

SDET coding challenge

PROJECT DETAILS

- Programming Language: Java
- Framework: Page Object Model using Selenium and Cucumber
- IDE Used: Eclipse
- External Libraries: Selenium, Cucumber, JUnit, extent reports, Log4j
(External libraries are defined in the maven pom.xml file)

FEATURES

- Test code is runnable and tested on Windows, Mac OS X and Linux.
- The automation can be via run via the UI within a browser.
- A reports will be generated at the end to the *target/cucumber-reports* directory, test-output directory, json_optput directory and junit-xml directory in the project.
- Test can be run from the command line.
- The host can be configured via the configuration file *config.properties*.
- Test can be configured to run on various browsers as configured using the file *config.properties*.
- Test service credentials and all other settings can be configured using the file *config.properties*.
- Test contains mobile browser tests as well
- Headless browser functionality has been implemented ChromeHeadless browser
- Screenshots will be captured for test failures using *WebDriverEventListener*.
- Project is based on maven.
- Git is used for source code management.

HOW TO RUN

1. Open command prompt / terminal.
2. Navigate to the test project directory.
3. run "mvn test".

TECHNICAL TRADEOFFS

Changing parameters like incidents will make it difficult to write tests to verify whether the correct number of incidents are displayed in the map.

SELECTED TEST CASES

Test cases were selected automate after considering their importance.

For Desktop Browser Verification:

- Register new user
- Sign in using username and password of existing user
- Verify map traffic view
- Verify map incident view
- Verify map parking view
- Verify map camera view
- Verify map toll view
- Verify displaying specific toll details in a popup
- Check live image from a selected camera
- Check live incident location on map

*This test suite for desktop browser can be further enhanced to cover more functionality.

Test case	Problem	Solution
Register new user.	Need to verify whether the website allows to register new users.	User profile data (first name, last name, username, password, address, etc.) has been generated using a random data generator as the website does not allows to use same username and email address to create multiple users. The generated profile is saved in the file test-data.csv. Generated random user data has been used to create the user account in the website. Created user account has been verified using the 'Account successfully created' text appear after submitting the form.
Sign in using username and password of an existing user.	Need to verify whether existing users can successfully sign in using their credentials.	Used an existing user's username and password to sign in.
Verify traffic view in map.	Need to verify whether traffic map view is displayed when traffic button is clicked.	System displays live traffic legend panel when map switched to traffic map view. The existence of the traffic legend panel is used to verify the functionality.

Verify incident view in map.	Need to verify whether incident map view displayed when incident button is clicked.	<p>System displays incident icons when map switched to incident map view.</p> <p>The existence of incident icons is used to verify the functionality.</p>
Verify parking view in map.	Need to verify whether parking map view displayed when parking button is clicked.	<p>System displays parking icons when map switched to parking map view.</p> <p>The existence of parking icons is used to verify the functionality.</p> <p>Since the number of parking icons displayed in the map depends on the map zoom level it was difficult to verify whether the expected number of parking lots are displayed in the map.</p> <p>One solution to this problem is to take the summation of numbers appear with each parking lot icon as the total no of parking lots.</p>
Verify camera view in map.	Need to verify whether camera map view displayed when camera button is clicked.	<p>System displays camera icons when map switched to camera map view.</p> <p>The existence of camera icons is used to verify the functionality.</p> <p>Since the number of camera icons displayed in the map depends on the map zoom level it was difficult to verify whether the exact number of cameras are displayed in the map.</p> <p>One solution to this problem is to take the summation of numbers appear with each camera icon as the total no of cameras.</p>
Verify toll view in map.	Need to verify whether toll map view displayed when toll button is clicked.	<p>System displays toll icons when map switched to toll map view.</p> <p>The existence of toll icons is used to verify the functionality.</p> <p>Since the number of toll icons displayed in the map depends on the map zoom level it was difficult to verify whether the exact number of tolls are displayed in the map. One solution to this problem is to take the summation of numbers appear with each toll icon as the total no of tolls.</p>

Verify the display of specific toll details in a popup.	Need to verify whether details of selected toll displayed on a popup window in map.	<p>Toll name to be used in test is configurable in configuration file.</p> <p>List of tolls are displayed in homepage info panel.</p> <p>Toll name specified in configuration file will be selected from toll list during the test.</p> <p>Toll name is verified against the toll pop up name.</p>
Check live image from a selected camera	Need to verify whether live images of selected camera displayed on a popup window in map.	<p>Camera name to be used in test is configurable in configuration file.</p> <p>List of cameras are displayed in homepage info panel.</p> <p>Camera name specified in configuration file will be selected from camera list during the test.</p> <p>Camera name is verified against the camera pop up name.</p> <p>Test also verifies whether latest image from camera displayed in popup.</p>
Check live incident location in map.	Need to verify whether details of selected incident displayed on a popup window in map.	<p>List of incidents are displayed in homepage info panel.</p> <p>First incident from the list will be selected during the test.</p> <p>Incident details are verified against the incident details displayed in home page info panel.</p>

For Mobile Browser Verification:

- Sign in using username and password of an existing user
- View incident location
- View live camera image

*This test suite for mobile browser can be further enhanced to cover more functionality.

Test case	Problem	Solution
Sign in using username and password of an existing user	Need to verify whether existing users can successfully sign in using their username and password.	Used an existing user's username and password to sign in.

View incident details on map	Need to verify whether details of selected incident displayed on incident page.	<p>List of incidents are displayed in incidents page.</p> <p>First incident from the list will be selected during the test.</p> <p>Incident details are verified against the incident details displayed in incidents list.</p>
View live camera image	Need to verify whether live images of selected camera displayed on camera page.	<p>Camera name to be used in test is configurable in configuration file.</p> <p>List of cameras are displayed in cameras page.</p> <p>Camera name specified in configuration file will be selected from camera list during the test.</p> <p>Camera name is verified against the camera page name.</p> <p>Test also verifies whether latest image from camera displayed.</p>