**TRƯỜNG ĐẠI HỌC TÀI NGUYÊN MÔI TRƯỜNG TP.HCM**

**KHOA: HỆ THỐNG THÔNG TIN VÀ VIỄN THÁM**

****

**BÁO CÁO THỰC TẬP TỐT NGHIỆP**

**CHUYÊN NGÀNH: CÔNG NGHỆ THÔNG TIN**

**ĐỀ TÀI: XÂY DỰNG WEBSITE QUẢN LÝ**

**HỌC PHẦN SINH VIÊN**

Giảng viên hướng dẫn: **ThS Phạm Trọng Huynh**

Sinh viên thực hiện: **Trần Lê Tiến Hòa**

Lớp: **08\_CNPM**

Khóa: **2019-2023**

**TP. HỒ CHÍ MINH, THÁNG 2 NĂM 2023**

**TRƯỜNG ĐẠI HỌC TÀI NGUYÊN MÔI TRƯỜNG TP.HCM**

**KHOA: HỆ THỐNG THÔNG TIN VÀ VIỄN THÁM**

****

**BÁO CÁO THỰC TẬP TỐT NGHIỆP**

**CHUYÊN NGÀNH: CÔNG NGHỆ THÔNG TIN**

**ĐỀ TÀI: XÂY DỰNG WEBSITE QUẢN LÝ**

**HỌC PHẦN SINH VIÊN**

Giảng viên hướng dẫn: **ThS Phạm Trọng Huynh**

Sinh viên thực hiện: **Trần Lê Tiến Hòa**

Lớp: **08\_CNPM**

Khóa: **2019-2023**

**TP. HỒ CHÍ MINH, THÁNG 2 NĂM 2023**

**MỞ ĐẦU**

Sự ra đời và phát triển của công nghệ thông tin là một trong những thành tựu vĩ đại nhất của con người. Nhịp độ phát triển công nghệ thông tin đang là một vấn đề rất được các nghành khoa học, giáo dục, kinh tế,.. quan tâm. Nó hiện hữu với một tầm vóc hết sức mạnh mẽ, to lớn và ngày càng lớn mạnh thêm. Tin học giữ vai trò đặc biệt trong hoạt động của con người, trong đời sống xã hội, trong khoa học kỹ thuật, trong sản xuất kinh doanh, trong quản lý về giáo dục, trong các hoạt động mua bán hàng hoá....

Ứng dụng website trong việc quảng cáo, cập nhật thông tin không còn xa lạ với các nghành nghề hiện nay. Lợi ích mà các chương trình quản lý mang lại khiến người ta không thể phủ nhận tính hiệu quả của nó.

Lĩnh vực Giáo dục - Đào tạo - Trường học cũng không phải là ngoại lệ. Để có thể theo kịp với đà phát triển của xã hội thì ngành giáo dục cũng phải luôn cập nhật những thông tin, công nghệ mới nhất vậy nên website là một hành trang không thể thiếu trong việc tạo mối liên kết giữa học sinh, sinh viên cũng như là phụ huynh đối với nhà trường. Để nhìn nhận một cách khách quan thì lĩnh vực giáo dục, trường học cũng được xem là một lĩnh vực kinh doanh, mà một khi đã kinh doanh thì ắt có cạnh tranh. Mà để có thể cạnh tranh một cách hiệu quả thì điều đầu tiên phải có một chiến lược marketing hiệu quả. Và thiết kế website giáo dục, trường học chính là một phần của các chiến lược đó. Đằng sau những website trường hay Fanpage đã quá phổ biến thì vẫn đòi hỏi việc quản lý sinh viên – học phần đều được sử dụng, khai thác trên nền tảng ứng dụng hay website. Trang web này sẽ cho phép quản lý các thông tin về sinh viên, bao gồm thông tin cá nhân, lịch học, điểm số, v.v. Nó cũng cho phép tra cứu thông tin và thống kê dữ liệu. Việc sử dụng trang web này sẽ giúp giảm thiểu tối đa sự rối loạn và làm cho việc quản lý thông tin trở nên dễ dàng và hiệu quả hơn.

Việc quản lý trang website học phần sinh viên đòi hỏi bộ phận quản lý phải thực hiện nhiều nghiệp vụ phức tạp. Để thực hiện đề tài này, chúng ta sẽ phải tìm hiểu kỹ về công nghệ web, lập trình, và thiết kế hệ thống. Chúng ta sẽ sử dụng các công cụ và ngôn ngữ phát triển web để hoàn thành dự án này, chính từ sự yếu tố và hiệu quả của các phần mềm/website quản lý nghiệp vụ thông tin sinh viên – học phần hiện nay nên em quyết định chọn đề tài “ Xây dựng phần mềm quản lý học phần sinh viên”.

**LỜI CẢM ƠN**

Trong lời đầu tiên của báo cáo đồ án thực tập tốt nghiệp với đề tài “Xây dựng website quản lý học phần sinh viên” này, em muốn gửi những lời cảm ơn và biết ơn chân thành nhất của mình đến tất cả những người đã hỗ trợ, giúp đỡ em về kiến thức và tinh thần trong quá trình thực hiện đồ án.

Em xin chân thành cám ơn giảng viên - ThS Phạm Trọng Huynh, Giảng viên Khoa Hệ Thống Thông Tin Và Viễn Thám, Trường Đại học Tài Nguyên Và Môi Trường TP. Hồ Chí Minh, là giảng viên hướng dẫn em trong suốt quá trình thực tập đã tận tình giúp đỡ chi tiết để em có đủ kiến thức và vận dụng chúng vào đồ án thực tập tốt nghiệp này. Kính chúc thầy thật nhiều sức khỏe, hạnh phúc.

Em xin cảm ơn các anh chị trong công ty Trung tâm tin học thống kê khu vực II (Cosis) và anh Lều Văn Hoan đã trực tiếp hướng dẫn, giúp đỡ em trong suốt quá trình thực tâp thực hiện đồ án.

Xin chân thành cảm ơn ban giám hiệu nhà trường đã tạo điều kiện tốt nhất trong quá trình học tập. Do thời gian thực hiện có hạn, kiến thức còn nhiều hạn chế nên đồ án thực hiện chắc chắn không tránh khỏi những thiếu sót nhất định. Em rất mong nhận được ý kiến đóng góp của thầy cô giáo và các bạn bè để em có thêm kinh nghiệm và tiếp tục hoàn thiện đồ án của mình.

Em xin chân thành cảm ơn!

**NHẬN XÉT**

**(Của Cán bộ hướng dẫn tại đơn vị thực tập)**

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

…………………………………………………………………………………….

XÁC NHẬN CỦA CƠ QUAN

..…, ngày…..tháng…..năm…

THỦ TRƯỞNG

CÁN BỘ HƯỚNG DẪN

(Ký tên, đóng dấu)

(Ký tên)

**NHẬN XÉT**

**(Của giảng viên hướng dẫn tại khoa)**

…………………………………………………………………………………………..

…………………………………………………………………………………………..

…………………………………………………………………………………………..

…………………………………………………………………………………………..

…………………………………………………………………………………………..

…………………………………………………………………………………………..

………………………………………………………….……………………………….

…………………………………………………………………………………………..

…………………………………………………………………………………………..

**Kết luận: Đồng ý hoặc Không đồng ý cho sinh viên nộp báo cáo thực tập**

..…, ngày…..tháng…..năm…

CÁN BỘ HƯỚNG DẪN

(Ký tên)

**NHẬN XÉT**

**(Của giáo viên phản biện)**

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

..…, ngày…..tháng…..năm…

NGƯỜI NHẬN XÉT

(Ký tên)

**LỊCH LÀM VIỆC**

(Của sinh viên tại nơi thực tập)

Họ và tên sinh viên: Trần Lê Tiến Hòa

Đơn vị thực tập: Trung tâm tin học thống kê khu vực II (Cosis)

Họ và tên người hướng dẫn (đơn vị thực tập): Lều Văn Hoan

Thời gian thực tập từ ngày 28 tháng 11 năm 2022 đến ngày 15 tháng 1 năm 2023

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tuần | Nội dung  công việc được giao | Tự nhận  xét về  mức độ  hoàn  thành | Nhận xét của  CB hướng dẫn | Chữ ký của  CB hướng  dẫn |
| **1**  28/11-2/12 | Tham quan công ty, hướng dẫn  cách làm việc, giờ giấc làm  việc, thái độ, tác phong, kĩ  năng, kỉ luật của công ty. | Hoàn  thành |  |  |
| **2**  5/12-9/12 | Tìm hiểu, đọc tài liệu phân tích hệ thống số liệu thuế - Phân tích hệ thống số liệu thuế doanh nghiệp thực tế | Hoàn  thành |  |  |
| **3** 12/12-16/12 | Thiếu kế giao diện hệ thống thuế | Hoàn  thành |  |  |
| **4** 19/12-23/12 | Tiến hành xây dựng CSDL và back-end cho hệ thống thuế | Hoàn  thành |  |  |
| **5** 26/12-30/12 | Tiếp tục xây dựng back-end cho hệ thống thuế | Hoàn  thành |  |  |
| **6** 2/1-6/1 | Hoàn thiện các module, đánh giá sản phẩm hệ thống thuế | Hoàn  thành |  |  |
| **7** 9/1-15/1 | Xây dựng website quản lý | Hoàn  thành |  |  |

TP. Hồ Chí Minh, ngày….tháng…..năm 2023

**Sinh Viên**

MỤC LỤC

[CHƯƠNG I: TỔNG QUAN 1](#_Toc126795617)

[1.1. Giới thiệu đơn vị thực tập 1](#_Toc126795618)

[1.1.1. Thông tin cơ bản 1](#_Toc126795619)

[1.1.2. Cơ cấu tổ chức 1](#_Toc126795620)

[1.1.3. Lịch sử hình thành – phát triển 1](#_Toc126795621)

[1.1.4. Chức năng và nhiệm vụ 1](#_Toc126795622)

[1.2. Giới thiệu nội dung công việc được giao thực tập 3](#_Toc126795623)

[1.3. Phạm vi đề tài 3](#_Toc126795624)

[CHƯƠNG II: CƠ SỞ LÝ LUẬN 4](#_Toc126795625)

[2.1. Lý thuyết 4](#_Toc126795626)

[2.2. Kĩ thuật 4](#_Toc126795627)

[2.2.1. Ngôn ngữ Java 5](#_Toc126795628)

[2.2.2. MySql Server 7](#_Toc126795629)

[2.2.3. Spring Boot 8](#_Toc126795630)

[2.2.4. GitHub 9](#_Toc126795631)

[2.3. Xây dựng mô hình ứng dụng 15](#_Toc126795632)

[2.3.1. Luồng xử lý thông tin 16](#_Toc126795633)

[2.3.2. Ưu điểm 17](#_Toc126795634)

[2.3.3. Nhược điểm 18](#_Toc126795635)

[CHƯƠNG III: CÀI ĐẶT THỬ NGHIỆM 19](#_Toc126795636)

[3.1. Phương pháp nghiên cứu 19](#_Toc126795637)

[3.2. Phương pháp thực hiện 19](#_Toc126795638)

[3.2.1. Khảo sát thực tiễn 19](#_Toc126795639)

[3.2.2. Phân tích hệ thống 20](#_Toc126795640)

[3.2.3. Thiết kế - hoàn thiện giao diện 50](#_Toc126795641)

[CHƯƠNG IV: KẾT LUẬN 59](#_Toc126795642)

[4.1. Kết luận 59](#_Toc126795643)

[4.1.1. Kết quả đạt được 59](#_Toc126795644)

[4.1.2. Đánh giá kết quả đạt được 59](#_Toc126795645)

[4.1.3. Kiến thức đạt được 60](#_Toc126795646)

[4.1.4. Những tích lũy đạt được trong quá trình thực tập 60](#_Toc126795647)

[4.2. Hướng phát triển 60](#_Toc126795648)

[TÀI LIỆU THAM KHẢO 62](#_Toc126795649)

**DANH MỤC HÌNH**

[Hình 2. 1: Ngôn ngữ Java 5](#_Toc126796079)

[Hình 2. 2: Giới thiệu GitHub 10](#_Toc126796080)

[Hình 2. 3: Mô hình MVC 15](#_Toc126796081)

[Hình 2. 4: Luồng xử lí thông tin của mô hình MVC 16](#_Toc126796082)

[Hình 3. 1: Sơ đồ ERD 20](#_Toc126796083)

[Hình 3. 2: Sơ đồ tổ chức Database 21](#_Toc126796084)

[Hình 3. 3: Sơ đồ tổng quát hệ thống 22](#_Toc126796085)

[Hình 3. 4: Usecase đăng nhập 23](#_Toc126796086)

[Hình 3. 5: Usecase quên mật khẩu 24](#_Toc126796087)

[Hình 3. 6: Usecase quản lý học phần 26](#_Toc126796088)

[Hình 3. 7: Usecase xem thông tin học phần 27](#_Toc126796089)

[Hình 3. 8: Usecase thêm học phần 28](#_Toc126796090)

[Hình 3. 9: Usecase sửa học phần 29](#_Toc126796091)

[Hình 3. 10: Usecase tìm kiếm học phần 30](#_Toc126796092)

[Hình 3. 11: Usecase quản lý điểm 32](#_Toc126796093)

[Hình 3. 12: Usecase xem điểm 33](#_Toc126796094)

[Hình 3. 13: Usecase thêm điểm 34](#_Toc126796095)

[Hình 3. 14: Usecase sửa điểm 36](#_Toc126796096)

[Hình 3. 15: Usecase tìm kiếm điểm 37](#_Toc126796097)

[Hình 3. 16: Usecase quản lý đăng ký học phần 39](#_Toc126796098)

[Hình 3. 17: Usecase đăng ký học phần 40](#_Toc126796099)

[Hình 3. 18: Usecase hủy đăng ký học phần 41](#_Toc126796100)

[Hình 3. 19: Usecase quản lý lớp học phần 43](#_Toc126796101)

[Hình 3. 20: Usecase xem lịch học 44](#_Toc126796102)

[Hình 3. 21: Usecase thêm lớp học phần 45](#_Toc126796103)

[Hình 3. 22: Usecase sửa lớp học phần 47](#_Toc126796104)

[Hình 3. 23: Usecase quản lý chương trình đào tạo 48](#_Toc126796105)

[Hình 3. 24: Giao diện đăng nhập 50](#_Toc126796106)

[Hình 3. 25: Giao diện quên mật khẩu 50](#_Toc126796107)

[Hình 3. 26: Giao diện quản lý điểm 51](#_Toc126796108)

[Hình 3. 27: Giao diện thông tin điểm sinh viên 51](#_Toc126796109)

[Hình 3. 28: Giao diện quản lý học phần 52](#_Toc126796110)

[Hình 3. 29: Giao diện thông tin học phần thuộc khoa Viễn thám 52](#_Toc126796111)

[Hình 3. 30: Giao diện thêm học phần 53](#_Toc126796112)

[Hình 3. 31: Giao diện sửa học phần 53](#_Toc126796113)

[Hình 3. 32: Giao diện quản lý lớp học phần 54](#_Toc126796114)

[Hình 3. 33: Giao diện tìm kiếm lớp học phần 54](#_Toc126796115)

[Hình 3. 34: Giao diện thêm lớp học phần 55](#_Toc126796116)

[Hình 3. 35: Giao diện sửa lớp học phần 55](#_Toc126796117)

[Hình 3. 36: Giao diện mở lớp đăng ký học phần 56](#_Toc126796118)

[Hình 3. 37: Giao diện sinh viên đăng kí học phần 56](#_Toc126796119)

[Hình 3. 38: Giao diện danh sách chương trình đào tạo 57](#_Toc126796120)

[Hình 3. 39: Giao diện thêm chương trình đào tạo 57](#_Toc126796121)

[Hình 3. 40: Giao diện học phần sinh viên đang có trong đợt học 58](#_Toc126796122)

[Hình 3. 41: Giao diện lịch học cá nhân 58](#_Toc126796123)

**DANH MỤC BẢNG**

[Bảng 3. 1: Usecase đăng nhập 23](#_Toc126796142)

[Bảng 3. 2: Usecase quên mật khẩu 24](#_Toc126796143)

[Bảng 3. 3: Usecase quản lý học phần 26](#_Toc126796144)

[Bảng 3. 4: Usecase xem học phần 27](#_Toc126796145)

[Bảng 3. 5: Usecase thêm học phần 28](#_Toc126796146)

[Bảng 3. 6: Usecase sửa thông tin học phần 29](#_Toc126796147)

[Bảng 3. 7: Usecase tìm học phần 31](#_Toc126796148)

[Bảng 3. 8: Usecase quản lý điểm 32](#_Toc126796149)

[Bảng 3. 9: Usecase xem điểm 33](#_Toc126796150)

[Bảng 3. 10: Usecase thêm điểm 34](#_Toc126796151)

[Bảng 3. 11: Usecase chỉnh sửa điểm 36](#_Toc126796152)

[Bảng 3. 12: Usecase tìm kiếm điểm 38](#_Toc126796153)

[Bảng 3. 13: Usecase quản lý đăng ký học phần 39](#_Toc126796154)

[Bảng 3. 14: Usecase đăng ký học phần 40](#_Toc126796155)

[Bảng 3. 15: Usecase hủy đăng ký học phần 41](#_Toc126796156)

[Bảng 3. 16: Usecase quản lý lớp học phần 43](#_Toc126796157)

[Bảng 3. 17: Usecase xem lịch học 44](#_Toc126796158)

[Bảng 3. 18: Usecase thêm lớp học phần 45](#_Toc126796159)

[Bảng 3. 19: Usecase sửa lớp học 47](#_Toc126796160)

[Bảng 3. 20: Usecase quản lý chương trình đào tạo 48](#_Toc126796161)

# **CHƯƠNG I: TỔNG QUAN**

* 1. **Giới thiệu đơn vị thực tập**

### **Thông tin cơ bản**

* Tên đơn vị : Trung tâm tin học thống kê khu vực II (Cosis).
* Địa chỉ : 54A, Nơ Trang Long, P.14, Q.Bình Thạnh, TP.Hồ Chí Minh.
* Fax : (028) 35515432.
* SĐT: (028) 35515431 – 35515433.
* Website: Cosis.vn.

### **Cơ cấu tổ chức**

Gồm có Ban Giám đốc và 6 phòng:

* Phòng tổ chức – Hành chánh
* Phòng kế hoạch – Kinh doanh
* Phòng phát triển ứng dụng
* Phòng xử lý thông tin
* Phòng kỹ thuật và quản trị hệ thống
* Phòng đào tạo

### **Lịch sử hình thành – phát triển**

* Trung tâm Tin học Thống kê Khu vực II - tên giao dịch COSIS - tiền thân là Công ty Dịch vụ Thông tin Thống kê được thành lập từ năm 1988.
* Năm 1992 - đổi thành Trung tâm Xử lý Thông tin Thống Kê thuộc Cục thống kê TP.Hồ Chí Minh.
* Năm 2004 - đổi thành Trung tâm Tin học Thống kê Khu vực II, đơn vị sự nghiệp trực thuộc Tổng cục Thống kê - Bộ Kế hoạch & Đầu tư.

### **Chức năng và nhiệm vụ**

Trung tâm có các nhiệm vụ, chức năng chính như:

Quản lý, vận hành hệ thống hạ tầng CNTT– truyền thông tại khu vực II của Tổng cục Thống kê bao gồm:

* Trung tâm máy chủ khu vực II;
* Trung tâm sao lưu dữ liệu của Tổng cục Thống kê tại TPHCM;
* Hệ thống thư điện tử khu vực II của Tổng cục Thống kê;
* Thực hiện sao lưu định kỳ dữ liệu của Tổng cục Thống kê;
* Thực hiện bảo trì hệ thống hạ tầng CNTT tại khu vực II của Tổng cục Thống kê;

Quản lý, vận hành, bảo trì hệ thống mạng của đơn vị; bảo đảm kết nối thông suốt hệ thống mạng các đơn vị được phân công phụ trách với hệ thống mạng của cơ quan Tổng cục Thống kê.

Xử lý dữ liệu điều tra, tổng điều tra và báo cáo thống kê; xây dựng cơ sở dữ liệu, kho dữ liệu thống kê.

Phát triển và hướng dẫn các phần mềm ứng dụng; phối hợp lựa chọn giải pháp công nghệ mới áp dụng trong ngành Thống kê.

Thực hiện bảo trì và hỗ trợ, tư vấn kỹ thuật về vận hành đối với hệ thống mạng, thiết bị tin học của các đơn vị khu vực II.

Thực hiện đào tạo, bồi dưỡng về CNTT.

Hợp tác, liên kết với với các tổ chức, cơ quan, đơn vị trong nước và ngoài nước nghiên cứu và phát triển CNTT vào công tác thống kê .

Thực hiện công nghiệp phần mềm, dịch vụ về lĩnh vực CNTT. Cung cấp các dịch vụ:

* Xây dựng phần mềm.
* Xử lý số liệu.
* Thiết kế và thi công hệ thống mạng.
* Đào tạo tin học.
  1. **Giới thiệu nội dung công việc được giao thực tập**

Được làm việc tại một trung tâm tin học chuyên về lĩnh vực thống kế, ở Trung tâm tin học thống kê khu vực II (Cosis) em thực tập với vị trí đó là thực tập sinh IT. Với vị trí công việc này, nhiệm vụ chủ yếu của em đó là nghiên cứu và tham khảo các tài liệu về công nghệ, học hỏi thêm các nghiệp vụ thống kê, tạo demo sử dụng các công nghệ đã tìm hiểu được và được tạo điều kiện tham gia một số các dự thực tế của trung tâm, ngoài ra còn được trung tâm phân công nhiệm vụ xây dựng website quản lý bằng những công nghệ đã tìm hiểu qua. Ngoài ra còn được tiếp xúc và nghiên cứu các sản phẩm của trung tâm như phần mềm quản lý Lập báo cáo thống kê Sở - Ngành, Phần mềm Số hóa và quản lý văn bản, Phần mềm quản lý Lưu trữ và tra cứu báo cáo thống kê, Phần mềm Giao kế hoạch – Chấm điểm thi đua.

* 1. **Phạm vi đề tài**

Ở mục trước em đã giới thiệu về nội dung công việc được giao thực tập. Ở mục này sẽ là phạm vi đề tài mà em trình bày trong báo cáo.

Tên đề tài em chọn đó là “Xây dựng website quản lý học phần sinh viên”. Một website quản lý có rất nhiều những module, những chức năng khác nhau có thể kể đến như là : Quản lý học phần, quản lý điểm, quản lý chương trình đào tạo, quản lý đăng ký học phần, .v.v. Công việc của emq tập trung vào việc xây dựng một vài chức năng như: Chức năng quản lý như thêm, sửa, xóa các học phần, sinh viên; Ngoài ra có thể xem điểm, giảng viên, chương trình đào tạo, đăng ký học phần, gửi thông báo qua gmail,…

# **CHƯƠNG II: CƠ SỞ LÝ LUẬN**

* 1. **Lý thuyết**

Trong lĩnh vực giáo dục như hiện nay việc tin học hoá các hoạt động quản lý là rất cần thiết. Quản lý trang website học phần sinh viên đòi hỏi bộ phận quản lý phải thực hiện nhiều nghiệp vụ phức tạp. Tạo cho ai có nhu cầu muốn biết tất cả các thông tin về trường học của mình hoặc trường học khác như thông tin về Khoa, Giảng viên, Chương trình đào tạo, Điểm thi, lịch học,…

Chính từ sự cần thiết và hiệu quả của các phần mềm/website quản lý nghiệp vụ thông tin sinh viên hiện nay nên em quyết định chọn đề tài “ Xây dựng Website quản lý học phần sinh viên”.

* 1. **Kĩ thuật**

Về kỹ thuật để xây dựng trang web thì dưới đây là một số kỹ thuật và công nghệ được áp dụng.

* Ngôn ngữ lập trình : Sử dụng ngôn ngữ java, Javascript.
* Phần mềm : Sử dụng các phần mềm như Visual Studio 2019, MySQL Server, Google Chrome…
* Framework : Spring boot.
* Ngoài ra còn sử dụng nhiều công cụ như: Chrome DevTools, Terminal, Nuget Package Manager, GitHub, …
  + 1. **Ngôn ngữ Java**
       1. **Khái niệm**

Logo, company name

Description automatically generated

Hình 2. 1: Ngôn ngữ Java

Java là một ngôn ngữ lập trình hướng đối tượng, dựa trên lớp được thiết kế để có càng ít phụ thuộc thực thi càng tốt. Nó là ngôn ngữ lập trình có mục đích chung cho phép các nhà phát triển ứng dụng viết một lần, chạy ở mọi nơi , nghĩa là mã Java đã biên dịch có thể chạy trên tất cả các nền tảng hỗ trợ Java mà không cần biên dịch lại. Các ứng dụng Java thường được biên dịch thành bytecode có thể chạy trên bất kỳ máy ảo Java (JVM) nào bất kể kiến trúc máy tính bên dưới. Cú pháp của Java tương tự như C và C++, nhưng có ít cơ sở cấp thấp hơn các ngôn ngữ trên.

Java được khởi đầu bởi James Gosling và bạn đồng nghiệp ở Sun MicroSystem năm 1991. Ban đầu Java được tạo ra nhằm mục đích viết phần mềm cho các sản phẩm gia dụng, và có tên là Oak.

Java được phát hành năm 1994, đến năm 2010 được Oracle mua lại từ Sun MicroSystem.

* + - 1. **Ứng dụng**

Viết ứng dụng web (J2EE): Java thường được sử dụng để xây dựng các hệ thống web lớn đòi hỏi độ bảo mật cao, số lượng người dùng lớn như ngân hàng, phần mềm quản lý bệnh viện, CRM, HRM.... Đối với các website nhỏ thông thường rất ít viết bằng Java.

Viết ứng dụng mobile (J2ME): Trước đây nền tảng J2ME thường được sử dụng để viết game và app cho di động feature phone (file .jar) và giờ đây khi smartphone Android lên ngôi Java lại tiếp tục được sử dụng để viết app và game cho nền tảng Android (file .apk).

Viết ứng dụng desktop (J2SE): Các ứng dụng desktop viết bằng Java thật sự không nhiều có thể kể đến một số phần mềm như JMeter hoặc Designer Vista. Lợi thế lớn nhất của ứng dụng Java là bạn chỉ viết một lần và sau đó có thể đem chương trình lên Windows, Linux hay Mac để chạy mà không cần phải viết lại. Tuy nhiên do chạy trên JVM nên performance của ứng dụng thấp hơn một chút so với các ngôn ngữ như C/C++, C#.

* + - 1. **Ưu điểm**
* Java là ngôn ngữ độc lập với nền tảng vì chúng ta có thể chạy mã Java ở bất kỳ máy nào mà không cần phần mềm đặc biệt.
* Là ngôn ngữ lập trình cấp cao dễ học và dễ hiểu.
* Là ngôn ngữ hướng đối tượng làm tăng khả năng phát triển mã dễ dàng và tăng hiệu quả.
* Là ngôn ngữ an toàn vì Java không sử dụng con trỏ.
* Quản lý bộ nhớ hiệu quả.
* Hỗ trợ đa luồng, người dùng có thể thực hiện cùng lúc nhiều chương trình.
* Có nhiều tính năng: tự động thu gom rác, không sử dụng con trỏ, xử lý ngoại lệ…
  + - 1. **Nhược điểm**
* Là ngôn ngữ cấp cao nên phải xử lý các mức biên dịch và trừu tượng của một máy ảo.
* Hiệu suất kém.
* Có ít trình xây dựng GUI (Giao diện người dùng đồ họa) như Swing, SWT, JSF và JavaFX.
* Nếu viết những đoạn mã dài phức tạp dễ ảnh hưởng đến khả năng đọc của mã.
  + 1. **MySql Server**
       1. **Khái niệm**

MySQL là hệ quản trị cơ sở dữ liệu tự do nguồn mở phổ biến nhất thế giới và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng. Vì MySQL là hệ quản trị cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh. Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng có truy cập CSDL trên internet. Người dùng có thể tải về MySQL miễn phí từ trang chủ. MySQL có nhiều phiên bản cho các hệ điều hành khác nhau: phiên bản Win32 cho các hệ điều hành dòng Windows, Linux, Mac OS X, Unix, FreeBSD, NetBSD, Novell NetWare, SGI Irix, Solaris, SunOS,..

MySQL là một trong những ví dụ rất cơ bản về Hệ Quản trị Cơ sở dữ liệu quan hệ sử dụng Ngôn ngữ truy vấn có cấu trúc (SQL).

MySQL được sử dụng cho việc bổ trợ Node.js, PHP, Perl, và nhiều ngôn ngữ khác, làm nơi lưu trữ những thông tin trên các trang web viết bằng NodeJs, PHP hay Perl,...

* + - 1. **Ưu điểm**
* Nhanh chóng: Nhờ vào việc đưa ra một số những tiêu chuẩn và cho phép MySQL làm việc hiệu quả cũng như tiết kiệm chi phí, giúp gia tăng tốc độ thực thi.
* Mạnh mẽ và khả năng mở rộng: MySQL hoàn toàn có thể xử lý số lượng lớn dữ liệu và đặc biệt hơn thế nữa thì nó còn có thể mở rộng nếu như cần thiết.
* Đa tính năng: Ưu điểm MySQL là gì? MySQL hiện đang hỗ trợ nhiều những chức năng SQL rất được mong chờ từ 1 hệ quản trị CSDL quan hệ cả gián tiếp cũng như trực tiếp.
* Độ bảo mật cao: MySQL là gì? Hiện tại nó đang rất thích hợp cho những ứng dụng truy cập CSDL thông qua internet khi sở hữu rất nhiều những tính năng về bảo mật và thậm chí là đang ở cấp cao.
* Dễ dàng sử dụng: MySQL đang là cơ sở dữ liệu dễ sử dụng, ổn định, tốc độ cao và hoạt động trên rất nhiều những hệ điều hành đang cung cấp 1 hệ thống lớn những hàm tiện ích rất mạnh.
  + - 1. **Nhược điểm**
* Dung lượng hạn chế: Trong trường hợp nếu như số lượng bản ghi của bạn đang lớn dần lên thì khi đó quá trình truy xuất dữ liệu sẽ diễn ra vô cùng khó khăn. Như vậy cần phải áp dụng rất nhiều những biện pháp khác nhau để có thể gia tăng được tốc độ truy xuất những dữ liệu ví dụ như tạo cache MySQL hoặc chia tải database ra nhiều server.
* Độ tin cậy: Nhược điểm MySQL là gì? Theo đó cách thức nhận chức năng cụ thể đang được xử lý cùng với MySQL (ví dụ như kiểm toán, những giao dịch, tài liệu tham khảo,…) khiến cho nó trở nên kém tin cậy hơn một số những hệ quản trị về cơ sở dữ liệu có quan hệ khác.
* Giới hạn: MySQL là gì? Theo thiết kế thì MySQL không có ý định thực hiện toàn bộ và nó đang đi kèm cùng với những hạn chế liên quan tới chức năng mà một số ứng dụng có thể cần tới.
  + 1. **Spring Boot**
       1. **Sping Boot là gì?**

Từ những thông tin ở trên, ta có thể khẳng định Spring Boot chính là một Java framework siêu to và khổng lồ và có nhiều khả năng hữu ích vì nó có thể giúp lập trình viên giải quyết rất nhiều vấn đề. So với framework Spring thông thường, Spring Boot tỏ ra những lợi thế vượt trội. Khi sử dụng Spring Boot, rất nhiều thứ được cải tiến hỗ trợ lập trình viên như:

* Auto config: tự động cấu hình thay lập trình viên, bạn chỉ cần viết code và tiến hành chạy hệ thống là được.
* Dựa trên các Annotation để tạo lập các bean thay vì XML.
* Server Tomcat có thể được nhúng ngay trong file JAR build ra và có thể chạy ở bất kì đâu mà java chạy được.

Khi sử dụng Spring Boot, lập trình viên chỉ cần:

* Sử dụng Spring Initializr: nhập các thông tin của dự án (project), chọn thư viện (Library) rồi tải code về máy.
* Mở mã nguồn (source code) và bắt đầu viết code.
* Có thể chạy ngay trong IDE, hoặc build thành file JAR mà không cần cấu hình config cho server nữa.
  + - * 1. **AOP, Aspects and Instrumentation**

Những module này hỗ trợ cài đặt lập trình hướng khía cạnh (Aspect Oriented Programming), hỗ trợ tích hợp với AspectJ.

* + - * 1. **Data Access / Integration**

Nhóm này bao gồm JDBC, ORM, OXM, JMS và module Transaction. Những module này cung cấp khả năng giao tiếp với database

* + - * 1. **Web**

Hay còn gọi là Spring MVC Nhóm này gồm Web, Web-Servlet… hỗ trợ việc tạo ứng dụng web.

* + - 1. **Các lợi ích của Spring Boot**
* Spring cho phép lập trình viên sử dụng POJOs. Việc sử dụng POJOs giúp bạn không phải làm việc với EJB, ứng dụng, các luồng chạy, cấu hình… đơn giản hơn rất nhiều.
* Spring được tổ chức theo kiểu mô đun. Số lượng các gói và các lớp khá nhiều, nhưng bạn chỉ cần quan tâm đến những gì bạn cần và không cần quan tâm đến phần còn lại.
* Spring hỗ trợ sử dụng khá nhiều công nghệ như ORM Framework, các logging framework, JEE, các thư viện tạo lịch trình (Quartz và JDK timer)…
* Module Web của Spring được thiết kế theo mô hình MVC nên nó cung cấp đầy đủ các tính năng giúp thay thế các web framework khác như Struts.
  + 1. **GitHub**
       1. **GitHub là gì?**

Đối với những ai làm trong lĩnh vực công nghệ thông tin, chắc hẳn đã biết GitHub là gì. Tuy nhiên, để tư duy có hệ thống, bạn nên xem lại khái niệm một lần nữa. Đây là dịch vụ nổi tiếng trong cung cấp lưu trữ mã nguồn Git.

GitHub là gì? GitHub là công cụ hỗ trợ đắc lực trong quá trình code của lập trình viên

Nguồn tư liệu này phục vụ cho các dự án phần mềm với đầy đủ các tính năng. Bên cạnh đó còn là sự bổ sung những đặc tính hỗ trợ liên quan đến Social. Thông qua đó, các developer trao đổi, tương tác với nhau thuận tiện hơn.

Logo

Description automatically generated

Hình 2. 2: Giới thiệu GitHub

Tóm lại, những người mới làm quen cần hiểu GitHub là gì? Đây là công cụ giúp quản lý, kiểm tra source code, tổ chức theo dạng dữ liệu phân tán. Các phần code của cả nhóm sẽ được đồng bộ trước khi lên server.

* + - 1. **Ưu điểm của GitHub**

Những lợi ích cụ thể các nhà lập trình được hưởng từ GitHub là gì? Bạn sẽ biết điều đó thông qua thế mạnh tổng hợp dưới đây. Các coder chắc chắn nhận thấy tiềm năng và động lực để ứng dụng ngay trong thực tế.

* + - * 1. **Dễ dàng quản lý Source Code**

Dễ dàng quản lý Source Code là điều đã được nói đến khi giải thích GitHub là gì. Khi bạn tạo repo, toàn bộ thành quả sẽ được lưu trên công cụ này. Tại đây, bạn dễ dàng theo dõi lại quá trình mình đã làm việc.

Điều đó được nhìn thấy thông qua các comment sau mỗi lần commit. Điều thú vị hơn nữa chính là nhiều người có thể cùng làm một repo. Từ đó, bạn biết được cụ thể ai đã làm gì, chỉnh sửa ở đâu.

Tiếp theo, nguồn của bạn được phát triển theo nhiều nhánh khác nhau. Nguyên tắc làm việc với GitHub là gì? Bạn có thể rẽ các hướng để phát triển, mở rộng dự án. Đến cuối cùng, các phần phải merge lại để tạo ra project hoàn chỉnh.

* + - * 1. **Tracking những thay đổi qua các version**

Đối với những dự án có nhiều thành viên cùng phát triển, hiểu GitHub là gì rất cần thiết. Khi làm theo nhóm, nhu cầu lớn nhất là theo dõi revisions – ai thay đổi cái gì, lúc nào và lưu tệp ở đâu.

Công cụ đã tính đến nhu cầu này và tối ưu tốt nhất có thể. Điều đó thể hiện qua việc luôn lưu lại những thay đổi đã push lên repository. Giống như Google Drive, bạn sẵn có phiên bản dự phòng để thay thế cho tình huống bị xóa mất dữ liệu chưa lưu.

* + - * 1. **Markdown**

Markdown trong GitHub là gì? Đây là cách định dạng text trên web. Bạn có thể chỉnh sửa việc hiển thị của document, format,…Chẳng hạn như dạng in đậm, in nghiêng, thêm hình, tạo list cho những thứ cần Markdown.

Hầu hết, hình thức này chỉ là đoạn text đơn thuần với những ký tự đặc biệt được chèn thêm vào. Trong GitHub, bạn dễ dàng sử dụng ở các vị trí như: Git, Comments tại Issues và Pull Requests. File có đuôi .md hay .markdown extension cũng xuất hiện.

* + - * 1. **Khẳng định chuyên môn**

Sau khi biết GitHub là gì và cách sử dụng, trình độ chuyên môn của bạn nâng lên tầm cao mới. Những Source Code luôn là minh chứng tốt nhất để bạn khẳng định mình là developer thực thụ.

Trong mắt nhà tuyển dụng, GitHub giống như chiếc máy liar – detech. Thông qua đó, họ dễ dàng phân biệt developer thật và giả. Vì thế, hãy đầu tư một tài khoản và dẫn link vào CV.

Thậm chí, nhiều công ty lớn trên thế giới còn xem đây là yêu cầu trong quy trình tuyển dụng. Những đóng góp của bạn cho cộng động Coder chính là lợi thế cạnh tranh không thể phủ nhận.

* + - * 1. **Cải thiện khả năng Code**

GitHub hướng dẫn cung cấp cho bạn một cách học vô cùng hiệu quả. Điều này có được nhờ hàng vạn dự án mở, người đóng góp và hàng tỷ commit mỗi ngày.

Chỉ cần đăng nhập vào nền tảng, một kho tàng kiến thức update nhất đã mở ra. Việc tự phân tích, so sánh giúp tiếp cận kiến thức chuyên môn chủ động hơn. Bên cạnh đó, đây là cơ hội cọ sát với project gần gũi trong thực tế.

* + - * 1. **Kho tài nguyên tuyệt vời**

Khi tìm hiểu GitHub là gì, bạn sẽ thấy chức năng Explore tuyệt vời như thế nào. Người dùng dễ dàng theo dõi, tìm kiếm các open source projects theo chính xác technology pattern.

Công cụ hỗ trợ Code Search không kể ở dưới dạng dự án riêng hay website. Bên cạnh đó, kỹ thuật SEO khá tốt nên người dùng có thể tìm kiếm bất kỳ Code String nào được chia sẻ.

* + - * 1. **GitHub Action**

Sau khi cài đặt GitHub sẽ xuất hiện những Workflow Scripts tự động chạy trên server. Developer có thể dùng để phản hồi sự kiện trên repositories hoặc thực hiện vài action.

Giả sử, bạn muốn viết một tiện ích nhỏ. Công cụ tự động tạo Git Tafs khi số phiên của package.json thay đổi. Bạn sẽ thấy ý nghĩa của thao tác này rất lớn. Đặc biệt là khi truy tìm code ngược về bản phát hành.

* + - * 1. **Github Package Registry**

Package Registry trong GitHub là gì? Tính năng này cho phép lập trình viên duy trì Distribution Registries cá nhân. Trong đó bao gồm npm, docker, maven, nuget và Ruby gems.

Vì thế, bạn nên tạo cho mình một tài khoản GitHub ngay bây giờ. Đây chính là cơ hội tạo những dự án riêng và chia sẻ với mọi người.

* + - * 1. **Mở rộng quan hệ**

Khi biết đến GitHub là gì đồng nghĩa với việc mở rộng mối quan hệ với người cùng chuyên ngành. Bởi lẽ, có hàng ngàn developer toàn cầu đang tham gia mạng lưới rộng lớn. Họ chia sẻ kinh nghiệm, đóng góp những ý tưởng độc đáo.

Git cho phép User chia sẻ code, text fragments và bất kỳ thông tin nào với các dev khác. Do đó, bạn có thể sử dụng để trao đổi text hay gists work như git repositories, từ đó tách ra và update.

* + - 1. **GitHub dùng để làm gì?**

Trong phần dưới đây, bạn sẽ được giải đáp rõ hơn về câu hỏi GitHub dùng để làm gì. Thông qua những tính năng chuyên sâu giúp bạn chủ động hơn khi khám phá công cụ. Đặc biệt là biết mình nên ưu tiên tập trung khai thác mảng nào.

* + - * 1. **Kéo và thả code**

Khi tìm hiểu GitHub là gì, bạn sẽ thấy tính năng Gist cho phép lưu trữ các đoạn code. Các đoạn mã với số lượng lớn có thể được tìm thấy và phê duyệt bằng nhiều ngôn ngữ. Thao tác sử dụng vô cùng dễ dàng và trực quan.

Tuy vậy, có một tính năng hỗ trợ không phải Dev nào cũng biết. Người dùng có thể thêm code trực tiếp từ các tệp tin trong máy tính. Thao tác rất đơn giản bằng hành động kéo, thả là đã được copy tức thì.

* + - * 1. **Tạo** **folder qua giao diện web**

Rất nhiều người quản lý repo thông qua ứng dụng GitHub miễn phí. Ngoài ra, công cụ cũng đã xây dựng một tính năng hỗ trợ gọi là Web Flow.

Nhờ thế, việc theo dõi, lưu trữ repo thông qua giao diện web Git dễ dàng hơn nhiều. Với đặc tính này, mọi người cảm thấy thích thú và tiện lợi hơn khi biết GitHub để làm gì.

* + - * 1. **Git URL Shortener**

Sử dụng URL Shortener trong GitHub là gì? Mọi người vẫn luôn có xu hướng chia sẻ mọi thứ từ hình ảnh, trạng thái đến tin tức của mình trên Social. Các lập trình viên có lẽ cũng mong muốn như vậy.

Họ thích chia sẻ repo GitHub của mình để nhiều người khác biết đến. Tuy vậy, URL đôi khi lại quá dài cho việc share. Chẳng hạn, Twitter chỉ chấp nhận chứa tối đa 140 ký tự. Hiểu được điều đó, Git.io đã ra đời để hỗ trợ nhu cầu của bạn.

* + - * 1. **Tìm tập tin/file**

Như đã biết khi tìm hiểu GitHub là gì, mọi thao tác đều diễn ra nhanh chóng. Khả năng điều hướng và tìm kiếm bất kỳ repo nào cũng không ngoại lệ.

Nhiều người chưa biết đến tính năng này do bị ẩn và chỉ hoạt động khi dùng phím tắt. Để kích hoạt, bạn chỉ cần nhấn phím T rồi next ↑ và ↑ nhảy qua các tệp lên và xuống. Một cách khác nữa là nhập tên file sẽ xuất hiện cụ thể tức thì.

* + - * 1. **Sử dụng Github Emoji**

Trước khi biết GitHub là gì, mọi người thường nghĩ đến hàng triệu dòng code nhàm chán. Thế những, tương tự như mạng xã hội, nền tảng cũng tích hợp một số Emoji.

Nhờ thế, việc review hoặc bình luận các đoạn code trở nên sinh động hơn. Bạn có thể khám phá thông qua bảng mã trong Cheat Emoji.

* + - * 1. **Sử dụng Github CLI**

Sử dụng dạng CLI trong GitHub là gì? Như đã biết, hầu hết mọi người thích làm việc bằng cách dùng GUI. Thế nhưng, việc chọn lựa giao diện dòng lệnh CLI cũng không hiếm.

Chức năng này được bắt đầu với từ khóa “hub”. Trong đó, bạn sẽ thấy bổ sung đầy đủ các cú pháp của Git.

* + - * 1. **Linking Lines**

Sự tiện lợi nằm ở việc chỉ ra và chia sẻ các dòng code cụ thể trong file bất kỳ. Bạn được phép làm điều này bằng thao tác thêm tham số #L kết hợp dòng ở cuối URL. Ngoài ra, bạn cũng có thể chọn một số các dòng bằng cách chỉ định nơi bắt đầu và kết thúc.

* + - * 1. **Task Checklist**

Trong phần giải thích sự tiện lợi của GitHub là gì, bạn đã thấy tích hợp sẵn của nhiều Markdown. Người dùng dễ dàng thêm list các checkbox trong pull request bằng việc sử dụng – [ ] hoặc – [x].

Tuy nhiên, một lưu ý là các cú pháp chỉ xuất hiện trong một danh sách các item. Những [ ] dấu hiệu phải được bắt đầu với một dấu gạch ngang -.

Dành cho những ai chưa có nhiều thời gian khám phá GitHub là gì, công cụ hỗ trợ file CSV. Nếu bạn include một tệp ở dạng này sẽ được hiển thị thành định dạng dữ liệu.

Thậm chí, công cụ còn cho phép tìm kiếm thông qua đó. Không những vậy, Git cũng tự động hiển thị bản đồ với định dạng geoJSON và 3D với phần mở rộng STL.

* + - * 1. **Get Octodex**

Điều cuối cùng nhưng cũng không kém phần quan trọng khi tìm hiểu GitHub là gì? Công cụ có nhiều phiên bản linh vật. Trong đó, Octodex là một tập hợp các version sáng tạo thay thế cho Octocat.

Tại đây, người dùng tìm thấy Female Codertocat, Octo Liberty, Spider Cat, Megacat, và một loạt các Octocat thú vị khác. Các User có thể sử dụng để làm hình đại diện cá nhân.

* 1. **Xây dựng mô hình ứng dụng**

Mô hình xây dựng website này dựa trên mô hình ba lớp MVC (Model – View – Controller).



Hình 2. 3: Mô hình MVC

MVC là một mẫu kiến trúc phần mềm để tạo lập giao diện người dùng trên máy tính. MVC chia một ứng dụng thành ba phần tương tác được với nhau để tách biệt giữa cách thức mà thông tin được xử lý nội hàm và phần thông tin được trình bày và tiếp nhận từ phía người dùng. Mô hình MVC giúp phát triển phần mềm một cách cô lập các nguyên tắc và nghiệp vụ và giao diện người dùng một cách rõ ràng hơn, thuận lợi cho việc phát triển và bảo trì.

* Model là thành phần tượng trưng cho dữ liệu của chương trình phần mềm, nó là nơi tương tác trực tiếp với dữ liệu hoặc hệ quản trị cơ sở dữ liệu. Nó bao gồm các class chứa các phương thức kết nối với database, truy vấn, thêm/xóa/sửa dữ liệu.
* View là thành phần tượng trung cho giao diện tương tác trực tiếp với người dùng, nó được cấu tạo từ những file HTML, CSS,... và đảm nhận việc hiển thị dữ liệu.
* Controller là thành phần được gọi là bộ điểu khiển. Nó làm nhiệm vụ xử lý logic, xử lý các yêu cầu từ phía người dùng người dùng, làm trung gian kết nối giữa Model và View.
  + 1. **Luồng xử lý thông tin**

Diagram

Description automatically generated

Hình 2. 4: Luồng xử lí thông tin của mô hình MVC

* Đầu tiên, client sẽ gửi một request tới server thông qua Controller.
* Controller tiếp nhận request, và xử lý luồng đi tiếp theo của yêu cầu. Trong trường hợp chỉ chuyển từ trang này sang trang khác mà không yêu cầu gửi và nhận dữ liệu thì lúc này Controller sẽ redirect(chuyển hướng) lại cho browser và kết thúc luồng. Lúc này luồng xử lý sẽ đi theo thứ tự 1>6>7> 8.
* Nếu request của người dùng yêu cầu phải xử lý dữ liệu thì Controller gọi xuống Model để lấy dữ liệu. Lúc này Model sẽ tương tác với database để lấy dữ liệu, dữ liệu được Model gửi về Controller, Controller gọi đến View phù hợp với request kèm theo dữ liệu cho View, View sẽ lắp dữ liệu tương ứng vào HTML và gửi lại một HTML cho Controller sau khi thực hiện xong nhiệm vụ của mình. Cuối cùng Controller sẽ trả kết quả về Browser. Lúc này luồng xử lý sẽ đi từ 1 > 2 > 3 > 4 > 5 > 6 > 7 > 8.
* Nếu request của người dùng chỉ yêu cầu dữ liệu nhưng không chuyển trang thì thay vì Controller sẽ trả dữ liệu về cho View thì Controller sẽ respone lại cho trình duyệt thông qua API, dữ liệu trả về thường sẽ là dạng JSON. Luồng xử lý lúc này sẽ là 1 > 2 > 3 > 4 > 5 > 8.
  + 1. **Ưu điểm**
* Tải nhanh, tối ưu lượng băng thông: MVC không sử dụng viewstate nên sẽ giúp tiết kiệm băng thông cho website. Người dùng có thể sử dụng ứng dụng trên web cần tương tác để gửi và nhận dữ liệu một cách liên tục. Điều này cũng giúp giảm băng thông cho website hoạt động tốt cũng như ổn định hơn.
* Kiểm tra dễ dàng hơn: Với MVCC, bạn có thể dễ dàng thực hiện các công việc như kiểm tra, rà soát lỗi phần mềm trước khi đưa tới tay người tiêu dùng. Việc này đảm bảo chất lượng và độ uy tín sản phẩm cao hơn.
* Có bộ control ưu việt: Trên các nền tảng website thì ngôn ngữ lập trình như CSS, HTML, JavaScript có vai trò vô cùng quan trọng. Việc sử dụng mô hình MVC sẽ giúp bạn có bộ control ưu việt trên nền tảng ngôn ngữ hiện đại với nhiều hình thức khác nhau.
* View và size: View là nơi dùng để lưu trữ dữ liệu. Khi có yêu cầu được thực hiện càng lớn thì càng kích thích tệp lớn hơn. Khi đó, tốc độ đường truyền mạnh cũng bị giảm. Việc sử dụng mô hình MVC sẽ giúp bạn tiết kiệm được diện tích băng thông một cách tối ưu.
* Chức năng SOC (Separation of Concern): Chức năng này cho phép bạn có thể phân tách rõ ràng các thành phần như model, data, giao diện, nghiệp vụ.
* Kết hợp: Mô hình MVC cho phép bạn code thoải mái trên nền tảng web. Điều này giúp máy chủ giảm tải được khá nhiều.
* Đơn giản: Đây là mô hình có kết cấu tương tối đơn giản. Bạn không cần có chuyên môn cao vẫn có thể sử dụng được.

* + 1. **Nhược điểm**

Ngoài các ưu điểm nổi bật bên trên, mô hình MVC cũng có những hạn chế nhất định, cụ thể như:

* Cần nhiều thời gian để tìm hiểu nếu muốn áp dụng đúng đắn.
* Tốn kém thời gian và công sức để quản lý tổ chức file.
* Xây dựng quy trình tương đối phức tạp, bởi vậy không cần thiết áp dụng mô hình này cho các dự án nhỏ.

**CHƯƠNG III: CÀI ĐẶT THỬ NGHIỆM**

* 1. **Phương pháp nghiên cứu**
* Khảo sát tình hình chung về nhu cầu sử dụng Website để quản lý hệ thống thông tin trường học: Các trường Trung học phổ thông, Đại học đều đang chuyển hướng cho mình về hệ thống Website quản lý thông tin học sinh sinh viên. Đặc biệt là cấp bậc Đại học, website quản lý ngoài mục đích quản lý ra thì nó còn là bộ mặt của Trường nhằm nhiều mục đích “Marketing” khác. Chính vì tính đa dạng mục đích sử dụng nên xu hướng mỗi trường đều có riêng cho mình một thậm chí là hai website để sử dụng với mục đích khác nhau.
* Đưa ra nhận xét tình hình thực tiễn hiện tại
* Lợi ích: Quản lý thông tin, giới thiệu,…
* Ảnh hưởng: Hầu như là không có
* Giải pháp và mục tiêu cho sản phẩm: Ngoài các chức năng cơ bản như Quản lý học phần, quản lý điểm thi, quản lý thông tin tuyển sinh, chương trình đào tạo, đăng ký lịch học, học phần,… thì sản phẩm cần có tính đột phá, có thêm nhưng tính năng mới phù hợp với tình hình thực tế; Ví dụ như hướng đến chức năng tương tác giữa nhà trường và sinh viên trong quá trình tuyển sinh bằng hình thức chat trực tuyết, video-call hay có thể lấy tài liệu học tập,…

## **Phương pháp thực hiện**

* + 1. **Khảo sát thực tiễn**

Qua thời gian khảo sát các Website quản lý hệ thống của các cổng thông tin giáo dục nói chung cũng như các trường Đại học nói riêng, em đã rút ra được một số vấn đề như sau:

* Các trang web đều cung cấp các thông tin đa dạng: Chương trình đào tạo, Giảng viên, Sinh viên, Điểm thi,…
* Cung cấp cho người dùng giao diện dễ sử dụng nhiều chức năng đơn giản như đăng ký tài khoản trên trang web, tìm kiếm thông tin nhanh chóng, nhận thông tin từ web thông qua email,…
* Liên kết với các trang mạng xã hội để người dùng dễ dàng nhận thông tin khi có cập nhật mới.

Đối với đề tài của em **“**đối tượng sử dụng**”** gồm “quản lý hệ thống” và người dùng qua quá trình khảo sát trên có thể đánh giá các nền tảng phù hợp với để tài, tìm hiểu và phát triển dựa trên sự giúp đỡ hướng dẫn từ Công ty thực tập sẽ giúp cho việc thiết kế website cũng như thi công triển khai làm website trở nên nhanh chóng và tiết kiệm được chi phí .

* + 1. **Phân tích hệ thống**

Diagram

Description automatically generated

Hình 3. 1: Sơ đồ ERD

Diagram

Description automatically generated

Hình 3. 2: Sơ đồ tổ chức Database

Hệ thống sẽ cài đặt cho 2 đối tượng người sử dụng là khách hàng và quản lý cửa hàng với các tính năng như sau:

**Diagram

Description automatically generated**

Hình 3. 3: Sơ đồ tổng quát hệ thống

* + - 1. **Đăng nhập**

Sinh viên và người quản lý đều có thể đăng nhập vào hệ thống bằng tài khoản và mật khẩu. Tài khoản của người quản lý được nhà trường cung cấp. Tài khoản của sinh viên thì được tạo tự động từ hệ thống quản lý sinh viên. Khi người quản lý đăng nhập vào hệ thống thì hệ thống hiện ra giao diện dành cho người quản lý, có thêm các chức năng quản lý. Sinh viên đăng nhập vào hệ thống thành công thì hệ thống hiện ra giao diện cho sinh viên.

Diagram

Description automatically generated

Hình 3. 4: Usecase đăng nhập

Bảng 3. 1: Usecase đăng nhập

|  |  |
| --- | --- |
| **Use case name** | Đăng nhập |
| **Participating actor** | Sinh viên, Người quản lý |
| **Flow of events** | B1: Người dùng nhập vào tài và mật khẩu vào giao diện.  B2: Người dùng nhấn đăng nhập  B3: Hệ thống kiểm tra tài khoản có tồn tại hay không, không thì báo đăng nhập không thành công, yêu cầu khách hàng nhập lại, nếu đúng thì dựa vào vai trò của tài khoản để xuất hiện giao diện theo vai trò đó. |
| **Entry condition** | Có tài khoản của hệ thống |
| **Exit condition** | Không có |
| **Quallty requirements** | UC cho phép người dùng đăng nhập vào hệ thống |

* + - 1. **Quên mật khẩu**

Người dùng nhấn vào nút quên mật khẩu khi người dùng quên mật khẩu tài khoản. Sau đó hệ thống hiện ra cho người dùng nhập email sinh viên hoặc người quản lý. Sau khi nhập và xác nhận email thì hệ thống hiện gửi mã xác nhận đến email đó và hiện ra giao diện nhập mã xác nhận. Sau đó người dùng nhập mã xác nhận vào form và nhấn xác nhận, sau đó hệ thống xác nhận độ chính xác của mã xác nhận, nếu đúng thì hiện ra form nhập mật khẩu mới, sai thì yêu cầu gửi lại mã xác nhận. Sau đó người dùng nhập vào mật khẩu mới, nếu hợp lệ thì hệ thống thay đổi nếu không yêu cầu nhập lại.

**Diagram

Description automatically generated**

Hình 3. 5: Usecase quên mật khẩu

Bảng 3. 2: Usecase quên mật khẩu

|  |  |
| --- | --- |
| **Use case name** | Quên mật khẩu |
| **Participating actor** | Sinh viên |
| **Flow of events** | B1: Sinh viên nhấn vào nút quên mật khẩu.  B2: Hệ thống hiện ra modal để nhập email sinh viên.  B3: Sinh viên nhập email vào và nhấn gửi mã xác nhận.  B4: hệ thống gửi mã xác nhận vào email sinh viên đó và hiện ra Modal để nhập mã xác nhận.  B5: Sinh viên nhập vào mã xác nhận đã nhận đc ở email và nhấn xác nhận.  B6: Hệ thống kiểm tra mã xác nhận chính xác hay chưa, nếu không đúng yêu cầu nhập lại email và gửi lại mã xác nhận, ngược lại sẽ hiện ra form nhập mật khẩu mới.  B7: Sinh viên nhập vào mật khẩu mới và xác nhận.  B8: Hệ thống thay đổi mật khẩu của sinh viên thành công và thông báo ra màn hình. |
| **Entry condition** | Người dùng quên mật khẩu của tài khoản |
| **Exit condition** | Không có |
| **Quallty requirements** | Sinh viên đổi được mật khẩu |

* + - 1. **Quản lý học phần**

Khi đăng nhập vào hệ thống với chức năng của người quản lý thì sẽ có chức năng quản lý học phần trên header. Khi nhấn vào thì giao diện hiện ra là ra khoa của trường cho người quản lý lựa chọn.

**Diagram

Description automatically generated**

Hình 3. 6: Usecase quản lý học phần

Bảng 3. 3: Usecase quản lý học phần

|  |  |
| --- | --- |
| **Use case name** | Quản lý học phần |
| **Participating actor** | Người quản lý |
| **Flow of events** | B1: Người quản lý nhấn vào chức năng quản lý học phần  B2: Hệ thống hiện ra các khoa của trường  B3: Người quản lý có thể chọn 1 khoa  B4: Hệ thống hiện ra các học phần thuộc khoa đã chọn  B5: Người quản lý thao tác chức năng trên hệ thống, hệ thống sẽ xử lý yêu cầu theo chức năng. |
| **Entry condition** | Đã được đăng nhập với tư cách là người quản lý và nhấn vào nút quản lý học phần |
| **Exit condition** | Không có |
| **Quallty requirements** | Người quản lý thấy được các khoa, và các học phần của khoa, thao tác xử lý chức năng. |

* + - 1. **Xem thông tin học phần**

Chức năng này đều có thể sử dụng được bởi sinh viên và người quản lý. Cho phép người dùng chọn khoa có học phần cần xem. Sau đó chọn vào học phần cần xem chi tiết thông tin.

**Diagram

Description automatically generated**

Hình 3. 7: Usecase xem thông tin học phần

Bảng 3. 4: Usecase xem học phần

|  |  |
| --- | --- |
| **Use case name** | Xem học phần |
| **Participating actor** | Sinh viên – người quản lý |
| **Flow of events** | B1: Sinh viên – người quản lý nhấn vào tab xem học phần  B2: Hệ thống tìm kiếm và hiện ra các khoa của trường lên.  B3: Người dùng chọn khoa có học phần cần xem.  B4: Hệ thống tìm kiếm và hiện các học phần của khoa đó  B5: Người dùng chọn học phần cần xem  B6: Hệ thống hiện ra giao diện thông tin chi tiết học phần. |
| **Entry condition** | Đã đăng nhập thành cồng vào hệ thống và có chức năng là sinh viên |
| **Exit condition** | Không có |
| **Quallty requirements** | Hiện ra các khoa của trường, các học phần của khoa và thông tin chi tiết của học phần |

* + - 1. **Thêm học phần**

Chức năng này được hiển thị trong phần quản lý học phần, có nghĩa là chỉ có người quản lý có quyền chỉnh sửa thông tin học phần. Sau khi vào chức năng quản lý học phần người quản lý chọn chức năng thêm học phần. Hệ thống hiện ra giao diện nhập thông tin học phần mới. Người quản lý nhập vào thông tin học phần mới và nhấn thêm. Hệ thống hiện ra thông báo xác nhận.

Diagram

Description automatically generated

Hình 3. 8: Usecase thêm học phần

Bảng 3. 5: Usecase thêm học phần

|  |  |
| --- | --- |
| **Use case name** | Thêm học phần |
| **Participating actor** | Người quản lý |
| **Flow of events** | B1: Người quản lý chọn chức năng quản lý học phần  B2: Hệ thống hiện ra các năm học học kỳ  B3: Người quản lý chọn năm học học kỳ  B4: Hệ thống hiện ra các khoa học phần thuộc năm học học kỳ đó  B5: Người quản lý chọn thêm học phần cho khoa đã chọn  B6: Hệ thống hiển thị form học phần và yêu cầu nhập thông tin  B7: Người quản lý cần nhập thông tin học phần  B8 Người quản lý chọn thêm để hệ thống lưu vào cơ sở dữ liệu |
| **Entry condition** | Đã đăng nhập thành công vào hệ thống và có chứ năng là người quản lý |
| **Exit condition** | Không có |
| **Quallty requirements** | Người quản lý thêm học phần thành công |

* + - 1. **Sửa thông tin học phần**

Chức năng này được hiển thị trong phần quản lý học phần, có nghĩa là chỉ có người quản lý có quyền chỉnh sửa thông tin học phần. Sau khi vào chức năng quản lý học phần người quản lý chọn chức năng chỉnh sửa thông tin học phần. Sau đó sẽ có giao diện nhập thông tin học phần mới hiện ra, người quản lý nhập thông tin học phần mới rồi xác nhận thay đổi.

**Diagram

Description automatically generated**

Hình 3. 9: Usecase sửa học phần

Bảng 3. 6: Usecase sửa thông tin học phần

|  |  |
| --- | --- |
| **Use case name** | Sửa học phần |
| **Participating actor** | Người quản lý |
| **Flow of events** | B1: Người quản lý vào chức năng chức năng quản lý học phần  B2: Người quản lý chọn chức năng sửa học phần  B3: Người quản lý chọn học phần cần sửa  B4: Người quản lý nhập vào thông tin mới của học phần  B5: Hệ thống xác nhận độ chính xác và lưu vào csdl và hiện ra màn hình thông báo thành công |
| **Entry condition** | Đã đăng nhập thành công vào hệ thống |
| **Exit condition** | Tài khoản có chức năng là người quản lý |
| **Quallty requirements** | Người quản lý sửa đổi được thông tin của học phần. |

* + - 1. **Tìm kiếm học phần**

Khi vào chức năng quản lý học phần hoặc xem thông tin học phần đều có chức chức năng tìm kiếm học phần mở góc trên bên phải màn hình.. Người dùng có thể nhập vào từ khoá liên quan đến học phần cần tìm kiếm và nhấn tìm kiếm. Hệ thông tìm kiếm các học phần liên quan đến từ khoá người dùng cần tìm và hiện ra màng hình.

**Diagram

Description automatically generated**

Hình 3. 10: Usecase tìm kiếm học phần

Bảng 3. 7: Usecase tìm học phần

|  |  |
| --- | --- |
| **Use case name** | Tìm kiếm điểm |
| **Participating actor** | Sinh viên và người quản lý |
| **Flow of events** | B1: Sinh viên và người quản lý chọn chức năng tìm kiếm điểm và nhập vào từ khoá liên quan đến học phần, lớp học phần cần tìm kiếm điểm  B2: Hệ thống tìm ra kết quả học tập có thông tin liên quan đến từ khoá và xuất ra màn hình các kết quả học tập đó |
| **Entry condition** | Đã đăng nhập thành công vào hệ thống |
| **Exit condition** | Không có |
| **Quallty requirements** | Sinh viên – người quản lý tìm kiếm học phần cá nhân |

* + - 1. **Quản lý điểm**

Khi đăng nhập vào hệ thống với chức năng của người quản lý thì sẽ có chức năng quản lý điểm trên header. Khi nhấn vào thì giao diện hiện ra các lớp học phần có đang diễn trong trong đợt học hiện tại, người quản lý có thể chọn các đợt học khác ở trong combobox để hiện các lớp học phần khác ở các đợt học khác. Người quản lý chọn 1 lớp học phần để có thể thấy điểm của các sinh viên tham gia lớp học phần đó.

**Diagram

Description automatically generated**

Hình 3. 11: Usecase quản lý điểm

Bảng 3. 8: Usecase quản lý điểm

|  |  |
| --- | --- |
| **Use case name** | Quản lý điểm |
| **Participating actor** | Người quản lý |
| **Flow of events** | B1: Người quản lý chọn chức năng Quản lý điểm  B2: Hệ thống hiện ra giao diện để người quản lý chọn năm học học kỳ cần xem điểm  B3: Người quản chọn năm học học kỳ và nhấn xác nhận  B4: Hệ thống tìm kiếm tất cả các điểm của sinh viên đó trong năm học học kỳ đó và hiện ra màn hình |
| **Entry condition** | Đã đăng nhập thành công vào hệ thống |
| **Exit condition** | Không có |
| **Quallty requirements** | Người quản lý thực hiện quản lý điểm học phần, thao tác thực hiện trong quản lý điểm |

* + - 1. **Xem điểm**

Chỉ có sinh viên mới có chức năng xem điểm học phần cá nhân. Sau khi đăng nhập sinh viên sẽ có chức năng xem điểm học phần cá nhân. Sau khi vào chức năng xem điểm sẽ hiện ra 1 danh sách các học phần mà sinh viên đó có điểm, có thể sắp xếp theo đợt học. Khi nhấn vào xem chi tiết thì hệ thống hiện ra thông tin chi tiết điểm của học phần đó của sinh viên đó

**Chart, funnel chart

Description automatically generated**

Hình 3. 12: Usecase xem điểm

Bảng 3. 9: Usecase xem điểm

|  |  |
| --- | --- |
| **Use case name** | Xem điểm |
| **Participating actor** | Sinh viên |
| **Flow of events** | B1: Sinh vien chọn chức năng xem điểm  B2: Hệ thống hiện ra giao diện để sinh viên chọn năm học học kỳ cần xem điểm  B3: Sinh viên chọn năm học học kỳ và nhấn xác nhận  B4: Hệ thống tìm kiếm tất cả các điểm của sinh viên đó trong năm học học kỳ đó và hiện ra màn hình  B5: Sinh viên chọn học phần cần xem điểm  B6: Hệ thống hiện thông tin chi tiết điểm học phần đó. |
| **Entry condition** | Đã đăng nhập thành công vào hệ thống |
| **Exit condition** | Không có |
| **Quallty requirements** | Sinh viên xem thông tin chi tiết của điểm học phần |

* + - 1. **Thêm điểm**

Chức năng này được hiển thị trong phần quản lý điểm, có nghĩa là chỉ có người quản lý có quyền thêm điểm học phần. Sau khi vào chức năng quản lý điểm, người quản lý sẽ chọn lớp học phần có sinh viên cần thêm điểm. Sau đó chọn sinh viên cần thêm điểm. Hệ thống sẽ hiện ra giao diện nhập điểm. Người quản lý nhập xong thông tin điểm và xác nhận thêm.

**Diagram

Description automatically generated**

Người quản lý

Hình 3. 13: Usecase thêm điểm

Bảng 3. 10: Usecase thêm điểm

|  |  |
| --- | --- |
| **Use case name** | Thêm điểm |
| **Participating actor** | Người quản lý |
| **Flow of events** | B1: Người quản lý chọn chức năng quản lý điểm.  B2: Hệ thống hiện ra các năm học học kỳ  B3: Người quản lý chọn năm học học kì  B4: Hệ thống hiện ra các lớp học phần thuộc năm học học kỳ đó  B5: Người quản lý chọn lớp học cần thêm điểm  B6: Hệ thống hiện ra các sinh viên của lớp học đó  B7: Người quản lý chọn sinh viên cần thêm điểm và nhấn chức năng thêm điểm  B8: Hệ thống hiện ra giao diện cho người quản lý nhập thông tin điểm  B9: Người quản lý nhấn nút thêm  B10: Hệ thống hiện ra thông báo xác nhận, nếu người quản lý xác nhận thì hệ thống kiểm tra thông tin và lưu vào csdl, nếu không thì quay trở lại. |
| **Entry condition** | Đã đăng nhập thành công vào hệ thống và có chứ năng là người quản lý |
| **Exit condition** | Không có |
| **Quallty requirements** | Người quản lý thêm điểm thành công |

* + - 1. **Sửa điểm**

Chức năng này được hiển thị trong phần quản lý điểm, có nghĩa là chỉ có người quản lý có quyền sửa điểm học phần của sinh viên. Sau khi vào chức năng quản lý điểm, người quản lý sẽ chọn lớp học phần có sinh viên cần sửa điểm. Sau đó chọn sinh viên cần sửa điểm. Hệ thống sẽ hiện ra giao diện nhập thông tin điểm mới. Người quản lý nhập xong thông tin điểm mới và xác nhận thay đổi.

**Diagram

Description automatically generated**

Người quản lý

Hình 3. 14: Usecase sửa điểm

Bảng 3. 11: Usecase chỉnh sửa điểm

|  |  |
| --- | --- |
| **Use case name** | Chỉnh sửa điểm |
| **Participating actor** | Người quản lý |
| **Flow of events** | B1: Người quản lý chọn chức năng quản lý điểm.  B2: Hệ thống hiện ra các năm học học kỳ  B3: Người quản lý chọn năm học học kì  B4: Hệ thống hiện ra các lớp học phần thuộc năm học học kỳ đó  B5: Người quản lý chọn lớp học cần chỉnh sửa điểm  B6: Hệ thống hiện ra các sinh viên của lớp học đó  B7: Người quản lý chọn sinh viên cần sửa điểm và nhấn chức năng sửa điểm  B8: Hệ thống hiện ra giao diện cho người quản lý nhập thông tin điểm mới  B9: Người quản lý nhấn nút chỉnh sửa  B10: Hệ thống hiện ra thông báo xác nhận, nếu người quản lý xác nhận thì hệ thống kiểm tra thông tin và lưu vào csdl, nếu không thì quay trở lại. |
| **Entry condition** | Đã đăng nhập thành công vào hệ thống và có chứ năng là người quản lý |
| **Exit condition** | Không có |
| **Quallty requirements** | Người quản lý sửa điểm thành công |

* + - 1. **Tìm kiếm điểm**

Khi vào chức năng xem điểm hoặc chức năng quản lý điểm đều có thanh tìm kiếm điểm ở góc trên bên phải màng hình. Người dùng có thể nhập vào từ khoá liên quan đến điểm học phần cần tìm kiếm và nhấn tìm kiếm. Hệ thông sẽ tìm kiếm thông tin điểm liên quan đến từ khoá rồi hiện ra màn hình.

**Chart

Description automatically generated with medium confidence**

Hình 3. 15: Usecase tìm kiếm điểm

Bảng 3. 12: Usecase tìm kiếm điểm

|  |  |
| --- | --- |
| **Use case name** | Tìm kiếm điểm |
| **Participating actor** | Sinh viên – người quản lý |
| **Flow of events** | B1: Sinh viên chọn chức năng tìm kiếm điểm và nhập vào từ khoá liên quan đến học phần, lớp học phần cần tìm kiếm điểm  B2: Hệ thống tìm ra kết quả học tập có thông tin liên quan đến từ khoá và xuất ra màn hình các kết quả học tập đó |
| **Entry condition** | Đã đăng nhập thành công vào hệ thống |
| **Exit condition** | Không có |
| **Quallty requirements** | Sinh viên – người quản lý tìm kiếm điểm thành công |

* + - 1. **Quản lý đăng ký học phần.**

Khi đăng nhập vào hệ thống với chức năng của người quản lý thì sẽ có chức năng quản lý đăng ký học phần trên header. Sau khi nhấn vào chức năng quản lý đăng ký học phần. Hệ thống sẽ hiện ra giao diện quản lý đăng ký học phần, với các lớp học phần đang mở hoặc chờ được mở đăng lý học phần. Có mở lớp, xắp sếp thời gian theo ngày giờ để cho phép sinh viên có thể đăng ký học phần.

**Diagram

Description automatically generated**

Hình 3. 16: Usecase quản lý đăng ký học phần

Bảng 3. 13: Usecase quản lý đăng ký học phần

|  |  |
| --- | --- |
| **Use case name** | Quản lý đăng ký lớp học phần |
| **Participating actor** | Người quản lý |
| **Flow of events** | B1: Người quản lý nhấn vào chức năng quản lý đăng ký học phần  B2: Hệ thống hiện ra danh sách của các lớp học phần đang trong thời gian cho phép đăng ký  B3: Người quản lý có thể nhấn vào combobox để sắp xếp theo các lớp còn nhiều vị trí trống nhất |
| **Entry condition** | Đã đăng nhập thành công vào hệ thống và có chức năng là người quản lý |
| **Exit condition** | Không có |
| **Quallty requirements** | Hiện ra các lớp học phần đang trong thời gian đăng ký, thao tác chức năng trong quản lý đăng ký học phần |

* + - 1. **Đăng ký học phần**

Chức năng đăng ký học phần chỉ có sinh viên có thể sử dụng được. Sau khi đăng nhập vào hệ thống, sinh viên sẽ có chức năng đăng ký học phần. Sau khi nhấn vào hệ thống sẽ hiện ra giao diện các lớp có học phần có trong danh sách học phần của sinh viên đó trong đợt học, học kỳ này, và là các lớp chưa đủ sinh viên tham gia. Sinh viên chọn 1 lớp học phần. Hệ thống hiện ra giao diện thông tin chi tiết của lớp học phần đó và nút đăng ký tham gia. Sinh viên nhán vào nút đăng ký tham gia và và hệ thống hiện ra thông báo xác nhận đăng ký.

**Chart

Description automatically generated**

Hình 3. 17: Usecase đăng ký học phần

Bảng 3. 14: Usecase đăng ký học phần

|  |  |
| --- | --- |
| **Use case name** | Đăng ký học phần |
| **Participating actor** | Sinh viên |
| **Flow of events** | B1: Sinh viên mục đăng ký học phần  B2: Thông tin các học phần có thể đăng ký hiện ra  B3: Sinh viên cho lớp cần đăng ký học phần.  B4: Hệ thống hiện ra thông tin của lớp học phần đó nút xác nhận để đăng ký  B5: Sinh viên nhập thông tin và đăng ký thành công |
| **Entry condition** | Đã đăng nhập thành công vào hệ thống |
| **Exit condition** | Không có |
| **Quallty requirements** | Sinh viên đăng ký học phần thành công |

* + - 1. **Huỷ đăng ký học phần**

Chức năng huỷ đăng ký học phần chỉ có sinh viên có thể sử dụng được. Sau khi đăng nhập vào hệ thống, sinh viên sẽ có chức năng huỷ đăng ký học phần. Sau khi nhấn vào chức năng này, hệ thống sẽ hiện ra các lớp học phần mà sinh viên đó đã đăng ký học phần và lớp đó phải còn trong trong thời gian cho phép đăng ký học phần. Sinh viên sẽ chọn đơn đăng ký học phần cần huỷ. Hệ thống hiện ra giao diện cần xác nhận huỷ. Sinh viên xác nhận và huỷ đăng ký học phần.

**Diagram

Description automatically generated**

Hình 3. 18: Usecase hủy đăng ký học phần

Bảng 3. 15: Usecase hủy đăng ký học phần

|  |  |
| --- | --- |
| **Use case name** | Huỷ đăng ký học phần |
| **Participating actor** | Sinh viên |
| **Flow of events** | B1: Sinh viên chọn chức năng quản lý đăng ký học phần cá nhân  B2: Sinh viên chọn lớp đã đăng ký học phần(còn trong thời gian cho phép huỷ)  B3: Nhấn nút huỷ đăng ký.  B4: Hệ thống hiện ra thông báo xác nhận, nếu sinh viên nhấn xác nhận thì hệ thống huỷ đăng ký học phần đó và cập nhập xuống csdl, nếu huỷ bỏ thì quay trở lại |
| **Entry condition** | Đã đăng nhập thành công vào hệ thống |
| **Exit condition** | Còn trong thời gian cho phép huỷ đăng ký |
| **Quallty requirements** | Sinh viên huỷ đăng ký học phần thành công |

* + - 1. **Quản lý lớp học phần**

Khi đăng nhập vào hệ thống với chức năng của người quản lý thì sẽ có chức năng quản lý lớp học phần trên header. Sau khi nhấn vào chức năng quản lý lớp học phần. Hệ thống sẽ hiện ra giao diện quản lý lớp học phần. Giao diện sẽ hiển thị tất cả các lớp học phần trong đợt học hiện tại. Ngoài ra còn có thể chọn hiển thị cho các đợt học khác.

**Diagram

Description automatically generated**

Hình 3. 19: Usecase quản lý lớp học phần

Bảng 3. 16: Usecase quản lý lớp học phần

|  |  |
| --- | --- |
| **Use case name** | Quản lý lớp học học phần |
| **Participating actor** | Người quản lý |
| **Flow of events** | B1: Người quản lý chọn chức năng quản lý lớp học học phần.  B2: Hệ thống hiện ra giao diện để người quản lý chọn năm học học kỳ cần xem lớp học  B3: Người quản lý chọn năm học học kỳ cần xem lớp học  B4: Hệ thống tìm kiếm các lớp học học phần có trong học kỳ đó. |
| **Entry condition** | Đã đăng nhập thành công vào hệ thống |
| **Exit condition** | Không có |
| **Quallty requirements** | Người quản lý quản lý lớp theo đợt học |

* + - 1. **Xem lịch học**

Chức năng này chỉ có thể được sử dụng bởi sinh viên. Sinh viên sau khi đăng nhập vào hệ thống thì sẽ có chức năng xem thời khoá biểu. Sinh viên vào chức năng xem thời khoá biểu thì có thể thấy thời khoá biểu của đợt học hiện tại. Ngoài ra còn có thể xem của các học kỳ, đợt học khác.

**Diagram

Description automatically generated**

Hình 3. 20: Usecase xem lịch học

Bảng 3. 17: Usecase xem lịch học

|  |  |
| --- | --- |
| **Use case name** | Xem lịch học |
| **Participating actor** | Sinh viên |
| **Flow of events** | B1: Sinh viên chọn chức năng xem lịch học cá nhân.  B2: Hệ thống hiện ra giao diện để sinh viên chọn năm học học kỳ cần xem lịch học  B3: Sinh viên chọn năm học học kỳ cần xem lịch học  B4: Hệ thống tìm kiếm các học phần mà sinh viên đó có trong học kỳ đó, các lớp học đã đăng ký và chưa đăng ký, thời gian học, ca học để hiện ra màn hình. |
| **Entry condition** | Đã đăng nhập thành công vào hệ thống |
| **Exit condition** | Không có |
| **Quallty requirements** | Sinh viên tìm kiếm lịch học cá nhân tuỳ vào năm học học kì |

* + - 1. **Thêm lớp học phần**

Chức năng này được hiển thị trong phần quản lý lớp học phần, có nghĩa là chỉ có người quản lý có quyền thêm lớp học phần. Sau khi vào chức năng quản lý lớp học phần, hệ thống sẽ hiện ra giao diện quản lý lớp học phần. Người quản lý sẽ nhấn vào nút thêm lớp học phần. Hệ thống sẽ hiển thị ra giao diện thêm thông tin của lớp học phần sau đó nhấn thêm. Hệ thống kiểm ra và hiện ra thông báo xác nhận.

**Diagram

Description automatically generated**

Hình 3. 21: Usecase thêm lớp học phần

Bảng 3. 18: Usecase thêm lớp học phần

|  |  |
| --- | --- |
| **Use case name** | Thêm lớp học học phần |
| **Participating actor** | Người quản lý |
| **Flow of events** | B1: Người quản lý chọn chức năng quản lý lớp học phần  B2: Hệ thống hiện ra các năm học học kỳ  B3: Người quản lý chọn năm học học kì  B4: Hệ thống hiện ra các lớp học phần thuộc năm học học kỳ đó  B5: Người quản lý chọn thêm lớp học phần  B6: Hệ thống hiện ra form thêm lớp học phần  B7: Người quản lý nhập thông tin lớp học  B8: Người quản lý nhấn nút thêm  B9: Hệ thống hiện ra thông báo xác nhận, nếu người quản lý xác nhận thì hệ thống kiểm tra thông tin và lưu vào csdl, nếu không thì quay trở lại. |
| **Entry condition** | Đã đăng nhập thành công vào hệ thống |
| **Exit condition** | Không có |
| **Quallty requirements** | Người quản lý thêm lớp học phần thành công |

* + - 1. **Sửa lớp học phần**

Chức năng này được hiển thị trong phần quản lý lớp học phần, có nghĩa là chỉ có người quản lý có quyền sửa thông tin của lớp học phần. Sau khi vào chức năng quản lý lớp học phần, hệ thống sẽ hiện ra giao diện quản lý lớp học phần. Sau đó người quản lý chọn 1 lớp học phần cần sửa thông tin lớp học phần. Sau đó hệ thống sẽ hiện ra giao diện nhập thông tin lớp học phần mới. Người quản lý nhập xong và thay đổi, hệ thống hiện ra thông báo xác nhận thay đổi. Sau khi nay đổi tất cả các thông tin của việc đăng ký lớp học phần này cũng sẽ thay đổi theo.

**Diagram

Description automatically generated**

Hình 3. 22: Usecase sửa lớp học phần

Bảng 3. 19: Usecase sửa lớp học

|  |  |
| --- | --- |
| **Use case name** | Chỉnh lớp học |
| **Participating actor** | Người quản lý |
| **Flow of events** | B1: Người quản lý chọn chức năng quản lý lớp học phần  B2: Hệ thống hiện ra các năm học học kỳ  B3: Người quản lý chọn năm học học kì  B4: Hệ thống hiện ra các lớp học phần thuộc năm học học kỳ đó  B5: Người quản lý chọn lớp học cần chỉnh sửa điểm  B6: Hệ thống hiện ra các sinh viên của lớp học đó  B7: Người quản lý chọn sinh viên cần sửa điểm và nhấn chức năng sửa điểm  B8: Hệ thống hiện ra giao diện cho người quản lý nhập thông tin điểm mới  B9: Người quản lý nhấn nút chỉnh sửa  B10: Hệ thống hiện ra thông báo xác nhận, nếu người quản lý xác nhận thì hệ thống kiểm tra thông tin và lưu vào csdl, nếu không thì quay trở lại. |
| **Entry condition** | Đã đăng nhập thành công vào hệ thống và có chứ năng là người quản lý |
| **Exit condition** | Không có |
| **Quallty requirements** | Người quản lý sửa thông tin lớp học thành công |

* + - 1. **Quản lý chương trình đào tạo**

Khi đăng nhập vào hệ thông với quyền truy cập là người quản lý thì mới có thể sử dụng chức năng này, người quản lý vào chức năng quản lý chương trình đào tạo sẽ bao gồm xem chương trình đào tạo theo khoa, năm học học kì. Nếu muốn thêm chương trình đào tạo, người quản lý cũng có thể thêm với chức năng tương tụ.

**Diagram

Description automatically generated**

Hình 3. 23: Usecase quản lý chương trình đào tạo

Bảng 3. 20: Usecase quản lý chương trình đào tạo

|  |  |
| --- | --- |
| **Use case name** | Quản lý Chương trình đào tạo |
| **Participating actor** | Người quản lý |
| **Flow of events** | B1: Người quản lý chọn chức năng quản lý Chương trình đào tạo  B2: Hệ thống hiện ra giao diện để người quản lý chọn năm học học kỳ cần xem chương trình  B3: Người quản lý thực hiện thao tác chức năng trên màn hình với Chức năng Quản lý chương trình đào tạo. |
| **Entry condition** | Đã đăng nhập thành công vào hệ thống |
| **Exit condition** | Không có |
| **Quallty requirements** | Người quản lý quản lý chương trình đào tạo. |

* + 1. **Thiết kế - hoàn thiện giao diện**
       1. **Giao diện đăng nhập**

Graphical user interface, application

Description automatically generated

Hình 3. 24: Giao diện đăng nhập

* + - 1. **Giao diện quên mật khẩu**

Graphical user interface, application, website

Description automatically generated

Hình 3. 25: Giao diện quên mật khẩu

* + - 1. **Giao diện quản lý điểm**

Graphical user interface, application

Description automatically generated

Hình 3. 26: Giao diện quản lý điểm

* + - 1. **Giao diện thông tin điểm sinh viên**

Graphical user interface, text, application, email

Description automatically generated

Hình 3. 27: Giao diện thông tin điểm sinh viên

* + - 1. **Giao diện quản lý học phần**

Graphical user interface, website

Description automatically generated

Hình 3. 28: Giao diện quản lý học phần

* + - 1. **Giao diện thông tin danh sách học phần**

Graphical user interface, application

Description automatically generated

Hình 3. 29: Giao diện thông tin học phần thuộc khoa Viễn thám

* + - 1. **Giao diện thêm học phần**

Graphical user interface, text, application, Teams

Description automatically generated

Hình 3. 30: Giao diện thêm học phần

Graphical user interface, text, application, email

Description automatically generated

Hình 3. 31: Giao diện sửa học phần

* + - 1. **Giao diện quản lý lớp học phần**

Graphical user interface, application

Description automatically generated

Hình 3. 32: Giao diện quản lý lớp học phần

Graphical user interface, text, application

Description automatically generated

Hình 3. 33: Giao diện tìm kiếm lớp học phần

* + - 1. **Giao diện thêm lớp học phần**

Graphical user interface, text, application, email, Teams

Description automatically generated

Hình 3. 34: Giao diện thêm lớp học phần

* + - 1. **Giao diện sửa lớp học phần**

Graphical user interface, text, application, Teams

Description automatically generated

Hình 3. 35: Giao diện sửa lớp học phần

* + - 1. **Giao diện mở lớp đăng ký học phần**

Graphical user interface, diagram

Description automatically generated with medium confidence

Hình 3. 36: Giao diện mở lớp đăng ký học phần

* + - 1. **Giao diện sinh viên đăng ký học phần**

Graphical user interface, application

Description automatically generated

Hình 3. 37: Giao diện sinh viên đăng kí học phần

* + - 1. **Giao diện danh sách chương trình đào tạo**

Graphical user interface, text, application

Description automatically generated

Hình 3. 38: Giao diện danh sách chương trình đào tạo

* + - 1. **Giao diện thêm chương trình đào tạo**

Graphical user interface, text, application, Teams

Description automatically generated

Hình 3. 39: Giao diện thêm chương trình đào tạo

* + - 1. **Giao diện học phần**

Graphical user interface, website

Description automatically generated

Hình 3. 40: Giao diện học phần sinh viên đang có trong đợt học

* + - 1. **Giao diện lịch học cá nhân của sinh viên**

Graphical user interface, application, Word

Description automatically generated

Hình 3. 41: Giao diện lịch học cá nhân

**CHƯƠNG IV: KẾT LUẬN**

## **Kết luận**

### **Kết quả đạt được**

Sau quá trình nghiên cứu và phát triển, xây dựng đề tài, em đã đạt được những kết quả sau:

* Tìm hiểu được nghiệp vụ của một hệ thống quản lý học phần sinh viên bao gồm: Quản lý đăng kí học phần, Quản lý điểm, Quản lý lịch học/lớp học.
* Đã xây dựng được phần mềm quản lý học phần sinh viên với các chức năng xoay quanh nghiệp vụ quản lý học phần: Xem thông tin học phần, điểm, lịch học; Đăng ký học phần; Quản lý điểm, lớp học, lịch học.
* Chương trình có khả năng phân quyền người dùng linh hoạt.
* Gửi thông báo về Email sinh viên.

### **Đánh giá kết quả đạt được**

#### **Ưu điểm**

* Website có giao diện thân thiện, thuận tiện cho người sử dụng.
* Hỗ trợ nhiều chức năng phù hợp với thực tế của việc tìm kiếm thông tin, chỉnh sửa và các chức năng quản lý.
* Có nhiều chức năng xử lý linh hoạt các yêu cầu thực tế.
* Dễ dàng sử dụng và quản lý.
* Cơ sở dữ liệu tương đối lớn và tính bảo mật cao, mật khẩu được mã hóa dưới dạng BcryptPassword.

#### **Nhược điểm**

* Khả năng, năng lực còn hạn chế nên một số vẫn chỉ còn trên ý tưởng, chưa được thực thi.
* Một số chức năng còn chưa hoạt động ổn định.
* Chưa giải quyết trọn vẹn các vấn đề phát sinh trong quá trình quản lý.
* Chương trình có tính thẩm mỹ chưa cao.

### **Kiến thức đạt được**

* Củng cố kiến thức về JavaScrip, HTML và CSS
* Áp dụng mô hình MVC vào sản phẩm
* Sử dụng GitHub

### **Những tích lũy đạt được trong quá trình thực tập**

Qua quá trình thực tập em đã nghiên cứu và học hỏi được rất nhiều về cả kiến thức chuyên ngành cũng như các kỹ năng lập trình. Em đã củng cố cho bản thân nhiều kiến thức quan trọng như là: Sử dụng thành thạo ngôn ngữ lập trình JAVA, HTML, CSS…, thành thạo hơn về cơ sở dữ liệu MySQL, các kỹ thuật lập trình trình, lập trình theo mô hình MVC.

Cùng với đó, em cũng học hỏi thêm được Javascript, biết sử dụng các công cụ như Github, các gói thư viện hữu ích như jQuery và được hướng dẫn về cách xử lý các tình huống thực tế. Ngoài ra em còn biết được thêm cách tự nghiên cứu tài liệu, biết được nhiều website cung cấp các tài liệu uy tín. Bên cạnh nội dung báo cáo trên, trong quá trình thực tập em còn được tham gia nghiên cứu Sources code, và tiếp cận với những sản phẩm đã có của Trung tâm. Thực tập không những cung cấp kiến thức chuyên ngành mà em còn học được cách giao tiếp, trao đổi kết nối với các anh chị trong Công ty, phát triển những kỹ năng mềm.

Để nói về những gì đóng góp được cho cơ quan thực tập thì chưa thực sự chưa có gì đáng kể ngoài việc hoàn thành các công việc được giao. Em vẫn đang nô lực rèn luyện thêm để có thể tham gia vào quy trình phát triển ứng dụng của công ty sau này. Những điều mà em có thể rút ra được sau quá trình thực tập đó là luôn cần có thái độ tốt, ham học hỏi, nhiệt tình trong công việc, tự tin và tăng khả năng giao tiếp, quan tâm tới mọi người. Sau quá trình thực tập với những gì nghiên cứu và làm được em cố gắng tiếp tục nghiên cứu thêm và phát triển nó thêm để hoàn thiện hơn. Mong rằng có thể sử dụng trong tương lai.

* 1. **Hướng phát triển**
* Xây dựng, cải thiện giao diện thân thiện với người dùng hơn.
* Hoàn thiện các chức năng còn thiếu.
* Tìm hiểu, xây dựng nhưng chức năng thực tế từ bên ngoài đưa vào hệ thống.
* Sử dụng linh hoạt dữ liệu, các câu lệnh truy vấn để tăng tốc độ xử lý khi sử dụng hệ thống.
* Đưa phần mềm ứng dụng vào được thực tế và thu thập ý kiến của người dùng để càng hoàn thiện sản phẩn một cách tối ưu nhất.
* Tăng tính bảo mật.
* Cải thiện tốc độ truy cập.
* Đẩy sản phẩm lên Localhost.
* Mở rộng thành Website hệ thống trường Đại học có tích hợp Quản lý học phần sinh viên.

# **TÀI LIỆU THAM KHẢO**

**Tiếng Việt:**

1. Hồ Hoàng Triết, *Căn bản thiết kế Web HTML JavaScrip ASP*, Nhà xuất bản thống kê 2002.
2. Phạm Trung Kiên, *Kỹ thuật lập trình Web với Ajax*, Nhà xuất bản Hồng Đức 2007.
3. Thạc Bình Cường – Vũ Thị Hậu, *Giáo trình thiết kế web*, Nhà xuất bản Giáo dục Việt Nam, 2011.
4. Charles Wyke Smith, *Định kiểu web với CSS*, Nhà xuất bản Trí thức, 2013
5. Jole Murach , *Lập Trình Cơ Bản PHP Và MySQL,* Nhà xuất bản Khoa học kỹ thuật 2013.
6. Đậu Quang Tuấn, *Tự học thiết kế web bằng Java Script*, Nhà xuất bản Giao thông vận tải 2006
7. *Think Java – Có suy nghĩ như nhà khoa học máy tính*, Giáo trình tự do theo giấy phép Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License
8. Đoàn Văn Ban, *Lập trình hướng đối tượng với Java*, Nhà xuất bản khoa học và kỹ thuật
9. Trịnh Anh Toàn, *Thiết kế trang web ấn tượng và hiệu quả,* Nhà xuất bản Trẻ

**Tiếng Anh :**

1. Jon Duckett, JavaScript & JQuery: Interactive Front-End Web Development, Publisher: Wiley; 1 edition (July 21,2014).
2. Daniel Schneller – Udo Schwedt ,*MySQL Admin Cookbook,* Packt Publishing (March 30, 2010)
3. Josh Long – Kenny Bastani, *Cloud Native Java* ,O'Reilly Media September 19, 2017
4. Jennifer Robbins*, Learning Web Design*, O'Reilly Media June 19, 2018.
5. Joshua, *Effective Java*, December 27, 2017.
6. Cay Horstmann, *Core Java: Fundamentals, Volume 1 (Oracle Press Java)*, Oracle Press December 21, 2021.
7. Robert C. Martin, *Clean Code: A Handbook of Agile Software Craftsmanship*, Pearson August 1, 2008.
8. Eric Freeman – Elisabeth Robson, *Head First Design Patterns: Building Extensible and Maintainable Object-Oriented Software 2nd Edition*, O'Reilly Media January 12, 2021.
9. Craig Walls, *Spring in Action*, Manning November 5, 2018.
10. Kathy Sierra – Bert Bates, *Head First Java*, O'Reilly Media March 15, 2005.
11. Herbert Schildt, *Java: A Beginner's Guide, Eighth Edition*, McGraw Hill November 2, 2018.
12. Ivelin Demirov, *Learn JavaScript Visually*, Hardcover – July 18, 2014