Representative Values, Choices, and Constraints for the Tests

Test Plans

Name: Md . Farnas Utsho

Get Student List and Information for Individual ID

Parameter**:**Value of student ID

Choice: Id

1. valid id

2.No Id

3. Id > number of students

4. Add other parameters

Parameter: Response Body

Choice:JSON body

1. Correct format of student list

2. Contains TimeStamp [Single]

3. Response time [Single]

4. Contains Valid student information [If student ID is provided in the endpoint]

Create an entry Using post-request

Parameter:Name

Choice: String

1. Empty
2. Valid input
3. Long string [Single]

4. same name more than once [Single]

Parameter: Person Number

Choice: String

1. Empty
2. Valid input
3. Not String [Single]
4. same number more than once

Choice :Format of Person Number

1.Valid Format

2.Invalid Format [ ]

3.No hyphen

4. hyphen more than one [Single]

Parameter: Course Id

Choice: List

1. Empty List [Error]
2. Valid List

Choice:Format

1. Valid
2. first 3 digit
3. last 4 string

Route: /student

URL: http://127.0.0.1:5000/student

Method: Get

Description: Get a list of all students,

with their IDs.

Test Case 1: Verify that Get request with the endpoint { /student } returns a list containing all student's information

Test Case 2: Verify that the Get request with no endpoint returns a 404 status code

Test Case 3: Verify that the response body is in the correct format when the request is successful

Test case 4: Verify that the response body contains a timestamp when the request is successful.

Test Case 5: Verify that the response time is less than 10ms when the request is successful

Route : /student /{student_id}

URL: http://127.0.0.1:5000/student/6

Method: Get

Description: Get data for a single Student.

Test Case 6: Verify that the Get request with the ID provided at the endpoint returns the correct student information

Test Case 7: Verify That an error message occurs when the ID is not in the list

Test Case 8: Verify that an error message occurs when other parameter is added instead of the id

Route: /create

URL: http://127.0.0.1:5000/create

Method:POST

Description: Create a new student record in the database, if a record does not already exist for that person number.

Test Case 9: Verify the Post request with an empty name show an error message

Body :

```
{ "name": "",
```

```
  "personnummer": "950101-3921",
  "courses_passed": ["CSE1234", "DSE5678"]
```

Test Case 10: Verify that the Post request with a valid name is successful

Body :

```
  {"name": "Jacky",
  "personnummer": "950101-4321",
  "courses_passed": ["CSE1234", "DSE5678"]
}
```

Test Case 11: Verify that the Post request with the same name show an error message

Body :

```
 {  "name": "Jacky",
  "personnummer": "950101-6321",
  "courses_passed": ["CSE1234", "DSE5678"]
}
```

Test Case 12: Verify that the Post request with the correct person number format is successful

Body :

```
 {  "name": "Alpha",
  "personnummer": "950101-7321",
  "courses_passed": ["CSE1234", "DSE5678"]
}
```

Test Case 13:Verify that the Post request with the used person number shows error message

Body :

```
  { "name": "Ron Jack",
```

```
  "personnummer": "950101-3321",
  "courses_passed": ["CSE1234", "DSE5678"]
}
```

Test case 14:Verify that the Post request with incorrect course ID format show an error message

Body :

```
{ "name": "Ron Jack",
  "personnummer": "950101-3821",
  "courses_passed": ["CSE124", "DSE5678"]
}
```

Test case 15: Verify that the Post request with an empty course list shows an error message

Body :

```
{ "name": "Ron Jack",
  "personnummer": "950101-3721",
  "courses_passed": []
}
```

Test Case No	Expected Result	Actual Result	Passed / Failed
1	The response body must Contain the Student's list	The response body Contains the Student's list	Passed
2	The Status Code should be 404	Status code 404	Passed
3	Response Body must contain the Correct format of JSON	Response Body contains the Correct format of JSON	Passed
4	The response body must contain TimeStamp	The response body does not contain TimeStamp	Failed
5	The response should be time is less than 10ms	Response time is less than 10ms	Passed

6	The response body must have the name of the student with the respective ID passed.	The response body has the name of the student with the respective ID passed.	Passed
7	Response Body should contain an error message	The response body contains an error message	Passed
8	The status code should be 404	The status code is 200	Failed
9.	Response Body should display a message blank or null name	Response body has message Blank or null name	Passed
10	The status code should be 201(Created)	Status is 201 (Created)	Passed
11	Post requests with an existing name must show an error message	No error message,post request is successful	Failed
12	Post request with a correct format of person number should be successful	The post request is successful. Status code 201(Created)	Passed
13	Post request with a used person number should display an error message	The response body contains an error message	Passed
14	Post requests with incorrect course ID format should display an error message in response	In response error message occurs that says malformed course ID	Passed
15	Post requests with a null list should display an error message in response	Error Message is not displayed	Failed

Test Case 1:

Route: /update/{student_id}

URL: http://127.0.0.1:5000/update/2

Method: PUT

Choice: Response Status Code Value

Representative Values: An HTTP status code indicating a successful response, such as 200.

Constraints: SINGLE

Goal: To verify that the HTTP response status code returned by the server matches the expected status code.

Purpose: This test case ensures that the server responds with the expected status code, indicating whether the request was successful or encountered an error. It helps to validate the correctness of the server's behavior and the consistency of the API's response.

Test Case 2:

Route: /update/{student_id}

URL: http://127.0.0.1:5000/update/2

Method: PUT

Choice: Content-Type Header Value

Representative Values: The return Content-Type header must be with the value application/JSON.

Constraints: SINGLE

Goal: To ensure that the HTTP response contains the required Content-Type header with the value application/JSON.

Purpose: This test case validates that the server responds with the correct content type, indicating that the response body is formatted as JSON data.

Test Case 3:

Route: /update/{student_id}

URL: http://127.0.0.1:5000/update/2

Method: PUT

Choice: Presence of Error Message in response data

Representative Values: The response data includes an "error" field indicating the presence of an error message. Here error message is absent in the response data.

Constraints: ERROR

Goal: To ensure that an error message is present in the response data when an error occurs.

Purpose: This test case verifies the handling of errors by the API. It checks whether the server correctly returns an error message when requested data is invalid or when an error occurs during processing. Verifying the presence of an error message helps ensure that the API provides informative feedback.

Test Case 4:

Route: /update/{student_id}

URL: http://127.0.0.1:5000/update/2

Method: PUT

Choice: Error Message Content in response data

Representative Values: The response data includes an "error" field, but its value is an empty string.

Constraints: ERROR

Goal: To ensure that the error message returned in the response data is a non-empty string.

Purpose: This test case verifies the completeness and clarity of error messages returned by the API. It checks whether the error message is informative and meaningful, helping clients understand the nature of the error encountered. Verifying the presence of a non-empty error message enhances the usability and troubleshooting capabilities of the API.

Test case 5:

Route: /update/{student_id}

URL: http://127.0.0.1:5000/update/2

Method: PUT

Choice: The requested HTTP method

Representative Values: The request method is "PUT".

Constraints: SINGLE

Goal: To ensure that the HTTP method used in the request is "PUT".

Purpose: This test case validates whether the API endpoint is accessed using the correct HTTP method. Verifying the HTTP method helps ensure that the client is interacting with the API endpoint as intended, preventing unintended actions or security vulnerabilities.

Test case 6:

Route: /delete/{student_id}

URL: http://127.0.0.1:5000/delete/5

Method: DELETE

Choice: The HTTP status code is returned in the response.

Representative Values: The response status code is 200, indicating a successful request.

Constraints: SINGLE

Goal: To ensure that the HTTP response status code is 200.

Purpose: This test case validates whether the server successfully processes the request and returns a successful response. Verifying the HTTP status code helps ensure that the API endpoint behaves as expected and fulfills the client's request.

Test Case 7:

Route: /delete/{student_id}

URL: http://127.0.0.1:5000/delete/5

Method: DELETE

Choice: Check if the response indicates an error.

Representative Values: The response does not include any error message or indication.

Constraints: ERROR

Goal: To verify whether the response indicates an error condition.

Purpose: This test case checks whether the API response contains an indication of an error. It ensures that the client receives appropriate feedback when an error occurs during the request processing.

Test Case 8:

Route: /delete/{student_id}

URL: http://127.0.0.1:5000/delete/5

Method: DELETE

Choice: The content of the "deleted" field in the response data.

Representative Values: The response data includes a "deleted" field with a non-empty string value.

Constraints: SINGLE

Goal: To ensure that the "deleted" field in the response data is a non-empty string.

Purpose: This test case validates the completeness of the response data regarding the "deleted" field. It checks whether the API provides a meaningful indication when a record is successfully deleted.

Test Case 9:

Route: /delete/{student_id}

URL: http://127.0.0.1:5000/delete/5

Method: DELETE

Choice: The value of the "Content-Type" header in the HTTP response.

Representative Values: The response includes the "Content-Type" header with the value "application/JSON".

Constraints: SINGLE

Goal: To ensure that the "Content-Type" header in the response indicates JSON content.

Purpose: This test case validates whether the API response is properly formatted as JSON data. Verifying the "Content-Type" header value helps ensure that the client receives data in the expected format.

Test Case 10:

Route: /delete/{student_id}

URL: http://127.0.0.1:5000/delete/5

Method: DELETE

Choice: Whether the response data indicates the presence of an error.

Representative Values: The response data does not include an "error" field.

Constraints: ERROR

Goal: To verify whether the response indicates the presence of an error.

Purpose: This test case checks whether the API response includes an error indication, providing feedback to the client application about the success or failure of the request.

Test Case 11:

Route: /update/{student_id}

URL: http://127.0.0.1:5000/update/2

Method: PUT

Choice: Check the student ID is present in the response data.

Representative Values: The response data includes the student ID.

Constraints: SINGLE

Goal: To verify the presence of the student ID in the response data.

Purpose: This test case ensures that the API returns the student ID as part of the response data. Verifying the presence of the student ID helps ensure that the relevant student information is included in the response, allowing client applications to correctly identify and process the data.

Test Case 12:

Route: /update/{student_id}

URL: http://127.0.0.1:5000/update/2

Method: PUT

Choice: The format of the personnummer in the response data.

Representative Values: The personnummer in the response data matches the expected format.

Constraints: SINGLE:

Goal: To ensure that the personnummer in the response data has the correct format.

Purpose: This test case validates the correctness of the personnummer format returned by the API. Verifying the format helps ensure that the personnummer is accurately represented and can be correctly interpreted by client applications.

Test Case 13:

Route: /update/{student_id}

URL: http://127.0.0.1:5000/update/2

Method: PUT

Choice: Check the student name is present in the response data.

Representative Values: The response data includes the student name.

Constraints: SINGLE

Goal: To ensure the presence of the student name in the response data.

Purpose: This test case validates whether the API returns the student name as part of the response data. Verifying the presence of the student name ensures that essential information about the student is included in the response, enabling client applications to accurately identify and display the student's details.

Test Case No	Expected Result	Actual Result	Passed / Failed
1	The status code should be 200	The status code is 200	Passed
2	Expected result JSON Value	Responded with JSON	Passed
3	Should show an error message in response	Absent error message	Failed
4	The response should contain a non-empty string	The response contains a string	Failed
5	The expected request method is PUT	The request method is PUT	Passed
6	The status code should be 200	The status code is 200.	Passed
7	The response should contain an error message	The response does not contain an error message	Failed
8	The expected deleted field is a non-empty string	Responded in non-empty string	Passed
9.	Expected result JSON Value	Responded with JSON	Passed
10	Should show an error message in response	Absent error message	Failed
11	Student ID is present in response data	Student ID id is present	Passed
12	Expected the correct personnumber format	Returns the correct personnumber format	Passed
13	Expected Student name is present	Student name is present	Passed

Test Case 1:

Route: /program/{program_id}

URL: http://127.0.0.1:5000/program/1

Method: GET

Choice: Response Status Code

Representative Value: To check whether the response status code is 200 or not.

Constraint: SINGLE

Test Case Goal: This test case verifies that the system responds with a status code indicating successful processing of the request.

Test Case Purpose: This test confirms the endpoint responds appropriately for successful processing.

Test Case 2:

Route: /program/{program_id}

URL: http://127.0.0.1:5000/program/1

Method: GET

Choice: Response Time

Representative Value: Response time less than 200ms

Constraint: SINGLE

Test Case Goal: This test case verifies that the system responds with a response time below 200 milliseconds.

Test Case Purpose: This test helps assess the API's performance. A response time below 200ms indicates a very fast response.

Test Case 3:

Route: /program/{program_id}

URL: http://127.0.0.1:5000/program/2

Method: GET

Choice: Response Status Code

Representative Value: To check whether the response status code is 200 or not.

Constraint: SINGLE

Test Case Goal: This test case verifies that the system responds with a status code indicating successful processing of the request.

Test Case Purpose: This test confirms the endpoint responds appropriately for successful processing.

Test Case 4:

Route: /program/{program_id}

URL: http://127.0.0.1:5000/program/2

Method: GET

Choice: Response Body Content

Representative Value: The test uses "DSE2116" as the specific string to search for in the response body.

Constraint: SINGLE.

Test Case Goal: This test case verifies that the response body contains the string "DSE2116".

Test Case Purpose: This test helps identify whether a specific piece of information (DSE2116) is present in the response data.

Test Case 5:

Route: /program/{program_id}

URL: http://127.0.0.1:5000/program/2

Method: GET

Choice: Response Body Content

Representative Value: The test uses "DSE5000" as the specific string to search for in the response body.

Constraint:ERROR

Test Case Goal: This test case verifies that the response body does not contain the string "DSE5000".

Test Case Purpose: This test helps identify whether DSE5000 is present in the response data or not.

Test Case 6:

Route: /program/{program_id}

URL: http://127.0.0.1:5000/program/3

Method: GET

Choice: Response Status Code

Representative Value: To check whether the response status code is 200 or not.

Constraint: ERROR [As, program id 3 does not exist].

Test Case Goal: This test case verifies that the system is responding correctly.

Test Case Purpose: This test confirms the endpoint responds appropriately for successful processing.

Test Case 7:

Route: /finished/{student_id}/ {program_id}

URL: http://127.0.0.1:5000/finished/1/1

Method: GET

Choice: Response Status Code

Representative Value: To check whether the response status code is 200 or not.

Constraint: SINGLE

Test Case Goal: This test case verifies that the system responds with a status code indicating successful processing of the request.

Test Case Purpose: This test confirms the endpoint responds appropriately for successful processing.

Test Case 8:

Route: /finished/{student_id}/ {program_id}

URL: http://127.0.0.1:5000/finished/1/1

Method: GET

Choice: Response Body Content (Structure and Values)

Representative Value: The provided JSON object represents the expected structure and data for a successful response.

Constraint: ERROR [As, completed_course object is incorrect and the value of completed_courses is invalid]

Test Case Goal: This test case verifies that the response body from the API exactly matches the expected structure and data values.

Test Case Purpose: This test ensures the API endpoint returns the correct information in the desired format.

Test Case 9:

Route: /finished/{student_id}/ {program_id}

URL: http://127.0.0.1:5000/finished/-1/-1

Method: GET

Choice: Response Status Code

Representative Value: To check whether the response status code is 200 or not.

Constraint: ERROR[As, student id -1 and program id -1 are invalid]

Test Case Goal: This test case verifies that the system responds with a status code indicating successful processing of the request.

Test Case Purpose: This test confirms the endpoint responds appropriately for successful processing.

Test Case No	Expected Result	Actual Result	Passed / Failed
1	The status code should be 200	The status code is 200	Passed
2	Expected response time less than 200 ms	Responding time is below 200 ms.	Passed
3	The status code should be 200	The status code is 200	Passed
4	Searching for specific data in the JSON body without any error.	Expected result is found	Passed
5	Searching for “DSE5000” in response body content will give an error message.	Error is found here.	Passed
6	The status code should be 200	The status code is not 200.	Failed
7	The status code should be 200	The status code is 200.	Passed

8	The JSON object in response data should be matched for successful response.	Error is found in JSON object.	Failed
9.	There will throw an error with 404 response code.	An error is occurred.	Passed