Assignment 01

Course Name: Software Testing and Quality Assurance

Course ID: CSE 4495

Section: A

Group Name: ARKv2.0

Group Members:

Md. Abdur Rahman (011202260)

Md. Kabirul Hossain (011202026)

Submitted on: 17th Sept, 2024

**API 1: (/student):**

|  |  |  |
| --- | --- | --- |
| **#Choices** | **Rationale** | **#Specification** |
| 1 | For this API, the number of testable choice is 1 since it does not any more testable choice. | 1 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Request Type** | **Body** | **URL** | **Assertions** | **Description** |
| GET | This request does not have a body | http://127.0.0.1:5000/student | pm.**test**("Response time is less than 100ms", **function** () {  pm.expect(pm.response.responseTime).to.be.below(100);  });  pm.**test**("Status code is 200", **function** () {  pm.response.to.have.status(200);  }); | Our goal for this test was to check whether all student info is available, and to check if the API response time is below 100ms with a status code of 200;  For this test, the test result was successfully executed |

Test case check for API 1:

**API 2: (/student/<id>):**

|  |  |  |  |
| --- | --- | --- | --- |
| **Choice 1: “**ID” | **http requests** | **Output** | **Decision** |
| 1 | http://127.0.0.1:5000/student/1 | "courses\_passed" | valid ID |
| 0 > | http://127.0.0.1:5000/student/-1 | "error": "Student ID -1 does not exist" | negative ID doesn’t exist |
| 0 | http://127.0.0.1:5000/student/0 | "error": "Student ID 0 does not exist" | ID 0 doesn’t exist |
| 9 | http://127.0.0.1:5000/student/9 | "courses\_passed" | max valid ID |
| > 9 | http://127.0.0.1:5000/student/10 | "error": "Student ID 10 does not exist | greater than max ID |
| float | http://127.0.0.1:5000/student/1.1 | "error": "Student ID 1.1 is not formatted correctly." | ID isn’t in float datatype |
| C | http://127.0.0.1:5000/student/C | "error": "Student ID C is not formatted correctly." | character is invalid |
| Total #Specifications: 7 | | |  |

Our goal was to check the information from *valid ID.* From our findings, we noticed that the API doesn’t hold any ID with value more than 9, and the datatype of the ID is always in integer data type.

A demo for some test cases is provided below:

|  |  |  |
| --- | --- | --- |
| **Valid** | **Doesn’t Exist** | **Formatted Error** |
|  |  |  |

**API 3: (/create):**

|  |  |  |
| --- | --- | --- |
| **Choice 1:** Name | **Choice 2:** Personnummer | **Choice 3:** CoursesPassed |
| Empty | [yy: 00-99][mm: 01-12][dd: 01-28]-[0000-9999] | [A-Z][00000-10000] |
| Special Characters | [yy: 00-99][mm: 00][dd: --]-[0000-9999] | [A-Z][000-999] |
| Numeric Values | [yy: 00-99][mm: >12][dd: --]-[0000-9999] | [A-Z][0000-9999] |
| String [A-Z a-z] | [yy: 00-99][mm: 01-12][dd: 00]-[0000-9999] |  |
|  | [yy: 00-99][mm: 01-12][dd: >31]-[0000-9999] |  |
|  | [yy: 00-99][mm: 01-12][dd: >31]-[0000-9999] |  |
|  | [yy: 00-99][mm: 02][dd: >28]-[0000-9999] |  |
| #Total Specification: 4 \* 7 \* 3 = 84 | | |

After deducing specifications:

|  |  |  |
| --- | --- | --- |
| **Choice 1:** Name | **Choice 2:** Personnummer | **Choice 3:** CoursesPassed |
| Empty | dont care (-) | **-** |
| Special Characters | **-** | **-** |
| Numeric Values | **-** | **-** |
| String [A-Z a-z] | [yy: 00-99][mm: 01-12][dd: 01-28]-[0000-9999] | [A-Z][0000-9999] |
| **-** | [yy: 00-99][mm: 00][dd: --]-[0000-9999] | **-** |
| **-** | [yy: 00-99][mm: >12][dd: --]-[0000-9999] | **-** |
| **-** | [yy: 00-99][mm: 01-12][dd: 00]-[0000-9999] | **-** |
| **-** | [yy: 00-99][mm: 01-12][dd: >31]-[0000-9999] | **-** |
| **-** | [yy: 00-99][mm: 01-12][dd: >31]-[0000-9999] | **-** |
| **-** | [yy: 00-99][mm: 02][dd: >28]-[0000-9999] | **-** |
| **-** | - | [A-Z][9999<] |
| **-** | - | [A-Z][999>] |
| **#Specification:** 3 + 2 + 6 + (1 \* 1 \* 1) = 12 | | |

A demo for some test cases is provided below:

|  |  |  |
| --- | --- | --- |
| **Valid**  (but should’ve been invalid) |  | Names shouldn’t have contained special chars/numeric values; February month cannot have 30 days |
| **Invalid Course Format** |  | courses have to contain 3 characters followed by 4 digits |
| **Corrected** |  | Passed |

**API 4: (/update/{student\_id}):**

|  |  |  |
| --- | --- | --- |
| **Choice 1:** ID | **Choice 2:** Name | **Choice 3:** CoursesPassed |
| Exists | Empty | [A-Z][9999<] |
| Doesn’t Exist | Special Characters | [A-Z][999>] |
|  | Numeric Values | [A-Z][0000-9999] |
|  | String [A-Z a-z] |  |
| #Total Specification: 2 \* 4 \* 3 = 24 | | |

After deducing specifications:

|  |  |  |
| --- | --- | --- |
| **Choice 1:** ID | **Choice 2:** Name | **Choice 3:** CoursesPassed |
| Doesn’t Exist | **-** | **-** |
| - | Empty | **-** |
| - | Special Characters | **-** |
| - | Numeric Values |  |
| - | - | [A-Z][9999<] |
| - | - | [A-Z][999>] |
| Exists | String [A-Z a-z] | [A-Z][0000-9999] |
| #Total Specification: 1 + 3 + 2 + (1 \* 1 \* 1) = 7 | | |

A demo for some test cases is provided below:

|  |  |  |
| --- | --- | --- |
| **Valid** (Name Change) |  | Name changed from “Romano**va**” to “Romano**ff**”, and added an additional course; |
| **Invalid Course Format** |  | Issue with ID 8:  The info cannot be updated for this one, as the personnummer is already invalid |

**API 5: (/delete/{student\_id}):**

|  |  |  |  |
| --- | --- | --- | --- |
| **Choice 1:** “student\_id” | **http requests** | **Output** | **Decision** |
| 11 | http://127.0.0.1:5000/student/11 | "deleted": "11" | valid ID |
| 0 > | http://127.0.0.1:5000/student/-1 | "error": "Student ID -1 does not exist" | negative ID doesn’t exist |
| 0 | http://127.0.0.1:5000/student/0 | "error": "Student ID 0 does not exist" | ID 0 doesn’t exist |
| > 14 | http://127.0.0.1:5000/student/15 | "error": "Student ID 15 does not exist | greater than max ID |
| float | http://127.0.0.1:5000/student/1.1 | "error": "Student ID 1.1 is not formatted correctly." | ID isn’t in float datatype |
| Chars | http://127.0.0.1:5000/student/C | "error": "Student ID C is not formatted correctly." | character is invalid |
| Total #Specifications: 6 | | |  |

1. *B. We have added some more students, as a result our current maximum student ID is 14]*

A demo for some test cases is provided below:

|  |  |
| --- | --- |
| **Before** | **After** |
|  |  |

**API 6: (/program):**

|  |  |  |
| --- | --- | --- |
| **#Choices** | **Rationale** | **#Specification** |
| 1 | For this API, the number of testable choice is 1 since it does not any more testable choice. | 1 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Request Type** | **Body** | **URL** | **Assertions** | **Description** |
| GET | This request does not have a body | http://127.0.0.1:5000/program | pm.**test**("Successful POST request", **function** () {  pm.expect(pm.response.code).to.be.oneOf([201, 202]);  });  pm.**test**("Response time is less than 200ms", **function** () {  pm.expect(pm.response.responseTime).to.be.below(200);  });  pm.**test**("Status code is 200", **function** () {  pm.response.to.have.status(200);  }); | Our goal for this test was to get all programs with their respective IDs. The test case was checked to see:   1. if the response time is less than 200 ms 2. if the status code is 200 3. if the POST request was successful (with either 201 or 202 response code).   from the test, 2/3 were successful. |

Test case check for API 1:

**API 7: (/program/{program\_id}):**

|  |  |  |  |
| --- | --- | --- | --- |
| **Choice 1:** “program\_id” | **http requests** | **Output** | **Decision** |
| 1 | http://127.0.0.1:5000/program/1 | "courses\_required": [  "CSE2213","CSE1110","CSE1111","CSE1114","CSE1115","CSE1116","CSE2117","CSE2118","CSE3119","CSE4010"] | valid ID |
| 0 > | http://127.0.0.1:5000/program/-1 | "error": "Program ID -1 does not exist" | negative ID doesn’t exist |
| 0 | http://127.0.0.1:5000/program/0 | "error": "Program ID 0 does not exist" | ID 0 doesn’t exist |
| > 2 | http://127.0.0.1:5000/program/3 | "error": "Program ID 3 does not exist" | greater than max program ID |
| float | http://127.0.0.1:5000/program/1.1 | "error": "Program ID 1.1 is not formatted correctly." | program ID isn’t in float datatype |
| Chars | http://127.0.0.1:5000/program/C | "error": "Program ID C is not formatted correctly."  formatted correctly." | character is invalid |
| Total #Specifications: 6 | | |  |

**API 8: (/finished/{student\_id}/{program\_id}):**

|  |  |
| --- | --- |
| **Choice 1:** student\_id | **Choice 2:** program\_id |
| 1 | 1 |
| < 1 | < 1 |
| > 14 | > 2 |
| Float | Float |
| Chars | Chars |
| #Total Specification: 5 \* 5 = 25 | |

Demo test cases are provided below:

|  |  |  |
| --- | --- | --- |
| **URL** | **Output** | **Result** |
| http://127.0.0.1:5000/finished/1/1 | "completed\_courses": 10,  "status": **true** | Pass |
| http://127.0.0.1:5000/finished/1/0 | "error": "Program ID 0 does not exist" | Fail |
| http://127.0.0.1:5000/finished/2/2 | "completed\_courses": 0,  "status": **false** | Pass |
| http://127.0.0.1:5000/finished/2/-1 | "error": "Program ID -1 does not exist" | Fail |
| http://127.0.0.1:5000/finished/2/3 | "error": "Program ID 3 does not exist" | Fail |
| http://127.0.0.1:5000/finished/\*/3 | "error": "Student ID \* is not formatted correctly." | Fail |