### 思泉（CRM）客户管理系统测试计划

### **一、目的**

为明确软件需求、安排项目规划与进度、组织软件开发与测试，撰写本文档。确定“思泉（CRM）客户管理系统”的测试环境、测试工具、测试范围，列出测试用例编写的相关约定，确定所需资源并对“思泉（CRM）客户管理系统”的工具进行估计，列出软件系统测试项目可交付元素。

### **二、 概述**

1.背景介绍

　“思泉（CRM）客户管理系统”专为销售人员量身定做，实现如下功能：轻松管理日益增多的客户资源；提供科学的发展潜在客户的步骤，按步照班，渐入佳境；客户的资料和联络记录永远在手，不怕销售人员流动 ；销售机会的进展和跟踪，一切进展尽在掌握；日程安排井井有条，时间分配更科学；人性化事件提醒，重要活动提前准备；通过报表，时时了解销售人员的时间精力都花费在什么客户身上；清楚了解每个客户的购买记录，销售业绩尽在把握；用手机短信功能向客户发布新产品上市、研讨会等信息，客户倍感关怀；统计分析功能帮助您分析客户的分布、行业、类型、来源，为决策提供关键信息；邮寄信封标签打印功能，摆脱繁重的手工劳动；任意条件快速批量提取email地址，节省时间发展更多客户。企业实施“思泉客户管理系统”战略本质目标是与那些有价值的客户建立稳定的长期双赢关系，进而为企业在激烈的市场竞争中赢得优势。

2.测试计划读者范围

测试工程师，开发经理，项目经理，实施负责人

### 3. 测试基本内容

3.1 测试环境

操作系统：Win10

硬件环境

处理器：QEMU  Virtual CPU VERSION  2.29GHz  内存：4G

系统类型：64位操作系统 软件环境：CRM

3.2 测试工具

TAPD

3.3 测试范围

   3.3.1 测试对象

被测系统为CRM3.0版本，使用HTML开发的。

3.3.2 需要测试的特性

本次系统测试要求包含以下业务流程:登陆、客户资源、售后服务、资源服务、费用管理、机会管理、个人设置、市场管理、系统工具、行政办公、基础数据、系统报表、系统管理。

3.3.3不需要测试的特性

本次系统测试不需要包含的内容： 上述业务流程(3.3.2)之外的所有业务流程。

4. 测试用例设计

4.1 测试用例相关约定

在设计测试用例时，你需要定义程序的操作来确保程序的各方面都被测试到。为了确保清楚，准确的捕获到了完成一个操作所需要的所有行为，要满足下面条件：

1) 测试用例的目标清楚，并能满足软件质量的各个方面，包括功能测试、性能测试、安全性测试、故障转移测试、负载测试等。

2) 设计思路正确、清晰。例如，通过序列图、状态图、工作流程图、数据流程图等来描述待测试的功能特性或非功能特性。

3) 在组织和分类上，测试用例层次清楚、结构合理。测试用例的层次与产品特性的结构/层次相一致，或者与测试的目标/子目标的分类/层次相一致，并具有合理的优先级或执行顺序。

4) 测试用例覆盖所有测试点、覆盖所有已知的用户使用场景(User scenario)，也就是说每个测试点都有相应数量的测试用例来覆盖，而且将各种用户使用场景通过矩阵或因果图等方式列出来，找到相对应的测试用例。

5) 测试手段的区别对待。在设计测试用例时，就要全面考量测试的手段，哪些方面可以通过工具测试，哪些方面不得不用手工测试，对不同手段的测试用例区别对待。

6) 有充分的负面测试。作为测试用例，不仅要测试正确的输入和操作，还要测试各种各样的例外情况，如边界条件、不正确的操作、错误的数据输入等。

7) 没有重复、冗余的测试用例，满足相应的行业标准等。

4.2 衡量测试用例设计的质量标准

4.2.1系统性

1) 对于系统业务流程要能够完整说明整个系统的业务需求、系统由几个子系统组成以及它们之间的关系；

2) 对于模块业务流程要能够说明清楚子系统内部功能、重要功能点以及它们之间的关系；

4.2.2连贯性

1) 对于系统业务流程来说，各个子系统之间是如何连接在一起，如果需要接口，各个子系统之间是否有正确的接口；如果是依靠页面链接，页面链接是否正确；

2) 对于模块业务流程来说，同级模块以及上下级模块是如何构成一个子系统，其内部功能接口是否连贯；

4.2.3相关性

1) 考虑各个产品之间的相关性，当某个产品某个页面的字段发生增删改时，其它产品是否有相应变化，和后台数据库之间是否匹配

2) 当某个产品增加某个功能时，其它相关产品是否有相应措施  4.2.4全面性

1) 应尽可能覆盖程序的各种路径

2) 应尽可能覆盖系统的各个业务

3) 应考虑存在跨年、跨月的数据

4) 大量数据并发测试的准备

5) 系统中各功能、业务的异常情况

4.2.5.正确性

1) 输入用户实际数据以验证系统是否满足需求规格说明书的需求。 2) 测试用例中的测试点应保证至少覆盖需求规格说明书中的各项功能。

4.2.6符合正常业务惯例

1) 测试数据应符合用户实际工作业务流程

2) 兼顾各种业务变化的可能

3) 要符合当前业务行业法律，法规。

4.2.7容错性（健壮性）

1) 程序能够接收正确数据输入并且产生正确（预期）的输出，输入非法数据（非法类型、不符合要求的数据、溢出数据等），程序应能给出提示并进行相应处理。

2) 在设计测试用例时，你需要定义程序的操作来确保程序的各方面都被测试到。为了确保清楚，准确的捕获到了完成一个操作所需要的所有行为，要满足下面条件：每一步都用主动语态书写，使用主动语态的好处是使得测试执行人员 。

5.实施计划

5.1 测试进度安排

本次测试的时间安排如下：

/里程碑/ /执行者/

测试用例设计  朱元伟、熊熬、任俊鹏、熊崇康、李深蓝

用例评审 朱元伟

测试执行 朱元伟、熊熬、任俊鹏、熊崇康、李深蓝

系统测试总结 朱元伟

5.2 测试人员安排以及职责

人员  /角色/  /职责、任务/

朱元伟 测试组长

熊熬 测试人员

任俊鹏 测试人员

熊崇康 测试人员

李深蓝 测试人员

具体工作：系统测试、需求分析，设计测试用例，导入测试用例，执行测试，记录测试，执行日志，缺陷追踪；

5.3 输出要求

《测试计划》、《测试用例》、《测试数据》、《测试缺陷报告》、《测试总结报告》

6 测试方法

本次测试是CRM的系统测试，确保：

6.1黑盒测试方法

6.1.1等价类划分法

将所有可能的输入数据（有效的和无效的）划分成若干个等价类。 6.1.2边界值分析法

指对输入的边界条件进行分析，设计出针对边界值的测试用例。

6.1.3因果图法

就是利用图解法分析软件输入(原因)和输出条件(结果)之间的关系，以设计测试用例的方法。因果图法适合于检查程序输入条件的多种情况的组合，并最终生成判定表，来获得对应的测试用例。

6.1.4功能图法

功能图是描述程序状态变化、转移的过程，因为软件运行或操作的过程可以看作是其状态不断发生变化的过程。测试用例的设计就是如何覆盖所有软件表现出来的状态，即在满足输入/输出的一组条件下，软件运行是一系列有次序的、受控制的状态变化过程。

6.1.5错误推测法

推测法主要依赖经验、直觉来作出简单的判断甚至是猜测，给出可能存在缺陷的条件、场景等，在找到缺陷后，设计出相应的测试用例。

6.1.6正交实验设计方法 主要步骤是：

1) 对软件需求规格说明中的功能要求进行划分(层层分解与展开)，分解成具体的、相对独立的基本功能。

2) 根据基本功能的质量需求，找出影响其功能实现的操作对象和外部因素，每个因素的取值可以看作水平，多个取值就存在多个水平。

3) 确定待测试软件中所有因素及其权值，这是测试用例设计的关键，确保全面、准确。权值是依据各因素的影响范围、发生的频率和质量的需求来确定的。

4) 加权筛选，生成因素分析表。

5) 利用正交表构造测试数据集，正交表的每一行，就是一条测试用例。考虑交互作用不可忽略的处理因素和不可混杂的原则，有交互作用的组合优先安排。

6) 利用正交实验设计方法设计测试用例，可控制生成的测试用例数量，覆盖率高且测试效率高。

6.1.7接口间测试

测试各个模块相互间的协调和通信情况，数据输入输出的一致性和正确性。

6.1.8数据库测试

依据数据库设计规范对软件系统的数据库结构、数据表及其之间的数据调用关系进行测试。

6.1.9可理解（操作）性

理解和使用该系统的难易程度（界面友好性）。

6.1.10可移植性

在不同操作系统及硬件配置情况下的运行性。

6.2软件测试的一些准则

软件测试从不同的角度出发会派生出两种不同的测试原则，从用户的角度出发，就是希望通过软件测试能充分暴露软件中存在的问题和缺陷，从而考虑是否可以接受该产品，从开发者的角度出发，就是希望测试能表明软件产品不存在错误，已经正确地实现了用户的需求，确立人们对软件质量的信心。为了达到上述的原则，那么需要注意以下几点：

1．应当把“尽早和不断的测试”作为开发者的座右铭

2．程序员应该避免检查自己的程序，测试工作应该由独立的专业的软件测试机构来完。

3．设计测试用例时应该考虑到合法的输入和不合法的输入以及各种边界条件，特殊情况要制造极端状态和意外状态，比如网络异常中断、电源断电等情况。

1. 一定要注意测试中的错误集中发生现象，这和程序员的编程水平和习惯有很大的关系。

5．对测试错误结果一定要有一个确认的过程，一般有A测试出来的错误，一定要有一个B来确认，严重的错误可以召开评审会进行讨论和分析。

6．制定严格的测试计划，并把测试时间安排的尽量宽松，不要希望在极短的时间内完成一个高水平的测试。

7．回归测试的关联性一定要引起充分的注意，修改一个错误而引起更多的错误出现的现象并不少见。

8．妥善保存一切测试过程文档，意义是不言而喻的，测试的重现性往往要靠测试文档。

要制造极端状态和意外状态，比如网络异常中断、电源断电等情况。

### 7. 测试的各项标准

7.1 测试项通过/失败的标准

一般有“基于测试用例”和“基于缺陷密度”两种评比准则，在这里我们采用前者。 准则如下：

 功能性测试用例通过率达到95％； 

非功能性测试用例通过率达到90％； 

沒有高于优先级 3以上的缺陷。 

备选通过办法：

根据实际情况由软件开发部门的经理、项目经理和测试负责人等共同讨论确定本阶段是否结束。

7.2 中断测试和恢复测试的判断标准

缺陷数量大于100时中断测试直至缺陷修复到10时恢复；

当代码不全时停止测试直至代码全面恢复测试；

当缺陷严重程度为4的个数超过总体缺陷的1/2时停止测试；  当缺陷优先级为1的个数超过总体缺陷1/3时停止测试；

### 8. 缺陷跟踪

8.1 缺陷类型

本次测试过程中缺陷的管理将在ALM中进行，缺陷大致包含如下状态：

缺陷类型 具体含义

冗余代码多，代码冗余，即是编程时不必要的代码段。

兼容性差，软件从某一环境转移到另一环境后不能正常运行，可操作性差，软件难以理解,不容易使用,运行缓慢。界面不友好 最终用户会认为界面不好。

与需求不一致 软件没有实现产品规格说明所要求的功能模块; 软件实现了产品规格说明没有提到的功能模块。可扩展性差，软件在原有的功能上不容易实现新增其他新的功能。

### 8.2缺陷严重程度和优先等级

缺陷严重程度： 严重级别

严重程度描述

## 1-Low

使用不方便的问题 对软件的改进建议：

1. 容易给用户误解和歧义的提示 ；

2) 界面需要改进的 ；

3) 对有疑虑的文档，提出修改建议

2-Medium

界面非关键信息错误

•微小的错误，不会影响系统的功能

•风格不统一，包括相近流程的界面布局相异，相同的问题点提示信息相异，但对用户的使用方法和使用习惯不造成影响（需求中明确的风格要求除外）如帮助、提示信息不完整，有错误，但不影响用户使用。 不正确的，但有使系统使用起来不太方便的错误：

1. 系统的提示语不明确，不简明

2) 滚动条无效

3) 可编辑区和不可编辑区不明显

4) 光标跳转设置不好，鼠标（光标）定位错误

5) 上下翻页，首尾页定位错误

6) 界面不一致，或界面不正确

7) 日期或时间初始值错误（起止日期、时间没有限定）

8) 按钮或标签上有拼写错误的单词、不正确的大小写

该问题是一个不准确或容易误解的行为，但不会引起下面（3、4、5级别）列出的问题

## 3-High

功能缺失或错误，界面关键信息错误

•该问题增加了安装、测试或用户操作的复杂度或成本 •该问题轻微降低了系统的性能，但系统仍然能工作

•非核心功能实现不完整或不正确，但对系统影响很小，系统仍然能工作 •业务流程对应的功能未实现，但是有替代方法解决，不影响实际的使用 •部署文档描述不明确，增加部署难度 不正确的，但不会影响系统稳定性的：

1. 过程调用或其它脚本错误

2) 系统刷新错误

3) 产生错误结果，如计算结果错误等

4) 功能的实现有问题。如在系统实现的界面上，一些可接受输入的控件

点击后无作用，对数据库的操作不能正确实现

1. 编码时数据类型、长度定义错误的
2. 对用户的使用有操作顺序上的限制
3. 虽然正确性不受影响，但系统性能和响应时间受到影响

## 4-Very High

导致系统崩溃、数据丢失、严重系统资源泄露，关键功能缺失或错误

•该问题会严重降低系统的性能 •业务流程不正确

•需求实现不完整，设计实现上的缺陷，且无替代方法，如：设计了3条路上山，但是实际只有一条可以上

•该问题不符合需求规格书

•配置项设计错误，无法正常配置，或配置后，测试中出现与配置相关的错误

•部署文档错误，导致部署失败

•与其它网元的接口，调用或提供错误

•申报信息提交错误，可继续测试（如联网申报、分类错误、乱码、违禁信息），但影响应用后续审核上线；

## 5-Urgent

必须马上解决的，根据情况可以要求项目组立刻发布新版本，阻碍流程、系统崩溃导致开发或测试无法进行或程序无法正常运行的缺陷。提交物缺失，导致测试、部署和维护无法正常进行

•需求未实现

•正常的操作，导致系统（进程）崩溃 •系统不能启动或启动后无法正常工作

•系统（进程）经常自动崩溃（至少一天一次）

8.3缺陷的优先级：

优先级 优先级描述

1. Low 可能会修复，但是也能不修复
2. 2-Medium 如果时间允许应该修复
3. 3-High 在产品发布前必须修复
4. Very High 尽快修复
5. 5-Urgent 立即修复，停止进一步测试