|  |
| --- |
| **TRƯỜNG ĐẠI HỌC THÁI BÌNH**  **KHOA CÔNG NGHỆ THÔNG TIN**    **KHÓA LUẬN TỐT NGHIỆP**  **NGÀNH CÔNG NGHỆ THÔNG TIN**  **TÊN ĐỀ TÀI**  **THIẾT KẾ VÀ XÂY DỰNG WEBSITE QUẢN LÝ HỒ SƠ CÁN BỘ, GIẢNG VIÊN, VIÊN CHỨC VÀ NGƯỜI LAO ĐỘNG CỦA TRƯỜNG ĐẠI HỌC THÁI BÌNH**    **Sinh viên thực hiện : Lê Thế Kỷ**  **Mã sinh viên : 1901570**  **Lớp : ĐH8CNTT**  **Khóa đào tạo : 2019 - 2023**  **Giảng viên hướng dẫn : Bùi Thị Hòa**  *Thái Bình, tháng 6 năm 2023* |

**LỜI MỞ ĐẦU**

Với sự phát triển bùng nổ của website internet toàn cầu, thương mại điện tử đang đạt được những thành công đáng kể. Trong đó website quản lý là một công cụ không thể thiếu để tạo nên thành công bền vững cho nhà trường trong thời đại công nghệ 4.0.

Nếu không có website, các đơn vị quản lý vẫn có thể hoạt động, tuy nhiên sẽ bỏ lỡ rất nhiều cơ hội từ những người dùng tiềm năng và gặp khó khăn trong việc mở rộng, tiếp cận thị trường mới.

Website quản lý cung cấp cho người dùng những lựa chọn linh hoạt và tiện lợi trong việc quản lý hồ sơ. Về phía nhà trường, website mang lại sự tiện dụng trong việc cập nhập và quản lý các thông tin cho website, tạo khả năng liên lạc nhanh và dễ dàng với người dùng.

Chính vì nguyên do đó, việc “Thiết kế và xây dựng Website quản lý hồ sơ cán bộ, giảng viên, viên chức và người lao động của trường Đại học Thái Bình” là thực sự cần thiết trong xã hội 4.0 hiện nay. Nhằm nâng cao chất lượng phục vụ người dùng, cung cấp đầy đủ thông tin cho người quản lý, dễ dàng tiếp cận với hồ sơ, xây dựng hình ảnh tốt cho nhà trường trong mắt người dùng và tạo cơ hội tăng trưởng vượt bậc …

**LỜI CAM ĐOAN**

Em xin cam đoan đề tài: “Thiết kế và xây dựng Website quản lý hồ sơ cán bộ, giảng viên, viên chức và người lao động của trường Đại học Thái Bình” là một công trình nghiên cứu độc lập của em dưới sự hướng dẫn của giảng viên: Bùi Thị Hòa. Ngoài ra không có bất cứ sự sao chép nào của người khác. Đề tài, nội dung báo cáo khóa luận là sản phẩm mà em đã nỗ lực nghiên cứu trong quá trình học tập tại trường đại học Thái Bình. Các số liệu, kết quả trình bày trong báo cáo là hoàn toàn trung thực, em xin chịu hoàn toàn trách nhiệm, chịu mọi hình thức kỷ luật của bộ môn, của khoa và của nhà trường nếu như có phát hiện gian lận trong khóa luận.

Tác giả Dự án

**Lê Thế Kỷ**

**LỜI CẢM ƠN**

Trước tiên em xin gửi lời cảm ơn chân thành sâu sắc tới các thầy cô giáo trong trường Đại học Thái Bình nói chung và các thầy cô giáo trong khoa Công nghệ thông tin nói riêng đã tận tình giảng dạy, truyền đạt cho em những kiến thức, kinh nghiệm quý báu trong suốt thời gian bốn năm học vừa qua.

Đặc biệt em xin gửi lời cảm ơn đến cô giáo Bùi Thị Hòa đã tận tình giúp đỡ, trực tiếp chỉ bảo, huớng dẫn em trong suốt quá trình thực hiện khóa luận tốt nghiệp. Trong thời gian làm việc, em không ngừng tiếp thu thêm nhiều kiến thức bổ ích mà còn học tập được tinh thần làm việc, thái độ nghiên cứu khoa học nghiêm túc, hiệu quả, đây là những điều rất cần thiết cho em trong quá trình học tập và công tác sau này.

Sau cùng xin gửi lời cảm ơn chân thành tới gia đình, bạn bè đã động viên, đóng góp ý kiến và giúp đỡ em trong quá trình học tập, nghiên cứu và hoàn thành khoá luận tốt nghiệp.

Một lần nữa, em xin chân thành cảm ơn.!

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

**Giảng viên**

*(Ký, ghi rõ họ tên)*

**NHẬN XÉT CỦA GIẢNG VIÊN PHẢN BIỆN 1**

**Giảng viên**

*(Ký, ghi rõ họ tên)*

**NHẬN XÉT CỦA GIẢNG VIÊN PHẢN BIỆN 2**

**Giảng viên**

*(Ký,ghi rõ họ tên)*

MỤC LỤC

[CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI 1](#_Toc136288747)

[1.1. Thông tin chung 1](#_Toc136288748)

[1.2. Mục tiêu của đề tài 1](#_Toc136288749)

[1.3. Giới hạn và phạm vi của đề tài 1](#_Toc136288750)

[1.4. Nội dung thực hiện 2](#_Toc136288751)

[1.5. Phương pháp tiếp cận 2](#_Toc136288752)

[1.6. Kết quả chính đạt được 2](#_Toc136288753)

[CHƯƠNG 2. KHẢO SÁT THỰC TẾ PHÂN TÍCH YÊU CẦU 4](#_Toc136288754)

[2.1. Khảo sát trường đại học Thái Bình 4](#_Toc136288755)

[2.2. Yêu cầu của website sau khảo sát 4](#_Toc136288756)

[2.3. Các chức năng của website 6](#_Toc136288757)

[2.4. Các tác nhân của website 6](#_Toc136288758)

[CHƯƠNG 3. PHÂN TÍCH THIẾT KẾ HỆ THỐNG 8](#_Toc136288759)

[3.1. Biểu đồ Use-case 8](#_Toc136288760)

[3.2. Biểu đồ tuần tự 22](#_Toc136288761)

[3.3. Biểu đồ hoạt động 24](#_Toc136288762)

[3.4. Biểu đồ lớp 25](#_Toc136288763)

[3.4. Thiết kế cơ sở dữ liệu 28](#_Toc136288764)

[3.4.1. Các bảng dữ liệu 28](#_Toc136288765)

[CHƯƠNG 4. THIẾT KẾ GIAO DIỆN WEBSITE 33](#_Toc136288766)

[CHƯƠNG 5. CÀI ĐẶT MỘT SỐ CHỨC NĂNG CHO WEBISTE 39](#_Toc136288767)

[5.1. Chức năng hợp đồng trên server 39](#_Toc136288768)

[5.2. Chức năng phòng ban trên server 44](#_Toc136288769)

[5.3. Chức năng định tuyến từ máy client lên server 48](#_Toc136288864)

[5.4. Chức năng điều hướng trên website 53](#_Toc136288865)

[CHƯƠNG 6. KẾT LUẬN 62](#_Toc136288866)

[6.1. Kết quả đạt được của đề tài 62](#_Toc136288867)

[6.2. Hạn chế của đề tài 62](#_Toc136288868)

[6.3. Hướng phát triển của đề tài 62](#_Toc136288869)

[TÀI LIỆU THAM KHẢO 63](#_Toc136288870)

DANH SÁCH HÌNH VẼ

[Hình 2. 1: Trường đại học Thái Bình 4](#_Toc73953922)

[Hình 3. 1: Biểu đồ use-case tổng quát 8](#_Toc73953923)

[Hình 3. 2: Biểu đồ use-case đăng nhập 9](#_Toc73953924)

[Hình 3. 3: Biểu đồ use-case quản lý phòng ban 1](#_Toc73953925)0

[Hình 3. 4: Biểu đồ use-case quản lý nhân viên 1](#_Toc73953926)1

[Hình 3. 5: Biểu đồ use-case quản lý hợp đồng 1](#_Toc73953927)2

[Hình 3. 6: Biểu đồ use-case quản lý kế hoạch đào tạo 1](#_Toc73953928)3

[Hình 3. 7: Biểu đồ use-case quản lý nhà trình độ chuyên môn 1](#_Toc73953929)5

[Hình 3. 8: Biểu đồ use-case quản lý đơn từ 1](#_Toc73953930)6

[Hình 3. 9: Biểu đồ use-case quản lý nâng lương cán bộ 1](#_Toc73953931)7

[Hình 3. 10: Biểu đồ use-case giáo viên 19](#_Toc73953932)

[Hình 3. 11: Biểu đồ use-case nhân viên 20](#_Toc73953933)

[Hình 3. 12: Biểu đồ tuần tự chức năng đăng nhập 2](#_Toc73953934)2

[Hình 3. 13: Biều đồ tuần tự chức thêm phòng ban 22](#_Toc73953935)

[Hình 3. 14: Biểu đồ tuần tự chức năng sửa phòng ban 23](#_Toc73953936)

[Hình 3. 15: Biểu đồ hoạt động xóa phòng ban 23](#_Toc73953937)

[Hình 3. 16: Biểu đồ lớp 2](#_Toc73953939)4

[Hình 3. 17: Mô hình quan hệ diagram 25](#_Toc73953940)

DANH SÁCH BẢNG BIỂU

[Bảng 3. 1: Bảng danh sách các đối tượng 2](#_Toc73954232)6

[Bảng 3. 2:Bảng Staff](#_Toc73954233) 26

[Bảng 3. 3: Bảng Plan](#_Toc73954234) 27

[Bảng 3. 4: Bảng User](#_Toc73954235) 27

[Bảng 3. 5: Bảng LeaveRequest](#_Toc73954236) 27

[Bảng 3. 6: Bảng Squelizemeta](#_Toc73954237) 28

[Bảng 3. 7:Bảng Employee](#_Toc73954238) 28

[Bảng 3. 8: Bảng Department](#_Toc73954239) 29

[Bảng 3. 9: Bảng Teacher](#_Toc73954240) 29

[Bảng 3. 10: Bảng Promotion](#_Toc73954241) 29

[Bảng 3. 11: Bảng Qualification](#_Toc73954242) 30

[Bảng 3. 12: Bảng Contract](#_Toc73954243) 30

DANH SÁCH TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| Từ viết tắt | Từ đầy đủ | Giải thích |
| CSDL | Cơ sở dữ liệu | Cơ sở dữ liệu |
| CMS | **C**ontent management system | Hệ thống quản trị nội dung |
| HTML | **Hypertext Markup Language** | Ngôn ngữ siêu đánh dấu văn bản |
| MVC | **Model – View – Controler** | Mô hình Model-View-Controler |
| OOP | Object oriented program | Lập trình hướng đối tượng |
| ORM | Object-Relational Mapping | Bản đồ quan hệ giữa các đối tượng |
| SQL | Structured Query Language | Ngôn ngữ truy vấn cấu trúc |

# CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI

## 1.1. Thông tin chung

**Tên đề tài:** Thiết kế và xây dựng Website quản lý hồ sơ cán bộ, giảng viên, viên chức và người lao động của trường Đại học Thái Bình.

**Sinh viên thực hiện:** Lê Thế Kỷ

**Lớp:** ĐH8CNTT

**Hệ đào tạo:** Chính quy

**Điện thoại:** 0818268240

**Email:** [thekyle997@gmail.com](mailto:thekyle997@gmail.com)

**Thời gian thực hiện:** tháng 4-6 năm 2023.

1.2. Mục tiêu của đề tài

* Phân tích được bài toán Thiết kế và xây dựng Website quản lý hồ sơ cán bộ, giảng viên, viên chức và người lao động của trường Đại học Thái Bình.
* Tìm hiểu cơ sở lý thuyết của quản lý cơ sở dữ liệu My SQL và ngôn ngữ VueJS, NodeJS, mô hình MVC.
* Phân tích và thiết kế CSDL, các biểu đồ use-case và biểu đồ tuần tự.
* Thiết kế giao diện website quản lý và giao diện trang quản trị.
* Xây dựng website quản lý cho một cửa hàng dựa trên thư viện của Javasript cho phần Front-end VueJS, sử dụng NodeJS để viết API, đảm bảo việc Back-end dựa trên mô hình MVC, sử dụng Sequelize làm ORM framework và đồng thời đảm bảo việc CSDL MySQL được đặt trên Xampp.

1.3. Giới hạn và phạm vi của đề tài

Đề tài bước đầu được xây dựng để áp dụng cho việc quản lý website quản lý online trên website cho trường đại học Thái Bình và sau đó có thể phát triển thêm và ứng dụng cho các nhà trường quản lý lớn hơn trong thực tế.

Đề tài xây dựng sử dụng công nghệ dựa trên các framework của Javascript như VueJS, NodeJS và phát triển định hướng MVC, và sử dụng môi trường lập trình là Visual Studio Code đồng thời CSDL được thiết kế dựa trên MySQL.

1.4. Nội dung thực hiện

Nội dung thực hiện, nghiên cứu cụ thể của đề tài:

* Hiểu được cách sử dụng cơ bản của framework VueJS, NodeJS, trong đó có các đối tượng, biến, hàm, phương thức, thuộc tính…
* Viết RestfulAPI với NodeJS để giao tiếp giữa Back-end và Front-end
* Tìm hiểu mô hình MVC, trong thiết kế và xây dựng Back-end cho website với thư viện Express, NodeJS.
* Tìm hiểu cách sử dụng phần mềm Visual Studio Code và Xampp để thiết kế giao diện và xây dựng các chức năng website quản lý bằng các framework VueJS, NodeJS cũng như CSDL với MySQL.
* Biết cách sử dụng hệ quản trị CSDL bằng My SQL.
* Khảo sát và phân tích yêu cầu quy trình quản lý.
* Thiết kế và đặc tả website.
* Xây dựng cơ sở dữ liệu.
* Lập trình các module của website.
* Kiểm thử website.
* Triển khai thực nghiệp website trên mạng Internet.

1.5. Phương pháp tiếp cận

Nghiên cứu quy trình quản lý công tác giáo viên chủ nhiệm cho trường học và mạng internet, nghiên cứu ngôn ngữ VueJS, NodeJS, mô hình MVC (Model - View – Controller).

Sử dụng các phương pháp nghiên cứu:

* Phương pháp đọc tài liệu.
* Phương pháp phân tích mẫu, thực nghiệm.

## 1.6. Kết quả chính đạt được

* Thiết kế và xây dựng thành công website quản lý cho trường đại học Thái Bình.
* Ứng dụng phát triển được công nghệ thiết kế web bằng VueJS, NodeJS, RestfulAPI và MySQL.
* Tìm hiểu nắm bắt được các kiến thức về HTML, CSS, VueJS, NodeJS, MySQL và mô hình Model - View - Controller.

Với quy mô trang web vừa và nhỏ, cộng thêm thời gian không cho phép nên đề tài “Thiết kế và xây dựng Website quản lý hồ sơ cán bộ, giảng viên, viên chức và người lao động của trường Đại học Thái Bình” có thể chưa có đầy đủ các chức năng hoàn thiện của một trang web quản lý. Trong quá trình làm còn nhều sai sót, do kiến thức có hạn lên website còn ít nhiều thiếu sót về việc hiển thị cũng như tối ưu thuật toán, thời gian thực hiện. Trong tương lai em sẽ sẽ cố gắng khắc phục những sai sót này.

# CHƯƠNG 2. KHẢO SÁT THỰC TẾ PHÂN TÍCH YÊU CẦU

2.1. Khảo sát trường đại học Thái Bình

**Tên cửa trường**: Trường đại học Thái Bình.

**Địa chỉ**: Tân Bình, Thái Bình.

**Hotline**: [0227 3633 134](https://www.google.com/search?q=tbu&oq=tbu&aqs=chrome.0.69i59j46i131i433i512j0i512l3j69i60l3.611j0j7&sourceid=chrome&ie=UTF-8&bshm=nce/1,nce/1).

**Facebook**: https://www.facebook.com/tuyensinh.tbu.



Hình 2.1: Trường đại học Thái Bình

Với rất nhiều năm hoạt động trong lĩnh vực, trường đại học Thái Bình nhận được nhiều lời khen và sự tin tưởng từ hàng ngàn phụ huynh trên khắp khu vực Thái Bình và các tỉnh thành lân cận.

Nhu cầu quản lý hồ sơ online là vấn đề đáng quan tâm của cửa hàng. Vì vậy nhà trường mong muốn xây dựng 1 website quản lý hồ sơ trực tuyến trực tuyến, cũng như hỗ trợ người dùng và giáo viên các thông tin cần thiết.

## 2.2. Yêu cầu của website sau khảo sát

**Phần quản lý trong admin:**

* Admin quản lý toàn bộ hoạt động của cửa hàng.
* Quản lý tài khoản.
* Quản lý nhân viên.
* Khi nhà trường có quyết định gia hạn hợp đồng hay ký kết một hợp đồng hoặc hủy hợp đồng thì người quản trị có thể cập nhật ngay lập tức.
* Khi nhà trường đưa ra kế hoạch đào tạo thì người quản trị là người thêm kế hoạch đó vào.
* Quản lý đơn từ.
* Quản lý lịch sử nâng lương cán bộ.
* Quản lý trình độ chuyên môn.

**Phần giao diện quản lý:**

* Website không nên quá phức tạp.
* Dung lượng file không quá lớn.
* Thanh menu thật đơn giản.
* Phải có thông tin liên hệ.
* Font chữ đơn giản, dễ nhìn, màu sắc hài hòa.
* An toàn và bảo mật dữ liệu.
* Kiểm tra website có tương thích với các trình duyệt phổ biến hay không.
* Hiển thị được các nội dung tới người dùng.
* Cho phép người dùng xem thông tin chi tiết.
* Cho phép giáo viên có thể xuất excel ngay lập tức.
* Cho phép người dùng đăng ký tài khoản để nhận các tin.

+ Cho phép liên hệ với người quản trị.

**Các yêu cầu phi chức năng:**

* Giao diện website dễ sử dụng, trực quan, thân thiện với người dùng.
* Website phải xử lý nhanh chóng và chính xác.
* Tính bảo mật và độ an toàn cao
* Tương thích với đa phần các trình duyệt web hiện tại.

## 2.3. Các chức năng của website

**Trang web quản trị:**

* Quản lý tài khoản.
* Quản lý nhân viên.
* Quản lý phòng ban.
* Quản lý hợp đồng.
* Quản lý kế hoạch đào tạo.
* Quản lý lịch sử nâng lương cán bộ.
* Quản lý hợp đồng.
* Quản lý đơn từ.
* Quản lý trình độ chuyên môn.

## 2.4. Các tác nhân của website

Dựa vào mô tả bài toán, ta có thể xác định được các tác nhân chính của website như sau:

* Tác nhân Admin: người thực hiện các chức năng quản trị website, tài khoản, người dùng, phân quyền cho người dùng, quản lý giao diện, cấu hình Website và có toàn quyền chỉnh sửa cập nhật trên trang Admin. Để thực hiện các chức năng quản trị trong website tác nhân này bắt buộc phải thực hiện đăng nhập.
* Tác nhân giáo viên có các chức năng sau: quản lý tài khoản cá nhân, quản lý đơn từ cá nhân, có thể xem được hợp đồng, phòng ban, kế hoạch đào tạo, xuất excel.
* Tác nhân nhân viên có các chức năng sau: quản lý tài khoản cá nhân, quản lý đơn từ cá nhân, có thể xem được hợp đồng, phòng ban, kế hoạch đào tạo, xuất excel.

# CHƯƠNG 3. PHÂN TÍCH THIẾT KẾ HỆ THỐNG

3.1. Biểu đồ Use-case

Qua khảo sát hiện trạng thực tế, kết hợp với, việc tham khảo tài liệu và phân tích ý kiến của nhà trường, website có các tác nhân sau:

* Biểu đồ Use-case tổng quát

A picture containing diagram, text, drawing, sketch

Description automatically generated

Hình 3.1: Biểu đồ use-case tổng quát

* Biểu đồ Use-case đăng nhập

A picture containing diagram, circle, sketch, line

Description automatically generated

Hình 3.2: Biểu đồ use-case đăng nhập

* Tác nhân: Admin, Người dùng
* Mô tả: Use-case cho admin, người dùng đăng nhập vào website.
* Điều kiện trước: admin đăng nhập vào website.
* Dòng sự kiện chính:
* Chọn chức năng đăng nhập.
* Giao diện đăng nhập hiển thị.
* Nhập Tài khoản, mật khẩu vào giao diện đăng nhập.
* Website kiểm tra tài khoản và mật khẩu nhập của admin. Nếu nhập sai thì chuyển sang dòng sự kiện rẽ nhánh A1. Nếu nhập đúng thì website cấp cho admin 1 token để duy đăng nhập và chuyển tới trang quản trị. Khi token hết hạn thì tài khoản sẽ tự động đăng xuất.
* Use-case kết thúc.
* Dòng sự kiện rẽ nhánh:
* Dòng rẽ nhánh A1: Admin, Người dùng đăng nhập không thành công.
* Website thông báo quá trình đăng nhập không thành công do tài khoản hoặc mật khẩu.
* Chọn nhập lại website yêu cầu nhập lại tài khoản, mật khẩu.
* Use-case kết thúc.
* Kết quả: Admin, Người dùng đăng nhập thành công và có thể sử dụng các chức năng quản lý tương ứng trong trang chủ.
* Biểu đồ use-case quản lý phòng ban
* A picture containing circle, text, diagram, sketch

  Description automatically generated

Hình 3.3: Biểu đồ use-case quản lý phòng ban

* Tác nhân: Admin
* Mô tả: use-case cho phép xem, thêm, sửa, xóa thông tin phòng ban trong website.
* Điều kiện trước: admin đã đăng nhập vào website.
* Dòng sự kiện chính:
* Người sử dụng chọn kiểu tác động: thêm, sửa, xóa thông tin phòng ban.
* Thêm thông tin phòng ban: chọn thêm phòng ban, website hiển thị giao diện nhập thông tin phòng ban, admin nhập thông tin phòng ban, nếu thành công website đưa ra thông báo, nếu sai thực hiện dòng rẽ nhánh A1, lưu thông tin vào danh sách phòng ban.
* Sửa thông tin phòng ban: website hiển thị danh sách phòng ban, chọn phong ban cần sửa, nhập các thông tin cần thay đổi, nếu việc thay đổi thông tin thành công website sẽ đưa ra thông báo, nếu sai thì thực hiện dòng rẽ nhánh A1, lưu thông tin danh sách phòng ban.
* Xóa thông tin phòng ban: website hiển thị danh sách phòng ban, chọn phòng ban cần xóa, nếu việc thay đổi thông tin thành công website sẽ đưa ra thông báo, nếu sai thì thực hiện dòng rẽ nhánh A1, lưu thông tin danh sách phòng ban.
* Use-case kết thúc
* Dòng sự kiện rẽ nhánh A1: website thông báo việc nhập dữ liệu không hợp lệ, nhập lại thông tin, quay lại bước 1 của dòng sự kiện chính.
* Kết quả: các thông tin về tin tức được cập nhật trong cơ sở dữ liệu.
* Biểu đồ use-case quản lý nhân viên.
* A picture containing circle, diagram, sketch, text

  Description automatically generated

Hình 3.4: Biểu đồ use-case quản lý nhân viên

* Tác nhân: Admin
* Mô tả: use-case cho phép xem, thêm, sửa, xóa thông tin nhân viên trong website.
* Điều kiện trước: admin đã đăng nhập vào website.
* Dòng sự kiện chính:
* Người sử dụng chọn kiểu tác động: thêm, sửa, xóa thông tin nhân viên.
* Thêm thông tin nhân viên: chọn thêm nhân viên, website hiển thị giao diện nhập thông tin nhân viên, người sử dụng nhập thông tin nhân viên, nếu thành công website đưa ra thông báo, nếu sai thực hiện dòng rẽ nhánh A1, lưu thông tin vào danh sách nhân viên.
* Sửa thông tin nhân viên: website hiển thị danh sách nhân viên, chọn nhân viên cần sửa, nhập các thông tin cần thay đổi, nếu việc thay đổi thông tin thành công website sẽ đưa ra thông báo, nếu sai thì thực hiện dòng rẽ nhánh A1, lưu thông tin danh sách nhân viên.
* Xóa thông tin nhân viên: website hiển thị danh sách nhân viên, chọn nhân viên cần xóa, nếu việc thay đổi thông tin thành công website sẽ đưa ra thông báo, nếu sai thì thực hiện dòng rẽ nhánh A1, lưu thông tin danh sách nhân viên.
* Use-case kết thúc
* Dòng sự kiện rẽ nhánh A1: website thông báo việc nhập dữ liệu không hợp lệ, nhập lại thông tin, quay lại bước 1 của dòng sự kiện chính.
* Kết quả: các thông tin về nhân viên được cập nhật trong cơ sở dữ liệu.
* Biểu đồ use-case quản lý hợp đồng
* A picture containing circle, diagram, sketch, text

  Description automatically generated

Hình 3.5: Biểu đồ use-case quản lý hợp đồng.

* Tác nhân: Admin
* Mô tả: use-case cho phép xem, thêm, sửa, xóa thông tin hợp đồng trong website.
* Điều kiện trước: admin đã đăng nhập vào website.
* Dòng sự kiện chính:
* Người sử dụng chọn kiểu tác động: thêm, sửa, xóa thông tin hợp đồng.
* Thêm thông tin hợp đồng: chọn thêm hợp đồng, website hiển thị giao diện nhập thông tin hợp đồng, người sử dụng nhập thông tin hợp đồng, nếu thành công website đưa ra thông báo, nếu sai thực hiện dòng rẽ nhánh A1, lưu thông tin vào danh sách hợp đồng.
* Sửa thông tin hợp đồng: website hiển thị danh sách hợp đồng, chọn hợp đồng cần sửa, nhập các thông tin cần thay đổi, nếu việc thay đổi thông tin thành công website sẽ đưa ra thông báo, nếu sai thì thực hiện dòng rẽ nhánh A1, lưu thông tin danh sách hợp đồng.
* Xóa thông tin hợp đồng: website hiển thị danh sách hợp đồng, chọn hợp đồng cần xóa, nếu việc thay đổi thông tin thành công website sẽ đưa ra thông báo, nếu sai thì thực hiện dòng rẽ nhánh A1, lưu thông tin danh sách hợp đồng.
* Use-case kết thúc
* Dòng sự kiện rẽ nhánh A1: website thông báo việc nhập dữ liệu không hợp lệ, nhập lại thông tin, quay lại bước 1 của dòng sự kiện chính.
* Kết quả: các thông tin về hợp đồng được cập nhật trong cơ sở dữ liệu.
* Biểu đồ use-case quản lý kế hoạch đào tạo.
* A picture containing circle, diagram, text, sketch

  Description automatically generated

Hình 3.6: Biểu đồ use-case quản lý kế hoạch đào tạo.

* Tác nhân: Admin
* Mô tả: use-case cho phép xem, thêm, sửa, xóa thông tin kế hoạch đào tạo trong website.
* Điều kiện trước: admin đã đăng nhập vào website.
* Dòng sự kiện chính:
* Người sử dụng chọn kiểu tác động: thêm, sửa, xóa thông tin kế hoạch đào tạo.
* Thêm thông tin kế hoạch đào tạo: chọn thêm kế hoạch đào tạo, website hiển thị giao diện nhập thông tin kế hoạch đào tạo, người sử dụng nhập thông tin kế hoạch đào tạo, nếu thành công website đưa ra thông báo, nếu sai thực hiện dòng rẽ nhánh A1, lưu thông tin vào danh sách kế hoạch đào tạo.
* Sửa thông tin kế hoạch đào tạo: website hiển thị danh sách kế hoạch đào tạo, chọn kế hoạch đào tạo cần sửa, nhập các thông tin cần thay đổi, nếu việc thay đổi thông tin thành công website sẽ đưa ra thông báo, nếu sai thì thực hiện dòng rẽ nhánh A1, lưu thông tin danh sách kế hoạch đào tạo.
* Xóa thông tin kế hoạch đào tạo: website hiển thị danh sách kế hoạch đào tạo, chọn kế hoạch đào tạo cần xóa, nếu việc thay đổi thông tin thành công website sẽ đưa ra thông báo, nếu sai thì thực hiện dòng rẽ nhánh A1, lưu thông tin danh sách kế hoạch đào tạo.
* Use-case kết thúc
* Dòng sự kiện rẽ nhánh A1: website thông báo việc nhập dữ liệu không hợp lệ, nhập lại thông tin, quay lại bước 1 của dòng sự kiện chính.
* Kết quả: các thông tin về kế hoạch đào tạo được cập nhật trong cơ sở dữ liệu.
* Biểu đồ use-case quản lý trình độ chuyên môn
* A diagram of a company

  Description automatically generated with low confidence

Hình 3.7: Biểu đồ use-case quản lý trình độ chuyên môn

* Tác nhân: Admin
* Mô tả: use-case cho phép xem, thêm, sửa, xóa thông tin trình độ chuyên môn trong website.
* Điều kiện trước: admin đã đăng nhập vào website.
* Dòng sự kiện chính:
* Người sử dụng chọn kiểu tác động: thêm, sửa, xóa thông tin trình độ chuyên môn.
* Thêm thông tin trình độ chuyên môn: chọn thêm trình độ chuyên môn, website hiển thị giao diện nhập thông tin trình độ chuyên môn, người sử dụng nhập thông tin trình độ chuyên môn, nếu thành công website đưa ra thông báo, nếu sai thực hiện dòng rẽ nhánh A1, lưu thông tin vào danh sách trình độ chuyên môn.
* Sửa thông tin trình độ chuyên môn: website hiển thị danh sách trình độ chuyên môn, chọn trình độ chuyên môn cần sửa, nhập các thông tin cần thay đổi, nếu việc thay đổi thông tin thành công website sẽ đưa ra thông báo, nếu sai thì thực hiện dòng rẽ nhánh A1, lưu thông tin danh sách trình độ chuyên môn.
* Xóa thông tin trình độ chuyên môn: website hiển thị danh sách trình độ chuyên môn, chọn trình độ chuyên môn cần xóa, nếu việc thay đổi thông tin thành công website sẽ đưa ra thông báo, nếu sai thì thực hiện dòng rẽ nhánh A1, lưu thông tin danh sách trình độ chuyên môn.
* Use-case kết thúc
* Dòng sự kiện rẽ nhánh A1: website thông báo việc nhập dữ liệu không hợp lệ, nhập lại thông tin, quay lại bước 1 của dòng sự kiện chính.
* Kết quả: các thông tin về trình độ chuyên môn được cập nhật trong cơ sở dữ liệu.
* Biểu đồ use-case quản lý đơn từ
* A picture containing circle, diagram, text, sketch

  Description automatically generated

Hình 3.8: Biểu đồ use-case quản lý đơn từ

* Tác nhân: Admin
* Mô tả: use-case cho phép xem, thêm, sửa, xóa, tìm kiếm thông tin đơn từ trong website.
* Điều kiện trước: admin đã đăng nhập vào website.
* Dòng sự kiện chính:
* Người sử dụng chọn kiểu tác động: thêm, sửa, xóa thông tin đơn từ.
* Thêm thông tin đơn từ: chọn thêm đơn từ, website hiển thị giao diện nhập thông tin đơn từ, người sử dụng nhập thông tin đơn từ, nếu thành công website đưa ra thông báo, nếu sai thực hiện dòng rẽ nhánh A1, lưu thông tin vào danh sách đơn từ.
* Sửa thông tin đơn từ: website hiển thị danh sách đơn từ, chọn đơn từ cần sửa, nhập các thông tin cần thay đổi, nếu việc thay đổi thông tin thành công website sẽ đưa ra thông báo, nếu sai thì thực hiện dòng rẽ nhánh A1, lưu thông tin danh sách đơn từ.
* Xóa thông tin đơn từ: website hiển thị danh sách đơn từ, chọn đơn từ cần xóa, nếu việc thay đổi thông tin thành công website sẽ đưa ra thông báo, nếu sai thì thực hiện dòng rẽ nhánh A1, lưu thông tin danh sách đơn từ.
* Use-case kết thúc
* Dòng sự kiện rẽ nhánh A1: website thông báo việc nhập dữ liệu không hợp lệ, nhập lại thông tin, quay lại bước 1 của dòng sự kiện chính.
* Kết quả: các thông tin về đơn từ được cập nhật trong cơ sở dữ liệu.
* Biểu đồ use-case quản lý lịch sử nâng lương cán bộ
* A picture containing text, sketch, diagram, circle

  Description automatically generated

Hình 3.9: Biểu đồ use-case quản lý lịch sử nâng lương cán bộ

* Tác nhân: Admin
* Mô tả: use-case cho phép xem, thêm, sửa, xóa thông tin lịch sử nâng lương cán bộ trong website.
* Điều kiện trước: admin đã đăng nhập vào website.
* Dòng sự kiện chính:
* Người sử dụng chọn kiểu tác động: thêm, sửa, xóa thông tin lịch sử nâng lương cán bộ.
* Thêm thông tin lịch sử nâng lương cán bộ: chọn thêm lịch sử nâng lương cán bộ, website hiển thị giao diện nhập thông tin lịch sử nâng lương cán bộ, người sử dụng nhập thông tin lịch sử nâng lương cán bộ, nếu thành công website đưa ra thông báo, nếu sai thực hiện dòng rẽ nhánh A1, lưu thông tin vào danh sách lịch sử nâng lương cán bộ.
* Sửa thông tin lịch sử nâng lương cán bộ: website hiển thị danh sách lịch sử nâng lương cán bộ, chọn lịch sử nâng lương cán bộ cần sửa, nhập các thông tin cần thay đổi, nếu việc thay đổi thông tin thành công website sẽ đưa ra thông báo, nếu sai thì thực hiện dòng rẽ nhánh A1, lưu thông tin danh sách lịch sử nâng lương cán bộ.
* Xóa thông tin lịch sử nâng lương cán bộ: website hiển thị danh sách lịch sử nâng lương cán bộ, chọn lịch sử nâng lương cán bộ cần xóa, nếu việc thay đổi thông tin thành công website sẽ đưa ra thông báo, nếu sai thì thực hiện dòng rẽ nhánh A1, lưu thông tin danh sách lịch sử nâng lương cán bộ.
* Use-case kết thúc
* Dòng sự kiện rẽ nhánh A1: website thông báo việc nhập dữ liệu không hợp lệ, nhập lại thông tin, quay lại bước 1 của dòng sự kiện chính.
* Kết quả: các thông tin về lịch sử nâng lương cán bộ được cập nhật trong cơ sở dữ liệu.
* Biểu đồ use-case giáo viên
* A picture containing diagram, drawing, sketch, circle

  Description automatically generated

Hình 3.10: Biểu đồ use-case giáo viên

* Tác nhân: Teacher
* Mô tả: use-case cho phép xem hợp đồng, kế hoạch đào tạo, đơn từ, cho phép quản lý đơn từ cá nhân: xem, thêm, sửa, xóa đơn từ cá nhân; và quản lý thông tin: xem, sửa thông tin trong website.
* Điều kiện trước: teacher đã đăng nhập vào website.
* Dòng sự kiện chính:
* Người sử dụng chọn kiểu tác động: thêm, sửa, xóa thông tin đơn từ cá nhân.
* Thêm đơn từ cá nhân: chọn thêm đơn từ cá nhân, website hiển thị giao diện nhập thông tin đơn từ cá nhân, người sử dụng nhập thông tin đơn từ cá nhân, nếu thành công website đưa ra thông báo, nếu sai thực hiện dòng rẽ nhánh A1, lưu thông tin vào danh sách đơn từ cá nhân.
* Sửa thông tin đơn từ cá nhân: website hiển thị danh sách đơn từ cá nhân, chọn đơn từ cá nhân cần sửa, nhập các thông tin cần thay đổi, nếu việc thay đổi thông tin thành công website sẽ đưa ra thông báo, nếu sai thì thực hiện dòng rẽ nhánh A1, lưu thông tin danh sách đơn từ cá nhân.
* Xóa thông tin đơn từ cá nhân: website hiển thị danh sách đơn từ cá nhân, chọn đơn từ cá nhân cần xóa, nếu việc thay đổi thông tin thành công website sẽ đưa ra thông báo, nếu sai thì thực hiện dòng rẽ nhánh A1, lưu thông tin danh sách đơn từ cá nhân.
* Use-case kết thúc
* Dòng sự kiện rẽ nhánh A1: website thông báo việc nhập dữ liệu không hợp lệ, nhập lại thông tin, quay lại bước 1 của dòng sự kiện chính.
* Kết quả: các thông tin về đơn từ cá nhân được cập nhật trong cơ sở dữ liệu.
* Biểu đồ use-case nhân viên
* A picture containing diagram, circle, drawing, sketch

  Description automatically generated

Hình 3.11: Biểu đồ use-case nhân viên

* Tác nhân: Staff
* Mô tả: use-case cho phép xem hợp đồng, kế hoạch đào tạo, đơn từ, cho phép quản lý đơn từ cá nhân: xem, thêm, sửa, xóa đơn từ cá nhân; và quản lý thông tin: xem, sửa thông tin trong website.
* Điều kiện trước: staff đã đăng nhập vào website.
* Dòng sự kiện chính:
* Người sử dụng chọn kiểu tác động: thêm, sửa, xóa thông tin đơn từ cá nhân.
* Thêm đơn từ cá nhân: chọn thêm đơn từ cá nhân, website hiển thị giao diện nhập thông tin đơn từ cá nhân, người sử dụng nhập thông tin đơn từ cá nhân, nếu thành công website đưa ra thông báo, nếu sai thực hiện dòng rẽ nhánh A1, lưu thông tin vào danh sách đơn từ cá nhân.
* Sửa thông tin đơn từ cá nhân: website hiển thị danh sách đơn từ cá nhân, chọn đơn từ cá nhân cần sửa, nhập các thông tin cần thay đổi, nếu việc thay đổi thông tin thành công website sẽ đưa ra thông báo, nếu sai thì thực hiện dòng rẽ nhánh A1, lưu thông tin danh sách đơn từ cá nhân.
* Xóa thông tin đơn từ cá nhân: website hiển thị danh sách đơn từ cá nhân, chọn đơn từ cá nhân cần xóa, nếu việc thay đổi thông tin thành công website sẽ đưa ra thông báo, nếu sai thì thực hiện dòng rẽ nhánh A1, lưu thông tin danh sách đơn từ cá nhân.
* Use-case kết thúc
* Dòng sự kiện rẽ nhánh A1: website thông báo việc nhập dữ liệu không hợp lệ, nhập lại thông tin, quay lại bước 1 của dòng sự kiện chính.
* Kết quả: các thông tin về đơn từ cá nhân được cập nhật trong cơ sở dữ liệu.

* 3.2. Biểu đồ tuần tự
* Biểu đồ tuần tự chức năng đăng nhập

A picture containing text, diagram, technical drawing, plan

Description automatically generated

Hình 3.12: Biều đồ tuần tự chức năng đăng nhập

* Biểu đồ tuần tự thêm phòng ban

A diagram of a flowchart

Description automatically generated with low confidence

Hình 3.13: Biểu đồ tuần tự chức năng thêm phòng ban

* Biểu đồ tuần tự sửa phòng ban

A picture containing text, diagram, parallel, line

Description automatically generated

Hình 3.14: Biểu đồ tuần tự sửa phòng ban

3.3. Biểu đồ hoạt động

* Biểu đồ hoạt động xóa phòng ban

A diagram of a flowchart

Description automatically generated with low confidence

Hình 3.15: Biểu đồ hoạt động xóa phòng ban

## 3.4. Biểu đồ lớp

* Biểu đồ lớp

A picture containing text, font, parallel, diagram

Description automatically generated

*Hình 3.16: Biểu đồ lớp*

* Mô hình quan hệ dữ liệu (diagram)

A screenshot of a computer

Description automatically generated with medium confidence

Hình 3.17: Mô hình quan hệ dữ liệu diagram

* Danh sách lớp đối tượng

| **STT** | **Tên thực thể** | **Mô tả** |
| --- | --- | --- |
| 1 | Users | Mô tả các thuộc tính và các phương thức liên quan tới tài khoản người dùng |
| 2 | Teacher | Mô tả các thuộc tính và các phương thức liên quan tới giáo viên |
| 3 | Staff | Mô tả các thuộc tính và các phương thức liên quan tới nhân viên |
| 4 | LeaveRequest | Mô tả các thuộc tính và các phương thức liên quan tới đơn từ |
| 5 | Employee | Mô tả các thuộc tính và các phương thức liên quan tới nghiệp vụ quản lý thông tin người dùng |
| 6 | Plan | Mô tả các thuộc tính và các phương thức liên quan tới các kế hoạch đào tạo |
| 7 | Contract | Mô tả các thuộc tính và các phương thức liên quan tới hợp đồng |
| 8 | Qualification | Mô tả các thuộc tính và các phương thức liên quan tới trình độ chuyên môn |
| 9 | Promotion | Mô tả các thuộc tính và các phương thức liên quan tới lịch sử nâng lương cán bộ |
| 10 | Department | Mô tả các thuộc tính và các phương thức liên quan tới phòng ban |

Bảng 3.1: Bảng danh sách các đối tượng

## 3.4. Thiết kế cơ sở dữ liệu

* Bảng Staff

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Mô tả |
| 1 | staffID | int(11) | Khóa chính, tự tăng |
| 2 | employeeID | int(11) | Khóa phụ |
| 3 | role | varchar(255) | Vai trò |
| 4 | createAt | datetime | Ngày tạo |
| 5 | updateAt | datetime | Ngày chỉnh sửa |

Bảng 3.2:Bảng Staff

* Bảng Plan

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Mô tả |
| 1 | planID | int(11) | Khóa chính, tự tăng |
| 2 | employeeID | int(11) | Khóa phụ |
| 3 | position | varchar(255) | Vị trí |
| 4 | upcomingPosition | varchar(255) | Vị trí sắp tới |
| 5 | trainingCourse | varchar(255) | Khóa huấn luyện |
| 6 | startDate | datetime | Ngày bắt đầu |
| 7 | endDate | datetime | Ngày kết thúc |
| 8 | createAt | datetime | Ngày tạo |
| 9 | updateAt | datetime | Ngày chỉnh sửa |

Bảng 3.3: Bảng Plan

* Bảng User

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Mô tả |
| 1 | userID | int | Khóa chính |
| 2 | username | varchar(255) | Tên người dùng |
| 3 | password | varchar(255) | Mật khẩu |
| 4 | roleid | varchar(255) | Mô tả quyền |
| 5 | accountTypeName | varchar(255) | Tên loại tài khoản |
| 6 | image | varchar(255) | Ảnh |
| 7 | createAt | datetime | Ngày tạo |
| 8 | updateAt | datetime | Ngày chỉnh sửa |

Bảng 3.4: Bảng User

* Bảng LeaveRequest

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Mô tả |
| 1 | leaveRequestID | int(11) | Khóa chính, tự tăng |
| 2 | employeeID | int(11) | Khóa phụ |
| 3 | requetsDate | datetime | Ngày yêu cầu |
| 4 | leaveType | varchar(255) | Loại nghỉ phép |
| 5 | startDate | datetime | Ngày bắt đầu |
| 6 | endDate | datetime | Ngày kết thúc |
| 7 | createAt | datetime | Ngày tạo |
| 8 | updateAt | datetime | Ngày chỉnh sửa |

Bảng 3.5: Bảng LeaveRequest

* Bảng Sequelizemeta

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Mô tả |
| 1 | name | varchar(255) | Tên |

Bảng 3.6: Bảng Sequelizemeta

* Bảng Employee

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Mô tả |
| 1 | employeeID | int | Khóa chính, tự tăng |
| 2 | employee\_userID | int | Mã tài khoản |
| 3 | employeeName | varchar(255) | Tên người dùng |
| 4 | dob | datetime | Ngày sinh |
| 5 | gender | tinyint(1) | Giới tính |
| 6 | addrees | varchar(255) | Địa chỉ |
| 7 | phoneNum | varchar(255) | Số điện thoại |
| 8 | email | varchar(255) | Email |
| 9 | joiningDate | varchar(255) | Ngày tham gia |
| 10 | type | varchar(255) | Loại |
| 11 | salary | varchar(255) | Lương |
| 12 | insuranceContribution | int(11) | Bảo hiểm |
| 13 | isHighLevelTeacher | tinyint(1) | Trình độ |
| 14 | isActive | tinyint(1) | Nhân viên này đã nghỉ hay chưa |
| 15 | createAt | datetime | Ngày tạo |
| 16 | updateAt | datetime | Ngày chỉnh sửa |

Bảng 3.7:Bảng Employee

* Bảng Department

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Mô tả |
| 1 | departmentID | int(11) | Khóa chính, tự tăng |
| 2 | departmentName | varchar(255) | Tên phòng ban |
| 3 | addrees | varchar(255) | Địa chỉ |
| 4 | phoneNumber | varchar(255) | Số điện thoại |
| 5 | createAt | datetime | Ngày tạo |
| 6 | updateAt | datetime | Ngày chỉnh sửa |

Bảng 3.8: Bảng Department

* Bảng Teacher

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Mô tả |
| 1 | teacherID | int(11) | Khóa chính, tự tăng |
| 2 | employeeID | int(11) | Khóa phụ |
| 3 | departmentID | varchar(255) | Khóa phụ |
| 4 | courseTaught | datetime | Khóa học đã dạy |
| 5 | createAt | datetime | Ngày tạo |
| 6 | updateAt | datetime | Ngày chỉnh sửa |

Bảng 3.9: Bảng Teacher

* Bảng Promotion

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Mô tả |
| 1 | promotionID | int(11) | Khóa chính, tự tăng |
| 2 | employeeID | int(11) | Khóa phụ |
| 3 | archievement | varchar(255) | Thành tích |
| 4 | degree | text | Bằng cấp |
| 5 | actualDate | datetime | Ngày thực tế |
| 6 | realDate | datetime | Ngày thật |
| 7 | createAt | datetime | Ngày tạo |
| 8 | updateAt | datetime | Ngày chỉnh sửa |

Bảng 3.10: Bảng Promotion

* Bảng Qualification

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Mô tả |
| 1 | qualificationID | int(11) | Khóa chính, tự tăng |
| 2 | employeeID | int(11) | Khóa phụ |
| 3 | degree | varchar(255) | Bằng cấp |
| 3 | field | varchar(255) | Chuyên ngành |
| 4 | institution | varchar(255) | Đại học |
| 5 | yearCompleted | varchar(255) | Năm hoàn thành |
| 6 | createAt | datetime | Ngày tạo |
| 7 | updateAt | datetime | Ngày chỉnh sửa |

Bảng 3. 11: Bảng Qualification

* Bảng Contract

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Mô tả |
| 1 | contractID | int(11) | Khóa chính, tự tăng |
| 2 | employeeID | int(11) | Khóa phụ |
| 3 | contractType | varchar(255) | Loại hợp đồng |
| 4 | startDate | datetime | Ngày bắt đầu |
| 5 | endDate | datetime | Ngày kết thúc |
| 6 | createAt | datetime | Ngày tạo |
| 7 | updateAt | datetime | Ngày chỉnh sửa |

Bảng 3. 12: Bảng Contract

# CHƯƠNG 4. THIẾT KẾ GIAO DIỆN WEBSITE

* 1. Giao diện trang quản lý tài khoản

A screenshot of a computer

Description automatically generated

Hình 4.1: Giao diện trang quản lý tài khoản

* 1. Giao diện trang quản lý phòng ban

A screenshot of a computer

Description automatically generated

Hình 4.2: Giao diện trang quản lý phòng ban

* 1. Giao diện trang quản lý nhân viên

A screenshot of a computer

Description automatically generated

Hình 4.3: Giao diện trang quản lý nhân viên

* 1. Giao diện trang khi xóa nhân viên

A screenshot of a computer

Description automatically generated

Hình 4.4: Giao diện trang khi xóa nhân viên

* 1. Giao diện trang quản lý hợp đồng

A screenshot of a computer

Description automatically generated

Hình 4.5: Giao diện trang quản lý hợp đồng

* 1. Giao diện trang quản lý kế hoạch đào tạo

A picture containing text, screenshot, software, computer icon

Description automatically generated

Hình 4.6: Giao diện trang quản lý kế hoạch đào tạo

* 1. . Giao diện trang quản lý trình độ chuyên môn

A screenshot of a computer

Description automatically generated

Hình 4.7: Giao diện trang quản lý trình độ chuyên môn

* 1. Giao điện trang quản lý đơn từ

A screenshot of a computer

Description automatically generated

Hình 4.8: Giao diện trang quản lý đơn từ

* 1. Giao diện trang quản lý lịch sử nâng lương cán bộ

A picture containing text, screenshot, software, computer icon

Description automatically generated

Hình 4.9: Giao diện trang quản lý lịch sử nâng lương cán bộ

# CHƯƠNG 5. CÀI ĐẶT MỘT SỐ CHỨC NĂNG CHO WEBISTE

## 5.1. Chức năng hợp đồng trên server

* RestfulAPI được thiết kế trong Controller/Contract

//Import Model Employee từ models đồng thời sử dung Op từ thư viện Sequelize

const { Contract } = require("../models");

const { Employee } = require("../models");

const { Op } = require("sequelize");

// API tạo contract mới

const create = async (req, res) => {

try {

const { employeeID, contractType, startDate, endDate } = req.body;

await Contract.create({

employeeID,

contractType,

startDate,

endDate,

});

res.status(201).json({ message: "Contract created" });

} catch (error) {

console.error(error);

res.status(500).json({ message: "Internal server error", error: error });

}

};

//API cập nhật contract của nhân viên

const put = async (req, res) => {

try {

const { contractID, employeeID, contractType, startDate, endDate } =

req.body;

const contract = await Contract.findByPk(contractID);

if (!contract) {

return res.status(404).send("Contract not found");

}

await Contract.update({

contractID,

employeeID,

contractType,

startDate,

endDate,

});

res.status(201).json({ message: "Contract edited successfully" });

} catch (error) {

console.error(error);

res.status(500).json({ message: "Internal server error", error: error });

}

};

//API xóa contract của nhân viên

const deleteContract =

("/",

async (req, res) => {

try {

const { contractID } = req.query;

if (!contractID) {

return res

.status(400)

.json({ message: "contractID parameter is required" });

}

const numDeleted = await Contract.destroy({ where: { contractID } });

if (numDeleted === 0) {

return res.status(404).json({

message: `No Contract with contractID=${contractID} was found`,

});}

res.json({

message: `Contract with contractID=${contractID} was deleted successfully`,

});

} catch (error) {

console.error(error);

res.status(500).json({ message: "Internal server error", contractID });

}});

//API lấy về danh sách toàn bộ các hợp đồng trong hệ thống

const getAll = async (req, res) => {

try {

const contracts = await Contract.findAll({

include: [{ model: Employee }],

});

res.json(contracts);

} catch (error) {

console.error(error);

res.status(500).json({ message: "Internal server error", error: error });

}};

//API lấy về thông tin cụ thể của một hợp đồng dựa trên ID của hợp đồng đó

const getContractByID =

("/",

async (req, res) => {

try {

const { contractID } = req.query;

if (!contractID) {

return res

.status(400)

.json({ message: "contractID parameter is required" });

}

const whereClause = {};

if (contractID) {

whereClause.contractID = {

[Op.like]: `%${contractID}%`,

};

}

const contract = await Contract.findAll({

where: whereClause,

include: [{ model: Employee }],

});

res.json(contract);

} catch (error) {

console.error(error);

res.status(500).json({ message: "Internal server error", error: error });

}});

//Xuất ra module các API trên

module.exports = {

create,

put,

deleteContract,

getAll,

getContractByID,

};

## 5.2. Chức năng phòng ban trên server

//Import mô hình Department từ models

const { Department } = require("../models");

const { Op } = require("sequelize");

//API tạo mới phòng ban - Department

const create = async (req, res) => {

try {

const { departmentName, address, phoneNum } = req.body;

await Department.create({

departmentName,

address,

phoneNum,

});

res.status(201).json({ message: "Department created" });

} catch (error) {

console.error(error);

res.status(500).json({ message: "Internal server error", error: error });

}};

//API cập nhật thông tin phòng ban

const put = async (req, res) => {

try {

const { departmentID, departmentName, address, phoneNum } = req.body;

const department = await Department.findByPk(departmentID);

if (!department) {

return res.status(404).send("Department not found");

}

await Department.update({

departmentID,

departmentName,

address,

phoneNum,

});

res.status(201).json({ message: "Department edited successfully" });

} catch (error) {

console.error(error);

res.status(500).json({ message: "Internal server error", error: error });

}};

//API xóa phòng ban dựa trên ID

const deleteDepartment =

("/",

async (req, res) => {

try {

const { departmentID } = req.query;

if (!departmentID) {

return res

.status(400)

.json({ message: "departmentID parameter is required" });

}

const numDeleted = await Department.destroy({ where: { departmentID } });

if (numDeleted === 0) {

return res.status(404).json({

message: `No Department with departmentID=${departmentID} was found`,

});}

res.json({

message: `Department with departmentID=${departmentID} was deleted successfully`,

});

} catch (error) {

console.error(error);

res.status(500).json({ message: "Internal server error", departmentID });

}});

//API lấy về danh sách toàn bộ phòng ban

const getAll = async (req, res) => {

try {

const departments = await Department.findAll();

res.json(departments);

} catch (error) {

console.error(error);

res.status(500).json({ message: "Internal server error", error: error });

}

};

//API lấy về chi tiết thông tin 1 phòng ban

const getDepartmentByID =

("/",

async (req, res) => {

try {

const { departmentID } = req.query;

if (!departmentID) {

return res

.status(400)

.json({ message: "departmentID parameter is required" });

}

const whereClause = {};

if (departmentID) {

whereClause.departmentID = {

[Op.like]: `%${departmentID}%`,

};

}

const department = await Department.findAll({ where: whereClause });

res.json(department);

} catch (error) {

console.error(error);

res.status(500).json({ message: "Internal server error", error: error });

}

});

//Xuất ra các module trên

module.exports = {

create,

put,

deleteDepartment,

getAll,

getDepartmentByID,

};

## 5.3. Chức năng định tuyến từ máy client lên server

//Import tất cả các module API cần thiết, các middleware và 1 số chức năng khác

import express from "express";

import homeController from "../controller/homeController";

import userController from "../controller/userController";

import departmentController from "../controller/departmentController";

import contractController from "../controller/contractController";

import employeeController from "../controller/employeeController";

import leaveRequestController from "../controller/leaveRequestController";

import planController from "../controller/planController";

import promotionController from "../controller/promotionController";

import qualificationController from "../controller/qualificationController";

import staffController from "../controller/staffController";

import teacherController from "../controller/teacherController";

import authToken from '../middleware/authToken';

const cors = require('cors');

const upload = require('../middleware/multer');

let router = express.Router();

//khởi tạo đường dẫn cho các API trên

let initWebRoutes = (app) => {

app.use(cors());

//upload file

router.post("/api/uploadFile", authToken.verifyUserToken, upload.single('file'), homeController.uploadImage);

//user

router.post("/api/register", upload.single('image'), userController.register);

router.post("/api/login", userController.login);

router.post("/api/changepassword", authToken.verifyUserToken, userController.changePassword);

router.get("/api/getAllUser", authToken.verifyUserToken, userController.getAll);

router.delete("/api/user/delete", authToken.verifyUserToken, userController.deleteUser);

//department

router.post("/api/department/create", authToken.verifyUserToken ,departmentController.create);

router.put("/api/department/edit", authToken.verifyUserToken, departmentController.put);

router.delete("/api/department/delete", authToken.verifyUserToken, departmentController.deleteDepartment);

router.get("/api/department/getAll", authToken.verifyUserToken, departmentController.getAll);

router.get("/api/department/getDepartment", authToken.verifyUserToken, departmentController.getDepartmentByID);

//employee

router.post("/api/employee/create", authToken.verifyUserToken ,employeeController.create);

router.put("/api/employee/edit", authToken.verifyUserToken, employeeController.put);

router.delete("/api/employee/delete", authToken.verifyUserToken, employeeController.deleteEmployee);

router.get("/api/employee/getAll", authToken.verifyUserToken, employeeController.getAll);

router.get("/api/employee/getEmployee", authToken.verifyUserToken, employeeController.getEmployeeByID);

router.get("/api/employee/getActiveEmployee", authToken.verifyUserToken, employeeController.getAllActive);

router.put("/api/employee/deActive", authToken.verifyUserToken, employeeController.deActive);

//contract

router.post("/api/contract/create", authToken.verifyUserToken ,contractController.create);

router.put("/api/contract/edit", authToken.verifyUserToken, contractController.put);

router.delete("/api/contract/delete", authToken.verifyUserToken, contractController.deleteContract);

router.get("/api/contract/getAll", authToken.verifyUserToken, contractController.getAll);

router.get("/api/contract/getContract", authToken.verifyUserToken, contractController.getContractByID);

//staff

router.post("/api/staff/create", authToken.verifyUserToken ,staffController.create);

router.put("/api/staff/edit", authToken.verifyUserToken, staffController.put);

router.delete("/api/staff/delete", authToken.verifyUserToken, staffController.deleteStaff);

router.get("/api/staff/getAll", authToken.verifyUserToken, staffController.getAll);

router.get("/api/staff/getStaff", authToken.verifyUserToken, staffController.getStaffByID);

//plan

router.post("/api/plan/create", authToken.verifyUserToken ,planController.create);

router.put("/api/plan/edit", authToken.verifyUserToken, planController.put);

router.delete("/api/plan/delete", authToken.verifyUserToken, planController.deletePlan);

router.get("/api/plan/getAll", authToken.verifyUserToken, planController.getAll);

router.get("/api/plan/getPlan", authToken.verifyUserToken, planController.getPlanByID);

//qualification

router.post("/api/qualification/create", authToken.verifyUserToken ,qualificationController.create);

router.put("/api/qualification/edit", authToken.verifyUserToken, qualificationController.put);

router.delete("/api/qualification/delete", authToken.verifyUserToken, qualificationController.deleteQualification);

router.get("/api/qualification/getAll", authToken.verifyUserToken, qualificationController.getAll);

router.get("/api/qualification/getQualification", authToken.verifyUserToken, qualificationController.getQualificationByID);

//leaveRequest

router.post("/api/leaveRequest/create", authToken.verifyUserToken ,leaveRequestController.create);

router.put("/api/leaveRequest/edit", authToken.verifyUserToken, leaveRequestController.put);

router.delete("/api/leaveRequest/delete", authToken.verifyUserToken, leaveRequestController.deleteLeaveRequest);

router.get("/api/leaveRequest/getAll", authToken.verifyUserToken, leaveRequestController.getAll);

router.get("/api/leaveRequest/getLeaveRequest", authToken.verifyUserToken, leaveRequestController.getLeaveRequestByID);

// promotion

router.post("/api/promotion/create", authToken.verifyUserToken ,promotionController.create);

router.put("/api/promotion/edit", authToken.verifyUserToken, promotionController.put);

router.delete("/api/promotion/delete", authToken.verifyUserToken, promotionController.deletePromotion);

router.get("/api/promotion/getAll", authToken.verifyUserToken, promotionController.getAll);

router.get("/api/promotion/getPromotion", authToken.verifyUserToken, promotionController.getPromotionByID);

// teacher

router.post("/api/teacher/create", authToken.verifyUserToken ,teacherController.create);

router.put("/api/teacher/edit", authToken.verifyUserToken, teacherController.put);

router.delete("/api/teacher/delete", authToken.verifyUserToken, teacherController.deleteTeacher);

router.get("/api/teacher/getAll", authToken.verifyUserToken, teacherController.getAll);

router.get("/api/teacher/getTeacher", authToken.verifyUserToken, teacherController.getTeacherByID);

return app.use("/", router);

};

module.exports = initWebRoutes;

## 5.4. Chức năng điều hướng trên website

import Vue from 'vue'

import VueRouter from 'vue-router'

Vue.use(VueRouter)

const routes = [

{

path: '/',

name: 'main',

meta: { title: 'Main' },

redirect: '/home',

component: () => import('@/view/Main.vue'),

children: [

{

path: '/home',

name: 'home',

meta: { title: 'Home' },

component: () => import('@/components/Dashboard/Dashboard.vue'),

children: [],

},

{

path: '/check-your-email',

name: 'check-your-email',

meta: { title: 'Check Your Email - CEC App' },

component: () => import('@/view/CheckYourEmail.vue'),

},

{

path: '/version',

name: 'version-view',

meta: { title: 'Version' },

component: () => import('@/components/ListVersion/ListVersion.vue'),

children: [],

},

{

path: '/my-profile',

name: 'my-profile',

meta: { title: 'My Profile' },

component: () => import('@/components/MyProfile/MyProfile.vue'),

children: [],

},

{

path: '/my-profile/edit/:userId',

name: 'my-profile-edit',

meta: { title: 'My Profile - Edit' },

component: () => import('@/components/MyProfile/Edit.vue'),

children: [],

},

{

path: '/my-profile/change-password',

name: 'my-profile-change-password',

meta: { title: 'My Profile - Change Password' },

component: () => import('@/components/MyProfile/ChangePassword.vue'),

children: [],

},

{

path: '/my-profile/reset-password',

name: 'my-profile-reset-password',

meta: { title: 'My Profile - Reset Password' },

component: () => import('@/components/MyProfile/ResetPassword.vue'),

children: [],

},

{

path: '/account-management',

name: 'account-management',

meta: { title: 'Account Management' },

redirect: '/account-management/add-new',

component: () => import('@/view/Sub.vue'),

children: [

{

path: '/account-management/overview',

name: 'account-management-overview',

meta: { title: 'Account Management - Over View' },

component: () => import('@/components/AccountManagement/OverView.vue'),

children: [],

},

{

path: '/account-management/add-new',

name: 'account-management-add-new',

meta: { title: 'Account Management - Add New' },

component: () => import('@/components/AccountManagement/AddNew.vue'),

children: [],

},

],

},

{

path: '/department-management',

name: 'department-management',

meta: { title: 'Department Management' },

redirect: '/department-management/overview',

component: () => import('@/view/Sub.vue'),

children: [

{

path: '/department-management/overview',

name: 'department-management-overview',

meta: { title: 'Department Management - Over View' },

component: () => import('@/components/DepartmentManagement/Overview.vue'),

children: [],

},

],

},

{

path: '/employee-management',

name: 'employee-management',

meta: { title: 'Employee Management' },

redirect: '/employee-management/overview',

component: () => import('@/view/Sub.vue'),

children: [

{

path: '/employee-management/overview',

name: 'employee-management-overview',

meta: { title: 'Employee Management - Over View' },

component: () => import('@/components/EmployeeManagement/Overview.vue'),

children: [],

},

{

path: '/employee-management/add-new',

name: 'employee-add-new',

meta: { title: 'Employee Management - Add New' },

component: () => import('@/components/EmployeeManagement/AddNew.vue'),

children: [],

},

],

},

{

path: '/contract-management',

name: 'contract-management',

meta: { title: 'Contract Management' },

redirect: '/contract-management/add-new',

component: () => import('@/view/Sub.vue'),

children: [

{

path: '/contract-management/overview',

name: 'contract-management-overview',

meta: { title: 'Contract Management - Over View' },

component: () => import('@/components/ContractManagement/Overview.vue'),

children: [],

},

],

},

{

path: '/plan-management',

name: 'plan-management',

meta: { title: 'Plan Management' },

redirect: '/plan-management/add-new',

component: () => import('@/view/Sub.vue'),

children: [

{

path: '/plan-management/overview',

name: 'plan-management-overview',

meta: { title: 'Plan Management - Over View' },

component: () => import('@/components/PlanManagement/Overview.vue'),

children: [],

},

],

},

{

path: '/qualification-management',

name: 'qualification-management',

meta: { title: 'Qualification Management' },

redirect: '/qualification-management/add-new',

component: () => import('@/view/Sub.vue'),

children: [

{

path: '/qualification-management/overview',

name: 'qualification-management-overview',

meta: { title: 'Qualification Management - Over View' },

component: () => import('@/components/QualificationManagement/Overview.vue'),

children: [],

},

],

},

{

path: '/leave-request-management',

name: 'leave-request-management',

meta: { title: 'Leave Request Management' },

redirect: '/leave-request-management/add-new',

component: () => import('@/view/Sub.vue'),

children: [

{

path: '/leave-request-management/overview',

name: 'leave-request-management-overview',

meta: { title: 'Leave Request Management - Over View' },

component: () => import('@/components/LeaveRequestManagement/Overview.vue'),

children: [],

},

],

},

{

path: '/promotion-management',

name: 'promotion-management',

meta: { title: 'Promotion Management' },

redirect: '/promotion-management/add-new',

component: () => import('@/view/Sub.vue'),

children: [

{

path: '/promotion-management/overview',

name: 'promotion-management-overview',

meta: { title: 'Promotion Management - Over View' },

component: () => import('@/components/PromotionManagement/Overview.vue'),

children: [],

},

],

},

],

},

{

path: '/login',

name: 'login',

meta: { title: 'Login - Vue App' },

component: () => import('@/view/Login.vue'),

},

]

const router = new VueRouter({

mode: 'history',

base: process.env.BASE\_URL,

routes,

})

export default router

const DEFAULT\_TITLE = 'Hệ thống quản lý hồ sơ'

router.beforeEach((to, from, next) => {

Vue.nextTick(() => {

document.title = DEFAULT\_TITLE

})

const publicPages = ['/login']

const authRequired = !publicPages.includes(to.path)

const currentUser = JSON.parse(localStorage.getItem('currentUser')) || {}

if (to.path === '/home') {

if (!authRequired && !currentUser) {

next('/login')

} else {

next()

}

} else {

next()

}

})

# CHƯƠNG 6. KẾT LUẬN

6.1. Kết quả đạt được của đề tài

Sau một thời gian tìm hiểu và nghiên cứu đề tài “Thiết kế và xây dựng Website quản lý hồ sơ cán bộ, giảng viên, viên chức và người lao động của trường Đại học Thái Bình” đề tài đã đạt được một số kết quả sau:

* Thông qua nghiệp vụ kết hợp với mục đích thực hiện tin học hóa, mô tả chức năng của từng website, xây dựng các biểu đồ UML và cơ sở dữ liệu thông qua công cụ draw.io.
* Xây dựng chương trình thông qua kết quả thu thập được qua quá trình khảo sát, phân tích website và đặc tả giao diện

Ngoài những kết quả trên, qua quá trình thực hiện đề tài “Thiết kế và xây dựng Website quản lý hồ sơ cán bộ, giảng viên, viên chức và người lao động của trường Đại học Thái Bình” em đã có được:

* Những kinh nghiệm quý báu về quy trình và nghiệp vụ quản lý website quản lý.
* Trau dồi kiến thức về VueJS, NodeJS và MVC Framework.
* Rèn luyện kỹ năng làm việc cá nhân, khả năng tư duy và sáng tạo.

6.2. Hạn chế của đề tài

Do thời gian nghiên cứu chưa được nhiều, bên cạnh những kết quả đạt được còn tồn tại một số vấn đề hạn chế:

* Hiển thị trong trang chủ chưa được tối ưu hóa nhất.
* Chưa triển khai website thực tế được.

6.3. Hướng phát triển của đề tài

Với mục đích ngày càng hoàn thiện để đáp ứng tốt nhất cho việc quản lý online cho trường đại học Thái Bình. Cài đặt website hoàn chỉnh hơn, để chương trình thực sự hữu ích hơn, đáp ứng những nghiệp vụ phức tạp hơn, dữ liệu lớn hơn.

TÀI LIỆU THAM KHẢO

[1] https://www.youtube.com/watch?v=Ph3qbiCldE0&list=PLSMUwja5VsJU-ry82soUtUmUF73YcuabC

[3] <https://www.w3schools.com/>,

[4] <https://stackoverflow.com/>,

[5] <https://topdev.vn/blog/from-mvc-to-modern-web-frameworks/>