BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC GIA ĐỊNH**

**KHOA: CÔNG NGHỆ THÔNG TIN**

Logo, company name

Description automatically generated

**TIỂU LUẬN**

**WEBSITE BÁN THIẾT BỊ MÁY TÍNH TRỰC TUYẾN**

**MÔN: LẬP TRÌNH NODEJS**

Ngành: **CÔNG NGHỆ THÔNG TIN**

Chuyên ngành (*nếu có*): **CÔNG NGHỆ THÔNG TIN**

Giảng viên hướng dẫn: **LÊ HUỲNH PHƯỚC**

Sinh viên thực hiện: **ĐÀM VĂN ANH KHOA**

MSSV: **2108110058**

Lớp: **K15DCPM04**

TP. Hồ Chí Minh, 10 tháng 4 năm 2024

**Khoa/Viện: CÔNG NGHỆ THÔNG TIN**

**NHẬN XÉT VÀ CHẤM ĐIỂM CỦA GIẢNG VIÊN**

**TIỂU LUẬN MÔN: LẬP TRÌNH NODEJS**

**Họ và tên sinh viên: Đàm Văn Anh Khoa**

1. **Tên đề tài**: Website giới thiệu bán thiết bị máy tính trực tuyến

1. **Nhận xét**:

***Những kết quả đạt được:***

***Những hạn chế:***

1. **Điểm đánh giá** *(theo thang điểm 10, làm tròn đến 0.5):*

Sinh viên:……………………………………………………………………………….

Điểm số: ……….…… Điểm chữ: ……………………………………………………..

|  |  |
| --- | --- |
|  | *TP. HCM, ngày … tháng … năm 20……*  **Giảng viên chấm thi**  *(Ký và ghi rõ họ tên)* |

**LỜI MỞ ĐẦU**

Thế giới đang trải qua sự thay đổi không ngừng và thời đại 4.0 đang mở ra những cơ hội và thách thức mới. Nhưng với vai trò của một người trẻ, chúng ta có trách nhiệm tham gia và đóng góp vào công cuộc thay đổi này. Thời đại 4.0 được định nghĩa là sự kết hợp của các công nghệ, giúp thu gọn khoảng cách về vật lý, số hóa và sinh học. Vì vậy, chúng ta cần tiếp thu, học hỏi và sáng tạo để phát triển những thành quả của cuộc cách mạng này và giúp đất nước tiến lên, hòa nhập với thế giới và tiến gần hơn đến các nước phát triển.

Cuộc cách mạng khoa học kỹ thuật phát triển như vũ bão ngày nay là cuộc cách mạng công nghệ. Khoa học phát triển với sự gia tăng 25% / năm đã thu hút một bộ phận lớn nhân lực vào lĩnh vực này làm cho đội ngũ những người làm khoa học gia tăng nhanh chóng. Lực lượng những người làm khoa học tăng lên theo cấp số cộng kéo theo tài liệu khoa học và những sản phẩm nghiên cứu của họ tăng lên theo cấp số nhân. Tất cả đã tạo nên một khối lượng thông tin khổng lồ và không ngừng phát triển, dẫn đến bùng nổ thông tin.

Nhà quản lý, các nhà nghiên cứu, các nhà công nghệ, các nhà giáo dục, các nhà sản xuất kinh doanh, lập trình viên… không chỉ là người dùng tin mà còn là những người sản xuất ra thông tin mới. Sự gia tăng nhanh chóng khối lượng tri thức khoa học tác động mạnh mẽ tới các hoạt động của các cơ quan thông tin – thư viện  như cơ cấu của kho tài liệu, làm cho số lượng và chủng loại tài liệu tăng lên gấp bội.

Thêm vào đó, nhu cầu đòi hỏi rút ngắn đáng kể thời gian hữu ích của tài liệu làm cho các nhà quản lý phải thường xuyên bổ sung vốn tài liệu và không ngừng phải xử lý chúng, hoặc bằng thủ công, hoặc bằng tự động hóa. Sự bùng nổ thông tin gắn liền với sự bùng nổ của công nghệ đặc biệt trên ba lĩnh vực có mối liên hệ chặt chẽ với công tác thông tin – thư viện là: Tin học – Viễn thông và vi xử lý – hạt nhân của công nghệ thông tin hiện đại.

Ngày nay Internet đã trở thành dịch vụ phổ biến và thiết yếu và có ảnh hưởng

sâu rộng tới thói quen, sinh hoạt, giải trí của nhiều người. Cùng với sự phát triển

nhanh chóng của Internet thì các hình thức mua và bán hàng hóa cho mọi người

ngày càng đa dạng và phát triển hơn. Các ứng dụng Web ngày càng trở nên phổ

biến. Trước nhu cầu đó, cùng với yêu cầu môn học, nhóm chúng em quyết định

chọn đề tài Xây dựng Website bán hàng trực tuyến, cụ thể là bán thiết bị máy tính.

Đây là một hiện tượng phổ biến ngày nay, đang có chiều hướng về cuộc sống hiện đại của con người chúng ta trong thời đại ngày hôm nay và tương lai. Tạo ra nhiều xu hướng, đáp ứng những nhu cầu cần thiết của con người, xây dựng những ứng dụng có ích cho xã hội và cá nhân, tiến bộ vượt bậc theo từng thời điểm, giúp công nghệ số phát triển thêm trong tương lai không xa.

MỤC LỤC

[PHẦN 1: TÌM HIỂU VỀ CÔNG NGHỆ 6](#_Toc164164357)

[1.1. Lập trình viên là gì? 6](#_Toc164164358)

[1.2. Lập trình viên học ngành gì? 6](#_Toc164164359)

[1.3. Nhu cầu nhân lực và cơ hội việc làm cho lập trình viên 8](#_Toc164164360)

[1.4. Lập trình NodeJS 9](#_Toc164164361)

[1.4.1. NodeJS là gì? 9](#_Toc164164362)

[1.4.2. Các thuật ngữ liên quan đến NodeJS 10](#_Toc164164363)

[1.4.3. Tính chất của NodeJS 11](#_Toc164164364)

[1.4.4. Ưu điểm của NodeJS 11](#_Toc164164365)

[1.4.5. Thành phần của NodeJS 13](#_Toc164164366)

[1.4.6. NodeJS phù hợp với những ứng dụng nào? 18](#_Toc164164367)

[1.5. MongoDB 20](#_Toc164164368)

[1.5.1. MongoDB là gì? 20](#_Toc164164369)

[1.5.2. Công dụng của MongoDB 21](#_Toc164164370)

[1.5.3. Các thuật ngữ MongoDB thường dùng 21](#_Toc164164371)

[1.5.4. Mối tương quan giữa thuật ngữ MongoDB và RDBMS 23](#_Toc164164372)

[1.5.5. MongoDB hoạt động như thế nào? 24](#_Toc164164373)

[1.5.6. Ưu điểm và khuyết điểm của MongoDB 27](#_Toc164164374)

[**1.5.7. So sánh MongoDB và MySQL** 29](#_Toc164164375)

[PHẦN 2: WEBSITE BÁN THIẾT BỊ MÁY TÍNH BẰNG NODEJS VÀ MONGODB 30](#_Toc164164376)

[1. Usecase diagram Admin LV0 30](#_Toc164164377)

[2. Usecase diagram Seller LV0 30](#_Toc164164378)

[3. Usecase diagram User LV0 31](#_Toc164164379)

[4. Usecase diagram Admin LV1 33](#_Toc164164380)

[5. Usecase diagram User LV1 35](#_Toc164164381)

[6. Usecase Description Admin 39](#_Toc164164382)

[6.1. Use case 1.0 + 1.2: (UC1.0 + 1.2- Search for an account and Show a list of current accounts) 39](#_Toc164164383)

[6.2. Use case 1.1: (UC1.1-Create a child admin account) 41](#_Toc164164384)

[6.3. Use case 1.3: (UC1.3-Account activation/blocking) 43](#_Toc164164385)

[6.4. Use case 2.0 + 2.2: (UC2.0 + 2.2-Search the snar catalog and Show the current category list) 45](#_Toc164164386)

[6.5. Use case 2.1: (UC2.1-Create a new category) 47](#_Toc164164387)

[6.6. Use case 2.3: (UC2.3-Edit a catalogue) 49](#_Toc164164388)

[6.7. Use case 2.4: (UC2.4- Delete a category) 51](#_Toc164164389)

[6.8. Use case 3.0 + 3.2: (UC3.0 + 3.2-Search for products and Show current product list) 53](#_Toc164164390)

[6.9. Use case 3.1: (UC3.1-View product details) 55](#_Toc164164391)

[6.10. Use case 3.3: (UC3.3- Approve) 57](#_Toc164164392)

[6.11. Use case 4.0 + 4.2: (UC4.0 + 4.2- Search for suppliers and Show supplier list) 59](#_Toc164164393)

[6.12. Use case 4.1: (UC4.1-Create a new provider) 61](#_Toc164164394)

[Tài liệu tham khảo 62](#_Toc164164395)

# PHẦN 1: TÌM HIỂU VỀ CÔNG NGHỆ

## Lập trình viên là gì?

Lập trình viên có thể làm các hình thức lập trình như lập trình game, lập trình mobile, lập trình web, lập trình hệ thống và lập trình Database. Công việc chính của người lập trình là viết phần mềm với các công việc chính là xây dựng mới một ứng dụng, nâng cấp và sửa chữa các ứng dụng có sẵn, xây dựng các chức năng xử lý, nghiên cứu và phát triển công nghệ mới,…cụ thể công việc như sau:

Lập trình viên là những người tạo ra những đoạn code bằng các ngôn ngữ lập trình, nền tảng và các công cụ hỗ trợ như Html, Css, Java, Python, C++, C#, PHP… để xây dựng nên các phần mềm, ứng dụng, website, App… trên nền tảng máy tính và di động.

Lập trình viên cập nhật chương trình và mở rộng chương trình có sẵn, phát triển phần mềm và thiết kế phần mềm, xây dựng và sử dụng công cụ phần mềm hỗ trợ máy tính(CASE) để tự động hóa đoạn mã code.

## 1.2. Lập trình viên học ngành gì?

Lập trình viên là nghề cần có khối lượng kiến thức nền tảng vững và đồ sộ nên các bạn sinh viên cần học chắc, tư duy logic, sáng tạo cho các ngành học dành cho lập trình viên. Các bạn có nền tảng học ngành Công nghệ thông tin ra trường sẽ thường theo đuổi nghề lập trình viên. Tuy nhiên không phải tất cả lập trình viên đều học công nghệ thông tin và ngược lại.

Thông thường để trở thành một lập trình viên sẽ cần học một số chuyên ngành quen thuộc như sau:

**Hệ thống thông tin**

Hệ thống thông tin là ngành học ở mức độ tổng quan bao gồm việc tìm hiểu về quy trình thu thập thông tin, phân tích, đánh giá, thiết kế hệ thống thông tin, quản trị, vận hành và phân phối thông tin tham mưu cho những người ra quyết định. Một số ngôn ngữ lập trình bạn sẽ học như: Java, PHP, C#, SQL… Trong đó tập trung vào SQL nhiều hơn với mục đích quản lý và phân tích cơ sở dữ liệu.

Hệ thống thông tin mới chỉ là kiến thức tổng quát, chưa đi sâu vào kiến thức lập trình phần mềm. Nếu muốn trở thành lập trình viên, bạn cần học thêm nhiều mảng nữa. Có thể học trong các trường đại học, cao đẳng, học online hoặc trung tâm dạy lập trình…

**Khoa học máy tính**

Khoa học máy tính là chuyên ngành phù hợp với các bạn yêu thích nghiên cứu, tìm hiểu về máy tính từ những điều nhỏ nhất. Các môn học của ngành khoa học máy tính sẽ là các môn học về cấu trúc dữ liệu máy tính, trí tuệ nhân tạo AI, ngôn ngữ lập trình, hệ điều hành, xử lý dữ liệu,… các kiến thức liên quan đến toán học, khoa học và máy tính chuyên sâu.

**Kỹ thuật máy tính**

Kỹ thuật máy tính cũng là ngành học mà các bạn được học các ngôn ngữ lập trình như Java, PHP, C++…để phát triển phần cứng cho máy tính, ko chuyên sâu về phần mềm của máy tính như thiết kế chip máy tính, thiết kế hệ thống điều khiển IOT, hệ thống điều khiển tự động, công nghệ Robotic. Tuy nhiên để trở thành lập trình viên thì bạn cần học thêm các kiến thức chuyên sâu hơn về kiến thức của ngành kỹ thuật máy tính.

**Công nghệ phần mềm**

Công nghệ phần mềm sẽ là chuyên ngành mà bạn được học về quy trình xây dựng và phát triển phần mềm, các công cụ và nền tảng công nghệ, kiến thức về thu thập yêu cầu, phân tích, thiết kế, lập trình, kiểm thử, vận hành và bảo trì phần mềm… Đây sẽ là ngành học chuyên sâu về công việc của một lập trình viên nhất.

**Mạng máy tính và Viễn thông**

Bạn có thể trở thành một nhà thiết kế và phát triển phần mềm mạng hoặc nhân sự chuyên thiết kế mạng cho các trung tâm dữ liệu, nhà cung cấp và ngân hàng. Ứng dụng truyền thông xã hội và công nghệ web, ứng dụng công nghệ trong thông tin quản lý, giám sát hoạt động của doanh nghiệp hiện đang là hai lĩnh vực được chú trọng trong việc phát triển xã hội hiện nay. Một lập trình viên giỏi có thể xây dựng nền tảng và kỹ năng từ chuyên ngành mạng máy tính và viễn thông này.

## 1.3. Nhu cầu nhân lực và cơ hội việc làm cho lập trình viên

Trên thị trường cung ứng nhân lực về lập trình viên hiện nay vẫn còn thiếu rất nghiêm trọng trải qua 3 năm gần đây. Cụ thể, năm 2019, thị trường dự báo nhu cầu nhân sự về lập trình viên của Việt Nam thiếu hụt từ 70.000 – 90.000, năm 2020 là 100.000 nhân sự và năm 2021 là 190.00 nhân sự và còn có xu hướng tăng lên trong những năm sắp tới đặc biệt là nhân sự chất lượng đáp ứng được nhu cầu công việc của các doanh nghiệp, tổ chức.

Nghề lập trình viên hiện nay có mức lương và mức thu nhập khủng so với các ngành nghề khác, bởi giá trị người làm nghề này mang lại cho doanh nghiệp là vô cùng lớn. Thực trạng nhân sự về lập trình viên hiện nay chính là nguồn cung không đủ cầu nên vô hình chung tăng giá trị thu nhập cho những người đáp ứng được chất lượng công việc.

Chính vì nhu cầu thị trường về nhân lực nghề lập trình viên lớn nên tạo rất nhiều cơ hội việc làm cho các bạn nhân sự theo ngành công nghệ thông tin để theo đuổi nghề lập trình viên. Các bạn sẽ không lo thiếu việc, chỉ cần bạn chọn đúng chuyên ngành, cố gắng trau dồi kiến thức, cọ xát và trải nghiệm môi trường làm việc thì sẽ có rất nhiều cơ hội việc làm với mức lương tốt ngay từ khi bạn mới ra trường.

## 1.4. Lập trình NodeJS

### 1.4.1. NodeJS là gì?

Được phát hành vào năm 2009, NodeJS, hay còn được biết với tên gọi chính thức là Node.js, là môi trường thời gian chạy (runtime environment) JavaScript đa nền tảng và mã nguồn mở. NodeJS cho phép các lập trình viên tạo cả ứng dụng front-end và back-end bằng JavaScript.

Điều này nghe có vẻ vừa đơn giản vừa thú vị. Nhưng đối với người mới bắt đầu, định nghĩa này có thể vẫn còn chưa rõ ràng. Vì vậy, hãy chia nhỏ định nghĩa này để hiểu rõ ý nghĩa của NodeJS.

#### 1.4.1.1. NodeJS là mã nguồn mở

Điều này có nghĩa là mã nguồn của NodeJS được cung cấp công khai. Và được duy trì bởi những người đóng góp từ khắp nơi trên thế giới.

#### 1.4.1.2. NodeJS hỗ trợ đa nền tảng

NodeJS không phụ thuộc vào bất kỳ phần mềm hệ điều hành nào mà đều có thể hoạt động trên Linux, macOS hoặc Windows.

#### 1.4.1.3. NodeJS là môi trường thời gian chạy mã JavaScript

Khi bạn viết code JavaScript trong trình soạn thảo văn bản, code đó không thể thực hiện bất kỳ tác vụ nào trừ khi bạn thực thi (hoặc chạy) nó. Và để chạy code, bạn cần có môi trường thời gian chạy.

Các trình duyệt như Chrome và Firefox có môi trường thời gian chạy. Đó là lý do tại sao họ có thể chạy code JavaScript. Trước khi NodeJS được tạo, JavaScript chỉ có thể chạy trên trình duyệt và chỉ được sử dụng để xây dựng các ứng dụng front-end.

NodeJS cung cấp môi trường thời gian chạy bên ngoài trình duyệt. Nó cũng được xây dựng trên công cụ JavaScript của Chrome (V8 Engine). Điều này giúp bạn có thể xây dựng các ứng dụng back-end bằng cách sử dụng cùng ngôn ngữ lập trình JavaScript mà bạn quen thuộc.

### 1.4.2. Các thuật ngữ liên quan đến NodeJS

#### 1.4.2.1. I/O (input/output)

Là viết tắt của input/output, thuật ngữ I/O chủ yếu đề cập đến sự tương tác của chương trình với hệ thống.

Ví dụ: Các hoạt động I/O có thể bao gồm việc đọc/ ghi dữ liệu từ/ vào disk, tạo các yêu cầu HTTP và trao đổi với cơ sở dữ liệu. Hoạt động này rất chậm so với việc truy cập bộ nhớ (RAM) hoặc thực hiện công việc trực tiếp trên CPU.

#### 1.4.2.2. Không đồng bộ (Asynchronous)

Thực thi không đồng bộ đề cập đến cách thực thi không theo trình tự xuất hiện trong code. Trong lập trình không đồng bộ, chương trình sẽ không đợi tác vụ hoàn thành mà đã có thể chuyển sang tác vụ tiếp theo.

#### 1.4.2.3. Không chặn (Non-blocking)

Chặn (blocking) đề cập đến hành động chặn việc thực thi tiếp theo cho đến khi tác vụ đó kết thúc trong khi không chặn (non-blocking) đề cập đến hành động không chặn việc thực thi.

Kết hợp với thuật ngữ “không đồng bộ” ở trên, bạn có thể hiểu rằng các phương thức non-blocking diễn ra một cách không đồng bộ.

#### 1.4.2.4. Sự kiện (Event) và Lập trình Hướng sự kiện (Event-driven programming)

Sự kiện là các hành động do người dùng hoặc hệ thống tạo ra, như nhấp chuột, tải xuống tệp hoàn tất hoặc lỗi phần cứng hoặc phần mềm.

Lập trình hướng sự kiện là một mô hình lập trình trong đó luồng chương trình được xác định bởi các sự kiện. Một chương trình hướng sự kiện sẽ thực hiện các hành động để đáp lại các sự kiện. Một sự kiện xảy ra sẽ kích hoạt hàm callback.

### 1.4.3. Tính chất của NodeJS

#### 1.4.3.1. Không đồng bộ về bản chất và Hướng sự kiện (Asynchronous in Nature and Event-driven)

Các máy chủ được tạo bằng NodeJS không bao giờ chờ đợi dữ liệu từ một [API](https://itviec.com/blog/api-la-gi/) cụ thể, thay vào đó, nó sẽ chuyển trực tiếp sang API tiếp theo.

Để nhận và theo dõi tất cả phản hồi của các yêu cầu API trước đó, NodeJS tuân theo cơ chế hướng sự kiện. Do đó, chúng ta có thể nói rằng về bản chất tất cả các API của NodeJS đều không bị chặn (non-blocking).

#### 1.4.3.2Kiến trúc đơn luồng (Single Threaded Architecture)

NodeJS hoạt động trên một luồng duy nhất.

Dựa trên kiến trúc “Mô hình vòng lặp sự kiện đơn luồng” (Single Threaded Event Loop Model), NodeJS có thể xử lý nhiều yêu cầu của máy khách.

Vòng lặp sự kiện được thực thi trên một luồng chính duy nhất. Vòng lặp sự kiện cho phép NodeJS thực hiện tất cả các hoạt động I/O không chặn mặc dù JavaScript là đơn luồng.

### 1.4.4. Ưu điểm của NodeJS

#### 1.4.4.1. NodeJS có thể mở rộng

Các ứng dụng NodeJS có khả năng mở rộng cao vì chúng hoạt động không đồng bộ vì các yêu cầu đồng thời có thể được xử lý rất hiệu quả bằng NodeJS.

NodeJS hoạt động trên một luồng đơn nên khi có một yêu cầu đến, NodeJS sẽ bắt đầu xử lý yêu cầu đó và sẵn sàng xử lý yêu cầu tiếp theo.

Tính năng hấp dẫn nhất của NodeJS là khả năng phân vùng các ứng dụng theo chiều ngang và quy trình phân vùng này chủ yếu đạt được nhờ sử dụng các tiến trình con. Bằng cách sử dụng tính năng này, các phiên bản ứng dụng riêng biệt được cung cấp cho các đối tượng mục tiêu khác nhau và cho phép chúng đáp ứng sở thích của khách hàng.

#### 1.4.4.2. Thời gian thực thi code nhanh

Công cụ thời gian chạy (runtime motor) JavaScript V8 được NodeJS sử dụng và cũng được Google Chrome sử dụng. Một trình bao bọc được trung tâm cung cấp cho JavaScript và vì lý do đó, công cụ thời gian chạy trở nên nhanh hơn.

Công cụ JavaScript V8 của Google Chrome là nền tảng của NodeJS, cho phép thực thi mã nhanh hơn. Công cụ này biên dịch code JavaScript thành code máy, giúp code được triển khai hiệu quả và dễ dàng hơn và nhanh hơn.

Đồng thời, việc sử dụng các khái niệm như lập trình không đồng bộ và cách vận hành non-blocking trên các hoạt động I/O cũng giúp nâng cao hiệu suất của NodeJS.

#### 1.4.4.3. Khả năng tương thích trên nhiều nền tảng

Các loại hệ điều hành khác nhau như Windows, UNIX, LINUX, MacOS và các thiết bị di động khác đều có thể sử dụng NodeJS.

#### 1.4.4.4. Sử dụng JavaScript

NodeJS sử dụng JavaScript. Hầu hết các lập trình viên đều quen thuộc với JavaScript, vì vậy đối với họ, việc hiểu NodeJS trở nên rất dễ dàng hơn.

#### 1.4.4.5. Truyền dữ liệu nhanh

Thời gian xử lý những dữ liệu đã được truyền đến các luồng khác nhau thường sẽ mất nhiều thời gian. Trong khi đó, để xử lý dữ liệu, NodeJS chỉ mất một khoảng thời gian rất ngắn và thực hiện với tốc độ nhanh.

NodeJS tiết kiệm rất nhiều thời gian vì các tệp được NodeJS xử lý và tải lên đồng thời. Do đó, tốc độ tổng thể của truyền dữ liệu và video được cải thiện nhờ NodeJS.

#### 1.4.4.6. Không có bộ đệm

Dữ liệu không bao giờ được lưu vào bộ đệm trong ứng dụng NodeJS.

#### 1.4.4.7. Tiết kiệm thời gian, công sức và chi phí

NodeJS được trang bị một kho thư viện khổng lồ – NPM (Node Package Manager). Các developer có thể sử dụng lại các module trong code và đưa các chức năng đa dạng vào bất kỳ ứng dụng nào.

Kho lưu trữ nguồn mở này giúp giảm đáng kể chi phí và nỗ lực phát triển, đồng thời cũng rút ngắn thời gian triển khai và thúc đẩy các giải pháp đổi mới.

#### 1.4.4.8. Một ngôn ngữ

Một trong những lợi ích chính của NodeJS là khả năng viết toàn bộ cơ sở hạ tầng của bất kỳ ứng dụng web nào chỉ bằng một ngôn ngữ – JavaScript. Do đó, lập trình viên không cần tốn nhiều thời gian học các ngôn ngữ khác nhau để đáp ứng.

#### 1.4.4.9. Hỗ trợ cộng đồng mạnh mẽ

NodeJS có sự hỗ trợ cộng đồng và hỗ trợ mạnh mẽ vì là mã nguồn mở. Vì vậy, các nhà phát triển có thể tìm kiếm sự trợ giúp từ các chuyên gia từ khắp nơi trên thế giới. Điều này thúc đẩy các dự án phát triển.

### 1.4.5. Thành phần của NodeJS

#### 1.4.5.1. Module

Các module giống như các thư viện JavaScript có thể được sử dụng trong ứng dụng NodeJS để bao gồm một tập hợp các hàm.

Để đưa một module vào ứng dụng NodeJS, hãy sử dụng hàm require() với dấu ngoặc đơn chứa tên của module.

NodeJS có nhiều module cung cấp chức năng cơ bản cần thiết cho một ứng dụng web. Một số trong số đó được đề cập trong bảng này:

|  |  |
| --- | --- |
| Module chính | Mô tả |
| http | Bao gồm class, method và event để tạo máy chủ NodeJS http |
| util | Bao gồm các hàm hữu dụng cho lập trình viên |
| fs | Bao gồm các sự kiện, class, và phương thức để xử lý file vận hành I/O |
| url | Bao gồm những phương thức phân tích cú pháp URL |
| querystring | Bao gồm những phương thức để làm việc với query string |
| stream | Bao gồm những phương thức để xử lý các luồng dữ liệu |
| zlib | Bao gồm những phương thức để nén hoặc giải nén file |

*Các module trong NodeJS*

#### 1.4.5.2. Bảng điều khiển (Console)

Bảng điều khiển là một module cung cấp phương pháp gỡ lỗi tương tự như bảng điều khiển JavaScript cơ bản do trình duyệt cung cấp. Nó print tin nhắn tới thiết bị stdout and stderr.

//Viết "hello world" đến console

console.log('hello world');

#### 1.4.5.3. Cụm (Cluster)

NodeJS được xây dựng dựa trên khái niệm lập trình đơn luồng. Cụm là một module cho phép thực hiện đa luồng bằng cách tạo các tiến trình con chia sẻ cùng một cổng máy chủ và chạy đồng thời.

Một cụm có thể được thêm vào một ứng dụng theo cách sau:

//Bao gồm module cụm trong ứng dụng

var cluster = require('clustr');

if (cluster.isWorker) {

console.log('Child thread');

} else {

console.log('Parent thread');

cluster.fork();

cluster.fork();

}

#### 1.4.5.4. Đối tượng toàn cục (Global)

Các đối tượng toàn cục trong NodeJS có sẵn trong tất cả các module. Các đối tượng này là các hàm, module, chuỗi, v.v. Một số đối tượng toàn cục của NodeJS được đề cập trong bảng bên dưới:

|  |  |
| --- | --- |
| Đối tượng toàn cục | Mô tả |
| \_\_\_dirname | Chỉ định tên các thư mục chứa code |
| \_\_\_filename | Chỉ định tên file chứa code |
| exports | Là trích dẫn của module.exports nhưng ngắn hơn |
| module | Là trích dẫn cho module hiện tại |
| require | Dùng để nhập module, file local và JSON |

*Đối tượng toàn cục*

#### 1.4.5.5. Xử lý lỗi

Các ứng dụng NodeJS thường gặp phải bốn loại lỗi phổ biến sau:

|  |  |  |
| --- | --- | --- |
| 4 lỗi trong trải nghiệm ứng dụng NodeJS | Lỗi JavaScript thông thường | <EvalError>, <SyntaxError>, <RangeError>, <ReferenceError>, <TypeError>, <URIError> |
| Lỗi hệ thống | <File không tồn tại>, <đóng socket> |
| Lỗi người dùng | Lỗi chỉ định bởi user trong code |
| Lỗi assertion | Loại lỗi đặc biệt chỉ xảy ra khi vi phạm logic |

*Lỗi trong NodeJS*

Lỗi trong NodeJS được xử lý thông qua các ngoại lệ. Ví dụ: hãy xử lý lỗi xảy ra khi chúng ta chia một số cho 0. Lỗi này sẽ làm hỏng ứng dụng NodeJS, vì vậy chúng ta nên xử lý lỗi này để tiếp tục thực thi ứng dụng bình thường.

try {

var m = 1;

var n = 1/O;

}

catch (err) {

//Xử lý lỗi ở đây

}

#### 1.4.5.6. Luồng (Streaming)

Luồng là đối tượng cho phép bạn đọc hoặc ghi dữ liệu liên tục. Có bốn loại luồng:

1. Có thể đọc được: Đây là các loại luồng mà dữ liệu có thể được đọc
2. Có thể ghi: Đây là các loại luồng mà dữ liệu có thể được ghi vào
3. Duplex: Đây là các loại luồng có thể được đọc và ghi
4. Chuyển đổi: Các luồng có thể thao tác dữ liệu trong khi nó đang được đọc hoặc ghi

#### 1.4.5.7. Bộ đệm (Buffer)

Bộ đệm là một module cho phép xử lý các luồng chỉ chứa dữ liệu nhị phân. Một bộ đệm trống có độ dài ’10’ có thể được tạo bằng phương pháp này:

var buf = Buffer.alloc(10);

#### 1.4.5.8. Miền (Domain)

Module miền chặn các lỗi vẫn chưa được xử lý. Hai phương pháp được sử dụng để chặn các lỗi này:

1. Liên kết nội bộ: Bộ phát lỗi thực thi code của nó bên trong phương thức chạy
2. Liên kết bên ngoài: Bộ phát lỗi được thêm rõ ràng vào miền thông qua phương thức thêm của nó

#### 1.4.5.9. DNS

Module DNS được sử dụng để kết nối với máy chủ DNS và thực hiện phân giải (resolve) tên bằng phương pháp sau:

dns.resolve()

Module DNS cũng được sử dụng để thực hiện phân giải tên mà không cần kết nối mạng bằng phương pháp sau:

dns.lookup()

#### 1.4.5.10. Trình gỡ lỗi (Debugger)

NodeJS bao gồm một tiện ích gỡ lỗi mà có thể truy cập thông qua một ứng dụng gỡ lỗi tích hợp sẵn. Trình gỡ lỗi NodeJS không có nhiều tính năng nhưng hỗ trợ kiểm tra code đơn giản.

Trình gỡ lỗi có thể được sử dụng trong thiết bị đầu cuối bằng cách sử dụng từ khóa *inspect* trước tên tệp JavaScript. Để kiểm tra một tệp, ví dụ: myscript.js, bạn có thể làm theo phương pháp này:

$ node inspect myscript.js

### 1.4.6. NodeJS phù hợp với những ứng dụng nào?

**Từ công ty khởi nghiệp đến các doanh nghiệp thuộc nhiều quy mô khác nhau đều có thể chọn NodeJS khi cần phát triển một ứng dụng web ổn định và có thể mở rộng. Do tính chất hướng đến sự kiện, NodeJS vượt trội hơn các mô hình làm việc khác.**

**Sau đây là một vài ví dụ ứng dụng nổi bật:**

#### 1.4.6.1. Công cụ cộng tác thời gian thực (Real-Time Collaboration Tools)

NodeJS là một công cụ mạnh mẽ để tạo ra các công cụ cộng tác theo thời gian thực, từ ứng dụng làm việc chung đến quản lý dự án, hội thảo video và âm thanh cũng như chỉnh sửa tài liệu cộng tác.

Điều đó giải thích tại sao NodeJS là nền tảng cho nhiều công cụ cộng tác phổ biến thuộc sở hữu của các công ty công nghệ lớn, chẳng hạn như Trello để quản lý dự án và Slack để trò chuyện nhóm và liên lạc nhóm từ xa. Kiến trúc dựa trên sự kiện và không đồng bộ của NodeJS cung cấp nền tảng và hoàn toàn phù hợp với việc xây dựng và mở rộng các ứng dụng cộng tác.

#### 1.4.6.2. Ứng dụng thời gian thực

Xây dựng các ứng dụng mạng thời gian thực là một trong những tính năng tốt nhất của NodeJS. Bạn có thể làm tất cả, từ xây dựng các ứng dụng trò chuyện thời gian thực như ứng dụng nhắn tin tức thời (IM) và trò chuyện chuyển tiếp qua internet (IRC) đến xây dựng các ứng dụng thời gian thực phức tạp.

NodeJS cung cấp các tính năng cơ bản để xây dựng và triển khai các ứng dụng mạng và thời gian thực bằng API sự kiện. Nó có thể tạo một đối tượng, được gọi là bộ phát, phát ra các sự kiện được đặt tên theo định kỳ và có thể được xử lý bởi các trình xử lý sự kiện.

Nhờ kiến trúc dựa trên sự kiện, NodeJS hoạt động hiệu quả với giao thức WebSockets, hỗ trợ giao tiếp hai chiều theo thời gian thực giữa máy chủ và máy khách.

NodeJS có hỗ trợ WebSockets tuyệt vời thông qua các thư viện như Socket.io và WebSocket-node, mà bạn có thể sử dụng để tạo và triển khai các cuộc trò chuyện theo thời gian thực một cách nhanh chóng và hiệu quả.

#### 1.4.6.3. Ứng dụng dựa trên vị trí

Bản chất không đồng bộ của NodeJS và khả năng cung cấp các bản cập nhật theo thời gian thực khiến nó trở thành lựa chọn hàng đầu để xây dựng các ứng dụng dựa trên vị trí địa lý.

Vào năm 2020, số lượng ứng dụng dựa trên vị trí đã tăng lên. Hầu hết các ứng dụng này sử dụng NodeJS và các ngăn xếp liên quan khác do các tính năng của NodeJS như kiến trúc dựa trên sự kiện và lập trình không đồng bộ để xây dựng các ứng dụng mạng và thời gian thực.

#### 1.4.6.4. Ứng dụng một trang (Single-Page App)

Ứng dụng một trang (SPA) không phải là khái niệm mới đối với lập trình web. Đó là một từ thông dụng phổ biến mô tả cách tiếp cận trong đó toàn bộ ứng dụng nằm gọn trên một trang duy nhất, với trải nghiệm tốt hơn của lập trình viên.

NodeJS rất lý tưởng cho các SPA vì nó xử lý các cuộc gọi không đồng bộ và khối lượng công việc I/O nặng nhanh hơn và hiệu quả hơn. Để chuyển đổi dữ liệu liền mạch giữa Chế độ xem và máy chủ, vòng lặp sự kiện NodeJS được phép “trì hoãn” nhiều yêu cầu đồng thời từ máy khách, cho phép xử lý trơn tru.

NodeJS cũng là một lựa chọn tuyệt vời để xây dựng phần back-end của SPA với cách tiếp cận dựa trên dữ liệu. Chúng ta có thể sử dụng các khung JavaScript như Express.js, Adonis.js hoặc Koa để lập trình phần back-end SPA phức tạp và sử dụng nhiều dữ liệu.

Ngoài ra, bạn cũng có thể phát triển các loại ứng dụng web sau bằng NodeJS:

* Ứng dụng quản lý dự án
* Ứng dụng truyền thông xã hội
* Ứng dụng trò chơi
* Ứng dụng phát video trực tiếp
* Diễn đàn thảo luận hoặc trò chuyện
* Ứng dụng IoT

## 1.5. MongoDB

### 1.5.1. MongoDB là gì?

MongoDB là một phần mềm mã nguồn mở dùng để quản trị cơ sở dữ liệu NoSQL.

NoSQL (Not only SQL) được sử dụng thay thế cho cơ sở dữ liệu quan hệ (Relational Database – RDB) truyền thống. Cơ sở dữ liệu NoSQL khá hữu ích trong khi làm việc với các tập dữ liệu phân tán lớn. MongoDB là một công cụ có thể quản lý thông tin hướng document cũng như lưu trữ hoặc truy xuất thông tin.

Trong khi đó, ngôn ngữ truy vấn có cấu trúc (SQL) là ngôn ngữ lập trình được tiêu chuẩn hóa, dùng để quản lý cơ sở dữ liệu quan hệ. Dữ liệu được chuẩn hóa SQL dưới dạng schema và table và mọi table đều có cấu trúc cố định.

Hiện nay, có nhiều công ty toàn cầu sử dụng MongoDB để lưu trữ lượng dữ liệu “khổng lồ” của họ như Facebook, Nokia, eBay, Adobe, Google,…

### 1.5.2. Công dụng của MongoDB

MongoDB giúp các tổ chức lưu trữ lượng lớn dữ liệu trong khi vẫn hoạt động nhanh chóng. Ngoài lưu trữ dữ liệu, MongoDB còn được sử dụng trong các trường hợp sau:

* Tích hợp một lượng lớn dữ liệu đa dạng
* Mô tả các cấu trúc dữ liệu phức tạp, biến hoá
* Cung cấp dữ liệu cho các ứng dụng hiệu suất cao
* Hỗ trợ các ứng dụng đám mây lạ và đa đám mây
* Hỗ trợ phương pháp phát triển Agile

Thay vì sử dụng các table và row như trong cơ sở dữ liệu quan hệ, vì là cơ sở dữ liệu NoSQL, MongoDB được tạo thành từ collection và document. Document được tạo thành từ các cặp khóa-giá trị (là đơn vị dữ liệu cơ bản của MongoDB). Còn collection, tương đương với table trong SQL, là nơi chứa các bộ document.

### 1.5.3. Các thuật ngữ MongoDB thường dùng

**\_id**

\_id là một trường bắt buộc trong mọi document của MongoDB. \_id được sử dụng để đại diện cho tính duy nhất của một document trong một collection. Trường \_id hoạt động giống như khóa chính (primary key) của document.

\_id là một số thập lục phân 12 byte đảm bảo tính duy nhất của mọi document. Bạn có thể cung cấp \_id trong khi chèn document. Trong 12 byte này:

* 4 byte đầu tiên đại diện cho thời điểm hiện tại (dựa trên hệ giây của Unix Epoch);
* 3 byte tiếp theo cho id máy;
* 2 byte tiếp theo cho process id của máy chủ MongoDB;
* 3 byte cuối cùng là giá trị gia tăng đơn giản.

Nếu bạn không cung cấp được số id thì MongoDB sẽ tự động cung cấp một id duy nhất cho document của bạn.

**Document**

Document là đơn vị lưu trữ dữ liệu cơ bản trong cơ sở dữ liệu MongoDB. Document mang vai trò tương tự như row trong các hệ thống cơ sở dữ liệu quan hệ truyền thống.

Document là một cách để sắp xếp và lưu trữ dữ liệu dưới dạng một tập hợp các cặp field-value. Document trong MongoDB không cần phải có cùng một bộ field hoặc cấu trúc với các document khác trong cùng một collection.

Đồng thời, các field chung trong document của một collection có thể chứa các loại dữ liệu khác nhau.

**Collection**

Collection là một tập hợp các document MongoDB. Collection tương tự như table trong hệ thống cơ sở dữ liệu quan hệ. Các collection có tính chất schema less, do đó các document trong cùng một collection có thể có các trường khác nhau.

Thông thường, một collection chứa các document có mục đích tương tự hoặc liên quan với nhau.

**Database**

Trong MongoDB, database là một container vật lý chứa tập hợp các collection. Một database có thể chứa 0 collection hoặc nhiều collection.

Một phiên bản máy chủ MongoDB có thể lưu trữ nhiều database và không có giới hạn về số lượng database có thể được lưu trữ trên một phiên bản, nhưng giới hạn ở không gian bộ nhớ ảo có thể được phân bổ bởi hệ điều hành.

### 1.5.4. Mối tương quan giữa thuật ngữ MongoDB và RDBMS

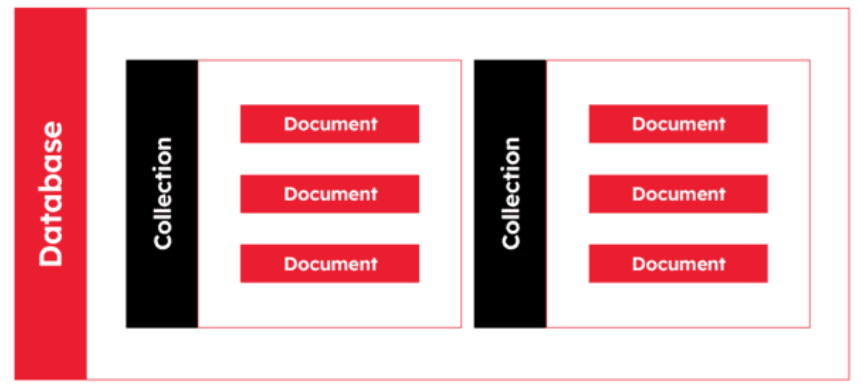
|  |  |
| --- | --- |
| RDBMS | MongoDB |
| Database | Database |
| Table | Collection |
| Tuple/Row | Document |
| Column | Field |
| Table Join | Embedded Documents |
| Primary Key | Primary Key (Khóa mặc định \_id do chính MongoDB cung cấp) |

### 1.5.5. MongoDB hoạt động như thế nào?

#### 1.5.5.1. MongoDB lưu trữ dữ liệu như thế nào?

Như chúng ta biết rằng MongoDB là một máy chủ cơ sở dữ liệu và dữ liệu được lưu trữ trong các cơ sở dữ liệu này. Hay nói cách khác, môi trường MongoDB cung cấp cho bạn một máy chủ mà bạn có thể khởi động và sau đó tạo nhiều cơ sở dữ liệu trên đó bằng MongoDB.

Nhờ vào cơ sở dữ liệu NoSQL, dữ liệu được lưu trữ dưới dạng collection và document. Do đó, cơ sở dữ liệu, collection và document có mối liên hệ với nhau như hình dưới đây:



Trong máy chủ MongoDB, bạn có thể tạo nhiều cơ sở dữ liệu và nhiều collection.

Cách cơ sở dữ liệu MongoDB chứa các collection cũng giống như cách cơ sở dữ liệu MySQL chứa các table.

Bên trong collection, chúng ta có document. Các document này chứa dữ liệu mà bạn muốn lưu trữ trong cơ sở dữ liệu MongoDB và một collection có thể chứa nhiều document. Đồng thời, với tính chất schema-less (không cần một cấu trúc lưu trữ dữ liệu), document này không nhất thiết phải giống với document khác.

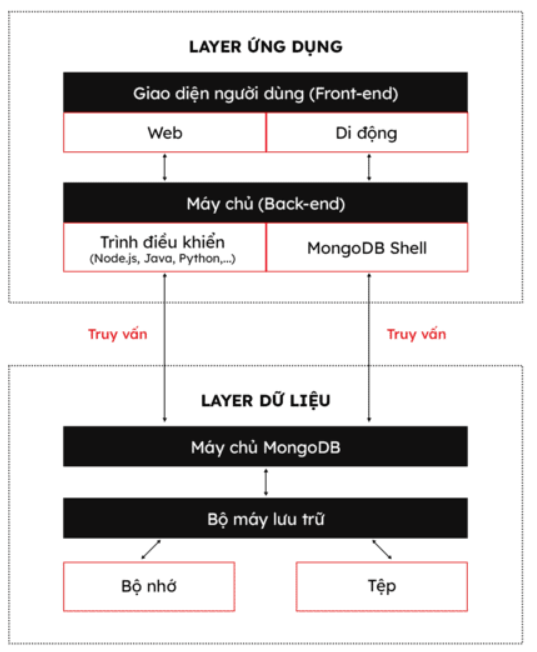
Các document được tạo bằng cách sử dụng các field (trường). Các field là các cặp khóa-giá trị trong document, giống như các column trong cơ sở dữ liệu quan hệ. Giá trị của các field có thể là bất kỳ loại dữ liệu BSON nào như double, string, boolean, v.v.

MongoDB lưu trữ dữ liệu ở định dạng BSON document. Ở đây, BSON là đại diện cho định dạng mã hoá nhị phân của các tài liệu JSON (chữ B trong BSON là viết tắt của Binary). Hay nói cách khác, trong phần backend, máy chủ MongoDB chuyển đổi dữ liệu JSON thành dạng nhị phân, được gọi là BSON, và BSON này có thể được lưu trữ và truy vấn hiệu quả hơn.

Kích thước tối đa của BSON document là 16 MB.

Trong MongoDB document, bạn được phép lưu trữ dữ liệu lồng nhau. Việc lồng dữ liệu này cho phép bạn tạo các mối quan hệ phức tạp giữa dữ liệu và lưu trữ chúng trong cùng một document, giúp cho quá trình làm việc và tìm nạp dữ liệu hiệu quả hơn so với SQL.

#### 1.5.5.2. MongoDB hoạt động như thế nào?



MongoDB hoạt động với hai layer:

* Layer Ứng dụng
* Layer Dữ liệu

Layer Ứng dụng còn được gọi là Layer Trừu tượng Cuối cùng (Final Abstraction Layer), gồm hai phần, đầu tiên là Front-end (Giao diện người dùng) và thứ hai là Back-end (máy chủ):

* Giao diện người dùng là nơi người dùng sử dụng MongoDB với sự trợ giúp của Web hoặc Di động. Web và thiết bị di động này bao gồm các trang web, ứng dụng di động, ứng dụng mặc định của Android, ứng dụng iOS, v.v.
* Phần back-end chứa một máy chủ được sử dụng để thực hiện logic phía máy chủ và cũng chứa trình điều khiển hoặc MongoDB shell để tương tác với máy chủ MongoDB với sự trợ giúp của truy vấn.

*MongoDB shell là một bảng điều khiển JavaScript tương tác với MongoDB mà bạn có thể sử dụng để kết nối với máy chủ cơ sở dữ liệu và thực thi các lệnh trên đó, cho phép bạn thực hiện các tác vụ quản trị và đọc, viết hoặc thao tác dữ liệu trực tiếp.*

Các truy vấn này được gửi đến máy chủ MongoDB thuộc Layer Dữ liệu. Bây giờ, máy chủ MongoDB nhận các truy vấn và chuyển các truy vấn đã nhận tới công cụ lưu trữ vì bản thân máy chủ MongoDB không trực tiếp đọc hoặc ghi dữ liệu vào tệp hoặc đĩa hoặc bộ nhớ.

Sau khi chuyển các truy vấn nhận được tới bộ máy lưu trữ, bộ máy lưu trữ chịu trách nhiệm đọc hoặc ghi dữ liệu trong tệp hoặc bộ nhớ.

### 1.5.6. Ưu điểm và khuyết điểm của MongoDB

#### 1.5.6.1. Ưu điểm MongoDB

MongoDB mang đến cho người sử dụng một số ưu điểm:

* Không schema: Giống như các cơ sở dữ liệu NoSQL khác, MongoDB không yêu cầu các schema được xác định trước.
* MongoDB lưu trữ bất kỳ loại dữ liệu nào: Điều này cho phép người dùng linh hoạt tạo số lượng trường trong document theo nhu cầu, và giúp việc mở rộng cơ sở dữ liệu MongoDB trở nên dễ dàng hơn so với cơ sở dữ liệu quan hệ truyền thống.
* Hướng document: Một trong những ưu điểm của việc sử dụng document là các đối tượng này ánh xạ tới các kiểu dữ liệu gốc trong một số ngôn ngữ lập trình. Việc có các document được nhúng cũng làm giảm nhu cầu kết nối cơ sở dữ liệu, điều này có thể làm giảm chi phí.
* Khả năng mở rộng: Kiến trúc mở rộng theo chiều ngang của MongoDB giúp bạn tạo ra một ứng dụng có thể xử lý được lưu lượng truy cập tăng đột biến khi doanh nghiệp của bạn phát triển. Ngoài ra, việc phân chia dữ liệu (sharding) cho phép cơ sở dữ liệu phân phối dữ liệu trên một cụm máy. MongoDB cũng hỗ trợ tạo vùng dữ liệu dựa trên shard key.
* Hỗ trợ bên thứ ba: MongoDB hỗ trợ một số công cụ lưu trữ và cung cấp API công cụ lưu trữ có thể cắm được (pluggable storage engine API) cho phép các bên thứ ba phát triển công cụ lưu trữ dữ liệu riêng.
* Linh hoạt lưu trữ tệp dung lượng lớn: MongoDB phát triển hệ thống tệp riêng GridFS, gần giống với hệ thống tệp phân tán Hadoop. Việc sử dụng hệ thống tệp nhằm để lưu trữ các tệp vượt qua kích thước giới hạn của BSON (16 MB cho mỗi document).

#### 1.5.6.2. Khuyết điểm MongoDB

Mặc dù MongoDB mang lại nhiều giá trị lớn, công cụ này vẫn có một số nhược điểm:

* Tính liên tục: Với chiến lược chuyển đổi dự phòng tự động, người dùng chỉ có thể thiết lập một node master trong cụm MongoDB. Nếu node master bị lỗi, một node khác sẽ tự động chuyển đổi thành master mới. Quá trình chuyển đổi này đảm bảo tính liên tục, nhưng không diễn ra tức thời mà có thể mất tới một phút.

Để bạn dễ so sánh, bạn có thể lấy cơ sở dữ liệu Cassandra NoSQL hỗ trợ nhiều node master làm ví dụ. Với Cassandra NoSQL, nếu một master ngừng hoạt động, một master khác sẽ sẵn sàng, tạo ra cơ sở hạ tầng cơ sở dữ liệu có tính sẵn sàng cao.

* Giới hạn ghi: Node master duy nhất của MongoDB cũng làm giới hạn lại tốc độ ghi dữ liệu vào cơ sở dữ liệu. Việc ghi dữ liệu phải được ghi trên node master và việc ghi thông tin mới vào cơ sở dữ liệu bị giới hạn bởi khả năng của node master đó.
* Tính nhất quán của dữ liệu: MongoDB không cung cấp tính toàn vẹn tham chiếu đầy đủ thông qua việc sử dụng các ràng buộc khóa ngoại (foreign-key), điều này có thể ảnh hưởng đến tính nhất quán của dữ liệu.
* Bảo mật: Tính năng xác thực người dùng không được mặc định bật trong cơ sở dữ liệu MongoDB. Để bảo mật hệ thống trước các cuộc tấn công của tin tặc, bạn có thể thủ công thiết lập các cài đặt chặn những kết nối lạ và không an toàn.

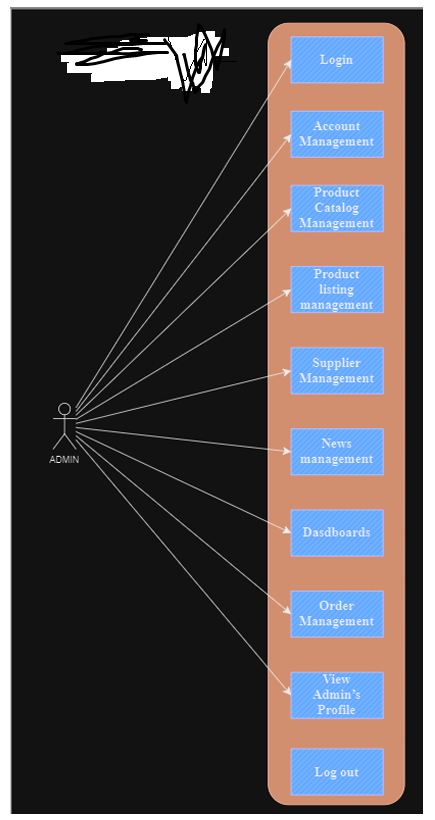
**1.5.7. So sánh MongoDB và MySQL**

Hãy so sánh trực tiếp những khái niệm cơ bản nhất của hai cơ sở dữ liệu phổ biến này:

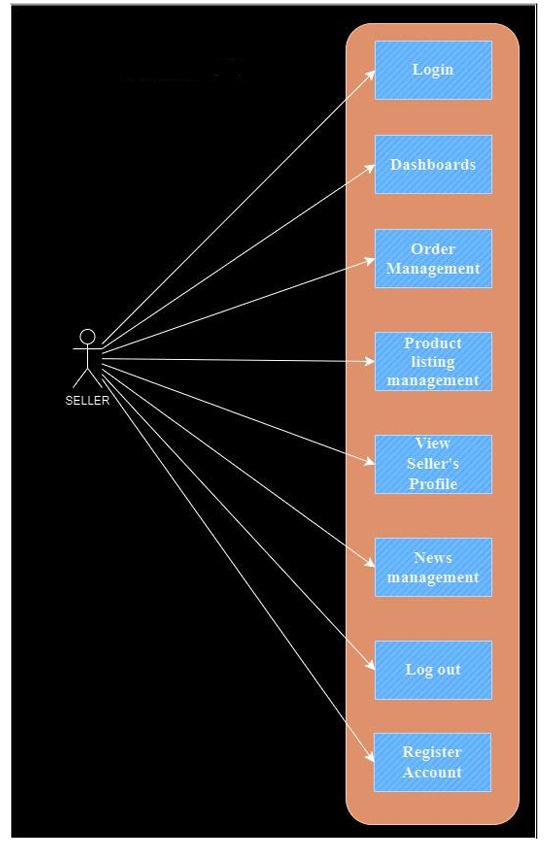
|  |  |  |
| --- | --- | --- |
|  | MongoDB | MySQL |
| Ưu tiên | Thân thiện với cloud | Mức độ bảo mật dữ liệu cao |
| Cấu trúc dữ liệu | Không cấu trúc, hoặc cấu trúc dữ liệu có tiềm năng phát triển nhanh | Có cấu trúc |
| Đại diện dữ liệu | JSON document | Table và row |
| Hỗ trợ JOIN | Không | Có |
| Ngôn ngữ truy vấn | JavaScript | SQL |
| Schema | Không cần schema | Cần xác định column và table |
| Hiệu suất phát triển | Nhanh | Chậm |
| Tính nguyên tử của transaction | Không hỗ trợ đầy đủ tất cả các hoạt động nhưng hỗ trợ các transaction đa document | Hỗ trợ tính nguyên tử của transaction |

# PHẦN 2: WEBSITE BÁN THIẾT BỊ MÁY TÍNH BẰNG NODEJS VÀ MONGODB

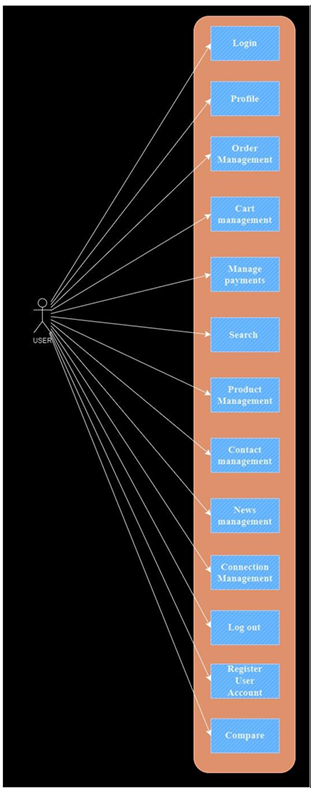
## 1. Usecase diagram Admin LV0



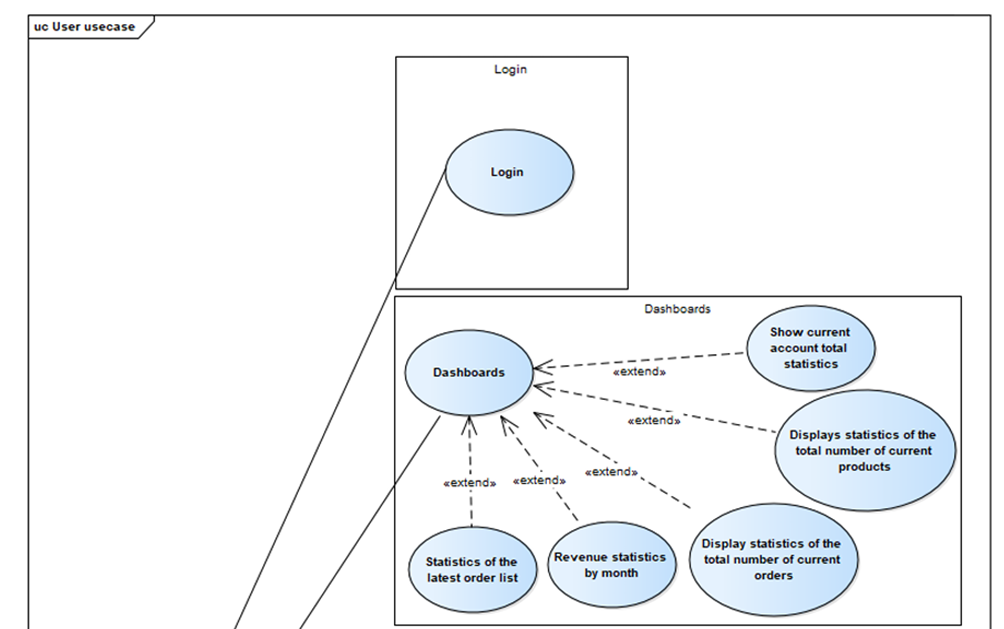
# 2. Usecase diagram Seller LV0

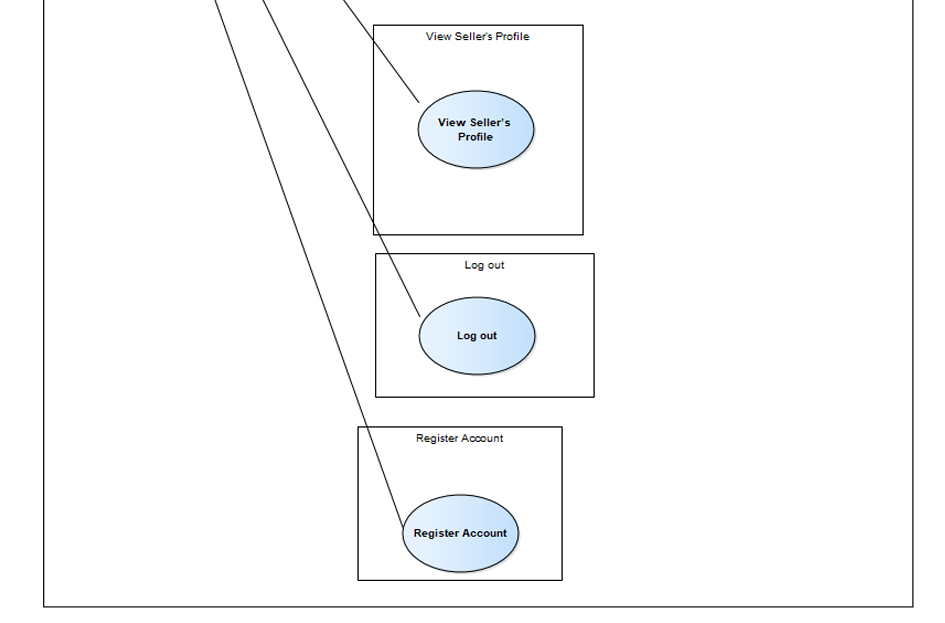
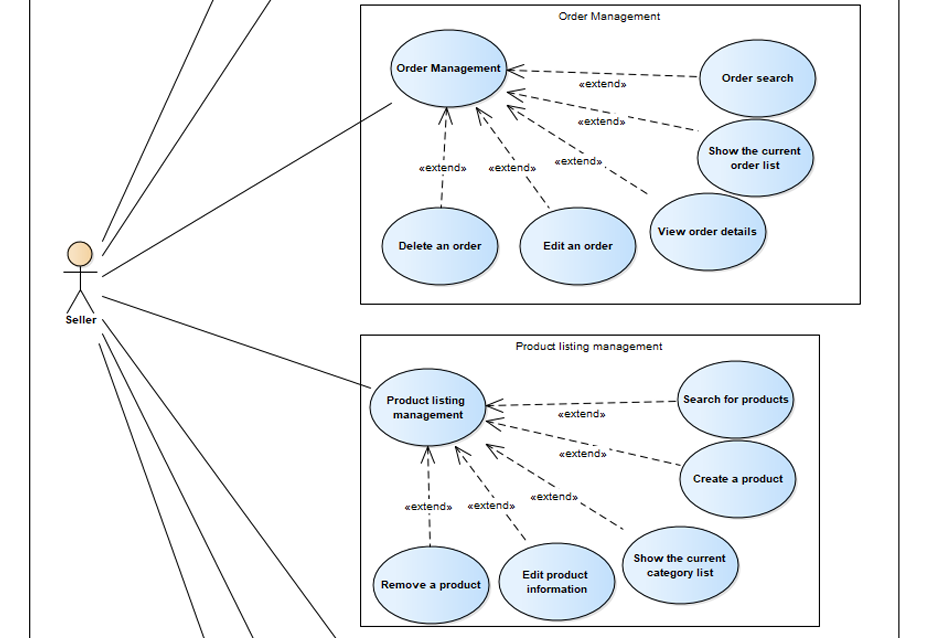


## 3. Usecase diagram User LV0

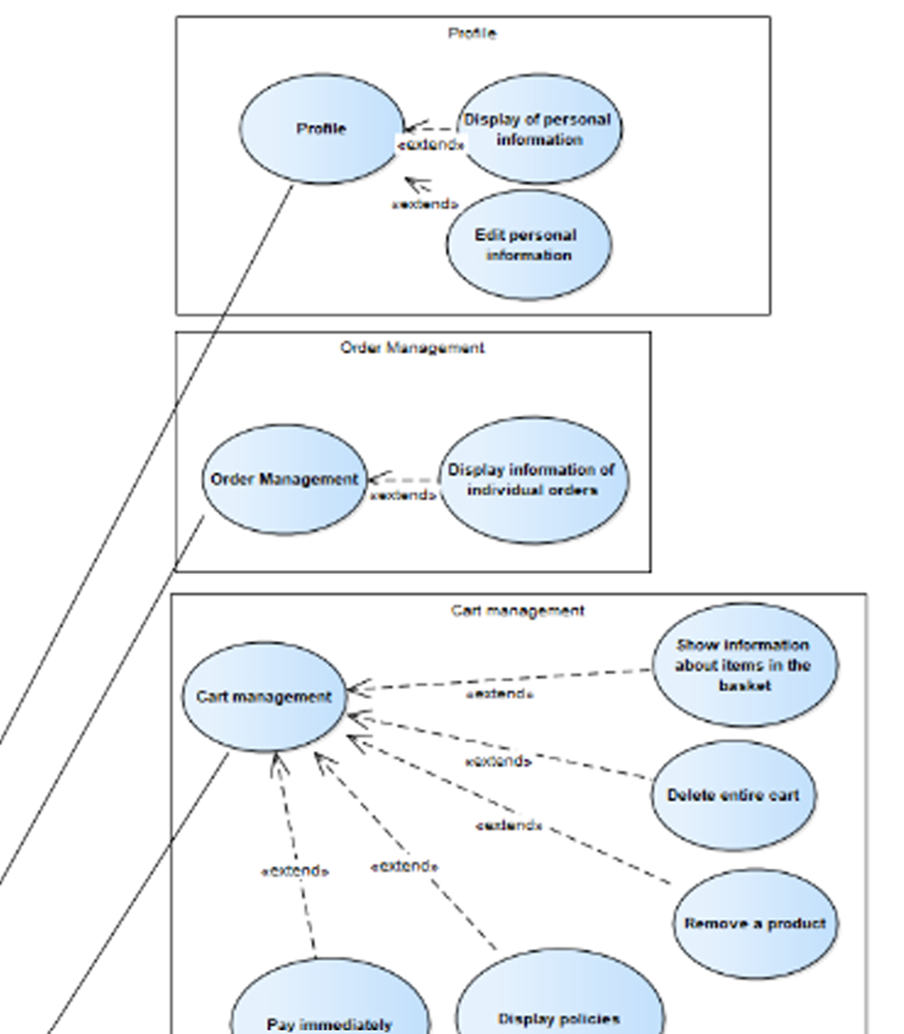


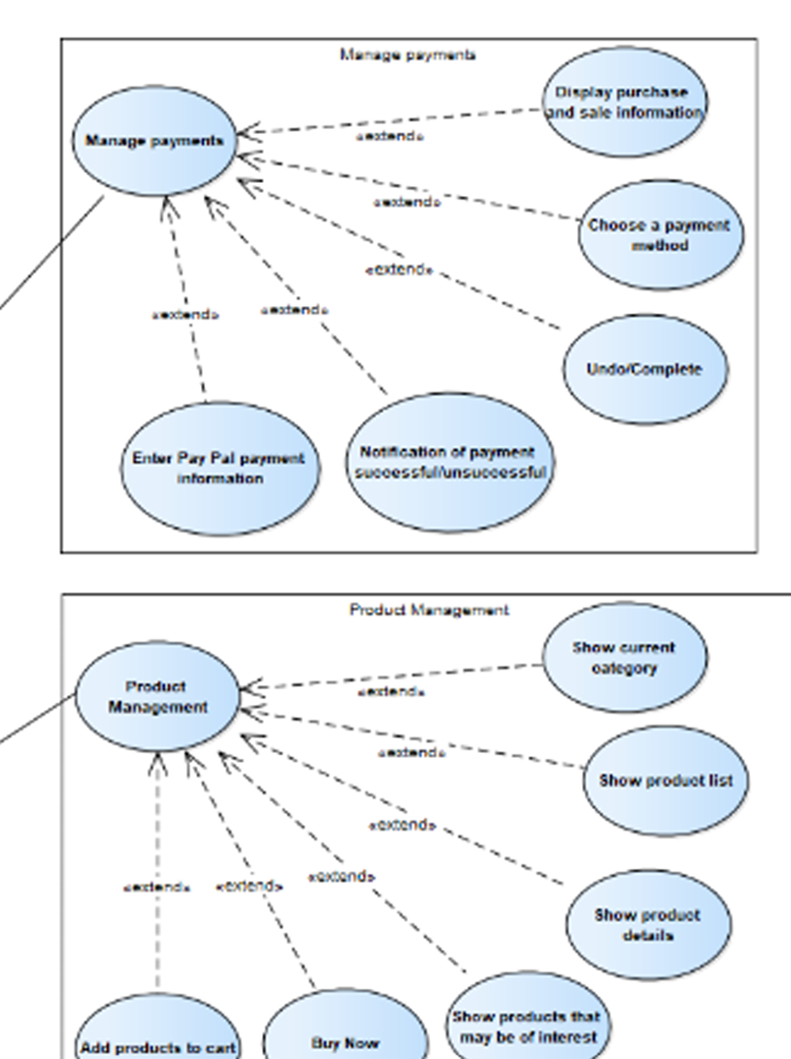
## 4. Usecase diagram Admin LV1

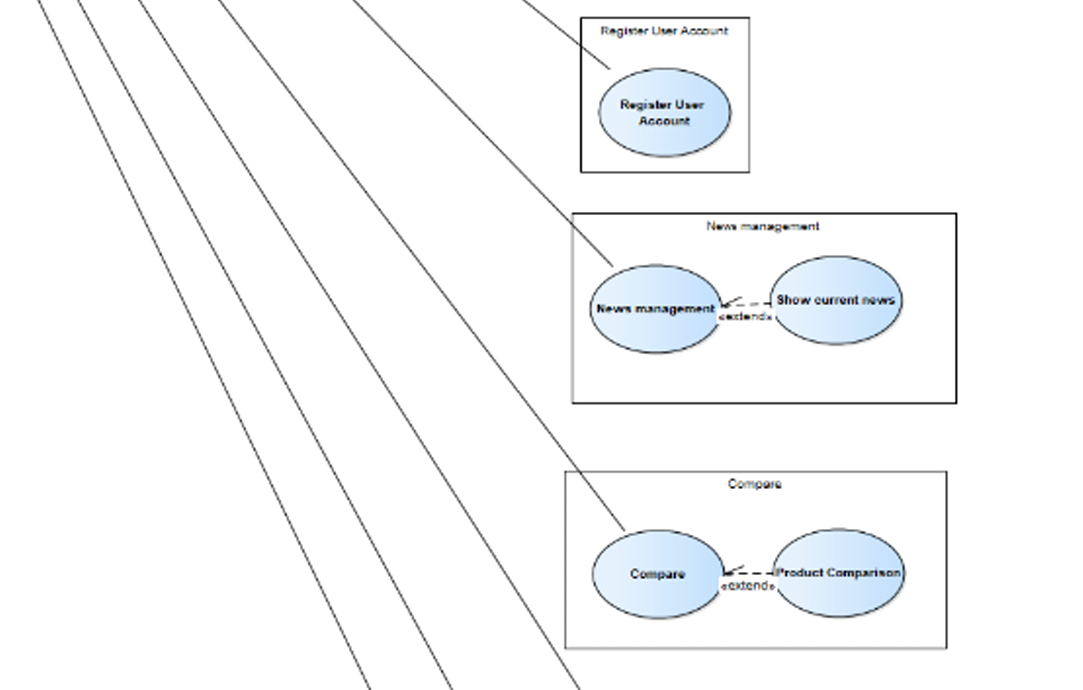
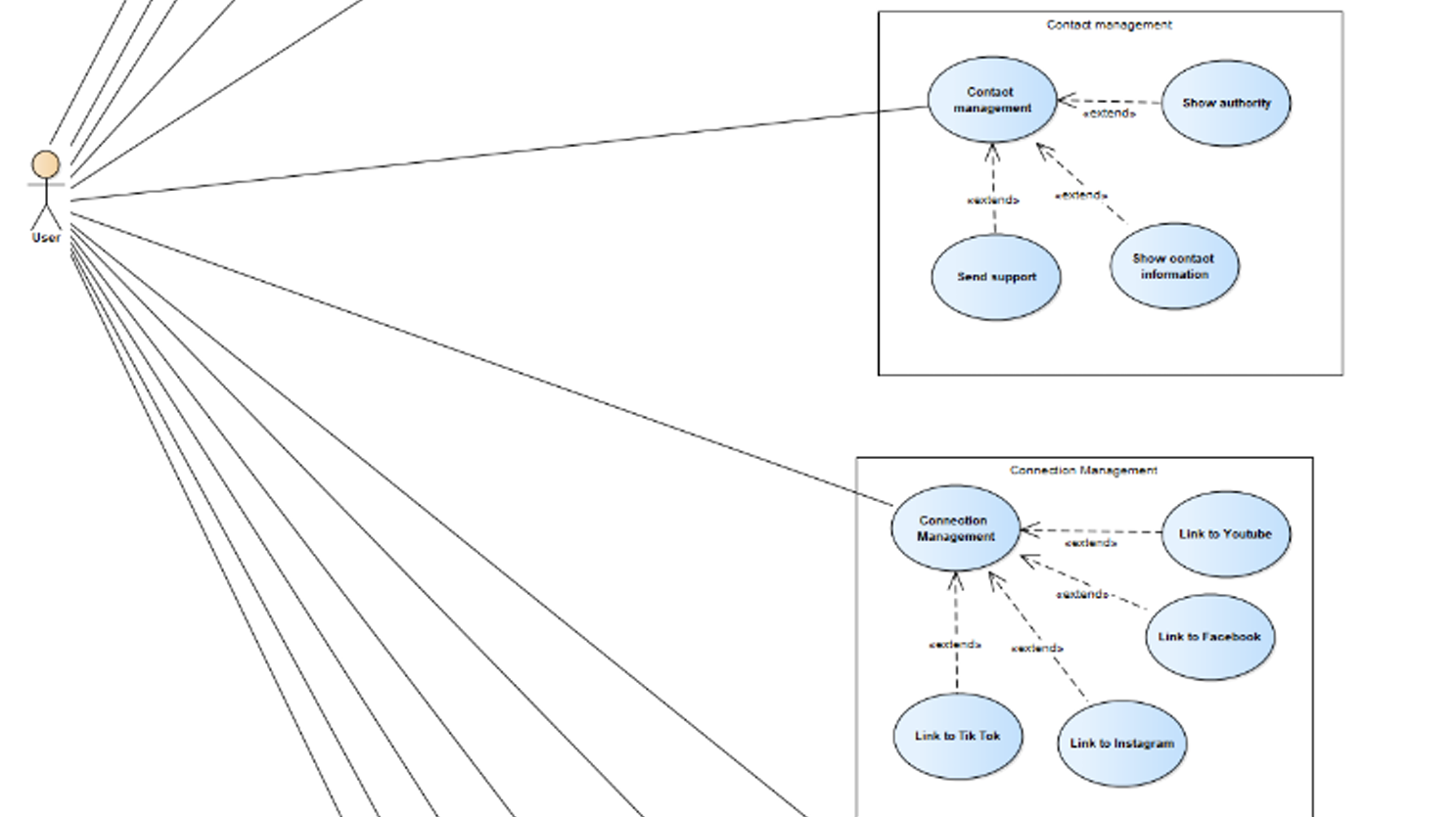
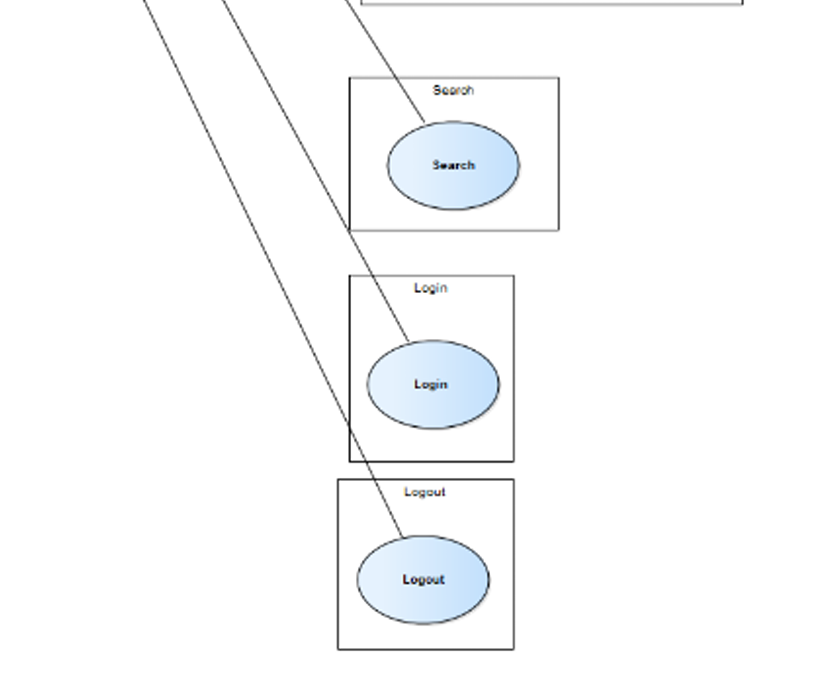




## 5. Usecase diagram User LV1







## 6. Usecase Description Admin

### 6.1. Use case 1.0 + 1.2: (UC1.0 + 1.2- Search for an account and Show a list of current accounts)

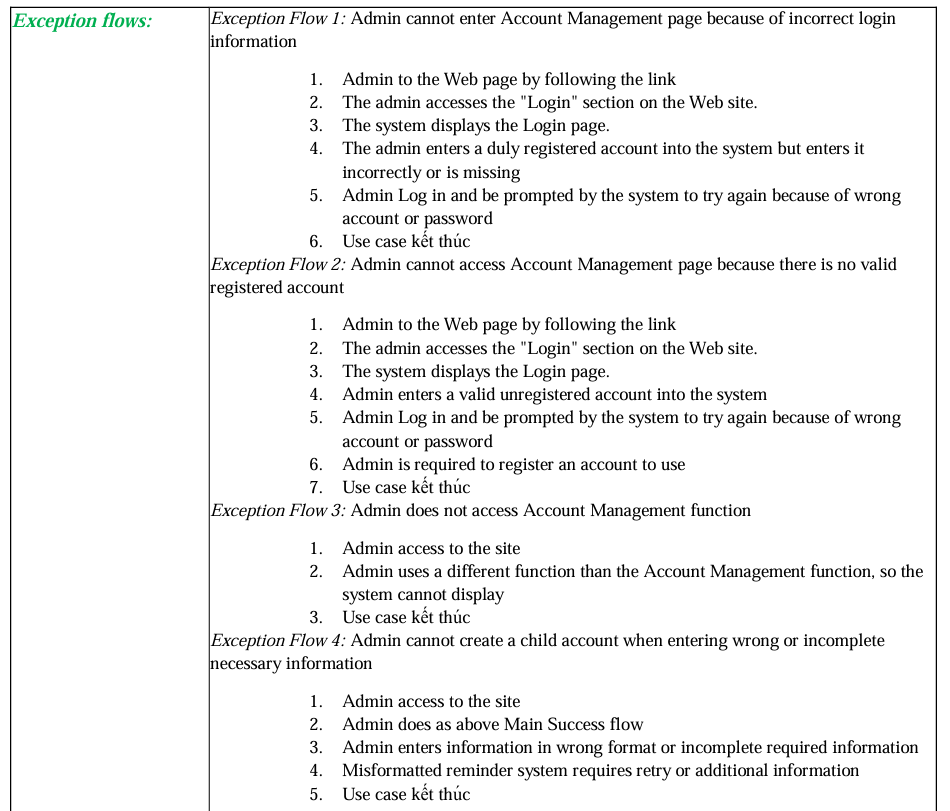
|  |  |
| --- | --- |
| Use Case ID: | UC1.0+1.2 |
| Use case Name: | Search for an account and Show a list of current accounts |
| Creator: | Person updated: |
| Creation date: | 16/01/2024 Date updated: 24/01/2024 |
| Primary actor: | Admin |
| Brief description: | Admin can manage and control the current list of accounts on the system |
| Pre-conditions: | Admin needs a General Manager account |
|  | Admin needs to access the Admin Management page |
|  | Admin access to Account Management page |
| Post-conditions: | Search for accounts on demand Allows the display of the list of current accounts and search accounts on demand |
| Main Success flow: | 1 Admin to the Web page by following the link 2. The admin accesses the Login section on the Web site. |
|  | 3. The system displays the Login page. |
|  | 4 The admin enters the duly registered account into the system and logs in |
|  | 5 Successful login notification system |
|  | 6. The system displays Dashboard page information ( Number of members Number of products, Number of categories, Order number Revenue by month Last order) |
|  | 7. Admin access to Account Management |
|  | 8. The system displays the list of accounts, the search bar and related functions |
|  | 9. Use case ends |

|  |  |
| --- | --- |
| Exception flows: | Exception Flow 1 Admin cannot enter Account Management page because of incorrect login |
|  | information |
|  | 1 Admin to the Web page by following the link |
|  | 2. The admin accesses the Login section on the Web site. |
|  | 3. The system displays the Login page. 4. The admin enters a duly registered account into the system but enters it incorrectly or is missing 5. Admin Log in and be prompted by the system to try again because of WtOng account or password 6. Use case kết thúc |
|  | Exception Flow 2: Admin cannot access Account Management page because there is no valid registered account |
|  | 1 Admin to the Web page by following the link 2. The admin accesses the Login section on the Web site. 3. The system displays the Login page |
|  | 4. Admin enters a valid unregistered account into the system |
|  | 5. Admin Log in and be prompted by the system to try again because of Wrong |
|  | account or password 6. Admin is required to register an account to use 7. Use case kết thúc |
|  | Exception Flow 3: Admin does not access Account Management function |
|  | 1 Admin access to the site - 2. Admin uses a different function than the Account Management function, so the system cannot display 3. Use case kết thúc |
| Trigger: | The user opens the Login interface, then enters the Dashboards interface, Account Management |
| Priority: | Tall |

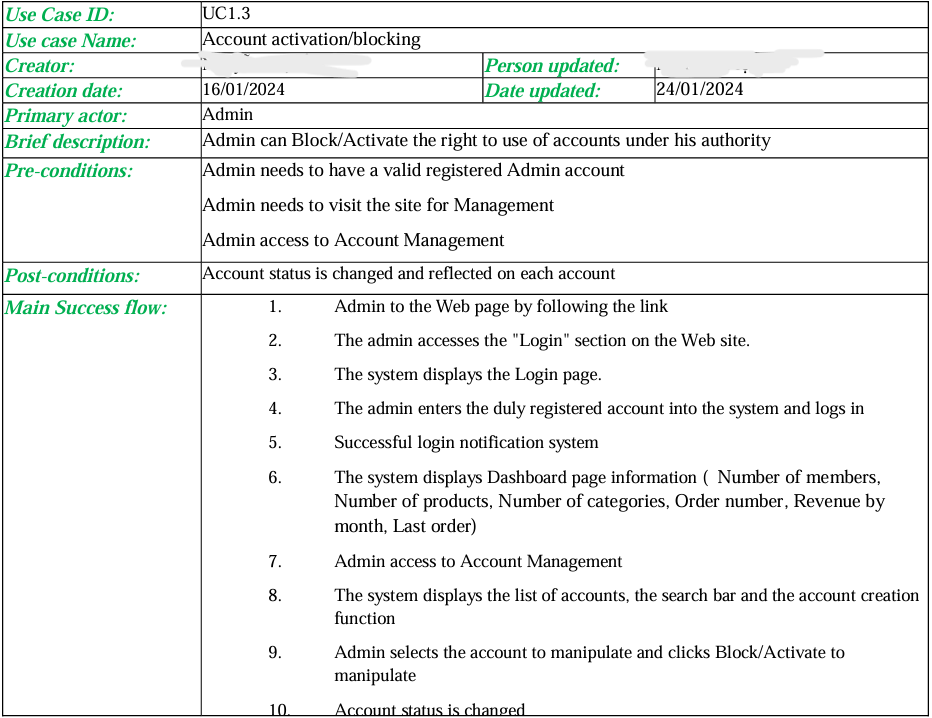


### 6.2. Use case 1.1: (UC1.1-Create a child admin account)

|  |  |  |
| --- | --- | --- |
| Use Case ID: | UC1.1 | |
| Use case Name: | Create a child admin account | |
| Creator: | Person updated:  5 | |
| Creation date: | 16/01/2024  Date updated:  24/01/2024 | |
| Primary actor: |  | |
| Brief description: |  | Admins can create child admin accounts |
| Pre-conditions: |  | Admins need to have a registered admin account  Admin needs to visit the site for Management  Admin access to Account Management |
| Post-conditions: |  | Child Admin account successfully created |
| Main Success How: |  | 1. Admin to the Web page by following the link  2. The admin accesses the "Login" section on the Web site.  3. The system displays the Login page.  4. The admin enters the duly registered account into the system and logs in  5. Successful login notification system  6. The system displays Dashboard page information ( Number of members,  Number of products, Number of categories, Order number, Revenue by  month, Last order)  7. Admin access to Account Management  8. The system displays the list of accounts, the search bar and the account creation  function  9. Admin creates a child Admin account by entering all necessary and valid  information |

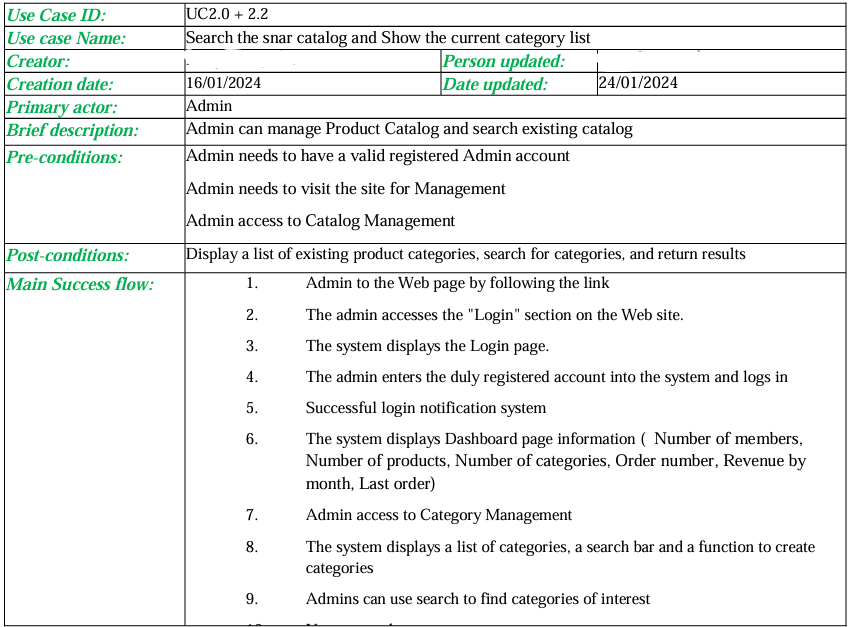


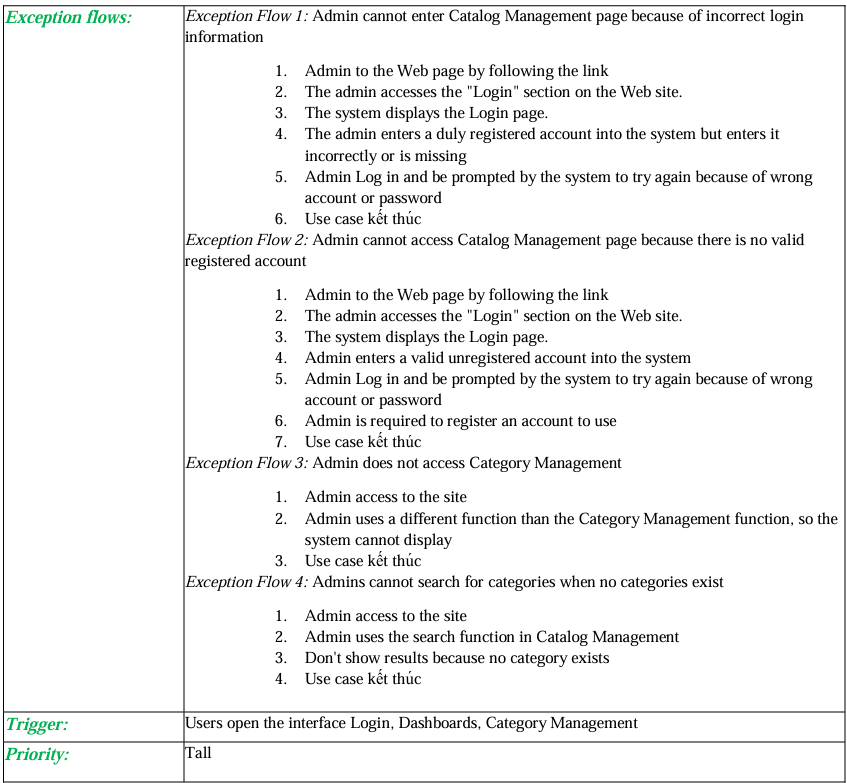
### 6.3. Use case 1.3: (UC1.3-Account activation/blocking)



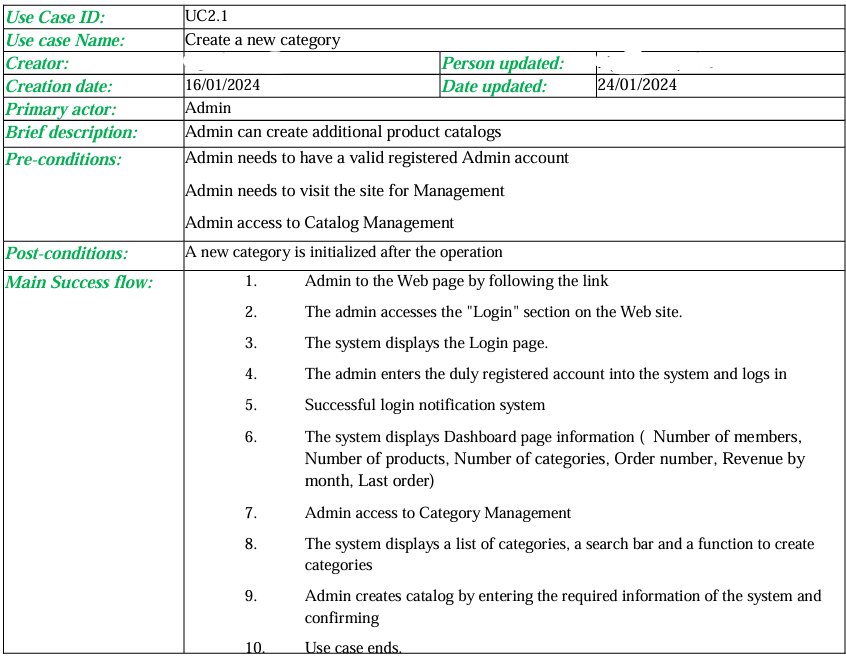


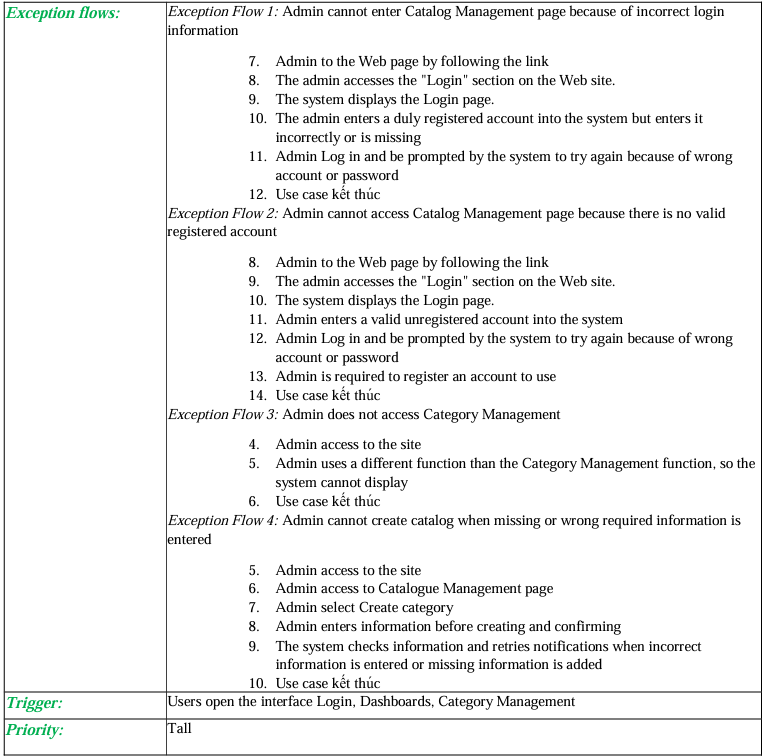
### 6.4. Use case 2.0 + 2.2: (UC2.0 + 2.2-Search the snar catalog and Show the current category list)



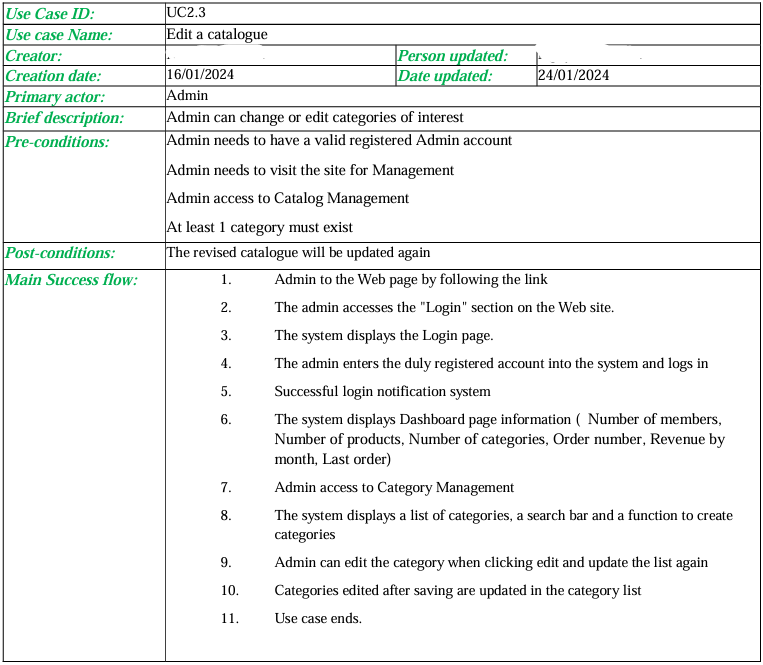


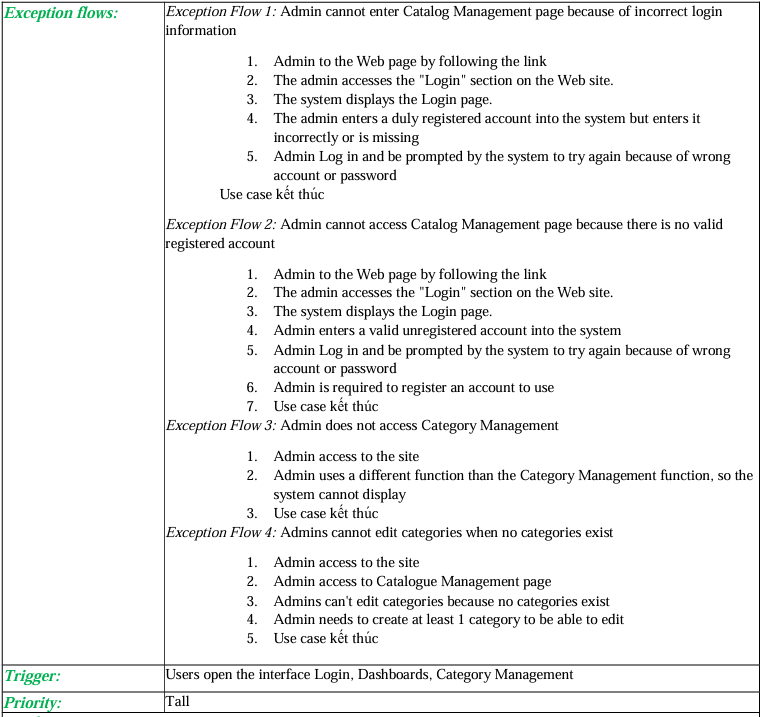
### 6.5. Use case 2.1: (UC2.1-Create a new category)



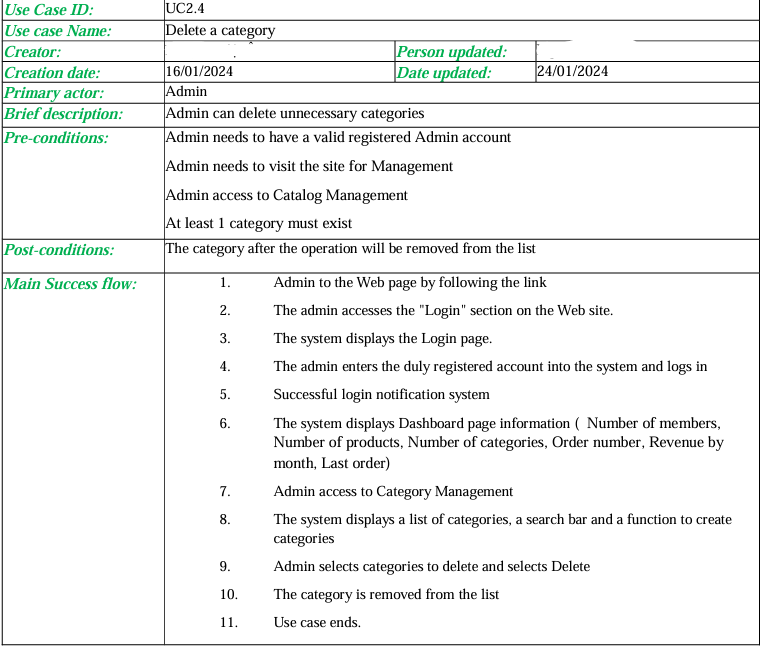


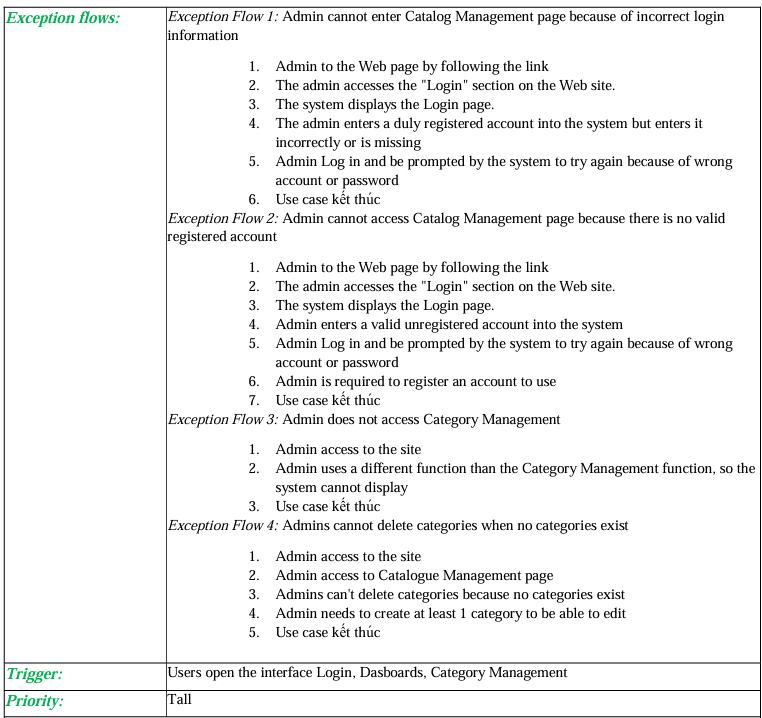
### 6.6. Use case 2.3: (UC2.3-Edit a catalogue)



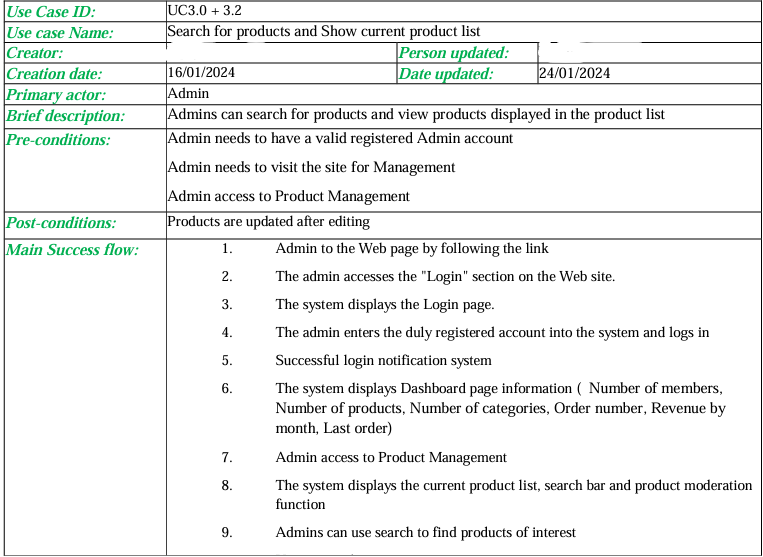


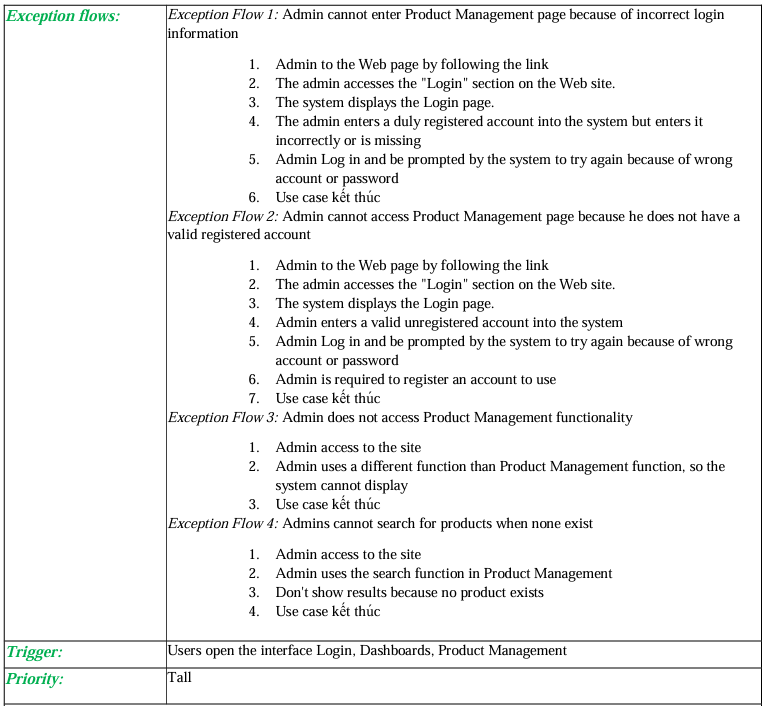
### 6.7. Use case 2.4: (UC2.4- Delete a category)



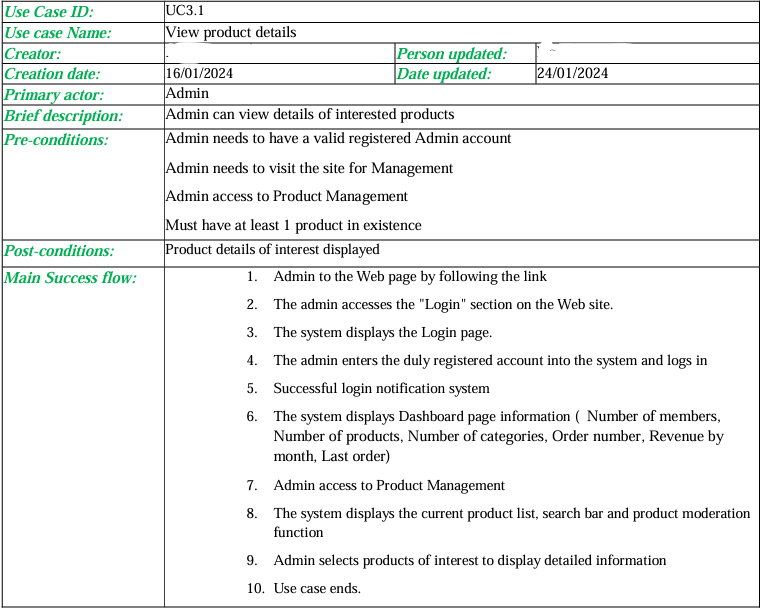


### 6.8. Use case 3.0 + 3.2: (UC3.0 + 3.2-Search for products and Show current product list)



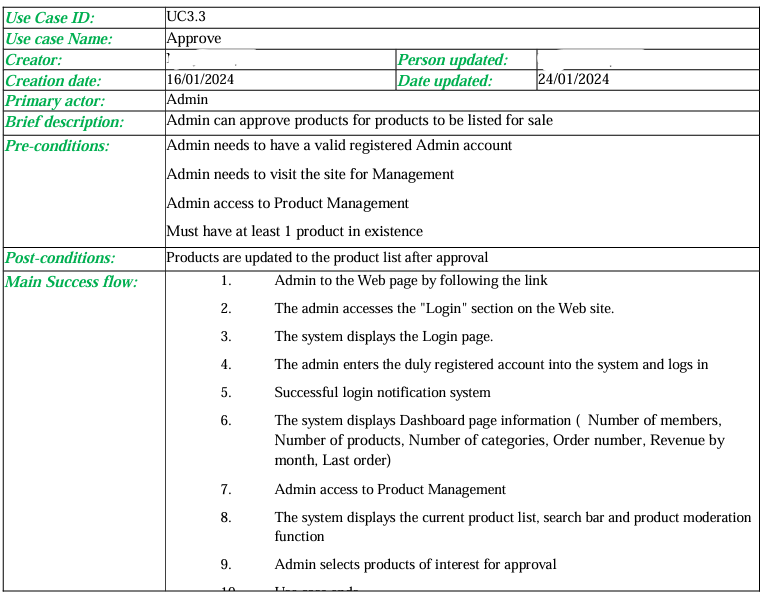


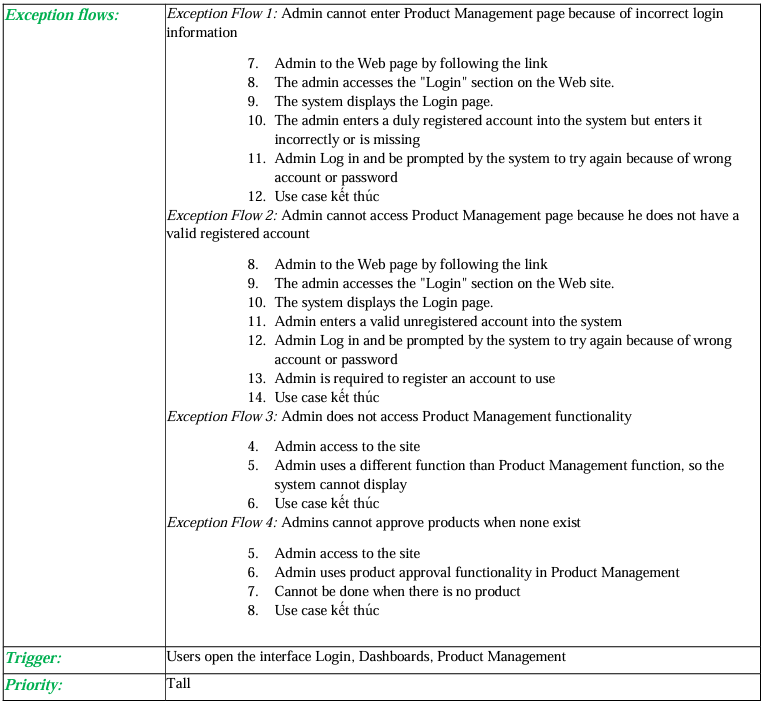
### 6.9. Use case 3.1: (UC3.1-View product details)



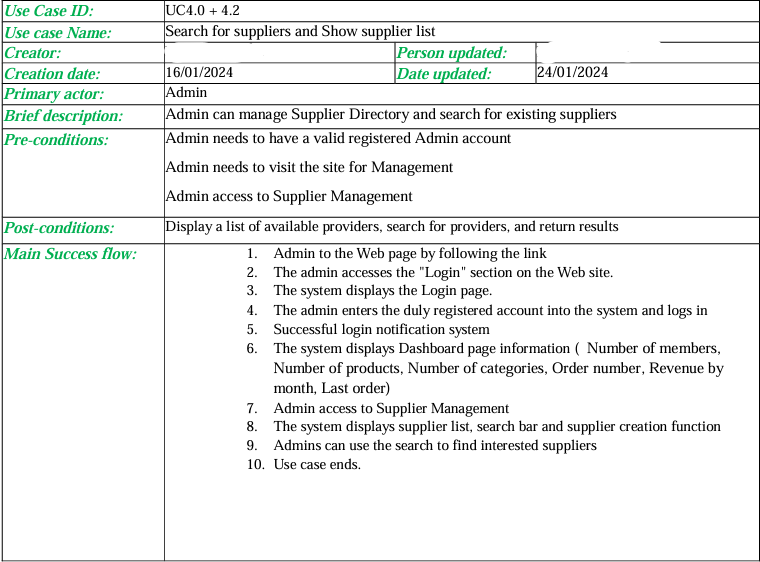


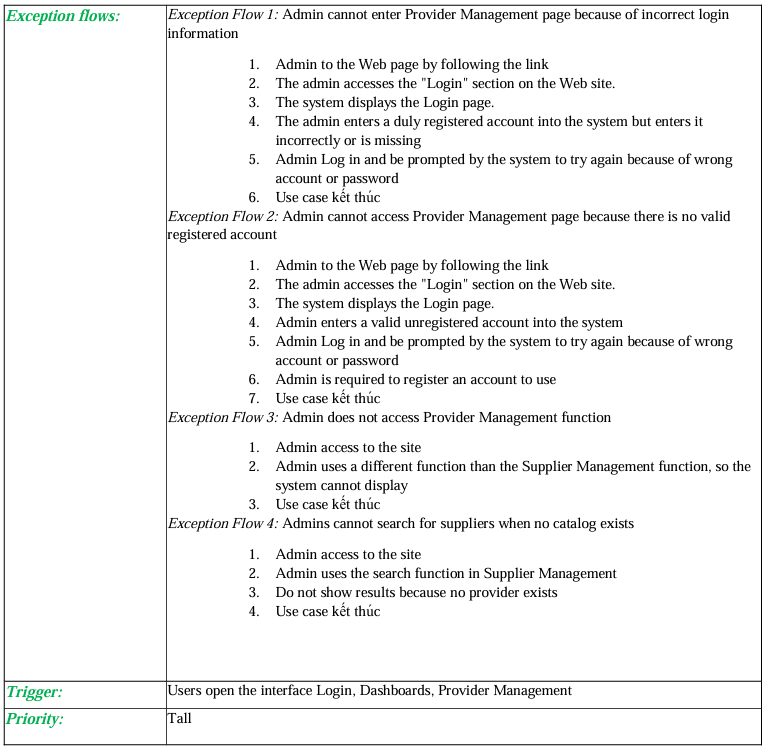
### 6.10. Use case 3.3: (UC3.3- Approve)



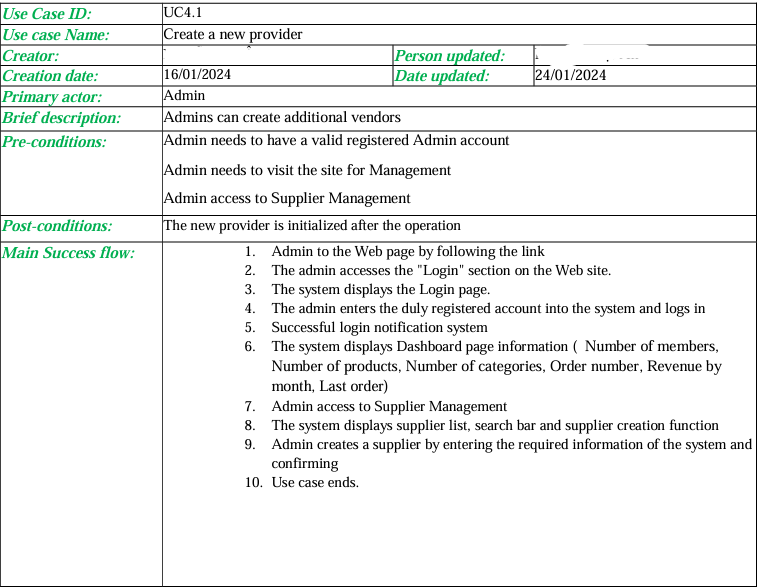


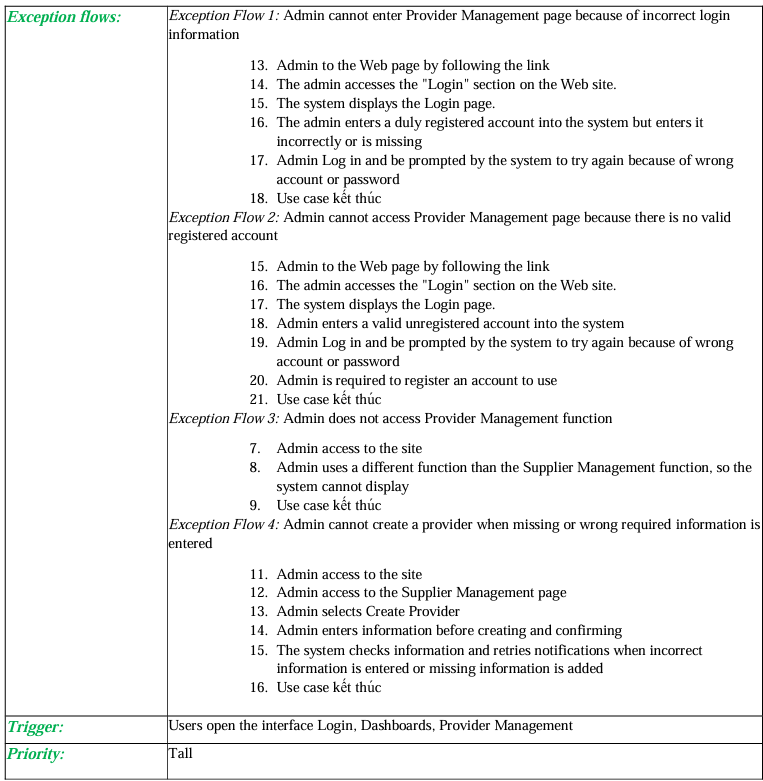
### 6.11. Use case 4.0 + 4.2: (UC4.0 + 4.2- Search for suppliers and Show supplier list)





### 6.12. Use case 4.1: (UC4.1-Create a new provider)





# Tài liệu tham khảo

Nodejs.com

Laptrinhvien.com

Huongdancaidatvasudungnodejs

Mongodb.com

Huongdansudungmongodb.com

Xaydungwebsite.com