

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THỰC PHẨM TP. HCM
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO NGHIỆM THU ĐỀ TÀI NCKH CẤP
TRƯỜNG

ĐỀ TÀI:
XÂY DỰNG ỨNG DỤNG TRAO ĐỔI DỮ LIỆU CỤ HỌC TẬP
CHO SINH VIÊN TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THỰC
PHẨM THÀNH PHỐ HỒ CHÍ MINH

Giảng viên hướng dẫn:

ThS. Trần Thị Bích Vân

Thực hiện đề tài:

Sinh viên: Lê Việt Hào

THÀNH PHỐ HỒ CHÍ MINH - 2018

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THỰC PHẨM TP. HCM
KHOA CÔNG NGHỆ THÔNG TIN



**BÁO CÁO NGHIỆM THU ĐỀ TÀI NCKH CẤP
TRƯỜNG**

ĐỀ TÀI:
**XÂY DỰNG ỨNG DỤNG TRAO ĐỔI DỮ LIỆU CỤ HỌC TẬP
CHO SINH VIÊN TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP
THỰC PHẨM THÀNH PHỐ HỒ CHÍ MINH**

Giảng viên hướng dẫn:

ThS. Trần Thị Bích Vân

Thực hiện đề tài:

Sinh viên: Lê Việt Hào

THÀNH PHỐ HỒ CHÍ MINH - 2018

LỜI CẢM ƠN

Lời đầu tiên em xin gửi tới Quý Thầy Cô Khoa Công nghệ Thông tin trường Đại học Công Nghiệp Thực Phẩm Thành Phố Hồ Chí Minh lời cảm ơn chân thành nhất. Em xin trân trọng gửi lời cảm ơn đến cô Trần Thị Bích Vân đã tận tình hướng dẫn, chỉ bảo, giúp đỡ em trong suốt quá trình làm đề tài.

Em cũng xin gửi lời cảm ơn chân thành đến quý Thầy Cô giáo trường Đại Học Công Nghiệp Thực Phẩm Thành Phố Hồ Chí Minh nói chung, quý Thầy Cô trong chuyên ngành công nghệ phần mềm nói riêng, đã trang bị cho em kiến thức về các môn đại cương cũng như các môn chuyên ngành, giúp em có được cơ sở lý thuyết vững vàng và tạo điều kiện giúp đỡ em trong suốt quá trình học tập.

Với tất cả nỗ lực của bản thân, em đã rất cố gắng để hoàn thành đề tài nhưng chắc chắn không tránh khỏi những sai sót. Kính mong sự góp ý, chỉ bảo chân thành của quý Thầy Cô và các bạn để đề tài phát triển tốt hơn.

Một lần nữa em chân thành cảm ơn quý Thầy Cô và xin chúc quý Thầy Cô luôn được nhiều sức khỏe, thành công và may mắn trong công việc cũng như trong cuộc sống.

Một lần nữa em xin chân thành cảm ơn!

Sinh viên Lê Việt Hào

MỤC LỤC

LỜI CẢM ƠN.....	i
MỤC LỤC	ii
CÁC KÝ HIỆU VÀ CHỮ VIẾT TẮT	v
DANH SÁCH BẢNG.....	vi
DANH SÁCH HÌNH.....	vii
MỞ ĐẦU	1
CHƯƠNG 1	2
TỔNG QUAN.....	2
1.1. ĐẶT VẤN ĐỀ	2
1.2. MỤC TIÊU NGHIÊN CỨU CỦA ĐỀ TÀI	2
1.3. PHẠM VI VÀ NỘI DUNG NGHIÊN CỨU	2
1.3.1. Phần nghiên cứu cơ sở lý thuyết	2
1.3.2. Phần thực nghiệm.....	3
1.4. KẾT QUẢ ĐẠT ĐƯỢC	3
1.5. KẾT CHƯƠng.....	3
CHƯƠNG 2.....	4
TÌM HIỂU CÔNG NGHỆ LẬP TRÌNH WEB VỚI NODE.JS, MONGODB VÀ VUE.JS..	4
2.1. GIỚI THIỆU	4
2.2. TÌM HIỂU CÔNG NGHỆ NODE.JS	4
2.2.1. Node.js là gì?.....	4
2.2.2. Tại sao sử dụng Node.js?	4
2.2.3. Đánh giá ưu nhược điểm của Node.js	5
2.3. TÌM HIỂU HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU MONGODB	5
2.3.1. MongoDB là gì?.....	5
2.3.2. Tại sao sử dụng MongoDB?.....	6
2.3.3. Đánh giá ưu nhược điểm của MongoDB	7
2.4. TÌM HIỂU THƯ VIỆN VUE.JS.....	9
2.4.1. Vue.js là gì?.....	9
2.4.2. Đánh giá ưu nhược điểm của Vue.js	10
2.5. KẾT CHƯƠng.....	11
CHƯƠNG 3	12
KHẢO SÁT HỆ THỐNG.....	12
3.1. ĐÁNH GIÁ CHUNG HIỆN TRẠNG TRAO ĐỔI DỤNG CỤ HỌC TẬP	12

3.2.	MÔ TẢ NGHIỆP VỤ	12
3.3.	MÔ HÌNH BPM HIỆN TRẠNG TRAO ĐỔI DỤNG CỤ HỌC TẬP	13
3.4.	KẾT CHƯƠNG.....	14
CHƯƠNG 4.....		14
PHÂN TÍCH HỆ THỐNG.....		15
4.1.	GIỚI THIỆU	15
4.2.	PHÂN TÍCH XỬ LÝ	15
4.2.1.	Giới thiệu mô hình phân cấp chức năng (Business Function Diagram - BFD)	15
4.2.2.	Mô hình phân cấp chức năng (BFD) của hệ thống	15
4.2.3.	Giới thiệu mô hình DFD (Data Flow Diagram).....	16
4.2.4.	Mô hình DFD của hệ thống.....	16
4.3.	PHÂN TÍCH DỮ LIỆU	17
4.3.1.	Khái quát mô hình cơ sở dữ liệu quan niệm (CDM).....	17
4.3.2.	Mô hình CDM cho hệ thống	18
4.3.3.	Bảng sơ đồ dữ liệu.....	20
4.4.	KẾT CHƯƠNG.....	23
CHƯƠNG 5.....		23
THIẾT KẾ HỆ THỐNG		24
5.1.	GIỚI THIỆU	24
5.2.	MÔ HÌNH QUAN HỆ	24
5.2.1.	Giới thiệu mô hình quan hệ.....	24
5.2.2.	Lược đồ cơ sở dữ liệu quan hệ	25
5.3.	RÀNG BUỘC TOÀN VỆN	26
5.3.1.	Ràng buộc tham chiếu.....	26
5.3.2.	Ràng buộc liên bộ.....	26
5.3.3.	Ràng buộc liên thuộc tính – liên quan hệ	26
5.4.	THIẾT KẾ GIAO DIỆN	27
5.4.1.	Giao diện đăng nhập.....	27
5.4.2.	Giao diện trang chủ	28
5.4.3.	Giao diện trang thông tin cá nhân	28
5.4.4.	Giao diện trang quản trị.....	29
5.5.	KẾT CHƯƠNG.....	29
KẾT LUẬN ĐỀ XUẤT VÀ HƯỚNG PHÁT TRIỂN		30
Những hạn chế khó khăn		30
Hướng phát triển		30
TÀI LIỆU THAM KHẢO		31

NHẬN XÉT CỦA GIÁO VIÊN	32
------------------------------	----

CÁC KÝ HIỆU VÀ CHỮ VIẾT TẮT

Ký hiệu	Diễn giải
VPS	Viết tắt của Virtual Private Server (Máy chủ riêng ảo). Một VPS cũng giống như shared host, tức là có nhiều VPS được đặt trên cùng một máy chủ vật lý, phục vụ nhu cầu của các website lớn.
Dedicated server	Máy chủ dùng riêng độc lập dành cho những website lớn.
Schema	Schema là khái niệm mới có trong Sql server 2005 trở lên. Schema mặc định của các object trong database là dbo. Có thể hiểu schema là một namespace đối với các đối tượng database
JSON	JSON là một kiểu định dạng dữ liệu trong đó sử dụng văn bản thuần túy, định dạng JSON sử dụng các cặp <i>key - value</i> để dữ liệu sử dụng.
RDMS	RDBMS là viết tắt của Relational Database Management System có nghĩa là hệ quản trị cơ sở dữ liệu quan hệ.
Cluster	Là một nhóm các thành phần hoạt động cùng nhau để cung cấp một dịch vụ nào đó.
Replica set	Replica set là một hệ replication trong MongoDB. Tập data sẽ được nhân bản trên nhiều server thay vì tập trung trên một single server. Nhờ vậy, replica set cung cấp tính năng high availability và dự phòng. Nó cũng scale read request cho mongodb. Mô hình của replica set trong mongodb gần giống replication trong mysql.
NoSQL	Thuật ngữ NoSQL được giới thiệu lần đầu vào năm 1998 sử dụng làm tên gọi chung cho các lightweight open source relational database (cơ sở dữ liệu quan hệ nguồn mở nhỏ) nhưng không sử dụng SQL cho truy vấn.
SQL	SQL là viết tắt của Structured Query Language có nghĩa là ngôn ngữ truy vấn có cấu trúc, là một ngôn ngữ máy tính để lưu trữ, thao tác và truy xuất dữ liệu được lưu trữ trong một cơ sở dữ liệu quan hệ.
CRUD	Là khái niệm viết tắt của "Create, Read, Update, Delete"

DANH SÁCH BẢNG

<i>Bảng 2.1 So sánh giữa SQL DB schema và Mongo DB schema</i>	<i>6</i>
<i>Bảng 4.1Bảng User.....</i>	<i>20</i>
<i>Bảng 4.2 Bảng Post</i>	<i>20</i>
<i>Bảng 4.3 Bảng PostCategory</i>	<i>20</i>
<i>Bảng 4.4 Bảng PostFileName</i>	<i>20</i>
<i>Bảng 4.5 Bảng Comment</i>	<i>21</i>
<i>Bảng 4.6 Bảng Reply</i>	<i>21</i>
<i>Bảng 4.7 Bảng Notify.....</i>	<i>21</i>
<i>Bảng 4.8 Bảng NotifyStatus.....</i>	<i>22</i>
<i>Bảng 4.9 Bảng Gender.....</i>	<i>22</i>
<i>Bảng 4.10 Bảng Exchange.....</i>	<i>22</i>
<i>Bảng 4.11 Bảng Event</i>	<i>22</i>
<i>Bảng 4.12 Bảng Search</i>	<i>23</i>
<i>Bảng 4.13 Bảng ToolCategory</i>	<i>23</i>
<i>Bảng 4.14 Bảng Right.....</i>	<i>23</i>
<i>Bảng 5.1 Ràng buộc tham chiếu.....</i>	<i>26</i>
<i>Bảng 5.2 Ràng buộc liên bộ.....</i>	<i>26</i>
<i>Bảng 5.3 Ràng buộc liên thuộc tính – liên quan hệ</i>	<i>26</i>

DANH SÁCH HÌNH

<i>Hình 2-1 Cấu trúc một document trong MongoDB</i>	<i>6</i>
<i>Hình 2-2 Hiệu năng của MongoDB.....</i>	<i>7</i>
<i>Hình 2-3 Khả năng mở rộng của MongoDB</i>	<i>8</i>
<i>Hình 2-4 Bảng benchmark các framework Javascript nổi tiếng nhất hiện nay</i>	<i>9</i>
<i>Hình 2-5 Mô hình MVVM trong Vue.js</i>	<i>10</i>
<i>Hình 3-1 Mô hình BPM hiện trạng trao đổi dụng cụ học tập</i>	<i>13</i>
<i>Hình 4-1 Mô hình phân cấp chức năng (BFD) của hệ thống</i>	<i>15</i>
<i>Hình 4-2 Mô hình DFD mức 0</i>	<i>16</i>
<i>Hình 4-3 Mô hình DFD phân rã mức 1</i>	<i>17</i>
<i>Hình 4-4 Mô hình CDM cho hệ thống.....</i>	<i>19</i>
<i>Hình 5-1 Lược đồ cơ sở dữ liệu quan hệ.....</i>	<i>25</i>
<i>Hình 5-2 Giao diện đăng nhập ứng dụng.....</i>	<i>27</i>
<i>Hình 5-3 Giao diện trang chủ.....</i>	<i>28</i>
<i>Hình 5-4 Giao diện trang thông tin cá nhân</i>	<i>28</i>
<i>Hình 5-5 Giao diện trang quản trị</i>	<i>29</i>

MỞ ĐẦU

Ngày nay, Công nghệ Thông tin đã và đang phát triển mạnh mẽ, việc ứng dụng công nghệ thông tin vào nhiều lĩnh vực khác nhau của xã hội là việc không thể thiếu. Trong một đơn vị trường học, việc trao đổi dụng cụ học tập là nhu cầu cần thiết. Trên thực tế việc trao đổi dụng cụ học tập giữa các sinh viên hiện nay chỉ diễn ra trong môi trường những người bạn với nhau hoặc thông qua giới thiệu, trên mạng xã hội, ... nguồn trao đổi vẫn còn hạn chế về không gian, thời gian, người có nhu cầu khó gặp nhau, hoặc nguồn tài liệu quá dồi dào, không tập trung,... Việc triển khai một ứng dụng hỗ trợ sinh viên trao đổi dụng cụ học tập nhằm tiết kiệm thời gian và nâng cao hiệu quả trong việc tìm kiếm nguồn tài liệu học tập là nhu cầu cần thiết.

Đồ án dưới đây sẽ phân tích đầy đủ, chi tiết các bước cần thiết, kế hoạch thực hiện, hướng phát triển của đề tài nghiên cứu.

Ứng dụng này tạo ra một môi trường, một cộng đồng giúp sinh viên có một nơi trao đổi dụng cụ học tập rộng mở hơn, tập trung hơn, tăng lượng tương tác cho nhu cầu trao đổi dụng cụ học tập.

Để xây dựng một ứng dụng trao đổi dụng cụ học tập một cách thiết thực và gần gũi, chúng ta phải đi khảo sát từ thực tế, và các phần mềm mẫu. Từ kết quả khảo sát sẽ bắt đầu phân tích nghiệp vụ, phân tích cơ sở dữ liệu và cách lưu trữ dữ liệu. Thông qua việc phân tích, sẽ đưa ra chọn lựa công cụ và giải pháp phù hợp để thực hiện đề tài. Sau đó sẽ thiết kế giao diện cho hoàn chỉnh. Tiếp theo code thực thi các chức năng đã phân tích. Cuối cùng sẽ kiểm tra lại toàn bộ quá trình xây dựng ứng dụng để tìm kiếm và khắc phục các lỗi để hoàn thiện ứng dụng trao đổi dụng cụ học tập, dùng thử và đánh giá kết quả đạt được. Xác định hướng phát triển của đề tài.

CHƯƠNG 1

TỔNG QUAN

1.1. ĐẶT VẤN ĐỀ

Trao đổi đồ dùng học tập là nhu cầu cần thiết diễn ra thường xuyên tại các trường học. Với sự phát triển của công nghệ thông tin hiện nay ngoài việc trao đổi trực tiếp, sinh viên có thể trao đổi qua mạng xã hội, các trang web,... Phần lớn sinh viên phải chủ động tìm kiếm tài liệu phù hợp với nhu cầu học tập của mình. Khó khăn hiện tại là các tài liệu trên mạng xã hội, trên các trang web rất phong phú nhưng nội dung đa dạng khó tiếp cận với sinh viên, trong khi tài liệu từ các sinh viên chia sẻ với nhau sẽ có xác suất hữu ích cao hơn, đáng tin cậy hơn vì có sự tương đồng trong cùng môi trường học tập nhưng lại khó khăn trong việc liên kết số lượng lớn các sinh viên có nhu cầu trao đổi phù hợp với nhau. Vấn đề đặt ra là phải áp dụng công nghệ thông tin để xây dựng ứng dụng, giúp sinh viên trao đổi dụng cụ học tập dễ dàng hơn, tạo môi trường hỗ trợ cho việc trao đổi dụng cụ học tập tốt hơn, thông tin tìm kiếm đáng tin cậy, xác suất kết quả tìm kiếm hữu ích cao và tiết kiệm được thời gian.

1.2. MỤC TIÊU NGHIÊN CỨU CỦA ĐỀ TÀI

Để xây dựng hệ thống web server phục vụ được lượng lớn kết nối cùng lúc, chúng em nghiên cứu sử dụng công nghệ Node.js cùng với các thư viện hỗ trợ đi kèm. Nghiên cứu sử dụng hệ quản trị cơ sở dữ liệu MongoDB, là một hệ quản trị cơ sở dữ liệu dạng NoSQL để làm giảm bớt đi các xử lý ràng buộc nặng nề như các hệ quản trị cơ sở dữ liệu SQL khác, tăng tốc độ xử lý dữ liệu cho hệ thống. Nghiên cứu thư viện lập trình web front end Vue.js để thiết kế website dạng single page application, giúp giảm số lần request đến server để lấy dữ liệu, dễ dàng quản lý code và bảo trì, nâng cao trải nghiệm người dùng. Khảo sát hiện trạng trao đổi dụng cụ học tập thực tế, phân tích và xây dựng ứng dụng trao đổi dụng cụ học tập hoàn thiện dựa trên các công nghệ đã nghiên cứu.

1.3. PHẠM VI VÀ NỘI DUNG NGHIÊN CỨU

Trong phạm vi đề tài **“Xây dựng ứng dụng trao đổi dụng cụ học tập cho sinh viên trường Đại học Công nghiệp Thực phẩm thành phố Hồ Chí Minh”** chúng em quan tâm đến các vấn đề sau: tìm hiểu công nghệ Node.js xây dựng hệ thống web server, tìm hiểu hệ quản trị cơ sở dữ liệu MongoDB để quản lý dữ liệu cho ứng dụng, tìm hiểu công nghệ Vue.js thiết kế single page application trên nền web, xây dựng hoàn chỉnh ứng dụng trao đổi dụng cụ học tập cho sinh viên.

1.3.1. Phần nghiên cứu cơ sở lý thuyết

Trong phạm vi đề tài, chúng em thực hiện tìm hiểu các vấn đề công nghệ Node.js, MongoDB, Vue.js. Sau đó chúng em sẽ đưa ra cách tổ chức, cấu trúc hệ thống ứng dụng thích hợp với các công nghệ đã nghiên cứu ở trên và đáp ứng được yêu cầu của ứng dụng.

1.3.2. Phần thực nghiệm

Thiết kế, tổ chức, cấu trúc và hệ thống ứng dụng. Tiến hành xây dựng ứng dụng hoàn thiện đầy đủ các tính năng cần thiết của một ứng dụng tìm kiếm, trao đổi dụng cụ học tập.

1.4. KẾT QUẢ ĐẠT ĐƯỢC

Ứng dụng được các công nghệ Node.js, MongoDB, Vue.js xây dựng ứng dụng trao đổi dụng cụ học tập cho sinh viên. Ứng dụng quản lý được thông tin sinh viên, các bài đăng tin trao đổi theo loại dụng cụ, bài đăng tìm kiếm dụng cụ học tập, giúp sinh viên bình luận trao đổi. Ngoài ra ứng dụng còn quản lý được các lượt trao đổi dụng cụ của sinh viên, các lượt trao đổi dụng cụ của một bài đăng. Ứng dụng sẽ thông báo đến sinh viên khi có một bài đăng mới phù hợp với nội dung trong bài đăng trao đổi của sinh viên.

1.5. KẾT CHƯƠng

Ở chương này chúng em đã nêu lên sự cần thiết để xây dựng một ứng dụng hỗ trợ việc trao đổi dụng cụ học tập dành cho sinh viên, từ đó đặt ra các mục tiêu nghiên cứu công nghệ, khảo sát phân tích từ thực tế. Xác định phạm vi, nội dung nghiên cứu của đề tài và kết quả cần phải đạt được sau khi hoàn thành đề tài. Chương sau sẽ tập trung vào phân nghiên cứu các công nghệ sử dụng trong đề tài.

CHƯƠNG 2

TÌM HIỂU CÔNG NGHỆ LẬP TRÌNH WEB VỚI NODE.JS, MONGODB VÀ VUE.JS

2.1. GIỚI THIỆU

Ở chương trước chúng ta đã có được cái nhìn tổng quan về đề tài nghiên cứu. Chương này sẽ giới thiệu các công nghệ đã tìm hiểu để phục vụ xây dựng ứng dụng trao đổi dụng cụ học tập cho sinh viên. Gồm ba công nghệ chính là: Node.js, MongoDB và Vue.js.

2.2. TÌM HIỂU CÔNG NGHỆ NODE.JS

2.2.1. Node.js là gì?

Node.js là một nền tảng chạy trên môi trường V8 JavaScript runtime – một trình thông dịch JavaScript cực nhanh chạy trên trình duyệt Chrome. Khác với hầu hết các chương trình JavaScript, Node.js không chạy trên một trình duyệt mà chạy trên server. JavaScript là một ngôn ngữ dựa trên sự kiện, vì vậy bất cứ thứ gì xảy ra trên server đều tạo ra một sự kiện non-blocking. Mỗi kết nối mới sinh ra một sự kiện; dữ liệu nhận được từ một upload form sinh ra một sự kiện data-received; việc truy vấn dữ liệu từ database cũng sinh ra một sự kiện. Trong thực tế, điều này có nghĩa là một trang web Node.js sẽ chẳng bao giờ bị khóa (lock up) và có thể hỗ trợ cho hàng chục nghìn user truy cập cùng lúc. Node.js đóng vai trò của server - Apache - và thông dịch mã ứng dụng chạy trên nó. Giống như Apache, có rất nhiều module (thư viện) có thể được cài đặt để bổ sung thêm các đặc trưng và chức năng - như lưu trữ dữ liệu, hỗ trợ file Zip, đăng nhập bằng Facebook, hoặc các cổng thanh toán. Dĩ nhiên, nó không có nhiều thư viện như PHP, nhưng Node.js vẫn đang ở trong giai đoạn ban đầu và có một cộng đồng rất mạnh mẽ ở đằng sau nó.

Một khái niệm cốt lõi của Node.js đó là các function bất đồng bộ (asynchronous functions). Với hầu hết các ngôn ngữ kịch bản máy chủ, chương trình phải đợi mỗi function thực thi xong trước khi có thể tiếp tục chạy tiếp. Với Node.js, có thể xác định các function sẽ chạy để hoàn thành một tác vụ nào đó, trong khi phần còn lại của ứng dụng vẫn chạy đồng thời, đó là một trong những đặc trưng tiêu biểu của Node.js.

2.2.2. Tại sao sử dụng Node.js?

Node.js có tốc độ thực thi nhanh và xử lý hàng ngàn kết nối đồng thời, thích hợp cho phát triển các ứng dụng kết nối nhiều người dùng.

Ngôn ngữ lập trình là JavaScript, là ngôn ngữ không thể thiếu cho hầu hết mọi web client, tiện lợi để sử dụng một ngôn ngữ cho cả server và web client.

2.2.3. Đánh giá ưu nhược điểm của Node.js

Ưu điểm:

Đầu tiên là ưu điểm về tốc độ thực thi và khả năng mở rộng. Node.js có tốc độ rất nhanh. Đó là một yêu cầu khá quan trọng để tạo ra một sản phẩm lớn và muốn đảm bảo có thể mở rộng nhanh chóng, đáp ứng được một lượng lớn người dùng khi trang web phát triển lên. Node.js có thể xử lý hàng ngàn kết nối đồng thời trong khi PHP thì không thể.

Nhược điểm:

Giống như hầu hết các công nghệ mới, việc triển khai Node.js trên host không phải là điều dễ dàng. Không thể đơn giản tải lên một ứng dụng Node.js và mong chờ nó hoạt động tốt trên một web hosting xài chung. VPS và dedicated server là một sự lựa chọn tốt hơn - có thể cài đặt Node.js trên chúng. Thậm chí dễ hơn là sử dụng một dịch vụ có khả năng mở rộng như là Heroku và có thể hoàn toàn an tâm để phát triển trang web trên đó - chỉ cần trả tiền khi cần thêm nhiều tài nguyên hơn. Một nhược điểm lớn khác của Node.js đó là nó vẫn đang trong giai đoạn phát triển ban đầu, điều này có nghĩa là một số đặc trưng sẽ thay đổi trong quá trình phát triển tiếp theo.

2.3. TÌM HIỂU HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU MONGODB

2.3.1. MongoDB là gì?

MongoDB là một trong những cơ sở dữ liệu mã nguồn mở NoSQL phổ biến nhất được biết bằng C++. Tính đến tháng 2/2015, MongoDB được xếp thứ 4 trong số các hệ thống cơ sở dữ liệu phổ biến nhất. Nó được phát triển bởi công ty 10gen sau này được biết đến với tên MongoDB Inc.

MongoDB là cơ sở dữ liệu hướng tài liệu, nó lưu trữ dữ liệu trong các document dạng JSON với schema động rất linh hoạt. Nghĩa là có thể lưu các bản ghi mà không cần lo lắng về cấu trúc dữ liệu như là số trường (field), kiểu của trường lưu trữ. Tài liệu MongoDB tương tự như các đối tượng JSON.

```

{
  name: "sue",
  age: 26,
  status: "A",
  groups: [ "news", "sports" ]
}

```

← field: value
 ← field: value
 ← field: value
 ← field: value

Hình 0-1 Cấu trúc một document trong MongoDB

Nền tảng và ngôn ngữ hỗ trợ:

MongoDB là một cơ sở dữ liệu NoSQL hỗ trợ đa nền tảng, nó có thể chạy trên Windows, Linux và Mac...Nó hỗ trợ hầu hết các ngôn ngữ lập trình phổ biến như C#, Java, PHP, Javascript...và các môi trường phát triển khác nhau.

So sánh giữa SQL DB schema và Mongo DB schema

Bảng 0.1 So sánh giữa SQL DB schema và Mongo DB schema

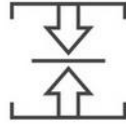
SQL DB	MongoDB
Table	Collection
Row	Document
Column	Field
Joins	Embedded document, linking
Primary key	Primary key (mặc định là _id do chính Mongo tạo)

2.3.2. Tại sao sử dụng MongoDB?

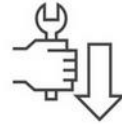
RDBMS lưu trữ dữ liệu trong các định dạng bảng và sử dụng ngôn ngữ truy vấn cấu trúc (SQL) để truy vấn cơ sở dữ liệu. RDBMS cũng định nghĩa sẵn các schema theo yêu cầu và tập hợp các quy tắc để định nghĩa quan hệ giữa các trường trong bảng.



7x-10x Better Performance



80% Less Storage



95% Reduction in Ops

Hình 0-2 Hiệu năng của MongoDB

Nhưng MongoDB lưu trữ dữ liệu trong các tài liệu thay vì các bảng. Có thể thay đổi cấu trúc các bản ghi (được gọi là document trong MongoDB) đơn giản chỉ cần thêm mới các trường hoặc xóa các trường có sẵn. Khả năng này của MongoDB giúp trình bày các quan hệ dạng thứ bậc, để lưu trữ mảng, và các cấu trúc phức tạp khác một cách dễ dàng. MongoDB cung cấp hiệu suất cao, tính sẵn có cao và dễ dàng mở rộng.

2.3.3. Đánh giá ưu nhược điểm của MongoDB

Ưu điểm:

Mã nguồn mở:

MongoDB là phần mềm mã nguồn mở miễn phí, có cộng đồng phát triển rất lớn

Hiệu năng cao:

Tốc độ truy vấn (find, update, insert, delete) của MongoDB nhanh hơn hẳn so với các hệ quản trị cơ sở dữ liệu quan hệ (RDBMS).

Thử nghiệm cho thấy tốc độ insert của MongoDB có thể nhanh tới gấp 100 lần so với MySQL (So sánh hiệu suất của MongoDB với MySQL).

Tại sao MongoDB có hiệu năng cao như thế? có các lý do sau:

MongoDB lưu dữ liệu dạng JSON, khi insert nhiều đối tượng thì nó sẽ là insert một mảng JSON gần như với trường hợp insert một đối tượng.

Dữ liệu trong MongoDB không có sự ràng buộc lẫn nhau như trong RDBMS, khi insert, xóa hay update nó không cần phải mất thời gian kiểm tra xem có thỏa mãn các bảng liên quan như trong RDBMS.

Dữ liệu trong MongoDB được đánh chỉ mục (đánh index) nên khi truy vấn nó sẽ tìm rất nhanh.

Khi thực hiện insert, find... MongoDB sẽ khóa các thao tác khác lại, ví dụ khi nó thực hiện find(), trong quá trình find mà có thêm thao tác insert, update thì nó sẽ dừng hết lại để chờ find() xong đã.

Dữ liệu linh hoạt:

MongoDB là document database, dữ liệu lưu dưới dạng JSON, không bị bó buộc về số lượng field, kiểu dữ liệu... có thể insert thoải mái dữ liệu mong muốn.

Là Rich Query Language:

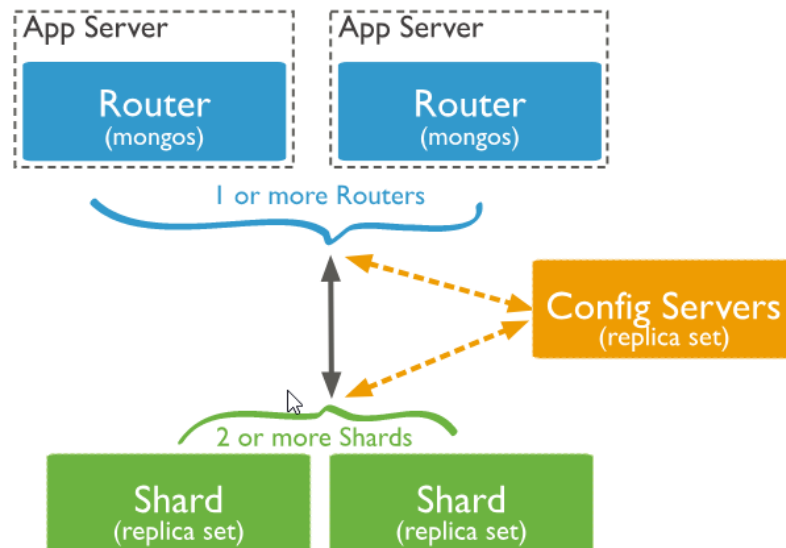
MongoDB là một rich query language tức là nó có sẵn các method để thực hiện create, read, update, delete dữ liệu (CRUD)

Tính sẵn có:

MongoDB hỗ trợ replica set nhằm đảm bảo việc sao lưu và khôi phục dữ liệu

Khả năng mở rộng Horizontal Scalability:

Trong MongoDB có một khái niệm cluster là cụm các node chứa dữ liệu giao tiếp với nhau, khi muốn mở rộng hệ thống ta chỉ cần thêm một node mới vào cluster.



Hình 0-3 Khả năng mở rộng của MongoDB

Nhược điểm:

MongoDB không có các tính chất ràng buộc như trong RDBMS nên dễ bị làm sai dữ liệu.

Không hỗ trợ join giống như RDBMS nên khi viết function join trong code ta phải làm bằng tay khiến cho tốc độ truy vấn bị giảm.

Sử dụng nhiều bộ nhớ: do dữ liệu lưu dưới dạng key-value, các collection chỉ khác về value do đó key sẽ bị lặp lại. Không hỗ trợ join nên sẽ bị dư thừa dữ liệu (trong RDBMS thì ta chỉ cần lưu 1 bản ghi rồi các bản ghi khác tham chiếu tới còn trong MongoDB thì không).

Bị giới hạn kích thước bản ghi: mỗi document không được có kích thước lớn hơn 16Mb và cấp độ lồng các document con trong 1 document không được lớn 100.

2.4. TÌM HIỂU THƯ VIỆN VUE.JS

2.4.1. Vue.js là gì?

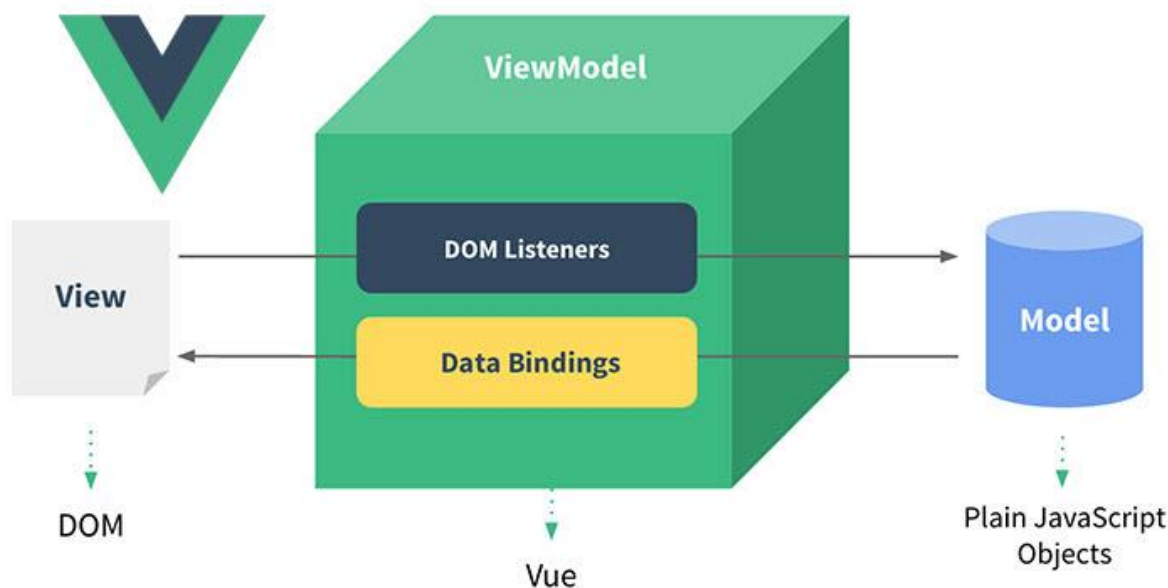
Vue.js là một framework Javascript tiên tiến trong xây dựng giao diện người dùng, không giống như các framework khác, Vue.js được xây dựng từ những dòng code cơ bản nhất nhằm tối ưu tốc độ. Thư viện của Vue.js chỉ tập trung vào lớp hiển thị, rất đơn giản để tiếp cận và dễ dàng tích hợp vào các hệ thống khác. Vue.js cũng có khả năng cung cấp các ứng dụng web đơn trang Single Page Application (toàn bộ website chỉ là một trang) cho phép kết hợp với nhiều các công cụ hiện đại, như Laravel.

Vue.js được sử dụng để xây dựng giao diện người dùng giống như React (sử dụng bởi Facebook), Angular (được xây dựng bởi Google), Ember... Tuy nhiên, Vue.js có tốc độ tạo trang (render) rất nhanh và chiếm khá ít bộ nhớ. Vue.js có một thứ hạng cao trong bảng benchmark các framework Javascript nổi tiếng nhất hiện nay.

	angular v16.1. keyed	angular v2.4.3. keyed	bobril v5.0.4	dommm v2.0.1. keyed	elm v0.18.0	ember v2.10.0. beta.3	ember v2.6.1	kniv v1.0.0. rc2	knockout v3.4.1	mithril v0.2.5	mithril v1.0.0. alpha	na v1.0.0. beta.1.1.0. keyed	plastiq v1.33.0	preact v7.1.0	reactive v0.8.9. keyed	react-ike v0.15.30	react v15.4.2. redux. v3.6.0	react v15.4.2. mobx. v3.6.0	react v15.4.2. redux. v3.6.0	vidom v0.7.1	vue v2.1.10. keyed	vanillajs- keyed
create rows Duration for creating 1000 rows after the page loaded.	254.72 ± 8.58 (1.96)	178.12 ± 3.25 (1.37)	142.90 ± 3.32 (1.15)	138.61 ± 1.78 (1.07)	164.66 ± 8.10 (1.27)	309.09 ± 11.50 (2.38)	348.79 ± 9.77 (2.98)	133.50 ± 2.55 (0.59)	338.68 ± 5.81 (2.81)	224.43 ± 2.15 (1.73)	153.54 ± 2.22 (1.18)	202.57 ± 4.82 (0.99)	176.55 ± 11.79 (1.38)	171.79 ± 5.04 (1.32)	320.99 ± 2.38 (2.47)	166.41 ± 7.13 (1.28)	184.29 ± 1.53 (1.42)	235.96 ± 8.58 (1.82)	208.07 ± 8.58 (1.80)	145.61 ± 3.95 (1.12)	151.97 ± 2.52 (1.17)	129.80 ± 2.65 (1.50)
replicate all rows Duration for updating all 1000 rows of the table (with 5 warmup iterations)	253.66 ± 9.23 (1.80)	188.54 ± 6.65 (1.34)	157.97 ± 4.48 (1.12)	150.15 ± 1.22 (1.07)	168.52 ± 3.18 (1.20)	283.64 ± 4.39 (2.01)	303.32 ± 8.82 (2.15)	140.96 ± 0.80 (1.00)	359.00 ± 16.78 (2.85)	238.86 ± 1.25 (1.69)	163.23 ± 2.88 (1.18)	1006.57 ± 6.82 (7.14)	175.93 ± 6.98 (1.25)	190.51 ± 4.08 (1.35)	319.95 ± 5.60 (2.27)	219.74 ± 2.04 (1.58)	198.68 ± 6.05 (1.41)	224.46 ± 6.05 (1.59)	213.75 ± 1.88 (1.52)	154.55 ± 1.87 (1.10)	158.72 ± 2.70 (1.13)	140.94 ± 1.42 (1.00)
partial update Time to update the first of every 10th row (with 5 warmup iterations)	10.70 ±0.30 (1.00)	9.86 ±0.30 (1.00)	11.22 ±0.38 (1.00)	15.07 ±0.89 (1.00)	22.05 ±1.42 (1.38)	14.91 ±0.38 (1.00)	39.04 ±1.22 (2.44)	9.33 ±0.14 (1.00)	9.64 ±0.22 (1.00)	51.39 ±2.04 (3.21)	18.92 ±0.48 (1.18)	11.04 ±0.28 (1.00)	15.96 ±1.74 (1.00)	12.72 ±0.49 (1.00)	15.56 ±0.85 (1.00)	27.62 ±0.28 (1.73)	15.44 ±0.77 (1.00)	14.45 ±0.77 (1.00)	17.10 ±1.07 (1.00)	11.29 ±0.80 (1.00)	15.62 ±0.25 (1.00)	9.62 ±0.04 (1.00)
select row Duration to highlight a row in response to a click on the row (with 5 warmup iterations)	6.12 ±0.29 (1.00)	3.69 ±1.81 (1.00)	4.72 ±2.45 (1.00)	7.94 ±0.78 (1.00)	15.07 ±1.18 (1.00)	5.94 ±0.85 (1.00)	9.39 ±0.83 (1.00)	3.60 ±1.51 (1.00)	9.53 ±0.88 (1.00)	36.51 ±1.42 (2.28)	11.39 ±0.44 (1.00)	6.38 ±0.11 (1.00)	5.06 ±0.24 (1.00)	5.14 ±2.15 (1.00)	7.03 ±0.38 (1.00)	20.09 ±0.84 (1.25)	6.86 ±1.83 (1.00)	4.51 ±0.75 (1.00)	7.58 ±1.88 (1.00)	4.94 ±1.88 (1.00)	8.37 ±0.82 (1.00)	2.19 ±0.83 (1.00)
swap rows Time to swap 2 rows on a 1K table (with 5 warmup iterations)	12.15 ±0.27 (1.00)	11.34 ±0.38 (1.00)	10.32 ±0.37 (1.00)	15.00 ±0.88 (1.00)	25.80 ±0.80 (1.81)	15.26 ±0.88 (1.00)	37.63 ±1.43 (2.35)	10.24 ±0.88 (1.00)	12.99 ±0.87 (1.00)	52.97 ±0.88 (3.31)	18.86 ±0.88 (1.18)	138.84 ± 0.88 (8.88)	11.04 ±0.28 (1.00)	11.89 ±0.21 (1.00)	18.28 ±1.94 (1.14)	27.60 ±0.37 (1.73)	13.42 ±0.85 (1.00)	17.29 ±0.38 (1.00)	14.00 ±0.34 (1.00)	13.35 ±0.88 (1.00)	16.81 ±0.88 (1.00)	9.67 ±0.23 (1.00)
remove row Duration to remove a row (with 5 warmup iterations)	51.91 ±2.08 (1.12)	48.08 ±0.60 (1.00)	50.66 ±1.83 (1.00)	54.28 ±1.16 (1.17)	75.53 ±1.21 (1.82)	53.36 ±1.38 (1.15)	81.72 ±1.18 (1.78)	49.76 ±1.88 (1.07)	60.19 ±0.89 (1.28)	88.99 ±1.09 (1.81)	58.02 ±1.88 (1.25)	150.21 ± 0.83 (3.23)	54.62 ±1.08 (1.17)	49.30 ±1.81 (1.05)	62.90 ±1.17 (1.35)	67.50 ±0.43 (1.45)	50.04 ±1.01 (1.00)	61.03 ±2.41 (1.31)	55.54 ±1.88 (1.18)	52.00 ±2.18 (1.12)	55.36 ±2.18 (1.18)	46.51 ±0.84 (1.00)
create many rows Duration to create 10,000 rows	2208.73 ± 39.23 (1.75)	1832.15 ± 11.42 (1.45)	1425.61 ± 24.13 (1.13)	1413.02 ± 14.52 (1.12)	1599.92 ± 26.48 (1.27)	2496.88 ± 24.84 (2.31)	2918.16 ± 94.55 (2.31)	1327.86 ± 18.58 (1.05)	3369.83 ± 104.82 (2.87)	2775.28 ± 40.40 (2.20)	1641.46 ± 18.80 (1.30)	2147.02 ± 13.45 (1.70)	1745.30 ± 24.88 (1.38)	1854.13 ± 47.47 (1.47)	2960.74 ± 47.00 (2.35)	2257.77 ± 48.29 (1.79)	1824.01 ± 11.51 (1.45)	2131.22 ± 34.17 (1.69)	1886.40 ± 23.79 (1.50)	1422.61 ± 18.48 (1.13)	1551.08 ± 17.83 (1.23)	1280.60 ± 13.52 (1.00)
append rows to large table Duration for adding 1000 rows on a table of 10,000 rows	375.53 ± 7.19 (1.53)	288.12 ± 3.38 (1.17)	263.95 ± 3.80 (1.07)	348.49 ± 3.88 (1.42)	457.14 ± 16.81 (1.88)	457.75 ± 8.18 (1.88)	653.26 ± 14.54 (2.85)	245.79 ± 9.18 (1.00)	3782.61 ± 88.28 (15.38)	1312.40 ± 22.08 (5.34)	403.27 ± 4.74 (1.84)	344.85 ± 12.78 (1.40)	319.21 ± 9.58 (1.30)	316.13 ± 2.98 (1.28)	472.70 ± 11.08 (1.82)	1890.49 ± 10.53 (7.88)	327.64 ± 8.73 (1.33)	399.67 ± 8.93 (1.83)	352.52 ± 1.47 (1.43)	320.57 ± 3.83 (1.30)	369.41 ± 6.78 (1.50)	267.96 ± 3.24 (1.00)
clear rows Duration to clear the table filled with 10,000 rows	772.27 ± 8.18 (3.71)	342.39 ± 8.88 (1.85)	216.50 ± 3.86 (1.04)	228.69 ± 3.03 (1.15)	221.22 ± 4.12 (1.05)	251.29 ± 4.12 (1.21)	549.75 ± 15.83 (2.64)	267.91 ± 4.74 (1.00)	585.13 ± 21.97 (2.81)	296.66 ± 2.78 (1.43)	237.72 ± 3.90 (1.14)	269.36 ± 9.72 (1.50)	244.32 ± 5.52 (1.18)	341.62 ± 5.38 (1.84)	663.90 ± 10.51 (3.18)	363.14 ± 2.36 (1.75)	410.94 ± 7.88 (1.88)	470.46 ± 8.38 (2.28)	415.37 ± 3.83 (2.00)	217.30 ± 4.88 (1.05)	248.82 ± 3.33 (1.20)	209.50 ± 3.24 (1.01)
slowdown geometric mean	1.50	1.21	1.06	1.10	1.34	1.43	2.14	1.82	2.15	2.36	1.22	2.15	1.17	1.22	1.71	1.86	1.26	1.43	1.33	1.09	1.15	1.01

Hình 0-4 Bảng benchmark các framework Javascript nổi tiếng nhất hiện nay

Sử dụng Vue.js trong giao diện người dùng



Mô hình MVVM trong Vue.js

Hình 0-5 Mô hình MVVM trong Vue.js

Vue.js sử dụng mô hình MVVM (Model-View-ViewModel) với 3 đối tượng cần quan tâm là Model, View và ViewModel.

2.4.2. Đánh giá ưu nhược điểm của Vue.js

Ưu điểm:

- **Kích thước nhỏ gọn:**

Sự thành công của một JavaScript framework phụ thuộc vào kích thước của nó, kích thước nhỏ hơn sẽ được sử dụng nhiều hơn. Một trong những ưu điểm lớn nhất của Vue.js là kích thước nhỏ khoảng 18-21KB.

- **Dễ hiểu và phát triển ứng dụng:**

Một trong những lý do khiến Vue.js trở nên phổ biến là nó khá dễ hiểu. Người dùng có thể dễ dàng tích hợp Vue.js vào dự án web của mình vì cấu trúc đơn giản của nó.

- **Tính linh hoạt cao:**

Tính linh hoạt cao là một lợi thế khác của Vue.js, nó cho phép người dùng thiết kế template của mình trong file HTML, file JavaScript. Tính linh hoạt này cũng giúp dễ hiểu đối với các nhà phát triển React.js, Angular.js và bất kỳ JavaScript framework mới khác. Vue.js đã chứng tỏ rất nhiều lợi ích trong việc phát triển các ứng dụng đơn giản chạy trực tiếp từ trình duyệt.

- **Giao tiếp hai chiều:**

Vue.js cũng tạo cơ chế cho truyền thông hai chiều vì kiến trúc MVVM của nó, việc này làm cho Vue.js dễ dàng xử lý các khối HTML. Về khía cạnh này, Vue.js có vẻ rất giống với Angular.js

2.5. KẾT CHƯƠNG

Qua chương này, chúng em đã giới thiệu về ba công nghệ cần thiết cho đề tài là Node.js, MongoDB và Vue.js. Sơ lược về các công nghệ, phân tích ưu, nhược điểm của chúng và lý do để sử dụng trong đề tài này. Node.js thích hợp để làm nền tảng chạy ứng dụng phía server để phục vụ số lượng lớn kết nối đồng thời. MongoDB loại bỏ các kiểm tra ràng buộc rườm rà, đơn giản hóa dữ liệu lưu trữ nhằm tăng tốc độ truy vấn dữ liệu. Vue.js thích hợp để xây dựng ứng dụng web kiểu single page application, thiết kế cấu trúc component thuận tiện để bảo trì và nâng cao trải nghiệm người dùng. Ở chương sau sẽ trình bày về hiện trạng khảo sát thực tế, quy trình trao đổi dụng cụ học tập của sinh viên.

CHƯƠNG 3

KHẢO SÁT HỆ THỐNG

3.1. ĐÁNH GIÁ CHUNG HIỆN TRẠNG TRAO ĐỔI DỤNG CỤ HỌC TẬP

Quá trình khảo sát được tiến hành tại trường Đại học Công nghiệp Thực phẩm TP. HCM. Theo kết quả khảo sát phần lớn sinh viên có nhu cầu trao đổi dụng cụ học tập sẽ thực hiện trên hình thức trao đổi trực tiếp hoặc trao đổi qua mạng xã hội. Các loại dụng cụ học tập được trao đổi thông dụng là sách, giáo trình, tài liệu photo, các tài liệu điện tử như word, powerpoint, video clip, ... Nhìn chung mọi quá trình trao đổi dụng cụ học tập đều được thực hiện thủ công, sinh viên chủ động tìm người trao đổi phù hợp với nhu cầu, quá trình tìm kiếm và trao đổi diễn ra chậm, phạm vi trao đổi hẹp, phần lớn là trao đổi với người quen. Nhưng bù lại là có độ tin cậy cao, thông tin trao đổi rõ ràng.

Từ kết quả khảo sát thực tế, nhóm nghiên cứu đã tiến hành phân tích để xây dựng các chức năng cho phần mềm, khắc phục được những hạn chế vừa nêu ở trên. Phần mềm phải đảm bảo được các chức năng sau:

- Quản lý được thông tin sinh viên.
- Cho phép sinh viên đăng tin tìm kiếm hoặc cung cấp dụng cụ học tập.
- Quản lý bài đăng.
- Cho phép sinh viên bình luận trao đổi.
- Thông báo mỗi khi có bài đăng phù hợp với nhu cầu.

3.2. MÔ TẢ NGHIỆP VỤ

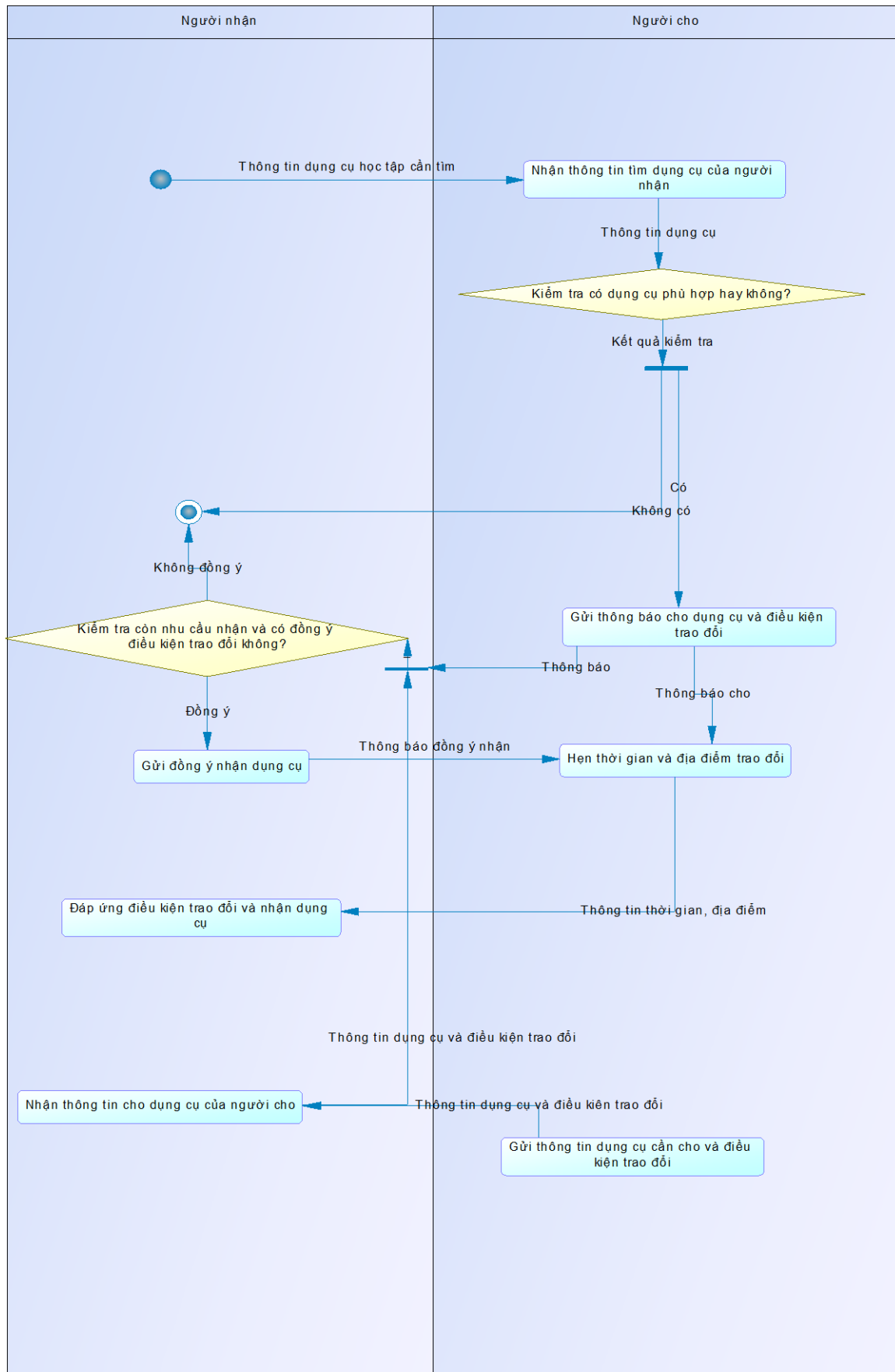
Tìm dụng cụ học tập

Sinh viên truyền thông tin tìm kiếm dụng cụ học tập của mình cho các sinh viên khác, sinh viên có dụng cụ học tập phù hợp sẽ ra điều kiện trao đổi, nếu cả hai đồng ý điều kiện thì sẽ tiếp tục hẹn thời gian và địa điểm thực hiện trao đổi dụng cụ học tập, ngược lại kết thúc trao đổi.

Cung cấp dụng cụ học tập

Sinh viên truyền thông tin cung cấp dụng cụ học tập mình đang có cùng với điều kiện trao đổi cho các sinh viên khác, sinh viên có nhu cầu cần tìm và đáp ứng được điều kiện đưa ra sẽ liên hệ hẹn thời gian và địa điểm thực hiện trao đổi dụng cụ học tập, ngược lại kết thúc trao đổi.

3.3. MÔ HÌNH BPM HIỆN TRẠNG TRAO ĐỔI DỤNG CỤ HỌC TẬP



Hình 0-1 Mô hình BPM hiện trạng trao đổi dung cụ học tập

3.4. KẾT CHƯƠng

Chương này chúng em đã khảo sát và đánh giá được sự khó khăn trong việc trao đổi dụng cụ học tập của sinh viên trường Đại học Công nghiệp Thực phẩm Tp. HCM. Qua quá trình khảo sát, chúng em đã đề ra các chức năng cần có trong ứng dụng hỗ trợ trao đổi dụng cụ học tập. Đồng thời cũng thiết kế mô hình mô tả nghiệp vụ trao đổi dụng cụ học tập thực tế. Chương sau sẽ đi sâu vào phân tích hệ thống đã khảo sát.

CHƯƠNG 4

PHÂN TÍCH HỆ THỐNG

4.1. GIỚI THIỆU

Ở chương trước chúng ta đã có cái nhìn thực tế về hiện trạng và quy trình trao đổi dụng cụ học tập của sinh viên. Chương này chúng em sẽ phân tích chi tiết về hệ thống ứng dụng trao đổi dụng cụ học tập.

4.2. PHÂN TÍCH XỬ LÝ

4.2.1. Giới thiệu mô hình phân cấp chức năng (Business Function Diagram - BFD)

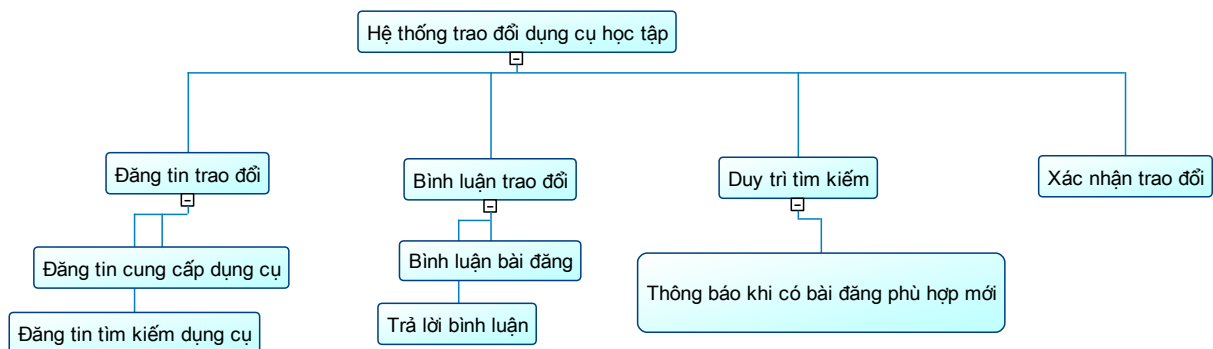
Mô hình phân cấp chức năng là công cụ để mô tả hệ thống qua phân rã có thứ bậc chức năng cho phép phân rã dần các chức năng mức cao thành chức năng chi tiết nhỏ hơn và kết quả cuối cùng thu được một cây chức năng. Mô hình phân cấp chức năng xác định một cách rõ ràng, dễ hiểu cái gì xảy ra trong hệ thống.

Gồm 2 thành phần chính: Các chức năng và các kết nối.

Mục đích:

- Xác định phạm vi hệ thống cần phân tích.
- Tiếp cận logic tới hệ thống, làm sáng tỏ các chức năng.
- Làm rõ nội dung từng công việc và trách nhiệm của từng bộ phận trong hệ thống.
- Loại các thao tác, tiến trình trùng lặp hoặc dư thừa.

4.2.2. Mô hình phân cấp chức năng (BFD) của hệ thống



Hình 0-1 Mô hình phân cấp chức năng (BFD) của hệ thống

4.2.3. Giới thiệu mô hình DFD (Data Flow Diagram)

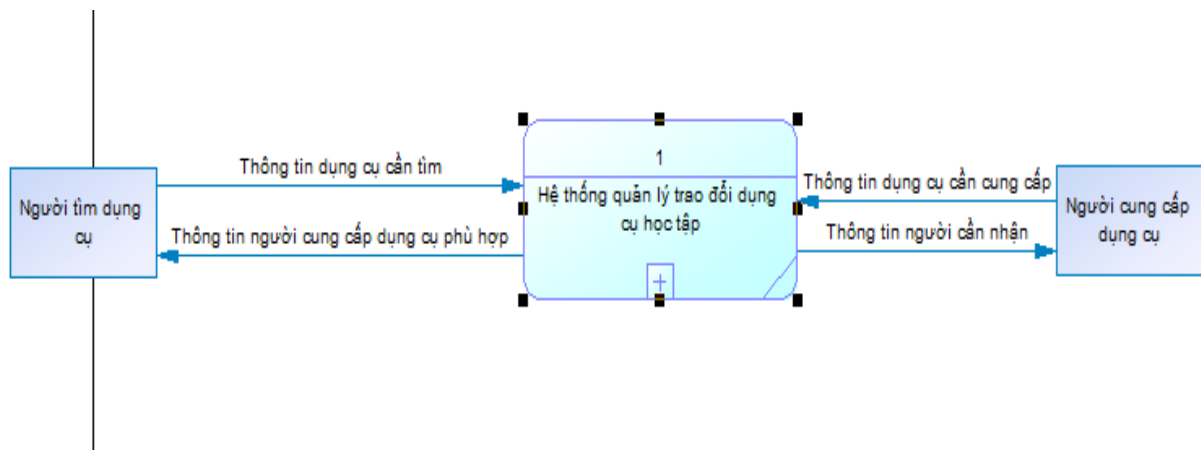
DFD biểu diễn sự kết nối giữa các hoạt động của hệ thống, thông qua việc trao đổi dữ liệu khi hệ thống hoạt động. Lưu đồ dòng dữ liệu phải thể hiện được:

- Xử lý nào phụ thuộc vào các xử lý nào.
- Dữ liệu cần cho mỗi xử lý.

Phương pháp tạo ra DFD: phân rã các hoạt động chi tiết dần đến khi có thể chuyển cho người lập trình triển khai. Mục đích của một DFD, một DFD sẽ mô tả:

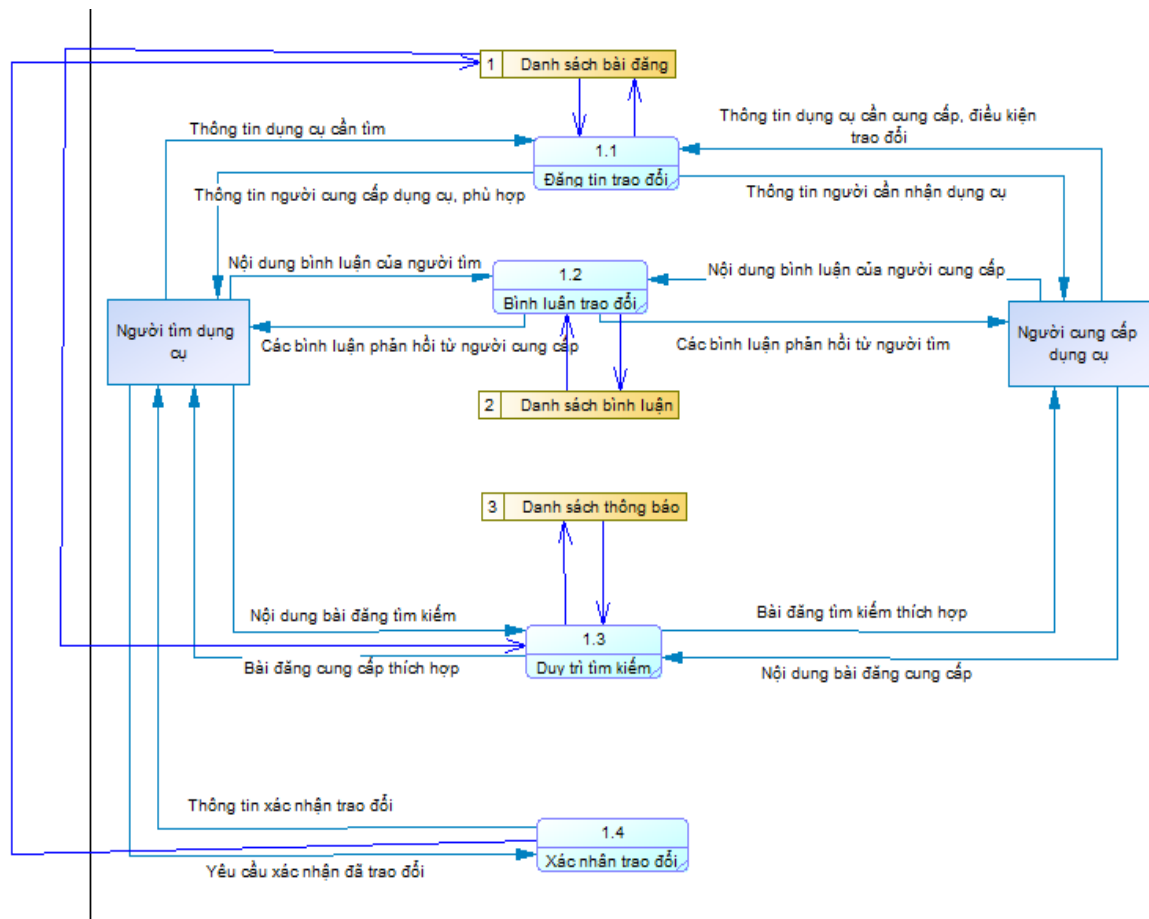
- Các quy trình xử lý bên trong hệ thống.
- Các kho lưu trữ hỗ trợ cho các thao tác của hệ thống.

4.2.4. Mô hình DFD của hệ thống



Hình 0-2 Mô hình DFD mức 0

Hệ thống hỗ trợ tương tác giữa người tìm dụng cụ và người cung cấp dụng cụ trong suốt quá trình trao đổi dụng cụ học tập. Cho phép sinh viên tạo thông tin tìm kiếm dụng cụ học tập và nhận được thông tin người cung cấp phù hợp. Sinh viên cũng có thể tạo tin cung cấp dụng cụ học tập và nhận được thông tin người có nhu cầu phù hợp.



Hình 0-3 Mô hình DFD phân rã mức 1

Quy trình trao đổi dụng cụ học tập được chia làm 4 quy trình chính với sự tương tác giữa các sinh viên với nhau:

- Đăng tin trao đổi
- Bình luận trao đổi
- Duyệt tìm kiếm
- Xác nhận trao đổi

Quá trình thực hiện sẽ có những thông tin cần thiết phải lưu trữ như sau:

- Danh sách bài đăng
- Danh sách bình luận của mỗi bài đăng
- Danh sách thông báo

4.3. PHÂN TÍCH DỮ LIỆU

4.3.1. Khái quát mô hình cơ sở dữ liệu quan niệm (CDM)

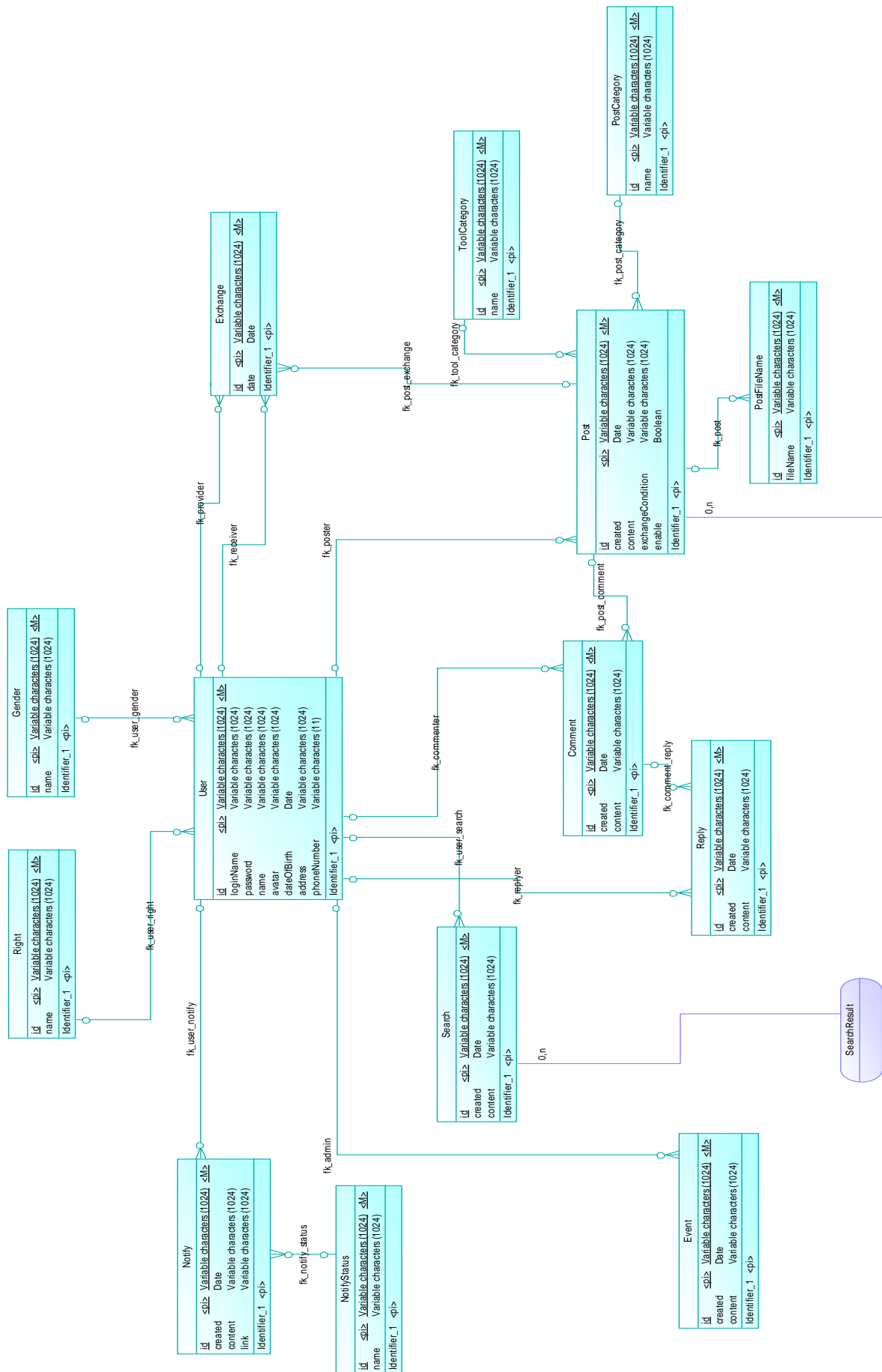
Mô hình dữ liệu quan niệm (Conceptual Data Model – CDM) là một dạng mô hình dùng để biểu diễn các đối tượng dữ liệu trong một hệ thống. Có nhiều dạng mô hình quan niệm dữ liệu như: mô hình dữ liệu mạng, mô hình dữ liệu phân cấp, mô

hình thực thể kết hợp,...Trong đó mô hình thực thể kết hợp là một trong những mô hình biểu diễn dữ liệu thông dụng nhất và được chọn để thiết kế trong chương này.

Các đối tượng chính trong mô hình:

- Thực thể: Là một đối tượng trong thế giới thực tồn tại một cách cụ thể hay tồn tại quan niệm. Ví dụ sinh viên, môn học, lớp.
- Thuộc tính: Là đối tượng mô tả thực thể mà nó trực thuộc. Ví dụ: Mã sinh viên, họ tên của sinh viên mô tả cho thực thể Sinh Viên.
- Mối kết hợp: Là đối tượng dùng liên kết hai thực thể với nhau bởi các dạng sau:
 - + Mối kết hợp Một – Nhiều: $1 - n$
 - + Mối kết hợp Một – Một: $1 - 1$
 - + Mối kết hợp Nhiều – Nhiều: $n - n$

4.3.2. Mô hình CDM cho hệ thống



Hình 0-4 Mô hình CDM cho hệ thống

4.3.3. Bảng sưu liệu

Bảng 0.1 Bảng User

Field	Type	Size	Note
Id	Varchar	1024	Mã người dùng
loginName	Varchar	1024	Tên đăng nhập
Password	Varchar	1024	Mật khẩu đăng nhập
Name	Varchar	1024	Tên người dùng
Avatar	Varchar	1024	Liên kết hình đại diện
DateOfBirth	Date		Ngày sinh
Address	Varchar	1024	Địa chỉ
PhoneNumber	Varchar	11	Số điện thoại
RightId	Varchar	1024	Mã nhóm quyền
GenderId	Varchar	1024	Mã giới tính

Bảng 0.2 Bảng Post

Field	Type	Size	Note
Id	Varchar	1024	Mã bài đăng
Created	Date		Thời gian tạo
Content	Varchar	1024	Nội dung bài đăng
ExchangeCondition	Varchar	1024	Điều kiện trao đổi
Enable	Boolean		Đang hoạt động
PosterId	Varchar	1024	Mã người đăng
ToolCategoryId	Varchar	1024	Mã loại dụng cụ
PostCategoryId	Varchar	1024	Mã loại bảng đăng

Bảng 0.3 Bảng PostCategory

Field	Type	Size	Note
Id	Varchar	1024	Mã loại bài đăng
Name	Varchar	1024	Tên loại bài đăng

Bảng 0.4 Bảng PostFileName

Field	Type	Size	Note
Id	Varchar	1024	Mã tệp
Filename	Varchar	1024	Tên tệp

Bảng 0.5 Bảng Comment

Field	Type	Size	Note
Id	Varchar	1024	Mã bình luận
Created	Date		Thời gian tạo
Content	Varchar	1024	Nội dung bình luận
CommenterId	Varchar	1024	Mã người bình luận
PostId	Varchar	1024	Mã bài đăng

Bảng 0.6 Bảng Reply

Field	Type	Size	Note
Id	Varchar	1024	Mã trả lời
Created	Date		Thời gian tạo
Content	Varchar	1024	Nội dung trả lời
ReplyerId	Varchar	1024	Mã người trả lời
CommentId	Varchar	1024	Mã bình luận

Bảng 0.7 Bảng Notify

Field	Type	Size	Note
Id	Varchar	1024	Mã thông báo
Created	Date		Thời gian tạo
Content	Varchar	1024	Nội dung thông báo
Link	Varchar	1024	Liên kết đến bài đăng
UserId	Varchar	1024	Mã người dùng được thông báo
NotifyStatusId	Varchar	1024	Mã trạng thái thông báo

Bảng 0.8 Bảng NotifyStatus

Field	Type	Size	Note
Id	Varchar	1024	Mã trạng thái thông báo
Name	Varchar	1024	Tên trạng thái thông báo

Bảng 0.9 Bảng Gender

Field	Type	Size	Note
Id	Varchar	1024	Mã giới tính
Name	Varchar	1024	Tên giới tính

Bảng 0.10 Bảng Exchange

Field	Type	Size	Note
Id	Varchar	1024	Mã trao đổi
Date	Date		Thời gian trao đổi
ProviderId	Varchar	1024	Mã người cung cấp
ReceiverId	Varchar	1024	Mã người nhận
PostId	Varchar	1024	Mã bài đăng được trao đổi

Bảng 0.11 Bảng Event

Field	Type	Size	Note
Id	Varchar	1024	Mã sự kiện
Created	Date		Thời gian tạo
Content	Varchar	1024	Nội dung sự kiện
AdminId	Varchar	1024	Mã người tạo sự kiện

Bảng 0.12 Bảng Search

Field	Type	Size	Note
Id	Varchar	1024	Mã tìm kiếm
Created	Date		Thời gian tạo
Content	Varchar	1024	Nội dung tìm kiếm
UserId	Varchar	1024	Mã người tìm kiếm

Bảng 0.13 Bảng ToolCategory

Field	Type	Size	Note
Id	Varchar	1024	Mã loại dụng cụ học tập
Name	Varchar	1024	Tên loại dụng cụ học tập

Bảng 0.14 Bảng Right

Field	Type	Size	Note
Id	Varchar	1024	Mã nhóm quyền
Name	Varchar	1024	Tên nhóm quyền

4.4. KẾT CHƯỠNG

Chương này đã trình bày được các mô hình phân cấp chức năng, mô hình luồng dữ liệu, mô hình dữ liệu quan niệm của hệ thống. Cho thấy được tổ chức hoạt động của hệ thống, các xử lý và dòng dữ liệu vào ra, các thực thể cần lưu trữ trong hệ thống. Chương sau sẽ đi vào thiết kế dữ liệu, giao diện cho hệ thống.

CHƯƠNG 5

THIẾT KẾ HỆ THỐNG

5.1. GIỚI THIỆU

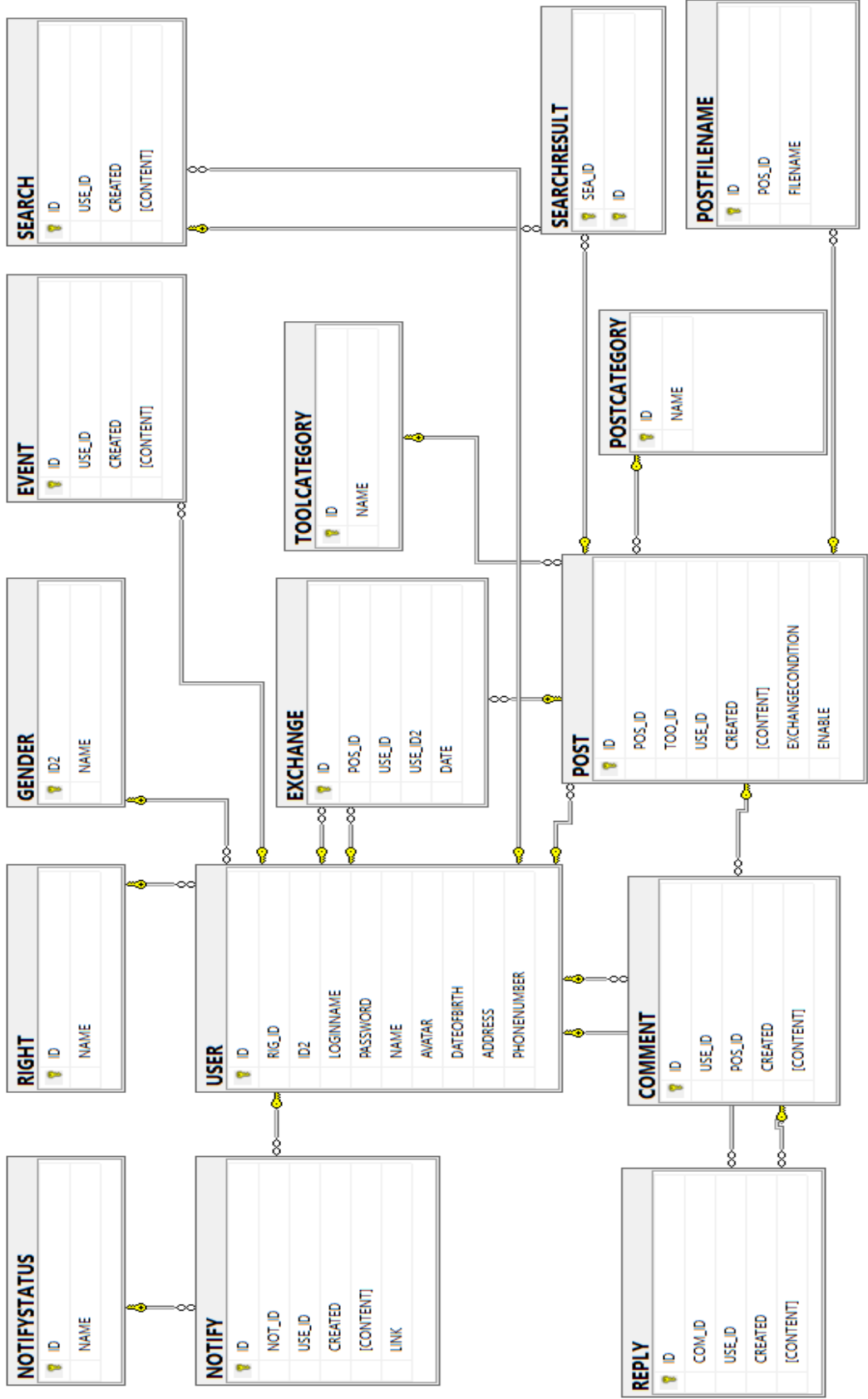
Ở chương trước chúng ta đã phân tích được các mô hình mô tả hoạt động của hệ thống. Chương này sẽ tiến hành thiết kế dữ liệu, xử lý và giao diện cho ứng dụng.

5.2. MÔ HÌNH QUAN HỆ

5.2.1. Giới thiệu mô hình quan hệ

Mô hình dữ liệu quan hệ bao gồm một hoặc nhiều quan hệ (Relation). Thực thể và thuộc tính trong mô hình CDM trở thành quan hệ và thuộc tính của quan hệ. Mỗi kết hợp sẽ trở thành khoá ngoại.

Lược đồ cơ sở dữ liệu quan hệ



Hình 0-1 Lược đồ cơ sở dữ liệu quan hệ

5.3. RÀNG BUỘC TOÀN VỆN

Các ràng buộc quan trọng:

5.3.1. Ràng buộc tham chiếu

Bảng 0.1 Ràng buộc tham chiếu

Phát biểu	Mỗi người dùng phải thuộc một nhóm quyền			
Nội dung	$\forall u \in User, \exists r \in Right: u.rightId = u.id$			
Bảng tầm ảnh hưởng	Quan hệ	Thêm	Xóa	Sửa
	User	+	-	+(rightId)
	Right	-	+	+(id)

Phát biểu	Mỗi bài đăng phải do một người dùng đăng			
Nội dung	$\forall p \in Post, \exists u \in User: p.posterId = u.id$			
Bảng tầm ảnh hưởng	Quan hệ	Thêm	Xóa	Sửa
	Post	+	-	+(PosterId)
	User	-	+	+(id)

5.3.2. Ràng buộc liên bộ

Bảng 0.2 Ràng buộc liên bộ

Phát biểu	Số điện thoại của mỗi người dùng không trùng nhau			
Nội dung	$\forall u1, u2 \in User, u1.phoneNumber \neq u2.phoneNumber$			
Bảng tầm ảnh hưởng	Quan hệ	Thêm	Xóa	Sửa
	User	+	-	+(phoneNumber)

5.3.3. Ràng buộc liên thuộc tính – liên quan hệ

Bảng 0.3 Ràng buộc liên thuộc tính – liên quan hệ

Phát biểu	Ngày trao đổi phải sau ngày sinh của người dùng			
Nội dung	$\forall e \in Exchange, \exists u \in User: e.date > u.dateOfBirth$			
Bảng tầm ảnh hưởng	Quan hệ	Thêm	Xóa	Sửa
	Exchange	+	-	+(date)
	User	-	+	+(dateOfBirth)

Phát biểu	Ngày thông báo phải sau ngày sinh của người dùng			
Nội dung	$\forall n \in Notify, \exists u \in User: n.created > u.dateOfBirth$			
Bảng tầm ảnh hưởng	Quan hệ	Thêm	Xóa	Sửa
	Notify	+	-	+(created)
	User	-	+	+(dateOfBirth)

Phát biểu	Ngày đăng tin phải sau ngày sinh của người dùng			
Nội dung	$\forall p \in Post, \exists u \in User: p.created > u.dateOfBirth$			
Bảng tầm ảnh hưởng	Quan hệ	Thêm	Xóa	Sửa
	Post	+	-	+(created)
	User	-	+	+(dateOfBirth)

Phát biểu	Ngày bình luận phải sau ngày sinh của người dùng			
Nội dung	$\forall c \in Comment, \exists u \in User: c.created > u.dateOfBirth$			
Bảng tầm ảnh hưởng	Quan hệ	Thêm	Xóa	Sửa
	Comment	+	-	+(created)
	User	-	+	+(dateOfBirth)

Phát biểu	Ngày trả lời bình luận phải sau ngày sinh của người dùng			
Nội dung	$\forall r \in Reply, \exists u \in User: r.created > u.dateOfBirth$			
Bảng tầm ảnh hưởng	Quan hệ	Thêm	Xóa	Sửa
	Reply	+	-	+(created)
	User	-	+	+(dateOfBirth)

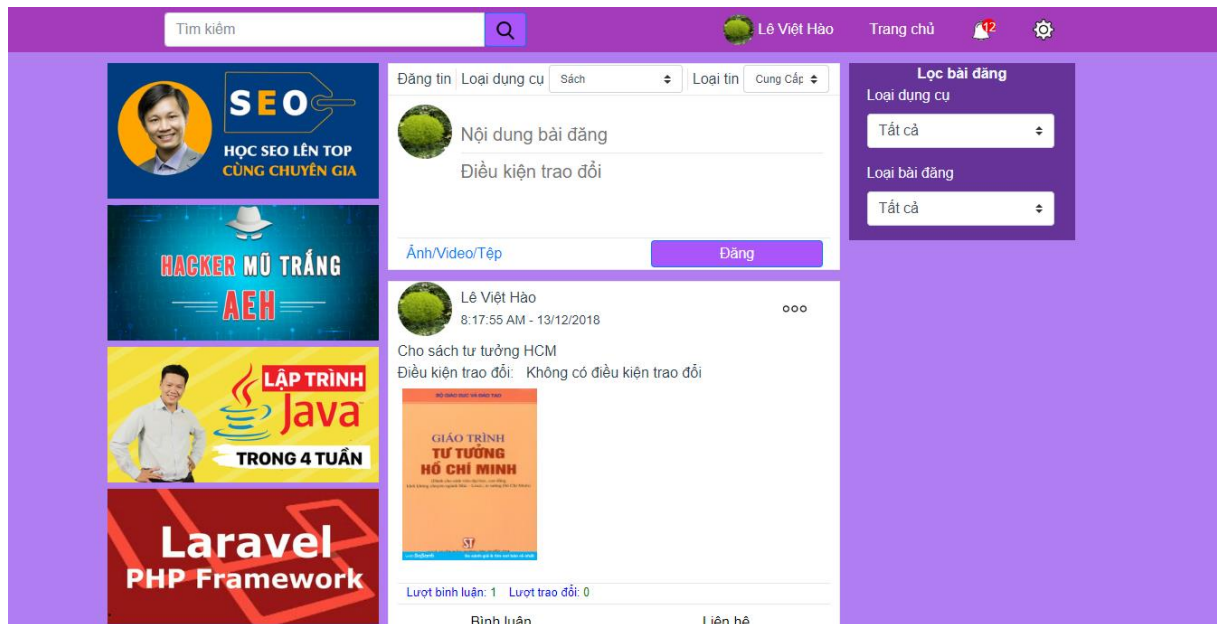
5.4. THIẾT KẾ GIAO DIỆN

5.4.1. Giao diện đăng nhập

Hình 0-2 Giao diện đăng nhập ứng dụng

Sinh viên cần tạo tài khoản để có thể sử dụng ứng dụng, sau khi có tài khoản sinh viên nhập tên đăng nhập và mật khẩu để đăng nhập vào ứng dụng.

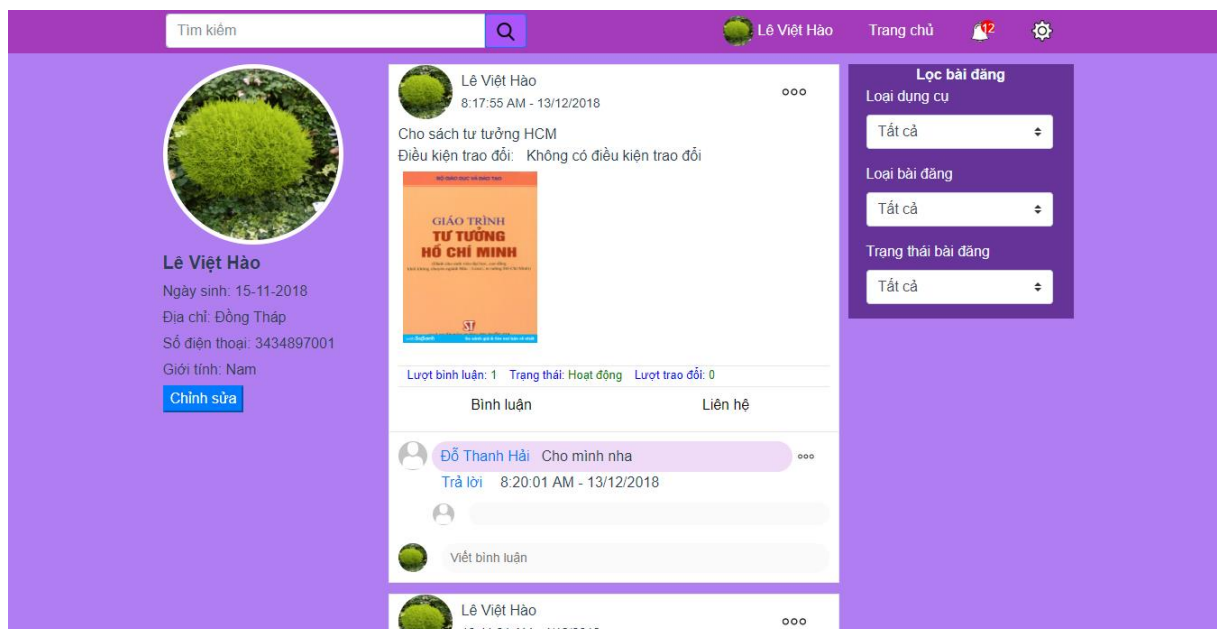
5.4.2. Giao diện trang chủ



Hình 0-3 Giao diện trang chủ

Các bài đăng tin trao đổi dụng cụ học tập mới nhất được hiển thị trên trang chủ. Ở đây sinh viên cũng có thể tạo bài đăng mới, tìm kiếm dụng cụ học tập, bình luận trao đổi trên bài đăng, xác nhận đã trao đổi.

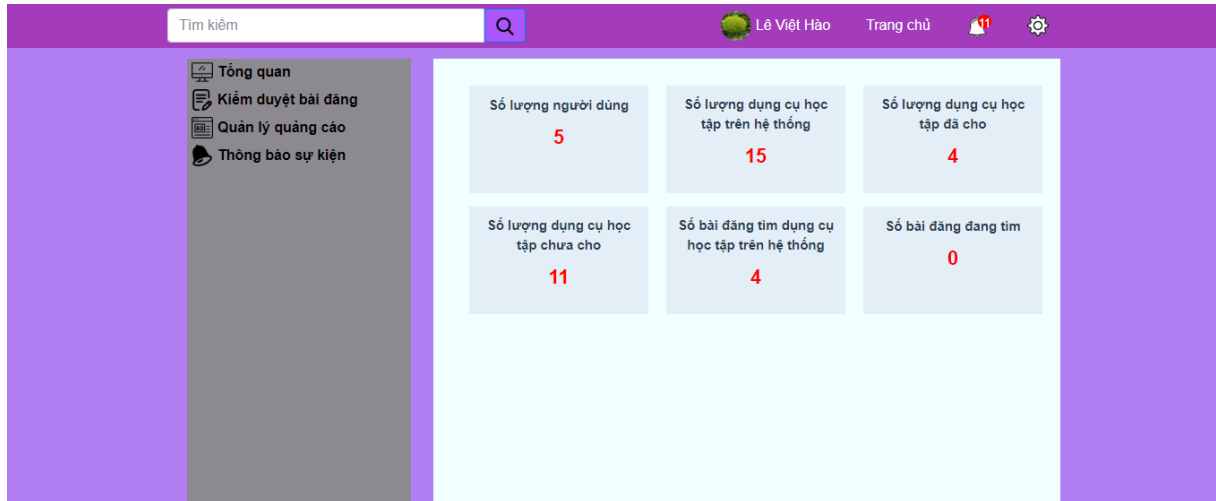
5.4.3. Giao diện trang thông tin cá nhân



Hình 0-4 Giao diện trang thông tin cá nhân

Ở đây sinh viên quản lý trang cá nhân của mình, gồm thông tin cá nhân, các bài đăng của mình, cập nhật các thông tin tương ứng.

5.4.4. Giao diện trang quản trị



Hình 0-5 Giao diện trang quản trị

Quản trị viên sẽ xem các số liệu thống kê của ứng dụng ở trang này, duyệt bài đăng của người dùng, gửi thông báo sự kiện cho sinh viên, quản lý các quảng cáo hiện trên trang.

5.5. KẾT CHƯƠng

Chúng ta đã thiết kế đầy đủ về cơ sở dữ liệu, phân tích các ràng buộc toàn vẹn quan trọng, xử lý và giao diện cho các màn hình chức năng. Ứng dụng đã có thể đi vào hoạt động thử nghiệm.

KẾT LUẬN ĐỀ XUẤT VÀ HƯỚNG PHÁT TRIỂN

Sau thời gian nghiên cứu và làm việc dưới sự hướng dẫn trực tiếp của cô Trần Thị Bích Vân, em đã hoàn thành đề tài đúng hạn, đáp ứng được các yêu cầu và mục tiêu đã đề ra. Tổng kết lại kết quả trong thời gian vừa qua, em đã thực hiện được:

- Hoàn tất tài liệu phân tích thiết kế hệ thống. Đây là tài liệu rất quan trọng và cần thiết trước khi bắt đầu giai đoạn viết code cho chương trình.
- Thiết kế cơ sở dữ liệu dựa theo quy trình khảo sát và phân tích nghiệp vụ.
- Tìm hiểu và áp dụng được các công nghệ sử dụng trong đề tài nghiên cứu.
- Xây dựng hoàn thiện ứng dụng trao đổi dụng cụ học tập dành cho sinh viên.

Những hạn chế khó khăn

Các mô hình và cơ sở dữ liệu chưa thực sự hoàn chỉnh do đây là dự án thực tế nên không tránh khỏi những vấn đề phát sinh trong quá trình thực tế.

Vì chưa từng làm việc với Node.js, MongoDB, và Vue.js trước đó nên còn gặp nhiều khó khăn trong quá trình tìm hiểu cũng như thực hiện.

Hướng phát triển

- Phát triển các chức năng hoàn thiện hơn.
- Phát triển ứng dụng trên điện thoại di động
- Tích hợp trí tuệ nhân tạo để kiểm tra nội dung bài đăng lành mạnh

Dù em đã rất cố gắng tuy nhiên kết quả nghiên cứu không thể tránh khỏi những thiếu sót, kính mong quý thầy cô tận tình hướng dẫn thêm, góp ý xây dựng để em có thể tự mình đúc rút và hoàn thiện! Em xin chân thành cảm ơn quý Thầy Cô đặc biệt là cô Trần Thị Bích Vân đã tận tình hướng dẫn em suốt thời gian qua.

TÀI LIỆU THAM KHẢO

Tiếng việt

[1] ThS Hoàng Thị Liên Chi (Chủ biên) – ThS Nguyễn Văn Lễ, Bài giảng Phân tích thiết kế hệ thống thông tin (Lưu hành nội bộ), Đại học Công Nghiệp Thực Phẩm TP.HCM, Tháng 1 năm 2015.

[2] ThS Hoàng Thị Liên Chi (Chủ biên) – ThS Nguyễn Văn Lễ, Giáo trình Cơ sở dữ liệu (Lưu hành nội bộ), Đại học Công Nghiệp Thực Phẩm TP.HCM, Năm 2015

[3] ThS Nguyễn Thị Bích Ngân (Chủ biên) - Slide bài giảng môn Công nghệ phần mềm, Khoa CNTT, ĐH CNTP TPHCM

Website

[4] Express Tutorial: The Local Library website https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/Tutorial_local_library_website

[5] Vue.js Guide <https://vuejs.org/v2/guide>

[6] MongoDB Document <https://docs.mongodb.com/manual>

[7] Node.js Document <https://nodejs.org/en/docs>

NHẬN XÉT CỦA GIÁO VIÊN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....