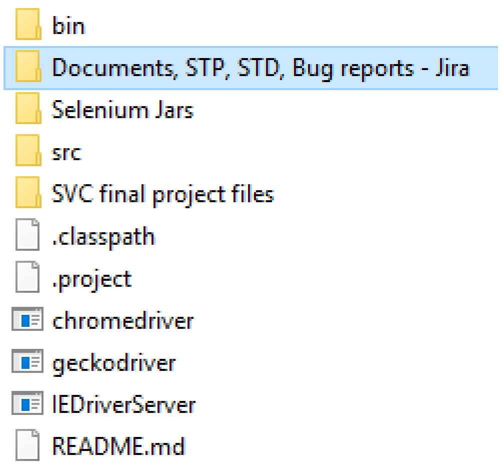
STP – Trivia final project

This STP main purpose is to map the final project of the SVCollege Automation-Course. I added the STP to guide outsiders that may look at this project, and help them understand what I created.

The main purpose of the SVCollege Automation-Course is to acquire the needed skills, to become a productive member in QA team.

System (Trivia) description - Trivia is an app that created by SVCollege, to test the SVCollege students, In this app the user is asked to enter questions and answers, Then the questions are presented to the user as a Trivia game, If the user choses the right answers the final page of the Trivia will show "Success!!", and if not, the final page will show "Failed!!", as shown in the picture:

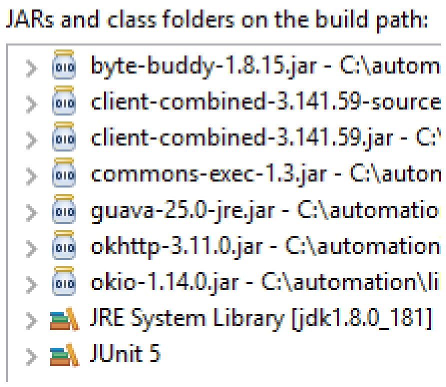
Documents – STD for each test was written as first step and its located in the folder: "Documents, STP, STD, Bug reports - Jira" – That can be found in the main project folder, as shown in the picture:



bin
Documents, STP, STD, Bug reports - Jira
Selenium Jars
src
SVC final project files
.classpath
.project
chromedriver
geckodriver
IEDriverServer
README.md

Work force – This is an individual project created by me Oren Vilderman.

Environments – In order to run the test on Eclipse, Add the JUnit 5 library to the project. As well as the included Selenium jars, As shown in the picture:



JARs and class folders on the build path:

- byte-buddy-1.8.15.jar - C:\autom
- client-combined-3.141.59-source
- client-combined-3.141.59.jar - C:\
- commons-exec-1.3.jar - C:\auton
- guava-25.0-jre.jar - C:\automatio
- okhttp-3.11.0.jar - C:\automation
- okio-1.14.0.jar - C:\automation\li
- JRE System Library [jdk1.8.0_181]
- JUnit 5

There is a .zip file in the "Selenium Jars" folder that include all the needed jars, the path to the file is "Selenium Jars\selenium-java-3.141.59.zip**".**

The .exe files on the main folder are needed drivers:
"chromedriver" is the driver for chrome browser,
"geckodriver" is the driver for Firefox browser,
"IEDriverServer" is the driver for Internet Explorer browser.

These three browsers are prerequisites, to run this project.

Tools – Jira Bug reports.
Include:

1. Subject – Short description of the bug.

2. Steps to reproduce - Each integer represent a step.

3. Bug – The numbers with decimal represent the test number, the two parts of the number have different meaning:
   1. The integer part represent the times tests was done after step in this class.
   2. The decimal part represent the number of test after the last step.

   Example: The number "25.17" represent the 18th (0-17 = 18) test that was done after some step, and the 25 represent the times tests was done after a new step in this class of tests.

4. Severity – Five categories from Very Low to Very High

5. Labels – Common labeling words for bugs.

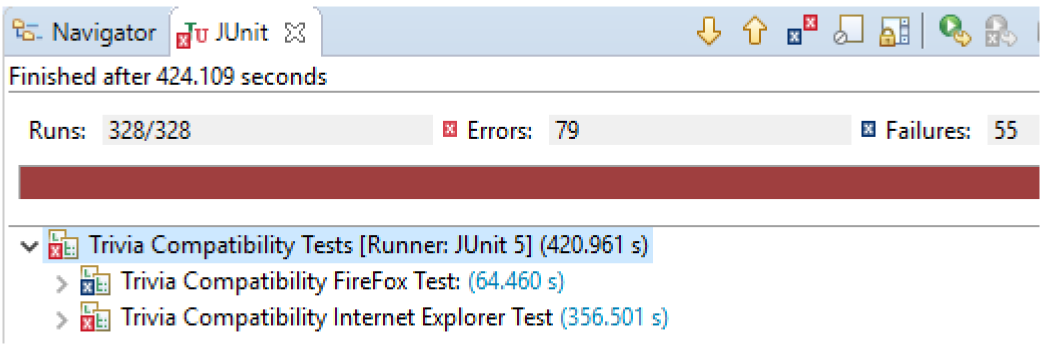
Entry and exit criteria for tests – No error or failure in any test or class will stop the other tests or steps. This project's goal is to demonstrate throwing of errors and failures at different stages and from different classes, every error, failure, and successful test is shown on the console with details.

End-to-End – The test is performed E2E meaning that internet connection is needed and that tests will include "isDisplayed" tests, to show a variety of different tests that can be performed on the Trivia app.

Tests managements – Every STD is written in a format that represent the "Test.java" file, for example in the Compatibility STD the main test class contain two different Compatibility test classes, every test class have its different: Steps, Tests, Errors and Failures.

The top line represent the "TriviaCompatibilityTest.java" class,
It show the total amounts for each of the inner, nested classes.
The "Actual Results" represent the Junit test results for the online version of the Trivia app.

Total		**Test Procedure**	**Expected Result**	**Actual Result**	**Build Num.**	**Notes**
Steps	**Tests**					
68	328	Compatibility - Browsers - (Sanity) - Trivia – Web end to end test	328/328	194/328	1	79 - Errors 55 - Failures
34	164	Compatibility - Firefox browser - (Sanity) - Trivia – Web end to end test	164/164	150/164	1	14 - Failures
34	164	Compatibility - Internet Explorer browser - (Sanity) - Trivia – Web end to end test	164/164	44/164	1	79 - Errors 41 - Failures



Finished after 424.109 seconds

Runs: 328/328 Errors: 79 Failures: 55

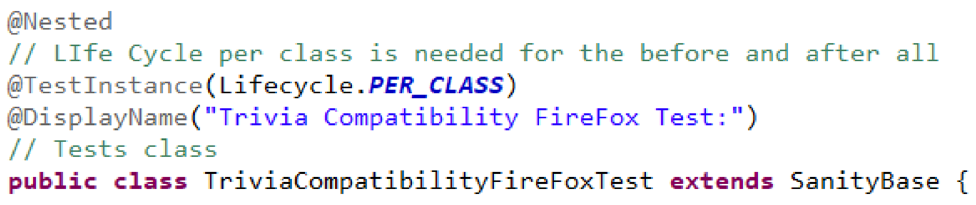
Trivia Compatibility Tests [Runner: JUnit 5] (420.961 s)
Trivia Compatibility FireFox Test: (64.460 s)
Trivia Compatibility Internet Explorer Test (356.501 s)

Test cases – The project contain six different tests.

1. Sanity	-The sanity test is a positive and minimal test that test if the Trivia app work by the plan in the SRS.
2. Functionality	- The functionality test is a positive test for all the app functions and features that are mention in the SRS.
3. Boundary Values	- The Boundary Values test is a test for the minimal and maximal values for every field in the app that receive input from the user.
4. Error Handling	- The Error Handling test is a negative test that test for scenarios that was not mention in the SRS and will cause errors in the Trivia app.
5. Integration	- The Integration test is a test for the connection with other systems - Share in Facebook option.
6. Compatibility	- The compatibility test is a test for the trivia app on other browsers - Firefox and Internet Explorer.

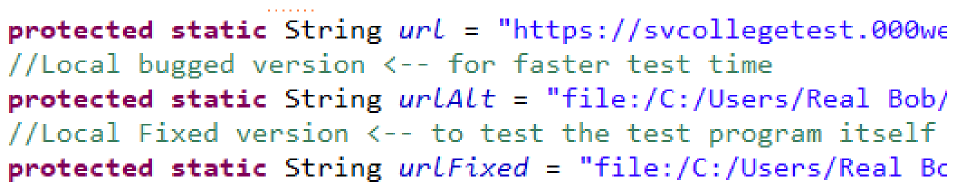
Personal note **- Technical specifications -**
The "REDME.md" file contains the information about the proper way to run the tests and in what classes and lines there are unique method I want others to see.

When I thought that parts of the code should be clarify, I added comments inside the code.

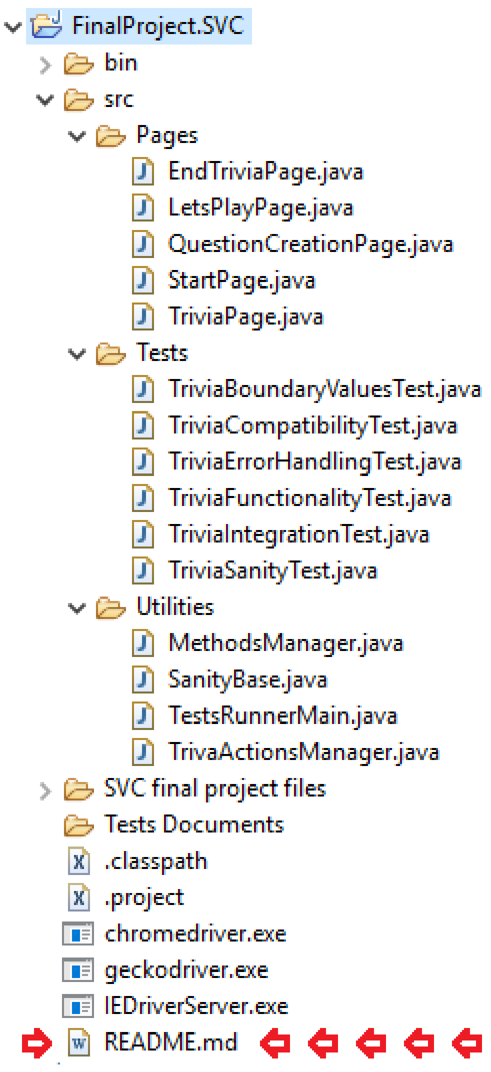


```
@Nested
// LIfe Cycle per class is needed for the before and after all
@TestInstance(Lifecycle.PER_CLASS)
@DisplayName("Trivia Compatibility FireFox Test:")
// Tests class
public class TriviaCompatibilityFireFoxTest extends SanitySBase {
```

Some parts of the code are not been used by default, the comments next to them meant to help others understand what they can be used for.



```
protected static String url = "https://svcollegetest.000we
//Local bugged version <-- for faster test time
protected static String urlAlt = "file:/C:/Users/Real Bob/
//Local Fixed version <-- to test the test program itself
protected static String urlFixed = "file:/C:/Users/Real Bc
```



```
FinalProject.SVC
  bin
  src
    Pages
      EndTriviaPage.java
      LetsPlayPage.java
      QuestionCreationPage.java
      StartPage.java
      TriviaPage.java
    Tests
      TriviaBoundaryValuesTest.java
      TriviaCompatibilityTest.java
      TriviaErrorHandlingTest.java
      TriviaFunctionalityTest.java
      TriviaIntegrationTest.java
      TriviaSanityTest.java
    Utilities
      MethodsManager.java
      SanityBase.java
      TestsRunnerMain.java
      TrivaActionsManager.java
  SVC final project files
  Tests Documents
  .classpath
  .project
  chromedriver.exe
  geckodriver.exe
  IEDriverServer.exe
  README.md
```