* Java + Selenium WebDriver
* Eclipse Project (Make sure to have all the local dependencies bundled inside the project to ensure that it will run on other machines, create it as a Maven project, or use Maven-Ant tasks)
* Solution Requirements:
* Architecture: Hybrid Test Automation Framework (Modular \*reusable methods\* and Data-driven) with an Object-Oriented design
* Document your work
* Nothing should be hard-coded in the scripts:
* Test data & object identifiers (object repository) should be stored in external files (CSV, XML, Excel...)
* Selection of execution browser should be managed from a configuration file (any format), or in runtime
* Cross-browser support (Internet Explorer, Firefox, Chrome)
* Desktop Support (using any tool, like Sikuli, Winium Driver, White Driver or AutoIt)
* Use JUnit/TestNG tests to verify/assert that each action was performed successfully
* Generate a detailed execution status report containing the status of each of the executed steps and the executed test as a whole (HTML, Excel, etc…)
* Customized reports including screenshots for failures are preferred
* Report Assertion failures and other exceptions properly in the execution report
* Recover from assertion failures and other exceptions by closing the browser window, terminating test execution, and generating the report accordingly.
* Extras (Worth bonus points, but only if the required items were completed with a good quality):
* Support remote test execution using Selenium Grid
* Create a Keyword-driven framework to drive your solution
* Project delivery is left for you to decide
* Scenario:
* Preconditions: Create 2 email accounts on any email provider manually (because usually, this part will contain captcha verification).
* Steps:
* Login from Email account #1
* Send an email with a unique/dynamic title (with timestamp ex.: dd/mm/yyyy HH:MM) and an attachment to email #2 (use the desktop tool to upload the file).
* Log in to Email #2
* Verify the received email content and download the attachment file