Fluent Design Pattern:

 The term *Fluent interface* was coined by *Martin Fowler* and *Eric Evans*. Fluent API means to build an API in such way so that it meets the following criteria:

* The API user can understand the API very easily.
* The API can perform a series of actions to finish a task. In Java, we can do it with a series of method calls (chaining of methods).
* Each method's name should be domain-specific terminology.
* The API should be suggestive enough to guide API users on what to do next and what possible operations users can take at a particular moment.

In terms of test automation and frameworks, the Fluent design pattern was introduced which makes use of best practices of JAVA programming language, OOPS concepts and the best from multiple design patterns like page object model (POM), Singleton, Builder pattern and static factory methods.

**Page class:**

The below image depicts the structure of page classes in our framework.

A screenshot of a computer

Description automatically generated with medium confidence

Here we take Page Object Model one step above from class level to package level. Thus, we can leverage the power of POM to better organize our code for increase in maintenance speed and reduction of duplicate code.

Page class contains the protected static locators.

The Singleton pattern in fluent interface is generally used for WebDriver object.

Let us take an example of HomePage.java

Text

Description automatically generated

Page class also contains a static factory method called getHomePage() which in turn creates the instances of the three controllers called HomeActController, HomeVerifyController and HomeGetController.

**Test plan:**

Consider the below code image in test class:

Text

Description automatically generated

The method – userClicksTweetButton()

is part of actController of HomePage which returns the object of the next page – this is called page chaining.

Graphical user interface, text, chat or text message

Description automatically generated

And the next method getting called – userNotAllowedToTweet()

Is part of verifyController of TweetModalPage.

Text

Description automatically generated

Thus, we can use the Builder pattern fluency technique to generate a fluent design for our test case.