Create Quiz Test Requirements

Input Space Partitions

1. Input variables:

5 input boxes: Name, lenght (in minutes, open date, due date, number of submissions(NOS) )

2 buttons: Create questions, cancel

3 browser features: back, forward, reload

2. Definition of charasteristics of input variables:

char 1: name is empty or not

char 2: lenght and NOS are numbers or not

char 3: lenght and NOD compare to 0

char 4: open and due area dates

char 5: form button is clicked

char 6: browser feature clicked

3. Partitions:

char 1: [empty, not empty]

char 2: [number, not number]

char 3: [neg, 0, pos]

char 4: [date, not date]

char 5: [create questions, cancel]

char 6: [none, back, forward, reload]

4. Completeness and disjointness:

It is complete but not disjoint

5. Values:

char 1: ["", "a" ]

char 2: [1,a]

char 3: [-1, 0, 1]

char 4: [11/14/2018 00:00, 12]

char 5: [create questions, cancel]

char 5: [none, back, forward, reload]

6. Test requirements that satifies Base Choice coverage:

Base tr:

tr 1: ["a", 1, 1, 11/14/2018 00:00, create questions, none]

Base coverage:

tr 2: ["a", 1, 1, 11/14/2018 00:00, cancel, none]

tr 3: ["a", 1, 1, 12, create questions, none]

tr 4: ["a", 1, 0, 11/14/2018 00:00, create questions, none]

tr 5: ["a", 1, -1, 11/14/2018 00:00, create questions, none]

tr 6: ["a", a, 1, 11/14/2018 00:00, create questions, none]

tr 7: ["", 1, 1, 11/14/2018 00:00, create questions, none]

tr 8: ["a", 1, 1, 11/14/2018 00:00, create questions, back]

tr 9: ["a", 1, 1, 11/14/2018 00:00, create questions, forward]

tr 10: ["a", 1, 1, 11/14/2018 00:00, create questions, reload]

Infeasible tests:

tr 4

tr 5

tr 6

Fix infeasible test:

tr 4: ["a", 0, 0, 11/14/2018 00:00, create questions]

tr 5: ["a", -1, -1, 11/14/2018 00:00, create questions]

tr 6: ["a", a, a, 11/14/2018 00:00, create questions]

7. Test set:

tc 1: a, 1, 11/14/2018 00:00, 11/14/2018 00:00, 1, create questions, none (base)

Expected output: next page (create questions)

tc 2: a, 1, 11/14/2018 00:00, 11/14/2018 00:00, 1, cancel, none

Expected output: stay on current page (create quiz)

tc 3: a, 1, 12, 12, 1, create questions

Expected output: error message ("enter valid date")

tc 4: a, 0, 11/14/2018 00:00, 11/14/2018 00:00, 0, create questions

Expected output: unknown, perhaps invalid input message

tc 5: a, -1, 11/14/2018 00:00, 11/14/2018 00:00, -1, create questions

Expected output: error message ("enter valid number")

tc 6: a, a, 11/14/2018 00:00, 11/14/2018 00:00, a, create questions

Expected output: error message ("enter valid number")

tc 7: "", a, 11/14/2018 00:00, 11/14/2018 00:00, a, create questions

Expected output: error message ("enter valid number")

tc 8: a, 1, 11/14/2018 00:00, 11/14/2018 00:00, 1, create questions, back

Expected output: go from create question back to create quiz page, and quiz values are

still displayed correctly

tc 9: a, 1, 11/14/2018 00:00, 11/14/2018 00:00, 1, create questions, forward

Expected output: stay in create question page and quiz name doesn't disappear

tc 10: a, 1, 11/14/2018 00:00, 11/14/2018 00:00, 1, create questions, reload

Expected output: stay in create question page and correct quiz name displays

8. Additional tests

An additional test was created to make sure that the correct page shows up with the given url.

**Test Results:**

The base case and test case 2 pass. In other words if the input is valid then the page displays expected behavior.

Conversely, test cases 3, 4, 5, 6, and 7 fail because the page does not meet the requirements for invalid input (see fig. CQ1). In other words, if the input is invalid the page doesn't display expected behavior. Developers must be notified to update page behavior to pass tests.

To ensure tests are working properly, tests were changed temporally to accept actual output as expected result. After this change all tests passed (see fig. CQ2), so test harness is working properly, and problems lie on the page's code.

Fig. CQ1: Some tests failing because page doesn't meet requirements

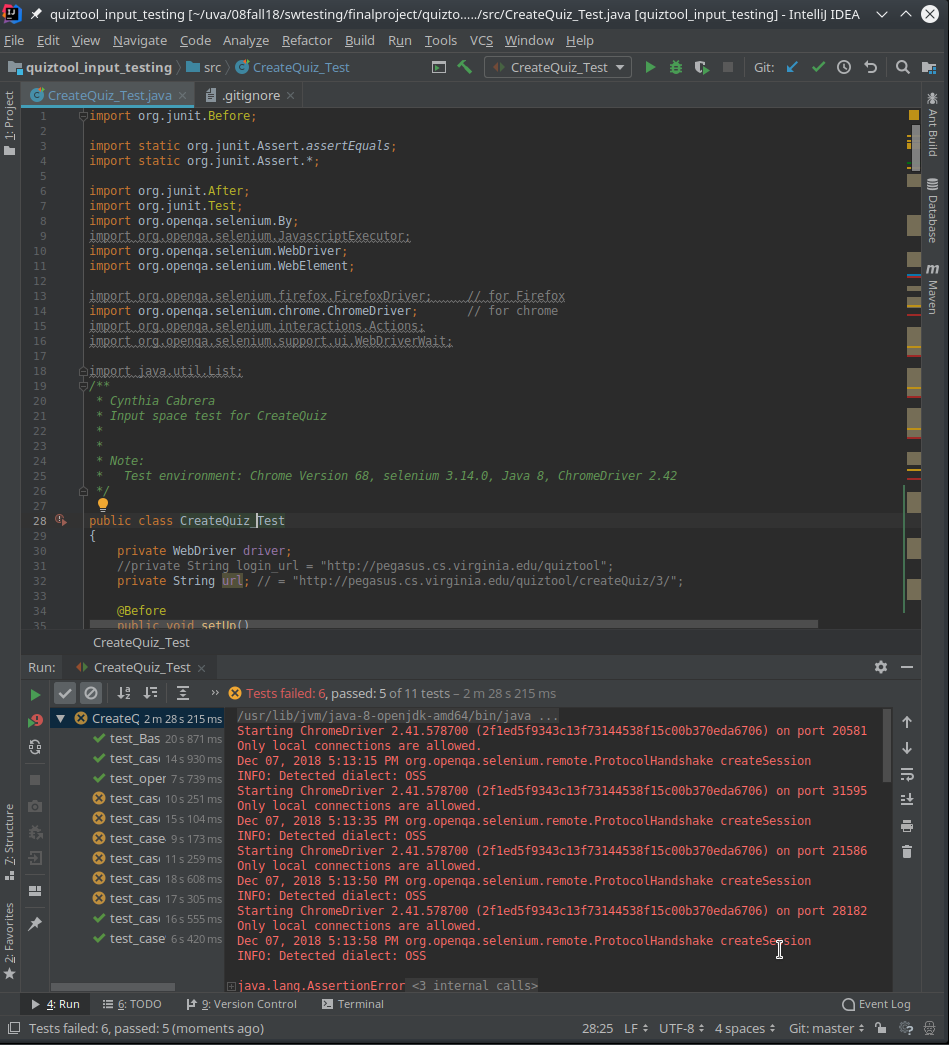
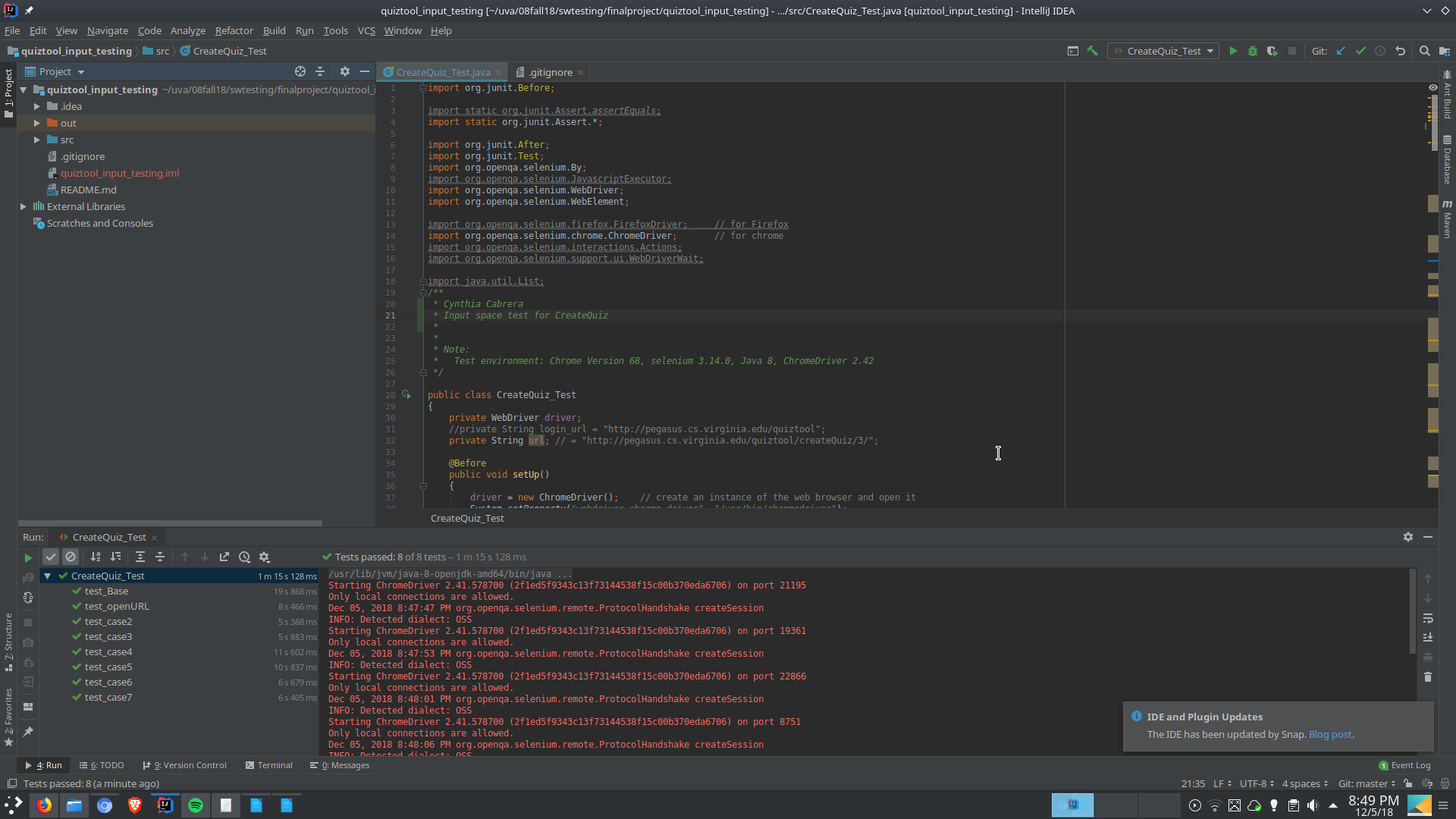


Fig. CQ2: All test passing to ensure test harness is working properly



Automated Tests on Selenium

import org.junit.Before;

import static org.junit.Assert.*assertEquals*;

import static org.junit.Assert.\*;

import org.junit.After;

import org.junit.Test;

import org.openqa.selenium.By;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.firefox.FirefoxDriver; // for Firefox

import org.openqa.selenium.chrome.ChromeDriver; // for chrome

import org.openqa.selenium.interactions.Actions;

import org.openqa.selenium.support.ui.WebDriverWait;

import java.util.List;

*/\*\**

*\* Cynthia Cabrera*

*\* Input space test for CreateQuiz*

*\**

*\**

*\* Note:*

*\* Test environment: Chrome Version 68, selenium 3.14.0, Java 8, ChromeDriver 2.42*

*\*/*

public class CreateQuiz\_Test

{

private WebDriver driver;

//private String login\_url = "http://pegasus.cs.virginia.edu/quiztool";

private String url; // = "http://pegasus.cs.virginia.edu/quiztool/createQuiz/3/";

@Before

public void setUp()

{

driver = new ChromeDriver(); // create an instance of the web browser and open it

System.*setProperty*("webdriver.chrome.driver", "/usr/bin/chromedriver");

//Login into quiztool app

url = "http://pegasus.cs.virginia.edu/quiztool/login";

driver.get(url); // open the given url

driver.findElement(By.*name*("username")).sendKeys("professorM");

driver.findElement(By.*name*("password")).sendKeys("swtesting1");

WebElement element = driver.findElement(By.*xpath*("(/html/body/div/div/form/div/button[@type='submit'])[1]"));

element.click();

// Go to createQuiz

url = "http://pegasus.cs.virginia.edu/quiztool/createQuiz/1/";

driver.get(url); // open the given url

}

@After

public void teardown()

{

driver.quit(); // close the browser

}

@Test

public void test\_openURL()

{

*assertTrue*(driver.getPageSource().contains("Create a Quiz")); // check if we are on the right page

}

@Test

public void test\_Base() // test case #1

{

// Fill input fields

driver.findElement(By.*name*("quiz\_name")).clear();

driver.findElement(By.*name*("quiz\_name")).sendKeys("a");

driver.findElement(By.*name*("duration")).clear();

driver.findElement(By.*name*("duration")).sendKeys("1");

driver.findElement(By.*name*("open\_date")).clear();

driver.findElement(By.*name*("open\_date")).sendKeys("11/14/2018 00:00");

driver.findElement(By.*name*("due\_date")).clear();

driver.findElement(By.*name*("due\_date")).sendKeys("11/14/2018 00:00");

driver.findElement(By.*cssSelector*("body")).click(); // Click on the background to deselect the datepicker

driver.findElement(By.*name*("number\_of\_allowed\_submissions")).clear();

driver.findElement(By.*name*("number\_of\_allowed\_submissions")).sendKeys("1");

driver.findElement(By.*cssSelector*("body")).click(); // Click on the background to deselect the input field

// Submit

WebElement element = driver.findElement(By.*xpath*("(/html/body/div[1]/div/div[2]/form/div/button[@type='submit'])[1]"));

element.click();

// Check result

*assertTrue*(driver.getPageSource().contains("Question text"));

}

@Test

public void test\_case2()

{

// Fill input fields

driver.findElement(By.*name*("quiz\_name")).clear();

driver.findElement(By.*name*("quiz\_name")).sendKeys("a");

driver.findElement(By.*name*("duration")).clear();

driver.findElement(By.*name*("duration")).sendKeys("1");

driver.findElement(By.*name*("open\_date")).clear();

driver.findElement(By.*name*("open\_date")).sendKeys("11/14/2018 00:00");

driver.findElement(By.*name*("due\_date")).clear();

driver.findElement(By.*name*("due\_date")).sendKeys("11/14/2018 00:00");

driver.findElement(By.*cssSelector*("body")).click(); // Click on the background to deselect the datepicker

driver.findElement(By.*name*("number\_of\_allowed\_submissions")).clear();

driver.findElement(By.*name*("number\_of\_allowed\_submissions")).sendKeys("1");

driver.findElement(By.*cssSelector*("body")).click(); // Click on the background to deselect the input field

driver.findElement(By.*linkText*("Cancel")).click();

//element.click();

//try { Thread.sleep(1000); } catch (InterruptedException e) { }// *TODO Auto*

*assertTrue*(driver.getPageSource().contains("CS 4501"));

}

@Test

public void test\_case3()

{

// Fill input fields

driver.findElement(By.*name*("quiz\_name")).clear();

driver.findElement(By.*name*("quiz\_name")).sendKeys("a");

driver.findElement(By.*name*("duration")).clear();

driver.findElement(By.*name*("duration")).sendKeys("1");

driver.findElement(By.*name*("open\_date")).clear();

driver.findElement(By.*name*("open\_date")).sendKeys("12");

driver.findElement(By.*name*("due\_date")).clear();

driver.findElement(By.*name*("due\_date")).sendKeys("12");

driver.findElement(By.*cssSelector*("body")).click(); // Click on the background to deselect the datepicker

driver.findElement(By.*name*("number\_of\_allowed\_submissions")).clear();

driver.findElement(By.*name*("number\_of\_allowed\_submissions")).sendKeys("1");

driver.findElement(By.*cssSelector*("body")).click(); // Click on the background to deselect the input field

// Click cancel

try { Thread.*sleep*(1000); } catch (InterruptedException e) { }// *TODO Auto*

WebElement element = driver.findElement(By.*xpath*("(/html/body/div[1]/div/div[2]/form/div/button[@type='submit'])[1]"));

element.click();

// Check result

// Page should give error message "Invalid date"

*assertTrue*(driver.getPageSource().contains("Invalid date"));

// But instead breaks down

//assertTrue(driver.getPageSource().contains("TypeError"));

}

@Test

public void test\_case4()

{

// Fill input fields

driver.findElement(By.*name*("quiz\_name")).clear();

driver.findElement(By.*name*("quiz\_name")).sendKeys("a");

driver.findElement(By.*name*("duration")).clear();

driver.findElement(By.*name*("duration")).sendKeys("1");

driver.findElement(By.*name*("open\_date")).clear();

driver.findElement(By.*name*("open\_date")).sendKeys("11/14/2018 00:00");

driver.findElement(By.*name*("due\_date")).clear();

driver.findElement(By.*name*("due\_date")).sendKeys("11/14/2018 00:00");

driver.findElement(By.*cssSelector*("body")).click(); // Click on the background to deselect the datepicker

driver.findElement(By.*name*("number\_of\_allowed\_submissions")).clear();

driver.findElement(By.*name*("number\_of\_allowed\_submissions")).sendKeys("0");

driver.findElement(By.*cssSelector*("body")).click(); // Click on the background to deselect the input field

// Submit

try { Thread.*sleep*(1000); } catch (InterruptedException e) { }// *TODO Auto*

WebElement element = driver.findElement(By.*xpath*("(/html/body/div[1]/div/div[2]/form/div/button[@type='submit'])[1]"));

element.click();

// Check result

// Page should give error message "Invalid duration"

*assertTrue*(driver.getPageSource().contains("Invalid duration"));

// But instead it accepts value

//assertTrue(driver.getPageSource().contains("Question text"));

}

@Test

public void test\_case5()

{

// Fill input fields

driver.findElement(By.*name*("quiz\_name")).clear();

driver.findElement(By.*name*("quiz\_name")).sendKeys("a");

driver.findElement(By.*name*("duration")).clear();

driver.findElement(By.*name*("duration")).sendKeys("1");

driver.findElement(By.*name*("open\_date")).clear();

driver.findElement(By.*name*("open\_date")).sendKeys("11/14/2018 00:00");

driver.findElement(By.*name*("due\_date")).clear();

driver.findElement(By.*name*("due\_date")).sendKeys("11/14/2018 00:00");

driver.findElement(By.*cssSelector*("body")).click(); // Click on the background to deselect the datepicker

driver.findElement(By.*name*("number\_of\_allowed\_submissions")).clear();

driver.findElement(By.*name*("number\_of\_allowed\_submissions")).sendKeys("-1");

driver.findElement(By.*cssSelector*("body")).click(); // Click on the background to deselect the input field

// Submit

try { Thread.*sleep*(1000); } catch (InterruptedException e) { }// *TODO Auto*

WebElement element = driver.findElement(By.*xpath*("(/html/body/div[1]/div/div[2]/form/div/button[@type='submit'])[1]"));

element.click();

// Check result

// Page should give error message "Invalid duration"

*assertTrue*(driver.getPageSource().contains("Invalid duration"));

// But instead it accepts value

//assertTrue(driver.getPageSource().contains("Question text"));

}

@Test // (expected = NullPointerException.class)

public void test\_case6()

{

// Fill input fields

driver.findElement(By.*name*("quiz\_name")).clear();

driver.findElement(By.*name*("quiz\_name")).sendKeys("a");

driver.findElement(By.*name*("duration")).clear();

driver.findElement(By.*name*("duration")).sendKeys("1");

driver.findElement(By.*name*("open\_date")).clear();

driver.findElement(By.*name*("open\_date")).sendKeys("11/14/2018 00:00");

driver.findElement(By.*name*("due\_date")).clear();

driver.findElement(By.*name*("due\_date")).sendKeys("11/14/2018 00:00");

driver.findElement(By.*cssSelector*("body")).click(); // Click on the background to deselect the datepicker

driver.findElement(By.*name*("number\_of\_allowed\_submissions")).clear();

driver.findElement(By.*name*("number\_of\_allowed\_submissions")).sendKeys("a");

driver.findElement(By.*cssSelector*("body")).click(); // Click on the background to deselect the input field

// Submit

try { Thread.*sleep*(1000); } catch (InterruptedException e) { }// *TODO Auto*

WebElement element = driver.findElement(By.*xpath*("(/html/body/div[1]/div/div[2]/form/div/button[@type='submit'])[1]"));

element.click();

// Check result

// Page should give error message "Invalid duration"

*assertTrue*(driver.getPageSource().contains("Invalid duration"));

// But instead it doesn't give error message but it won't submit which is a good behavior

//assertTrue(driver.getPageSource().contains("Create a Quiz"));

}

@Test // (expected = NullPointerException.class)

public void test\_case7()

{

// Fill input fields

driver.findElement(By.*name*("quiz\_name")).clear();

driver.findElement(By.*name*("quiz\_name")).sendKeys("");

driver.findElement(By.*name*("duration")).clear();

driver.findElement(By.*name*("duration")).sendKeys("1");

driver.findElement(By.*name*("open\_date")).clear();

driver.findElement(By.*name*("open\_date")).sendKeys("11/14/2018 00:00");

driver.findElement(By.*name*("due\_date")).clear();

driver.findElement(By.*name*("due\_date")).sendKeys("11/14/2018 00:00");

driver.findElement(By.*cssSelector*("body")).click(); // Click on the background to deselect the datepicker

driver.findElement(By.*name*("number\_of\_allowed\_submissions")).clear();

driver.findElement(By.*name*("number\_of\_allowed\_submissions")).sendKeys("1");

driver.findElement(By.*cssSelector*("body")).click(); // Click on the background to deselect the input field

// Submit

try { Thread.*sleep*(1000); } catch (InterruptedException e) { }// *TODO Auto*

WebElement element = driver.findElement(By.*xpath*("(/html/body/div[1]/div/div[2]/form/div/button[@type='submit'])[1]"));

element.click();

// Check result

// Page should give error message "Please fill out this field."

*assertTrue*(driver.getPageSource().contains("Please fill out this field"));

// But instead it accepts value

//assertTrue(driver.getPageSource().contains("Create a Quiz"));

}

}