ZyLab Library Automation Test Report

## Version 1.0

**Prepared By: Unni Kannan**

***Introduction:***

The purpose of this document is to give high level overview about testing roadmap that will be followed for testing of **Zylab** application. This document will list the detailed understanding about the Testing workflow that will be followed during Testing Life cycle.

# System Description:

**Zylab** application used to maintain Books information. A new user can list the books, search the books, can delete the books, update/modify the details of the existing books.

# Scope of Testing:

### In-Scope:

1. ***Functional Testing:***
   1. Automation testing should be done in order to reduce manual regression efforts for repetitive tests on Zylab Library Application.

### Non-Functional Testing:

* 1. UI testing should be done in order to verify that UI is not getting distorted after performing various actions and also to confirm that application is easy to use.
  2. Cross-browser testing should be done in order to verify that application works as intended on supported browsers.

### Out of Scope:

1. Performance testing
2. Security testing

# Test Strategy:

This section lists down the strategy to ensure the quality of Zylab Library application via manual & automation testing.

1. Manually explore application to understand basic functionality.
2. Identify features in application.
3. Design test scenarios for testing of various features in application.
4. Identify automatable test scenarios and mark the test cases accordingly.
5. Design test scripts for automation testing
6. Execute both manual and automation tests against application in test.
7. Analyze test results to identify defects in application.

# Tools used for automation:

List of all the software and tools required for developing and executing automation framework.

1. Java (version 1.8)
2. Selenium (version 3.141.59)
3. TestNG (version 6.14.3)
4. Maven (version 4.0.0)
5. Log4j (version 1.2.17)
6. Google Chrome (version 79.0.3945.130)
7. Mozilla Firefox (version 72.0.1 (64-bit))

# Automation Scripts Execution Steps:

To execute test scripts please follow below steps:

1. Install Java & maven and set their respective paths in system variables.
2. Clone project (git clone <https://github.com/unkannan/zelenium/tree/HomeBranch/ZyLabLibAutomation> )
3. Or Extract the zip file
4. Open COMMAND PROMPT (cmd)
5. Navigate to the folder ‘ZyLabLibAutomation’
6. Clean and compile project using

*mvn clean compile*

1. To execute automation scripts, execute any of the following commands from command prompt:
   1. To execute all automation scripts (by default it will run on firefox) mvn test
   2. To execute on particular browser, execute below command with parameter as 'firefox' or 'chrome'

mvn test -Dbrowser=*<browser-name>*

**for e.g.:** mvn test -Dbrowser=firefox

or mvn test -Dbrowser=chrome

1. Logs available in Log folder
2. Test reports are available in below location

# \target\test-output\emailable-report

# \target\test-output\index.html

# Test Deliverables:

1. ***Functionality Analysis:*** This section contains the details about total number of manual test cases per functionality. Below table contains total number of manual test cases per functionality. For details refer **Test Cases** worksheet in **"Zylab Library Consolidated Report.xlsx".**
2. ***Automated Details:*** *Automation test scripts are available at Github location*

<https://github.com/unkannan/zelenium/tree/HomeBranch/ZyLabLibAutomation>

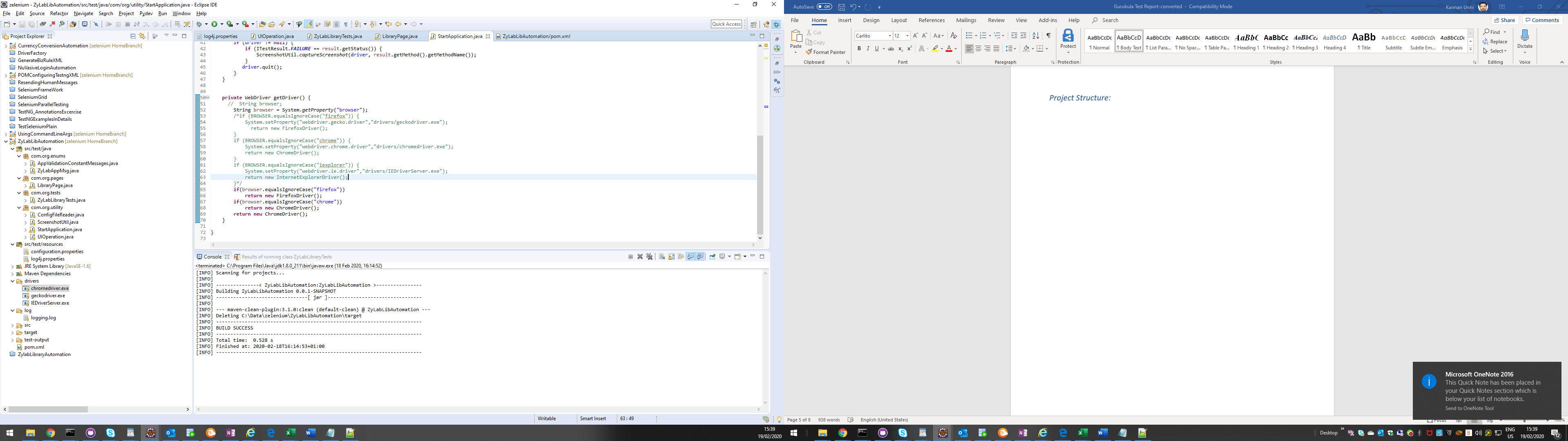
1. ***Defect List: List of defects*** found while executing manual & automated test scenarios is reported in **'Defects'** worksheet in **" Zylab Library Consolidated Report.xlsx ".**

# Automation Project structure:

This project is implemented in Page Object Model (POM) which helps in making the code more readable, maintainable, and reusable. Mentioned below are the description about different components of framework:

1. ***pom.xml:*** This file is used to perform build actions(clean, compile, install, test etc.) on project. All dependencies are included in pom.xml file.
2. ***drivers:*** This folder contains browser driver executables for chrome and firefox drivers.
3. ***log:*** This folder contains auto-generated log file.
4. ***com.org.enums:*** This folder contains common constants and enums which are present throughout the application.
5. ***Com.org.pages:*** This folder contains files corresponding to each functionality and has locators and methods to be used in the test methods.
6. ***Com.org.tests:*** This folder contains test files corresponding to each functionality. Each file contains tests to test the functionality end to end.
7. ***Com.org.utility:*** This folder contains classes to be utilized throughout the framework and is responsible to instantiate driver, read from properties file etc.
8. ***resources:*** This folder contains properties files which has details about the test application(url, username, password) and log4j properties.

*Project Structure:*



# Automation Coverage:

Below table & bar chart contains the total automation coverage of **Zy Lab Library** application. Out of total 21 manual test cases, 100% test cases are automated.

|  |  |
| --- | --- |
| **Automation Coverage** | |
| **Total Number of test cases** | **21** |
| **Not Automated** | 0 |
| **Automated** | 21 |

# Automation Execution Status:

Below table & bar chart contains execution status of **Zylab Library** automation.

|  |  |
| --- | --- |
| **Automation Execution Status** | |
| **Pass** | 19 |
| **Fail** | 2 |
| **Skip** | 0 |
| **Total:** | **21** |

## Notes:

1. All **2 failures** are valid **defects** and are logged in the **'Defects'** worksheet in

**‘Zylab Library Consolidated Report.xlsx.xlsx".**

1. Framework has capability to **capture screenshot** for every failed test scripts. All the screenshots are stored at **". \target\screenshots".**

# TestNG Result:

# 1. Verify emailable-report at below location

# \target\test-output\emailable-report



+

# 2. Verify testing reports here

# \target\test-output\index.html

