### Test Plan for Selenium Java Test Suite with TestNG for Remote and Local Execution

**Project Title**: Test Plan for Parallel Execution of Web Application on Sauce Labs (Windows 7/10, Chrome and Firefox)

### **1. Introduction**

This test plan describes the execution strategy for running the test suite using **Selenium WebDriver**, **TestNG**, and **Sauce Labs**. The suite is designed to run tests both locally (on a developer's machine) and remotely (on Sauce Labs for cross-browser compatibility). The tests will execute **parallel tests** on multiple configurations:

* **Browsers**: Chrome, Mozilla Firefox
* **Operating Systems**: Windows 7, Windows 10

The suite includes various user interactions, UI checks, and validation of application functionality, covering different features like name editing, spinning wheel, theme settings, language settings, etc.

### **2. Tools and Technologies**

1. **Selenium WebDriver**: For automating browser interactions and validating web application functionality.
2. **TestNG**: A test framework for managing tests, including parallel execution, test groups, and prioritization.
3. **Sauce Labs**: For running tests remotely in a cloud-based environment on different configurations.
4. **Java**: Programming language used for writing test scripts.
5. **Maven/Gradle**: To manage dependencies and build the project.
6. **WebDriverManager**: For managing browser drivers (if needed for local execution).
7. **Robot/Custom Utility Libraries**: For handling specific user inputs like pressing keys and saving files (e.g., handling screen captures).

### **3. Test Suite Overview**

The test suite includes 14 test methods that verify different aspects of the **Wheel of Names** application. The tests are organized in priority order, and the suite is configured to run **in parallel** using TestNG's parallel execution feature.

### **4. Test Configuration**

1. **Execution Environment**:
   1. **Local Execution**: For local testing (usually on the developer’s machine).
   2. **Remote Execution (Sauce Labs)**: For cloud-based testing with different browser-OS combinations.
2. **Parallel Execution**:
   1. **Browsers**: Chrome, Firefox
   2. **Operating Systems**: Windows 7, Windows 10
3. **TestNG Parallel Execution Configuration**:
   1. **Parallelism**: The suite will run **tests in parallel** on different combinations of browsers and operating systems.
   2. **Thread Count**: Configured to run 4 parallel tests (Windows 7 Chrome, Windows 7 Firefox, Windows 10 Chrome, Windows 10 Firefox).
   3. **TestNG XML Configuration**: <suite name="Parallel Execution" parallel="tests" thread-count="4">  
       <test name="Test Chrome on Windows 7">  
       <parameter name="browser" value="chrome"/>  
       <parameter name="os" value="Windows 7"/>  
       <classes>  
       <class name="tests.TestClass"/>  
       </classes>  
       </test>  
        
       <test name="Test Firefox on Windows 7">  
       <parameter name="browser" value="firefox"/>  
       <parameter name="os" value="Windows 7"/>  
       <classes>  
       <class name="tests.TestClass"/>  
       </classes>  
       </test>  
        
       <test name="Test Chrome on Windows 10">  
       <parameter name="browser" value="chrome"/>  
       <parameter name="os" value="Windows 10"/>  
       <classes>  
       <class name="tests.TestClass"/>  
       </classes>  
       </test>  
        
       <test name="Test Firefox on Windows 10">  
       <parameter name="browser" value="firefox"/>  
       <parameter name="os" value="Windows 10"/>  
       <classes>  
       <class name="tests.TestClass"/>  
       </classes>  
       </test>  
      </suite>

### **5. Test Case Details**

The suite is divided into multiple test methods, each targeting a specific feature of the application. Below are the details of each test:

1. **Test Case 1: test01\_open\_home\_page\_and\_verify**
   1. **Objective**: Verify that the home page loads correctly and that the initial names are displayed correctly.
   2. **Steps**:
      1. Open the home page.
      2. Validate the title and list of names.
   3. **Assertions**:
      1. Validate the title contains "Wheel of Names".
      2. Validate the initial names list is correct.
2. **Test Case 2: test02\_edit\_page\_title\_and\_description**
   1. **Objective**: Test the ability to edit the title and description of the page.
   2. **Assertions**:
      1. Validate the updated title and description.
3. **Test Case 3: test03\_adding\_names\_into\_pad**
   1. **Objective**: Verify that names can be added to the names pad.
   2. **Assertions**:
      1. Validate the count of names in the pad.
      2. Validate that the added names match the expected values.
4. **Test Case 4: test04\_shuffle\_the\_names\_in\_pad**
   1. **Objective**: Verify that the names in the pad can be shuffled.
   2. **Assertions**:
      1. Validate that the names are shuffled.
5. **Test Case 5: test05\_sort\_the\_names\_in\_pad**
   1. **Objective**: Verify that names can be sorted.
   2. **Assertions**:
      1. Validate that the names are sorted in ascending order.
6. **Test Case 6: test06\_basic\_spin\_of\_the\_wheel**
   1. **Objective**: Verify the spinning functionality and result.
   2. **Assertions**:
      1. Validate the pop-up message and winner.
      2. Validate the time taken for the spin.
7. **Test Case 7: test07\_customize\_spinningTime\_and\_message**
   1. **Objective**: Verify that the spinning time and message can be customized.
   2. **Assertions**:
      1. Validate the customized message and time.
8. **Test Case 8: test08\_results\_counter**
   1. **Objective**: Validate the results counter after multiple spins.
   2. **Assertions**:
      1. Validate the number of results before and after spins.
9. **Test Case 9: test09\_save\_wheel\_on\_myPC**
   1. **Objective**: Verify that the user can save the wheel.
   2. **Assertions**:
      1. Validate the confirmation message upon saving.
10. **Test Case 10: test10\_new\_wheel\_functionality**
    1. **Objective**: Test the new wheel creation functionality.
    2. **Assertions**:
       1. Validate that the new wheel is created and contains the correct names.
11. **Test Case 11: test11\_load\_my\_saved\_wheel**
    1. **Objective**: Verify that a saved wheel can be loaded correctly.
    2. **Assertions**:
       1. Validate that the saved wheel loads with the correct title and names.
12. **Test Case 12: test12\_shuffle\_and\_sort\_buttons\_disabled\_during\_spin**
    1. **Objective**: Ensure the shuffle and sort buttons are disabled during spin.
    2. **Assertions**:
       1. Validate that the buttons are disabled during spin.
13. **Test Case 13: test13\_redirection\_comunity\_link\_check**
    1. **Objective**: Test the redirection of the community link.
    2. **Assertions**:
       1. Validate that the title of the redirected page contains "FAQ".
14. **Test Case 14: test14\_set\_and\_verify\_dark\_theme**
    1. **Objective**: Verify the dark theme can be set and verified by pixel analysis.
    2. **Assertions**:
       1. Validate that the dark theme is set by detecting black pixels.
15. **Test Case 15: test15\_Languages\_setting\_and\_verification**
    1. **Objective**: Test the language settings and verification.
    2. **Assertions**:
       1. Validate the number of languages.
       2. Validate that the selected language is correctly set.

### **6. Test Data**

* **Data Inputs**:
  + Predefined names and phrases like "Ali", "Beatriz", "Charles", etc.
  + Randomized data for testing different inputs.
  + Language data for testing localization functionality.
* **Expected Outputs**:
  + Correct names and titles are displayed in the app.
  + Proper spin results and confirmation messages.
  + Correct saving and loading functionality for wheels.
  + Correct behavior when shuffle/sort is applied during or after a spin.

### **7. Execution Strategy**

* **Local Testing**: Tests can be executed locally using the developer’s machine, configured with necessary drivers and browsers.
* **Remote Testing (Sauce Labs)**: The test suite will run in parallel across various OS-browser combinations (Windows 7/10 and Chrome/Firefox) on Sauce Labs.
* **Execution Time**: The total execution time depends on the complexity of the tests and network speed. With parallel execution, all 4 combinations will run concurrently.

### **8. Reporting and Logging**

* **Logs**: Selenium logs will capture browser console logs, network errors, and other detailed logs during execution.
* **TestNG Reports**: Detailed HTML reports will be generated by TestNG to summarize the results of each test execution, including test status (Pass/Fail) and error logs.

### **9. Conclusion**

This test plan ensures that the test suite will be executed successfully both locally and remotely (on Sauce Labs) with full coverage of the application features. The parallel execution across multiple OS/browser configurations provides comprehensive testing of the **Wheel of Names** application in various environments.