

## MỤC LỤC

<b>LỜI CẢM ƠN</b> .....	1
<b>LỜI NÓI ĐẦU</b> .....	2
<b>CHƯƠNG 1: CƠ SỞ LÝ THUYẾT HỆ THỐNG THÔNG TIN</b> .....	3
1.1. Khái niệm về hệ thống thông tin .....	3
1.2. Hệ thống thông tin quản lý .....	3
1.2.1. Phân loại hệ thống thông tin .....	4
1.2.2. Các giai đoạn của phân tích thiết kế hệ thống .....	7
1.3. Tổng quan về SQL Server và cơ sở dữ liệu quan hệ .....	8
1.3.1. Khái quát về quá trình phát triển của MS SQL Server .....	8
1.3.2. Vài nét về cơ sở dữ liệu mô hình quan hệ .....	9
1.3.3. Thiết kế cơ sở dữ liệu quan hệ .....	11
1.3.4. SQL Server và mô hình Client/ Server .....	12
1.3.5. Tạo và cài đặt cơ sở dữ liệu .....	13
1.4. Microsoft visual basic 6.0 .....	21
1.4.1. Khái quát về sự phát triển của công cụ lập trình Visual Basic 6.0 .....	21
1.4.2. Kiến thức cơ bản về lập trình VB6.0 .....	22
<b>CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG</b> .....	33
2.1. Khảo sát hệ thống .....	33
2.2. Phân tích hiện trạng hệ thống .....	33
2.2.1. Nhập sách .....	33
2.2.2. Mượn sách .....	33
2.2.3. Trả sách .....	34
2.2.4. Báo cáo thống kê .....	34
2.3. Mô tả hoạt động của thư viện .....	34
2.4. Bảng nội dung công việc .....	35
2.5. Mô hình nghiệp vụ .....	37
2.5.1. Bảng phân tích xác định các chức năng, tác nhân và hồ sơ .....	37
2.5.2. Biểu đồ ngữ cảnh .....	38
2.5.3. Nhóm dần các chức năng .....	39
2.5.4. Sơ đồ phân rã chức năng .....	40
2.5.5. Hồ sơ dữ liệu sử dụng .....	42
2.5.6. Ma trận thực thể chức năng .....	43
2.5.7. Biểu đồ hoạt động của từng nghiệp vụ .....	44
2.5.8. Sơ đồ luồng dữ liệu .....	48
2.6. THIẾT KẾ CƠ SỞ DỮ LIỆU .....	52
2.6.1. Mô hình liên kết thực thể (ER) .....	52
2.6.2. Mô hình dữ liệu quan hệ .....	55
2.6.3. Mô hình quan hệ .....	58
<b>CHƯƠNG 3: THIẾT KẾ CÀI ĐẶT CHƯƠNG TRÌNH</b> .....	59
3.1. Sơ đồ chức năng hệ thống .....	59
3.2. Thiết kế giao diện .....	60
3.2.1. Giao diện form “Cập nhật danh mục sách” .....	60
3.2.2. Giao diện form “Cập nhật thông tin mượn trả sách” .....	61
3.2.3. Giao diện form “Cập nhật thông tin độc giả” .....	62
3.2.4. Giao diện form “Cập nhật danh sách thủ thư” .....	63
3.2.5. Giao diện form “Cập nhật thông tin biên bản sự cố” .....	64
<b>KẾT LUẬN</b> .....	65
<b>TÀI LIỆU THAM KHẢO</b> .....	66

## LỜI CẢM ƠN

Trước tiên em xin gửi lời cảm ơn chân thành tới thầy Phùng Anh Tuấn người đã trực tiếp hướng dẫn, giúp đỡ tận tình để em hoàn thiện đồ án tốt nghiệp này.

Em xin gửi lời cảm ơn tới các thầy cô giáo khoa công nghệ thông tin của trường Đại học Dân lập Hải Phòng đã dìu dắt, dạy dỗ chúng em về kiến thức chuyên môn tinh thần độc lập, sáng tạo để em có những kiến thức thực hiện đề tài này.

Đồng thời em xin chân thành cảm ơn tới các thầy cô, các anh chị công tác tại trung tâm Aptech- Hải Phòng đã giúp đỡ, tạo điều kiện cho em khi triển khai đề tài.

Em xin bày tỏ lòng biết ơn sâu sắc của mình tới gia đình, bạn bè, những người luôn sát cánh bên em, tạo mọi điều kiện để em hoàn thành đồ án này.

Trong quá trình thực hiện đề tài mặc dù đã cố gắng hết sức nhưng do thời gian và khả năng có hạn nên không tránh khỏi những thiếu sót. Vì vậy em rất mong quý thầy cô góp ý và giúp đỡ để đồ án của em được hoàn thiện hơn.

Em xin chân thành cảm ơn!

*Hải Phòng, ngày 10 tháng 07 năm 2010*

Sinh viên

Bùi Thị Quỳnh Liên

## LỜI NÓI ĐẦU

Ngày nay với sự phát triển mạnh mẽ của xã hội, của khoa học công nghệ thì chúng ta không thể phủ nhận vai trò to lớn của ngành công nghệ thông tin. Việc ứng dụng công nghệ thông tin trong nhiều lĩnh vực của đời sống, xã hội, và trong sản xuất cung như quản lý. Nhờ vậy mà chỉ trong một thời gian ngắn nhưng nó đã đem lại hiệu quả vô cùng to lớn, góp phần thúc đẩy sự phát triển của xã hội cũng như nâng cao chất lượng cuộc sống cho con người.

Từ những lợi ích của công nghệ thông tin thì việc đào tạo ra đội ngũ có hiểu biết về công nghệ thông tin và vận dụng triệt để nó vào trong thực tế cuộc sống. Chính vì nắm bắt được tình hình chung đó, Aptech tự hào là nhà cung cấp dịch vụ đào tạo Công nghệ Thông tin (CNTT) hàng đầu tại Việt Nam cấp Chứng chỉ Quốc tế. Các Trung tâm Đào tạo Lập trình viên Quốc tế Aptech tại Việt Nam vinh dự là thành viên trong số hơn 3500 trung tâm đào tạo của Tập đoàn Aptech, tập đoàn lớn nhất thế giới về đào tạo CNTT, có mặt ở 53 quốc gia. Aptech - Hải Phòng là một thành viên của Aptech luôn cập nhật những chương trình giảng dạy mới cũng như cách thức quản lý tạo hiệu quả cho hoạt động và phát triển của trung tâm. Song song với việc đào tạo thì việc áp dụng công nghệ thông tin vào trong quản lý là điều tất yếu. Tuy vậy thì với một số lĩnh vực vẫn còn làm thủ công. Nhưng mục tiêu để đơn giản hóa và tiện nghi hóa luôn là mục tiêu cho việc quản lý của trung tâm. Vì vậy việc triển khai các chương trình trợ giúp cho quản lý đang được tiến hành và sẽ cho vào hoạt động trong thực tế. Như vậy chắc chắn sẽ làm cho Aptech-Hải Phòng khẳng định được thương hiệu của mình trong lĩnh vực công nghệ thông tin.

# CHƯƠNG 1: CƠ SỞ LÝ THUYẾT HỆ THỐNG THÔNG TIN

## 1.1. Khái niệm về hệ thống thông tin

Có nhiều định nghĩa về hệ thống thông tin khác nhau và các khái niệm liên quan. Về mặt kỹ thuật, hệ thống thông tin (HTTT) được xác định như một tập hợp các thành phần được tổ chức để thu thập, xử lý, lưu trữ, phân phối và biểu diễn thông tin, trợ giúp việc ra quyết định và kiểm soát hoạt động trong một tổ chức. Ngoài các chức năng kể trên, nó còn có thể giúp người quản lý phân tích các vấn đề, cho phép nhìn thấy được một cách trực quan những đối tượng phức tạp, tạo ra các sản phẩm mới. Liên quan đến HTTT là các khái niệm sẽ đề cập tới như dữ liệu, thông tin, hoạt động thông tin, xử lý dữ liệu, giao diện...

**Dữ liệu (Data)** là những mô tả về sự vật, con người và sự kiện trong thế giới mà chúng ta gặp bằng nhiều cách thể hiện khác nhau, như bằng ký tự, chữ viết, biểu tượng, hình ảnh, âm thanh, tiếng nói,...

**Thông tin (Information)** cũng như dữ liệu, đã có nhiều định nghĩa khác nhau. Một định nghĩa bao trùm hơn cả, xem thông tin là dữ liệu được đặt vào một ngữ cảnh với một hình thức thích hợp và có lợi cho người sử dụng cuối cùng.

**Các hoạt động thông tin (Information Activities)** là các hoạt động xảy ra trong một HTTT, bao gồm việc nắm bắt, xử lý, phân phối, lưu trữ, trình diễn dữ liệu và kiểm tra các hoạt động trong HTTT.

**Xử lý (Processing)** dữ liệu được hiểu là các hoạt động lên dữ liệu như tính toán, so sánh, sắp xếp, phân loại, tổng hợp... làm cho nó thay đổi về nội dung, vị trí hay cách thể hiện.

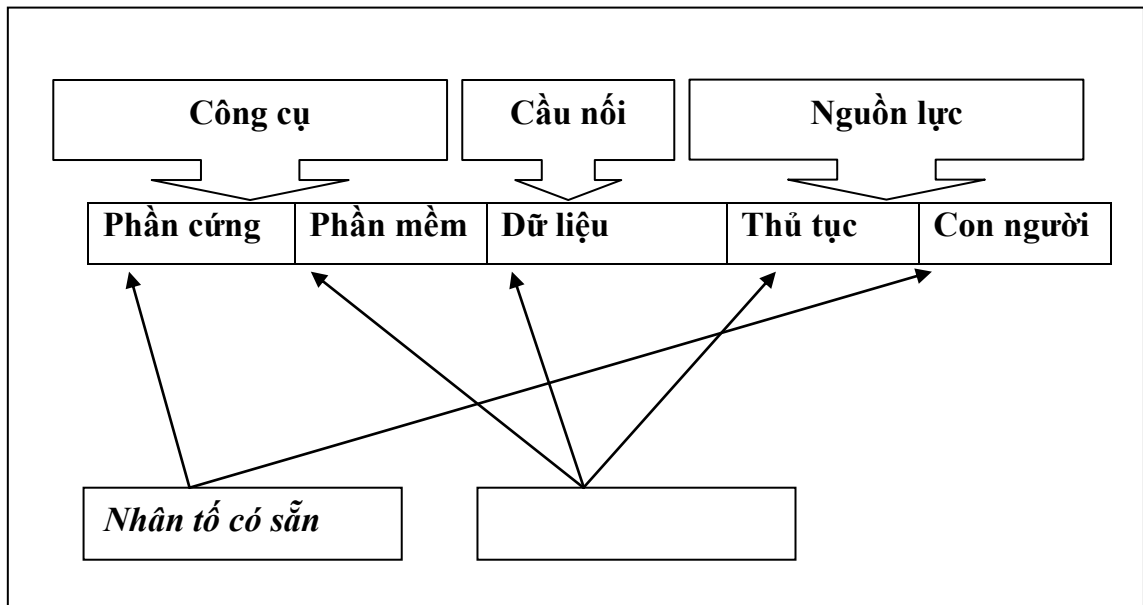
**Giao diện (Interface)** là chỗ mà tại đó hệ thống trao đổi dữ liệu với hệ thống khác hay môi trường. Ví dụ, giao diện của một HTTT thường là màn hình, bàn phím, chuột, micro, loa hay card mạng...

**Môi trường (Environment)** là phần của thế giới không thuộc hệ thống mà có tương tác với hệ thống thông qua các giao diện của nó.

## 1.2. Hệ thống thông tin quản lý

Trong các HTTT, HTTT quản lý (Management Information System - MIS) được biết đến sớm và phổ biến nhất. Đối tượng phục vụ của HTTT quản lý thực sự rộng hơn rất nhiều so với ý nghĩa bản thân tên gọi của nó. HTTT quản lý là sự phát triển và sử dụng HTTT có hiệu quả trong một tổ chức (Keen, Peter G.W.- một người đứng đầu

trong lĩnh vực này).



Hình 1.1. Các yếu tố cấu thành của HTTT

Năm yếu tố cấu thành của HTTT quản lý xét ở trạng thái tĩnh là: Thiết bị tin học máy tính, các thiết bị, các đường truyền,...(phần cứng), các chương trình (phần mềm), dữ liệu, thủ tục- quy trình và con người (hình 1.1). Các định nghĩa về HTTT trên đây giúp cho việc định hướng quá trình phân tích, thiết kế hệ thống. Tuy vậy, sự mô tả này là chưa đủ, cần đi sâu phân tích cụ thể mới có được sự hiểu biết đầy đủ về một hệ thống thực và cho phép ta xây dựng cơ sở dữ liệu các chương trình và việc bố trí các thành phần bên trong nó.

### 1.2.1. Phân loại hệ thống thông tin

#### 1. Hệ thống tự động văn phòng

Hệ thống tự động văn phòng (Office Automation System - OAS) là HTTT gồm máy tính với các hệ phần mềm như hệ xử lý văn bản, hệ thư tín điện tử, hệ thống lập lịch làm việc, bảng tính, chương trình trình diễn báo cáo...cùng các thiết bị khác như máy fax, máy in, điện thoại tự ghi...chúng được thiết lập nhằm tự động hóa công việc ghi chép, tạo văn bản và giao dịch bằng lời, bằng văn bản làm tăng năng suất cho những người làm công việc văn phòng.

#### 2. Hệ thống truyền thông

Hệ thống truyền thông (Communication System - CS) giúp cho việc thực hiện các trao đổi thông tin, dữ liệu giữa các thiết bị dưới các hình thức khác nhau với những khoảng cách xa một cách dễ dàng, nhanh chóng và chất lượng. Hệ thống này đóng vai

trò phục vụ cho các HTTT quản lý, hệ trợ giúp điều hành và các hệ khác hoạt động hiệu quả. Ngày nay, trong điều kiện phát triển của internet, truyền thông được xem như bộ phận của HTTT.

### **3. Hệ thống xử lý giao dịch**

Hệ thống xử lý giao dịch (Transaction Processing System - TPS) là một HTTT nghiệp vụ. Nó phục vụ cho hoạt động của tổ chức ở mức vận hành. Nó thực hiện việc ghi nhận các giao dịch hàng ngày cần thiết cho hoạt động nghiệp vụ của tổ chức để giao dịch với khách hàng, nhà cung cấp, những người cho vay vốn... như hệ thống lập hóa đơn bán hàng, hệ thống giao dịch ở các ngân hàng, hệ thống bán vé của các hãng hàng không... Nó là HTTT cung cấp nhiều dữ liệu nhất cho các hệ thống khác trong tổ chức.

### **4. Hệ thống cung cấp thông tin thực hiện**

Hệ thống cung cấp thông tin thực hiện (Executive Information System - EIS) có từ rất sớm, nó cung cấp các thông tin thực hiện trong một tổ chức. Nó là hệ máy tính nhằm tổng hợp và làm các báo cáo về khoảng thời gian nhất định. Các tổng hợp, báo cáo được thực hiện theo mẫu với nội dung, quy trình tổng hợp rất đơn giản, rõ ràng và có định hạn thời gian.

### **5. Hệ thống thông tin quản lý**

Hệ thống thông tin quản lý (Management Information System - MIS) trợ giúp các hoạt động quản lý của tổ chức như lập kế hoạch, kiểm tra thực hiện, tổng hợp và làm báo cáo, làm các quyết định quản lý trên cơ sở các quy trình thủ tục cho trước. Nhìn chung, nó sử dụng dữ liệu từ các hệ xử lý giao dịch và tạo ra các báo cáo định kỳ hay theo yêu cầu. Hệ này không mềm dẻo và ít có khả năng phân tích.

### **6. Hệ trợ giúp quyết định**

Hệ trợ giúp quyết định (Decision Support System - DSS) là hệ máy tính được sử dụng ở mức quản lý của tổ chức. Nó còn có nhiệm vụ tổng hợp các dữ liệu và tiến hành các phân tích bằng các mô hình để trợ giúp cho các nhà quản lý ra những quyết định có quy trình (bán cấu trúc) hay hoàn toàn không có quy trình biết trước (không có cấu trúc). Nó phải sử dụng nhiều loại dữ liệu khác nhau nên các cơ sở dữ liệu phải được tổ chức và liên kết tốt. Hệ còn có nhiều phương pháp xử lý (các mô hình khác nhau) được tổ chức để có thể sử dụng linh hoạt. Các hệ này thường được xây dựng chuyên dụng cho mỗi tổ chức cụ thể mới đạt hiệu quả cao.

## **7. Hệ chuyên gia**

Hệ chuyên gia (Expert System - ES) là một hệ trợ giúp quyết định ở mức chuyên sâu. Ngoài những kiến thức, kinh nghiệm của các chuyên gia và các luật suy diễn, nó còn có thể trang bị các thiết bị cảm nhận để thu các thông tin từ những nguồn khác nhau. Hệ có thể xử lý, và dựa vào các luật suy diễn để đưa ra những quyết định là ở chỗ: hệ chuyên gia yêu cầu những thông tin xác định đưa vào để đưa ra quyết định có chất lượng cao trong lĩnh vực hẹp, dùng ngay được.

## **8. Hệ trợ giúp điều hành**

Hệ trợ giúp điều hành (Execution Support System - ESS) được sử dụng ở mức quản lý chiến lược của tổ chức. Nó được thiết kế hướng sự trợ giúp cho các quyết định không cấu trúc bằng việc làm ra các đồ thị phân tích trực quan và các giao dịch rất thuận tiện với môi trường. Hệ được thiết kế để cung cấp hay chốt lọc các thông tin đa dạng lấy từ môi trường hay từ các hệ thống thông tin quản lý, hệ trợ giúp quyết định.

## **9. Hệ trợ giúp làm việc theo nhóm**

Trong điều kiện nhiều người cùng tham gia thực hiện một nhiệm vụ. Hệ trợ giúp làm việc theo nhóm (Groupware System - GS) cung cấp các phương tiện trợ giúp sự trao đổi trực tuyến các thông tin giữa các thành viên trong nhóm, làm rút ngắn sự ngăn cách giữa họ cả về không gian và thời gian.

## **10. Hệ thống thông tin tích hợp**

Một HTTT của tổ chức thường gồm một vài loại HTTT cùng được khai thác. Có như vậy mới đáp ứng được mục tiêu của tổ chức. Điều này cho thấy, cần phải tích hợp nhiều HTTT khác loại để đảm bảo sự hoạt động hiệu quả của tổ chức. Việc tích hợp các HTTT trong một tổ chức có thể tiến hành theo hai cách: xây dựng một hệ thống thông tin tích hợp (Integrated Information System - IIS) tổng thể hoặc tích hợp các hệ đã có bằng việc ghép nối chúng nhờ các “cầu nối”. Việc sử dụng các hệ tích hợp tổng thể thường đưa tổ chức đến một hệ thống tập trung, một sự phối hợp và kiểm soát chặt chẽ. Nhưng chúng cũng tạo ra sức ỳ về quản lý, và sự quan liêu trong hoạt động và khó thay đổi. Khi sự tập trung của một HTTT đã đạt đến một điểm bão hòa, nhiều tổ chức bắt đầu cho các bộ phận của mình tiếp tục phát triển những hệ con với các đặc thù riêng. Ngày nay trong môi trường web, nhiều hệ thống phát triển trên môi trường này có thể tích hợp bằng cách ghép nối với nhau một cách dễ dàng nhờ công cụ portal.

## 1.2.2. Các giai đoạn của phân tích thiết kế hệ thống

### 1. Khảo sát hiện trạng xác định yêu cầu hệ thống

Xác định yêu cầu là bước đầu tiên và quan trọng của một hệ thống thông tin, nó quyết định đến chất lượng hệ thống thông tin được xây dựng trong các bước sau này. Việc thu thập các thông tin của hệ thống chính là việc tiến hành khảo sát hệ thống, việc khảo sát hệ thống được chia làm hai giai đoạn:

- Giai đoạn khảo sát sơ bộ: nhằm hình thành dự án phát triển hệ thống thông tin.
- Giai đoạn khảo sát chi tiết: nhằm thu thập các thông tin chi tiết của hệ thống phục vụ phân tích yêu cầu thông tin làm cơ sở cho các bước sau này.

### 2. Phân tích hệ thống

#### a) Phân tích hệ thống về chức năng

Phân tích hệ thống về chức năng hiểu một cách đơn giản là xác định các chức năng nghiệp vụ cần được tiến hành của hệ thống sau khi đã khảo sát thực tế và đi sâu vào các thành phần của hệ thống.

Các bước tiến hành:

- Diễn tả chức năng từ mức vật lý về mức logic, từ mức đại thể về mức chi tiết.
- Xây dựng sơ đồ phân cấp chức năng.
- Xây dựng dòng dữ liệu.

#### b) Phân tích hệ thống về dữ liệu

Phân tích hệ thống về dữ liệu là việc phân tích về cấu trúc thông tin được dùng và được tổ chức bên trong hệ thống đang khảo sát, xác định được mối quan hệ tự nhiên giữa các thành phần thông tin, hay nói cách khác, đây là quá trình lập lược đồ khái niệm về dữ liệu, làm căn cứ cho việc thiết kế cơ sở dữ liệu sau này.

Việc phân tích dữ liệu thường thực hiện qua hai giai đoạn:

- Đầu tiên lập lược đồ dữ liệu theo mô hình thực thể/liên kết, nhằm phát huy thế mạnh về tính trực quan và dễ vận dụng của mô hình này, bao gồm:
  - + Xác định các kiểu thực thể cùng với các kiểu thuộc tính của nó.
  - + Xác định các mối quan hệ giữa các kiểu thực thể.
- Tiếp đó hoàn thiện lược đồ dữ liệu theo mô hình quan hệ nhằm lợi dụng cơ sở lý luận chặt chẽ của mô hình này trong việc chuẩn hóa lược đồ, bao gồm:
  - + Xác định các kiểu thuộc tính của các kiểu thực thể.
  - + Chuẩn hóa danh sách các thuộc tính, từ đó xác định các kiểu thực thể đã được



chuẩn hóa.

+ Xác định mối quan hệ.

### 3. Thiết kế hệ thống

Trong khi giai đoạn phân tích nghiệp vụ thuần túy xử lý cho quan điểm logic về hệ thống, thì giai đoạn thiết kế hệ thống bao gồm việc xem xét ngay lập tức cả khả năng cài đặt các yêu cầu nghiệp vụ này bằng cách sử dụng máy tính.

Tùy theo quy mô của hệ thống mà các giai đoạn thiết kế có thể áp dụng khác nhau. Sau đây là các tiến trình đơn giản nhất trong thiết kế hệ thống:

- Thiết kế được xem xét bắt đầu từ màn hình tương tác, các dữ liệu vào và ra (các báo cáo) đến cơ sở dữ liệu và các tiến trình xử lý chi tiết bên trong.
- Thiết kế vật lý: là quá trình chuyển mô hình logic trừu tượng thành bản thiết kế vật lý hay các đặc tả kỹ thuật. Trong thiết kế vật lý cần phải quyết định chọn ngôn ngữ lập trình, hệ cơ sở dữ liệu, cấu trúc file tổ chức dữ liệu, phần cứng, hệ điều hành và môi trường mạng cần xây dựng.
- Hoàn thiện chương trình.

#### 1.3. Tổng quan về SQL Server và cơ sở dữ liệu quan hệ

##### 1.3.1. Khái quát về quá trình phát triển của MS SQL Server

SQL là sản phẩm Database Engine của hãng Microsoft, đang gây sốc trên thị trường. Microsoft có ý định đầu tư và hỗ trợ cho sản phẩm này đặc biệt là việc đưa MS SQL Server trở thành hệ thống cơ sở dữ liệu chính trong công nghiệp máy tính với hệ điều hành Windows NT.

Trong những năm 1970, hãng IBM đã phát minh ra ngôn ngữ máy tính được thiết kế đặc trưng dùng trong việc truy vấn cơ sở dữ liệu được gọi là SEQEL (Structured English Query Language). Theo thời gian, ngôn ngữ này đã được bổ sung đến mức nó không chỉ là ngôn ngữ dùng cho truy vấn mà còn có thể tạo ra các cơ sở dữ liệu và quản lý tính năng bảo mật của hệ thống cơ sở dữ liệu. IBM đã công bố rộng rãi, và nó được biết đến với tên gọi là SQL. Do đó có thể gọi nó là sequel hay đánh vần nó thành “S-Q-L”. Có nhiều phiên bản khác nhau của SQL được dùng cho các hệ thống cơ sở dữ liệu hiện nay. SQL Server của Microsoft đang dùng phiên bản Transact-SQL, hay T-SQL, thay cho Transact SQL.

Microsoft bắt đầu xây dựng SQL Server (một sản phẩm cơ sở dữ liệu sử dụng ngôn ngữ SQL) với Sybase để dùng cho OS/2. Vào năm 1988 phiên bản thử nghiệm

“Ashton-Tate/Microsoft SQL Server” chạy trên môi trường OS/2 và đến năm 1989 phiên bản 1.0 được phát hành đánh dấu sự kết thúc liên kết với Ashton-Tate.

Năm 1990 phiên bản đầu tiên với tên Microsoft SQL Server 1.2 được phát hành với các công cụ tiện ích, thư viện lập trình và công cụ quản lý. Phiên bản này sửa lỗi cho phiên bản 1.0 và chạy trên môi trường OS/2. Điều quan trọng là nó hỗ trợ client hệ điều hành Windows 3.0.

Năm 1991 phát hành phiên bản Microsoft SQL Server 1.11. Vào thời điểm này, Microsoft không tham gia phát triển OS/2 với IBM mà dồn sức theo đuổi phát triển hệ điều hành nhiều người dùng Windows NT.

Đầu năm 1992 phát hành phiên bản Microsoft SQL Server 4.2 gồm giao diện đồ họa quản lý cơ sở dữ liệu trên môi trường Windows.

Cuối năm 1992 phiên bản MS SQL Server chạy trên môi trường Windows NT ra đời. Vào thời điểm này, Microsoft quyết định chỉ tiếp tục phát triển trên môi trường Windows NT. Phiên bản này hỗ trợ kiến trúc 32-bit.

Giữa năm 1995 phiên bản MS SQL Server 6.0 được phát hành với nhiều tính năng mới: tạo bản sao (replication), con trỏ dịch chuyển được (scrollable) và những công cụ quản lý dữ liệu. Phiên bản này trở thành đối thủ cạnh tranh thực sự của cơ sở dữ liệu khác như: Sybase, Infomix và Oracle database.

Đến năm 1996 phiên bản MS SQL Server được phát hành. Phiên bản này phù hợp với chuẩn SQL (ANSI SQL).

Năm 1999 phiên bản SQL Server 7.0 ra đời. Phiên bản này là một máy cơ sở dữ liệu (Database Engine) hoàn toàn được xây dựng lại, đầy đủ mức độ khoá cấp thấp, tiến trình truy vấn mới. Nó cũng bao gồm những tiện ích quản lý những Wizard. Phiên bản này đáp ứng yêu cầu của khả năng mở rộng, phát triển từ ứng dụng cho máy để bàn đến ứng dụng thương mại Internet. Phiên bản này có thể chạy trên Windows NT và Windows 9x.

Phiên bản mới nhất cuối cùng là MS SQL Server 2000 được phát hành vào giữa năm 2000. Phiên bản này hỗ trợ cho XML, động cơ tìm kiếm (Full text search) cho phép chạy trên môi trường đa xử lý (Multiprocessor).

### **1.3.2. Vài nét về cơ sở dữ liệu mô hình quan hệ**

Cơ sở dữ liệu quan hệ là cơ sở dữ liệu bên trong nó được tổ chức thành các bảng. Các bảng được tổ chức bằng cách nhóm dữ liệu theo cùng chủ đề và có chứa các

cột và các dòng thông tin. Sau đó các bảng này lại liên hệ với nhau bởi bộ Database Engine kho nó yêu cầu.

Nói chung, một cơ sở dữ liệu có thể hiểu như là tập hợp các dữ liệu có liên quan với nhau.

Các đối tượng cơ sở dữ liệu quan hệ bao gồm:

**Table (bảng):** Là những đối tượng chứa các kiểu dữ liệu và các dữ liệu thô thực sự. Field (trường) hay Column (cột) : đây là các thành phần của bảng dùng để chứa dữ liệu.

Data Type (kiểu dữ liệu): Có nhiều kiểu dữ liệu khác nhau để chọn, như kiểu ký tự (charater), số (Number), ngày tháng (Date), và các dữ liệu khác. Mỗi một kiểu dữ liệu sẽ được gán cho một trường nào đó bên trong bảng.

**Stored Procedure (thủ tục được lưu):** đối tượng này tương tự như một macro mà trong đó mã T-SQL có thể viết và lưu dưới một tên gọi. Chạy đối tượng này tức là ta đang chạy mã T-SQL chứa trong đó.

**Triger:** Triger là một Stored Procedure hoạt động khi dữ liệu được thêm vào, chỉnh sửa, hay được xoá từ cơ sở dữ liệu.

**Rules (quy luật):** Một rule khi được gán cho một cột nào đó sẽ đảm bảo sao cho dữ liệu đưa vào phải phù hợp với tiêu chuẩn mà bạn đặt ra.

**Primary Key (khoá chính):** Mặc dù không phải là một đối tượng trong cơ sở dữ liệu, các khoá là tối cần cho cơ sở dữ liệu. Các khoá chính bắt buộc là duy nhất trong các hàng, cung cấp cách xác định duy nhất các mục mà ta muốn lưu trữ.

**Foreign Key (khoá ngoại):** Cũng không phải là một đối tượng trong cơ sở dữ liệu, một khoá ngoại là các cột tham chiếu đến khoá chính hay các ràng buộc duy nhất của bảng khác. SQL Server sử dụng khoá chính và khoá ngoại để liên hệ dữ liệu với nhau từ các bảng riêng biệt khi một câu truy vấn được thực hiện.

**Constaraintsc (ràng buộc):** Ràng buộc là cơ chế nhằm bảo đảm tính toàn vẹn dữ liệu, phụ thuộc vào hệ thống và server.

**Default (mặc định):** một mặc định có thể xác lập trên trường sao cho nếu không có dữ liệu nào được đưa vào thì giá trị mặc định sẽ được dùng đến.

**Views:** Một view về cơ bản là một câu truy vấn được lưu trữ trong cơ sở dữ liệu có thể tham chiếu đến một hay nhiều bảng. Nó có thể được tạo ra và được lưu tại sao cho nó có thể dùng dễ dàng về sau. Một view thường chứa một số cột được lấy từ một bảng hay từ một liên kết hai hay nhiều bảng.

**Index (chỉ mục):** Chỉ mục là một trong cách tổ chức dữ liệu để truy cập nhanh quá trình truy vấn.

### 1.3.3. Thiết kế cơ sở dữ liệu quan hệ

#### 1. Quá trình thiết kế một cơ sở dữ liệu quan hệ

Quá trình thiết kế một cơ sở dữ liệu quan hệ gồm các bước sau :

- Phân tích hoàn cảnh, thu nhập thông tin về dữ liệu được đề cập trong bài toán.
- Quyết định các bảng gồm các cột, kiểu dữ liệu, và chiều dài của chúng.
- Chuẩn hoá các bảng dữ liệu đã xây dựng được.
- Tạo cơ sở dữ liệu, các bảng và thiết lập mối quan hệ giữa các bảng .

Khi ta nắm được mối liên hệ dữ liệu, ta sẽ tổ chức nó thành các bảng có quan hệ với nhau và tuân thủ theo quy luật chuẩn hoá (Normalization rules).

Để bắt đầu quá trình thiết kế, ta xem xét kỹ hoàn cảnh nghiệp vụ và về công việc đang muốn thực hiện. Tận dụng các ý kiến tập thể về các thông số, các biến số khác nhau và cách ghép chúng với nhau trong các bảng. Nghiên cứu các mẫu báo cáo và câu truy vấn phục vụ cho mục đích của người dùng.

<b>Việc làm cần thiết trong thiết kế CSDL</b>	<b>Việc không cần làm trong thiết kế CSDL</b>
Hỏi người sử dụng xem họ cần gì	Đừng coi nhẹ người sử dụng
Tạo danh sách các đối tượng	Đừng tạo các đối tượng mà không bao giờ dùng đến
Nên giữ cho tên đối tượng ngắn gọn nhưng mô tả được ý nghĩa của đối tượng	Không dùng các tên phức tạp, các tên có dấu cách, các tên có ký tự đặc biệt, vì rất khó gõ chúng
Tổ chức các thuộc tính của đối tượng thành các nhóm chuẩn xác	Đừng tạo cột chữ nhiều giá trị
Tạo các cột có tên giống nhau trong các cột khác nhau để sau đó liên hệ chúng với nhau. Các cột này trở thành khoá chính và khoá ngoại	
Kiểm tra lại thiết kế với một số dữ liệu mẫu	Đừng quá chắc rằng thiết kế khi làm việc tốt với 5 dòng, cũng thực hiện nhanh như vậy với 500,000 dòng
Tạo ít nhất 1 chỉ mục cho các bảng sẽ được truy vấn	Đừng tạo quá nhiều chỉ mục cho 1 bảng (quá 5)

Hãy thực hiện theo nguyên tắc đặt tên chuẩn cho đối tượng của cơ sở dữ liệu. Việc này giúp ta dễ dàng làm việc với các đối tượng bảng có tên Employees, và dùng idxLastname cho chỉ mục LastName	Đừng quên làm tài liệu thiết kế
--	---------------------------------

## 2. Chuẩn hoá dữ liệu

Sau khi đã quyết định về các cột, ta cần tổ chức dữ liệu thành các bảng có quan hệ. Quá trình này gọi là chuẩn hoá dữ liệu.

Các quy luật được xây dựng để chuẩn hoá dữ liệu:

- Dạng chuẩn thức nhất (First Normal Form - 1st NF): Theo quy luật này một cột không thể chứa nhiều giá trị.
- Dạng chuẩn thứ hai (Second Normal - 2nd NF): theo quy luật này mỗi cột không phải là khoá cần phải phụ thuộc vào khoá đầy đủ, và không phụ thuộc vào chỉ một phần của khoá chính.
- Dạng chuẩn thứ ba (Third Normal - 3rd NF): Tương tự như quy luật trước, theo quy luật này tất cả các cột không khoá không được phụ thuộc vào các cột không khoá khác.

### 1.3.4. SQL Server và mô hình Client/ Server

#### 1. Khái quát về mô hình Client/ Server

SQL Server của Microsoft là mô hình cơ sở dữ liệu kiểu client/ server.

Client/ server có thể được xác định như một ứng dụng được chia thành 2 phần : phần chạy trên server, và phần chạy trên các workstations (máy trạm). Phía Server của ứng dụng cung cấp độ bảo mật, tính chịu lỗi (Fault - tolerance), hiệu suất, chạy đồng thời, và sao lưu dự phòng tin cậy. Phía Client cung cấp giao diện với nhà sử dụng và có chứa các báo cáo, các truy vấn, và các Forms. SQL Server là phần server của chương trình, có nhiều Client khác nhau được chọn để từ đó có thể kết nối vào SQL Server, trong đó có cả các tiện ích của SQL Server, chẳng hạn như SQL Server Query Analyzer.

#### 2. Ưu điểm của mô hình Client / Server

Các ưu điểm của Client là :

- Dễ sử dụng

- Hỗ trợ trên nhiều nền phần cứng
- Hỗ trợ trên nhiều ứng dụng phần mềm
- Quen thuộc với người sử dụng

Các ưu điểm của Server là :

- Đáng tin cậy
- Chạy đồng thời nhiều ứng dụng
- Khả năng chịu lỗi cao
- Phần cứng hiệu suất cao
- Điều kiện tập trung

Trong mô hình Client/Server, khi truy vấn chạy, server sẽ dò tìm cơ sở dữ liệu và chỉ gửi những dòng nào phù hợp với yêu cầu của Client. Điều này không chỉ tiết kiệm mà còn có thể thực thi nhanh hơn so với khi trạm làm việc thực hiện các truy vấn đó, chỉ cần Server đủ mạnh.

### **1.3.5. Tạo và cài đặt cơ sở dữ liệu**

#### **1. Quy định về định danh (đặt tên) trong SQL Server**

Tất cả các bảng cũng như tên cột và tên đối tượng đề phải tuân theo các quy luật định danh của SQL Server như sau:

- Định danh (Identifiers) có thể có chiều dài từ 1 đến 128 ký tự Unicode, bao gồm chữ cái, ký hiệu, và số.

- Ký tự đầu tiên phải là chữ cái hay một trong những ký hiệu sau: @, #, hoặc \_ .  
Ký hiệu @ và # có ý nghĩa đặc biệt trong SQL Server.

- Có thể dùng ký tự #, \$, và \_ cho các ký tự đứng sau ký tự đầu tiên.

- Các định danh bắt đầu với ký hiệu @ được dùng cho các biến cục bộ.

- Các định danh bắt đầu với ký hiệu # được dùng để báo hiệu rằng đối tượng được tạo là đối tượng tạm thời và có thể tham chiếu bởi người sử dụng trong cùng phiên làm việc.

- Các định dạng bắt đầu với ký hiệu ## được dùng để báo hiệu rằng đối tượng được tạo là đối tượng tạm thời toàn cục và có thể tham chiếu bởi tất cả mọi người sử dụng trong cơ sở dữ liệu .

- Bạn có thể dùng khoảng trắng trong định danh, nhưng sau đó để tham chiếu đến các đối tượng này phải bao chúng trong dấu nháy hoặc dấu ngoặc vuông. Người sử

dụng cú pháp ngoặc vuông thường được dùng hơn do tránh sự nhầm lẫn giữa chuỗi và định danh.

## 2. Tạo cơ sở dữ liệu

Để tạo cơ sở dữ liệu mới trong SQL Server ta có thể dùng một trong 3 phương thức sau:

- Database Creation Wizard
- SQL Server Enterprise Manager
- Câu lệnh CREATE DATABASE

Trong phạm vi báo cáo ta chỉ khảo sát việc tạo cơ sở dữ liệu bằng lệnh CREATE DATABASE theo cú pháp sau:

```
CREATE DATABASE Database_Name
[ ON{[PRIMARY]
(NAME = logical_name,
FILENAME='physical_name'
[, SIZE=size]
[.MAXSIZE= mã_size" UNLIMITED]
[,FILEGROWTH=growth_increment]
}1,...n]]
[LOG ON
{(NAME=logical_name,
FILENAME='physical_name'
[,SIZE=size]
[.MAXSIZE=max_size:UNLIMITED]
[,FILEGROWTH=growth_increment]
}1,...n]]
[FOR LOAD :FOR ATTACH]
```

### ***Trong đó***

- Database \_ name : Tham số đại diện toàn bộ cơ sở dữ liệu.
- ON PRIMARY: Tham số này đặc tả file cơ sở dữ liệu này là thành viên của nhóm file nào
- NAME : Tham số này xác định tên file sẽ dùng trong SQL Server để gán cho file cơ sở dữ liệu vật lý trên đĩa cứng.

- FILENAME: Tham số là tên đường dẫn và tên file liên quan đến nơi lưu trữ dữ liệu trên đĩa cứng.
- SIZE: Tham số xác định kích thước file cơ sở dữ liệu. Kích thước mặc định bằng kích thước của file mẫu .
- MAXSIZE: Tham số này xác định kích thước tối đa mà cơ sở dữ liệu có thể đạt tới. Nếu không có tham số này thì tùy chọn AUTOGROWTH được chọn và cơ sở dữ liệu có thể chiếm toàn bộ ổ cứng.
- FILEGROWTH: Tham số này xác định gia lượng tăng được dùng để phát triển tự động cơ sở dữ liệu này. Giá trị mặc định là 1M.
- LOGON: Miêu tả nơi định vị các file nhật ký thao tác và kích thước của chúng.
- FORLOAD: tham số này đánh dấu cơ sở dữ liệu dành cho dbo. Tùy chọn này cung cấp để tương thích ngược để sử dụng trong các Version trước đó.
- FORATTACH: Tham số này gắn lại một tập các file hình thành lên cơ sở dữ liệu. Các file này được tạo trước và sau đó được tách khỏi SQL Server.
- N: có thể tạo cơ sở dữ liệu gồm n files. Khi ta xác định sử dụng nhiều file dữ liệu, SQL Server sẽ tự động trải thông tin qua tất cả các file dữ liệu được xác định . điều này giúp giảm tranh chấp và các điểm nóng (hotspot) trong dữ liệu.

### 3. Tạo bảng

#### a) Sơ lược về bảng

Bảng (Table) dùng để lưu tất cả dữ liệu trong cơ sở dữ liệu, và được tổ chức thành các hàng và cột (bản ghi và trường). Mỗi cột trong bảng có thể lưu trữ một loại thông tin nhất định gọi là kiểu dữ liệu.

Mỗi bảng đại diện cho một tập hợp thực thể trong cơ sở dữ liệu. Mỗi hàng trong bảng chính là một thực thể cụ thể, với các thuộc tính được miêu tả bởi giá trị của các cột tương ứng .

Mỗi cột (Trường) trong bảng cần được đặt tên, kiểu dữ liệu, độ rộng, và phải xác định rõ ràng cho phép có giá trị Null hay không. Các cột có thể được đặt tên theo bất kỳ thứ tự nào trong bảng. Tên cột phải duy nhất trong một bảng và phải tuân theo các quy luật định danh của SQL Server.

Mỗi cơ sở dữ liệu cho phép khai báo tới 2 triệu bảng, mỗi bảng có thể tối đa 1.024 cột. Chiều dài mỗi hàng tối đa là 8.092 ký tự. Độ rộng tối đa cho mỗi cột là 8.00 ký tự và bạn có thể có tới 1.024 cột trong một hàng, tuy nhiên, một hàng không thể



vượt quá một trang dữ liệu 8.192 byte cho dữ liệu và overhead (các dữ liệu phụ). Trường hợp ngoại lệ là đối với văn bản và kiểu dữ liệu 8.192 byte cho dữ liệu hình ảnh, cho phép tới 2GB thông tin văn bản và thông tin nhị phân. Những thông tin này không lưu trữ trong bảng nên không bị hạn chế bởi giới hạn trong một trang.

**b) Các kiểu dữ liệu**

Các kiểu dữ liệu mà SQL Server hỗ trợ :

<b>Loại dữ liệu</b>	<b>Kiểu dữ liệu</b>	<b>Chú giải</b>
Chuỗi ký tự (String)	Char(n), Varchar(n)	Lưu trữ các chuỗi ký tự Max n= 8,000 ký tự
Nhị phân (Binary)	Binary(n) Varbinary(n)	Lưu trữ thông tin nhị phân trong cặp 2 byte. Max n= 8,000 byte.
Số nguyên (integer)	Int, Smallint, Tinyint	Lưu trữ các giá trị nguyên Int 4 byte ±2,147,483,647; Smallint 2 byte±32,767; Tinyint 1 byte 0-255
Số gần đúng (Approximate Numeric)	Float, Real	Lưu trữ số gần đúng  Độ chính xác      Kích thước lưu trữ 1- 9                      5 byte 10-19                    9 byte 20-28                    13 byte 30-38                    17 byte
Số chính xác (Exact Number)	Decimal, Numeric	Lưu trữ các số chính xác  Độ chính xác      Kích thước lưu trữ 1- 9                      5 byte 10-19                    9 byte 20-28                    13 byte 30-38                    17 byte  Khi khai báo phải xác định cả độ tin chính xác (precision) và tỷ lệ (scale)
Kiểu đặc biệt (Special)	Bit, Text, Image	Lưu trữ bit, Văn bản nhiều hơn 8.000 byte, hay dữ liệu hình ảnh bit: lưu trữ thông tin logic.

		Text & Image: có thể lưu dữ liệu 2GB. Khi khai báo, con trỏ 16 byte sẽ được bổ sung vào hàng. Con trỏ này sẽ trỏ đến trang dữ liệu KB đầu tiên, thì một con trỏ 16 byte sẽ được phát sinh để trỏ tới các trang BLOB
Ngày và Giờ (Date and Time)	Datetime, smalldatetime	Lưu trữ ngày giờ Datetime 8 byte : 1/1/1753 – 31/12/9999 Small datetime: 4 byte : 1/1/1900 -6/6/2079
Tiền tệ (Money)	Money, Smallmoney	Lưu trữ các giá trị tiền tệ. Cả 2 loại đều có scale là 4. Money: 8 byte : 922,337,203,685,447.5805 Small Money: 4 byte :±214,748,3647
Kiểu tự động tăng (Auto-Incrementing Datatypes)	Dentity, Timestamp	Lưu trữ các giá trị cho phép sẽ tăng tự động hay được gán trị SQL Server
Tương đương (Sysonyms)	Character->Char Character(n)->Char(n) Char varying ->Varchar Char varying(n)->Varchar(n) Binary varying ->Varbinary Dec->Decimal Double precision->Float Float(n) n=1-7 ->Real Float(n) n=8-15 -> float Integer-> Int	Ánh xạ kiểu dữ liệu ANSI thành kiểu dữ liệu SQL Server

Kiểu người dùng định nghĩa (User-Defined)		Kiểu dữ liệu riêng do người dùng định nghĩa để lưu trữ thông tin
---	--	--

- **Sử dụng câu lệnh**

Cú pháp:

CREATE TABLE [database. [owner.] table\_name

(

Column\_name datatype[ identity/constraint/ NULL/NOT NULL]

[,...]

)

**Trong đó**

- Table\_name: Tên bảng mới, tuân thủ theo quy luật định danh
- Column\_name: Tên cột, tuân thủ theo quy luật định danh
- Datatype: Kiểu dữ liệu cột

Phần còn lại là tùy chọn. Có thể xác định thuộc tính Identity, các ràng buộc trường và khả năng có giá trị NULL.

- **Cú pháp xoá bảng**

DROP TABLE Table\_name

table\_name: Tên bảng cần xoá.

#### 4. Truy vấn dữ liệu

- **Các khái niệm cơ bản dùng trong truy vấn**

**Chọn danh sách:**

Danh sách lựa chọn được dùng trong truy vấn là danh sách liệt kê các cột dữ liệu, các biểu thức tính toán mà kết quả của truy vấn sau khi thực hiện phải đưa ra. Các thành phần của danh sách lựa chọn được phân cách nhau bởi dấu ‘,’.

Đối với cột dữ liệu: Những cột dữ liệu được lấy ra từ các bảng mà các cột này có mặt ở nhiều bảng khác nhau thì khi viết phải chỉ rõ trường này thuộc cơ sở dữ liệu nào theo cú pháp: [Tên Bảng]. [Tên Trường]. Trong đó [ Tên bảng] là tên bảng nguồn truy suất dữ liệu hoặc bí danh của bảng này ; [ Tên trường] là tên trường truy suất thông tin.

Đối với các biểu thức tính toán: cột này là kết quả của các phép toán ghép lại

phải được thiết lập theo đúng thứ tự ưu tiên thực hiện của các phép toán. Đối với các toán hạng là các cột thì phải tuân thủ kiểu viết đối với cột dữ liệu .

Các mục lựa chọn ta có thể thay đổi tiêu đề kết quả truy suất theo cú pháp :

<column\_name/expression>as new\_column\_name *hoặc*

new\_column\_name=<column\_name/expression>new\_column\_name

**Chú ý :** Đối với dữ liệu là các hằng số kiểu ký tự hoặc kiểu ngày giờ thì phải được bao trong dấu ‘’.

- **Cú pháp truy vấn**
- **Truy vấn lựa chọn**

SELECT [ALL/DISTINCT][TOP n [PERCENT] < select\_list>

[FROM <table\_sources>]

[WHERE <search\_condition>]

[ORDER BY <order\_list. ]

**Trong đó :**

<select\_list>: Danh sách chọn

<table\_sources>: nguồn dữ liệu

<search\_condition>: điều kiện lọc

<order\_list>: điều kiện sắp xếp

ALL,DISTINCT: kết quả truy vấn toàn bộ hay loại bỏ những hàng giống nhau.

TOP n [PERCENT]: kết quả truy vấn là một phần n dòng hoặc n %.

- **Truy vấn tạo bảng**

Lệnh SELECT INTO cho phép tạo mới một bảng dữ liệu cơ sở trong kết quả truy vấn. Bảng mới này dựa vào các cột mà ta định danh trong danh sách lựa chọn. Ta có thể tạo 2 kiểu bảng, cố định và tạm thời theo cú pháp:

SELECT column\_list

INTO new\_table\_name

FROM table\_list

WHERE search\_criteria

**Trong đó :**

new\_table\_name: là tên của bảng mới được tạo, nó không cần định nghĩa dữ liệu. Tên bảng phải là duy nhất và phải tuân thủ theo quy định đặt tên của SQL. Các cột trong column\_list bắt buộc phải có tiêu đề và được đặt tên là duy nhất

Các thành phần khác cũng như phần truy vấn lựa chọn

- **Truy vấn chèn dữ liệu**

Lệnh INSERT cơ bản thêm 1 hàng vào một bảng tại một thời điểm. Các biến thể của lệnh INSERT cho phép thêm nhiều hàng bằng cách chọn dữ liệu từ bảng khác hay thực thi một thủ tục được lưu (scored procedure). Trong bất cứ trường hợp nào ta cần phải biết về cấu trúc của bảng mà ta cần thiết phải chèn dữ liệu vào như:

- + Số cột trong bảng .
- + Kiểu dữ liệu từng cột
- + Một vài lệnh INSERT yêu cầu phải biết tên cột

Nắm vững những ràng buộc, và các thuộc tính của cột như thuộc tính định danh (Identity).

Cú pháp :

```
INSERT [INTO]
{< table_or_view>}
{[column_list]}
VALUE
({DEFAULT/constant_expression}[,...])/
select_statment/
execute_statment}/
DEFAULT VALUES }
```

Những cột không đề cập trong danh sách khi ta chèn một hàng nào mới vào, thì SQL SERVER phải tự xác định giá trị cho các cột này. Để SQL Server có thể tự định giá trị, mỗi cột không được đề cập trong danh sách phải tuân theo một tiêu chuẩn sau:

- + Cột được gán giá trị mặc định
- + Cột là cột định danh
- + Cột cho phép giá trị NULL
- + Cột có kiểu dữ liệu là timestamp

- **Truy vấn cập nhật dữ liệu**

Lệnh UPDATE cho phép ta thay đổi giá trị của các cột trong hàm theo cú pháp:

```
UPDATE {table_name/view}
SET column_name1={express1 /NULL/select_statment}
[,column_name2=...]
```

[WHERE search\_conditions]

Lệnh UPDATE có thể thay đổi cho nhiều cột. Từ khoá SET chỉ xuất hiện một lần, và các cột khác nhau được thay đổi cách khác nhau dấu ‘,’.

## 1.4. Microsoft visual basic 6.0

### 1.4.1. Khái quát về sự phát triển của công cụ lập trình Visual Basic 6.0

Visual Basic (VB) đã được phát triển qua nhiều phiên bản nhằm theo kịp tốc độ phát triển của công nghệ phần mềm.

Visual Basic 1.0 ra đời vào giữa những năm 1991. Đây là bản phát triển từ QuickBasic. VB được thiết kế để sử dụng cho phép thế hệ lập trình viên mới tạo những ứng dụng chạy trên môi trường Windows. Cốt lõi của phiên bản này là cung cấp một số công cụ đồ họa để tạo giao diện như textbox, combobox ...

Đầu năm 1992, phiên bản Visual Basic Professional Toolkit ra đời. Phiên bản này cập nhật một số công cụ hỗ trợ. Vào thời điểm này, phương pháp hướng đối tượng và tính dừng lại của chương trình trở lên quan trọng. Mặc dù chưa thể hiện được ngôn ngữ hướng đối tượng nhưng với các công cụ hỗ trợ, VB tiến dần đến mục tiêu trên. Những công cụ tùy biến thực thi các chức năng đặc biệt có thể dừng lại. Khả năng mở rộng sẽ là một trong những mục tiêu chính của các ứng dụng trong tương lai.

Visual Basic 2.0 ra đời năm 1992 bao gồm kiểu dữ liệu biến thể (variant), xác định hằng số True, False và biến đối tượng. Đây là một bước tiến trong việc xây dựng môi trường phát triển ngôn ngữ hướng đối tượng. Vào thời điểm này chỉ có VBSQL và ODBC API là phương pháp truy cập dữ liệu mà người phát triển có thể dùng. Cả 2 phương pháp này cung cấp truy cập 16-bit. VBSQL là khởi tổ của phương thức giao tiếp giữa SQL và VB

Năm 1993 Visual Basic 3.0 ra đời bao gồm các công cụ chuẩn. Những động cơ này cung cấp động cơ truy xuất cơ sở dữ liệu trong một ứng dụng với mã lệnh rất ít. Phiên bản này hỗ trợ công cụ tùy chọn VBX (16-Bit). Đi kèm phiên bản này là động cơ cơ sở dữ liệu Jet (phiên bản 1.1) Jet được dùng cho kết nối dữ liệu thông qua DAO (Data Access Objects) hoặc điều khiển data. Mặc dù Jet được phát triển đến ngày nay (phiên bản 4.0), nhưng ADO là một trong những thành phần chính trong chiến lược phát triển của Microsoft.

Visual Basic 4.0 hoàn thành vào năm 1995 được xem là một bước tiến bộ. Nó đuổi kịp những tiến bộ trong lĩnh vực phát triển phần mềm bởi công nghệ kết hợp

OLE (Object Linking and Embedding) và khả năng tạo những đối tượng. Phiên bản này cũng hỗ trợ điều khiển tùy biến 32-bit được gọi là OCX. Sự thay đổi này cũng được thực hiện bởi các ứng dụng khác như Excel và Word. Vì thế chúng có thể được truy cập từ bên trong ứng dụng Visual Basic.

Một phương thức truy cập dữ liệu mới thích hợp trong phiên bản này là RDO (Remote Data Objects) và Remote data control. RDO là một phương pháp truy cập dữ liệu được thiết kế thay thế cho DAO. Thư viện Active 32-bit này nhanh hơn và nhỏ hơn DAO và được thiết kế với một hệ thống đối tượng phân cấp giống như ODBC API. Tuy nhiên điều khiển này có những lỗi lớn không được phát triển tiếp.

Năm 1997 phiên bản 5.0 được phát hành. Nó hỗ trợ chuẩn COM của Microsoft và cho phép tạo các điều khiển Active X. Phiên bản này là một bước tiến vượt bậc bởi những người phát triển có thể dùng VB để tạo các điều khiển và thư viện liên kết động DLL riêng cho họ.

Phiên bản 6.0 được phát hành vào năm 1998. VB 6.0 hỗ trợ một cách lập trình tương đối mới, lập trình hướng đối tượng (Object - Oriented Programming).

Với lập trình hướng đối tượng ta chia nhỏ bài toán cần giải quyết thành các đối tượng. Mỗi đối tượng có những đặc điểm, mà gọi là thuộc tính (Properties) và các chức năng riêng biệt mà ta gọi là phương thức (Methods). Theo những yêu cầu đề ra, phiên bản này tăng cường phương pháp mới giao tiếp với SQL Server. Nó cải tiến các truy cập dữ liệu, nhiều công cụ và điều khiển mới cho giao tiếp với cơ sở dữ liệu (ADO), cung cấp các tính năng Web và những wizard mới. Phương pháp này đánh dấu với công nghệ ADO 2.0 – Phương thức truy cập dữ liệu tốt nhất và nhanh nhất hiện nay. Nó giao tiếp với OLE DB tương tự như RDO nhưng nhỏ hơn và có cấu trúc phân cấp đơn giản hơn. ADO được thiết kế cho thực thi các ứng dụng kinh doanh hay ứng dụng Internet. Phiên bản hiện nay là 2.6.

#### **1.4.2. Kiến thức cơ bản về lập trình VB6.0**

### **1. Các khái niệm cơ bản**

#### **a) Thuộc tính**

Mỗi đối tượng đều có bộ thuộc tính mô tả đối tượng. Mặc dù mỗi đối tượng có những bộ thuộc tính khác nhau, nhưng trong đó, có những thuộc tính thông dụng cho hầu hết đối tượng (Form, Control). Ta có thể xem đầy đủ các thuộc tính của từng đối tượng thông qua cửa sổ Properties)

Các thuộc tính thông dụng gồm :

Thuộc tính	Mô tả
Left/Top	Vị trí tương ứng của cạnh trái / cạnh trên so với đối tượng (điều khiển) chứa nó
Height/Width	Tương ứng chiều cao / Chiều rộng của đối tượng (điều khiển)
Name	Giá trị là chuỗi được dùng đặt tên đối tượng (điều khiển)
Enable	Giá trị Logic (True hoặc False) quyết định người sử dụng có thể tương tác với điều khiển hay không
Visible	Giá trị Logic (True hoặc False) quyết định người sử dụng có thể nhìn thấy điều khiển hay không

**b) Phương thức**

Phương thức là những chương trình chứa trong điều khiển, cho điều khiển biết cách thức để thực hiện một công việc nào đó. Mỗi điều khiển có những phương thức khác nhau nhưng vẫn có một số phương thức thông dụng cho hầu hết các điều khiển.

Các phương thức thông dụng gồm:

Phương thức	Mô tả
Move	Thay đổi vị trí một đối tượng theo yêu cầu của chương trình
Drag	Thi hành hoạt động kéo và thả người sử dụng
Setfocus	Lựa chọn / chuyển tới đối tượng được chỉ ra trong Code
Zorder	Quy định thứ tự xuất hiện các điều khiển trên màn hình

**c) Sự kiện**

Nếu như thuộc tính mô tả đối tượng, phương thức chỉ ra cách thức đối tượng hành động thì sự kiện là những phản ứng của đối tượng. Tương tự như thuộc tính và phương thức, mỗi điều khiển có một bộ các sự kiện khác nhau, nhưng trong đó có một số sự kiện thông dụng với hầu hết các điều khiển. Các sự kiện này xảy ra thường là kết quả của một hành động nào đó. Kiểu sự kiện này được gọi là sự kiện khởi tạo bởi người sử dụng, và ta phải tạo Code cho nó.



Các phương thức thông dụng gồm:

Sự kiện	Xảy ra khi
Change	Người sử dụng sửa đổi chuỗi ký tự trong textbox, combobox...
Click	Người sử dụng bấm chuột lên đối tượng
Dbclick	Người sử dụng bấm chuột đúp lên đối tượng
Dragdrop	Người sử dụng kéo rê một đối tượng sang vị trí khác
Dragover	Người sử dụng kéo rê một đối tượng ngang qua một điều khiển khác
Gotfocus	Đưa ra một đối tượng vào tầm ngắm của người sử dụng
KeyDown	Người sử dụng nhấn một nút trên bàn phím trong khi đối tượng này đang trong tầm ngắm
KeyPress	Người sử dụng nhấn và thả một phím trên bàn phím trong khi đối tượng này đang trong tầm ngắm

Khi tạo một chương trình bằng VB, ta lập trình chủ yếu theo sự kiện. Điều này có nghĩa là chương trình chỉ thi hành khi người sử dụng thực hiện một thao tác trên giao diện hoặc có việc gì đó xảy ra trong hệ điều hành Windows.

Khi có sự kiện xảy ra Windows sẽ gửi một thông điệp tới ứng dụng. Ứng dụng đọc thông điệp và thi hành đoạn code đáp ứng sự kiện đó. Nếu không có đoạn code xử lý thì ứng dụng bỏ qua sự kiện này. VB sẽ tự động phát sinh các thủ tục xử lý sự kiện ngay khi ta chọn tên sự kiện trong cửa sổ code.

## 2. Khái quát về xây dựng một ứng dụng bằng VB 6.0

### a) Những quy định khi viết chương trình

Để chương trình dễ đọc, dễ hiểu và dễ bảo trì, ta cần tuân thủ các bước sau:

- Đặt tên đối tượng biến, hằng và các thủ tục.
- Định dạng chuẩn cho các tiêu đề và chú thích trong chương trình.
- Các khoảng trắng, định dạng và giống hàng trong chương trình.

#### • Quy định đặt tên

##### - Tên và khai báo:

Khi đặt tên ta phải tuân thủ:

- + Tên phải bắt đầu bằng một ký tự.

+ Tên không được chứa dấu chấm hoặc ký tự đặc biệt dùng cho khai báo dữ liệu.

+ Tên không được dài quá 255 ký tự. Tên của điều khiển, biểu mẫu, lớp và module không quá 40 ký tự.

+ Tên không được trùng với từ khóa hoặc hàm hoặc toán tử dùng trong VB.

Chiều dài tên không nên quá 32 ký tự vì nó có thể khó đọc trên màn hình VGA.

Trong trường hợp biểu mẫu hay điều khiển trùng tên với từ khóa ta phải chỉ rõ điều khiển này thuộc biểu mẫu nào hoặc dùng dấu ngoặc vuông [].

Để yêu cầu phải khai báo tất cả các biến trước khi sử dụng ta vào trong Editor Tab của Option Dialog, đánh dấu lựa chọn Require Variable Declaration hoặc gõ lệnh Option Explicit trong chương trình.

Định nghĩa biến và hằng nên nhóm vào một nơi thay vì chứa rời rạc trong nhiều tập tin. Các hằng tổng quát nên gom lại trong một module riêng để phân biệt với các khai báo khác của ứng dụng .

#### - Tên biến và hằng

Ta dùng cụm từ đầu để đặt tên cho các hằng và biến theo phạm vi hoạt động của hằng và biến.

Phạm vi hoạt động	Cụm từ đầu	Ví dụ
Toàn cục (Global,Public)	G	gstrUserName
Module	M	mblnCalCInProgress
Cục bộ (Private)	P	dblVelocity

Biến/hằng toàn cục là biến được khai báo Public trong Module chuẩn hoặc Module của biểu mẫu.

- Biến/hằng module là biến được khai báo Private trong Module chuẩn hoặc Module của biểu mẫu.

Tên nên dùng ký tự in hoa và ký tự in thường để diễn tả mục đích sử dụng. Một từ phải bắt đầu bằng ký tự in hoa.

Ví dụ: mintUserListMax: Số phần tử tối đa trong danh sách người sử dụng (giá trị integer, phạm vi Module); gstrNewLine: ký tự sang hàng mới (giá trị chuỗi, phạm vi toàn cục).

Kế sau cụm từ miêu tả phạm vi hoạt động là cụm từ miêu tả kiểu dữ liệu.

Kiểu dữ liệu	Cụm từ	Ví dụ
Boolean	bln	blnFound
Byte	byt	bytRasterdata
Đối tượng Collection	col	colWidget
Currentcy	Cur	Currevenue
Date(time)	dtm	dtmStart
Double	dbl	dblTolerance
Error	err	errOrderNum
Integer	Int	intQuantity
Đối tượng	obj	objCurrent
Single	sng	sngAverage
String	Str	strFileName
Kiểu tự định nghĩa	udt	udtEmployee
Variant	vnt	vntChecksum

#### - Đặt tên cho hàm và thủ tục

Đặt tên cho hàm, thủ tục nên bắt đầu là động từ như IniNameArray hoặc CloseDialog

#### - Đặt tên cho đối tượng

Ta nên dùng cụm từ đứng đầu trong danh sách dưới đây để đặt tên cho các điều khiển.

Kiểu điều khiển	Cụm từ đứng đầu	Ví dụ
3D Panel	pnl	pnlGroup
ADO Data	ado	adoBiblio
Animated button	Ani	aniMailbox
Checkbox	chk	chkReadOnly
Combo box, drop-downlist box	cbo	cboEnglish
Command button	cmd	cmdExit
Common Dialog	dlg	dlgFileOpen
Communications	com	comFax
Control (used within procedures when the specific type is unknown)	ctr	ctrCurrent
Data	dat	datBiblio
Data-bound combo box	Dbcbo	dbcboLanguage

Data-bound grid	dbgrd	dbgrdQueryResult
Data combo	dbc	dbcAuthor
Data grid	dgd	dgdTitles
Data list	dbl	dblPublisher
Data repeater	drp	drpLocation
Data picker	dtp	Dtp Published
Directory list box	drv	drvTarget
File list box	fil	filSource
Flat scroll bar	fsb	fsbMove
Form	frm	frmEntry
Frame	fra	FraLanguage
Gauge	gau	gauStatus
Graph	Gra	graRevenue
Grid	grd	grdPrices
Hierarchical flexgrid	flex	flexOrders
Horizontal scxroll bar	hsb	Hsb Volume
Image	img	imgIcon
Image combo	imgcbo	imgcboProduct
Image list	ils	ilsAllIcon
Label	lbl	lblHelpMessage
Lightweight check box	lwchk	lwchkArchive
Lightweight combo box	lwcbo	lwcboGerman
Lightweight command button	lwcmd	lwcmdRemove
Lightweight frame	lwfra	lwfraSaveOptions
Lightweight Horizontal scxroll bar	lwhsb	lwhsbVolume
Lightweight list box	lwlst	lwlstCostCenter
Lightweight Option Button	lwopt	lwoptIncomeLevel
Lightweight text box	lwtxt	lwtxtStreet
Lightweight vertical scroll bar	lwvsb	lwvsbVolume
Line	lin	linVertical
List box	lst	lstPolicy code

List View	lvw	lvwHeadins
MAPI message	mpm	Mpmsent message
MAPI session	mps	Mps session
MCI	mci	mciVideo
Menu	mnu	mnuFileOpen
Month View	mvw	mvwPeriod
MS chart	ch	chSalebyRegion
MS Flex grid	msg	msgClient
MS Tab	mst	mstFirst
OLE Container	ole	oleWorksheet
Opition button	opt	optGender
Picture box	Pic	picVGA
Picture clip	clp	clpToolbar
ProgressBar	prg	prgLoadFile
Remote Data	rd	rdTitles
RichTextBox	Rtf	rtfReport
Shape	shp	shpCircle
Slider	sld	sldScale
Spin	spn	spnPages
Status Bar	sta	StaDate Time
SysInfo	Sys	Sys Monitor
Text box	Txt	txtlastName
Timer	tmr	tmrAlarm
Toolbar	tlb	tlbActions
TreeView	tre	treOrganization
Updown	upd	updDirection
Vertical Scroll bar	vsb	vsbRate

- **Quy ước về cấu trúc lập trình**

- **Soạn thảo mã nguồn**

\* Ta luôn dùng toán tử '&' để ghép chuỗi ký tự và toán tử '+' để làm việc với dữ liệu kiểu số. Nếu ta dùng toán tử '+' để ghép ký tự thì nhiều trường hợp không cho

kết quả mong muốn.

- \* Ta sử dụng Tab hoặc khoảng trắng để lùi đầu dòng tạo phân biệt các khối với nhau.
- \* Các dòng chương trình theo sau câu chú thích nên thụt vào một Tab. Các khối con thụt vào một Tab nữa.

- \* Bắt buộc sử dụng Option Explicit ở mỗi bắt đầu các Module.

Tất cả các thủ tục và hàm nên bắt đầu bằng một đoạn chú thích ngắn gọn, giải thích chức năng của nó, mô tả tham số truyền vào thủ tục/hàm và giá trị trả về đặc biệt là các tham chiếu.

Thành phần chú thích	Nội dung chú thích
Mục đích	Thủ tục/hàm thực hiện chức năng gì.
Điều kiện	Danh sách các biến bên ngoài, các điều khiển, tập tin hoặc các thành phần khác cần nó trước khi gọi thủ tục/hàm này.
Hiệu quả	Danh sách các biến ngoài, các điều khiển, tập tin hoặc các thành phần khác sẽ bị ảnh hưởng khi gọi thủ tục/hàm.
Đầu vào	Các tham số truyền vào thủ tục/hàm. Mỗi tham số nên đặt trên một dòng và chú thích riêng cho từng tham số.
Trả về	Giải thích giá trị trả về của hàm.

**Ví Dụ:**

‘Purpose: locates the first occurrence of a

‘ Specefid user in the userList array.

‘ Inputs

‘strUserList(): the list of users to be searched.

‘strTargetUser : the name of the user to search for.

‘ Return: The index of the first occurrence of the

‘ rsTargetUser in the rasUserList array.

‘ If target user is not found, return – 1.

\*\*\*\*\*

Function intFindUser (strUserList()As string, str TargetUser As\_String) As Integer

Dim I As Integer ‘Loop counter

Dim blnFound As Integer ‘Target Found flag.

intFindUser = -1

```

i = 0
While i <= Ubound(strUserList) And not blnFound
If strUserList(i) = strTargetUser Then
blnFound = True
intFindUser = i
End If
i = i + 1
Wend
End Function

```

Trong phần đầu của Module. bas có chứa các khai báo hằng toàn cục cho hệ thống, ta cần thêm đoạn chú thích mô tả ứng dụng, liệt kê các đối tượng, dữ liệu chính, các thủ tục giải thuật, hộp thoại, cơ sở dữ liệu và các thành phần liên quan đến hệ thống. Ta có thể thêm một đoạn chương trình mô phỏng giải thuật nếu cần.

#### - Tạo chuỗi String cho MsgBox, Inputbox và câu truy vấn

Khi cần phải tạo các sô ký tự dài ta sử dụng ký tự gạch dưới để nối thành nhiều dòng. Điều này giúp ta dễ đọc và Debug mã nguồn các sô ký tự dễ dàng hơn. Kỹ thuật này đặc biệt hữu dụng cho các trường hợp sử dụng Message Box, Input Box và tạo câu truy vấn SQL.

#### Ví dụ :

```

Dim Msg as String
Msg= " This is paragraph that will be" _
& " in a message box. This text is" _
& " broken into several lines of code" _
& " in the source code, making it easier" _
& " for the programmer to read and debug."
MsgBox Msg

```

#### Hoặc

```

Dim QRY as String
QRY= " SELSECT *" _
& "FROM Tiltes" _
& " WHERE [Year Published]>1988"
TitleQry.SQL= QRY

```

**- Font chữ và layout**

Vì ta sử dụng tiếng Việt theo mã TCVN3 nên toàn bộ hệ thống ta sử dụng Font VK Sans Serif cỡ chữ 8 với mọi Item. Loại trừ Caption của Frame phải là chữ đậm có màu xanh đậm. Màn hình hiển thị ở chế độ 600x800. Các nút bấm đặt ở cuối bên phải của form, các nút cách nhau 80 bit.

**- Dạng cấu trúc trong mỗi thủ tục, Function****Đối với thủ tục:**

On Error Go To ErrHandle:

    'code

On Error Go To 0

Exit Sub

ErrHandle:

    'code

On Error Go To 0

**Đối với Function:**

On Error Go To ErrHandle:

    'code

On Error Go To 0

Exit Function

ErrHandle:

Errstr= Str(Err.Number)&"."& Err.Description

ErrStr= Errstr & Chr(13) Err.Source

MsgBox ErrStr", " Lỗi

On Error Go To 0

**b) Các bước xây dựng một ứng dụng bằng VB 6.0**

Thế mạnh của VB là sử dụng các điều khiển và tận dụng tối đa khả năng lập trình của chúng. Một điều khiển thực chất là một cửa sổ được lập trình sẵn bên trong. Một điều khiển chứa đựng một chương trình được lập sẵn và chương trình này có thể thích hợp một cách dễ dàng vào ứng dụng có sử dụng điều khiển.

Để xây dựng một ứng dụng bằng VB ta cần làm các bước sau:

**B1:** Thiết kế giao diện ứng dụng bao gồm:

- Thiết kế các menu cần dùng.



- Thiết kế các cửa sổ, thiết lập kích thước và các đặc tính khác của chúng.
- Thiết lập các Control trên form: các nút lệnh, các Label, các textbox...sẵn có do VB hỗ trợ và có thể là các Control do các hãng thứ 3 cung cấp giúp ta thiết kế để thực hiện một số chức năng mong muốn.

**B2:** Viết mã kích hoạt các giao diện đã thiết kế ở B1 và các mã đáp ứng các sự kiện mà người sử dụng tác động lên giao diện. Đó là điểm mấu chốt cần phải nhận thức rõ trong khâu lập trình VB. Visual Basic xử lý mã chỉ để đáp ứng với các sự kiện.

Chương trình Visual Basic được xem như gồm một loạt các phần độc lập chỉ “thức giấc” để đáp ứng các sự kiện mà ta báo cho chúng biết trước. Bởi vậy các dòng mã thi hành trong một chương trình Visual Basic phải nằm trong các thủ tục hoặc các hàm. Các dòng mã bị cô lập sẽ không làm việc.

**B3:** Chạy ứng dụng, tìm lỗi và chỉnh sửa.

## CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

### 2.1. Khảo sát hệ thống

Thư viện sách là nơi dành cho học viên, sinh viên cũng như cán bộ công nhân viên, giảng viên có nhu cầu về việc tìm hiểu các thông tin phục vụ cho việc học tập, giảng dạy từ căn bản tới chuyên sâu có thể tìm được cho mình các thể loại sách phù hợp với nhu cầu và đúng chuyên môn của mình. Yêu cầu quản lý về việc nhập sách của thư viện đã rất khó, còn thêm phải quản lý việc cho mượn sách hằng ngày đối với nhiều độc giả. Vì vậy cần một chương trình quản lý để giúp thủ thư dễ dàng hơn trong công việc của mình.

Các công việc chính của thư viện gồm:

- Nhập sách từ các nhà cung cấp.
- Cho mượn sách khi độc giả có nhu cầu mượn.
- Nhận lại sách khi độc giả trả sách và xử lý sự cố trong quá trình trả sách.
- Báo cáo thống kê tình hình nhập và mượn trả sách của thư viện.

### 2.2. Phân tích hiện trạng hệ thống

#### 2.2.1. Nhập sách

Sau khi thu thập đầy đủ thông tin cung cấp tài liệu, việc tiến hành lựa chọn nhà cung cấp tài liệu phù hợp cần dựa trên danh sách các nhà cung cấp đã được phê duyệt. Trong trường hợp các nhà cung cấp trong danh sách được phê duyệt không có tài liệu cần thiết, thủ thư cần tiến hành việc thu thập các thông tin về các nhà cung cấp khác, đánh giá so sánh các yếu tố về chất lượng, giá cả và thời gian của các nhà cung cấp này, đề nghị lãnh đạo quyết định về việc lựa chọn nhà cung cấp phù hợp. Nhà cung cấp được lựa chọn sẽ được ghi nhận lại và cập nhật vào danh sách nhà cung cấp được phê duyệt hàng năm, nếu thấy cần thiết cho việc mua tài liệu thường xuyên. Sách được nhập về thì cấp mã số, mã số này ghi trong sổ và bìa sách. Đồng thời thủ thư làm "loan card" ghi thông tin của từng sách để lưu giữ.

#### 2.2.2. Mượn sách

Khi độc giả có nhu cầu mượn sách tại thư viện thì phải xuất trình thẻ độc giả cho thủ thư. Khi thủ thư kiểm tra thẻ để xác minh là đúng thì mới tiến hành kiểm tra sách mượn còn hay không, nếu còn thì ghi và đưa phiếu mượn đồng thời ghi thông tin mượn sách vào loan card, ngược lại thì thông báo sách đã hết.

### 2.2.3. Trả sách

Khi độc giả có nhu cầu trả sách thì thủ thư phải kiểm tra sách và thông tin phiếu mượn, nếu không xảy ra sự cố gì thì nhận lại sách và ghi thông tin trả vào loan card đồng thời trả lại thẻ thư viện. Ngược lại sách nhận lại bị rách hoặc hư hỏng, sách mất... Tùy theo tình trạng của sách mà thủ thư lập biên bản phạt và ghi phiếu thu. Độc giả phải chấp hành theo xử lý của thủ thư.

### 2.2.4. Báo cáo thống kê

Sau từng quý hoặc thời gian cụ thể thủ thư phải làm báo cáo về tình hình mượn và trả của thư viện cho lãnh đạo trung tâm biết để có thông tin điều chỉnh sao cho phù hợp với tình hình chung của thư viện.

## 2.3. Mô tả hoạt động của thư viện

- Khi thư viện có nhu cầu nhập sách (các loại sách phù hợp với tình hình học tập cũng như giảng dạy), thủ thư gửi thông tin về sách cần mua với nhà cung cấp, nhà cung cấp sẽ gửi bảng báo giá, sau đó thì thủ thư sẽ **lập và gửi phiếu đặt sách** tới nhà cung cấp. Nhà cung cấp sẽ **bàn giao sách** cho trung tâm kèm theo phiếu giao sách, sau khi thủ thư **kiểm tra sách** đúng yêu cầu của trung tâm, thủ thư sẽ **lập phiếu nhập sách** và tiến hành nhập sách vào thư viện đồng thời nhà cung cấp ký vào phiếu nhập sách. Ngược lại **thông báo trả lại sách sai yêu cầu**. Các thông tin nhập sách được **ghi sổ nhập sách**. Sau đó thủ thư sẽ **gửi phiếu nhập sách** cho kế toán để thanh toán cho nhà cung cấp.

- Khi độc giả có nhu cầu mượn sách, độc giả phải xuất trình thẻ thư viện, **gửi yêu cầu mượn sách**. Thủ thư **kiểm tra thẻ thư viện**, nếu thấy thẻ không hợp lệ thì thông báo thẻ thư viện sai ngay cho độc giả. Nếu đúng thì thủ thư phải **kiểm tra sách mượn còn hay không**, nếu không còn sách thì **thông báo sách hết** cho độc giả. Ngược lại thì **ghi và đưa phiếu mượn** cho độc giả. Thông tin mượn sách được **ghi vào “loan card”**.

- Khi độc giả có **yêu cầu trả sách** thì thủ thư phải **đối chiếu thông tin sách mượn và phiếu mượn**, nếu không xảy ra sự cố gì thì nhận lại sách và **trả lại thẻ thư viện** cho độc giả đồng thời **ghi thông tin sách trả vào loan card**. Ngược lại **thông báo tình trạng sách trả** cho độc giả: sách nhận lại bị rách hoặc hư hỏng, sách mất... Tùy theo tình trạng của sách mà thủ thư **lập biên bản sự cố** và căn cứ tình trạng của sách để xử lý đồng thời **lập phiếu thu** để độc giả nộp tiền phạt. Thông tin sự cố

được **ghi sổ sự cố**. Độc giả phải chấp hành theo xử lý của thủ thư.

- Sau từng quý hoặc thời gian cụ thể thủ thư phải **lập báo cáo thống kê** về tình hình mượn và trả sách tại thư viện cho lãnh đạo trung tâm biết để có thông tin điều chỉnh sao cho phù hợp với tình hình chung của thư viện.

#### 2.4. Bảng nội dung công việc

STT	Tên công việc	Đối tượng thực hiện	HSDL
1.	Lựa chọn nhà cung cấp	Thủ thư	Danh sách nhà cung cấp
2.	Gửi thông tin sách nhập	Thủ thư	
3.	Lập và gửi phiếu đặt sách	Thủ thư Nhà cung cấp	Phiếu đặt sách
4.	Bàn giao sách	Nhà cung cấp	Phiếu giao sách
5.	Kiểm tra sách nhập	Thủ thư Nhà cung cấp	
6.	Trả lại sách sai yêu cầu	Thủ thư Nhà cung cấp	
7.	Lập phiếu nhập sách	Thủ thư Nhà cung cấp	Phiếu nhập sách
8.	Gửi phiếu nhập sách	Thủ thư Kế toán	Phiếu nhập sách
9.	<b>Ghi sổ nhập sách</b>	<b>Thủ thư</b>	<b>Sổ nhập sách</b>
10.	Gửi yêu cầu mượn sách	Độc giả- Thủ thư	
11.	Nhận và kiểm tra thẻ thư viện	Thủ thư- Độc giả	
12.	Thông báo thẻ thư viện hết hạn	Thủ thư- Độc giả	
13.	Kiểm tra còn sách mượn	Thủ thư	
14.	Thông báo hết sách	Thủ thư- Độc giả	
15.	Ghi và đưa phiếu cho mượn sách	Thủ thư Độc giả	Phiếu mượn sách

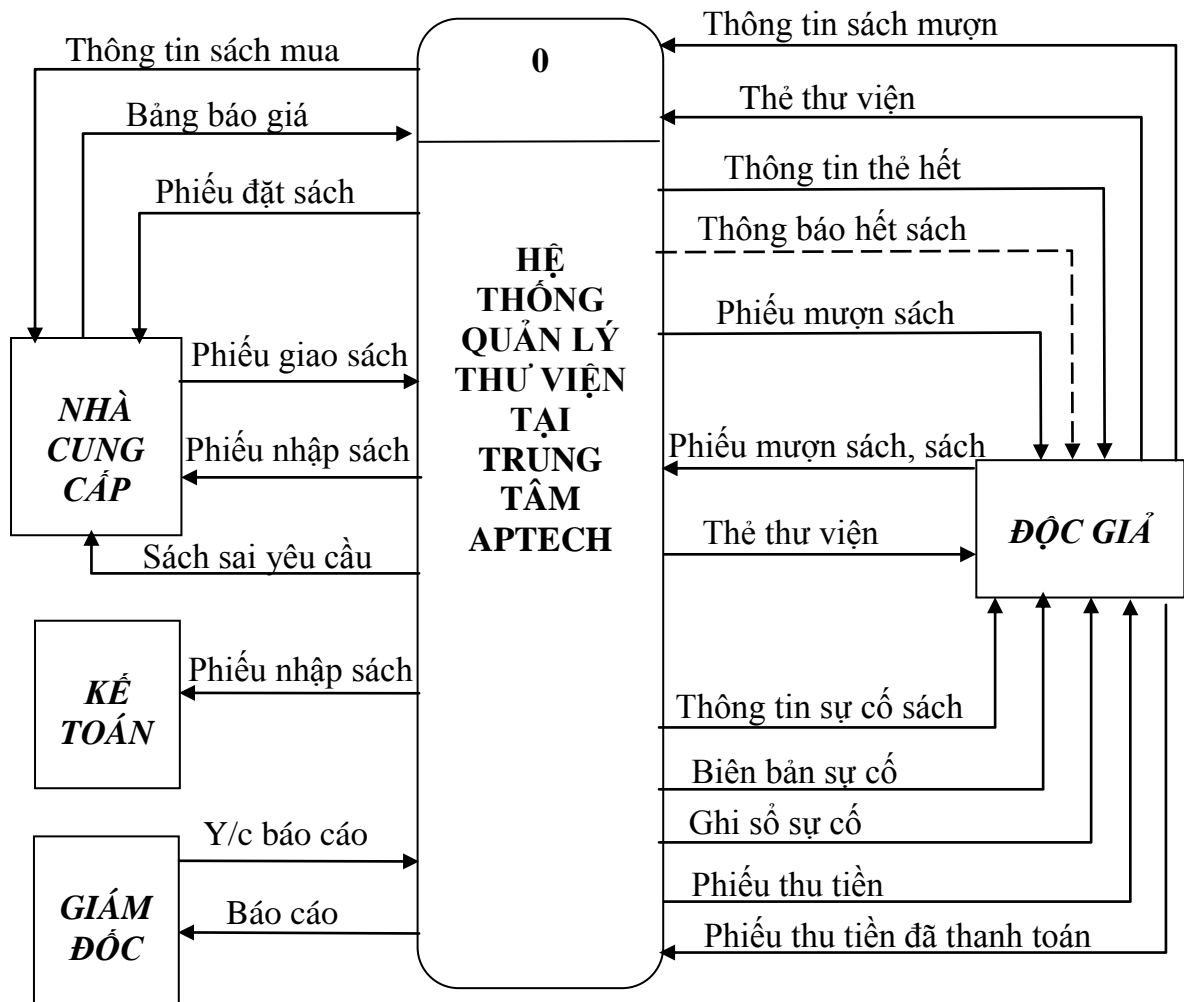
16.	<b><i>Ghi thông tin mượn sách vào loan card</i></b>	<b><i>Thủ thư</i></b>	<b><i>Loan card</i></b>
17.	Thông tin trả sách	Độc giả	
18.	Đối chiếu thông tin phiếu mượn, sách mượn	Thủ thư	Phiếu mượn sách
19.	Thông báo tình trạng sách trả	Thủ thư Độc giả	
20.	Ghi thông tin sách trả vào loan card	Thủ thư	Loan card
21.	Trả lại thẻ thư viện	Thủ thư- Độc giả	
22.	Lập biên bản sự cố	Thủ thư- Độc giả	Biên bản sự cố
23.	Lập phiếu thu	Thủ thư- Độc giả	Phiếu thu
24.	<b><i>Ghi sổ sự cố</i></b>	<b><i>Thủ thư</i></b>	<b><i>Sổ sự cố</i></b>
25.	<b><i>Lập báo cáo thống kê</i></b>	<b><i>Thủ thư</i></b>	<b><i>Bản báo cáo</i></b>

**2.5. Mô hình nghiệp vụ**

**2.5.1. Bảng phân tích xác định các chức năng, tác nhân và hồ sơ**

<b>Động từ + Bổ ngữ</b>	<b>Danh từ</b>	<b>Nhận xét</b>
Lập phiếu đặt sách	Phiếu đặt sách	HSDL
Kiểm tra sách nhập	Thủ thư- Nhà cung cấp	Tác nhân
Trả lại sách nhập sai yêu cầu	Thủ thư- Nhà cung cấp	Tác nhân
Lập phiếu nhập sách	Phiếu nhập sách Thủ thư- Nhà cung cấp	HSDL Tác nhân
Gửi phiếu nhập sách cho kế toán	Thủ thư- Kế toán Phiếu nhập sách	Tác nhân HSDL
Ghi loan card	Loan card	HSDL
<b><i>Ghi sổ nhập sách</i></b>	<b><i>Sổ nhập sách</i></b>	<b><i>HSDL</i></b>
Nhận yêu cầu mượn sách	Độc giả	Tác nhân
Thông báo thẻ thư viện hết hạn	Thủ thư-Độc giả	Tác nhân
Kiểm tra thẻ thư viện	Thủ thư-Độc giả	Tác nhân
Kiểm tra còn sách mượn	Loan card- Thủ thư	HSDL
Thông báo hết sách	Thủ thư-Độc giả	Tác nhân
Ghi phiếu mượn sách	Phiếu mượn sách	HSDL
<b><i>Ghi tt mượn vào loan card</i></b>	<b><i>Loan card</i></b>	<b><i>HSDL</i></b>
Thông tin trả sách	Độc giả-Thủ thư	Tác nhân
Đối chiếu phiếu mượn và sách mượn	Phiếu mượn	HSDL
TB tình trạng sách trả	Thủ thư- Độc giả	Tác nhân
Trả lại thẻ thư viện	Thủ thư- Độc giả	Tác nhân
<b><i>Ghi tt sách trả vào loan card</i></b>	<b><i>Loan card</i></b>	<b><i>HSDL</i></b>
Lập biên bản sự cố	Biên bản sự cố	HSDL
Ghi phiếu thu	Phiếu thu	HSDL
<b><i>Ghi sổ sự cố</i></b>	<b><i>Sổ sự cố</i></b>	<b><i>HSDL</i></b>
<b><i>Lập báo cáo</i></b>	<b><i>Bản báo cáo</i></b>	<b><i>HSDL</i></b>

**2.5.2. Biểu đồ ngữ cảnh**



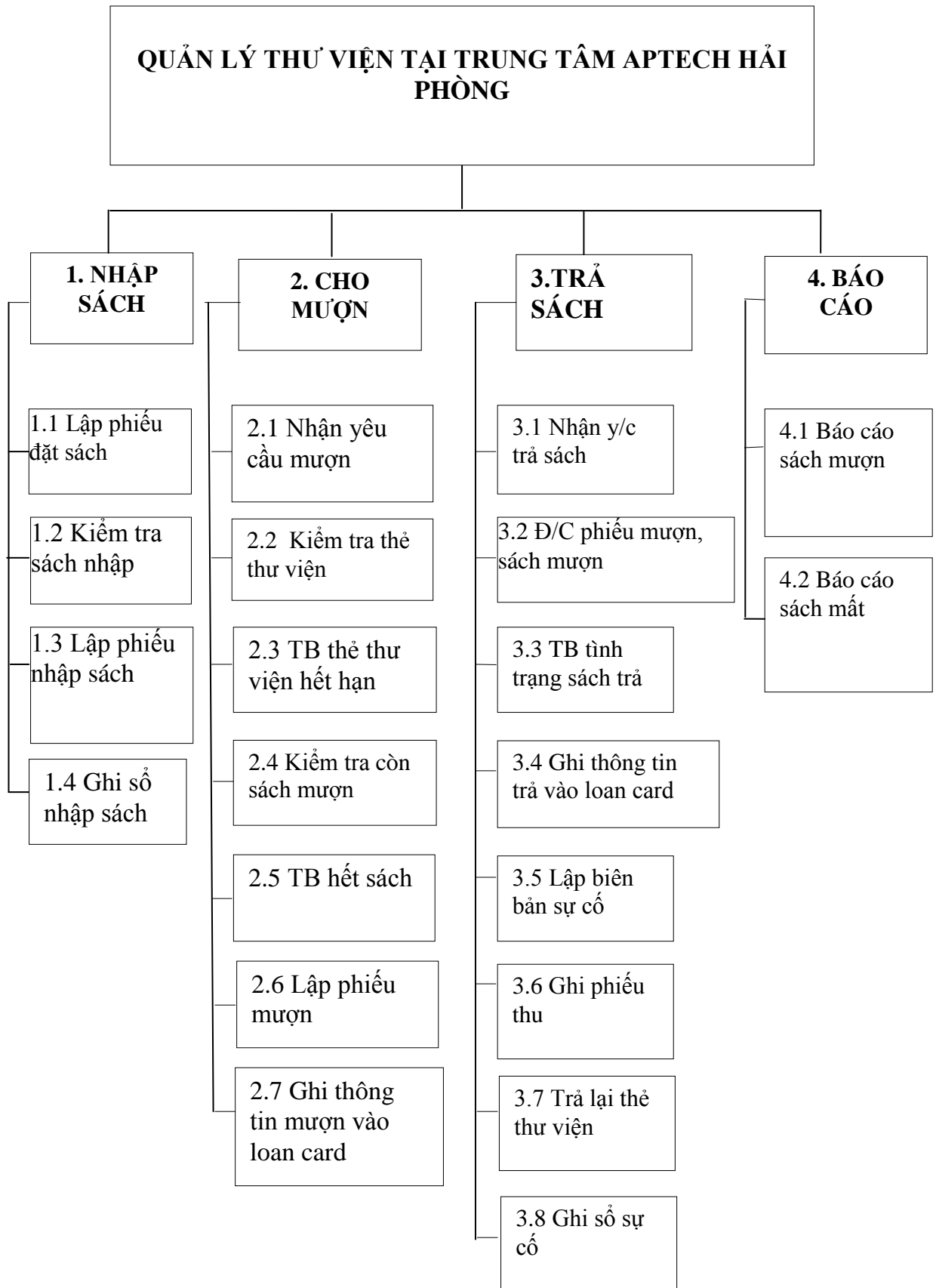
**2.5.3. Nhóm dần các chức năng**

Các chức năng chi tiết (lá)	Nhóm lần 1	Nhóm lần 2
Gửi phiếu đặt sách	<i>Nhập sách</i>	<b>Hệ thống quản lý thư viện tại trung tâm APTECH HẢI PHÒNG</b>
Kiểm tra sách nhập		
Lập phiếu nhập sách		
Ghi sổ nhập sách		
Nhận yêu cầu mượn sách	<i>Cho mượn</i>	
Kiểm tra thẻ thư viện		
TB thẻ thư viện hết hạn sử dụng		
Kiểm tra còn sách mượn		
Thông báo hết sách		
Lập phiếu mượn		
Ghi thông tin mượn vào loan card		
Nhận y/c trả sách	<i>Trả sách</i>	
Đôi chiếu thông tin phiếu mượn, sách mượn		
TB tình trạng sách trả		
Ghi thông tin trả vào loan card		
Trả lại thẻ thư viện		
Lập biên bản sự cố		
Ghi phiếu thu		
Ghi sổ sự cố		
Báo cáo sách mượn	<i>Báo cáo</i>	
Báo cáo sách mất		



**2.5.4. Sơ đồ phân rã chức năng**

**1. Sơ đồ**



**Mô tả chi tiết các chức năng lá****\* Nhập sách**

- Lập phiếu đặt sách: sau khi nhà cung cấp đã gửi trung tâm thông tin chi tiết về sách cần nhập, nếu không có vấn đề gì thì trung tâm sẽ lập và gửi phiếu đặt sách đến nhà cung cấp.

- Kiểm tra sách: thủ thư tiến hành đối chiếu sách nhập với phiếu đặt sách xem có đề gì về số lượng sách nhập cũng như chất lượng của sách.

- Lập và gửi phiếu nhập sách: khi nhà cung cấp chuyển sách, thủ thư sẽ lập phiếu nhập sách và tiến hành nhập sách vào thư viện. Sau đó thủ thư sẽ gửi phiếu nhập sách cho kế toán tiến hành thanh toán với nhà cung cấp.

- Ghi sổ nhập sách: thủ thư ghi sổ nhập sách khi trung tâm nhập sách từ nhà cung cấp.

**\* Cho mượn**

- Nhận yêu cầu mượn sách: khi độc giả có nhu cầu mượn sách sẽ gửi yêu cầu mượn sách cho thủ thư.

- Nhận và kiểm tra thẻ thư viện: khi độc giả có nhu cầu mượn sách phải xuất trình thẻ thư viện, thủ thư sẽ tiến hành kiểm tra thư viện là đúng hay không.

- Kiểm tra còn sách mượn: thủ thư kiểm tra sách mượn có còn hay không, nếu còn thì ghi phiếu mượn, ngược lại thông báo hết sách cho độc giả.

- Lập và đưa phiếu mượn: thủ thư sẽ ghi thông tin phiếu mượn, đồng thời đưa cho độc giả phiếu mượn.

- Ghi thông tin mượn sách vào loan card.

**\* Trả sách**

- Nhận yêu cầu trả sách: khi độc giả có nhu cầu trả sách sẽ gửi yêu cầu trả sách cho thủ thư

- Đối chiếu thông tin phiếu mượn, sách mượn: khi độc giả đến trả sách, thủ thư đối chiếu phiếu mượn và sách mượn xem có đúng hay không. Nếu không đúng thì sẽ thông báo sự cố cho độc giả. Ngược lại sẽ trả thẻ thư viện cho độc giả đồng thời ghi thông tin trả sách vào loan card.

- Thông báo tình trạng sách trả: sau khi kiểm tra sách trả, thủ thư phải thông báo cho độc giả biết tình trạng của sách để độc giả biết và chịu sự xử lý của thủ thư.

- Ghi thông tin trả vào loan card: thủ thư sẽ ghi thông tin trả sách vào loan card.

- Trả lại thẻ thư viện: sau khi nhận lại sách sẽ trả lại thẻ thư viện cho độc giả.
- Lập biên bản sự cố: căn cứ vào tình trạng của sách để ghi biên bản sự cố.
- Ghi phiếu thu: quy định số tiền phải nộp của độc giả.
- Ghi sổ sự cố: thủ thư ghi các thông tin về sự cố trong sổ sự cố.

**\* Báo cáo thống kê**

- Lập báo cáo: thủ thư tiến hành thống kê tình hình mượn trả sách để lập báo cáo với lãnh đạo trung tâm.

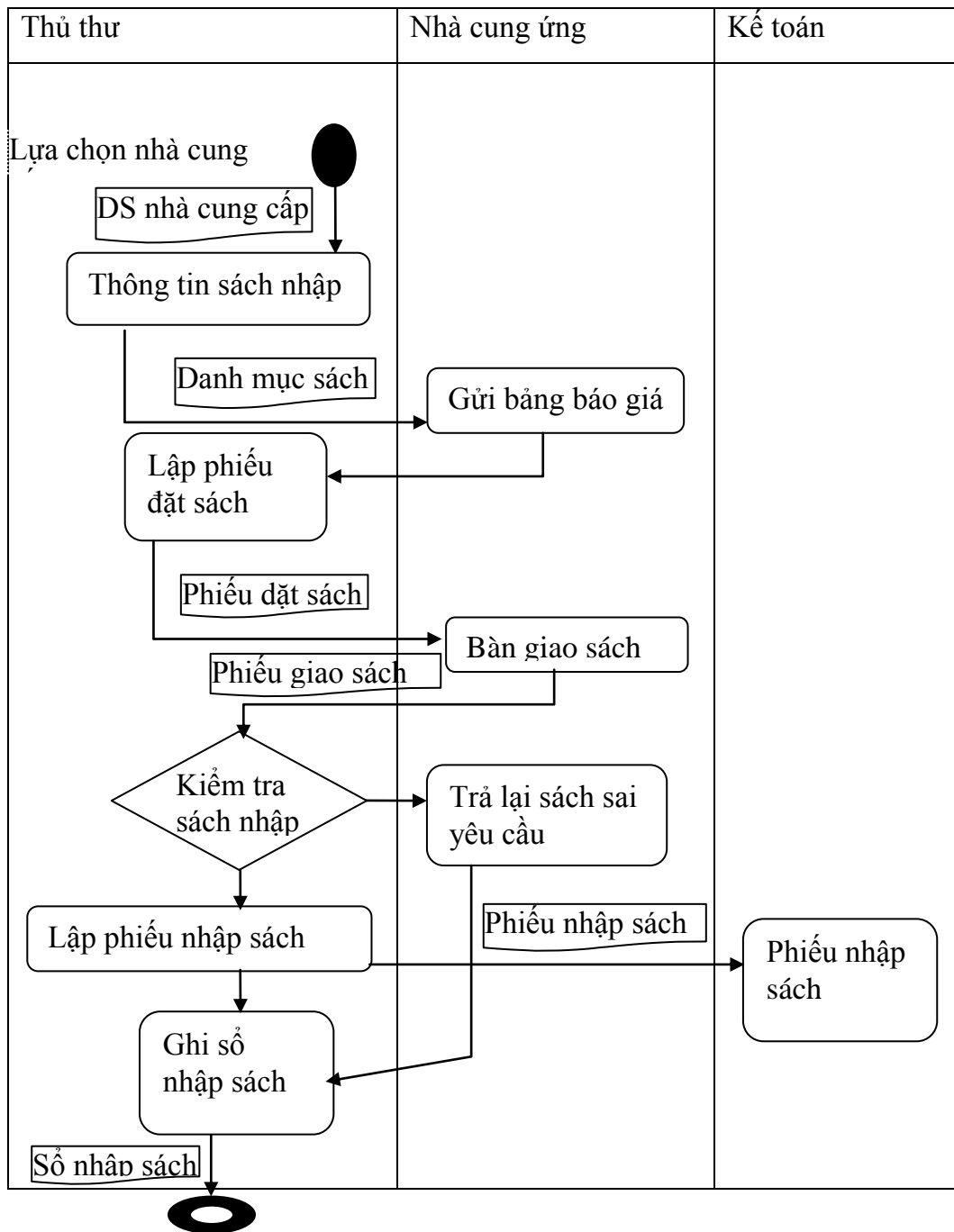
**2.5.5. Hồ sơ dữ liệu sử dụng**

- a. Danh mục sách
- b. Danh sách nhà cung cấp
- c. Phiếu đặt sách
- d. Phiếu giao sách
- e. Phiếu nhập sách
- f. Phiếu mượn sách
- g. Loan card
- h. Phiếu thu
- i. Biên bản sự cố
- j. Sổ nhập sách
- k. Sổ sự cố
- l. Báo cáo

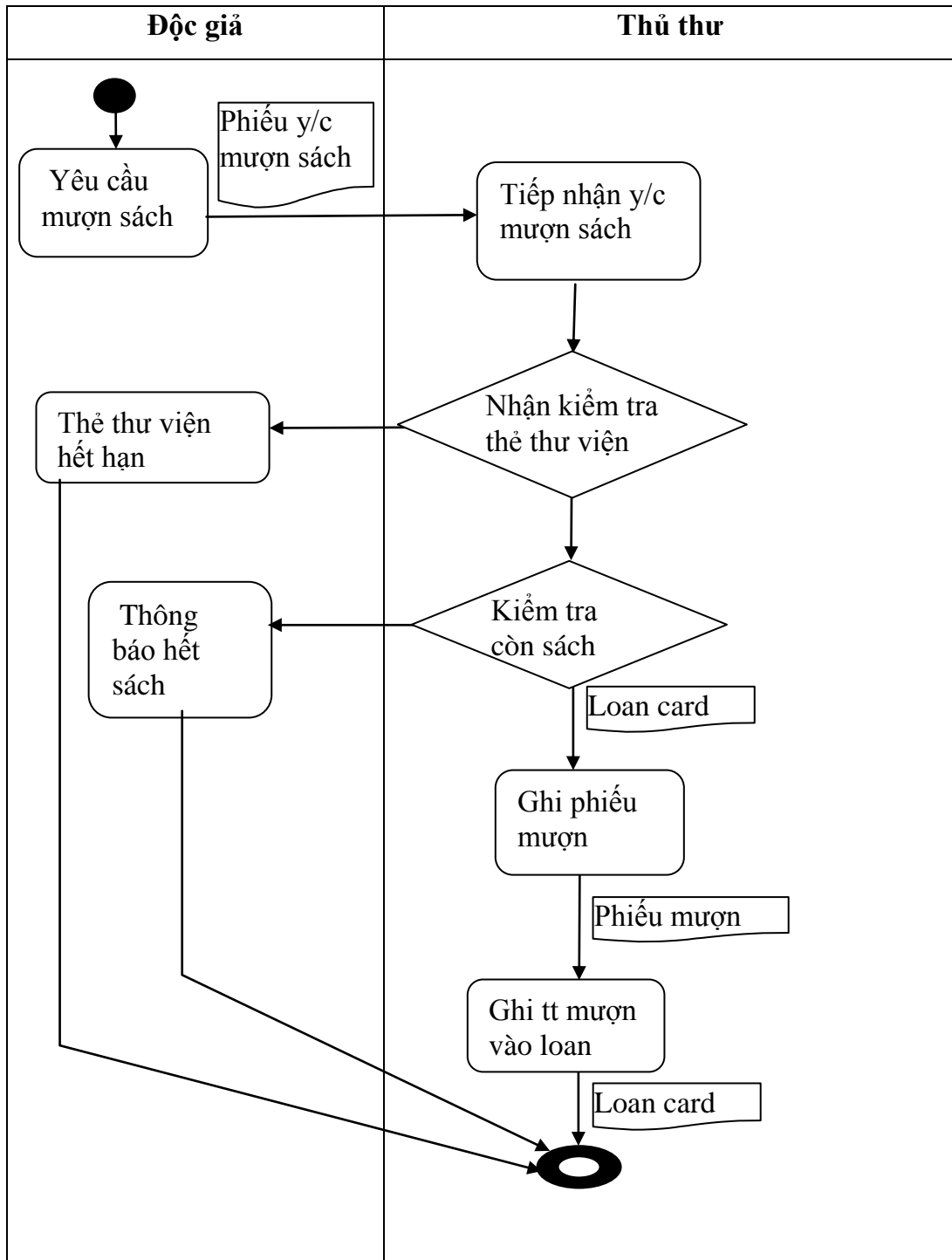
**2.5.6. Ma trận thực thể chức năng**

<b>Các hồ sơ dữ liệu</b>												
a. Danh mục sách												
b. Danh sách nhà cung cấp												
c. Phiếu đặt sách												
d. Phiếu giao sách												
e. Phiếu nhập sách												
f. Phiếu mượn sách												
g. Loan card												
h. Phiếu thu												
i. Biên bản sự cố												
j. Sổ nhập sách												
k. Sổ sự cố												
l. Báo cáo												
<b>Các chức năng nghiệp vụ</b>	<b>a</b>	<b>b</b>	<b>c</b>	<b>d</b>	<b>e</b>	<b>f</b>	<b>g</b>	<b>h</b>	<b>i</b>	<b>j</b>	<b>k</b>	<b>l</b>
1. Nhập sách	U	U	C	R	C		U			U		
2. Mượn sách	R					C	U					
3. Trả sách						R	U	C	C		U	
4. Báo cáo							R		R		R	C

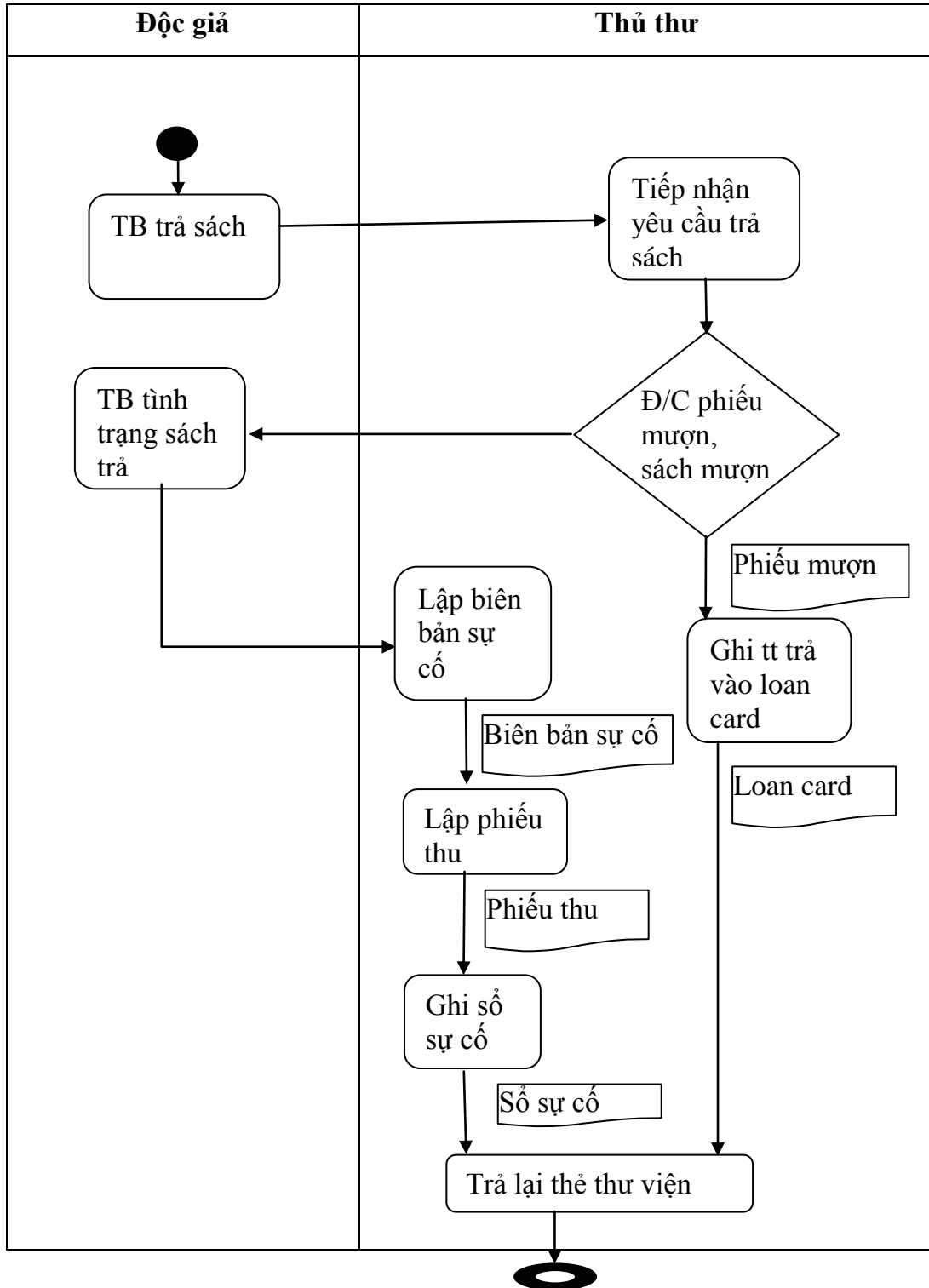
**2.5.7. Biểu đồ hoạt động của từng nghiệp vụ**  
**1. Biểu đồ hoạt động của nghiệp vụ nhập sách**



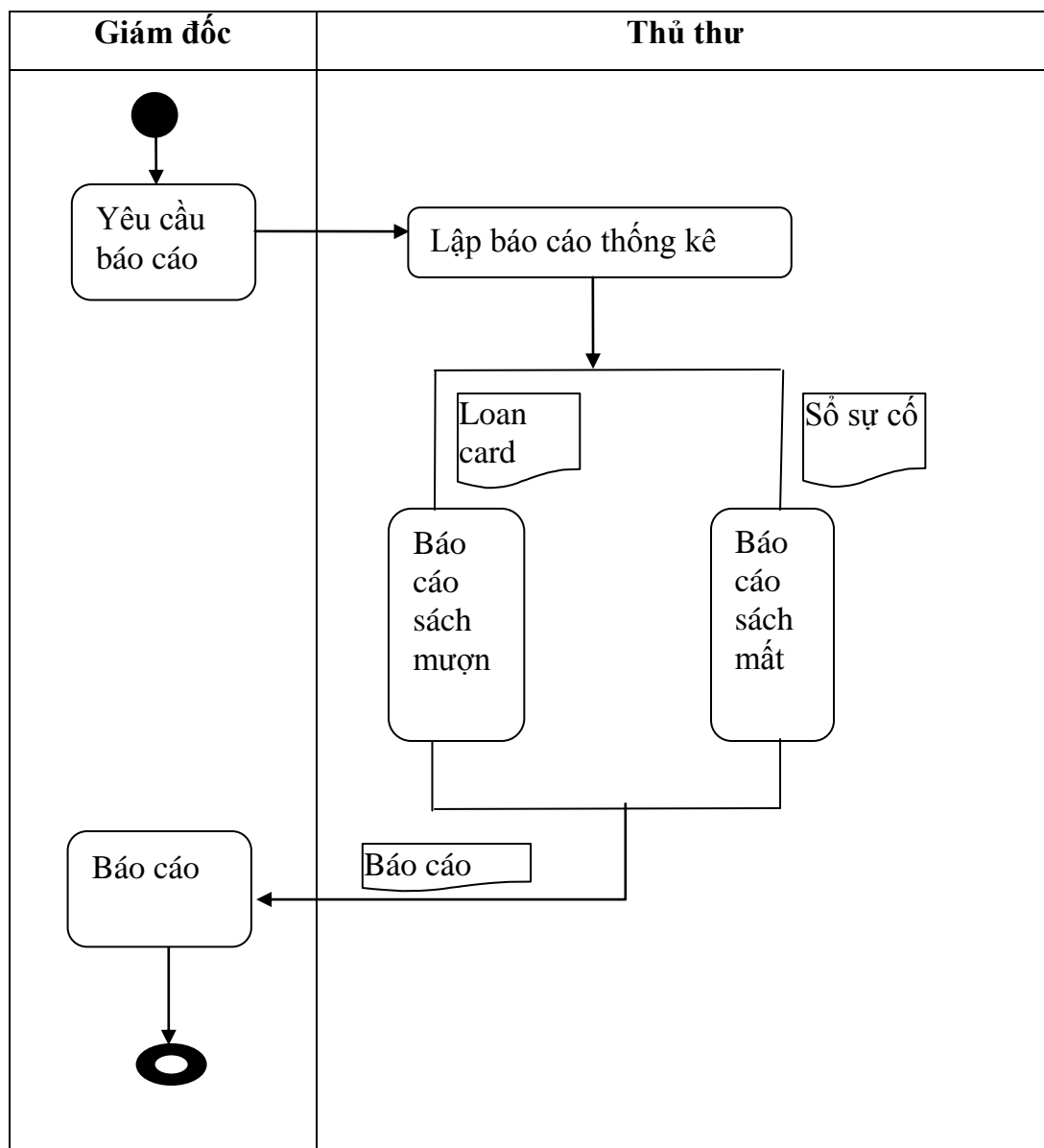
**2. Biểu đồ hoạt động của nghiệp vụ mượn sách**



**3. Biểu đồ hoạt động của nghiệp vụ trả sách**



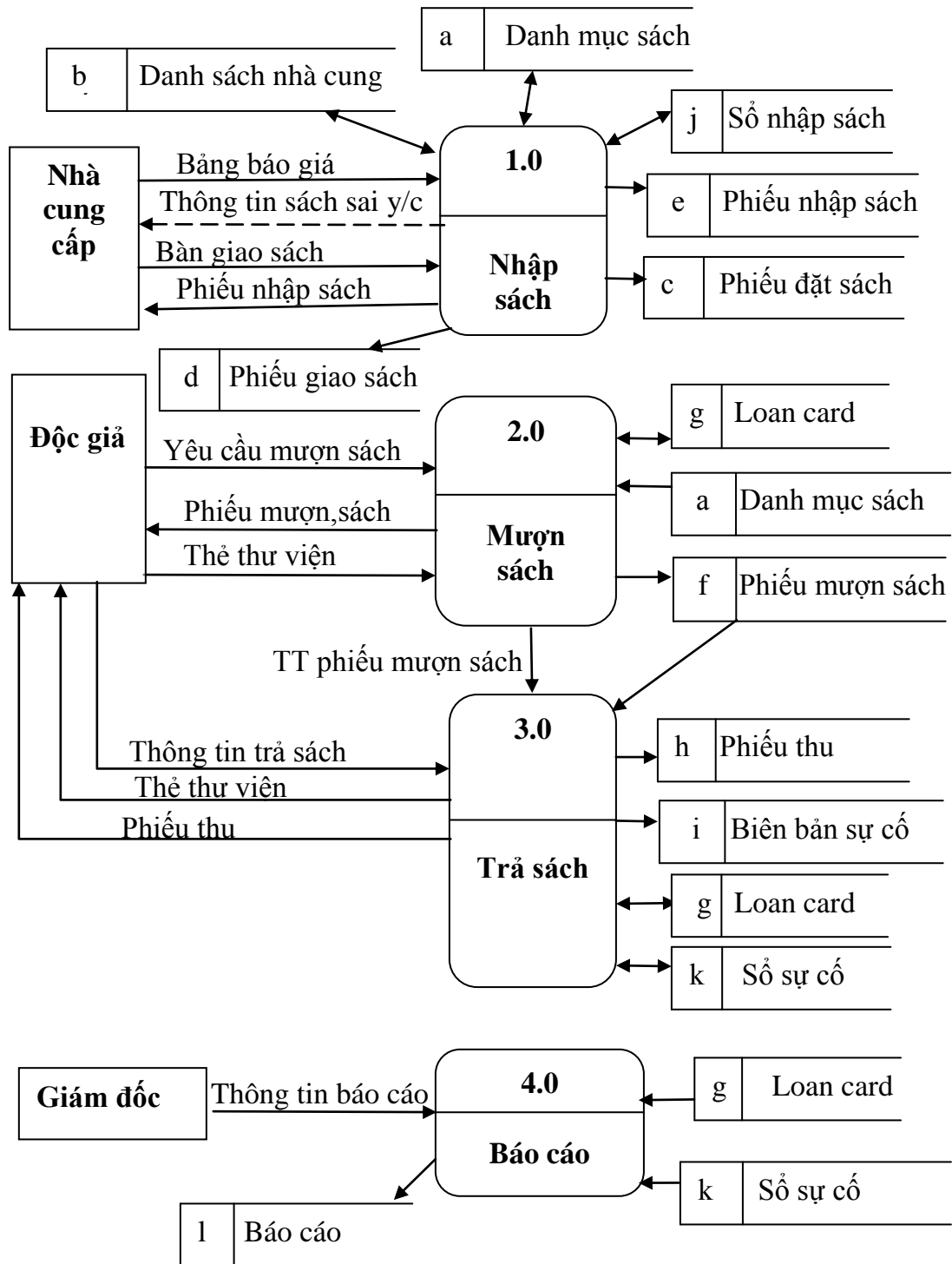
**4. Biểu đồ hoạt động của nghiệp vụ báo cáo**





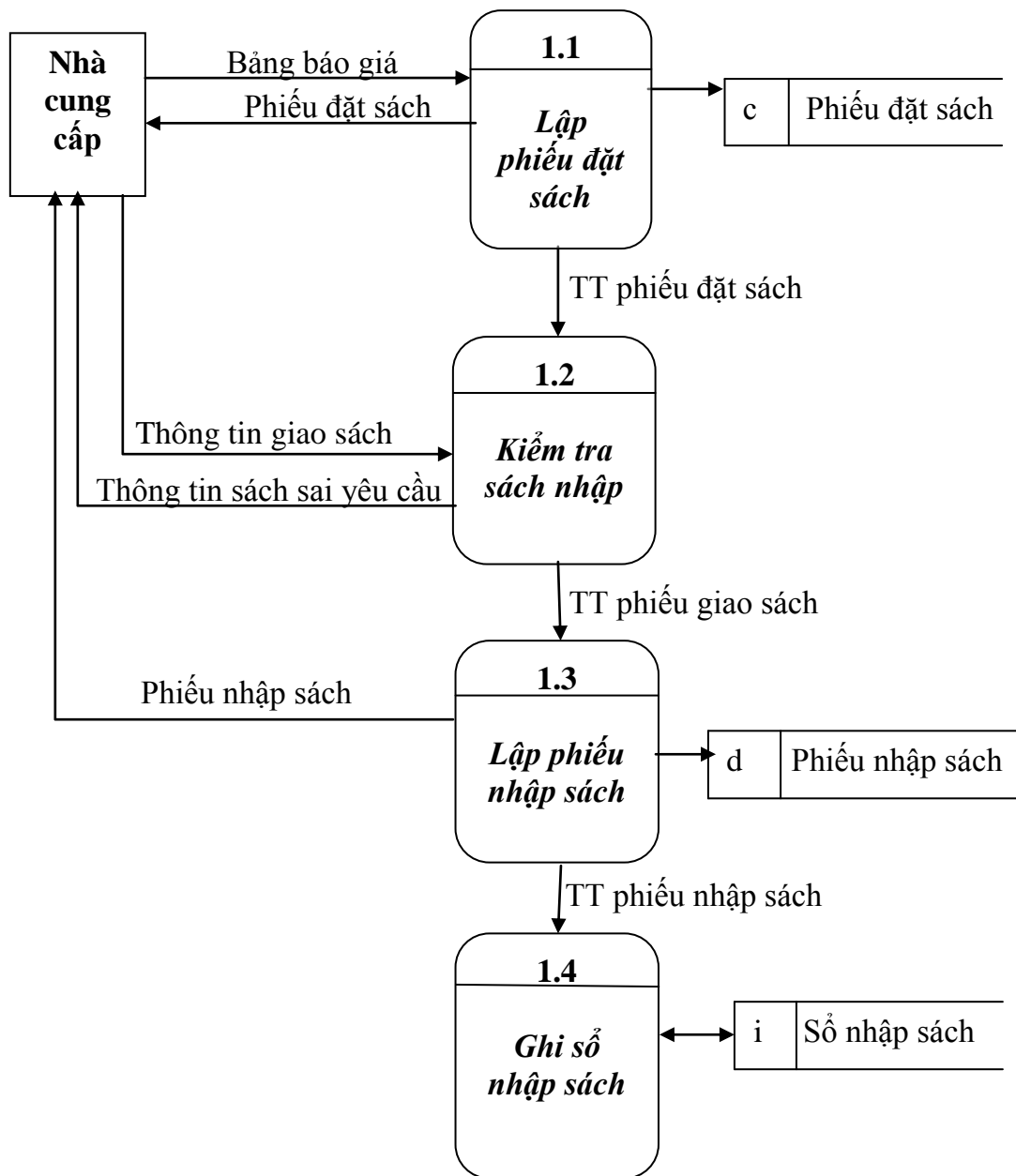
**2.5.8. Sơ đồ luồng dữ liệu**

**1. Sơ đồ luồng dữ liệu mức 0**

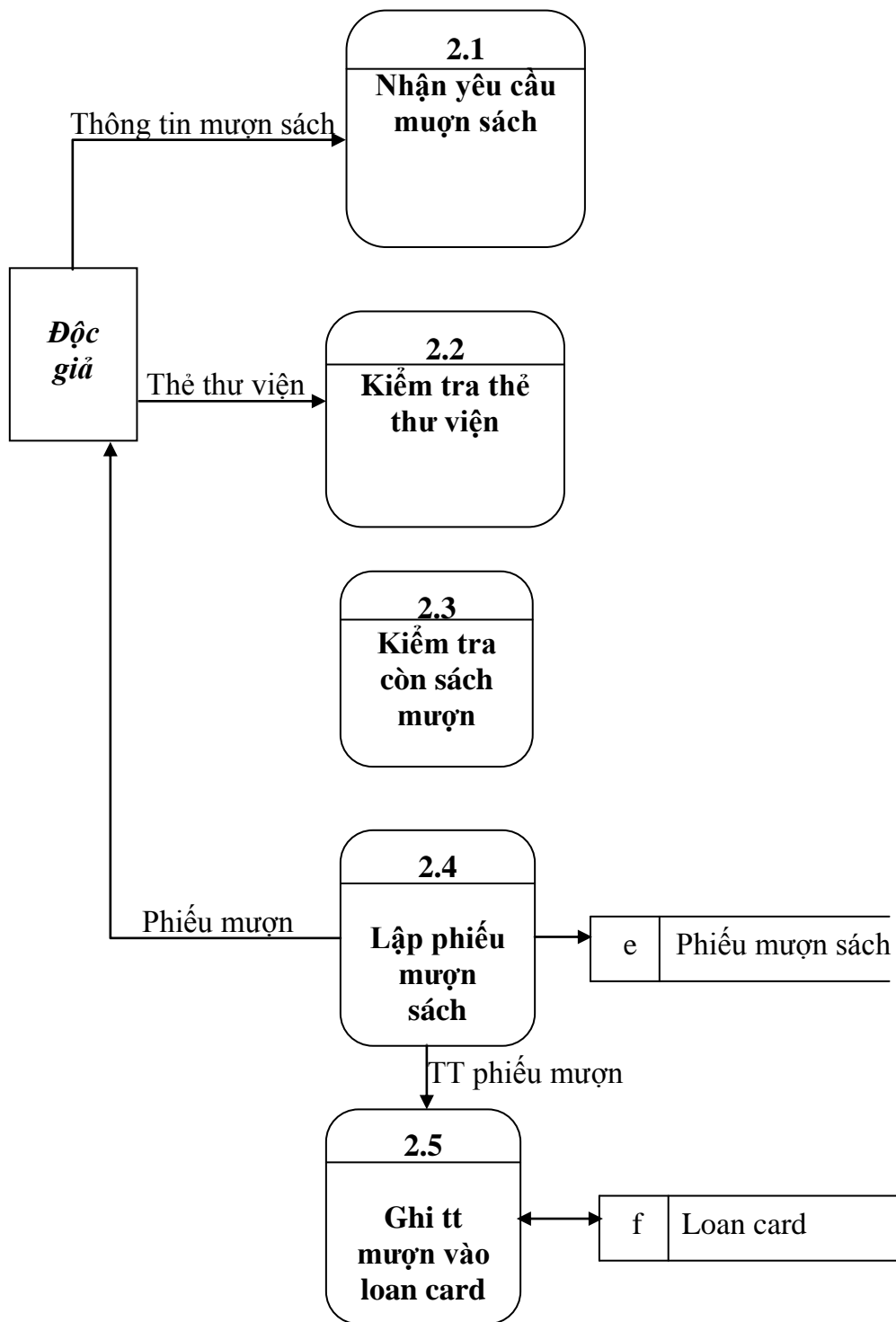


**2. Sơ đồ luồng dữ liệu mức 1**

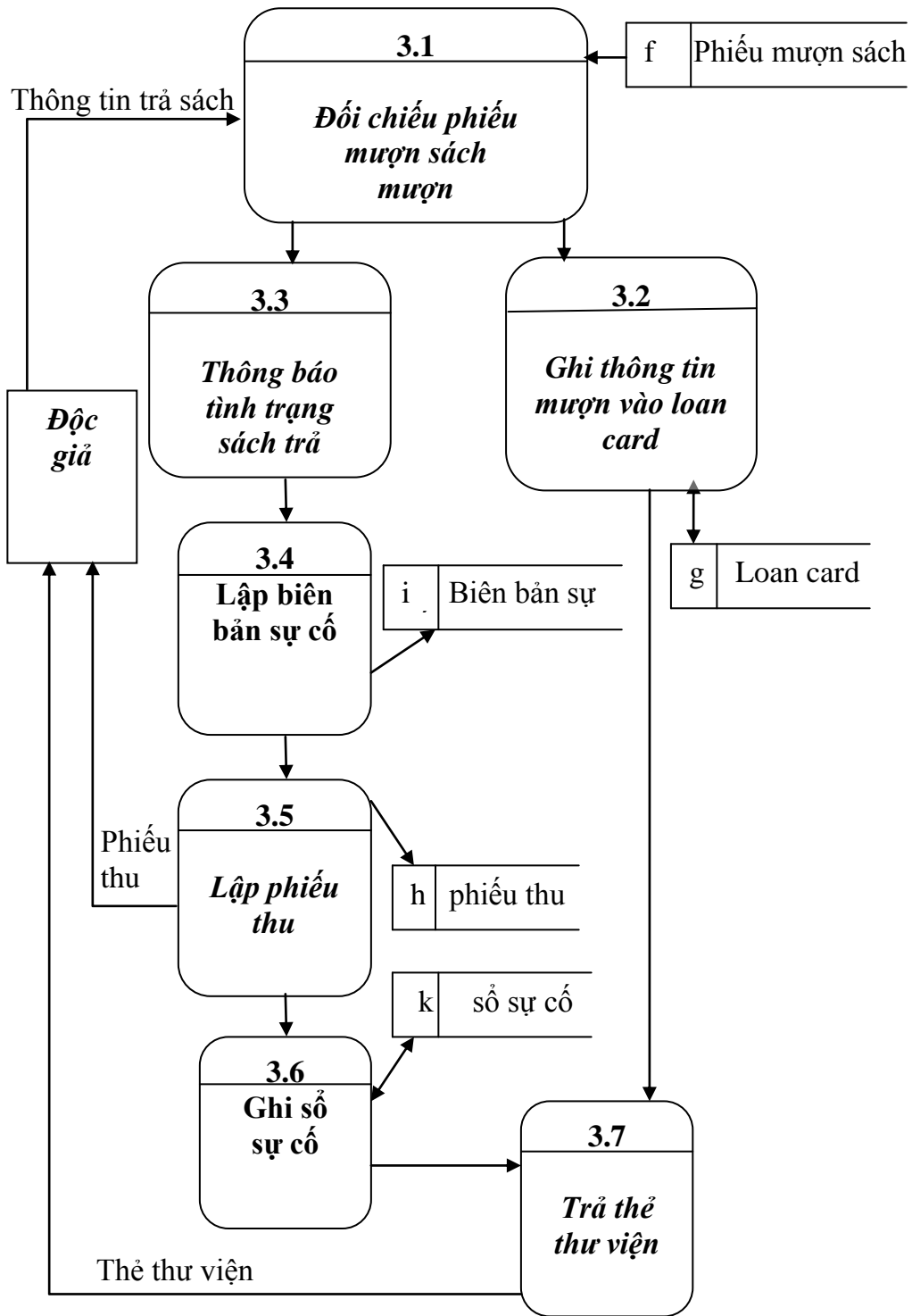
**a) Sơ đồ luồng dữ liệu mức 1 tiến trình “nhập sách”**



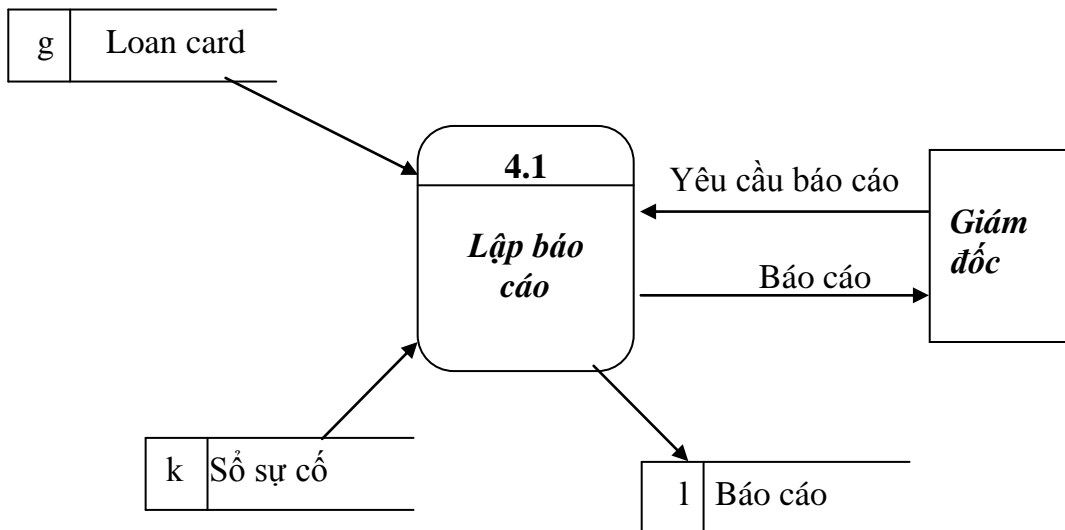
**b) Sơ đồ luồng dữ liệu mức 1 tiến trình “mượn sách”**



c) Sơ đồ luồng dữ liệu mức 1 tiến trình “trả sách”



**d) Sơ đồ luồng dữ liệu mức 1 tiến trình “báo cáo”**



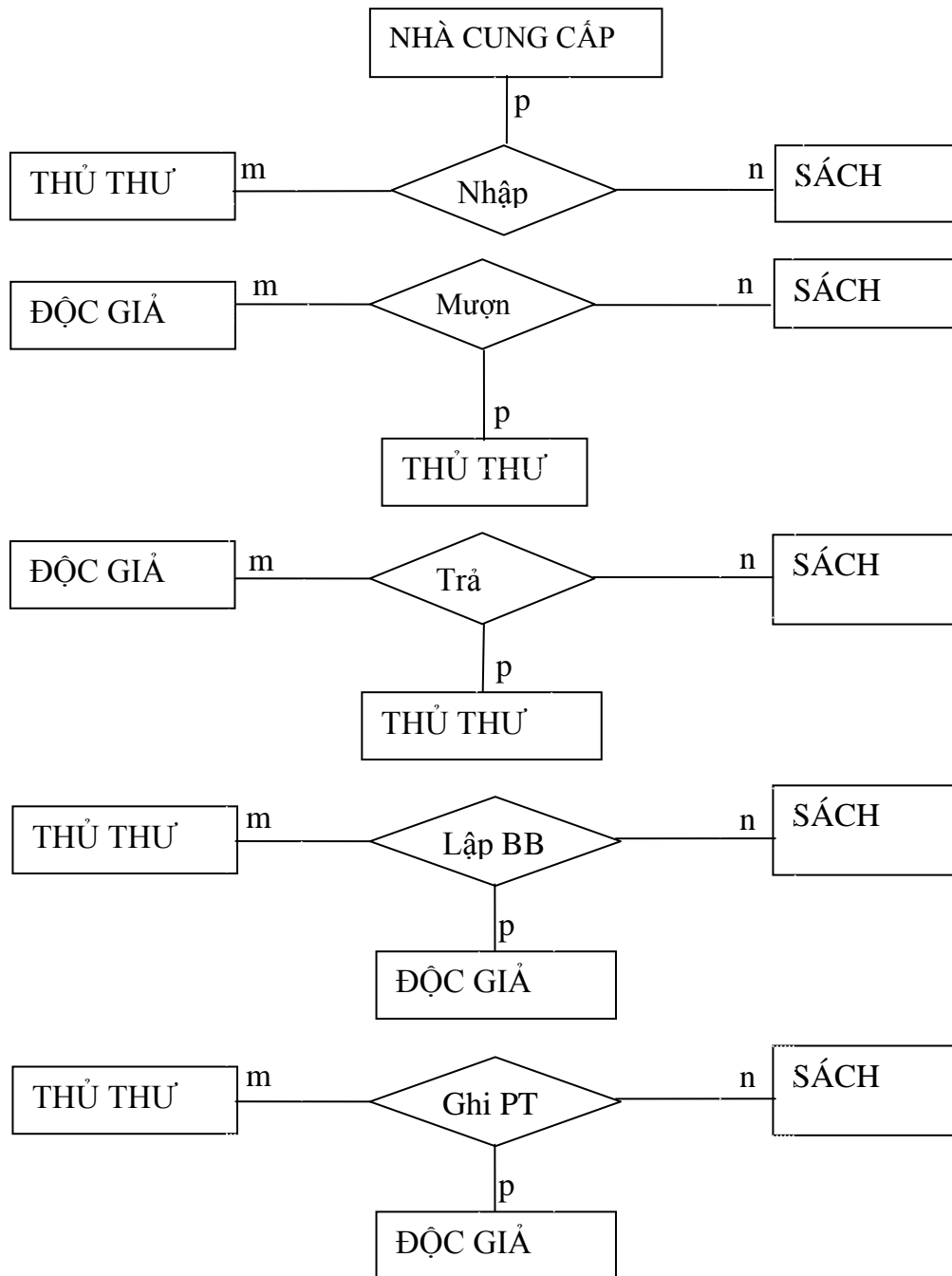
**2.6. THIẾT KẾ CƠ SỞ DỮ LIỆU**

**2.6.1. Mô hình liên kết thực thể (ER)**

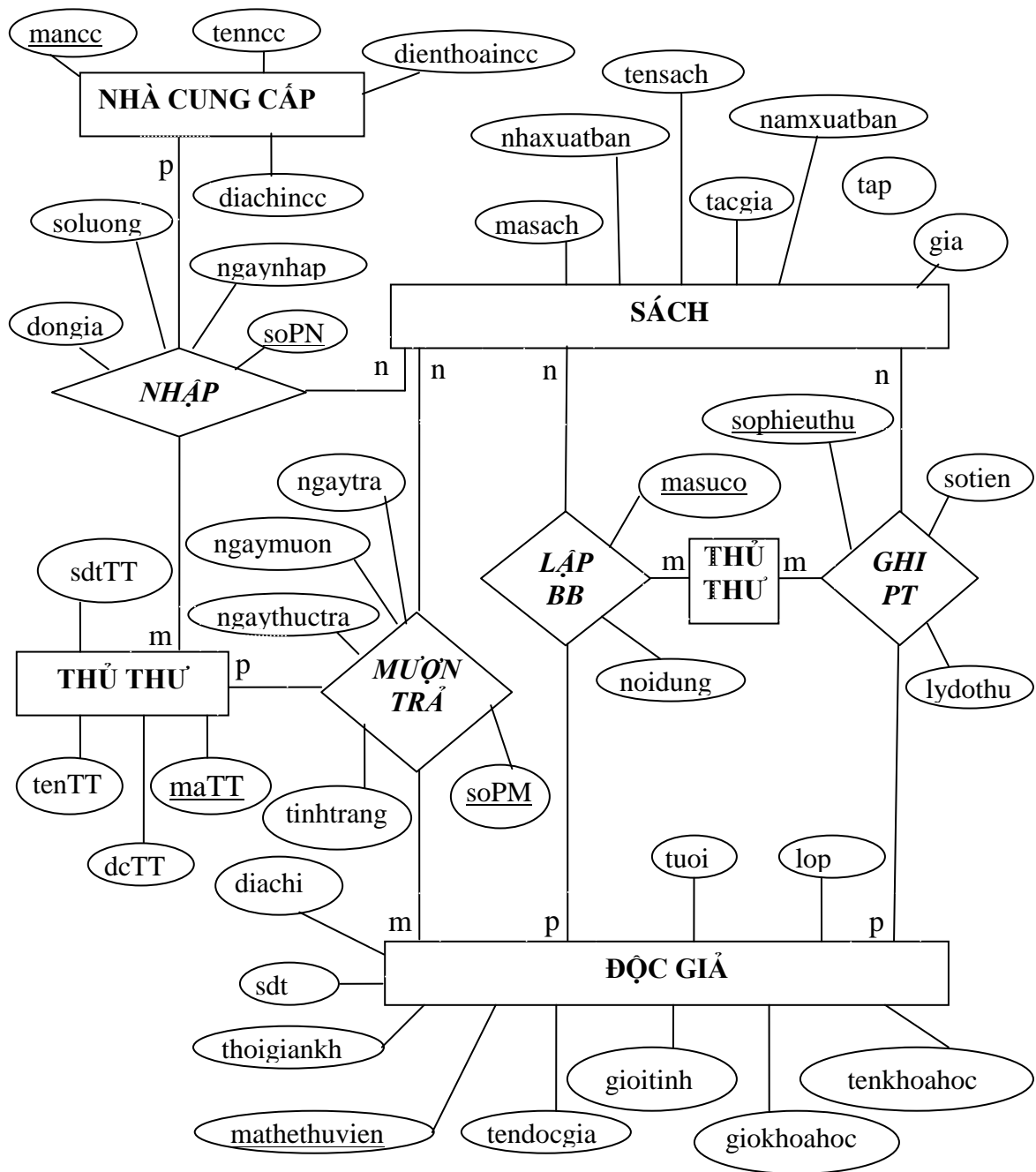
**1. Xác định các kiểu thực thể, các thuộc tính và thuộc tính khóa của thực thể**

Stt	Kiểu thực thể	Thuộc tính	Thuộc tính khóa
01	ĐỘC GIẢ	Mã thẻ thư viện, họ tên độc giả, địa chỉ độc giả, số điện thoại độc giả, giới tính, tuổi, lớp, khóa học, giờ khóa học, thời gian khóa học.	Mã thẻ thư viện
02	THỦ THƯ	Mã thủ thư, tên thủ thư, địa chỉ thủ thư, số điện thoại thủ thư.	Mã thủ thư
03	NHÀ CUNG CẤP	Mã nhà cung cấp, tên nhà cung cấp, địa chỉ nhà cung cấp, số điện thoại nhà cung cấp.	Mã nhà cung cấp
04	SÁCH	Mã sách, tên sách, đơn vị tính, giá, năm xuất bản, nhà xuất bản, tập.	Mã sách

**2. Xác định liên kết**



3. Mô hình ER



**Mô hình dữ liệu quan hệ****1. Bước 1:**

**Áp dụng thuật toán chuyển mô hình quan hệ ER thành các quan hệ sau:**

**- Biểu diễn các thực thể:**

**NHÀ CUNG CẤP**

NHÀ CUNG CẤP (Mã nhà cung cấp, tên nhà cung cấp, địa chỉ nhà cung cấp, số điện thoại nhà cung cấp)

**ĐỘC GIẢ**

ĐỘC GIẢ (Mã thẻ thư viện, tên độc giả, địa chỉ độc giả, số điện thoại độc giả, lớp, khóa học, thời gian khóa học, giờ khóa học, tên khóa học)

**THỦ THƯ**

THỦ THƯ (Mã thủ thư, tên thủ thư, địa chỉ thủ thư, số điện thoại thủ thư)

**SÁCH**

SÁCH (Mã sách, tên sách, năm xuất bản, nhà xuất bản, tập, giá, tác giả)

**- Biểu diễn các mối quan hệ:**

**Nhập sách**

PHIẾU NHẬP SÁCH (Số phiếu nhập, ngày nhập, mã nhà cung cấp, mã thủ thư, mã sách)

**Mượn trả**

PHIẾU MƯỢN TRẢ ( Số phiếu mượn, ngày mượn, ngày trả, ngày thực trả, mã sách, tên sách, mã thủ thư, mã thẻ thư viện, tình trạng)

**Lập biên bản sự cố**

BIÊN BẢN SỰ CỐ (mã sự cố, nội dung sự cố, ngày lập, mã thủ thư, mã thẻ thư viện, mã sách)

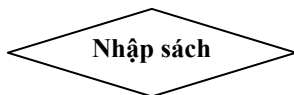




PHIẾU THU (Số phiếu thu, lý do, số tiền, mã thủ thư, mã thẻ thư viện, mã sách)

## 2. Bước 2:

Trong các quan hệ trên ta nhận thấy một số quan hệ còn dư thừa dữ liệu. Bởi vậy ta tách thành quan hệ mới như sau:



PHIẾU NHẬP SÁCH (Số phiếu nhập, ngày nhập, mã nhà cung cấp, mã thủ thư)

DÒNG PHIẾU NHẬP (Số phiếu nhập, mã sách, số lượng)

### c) Các quan hệ

#### 1. NCC

<u>Mã ncc</u>	Tên ncc	Địa chỉ ncc	Số điện thoại ncc

#### 2. THỦ THƯ

<u>Mã thủ thư</u>	Tên thủ thư	Địa chỉ thủ thư	Số điện thoại thủ thư

#### 3. SÁCH

<u>Mã sách</u>	Tên sách	Nhà XB	NămXB	Tác giả	Giá	Tập

#### 5. PHIẾU NHẬP SÁCH

<u>Số phiếu nhập</u>	Ngày nhập	Mã nhà cung cấp	Mã thủ thư

**6. DÒNG PHIẾU NHẬP**

<u>Số phiếu nhập</u>	Số lượng	Mã sách

**7. PHIẾU MƯỢN TRẢ**

<u>Số phiếu mượn</u>	Ngày mượn	Ngày trả	Ngày thực trả	Mã thủ thư	Mã thẻ thư viện	Tình trạng	Mã sách

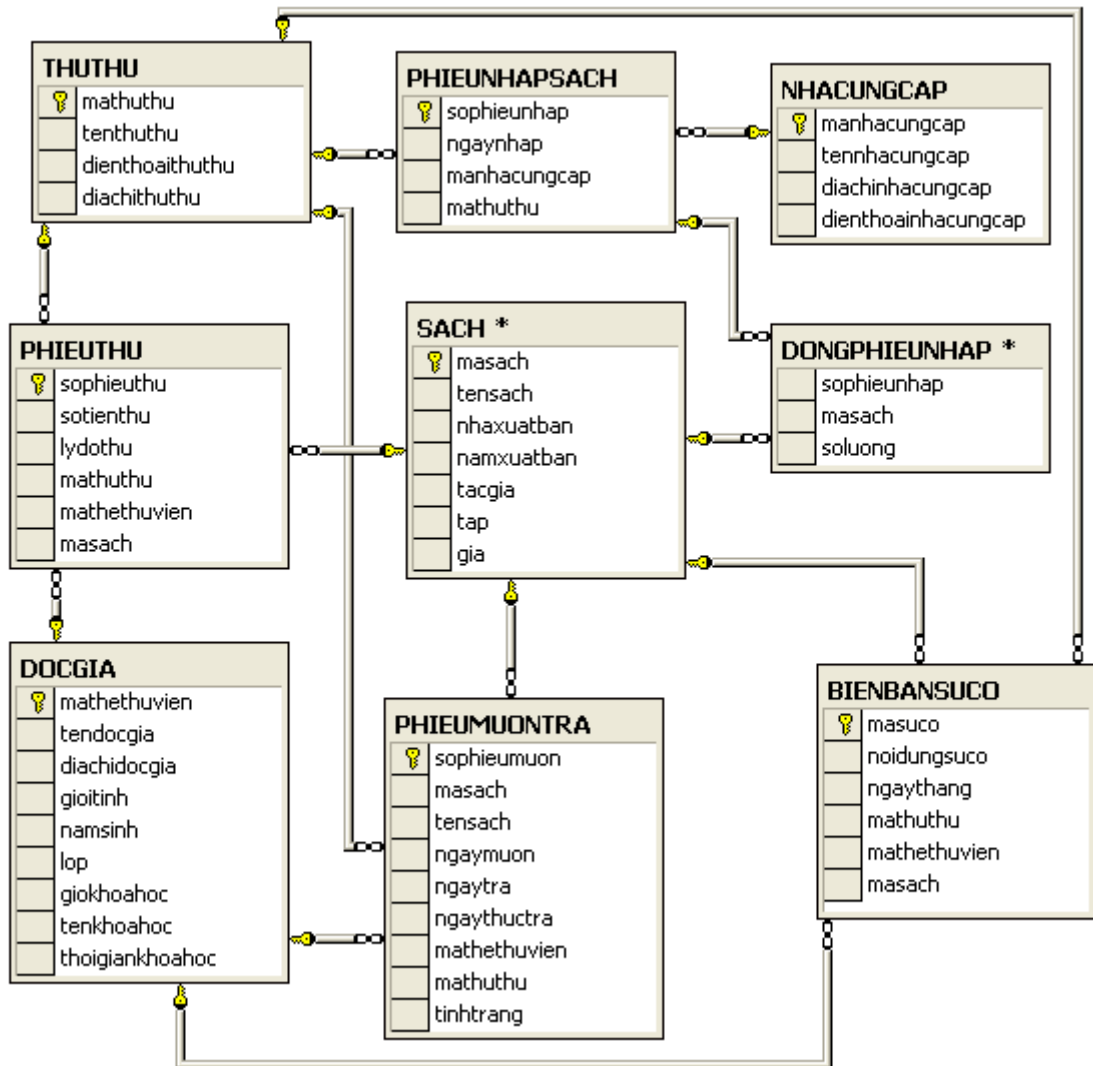
**8. BIÊN BẢN SỰ CỐ**

<u>Mã sự cố</u>	Nội dung sự cố	Ngày lập	Mã thủ thư	Mã thẻ thư viện	Mã sách

**9. PHIẾU THU**

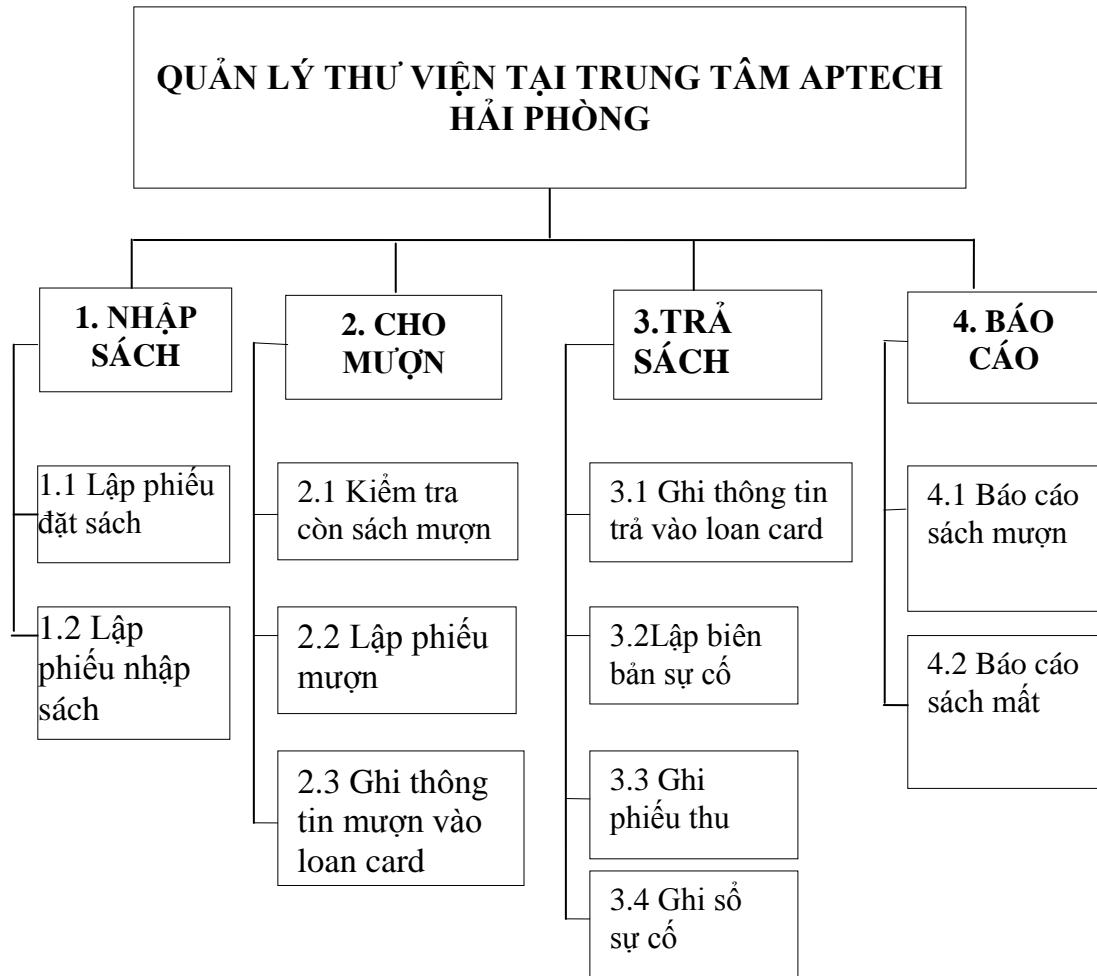
<u>Số phiếu thu</u>	Lý do	Số tiền	Mã thẻ thư viện	Mã thủ thư	Mã sách

2.6.2. Mô hình quan hệ



## CHƯƠNG 3: THIẾT KẾ CÀI ĐẶT CHƯƠNG TRÌNH

### 3.1. Sơ đồ chức năng hệ thống



## 3.2. Thiết kế giao diện

### 3.2.1. Giao diện form “Cập nhật danh mục sách”

**CẬP NHẬT DANH MỤC SÁCH**

Mã sách:  Tập:

Tên sách:  Giá:

Nhà xuất bản:

Năm xuất bản:

Tác giả:

Xem đầu Xem cuối Thêm Không lưu Xóa Tìm kiếm

Xem sau Xem trước Sửa Lưu Thoát

Danh mục sách							
	Mã sách	Tên sách	Nhà xuất bản	Năm xuất bản	Tác giả	Tập	Giá
▶	S01	Lập trình C	Quốc gia	2004	Nguyễn Huy	3	5000
	S02	Tin học căn bản	Việt hà	2001	Nguyễn Hùng	2	6000
	S03	Visual	Quốc gia	2004	Lê Thanh	1	7000
	S04	Phân tích	Thanh niên	1999	Đỗ Duy	2	8000
	S05	Lập trình Java	Quốc gia	2000	Văn Thành	2	4000
*							

**3.2.2. Giao diện form “Cập nhật thông tin mượn trả sách”**

Cap nhap thong tin muon tra sach
⏏ ⏏ ⏏

### CẬP NHẬT THÔNG TIN MƯỢN TRẢ SÁCH

Số phiếu mượn	PM01	Ngày trả	4/6/2010
Tên sách	Lập trình C ▾	Ngày thực trả	3/6/2010
Mã sách	S01	Mã thẻ thư viện	TV01 ▾
Ngày mượn	4/5/2010	Mã thủ thư	TT01 ▾

Xem đầu	Xem cuối	Thêm	Không lưu	Xóa	Tìm kiếm
Xem sau	Xem trước	Sửa	Lưu	Thoát	

Danh mục sách mượn								
	Số PM	Mã sách	Tên sách	Ngày mượn	Ngày trả	Ngày thực trả	Mã thẻ thư viện	Mã thủ thư
▶	PM01	S03	Visual	4/5/2010	4/6/2010	3/6/2010	TV01	TT01
	PM02	S01	Lập trình C	3/9/2009	3/1/2010	1/1/2010	TV02	TT02
	PM03	S01	Lập trình C	3/7/2010	6/8/2010	9/7/2010	TV01	TT02
*								

## 3.2.3. Giao diện form “Cập nhật thông tin độc giả”

**CẬP NHẬT THÔNG TIN ĐỘC GIẢ**

Mã thẻ thư viện:       Lớp:

Tên độc giả:       Giờ khóa học:

Địa chỉ độc giả:       Tên khóa học:

Giới tính:       Thời gian khóa học:

Năm sinh:

Xem đầu    Xem cuối    Thêm    Không lưu    Xóa    Tìm kiếm

Xem sau    Xem trước    Sửa    Lưu    Thoát

Danh sách độc giả								
	Mã thẻ thư viện	Tên độc giả	Địa chỉ	Giới tính	Năm sinh	Lớp	Giờ khóa học	Tên khóa học
▶	TV01	Nguyễn Nam Dũng	Hải Phòng	nam	1988	CNDA1	9h30	CNDA
	TV02	Cao Kỳ Duyên	Hải Dương	nu	1989	QT2	7h30	QT
	TV03	Phạm Thùy Chi	Bắc Giang	nu	1987	CNDA2	2h30	CNDA
	TV04	Hoàng Yến	Hải Phòng	nu	1988	QT1	7h30	QT

### 3.2.4. Giao diện form “Cập nhật danh sách thủ thư”

**CẬP NHẬT DANH SÁCH THỦ THƯ**

Mã thủ thư:

Tên thủ thư:

Địa chỉ:

Số điện thoại:

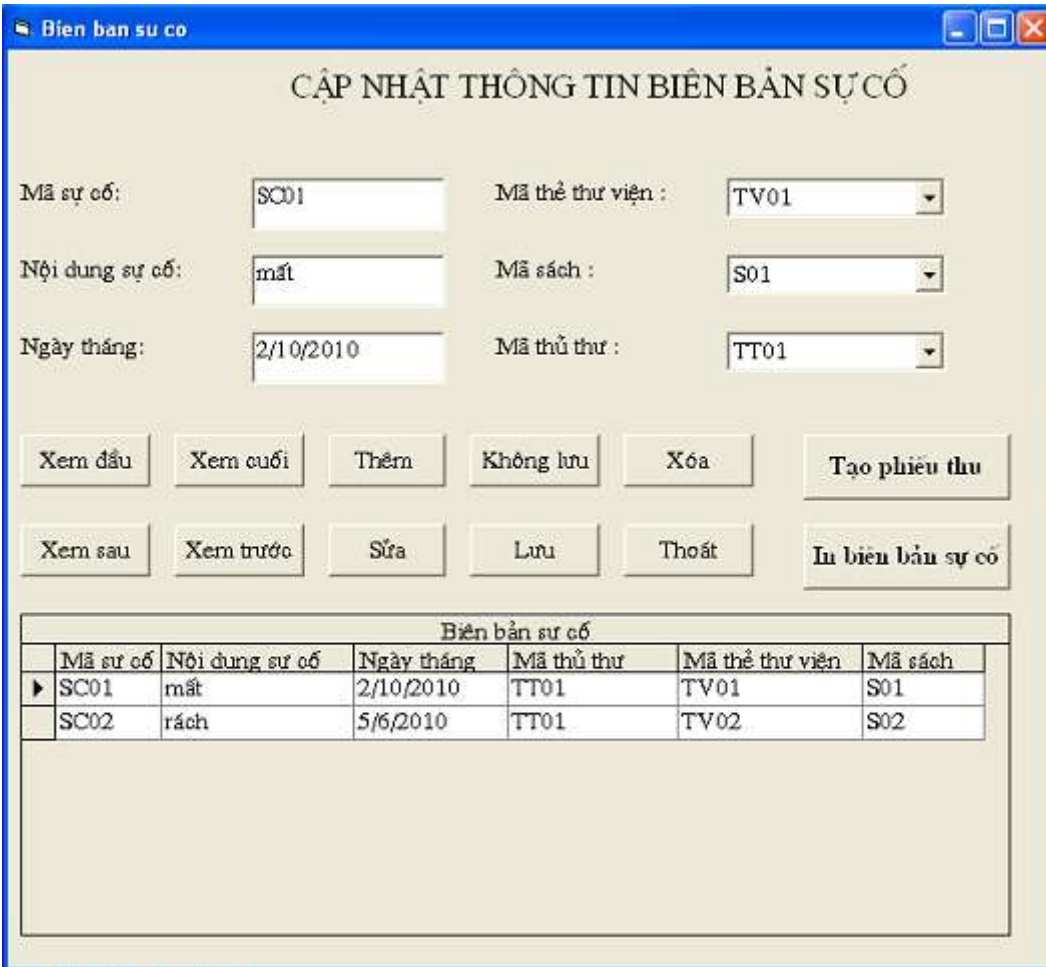
Xem đầu   Xem cuối   Thêm   Không lưu   Xóa   Tìm kiếm

Xem sau   Xem trước   Sửa   Lưu   Thoát

Danh sách thủ thư				
	Mã thủ thư	Tên thủ thư	Điện thoại	Địa chỉ
▶	TT01	Cao Kỳ Duyên	0978567483	Hải Phòng
	TT02	Nguyễn Thu NHàn	0978678657	Hải Dương
*				



### 3.2.5. Giao diện form “Cập nhật thông tin biên bản sự cố”



**Biên bản sự cố**

**CẬP NHẬT THÔNG TIN BIÊN BẢN SỰ CỐ**

Mã sự cố:  Mã thẻ thư viện:

Nội dung sự cố:  Mã sách:

Ngày tháng:  Mã thủ thư:

Xem đầu Xem cuối Thêm Không lưu Xóa Tạo phiếu thu

Xem sau Xem trước Sửa Lưu Thoát In biên bản sự cố

Biên bản sự cố						
	Mã sự cố	Nội dung sự cố	Ngày tháng	Mã thủ thư	Mã thẻ thư viện	Mã sách
▶	SC01	mất	2/10/2010	TT01	TV01	S01
	SC02	rách	5/6/2010	TT01	TV02	S02

## KẾT LUẬN

Việc áp dụng công nghệ thông tin trong tất cả các lĩnh vực của đời sống cũng như xã hội đã làm thay đổi căn bản diện mạo của xã hội cũng như đem lại những tiện ích to lớn. Với những kiến thức được học trong thời gian 4 năm qua. Cùng với những kinh nghiệm được các thầy cô truyền cho, sau khi nhận đề tài tốt nghiệp này em đã cố gắng khảo sát thực tế, đồng thời tham khảo bài học của những người đi trước. Thực tế đã giúp em biết cách tổ chức thiết kế một chương trình để áp dụng trong thực tế. Và thấy được vai trò to lớn của công nghệ thông tin trong xã hội hiện nay. Nhờ đó mà đã giảm rất nhiều sức lực của con người mà vẫn đem lại hiệu quả cao trong công việc.

Mong muốn có một giao diện thân thiện với người dùng, dễ thao tác và sử dụng. Nhưng vẫn đáp ứng được các chức năng mà hệ thống cần cung cấp là mục tiêu mà đề tài của hướng tới. Tuy nhiên với trình độ bản thân còn nhiều hạn chế, chưa thực sự hiểu hết được những khả năng mà hệ thống sẽ xảy ra để nắm vững toàn bộ vấn đề. Mặc dù đã xây dựng được chương trình gần gũi với người dùng nhưng nó vẫn còn nhiều hạn chế về chức năng. Chính vì vậy nên rất cần những ý kiến đóng góp để đồ án hoàn thiện hơn.

## TÀI LIỆU THAM KHẢO

- [1]. *Giáo trình phân tích thiết kế hệ thống thông tin hiện đại* - PGS. Nguyễn Văn Vy, nhà xuất bản thống kê Hà Nội- 2004.
- [2]. *Phân tích và thiết kế hệ thống thông tin* - PGS.TS: Nguyễn Văn Ba, nhà xuất bản đại học quốc gia Hà Nội.
- [3]. *Lập trình SQL SERVER 2000 căn bản* - Đoàn Thiện Ngân, Đoàn Thị Thanh Diễm, Hoàng Đức Hải, nhà xuất bản lao động xã hội.
- [4]. Website chính thức của trung tâm: <http://hp-apttech.edu.vn/>