

EECS 4313 Assignment 3

Data Flow Testing, Slice-Based Testing and Mutation Testing

Student Name — Student Number — EECS Account

Edward Vaisman — 212849857 — eddyv

Robin Bandzar — 212200531 — cse23028

Kirusanth Thiruchelvam — 212918298 — kirusant

Sadman Sakib Hasan — 212497509 — cse23152

April 4, 2018

Contents

1	BORG Calendar	3
2	JPetStore	4
2.1	Scenario 1: Returning User	5
2.1.1	Overview	5
2.1.2	Load Test Properties	7
2.1.3	Executing Load Test	7
2.1.4	Analysing Load Test	7
2.2	Scenario 2: New User	10
2.2.1	Overview	10
2.2.2	Load Test Properties	13
2.2.3	Executing Load Test	13
2.2.4	Analysing Load Test	13

List of Figures

1	Recording Controller for Returning User Scenario	6
2	View Results Tree for Returning User Scenario	7
3	Apache Access Log Snapshot for Returning User Scenario	8
4	Windows Performance Monitor Snapshot for Returning User Scenario .	9
5	JConsole Snapshot for Returning User Scenario	9
6	Recording Controller for New User Scenario	11
7	CSV Credential File for New User Scenario	12
8	HTTP Post Request for New User Scenario	12
9	View Results Tree for New User Scenario	13
10	Apache Access Log Snapshot for New User Scenario	14
11	Windows Performance Monitor Snapshot for New User Scenario	15
12	JConsole Snapshot for New User Scenario	15

1 BORG Calendar

- The data flow analysis you performed and the calculation of the coverage metrics. You must show which test cases are responsible for which dc-paths.
- A description of the test cases you added to improve coverage. If your coverage was already high, discuss how your testing was able to achieve this.
- The slices that you identified and the percentage of slices that your testing covers. You must show which test cases are responsible for which slices.
- A description of the test cases you added to improve slice coverage. If your coverage was already high, discuss how your testing was able to achieve this.
- Evaluate the effectiveness of your test cases using mutation testing. Discuss and address any issues if you have found in your written report.
- Attaching bug reports if bugs are discovered using your testing methods. You should use the same bug report format as in Assignment 1. Do not file these bug reports to the projects bug report system.
- An appendix with the specification of the methods you are testing

2 JPetStore

After exploring the JPetStore system, we came up with some realistic non-trivial test scenarios that can be carried out for load testing using JMeter. The following subsections cover each scenarios, description on how it was load tested and the result analysis of the load test.

Following are the system specifications for which the load test was conducted under:

- **Operating System:** Windows 10 Pro 64-bit (10.0, Build 16299)
- **Language:** English (Regional Setting: English)
- **System Manufacturer:** Hewlett-Packard
- **System Model:** HP 15 TouchSmart Notebook PC
- **BIOS:** F.10
- **Processor:** AMD A6-5200 APU with Radeon(TM) HD Graphics (4 CPUs), 2.0GHz
- **Memory:** 6144MB RAM
- **Java Version:** 1.8.0_151-b12
- **Apache Tomcat Version:** 7.0.85
- **JMeter Version:** 2.11 r1554548

2.1 Scenario 1: Returning User

2.1.1 Overview

The first test scenario is for an existing user in the system. The testing scenario will consist of a returning user logging in, selecting one of each of the 5 possible items sold in JPetStore, adding the items to the cart, performing a checkout of the cart and finally logging out. The following describes an exact breakdown of the steps the load test will carry out:

- Access the JPetStore Homepage (<http://localhost:8080/jpetstore/>).
- Click *Enter the Store*.
- Click on the *Sign in* button.
- Enter sign-in credentials and click *Login* By default, we will be using *j2ee* user for this scenario.
- Go to the *Fish* section, select a fish item, add it to the cart and return to the main menu.
- Go to the *Dogs* section, select a dog item, add it to the cart and return to the main menu.
- Go to the *Reptiles* section, select a reptile item, add it to the cart and return to the main menu.
- Go to the *Cats* section, select a cat item, add it to the cart and return to the main menu.
- Go to the *Birds* section, select a bird item, add it to the cart.
- Proceed to checkout and follow the steps until the order has been placed.
- Return back to the main menu and sign out.

The following images depicts the Recording Controller for the test case scenario and the pages our load test will navigate through per each iteration:

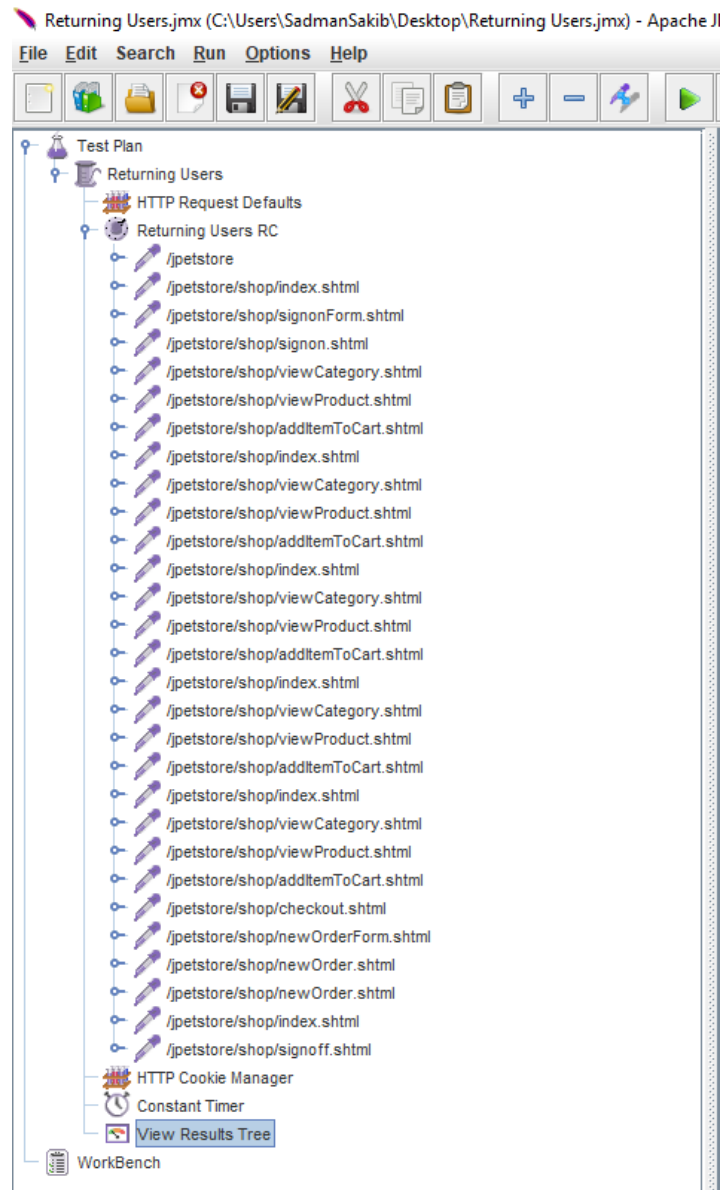


Figure 1: Recording Controller for Returning User Scenario

2.1.2 Load Test Properties

Following are the load test properties applied for testing this scenario:

- Number of Thread (users): 5
- Ramp-up Period (in seconds): 10
- Loop Count: 30
- Thread Delay (in milliseconds): 1000

2.1.3 Executing Load Test

After setting up the load test plan using JMeter, we executed the test. The test run was for approximately 18 minutes completing all 30 iterations. The following diagrams show the result tree of the test run.

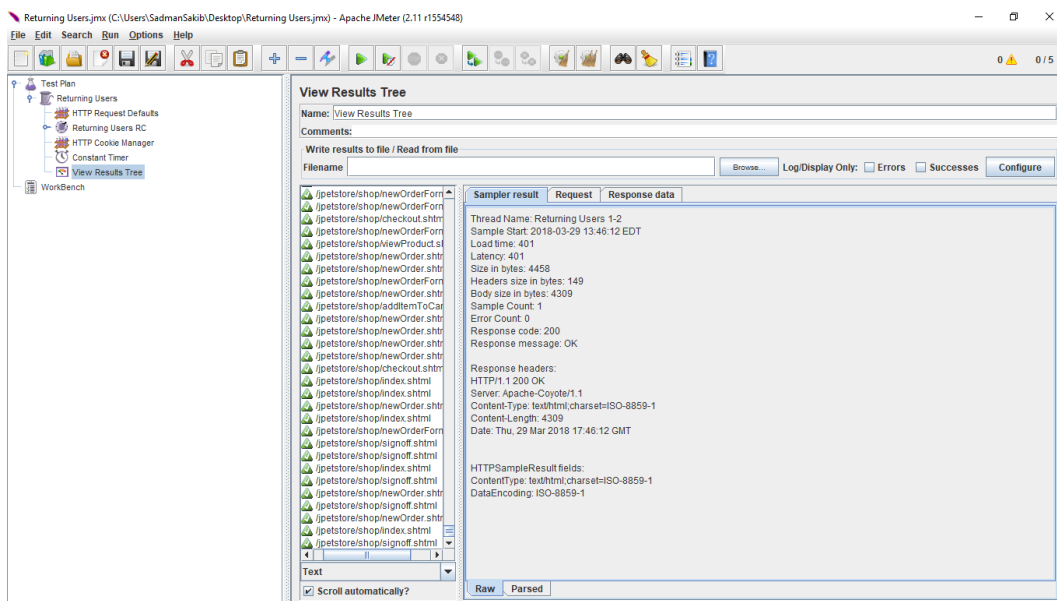


Figure 2: View Results Tree for Returning User Scenario

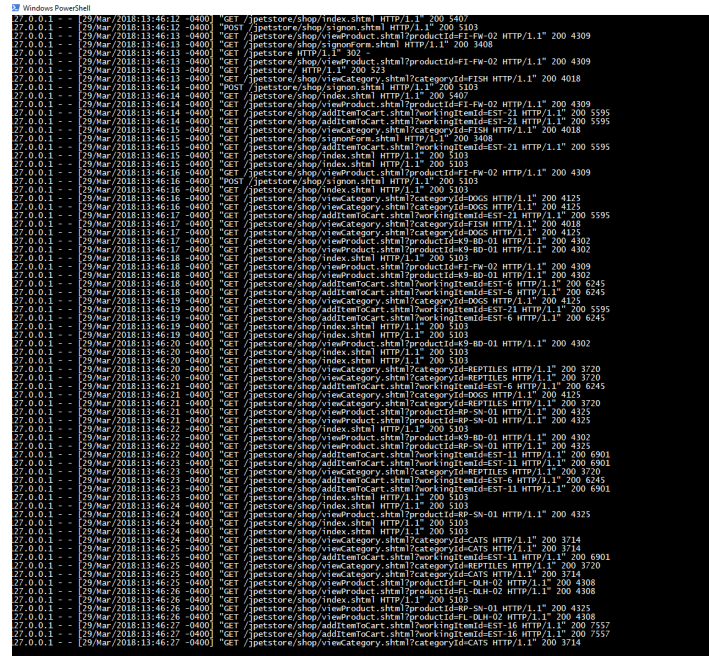
2.1.4 Analysing Load Test

The following statistics were gathered from the Apache access logs. The access log file was parsed to get the start of the load test and the end of the load test and the total time of the load test execution.

Within that timeframe, the total number of requests was found by simply executing: *wc -l ruser_log.txt*. The total number of GET and POST requests were found by executing *grep -c -w "GET" ruser_log.txt* and *grep -c -w "POST" ruser_log.txt* respectively.

- **Test Duration:** Approximately 18 minutes
- **Total Number of Requests:** 4714
- **Number of GET Requests:** 4399 (93% of all requests)
- **Number of POST Requests:** 315 (7% of all requests)
- **Success HTTP Codes:** 200 (Success) and 302 (Found)
- **Failure HTTP Codes:** None

Following is a snapshot of the Apache access log:



```

22.0.0.1 - [29/Mar/2018:13:46:12 -0400] GET /petstore/shop/index.shtml HTTP/1.1" 200 540/
22.0.0.1 - [29/Mar/2018:13:46:12 -0400] POST /petstore/shop/signon.shtml HTTP/1.1" 200 5103/
22.0.0.1 - [29/Mar/2018:13:46:13 -0400] GET /petstore/shop/viewProduct.shtml?productId=FW-02 HTTP/1.1" 200 4309/
22.0.0.1 - [29/Mar/2018:13:46:13 -0400] GET /petstore/shop/signonForm.shtml HTTP/1.1" 200 3408/
22.0.0.1 - [29/Mar/2018:13:46:13 -0400] GET /petstore HTTP/1.1" 200 323/
22.0.0.1 - [29/Mar/2018:13:46:13 -0400] GET /petstore/shop/viewProduct.shtml?productId=FW-02 HTTP/1.1" 200 4309/
22.0.0.1 - [29/Mar/2018:13:46:13 -0400] GET /petstore/shop/viewCategory.shtml?categoryId=FW-02 HTTP/1.1" 200 4018/
22.0.0.1 - [29/Mar/2018:13:46:14 -0400] POST /petstore/shop/signon.shtml HTTP/1.1" 200 5103/
22.0.0.1 - [29/Mar/2018:13:46:14 -0400] GET /petstore/shop/index.shtml HTTP/1.1" 200 540/
22.0.0.1 - [29/Mar/2018:13:46:14 -0400] GET /petstore/shop/viewProduct.shtml?productId=FW-02 HTTP/1.1" 200 4309/
22.0.0.1 - [29/Mar/2018:13:46:14 -0400] GET /petstore/shop/addItemToCart.shtml?workingItemId=EST-21 HTTP/1.1" 200 5595/
22.0.0.1 - [29/Mar/2018:13:46:14 -0400] GET /petstore/shop/addItemToCart.shtml?workingItemId=EST-21 HTTP/1.1" 200 5595/
22.0.0.1 - [29/Mar/2018:13:46:15 -0400] GET /petstore/shop/viewCategory.shtml?categoryId=FW-02 HTTP/1.1" 200 4018/
22.0.0.1 - [29/Mar/2018:13:46:15 -0400] GET /petstore/shop/signonForm.shtml HTTP/1.1" 200 3408/
22.0.0.1 - [29/Mar/2018:13:46:15 -0400] GET /petstore/shop/addItemToCart.shtml?workingItemId=EST-21 HTTP/1.1" 200 5595/
22.0.0.1 - [29/Mar/2018:13:46:15 -0400] GET /petstore/shop/index.shtml HTTP/1.1" 200 5103/
22.0.0.1 - [29/Mar/2018:13:46:15 -0400] GET /petstore/shop/index.shtml HTTP/1.1" 200 5103/
22.0.0.1 - [29/Mar/2018:13:46:16 -0400] GET /petstore/shop/viewProduct.shtml?productId=FW-02 HTTP/1.1" 200 4309/
22.0.0.1 - [29/Mar/2018:13:46:16 -0400] POST /petstore/shop/signon.shtml HTTP/1.1" 200 5103/
22.0.0.1 - [29/Mar/2018:13:46:16 -0400] GET /petstore/shop/index.shtml HTTP/1.1" 200 5103/
22.0.0.1 - [29/Mar/2018:13:46:16 -0400] GET /petstore/shop/viewCategory.shtml?categoryId=0005 HTTP/1.1" 200 4125/
22.0.0.1 - [29/Mar/2018:13:46:16 -0400] GET /petstore/shop/viewCategory.shtml?categoryId=0005 HTTP/1.1" 200 4125/
22.0.0.1 - [29/Mar/2018:13:46:16 -0400] GET /petstore/shop/viewCategory.shtml?categoryId=FW-02 HTTP/1.1" 200 5595/
22.0.0.1 - [29/Mar/2018:13:46:17 -0400] GET /petstore/shop/viewCategory.shtml?categoryId=FW-02 HTTP/1.1" 200 4018/
22.0.0.1 - [29/Mar/2018:13:46:17 -0400] GET /petstore/shop/viewCategory.shtml?categoryId=0005 HTTP/1.1" 200 4125/
22.0.0.1 - [29/Mar/2018:13:46:17 -0400] GET /petstore/shop/viewProduct.shtml?productId=RS-01 HTTP/1.1" 200 4302/
22.0.0.1 - [29/Mar/2018:13:46:17 -0400] GET /petstore/shop/index.shtml HTTP/1.1" 200 5103/
22.0.0.1 - [29/Mar/2018:13:46:18 -0400] GET /petstore/shop/viewProduct.shtml?productId=FW-02 HTTP/1.1" 200 4309/
22.0.0.1 - [29/Mar/2018:13:46:18 -0400] GET /petstore/shop/viewProduct.shtml?productId=RS-01 HTTP/1.1" 200 4309/
22.0.0.1 - [29/Mar/2018:13:46:18 -0400] GET /petstore/shop/addItemToCart.shtml?workingItemId=EST-6 HTTP/1.1" 200 6245/
22.0.0.1 - [29/Mar/2018:13:46:18 -0400] GET /petstore/shop/addItemToCart.shtml?workingItemId=EST-6 HTTP/1.1" 200 6245/
22.0.0.1 - [29/Mar/2018:13:46:19 -0400] GET /petstore/shop/viewCategory.shtml?categoryId=0005 HTTP/1.1" 200 4125/
22.0.0.1 - [29/Mar/2018:13:46:19 -0400] GET /petstore/shop/addItemToCart.shtml?workingItemId=EST-21 HTTP/1.1" 200 5595/
22.0.0.1 - [29/Mar/2018:13:46:19 -0400] GET /petstore/shop/addItemToCart.shtml?workingItemId=EST-6 HTTP/1.1" 200 6245/
22.0.0.1 - [29/Mar/2018:13:46:19 -0400] GET /petstore/shop/index.shtml HTTP/1.1" 200 5103/
22.0.0.1 - [29/Mar/2018:13:46:19 -0400] GET /petstore/shop/index.shtml HTTP/1.1" 200 5103/
22.0.0.1 - [29/Mar/2018:13:46:20 -0400] GET /petstore/shop/index.shtml HTTP/1.1" 200 5103/
22.0.0.1 - [29/Mar/2018:13:46:20 -0400] GET /petstore/shop/index.shtml HTTP/1.1" 200 5103/
22.0.0.1 - [29/Mar/2018:13:46:20 -0400] GET /petstore/shop/index.shtml HTTP/1.1" 200 5103/
22.0.0.1 - [29/Mar/2018:13:46:20 -0400] GET /petstore/shop/viewCategory.shtml?categoryId=REPTILES HTTP/1.1" 200 3720/
22.0.0.1 - [29/Mar/2018:13:46:20 -0400] GET /petstore/shop/viewCategory.shtml?categoryId=REPTILES HTTP/1.1" 200 3720/
22.0.0.1 - [29/Mar/2018:13:46:20 -0400] GET /petstore/shop/addItemToCart.shtml?workingItemId=EST-6 HTTP/1.1" 200 6245/
22.0.0.1 - [29/Mar/2018:13:46:21 -0400] GET /petstore/shop/viewCategory.shtml?categoryId=0005 HTTP/1.1" 200 4125/
22.0.0.1 - [29/Mar/2018:13:46:21 -0400] GET /petstore/shop/viewCategory.shtml?categoryId=REPTILES HTTP/1.1" 200 3720/
22.0.0.1 - [29/Mar/2018:13:46:21 -0400] GET /petstore/shop/viewProduct.shtml?productId=BP-SN-01 HTTP/1.1" 200 4325/
22.0.0.1 - [29/Mar/2018:13:46:21 -0400] GET /petstore/shop/viewProduct.shtml?productId=BP-SN-01 HTTP/1.1" 200 4325/
22.0.0.1 - [29/Mar/2018:13:46:22 -0400] GET /petstore/shop/index.shtml HTTP/1.1" 200 5103/
22.0.0.1 - [29/Mar/2018:13:46:22 -0400] GET /petstore/shop/viewProduct.shtml?productId=BP-SN-01 HTTP/1.1" 200 4302/
22.0.0.1 - [29/Mar/2018:13:46:22 -0400] GET /petstore/shop/viewProduct.shtml?productId=BP-SN-01 HTTP/1.1" 200 4325/
22.0.0.1 - [29/Mar/2018:13:46:22 -0400] GET /petstore/shop/addItemToCart.shtml?workingItemId=EST-11 HTTP/1.1" 200 6901/
22.0.0.1 - [29/Mar/2018:13:46:23 -0400] GET /petstore/shop/viewCategory.shtml?categoryId=REPTILES HTTP/1.1" 200 6901/
22.0.0.1 - [29/Mar/2018:13:46:23 -0400] GET /petstore/shop/addItemToCart.shtml?workingItemId=EST-11 HTTP/1.1" 200 6901/
22.0.0.1 - [29/Mar/2018:13:46:23 -0400] GET /petstore/shop/addItemToCart.shtml?workingItemId=EST-6 HTTP/1.1" 200 6245/
22.0.0.1 - [29/Mar/2018:13:46:23 -0400] GET /petstore/shop/addItemToCart.shtml?workingItemId=EST-11 HTTP/1.1" 200 6901/
22.0.0.1 - [29/Mar/2018:13:46:24 -0400] GET /petstore/shop/index.shtml HTTP/1.1" 200 5103/
22.0.0.1 - [29/Mar/2018:13:46:24 -0400] GET /petstore/shop/index.shtml HTTP/1.1" 200 5103/
22.0.0.1 - [29/Mar/2018:13:46:24 -0400] GET /petstore/shop/index.shtml HTTP/1.1" 200 5103/
22.0.0.1 - [29/Mar/2018:13:46:24 -0400] GET /petstore/shop/viewCategory.shtml?categoryId=CATS HTTP/1.1" 200 3714/
22.0.0.1 - [29/Mar/2018:13:46:24 -0400] GET /petstore/shop/viewCategory.shtml?categoryId=CATS HTTP/1.1" 200 3714/
22.0.0.1 - [29/Mar/2018:13:46:25 -0400] GET /petstore/shop/addItemToCart.shtml?workingItemId=EST-11 HTTP/1.1" 200 6901/
22.0.0.1 - [29/Mar/2018:13:46:25 -0400] GET /petstore/shop/viewCategory.shtml?categoryId=REPTILES HTTP/1.1" 200 3720/
22.0.0.1 - [29/Mar/2018:13:46:25 -0400] GET /petstore/shop/viewCategory.shtml?categoryId=CATS HTTP/1.1" 200 3714/
22.0.0.1 - [29/Mar/2018:13:46:25 -0400] GET /petstore/shop/viewProduct.shtml?productId=DLN-02 HTTP/1.1" 200 4308/
22.0.0.1 - [29/Mar/2018:13:46:26 -0400] GET /petstore/shop/viewProduct.shtml?productId=DLN-02 HTTP/1.1" 200 4308/
22.0.0.1 - [29/Mar/2018:13:46:26 -0400] GET /petstore/shop/index.shtml HTTP/1.1" 200 5103/
22.0.0.1 - [29/Mar/2018:13:46:26 -0400] GET /petstore/shop/viewProduct.shtml?productId=BP-SN-01 HTTP/1.1" 200 4325/
22.0.0.1 - [29/Mar/2018:13:46:26 -0400] GET /petstore/shop/viewProduct.shtml?productId=FW-02 HTTP/1.1" 200 4308/
22.0.0.1 - [29/Mar/2018:13:46:27 -0400] GET /petstore/shop/addItemToCart.shtml?workingItemId=EST-16 HTTP/1.1" 200 7557/
22.0.0.1 - [29/Mar/2018:13:46:27 -0400] GET /petstore/shop/viewCategory.shtml?categoryId=CATS HTTP/1.1" 200 3714/

```

Figure 3: Apache Access Log Snapshot for Returning User Scenario

Following is a snapshot of the Windows Performance Monitor and Java Monitor Console during the load test execution:

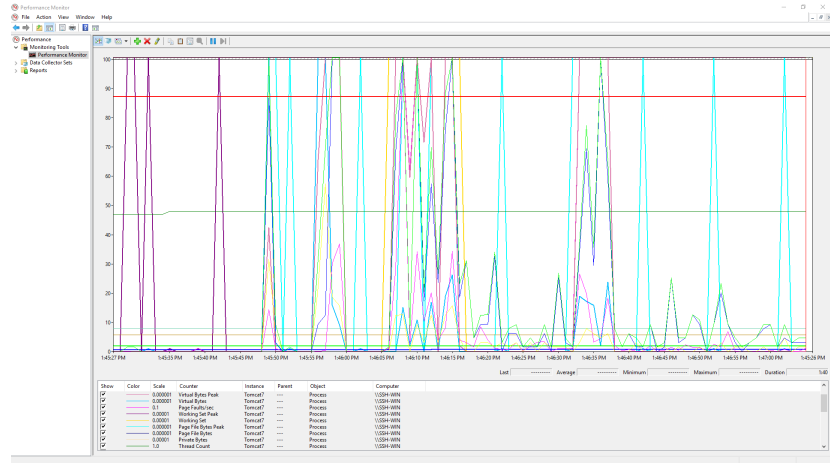


Figure 4: Windows Performance Monitor Snapshot for Returning User Scenario

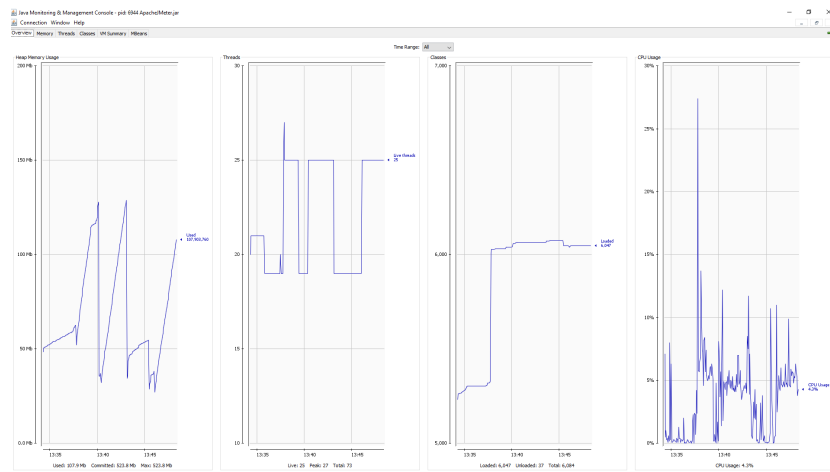


Figure 5: JConsole Snapshot for Returning User Scenario

Conclusion: The load test scenario for returning user made about 4714 requests and the test ran for approximately 18 minutes. Despite some natural high spikes on the performance monitor and java monitor, the load test was carried out successfully without any crashes or unexpected behaviour in the application.

2.2 Scenario 2: New User

2.2.1 Overview

The second test scenario is for a new user in the system. The testing scenario will consist of registering a new user, selecting one of each of the 5 possible items sold in JPetStore, adding the items to the cart, performing a checkout of the cart and finally logging out. The following describes an exact breakdown of the steps the load test will carry out:

- Access the JPetStore Homepage (<http://localhost:8080/jpetstore/>).
- Click *Enter the Store*.
- Click on the *Sign in* button.
- Click *Register now*.
- Enter the sign-up credentials and click *Create Account*. The username and password for signing up would be loaded from a CSV file, whereas the other fields will be supplied the value of *abc*.
- Go to the *Fish* section, select a fish item, add it to the cart and return to the main menu.
- Go to the *Dogs* section, select a dog item, add it to the cart and return to the main menu.
- Go to the *Reptiles* section, select a reptile item, add it to the cart and return to the main menu.
- Go to the *Cats* section, select a cat item, add it to the cart and return to the main menu.
- Go to the *Birds* section, select a bird item, add it to the cart.
- Proceed to checkout and follow the steps until the order has been placed.
- Return back to the main menu and sign out.

The following images depicts the Recording Controller for the test case scenario and the pages our load test will navigate through per each iteration:

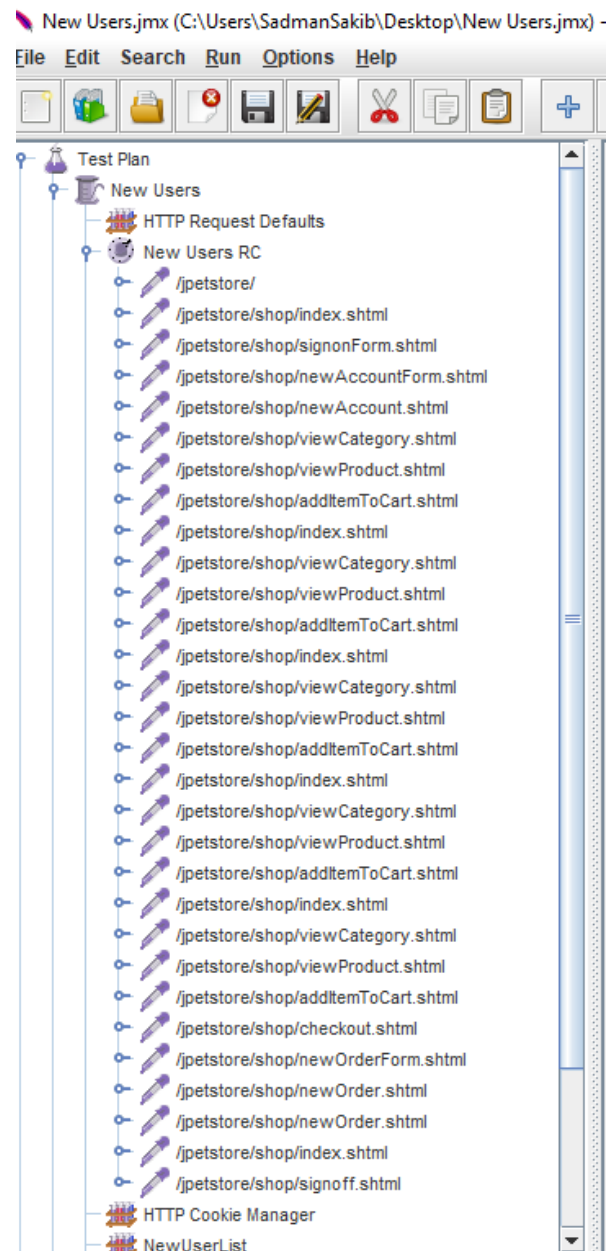


Figure 6: Recording Controller for New User Scenario

The following images depicts the snapshot of the CSV file containing usernames and password for registering new users. On the POST request for registering a new user we load the credentials and use them through variable names such as `${username}` and `${password}`.

1	newuser11	password
2	newuser12	password
3	newuser13	password
4	newuser14	password
5	newuser15	password
6	newuser16	password
7	newuser17	password
8	newuser18	password
9	newuser19	password
10		
11		
12		
13		
14		
15		
16		
17		
18		
19		
20		
21		
22		
23		
24		
25		
26		
27		
28		
29		
30		
31		
32		
33		
34		
35		
36		
37		
38		
39		
40		

Figure 7: CSV Credential File for New User Scenario

HTTP Request

Name:

Comments:

Web Server

Server Name or IP:
Port Number:

Timeouts (milliseconds)
Connect:
Response:

HTTP Request

Implementation:
Protocol (http):
Method:
Content encoding:

Path:

☐ Redirect Automatically
☒ Follow Redirects
☒ Use KeepAlive
☐ Use multipart/form-data for POST
☐ Browser-compatible headers

Parameters
Body Data

Send Parameters With the Request:

Name:	Value	Encode?	Include Equals?
username	<input type="text" value="\${username}"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
password	<input type="text" value="\${password}"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
repeatedPassword	<input type="text" value="\${password}"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Detail
Add
Add from Clipboard
Delete
Up
Down

Figure 8: HTTP Post Request for New User Scenario

2.2.2 Load Test Properties

Following are the load test properties applied for testing this scenario:

- Number of Thread (users): 5
- Ramp-up Period (in seconds): 5
- Loop Count: 30
- Thread Delay (in milliseconds): 1000

2.2.3 Executing Load Test

After setting up the load test plan using JMeter, we executed the test. The test run was for approximately 15 minutes completing all 30 iterations. The following diagrams show the result tree of the test run.

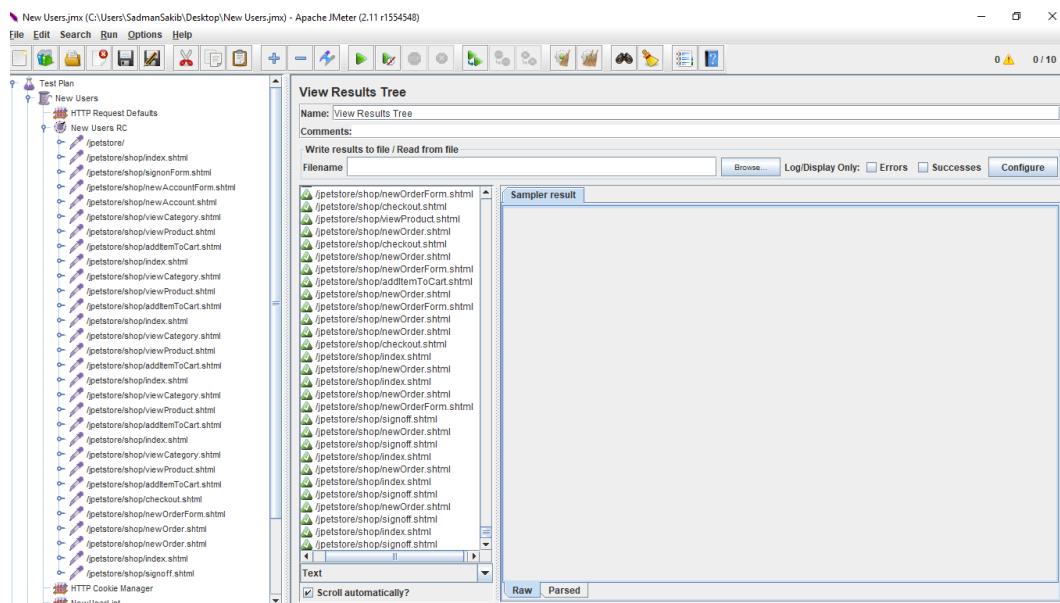


Figure 9: View Results Tree for New User Scenario

2.2.4 Analysing Load Test

The following statistics were gathered from the Apache access logs. The access log file was parsed to get the start of the load test and the end of the load test and the total time of the load test execution.

Following is a snapshot of the Windows Performance Monitor and Java Monitor Console during the load test execution:

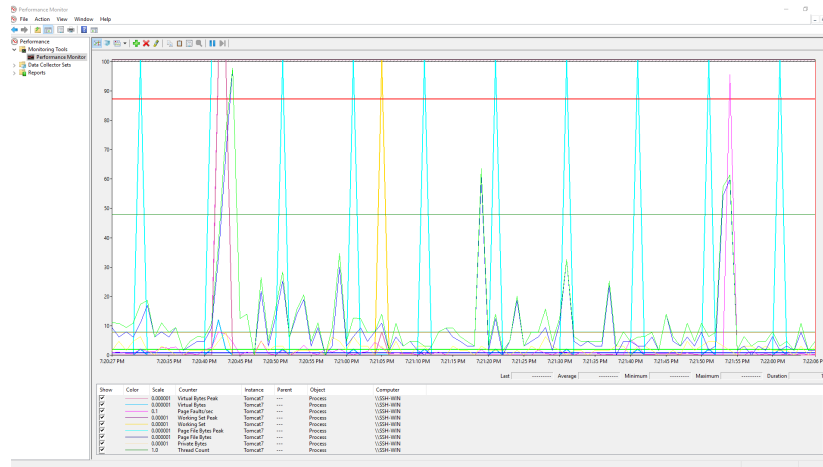


Figure 11: Windows Performance Monitor Snapshot for New User Scenario

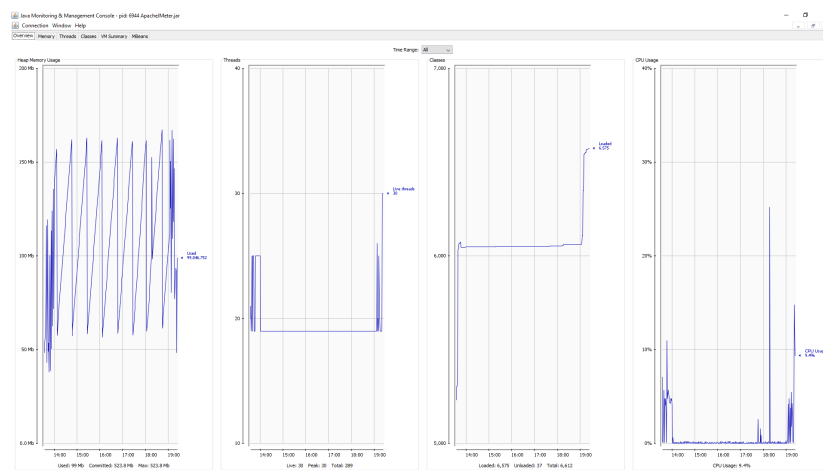


Figure 12: JConsole Snapshot for New User Scenario

Conclusion: The load test scenario for new user made about 7500 requests and the test ran for approximately 15 minutes. Despite some natural high spikes on the performance monitor and java monitor, the load test was carried out successfully without any crashes or unexpected behaviour in the application.