
Group 07

**E-commerce CSGo
Software Architecture Document**

Version 1.0

Revision History

Date	Version	Description	Author
30/11/2022	1.0	Bản sơ khởi cho tài liệu Software Architecture Development (mục 5, mục 6 để trống cho PA4	PDTDuc, PNKMinh, NHPThinh, PTAVu, VMHieu
14/12/2022	1.1	Cập nhật mục 5,6 theo yêu cầu của PA4	NHPThinh, VMHieu, PDTDuc

Table of Contents

1.	Introduction	4
1.1	Purpose	4
1.2	Scope	4
1.3	References	4
2.	Architectural Goals and Constraints	4
3.	Use-Case Model	5
4.	Logical View	6
4.1	Component: Controller	7
4.2	Component: View	9
4.3	Component: Model	10
4.4	Component: Database	11
5.	Deployment	12
6.	Implementation View	13

Software Architecture Document

1. Introduction

1.1 Purpose

Tài liệu này định nghĩa kiến trúc phần mềm về các ràng buộc, thiết kế logic và cài đặt của trang web giao dịch vật phẩm ảo E-commerce CSGO. Nó bao gồm các quy định kiến trúc quan trọng, các mô hình và các sơ đồ UML cho từng thành phần của hệ thống giúp cho các thành viên có cái nhìn bao quát về hệ thống và phát triển trang web một cách hiệu quả đúng với mục tiêu đề ra.

1.2 Scope

Tài liệu Kiến trúc Phần mềm này cung cấp góc nhìn tổng quan về kiến trúc của trang web giao dịch vật phẩm ảo E-commerce CSGO được phát triển bởi nhóm 07 để tạo ra môi trường an toàn, minh bạch, dễ dàng cho phép người dùng giao dịch, trao đổi vật phẩm trong game CSGO.

1.3 References

[1] Vision document E-commerce CSGO

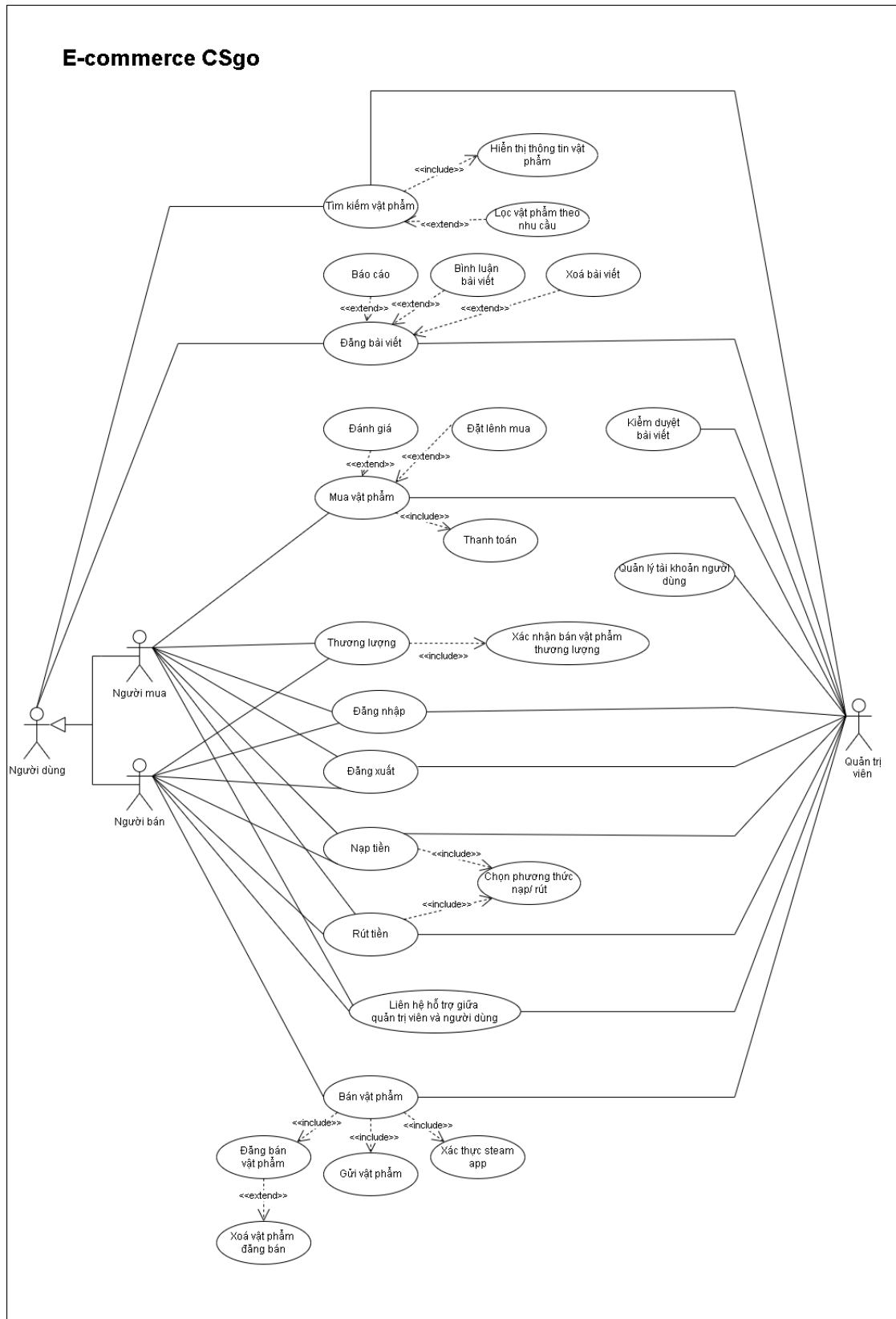
[2] Tài liệu LN07 trên moodle

2. Architectural Goals and Constraints

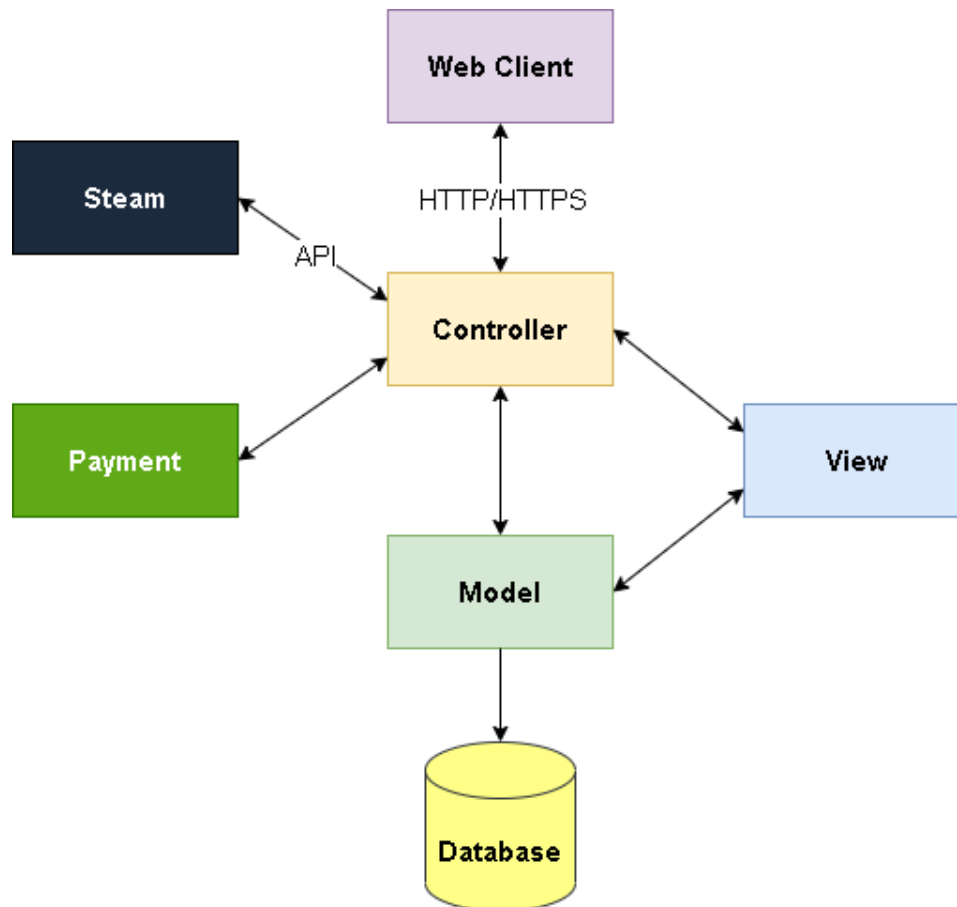
Các mục tiêu và ràng buộc quan trọng của trang web E-commerce CSGO:

- Môi trường thực thi ứng dụng: Web.
- Đảm bảo về mặt tương thích giao diện để có thể hiển thị trên nhiều thiết bị khác nhau như laptop, điện thoại, máy tính bảng.
- Đảm bảo thông tin người dùng phải được mã hóa và xác thực thông qua email hoặc số điện thoại.
- Tất cả các yêu cầu về hiệu suất và trải nghiệm người dùng ở tài liệu Vision document phải được đảm bảo và xem xét trong quá trình phát triển.

3. Use-Case Model



4. Logical View



Trang web E-commerce được xây dựng bao gồm các thành phần chính sau đây:

- View:

- + Mỗi view là một trang web hiển thị dữ liệu. Dữ liệu mà view hiển thị là do Controller lấy từ Model cung cấp
- + Công nghệ sử dụng: Handlebars framework để hỗ trợ binding dữ liệu vào các template để hiển thị ra website
- + Ngôn ngữ lập trình: HTML, CSS, JavaScript

- Controller:

- + Khi Web Client gửi yêu cầu, controller đóng vai trò trung gian thực hiện gọi các hàm trong Model cũng như nạp các file cần thiết trong View để hiển thị lên cho người dùng
- + Công nghệ sử dụng:
 - Express framework của NodeJS để nhận các HTTP/HTTPS request, trả về các trang HTML
 - Handlebars framework để hỗ trợ binding dữ liệu vào các template để hiển thị ra website

- Framework steam-user, steam-trading-manager,... để hỗ trợ việc lấy dữ liệu và gửi nhận vật phẩm trên Steam
- Lấy dữ liệu và thanh toán qua MomoApi hoặc PaypalApi

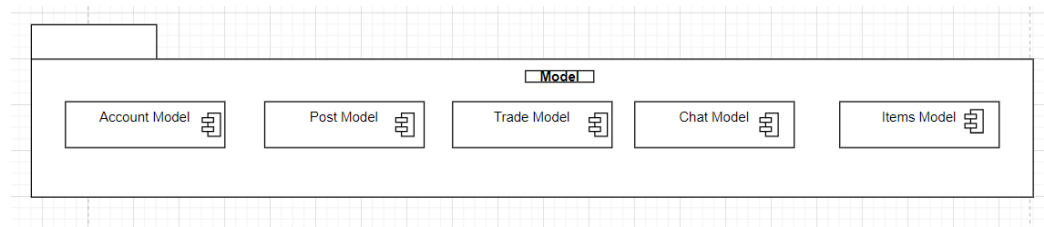
+ Ngôn ngữ lập trình: JavaScript

- Model:

+ Kết nối, cung cấp, ghi ,chỉnh sửa dữ liệu trong cơ sở dữ liệu cho View thông qua Controller

+ Công nghệ sử dụng: Mongoose framework của NodeJS để mô hình hóa đối tượng cho MongoDB giúp dễ dàng tương tác với dữ liệu trong MongoDB

+ Ngôn ngữ lập trình: Javascript



- Database:

+ Lưu trữ toàn bộ dữ liệu của trang web

+ Cơ sở dữ liệu: MongoDB (NoSQL database). Sử dụng MongoDB vì có hiệu suất cao, linh hoạt phù hợp với hệ thống web

- Payment:

+ Trang web thực hiện thanh toán qua cổng thanh toán VNPay, Momo, Paypal

- Steam:

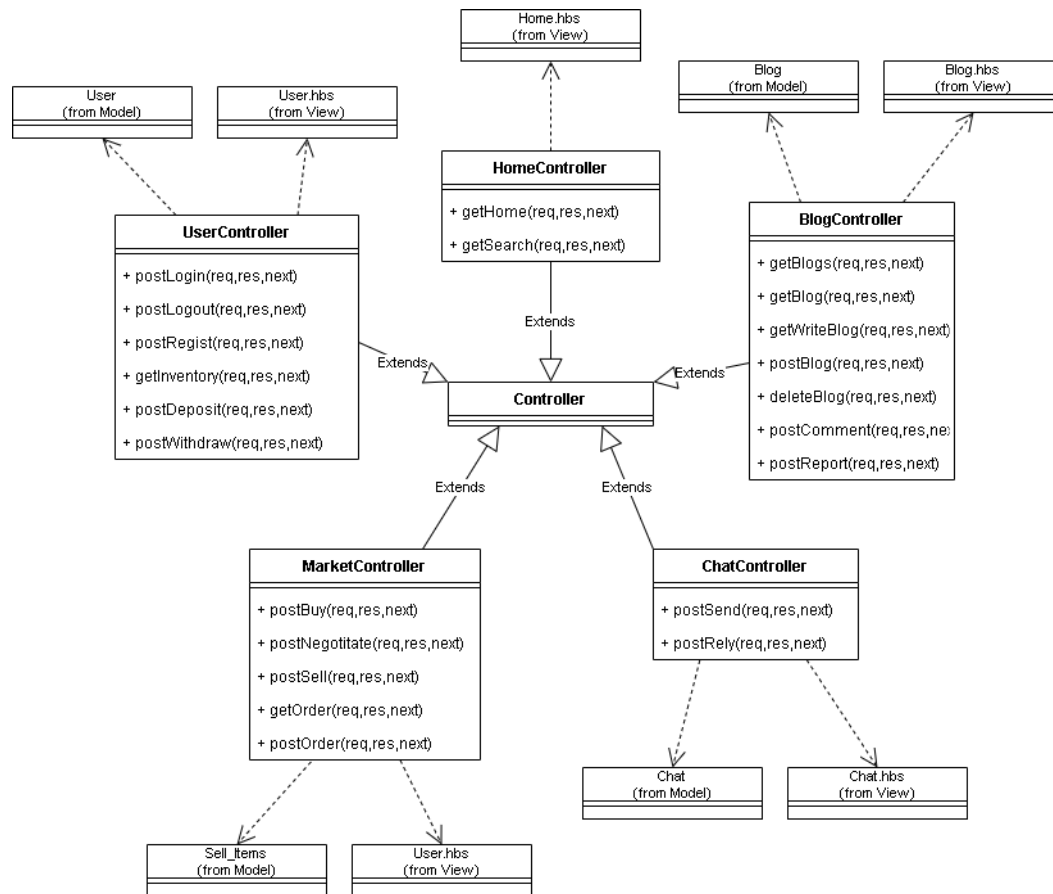
+ Trang chứa thông tin và vật phẩm trong của người dùng, các giao dịch gửi, nhận vật phẩm cũng thực hiện thông qua Steam

4.1 Component: Controller

Mô tả: Thành phần Controller bao gồm cái lớp sau để điều khiển sự tương tác của nó và các thành phần khác với nhau.

- **HomeController:** Xử lý việc lấy dữ liệu và hiển thị cho người dùng ở trang chủ.
- **UserController:** Xử lý các tác vụ theo yêu cầu của người dùng.
- **BlogController:** Quản lý các bài đăng.
- **MarketController:** Xử lý các tác vụ mua, bán hoặc giao dịch vật phẩm.
- **ChatController:** Quản lý tin nhắn.

Class diagram:



Giải thích key class:

- **Home controller:**
 - + getHome(req,res,next): lấy dữ liệu và hiển thị trang chủ
 - + getSearch(req,res,next): xử lý dữ liệu và hiển thị kết quả search vật phẩm
- **User controller:**
 - + postLogin(req,res,next): lấy dữ liệu và xử lý việc đăng nhập
 - + postLogout(req,res,next): lấy dữ liệu và xử lý việc đăng xuất
 - + postRegist(req,res,next): lấy dữ liệu và xử lý việc đăng ký
 - + getInventory(req,res,next): lấy dữ liệu và hiển thị kho đồ người dùng
 - + postDeposit(req,res,next): lấy dữ liệu và xử lý việc nạp tiền
 - + postWithdraw(req,res,next): lấy dữ liệu và xử lý việc rút tiền
- **Blog controller:**
 - + getBlogs(req,res,next): xử lý dữ liệu và hiển thị danh sách cái bài viết

+ getBlog(req,res,next): xử lý và hiển thị 1 bài viết cụ thể

+ getWriteBlog(req,res,next): hiển thị trang viết bài viết

+ postBlog(req,res,next): xử lý đăng bài viết

+ deleteBlog(req,res,next): xử lý xóa bài viết

+ postComment(req,res,next): xử lý đăng bình luận

+ postReport(req,res,next): xử lý gửi báo cáo

- **Market controller:**

+ postBuy(req,res,next): xử lý mua vật phẩm

+ postSell(req,res,next): xử lý bán vật phẩm

+ getOrder(req,res,next): đến trang đặt lệnh mua

+ postOrder(req,res,next): xử lý người dùng đặt lệnh mua

+ postNegotitate(req,res,next): xử lý việc người dùng thương lượng giá

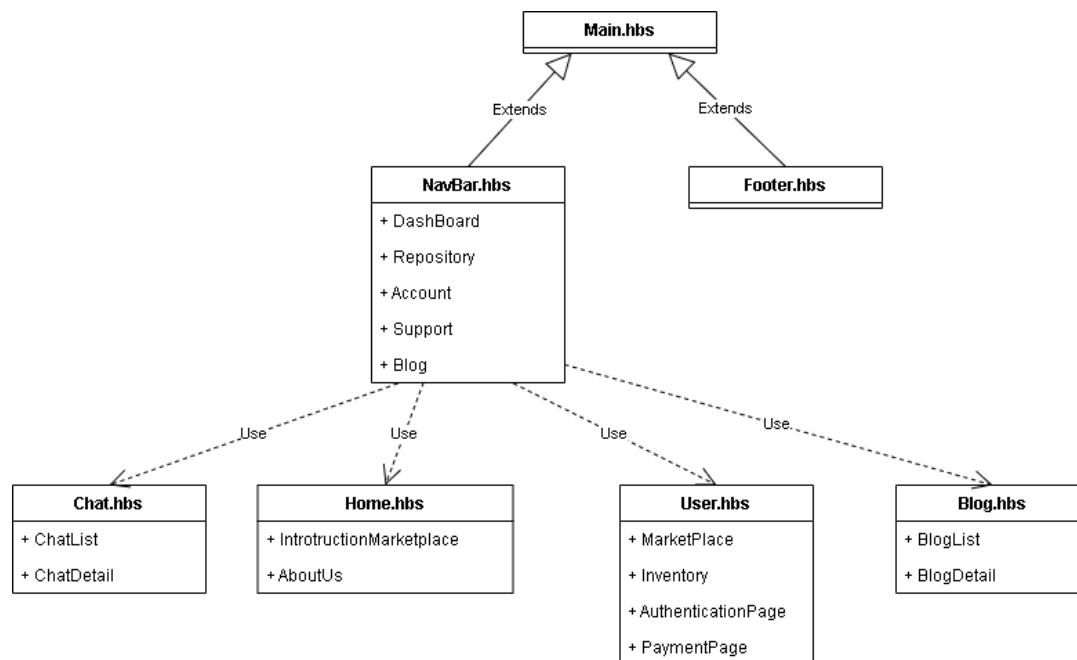
- **Chat controller:**

+ postSend(req,res,next): xử lý người dùng gửi tin nhắn

+ postReply(req,res,next): xử lý việc quản trị viên trả lời tin nhắn

4.2 Component: View

Class diagram:

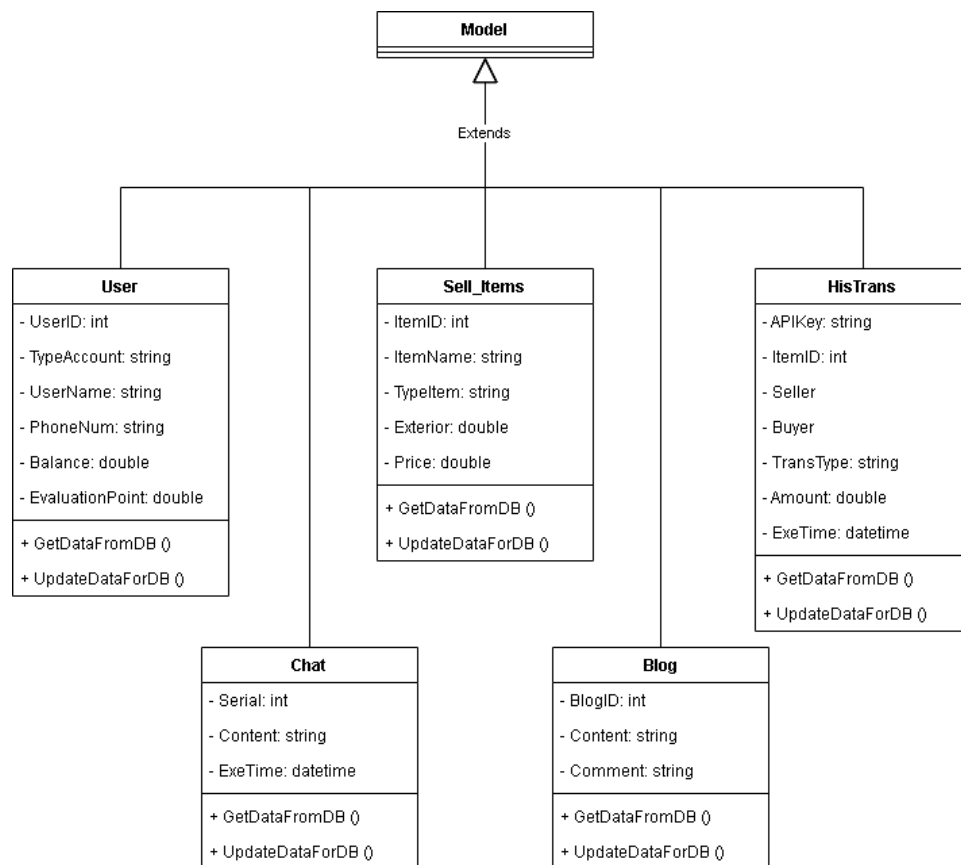


Giải thích key class:

- **Layouts:**
 - + main.hbs: chứa template toàn trang web
- **Partials:**
 - + navbar.hbs: template Thanh navigator của trang web
 - + footer.hbs: template phần Footer của trang web.
- **Body:** template phần body của trang web
 - + home.hbs: template trang chủ
 - + user.hbs: template giao diện người dùng
 - + blog.hbs: template bài viết
 - + chat.hbs: template chat

4.3 Component: Model

Class diagram:

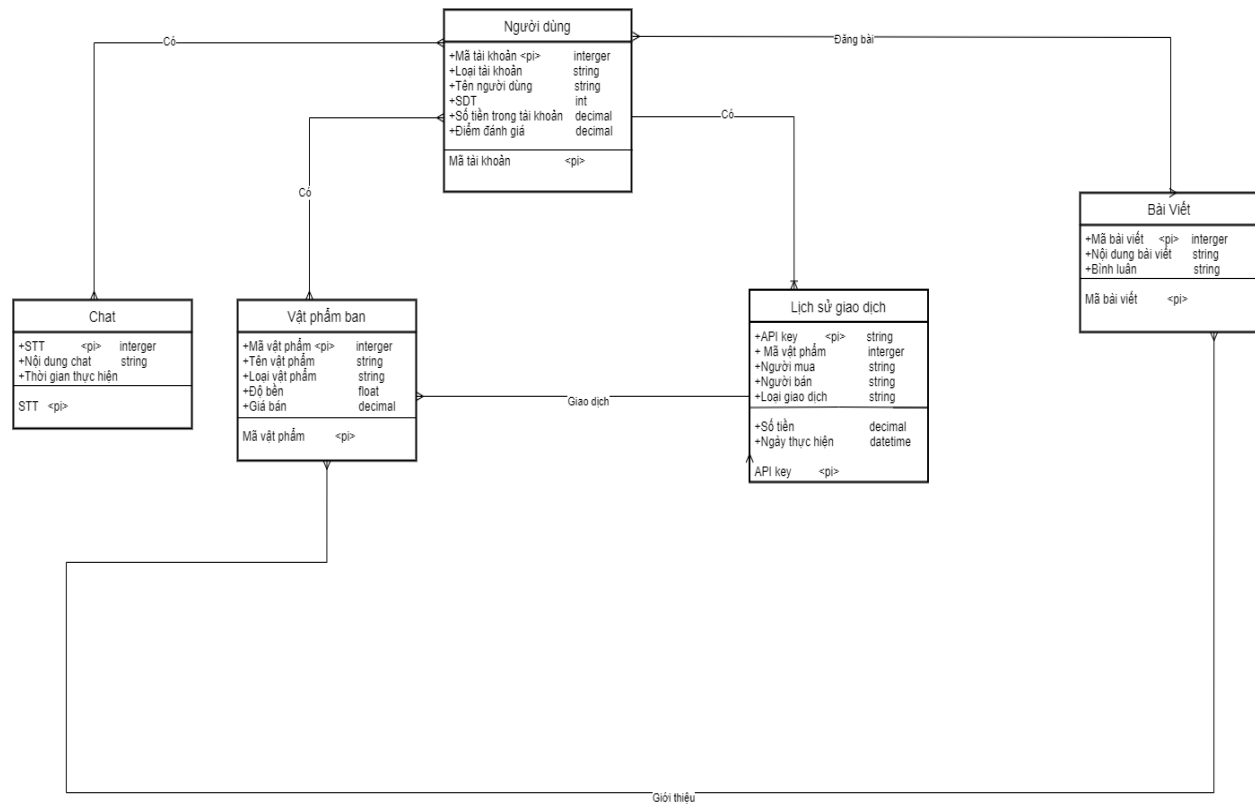


Giải thích key class:

- **User:** Lớp chứa thông tin người dùng đồng thời có chức năng lấy và cập nhật dữ liệu người dùng.
- **Sell_Items:** Lớp chứa thông tin vật phẩm đang được bán và quản lý những vật phẩm đó.
- **HisTrans:** Lớp chứa thông tin lịch sử giao dịch.
- **Chat:** Lớp chứa và quản lý tin nhắn.
- **Blog:** Lớp chứa và quản lý bài đăng.

4.4 Component: Database

Mô hình:



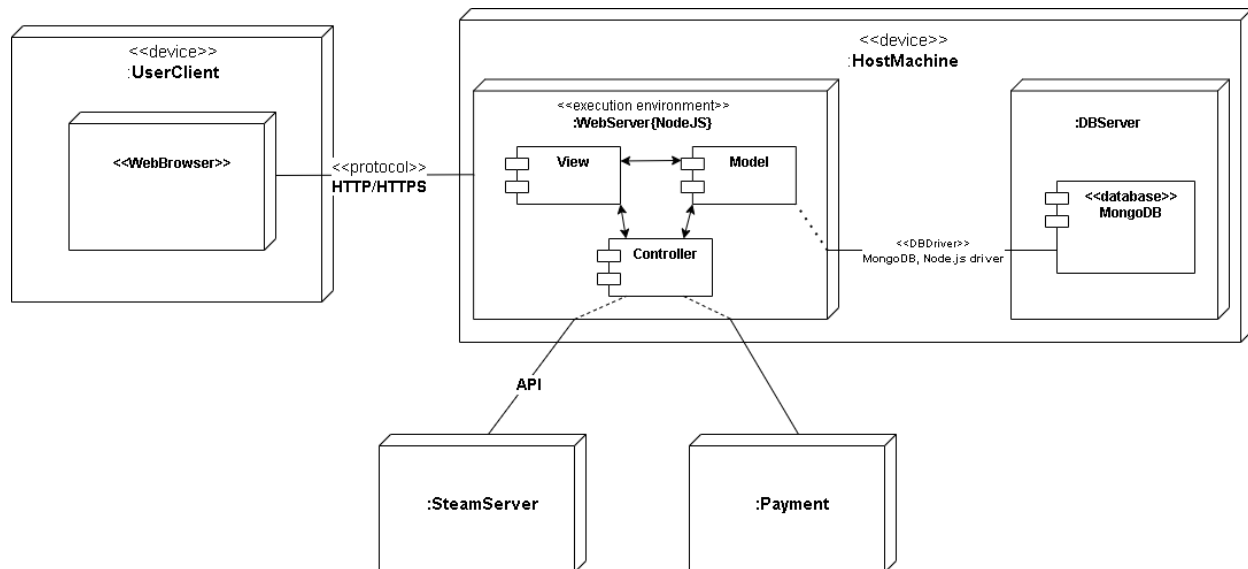
Giải thích key class:

Mongo Documents:

- + **Vật phẩm:** Lưu thông tin chi tiết của vật phẩm
- + **Người dùng:** Lưu các thông tin chi tiết của người dùng
- + **Lịch sử giao dịch:** lưu thông tin lịch sử giao dịch của người dùng
- + **Đánh giá:** lưu thông tin đánh giá về người dùng
- + **Chat:** Lưu thông tin cuộc trò chuyện của người dùng

+ **Bài viết** :Lưu thông tin chi tiết bài viết giới thiệu vật phẩm

5. Deployment



E-commerce CSGO được triển khai trên mô hình web server-client giao tiếp với nhau thông qua các giao thức HTTP/HTTPS và TCP/IP. Mô hình gồm các node chính sau:

+ **Web browser**: Đây là node ở phía người dùng truy cập E-commerce CSGO thông qua web browser.

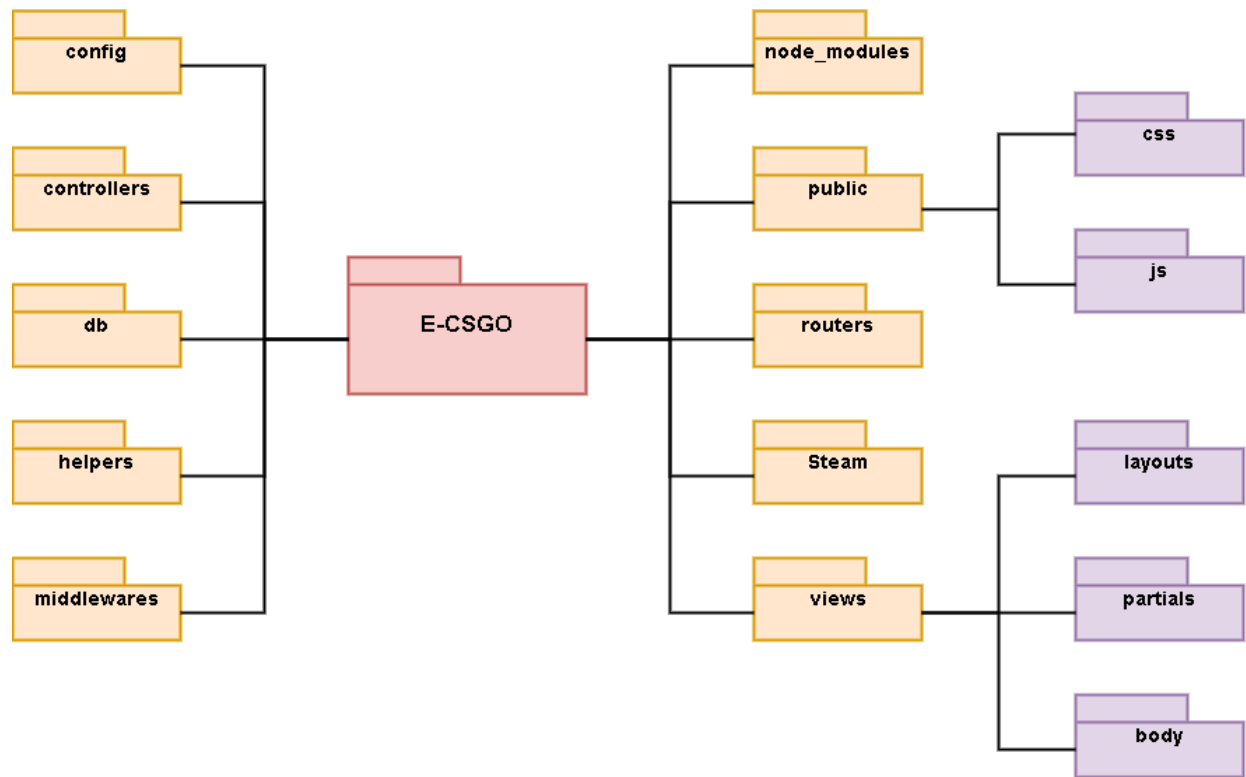
+ **Web server**: Đây là node ở phía server chịu trách nhiệm giao tiếp với người dùng và database. Web được tổ chức theo mô hình MVC, nên khi nhận yêu cầu từ người dùng, server sẽ thực hiện các luồng MVC để thực hiện xử lý và trả lời cho người dùng. Ở phía database, server giao tiếp thông qua giao thức TCP/IP thông qua các cài đặt kết nối và truy xuất ở tầng truy cập dữ liệu (Data access layer).

+ **Database server**: Đây là node ở phía server chịu trách nhiệm lưu trữ cơ sở dữ liệu của trang web và giao tiếp với Web server thông qua việc truy xuất dữ liệu.

+ **Steam Server**: Đây là node chịu trách nhiệm trong việc xử lý thông tin của người dùng, các vật phẩm cũng như các giao dịch.

+ **Payment**: Đây là node chịu trách nhiệm trong việc thanh toán. Trang web thực hiện thanh toán qua cổng thanh toán VNPay, Momo, Paypal.

6. Implementation View



Cấu trúc source code được tổ chức theo mô hình như sau:

- + **Thư mục config:** Chứa config cho các thư viện hỗ trợ
- + **Thư mục controllers:** Chứa các thành phần controllers.
- + **Thư mục routers:** chứa các thành phần để định tuyến
- + **Thư mục steam:** chứa các hàm, quản lý liên quan đến Steam
- + **Thư mục db:** Chứa các thiết lập kết nối và các cài đặt liên quan đến cơ sở dữ liệu.
- + **Thư mục public:** Chứa các file js,css, hình ảnh của website mà người dùng có thể truy cập hoặc để sử dụng trong view
- + **Thư mục views:** Thư mục chứa các templates của các thành phần trang web.
 - **Body:** template phần body
 - **Partials:** template phần header và footer
 - **Layouts:** template chính của trang web
- + **Thư học helpers:** Thư mục chứa các thành phần hỗ trợ của handlebars
- + **Thư mục middlewares:** thư mục chứa các hàm tiền xử lý
- + **Thư mục node_modules:** thư mục chứa các thư viện sử dụng.