时间管理软件测试计划

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 文档标识： | |  | | | | 当前版本： | | 1.0 | | |
| 当前状态： | | 草稿 | | | ✓ | 发布日期： | |  | |
| 发布 | | |  |
| 修改历史 | | | | | | | | | |
| 日期 | 版本 | | 作者 | 修改内容 | | | 评审号 | | 变更控制号 |
|  |  | |  |  | | |  | |  |
|  |  | |  |  | | |  | |  |

**目 录**

[1 测试目的 3](#_Toc268146952)

[2 测试范围 3](#_Toc268146953)

[3 资源要求 3](#_Toc268146954)

[3.1 人力资源 3](#_Toc268146955)

[3.2 测试环境 3](#_Toc268146956)

[3.3 测试进度 4](#_Toc268146957)

[4 实施测试 4](#_Toc268146958)

[4.1 测试进度 4](#_Toc268146959)

[4.2. 测试策略 4](#_Toc268146961)

4.2.1 黑盒测试..................................................................................................4

4.2.2 白盒测试.................................................................................................5

[4.3 可交付件 5](#_Toc268146963)

[5 评审意见 6](#_Toc268146964)

[6 参考资料 6](#_Toc268146964)

测试目的

通过测试，达到以下目标：

* 测试时间管理软件是否达到设计的要求，包括：各个功能点是否以实现。
* 软件规定的操作和运行稳定。

Bug数和缺陷率控制在可接收的范围之内

测试范围

测试软件的功能是否满足用户的需求，性能是否优越以及系统所存在的问题。对系统的各个模块进行详细的测试，并记录测试的结果，对测试的结果进行细致的分析处理。测试时对系统的各个功能模块进行拆分测试，并以每一个模块都要测试到。对所有可能的结果进行测试，以及测试过程中存在的问题进行分析，然后提交测试的记录。最后，对软件存在的问题以及性能的测试进行全面分析，并给予记录。

在测试的过程中需要提出各个问题的假设，以及根据需求报告文档中存在的项目功能模块和用户的需求来改善系统。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。

资源要求

## 人力资源

|  |  |  |  |
| --- | --- | --- | --- |
| **角色** | **人数** | **姓名** | **职责** |
| 测试设计人员 | 1 | 马思腾 | 测试计划制定 |
| 测试实施人员 | 1 | 邵若川 | 测试计划实施 |
| 测试分析总结 | 1 | 唐子煜 | 记录所有的测试结果 |

## 测试环境

|  |  |
| --- | --- |
| **硬件环境** | 相关PC等。 |
| **软件环境** | WINDOWS 7 |

## 测试进度

|  |  |  |
| --- | --- | --- |
| **工作阶段** | **所需工作日** | **占项目的比例** |
| 测试设计阶段 | 1 | 25%% |
| 测试实施阶段 | 1 | 30% |
| 测试执行阶段 | 1 | 25% |
| 测试总结阶段 | 1 | 20% |

实施测试

## 4.1 测试类型

**黑盒测试**

**白盒测试**

## 4.2 测试策略

4.2.1黑盒测试

单元测试（PDL）

1新建测试

PROCEDURE 新建测试

do

for i=1 to 502 do

运行新建模块

end

期待结果：1-500条能正确输入 501条报错

2修改测试

PROCEDURE 修改测试

DO

for i=1 to 502 do

获得1-1000的随机数 j

运行修改模块j

END

期待结果：如果有可以找到随机数j代表的事件则修改，否则报错提示未找到

3删除测试

PROCEDURE 删除测试

DO

for i=1 to 502 do

获得1-1000的随机数 j

运行删除模块j

END

期待结果：如果有可以找到随机数j代表的事件则删除，否则报错提示未找到

4 倒数以及时间测试

PROCEDURE 倒数测试

DO

for i=1 to 502 do

运行新建模块（每个事件中相差的时间为1秒）

运行倒数模块

END

期待结果：程序能每隔1秒跳出一个提示窗口并且 内存占用并不能太大。

4.2.1白盒测试

**package** cn.list.control;

**import** **static** org.junit.Assert.\*;

**import** java.util.Date;

**import** org.junit.Test;

**import** cn.list.model.Event;

**import** cn.list.util.BaseException;

**import** cn.list.util.BusinessException;

**import** cn.list.util.DbException;

**public** **class** EventManageText {

@Test

**public** **void** testCreateEvent1() **throws** BusinessException, DbException {

//全部数据都有

Date begin=**new** Date();

Date end=**new** Date();

EventManage test=**new** EventManage();

test.CreateEvent("asd", begin, end, **true**, "asdfasdf", "dsafadsf", "asdfasdf", 2);

*fail*("Not yet implemented");

}

@Test

**public** **void** testCreateEvent2() **throws** BusinessException, DbException {

//没名字

Date begin=**new** Date();

Date end=**new** Date();

EventManage test1=**new** EventManage();

**try**{

test1.CreateEvent(**null**, begin, end, **true**, "asdfasdf", "dsafadsf", "asdfasdf", 2);

}**catch**(BaseException e){

e.printStackTrace();

}

*fail*("Not yet implemented");

}

@Test

**public** **void** testCreateEven4t() **throws** BusinessException, DbException {

//没起始时间

Date begin=**new** Date();

Date end=**new** Date();

EventManage test2=**new** EventManage();

**try**{

test2.CreateEvent("AD", **null**, end, **true**, "asdfasdf", "dsafadsf", "asdfasdf", 2);

}**catch**(BaseException e){

e.printStackTrace();

}

*fail*("Not yet implemented");

}

@Test

**public** **void** testCreateEvent5() **throws** BusinessException, DbException {

//没结束时间

Date begin=**new** Date();

Date end=**new** Date();

EventManage test3=**new** EventManage();

**try**{

test3.CreateEvent("asdas", begin, **null**, **true**, "asdfasdf", "dsafadsf", "asdfasdf", 2);

}**catch**(BaseException e){

e.printStackTrace();

}

*fail*("Not yet implemented");

}

@Test

**public** **void** DelEventtest() **throws** BusinessException, DbException{

//正常查询

EventManage test=**new** EventManage();

Event a=**new** Event();

**try**{

a=test.SerchEvent(1);

}**catch**(BaseException e){

e.printStackTrace();

}

}

@Test

**public** **void** DelEventtest2() **throws** BusinessException, DbException{

//如果查询id不存在

EventManage test=**new** EventManage();

Event a=**new** Event();

**try**{

a=test.SerchEvent(0);

}**catch**(BaseException e){

e.printStackTrace();

}

}

}

## 4.3 可交付件

|  |  |  |
| --- | --- | --- |
| **交付件** | **作者** | **交付时间** |
| 《测试计划》 | 马思腾 | 2016.4.25 |
| 《测试用例》 | 邵若川 |  |
| 《测试总结报告》 | 唐子煜 |  |

评审意见

|  |  |  |
| --- | --- | --- |
| **审核人** | **签名** | **日期** |
| 项目经理 |  |  |

6 参考资料

列出编写本计划及测试整个过程中所要参考的文件、资料。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **编号** | **资料名称** | **作者** | **日期** | **出版单位** |
| 1 | 《软件工程导论》第六版 | 张海藩  牟永敏 | 2013.8 | 清华大学出版社 |