Agile POS

前台设计说明书

Version 1.1

Prepared by Zhang Yi

2011/3/12

Copyright © 2011

深圳库瑞德科技有限公司

目录

[1.引言 4](#_Toc292699383)

[1.1 项目介绍 4](#_Toc292699384)

[1.2 文档概述 4](#_Toc292699385)

[2.总体设计 4](#_Toc292699386)

[2.1 需求概述 4](#_Toc292699387)

[2.2 软件结构 5](#_Toc292699388)

[2.3 运行环境 5](#_Toc292699389)

[2.3.1 软件环境 5](#_Toc292699390)

[2.3.2 硬件配置 5](#_Toc292699391)

[3.详细设计 5](#_Toc292699392)

[3.1 页面切换和动态生成 5](#_Toc292699393)

[3.1.1 功能 5](#_Toc292699394)

[3.1.2 模块逻辑 6](#_Toc292699395)

[3.1.3 存储分配 6](#_Toc292699396)

[3.2 开台 7](#_Toc292699397)

[3.2.1 功能 7](#_Toc292699398)

[3.2.2 模块逻辑 7](#_Toc292699399)

[3.2.3 存储分配 8](#_Toc292699400)

[3.2.4 测试要点 8](#_Toc292699401)

[3.3 点菜 9](#_Toc292699402)

[3.3.1 功能 9](#_Toc292699403)

[3.3.2 模块逻辑 9](#_Toc292699404)

[3.3.3 存储分配 10](#_Toc292699405)

[3.3.4 测试要点 11](#_Toc292699406)

[3.4 跟单 11](#_Toc292699407)

[3.4.1 功能 11](#_Toc292699408)

[3.4.2 模块逻辑 14](#_Toc292699409)

[3.4.3 存储分配 14](#_Toc292699410)

[3.4.4 测试要点 14](#_Toc292699411)

[3.5 结账 14](#_Toc292699412)

[3.5.1 功能 15](#_Toc292699413)

[3.5.2 模块逻辑 15](#_Toc292699414)

[3.5.3 存储分配 15](#_Toc292699415)

[3.6 RVC配置 16](#_Toc292699416)

[3.7 登陆和权限控制 16](#_Toc292699417)

[3.8 数据库双机热备份 16](#_Toc292699418)

[3.9 多语言支持 17](#_Toc292699419)

[3.10 高低档版本 17](#_Toc292699420)

[附录 A: 术语表 17](#_Toc292699421)

[附录 B: 参考资料 17](#_Toc292699422)

版本历史

|  |  |  |  |
| --- | --- | --- | --- |
| **作者** | **时间** | **版本** | **备注说明** |
| 张毅 | 2011/03/12 | 0.1 | 原始草稿版 |
|  |  |  |  |

# 1.引言

## 项目介绍

略

## 文档概述

本设计文档详细描述了要实现的系统，包含设计和实现系统的精确标准。同时，提供了评审和测试该设计的方法。文档的范围主要包括概要设计和详细设计两部分。在概要设计中，主要描述了系统的主要模块以及模块之间的关系；在详细设计中，详细描述了每一模块的介绍、功能以及一些关键的实现伪代码。

本文档适合的读者包括：系统实现人员，系统测试人员，项目评审人员。

# 2.总体设计

## 需求概述

概要设计主要提供系统 的高层设计，包括大概的用户界面、系统架构图、包含的类等。

## 软件结构

## 运行环境

### 2.3.1 软件环境

操作系统：

Windows® XP

Windows 7

数据库：MySQL Sever 5.1及以上版本

ODBC驱动：mysql-connector-odbc-5.1.8-win32及以上版本

（3.\*的驱动会造成中文乱码）

### 2.3.2 硬件配置

CPU Pentium 800MHz 以上

内存256 MB以上

至少1GB硬盘空间

# 3.详细设计

## 页面切换和动态生成

### 3.1.1 功能

为了支持可配置的页面，所有的显示页面按钮都是动态生成。为了使按钮能够自定义颜色、字体、图片等属性，采用CRoundButton2类。

### 3.1.2 模块逻辑

在程序初始化时从配置文件中读取每个页面的信息，一次性创建好对话框。切换页面时只需显示相应的对话框即可。为了增加可扩展性，在每次切换页面时调用CPosPage的OnSetActive()函数。

**动态切换页面(next\_page)：**

不同的饭店有不同的桌位信息页面，还可能有多个桌位页面。这就需要页面可以动态添加并动态切换。对于不同RVC的显示使用不同的ini配置文件初始化，切换RVC程序会自动重启。

在PAGE文件夹中，预定的页面文件名为IDD\_XXX.ini，用户可以动态添加页面，文件名为DYN\_10\_XXX.ini。中间的数字表示加载的顺序。

### 3.1.3 存储分配

CRoundButtonStyle 用flyweight享元模式生成，节省资源。为了防止控件ID冲突，ID的范围定义为如下：

VC自动生成的控件ID从1000开始

桌位按钮、菜品从2000开始

自定义按钮从3000开始

SLU从4000开始

IDC\_BUTTON\_NEXTPAGE 5000

示例的页面配置文件如下：

|  |
| --- |
| [PAGE]  btn\_num = 1  btn\_name1 = "Sign In 登陆"  rect1 = RECT(75,68,132,140)  icon1 = "Icon\icon\_login.ico"  icon\_position1 = "down"  bkg\_color1 = RGB(255,255,0)  next\_page1 = 1  function1 = 3017 |

## 开台

### 3.2.1 功能

顾客一进入餐厅，就需要前台设置客人就餐的台位及人数。

特殊要求：

需要能查看开台时间（绝对时间和经过的时间），可以预防逃单。

对VIP区可配置操作员是否有权限操作

### 3.2.2 模块逻辑

1.输入客人数

2.进入点菜页面

在order\_head中查找table\_id，如果没有则新开桌，自动生成order\_head\_id。同时对该行锁定，其他用户不能操作该行。

采用事务：如果程序挂掉，会造成已操作数据全部丢失

采用标志位：程序挂掉，会导致锁住，其他POS无法操作

所以当前采用的是命名锁，当连接断开的时候命名锁会自动释放。

数据库逻辑如下图：



**桌态颜色**

桌子有以下的状态：

1）正常

2）已预订

3）正在使用/已开台

4）已送厨

5）停用

6）超时未付款

不同的桌态在开台界面用不同颜色显示。

当一张桌子的状态改变时，修改tables表中的字段。



### 3.2.3 存储分配

使用到的数据库表：

order\_head

history\_order\_head

### 3.2.4 测试要点

1. 多个用户同时点一张单的时候是否正常。

2. 开单过程中数据库异常的处理

3. 在餐桌界面是否会定时刷新桌态

## 点菜

### 3.3.1 功能

电子菜谱

显示菜式的彩图，每页十个图片左右，菜式图片下是菜式的名称与价格。

菜品的行列数可配置。是否显示价格的配置。

对于打包带走的作出标志

时价出品和需要称重商品的处理

VIP账单/加急账单：提示厨房和传菜部慎重对待

### 3.3.2 模块逻辑

1. 菜品分类的处理，有2种方式
2. 在数据库中建立一、二级菜单
3. 只有SLU，通过切换页面实现

为了方便，使用两级菜单实现。中间显示大的分类（item\_main\_group）：食品、饮料等，点击后菜品区显示二级分类（即SLU），点击SLU后才显示对应的菜品。注意：数据表item\_second\_group未使用！

#define IDC\_CLASS\_BUTTON 4000 //大的分类从4000开始

#define IDC\_SLU\_BUTTON 4100 //SLU从4100开始

可以在页面中自由配置这两类按钮，当点击IDC\_CLASS\_BUTTON时，显示该大类里面的小分类SLU；当点击IDC\_SLU\_BUTTON时显示该SLU下的菜品。

1. 数量、规格操作

有两种处理方式：

1. 每个操作是单独的按钮(micros)
2. 点击菜品后自动弹出操作按钮(饮食通)

采用micros的方式。

点击规格键弹出对话框修改可选规格。



1. 设置‘打包带走’

order\_head表中eat\_type指明整张单是堂食还是外带；order\_detail表中eat\_type指明具体某个菜是堂食还是外带。

1. 配料

判断required\_condiment字段，如果非空则自动弹出需要的配料，不选择不能退出。

从数据库找到配料，配料按钮与菜品按钮不同，点击完后自动返回到前一页。

配料界面重用oderdlg，跳转到配料界面时，隐藏原来的按钮，将配料相关按钮显示

### 3.3.3 存储分配

数据结构

|  |
| --- |
| struct OrderDetail  {  BOOL b\_saved;//是否已保存到数据库  BOOL weight\_required;//是否需要称重  BOOL b\_hascondiment;  int order\_id;//order\_detail\_id  int item\_id;//菜品的id  int n\_checkID;//分单号  // int condiment\_group;//需要的配料种类,非0表示有配料  int n\_belong\_item;//属于哪个菜的配料  int n\_eattype;//堂食还是外带  double quantity;//菜品的数量或重量  double item\_price;//单价  double total\_price;//总价=单价\*数量  wchar\_t unit[10];//菜品单位  wchar\_t item\_name[64];//菜品名称  }; |

CTypedPtrList<CPtrList,OrderDetail \*> m\_orderList;//已点菜品列表

记录全局的点菜列表，在不同页面间切换时从该列表中读取显示。

使用到的数据库表：

menu\_item

menu\_item\_class

item\_main\_group

descriptors\_menu\_item\_slu

order\_detail

### 3.3.4 测试要点

1. 不同数量的菜品，总计的金额是否正确

## 跟单

### 3.4.1 功能

跟单是对之前开单顾客的服务跟进，包括：

移台、改人数、加菜、退菜、合并桌位，拆分桌位（需要有账单号以便后续查询）

#### 3.4.1.1 退菜

退菜的处理：

需要权限，不能随意退菜

若菜品还在制作，通知厨房停止；若菜品已制作完成，则做额外的处理

初始化时从数据库中读取已点菜品，置标志位bSaved=TRUE；

新点菜品加入同一个列表bSaved=FALSE；

退菜时bSaved=TRUE需要权限从数据库删除

对于已打印的,在单子上打印负数。

退菜在数据库中有2种形式：一是is\_return\_item置为TRUE的，是先点了菜后找到已点的菜退掉；二是菜品的quantity为负的，这种可以在点单上显示。

当前反应退菜操作较为复杂，特别是点多个菜只退其中一部分的时候。改进为如下：

1. 选中菜品，然后点击void键
2. 如果菜品的数量>1，弹出对话框让用户输入退菜的数量
3. 在数据库中插入退菜(新增记录，is\_return\_item=TRUE)
4. 更新原菜品数量为退菜后的数量

#### 3.4.1.2 移台

Micros实现：

1. 新开台
2. 选择调台
3. 输入桌号
4. 将输入的那桌账单移动到新开的台

使用饮食通的模式，直接输入想要移动到的桌号，更新数据库中的桌号即可。

可以考虑更方便的实现（拖动）。

移台需要加入权限控制，以免飞单。

#### 3.4.1.3 分单

分单功能参考aloha的实现，对于多张单的显示：

左侧由一个CCoolTabCtrl作为容器，每张单由CCheckDlg对话框实现。

修改CCoolTabCtrl，当只有一个页面的时候不显示标签页；添加删除页面的功能。

Split分单对话框有如下功能：

1. Add –加单

往order\_head中插入分单信息，check\_id表示了每张分单的ID

1. Combine—全选
2. 移动的项用红色标示
3. Split items—拆分单个菜品

未下单的不能拆分成散的，例如1份菜在未下单时是不能拆分成2份的，以免厨房造成误解。

拆单需要将数据库中原有记录删除，新增成多条记录。

#### 3.4.1.4 退款

退款需要经理以上的权限，不能由收银员随意操作。

退款实际上是新增一张金额为负数的单。

操作流程如下：

1. 点击退款键
2. 弹出权限对话框，确认退款权限
3. 选择退款原因
4. 跳转到点菜界面，在退款状态，点的菜都是负的

#### 3.4.1.5 合台、合单

假设从A台合并到B台，数据库需要如下改变：

order\_head表：

更新customer\_num

check\_id相同的两张表需要合并should\_amount、actual\_amount

order\_detail表：

更新A台点菜的order\_head\_id

合单将一个单合并到座位的第一张单。

* 点进一张桌
* 点击 ‘合单’
* 显示当前已开的单供用户选择

假设从A3单合并到B桌的第一张单B1

order\_head表合并should\_amount、actual\_amount

order\_detail表将A桌3号单更新为B桌1号单（order\_head\_id、check\_id）

### 3.4.2 模块逻辑

### 3.4.3 存储分配

使用到的数据库表：

order\_detail

order\_head

reasons

### 3.4.4 测试要点

1. 退款状态下各种按钮能否点击

## 结账

顾客完成就餐则需要结帐。提供多种结账方式：现金、刷卡、会员卡、欠账、优惠券抵用等。另外结账常常要计算折扣，不同等级的顾客给予不同折扣。

结账时自动计算出顾客的花费，提示需要找补的零钱，自动打开钱箱。

### 3.5.1 功能

#### 3.5.1.1 Service/Discount

服务费和折扣与菜品一样，存储在order\_detail表中。menu\_item\_id -1标示服务费, -2标示折扣 。

由于有分单的情况，服务费和折扣在分单页面不显示，需要先分好单再加入服务费、折扣。

#### 3.5.1.2 某些菜不能打折

对于某些菜不能打折或者使用较高的折扣。

使用menu\_item\_class表的discount\_itemizer字段。若不设置默认所有折扣都能使用。若设置了值则只有discount\_service\_id大于discount\_itemizer的折扣才能使用。

这样处理折扣需要按照特定顺序添加（id越大的折扣值越低），后续可以考虑在discount\_service中设置折扣的级别。

#### 3.5.1.3 最低消费

方式一、 每张桌子都设置最低消费金额。 配置较为麻烦，而且修改起来很不方便。

方式二、 添加最低消费consumption\_limit表，对桌子配置使用哪种消费限制。

### 3.5.2 模块逻辑

各种付款方式不显示在单上，只记录总数。但是在数据库中记录付款的详细信息。

折扣页面：采用模态对话框，不覆盖账单信息，这样可以看到基本的信息。

各种结账方式只记录类型，通过tender\_media表确定该付款方式的行为：是否需要权限、是否要输入备注、是否要输入数量（折扣券）。

### 3.5.3 存储分配

使用到的数据库表：

discount\_service

history\_order\_detail

history\_order\_head

tender\_media

payment

## RVC配置

显示时间格式

时间排列，日期可以自由设置为中国排列习惯（年-月-日）或外国习惯（月-日-年）或（日-月-年）

菜品页行列数

用户ID位数可配置

## 登陆和权限控制

### 3.7.1 功能

需要用到权限控制的地方较多，在此处列举如下（不全的在后续添加）：

* 退菜
* 退款
* 移台
* 查看报表
* VIP区操作
* 宴请等付款操作

权限控制在数据表employee\_class中配置。

### 3.7.2模块逻辑

权限控制由函数RequeastAuth完成。当用户登录时，获取用户的所有权限，存储在一个map中。当需要认证授权的时候，首先查看当前登录用户是否有权限，若没有则弹出对话框输入授权用户。



## 数据库双机热备份

Mysql自带了replication热备份功能。

根据资料MySQL 5.5 对该特性做了改进。

需要考虑是自动切换到备份服务器还是由用户手动切换。

用户手动切换：

程序启动时，如果是连接到备份服务器需要提示。

## 套餐

### 3.9.1 功能

该功能实现combo meals功能。可以对菜品实现可选，并且操作可以返回上一步、中途取消。

### 3.9.2 模块逻辑

当菜品的类型为ITEM\_SET时表示套餐。套餐通过查询course\_group和course\_detail表，先自动加入必须菜品；对于可选菜品，根据分组显示，由用户选择。

1. 可选多种的情况（如3选2）：

分组添加一个字段，表明重复的次数

1. 带必选配料的菜品

带必选配料的菜只能配置成可选，当用户点击选择后，自动弹出配料界面供用户选择

右侧显示该套餐内可选的菜品。通过搜索course\_group显示。





左侧显示套餐已点的内容，用户可以选中后点击‘做法’‘数量’‘删除’进行修改。点击取消可中途退出点菜，套餐将不会生效。当点击完成时检查每个组的菜品数目是否符合要求，对不符合要求的组进行提示并且自动跳转到该组显示。

中间显示套餐的可选组别，当前组的菜选择完后自动跳转到下一个组。内部需要存储每一个组点的菜，选择组别的时候可以返回到该组，并修改已点的内容。

套餐换菜

如果用户输入错误，可以通过套餐换菜按钮来更正。

在点菜界面新增按钮‘套餐换菜’，点击后再次弹出套餐对话框。

初始化时将该套餐的所有菜品载入。

点击VOID后记录退菜的数量void\_num，之后用户最多点void\_num个菜。

需要记录在套餐换菜模式下新增的菜和退掉的菜，点击完成后保存到数据并厨打。

### 3.9.3 存储分配

由于用户可以在不同套餐组中来回切换，需要存储每一个组的详细信息。

|  |
| --- |
| struct CourseInfo  {  int id;//组号  int total\_select;//该套餐组总共可点的次数  int can\_select;//剩余可点的份数  int slu\_id;  vector<OrderDetail\*> m\_orders;//该组内已点菜品  }; |

## 开台关联点菜

通过表order\_detail\_default实现。对于普通菜品，开台后从数据库表中查询并自动添加到已点菜品中。

折扣和服务费：

对于ID为-1（ITEM\_ID\_SERVICE）的为服务费；-2（ITEM\_ID\_DISCOUNT）的为折扣。

discount\_service\_id为使用的折扣或服务费id。

对于服务费和折扣，需要实时刷新价格。服务费和折扣的计算都按菜品的总价计算，不同顺序对结果不会有影响。

## 抹零

1. 开台自动收取折扣和服务费

当配置为整单打折和百分比折扣时，才会自动更新折扣金额。

对于部分打折或固定金额折扣，新点菜后不需要重新计算。

1. 抹零

抹零的计算有2种：

一是针对当前计算出来的折扣抹零，例如使用了一个8.8折，计算出来折扣为10.3元，这时可以使用抹零，实际显示的为8.8折 -10元。

二是针对整单抹零，例如一张单的总金额为90.5元，可以使用抹零折扣使总价为90元。

使用数据库assume\_entire\_amount字段

## 估清

对于设置了估清的菜品，在右上角显示剩余数量，每点一次扣除。当数量减到0后不再继续往下减。剩余数量除了在内存中更新外，还要同步到数据库中。如果剩余数量小于等于0，会弹出提示框，用户仍然可以继续点菜。

下面的操作会影响到剩余数量：

* 点菜
* 选中菜品后退菜
* 选中后修改菜品数量

对于点击void，然后退菜和退款中进行的退菜，不将数量加回去。因为这时菜品可能已经制作完成了。

注意：如果点了菜未送厨或存储到数据库，这时关掉程序，会导致剩余数量计算不准确。

## 菜品等叫/叫起

order\_detail表增加description文本字段，作为菜品的备注信息。

备注信息只在点菜界面显示，送厨时打印；在付款界面不显示，收银单也不打印出来。

**等叫处理：**

等叫作为一种内置的备注，点菜时可插入等叫备注。

**整桌等叫：**

每个菜都添加等叫备注。

‘叫起’改为‘起菜’，当发现菜品备注中有‘等叫’字样时，删除全部等叫，并添加‘起菜’备注。





**菜品做法：**

菜品做法通过查询配料表显示，在下方添加固定的‘多’‘少’‘免’等字样。

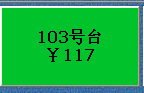
## 清机日结操作

1. 检查是否有未结单，提示用户
2. 设置估清菜品
3. 重置账单号
4. 打印日结报表
5. 发送短信给经理

## 桌台显示信息

桌子开台后，显示详细的信息，包括：

* 桌名
* 客人数
* 开单时间 \*
* 消费金额





桌台界面的右下侧，显示当天的营业状况：

* 已开桌数
* 总金额
* 总客数/当前客人数

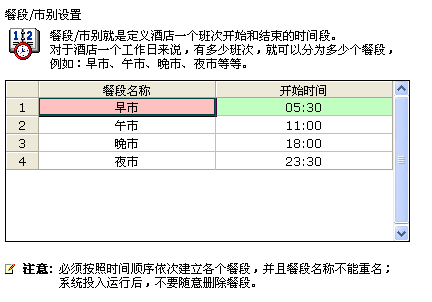








## 餐段市别





1.登录后显示当前的市别

2.日结报表统计各个市别的销售信息

3.交接班打印当前班次

## 宴会酒席

酒席预订在后台预订中处理，前台只包括预订抵达的操作。

* 从预订表party\_table中查询预订了哪些桌子
* 检查这些桌子是否都是空台，如果有占用的让用户选择换桌
* 多个桌子统一开台，添加菜
* order\_head添加party\_id字段，相同值的为关联桌子
* 付款时自动提示有关联桌，是否合并付款；合并付款则将其他桌子合并到当前桌，然后正常付款，收银单菜品需要自动合并。
* 送厨时提示有关联桌，是否一起送厨
* 临时添加桌子关联到宴会中;新开台，在功能中添加设置关联桌

## 税率

税率的计算基本同服务费。但是统计的显示需要考虑。

小计、已付、应付为总是显示。

以下为2进制表示

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 8 | 4 | 2 | 1 |
|  | 税率 | 已点菜数 | 折扣 | 服务费 |

后台可设置税率的显示名称，例如在澳大利亚一般叫GST

order\_head表添加tax\_amount，便于统计和显示

通过菜品属性tax\_group查找到所属的税率组，在点菜的时候计算该菜品的税额。改数量、赠送、改价等操作均需要重新计算税额。

对于菜品打折如果不做处理会造成多收税额。所以对每个菜的税额计算要加上折扣，目前只支持整单折扣。具体计算见[3.20](#_折扣、服务费、税率计算)

## 位号

座位的操作类似菜品数量，可以在点菜前输入，也可在点菜后选中菜品修改。

未选中菜品时，设置位号对后面的菜都生效。

## 折扣、服务费、税率计算

除了比例型的折扣/服务费，还需要考虑固定金额的。

固定折扣：20元现金券；员工餐抵扣$5

固定服务费：开瓶费，冲茶费，Cover charge入场费

关于税率，统一都是折扣后的金额再计算税额，如果勾选了GST，服务费也计算税额。

**计算方式一：**

**服务费当成税来处理，先计算折扣，后计算服务费**

|  |
| --- |
| subtotal=菜品小计  -discount=减折扣  +service=加服务费=(subtotal- discount)\*服务费比例  +tax=税额=(subtotal+service - discount)\*税率 |

计算折扣时，金额服务费或部分比例服务费可按配置决定是否打折。

**计算方式二（不采用）：**

**服务费当成菜品处理，先计算服务费，后计算折扣**

|  |
| --- |
| subtotal=菜品小计  +service=加服务费  -discount=减折扣=(subtotal+service)\*折扣比例  +tax=税额=(subtotal+service-discount)\*税率 |

服务费相当于一种特殊的菜品，可以加上是否打折属性。

**计算方式三：**

**Micros的设计，折扣和服务费同等级，都按菜品总价计算**

经过考虑采用第一种计算方法。下面已一张账单为例计算

|  |
| --- |
| 菜品总计: 100  代金券折:-20  整单9折:-10 |
| 小计:70  入场费: +50 （若配置为可打折，则+45）  10%服务费:+7 (可配置为不打折，则+10) |
| Tax 7%: +4.9 |

order\_detail 新增字段discount\_price用于记录该菜品的打折金额。

sales\_amount用于记录该菜品产生的销售额。

折上折比较复杂，软件禁用折上折（只支持一个整单折扣）

考虑到计算很复杂，只在买单的时候统一计算每个菜的折扣。

1. 折上折的情况，当前的软件限制，只有最后一个整单折扣才会自动更新。所以计算时只计算最后一个折扣。 (菜总折扣<实际折扣)
2. 新增部分折扣，对选中的菜品计算discount\_price,如果折上折只算一次
3. 删除部分折扣，忽略不管(菜总折扣>实际折扣)
4. 免费菜的处理，不做处理，因为折扣金额并没有变
5. 折扣后改价(无法预知，不做修改)
6. 抹零折扣、金额折扣(菜总折扣<实际折扣)
7. 反结帐后折扣不对

套餐菜品金额

course\_detail表中的price字段表示该菜品在套餐中产生的销售额。

点菜时存储到sales\_amount，套餐为0，套餐中配菜为price值

（配置时可能会有配菜的价格相加不等于套餐价格！！）

最终付款时的逻辑为：

开始

找到最后一个整单折扣

对每个可以打折的菜计算discount\_price，加上原值

找到

计算sales\_amount=actual-discount

套餐和套餐配菜belong=-1不修改

没找到

## 反结帐

反结帐用于已经结帐的单进行修改。将已结的单恢复到未结状态。

order\_head新增字段edit\_time存储反结帐编辑的时间。若该字段不为空，说明进行了反结帐，不再修改order\_end\_time。payment中新增的payment\_time改为原来的结单时间。

## 调味品做法

调味品和做法合并，在一个地方配置，后台都存储为menu\_item。

前台方面：

1. 必选调味品自动弹出，与之前处理相同
2. 做法按钮：对于没有价格的调味品，按当前做法添加到detail表的描述字段；对于带价格的调味品，当成特殊菜新增到detail表，并且显示到原菜品的下方。

只选择一个：可以添加带价格的菜

选择多个：对第一个添加。如果选择的调味品是一样的，则显示出来，可以统一修改。如果不是一样的，则提示用户，所选菜品的调味品并非一致。

OrderDlg初始化显示时也要注意显示配料的顺序

1. 去除原来的配料按钮
2. 新增‘取消做法’按钮，清除最近一次的做法。界面改大，该对话框在如下功能中重用：整桌备注、通知厨房、套餐中的调味品。



后台方面：

1. 放在一个地方配置
2. 可以重用已有配料

PDA和平板需要同步，去除做法。

## 操作历史记录

前台记录关键的操作，操作人员，时间等。后台在报表中查询。参考数据库critical\_operations (关键操作表)

关键操作包括：赠送、改价、退菜、反结帐、单品转台、合台、合单、退款、交接班，日结

为防止数据无限增大，在日结时清除该表，只保留最近6个月的记录。

## 收银备用金

在收银功能界面新增‘存入现金’和‘取出现金’按钮。参考数据库cashbox\_in\_out(钱箱金额变化表)

存入现金操作流程：

1. 检查是否有open\_drawer权限。

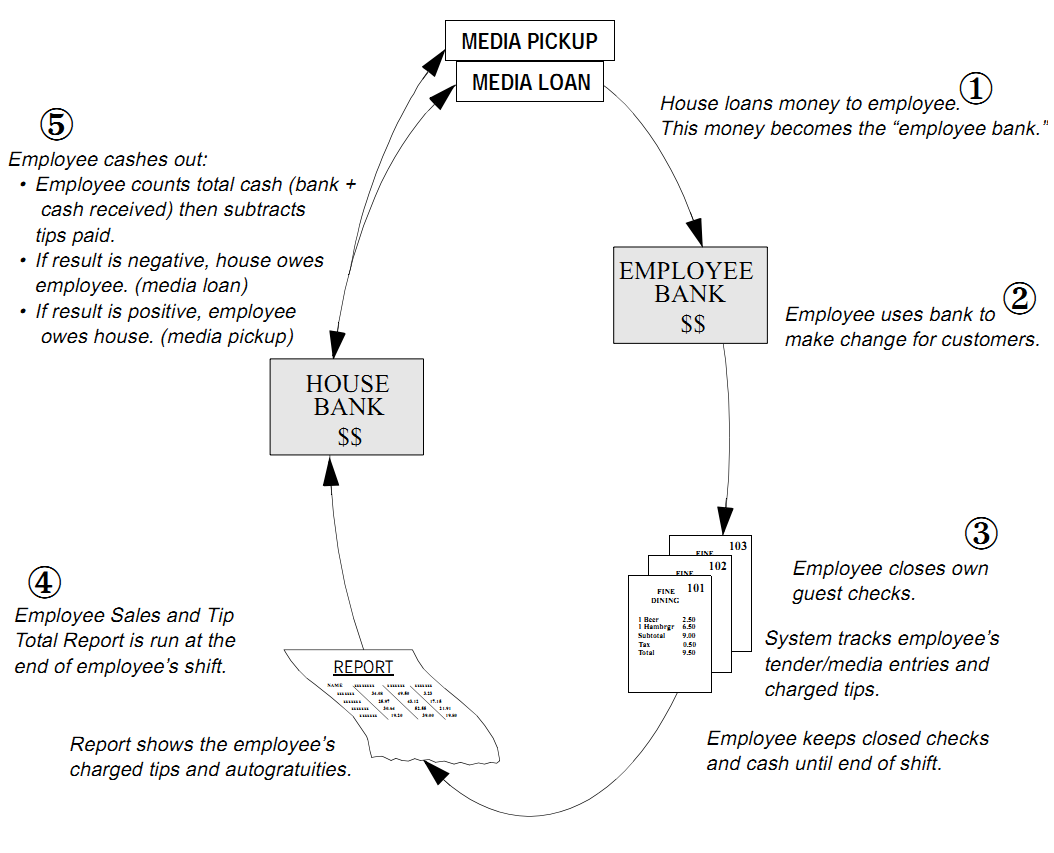
2. 输入存入金额

3. 自动打开钱箱

在交班报表统计该部分金额

+交班借零 500.00

应有现金 500.0



## 桌注

order\_head表加入remark字段，在厨房的每张单中打印。

输入桌注时重用做法对话框。

在打印模板中加入每张单的桌注。

## 桌台状态改进

分两个tab实现：导航模式和查单模式。

导航模式与当前模式相同。做如下改进：

* 空白桌台显示几人桌。
* 支持快捷键开台
* 新增菜已上齐的状态



查单模式下一侧显示已开桌的简要情况，一侧显示单据的内容。

在查单模式下可快速做一些简单操作，包括：转台，打预结单，整单催菜。

用户可快速跳转到点菜或者收银界面

新增一个已上齐菜的按钮，点击后将桌态设置为上齐菜。超过一定时间未设置该状态则提示用户。



## 系统启动时提示

在做日结的时候将相关数据存储，启动时显示。

欢迎您使用我们的产品，聚客餐饮总共已为您点单XX张，服务XX人，XX天。

数据存储在macros\_key表中，新增字段。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 中文名称 | 类型 | 空值 | 缺省值 | 说明 |
| total\_checks | 总单数 | int | NO |  |  |
| total\_guests | 总客人数 | int | NO |  |  |



## 显示系统状态

系统状态包括：

当前网络连接状态。调用API InternetGetConnectedState

数据库主机状态。（可以不要，连不上数据库所有操作都无法进行）

各个打印机的状态:

从数据库中查询使用驱动模式的物理打印机,通过驱动获取状态。如果状态不准确,则使用ping的方式查看设置是否在线.

前台启动一个单独的线程来做查询操作,也可以考虑放到单独的服务进程中,这样不会对前台的稳定性造成影响。

状态的显示：

* 只显示错误的状态
* 放在桌台界面显示，如果字数太多可以滚动显示跑马灯效果。
* 点击错误后弹出检测工具，进行更加准确的检测

## 软件升级包的制作

当前已经提供了公开下载的安装包，后续的软件升级需要有所考虑。升级包括两个部分：1.数据库升级；2.程序文件的升级。

对于文件的升级比较简单，直接覆盖文件即可。需要注意的问题包括：

1. 用户可自定义修改的文件：打印模板和ini文件不用升级；界面显示Page目录可以覆盖升级，但是对原来的做备份。
2. 升级前关闭所有在运行的服务进程
3. 升级完成后重启电脑或者重启服务

对于数据库的升级，在数据库中添加字段记录当前的数据库版本。每次升级时通过比较版本运行需要升级的sql文件，并更新数据库版本。

当前在注册表中保存了程序的安装目录和版本，升级时可以读取目录并且升级完成后设置版本号。

升级包用inno setup制作较为方便，但是数据库升级可能需要编写exe辅助实现。

# 附录 A: 术语表

# 附录 B: 参考资料