

# 实验1 完成一个简单的测试过程

## 0. 写在最前面

### 1. 实验目的

### 2. 实验内容

### 3. 实验过程

#### 3.1. 登录界面

功能点分析

输入/输出以及其他操作

设计测试用例并进行测试

异常/缺陷记录表

#### 3.2. 浏览购物主页

功能点分析

输入/输出及其他操作

设计测试用例并进行测试

异常/缺陷记录表

#### 3.3. 购物车管理页面

功能点分析

输入/输出以及其他操作

设计测试用例并进行测试

异常/缺陷记录表

#### 3.4. 用户信息填写界面

功能点分析

输入/输出及其他操作

设计测试用例并进行测试

异常/缺陷记录表

#### 3.5. 结算信息界面

功能点分析

输入/输出及其他操作

设计测试用例并进行测试

异常/缺陷记录表

#### 3.6. 订单完成提示界面

功能点分析

输入/输出及其他操作

设计测试用例并进行测试

异常/缺陷记录表

### 4. 实验结果

#### 4.1. 不同用户的不同反馈结果

#### 4.2. 总结与感受

## 0. 写在最前面

在ISO 26262中对Error/Fault/Failure的定义如下：

- 差错（Error）：计算的、观测的、测量的值或条件与真实的、规定的、理论上正确的值或条件之间的差异。
- 缺陷（Fault）：可引起要素（element）或相关项(item) 失效（fail）的异常情况；
- 失效（Failure）：由于缺陷而导致要素（element）或相关项(item)预期功能的终止。

也就是说Failure是要在明确预期功能的前提下进行判断的。这个实验最大的问题就是没有提供需求文档和设计文档。这样我就无法得知原始的需求以及对应的功能应该是怎样的。后面所有的分析都是基于一般购物网站的基础功能进行的分析。

## 1. 实验目的

- (1) 对软件测试有一个直观的感受。
- (2) 对软件测试过程有一个完整的感受。
- (3) 加强对基本概念的理解。

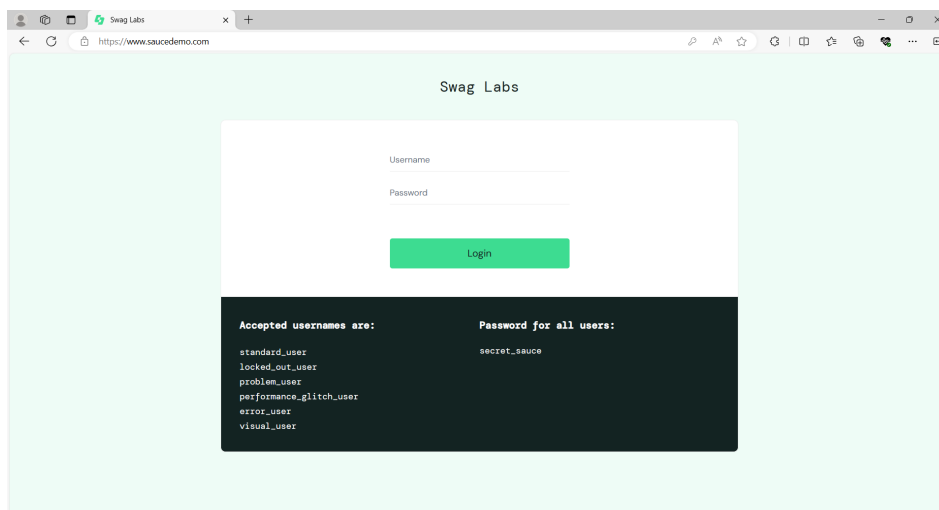
## 2. 实验内容

- (1) 针对 <https://www.saucedemo.com> 进行分析。
- (2) 先做一些初步的功能测试分析，如了解功能操作的路径，有哪些输入数据，有哪些特殊、异常的数据或操作。
- (3) 基于上述的分析，像用户使用产品操作软件一样，进行手工测试，发现缺陷并记录。
- (4) 完成一个非规范的测试报告。

## 3. 实验过程

接下来我将分不同功能点，列出其对应的测试场景、输入数据、异常/缺陷记录列表。

### 3.1. 登录界面



#### 功能点分析

这里我翻这个网页的源代码看了（js），打开网页后唯一一个可以交互的按钮就是Login，唯一可以交互的输入框就是Username和Password，这样排除了可能存在按钮或输入框十分隐蔽没有被用户发现的情况。

```
<div className="login_wrapper">
  <div className="login_wrapper-inner">
    <div id="login_button_container" className="form_column">
      <div className="login-box">
        <form onSubmit={handleSubmit}>
          <InputError
            isError={Boolean(error)}
            type={INPUT_TYPES.TEXT}
            value={username}
            onChange={handleUserChange}
            testId="username"
            placeholder="Username"
            // Custom
            id="user-name"
            name="user-name"
            autoComplete="off"
            autoCapitalize="none"
          />
        </form>
      </div>
    </div>
  </div>
</div>
```

很显然这就是个比较trivial的登录界面，**唯一的功能就是用户输入用户名与密码，然后点击登录获得反馈（登陆成功/不成功）。**

输入/输出以及其他操作

操作：

- 1. 输入用户名
- 2. 输入密码
- 3. 点击Login按钮
- 4. 浏览其他内容

输入/输出：

- 1. 对于用户名：输入类型为字符串
  - a. 输入Accepted username列表中的一个用户名
  - b. 输入一个不属于Accepted username中的用户名
  - c. 不输入用户名（或者说输入一个空字符串）
- 2. 对于密码：输入类型为字符串
  - a. 输入Password for all users中的公用密码
  - b. 输入一个和公用密码不同的密码
  - c. 不输入密码（或者说输入一个空字符串）
- 3. 对于Login按钮：没有输入内容的类型
  - a. 点击Login按钮，观察输出
  - b. 连续单击Login按钮（测试防抖），观察输出
  - c. 不点击Login按钮，观察输出
  - d. 按一下回车键，观察输出

设计测试用例并进行测试

对于每个测试用例，我无法把所有的界面截图全部截出来，否则整个测试文档就会变得非常冗长。如果你完全按照我撰写的操作文档尝试复现，你会看到我记录的结果是真实的。

输入						
Username	standard_username	standard_username	standard_username	standard_username	mamba	standard_username
Password	secret_sauce	whatCanIsay	secret_sauce		secret_sauce	secret_sauce
Login按钮	单击	单击	不点击	单击	单击	连续单击
输出						
	页面跳转到/inventory.html	错误提示：Epic sadface: Username and password do not match any user in this service	无其他输出	错误提示：Epic sadface: Password is required	错误提示：Epic sadface: Username and password do not match any user in this service	页面跳转到/inventory.html

输入	
Username	字符全部都是'a'，长度为1e5+7的字符串
Password	secret_sauce
Login按钮	单击
输出	

错误提示：Epic sadface: Username and password do not match any user in this service

这里我写了个py脚本，用来生成一个非常长的字符串：一个字符全部都是'a'，长度为1e5+7的字符串。用来作为用户名或密码的输入。

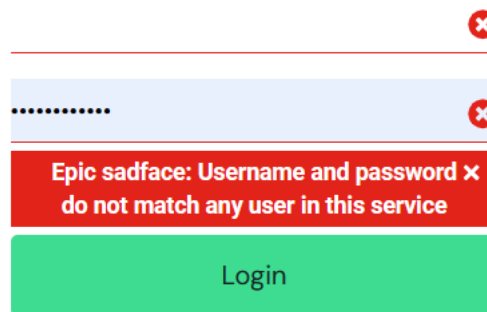
```
time = pow(10, 5) + 7

# 需要自己在代码工作的目录下建立好manyA/manyA.txt文件
with open('./manyA/manyA.txt', 'a') as file:
    for i in range(1, time):
        file.write('a')
```

一般来说字符串除了空串，没有什么边界值，因为字符串从理论上来说就是可以包含任意字符，且长度任意长的。

可以看出来，这里的功能即，输入用户名与密码，如果用户名和密码都已经录入过数据库，且匹配，那么登录成功，否则失败。

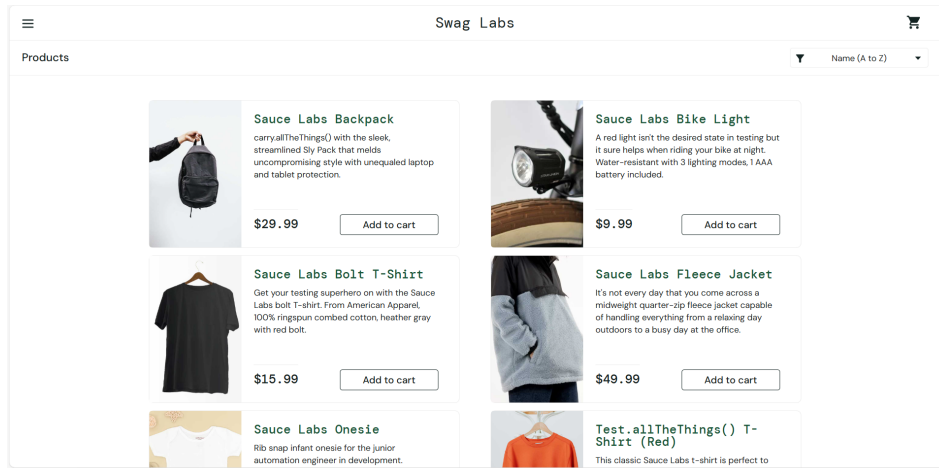
值得注意的是，这里最后一个测试用例，在我把长度为1e5+7的字符串复制输入输入框后，后半部分的字符不再显示，这应该是前端做了限制，不一定属于缺陷。



### 异常/缺陷记录表

比较遗憾的是，这个模块的功能点均通过了测试。虽然说有人认为一个成功的测试要找出系统的缺陷，但不一定是所有模块在事实上都一定有缺陷的。

## 3.2. 浏览购物主页



## 功能点分析

### 1. 商品购买功能

- 点击Add to cart按钮，可以将某项商品加入购物车，同时右上角购物车内商品数量+1
- 当一个商品已经加入购物车后，对应的Add to cart按钮将变为Remove按钮
- 点击Remove按钮，该项商品将移出购物车，同时右上角购物车内商品数量-1
- 点击绿色字体的商品名称，还可以进入到商品的

### 2. 商品排序功能

- 右上角排序可以选择4个顺序，分别为
  - 按名称升序
  - 按名称降序
  - 按价格升序
  - 按价格降序

### 3. 其他功能

- 点击左上角三个横线的按钮，将拖出一个侧边栏

All Items

About

Logout

Reset App State



Sauce Labs

carry.allTheThin  
streamlined Sly  
uncompromising  
and tablet prot

\$29.99

总计有这些子功能：

- i. All Items: 单击后显示所有商品
- ii. About：跳转[Sauce Labs: Cross Browser Testing, Selenium Testing & Mobile Testing](#)官网
- iii. Logout：退出登录
- iv. Reset the App State：清空购物车

## 输入/输出及其他操作

对于每一个按钮，都只有两种输入：

1. 单击（可能多次）
2. 不单击

这个页面没有任何输入框。

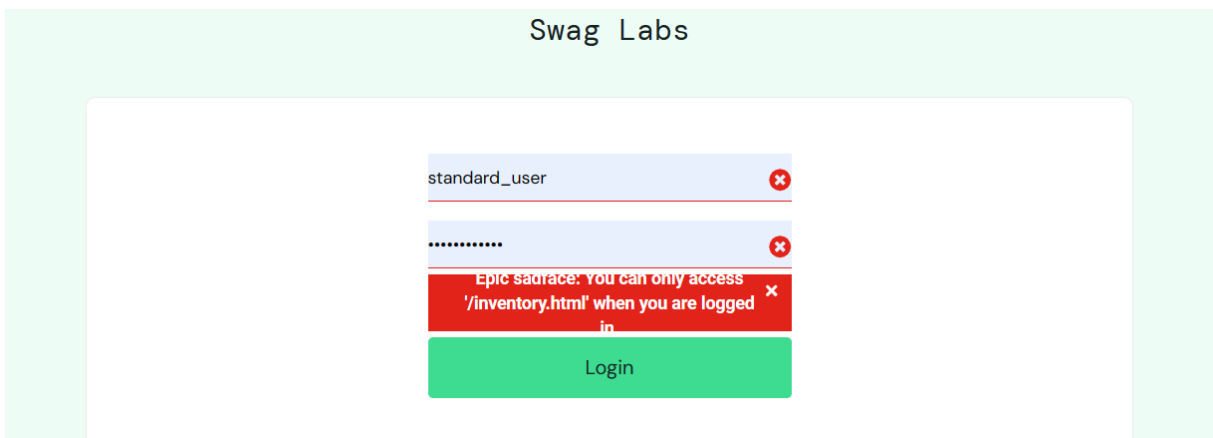
对于每个按钮点击后的输出，在功能点分析中已经有所介绍。

## 设计测试用例并进行测试

这里的按钮非常复杂，全部组合起来将生成非常多的测试用例，这里我将挑选一些具有典型、代表性的测试用例进行测试：

1. 添加单个商品测试
  - a. 输入：选择一个商品，点击一次Add to cart
  - b. 输出：Add to cart变为Remove，右上角购物车内商品数+1
2. 添加多个商品测试
  - a. 输入：选择不小于一个商品，分别点击Add to cart
  - b. 输出：选择过的Add to cart均变为Remove，右上角购物车内商品数为选择商品的总数
3. 商品移出购物车测试
  - a. 输入：点击Remove按钮
  - b. 输出：Remove按钮变回Add to cart按钮，右上角购物车内商品数-1（当商品数为0时，不存在Remove按钮）
4. 排序功能测试
  - a. 按名称升序测试：

- i. 输入：保持页面不变即可，初始状态即为按名称升序
  - ii. 输出：所有商品按商品名称的字典序升序排列
- b. 按名称降序测试
  - i. 输入：选择Name(Z to A)
  - ii. 输出：所有商品按商品名称的字典序降序排列
- c. 按价格升序测试
  - i. 输入：选择Price(low to high)
  - ii. 输出：所有商品按价格升序
- d. 按价格降序测试
  - i. 输入：选择Price(high to low)
  - ii. 输出：所有商品按价格降序
- 5. 商品详情查看测试
  - a. 输入：点击某个绿色字体的商品名称
  - b. 输出：跳转到商品详情页（在这个商品详情页上进行Add to cart和Remove也是没问题的）
- 6. 官网查看重定向测试：
  - a. 输入：点击About按钮
  - b. 输出：跳转[Sauce Labs: Cross Browser Testing, Selenium Testing & Mobile Testing](#)官网
- 7. 登出测试
  - a. 输入：点击Logout按钮
  - b. 输出：回退到登录页面。并且如果此时手动尝试重定向到inventory.html将给出错误报告：**Epic sadface: You can only access '/inventory.html' when you are logged in.**



- 8. 重置购物车状态测试：
  - a. 在购物车为空时：
    - i. 输入：点击Reset App State按钮
    - ii. 输出：无变化
  - b. 在购物车不为空时
    - i. 输入：点击Reset App State按钮
    - ii. 输出：购物车清空，但Remove按钮没有变回Add to cart

9. 购物车持久化测试：
- a. 查看About对持久化的影响：
    - i. 输入：在购物车不为空的情况下，点击About按钮，再返回到购物页面
    - ii. 输出：购物车状态无变化（保证了持久化）
  - b. 查看Logout对持久化的影响
    - i. 输入：在购物车不为空的情况下，点击Logout按钮，然后再重新登陆
    - ii. 输出：购物车状态无变化
  - c. 查看排序对持久化的影响
    - i. 输入：在购物车不为空的情况下，选择某种排序规则
    - ii. 输出：购物车状态无变化

异常/缺陷记录表

本测试模块的缺陷在于：Reset App State按钮虽然清空了购物车，但却没有将Remove按钮变回Add to cart按钮，可能会误导用户。

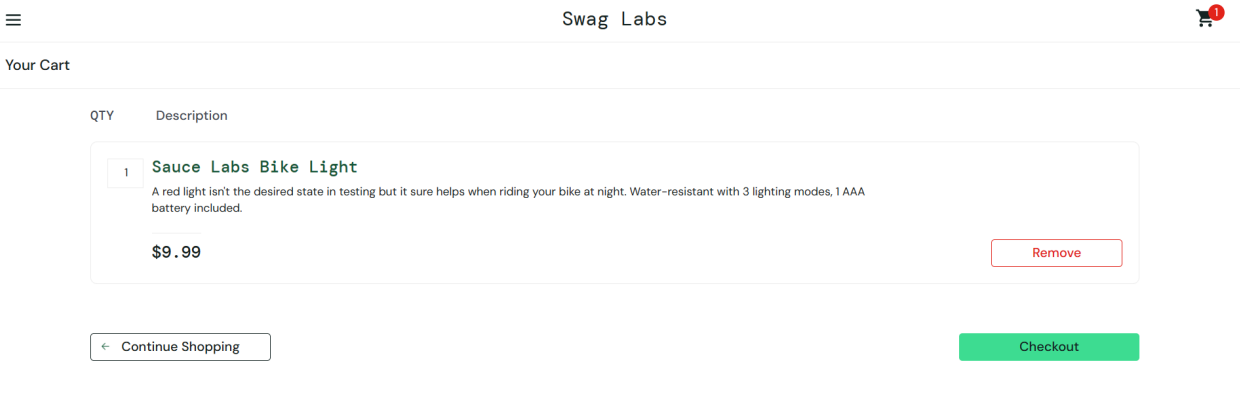
异常/缺陷	复现方式
Reset App State不会重置按钮状态	先随意添加一个/些商品到购物车，点击Reset App State按钮，这些商品的Remove按钮没有变回Add to cart按钮

备注：本测试模块中，还有twitter\facebook\linkedin等国外企业的超链接，由于我国法律限制，我就不访问这些外国企业的网站了。

又：**排序功能是不具备持久化的**。如果你利用About这样的按钮跳转了其他网页，再访问购物页面时，你会发现商品的顺序又变回了默认排序。我不认为这是缺陷/异常，因为这不是必需的功能。可能最初的设计就是这样的，抑或是需求文档中没有对这一块的要求。总之，这不是缺陷，这就是一个很普通的feature。

3.3. 购物车管理页面

在购物主页的右上角，点击购物车，可以进入购物车管理页面。



功能点分析

- 这个界面具有以下功能点：
- 1. 返回购物主页功能：点击Continue Shopping可以返回购物主页
  - 2. 进入用户信息填写界面功能：点击Checkout进入用户信息填写界面
  - 3. 移除购物车内商品功能：点击Remove按钮可以移除已经选入到购物车的商品
- 侧边栏功能在上一个测试模块中已经介绍，这里不再赘述。



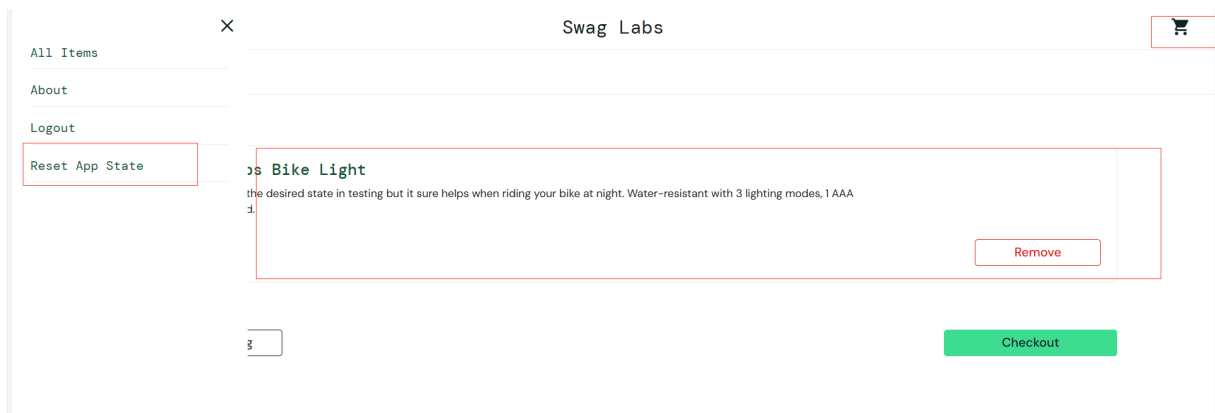
## 输入/输出以及其他操作

这里的输入仅来自于按钮的点击，并不含有输入框。

对于每个按钮点击后的输出，功能点分析中已经介绍。

## 设计测试用例并进行测试

1. 返回购物主页测试
  - a. 输入：点击Continue Shopping按钮
  - b. 输出：返回到购物主页
2. 进入用户信息填写界面测试：
  - a. 输入：点击Checkout按钮
  - b. 输出：进入到用户信息填写界面
3. 移除购物车内商品测试：
  - a. 输入：点击Remove按钮，并返回到购物主页
  - b. 输出：该商品条目立刻从购物车内移除，并且购物主页对应商品的按钮变回Add to cart
4. 利用Reset App State进行购物车一键清除测试：
  - a. 输入：点击Reset App State按钮，并返回到购物主页
  - b. 输出：右上角购物车内的数字清空，但购物车管理界面的所有商品仍保留，且返回到购物主页后对应的商品的Remove按钮没有重置。不过立刻刷新这个页面后，购物车内的商品条目就会被清空。



5. 购物车商品条目信息测试：
  - a. 输入：添加多个商品到购物车，然后进入购物车管理界面
  - b. 输出：所有商品前全部都有一个数字1，根据QTY为quantity的缩写，判断这个数字是商品的数量，但这个数字不具备任何可供修改的操作。

QTY	Description
1	<b>Sauce Labs Bike Light</b> A red light isn't the desired state in testing but it sure helps when riding your bike at night. Water-resistant with 3 lighting modes, 1 AAA battery included. \$9.99 <a href="#">Remove</a>
1	<b>Sauce Labs Backpack</b> carry.allTheThings() with the sleek, streamlined Sly Pack that melds uncompromising style with unequaled laptop and tablet protection. \$29.99 <a href="#">Remove</a>
1	<b>Sauce Labs Bolt T-Shirt</b> Get your testing superhero on with the Sauce Labs bolt T-shirt. From American Apparel, 100% ringspun combed cotton, heather gray with red bolt. <a href="#">Remove</a>

异常/缺陷记录表

本测试模块的缺陷在于：Reset App State按钮虽然清空了右上角购物车内的商品数量，但没有实时地清空购物车，可能会误导用户。

异常/缺陷	复现方式
Reset App State不会重置购物车内商品跳舞	先随意添加一个/些商品到购物车，点击Reset App State按钮，购物车内的商品条目不会被实时地清空，需要刷新一次页面才会看到被清空

备注：对于某类商品，其购买数量是一个0-1值，也即要么不买，要买就买一件。但是，没有需求文档和设计文档，我无法得知这是功能还是缺陷，也许需求上就是这样规定的。我无法妄自揣测，把这一项当作一个缺陷填写到表格中。

3.4. 用户信息填写界面

点击Checkout，可以进入用户信息填写界面。

Checkout: Your Information

First Name

Last Name

Zip/Postal Code

Cancel

Continue

功能点分析

- 1. 返回购物车管理界面：点击Cancel按钮即可返回购物车管理界面
- 2. 完成结算按钮：点击Continue按钮可以完成结算
- 3. 填写用户信息：
  - a. 填写First Name
  - b. 填写Last Name
  - c. 填写邮政编码

输入/输出及其他操作

操作：

- 1. 输入First Name
- 2. 输入Last Name
- 3. 输入邮政编码
- 4. 点击Cancel按钮
- 5. 点击Continue按钮

输入/输出：

- 1. 对于First Name/Last Name/邮政编码：输入类型均为字符串
  - a. 输入/不输入First Name
  - b. 输入/不输入Last Name
  - c. 输入/不输入邮政编码

设计测试用例并进行测试

输入				
First Name	不输入	任意	任意	字符全部都是'a'，长度为1e5+7的字符串
Last Name	任意	不输入	任意	字符全部都是'a'，长度为1e5+7的字符串
Postal Code	任意	任意	不输入	字符全部都是'a'，长度为1e5+7的字符串
输出				
	错误提示：Error: First Name is required	错误提示：Error: Last Name is required	错误提示：Error: Postal Code is required	无特别输出

异常/缺陷记录表

虽然这个测试模块提供了输入的非空校验。但却没有提供合法性校验，因此缺陷基本上出在了合法性校验上。


异常/缺陷	复现方式
First Name输入缺少合法性校验	输入一个字符全部都是'a'，长度为1e5+7的字符串，显然全世界没有人的名字是这个，但是可以通过合法性校验
Last Name输入缺少合法性校验	输入一个字符全部都是'a'，长度为1e5+7的字符串，显然全世界没有人的名字是这个，但是可以通过合法性校验
Postal Code输入缺少合法性校验	输入一个字符全部都是'a'，长度为1e5+7的字符串，显然全世界没有人一个地区的邮政编码是这个，但可以通过合法性校验。

3.5. 结算信息界面

在填写信息界面点击Continue，可以进入结算信息界面。

☰

Swag Labs



Checkout: Overview

QTY	Description
1	<div><div>Sauce Labs Backpack</div><div>carryallTheThings() with the sleek, streamlined Sly Pack that melds uncompromising style with unequaled laptop and tablet protection.</div><div>\$29.99</div></div>

Payment Information

SauceCard #31337

Shipping Information

Free Pony Express Delivery!

Price Total

Item total: \$29.99

Tax: \$2.40

Total: \$32.39

← Cancel

Finish

## 功能点分析

1. 返回购物页面功能：点击Cancel，返回购物主页
2. 完成订单提交功能：点击Finish，完成订单提交
3. 查看订单信息功能：
  - a. 支付信息
  - b. 运输信息
  - c. 总金额
    - i. 商品金额
    - ii. 税收金额

## 输入/输出及其他操作

所有输入都来自于按钮点击，无输入框。

对应的输出在功能点介绍中已经涉及。

## 设计测试用例并进行测试

1. 返回购物页面测试：
  - a. 输入：点击Cancel
  - b. 输出：返回购物主页
2. 完成订单提交测试：
  - a. 输入：点击Finish
  - b. 输出：完成订单提交
3. 总金额计算测试：
  - a. 输入：无
  - b. 输出：观察Item total项，计算正确。
4. 订单信息测试：
  - a. 输入：无
  - b. 输出：观察Payment Information，发现所有订单都是SauceCard #31337

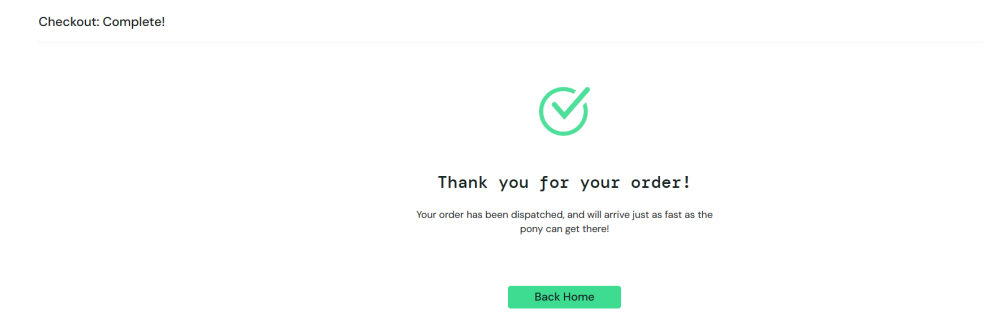
备注：对于tax，由于缺少需求文档和设计文档，我无法得知具体的计算公式，因此无法判断这是否是一个缺陷。

## 异常/缺陷记录表

异常/缺陷	复现方式
订单ID不具备唯一性	所有订单ID均为SauceCard #31337

### 3.6. 订单完成提示界面

点击Finsih，即可进入该界面。



#### 功能点分析

- 1. 返回购物主页功能：点击Back Home，返回购物主页。

#### 输入/输出及其他操作

唯一输入来自点击Back Home按钮。

唯一输出是返回到购物主页。

#### 设计测试用例并进行测试

- 1. 返回购物主页测试：
  - a. 输入：点击Back Home
  - b. 输出：返回购物主页

#### 异常/缺陷记录表

无

## 4. 实验结果

### 4.1. 不同用户的不同反馈结果

该网站提供了很多不同的用户。以上的测试都是基于一个叫做standard\_user进行的。实际上当我对其他用户进行测试时，基于同一个测试用例得到了不同的测试结果。

例如对于problem\_user，在填写用户表单时无法填入Last Name，一旦填入就会被自动清除：

aaa

Last Name

aaa

Error: Last Name is required

包括所有商品图片全部被替换成了一只衔着棒球的斗牛犬：

Sauce Labs Backpack

carry.allTheThings() with the sleek, streamlined Sly Pack that melds uncompromising style with unequaled laptop and tablet protection.

\$29.99

Remove

Sauce Labs Bike Light

A red light isn't the desired state in testing but it sure helps when riding your bike at night. Water-resistant with 3 lighting modes, 1 AAA battery included.

\$9.99

Remove

Sauce Labs Bolt T-Shirt

Get your testing superhero on with the Sauce Labs bolt T-shirt. From American Apparel, 100% ringspun combed cotton, heather gray with red bolt.

\$15.99

Add to cart

Sauce Labs Fleece Jacket

It's not every day that you come across a midweight quarter-zip fleece jacket capable of handling everything from a relaxing day outdoors to a busy day at the office.

\$49.99

Add to cart

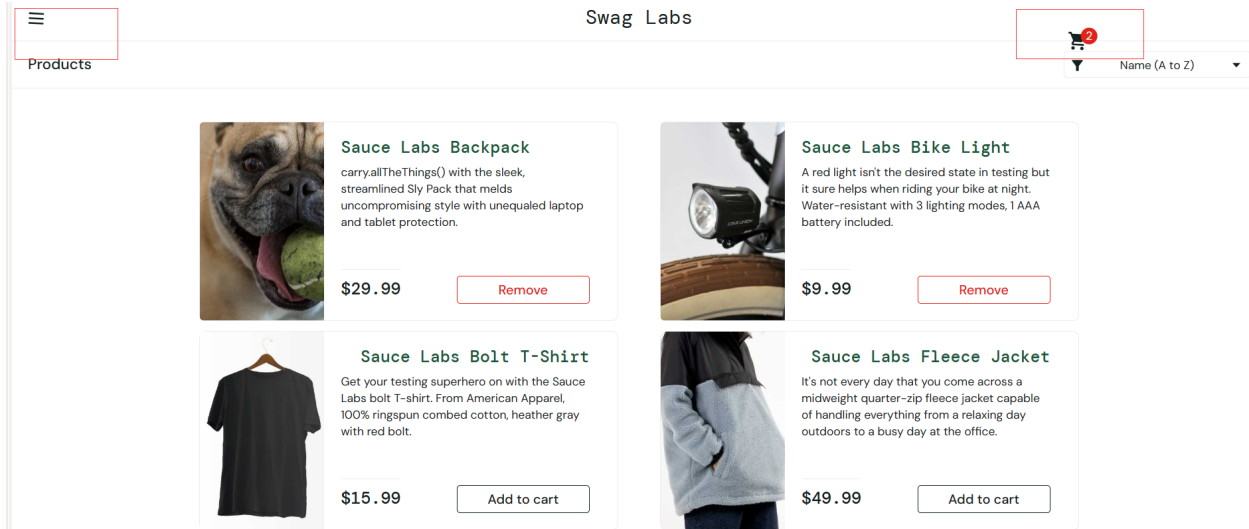
而对于error\_user，他甚至不需要填完表单就能够进入结算。例如他可以缺省Last Name这一项。

而对于不同的用户，购物车信息居然是互通的，也就是standard\_user和error\_user共享了一个购物车。

对于visual\_user，他看到的界面的按钮和图标都是歪的，偏离原位置的：

实验1 完成一个简单的测试过程

14



这些都是这个网站的缺陷（因为这肯定不符合一般购物网站的基础功能）

## 4.2. 总结与感受

本次测试，我分析了这个网站各个模块的功能点以及输入/输出，设计了测试用例，并针对不同用户进行了测试，发现了该网站的一些功能上的缺陷。

对于这个网站的设计，我有一些想法如下：

1. 缺少了用户注册功能
2. 某类商品在一次购买时，数量至多只能为1
3. 用户信息填写不具备唯一性：例如我叫做Gilfoyle Li，邮政编码250000。那么该地区可能有多个叫做Gilfoyle Li的人，到底寄给谁呢？还是说给所有叫做Gilfoyle Li的人都发一份商品？
4. Tax计算没有细则
5. Token时限很短，大概不操作5分钟就会被刷新掉token导致重新登陆。用户之前的操作可能未保存。例如一个用户本来在填写表单，token刷新后他被迫登出，重新登入后这个表单的内容就没了。

以上都不能明确地算是缺陷，因为可能需求上本来如此。

在进行完测试后，我深感手动测试的不易。因为即便是如此简单的一个网站，将所有输入组合在一起，也会生成一个很庞大的测试用例集合。更不要说那些拿上生产环境的网站，想要做到某项指标覆盖率为100%的手动测试，实在太困难。我也了解到利用脚本和自动测试框架进行测试的重要性。

另外，在测试过程中，我尝试利用浏览器开发工具修改网页内容，例如修改商品购买数量为一个指定的数字。但我查阅了软件测试的定义：

使用人工或自动的手段来运行或测定某个系统的过程，其目的在于检验它是否满足规定的需求或弄清预期结果与实际结果之间的差别。它是软件开发过程的重要组成部分，是软件质量保证的关键环节。

我们要运行或测定这个系统，是**在他已有的功能上测试**，而不是自己修改了他提供的功能后再测试，后者叫做hack，是一种带有破坏性质的行为，这在软件测试中是不被允许的。