

**软件测试**

**实验报告**

|  |  |
| --- | --- |
| **课程名称：** | **软件测试** |
| **学生姓名：** | **彭晓轩** |
| **学生学号：** | **201730683321** |
| **学生专业：** | **软件工程** |
| **开课学期：** | **2019-2020学年第一学期** |

**软件学院**

**2019年11月22日**

一．实验目的

1. 建立Shopping软件系统测试环境
2. 评估Shopping软件系统可维护性
3. 实验分析
4. 软件测试环境

软件测试环境包括设计环境，实施环境和管理环境三部分，是指为了完成软件测试工作所必需的硬件、软件、设备、数据的总称。软件测试环境是软件测试实施的一个重要阶段，软件测试环境适合与否会严重影响测试结果的真实性和正确性。

1. 软件可维护性

软件可维护性是指纠正软件系统出现的错误和缺陷，以及为满足新的要求进行修改、扩充或压缩的容易程度。可维护性、可使用性、可靠性是衡量软件质量的主要质量特性，也是用户十分关心的几个方面。软件的可维护性是软件开发阶段各个时期的关键目标。目前被广泛用于度量软件系统可维护性的七个特性是：可理解性、可使用性、可测试性、可移植性、可修改性、可靠性、效率。

1. 实验过程
2. 软件测试环境

下面各图展示了软件测试环境：



图1 硬件环境，个人笔记本电脑相关配置



图2 软件环境，Windows10操作系统

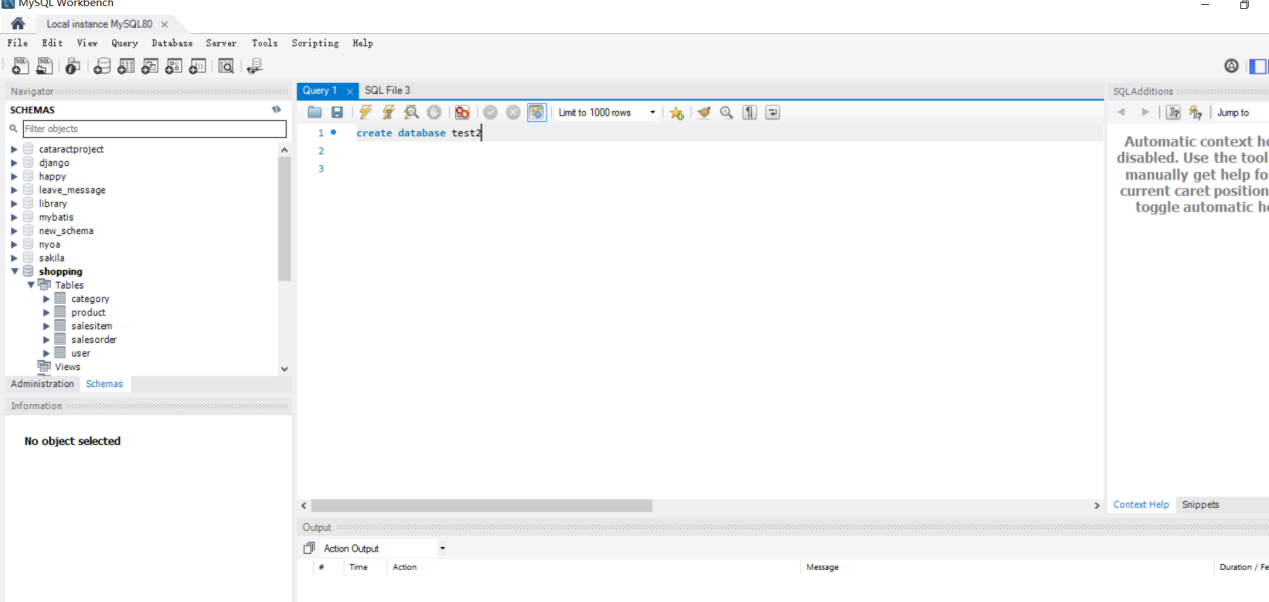


图3 软件环境，使用MySQL数据库8.0.15版本

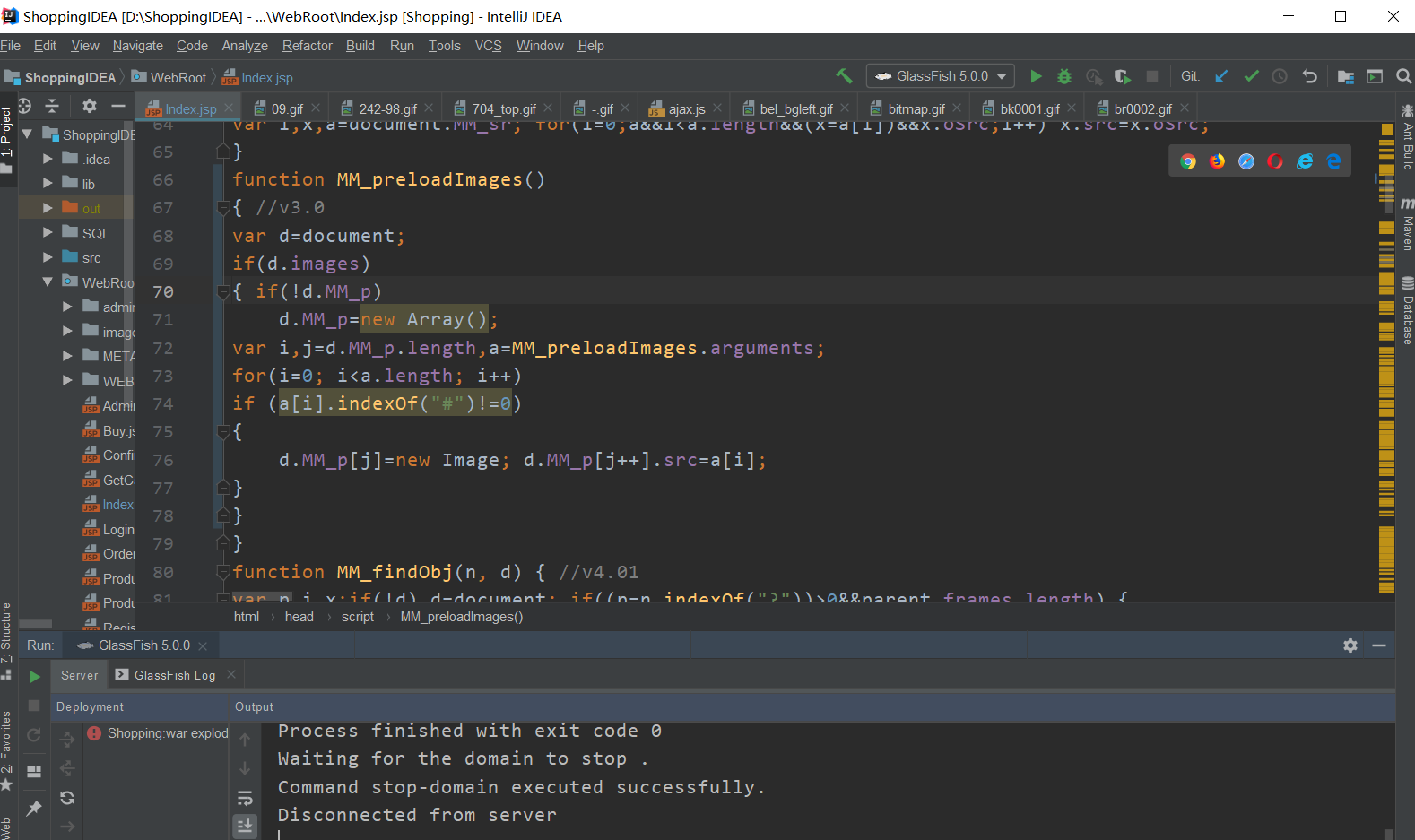


图4 软件环境，IntelliJ IDEA Ultimate 2019版本以及GlassFish 5.0.0版本

下面各图展示了搭建好的Shopping系统的被测环境：



图5 被测模块截图1

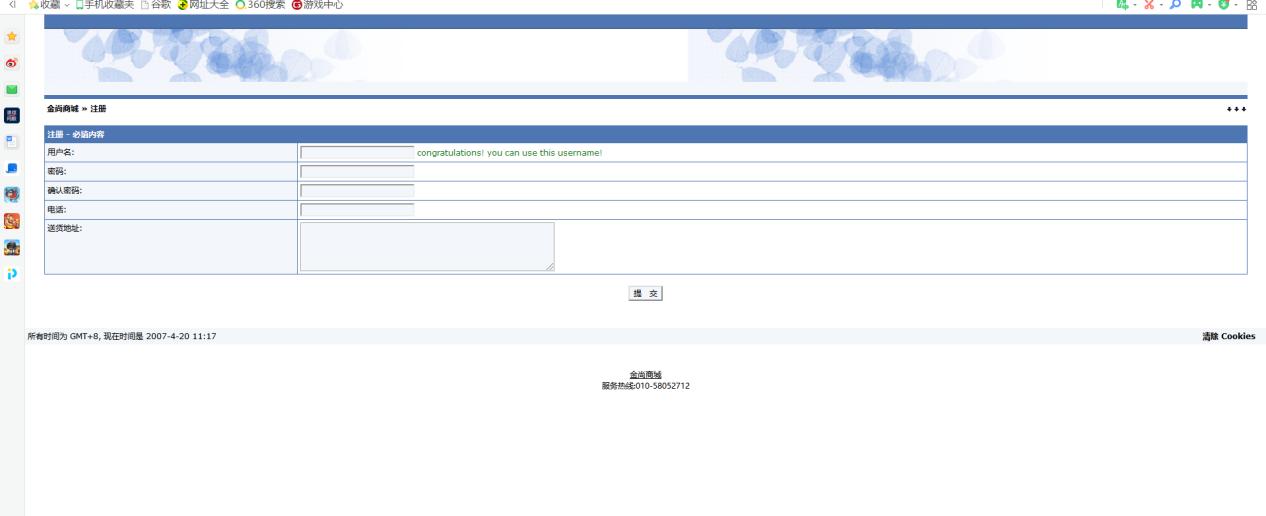


图6 被测模块截图2

1. 软件可维护性

建立质量检查表如下：

|  |  |  |
| --- | --- | --- |
| 特性 | 问题 | 评估 |
| 可理解性 | 代码模块化？ | Yes |
| 代码风格一致性？ | Yes |
| 不使用令人捉摸不定或含糊不清的代码？ | No |
| 使用有意义的数据名和过程名？ | No |
| 代码结构化？ | Yes |
| 代码完整性？ | Yes |
| 可靠性 | 可靠性度量指标是否在可接受范围？ | Yes |
| 可测试性 | 程序是否模块化? 结构是否良好? | Yes |
| 程序是否可理解? 程序是否可靠? | No |
| 程序是否能显示任意中间结果? | No |
| 程序是否能以清楚的方式描述它的输出? | Yes |
| 程序是否能及时地按照要求显示所有的输入? | Yes |
| 程序是否有跟踪及显示逻辑控制流程的能力? | Yes |
| 程序是否能从检查点再启动? | Yes |
| 程序是否能显示带说明的错误信息? | Yes |
| 可修改性 | 程序是否易于修改？ | No |
| 可移植性 | 是否是用高级的独立于机器的语言来编写程序? | Yes |
| 是否使用广泛使用的标准化的程序设计语言来编写程序? 是否仅使用了这种语言的标准版本和特性? | Yes |
| 程序中是否使用了标准的普遍使用的库功能和子程序? | Yes |
| 程序中是否极少使用或根本不使用操作系统的功能? | Yes |
| 程序在执行之前是否初始化内存? | Yes |
| 程序在执行之前是否测定当前的输入／输出设备? | No |
| 程序是否把与机器相关的语句分离了出来，集中放在了一些单独的程序模块中，并有说明文件? | Yes |
| 程序是否结构化? 并允许在小一些的计算机上分段(覆盖)运行? | No |
| 程序中是否避免了依赖于字母数字或特殊字符的内部位表示? | Yes |
| 效率 | 程序是否模块化? 结构是否良好? | Yes |
| 是否消除了无用的标号与表达式，以充分发挥编译器优化作用? | Yes |
| 程序的编译器是否有优化功能? | Yes |
| 是否把特殊子程序和错误处理子程序都归入了单独的模块中? | No |
| 是否以快速的数学运算代替了较慢的数学运算? | Yes |
| 是否尽可能地使用了整数运算，而不是实数运算? | Yes |
| 是否在表达式中避免了混合数据类型的使用，消除了不必要的类型转换? | Yes |
| 程序是否避免了非标准的函数或子程序的调用? | Yes |
| 在几条分支结构中，是否最有可能为“真”的分支首先得到测试? | Yes |
| 在复杂的逻辑条件中，是否最有可能为“真”的表达式首先得到测试? | Yes |
| 可使用性 | 程序是否具有自描述性? | Yes |
| 程序是否能始终如一地按照用户的要求运行? | Yes |
| 程序是否让用户对数据处理有一个满意的和适当的控制? | No |
| 程序是否容易学会使用? | No |
| 程序是否使用数据管理系统来自动地处理事务性工作和管理格式化、地址分配及存储器组织？ | Yes |
| 程序是否具有容错性? | Yes |
| 程序是否灵活? | No |

据此判断，该程序的可理解性、可修改性和可使用性比较低，可测试性、可移植性、效率和可靠性比较高。

为了提高Shopping软件系统的可维护性，可以采取以下方法：

①建立明确的软件质量目标和优先级

②使用提高软件质量的技术和工具

③进行明确的质量保证审查

④选择可维护的程序设计语言

⑤改进程序的文档

1. 实验总结

本次实验主要进行了对给定软件系统和测试模块的软件测试环境的搭建和软件可维护性的评估。只有在一个稳定安全可靠的软件测试环境下，我们才可以合理正确有效率地对被测试模块应用各种软件测试方法和技巧。另外，我们开发的软件系统不仅需要通过测试，还应该考虑后续若干年的维护问题，在开发前就要考虑到后续维护的成本和及时采取合理措施来提高软件系统的可维护性。