**ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**TRƯỜNG CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG**

### **A close up of a sign Description automatically generated**

**ĐỒ ÁN TỐT NGHIỆP**

**Xây dựng ứng dụng web cho Shopify Store**

**PHẠM LÊ MINH TRÍ**

tri.plm173411@sis.hust.edu.vn

**Ngành Công nghệ Thông tin và Truyền thông**

**Chuyên ngành Khoa học máy tính**

| **Giảng viên hướng dẫn:** | ThS. Nguyễn Tiến Thành |
| --- | --- |
| **Bộ môn:** | Công nghệ phần mềm |
| **Viện:** | Công nghệ Thông tin và Truyền thông |
| **HÀ NỘI, 2022** | |

**Lời cam kết**

Họ và tên sinh viên: Phạm Lê Minh Trí

Điện thoại liên lạc: 0334930853 Email: tri.plm173411@sis.hust.edu.vn

Lớp: KHMT.01.K62 Hệ đào tạo: Đại học đại trà

Tôi – *Phạm Lê Minh Trí* – cam kết Đồ án Tốt nghiệp (ĐATN) là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *ThS. Nguyễn Tiến Thành*. Các kết quả nêu trong ĐATN là trung thực, là thành quả của riêng tôi, không sao chép theo bất kỳ công trình nào khác. Tất cả những tham khảo trong ĐATN – bao gồm hình ảnh, bảng biểu, số liệu, và các câu từ trích dẫn – đều được ghi rõ ràng và đầy đủ nguồn gốc trong danh mục tài liệu tham khảo. Tôi xin hoàn toàn chịu trách nhiệm với dù chỉ một sao chép vi phạm quy chế của nhà trường.

|  | *Hà Nội, ngày tháng năm*  Tác giả ĐATN  *Họ và tên sinh viên* |
| --- | --- |

**Lời cảm ơn**

Lời đầu tiên, em xin gửi lời cảm ơn sâu sắc đến ThS. Nguyễn Tiến Thành. Thầy là người đã hướng dẫn, giúp đỡ em rất nhiều trong quá trình nghiên cứu, thực hiện và hoàn thiện ĐATN này.

Em cũng xin cảm ơn tới tất cả thầy cô đã chỉ dạy cho em những kiến thức quý báu trong suốt quá trình học tập, cung cấp một hành trang thiết thực cho tương lai sắp tới.

Cuối cùng, em xin cảm ơn gia đình, bạn bè, đồng nghiệp - những người luôn ở bên cạnh, cổ vũ và động viên tinh thần, giúp em vượt qua những khó khăn trong suốt quá trình học tập tại Đại Học Bách Khoa Hà Nội.

Do thời gian có hạn, cũng như kiến thức chuyên môn còn hạn hẹp, ĐATN của em sẽ không tránh khỏi những thiết sót và hạn chế. Em rất mong nhận được những đánh giá, nhận xét từ thầy cô và các bạn, để từ đó nhận thấy những ưu, nhược điểm của bản thân, đúc kết và xây dựng kinh nghiệm làm việc một cách vững vàng hơn.

Em xin chân thành cảm ơn!

# **TÓM TẮT NỘI DUNG ĐỒ ÁN**

Nội dung của báo cáo đồ án tốt nghiệp được chia thành ba chương chính, với nội dung được tóm tắt như sau:

**Chương 1: Khảo sát, đặc tả yêu cầu bài toán**

Ở chương này, các yêu cầu ban đầu của bài toán được nêu lên, bao gồm khảo sát hiện trạng thực tế, định hướng giải pháp, các chức năng chính, yêu cầu phi chức năng của ứng dụng xây dựng.

**Chương 2: Phân tích, thiết kế bài toán**

Với những yêu cầu đã đặt ra ở chương 1, chương 2 sẽ làm rõ các bước phân tích và thiết kế kiến trúc ứng dụng, lựa chọn công nghệ cũng như nghiên cứu các giải pháp để giải quyết những yêu cầu đặt ra.

**Chương 3: Kết quả đạt được**

Sau quá trình phân tích và thiết kế, ứng dụng sẽ được triển khai xây dựng. Chương 3 nêu lên kết quả đạt được, quá trình kiểm thử cũng như triển khai ứng dụng thực tế.

Cuối cùng, phần kết luận sẽ tổng kết lại quá trình xây dựng ứng dụng, những ưu, nhược điểm đã đạt được cũng như định hướng phát triển mở rộng trong tương lai.

# **DANH MỤC HÌNH VẼ**

# **DANH MỤC BẢNG**

# **DANH MỤC CÁC THUẬT NGỮ VÀ TỪ VIẾT TẮT**

|  |  |
| --- | --- |
| **Front-end** | Hiển thị dữ liệu ứng dụng |
| **Back-end** | Xử lý các logic, dữ liệu của ứng dụng |
| **Platform** | Nền tảng tạo dựng môi trường thực thi ứng dụng |
| **Framework** | Bộ khung các thư viện được đóng gói |
| **TMĐT** | Thương mại điện tử |
| **CDN** | Content Delivery Network |
| **SEO** | Search Engine Optimization (tối ưu hóa công cụ tìm kiếm) |
| **AWS** | Amazon Web Services |
| **SSR** | Server Side Rendering |
| **CSR** | Client Side Rendering |
|  |  |
|  |  |

# **MỤC LỤC**

# **CHƯƠNG 1. KHẢO SÁT, ĐẶC TẢ YÊU CẦU BÀI TOÁN**

Trong quá trình phát triển và xây dựng phần mềm, bước khởi đầu sẽ là xác định được yêu cầu của khách hàng. Ngoài ra, sự hợp tác và khảo sát thị trường là cần thiết để xác định khách hàng mong muốn sản phẩm như thế nào. Vì vậy quá trình này sẽ là nền móng để các bước tiếp theo phát triển. Chương 1 sẽ làm rõ các yêu cầu đặt ra cũng như các tính năng sẽ xây dựng trong ứng dụng.

## **1.1 Đặt vấn đề**

Trong những năm gần đây, thị trường thương mại điện tử (TMĐT) ngày càng được mở rộng và hiện đã trở thành phương thức kinh doanh phổ biến được doanh nghiệp, cá nhân biết đến. Mặc dù gặp những ảnh hưởng trong những năm gần đây do đại dịch COVID-19, thương mại điện tử vẫn có những bước tăng tốc vô cùng mạnh mẽ. Cùng với những bước phát triển đó, hàng loạt những e-commerce platform nổi lên và thống lĩnh thị trường, nổi bật có thể kể đến như Woocommerce, Magento và đặc biệt là Shopify. Vào thời điểm 8/2021, có hơn 1.700.000 chủ cửa hàng đang sử dụng nền tảng này, biến Shopify trở thành nền tảng thương mại điện tử số 1 thế giới.

Không giống với những nền tảng như Magento hay OpenCart đòi hỏi người sử dụng cần phải có kiến thức về website, hosting hay HTML mới sử dụng được; Shopify được thiết kế dành cho tất cả mọi người, với một người bình thường không cần bất cứ kĩ năng nào về lập trình để bắt đầu với Shopify. Cụ thể hơn Shopify có hàng loạt những ưu điểm như sau:

* Dễ sử dụng
* Giao diện đẹp mắt với kho giao diện hàng trăm themes
* Giải pháp thương mại điện tử trọn gói
* Công nghệ lưu trữ đám mây Cloud Services
* Hiệu suất cao với công nghệ CDN
* Bán hàng đa kênh (multi-channel)
* Tích hợp quảng cáo Facebook, Google
* Thiết kế chuẩn SEO và Mobile Friendly
* App Store khổng lồ với hơn 5000 ứng dụng
* Hỗ trợ trực tuyến 24/7

Để đáp ứng đa dạng nhu cầu khách hàng, Shopify còn cho phép các chủ store có thể tích hợp các shopify app vào store của mình, từ đó mở rộng thêm các tính năng mà store cung cấp, nâng cao trải nghiệm người dùng. Shopify cũng cho phép các nhà phát triển có thể tương tác với Database của Shopify thông qua hệ thống API mà Shopify cung cấp, từ đó cho ra đời các ứng dụng với nhiều tính năng khác nhau, có thể tích hợp vào store của khách hàng.

Mặc dù với số lượng ứng dụng tương đối lớn trên Shopify Store, nhưng nhu cầu về việc mở rộng tính năng cho store chưa bao giờ dừng lại. Điều này yêu cầu những nhà phát triển liên tục cho ra đời những ứng dụng để đáp ứng được lượng nhu cầu đó**.**

## **1.2 Khảo sát hiện trạng**

Để tăng trưởng và đáp ứng được nhu cầu khách hàng, các chủ sẽ cần thêm rất nhiều tính năng cho website của mình, Shopify App Store với hơn 5000 ứng dụng (miễn phí và trả phí) có thể đáp ứng mọi nhu cầu của họ.

Những tính năng phổ biến như livechat, đánh giá sản phẩm, email marketing, mua hàng tích điểm, thiết kế landing page, upsell & cross-sell, khôi phục giỏ hàng chưa hoàn tất,... cùng vô vàn tính năng khác.

Tuy nhiên số lượng các app và các tính năng phổ biến trên vẫn chưa thể đáp ứng hết được nhu cầu của mở rộng tính năng của khách hàng. Yêu cầu cần có thêm các ứng dụng để đáp ứng được nhu cầu đó. Trong đó, vấn đề mua/giao hàng định kỳ cùng các ưu đãi khi mua hàng, tích hợp tính năng quản lý và thống kê doanh số cũng là một vấn đề lớn cần được giải quyết.

Kể từ tháng 10/2020, Shopify cho ra mắt Subscription APIs, thông qua đó các nhà phát triển có thể xây dựng các subscription apps, giải quyết được vấn đề nói trên.

## **1.3 Đề xuất giải pháp**

Nắm bắt được nhu cầu khổng lồ về việc mở rộng các tính năng cho website cả khách hàng, cùng với nhận thấy vấn đề cần giải quyết đã nêu ở trên, ĐATN của em hướng tới giải quyết được một phần nào đó những yêu cầu này, xây dựng một ứng dụng một ứng dụng cho Shopify Store, cụ thể là ứng dụng **Subscription & Recurring Pay**.

Ứng dụng cho phép thanh toán một lần, giao hàng định kỳ, khuyến mãi khi mua hàng, tích hợp tính năng quản lý và thống kê doanh thu, email notifications … cùng rất nhiều những tính năng khác sẽ trình bày ở các phần sau. Ứng dụng này sẽ có thể đáp ứng và giải quyết phần lớn yêu cầu mà khách hàng cần ở một subscription apps. Ứng dụng hiện tại đã chính thức đưa lên Shopify Store và có lượng người dùng thực tế.

Mã nguồn của ứng dụng sẽ chia làm 5 phần:

* Subscription CMS ( Frontend ) : Hiển thị dữ liệu, tương tác với người dùng ( chủ store)
* Subscription API ( Backend ) : xử lý các nghiệp vụ logic, kết nối với dữ liệu để thực hiện các tính năng
* Subscription Proxy : hiển thị dữ liệu, tương tác với người dùng ( khách mua hàng )
* Automation for billing & renewing subscriptions : Xử lý nghiệp vụ, logic tự động gia hạn đăng ký
* Email service : Xử lý nghiệp vụ, logic để gửi email notifications thông qua AWS

Phần backend sẽ được xây dựng trên platform NodeJs, framework KoaJS, sử dụng ngôn ngữ lập trình JavaScript. Với ưu điểm nhanh hơn và nhẹ hơn của JavaScript so với các ngôn ngữ lập trình khác, ứng dụng xây dựng sẽ đảm bảo một hiệu năng tốt nhất cho người dùng. Cùng với đó là việc sử dụng Shopify API và Shopify GraphQL mà Shopify cung cấp để tương tác với Database của Shopify.

Phần frontend sử dụng framework NextJS với cơ chế Server Side Rendering (SSR) mang lại hiệu năng và khả năng SEO tốt hơn Client Side Rendering (CSR), thư viện UI Shopify Polaris.

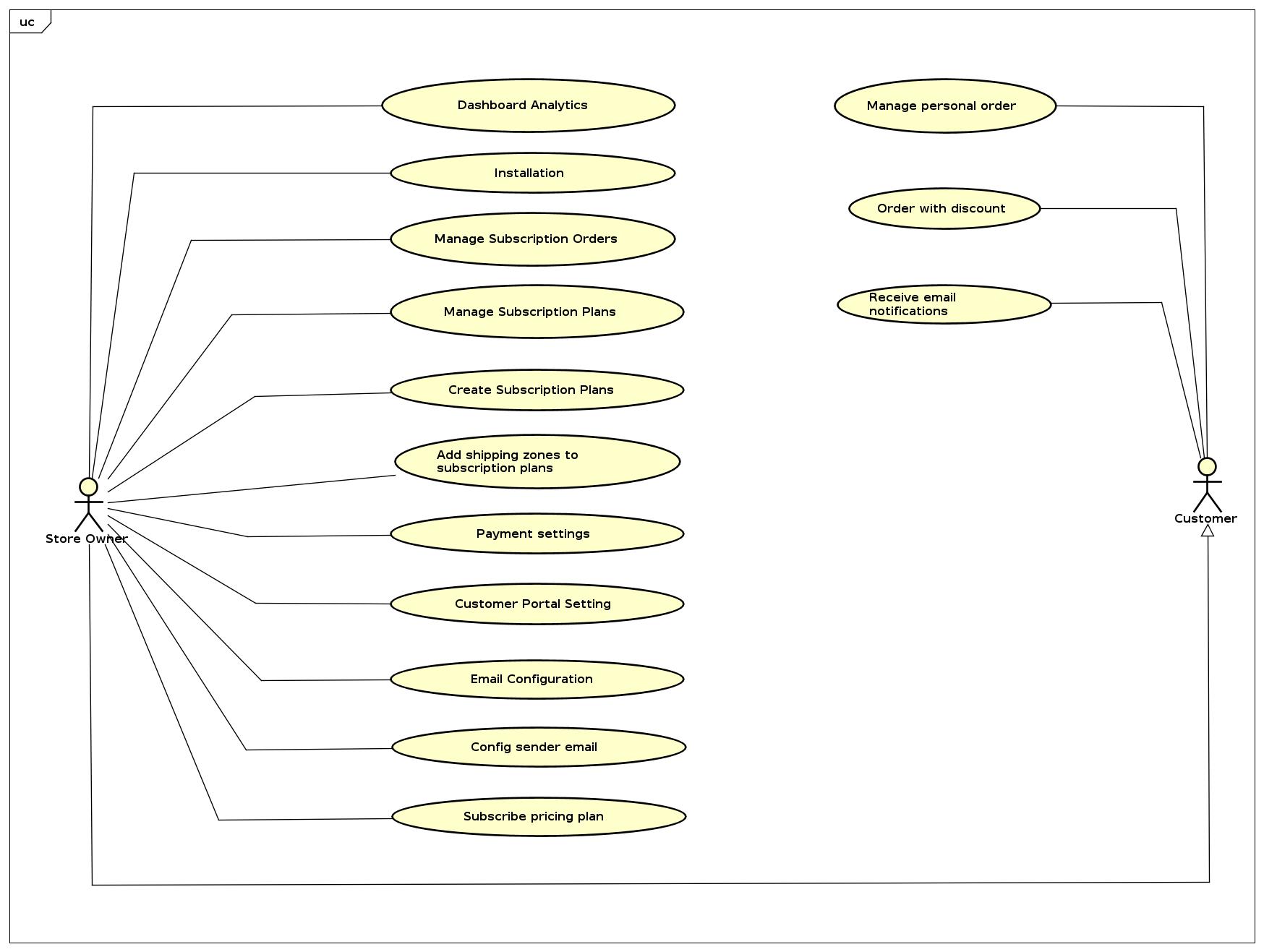
Đối với ứng dụng Subscription & Recurring Pay , dữ liệu được lưu trữ bằng cơ sở dữ liệu MySQL. PlatForm NodeJs hỗ trợ thư viện Sequelize giúp tối ưu hóa câu truy vấn, giảm thiểu thời gian khi truy cập dữ liệu và dễ dàng mở rộng.

## 

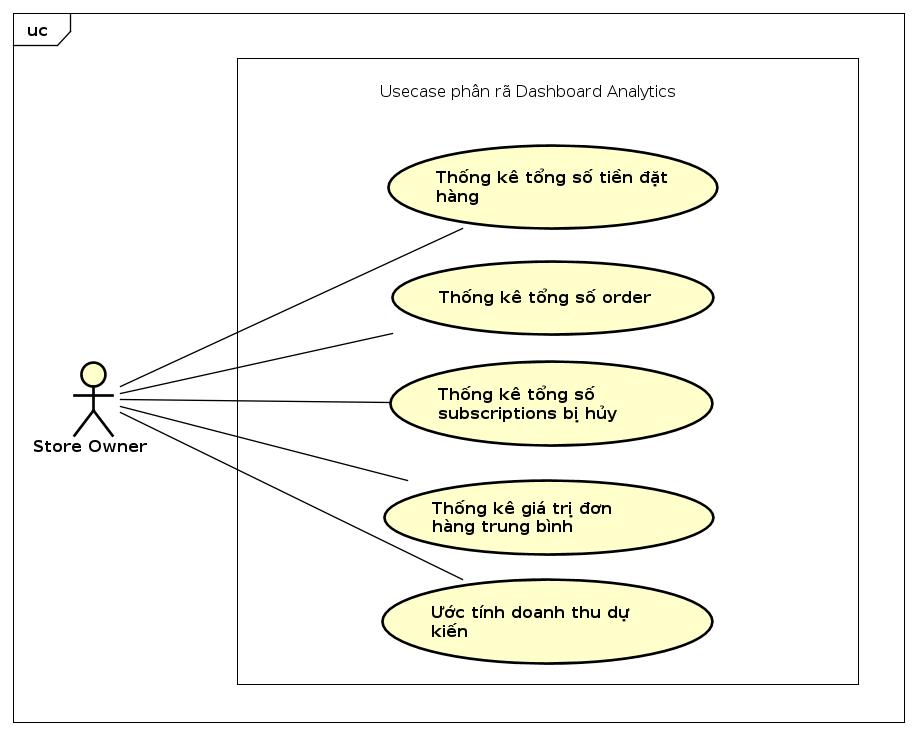
## 

## **1.4 Tổng quan chức năng**

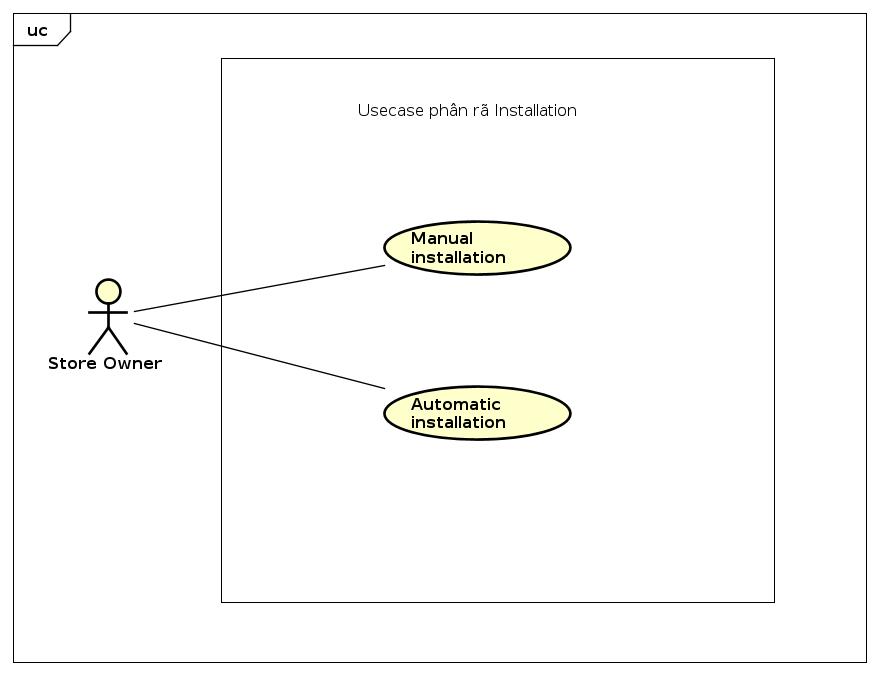
### **1.4.1 Biểu đồ use case tổng quan**



### **1.4.2 Biểu đồ usecase phân rã chức năng Dashboard Analytics**



### **1.4.3 Biểu đồ usecase phân rã chức năng installation**



### 

### 

### 

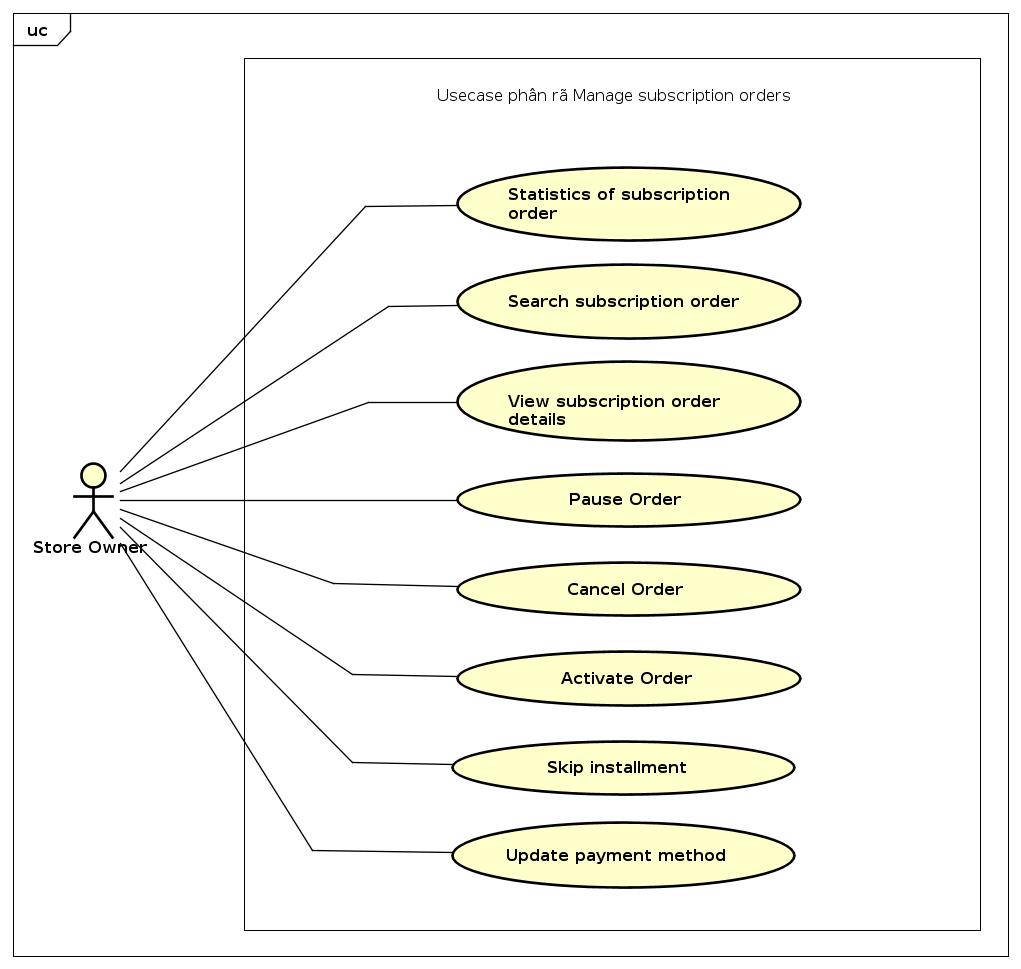
### 

### 

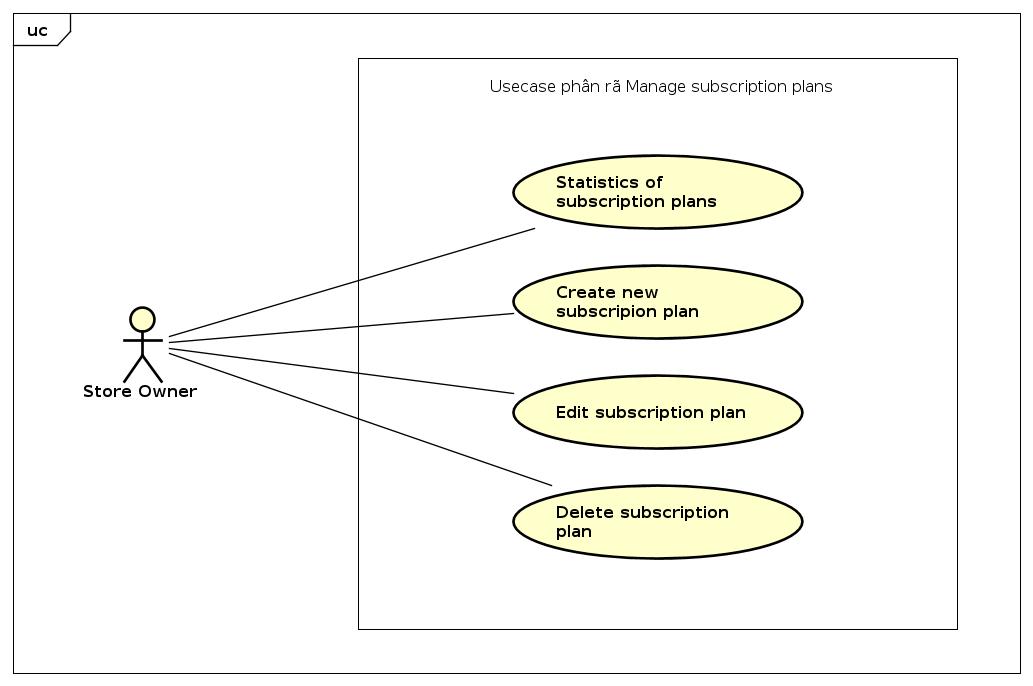
### 

### 

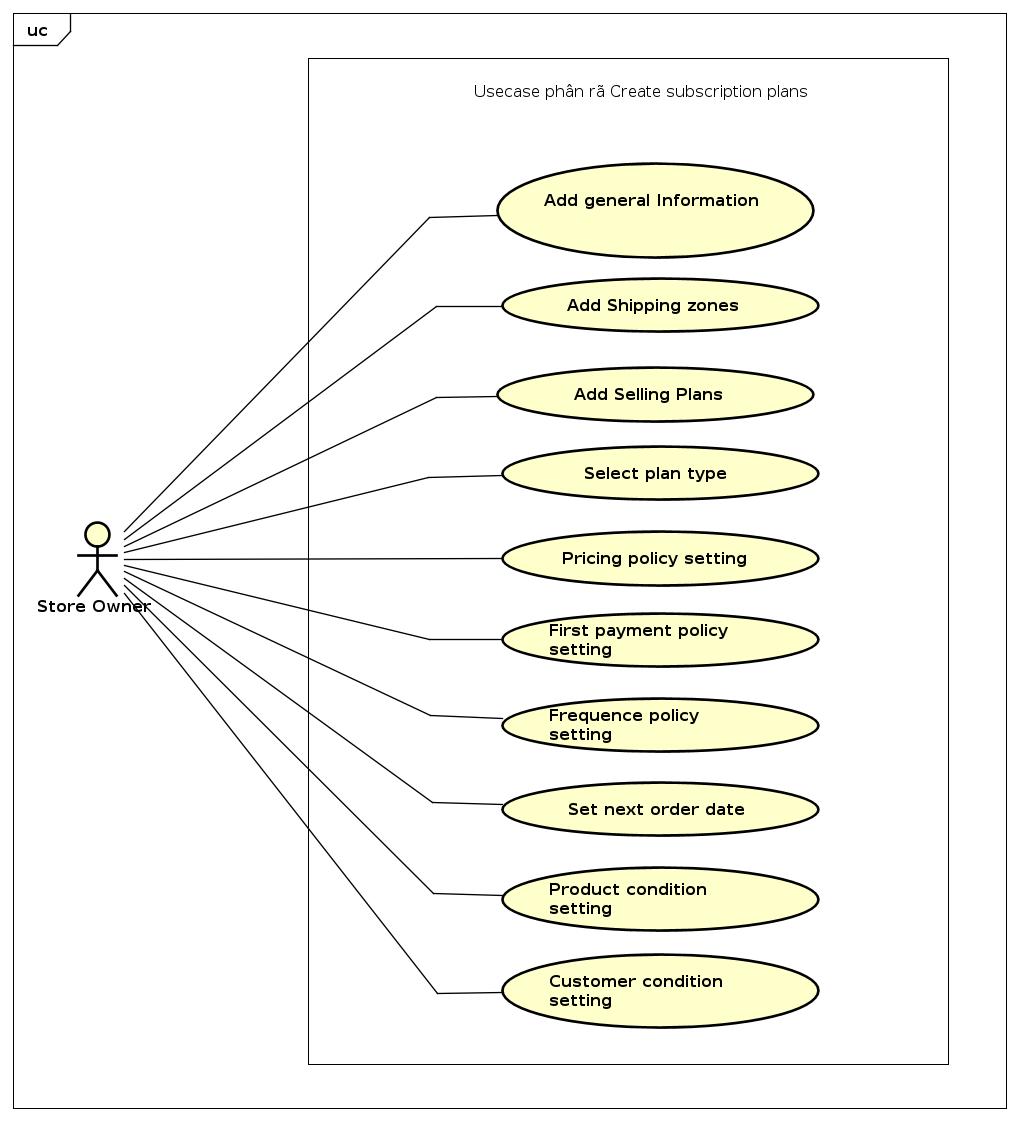
### **1.4.4 Biểu đồ usecase phân rã chức năng Manage subscription orders**



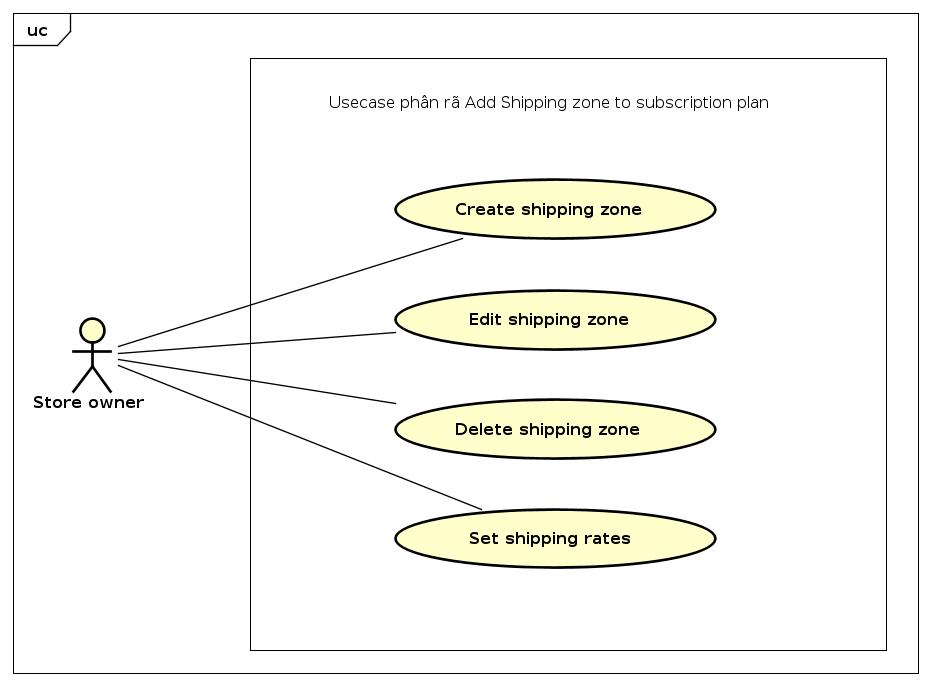
### **1.4.5 Biểu đồ usecase phân rã chức năng Manage Subscription Plans**



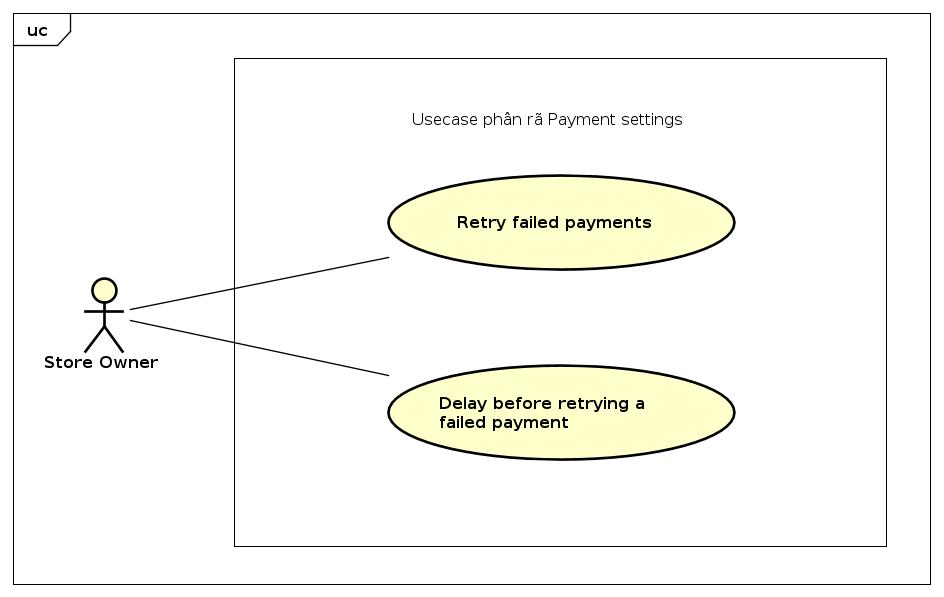
### **1.4.6 Biểu đồ usecase phân rã chức năng Create subscription plans**



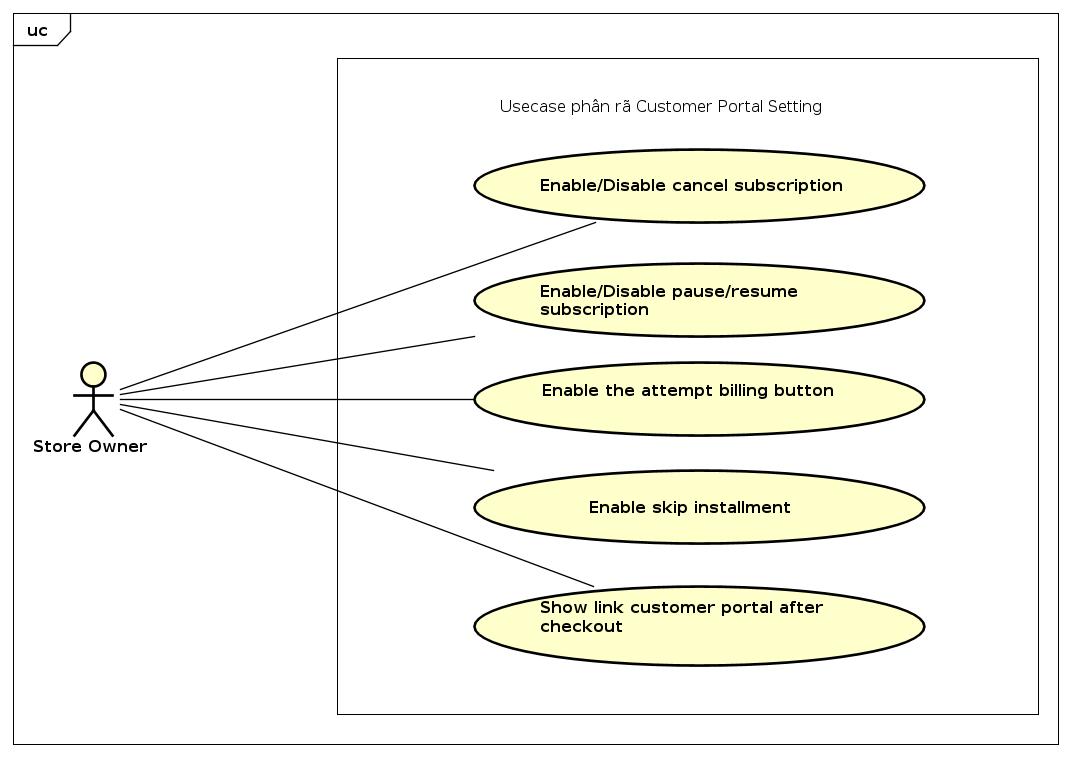
### **1.4.7 Biểu đồ usecase phân rã chức năng Add Shipping zones to subscription plans**



### **1.4.8 Biểu đồ usecase phân rã chức năng Payment setting**



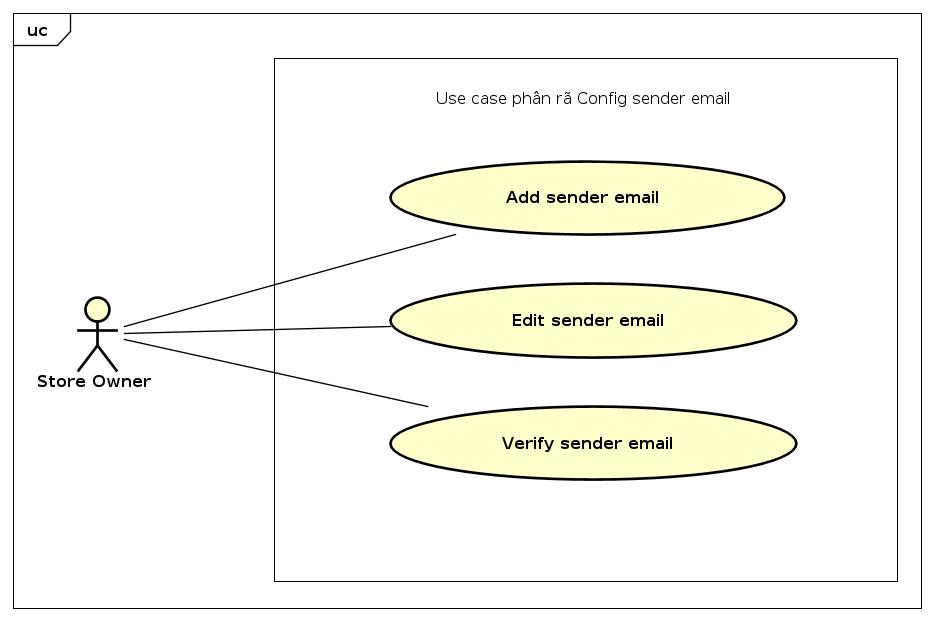
### **1.4.9 Biểu đồ usecase phân rã chức năng Customer Portal Setting**



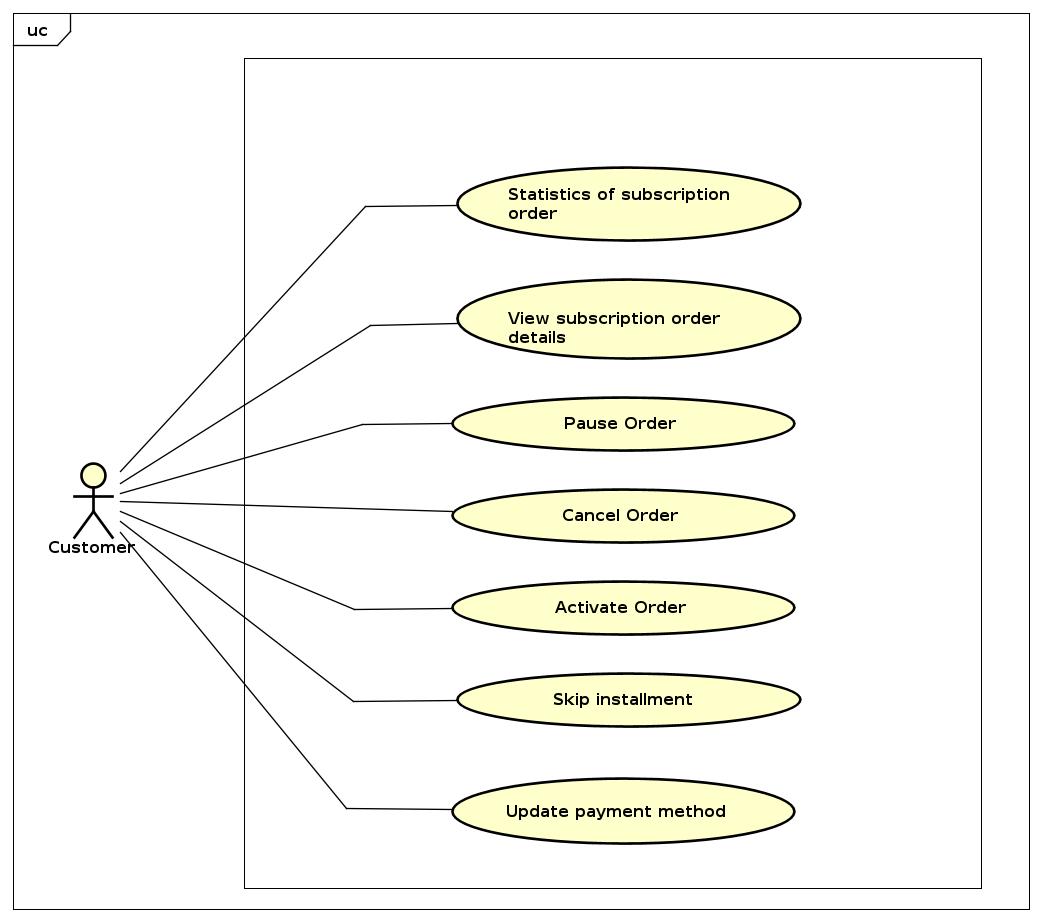
### **1.4.10 Biểu đồ usecase phân rã chức năng Email Configuration**



### **1.4.11 Biểu đồ usecase phân rã chức năng** **Config sender email**



### **1.4.12 Biểu đồ usecase phân rã chức năng Manage personal orders**



## **1.5 Đặc tả chức năng**

## **1.6 Yêu cầu phi chức năng**

# **CHƯƠNG 2. PHÂN TÍCH, THIẾT KẾ BÀI TOÁN**

## **2.1 Công nghệ sử dụng**

### **2.1.1 Ngôn ngữ JavaScript**

Javascript, hay thường được gọi tắt là JS, cùng với HTML và CSS là bộ ba tạo nên hệ thống các trang website. Nếu như HTML là bộ xương sống, CSS phủ lên lớp áo màu sắc thì JS sẽ chịu trách nhiệm cho các chuyển động tương tác với người dùng khi sử dụng website.

Là một ngôn ngữ có cú pháp rành mạch, đơn giản và dễ học, JS đang là ngôn ngữ lập trình phổ biến hơn bao giờ hết. Các trình duyệt website hiện đại đều hỗ JS, giúp cho việc trải nghiệm tương tác của người dùng khi sử dụng website trở nên sinh động và đa dạng hơn, đồng thời, nếu thao tác được xử lý ở phía máy khách hàng thì sẽ giảm thiểu lượng truy cập phải xử lý cho máy chủ. Ngoài ra, với khả năng tương tác và mở rộng cao, JS có thể được “nhúng” vào các ứng dụng website khác nhau thông qua các extension (tiện ích mở rộng), góp phần làm đơn giản hóa thao tác sử dụng các chức năng website.

Với ưu điểm chỉ dịch một lần, nhưng có thể chạy nhiều tiến trình song song, JS sẽ có thời gian biên dịch mã nguồn ngắn hơn so với các ngôn ngữ lập trình truyền thống như là Java, .NET hay PHP. Không những vậy, JS còn được sử dụng trên các máy chủ sử dụng platform NodeJS, giúp việc đồng bộ ngôn ngữ giữa máy chủ và máy khách hàng trở nên dễ dàng hơn.

Lựa chọn JS làm ngôn ngữ chủ yếu xây dựng ứng dụng **Subscription & Recurring Pay** đảm bảo hiệu quả cao nhất cũng như tính dễ mở rộng, dễ quản lý và tái sử dụng mã nguồn trong tương lai sắp tới.

### **2.1.2 Platform NodeJs và Framework KoaJS**

Node.js là một nền tảng chạy trên môi trường V8 JavaScript runtime - một trình thông dịch JavaScript cực nhanh chạy trên trình duyệt Chrome.

NodeJS có thể chạy trên nhiều nền tảng hệ điều hành khác nhau từ Window cho tới Linux, OS X.NodeJS cung cấp các thư viện phong phú ở dạng Javascript Module khác nhau giúp đơn giản hóa việc lập trình và giảm thời gian ở mức thấp nhất.

Những tính năng đặc trưng của NodeJS: Realtime, không đồng bộ, chạy rất nhanh, đơn luồng nhưng khả năng mở rộng cao, không đệm, đã được cấp giấy phép bởi MIT License.

Để tạo dựng một bộ khung cấu trúc của ứng dụng thì NodeJs cần tích hợp thêm các framework bên trong nó. Với ưu thế mềm dẻo, tối ưu hóa và linh hoạt, KoaJS framework là lựa chọn phổ biến hàng đầu để tích hợp với NodeJs.

### **2.1.3 Framework NextJS**

Next.js là một framework dùng để phát triển các ứng dụng React theo tư tưởng Isomorphic (Universal), được phát triển bởi Zeit. Nhờ Next.js chúng ta có thể dễ dàng tạo được một ứng dụng React theo tư tưởng Isomorphic, có bao gồm chức năng Server Side Rendering (SSR).

Nextjs tích hợp nhiều tính năng như:

* Pre-rendering, cả static generation (SSG) và server-side rendering (SSR)
* Tách mã tự động để tải trang nhanh hơn
* Hỗ trợ css và sass, hỗ trợ bất kì thư viện css, js nào
* Hỗ trợ refresh page nhanh chóng ở môi trường development
* Hoàn toàn có thể mở rộng

### **2.1.4 MySQL**

Mọi dữ liệu khách hàng khi sử dụng ứng dụng cần được lưu trữ một cách tối ưu và hiệu quả. Vì vậy, việc lựa chọn một hệ quản trị dữ liệu tối ưu thực sự rất cần thiết.

Hiện nay, mô hình lưu trữ dữ liệu được chia thành 2 loại chính: mô hình dữ liệu quan hệ RDBMS (Relational Database Management System) và mô hình dữ liệu phi quan hệ NoSQL (Not only SQL).

Mô hình dữ liệu quan hệ (tiêu biểu là hệ quản trị dữ liệu MySQL) có tiêu chí ràng buộc dữ liệu rất mạnh mẽ. Mọi dữ liệu khi truy xuất cơ sở dữ liệu quan hệ đều được kiểm tra ràng buộc chặt chẽ. Điều này sẽ đảm bảo tính toàn vẹn cho dữ liệu.

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.

Ưu điểm của MySQL

* Sử dụng dễ dàng: MySQL là cơ sở dữ liệu tốc độ cao và ổn định, công cụ này dễ sử dụng và hoạt động trên nhiều hệ điều hành cung cấp hệ thống lớn các hàm tiện ích.
* Tính bảo mật cao: MySQL phù hợp với các ứng dụng có truy cập cơ sở dữ liệu trên internet vì nó sở hữu nhiều tính năng bảo mật, thậm chí là bảo mật cấp cao.
* Đa tính năng: MySQL có thể hỗ trợ hàng loạt các chức năng SQL từ hệ quản trị cơ sở dữ liệu quan hệ trực tiếp và cả gián tiếp.
* Khả năng mở rộng và mạnh mẽ: Công cụ MySQL có khả năng xử lý khối dữ liệu lớn và có thể mở rộng khi cần thiết.
* Tương thích trên nhiều hệ điều hành: MySQL tương thích để chạy trên nhiều hệ điều hành, như Novell NetWare, Windows \* Linux \*, nhiều loại UNIX \* (như Sun \* Solaris \*, AIX và DEC \* UNIX), OS / 2, FreeBSD \*,.... MySQL cũng cung cấp phương tiện mà các máy khách có thể chạy trên cùng một máy tính với máy chủ hoặc trên một máy tính khác (giao tiếp qua mạng cục bộ hoặc Internet).
* Cho phép khôi phục: MySQL cho phép các transaction được khôi phục, cam kết và phục hồi sự cố.

Nhược điểm của MySQL

* MySQL bị hạn chế dung lượng, cụ thể, khi số bản ghi của người dùng lớn dần, sẽ gây khó khăn cho việc truy xuất dữ liệu, khiến người dùng cần áp dụng nhiều biện pháp để tăng tốc độ chia sẻ dữ liệu như chia tải database ra nhiều server, hoặc tạo cache MySQL.
* So với Microsoft SQL Server hay Oracle, độ bảo mật của MySQL chưa cao bằng. Và quá trình Restore cũng có phần chậm hơn. Tuy nhiên, những nhược điểm này không đáng kể với những hệ quản trị nội dung cỡ trung, bởi chỉ những hệ thống lớn hoặc rất lớn mới phát sinh các yêu cầu phức tạp hơn.

### **2.1.5 Liquid**

Liquid là một mã nguồn mở, Ngôn ngữ mẫu được sáng tạo bởi Shopify dựa trên nền tảng ngôn ngữ Ruby. Nó là thành phần chính của các theme trong Shopify và được sử dụng để tải các nội dung động lên giao diện các cửa hàng. Liquid sử dụng sự kết hợp các tag (nhãn), object (đối tượng) và filter (bộ lọc) để tải những nội dung động. Chúng được sử dụng bên trong các tài liệu mẫu của Liquid, nhóm các tệp tin này tạo thành một theme.

Ưu điểm của Liquid

* Cấu trúc đơn giản, dễ nhớ, dễ sử dụng
* Kết hợp dễ dàng với cấu trúc HTML có sẵn
* Người dùng có thể tự tùy biến giao diện hiển thị mà không cần liên quan đến backend

### **2.1.6 Shopify API & GraphQL**

### **2.1.7 Shopify Polaris**

Polaris là một thư viện hỗ trợ cho việc xây dựng giao diện theo tiêu chuẩn của Shopify