**TEST EXECUTION DOCUMENT**

**AGILE PRODUCT DEVELOPMENT SYSTEM**

Neemrana, 25th Nov, 2016

Software engineering project

NIIT University

Department of computer science

**Course**

CS-301 Software engineering project

**Project group**

Saurav Singh

Ronak Jain

Raghav Mehta

Shiv Bajpai

Siddhant Misra

**Customer**

FI-Street

**Project Masters**

Rahul channgani

Amit kumar

**Contents**

1. Introduction
2. Unit testing
3. Integration testing

1. System testing
2. Errors/Bugs
3. **Introduction**

This is the test documentation of the Agile product development system of software engineering project at the computer science department of the NIIT university, Neemrana.

Agile based PDS, an open software mainly used for the systematic development of the product by creating modules and sub-modules on which a team of people are assigned and helps in developing the product.

The purpose of this software engineering project is to provide a low-level description of the agile based product development system, providing insight into the structure and design of each component.

In short, this document will provide you with all the insights of our product with a proper understanding of the inner working of Agile-PDS and with the appropriate test cases involved in our project followed by bugs and its fixation.

-> Section 2 introduces the results of unit testing.

-> Section 3 introduces the results of integration testing.

-> Section 4 introduces the results of system testing.

-> In section 5 the bugs found during test phase are listed

All test cases were ﬁrst executed and then if we found some bugs, then they were ﬁxed at that present moment only. Then all test cases were executed again and no further bugs were found.

1. **Unit testing**

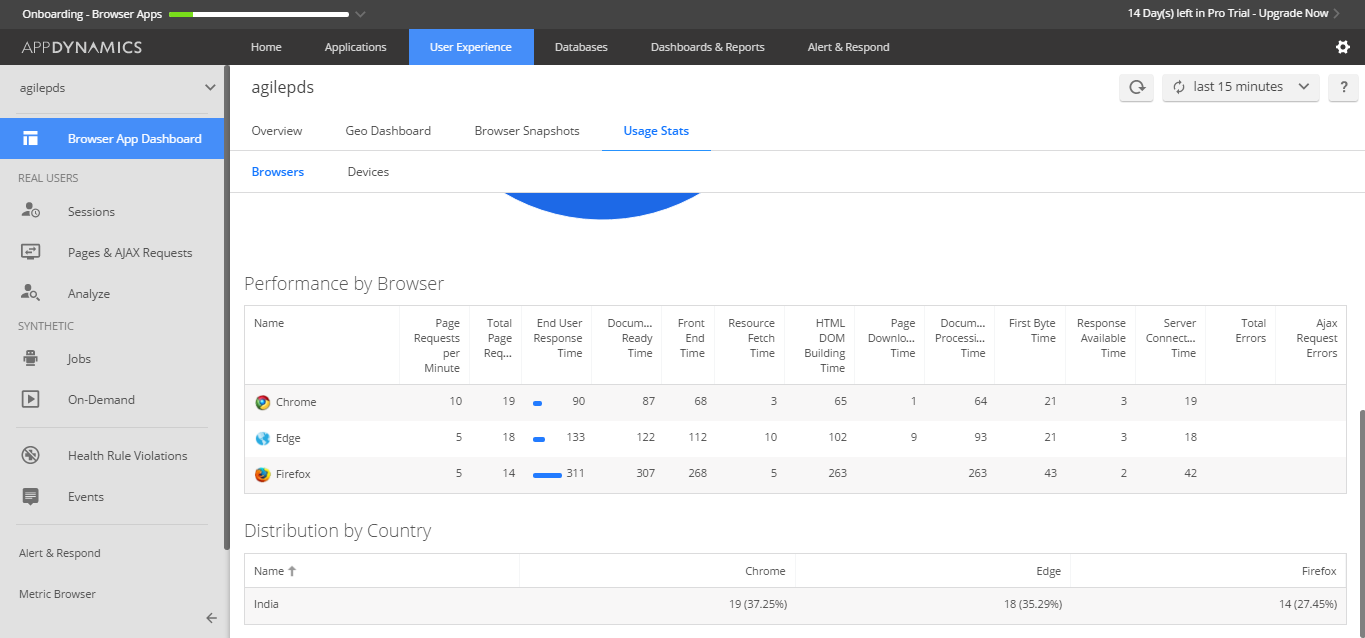
Unit tests were created as APPDYNAMICS tests. All the components were included in this run. Basically we added all the code semantics in our JAVA EE DEVELOPMENT TOOL with the addition of a script of appdynamics too so that it would be easier for all the HTML, CSS and BOOTSTRAP files.

These files were created and developed in ECLIPSE NEON.1 with the integration of MAVEN and all the libraries required for all the languages that we used in the form of dependencies.

When we finished with this integration and code completion, we then moved towards the testing part in which we just added the appdynamics script to all our component’s .JSP file so that when we run that particular code the testing could automatically be done on appdynamics server because this server shows the real time progress report of our software system and helps in the development process by removing the errors and bugs.

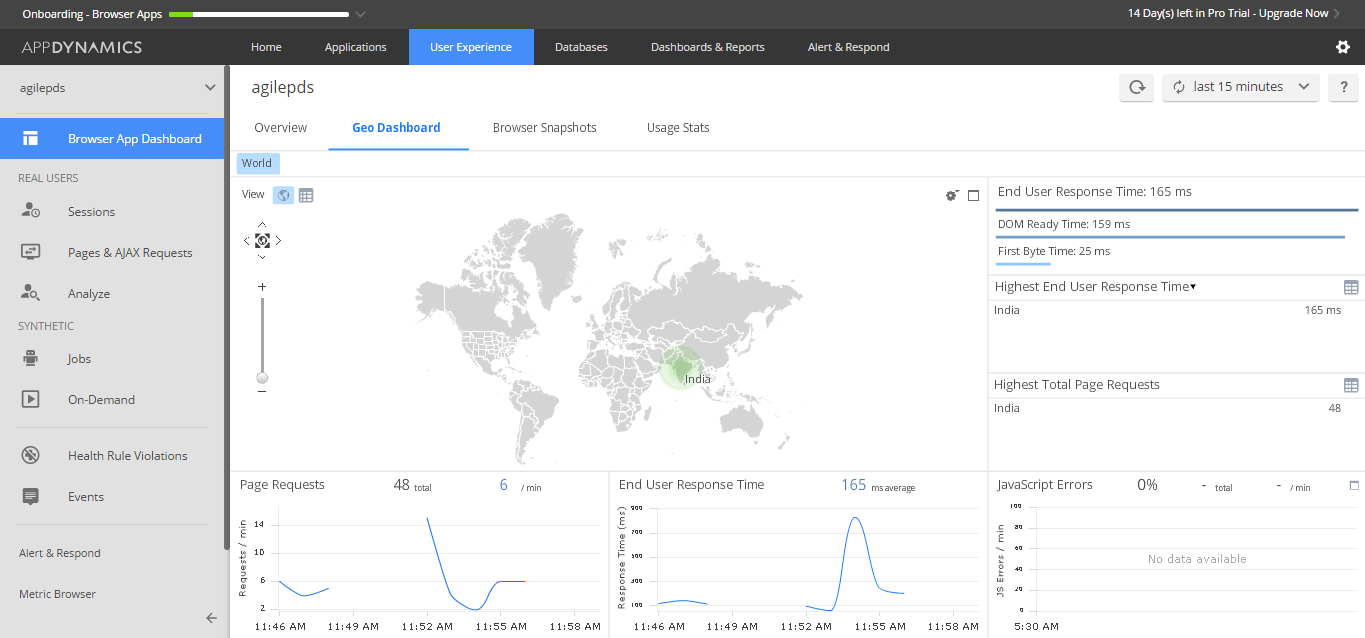
To execute this, we added a ardum.jsp file in our code which was linked with the script that we added in all the component’s .JSP file.

The appdynamics tests were run firsts and it was observed that when we executed in three different browsers it showed no errors or bugs.



1. **Integration testing**

In this section, the execution of HTML, CSS files of every components are shown with the statistics of errors and user response time with respect to every browser on which that software is made run through.

****

**Test case 1 – Login.jsp**

1. DESCRIPTION :

This test was run to check whether Login page is working properly or not. (Input values login and password).

1. ERRORS FOUND :

No errors were found during the run.

3. OUTPUT:

For right values it directed to respective pages.

For wrong values it gave the desired result as “Invalid login Credentials”.

For no values Bootstrap return’s the respective error- as for email box it accepts only the email format. An exception is generated according to it.

**Test case 2 – Validation.jsp**

1. DESCRIPTION :

This test was run to check the connectivity with the server and the java-script embedded inside the validation.jsp

1. ERRORS FOUND:

No errors were found.

3. OUTPUT

It was successfully querying the queries to and fro from the server.

**Test case 3 – Manager.jsp**

1. DESCRIPTION :

This test was run to check the UI of manager is working successfully or not.

1. ERRORS FOUND:

No errors were found.

3. OUTPUT

It was successfully working as expected.It showed all the expected options.

**Test case 4 – Employee.jsp**

1. DESCRIPTION :

This test was run to check the working of Employee.jsp.

1. ERRORS FOUND:

No errors were found.

3. OUTPUT

It was successfully working as expected.It showed all the expected options.

**Test case 5 – Customer.jsp**

1. DESCRIPTION :

This test was run to check the working of Customer.jsp

1. ERRORS FOUND:

No errors were found.

3. OUTPUT

It was successfully working as expected. It showed all the expected options.

**Test case 6 – Task Creation Manager.jsp**

1. DESCRIPTION :

This test was run to check the working of Task Creation function

In manager.jsp

1. ERRORS FOUND:

No errors were found.

3. OUTPUT

For right values it directed to respective pages.

For no values Bootstrap return’s the respective error- as for email box it accepts only the email format. An exception is generated according to it.

**Test case 7 – Assigning Employees to task Manager.jsp**

1. DESCRIPTION :

This test was run to check the working of Assigning Employees to task in manager.jsp

1. ERRORS FOUND:

No errors were found.

3. OUTPUT

For right values it directed to respective pages.

For no values Bootstrap return’s the respective error- as for email box it accepts only the email format. An exception is generated according to it.

**Test case 8 – Review form in Customer.jsp**

1. DESCRIPTION :

This test was run to check the working of Review form in Customer.jsp

1. ERRORS FOUND:

No errors were found.

3. OUTPUT

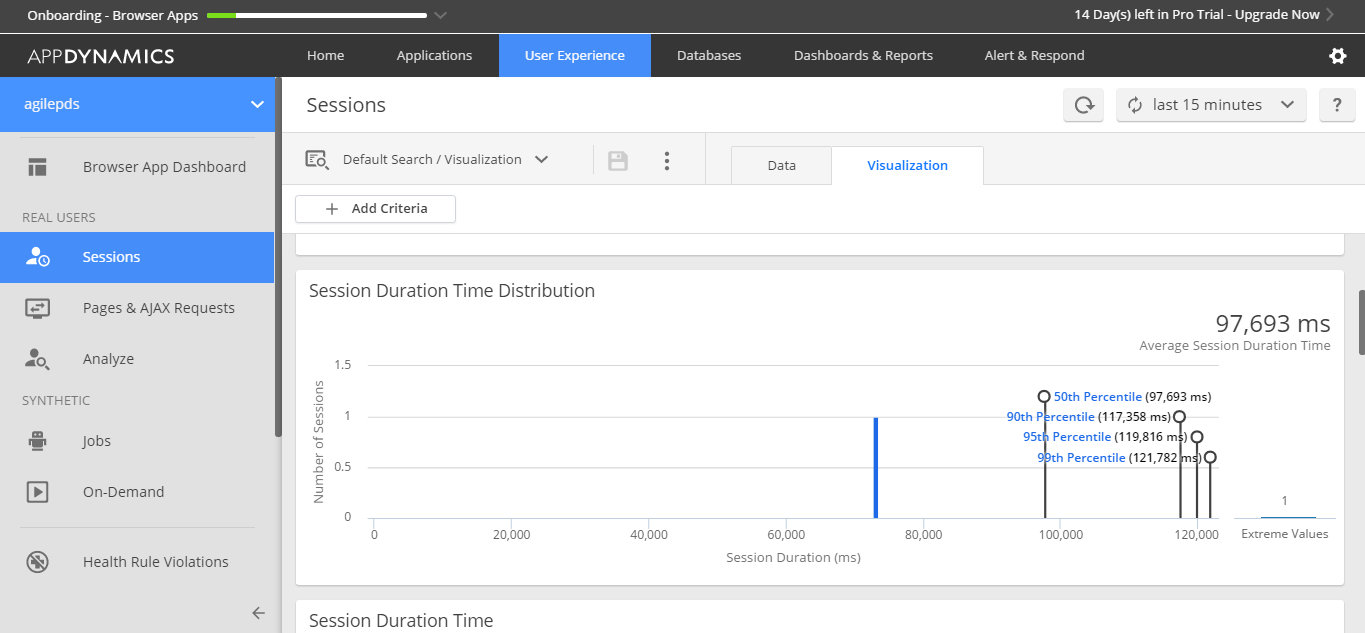
For right values it directed to respective pages.

For no values Bootstrap return’s the respective error- as for email box it accepts only the email format. An exception is generated according to it.

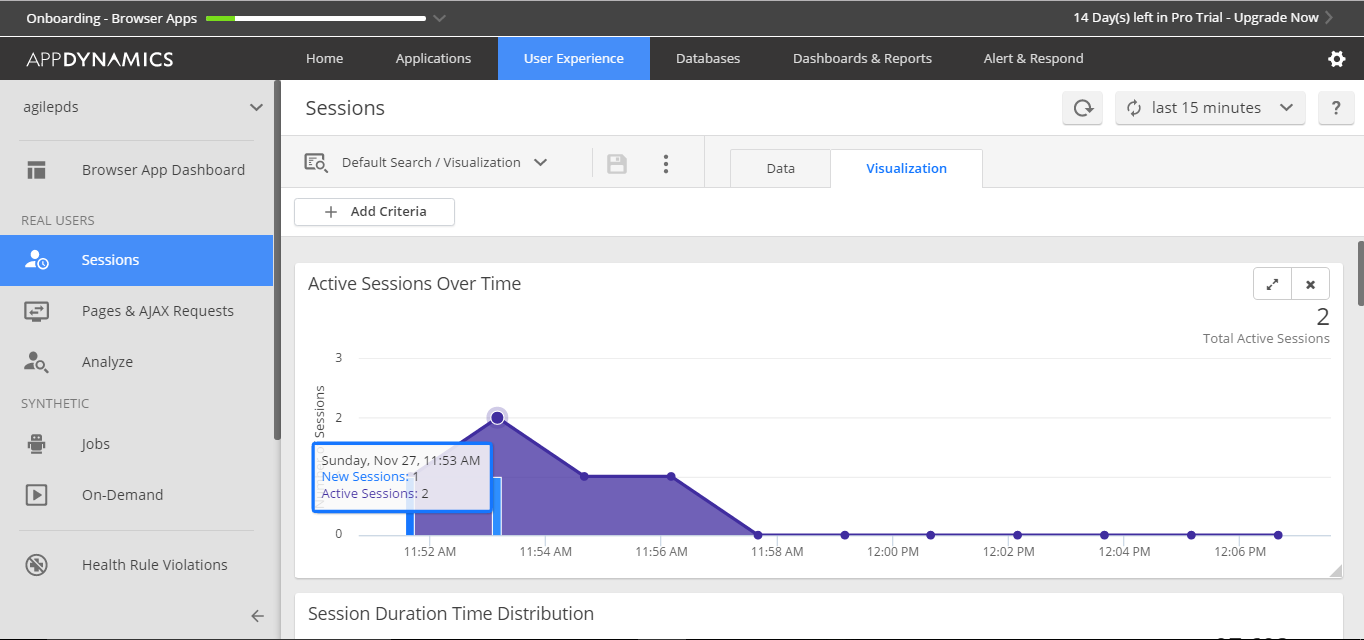
1. **SYSTEM TESTING**

Appdynamics server is the platform which checks all the test cases and tells us which error is there and what % of your software is full of errors or bugs.

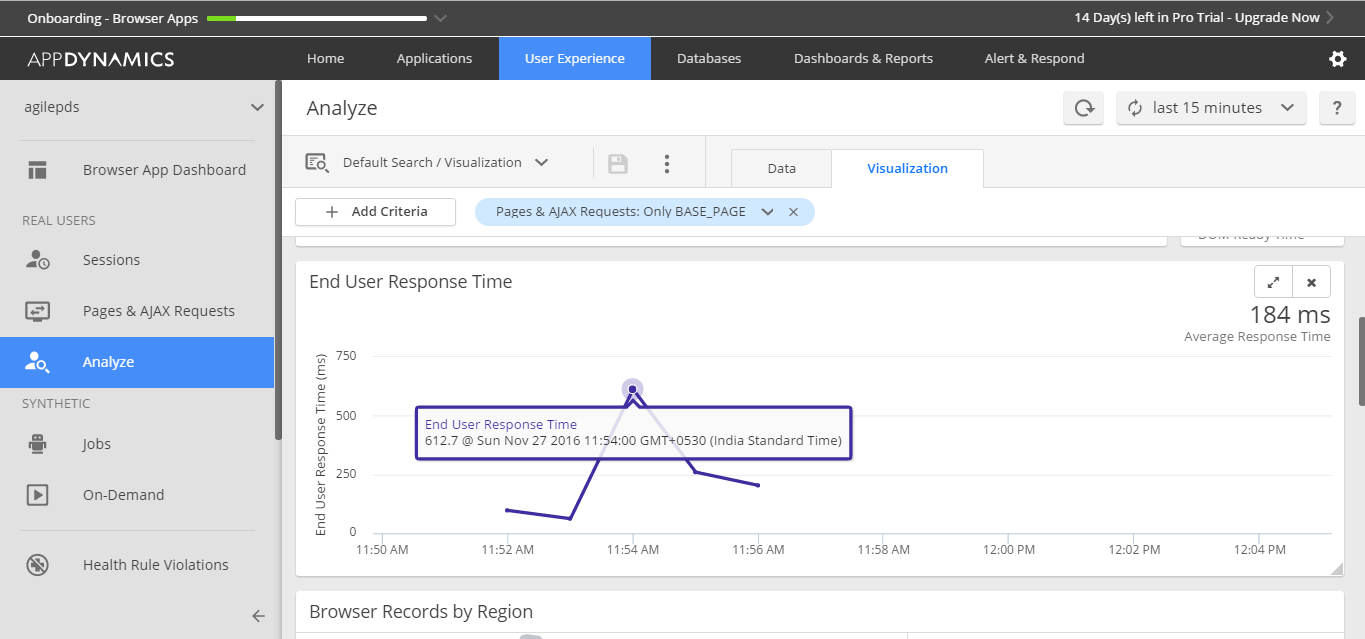
Appdynamics itself consists of all the test cases,

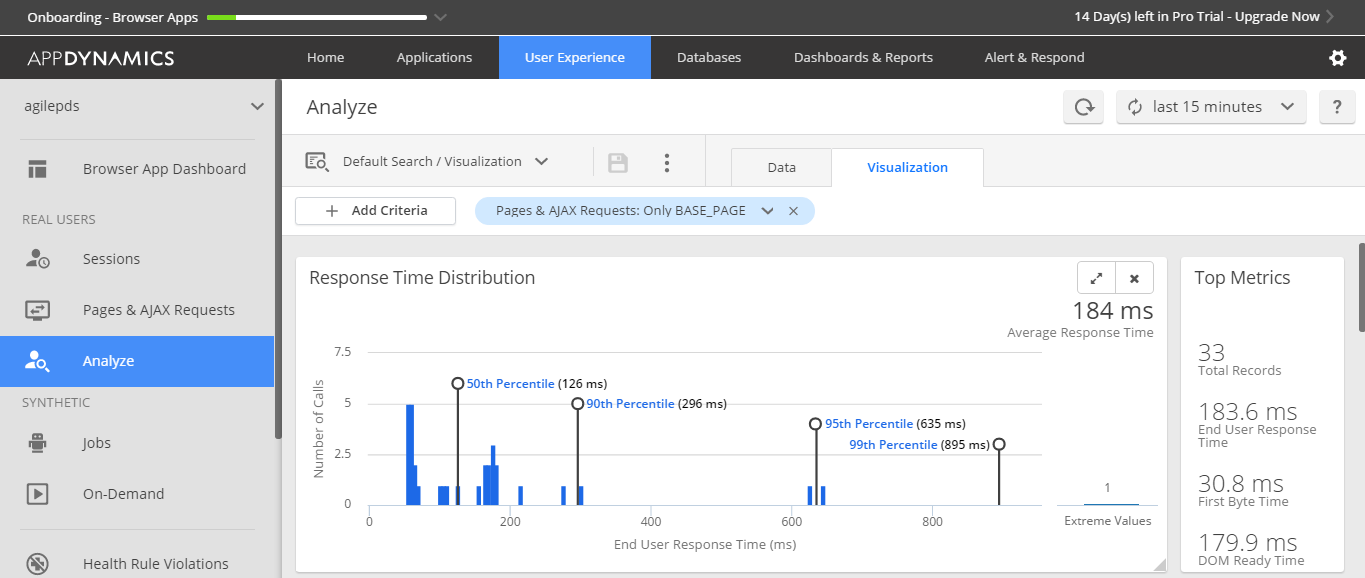


-> This image tells us about the average session duration of all the components testing done over Appdynamics.

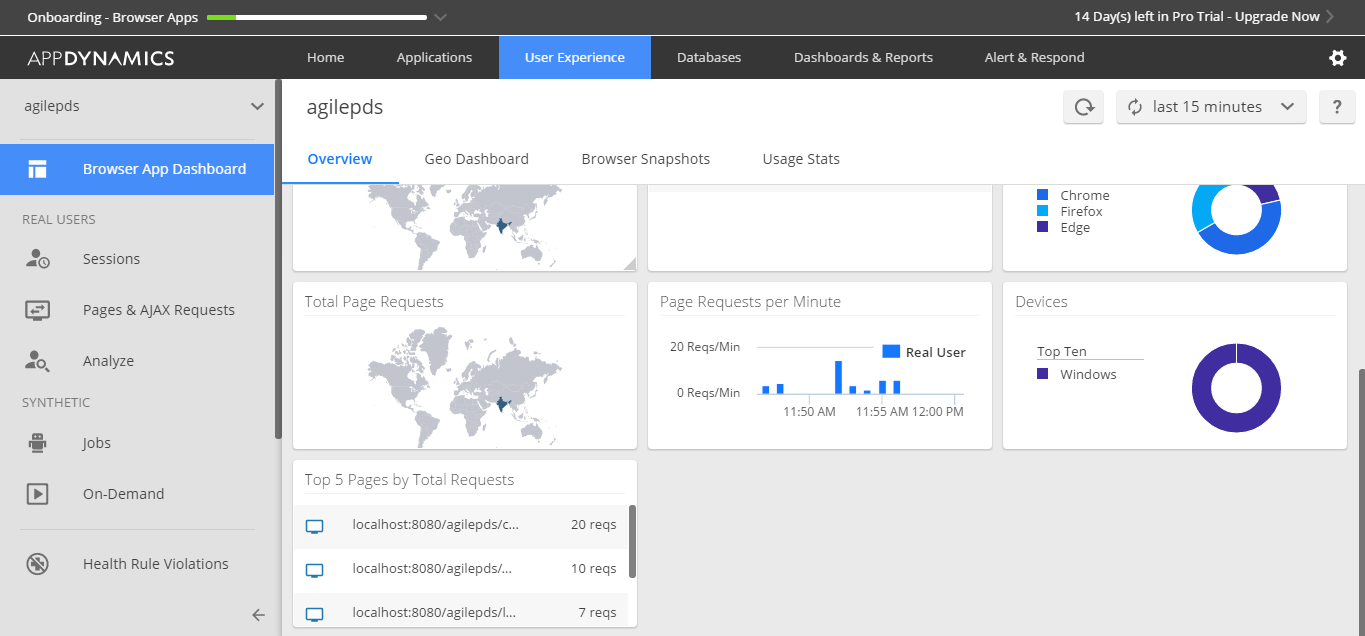


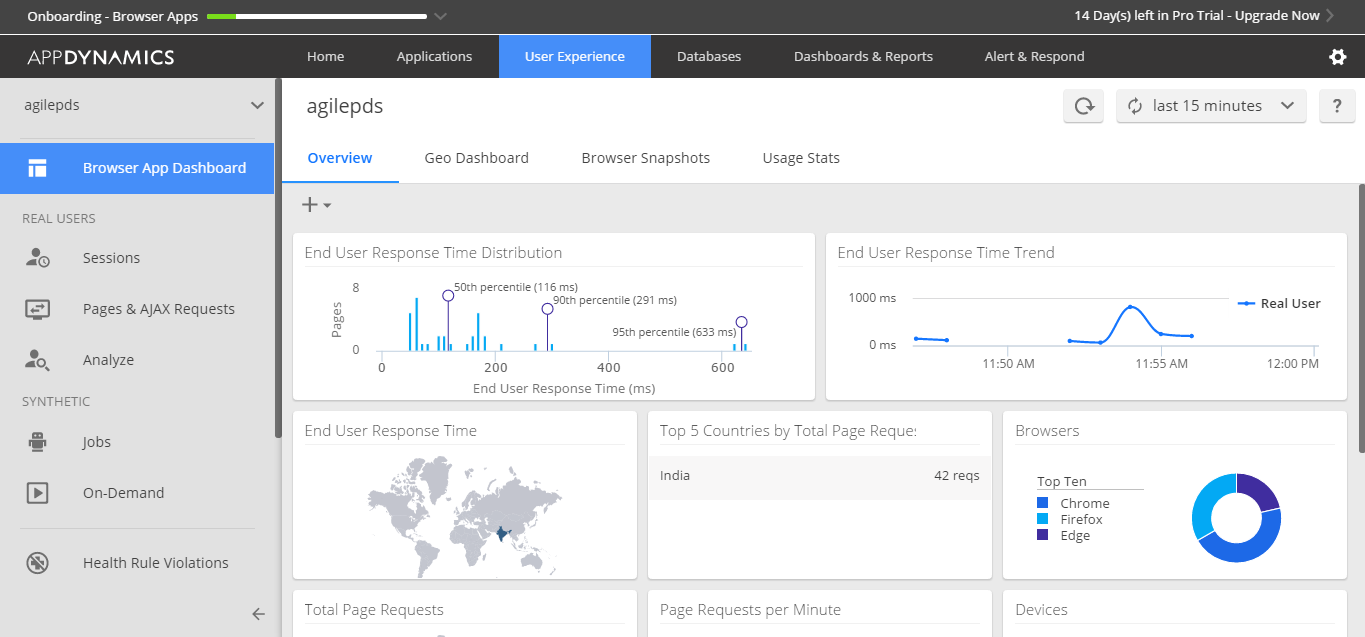
-> This graph tells us the real session time of a user on a component on our system.

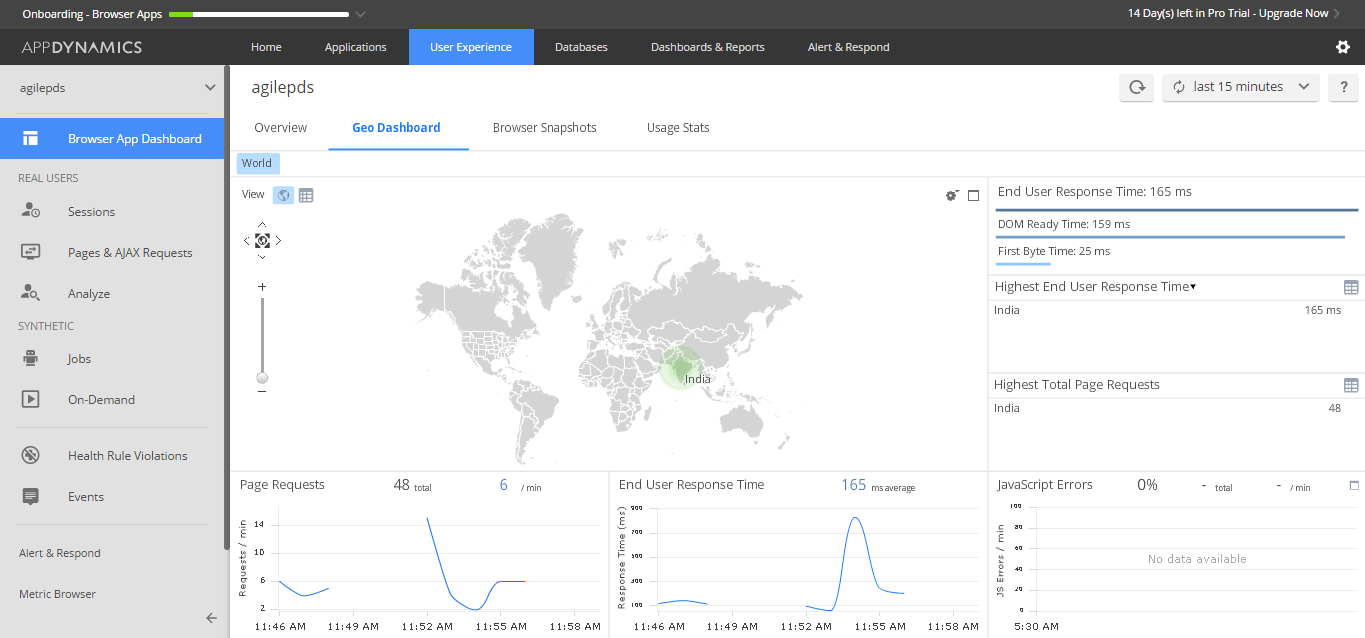




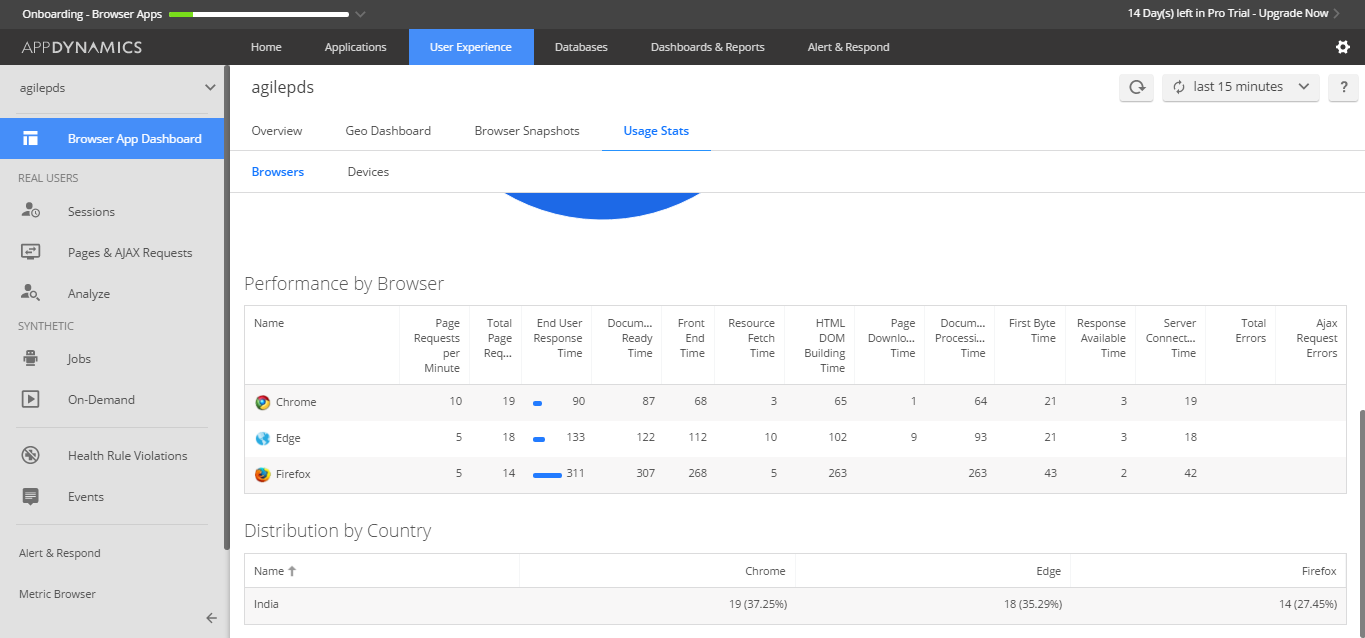
-> This image depicts about the queries getting executed during the test phase and telling us the time taken by each of them/cumulative report.

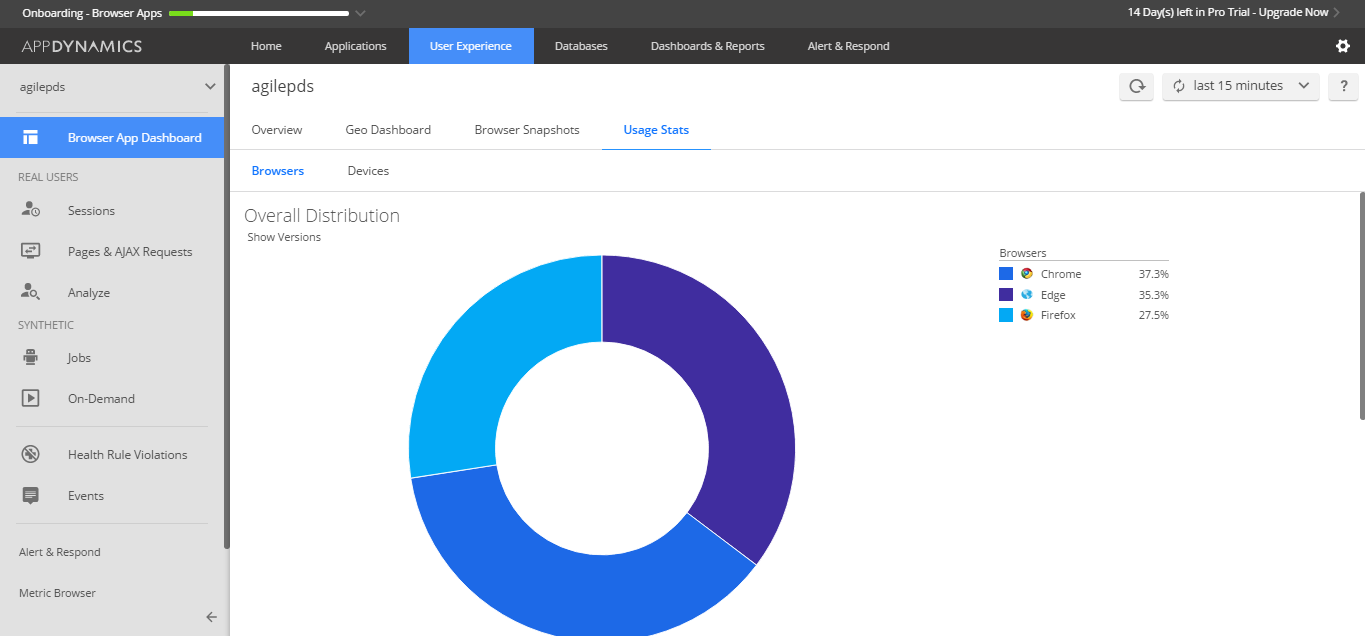






-> This image shows the geographic location showing which regions have the highest loads, the longest response times and the most errors etc. so here it is showing India.





-> This shows the overall report of the software being tested on different systems and giving us the report regarding the errors and active user response time and end user response time.

1. **Errors/Bugs :**

No errors were found.