时间戳转换工具

测试报告

版本 <1.1>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| <12/04/2020> | <1.0> | 添加使用等价类方法进行功能测试的报告 | 赵胜龙 |
| <12/04/2020> | <1.1> | 添加使用边界值方法进行功能测试的报告 | 柳清源 |
|  |  |  |  |
|  |  |  |  |

目录

1. 简介 4

1.1 目的 4

1.2 范围 4

1.3 定义、首字母缩写词和缩略语 4

1.4 参考资料 4

1.5 概述 4

2. 测试概要 4

3. 测试环境 4

4. 测试结果及分析 5

4.1 需求覆盖率及缺陷分布 5

4.2 缺陷严重程度 5

5. 缺陷清单 5

5.1 功能性缺陷 5

6. 测试结论与建议 6

测试报告

# 简介

## 目的

本测试报告是在学习完上海交通大学软件学院课程《软件测试》（课程代码SE213）黑盒测试后的作业。目的是在实践中使用课程中学习到的边界值测试、等价类测试、决策表测试等方法以提高对软件测试方法的认识，从而提高软件测试技巧。

## 范围

此测试报告主要针对蛙蛙工具中的功能进行黑盒测试；

测试中主要使用到的方法包括边界值测试、等价类测试；

## 定义、首字母缩写词和缩略语

蛙蛙：蛙蛙工具——时间戳转换工具

Timestamp: Unix 时间戳

timestamp: 蛙蛙的主要功能模块

unix2time: 蛙蛙中将时间戳转换为日期的模块名称

time2unix: 蛙蛙中将日期转换为时间的模块名称

UI: 蛙蛙的用户界面

## 参考资料

《蛙蛙时间戳工具规格说明书》 v1.0

## 概述

本测试报告包含测试概要、测试环境、测试结果分析、缺陷报告、测试总结五大部分。测试方法主要使用了边界值测试和等价类测试。测试用例添加在同目录下的《测试用例》文件中。测试用例主要针对《蛙蛙时间戳工具规格说明书》中明确的软件功能设计。

# 测试概要

测试时间：10/04/2020 – 18/04/2020

地点：受疫情影响，测试地点均在各自家中

人员：

《蛙蛙时间戳工具规格说明书》撰写者——戴方越

两名软件测试人员 —— 柳清源、赵胜龙

测试计划统筹人员 —— 芮召普

测试方法：黑盒测试，包括功能测试、系统集成测试等

测试内容：针对《蛙蛙时间戳工具规格说明书》中明确的软件功能和需求进行测试，观察实际输出和理想输出之间的区别，并记录。

# 测试环境

本时间戳转换工具是蛙蛙工具集中的一个，软件运行在web环境中。

其中使用等价类方法进行测试时，浏览器为Google chrome v80.0.3987.163正式版;操作系统为Windows 10家庭版 18362.720

# 测试结果及分析

## 需求覆盖率及缺陷分布

**表1 需求覆盖率及缺陷分布**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **测试内容** | | **测试**  **用例数** | **需求**  **覆盖率** | **缺陷数** | **缺陷率** | **备注** |
| **功能项** | 输入Unix时间戳，输出相应的事件 | 31 | 100% | 5 | 16.13% |  |
| 输入时间，输出相应的时间戳 | 108 | 100% | 40 | 37.03% |  |
|  | **总计** | 139 | 100% | 45 | 32.37% |  |

## 缺陷严重程度

**表2 缺陷严重程度**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **严重程度** | **致命** | **严重** | **一般** | **细微** | **总数** |
| **缺陷个数** | 4 | 41 | 0 | 0 | 45 |
| **占缺陷百分比** | 8.89% | 91.11% | 0 | 0 | 100% |

# 缺陷清单

## 功能性缺陷

**表3 功能性缺陷列表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序**  **号** | **缺陷**  **编号** | **严重程度** | **功能模块** | **缺陷描述** |
| 1 | Bug001 | High | 输入日期，得到时间戳 | 输入范围越界未检查 |
| 2 | Bug002 | Urgent | 输入日期，得到时间戳 | 输入格式未检查 |
| 3 | Bug003 | High | 输入时间戳，得到日期 | 输入范围未检查 |
| 4 | Bug004 | High | 输入时间戳，得到日期 | 范围越界到一定程度，直接抛出了javascript的异常 |

表中相关项说明：

1. 缺陷编号：为每个缺陷赋予一个唯一的编号，可以通过此编号对缺陷进行跟踪。例如：Bug001。
2. 缺陷严重程度：缺陷可以根据严重程度分为以下几种情况。
3. 致命（fatal）：致命的错误，测试执行直接导致系统死机、蓝屏、挂起、或是程序非法退出；系统的主要功能或需求没有实现。
4. 严重（critical）：严重错误，系统的次要功能点或需求点没有实现；数据丢失或损坏。执行软件主要功能的测试用例导致系统出错，程序无法正常继续执行；程序执行过于缓慢或是占用过大的系统资源。
5. 一般（major）：不太严重的错误，这样的缺陷虽然不影响系统的基本使用，但没有很好地实现功能，没有达到预期的效果。如次要功能丧失，界面错误，打印内容、格式错误，提示信息不太正确，或用户界面太差，简单的输入限制未放在前台进行控制，删除操作未给出提示，操作时间长等。
6. 细微（minor）：一些小问题，对功能几乎没有影响，产品及属性仍可使用。如软件的实际执行过程与需求有较小的差异；程序的提示信息描述容易使用户产生混淆；有个别错别字、 文字排列不整齐；辅助说明描述不清楚，显示格式不规范，长时间操作未给用户进度提示，提示窗口文字未采用行业术语，可输入区域和只读区域没有明显的区分标志，系统处理未优化等。
7. 功能模块：所测试并出现该缺陷的功能模块名称。
8. 缺陷描述：对缺陷的详细描述。描述缺陷的重现步骤和问题所在，描述需清晰明了，使研发人员可以按照描述将缺陷重现出来。必要时可以放上相应的截图，以便于研发人员理解问题所在。

# 测试结论与建议

由于软件功能数量少，比较简单，本次测试观察到的缺陷集中在一到两种问题上，尤其突出的是输入范围、输入格式的错误。由以上的测试，可以看出，时间戳转换工具在输入合法性的检查上有所欠缺，亟需加强。

本次测试主要采用了边界值测试方法和等价类测试方法。两种方法互相独立，有两位测试人员进行，在实际的测试中可能比较冗余，但是因为练习的需求，我们还是将两种方法独立开来，有两名人员实施执行。因此，在缺陷率达到30%的情况下，缺陷数却只有不多的4~5个。