

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG**

-----oOo-----



**ĐỒ ÁN TỐT NGHIỆP
NGÀNH CÔNG NGHỆ THÔNG TIN**

HẢI PHÒNG 2013

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG
-----o0o-----

**TÌM HIỂU VÀ XÂY DỰNG MODUL QUẢN TRỊ
NGƯỜI DÙNG**

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY
Ngành: Công Nghệ Thông Tin

HẢI PHÒNG 2013

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

-----o0o-----

**TÌM HIỂU VÀ XÂY DỰNG MODUL QUẢN TRỊ
NGƯỜI DÙNG**

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY

Ngành: Công Nghệ Thông Tin

Sinh viên thực hiện: Trần Quang Vũ

Giáo viên hướng dẫn: Ths. Đỗ Văn Chiểu

Mã số sinh viên: 121177

HẢI PHÒNG 2013

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc

-----o0o-----

NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP

Sinh viên: Trần Quang Vũ

Mã số sinh viên: 121177

Lớp: CT1201

Ngành: Công nghệ thông tin

Tên đề tài: **Tìm hiểu và xây dựng modul quản trị người dùng**

NHIỆM VỤ ĐỀ TÀI

1. Nội dung và các yêu cầu cần giải quyết trong nhiệm vụ đề tài tốt nghiệp

a. Nội dung:

- Tìm hiểu về ngôn ngữ lập trình PHP
- Tìm hiểu về hệ quản trị cơ sở dữ liệu MySQL
-

b. Các yêu cầu cần giải quyết

- Xây dựng được tài liệu tổng quan về ngôn ngữ lập trình PHP và hệ quản trị cơ sở dữ liệu MySQL.

- Xây dựng một modul quản trị người dùng cho ứng dụng web kết hợp sử dụng ngôn ngữ lập trình PHP và hệ quản trị cơ sở dữ liệu MySQL.

2. Các số liệu cần thiết để thiết kế, tính toán.

3. Địa điểm thực tập

CÁN BỘ HƯỚNG DẪN ĐỀ TÀI TỐT NGHIỆP

Người hướng dẫn thứ nhất:

Họ và tên:

Học hàm, học vị:

Cơ quan công tác:

Nội dung hướng dẫn:

.....
.....
.....
.....

Người hướng dẫn thứ hai:

Họ và tên:

Học hàm, học vị:

Cơ quan công tác:

Nội dung hướng dẫn:

.....
.....
.....

Đề tài tốt nghiệp được giao ngày tháng năm 2013

Yêu cầu phải hoàn thành trước ngày tháng năm 2013

Đã nhận nhiệm vụ: Đ.T.T.N

Đã nhận nhiệm vụ: Đ.T.T.N

Sinh viên

Cán bộ hướng dẫn Đ.T.T.N

Hải phòng, ngày ... tháng ... năm 2013

HIỆU TRƯỞNG

GS.TS.NGŨT Trần Hữu Nghị

PHẦN NHẬN XÉT TÓM TẮT CỦA CÁN BỘ HƯỚNG DẪN

1. Tinh thần thái độ của sinh viên trong quá trình làm đề tài tốt nghiệp:

.....
.....
.....
.....
.....
.....

2. Đánh giá chất lượng của đề tài tốt nghiệp (so với nội dung yêu cầu đã đề ra trong nhiệm vụ đề tài tốt nghiệp)

.....
.....
.....
.....
.....
.....

3. Cho điểm của cán bộ hướng dẫn:
(Điểm ghi bằng số và chữ)

.....
.....
.....

2013

Ngày..... Tháng.....năm

Cán bộ hướng dẫn chính

(Ký, ghi rõ họ tên)

MỤC LỤC

MỤC LỤC.....	8
DANH SÁCH HÌNH ẢNH.....	18
DANH MỤC CÁC TỪ VIẾT TẮT	19
LỜI CẢM ƠN	21
GIỚI THIỆU	22
CHƯƠNG 1: GIỚI THIỆU BÀI TOÁN, LỰA CHỌN HƯỚNG TIẾP CẬN VÀ CÔNG NGHỆ	23
1.1 PHÁT BIỂU BÀI TOÁN.....	23
1.2 HƯỚNG TIẾP CẬN.....	23
1.3 LỰA CHỌN CÔNG NGHỆ.	26
1.4 CÔNG NGHỆ WEB.....	28
1.4.1 HTTP và HTTPS.....	28
1.4.2 Phương thức trong HTTP.....	29
1.4.3 HTML	29
1.4.4 Java Script	29
1.4.5 Tổng quan về PHP	30
1.4.6 Tổng quan về MySQL.....	39
1.4.7 Kết hợp PHP và MySQL trong ứng dụng Website.....	41
1.4.8 Giới thiệu về CSS.....	42
1.4.9 Apache và IIS	42
CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG.....	44
2.1 KHẢO SÁT VÀ PHÂN TÍCH HỆ THỐNG.....	44
2.1.1 Khảo sát.....	44
2.1.2 Phân tích hệ thống.....	45
2.2 THIẾT KẾ GIAO DIỆN	58
2.3 MỘT SỐ KỸ THUẬT LẬP TRÌNH.....	62
2.3.1 Kỹ thuật bảo mật tài khoản.	62
2.3.2 Kỹ thuật gửi mail.	62
2.3.3 Kỹ thuật khóa tài khoản.	63

2.3.4 Kỹ thuật phân quyền.....	64
CHƯƠNG 3: THỬ NGHIỆM CHƯƠNG TRÌNH	65
3.1 XÂY DỰNG HỆ THỐNG.....	65
3.2 MỘT SỐ GIAO DIỆN VÀ KẾT QUẢ KHI THỬ NGHIỆM HỆ THỐNG	65
KẾT LUẬN.....	68
TÀI LIỆU THAM KHẢO.....	69

DANH SÁCH HÌNH ẢNH

Hình 1. 1: Xuất ra trình duyệt	32
Hình 1. 2: Liên kết 2 chuỗi.....	32
Hình 1. 3: Biến trong PHP	33
Hình 1. 4: Hằng trong PHP	34
Hình 1. 5: Liên kết chuỗi và biến trong PHP	34
Hình 1. 6: Kiểu dữ liệu trong PHP	35
Hình 1. 7: Hàm GETTYPE	35
Hình 1. 8: Phương thức POST	36
Hình 1. 9: Loại dữ liệu trong MySQL.....	40
Hình 2. 1: Biểu đồ ngữ cảnh <i>Module quản trị người dùng</i>	48
Hình 2. 2: Biểu đồ phân rã chức năng <i>Module quản trị người dùng</i>	51
Hình 2. 3: Ma trận thực thể chức năng	52
Hình 2. 4: Mô hình ER.....	54
Hình 2. 5: Mô hình quan hệ	58
Hình 2. 6: Giao diện đăng nhập hệ thống	59
Hình 2. 7: Giao diện đăng ký tài khoản	59
Hình 2. 8: Giao diện đổi mật khẩu	60
Hình 2. 9: Giao diện quên mật khẩu	60
Hình 2. 10: Giao diện thông tin cá nhân	61
Hình 2. 11: Giao diện quản lý thông tin thành viên.....	61
Hình 3. 1: Đăng ký tài khoản thành công.....	57
Hình 3. 2: Trang cá nhân khi đã cập nhật thông tin.....	58
Hình 3. 3: Trang quản lý người dùng của Admin.....	58
Hình 3. 4: Trang phân quyền cho người dùng.....	59

DANH MỤC CÁC TỪ VIẾT TẮT

Stt	Tên viết tắt	Tên đầy đủ	Mô tả
1	HTML	Hyper Text Markup Language	Ngôn ngữ đánh dấu siêu văn bản
2	LAN	Local area network	Mạng cục bộ
3	IBM	International Business Machines	Tập đoàn công nghệ máy tính đa quốc gia
4	IP	Internet Protocol	Giao thức hướng dữ liệu được sử dụng bởi các máy chủ nguồn và đích để truyền dữ liệu trong một liên mạng và chuyển mạch gói.
5	OSI	Open Systems Interconnection Reference Model	Một thiết kế dựa vào nguyên lý tầng cấp, lý giải một cách trừu tượng kỹ thuật kết nối truyền thông giữa các máy vi tính và thiết kế giao thức mạng giữa chúng
6	TCP	Transmission Control Protocol	Sử dụng TCP, các ứng dụng trên các máy chủ được nối mạng có thể tạo các "kết nối" với nhau, mà qua đó chúng có thể trao đổi dữ liệu hoặc các gói tin.
7	HTTP	HyperText Transfer Protocol	Là giao thức liên hệ thông tin giữa Máy cung cấp dịch vụ (<i>Web server</i>) và Máy sử dụng dịch vụ (<i>Web client</i>)
8	HTTS	Hypertext Transfer Protocol Secure	Kết hợp giữa giao thức HTTP và giao thức bảo mật SSL hay TLS cho phép trao đổi thông tin một cách bảo mật trên Internet.
9	URL	Uniform Resource Locator	Được dùng để tham chiếu tới tài nguyên trên Internet.

Stt	Tên viết tắt	Tên đầy đủ	Mô tả
10	WWW	World Wide Web	Một không gian thông tin toàn cầu mà mọi người có thể truy nhập (đọc và viết) qua các máy tính nối với mạng Internet
11	XHTML	Extensible HyperText Markup Language	Một ngôn ngữ đánh dấu có cùng các khả năng như HTML, nhưng có cú pháp chặt chẽ hơn
12	XML	eXtensible Markup Language	Ngôn ngữ đánh dấu với mục đích chung do W3C đề nghị, để tạo ra các ngôn ngữ đánh dấu khác
13	CSS	Cascading Style Sheets	Các tập tin định kiểu theo tầng
19	IIS	Internet information server	Một dịch vụ tùy chọn của Windows NT Server cung cấp các tính năng về Website

LỜI CẢM ƠN

Trong lời đầu tiên của báo cáo đồ án tốt nghiệp “Tìm hiểu và xây dựng modul quản trị người dùng” này, em muốn gửi những lời cảm ơn và biết ơn chân thành nhất của mình tới tất cả những người đã hỗ trợ, giúp đỡ em về kiến thức và tinh thần trong quá trình thực hiện đồ án.

Trước hết, em xin chân thành cảm ơn Thầy Giáo - Ths. Đỗ Văn Chiêu, Giảng viên Khoa Công Nghệ Thông Tin, Trường ĐHDL Hải Phòng, người đã trực tiếp hướng dẫn, nhận xét, giúp đỡ em rất tận tình trong suốt quá trình thực hiện đồ án.

Xin chân thành cảm ơn GS.TS. NGUYỄN Trần Hữu Nghị Hiệu trưởng Trường Đại học Dân lập Hải Phòng, ban giám hiệu nhà trường, các thầy cô trong Khoa Công Nghệ Thông Tin và các phòng ban nhà trường đã tạo điều kiện tốt nhất cho em cũng như các bạn khác trong suốt thời gian học tập và làm tốt nghiệp.

Cuối cùng em xin gửi lời cảm ơn đến gia đình, bạn bè, người thân đã giúp đỡ động viên em rất nhiều trong quá trình học tập và làm Đồ án Tốt Nghiệp.

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 để em có thêm kinh nghiệm và tiếp tục hoàn thiện đồ án của mình.

Em xin chân thành cảm ơn!

Hải Phòng, ngày tháng năm 2013

Sinh viên

GIỚI THIỆU

Với bất kỳ trang web nào hiện nay cũng đều rất quan tâm đến việc quản lý người dùng của hệ thống, nó giúp người quản trị nắm được trình trạng hoạt động của trang web (lượt view, số người sử dụng hệ thống quản lý - đối với các trang web quản lý, lịch sử người dùng). Chính vì vậy việc phát triển và xây dựng module quản lý người dùng hệ thống mang một vai trò rất quan trọng trong việc xây dựng trang web. Do nhu cầu thực tế nên cần có 1 module làm công việc quản trị người dùng :

- Quản lý số lượng người dùng của hệ thống, thông tin cá nhân.
- Quản lý số lượng người dùng đang online hay offline, lịch sử hoạt động của người dùng.
- Phân quyền sử dụng cho người dùng.

Bố cục của đề án như sau. Chương 1: Giới thiệu bài toán, lựa chọn hướng tiếp cận và công nghệ. Chương 2: Phân tích thiết kế hệ thống. Chương 3: Giới thiệu một số kỹ thuật lập trình. Cuối cùng là phần kết luận và hướng phát triển tiếp theo của đề án.

CHƯƠNG 1: GIỚI THIỆU BÀI TOÁN, LỰA CHỌN HƯỚNG TIẾP CẬN VÀ CÔNG NGHỆ

1.1 PHÁT BIỂU BÀI TOÁN.

Module quản trị người dùng giúp người quản trị quản lý được thông tin của người dùng, số lượng các tài khoản, tài khoản online - offline , cũng như lịch sử hoạt động của từng tài khoản .

Module bao gồm các chức năng:

- Người dùng:
 - Đăng ký tài khoản và đăng nhập vào hệ thống
 - Cập nhật thông tin cá nhân , đổi mật khẩu lấy lại mật khẩu khi bị quên
 - Sử dụng các chức năng do người quản trị phân cho
- Người quản trị:
 - Quản trị người dùng hệ thống : Số lượng người dùng, thông tin cá nhân, trạng thái của người dùng.....
 - Xóa và khôi phục tài khoản bị xóa
 - Phân quyền theo chức năng cho từng tài khoản
 - Quản lý lịch sử của người dùng

1.2 HƯỚNG TIẾP CẬN.

- Có nhiều mô hình phát triển hệ thống . Trong đó mô hình dữ liệu tập trung, mô hình *Client-Server* và *Web-based* là các mô hình được sử dụng nhiều nhất .

Mô hình dữ liệu tập trung (*Centralized database model*): Trong mô hình này, các thành phần xử lý ứng dụng, phần mềm cơ sở dữ liệu và bản thân cơ sở dữ liệu đều ở trên một bộ xử lý. Ví dụ người dùng máy tính cá nhân có thể chạy các chương trình ứng dụng có sử dụng phần mềm cơ sở dữ liệu Oracle để truy nhập tới cơ sở dữ liệu nằm trên đĩa cứng của máy tính cá nhân đó. Từ khi các thành phần ứng dụng, phần mềm cơ sở dữ liệu và bản thân cơ sở dữ liệu cùng nằm trên một

máy tính thì ứng dụng đã thích hợp với mô hình tập trung. Hầu hết công việc xử lý luồng thông tin chính được thực hiện bởi nhiều tổ chức mà vẫn phù hợp với mô hình tập trung. Ví dụ một bộ xử lý của máy *mainframe* chạy phần mềm cơ sở dữ liệu IMS hoặc DB2 của IBM có thể cung cấp cho các trạm làm việc ở các vị trí phân tán sự truy nhập nhanh chóng tới cơ sở dữ liệu trung tâm. Tuy nhiên trong rất nhiều hệ thống như vậy, cả 3 thành phần của ứng dụng cơ sở dữ liệu đều thực hiện trên cùng một máy *mainframe* do vậy cấu hình này cũng thích hợp với mô hình tập trung

Mô hình *Client-Server* là một mô hình nổi tiếng trong mạng máy tính, được áp dụng rất rộng rãi và là mô hình của mọi trang *web* hiện có. Ý tưởng của mô hình này là máy con (đóng vai trò là máy khách) gửi một yêu cầu (*request*) để máy chủ (đóng vai trò người cung ứng dịch vụ), máy chủ sẽ xử lý và trả kết quả về cho máy khách. Thuật ngữ *server* được dùng cho những chương trình thi hành như một dịch vụ trên toàn mạng. Các chương trình *server* này chấp nhận tất cả các yêu cầu hợp lệ đến từ mọi nơi trên mạng, sau đó nó thi hành dịch vụ và trả kết quả về máy yêu cầu. Một chương trình được coi là *client* khi nó gửi các yêu cầu tới máy có chương trình *server* và chờ đợi câu trả lời từ *server*. Chương trình *server* và *client* nói chuyện với nhau bằng các thông điệp (*messages*) thông qua một cổng truyền thông liên tác IPC (*Interprocess Communication*). Để một chương trình *server* và một chương trình *client* có thể giao tiếp được với nhau thì giữa chúng phải có một chuẩn để nói chuyện, chuẩn này được gọi là giao thức. Nếu một chương trình *client* nào đó muốn yêu cầu lấy thông tin từ *server* thì nó phải tuân theo giao thức mà *server* đó đưa ra. Bản thân chúng ta khi cần xây dựng một mô hình *client/server* cụ thể thì ta cũng có thể tự tạo ra một giao thức riêng nhưng thường chúng ta chỉ làm được điều này ở tầng ứng dụng của mạng. Với sự phát triển mạng như hiện nay thì có rất nhiều giao thức chuẩn trên mạng ra đời nhằm đáp ứng nhu cầu phát triển này. Các giao thức chuẩn (ở tầng mạng và vận chuyển) được sử dụng rộng rãi nhất hiện nay như: giao thức TCP/IP, giao thức SNA của IBM, OSI, ISDN, X.25 hoặc giao thức LAN-to-LAN NetBIOS. Một máy tính chứa chương trình *server* được coi là một máy chủ hay máy phục vụ (*server*) và máy chứa chương trình *client* được coi là máy tớ (*client*). Mô hình mạng trên đó có các máy chủ và máy tớ giao tiếp với nhau theo 1

hoặc nhiều dịch vụ được gọi là mô hình *client/server*. Thực tế thì mô hình *client/server* là sự mở rộng tự nhiên và tiện lợi cho việc truyền thông liên tiến trình trên các máy tính cá nhân. Mô hình này cho phép xây dựng các chương trình *client/server* một cách dễ dàng và sử dụng chúng để liên tác với nhau để đạt hiệu quả hơn.

Web-based technology hay công nghệ dựa trên nền tảng *web* là một thuật ngữ dùng để chỉ những những ứng dụng (*application*) hay phần mềm được sử dụng dựa trên nền tảng *web*. Tức là những ứng dụng hay phần mềm có thể truy cập thông qua trình duyệt trên hệ thống mạng như *Internet* hay *intranet*. Những ứng dụng *web* chính là những phần mềm máy tính được mã hóa thông qua những ngôn ngữ được trình duyệt hỗ trợ như là HTML, JavaScript...

Những ứng dụng dựa trên nền tảng *web* hay ứng dụng *web* (*web application*) ngày càng trở lên rất phổ biến vì những ưu điểm vượt trội của nó, mà đặc biệt là ưu điểm to lớn đối với người sử dụng (hay người sử dụng cuối cùng) trên các máy trạm (*clients*).

Ưu điểm đối với phần máy trạm ở chỗ sử dụng những ứng dụng (*application*) hay phần mềm (chẳng hạn gmail, những điểm bán lẻ,...) mà không cần phải cài đặt chương trình gì mà chỉ cần chạy thông qua *web*. Hơn nữa việc *update* và duy trì hệ thống cũng không cần cài đặt gì tại máy trạm. Hiện tại, với FAST thì máy trạm cần phải cài đặt rất nhiều như: *Framework*, *Crystal report*, SQL, Chương trình, ...

Với việc sử dụng trình duyệt (*Browser*) thì người dùng có thể sử dụng máy tính tại bất kỳ đâu có kết nối *Internet* với đường truyền tốt đều có thể làm việc với máy chủ (*server*) từ rất xa. Ví dụ: Trong tương lai nếu FAST ứng dụng công nghệ *Web-based* thì máy chủ tại văn phòng cài đặt chương trình *Fast* và còn máy trạm (*client*) có thể sử dụng tại bất kỳ máy tính nào có kết nối *Internet*, điều này rất thuận lợi khi làm việc từ xa, làm việc trên toàn cầu.

Bên cạnh những ưu điểm vượt trội về máy trạm, những ứng dụng *web* còn rất nhiều ưu điểm khác như: Tự động *update* chương trình thông qua việc *update* tại

máy chủ, việc dùng trình duyệt làm việc có thể kết hợp với các ứng dụng *web* khác như mail, tìm kiếm. Người sử dụng có thể chạy chương trình trên mọi hệ điều hành như *windows, Linux, Mac...* bởi chúng ta chỉ cần có mỗi trình duyệt để làm việc. Ngoài ra, máy tính của chúng ta cũng ko cần đòi hỏi quá cao về cấu hình, đĩa trống....

Từ những phân tích trên với bài toán trong đề án này em lựa chọn mô hình *Web-based* và phân tích theo hướng chức năng.

1.3 LỰA CHỌN CÔNG NGHỆ.

Có 2 công nghệ được sử dụng trong phát triển phần mềm là công nghệ nguồn mở và công nghệ nguồn đóng.

- **Công nghệ nguồn đóng** là phần mềm mà mã nguồn không được công bố. Muốn sử dụng phần mềm nguồn đóng chỉ có một cách duy nhất là mua lại bản quyền sử dụng từ các nhà phân phối chính thức của hãng. Các hình thức tự do sao chép và sử dụng phần mềm nguồn đóng bị xem như là không hợp pháp. Các ngôn ngữ lập trình cho công nghệ nguồn đóng như là các ngôn ngữ lập trình mà mã nguồn được mã hóa khi đóng gói chương trình. Đối với lập trình *web* thì ASP.NET là một ngôn ngữ điển hình hiện nay đi cùng với hệ quản trị *SQL Server*.

ASP.NET là một nền tảng ứng dụng *web* (*web application framework*) được phát triển và cung cấp bởi Microsoft, cho phép những người lập trình tạo ra những trang *web* động, những ứng dụng *web* và những dịch vụ *web*. Lần đầu tiên được đưa ra thị trường vào tháng 2 năm 2002 cùng với phiên bản 1.0 của *.NET framework*, là công nghệ nối tiếp của Microsoft's *Active Server Pages(ASP)*. *ASP.NET* được biên dịch dưới dạng *Common Language Runtime (CLR)*, cho phép những người lập trình viết mã *ASP.NET* với bất kỳ ngôn ngữ nào được hỗ trợ bởi *.NET language*.

SQL Server: Là một hệ thống quản lý cơ sở dữ liệu (*Relational Database Management System (RDBMS)*) sử dụng *Transact-SQL* để trao đổi dữ liệu giữa *Client computer* và *SQL Server computer*. Một *RDBMS* bao gồm

databases, database engine và các ứng dụng dùng để quản lý dữ liệu và các bộ phận khác nhau trong RDBMS. Được tối ưu để có thể chạy trên môi trường cơ sở dữ liệu rất lớn (Very Large Database Environment) lên đến Tera-Byte và có thể phục vụ cùng lúc cho hàng ngàn user. SQL Server 2005 có thể kết hợp với các server khác như Microsoft Internet Information Server (IIS), E-Commerce Server, Proxy Server....Các phiên bản được sử dụng phổ biến hiện nay là Microsoft SQL Server 2005 và Microsoft SQL Server 2008.

- **Công nghệ nguồn mở:** Phần mềm với mã nguồn được công bố và sử dụng một giấy phép nguồn mở (*General Public Licence – GPL*). Giấy phép này cho phép bất cứ ai cũng có thể nghiên cứu, thay đổi và cải tiến phần mềm, và phân phối phần mềm ở dạng chưa thay đổi hoặc đã thay đổi. Công nghệ nguồn mở là công nghệ đang được phổ biến và ưa dùng hiện nay. Với bộ mã nguồn mở đồ sộ với các module, template... hoàn toàn miễn phí hoặc miễn phí một phần đáp ứng nhu cầu xây dựng hình ảnh, mạng lưới kinh doanh, giải trí cho các cá nhân và tổ chức. Cá nhân và tổ chức có thể sử dụng trực tiếp hoặc tùy chỉnh theo mục đích, yêu cầu sử dụng của mình một cách dễ dàng. Các ngôn ngữ lập trình cho công nghệ nguồn mở như là các ngôn ngữ lập trình mà mã nguồn là mã nguồn mở. Đối với lập trình *web* thì PHP là một ngôn ngữ điển hình hiện nay kết hợp với hệ quản trị cơ sở dữ liệu MySQL.

PHP (viết tắt hội quy "*PHP: Hypertext Preprocessor*") là một ngôn ngữ lập trình kịch bản hay một loại mã lệnh chủ yếu được dùng để phát triển các ứng dụng viết cho máy chủ, mã nguồn mở, dùng cho mục đích tổng quát. Nó rất thích hợp với *web* và có thể dễ dàng nhúng vào trang HTML. Do được tối ưu hóa cho các ứng dụng *web*, tốc độ nhanh, nhỏ gọn, cú pháp giống C và Java, dễ học và thời gian xây dựng sản phẩm tương đối ngắn hơn so với các ngôn ngữ khác nên PHP đã nhanh chóng trở thành một ngôn ngữ lập trình *web* phổ biến nhất thế giới.

Ngôn ngữ, các thư viện, tài liệu gốc của PHP được xây dựng bởi cộng đồng và có sự đóng góp rất lớn của Zend Inc., công ty do các nhà phát triển cốt lõi của

PHP lập nên nhằm tạo ra một môi trường chuyên nghiệp để đưa PHP phát triển ở quy mô doanh nghiệp.

MySQL là *relational database management system (RDBMS)* tạm dịch là hệ thống quản lý cơ sở dữ liệu MySQL . Là hệ quản trị cơ sở dữ liệu mã 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 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. MySQL miễn phí hoàn toàn cho nên bạn có thể tải về MySQL từ trang chủ.

1.4 CÔNG NGHỆ WEB.

1.4.1 HTTP và HTTPS

- a) **HTTP** (Tiếng Anh: *HyperText Transfer Protocol* - Giao thức truyền tải siêu văn bản) là một trong năm giao thức chuẩn về mạng *Internet*, được dùng để liên hệ thông tin giữa Máy cung cấp dịch vụ (*Web server*) và Máy sử dụng dịch vụ (*Web client*) là giao thức *Client/Server* dùng cho *World Wide Web-WWW*, HTTP là một giao thức ứng dụng của bộ giao thức TCP/IP (các giao thức nền tảng cho *Internet*).

- b) **HTTPS** là viết tắt của "*Hypertext Transfer Protocol Secure*", Nó là một sự kết hợp giữa giao thức HTTP và giao thức bảo mật SSL hay TLS cho phép trao đổi thông tin một cách bảo mật trên *Internet*. Giao thức HTTPS thường được dùng trong các giao dịch nhạy cảm, cần tính bảo mật cao.

1.4.2 Phương thức trong HTTP

Phương thức được sử dụng để chỉ thị cho *server* thi hành một công việc đặc biệt. Hầu hết các *server* bao gồm cả IIS chỉ hỗ trợ 3 phương thức bởi vì các phương thức không được thi hành bởi hầu hết các trình duyệt.

- a) **GET**: Phương thức này chỉ thị cho HTTP *server* gửi đến một đối tượng (các *Website*, file, hình ảnh...) bằng địa chỉ URL. Phương thức này chỉ dùng trên HTTP *Client*.
- b) **HEAD**: Phương thức này cũng giống như *GET*, tuy nhiên nó chỉ trả về thông tin header của đối tượng chứ không phải toàn bộ dữ liệu.
- c) **POST**: Phương thức này được sử dụng bởi HTTP *Client* để gửi một đối tượng lên *server*.

1.4.3 HTML

HTML (tiếng Anh, viết tắt cho *HyperText Markup Language*, hay là "Ngôn ngữ Đánh dấu Siêu văn bản") là một ngôn ngữ đánh dấu được thiết kế ra để tạo nên các trang *web* với các mẫu thông tin được trình bày trên *World Wide Web*. HTML được định nghĩa như là một ứng dụng đơn giản của SGML và được sử dụng trong các tổ chức cần đến các yêu cầu xuất bản phức tạp. HTML đã trở thành một chuẩn *Internet* do tổ chức *World Wide Web Consortium (W3C)* duy trì. Phiên bản chính thức mới nhất của HTML là HTML 4.01 (1999). Sau đó, các nhà phát triển đã thay thế nó bằng XHTML. Hiện nay, HTML đang được phát triển tiếp với phiên bản HTML5 hứa hẹn mang lại diện mạo mới cho *Web*.

1.4.4 Java Script

JavaScript là một ngôn ngữ lập trình kịch bản dựa trên đối tượng được phát triển từ các ý niệm nguyên mẫu. Ngôn ngữ này được dùng rộng rãi cho các trang

web, nhưng cũng được dùng để tạo khả năng viết script sử dụng các đối tượng nằm sẵn trong các ứng dụng. Nó vốn được phát triển bởi *Brendan Eich* tại Hãng truyền thông *Netscape* với cái tên đầu tiên *Mocha*, rồi sau đó đổi tên thành *LiveScript*, và cuối cùng thành *JavaScript*. Giống *Java*, *JavaScript* có cú pháp tương tự *C*. *Js* là phần mở rộng thường được dùng cho tập tin mã nguồn *JavaScript*.

JavaScript là một ngôn ngữ lập trình dựa trên nguyên mẫu với cú pháp phát triển từ *C*. Giống như *C*, *JavaScript* có khái niệm từ khóa, do đó *JavaScript* gần như không thể được mở rộng.

1.4.5 Tổng quan về PHP

Lịch sử phát triển

PHP

Được phát triển từ một sản phẩm có tên là *PHP/FI*. *PHP/FI* do *Rasmus Lerdorf* tạo ra năm 1995, ban đầu được xem như là một tập con đơn giản của các mã kịch bản *Perl* để theo dõi tình hình truy cập đến bản sơ yếu lý lịch của ông trên mạng. Ông đã đặt tên cho bộ mã kịch bản này là '*Personal Home Page Tools*'.

PHP 3

PHP 3.0 là phiên bản đầu tiên cho chúng ta thấy một hình ảnh gần gũi với các phiên bản *PHP* mà chúng ta được biết ngày nay. Nó đã được *Andi Gutmans* và *Zeev Suraski* tạo ra năm 1997 sau khi viết lại hoàn toàn bộ mã nguồn trước đó. *PHP 3.0* đã chính thức được công bố vào tháng 6 năm 1998, sau thời gian 9 tháng được cộng đồng kiểm nghiệm.

PHP 4

Vào mùa đông năm 1998, ngay sau khi *PHP 3.0* chính thức được công bố, *Andi Gutmans* và *Zeev Suraski* đã bắt đầu bắt tay vào việc viết lại phần lõi của *PHP*. Một động cơ mới, có tên '*Zend Engine*' (ghép từ các chữ đầu trong tên của *Zeev* và *Andi*), đã đáp ứng được các nhu cầu thiết kế này một cách thành công, và lần đầu tiên được giới thiệu vào giữa năm 1999. *PHP 4.0*, dựa trên động cơ này, và đi kèm

với hàng loạt các tính năng mới bổ sung, đã chính thức được công bố vào tháng 5 năm 2000, gần 2 năm sau khi bản PHP 3.0 ra đời.

PHP 5

Sự thành công hết sức to lớn của PHP 4.0 đã không làm cho nhóm phát triển PHP tự mãn. Cộng đồng PHP đã nhanh chóng giúp họ nhận ra những yếu kém của PHP 4 đặc biệt với khả năng hỗ trợ lập trình hướng đối tượng (OOP), xử lý XML, không hỗ trợ giao thức máy khách mới của MySQL 4.1 và 5.0, hỗ trợ dịch vụ *web* yếu. Những điểm này chính là mục đích để Zeev và Andi viết *Zend Engine 2.0*, lõi của PHP 5.0. Ngày 29 tháng 6 năm 2003, PHP 5 Beta 1 đã chính thức được công bố để cộng đồng kiểm nghiệm. Đó cũng là phiên bản đầu tiên của *Zend Engine 2.0*. Phiên bản Beta 2 sau đó đã ra mắt vào tháng 10 năm 2003 với sự xuất hiện của hai tính năng rất được chờ đợi: *Iterators*, *Reflection* nhưng *namespaces* một tính năng gây tranh cãi khác đã bị loại khỏi mã nguồn. Ngày 21 tháng 12 năm 2003: PHP 5 *Beta 3* đã được công bố để kiểm tra với việc phân phối kèm với Tidy, bỏ hỗ trợ *Windows 95*, khả năng gọi các hàm PHP bên trong XSLT, sửa chữa nhiều lỗi và thêm khá nhiều hàm mới. PHP 5 bản chính thức đã ra mắt ngày 13 tháng 7 năm 2004 sau một chuỗi khá dài các bản kiểm tra thử bao gồm *Beta 4*, *RC 1*, *RC2*, *RC3*. Mặc dù coi đây là phiên bản sản xuất đầu tiên nhưng PHP 5.0 vẫn còn một số lỗi trong đó đáng kể là lỗi xác thực HTTP.

a) Cấu trúc cơ bản

PHP cũng có thẻ bắt đầu và kết thúc giống với ngôn ngữ HTML. Chỉ khác, đối với PHP chúng ta có nhiều cách để thể hiện.

Cách 1: Cú pháp chính:

```
<?php Mã lệnh PHP ?>
```

Cách 2: Cú pháp ngắn gọn

```
<? Mã lệnh PHP ?>
```

Cách 3: Cú pháp giống với ASP.

```
<% Mã lệnh PHP %>
```


Cách 4: Cú pháp bắt đầu bằng *script*

```
<script language=php>  
  
.....  
  
</script>
```

Mặc dù có 4 cách thể hiện. Nhưng đối với 1 lập trình viên có kinh nghiệm thì việc sử dụng cách 1 vẫn là lựa chọn tối ưu.

Trong PHP để kết thúc 1 dòng lệnh chúng ta sử dụng dấu ";"

Để chú thích 1 đoạn dữ liệu nào đó trong PHP ta sử dụng dấu "//" cho từng dòng. Hoặc dùng cặp thẻ "/*.....*/" cho từng cụm mã lệnh.

Ví dụ: <?php echo "Hello world!"; ?>

b) Xuất giá trị ra trình duyệt

Để xuất dữ liệu ra trình duyệt chúng ta có những dòng cú pháp sau:

```
+ echo "Thông tin";  
  
+ printf "Thông tin";
```

Thông tin bao gồm: biến, chuỗi, hoặc lệnh HTML

```
1 <?php  
2 Echo "Hello word";  
3 Printf"<br><font color=red>Who Are You ?</font>";  
4 ?>  
5
```

Hình 1. 1: Xuất ra trình duyệt

Nếu giữa hai chuỗi muốn liên kết với nhau ta sử dụng dấu "."

```
1 <?php  
2 Echo "Hello"."who are you ?";  
3 ?>  
4
```

Hình 1. 2: Liên kết 2 chuỗi

c) Biến, hằng, chuỗi và các kiểu dữ liệu

- **Biến**

Biến được xem là vùng nhớ dữ liệu tạm thời. Và giá trị có thể thay đổi được. Biến được bắt đầu bằng ký hiệu "\$". Và theo sau chúng là 1 từ, 1 cụm từ nhưng phải viết liền hoặc có gạch dưới.

Một biến được xem là hợp lệ khi nó thỏa các yếu tố:

- + Tên của biến phải bắt đầu bằng dấu gạch dưới và theo sau là các ký tự, số hay dấu gạch dưới.
- + Tên của biến không được phép trùng với các từ khóa của PHP.

Trong PHP để sử dụng 1 biến chúng ta thường phải khai báo trước, tuy nhiên đối với các lập trình viên khi sử dụng họ thường xử lý cùng một lúc các công việc, nghĩa là vừa khai báo vừa gán dữ liệu cho biến.

Bản thân biến cũng có thể gán cho các kiểu dữ liệu khác. Và tùy theo ý định của người lập trình mong muốn trên chúng.

```
1 <?
2 $a= 100 // biến a ở đây có giá trị là 100.
3 $a= "PHP is easy" // Biến a ở đây có giá trị "PHP Is easy".
4 Biena=123 //Có lỗi vì bắt đầu 1 biến phải có dấu "$"
5 $123a="PHP" //Có lỗi vì phần tên bắt đầu của biến là dạng số.
6 ?>
```

Hình 1. 3: Biến trong PHP

- **Hằng**

Nếu biến là cái có thể thay đổi được thì ngược lại hằng là cái chúng ta không thể thay đổi được. Hằng trong PHP được định nghĩa bởi hàm *define* theo cú pháp: *define (string tên_hằng, giá_trị_hằng)*.

Cũng giống với biến hằng được xem là hợp lệ thì chúng phải đáp ứng 1 số yếu tố:

- + Hằng không có dấu "\$" ở trước tên.
- + Hằng có thể truy cập bất cứ vị trí nào trong mã lệnh
- + Hằng chỉ được phép gán giá trị duy nhất 1 lần.

+ Hằng thường viết bằng chữ in để phân biệt với biến

```
1 <?
2 define ("C", "COMPANY");
3 define ("YELLOW", "#ffff00");
4 echo "Gia tri cua C la". C;
5 ?>
```

Hình 1. 4: Hằng trong PHP

- **Chuỗi**

Chuỗi là một nhóm các ký tự, số, khoảng trắng, dấu ngắt được đặt trong các dấu nháy.

Ví dụ:

```
'Hello'
```

Để tạo 1 biến chuỗi, chúng ta phải gán giá trị chuỗi cho 1 biến hợp lệ.

Ví dụ:

```
$first_name= "Nguyen";
```

```
$last_name= 'Van A';
```

Để liên kết 1 chuỗi và 1 biến chúng ta thường sử dụng dấu "."

```
1 <?php
2 $test="QHOnline.Info";
3 echo "welcome to".$test;
4 echo "<br><font color=red>welcome to".$test."</font><br>";
5 ?>
6
```

Hình 1. 5: Liên kết chuỗi và biến trong PHP

- **Kiểu dữ liệu**

Các kiểu dữ liệu khác nhau chiếm các lượng bộ nhớ khác nhau và có thể được xử lý theo cách khác nhau khi chúng được theo tác trong 1 *script*.

Trong PHP chúng ta có 6 kiểu dữ liệu chính như sau:

Kiểu Dữ Liệu	Vi dụ	Mô Tả
Integer	10	Một số nguyên
Double	5.208	Kiểu số thực
String	"How are you ?"	Một tập hợp các ký tự
Boolean	True or False	Giá trị true hoặc false
Object	Hướng đối tượng trong PHP	
Array	Mảng trong PHP, chứa các phần tử.	

Hình 1. 6: Kiểu dữ liệu trong PHP

Chúng ta có thể sử dụng hàm dựng sẵn **GETTYPE()** của PHP4 để kiểm tra kiểu của bất kỳ biến.

```

1 <?php
2 $a= 5;
3 echo gettype($a); // Integer.
4 $a="qhonline.info";
5 echo gettype($a); //String
6 ?>
7 |

```

Hình 1. 7: Hàm GETTYPE

d) Các phương thức được sử dụng trong lập trình PHP

Có 2 phương thức được sử dụng trong lập trình PHP là *GET* và *POST*

- **Phương thức GET**

Phương thức này cũng được dùng để lấy dữ liệu từ *form* nhập liệu. Tuy nhiên nhiệm vụ chính của nó vẫn là lấy nội dung trang dữ liệu từ *web server*.

Ví dụ:

Với url sau: `shownews.php?id=50`

Vậy với trang *shownews* ta dùng hàm `$_GET['id']` sẽ được giá trị là 50.

- **Phương thức POST**

Phương thức này được sử dụng để lấy dữ liệu từ *form* nhập liệu. Và chuyển chúng lên trình chủ *webservice*.

```

<?php
    echo "Welcome ".$_POST['hoten']."!";
?>
<html>
    <form name="test" action="#" method="POST">
        Họ tên <input type="text" name="hoten"/>
        <input type="submit" name="OK" value="OK"/>
    </form>
</html>

```

Hình 1. 8: Phương thức POST

e) *Cookie* và *Session* trong PHP

Cookie và *Session* là hai phương pháp sử dụng để quản lý các phiên làm việc giữa người sử dụng và hệ thống

- ***Cookie***

Cookie là 1 đoạn dữ liệu được ghi vào đĩa cứng hoặc bộ nhớ của máy người sử dụng. Nó được trình duyệt gửi ngược lên lại *server* mỗi khi browser tải 1 trang *web* từ *server*.

Những thông tin được lưu trữ trong *cookie* hoàn toàn phụ thuộc vào *Website* trên *server*. Mỗi *Website* có thể lưu trữ những thông tin khác nhau trong *cookie*, ví dụ thời điểm lần cuối ta ghé thăm *Website*, đánh dấu ta đã login hay chưa, v.v...

Cookie được tạo ra bởi *Website* và gửi tới browser, do vậy 2 *Website* khác nhau (cho dù cùng host trên 1 *server*) sẽ có 2 *cookie* khác nhau gửi tới *browser*. Ngoài ra, mỗi browser quản lý và lưu trữ *cookie* theo cách riêng của mình, cho nên 2 *browser* cùng truy cập vào 1 *Website* sẽ nhận được 2 *cookie* khác nhau

1. Để thiết lập *cookie* ta sử dụng cú pháp:

```
Setcookie("tên cookie", "giá trị", thời gian sống)
```

Tên *cookie* là tên mà chúng ta đặt cho phiên làm việc.

Giá trị là thông số của tên *cookie*.

Ví dụ: `setcookie("name", "admin", time()+3600);`

2. Để sử dụng lại *cookie* vừa thiết lập, chúng ta sử dụng cú pháp:

Cú pháp: `$_COOKIE["tên cookies"]`

Tên *cookie* là tên mà chúng ta thiết lập phía trên.

3. Để hủy 1 *cookie* đã được tạo ta có thể dùng 1 trong 2 cách sau:

+ Cú pháp: `setcookie("Tên cookie")`

Gọi hàm *setcookie* với chỉ duy nhất tên *cookie* mà thôi

+ Dùng thời gian hết hạn *cookie* là thời điểm trong quá khứ.

Ví dụ: `setcookie("name", "admin", time() - 3600);`

- **Session**

Một cách khác quản lý người sử dụng là *Session*. *Session* được hiểu là khoảng thời gian người sử dụng giao tiếp với 1 ứng dụng. Một *session* được bắt đầu khi người sử dụng truy cập vào ứng dụng lần đầu tiên, và kết thúc khi người sử dụng thoát khỏi ứng dụng. Mỗi *session* sẽ có được cấp một định danh (ID) khác nhau.

- Để thiết lập 1 *session* ta sử dụng cú pháp: `session_start()`

Đoạn code này phải được nằm trên các kịch bản HTML. Hoặc những lệnh *echo*, *printf*.

Để thiết lập 1 giá trị *session*, ngoài việc cho phép bắt đầu thực thi *session*. Chúng ta còn phải đăng ký 1 giá trị *session*. Để tiện cho việc gán giá trị cho *session* đó.

Ta có cú pháp sau: `session_register("Name")`

4. Giống với *cookie*. Để sử dụng giá trị của *session* ta sử dụng mã lệnh sau:

Cú pháp: `$_SESSION["name"]`

Với Name là tên mà chúng ta sử dụng hàm `session_register("name")` để khai báo.

5. Để hủy bỏ giá trị của *session* ta có những cách sau:

`session_destroy()` // Cho phép hủy bỏ toàn bộ giá trị của *session*

`session_unset()` // Cho phép hủy bỏ *session* .

f) Hàm

Để giảm thời gian lặp lại 1 thao tác *code* nhiều lần, PHP hỗ trợ người lập trình việc tự định nghĩa cho mình những hàm có khả năng lặp lại nhiều lần trong *Website*. Việc này cũng giúp cho người lập trình kiểm soát mã nguồn một cách mạch lạc. Đồng thời có thể tùy biến ở mọi trang. Mà không cần phải khởi tạo hay viết lại mã lệnh như HTML thuần

- **Hàm tự định nghĩa**

Cú pháp:

```
function function_name()  
{  
    //Lệnh thực thi  
}
```

Tên hàm có thể là một tổ hợp bất kỳ những chữ cái, con số và dấu gạch dưới, nhưng phải bắt đầu từ chữ cái và dấu gạch dưới.

- **Hàm tự định nghĩa với các tham số**

Cú pháp:

```
function function_name($gt1,$gt2)  
{  
    //Lệnh thực thi  
}
```

- **Hàm tự định nghĩa với giá trị trả về**

Cú pháp:

```
function function_name(Có hoặc không có đối số)  
{  
    // Lệnh thực thi  
    return giatri;  
}
```

- **Gọi lại hàm**

PHP cung cấp nhiều hàm cho phép triệu gọi lại file. Như hàm `include("URL đến file")`, `require("URL Đến file")`.

Ngoài hai cú pháp trên còn có `include_once()`, `require_once()`. Hai hàm này cũng có trách nhiệm gọi lại hàm. Những chúng sẽ chỉ gọi lại duy nhất 1 lần mà thôi.

1.4.6 Tổng quan về MySQL

MySQL là hệ quản trị dữ liệu miễn phí, được tích hợp sử dụng chung với *apache*, PHP. Chính yếu tố phát triển trong cộng đồng mã nguồn mở nên MySQL đã qua rất nhiều sự hỗ trợ của những lập trình viên yêu thích mã nguồn mở. MySQL cũng có cùng một cách truy xuất và mã lệnh tương tự với ngôn ngữ SQL. Nhưng MySQL không bao quát toàn bộ những câu truy vấn cao cấp như SQL. Về bản chất MySQL chỉ đáp ứng việc truy xuất đơn giản trong quá trình vận hành của *Website* nhưng hầu hết có thể giải quyết các bài toán trong PHP.

a) Khởi động và sử dụng

Chúng ta sử dụng *command* như sau:

```
Mysql -hname -uuser -ppass
```

Để truy cập vào cơ sở dữ liệu.

Hoặc sử dụng bộ *appserv* để vào nhanh hơn theo đường dẫn sau:

```
Start/ Appserv/ Mysql command Line client
```

Sau đó nhập *password* mà chúng ta đã đặt vào.

b) Một số thuật ngữ

NULL: Giá trị cho phép rỗng.

AUTO_INCREMENT: Cho phép giá trị tăng dần (tự động).

UNSIGNED: Phải là số nguyên dương

PRIMARY KEY: Cho phép nó là khóa chính trong bảng.

c) Loại dữ liệu trong MySQL

Kiểu dữ liệu	Mô Tả
Char	Định dạng text có chiều dài từ 0->255
Varchar	Định dạng text có chiều dài từ 0->255
Text	Định dạng text có chiều dài 0->65535
Longtext	Định dạng text có chiều dài 0->4294967215
INT	Định dạng số có chiều dài từ 0->4294967215
Float	Định dạng số thập phân có chiều dài nhỏ
Double	Định dạng số thập phân có chiều dài lớn
Date	Định dạng thời gian theo định dạng: YYYY-MM-DD
DateTime	Định dạng thời gian theo định dạng: YYYY-MM-DD HH:MM:SS

Hình 1. 9: Loại dữ liệu trong MySQL

d) Những cú pháp cơ bản

- Tạo một cơ sở dữ liệu:

```
CREATE DATABASE tên_cơ_sở_dữ_liệu;
```

Cú pháp sử dụng cơ sở dữ liệu: Use tên_database;

Cú pháp thoát khỏi cơ sở dữ liệu: *Exit*

- Tạo một bảng trong cơ sở dữ liệu:

```
CREATE TABLE user (<tên_cột>
<mô_tả>, ..., <tên_cột_n>... .<mô_tả_n>)
```

- Hiện thị có bao nhiêu bảng: show tables;
- Hiện thị có bao nhiêu cột trong bảng: show columns from table;
- Thêm 1 cột vào bảng:

```
ALTER TABLE tên_bảng ADD <tên_cột> <thuộc_tính>
AFTER <tên_cột>
```

- Thêm giá trị vào bảng:

```
INSERT INTO Tên_bảng(tên_cột)
VALUES (Giá_trị_tương_ứng);
```

- Truy xuất dữ liệu:

```
SELECT tên_cột FROM Tên_bảng;
```

- Truy xuất dữ liệu với điều kiện:

```
SELECT tên_cột FROM Tên_bảng WHERE điều_kiện;
```

- Truy xuất dữ liệu và sắp xếp theo trình tự:

```
SELECT tên_cột FROM Tên_bảng
WHERE điều_kiện (có thể có where hoặc không)
ORDER BY Theo quy ước sắp xếp.
```

Trong đó quy ước sắp xếp bao gồm hai thông số là ASC (từ trên xuống dưới), DESC (từ dưới lên trên).

- Truy cập dữ liệu có giới hạn:

```
SELECT tên_cột FROM Tên_bảng
WHERE điều_kiện (có thể có where hoặc không)
LIMIT vị trí bắt đầu, số record muốn lấy ra
```

- Cập nhật dữ liệu trong bảng:

```
Update tên_bảng set tên_cột=Giá trị mới
WHERE (điều_kiện).
```

Nếu không có ràng buộc điều kiện, chúng sẽ cập nhật toàn bộ giá trị mới của các *record* trong bảng.

- Xóa dữ liệu trong bảng:

```
DELETE FROM tên_bảng WHERE (điều_kiện).
```

Nếu không có ràng buộc điều kiện, chúng sẽ xóa toàn bộ giá trị của các *record* trong bảng.

1.4.7 Kết hợp PHP và MySQL trong ứng dụng *Website*

a) Kết nối cơ sở dữ liệu

```
mysql_connect("hostname", "user", "pass");
```

b) Lựa chọn cơ sở dữ liệu

```
mysql_select_db("tên_CSDL");
```

c) Thực thi câu lệnh truy vấn

```
mysql_query("Câu truy vấn ở đây");
```

d) Đếm số dòng dữ liệu trong bảng

```
mysql_num_rows();
```

e) **Lấy dữ liệu từ bảng đưa vào mảng**

```
mysql_fetch_array();
```

f) **Đóng kết nối cơ sở dữ liệu**

```
mysql_close();
```

1.4.8 Giới thiệu về CSS

- **CSS** là các tập tin định kiểu theo tầng (*Cascading Style Sheets* (CSS)) được dùng để miêu tả cách trình bày các tài liệu viết bằng ngôn ngữ HTML và XHTML. Ngoài ra ngôn ngữ định kiểu theo tầng cũng có thể dùng cho XML, SVG, XUL. Các đặc điểm kỹ thuật của CSS được duy trì bởi *World Wide Web Consortium* (W3C). Thay vì đặt các thẻ qui định kiểu dáng cho văn bản HTML (hoặc XHTML) ngay trong nội dung của nó, bạn nên sử dụng CSS.
- Tác dụng của CSS: Hạn chế tối thiểu việc làm rối mã HTML của trang *Web* bằng các thẻ quy định kiểu dáng (chữ đậm, chữ in nghiêng, chữ có gạch chân, chữ màu), khiến mã nguồn của trang *Web* được gọn gàng hơn, tách nội dung của trang *Web* và định dạng hiển thị, dễ dàng cho việc cập nhật nội dung. Tạo ra các kiểu dáng có thể áp dụng cho nhiều trang *Web*, giúp tránh phải lặp lại việc định dạng cho các trang *Web* giống nhau.

1.4.9 Apache và IIS

- a) **Apache** hay là chương trình máy chủ HTTP là một chương trình dành cho máy chủ đối thoại qua giao thức HTTP. Apache chạy trên các hệ điều hành tương tự như *Unix*, *Microsoft Windows*, *Novell Netware* và các hệ điều hành khác. Apache đóng một vai trò quan trọng trong quá trình phát triển của mạng *web*.

Khi được phát hành lần đầu, Apache là chương trình máy chủ mã nguồn mở duy nhất có khả năng cạnh tranh với chương trình máy chủ tương tự của *Netscape Communications Corporation* mà ngày nay được biết đến qua tên thương mại *Sun Java System Web Server*... Từ tháng 4 năm 1996, Apache trở thành một chương trình máy chủ HTTP thông dụng nhất.

- b) **IIS (INTERNET INFORMATION SERVICES)** là một dịch vụ tùy chọn của *Windows NT Server* cung cấp các tính năng về *Website*. IIS là một thành phần cơ

bản để xây dựng một *Internet* hoặc *intranet server* trên nền Windows NT 4.0, Workstation và Win 95. IIS được tích hợp đầy đủ trong Windows NT 4.0. Với một bộ đầy đủ IIS và Windows NT 4.0 người sử dụng sẽ nhận được sự thuận tiện khi xây dựng một cơ chế bảo mật trên Windows NT *server* và Windows NT File System (NTFS)

CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

2.1 KHẢO SÁT VÀ PHÂN TÍCH HỆ THỐNG.

2.1.1 Khảo sát.

- Giới thiệu chung :

Với bất kỳ trang web nào hiện nay cũng đều rất quan tâm đến việc quản lý người dùng của hệ thống, nó giúp người quản trị nắm được trình trạng hoạt động của trang web (lượt view, số người sử dụng hệ thống quản lý- đối với các trang web quản lý, lịch sử người dùng). Chính vì vậy việc phát triển và xây dựng module quản lý người dùng hệ thống mang một vai trò rất quan trọng trong việc xây dựng trang web.

Trong đồ án này em xin đi sâu nghiên cứu về việc xây dựng module quản lý người dùng với 3 vai trò quan trọng : Người quản trị, Người dùng hệ thống(có tài khoản trong hệ thống), Người dùng bình thường (chưa có tài khoản).

- Các yêu cầu xây dựng module:

Người quản trị có thể quản lý số lượng, tình trạng người sử dụng hệ thống đối với người dùng có tài khoản hoặc người dùng không có tài khoản (Khách) .

Phân quyền theo các chức năng hệ thống cho người dùng có tài khoản.....

Quản lý lịch sử người dùng (lịch sử đăng nhập-đăng xuất,tạo tài khoản,xóa tài khoản.....)

Người dùng có thể xem và thay đổi thông tin cá nhân và thực hiện các quyền mà người quản trị cho phép .

Khi người sử dụng chức năng quên mật khẩu, hệ thống sẽ tự động gửi *Email* phản hồi cho khách . Khách hàng sẽ phải cung cấp thông tin đầy đủ chính xác cùng với một mã xác nhận tự động (*captcha*) để lấy lại thông tin mật khẩu của tài khoản .

2.1.2 Phân tích hệ thống

a) Quy trình hoạt động của hệ thống đối với module quản lý người dùng

Khi khách đăng nhập vào hệ thống, hệ thống kiểm tra xem người dùng có mật khẩu đăng nhập chưa, nếu chưa có thì khách cần đăng ký tài khoản. Khi khách đăng ký tài khoản, hệ thống kiểm tra xem người dùng đã nhập đầy đủ thông tin chưa, nếu chưa thì hệ thống thông báo người dùng cần “**Nhập đầy đủ thông tin!**”. Sau đó hệ thống kiểm tra xem mật khẩu nhập lại có đúng không, nếu sai hệ thống yêu cầu nhập lại, nếu đúng thì hệ thống tiếp tục kiểm tra mã an toàn xem nhập mà an toàn có đúng không, nếu sai thì hệ thống yêu cầu nhập lại mã an toàn. Nếu đúng hết hệ thống kiểm tra xem tài khoản đã có chưa nếu có thì hệ thống thông báo “**Tài khoản đã có**”. Nếu chưa có thì hoàn thành quá trình đăng ký, hệ thống tự nhảy sang trang đăng nhập

Khi người dùng quên mật khẩu, người dùng sẽ kích vào “**Quên mật khẩu**” hệ thống đưa khách đến trang **Thay đổi mật khẩu**, người dùng cần nhập tài khoản, mật khẩu mới, nhập lại mật khẩu mới, mail đăng ký của tài khoản.

Nếu đã đủ các điều kiện thì người dùng đăng nhập thành công vào hệ thống, và trong đó người dùng được sử dụng các chức năng mà người quản trị đã phân cho mình

Người quản trị đăng nhập vào hệ thống hệ thống kiểm tra quyền của người quản trị, nếu đủ điều kiện thì cho vào trang quản lý thông tin người dùng, nếu không đủ điều kiện thì không cho vào

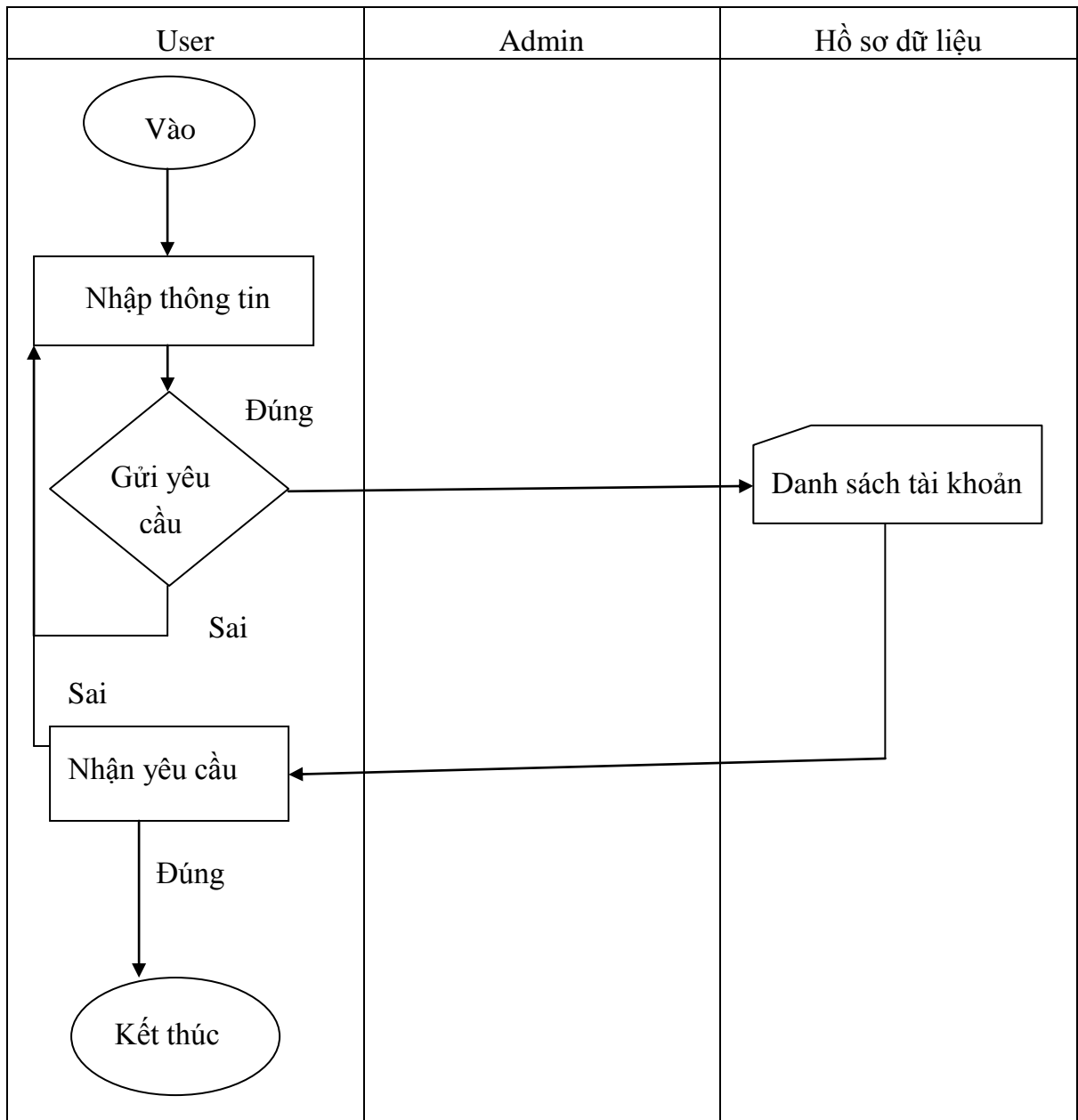
b) Các chức năng của người dùng trong hệ thống:

- Admin :
 - Quản lý thông tin cá nhân
 - Trạng thái của các tài khoản đăng nhập hệ thống
 - Thông tin của các tài khoản, lịch sử
 - Phân quyền theo chức năng cho các tài khoản
- User :

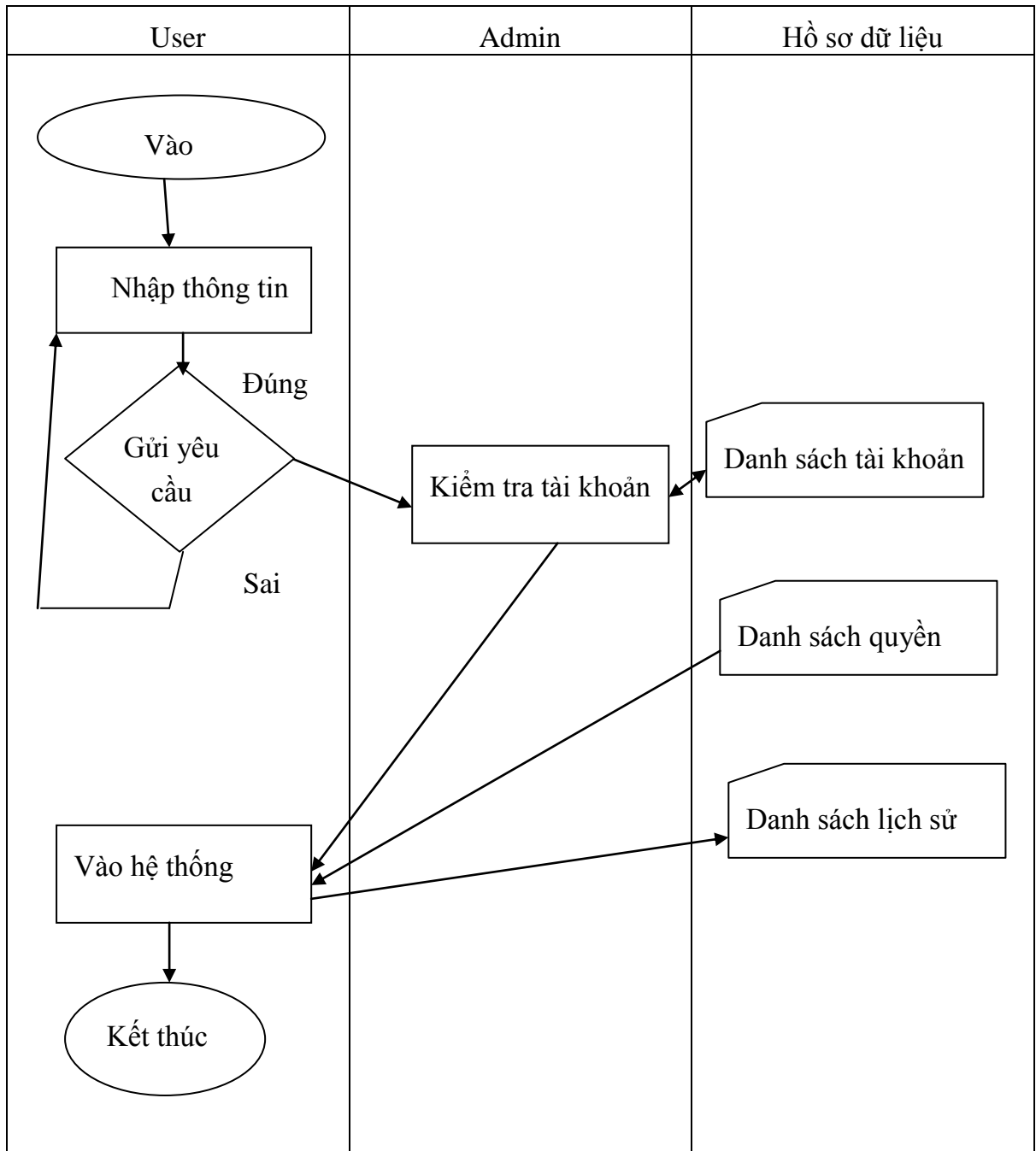
- Đăng ký tài khoản
- Đăng nhập, đăng xuất
- Cập nhật thông tin cá nhân
- Sử dụng các chức năng được cấp trong hệ thống

c) Lưu đồ hoạt động

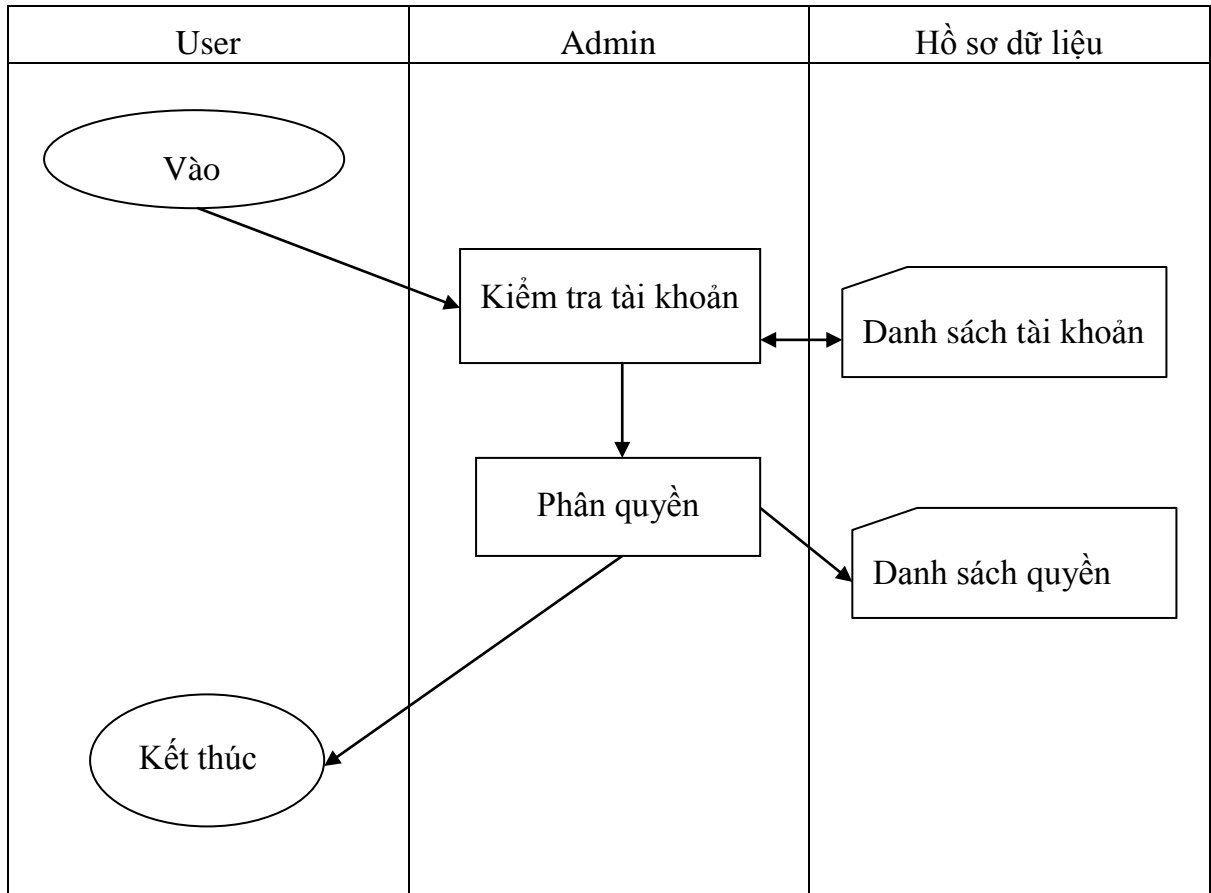
- Cho tiến trình đăng ký



- Tiến trình đăng nhập



- Tiến trình phân quyền



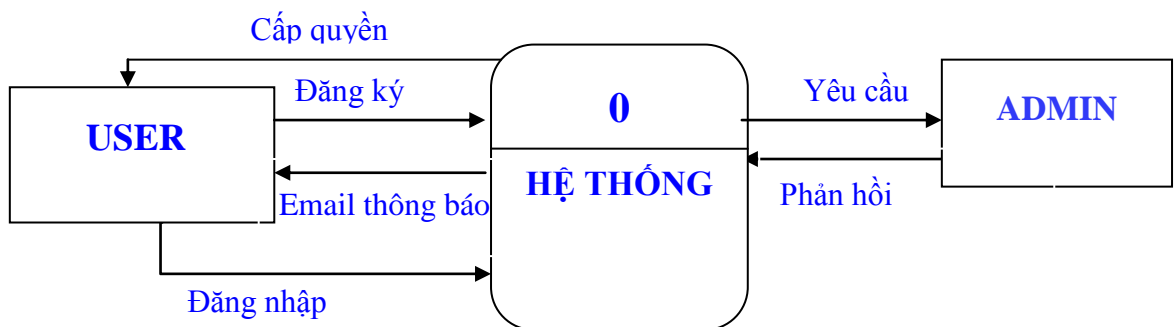
d) Vẽ biểu đồ ngữ cảnh

Xác định tác nhân

- USER

- ADMIN

Vẽ biểu đồ



Hình 2. 1: Biểu đồ ngữ cảnh *Module quản trị người dùng*

e) **Xác định biểu đồ phân rã chức năng**

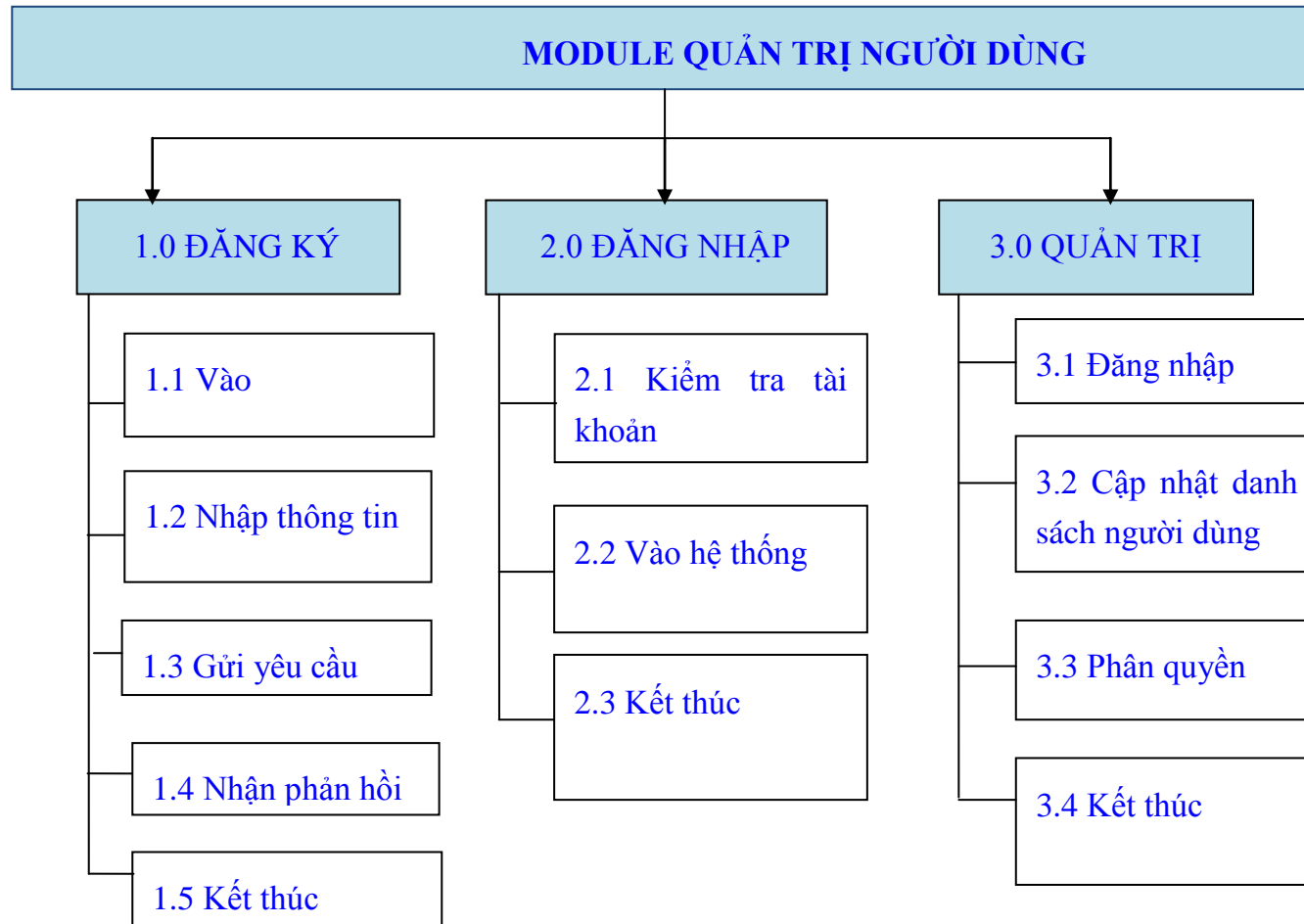
Lập bảng phân tích

Động từ + Bổ ngữ
<ul style="list-style-type: none">- Vào- Đăng ký- Đăng nhập- Nhập thông tin- Gửi yêu cầu- Đợi phản hồi- Kiểm tra tài khoản- Lịch sử- Cập nhật thông tin- Nhận email thông báo- Tương tác hệ thống- Phân quyền cho người dùng- Cập nhật danh sách người dùng

Lập bảng nhóm các thành phần

Động từ + Bổ ngữ	Chức năng hệ thống
<ul style="list-style-type: none">- Vào hệ thống- Nhập thông tin- Gửi yêu cầu- Nhận phản hồi	Đăng ký
<ul style="list-style-type: none">- Kiểm tra tài khoản- Tương tác hệ thống- Lịch sử	Đăng nhập
<ul style="list-style-type: none">- Cập nhật danh sách người dùng- Phân quyền cho người dùng	Quản trị

Vẽ biểu đồ phân rã chức năng



Hình 2. 2: Biểu đồ phân rã chức năng *Module quản trị người dùng*

f) Liệt kê hồ sơ

Các hồ sơ
a: Danh sách người dùng c: Danh sách quyền e: Danh sách lịch sử

g) Vẽ ma trận

Các thực thể chức năng			
a Danh sách người dùng			
b Danh sách quyền			
c Danh sách lịch sử			
Các chức năng nhiệm vụ			
	a	b	c
1. Đăng ký	U	C	
2. Đăng nhập	U	R	C
3. Quản trị	U	C	R

Hình 2. 3: Ma trận thực thể chức năng

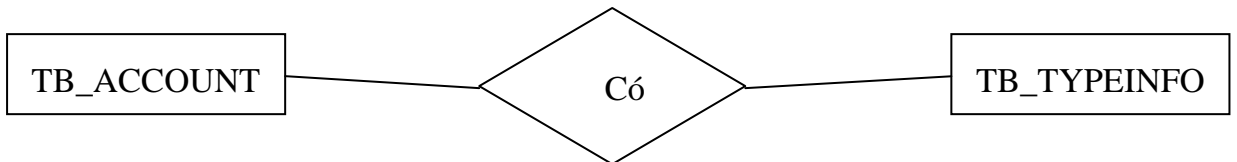
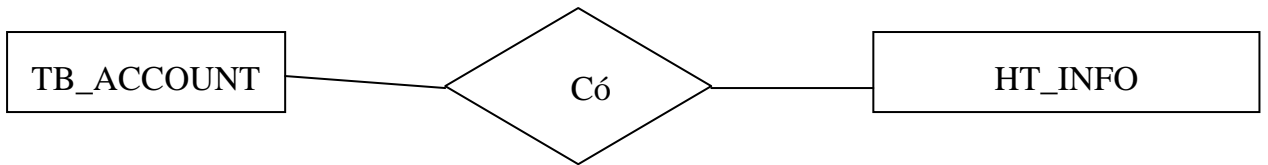
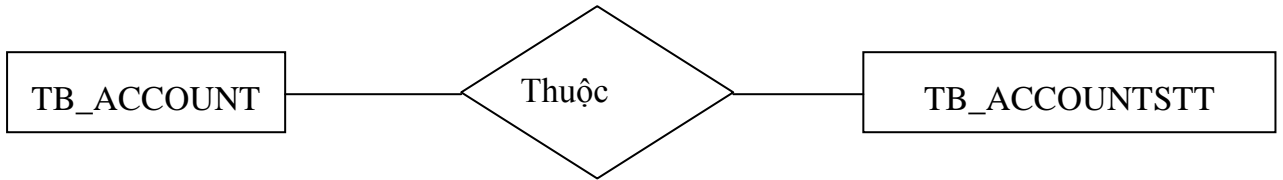
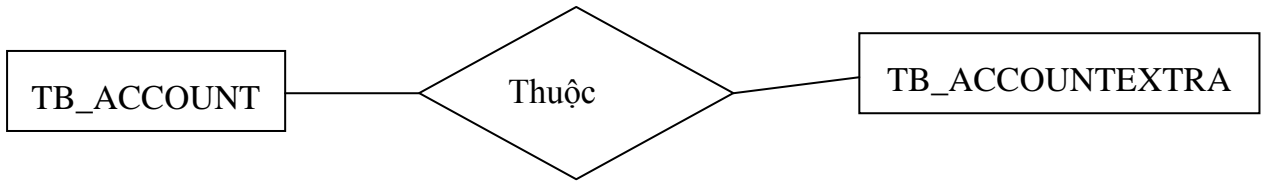
h) Thiết kế mô hình thực thể ER :

- Xác định kiểu thực thể, các thuộc tính và các thuộc tính khóa của thực thể

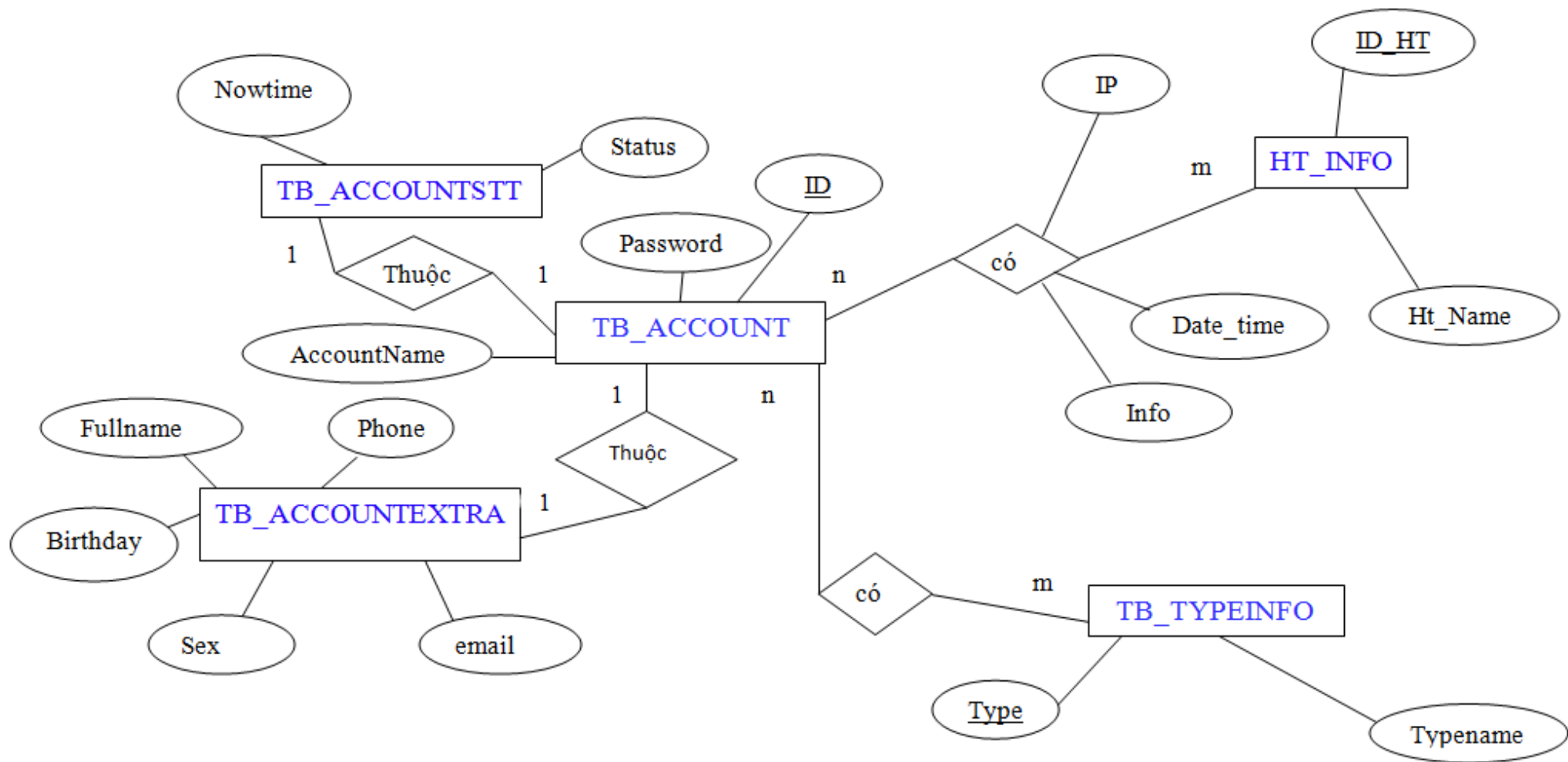
STT	Kiểu thực thể	Thuộc tính	Thuộc tính khóa
1	Account	ID, AccountName, Fullname, Birthday, Sex, Email, Phone	ID

2	TYPEINFO	Type, Typename	Type
3	HT_INFO	ID_HT, HT_Name	ID_HT

- Xác định các kiểu liên kết



- Mô hình ER :



Hình 2. 4: Mô hình ER

i) Thiết kế cơ sở dữ liệu:

Thiết kế các bảng dữ liệu:

Để thiết kế các bảng dữ liệu dưới đây ta dùng hệ quản trị cơ sở dữ liệu MySQL

- Bảng Account

Là bảng lưu thông tin các Account, thông tin lưu trữ bao gồm:

Tên Trường	Kiểu Dữ Liệu	Độ dài	Ghi Chú
<u>ID</u>	INT	11	Mã Tài Khoản
AccountName	VARCHAR	255	Tài Khoản
Password	VARCHAR	40	Mật Khẩu

- Bảng Accountextra:

Là bảng lưu thông tin của tài khoản, thông tin lưu trữ gồm:

Tên Trường	Kiểu Dữ Liệu	Độ dài	Ghi Chú
<u>ID</u>	INT	11	Mã Tài Khoản
FullName	VARCHAR	255	Tên đầy đủ
Birthday	DATE		Sinh nhật
Sex	INT	11	Giới tính

Email	TEXT		Địa chỉ mail
Phone	TEXT		Số điện thoại

- **Bảng Accountstt:**

Là bảng lưu nowtime và trạng thái, thông tin lưu trữ gồm:

Tên Trường	Kiểu Dữ Liệu	Độ dài	Ghi Chú
<u>ID</u>	INT	11	Mã Tài Khoản
Nowtime	INT	11	Nowtime
Status	INT	11	Trạng thái

- **Bảng Accounttype :**

Là bảng lưu thông tin id cấp quyền, thông tin lưu trữ gồm:

Tên Trường	Kiểu Dữ Liệu	Độ dài	Ghi Chú
ID	INT	11	Mã tài khoản
Type	INT	11	ID quyền

- **Bảng typeinfoe:**

Là bảng lưu thông tin quyền, thông tin lưu trữ gồm:

Tên Trường	Kiểu Dữ Liệu	Độ dài	Ghi Chú
Type	INT	11	ID quyền
TypeName	VARCHAR	255	Tên quyền

- **Bảng history info:**

Là bảng lưu thông tin các loại lịch sử, thông tin gồm:

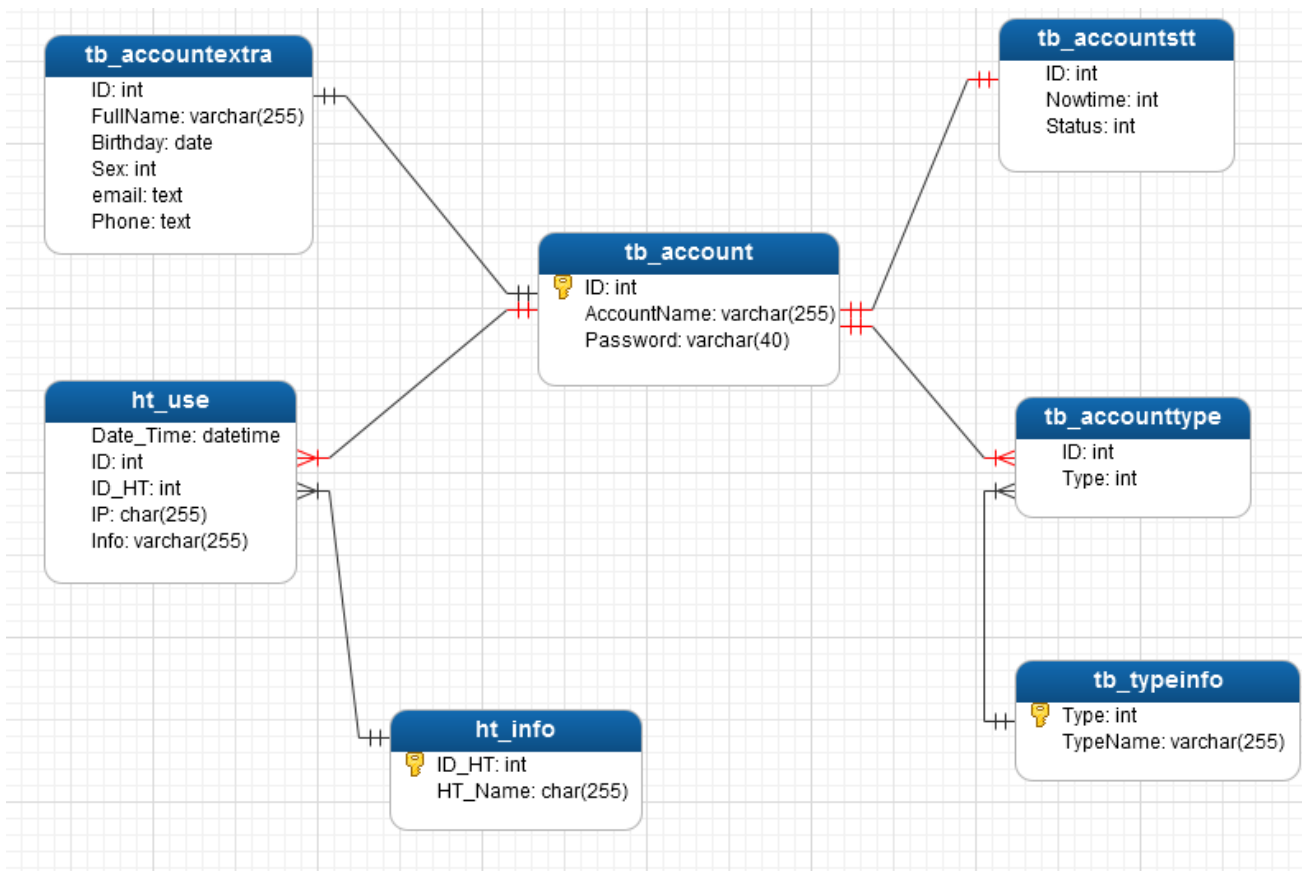
Tên Trường	Kiểu Dữ Liệu	Độ dài	Ghi Chú
<u>ID_History</u>	INT	11	Mã lịch sử
History_Name	Char	255	Tên loại lịch sử

- **Bảng history user:**

Là bảng lưu thông tin chi tiết lịch sử, thông tin gồm:

Tên Trường	Kiểu Dữ Liệu	Độ dài	Ghi Chú
Date_time	DATETIME		Ngày tháng
ID	INT	11	Mã tài khoản
ID_history	INT	11	Mã lịch sử
IP	CHAR	255	IP của người dùng
Info	VARCHAR	255	Chi tiết loại lịch sử

j) Mô hình quan hệ:



Hình 2. 5: Mô hình quan hệ

2.2 THIẾT KẾ GIAO DIỆN

- Trang đăng nhập:

Đăng Nhập Hệ Thống.

AccountName

Password

[Quên Mật Khẩu](#) . [Đăng ký](#)

Hình 2. 6: Giao diện đăng nhập hệ thống

- Đăng ký:

Đăng Ký Tài Khoản

Tài khoản :

Mật khẩu :

Nhập lại Mật khẩu :

Email:

Mã an toàn **9ab93**

[Trò lại](#)

Hình 2. 7: Giao diện đăng ký tài khoản

- Đổi mật khẩu :

[Trang chủ](#) | [Thông tin cá nhân](#) | [Quản lý thành viên](#)

Đổi Mật Khẩu.

Mật khẩu :	<input type="text"/>
Mật khẩu mới :	<input type="text"/>
Nhập lại mật khẩu mới:	<input type="text"/>

Hình 2. 8: Giao diện đổi mật khẩu

- Quên mật khẩu:

Thay đổi mật khẩu mới

Tên đăng nhập

Mật khẩu

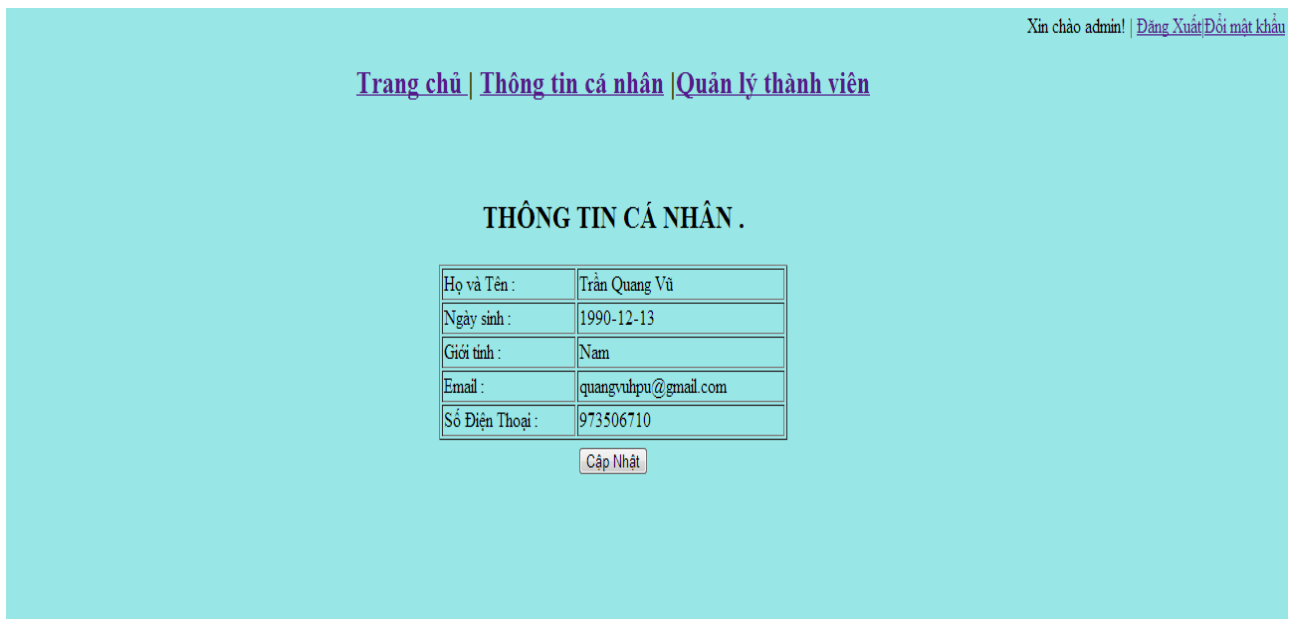
Nhập lại mật khẩu

Email

[Trở lại](#)

Hình 2. 9: Giao diện quên mật khẩu

- Thông tin cá nhân:



Hình 2. 10: Giao diện thông tin cá nhân

- Quản lý thông tin thành viên:



Hình 2. 11: Giao diện quản lý thông tin thành viên

2.3 MỘT SỐ KỸ THUẬT LẬP TRÌNH.

2.3.1 Kỹ thuật bảo mật tài khoản.

Khi người dùng đăng ký tài khoản vào hệ thống thì “Mật khẩu” sẽ được mã hóa bằng kỹ thuật mã hóa *md5*. Và sử dụng mã *captcha* để tránh người dùng ảo.

Ví dụ:

```
$pass=md5($pass); // mã hóa mật khẩu bằng md5.  
  
// Insert tài khoản vào hệ thống với pass được mã hóa  
  
$str="insert into tb_account value('{ $maxID}','{ $user}','{ $pass}')";
```

2.3.2 Kỹ thuật gửi mail.

Khi người dùng sử dụng chức năng quên mật khẩu thì hệ thống tự động gửi về mail đăng ký của tài khoản một mã xác nhận với việc sử dụng sự hỗ trợ của *phpmailer* và *smtp*. Người dùng sử dụng mã xác nhận này để thay đổi mật khẩu tài khoản của mình

Ví dụ . Code gửi mail khi quên tài khoản

```
$s = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789';  
  
mt_srand ((double)microtime() * 1000000);  
  
$sscode= "";  
  
for ($i=0;$i<30;$i++)  
  
$sscode .= substr($s, (mt_rand()%(strlen($s))), 1);  
  
$_SESSION['sec_code_cpass']=$sscode;  
  
$mail = new PHPMailer(); // create a new object  
  
$mail->IsSMTP(); // enable SMTP  
  
$mail->SMTPDebug = 0; // debugging:1 = errors and messages, 2 = messages only
```

```

$mail->SMTPAuth = true; // authentication enabled

$mail->SMTPSecure = 'ssl'; // secure transfer enabled REQUIRED for GMail

$mail->Host = 'smtp.gmail.com';

$mail->Port = 465;

$mail->Username = "quangvuct1201@gmail.com"; // tai khoan dang nhap de gui
email

$mail->Password = "vuct1201"; // mat khau gui email

$mail->From = "quangvuct1201@gmail.com"; // email se duoc thay the email trong
smtp

$mail->AddReplyTo("quangvuct1201@gmail.com"); // email cho phep nguai dung
reply lai

$mail->FromName = "Administrator"; // ho ten nguai gui email

$mail->WordWrap = 50;

$mail->IsHTML('text/html'); //text/html | text/plain, default:text/html

$mail->AltBody = "usermanager"; //Text Body

$mail->Body = "Your security code : ".$sscode." "; //HTML Body

$mail->Subject = "Security code";

$mail->AddAddress($email); // email nguai nhan

$mail->Send();

$mail->ClearAddresses();

```

2.3.3 Kỹ thuật khóa tài khoản.

Khi người quản trị muốn dừng hoạt động của tài khoản nào thì sẽ sử dụng chức năng *Delete* trong phần quản lý thông tin người dùng. Người quản trị có thể khôi phục tài khoản bằng cách kích hoạt chế độ phục hồi trong phần quản lý người dùng. Nếu muốn xóa hẳn tài khoản thì người quản trị sẽ dùng chức năng *Clear* để xóa hẳn tài khoản đó khỏi hệ thống.

Người quản trị muốn khóa không cho tài khoản đăng nhập vào hệ thống thì có thể gán cho tài khoản đó ở chế độ “Lose” (“Type”=99 trong cơ sở dữ liệu tại bảng “tb_typeinfo”).

2.3.4 Kỹ thuật phân quyền.

Khi một tài khoản đăng ký vào hệ thống sẽ mặc định tài khoản đó được phép đăng nhập vào hệ thống. Người quản trị muốn cho người dùng đó được phép thực hiện chức năng gì trong hệ thống thì sẽ phân quyền cho tài khoản đó. Khi người dùng đăng nhập vào hệ thống thì người dùng sẽ được sử dụng các chức năng được phân quyền đó

- Phân quyền:

```
// Lấy toàn bộ danh sách các quyền trong csdl
```

```
$sql = "SELECT * from từ bảng chứa các quyền";
```

```
$result = mysql_query($sql, $conn);
```

```
// Phân quyền cho tài khoản nào thì tích vào những quyền muốn phân cho  
tài khoản đó sử dụng các nút checkbox
```

```
<input type="checkbox" name="<?php echo $row[0];?>" id="<?php echo  
$row[0];?>" value="<?php echo $row[0];?>"
```

CHƯƠNG 3: THỬ NGHIỆM CHƯƠNG TRÌNH

3.1 XÂY DỰNG HỆ THỐNG

Qua quá trình phân tích thiết kế ở chương 2, trong chương này em đã thử nghiệm chương trình trên hệ thống như sau:

3.1.1 Phần cứng

- Intel(R) Core(TM) i5-3210M CPU @2.50GHz (4 CPUs), ~2.5GHz.
- Mainboard: GIGABYTE
- HDD: Samsung 500 GB
- RAM: 4GB

3.1.2 Phần mềm

- Hệ điều hành Windows 7 Ultimate. Service Pack 1
- XAMPP 3.1.0 gồm PHP, MySQL, Apache,...

3.2 MỘT SỐ GIAO DIỆN VÀ KẾT QUẢ KHI THỬ NGHIỆM HỆ THỐNG

- a) Giao diện thử nghiệm
- **Đăng ký tài khoản thành công cho người dùng**



The image shows a web form titled "Đăng Ký Tài Khoản" (Register Account) with a red header. Below the title, it says "Đăng ký tài khoản thành công !" (Account registration successful!). The form contains five input fields: "Tài khoản :" (Username), "Mật khẩu :" (Password), "Nhập lại Mật khẩu :" (Repeat Password), "Email:", and "Mã an toàn" (Security Code). The "Mã an toàn" field is filled with the number "49541". At the bottom, there are two buttons: "Đồng ý" (Agree) and "Đăng nhập" (Login).

Hình 3. 1: Đăng ký tài khoản thành công

- **Trang thông tin cá nhân sau khi cập nhật**

THÔNG TIN CÁ NHÂN .

Họ và Tên :	Trần Quang Vũ
Ngày sinh :	1990-12-13
Giới tính :	Nam
Email :	quangvuhpu@gmail.com
Số Điện Thoại :	973506710

Hình 3. 2: Trang cá nhân khi đã cập nhật thông tin

- **Trang quản lý thông tin người dùng của người quản trị (admin)**

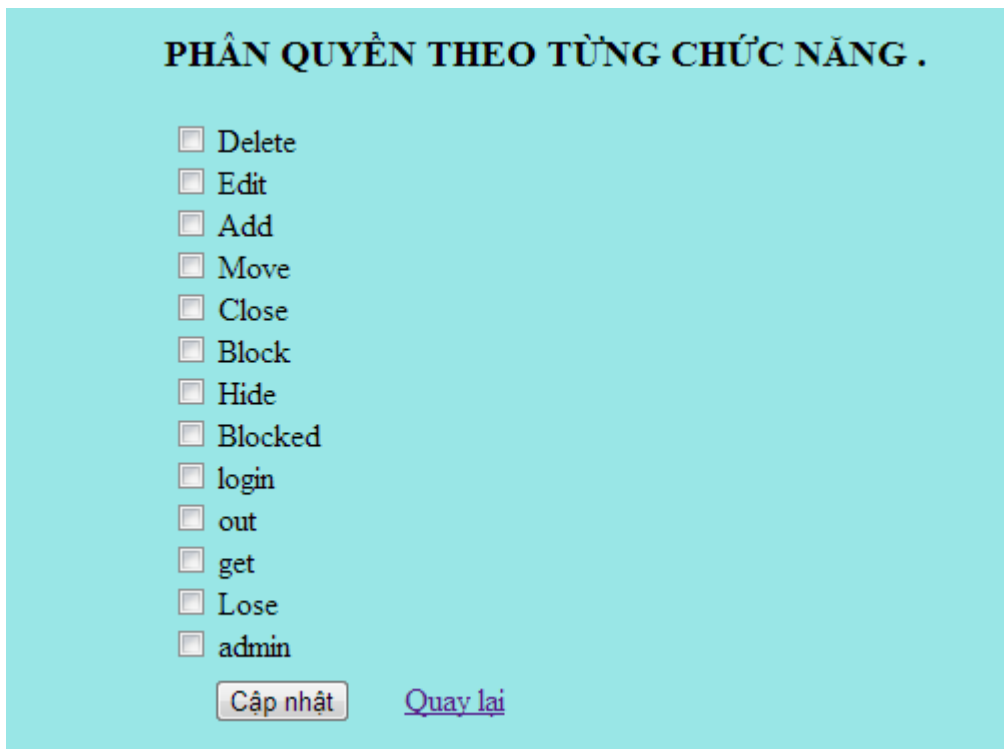
[Trang chủ](#) | [Thông tin cá nhân](#) | [Quản lý thành viên](#)

KIỂM TRA THÔNG TIN THÀNH VIÊN

ID	Tài Khoản	Họ và tên	Ngày Sinh	Giới tính	Địa chỉ Email	Số Điện thoại	Trạng thái					
4	vananh	Tran Van Son	1963-03-06	Nam	son@gmail.com	2234342545	Offline	Deleted	<input type="button" value="Edit"/>	<input type="button" value="Restore"/>	<input type="button" value="Del"/>	<input type="button" value="Clear"/>
8	vananh1	nguyen van anh	1998-03-03	Nam	vananh1@gmail.com	12144234234	Offline		<input type="button" value="Edit"/>		<input type="button" value="Del"/>	<input type="button" value="Clear"/>
14	vanem		0000-00-00	Nữ	vanem@gmail.com		Offline		<input type="button" value="Edit"/>		<input type="button" value="Del"/>	<input type="button" value="Clear"/>
15	vanem1		0000-00-00	Nữ	vanem1@gmail.com		Offline		<input type="button" value="Edit"/>		<input type="button" value="Del"/>	<input type="button" value="Clear"/>
16	vananh12		0000-00-00	Nữ	abc@gmail.com		Offline		<input type="button" value="Edit"/>		<input type="button" value="Del"/>	<input type="button" value="Clear"/>

Hình 3.3: Trang quản lý người dùng của Admin

- **Trang phân quyền cho người dùng**



Hình 3. 4: Trang phân quyền cho người dùng

b) Kết quả của thử nghiệm

Người dùng đăng ký, đăng nhập vào hệ thống, cập nhật thông tin, đổi mật khẩu, sử dụng các chức năng được cấp quyền một cách dễ dàng trong hệ thống.

Người quản trị viên dễ dàng quản lý được thông tin của các tài khoản khi tham gia vào hệ thống, dễ dàng cấp và hủy quyền của người dùng, block hoặc xóa hoàn toàn các tài khoản không cho phép sử dụng.....

KẾT LUẬN

Đồ án “*Tìm hiểu và xây dựng module quản trị người dùng*” sẽ giúp cho người quản trị có thể quản lý được hệ thống người dùng hệ thống của mình.

Qua quá trình thực hiện đồ án, em đã tổng hợp lại được các kiến thức trong thời gian đã học tại Trường. Đồng thời, em đã tìm hiểu và nắm được quy trình xây dựng hệ thống thông tin quản lý, nâng cao được kỹ năng lập trình, có thể chủ động trong việc tìm kiếm các thông tin để phục vụ công việc.

Hướng phát triển tiếp theo của đồ án là xây dựng và kết hợp modul quản trị người dùng với các trang web, hệ thống quản lý giúp cho người quản trị dễ dàng trong việc quản lý người dùng trong hệ thống của mình.

TÀI LIỆU THAM KHẢO

1. *HTML and CSS design and build Website* by Jon Duckett.
2. *Learning PHP, MySQL, JavaScript, and CSS: A Step by Step Guide to Creating Dynamic Websites* by Robin Nixon.
3. *Phân tích và thiết kế hệ thống thông tin quản lý* –Nguyễn Văn Vy.
4. *PHP and MySQL Web Development* by Luke Welling and Laura Thomson.
5. *CSS and Documents* by Eric A.Meyer

