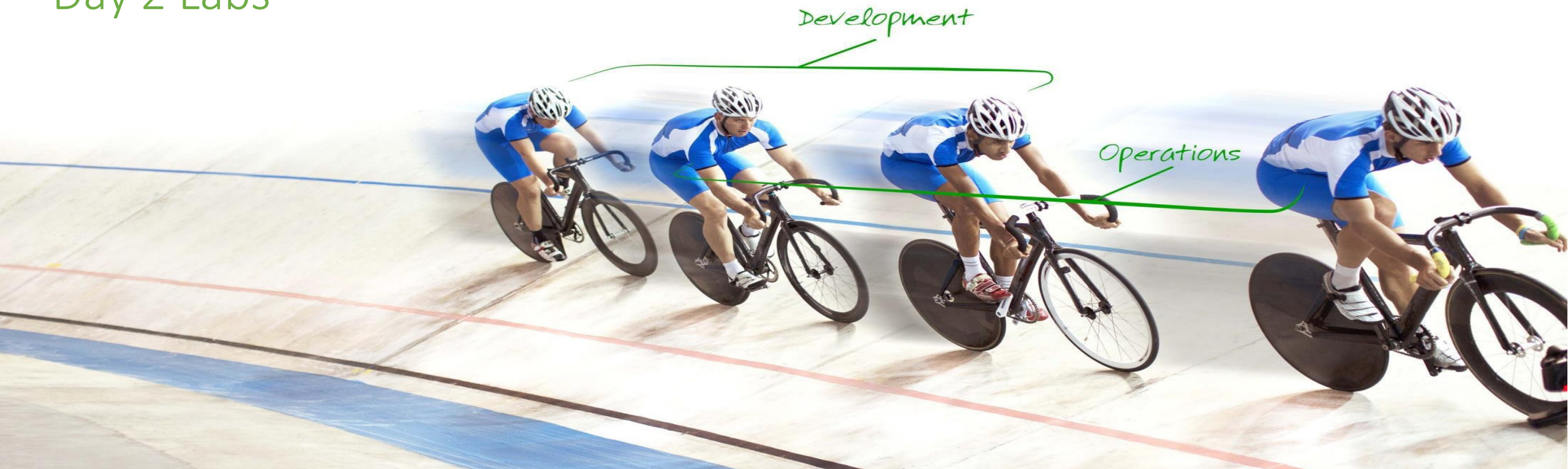




Behavior Driven Development (BDD) Day 2 Labs





Micro Lab #3 – Creating a Background

Micro Lab #3 – Add a Background

Step 1 – Refactor browsing around feature

- A `Background` is used when the same Gherkin statements are repeated at the beginning of each scenario
- Notice how the following statements are repeatedly used for each scenario

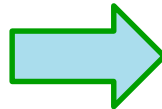
```
Given I have opened the browser
And I am on the home page
```

1. Add a background section within the “browsing_around.feature”
2. Run the test and ensure it passes

```
Feature: Browsing around

Scenario: Login and Vets
  Given I have opened the browser
  And I am on the home page
  When I open veterinarians page
  Then I search for text "Veterinarian"

Scenario: Login and check owner based on last name
  Given I have opened the browser
  And I am on the home page
  When I search owner "Franklin"
  Then I get owner "Franklin" Informations
```



```
Feature: Browsing around

Background:
  Given I have opened the browser
  And I am on the home page

Scenario: Login and Vets
  When I open veterinarians page
  Then I search for text "Veterinarian"

Scenario: Login and check owner based on last name
  When I search owner "Franklin"
  Then I get owner "Franklin" Informations
```

Micro Lab #3 – Add a Background

Step 2 – View the updated Cucumber report

1. Using Windows Explorer navigate to the **cucumber-pretty.html** file on your workstation, located at `{workspace_location}\target\cucumber-reports` and open the cucumber-pretty.html file
2. You should see the below Cucumber report

Feature: Browsing around **Background:**

- ✓ **Given** I have opened the browser
- ✓ **And** I am on the home page

Scenario: Login and Vets

- ✓ **When** I open veterinarians page
- ✓ **Then** I search for text "Veterinarian"

Scenario: Login and check owner based on last name

- ✓ **When** I search owner "Franklin"
- ✓ **Then** I get owner "Franklin" Informations



Micro Lab #4 – Add a Scenario Outline

Micro Lab #4 – Add a Scenario Outline

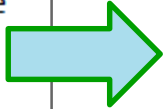
Step 1 – Scenario outline requirements

- A Scenario Outline is used when the same scenario is repeated with different datapoints
- Notice how following statements are used for each scenario

Steps :

1. Add a scenario outline section for “browsing_around.feature”
2. Run the test and ensure it passes

```
Scenario: Login and check owner based on last name
  When I search owner "Franklin"
  Then I get owner "Franklin" Informations
```



```
Scenario Outline: Login and check owner based on last name
  When I search owner "<username>"
  Then I get owner "<username>" Informations
```

Examples:

```
|username|
|Franklin|
|Black|
|Coleman|
```

Micro Lab #4 – Add a Scenario Outline

Step 2 – View the updated Cucumber report

1. Using Windows Explorer navigate to the **cucumber-pretty.html** file on your workstation, located at `{workspace_location}\target\cucumber-reports` and open the cucumber-pretty.html file
2. You should a similar Cucumber report:

Feature: Browsing around

Background:

- ✓ **Given** I have opened the browser
- ✓ **And** I am on the home page

Scenario: Login and Vets

- ✓ **When** I open veterinarians page
- ✓ **Then** I search for text "Veterinarian"

Scenario Outline: Login and check owner based on last name

- ✓ **When** I search owner "<username>"
- ✓ **Then** I get owner "<username>" Informations

Examples:

	username
✓	Franklin
✓	Black
✓	Coleman



Micro Lab #5 – Add Tags

Micro Lab #5 – Add Tags

Step 1 – Update existing feature with Tags

Tags provide a way to organize scenario execution

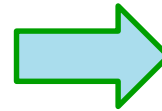
1. Add two tags: @SmokeTest and @RegressionTest in “browsing_around.feature”

```
Feature: Browsing around

Background:
  Given I have opened the browser
  And I am on the home page

Scenario: Login and Vets
  When I open veterinarians page
  Then I search for text "Veterinarian"

Scenario Outline: Login and check owner based on last name
  When I search owner "<username>"
  Then I get owner "<username>" Informations
|
Examples:
|username|
|Franklin|
|Black|
|Coleman|
```



```
Feature: Browsing around

Background:
  Given I have opened the browser
  And I am on the home page

@SmokeTest
Scenario: Login and Vets
  When I open veterinarians page
  Then I search for text "Veterinarian"

@RegressionTest
Scenario Outline: Login and check owner based on last name
  When I search owner "<username>"
  Then I get owner "<username>" Informations

Examples:
|username|
|Franklin|
|Black|
|Coleman|
```

Micro Lab #5 – Add Tags

Step 2 – Update Runner class

1. Open ~\testPetClinic\src\test\java\testPetclinic\testPetclinic\runTest.java
2. Update Cucumber.Options
3. Add **tags = "@SmokeTest and not @RegressionTest"** and save changes

```
@RunWith(Cucumber.class)

@CucumberOptions(features = "src/test/resources/features/", glue = { "stepdefs" }, plugin = { "pretty",
    "html:target/cucumber-reports/cucumber-pretty", "json:target/cucumber-reports/CucumberTestReport.json",
    "junit:target/cucumber-reports/cucumber.xml",
    "rerun:target/cucumber-reports/rerun.txt" })
```



```
@RunWith(Cucumber.class)

@CucumberOptions(features = "src/test/resources/features/", glue = { "stepdefs" }, plugin = { "pretty",
    "html:target/cucumber-reports/cucumber-pretty", "json:target/cucumber-reports/CucumberTestReport.json",
    "junit:target/cucumber-reports/cucumber.xml",
    "rerun:target/cucumber-reports/rerun.txt" }, tags = "@SmokeTest and not @RegressionTest")
```

Micro Lab #5 – Add Tags

Step 3 – View the updated Cucumber report

1. Using Windows Explorer navigate to the **cucumber-pretty.html** file on your workstation, located at `{workspace_location}\target\cucumber-reports` and open the cucumber-pretty.html file
2. You should a similar Cucumber report:

Feature: Browsing around

Background:

✓ **Given** I have opened the browser

✓ **And** I am on the home page

@SmokeTest

Scenario: Login and Vets

✓ **When** I open veterinarians page

✓ **Then** I search for text "Veterinarian"

@RegressionTest

Scenario Outline: Login and check owner based on last name

? **When** I search owner "<username>"

? **Then** I get owner "<username>" Informations

Examples:

	username
?	Franklin
?	Black
?	Coleman

Micro Lab #5 – Using tags

Step 4 – Experiment with different options

1. Open `~\testPetClinic\src\test\java\testPetclinic\testPetclinic\runTest.java`
2. Update Cucumber.Options `tags = "@SmokeTest or @RegressionTest"` save changes and run tests to see results

Micro Lab #6 – Add a Hook

Step 1 – Add a hook to close out the browser

At this point, we're probably tired of closing out all the browser windows our test automation is opening. We can address this with a hook and more automation.

1. At the bottom of your step-definition file, let's add an @After hook to cleanup all the browsers we opened:

```
@After
public void cleanUp() {
    driver.close();
}
```

2. Re-run your automation test and now you will see your browsers close once the automation has completed

Micro Lab #6 – Add a tagged Hook

Step 1 – Add a tagged hook

Sometimes we may want to perform some steps before or after a scenario for a particular tag. We can add tagged or conditional hooks for the same as shown below

1. At the bottom of your step-definition file, let's add an @Before hook to add a tag hook.

```
@Before(value = "@SmokeTest")
public void stepsBeforeSmokeTest() {
    System.out.println("Some steps before running smoke test ");
}
```



Lab #2: Add New Scenarios and Step Definitions

Lab #2 - Add New Scenarios and Step Definitions

Use the skills you've learned to add three new stories to the test project

JIRA-1235: Find all owners

Feature: A Vet can get a listing of all Pet Owners

As a Veterinarian

I Want To search with an empty string on the find owners page

So That I can retrieve a listing of all owners

JIRA-1236:

Feature: Owner does not exist prompt

As a Veterinarian

I Want To search for any Pet Owner's last name in our database

So That I will know if the Pet Owner exists or not

Reminder

Gherkin Template

Scenario: Find all owners

Given ???

When ???

And ???

Then ???

Lab #2 cont - Add New Scenarios and Step Definitions

Use the skills you've learned to add three new stories to the test project

Smoke Test Scenario

JIRA-1237: Logo and home page menu

Feature: Ensure logo and home page menu have not changed

As a Veterinarian

I Want To open the Pet Clinic website

So That I can view the Pet Clinic logo and main navigation menu

Reminder

Gherkin Template

Scenario: Home Page
smoke test

Given ???

When ???

And ???

Then ???

Lab #2– Add New Scenarios and Step Definitions

Step 2 – View the updated Cucumber report

Verify all your test passed by viewing the updated Cucumber report:

Feature: Browsing around

Scenario: The home page Menu's

- ✓ Given I have opened the browser
- ✓ When I am on the home page
- ✓ Then I should see "PetClinic"
- ✓ And I should see the Menu Tabs "Home" "Find owners" "Veterinarians" "Error"

Scenario: Find All Owners

- ✓ Given I have opened the browser
- ✓ And I am on the home page
- ✓ When I search owner " "
- ✓ Then I get a list of "Owners" Information
- ✓ And I should see Owners grid with valid results

Scenario: Find Owners with Wrong Last Name

- ✓ Given I have opened the browser
- ✓ And I am on the home page
- ✓ When I search owner "wrong"
- ✓ Then I should see error message 'has not been found'



LAB #3: Option 2 - Rock Paper Scissors

Lab #3 – Rock, Paper, Scissors

Implement with BDD and JUnit

User Story:

Feature: Play game Rock, Paper, Scissors

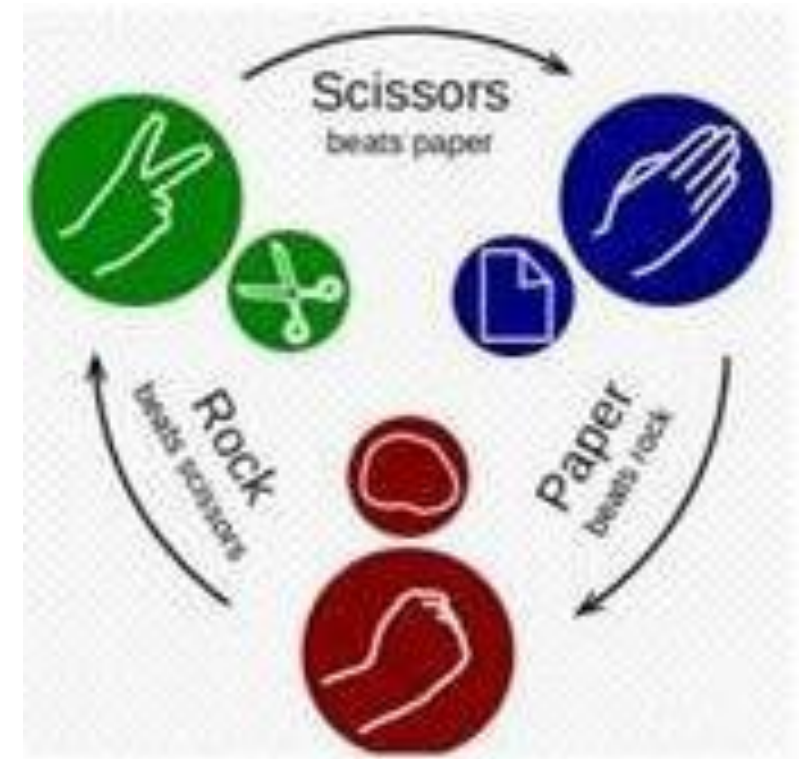
As a player,

I would like to throw rock, paper, or scissors

So that I might win the outcome

Rules:

Player1	Player2	Outcome	WINNER
PAPER	PAPER	BOTH SAME, ALWAYS TIE	TIE
ROCK	SCISSORS	ROCK BREAKS SCISSORS	P1
ROCK	PAPER	PAPER COVERS ROCK	P2
SCISSORS	PAPER	SCISSORS CUTS PAPER	P1
SCISSORS	ROCK	ROCK BREAKS SCISSORS	P2
PAPER	ROCK	PAPER COVERS ROCKS	P1
PAPER	SCISSORS	SCISSOR CUTS PAPER	P2



Lab #3 cont. – Rock, Paper, Scissors

Implement with BDD and JUnit

How to execute JUnit testing:

- Right-click on your runner file (runTest.java) and select “Run As” > “JUnit Test”

Hint:

Leverage the JUnit Assertions API for your “then” statements:

- `Assert.assertEquals({EXPECTED_OUTCOME}, {ACTUAL_OUTCOME});`



Micro Lab #7 – BDD Workflow (Optional – time permitting)

Micro Lab #7 – BDD Workflow

Scenario Background

The PetClinic has recently hired a marketing consultant to increase brand awareness and customer engagement

The first item they have asked to change is on the “Veterinarians” page.

- They think that shortening the on-page title from “Veterinarians” to “Vets” will make the site seem more friendly to Pet Owners / Customers.

The clinic owner is a fan of the change

- He called a front end developer into his office and had the developer make the change and push it to the production site, because “you should NOT need to run a bunch of tests for something so small.”
- Because the front end developer does not have visibility into the full suite of tests that are run, he makes the change in all environments, including the live site.

However, we know that there is a simple test that validates loading of the “Vets” page, that looks for the word “Veterinarians” on the page.

In this lab, we will simulate making this change outside of the normal process and running the full list of tests on the next code check-in and how we can change the test to reflect the new changes to the site.

Micro Lab #7 – BDD Workflow

Step 1 – Change the “Veterinarians” page to the “Vets” page

1. Navigate to your “spring-petclinic” project
2. Search for file “vetList.jsp” with the short-cut key shift+ctrl+R
 - Or navigate the directory path: “spring-petclinic\src\main\webapp\WEB-INF\jsp\vets\vetList.jsp”
3. Change line 18 from “**Veterinarians**” to “**Vets**”
4. Save the file.

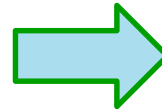
```
*vetList.jsp
<!DOCTYPE html>

<%@ page session="false" %>
<%@ taglib prefix="spring" uri="http://www.springframework.org/tags" %>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ taglib prefix="datatables" uri="http://github.com/dandelion/datatables" %>

<html lang="en">

<jsp:include page="../fragments/staticFiles.jsp"/>

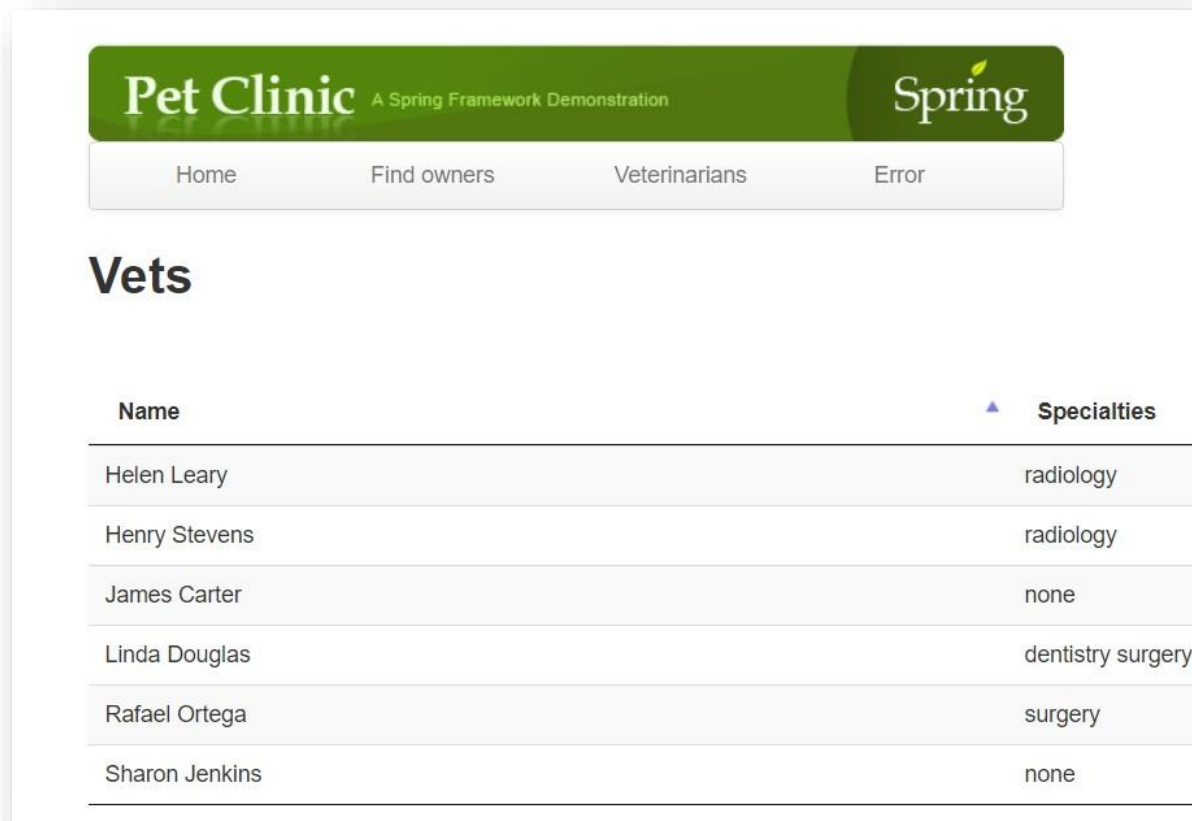
<body>
<div class="container">
  <jsp:include page="../fragments/bodyHeader.jsp"/>
  <h2>Veterinarians</h2>
```



```
12 <jsp:include page="../fragments/staticFiles.jsp"/>
13
14 <body>
15 <div class="container">
16   <jsp:include page="../fragments/bodyHeader.jsp"/>
17
18   <h2>Vets</h2>
```


Micro Lab #7 – BDD Workflow

Step 2 – Refresh Pet Clinic application in the browser



The screenshot shows the Pet Clinic application interface. At the top, there is a green header bar with the text "Pet Clinic" and "A Spring Framework Demonstration" on the left, and the "Spring" logo on the right. Below the header is a navigation bar with four links: "Home", "Find owners", "Veterinarians", and "Error". The main content area is titled "Vets" and contains a table of veterinarians. The table has two columns: "Name" and "Specialties". The table lists six veterinarians: Helen Leary (radiology), Henry Stevens (radiology), James Carter (none), Linda Douglas (dentistry surgery), Rafael Ortega (surgery), and Sharon Jenkins (none).

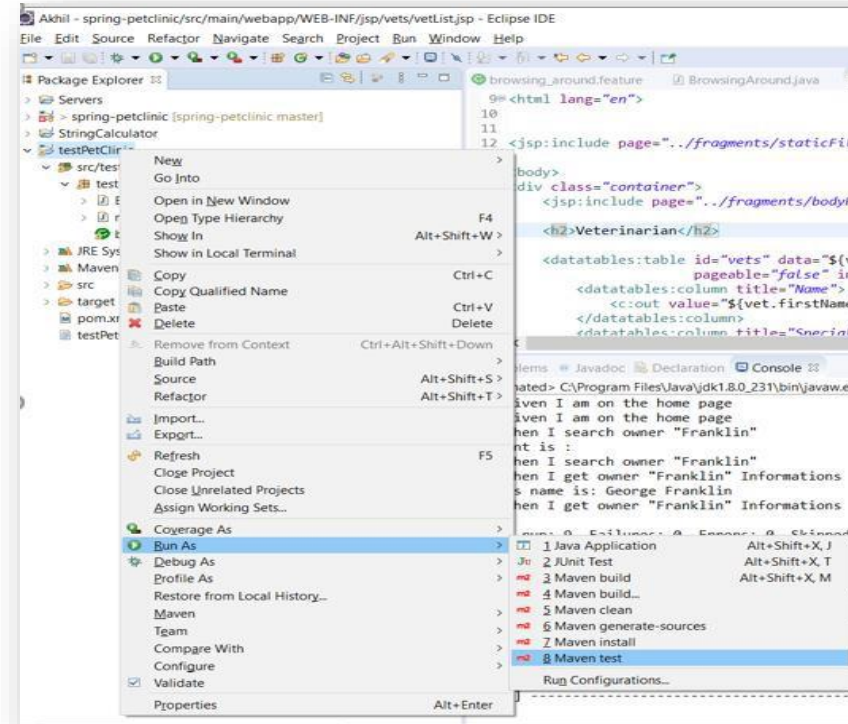
Name	Specialties
Helen Leary	radiology
Henry Stevens	radiology
James Carter	none
Linda Douglas	dentistry surgery
Rafael Ortega	surgery
Sharon Jenkins	none

Micro Lab #7 – BDD Workflow

Step 3 – Run all tests in the testPetClinic project

You will noticed the “Then” step – I find the page heading “**Veterinarians**” has failed.

1. Right click on the project testPetClinic => “Run as” -> “Maven Test”
2. Open the Cucumber cucumber-pretty.html file



Micro Lab #7 – BDD Workflow

Step 4 – Handle the failed test (Option #1)

- You check with the Product Owner, and she informs you that the PetClinic owner ordered that change personally, and it happened outside the normal process.
- We can reasonably assume that the CODE is correct and the TEST is not.
- Therefore, we update the feature file.

Feature: Browsing around

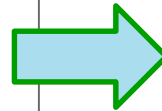
Background:

Given I have opened the browser
And I am on the home page

@SmokeTest

Scenario: Login and Vets

When I navigate to the veterinarians page
Then I find the page heading "Veterinarians"



Feature: Browsing around

Background:

Given I have opened the browser
And I am on the home page

@SmokeTest

Scenario: Login and Vets

When I navigate to the veterinarians page
Then I find the page heading "Vets"

Micro Lab #7 – BDD Workflow

Step 4 – Handle the failed test (Option #2)

- You check with the Product Owner, and she informs you that the PetClinic owner ordered that change personally, and it happened outside the normal process.
- However, the owner has reconsidered and thinks “Veterinarians” looks more professional on the site.
- We can reasonably assume that the CODE is incorrect and the TEST is correct.
- Therefore, we rollback the change to line 18 of vetList.jsp from “Vets” to “Veterinarians.”

End of Day 2 Document