|  |  |
| --- | --- |
| TEST AUTOMATION FRAMEWORK | Abstract  This Document provide high level view on the automation framework design and capabilities |

## 

## 

Framework Contents

[Overview of Framework 1](#_Toc98846599)

[Prerequisites 1](#_Toc98846600)

[Execution 2](#_Toc98846601)

[Framework Architecture 2](#_Toc98846602)

[Maven Dependency 3](#_Toc98846603)

[Framework Components 3](#_Toc98846604)

[ReportSample 5](#_Toc98846605)

[CircleCI Integration 6](#_Toc98846606)

# Overview of Framework

The friday-selenium framework based on Page Object Model, Selenium, TestNG using Java. Which can be used across different web based applications.

* TestNG makes automated tests more structured, readable, maintainable and user-friendly.
* Selenium is the most popularly used freeware and open source automation tool. Selenium easily integrates with various development platforms such as Jenkins, Maven, TestNG, QMetry, SauceLabs, etc
* **Page Object Model (POM)** is a design pattern, popularly used in test automation that creates Object Repository for web UI elements. The advantage of the model is that it reduces code duplication and improves test maintenance.

# Prerequisites

* Java jdk-1.8 or higher
* Apache Maven 3 or higher
* Eclipse/IntelliJ IDEA IDE

# Execution

* Clone the repository.
* Open command prompt and go to project directory.
* Run mvn clean install -Dbrowser=chrome
* By default tests are run using Google Chrome.

|  |  |
| --- | --- |
| Browser | Example |
| Chrome | -Dbrowser=chrome |
| Firefox | -Dbrowser=firefox |
| Edge | - Dbrowser=edge |

# Framework Architecture

Graphical user interface, diagram, application

Description automatically generated

# Maven Dependency

Below is the dependency included in the POM .XML

* TestNG
* Selenium
* extentreports
* poi
* log4j
* webdrivermanager

# Framework Components

The framework consists of the below major components

* Test class
* Business specific library
* Generic Utils
* Page Object
* Reports Generation

Project structure:

A screenshot of a computer

Description automatically generated with medium confidence

Test Class:

This consist of the baseclass and the test class(in our case insuranceTest).Insurance Test extends the baseclass for the TestNG annotation

Pagefactory

bussinessspecific

BaseTest

InsuranceTest

Extends the base test

Initiates brower & return testData

Call the resuable function for the vehicle selection

Call the Webelemt and perform t

the operation like click, select,sendkey

Response returned back

The control returned to insurance test from the bussinessspecific

Business specific library:

This has the logic for the vehicle selection and also the insurance precondition selection.

Generic Utils:

The function related to the excel,reports config, reports utils are available under the utils folder.

PageObject:

Page Factory is a class provided by [Selenium WebDriver](https://www.browserstack.com/guide/selenium-webdriver-tutorial) to support Page Object Design patterns. In Page Factory, we have used @FindBy annotation. The initElements method is used to initialize web elements

Report Generation:

Extent report library is used for the report generation, we have used the logging message to add the test steps and also to capture the screenshot on the failures.

# ReportSample

* Overview of the execution results, total pass, failed,system details

Graphical user interface, application

Description automatically generated

* Detailed steps on each testcase and also the screenshot for required steps.

Graphical user interface, application

Description automatically generated

* During failure we have exception and the screenshot captured.Graphical user interface, application

  Description automatically generated

# CircleCI Integration

This project is integrated with the circleCI and for each code change the build will be triggered and the execution is done and the reports are logged in the artifact.

Graphical user interface, text, application, email

Description automatically generated

Reports are uploaded

Graphical user interface, text, application, email

Description automatically generated