**ĐẠI HỌC PHENIKAA**

**KHOA CÔNG NGHỆ THÔNG TIN**

**Ảnh có chứa văn bản, Phông chữ, biểu tượng, Đồ họa

Mô tả được tạo tự động**

**BÀI TẬP LỚN**

**HỌC PHẦN: KĨ THUẬT PHẦN MỀM**

**ĐỀ TÀI: Việc quản lý cựu sinh viên**

**Nhóm sinh viên thực hiện: Nhóm 07**

|  |  |  |
| --- | --- | --- |
| 1. | Trịnh Đình Đức Trung | 23010108 |
| 2. | Nguyễn Việt Cường | 23010551 |
| 3. | Hoàng Quang Vinh | 23010110 |
| 4. | Nguyễn Văn Minh | 23010778 |

**Lớp: COUR01.LT8**

**Mã học phần: CSE702025**

**Giảng viên: Vũ Quang Dũng**

**Hà Nội, tháng 06 – 2025**

**MỤC LỤC**

[**MỞ ĐẦU** 1](#_Toc201276181)

[**CHƯƠNG 1. GIỚI THIỆU VỀ PHÁT TRIỂN HỆ THỐNG PHẦN MỀM** 2](#_Toc201276182)

[1.1. Tổng quan hệ thống 2](#_Toc201276183)

[1.2. Mục tiêu phát triển 2](#_Toc201276184)

[1.3. Phạm vi hệ thống 2](#_Toc201276185)

[**CHƯƠNG 2. MÔ HÌNH PHÁT TRIỂN PHẦN MỀM WATERFALL** 4](#_Toc201276186)

[2.1. Giới thiệu mô hình Waterfall 4](#_Toc201276187)

[2.2. Các giai đoạn trong mô hình Waterfall và áp dụng vào hệ thống quản lý cựu sinh viên 4](#_Toc201276188)

[**CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ** 7](#_Toc201276189)

[3.1. Phân tích yêu cầu 7](#_Toc201276190)

[3.2. Yêu cầu chức năng (Functional Requirements) 7](#_Toc201276191)

[*3.1.1 Giao diện Login* 7](#_Toc201276192)

[*3.1.2 Alumni* 7](#_Toc201276193)

[*3.1.3 Admin* 7](#_Toc201276194)

[*3.1.4 Guest (khách truy cập)* 8](#_Toc201276195)

[3.3. Biểu đồ Use Case 8](#_Toc201276196)

[3.4. UML Diagram 9](#_Toc201276197)

[*3.4.1 Các gói chính (Packages)* 9](#_Toc201276198)

[*3.4.2. Sơ đồ lớp (Class Diagram)* 10](#_Toc201276199)

[*3.4.3 Chi tiết các lớp:* 10](#_Toc201276200)

[3.5. Kết luận 11](#_Toc201276201)

[**CHƯƠNG 4. GIAO DIỆN NGƯỜI DÙNG VÀ UI FLOW** 12](#_Toc201276202)

[4.1. Giao diện người dùng 12](#_Toc201276203)

[*4.1.1 Giao diện Alumni:* 12](#_Toc201276204)

[*4.1.2 Giao diện Admin:* 12](#_Toc201276205)

[*4.1.3 Giao diện Guest:* 12](#_Toc201276206)

[4.2. UI Flow (Luồng giao diện người dùng) 13](#_Toc201276207)

[**CHƯƠNG 5. FLOW ĐẶC TẢ HOẠT ĐỘNG** 14](#_Toc201276208)

[5.1. Biểu đồ hoạt động (Activity Diagram) 14](#_Toc201276209)

[5.2. Biểu đồ trình tự (Sequence Diagram) 14](#_Toc201276210)

[**CHƯƠNG 6. KIỂM THỬ HỆ THỐNG** 16](#_Toc201276211)

[6.1. Các loại kiểm thử áp dụng 16](#_Toc201276212)

[6.2. Kế hoạch kiểm thử các chức năng 16](#_Toc201276213)

[6.3. Kết quả tổng hợp 16](#_Toc201276214)

[**CHƯƠNG 7. ƯU ĐIỂM, HẠN CHẾ VÀ HƯỚNG PHÁT TRIỂN** 17](#_Toc201276215)

[7.1. Ưu điểm 17](#_Toc201276216)

[7.2. Hạn chế 17](#_Toc201276217)

[7.3. Hướng phát triển 17](#_Toc201276218)

[**KẾT LUẬN** 18](#_Toc201276219)

[**TÀI LIỆU THAM KHẢO** 19](#_Toc201276220)

**LỜI CAM ĐOAN**

Chúng em cam đoan bài tập lớn môn học là sản phẩm trí tuệ của tập thể nhóm chúng em. Mọi thông tin, dữ liệu hình ảnh ,…được sử dụng từ các nguồn khác đều được trích dẫn đầy đủ và có thể tìm thấy các tài liệu liên quan thông qua tài liệu tham khảo.

Chúng em xin chịu trách nhiệm hoàn toàn về nội dung bài tập lớn môn học chúng em đã nộp.

**LỜI CẢM ƠN**

Chúng em xin gửi lời cảm ơn chân thành đến Thầy Vũ Quang Dũng đã giảng dạy chúng em trong suốt học kì vừa qua. Cảm ơn thầy đã hướng dẫn chúng em thực hiện xong bài tập lớn môn này.

Tuy nhiên do thời gian hạn hẹp, mặc dù đã nỗ lực hết sức mình, nhưng chắc chắn rằng bài tập lớn khó tránh khỏi thiếu sót và lỗi lầm .Chúng em rất mong nhận sự thông cảm và chỉ bảo tận tình của thầy và các bạn.

Chúng em xin chân thành cảm ơn !

***Hà Nội, ngày 18 tháng 6 năm 2025***

**PHÂN CÔNG CÔNG VIỆC**

|  |  |
| --- | --- |
| **Thành viên** | **Công việc đảm nhận** |
| Nguyễn Việt Cường | Phân tích yêu cầu, Viết tài liệu chương 1–2 |
| Hoàng Quang Vinh | Thiết kế CSDL, giao diện người dùng (Alumni) |
| Nguyễn Văn Minh | Lập trình chức năng Admin và backend |
| Trịnh Đình Đức Trung | Kiểm thử hệ thống, tổng hợp báo cáo, chương 6–7 |

**NHẬN XÉT CỦA GIẢNG VIÊN**

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

Hà Nội, ngày tháng năm 2025

Giảng Viên

# **MỞ ĐẦU**

Trong thời đại số hóa hiện nay, việc duy trì và phát triển mối quan hệ giữa nhà trường và các thế hệ sinh viên đã tốt nghiệp ngày càng trở nên quan trọng. Cựu sinh viên không chỉ là những người đại diện cho hình ảnh, chất lượng đào tạo của nhà trường mà còn là nguồn lực quý báu trong việc phát triển các hoạt động học thuật, nghề nghiệp và đóng góp cộng đồng. Nhằm mục tiêu tăng cường sự gắn kết giữa nhà trường và cựu sinh viên, nhóm chúng em đã xây dựng hệ thống “Quản lý cựu sinh viên”. Hệ thống này hỗ trợ việc lưu trữ, cập nhật thông tin, tổ chức sự kiện, thống kê đóng góp và tạo diễn đàn kết nối giữa các cựu sinh viên với nhau cũng như với nhà trường.

Thông qua báo cáo này, nhóm sẽ trình bày toàn bộ quy trình phân tích, thiết kế, xây dựng và kiểm thử hệ thống quản lý cựu sinh viên, với mong muốn ứng dụng hiệu quả vào thực tế cũng như cải thiện kỹ năng phát triển phần mềm theo hướng tiếp cận chuyên nghiệp.

Đồ án không chỉ là cơ hội để nhóm em áp dụng các kiến thức lý thuyết đã học về quy trình phát triển phần mềm, mà còn là dịp để rèn luyện kỹ năng giải quyết các vấn đề thực tế trong việc phát triển một sản phẩm công nghệ phức tạp. Nhóm em hy vọng rằng, qua đề tài này, có thể mang lại một cái nhìn toàn diện hơn về quy trình xây dựng phần mêm và các thách thức mà một kỹ sư phần mềm cần đối mặt trong môi trường phát triển phần mềm hiện đại.

Nhóm em chân thành cảm ơn sự hướng dẫn và hỗ trợ từ giảng viên, đồng thời hy vọng rằng dự án này sẽ nhận được sự đóng góp ý kiến quý báu từ quý thầy cô và các bạn học để có thể hoàn thiện hơn.

# **CHƯƠNG 1. GIỚI THIỆU VỀ PHÁT TRIỂN HỆ THỐNG PHẦN MỀM**

## **1.1. Tổng quan hệ thống**

Hệ thống quản lý cựu sinh viên là một phần mềm ứng dụng web được thiết kế nhằm hỗ trợ nhà trường trong việc lưu trữ, theo dõi và cập nhật thông tin của sinh viên sau khi tốt nghiệp. Mục tiêu chính của hệ thống là giúp nhà trường duy trì kết nối với cựu sinh viên, khuyến khích sự tham gia của họ vào các hoạt động trường lớp, đồng thời hỗ trợ công tác thống kê và xây dựng cơ sở dữ liệu alumni phục vụ các mục tiêu giáo dục và hợp tác lâu dài.

* Các thành phần chính của hệ thống bao gồm:
* Alumni (cựu sinh viên): Người dùng có thể đăng ký, đăng nhập, cập nhật hồ sơ cá nhân, gửi bài viết diễn đàn, xem sự kiện và đóng góp.
* Admin (quản trị viên): Người quản lý hệ thống có quyền tạo sự kiện, duyệt bài viết, chỉnh sửa thông tin alumni, thống kê và báo cáo hoạt động.
* Guest (khách truy cập): Người dùng không đăng nhập có thể tra cứu thông tin alumni công khai và theo dõi sự kiện.
* Các hoạt động chính của hệ thống:
* Quản lý hồ sơ cựu sinh viên
* Quản lý sự kiện, thông báo
* Diễn đàn trao đổi giữa các alumni
* Ghi nhận và thống kê đóng góp tài chính, hỗ trợ
* Tìm kiếm và kết nối các alumni theo ngành, khóa học hoặc nghề nghiệp

## **1.2. Mục tiêu phát triển**

* Xây dựng nền tảng trực tuyến thân thiện, dễ sử dụng.
* Hỗ trợ alumni cập nhật thông tin, tham gia diễn đàn, đóng góp.
* Giúp admin quản lý dễ dàng alumni, sự kiện và bài viết.
* Tạo môi trường kết nối lâu dài giữa nhà trường và cựu sinh viên.

## **1.3. Phạm vi hệ thống**

Người dùng chính: Cựu sinh viên, quản trị viên, khách truy cập.

Chức năng: Đăng ký, đăng nhập, cập nhật thông tin, tham gia diễn đàn, theo dõi sự kiện và quản lý đóng góp

# **CHƯƠNG 2. MÔ HÌNH PHÁT TRIỂN PHẦN MỀM WATERFALL**

## **2.1. Giới thiệu mô hình Waterfall**

Mô hình Waterfall (Thác nước) là một trong những mô hình phát triển phần mềm truyền thống, được thực hiện theo trình tự tuyến tính và tuần tự chặt chẽ. Mỗi giai đoạn trong quá trình phát triển phải được hoàn tất hoàn toàn trước khi chuyển sang bước tiếp theo.

* Đặc điểm chính:
* Quy trình phát triển diễn ra theo các bước nối tiếp nhau.
* Dễ quản lý, dễ theo dõi tiến độ.
* Phù hợp với dự án có yêu cầu rõ ràng và ít thay đổi.

## **2.2. Các giai đoạn trong mô hình Waterfall và áp dụng vào hệ thống quản lý cựu sinh viên**

***Giai đoạn 1: Phân tích yêu cầu (Requirement Analysis)***

* Mục tiêu: Thu thập và xác định các yêu cầu chức năng, phi chức năng từ người dùng và nhà trường.
* Áp dụng vào hệ thống:
* Phỏng vấn cán bộ quản lý đào tạo, cựu sinh viên để xác định nhu cầu.
* Yêu cầu chính: quản lý hồ sơ cựu sinh viên, diễn đàn trao đổi, sự kiện, đóng góp.
* Kết quả: Tài liệu yêu cầu phần mềm (SRS - Software Requirement Specification) được lập và thông qua.

***Giai đoạn 2: Thiết kế hệ thống (System Design)***

* Mục tiêu: Chuyển các yêu cầu thành bản thiết kế chi tiết về kiến trúc, dữ liệu và giao diện người dùng.
* Áp dụng vào hệ thống:
* Thiết kế sơ đồ Use Case thể hiện mối quan hệ giữa người dùng và chức năng hệ thống.
* Vẽ sơ đồ lớp (Class Diagram) thể hiện các thực thể như Alumni, Event, Admin, ForumPost, Donation.
* Thiết kế giao diện người dùng (UI) cho các vai trò: Alumni, Admin, Guest.
* Thiết kế cơ sở dữ liệu gồm 5 bảng chính: alumni, admin, events, forum\_posts, donations.

***Giai đoạn 3: Cài đặt (Implementation)***

* Mục tiêu: Viết mã nguồn theo thiết kế đã phê duyệt.
* Áp dụng vào hệ thống:
* Sử dụng ngôn ngữ PHP, MySQL làm hệ quản trị cơ sở dữ liệu.
* Tạo module:

alumni/: chứa các file giao diện người dùng alumni.

admin/: chứa chức năng quản lý (sự kiện, bài viết...).

includes/: chứa file kết nối cơ sở dữ liệu.

* Giao diện được thiết kế với HTML, CSS (Bootstrap) và PHP để xử lý logic.

***Giai đoạn 4: Kiểm thử (Testing)***

* Mục tiêu: Phát hiện và sửa lỗi, đảm bảo hệ thống hoạt động đúng yêu cầu.
* Áp dụng vào hệ thống:
* Kiểm thử chức năng: đăng ký, đăng nhập, gửi bài viết, duyệt sự kiện.
* Kiểm thử giao diện: hiển thị đúng trên các trình duyệt phổ biến (Chrome, Firefox).
* Sử dụng dữ liệu giả lập để kiểm tra độ chính xác khi truy vấn và thống kê.
* Kiểm thử bảo mật cơ bản (SQL Injection).

***Giai đoạn 5: Triển khai (Deployment)***

* Mục tiêu: Đưa hệ thống vào môi trường sử dụng thực tế.
* Áp dụng vào hệ thống:
* Cài đặt hệ thống trên localhost (XAMPP) để trình diễn.
* Hướng dẫn admin thao tác và nhập dữ liệu mẫu.
* Chuẩn bị kịch bản hướng dẫn người dùng cựu sinh viên đăng ký và sử dụng hệ thống.

***Giai đoạn 6: Bảo trì (Maintenance)***

* Mục tiêu: Sửa lỗi phát sinh sau triển khai, cập nhật theo nhu cầu người dùng.
* Áp dụng vào hệ thống: Thu thập phản hồi từ người dùng sau buổi demo thử nghiệm.
* Cập nhật các tính năng mới: lọc alumni theo ngành, xuất danh sách ra Excel, gửi email tự động.
* Vá lỗi giao diện chưa tương thích mobile và lỗi thời gian hiển thị diễn đàn.

# **CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ**

## **3.1. Phân tích yêu cầu**

Trong bối cảnh chuyển đổi số giáo dục, nhà trường cần một công cụ giúp lưu trữ, kết nối và thống kê thông tin của các cựu sinh viên đã tốt nghiệp. Hệ thống Quản lý cựu sinh viên (Alumni Management System) được thiết kế nhằm mục tiêu: giúp quản trị viên nhà trường quản lý alumni, kết nối cựu sinh viên với nhau thông qua sự kiện, diễn đàn, và thống kê các hoạt động đóng góp.

Hệ thống phục vụ ba đối tượng chính: Cựu sinh viên (Alumni), Quản trị viên (Admin) và Khách truy cập (Guest). Dựa vào mô hình hướng đối tượng, nhóm đã xây dựng hệ thống theo các chức năng chính sau:

## **3.2. Yêu cầu chức năng (Functional Requirements)**

### ***3.1.1 Giao diện Login***

* Cho phép cả Admin và Alumni đăng nhập bằng email và mật khẩu.
* Xác thực thông tin đăng nhập qua cơ sở dữ liệu.
* Chuyển đến giao diện riêng tương ứng với vai trò.

### ***3.1.2 Alumni***

* Đăng ký (Sign Up): Nhập tên, email, mật khẩu, khóa học, nghề nghiệp. Dữ liệu được lưu vào bảng alumni.
* Cập nhật hồ sơ: Sửa thông tin cá nhân.
* Xem sự kiện: Danh sách sự kiện đang hoạt động hoặc đã qua.
* Gửi bài viết diễn đàn: Alumni có thể đăng bài chia sẻ, kinh nghiệm.
* Xem lịch sử đóng góp: Tra cứu các khoản quyên góp trước đó.

### ***3.1.3 Admin***

* Quản lý alumni: Xem danh sách, chỉnh sửa hoặc xóa alumni.
* Tạo sự kiện: Nhập tiêu đề, mô tả, thời gian tổ chức.
* Duyệt bài viết diễn đàn: Kiểm tra nội dung trước khi bài viết hiển thị công khai.
* Thống kê đóng góp: Xem tổng số tiền đóng góp theo thời gian hoặc theo alumni.

### ***3.1.4 Guest (khách truy cập)***

* Xem danh sách alumni theo ngành, khóa học.
* Xem danh sách sự kiện được công khai.

## **3.3. Biểu đồ Use Case**

Biểu đồ Use Case mô tả mối quan hệ giữa các vai trò người dùng với các chức năng chính.

Diagram of a user interface

Description automatically generated

* **Tác nhân:**
* Alumni: Đăng ký, đăng nhập, cập nhật hồ sơ, tham gia diễn đàn, xem sự kiện, xem lịch sử đóng góp.
* Admin: Quản lý thông tin alumni, duyệt bài viết, tạo sự kiện, thống kê.
* Guest: Xem thông tin alumni và sự kiện công khai.
* **Các Use Case tiêu biểu:**
* Đăng ký / đăng nhập
* Quản lý hồ sơ cá nhân
* Gửi bài viết
* Tạo sự kiện
* Duyệt nội dung
* Tra cứu đóng góp
* Mối quan hệ:

<<include>>: Duyệt bài viết include Xem bài viết.

<<extend>>: Guest có thể extend chức năng alumni nếu đăng ký tài khoản.

## **3.4. UML Diagram**

### ***3.4.1 Các gói chính (Packages)***

* Controller: Xử lý logic tương tác người dùng.
* Model: Chứa các lớp đại diện dữ liệu như Alumni, Event, ForumPost.
* DAO: Giao tiếp với CSDL, thực hiện CRUD.
* View: Giao diện người dùng (HTML/PHP).

### ***3.4.2. Sơ đồ lớp (Class Diagram)***

A diagram of a program

Description automatically generated

### ***3.4.3 Chi tiết các lớp:***

a. Model Classes:

* Alumni: Chứa thông tin cá nhân cựu sinh viên.
* Admin: Tài khoản quản trị viên.
* Event: Sự kiện do nhà trường tạo.
* ForumPost: Bài viết của alumni trên diễn đàn.
* Donation: Lịch sử đóng góp của alumni.

b. DAO Classes:

* AlumniDAO: Thêm, sửa, xóa alumni.
* EventDAO: Quản lý bảng sự kiện.
* ForumPostDAO: Truy xuất bài viết.
* DonationDAO: Truy xuất lịch sử đóng góp.
* AdminDAO: Kiểm tra tài khoản admin.

c. Controller Classes:

* LoginController: Xử lý đăng nhập.
* AlumniController: Cập nhật thông tin cá nhân.
* EventController: Tạo / sửa sự kiện.
* ForumController: Quản lý bài viết.
* DonationController: Xử lý thống kê đóng góp.

d. View Files (PHP/HTML):

* login.php, register.php
* dashboard.php
* manage\_events.php, donate.php, forum.php
* Giao diện chia theo folder alumni/, admin/, guest/.

## **3.5. Kết luận**

Thông qua việc áp dụng phương pháp phân tích hướng đối tượng và mô hình MVC (Model-View-Controller), hệ thống đã được thiết kế đầy đủ về mặt chức năng, kiến trúc và cơ sở dữ liệu. Các sơ đồ Use Case và Class giúp dễ dàng mở rộng và bảo trì hệ thống trong tương lai.

# **CHƯƠNG 4. GIAO DIỆN NGƯỜI DÙNG VÀ UI FLOW**

## A screenshot of a website Description automatically generated**4.1. Giao diện người dùng**

Hệ thống được thiết kế với giao diện trực quan, dễ sử dụng phù hợp với từng đối tượng sử dụng.

### ***4.1.1 Giao diện Alumni:***

* Trang đăng ký (register.php)
* Trang đăng nhập (login.php)
* Trang hồ sơ cá nhân
* Trang xem sự kiện
* Trang viết bài diễn đàn
* Trang xem lịch sử đóng góp

### ***4.1.2 Giao diện Admin:***

* Trang quản lý alumni
* Trang tạo/sửa sự kiện
* Trang duyệt bài viết diễn đàn
* Trang thống kê đóng góp

### ***4.1.3 Giao diện Guest:***

* Trang xem danh sách alumni công khai
* Trang xem sự kiện

## **4.2. UI Flow (Luồng giao diện người dùng)**

* Guest → [Trang chủ] → [Đăng nhập / Đăng ký] → Alumni hoặc Admin Dashboard
* Alumni → [Dashboard] → [Hồ sơ] → [Cập nhật / Xem sự kiện / Viết bài / Xem đóng góp]
* Admin → [Dashboard] → [Quản lý alumni / Tạo sự kiện / Duyệt bài viết / Thống kê]

**4.3. Sơ đồ chi tiết về về giao diện người dùng và UI Flow**

A diagram of a user flow

Description automatically generated

# **CHƯƠNG 5. FLOW ĐẶC TẢ HOẠT ĐỘNG**

## **5.1. Biểu đồ hoạt động (Activity Diagram)**

* Hoạt động đăng ký
* Hoạt động gửi bài diễn đàn
* Hoạt động tạo sự kiện

A group of rectangular objects with different colored text

Description automatically generated with medium confidence

## **5.2. Biểu đồ trình tự (Sequence Diagram)**

* Sequence: Alumni → gửi bài viết → hệ thống kiểm tra → chờ duyệt → Admin duyệt → bài viết hiển thị
* Sequence: Admin → tạo sự kiện → lưu cơ sở dữ liệu → Alumni có thể xem

A screenshot of a chat

Description automatically generated

# **CHƯƠNG 6. KIỂM THỬ HỆ THỐNG**

## **6.1. Các loại kiểm thử áp dụng**

* Kiểm thử chức năng (Function testing)
* Kiểm thử giao diện (UI testing)
* Kiểm thử bảo mật cơ bản (SQL Injection)

## **6.2. Kế hoạch kiểm thử các chức năng**

|  |  |  |
| --- | --- | --- |
| **Chức năng** | **Tình trạng** | **Mô tả kiểm thử** |
| **Đăng ký Alumni** | Đạt | Nhập dữ liệu đúng/sai, kiểm tra lưu trữ |
| **Đăng nhập Alumni/Admin** | Đạt | Kiểm tra xác thực tài khoản |
| **Tạo sự kiện** | Đạt | Admin tạo, kiểm tra hiển thị |
| **Gửi bài viết** | Đạt | Alumni gửi, kiểm duyệt, hiển thị |
| **Xem thống kê đóng góp** | Đạt | Kiểm tra tổng hợp dữ liệu |

## **6.3. Kết quả tổng hợp**

* Tỷ lệ chức năng hoàn thành: 100%
* Số lỗi được phát hiện: 5 (đã sửa)
* Giao diện tương thích với các trình duyệt phổ biến: Chrome, Firefox

# **CHƯƠNG 7. ƯU ĐIỂM, HẠN CHẾ VÀ HƯỚNG PHÁT TRIỂN**

## **7.1. Ưu điểm**

* Giao diện thân thiện, rõ ràng, dễ sử dụng.
* Hệ thống hoạt động ổn định, đầy đủ chức năng cơ bản.
* Sử dụng PHP + MySQL phù hợp với hệ thống web đơn giản.

## **7.2. Hạn chế**

* Chưa có phân quyền chi tiết theo nhóm Alumni.
* Thiếu tính năng gửi email thông báo tự động.
* Giao diện chưa tối ưu cho thiết bị di động.

## **7.3. Hướng phát triển**

* Nâng cấp giao diện tương thích với thiết bị di động.
* Bổ sung chức năng gửi email nhắc nhở sự kiện.
* Phân quyền Alumni theo khóa học, ngành học.
* Tích hợp chức năng export báo cáo (PDF/Excel).

# **KẾT LUẬN**

Hệ thống quản lý cựu sinh viên là một giải pháp hữu ích giúp nhà trường dễ dàng lưu trữ, theo dõi và kết nối với cựu sinh viên. Việc áp dụng mô hình phát triển Waterfall đã giúp nhóm triển khai dự án một cách rõ ràng, có trình tự và hoàn chỉnh. Mặc dù vẫn còn một số hạn chế, hệ thống đã đạt được các mục tiêu đề ra và sẵn sàng triển khai thực tế với những cải tiến trong tương lai.

# **TÀI LIỆU THAM KHẢO**

1. <https://www.php.net/>
2. <https://www.mysql.com/>
3. <https://www.w3schools.com/>
4. Giáo trình Phân tích thiết kế hệ thống thông tin - NXB GD
5. Tài liệu hướng dẫn sử dụng XAMPP, Bootstrap