# WEB UI Automation- Selenium with Behaviour-Driven Development

**Introduction:**

Cucumber is a BDD testing framework that allows people without programming background write specifications that can be translated to unit test requirements almost automatically.

**Used tools & frameworks**

* Java
* IDE (Eclipse)
* Maven
* Cucumber
* Selenium

**Dependencies:**

* Cucumber-core
* Cucumber-html
* cobertura code coverage
* Cucumber-java
* Cucumber-junit
* Cucumber-jvm-deps
* Cucumber-reporting
* Hamcrest-core
* Gherkin
* Junit
* Selenium-Java

Source:  <https://mvnrepository.com/search?q=Cucumber>

**Framework:** Behavioural Driven Development framework using Cucumber/Selenium

* Features folder contains application features
* Each feature file can write with multiple scenarios
* POM (page object model) is included
* Step definition is written with methods defined in feature file
* Common methods (wait, actions etc.) be maintained at Utilities
* Test runner is used to execute feature wise or w.r.t tags
* Tags implemented (ex; @sanity)
* Background steps included in feature file which is used for all scenarios in that feature file
* Hooks implemented to run method before and after each scenario
* Configuration file is defined to provide values to properties
* Multi browser testing is supported (chrome/firefox/IE)

An example,

An eCommerce shopping web application, implemented with below features

1. Order T-Shirt and the order validation in order's history

2. Updating personal information of account user

This translates to the following Gherkin syntax, which could be even written by the customer or project lead. This can easily be read and understood by non-programmers!

Graphical user interface, text

Description automatically generated

**Test Execution:**

1. It can run from Test runner.java > JUnit Test
2. It can run from pom.xml>maven.test
3. It can also extend to run from command prompt
4. It can run by building it from Jenkins

**Test report:** HTML Report

It can be allowed to use log4j or extent or AUT reporting etc. for future purpose.

**Future Work:**

1. To support parallel testing.
2. To integrate with external reporting tools for better visibility
3. To integrate with CI/CD pipeline using Jenkins