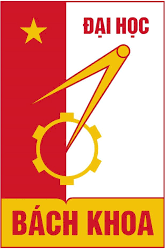
TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



**Software design description**

**Project 2 – IT3930**

**Đề tài: Xây dựng Hệ thống học trực tuyến**

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

**Sinh viên thực hiện: Phạm Văn Hậu - 20173101**

**Hà Nội, Tháng 6 năm 2020**

# **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 các khóa học, thông tin tài khoản sử dụng.
* 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**

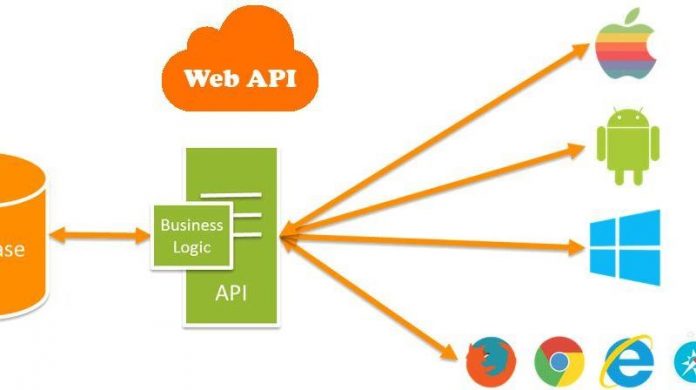
Bản thiết kế này chia làm 5 phần chính:

* 1. Mô hình hóa cấu trúc hệ thống: Cấu trúc hệ thống tổng quan
  2. Thiết kế lớp chi tiết: các lớp tham gia hệ thống
  3. Mô hình hóa hành vi hệ thống: Trình tự hoạt động
  4. Thiết kế cơ sở dữ liệu
  5. 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 theo REST API



* Tổng quan mô hình REST API
* **REST (REpresentational State Transfer)** được đưa ra vào năm 2000, trong luận văn tiến sĩ của Roy Thomas Fielding (đồng sáng lập giao thức HTTP). Nó là một dạng chuyển đổi cấu trúc dữ liệu, là một phong cách kiến trúc cho việc thiết kế các ứng dụng có kết nối. Nó sử dụng HTTP đơn giản để tạo cho giao tiếp giữa các máy. Vì vậy, thay vì sử dụng một URL cho việc xử lý một số thông tin người dùng, REST gửi một yêu cầu HTTP như GET, POST, DELETE, vv đến một URL để xử lý dữ liệu. Trường hợp này sẽ sử dụng GET
* **API** (Application Programming Interface) là giao diện lập trình ứng dụng giúp tạo ra các phương thức kết nối với các thư viện và ứng dụng khác nhau.
* **REST API** là một ứng dụng chuyển đổi cấu trúc dữ liệu có các phương thức để kết nối với các thư viện và ứng dụng khác. REST API không được xem là một công nghệ, nó là một giải pháp để tạo ra các ứng dụng web services thay thế cho các kiểu khác như SOAP, WSDL (Web Service Definition Language),…

2.1.2. 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(được cấu hình trong file appsettings.json ).
    - **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.

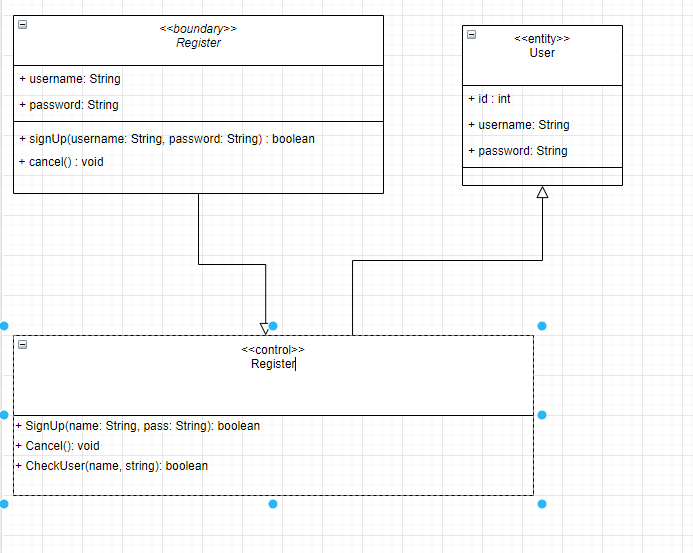
2.1.3. **Mô tả kịch bản hệ thống hướng theo mô hình MVC**

* Kịch bản đăng nhập: User gửi yêu cầu đăng nhập đến Login(LogInController), LogInController tiếp nhận request, 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 đăng kí: User gửi yêu cầu đăng kí tài khoản mới đến Register( LogInController), LogInController tiếp nhận request, LogInController lấy kết quả xác định tồn tại user hay không từ lớp Model(User), nếu chưa tồn tại, cho phép đăng kí, nếu tồn tại không cho phép đăng kí, sau đó, controller này gọi đến 1 lớp view trong package View.
* Kịch bản đăng xuất: Khi User đã đăng nhập vào trong hệ thống, User gửi yêu cầu đăng xuất đến Logout(LogInController), LogInController tiếp nhận request, đồng thời xóa thông tin phiên người dùng hiện tại, chuyển đến Trang chủ(View) khi đăng xuất thành công
* Kịch bản Đăng ký khóa học: User gửi yêu cầu đăng ký khóa học đến AddCourse(HomeController), controller này tiếp nhận request, rồi thêm khóa học vào danh sách các khóa học của User, sau đó, controller này gọi đến 1 lớp view trong package View, và hiển thị thông tin các khóa học của User.
* Kịch bản Sử dụng khóa học: Khi User vào xem danh sách các khóa học của mình đã đăng ký, khi nhấn vào button “Vào học” sẽ được gọi đến Study(Controller), controller này tiếp nhận request, rồi lấy thông tin của khóa học này, sau đó , controller này gọi đến 1 lớp view trong package View, và hiển thị danh sách bài giảng của khóa học này
* Kịch bản Thống kê khóa học theo thể loại: Sau khi User click vào thể loại khóa học muốn xem sẽ được gửi đến GetCourseByCategory(HomeController), controller này tiếp nhận request, rồi lấy danh sách các khóa học theo thể loại tương ứng, trả về data dạng Json. Sau đó, gọi Ajax để lấy dữ liệu và hiển thị ra màn hình

**2.2**. Các lớp tham gia ca sử dụng.

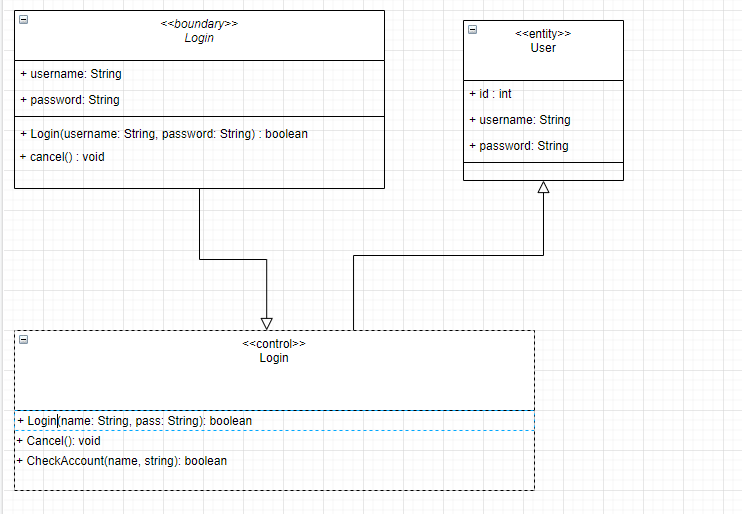
Các lớp tham gia ca sử dụng gồm 3 loại:  
❖ Boundaries (các lớp biên hay các lớp đối thoại): tương ứng với tầng View  
❖ Entities (các lớp thực thể hay các lớp lĩnh vực): tương ứng với tầng Model  
❖ Controls (các lớp điều khiển): tương ứng với tầng Controller

**2.2.1**. Ca sử dụng Đăng kí:



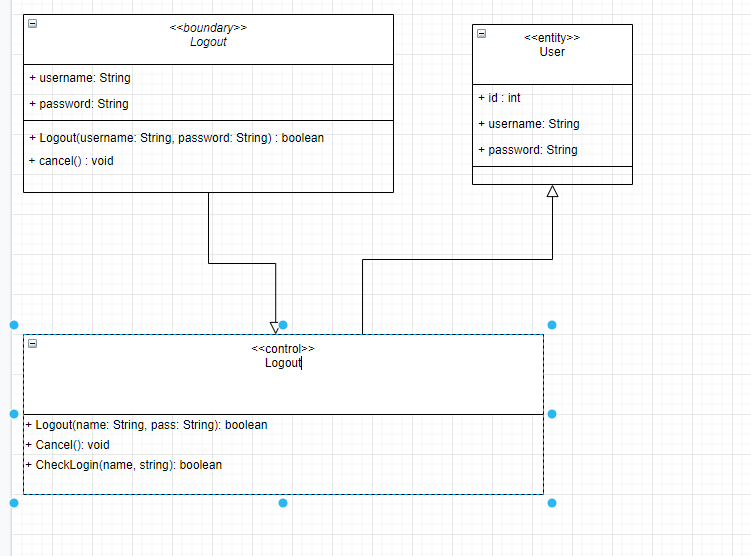
*Hình 2: Biểu đồ lớp tham gia ca sử dụng Đăng kí*

**2.2.2**. Ca sử dụng Đăng nhập



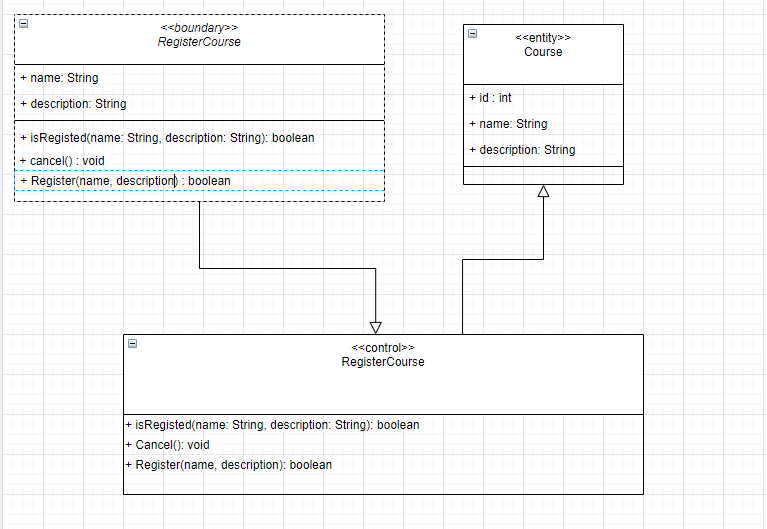
*Hình 3: Biểu đồ lớp tham gia ca sử dụng Đăng nhập*

**2.2.3**. Ca sử dụng Đăng xuất



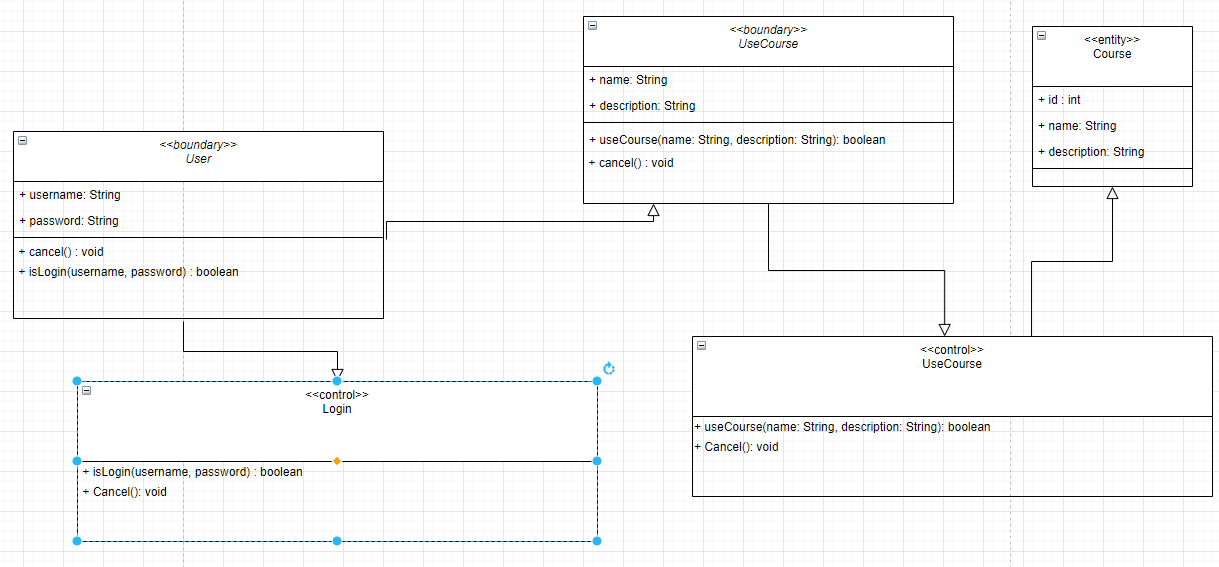
*Hình 4: Biểu đồ lớp tham gia ca sử dụng Đăng xuất*

**2.2.4**. Ca sử dụng Đăng ký khóa học



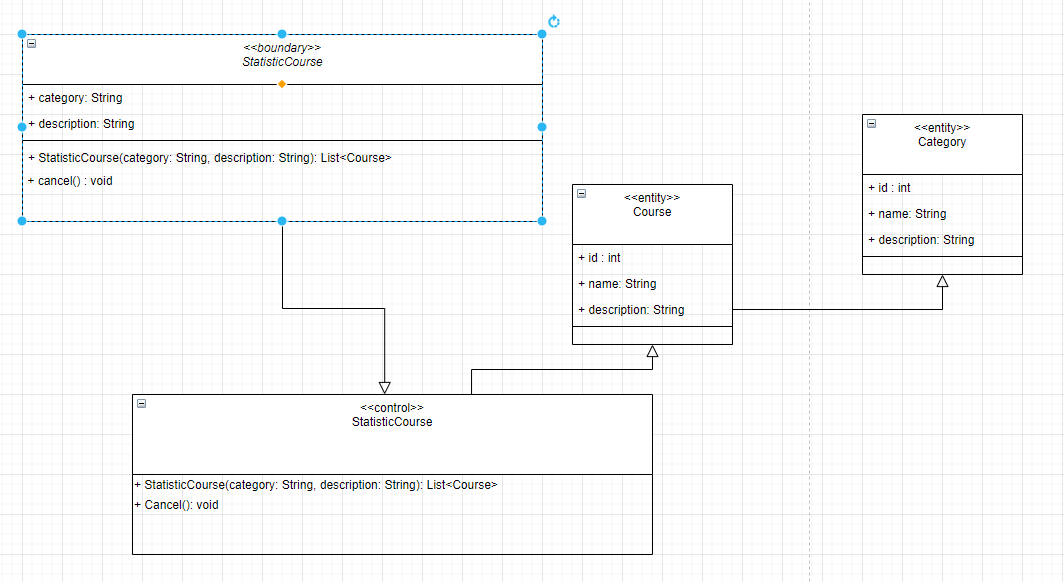
*Hình 5: Biểu đồ lớp tham gia ca sử dụng Đăng ký khóa học*

**2.2.5**. Ca sử dụng Sử dụng khóa học



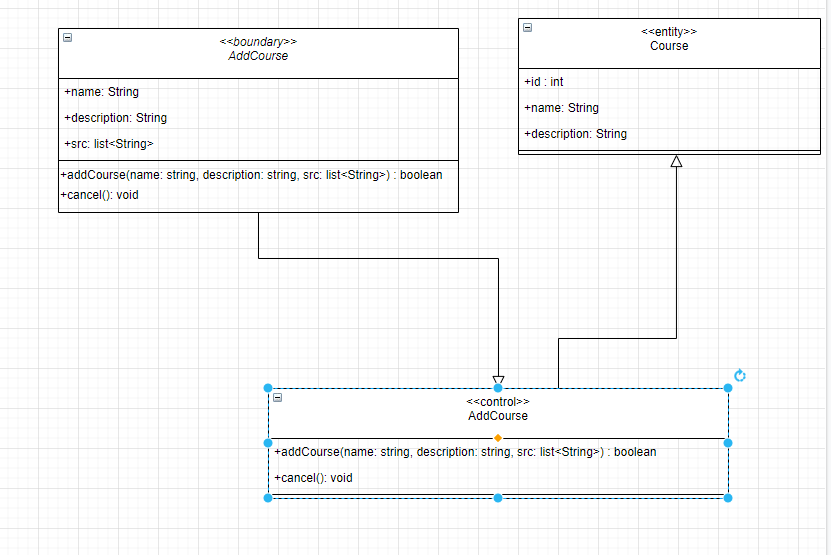
*Hình 6: Biểu đồ lớp tham gia ca sử dụng Sử dụng khóa học*

**2.2.6**. Ca sử dụng Thống kê khóa học



*Hình 7: Biểu đồ lớp tham gia ca sử dụng Thống kê khóa học*

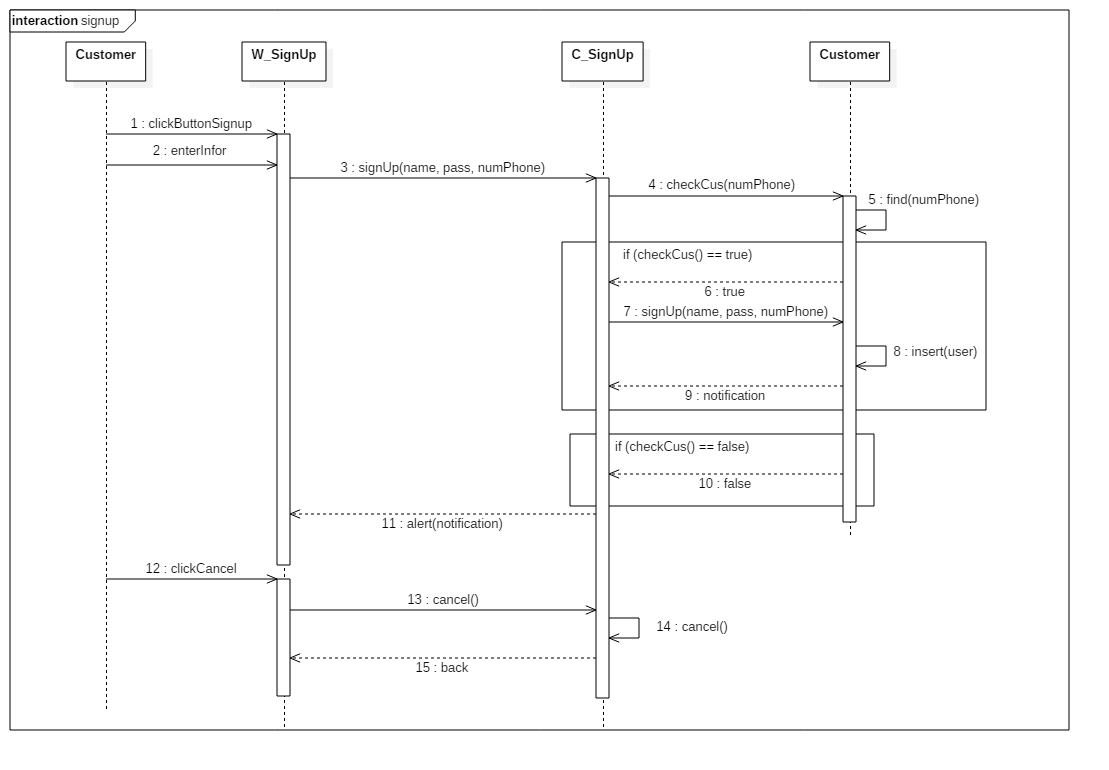
**2.2.7**. Ca sử dụng Thêm khóa học



*Hình 8: Biểu đồ lớp tham gia ca sử dụng Thêm khóa học*

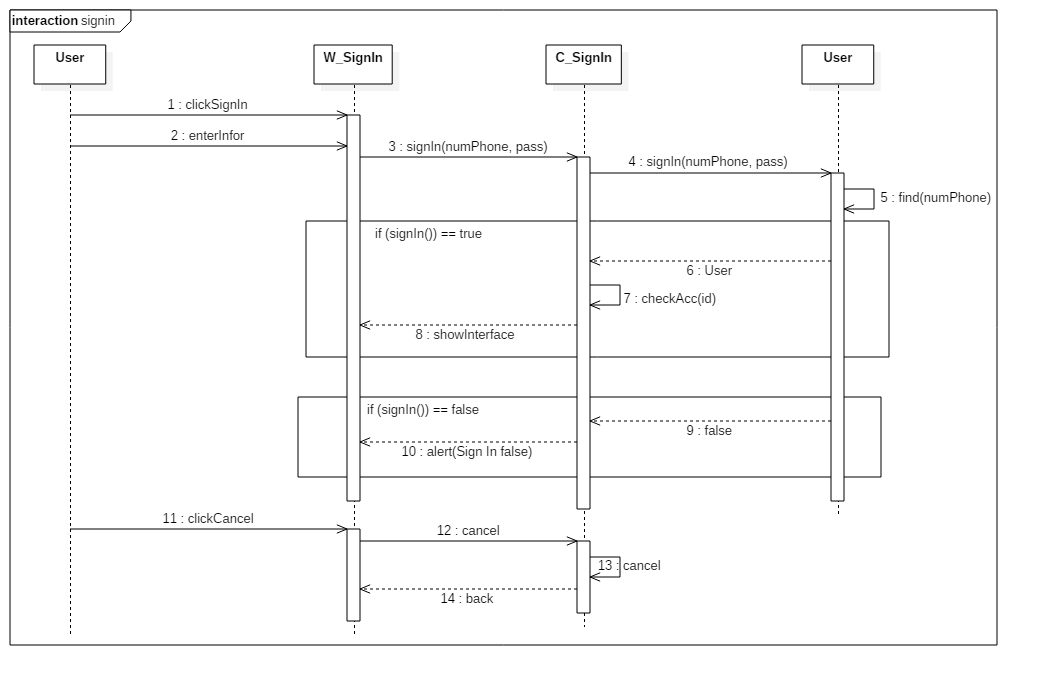
**CHƯƠNG 3: MÔ HÌNH HOÁ HÀNH VI HỆ THỐNG**

**3.1. Hành vi Đăng kí**

****

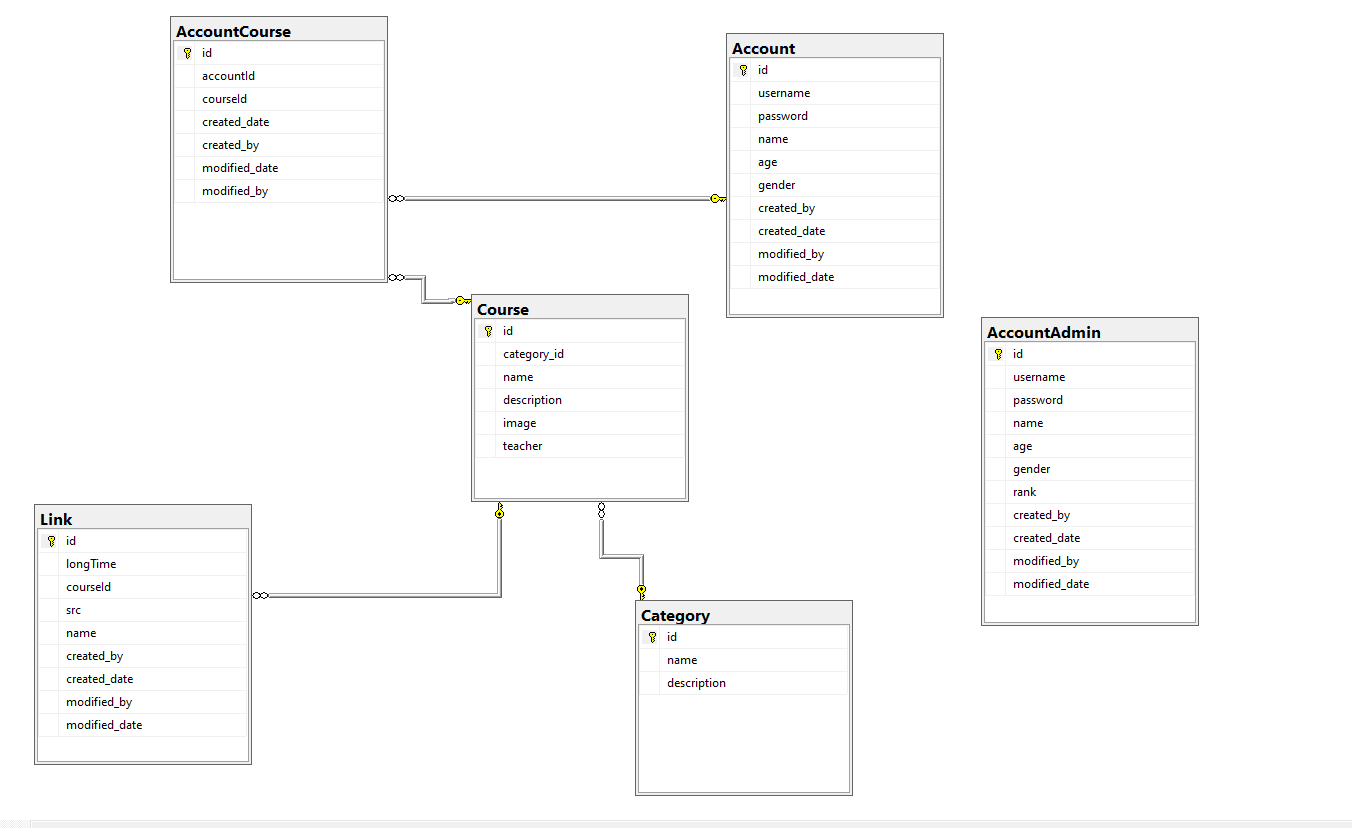
*Hình 8: Biểu đồ Hành vi Đăng kí*

**3.2. Hành vi Đăng nhập**

****

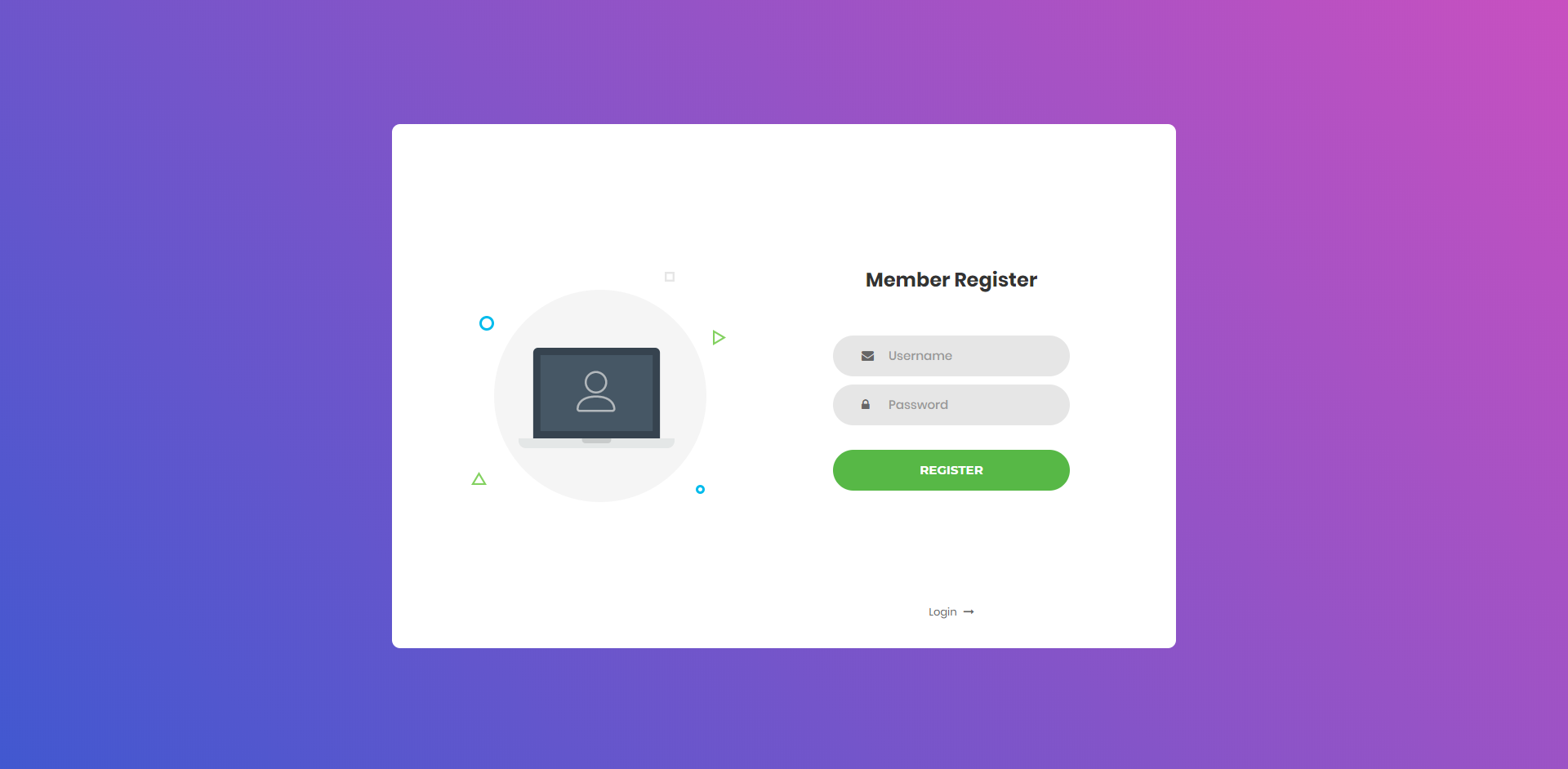
*Hình 9: Biểu đồ Hành vi Đăng nhập*

**CHƯƠNG 4: THIẾT KẾ CƠ SỞ DỮ LIỆU**

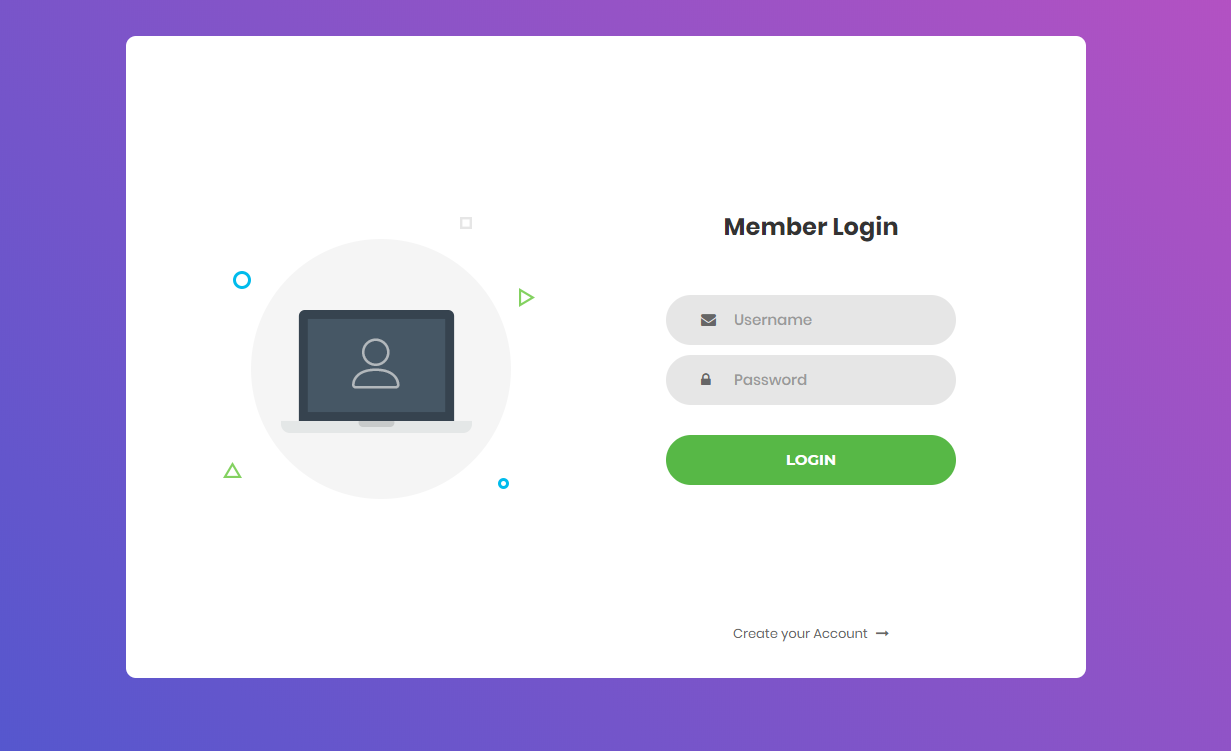


**CHƯƠNG 5: THIẾT KẾ GIAO DIỆN**

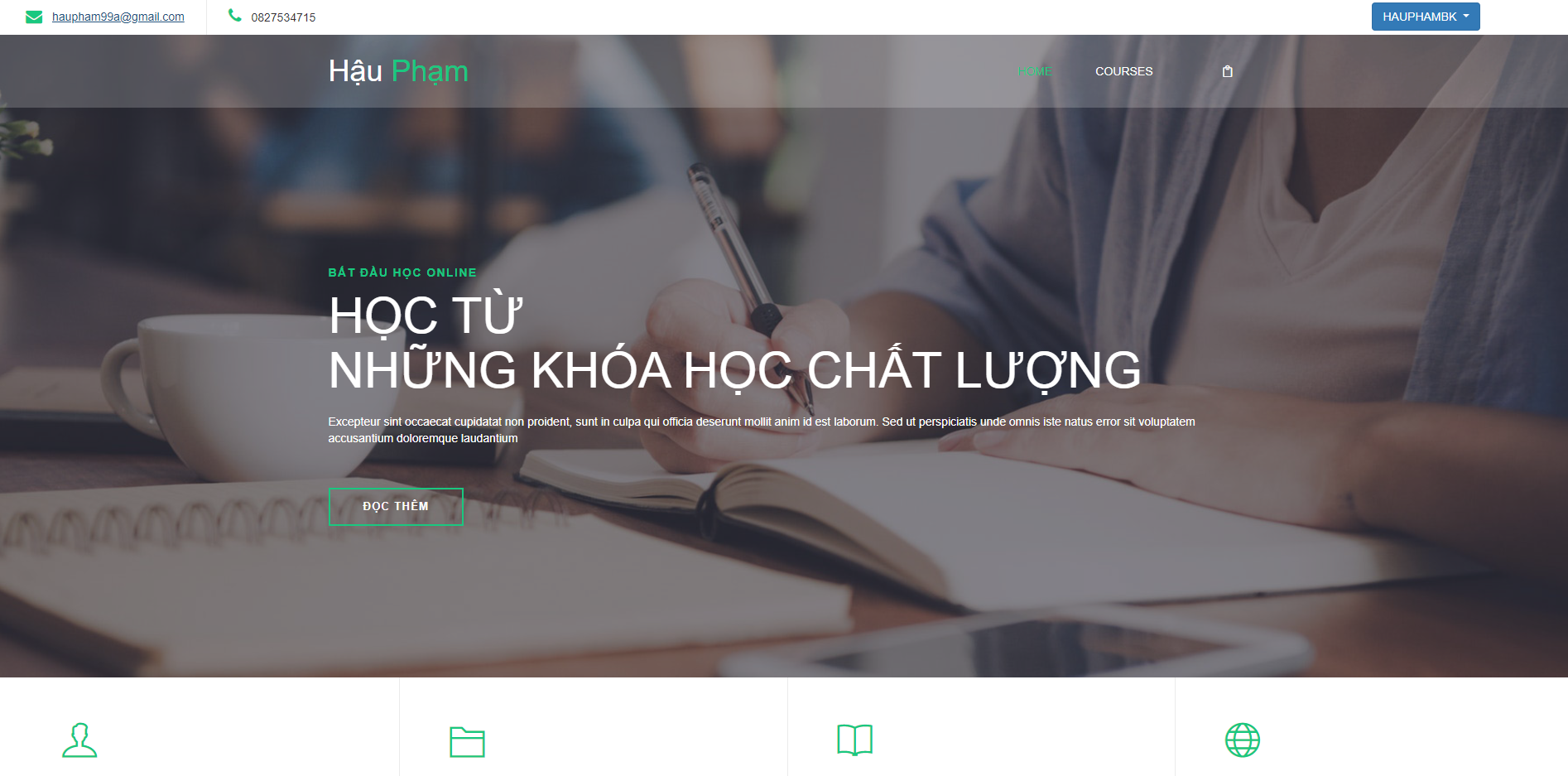
**5.1. Giao diện đăng kí**



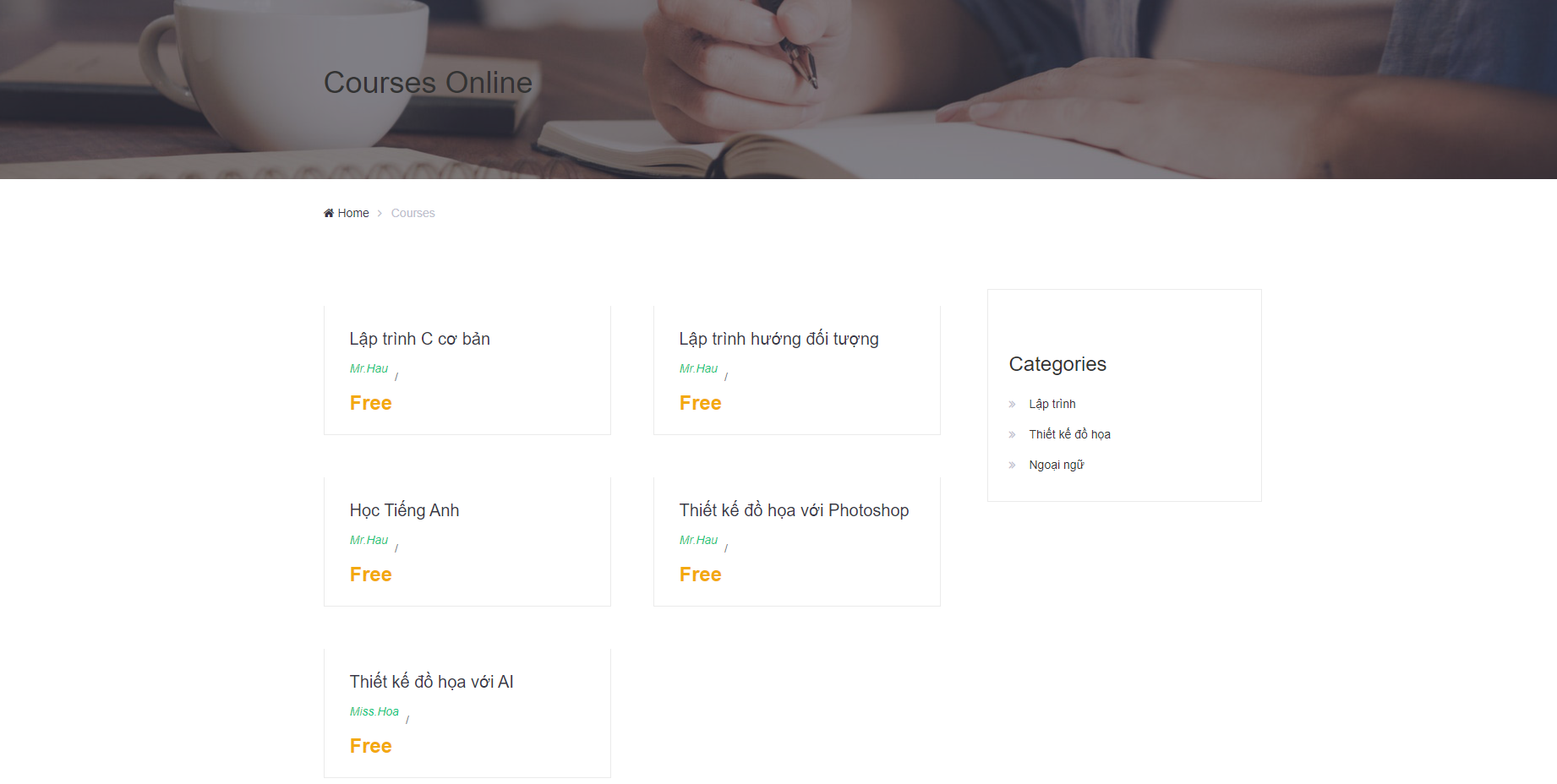
**5.2. Giao diện đăng nhập**



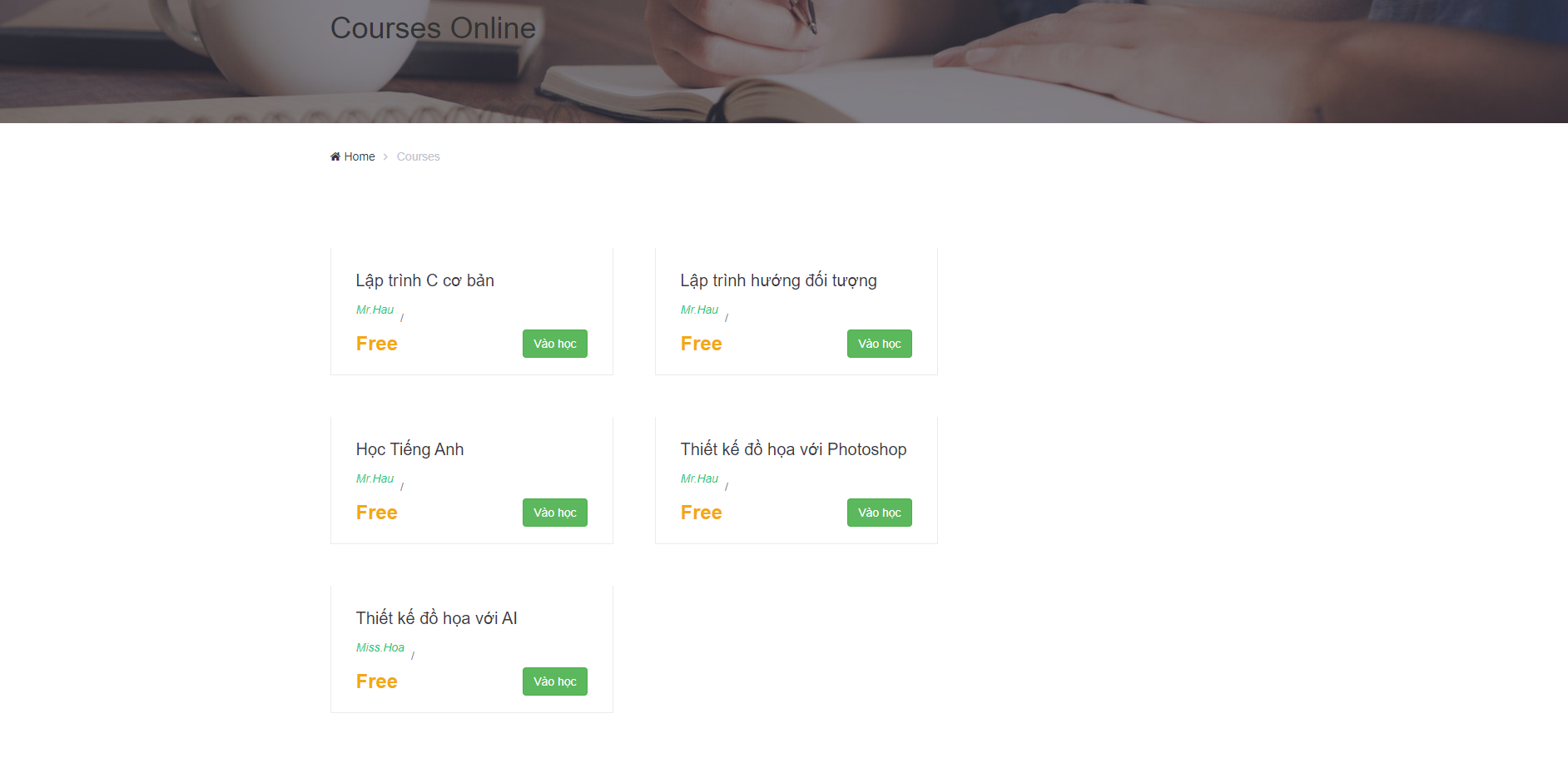
**5.3. Giao diện trang chủ**



**5.4. Giao diện Danh sách khóa học**



**5.5. Giao diện khóa học của tôi**



**5.6. Giao diện khóa học**

