Automation guidelines

Table of Contents

[Generic 1](#_Toc2054459654)

[Automaton selection criteria 1](#_Toc20026641)

[Automation Tool setup 1](#_Toc822648017)

[Repository 1](#_Toc467892839)

[Local Machine Setup 2](#_Toc2127053601)

[Coding Standards 2](#_Toc1813217592)

[Reporting 2](#_Toc1765340867)

[Sample Reports 2](#_Toc761420731)

[Screenshots 4](#_Toc785263865)

[Logs 4](#_Toc390064730)

[Integration 5](#_Toc1569390375)

[Jenkins 5](#_Toc1656316191)

[Jira 5](#_Toc1286562455)

[Troubleshoot Documentation…………………………………………………………………………………………………………….5](#_Troubleshoot_documentation)

# Generic

## Automaton selection criteria

Applications which fall under the criteria below will be considered for automation.

1. Customer facing applications.
2. Applications having critical transactions / functionalities.
3. Applications have high frequency releases.
4. Applications have new age architecture and cloud-based applications.

# Automation Tool setup

Automation engineers would be using CARINA framework for web and mobile automation. Refer attached document for framework setup.

Refer document - Carina Web Automation Framework Guide v1.0.docx from Setup&Integration folder.

# Refer document - Carina Appium Automation Framework Guide v1.0.docx Setup&Integration folder.

# Repository

Automation test engineers would be using source code management tool available with the bank and follow the standard practices on managing the code as per attached document.

# Refer document - Source Code Management Practices.docx from Setup&Integration folder.

# Local Machine Setup

Automation engineer local machine should have following setup.

* JDK / JRE installed with JAVE\_HOME configured.
* Eclipse with maven and TestNG plugin should be configured.
* CARINA framework files with base project setup in eclipse.
* Eclipse integrated with source code management tool and connected to the proper git branch.
* All applications under the scope should be accessible.
* Chrome/Edge browsers matching the machine’s browser version should be available.
* Access to maven repository should be enabled for downloading the dependencies.
* Access to device farms should be available.
* Android studio, Xcode, android SDK, NodeJS and Appium server should be installed with necessary configurations.

# 

# Coding Standards

Automation testers should follow attached standard coding practices throughout the project.

# Refer document - Automation Coding Standards\_V1\_0.docx from Setup&Integration folder.

# Reporting

Extent reports are integrated within the framework, which would be generated after every execution. Necessary screenshots and logs would be captured as part of the execution and will be available.

## Sample Reports

A screenshot of a computer

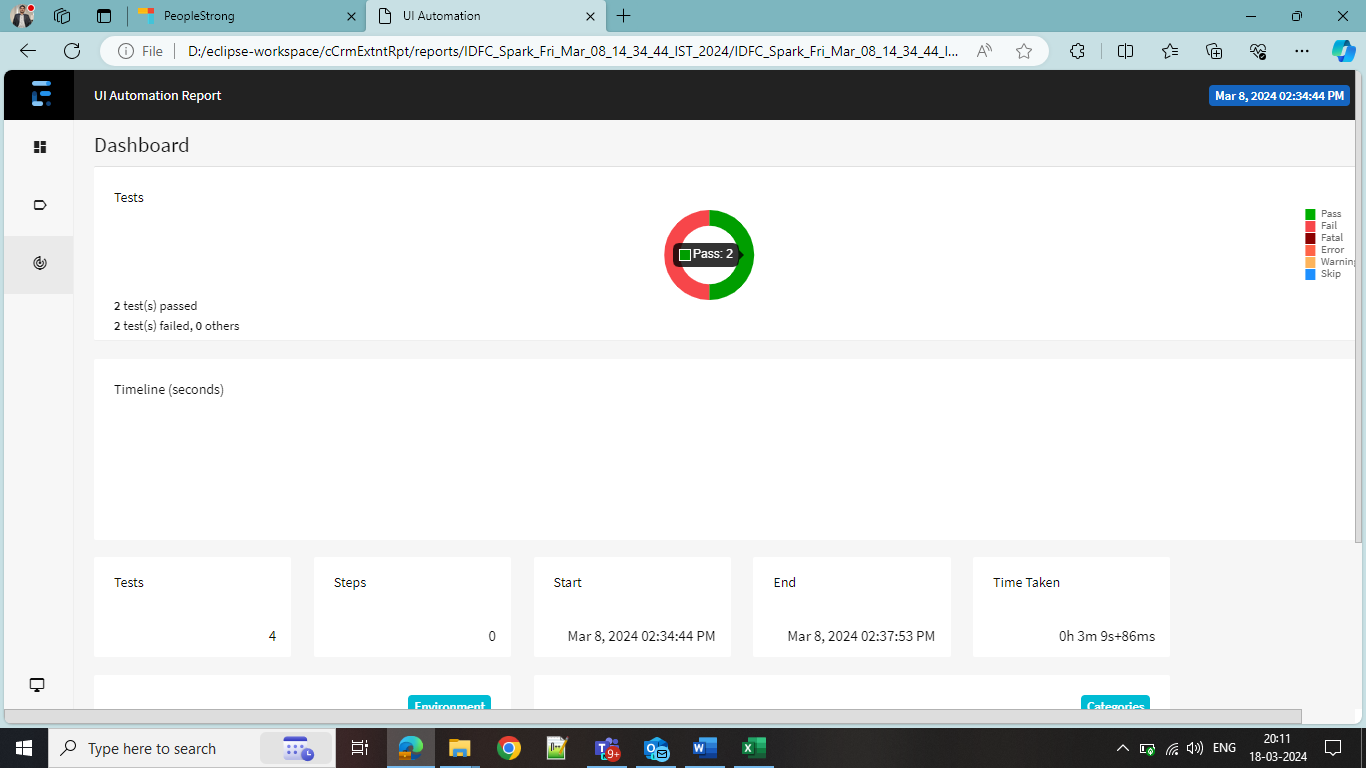
Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated



# A screenshot of a computer Description automatically generated

## Screenshots

Screenshots captured during execution are stored testcase-wise locally and will be accessible with the reports.

A computer screen shot of a button

Description automatically generated

## Logs

Logs are embedded along the scripts during script development and would be available with the reports.

A screenshot of a computer

Description automatically generated

# Integration

## Jenkins

* Once automation scripts are created, they should be identified and created as different test packs such as BVT, sanity and regression.
* As part of CI/CD integration, created test packs would be deployed in Jenkins build pipelines.
* These quality gates would validate the deployments on different environments.
* Pass threshold would be defined for each quality gate.
* Reports would be available from Jenkins.

## Jira

Refer document - Jira Xray\_Integration 1.pdf from Setup&Integration folder.

# Troubleshoot documentation

Refer document - Troubleshooting Documentation for Automation Setup.docx