

测试计划书

H&G



2019-5-13

Microsoft

河北师范大学

[1引言 3](#_Toc8630358)

[1.1编写目的 3](#_Toc8630359)

[1.2背景 3](#_Toc8630360)

[1.3范围 3](#_Toc8630361)

[1.3.1总体描述 3](#_Toc8630362)

[1.3.2详细描述 4](#_Toc8630363)

[1.3.3相关风险 4](#_Toc8630364)

[1.4参考资料 4](#_Toc8630365)

[1.5测试提交成果 5](#_Toc8630366)

[1.5.1需求规格说明书（修订版） 5](#_Toc8630367)

[1.5.2概要设计说明书（修订版） 5](#_Toc8630368)

[2 测试策略 5](#_Toc8630369)

[2.1测试类型 5](#_Toc8630370)

[2.1.1 数据和数据库完整性测试 5](#_Toc8630371)

[2.1.2 白盒测试 5](#_Toc8630372)

[2.1.3 功能测试 6](#_Toc8630373)

[2.1.4 UI测试 6](#_Toc8630374)

[2.1.5 性能测试 6](#_Toc8630375)

[2.2 版本发布策略 6](#_Toc8630376)

[2.3 阶段测试策略 6](#_Toc8630377)

[2.4 Bug管理策略 7](#_Toc8630378)

[2.4.1 错误优先级 7](#_Toc8630379)

[2.4.2 严重程度分类 7](#_Toc8630380)

[2.5 内部例会 8](#_Toc8630381)

[3 测试环境 8](#_Toc8630382)

[4 测试工具 8](#_Toc8630383)

[5 测试通过准则 8](#_Toc8630384)

[6 测试人员分配 8](#_Toc8630385)

[6.1 项目总体人员分配 8](#_Toc8630386)

[6.2 各项测试人员分配 9](#_Toc8630387)

# 1引言

## 1.1编写目的

测试文档作为指导此测试项目的基础，帮助我们安排合适的资源和进度避免可能的风险，本文档有助于实现以下目标：

1.确定所有项目的信息和应测试的软件结构

2.列出推荐的测试需求

3.推荐可采用的测试策略，并对这些策略加以详细说明

4.确定所需要的资源，并对测试的工作量进行估计

5.列出测试目的可交付元素，包括用例以及测试报告等。

## 1.2背景

随着时代的发展和社会的进步，以及无线网络技术的迅速发展，有线网络的管理系统越来越不能满足某些行业的需要，在这些行业当中，工作人员经常需要在移动的状态下及时进行数据的采集和查询，所以无线网络管理系统正在逐步应用于各个行业和领域，在H&G系统中，工作人员可以随时随地在移动状态中即可进行业务的处理，大大提高系统的业务处理能力和工作效率，另外餐饮业是人类社会不可缺少的行业之一，虽然营业额在提升，但是竞争越来越激烈，在这种情况下，提高运作效率以及对市场的应变能力尤为重要，于是将移动餐厅管理系统应用于餐饮行业，使餐桌，总台，后厨实现无缝对接，提高服务效率，节省人力资源，通过数据汇总分析，为管理人员改进菜系提供依据，根据消费记录，分析客户喜好，提供个性化服务，提高管理水平，提高竞争力。

## 1.3范围

### 1.3.1总体描述

由于各种活动的相互影响和制约，系统的设计完成中可能存在某些错误，软件测试主要是对H&G系统进行全面检查，及时发现程序中的逻辑错误，以保证产品的正确性和可靠性。

具体结合到系统操作，基本应该测试以下内容：

1）易用性：即人机界面；

2）性能:即检查快速载入和导出数据、检查系统的响应等；

3）功能：即用户在该系统中可以进行的各种操作

4）业务规则：即检查对业务流程的描述是否准确，考虑与目标用户的业务环境是否契合等

5）事务准确性：即确保事务正确完成，确保被取消的事务回滚正确等

6）数据有效与完整性：即检查数据的格式是否正确，确保字符集适当等

7）系统可靠性：即检查程序，数据库的故障处理能力，数据恢复能力等

### 1.3.2详细描述

测试主要分为4个阶段：

1）完善文档。为了保证测试工作的正常运行，首先要完善需求规格说明书和概要设计两个总要的测试参考文档

2）将整个H&G程序按功能分为4个模块（车辆调度；资源管理；数据管理；系统设置），分别对每个模块进行单元测试，在该阶段我们在每个单独的程序块中消除场内的逻辑，功能上的缺陷和错误，保证每个模块作为一个单元能正确执行，并为上一级测试做准备，采用自底向上的方法免去桩模块开发，高质量的程序单元是组成可靠系统的基础，对单元性能的评价有：1，单元接口，2，局部数据结构，3,重要的执行路径测试，4，出错处理路径，5，影响上述几点的边界条件测试

3）进行集成测试，将各个模块进行集中和装配，形成一个完整的软件后就可以进行集成测试，集成测试除了进一步检测和排除子系统结构或相应程序结构上的错误之外，还应该验证所有的系统单元配合是否合适，整体性能和功能是否完整，它采用自底向上集成方法

4）进行系统程序。首先对整个程序进行功能测试，验证软件的功能和性能以及其它特点是否与用户的需求一致。然后进行系统非功能测试，结合本系统特点，侧重于易用性测试，性能测试，强度测试，可靠性测试，可恢复性测试等。可分成表示层测试，业务层测试，数据层测试，全面对其进行测试

### 1.3.3相关风险

本次测试过程中，可能出现的风险如下：

1）bug的修复情况

2）模块功能的实现情况

3）系统整体功能的实现情况

4）代码的编写质量

5）人员经验以及对软件的熟悉度

6）开发人员，测试人员关于项目约定的执行情况

7）人员调整导致研发周期延迟

8）开发时间的缩短导致某些测试计划无法执行

## 1.4参考资料

参考概要设计说明书和需求规格说明书

## 1.5测试提交成果

### 1.5.1需求规格说明书（修订版）

### 1.5.2概要设计说明书（修订版）

1）单元测试报告

2）集成测试报告

3）系统测试报告

4）验收测试报告

# 2 测试策略

## 2.1测试类型

### 2.1.1 数据和数据库完整性测试

概述：数据与数据库完整测试是指测试关系型数据库完整性原则以及数据合理性测试。

数据库完整性原即：

主码完整性：主码不能为空；

外码完整性：外码必须等于对应的主码或者为空。

数据合理性指数据在数据库中的类型，长度，索引等是否建的比较合理。

在项目名称中，数据库和数据库进程应作为一个子系统来进行测试。在测试这些子系统时，不应将测试对象的用户界面用作数据的接口。对于数据库管理系统 (DBMS)，还需要进行深入的研究，以确定可以支持测试的工具和技术。

### 2.1.2 白盒测试

概述：白盒测试是基于代码的测试，测试人员通过阅读程序代码或者通过使用开发工具中的单步调试来判断软件的质量，一般黑盒测试由项目经理在程序员开发中来实现。白盒测试分为动态白盒测试和静态白盒测试 。

分类：

静态白盒测试

利用眼睛，浏览代码，凭借经验，找出代码中的错误或者代码中不符合书写规范的地方。比如，代码规范中规定，函数必须为动宾结构。而黑盒测试发现一个函数定义如下： 。

动态白盒测试

利用开发工具中的调式工具进行测试。比如一段代码有4个分支，输入4组不同的测试数据使4组分支都可以走通而且结果必须正确。

### 2.1.3 功能测试

概述：功能测试指测试软件各个功能模块是否正确，逻辑是否正确。   
对测试对象的功能测试应侧重于所有可直接追踪到用例或业务功能和业务规则的测试需求。这种测试的目标是核实数据的接受、处理和检索是否正确，以及业务规则的实施是否恰当。此类测试基于黑盒技术，该技术通过图形用户界面 (GUI) 与应用程序进行交互，并对交互的输出或结果进行分析，以此来核实应用程序及其内部进程。功能测试的主要参考为类似于功能说明书之类的文档。 比如一个对电子商务系统，前台用户浏览商品-放入购物车-进入结账台，后台处理订单，配货，付款，发货，这一系列流程必须正确无误的走通，不能存在任何的错误。

### 2.1.4 UI测试

概述：UI测试指测试用户界面的风格是否满足客户要求，文字是否正确，页面美工是否好看，文字，图片组合是否完美，背景是否美观，操作是否友好等等   
用户界面 (UI) 测试用于核实用户与软件之间的交互。UI 测试的目标是确保用户界面会通过测试对象的功能来为用户提供相应的访问或浏览功能。另外，UI 测试还可确保 UI 中的对象按照预期的方式运行，并符合公司或行业的标准。包括用户友好性，人性化，易操作性测试。UI测试比较主观，与测试人员的喜好有关

比如：页面基调颜色刺眼；用户登入页面比较难于找到，文字中出现错别字，页面图片范围太广等都属于UI测试中的缺陷，但是这些缺陷都不太严重。

### 2.1.5 性能测试

概述：性能测试主要测试软件测试的性能，包括负载测试，强度测试，数据库容量测试，基准测试以及基准测试

## 2.2 版本发布策略

原则1：当进行首轮测试时，若系统主干功能不能通过冒烟测试，则需要开发组重新发布版本，再对新版本进行首轮测试

原则2：遵循每日构建原则。每日构建工作由测试团队负责，每日发布新的测试版本（Build）并对其进行BVT测试，BVT测试通过后针对该Build进行细测。要求：每个成功的Build都应该通过BVT测试。

原则3：每当完成修复就建立新版本，并进行BVT测试，如果未通过BVT测试（仍存在bug过多或bug级别严重），则可要求重新发布版本，进行第二次BVT测试。

## 2.3 阶段测试策略

针对实际项目情况，测试阶段分为：

1、**单元测试阶段：**由开发人员针对个人负责的单元或模块进行单元测试。通过本阶段后进行下一阶段。

2、**冒烟测试阶段：**验证系统第一版本的关键功能，目的验证该系统版本是否可用，是否能进行具体功能细测，若出现过多限制后续测试的阻塞级别bug，则需要请开发组发布新版本。

3.**界面/易用性测试：**从软件使用的合理性和方便性等角度对软件系统进行检查，来发现软件中不方便用户使用的地方。

4、**初测/细测阶段：**重点验证软件功能是否满足需求，该阶段由测试人员完成。测试成员对功能点进行独立测试，并编写简易的测试用例。同时，在该用例执行中记录该用例执行的状态（通过 / 未执行 / bug ID）

5、**BVT测试阶段：**针对新Build进行版本功能验证，通过后方可进行新功能点的细测阶段。

6、**回归测试阶段：**重点在于验证bug是否解决及相关功能是否受影响。

## 2.4 Bug管理策略

### 2.4.1 错误优先级

立刻级别：必须立即修改；

紧急级别：立即修改（最长时间不得超过2天）；

高级级别：在投入正式运行前修改；

普通级别：在投入正式运行前大部分需修改，未修改的缺陷需进行讨论；

低级别：可延期修改或不修改。

### 2.4.2 严重程度分类

1.严重：

1）用户需求未实现或实现错误（影响到用户完成业务）；

2）导致被测软件响应明显很慢或死机、非法退出、崩溃；

3）导致后台数据受损或丢失。

2.中等：

1）用户需求实现错误（不影响用户完成业务）

2）用户使用不频发的功能，响应时间超出忍耐程度；

3）UI上存在错误引导用户的信息。

3.轻微：

1）UI空间不符合界面规范；

2）影响UI友好性。

## 2.5 内部例会

执行测试阶段开始后，测试组每天上班前进行测试的作战会议。

# 3 测试环境

软件测试环境包含硬件环境和软件环境，硬件换进主要是PC机，软件环境包括软件运行的操作系统 (主流的操作系统：windows、Linux、Unix)，数据库（Oracle、MySQL、SqlServer、DB2等）、web应用服务器（Apache、IIS、tomcat、Nginx等）和集群环境（如负载均衡）

# 4 测试工具

抓包工具Fiddler

缺陷管理工具Jira。

压力测试工具Jmeter

接口测试工具PostMan

远程连接服务器工具XShell

数据库管理工具Navicat

死链测试工具Xenu

虚拟机管理工具VMware

文档整理工具-百度脑图

# 5 测试通过准则

1、软件需求分析说明书中定义的所有功能已全部实现，性能指标全部达到要求。  
2、所有测试项没有残余的一级二级三级的错误。  
3、立项审批表、需求分析文档、设计文档和编码实现一致。  
4、验收测试工件齐全（测试计划，测试用例，测试日志，测试通知单，测试分析报告）

# 6 测试人员分配

## 6.1 项目总体人员分配

|  |  |  |
| --- | --- | --- |
| 角色 | 分配的角色 | 具体职责 |
| 测试人员 | 霍宁肖  王紫薇  王洁 | 制定维护测试计划，设计测试用例与测试进程，搭建测试环境，执行集成测试，功能测试、系统测试、验收测试，提交测试报告 |
| 审查人员 | 刘子璇  李敬单 | 单元测试、监督测试，并协助修改代码 |

## 6.2 各项测试人员分配

|  |  |  |
| --- | --- | --- |
| 任务 | 主要负责人 | 所需工具 |
| 1.需求复审 | 刘子璇 |  |
| 2.单元测试 | 各个模块开发人员 |  |
| 3.集成测试 | 刘子璇 |  |