

Contents

[1 Overview 3](#_Toc20840320)

[2 Introduction to Behavior Driven Development (BDD) 3](#_Toc20840321)

[3 General BDD Guidelines to be followed 3](#_Toc20840322)

# Overview

This document contains the list of Behavior Driven Development (BDD) guidelines to be followed when creating automation scripts. These coding guidelines mentioned are inclined towards a java based project however the BDD guidelines are programming language independent.

# Introduction to Behavior Driven Development (BDD)

BDD is a methodology for developing software through continuous **example-based**

communication between developers, QAs, and BAs.

In BDD, examples are called **Scenarios**. Scenarios are structured around the *Context–Action– Outcome* pattern and are written in a special format called ***Gherkin***.

The scenarios are a way of explaining (in plain English) how a given feature should behave in different situations or with different input parameters.

# BDD Guidelines to be followed

**Generic Guidelines:**

1. Limit ‘Then’ (validation statements) in a scenario to 3 (some exceptions might exist)
2. Perform a spell check for each scenario/feature file.
3. Parameterization can be done in two different ways as below

Example 1: “Parameter” . This is one time and there is no need to mention it in examples

Example 2: ”<Parameter>” This is for multiple values required for same parameter. Name parameter with mentioned format and their value in examples.

1. Expected result should be precise and clear. It should not be like “desired value should be update”. Please mention what are these desired values.
2. Steps and expected result should be meaningful. lines not adding any value should be removed.

**Background Guidelines:**

* 1. Background is identified with Pre-requisite keyword in BDD cases
  2. Steps common for module /many TC should be part of pre-requisite.
  3. ‘Pre-Requisite’ needs to be provided depending on the scenario based on the need. It will run for each scenario sharing same Pre-requisite.
  4. Pre-requisite should not have Expected result.

**Scenario and Scenario Outline Guidelines:**

1. Scenario description is of the format: Verify <action, workflow>
2. Scenario keyword for no examples and Scenario Outline keyword with example
3. Each scenario should be complete in itself and should not have multiple validations.

**Steps Guidelines:**

1. Declarative style of writing a feature file will be followed. This means that the GWT statements would have what needs to be done without providing much information on how’s
   * 1. Given – would contain a pre-condition
     2. When – action/step by the actor(user)
     3. Then – outcome/result that is to be verified
2. Capitalize Gherkin keywords
3. One scenario should cover only one responsibility i.e. Atomicity

E.g.: Login tests should be part of login feature files. This file should not contain any scenario that does not test login feature

1. Each scenario should run independently without any dependencies on other features or scenarios. It can have a prerequisite though as a “Background”
2. Mention application name for login statement
3. In case of common actions, GWT statement should contain the name of the page on which the action needs to be performed

E.g. click search button on find a provider page

1. Adjust the parameters (that pass from the GWT statements) in middle of the sentence instead of using it at start or end
2. 1 Line should have 1 when only.
3. Expected result column for Given statement should not be empty.
4. Step descriptions use parameters in the format “<>”

**Database Statement Format:**

1. For writing DB name we have 3 cases.

Case-1:  When user want to connect Accretive database and want to fetch data from Accretive database using DB query.

Case-2:  When user want to connect Tran database and want to fetch data from Tran database using DB query.

Note: In Tran database user can connect with any facility like (SJMA, SJMC, SJOK etc.)

Case-3:  When user want to connect AHcrossite database and want to fetch data from AHcrossite database using DB query.

**Solution:**

* For cases like  #1 and #2, you should be writing in below format: (connects by default to Accretive database and tran db through facility name)

When user login to SQL server and connect to facility database

And user runs the <queryName> query

* For cases like #3, format will be like: (Synonym not available or can’t be used i.e. need to directly connect to DB other than Accretive)

When user login to “DBname” server

And user runs the <queryName> query

* In case#3, if facility wise connection is required too, it should be as below:

When user login to “DBname” server and connect to facility database

And user runs the <queryName> query