TRƯỜNG ĐẠI HỌC BÁCH KHOA

KHOA CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN

CƠ SỞ NGÀNH MẠNG

HỆ ĐIỀU HÀNH:

**THÔNG QUA SOCKET, MULTHREAD GIẢI QUYẾT VẤN ĐỀ ĐỒNG BỘ HOÁ.**

LẬP TRÌNH MẠNG

**VIẾT CHƯƠNG TRÌNH ĐẢM BẢO CHỐNG TRÙNG VÉ TÀU HỎA KHI PHÂN TÁN CHỨC NĂNG BÁN VÉ CHO CÁC GA TRONG HỆ THỐNG ĐƯỜNG SẮT.**

Giáo viên hướng dẫn : Ts. LÊ THỊ MỸ HẠNH

Sinh viên thực hiện : LÊ CÔNG TRỊNH

Lớp : 18TB2

Mã sinh viên : 102180283

Nhóm : 16N12

# MỞ ĐẦU

Đồ án Cơ sở ngành Mạng là học phần rất quan trọng trong quá trình học tập của sinh viên. Đồ án này giúp sinh viên tiếp cận và rèn luyện các kiến thức, kỹ năng liên quan đến Môn học Nguyên lý hệ điều hành và Lập trình mạng.

Trong khuôn khổ đồ án, dưới sự hướng dẫn của Thầy thS. Mai Văn Hà em đã chọn đề tài “Viết chương trình đảm bảo chống trùng vé tàu hỏa khi phân tán chức năng bán vé cho các ga trong hệ thống đường sắt.” cho cả hai phần nguyên lý hệ điều hành và lập trình mạng. Thông qua đề tài này, em đã hiểu rõ hơn phần nào về nguyên lý hoạt động xử lý “Thông qua Socket, Multhread giải quyết vấn đề đồng bộ hoá”.

Do thời gian và kiến thức của em còn hạn chế nên không tránh khỏi những sai sót nhất định trong quá trình làm đồ án này. Rất mong nhận được sự góp ý của quý thầy cô.

Em xin chân thành cảm ơn!

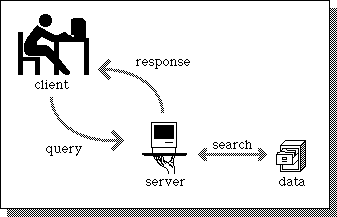
# NGUYÊN LÝ HỆ ĐIỀU HÀNH

TIÊU ĐỀ: THÔNG QUA SOCKET, MULTHREAD GIẢI QUYẾT VẤN ĐỀ ĐỒNG BỘ HOÁ

## CƠ SỞ LÝ THUYẾT

### Mô Hình Client/Server:

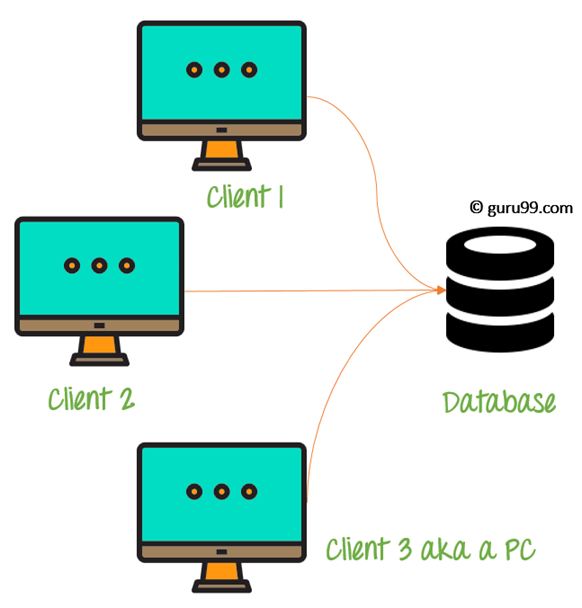
* Thuật ngữ Client/Server xuất hiện vào đầu thập niên 80
* Dạng phổ biết của mô hình ứng dụng phân tán
* Là mô hình mạng cơ bản nhất hiện nay
* Một số ứng dụng Client/Server phổ biến hiện nay:
  + - Email
    - FTP
    - Web
* Mô hình Client/Server cung cấp một cách tiếp cận tổng quát để chia sẽ tài nguyên trong hệ phân tán.
* Cả tiến trình Client/Server đều có thể chạy trên cùng 1 máy tính.
* Một tiến trình Server có thể sử dụng dich vụ của server khác.
* Mô hình truyền tin Client/Server hướng tới việc cung cấp dịch vụ.
* Quá trình trao đổi bao gôm:
  + - Bước 1: Truyền một tiến trình yêu cầu từ Client đến Server
    - Bước 2: Yêu cầu được Server xử lý
    - Bước 3: Truyền đáp ứng cho client
* Mô hình truyền tin này liên quan đến việc truyền hai thông điệp và được đồng bộ hóa giữa Client và Server
  + - Ở bước 1, tiến trình Server phải nhận được thông điệp yêu cầu ngay khi nó đến và phát ra yêu cầu Client phải tạm dừng
    - Ở bước 3, tiến trình client ở trạng thái chờ cho đến khi nó nhận được đáp ứng do server gởi về.
* Mô hình Client/Server được cài đặt dựa trên các thao tác cơ bản là gởi và nhận.



* Chế độ phong tỏa(blocked)
* Khi tiến trình phát ra lệnh gởi dữ liệu, việc thực hiện tiến trình sẽ bị tạm dừng cho đến khi tiến trình nhận phát ra lệnh nhận dữ liệu.
* Tương tự với tiến trình nhận dữ liệu. Nếu tiến trình Client hoặc Server phát lệnh nhận dữ liệu, mà tại đó chưa có lệnh dữ liệu được gởi tới thì thực thi tiến trình cũng bị dừng lại cho đến khi dữ liệu được gởi đến
* Chế độ không phong tả (non-blocked)
* Trong chế độ này, khi client hay server phát lệnh gởi dữ liêu thực sự, việc thực thi của tiến trình vẫn được tiến hành mà không quan tâm đến việc tiến trình nào phát ra lệnh nhận dữ liệu đó hay không.
* + Tương tự cho trường họp nhận dữ liệu, khi tiến trình phát ra lệnh nhận dữ liệu, nó sẽ nhận dữ liệu hiện có, việc thực thi của tiến trình vẫn được tiến hành mà không quan tâm đến tiến trình nào phát lệnh gởi dữ liệu hay không

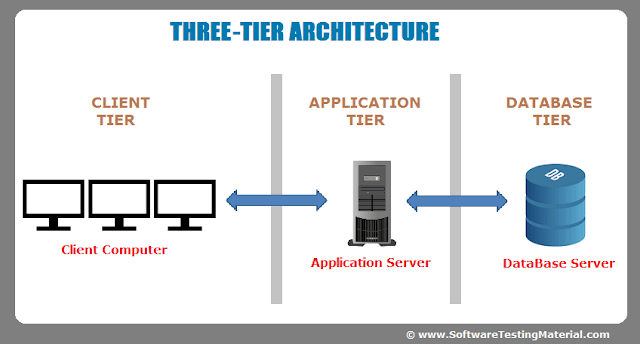
#### Các kiến trúc Client/Server

##### Client/Server hai tầng



* Trong ứng dụng hai tầng truyền thống, khối lượng xử lý dành cho phía Client
* Khi đo server chỉ đóng vai trò như kiểm soát luồng vào ra giữa các ứng dụng và dữ liệu.
* Hiệu năng của ứng dụng bị giảm do tài nguyên hạn chế của PC và khối lượng dữ liệu truyền đi trên mạng cũng tăng theo

##### Client/Server ba tầng



* Ta có thể tránh được các vấn đề của kiến trúc Client/Server hai tầng bằng cách mở rộng kiến trúc thành 3 tầng.
* Một kiến trúc 3 tầng có thêm một tầng mới tách biệt việc xử lý dữ liệu ở tầng trung tâm
* Theo kiến trúc 3 tầng, một ứng dụng chia ra làm 3 tầng riêng biệt nhau
* Tầng đầu tiên: là tầng trình diễn thường bao gồm các giao diện đồ họa. Tầng trình diễn thông thường nhận dữ liệu và định dạng nó để hiển thị.
* Tầng thứ hai: là tầng trung gian hay tầng tác nghiệp.
* Tầng thứ ba: chứa dữ liệu cần thiết cho ứng dụng. Tầng thứ 3 cơ bản chương trình thực hiện các lời gọi hàm để tìm kiếm dữ liệu cần thiết.

##### Client/Server n-tầng:

* Kiến trúc n-tầng được chia thành như sau:
* Tầng giao diện người dùng: quản lý tương tác người dùng và ứng dụng
* Tầng logic trình diễn: xác định cách thức hiển thị người dùng và yêu cầu của người dùng được quản lý như thế nào.
* Tầng logic tác nghiệp: Mô hình hóa các quy tắt tác nghiệp.
* Tầng dịch vụ hạ tầng: Cung cấp chức năng bổ trợ cho ứng dụng.

### Socket:

#### Socket là gì?

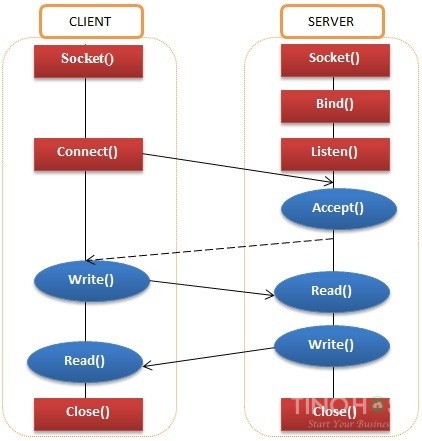
* Socket là một điểm cuối (end-point) của liên kết truyền thông hai chiều (two-way communication) giữa hai chương trình chạy trên mạng. Các lớp Socket được sử dụng để biểu diễn kết nối giữa client và server, được ràng buộc với một cổng port (thể hiện là một con số cụ thể) để các tầng TCP (TCP Layer) có thể định danh ứng dụng mà dữ liệu sẽ được gửi tới.
* Lập trình socket là lập trình cho phép người dùng kết nối các máy tính truyền tải và nhận dữ liệu từ máy tính thông qua mạng
* Hiểu đơn giản, socket là thiết bị truyền thông hai chiều gửi và nhận dữ liệu từ máy khác.
* Cách thức hoạt động
* Là một giao diện lập trình ứng dụng mạng, socket giúp các bạn lập trình kết nối các ứng dụng để truyền và nhận giữ liệu trong môi trường có kết nối Internet bằng cách sử dụng phương thức TCPIP và UDP.
* Khi cần trao đổi dữ liệu cho nhau thì 2 ứng dụng cần phải biết thông tin tối thiểu là IP và sô hiểu cổng của ứng dụng kia.
* 2 ứng dụng có thể nằm cùng trên một máy
* 2 ứng dụng cùng nằm trên một máy không được cùng số hiệu cổng

#### Phân loại Socket

##### + Stream Socket

Dựa trên giao thức TCP( Tranmission Control Protocol), việc truyền dữ liệu chỉ thực hiện giữa 2 quá trình đã thiết lập kết nối. Do đó, hình thức này được gọi là socket hướng kết nối.

**+ Lập trình Socket với TCP**



**Ưu điểm:** Có thể dùng để liên lạc theo mô hình client và sever. Nếu là mô hình client /sever thì sever lắng nghe và chấp nhận từ client. Giao thức này đảm bảo dữ liệu được truyền đến nơi nhận một cách đáng tin cậy, đúng thứ tự nhờ vào cơ chế quản lý luồng lưu thông trên mạng và cơ chế chống tắc nghẽn. Đồng thời, mỗi thông điệp gửi phải có xác nhận trả về và các gói tin chuyển đi tuần tự.

**Hạn chế:** Có một đường kết nối (địa chỉ IP) giữa 2 tiến trình nên 1 trong 2 tiến trình kia phải đợi tiến trình kia yêu cầu kết nối.

##### + Datagram Socket

Dựa trên giao thức UDP( User Datagram Protocol) việc truyền dữ liệu không yêu cầu có sự thiết lập kết nối giữa 2 quá trình. Do đó, hình thức này được gọi là socket không hướng kết nối.

**+ Lập trình Socket với UDP**

**Ưu điểm:** Do không yêu cầu thiết lập kết nối, không phải có những cơ chế phức tạp nên tốc độ giao thức khá nhanh, thuận tiện cho các ứng dụng truyền dữ liệu nhanh như chat, game…..

**Hạn chế:** Ngược lại với giao thức TCP thì dữ liệu được truyền theo giao thức UDP không được tin cậy, có thế không đúng trình tự và lặp lại.

### Đa luồng:

#### Khái niệm về đa luồng trong java

Đa luồng (multithreading) trong java là một tiến trình thực hiện nhiều luồng đồng thời.

* Luồng (thread) về cơ bản là một tiến trình con (sub-process). Nó là đơn vị nhỏ nhất của tiến trình. Đa tiến trình (multiprocessing) và đa luồng (multithreading) cả hai được sử dụng để tạo ra hệ thống đa nhiệm (multitasking).
* Nhưng chúng ta sử dụng đa luồng nhiều hơn đa tiến trình bởi vì các luồng chia sẻ một vùng bộ nhớ chung. Chúng không phân bổ vùng bộ nhớ riêng biệt để tiết kiệm bộ nhớ, và chuyển đổi ngữ cảnh giữa các luồng mất ít thời gian hơn tiến trình.
* Đa luồng trong java được sử dụng hầu hết trong các game, hoạt hình,...

#### Ưu điểm của đa luồng trong java

* Nó không chặn người sử dụng vì các luồng là độc lập và bạn có thể thực hiện nhiều công việc cùng một lúc.
* Bạn có thể thực hiện nhiều hoạt động với nhau để tiết kiệm thời gian.
* Luồng là độc lập vì vậy nó không ảnh hưởng đến luồng khác nếu ngoại lệ xảy ra trong một luồng duy nhất.

**Trong Client/Server**

* Khi khối lượng công việc mà server cần xử lý của một yêu cầu client là quá lớn và không biết được thời điểm hoàn thành công việc xử lý thì các server này không thể chấp nhận được.
* Để khắc phục điều này, ta quản lý mỗi phiên của client bằng một tuyến đoạn riêng, cho phép client làm việc với nhiều client đồng thời.
* Server này được gọi là server tương tranh và tạo ra một tuyến đoạn để quản lý từng yêu cấu sau đó tiếp tục nghe từ các client khác.
* Các server đơn tuyến đoạn chỉ quản lý được môt liên kết tại một thời điểm.
* Trong thực tết một server có thể phải quản lý nhiều liên kết cùng một lúc
* Để thực hiện điều này server chấp nhận các liên kết và chuyển các liên kết này cho từng tuyến xử lý.

Đồng bộ là gì? Tại sao lại cần đồng bộ?

#### Tại sao cần đồng bộ?

Trong kỹ thuật đa luồng, nếu các luồng sử dụng dữ liệu độc lập thì ta không có gì phải tranh luận. Nhưng nếu trên hệ thống nhiều CPU hoặc CPU đa nhân hay CPU hỗ trợ siêu phân luồng, các luồng sẽ thục sự hoạt động song song tại cùng 1 thời điểm. Như vậy, nếu các luồng này cùng truy xuất đến 1 biến dữ liệu hoặc 1 phương thức nhờ vào lý do đã nói ở phần trên, điều này có thể gây ra việc sai lệch dữ liệu.

public class Counter {

int count=0;

public void tang() {

count=count+1;

}

}

Giả sử rằng, tại cùng 1 thời điểm có 2 luồng cùng lúc gọi phương thức tang() trên 1 đối tượng thuộc lớp Counter.

Như vậy, cùng 1 lúc, 2 luồng cùng lấy ra được giá trị count hiện tại là 0, và cùng lúc cộng thêm 1 vào giá trị count này thành 1, sau đó cùng ghi giá trị mới cộng lại được là 1 lên RAM.

Nếu thực sự như vậy, sau khi cả 2 luồng thực hiện công việc thì count có giá trị 1. Tuy nhiên, 2 luồng cùng tăng giá trị count thì count phải có giá trị 2 mới đúng là việc chúng ta mong muốn

=> Việc sắp xếp thứ tự truy xuất đối tượng lúc này là thật sự cần thiết khi các luồng có dùng chung dữ liệu.

#### Đồng bộ hóa là gì?

Như đã nói ở trên, việc sắp xếp thứ tự các luồng truy xuất đối tượng thật sự cần thiết trong kỹ thuật đa luồng. Đồng bộ hóa (synchronized) chính là việc xắp xếp thứ tự các luồng khi truy xuất vào cùng đối tượng sao cho không có sự xung đột dữ liệu. Nói cách khác, đồng bộ hóa tức là thứ tự hóa.

## PHÂN TÍCH THIẾT KẾ HỆ THỐNG

### Phân tích đề tài:

Viết chương trình đảm bảo chống trùng vé tàu hỏa khi phân tán chức năng bán vé cho các ga trong hệ thống đường sắt.

### Phân tích thiết kế hệ thống

- Ứng dụng được xây dựng dựa trên mô hình Client / Server sử dụng lập trình Socket . Ứng dụng được xây dựng gồm 2 phần : phía Client (cho khách sử dụng ) và phía server

- Phía server sẽ kết nối vào thao tác để lấy thông tin từ cơ sở dữ liệu rồi gửi xuống phía client.

- Đảm bảo chống trùng đối với chứng năng đặt vé

- Đồng bộ chức năng đặt vé để không có 2 vé cũng đặt một lúc

## TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ

1.Client/Server sử dụng java FX

## KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### Những kết quả đạt được

* Hiểu được đa luồng trong lập trình java
* Hiểu được cơ chế hoạt động của các tiến trình, cách tạo ra tiến trình con để thực hiện một phần công việc của tiến trình cha.
* Hiểu được giao thức TCP, socket và ứng dụng vào lập trình
* Áp dụng cơ chế đồng bộ các tiến trình

### Hướng phát triển:

* Phát triển thêm các chức năng kiểm tra vé, in vé, ……
* Phát triển ứng dụng trên nền tải web để khách hàng có thể truy cập

# LẬP TRÌNH MẠNG

**TIÊU ĐỀ: TÌM HIỂU SỬ DỤNG KỸ THUẬT LẬP TRÌNH SOCKET XÂY DỰNG CHƯƠNG TRÌNH TƯ VẤN CHĂM SÓC SẮC ĐẸP**

1 CHƯƠNG 1. CƠ SỞ LÝ THUYẾT

1.1 1.1.1. Ngôn ngữ sử dụng: Java

Java là một ngôn ngữ lập lập trình, được phát triển bởi Sun Microsystem vào năm 1995, là ngôn ngữ kế thừa trực tiếp từ C/C++ và là một ngôn ngữ lập trình hướng đối tượng.

1.2 1.1.2. Cơ sở dữ liệu : MySQL

MySQL là một hệ thống quản trị cơ sở dữ liệu mã nguồn mở (gọi tắt là RDBMS) hoạt động theo mô hình client-server. Với RDBMS là viết tắt của Relational Database Management System. MySQL được tích hợp apache, PHP. MySQL quản lý dữ liệu thông qua các cơ sở dữ liệu. Mỗi cơ sở dữ liệu có thể có nhiều bảng quan hệ chứa dữ liệu. MySQL cũng có cùng một cách truy xuất và mã lệnh tương tự với ngôn ngữ SQL. MySQL được phát hành từ thập niên 90s.

1.3 1.1.3. Thiết kế giao diện: HTML5 + CSS3 + BOOTSTRAP4

HTML (tiếng Anh, viết tắt cho HyperText Markup Language, hay là "Ngôn ngữ Đánh dấu Siêu văn bản") là một ngôn ngữ đánh dấu được thiết kế ra để tạo nên các trang web với các mẩu thông tin được trình bày trên World Wide Web.

CSS là chữ viết tắt của Cascading Style Sheets, nó là một ngôn ngữ được sử dụng để tìm và định dạng lại các phần tử được tạo ra bởi các ngôn ngữ đánh dấu (ví dụ như HTML). Bạn có thể hiểu đơn giản rằng, nếu HTML đóng vai trò định dạng các phần tử trên website như việc tạo ra các đoạn văn bản, các tiêu đề, bảng,…thì CSS sẽ giúp chúng ta có thể thêm một chút “phong cách” vào các phần tử HTML đó như đổi màu sắc trang, đổi màu chữ, thay đổi cấu trúc,…rất nhiều.

Bootstrap là 1 framework HTML, CSS, và JavaScript cho phép người dùng dễ dàng thiết kế website theo 1 chuẩn nhất định, tạo các website thân thiện với các thiết bị cầm tay như mobile, ipad, tablet,...

1.4 1.1.4. Framework: Java spring

Spring là một Framework phát triển các ứng dụng Java được sử dụng bởi hàng triệu lập trình viên. Nó giúp tạo các ứng dụng có hiệu năng cao, dễ kiểm thử, sử dụng lại code…

Spring Framework được xây dựng dựa trên 2 nguyên tắc design chính là: Dependency Injection và Aspect Oriented Programming.

Những tính năng core (cốt lõi) của Spring có thể được sử dụng để phát triển Java Desktop, ứng dụng mobile, Java Web. Mục tiêu chính của Spring là giúp phát triển các ứng dụng J2EE một cách dễ dàng hơn dựa trên mô hình sử dụng POJO (Plain Old Java Object)

Hình 1.1. Kiến trúc tổng thể của spring frame work

1.5 1.1.5. Mô hình MVC (Model-View-Controller)

MVC là viết tắt của Model – View – Controller. Là một kiến trúc phần mềm hay mô hình thiết kế được sử dụng trong kỹ thuật phần mềm. Nói cho dễ hiểu, nó là mô hình phân bố source code thành 3 phần, mỗi thành phần có một nhiệm vụ riêng biệt và độc lập với các thành phần khác.

Hình 1.2. Mô hình MVC

1.6 1.1.6. Công nghệ JPA để tương tác với CSDL

JPA (Java Persistence API) là 1 giao diện lập trình ứng dụng Java, nó mô tả cách quản lý các mối quan hệ dữ liệu trong ứng dụng sử dụng Java Platform.

JPA cung cấp một mô hình POJO persistence cho phép ánh xạ các table/các mối quan hệ giữa các table trong database sang các class/mối quan hệ giữa các object.

1.2. Mô tả HỆ THỐNG

1.7 1.2.1. Khảo sát hiện trạng

Sau khi khảo sát một số website bán sách hiện nay, nhóm em đã nắm bắt được các thông tin sau:

Quản lý khách hàng: Mỗi khách hàng được quản lý các thông tin sau đây: Họ tên, địa chỉ, điện thoại. Ngoài ra, đối với khách hàng có nhu cầu thanh toán online, có thể quản lý thêm số tài khoản của khách.

Quản lý seller: Mỗi seller hay cộng tác viên được quản lý các thông tin sau đây: Xử lý các đơn hàng mà seller bán, xem tất cả các mặt hàng có trên hệ thống.

Quản lý sách: Mỗi loại sách được quản lý những thông tin: Tên sách, đơn giá, trạng thái, hình ảnh, mô tả.

Quá trình đặt hàng của khách hàng: khách hàng xem và lựa chọn mặt hàng cần mua. Nếu lựa chọn thành công, mặt hàng được đưa vào giỏ hàng. Khách hàng tiếp túc lựa chọn thanh toán thì thành lập đơn hàng.

Quản lý đơn hàng: Mỗi đơn hàng được quản lý lý những thông tin : Thông tin người mua, ngày giờ, số lượng, thành tiền, tình trạng,

Từ những thông tin ở trên, hệ thống của nhóm em được xây dựng để phục vụ cho 3 đối người dùng: Admin, khách hàng và seller.

Khách hàng là những người có nhu cầu mua sách. Khác với việc đặt hàng trực tiếp tại các nhà sách, khách hàng phải hoàn toàn tự tìm kiếm theo tên mặt hàng, khu vực để có thể mua được hàng. Trên web, các mặt hàng sẽ được sắp xếp và phân theo từng loại mặt hàng giúp cho khách hàng dễ dàng tìm kiếm.

Trong hoạt động này, khách hàng chỉ cần chọn một mặt hàng nào đó từ danh mục các mặt hàng thì những thông tin về mặt hàng đó sẽ hiển thị lên màn hình như: hình ảnh, đơn giá, mô tả…và bên cạnh là những liên kết để thêm hàng hóa vào giỏ hàng. Đây là giỏ hàng điện tử mà trong đó chứa các thông tin về hàng hóa lẫn số lượng khách mua và hoàn toàn được cập nhật trong giỏ.

Khi khách hàng muốn đặt hàng, hệ thống sẽ lấy thông tin từ giỏ hàng và thiết lập đơn hàng.

Seller sẽ là người có nhu cầu bán sách thông qua website để ăn chênh lệch giá và phần trăm chiết khấu của hệ thống. Khi seller chốt được sách thì sẽ đặt trên hệ thống để dễ dàng quản lý.

Admin ( người quản trị ) chính là người quản lý và điều hành toàn bộ hoạt động của hệ thống. Admin sẽ có thể quản lý toàn bộ thông tin trong hệ thống (khách hàng, seller, đơn hàng,…

1.2.2. Chức năng

Website phục vụ cho mục đích giới thiệu và bán các sản phẩm của cửa hàng, có thể giúp người mua biết được chi tiết sản phẩm với giá cả chính xác nhất. Website của nhóm sẽ gồm những chức năng chính sau đây

- Cho phép cập nhật thêm sách vào CSDL hoặc xóa bớt sách

- Cập nhật thông tin sách như tác giả, giá bán, thể loại, mô tả, …

- Hiển thị danh sách mặt hàng theo các danh mục (thể loại, lớp, môn học,…)

- Tìm kiếm sách.

- Quản lý đơn đặt hàng.

- Xử lý đơn hàng.

- Quản lý , hiển thị thông tin khách hàng

- Quản lý, hiển thị thông tin seller

- Quản lý giỏ hàng .

- Thống kê số liệu (đơn hàng, khách hàng, sản phẩm được mua, …)

-

CHƯƠNG 3: TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ

1.Client/Server sử dụng Spring

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. Những kết quả đạt được

Áp dụng được

Hiểu được tiến trình trong lập trình mang.

2. Hướng phát triển

- Deploy ứng dụng web lên server thật : các chức năng của website vận hành tốt khi triển khai trên server thật .

• Áp dụng nhiều công nghệ mới trên front-end như react.js , vue.js , angular.js để cải thiện load trang và độ thân thiện khi thao tác giữa các chức năng.

KẾT LUẬN CHUNG

* Sau quá trình thực hiên đồ án Cơ sở ngành mạng, em cảm thấy mình đã có nhiều kiến thức về về cả hệ điều hành như tiến trình, giao tiếp giữa các tiến trình cũng như trong lĩnh vực mạng như mô hình client-server, socket.
* Ngoài ra, em đã biết cách tổ chức sắp xếp thời gian học tập hợp lý.
* Cũng qua đồ án này, em nhận thấy mình còn nhiều thiếu sót trong việc bổ sung kiến thức chuyên môn.
* Qua đây, cũng là bài học cho bản thân em về việc nâng cao khả năng tự học, không ngừng cố gắng để củng cố kiến thức.