Website đăng kí học phần

Software Architecture Document

Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 26/11/2022 | 1.0 | Hoàn thành các mục Introduction, Architectural Goals and Constraints, Use Case Model, Logical View, Class Diagram, Components Description | Group 12 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 4

2. Architectural Goals and Constraints 4

3. Use-Case Model 4

4. Logical View 4

4.1 Component: abc 4

5. Deployment 4

6. Implementation View 4

Software Architecture Document

# Introduction

## Purpose

Tài liệu Software Architecture Document đưa ra một cái nhìn khái quát và tổng thể về các kiến trúc được sử dụng trong hệ thống “Website đăng ký học phần” mà nhóm sẽ phát triển. Sử dụng một số các mô tả kiến trúc khác nhau để cho thấy được các phương diện về dữ liệu, cách thức hệ thống liên kết các phương diện với nhau, v.v…. và để đánh dấu những chọn lựa có ý nghĩa và tầm vóc quan trọng về các kiến trúc của hệ thống.

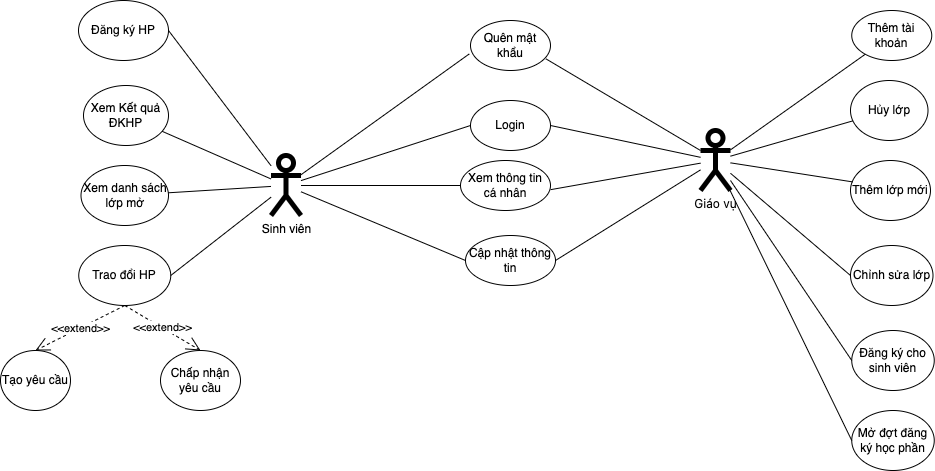
## Scope

Nhóm 12 thực hiện dự án “Website đăng ký học phần” nhắm đến việc đạt được các tiêu chí chính mà nhóm đề ra về tính hoàn chỉnh, độ hiệu quả và khả năng đáp ứng nhu cầu người dùng của hệ thống nhóm phát triển tạo ra. Từ những lý do đó, tài liệu này sẽ biểu diễn một số kiến trúc quan trọng được sử dụng trong xây dựng hệ thống “Website đăng ký học phần” để có thể chỉ ra đâu là những lựa chọn thiết kế phù hợp với các tiêu chí và ở mức độ ra sao. Các use cases chính sẽ được biểu diễn thông qua các mô hình thiết kế để trực quan hóa các nội dung có trong thiết kế, dễ hình dung hơn để dựa vào đó xác định các hướng đi tiếp theo trong quá trình phát triển.

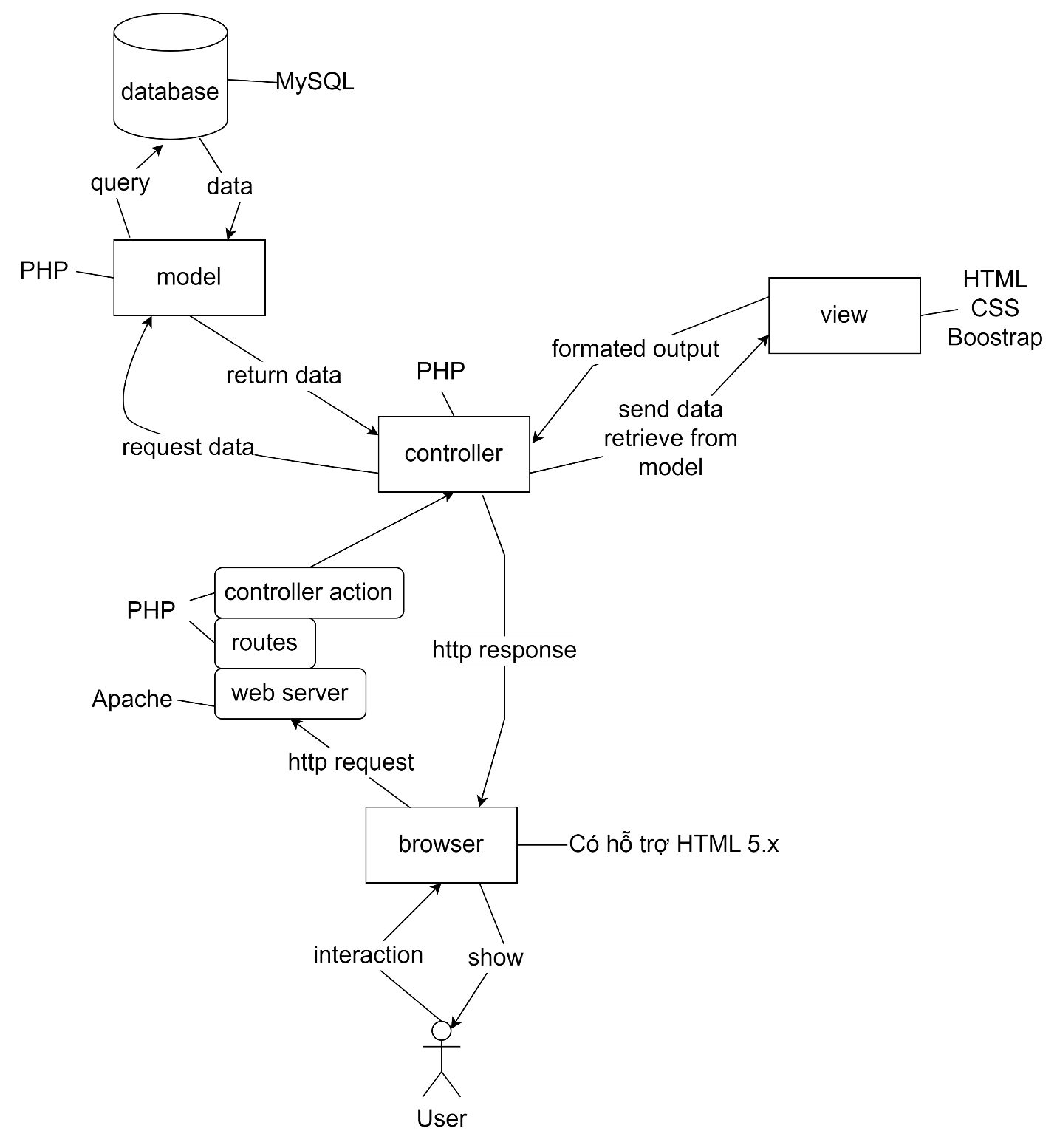
# Architectural Goals and Constraints

* Hệ thống phải đảm bảo quá trình kết nối, truyền tải dữ liệu qua lại giữa các thành phần được mã hóa. Giúp bảo mật thông tin của người sử dụng (sinh viên, giáo vụ). Giao thức https yêu cầu trình duyệt phải mã hóa những dữ liệu được trao đổi với một trang web. Việc mã hóa giúp giấu dữ liệu và giảm bớt nguy cơ thông tin của bạn bị xem hoặc sửa đổi. Điều này rất quan trọng khi trang web có dữ liệu nhạy cảm như thông tin cá nhân hoặc thông tin tài chính của bạn. Cũng như hệ thống cũng phải thực hiện các hành vi bảo mật như đăng nhập.
* Những phần logic quan trọng của hệ thống không được xuất hiện ở phía người dùng, các tài nguyên được tải xuống phía user không bao gồm chi tiết xử lý này. Cấu trúc thư mục lưu trữ mã nguồn phải tuân thủ theo các kiến trúc cụ thể.
* Hệ thống phải điều hướng nội dung đúng với mục đích sử dụng và mong muốn của người dùng đối với từng chức năng cụ thể. Hành vi điều hướng website.
* Hệ CSDL sử dụng phải đảm bảo các ràng buộc về mặt dữ liệu, cũng như hệ thống phải sử dụng đúng thông tin mà người dùng để trình bày đối với từng trang hiển thị cụ thể. Sử dụng hệ cơ sở dữ liệu quan hệ để đảm bảo tính bền vững của dữ liệu.
* Giao diện người dùng trực quan, phù hợp với mục đích, yêu cầu của hệ thống.

# Use-Case Model



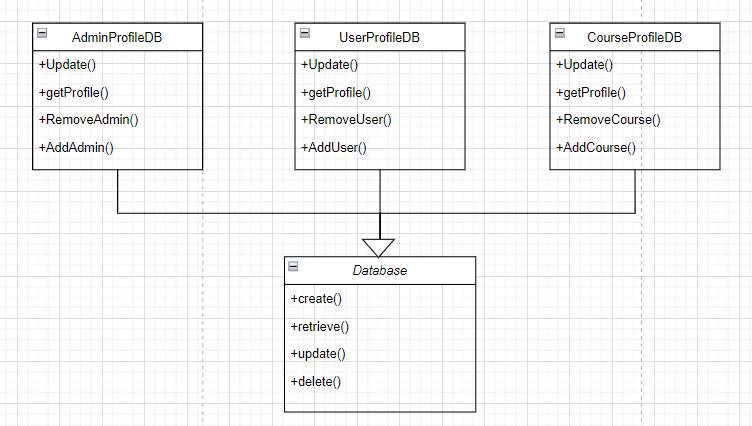
# Logical View

**

Hình 4‑1 Mô hình MVC cho Website Đăng ký học phần

|  |  |  |  |
| --- | --- | --- | --- |
| Thành phần | Miêu tả | Công nghệ sử dụng | Lý do sử dụng công nghệ đó |
| Database | Lưu trữ dữ liệu thành các bảng có mối liên kết với nhau | MySQL | Dữ liệu có cấu trúc rõ ràng, dễ quản lý, truy xuất bằng câu lệnh SQL có cú pháp chặt chẽ, dễ hiểu. |
| Model | Có trách nhiệm quản lý dữ liệu; nó lưu trữ và truy xuất các thực thể được ứng dụng sử dụng, thường là từ cơ sở dữ liệu và chứa logic được ứng dụng thực hiện. | PHP | Dễ dàng khởi tạo kết nối tới Database |
| View | Có nhiệm vụ hiển thị dữ liệu do mô hình cung cấp ở một định dạng cụ thể | HTML  CSS  Bootstrap | Dễ hỗ trợ, Bootstrap giúp tiết kiệm thời gian định dạng giao diện |
| Controller | Controller xử lý các Model và View để làm việc cùng nhau. Controller nhận được yêu cầu từ máy khách, gọi model để thực hiện các hoạt động được yêu cầu và gửi dữ liệu đến View | PHP | Ngôn ngữ quen thuộc, dễ dàng điều phối |
| Web Server | Chứa website, Xử lý HTTP request | Apache | Là một web server phổ biến nhất thế giới cho phép thiết lập một website an toàn mà không tốn nhiều công sức |

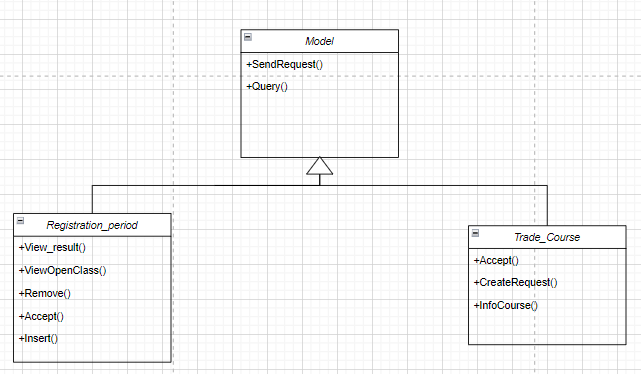
## Component: Database



Database có vai trò chứa dữ liệu của những đầu vào như Admin, User, Course,…

Trong mỗi đầu vào đều có những thông tin chi tiết và những tác vụ cần làm: thêm, xóa, sửa, cập nhật, hiển thị thông tin

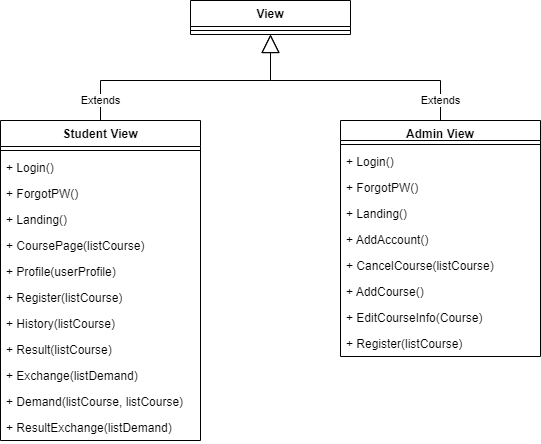
## Component: Model



Model dùng để xử lý logic, có các hàm để gửi yêu cầu và truy vấn để lấy dữ liệu và đầu ra là dữ liệu.

Các hàm model sẽ có các lớp trao đổi học phần và đăng ký học phần với các tác vụ như thêm, hiển thị, tạo yêu cầu,…

## Component: View



View đóng vai trò nhận dữ liệu và định dạng lại thành giao diện người dùng. Do website có 2 loại người dùng là sinh viên và admin nên chia làm 2 lớp Student View và Admin View

Mỗi hàm trong lớp đóng vai trò nhận dữ liệu/yêu cầu từ Controller, tiến hành định dạng để có thể hiển thị thành giao diện người dùng có thể tương tác

## Component: Controller

Diagram

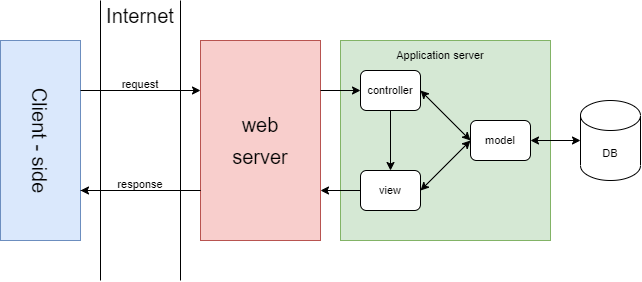
Description automatically generated

Controller đóng vai trò điều khiển hoạt động của View và Model để có thể làm việc cùng nhau (mô hình MVC)

Các lớp kế thừa đại diện cho chức năng của mỗi trang giao diện (chưa bao gồm các lớp chức năng của admin)

Mỗi hàm bên trong các lớp ứng với một hàm điều khiển hoạt động

## Component: Webserver



Webserver lưu trữ toàn bộ dữ liệu website và nắm quyền quản lý. Web server có thể lấy thông tin request từ phía trình duyệt web và gửi phần hồi tới máy khách thông qua giao thức HTTP

# Deployment

# Implementation View