Malta College of Arts, Science & Technology
INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGY

Software Test Automation

B.Sc. (Hons) in Software Development

Home Assignment 2 Part 2

Instructions to Students

Read the following instructions carefully before you start the assignment. If you do not understand any of them, ask your lecturer for further explanation.

- This assignment (Part 2) has a total weight of **24%.**
- This version of the assignment is meant for **SWD students**.
- Plagiarism is strictly prohibited and will be penalised through a referral and other disciplinary procedures as per MCAST Policies, Procedures and Regulations.
- Submit through Moodle (documentation and code).

Rubric

Task	No Marks	Partial Marks	Full Marks	Marks
KU7	Task not attempted or cannot be executed due to syntax errors or other major issues.	Task attempted but incomplete, with less than 5 unique tests. 1 mark will be awarded for each accepted test.	5 unique and valid tests	
AA5	No feature files submitted or cannot be executed due to syntax errors.	Feature files contain more than five test cases. (2 marks) 1-2 marks for meaningful naming of test cases, proper indentation, and overall good quality.	Feature files have at least ten unique good quality cases that cover different valid and invalid cases.	
KU6	No acceptance test implementation.	Acceptance test implementation contain more than half of the required cases. (2 marks)	TaskRunner.main() correctly implemented. At least ten acceptance tests implemented correctly that handle all situations.	
AA2	No Documentation submitted.	Partial marks for vague, incomplete, or partially incorrect documentation. (1-6 marks)	All documentation submitted was correct and complete.	

Testing a REST API (5 marks)

For this task it is recommended to use REST Assured.

KU7 Select an endpoint from a REST API of your choice (excluding the APIs used in the demos provided with this course). It is your responsibility to choose a REST API endpoint that is not too simple in such a way that you cannot complete the task fully.
Note You might want to use APIs that require simple authentication (e.g., API key).

For the selected endpoint write at least 5 unique tests that use it in different ways to produce different outputs, such as:

- Missing authentication token
- Invalid authentication token
- Missing parameters
- Invalid parameters values
- Correct endpoint usage
- Using different request types (e.g., GET, POST, etc.)

Acceptance testing using Cucumber/Gherkin (19 marks)

Using your work for Part 1 as a starting point, add a class called **TaskRunner** with a **main()** method that allows the user to create and execute tasks by providing the list of parameters as command line arguments.

AA5 Produce the following documentation:

- Produce one or more feature files (given-when-then) to be used in acceptance testing of **TaskRunner**. At least 10 cases should be included that include valid, invalid, incomplete, and missing parameters. (5+2 marks)
  Note: Implement these tests in criterion KU6.

KU6 Meeting user expectations:

- Write 10 acceptance tests defined in criterion AA5 to make sure that **TaskRunner** can correctly handle valid and invalid user input without crashing. (5 marks)

AA2 Importance of Acceptance testing:

- Identify and describe at least one improvement that was identified when performing the acceptance testing above. This can be a bug, adding a missing feature, or implementing a new feature. (2 marks)
- Discuss some refactoring (include before and after code snippets) identified during acceptance testing that would improve your design. (2 marks)
- Explain the importance of Cucumber/Gherkin in the process of test user interactions, using your experience in this project or other work you did during the semester. (3 marks)