CZ3003 - Software System Analysis and Design

Testing for *Ultrareign*

Group 3 - Anything

Name	Matriculation No.
Tan Yunliang	**U1922635H**
Teh Hock Tuck	**U1921989E**
Nur Lydia Afiqah	**U1922339C**
Goh Tse Yinn Sheryl	**U1922180C**
Ernest Tan Yan Heng	**U1920436K**

Table of Contents

Introduction

Our testing will be conducted in 2 phases, Unit Testing and Integration Testing. For Unit Testing, we will be covering the Web Front-end and the API. Integration testing will be conducted post-integration of our web-based frontend, backend and game application.

Testing Techniques

For the web front-end, we will be using **Equivalence Class Partitioning**, which is a black-box testing technique that divides the input domain into valid and invalid inputs. The tests will simulate user inputs for valid and invalid cases, and use assertions to check that the correct system behaviour is observed.

Unit testing

Web Front-end

The web front-end was built upon the next.js framework, and testing will be done using Cypress, which is a End-to-End testing framework for front-end testing. Cypress is useful as it can simulate user input within the front-end, as well as create assertions to test for specific results.

Additionally, Cypress provides hooks with the *before()* function, which allows for preset input and conditions before conducting each test.

Our test cases will be split primarily into Student test cases and Professor test cases, and then further segregating into the individual functionalities of each specific user, and further into invalid and valid test cases as mentioned earlier

Steps before testing
1. Install *Cypress* using NPM
   a. Using command *npm install cypress --save-dev*
2. Add Cypress into *package.json* scripts
   a. Add "e2e" : "cypress open"
3. Run Cypress to create necessary files in project
   a. Using command *npm run e2e*

Test Case Coverage

Test Case 1- Student Login

Test Case Name	Wrong password
Test Case ID	1
Description	Student inputs username and wrong password and login fails
Preconditions needed to run this test case	Student must have an account in system
Event Sequence	
Input events (performed by tester)	*Output events (observed by system tester)*
Username: stud1 Password: wrongpassword	
Click 'Login' button	
	"Wrong Input" alert shown

Test Case Name	Empty Field
Test Case ID	2

Description	Student does not input either username or password and login fails
Preconditions needed to run this test case	Student must have an account in system
Event Sequence	
Input events (performed by tester)	Output events (observed by system tester)
Username: stud1 Password:	
Click 'Login' button	
	"Enter all fields!" alert shown

Test Case Name	Successful login
Test Case ID	3
Description	Student inputs username and corresponding password and logins successfully
Preconditions needed to run this test case	Student must have an account in system
Event Sequence	
Input events (performed by tester)	Output events (observed by system tester)
Username: stud1 Password: stud1pw	
Click 'Login' button	
	Student logins successfully and brought to "Play" page

Test Case 2- Student create questions

Test Case Name	Any field left empty
Test Case ID	3
Description	Student leaves a field empty and unable to successfully create question
Preconditions needed to run this test case	Student is logged in
Event Sequence	
Input events (performed by tester)	Output events (observed by system tester)
World: Custom Level: 1 Question: 1 Question: What is 3 x 3? Options: 4 5 9 Answer: C	

Click "Done" button	
	"Enter all fields" alert shown.

Test Case Name	All fields filled
Test Case ID	4
Description	Student enters all fields and is able to successfully create question
Preconditions needed to run this test case	Student is logged in
Event Sequence	
Input events (performed by tester)	*Output events (observed by system tester)*
World: Custom Level: 1 Question: 1 Question: What is 3 x 3? Options: 4 5 9 10 Answer: C	
Click "Done" button	
	Question successfully created and student is brought to "Read" page to view created question

Test Case 3 - Student update questions

Test Case Name	Any field left empty
Test Case ID	5
Description	Student leaves a field empty and is unable to update question successfully
Preconditions needed to run this test case	Student is logged in Student chooses a question to update
Event Sequence	
Input events (performed by tester)	*Output events (observed by system tester)*
Student deletes option C field	
Click "Done" button	
	"Enter all fields" alert shown.

Test Case Name	All fields updated and filled
Test Case ID	6

Description	Student enters all fields and is able to successfully update question
Preconditions needed to run this test case	Student is logged in
Event Sequence	

Input events (performed by tester)	*Output events (observed by system tester)*
Student changes option D to "23"	
Click "Done" button	
	Question successfully updated question and student is brought to "Read" page to view updated question

Test Case 4 - Professor Login

Test Case Name	Wrong password
Test Case ID	7
Description	Professor inputs username and wrong password and login fails
Preconditions needed to run this test case	Professor must have an account in system
Event Sequence	

Input events (performed by tester)	*Output events (observed by system tester)*
Username: prof1 Password: wrongpassword	
Click 'Login' button	
	"Wrong Input" alert shown

Test Case Name	Empty Field
Test Case ID	8
Description	Professor does not input either username or password and login fails
Preconditions needed to run this test case	Professor must have an account in system
Event Sequence	

Input events (performed by tester)	*Output events (observed by system tester)*
Username: prof1 Password:	
Click 'Login' button	
	"Enter all fields!" alert shown

Test Case Name	Successful login
Test Case ID	9

Description	Professor inputs username and corresponding password and logins successfully
Preconditions needed to run this test case	Professor must have an account in system
Event Sequence	
Input events (performed by tester)	*Output events (observed by system tester)*
Username: prof1 Password: prof1pw	
Click 'Login' button	
	Professor logins successfully and brought to "Summary Report" page

Test Case 5 - Summary Report

Test Case Name	View Overall Summary Report
Test Case ID	10
Description	Professor able to view overall summary report
Preconditions needed to run this test case	Professor is logged in
Event Sequence	
Input events (performed by tester)	*Output events (observed by system tester)*
Clicks on "Summary Report" on navigation bar	
	Professor able to see an overview of all his students' scores

Test Case Name	View Student Summary Report
Test Case ID	11
Description	Professor able to view students' summary report
Preconditions needed to run this test case	Professor is logged in Professor is on "Summary Report" page
Event Sequence	
Input events (performed by tester)	*Output events (observed by system tester)*
Chooses a student name from dropdown bar in "Summary Report" page	
	Professor able to see an overview of specified students' scores

Test Case 6 - Professor create questions

Test Case Name	Any field left empty
Test Case ID	12

Description	Professor leaves a field empty and unable to successfully create question
Preconditions needed to run this test case	Professor is logged in
Event Sequence	
Input events (performed by tester)	*Output events (observed by system tester)*
World: Custom Level: 1 Question: 1 Question: What is 3 x 3? Options: 4 5 9 Answer: C	
Click "Done" button	
	"Enter all fields" alert shown.

Test Case Name	All fields filled
Test Case ID	13
Description	Professor enters all fields and is able to successfully create question
Preconditions needed to run this test case	Professor is logged in
Event Sequence	
Input events (performed by tester)	*Output events (observed by system tester)*
World: Custom Level: 1 Question: 1 Question: What is 3 x 3? Options: 4 5 9 10 Answer: C	
Click "Done" button	
	Question successfully created and professor is brought to "Read" page to view created question

Test Case 7 - Professor update questions

Test Case Name	Any field left empty
Test Case ID	14

Description	Professor leaves a field empty and is unable to update question successfully
Preconditions needed to run this test case	Professor is logged in Professor chooses a question to update
Event Sequence	
Input events (performed by tester)	*Output events (observed by system tester)*
Professor deletes option C field	
Click "Done" button	
	"Enter all fields" alert shown.

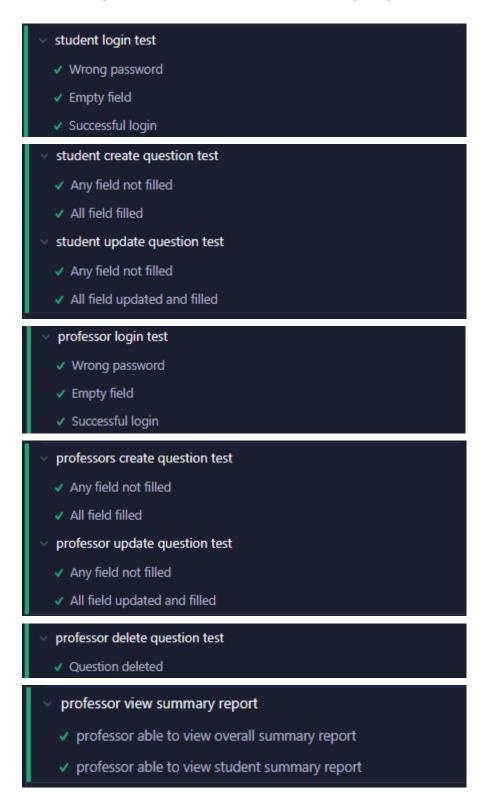| Test Case Name	All fields updated and filled
Test Case ID	15
Description	Professor enters all fields and is able to successfully update question
Preconditions needed to run this test case	Professor is logged in
Event Sequence	
Input events (performed by tester)	*Output events (observed by system tester)*
Professor changes option D to "23"	
Click "Done" button	
	Question successfully updated question and student is brought to "Read" page to view updated question

Test Case 8 - Delete question

Test Case Name	Question Deleted
Test Case ID	16
Description	Professor removes a question from the database
Preconditions needed to run this test case	Professor is logged in
Event Sequence	
Input events (performed by tester)	*Output events (observed by system tester)*
Professor clicks on desired question	
	Window alert shows "Are you sure you wish to delete this question?"
Professor clicks on "Confirm"	
	Window confirmation show "Question Deleted"

Summary of Automated black-box testing results
The results of the testing completed using the automated Cypress test scripts automated test script are shown below. Detailed screenshots of the specific tests are included in Appendix A.

As seen from these screenshots, the results of the Equivalence Partitioning tests are shown below, with all of them being successful. This completes the testing programme created for the web frontend of *Ultrareign*.

> student login test
  ✔ Wrong password
  ✔ Empty field
  ✔ Successful login

> student create question test
  ✔ Any field not filled
  ✔ All field filled
> student update question test
  ✔ Any field not filled
  ✔ All field updated and filled

> professor login test
  ✔ Wrong password
  ✔ Empty field
  ✔ Successful login

> professors create question test
  ✔ Any field not filled
  ✔ All field filled
> professor update question test
  ✔ Any field not filled
  ✔ All field updated and filled

> professor delete question test
  ✔ Question deleted

> professor view summary report
  ✔ professor able to view overall summary report
  ✔ professor able to view student summary report

Integration testing

<u>HTTPS Endpoints</u>
The backend was built using the Strapi Content Management System, and testing was done using Swagger, which is a endpoint framework used to build and describe API endpoints. Swagger is also used for testing via the API documentation page, which allows for endpoint testing.

<u>Steps Before Testing</u>
1. Deploy Dev Environment on local machine
    a. Setting .env and server.js config file for connection to testing database (*remember to whitelist ip in digitalocean).
    b. Hosting on Localhost under development.
2. Ensure Data and content has test data as specified under Test/testdata .json files.
3. Setup SwaggerUI with All-Access JWT Token.
    a. Navigate to http://localhost:1337/documentation/v1.0.0#/
    b. Authorize Website to call Strapi Endpoint with All-Access Token:
       c047f2498e4be5f8c3d8502d9e918f22a1ba3e7fb89d8fde41dc71dbd6b5f6b972177d8695ec8 1abd4cfadf0afdab03468f84a269b20692080795cf39f2a50913f8f6f68139802ad45384ce297e ad1318d23a1c2569c0bacdf95580e92476f9637c91901bcc3ce168b6e37f6544f665f2a3e5f2b 5139deeb941dab0dc867bb48

<u>Test Case Coverage</u>
1. User-Permissions
    a. Login with Email/Password - /auth/local (post)
    b. Register for Email/Password - /auth/local/register (post)
    c. Login with Facebook - /auth/connect/facebook (get)
    d. Callback from Facebook after authentication - /auth/facebook/callback (get)
    e. Get List of Users - /users (get)
    f. Get Current LoggedIn User - /users/me (get)
2. Assignment
    a. Get List of Assignment - /assignments (get)
    b. Modify Assignment - /assignments/{id} (post/get/put/delete)
3. Assignment-score-composition
    a. Get List of Assignment-score-composition - /assignment-score-compositions (get)
    b. Modify Assignment-score-composition - /assignment-score-compositions/{id} (post/get/put/delete)
4. Level
    a. Get List of Level - /levels (get)
    b. Modify Level - /levels/{id} (post/get/put/delete)
5. Level-question–composition
    a. Get List of Level-question–composition - /level-question–compositions (get)
    b. Modify Level-question–composition - /level-question–compositions/{id} (post/get/put/delete)
6. Question
    a. Get List of Question - /questions (get)
    b. Modify Question /questions/{id} (post/get/put/delete)
7. Score
    a. Get List of Score - /scores (get)
    b. Modify Score - /scores/{id} (post/get/put/delete)
8. World
    a. Get List of World - /worlds (get)
    b. Modify World - /worlds/{id} (post/get/put/delete)
9. World-level-composition
    a. Get List of World-level-composition - /world-level-compositions (get)

b. Modify World-level-composition - /world-level-compositions/{id} (post/get/put/delete)

Testcase 1: User-Permissions

Test Case ID: 1a	Test Priority (L/M/H): H
Test Title: Login with Email/Password - /auth/local (post)	
Description: Normal Login using Email & Password	
Designed By: Tan Yunliang	Test Designed Date: 5/11/2022
Test Executed By: Tan Yunliang	Test Executed Date: 5/11/2022
Pre-conditions: 10. All Access Token Provided	
Dependencies: NIl	

Event Sequence:		
Input Event	Expected Event	Output Event

http://localhost:1337/api/auth/local is query with the following:
 	
curl -X 'POST' \
 	
'http://localhost:1337/api/auth/local' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer c047f2498e4be5f8c3d8502d9e918f22a1ba3e7fb89d8fde41dc71dbd6b5f6b972177d8695ec81abd4cfadf0afdab03468f84a269b20692080795cf39f2a50913f8f6f68139802ad45384ce297ead1318d23a1c2569c0bacdf95580e92476f9637c91901bcc3ce168b6e37f6544f665f2a3e5f2b5139deeb941dab0dc867bb48' \
  -H 'Content-Type: application/json' \
  -d '{
  "identifier": "professor1",
  "password": "professor1"
}'
	Server Responded with code 200	**Server Responded with code 200:**
		
		{
		  "jwt": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MSwiaWF0IjoxNjY3NjI1MDc3LCJleHAiOjE2NzAyMTcwNzd9.8KzGS_nSKIIkaHhJsTzkz9foNRGCDurRkDXHmK8T-pE",
		  "user": {
		  "id": 1,
		  "username": "professor1",
		  "email": "professor1@school.com",
		  "provider": "local",
		  "confirmed": true,
		  "blocked": false,
		  "createdAt": "2022-11-04T17:23:00.278Z",
		  "updatedAt": "2022-11-04T17:23:00.278Z"
		  }
		}

http://localhost:1337/api/auth/local is query with the following:
 	
curl -X 'POST' \
 	
'http://localhost:1337/api/auth/local' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer c047f2498e4be5f8c3d8502d9e9
	Server Responded with code 400	**Server Responded with code 400:**
		
		{
		  "data": null,
		  "error": {
		  "status": 400,
		  "name": "ValidationError",
		  "message": "Invalid identifier or password",
		  "details": {}

18f22a1ba3e7fb89d8fde41dc71d
bd6b5f6b972177d8695ec81abd4
cfadf0afdab03468f84a269b2069
2080795cf39f2a50913f8f6f68139
802ad45384ce297ead1318d23a
1c2569c0bacdf95580e92476f963
7c91901bcc3ce168b6e37f6544f6
65f2a3e5f2b5139deeb941dab0d
c867bb48' \
  -H 'Content-Type:
application/json' \
  -d '{
  "identifier": "external1",
  "password": "external1"
}'

Test Case ID: 1b	Test Priority (L/M/H): M
Test Title: Register for Email/Password - /auth/local/register (post)	
Description: Registering for Email and Password	
Designed By: Tan Yunliang	Test Designed Date: 5/11/2022
Test Executed By: Tan Yunliang	Test Executed Date: 5/11/2022
Pre-conditions: 　1.　All Access Token Provided	
Dependencies: Nil	

Event Sequence:

Input Event	Expected Event	Output Event

http://localhost:1337/api/auth/local/register is query with the following:
 	
curl -X 'POST' \
 	
'http://localhost:1337/api/auth/local/register' \
　-H 'accept: application/json' \
　-H 'Authorization: Bearer c047f2498e4be5f8c3d8502d9e918f22a1ba3e7fb89d8fde41dc71dbd6b5f6b972177d8695ec81abd4cfadf0afdab03468f84a269b20692080795cf39f2a50913f8f6f68139802ad45384ce297ead1318d23a1c2569c0bacdf95580e92476f9637c91901bcc3ce168b6e37f6544f665f2a3e5f2b5139deeb941dab0dc867bb48' \
　-H 'Content-Type: application/json' \
　-d '{
	Server Responded with code 200	**Server Responded with code 200:**
		
		{
		　"jwt": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6NywiaWF0IjoxNjY3NjI1ODg1LCJleHAiOjE2NzAyMTc4ODV9.LcDXY3j8jyzz-E176AL2Uz5xUwX3gZMfoFYbTULWoys",
		　"user": {
		　　"id": 7,
		　　"username": "newStudent",
		　　"email": "newstudent@school.com",
		　　"provider": "local",
		　　"confirmed": true,
		　　"blocked": false,
		　　"createdAt": "2022-11-05T05:24:45.858Z",
		　　"updatedAt": "2022-11-05T05:24:45.858Z",

"username": "newStudent",
  "email":
"newStudent@school.com",
  "password": "newStudent"
}'
		"role": {
		    "id": 1,
		    "name": "Professor",
		    "description": "Professor",
		    "type": "authenticated",
		    "createdAt":
		"2022-11-04T17:17:18.144Z",
		    "updatedAt":
		"2022-11-04T18:16:50.094Z"
		    }
		  }
		}

Test Case ID: 1c	**Test Priority (L/M/H):** H
Test Title: Login with Facebook - /auth/connect/facebook (get)	
Description: Auth Provided by Facebook	
Designed By: Tan Yunliang	**Test Designed Date:** 5/11/2022
Test Executed By: Tan Yunliang	**Test Executed Date:** 5/11/2022

Pre-conditions:
    1. Setup Reverse Proxy to Localhost Dev Strapi
    2. Setup Whitelist from facebook Authentication and Callback to Strapi	

Dependencies: Nil	

Event Sequence:		
Input Event	**Expected Event**	**Output Event**
Navigate to <reverse proxy>/auth/connect/facebook	Redirect to Facebook Login Page	Redirects to Facebook Login Page

Test Case ID: 1d	**Test Priority (L/M/H):** H
Test Title: Callback from Facebook after authentication - /auth/facebook/callback (get)	
Description: Endpoint called by Facebook after Authenticating User	
Designed By: Tan Yunliang	**Test Designed Date:** 5/11/2022
Test Executed By: Tan Yunliang	**Test Executed Date:** 5/11/2022

Pre-conditions:
    1. Login with Facebook - /auth/connect/facebook (get) was run
    2. Authenticated by Facebook	

Dependencies: Nil	

Event Sequence:		
Input Event	**Expected Event**	**Output Event**

Login with into redirect Facebook Page	**Server Responded with code 200:**	**Server Responded with code 200:**

		{
		  "jwt": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6NywiaWF0IjoxNjY3NjI1ODg1LCJleHAiOjE2NzAyMTc4ODV9.LcDXY3j8jyzz-E176AL2Uz5xUwX3gZMfoFYbTULWoys",
		  "user": {
		    "id": 7,
		    "username": "newStudent",
		    "email": "newstudent@school.com",
		    "provider": "local",
		    "confirmed": true,
		    "blocked": false,
		    "createdAt": "2022-11-05T05:24:45.858Z",
		    "updatedAt": "2022-11-05T05:24:45.858Z",
		    "role": {
		      "id": 1,
		      "name": "Professor",
		      "description": "Professor",
		      "type": "authenticated",
		      "createdAt": "2022-11-04T17:17:18.144Z",
		      "updatedAt": "2022-11-04T18:16:50.094Z"
		    }
		  }
		}

Test Case ID: 1e	**Test Priority (L/M/H):** M
Test Title: Get List of Users - /users (get)	
Description: Serves all users in database	
Designed By: Tan Yunliang	**Test Designed Date:** 5/11/2022
Test Executed By: Tan Yunliang	**Test Executed Date:** 5/11/2022
Pre-conditions: 1. User Authentication JWT Token Provided / All Access Token Provided	
Dependencies: Nil	

Event Sequence:		
Input Event	**Expected Event**	**Output Event**
http://localhost:1337/api/users	**Server Responded with code**	**Server Responded with code**

is query with the following:	200	200:

curl -X 'GET' \
  'http://localhost:1337/api/users' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer c047f2498e4be5f8c3d8502d9e918f22a1ba3e7fb89d8fde41dc71dbd6b5f6b972177d8695ec81abd4cfadf0afdab03468f84a269b20692080795cf39f2a50913f8f6f68139802ad45384ce297ead1318d23a1c2569c0bacdf95580e92476f9637c91901bcc3ce168b6e37f6544f665f2a3e5f2b5139deeb941dab0dc867bb48'
		[
		  {
		    "id": 1,
		    "username": "professor1",
		    "email": "professor1@school.com",
		    "provider": "local",
		    "confirmed": true,
		    "blocked": false,
		    "createdAt": "2022-11-04T17:23:00.278Z",
		    "updatedAt": "2022-11-04T17:23:00.278Z"
		  },
		  {
		    "id": 2,
		    "username": "student1",
		    "email": "student1@school.com",
		    "provider": "local",
		    "confirmed": true,
		    "blocked": false,
		    "createdAt": "2022-11-04T17:25:25.550Z",
		    "updatedAt": "2022-11-04T17:25:25.550Z"
		  },
		  {
		    "id": 3,
		    "username": "professor2",
		    "email": "professor2@school.com",
		    "provider": "local",
		    "confirmed": false,
		    "blocked": false,
		    "createdAt": "2022-11-04T17:25:58.415Z",
		    "updatedAt": "2022-11-04T17:27:41.078Z"
		  },
		  {
		    "id": 4,
		    "username": "student2",
		    "email": "student2@school.com",
		    "provider": "local",
		    "confirmed": false,
		    "blocked": false,
		    "createdAt": "2022-11-04T17:28:09.958Z",
		    "updatedAt": "2022-11-04T17:28:09.958Z"
		  },
		  {
		    "id": 5,
		    "username": "professor3",
		    "email":

"professor3@school.com",
    "provider": "local",
    "confirmed": true,
    "blocked": true,
    "createdAt":
"2022-11-04T17:28:51.253Z",
    "updatedAt":
"2022-11-04T17:28:51.253Z"
  },
  {
    "id": 6,
    "username": "student3",
    "email":
"student3@school.com",
    "provider": "local",
    "confirmed": true,
    "blocked": true,
    "createdAt":
"2022-11-04T17:29:16.901Z",
    "updatedAt":
"2022-11-04T17:29:16.901Z"
  }
]

Test Case ID: 1f		Test Priority (L/M/H): H
Test Title: Get Current LoggedIn User - /users/me (get)		
Description: Using JWT Token to get current user data		
Designed By: Tan Yunliang		**Test Designed Date:** 5/11/2022
Test Executed By: Tan Yunliang		**Test Executed Date:** 5/11/2022
Pre-conditions: 1. User Authentication JWT Token Provided		
Dependencies: 1. Testcase 1a for JWT Token		

Event Sequence:		
Input Event	**Expected Event**	**Output Event**

http://localhost:1337/api/users/me is query with the following:
 	
curl -X 'GET' \
 	
'http://localhost:1337/api/users/me' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MSwiaWF0IjoxNjY3NjI1MDc3LCJleHAiOjE2NzAyMTcwNzd9.8KzGS_nSKIIkaHh
	Server Responded with code 200	**Server Responded with code 200:**
		
		{
		  "id": 1,
		  "username": "professor1",
		  "email":
		"professor1@school.com",
		  "provider": "local",
		  "confirmed": true,
		  "blocked": false,
		  "createdAt":
		"2022-11-04T17:23:00.278Z",

JsTzkz9foNRGCDurRkDXHmK8T-pE'		"updatedAt": "2022-11-04T17:23:00.278Z" }

http://localhost:1337/api/users/me is query with the following:
 	
curl -X 'GET' \
 	
'http://localhost:1337/api/users/me' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer c047f2498e4be5f8c3d8502d9e918f22a1ba3e7fb89d8fde41dc71dbd6b5f6b972177d8695ec81abd4cfadf0afdab03468f84a269b20692080795cf39f2a50913f8f6f68139802ad45384ce297ead1318d23a1c2569c0bacdf95580e92476f9637c91901bcc3ce168b6e37f6544f665f2a3e5f2b5139deeb941dab0dc867bb48'
	Server Responded with code 401	**Server Responded with code 401:**
		
		{
		 "data": null,
		 "error": {
		  "status": 401,
		  "name": "UnauthorizedError",
		  "message": "Unauthorized",
		  "details": {}
		 }
		}

Testcase 2: Assignment

Test Case ID: 2a	**Test Priority (L/M/H):** M
Test Title: Get List of Assignment - /assignments (get)	
Description: Serves all Assignments in database	
Designed By: Tan Yunliang	**Test Designed Date:** 5/11/2022
Test Executed By: Tan Yunliang	**Test Executed Date:** 5/11/2022
Pre-conditions: 1. User Authentication JWT Token Provided / All Access Token Provided	
Dependencies: Nil	

Event Sequence:		
Input Event	**Expected Event**	**Output Event**

http://localhost:1337/api/assignment is query with the following:
 	
curl -X 'GET' \
 	
'http://localhost:1337/api/assignments' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer c047f2498e4be5f8c3d8502d9e918f22a1ba3e7fb89d8fde41dc71dbd6b5f6b972177d8695ec81abd4cfadf0afdab03468f84a269b20692080795cf39f2a50913f8f6f68139802ad45384ce297ead1318d23a1c2569c0bacdf95580e92476f9637c91901bcc3ce168b6e37f6544f665f2a3e5f2b5139deeb941dab0dc867bb48'
	Server Responded with code 200	**Server Responded with code 200:**
		
		{
		  "data": [
		    {
		      "id": 1,
		      "attributes": {
		        "level_completed": 1,
		        "createdAt": "2022-11-04T18:03:52.331Z",
		        "updatedAt": "2022-11-04T18:03:52.331Z"
		      }
		    }
		  ],
		  "meta": {
		    "pagination": {
		      "page": 1,
		      "pageSize": 25,
		      "pageCount": 1,
		      "total": 1
		    }
		  }
		}

Test Case ID: 2b	**Test Priority (L/M/H):** H
Test Title: Modify Assignment - /assignments/{id} (post/get/put/delete)	
Description: Create / Update / Delete Assignment Instances	
Designed By: Tan Yunliang	**Test Designed Date:** 5/11/2022
Test Executed By: Tan Yunliang	**Test Executed Date:** 5/11/2022
Pre-conditions:	

1. User Authentication JWT Token Provided / All Access Token Provided		

Dependencies: Nil

Event Sequence:

Input Event	Expected Event	Output Event

http://localhost:1337/api/assignment is query with the following:
 	
curl -X 'POST' \
 	
'http://localhost:1337/api/assignments' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer c047f2498e4be5f8c3d8502d9e918f22a1ba3e7fb89d8fde41dc71dbd6b5f6b972177d8695ec81abd4cfadf0afdab03468f84a269b2069
2080795cf39f2a50913f8f6f68139802ad45384ce297ead1318d23a1c2569c0bacdf95580e92476f9637c91901bcc3ce168b6e37f6544f665f2a3e5f2b5139deeb941dab0dc867bb48' \
  -H 'Content-Type: application/json' \
  -d '{
  "data": {
    "world": "1",
    "student": "1",
    "level_completed": 0,
"assignment_score_composition": "1"
  }
}'
	Server Responded with code 200	**Server Responded with code 200:**
		
		{
		  "data": {
		   "id": 4,
		    "attributes": {
		     "level_completed": 0,
		     "createdAt": "2022-11-05T06:09:32.958Z",
		     "updatedAt": "2022-11-05T06:09:32.958Z"
		    }
		  },
		  "meta": {}
		}

http://localhost:1337/api/assignment/1 is query with the following:
 	
curl -X 'GET' \
 	
'http://localhost:1337/api/assignments/1' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer c047f2498e4be5f8c3d8502d9e918f22a1ba3e7fb89d8fde41dc71dbd6b5f6b972177d8695ec81abd4cfadf0afdab03468f84a269b2069
2080795cf39f2a50913f8f6f68139802ad45384ce297ead1318d23a1c2569c0bacdf95580e92476f9637c91901bcc3ce168b6e37f6544f665f2a3e5f2b5139deeb941dab0d
	Server Responded with code 200	**Server Responded with code 200:**
		
		{
		  "data": {
		   "id": 1,
		    "attributes": {
		     "level_completed": 1,
		     "createdAt": "2022-11-04T18:03:52.331Z",
		     "updatedAt": "2022-11-04T18:03:52.331Z"
		    }
		  },
		  "meta": {}
		}

c867bb48'		

http://localhost:1337/api/assignment/4 is query with the following:
 	
curl -X 'PUT' \
 	
'http://localhost:1337/api/assignments/4' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer c047f2498e4be5f8c3d8502d9e918f22a1ba3e7fb89d8fde41dc71dbd6b5f6b972177d8695ec81abd4cfadf0afdab03468f84a269b20692080795cf39f2a50913f8f6f68139802ad45384ce297ead1318d23a1c2569c0bacdf95580e92476f9637c91901bcc3ce168b6e37f6544f665f2a3e5f2b5139deeb941dab0dc867bb48' \
  -H 'Content-Type: application/json' \
  -d '{
  "data": {
    "world": "1",
    "student": "1",
    "level_completed": 1,
 	
"assignment_score_composition": "1"
  }
}'
	Server Responded with code 200	**Server Responded with code 200:**
		{
		  "data": {
		    "id": 4,
		    "attributes": {
		      "level_completed": 1,
		      "createdAt": "2022-11-05T06:09:32.958Z",
		      "updatedAt": "2022-11-05T06:14:38.151Z"
		    }
		  },
		  "meta": {}
		}

http://localhost:1337/api/assignment/4 is query with the following:
 	
curl -X 'DELETE' \
 	
'http://localhost:1337/api/assignments/4' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer c047f2498e4be5f8c3d8502d9e918f22a1ba3e7fb89d8fde41dc71dbd6b5f6b972177d8695ec81abd4cfadf0afdab03468f84a269b20692080795cf39f2a50913f8f6f68139802ad45384ce297ead1318d23a1c2569c0bacdf95580e92476f9637c91901bcc3ce168b6e37f6544f665f2a3e5f2b5139deeb941dab0dc867bb48'
	Server Responded with code 200	**Server Responded with code 200:**
		{
		  "data": {
		    "id": 4,
		    "attributes": {
		      "level_completed": 1,
		      "createdAt": "2022-11-05T06:09:32.958Z",
		      "updatedAt": "2022-11-05T06:14:38.151Z"
		    }
		  },
		  "meta": {}
		}

Testcase 3 : Assignment-score-composition

Test Case ID: 3a	Test Priority (L/M/H): M
Test Title: Get List of Assignment-score-composition - /assignment-score-compositions (get)	
Description: Serves all Assignment-score-compositions in database	
Designed By: Tan Yunliang	Test Designed Date: 5/11/2022
Test Executed By: Tan Yunliang	Test Executed Date: 5/11/2022
Pre-conditions: 1. User Authentication JWT Token Provided / All Access Token Provided	
Dependencies: Nil	

Event Sequence:		
Input Event	**Expected Event**	**Output Event**

http://localhost:1337/api/assignment is query with the following:
 	
curl -X 'GET' \
 	
'http://localhost:1337/api/assignment-score-compositions' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer c047f2498e4be5f8c3d8502d9e918f22a1ba3e7fb89d8fde41dc71dbd6b5f6b972177d8695ec81abd4cfadf0afdab03468f84a269b2069 2080795cf39f2a50913f8f6f68139 802ad45384ce297ead1318d23a 1c2569c0bacdf95580e92476f963 7c91901bcc3ce168b6e37f6544f6 65f2a3e5f2b5139deeb941dab0d c867bb48'
	Server Responded with code 200	**Server Responded with code 200:**
		
		{
		  "data": [
		    {
		      "id": 1,
		      "attributes": {
		       "createdAt": "2022-11-04T18:04:38.247Z",
		        "updatedAt": "2022-11-04T18:04:38.247Z"
		      }
		    }
		  ],
		  "meta": {
		    "pagination": {
		      "page": 1,
		      "pageSize": 25,
		      "pageCount": 1,
		      "total": 1
		    }
		  }
		}

Test Case ID: 3b	Test Priority (L/M/H): H
Test Title: Modify Assignment-score-composition - /assignment-score-compositions/{id} (post/get/put/delete)	
Description: Create / Update / Delete Assignment-score-composition Instances	
Designed By: Tan Yunliang	Test Designed Date: 5/11/2022
Test Executed By: Tan Yunliang	Test Executed Date: 5/11/2022

Pre-conditions:
1. User Authentication JWT Token Provided / All Access Token Provided

Dependencies: Nil

Event Sequence:

Input Event	Expected Event	Output Event

http://localhost:1337/api/assignment-score-compositions is query with the following:
 	
curl -X 'POST' \
 	
'http://localhost:1337/api/assignment-score-compositions' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer c047f2498e4be5f8c3d8502d9e918f22a1ba3e7fb89d8fde41dc71dbd6b5f6b972177d8695ec81abd4cfadf0afdab03468f84a269b2069 2080795cf39f2a50913f8f6f68139 802ad45384ce297ead1318d23a 1c2569c0bacdf95580e92476f963 7c91901bcc3ce168b6e37f6544f6 65f2a3e5f2b5139deeb941dab0d c867bb48' \
  -H 'Content-Type: application/json' \
  -d '{
  "data": {
    "assignment": "1",
    "scores": [
     "1",
     "2"
    ]
  }
}'
	Server Responded with code 200	**Server Responded with code 200:**
		
		{
		  "data": {
		   "id": 2,
		   "attributes": {
		    "createdAt": "2022-11-05T08:08:37.709Z",
		    "updatedAt": "2022-11-05T08:08:37.709Z"
		   }
		  },
		  "meta": {}
		}

http://localhost:1337/api/assignment-score-compositions/2 is query with the following:
 	
curl -X 'GET' \
 	
'http://localhost:1337/api/assignment-score-compositions/2' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer c047f2498e4be5f8c3d8502d9e918f22a1ba3e7fb89d8fde41dc71dbd6b5f6b972177d8695ec81abd4cfadf0afdab03468f84a269b2069 2080795cf39f2a50913f8f6f68139 802ad45384ce297ead1318d23a 1c2569c0bacdf95580e92476f963 7c91901bcc3ce168b6e37f6544f6
	Server Responded with code 200	**Server Responded with code 200:**
		
		{
		  "data": {
		   "id": 2,
		   "attributes": {
		    "createdAt": "2022-11-05T08:08:37.709Z",
		    "updatedAt": "2022-11-05T08:08:37.709Z"
		   }
		  },
		  "meta": {}
		}

65f2a3e5f2b5139deeb941dab0d c867bb48'		

http://localhost:1337/api/assig nment-score-compositions/2 is query with the following:
 	
curl -X 'PUT' \
 	
'http://localhost:1337/api/assignm ent-score-compositions/2' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer c047f2498e4be5f8c3d8502d9e9 18f22a1ba3e7fb89d8fde41dc71d bd6b5f6b972177d8695ec81abd4 cfadf0afdab03468f84a269b2069 2080795cf39f2a50913f8f6f68139 802ad45384ce297ead1318d23a 1c2569c0bacdf95580e92476f963 7c91901bcc3ce168b6e37f6544f6 65f2a3e5f2b5139deeb941dab0d c867bb48' \
  -H 'Content-Type: application/json' \
  -d '{
  "data": {
    "scores": [
      "1"
    ]
  }
}'
	Server Responded with code 200	**Server Responded with code 200:**
		{
		  "data": {
		   "id": 2,
		   "attributes": {
		     "createdAt": "2022-11-05T08:08:37.709Z",
		     "updatedAt": "2022-11-05T08:10:30.906Z"
		   }
		  },
		  "meta": {}
		}

http://localhost:1337/api/assig nment-score-compositions/2 is query with the following:
 	
curl -X 'DELETE' \
 	
'http://localhost:1337/api/assignm ent-score-compositions/2' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer c047f2498e4be5f8c3d8502d9e9 18f22a1ba3e7fb89d8fde41dc71d bd6b5f6b972177d8695ec81abd4 cfadf0afdab03468f84a269b2069 2080795cf39f2a50913f8f6f68139 802ad45384ce297ead1318d23a 1c2569c0bacdf95580e92476f963 7c91901bcc3ce168b6e37f6544f6 65f2a3e5f2b5139deeb941dab0d c867bb48'
	Server Responded with code 200	**Server Responded with code 200:**
		{
		  "data": {
		   "id": 2,
		   "attributes": {
		     "createdAt": "2022-11-05T08:08:37.709Z",
		     "updatedAt": "2022-11-05T08:10:30.906Z"
		   }
		  },
		  "meta": {}
		}

Testcase 4 : Level

Test Case ID: 4a	Test Priority (L/M/H): M
Test Title: Get List of Level - /levels (get)	
Description: Serves all Levels in database	
Designed By: Tan Yunliang	Test Designed Date: 5/11/2022
Test Executed By: Tan Yunliang	Test Executed Date: 5/11/2022
Pre-conditions: 1. User Authentication JWT Token Provided / All Access Token Provided	
Dependencies: Nil	

Event Sequence:		
Input Event	**Expected Event**	**Output Event**

http://localhost:1337/api/levels **assignment is query with the following:**
 	
curl -X 'GET' \
  'http://localhost:1337/api/levels' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer c047f2498e4be5f8c3d8502d9e918f22a1ba3e7fb89d8fde41dc71dbd6b5f6b972177d8695ec81abd4cfadf0afdab03468f84a269b20692080795cf39f2a50913f8f6f68139802ad45384ce297ead1318d23a1c2569c0bacdf95580e92476f9637c91901bcc3ce168b6e37f6544f665f2a3e5f2b5139deeb941dab0dc867bb48'
	Server Responded with code 200	**Server Responded with code 200:**
		
		{
		  "data": [
		   {
		    "id": 1,
		    "attributes": {
		     "name": "(world 1 - level 1)",
		     "description": "Mathematic Level",
		     "question_count": 3,
		     "level_type": "Normal",
		     "createdAt": "2022-11-04T17:54:16.193Z",
		     "updatedAt": "2022-11-04T17:54:16.193Z"
		    }
		   },
		   {
		    "id": 2,
		    "attributes": {
		     "name": "(world 1 - level 2)",
		     "description": "Mathematic Level",
		     "question_count": 3,
		     "level_type": "Normal",
		     "createdAt": "2022-11-04T17:54:40.703Z",
		     "updatedAt": "2022-11-04T17:54:40.703Z"
		    }
		   },
		   {
		    "id": 3,
		    "attributes": {

          "name": "(world 1 - level 3)",
          "description": "Mathematic Level",
          "question_count": 3,
          "level_type": "Normal",
          "createdAt": "2022-11-04T17:55:03.729Z",
          "updatedAt": "2022-11-04T17:55:03.729Z"
        }
      }
    ],
    "meta": {
      "pagination": {
        "page": 1,
        "pageSize": 25,
        "pageCount": 1,
        "total": 3
      }
    }
  }
}

Test Case ID: 4b	Test Priority (L/M/H): H
Test Title: Modify Level - /levels/{id} (post/get/put/delete)	
Description: Create / Update / Delete Level Instances	
Designed By: Tan Yunliang	Test Designed Date: 5/11/2022
Test Executed By: Tan Yunliang	Test Executed Date: 5/11/2022
Pre-conditions: 1. User Authentication JWT Token Provided / All Access Token Provided	
Dependencies: Nil	

Event Sequence:		
Input Event	**Expected Event**	**Output Event**

http://localhost:1337/api/levels is query with the following:
 	
 curl -X 'POST' \
  'http://localhost:1337/api/levels' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer c047f2498e4be5f8c3d8502d9e918f22a1ba3e7fb89d8fde41dc71dbd6b5f6b972177d8695ec81abd4cfadf0afdab03468f84a269b20692080795cf39f2a50913f8f6f68139802ad45384ce297ead1318d23a1c2569c0bacdf95580e92476f9637c91901bcc3ce168b6e37f6544f6
	Server Responded with code 200	**Server Responded with code 200:**
		
		 {
		  "data": {
		   "id": 5,
		   "attributes": {
		   "name": "(world 1 - level 1)",
		    "description": "Mathematic Level",
		    "question_count": 3,
		    "level_type": "Normal",
		    "createdAt": "2022-11-05T06:42:18.477Z",
		    "updatedAt": "2022-11-05T06:42:18.477Z"

65f2a3e5f2b5139deeb941dab0d
c867bb48' \
  -H 'Content-Type:
application/json' \
 -d '{
  "data": {
      "name" : "(world 1 - level
1)",
      "description" :
"Mathematic Level",
      "question_count" : 3,
      "level_type": "Normal",
      "creator": "1"
  }
}'
		}
		 },
		 "meta": {}
		}

**http://localhost:1337/api/levels/
5 is query with the following:**
 	
curl -X 'GET' \
 	
'http://localhost:1337/api/levels/5'
\
  -H 'accept: application/json' \
  -H 'Authorization: Bearer
c047f2498e4be5f8c3d8502d9e9
18f22a1ba3e7fb89d8fde41dc71d
bd6b5f6b972177d8695ec81abd4
cfadf0afdab03468f84a269b2069
2080795cf39f2a50913f8f6f68139
802ad45384ce297ead1318d23a
1c2569c0bacdf95580e92476f963
7c91901bcc3ce168b6e37f6544f6
65f2a3e5f2b5139deeb941dab0d
c867bb48'
	**Server Responded with code
	200**
		**Server Responded with code
		200:**
		
		{
		  "data": {
		   "id": 5,
		   "attributes": {
		   "name": "(world 1 - level 1)",
		   "description": "Mathematic
		Level",
		   "question_count": 3,
		   "level_type": "Normal",
		   "createdAt":
		"2022-11-05T06:42:18.477Z",
		   "updatedAt":
		"2022-11-05T06:42:18.477Z"
		   }
		 },
		 "meta": {}
		}

**http://localhost:1337/api/levels/
5 is query with the following:**
 	
curl -X 'PUT' \
 	
'http://localhost:1337/api/levels/5'
\
  -H 'accept: application/json' \
  -H 'Authorization: Bearer
c047f2498e4be5f8c3d8502d9e9
18f22a1ba3e7fb89d8fde41dc71d
bd6b5f6b972177d8695ec81abd4
cfadf0afdab03468f84a269b2069
2080795cf39f2a50913f8f6f68139
802ad45384ce297ead1318d23a
1c2569c0bacdf95580e92476f963
7c91901bcc3ce168b6e37f6544f6
65f2a3e5f2b5139deeb941dab0d
c867bb48' \
  -H 'Content-Type:
application/json' \
  -d '{
	**Server Responded with code
	200**
		**Server Responded with code
		200:**
		{
		  "data": {
		   "id": 5,
		   "attributes": {
		   "name": "(world 1 - level 1)",
		   "description": "Mathematic
		Level",
		   "question_count": 3,
		   "level_type": "Custom",
		   "createdAt":
		"2022-11-05T06:42:18.477Z",
		   "updatedAt":
		"2022-11-05T06:48:32.083Z"
		   }
		 },
		 "meta": {}
		}

``
  "data": {
    "level_type": "Custom"
  }
}'
``		

http://localhost:1337/api/assignment/4 is query with the following:
 	
curl -X 'DELETE' \
 	
'http://localhost:1337/api/levels/5' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer c047f2498e4be5f8c3d8502d9e918f22a1ba3e7fb89d8fde41dc71dbd6b5f6b972177d8695ec81abd4cfadf0afdab03468f84a269b20692080795cf39f2a50913f8f6f68139802ad45384ce297ead1318d23a1c2569c0bacdf95580e92476f9637c91901bcc3ce168b6e37f6544f665f2a3e5f2b5139deeb941dab0dc867bb48'
	Server Responded with code 200	**Server Responded with code 200:**
		``
		{
		  "data": {
		    "id": 5,
		    "attributes": {
		      "name": "(world 1 - level 1)",
		      "description": "Mathematic Level",
		      "question_count": 3,
		      "level_type": "Custom",
		      "createdAt": "2022-11-05T06:42:18.477Z",
		      "updatedAt": "2022-11-05T06:48:32.083Z"
		    }
		  },
		  "meta": {}
		}
		``

Testcase 5 : Level-question–composition

Test Case ID: 5a	Test Priority (L/M/H): M
Test Title: Get List of Level-question–composition - /level-question–compositions (get)	
Description: Serves all Level-question–compositions in database	
Designed By: Tan Yunliang	Test Designed Date: 5/11/2022
Test Executed By: Tan Yunliang	Test Executed Date: 5/11/2022
Pre-conditions: 1. User Authentication JWT Token Provided / All Access Token Provided	
Dependencies: Nil	

Event Sequence:		
Input Event	Expected Event	Output Event

http://localhost:1337/api/level-question-compositions is query with the following:
 	
curl -X 'GET' \
 	
'http://localhost:1337/api/level-question-compositions' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer c047f2498e4be5f8c3d8502d9e918f22a1ba3e7fb89d8fde41dc71dbd6b5f6b972177d8695ec81abd4cfadf0afdab03468f84a269b20692080795cf39f2a50913f8f6f68139802ad45384ce297ead1318d23a1c2569c0bacdf95580e92476f9637c91901bcc3ce168b6e37f6544f665f2a3e5f2b5139deeb941dab0dc867bb48'
Request URL
	Server Responded with code 200	**Server Responded with code 200:**
		
		{
		  "data": [
		    {
		      "id": 1,
		      "attributes": {
		        "question_position": 1,
		        "createdAt": "2022-11-04T17:55:15.914Z",
		        "updatedAt": "2022-11-04T17:55:15.914Z"
		      }
		    },
		    {
		      "id": 2,
		      "attributes": {
		        "question_position": 2,
		        "createdAt": "2022-11-04T17:55:24.955Z",
		        "updatedAt": "2022-11-04T17:55:24.955Z"
		      }
		    },
		    {
		      "id": 3,
		      "attributes": {
		        "question_position": 3,
		        "createdAt": "2022-11-04T17:56:57.688Z",
		        "updatedAt": "2022-11-04T17:56:57.688Z"
		      }
		    },
		    {
		      "id": 4,
		      "attributes": {

          "question_position": 1,
          "createdAt":
"2022-11-04T17:57:11.959Z",
          "updatedAt":
"2022-11-04T18:00:12.257Z"
        }
      },
      {
        "id": 5,
        "attributes": {
          "question_position": 2,
          "createdAt":
"2022-11-04T17:57:35.269Z",
          "updatedAt":
"2022-11-04T17:57:35.269Z"
        }
      },
      {
        "id": 6,
        "attributes": {
          "question_position": 3,
          "createdAt":
"2022-11-04T17:58:57.557Z",
          "updatedAt":
"2022-11-04T18:00:00.643Z"
        }
      },
      {
        "id": 7,
        "attributes": {
          "question_position": 1,
          "createdAt":
"2022-11-04T17:59:10.412Z",
          "updatedAt":
"2022-11-04T17:59:10.412Z"
        }
      },
      {
        "id": 8,
        "attributes": {
          "question_position": 2,
          "createdAt":
"2022-11-04T17:59:36.918Z",
          "updatedAt":
"2022-11-04T17:59:36.918Z"
        }
      },
      {
        "id": 9,
        "attributes": {
          "question_position": 3,
          "createdAt":
"2022-11-04T17:59:48.193Z",
          "updatedAt":
"2022-11-04T17:59:48.193Z"
        }
      }
    ],
    "meta": {

| | | "pagination": {
  "page": 1,
  "pageSize": 25,
  "pageCount": 1,
  "total": 9
  }
 }
} |

Test Case ID: 5b	Test Priority (L/M/H): H
Test Title: Modify Level-question–composition - /level-question–compositions/{id} (post/get/put/delete)	
Description: Create / Update / Delete Level-question–composition Instances	
Designed By: Tan Yunliang	**Test Designed Date:** 5/11/2022
Test Executed By: Tan Yunliang	**Test Executed Date:** 5/11/2022
Pre-conditions: 1. User Authentication JWT Token Provided / All Access Token Provided	
Dependencies: Nil	

Event Sequence:		
Input Event	**Expected Event**	**Output Event**

http://localhost:1337/api/level-question-compositions is query with the following:
 	
curl -X 'POST' \
 	
'http://localhost:1337/api/level-question-compositions' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer c047f2498e4be5f8c3d8502d9e918f22a1ba3e7fb89d8fde41dc71dbd6b5f6b972177d8695ec81abd4cfadf0afdab03468f84a269b20692080795cf39f2a50913f8f6f68139802ad45384ce297ead1318d23a1c2569c0bacdf95580e92476f9637c91901bcc3ce168b6e37f6544f665f2a3e5f2b5139deeb941dab0dc867bb48' \
  -H 'Content-Type: application/json' \
  -d '{
  "data": {
   "question": "1",
   "level": "1",
   "question_position": 0
  }
}'
	Server Responded with code 200	**Server Responded with code 200:**
		
		{
		  "data": {
		  "id": 10,
		   "attributes": {
		    "question_position": 0,
		    "createdAt": "2022-11-05T07:48:18.477Z",
		    "updatedAt": "2022-11-05T07:48:18.477Z"
		   }
		  },
		  "meta": {}
		}

http://localhost:1337/api/level-question-compositions/10 is query with the following:
 	
curl -X 'GET' \
 	
'http://localhost:1337/api/level-question-compositions/10' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer c047f2498e4be5f8c3d8502d9e918f22a1ba3e7fb89d8fde41dc71dbd6b5f6b972177d8695ec81abd4cfadf0afdab03468f84a269b20692080795cf39f2a50913f8f6f68139802ad45384ce297ead1318d23a1c2569c0bacdf95580e92476f9637c91901bcc3ce168b6e37f6544f665f2a3e5f2b5139deeb941dab0dc867bb48'
	Server Responded with code 200	**Server Responded with code 200:**
		
		{
		  "data": {
		   "id": 10,
		   "attributes": {
		    "question_position": 0,
		    "createdAt": "2022-11-05T07:48:18.477Z",
		    "updatedAt": "2022-11-05T07:48:18.477Z"
		   }
		  },
		  "meta": {}
		}

http://localhost:1337/api/level-question-compositions/10 is query with the following:
 	
curl -X 'PUT' \
 	
'http://localhost:1337/api/level-question-compositions/10' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer c047f2498e4be5f8c3d8502d9e918f22a1ba3e7fb89d8fde41dc71dbd6b5f6b972177d8695ec81abd4cfadf0afdab03468f84a269b20692080795cf39f2a50913f8f6f68139802ad45384ce297ead1318d23a1c2569c0bacdf95580e92476f9637c91901bcc3ce168b6e37f6544f665f2a3e5f2b5139deeb941dab0dc867bb48' \
  -H 'Content-Type: application/json' \
  -d '{
  "data": {
   "question": "2",
   "level": "1",
   "question_position": 1
  }
}'
	Server Responded with code 200	**Server Responded with code 200:**
		
		{
		  "data": {
		   "id": 10,
		   "attributes": {
		    "question_position": 1,
		    "createdAt": "2022-11-05T07:48:18.477Z",
		    "updatedAt": "2022-11-05T07:50:58.829Z"
		   }
		  },
		  "meta": {}
		}

http://localhost:1337/api/level-question-compositions/10 is query with the following:
 	
curl -X 'DELETE' \
 	
'http://localhost:1337/api/level-qu
	Server Responded with code 200	**Server Responded with code 200:**
		
		{
		  "data": {
		   "id": 10,
		   "attributes": {

estion-compositions/10' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer
c047f2498e4be5f8c3d8502d9e9
18f22a1ba3e7fb89d8fde41dc71d
bd6b5f6b972177d8695ec81abd4
cfadf0afdab03468f84a269b2069
2080795cf39f2a50913f8f6f68139
802ad45384ce297ead1318d23a
1c2569c0bacdf95580e92476f963
7c91901bcc3ce168b6e37f6544f6
65f2a3e5f2b5139deeb941dab0d
c867bb48'

   "question_position": 1,
   "createdAt":
"2022-11-05T07:48:18.477Z",
   "updatedAt":
"2022-11-05T07:50:58.829Z"
   }
  },
  "meta": {}
}

Testcase 6 : Question

Test Case ID: 6a	**Test Priority (L/M/H):** M
Test Title: Get List of Question - /questions (get)	
Description: Serves all Questions in database	
Designed By: Tan Yunliang	**Test Designed Date:** 5/11/2022
Test Executed By: Tan Yunliang	**Test Executed Date:** 5/11/2022
Pre-conditions: 1. User Authentication JWT Token Provided / All Access Token Provided	
Dependencies: Nil	

Event Sequence:		
Input Event	**Expected Event**	**Output Event**

http://localhost:1337/api/questionsassignment is query with the following:
 	
curl -X 'GET' \
 	
'http://localhost:1337/api/questions' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer c047f2498e4be5f8c3d8502d9e918f22a1ba3e7fb89d8fde41dc71dbd6b5f6b972177d8695ec81abd4cfadf0afdab03468f84a269b20692080795cf39f2a50913f8f6f68139802ad45384ce297ead1318d23a1c2569c0bacdf95580e92476f9637c91901bcc3ce168b6e37f6544f665f2a3e5f2b5139deeb941dab0dc867bb48'
	Server Responded with code 200	**Server Responded with code 200:**
		
		{
		  "data": [
		   {
		    "id": 1,
		    "attributes": {
		     "name": "Addition Question (world 1 - level 1 - question 1)",
		     "description": "What is 2+1 ?",
		     "fake_answers": "2,3,4,5",
		     "actual_answer": "B",
		     "createdAt": "2022-11-04T17:30:48.337Z",
		     "updatedAt": "2022-11-04T17:30:48.337Z"
		    }
		   },
		   {
		    "id": 2,
		    "attributes": {
		     "name": "Square Root Question (world 1 - level 1 - question 2)",
		     "description": "What is the square root of 16 ?",
		     "fake_answers": "2,3,4,5",
		     "actual_answer": "C",
		     "createdAt": "2022-11-04T17:31:20.201Z",
		     "updatedAt": "2022-11-04T17:31:20.201Z"
		    }
		   },
		   {
		    "id": 3,

      "attributes": {
        "name": "Prime Number Question (world 1 - level 1 - question 3)",
        "description": "What is Not a prime number ?",
        "fake_answers": "2,1,3,13",
        "actual_answer": "B",
        "createdAt": "2022-11-04T17:31:53.810Z",
        "updatedAt": "2022-11-04T17:31:53.810Z"
      }
    },
    {
      "id": 4,
      "attributes": {
        "name": "Division Question (world 1 - level 2 - question 1)",
        "description": "What is 81/9 ?",
        "fake_answers": "42,36,9,72",
        "actual_answer": "C",
        "createdAt": "2022-11-04T17:32:20.864Z",
        "updatedAt": "2022-11-04T17:32:20.864Z"
      }
    },
    {
      "id": 5,
      "attributes": {
        "name": "Multiplication Question (world 1 - level 2 - question 2)",
        "description": "What is 9*2 ?",
        "fake_answers": "2,9,27,18",
        "actual_answer": "D",
        "createdAt": "2022-11-04T17:32:50.709Z",
        "updatedAt": "2022-11-04T17:32:50.709Z"
      }
    },
    {
      "id": 6,
      "attributes": {
        "name": "Square Root Question (world 1 - level 2 - question 3)",
        "description": "What is square root of 4 ?",
        "fake_answers": "2,3,4,5",
        "actual_answer": "A",
        "createdAt": "2022-11-04T17:33:13.155Z",
        "updatedAt":

		"2022-11-04T17:33:13.155Z"
		    }
		  },
		  {
		    "id": 7,
		    "attributes": {
		     "name": "Addition Question (world 1 - level 3 - question 1)",
		     "description": "What is 2+2 ?",
		     "fake_answers": "4,6,7,8",
		     "actual_answer": "A",
		     "createdAt": "2022-11-04T17:51:46.961Z",
		     "updatedAt": "2022-11-04T17:51:54.966Z"
		    }
		  },
		  {
		    "id": 8,
		    "attributes": {
		     "name": "Square Root Question (world 1 - level 3 - question 2)",
		     "description": "What is the square root of 25?",
		     "fake_answers": "2,3,4,5",
		     "actual_answer": "D",
		     "createdAt": "2022-11-04T17:52:25.156Z",
		     "updatedAt": "2022-11-04T17:52:25.156Z"
		    }
		  },
		  {
		    "id": 9,
		    "attributes": {
		     "name": "Compound Question (world 1 - level 3 - question 3)",
		     "description": "What is 36 + 9/3 ?",
		     "fake_answers": "36,9,39,27",
		     "actual_answer": "C",
		     "createdAt": "2022-11-04T17:52:58.215Z",
		     "updatedAt": "2022-11-04T17:52:58.215Z"
		    }
		  }
		],
		 "meta": {
		  "pagination": {
		   "page": 1,
		   "pageSize": 25,
		   "pageCount": 1,
		   "total": 9
		  }

		} }

Test Case ID: 6b	**Test Priority (L/M/H):** H

Test Title: Modify Question /questions/{id} (post/get/put/delete)

Description: Create / Update / Delete Question Instances

Designed By: Tan Yunliang	**Test Designed Date:** 5/11/2022

Test Executed By: Tan Yunliang	**Test Executed Date:** 5/11/2022

Pre-conditions: 1. User Authentication JWT Token Provided / All Access Token Provided

Dependencies: Nil

Event Sequence:

Input Event	Expected Event	Output Event

http://localhost:1337/api/questions is query with the following:
 **curl -X 'POST' **
 	
 'http://localhost:1337/api/questions' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer c047f2498e4be5f8c3d8502d9e918f22a1ba3e7fb89d8fde41dc71dbd6b5f6b972177d8695ec81abd4cfadf0afdab03468f84a269b20692080795cf39f2a50913f8f6f68139802ad45384ce297ead1318d23a1c2569c0bacdf95580e92476f9637c91901bcc3ce168b6e37f6544f665f2a3e5f2b5139deeb941dab0dc867bb48' \
  -H 'Content-Type: application/json' \
  -d '{
  "data": {
      "name" : "Addition Question (world 1 - level 1 - question 1)",
      "description" : "What is 2+1 ?",
      "fake_answers" : "2,3,4,5",
      "actual_answer": "B",
      "creator": "1"
 	
  }
 }
	Server Responded with code 200	**Server Responded with code 200:**
		
		 {
		  "data": {
		   "id": 11,
		   "attributes": {
		    "name": "Addition Question (world 1 - level 1 - question 1)",
		    "description": "What is 2+1 ?",
		    "fake_answers": "2,3,4,5",
		    "actual_answer": "B",
		    "createdAt": "2022-11-05T06:59:02.093Z",
		    "updatedAt": "2022-11-05T06:59:02.093Z"
		   }
		  },
		  "meta": {}
		 }

http://localhost:1337/api/questions/11 is query with the following:
 	
curl -X 'GET' \
 	
'http://localhost:1337/api/questions/11' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer c047f2498e4be5f8c3d8502d9e918f22a1ba3e7fb89d8fde41dc71dbd6b5f6b972177d8695ec81abd4cfadf0afdab03468f84a269b2069 2080795cf39f2a50913f8f6f68139 802ad45384ce297ead1318d23a 1c2569c0bacdf95580e92476f963 7c91901bcc3ce168b6e37f6544f6 65f2a3e5f2b5139deeb941dab0d c867bb48'
	Server Responded with code 200	**Server Responded with code 200:**
		
		{
		  "data": {
		   "id": 11,
		   "attributes": {
		    "name": "Addition Question (world 1 - level 1 - question 1)",
		    "description": "What is 2+1 ?",
		    "fake_answers": "2,3,4,5",
		    "actual_answer": "B",
		    "createdAt": "2022-11-05T06:59:02.093Z",
		    "updatedAt": "2022-11-05T06:59:02.093Z"
		   }
		  },
		  "meta": {}
		}

http://localhost:1337/api/questions/11 is query with the following:
 	
curl -X 'PUT' \
 	
'http://localhost:1337/api/questions/11' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer c047f2498e4be5f8c3d8502d9e918f22a1ba3e7fb89d8fde41dc71dbd6b5f6b972177d8695ec81abd4cfadf0afdab03468f84a269b2069 2080795cf39f2a50913f8f6f68139 802ad45384ce297ead1318d23a 1c2569c0bacdf95580e92476f963 7c91901bcc3ce168b6e37f6544f6 65f2a3e5f2b5139deeb941dab0d c867bb48' \
  -H 'Content-Type: application/json' \
  -d '{
  "data": {
    "actual_answer": "A"
  }
}'
	Server Responded with code 200	**Server Responded with code 200:**
		
		{
		  "data": {
		   "id": 11,
		   "attributes": {
		    "name": "Addition Question (world 1 - level 1 - question 1)",
		    "description": "What is 2+1 ?",
		    "fake_answers": "2,3,4,5",
		    "actual_answer": "A",
		    "createdAt": "2022-11-05T06:59:02.093Z",
		    "updatedAt": "2022-11-05T07:02:23.112Z"
		   }
		  },
		  "meta": {}
		}

http://localhost:1337/api/questions/11 is query with the following:
 	
curl -X 'DELETE' \
 	
'http://localhost:1337/api/questions/11' \
	Server Responded with code 200	**Server Responded with code 200:**
		
		{
		  "data": {
		   "id": 11,
		   "attributes": {
		    "name": "Addition Question

-H 'accept: application/json' \
  -H 'Authorization: Bearer
c047f2498e4be5f8c3d8502d9e9
18f22a1ba3e7fb89d8fde41dc71d
bd6b5f6b972177d8695ec81abd4
cfadf0afdab03468f84a269b2069
2080795cf39f2a50913f8f6f68139
802ad45384ce297ead1318d23a
1c2569c0bacdf95580e92476f963
7c91901bcc3ce168b6e37f6544f6
65f2a3e5f2b5139deeb941dab0d
c867bb48'
Request URL
		(world 1 - level 1 - question 1)",
		    "description": "What is 2+1
		?",
		    "fake_answers": "2,3,4,5",
		    "actual_answer": "A",
		    "createdAt":
		"2022-11-05T06:59:02.093Z",
		    "updatedAt":
		"2022-11-05T07:02:23.112Z"
		  }
		 },
		  "meta": {}
		}

Testcase 7 : Score

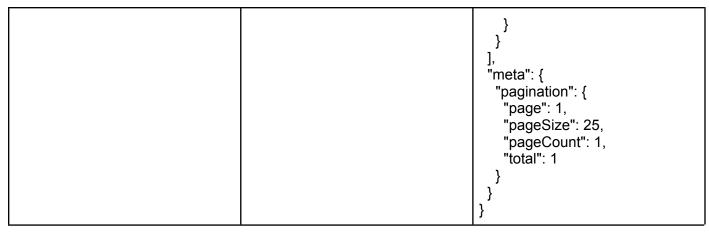| Test Case ID: 7a	Test Priority (L/M/H): M
Test Title: Get List of Score - /scores (get)	
Description: Serves all Scores in database	
Designed By: Tan Yunliang	Test Designed Date: 5/11/2022
Test Executed By: Tan Yunliang	Test Executed Date: 5/11/2022
Pre-conditions: 1. User Authentication JWT Token Provided / All Access Token Provided	
Dependencies: Nil	

Event Sequence:		
Input Event	**Expected Event**	**Output Event**

http://localhost:1337/api/scores **assignment is query with the following:**
 	
curl -X 'GET' \
  'http://localhost:1337/api/scores' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer c047f2498e4be5f8c3d8502d9e918f22a1ba3e7fb89d8fde41dc71dbd6b5f6b972177d8695ec81abd4cfadf0afdab03468f84a269b20692080795cf39f2a50913f8f6f68139802ad45384ce297ead1318d23a1c2569c0bacdf95580e92476f9637c91901bcc3ce168b6e37f6544f665f2a3e5f2b5139deeb941dab0dc867bb48'
	Server Responded with code 200	**Server Responded with code 200:**
		
		{
		  "data": [
		    {
		      "id": 1,
		      "attributes": {
		        "questions_attempted": 3,
		        "questions_correct": 3,
		        "createdAt": "2022-11-04T18:04:25.858Z",
		        "updatedAt": "2022-11-04T18:04:25.858Z"
		      }
		    },
		    {
		      "id": 2,
		      "attributes": {
		        "questions_attempted": 3,
		        "questions_correct": 3,
		        "createdAt": "2022-11-04T18:04:55.197Z",
		        "updatedAt": "2022-11-04T18:04:55.197Z"
		      }
		    }
		  ],
		  "meta": {
		    "pagination": {
		      "page": 1,
		      "pageSize": 25,
		      "pageCount": 1,
		      "total": 2
		    }
		  }
		}

Test Case ID: 7b		Test Priority (L/M/H): H
Test Title: Modify Score - /scores/{id} (post/get/put/delete)		
Description: Create / Update / Delete Score Instances		
Designed By: Tan Yunliang		**Test Designed Date:** 5/11/2022
Test Executed By: Tan Yunliang		**Test Executed Date:** 5/11/2022
Pre-conditions: 1. User Authentication JWT Token Provided / All Access Token Provided		
Dependencies: Nil		

Event Sequence:		
Input Event	**Expected Event**	**Output Event**

http://localhost:1337/api/scores is query with the following:
 	
curl -X 'POST' \
  'http://localhost:1337/api/scores' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer c047f2498e4be5f8c3d8502d9e918f22a1ba3e7fb89d8fde41dc71dbd6b5f6b972177d8695ec81abd4cfadf0afdab03468f84a269b20692080795cf39f2a50913f8f6f68139802ad45384ce297ead1318d23a1c2569c0bacdf95580e92476f9637c91901bcc3ce168b6e37f6544f665f2a3e5f2b5139deeb941dab0dc867bb48' \
  -H 'Content-Type: application/json' \
  -d '{
  "data": {
    "level": "1",
    "player": "1",
    "questions_attempted": 3,
    "questions_correct": 3,
 	
"assignment_score_composition": "1"
  }
}'
	Server Responded with code 200	**Server Responded with code 200:**
		
		{
		  "data": {
		    "id": 5,
		    "attributes": {
		      "questions_attempted": 3,
		      "questions_correct": 3,
		      "createdAt": "2022-11-05T07:11:38.409Z",
		      "updatedAt": "2022-11-05T07:11:38.409Z"
		    }
		  },
		  "meta": {}
		}

http://localhost:1337/api/scores/3 is query with the following:
 	
curl -X 'GET' \
 	
'http://localhost:1337/api/scores/3' \
  -H 'accept: application/json' \
	Server Responded with code 200	**Server Responded with code 200:**
		
		{
		  "data": {
		    "id": 3,
		    "attributes": {
		      "questions_attempted": 3,

-H 'Authorization: Bearer c047f2498e4be5f8c3d8502d9e9 18f22a1ba3e7fb89d8fde41dc71d bd6b5f6b972177d8695ec81abd4 cfadf0afdab03468f84a269b2069 2080795cf39f2a50913f8f6f68139 802ad45384ce297ead1318d23a 1c2569c0bacdf95580e92476f963 7c91901bcc3ce168b6e37f6544f6 65f2a3e5f2b5139deeb941dab0d c867bb48'		"questions_correct": 3, "createdAt": "2022-11-05T07:09:06.905Z", "updatedAt": "2022-11-05T07:09:06.905Z" } }, "meta": {} }

http://localhost:1337/api/score s/3 **is query with the following:**
 	
curl -X 'PUT' \
 	
'http://localhost:1337/api/scores/3 ' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer c047f2498e4be5f8c3d8502d9e9 18f22a1ba3e7fb89d8fde41dc71d bd6b5f6b972177d8695ec81abd4 cfadf0afdab03468f84a269b2069 2080795cf39f2a50913f8f6f68139 802ad45384ce297ead1318d23a 1c2569c0bacdf95580e92476f963 7c91901bcc3ce168b6e37f6544f6 65f2a3e5f2b5139deeb941dab0d c867bb48' \
  -H 'Content-Type: application/json' \
  -d '{
  "data": {
    "questions_attempted": 2,
  }
}'
	Server Responded with code 200	**Server Responded with code 200:**
		
		{
		  "data": {
		    "id": 3,
		    "attributes": {
		     "questions_attempted": 2,
		     "questions_correct": 3,
		     "createdAt": "2022-11-05T07:09:06.905Z",
		     "updatedAt": "2022-11-05T07:14:20.864Z"
		    }
		  },
		  "meta": {}
		}

http://localhost:1337/api/score s/3 **is query with the following:**
 	
curl -X 'DELETE' \
 	
'http://localhost:1337/api/scores/3 ' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer c047f2498e4be5f8c3d8502d9e9 18f22a1ba3e7fb89d8fde41dc71d bd6b5f6b972177d8695ec81abd4 cfadf0afdab03468f84a269b2069 2080795cf39f2a50913f8f6f68139 802ad45384ce297ead1318d23a 1c2569c0bacdf95580e92476f963 7c91901bcc3ce168b6e37f6544f6 65f2a3e5f2b5139deeb941dab0d
	Server Responded with code 200	**Server Responded with code 200:**
		
		{
		  "data": {
		    "id": 3,
		    "attributes": {
		     "questions_attempted": 2,
		     "questions_correct": 3,
		     "createdAt": "2022-11-05T07:09:06.905Z",
		     "updatedAt": "2022-11-05T07:14:20.864Z"
		    }
		  },
		  "meta": {}
		}

c867bb48'		

Test Case ID: 8a	Test Priority (L/M/H): M
Test Title: Get List of World - /worlds (get)	
Description: Serves all worlds in database	
Designed By: Tan Yunliang	**Test Designed Date:** 5/11/2022
Test Executed By: Tan Yunliang	**Test Executed Date:** 5/11/2022
Pre-conditions: 1. User Authentication JWT Token Provided / All Access Token Provided	
Dependencies: Nil	

Event Sequence:		
Input Event	**Expected Event**	**Output Event**

http://localhost:1337/api/worlds is query with the following:
 	
curl -X 'GET' \
  'http://localhost:1337/api/worlds' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer c047f2498e4be5f8c3d8502d9e918f22a1ba3e7fb89d8fde41dc71dbd6b5f6b972177d8695ec81abd4cfadf0afdab03468f84a269b20692080795cf39f2a50913f8f6f68139802ad45384ce297ead1318d23a1c2569c0bacdf95580e92476f9637c91901bcc3ce168b6e37f6544f665f2a3e5f2b5139deeb941dab0dc867bb48'
	Server Responded with code 200	**Server Responded with code 200:**
		
		{
		  "data": [
		    {
		      "id": 1,
		      "attributes": {
		        "Name": "CZ1060 Mathematic For Kids (world 1)",
		        "Description": "Dolor magna eget est lorem ipsum. Lacus sed viverra tellus in hac habitasse platea dictumst vestibulum. Faucibus pulvinar elementum integer enim neque volutpat ac tincidunt vitae. Purus gravida quis blandit turpis cursus in. Cursus vitae congue mauris rhoncus aenean. Quam viverra orci sagittis eu volutpat odio. Elementum nisi quis eleifend quam adipiscing. At volutpat diam ut venenatis tellus. Massa enim nec dui nunc mattis enim ut tellus. Phasellus faucibus scelerisque eleifend donec pretium vulputate sapien. Varius sit amet mattis vulputate enim. Libero volutpat sed cras ornare arcu. Faucibus vitae aliquet nec ullamcorper sit amet risus nullam eget.",
		        "Level_Count": 3,
		        "createdAt": "2022-11-04T17:22:14.685Z",
		        "updatedAt": "2022-11-04T17:22:14.685Z"

```
      }
     }
    ],
    "meta": {
     "pagination": {
      "page": 1,
      "pageSize": 25,
      "pageCount": 1,
      "total": 1
     }
    }
   }
```

Test Case ID: 8b	Test Priority (L/M/H): H
Test Title: Modify World - /worlds/{id} (post/get/put/delete)	
Description: Create / Update / Delete World Instances	
Designed By: Tan Yunliang	**Test Designed Date:** 5/11/2022
Test Executed By: Tan Yunliang	**Test Executed Date:** 5/11/2022
Pre-conditions: 1. User Authentication JWT Token Provided / All Access Token Provided	
Dependencies: Nil	

Event Sequence:		
Input Event	**Expected Event**	**Output Event**

http://localhost:1337/api/worlds is query with the following:
 	
curl -X 'POST' \
  'http://localhost:1337/api/worlds' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer c047f2498e4be5f8c3d8502d9e918f22a1ba3e7fb89d8fde41dc71dbd6b5f6b972177d8695ec81abd4cfadf0afdab03468f84a269b20692080795cf39f2a50913f8f6f68139802ad45384ce297ead1318d23a1c2569c0bacdf95580e92476f9637c91901bcc3ce168b6e37f6544f665f2a3e5f2b5139deeb941dab0dc867bb48' \
  -H 'Content-Type: application/json' \
  -d '{
  "data": {
    "Name": "CZ1060 Mathematic For Kids (world 1)",
    "Description": "Dolor magna eget est lorem ipsum. Lacus sed
	Server Responded with code 200	**Server Responded with code 200:**
		
		{
		  "data": {
		    "id": 2,
		    "attributes": {
		     "Name": "CZ1060 Mathematic For Kids (world 1)",
		     "Description": "Dolor magna eget est lorem ipsum. Lacus sed viverra tellus in hac habitasse platea dictumst vestibulum. Faucibus pulvinar elementum integer enim neque volutpat ac tincidunt vitae. Purus gravida quis blandit turpis cursus in. Cursus vitae congue mauris rhoncus aenean. Quam viverra orci sagittis eu volutpat odio. Elementum nisi quis eleifend quam adipiscing. At volutpat diam ut venenatis tellus. Massa enim nec dui nunc mattis enim ut tellus. Phasellus faucibus scelerisque eleifend donec

viverra tellus in hac habitasse platea dictumst vestibulum. Faucibus pulvinar elementum integer enim neque volutpat ac tincidunt vitae. Purus gravida quis blandit turpis cursus in. Cursus vitae congue mauris rhoncus aenean. Quam viverra orci sagittis eu volutpat odio. Elementum nisi quis eleifend quam adipiscing. At volutpat diam ut venenatis tellus. Massa enim nec dui nunc mattis enim ut tellus. Phasellus faucibus scelerisque eleifend donec pretium vulputate sapien. Varius sit amet mattis vulputate enim. Libero volutpat sed cras ornare arcu. Faucibus vitae aliquet nec ullamcorper sit amet risus nullam eget.",
 	
    "Level_Count": 3,
    "Creator": "1"
  }
}'
		pretium vulputate sapien. Varius sit amet mattis vulputate enim. Libero volutpat sed cras ornare arcu. Faucibus vitae aliquet nec ullamcorper sit amet risus nullam eget.",
		    "Level_Count": 3,
		    "createdAt": "2022-11-05T07:21:10.225Z",
		    "updatedAt": "2022-11-05T07:21:10.225Z"
		  }
		 },
		  "meta": {}
		}

http://localhost:1337/api/worlds/2 is query with the following:
 	
curl -X 'GET' \
 	
'http://localhost:1337/api/worlds/2' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer c047f2498e4be5f8c3d8502d9e918f22a1ba3e7fb89d8fde41dc71dbd6b5f6b972177d8695ec81abd4cfadf0afdab03468f84a269b20692080795cf39f2a50913f8f6f68139802ad45384ce297ead1318d23a1c2569c0bacdf95580e92476f9637c91901bcc3ce168b6e37f6544f665f2a3e5f2b5139deeb941dab0dc867bb48'
	Server Responded with code 200	**Server Responded with code 200:**
		
		{
		  "data": {
		   "id": 2,
		    "attributes": {
		     "Name": "CZ1060 Mathematic For Kids (world 1)",
		     "Description": "Dolor magna eget est lorem ipsum. Lacus sed viverra tellus in hac habitasse platea dictumst vestibulum. Faucibus pulvinar elementum integer enim neque volutpat ac tincidunt vitae. Purus gravida quis blandit turpis cursus in. Cursus vitae congue mauris rhoncus aenean. Quam viverra orci sagittis eu volutpat odio. Elementum nisi quis eleifend quam adipiscing. At volutpat diam ut venenatis tellus. Massa enim nec dui nunc mattis enim ut tellus. Phasellus faucibus scelerisque eleifend donec pretium vulputate sapien. Varius sit amet mattis vulputate enim. Libero volutpat sed cras ornare arcu. Faucibus vitae aliquet nec ullamcorper sit amet risus nullam eget.",
		     "Level_Count": 3,
		     "createdAt":

		"2022-11-05T07:21:10.225Z",
		   "updatedAt":
		"2022-11-05T07:21:10.225Z"
		  }
		 },
		 "meta": {}
		}

http://localhost:1337/api/worlds/2 is query with the following:
 	
curl -X 'PUT' \
 	
'http://localhost:1337/api/worlds/2' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer c047f2498e4be5f8c3d8502d9e918f22a1ba3e7fb89d8fde41dc71dbd6b5f6b972177d8695ec81abd4cfadf0afdab03468f84a269b2069 2080795cf39f2a50913f8f6f68139 802ad45384ce297ead1318d23a 1c2569c0bacdf95580e92476f963 7c91901bcc3ce168b6e37f6544f6 65f2a3e5f2b5139deeb941dab0d c867bb48' \
  -H 'Content-Type: application/json' \
  -d '{
  "data": {
    "Description": "Modified"
  }
}'
	Server Responded with code 200	**Server Responded with code 200:**
		
		{
		 "data": {
		  "id": 2,
		  "attributes": {
		   "Name": "CZ1060 Mathematic For Kids (world 1)",
		   "Description": "Modified",
		   "Level_Count": 3,
		   "createdAt": "2022-11-05T07:21:10.225Z",
		   "updatedAt": "2022-11-05T07:23:49.273Z"
		  }
		 },
		 "meta": {}
		}

http://localhost:1337/api/worlds/2 is query with the following:
 	
curl -X 'DELETE' \
 	
'http://localhost:1337/api/worlds/2' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer c047f2498e4be5f8c3d8502d9e918f22a1ba3e7fb89d8fde41dc71dbd6b5f6b972177d8695ec81abd4cfadf0afdab03468f84a269b2069 2080795cf39f2a50913f8f6f68139 802ad45384ce297ead1318d23a 1c2569c0bacdf95580e92476f963 7c91901bcc3ce168b6e37f6544f6 65f2a3e5f2b5139deeb941dab0d c867bb48'
	Server Responded with code 200	**Server Responded with code 200:**
		
		{
		 "data": {
		  "id": 2,
		  "attributes": {
		   "Name": "CZ1060 Mathematic For Kids (world 1)",
		   "Description": "Modified",
		   "Level_Count": 3,
		   "createdAt": "2022-11-05T07:21:10.225Z",
		   "updatedAt": "2022-11-05T07:23:49.273Z"
		  }
		 },
		 "meta": {}
		}

Testcase 9 : World-level-composition

Test Case ID: 9a	Test Priority (L/M/H): M
Test Title: Get List of World-level-composition - /world-level-compositions (get)	
Description: Serves all World-level-compositions in database	
Designed By: Tan Yunliang	Test Designed Date: 5/11/2022
Test Executed By: Tan Yunliang	Test Executed Date: 5/11/2022
Pre-conditions: 1. User Authentication JWT Token Provided / All Access Token Provided	
Dependencies: Nil	

Event Sequence:		
Input Event	**Expected Event**	**Output Event**

http://localhost:1337/api/world-level -compositions is query with the following:
 	
curl -X 'GET' \
 	
'http://localhost:1337/api/world-level-compositions' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer c047f2498e4be5f8c3d8502d9e918f22a1ba3e7fb89d8fde41dc71dbd6b5f6b972177d8695ec81abd4cfadf0afdab03468f84a269b20692080795cf39f2a50913f8f6f68139802ad45384ce297ead1318d23a1c2569c0bacdf95580e92476f9637c91901bcc3ce168b6e37f6544f665f2a3e5f2b5139deeb941dab0dc867bb48'
	Server Responded with code 200	**Server Responded with code 200:**
		
		{
		  "data": [
		    {
		      "id": 2,
		      "attributes": {
		       "level_position": 1,
		       "createdAt": "2022-11-05T07:39:21.174Z",
		       "updatedAt": "2022-11-05T07:39:21.174Z"
		      }
		    },
		    {
		      "id": 3,
		      "attributes": {
		       "level_position": 2,
		       "createdAt": "2022-11-05T07:39:35.091Z",
		       "updatedAt": "2022-11-05T07:39:35.091Z"
		      }
		    },
		    {
		      "id": 4,
		      "attributes": {
		       "level_position": 3,
		       "createdAt": "2022-11-05T07:39:45.049Z",
		       "updatedAt": "2022-11-05T07:39:45.049Z"
		      }
		    }
		  ],
		  "meta": {

```
                                                    "pagination": {
                                                      "page": 1,
                                                      "pageSize": 25,
                                                      "pageCount": 1,
                                                      "total": 3
                                                    }
                                                  }
                                                }
```

Test Case ID: 9b	Test Priority (L/M/H): H
Test Title: Modify World-level-composition - /world-level-compositions/{id} (post/get/put/delete)	
Description: Create / Update / Delete World-level-composition Instances	
Designed By: Tan Yunliang	**Test Designed Date:** 5/11/2022
Test Executed By: Tan Yunliang	**Test Executed Date:** 5/11/2022
Pre-conditions: 　　1.　User Authentication JWT Token Provided / All Access Token Provided	
Dependencies: Nil	

Event Sequence:		
Input Event	**Expected Event**	**Output Event**

http://localhost:1337/api/world-level-compositions is query with the following:
 	
curl -X 'POST' \
 	
'http://localhost:1337/api/world-level-compositions' \
　-H 'accept: application/json' \
　-H 'Authorization: Bearer c047f2498e4be5f8c3d8502d9e918f22a1ba3e7fb89d8fde41dc71dbd6b5f6b972177d8695ec81abd4cfadf0afdab03468f84a269b2069 2080795cf39f2a50913f8f6f68139 802ad45384ce297ead1318d23a 1c2569c0bacdf95580e92476f963 7c91901bcc3ce168b6e37f6544f6 65f2a3e5f2b5139deeb941dab0d c867bb48' \
　-H 'Content-Type: application/json' \
　-d '{
　"data": {
　　"level": "1",
　　"world": "1",
　　"level_position": 1
　}
}'
	Server Responded with code 200	**Server Responded with code 200:**
		
		{
		　"data": {
		　　"id": 5,
		　　"attributes": {
		　　　"level_position": 1,
		　　　"createdAt": "2022-11-05T07:41:39.234Z",
		　　　"updatedAt": "2022-11-05T07:41:39.234Z"
		　　}
		　},
		　"meta": {}
		}

http://localhost:1337/api/world-level-compositions/5 is query with the following:
 	
curl -X 'GET' \
 	
'http://localhost:1337/api/world-level-compositions/5' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer c047f2498e4be5f8c3d8502d9e918f22a1ba3e7fb89d8fde41dc71dbd6b5f6b972177d8695ec81abd4cfadf0afdab03468f84a269b20692080795cf39f2a50913f8f6f68139802ad45384ce297ead1318d23a1c2569c0bacdf95580e92476f9637c91901bcc3ce168b6e37f6544f665f2a3e5f2b5139deeb941dab0dc867bb48'
	Server Responded with code 200	**Server Responded with code 200:**
		
		{
		  "data": {
		   "id": 5,
		   "attributes": {
		    "level_position": 1,
		    "createdAt": "2022-11-05T07:41:39.234Z",
		    "updatedAt": "2022-11-05T07:41:39.234Z"
		   }
		  },
		  "meta": {}
		}

http://localhost:1337/api/world-level-compositions/5 is query with the following:
 	
curl -X 'PUT' \
 	
'http://localhost:1337/api/world-level-compositions/5' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer c047f2498e4be5f8c3d8502d9e918f22a1ba3e7fb89d8fde41dc71dbd6b5f6b972177d8695ec81abd4cfadf0afdab03468f84a269b20692080795cf39f2a50913f8f6f68139802ad45384ce297ead1318d23a1c2569c0bacdf95580e92476f9637c91901bcc3ce168b6e37f6544f665f2a3e5f2b5139deeb941dab0dc867bb48' \
  -H 'Content-Type: application/json' \
  -d '{
  "data": {
   "level_position": 2
  }
}'
	Server Responded with code 200	**Server Responded with code 200:**
		
		{
		  "data": {
		   "id": 5,
		   "attributes": {
		    "level_position": 2,
		    "createdAt": "2022-11-05T07:41:39.234Z",
		    "updatedAt": "2022-11-05T07:43:28.078Z"
		   }
		  },
		  "meta": {}
		}

http://localhost:1337/api/world-level-compositions/5 is query with the following:
 	
curl -X 'DELETE' \
 	
'http://localhost:1337/api/world-level-compositions/5' \
	Server Responded with code 200	**Server Responded with code 200:**
		
		{
		  "data": {
		   "id": 5,
		   "attributes": {
		    "level_position": 2,

-H 'accept: application/json' \\ -H 'Authorization: Bearer c047f2498e4be5f8c3d8502d9e9 18f22a1ba3e7fb89d8fde41dc71d bd6b5f6b972177d8695ec81abd4 cfadf0afdab03468f84a269b2069 2080795cf39f2a50913f8f6f68139 802ad45384ce297ead1318d23a 1c2569c0bacdf95580e92476f963 7c91901bcc3ce168b6e37f6544f6 65f2a3e5f2b5139deeb941dab0d c867bb48'		"createdAt": "2022-11-05T07:41:39.234Z", "updatedAt": "2022-11-05T07:43:28.078Z" } }, "meta": {} }

Non-Functional Testing

Non Functional Requirements	Testing Results
The system shall have a response time of less than 10 seconds after every user action	Testing is done on 10 different computers with different specifications. The maximum response time is 4 seconds, which is comfortably below 10 seconds.
The system must offer informative feedback	During testing, it can be seen that feedback was provided for every action user takes, like highlighting area clicked, alerts shown, etc
The system must display error messages that allows the user to know what went wrong when an error occurs	During testing, it can be seen that an alert is shown for every wrong action user has taken which results in an invalid outcome.
The system must allow easy reversal of the user's actions	During testing, the "Back" and "Cancel" buttons allow the user to easily reverse their last action. The interface is easily understood on how to reverse actions.

Appendix A: Automated Test Scripts for Web Frontend

```javascript
describe('student login test', () => {
  beforeEach(() => {
    cy.visit('http://localhost:3000')
  })

  it('Wrong password ', () => {
    cy.get('#username').type("stud1")
    cy.get('#password').type("stud1")
    cy.get('#btn-login').click()
    cy.on('window:alert', (str) => {
      expect(str).to.equal(`Wrong input!`)
    })
  })

  it('Empty field ', () => {
    cy.get('#username').type("stud1")
    cy.get('#btn-login').click()
    cy.on('window:alert', (str) => {
      expect(str).to.equal(`Enter all fields!`)
    })
  })

  it('Successful login ', () => {
    cy.get('#username').type("stud1")
    cy.get('#password').type("stud1pw")
    cy.get('#btn-login').click()
    cy.url().should('include', '/play')
  })

});
```

Figure 1: Student Login test cases

```
describe('student create question test', () => {
  beforeEach(() => {
    cy.visit('http://localhost:3000')
    cy.get('#username').type("stud1")
    cy.get('#password').type("stud1pw")
    cy.get('#btn-login').click()
    cy.get("nav").contains("Create").click();
  })

  it('Any field not filled ', () => {
    cy.get('#worldCustom').click({force:true})
    cy.get('#level1').click({force:true})
    cy.get('#question1').click({force:true})
    cy.get('#inputQuestion').type("What is 1 x 1?")
    cy.get('#optionA').type("1")
    cy.get('#optionB').type("2")
    cy.get('#optionC').type("3")
    cy.get('#answerA').click({force:true})
    cy.get('#btn-ok-create').click()
    cy.on('window:alert', (str) => {
      expect(str).to.equal(`Enter all fields!`)
    })
  })

  it('All field filled ', () => {
    cy.get('#worldCustom').click({force:true})
    cy.get('#level1').click({force:true})
    cy.get('#question1').click({force:true})
    cy.get('#inputQuestion').type("What is 1 x 1?")
    cy.get('#optionA').type("1")
    cy.get('#optionB').type("2")
    cy.get('#optionC').type("3")
    cy.get('#optionD').type("4")
    cy.get('#answerA').click({force:true})
    cy.get('#btn-ok-create').click()
    cy.url().should('include', '/read')
  })
});
```

Figure 2: Student Create Question test cases

```
describe('student update question test', () => {
  beforeEach(() => {
    cy.visit('http://localhost:3000')
    cy.get('#username').type("stud1")
    cy.get('#password').type("stud1pw")
    cy.get('#btn-login').click()
    cy.get("nav").contains("Update").click();
    cy.get('#49').click()
  })

  it('Any field not filled ', () => {
    cy.get('#optionC').clear()
    cy.get('#btn-ok-create').click()
    cy.on('window:alert', (str) => {
      expect(str).to.equal(`Enter all fields!`)
    })
  })

  it('All field updated and filled ', () => {
    cy.get('#optionC').clear()
    cy.get('#optionC').type("23")
    cy.get('#btn-ok-create').click()
    cy.url().should('include', '/choose-update-questions')
  })
});
```

Figure 3: Student Update test cases

```
describe('professor login test', () => {
  beforeEach(() => {
    cy.visit('http://localhost:3000')
  })

  it('Wrong password ', () => {
    cy.get('#username').type("prof1")
    cy.get('#password').type("prof1")
    cy.get('#btn-login').click()
    cy.on('window:alert', (str) => {
      expect(str).to.equal(`Wrong input!`)
    })
  })

  it('Empty field ', () => {
    cy.get('#username').type("prof1")
    cy.get('#btn-login').click()
    cy.on('window:alert', (str) => {
      expect(str).to.equal(`Enter all fields!`)
    })
  })

  it('Successful login ', () => {
    cy.get('#username').type("prof1")
    cy.get('#password').type("prof1pw")
    cy.get('#btn-login').click()
    cy.url().should('include', '/summary-report')
  })
});
```

Figure 4: Professor Login test cases

```
describe('professors create question test', () => {
  beforeEach(() => {
    cy.visit('http://localhost:3000')
    cy.get('#username').type("prof1")
    cy.get('#password').type("prof1pw")
    cy.get('#btn-login').click()
    cy.get("nav").contains("Create").click();
  })

  it('Any field not filled ', () => {
    cy.get('#world3').click({force:true})
    cy.get('#level1').click({force:true})
    cy.get('#question1').click({force:true})
    cy.get('#inputQuestion').type("What is 10 x 10?")
    cy.get('#optionA').type("100")
    cy.get('#optionB').type("200")
    cy.get('#optionC').type("300")
    cy.get('#answerA').click({force:true})
    cy.get('#btn-ok-create').click()
    cy.on('window:alert', (str) => {
      expect(str).to.equal(`Enter all fields!`)
    })
  })

  it('All field filled ', () => {
    cy.get('#world3').click({force:true})
    cy.get('#level1').click({force:true})
    cy.get('#question1').click({force:true})
    cy.get('#inputQuestion').type("What is 10 x 10?")
    cy.get('#optionA').type("100")
    cy.get('#optionB').type("200")
    cy.get('#optionC').type("300")
    cy.get('#optionD').type("400")
    cy.get('#answerA').click({force:true})
    cy.get('#btn-ok-create').click()
    cy.url().should('include', '/read')
  })
});
```

Figure 5: Professor create question test cases

```
describe('professor update question test', () => {
  beforeEach(() => {
    cy.visit('http://localhost:3000')
    cy.get('#username').type("prof1")
    cy.get('#password').type("prof1pw")
    cy.get('#btn-login').click()
    cy.get("nav").contains("Update").click();
    cy.get('#48').click()
  })

  it('Any field not filled ', () => {
    cy.get('#optionC').clear()
    cy.get('#btn-ok-create').click()
    cy.on('window:alert', (str) => {
      expect(str).to.equal(`Enter all fields!`)
    })
  })

  it('All field updated and filled ', () => {
    cy.get('#optionC').clear()
    cy.get('#optionC').type("230")
    cy.get('#btn-ok-create').click()
    cy.url().should('include', '/choose-update-questions')
  })
});
```

Figure 6: Professor update question test cases

```
describe('professor delete question test', () => {
  beforeEach(() => {
    cy.visit('http://localhost:3000')
    cy.get('#username').type("prof1")
    cy.get('#password').type("prof1pw")
    cy.get('#btn-login').click()
    cy.get("nav").contains("Delete").click();
    cy.get('#43').click()
  })

  it('Question deleted ', () => {
    cy.on('window:confirm', (str) => {
      expect(str).to.equal("Are you sure you wish to delete this question?")
    })
    cy.on('window:alert', (str) => {
      expect(str).to.equal("Question deleted")
    })
  })
});
```

Figure 7: Professor delete question test case

```
describe('professor view summary report', () => {
    beforeEach(() => {
        cy.visit('http://localhost:3000')
        cy.get('#username').type("prof1")
        cy.get('#password').type("prof1pw")
        cy.get('#btn-login').click()
    })

    it('professor able to view overall summary report', () => {
        cy.url().should('include', '/summary-report')
    })

    it('professor able to view student summary report', () => {
        cy.get('select').select('stud1')
        cy.url().should('include', '/summary-report-student?user=stud1')
    })

})
```

Figure 8: Professor Summary Report test cases