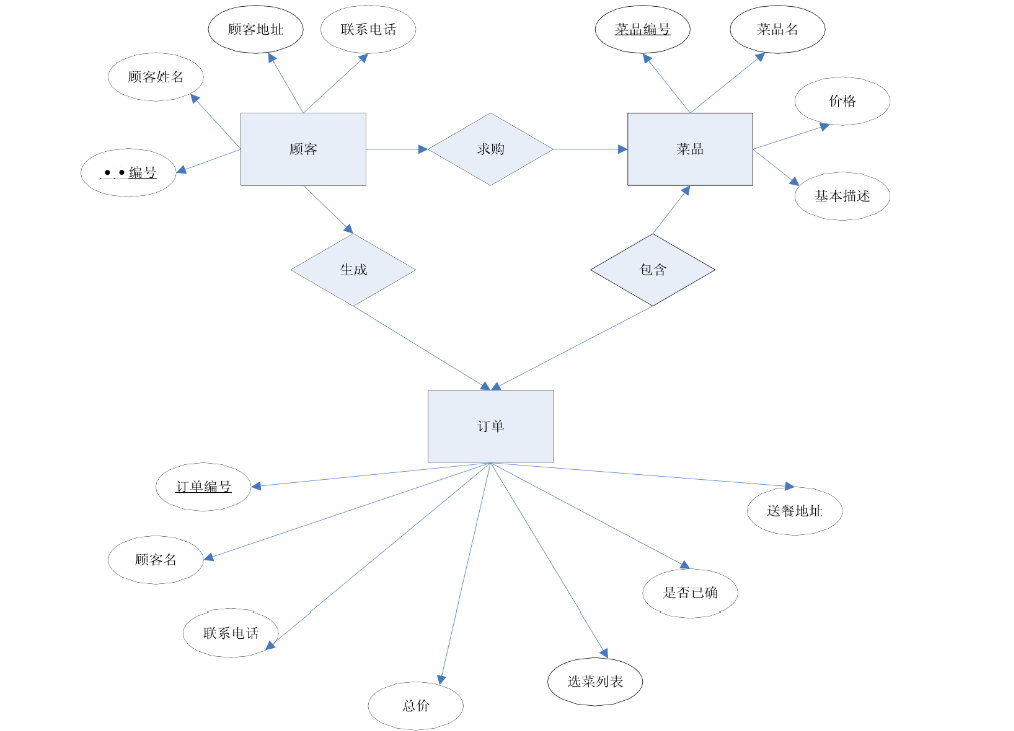
给出本系统内所使用的每个数据结构的名称、标识符以及它们之中每个数据项、记录、文卷和系的标识、定义、长度及它们之间的层次的或表格的相互关系。

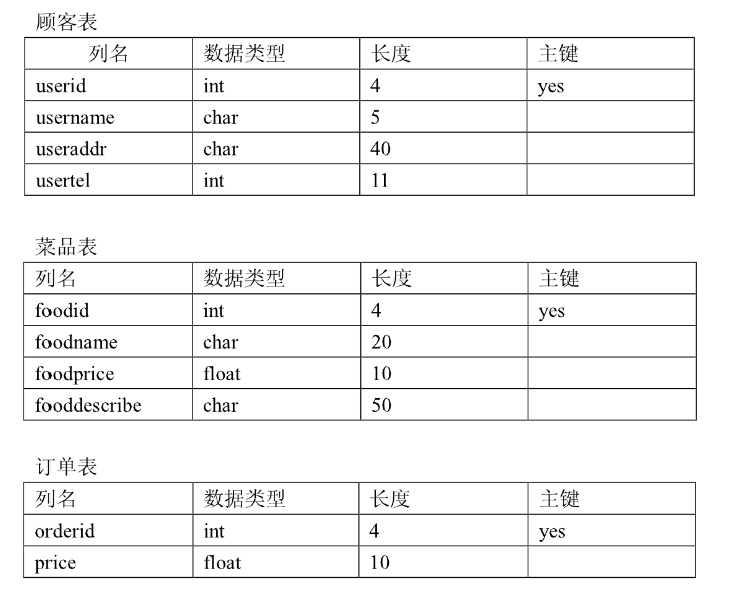
1、菜品表FOODLIST (菜品名，菜品编号，菜品价格，菜品介绍)

2、订单表ORDERLIST (订单编号，菜品数量，菜品编号，客户编号)

3、客户表USERTLIST (客户编号，客户姓名，联系电话，地址)

其ER图如下所示:





## 5.2物理结构设计要点

数据结构为关系型数据库，所以，在程序中可以通过标准的SQL语句与数据结构进行交互，交互过程中采用通用的数据库访问接口。

## 5.3数据结构与程序的关系

系统的数据结构由标准数据库语言SQL生成。

例如创建系统用户表格的程序用SQL表示就是:

CREATE TABLE[user].[user\_Info](

[userid][char](10)COLLATE Chinese\_PRC\_CI\_AS NOT NULL,

[usertel][char](10)COLLATE Chinese\_PRC\_CI\_AS NULL,

[useraddr][char](10)COLLATE Chinese\_PRC\_CI\_AS NULL) ON [PRIMARY]

# 6.系统出错处理设计

## 6.1出错信息

**出错信息表**

|  |  |
| --- | --- |
| 出错信息 | 出错时，系统输出信息的形式、含义及处理方法。 |
| 乱码 | 系统输出信息的形式为乱码，解决办法:对提交的汉字信息从新进行编码。 |
| 注册 | 输出信息，注册不成功，解决办法:输入符合软件规定的会员名和密码。 |
| 登录 | 输出信息登陆不成立，解决办法:输入已经注册的文件名和密码。 |
| 菜品添加 | 输出信息，没有添加成功，解决办法:仔细检查插入语句是否出错，或者是是否有数据库连接代码。 |

## 6.2补救措施

将上面的给出的出错信息进行分析，得到主要的错误可能有:

(1）数据库连接错误:这类错误主要是数据库设置不正确引起的，我们只要取消本次操作，提醒维护人员自己检查数据库问题即可。

(2）输入错误:主要是用户输入不规范造成的，我们在尽量减少用户出错的条件的情况下对用户进行提醒，然后再次操作。

(3）乱码:主要发生在提交汉字信息的情况下，这是我们只需要对提交的汉字从新进行编码即可。

(4）其他操作错误:对于用户的不正当操作，有可能是程序发生错误。我们主要是种植操作，并提醒用户种植的原因和操作规范。

(5）其他不可预知的错误:程序也会有一些我们无法预知或没有考虑完全的错误，我们对此不可能做出完全的异常处理，为了保证数据的安全，要经常对数据库进行备份。然后犯规错误信息，以逐步完善程序。

## 6.3系统维护设计

软件的维护主要包括，数据库的维护和软件功能的维护。

1. 对于数据库[[1]](#endnote-0)的维护，主要是经常备份数据库的内容，以防止数据库内容的丢失。
2. 对于软件功能方面的维护由于采用了模块化得设计方法，每个模块之间相互独立型较强，这样对软件的维护带来了很大的方便，对于单独功能的修改只需修改相应的模块即可。而对于功能的添加，只要增加相应的模块即可。我们还会对软件进行定期的维护和修改，以使其更加安全，更加方便和快捷。
3. **实验结论**

1. [↑](#endnote-ref-0)