

AUTOMATED SOFTWARE TESTING (502072)

Lab 2 – Robot Framework Prepared by Nguyen Thanh Quan (MSc)

1. GOALS: This lab helps students to

- Practise automation testing of web applications using.
- Be able to write test cases with Ride.
- Be able to run test cases.
- Understand various functionalities of Robot Framework.

2. OBJECTIVES

- Install automated testing environment with Selenium/Robot Framework.
- Understand Ride.
- Write and execute test cases.
- Understand keywords and data driven test cases.

3. CONTENT

3.1 PREREQUISITES

To practise complete this assignment, students must prepare to install the following libraries and packages. More importantly, students should have a basic understanding of testing concepts.

- Python
Go to python official site – <https://www.python.org/downloads/> and download the latest version or the prior version of python as per your operating system.
Remember to set PATH correctly to use Python after installation.
- Pip
PIP gets installed along with python. Run **`pip --version`** to check pip version
- Robot Framework
Use pip – python package manager to install the robot framework and the command for it is as follows

`pip install robotframework`
`robot --version` to check robot framework version.
- wxPython for Ride IDE
wxPython is needed for Robot Framework Ride, which is an IDE for Robot Framework.

Windows:

<https://sourceforge.net/projects/wxpython/files/wxPython/2.8.12.1/>

Linux: Install wxPython with the package manager of OS.

- Robot Framework Ride
Use pip command to install Ride IDE.
`pip install robotframework-ride`.
To open Ride IDE, run **`ride.py`**

3.2 INTRODUCTION TO ROBOT FRAMEWORK

Robot Framework is an open source test automation framework for acceptance testing and acceptance test-driven development. It follows different test case styles – keyword-driven, behaviour-driven and data-driven for writing test cases. This feature makes it very easy to understand.

Test cases are written using keyword style in a tabular format. Robot Framework provides good support for external libraries, tools that are open source and can be used for automation. The most popular library used with Robot Framework is Selenium Library (which will be introduced in next lab assignments) used for web development & UI testing.

Robot framework works fine on all the Operating Systems available. The framework is built on Python and runs on Jython (JVM) and IronPython (.NET).

Robot Framework Features

- **Tabular format for test cases:** Robot framework comes with a simple tabular format where the test cases are written using keywords. It is easy for a new developer to understand and write test cases.
- **Keywords:**
 - Robot framework comes with built-in keywords available with robot framework, keywords available from the libraries like Selenium Library (open browser, close browser, maximize browser, etc.).
 - We can also create user-defined keywords, which are a combination of other user-defined keywords or built-in or library keywords.
 - We can also pass arguments to those keywords, which make the user-defined keywords like functions that can be reused.
- **Variables:** Robot framework supports variables – scalar, list and dict. Variables in robot framework are easy to use and are of great help while writing complex test cases.
- **Libraries:** Robot framework has support for a lot of external libraries like SeleniumLibrary, Database Library, FTP Library and http library. SeleniumLibrary is mostly used as it helps to interact with the browsers and helps with web application and UI testing. Robot framework also has its own built-in libraries for strings, date, numbers etc.
- **Resources:** Robot framework also allows the import of robot files with keywords externally to be used with test cases. Resources are very easy to use and are of great help when we need to use some keywords already written for other test projects.
- **Data driven test cases:** Robot framework supports keyword driven style test

cases and data driven style. Data driven works with high-level keyword used as a template to the test suite and the test cases are used to share data with the high-level keyword defined in the template. It makes the work very easy for testing UI with different inputs.

- **Test Case Tagging:** Robot framework allows to tag test-cases so that we can either run the tags test-cases or skip the tagged testcases. Tagging helps when we want to run only a group of test cases or skip them.
- **Reports and Logs:** Robot framework provides all the details of test suite, test case execution in the form of report and logs. All the execution details of the test case are available in the log file. The details like whether the test case has failed or passed, time taken for execution, steps followed to run the test case are provided.
- **RIDE:** This editor available with Robot framework helps in writing and running test cases. The editor is very easy to install and use. RIDE makes life easy for writing test cases by providing framework specific code completion, syntax highlighting, etc. Creation of project, test suite, test case, keywords, variables, importing library, executing, tagging the test case is easily done in the editor. Robot framework also provides plugins for eclipse, sublime, Textmate, Pycharm that has support for robot test cases.

Robot Framework Advantages

- Easy to install and helps in creating and executing test cases. Any new comer can easily understand and does not need any high level knowledge of testing to get started with robot framework.
- Supports keyword-driven, behaviour-driven and data-driven style of writing test cases.
- A good support for external libraries. Most used is Selenium Library, which is easy to install and use in robot framework.

Robot Framework Disadvantages

Robot lacks support for if-else, nested loops, which are required when the code gets complex.

3.3 RIDE INTRODUCTION

Run the command ``ride.py`` on terminal to start Ride IDE. The interface of Ride looks like below

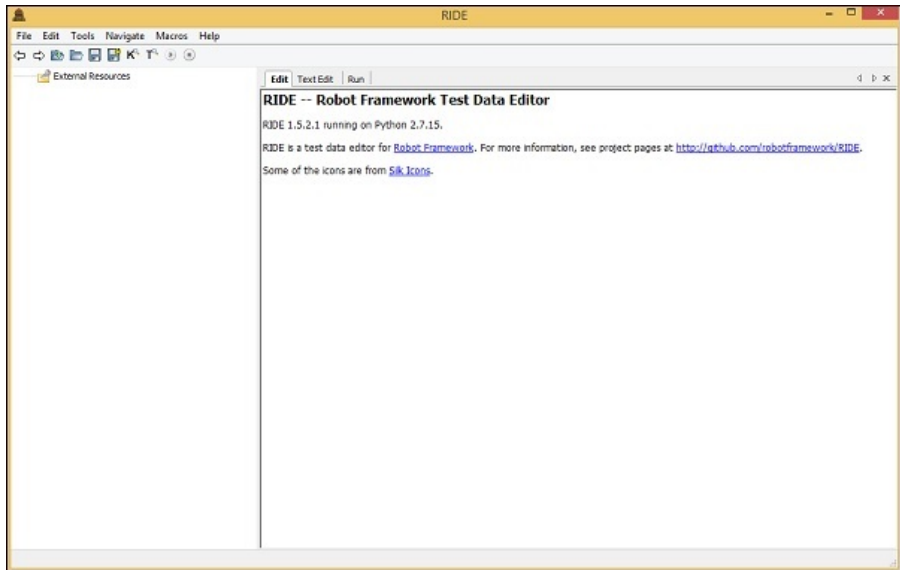


Figure 1 - Ride

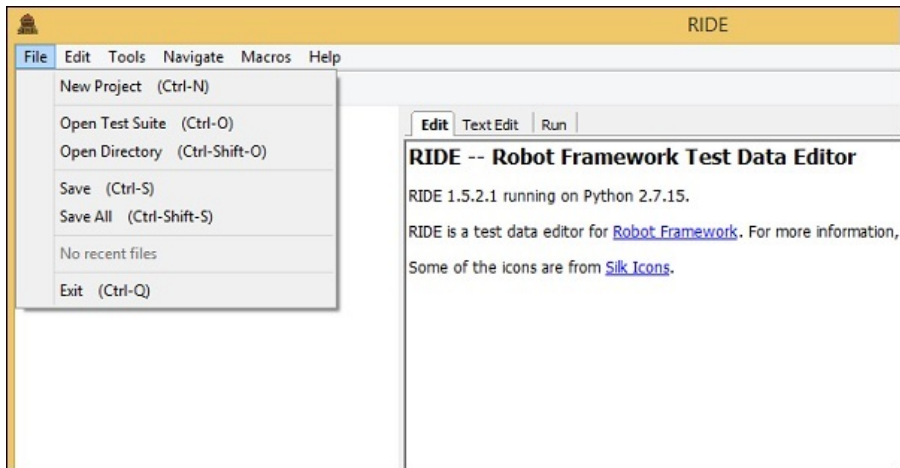


Figure 2 – New project

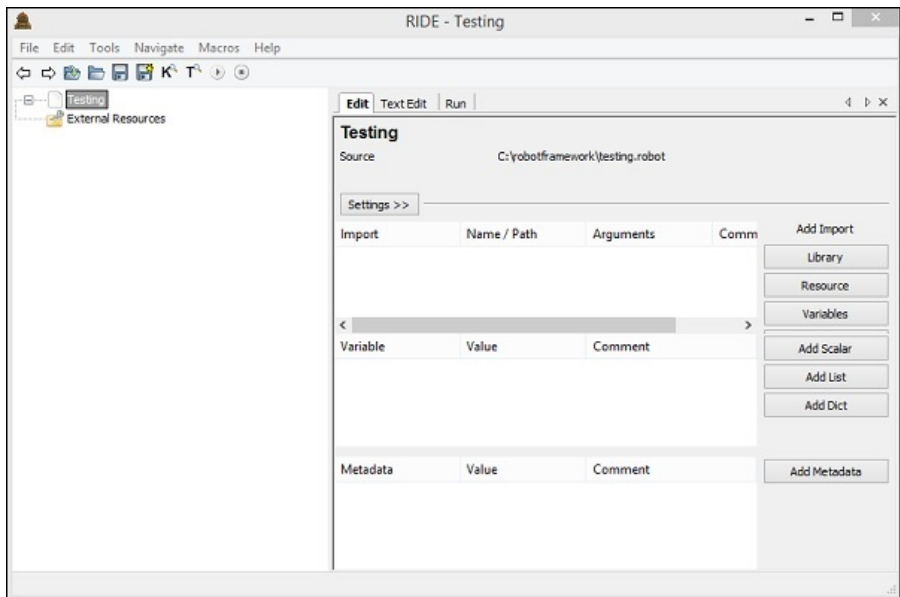


Figure 3 – Test editor

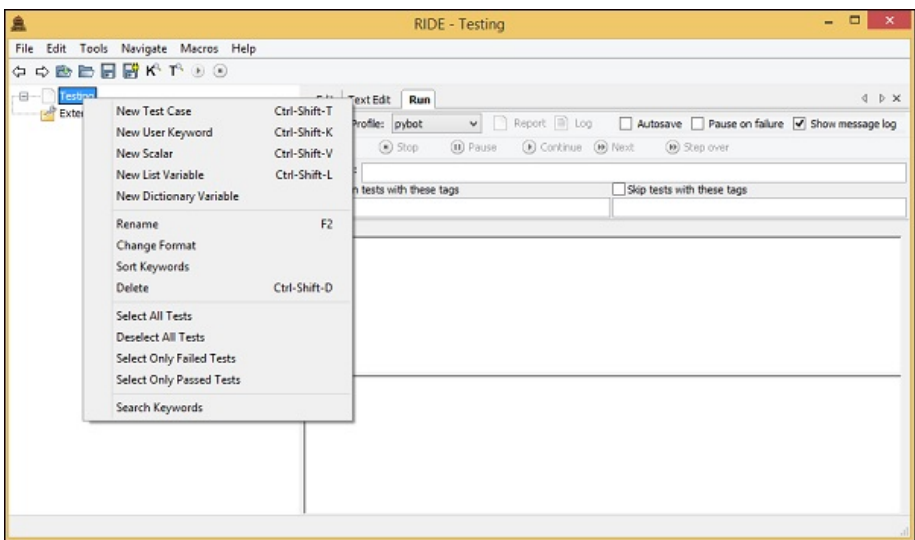


Figure 4 – New test cases

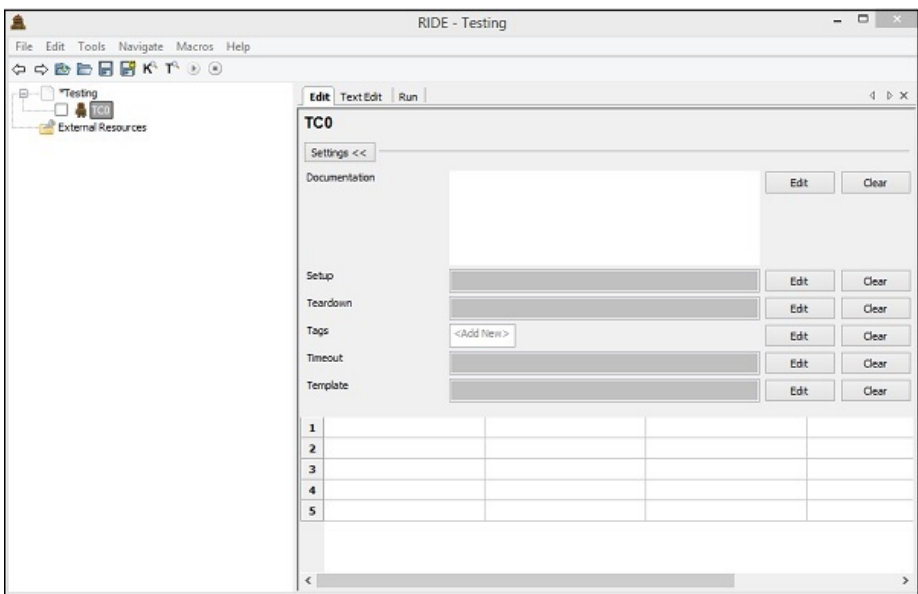


Figure 5 – Test case editor

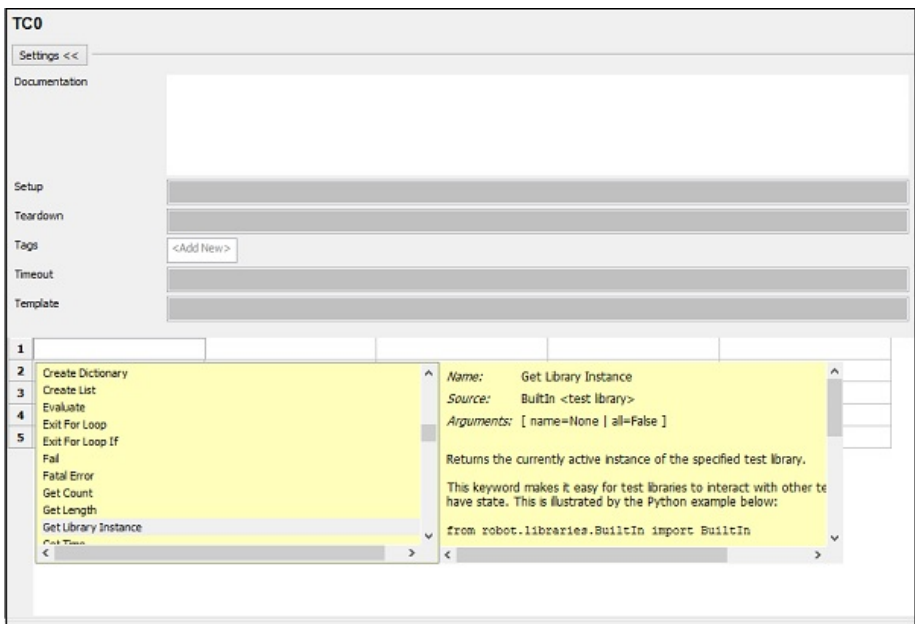


Figure 6 – Keywords hint

- To get the list of keywords available with robot framework, simple press ctrl+space in the tabular format as shown below and it will display all the keywords available
- Search here:
<https://robotframework.org/robotframework/latest/libraries/BuiltIn.html>

3.4 EXAMPLE OF TEST EXECUTION AND RESULTS (PASS/FAILED)

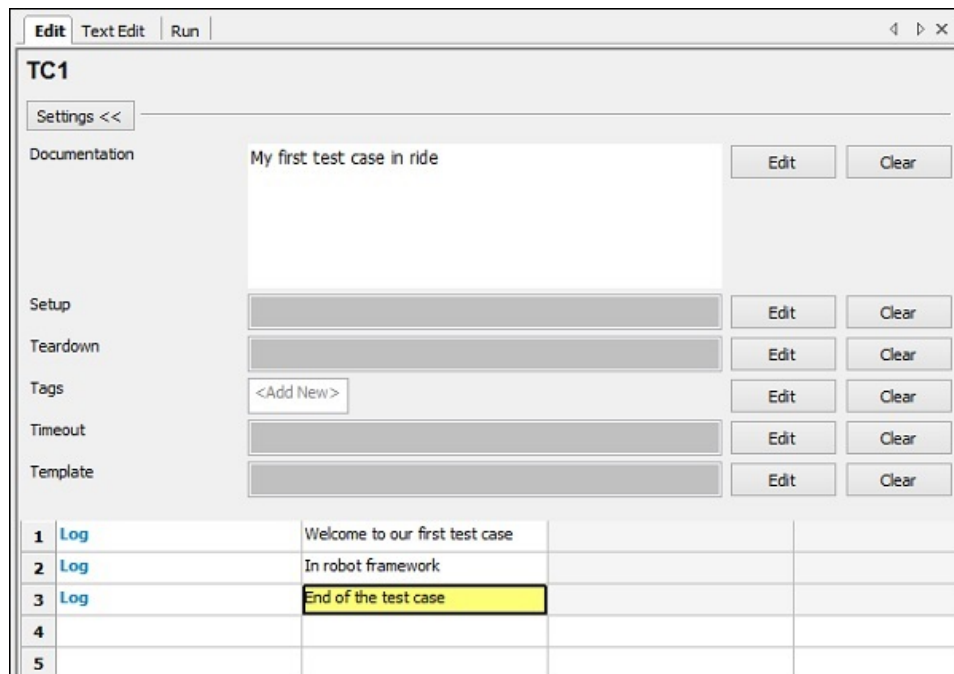


Figure 7 – Valid test

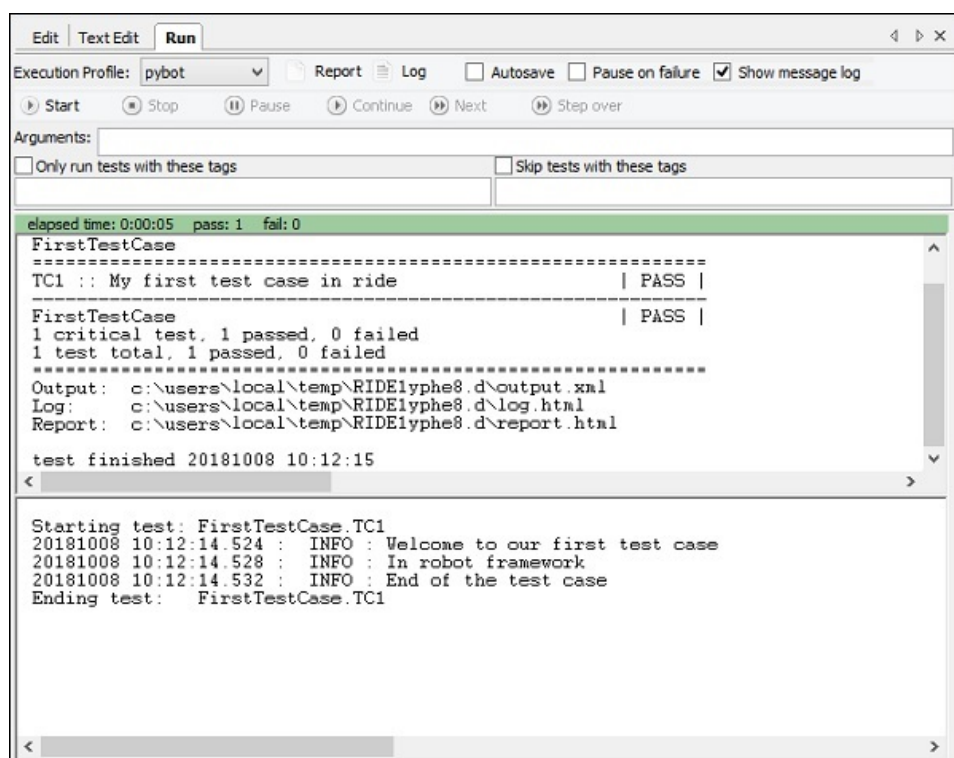


Figure 8 – Test run

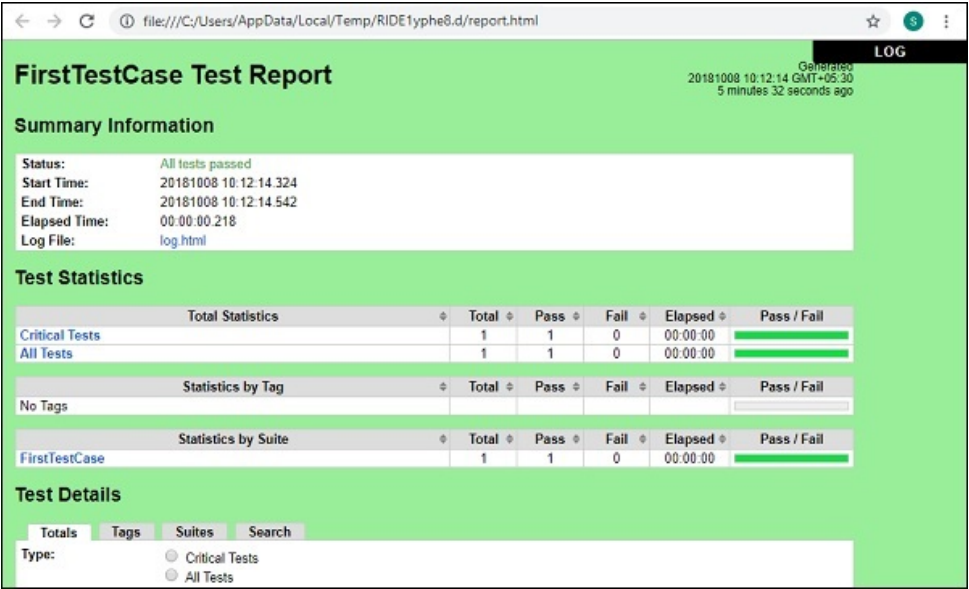


Figure 9 – Test report/pass

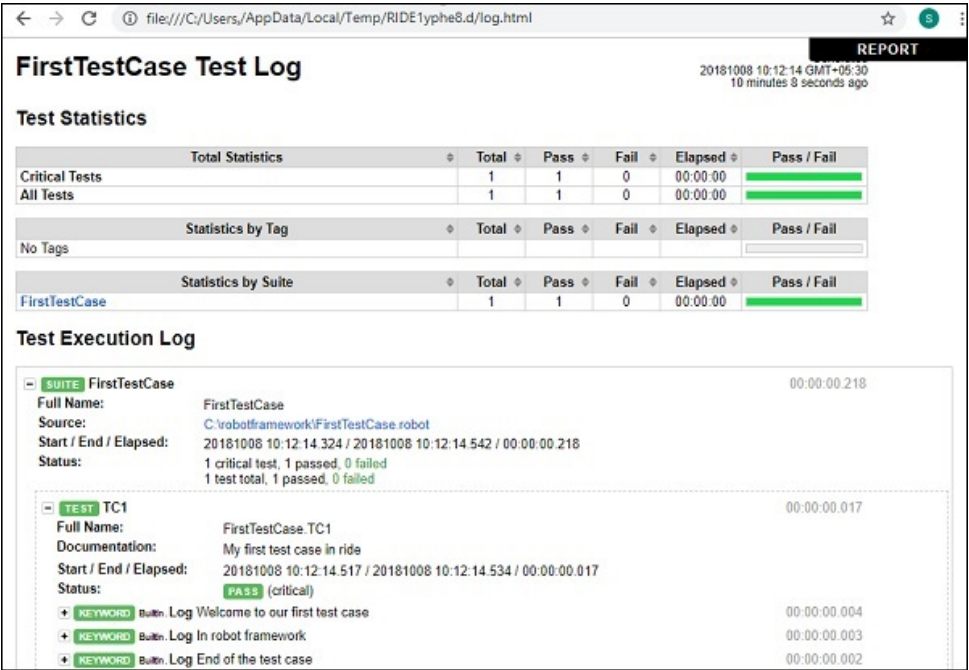


Figure 10 – Test log/pass

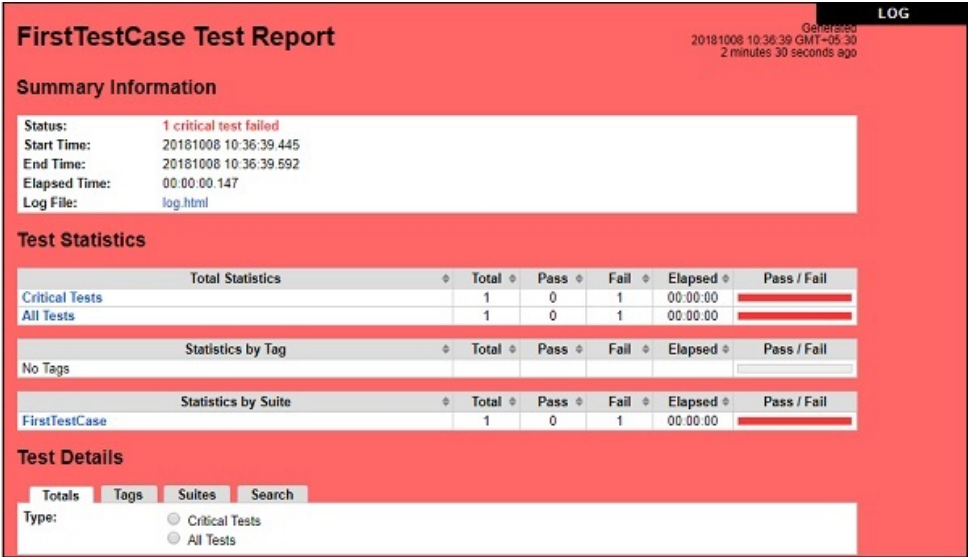


Figure 11 – Test report/failed

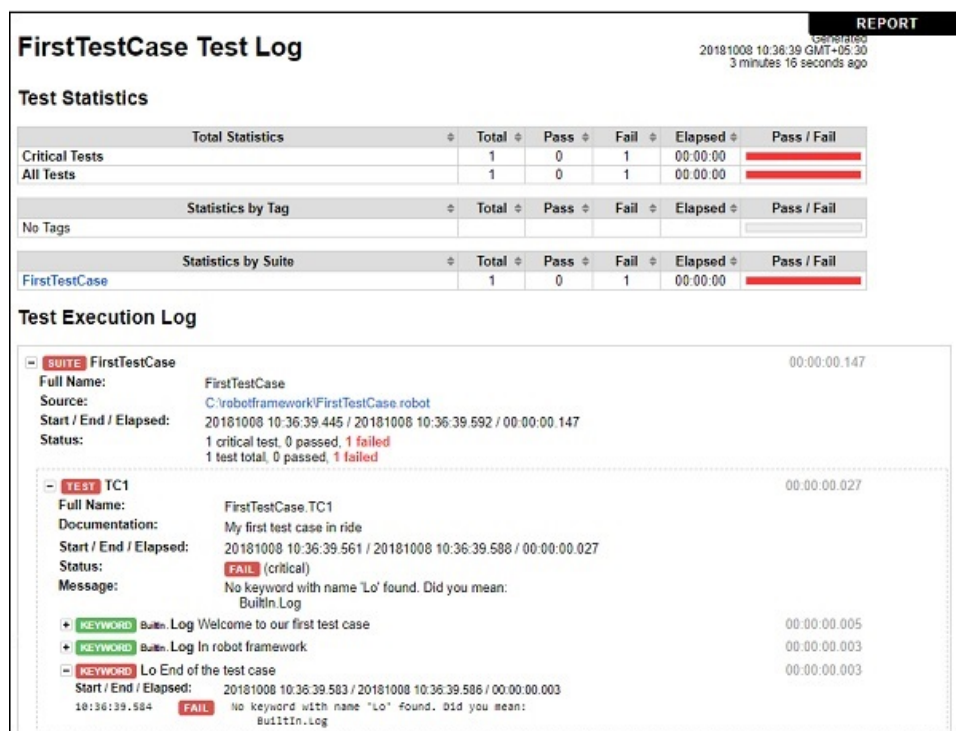


Figure 12 – Test log/failed

3.5 PRACTICE

The official document page of Robot Framework provides users with detailed explanation and demo, students should refer to the page during the time working with this lab.

<https://docs.robotframework.org/docs>

<https://robotframework.org/robotframework/latest/RobotFrameworkUserGuide.html#introduction>

Let create and run the following test scenarios and collect the results:

- **Assignment 1:** Open TDTU homepage on browser (chrome), minimize, maximize browser window. Finally, close browser
- **Assignment 2:** Define custom keywords (user-defined keywords) as follows:
 - Keyword1: `CompareNumber` \${arg1} | \${arg2}
 - Keyword2: `DisplayComparisonMsg` \${comparison_result}
- **Assignment 3:** Refer to the following document to understand how to create Library to write your own Library and test it with Robot Framework

https://docs.robotframework.org/docs/extending_robot_framework/custom-libraries/python_library

- **Assignment 3.1:** Write a python library to solve the quadratic equation for degree two.
- **Assignment 3.2:** Define and run test cases for the quadratic equation for degree two just developed, and then collect the report.

4. REFERENCES

[1] Daich, G., Price, G., Ragland, B., Dawood, M. "Software Test Technologies Report." STSC, Hill Air Force Base, Utah, August 1994.

[2] <https://robotframework.org/>

5. REVISION HISTORY

Revision	Date	Author(s)	Description
1.0	Dec 2023	Nguyen Thanh Quan (MSc)	Created