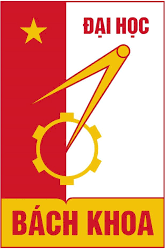
TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN ĐÀO TẠO QUỐC TẾ



**Software design description**

**Đồ án các công nghệ xây dựng hệ thống thông tin**

**Đề tài: Hệ thống quản lí điểm và hồ sơ sinh viên**

**Giáo viên hướng dẫn: TS. Đào Thành Chung**

**Nhóm sinh viên thực hiện:**

**Vũ Văn Cường-20168069**

**Nguyễn Quốc Trung-20158391**

**Hà Nội, Tháng 10 năm 2018**

# **CHƯƠNG 1: GIỚI THIỆU**

**1.1. Mục đích**

* Tài liệu này là bản thiết kế chi tiết hệ thống, bao gồm các thiết kế: phân tích chi tiết lớp (analyst class), cơ sở dữ liệu (database), giao diện người dùng (graphic userinterface).
* Dựa trên phân tích có trong tài liệu đặc tả SRS đưa ra quy trình hoạt động của hệ thống thông qua các biểu đồ phân tích chi tiết lớp và biểu đồ tuần tự (sequence diagram) tương ứng cho các hành vi của từng ca sử dụng.
* Đưa ra thiết kế cơ sở dữ liệu cụ thể, cơ sở dữ liệu cho phép lưu trữ thông tin về người dùng, thông tin về điểm và bằng cấp sinh viên.
* Từ tài liệu thiết kế này, nhóm phát triển hệ thống có cơ sở để lập trình phát triển, hình thành hệ thống cho người dùng. Thuận tiện cho việc kết nối các module của hệ thống với nhau.

**1.2. Cấu trúc tài liệu**

Bản thiết kế này chia làm 5 phần chính:

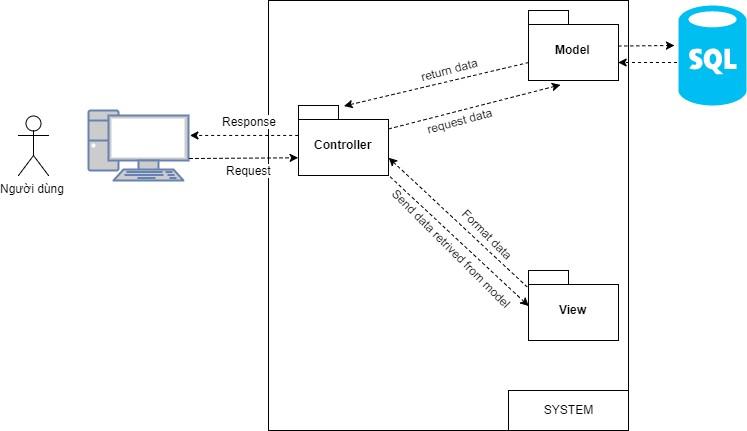
* 1. Mô hình hóa cấu trúc hệ thống: Cấu trúc hệ thống tổng quan
  2. Mô hình hóa hành vi hệ thống: Trình tự hoạt động
  3. Thiết kế cơ sở dữ liệu
  4. Thiết kế giao diện người dùng

**CHƯƠNG 2: MÔ HÌNH HÓA CẤU TRÚC HỆ THỐNG**

**2.1. Các tầng và mối quan hệ trên hệ thống**

2.1.1. Hệ thống được xây dựng dựa theo mô hình MVC

* Tổng quan mô hình MVC (Model – View - Controller)



*Hình 1. Mô hình MVC*

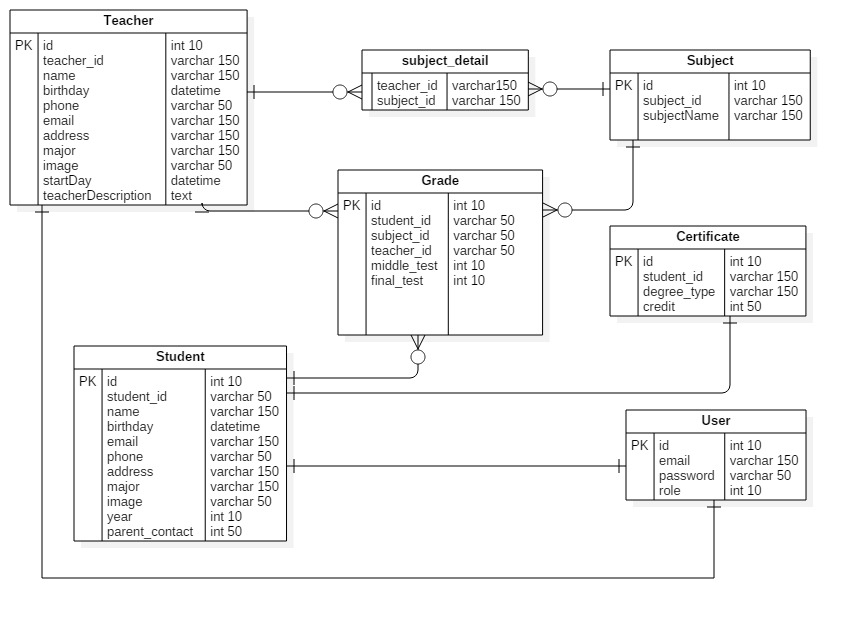
* + Chức năng của từng tầng trong mô hình MVC
    - **Controller:** Đây là nơi đầu tiền nhận các yêu cầu (requests), phân tích yêu cầu, khởi tạo và gọi model, sau đó nhận các hồi đáp (response) từ model và gửi ra các lớp giao diện (view). Trong thực tế, Controller được gọi từ điểm đầu vào của ứng dụng(mà thường là các file index.php hoặc index.html). Tập tin này sẽ giao toàn bộ các yêu cầu gửi từ client cho controller xử lý.
    - **Model**: Đại diện cho dữ liệu và logic của ứng dụng, thường hay gọi là business logic, có trách nhiệm:
* Thêm, sửa, xoá dữ liệu…
* Là nơi thực thi logic nghiệp vụ của ứng dụng.
  + - **View**: Chịu trách nhiệm định đạng lại dữ liệu được truyền ra từ model. Dữ liệu được truyền ra có thể có nhiều định dạng khác nhau tùy vào cách người lập trình xử lý như là xml, json, array.

2.1.2. **Mô tả kịch bản hệ thống hướng theo mô hình MVC**

* Kịch bản đăng nhập: User gửi yêu cầu đăng nhập đến Controller(LoginController), LoginController tiếp nhận yêu cầu, LoginController lấy kết quả xác định tồn tại user hay không từ lớp Model(User), rồi gọi lớp View tương ứng với các trường hợp tồn tại và không tồn tại user trong database.
* Kịch bản tìm kiếm thông tin: User gửi yêu cầu tìm kiếm đến Controller(search), Controller tiếp nhận yêu cầu và lấy thông tin trùng với thông tin mà người dùng đang tim kiếm tìm kiếm trong Model, rồi trả về lớp View tương ứng với các trường hợp tồn tại và không tồn tại trong database
* Kịch bản xem thông tin: User gửi yêu cầu xem thông tin đến Controller(view), Controller lấy kết quả từ Model, rồi trả về thông tin chi tiết trên lớp View tương ứng.
* Kịch bản sửa thông tin: User gửi yêu cầu sửa thông tin đến Controller(edit), Controller lấy kết quả từ Model, rồi trả về những thông tin có thể sửa trên lớp view. User nhập thông tin cần sửa rồi gửi yêu cầu lưu thông tin. Controller tiếp nhận yêu cầu rồi lưu thông tin đã thay đổi vào database.
* Kịch bản thêm thông tin (thêm điểm, người dùng): User gửi yêu cầu thêm thông tin(điểm, người dùng) đến Controller(add), Controller lấy kết quả bảng trong Model rồi trả về View tương ứng. User nhập thông tin cần thêm rồi gửi yêu cầu lưu thông tin. Controller tiếp nhận yêu cầu rồi lưu thông tin vào database

2.2. Sơ đồ triển khai hệ thống

2.3. Mô hình hoá lĩnh vực ứng dụng

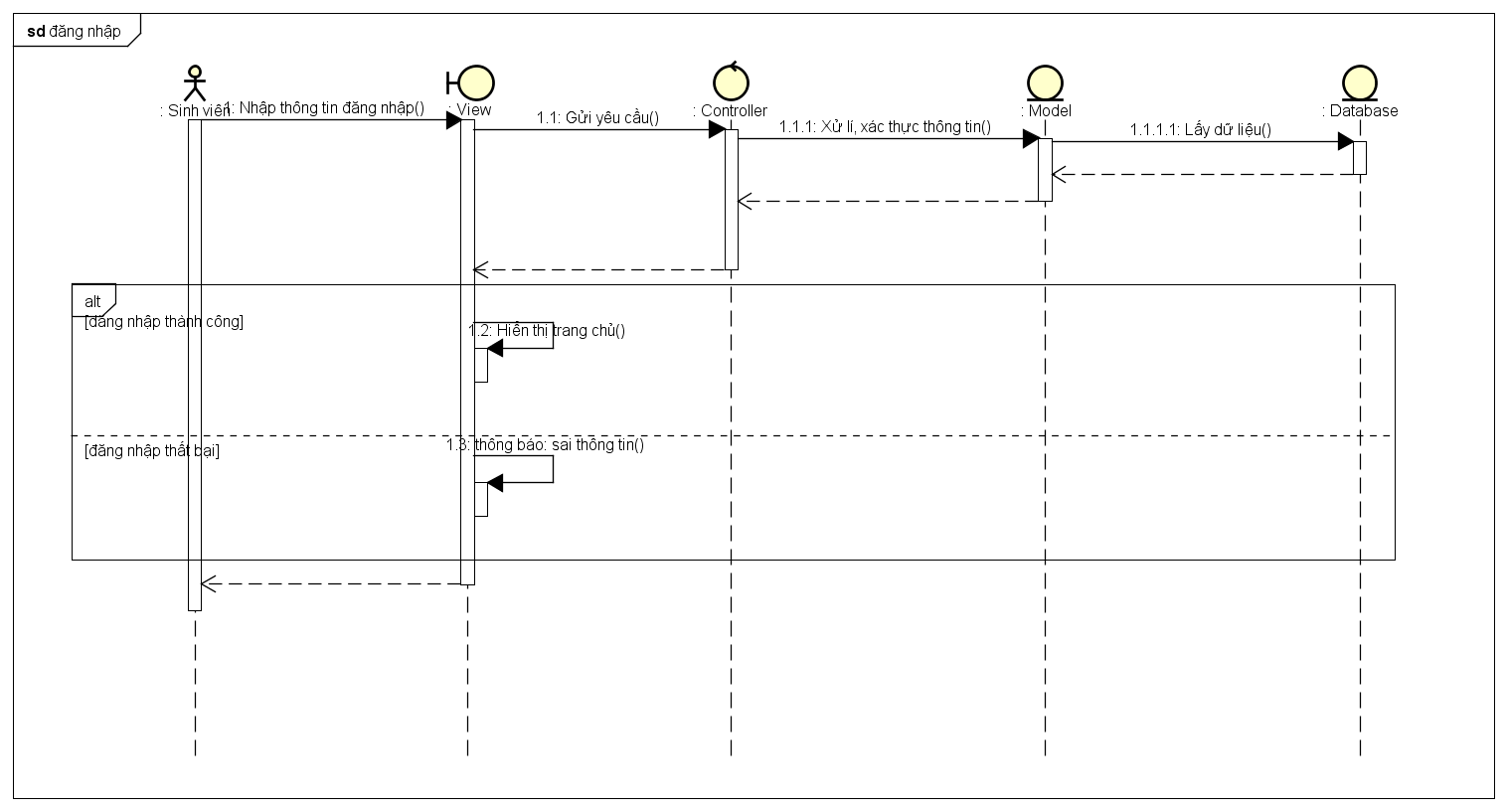


2.4. Các lớp tham gia ca sử dụng.

Các lớp tham gia ca sử dụng gồm 3 loại:  
❖ Boundaries (các lớp biên hay các lớp đối thoại): tương ứng với tầng View  
❖ Entities (các lớp thực thể hay các lớp lĩnh vực): tương ứng với tầng Model  
❖ Controls (các lớp điều khiển): tương ứng với tầng Controller

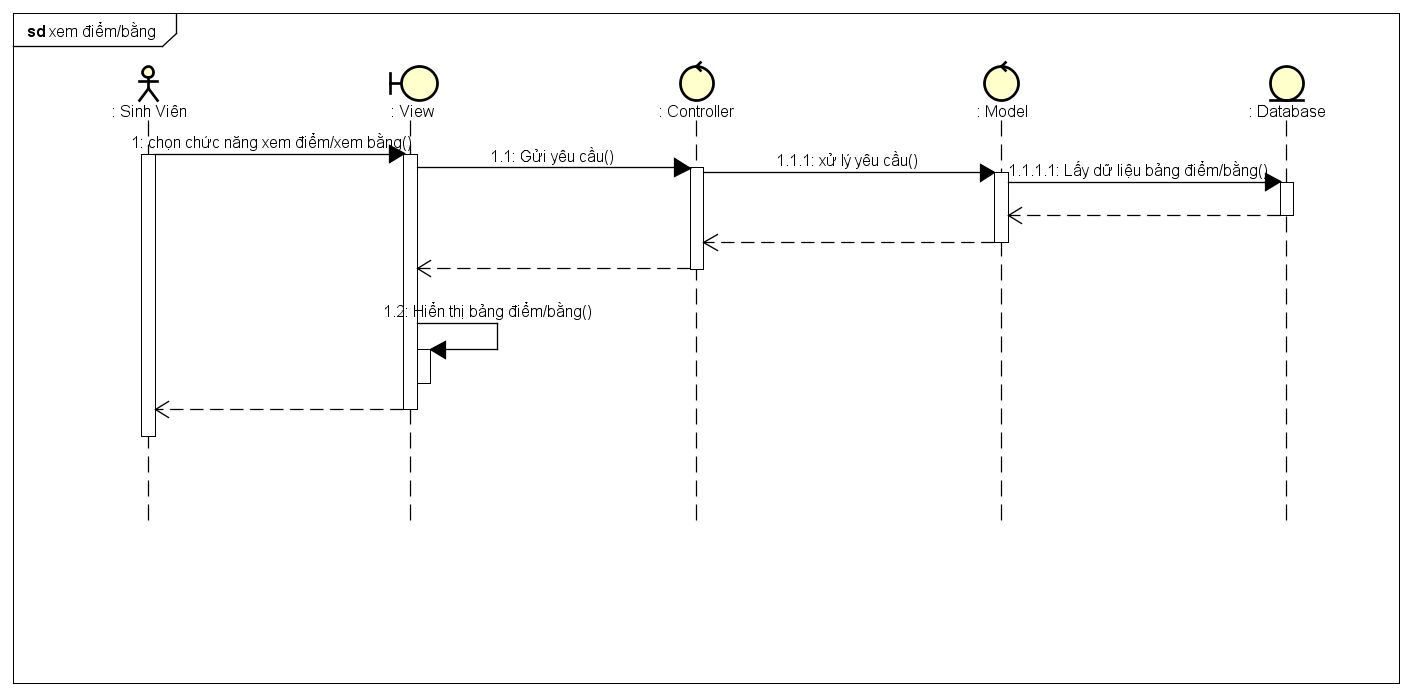
**CHƯƠNG 3: MÔ HÌNH HOÁ HÀNH VI HỆ THỐNG**

**3.1. Hành vi Đăng nhập**

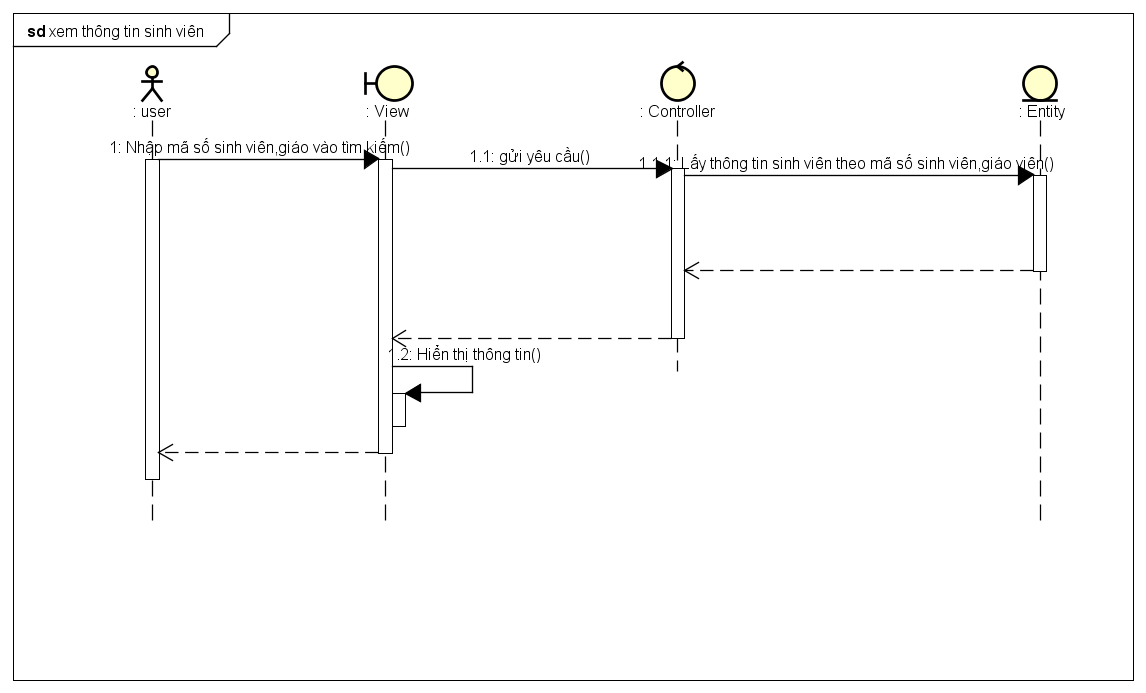


*Hình 10: Biểu đồ Hành vi Đăng nhập*

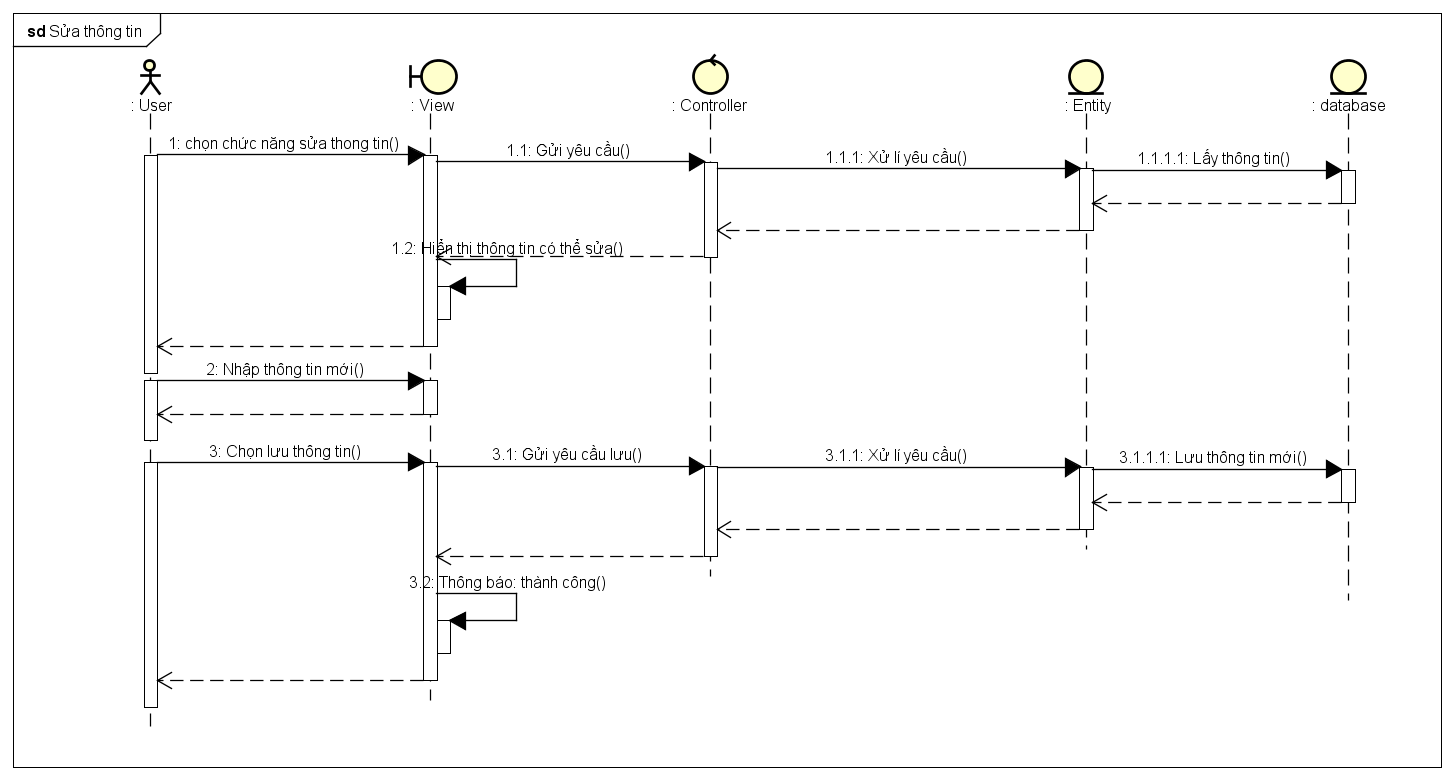
**3.2. Hành vi Xem điểm, bằng**



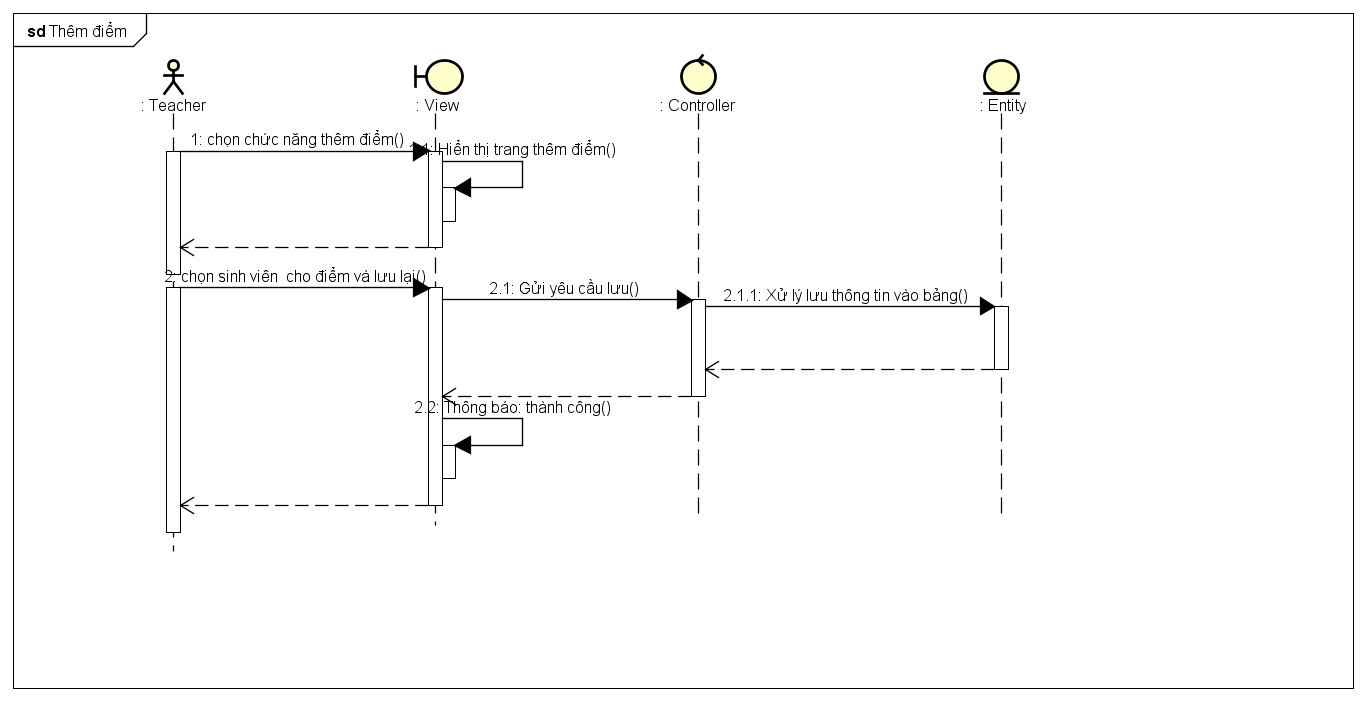
**3.3. Hành vi Xem thông tin**



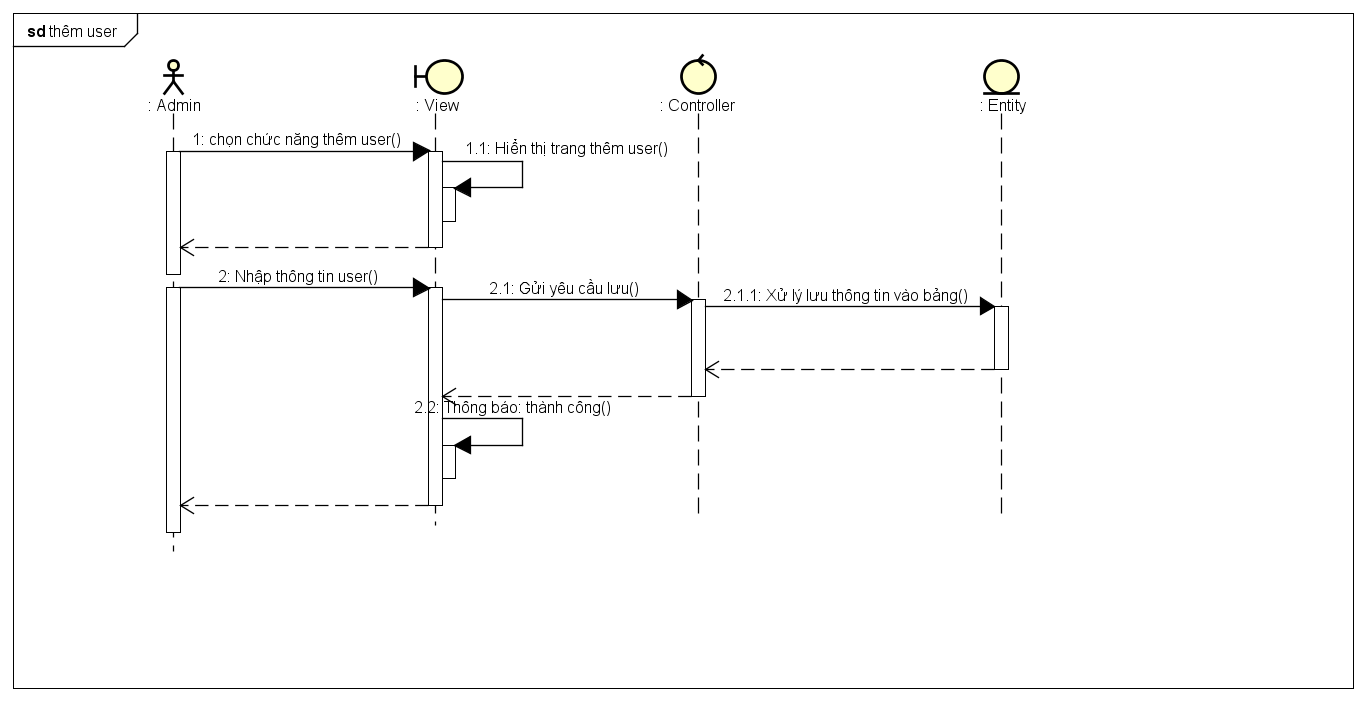
**3.4. Hành vi sửa thông tin**



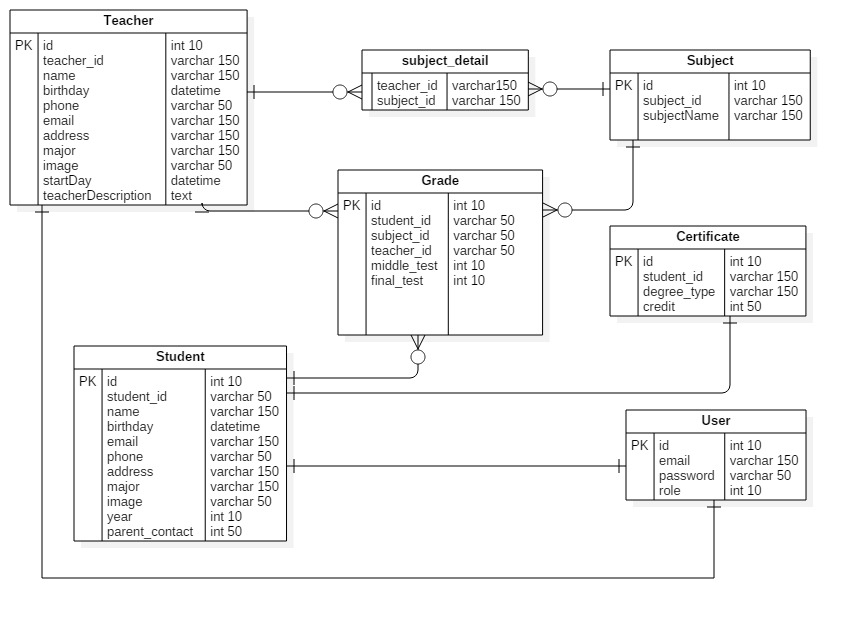
**3.5. Hành vi Thêm điểm**



**3.6. Hành vi Thêm người dùng**

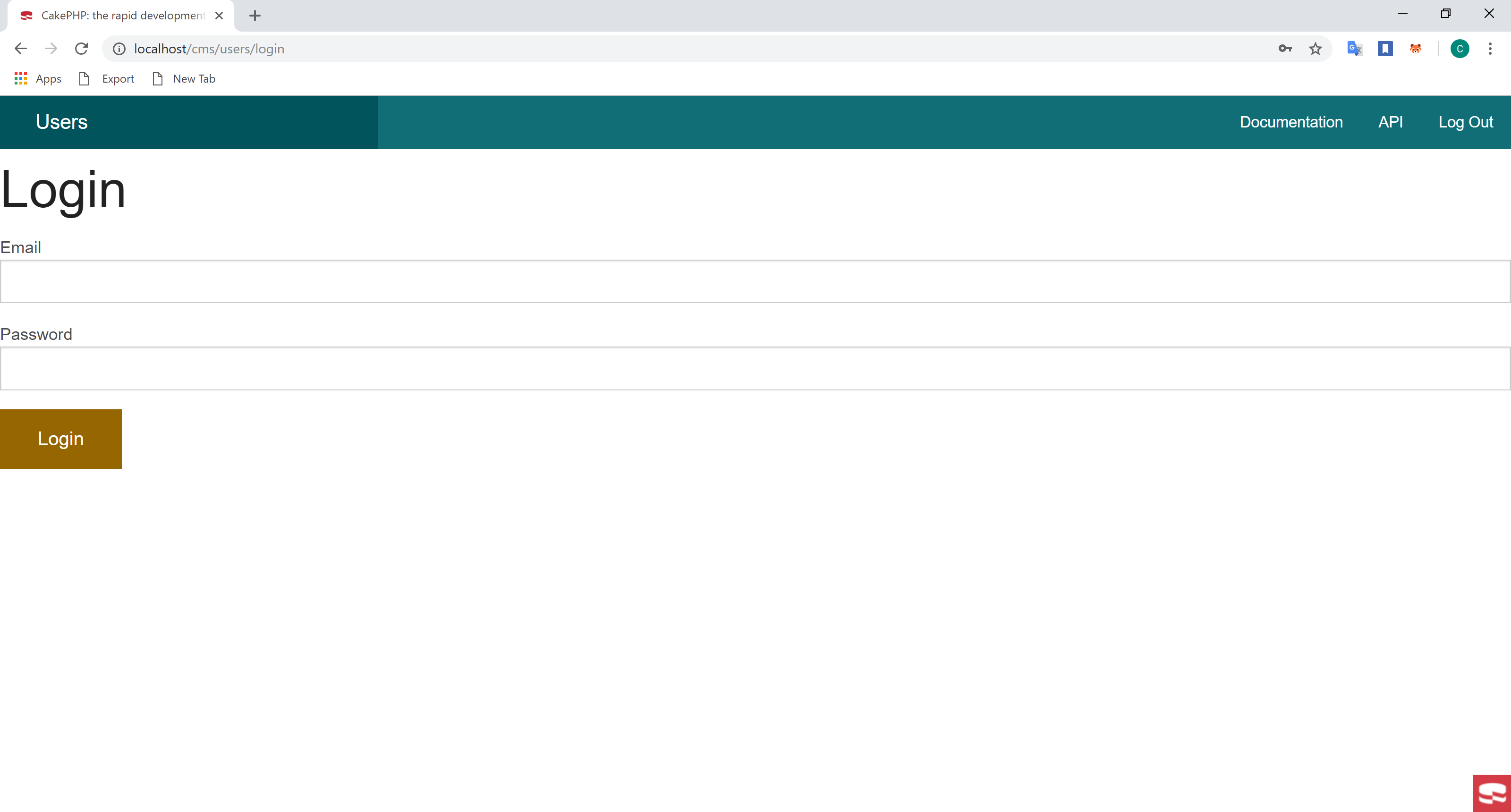


**CHƯƠNG 4: THIẾT KẾ CƠ SỞ DỮ LIỆU**



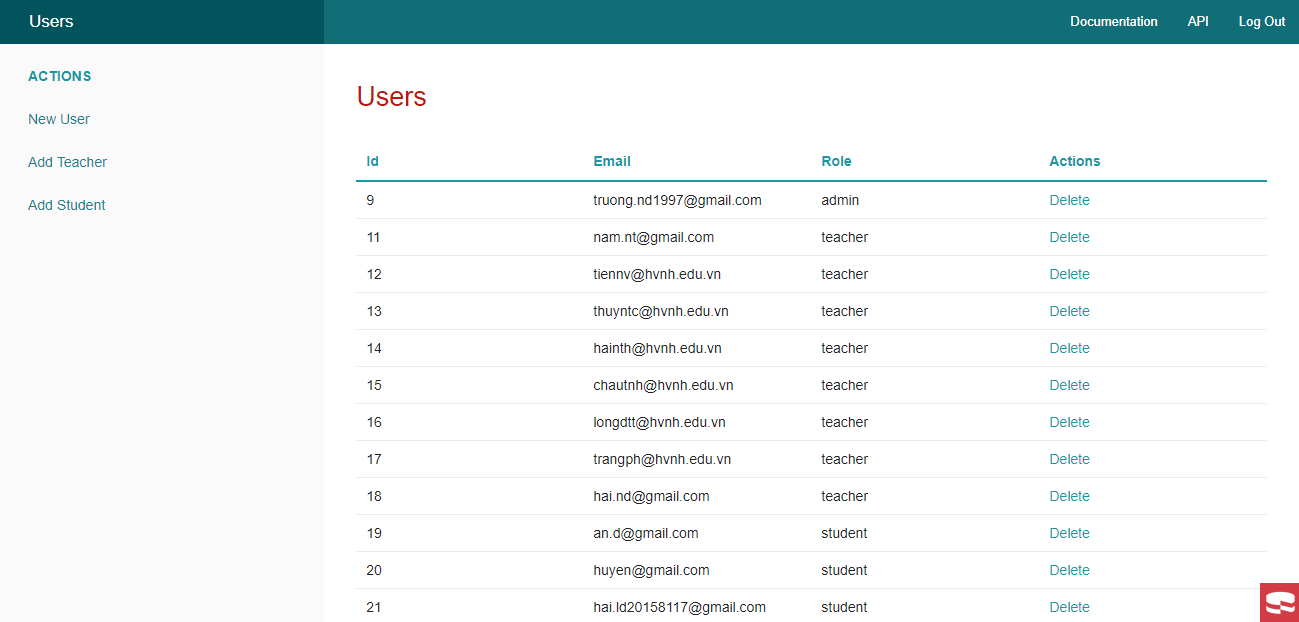
**CHƯƠNG 5: THIẾT KẾ GIAO DIỆN**

**5.1. Giao diện đăng nhập**

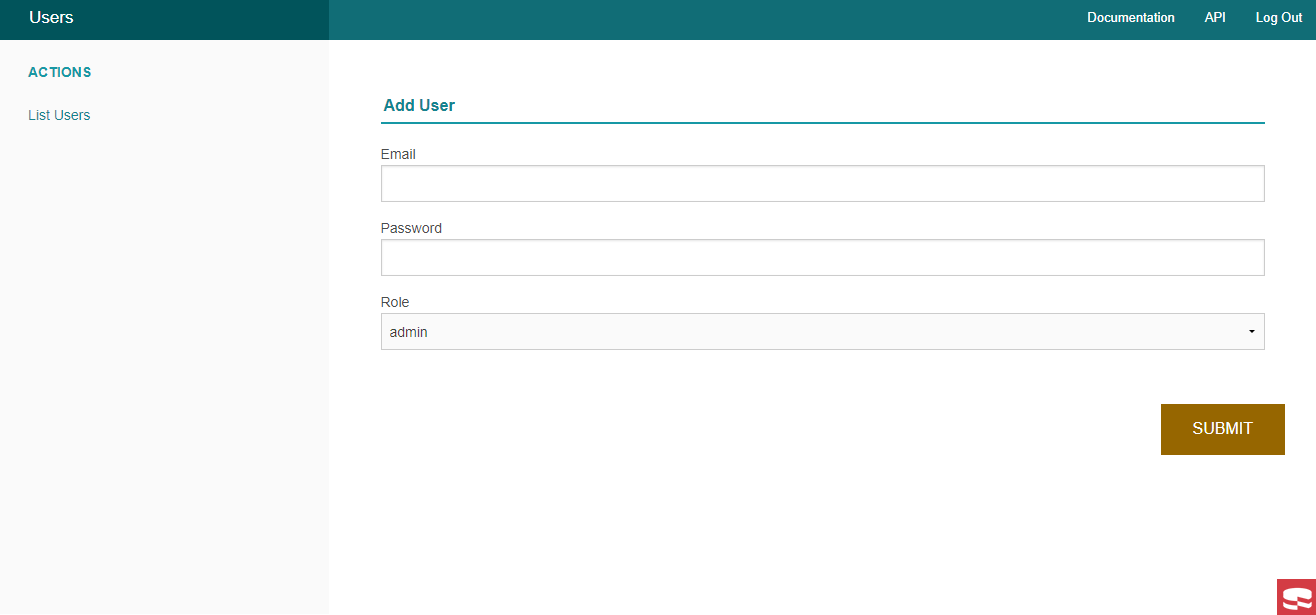


**5.2. Admin**

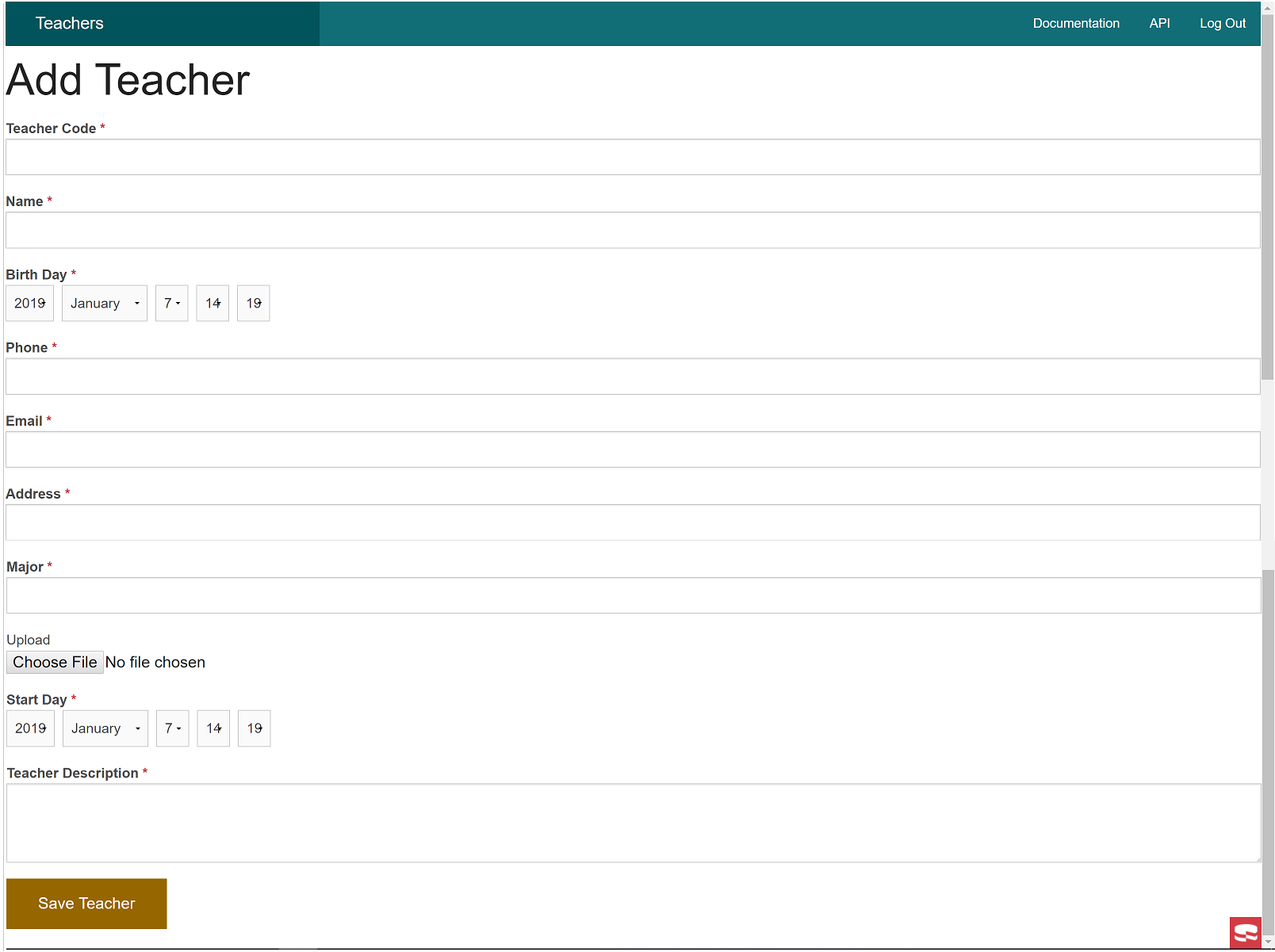
**a, Admin index**



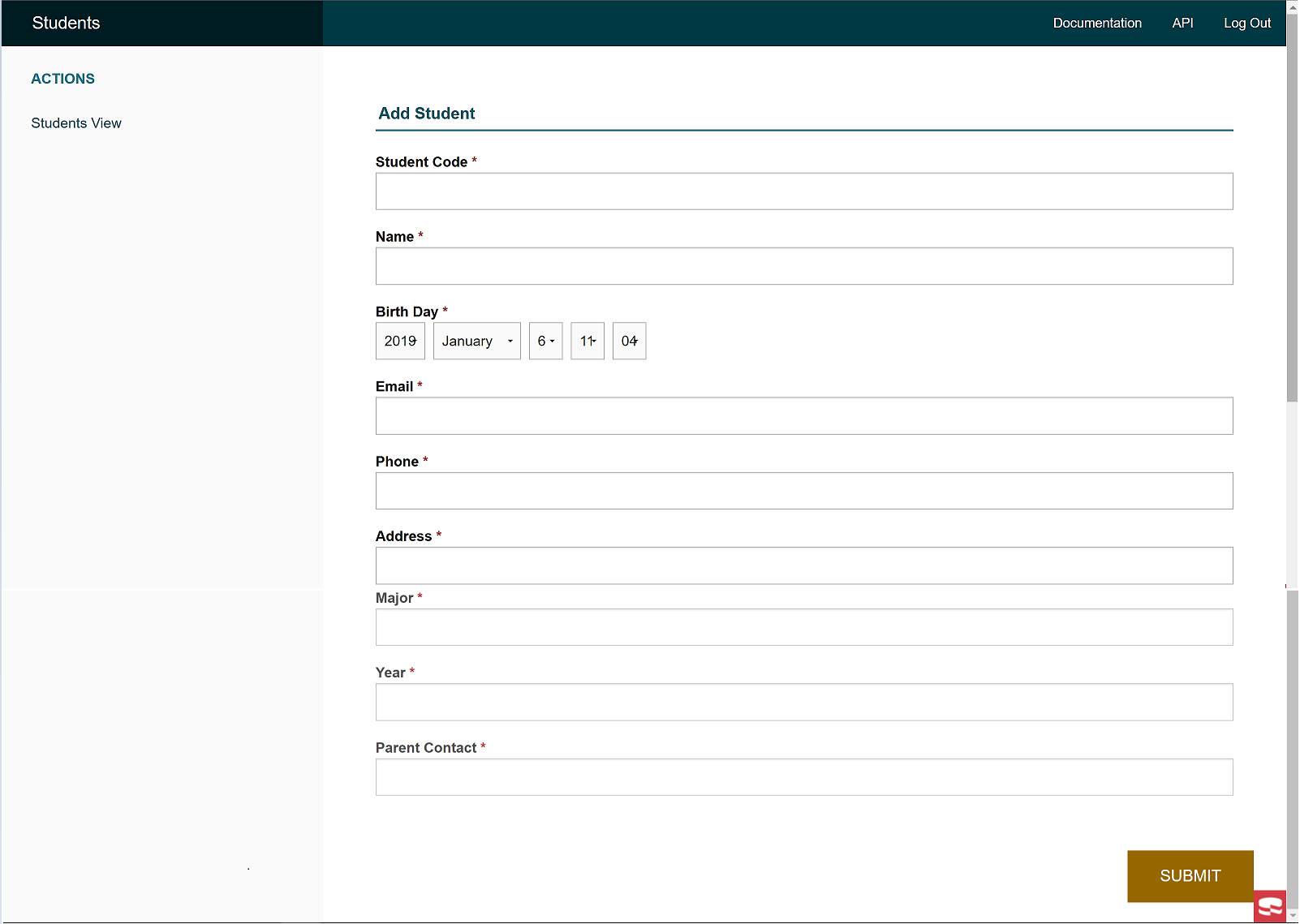
**b, Thêm user**



**c, Thêm teacher**

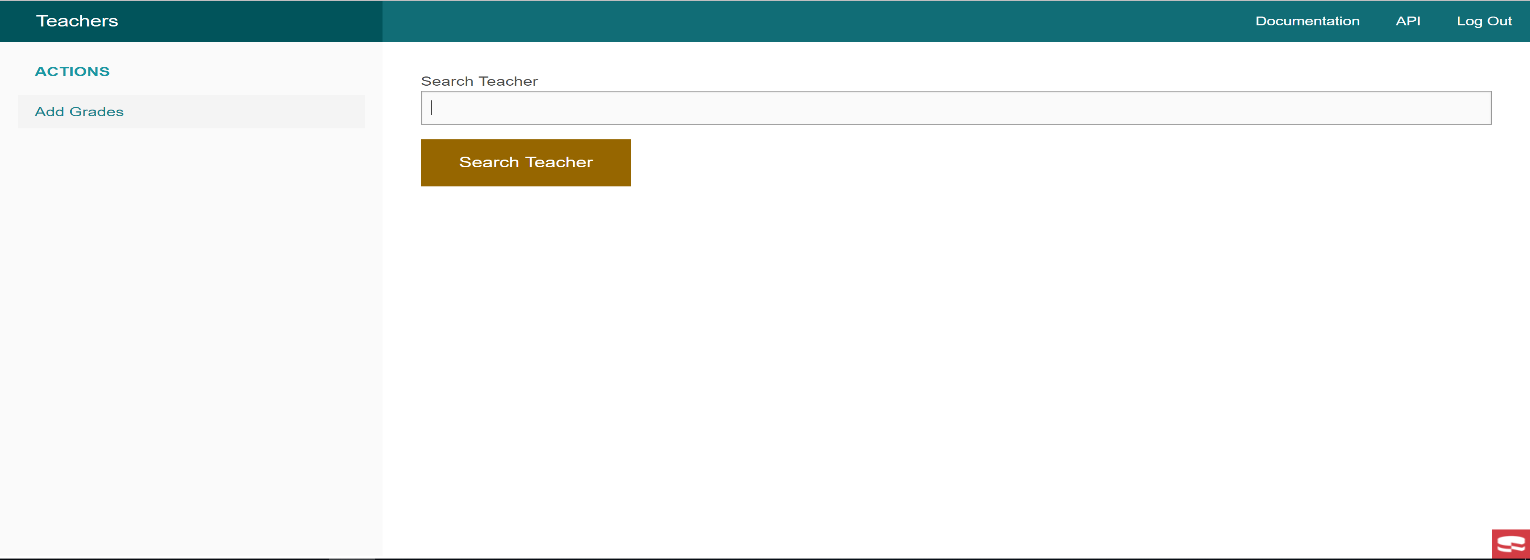


**d, Thêm student**

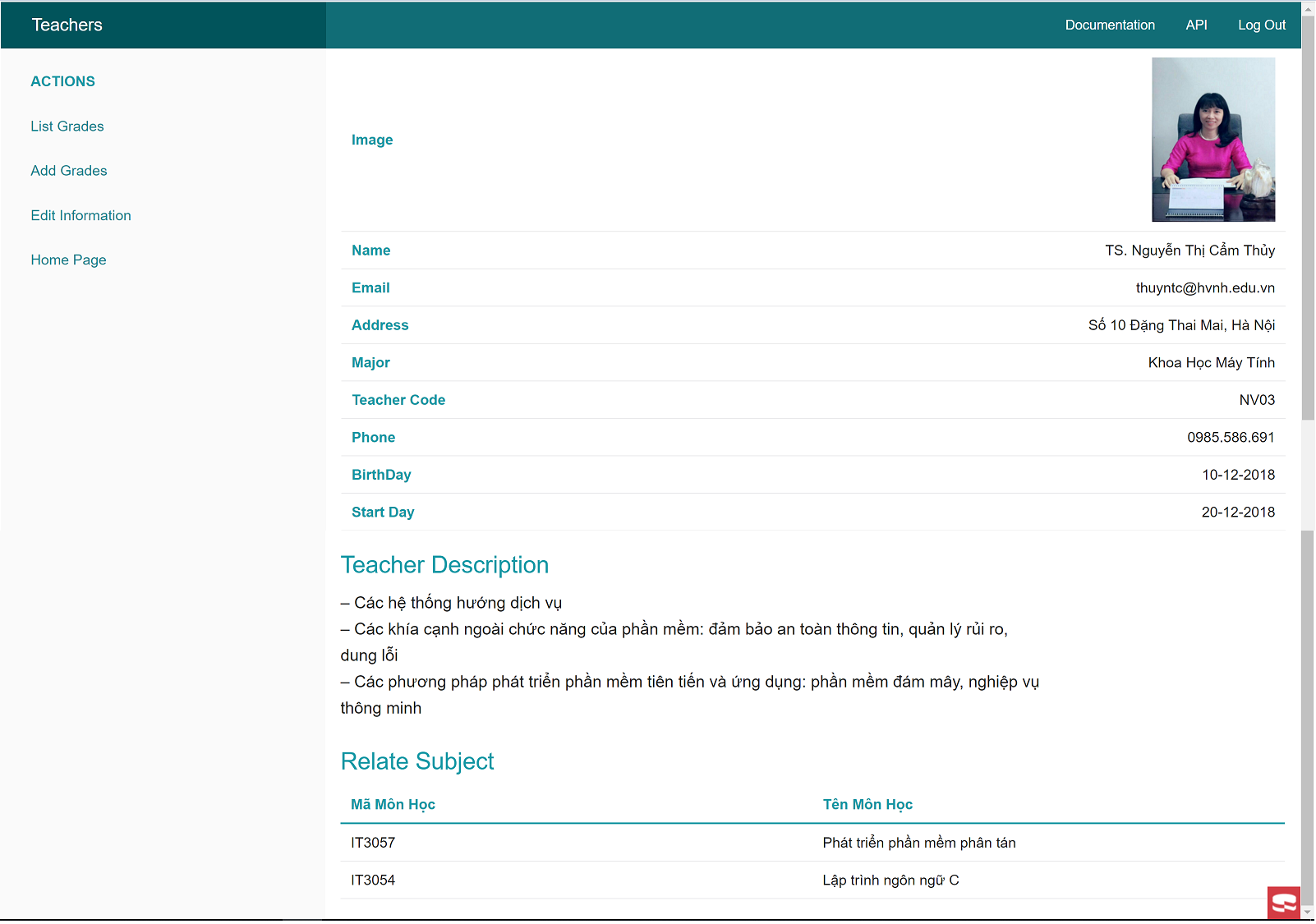


**5.3. Teacher**

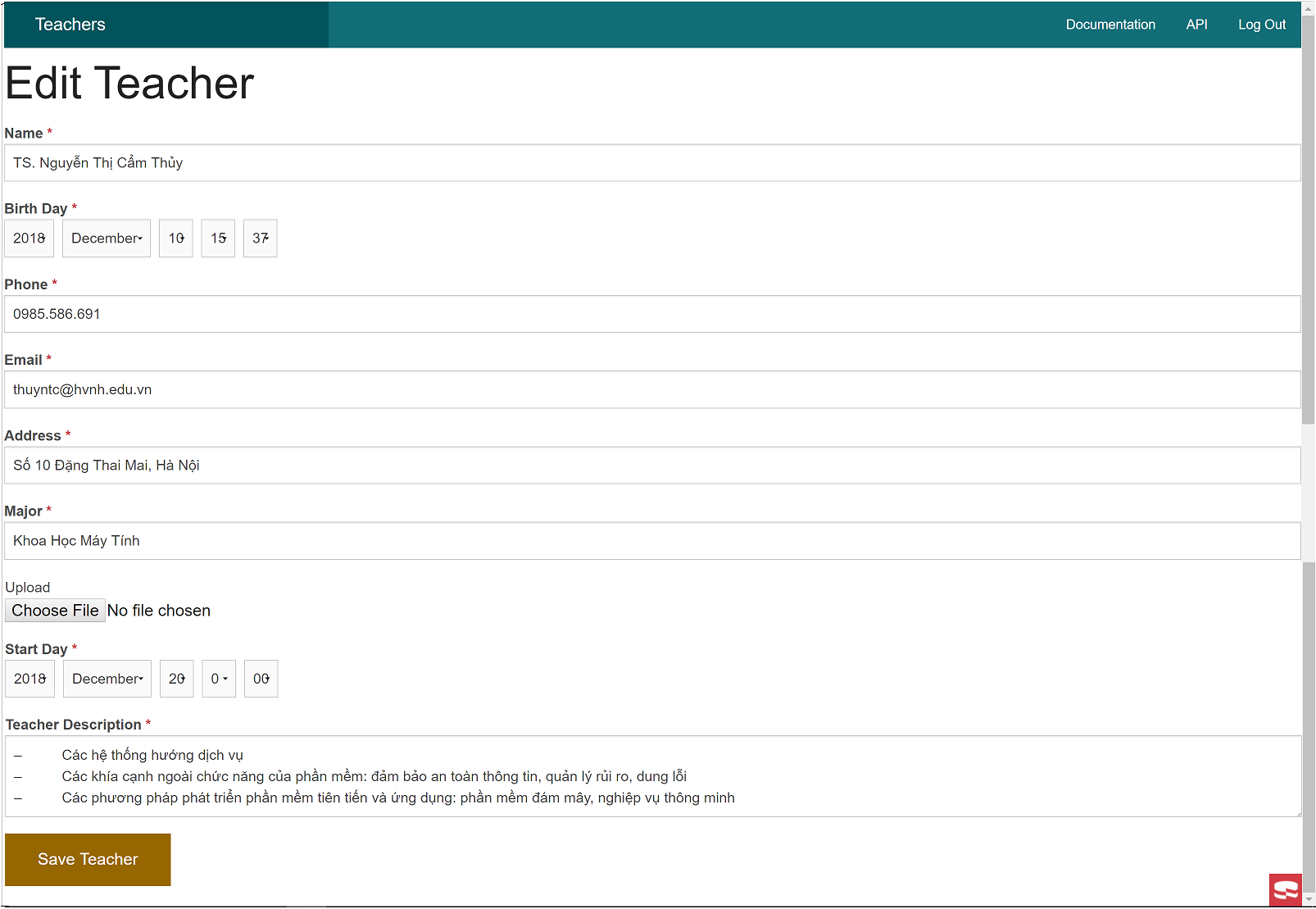
**a, Teacher-Search**



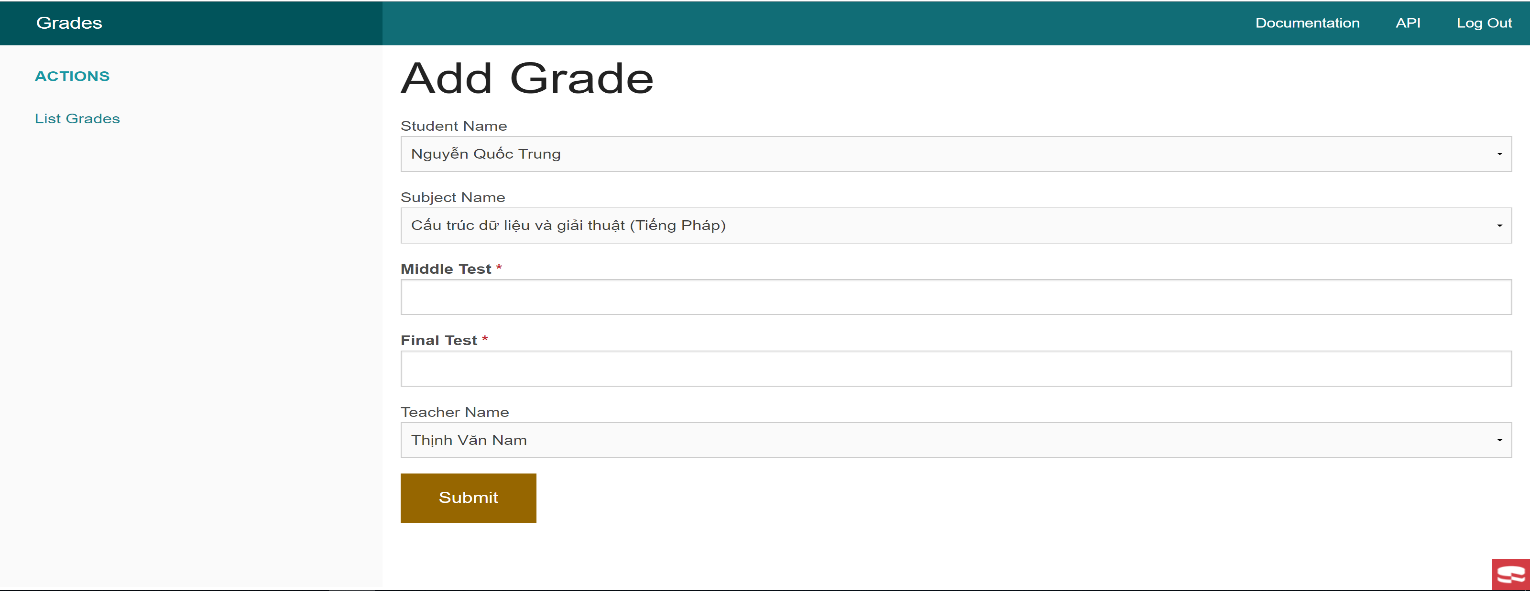
**b, Xem thông tin giáo viên**



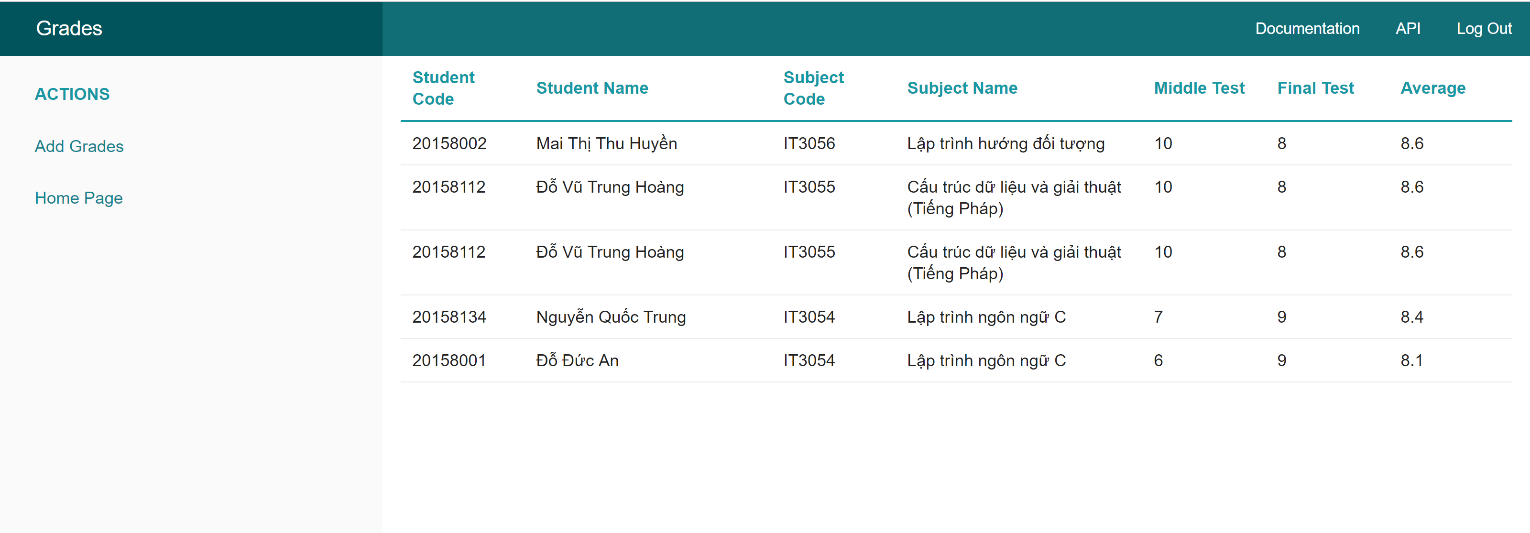
**c, Sửa thông tin giáo viên**



**d, Thêm điểm**

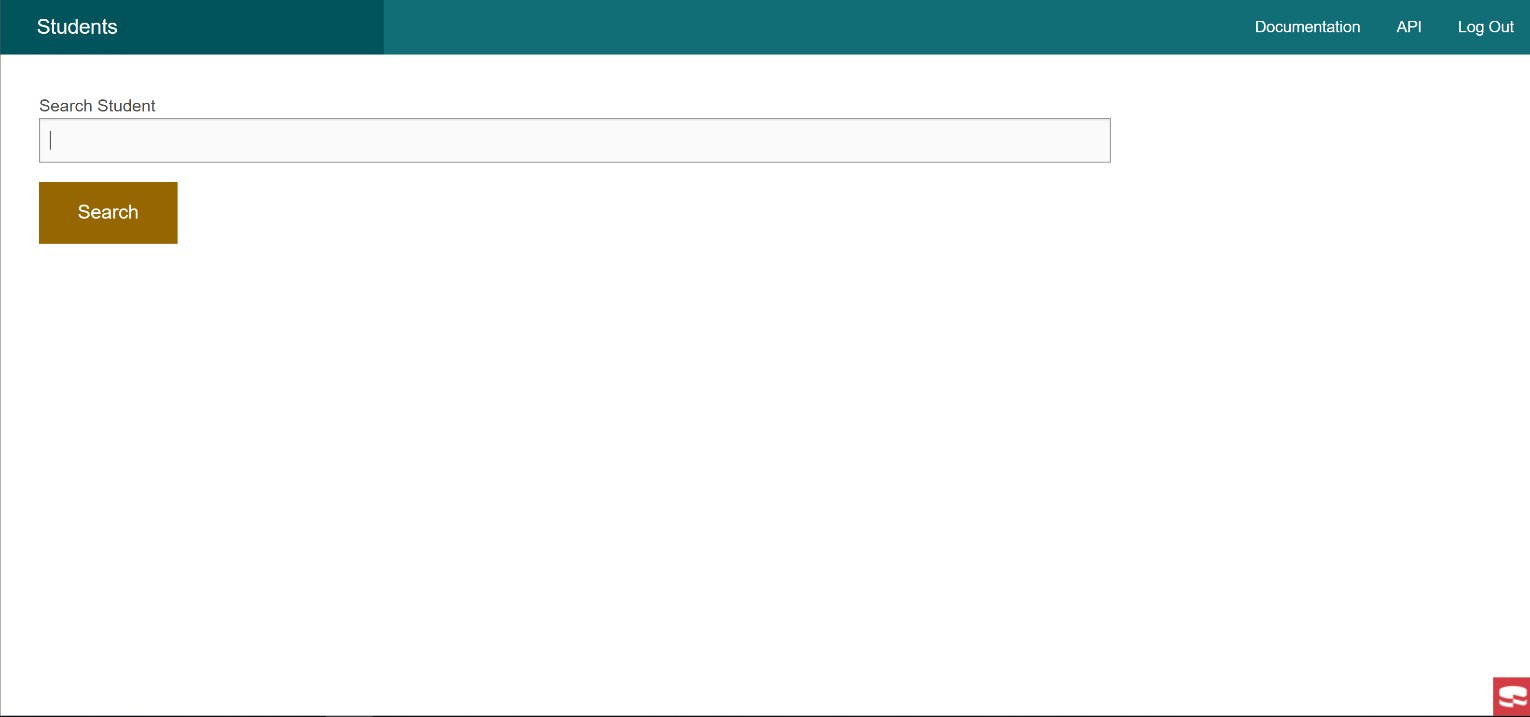


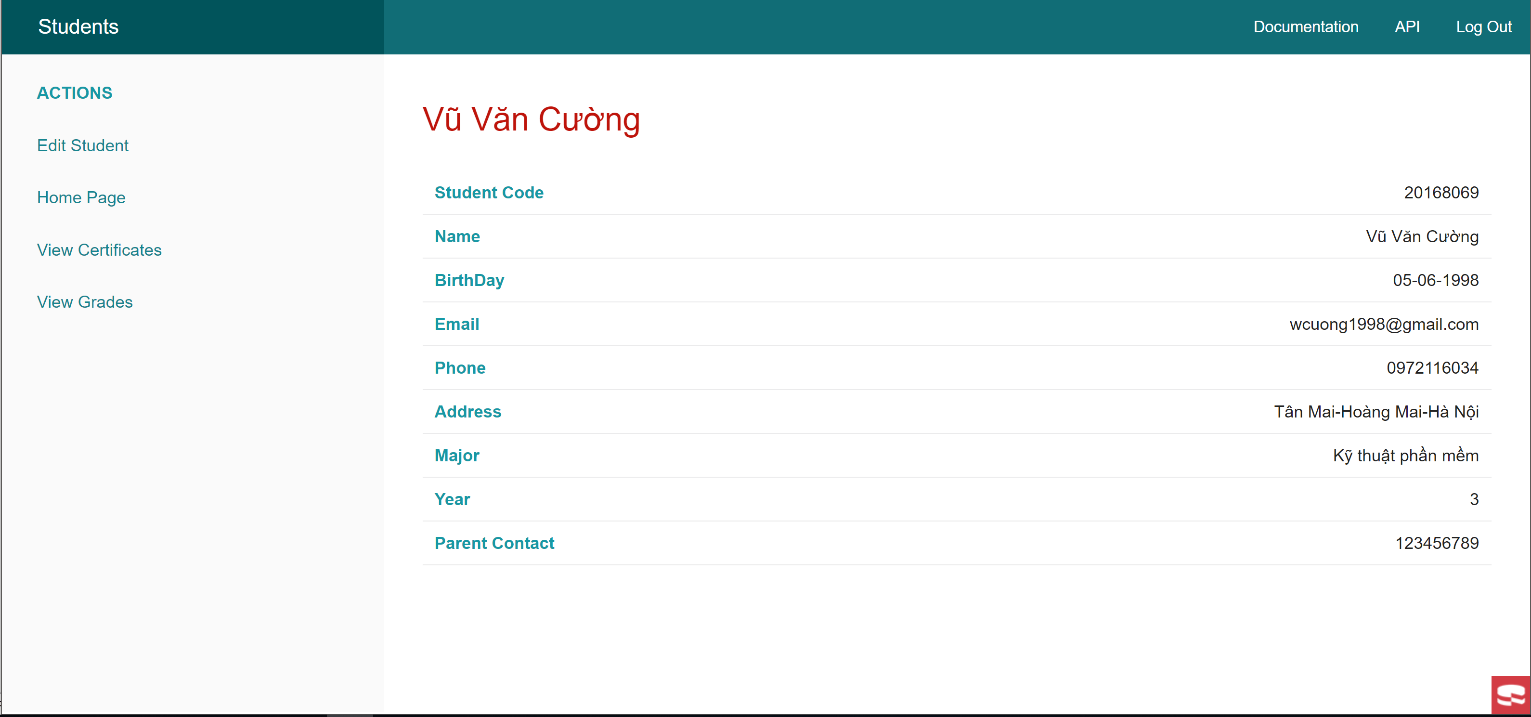
**e, Danh sách điểm**



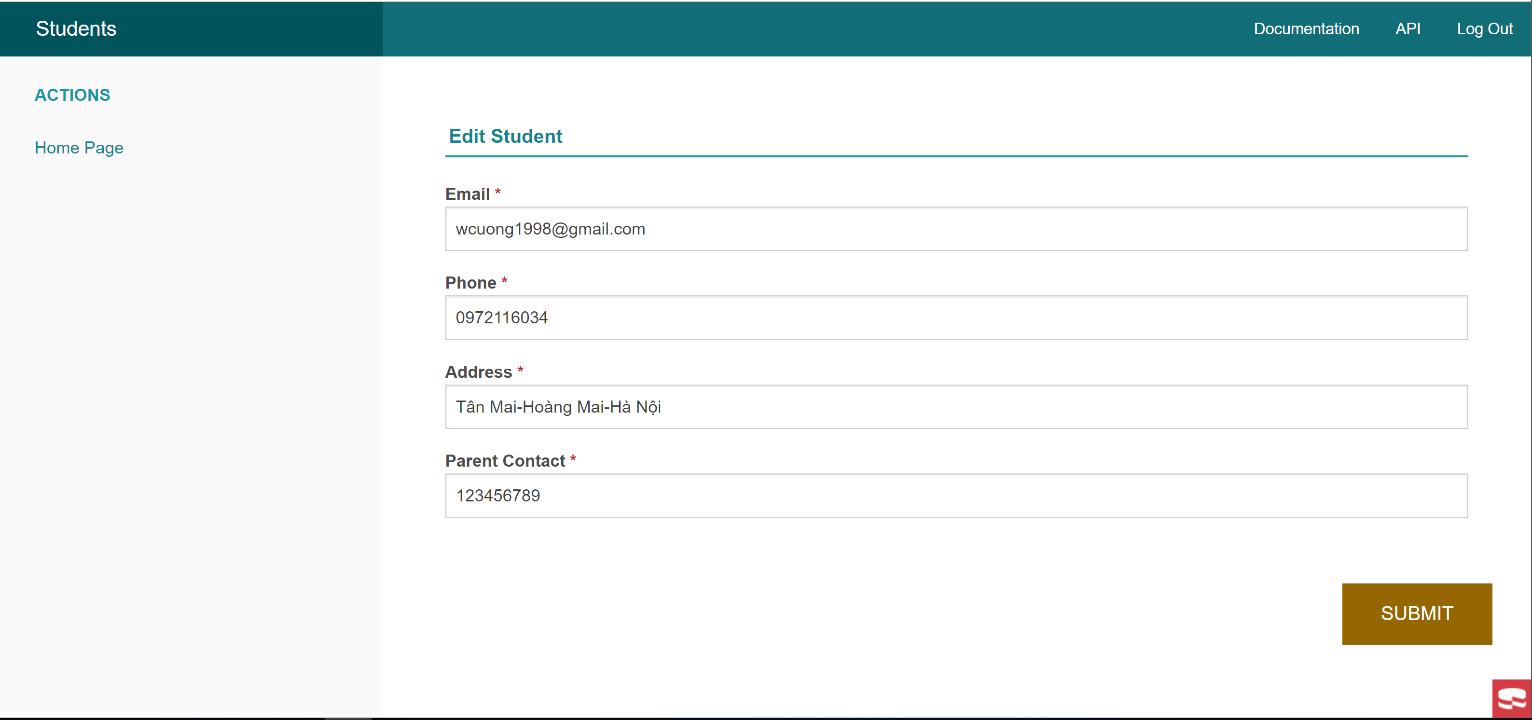
**5.4. Student**

**a, Search Student**

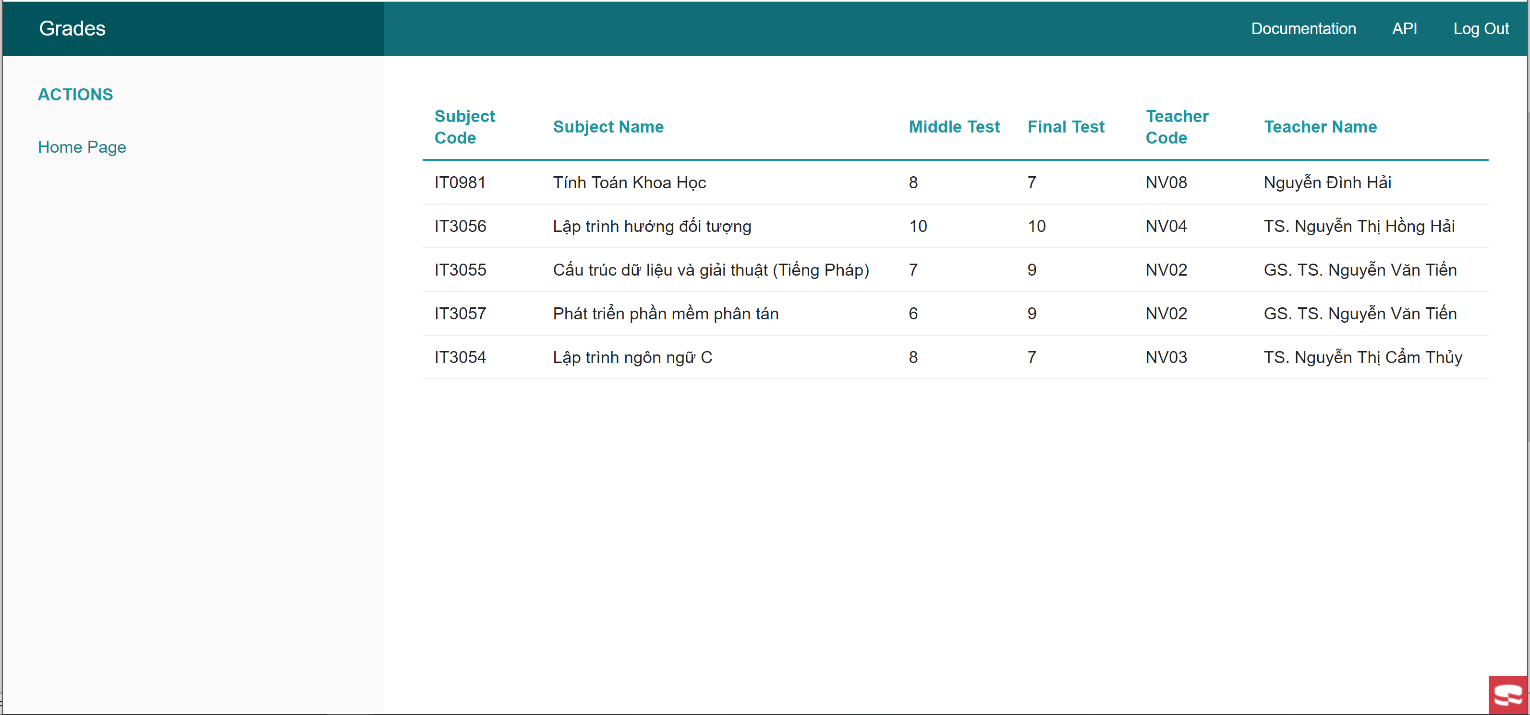


**b, Xem thông tin sinh viên**

**c, Sửa thông tin sinh viên**



**d, Xem điểm**



**e, Xem bằng**

