# 功能测试方案文档

# 1．简介

## 1.1目的

“开心农场”这一“测试方案”文档有助于实现以下目标：

1. 确定现有项目的信息和应测试的软件构件。
2. 列出推荐的测试需求（高级需求）。
3. 推荐可采用的测试策略，并对这些策略加以说明。
4. 确定所需的资源，并对测试的工作量进行估计。
5. 列出测试项目的可交付元素。

## 1.2背景

随着互联网技术的发展，我国“共享产业”迎来了发展机遇期,回首过去，在中国这片传统的农业大国里，长期存在着生产管理落后、农产品质量监管难、农产品销售难等难题，互联网+的春风来临农业的发展将迎来新的契机。农场管理水平低下，是农场不赚钱的主要原因，实现农场智能化管理，以及土地的智能化管理，可以有效的解决这些问题。除此之外，目前我国农村大量青壮年流失，使得很多土体被荒废，而居住在环境拥挤、竞争激烈的城市的白领又向往农村休闲、安逸的生活；社会需求是技术发展最好的驱动力，在这供、需双重机制的作用下，共享农场应运而生。兰州市某农业企业想实现生产经营模式转型、降低劳务成本，实现土地资源利润最大化。鉴于此我们小组想要开发一款互联网农业管理系统，该系统集线上租地种植、土地托管种植、农业认养、农业电商、农产品营销功能等为一体。

待开发的软件系统名称：开心农场信息系统

1. 本项目任务提出者：软件工程教师代祖华
2. 开发者：狗蛋带仨妞小组
3. 小组成员：杨雪雪（PM),王国伟，张琳，王英雪
4. 用户：农户、城市居民
5. 实现该软件的计算中心或计算机网络：西北师范大学局域网
6. 该软件系统同其他系统或机构的基本的相互来往关系：该系统由狗蛋带仨妞小组全权负责

## 1.3范围

描述测试的各个阶段（例如，单元测试、集成测试或系统测试），并说明本计划所针对的测试类型（如功能测试或性能测试）。

简要地列出测试对象中将接受测试或将不接受测试的那些性能和功能。

如果在编写此文档的过程中做出的某些假设可能会影响测试设计、开发或实施，则列出所有这些假设。

列出可能会影响测试设计、开发或实施的所有风险或意外事件。

列出可能会影响测试设计、开发或实施的所有约束。

# 2. 测试参考文档和测试提交文档

## 2.1测试参考文档

下表列出了制定测试计划时所使用的文档，并标明了各文档的可用性：

注：可适当地删除或添加文档项。





## 2.2测试提交文档

功能测试方案文档

# 3.测试进度

测试活动：测试开心农场信息系统数据，界面以及跳转状态等。

计划开始日期：2020.06.26

实际开始日期：2020.06.26

结束日期：2020.06.30

制定测试计划：我们小组将从以下几个方面进行此次项目的测试：

系统测试：即将已经集成好的软件系统，作为整个计算机系统的一个元素，与计算机硬件、外设、某些支持软件、数据和人员等其它系统元素结合在一起，在实际运行（使用）环境下，对计算机系统进行一系列测试活动。

性能测试：通过自动化的测试工具模拟多种正常、峰值以及异常负载条件来对系统的各项性能指标进行测试。

安全测试：安全测试检测系统对非法入侵的防范能力。

对测试进行评估：测试要考虑到各方面会出现的问题进行一一测试。

产品发布：测试完成后考虑产品发布。

# 4.测试资源

## 4.1人力资源

测试组组长：组织测试；

测试分析员：负责设计和实现测试脚本和测试用例；

测试者：负责执行测试脚本中记录的测试用例。

## 4.2测试环境

|  |  |
| --- | --- |
| 操作系统 | Windows 10 |
| 开发工具 | IntelliJ IDEA |
| 数据库 | Mysql 5.7 |
| JDK | 1.8 |
| Web服务器 | Tomcat 8.0 |
| 单元测试框架 | Junit4 |
| Web网页设计 | HBuilder Sublime3 |

# 5.系统风险、优先级

## 5.1软件测试风险

软件测试是一项存在风险的工作，它是不可避免的，总是存在的。作为一名测试管理人员必须在平时的工作中，分析这些风险的类别，并且找出对策尽最大程度的降低这些风险。

（1）需求变更风险，应对之策：多和有决策权的核心用户多沟通，争取更充分的研发时间和测试时间。

（2）人员风险，应对之策：对于核心的测试人员可能离职而延误测试的情况，作为测试管理者可以在平时给这些核心人员配置一些可以候补的测试人员来向他们学习，以避免这些核心人的请假、离职的时候，可以立即补充上来。另外，对于一些关键的业务和技术一定要有文档。

（3）代码质量的风险，应对之策：对于程序员的提交给测试部门的代码一定要在前期做好充足的单元测试、对于核心模块的代码一定要有资深的研发工程师进行前期检查。

（4）测试深度和广度的风险，应对之策：测试工程师在写测试用例的时候尽量提高测试用例的覆盖率（在测试用例完成后组织进行测试用例的评审工作），如果测试用例能覆盖不同的用户千变万化的操作最好。特别是一些边界值、深层次的逻辑关系等。以及用户实际使用环境下的场景（比如大用户量的并发操作等）。

## 5.2测试用例的优先级：

（1）小版本确认测试（Build Verification Tests (BVTs)：也叫做“冒烟测试”，一组你想先运行的以确定这个给出的小版本是否可以测试的测试用例。如果你不能访问每一个功能区域或执行其他测试用例依赖的基本操作，那么在执行这个优先的测试用例之前，试图做其他任何的测试都是没有意义的，因为他们大多数肯定要失败。  
（2）高（Highs）：最常执行以保证功能性是稳定的，目标的行为和能力可以正常的工作，和重要的错误和边界被测试的测试用例的集合。  
（3）中（Mediums）：这是使给出的功能区域或功能变得更详细，检查功能的多数方面包括边界，错误和配置测试的测试用例  
（4）低（Lows）：这是通常最少被执行的测试用例。但这并不意味着这些测试都不重要，只是说他们在项目的生命期间里不是常常被运行，例如GUI，错误信息，可用性，压力和性能测试。

# 6.测试策略

## 6.1链接测试

　　链接是**Web** 网站的一个主要特征，它是在页面之间切换和引导用户去一些未知地址页面的主要手段，链接测试的内容：

　　（1）测试所有链接是否按需求链接到了应该链接的页面；

　　（2）测试所链接的页面是否存在；

　　（3）保证Web 网站上没有孤立的页面。

　　所谓孤立页面是指没有链接指向该页面，只有知道正确的URL 地址才能访问。

　　链接测试可以手动进行，也可以自动进行。链接测试必须在集成测试阶段完成，也就是说，在整个Web 网站的所有页面开发完成之后进行链接测试。常用测试工具有Xenu（测试链接的正确性的工具）

## 6.2表单测试

　　表单就是一些需要在线显示和填写的表格。表单有一些标准操作，如确认、保存、提交等。

　　主要测试表单的正确性和规范性，是否适合常用表单的使用习惯；主要测试方法为：边界值测试、等价类测试，以及异常类测试。

## 6.3 Cookies测试

　　Cookie是一个由网页服务器放在您硬盘上的非常小的文本文件，它本质上就像您的身份证明一样，并且不能像代码那样被执行或被用来散布病毒。它只能被您使用并且只能由提供的服务器读取。

　　使用cookies的目的：帮您节约时间。如果您自定义页面，或注册产品或服务。cookie记住您的身份，当下一次您再次访问的时候，将显示您需要的信息，将帮您填入任何您已经回答过的问题。

　　Cookies测试内容：通常用来存储用户信息和用户在某些应用系统上的操作序列，当一个用户使用Cookies访问了某一个应用系统时，Web 服务器将发送关于用户的信息，并把该信息以Cookies 的形式存储在客户端计算机上，这可用来创建动态和自定义页面或者存储登录等信息。

　　（1）Cookies是否能正常**工作**；

　　（2）Cookies是否按预定的时间进行保存；

　　（3）刷新对Cookies 有什么影响等。

## 6.4数据库测试

　　（1）数据校验：根据业务规则，需要对用户输入进行校验，则要保证这些校验功能正常工作。

　　一般测试数据的一致性错误和输出错误。

　　（2）数据一致性错误：主要是由于用户提交的表单信息不正确而造成的；

　　（3）输出错误：主要是由于网络速度或程序设计问题等引起的

## 6.4可用性测试

　　可用性/易用性方面的测试一般采用手工测试的方法进行评判。

## 6.5导航测试

　　导航测试是导航描述了用户在一个页面内操作的方式，在不同的用户接口控制之间，例如按钮、对话框、列表和窗口等；或在不同的连接页面之间。常采用手工对网页进行浏览、根据一般用户的浏览习惯来进行评判。

　　导航测试的内容：

　　（1）导航是否直观

　　（2）Web 系统主要部分是否可以通过主页访问

　　（3）Web系统是否需要站点地图、搜索引擎或其他的导航器帮助

　　（4）是否缺少返回上一目录的导航功能（虽然可以通过直接点击来实现，但是加入这个功能会更方便，因为大多数用户查找问题或文档时都是先查找同一个目录）

　　（5）导航条、菜单、连接的风格是否一致

　　（6）各种提示是否准确，确保用户凭直觉就知道是否还有内容，内容在什么地方。一般最好让最终用户参与导航测试，效果将更加明显。

## 6.6图形测试

　　一个Web 网站的图形可以包括图片、动画、边框、颜色、字体、背景、按钮等。图形测试是网页美观测试的一部分，常采用手工测试。

　　图形测试的内容：

　　（1） 要确保图形有明确的用途，图片或动画不要胡乱地堆在一起，以免浪费传输时间。图片尺寸要尽量地小，并且要能清楚地说明某件事情

　　（2）验证所有页面字体的风格是否一致

　　（3）背景颜色应该与字体颜色和前景颜色相搭配

　　（4）图片的大小和质量也是一个很重要的因素，一般采用JPG 或GIF 压缩

## 6.7内容测试

　　内容测试用来检验web网站系统提供信息的正确性、准确性和相关性。如文字标题是否与文字内容符合，是否存在不需要的文字，是否有相应的操作提示信息（成功、失败、不符合校验等）

　　常采用界面浏览的方式。

## 6.8整体界面测试

　　测试整个网站系统的页面结构设计是否符合用户需求规范。一般常采用界面浏览的方式，最好是有最终用户的参与。例如，当用户浏览Web 网站时，应考虑是否感到舒适？是否凭直觉就知道要找的信息在什么地方？整个Web 应用系统的设计风格是否一致？

## 6.9平台测试

　　采用不同的**操作系统**平台对网站进行测试。最常见的有**Windows**、Unix、Macintosh、**Linux** 等。Web 网站的最终用户究竟使用哪一种操作系统，取决于用户系统的配置。

　　同一个应用可能在某些操作系统下能正常运行，但在另外的操作系统下可能会运行失败。因此，在Web 系统发布之前，需要在各种操作系统下对Web 系统进行兼容性测试。

## 6.10浏览器测试

　　使用不同的浏览器对网站进行浏览测试，查看网站在不同浏览器中的兼容性问题。浏览器是Web系统客户端最核心的软件，来自不同厂商的浏览器对Java，、JavaScript、ActiveX、plug-ins 或不同的HTML 有不同的支持。

　　另外，框架和层次结构风格在不同的浏览器中也有不同的显示，甚至根本不能显示。不同的浏览器对安全性和Java 的设置也不一样。

## 6.11分辨率测试

　　对屏幕的分辨率进行调节来查看网站在不同分辨率下的显示效果，比如；分辨率低时界面文字显示太大，而分辨率高时又有些文字显示时太小。

　　页面版式在640x400、600x800 或1024x768 的分辨率模式下是否显示正常?

　　最后补充一点：测试时注意多用F12调试，有些错误界面中没有，但是调试时会出现。