BDD:

BDD is nothing but, it is an approach which is used in Agile projects…

Basically, the main purpose of this BDD approach is to write these requirements / testcases in a template format using cucumber Gherkin Syntax

In such a way that all the team members of the project like Stake holders / clients, developers, testers, , product owners / BAs understand these requirement in the same perspective....

Example for **before** BDD / cucumber Gherkins syntax and **after** BDD / cucumber Gherkins syntax

<https://www.guru99.com/gherkin-test-cucumber.html#:~:text=After%20Gherkin-,Gherkin%20Syntax,should%20start%20with%20a%20%23%20sign>.

So, Step 1 is, we get these user stories / requirements which are written in Gherkin Syntax as:

For Ex say:

There is a user story written in Gherkin syntax as

**In order to** ->: Here is the Business Outcome (Achieve something)

For Ex: In Order To -> Achieve Creating a New Account Registration OR

-> Achieve Adding a private wish list

-> Achieve Paying the cart total using Credit Card etc.,

**As a** -> Here is the Role

For Ex: Is it as a Customer / or as an Admin / or as a Merchant u want to create a New Account

**I want To** -> Here Actions are performed

For Ex: Create A New Customer Account

And then, Step 2 is, we start writing Test Cases on these User Stories again in Gherkin Syntax as:

**Given:** Here we write pre-requisites, like

For Ex: User shall initiate the browser, launch the application

And User is on landing page

**When:** Here we perform actions, like

For Ex: User shall enter valid values of FN, LN, DOB, Email, Password etc (Ph #, Address)

And Click on submit “Button”

**Then:** Here we do all the validations and navigations, like

For Ex: User successfully creates a New Customer Account

And User shall navigate to Home page

And Home page welcome message should be validated

**Framework:**

We use Maven Build Tool to build our cucumber project

and give “Group ID” and “Artifact ID”

Group ID: EBS – Enterprise Business Services

Artifact ID: SSM – Sales Services and Management

Add all the needed dependencies to POM.xml file for the cucumber to Up and Running

cucumber -java and cucumber -Junit jars

We use “Natural” (Natural 0.7.6v) plugin (Natural Plugin is added Eclipse) to write the testcases in Gherkin Syntax and

We maintain all our work in 3 different files or components of cucumber which are created under src/test/java, files or components like:

Feature Files: Here we maintain all our manual test cases which are written in Gherkin Syntax, with “Given”, “When”, “Then” Scenarios…

Step Definition File: Here we maintain all our automation code for the features in “Features file” and then is,

Test Runner File: In this file we give @RunWith & @CucumberOptions annotations.

we use this file to run our test cases by giving the path of the “feature” package and “stepDefinition” package which are linked in @CucumberOptions annotations.

For Ex: @RunWith(cucumber.class)

@CucumberOptions(features="src/test/java/features", glue="stepDefinitions")

We also use “Tidy Gherkin” plugin (Tidy Gherkin Plugin is added to Chrome browser) to automatically generate all the Step-Definition automation codes from the feature file

We