**SCHOOL OF COMPUTING (SoC)**

**ST0503 Back-end Web Development**

**2020/2021 SEMESTER 2**

**ASSIGNMENT 1**

**Instructions and Guidelines:**

1. The assignment **source code** must be submitted before 4th Jan 2021, 9am. You are required to submit your source codes to the BlackBoard. Remember to provide your Class, Group, Admission Number(s) and Name(s) on the softcopy.
2. A one page word document showing the tables you have created and their linkage (including foreign keys) should be included in the submission.
3. You are required to **clearly** provide instructions on how to setup the project on the lecturer's laptop in a text file that is to be included in the softcopy submission.
4. Students are to work in a group of 1-2 members.
5. **Students of 2 members group must complete one of the 2 DEFINED additional features (without extra marks) stated in the document or be penalized 10 marks**.
6. Marks will be given separately for each student in the group, depending on his/her contribution to the assignment. The assignment will account for **30%** of your final grade*.*
7. The assignment should be implemented using Node JS, Express and MySQL.
8. The interview will be conducted during the lessons in the week of **4th Jan 2021**. You are expected to explain the program logic and modify the program during the interview. **If you are absent, you will be awarded zero mark for the assignment.**
9. Your application will be tested with POSTMAN.
10. **No marks will be awarded**, if the work is copied or you have allowed others to copy your work. Warning: Plagiarism means passing off as one's own the ideas, works, writings, etc., which belong to another person. In accordance with this definition, you are committing plagiarism if you copy the work of another person and turning it in as your own, even if you would have the permission of that person. Plagiarism is a serious offence and disciplinary action will be taken against you. If you are guilty of plagiarism, you may fail all modules in the semester, or even be liable for expulsion.
11. 50% of the marks will be deducted for assignments that are received within ONE (1) calendar day after the submission deadline. No marks will be given thereafter.

Exceptions to this policy will be given to students with valid LOA on medical or

compassionate grounds. Students will need to inform the lecturer as soon as reasonably possible. Students are not to assume on their own that their deadline has been extended.

# Assignment 1: SP Games

## Background

With the introduction of new console machines, SP Games would like to open an electronic store to sell electronic games in view of the anticipated demand.

As such, SP Games has tasked you to design the backend API specs for SP Games. The API specs would support functionalities such as user registration, displaying and entry of games info, user reviews etc.

# Assignment Requirements

You are required to fulfil the following basic requirements:

* Create a new MySQL database with the tables needed for this project.
* Proper database design with correct columns, data types and use of primary and foreign key constraints.
* Create an Express server that comply with the specs provided.
* Consume data from MySQL using the mysql library.

Bonus Requirements:

* Create endpoint for image uploading/storage and retrieval of games (with image info) from the server. Server should only accept **jpg** images below **1 MB**.
* Enable a game to be assigned to more than 1 category (many-to-many relationship for game to category). Insertion of game for endpoint 6 and retrieval of ALL categories for game info endpoint 7 must also be updated to reflect this change.
* Other reasonable and relevant features for assignment as advanced feature (You can check with your lecturer). Depending on complexity and relevancy, you will be awarded appropriate marks.

**(Students of 2-member group must successfully complete one of the above 2 defined advanced features – image or game supporting many categories(without extra marks awarded) or face a penalty of 10 marks. 2-member teams must also provide a listing detailing the contribution of each member. This is used to determine the individual contribution, and members who do not contribute will be PENALIZED).**

**Grading Guidelines**

The assignment will be assessed based on the following criteria:

* Demonstrate and satisfy the web api endpoint functionalities listed below to access/update and return relevant data from the database upon success or failure(75%)

Note: There are 12 APIs, with each api taking 5-10 marks.

Components of grading per api is based primarily on correctness (returning the right json data with proper Model, Controller layer and database calls) and returning the right failure message.

* Proper database design and creation of tables(Game, category, user and reviews) with sample data in MySQL (10%)
* Advanced Features (15%)
* Amount of individual contribution to the project
* Question & Answer during the interview

# SP Games API Specs

For this assignment, you are required to create the following endpoints. The response message body for failed operations can be determined by you in the API.

## 1)

## Endpoint: GET /users/

**Request body schema**: N/A

### Success Response

**Code**: 200 OK

**Content**: Array of all the users in the database, who may be admin or customer type:

[  
 {  
 "userid": 1,  
 "username": "Terry Tan",  
 “email”: “[terry@gmail.com](mailto:terry@gmail.com)”,  
 "type": "Customer",  
 "profile\_pic\_url": "https://www.abc.com/terry.jpg",  
 "created\_at": "2020-11-02 17:54:57"  
 },  
 ...  
]

### Error Response(s)

**Condition**: Unknown error

**Code**: 500 Internal Server Error

## 2)

## Endpoint: POST /users/

Used to add a new user to the database.

### Success Response

**Code**: 201 Created

**Content**: ID of the newly created user:

{  
 "userid": 1  
}

**Request Body**:

Ensure that the id and created timestamp are autogenerated and not provided by the user.

{  
 “username": "Terry Tan",

“email”: “[terry@gmail.com](mailto:terry@gmail.com)”,  
 "type": "Customer",  
 "profile\_pic\_url": "https://www.abc.com/terry.jpg"  
  
}

### Error Response(s)

**Condition**: Unknown error

**Code**: 500 Internal Server Error

## 3)

## Endpoint: GET /users/:id/

Retrieve a single user by their id.

**Request body schema**: N/A

### Success Response

**Code**: 200 OK

**Content**: A single user:

{  
 "userid": 1,  
 "username": "Terry Tan",

“email”: “[terry@gmail.com](mailto:terry@gmail.com)”,  
 "profile\_pic\_url": "https://www.abc.com/terry.jpg",

"type": "Customer",  
 "created\_at": "2020-11-02 17:54:57"  
}

### Error Response(s)

**Condition**: Unknown error

**Code**: 500 Internal Server Error

## 4)

## Endpoint: POST /category

Inserts a new category.

**Request body schema**:

{  
 “catname": "Action",

“description”: “An action game emphasizes physical challenges, including hand–eye coordination and reaction-time”  
   
  
}

### Success Response

**Code**: 204 No Content

**Content**: N/A

### Error Response(s)

**Condition: The category name provided already exists.**

**Code: 422 Unprocessable Entity**

**Condition**: Unknown error

**Code**: 500 Internal Server Error

## 5)

## Endpoint: PUT /category/:id/

Update a category. ID and created timestamp should not be updatable.

**Request body schema**:

Refer to the request body schema for the POST /category endpoint.

### Success Response

**Code**: 204 No Content

**Content**: N/A

### Error Response(s)

**Condition: The category name provided already exists.**

**Code: 422 Unprocessable Entity**

**Condition**: Unknown error

**Code**: 500 Internal Server Error

## 6)

## Endpoint: POST /game

Used to add a new game to the database.

**Request body schema**:

Ensure that the id and created timestamp are autogenerated and not provided by the user.

{  
   
 "title": "Assassin’s Creed Valhalla",  
 "description": "Assassin's Creed Valhalla is an action role-playing video game developed by Ubisoft Montreal and published by Ubisoft"  
 "price": “69.90”,

“platform”: “PC”,

“categoryid”: “1”,

“year”:2020   
}

### Success Response

**Code**: 201 Created

**Content**: ID of the newly created game id:

{  
 "gameid": 1  
}

### Error Response(s)

**Condition**: Unknown error

**Code**: 500 Internal Server Error

## 

## 7)

## Endpoint: GET /games/:platform

Retrieves all games of a certain platform(eg PC,PS5,XBOX X etc)

**Request body schema**: N/A

### Success Response

**Code**: 200 OK

**Content**: Array of all the games listings for PC platform (Table join is required). Take note of the columns data to be returned:

[  
   
 {

"gameid": 1,  
 "title": "Assassin’s Creed Valhalla",  
 "description": "Assassin's Creed Valhalla is an action role-playing video game developed by Ubisoft Montreal and published by Ubisoft"  
  
 "price": “69.90”,

“platform”: “PC”,

“catid”:1,

“catname”: “Action”,

“year”:2020  
 "created\_at": "2020-11-02 17:56:50"  
   
},

…..

]

### Error Response(s)

**Condition**: Unknown error

**Code**: 500 Internal Server Error

## 8)

## Endpoint: DELETE /game/:id

Deletes a game given its id. **The associated reviews related to the game would also be deleted (Cascade delete)**. Idempotent.

**Request body schema**: N/A

### Success Response

**Code**: 204 No Content

**Content**: N/A

### Error Response(s)

**Condition**: Unknown error

**Code**: 500 Internal Server Error

## 9)

## Endpoint: PUT /game/:id

Updates a game listing.

**Request body schema**: Refer to the schema for POST /game.

### Success Response

**Code**: 204 No Content

**Content**: N/A

### Error Response(s)

**Condition**: Unknown error

**Code**: 500 Internal Server Error

## 10)

## Endpoint: POST /user/:uid/game/:gid/review/

Used to add a new review to the database for a given user and game.

### Success Response

**Code**: 201 Created

**Content**: ID of the newly created user:

{  
 "reviewid": 1  
}

**Request Body**:

Ensure that the id and created timestamp are autogenerated and not provided by the user.

{  
 "content": "Enjoyed the game! The story and gameplay was good!",

“rating”: “5”  
}

### Error Response(s)

**Condition**: Unknown error

**Code**: 500 Internal Server Error

11)

## Endpoint: GET /game/:id/review

Retrieves reviews of a particular game, including info like the username. (A table join is required). Note the created\_at field retrieved is the creation datetime of the game review.

### Success Response

**Code**: 200 OK

**Content**:

[  
 {  
 "gameid": “1”,  
 "content": "Enjoyed the game! The story and gameplay was great!",

“rating”: “5”,  
 "username": "Terry Tan",  
 "created\_at": "2020-11-22 18:59:57"  
 },  
 ...  
]

### Error Response(s)

**Condition**: Unknown error

**Code**: 500 Internal Server Error