Assignment B – Cucumber in Agile Story Test Automation

Ismail Faruk - 260663521 - ismail.faruk@mail.mcgill.ca

Mustafa Khawaja - 260630187 - mustafa.khawaja@mail.mcgill.ca

Summary of files delivered

ECSE 428 Assignment B Report.pdf – Contains most of the deliverables, including story statement, description of environment used, source code, video of test and details of running tests and creating new scenarios. Also contains

gmail.feature – Cucumber feature file including a gherkin script using scenario outline - 428AssignmentB\Features\ gmail.feature

StepDefinitions.java – Contains the step definitions, methods of implementing the feature file - 428AssignmentB\src\test\java\com\mustafakha\cucumber\StepDefinitions.java

Summary of Deliverables

Story Statement

As a user with an email address
I would like to attach an image to my email
so I can send it to a recipient

Statement of acceptance tests in Gherkin

Feature: Sending emails with attachments through Gmail using Google Chrome

NormalFlow: Sending an email with an image to someone

Given I am on the Gmail homepage using Google Chrome

And I log in with valid credentials

When I compose a new email

And attach an image

Then I will send the email successfully

Alternate Flow

Scenario: Sending an email to myself with a file url

Given I am on the Gmail homepage using Google Chrome

And I log in with valid credentials

When I compose a new email

And send an attachment as a url

Then I will send the email successfully with a url

Alternate Flow

Scenario: Sending an email with a video to someone

Given I am on the Gmail homepage using Google Chrome

And I log in with valid credentials

When I compose a new email

And attach a video

Then I will send the email successfully

Alternate Flow

Scenario: Sending an email with a video and image to someone

Given I am on the Gmail homepage using Google Chrome

And I log in with valid credentials

When I compose a new email

And attach a video and image

Then I will send the email successfully

Error Flow

Scenario: Sending an email with an attachment that does not exist

Given I am on the Gmail homepage using Google Chrome

And I log in with valid credentials

When I compose a new email

And attach a file that does not exist

Then I will not send the email

Environment Description Summary

General Setup – The project was created using Java 8 and Maven 4, AutoIT scripts with Eclipse Java IDE. Selenium v3.1 was used for web automation. The tests were run with Eclipse IDE. We also used Windows Command Prompt and ran a maven test.

Cucumber

The dependencies for Cucumber v4.2, were installed using Maven. All cucumber tests run locally on the machine through Eclipse and command prompt. The feature file: **gmail.feature**, contains all the scenarios. Using command prompt, we can test by running the feature file locally and ensuring that the configurations are set accordingly. The following are the descriptions of the maven cucumber dependencies, taken from the **pom.xml** file:

```xml
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi=
"http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation=
"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

  <modelVersion>4.0.0</modelVersion>
  <groupId>428AssignmentB</groupId>
  <artifactId>428AssignmentB</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <dependencies>
  <!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java -
->
    <dependency>
        <groupId>org.seleniumhq.selenium</groupId>
        <artifactId>selenium-java</artifactId>
        <version>3.141.59</version>
    </dependency>

  <!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-java -->
<dependency>
    <groupId>io.cucumber</groupId>
    <artifactId>cucumber-java</artifactId>
    <version>4.2.0</version>
```

```xml
</dependency>

  <!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-junit -->
<dependency>
    <groupId>io.cucumber</groupId>
    <artifactId>cucumber-junit</artifactId>
    <version>4.2.0</version>
    <scope>test</scope>
</dependency>
<!-- https://mvnrepository.com/artifact/junit/junit -->
<dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.12</version>
    <scope>test</scope>
</dependency>
<dependency>
  <groupId>com.google.guava</groupId>
  <artifactId>guava</artifactId>
  <version>27.1-jre</version>
  </dependency>

  </dependencies>
</project>
```

Cucumber Gherkin Scripts

```gherkin
@tag
Feature: Sending emails with attachments through Gmail using Google Chrome

  @NormalFlow
  Scenario: Sending an email with an image to someone
    Given I am on the Gmail homepage using Google Chrome
    And I log in with valid credentials
    When I compose a new email
    And attach an image
    Then I will send the email successfully


  @AlternateFlow1-url_file
  Scenario: Sending an email to myself with a file url
    Given I am on the Gmail homepage using Google Chrome
    And I log in with valid credentials
    When I compose a new email
    And send an attachment as a url
```

```
    Then I will send the email successfully with a url

@AlternateFlow2-email_with_video
Scenario: Sending an email with a video to someone
    Given I am on the Gmail homepage using Google Chrome
    And I log in with valid credentials
    When I compose a new email
    And attach a video
    Then I will send the email successfully

@AlternateFlow3-multiple_attachments
Scenario: Sending an email with a video and image to someone
    Given I am on the Gmail homepage using Google Chrome
    And I log in with valid credentials
    When I compose a new email
    And attach a video and image
    Then I will send the email successfully


@ErrorFlow-file_does_not_exist
Scenario: Sending an email with an attachment that does not exist
    Given I am on the Gmail homepage using Google Chrome
    And I log in with valid credentials
    When I compose a new email
    And attach a file that does not exist
    Then I will not send the email
```

Source Code of step definitions

Step definition source code provided in **StepDefinitions.java** file

Feature with gherkin script provided in **gmail.feature** file

Instructions to install test environment and run tests

1. Clone the source code from the GitHub repository using the following link:
   https://github.com/ismailfaruk/ECSE428--Software-Engineering-Practice-AssignmentB

2. The repository contains the Chrome Web Driver. Unzip the file to extract the web driver.

3. Download Java SE Development Kit 8u202 from the link:
   https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

4. The repository also contains eclipse installer. Use it to download Eclipse Java Photon IDE.

5. Using the Eclipse IDE, build and compile the project and run gmail.feature

Video of Automated test

The video was uploaded on Google Drive and can be viewed using the following link:

Detailed instructions as to how to run all tests

Using Eclipse:

1. Open the project in eclipse
2. Build and run from the IDE

Using Windows Command Prompt

1. Open cmd.exe
2. Change directory to the folder of the project
3. Run "**mvn test"** to see if project build is successful

Detailed guidelines explaining how to add new gherkin scripts and associated step definitions

To add a new scenario to feature file:

1. To add new Gherkin scripts, start with a scenario
2. Follow the Gherkin rules
3. Describe an initial context, an event and an expected outcome

To implement them in the step definition file:

1. Run Windows Command Prompt
2. Go the directory of the project
3. Run "**mvn test**"
4. It will suggest a skeleton of how to implement the step definition for the new features added
5. Create new Java methods with an expression that links it to the new Gherkin script

For further information, refer to: https://docs.cucumber.io/guides/10-minute-tutorial/
For feature file: https://docs.cucumber.io/gherkin/reference/#feature
For step definitions: https://docs.cucumber.io/cucumber/step-definitions/

Description of the test environment

Cucumber is a testing framework for Behavior Driven Development, for acceptance tests and scenarios based on Gherkin scripts. Cucumber reads executable specifications written in plain text, the features, and validates the software on does what those specifications say. Cucumber uses step definitions to run each Gherkin step into runnable programming code to carry out what action should be performed by the step. Because Gherkin uses spoken languages, no new

language was required to be learned. Cucumber could be run and tested on a single IDE or using Windows Command Prompt.

A risk would be if Google Chrome is not installed or the user does not have a gmail ID, which limits the scope of the system. The current script assumes that google chrome would be used and does not check what the default browser of the system is. The current implementation also does not check if the email is correct or not.
A possible improvement would be to check what the system default browser is and run the corresponding web driver from a library of web drivers. Another improvement could be to choose the methods in the step definitions based on the domain of the email ID of the sender. This would allow more versatility. A possible solution to this could be to check if there are any email replies from the email service provider mentioning key words referring to failure to send email.

If the email GUI changed, the feature script would have to be updated based on the requirement of the new initial context and event. Following that the step definitions would have to change to reflect on the new email GUI. The navigation of the initial context and event would change. The expected outcome would remain the same. A new version of the web driver could be required if there are changes drastic enough.