**Title:**

201 – Test Automation with Java and Selenium-Internal assessment

**Revision History:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Reviewed By** | **Prepared By** | **Comments** |
| 16th Oct 2019 |  | Deepak Sharma |  |

**Project Overview:**

Selenium 201 course is a selenium hybrid framework, which is used to run multiple test case scripts and to capture all, passed and failed test cases of a particular website.

**Project Scope:**

Can be used as framework to run test cases and generate reports for any projects that has requirements of testing websites using selenium.

**Requirements:**

The framework satisfies the following components

1. Framework structure
2. Folder structure
3. Driver
4. Object repository
5. POM: for seamless addition of dependencies
6. XML file: for configuration of extent report and Testng suite.
7. Properties file: for getting the required properties in the framework
8. Test data – parameterization
9. Utilities
10. Configuration – Test, Global
11. Project Specific Scripts
12. Utilities
13. Results – TC level
14. Logs – Step level

**Technologies:**

* Selenium
* Core Java

**User guide for framework:**

1. **Packages**

|  |  |
| --- | --- |
| **Package name** | **Description** |
| **mindtree.yorbit.mercury.exceptions** | Used for handling different types of exceptions that may encounter during scripts run. |
| **mindtree.yorbit.mercury.util** | Utilities like Actions, ExcelReader, ExtentManager and TestConstants, which can be used across framework. |
| **mindtree.yorbit.mercury.pageobjects** | Contains the POM elements, which are selenium elements that are used frequently. |
| **mindtree.yorbit.mercury.testcases** | Contains test cases that has defined and executed. |
| **mindtree.yorbit.mercury.base** | Contains a base class which contains all the objects initialization and is extended to every test case. |

Table 1. Package names and Description

1. **Structure of a project:**

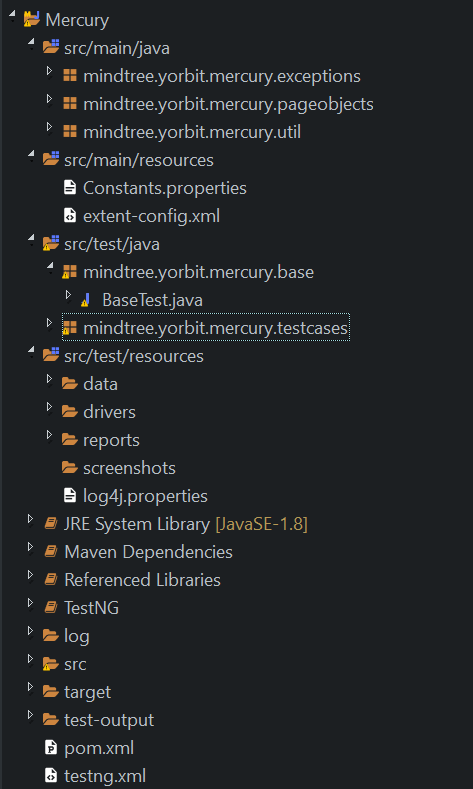
****

Fig. 1. Project Structure

1. **Flow Diagram of a framework:**

**testng.xml**

* Triggering of tests is done

using testing.xml file.

**Report Generation**

**Testcases**

* Internally testng.xml triggers execution of classes in test cases package.

Once tests are executed, reports are generated automatically.

Testcases:

1. All the tests are triggered.
2. Before methods and Before Tests are executed.
3. All the test steps will be executed.

Pages:

1. It contains all the web elements of different pages that are used to perform selenium operations.

Extentreport:

1. The execution steps and logs are logged.
2. Status of the executed test cases will be generated as report.
3. Every successful verification contains a screenshot.
4. **Execution results**

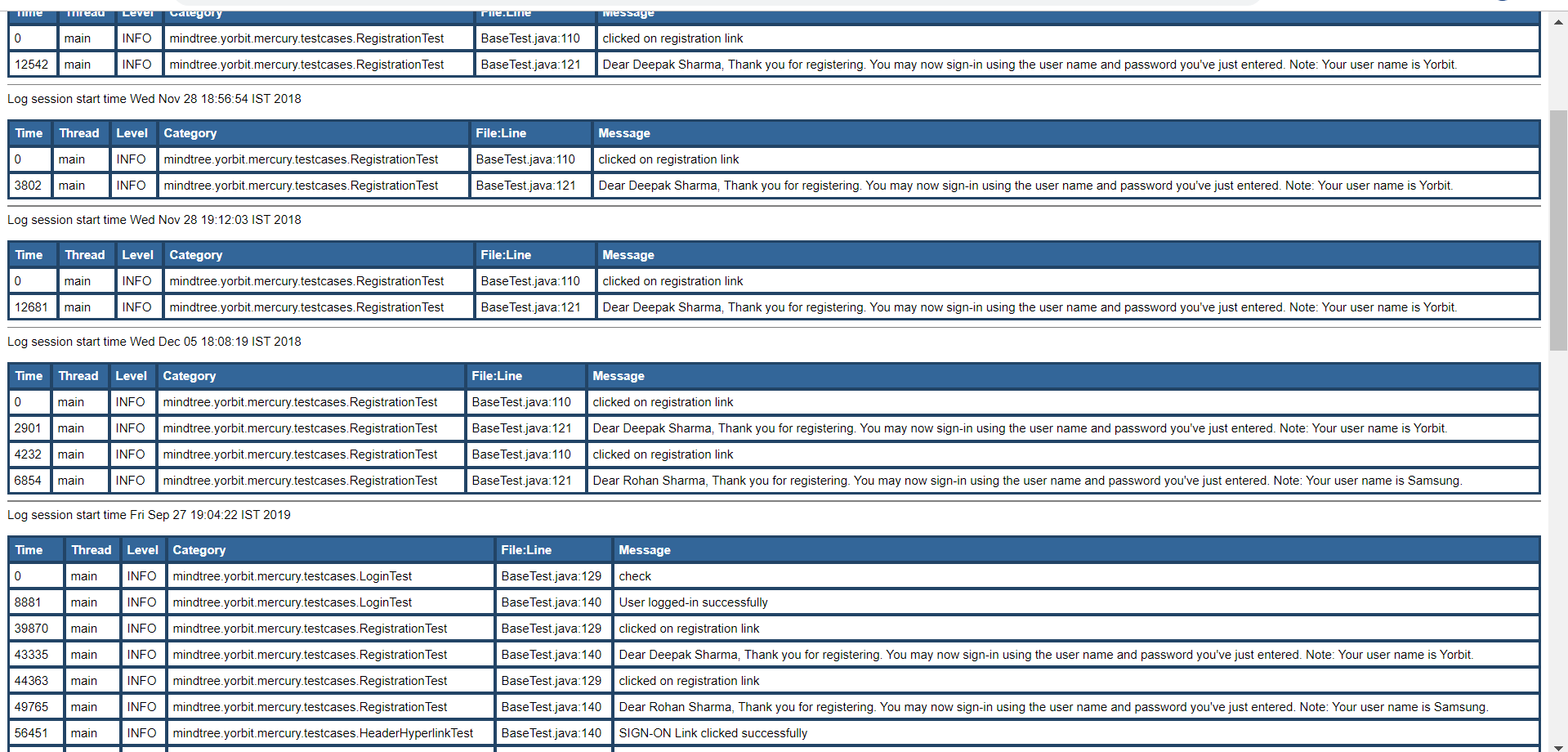
****

Figure 2. Execution Results:log4j report(console logs)

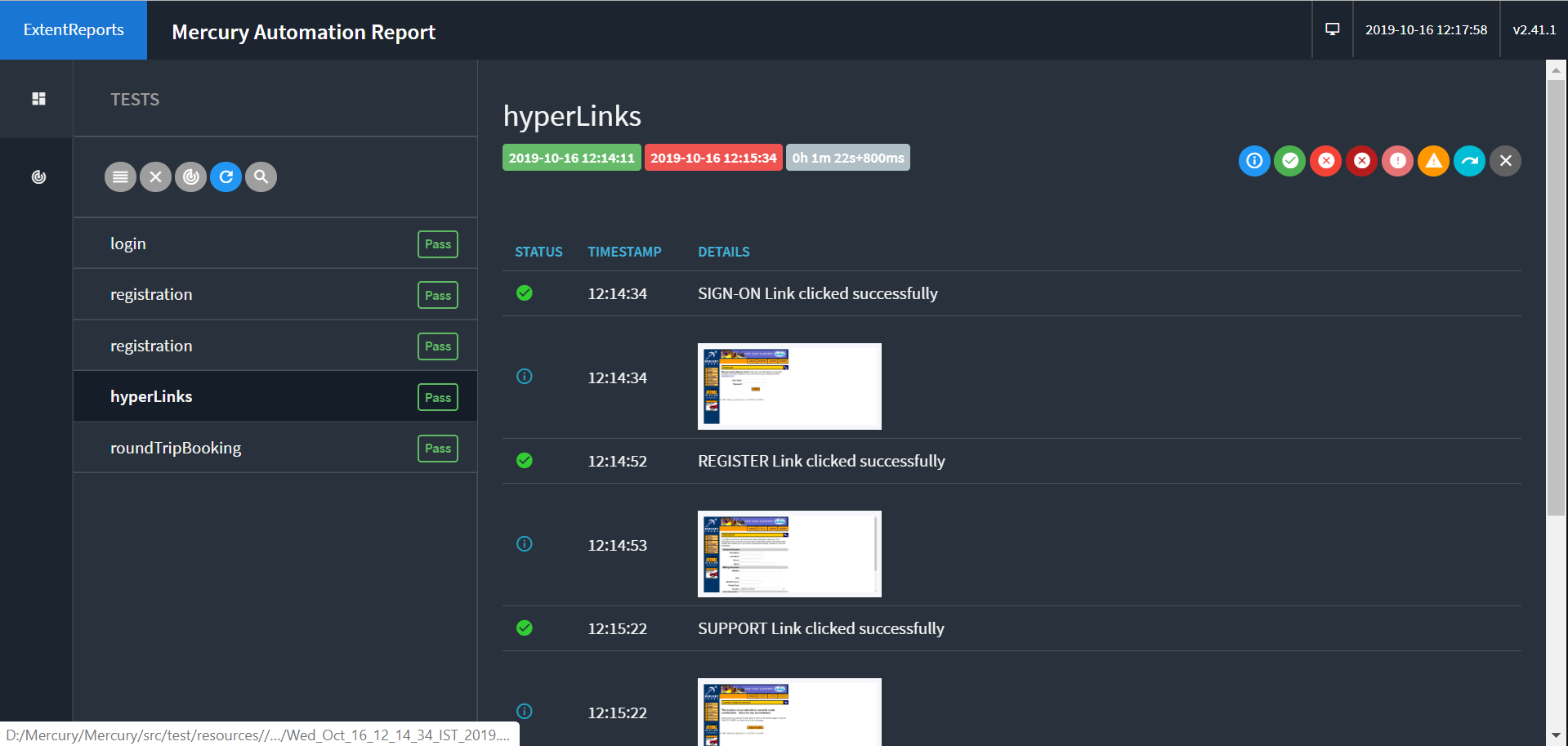
****

Figure 3. Execution Results: Extent report

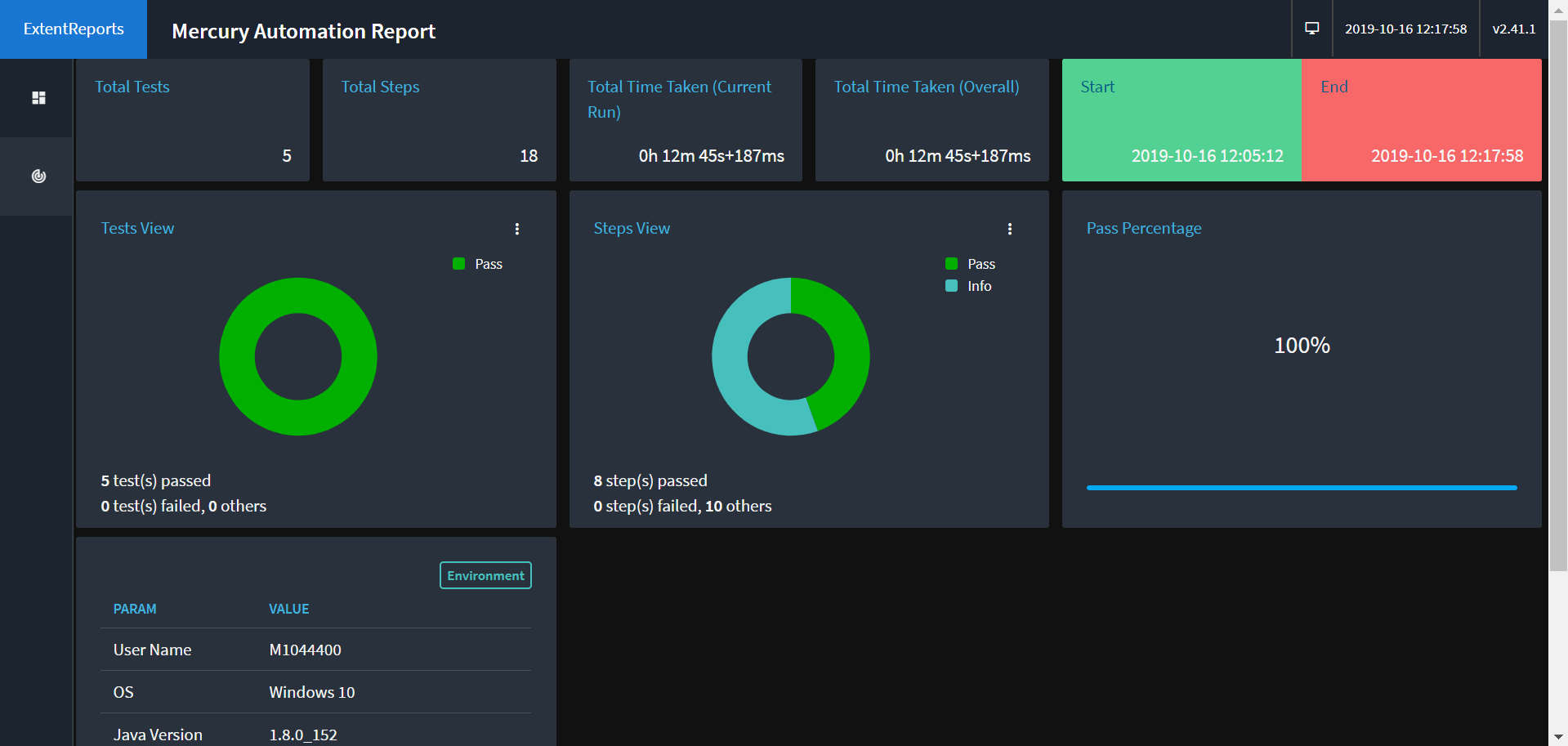
****

Figure 4. Dashboard View. Results Overview