****

**软件测试技术第二次实验报告**



**学 院 软件学院**

**专 业 软件工程**

**年 级 2016级**

**姓 名 梁哲粼**

**2019年 3 月 26 日**

# 软件测试技术第二次实验报告

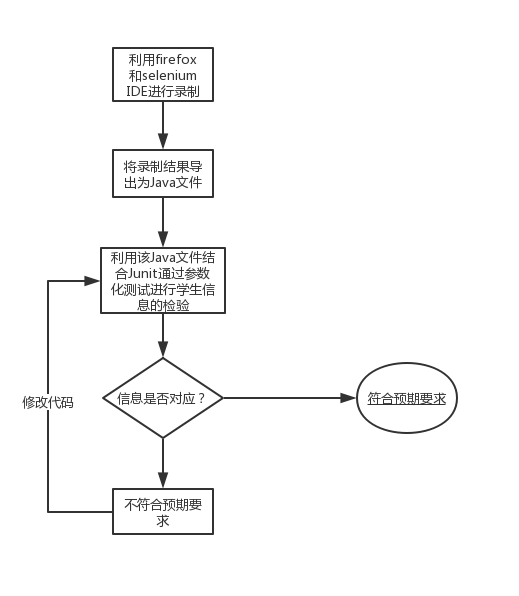
1. 需求分析（描述具体需求）

“软件测试名单.xlsx” contains information about the students, and <http://121.193.130.195:8800> can view someone’s information after logging in (student id as username, the last 6 digits of student id as password). Please check each record in the excel to make sure that each student’s information is consistent with the information on the website.

1. 概要设计（简单描述设计思路，配合UML图）

首先利用firefox的扩展selenium IDE结合所给网站进行操作进行录制，并将录制结果导出为Java代码。然后利用该代码进行编程，结合参数化方法进行自动化测试，来检验结果的正确性。

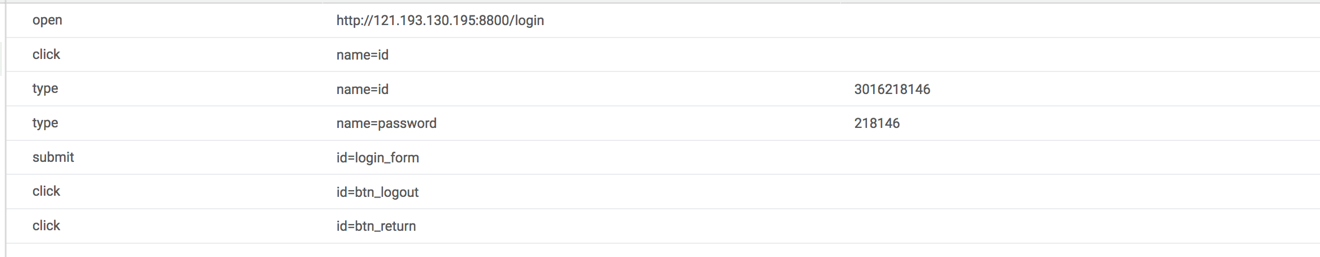
(UNL图如下所示)



1. 详细设计（详细描述具体如何实现，附代码及说明）

1：下载firefox，并在扩展中下载selenium IDE来进行人工测试。

2：在selenium IDE中,新建一个工程，网址定位所给的学生信息登录网址，进行录制，将人工所以进行的操作录制下来并导出为Java文件。由于新版本selenium IDE抛弃了导出功能，于是下载Katalon Recorder进行上述操作，最后得到Java文件。录制操作流程如图所示



Java代码如下图所示：

public class StuTest {

private WebDriver driver;

private String baseUrl;

private boolean acceptNextAlert = true;

private StringBuffer verificationErrors = new StringBuffer();

@Before

public void setUp() throws Exception {

driver = new FirefoxDriver();

baseUrl = "https://www.katalon.com/";

driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);

}

@Test

public void testStu() throws Exception {

driver.get("http://121.193.130.195:8800/login");

driver.findElement(By.name("id")).click();

driver.findElement(By.name("id")).clear();

driver.findElement(By.name("id")).sendKeys("3016218146");

driver.findElement(By.name("password")).clear();

driver.findElement(By.name("password")).sendKeys("218146");

driver.findElement(By.id("login\_form")).submit();

driver.findElement(By.id("btn\_logout")).click();

driver.findElement(By.id("btn\_return")).click();

}

@After

public void tearDown() throws Exception {

driver.quit();

String verificationErrorString = verificationErrors.toString();

if (!"".equals(verificationErrorString)) {

fail(verificationErrorString);

}

}

private boolean isElementPresent(By by) {

try {

driver.findElement(by);

return true;

} catch (NoSuchElementException e) {

return false;

}

}

private boolean isAlertPresent() {

try {

driver.switchTo().alert();

return true;

} catch (NoAlertPresentException e) {

return false;

}

}

private String closeAlertAndGetItsText() {

try {

Alert alert = driver.switchTo().alert();

String alertText = alert.getText();

if (acceptNextAlert) {

alert.accept();

} else {

alert.dismiss();

}

return alertText;

} finally {

acceptNextAlert = true;

}

}

}

3：通过该Java文件创建测试用例。首先导入poi.jar包来获得对excel文件读写的操作,将excel的学生名单存成三个数组，分别存放学号、姓名和github地址。操作excel代码如下

String driverPath = "/Users/maomao/Desktop/软件测试/实验二/geckodriver";

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

*driver* = **new** FirefoxDriver();

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

*driver*.manage().timeouts().implicitlyWait(30, TimeUnit.***SECONDS***);

*driver*.get(*baseUrl* + "/");

InputStream is = **new** FileInputStream("/Users/maomao/Desktop/软件测试/实验二/软件测试名单.xls");

HSSFWorkbook excel = **new** HSSFWorkbook(is);

HSSFSheet sheet = excel.getSheetAt(0);

*stdID* = **new** String[sheet.getLastRowNum()-1];

*name* = **new** String[sheet.getLastRowNum()-1];

*gitURL* = **new** String[sheet.getLastRowNum()-1];

**for** (**int** rowNum = 0; rowNum < sheet.getLastRowNum(); rowNum++) {

HSSFRow row = sheet.getRow(rowNum+2);

**if** (row == **null**)

**continue**;

HSSFCell cell1 = row.getCell((**short**) 1);

HSSFCell cell2 = row.getCell((**short**) 2);

HSSFCell cell3 = row.getCell((**short**) 3);

*stdID*[rowNum] = cell1.getStringCellValue();

*name*[rowNum] = cell2.getStringCellValue();

*gitURL*[rowNum] = cell3.getStringCellValue();

4：接着倒入selenium包，编写网上测试代码，测试的时候要用到参数化测试的方法，即将学生的所用信息存成一个二位数组，并以list的形式返回，同时创建构造函数。将list的内容赋给三个变量，最后创建test方法来检验excel中的内容是否和所给网站上的内容相符合。代码如下

**public** TestBaidu(String curstdID, String curName, String curgitURL) {

**this**.curtdID = curstdID;

**this**.curName = curName;

**this**.curgitURL = curgitURL;

}

@Test

**public** **void** testBaidu() **throws** Exception {

WebElement id = *driver*.findElement(By.*name*("id"));

id.click();

id.sendKeys(curtdID);

WebElement password = *driver*.findElement(By.*name*("password"));

password.click();

password.sendKeys(curtdID.substring(curtdID.length()-6));

*driver*.findElement(By.*id*("btn\_login")).click();

*assertEquals*(curtdID, *driver*.findElement(By.*id*("student-id")).getText());

*assertEquals*(curName, *driver*.findElement(By.*id*("student-name")).getText());

*assertEquals*(curgitURL, *driver*.findElement(By.*id*("student-git")).getText());

*driver*.findElement(By.*id*("btn\_logout")).click();

*driver*.findElement(By.*id*("btn\_return")).click();

}

1. 调试分析（在实验过程中遇到的问题以及如何解决）

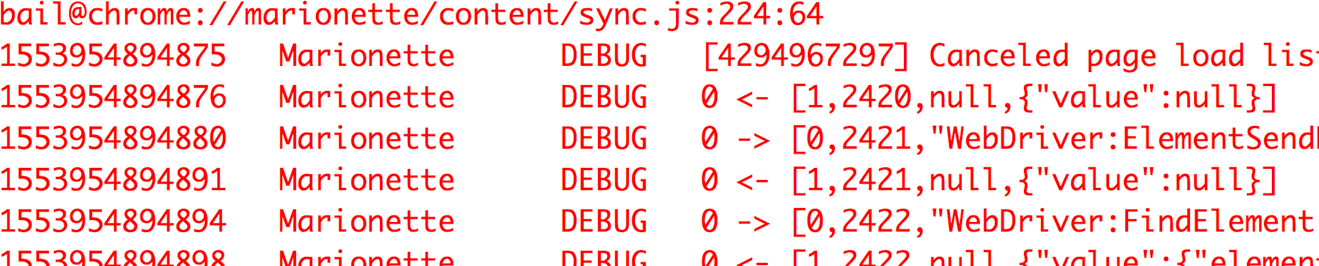
在编写test方法时进行了优化：

第一版的代码只是在一个Test方法里写了一个while循环测试，当测试失败的时候代码就会停止，而且不会说明是哪个学生的测试样例出现了问题，调试时会很麻烦

第二版的代码加入了参数化测试方法，利用参数化测试方法返回的list可以自动创建多个测试样例，每个测试样例对应一个学生信息，这样就解决了无法调试的问题。

1. 测试结果（描述输入和输出）

测试的部分输出以及结果如下所示：





六、总结

1.了解了selenium IDE以及katalon Recorder的使用方法

2.了解了如何利用Junit进行参数化测试。即需要一个return测试数据的list的方法、一个构造函数方法以及一个Test方法。

3.了解了如何利用Java进行excel的读写操作。