****

**软件测试技术第二次实验报告**



**学 院 智能与计算学部**

**专 业 软件工程**

**年 级 16**

**姓 名 陈森林**

**2019年 03 月 26 日**

# 软件测试技术第二次实验报告

1. 需求分析（描述具体需求）
2. 安装Selinium 到Eclipse
3. 安装firefox浏览器和SeleniumIDE
4. 使用seleniumIDE录制和查看页面
5. 检查“软件测试名单.xlsx”内的每条记录和网页信息是否一致
6. 概要设计（简单描述设计思路，配合UML图）
7. 使用poi导入xlsx文件
8. 通过selenium和gecko driver对火狐浏览器进行外部控制
9. 将xlsx内字段和页面字段比较
10. 详细设计（详细描述具体如何实现，附代码及说明）
11. 使用poi读取xlsx：

InputStream f=**new** FileInputStream("E:\\java proj\\theLab2\\testlist.xlsx");

XSSFWorkbook wb = **new** XSSFWorkbook(f);

XSSFSheet sheet = wb.getSheetAt(0);

**for**(**int** row=2; row<sheet.getLastRowNum();row++){

XSSFRow xssfrow=sheet.getRow(row);

XSSFCell cell = xssfrow.getCell(1);

1. 使用selenium和driver操控浏览器

String driverPath = System.*getProperty*("user.dir") + "/src/geckodriver.exe";

System.*setProperty*("webdriver.gecko.driver", driverPath);

driver = **new** FirefoxDriver();

baseUrl = "http://121.193.130.195:8800/login";

driver.manage().timeouts().implicitlyWait(600, TimeUnit.***SECONDS***);

String driverPath = System.*getProperty*("user.dir") +"/src/geckodriver.exe";

System.*setProperty*("webdriver.gecko.driver", driverPath);

driver = **new** FirefoxDriver();

baseUrl = "http://121.193.130.195:8800/login";

driver.manage().timeouts().implicitlyWait(600, TimeUnit.***SECONDS***);

1. 比较信息

*assertEquals*(xssfrow.getCell(2).toString(), driver.findElement(By.*id*("student-name")).getText());

*assertEquals*(xssfrow.getCell(3).toString(), driver.findElement(By.*id*("student-git")).getText());

1. 调试分析（在实验过程中遇到的问题以及如何解决）

在读取xlsx时，由于表格内是数字格式，但要比较的是字符串，想将其读取成字符串频繁报错，及时将表格内容设为文本也无济于事。后来通过poi的方法cell.setCellType(CellType.STRING)将表格强制设为字符串类型即可。

1. 测试结果（描述输入和输出）



发现原表格中有一条信息多了空格

1. 总结

了解到selenium的使用方法，已经通过java来进行对网页的操控

七、URL：https://github.com/csl784513390/SoftwareTest/tree/master/theLab2