**FrameWork Introduction:**

Framework is designed based on the design pattern approach- POM(Page Object Model) and it integrates both the API and Browser automation Testing.

To interact with the webElements of Web application, selenium with Java is used and the test run is done using TestNg framework.

External data like Excel files are handled with Apachi POI which helps to access multiple data required by the project.

BDD approach also been combined with the framework with the aid of Cucumber tool. Html reports are been generated for each step of the Testcases using Extent Report.

All these integration is achieved through Maven build tool.

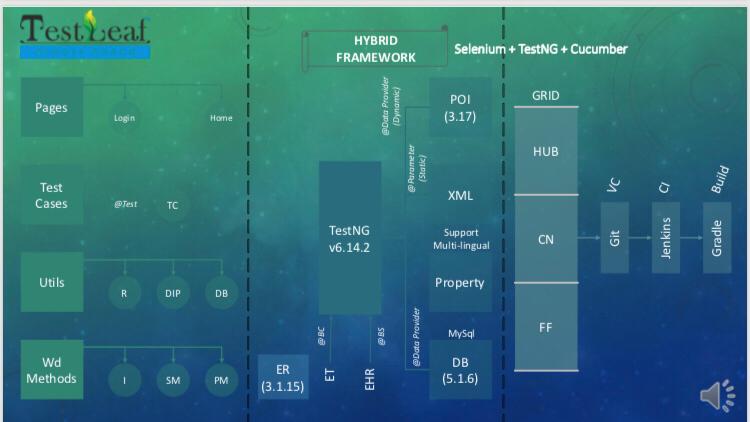
**FrameWork Architecture Pictorial representation:**

FIG:1

**Components of FrameWork:**

\*Test Data Component ->has the ability to retrieve the test data from db, excel, etc..

-It is parted into are two components

\*Dynamic data which are maintained in Excel format are handled with Apachi POI

\*Multilingual data and other common data is maintained using Property file.

\*Utility component-Utility folder holds the data like file readers, Webservices, dates..

Data Library-> provides data integration and helps to read the data from the excel file Reporter-> Extent Reporter is used to build the html report along with snapshots captured at each steps of execution.

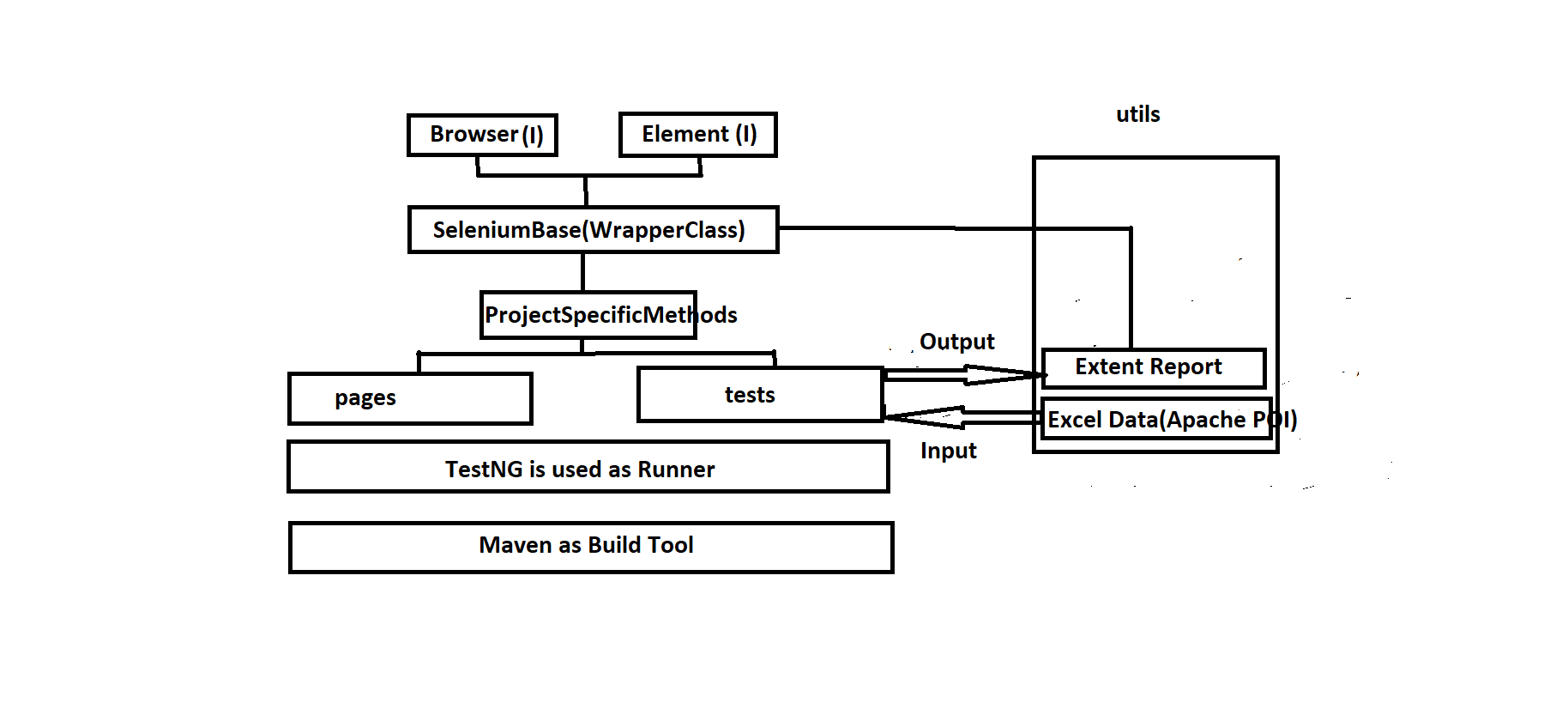
\*Interfaces->contains the basic design of the framework

\*Wrapper methods -> maintains the methods which implements the interfaces to manage the web element Interactions.

\*Pages-> shows the navigation of the application

\*Testcase -> starting point of the execution of each testcase which uses TestNg annotation

**Working interpretation of FrameWork:**

Fig:2

Framework is built using the build tool Maven.