

Final Exam TAE Academy

Create a hybrid (web / mobile) automation framework following the architectural and clean coding best practices. The framework should include the following Test Cases regarding web and mobile:

Web Module:

Based on the ESPN webpage (use the **following link**, do not open a region-based ESPN web page):

Web Browser URL: https://www.espnqa.com/?src=com&_adblock=true&espn=cloud

Create a single test for the following steps:

1. Go to https://www.espnqa.com/?src=com&_adblock=true&espn=cloud
2. Click 'Log in' element - Validate a modal is **present and contains**:
 - a. 'ESPN' Logo
 - b. 'Log In' button
 - c. 'Sign Up' button
3. Click '**Sign Up**' button and validate:
 - a. 'Sign Up' title is present
 - b. 'First Name' input is present
 - c. 'Last name' input is present
 - d. 'Email' input is present
 - e. 'Password' input is present
 - f. 'Sign Up' button
 - g. 'X' close button
4. Enter valid information in the form and click '**Sign Up**'
5. Once logged in, go to '**Watch**' and validate that at least one carousel is present:
 - a. Each card in the carousel has a title and a small description about streaming source
6. Click on the second card in the first carousel and validate
 - a. '**X**' button to close is present
7. Click '**X** button' to close
8. Go back to the landing page
9. Perform mouse hover the '**User Icon**' element located at the top right corner of the view:
 - a. Validate the element '**Nav text**' has the right information previously entered in the sign up modal: '**Welcome {{username}}!**'

10. Click **'Log Out'** element and validate:

- a. The user has logged out successfully i.e. **Validate the element 'Nav text' has text: 'Welcome!' without user name specified**

Mobile Module:

Based on the Disneyland mobile application, create automated test cases for the following scenarios.

1. Category List Test

Preconditions: none

#	Action	Expected Result	Notes
1	Tap Map Button available in Dashboard Page	A Map view should be shown	Map Button Option is available in the bottom bar, the second item from left to right
2	Tap Category List	Different categories are listed to select	(Attractions is default option)
3	Check if the Hotels Option is available in the list	Hotels Option should be available	

2. Privacy & Legal

Preconditions: none

#	Action	Expected Result	Notes
1	Tap on Menu Button available in Dashboard Page	A Menu button view should be shown	Menu Button Option is available in the bottom bar, the five item from left to right
2	Scroll down until bottom of the screen	Different categories are listed to select:	

		<ul style="list-style-type: none"> -My Profile -Property Rules -Link to Account -Help -Privacy & Legal 	
3	Tap on Privacy & legal	Privacy & Legal screen is displayed	
4	User can see a list of options	<p>The screen have 6 option:</p> <ul style="list-style-type: none"> -Privacy Policy -Terms of use -Supplemental Terms and Conditions -Legal Notices -Property Rules -Electronic Communication Disclosure 	

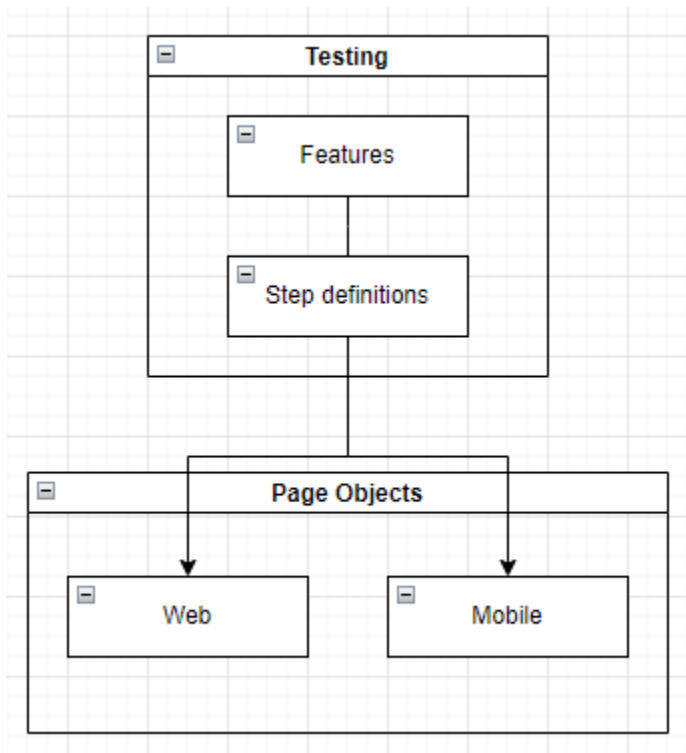
3. Add Plans option

Preconditions: None

#	Action	Expected Result	Notes
1	Tap Add Plans Button available in Dashboard Page	Add Plans Option List is shown to the user	Add Plans Button Option is available in the bottom bar, the third item from left to right
2	Check if the Reserve Dining Option is in the list	Reserve Dining Option should be enabled	

Rules to follow:

1. Be sure to use **Page Object Pattern**, **Assertions** (you decide how to validate each of the flows), and **Parameters**.
2. Avoid using implicit waits
3. Use a **BDD** approach for the implementation (Cucumber)
4. There should be different feature files for web and mobile modules
5. Appropriate tagging should be done in each scenario (Tags for unique identifiers, tags to identify if it is web or mobile, tags associated with features, etc).
6. The architecture of the framework should be defined by yourself following the best practices and design patterns. The following diagram may be used as a starting point:



7. Create a new github repository to upload the project and use git best practices on it: Multiple branches, multiple commits, gitignore.