

**ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO
ĐỒ ÁN LẬP TRÌNH
ĐỀ TÀI: “CHƯƠNG TRÌNH QUẢN LÝ THƯ VIỆN”**

**GIẢNG VIÊN HƯỚNG DẪN
PGS.TS. Nguyễn Thanh Bình
TS. Đặng Hoài Phương
TS. Trương Ngọc Châu**

**SINH VIÊN THỰC HIỆN
Sinh viên 1: Huỳnh Văn Quân
Sinh viên 2: Trịnh Xuân Phúc
Sinh viên 3: Nguyễn Quang Trường
Sinh viên 4: Nguyễn Văn Linh
Lớp: 18TCLC_DT1
Nhóm: DA_18**

Đà Nẵng, ngày 1 tháng 1 năm 2020

MỤC LỤC

LỜI NÓI ĐẦU.....	1
CHƯƠNG 1: CƠ SỞ LÝ THUYẾT.....	3
1.1. Cơ sở dữ liệu.....	3
1.1.1. Các khái niệm.....	3
1.1.1.1. Khái niệm cơ bản về cơ sở dữ liệu.....	3
1.1.1.2. Các tính chất của một cơ sở dữ liệu.....	3
1.1.1.3. Hệ quản trị cơ sở dữ liệu.....	3
1.1.2. Mô hình cơ sở dữ liệu.....	3
1.1.3. Mô hình Thực thể - Liên kết.....	4
1.1.4. Mô hình quan hệ.....	4
1.1.5. Thiết kế cơ sở dữ liệu.....	4
1.1.5.1. Thiết kế khái niệm.....	6
1.1.5.2. Lựa chọn phần mềm quản lý cơ sở dữ liệu.....	7
1.1.5.3. Thiết kế logic.....	7
1.1.5.4. Thiết kế vật lý.....	8
1.1.6. Ngôn ngữ SQL.....	8
1.1.6.1. Khái niệm.....	8
1.1.6.2. Các chức năng của SQL.....	8
1.1.7. Hệ quản trị SQL SERVER.....	9
1.1.7.1. Giới thiệu.....	9
1.1.7.2. Các thành phần cơ bản trong SQL server.....	9
1.2. Danh sách liên kết đôi.....	10
1.2.1. Khái niệm.....	10
1.2.2 Cấu trúc và các thao tác trên danh sách liên kết đôi.....	11
1.2.3. Các hoạt động cơ bản trên danh sách liên kết đôi:.....	11
1.3. Lập trình hướng đối tượng.....	13
1.3.1. Các khái niệm.....	13
1.3.1.1 Khái niệm cơ bản về lập trình hướng đối tượng(OOP).....	13
1.3.2. Các tính chất cơ bản.....	13
1.3.3. Một số khái niệm cần biết trong ngôn ngữ OOP.....	14
1.3.4. Ưu nhược điểm của lập trình hướng đối tượng.....	17
CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG.....	18
2.1. Khảo sát hiện trạng.....	18
2.2. Thiết kế cơ sở dữ liệu.....	18
2.2.1. Xác định thực thể.....	18
2.2.1.1 Thực thể BOOK.....	18
2.2.1.2. Thực thể BORROWRETURN.....	19
2.2.1.3. Thực thể STUDENT.....	19
2.2.1.4. Thực thể TYPEBOOK.....	19

2.2.2 Mô tả thực thể.....	19
2.2.3. Mô hình quan hệ ERD	20
2.2.4. Chuẩn hóa lược đồ quan hệ về dạng 3NF	21
2.3. Thiết kế các class.....	22
2.3.1. Thuộc tính.....	22
2.3.2. Phương thức.....	22
2.4. Cấu trúc dữ liệu	23
2.4.1. Class Node	23
2.4.2. Cấu trúc node.....	24
2.4.3. Các phương thức trên node.....	24
2.4.4. Cấu trúc danh sách liên kết đôi.....	24
2.4.5. Thao tác trên danh sách liên kết đôi	25
2.4.5.1. Thao tác lấy độ dài danh sách	25
2.4.5.2. Thao tác in danh sách liên kết	25
2.4.5.3. Thao tác in các phần tử có giá trị thuộc tính xác định	25
2.4.5.4. Thao tác tìm kiếm danh sách liên kết theo thuộc tính.....	25
2.4.5.6. Thao tác chèn phần tử trước node có giá trị thuộc tính xác định.....	26
2.4.5.7. Thao tác xóa node có giá trị thuộc tính xác định	26
2.4.5.8. Thao tác cập nhật lại một node	26
2.4.5.9. Thao tác cập nhật node tại vị trí xác định bởi thuộc tính.....	26
2.4.5.10. Thao tác sắp xếp danh sách liên kết.....	27
2.5. Phân tích thiết kế giải thuật	27
2.5.1. Tìm kiếm tuần tự	27
2.5.2. Sắp xếp chèn.....	27
CHƯƠNG 3: TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ.....	29
3.1. Chức năng.....	29
3.1.1. Sơ đồ phân cấp chức năng.....	29
3.1.2. Phân tích yêu cầu phi chức năng	29
3.1.3. Chức năng hiển thị.....	29
3.1.4. Chức năng tìm kiếm	29
3.1.5. Chức năng thêm.....	30
3.1.6. Chức năng xóa	30
3.1.7. Chức năng cập nhật	30
3.1.8. Chức năng sắp xếp.....	31
3.1.9. Chức năng mượn sách	31
3.1.10. Chức năng trả sách	31
3.2. Trình bày giao diện.....	32
3.2.1. Giao diện mở đầu	32
3.2.2. Menu chính	32

3.2.3. Menu hiển thị thông tin	33
3.2.4. Hiển thị thông tin sách.....	33
3.2.5. Hiển thị thông tin sinh viên	33
3.2.6. Hiển thị thông tin mượn trả	33
3.2.7. Hiển thị thông tin loại sách.....	34
3.2.8. Sắp xếp thông tin sinh viên theo họ tên.....	34
3.2.9. Sắp xếp thông tin mượn trả theo ngày mượn	34
3.2.10. Thêm sinh viên	35
3.3. Đánh giá kết quả	35
3.3.1. Đối tượng tiếp cận	35
3.3.2. Tính thuận tiện.....	35
3.3.3. Khả năng trở nên phổ biến	35
3.3.4. Vấn đề tiếp cận với người dùng	36
CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	36
4.1. Kết luận:	36
4.2. Hạn chế:	36
4.3. Hướng phát triển:.....	36
TÀI LIỆU THAM KHẢO.....	37

DANH SÁCH HÌNH VẼ

Hình 1. Quan điểm khác nhau của người quản lý và người thiết kế về dữ liệu	5
Hình 2. Các bước thiết kế cơ sở dữ liệu	6
Hình 3. Cấu trúc của một Node	10
Hình 4. Cấu trúc DSLK đôi	11
Hình 5. Thực thể BOOK.....	19
Hình 6. Thực thể BORROWRETURN	20
Hình 7. Thực thể STUDENT.....	20
Hình 8. Thực thể TYPEBOOK.....	20
Hình 9. Mô hình quan hệ ERD	20
Hình 10. Sơ đồ phân cấp chức năng.....	29
Hình 11. Giao diện chính.....	32
Hình 12. Menu chính	32
Hình 13. Menu hiển thị thông tin	33
Hình 14. Hiển thị thông tin sách.....	33
Hình 15. Hiển thị thông tin sinh viên	33
Hình 16. Hiển thị thông tin mượn trả	33
Hình 17. Hiển thị thông tin loại sách.....	34
Hình 18. Sắp xếp thông tin sinh viên theo họ tên.....	34
Hình 19. Sắp xếp thông tin mượn trả theo ngày mượn	34
Hình 20. Giao diện thêm sinh viên	35

DANH SÁCH BẢNG

Bảng 1. Thuộc tính các class	22
Bảng 2. Ví dụ phương thức BOOK.....	23
Bảng 3. Cấu trúc node	24
Bảng 4. Thao tác lấy độ dài danh sách liên kết	25
Bảng 5. Thao tác in danh sách liên kết.....	25
Bảng 6. Thao tác tìm các phần tử có giá trị thuộc tính xác định	25
Bảng 7. Thao tác tìm kiếm danh sách liên kết theo thuộc tính.....	25
Bảng 8. Thao tác chèn node vào cuối danh sách liên kết.....	26
Bảng 9. Thao tác chèn phần tử trước node có giá trị thuộc tính xác định.....	26
Bảng 10. Thao tác xóa node có giá trị thuộc tính xác định	26
Bảng 11. Thao tác cập nhật lại một node	26
Bảng 12. Thao tác cập nhật node tại vị trí xác định bởi thuộc tính.....	27
Bảng 13. Thao tác sắp xếp danh sách liên kết.....	27
Bảng 14. Giải thuật tìm kiếm tuần tự	27
Bảng 15. Giải thuật sắp xếp chèn	28

LỜI NÓI ĐẦU

Giới thiệu

Trong những năm gần đây, Đà Nẵng ngày càng phát triển mạnh và trở thành một trung tâm kinh tế, dịch vụ, công nghệ lớn. Do đó, việc phát triển các hệ thống phần mềm quản lý nhằm tối ưu trong công việc, quản lý hiệu quả con người, cơ sở vật chất... là một điều hết sức cần thiết. Là những sinh viên và cũng đang là những độc giả thường xuyên mượn sách thư viện, chúng em thường đối mặt với nhiều vấn đề của hệ thống quản lý, mất nhiều thời gian khi thực hiện các thủ tục. Vì thế, để áp dụng những kiến thức đã học vào thực tế, chúng em đã cùng nhau lên ý tưởng và xây dựng một chương trình mang tên “Quản lý thư viện” nhằm hướng tới xây dựng hệ thống thư viện tối ưu, giải quyết được nhiều vấn đề trong thực tiễn. Trong quá trình thực hiện đồ án, có nhiều kiến thức về lập trình, giải thuật,... mà có thể chúng em chưa nắm được nên sai sót là điều khó tránh khỏi. Kính mong quý thầy cô tận tình góp ý để đồ án ngày một hoàn thiện hơn.

Chúng em xin chân thành cảm ơn sự hướng dẫn tận tình của quý thầy cô là giảng viên trong khoa trong suốt quá trình thực hiện đồ án này!

Vấn đề thực tế

- Sinh viên thực hiện các thủ tục ở thư viện nhưng mất nhiều thời gian.
- Hệ thống quản lý thư viện chưa đáp ứng đủ nhu cầu thực tế như về lưu trữ, quản lý thông tin,...
- Việc sử dụng các phương pháp quản lý thư viện truyền thống gây mất thời gian, không kiểm soát tốt mọi thứ, dễ gây lỗi.

Vấn đề bao quát

- Cần xây dựng một ứng dụng quản lý thư viện
- Ứng dụng quản lý thư viện phải đáp ứng được đồng thời các yếu tố như tính tối giản, đầy đủ các chức năng cơ bản,...
- Ứng dụng phải đáp ứng được các nhu cầu thực tế, giao diện đơn giản, giúp người quản lý có thể tương tác dễ dàng.

Mục đích nghiên cứu

- Quản lý tốt thông tin của sách, sinh viên,...
- Thực hiện các thủ tục trong thư viện một cách tối ưu, nhanh nhất như đăng ký, mượn trả,...
- Nhằm hướng đến việc tạo ra một hệ thống hoàn chỉnh, áp dụng được cho nhiều mô hình thư viện.

Cách giải quyết

- Thiết kế và xây dựng một ứng dụng quản lý thư viện

Ý nghĩa lý luận và thực tiễn của đề tài

- Giúp sinh viên ứng dụng các kiến thức đã học vào thực tế, hình thành nên

tư duy khi thiết kế và xây ứng dụng hoàn chỉnh, tạo tiền đề để thực hiện những dự án lớn hơn.

- Giải quyết được các vấn đề mà thư viện và độc giả đang gặp phải, tạo ra một hệ thống quản lý tối ưu và chất lượng nhất.
- Có thể ứng dụng phổ biến cho các mô hình thư viện nhiều nơi.

CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

1.1. Cơ sở dữ liệu

1.1.1. Các khái niệm

1.1.1.1. Khái niệm cơ bản về cơ sở dữ liệu

Cơ sở dữ liệu là một tập hợp các dữ liệu có tổ chức, thường được lưu trữ và truy cập điện tử từ hệ thống máy tính. Khi cơ sở dữ liệu phức tạp hơn, chúng thường được phát triển bằng cách sử dụng các kỹ thuật thiết kế và mô hình hóa chính thức.

1.1.1.2. Các tính chất của một cơ sở dữ liệu

Một cơ sở dữ liệu biểu thị một khía cạnh nào đó của thế giới thực. Những thay đổi của thế giới thực phải được phản ánh một cách trung thực vào trong cơ sở dữ liệu. Những thông tin được đưa vào trong cơ sở dữ liệu tạo thành một không gian cơ sở dữ liệu.

Một cơ sở dữ liệu là một tập hợp dữ liệu liên kết với nhau một cách logic và mang một ý nghĩa cố hữu nào đó. Một cơ sở dữ liệu không phải là một tập hợp tùy tiện.

Một cơ sở dữ liệu được thiết kế và được phổ biến cho một mục đích riêng. Nó có một nguồn cung cấp dữ liệu, một mức độ tương tác với các sự kiện trong thế giới thực và một nhóm người quan tâm tích cực đến các nội dung của nó.

Một cơ sở dữ liệu có thể có kích thước tùy ý và có độ phức tạp thay đổi. Có những cơ sở dữ liệu có vài trăm bản ghi và có những cơ sở dữ liệu có dung lượng rất lớn. Do đó, việc tổ chức quản lý sao cho có thể tìm kiếm, cập nhật dữ liệu và lấy dữ liệu là điều cần thiết.

1.1.1.3. Hệ quản trị cơ sở dữ liệu

Hệ quản trị cơ sở dữ liệu là các chương trình cho phép người dùng tạo ra, duy trì và khai thác một cơ sở dữ liệu. Đó là phần mềm hệ thống cho phép định nghĩa, xây dựng, xử lý và chia sẻ CSDL giữa những người dùng khác nhau và ứng dụng. Định nghĩa CSDL gắn liền với việc chỉ định kiểu dữ liệu, cấu trúc và ràng buộc của dữ liệu được lưu trữ trong CSDL. Xây dựng CSDL là lưu trữ dữ liệu trên bộ nhớ phụ được điều khiển bởi một hệ quản trị cơ sở dữ liệu. Xử lý CSDL bao gồm các thao tác như truy xuất dữ liệu theo yêu cầu, cập nhật dữ liệu để phản ánh sự thay đổi của thế giới thực, phát sinh báo cáo từ dữ liệu. Chia sẻ CSDL cho phép nhiều người dùng và nhiều ứng dụng truy xuất đến CSDL.

Hệ quản trị CSDL cung cấp một giao diện giữa người sử dụng và dữ liệu. Một số HQT CSDL quan hệ đang phổ biến như: Microsoft Access, Microsoft SQL Server, Oracle,... Trong đề tài, nhóm chúng em đã tìm hiểu và sử dụng hệ quản trị Microsoft SQL Server.

1.1.2. Mô hình cơ sở dữ liệu

Các loại cấu trúc cơ sở dữ liệu và mối liên hệ giữa chúng đóng vai trò rất lớn trong việc xác định tính hiệu quả của hệ quản trị cơ sở dữ liệu. Vì vậy, thiết kế cơ sở dữ liệu trở thành hoạt động chính trong môi trường cơ sở dữ liệu.

Việc sử dụng các mô hình giúp thiết kế cơ sở dữ liệu đơn giản hơn nhiều. Các mô hình là sự trừu tượng đơn giản của các sự kiện trong thế giới thực. Các trừu tượng như vậy cho phép ta khảo sát các đặc điểm của các thực thể và các mối liên hệ được tạo ra giữa các thực thể đó. Việc thiết kế các mô hình tốt sẽ đưa ra các cơ sở dữ liệu tốt và trên cơ sở đó sẽ có ứng dụng tốt. Hiện nay, có các loại như mô hình quan hệ, mô hình mạng, mô hình phân cấp,...

1.1.3. Mô hình Thực thể - Liên kết

Mô hình phổ biến nhất để biểu diễn trừu tượng cấu trúc của một cơ sở dữ liệu là mô hình liên kết – thực thể. Trong mô hình ER, cấu trúc của dữ liệu được biểu diễn dưới dạng đồ thị như là một “lược đồ liên kết – thực thể” sử dụng các thành phần như các kiểu thực thể, các thuộc tính, các kiểu liên kết.

Thực thể là một đối tượng trong thế giới thực, có sự tồn tại độc lập. Một thực thể có thể là một đối tượng cụ thể hoặc có thể là một đối tượng trừu tượng. Trong thực thể có nhiều thuộc tính, đó là các đặc trưng cụ thể mô tả thực thể đó. Khi một thuộc tính của một kiểu thực thể tham chiếu đến một thực thể khác thì ta nói có tồn tại một liên kết. Có nhiều kiểu liên kết khác nhau như kiểu liên kết cấp 1, cấp 2,...

Dựa vào các kiến thức lý thuyết về mô hình Thực thể - Liên kết giúp chúng em hình dung được khái quát về cơ sở dữ liệu, tạo thuận lợi để thiết kế cơ sở dữ liệu.

1.1.4. Mô hình quan hệ

Mô hình quan hệ sử dụng khái niệm quan hệ toán học như khối xây dựng cơ sở và có lý thuyết của nó trong lý thuyết tập hợp và logic vị từ bậc nhất.

Mô hình quan hệ biểu thị cơ sở dữ liệu như một tập các quan hệ. Mỗi quan hệ có thể được biểu diễn như một bảng giá trị, mỗi dòng trong bảng biểu thị một tập hợp các giá trị dữ liệu liên quan với nhau. Trong mô hình quan hệ, mỗi dòng trong bảng biểu thị một sự kiện tương ứng với một thực thể hoặc một liên kết của thế giới thực. Tên bảng và tên cột dùng để giải thích ý nghĩa của các giá trị trong mỗi hàng. Mọi giá trị trong một cột đều cùng kiểu dữ liệu.

Theo thuật ngữ mô hình quan hệ hình thức, mỗi hàng được gọi là một bộ, mỗi đầu cột được gọi là một thuộc tính và bảng được gọi là một quan hệ. Kiểu dữ liệu mô tả các kiểu của dữ liệu xuất hiện trong mỗi cột gọi là một miền. Ngoài ra, các miền còn có ràng buộc giúp cho các bộ dữ liệu được thống nhất, tạo nên cơ sở dữ liệu chính xác, tối ưu.

1.1.5. Thiết kế cơ sở dữ liệu

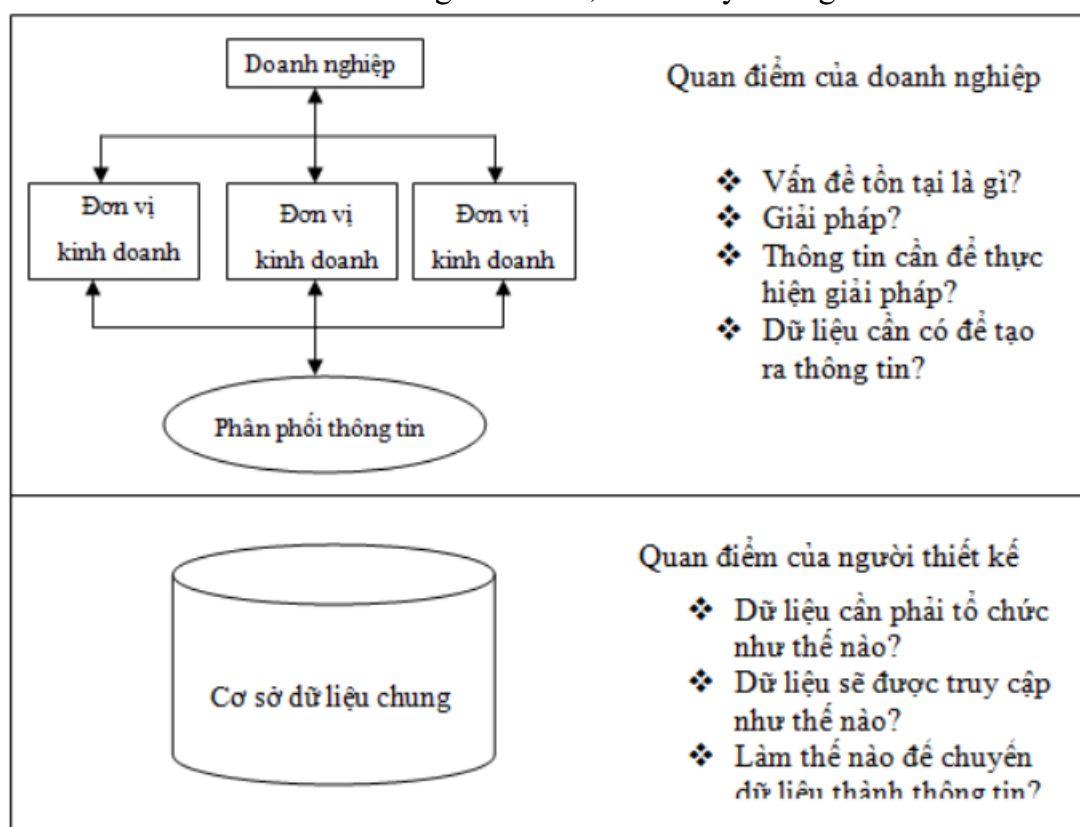
Thiết kế cơ sở dữ liệu là một bước hết sức quan trọng và cần thiết trong các hệ thống. Việc hệ thống có tối ưu, thực hiện tốt các chức năng hay không phụ thuộc rất nhiều vào cơ sở dữ liệu.

Việc thiết kế cơ sở dữ liệu có thể thực hiện bằng cách sử dụng hai giải pháp: dưới lên hoặc trên xuống. Phương pháp thiết kế từ dưới lên xem các mối liên kết cơ bản giữa các thuộc tính riêng rẽ như là điểm xuất phát và sử dụng chúng để xây

dựng nên các quan hệ. Giải pháp này còn có tên gọi là thiết kế bằng tổng hợp. Ngược lại, phương pháp thiết kế từ trên xuống, còn gọi là thiết kế bằng phân tích bắt đầu từ một số các nhóm thuộc tính trong các quan hệ nhận được từ thiết kế quan niệm và các hoạt động chuyển đổi. Sau đó, việc thiết kế bằng phân tích được áp dụng đối với các quan hệ một cách riêng rẽ và tập thể dẫn đến việc tách các quan hệ cho đến khi đạt được tính chất mong muốn.

Trong quá trình thiết kế cơ sở dữ liệu, điều cơ bản nhất là cần tập trung vào phân tích những tính chất cơ bản của dữ liệu tạo nên cơ sở dữ liệu. Khi thiết kế, ta cần chú ý tới cả hai khía cạnh hay hai cách nhìn nhận khác nhau ảnh hưởng lên cơ sở dữ liệu đó là cách nhìn nhận mang tính kinh tế và cách nhìn nhận của bản thân người thiết kế về cấu trúc của cơ sở dữ liệu.

Khi kiểm tra các bước đang tiến hành, cần chú ý những điểm sau:



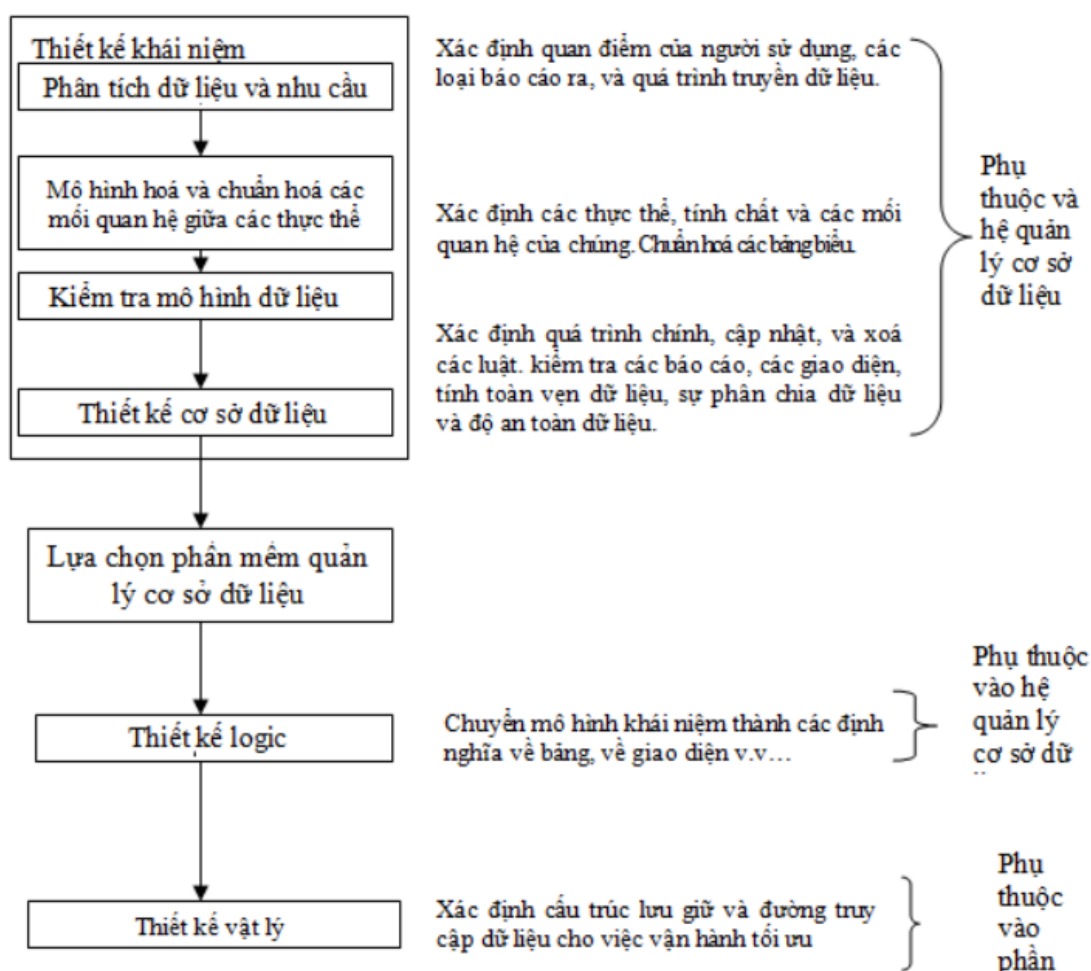
Hình 1. Quan điểm khác nhau của người quản lý và người thiết kế về dữ liệu

Quá trình tạo cơ sở dữ liệu có liên quan chặt chẽ tới việc phân tích và thiết kế một hệ thống lớn, chỉ là một Dữ liệu yếu tố trong cả hệ thống to lớn đó.

Những người phân tích hệ thống và người lập trình hệ thống đồng thời cũng phải chịu trách nhiệm thêm về các yếu tố khác trong hệ thống. Họ tạo ra các thủ tục cho phép chuyển đổi dữ liệu trong cơ sở dữ liệu thành những thông tin có ích. Thiết kế cơ sở dữ liệu không chỉ là duy trì một chuỗi quá trình, mà hơn thế nữa đó là một quá trình lặp đi lặp lại duy trì các thông tin phản hồi cho các bước trước đó.

1.1.5.1 Thiết kế khái niệm

Trong việc thiết kế khái niệm, mô hình hoá các dữ liệu nhằm tạo ra một cấu trúc cơ sở dữ liệu ngắn gọn giới thiệu những đối tượng thực sự trong phần lớn các sách có khả năng thực tiễn nhất. Mô hình khái niệm cần phải gắn liền với lĩnh vực doanh nghiệp và các chức năng của nó. Tại mức trừu tượng hoá này, dạng phần cứng và mô hình cơ sở dữ liệu còn chưa được xác định rõ ràng. Do đó, cần chắc chắn rằng dữ liệu trong mô hình là cần thiết và tất cả những dữ liệu đã được đưa vào cơ sở dữ liệu đều là cần thiết. Một điều cần chú ý nữa là các dữ liệu đã được thiết kế cần phải để dành những chỗ trống cho việc cập nhật và sửa đổi trong tương lai về sau.



Hình 2. Các bước thiết kế cơ sở dữ liệu

Mô hình hoá và tiêu chuẩn hoá các mối quan hệ giữa các thực thể

Trước khi tạo ra một mô hình quan hệ giữa các thực thể, người thiết kế cần phải chuẩn hoá toàn bộ những tài liệu được thiết kế. Các tiêu chuẩn để chuẩn hoá ở đây bao gồm các ký tự, các dấu hiệu, các dạng tài liệu viết, và những yếu tố khác

cần phải quan tâm trong việc trình bày một tài liệu. Việc này sẽ giúp các nhà thiết kế rất nhiều trong quá trình thiết kế về sau. Bất cứ một sự thiếu chuẩn hoá nào cũng có thể sẽ gây ra những khó khăn về việc truyền tin sau này đặc biệt là khi chương trình thực hiện trong một nhóm thống nhất. Ngay sau khi người quản lý xác định được bản chất các mối quan hệ giữa các thực thể trong một doanh nghiệp, người thiết kế cần nhanh chóng chuyển đổi chúng thành mô hình khái niệm.

Phân tán cơ sở dữ liệu

Các phần làm việc khác nhau sẽ nằm ở những vị trí vật lý hoàn toàn khác nhau. Các quá trình truy cập cơ sở dữ liệu cũng rất khác nhau từ vị trí này qua vị trí khác. Chẳng hạn như một quá trình bán buôn và một quá trình lưu trữ kho bãi sẽ nằm ở những vị trí hoàn toàn khác nhau. Nếu cơ sở dữ liệu có thể phân tán được trên toàn bộ hệ thống, người thiết kế buộc phải lưu ý tới việc phát triển hệ thống phân tán dữ liệu và các chiến lược phân tán cho cơ sở dữ liệu hoạt động có hiệu quả.

1.1.5.2. Lựa chọn phần mềm quản lý cơ sở dữ liệu

Khi lựa chọn phần mềm quản lý cơ sở dữ liệu, phần quan trọng nhất là cần phải nghiên cứu những ưu và nhược điểm của phần mềm đó. Người sử dụng có thể cũng nên được biết về những ưu và nhược điểm này để tránh được một số sai lầm trong thao tác. Khi nghiên cứu để thiết kế hệ thống cơ sở dữ liệu, những vấn đề sau đây thường được xem xét:

- **Chi phí:**

Bao gồm các chi phí mua bán, chi phí duy trì, chi phí điều hành, chi phí thiết lập, chi phí đào tạo, và chi phí chuyển giao.

- **Các công cụ và các đặc điểm của hệ quản lý cơ sở dữ liệu:**

Một số các công cụ trực thuộc phần mềm đó có thể phục vụ cho nhiệm vụ đòi hỏi cần phải được xem xét như khả năng tạo các mối liên hệ, trang trí màn hình, việc tạo các báo cáo, việc tạo các từ điển dữ liệu, v.v... Tất cả những yếu tố đó có thể làm cho công việc trở nên nhanh chóng hơn và tạo một môi trường làm việc tốt hơn cho doanh nghiệp. Những phương tiện hành chính trong cơ sở dữ liệu, các phương tiện tạo phần hỗ trợ trả lời các câu hỏi thắc mắc của người sử dụng, tính dễ sử dụng, tính an toàn dữ liệu, v.v... cũng là những yếu tố có ảnh hưởng tới việc lựa chọn hệ cơ sở dữ liệu cần thiết.

- **Mô hình chính:**

Mô hình thứ bậc, mô hình mạng hay mô hình quan hệ được sử dụng làm cơ sở để xây dựng hệ thống cũng là vấn đề cần làm rõ càng sớm càng tốt.

1.1.5.3 Thiết kế logic

Thiết kế logic có nghĩa là chuyển đổi từ thiết kế khái niệm thành mô hình bên trong một hệ thống quản lý cơ sở dữ liệu đã được chọn. Đối với một hệ thống quản lý cơ sở dữ liệu quan hệ, thiết kế logic bao gồm việc thiết kế các bảng, các chỉ số, các giao diện, các chuyển đổi, các thủ tục truy cập thông tin, v.v... Nói cách khác, thiết kế logic nghĩa là dịch mô hình khái niệm độc lập với phần mềm thành

các bảng biểu cần thiết và các hạn chế cần thiết khi cập nhật dữ liệu.

1.1.5.4 Thiết kế vật lý

Thiết kế vật lý là quá trình lựa chọn việc lưu trữ dữ liệu và các tính chất của dữ liệu được cập nhật của cơ sở dữ liệu. Thiết kế vật lý không chỉ ảnh hưởng tới vị trí dữ liệu trong thiết bị lưu trữ mà còn cả tới việc thực hiện hệ thống.

Thiết kế vật lý đặc biệt quan trọng đối với mô hình thứ bậc và mô hình mạng. Cơ sở dữ liệu dạng quan hệ có khả năng độc lập so với dạng thiết kế vật lý này hơn hai dạng mô hình trước. Tuy nhiên, trong quá trình thực hiện, bản thân mô hình quan hệ lại ảnh hưởng khá nhiều tới tính chất của dạng thiết kế vật lý. Chẳng hạn như, khi thực hiện, mô hình quan hệ có thể ảnh hưởng tới tính chất của phương tiện lưu trữ dữ liệu như thời gian tìm kiếm, kích cỡ các trang, số các đầu đọc và ghi, v.v...

1.1.6. Ngôn ngữ SQL

1.1.6.1. Khái niệm

SQL được viết tắt từ Structured Query Language, là ngôn ngữ truy vấn có cấu trúc. Về cơ bản, ngôn ngữ truy vấn có cấu trúc này được sử dụng để truy xuất dữ liệu hoặc giao diện khác như tạo, xóa trong cơ sở dữ liệu, lấy các hàng và sửa đổi các hàng với cơ sở dữ liệu quan hệ.

SQL cho phép truy cập và thao tác cơ sở dữ liệu. Nó là ngôn ngữ được dành riêng cho sử dụng trong lập trình và được thiết kế quản lý dữ liệu được giữ trong hệ thống quản lý cơ sở dữ liệu quan hệ hoặc để xử lý luồng trong hệ thống quản lý luồng dữ liệu quan hệ. SQL được sử dụng để giao tiếp với cơ sở dữ liệu, Theo ANSI, SQL là ngôn ngữ tiêu chuẩn cho các hệ thống quản lý cơ sở dữ liệu quan hệ.

1.1.6.2. Các chức năng của SQL

Ngôn ngữ SQL hỗ trợ người dùng rất nhiều trong việc thực thi các truy vấn đối với cơ sở dữ liệu. Nhờ vào SQL, việc truy vấn diễn ra nhanh chóng, chính xác, thực hiện được đúng yêu cầu từ người truy vấn. Các chức năng cơ bản của ngôn ngữ SQL gồm có:

- Lấy dữ liệu từ cơ sở dữ liệu.
- Chèn các bản ghi trong cơ sở dữ liệu.
- Cập nhật các bản ghi trong cơ sở dữ liệu.
- Xóa các bản ghi trong cơ sở dữ liệu.
- Tạo cơ sở dữ liệu mới.
- Tạo các thủ tục được lưu giữ trong cơ sở dữ liệu.
- Tạo các view, stored procedure functions trong cơ sở dữ liệu.
- SQL còn cho phép thiết lập quyền truy cập trên bảng, procedure và view
- Các ngôn ngữ truy vấn dữ liệu SQL.

1.1.6.3. Các ngôn ngữ truy vấn dữ liệu SQL

- DL (Data Definition Language) - ngôn ngữ định nghĩa dữ liệu, chứa các

lệnh như create, alter, drop, truncate.

- DML (Data Manipulation Language) - ngôn ngữ thao tác dữ liệu với các thao tác như insert, update, delete.

- DQL (Data Query Language) - ngôn ngữ truy vấn dữ liệu với lệnh select

- DCL (Data Control Language) - ngôn ngữ kiểm soát dữ liệu với các lệnh như grant, revoke.

- TCL (Transaction Control Language) - ngôn ngữ kiểm soát giao tác với lệnh như commit, rollback, savepoint.

1.1.7. Hệ quản trị SQL SERVER

1.1.7.1. Giới thiệu

SQL Server hay còn gọi là Microsoft SQL Server, viết tắt là MS SQL Server. Đây là một phần mềm được phát triển bởi Microsoft dùng để lưu trữ dữ liệu dựa trên chuẩn RDBMS, cũng là hệ quản trị cơ sở dữ liệu quan hệ đối tượng (ORDBMS). SQL Server là một trong 3 công nghệ dữ liệu dẫn đầu hiện nay cùng với Oracle Database và IBM's DB2.

SQL Server hoạt động là một máy chủ cơ sở dữ liệu. Chức năng chính của nó là lưu trữ và truy xuất dữ liệu theo yêu cầu của các ứng dụng phần mềm. Nó sử dụng câu lệnh SQL để trao đổi dữ liệu giữa máy Client và máy cài SQL Server.

SQL Server cung cấp đầy đủ công cụ để quản lý, từ giao diện GUI cho đến việc sử dụng ngôn ngữ truy vấn SQL. Ngoài ra điểm mạnh của nó là Microsoft nên có khá nhiều nền tảng kết hợp hoàn hảo với SQL Server như ASP.NET, C# xây dựng Winform, bởi vì nó hoạt động hoàn toàn độc lập.

1.1.7.2. Các thành phần cơ bản trong SQL server

Các thành phần cơ bản trong SQL Server gồm có: Reporting Services, Database Engine, Integration Services, Notification Services, Full Text Search Service,... Tất cả kết hợp với nhau tạo thành một giải pháp hoàn chỉnh giúp cho việc phân tích và lưu trữ dữ liệu trở nên dễ dàng hơn.

- Database Engine: Đây là một engine có khả năng chứa dữ liệu ở các quy mô dưới dạng support và table. Ngoài ra, nó còn có khả năng tự điều chỉnh ví dụ: trả lại tài nguyên cho hệ điều hành khi một user log off và sử dụng thêm các tài nguyên của máy khi cần.

- Integration Services: là tập hợp các đối tượng lập trình và các công cụ đồ họa cho việc sao chép, di chuyển và chuyển đổi dữ liệu. Khi bạn làm việc trong một công ty lớn thì dữ liệu được lưu trữ ở nhiều nơi khác nhau như được chứa trong: Oracle, SQL Server, DB2, Microsoft Access,... và bạn chắc chắn sẽ có nhu cầu di chuyển dữ liệu giữa các server này. Ngoài ra, bạn còn muốn định dạng dữ liệu trước khi lưu vào database. Chắc chắn Integration Services sẽ giúp bạn giải quyết được công việc này dễ dàng.

- Analysis Services: Đây là một dịch vụ phân tích dữ liệu rất hay của Microsoft. Dữ liệu khi được lưu trữ vào trong database mà bạn không thể lấy được những thông tin bổ ích thì coi như không có ý nghĩa gì. Chính vì thế, công cụ này ra

đòi giúp bạn trong việc phân tích dữ liệu một cách hiệu quả và dễ dàng bằng cách dùng kỹ thuật khai thác dữ liệu – datamining và khái niệm hình khối nhiều chiều – multi dimention cubes.

- Notification Services: Dịch vụ thông báo này là nền tảng cho sự phát triển và triển khai các ứng dụng soạn và gửi thông báo. Ngoài ra, dịch vụ này còn có chức năng gửi thông báo theo lịch thời đến hàng ngàn người đăng ký sử dụng trên nhiều loại thiết bị khác nhau.

- Reporting Services: là một công cụ tạo, quản lý và triển khai báo cáo bao gồm: server và client. Ngoài ra, nó còn là nền tảng cho việc phát triển và xây dựng các ứng dụng báo cáo.

- Full Text Search Service: là một thành phần đặc biệt trong việc truy vấn và đánh chỉ mục dữ liệu văn bản không cấu trúc được lưu trữ trong các cơ sở dữ liệu SQL Server.

- Service Broker: là một môi trường lập trình cho việc tạo ra các ứng dụng trong việc nhảy qua các Instance.

1.2. Danh sách liên kết đôi

1.2.1 Khái niệm

Trong khoa học máy tính, danh sách liên kết đôi là một cấu trúc dữ liệu được liên kết bao gồm một tập hợp các bản ghi được liên kết tuần tự được gọi là node. Mỗi node chứa ba trường: hai trường liên kết (tham chiếu đến node phía trước và nút tiếp theo trong chuỗi node) và một trường dữ liệu. Các liên kết trước và tiếp theo của các node bắt đầu và kết thúc, tương ứng, trở đến một loại kết thúc nào đó, điển hình là node sentinel hoặc null.

Cấu tạo một nút trong danh sách liên kết đôi được minh họa như sau:



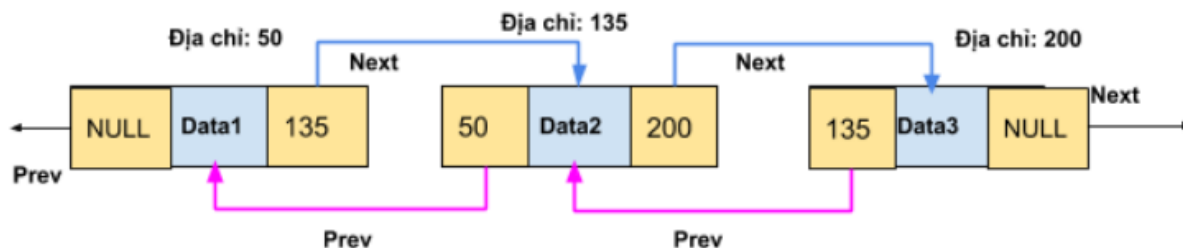
Hình 3. Cấu trúc của một Node

Trong đó:

- Prev: Là con trỏ tới node phía trước.
- Next: Là con trỏ tới node tiếp theo.
- Data: Trường chứa nội dung thông tin của node, kiểu dữ liệu của phần tử là do người dùng định nghĩa. Điều này giúp việc thao tác với danh sách trở nên dễ dàng hơn.

Việc sử dụng hai liên kết node Prev và Next giúp người dùng có thể duyệt danh sách theo hai chiều. Điều này giúp việc thao tác với danh sách trở nên đơn giản và thuận tiện hơn so với danh sách liên kết đơn.

Liên kết các node với nhau ta được danh sách liên kết đôi như sau:



Hình 4. Cấu trúc DSLK đôi

1.2.2 Cấu trúc và các thao tác trên danh sách liên kết đôi

Cấu trúc node của danh sách liên kết đôi:

```

class NODE{
    int data;
    NODE *Next;
    NODE *Prev;
};

class LIST{
    NODE *pHead;
    NODE *pTail;
};
    
```

1.2.3. Các hoạt động cơ bản trên danh sách liên kết đôi:

• Khởi tạo danh sách liên kết rỗng

Để khởi tạo danh sách liên kết rỗng, chúng em sử dụng hàm void Init() để khởi tạo con trỏ pHead, pTail của danh sách và gán cho NULL. void Init(LIST& L, int Level)

```

{
    L.pHead = L.pTail = NULL;
}
    
```

• Tạo node và gán giá trị vào node

Chúng em sử dụng hàm NODE* CreatNode(float DATA) để tạo thông tin cho node với giá trị là DATA.

```

NODE* CreatNode(float DATA)
{
    NODE* p = new NODE;
    if (p == NULL){
        printf("No Memory");
        system("pause");
    }
}
    
```



```

        return NULL;
    }
    else{
        p->data = DATA;
        p->Prev = NULL;
        p->Next = NULL;
        return p;
    }
}

```

- **Kiểm tra danh sách có rỗng hay không**

```

Bool isEmpty(LIST L){
    return (L.pHead == NULL);
}

```

- **Thêm node vào cuối danh sách**

Sử dụng hàm void AddTail(LIST& L, NODE* p) để thêm node p vào cuối danh sách, sử dụng pTail của danh sách trở node cần thêm, đồng thời gán p cho pTail.

```

void AddTail(LIST& L, NODE* p){
    if (L.pHead == NULL) L.pHead = L.pTail = p;
    else{
        L.pTail->Next = p;
        p->Prev = L.pTail;
        L.Tail = p;
    }
}

```

- **Xóa node ở cuối danh sách**

Sử dụng hàm void DelTail(LIST& L) để xóa node cuối danh sách.

```

void DelTail(LIST& L){
    if(!isEmpty(L)){
        NODE *p = L.pTail;
        L.PTail = L.pTail -> Prev;
        L.pTail -> Next = NULL;
        Delete p;
    }
}

```

1.2.4. Ứng dụng của danh sách liên kết đôi

- Bộ đệm của trình duyệt cho phép nhấn các nút BACK hoặc điều hướng

qua các trang trước.

- Các ứng dụng có danh sách được sử dụng gần đây nhất.
- Một ngăn xếp, bảng băm và cây nhị phân có thể được thực hiện bằng cách sử dụng danh sách liên kết đôi.
- Hoàn tác các tính năng trong xuất bản hoặc chỉnh sửa ứng dụng như Photoshop và Word.
- Một cách tuyệt vời để thể hiện một cổ bài trong một trò chơi có trong đó việc loại bỏ vật phẩm từ bất cứ đâu trong bộ bài là điều cần thiết.

1.3. Lập trình hướng đối tượng

1.3.1 Các khái niệm

1.3.1.1 Khái niệm cơ bản về lập trình hướng đối tượng(OOP)

Lập trình hướng đối tượng là một mẫu hình lập trình dựa trên khái niệm "công nghệ đối tượng", mà trong đó, đối tượng chứa đựng các dữ liệu, trên các trường, thường được gọi là các thuộc tính; và mã nguồn, được tổ chức thành các phương thức.

Phương thức giúp cho đối tượng có thể truy xuất và hiệu chỉnh các trường dữ liệu của đối tượng khác, mà đối tượng hiện tại có tương tác (đối tượng được hỗ trợ các phương thức "this" hoặc "self"). Trong lập trình hướng đối tượng, chương trình máy tính được thiết kế bằng cách tách nó ra khỏi phạm vi các đối tượng tương tác với nhau.

Ngôn ngữ lập trình hướng đối tượng khá đa dạng, phần lớn là các ngôn ngữ lập trình theo lớp, nghĩa là các đối tượng trong các ngôn ngữ này được xem như thực thể của một lớp, được dùng để định nghĩa một kiểu dữ liệu.

Các ngôn ngữ lập trình hướng đối tượng đáng chú ý gồm có Java, C++, C#, Python, PHP, Ruby, Perl, Object Pascal, Objective-C, Dart, Swift, Scala, Common Lisp, và Smalltalk.

1.3.2. Các tính chất cơ bản

Lập trình hướng đối tượng là phương pháp lập trình có 4 tính chất chính:

- **Tính trừu tượng**

Đây là khả năng của chương trình bỏ qua hay không chú ý đến một số khía cạnh của thông tin mà nó đang trực tiếp làm việc lên, nghĩa là nó có khả năng tập trung vào những cốt lõi cần thiết. Mỗi đối tượng phục vụ như là một "động tử" có thể hoàn tất các công việc một cách nội bộ, báo cáo, thay đổi trạng thái của nó và liên lạc với các đối tượng khác mà không cần cho biết làm cách nào đối tượng tiến hành được các thao tác. Tính chất này thường được gọi là sự trừu tượng của dữ liệu.

Tính trừu tượng còn thể hiện qua việc một đối tượng ban đầu có thể có một số đặc điểm chung cho nhiều đối tượng khác như là sự mở rộng của nó nhưng bản thân đối tượng ban đầu này có thể không có các biện pháp thi hành. Tính trừu tượng

này thường được xác định trong khái niệm gọi là lớp trừu tượng hay lớp cơ sở trừu tượng.

- **Tính đóng gói và che giấu thông tin**

Tính chất này không cho phép người sử dụng các đối tượng thay đổi trạng thái nội tại của một đối tượng. Chỉ có các phương thức nội tại của đối tượng cho phép thay đổi trạng thái của nó. Việc cho phép môi trường bên ngoài tác động lên các dữ liệu nội tại của một đối tượng theo cách nào là hoàn toàn tùy thuộc vào người viết mã. Đây là tính chất đảm bảo sự toàn vẹn của đối tượng.

- **Tính đa hình**

Thể hiện thông qua việc gửi các thông điệp (message). Việc gửi các thông điệp này có thể so sánh như việc gọi các hàm bên trong của một đối tượng. Các phương thức dùng trả lời cho một thông điệp sẽ tùy theo đối tượng mà thông điệp đó được gửi tới sẽ có phản ứng khác nhau. Người lập trình có thể định nghĩa một đặc tính (chẳng hạn thông qua tên của các phương thức) cho một loạt các đối tượng gần nhau nhưng khi thi hành thì dùng cùng một tên gọi mà sự thi hành của mỗi đối tượng sẽ tự động xảy ra tương ứng theo đặc tính của từng đối tượng mà không bị nhầm lẫn. Ví dụ: khi định nghĩa hai đối tượng "hinh_vuong" và "hinh_tron" thì có một phương thức chung là "chu_vi". Khi gọi phương thức này thì nếu đối tượng là "hinh_vuong" nó sẽ tính theo công thức khác với khi đối tượng là "hinh_tron".

- **Tính kế thừa (inheritance):**

Đặc tính này cho phép một đối tượng có thể có sẵn các đặc tính mà đối tượng khác đã có thông qua kế thừa. Điều này cho phép các đối tượng chia sẻ hay mở rộng các đặc tính sẵn có mà không phải tiến hành định nghĩa lại. Tuy nhiên, không phải ngôn ngữ định hướng đối tượng nào cũng có tính chất này.

1.3.3. Một số khái niệm cần biết trong ngôn ngữ OOP

Hiện nay các ngôn ngữ OOP phổ biến nhất đều tập trung theo phương pháp phân lớp trong đó có C++, Java, C# và Visual Basic.NET. Ngôn ngữ OOP hay còn gọi là ngôn ngữ lập trình hướng đối tượng, là một phương pháp thiết kế và phát triển phần mềm dựa trên kiến trúc lớp và đối tượng. Sau đây là một số khái niệm mà các ngôn ngữ này thường dùng tới.

- **Đối tượng**

Các dữ liệu và chỉ thị được kết hợp vào một đơn vị đầy đủ tạo nên một đối tượng. Đơn vị này tương đương với một chương trình con và vì thế các đối tượng sẽ được chia thành hai bộ phận chính: phần các phương thức và phần các thuộc tính. Các phương thức và các thuộc tính thường gắn chặt với thực tế các đặc tính và sử dụng của một đối tượng.

- **Lớp**

Một lớp có thể được hiểu là khuôn mẫu để tạo ra các đối tượng. Trong một

lớp, người ta thường dùng các biến để mô tả các thuộc tính và các hàm để mô tả các phương thức của đối tượng. Khi đã định nghĩa được lớp, ta có thể tạo ra các đối tượng từ lớp này. Để việc sử dụng được dễ dàng, thông qua hệ thống hàm tạo, người ta dùng lớp như một kiểu dữ liệu để tạo ra các đối tượng.

- **Lớp con**

Lớp con là một lớp thông thường nhưng có thêm tính chất kế thừa một phần hay toàn bộ các đặc tính của một lớp khác. Lớp chia sẻ sự kế thừa gọi là lớp cha.

- **Lớp trừu tượng hay lớp cơ sở trừu tượng**

Lớp trừu tượng là một lớp mà nó không thể thực thể hóa thành một đối tượng thực dụng được. Lớp này được thiết kế nhằm tạo ra một lớp có các đặc tính tổng quát nhưng bản thân lớp đó chưa có ý nghĩa (hay không đủ ý nghĩa) để có thể tiến hành viết mã cho việc thực thể hóa.

- **Phương thức**

Phương thức của một lớp thường được dùng để mô tả các hành vi của đối tượng hoặc của lớp.

Khi thiết kế, người ta có thể dùng các phương thức để mô tả và thực hiện các hành vi của đối tượng. Mỗi phương thức thường được định nghĩa là một hàm, các thao tác để thực hiện hành vi đó được viết tại nội dung của hàm. Khi thực hiện hành vi này, đối tượng có thể phải thực hiện các hành vi khác.

Ví dụ: điện thoại phải chuyển tín hiệu giọng nói sang dạng chuẩn trước khi chuyển lên tổng đài.

Cho nên một phương thức trong một lớp có thể sử dụng phương thức khác trong quá trình thực hiện hành vi của mình. Người ta còn định nghĩa thêm vài loại phương thức đặc biệt:

- Hàm tạo là hàm được dùng để tạo ra một đối tượng, cài đặt các giá trị ban đầu cho các thuộc tính của đối tượng đó.

- Hàm hủy là hàm dùng vào việc làm sạch bộ nhớ đã dùng để lưu đối tượng và hủy bỏ tên của một đối tượng sau khi đã dùng xong, trong đó có thể bao gồm cả việc xóa các con trỏ nội tại và trả về các phần bộ nhớ mà đối tượng đã dùng.

Nhiều lớp thư viện có sẵn hàm tạo mặc định (thông thường không có tham số) và hàm hủy.

- **Thuộc tính**

Thuộc tính của một lớp bao gồm các biến, các hằng, hay tham số nội tại của lớp đó. Ở đây, vai trò quan trọng nhất của các thuộc tính là các biến vì chúng sẽ có thể bị thay đổi trong suốt quá trình hoạt động của một đối tượng.

Các thuộc tính có thể được xác định kiểu và kiểu của chúng có thể là các kiểu dữ liệu cổ điển hay đó là một lớp đã định nghĩa từ trước. Như đã ghi, khi một

lớp đã được thực thể hoá thành đối tượng cụ thể thì tập hợp các giá trị của các biến nội tại làm thành trạng thái của đối tượng. Mỗi thuộc tính của một lớp còn được gọi là thành viên dữ liệu của lớp đó.

• Quan hệ giữa lớp và đối tượng

Lớp trong quan niệm thông thường là cách phân loại các thực thể dựa trên những đặc điểm chung của các thực thể đó. Do đó lớp là khái niệm mang tính trừu tượng hóa rất cao.

Ví dụ như lớp "người" dùng để chỉ những thực thể sống trên Trái Đất có những thuộc tính: có hai chân, hai bàn tay khéo léo, có tư duy, ngôn ngữ v.v và có phương thức: giao tiếp bằng ngôn ngữ, tư duy, đi, đứng bằng hai chân... Khi đó hai người cụ thể ông A, ông B là các đối tượng thuộc lớp người. Trong ngôn ngữ lập trình, ta cũng hiểu khái niệm lớp tương tự, cho nên ta có quá trình "Thực thể hóa" sau, tạo một đối tượng thuộc một lớp đã được ta định nghĩa (phân loại).

• Thực thể

Thực thể hóa là quá trình khai báo để có một tên (có thể được xem như là một biến) trở thành một đối tượng từ một lớp nào đó.

Một lớp sau khi được tiến hành thực thể hóa để có một đối tượng cụ thể gọi là một thực thể. Hay nói ngược lại một thực thể là một đối tượng riêng lẻ của một lớp đã định trước. Như các biến thông thường, hai thực thể của cùng một lớp có thể có trạng thái nội tại khác nhau (xác định bởi các giá trị hiện có của các biến nội tại) và do đó hoàn toàn độc lập nhau nếu không có yêu cầu gì đặc biệt từ người lập trình. Thực thể hóa: gần giống như cá nhân hóa. Một lớp khi được "cá nhân hóa" sẽ thành một đối tượng cụ thể.

• Công cộng

Công cộng là một tính chất được dùng để gán cho các phương thức, các biến nội tại, hay các lớp mà khi khai báo thì người lập trình đã cho phép các câu lệnh bên ngoài cũng như các đối tượng khác được phép dùng đến nó.

• Riêng tư

Riêng tư là sự thể hiện tính chất đóng mạnh nhất (của một đặc tính hay một lớp). Khi dùng tính chất này gán cho một biến, một phương thức thì biến hay phương thức đó chỉ có thể được sử dụng bên trong của lớp mà chúng được định nghĩa. Mọi nỗ lực dùng trực tiếp đến chúng từ bên ngoài qua các câu lệnh hay từ các lớp con sẽ bị phủ nhận hay bị lỗi.

• Bảo tồn

Tùy theo ngôn ngữ, sẽ có vài điểm nhỏ khác nhau về cách hiểu tính chất này. Nhìn chung đây là tính chất mà khi dùng để áp dụng cho các phương thức, các biến nội tại, hay các lớp thì chỉ có trong nội bộ của lớp đó hay các lớp con của nó

(hay trong nội bộ một gói như trong Java) được phép gọi đến hay dùng đến các phương pháp, biến hay lớp đó.

So với tính chất riêng tư thì tính bảo tồn rộng rãi hơn về nghĩa chia sẻ dữ liệu hay chức năng. Nó cho phép một số trường hợp được dùng tới các đặc tính của một lớp (từ một lớp con chẳng hạn).

1.3.4. Ưu nhược điểm của lập trình hướng đối tượng

Ưu điểm của lập trình hướng đối tượng OOP:

- Người dùng có thể dễ dàng quản lý code khi có sự thay đổi chương trình. Thay vì việc thay đổi các mã nguồn của đối tượng mà giờ đây lập trình viên chỉ cần thay đổi một số hàm thành phần.

- Dễ dàng mở rộng dự án.
- Giúp lập trình viên tiết kiệm đáng kể tài nguyên cho hệ thống.
- OOP có tính bảo mật cao và có khả năng tái sử dụng cao.
- Phù hợp khi thiết lập các phần mềm phức tạp, phần mềm lớn. OOP được đánh giá là dễ học, năng suất, thao tác đơn giản và dễ bảo trì.

Nhược điểm của phương pháp lập trình đối tượng OOP có thể khiến dữ liệu được xử lý tách rời, khi cấu trúc dữ liệu thay đổi sẽ dẫn đến việc thuật toán bị thay đổi theo. Bên cạnh đó, OOP không tự động khởi tạo và giải phóng dữ liệu động, không mô tả được đầy đủ và trung thực hệ thống trong thực tế.

CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

2.1. Khảo sát hiện trạng

Để xây dựng nên một hệ thống hoàn chỉnh và tối ưu, khảo sát hiện trạng là một bước vô cùng quan trọng, là tiền đề để thiết kế nên hệ thống đáp ứng đầy đủ và chính xác các nhu cầu trong thực tế.

Sau khi khảo sát và đánh giá, chúng em đã đưa ra các thông tin:

- Quản lý sinh viên: Mỗi sinh viên được quản lý thông tin: mã sinh viên, họ tên, khoa, lớp, số điện thoại.
- Quản lý sách: Sách ở đây được coi như là một đầu sách. Mỗi sách gồm các thông tin: mã sách, tên sách, tác giả, thể loại, số lượng gốc, số lượng còn.
- Quản lý mượn trả: Quản lý thông tin mượn trả sách của thư viện. Thông tin mượn trả gồm: thứ tự mượn trả, mã sinh viên, mã sách, ngày mượn, ngày trả, trạng thái.
- Quản lý loại sách: Thông tin sách gồm: mã loại, tên loại.

Chức năng chính:

- Đăng ký: Sinh viên đến mượn sách, nếu chưa đăng ký thông tin sinh viên thì phải đăng ký vào hệ thống.
- Mượn sách: Sinh viên mượn sách phải khai báo mã sinh viên, mã sách và thủ tục thực hiện cho đăng ký mượn.
- Trả sách: Sinh viên trả sách phải khai báo mã sinh viên và thủ tục căn cứ theo mã sinh viên, sách mượn, ngày mượn để thiết lập thông tin đã trả sách.
- Các chức năng thêm, tìm kiếm, xóa, cập nhật, sắp xếp thông tin.

2.2. Thiết kế cơ sở dữ liệu

2.2.1. Xác định thực thể

2.2.1.1 Thực thể BOOK

Chứa thông tin chi tiết về sách trong thư viện.

Các thuộc tính: CodeBook, NameBook, CodeType, Author, OriginQuantity, RemainQuantity.

- Mã sách (CodeBook): Đây là thuộc tính khóa dùng để phân biệt các sách.
- Tên sách (NameBook): Mô tả tên sách.
- Mã loại sách (CodeType): Cho biết mã loại của sách nhằm xác định tên thể loại sách.
- Tác giả (Author) : Cho biết tên tác giả viết sách
- Số lượng ban đầu của sách (OriginQuantity): Cho biết số lượng sách có cùng tên trong thư viện.
- Số lượng còn của sách (RemainQuantity): Cho biết số lượng sách có cùng đầu tên còn lại trong thư viện.

2.2.1.2. Thực thể BORROWRETURN

Chứa thông tin liên quan đến lịch sử mượn và trả của người dùng.

Các thuộc tính: Orders, CodeStudent, CodeBook, DateBorrow, DateReturn, Status.

- Đơn mượn sách (Orders): Đây là thuộc tính khóa dùng để phân biệt các đơn mượn sách.

- Mã sinh viên mượn sách (CodeStudent).

- Mã sách được mượn (CodeBook).

- Ngày bắt đầu mượn (DateBorrow), ngày kết thúc mượn (DateReturn): Đây là 2 thuộc tính cho biết khoảng thời gian có thể mượn sách.

- Trạng thái (Status): thuộc tính cho biết sách của đơn mượn sách tương ứng đó được trả hay chưa.

2.2.1.3. Thực thể STUDENT

Chứa thông tin của sinh viên.

Các thuộc tính: CodeStudent, NameStudent, Faculty, Class, PhoneNumber.

- Mã sinh viên (CodeStudent): Đây là thuộc tính khóa, nhờ thuộc tính này mà ta phân biệt được các sinh viên.

- Tên sinh viên (NameStudent): Cho biết tên của sinh viên.

- Khoa (Faculty): Đây là thuộc tính cho biết sinh viên thuộc khoa nào.

- Lớp (Class): Cho biết lớp học mà sinh viên đang học.

- Số điện thoại (PhoneNumber): Số điện thoại để liên lạc với sinh viên.

2.2.1.4. Thực thể TYPEBOOK

Chứa thông tin chi tiết về loại sách.

Các thuộc tính: CodeType, NameType.

- Mã loại sách (CodeType): Đây là thuộc tính khóa, nhờ thuộc tính này mà ta phân loại được các loại sách.

- Tên loại sách (NameType): Thuộc tính này cho biết tên thể loại sách.

2.2.2 Mô tả thực thể

• Thực thể BOOK

	Column Name	Data Type	Allow Nulls
🔑	CodeBook	nchar(11)	<input type="checkbox"/>
	NameBook	nchar(40)	<input checked="" type="checkbox"/>
	CodeType	nchar(15)	<input checked="" type="checkbox"/>
	Author	nchar(30)	<input checked="" type="checkbox"/>
	OriginQuantity	int	<input checked="" type="checkbox"/>
	RemainQuantity	int	<input checked="" type="checkbox"/>

Hình 5. Thực thể BOOK

• **Thực thể BORROWRETURN**

	Column Name	Data Type	Allow Nulls
🔑	Orders	nchar(10)	<input type="checkbox"/>
	CodeStudent	nchar(15)	<input checked="" type="checkbox"/>
	CodeBook	nchar(15)	<input checked="" type="checkbox"/>
	DateBorrow	date	<input checked="" type="checkbox"/>
	DateReturn	date	<input checked="" type="checkbox"/>
	Status	nchar(15)	<input checked="" type="checkbox"/>

Hình 6. Thực thể BORROWRETURN

• **Thực thể STUDENT**

	Column Name	Data Type	Allow Nulls
🔑	CodeStudent	nchar(9)	<input type="checkbox"/>
	NameStudent	nchar(35)	<input checked="" type="checkbox"/>
	Faculty	nchar(30)	<input checked="" type="checkbox"/>
	Class	nchar(20)	<input checked="" type="checkbox"/>
	PhoneNumber	nchar(20)	<input checked="" type="checkbox"/>

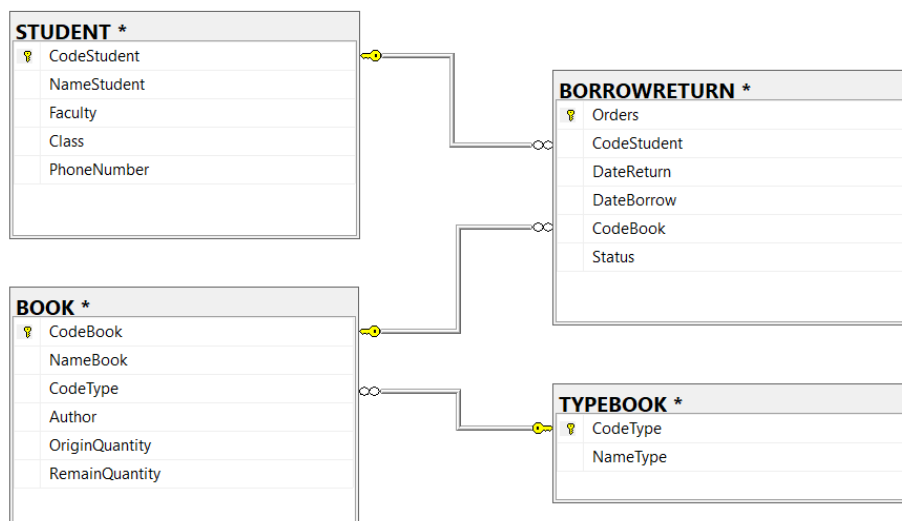
Hình 7. Thực thể STUDENT

• **Thực thể TYPEBOOK**

	Column Name	Data Type	Allow Nulls
🔑	CodeType	nchar(15)	<input type="checkbox"/>
	NameType	nchar(50)	<input checked="" type="checkbox"/>

Hình 8. Thực thể TYPEBOOK

2.2.3. Mô hình quan hệ ERD



Hình 9. Mô hình quan hệ ERD

2.2.4. Chuẩn hóa lược đồ quan hệ về dạng 3NF

- STUDENT(CodeStudent, NameStudent, Faculty, Class, PhoneNumber).
 - $F1 = \{ \text{NameStudent} \rightarrow \text{CodeStudent}, \text{Faculty} \rightarrow \text{CodeStudent}, \text{Class} \rightarrow \text{CodeStudent}, \text{PhoneNumber} \rightarrow \text{CodeStudent} \}$.
 - Lược đồ quan hệ chỉ chứa các thuộc tính nguyên tố: lược đồ dạng chuẩn 1NF.
 - Mã CodeStudent là khóa dự bị, các thuộc tính không khóa phụ thuộc hàm đầy đủ vào CodeStudent nên lược đồ dạng chuẩn 2NF.
 - Do $\text{CodeStudent} \rightarrow \text{NameStudent}$, $\text{CodeStudent} \rightarrow \text{Faculty}$, $\text{CodeStudent} \rightarrow \text{PhoneNumber}$, CodeStudent là siêu khóa nên lược đồ đạt chuẩn 3NF.
- BOOK(CodeBook, NameBook, CodeType, Author, OriginQuantity, RemainQuantity).
 - $F2 = \{ \text{CodeBook} \rightarrow \text{NameBook}, \text{CodeBook} \rightarrow \text{CodeType}, \text{CodeBook} \rightarrow \text{Author}, \text{CodeBook} \rightarrow \text{OriginQuantity}, \text{CodeBook} \rightarrow \text{RemainQuantity} \}$.
 - Lược đồ quan hệ chỉ chứa các thuộc tính nguyên tố: lược đồ ở dạng chuẩn 1NF.
 - CodeBook là khóa dự bị, các thuộc tính không khóa phụ thuộc hàm đầy đủ vào CodeBook nên lược đồ có dạng chuẩn 2NF.
 - Do $\text{CodeBook} \rightarrow \text{NameBook}$, $\text{CodeBook} \rightarrow \text{CodeType}$, $\text{CodeBook} \rightarrow \text{Author}$, $\text{CodeBook} \rightarrow \text{OriginQuantity}$, $\text{CodeBook} \rightarrow \text{RemainQuantity}$, CodeBook là siêu khóa, nên lược đồ ở dạng chuẩn 3NF.
- BORROWRETURN(Orders, CodeStudent, CodeBook, DateBorrow, DateReturn, Status).
 - $F3 = \{ \text{Order} \rightarrow \text{DateBorrow}, \text{Orders} \rightarrow \text{Status} \}$.
 - Lược đồ quan hệ chứa các thuộc tính nguyên tố: lược đồ dạng chuẩn 1NF.
 - Orders là khóa dự bị, các thuộc tính không khóa phụ thuộc hàm đầy đủ vào Orders nên lược đồ ở dạng chuẩn 2NF.
 - Do $\text{Order} \rightarrow \text{DateBorrow}$, $\text{Orders} \rightarrow \text{Status}$, Order là siêu khóa nên lược đồ đạt dạng chuẩn 3NF.
- TYPEBOOK(CodeType, NameType).
 - $F4 = \{ \text{CodeType} \rightarrow \text{NameType} \}$.
 - Lược đồ quan hệ chứa các thuộc tính nguyên tố: lược đồ ở dạng chuẩn 1NF.
 - CodeType là khóa dự bị, thuộc tính NameType phụ thuộc hàm đầy đủ vào SĐT: lược đồ ở dạng chuẩn 2NF.
 - Do $\text{CodeType} \rightarrow \text{NameType}$, CodeType là khóa dự bị nên lược đồ ở dạng chuẩn 3 NF.

2.3. Thiết kế các class

2.3.1. Thuộc tính

Theo như hướng giải quyết của nhóm, vì chúng em sử dụng danh sách liên kết nên ở đây chúng em đã thiết kế các class sao cho các thuộc tính đồng nhất với các thuộc tính của các bảng trong cơ sở dữ liệu.

Class	Student	Book	BorrowReturn	TypeBook
Thuộc tính	string CodeStudent; string NameStudent; string Faculty; string Class; string PhoneNumber;	string CodeBook; string NameBook; string CodeType; string Author; int OriginQuantity; int RemainQuantity;	string Order; string CodeStudent; string CodeBook; Dates DateBorrow; Dates DateReturn; string Status;	string CodeType; string NameType;

Bảng 1. Thuộc tính các class

2.3.2. Phương thức

Vì các class đều có chung các phương thức bao gồm: Hàm dựng, hủy, hàm khởi tạo giá trị của thuộc tính, hàm lấy giá trị của thuộc tính và các hàm đa năng hóa. Nên chúng em xin trình bày chung các phương thức của các class. Dưới đây là một ví dụ các phương thức của class Book.

Phương thức	Ví dụ
Hàm tạo và hàm hủy	Book(); Book(const Book&); Book(string, string, string, string, int, int); ~Book();
Hàm khởi giá trị và lấy giá trị thuộc tính	string GetCodeBook(); string GetNameBook(); string GetCodeType(); string GetAuthor(); int GetOriQua(); int GetReQua(); void SetCodeBook(string); void SetNameBook(string); void SetCodeType(string); void SetAuthor(string); void SetOriQua(int); void SetReQua(int);

Hàm đa năng hóa	<pre> friend ostream& operator<<(ostream&, const Book&); friend istream& operator>>(istream&, Book&); Book& operator=(Book const&); friend bool operator!=(const Book&, const Book&); friend bool operator==(const Book&, const Book&); friend bool operator<(const Book&, const Book&); friend bool operator>(const Book&, const Book&); friend bool operator<=(const Book&, const Book&); friend bool operator>=(const Book&, const Book&); </pre>
------------------------	--

Bảng 2. Ví dụ phương thức BOOK

2.4. Cấu trúc dữ liệu

Như đã trình bày ở chương 1, chúng em sử dụng danh sách liên kết đôi để lưu trữ và thao tác. Sau đây, chúng em xin trình bày về cấu trúc của node và danh sách liên kết đôi. Vì sử dụng danh sách liên kết đôi nhưng chúng em có 4 class nên không thể tạo 4 danh sách liên kết đôi của các class khác nhau mà ở đây chúng em sử dụng template để tạo cấu trúc và thao tác cho 1 danh sách liên kết mẫu và từ danh sách liên kết mẫu đó có thể sử dụng chung cho 4 class.

2.4.1. Class Node

```

template <class T>
class Node {
private:
    Node<T>* pNext;
    Node<T>* pPre;
    T Data;
public:
    Node();
    ~Node();
    Node(T&);

    Node<T>* getPNext();
    Node<T>* getPPre();
    T getData();

    void setpNext(Node<T>*);
    void setpPre(Node<T>*);
    void setData(T);
    Node<T> operator=(const Node<T>&);
    Node<T>* operator=(const Node<T>* &);
};

```

2.4.2. Cấu trúc node

Cấu trúc node gồm một con trỏ pNext, một con trỏ pPre và một biến Data kiểu dữ liệu T.

Biến	Chức năng
Node<T>* pNext	Biến con trỏ pNext trỏ đến node tiếp theo trong danh sách liên kết.
Node<T>* pPre	Biến con trỏ pPre trỏ đến node phía trước trong danh sách liên kết.
T Data	Biến Data nhằm lưu trữ dữ liệu cho node.

Bảng 3. Cấu trúc node

2.4.3. Các phương thức trên node

Node sử dụng các phương thức gồm:

- Khởi tạo node
- Hủy node
- Bộ phương thức set và get nhằm thiết lập và lấy giá trị từ node
- Toán tử = nhằm sao chép nội dung node.

2.4.4. Cấu trúc danh sách liên kết đôi

Trong đồ án cơ sở lần này, chúng em đã chọn cấu trúc danh sách liên kết đôi để phục vụ việc lưu trữ dữ liệu và thao tác với dữ liệu. Với những ưu điểm vốn có, danh sách liên kết đôi đã giải quyết tối ưu vấn đề lưu trữ dữ liệu và đơn giản hóa việc thao tác với danh sách. Áp dụng vào đồ án chúng em đã xây dựng danh sách liên kết với cấu trúc như sau:

```
template<class T>
class DoubleLinkedList
{
private:
    Node<T>* pHead;
    Node<T>* pTail;
    int length;
public:
    DoubleLinkedList();
    ~DoubleLinkedList();
    Node<T>* getpHead();
    ....
}
```

Trong đó :

- pHead, pTail lần lượt là các con trỏ lưu trữ địa chỉ node đầu, cuối.
- Length là độ dài của danh sách liên kết (length >= 0).
- Node là kiểu dữ liệu được định nghĩa ở mục “ 2.4.2”.

2.4.5. Thao tác trên danh sách liên kết đôi

2.4.5.1. Thao tác lấy độ dài danh sách

Phương thức
template<class T> int DoubleLinkedList<T>::getLength()
Input: Danh sách liên kết đôi. Output: Số phần tử có trong danh sách.

Bảng 4. Thao tác lấy độ dài danh sách liên kết

2.4.5.2. Thao tác in danh sách liên kết

Phương thức
template<class T> void DoubleLinkedList<T>::ShowFirst()
Input: Danh sách liên kết đôi. Output: In ra danh sách liên kết.

Bảng 5. Thao tác in danh sách liên kết

2.4.5.3. Thao tác in các phần tử có giá trị thuộc tính xác định

Phương thức
template<class T> template<class U> inline void DoubleLinkedList<T>::ShowBy(bool (*cmp)(T, U), U mAttribute)
Input: Danh sách liên kết đôi và giá trị thuộc tính mAttribute với kiểu dữ liệu U. Output: In danh sách các phần tử mà trong đó thuộc tính U có giá trị là mAttribute.

Bảng 6. Thao tác in các phần tử có giá trị thuộc tính xác định

2.4.5.4. Thao tác tìm kiếm danh sách liên kết theo thuộc tính

Phương thức
template<class T> template <classU> Node<T>*DoubleLinkedList<T>::SearchBy(bool(*cmp)(T, U), U mAttribute)
Input: Danh sách liên kết đôi và giá trị thuộc tính mAttribute với kiểu dữ liệu U. Output: Trả về node chứa giá trị thuộc tính mAttribute nếu tìm thấy trên danh sách liên kết.

Bảng 7. Thao tác tìm kiếm danh sách liên kết theo thuộc tính

2.4.5.5. Thao tác chèn node vào cuối danh sách liên kết

Phương thức
template<class T> void DoubleLinkedList<T>::InsertLast(T mData)

Input: Danh sách liên kết đôi, phần tử mData có kiểu dữ liệu là T.
Output: Chèn vào cuối danh sách liên kết một node mang giá trị mData có kiểu giá trị T.

Bảng 8. Thao tác chèn node vào cuối danh sách liên kết

2.4.5.6. Thao tác chèn phần tử trước node có giá trị thuộc tính xác định

Phương thức
<pre>template<class T> template <class U> inline bool DoubleLinkedList<T>::InsertBeforeBy(bool(*cmp)(T, U), U mAttribute, T mData)</pre>
<p>Input: Danh sách liên kết đôi và giá trị thuộc tính mAttribute với kiểu dữ liệu U, và giá trị mData kiểu T. Output: Chèn phần tử có giá thuộc tính mData trước node có giá trị thuộc tính mAttribute kiểu dữ liệu U.</p>

Bảng 9. Thao tác chèn phần tử trước node có giá trị thuộc tính xác định

2.4.5.7. Thao tác xóa node có giá trị thuộc tính xác định

Phương thức
<pre>template<class T> template <class U> inline bool DoubleLinkedList<T>::RemoveBy(bool(*cmp)(T, U), U mAttribute)</pre>
<p>Input: Danh sách liên kết đôi và giá trị thuộc tính mAttribute với kiểu dữ liệu U. Output: Xóa node chứa giá trị thuộc tính mAttribute nếu tìm thấy trên danh sách liên kết.</p>

Bảng 10. Thao tác xóa node có giá trị thuộc tính xác định

2.4.5.8. Thao tác cập nhật lại một node

Phương thức
<pre>template<class T> void DoubleLinkedList<T>::UpdateVal(Node<T>* it, T mData)</pre>
<p>Input: Danh sách liên kết đôi, và phần tử có dữ liệu T. Output: Cập nhật giá trị mData có kiểu T cho node it.</p>

Bảng 11. Thao tác cập nhật lại một node

2.4.5.9. Thao tác cập nhật node tại vị trí xác định bởi thuộc tính

Phương thức
<pre>template<class T> template <class U> bool DoubleLinkedList<T>::UpdateBy(bool(*cmp)(T, U), U mAttribute, T mData)</pre>

Input: Danh sách liên kết đôi và giá trị thuộc tính mAttribute với kiểu dữ liệu U, giá trị mData kiểu T.
Output: Cập nhật node chứa giá trị thuộc tính mAttribute thành node mang giá trị mData mang kiểu T.

Bảng 12. Thao tác cập nhật node tại vị trí xác định bởi thuộc tính

2.4.5.10. Thao tác sắp xếp danh sách liên kết

Phương thức
template<class T> void DoubleLinkedList<T>::Sort()
Input: Danh sách liên kết đôi. Output: Danh sách liên kết sau khi đã được sắp xếp.

Bảng 13. Thao tác sắp xếp danh sách liên kết

2.5. Phân tích thiết kế giải thuật

Trong đề tài, chúng em sử dụng giải thuật tìm kiếm tuần tự và sắp xếp chọn trên danh sách liên kết đôi.

2.5.1. Tìm kiếm tuần tự

Algorithm SearchVal(T target)
Input: target: dữ liệu cần tìm kiếm Output: Vị trí được tìm thấy
begin Node<T>* it ← pHead->getpNext(); while (it && it->getData() ≠ target) do it ← it->getpNext(); end while if !it then In thông báo endif return it; end
Độ phức tạp: O(n)

Bảng 14. Giải thuật tìm kiếm tuần tự

2.5.2. Sắp xếp chèn

Algorithm DoubleLinkedList<T>::Sort()
Input: Danh sách cần sắp xếp Output: Danh sách được sắp xếp
begin if (pHead->getpNext() != NULL) then if (pHead->getpNext()->getpNext() != NULL) then

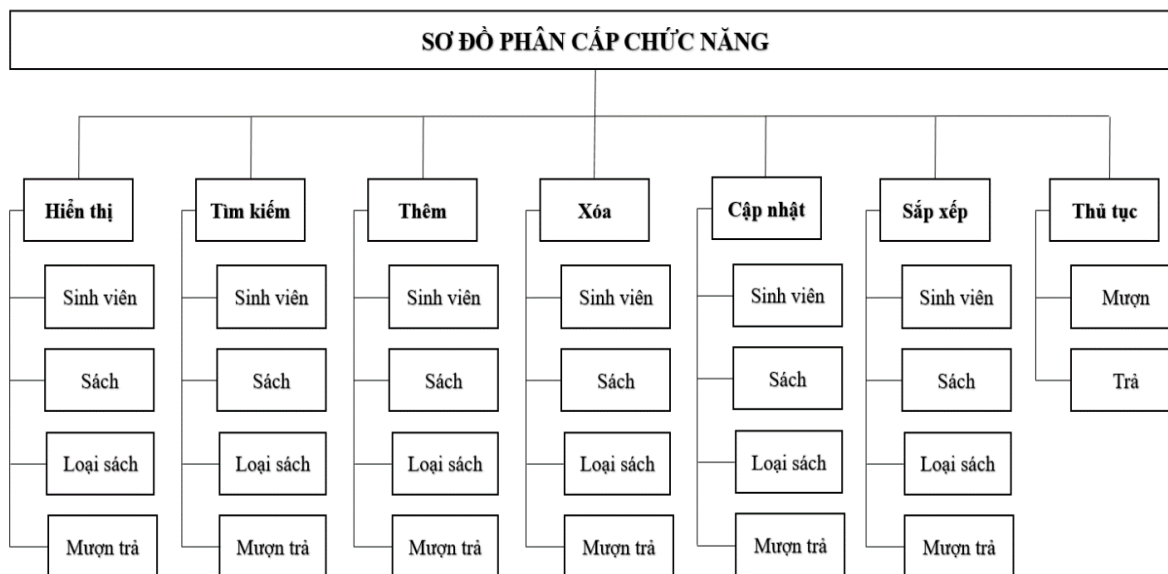
<pre>{ Node<T>* it1 = pHead->getpNext(); Node<T>* it2; while (it1->getpNext()) do { it2 ← it1->getpNext(); while (it2) do { if (it1->getData() > it2->getData()) then Swap(it1, it2); endif it2 ← it2->getpNext(); } endwhile t1 ← it1->getpNext(); } endwhile } endif end</pre>
Độ phức tạp: $O(n^2)$

Bảng 15. Giải thuật sắp xếp chèn

CHƯƠNG 3: TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ

3.1. Chức năng

3.1.1. Sơ đồ phân cấp chức năng



Hình 10. Sơ đồ phân cấp chức năng

3.1.2. Phân tích yêu cầu phi chức năng

- Giao diện dễ sử dụng, có tính thẩm mỹ cao.
- Đảm bảo bí mật thông tin.
- Hình thức thông tin hệ thống chính xác, uy tín.
- Các thao tác nhanh chóng, chính xác theo nhiều tiêu chí.

3.1.3. Chức năng hiển thị

Chức năng hiển thị nhằm hiển thị các thông tin như danh sách sinh viên, danh sách sách, danh sách loại sách, danh sách mượn trả. Ở mỗi thông tin, chương trình sẽ thực hiện duyệt qua danh sách liên kết đôi và in ra toàn bộ thông tin ở danh sách liên kết dưới dạng bảng.

- Đầu vào: Danh sách liên kết đôi.
- Đầu ra: In thông tin danh sách thông tin dưới dạng bảng.
- Xử lý:
 - Duyệt qua danh sách liên kết.
 - In thông tin từng node.

3.1.4. Chức năng tìm kiếm

Chức năng tìm kiếm nhằm tìm kiếm và in ra thông tin cần tìm. Chúng em đã thiết kế để tìm kiếm được theo bất kì các thuộc tính của class. Để tìm kiếm, chương

trình sẽ thực hiện duyệt qua danh sách liên kết đôi, nếu tìm thấy thì in thông tin đó ra, nếu không tìm thấy thì thông báo không tìm thấy.

- Đầu vào: Danh sách liên kết đôi, thông tin thuộc tính cần tìm.
- Đầu ra: In ra thông tin nếu tìm thấy và in thông báo nếu không tìm thấy.
- Xử lý:
 - Duyệt qua danh sách liên kết.
 - Nếu tìm thấy thông tin thì in thông tin đó .
 - Nếu không tìm thấy thì in thông báo.

3.1.5. Chức năng thêm

Chức năng thêm nhằm thêm và in ra thông tin đã thêm. Để thực hiện thêm, chương trình sẽ thêm vào cuối danh sách liên kết và in ra thông tin được thêm. Chức năng thêm áp dụng cho thêm sinh viên, sách, loại sách, thông tin mượn trả.

- Đầu vào: Danh sách liên kết, thông tin đối tượng cần thêm.
- Đầu ra: Danh sách liên kết đã được thêm, in thông tin cần thêm.
- Xử lý:
 - Thực hiện thêm đối tượng vào cuối danh sách liên kết.
 - In ra danh sách chứa thông tin nếu thêm thành công và in thông báo nếu thông tin đã có.

3.1.6. Chức năng xóa

Chức năng xóa nhằm xóa và in ra danh sách sau khi đã xóa. Để thực hiện xóa, chương trình sẽ duyệt qua danh sách liên kết và xóa thông tin nếu tìm thấy thuộc tính của thông tin cần xóa. Chức năng xóa áp dụng cho sinh viên, sách, loại sách, thông tin mượn trả.

- Đầu vào: Danh sách liên kết, thuộc tính của đối tượng cần xóa.
- Đầu ra: Danh sách liên kết đã được xóa thông tin, in danh sách liên kết sau khi đã xóa.
- Xử lý:
 - Thực hiện duyệt danh sách liên kết, tìm kiếm thông tin cần xóa.
 - Xóa thông tin .
 - In ra danh sách nếu xóa thành công và in thông báo nếu không tìm thấy thông tin.

3.1.7. Chức năng cập nhật

Chức năng cập nhật nhằm cập nhật và in ra danh sách chứa thông tin đã được cập nhật. Để thực hiện cập nhật, chương trình sẽ duyệt qua danh sách liên kết để tìm kiếm thông tin chứa thuộc tính cần cập nhật, sau đó thực hiện cập nhật lại thông tin. Cuối cùng, in ra danh sách chứa thông tin đã được cập nhật. Chức năng cập nhật áp dụng cho sinh viên, sách, loại sách, thông tin mượn trả.

- Đầu vào: Danh sách liên kết, thông tin cần cập nhật.
- Đầu ra: Danh sách chứa thông tin đã được cập nhật.
- Xử lý:
 - Thực hiện duyệt danh sách liên kết, tìm thông tin cần cập nhật theo khóa
 - Cập nhật lại thông tin.
 - In ra danh sách nếu cập nhật thành công và in thông báo nếu không tìm thấy thông tin.

3.1.8. Chức năng sắp xếp

Chức năng sắp xếp thực hiện sắp xếp thông tin trong danh sách liên kết. Chúng em đã thiết kế và thực hiện sắp xếp theo các thuộc tính bất kỳ của đối tượng. Ở đây, chúng em thực hiện giải thuật sắp xếp chọn như đã trình bày ở mục trước. Sau khi sắp xếp, chương trình in ra danh sách đã được sắp xếp.

- Đầu vào: Danh sách liên kết cần sắp xếp.
- Đầu ra: Danh sách liên kết được sắp xếp.
- Xử lý:
 - Thực hiện sắp xếp chèn theo thuộc tính bất kì.
 - In ra danh sách được sắp xếp.

3.1.9. Chức năng mượn sách

Chức năng mượn giúp thêm thông tin mượn trả khi có sinh viên đến mượn sách. Đầu tiên sẽ thực hiện nhập mã sinh viên. Nếu sinh viên chưa có trong dữ liệu thì thực hiện đăng ký. Nếu sinh viên đã có thông tin rồi thì thực hiện đăng kí thông tin mượn sách. Cuối cùng, in ra thông tin mượn trả.

- Đầu vào: Danh sách thông tin mượn trả.
- Đầu ra: Danh sách thông tin mượn trả đã được thêm thông tin.
- Xử lý
 - Nhập mã sinh viên, nếu sinh viên đã có thì tiếp tục, nếu chưa thì thực hiện đăng ký.
 - Nhập thông tin mượn trả.
 - Thêm thông tin mượn trả vào danh sách thông tin mượn trả.
 - In ra danh sách chứa thông tin đã được thêm.

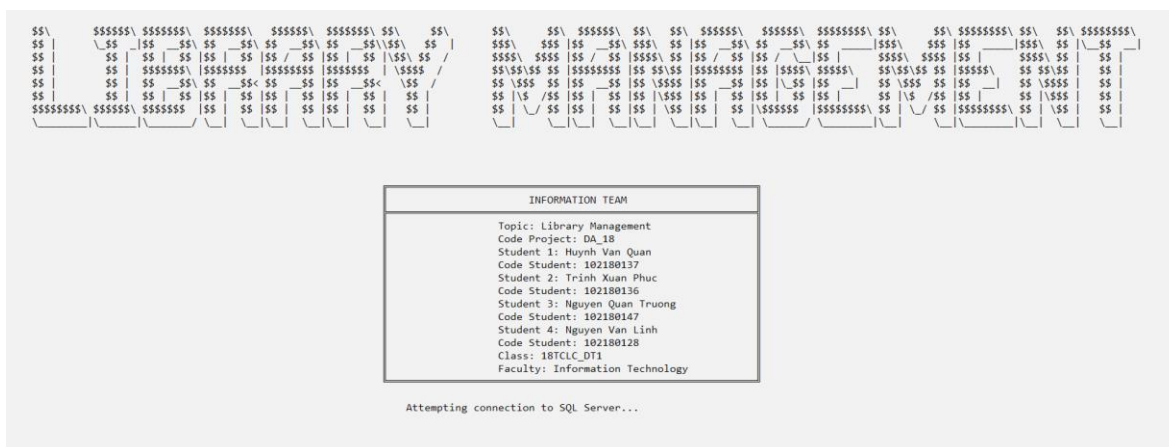
3.1.10. Chức năng trả sách

Chức năng trả sách giúp điều chỉnh thông tin mượn trả khi có sinh viên trả sách. Đầu tiên sẽ thực hiện nhập mã sinh viên. Sau đó, thực hiện cập nhật là thông tin ngày trả và trạng thái của thông tin mượn trả. Cuối cùng, in ra thông tin mượn trả.

- Đầu vào: Danh sách thông tin mượn trả.
- Đầu ra: Danh sách thông tin mượn trả chứa thông tin đã được chỉnh sửa.
- Xử lý:
 - Nhập mã sinh viên.
 - Tìm kiếm và thực hiện chỉnh sửa thông tin mượn trả.
 - In ra danh sách chứa thông tin đã được chỉnh sửa.

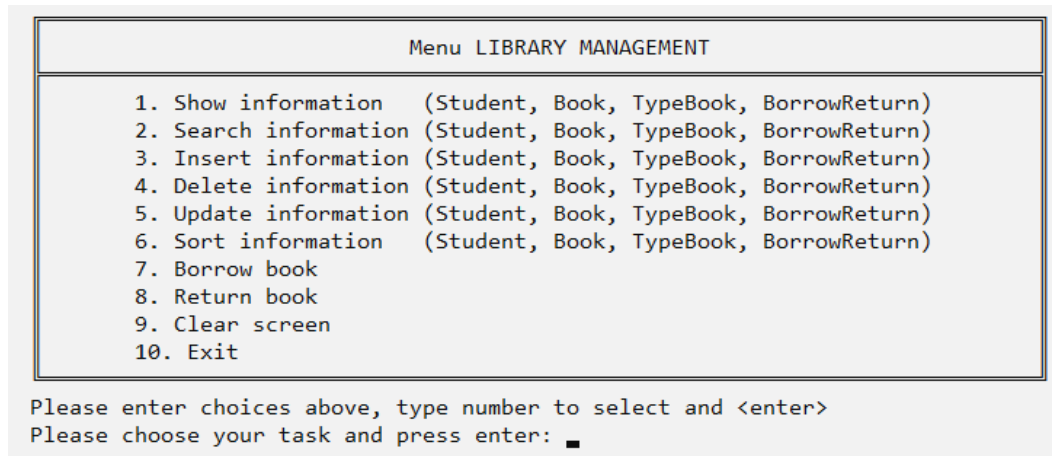
3.2. Trình bày giao diện

3.2.1. Giao diện mở đầu



Hình 11. Giao diện chính

3.2.2. Menu chính



Hình 12. Menu chính

3.2.3. Menu hiển thị thông tin

MENU SHOW INFORMATION

1. Show information student
2. Show information book
3. Show information type book
4. Show information borrow return
5. Clear screen
6. Return
7. Exit

Please enter choices above, type number to select and <enter>
Select your choice and press enter:

Hình 13. Menu hiển thị thông tin

3.2.4. Hiển thị thông tin sách

DATA BOOK						
Order	Code Book	Name Book	Code Type	Author	Origin Quantity	Remain Quantity
1	CT-001	De men phieu luu ky	TN	To Hoai	5	5
2	KH-001	Nguon goc cac loai	KH	Charles Darwin	9	9
3	KH-002	Lich su van vat	KH	Bill Bryson	4	4
4	KH003	Y tuong ve tuong lai	KH	Richard Watson	3	3
5	KH-003	Van vat van hanh	KH	David Macaulay	5	5
6	KN-001	Toi tai gioi ban cung the	KN	Adam Khoo	9	9
7	KN-002	Ky nang thuyet trinh	KN	Tran Cong Ninh	6	6

Hình 14. Hiển thị thông tin sách

3.2.5. Hiển thị thông tin sinh viên

DATA STUDENT					
Order	Code Student	Name Student	Faculty	Class	Phone Number
1	101170303	Tran Thanh Tam	Co khi	17CDT2	0122999456
2	102180137	Huynh Van Quan	Cong nghe thong tin	18TCLC_DT1	0367140514
3	102180156	Nguyen Van Nam	Cong nghe thong tin	18TCLC_DT2	0345222333
4	102180233	Nguyen Van Du	Cong nghe thong tin	18TCLC_DT1	0905051213
5	103170116	Tran Thu Ha	Co khi giao thong	17CKGT4	0357951258
6	103190206	Phan Nhu Tam	Co khi giao thong	19CKGT1	0159862347
7	105190206	Do Nhu Van	Dien	19D3	0125206307
8	107180300	Dinh Cong Tru	Hoa	18H1	0147258369
9	107190118	Doan Cong Tam	Hoa	18H4	0369852147

Hình 15. Hiển thị thông tin sinh viên

3.2.6. Hiển thị thông tin mượn trả

DATA BORROW RETURN					
Order	Code Student	Code Book	Date Borrow	Date Return	Status
000001	102180233	TH-001	10/10/2019	01/01/1900	Dang muon
000002	107190118	KN-003	08/12/2019	01/01/1900	Dang muon
000003	103170116	KN-002	05/10/2019	03/11/2019	Da tra
000004	103190206	LS-001	11/11/2019	07/12/2019	Da tra

Hình 16. Hiển thị thông tin mượn trả

3.2.7. Hiển thị thông tin loại sách

DATA TYPEBOOK		
Order	Code Type	Name Type
1	CT	Chính Trị
2	KH	Khoa Hoc
3	KN	Ky Nang
4	LS	Lich Su
5	TH	Tin Hoc
6	TN	Truyen Ngan
7	TT	Tieu Thuyet

Hình 17. Hiển thị thông tin loại sách

3.2.8. Sắp xếp thông tin sinh viên theo họ tên

SORT

1. Sort Student
2. Sort Book
3. Sort Borrow Return
4. Sort Type Book
5. Return

SORT STUDENT BY

1. Sort By Code Student
2. Sort By Name Student
3. Sort By Class
4. Sort By Faculty
5. Sort By Phone Number
6. Return

Enter your attribute's student to sort:

DATA STUDENT

Order	Code Student	Name Student	Faculty	Class	Phone Number
1	107180300	Dinh Cong Tru	Hoa	18H1	0147258369
2	105190206	Do Nhu Van	Dien	19D3	0125206307
3	107190118	Doan Cong Tam	Hoa	18H4	0369852147
4	102180137	Huynh Van Quan	Cong nghe thong tin	18TCLC_DT1	0367140514
5	102180233	Nguyen Van Du	Cong nghe thong tin	18TCLC_DT1	0905051213
6	102180156	Nguyen Van Nam	Cong nghe thong tin	18TCLC_DT2	0345222333
7	103190206	Phan Nhu Tam	Co khi giao thong	19CKGT1	0159862347
8	101170303	Tran Thanh Tam	Co khi	17CDT2	0122999456
9	103170116	Tran Thu Ha	Co khi giao thong	17CKGT4	0357951258

Hình 18. Sắp xếp thông tin sinh viên theo họ tên

3.2.9. Sắp xếp thông tin mượn trả theo ngày mượn

SORT

1. Sort Student
2. Sort Book
3. Sort Borrow Return
4. Sort Type Book
5. Return

SORT BORROW RETURN INFO BY

1. Sort By Code Student
2. Sort By Code Book
3. Sort By Date Borrow
4. Sort By Date Return
5. Sort By Status
6. Return

Enter your attribute to sort:

DATA BORROW RETURN

Order	Code Student	Code Book	Date Borrow	Date Return	Status
000003	103170116	KN-002	05/10/2019	03/11/2019	Da tra
000001	102180233	TH-001	10/10/2019	01/01/1900	Dang muon
000004	103190206	LS-001	11/11/2019	07/12/2019	Da tra
000002	107190118	KN-003	08/12/2019	01/01/1900	Dang muon

Hình 19. Sắp xếp thông tin mượn trả theo ngày mượn

3.2.10. Thêm sinh viên

Invite you choose task and press enter: 1
 Code student: 102180136
 Name student: Trinh Xuan Phuc
 Faculty: Cong nghe thong tin
 Class: 18TCLC_DT1
 Phone number: 0159753258
 Insert successfully

DATA STUDENT

Order	Code Student	Name Student	Faculty	Class	Phone Number
1	101170303	Tran Thanh Tam	Co khi	17CDT2	0122999456
2	102180137	Huynh Van Quan	Cong nghe thong tin	18TCLC_DT1	0367140514
3	102180156	Nguyen Van Nam	Cong nghe thong tin	18TCLC_DT2	0345222333
4	102180233	Nguyen Van Du	Cong nghe thong tin	18TCLC_DT1	0905051213
5	103170116	Tran Thu Ha	Co khi giao thong	17CKGT4	0357951258
6	103190206	Phan Nhu Tam	Co khi giao thong	19CKGT1	0159862347
7	105190206	Do Nhu Van	Dien	19D3	0125206307
8	107180300	Dinh Cong Tru	Hoa	18H1	0147258369
9	107190118	Doan Cong Tam	Hoa	18H4	0369852147
10	102180136	Trinh Xuan Phuc	Cong nghe thong tin	18TCLC_DT1	0159753258

Hình 20. Giao diện thêm sinh viên

3.3. Đánh giá kết quả

3.3.1. Đối tượng tiếp cận

Với sự phát triển của công nghệ 4.0, việc sử dụng các phương pháp quản lý thư viện truyền thống không đáp ứng được hết các nhu cầu hiện tại. Do đó, những phương pháp quản lý thư viện hiện đại ứng dụng công nghệ sẽ ngày càng có nhiều thư viện tiếp cận và sử dụng. Đối tượng sử dụng chủ yếu chính là các thư viện của các trường đại học chưa được công nghệ hóa hay mở rộng ra là thư viện của các trường tiểu học, trung học, các mô hình thư viện tư nhân,...

3.3.2. Tính thuận tiện

Đối với người mượn: Được thực hiện các thủ tục đăng ký, mượn trả sách một cách nhanh chóng. Thông tin được lưu trữ chính xác, lâu dài, mang tính bảo mật cao.

Đối với thư viện: Cải thiện được hệ thống quản lý thư viện, có thể quản lý và kiểm soát thông tin nhanh chóng. Nhờ vào ứng dụng, giúp giảm đáng kể số lượng thủ thư quản lý thư viện, giúp thủ thư thực hiện các thao tác quản lý thư viện một cách nhanh chóng, chính xác.

3.3.3. Khả năng trở nên phổ biến

Với thời đại công nghệ 4.0 thì việc ứng dụng các sản phẩm công nghệ vào thực tiễn ngày càng phổ biến. Trong đó, ứng dụng quản lý thư viện của chúng em sẽ đáp ứng được nhu cầu phần lớn các thư viện, giải quyết được nhiều vấn đề mà các thư viện đặt ra. Ngoài ra, với hệ thống quản lý tối ưu thì trong tương lai, ứng dụng

sẽ không chỉ phát triển ở một vài thư viện mà còn có thể quảng bá và giới thiệu rộng rãi đến các hệ thống thư viện chưa áp dụng công nghệ hoặc đã áp dụng công nghệ nhưng chưa đáp ứng được các nhu cầu thực tế.

3.3.4. Vấn đề tiếp cận với người dùng

Vì ứng dụng còn mới nên vấn đề được các thư viện biết đến và tiếp cận thì còn khó khăn. Vì vậy, chúng em cần đến sự hỗ trợ của nhà trường và các cơ quan giáo dục của thành phố. Với nhiều tính năng đáp ứng được các nhu cầu thực tế trong quản lý thư viện, nhóm chúng em tin rằng ứng dụng sẽ giúp tối ưu các hệ thống thư viện, làm hài lòng độc giả, thủ thư,... và được nhân rộng trong các mô hình thư viện thành phố và xa hơn là cả nước.

CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

4.1. Kết luận:

Nền công nghiệp 4.0 đang diễn ra một cách mạnh mẽ. Công nghệ thông tin đang phát triển ngày càng mạnh. Vì vậy, việc ứng dụng công nghệ thông tin vào việc tối ưu hóa hệ thống quản lý thư viện là một điều hết sức cần thiết. Ứng dụng sẽ giúp cho viện quản lý thư viện diễn ra một cách thuận lợi, chính xác. Bên cạnh đó, nó cũng giúp độc giả cảm thấy hài lòng với chất lượng dịch vụ của thư viện.

Bên cạnh đó, thông qua đồ án lần này, chúng em không chỉ vận dụng, khái quát được các kiến thức đã học mà còn kết hợp kiến thức đa môn, giúp giải quyết được nhiều vấn đề trong đề tài đặt ra và hướng đến là có một chương trình tối ưu, tốt nhất.

4.2. Hạn chế:

Do thời gian thực hiện đề tài tương đối hạn chế và chúng em cũng không thể tránh được những thiếu sót nhất định như chương trình có một số chức năng chưa hoàn thiện, giao diện cũng chưa thực sự được làm tốt nhất,... Trên đây là những mặt tồn tại lớn khi chúng em làm đồ án này. Và chắc chắn rằng sau này khi có đầy đủ kiến thức chuyên môn, chúng em nhất định sẽ tiến hành tối ưu và ngày một hoàn thiện sản phẩm hơn.

4.3. Hướng phát triển:

Để chương trình hoạt động tối ưu hơn, chúng em cần phải sử dụng một cấu trúc dữ liệu tốt hơn, thông tin cần được bảo mật cao hơn. Đồng thời, giải thuật trong cần phải được tối ưu để thao tác nhanh hơn. Bên cạnh đó, chúng em còn có ý tưởng phát triển thêm nhiều chức năng trong tương lai như đăng ký, thực hiện chính xác hơn các chức năng, xử lý được các trường hợp ngoại lệ và phải triển giao diện thật thân thiện, hoàn chỉnh. Cuối cùng, chúng em luôn tìm kiếm sự tối ưu nhất có thể và hướng đến một chương trình thông minh có thể tương tác bằng giọng nói, hoạt động tự động.

TÀI LIỆU THAM KHẢO

1. PGS. TS. Nguyễn Thanh Bình, Phân tích và thiết kế giải thuật, Nhà xuất bản giáo dục Việt Nam.
2. TS. Đặng Hoài Phương, Slide bài giảng môn lập trình hướng đối tượng.
3. TS. Trương Ngọc Châu – Nguyễn Đức Thuận, Phương pháp giải bài tập cơ sở dữ liệu quan hệ, Nhà xuất bản khoa học và kỹ thuật.
4. GS. Phạm Văn Ất, Giáo trình C++ & Lập trình hướng đối tượng, Nhà xuất bản Bách khoa Hà Nội.
4. Đại học FPT, Nhập môn cơ sở dữ liệu, Nhà xuất bản Bách khoa Hà Nội.
5. Website:
 - www.qhonline.info.
 - www.facebook.com.
 - www.sinhvienit.net.
 - www.vn-zoom.com.
 - www.slideshare.net.

