|  |  |
| --- | --- |
| logo (CMYK)-01 | BỘ GIÁO DỤC VÀ ĐÀO TẠO  **TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM** |

**ĐỒ ÁN CƠ SỞ**

**XÂY DỰNG CỔNG THÔNG TIN**

**CÁC THÔNG TIN TẠI ĐẠI HỌC CÔNG NGHỆ (HUTECH)**

Ngành: **CÔNG NGHỆ THÔNG TIN**

Chuyên ngành: **CÔNG NGHỆ PHẦN MỀM**

Giảng viên hướng dẫn : TS. Lê Thị Ngọc Thơ

Sinh viên thực hiện : **Lê Khôi Đệ**

MSSV: 1411010102 Lớp: 15DTH06

Sinh viên thực hiện : **Nguyễn Bá Tâm**

MSSV: 1611061022 Lớp: 16DTHC2

TP. Hồ Chí Minh, 2019

**LỜI CAM ĐOAN**

Nhóm em xin cam đoan những điều được viết trong báo cáo đều là những thành quả, quá trình lao động không ngừng nghỉ trong 12 tuần. Đều là những ý tưởng thô sơ được nhóm em nghĩ ra và nghiên cứu

Đề tài “Xây dựng cổng thông tin các thông tin tại đại học công nghệ (Hutech)” một ý tưởng xuất phát điểm là từ bản thân. Vì trong quá trình học tập và rèn luyện tại trường thì Điểm rèn luyện luôn chiếm một vai trò trong việc đánh giá một sinh viên. Sau khảo sát 100 bạn tại trường Đại Học Công Nghệ với câu hỏi “Bạn thường tham gia các hội thảo, sự kiện tại trường qua đâu?” thì 70% các bạn đều trả lời là qua mạng xã hội, 20% là qua các băng rôn được treo tại trường, còn lại là truyền miệng.

Tại sao đa số các sự kiện của trường đều được biết qua mạng xã hội, còn trang Website của trường lại không được cập nhật đầy đủ để sinh viên có thể theo dõi. Để trả lời câu hỏi trên thì nhóm em đã đi tìm hiểu và nghiên cứu thì thấy được rất nhiều vấn đề.

Trang Website của trường thường chỉ cập nhật các sự kiện tổ chức hàng năm hay các sự kiện nổi tiếng như Hutech Got Talent,.. Và mỗi sự kiện từng khoa thường nằm trong từng danh mục cho thấy thiếu thân thiện và thiếu tiện lợi. Còn thông qua mạng xã hội thì từng trang riêng lẻ, nhóm riêng lẻ hay bị bỏ sót nếu không chịu cập nhật thường xuyên.

Xuất phát từ những lý do trên, cùng với sự hướng dẫn của cô Lê Thị Ngọc Thơ, nhóm em thực hiện đề tài “Xây dựng cổng thông tin các thông tin tại đại học công nghệ (Hutech)” để vận dụng những kiến thức đã được học vào thực tiễn, xây dựng một hệ thống cập nhật sự kiện các thông tin tại trường Hutech.

**MỤC LỤC**

**[LỜI CAM ĐOAN 2](#_Toc21813)**

**[CHƯƠNG 1. TỔNG QUAN 5](#_Toc1310)**

[1.1 Giới thiệu đề tài 5](#_Toc9575)

[1.2 Nhiệm vụ đồ án 7](#_Toc20209)

*[1.2.1 Lý do hình thành đề tài](#_Toc9172)* [7](#_Toc9172)

*[1.2.2 Mục tiêu nghiên cứu ,phạm vi dự án](#_Toc7828)* [7](#_Toc7828)

*[1.2.3 Ý nghĩa thực tiễn](#_Toc14414)* [8](#_Toc14414)

[1.3 Cấu trúc đồ án 9](#_Toc10918)

**[CHƯƠNG 2. CƠ SỞ LÝ THUYẾT 10](#_Toc15546)**

[2.1 Mô tả công nghệ 10](#_Toc5140)

[2.2 Mô tả hệ thống - Các ràng buộc - Mô hình - Lý giải mô hình 13](#_Toc19056)

[2.2.1](#_Toc657) *[Chức năng chính](#_Toc657)* [13](#_Toc657)

*[2.2.2 Đặc tả yêu cầu](#_Toc22337)* [15](#_Toc22337)

[2.2.1.1 Đặc tả yêu cầu hệ thống 15](#_Toc30180)

[2.2.1.2 Đặc tả yêu cầu giao diện 15](#_Toc18450)

[2.2.1.3 Đặc tả yêu cầu chức năng 15](#_Toc31416)

*[2.2.3 UseCase](#_Toc3308)* [18](#_Toc3308)

*[2.2.3.1 Mô hình Usecase](#_Toc9929)* [18](#_Toc9929)

*[2.2.3.2 Danh sách Actor](#_Toc15091)* [19](#_Toc15091)

*[2.2.3.3 Danh sách Usecase](#_Toc8104)* [19](#_Toc8104)

*[2.2.3.4 Đặc tả mô hình Usecase](#_Toc31390)* [22](#_Toc31390)

*[2.2.4 Mô hình](#_Toc18540)* [30](#_Toc18540)

[2.2.4.1 Mô hình thực thể kết hợp ERD 30](#_Toc5412)

*[2.2.4.2 BFD - Business Function Diagram](#_Toc22018)* [31](#_Toc22018)

*[2.2.4.3 DFD - Data Flow Diagram](#_Toc521)* [32](#_Toc521)

**[CHƯƠNG 3. KẾT QUẢ THỰC NGHIỆM 36](#_Toc18178)**

[3.1 Mô tả các công việc thực nghiệm mà đề tài đã tiến hành 36](#_Toc7706)

[3.2 Kết quả nghiên cứu lý thuyết - thực nghiệm đạt được 39](#_Toc25398)

*[3.2.1 Kết quả nghiên cứu lý thuyết :](#_Toc18476)* [39](#_Toc18476)

*[3.2.2 Kết quả thực nghiệm đạt được](#_Toc13826)* [39](#_Toc13826)

[3.3 Kết quả sản phẩm phần mềm 40](#_Toc29583)

*[3.3.1 Hồ sơ thiết kế và cài đặt](#_Toc1905)* [40](#_Toc1905)

*[3.3.2 Giao diện](#_Toc284)* [48](#_Toc284)

**[CHƯƠNG 4. KẾT LUẬN VÀ KIẾN NGHỊ 51](#_Toc25428)**

[4.1 Kết quả đạt được 51](#_Toc16752)

[4.2 Những đóng góp 52](#_Toc389)

[4.3 Kết luận chung 54](#_Toc14330)

**[TÀI LIỆU THAO KHẢO 55](#_Toc24235)**

**CHƯƠNG 1. TỔNG QUAN**

* 1. **Giới thiệu đề tài**

Với mỗi trường đại học, việc xây dựng cập nhật và lưu trữ tất cả các thông tin về các hoạt động xoay quanh các lĩnh vực hỗ trợ việc học tập cho sinh viên là điều không thể không có nhất là khi chúng ta đang trong thời kì cách mạng công nghệ 4.0 thì việc luôn cập nhật thông tin là điều thiết yếu đối với mỗi sinh viên đang học tập và công tác tại các trường đại học.

Nhận thấy hầu hết các thông tin liên quan đến việc hỗ trợ học tập cho sinh viên ở các trường đại học nói chung và trường đại học công nghệ nói riêng đều nằm riêng lẻ cả các trang mạng xã hội, các group tập thể và cũng chỉ số ít người có liên quan trong các group hay các trang đó mới cập nhận được thông tin cần thiết còn phần đông nó cũng chưa được công khai hay có một cách nào đó để tất cả các sinh viên trong trường có thể cập nhật được.

Nắm bắt được thực trạng và nhu cầu thiết yếu của hầu hết sinh viên trong trường muốn tích lũy điểm rèn luyện, muốn tham gia tìm hiểu và bám sát các hướng đi của doanh nghiệp, muốn tham gia các sự kiện các hoạt động vui chơi giải trí để giảm áp lực học tập cũng như giao lưu học hỏi..v..v...Sau một thời gian bàn bạc và đưa ra các tình huống các giả thuyết thì nhóm em đã quyết định xây dựng một trang web riêng chuyên sàng lọc và cập nhật các thông tin cần thiết nhất dành cho các bạn sinh viên mang tên “ Cổng thông tin tại trường Đại Học Công Nghệ TP.HCM(Hutech)”.

**Project Charter**

Project Title : Xây dựng cổng thông tin các thông tin tại Đại Học Công Nghệ (Hutech)

Project Start Date : 01/04/2019 Projected Finish Date : 22/06/2019

Project Manager : Lê Khôi Đệ - 0782657635 - lekhoide@gmail.com

Project Objectives : Thiết kế và xây dựng Website “Cổng thông tin các thông tin tại Đại Học Công Nghệ (Hutech)” trên nền tảng công nghệ ASP.NET MVC 5 cho Đồ Án Cơ Sở hướng dẫn bởi giảng viên Lê Thị Ngọc Thơ trong vòng 12 tuần.

Approach :

* Xác định, phân tích và thiết kế trong 3 tuần đầu tiên. Thực hiện và kiểm thử trong 8 tuần. Vận hành và báo cáo kết quả cho giáo viên hướng dẫn 1 tuần còn lại.
* Mỗi tuần nộp báo cáo cho giáo viên hướng dẫn và nhận yêu cầu cho tuần tiếp theo.

Roles and Responsibilities :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Role | Name | Organization / Position | Contact Information | Mssv / Class |
| Lập kế hoạch cho nhóm  Thiết kế giao diện  Nghiên cứu công nghệ mới | LÊ KHÔI ĐỆ | Project Manager  Designer  Technical Master | 0782657635  lekhoide@gmail.com | 1411010102  15DTH06 |
| Thu thập và phân tích thông tin  Kiểm thử chức năng  Viết code | NGUYỄN BÁ TÂM | Bussiness Analyst  Tester  Programmer | 0333058893  tamnguyen.tgth@gmail.com | 1611061022  16DTHC2 |

Sign-off: (Signatures of all above stakeholders. Can sign by their names in table above.)

LÊ KHÔI ĐỆ

NGUYỄN BÁ TÂM

1. **Nhiệm vụ đồ án**
2. *Lý do hình thành đề tài*

Dự án “Cổng thông tin tại trường Đại Học Công Nghệ TP.HCM(Hutech)” là một dự án được xây dựng lên nhằm hỗ trợ sinh viên thấy được nhiều thông tin hơn, ở đây là các hội thảo, sự kiện, thông tin về các câu lạc bộ, hội nhóm, các cuộc thi được phân chia riêng lẻ ở 7 Trung Tâm, 8 Viện, 11 Phòng Ban, 13 Khoa khác nhau giờ đây sẽ được tổng hợp và thấy được qua dự án. Do đó sẽ giải quyết được rất nhiều bài toán mà sinh viên đang mắc phải khi cập nhật các thông tin, sự kiện tại trường.

1. *Mục tiêu nghiên cứu ,phạm vi dự án*

**Tên dự án :** Xây dựng cổng thông tin, thông tin tại Đại Học Công Nghệ (Hutech)

**Viết bởi :** Lê Khôi Đệ

**Ngày :** 11/04/2019

**Lý giải về dự án :**

Dự án “Xây dựng cổng thông tin, thông tin tại Đại Học Công Nghệ (Hutech)” được xây dựng nhằm hỗ trợ sinh viên thấy được nhiều thông tin hơn, thông tin ở đây là các hội thảo, sự kiện, câu lạc bộ, đội nhóm được chia ra riêng lẻ ở 7 Trung Tâm, 8 Viện, 11 Phòng Ban, 13 Khoa và nhiều nhỏ lẻ khác mà giờ đây sẽ được tổng hợp qua một Website. Qua phân tích khảo sát 100 bạn sinh viên Hutech, thì nhóm em rút được ba vấn đề cơ bản sau :

1. Đa số sinh viên tham gia hội thảo, sự kiện, câu lạc bộ,… Thứ nhất để cộng điểm rèn luyện, thứ hai vì niềm yêu thích, tò mò. Nhưng hiện nay đa số các hội thảo, sự kiện được cập nhật trên Facebook mà ở từng trang khác nhau, nhóm khác nhau gây tình trạng dễ bị bỏ sót nếu không thường xuyên cập nhật
2. Trên Website trường Hutech đa số chỉ cập nhật các hội thảo, sự kiện nổi bật bỏ qua các hội thảo, sự kiện nhỏ lẻ khác và thường nằm trong từng danh mục không phù hợp với người dùng khó tính và thiếu kiên nhẫn.
3. Còn các hội thảo, sự kiện nhỏ lẻ khác thì chỉ được truyền miệng qua các lớp học, các giảng viên, băng rôn từng nơi trong trường, từng cơ sở.

**Các tính chất yêu cầu của sản phẩm :**

1. Giao diện Website ưa nhìn
2. Trang quản trị dễ sử dụng
3. Các thông tin cần phải đầy đủ và rõ ràng, giúp người truy cập có thể tìm hiểu một thông tin một cách dễ dàng.
4. Các chức năng của Website phải dễ dàng sử dụng, tập trung vào những nhu cầu cần thiết và tránh các thủ tục rườm rà không phù hợp với người dùng khó tính và thiếu kiên nhẫn.
5. Thích hợp với màn hình điện thoại di động

**Các yêu cầu để đánh giá sự thành công của dự án :**

Đạt được mục tiêu mà nhóm đã đề ra.

Đáp ứng được thời hạn nộp dự án cho giảng viên hướng dẫn.

Thực hiện được những yêu cầu của giảng viên hướng dẫn giao cho từng tuần.

1. *Ý nghĩa thực tiễn*

Dự án “Cổng thông tin tại trường Đại Học Công Nghệ TP.HCM (Hutech)” được xây dựng nhằm giúp ích cho sinh viên về việc cập nhật các thông tin sự kiện tại trường và hơn hết dự án cũng có thể là một giải pháp để tổng hợp các sự kiện các thông tin tại trường Đại Học Công Nghệ Tp.HCM (Hutech).

1. **Cấu trúc đồ án**

**Chương 1 Tổng quan**

Giới thiệu đề tài

Nhiệm vụ đồ án

Lý do hình thành đề tài

Mục tiêu nghiên cứu, phạm vi giới hạn

Ý nghĩa thực tiễn

Cấu trúc đồ án

**Chương 2 Cơ sở lý thuyết**

Mô tả công nghệ

Mô tả hệ thống - Các ràng buộc - Những mô hình - Lý giải xây dựng mô hình

**Chương 3 Kết quả thực nghiệm**

Mô tả các công việc thực nghiệm đề tài đã tiến hành

Các kết quả nghiên cứu lý thuyết

Kết quả thực nghiệm đạt được

Kết quả sản phẩm phần mềm : hồ sơ thiết kế, cài đặt, giao diện

**Chương 4 Kết luận và kiến nghị**

Khẳng định những kết quả đạt được

Những đóng góp

Kết luận chung

**CHƯƠNG 2. CƠ SỞ LÝ THUYẾT**

1. **Mô tả công nghệ**

Những công nghệ mà nhóm đã áp dụng vào đề tài :

* Công nghệ ASP.NET.MVC 5 của Microsoft
* Công nghệ Entity Framework 6 Code First trong ASP.NET.MVC 5

Những công cụ mà nhóm đã áp dụng vào đề tài :

* Công cụ Visual Studio 2017
* Công cụ Microsoft SQL Server Management Studio 2017
* Công cụ Lucidchart
* Công cụ Trello

Những ngôn ngữ mà nhóm đã áp dụng vào đề tài :

* Ngôn ngữ C#
* Ngôn ngữ HTML
* Ngôn ngữ CSS

1. *Công nghệ ASP.NET.MVC 5 của Microsoft là gì?*

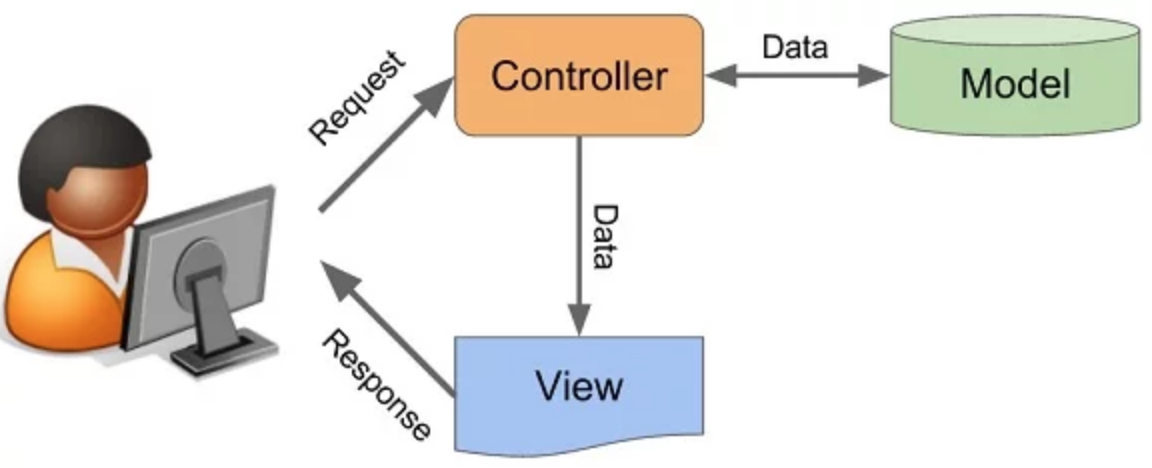
ASP.NET MVC 5 là một Framework lập trình web mới của Microsoft, công nghệ này ứng dụng mô hình MVC vào trong ASP.NET.

MVC là cụm từ viết tắt của Models - View - Controllers :

Models : nơi lưu trữ dữ liệu

View : giao diện người dùng được sử dụng bởi khách truy cập trang Web để nhìn thấy các dữ liệu

Controllers : chịu trách nhiệm xử lý các tương tác của người dùng với trang Web. Xác định loại View nào cần phải hiển thị và giao tiếp với Models.



Lợi ích ứng dụng Web dựa trên mô hình MVC

Dễ dàng quản lý sự phức tạp của ứng dụng bằng cách chia ứng dụng thành ba phần model, view, controller

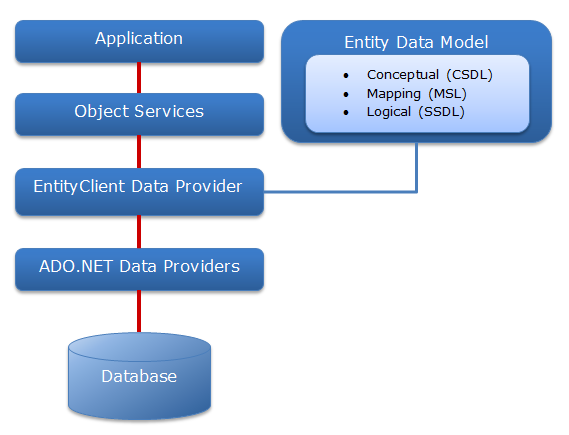
Sử dụng mẫu Front Controller, mẫu này giúp quản lý các requests (yêu cầu) chỉ thông qua một Controller

Hỗ trợ tốt cho các ứng dụng được xây dựng bởi những đội có nhiều lập trình viên và thiết kế mà vẫn quản lý được tính năng của ứng dụng

1. *Công nghệ Entity Framework 6 Code First trong ASP.NET.MVC 5*

Entity Framework là một nền tảng được sử dụng để làm việc với Database thông qua cơ chế ánh xạ Object/Relational Mapping (ORM). Nhờ đó, có thể truy vấn, thao tác với Database gián tiếp thông qua các đối tượng lập trình.

Kiến trúc của Entity Framework được minh họa như sau :



Một vài quy ước của Code-First cho phép tự động cấu hình một Model như sau :

* Quy ước tên bảng: Khi bạn tạo một class mô tả các thực thể là User sẽ được lưu trữ trong database thì Entity Framework mặc định sẽ tạo ra bảng có tên Users.
* Quy ước khóa chính: Khi bạn tạo thuộc tính có tên UserId trong lớp User của model, thuộc tính này được nhận là khóa chính. Hơn nữa, nó sẽ  thiết lập một cột khóa auto-incrementing để lưu trữ giá trị thuộc tính nếu kiểu khóa là số nguyên.
* Quy ước về mối quan hệ: Entity Framework cung cấp các quy ước khác nhau để nhận biết một mối quan hệ giữa 2 models dựa vào tên của thuộc tính và kiểu dữ liệu.

1. **Mô tả hệ thống - Các ràng buộc - Mô hình - Lý giải mô hình**
2. *Chức năng chính*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **NHÓM USECASE** | **USECASE** | **MÔ TẢ** | **ĐỘ KHÓ** |
| 1 | Đăng nhập | Đăng nhập | Xử lý đăng nhập vào hệ thống | Dễ |
| Đăng nhập bằng Google | Xử lý đăng nhập băng google vào hệ thống | Khó |
| Ghi nhớ đăng nhập | Xử lý ghi nhớ đăng nhập | Dễ |
| Quên mật khẩu | Xử lý quên mật khẩu bằng email cá nhân | Khó |
| Đăng xuất | Xử lý đăng xuất tài khoản ra ngoài hệ thống | Dễ |
| 2 | Đăng ký | Đăng ký | Xử lý đăng ký vào hệ thống | Dễ |
| Đăng ký bằng Google | Xử lý đăng ký bằng tài khoản google | Khó |
| 3 | Tìm kiếm | Tìm kiếm theo địa điểm | Xử lý tìm kiếm theo cơ sở của trường | Trung bình |
| Tìm kiếm theo tên sự kiện | Xử lý tìm kiếm theo tên sự kiện | Trung bình |
| Tìm kiếm theo ngày | Thực hiện tìm kiếm sự kiên theo ngày | Trung bình |
| Tìm kiếm theo phòng-ban / khoa / viện/ trung tâm / khác/ tất cả | Thực hiện tìm kiếm sự kiện theo phòng-ban / khoa / viện/ trung tâm / khác | Trung bình |
| 4 | Người dùng quản lý tài khoản cá nhân | Hồ sơ cá nhân | Xử lý xem xóa sửa thông tin của người dùng | Trung bình |
| Đổi mật khẩu | Xử lý yêu cầu đổi mật khẩu người dùng | Dễ |
| Sự kiện đã tạo | Thực hiện hiển thị sự kiện đã tạo của người dùng | Khó |
| Lưu sự kiện | Xử lý lưu trữ sự kiện của người dùng quan tâm | Khó |
| Đăng xuất | Xử lý đăng xuất tài khoản người dùng ra khỏi hệ thống | Dễ |
| 5 | Sự kiện đã tạo | Tạo sự kiện | Xử lý tạo sự kiện của chính người dùng | Khó |
| Chỉnh sửa sự kiện | Xử lý thực hiện yêu cầu chỉnh sửa sự kiện | Khó |
| Xóa sự kiện | Thực hiện yêu cầu xóa sự kiện | Dễ |
| 6 | Quản trị quản lý danh sách tài khoản cá nhân người dùng | Xem danh sách tài khoản cá nhân | Xử lý yêu cầu xem danh sách tất cả các tài khoản người dùng | Dễ |
| Cấm một tài khoản cá nhân truy câp | Xử lý cấm và thông báo người dùng tài khoản đó truy cập | Khó |

1. *Đặc tả yêu cầu*
2. Đặc tả yêu cầu hệ thống

Hệ thống có thể đáp ứng nhiều lượt truy cập cùng lúc

Hệ thống cơ sở dữ liệu có thể lưu trữ một lượng lớn tài khoản người dùng

Tốc độ xử lý nhanh hơn, khả năng truy cập tốt hơn

Sử dụng công nghệ bảo mật tiên tiến

Có sự cập nhật thông tin liên tục,chính xác

1. Đặc tả yêu cầu giao diện

Màu sắc được thiết kế phù hợp với từng loại sự kiện

Thiết kế các thanh công cụ phù hợp và tiện lợi cho người sử dụng

Hình ảnh kích cỡ phù hợp

Font chữ không quá cầu kì dễ nhìn dễ đọc

Sự kiện được thiết kế và sắp xếp tối ưu các nội dung chính cho người truy cập

1. Đặc tả yêu cầu chức năng

**Đăng ký**

Mô tả : Người dùng phải nhập : Họ và tên, Email, Password, Confrim Password. Sau đó người dùng mới tạo được tài khoản.

Quy định : Họ và tên không được để trống, Email phải ghi đúng cấu trúc email (vd: xxx@gmail.com), Password, Confrim Password phải hiện ký tự “\*” khi người dùng nhập để bảo mật.

**Đăng nhập**

Mô tả : Người dùng cần nhập đúng tên đăng nhập và mật khẩu. Sau khi đăng nhập thì người dùng mới có quyền thực hiện chức năng mà nhà quản trị phân quyền.

Quy định : Tên đăng nhập không được để trống và phải giống kiểu cấu trúc email (vd: xxx@gmail.com). Mật khẩu không được để trống và phải hiện ký tự “\*” khi người dùng nhập để bảo mật.

**Tìm kiếm**

Mô tả : Người dùng có thể nhập tên sự kiện, hay chọn thanh danh mục theo Phòng-Ban / Khoa / Viện / Trung tâm / Khác / Tất cả để tìm kiếm sự kiện. Hay chọn danh mục theo Hôm nay / Ngày mai / Tuần này để tìm kiếm sự kiện. Hay chọn danh mục địa điểm cơ sở A / cơ sở B / cơ sở E / cơ sở U để tìm kiếm sự kiện.

Quy định : Có thẻ nhập hoặc chọn thanh danh mục để tìm kiếm không bắt buộc.

**Tạo sự kiện**

Mô tả : Người dùng có thẻ xem, xóa, sửa sự kiện theo ý mình.

Quy định : Chức năng này người dùng phải tạo tài khoản để sử dụng.

**Hồ sơ cá nhân**

Mô tả : Người dùng có thể xem, xóa, sửa thông tin cá nhân của mình.

Quy định : Người dùng phải có tài khoản cá nhân.

**Đổi mật khẩu**

Mô tả : người dùng phải nhập đúng Password cũ, phải nhập Password mới và Confrim Password phải nhập đúng. Sau khi đổi mật khẩu thì tài khoản sẽ quay về Người dùng quản lý tài khoản cá nhân và có thông báo Password đã được thay đổi.

Quy định : Người dùng phải có tài khoản cá nhân. Passoword cũ, Password mới, Confrim Password khi người dùng nhập phải mã hóa kí tự thành “\*”.

**Lưu sự kiện**

Mô tả : Hiện ra danh sách các sự kiện kiện đã lưu, thể hiện chi tiết thông tin sự kiện đó.

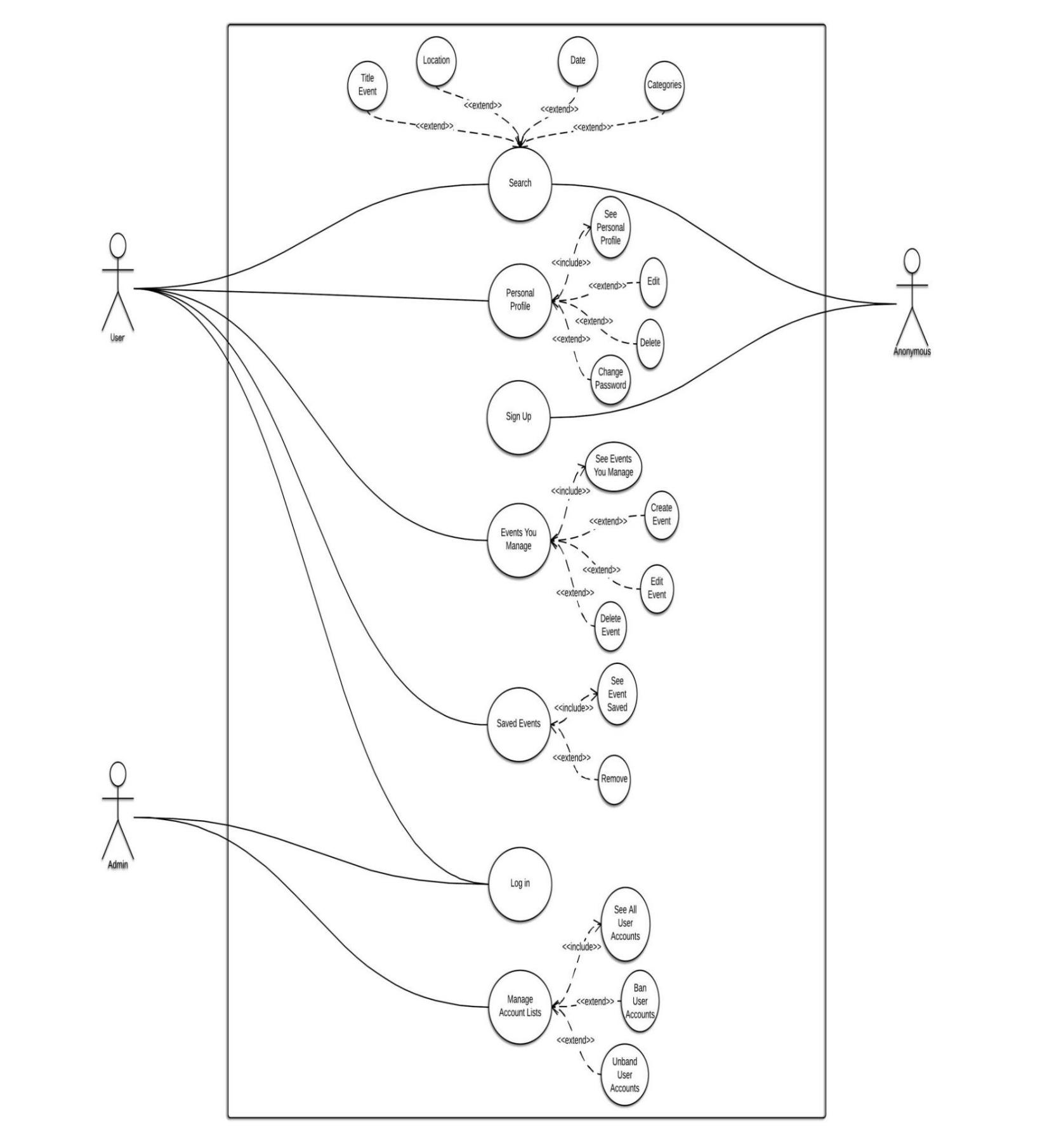
Quy định : Người dùng phải có tài khoản cá nhân, và phải bấm nút lưu sự kiện để thực hiện chức năng. Và chỉ có thể xem và hủy sự kiện đã lưu.

**Quản trị quản lý danh sách tài khoản cá nhân người dùng**

Mô tả : Khi đăng vào tài khoản được phân quyền quản trị. Thì người quản trị có thêm chức năng đã ẩn là xem danh sách tài khoản cá nhân người dùng đã đăng ký. Có quyền cấm một tài khoản khi tài khoản đã đăng nhập và sẽ nhận được thông báo.

Quy định : Chỉ cho phép danh xem danh sách số lượng tài khoản người dùng không được đăng nhập tài khoản đó để thực thi một mục đích khác. Password sẽ được mã hóa thành “\*” để bào mật cho người dùng.

1. *UseCase*
2. *Mô hình Usecase*



1. *Danh sách Actor*

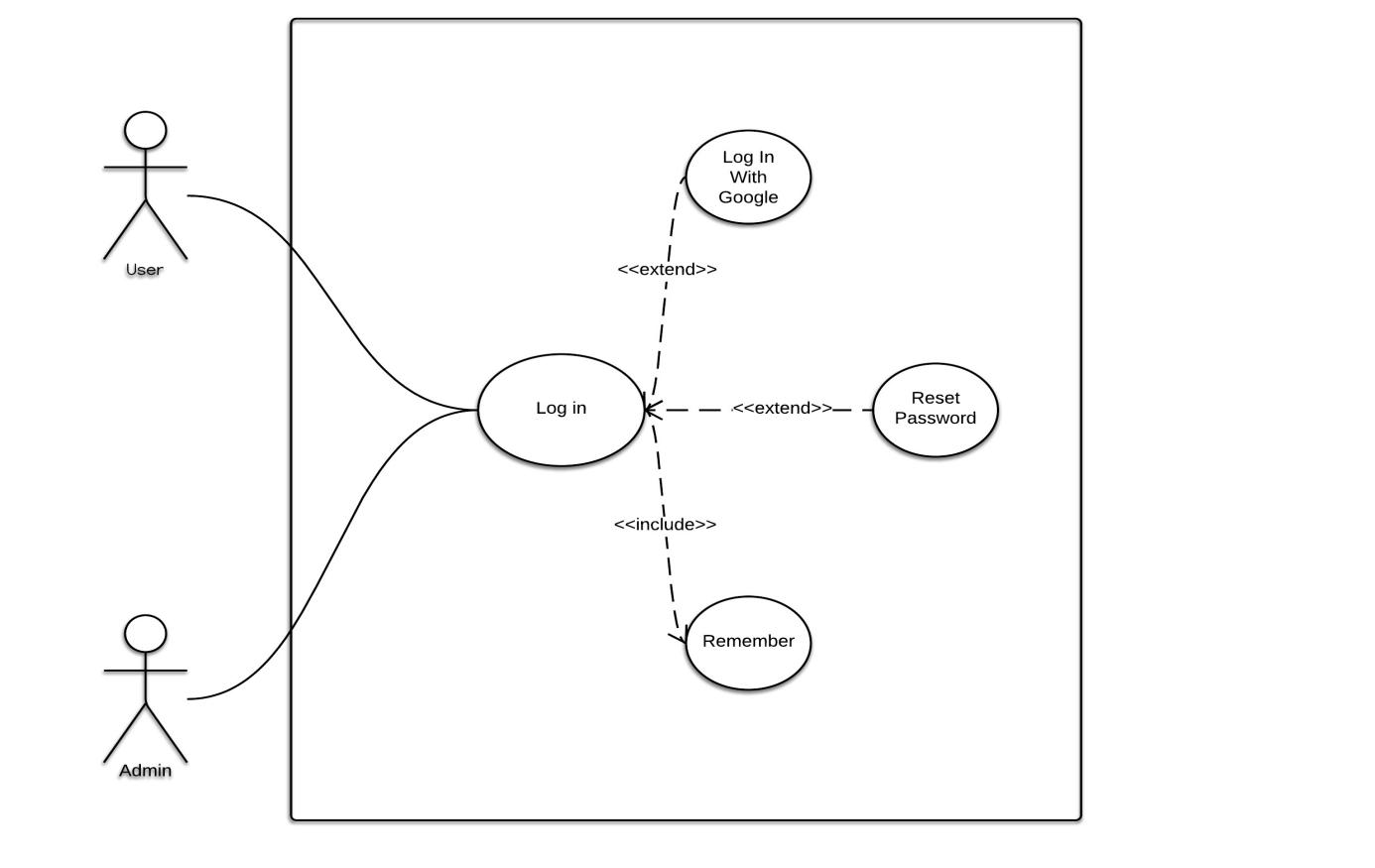
|  |  |  |
| --- | --- | --- |
| **STT** | **Tên Đối Tượng** | **Mô Tả** |
| 1 | Người dùng | Thành viên đã đăng ký tài khoản Website |
| 2 | Anonymous | Người truy cập Website nhưng chưa có tài khoản thành viên |
| 3 | Quản trị viên | Quản lý danh sách tài khoản. |

1. *Danh sách Usecase*

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Nhóm Usecase** | **Usecase** | **Mô Tả** |
| 1 | Đăng nhập | Đăng nhập | Xử lý đăng nhập vào hệ thống |
| Đăng nhập bằng Google | Xử lý đăng nhập băng Google vào hệ thống |
| Ghi nhớ đăng nhập | Xử lý ghi nhớ đăng nhập |
| Quên mật khẩu | Xử lý quên mật khẩu bằng Email cá nhân |
| 2 | Đăng ký | Đăng ký | Xử lý đăng ký vào hệ thống |
| Đăng ký bằng Google | Xử lý đăng ký bằng tài khoản Google |
| 3 | Tìm kiếm | Tìm kiếm theo địa điểm | Xử lý tìm kiếm theo cơ sở của trường |
| Tìm kiếm theo tên sự kiện | Xử lý tìm kiếm theo tên sự kiện |
| Tìm kiếm theo ngày | Thực hiện tìm kiếm sự kiên theo ngày |
| Tìm kiếm theo phòng-ban / khoa / viện/ trung tâm / tất cả | Thực hiện tìm kiếm sự kiện theo phòng-ban/khoa/viện/trung tâm/tất cả |
| 4 | Người dùng quản lý tài khoản cá nhân | Hồ sơ cá nhân | Xử lý xem xóa sửa thông tin của người dùng |
| Đổi mật khẩu | Xử lý yêu cầu đổi mật khẩu người dùng |
| Sự kiện đã tạo | Thực hiện hiển thị sự kiện đã tạo của người dùng |
| Lưu sự kiện | Xử lý lưu trữ sự kiện của người dùng quan tâm |
| Đăng xuất | Xử lý đăng xuất tài khoản người dùng ra khỏi hệ thống |
| 5 | Sự kiện đã tạo | Tạo sự kiện | Xử lý tạo sự kiện của chính người dùng |
| Chỉnh sửa sự kiện | Xử lý thực hiện yêu cầu chỉnh sửa sự kiện |
| Xóa sự kiện | Thực hiện yêu cầu xóa sự kiện |
| 6 | Quản trị quản lý danh sách tài khoản cá nhân người dùng | Xem danh sách tài khoản cá nhân | Xử lý yêu cầu xem danh sách tất cả các tài khoản người dùng |
| Cấm một tài khoản cá nhân truy câp | Xử lý cấm và thông báo người dùng tài khoản đó truy cập |
|  |  | Hủy cấm một tài khoản cá nhân | Xử lý hủy bỏ cấm truy cập đối với một tài khoản người dùng |

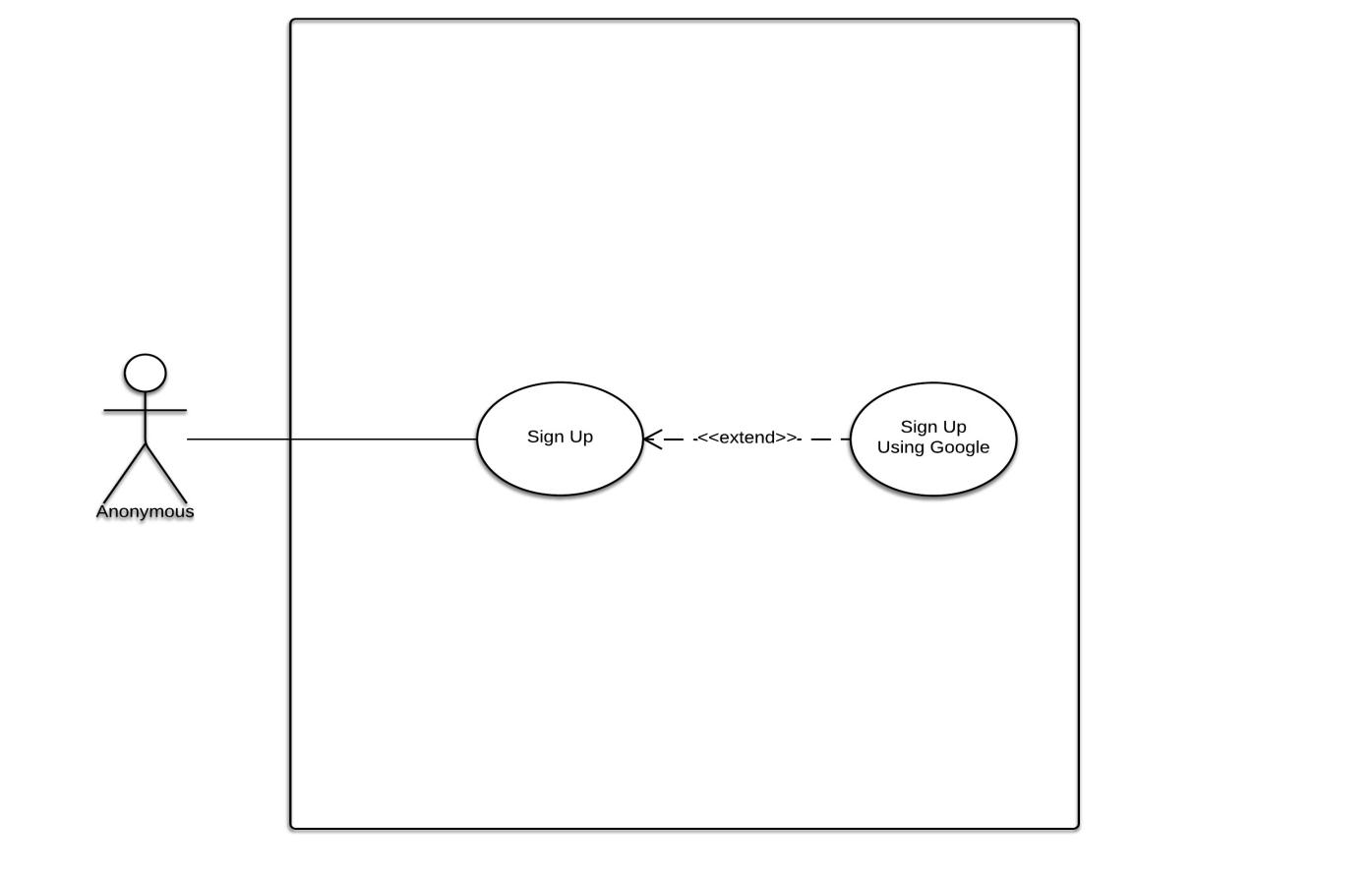
1. *Đặc tả mô hình Usecase*

**Usecase đăng nhập**

****

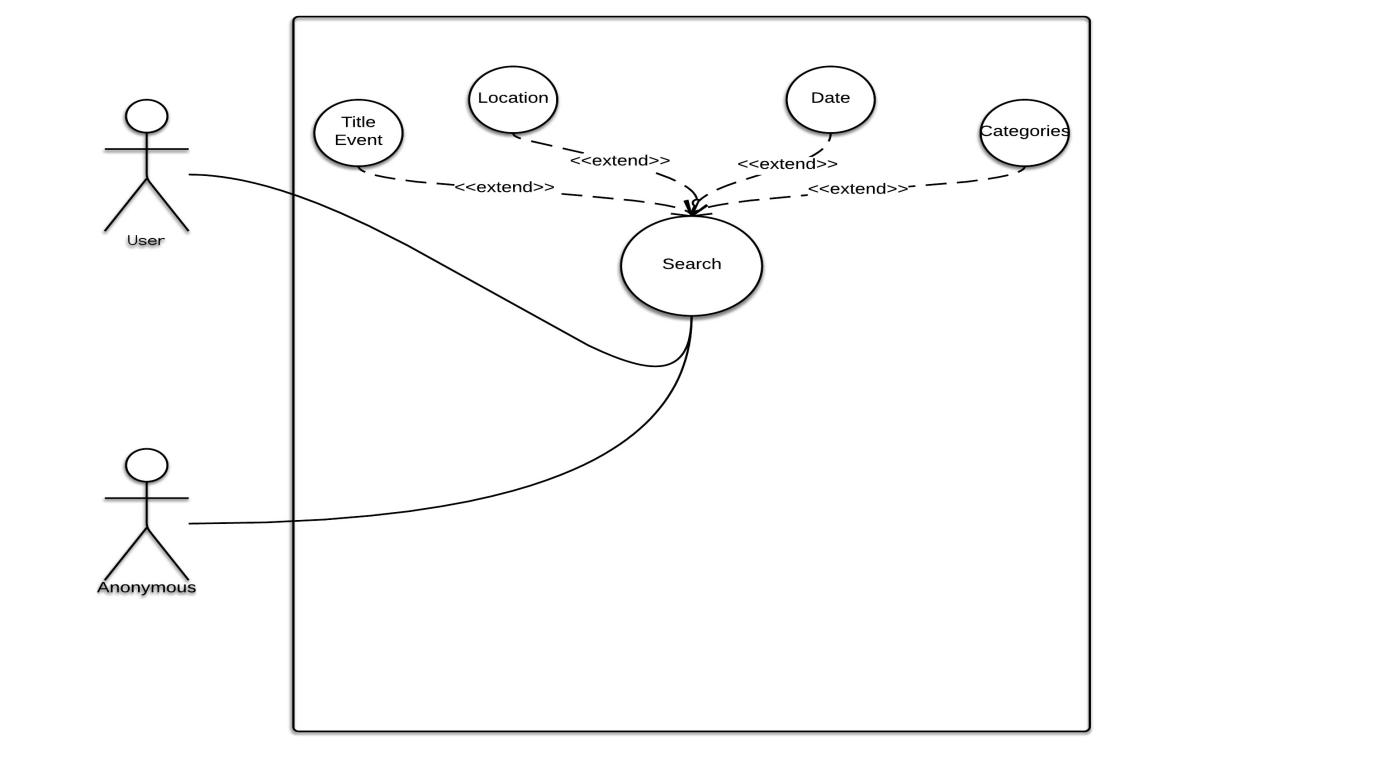
|  |  |
| --- | --- |
| **Tên Usecase** | **Đăng Nhập** |
| Tác Nhân | Người dùng đăng nhập vào Website |
| Tóm Tắt | Cho phép người dùng và quản trị đăng nhập vào hệ thống |
| Kết Quả | Người dùng sử dụng website theo quyền được cho phép |
| Kịch Bản Chính | Usercase được sử dụng khi người truy cập chọn đăng nhập   1. Người dùng bấm nút đăng nhập bằng tài khoản Website hoặc tài khoản Google 2. Người dùng nhập thông tin tài khoản Website hoặc tài khoản Google 3. Hệ thống kiểm tra tài khoản người dùng 4. Hệ thống báo đăng nhập thành công |
| Kịch Bản Phụ | Nếu người dùng nhập sai tài khoản mật khẩu hệ thống sẽ báo lỗi và yêu cầu người dùng nhập lại |

**Usecase đăng ký**

****

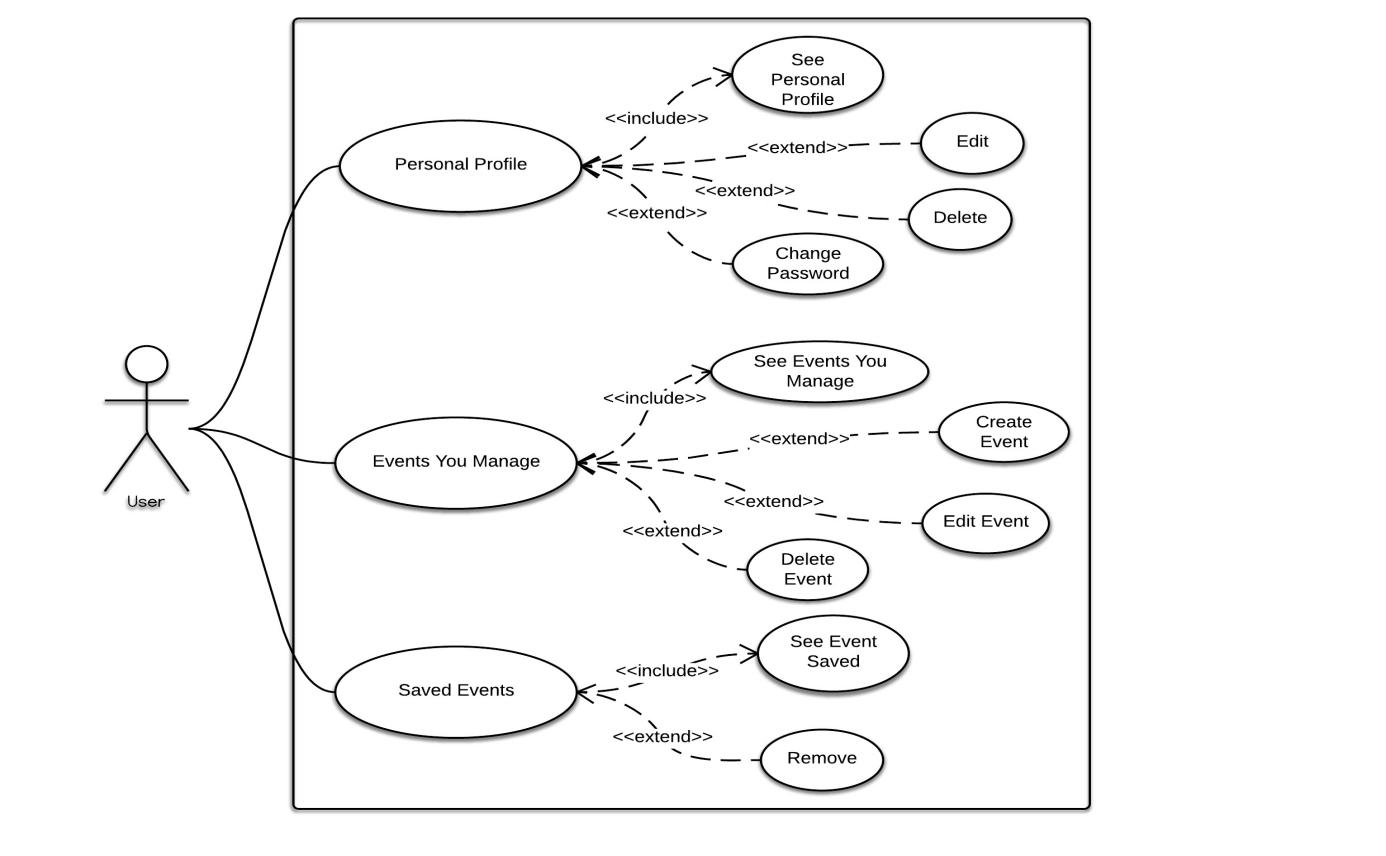
|  |  |
| --- | --- |
| **Tên Usecase** | **Đăng ký** |
| Tác Nhân | Người truy cập Website |
| Tóm Tắt | Cho phép người truy cập đăng ký tài khoản Website |
| Kết Quả | Người truy cập được tạo tài khoản Website |
| Kịch Bản Chính | Usercase được sử dụng khi người truy cập chọn đăng ký   1. Người dùng chọn đăng ký tài khoản bằng thông tin cá nhân hoặc bằng tài khoản Google 2. Người truy cập điền các thông tin cần thiết và bấm nút đăng ký 3. Hệ thống kiểm tra thông tin đăng ký tài khoản 4. Hệ thống báo đăng ký thành công |
| Kịch Bản Phụ | Nếu ngườ truy câp nhập sai hoặc chưa đầy đủ thông tin thì hệ thống sẽ báo lỗi và yêu cầu người dùng nhập lại thông tin |

**User case tìm kiếm**

****

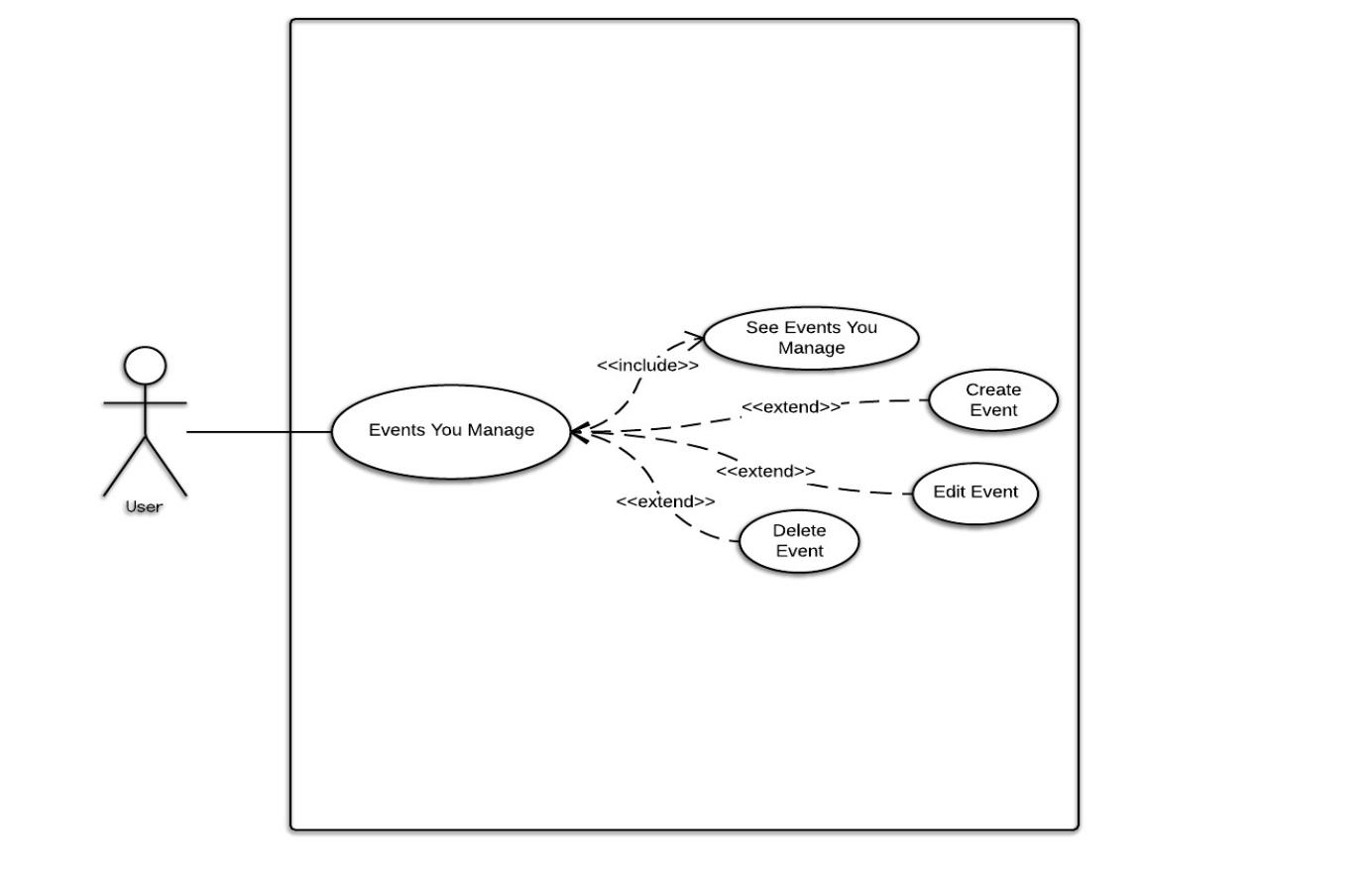
|  |  |
| --- | --- |
| **Tên Usecase** | **Tìm Kiếm** |
| Tác Nhân | Người dùng có hoặc chưa có tài khoản Website |
| Tóm Tắt | Cho phép người dùng và người vô danh tìm kiếm các sự kiện |
| Kết Quả | Tìm được sự kiện mong muốn |
| Kịch Bản Chính | Usercase được sử dụng khi người dùng chọn tìm kiếm   1. Người dùng chọn tìm kiếm 2. Ngươi dùng chọn các phương thức tìm kiếm theo:Thời gian, địa điểm,tên,khoa/phòng-ban/trung tâm/tất cả 3. Hệ thống kiểm tra thông tin tìm kiếm 4. Hiển thị các thông tin tìm kiếm theo yêu cầu ngươi dùng |
| Kịch Bản Phụ | Có thể các sự kiện người dùng yêu cầu không có nên Website sẽ không hiển thị bất cứ sự kiện gì |

**Usecase người dùng quản lý tài khoản cá nhân**

****

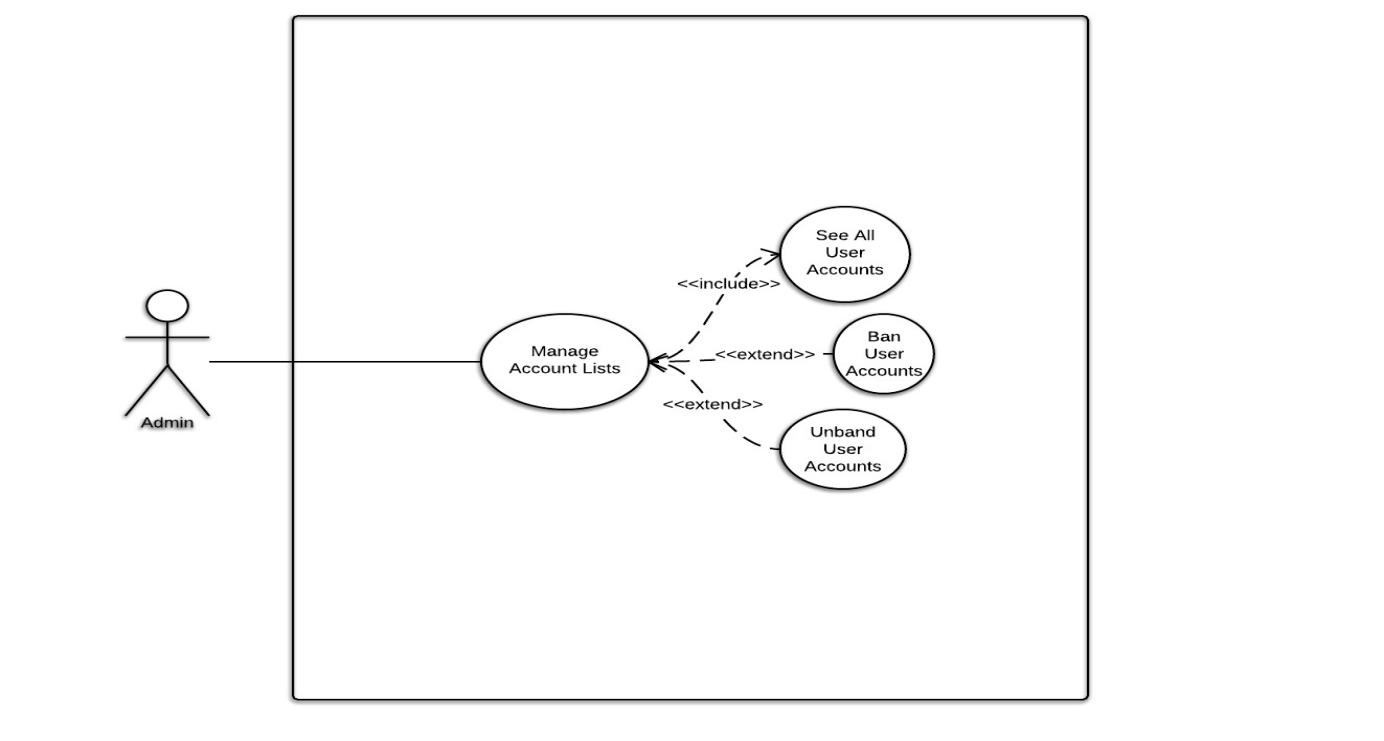
|  |  |
| --- | --- |
| **Tên Use case** | **Người dùng quản lý tài khoản cá nhân** |
| Tiền điều kiện | Người dùng phải đăng nhập hệ thống |
| Hậu điều kiện | Người dùng đăng nhập thành công |
| Đảm bảo thành công | Người dùng phải nhập tài khoản và mật khẩu khớp với tên đã đăng ký |
| Kích hoạt | Người dùng yêu cầu chức năng thông tin cá nhân |
| **Hành động tác nhân** | **Phản ứng hệ thống** |
| Hồ sơ cá nhân | -Hệ thống hiển thị thông tin cá nhân người dùng  -Người dùng thực hiện các yêu cầu chỉnh sửa thông tin cá nhân |
| Đổi mật khẩu | -Người dùng chọn đổi mật khẩu  -Yêu cầu người dùng nhập mật khẩu cũ và mới  -Xác nhận yêu cầu thay đổi mật khẩu  -Nếu mật khẩu mới không phù hợp thì hệ thống sẽ báo lỗi và yêu cầu người dùng nhập lại |
| Sự kiện đã tạo | -Hệ thống lưu thông tin sự kiện mà người dùng đã tạo vào trang cá nhân |
| Lưu sự kiện | -Lưu sự kiện vào trang cá nhân của người dùng khi bấm nút lưu sự kiện trên mỗi bản tin  -Hiển thị sự kiện đã lưu mỗi khi người dùng bấm vào các sự kiện đã lưu |
| Đăng xuất | -Người dùng muốn thoát khỏi Website  -Người dùng thực hiện bấm nút đăng xuất  -Hệ thống sẽ thông báo khảo sát yêu cầu có hoặc không để người dùng chọn  -Đăng xuất thành công |

**Use case sự kiện đã tạo**

****

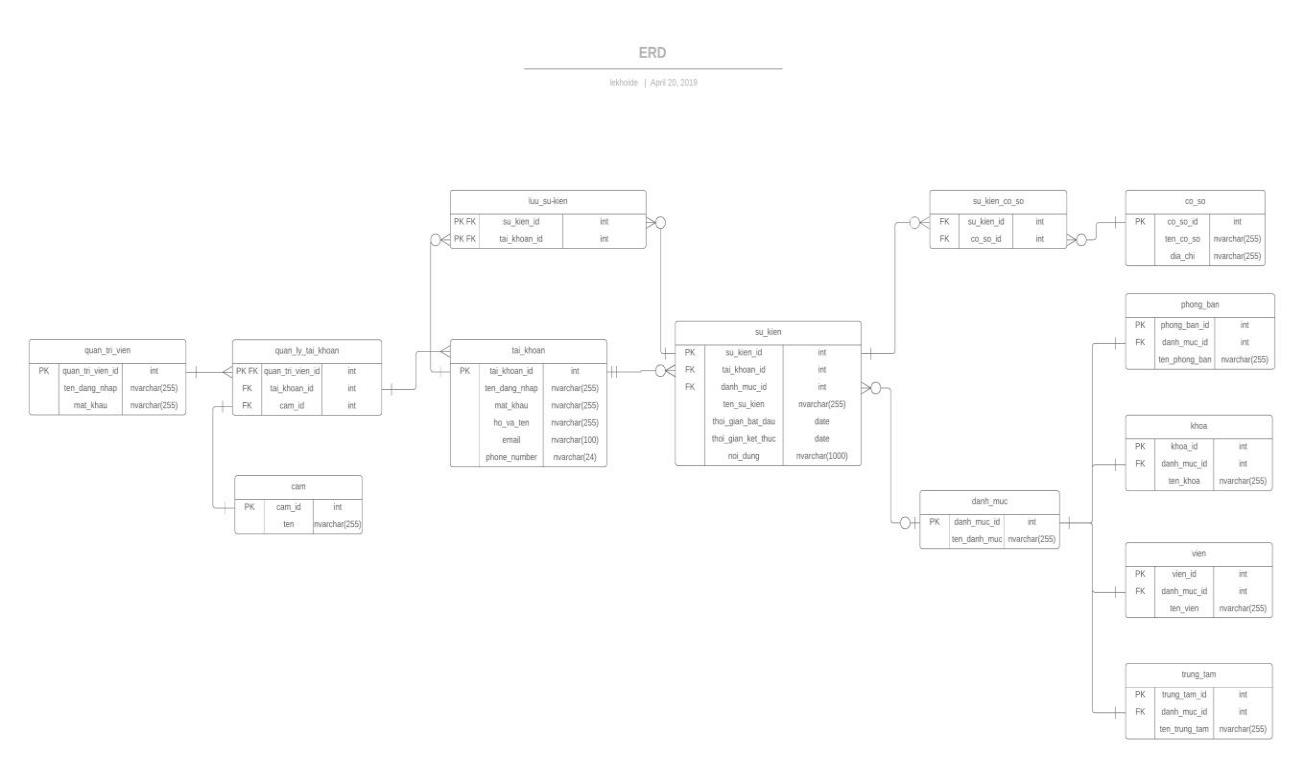
|  |  |
| --- | --- |
| **Tên Usecase** | **Sự kiện đã tạo** |
| Tiền điều kiện | Người dùng đăng nhập vào Website |
| Hậu điều kiện | Đăng nhập thành công |
| Đảm bảo thành công | Người dùng phải đăng nhập tài khoản và mật khẩu đúng với thông tin đăng ký |
| Kích hoạt | Thực hiện yêu cầu hiển thị trang cá nhân và bấm vào chức năng sự kiện đã tạo |
| Tạo sự kiện | -Người dùng thực hiện chức năng thiết lập một sự kiện khi bấm vào nút tạo sự kiện  -Hệ thống sẽ hiển thị các yêu cầu để người dùng tạo lập một sự kiện  -Tạo sự kiện thành công và hiển thị trên bảng tin website |
| Chỉnh sửa sự kiện | -Người dùng thực hiện yêu cầu chỉnh sửa sự kiện của cá nhân  -Hệ thống vẫn sẽ hiển thị các yêu cầu như chức năng tạo sự kiện  -Chỉnh sửa sự kiện thành công và hiển thị trên bảng tin website |
| Xóa sự kiện | -Truy cập danh sách sự kiện  -Thực hiện yêu cầu xóa sự kiện  -Khảo sát ý kiến có hoặc không về yêu cầu xóa sự kiện |

**Usecase quản trị quản lý danh sách tài khoản cá nhân người dùng**

****

|  |  |
| --- | --- |
| **Tên Usecase** | **Quản trị quản lý danh sách tài khoản cá nhân người dùng** |
| Tác nhân | Quản trị viên |
| Tóm tắt | Quản trị viên quản lý thông tin khách hàng |
| ĐK Tiên quyết | Quản trị viên phải đăng nhập vào hệ thống với quyền quản lý Website |
| Kết quả | Hệ thống lưu thông tin quản trị viên |
| Kịch bản chính | -Thực hiện quyền kiểm tra danh sách tài khoản Website  -Thực hiện nhiệm vụ cấm/mở quyền truy cập của người dùng |

1. *Mô hình*
2. Mô hình thực thể kết hợp ERD



**Danh sách cách thực thể**

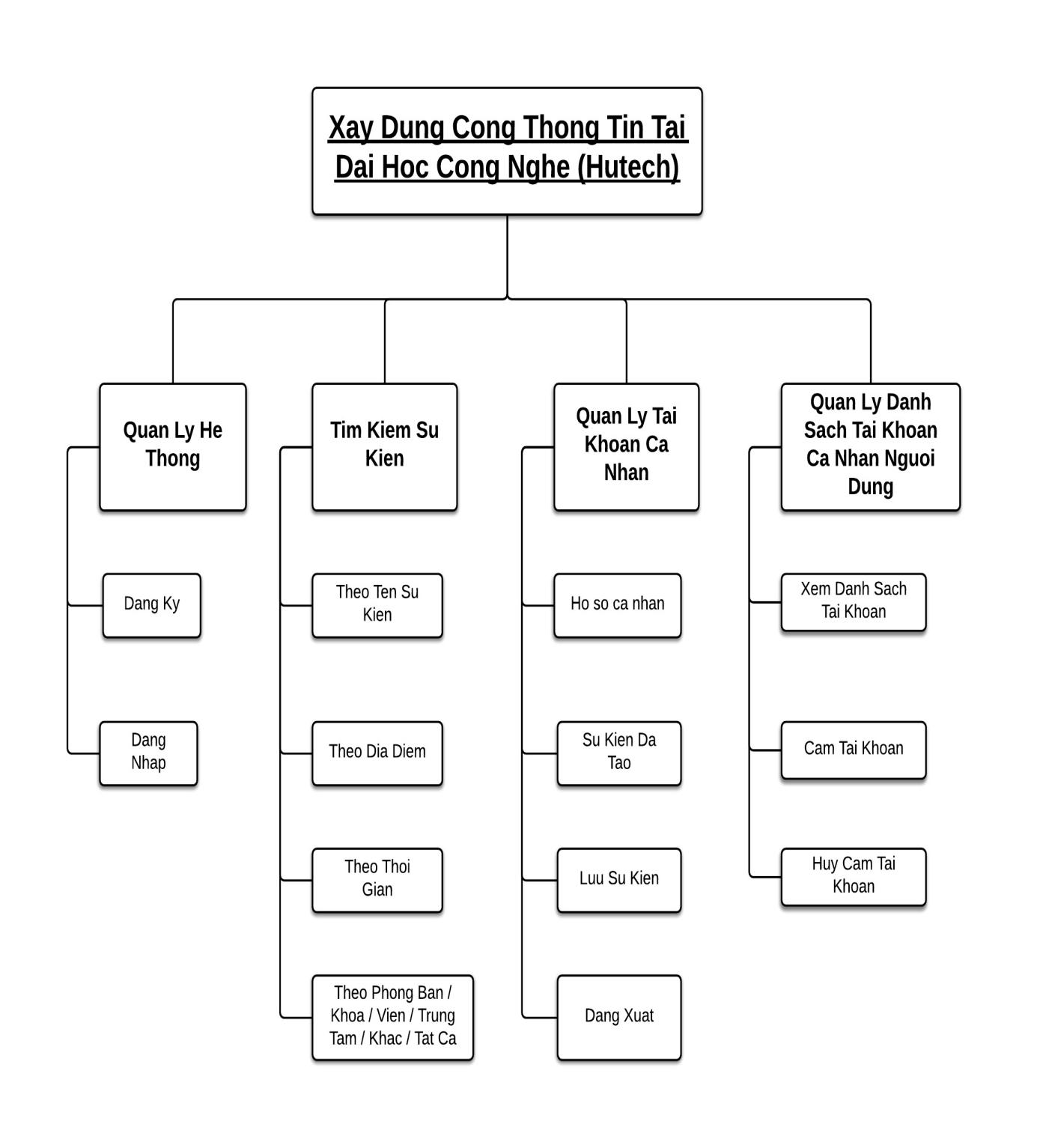
|  |  |
| --- | --- |
| **NAME** | **TABLE** |
| Tài Khoản | tai\_khoan |
| Sự Kiện | su\_kien |
| Sự Kiện Cơ Sở | su\_kien\_co\_so |
| Danh Mục | danh\_muc |
| Cơ Sở | co\_so |
| Phòng Ban | phòng\_ban |
| Khoa | khoa |
| Viện | vien |
| Trung Tâm | trung\_tam |
| Lưu Sự Kiện | lưu\_sư\_kien |
| Quản Lý Tài Khoản | quan\_ly\_tai\_khoan |
| Cấm | cam |
| Quản Trị Viên | quan\_tri\_vien |

1. *BFD - Business Function Diagram*

Công cụ biểu diễn việc phân rã có thứ bậc đơn giản các công việc cần thực hiện.

Mỗi công việc được chia ra làm các công việc con.

Độ phức tạp và kích cỡ hệ thống quyết định đến số mức chia ra.

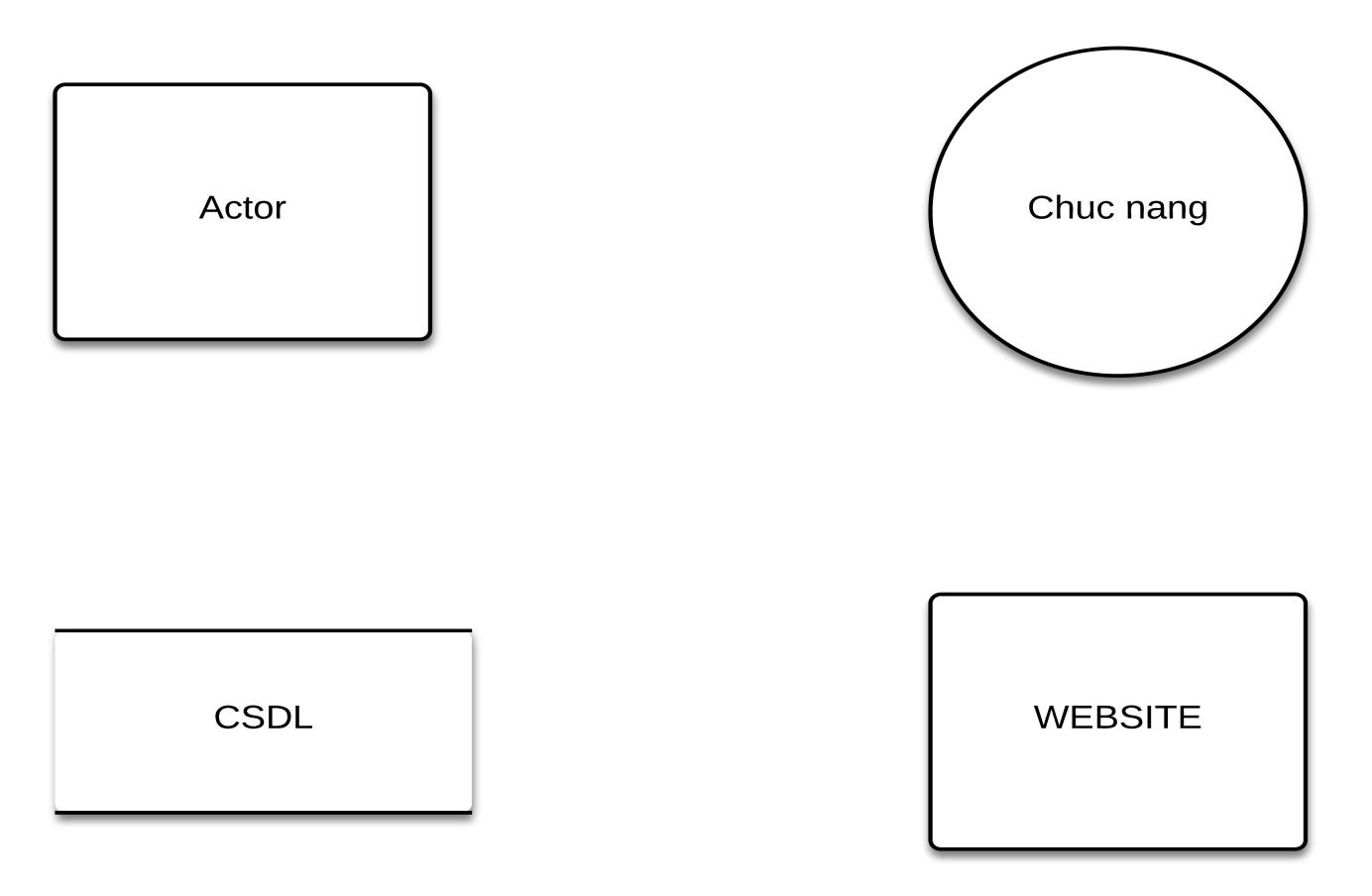


1. *DFD - Data Flow Diagram*

**Định nghĩa**

* Là một công cụ mô tả mối quan hệ thông tin giữa các công việc.
* Bổ sung các luồng thông tin nghiệp vụ, cho cái nhìn đầy đủ hơn về các mặt của hệ thống.
* Là một trong số các đầu vào cho quá trình thiết kế hệ thống

**Trình tự xây dựng mô hình luồng dữ liệu**

****

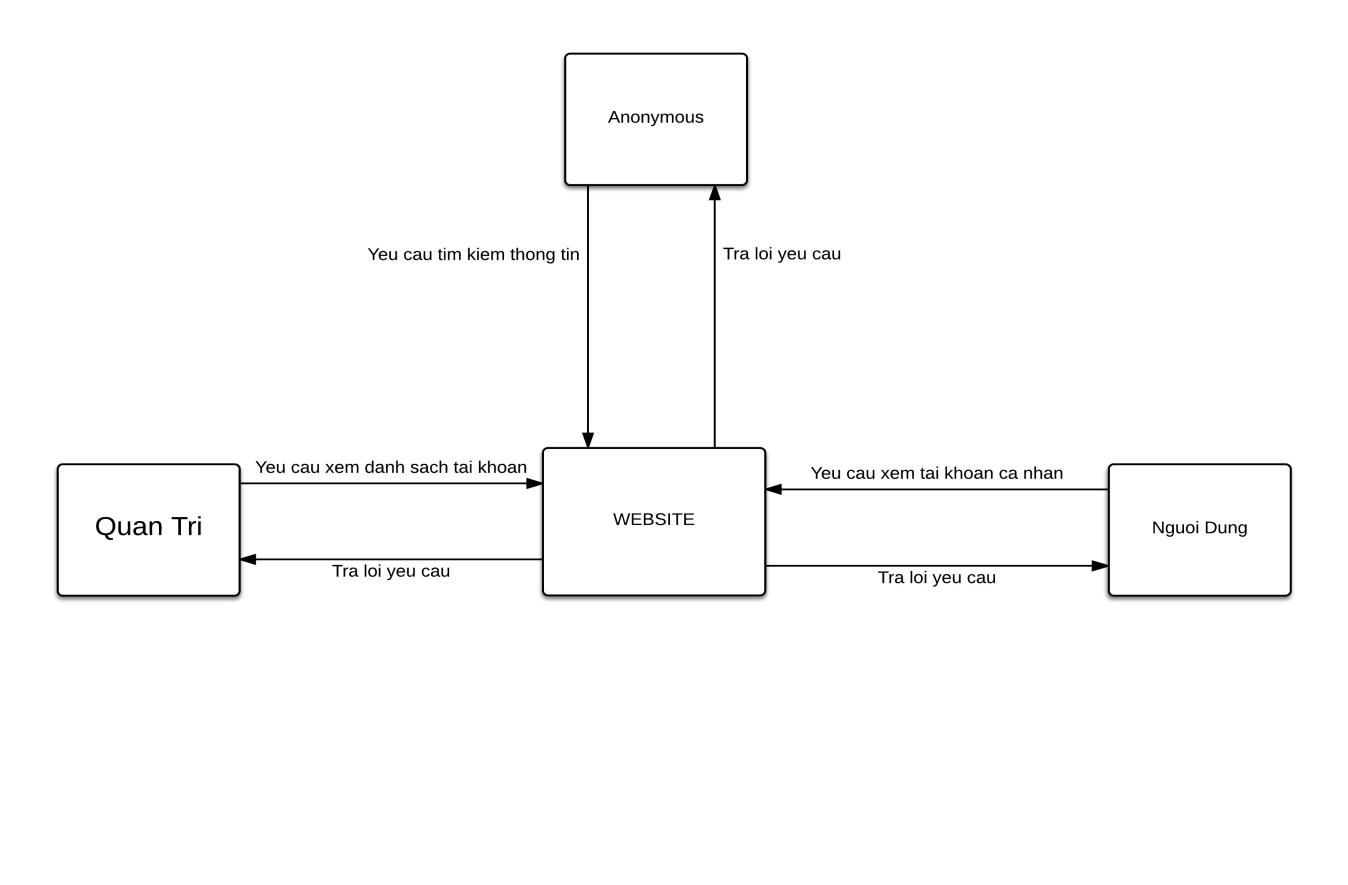
Actor: Tác nhân ngoài(Quản trị viên, Anonymuos, Người dùng)

Chức năng: Đại diện cho các chức năng trong hệ thống(Đăng ký, đăng nhập, tìm kiếm …)

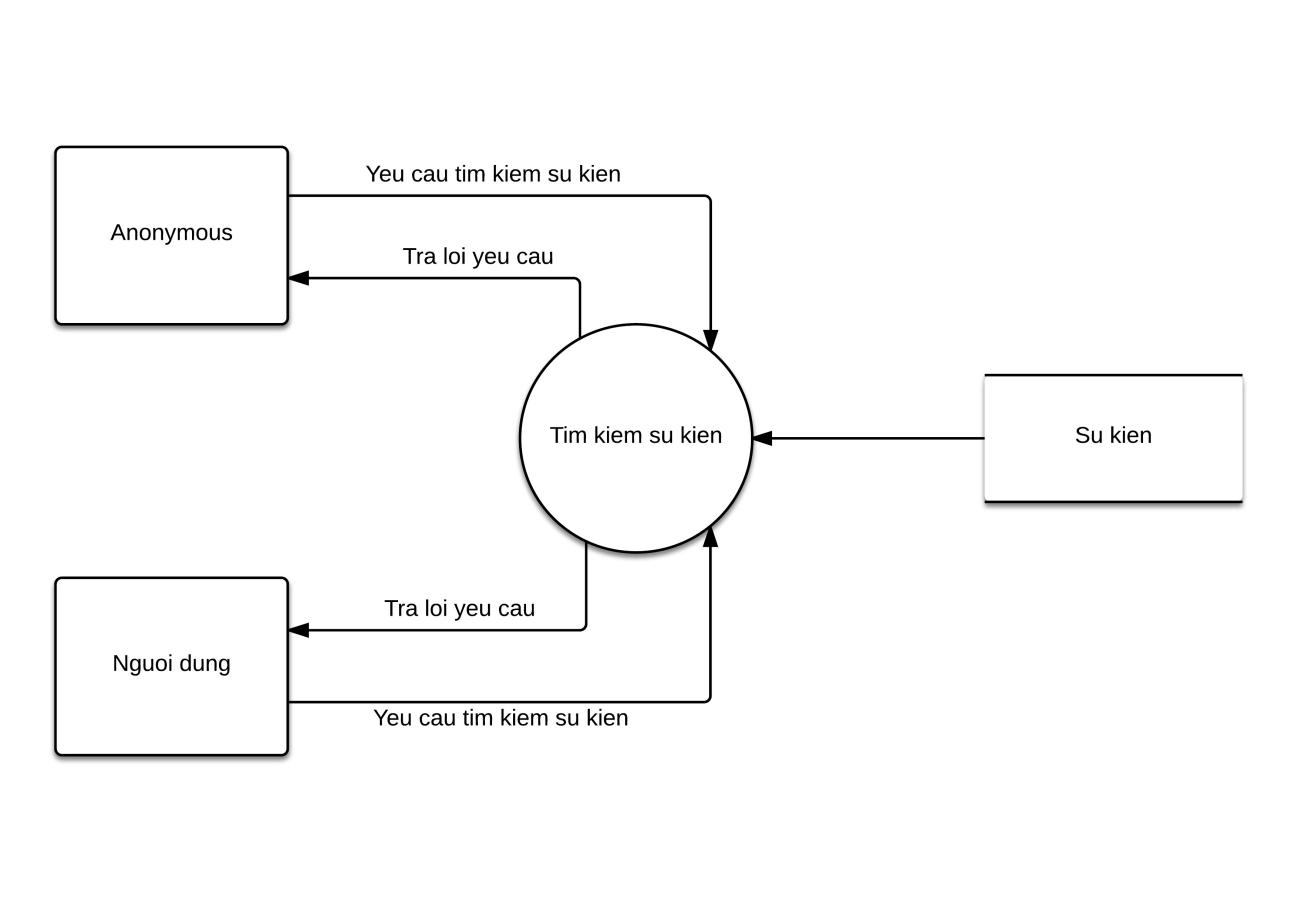
CSDL:Đại diện cho các thực thể trong CSDL

WEBSITE:Đại diện cho giao diện trang chủ

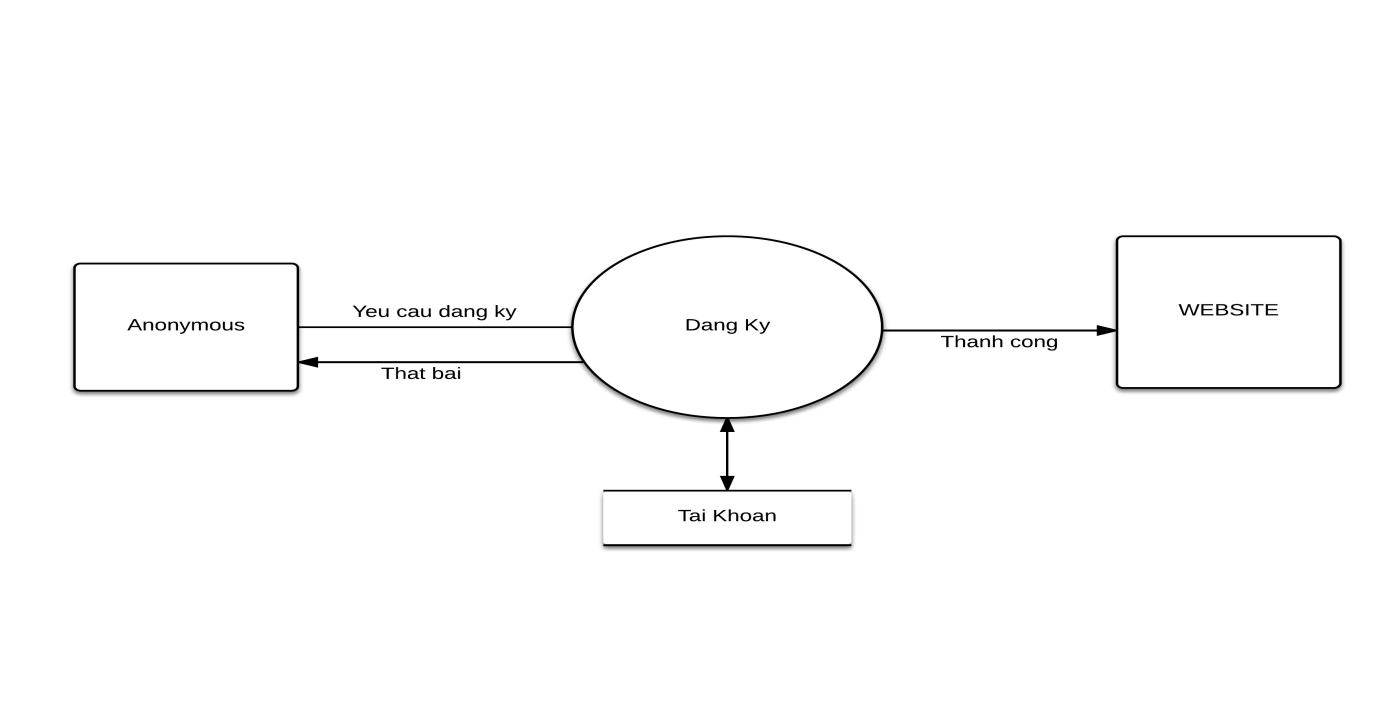
**Mức ngữ cảnh**

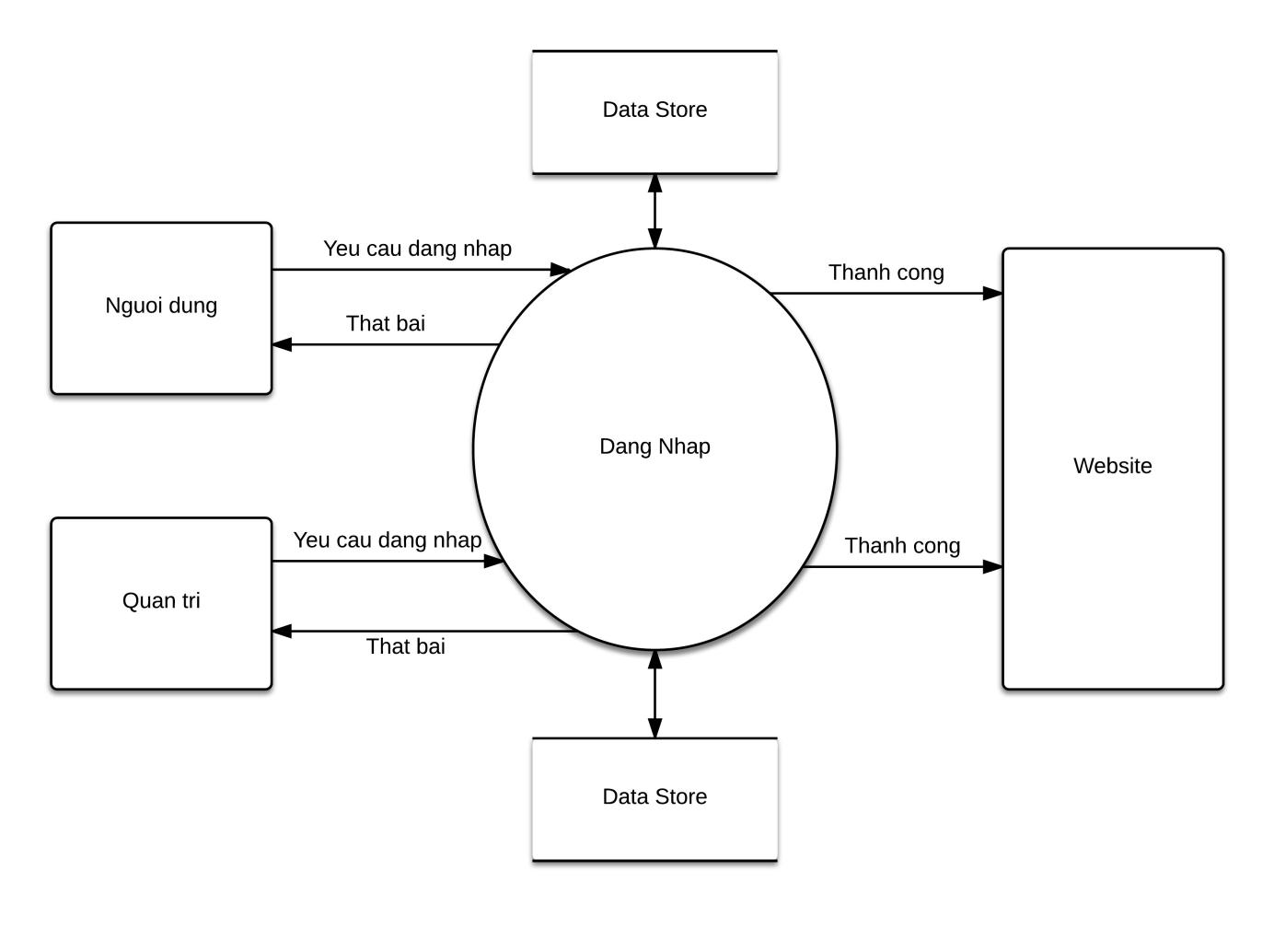


**Tìm kiếm sự kiện**

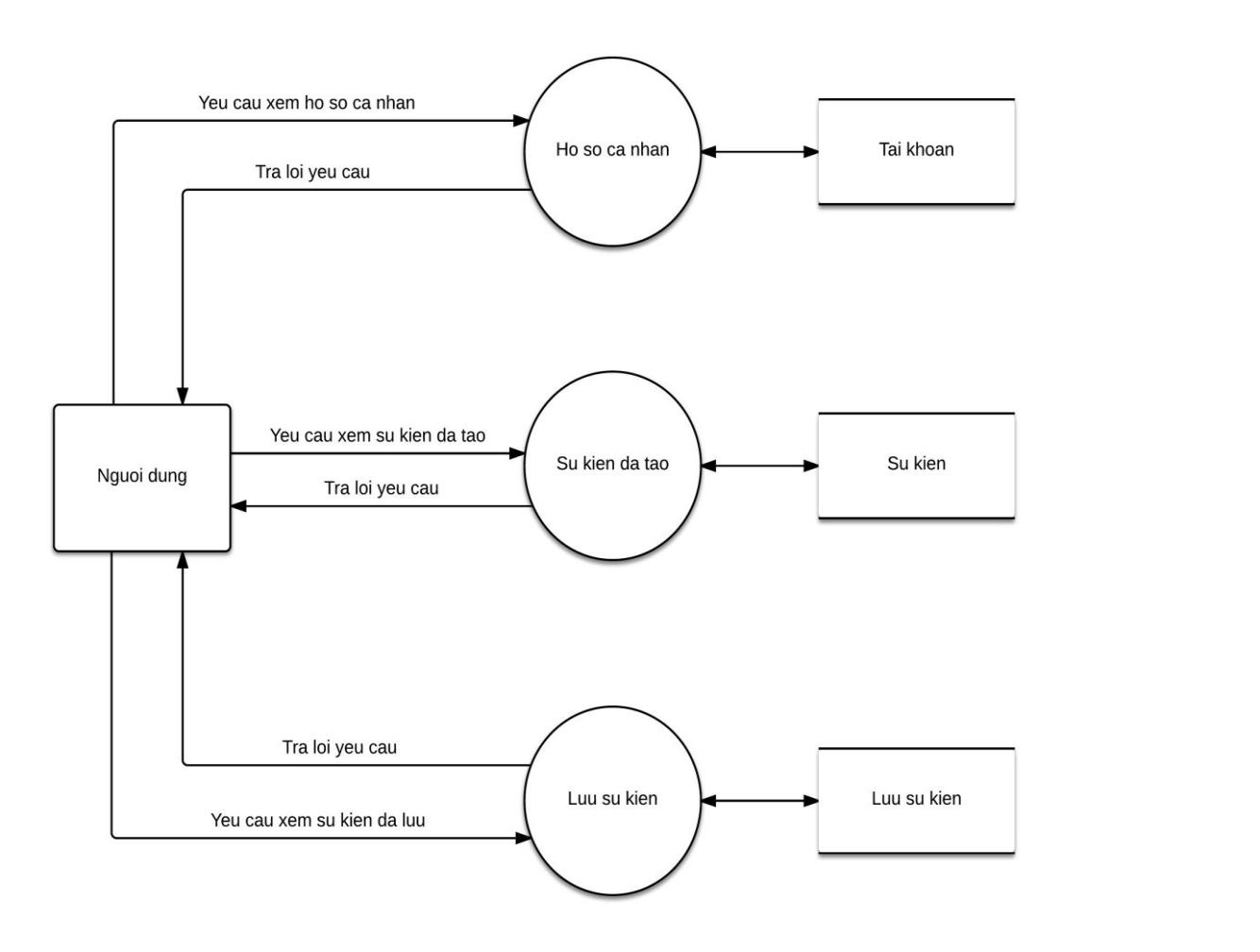


**Đăng ký-Đăng nhập**

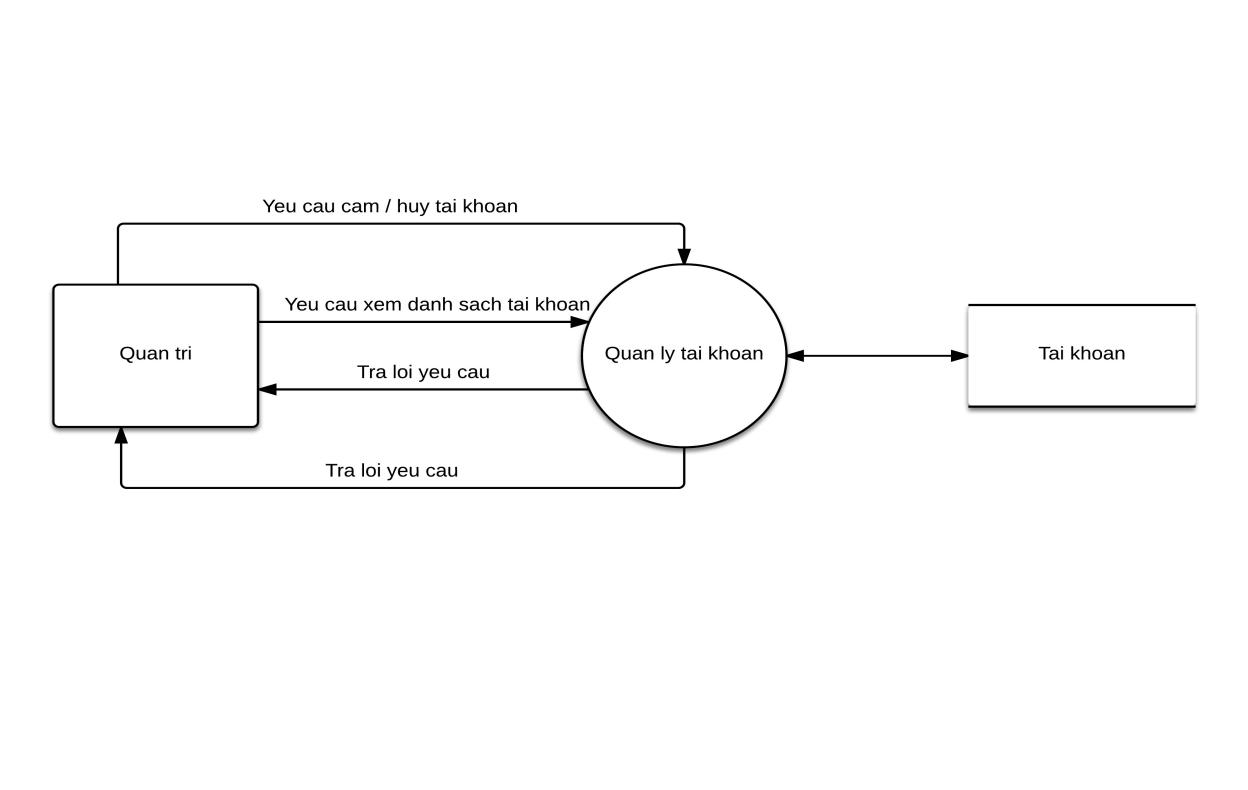




**Quản lý tài khoản cá nhân**



**Quản lý danh sách tài khoản cá nhân**



**CHƯƠNG 3. KẾT QUẢ THỰC NGHIỆM**

1. **Mô tả các công việc thực nghiệm mà đề tài đã tiến hành**
2. *Xác định (1/04/2019 - 07/04/2019)*
3. Nghiên cứu các Website tương tự
4. Xác định và phân tích các chức năng chính cần có
5. Xác định và phân tích cơ sở dữ liệu
6. Tổng kết và đánh giá sơ bộ
7. Lên kế hoạch chi tiết

- Project Manager lọc ra chức năng chính theo cấp độ

- Lập bảng chi tiết rủi ro và giải pháp đề xuất

- Lên bảng cấu trúc công việc

- Xác định số người tham gia dự án : 2 người

- Xác định thời gian bắt đầu và kết thúc : 1/04/2019 - 22/06/2019

1. *Phân tích và thiết kế (08/04/2019 - 21/04/2019)*
2. Yêu cầu (08/04 - 14/04)

- Đặc tả yêu cầu chức năng

- Đặc tả yêu cầu giao diện

- Đặc tả yêu cầu hệ thống

1. Dữ liệu (15/04 - 21/04)

- UseCase

- Mô hình UseCase

- Danh sách Actor

- Danh sách UseCase

- Đặc tả chi tiết UseCase

- ERD

- Mô hình thực thể kết hợp ERD

- Danh sách các thực thể

1. BFD

Mô hình phân rã chức năng - BFD

1. DFD

Mô hình luồng dữ liệu - DFD

1. Giao diện (15/04 - 21/04)

- Thiết kế giao diện Demo

- Thiết kế giao diện Đăng ký

- Thiết kế giao diện Đăng nhập

- Thiết kế giao diện Trang chủ

- Thiết kế giao diện Người dùng quản lý tài khoản cá nhân

- Thiết kế giao diện Quản trị quản lý danh sách tài khoản cá nhân người dùng

1. Nhận yêu cầu của giảng viên và cập nhật (Update)
2. *Thực hiện (22/04/2019 - 2/06/2019)*
3. Cơ sở dữ liệu (22/04 - 28/04)

- Hoàn chỉnh toàn bộ cơ sở dữ liệu

- Nhận yêu cầu giảng viên và cập nhật (Update)

1. Chức năng (29/04 - 2/06)

- Đăng ký bằng Google

- Đăng nhập bằng Google

- Ghi nhớ đăng nhập

- Quên mật khẩu

1. Chức năng người dùng quản lý tài khoản cá nhân (6/05 - 26/05)

- Xem thông tin cá nhân

- Xóa thông tin cá nhân

- Sửa thông tin cá nhân

- Đổi mật khẩu

- Tạo sự kiện

- Chỉnh sửa sự kiện

- Xóa sự kiện

- Lưu sự kiện

- Đăng xuất

1. Chức năng tìm kiếm sự kiện (27/05 - 2/06)

- Tìm kiếm sự kiện theo tên sự kiện

- Tìm kiếm sự kiện theo địa điểm

- Tìm kiếm sự kiện theo ngày

- Tìm kiếm sự kiện theo phòng-ban / khoa/ viện/ trung tâm/ khác / tất cả

1. Quản trị quản lý danh sách tài khoản cá nhân người dùng (27/05 - 2/06)

- Xem danh sách tài khoản cá nhân

- Cấm một tài khoản cá nhân truy cập

- Hủy cấm một tài khoản cá nhân truy cập

- Hoàn chỉnh toàn bộ chức năng

- Nhận yêu cầu của giảng viên và cập nhật (Update)

1. *Kiểm thử (3/06/2019 - 16/06/2019)*
2. Kiểm thử hệ thống (3/06 - 9/06)

- Kiểm tra các chức năng trong thiết kế

- Hiệu chỉnh và sửa lỗi

1. Kiểm thử chấp nhận (10/06 - 16/06)

- Demo cho giảng viên các chức năng của hệ thống

- Hiệu chỉnh và sửa lỗi theo yêu cầu của giảng viên

1. **Kết quả nghiên cứu lý thuyết - thực nghiệm đạt được**
2. *Kết quả nghiên cứu lý thuyết :*

* Bị trục trặc vào việc thiết kế trên Cơ sở dữ liệu
* Đề tài bị thay đổi hơi nhiều khi vào giai đoạn thực hiện không giống giai đoạn ban đầu khi phân tích
* Áp dụng thành công công nghệ Code First trên ASP.NET.MVC

1. *Kết quả thực nghiệm đạt được*

* Thiết kế được cơ sở dữ liệu có 3 bảng : Events, Department, Campus
* Sử dụng công nghe Entity Frameword 6 Code First để xây dựng nên nơi lưu trữ dữ liệu và xây dựng trang Website.
* Code và Test thành công việc xem, xóa, sửa thêm sự kiện có ràng buộc .

1. **Kết quả sản phẩm phần mềm**
2. *Hồ sơ thiết kế và cài đặt*
3. Thư mục Models

Bảng Model Event (Sự Kiện)

using BigEvent.ViewModels;using System;using System.Collections.Generic;using System.ComponentModel.DataAnnotations;using System.ComponentModel.DataAnnotations.Schema;using System.Linq;using System.Web;

namespace BigEvent.Models

{

    [Table("Event")]

    public class Event

    {

        public int EventID { get; set; }

        [Display(Name = "Tên sự kiện")]

        [StringLength(255, ErrorMessage = "Tối đa 255 ký tự")]

        [MinLength(6, ErrorMessage = "Tối thiểu có 6 ký tự")]

        [Required(AllowEmptyStrings = false, ErrorMessage = "Bắt buộc nhập")]

        public string Title { get; set; }

        [Display(Name = "Cơ sở")]

        public int CampusID { get; set; }

        [Display(Name = "Phòng")]

        public string Room { get; set; }

        [Display(Name = "Phòng-ban / Khoa / Viện / Trung tâm")]

        public int DepartmentID { get; set; }

        [Display(Name = "Ngày sự kiện")]

        [DataType(DataType.Date)]

        [DisplayFormat(DataFormatString = "{0:yyyy-MM-dd}", ApplyFormatInEditMode = true)]

        //[FutureDate(ErrorMessage = "Mời bạn nhập lại")]

        public DateTime StartDate { get; set; }

        [Display(Name = "Nội dung")]

        public string Content { get; set; }

        [Display(Name = "Ảnh sự kiện")]

        public string ImagePath { get; set; }

        [NotMapped]

        public HttpPostedFileBase ImageFile { get; set; }

        //public string AdministratorID { get; set; }

        //public virtual ApplicationUser Administrator { get; set; }

        public virtual Campus Campus { get; set; }

        public virtual ICollection<Department> Departments { get; set; }

    }

}

Bảng Model Campus (Cơ Sở)

using System;using System.Collections.Generic;using System.ComponentModel.DataAnnotations;using System.ComponentModel.DataAnnotations.Schema;using System.Linq;using System.Web;

namespace BigEvent.Models

{

    [Table("Campus")]

    public class Campus

    {

        [DatabaseGenerated(DatabaseGeneratedOption.None)]

        public int CampusID { get; set; }

        [Display(Name = "Tên cơ sở / Tên khu")]

        [StringLength(10, ErrorMessage = "Tối đa 10 ký tự")]

        [MinLength(1, ErrorMessage = "Tối thiểu có 1 ký tự")]

        //[RegularExpression(@"^[A-Z]+[a-zA-Z'\s]\*$")]

        public string CampusName { get; set; }

        public virtual ICollection<Event> Events { get; set; }

    }

}

Bảng Model Department (Phòng Ban - Khoa - Vien - Trung Tâm)

using System;using System.Collections.Generic;using System.ComponentModel.DataAnnotations;using System.ComponentModel.DataAnnotations.Schema;using System.Linq;using System.Web;

namespace BigEvent.Models

{

    [Table("Department")]

    public class Department

    {

        [DatabaseGenerated(DatabaseGeneratedOption.None)]

        public int DepartmentID { get; set; }

        [Display(Name = "Phòng-ban / Khoa / Viện / Trung tâm")]

        [StringLength(255, ErrorMessage = "Tối đa 255 ký tự")]

        [MinLength(3, ErrorMessage = "Tối thiểu có 3 ký tự")]

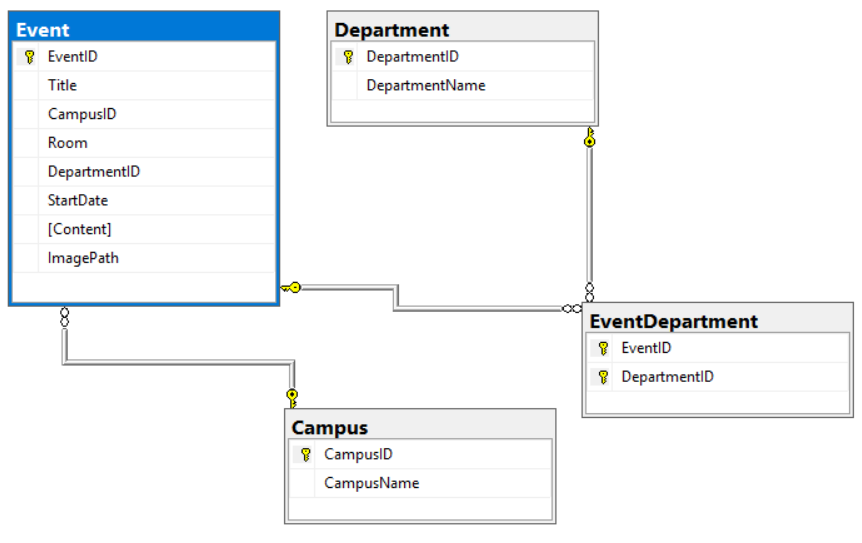
        public string DepartmentName { get; set; }

        public virtual ICollection<Event> Events { get; set; }

    }

}

Bảng Database Diagrams



1. Thư mục Controllers

Admin lấy tất cả các sự kiện

//Trang admin: lấy tất cả các sự kiện

       public ActionResult Index()

       {

           var viewModel = new EventIndexData();

           viewModel.Events = db.Event

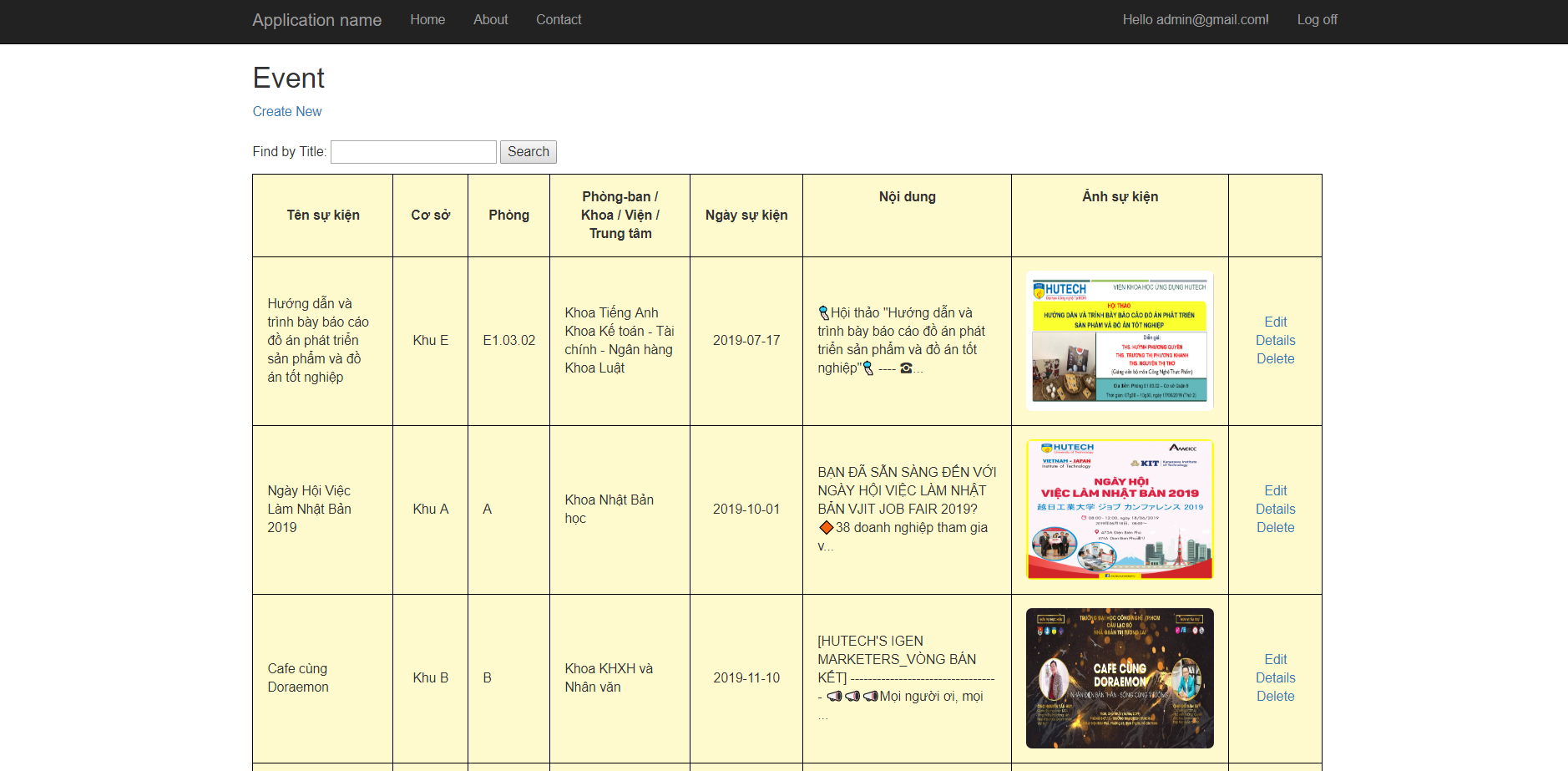
               .Include(i => i.Campus)

               .Include(i => i.Departments)

               .ToList();

           return View(viewModel);

       }



Admin tạo một sự kiện

//Trang admin: Tạo một sự kiện

       public ActionResult Create()

       {

           var @event = new Event();

           @event.Departments = new List<Department>();

           PopulateCampusDropDownList();

           PopulateAssignedDepartmentData(@event);

           return View();

       }

       [HttpPost]

       [ValidateAntiForgeryToken]

       public ActionResult Create([Bind(Include = "EventID, Title, CampusID, Room, DepartmentID, StartDate, Content, ImagePath, ImageFile")]Event @event, string[] selectedDepartments)

       {

           //THÊM ẢNH

           string FileName = Path.GetFileNameWithoutExtension(@event.ImageFile.FileName);

           string Extension = Path.GetExtension(@event.ImageFile.FileName);

           FileName = FileName + DateTime.Now.ToString("yymmssfff") + Extension;

           @event.ImagePath = "~/Content/Images/" + FileName;

           FileName = Path.Combine(Server.MapPath("~/Content/Images/"), FileName);

           if (selectedDepartments != null)

           {

               @event.Departments = new List<Department>();

               foreach (var department in selectedDepartments)

               {

                   var deparmentToAdd = db.Department.Find(int.Parse(department));

                   @event.Departments.Add(deparmentToAdd);

               }

           }

           //Ràng buộc khi thêm ảnh

           if (Extension.ToLower() == ".jpg" || Extension.ToLower() == ".jpeg" || Extension.ToLower() == ".png")

           {

               if (@event.ImageFile.ContentLength < 100000000)

               {

                   @event.ImageFile.SaveAs(FileName);

                   db.Event.Add(@event);

                   db.SaveChanges();

                   TempData["TaoThanhCong"] = "Sự kiện tạo thành công";

                   return RedirectToAction("Index");

               }

               else

               {

                   ViewBag.DungLuong = "Chỉ chấp nhận dung lượng ảnh dưới 100MB";

               }

           }

           else

           {

               ViewBag.DinhDangAnh = "Chỉ chấp nhận .jpg, .jpeg, .png";

           }

           PopulateCampusDropDownList(@event.CampusID);

           PopulateAssignedDepartmentData(@event);

           return View(@event);

       }

Admin chỉnh sửa một sự kiện

//Trang admin : Chỉnh sửa một sự kiện

        public ActionResult Edit(int? id)

        {

            if (id == null)

            {

                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

            }

            Event @event = db.Event

                .Include(i => i.Campus)

                .Include(i => i.Departments)

                .Where(i => i.EventID == id)

                .Single();

            Session["ImagePath"] = @event.ImagePath;

            PopulateCampusDropDownList(@event.CampusID);

            PopulateAssignedDepartmentData(@event);

            if (@event == null)

            {

                return HttpNotFound();

            }

            return View(@event);

        }

        [HttpPost]

        [ValidateAntiForgeryToken]

        public ActionResult Edit(int? id, string[] selectedDepartments)

        {

            if (id == null)

            {

                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

            }

            Event eventToUpdate = db.Event

                .Include(i => i.Campus)

                .Include(i => i.Departments)

                .Where(i => i.EventID == id)

                .SingleOrDefault();

            if (TryUpdateModel(eventToUpdate, "",

                        new string[] { "Title", "CampusID", "Room", "DepartmentID", "StartDate", "Content", "ImagePath", "ImageFile" }))

            {

                try

                {

                    if (eventToUpdate.ImageFile != null)

                    {

                        //Thêm ảnh

                        string FileName = Path.GetFileNameWithoutExtension(eventToUpdate.ImageFile.FileName);

                        string Extension = Path.GetExtension(eventToUpdate.ImageFile.FileName);

                        FileName = FileName + DateTime.Now.ToString("yymmssfff") + Extension;

                        eventToUpdate.ImagePath = "~/Content/Images/" + FileName;

                        FileName = Path.Combine(Server.MapPath("~/Content/Images/"), FileName);

                        //Ràng buộc khi thêm ảnh

                        if (Extension.ToLower() == ".jpg" || Extension.ToLower() == ".jpeg" || Extension.ToLower() == ".png")

                        {

                            if (eventToUpdate.ImageFile.ContentLength < 100000000)

                            {

                                //Điều kiện Campus và cập nhật bảng EventDepartment

                                if (String.IsNullOrWhiteSpace(eventToUpdate.Campus.CampusName))

                                {

                                    eventToUpdate.Campus = null;

                                }

                                UpdateEventDepartments(selectedDepartments, eventToUpdate);

                                //Nếu thêm ảnh thì sẽ xóa ảnh cũ trên sự kiện đó

                                eventToUpdate.ImageFile.SaveAs(FileName);

                                db.Entry(eventToUpdate).State = EntityState.Modified;

                                string OldImagePath = Request.MapPath(Session["ImagePath"].ToString());

                                if(System.IO.File.Exists(OldImagePath))

                                {

                                    System.IO.File.Delete(OldImagePath);

                                }

                                db.SaveChanges();

                                TempData["CapNhatThanhCong"] = "Cập nhật sự kiện thành công";

                                return RedirectToAction("Index");

                            }

                            else

                            {

                                ViewBag.DungLuong = "Chỉ chấp nhận dung lượng ảnh dưới 100MB";

                            }

                        }

                        else

                        {

                            ViewBag.DinhDangAnh = "Chỉ chấp nhận .jpg, .jpeg, .png";

                        }

                    }

                    else

                    {

                        //Điều kiện Campus và cập nhật bảng EventDepartment

                        if (String.IsNullOrWhiteSpace(eventToUpdate.Campus.CampusName))

                        {

                            eventToUpdate.Campus = null;

                        }

                        UpdateEventDepartments(selectedDepartments, eventToUpdate);

                        //Nếu thêm ảnh thì sẽ thay thế ảnh cũ trên sự kiện đó

                        eventToUpdate.ImagePath = Session["ImagePath"].ToString();

                        db.Entry(eventToUpdate).State = EntityState.Modified;

                        db.SaveChanges();

                        TempData["CapNhatThanhCong"] = "Sự kiện cập nhật thành công";

                        return RedirectToAction("Index");

                    }

                }

                catch (RetryLimitExceededException /\* dex \*/)

                {

                    //Log the error(uncomment dex variable name and add a line here to write a log.

                    ModelState.AddModelError("", "Unable to save changes. Try again, and if the problem persists, see your system administrator.");

                }

            }

            PopulateAssignedDepartmentData(eventToUpdate);

            PopulateCampusDropDownList(eventToUpdate.CampusID);

            return View(eventToUpdate);

        }

Admin xóa một sự kiện

//Trang admin: Xóa một sự kiện

        public ActionResult Delete(int? id)

        {

            if (id == null)

            {

                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

            }

            Event @event = db.Event.Find(id);

            if (@event == null)

            {

                return HttpNotFound();

            }

            return View(@event);

        }

        [HttpPost, ActionName("Delete")]

        [ValidateAntiForgeryToken]

        public ActionResult DeleteConfirmed(int id)

        {

            Event @event = db.Event

                .Include(i => i.Campus)

                .Include(i => i.Departments)

                .Where(i => i.EventID == id)

                .Single();

            //Nếu xóa thì sẽ xóa luôn File ảnh đã được nạp './Content/Images'

            string CurrentImage = Request.MapPath(@event.ImagePath);

            db.Entry(@event).State = EntityState.Deleted;

            if(System.IO.File.Exists(CurrentImage))

            {

                System.IO.File.Delete(CurrentImage);

            }

            db.Event.Remove(@event);

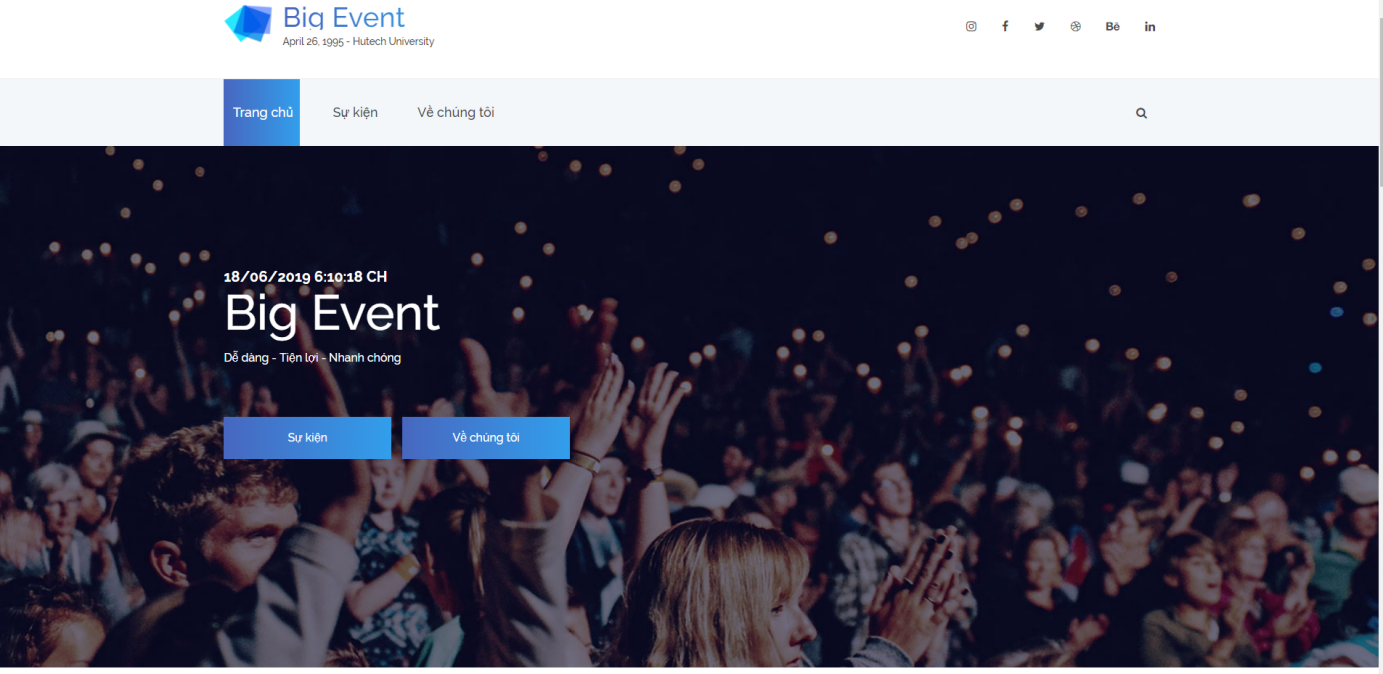
            db.SaveChanges();

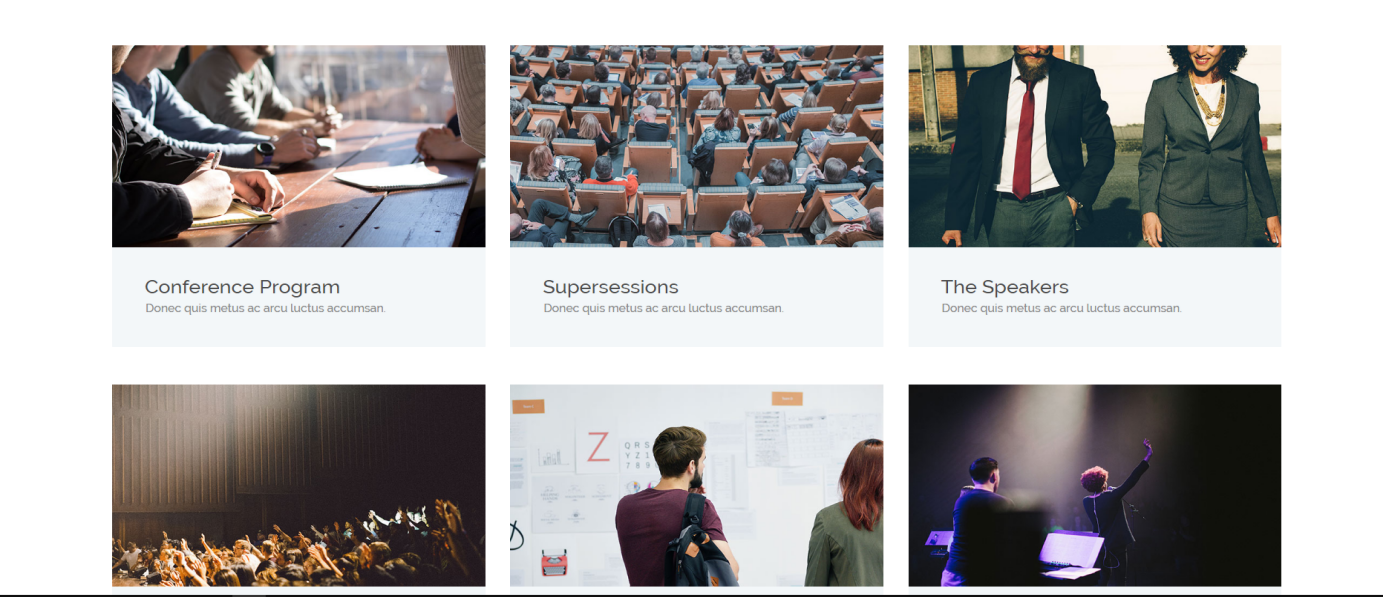
            TempData["XoaThanhCong"] = "Xóa sự kiện thành công";

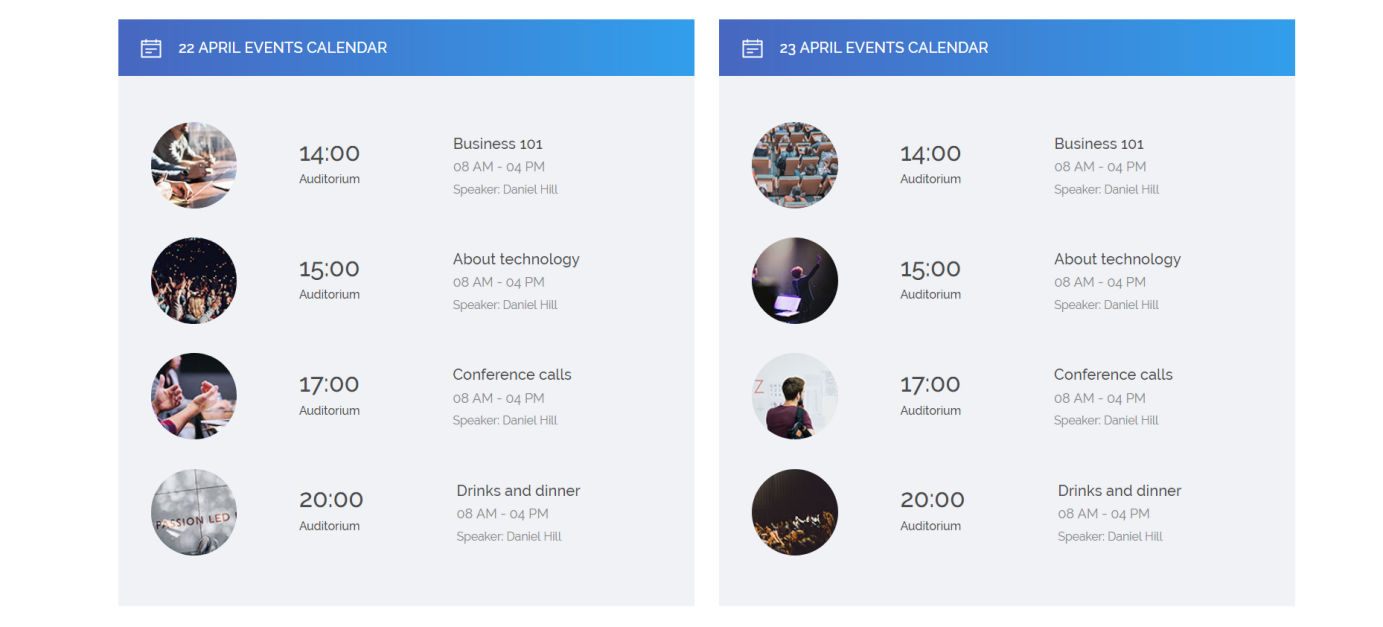
            return RedirectToAction("Index");

        }

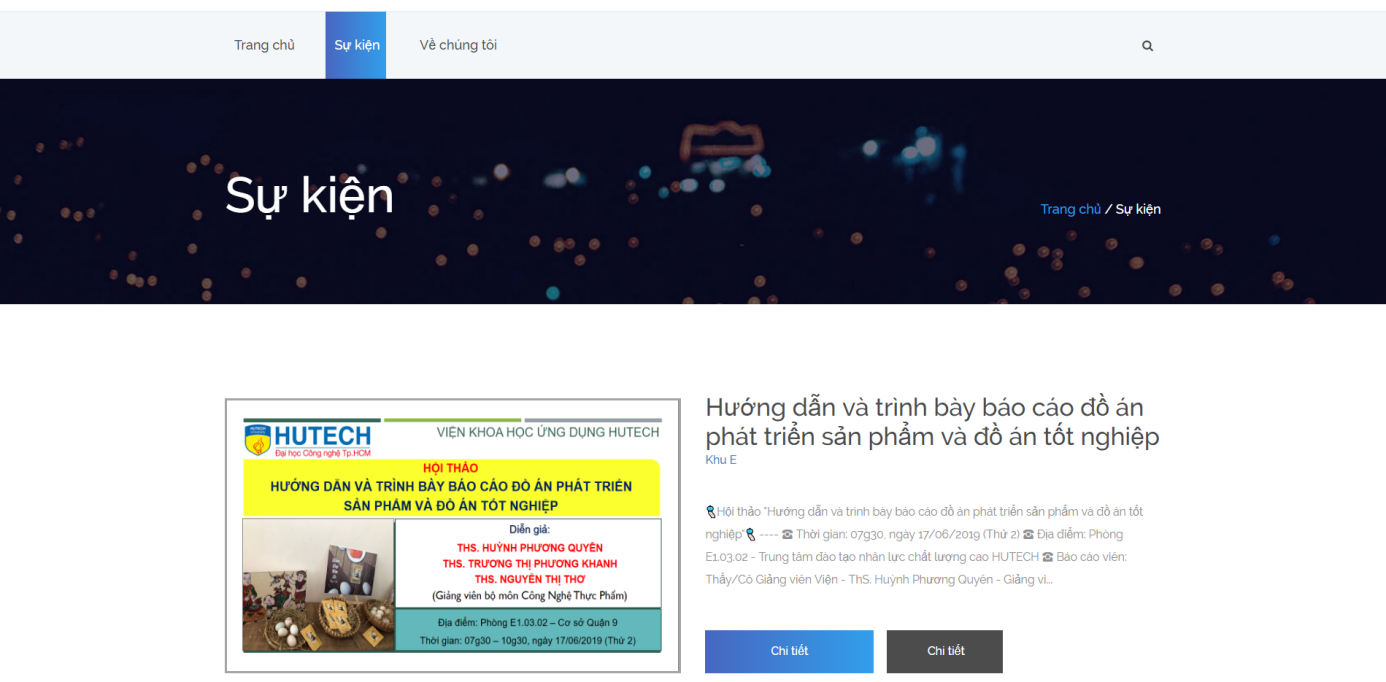
1. *Giao diện*
2. Giao diện trang chủ người xem

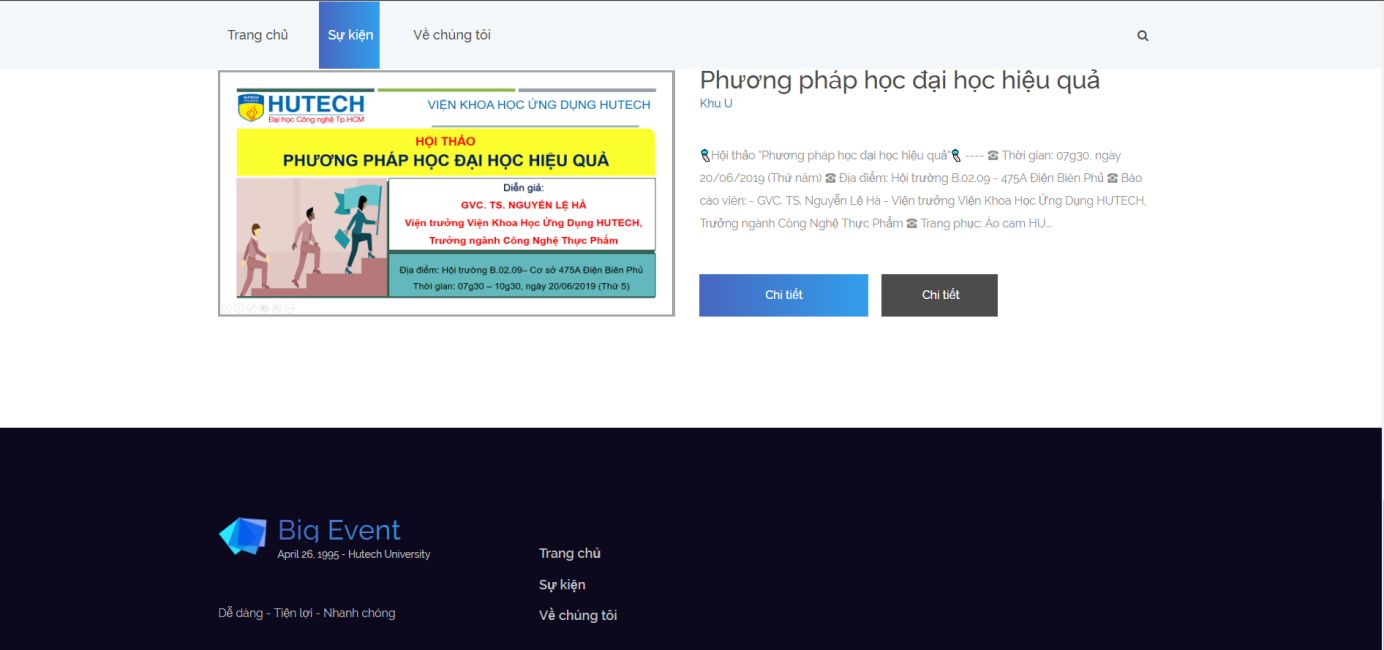




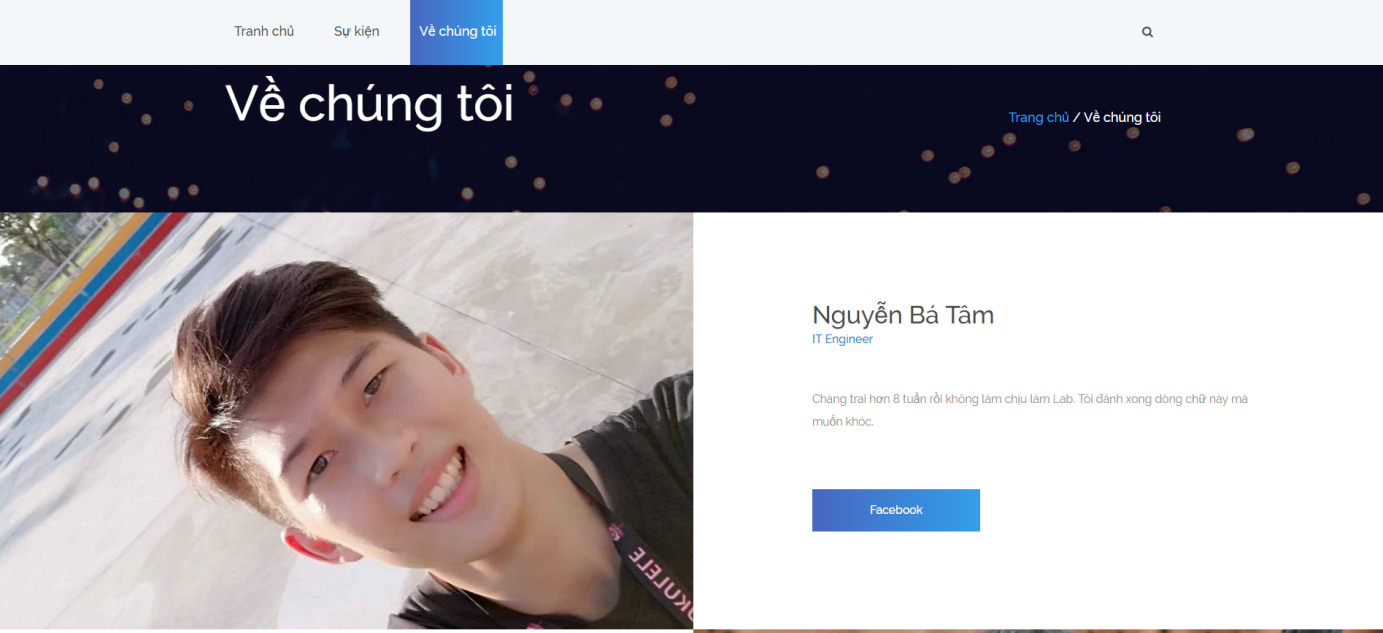


1. Giao diện sự kiện người xem





1. Giao diện về chúng tôi



**CHƯƠNG 4. KẾT LUẬN VÀ KIẾN NGHỊ**

* 1. **Kết quả đạt được**
* Xây dụng được một trang Webiste nói chung và hoàn thành đề tài nói riêng, kết quả đạt được không chỉ là một công việc xây dựng một trang web mà nó là cả một quá trình cố gắng không biết mỏi mệt trong vòng 12 tuần của nhóm.
* Trước một vấn đề về nhu cầu thông tin của sinh viên và sự phát triển mạnh mẽ của công nghệ thông tin thì việc tạo ra một trang web tin tức riêng giành cho các trường đại học là điều cần thiết và mang lại hiệu quả thực tế cao. Qua việc xây dựng là thực hiện đồ án cơ sở cũng giúp chúng em học hỏi thêm được nhiều điều về công việc của lập trình viên và nắm bắt được làm một website là như thế nào từ đó giúp chúng em có được tiền đề để phát triển thêm nhiều website khác.

1. **Những đóng góp**
2. *Nguyễn Bá Tâm*
3. Em đóng góp được gì vào đồ án môn học này ?

Trong quá trình làm đồ án em chưa có nhiều đóng góp cho nhóm cụ thể em làm được những công việc như sau:

* Viết đặc tả về yêu cầu hệ thống,giao diện và chức năng.
* Viết mô tả danh sách Use Case và danh sách Actor
* Viết quy trình kiểm thử Website
* Tạo chức năng Tìm kiếm sự kiện
* Tạo chức năng đăng nhập,đăng ký bằng tài khoản google
* Hoàn thành cuốn báo cáo tổng kết

1. Em học được gì sau khi đồ án cơ sở kết thúc ?

Trong quá trình làm đồ án cơ sở em đã học được rất nhiều điều cụ thể như sau:

* Học được kiến thức về công nghệ ASP.NET MVC
* Biết cách tạo ra một trang website
* Học được cách làm việc nhóm hiệu quả

1. Với vai trò trong nhóm em đánh giá mình như thế nào ?

Với vai trò được phân công trong nhóm em thấy mình chưa hoàn thành tốt các công việc được giao, vẫn cần nhiều sự giúp đỡ từ nhóm trưởng và còn rất nhiều thiếu sót cần phải cải thiện

1. *Lê Khôi Đệ*
2. Em đóng góp được gì vào đồ án môn học này ?

* Thiết kế cơ sở dữ liệu
* Code các chức năng xem, xóa, sửa, thêm sự kiện
* Thiết kế giao diện trang chủ, sự kiện, về chúng tôi

1. Em học được gì sau khi đồ án cơ sở kết thúc ?

* Học thêm nhiều kiến thức về Entity Framework 6 Code First
* Học được cách phân tích các mô hình DFD, BFD, ERD, Usecase
* Học được cách vẽ các mô hình DFD, BFD, ERD, Usecase

1. Với vai trò trong nhóm em đánh giá mình như thế nào ?

* Vai trò Leader : thất bại, vì còn chưa thúc đẩy thành viên trong nhóm để kết quả hơi bị tệ khi Deadline tới.
* Vai trò Designer : thất bại, vì dành rất ít thời gian quan tâm tìm hiểu thiết kế giao diện nên giao diện hiện tại chỉ mang tính chất cho có không đánh giá được đúng thực lực hiện tại vì có thể làm tốt hơn
* Vai trò Technical Master : thất bại, vì không giải quyết được các bài toán về các các chức năng như tìm kiếm, phân trang,...

1. **Kết luận chung**

Với mục tiêu đề ra ban đầu là Thiết kế và xây dựng Website “Cổng thông tin các thông tin tại Đại Học Công Nghệ (Hutech)” trên nền tảng công nghệ ASP.NET MVC hướng dẫn bởi giảng viên Lê Thị Ngọc Thơ trong vòng 12 tuần. Thì nhóm thấy cần phải phát triển đề tài hơn nữa vì còn hơi nghiệp dư và trau dồi kiến thức hơn nữa. Và mong rằng ý tưởng thô sơ mà chúng em thể hiện qua đề tài này sẽ được cô và các thầy chú ý. Nhóm em xin chân thành cảm ơn

**TÀI LIỆU THAO KHẢO**

[1] http://eitguide.net/mot-cai-nhin-tong-quan-ve-visual-studio-ide/

[2] https://codegym.vn/blog/2018/12/22/lap-trinh-c-la-gi/

[3] https://tndhuy.wordpress.com/2012/04/03/gi%E1%BB%9Bi-thi%E1%BB%87u-entity-framework-code-first/

[4] <https://www.youtube.com/playlist?list=PL-4b57MOnS8P7Lqxi8GieHEO1-1Iq21mb>

[5] [https://docs.microsoft.com/en-us/aspnet/?fbclid=IwAR1CHKdxuo0anACKVGZTDbX66P4hs7fkwdf8D2EptvsK2wNCtXpfX32gvkc#pivot=aspnet](https://docs.microsoft.com/en-us/aspnet/?fbclid=IwAR1CHKdxuo0anACKVGZTDbX66P4hs7fkwdf8D2EptvsK2wNCtXpfX32gvkc" \l "pivot=aspnet)

]6] <https://www.entityframeworktutorial.net/>

[7] <https://techmaster.vn/posts/33903/giai-thich-ve-mo-hinh-mvc?fbclid=IwAR356x_KhAbGOr5ls6vto6pyIJrRcAIKIS0Dxx6waxNOuq5AxQsUwJexLxw>