**Carina: Proof of Concept**

**Task to accomplish**

Use Carina to ensure that the responsive website works as expected on various devices and platforms.

**What is Carina?**

Carina is a Java-based test automation framework that unites all testing layers: Mobile applications (web, native, hybrid), WEB applications, REST services, Databases. Carina Framework is built on top of the most popular open-source solutions like Selenium, Appium, TestNG allowing to reduce dependence on the specific technology stack.

**Key Features**

* Carina supports all popular browsers (Edge, Safari, Chrome, Firefox) and mobile devices (iOS/Android). A special feature for a mobile domain: it reuses test automation code between IOS/Android up to 70-80%.
* Carina is cross-platform. Tests may be easily executed both on Unix or Windows OS. All you need is JDK 11 installed.
* The framework supports different types of databases - both relational and non-relational.

**PRE-REQUISITES:**

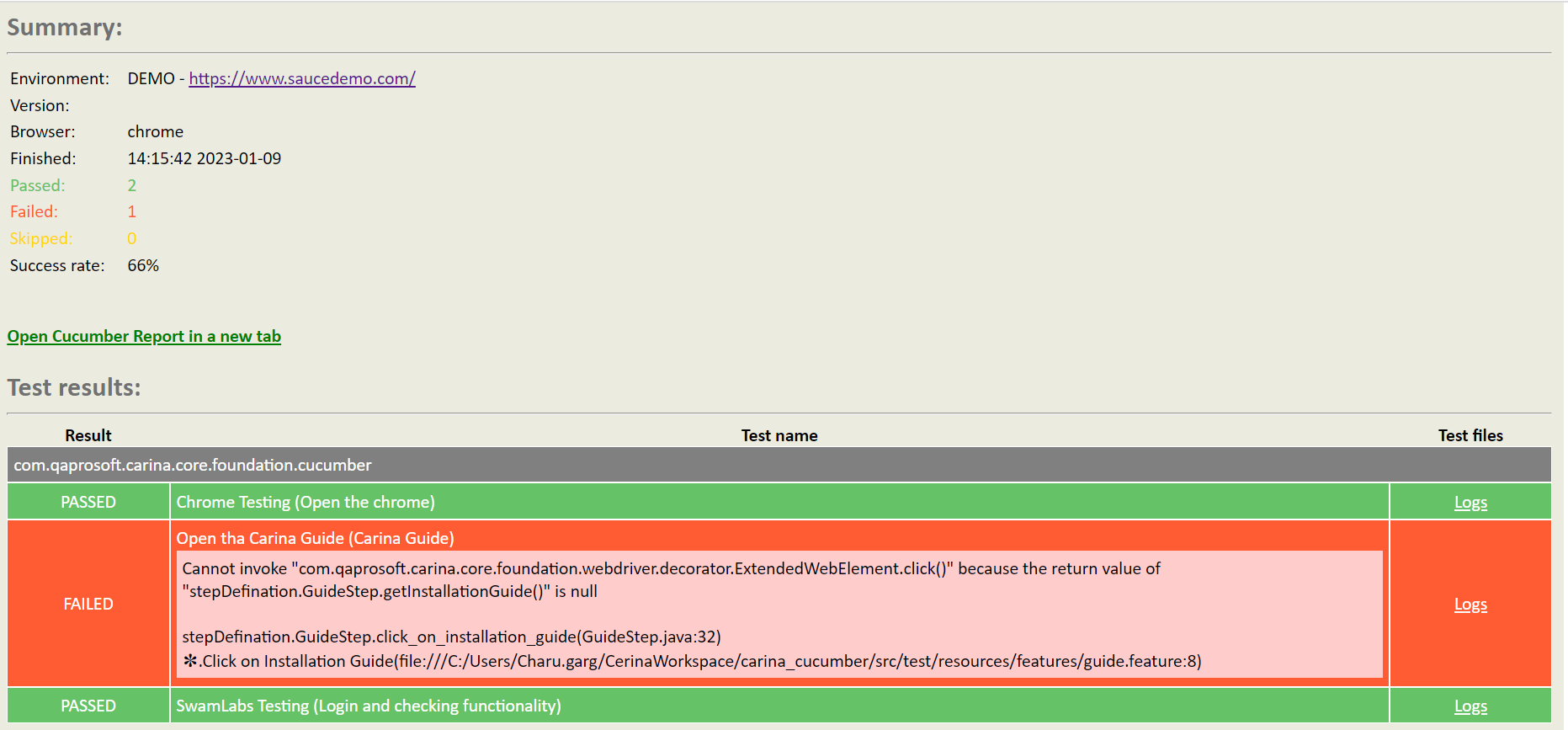
* Install and configure JDK 11+
* Install and configure [Apache Maven 3.6.0+](http://maven.apache.org/)
* Download the latest version of [Eclipse](http://www.eclipse.org/downloads/) and install [TestNG plugin](http://testng.org/doc/download.html), maven-dependency-plugin connector.

**Advantages of Carina:**

* Carina is free and open source software.
* Carina is a Selenium-based framework so no limitations at this point. Moreover, playing with the page opening and element loading strategies, you can tune this test automation framework to any tool set like React, Angular, Vue, Flutter, Google Web Toolkit and other popular framework.
* Unites all testing layers Mobile applications (web, native, hybrid), WEB applications, REST services, Databases.
* Carina supports all popular browsers (IE, Safari, Chrome, Firefox) and mobile devices (iOS/Android).
* Carina provides very thorough reporting for failed cases.
* It enables great flexibility in generating REST requests, and responses are dynamically changed by incoming arguments.
* Easy to learn as based on Selenium

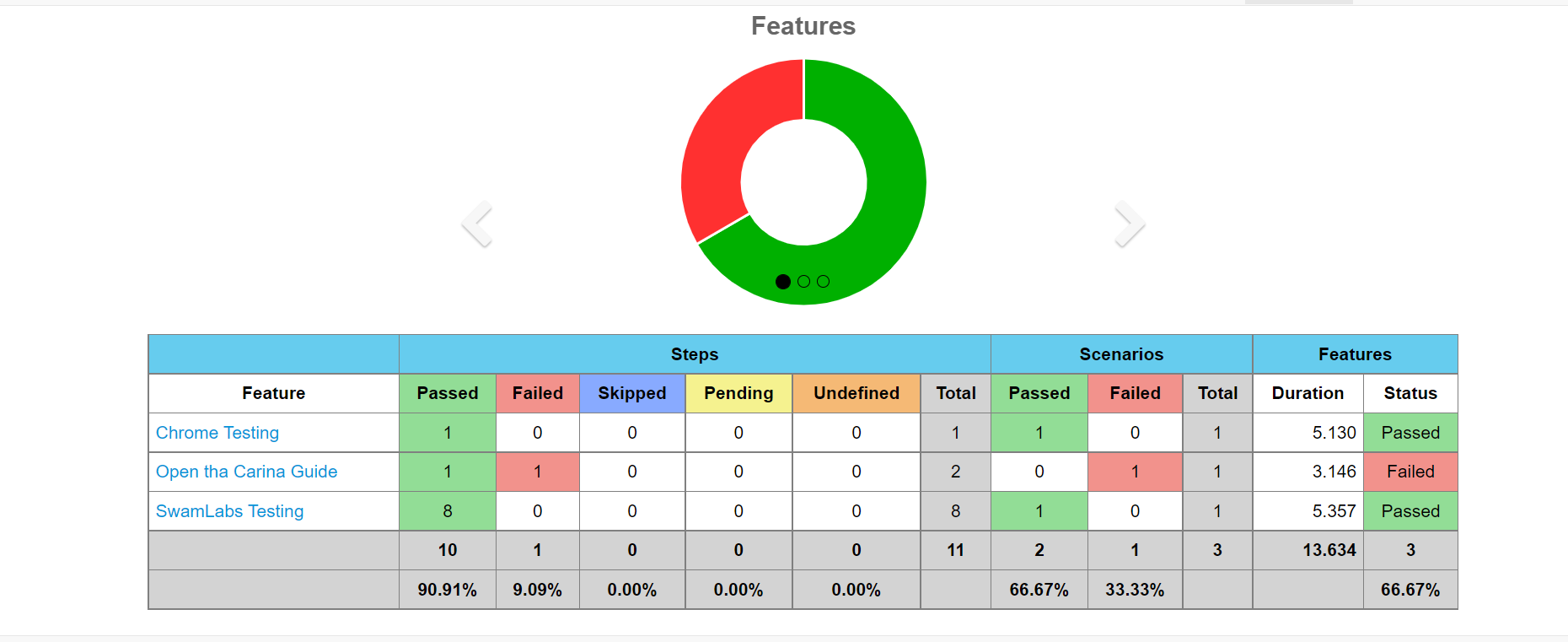
**Carina Reporting:**

An emailable HTML report generated by Carina. It's getting generated automatically once test is done.

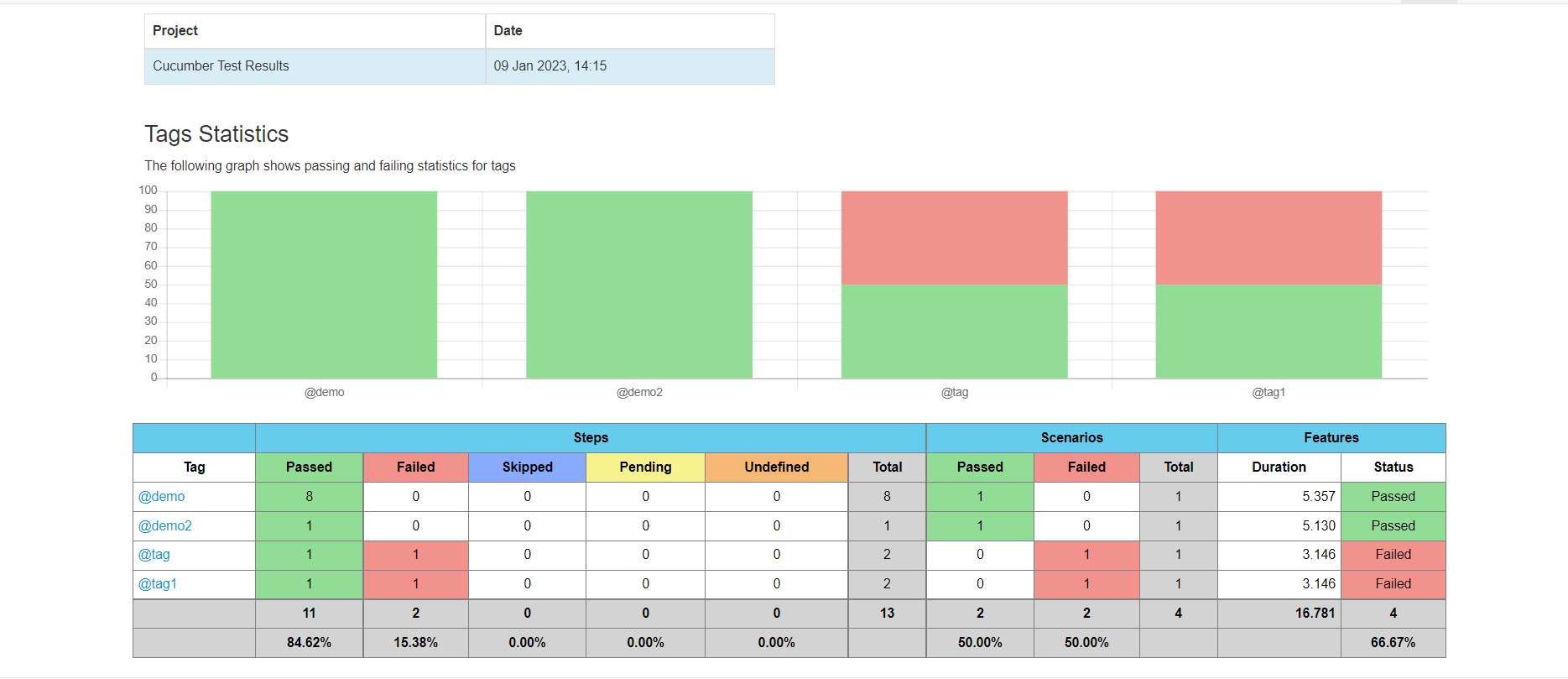


To get this link valid you'll need to specify proper path to your report folder using **report\_url** carina config property.  
Here is the sample of the report:

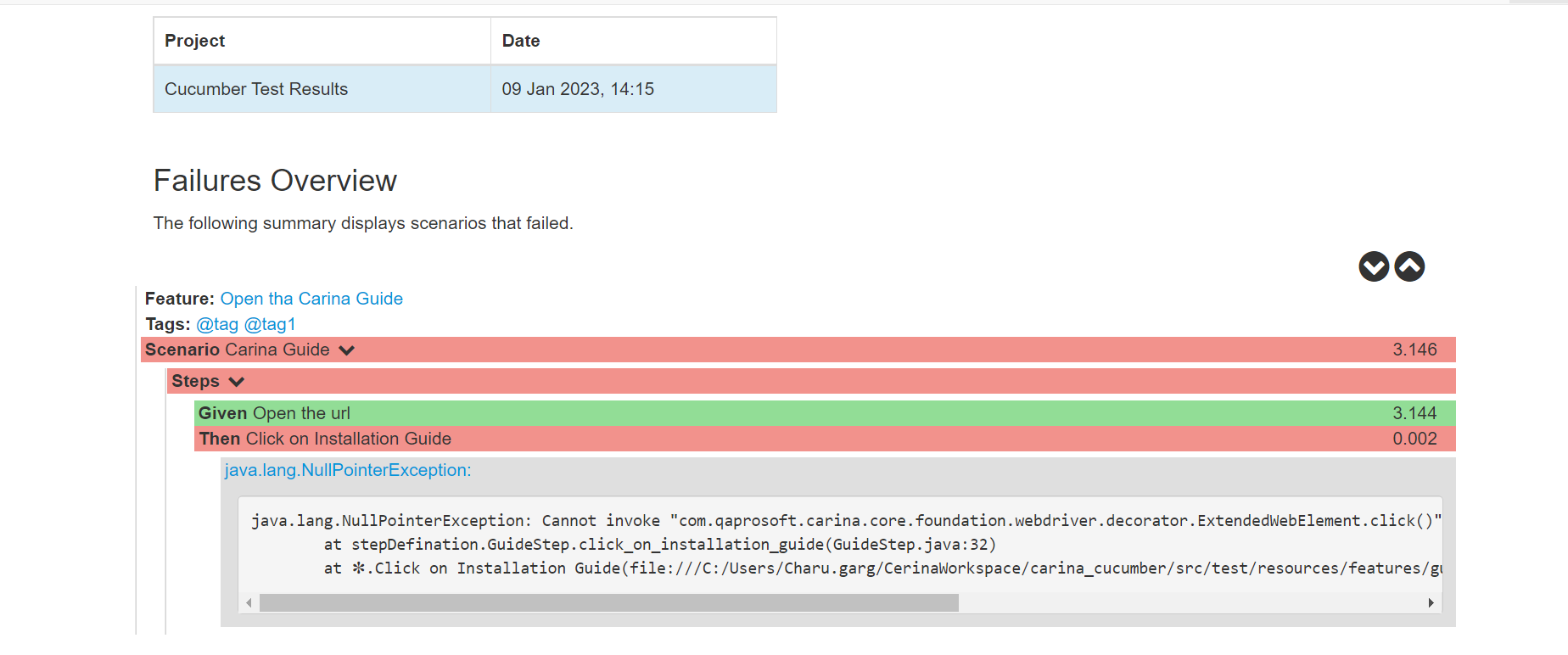
**According to features:**



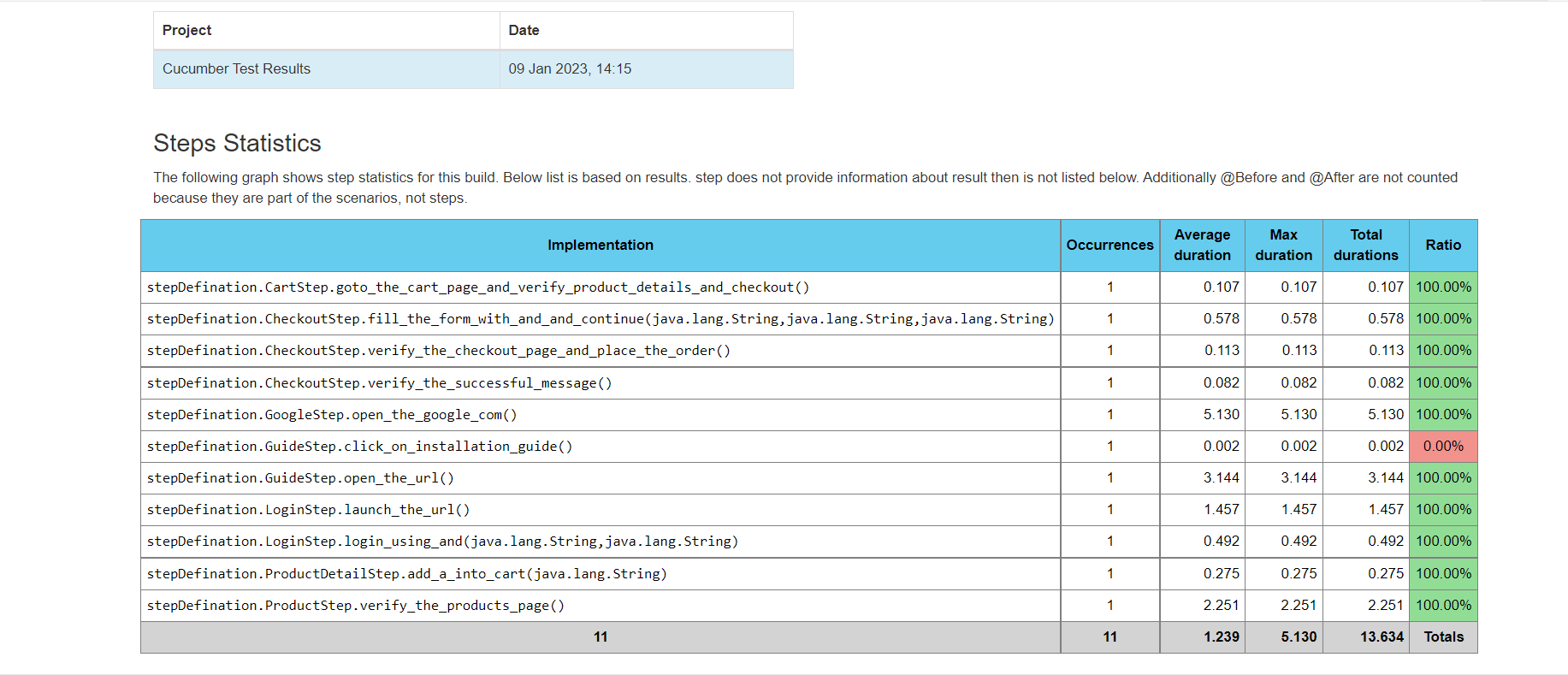
**According to Tags:**



**According to Failures:**



**According to Steps:**



**Point to Remember:**

* Page should extend **com.qaprosoft.carina.core.gui.AbstractPage**
* Use **com.qaprosoft.carina.core.foundation.webdriver.decorator.ExtendedWebElement** instead of Selenium WebElement.
* Locate Page Object classes in src/main/java
* Test method should start with **org.testng.annotations.Test** annotation
* Test class should implement **com.qaprosoft.carina.core.foundation.IAbstractTest**

**Disadvantages:**

* Very less community support.
* Customer support has slow response time.
* It doesn’t support Parameterized Testing
* It doesn’t support Move & Copy

**Values from POC:**

Describe all page objects on your own based on the most popular Page Objects pattern. Reuse Custom Type Page Factory to describe app specific pages (Web, IOS, Android, etc.). To make a cool test, you just need to add Verification Logic to your combined PageObjects and Unites all testing layers Mobile applications (web, native, hybrid), WEB applications, REST services, Databases.