TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

──────── \* ───────



Hệ thống website hỗ trợ học tiếng Nhật

Môn: Công nghệ Web tiên tiến

**Nhóm 6**

Biện Thành Chung – 20150393

Phạm Hoàn – 20151493

Phùng Văn Hưng– 20151883

Trần Viết Huy– 20151716

Giảng viên hướng dẫn:

Thầy Nguyễn Nhất Hải

*Hà Nội, ngày 20 tháng 4 năm 2019*

**Mục lục**

[1. Giới thiệu 3](#_Toc8487515)

[1.1. Mục đích 3](#_Toc8487516)

[1.2. Phạm vi 3](#_Toc8487517)

[1.3. Các tính năng 3](#_Toc8487518)

[1.4. Công nghệ sử dụng 3](#_Toc8487519)

[2. Phân tích thiết kế 4](#_Toc8487520)

[2.1. Thiết kế tổng quan 4](#_Toc8487521)

[2.1.1. *Kiến trúc Frontend* 4](#_Toc8487522)

[2.1.2. *Kiến trúc Backend* 6](#_Toc8487523)

[2.2. Các tác nhân 7](#_Toc8487524)

[2.3. Biểu đồ use case tổng quan 7](#_Toc8487525)

[2.4. Biểu đồ tương tác 7](#_Toc8487526)

[2.4.1. *Biểu đồ trình tự cho chức năng "Đăng nhập"* 7](#_Toc8487527)

[2.4.2. *Biểu đồ trình tự cho chức năng "Đăng kí "* 10](#_Toc8487528)

[2.4.3. *Biểu đồ trình tự cho chức năng "Đăng xuất"* 12](#_Toc8487529)

[2.4.4. *Biểu đồ trình tự cho chức năng "Luyện viết chữ "* 13](#_Toc8487530)

[2.4.5. *Biểu đồ trình tự cho chức năng "Thi Thử "* 14](#_Toc8487531)

[2.4.6. *Biểu đồ trình tự cho chức năng "Xem lượt thi cao nhất "* 16](#_Toc8487532)

[2.4.7. *Biểu đồ trình tự cho chức năng "Xem lịch sử "* 18](#_Toc8487533)

[2.4.8. *Biểu đồ trình tự cho chức năng "Xem video "* 19](#_Toc8487534)

[2.5. Thiết kế mô hình dữ liệu 20](#_Toc8487535)

[2.5.1. *Mô hình quan hệ giữa các bảng* 20](#_Toc8487536)

[2.5.2. *Thiết kế chi tiết bảng* 21](#_Toc8487537)

[2.6. Thiết kế giao diện người dùng 27](#_Toc8487538)

[2.6.1. *Giao diện chức năng "Đăng nhập"* 27](#_Toc8487539)

[2.6.2. *Giao diện chức năng "Đăng kí "* 27](#_Toc8487540)

[2.6.3. *Giao diện chính* 28](#_Toc8487541)

[2.6.4. *Giao diện chức năng "Luyện chữ viết"* 28](#_Toc8487542)

[2.6.5. *Giao diện chức năng "Thi thử"* 29](#_Toc8487543)

[2.6.6. *Giao diện chức năng "Xem lịch sử"* 29](#_Toc8487544)

[2.6.7. *Giao diện chức năng "Xem lượt thi cao nhất "* 30](#_Toc8487545)

[2.6.8. *Giao diện chức năng "Xem video "* 30](#_Toc8487546)

[2.6.9. *Giao diện chức năng "Kết quả bài test "* 31](#_Toc8487547)

[3. Phân tích hệ thống 31](#_Toc8487548)

[3.1. Kiến trúc class trong hệ thống 31](#_Toc8487549)

[3.1.1. *Biểu đồ lớp chi tiết cho chức năng “ Thi thử ”* 31](#_Toc8487550)

[3.1.2. *Biểu đồ lớp chi tiết cho chức năng “ Luyện viết chữ ”* 32](#_Toc8487551)

[3.1.3. *Biểu đồ lớp chi tiết cho chức năng “ Xem lịch sử ”* 34](#_Toc8487552)

[3.1.4. *Biểu đồ lớp chi tiết cho chức năng “ Đăng kí ”* 34](#_Toc8487553)

[4. Tổng kết 36](#_Toc8487554)

[4.1. Các tính năng đã hoàn thiện 36](#_Toc8487555)

[4.2. Ưu điểm 36](#_Toc8487556)

[4.3. Nhược điểm 36](#_Toc8487557)

[4.4. Hướng phát triển 36](#_Toc8487558)

[4.5. Kết quả đạt được sau môn học 36](#_Toc8487559)

# Giới thiệu

# Mục đích

* Thiết kế và xây dựng 1 website hỗ trợ học tiếng nhật
* Ý tưởng : Xây dựng 1 website hỗ trợ học tiếng nhật . Người dùng sẽ có các chức năng như tạo tài khoản mới, đăng nhập, đăng xuất, xem lịch sử kết quả thi trước đó, xem thông tin lượt thi có điểm số cao nhất . Đặc biệt, người dùng có thể trực tiếp thực hiện bài test, hệ thống xử lí đưa ra kết quả trực tiếp cho người dùng và hộ trợ người dùng tập viết tiếng nhật, hệ thống nhận dạng chữ viết gần đúng và đưa ra đáp án chính xác nhất, giúp ích cho việc mới bắt đầu làm quen với tiếng Nhật.

# Phạm vi

* Tài liệu mô tả thiết kế phần mềm này là một bản thiết kế chi tiết về hệ thống website hỗ trợ học tiếng Nhật . Trải qua các bước như thiết kế kiến trúc, thiết kế giao diện, thiết kế mô hình dữ liệu, người đọc hoàn toàn có thế hình dung phần mềm được xây dựng, thiết kế như thế nào. Dựa vào tài liệu này người lập trình hoặc nhóm người lập trình có thể xây dựng các modules và lập trình theo cấu trúc mà bản thiết kế này cung cấp.
* Các đối tượng sử dụng tài liệu này là sinh viên theo học bộ môn CNPM và những nhà phát triển phần mềm.

# Các tính năng

* Đăng nhập
* Đăng xuất
* Đăng ký
* Thi thử
* Nghe video
* Luyện viết chữ
* Xem lịch sử thi
* Xem lượt thi cao nhất

# Công nghệ sử dụng

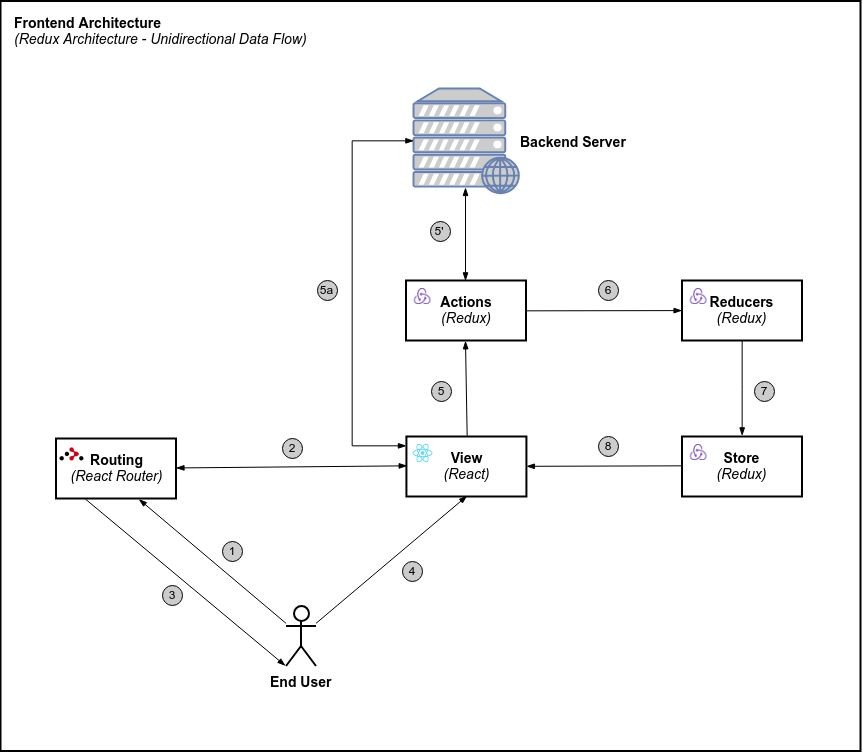
Nhóm lựa chọn phát triển cả phần Frontend và Backend. Giữa Frontend và Backend sẽ giao tiếp với nhau qua các HTTP Requests và HTTP Responses, với dạng dữ liệu là JSON.

* Đối với Frontend, nhóm lựa chọn sử dụng React kết hợp với Redux. Như vậy, ngôn ngữ lập trình chủ yếu là JavaScript, HTML và CSS.
* Đối với Backend, nhóm lựa chọn sử dụng Laravel Framework. Ngôn ngữ lập trình chính sẽ là PHP.

# Phân tích thiết kế

# Thiết kế tổng quan

# *Kiến trúc Frontend*

Kiến trúc gốc ở Frontend là Flux. Nhưng khi sử dụng các thư viện khác nhau, và đặc biệt là Redux thì kiến trúc Frontend thực tế của nhóm có sự thay đổi và được mô tả tổng quan trong **Hình 2.3**. Cụ thể luồng hoạt động như sau:

##### Hình 2.3. Biểu đồ mô tả kiến trúc Frontend

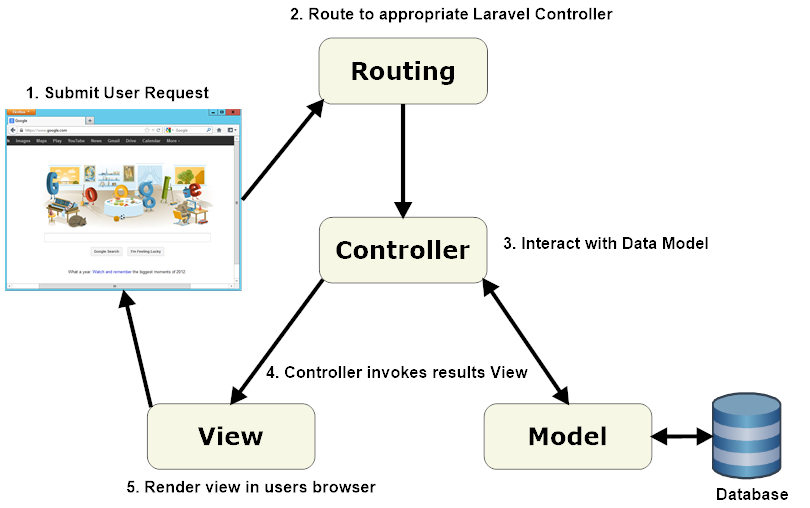
* + - 1. Người dùng truy cập vào một trang bất kỳ ở Browser. Yêu cầu truy cập này sẽ được chuyển đến React Router để xử lý.
      2. React Router sẽ phụ trách tìm đúng View (thực chất là một React Component) ứng với đường dẫn mà người dùng truy cập vào.
      3. Sau khi React Router đã tìm được đúng View ứng với đường dẫn thì phần View đó sẽ được hiển thị cho người dùng.
      4. Người dùng sẽ tương tác trực tiếp trên View. Các tương tác của người dùng có thể kích hoạt một Action của Redux nhưng cũng có thể không.
      5. Tương tác của người dùng trên View làm kích hoạt một Action (tương tác loại này có thể là submit form, đăng xuất...). (5') Nếu tương tác của người dùng đòi hỏi sự

tương tác với phía backend (backend server) thì trước khi Action được gửi tới Reducer ở bước 6, HTTP Request sẽ được thực hiện ở đây. Sau đó, khi backend server đã xử lý xong thì sẽ trả lại HTTP Response phù hợp. Nếu HTTP Response làm thay đổi trạng thái của ứng dụng thì sẽ lại tiếp tục thực hiện bước 6. (5a) Ngoài ra có thể trao đổi trực tiếp dữ liệu từ 1 React Component với backend server mà không cần thông qua Redux.

* + - 1. Sau đó Action sẽ được gửi tới Reducer tương ứng.
      2. Reducer là nơi quyết định xem sẽ xử lý Action vừa được gửi đến như thế nào. Xử lý ở đây chính là làm thay đổi trạng thái của ứng dụng (App State) phù hợp với loại Action và dữ liệu kèm theo nó. App State được cập nhật và lưu lại ở trong Redux Store (toàn bộ App State sẽ được lưu ở trong 1 Store duy nhất).
      3. App State sẽ quyết định xem View hiển thị những gì. Khi App State bị thay đổi thì cũng sẽ kéo theo View thay đổi.

Bước 2 và 3 đều lặp lại giống nhau với bất kỳ truy cập nào nên để ngắn gọn thì sẽ không mô tả ở trong các biểu đồ tương tác.

# *Kiến trúc Backend*



**Luồng xử lí trong mô hình MVC:**

* Mọi request từ phía người dùng đều phải qua Route
* Controller nhận yêu cầu này và đưa lệnh xử lý yêu cầu đó. Các lệnh thực thi với phần View thì cập nhật hoặc phục vụ yêu cầu trang web, với Model thì để trình diễn logic. Ta giả sử lệnh yêu cầu có yếu tố logic
* Model thực thi phần logic được lấy ra từ cơ sở dữ liệu và gửi trả lại phản hồi dựa trên hướng dẫn của Controller
* Controller truyền dữ liệu ra phần View, cập nhật giao diện hiển thị cho người dùng

# Các tác nhân

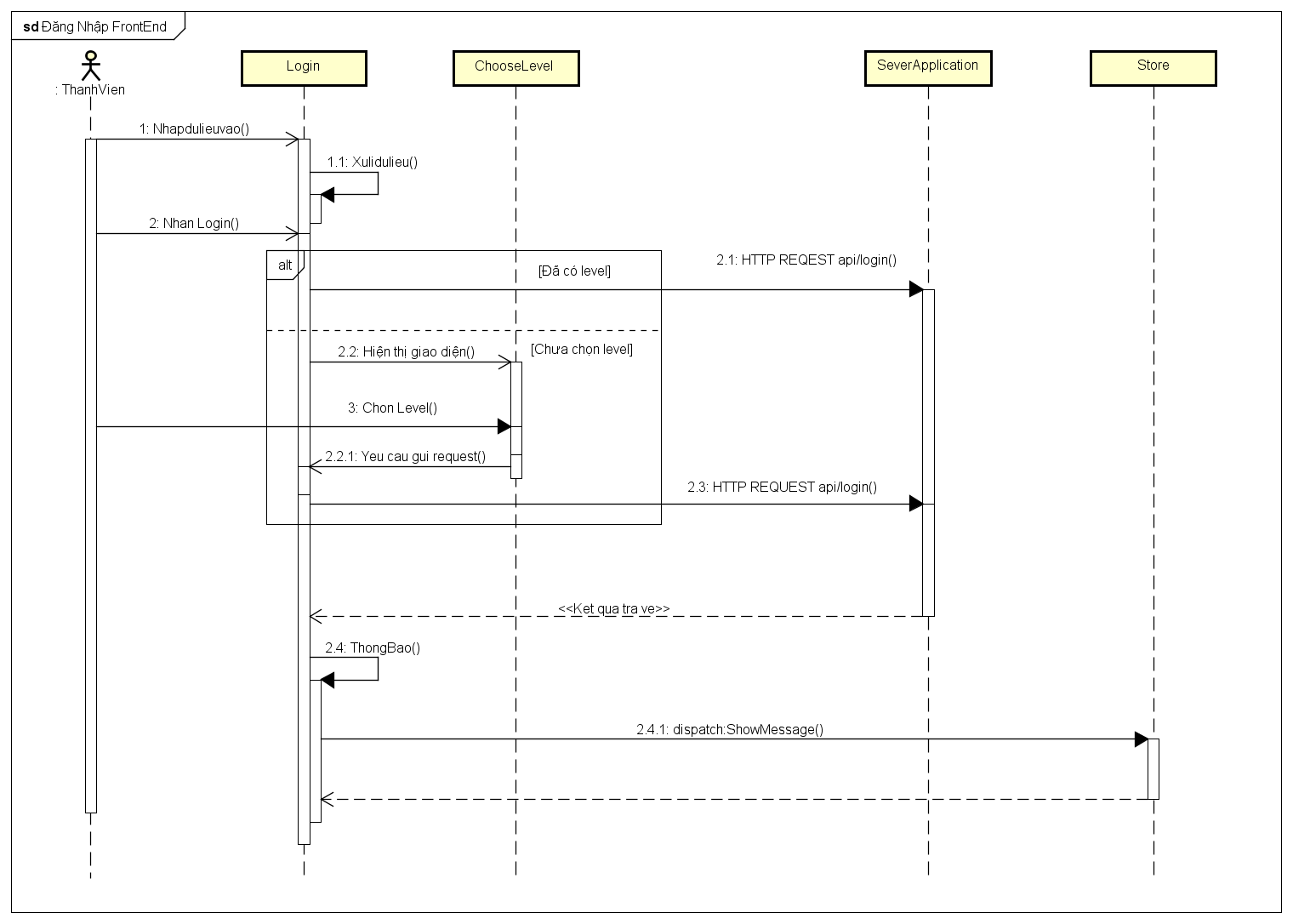
Hệ thống có 2 tác nhân là khách, thành viên

# Biểu đồ use case tổng quan

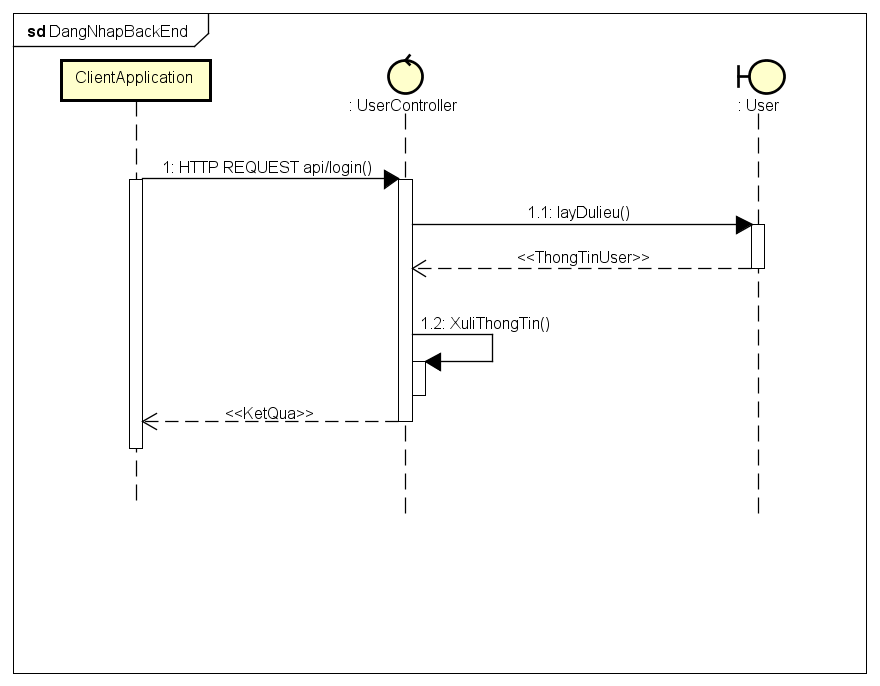
* 

# Biểu đồ tương tác

# *Biểu đồ trình tự cho chức năng "Đăng nhập"*

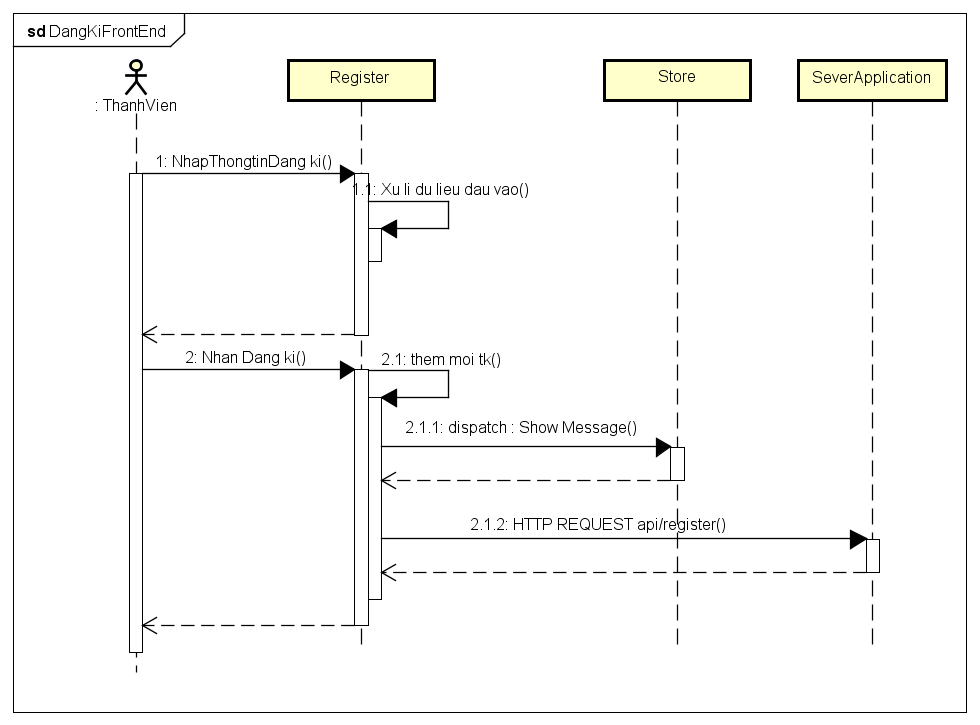
******

##### Hình 2.4.1a. Biểu đồ trình tự cho chức năng "Đăng nhập" (Frontend)



##### Hình 2.4.1b. Biểu đồ trình tự cho chức năng "Đăng nhập" (Backend)

# *Biểu đồ trình tự cho chức năng "Đăng kí "*

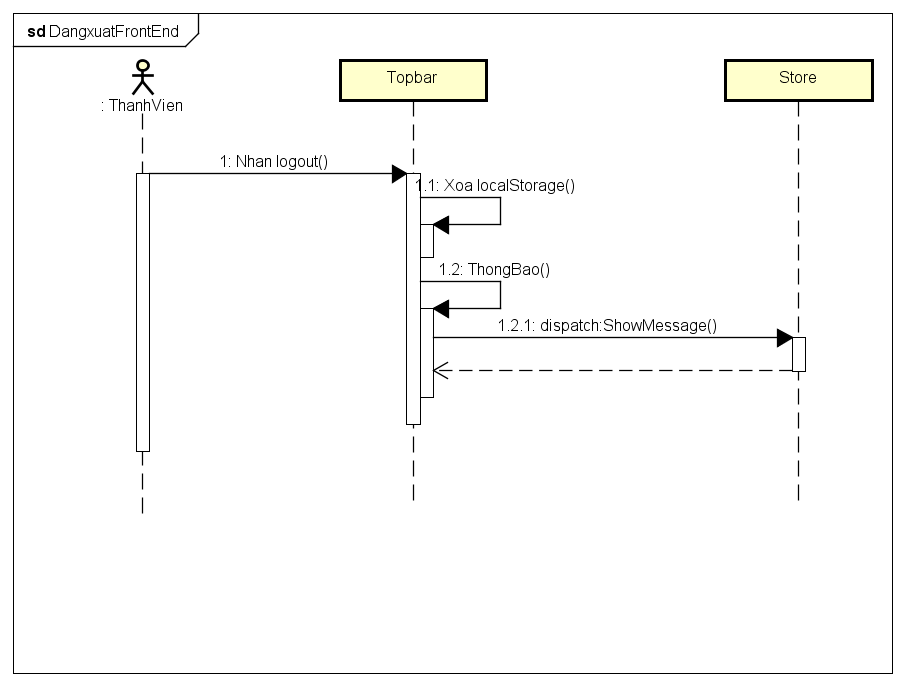
******

##### Hình 2.4.2a. Biểu đồ trình tự cho chức năng "Đăng kí " (Frontend)

##### C:\Users\hoan\Desktop\Dang Ki BackEnd.png

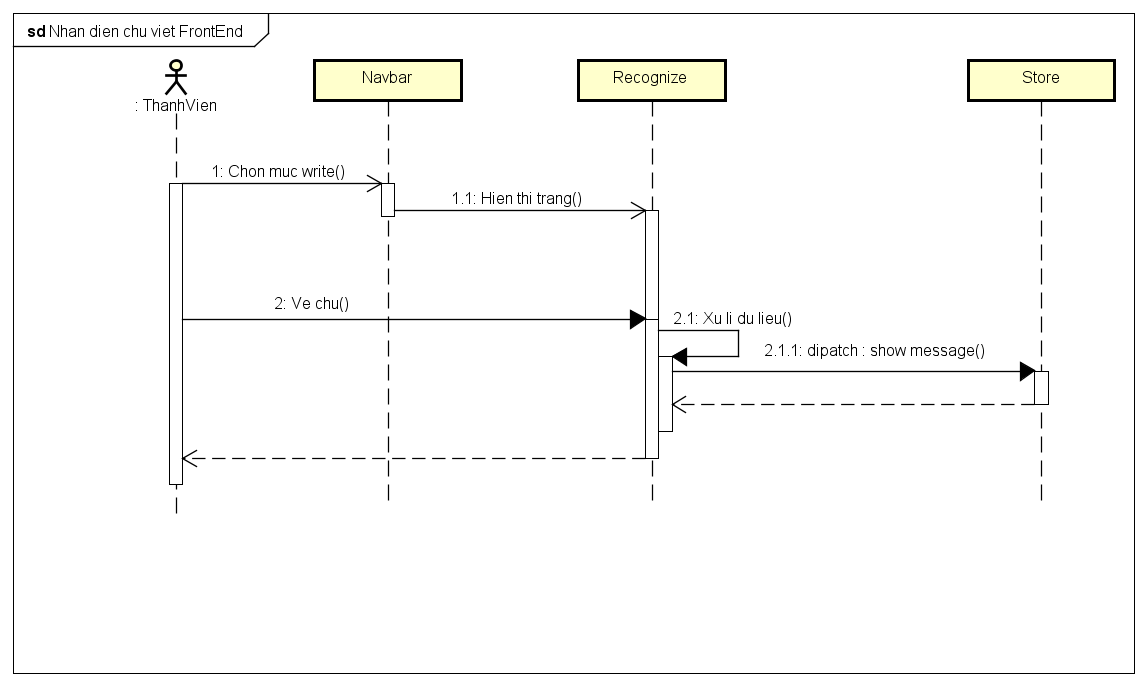
##### Hình 2.4.2b. Biểu đồ trình tự cho chức năng "Đăng kí " (Backend)

# *Biểu đồ trình tự cho chức năng "Đăng xuất"*

******

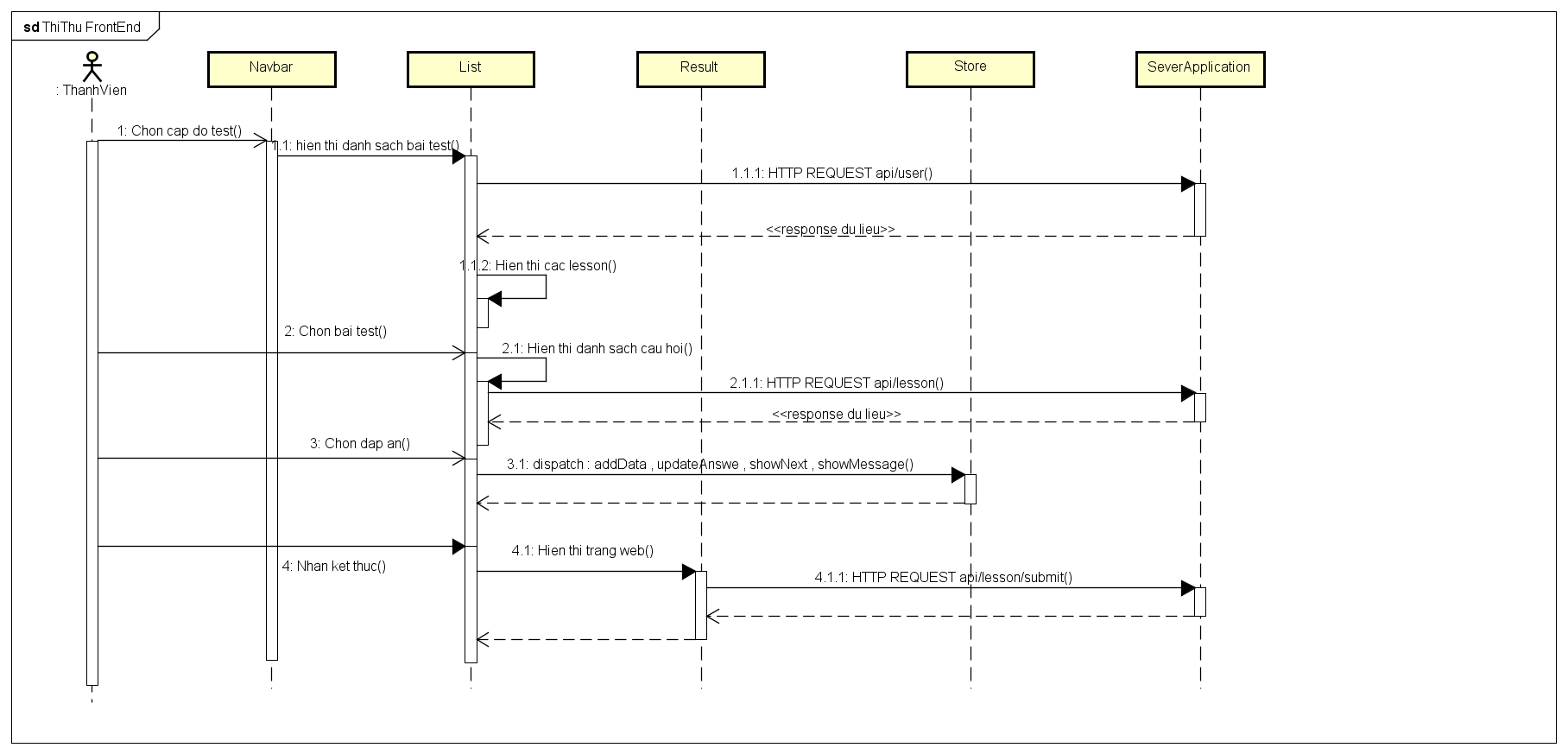
##### Hình 2.4.3. Biểu đồ trình tự cho chức năng "Đăng xuất" (Fontend)

# *Biểu đồ trình tự cho chức năng "Luyện viết chữ "*

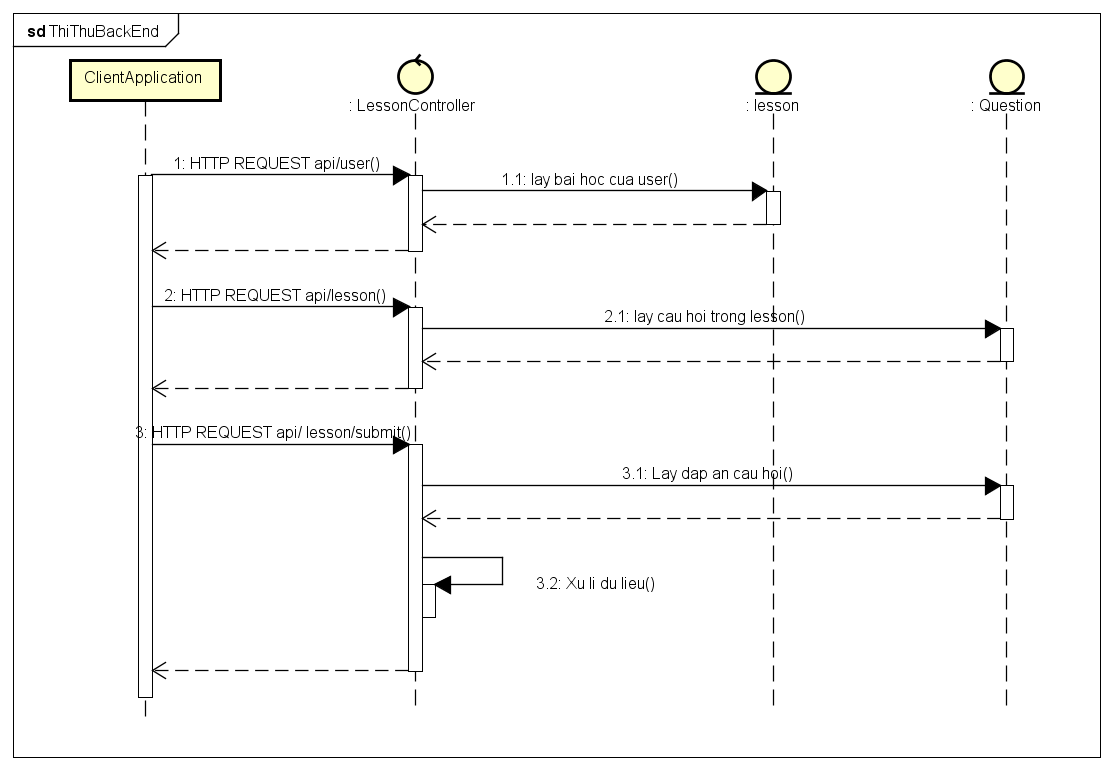
**

##### Hình 2.4.4. Biểu đồ trình tự cho chức năng "Luyện chữ viết " ( FrontEnd )

# *Biểu đồ trình tự cho chức năng "Thi Thử "*

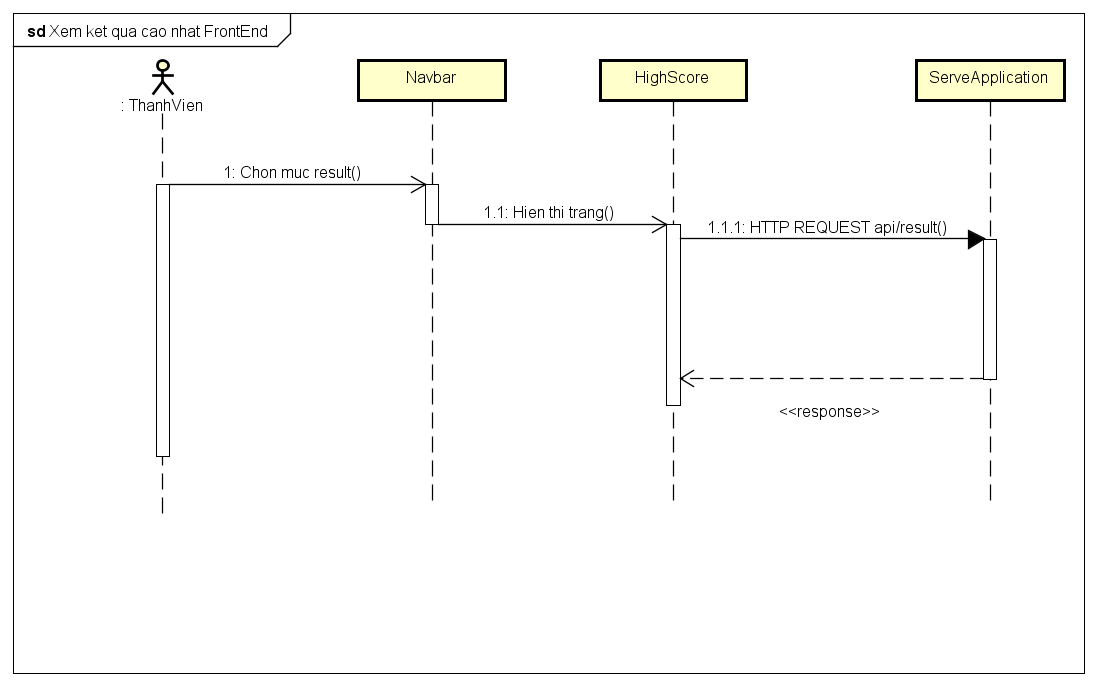
**

##### Hình 2.4.5a. Biểu đồ trình tự cho chức năng "Thi thử" (Frontend)

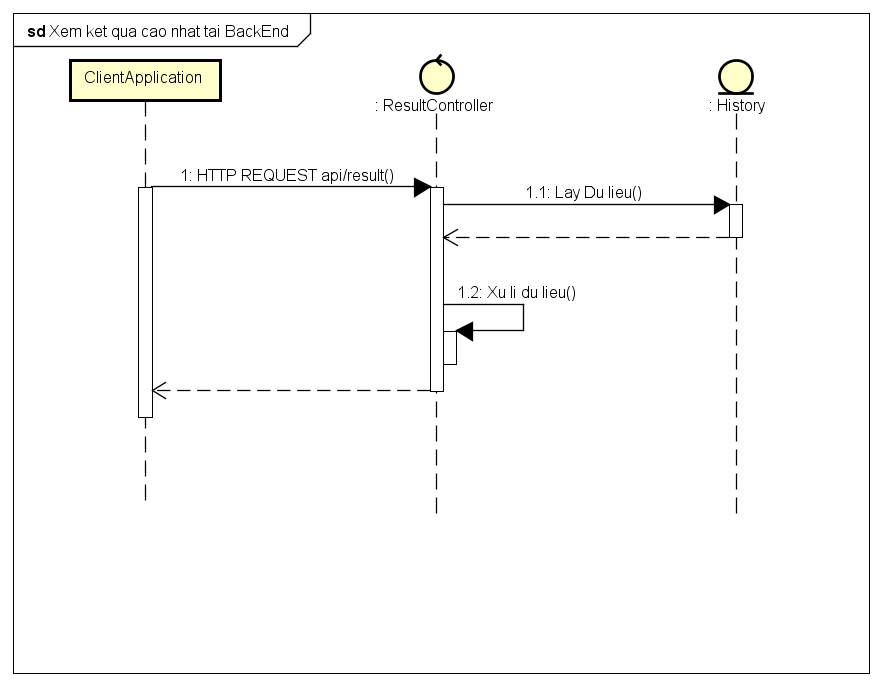
**

##### Hình 2.4.5b. Biểu đồ trình tự cho chức năng "Thi thử" (Backend)

# *Biểu đồ trình tự cho chức năng "Xem lượt thi cao nhất "*

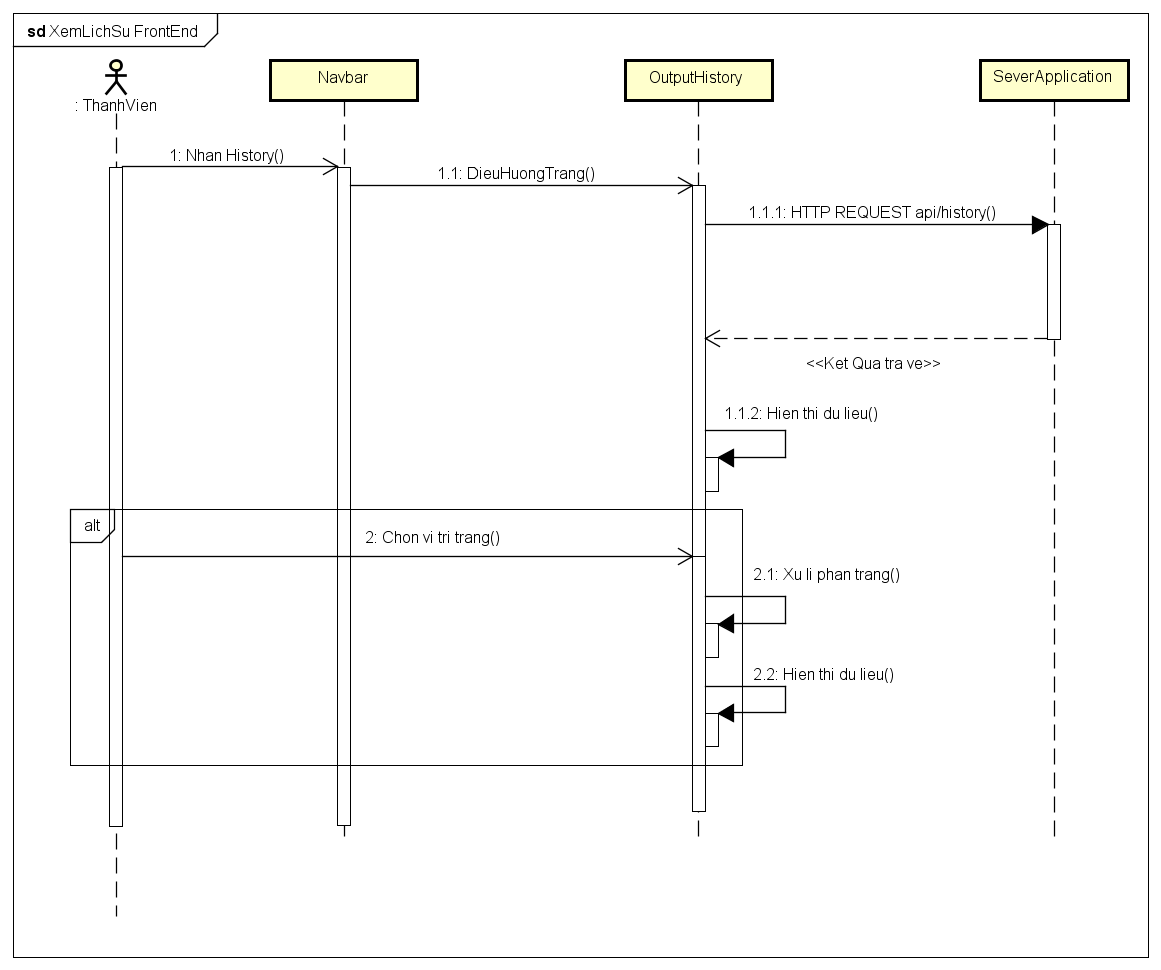
**

##### Hình 2.4.6a. Biểu đồ trình tự cho chức năng "Xem lượt thi cao nhất" (Frontend)

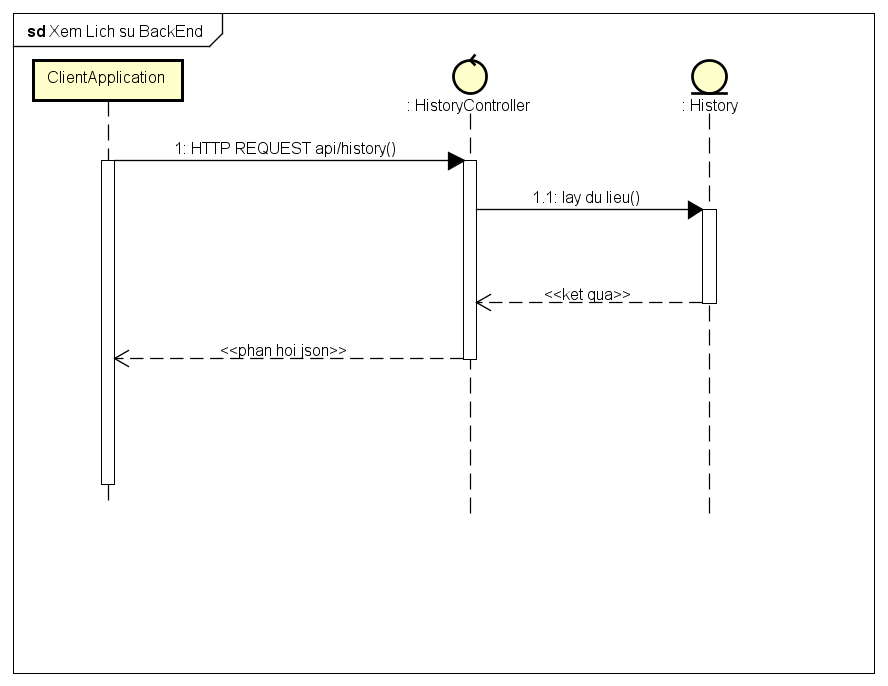
**

##### Hình 2.4.6b. Biểu đồ trình tự cho chức năng "Xem lượt thi cao nhất " (Backend)

# *Biểu đồ trình tự cho chức năng "Xem lịch sử "*

**

##### Hình 2.4.7a. Biểu đồ trình tự cho chức năng "Xem lịch sử " (Frontend)

**

##### Hình 2.4.7b. Biểu đồ trình tự cho chức năng "Xem lịch sử " (Backend)

# *Biểu đồ trình tự cho chức năng "Xem video "*

**

##### Hình 2.4.8a. Biểu đồ trình tự cho chức năng "Xem video " Fontend

# Thiết kế mô hình dữ liệu

# *Mô hình quan hệ giữa các bảng*

**

##### Hình 3.1: Mô hình quan hệ giữa các bảng

# *Thiết kế chi tiết bảng*

#### Bảng “**question**” lưu thông tin về câu hỏi Fill

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TT** | **Tên trường** | **Kiểu dữ liệu** | **Ghi chú** | **Ràng buộc** | **Mô tả** |
| 1 | id | int(10) | Khóa chính Duy nhất Tự động |  | Id tự tăng |
| 2 | question | varchar (255) | Duy nhất Bắt buộc |  | Tên câu hỏi Fill |
| 3 | answer | varchar (255) | Bắt buộc |  | Câu trả lời |
| 4 | word | varchar(255) | Bắt buộc |  | Chuỗi thứ tự đáp án |
| 5 | image | varchar(255) | Bắt buộc |  | Giá trị hình ảnh |

#### Bảng “**users**” lưu thông tin người dùng

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TT** | **Tên trường** | **Kiểu dữ liệu** | **Ghi chú** | **Ràng buộc** | **Mô tả** |
| 1 | id | bigint (20) | Khóa chính Duy nhất Tự động |  | Id tự tạo bởi Hibernate |
| 2 | level | Int(10) |  |  | Trình độ người test |
| 3 | password | varchar (255) | Bắt buộc | Ít nhất 8 kí tự | Mật khẩu đã mã hóa bằng BCrypt |
| 4 | name | varchar (40) | Bắt buộc | Từ 3 đến 40 kí tự | Họ tên người dùng |
| 5 | email | varchar(255) | Bắt buộc |  | Email người dùng |

#### Bảng “**questionabcd**” lưu thông tin tin về câu hỏi abcd

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TT** | **Tên trường** | **Kiểu dữ liệu** | **Ghi chú** | **Ràng buộc** | **Mô tả** |
| 1 | id | int(10) | Khóa chính Duy nhất Tự động |  | Id tự tăng |
| 2 | remaji | varchar (60) |  |  | Tên remaji |
| 3 | image | varchar(255) |  |  | Đường dẫn hình ảnh |
| 4 | question | varchar(255) |  |  | Tên câu hỏi |
| 5 | answer | varchar(255) |  |  | Tên câu trả lời |
| 6 | A | varchar(255) |  |  | Đáp án A |
| 7 | B | varchar(255) |  |  | Đáp án B |
| 8 | C | varchar(255) |  |  | Đáp án C |
| 9 | D | varchar(255) |  |  | Đáp án D |

#### Bảng “**result**” lưu thông lịch sử bài thi người dùng

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TT** | **Tên trường** | **Kiểu dữ liệu** | **Ghi chú** | **Ràng buộc** | **Mô tả & Ví dụ** |
| 1 | id | int(10) | Khóa chính Duy nhất Tự động |  | Id tự tăng |
| 2 | id\_lesson | int(10) | Bắt buộc , Khóa Phụ | Số dương | Id phân biệt của lesson |
| 3 | id\_user | int(10) | Bắt buộc , Khóa Phụ | Số dương | Id phân biệt của user |
| 4 | mark | varchar(255) |  |  | Điểm bài test |

#### Bảng “note” ghi chú câu hỏi Fill được note lại

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TT** | **Tên trường** | **Kiểu dữ liệu** | **Ghi chú** | **Ràng buộc** | **Mô tả & Ví dụ** |
| 1 | id | bigint (20) | Khóa chính Duy nhất Tự động |  | Id tự tăng |
| 2 | id\_user | varchar (7) | Khóa Phụ | Số dương | Id phân biệt user |
| 3 | id\_question | varchar (255) | Khóa Phụ | Số dương | Id phân biệt câu hỏi |

#### Bảng “**noteabcd**” ghi chú câu hỏi abcd được note lại

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TT** | **Tên trường** | **Kiểu dữ liệu** | | **Ghi chú** | **Ràng buộc** | **Mô tả & Ví dụ** |
| 1 | id | int(10) | | Khoá chính |  | Id tự tăng |
| 2 | id\_user | int(10) | | Khóa Phụ , Bắt buộc | Số dương | Id phân biệt người dùng |
| 3 | id\_question | int(10) | | Khóa Phụ , bắt buộc | Số dương | Id phân biệt câu hỏi abcd |
|  | | |

#### Bảng “fill” lưu mối liên hệ giữa lesson và question

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TT** | **Tên trường** | **Kiểu dữ liệu** | **Ghi chú** | **Ràng buộc** | **Mô tả & Ví dụ** |
| 1 | id | int(10) | Khoá chính |  | Id tự tăng |
| 2 | id\_lesson | int(10) | Khóa Phụ , bắt buộc | Số dương | Id của bài học |
| 3 | id\_question | int(10) | Khóa Phụ , bắt buộc | Số dương | Id phân biệt câu hỏi fill |

#### Bảng “choose” lưu thông tin học phần tiên quyết

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TT** | **Tên trường** | **Kiểu dữ liệu** | **Ghi chú** | **Ràng buộc** | **Mô tả & Ví dụ** |
| 1 | id | int(10) | Khoá chính |  | Id tự tăng |
| 2 | id\_lesson | int(10) | Khóa phụ , bắt buộc | Số dương | Id phân biệt bài học |
| 3 | id\_questionabcd | int(10) | Khóa Phụ , bắt buộc | Số dương | Id phân biệt câu hỏi abcd |

#### Bảng “lesson” lưu thông tin bài học

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TT** | **Tên trường** | **Kiểu dữ liệu** | **Ghi chú** | **Ràng buộc** | **Mô tả & Ví dụ** |
| 1 | id | int (10) | Khoá chính |  | Id tự tăng |
| 2 | rule\_id | int (10) | Bắt buộc , Khóa Phụ | Số dương | Mã phân biệt lesson thuộc mục nào |
| 3 | name | varchar(255) |  |  | Tên bài học |

#### Bảng “rule” lưu thông tin cấp độ của bài học

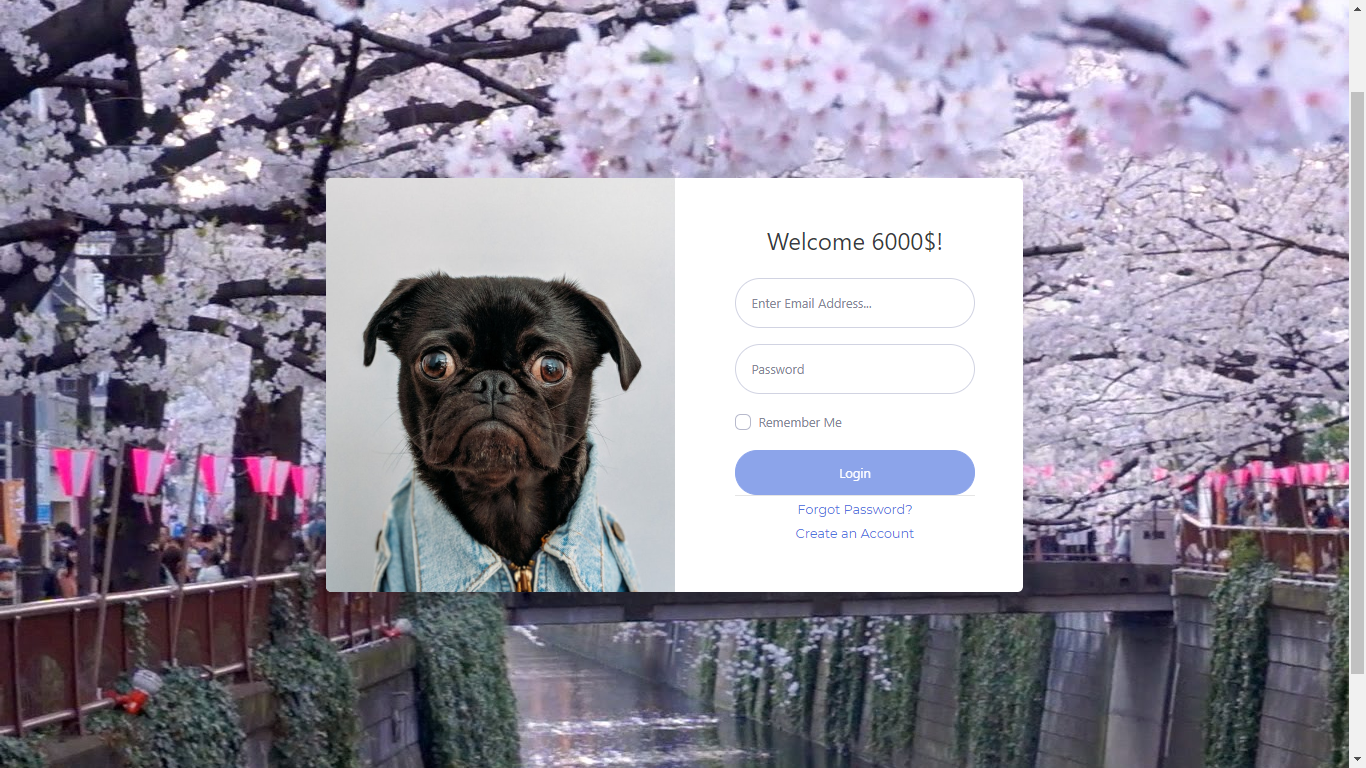
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TT** | **Tên trường** | **Kiểu dữ liệu** | **Ghi chú** | **Ràng buộc** | **Mô tả & Ví dụ** |
| 1 | id | int(10) | Khoá chính |  | Id tự tăng |
| 2 | name | varchar(255) |  |  | Tên của Rule |

#### Bảng “history” lưu kết quả của bài thi

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TT** | **Tên trường** | **Kiểu dữ liệu** | **Ghi chú** | **Ràng buộc** | **Mô tả & Ví dụ** |
| 1 | id | int(10) | Khoá chính |  | Id tự tăng |
| 2 | id\_lesson | int(10) | Khóa Phụ , bắt buộc | Só dương | Id phân biệt các lesson |
| 3 | id\_user | int(10) | Khóa Phụ , bắt buộc | Số dương | Id phân biệt user |
| 4 | mark | int(10) | Bắt buộc | từ 1-> 10 | Điểm |
| 5 | date | date | Bắt buộc |  | Ngày kết thúc bài test |

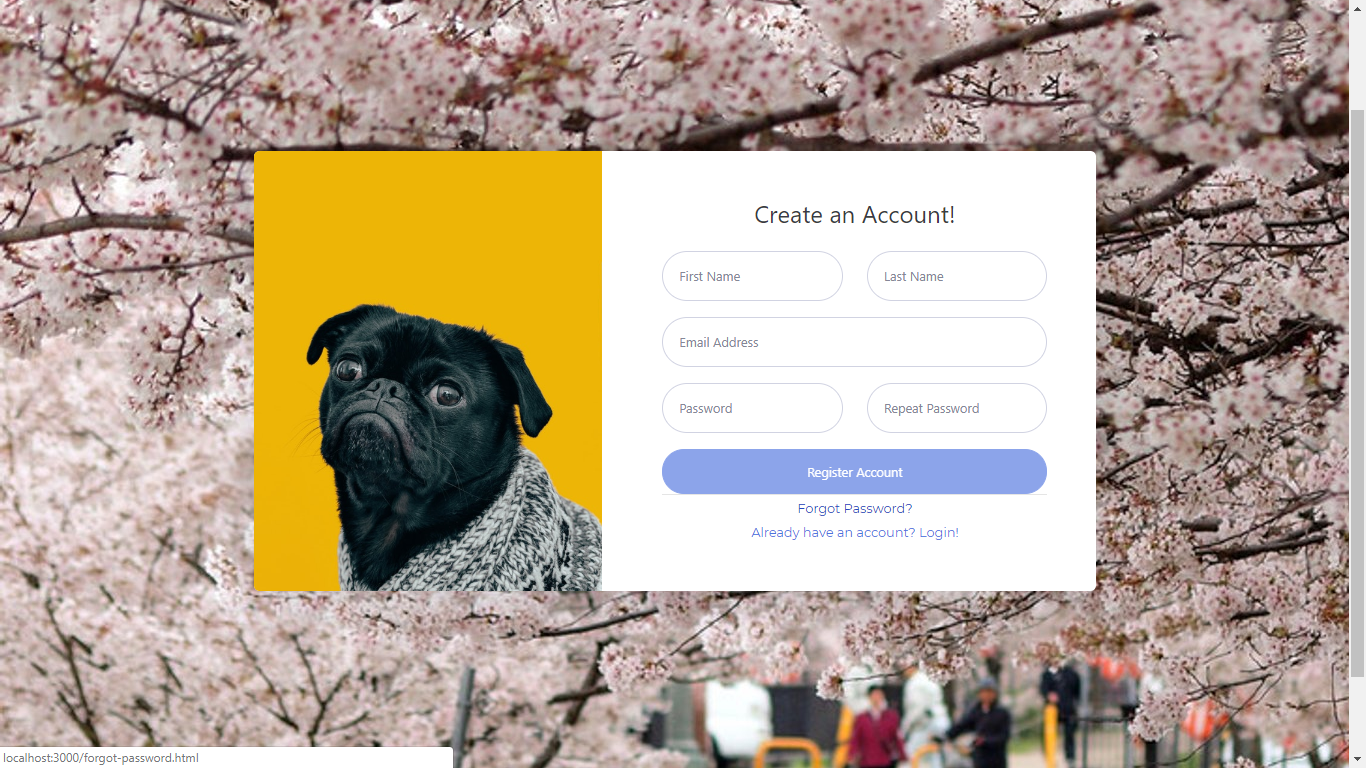
# Thiết kế giao diện người dùng

# *Giao diện chức năng "Đăng nhập"*



##### Hình 2.6.1. Giao diện chức năng "Đăng nhập"

# *Giao diện chức năng "Đăng kí "*



##### Hình 2.6.2. Giao diện chức năng "Đăng kí"

# *Giao diện chính*



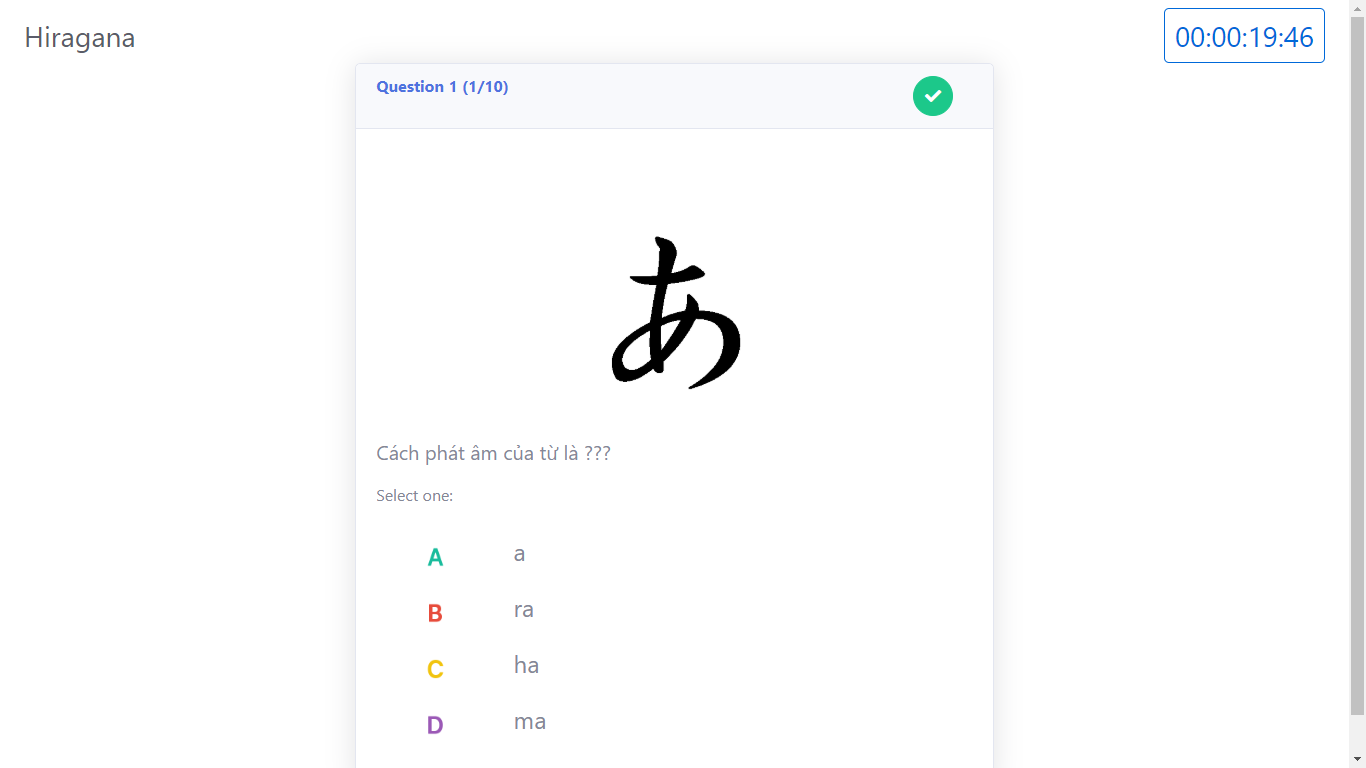
##### Hình 2.6.3. Giao diện chính

# *Giao diện chức năng "Luyện chữ viết"*



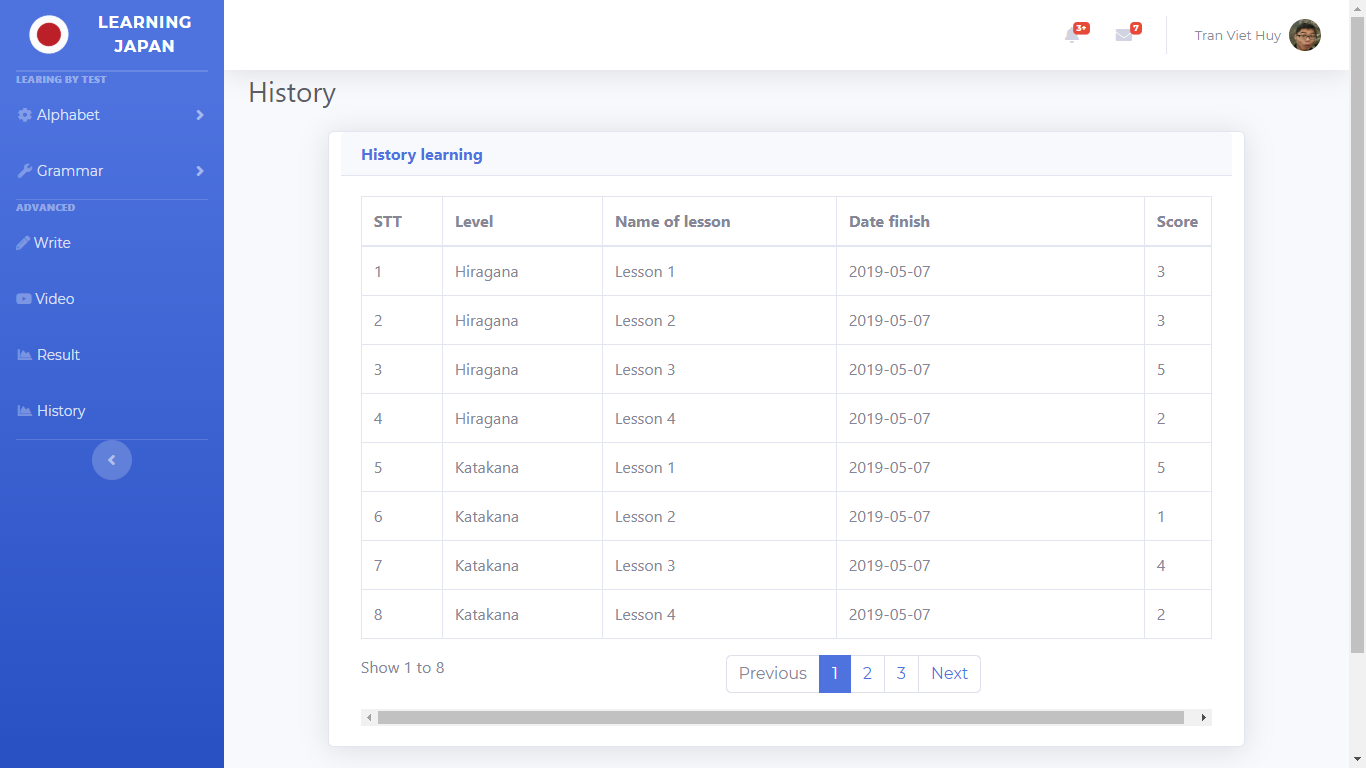
##### Hình 2.6.4. Giao diện chức năng "Luyện chữ viết"

# *Giao diện chức năng "Thi thử"*



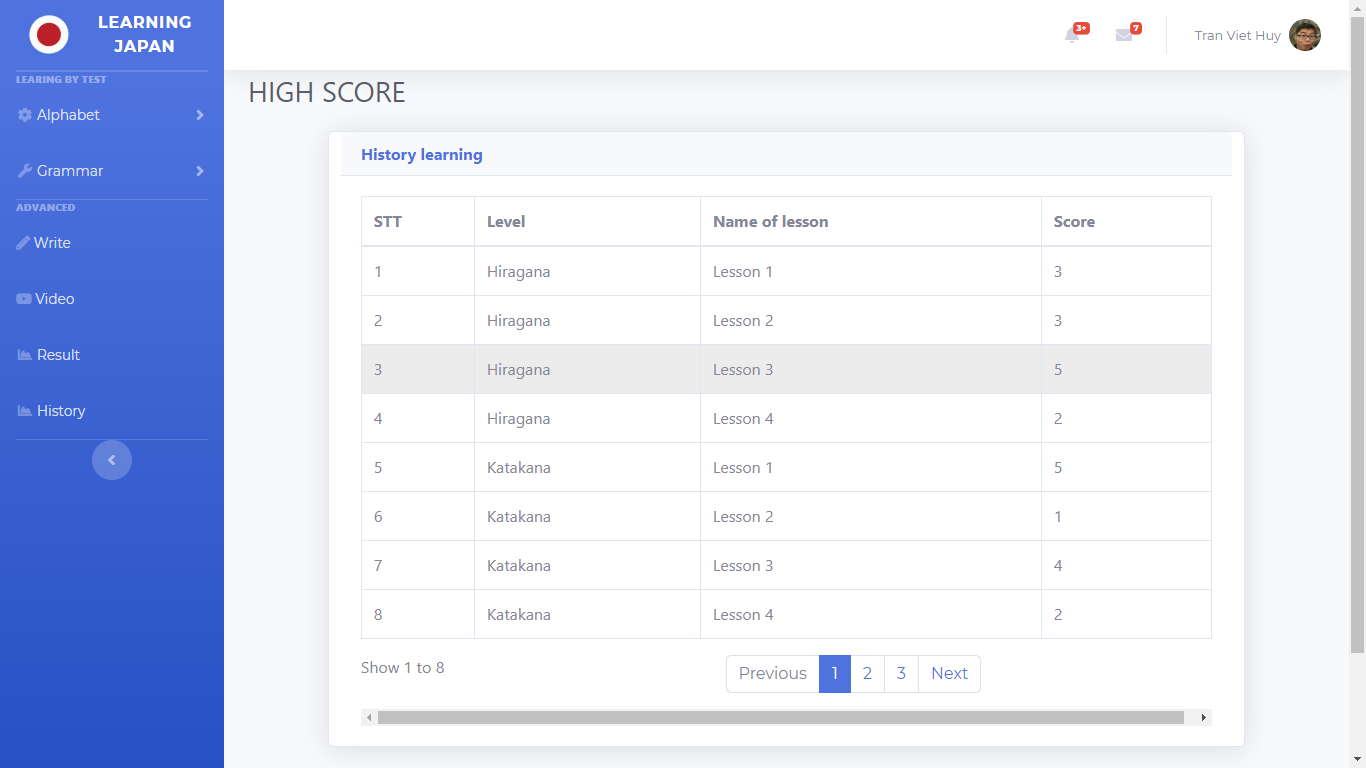
##### Hình 2.6.5. Giao diện chức năng "Thi thử"

# *Giao diện chức năng "Xem lịch sử"*



##### Hình 2.6.6. Giao diện chức năng "Xem lịch sử"

# *Giao diện chức năng "Xem lượt thi cao nhất "*



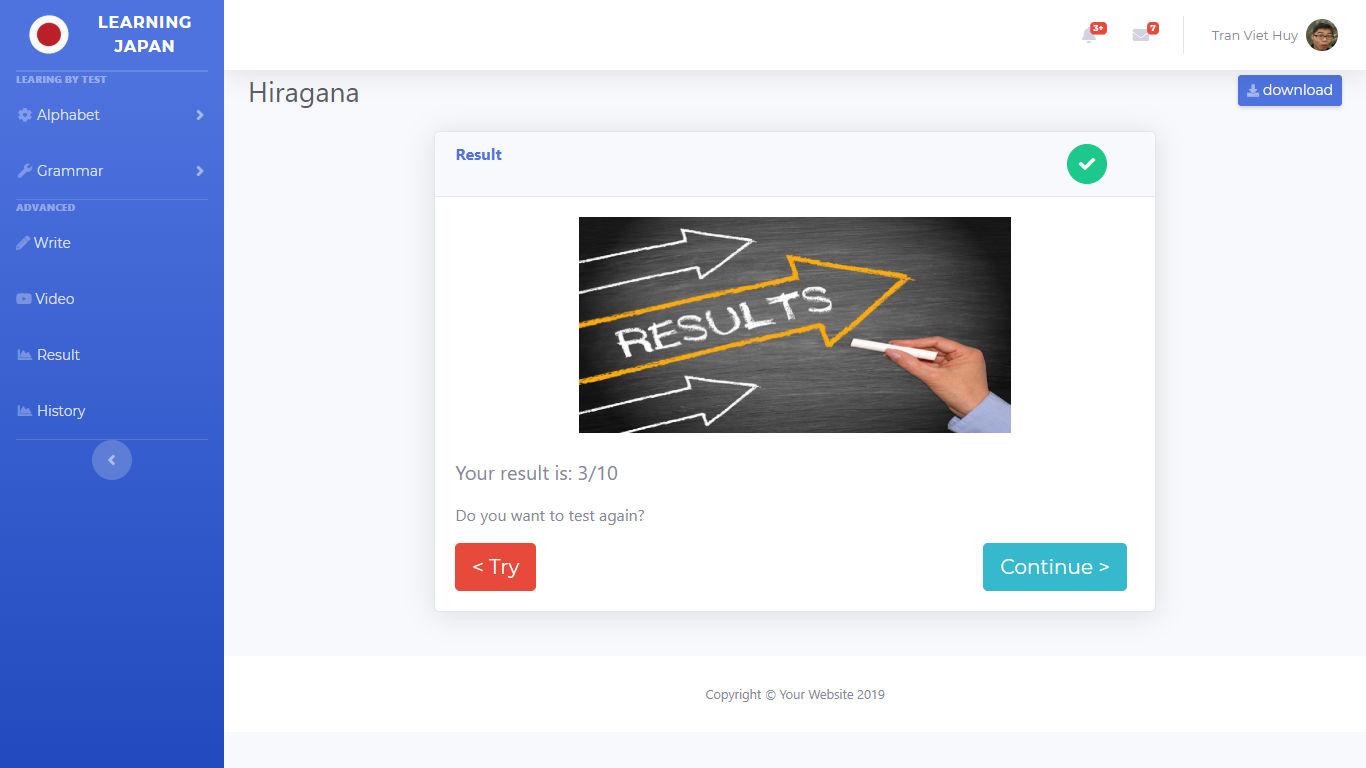
##### Hình 2.6.7. Giao diện chức năng "xem lượt thi cao nhất "

# *Giao diện chức năng "Xem video "*



##### Hình 2.6.8. Giao diện chức năng "xem video"

# *Giao diện chức năng "Kết quả bài test "*



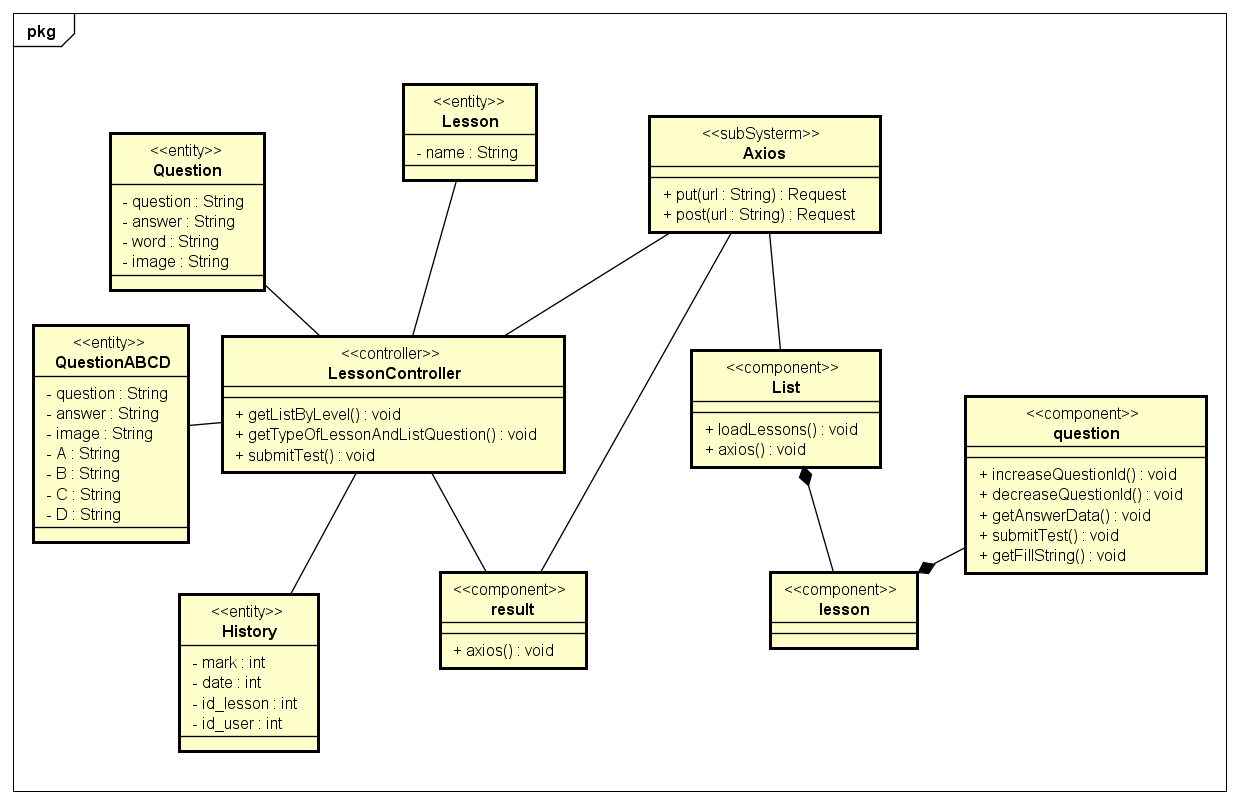
##### Hình 2.6.9. Giao diện chức năng "Kết quả bài test "

# Phân tích hệ thống

[**https://github.com/hungcon/Learning\_Japanese\_Online**](https://github.com/hungcon/Learning_Japanese_Online)

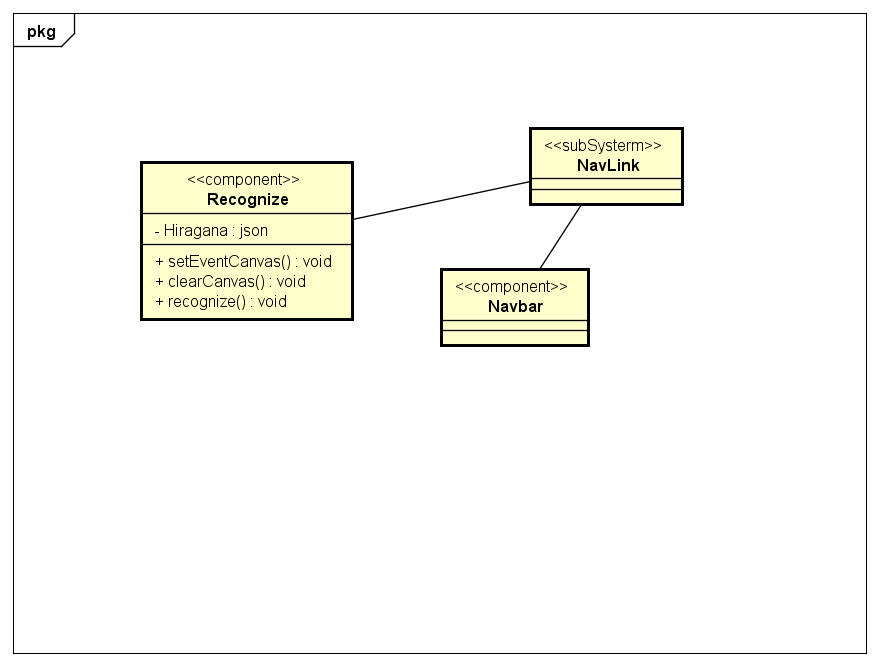
# Kiến trúc class trong hệ thống

# *Biểu đồ lớp chi tiết cho chức năng “ Thi thử ”*

**

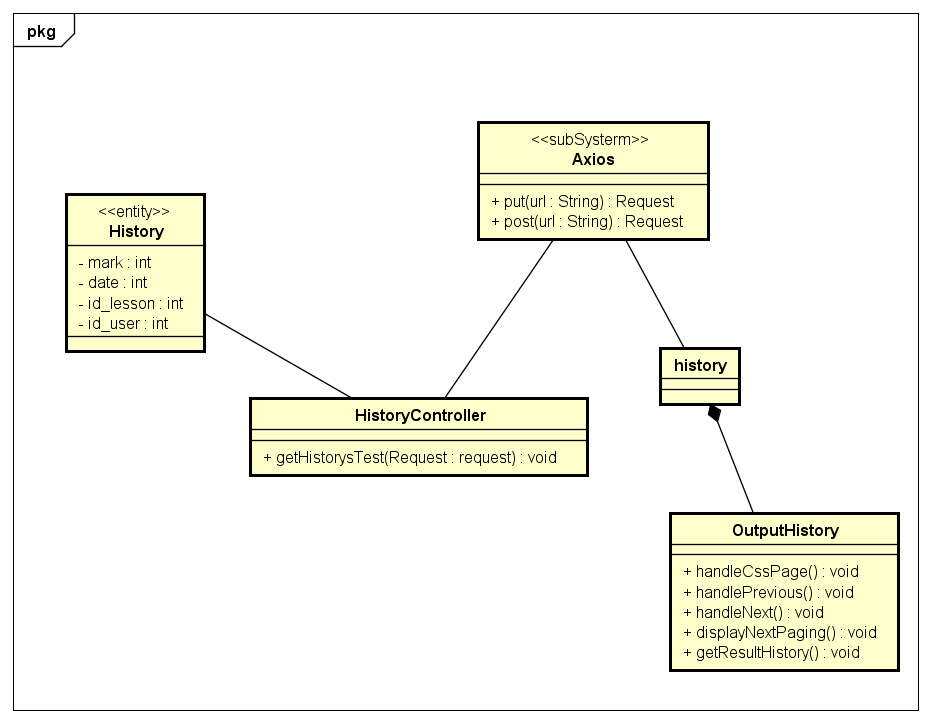
##### Hình 3.1.1. Biểu đồ lớp chi tiết cho chức năng thi thử

# *Biểu đồ lớp chi tiết cho chức năng “ Luyện viết chữ ”*

**

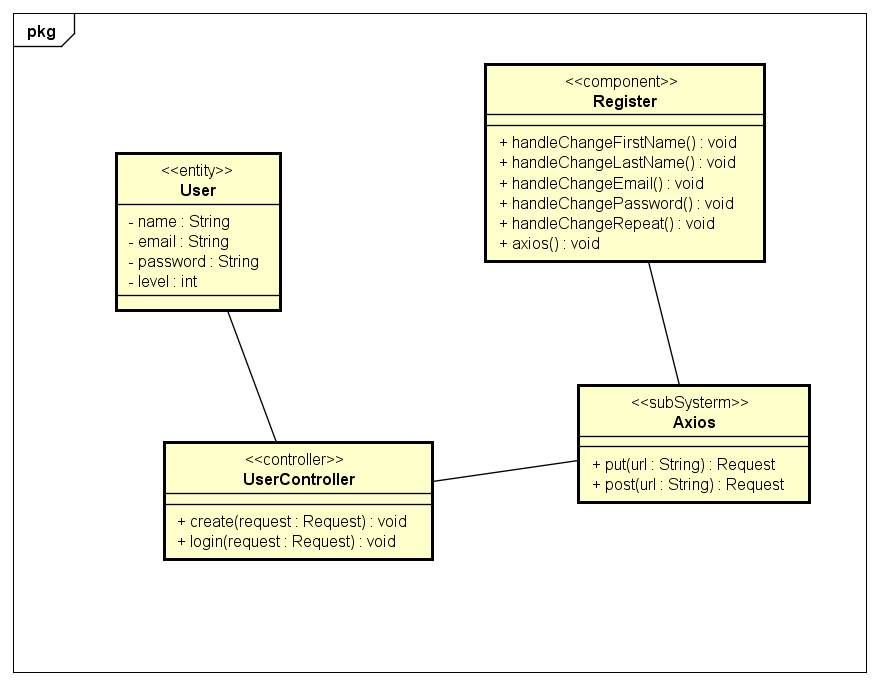
##### Hình 3.1.2. Biểu đồ lớp cho nhóm chức năng luyện viết chữ

# *Biểu đồ lớp chi tiết cho chức năng “ Xem lịch sử ”*

**

##### Hình 3.1.3. Biểu đồ lớp chi tiết cho chức năng xem lịch sử

# *Biểu đồ lớp chi tiết cho chức năng “ Đăng kí ”*

**

##### Hình 3.1.4. Biểu đồ lớp chi tiết cho chức năng đăng kí

# Tổng kết

# Các tính năng đã hoàn thiện

* Đăng nhập
* Đăng xuất
* Đăng ký
* Thi thử
* Nghe video
* Luyện viết chữ
* Xem lịch sử thi
* Xem lượt thi cao nhất

# Ưu điểm

- Hệ thống có giao diện đẹp mắt , phân chia bố cục nội dung rõ ràng, dễ nhìn

- Tốc độ xử lí dữ liệu nhanh sử dụng reactjs mà không cần load lại trang

- Có thông báo kết quả đến người dùng khi thực hiện xong các chức năng

- Dễ sử dụng và thực hiện các chức năng

# Nhược điểm

- Đầu vào xử lí thông tin người dùng còn hạn chế, tính bảo mật thông tin người dùng chưa cao

- Thông tin dữ liệu về bài test chưa nhiều

# Hướng phát triển

- Phát triển hệ thống có thể phục vụ nhiều lượt truy cập cùng 1 lúc,tăng tốc độ phản hồi khi điều hướng giữa các trang và hiện thị kết quả bài test.

- Bổ sung nhiều bài test hơn cho hệ thống

- Xây dựng chức năng cho phép dịch từ tiếng nhật sang tiếng việt

- Xây dựng chức năng chat trực tuyến, nơi mọi người có thể kết bạn , trò chuyện và trao đổi về kiến thức tiếng nhật

# Kết quả đạt được sau môn học

- Kĩ năng phân tích , thiết kế một website

- Biết cách sử dụng reactjs cùng laravel trong phát triển website

- Kỹ năng sử dụng git

- Kỹ năng làm việc nhóm hiệu quả