{虚拟宠物医院学习系统}测试计划

|  |  |  |
| --- | --- | --- |
| 文件状态：  [ ] 草稿  [ ] 正式发布  [√] 正在修改 | 文件标识： |  |
| 当前版本： | X.0.2 |
| 作 者： | 李可欣 |
| 完成日期： | 2018-03-15 |

版 本 历 史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本/状态 | 作者 | 参与者 | 起止日期 | 备注 |
| V0.1 | 李可欣 |  | 2018/3/8-2018/3/8 |  |
| V0.2 | 李可欣 |  | 2018/3/8-2018/3/15 |  |
|  |  |  |  |  |

目录

[1 概述 1](#_Toc508883920)

[1.1 目的 1](#_Toc508883921)

[1.2 背景 1](#_Toc508883922)

[1.3 范围 1](#_Toc508883923)

[2 测试交付件 2](#_Toc508883924)

[3 测试方法 2](#_Toc508883925)

[3.1 测试对象 2](#_Toc508883926)

[3.2 测试目的 2](#_Toc508883927)

[3.3 测试设计方法 3](#_Toc508883928)

[3.3.1手工测试 3](#_Toc508883929)

[3.3.2接口自动化 5](#_Toc508883930)

[4 被测特性 6](#_Toc508883931)

[5 不被测特性 6](#_Toc508883932)

[6 过程准则 6](#_Toc508883933)

[6.1 入口准则 6](#_Toc508883934)

[6.2 停止准则 6](#_Toc508883935)

[6.3 受阻准则 6](#_Toc508883936)

[6.4 恢复准则 6](#_Toc508883937)

[7 测试进度 7](#_Toc508883938)

[8 测试资源 7](#_Toc508883939)

[8.1 人力资源 7](#_Toc508883940)

[8.2 测试环境 7](#_Toc508883941)

[8.3 测试工具 8](#_Toc508883942)

[9 测试设计说明 8](#_Toc508883943)

[10 附件：测试用例 8](#_Toc508883944)

# 概述

## 目的

<虚拟宠物医院学习系统>功能测试计划有助于实现以下目标：了解软件的具体功能，在功能测试阶段，对软件的功能、性能、接口、数据结构等功能的具体测试结果与预期的要求进行分析，为完善及改进软件的功能提供依据。测试过程中使用手工测试及自动化Test Case的编写对系统进行测试，对访问量较大接口做压力测试等性能测试，从需求评审阶段测试人员需进入项目，最终交付功能完整且性能良好的软件。

## 背景

1. 待开发系统名称：虚拟宠物医院学习系统；
2. 本项目的任务提出者是软件开发实践二的老师，由本小组负责开发
3. 测试环境：本文档负责测试模块为虚拟宠物医院学习系统前台测试计划，实现前台用户在本系统中各种子功能，通过本软件，提高软件工程发现问题、分析问题、提出问题、解决问题的能力，同时增强对各种开发软件的使用能力

## 范围

读者对象：系统测试人员、开发人员

适用范围：

1. 测试目标：在虚拟宠物医院学习系统前台模块的功能测试中，需要测试用户注册/登录、角色扮演、职能学习、在线测试等四个主要功能模块
2. 系统性能指标要求如下：
3. 系统支持的在线用户数不低于500
4. 注册/登录、角色扮演、职能学习、在线测试等模块操作的平均响应时间不超过3s
5. 软硬件环境要求：
6. 系统可运行在Windows和Mac电脑上
7. 系统采用B/S架构，支持谷歌、Safari等浏览器的访问
8. 系统数据库使用MySQL
9. 界面需求：
10. 界面布局

a.屏幕不能拥挤

  \*  屏幕总体覆盖度不应该超过40％，而分组覆盖度不应该超过62％。

  \*  整个项目，采用统一的控件间距，通过调整窗体大小达到一致，即使在窗体大小不变的情况下，宁可留空部分区域，也不要破坏控件间的行间距。

b.控件按区域排列

  \*  一行控件纵向中对齐, 控件间距基本保持一致，行与行之间间距相同，靠窗体的控件距窗体边缘的距离应大于行间距。

c.有效组合

逻辑上相关联的控件应当加以组合以表示其关联性，反之，任何不相关的项目应当分隔开。在项目集合间用间隔对其进行分组，或者使用方框划分各自区域。

d.窗口缩放时，控件位置、布局

\*  固定窗口大小，不允许改变尺寸

  \*  改变尺寸的窗口，在窗口尺寸发生变化时控件的位置、大小做出相应的改变。

  \*  改变尺寸的窗口，在窗口改变尺寸时增加相应在的纵向、横向滚动条，以方便用户使用窗体上的控件。

1. 界面颜色搭配

a.统一色调

b.与操作系统统一，读取系统标准色表

c.遵循对比原则

e.颜色方案因显示器、显卡、操作系统等原因显示出不同的色彩

f.针对色盲、色弱用户，可以使用特殊指示符

# 测试交付件

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **交付件名称** | **负责人** | **交付日期** |
| 1 | 功能测试计划 | 李可欣 | 2018/3/8-2018/3/15 |
| 2 | 功能测试脚本 | 李可欣 | 2018/3/15-2018/3/25 |
| 3 | 功能测试用例 | 李可欣 | 2018/3/8-2018/3/15 |
| 4 | 测试记录缺陷 | 李可欣 | 2018/3/30-2018/4/20 |
| 5 | 测试报告 | 李可欣 | 2018/4/20-2018/4/25 |

# 测试方法

## 测试对象

虚拟宠物医院学习系统前台功能

## 测试目的

交付功能完整且性能良好的针对前台用户的软件

## 测试设计方法

3.3.1手工测试

手工测试——对于重复性较小、逻辑关联较少的功能点，在功能测试的时候采用手工测试的方法。手工测试的好处在于，测试用例编写具有较大的灵活性，可以对同一功能点进行边界值、非法字符、数值校验、逻辑值确定、数据幂等等多方面的测试。手工测试可以有效地发现单点功能缺陷，并且对于系统隐藏的逻辑错误有很大帮助。本次测试用例的设计采用准则为以下：

测试用例编写原则：

1. 黑盒测试用例设计：

* 等价类划分法：把所有可能输入的数据，即程序的输入域划分成若干子集，然后从每一个子集中选取少数具有代表性的数据作为测试用例。
* 边界值分析法：对输入或输出的边界值进行测试。作为等价类划分的补充，其测试用例来自等价类的边界。
* 错误推测法：基于经验和直接推测程序中可能存在的各种错误，从而有针对性的设计测试用例。列举出程序中所有可能的错误和容易发生错误的特殊情况，根据他们选择测试用例。
* 因果图法：利用图解法分析输入的各种组合情况，从而设计测试用例的方法，适合于检查程序输入条件的各种组合情况。
* 正交试验法：从大量的数据中挑选适量的、有代表性的点，从而合理地安排测试。
* 功能图法：由状态迁移图和布尔函数组成。状态迁移图用状态和迁移来描述，一个状态指出数据输入的位置或时间，而迁移则指明状态的改变，同时要依靠判定表或因果图表示的逻辑功能。
* 场景图法：事件触发时的情景便形成了场景，而同一事件不同的触发顺序和处理结果就形成事件流。

1. 白盒测试用例设计：

* 代码检查法：包括桌面检查、代码审查和走查等，主要检查代码和设计的一致性，代码对标准的遵循、可读性、代码逻辑表达的正确性、代码结构的合理性等方面；发现违背程序编写标准的问题，程序中不安全、不明确和模糊的部分，找出程序中不可移植部分、违背程序编程风格的内容。包括变量检查、命名和类型审查、程序逻辑审查、程序语法检查和程序结构检查等内容。
* 静态结构分析法：程序的结构形式是白盒测试的主要依据。通过使用测试工具分析程序源代码的系统结构、数据结构、内部控制逻辑等内部结构，生成函数调用关系图、模块控制流图、内部文件调用关系图、子程序表、宏和函数参数表等各类图形图标，可以清晰地标识整个软件系统的组成结构，使其便于阅读和理解，然后可以通过分析这些图标，检查软件有没有存在缺陷或错误。
* 静态质量度量法：根据ISO/IEC 9126质量模型作为基础，构造质量度量模型，用于评估软件的各个方面。该模型从上到下分成3层：质量因素（Factors）、分类标准（Criteria）和度量规则（metrics）。度量规则使用了代码行数、注释频度等参数度量软件的各种行为属性；软件的可维护性采用以下四个分类标准：可分析性、可修改性、稳定性、可测性。每个分类标准由一系列度量规则组成，各个规则分配一个权重，有规则的取值与权重值计算出每个分类标准的取值；质量因素根据各分类标准组合权重方法计算。
* 逻辑覆盖法：逻辑覆盖是以程序内部的逻辑结构为基础的设计测试用例的技术。分为语句覆盖（SC）、判定覆盖（DC）、条件覆盖（CC）、条件/判定覆盖（CC）、条件组合覆盖（MCC）、修正判定条件覆盖（MCDC）、点覆盖、边覆盖、路径覆盖
* 基本路径覆盖测试法：基本路径覆盖测试法是在程序控制流图的基础上，通过分析控制构造的环路复杂性，导出基本可执行路径集合，从而设计出测试用例的方法。设计出的测试用例要保证在测试中程序的语句覆盖100%，条件覆盖100%。
* 域测试：域测试是一种基于程序结构的测试方法，基于对程序输入空间（域）的分析，选择测试点进行测试。
* 符号测试：符号测试的基本思想是允许程序的输入不仅仅是具体的数值数据，而且包括符号值，符号值可以是基本的符号变量值，也可以是符号变量值的表达式。

1. 接口测试用例设计：

首先后台结构的基本分层结构：外界请求/调用服务🡪接口层🡪业务逻辑层🡪数据存取层🡪数据存取层🡪Data。接口测试是测试系统组件间接口的一种测试。接口测试主要用于检测外部系统与系统之间以及内部各个子系统之间的交互点。测试的重点是要检查数据的交换，传递和控制管理过程，以及系统间的相互逻辑依赖关系等。

覆盖要求：接口层逻辑、业务逻辑、数据存取逻辑

正确性要求：响应结果、持久化数据存取正确、并发控制正确性、业务处理原子性、业务处理的防重正确性

1. 接口层：
2. 接口可用性：比如服务是不是通的，请求服务会不会timeout，或返回一些常见的错误码，比如http404，500等；
3. 是否符合接口定义：比如请求参数可选，必选要求，返回结果参数集合是否正确，格式是否正确；
4. 接口参数取值是否符合业务定义
5. 业务逻辑：
6. 业务流程
7. 业务处理的原子性
8. 业务处理的并发控制
9. 业务处理的防重控制
10. 数据存取
11. 数据库连接是否正常：连接参数是否正确
12. 业务数据是否写入到DB：比如有分库分表逻辑的时候，是否写入到正确的库和表
13. 读取是否正常：是否能查询到正确的业务数据

3.3.2接口自动化

基于TestNG框架对一些常见的重复性较多的功能先编写Test Case，必要时可进行每日构建以帮助开发及测试人员及时发现bug，提高系统稳定性。手工及Jenkins做每日构建、jacoco统计覆盖率、Jmeter做并发测试等



1. 使用注解来表示case
2. 支持测试用例分组执行
3. 支持并发测试case编写
4. 提供了验证函数：Assertions
5. 生成测试结果报告

自动化执行与报告：

1. 整个测试项目目是基于maven构建的，使用Maven-SureFire-Plugin插件来执行case
2. 在maven构建生命周期的test阶段运行测试用例，产生两种不同形式的测试结果报告:纯文本与.xml文件
3. 命令:mvn test

# 被测特性

1. 前台用户的注册/登录：包括重复的用户名（幂等），不标准的用户名格式、错误的用户名/密码等场景
2. 角色扮演：包括三种角色的选择，及每种角色与其功能的对应关系
3. 职能学习：
4. 在线测试：包括选择病种考试，开始考试，查看成绩等功能

# 不被测特性

1. 本版本暂无，若后续在迭代中新增边界需求可考虑划分为不被测特性中

# 过程准则

## 入口准则

1. 代码单元测试覆盖率P1级接口达到80%
2. 需要开发自测的测试用例全部自测完成并通过
3. 自测过程中发现的bug修改完成并验证通过
4. 重要级别功能点发现bug数3~5个
5. 整体代码测试缺陷呈收敛走势。

## 停止准则

1. 测试期间发现的功能性bug全部修复且验证完成
2. 每个功能点发现bug数至少5个，正常场景、异常场景测试达到全覆盖
3. 非功能性测试用例通过率达到95%，系统测试报告提交并且获得批准

## 受阻准则

1. 开发提交的测试版本基本功能仍有bug，主流程有严重bug测试用例基本不能通过
2. 测试环境不稳定，无法进行测试

## 恢复准则

1. 基本功能的bug已修复，主流程bug已修复，主逻辑无较大问题
2. 测试环境稳定，可重新进行测试

# 测试进度

|  |  |  |
| --- | --- | --- |
| 测试活动 | 开始日期 | 结束日期 |
| 制定测试计划(包括用例) | *2018/3/8* | *2018/3/8* |
| 编写测试脚本 | *2018/3/8* | *2018/3/20* |
| 搭建测试环境 | *2018/3/20* | *2018/3/22* |
| 测试执行（包括修改缺陷的时间） | *2018/3/22* | *2018/4/5* |
| 回归测试 | *2018/4/5* | *2018/4/10* |
| 测试报告 | *2018/4/10* | *2018/4/15* |

# 测试资源

## 人力资源

|  |  |  |
| --- | --- | --- |
| 角色 | 所推荐的最少资源（所分配的专职角色数量） | 具体职责或注释 |
| 测试人员 | 1 | 1. 编写测试计划，把握项目测试整体进度 2. 梳理需求，编写测试用例 3. 项目提测之后根据测试用例进行功能测试，测试过程中记录发现bug，并对需要自动化测试的接口编写自动化测试，配置jacoco统计覆盖率 4. GUI测试、压力测试、性能测试整理性能测试报告 5. 提交项目整体测试报告 |
|  |  |  |
|  |  |  |
|  |  |  |

## 测试环境

|  |  |  |  |
| --- | --- | --- | --- |
| 软件环境（相关软件、操作系统等） | | | |
| 名称 | 版本 | 数量 | 获得途径 |
| 浏览器 |  | 1 |  |
| MySQL |  | 1 | 官网下载 |
| Python | 3.6.0 | 1 | 官网下载 |
| 硬件环境（网络、设备等） | | | |
| 名称 | 版本 | 数量 | 获得途径 |
| Mac |  |  |  |
| Windows |  |  |  |

## 测试工具

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用途 | 工具 | 版本 | 生产厂商/自产 | 获取途径 |
| 功能测试 | UFT |  | HP |  |
| 测试流程管理 | ALM |  | HP |  |
|  |  |  |  |  |
|  |  |  |  |  |

# 测试设计说明

*提示：分项列出被测对象的测试方法*

# 附件：测试用例

见附件测试用例