

Quizz Lab

1: Xác định yêu cầu hệ thống

1. Quản lý người dùng:
 - Đăng ký và đăng nhập người dùng.
 - Quản lý thông tin người dùng, bao gồm tên, địa chỉ email, mật khẩu, v.v.
 - Phân quyền và quản lý vai trò của người dùng (quản trị viên, giáo viên, học viên).
2. Quản lý câu hỏi:
 - Tạo, chỉnh sửa và xóa câu hỏi.
 - Gắn thẻ và phân loại câu hỏi theo danh mục, môn học, độ khó, v.v.
 - Hỗ trợ đa dạng hình thức câu hỏi như trắc nghiệm, điền khuyết, v.v.
 - Lưu trữ câu hỏi và đáp án vào cơ sở dữ liệu.
3. Tạo và quản lý gói câu hỏi:
 - Tạo gói câu hỏi bằng cách chọn các câu hỏi từ cơ sở dữ liệu.
 - Xác định số lượng câu hỏi trong mỗi gói.
 - Lưu trữ và quản lý gói câu hỏi đã tạo.
4. Thực hiện bài trắc nghiệm:
 - Hiện thị gói câu hỏi cho người dùng.
 - Hiện thị câu hỏi một cách ngẫu nhiên hoặc theo thứ tự đã định sẵn.
 - Ghi nhận và tính điểm cho các câu trả lời của người dùng.
 - Cung cấp phản hồi cho người dùng sau khi hoàn thành bài trắc nghiệm.
5. Thống kê và theo dõi kết quả:
 - Lưu trữ và hiển thị kết quả bài trắc nghiệm của người dùng.
 - Tính toán và hiển thị điểm số và phần trăm đạt được.
 - Cung cấp thống kê kết quả theo người dùng, gói câu hỏi, môn học, v.v.
6. Giao diện người dùng:
 - Thiết kế giao diện thân thiện và dễ sử dụng cho người dùng.
 - Tích hợp các tính năng tương tác như tìm kiếm câu hỏi, sắp xếp, lọc, v.v.
7. Bảo mật và quản lý hệ thống:
 - Đảm bảo an toàn thông tin người dùng và dữ liệu câu hỏi.
 - Sao lưu và khôi phục dữ liệu định kỳ.
 - Quản lý lỗi và ghi nhật ký hệ thống.

2. Phân tích yêu cầu hướng dịch vụ

a. Xác định các dịch vụ cần thiết

1. Dịch vụ Quản lý người dùng:

- Đăng ký người dùng mới: Hỗ trợ người dùng đăng ký tài khoản mới bằng cách cung cấp thông tin cần thiết và lưu trữ thông tin người dùng trong cơ sở dữ liệu.
- Đăng nhập người dùng: Xác thực danh tính người dùng bằng cách kiểm tra thông tin đăng nhập và phân quyền người dùng để cho phép truy cập vào các chức năng tương ứng.
- Quản lý thông tin người dùng: Cung cấp giao diện để người dùng cập nhật thông tin cá nhân, thay đổi mật khẩu và quản lý tài khoản của họ.

2. Dịch vụ Quản lý câu hỏi:

- Tạo câu hỏi mới: Cung cấp giao diện để người dùng tạo mới câu hỏi bằng cách nhập nội dung câu hỏi, lựa chọn hình thức câu hỏi và cung cấp đáp án.
- Chỉnh sửa câu hỏi: Cho phép người dùng chỉnh sửa nội dung câu hỏi, hình thức câu hỏi và đáp án đã được tạo trước đó.
- Xóa câu hỏi: Cho phép người dùng xóa câu hỏi không còn cần thiết khỏi cơ sở dữ liệu.

3. Dịch vụ Quản lý gói câu hỏi:

- Tạo gói câu hỏi mới: Cung cấp giao diện để người dùng tạo gói câu hỏi bằng cách chọn các câu hỏi từ cơ sở dữ liệu đã có và xác định số lượng câu hỏi trong gói.
- Chỉnh sửa gói câu hỏi: Cho phép người dùng thay đổi các câu hỏi trong gói câu hỏi và số lượng câu hỏi.
- Xóa gói câu hỏi: Cho phép người dùng xóa gói câu hỏi không còn cần thiết khỏi cơ sở dữ liệu.

4. Dịch vụ Thực hiện bài trắc nghiệm:

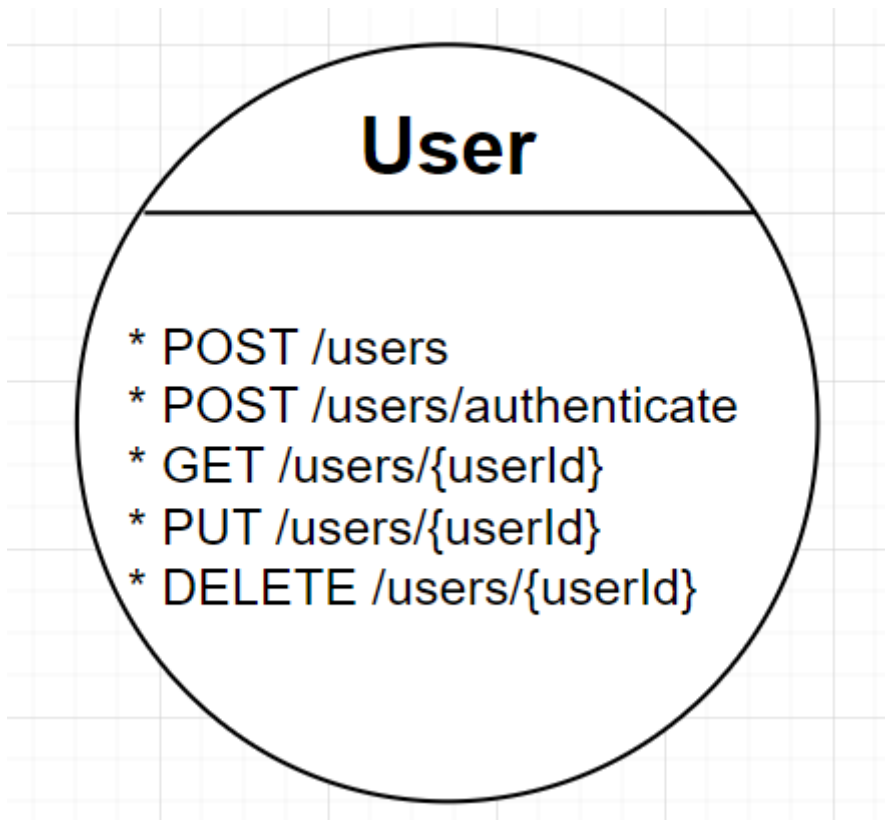
- Hiện thị gói câu hỏi: Cung cấp giao diện để người dùng xem gói câu hỏi và bắt đầu bài trắc nghiệm.
- Ghi nhận câu trả lời: Ghi nhận câu trả lời của người dùng và kiểm tra tính đúng sai.
- Tính điểm và cung cấp phản hồi: Tính toán điểm số dựa trên câu trả lời và cung cấp phản hồi cho người dùng sau khi hoàn thành bài trắc nghiệm.

5. Dịch vụ Thống kê và theo dõi kết quả:

- Lưu trữ kết quả bài trắc nghiệm: Lưu trữ kết quả bài trắc nghiệm của người dùng, bao gồm điểm số, phần trăm đạt được và thời gian hoàn thành.
- Tính toán và hiển thị kết quả: Tính toán điểm số dựa trên câu trả lời và cung cấp phản hồi cho người dùng sau khi hoàn thành bài trắc nghiệm.

b. Giao diện dịch vụ, phương thức truy cập và định nghĩa thông điệp dịch vụ

1. Dịch vụ Quản lý người dùng:



- Giao diện dịch vụ: RESTful API
- Phương thức truy cập: HTTP POST, GET, PUT, DELETE
- Định nghĩa thông điệp dịch vụ:
 - Đăng ký người dùng mới: POST /users
 - Xác thực người dùng: POST /users/authenticate
 - Lấy thông tin người dùng: GET /users/{userId}
 - Cập nhật thông tin người dùng: PUT /users/{userId}
 - Xóa người dùng: DELETE /users/{userId}

2. Dịch vụ Quản lý câu hỏi:

Question

- * POST /questions
- * GET
/questions/{questionId}
- * PUT
/questions/{questionId}
- * DELETE
/questions/{questionId}

- Giao diện dịch vụ: RESTful API
- Phương thức truy cập: HTTP POST, GET, PUT, DELETE
- Định nghĩa thông điệp dịch vụ:
 - Tạo câu hỏi mới: POST /questions
 - Lấy thông tin câu hỏi: GET /questions/{questionId}
 - Cập nhật thông tin câu hỏi: PUT /questions/{questionId}
 - Xóa câu hỏi: DELETE /questions/{questionId}

3. Dịch vụ Quản lý gói câu hỏi:

QuestionPackage

- * POST /question-packages
- * GET /question-packages/{packageId}
- * PUT /question-packages/{packageId}
- * DELETE /question-packages/{packageId}

- Giao diện dịch vụ: RESTful API
 - Phương thức truy cập: HTTP POST, GET, PUT, DELETE
 - Định nghĩa thông điệp dịch vụ:
 - Tạo gói câu hỏi mới: POST /question-packages
 - Lấy thông tin gói câu hỏi: GET /question-packages/{packageId}
 - Cập nhật thông tin gói câu hỏi: PUT /question-packages/{packageId}
 - Xóa gói câu hỏi: DELETE /question-packages/{packageId}
4. Dịch vụ Thực hiện bài trắc nghiệm:

Quizz

- * POST /quizzes/start
- * GET /quizzes/{quizId}/next-question
- * POST /quizzes/{quizId}/answers
- * POST /quizzes/{quizId}/finish

- Giao diện dịch vụ: RESTful API
- Phương thức truy cập: HTTP POST, GET
- Định nghĩa thông điệp dịch vụ:
 - Bắt đầu bài trắc nghiệm: POST /quizzes/start
 - Lấy câu hỏi tiếp theo: GET /quizzes/{quizId}/next-question
 - Gửi câu trả lời: POST /quizzes/{quizId}/answers
 - Kết thúc bài trắc nghiệm: POST /quizzes/{quizId}/finish

5. Dịch vụ Thống kê và theo dõi kết quả:

StatResult

- * GET /users/{userId}/quiz-results
- * GET /quiz-results/statistics

- Giao diện dịch vụ: RESTful API
- Phương thức truy cập: HTTP GET
- Định nghĩa thông điệp dịch vụ:
 - Lấy kết quả bài trắc nghiệm của người dùng: GET /users/{userId}/quiz-results
 - Lấy thống kê kết quả bài trắc nghiệm: GET /quiz-results/statistics

3. Mô hình kiến trúc hướng dịch vụ

1. Kiến trúc tổng quan:
 - Hệ thống sử dụng mô hình kiến trúc dựa trên Microservices.
 - Mỗi dịch vụ được thiết kế để thực hiện một chức năng cụ thể và độc lập với các dịch vụ khác.
 - Các dịch vụ giao tiếp thông qua giao diện dịch vụ (API) để truyền thông tin và thực hiện các tác vụ.
 - Sử dụng công nghệ và công cụ phù hợp cho việc triển khai và quản lý các dịch vụ như Docker, Kubernetes, hoặc các môi trường điện toán đám mây (cloud computing) như AWS, Azure, Google Cloud.
2. Các thành phần dịch vụ:
 - Dịch vụ Quản lý người dùng: Quản lý thông tin người dùng, xác thực và quản lý phiên làm việc.
 - Dịch vụ Quản lý câu hỏi: Quản lý thông tin câu hỏi, bao gồm tạo, lưu trữ và cập nhật câu hỏi.

- Dịch vụ Quản lý gói câu hỏi: Quản lý thông tin về gói câu hỏi, bao gồm tạo, lưu trữ và cập nhật gói câu hỏi.
 - Dịch vụ Thực hiện bài trắc nghiệm: Quản lý quá trình bài trắc nghiệm, bao gồm bắt đầu, lấy câu hỏi, gửi câu trả lời và kết thúc bài trắc nghiệm.
 - Dịch vụ Thống kê và theo dõi kết quả: Tính toán và cung cấp thông tin thống kê về kết quả bài trắc nghiệm của người dùng.
3. Cấu trúc và quy mô của hệ thống:
- Mỗi dịch vụ được triển khai riêng biệt và độc lập.
 - Sử dụng cơ sở dữ liệu riêng cho mỗi dịch vụ để lưu trữ thông tin liên quan.
 - Áp dụng các nguyên tắc và tiêu chuẩn thiết kế hướng dịch vụ như sự độc lập, tự đảm bảo, khả năng mở rộng và tái sử dụng.
4. Quy tắc và tiêu chuẩn:
- Tuân thủ các nguyên tắc thiết kế hướng dịch vụ như nguyên tắc đơn trách nhiệm (Single Responsibility Principle), nguyên tắc phân tách (Separation of Concerns), và nguyên tắc khả năng mở rộng (Scalability).
 - Sử dụng giao diện dịch vụ (API) chuẩn để đảm bảo tính tương thích và tương tác dễ dàng giữa các dịch vụ.
 - Áp dụng các tiêu chuẩn an ninh, bảo mật và quản lý lỗi để đảm bảo tính bảo mật và ổn định của hệ thống.

4. Thiết kế chi tiết dịch vụ

1. Dịch vụ Quản lý người dùng:

Chức năng:

1. Đăng ký người dùng mới:

- Chức năng: Cho phép người dùng đăng ký tài khoản mới trong hệ thống.
- Thông điệp: POST /users/register
- Giao diện (API):

```

POST /users/register

Request body: {

"username": "TEN\_DANG\_NHAP",

"password": "MAT\_KHAU",



```
"email": "EMAIL"
}
Response: {
 "user_id": "ID_NGUOI_DUNG"
}
...

```

## 2. Đăng nhập:

- Chức năng: Cho phép người dùng đăng nhập vào tài khoản của mình.
- Thông điệp: POST /users/login
- Giao diện (API):

```
...

POST /users/login

Request body: {
 "username": "TEN_DANG_NHAP",
 "password": "MAT_KHAU"
}

Response: {
 "user_id": "ID_NGUOI_DUNG",
 "access_token": "TOKEN_TRUY_CAP"
}
...

```

## 3. Xem thông tin người dùng:

- Chức năng: Cung cấp thông tin chi tiết về người dùng.
- Thông điệp: GET /users/{user\_id}
- Giao diện (API):

```
...

GET /users/{user_id}

```

```
Response: {
 "user_id": "ID_NGUOI_DUNG",
 "username": "TEN_DANG_NHAP",
 "email": "EMAIL",
 "created_at": "NGAY_TAO"
}
...
```

#### 4. Cập nhật thông tin người dùng:

- Chức năng: Cho phép người dùng cập nhật thông tin cá nhân của mình.
  - Thông điệp: PUT /users/{user\_id}
  - Giao diện (API):
- ...

```
PUT /users/{user_id}
Request body: {
 "username": "TEN_DANG_NHAP_MOI",
 "email": "EMAIL_MOI"
}
Response: {
 "user_id": "ID_NGUOI_DUNG"
}
...
```

#### 5. Xóa người dùng:

- Chức năng: Cho phép quản trị viên xóa tài khoản người dùng khỏi hệ thống.
  - Thông điệp: DELETE /users/{user\_id}
  - Giao diện (API):
- ...

```
DELETE /users/{user_id}
```

Response: 204 No Content

...

## 6. Tạo phiên mới:

- Chức năng: Cho phép người dùng tạo phiên mới để bắt đầu một phiên làm việc hoặc kết nối với hệ thống.

- Thông điệp: POST /sessions

- Giao diện (API):

...

POST /sessions

Response: {

  "session\_id": "ID\_PHIEN\_MOI"

}

...

## 7. Xác thực phiên:

- Chức năng: Cho phép kiểm tra tính hợp lệ của phiên hiện tại và xác thực người dùng.

- Thông điệp: GET /sessions/{session\_id}

- Giao diện (API):

...

GET /sessions/{session\_id}

Response: {

  "valid": true,

  "user\_id": "ID\_NGUOI\_DUNG"

}

...

## 8. Kết thúc phiên:

- Chức năng: Cho phép người dùng kết thúc phiên làm việc hoặc đăng xuất khỏi hệ thống.

- Thông điệp: DELETE /sessions/{session\_id}

- Giao diện (API):

```

DELETE /sessions/{session_id}

Response: 204 No Content

```

## 9. Bảo mật và quản lý:

- Thông điệp và giao thức: Sử dụng giao thức HTTP/HTTPS để giao tiếp với các yêu cầu và phản hồi dữ liệu.
- Giao diện (API): Thiết kế API cho việc đăng ký, đăng nhập và các chức năng quản lý người dùng.
- Định dạng dữ liệu: Sử dụng định dạng dữ liệu JSON để truyền và lưu trữ thông tin người dùng.
- Mã hóa mật khẩu: Mật khẩu người dùng được mã hóa trước khi lưu trữ trong hệ thống để đảm bảo tính bảo mật.
- Xác thực người dùng: Yêu cầu xác thực người dùng trước khi cho phép truy cập vào thông tin cá nhân và thực hiện các hành động nhất định.
- Phân quyền: Áp dụng phân quyền để quản lý quyền truy cập và hành động của người dùng trong hệ thống.
  - Quản trị viên: Quản trị viên có quyền truy cập và thực hiện các thao tác quản lý người dùng, bao gồm tạo, cập nhật và xóa người dùng. Quản trị viên cũng có quyền xem thông tin chi tiết của tất cả người dùng trong hệ thống.
  - Người dùng thông thường: Người dùng được phép tạo tài khoản, cập nhật thông tin cá nhân và đăng nhập vào hệ thống. Người dùng chỉ có quyền truy cập và cập nhật thông tin cá nhân của chính mình.
- Ghi nhật ký (logging): Ghi nhật ký các hoạt động liên quan đến quản lý người dùng để có thể giám sát và phân tích sau này.

## 2. Dịch vụ Quản lý câu hỏi:

Chức năng:

### 1. Tạo câu hỏi mới:

- Chức năng: Cho phép quản trị viên tạo câu hỏi mới bằng cung cấp nội dung câu hỏi, các lựa chọn câu trả lời và đáp án đúng.

- Thông điệp: POST /questions

- Giao diện (API):

...

POST /questions

Request body: {

"question": "NOI\_DUNG\_CAU\_HOI",

"choices": [

{

"choice\_id": "ID\_LUA\_CHON",

"choice\_text": "NOI\_DUNG\_LUA\_CHON"

},

...

],

"correct\_choice\_id": "ID\_LUA\_CHON\_DUNG"

}

Response: {

"question\_id": "ID\_CAU\_HOI"

}

...

## 2. Lưu trữ câu hỏi:

- Chức năng: Lưu trữ thông tin câu hỏi vào cơ sở dữ liệu để có thể truy xuất và sử dụng sau này.
- Thông điệp: Không có thông điệp riêng, thông tin câu hỏi sẽ được lưu trữ trong cơ sở dữ liệu.

## 3. Cập nhật câu hỏi:

- Chức năng: Cho phép quản trị viên cập nhật nội dung, lựa chọn câu trả lời và đáp án đúng của câu hỏi đã tồn tại.

- Thông điệp: PUT /questions/{question\_id}

- Giao diện (API):

---

PUT /questions/{question\_id}

Request body: {

"question": "NOI\_DUNG\_CAU\_HOI\_MOI",

"choices": [

{

"choice\_id": "ID\_LUA\_CHON",

"choice\_text": "NOI\_DUNG\_LUA\_CHON\_MOI"

},

...

],

"correct\_choice\_id": "ID\_LUA\_CHON\_DUNG\_MOI"

}

```
Response: {

 "question_id": "ID_CAU_HOI"

}
...
```

#### 4. Xóa câu hỏi:

- Chức năng: Cho phép quản trị viên xóa câu hỏi không còn cần thiết.
- Thông điệp: DELETE /questions/{question\_id}
- Giao diện (API):

```
...

DELETE /questions/{question_id}

Response: 204 No Content

...
```

#### 5. Bảo mật và quản lý

- Thông điệp và giao thức: Sử dụng giao thức HTTP/HTTPS để giao tiếp với các yêu cầu và phản hồi dữ liệu.
- Giao diện (API): Thiết kế API cho việc tạo, cập nhật và xóa câu hỏi.
- Định dạng dữ liệu: Sử dụng định dạng dữ liệu JSON để truyền và lưu trữ thông tin câu hỏi.
- Yêu cầu bảo mật và quản lý:
  - Quản trị viên: Quản trị viên có quyền truy cập và thực hiện các thao tác quản lý câu hỏi, bao gồm tạo, cập nhật và xóa câu hỏi. Quản trị viên cũng có quyền xem thông tin chi tiết của tất cả câu hỏi trong hệ thống.
  - Người dùng thông thường: Người dùng thông thường có quyền truy cập vào dịch vụ Quản lý câu hỏi để tạo mới, cập nhật và xóa câu hỏi của riêng mình. Họ không có quyền truy cập vào câu hỏi của người dùng khác.

### ***3. Dịch vụ Quản lý gói câu hỏi:***

Chức năng:

#### **1. Tạo gói câu hỏi mới:**

- Chức năng: Cho phép quản trị viên tạo gói câu hỏi mới bằng cung cấp danh sách câu hỏi từ dịch vụ Quản lý câu hỏi.

- Thông điệp: POST /question-packages

- Giao diện (API):

```

POST /question-packages

Request body: {

 "name": "TEN_GOI_CAU_HOI",

 "questions": [

 "ID_CAU_HOI_1",

 "ID_CAU_HOI_2",

 ...

]

}

Response: {

 "package_id": "ID_GOI_CAU_HOI"

}

```



## 2. Lưu trữ gói câu hỏi:

- Chức năng: Lưu trữ thông tin về gói câu hỏi vào cơ sở dữ liệu để có thể truy xuất và sử dụng sau này.
- Thông điệp: Không có thông điệp riêng, thông tin gói câu hỏi sẽ được lưu trữ trong cơ sở dữ liệu.

## 3. Cập nhật gói câu hỏi:

- Chức năng: Cho phép quản trị viên cập nhật danh sách câu hỏi trong gói câu hỏi đã tồn tại.

- Thông điệp: PUT /question-packages/{package\_id}

- Giao diện (API):

---

PUT /question-packages/{package\_id}

Request body: {

"name": "TEN\_GOI\_CAU\_HOI\_MOI",

"questions": [

"ID\_CAU\_HOI\_1",

"ID\_CAU\_HOI\_2",

...

]

}

Response: {

"package\_id": "ID\_GOI\_CAU\_HOI"

}

---

#### 4. Xóa gói câu hỏi:

- Chức năng: Cho phép quản trị viên xóa gói câu hỏi không còn cần thiết.

- Thông điệp: DELETE /question-packages/{package\_id}

- Giao diện (API):

...

DELETE /question-packages/{package\_id}

Response: 204 No Content

...

#### 5. Bảo mật và quản lý

- Thông điệp và giao thức: Sử dụng giao thức HTTP/HTTPS để giao tiếp với các yêu cầu và phản hồi dữ liệu.
- Giao diện (API): Thiết kế API cho việc tạo, cập nhật và xóa gói câu hỏi.
- Định dạng dữ liệu: Sử dụng định dạng dữ liệu JSON để truyền và lưu trữ thông tin gói câu hỏi.
- Yêu cầu bảo mật và quản lý:
  - Quản trị viên: Quản trị viên có quyền truy cập và thực hiện các thao tác quản lý gói câu hỏi, bao gồm tạo, cập nhật và xóa gói câu hỏi. Quản trị viên cũng có quyền xem thông tin chi tiết của tất cả gói câu hỏi trong hệ thống.
  - Người dùng thông thường: Người dùng thông thường có quyền truy cập vào dịch vụ Quản lý gói câu hỏi để tạo mới, cập nhật và xóa gói câu hỏi của riêng mình. Họ không có quyền truy cập vào gói câu hỏi của người dùng khác.

#### 4. Dịch vụ Thực hiện bài trắc nghiệm:

Chức năng:

##### 1. Bắt đầu bài trắc nghiệm:

- Chức năng: Cho phép người dùng bắt đầu một bài trắc nghiệm bằng cách chọn một gói câu hỏi từ dịch vụ Quản lý gói câu hỏi.

- Thông điệp: POST /quiz/start

- Giao diện (API):

```

POST /quiz/start

Request body: {

 "package_id": "ID_GOI_CAU_HOI"

}

Response: {

 "quiz_id": "ID_BAI_TRAC_NGHIEM"

}

```

## 2. Lấy câu hỏi:

- Chức năng: Cung cấp cho người dùng một câu hỏi từ gói câu hỏi đang thực hiện.

- Thông điệp: GET /quiz/questions/{quiz\_id}

- Giao diện (API):

```

GET /quiz/questions/{quiz_id}

Response: {

 "question_id": "ID_CAU_HOI",

 "question_text": "NỘI DUNG_CAU_HOI"

}

```

## 3. Gửi câu trả lời:

- Chức năng: Cho phép người dùng gửi câu trả lời cho câu hỏi hiện tại.

- Thông điệp: POST /quiz/questions/{question\_id}/answers

- Giao diện (API):

```

POST /quiz/questions/{question_id}/answers

Request body: {

 "answer": "CAU_TRA_LOI"

}

Response: {

 "result": "ket_qua"

}

...

4. Kết thúc bài trắc nghiệm:

- Chức năng: Đánh giá kết quả bài trắc nghiệm và cung cấp điểm số cho người dùng.

- Thông điệp: POST /quiz/finish

- Giao diện (API):

...

POST /quiz/finish

Request body: {

 "quiz_id": "ID_BAI_TRAC_NGHIEM"

}

Response: {

 "score": "DIEM_SO"

}

...

5. Bảo mật và quản lý

- Định dạng dữ liệu: Sử dụng định dạng dữ liệu JSON để truyền và lưu trữ thông tin câu hỏi và kết quả bài trắc nghiệm.
- Thông điệp và giao thức: Sử dụng giao thức HTTP/HTTPS để giao tiếp với các yêu cầu và phản hồi dữ liệu.
- Yêu cầu bảo mật và quản lý: Yêu cầu xác thực người dùng trước khi cho phép thực hiện bài trắc nghiệm :

- Quản trị viên có quyền thực hiện tất cả các bài trắc nghiệm trên hệ thống
- Người dùng có thể thực hiện bài trắc nghiệm của chính mình, bài trắc nghiệm công khai và bài trắc nghiệm không công khai nếu có mã
- Bảo mật dữ liệu: Đảm bảo tính bảo mật của dữ liệu truyền đi và lưu trữ thông tin câu hỏi và kết quả bài trắc nghiệm.

5. Dịch vụ Thống kê và theo dõi kết quả:

Chức năng:

1. Tính toán kết quả bài trắc nghiệm:

- Chức năng: Tính toán điểm số và kết quả bài trắc nghiệm của người dùng dựa trên câu trả lời đã gửi.

- Thông điệp: POST /quiz/{quiz_id}/calculate

- Giao diện (API):

```

POST /quiz/{quiz\_id}/calculate

Response: {

  "score": "DIEM\_SO",

  "result": "KET\_QUA"

}

```

2. Cung cấp thông tin thống kê:

- Chức năng: Cung cấp thông tin thống kê về kết quả bài trắc nghiệm theo từng người dùng hoặc gói câu hỏi.

- Thông điệp: GET /statistics/user/{user_id} hoặc GET /statistics/package/{package_id}

- Giao diện (API):

```

GET /statistics/user/{user\_id}

```
Response: {
 "user_id": "ID_NGUOI_DUNG",
 "average_score": "DIEM_TRUNG_BINH",
 "total_quizzes": "TONG_SO_BAI_TRAC_NGHIEM"
}
```

GET /statistics/package/{package\_id}

```
Response: {
 "package_id": "ID_GOI_CAU_HOI",
 "average_score": "DIEM_TRUNG_BINH",
 "total_quizzes": "TONG_SO_BAI_TRAC_NGHIEM"
}
...
```

### 3. Theo dõi kết quả:

- Chức năng: Theo dõi và lưu trữ kết quả bài trắc nghiệm của người dùng để có thể truy xuất và phân tích sau này.

- Thông điệp: POST /quiz/{quiz\_id}/result

- Giao diện (API):

...

POST /quiz/{quiz\_id}/result

```
Request body: {
 "user_id": "ID_NGUOI_DUNG",
 "score": "DIEM_SO",
 "result": "KET_QUA"
}
```

```
Response: {
 "result_id": "ID_KET_QUA"
}
```

...

#### 4. Cập nhật kết quả:

- Chức năng: Cho phép cập nhật kết quả bài trắc nghiệm của người dùng nếu cần thiết.

- Thông điệp: PUT /quiz/{quiz\_id}/result/{result\_id}

- Giao diện (API):

...

PUT /quiz/{quiz\_id}/result/{result\_id}

Request body: {

"score": "DIEM\_SO\_MOI",

"result": "KET\_QUA\_MOI"

}

Response: {

"result\_id": "ID\_KET\_QUA"

}

...

#### 5. Xóa kết quả:

- Chức năng: Cho phép xóa kết quả bài trắc nghiệm của người dùng nếu cần thiết.

- Thông điệp: DELETE /quiz/{quiz\_id}/result/{result\_id}

- Giao diện (API):

...

DELETE /quiz/{quiz\_id}/result/{result\_id}

Response: 204 No Content

...

## 6. Bảo mật và quản lý:

- Định dạng dữ liệu: Sử dụng định dạng dữ liệu JSON để truyền và lưu trữ thông tin câu hỏi và kết quả bài trắc nghiệm.
- Thông điệp và giao thức: Sử dụng giao thức HTTP/HTTPS để giao tiếp với các yêu cầu và phản hồi dữ liệu.
- Xác thực người dùng: Yêu cầu xác thực người dùng trước khi cho phép truy cập vào thông tin thống kê và theo dõi kết quả.
- Phân quyền: Áp dụng phân quyền để đảm bảo chỉ những người dùng có quyền truy cập mới có thể xem thông tin thống kê và theo dõi kết quả.
  - Quản trị viên có quyền xem thống kê và kết quả của mọi bài trắc nghiệm trên hệ thống.
  - Người dùng tạo trắc nghiệm chỉ có quyền xem thống kê và theo dõi kết quả bài trắc nghiệm của tất cả người dùng tham gia bài trắc nghiệm do họ tạo.
  - Người dùng tham gia trắc nghiệm do người khác tạo chỉ có quyền xem kết quả của bài trắc nghiệm do họ làm.
- Bảo mật dữ liệu: Đảm bảo tính bảo mật của dữ liệu truyền đi và lưu trữ thông tin kết quả bài trắc nghiệm.
- Ghi nhật ký (logging): Ghi nhật ký các hoạt động liên quan đến thống kê và theo dõi kết quả để có thể giám sát và phân tích sau này.