BỘ QUỐC PHÒNG

**HỌC VIỆN KỸ THUẬT QUÂN SỰ**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



**BÁO CÁO**

**THỰC TẬP CÔNG NGHIỆP**

**Họ và tên:** Đặng Đức Trung

**Lớp:** CNDL 14

**Mã sinh viên:** 15150419

**Supervisor:** Nguyễn Trọng Nhân

**Giáo viên phụ trách:** Vũ Đình Phái

Mục lục

[**I.** **Tên đề tài thực tập** 3](#_Toc8985451)

[**II.** **Nơi thực tập** 3](#_Toc8985452)

[**III.** **Nội dung thực tập** 3](#_Toc8985453)

[**1.** **Mục đích** 3](#_Toc8985454)

[**2.** **Các nội dung đã tham gia trong quá trình thực tập** 3](#_Toc8985455)

[**IV.** **Kết quả đạt được** 3](#_Toc8985456)

[**1.** **Nghiên cứu đề bài, thu thập tài liệu** 3](#_Toc8985457)

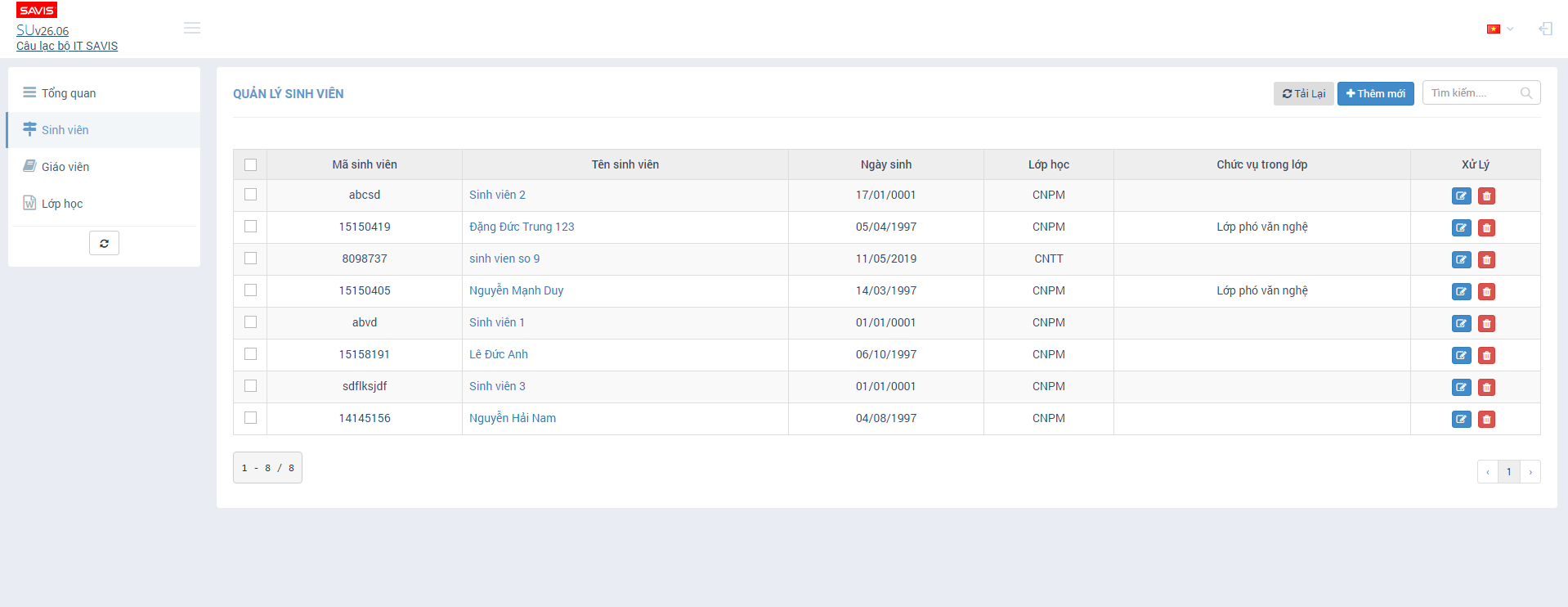
[**2.** **Các nội dung đã tiến hành trong quá trình thực tập** 3](#_Toc8985458)

[**3.** **Xây dựng chương trình** 5](#_Toc8985459)

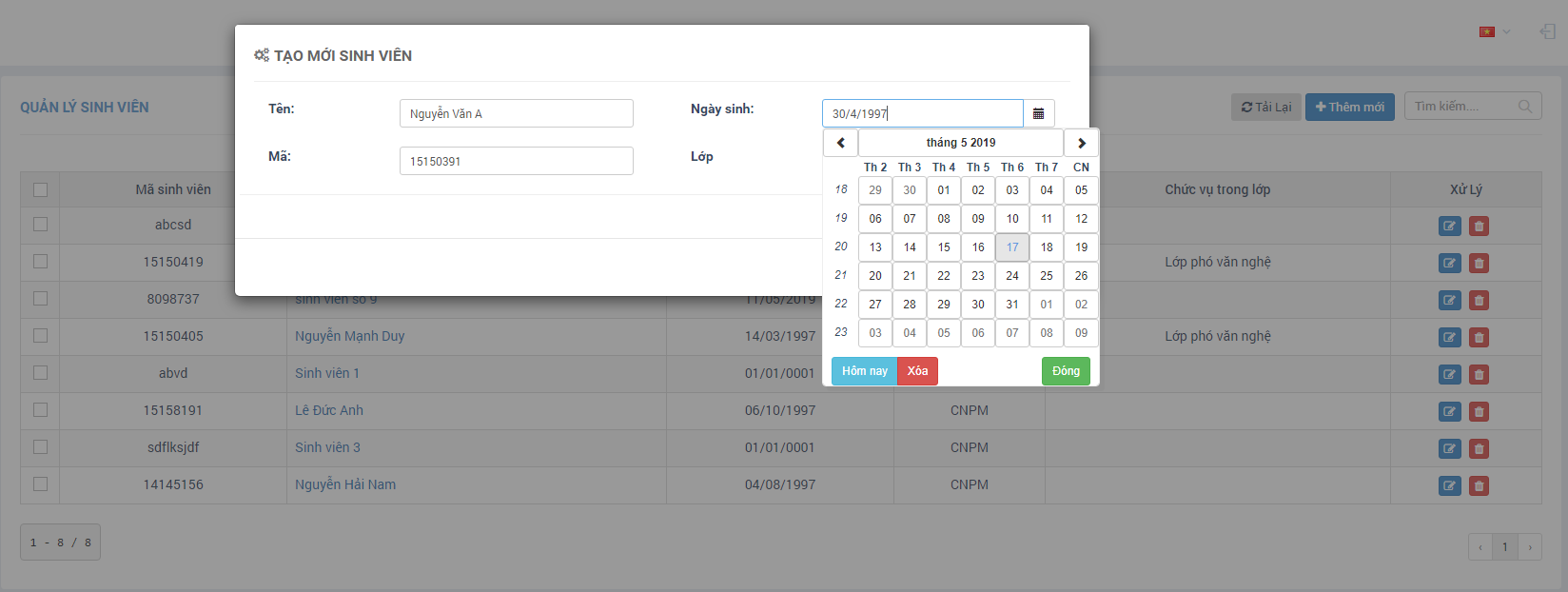
[**4.** **Những khó khăn trong quá trình thực tập** 10](#_Toc8985460)

[**V.** **Xác nhận của supervisor** 11](#_Toc8985461)

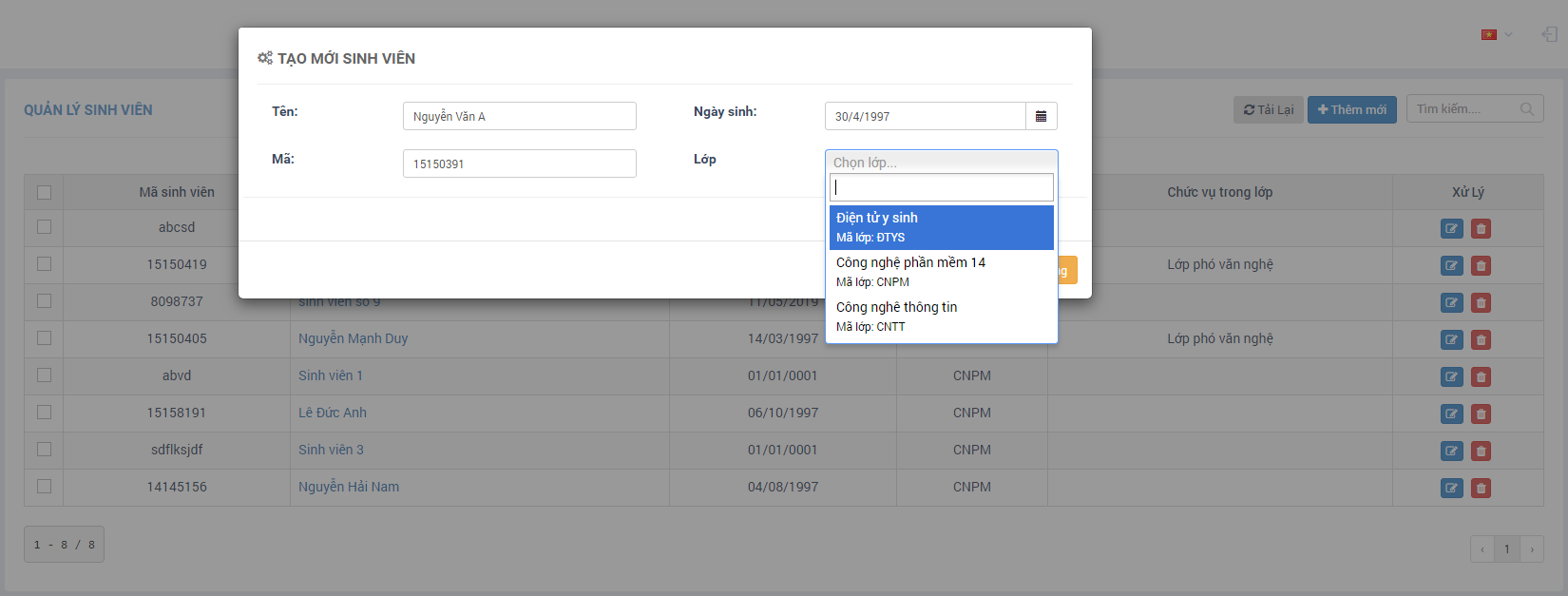
1. **Tên đề tài thực tập**
   1. Đề tài thực tập: tìm hiểu và sử dụng các framework của công ty và lập trình ứng dụng quản lý sinh viên đáp ứng các yêu cầu đề ra.
   2. Mô tả công việc:
      1. Ôn tập C#, SQL, Entity Framework.
      2. Đọc mô tả nghiệp vụ và thiết kế, xây dựng cơ sở dữ liệu.
      3. Tìm hiểu và lập trình ứng dụng web sử dụng API và AngularJS.
2. **Nơi thực tập**
   1. **Công ty thực tập: Công ty cổ phần công nghệ SAVIS**
   2. **Địa chỉ: Tầng 9, tòa nhà Việt Á, số 9 phố Duy Tân, phường Dịch Vọng Hậu, quận Cầu Giấy, TP. Hà Nội.**
   3. **Điện thoại: +(84-24) 3782 2345**
3. **Nội dung thực tập**
   1. **Mục đích**
      1. Tiếp cận với phương pháp lập trình ứng dụng mới sử dụng API.
      2. Phân tích bài toán và thiết kế database phù hợp.
      3. Làm quen và sử dụng framework của công ty.
   2. **Các nội dung đã tham gia trong quá trình thực tập**
      1. Ôn tập C# và tuân thủ naming convention.
      2. Tìm hiểu và thực hành về RESTful API.
      3. Tìm hiểu ASP.NET Web API 2 và ASP.NET Core Web API.
      4. Ôn tập SQL và Entity Framework.
      5. Lập trình API đáp ứng các yêu cầu của bài toán quản lý sinh viên theo framework của công ty.
      6. Tìm hiểu về Javascript framework: AngularJS.
      7. Lập trình giao diện sử dụng AngularJS theo framework của công ty.
4. **Kết quả đạt được**
   1. **Nghiên cứu đề bài, thu thập tài liệu**
      1. Đọc tài liệu về convention, qui trình làm việc và giới thiệu về API.
      2. Đọc documentation về lập trình API của Microsoft sử dụng .NET Framework và .NET Core Framework và thực hiện các ví dụ minh họa.
      3. Đọc file Excel mô tả nghiệp vụ của phần mềm cần xây dựng.
      4. Đọc documentation và hướng dẫn lập trình bằng framework AngularJS.
   2. **Các nội dung đã tiến hành trong quá trình thực tập**
      1. Đọc các tài liệu yêu cầu
         * Đọc tài liệu về qui trình làm việc: nắm bắt cơ bản về qui trình, thứ tự các công việc phải làm của một lập trình viên backend: Trước tiên ta phải nắm bắt được nghiệp vụ của bài toán tin học hóa. Sau đó thực hiện phân tích thiết kế chức năng cần thiết. Hoàn thành bước trên ta sẽ thực hiện thiết kế nguyên mẫu của phần mềm (class, interface, function…) đáp ứng các chức năng đã phân tích. Tiếp theo là thiết kế các API dựa trên nguyên mẫu đã xây dựng. Đến đây chúng ta đã hình dung được hoạt động của hệ thống, vì vậy sẽ bắt tay thiết kế CSDL để lưu trữ. Sau khi có đầy đủ tài liệu thiết kế, chúng ta sẽ bắt đầu lập trình từ dữ liệu cho đến các hàm xử lý nghiệp vụ và cuối cùng là ApiController. Việc cuối cùng nhưng không kém quan trọng đó là kiểm thử khả năng hoạt động của API và debug nếu có lỗi xảy ra.
         * Đọc tài liệu về naming convention: đặt tên các biến, hàm, thực thể… thuộc phạm vi chương trình một cách hợp lý, tên phải dễ hiểu và cần được thống nhất bằng cách sử dụng các qui tắc đặt tên như camelCase cho biến cục bộ, tham số; PascalCase được sử dụng cho tên class, interface, attribute, function, thêm các tiền tố như “is” cho kiểu bool, “str” cho string… và dấu “\_” cho các biến private. Các hằng số cần viết hoa toàn bộ và các chữ cách nhau bởi dấu “\_”.
         * Đọc tài liệu giới thiệu về REST API: API(Application Programming Interface) là một phương thức kết nối, tương tác các ứng dụng với nhau mà không cần mã nguồn hoặc lập trình các module kết nối riêng biệt cho từng ứng dụng khác nhau. REST(Representational State Transfer) là một thể thức giao tiếp giữa các ứng dụng không cần tồn tại trạng thái (session của client-server) và không cần biết tới bên cung cấp xử lý như thế nào.
      2. Làm các bài tập thực hành lập trình API theo mẫu của Microsoft documentation
         * Đọc tài liệu về REST API và làm theo các ví dụ trên cả .NET Framework và .NET Core Framework. Kết hợp sử dụng Entity Framework Code First. Các bài tập làm theo mẫu hướng dẫn của Web API 2 và .NET Core Web API.
         * Kết hợp ôn tập cách sử dụng Entity Framework Code First/Database First theo documentation của Microsoft.
      3. Đọc tài liệu mô tả và thiết kế prototype, cơ sở dữ liệu
         * Đọc các yêu cầu về thông tin cần lưu trữ và nghiệp vụ trong tài liệu mô tả để thiết kế các prototype class cho API bao gồm các interface, các chức năng, các model dữ liệu.
         * Xác định và thiết kế các API cần thiết phục vụ đúng và đủ các nghiệp vụ được mô tả trong tài liệu.
         * Thiết kế CSDL đáp ứng dựa trên nguyên mẫu đã lập, chuẩn hóa.
         * Tạo thêm các bảng dư thừa và trigger phục vụ cho lưu lại lịch sử dữ liệu.
      4. Lập trình và kiểm thử các API đáp ứng nghiệp vụ
         * Tìm hiểu framework backend của công ty và thực hiện theo khung: sử dụng Entity Framework Database First để import model của CSDL và auto-generate ra các class dựa theo các bảng trong CSDL. Tập hợp các phương thức truy vấn dữ liệu thành một interface.
         * Định nghĩa chi tiết xử lý, truy vấn dữ liệu và nội dung trả về cho generic class (tức là sử dụng chung cho các class).
         * Viết các DTO(Data Transfer Object) để chuyển đổi cấu trúc dữ liệu thô từ CSDL sang dạng hướng đối tượng.
         * Viết các interface chứa các function đáp ứng chức năng nghiệp vụ đã đề ra. Định nghĩa chi tiết xử lý trong các class handler.
         * Định nghĩa các ApiController chứa các API và qui định đường dẫn để đáp ứng các chức năng đã đặt ra theo từng thực thể.
         * Sử dụng các đường dẫn đã qui định để truy cập và kiểm thử API bằng phần mềm Postman/Fiddler.
      5. Lập trình giao diện ứng dụng web
         * Tìm hiểu framework của công ty được xây dựng bằng AngularJS.
         * Xây dựng service $routeProvider để định nghĩa cho các đường dẫn.
         * Xây dựng service theo nhóm thực thể gọi đến từng API cụ thể bằng đường dẫn đã được qui định.
         * Xây dựng các view và controller tương ứng và inject các service cần thiết.
   3. **Xây dựng chương trình**
      1. **Các chức năng dự kiến xây dựng**
         * Các chức năng thêm, sửa, xóa cơ bản với các thực thể, tìm kiếm theo tên, mã và tìm kiếm full-text search.
         * Cụ thể về chức năng xóa: do CSDL thiết kế để lưu trữ lại lịch sử thay đổi của một số thực thể, do vậy dữ liệu sẽ không thật sự xóa mà chỉ chuyển trạng thái IsActive sang false.
         * Đối với lớp: có thêm chức năng gán/sửa giáo viên, thay đổi người giữ chức vụ cán bộ lớp.
         * Đối với sinh viên: có thêm chức năng vào ra, lớp học.
         * Đối với giáo viên: có liệt kê được lịch sử các lớp học đã quản lý
      2. **Các chức năng đã xây dựng được**
         * Các chức năng trong quản lý sinh viên
           1. Giao diện chính



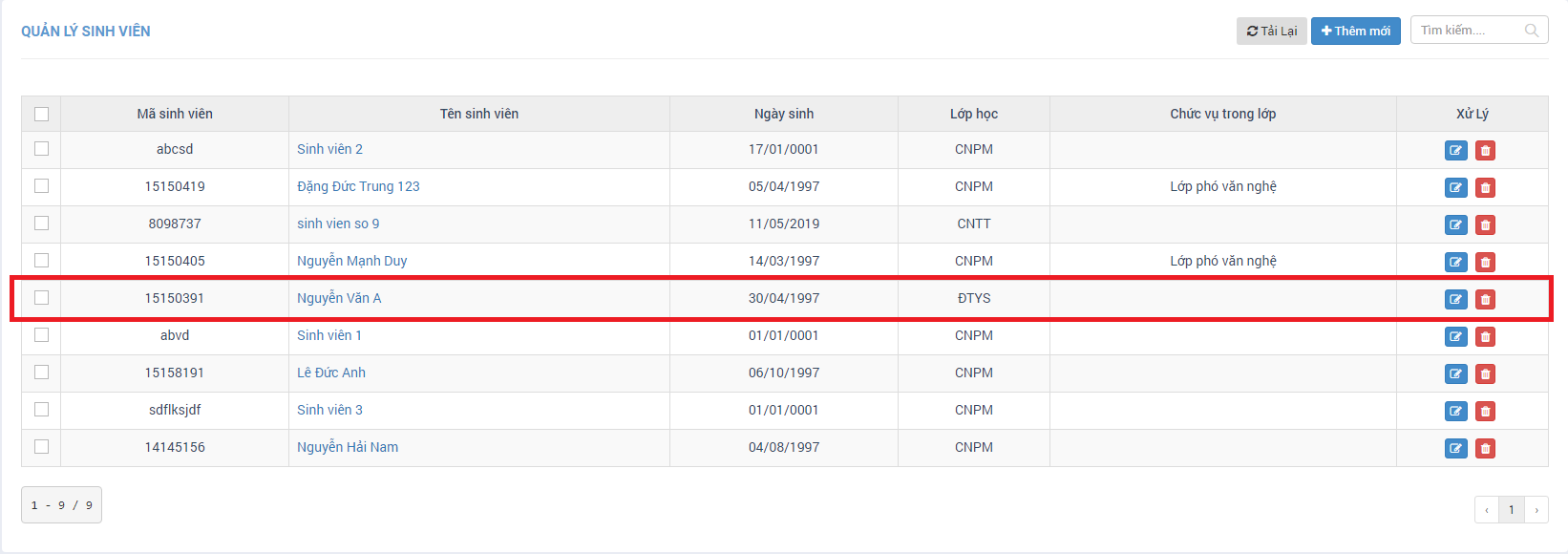
* + - * 1. Chức năng thêm mới



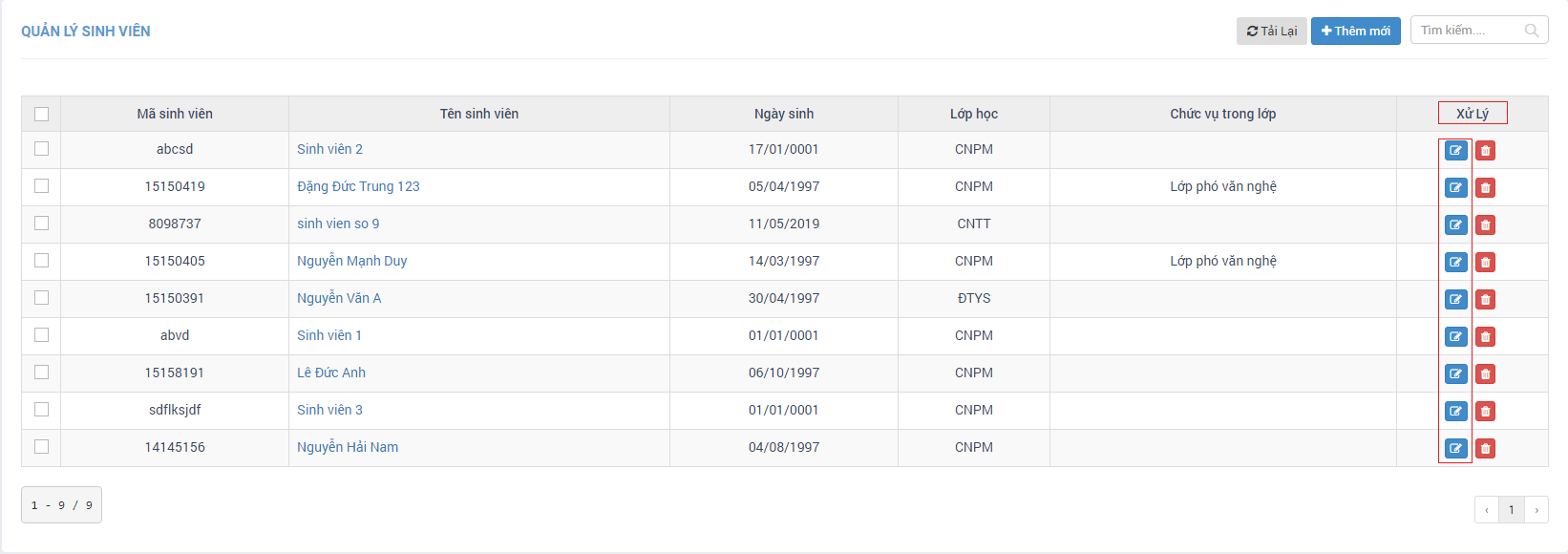
* Người dùng nhập tên sinh viên và mã cho sinh viên đó (hiện tại chưa định nghĩa format các loại mã để xây dựng cơ chế sinh mã tự động)



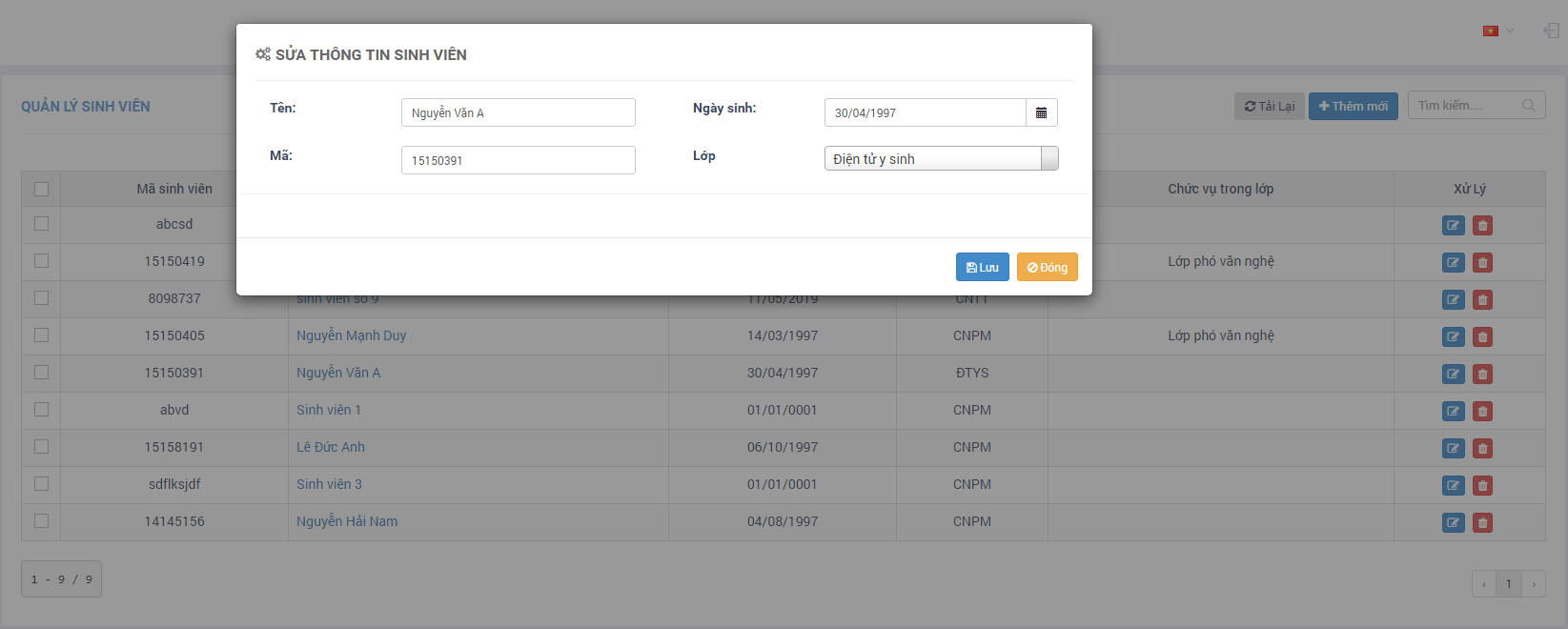
* Người dùng có thể chọn ngày sinh từ DateTimePicker hoặc nhập vào theo định dạng “dd/MM/yyyy” và tìm kiếm lớp cho sinh viên hoặc có thể để trống. Tất cả các trường cần nhập đủ trừ chọn lớp.



* + - * 1. Chức năng sửa



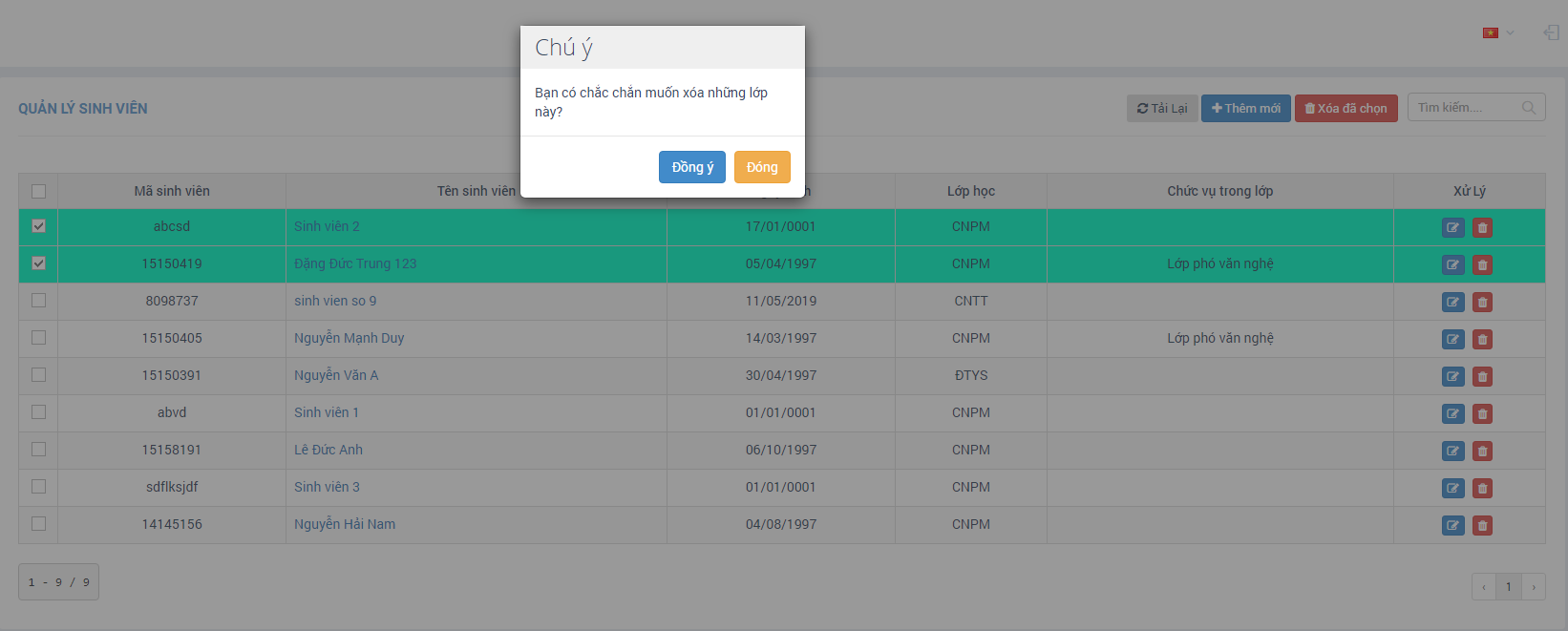
* Click chọn biểu tượng sửa trong cột xử lý hoặc tên sinh viên và cửa sổ sau sẽ hiện lên



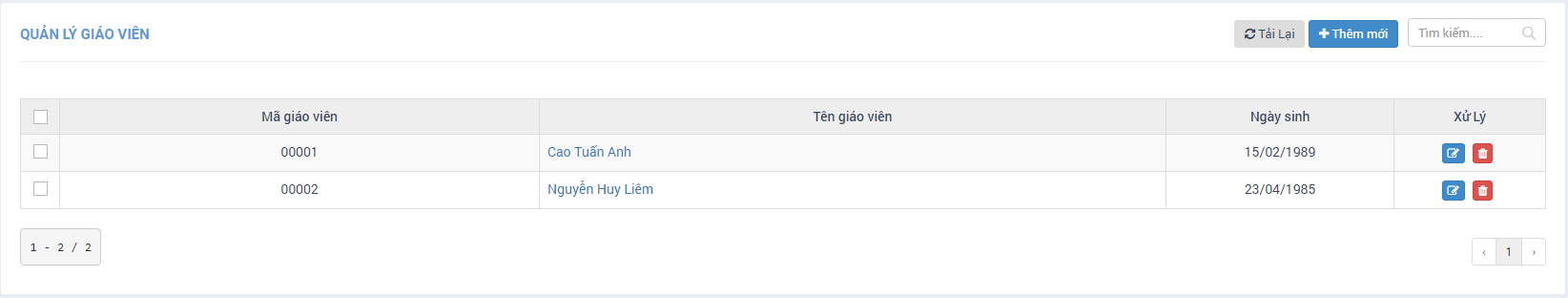
* Thực hiện sửa xóa theo nhu cầu
  + - * 1. Chức năng xóa, xóa nhiều cùng lúc



* Đơn xóa từng sinh viên sử dụng nút xóa có biểu tượng thùng rác bên cột xử lý
* Khi tích chọn từ hai bản ghi trở lên, nút “Xóa đã chọn” sẽ hiện ra.
* Trước khi xóa sẽ có cửa sổ cảnh báo xác nhận xóa.



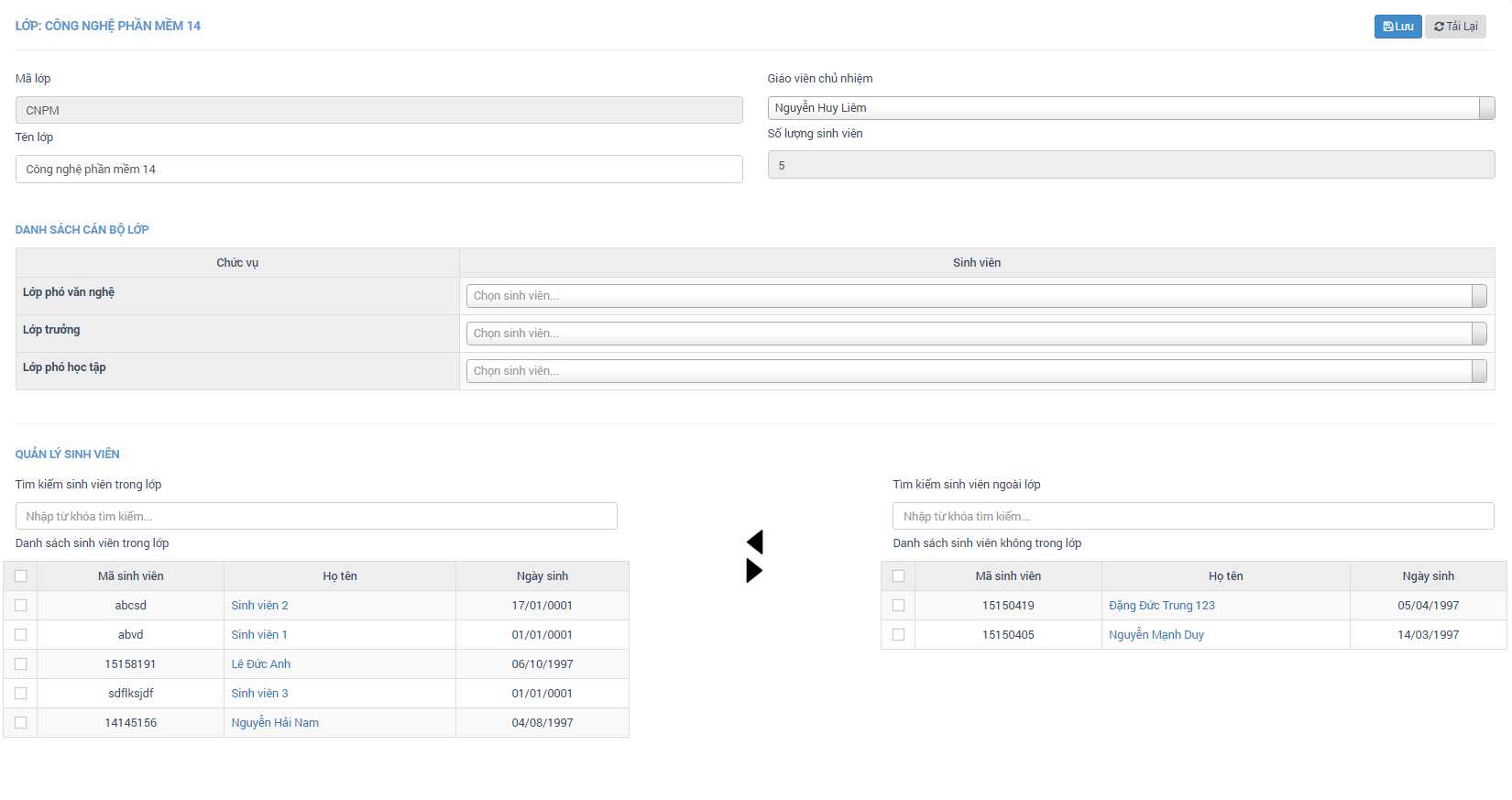
* + - * 1. Chức năng tải lại dữ liệu: chọn nút tải lại để load lại dữ liệu
      * Các chức năng trong quản lý giáo viên
        1. Giao diện chính



* + - * 1. Chức năng thêm mới: tương tự quản lý sinh viên
        2. Chức năng sửa: tương tự quản lý sinh viên
        3. Chức năng xóa, xóa nhiều cùng lúc: tương tự quản lý sinh viên
        4. Chức năng tải lại dữ liệu: chọn nút tải lại để load lại dữ liệu
      * Các chức năng trong quản lý lớp học
        1. Giao diện chính



* + - * 1. Chức năng thêm mới: tương tự quản lý sinh viên và giáo viên
        2. Chức năng sửa: tương tự quản lý sinh viên và giáo viên
        3. Chức năng xóa, xóa nhiều cùng lúc: tương tự quản lý sinh viên và giáo viên
        4. Chức năng tải lại dữ liệu: chọn nút tải lại để load lại dữ liệu
        5. Các chức năng chi tiết cho một lớp học



* Giao diện dễ hiểu: ta có thể thực hiện sửa đổi trong lớp học như sửa tên; tìm kiếm và sửa giáo viên chủ nhiệm; tìm kiếm và sửa danh sách cán bộ lớp; tìm kiếm và sửa danh sách cán bộ lớp
* Sử dụng các hai nút mũi tên màu đen để đưa sinh viên vào/ra lớp.
  + 1. **Các chức năng trong dự kiến nhưng chưa xây dựng được**
       - Chưa thực hiện được full-text search do đặt sai collation của CSDL.
       - Hiển thị danh sách cán bộ lớp: render các tag <ui-select> bằng vòng lặp nên gặp khó khăn trong quá trình binding “selected-model” với dữ liệu trong CSDL.
       - Phân quyền: khi sử dụng API, thông thường sẽ thông qua bên thứ 3 để tạo token cho việc xác thực. Có thể phân quyền bằng cách xử lý cách ứng dụng render các view và redirect. Tuy nhiên thời gian làm quen với framework còn hạn chế, chưa thể hoàn thiện phân quyền nhưng đã có thiết kế phân quyền đơn giản trong CSDL.
  1. **Những khó khăn trong quá trình thực tập**
     1. Thời gian thực tập ngắn, tìm hiểu và làm quen với các framework sẽ tốn khá nhiều thời gian với những bạn không quen đọc hiểu và truy code.
     2. Không thể hoàn thành một số hạng mục kế hoạch đã vạch ra theo đúng thời gian do chưa có nhiều kiến thức về những mảng đó và mất rất nhiều thời gian để tìm giải pháp hoặc đọc hiểu (các) documentation liên quan.

1. **Xác nhận của supervisor**

**Họ tên cán bộ hướng dẫn** **Kí tên/Ngày**

(Approving supervisor’s title and name) (Date and signature)

(Đóng dấu treo của cơ quan)