### 

### 

### 

### 

### 

### 

### **Selenium Automation Framework**

### **README**

### 

### 

### 

### 

### 

### 

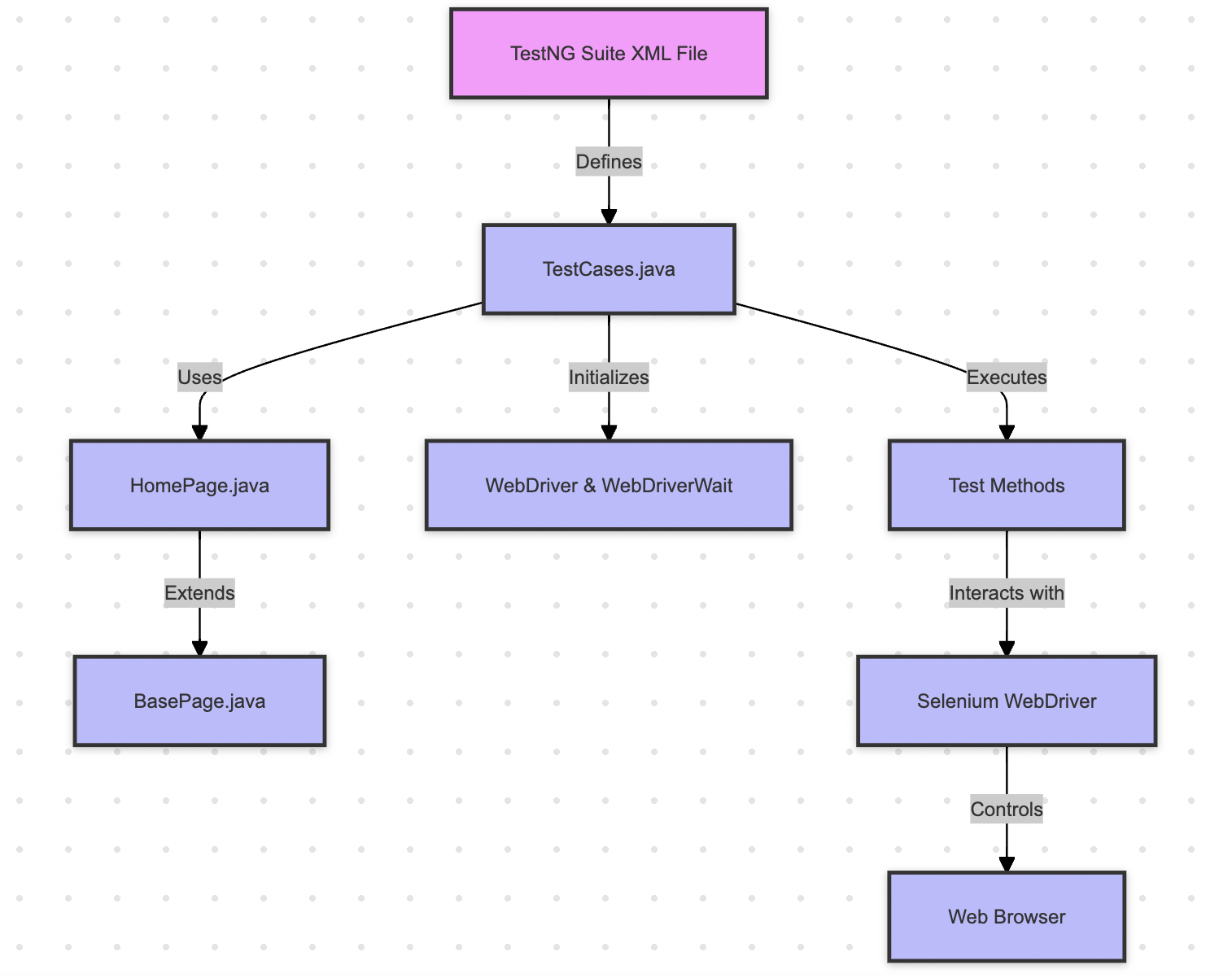
### 

### 

### **1. Deliverable Information**

### **Overview**

This Selenium Automation Framework is designed to test web applications using Selenium WebDriver, Java, and TestNG. It implements the Page Object Model (POM) to maintain a clean and manageable codebase.



### **Deliverables**

* **Page Object Model Implementation**: Classes for managing page elements and interactions.
* **Test Cases**: Predefined test cases to validate 6 testcases
* **Base Page Class**: Provides common functionality for all page classes.
* **TestNG Configuration**: XML file to configure and run the test suite.

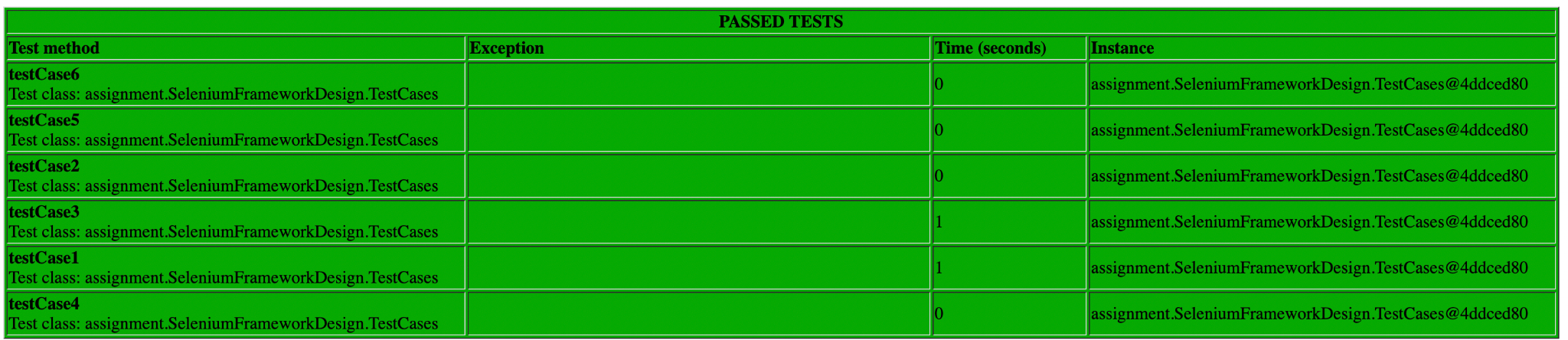
## **2. Report**

### **Test Results**

The test results can be reviewed in the TestNG reports generated after running the test suite. TestNG provides detailed HTML and XML reports, including test execution status, passed/failed tests, and error messages if any.

### **Reporting Details**

Below reports are generated in the test-output directory within the project root.



## 

## **3. How to Run It**

### **Prerequisites**

1. **Java**: Ensure Java is installed and properly configured on your system. (JDK 8 or above is recommended)
2. **Maven**: The project uses Maven for dependency management. Ensure Maven is installed.
3. **WebDriver**: Download the appropriate WebDriver executable for your browser (e.g., ChromeDriver for Chrome) and ensure it is accessible in your system's PATH.

### **Setup**

1. **Clone the Repository**  
   git clone <https://github.com/priyachaney/SeleniumFrameworkDesign.git>
2. **Configure WebDriver**

Download the WebDriver executable (e.g., ChromeDriver).

Set the path to the WebDriver executable  
System.setProperty("webdriver.chrome.driver","path/to/chromedriver");

1. **Install Dependencies**

Navigate to the project root and install dependencies using Maven  
 mvn install

1. **Running the Tests**

**TestNG XML Configuration**

* + Run the tests through your IDE by configuring TestNG to use the testng.xml file located in the src/test/java/com/seleniumframework/ directory.

## **4. Assumptions**

* The framework uses @BeforeClass to open the browser and load the URL before any test methods run, and @AfterClass to close the browser afterward.
* If needed, this can be adjusted to @BeforeMethod and @AfterMethod to open and close the browser for each test.

### **5. Improvement Areas**

* **Reporting**: Integrate Extent Reports for detailed HTML reports, offering better insights into test execution.
* **Data-Driven Testing**: Implement parameterization to run tests with data from external sources like Excel, enhancing test coverage.

### **Notes**

* **Troubleshooting**: Verify correct path configurations and installed dependencies. Refer to TestNG and Selenium documentation for further assistance.