

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC AN GIANG**  
**KHOA CÔNG NGHỆ THÔNG TIN**

**THỰC TẬP CUỐI KHÓA**

**TÊN ĐỀ TÀI**

**Đơn vị thực tập**

**HỌ TÊN SV**

**AN GIANG, 04-2020**

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC AN GIANG**  
**KHOA CÔNG NGHỆ THÔNG TIN**

**THỰC TẬP CUỐI KHÓA**

**TÊN ĐỀ TÀI**

**Đơn vị thực tập**

**HỌ TÊN - MSSV**

**GIẢNG VIÊN HƯỚNG DẪN**

**AN GIANG, 04-2020**

## **NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

## **Giảng viên hướng dẫn**

(Ký và ghi rõ họ tên)

### Nội dung nhận xét:

- Đồng ý hay không đồng ý cho sinh viên báo cáo TTCK; Nếu không đồng ý cần ghi rõ lý do.
  - Kết quả đạt được so với yêu cầu;
  - Ý kiến khác (nếu có).

## **LỜI CẢM ƠN**

# TÓM TẮT

## MỤC LỤC

LỜI CẢM ƠN .....	ii
TÓM TẮT .....	iii
MỤC LỤC .....	iv
DANH SÁCH HÌNH ẢNH .....	vi
DANH SÁCH BẢNG BIỂU .....	vi
DANH SÁCH CÁC TỪ VIẾT TẮT .....	vii
CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI.....	1
1.1 Lý do chọn đề tài .....	1
1.2 Phát biểu bài toán .....	1
1.3 Mục tiêu của đề tài .....	1
1.4 Hướng tiếp cận .....	1
1.5 Phạm vi đề tài .....	1
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT .....	1
2.1 Giới Thiệu Laravel Framework.....	1
2.2 Lịch sử phát triển Laravel Framework .....	3
2.3 Giới thiệu chung về Laravel.....	4
2.3.1 Cấu trúc thư mục mới: .....	4
2.3.2 Contracts .....	5
2.3.3 Route cache .....	5
2.3.4 Route Middleware.....	5
2.3.5 Controller Method Injection .....	5
2.3.6 Authentication Scaffolding .....	5
2.3.7 From Requests .....	5
2.3.8 Simple Controller Request Validation.....	5
2.4 So sánh Laravel Framework và các Framework khác.....	5
2.4.1 Sự phổ biến của Laravel Framework .....	5
2.4.2 So sánh Laravel và Symfony 2 .....	6
2.4.3 So sánh Laravel và CodeIgniter.....	7
CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ .....	8
3.1 Phân tích thiết kế hệ thống thông tin.....	8
3.1.1 Sơ đồ chức năng.....	8
3.1.2 Sơ đồ Usecase .....	8
3.1.3 Đặt tả Usecase.....	11
3.1.4 Sơ đồ tuần tự .....	13
3.1.5 Thiết kế hệ thống.....	25

3.1.6 Thiết kế Cơ sở dữ liệu.....	26
3.1.7 Mô tả các bảng cơ sở dữ liệu .....	27
CHƯƠNG 4: THIẾT KẾ GIAO DIỆN .....	31

## DANH SÁCH HÌNH ẢNH

HÌNH 1. GIỚI THIỆU LARAVEL FRAMEWORK .....	2
HÌNH 2. THỐNG KÊ MỨC ĐỘ SỬ DỤNG CỦA NGƯỜI DÙNG ĐỐI VỚI CÁC FRAMEWORK NĂM 2015. ....	6
HÌNH 3. SƠ ĐỒ CHỨC NĂNG .....	8
HÌNH 4. SƠ ĐỒ USECASE TỔNG QUÁT .....	9
HÌNH 5. SƠ ĐỒ PHÂN RÃ USECASE QUẢN LÝ HỌC BỘNG. ....	10
HÌNH 6. SƠ ĐỒ PHÂN RÃ USECASE TRAO HỌC BỘNG.....	10
HÌNH 7. SƠ ĐỒ PHÂN RÃ USECASE THỐNG KÊ.....	11
HÌNH 8. SƠ ĐỒ TUẦN TỰ ADMIN ĐĂNG NHẬP. ....	14
HÌNH 9. DANH SÁCH HỌC BỘNG. ....	15
HÌNH 10. SƠ ĐỒ TUẦN TỰ DANH SÁCH HỌC BỘNG. ....	15
HÌNH 11. SƠ ĐỒ TUẦN TỰ THÊM HỌC BỘNG. ....	16
HÌNH 12. SƠ ĐỒ TUẦN TỰ IMPORT HỌC BỘNG. ....	17
HÌNH 13. SƠ ĐỒ TUẦN TỰ XÓA HỌC BỘNG. ....	18
HÌNH 14. SƠ ĐỒ TUẦN TỰ SỬA HỌC BỘNG. ....	19
HÌNH 15. SƠ ĐỒ TUẦN TỰ DANH SÁCH SINH VIÊN.....	20
HÌNH 16. SƠ ĐỒ TUẦN TỰ TRAO HỌC BỘNG .....	21
HÌNH 17. SƠ ĐỒ TUẦN TỰ THỐNG KÊ. ....	22
HÌNH 18. SƠ ĐỒ TUẦN TỰ THỐNG KÊ THEO KHOA.....	23
HÌNH 19. SƠ ĐỒ TUẦN TỰ THỐNG KÊ THEO LỚP.....	24
HÌNH 20. SƠ ĐỒ TUẦN TỰ TRANG THỐNG KÊ THEO HỌC BỘNG .....	25
HÌNH 21. MÔ HÌNH TRIỂN KHAI HỆ THỐNG. ....	26
HÌNH 22. MÔ HÌNH TRIỂN KHAI TRÊN INTERNET. ....	26
HÌNH 23. SƠ ĐỒ LỚP. ....	27
HÌNH 24. SƠ ĐỒ DỮ LIỆU QUAN HỆ .....	27

## DANH SÁCH BẢNG BIỂU

BẢNG 1. SO SÁNH LARAVEL VÀ SYMFONY 2.....	7
BẢNG 2. SO SÁNH LARAVEL VÀ CODEIGNITER .....	7
BẢNG 3. CÁC KÝ HIỆU SƠ ĐỒ USECASE.....	8
BẢNG 4. MÔ TẢ USECASE ĐĂNG NHẬP. ....	11
BẢNG 5. MÔ TẢ USECASE QUẢN LÝ HỌC BỘNG. ....	12
BẢNG 6. MÔ TẢ USECASE TRAO HỌC BỘNG.....	13
BẢNG 7. MÔ TẢ USECASE TÌM KIẾM, THỐNG KÊ. ....	13
BẢNG 8. BẢNG DỮ LIỆU HỌC BỘNG .....	28
BẢNG 9. BẢNG DỮ LIỆU PHẠM VI HỌC BỘNG .....	28
BẢNG 10. BẢNG DỮ LIỆU LỊCH SỬ HỌC BỘNG .....	28
BẢNG 11. BẢNG DỮ LIỆU HỌC KỲ NĂM HỌC .....	29

BẢNG 12. BẢNG DỮ LIỆU KHOA.....	29
BẢNG 13. BẢNG DỮ LIỆU SINH VIÊN .....	29

## **DANH SÁCH CÁC TỪ VIẾT TẮT**

## **CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI**

### **1.1 Lý do chọn đề tài**

### **1.2 Phát biểu bài toán**

### **1.3 Mục tiêu của đề tài**

### **1.4 Hướng tiếp cận**

Website được viết bằng ngôn ngữ PHP (framework Laravel), hệ quản trị cơ sở dữ liệu Mysql, cùng với sự hỗ trợ của các ngôn ngữ khác như: CSS, JavaScript, Bootraps ...

### **1.5 Phạm vi đề tài**

Nội dung thực hiện đề tài giới hạn trong phạm vi: Thực hiện việc quản lý học bỗng tại phòng Công tác Sinh Viên.

## **CHƯƠNG 2: CƠ SỞ LÝ THUYẾT**

### **2.1 Giới Thiệu Laravel Framework**

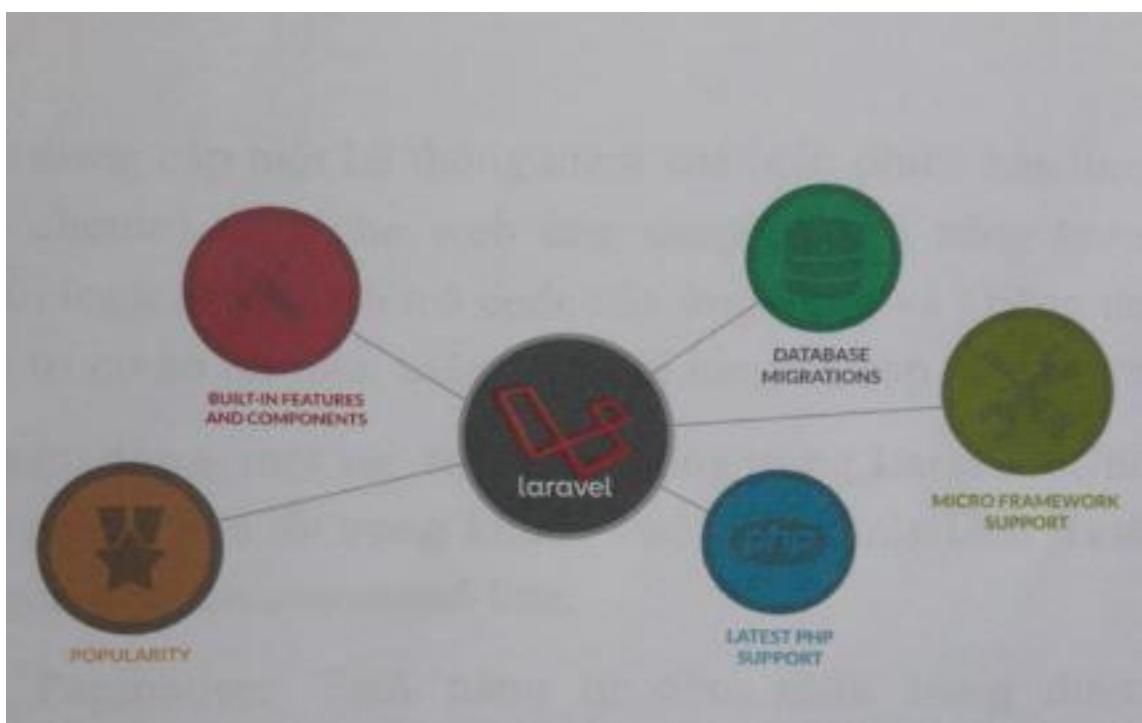
Laravel là một framework mã nguồn mở PHP miễn phí dùng để xây dựng ứng dụng website PHP được tạo bởi Taylor Otwell. Laravel được phát triển các ứng dụng website theo mô hình MVC. Laravel được ra đời vào ngày 04/2011, nó là một framework PHP khá mới mẻ nhưng nó lại có đầy đủ hướng dẫn (Document).

Có rất nhiều web giới thiệu Laravel Framework:

<http://laravel.com/docs>

<http://laravel-viet.net/>

Laravel đã phát triển vượt bậc và xuất bản nhiều phiên bản: Laravel 2, Laravel 3, Laravel 4.x, Laravel 5.x. Laravel 6.x



**Hình 1. Giới thiệu Laravel framework**

Các tính năng cơ bản của Laravel Framework gồm:

- **Bundles**: Ở laravel phiên bản 3.x, cung cấp một hệ thống đóng gói các module, với rất nhiều tính năng đi kèm.
- **Composer**: Ở laravel phiên bản 4.x, được sử dụng như một công cụ quản lý với tính năng như thêm các gói cài đặt, các chức năng PHP phụ trợ cho Laravel có trong kho Packagist.
- **Eloquent ORM (object relation mapping)**: ánh xạ các đối tượng và quan hệ cơ sở dữ liệu, cung cấp các phương thức nội bộ để thực thi đồng thời cũng bổ sung các tính năng hạn chế về mối quan hệ giữa các đối tượng cơ sở dữ liệu. Eloquent ORM trình bày các bảng trong cơ sở dữ liệu dưới dạng các lớp, cung

cấp thêm lựa chọn truy cập cơ sở dữ liệu trực tiếp mới mẻ hơn, chuyên nghiệp hơn.

- **Application logic:** Là một phần của phát triển ứng dụng, được sử dụng bởi bộ điều khiển controllers.
- **Routes:** Định nghĩa mối quan hệ giữa các đường dẫn (url), các liên kết (link) Khi một liên kết được tạo ra bằng cách sử dụng tên của routes, thì một định danh liên kết thống nhất sẽ được tạo ra bởi laravel.
- **Restful Controller:** cung cấp các tùy chọn để tách các logic phía sau các request HTTP POST, GET
- **Class auto loading:** cung cấp việc tải tự động các class trong PHP, mà không cần include các class vào. Tùy thuộc vào yêu cầu các class cần thiết sẽ được nạp vào, hạn chế các class không cần thiết.
- **View:** chứa các mã html, hiển thị dữ liệu được chỉ định bởi controller
- **Migrations:** cung cấp một hệ thống kiểm soát các phiên bản lược đồ cơ sở dữ liệu (database schema), làm cho web ứng dụng có khả năng tương tác phù hợp những thay đổi logic, các đoạn mã code của ứng dụng và những thay đổi cần thiết trong việc bố trí cơ sở dữ liệu, triển khai nói lỏng và cập nhật các ứng dụng.
- **Unit Testing:** đóng một vai trò quan trọng trong Laravel, Unit testing chứa rất nhiều các hệ thống unit testing, giúp phát hiện và ngăn chặn lỗi trong khuôn khổ nhất định. Unit Testing có thể được chạy thông qua tiện ích command-line.
- **Automatic pagination :** Tính năng tự động phân trang được tích hợp vào Laravel giúp đơn giản hóa các nhiệm vụ thực hiện phân trang so với các phương pháp thông thường.

Từ tháng 3 năm 2015, Laravel được xem là một trong những framework PHP phổ biến nhất, cùng với Symfony2, Nette, CodeIgniter, Yii.

## 2.2 Lịch sử phát triển Laravel Framework

Laravel được Taylor Otwell tạo ra như một giải pháp thay thế cho CodeIgniter, cung cấp nhiều tính năng quan trọng hơn như xác thực và phân quyền. Tôi cũng không chắc về điều này, nhưng có thể Taylor vốn là một .NET developer khi bắt đầu có nhu cầu làm việc với PHP khoảng vào những năm 2010-2011, đã chọn CodeIgniter khi đó đang là một ngôi sao mới nổi, thậm chí lấn át cả Symfony gạo cội. Và Taylor nhanh chóng nhận ra những điểm khiếm khuyết ở CodeIgniter, với tài năng và kiến thức xuất sắc về design-pattern của mình, Taylor quyết định tự mình tạo ra một framework sao cho thật đơn giản, dễ hiểu, hỗ trợ lập trình viên hiện thực ý tưởng một cách nhanh nhất bằng nhiều tính năng hỗ trợ như Eloquent ORM mạnh mẽ, xác thực đơn giản, phân trang hiệu quả, và hơn thế nữa.

Bản Laravel beta đầu tiên được phát hành vào ngày 9/6/2011, tiếp đó là **Laravel 1** phát hành trong cùng tháng. Laravel 1 bao gồm các tính năng như xác thực, bản địa hóa, model, view, session, định tuyến và các cơ cấu khác, nhưng vẫn còn thiếu controller, điều này làm nó chưa thật sự là một MVC framework đúng nghĩa.

**Laravel 2** được phát hành vào tháng 9 năm 2011, mang đến nhiều cài tiến từ tác giả và cộng đồng. Tính năng đáng kể bao gồm hỗ trợ controller, điều này thực sự biến Laravel 2 thành một MVC framework hoàn chỉnh, hỗ trợ Inversion of Control ([IoC](#)), hệ thống template Blade. Bên cạnh đó, có một nhược điểm là hỗ trợ cho các gói của nhà phát triển bên thứ 3 bị gỡ bỏ.

**Laravel 3** được phát hành vào tháng 2 năm 2012, với một số tính năng mới bao gồm giao diện dòng lệnh (CLI) tên “Artisan”, hỗ trợ nhiều hơn cho hệ thống quản trị cơ sở dữ liệu, chức năng ánh xạ cơ sở dữ liệu Migration, hỗ trợ “bắt sự kiện” trong ứng dụng, và hệ thống quản lý gói gọi là “Bundles”. Lượng người dùng và sự phổ biến tăng trưởng mạnh kể từ phiên bản Laravel 3.

**Laravel 4**, tên mã “Illuminate”, được phát hành vào tháng 5 năm 2013. Lần này thực sự là sự lột xác của Laravel framework, di chuyển và tái cấu trúc các gói hỗ trợ vào một tập được phân phối thông qua Composer, một chương trình quản lý gói thư viện phụ thuộc độc lập của PHP. Bố trí mới như vậy giúp khả năng mở rộng của Laravel 4 tốt hơn nhiều so với các phiên bản trước. Ra mắt lịch phát hành chính thức mỗi sáu tháng một phiên bản nâng cấp nhỏ. Các tính năng khác trong Laravel 4 bao gồm tạo và thêm dữ liệu mẫu (database seeding), hỗ trợ hàng đợi, các kiểu gửi mail, và hỗ trợ “xóa mềm” (soft-delete: record bị lọc khỏi các truy vấn từ Eloquent mà không thực sự xóa hẳn khỏi DB).

**Laravel 5** được phát hành trong tháng 2 năm 2015, như một kết quả thay đổi đáng kể cho việc kết thúc vòng đời nâng cấp Laravel lên 4.3. Bên cạnh một loạt tính năng mới và các cải tiến như hiện tại, Laravel 5 cũng giới thiệu cấu trúc cây thư mục nội bộ cho phát triển ứng dụng mới. Những tính năng mới của Laravel 5 bao gồm hỗ trợ lập lịch định kỳ thực hiện nhiệm vụ thông qua một gói tên là “Scheduler”, một lớp trừu tượng gọi là “Flysystem” cho phép điều khiển việc lưu trữ từ xa đơn giản như lưu trữ trên máy local – dễ thấy nhất là mặc định hỗ trợ dịch vụ Amazon S3, cải tiến quản lý assets thông qua “Elixir”, cũng như đơn giản hóa quản lý xác thực với các dịch vụ bên ngoài bằng gói “Socialite”.

**Laravel 5.1** phát hành vào tháng 6 năm 2015, là bản phát hành đầu tiên nhận được hỗ trợ dài hạn (LTS) với một kế hoạch fix bug lên tới 2 năm vào hỗ trợ và lỗi bảo mật lên tới 3 năm. Các bản phát hành LTS của Laravel được lên kế hoạch theo mỗi 2 năm.

### 2.3 Giới thiệu chung về Laravel.

Laravel 5.0 giới thiệu một ứng dụng có cấu trúc mới cho các Project Laravel phiên bản trước đây. Cấu trúc mới này phục vụ như một nền tảng tốt để xây dựng các ứng dụng mạnh mẽ trong Laravel, bao trùm các tiêu chuẩn tự động tải mới (PSR-4) thông qua ứng dụng.

#### 2.3.1 Cấu trúc thư mục mới:

Các thư mục app/model cũ được xóa hoàn toàn. Thay vào đó, tất cả code được lưu trực tiếp trong thư mục ứng dụng đã tạo. Mặc định là App. Không gian tên mặc định này có thể được thay đổi một cách nhanh chóng bằng cách sử dụng mới ứng dụng: tên lệnh artisan.

Controllers, middleware, và request (Một loại mới của class trong Laravel 5.0) được phân nhóm theo các thư mục App/Http. Tất cả middleware được chia thành các tập tin class.

App/Providers được thay thế App/start files từ các phiên bản Laravel 4.x. Từ đó có thể thêm các dịch vụ của bên thứ ba qua các file tải về dễ dàng hơn. Chẳng hạn như: bootstrap, error handing, logging, route loading.

Các files hỗ trợ ngôn ngữ và views được chuyển đến thư mục resources.

### 2.3.2 Contracts

Tất cả các thành phần giao diện được thực hiện tại illuminate/contracts.

### 2.3.3 Route cache

Nếu ứng dụng sử dụng Controllers Routes, có thể dùng lệnh artisan để tăng tốc độ của các tuyến đường, giúp xây dựng các ứng dụng với hơn 100 route.

### 2.3.4 Route Middleware

Laravel 5 hỗ trợ HTTP middleware, và authentication, CSRF “filters” được chuyển sang middleware. Middleware cung cấp một giao diện thống nhất duy nhất để thay thế tất cả các loại của filters, cho phép dễ dàng kiểm tra, từ chối, yêu cầu trước khi chúng ta nhập vào ứng dụng.

### 2.3.5 Controller Method Injection

Cá phương thức controllers hỗ trợ xây dựng hàm riêng nhanh chóng, không cần nhớ cấu trúc của hàm. Gõ lệnh controller hỗ trợ, hàm sẽ được tạo sẵn.

### 2.3.6 Authentication Scaffolding

Đăng ký thành viên, xác thực và thiết lập lại mật khẩu được đặt tại /views/auth. Tài khoản người dùng được đưa vào table Migration. Cho phép phát triển nhanh chóng các ý tưởng của ứng dụng, không làm chậm tốc độ nhờ authentication soạn sẵn. Các phương thức xác thực được đặt tại auth/login và auth/register routes. Các dịch vụ App\Services\Auth\Registrar xác nhận người sử dụng và người xây dựng.

### 2.3.7 From Requests

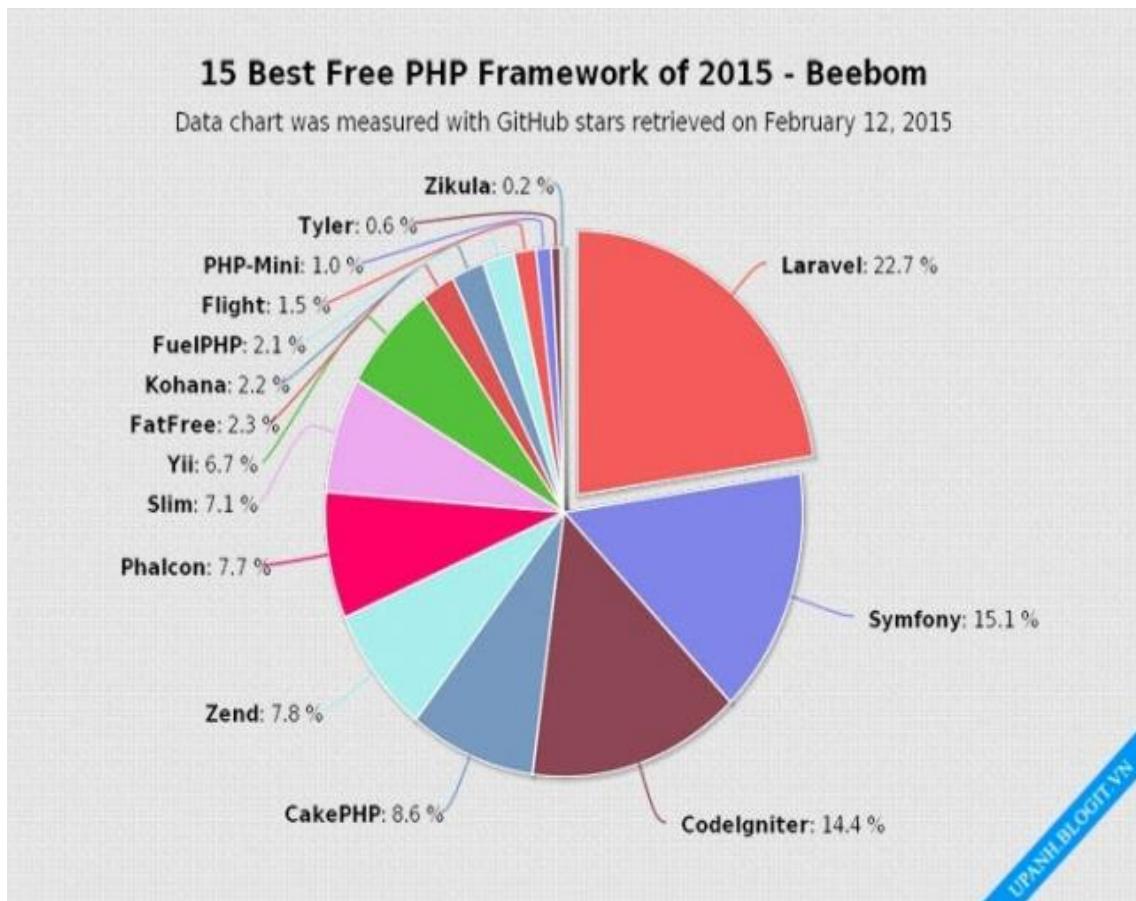
Laravel 5.0 hỗ trợ Form Requests, mở rộng ở Illuminate\ Foundation\ Http\ FormRequest class. Cho phép thiết lập các yêu cầu khi người dùng sử dụng ứng dụng. Ràng buộc kiểm tra thông tin người dùng cung cấp. Ví dụ: form đăng nhập thiết lập các ràng buộc: nhập đúng định dạng email, mật khẩu...

### 2.3.8 Simple Controller Request Validation

Laravel Framework hỗ trợ thiết lập câu lệnh báo lỗi riêng cho những ràng buộc.

## 2.4 So sánh Laravel Framework và các Framework khác

### 2.4.1 Sự phổ biến của Laravel Framework



**Hình 2. Thống kê mức độ sử dụng của người dùng đối với các framework năm 2015.**

Theo thống kê mức độ sử dụng của người dùng đối với Framework năm 2015 của các chuyên gia, Laravel Framework đứng đầu danh sách các Framework với tỷ lệ 22,7%. Symfony chiếm 15,1% và CodeIgniter chiếm 14,4%. Laravel Framework thu hút đông đảo người dùng bởi cú pháp câu lệnh đơn giản, dễ sử dụng và hỗ trợ nhiều dịch vụ.

#### 2.4.2 So sánh Laravel và Symfony 2

Laravel được xây dựng trên nền tảng của Symfony 2. Laravel vs Symfony được thiết kế hỗ trợ: Active-Record, Model-View-Controller, Dependency injection, RESTfull....

	<i>Laravel</i>	<i>Symfony 2</i>
--	----------------	------------------

<b>Database</b>	<ul style="list-style-type: none"> <li>• SQLite, MySQL, PostgreSQL, Redis, Microsoft BI, MongoDB.</li> </ul>	<ul style="list-style-type: none"> <li>• Microsoft BI, MongoDB, MySQL NoSQL, PostgreSQL, CouchDB, DynamoDB, GemFire, GraphDB, Membase, MemcacheDB, Oracle, ApacheJcrabbit.</li> </ul>
<b>Template language</b>	<ul style="list-style-type: none"> <li>• Blade, Php, Smarty, Twig.</li> </ul>	<ul style="list-style-type: none"> <li>• Twig, php.</li> </ul>

**Bảng 1. So sánh laravel và Symfony 2**

#### 2.4.3 So sánh Laravel và CodeIgniter

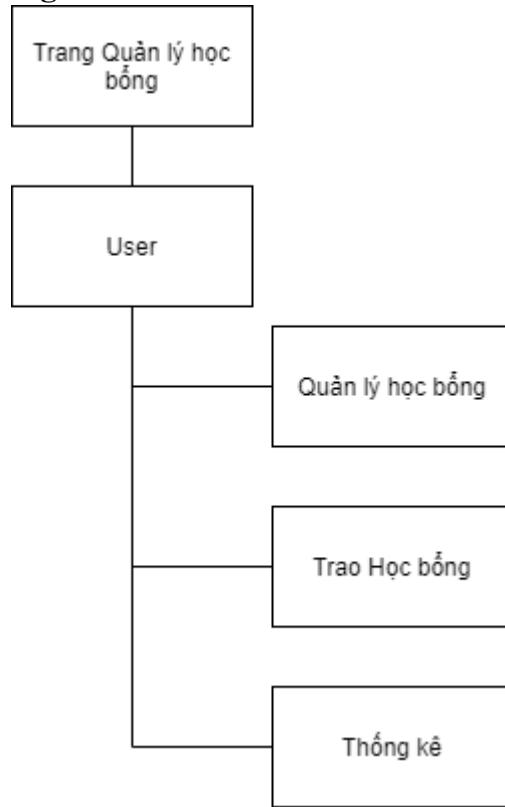
	<i>Laravel</i>	<i>CodeIgniter</i>
<b>View</b>	<ul style="list-style-type: none"> <li>• Hỗ trợ sử dụng Blade Template Engine.</li> </ul>	<ul style="list-style-type: none"> <li>• Không hỗ trợ Template Engine. Có thể tích hợp thêm.</li> <li>• Tốn thời gian</li> </ul>
<b>Database Model</b>	<ul style="list-style-type: none"> <li>• Câu lệnh code đơn giản ngắn gọn.</li> <li>• Hỗ trợ tương tác với Database qua nhiều phương thức.</li> </ul>	<ul style="list-style-type: none"> <li>• Hỗ trợ Active Record.</li> </ul>
<b>Route</b>	<ul style="list-style-type: none"> <li>• Hỗ trợ từ định nghĩa Route, Group Route...</li> </ul>	<ul style="list-style-type: none"> <li>• Quy định từng query riêng lẻ. Người dùng cần có kiến thức về Regex để có thể rewrite url tối ưu.</li> </ul>
<b>Các dịch vụ</b>	<ul style="list-style-type: none"> <li>• Cung cấp nhiều dịch vụ dễ sử dụng. (Unit, Testing, Authentication, Caching, Hooks/ Filter, Helper, Pagination)</li> </ul>	<ul style="list-style-type: none"> <li>• Hỗ trợ ít dịch vụ hơn Laravel.</li> </ul>

**Bảng 2. So sánh laravel và CodeIgniter**

## CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ

### 3.1 Phân tích thiết kế hệ thống thông tin.

#### 3.1.1 Sơ đồ chức năng



Hình 3. Sơ đồ chức năng

#### 3.1.2 Sơ đồ Usecase

Bảng 3. Các ký hiệu sơ đồ Usecase.

STT	Ký Hiệu	Chú Giải
1		Tác nhân (Actor)
2		Usecase
3		Quan hệ

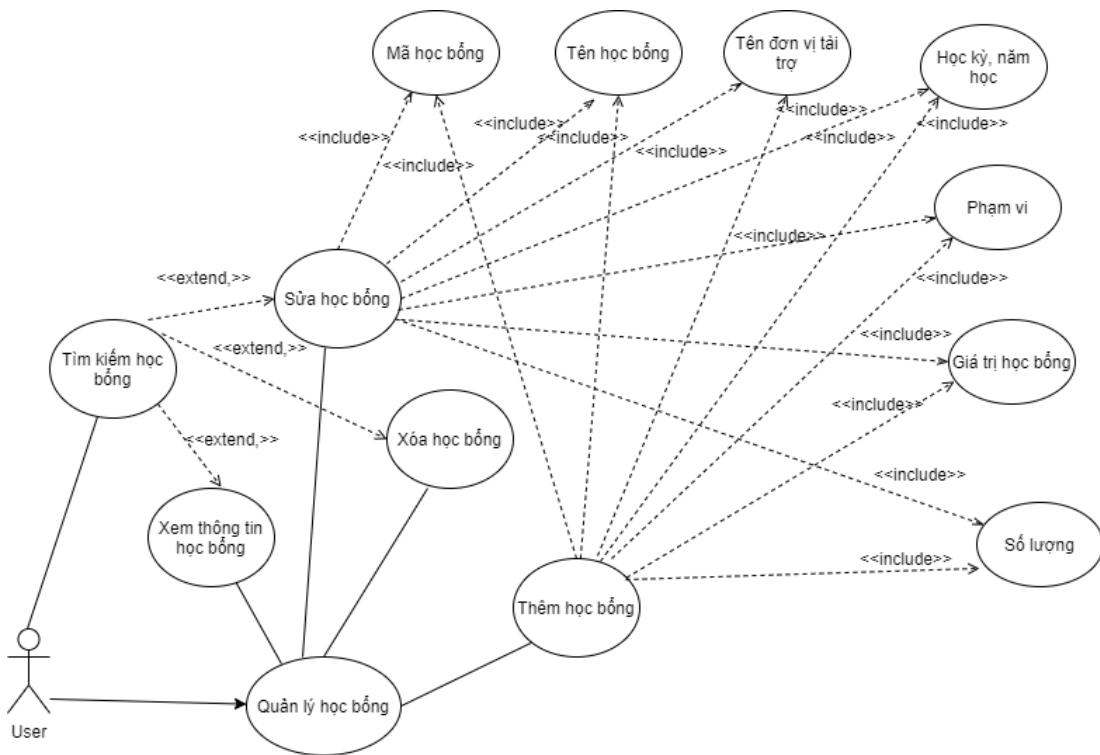
Từ những chức năng đã có ta có sơ đồ usecase tổng quát.



**Hình 4. Sơ đồ Usecase Tổng quát**

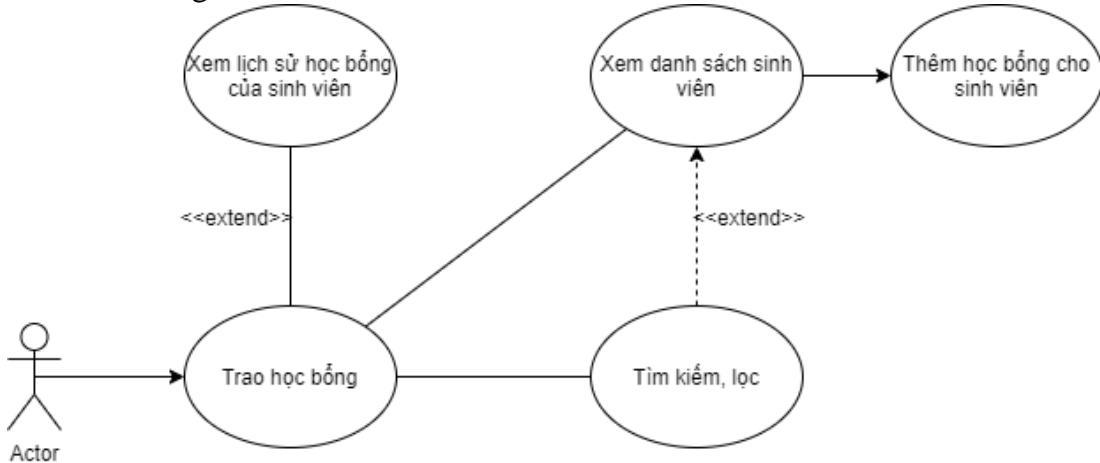
### 3.1.2.1 Actor

- ❖ Quản lý Học bổng



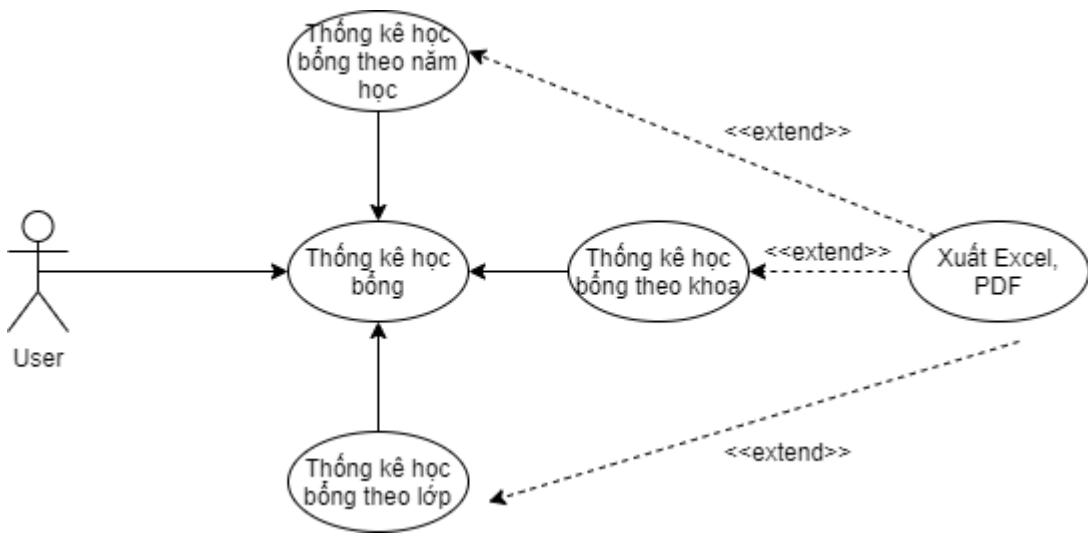
**Hình 5. Sơ đồ phân rã Usecase Quản lý học bổng.**

❖ Trao học bổng



**Hình 6. Sơ đồ phân rã Usecase Trao học bổng.**

❖ Thông kê



**Hình 7. Sơ đồ phân rã Usecase Thống kê**

### 3.1.3 Đặt tả Usecase

**Bảng 4. Mô tả Usecase Đăng nhập.**

<b>Tên usecase:</b> Đăng nhập	<b>ID:</b> UC1
<b>Tác nhân chính:</b> Người dùng	<b>Kiểu:</b> Chi tiết
<b>Những người tham gia:</b> Tất cả người dùng đã có tài khoản của hệ thống đều thực hiện được chức năng này.	
<b>Mô tả ngắn gọn:</b> Usecase này mô tả cách thức người dùng đăng nhập vào hệ thống.	
<b>Điều kiện tiên quyết:</b> Hệ thống phải kết nối thành công với server.	
<b>Sự kiện kích hoạt:</b> Người dùng truy cập vào website và chọn đăng nhập.	
<b>Mối quan hệ:</b>	

**Luồng sự kiện chính:**

- Hệ thống hiện thị một màn hình yêu cầu người sử dụng nhập email và mật khẩu.
- Người dùng nhập tên và mật khẩu
- Sau khi nhập xong nhấn Đồng ý
- Hệ thống sẽ chuyển thông tin lên server để xử lý
- Hệ thống sẽ kiểm tra tên và mật khẩu có trong cơ sở dữ liệu hay không
- Hệ thống sẽ dựa vào quyền của người đăng nhập để cấp cho những chức năng tương ứng .
- Sau đó hệ thống sẽ thông báo cho người dùng biết là đã đăng nhập thành công

**Luồng sự kiện phụ:**

- Nếu trong luồng sự kiện chính, người dùng nhập sai tên hoặc mật khẩu thì hệ thống sẽ thông báo là nhập tên và mật khẩu sai.
- Hệ thống sẽ hiển thị lại màn hình đăng nhập yêu cầu người dùng, đăng nhập lại.
- Người dùng có thể đăng nhập lại hoặc là hủy bỏ việc đăng nhập

**Kết quả:** Nếu đăng nhập thành công sẽ xuất hiện màn hình quản lý chính và tùy theo từng quyền của người dùng sẽ xuất hiện những chức năng tương ứng.

**Bảng 5. Mô tả Usecase Quản lý học bỗng.**

Tên usecase: Quản lý học bỗng	ID: UC2
Tác nhân chính: Admin	Kiểu: Chi tiết

**Mô tả ngắn gọn:**

Use-case này thể hiện chức năng quản lý học bỗng của người quản trị.

**Tạo mới học bỗng:**

- Người dùng nhấn button Thêm học bỗng. Hệ thống sẽ đưa ra màn hình tạo mới tin tức. Hệ thống yêu cầu người dùng nhập thông tin cho học bỗng. Người dùng nhấn button Lưu. Hệ thống sẽ thực hiện việc tạo mới học bỗng.
- Người có thể dùng import học bỗng mới bằng file Excel.

**Sửa học bỗng :**

- Người dùng nhấn button Sửa học bỗng ở danh sách học bỗng. Hệ thống đưa ra màn hình sửa học bỗng, yêu cầu người dùng nhập thông tin của học bỗng và nhấn button Lưu. Hệ thống sẽ thực hiện việc sửa học bỗng.

**Xóa học bỗng :**

- Người dùng nhấn button Xóa học bỗng ở danh sách học bỗng. Hệ thống đưa ra màn hình xác nhận việc xóa học bỗng. Hệ thống sẽ thực hiện việc xóa học bỗng.

**Điều kiện tiên quyết:** Người dùng phải đăng nhập thành công vào hệ thống và có quyền thao tác trên học bỗng.

**Kết quả:** Sau khi tạo mới hoặc sửa học bổng thành công, hệ thống sẽ đưa thông tin mới lưu vào cơ sở dữ liệu. Sau khi xóa học bổng thành công, hệ thống sẽ xóa thông tin của học bổng đó trong cơ sở dữ liệu.

**Bảng 6. Mô tả Usecase Trao học bổng.**

<b>Tên usecase:</b> Trao học bổng	<b>ID:</b> UC3
<b>Tác nhân chính:</b> Admin	<b>Kiểu:</b> Chi tiết
<b>Mô tả ngắn gọn:</b> Usecase này thể hiện chức năng trao học bổng cho sinh viên	
<b>Tìm kiếm, lọc sinh viên:</b>	
<ul style="list-style-type: none"> <li>Người dùng nhập điều kiện tìm kiếm, lọc trên khung Điểm học tập và Điểm rèn luyện. Hệ thống đưa ra màn hình danh sách sinh viên ứng với điều kiện đã lọc.</li> </ul>	
<b>Xem lịch sử học bổng của sinh viên:</b>	
<ul style="list-style-type: none"> <li>Người dùng nhấp vào button Xem lịch sử để xem lịch sử nhận học bổng của từng sinh viên. Trong trang Lịch sử học bổng, người dùng có thể chỉnh sửa thông tin nhận học bổng của sinh viên.</li> </ul>	
<b>Trao học bổng:</b>	
<ul style="list-style-type: none"> <li>Sau khi tìm kiếm, lọc sinh viên phù hợp. Người dùng nhấp vào nút Trao học bổng. Hệ thống đưa ra màn hình danh sách học bổng và giá trị cần trao.</li> <li>Khi bấm Lưu, hệ thống sẽ kiểm tra sinh viên đã nhận học bổng đó trong học kỳ này hay chưa và số tiền còn lại của học bổng có đủ để trao hay không.</li> <li>Hệ thống sẽ thực hiện việc lưu thông tin trao học bổng sau khi thỏa điều kiện.</li> </ul>	
<b>Điều kiện tiên quyết:</b> Người dùng phải đăng nhập thành công vào hệ thống và có quyền thao tác trên học bổng.	
<b>Kết quả:</b> Sau khi thực hiện thành công việc trao học bổng hoặc sửa thông tin trao học bổng. Hệ thống sẽ lưu thông tin việc trao học bổng vào cơ sở dữ liệu và đưa ra thông báo cho người dùng biết đã tạo mới hoặc sửa thành công.	

**Bảng 7. Mô tả Usecase Tìm kiếm, Thống kê.**

<b>Tên usecase:</b> Tìm kiếm, Thống kê	<b>ID:</b> UC4
<b>Tác nhân chính:</b> Admin	<b>Kiểu:</b> Chi tiết
<b>Mô tả ngắn gọn:</b> Usecase này thể hiện chức năng tìm kiếm, thống kê học bổng	
<b>Tìm kiếm:</b>	
<ul style="list-style-type: none"> <li>Người dùng nhập điều kiện tìm kiếm trên thanh tìm kiếm. Hệ thống đưa ra màn hình danh sách học bổng ứng với điều kiện đã lọc.</li> </ul>	

#### Xem thống kê:

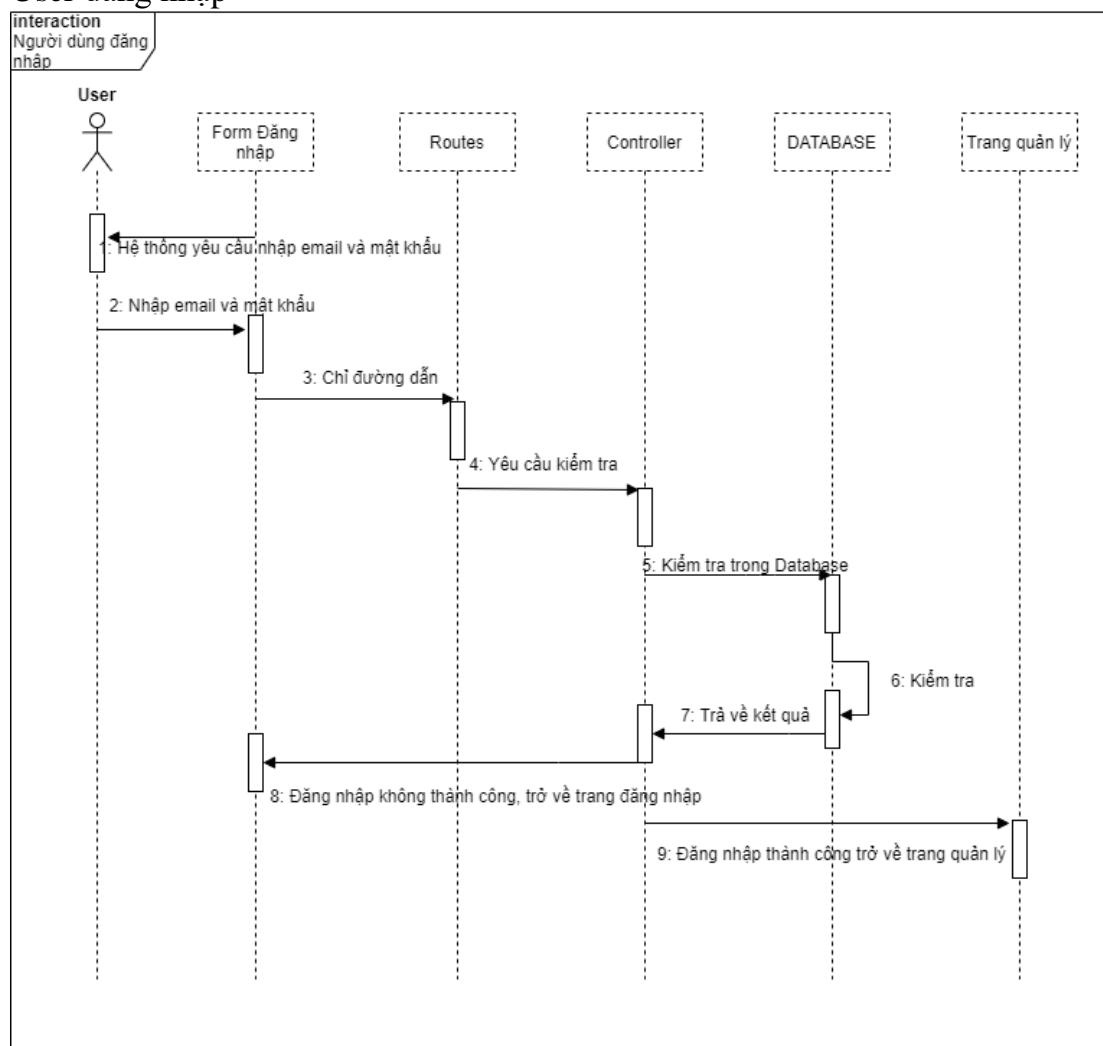
- Hệ thống sẽ thống kê các số liệu học bổng theo năm học. Người dùng xem thống kê theo phạm vi toàn trường, khoa và lớp. Người dùng có thể xuất số liệu thống kê theo dạng Excel hoặc PDF.

**Điều kiện tiên quyết:** Người dùng phải đăng nhập thành công vào hệ thống và có quyền thao tác trên học bổng.

**Kết quả:** Hệ thống hiển thị số liệu thống kê theo năm học. Xuất thống kê Excel hoặc PDF.

#### 3.1.4 Sơ đồ tuần tự

##### ❖ User đăng nhập

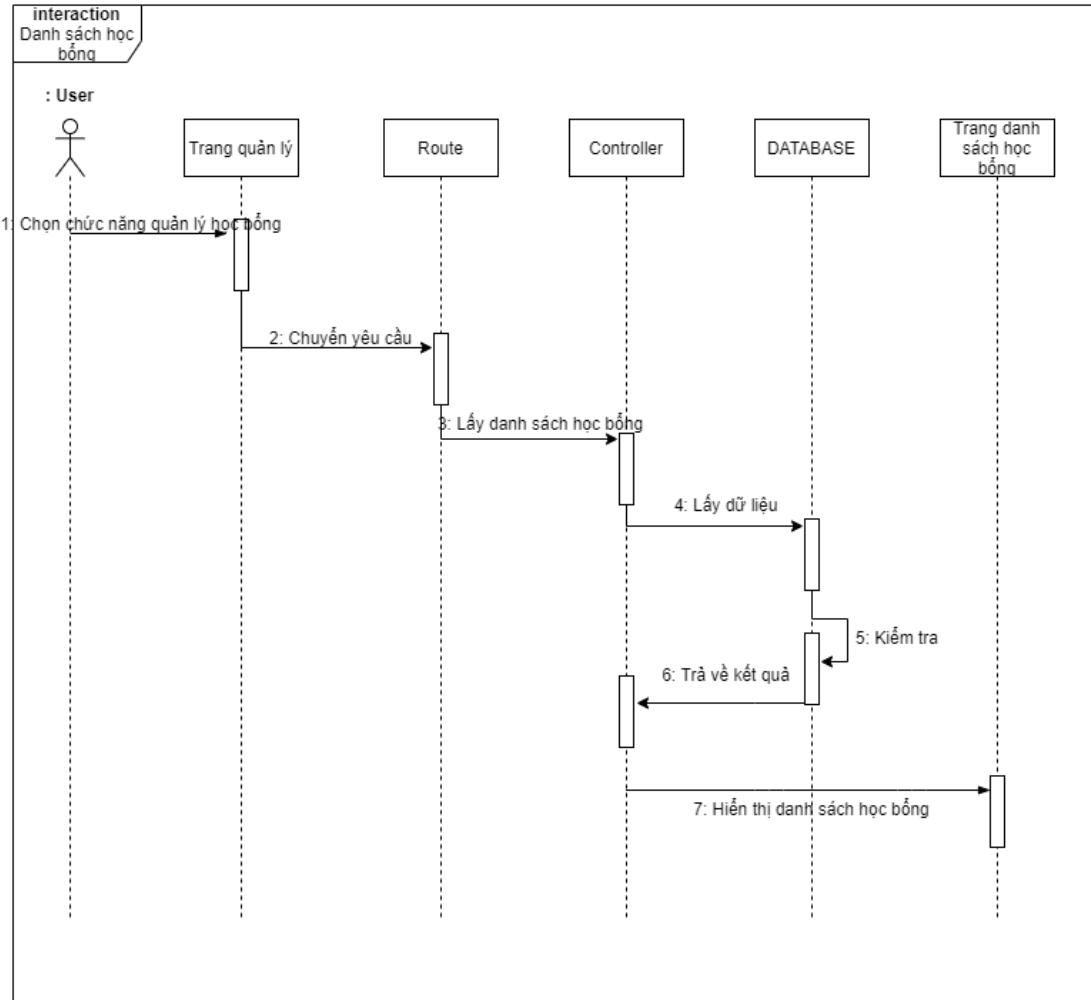


**Hình 8. Sơ đồ tuần tự Admin đăng nhập.**

##### ❖ Trang Danh Sách học bổng

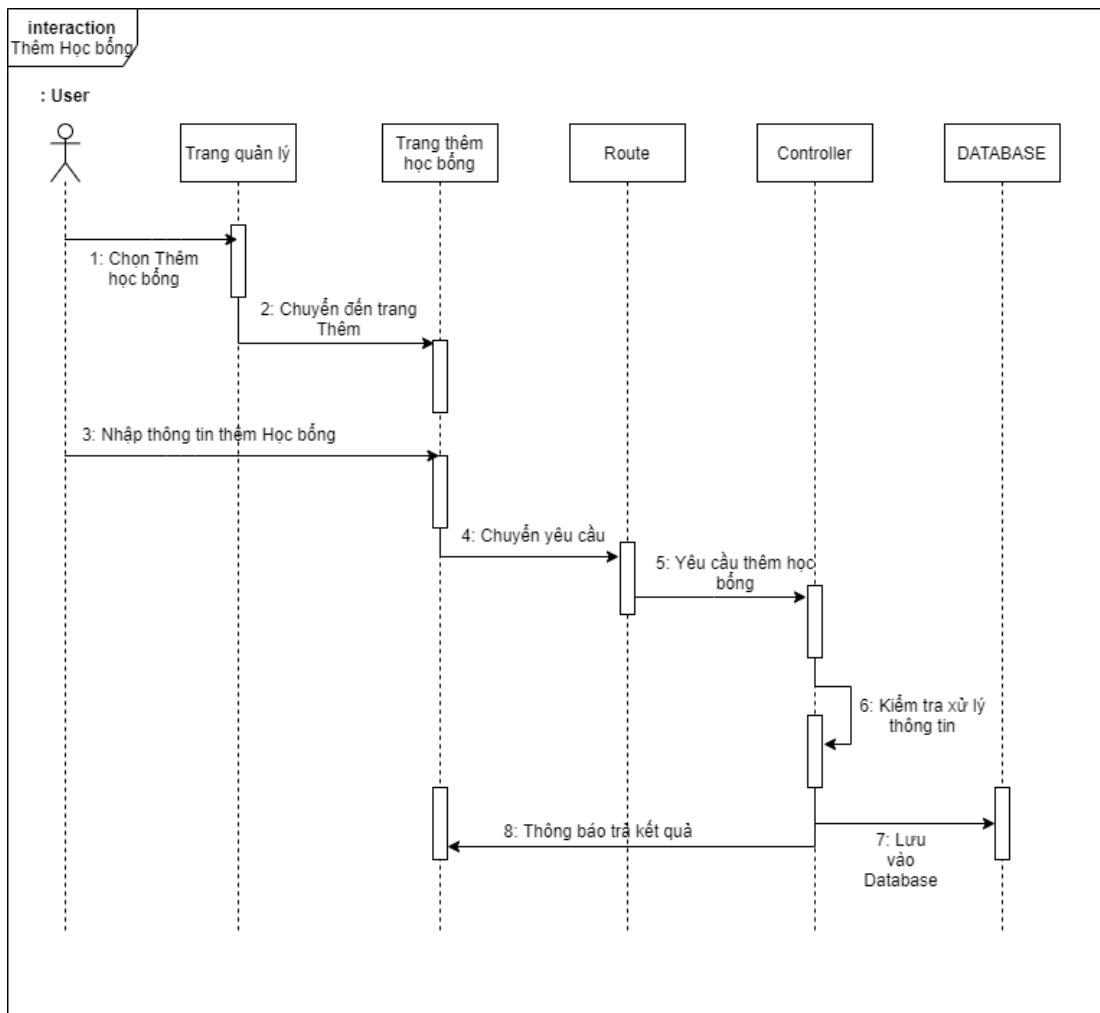
DANH SÁCH LIST										
STT	Mã học bổng	Tên học bổng	Đơn vị tài trợ	Phạm vi	Học kỳ, năm học	Giá trị học bổng	Số lượng	Đã trao	Trạng thái	Thao tác
1	HB001	Xổ số kiến thiết	Xổ số kiến thiết	Công nghệ thông tin Kinh tế - Quản trị kinh doanh	Học kỳ 1 - Năm học 2019 - 2020	300.000.000	30	4	Hiện hành	
2	HB002	Tiếp bước đến trường	Tiếp bước đến trường	Công nghệ thông tin	Học kỳ 1 - Năm học 2019 - 2020	500.000.000	20	2	Hiện hành	
3	HB003	Học Bổng Vietcombank	Vietcombank	Ngoại ngữ Sư phạm	Học kỳ 2 - Năm học 2018 - 2019	250.000.000	25	1	Đã hoàn thành	
4	HB004	Học bổng Khuyến tài Đoàn Tối	Quỹ Đoàn Tối	Toàn trường	Học kỳ 1 - Năm học 2019 - 2020	250.000.000	25	0	Hiện hành	

**Hình 9. Danh sách học bổng.**



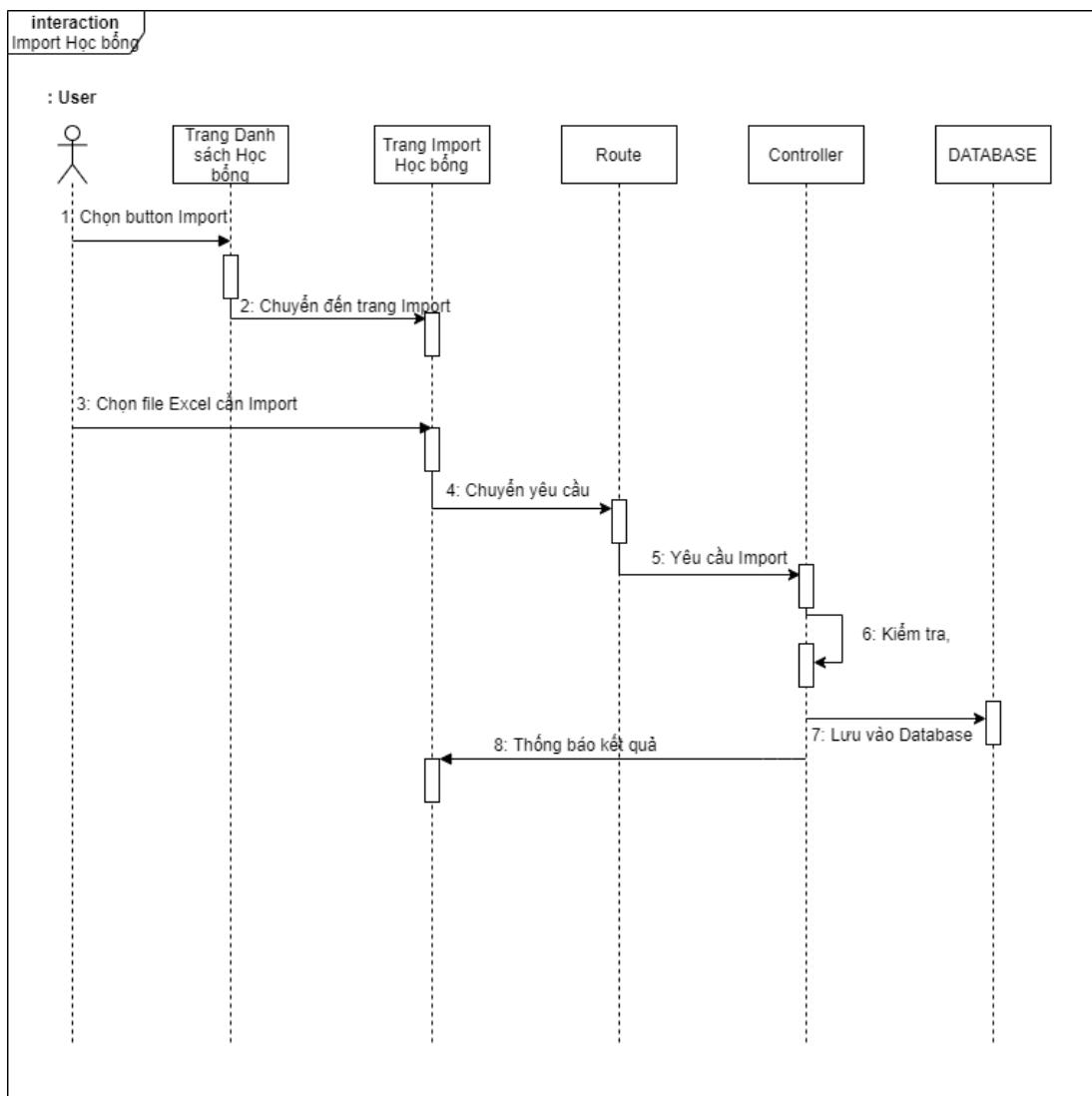
**Hình 10. Sơ đồ tuần tự Danh sách học bổng.**

❖ Thêm học bổng



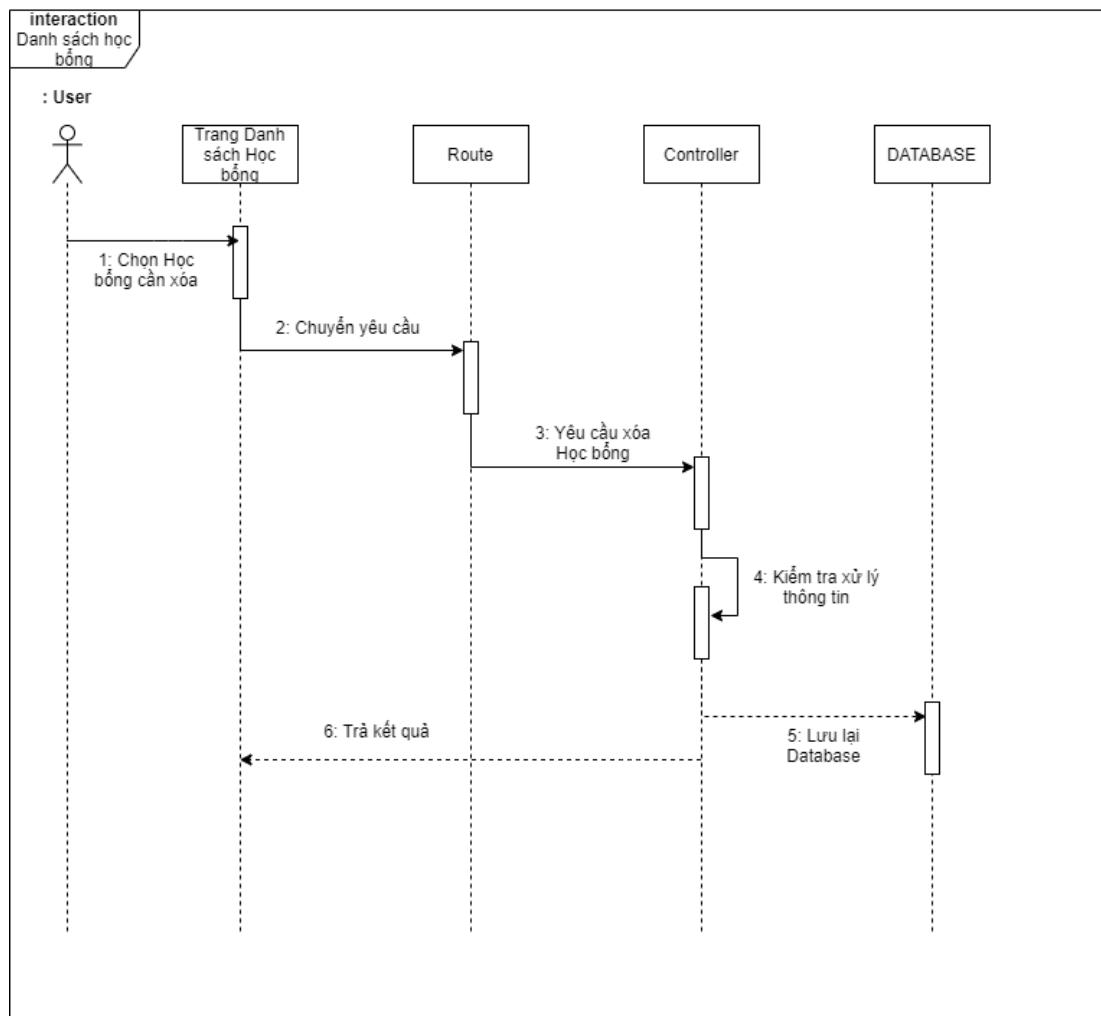
**Hình 11. Sơ đồ tuần tự Thêm học bổng.**

❖ Import Học bổng



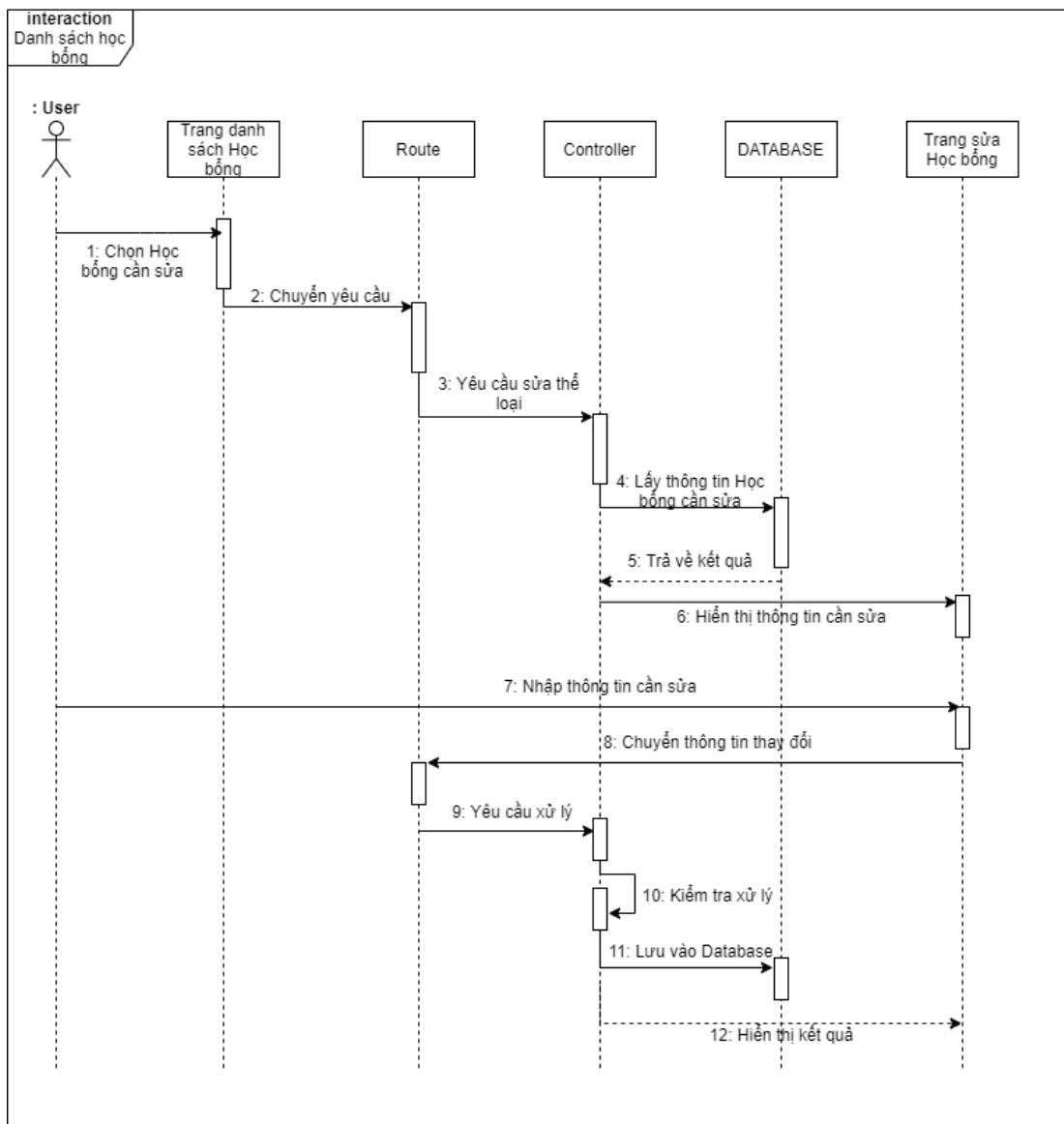
**Hình 12. Sơ đồ tuần tự Import học bổng.**

Xóa Học bổng



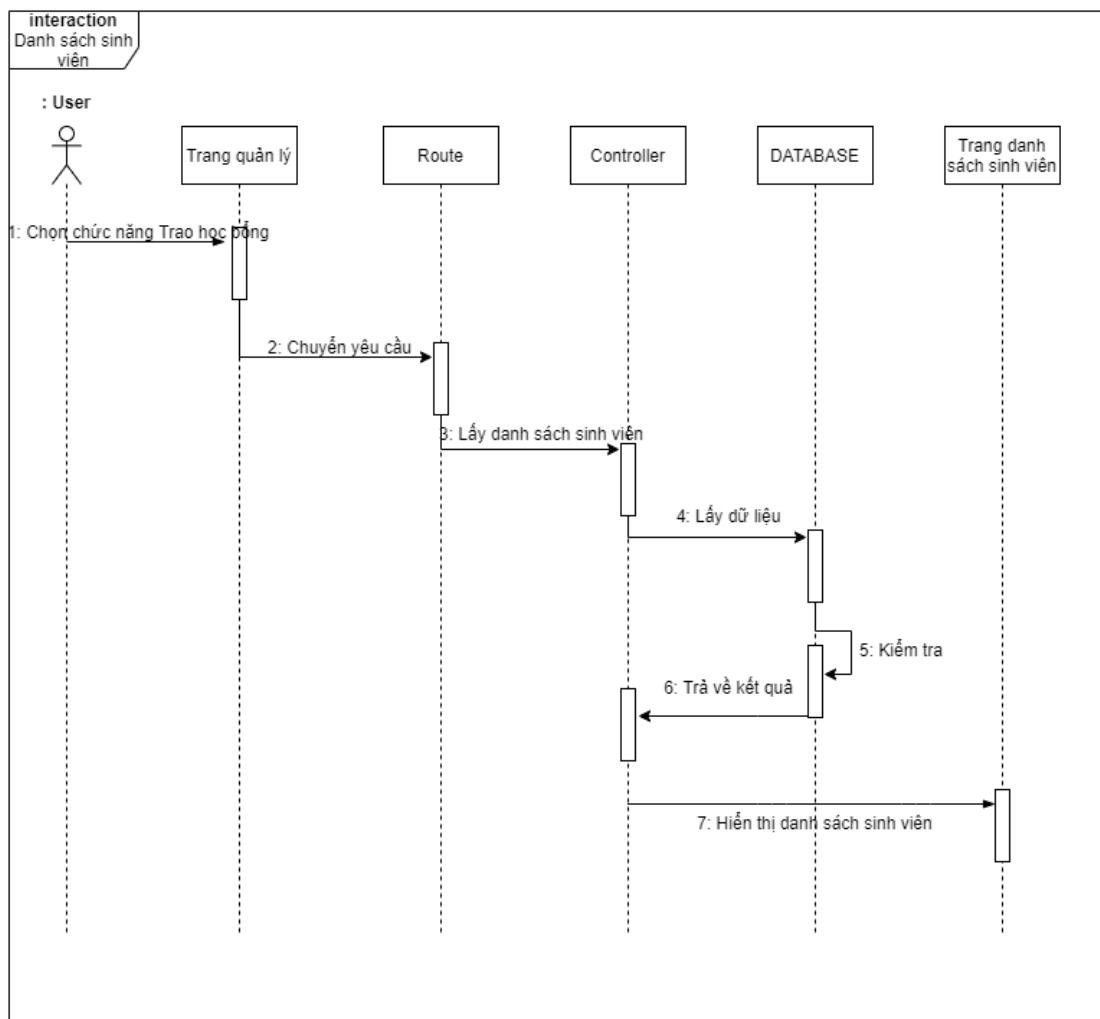
**Hình 13. Sơ đồ tuần tự Xóa Học bổng.**

❖ Sửa Học bổng



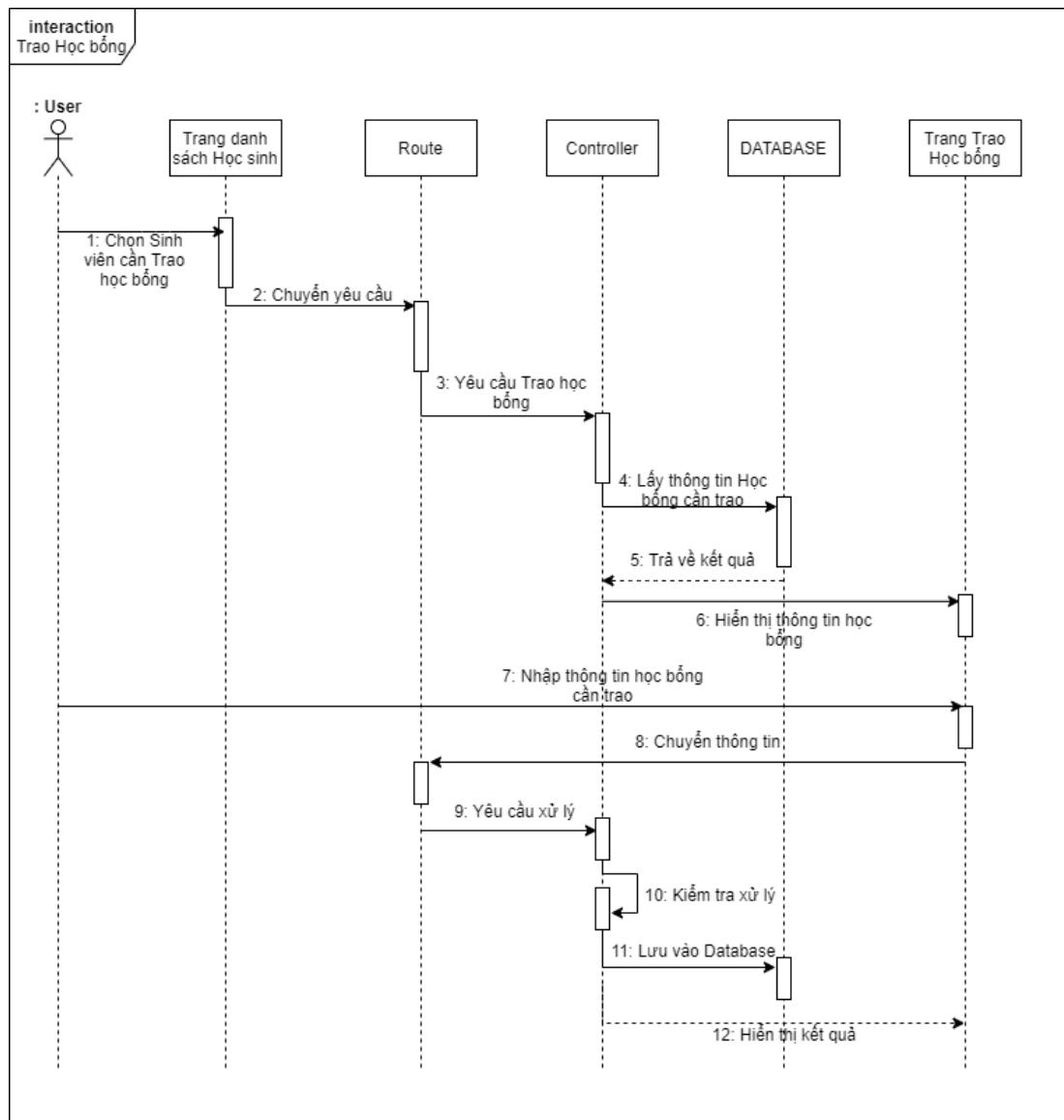
**Hình 14. Sơ đồ tuần tự Sửa Học bổng.**

- ❖ Trang Trao học bổng.



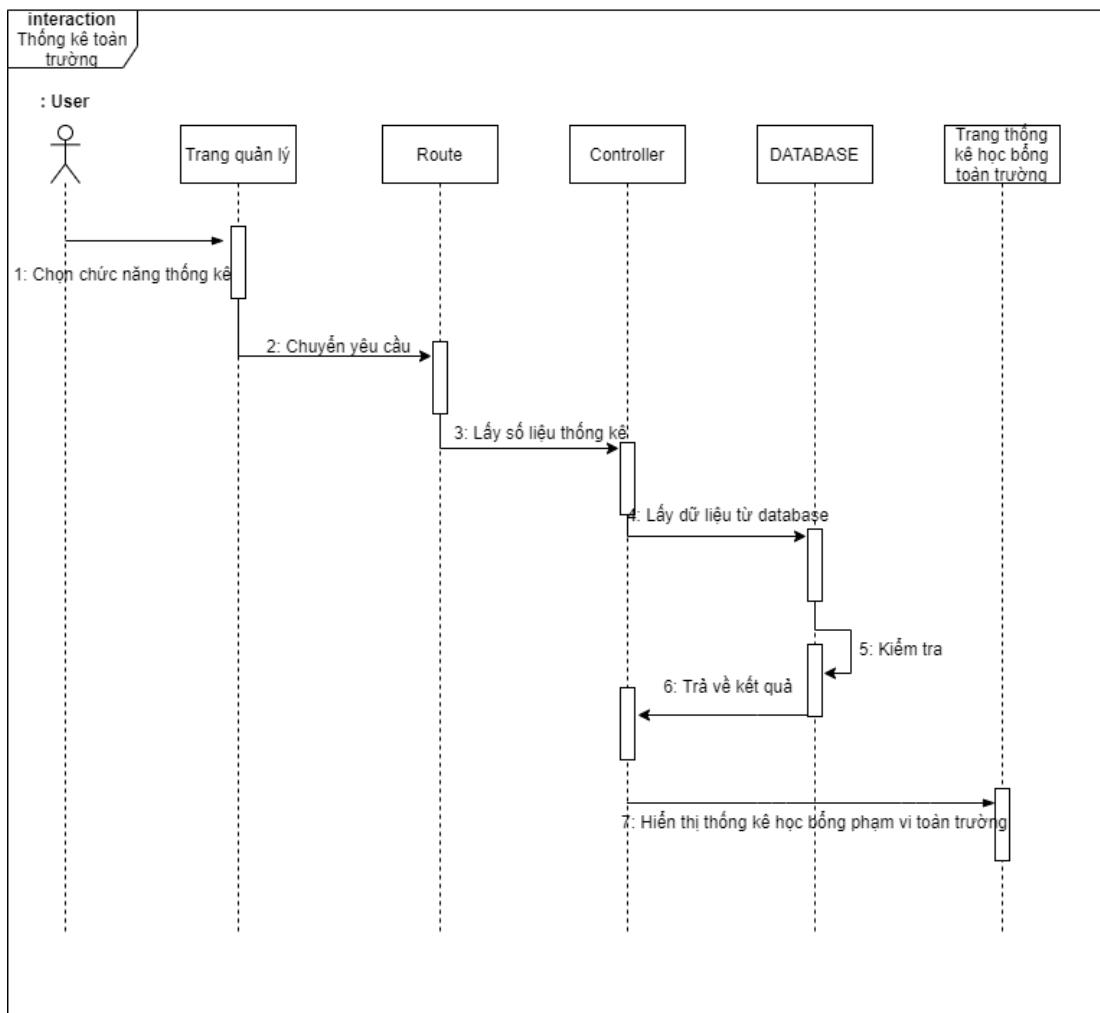
**Hình 15. Sơ đồ tuần tự Danh sách Sinh viên.**

❖ Trao học bổng



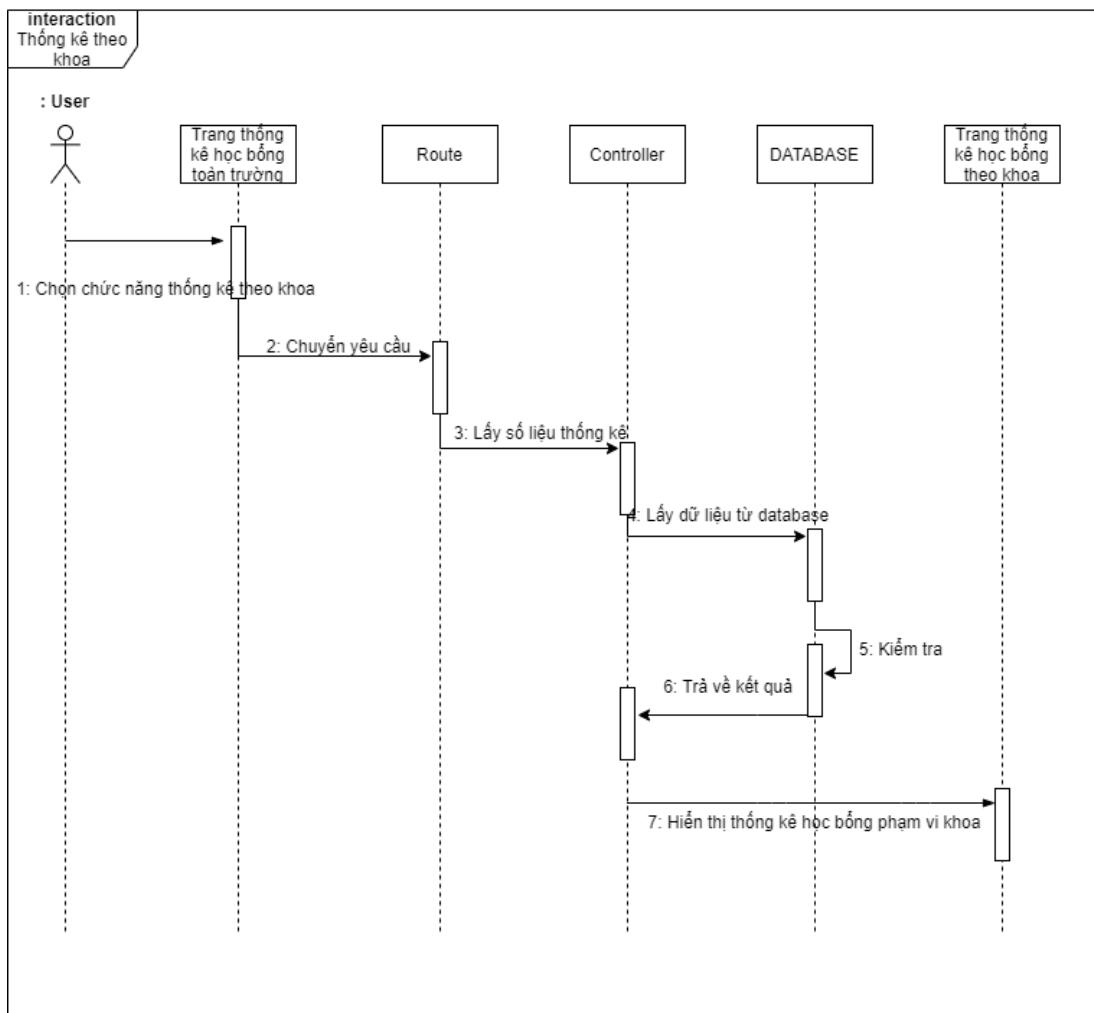
**Hình 16. Sơ đồ tuần tự Trao học bổng**

- ❖ Trang thống kê học bổng toàn trường



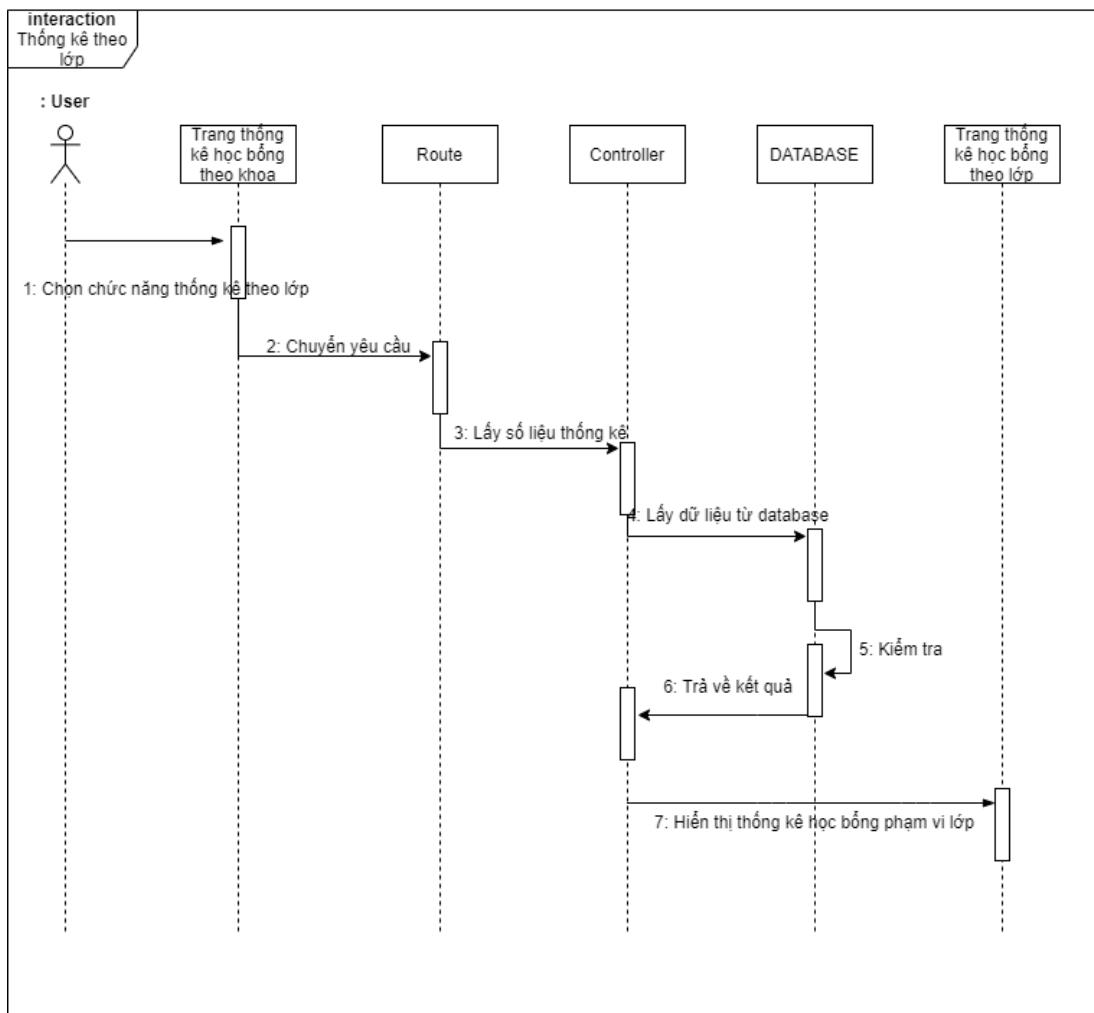
**Hình 17. Sơ đồ tuần tự Thống kê.**

- ❖ Trang học bổng theo khoa.



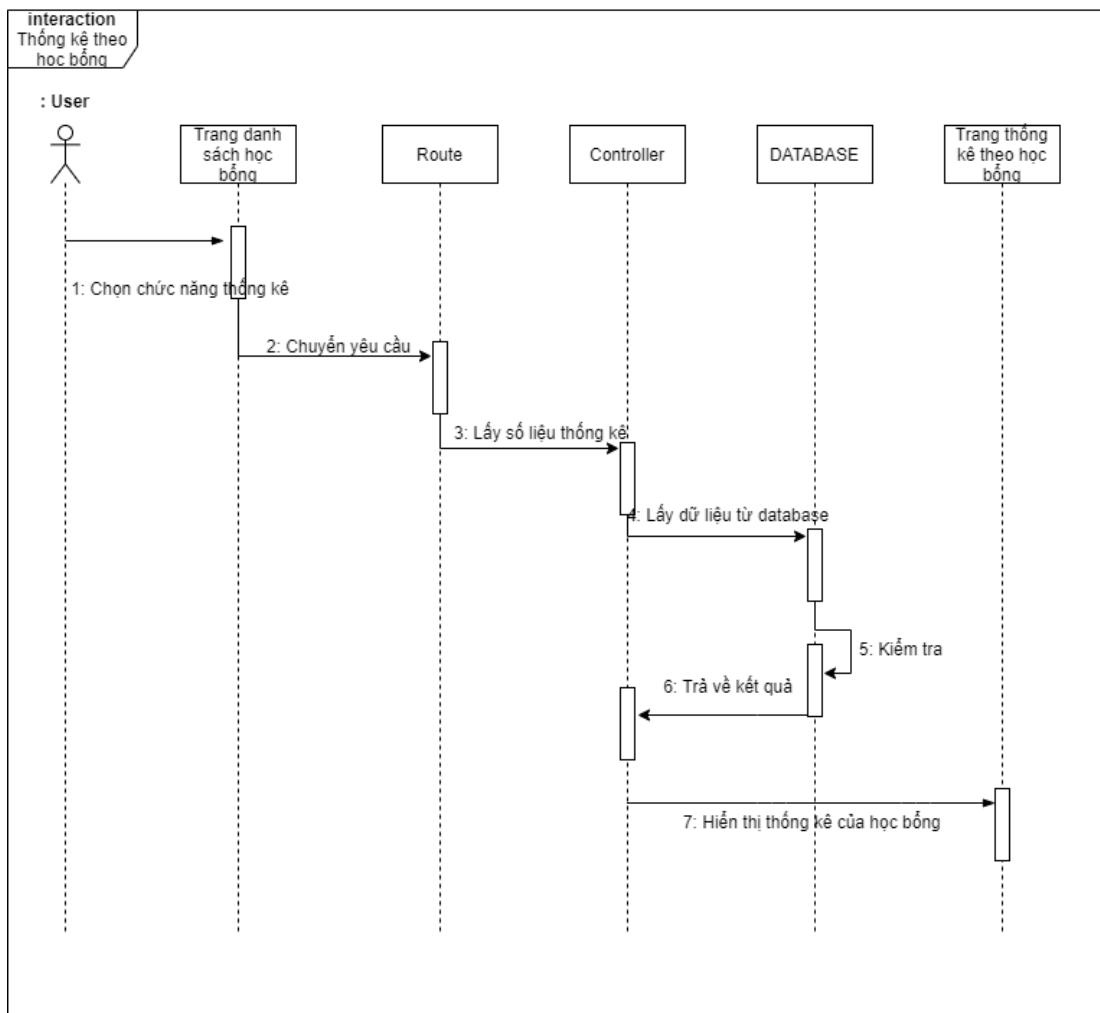
**Hình 18. Sơ đồ tuần tự Thống kê theo khoa.**

- ❖ Trang thống kê theo lớp



**Hình 19. Sơ đồ tuần tự Thống kê theo lớp.**

❖ Trang thống kê theo học bổng

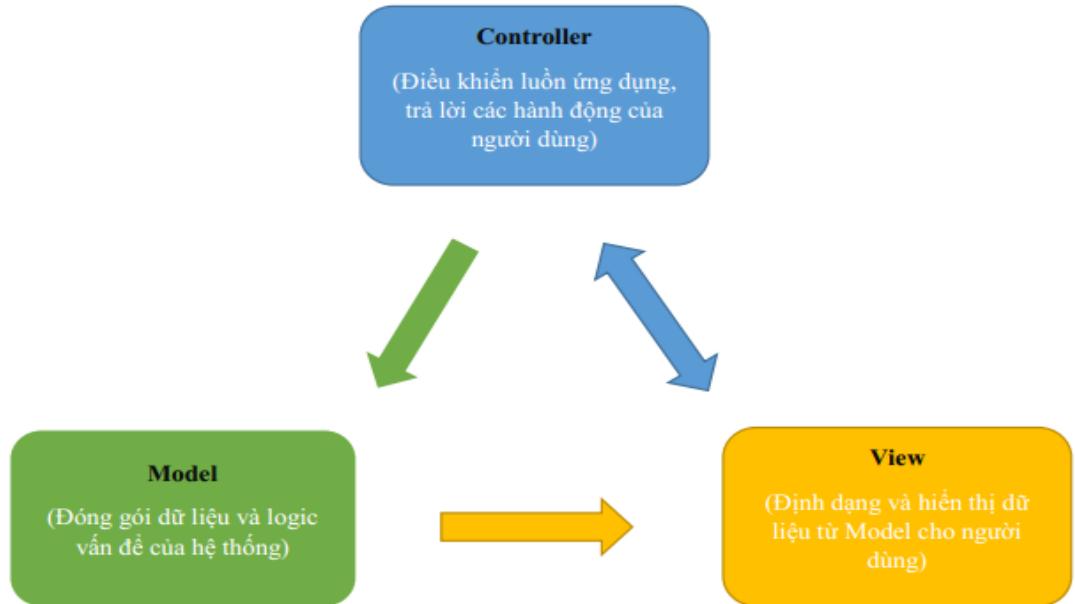


**Hình 20. Sơ đồ tuần tự Trang thống kê theo học bổng**

### 3.1.5 Thiết kế hệ thống

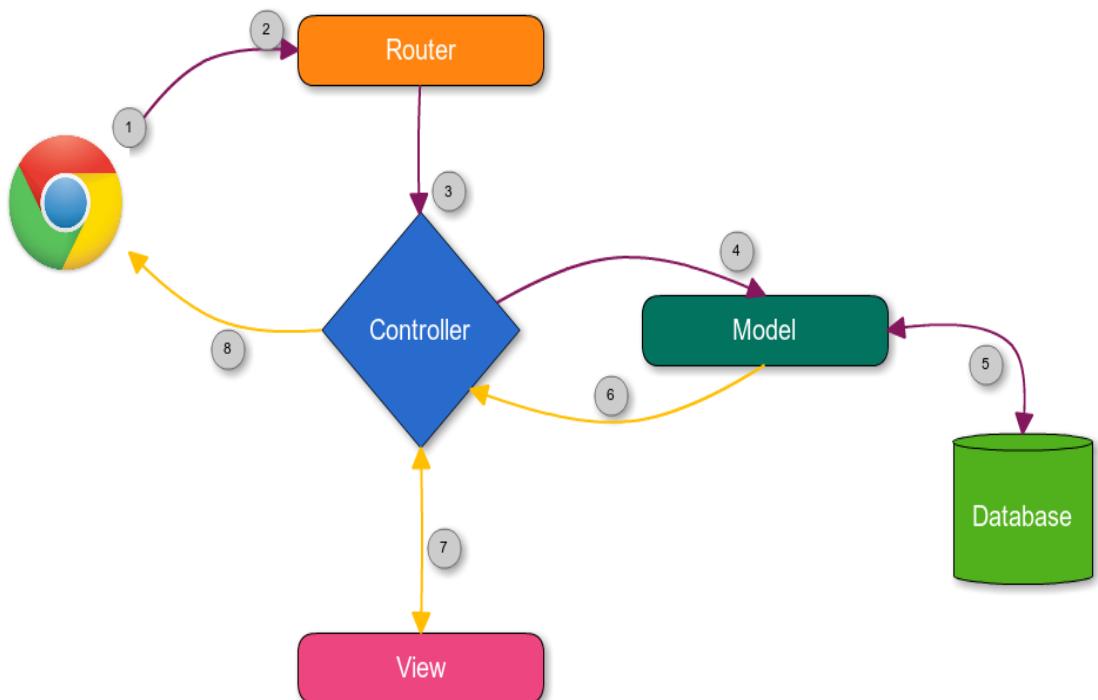
#### 3.1.5.1 Thiết kế cấu trúc

- Kiến trúc hệ thống



**Hình 21. Mô hình triển khai hệ thống.**

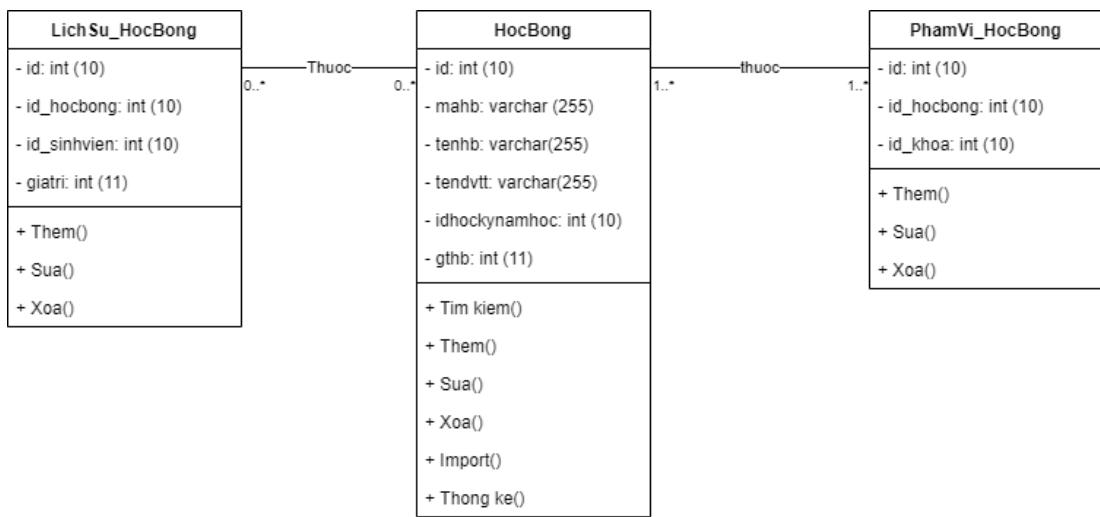
- Mô hình triển khai trên Internet



**Hình 22. Mô hình triển khai trên Internet.**

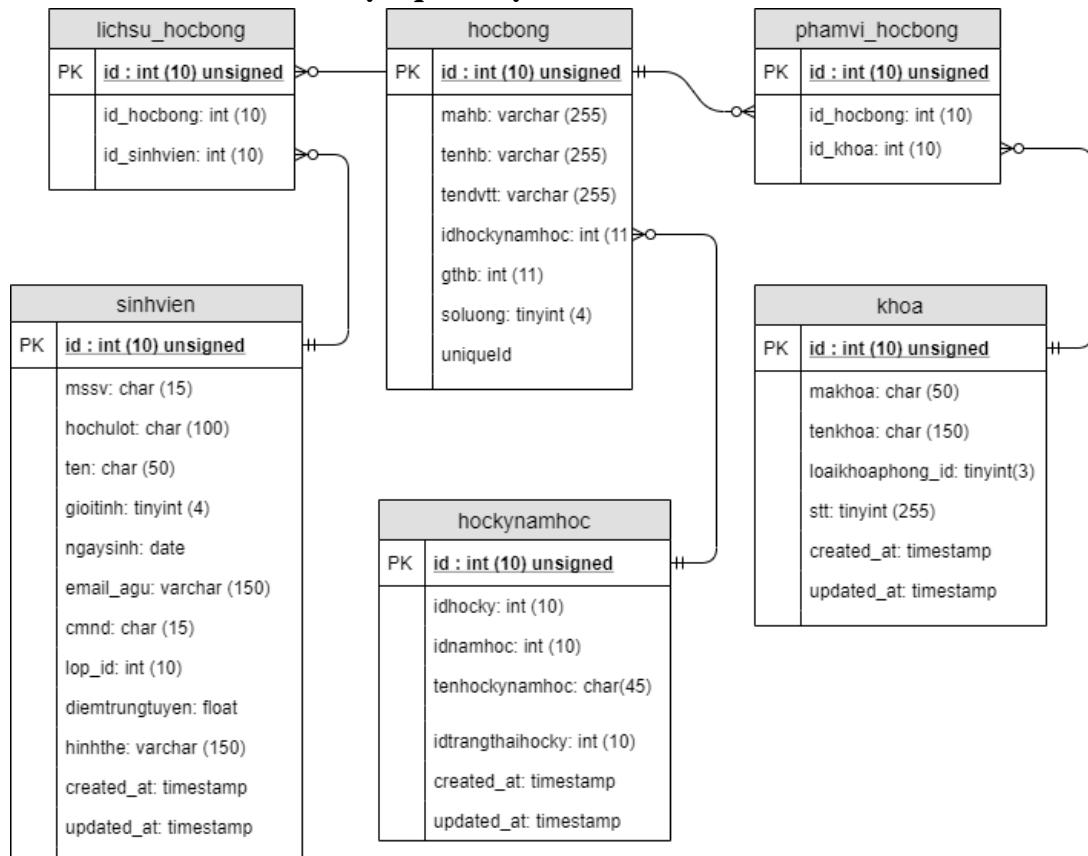
### 3.1.6 Thiết kế Cơ sở dữ liệu

#### 3.1.6.2 Sơ đồ lớp



**Hình 23. Sơ đồ lớp.**

### 3.1.6.2 Sơ đồ dữ liệu quan hệ



**Hình 24. Sơ đồ dữ liệu quan hệ**

### 3.1.7 Mô tả các bảng cơ sở dữ liệu

## Bảng học bổng

**Bảng 8. Bảng dữ liệu học bổng**

PK	Tên	Kiểu Dữ Kiệu	Mô Tả
#	id	Int(10)	Mã ID của học bổng, tự động tăng.
	mahb	Varchar(255)	Mã học bổng
	tenhb	Varchar(255)	Tên học bổng.
	tendvtt	Varchar(255)	Tên đơn vị tài trợ
	idhockynamhoc	Int (11)	Mã ID của Học kỳ năm học
	gthb	Int (11)	Giá trị học bổng
	soluong	Tinyint (4)	Số lượng học bổng
	created_at	Timestamp	Ngày tạo dữ liệu
	update_at	Timestamp	Ngày cập nhật dữ liệu

## Bảng Phạm vi học bổng

**Bảng 9. Bảng dữ liệu Phạm vi học bổng**

PK	Tên	Kiểu Dữ Kiệu	Mô Tả
#	id	Int(10)	Mã phạm vi của hệ thống, tự động tăng.
	id_hocbong	Int(10)	Mã ID của học bổng
	id_khoa	Int(10)	Mã ID của khoa
	created_at	Timestamp	Ngày tạo dữ liệu
	update_at	Timestamp	Ngày cập nhật dữ liệu

## Bảng Lịch sử học bổng

**Bảng 10. Bảng dữ liệu Lịch sử học bổng**

PK	Tên	Kiểu Dữ Kiệu	Mô Tả
#	id	Int(10)	Mã lịch sử học bổng của hệ thống, tự động tăng.
	id_hocbong	Int(10)	Mã ID của học bổng
	Id_sinhvien	Int(10)	Mã ID của sinh viên
	created_at	Timestamp	Ngày tạo dữ liệu

	update_at	Timestamp	Ngày cập nhật dữ liệu
--	-----------	-----------	-----------------------

### Bảng Học kỳ năm học

**Bảng 11. Bảng dữ liệu Học kỳ năm học**

PK	Tên	Kiểu Dữ Kiệu	Mô Tả
#	id	Int(10)	Mã học kỳ năm học của hệ thống, tự động tăng.
	idhockey	Int(10)	Mã ID của học kỳ
	idnamhoc	Int(10)	Mã ID của năm học
	tenhockynamhoc	Int (10)	Tên học kỳ năm học
	idtrangthaihockey	Int(10)	Mã ID trạng thái học kỳ
	created_at	Timestamp	Ngày tạo dữ liệu
	update_at	Timestamp	Ngày cập nhật dữ liệu

### Bảng Khoa

**Bảng 12. Bảng dữ liệu Khoa**

PK	Tên	Kiểu Dữ Kiệu	Mô Tả
#	id	Int(10)	Mã khoa của hệ thống, tự động tăng.
	makhoa	Char (50)	Mã khoa
	tenkhoa	Char(150)	Tên khoa
	loaikhoaphong_id	Tinyint(3)	Mã ID loại khoa phòng
	stt	Tinyint(255)	Số thứ tự
	created_at	Timestamp	Ngày tạo dữ liệu
	update_at	Timestamp	Ngày cập nhật dữ liệu

### Bảng Sinh viên

**Bảng 13. Bảng dữ liệu Sinh viên**

PK	Tên	Kiểu Dữ Kiệu	Mô Tả
#	id	Int(10)	Mã sinh viên của hệ thống, tự động tăng.
	mssv	Char (15)	Mã số sinh viên

	hochulot	Char(100)	Họ chữ lót
	ten	Char(50)	Tên
	gioitinh	tinyint(4)	Giới tính
	ngaysinh	date	Ngày sinh
	Email_agu	Varchar(150)	Email agu
	Cmnd	Int (10)	Chứng minh nhân dân
	Lop_id	Int (10)	Mã ID của lớp
	Diemtrungtuyen	Float	Điểm trúng tuyển
	Hinhthe	Varchar (150)	Hình thẻ
	created_at	Timestamp	Ngày tạo dữ liệu
	update_at	Timestamp	Ngày cập nhật dữ liệu

## **CHƯƠNG 4: THIẾT KẾ GIAO DIỆN**

### **TỔNG KẾT**

### **TÀI LIỆU THAM KHẢO**

