软件测试上机报告

第二次上机作业

学　院__智能与计算学部__
专　业____软件工程_____
姓　名_____陈姝宇_____
学　号___3017218119_____
年　级_____2017 级_____
班　级_____3 班_____

1. Experimental Requirements

Tasks:

1. Install Selenium with Eclipse.
2. Install Firefox and SeleniumIDE plugin.
3. Try to record and export scripts using SeleniumIDE.
4. Please complete the following task using Selenium Webdriver: "Selenium Lab.xlsx" contains information about the students, and http://103.120.226.190/selenium-demo/git-repo can view someone's information after logging in (student id as username, git address 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.

Requirements for the experiment:

1. Finish the tasks above individually.
2. Check in your java code to github or gitee
3. Please send your experiment report to 智慧树, the following information should be included in your report:
   a) The brief description that you install Selenium, Firefox and SeleniumIDE.
   b) Steps for recording and exporting scripts.
   c) Steps for testing the website using Selenium.

2. Environmental configuration

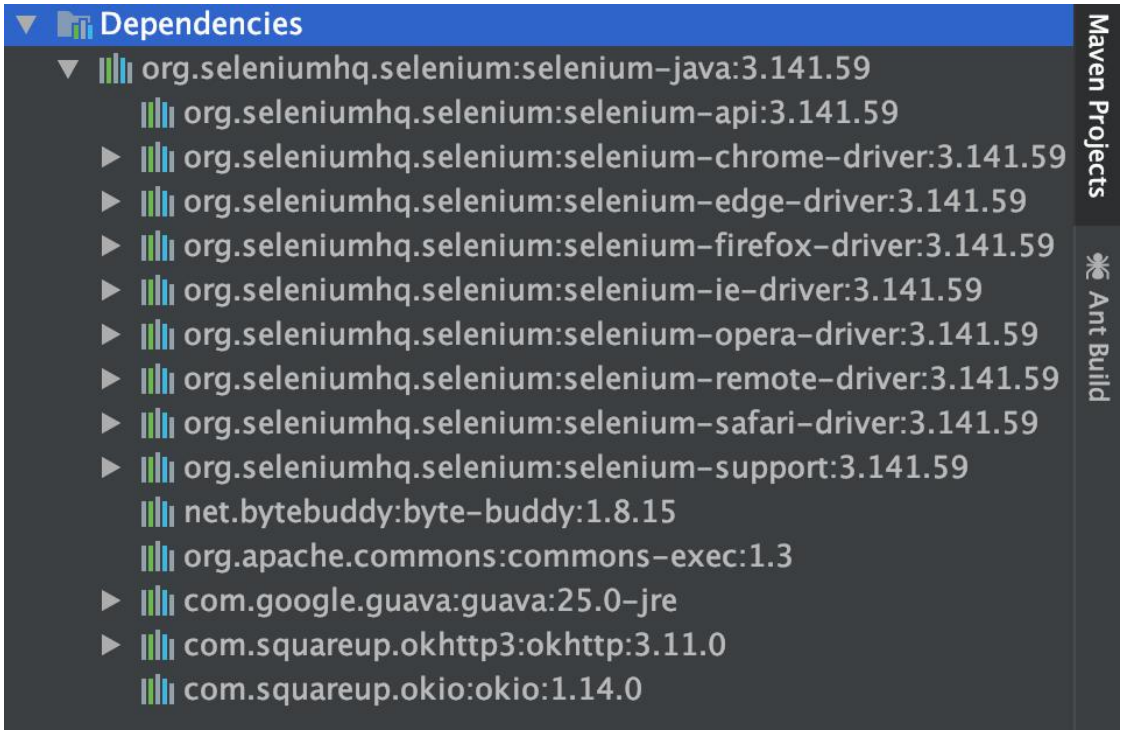
The brief description that I install Selenium in intelliJ IDEA:

1) Create maven project : ex2, add selenium dependency in pom.xml

```
<!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java -->
<dependencies>
    <dependency>
        <groupId>org.seleniumhq.selenium</groupId>
        <artifactId>selenium-java</artifactId>
        <version>3.141.59</version>
    </dependency>
</dependencies>
```

2) Auto-import :

3) After import successfully, the jar package imported through Maven will be displayed in [Maven projects] on the right side of the idea:



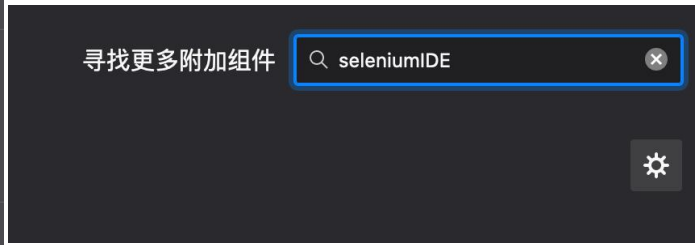

```
13        <dependency>
14            <groupId>org.seleniumhq.selenium</groupId>
15            <artifactId>selenium-java</artifactId>
16            <version>3.141.59</version>
17        </dependency>
```

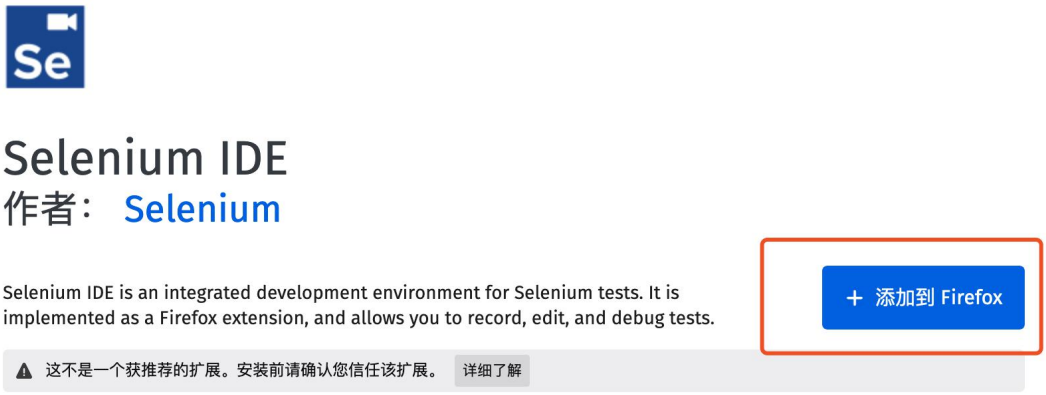
The brief description that I install Firefox and SeleniumIDE:

1) Visit the official website: http://www.firefox.com.cn/download/, and click ‘download’

2) Click ‘Firefox-latest.dmg' and move Firefox into applications, then open Firefox

3) Find component: seleniumIDE

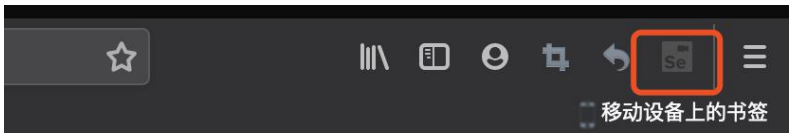




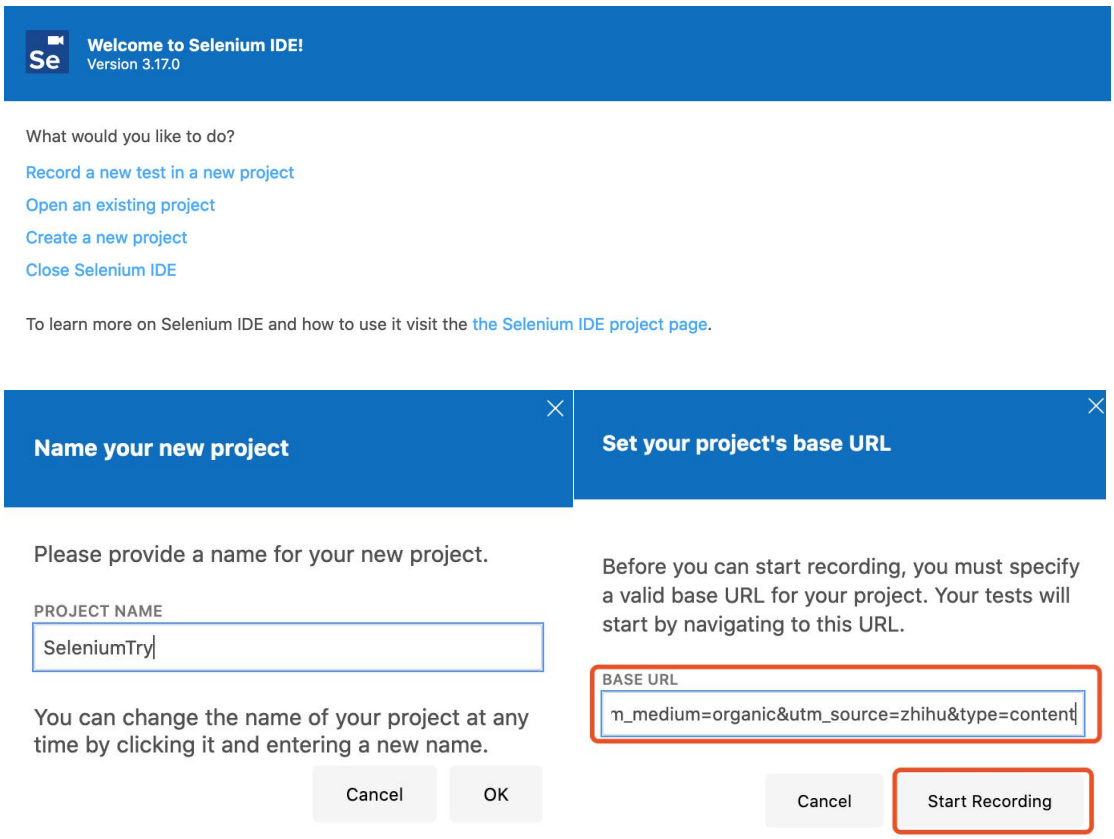
3. Recording and exporting scripts

Steps for recording and exporting scripts.

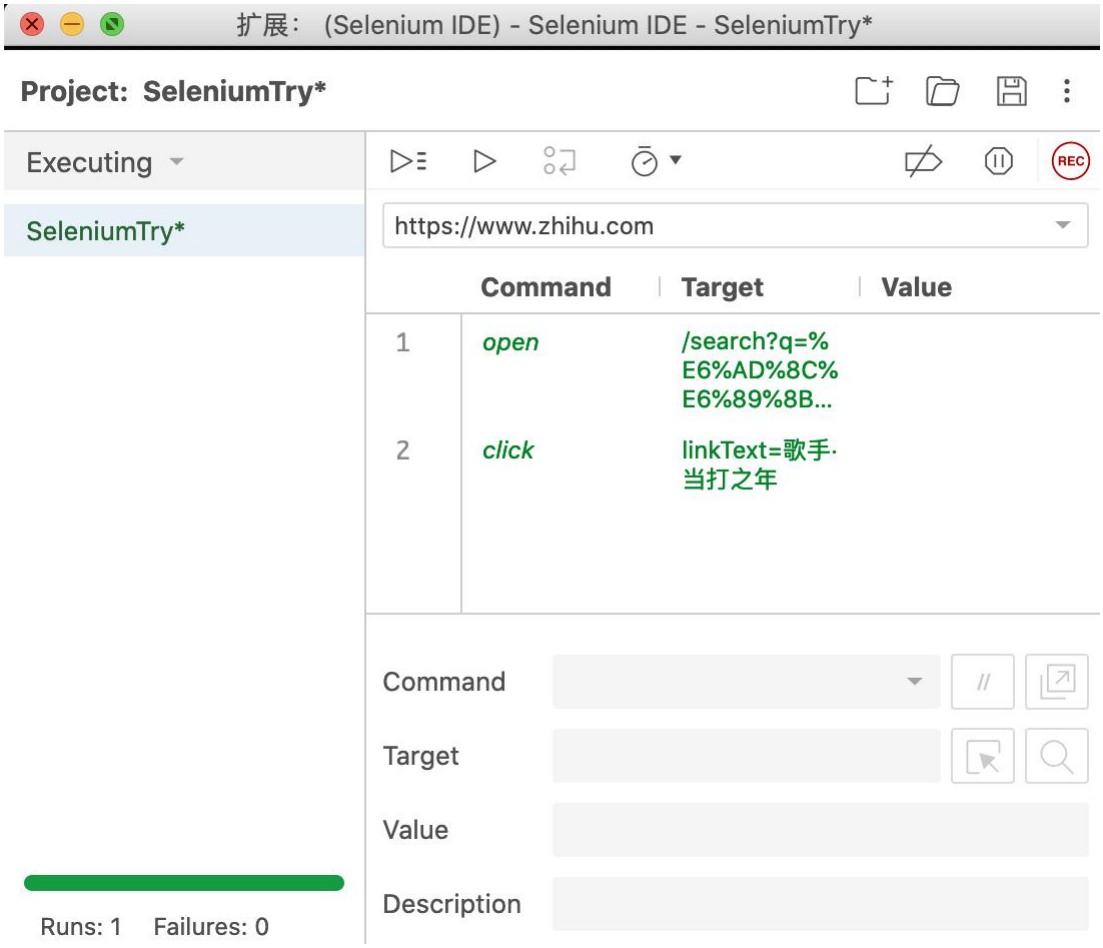
1) The icon of seleniumIDE at the right and top of firefox windows.



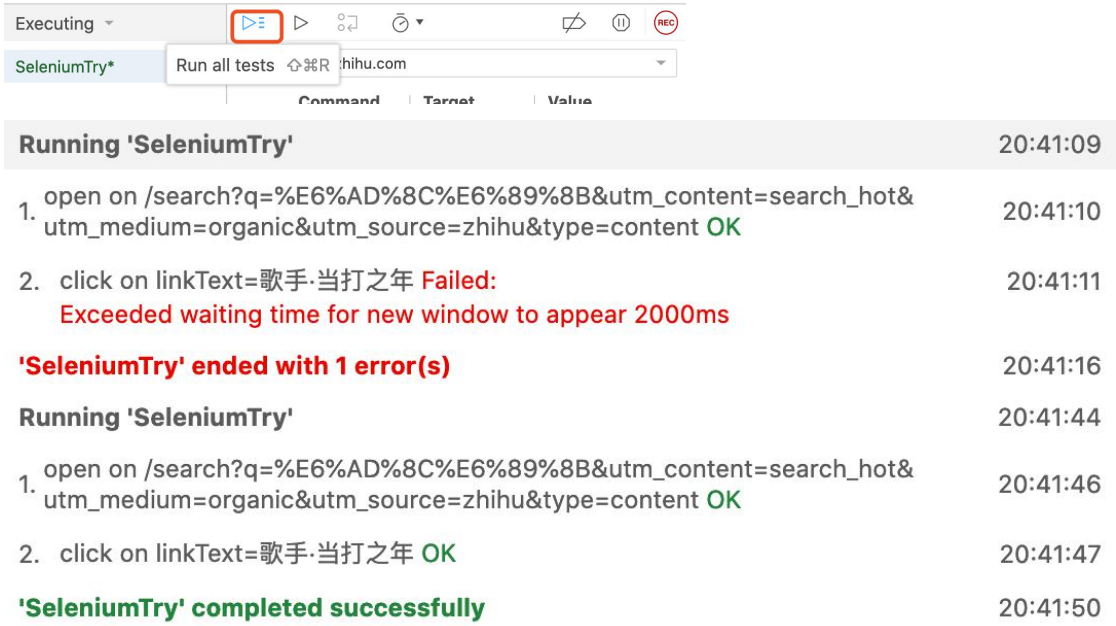
2) Click the seleniumIDE icon and choose ‘Record a new test in a new project’:



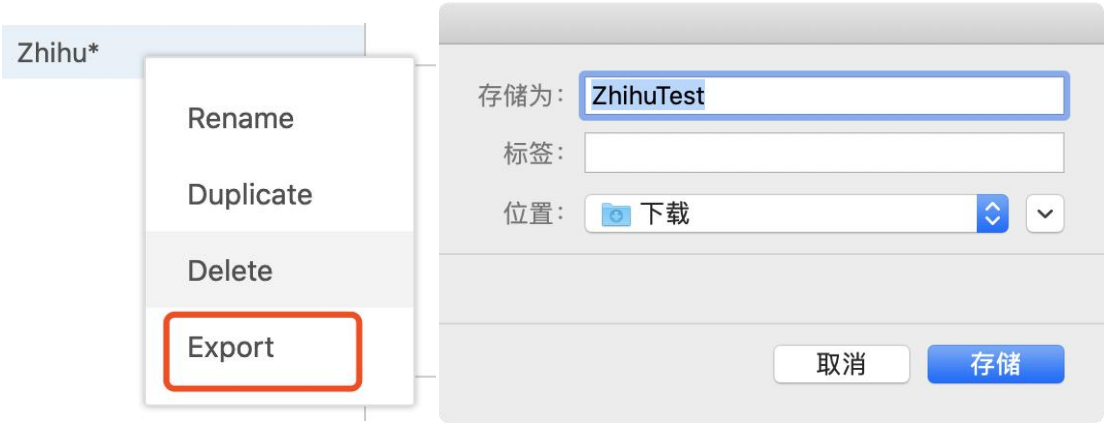
3) Create Project and add command



4) Click run tests and see log. At first, the test if failed because of popupwindow stoped by firefox. After I set the pop-up operation to allow, and then rerun the test case, it is successful.

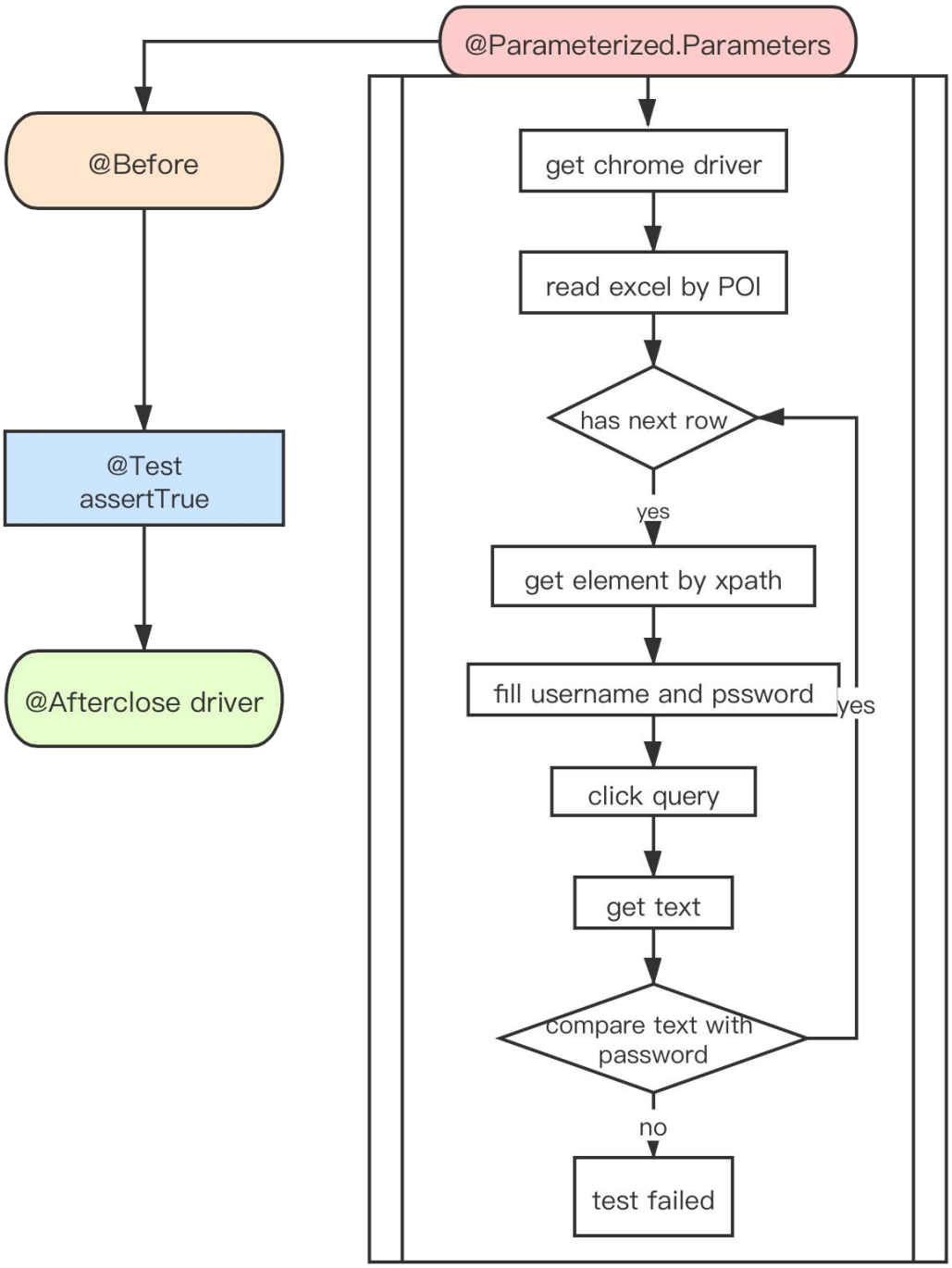


5) Right click Zhihu* and Choose Export, resulting in exporting script. And in Download document, there is a ZhihuTest.java. Open ZhihuTest.java.




```
public class ZhihuTest {
  private WebDriver driver;
  private Map<String, Object> vars;
  JavascriptExecutor js;
  @Before
  public void setUp() {
    driver = new FirefoxDriver();
    js = (JavascriptExecutor) driver;
    vars = new HashMap<String, Object>();
  }
  @After
  public void tearDown() { driver.quit(); }
  public String waitForWindow(int timeout) {
    try {
      Thread.sleep(timeout);
    } catch (InterruptedException e) {
      e.printStackTrace();
    }
    Set<String> whNow = driver.getWindowHandles();
    Set<String> whThen = (Set<String>) vars.get("window_handles");
    if (whNow.size() > whThen.size()) {
      whNow.removeAll(whThen);
    }
    return whNow.iterator().next();
  }
  @Test
  public void zhihu() {
    driver.get("https://www.zhihu.com/search?q=%E6%AD%8C%E6%89%8B&utm_content=se
    driver.manage().window().setSize(new Dimension( width: 1440,  height: 900));
    vars.put("window_handles", driver.getWindowHandles());
    driver.findElement(By.linkText("歌手·当打之年")).click();
    vars.put("win8614", waitForWindow( timeout: 2000));
    driver.switchTo().window(vars.get("win8614").toString());
  }
```

4. testing the website using Selenium



5. Source Code

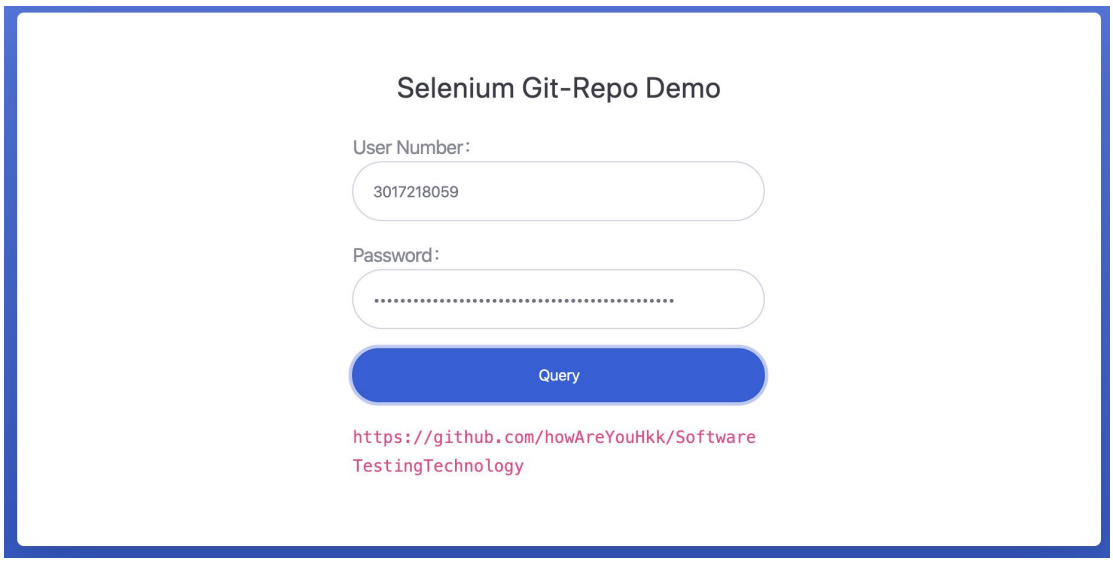
```
@RunWith(Parameterized.class)
public class Ex2 {

    private static WebDriver driver;
    private String expected;
    private String actual;

    public Ex2(String expected, String actual) {
        this.expected = expected;
        this.actual = actual;
    }

    @Before
    public void setUp() {
    }

    @Parameterized.Parameters
    public static Collection<Object[]> getData() {

        // create chromeDriver
        String driverPath = System.getProperty("user.dir") +
"/src/main/resources/driver/chromedriver";
        System.setProperty("webdriver.chrome.driver", driverPath);
        driver = new ChromeDriver();
        // baseurl
        driver.get("http://103.120.226.190/selenium-demo/git-repo");

        ArrayList<Object[]> datas = new ArrayList<Object[]>();

        String filePath = System.getProperty("user.dir") +
```

```
"/src/main/resources/info/Selenium+Lab2020.xlsx";
        File xlsFile = new File(filePath);
        Workbook workbook;
        try {
            workbook = WorkbookFactory.create(xlsFile);
            // first sheet
            Sheet sheet = workbook.getSheetAt(0);
            int row = 0;
            while (true) { // view every row
                Row r = sheet.getRow(row); // get this row
                String c1 = r.getCell(1).getStringCellValue(); // get username
                if (c1.equals("")) break; // if this row is null, then break out
                String c2 = r.getCell(2).getStringCellValue(); // get password

                // username
driver.findElement(By.xpath("/html/body/div/div/div/div/div/div/div[2]/di
v/form/div[1]/input")).sendKeys(c1);
                // password
driver.findElement(By.xpath("/html/body/div/div/div/div/div/div/div[2]/di
v/form/div[2]/input")).sendKeys(c2);
                // click query
driver.findElement(By.xpath("/html/body/div/div/div/div/div/div/div[2]/di
v/form/div[3]/input")).click();
                // query text
                String text =
driver.findElement(By.xpath("/html/body/div/div/div/div/div/div/div[2]/di
v/form/div[5]/code")).getText();

                datas.add(new Object[]{text, c2});
                row++; // next row
            }
```

```
        } catch (IOException e) {
            e.printStackTrace();
        }

        return datas;
    }

    @After
    public void tearDown() {
        driver.quit(); // close driver
    }

    @Test
    public void seleniumTest() {
        Assert.assertEquals(this.expected, this.actual);
    }
}
```

6. Operation Result

During test time, chrome look like below fiure:



After test finished, ‘Tests passed’ is exhibited on console. So each student’s information is consistent with the information on the website.