**餐厅服务管理系统**

**测试文档**

作者

学号

专业

备注

2019 年 6 月16日

**目录**

[变更记录](#_Toc471658479) 1

[I. 引言](#_Toc471658479) 2

[1.1 编写目的 6](#_Toc471658480)

[1.2 项目背景 7](#_Toc471658481)

[1.3 术语定义 8](#_Toc471658482)

[1.4 参考资料 9](#_Toc471658482)

[II. 任务概述](#_Toc471658487) 12

[2.1 目标 12](#_Toc471658480)

[2.2 测试环境 13](#_Toc471658481)

[2.3 需求概述 13](#_Toc471658481)

[2.4 条件与限制 13](#_Toc471658481)

[III. 计划 16](#_Toc471658490)

[3.1 测试方案 16](#_Toc471658480)

[3.2 测试项目 17](#_Toc471658481)

[3.3 测试准备 17](#_Toc471658481)

[3.4 测试机构及人员 17](#_Toc471658481)

Ⅳ[. 测试项目说明 18](#_Toc471658511)

[4.1 测试用例](#_Toc471658512) 18

[4.2 测试步骤及操作](#_Toc471658516) 18

[V. 评价](#_Toc471658520) 20

[5.1 准则 20](#_Toc471658512)

[5.2 结束标准 2](#_Toc471658512)0

Ⅵ. [测试日志](#_Toc471658525) 22

# 变更记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本号 | 修改点说明 | 变更人 | 审批人 | 审批日期 |
| V1.0 | 创建（包含测试计划、用例） | 唐婷 | 李怡甜 | 2019.06.09 |
| V1.1 | 补充回归测试，补充测试日志，更新最新测试结果 | 唐婷 |  |  |
| V1.2 | 补充软件配置复查 | 唐婷 |  |  |
|  |  |  |  |  |

（修改点的说明有如下几种：创建、修改+修改说明、删除+删除说明）

# Ⅰ. 引言

## 1.1 编写目的

此文档的预期读者为：用餐服务管理系统项目经理、系统软件测试人员和购买并使用系统的餐饮商。

正文部分，首先依据“点餐系统”的设计文档，为软件测试建立计划，指导软件测试人员进行测试。当项目初步实现后，测试人员依照本文档所提供计划进行测试，并将软件测试的过程、结果记录在本文档中

## 1.2 背景

“元气”餐厅服务管理系统，是将现代计算机网络技术应用于餐饮服务业的一项软件工程。其目的是实现顾客、前台、后厨的实时信息联通，以减少用餐高峰期人数过多引起的商家服务不及时甚至服务错误的问题。另外系统带有辅助性的菜单编辑、会员积分、菜品存量纪录、营业状况统计等功能，给餐厅商家提供运营的便利。

项目自201X年X月启动，经历了可行性分析、需求分析、系统设计和实现几个阶段。初步编程实现已经顺利进行，接下来应当进入软件测试阶段，以便发现和纠正系统的错误。

## 1.3 术语定义

本文档所提及的术语，其定义遵照GB/T11457标准。

~~LEM:~~ 元气用餐服务管理系统~~(Laboratory Equipment Management)~~;

用例序号：继承《用户需求说明书》需求文档中“业务需求说明”编号

## 1.4 参考资料

《软件工程导论（第6版）》 Roger S．Pressman著，机械工业出版社

《Java面向对象程序设计》，耿祥义、张跃平编著，清华大学出版社

《SQL Server 数据库教程（2008版）》，郑阿奇、刘启芬、顾韵华主编，人民邮电出版社

# Ⅱ. 任务概述

## 2.1 目标

系统最根本的目标是满足用户包括功能和性能的需求。具体而言，要求能够正确实现不同类型的使用者登录、注册功能，后台管理，前台使用功能。能够正确、有效地实现不同类型用户之间的信息交流功能。

## 2.2 测试环境

2.2.1 硬件环境

处理器在奔腾Ⅲ以上，内存在256MB以上的计算机

2.2.2 软件环境

操作系统：Microsoft Windows 2006及以上版本

数据库：Microsoft Office access；

开发工具：Java，PowerBuilder程序设计语言；

## 2.3 测试需求概述

2.3.1 数据需求

事件发生的时间；

2.3.2 事务需求

|  |  |  |
| --- | --- | --- |
| 角色 | 操作 | 备注 |
| 顾客 | 登陆、注册、点菜、催单、退菜、结账 | 登陆使用系统可以选择使用已有账号或者以游客身份登陆；结账可以使用三种方式 |
| 前台服务员 | 标记上菜、打印账单 | 是否打印账单依据顾客需求决定 |
| 厨师 | 接收订单、确认完成菜品 | 接受订单意味着反馈“接受”信息给前台 |
| 管理员 | 查看营业、编辑菜单、修改库存、修改会员信息 | 编辑菜单包括：新增、删除、修改菜品的名称及价格 |

## 2.4 条件与限制

测试过程需要具备以下条件：

1. 硬件设备：餐桌、厨房、前台（中心计算机）设备无故障运行
2. 软件系统保证：后台Windows2006以上、Access数据库正常运行
3. 人员齐备、各小组互相配合、小组内部协调
4. 项目剩余资金充裕、用于测试的时间充足

# III. 计划

## 3.1 测试方案

3.1.1 测试策略

本计划拟采用经典的测试策略，从“小型测试”开始，逐步过渡到“大型测试”。也就是从单元测试开始，逐步进入集成测试，最后进行确认测试和系统测试。当然，在集成过程中还应进行一系列的回归测试。

3.1.2 测试过程

1、设计测试用例并执行测试

2、发现问题、记录

3、调试、修改程序

4、执行测试用例，若不成功，返回步骤3

5、进行回归测试，若不成功，返回步骤3

6、测试通过，进入下一项测试

3.1.3 测试技术

单元测试：划分测试、基于故障的测试

集成测试：从动态模型导出测试用例

确认测试：α测试、β测试、黑盒测试技术

## 3.2 测试项目

|  |  |
| --- | --- |
| **测试项目** | **描述** |
| 单元测试 | 最小的可测试单元是封装起来的类和对象。因此，本阶段着重测试各类和类中封装的方法。 |
| 集成测试 | 此项目是对类间协作进行测试。采用基于线程的测试策略，把相应系统的一个输入或事件所需的类集成起来。分别集成并测试每个线程。 |
| 回归测试 | 前面的测试中中发现问题后，会对程序进行修改。应当应用回归测试保证修改没有产生副作用。 |
| 确认测试 | 此想测试的目的是验证软件的功能和性能是否达到用户的要求。可以应用传统的黑盒测试方法完成测试工作，集中检查用户可见的动作和用户可识别的输出。 |

## 3.3 测试准备

1. 各模块的主要负责人共同协商讨论

2. 阅读《软件规格说明书》、《设计说明书》，熟悉系统设计；

3. 进行测试任务安排

4. 选择合适的操作序列，编写测试用例。

## 3.4 测试人员及其任务安排

**表格设计必须修改**

|  |  |  |
| --- | --- | --- |
| 项目 | 测试者 | 任务 |
| 单元测试 | 张三 | 数据管理子系统的类测试 |
| 李四 | 人机交互子系统的类测试 |
| 王五 | 任务管理子系统的类测试 |
| 集成测试 | 张三、李四 | 测试Customer类及其协作类 |
| 张三、李四、王五 | 测试Servant类及其协作类 |
| 张三、李四、王五 | 测试Manager类及其协作类 |
| 张三、王五 | 测试Chief类及其协作类 |
| 确认测试 | 张三、李四、王五 | 黑盒测试 |
| 张三、李四、王五  受邀体验商家50人 | α测试 （受邀商家在项目实验室，在开发人员的指导下使用、测试，记录发现的问题） |
| 受邀体验商家200人 | β测试（受邀商家自行使用，半个月后反应发现的问题或改进建议） |
| 回归测试 | \ | 测试人员各自负责纠正自己测试出的错误 |

# Ⅳ. 测试项目详细说明

## 4.1 测试操作及测试用例

4.1.1 单元测试

|  |  |  |
| --- | --- | --- |
| 模块 | 被测试单元 | 测试操作 |
| 人  机  交  互 | User | 创建几个User对象，给成员变量赋不同的值；  调用getPassword()等方法，检查返回值是否正确读取成员变量的值；调用setPassword()等方法，打印出对象成员变量的值，检查成员变量的值是否进行了正确的修改； |
| Customer | 测试操作与User类单元测试类似，即检查各自每一项成员变量的读取和赋值方法。  （**每一个类所封装的成员变量读取、赋值方法都采用同样步骤进行测试，不再一一复述**） |
| Manager |
| Chef |
| Servant |
| Dish |
| Desk |
| Order |
| Bill |
| LoginFrame | 以loginFrame类为例，创建一个LoginFrame对象（即登录窗口）  检查所生成的界面的组件（包括标签、按钮、文本框等）是否缺失，布局是否与详细设计书中的规定有差别。  **（测试每一个窗口,都应进行此项检查,不再一一复述）**  窗口类还需测试事件监听 |
| MenuFrame |
| OrderFrame |
| BillFrame |
| CommentFrame |
| ServantFrame |
| ChiefFrame |
| ManagerFrame |
| VIPFrame |
| DishesFrame |
| FinanceFrame |
|  |
|  |
| 数  据  管  理 | InfoManager | 建立测试所用数据库文件,录入用户信息、菜品信息、营业信息等；  调用link()，close()方法，检查返回值获取链接成功与否的信息；  **此类为后三个manager类的父类，子类中继承父类的方法仍需测试，但测试用例同父类不再赘述，子类中覆盖的父类方法需重新设计测试用例** |
| UserManager | 调用addUser(), editUser()，showUser()方法，检查返回值以及数据库内是否正确增、删、改、查用户信息 |
| DishManager | 调用addDish(), editDish()，showDish()函数，检查返回值以及数据库内是否正确增、删、改、查菜品信息 |
| StoreManager | 调用addUser(), editUser()，showUser()函数，检查返回值以及数据库内是否正确增、删、改、查库存信息 |
| FinaceManager | 调用addIncome()，showIncome()函数，检查返回值数据库内是否正确增、查营业信息 |
| 任  务  管  理 | AccountServlet |  |
| QueueServlet |  |
| LoginServlet |  |
| UrgeServlet |  |
| CancelServlet |  |
|  |  |
| UpdateServlet |  |

用于测试的数据库：

（事先存好了一些用户信息，菜品信息，库存信息等等）

测试用例及其预期结果

|  |  |  |  |
| --- | --- | --- | --- |
| 模块 | 被测试单元 | 详细测试用例（输入值） | 期待值 |
| 人  机  交  互 | User | 默认值  Name=“Tina”, password=“eat123”, type=0  测试赋值函数用例  Name=“Tina”, password=“eat123”, type=0 | 访问函数输出值与预设值相同；  调用复制方法后，打印结果与输入值相同 |
| Customer |  |
| Manager |  |
| Chef |  |
| Servant |  |
| Dish | Name=“辣子鸡”, price=“34”, path= “c:desk/newFolder”, “size”= [null] |
| Desk |  |
| Order |  |  |
| Bill |  |  |
| LoginFrame | UserName = “张三”；password = “”  UserName = “张三”； password = “”（要有不同情况的测试用例） | 账号不满足正则表达式的，不能提交表单，显示修改提示; 满足的能提交并且正确地提交地信息。 |
| MenuFrame | 此面板没有输入操作，只有操作序列  Menu.Exit()  Menu.Submit() |  |
| OrderFrame | Menu.submit() |  |
| BillFrame | Bill.Submit() |  |
| CommentFrame |  |  |
| ServantFrame |  |  |
| ChiefFrame |  |  |
| ManagerFrame |  |  |
| VIPFrame |  |  |
| FinanceFrame |  |  |
| StoreFrame |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 数  据  管  理 | InfoManager | Boolean test1 = InfoManager.link();  Boolean test2 = InfoManager.close();  System.out.print(test1), System.out.print(test2) | True;  True; |
| UserManager | UserManager.add() |  |
| DishManager | DishManager.add()  DishManager.delet()  DishManager.edd(,,) | 菜单数据库增加菜品,且信息同测试输入值: |
| StoreManager |  |  |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 任  务  管  理 | LoginServlet |  |  |
| QueueServlet |  |  |
| LoginServlet |  |  |
| UrgeServlet |  |  |
| CancelServlet |  |  |
| UpdateServelt |  |  |
| accountServlet |  |  |
|  |  |  |

4.2.2 集群测试

本项测试中，通过类的状态图导出测试该类（及与其协作的类）的动态行为，构造操作序列，作为测试用例。

Customer类的状态转换图

pay

cancel

urge

submitOrder

checkMenue

addFood

deleteFood

logout

submitbill

addFood(2)

login

queue

selectDesk

waiting acct

ordering acct

eating acct

accounting acct

seating acct

leaving acct

printBill

withdraw (final)

withdraw

测试用例#s01：

queue·select·login·checkMenu·addFood·deleteDood·addFood·submitOrder·submitBill·pay·logout

测试用例#s02：

queue·select·withdraw(final)

测试用例#s03：

queue·select·login·checkMenue·addFood·submitOrder·addFood(2) ·checkMenu·addFood·submitOrder·submitBill·pay·printBill·logout

测试用例#s04：

queue·select·login·checkMenue·addFood·submitOrder·addFood(2) ·checkMenu·addFood·addFood·deleteFood·withdraw·submitBill ·printBill·pay·logout

测试用例#s05：

queue·select·login·checkMenue·addFood·submitOrder·urge·cancel·submitBill ·printBill·pay·logout

Servant 类的状态转换图

finish

print bill

checkBill

seatCustomer

check desk

logout

login

rest acct

waiting acct

accounting acct

ranking acct

测试用例#s11:

login·selectDesk·seatCustomer·logout

测试用例#s12:

login·checkBill·printBill·finish·logout

测试用例#s13:

login·selectDesk·seatCustomer·selectDesk·seatCustomer·logout

测试用例#s14:

login·selectDesk·seatCustomer·checkBill·printBill·finish·selectDesk·seatCustomer·logout

Chief 类的状态转换图

login

wait acct

busy acct

rest acct

logout

finishFood food

receiveOrder

cancelFood food

receiveCancel food

receiveUrge food

测试用例#s21：

login·receive order·finishFood·logout

测试用例#s22：

login·receiveOrder·finishFood·receiveOrder·finishFood·receiveOrder·finishFood·receiveOrder·finishFood·logout

测试用例#s23：

login·receive order·finishFood·receiveOrder·receiveUrge·finishFood·receive order·finishFood·logout

测试用例#s24：

login·receiveOrder·finishFood·receiveOorder·receiveUrge·receiveCancel·cancelFood·logout

Manager 类的状态转换图

editStore

checkStore

withdraw(2)

checkMenu

withdraw(1)

check user

logout

login

rest acct

waiting acct

work1 acct

accounting acct

withdraw(3)

work3acct

测试用例#s31：

login·checkVIP·addVIP·withdraw(1)·checkVIP·edit(VIP) ·withdraw(1)·logout

测试用例#s32：

login·checkMenu·addDish·deletDish·withdraw(2)·checkMenu·editDish·withdraw(2)·checkMenu ·withdraw(2)·logout

测试用例#s33：

login·checkStore·editStore·withdraw(3)·checkStore·editStore·withdraw(3)·checkStore·withdraw(3)·logout

测试用例#s34：

login·checkFinance·withdraw(4)·logout

测试用例#s35：

login·checkVIP·editVIP·withdraw(1)·checkMenu·editMenu·withdraw(2)·checkStore·editStore·withdraw(3)·checkVIP·withdraw(1)·checkMenu·withdraw(2)·checkStore·withdraw(3)·checkFinance·withdraw(4)·logout

4.2.3 回归测试

本项测试指重新执行已经做过测试的某些单元测试，保证集成测试的模块结合没有带来非预期的副作用。因此，此项目不必重新设计测试用例。

4.2.4 确认测试

预备阶段：复查软件配置。保证软件配置的所有成分都齐全，质量符合要求，文档与程序完全一致，具有完成软件维护所必须的细节，并且已经编号目录。

正式测试，第一阶段进行α测试：邀请系统体验者在系统开发者的指导下进行测试，由开发者记录发现的错误和使用中遇到的问题。（注意，在测试过程中应严格遵照用户指南及其他操作程序。）

确认测试阶段拟采用黑盒测试技术，其用例设计采用等价划分的办法，并重点测试边界值。

|  |  |  |
| --- | --- | --- |
| 项目 | 操作步骤 | 备注 |
| 安装测试 | 下载安装包  运行安装包 | 测试环境:  WindowsXP、Win7、win8、win10 |
| 运行测试 | 按步骤提示激活系统 | 输入正确密钥  输入不同类型错误密钥； |
| 餐厅工作人员随机进行所有工作界面可进行的操作 | 餐厅工作人员(包括服务员\厨师\经理),以及用餐顾客同时进行测试 |
| 用餐顾客随机进行所有服务界面可进行的操作 |
| 卸载测试 | 以管理员身份登陆  激活卸载操作并确认执行 | 测试环境：  WindowsXP、Win7、win8、win10 |

第二阶段进行β测试：发布试用版软件，邀请餐馆用户自行试用，接受错误信息反馈，征求修改意见或建议。测试结束时间为1个月。

## 4.2测试结果记录

4.2.1 单元测试

|  |  |  |  |
| --- | --- | --- | --- |
| 类名 | 成员变量 | 成员方法 | 结论 |
| User、Staff | 完整、  类型正确 | 尚未发现问题 | 可用 |
| Customer  Servant、Manager、Chief | （经初步调试后）  尚未发现问题 |
| Desk、Bill | 完整、类型正确 |  | 可用 |
| LoginFrame | 完整、类型正确 | 个别按钮可点击性在界面跳转前没有修改是否可点击的属性 | 待修改 |
| InfoManager、  UserManager、DishManager | 完整、类型正确 | 缺少链接数据库不成功和修改不成功的异常信息提示 | 待修改 |

4.2.2 集成测试

|  |  |  |
| --- | --- | --- |
| 操作序列编号 | 预期界面跳转 | 负责人 |
| #s01 | 排队选座、登陆、查看菜单、点菜、提交订单、支付并打印 | 张三  李四 |
| #s02 | 排队选座，未登录点击退出，顺利结束 |
| #s03 | 提交订单后，再次加菜，再次提交订单，结账； |
| #s04 | 再次加菜，放弃提交订单，点击退出，顺利结算、支付、打印账单 |
| #s05 | 提交订单后，顺利进行催单操作，顺利进行部分退菜操作，后正确结账并正确打印账单 |
| #s11 | 完成服务员登陆，选座，标记就座，选择退出 | 张三  李四  王五 |
| #s12 | 完成登陆，查账单，打印账单，结束交易，选择退出 |
| #s13 | 完成就座服务后，正确完成结账服务，选择退出 |
| #s14 | 多次交替完成就座、结账服务，选择退出 |
| #s21 | 厨师登陆，接收订单，完成烹饪，选择退出 | 张三  李四  王五 |
| #s22 | 登陆，多次连续完成接单、烹饪服务，选择退出 |
| #s23 | 登陆，在接单后，接收催单，完成烹饪，选择退出 |
| #s24 | 登陆，在接单后，接收催单，接收退菜，取消烹饪（分部分退菜和全部退菜情况），选择退出 |
| #s31 | 经理登陆，查看会员信息、编辑会员信息、退出管理、退出登录 | 张三  王五 |
| #s32 | 登陆，查看菜单信息、编辑菜单信息、退出管理、退出登录 |
| #s33 | 登陆，查看库存信息、修改库存信息、退出管理、退出登录 |
| #s34 | 正确完成：登陆，查看营业信息、退出管理、退出登录 |
| #s35 | 正确完成：登陆，随机交替多次完成管理会员、管理菜单、查看营业功能，退出登录 |

4.2.2 回归测试

(注：本表记录为多次调试后的结果，即原有错误消除后，测试结论随之更改)

|  |  |  |  |
| --- | --- | --- | --- |
|  | 成员变量 | 成员方法 | 结论 |
| User、Staff | 完整、类型正确 | 尚未发现问题 | 可用 |
| Customer | 完整、类型正确 | 尚未发现问题 | 可用 |
| Servant、Manager、Chief | 完整、类型正确 | 尚未发现问题 | 可用 |
| Desk、Order\Bill\Dish |  | 尚未发现问题 | 可用 |
|  |  |  |  |

4.2.4 确认测试

软件配置复查结果

|  |  |  |  |
| --- | --- | --- | --- |
| 文档名称 | 有/无 | 质量描述 | 责任人 |
| 可行性分析 | 有 |  | 张玉洁 |
| 需求说明书 | 有 |  | 李怡甜 |
| 设计说明书 | 有 |  | 田峰源、刘新蕾 |
| 用户使用手册 | 有 |  | 唐婷 |

α测试错误记录：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 错误/意见编码 | 发现时间 | 描述 | 是否处理 | 处理结果 |
|  |  |  |  |  |
|  |  |  |  |  |

β测试错误/意见记录：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 错误/意见编码 | 发现时间 | 描述 | 是否处理 | 处理结果 |
|  |  |  |  |  |
|  |  |  |  |  |

# V. 评估调整

## 5.1 测试评价标准

5.1.1 质量准则

错误率

|  |  |  |
| --- | --- | --- |
| 错误类别 | 描述 | 质量要求 |
| 需求错误 | 功能缺失 | 不存在 |
| 功能错误 | 不存在 |

性能标准:

顾客排队等候人数：10000，完全超过实际需求

1s内提交订单数量上限：

1s内访问数据库查看订单人数上限：

操作响应时间上限：2s

~~可靠性标准：~~

5.1.2 覆盖准则

单元测试用例覆盖单个类的所有操作；

集成测试用例覆盖所有的状态，即操作序列应使得被测类实例遍历所有允许的状态转换；

回归测试中，考虑到重复进行全部单元测试耗费时间太长，只要求测试用例覆盖所有集成测试中进行了修改的单元，以及根据经验可能出错的部分；

## 5.2 测试结束标准

传统地软件工程项目，可以以项目交付时间结束为标准，或者以项目预计资金结束，再或者以错误率控制在一定范围内为结束标准。为了提高本系统的质量，测试结束标准采用联合标准，即在时间、资金都充足的情况下，即使（暂时）认定测试结果完全贴合需求，测试不能结束，应当持续进行仿真测试，尽可能地排除潜在错误。

总而言之，项目测试结束时应当满足如下条件：

1. 达到质量准则要求
2. 项目交付时间已到、计划资金耗尽
3. 若资金耗尽或交付日期将至，但软件质量不合格，应当尽快进行错误率、可用性的再评估。根据评估结果，决定申请追加时间/资金或者放弃项目

# Ⅵ.测试日志

测试日志：