

## 目录

1.	项目目标.....	1
2.	需求分析.....	1
2.1.	涉众分析 .....	1
2.2.	用况分析 .....	1
3.	对象模型.....	6
3.1.	类图分析 .....	6
3.2.	主要功能建模.....	9
4.	OOD 模型 .....	11
4.1.	问题域部分 .....	11
4.2.	控制驱动部分 .....	11
4.3.	数据接口部分 .....	12
4.4.	人机交互部分 .....	14
5.	总结 .....	15

## 麦得劳汉堡套餐系统开发文档

安东 张爽 张俊超

### 1. 项目目标

- (1) 开发一个个性化汉堡定制系统，可以让消费者方便地自主选择食材，定制自己喜欢的汉堡。
- (2) 给消费者推荐汉堡套餐以及营养食材。

### 2. 需求分析

#### 2.1. 涉众分析

涉众角色：

- (1) 接待员：为消费者取餐、传餐
- (2) 主厨：制作汉堡；给消费者推荐汉堡。
- (3) 营养设计师：设计变种食材；修改食材营养成分信息。
- (4) 消费者：定制汉堡，下单消费。

注：实际上应有更多涉众，比如管理后台数据的管理员。本设计甲方并未提出设计该种角色的需求，因此本设计着重进行功能设计，没有考虑管理员类的角色。

#### 2.2. 用况分析

##### 1) 显示可选食材

显示某中食材类型下（面包、肉饼、蔬菜、芝士、酱料或其他）的所有可选食材列表，包括食材的图片、名称和价格等信息。

##### 显示可选食材

选定食材大类类型；

查询数据库；

显示该类型下所有食材的图片、名称和价格。

##### 2) 显示食材信息

显示某种具体食材的详细信息。

##### 显示食材信息

选定某种食材；

查询数据库；

显示该食材的所有详细信息（图片、名称、价格、营养成分）。

### 3) 消费者点餐—定制汉堡

消费者定制汉堡，并确认订单。

#### 消费者点餐

消费者点餐开始；

生成订单并初始化，输出提示信息；

消费者选择定制汉堡或者选择推荐；

if 选择定制汉堡 then

call 定制汉堡；

elif 选择推荐 then

call 消费者选择推荐；

end if；

显示汉堡详情和总价格；

提示消费者确认订单；

if 消费者确认订单 then

提示点餐完成，显示订单号；

通知主厨确认订单；

end if。

### 4) 定制汉堡

消费者选择食材定制汉堡。

#### 定制汉堡

定制汉堡开始；

打开定制页面，完成初始化

显示可选面包；

消费者选择面包；

记录所选面包（面包类型、价格、营养成分等）；  
显示可选肉饼（包括变种肉饼）；  
消费者选择肉饼和数量；  
记录所选肉饼和数量；  
显示可选蔬菜（包括变种蔬菜）；  
消费者选择蔬菜和数量；  
记录所选蔬菜和数量；  
显示可选芝士（包括变种芝士）；  
消费者选择芝士和数量；  
记录所选芝士和数量；  
显示可选酱料；显示添加剂选项；  
消费者选择酱料；  
消费者选择添加剂；  
记录所选酱料和添加剂；  
显示可选其他食材；  
消费者选择其他食材和数量；  
记录所选其他食材和数量；  
提示定制结束。

#### 5) 消费者选择主厨推荐

##### 选择主厨推荐

选择开始；  
显示所有可选的主厨推荐；  
消费者选择并确认；  
记录消费者的选择并返回。

#### 6) 取消订单

在订单可被取消的条件下，接受消费者取消订单的请求，并为消费者取消订单。

### 取消订单

消费者申请取消订单；

    查询订单状态；

    if 订单可被取消 then

        系统取消订单；

        更新订单状态；

        提示消费者取消完成；

    else

        提示消费者订单不可被取消；

    end if。

### 7) 显示订单详情

给消费者展示当前订单的详细信息：汉堡定制食材、订单价格、订单状态等信息。

### 显示订单详情

指定订单号（消费者点击查询）；

    查询数据库；

    显示定制详情、总价格、订单状态和时间。

### 8) 主厨确认订单

主厨确认新订单，并把订单状态更新为“已有主厨确认”。

### 主厨确认订单

主厨点击确认订单；

    更新订单状态。

### 9) 通知接待员取餐

汉堡制作完成后，打印订单详情和价格，并通知接待员取餐送餐给消费者。

### 通知接待员取餐

打印订单详情和价格；

给接待员发出通知；  
更新订单状态。

#### 10) 添加食材

##### **添加食材**

添加开始；

进入添加页食材面并初始化；

编辑食材类型、名称、营养参数；

编辑食材价格；

编辑食材图片；

提交；

将食材信息添加进数据库；

提示添加成功。

#### 11) 编辑食材属性

##### **编辑食材属性**

编辑开始；

显示食材详情，提供编辑接口；

编辑食材各种属性（营养参数、图片等）；

确认并提交；

将食材数据返回给数据库；

提示修改成功。

用况图如图 1 所示。

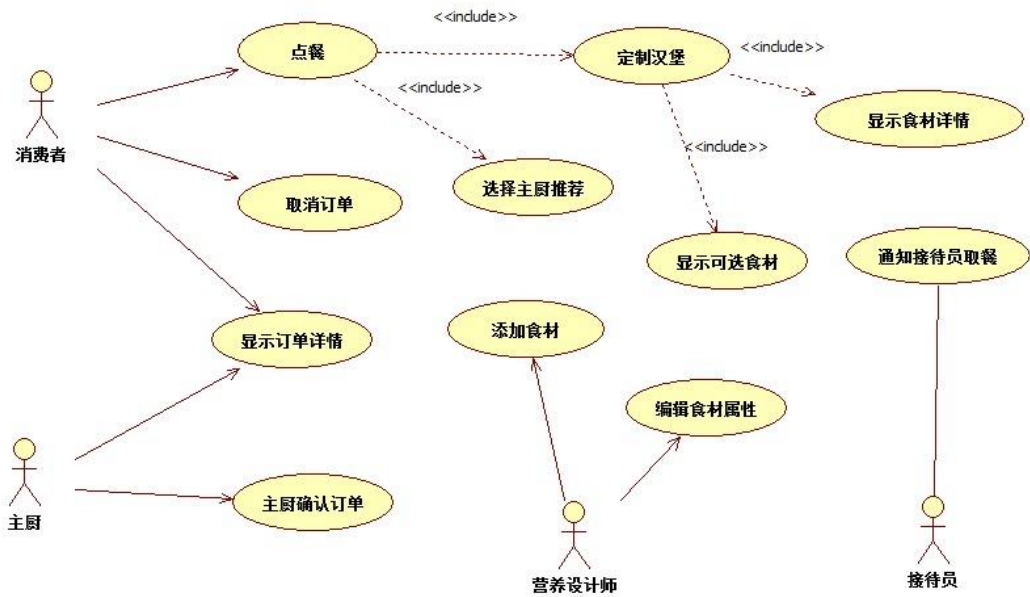


图 1 用况图

### 3. 对象模型

根据需求分析，建立对象模型如下（图 2）：

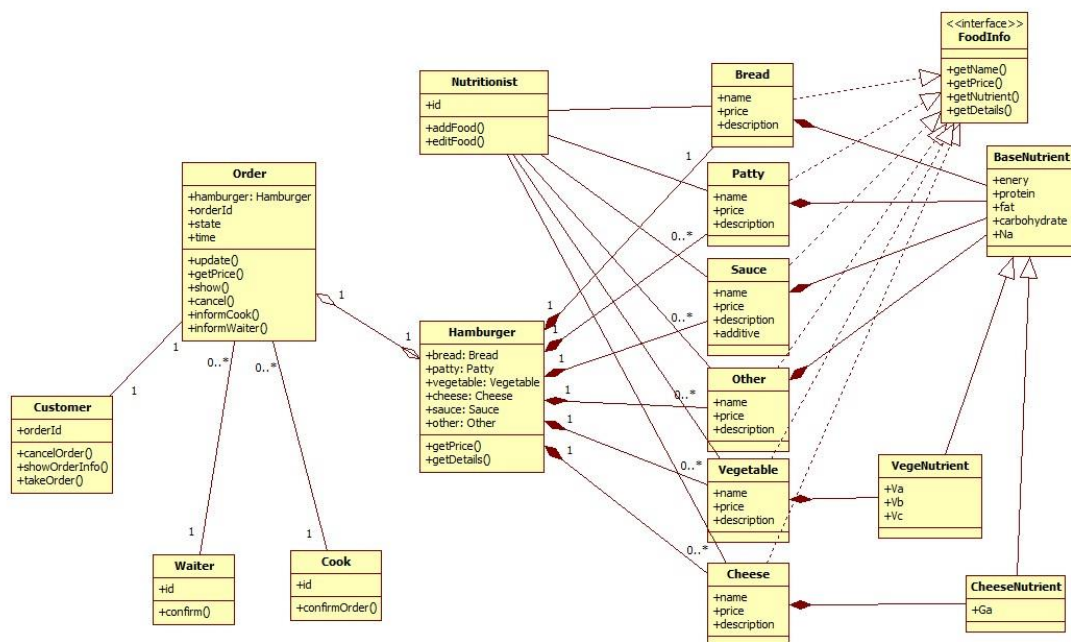


图 2 对象模型

#### 3.1. 类图分析

##### ● 涉众人员类：

##### 1) 接待员类(Waiter)

接待员接收通知，并送餐给消费者。

confirm 操作：表示确认送餐。

## 2) 主厨类 (Cook)

主厨接收通知，确认消费者的订单，并制作汉堡。

confirmOrder 操作：表示确认订单。

## 3) 营养设计师 (Nutritionist)

营养设计师设计并添加变种食材，编辑食材营养成分等参数。

addFood 操作：添加变种食材。

editFood 操作：编辑食材参数。

Waiter、Cook 和 Nutritionist 的属性 id 表示工号。

## 4) 消费者类 (Customer)

消费者有一个属性：orderId，表示消费者持有的订单号。接待员凭订单号送餐给消费者。

消费者涉及的操作主要有三个：点餐（下订单），查看订单详情和取消订单，分别对应 takeOrder，showOrderInfo 和 cancelOrder。

### ● 食材类：

六种食材分别建类：面包类 (Bread)、肉饼类 (Patty)、蔬菜类 (Vegetable)、芝士类 (Cheese)、酱料类 (Sauce) 和其他类 (Other)。

#### 属性：

##### 1) 共有公共属性

name 表示食材名称；price 表示食材单价；description 表示食材描述信息。

这里假设食材的用量和单位是系统确定好的，消费者不能指定或更改。

##### 2) 营养成分属性

分为基本营养成分和食材特有营养成分。

基本营养成分是每种食材都具有的营养成分，设计成单独的类 BaseNutrient，属性有：能量 (energy)、蛋白质 (protein)、脂肪 (fat)、糖类/碳水化合物 (sugar/carbohydrate) 和钠 (Na)。

食材特有的营养成分：某些食材有特殊的营养成分，具体的是蔬菜有维生素成分，芝士有钙成分。因此对于蔬菜和芝士的营养成分，分别设计两个营养成分类，



继承基本营养成分类 BaseNutrient：蔬菜营养成分类 VegeNutrient，特有属性有维生素 A(Va)、维生素 B(Vb) 和维生素 C(Vc)；芝士营养成分类 CheeseNutrient，特有属性有钙 (Ga)。

六种食材中，蔬菜类 (Vegetable) 聚合 VegeNutrient，芝士类 (Cheese) 聚合 CheeseNutrient，其他食材类均直接聚合 BaseNutrient。

### 3) 特有属性

只有酱料类 (Sauce) 有特有属性：additive，记录消费者选择的酱料添加剂 (盐、钙或者蛋白粉)。

### 操作：

六种食材设计了统一的操作 (get/set)，通过 FoodInfo 接口来实现：getName 表示获取食材名称；getPrice 表示获取食材单价；getNutrient 表示获取食材营养成分；getDetails 表示获取食材所有 (详细) 信息。写操作统一用 setDetails 表示。

读操作接口 (get) 是公共接口；写操作接口 (set) 开放给营养师 (nutritionist)。

## ● 汉堡类 (Hamburger)

汉堡类的属性即是六种组合的食材，**汉堡类和六种食材类是组合关系**。除了面包外，汉堡类和食材类都是 1 对多的关系，一个汉堡有且只有一个面包，但可以有多种多个其他的食材 (肉饼、蔬菜、酱料、芝士和其他)。

汉堡类的操作有两个：getPrice 表示汉堡总价格；getDetails 表示获取汉堡组成的六种食材的详情。

## ● 订单类 (Order)

订单类的属性有：hamburger 表示该订单所定制的汉堡；orderId 表示订单号，这个是订单唯一的标识符；state 表示订单状态；time 表示订单创建的时间。

订单状态 (state)：“等待下单”表示消费者正在定制汉堡中，尚未确认下单；“已下单”表示消费者定制完成，确认下单；“已有厨师确认”表示主厨确认了消费者的订单；“已有服务器确认完成”表示订单完成；“已取消”表示订单被消费者取消。

订单类的操作有：update 表示更新订单状态；getPrice 表示获取订单的总

价格；show 表示显示订单详情；cancel 表示取消订单；informCook 表示通知主厨确认订单；informWaiter 表示通知接待员取餐。

订单持有 Hamburger 类对象，与 Hamburger 类属于聚合关系。

### 3.2. 主要功能建模

下面用辅助模型刻画系统主要功能。图 3 刻画了消费者订餐的活动图；图 6 刻画了消费者定制汉堡的顺序图；图 4 刻画了消费者取消订单的活动图；图 5 刻画了订单的生存周期（状态机图）。

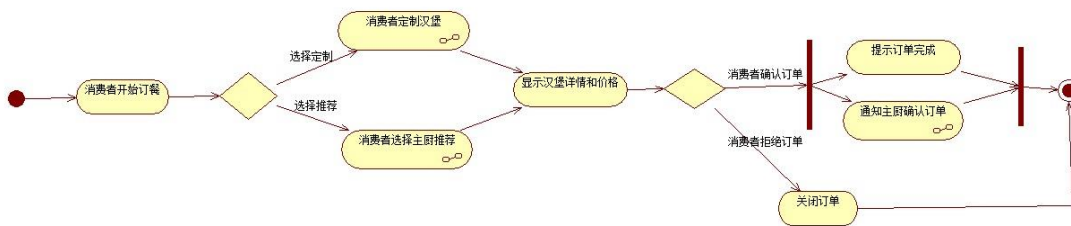


图 3 消费者订餐活动图

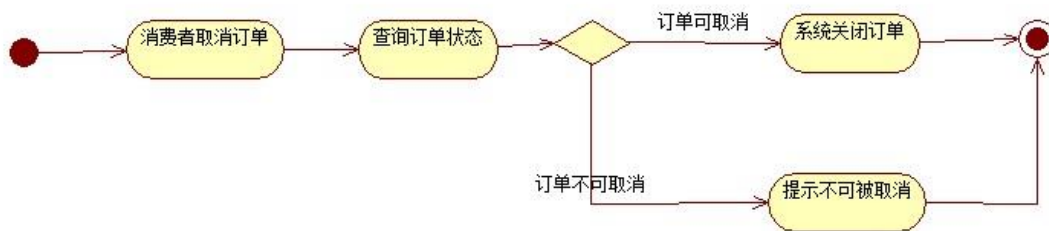


图 4 消费者取消订单活动图

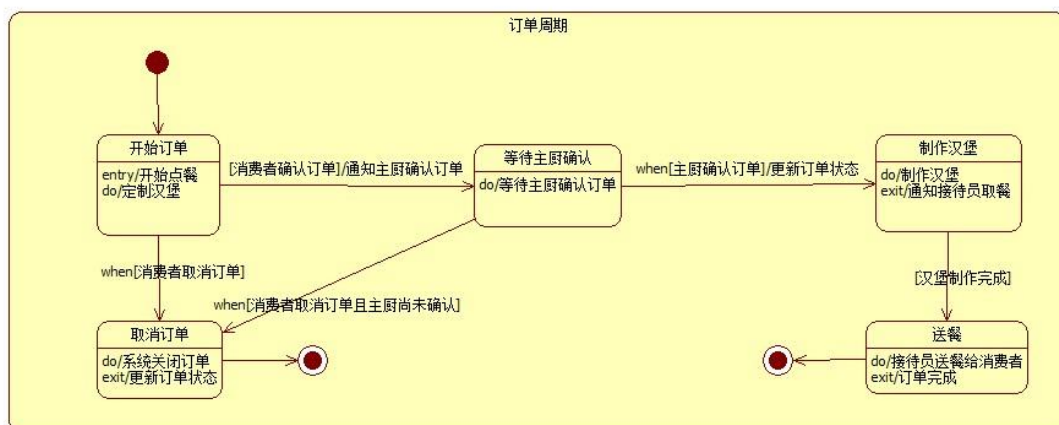


图 5 订单生存周期状态机图

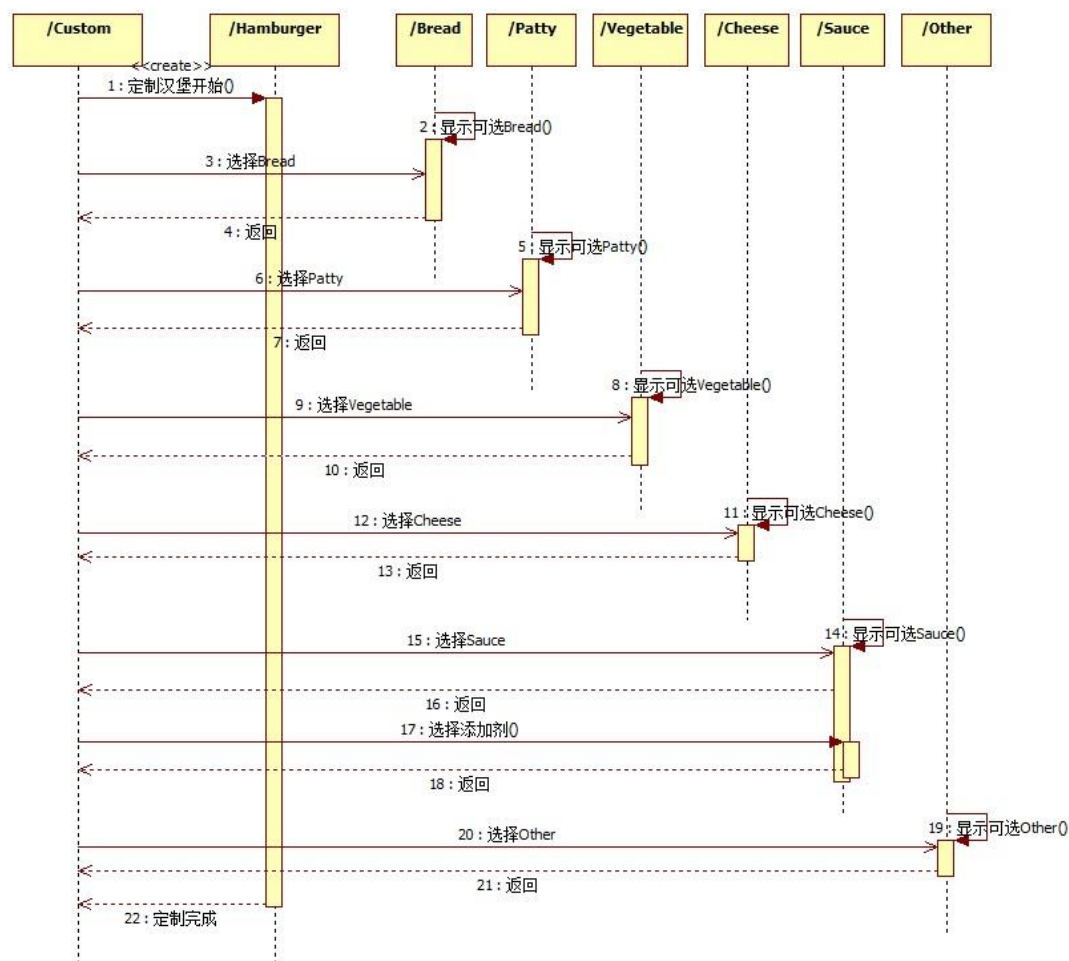


图 6 消费者定制汉堡顺序图

**消费者订餐：**如图 3，消费者有两种订餐方式，定制汉堡方式以及选择主厨推荐方式。消费者确认订单后，系统会触发通知主厨确认订单活动。

**消费者定制汉堡：**如图 6，消费者定制开始时，创建汉堡类；然后逐步选择六种食材，最后完成定制。

**消费者取消订单：**如图 4，消费者申请取消订单后，系统查询并判断订单状态：如果订单可取消（尚未被主厨确认），那么系统关闭订单（订单状态设置为已被取消），活动结束；如果订单不可被取消（订单已被厨师确认），那么提示用户不可取消订单，活动结束。

图 5 说明了订单的生存周期及其相应的事件和活动：

**开始订单状态：**

- 1) 订单开始后，消费者开始定制汉堡；
- 2) 消费者定制汉堡完成后，检查消费者是否确认订单，如果确认那么通知主厨

确认订单，进入等待主厨确认状态；

3) 开始订单状态中，如果消费者取消了订单，那么进入取消订单状态。

**等待主厨确认状态：**

1) 这个状态中等待主厨确认订单；

2) 如果消费者取消订单而且主厨尚未确认，那么进入取消订单状态；

3) 如果主厨确认了订单，那么同步更新订单状态（更新为“已有厨师确认”），进入制作汉堡状态。

**制作汉堡状态：**

1) 这个状态中做制作汉堡的操作；汉堡制作完成后，通知接待员取餐；

2) 如果汉堡制作完成，进入送餐状态。

**送餐状态：**

这个状态中接待员取餐并送餐给消费者，之后订单完成

**取消订单状态：**

这个状态中系统关系订单，更新订单状态（更新为“已被取消”）。

## 4. OOD 模型

本系统设计中采用 client/server 模式,web 服务端(java 语言),ios 客户端 (swift 语言); 采用关系型数据库系统 (mySQL) 进行数据管理。

### 4.1. 问题域部分

本设计的 00A（参照图 2 的对象模型）中，类之间没有多继承关系（存在单继承），多是聚合和关联。类图上不用做改动。

在提升性能上，考虑到系统客户端需要通过网络与服务端进行数据通信，为提高数据交换的效率和系统流畅性，设计缓冲区，对数据进行预读取。

### 4.2. 控制驱动部分

00A 中没有主动类的设计，这里加入主动类，类图如下。

人员类（消费者、营养设计师、主厨、接待员）全部设计成主动类。主动类发起整个系统的各种活动。

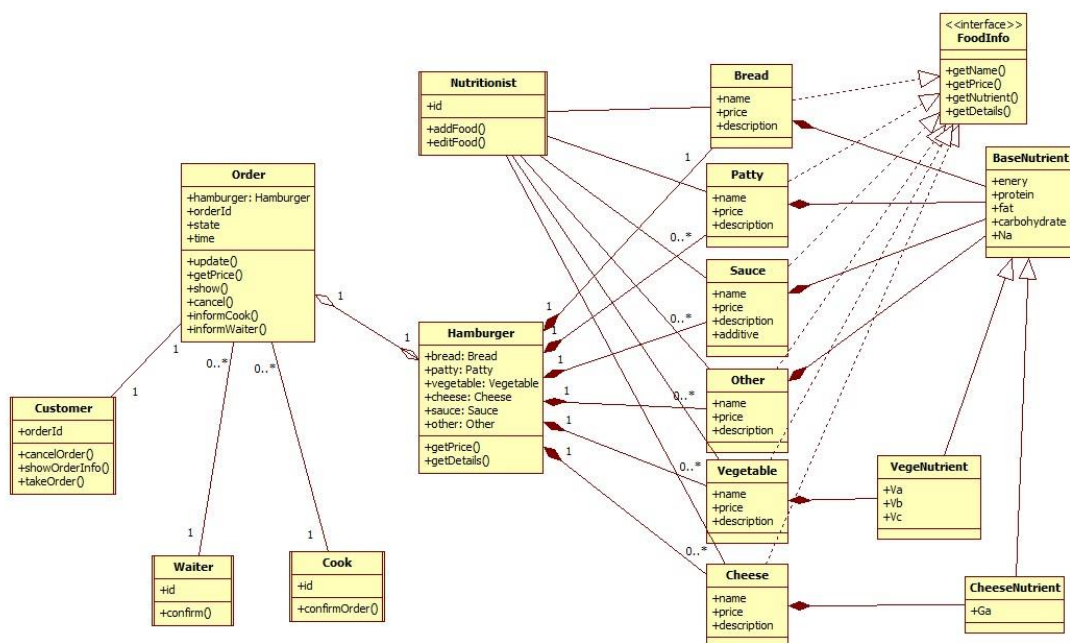


图 7 设计了主动类的类图

### 4.3. 数据接口部分

#### 1) 服务端数据库设计

根据类图（图 2），建立了三个数据表：food 表，food\_detail 表，order 表。

food 表存储制作汉堡所需要的六类食材，记录食材类型，以 food\_id 为主键。

food\_detail 表所有食材的具体食材信息，记录食材名称、单价、描述信息，以及营养成分信息，以 food\_detail\_id 为主键。这里对于营养成分采取冗余式设计，不对共有营养成分和特有营养成分进行区分，对每个食材，记录所有可能的营养成分信息。

food 表和 food\_detail 表为 1 对多的关系，因此 food\_detail 表中记录 food\_id（作为外键）。

order 表存储消费者所有的订单，记录订单号、订单状态、订单总价、订单创建时间，以 main\_id 为主键。

类图中订单类和汉堡类是 1 对 1 的关系，因此可以把汉堡类信息记录到 order 表中；汉堡类组合了六种食材，和每种食材的对应关系是 1 对 1，因此将汉堡类的六种组合食材记录到 order 表中。order 表记录六种组合食材的

food\_detail\_id。

人员类不需要建表（本系统主要提供订餐功能，因此这里不为店员设计数据库表）。

## 2) 网络接口设计

设计网络接口，提供统一的网络访问协议，供所有需要与服务端进行通信的类实现。

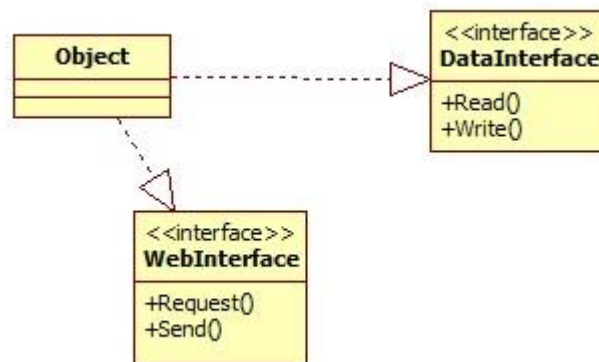


图 8 服务端数据接口

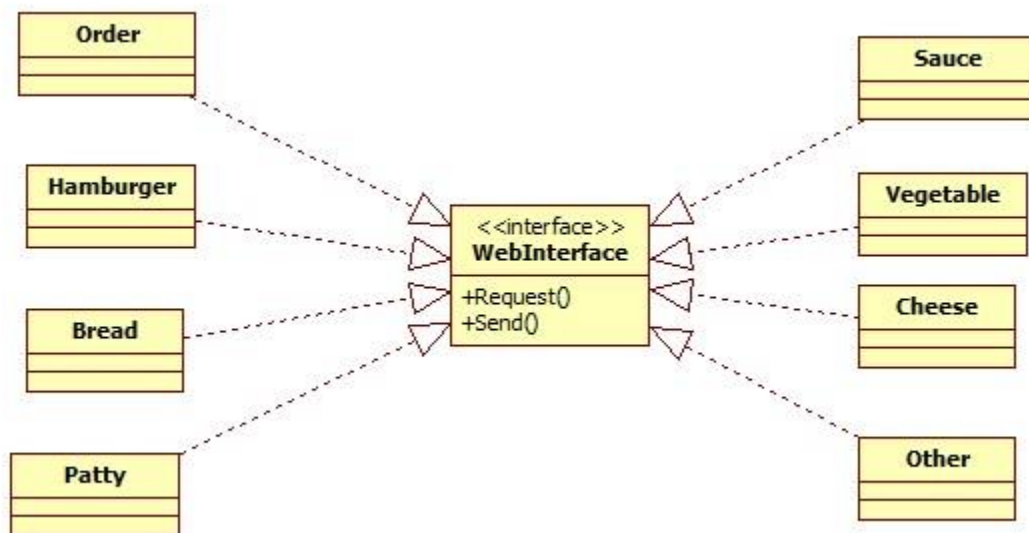


图 9 客户端网络接口设计

服务端的数据接口和网络接口如图 8。

服务端对数据库进行管理，因此设计了数据库访问接口 **DataInterface** 对数据库进行存取操作；**WebInterface** 表示网络接口，用于与客户端进行数据通信。**Object** 是一个持久化对象，实现网络接口和数据访问接口，进行数据操作和网

络通信。

客户端网络接口与相关类的关系如图 9。

网络接口提供两个操作：`request` 表示向服务端请求内容；`send` 表示向服务端发送内容。六种食材类、汉堡类和订单类实现网络接口，以此与服务端进行通信。

#### 4.4. 人机交互部分

这部分介绍系统主要的 UI 界面设计。

系统界面主要有：主界面、厨师界面、接待员界面、主厨推荐界面、食材选择界面、食材详情界面、订单界面、食材列表界面和食材编辑界面。

**主界面**是消费者点餐的入口，也是厨师界面、接待员界面和营养设计师相关界面（食材列表界面和食材编辑界面）的入口。

**厨师界面**：厨师确认消费者订单的界面。

**接待员界面**：接待员确认领餐的界面。

营养设计师相关的界面是：**食材列表界面**和**食材编辑界面**。

**食材列表界面**：列出数据库中所有种类的食材。

**食材编辑界面**：营养设计师在此界面添加变种食材以及修改食材参数。

消费者点餐涉及到：主厨推荐界面、食材选择界面、食材详情界面和订单界面。

**主厨推荐界面**：消费者在此界面选择主厨推荐的套餐并下单。

**食材选择界面**：消费者定制汉堡过程中选择食材（六大类食材）的界面。

**食材详情界面**：展示食材详细信息（名称、价格、图片等）的界面。

**订单界面**是显示当前订单详情的界面，消费者在此界面中确认订单。

图 7 展示了各界面类之间的关系，以关联关系体现。UI 类中各种 UI 控件（文本框、按钮、图片框、ListView 等）在这里没有体现。



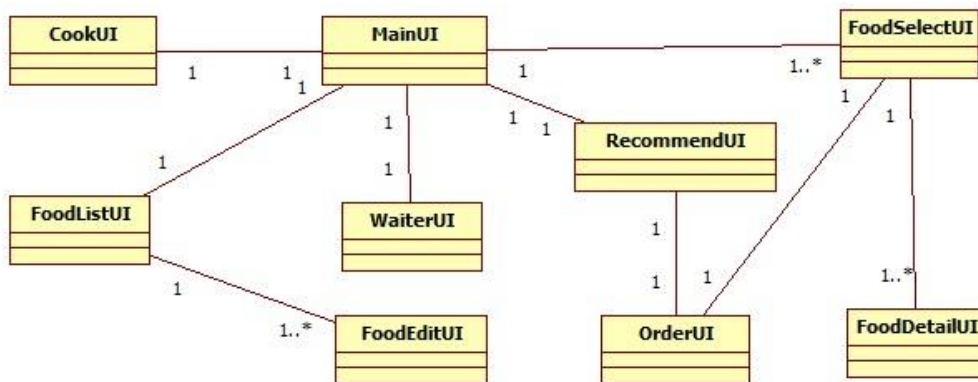


图 10 UI 类

#### 4.5. OOD 模型

加上 Web 接口类和 UI 类的对象模型如图 11 所示。

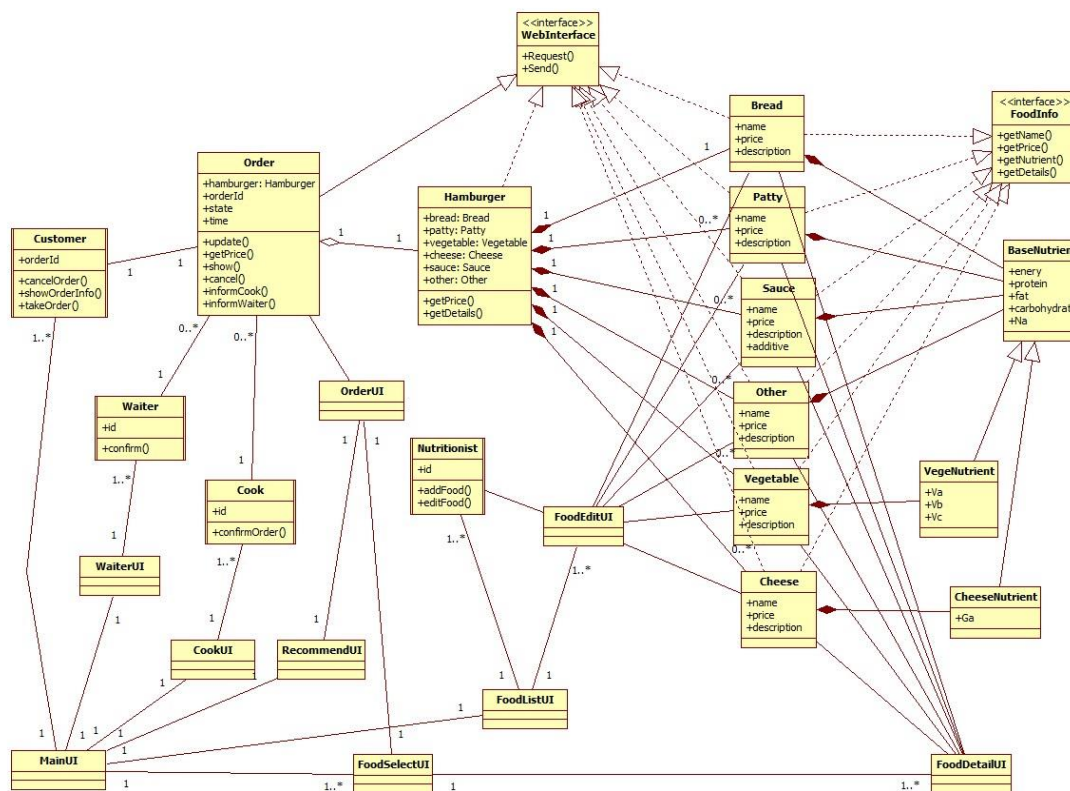


图 11 OOD 对象模型

## 5. 总结

至此我们完成了麦德劳汉堡套餐系统全部的设计和开发。按照需求方提出的功能需求，本文档完成了 OOA 和 OOD。OOA 部分，我们做出了需求分析和对象模型分析，对于本系统中主要的功能模块，我们用辅助对象模型加以更清晰的刻画。



OOD 部分，我们在 OOA 文档基础上建立设计方案，采用 client/server 模式进行开发。对控制驱动模块、数据管理模块、网络交互模块以及客户端人机交互模块进行了 OOD 设计，并完成了系统的全部开发。