

**TRƯỜNG ĐẠI HỌC AN GIANG**

**KHOA CÔNG NGHỆ THÔNG TIN**

-----🙞🙜🕮🙞🙜-----



**ĐỒ ÁN LẬP TRÌNH**

**SHOP CÔNG NGHỆ**

|  |  |
| --- | --- |
| *Giáo Viên Hướng Dẫn*: |  |
| *Sinh Viên Thực Hiện:* | Nguyễn Phúc Minh – DTH195307 |
|  |
|  |

An Giang 12, 2022

**MỤC LỤC**

[WEBSITE CÔNG NGHỆ 1](#_Toc122451906)

[CHƯƠNG 1 : MÔ TẢ BÀI TOÁN 1](#_Toc122451907)

[I. Mô tả chương trình 1](#_Toc122451908)

[II. Yêu cầu hệ thống 2](#_Toc122451909)

[III. Giới hạn phạm vi 2](#_Toc122451910)

[CHƯƠNG 2 : PHÂN TÍCH HỆ THỐNG 2](#_Toc122451911)

[I. Sơ Đồ BFD 2](#_Toc122451912)

[II. Sơ đồ DFD ngữ cảnh 3](#_Toc122451913)

[III. Sơ đồ DFD mức 0 3](#_Toc122451915)

[CHƯƠNG 3. CƠ SỞ LÝ THUYẾT 4](#_Toc122451916)

[CHƯƠNG 4. THIẾT KẾ CƠ SỞ DỮ LIỆU 7](#_Toc122451917)

[1. Bảng dữ liệu người dùng (users) 7](#_Toc122451918)

[2. Bảng dữ liệu sản phẩm (products) 8](#_Toc122451919)

[3. Bảng dữ liệu đánh giá(reviews) 8](#_Toc122451920)

[4. Bảng dữ liệu danh mục (categorys) 9](#_Toc122451921)

[5 Bảng dữ liệu mua hàng (orders) 9](#_Toc122451922)

[CHƯƠNG 5. THIẾT KẾ CHƯƠNG TRÌNH 10](#_Toc122451923)

[I. Thiết kế chức năng: 10](#_Toc122451924)

[II. Thiết kế giao diện : 11](#_Toc122451925)

[CHƯƠNG 6 : HƯỚNG PHÁT TRIỂN 16](#_Toc122451926)

[I. Nhận xét 16](#_Toc122451927)

[I. Hướng phát triển: 16](#_Toc122451928)

[II. Tài liệu tham khảo: 17](#_Toc122451929)

**NHẬN XÉT GIÁO VIÊN**

**…………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………**

**Kế hoạch thực hiện**

**I. Kế hoạch thực hiện:**

Tuần 1: Từ ngày 3 – 9/3/2022

* Tìm hiểu hệ thống, đặc tả.
* Nghiên cứu ngôn ngữ.

Tuần 2: Từ ngày 10 - 16/3/2022

* Nghiên cứu mô hình MERN.

Tuần 3: Từ ngày 17 - 23/3/2022

* Xác định chức năng.
* Phân tích (DFD,ERD).
* Thiết kế giao diện.

Tuần 4: Từ ngày 24 – 30/3/2022

* Hoàn thành báo cáo.
* Code demo.

# 

# **WEBSITE CÔNG NGHỆ**

**LỜI MỞ ĐẦU**

**-----🙞🙜🕮🙞🙜-----**

Ngày nay, xã hội phát triển mạnh mẽ và đi cùng với nó là sự phát triển của ngành công nghệ thông tin. Công nghệ được áp dụng hầu hết trong tất cả lĩnh vực : giáo dục, chính trị, y tế , quân sự,… Nó trở thành một nhân tố quan trọng trong sự phát triển của toàn nhân loại. Đặc biệt trong vấn đề quản lý đang được mọi người quan tâm và phát triển. Nhiều công nghệ được đưa vào cuộc sống, cùng với nó là sự ra đời của hàng loạt các phần mềm, chương trình quản lý nhằm phục vụ nhu cầu của người sử dụng trên toàn thế giới.

Hiện nay, đa số các cửa hàng công nghệ đều chưa sử dụng các phần mềm quản lý trực quan, mà thay vào đó là sử dụng các bước thủ công như ghi chép. Nó để lại nhiều khuyết điểm như : khó quản lý, dễ thất lạc dữ liệu, tính bảo mật tương đối thấp, khó lưu trữ,…..Với mục tiêu giúp các cửa hàng nâng cao cách quản lý cửa hàng, chúng em tìm hiểu và phát triển phần mềm “Quản lý cửa hàng công nghệ”.

Với “Phần mềm quản lý cửa hàng công nghệ” hy vọng sẽ giúp ích được nhiều trong việc quản lý cửa hàng một cách nhanh chóng và hiệu quả hơn. Với giao diện đơn giản , dễ sử dụng, phần mềm giúp giảm thời gian, công sức cho việc sắp xếp, tìm kiếm, tính toán.

# **CHƯƠNG 1 : MÔ TẢ BÀI TOÁN**

## **I. Mô tả chương trình**

Một cửa hàng yêu cầu viết một phần mềm quản lý về công nghệ. Chương trình cho phép quản lý hàng hóa, hóa đơn, chi tiết trong từng hóa đơn và khách hàng.

Chương trình cung cấp cho người dùng một tài khoản để đăng nhập vào hệ thống để sử dụng hệ thống quản lý này. Với tài khoản đăng nhập, hệ thống sẽ xác định người đăng nhập là chủ cửa hàng để cung cấp quyền hạn cho chủ cửa hàng.

Khi khách hàng yêu cầu một sản phẩm, cửa hàng kiểm tra sản phẩm trong kho. Nếu còn cửa hàng tư vấn cho khách hàng thông tin sản phẩm. Sau khi khách hàng đồng ý mua sản phẩm , cửa hàng lập một phiếu mua hàng gồm đầy đủ thông tin khách hàng(tên khách hàng, số điện thoại, ….) cùng với thông tin sản phẩm(tên sản phẩm, số lượng, thành tiền,….).

Cửa hàng gổm nhiều sản phẩm khác nhau với các linh kiện điện thoại, Khi khách hàng yêu cầu thanh toán, hệ thống tự động thanh toán tiền của khách hàng bao gồm : đơn giá các món hàng \* số lượng.

Hệ thống cửa hàng gồm các chức năng như : thêm mới, sửa , xóa, cập nhật, thống kê sản phẩm. Đồng thời còn thêm mới, sửa, cập nhật thông tin khách hàng. Ngoài ra, hệ thống còn có chức năng tính tồn kho để cửa hàng nhập hàng vào mỗi tháng, cung cấp kịp thời nhu cầu sử dụng cho khách hàng.

Hệ thống còn chức năng thống kê những mặt hàng nào đã được bán, đã hết hoặc còn ít trong kho.

Cuối cùng, hệ thống hỗ trợ việc sao lưu, phục hồi dữ liệu cho admin với excel

## **II. Yêu cầu hệ thống**

Nhằm giúp cửa hàng nắm rõ thông tin khách hàng và các sản phẩm, giúp cửa hàng tìm kiếm thông tin khách hàng một cách nhanh chóng và chính xác theo nhiều mục đích khác nhau. Để dễ dàng quản lý thông tin khách hàng, dễ dàng tính toán, xuất hóa đơn thanh toán theo nhu cầu thanh toán của khách hàng. Hỗ trợ cửa hàng linh hoạt hơn trong các hoạt động cập nhật, sửa đổi thông tin,…Giúp dễ dàng việc quản lý một cách dễ dàng, nhanh chóng và chính xác hơn. Dựa vào thông tin đầy đủ do hệ thống cung cấp, cửa hàng có thể dễ dàng quản lý được thông tin khách hàng, quản lý việc tính toán, xuất hóa đơn, quản lý về thống kê báo cáo,..

## III. Giới hạn phạm vi

Quản lý việc bán sản phẩm công nghệ, linh kiện và quản lý kho

# **CHƯƠNG 2 : PHÂN TÍCH HỆ THỐNG**

## I. Sơ Đồ BFD

Sơ đồ 1: sơ đồ BFD

## II. Sơ đồ DFD ngữ cảnh

## 

Nhà cung cấp

Quản trị

Quản Lý Bán Hàng

Khách hàng

Yêu Cầu

Phản hồi

Gửi hàng

Đặt hàng

Đặt hàng

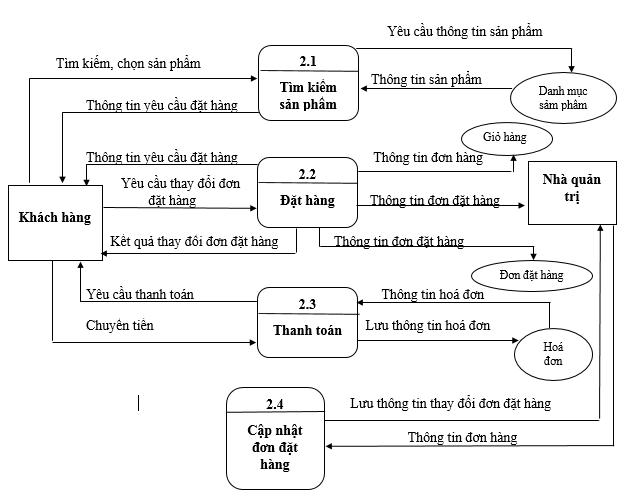
Giao dịch

Phản hồi

Yêu cầu

Sơ đồ *2*: sơ đồ DFD ngữ cảnh

## III. Sơ đồ DFD mức 0



Sơ đồ 3: sơ đồ DFD mức 0

# **CHƯƠNG 3. CƠ SỞ LÝ THUYẾT**

**I. NGÔN NGỮ JAVASCRIPT VÀ NODEJS EXPRESS FRAMEWORK**

**1. JavaScript là gì ?**

JavaScript (thường được viết tắt là JS) được biết đến là một ngôn ngữ lập trình theo dạng kịch bản được phát triển dựa trên các đối tượng phát triển sẵn có hay do chính tác giả tự định nghĩa ra nhằm giúp cho website trở nên hấp dẫn và sinh động hơn. Ngoài ra Javascript còn đảm nhiệm vai trò quan trọng đối với việc tương tác giữa các trang web với khách hàng khi nhận các phản ứng từ các khách hàng. Bởi chính những sự tiện ích và hiệu quả mà Javascript mang đã giúp ngày càng nhiều các lập trình viên lựa chọn và tin dùng. nó cũng được sử dụng trong nhiều môi trường không phải trình duyệt. có tính năng động và hỗ trợ các kiểu lập trình hướng đối tượng, mệnh lệnh và chức năng.

JavaScript chạy ở phía máy khách của web, có thể được sử dụng để thiết kế / lập trình cách các trang web hoạt động khi xảy ra sự kiện. JavaScript là một ngôn ngữ kịch bản dễ học và cũng mạnh mẽ, được sử dụng rộng rãi để kiểm soát hành vi của trang web.

**1.1. Tại sao chọn ngôn ngữ JavaScript**

Có nhiều lý do để sử dụng ngôn ngữ này chiếm ưu thế em xin nêu ra đây một số lý do cơ bản:

1. JavaScript rất linh hoạt: JavaScript giúp cho các lập trình viên xử lý bất kỳ khía cạnh nào của thiết kế ứng dụng. Bạn có thể sử dụng chúng kết hợp với framework ReactJs để lập trình Front End. Ngược lại, nếu muốn lập trình Backend, bạn hãy dùng JavaScript và Node.js.
2. Ngôn ngữ rất dễ học, dễ viết, cho phép người dùng sử dụng tối ưu hóa trên các thiết bị điện tử kể cả máy tính hay các thiết bị di động thông qua một bộ mã dùng trong thiết kế web responsive.

**1.2. Lịch sử phát triền**

JavaScript được bắt đầu vào năm 1995 tại Netscape Communications. Vào thời gian này, các nhà sản xuất Netscape browser nhận ra rằng, việc thêm một "glue language" để nâng cao trải nghiệm người dùng sẽ làm tăng sự tiêu thụ nội dung của người dùng. Vì vậy, họ đã đưa Brendan Eich nhúng vào Scheme Programming language. Tuy nhiên, tại thời điểm đó, Java hiện là một ngôn ngữ mới, nóng hổi của website, nên họ đã quyết định làm cho ngôn ngữ này gần gũi với cú pháp của Java. Kết quả đem lại chính là JavaScript, với các tính năng của Scheme, định hướng đối tượng của SmallTalk và cú pháp của Java. Phiên bản đầu tiên của ngôn ngữ này được đặt tên là Mocha vào tháng 5 năm 1995, đổi tên thành LiveScript vào tháng 9 năm 1995, và được đổi tên thành JavaScript vào tháng 12 năm 1995.

Năm 1996, JavaScript đã được gửi đến ECMA International để hoàn thiện trở thành, một đặc tính kỹ thuật tiêu chuẩn. Vào tháng 6 năm 1997, đặc tính kỹ thuật chính thức đầu tiên cho ngôn ngữ được phát hành là ECMA-262. Phiên bản mới nhất của ngôn ngữ là ECMAScript 2017 được phát hành vào tháng 6 năm 2017.

**2. NodeJs là gì ?**

Nodejs là một nền tảng (Platform) phát triển độc lập được xây dựng ở trên Javascript Runtime của Chorm mà chúng ta có thể xây dựng được các ứng dụng mạng một cách nhanh chóng và dể dàng mở rộng.

Nodejs được xây dựng và phát triển từ năm 2009, bảo trợ bởi công ty Joyent trụ sở tại California, Hoa Kỳ. Phần Core bên dưới của Nodejs được viết hầu hết bằng C++ nên cho tốc độ xử lý và hiệu năng khá cao.

Nodejs tạo ra được các ưng dụng có tốc độ xử lí nhanh, realtime thời gian thực.

Nodejs áp dụng cho các sản phẩm có lượng truy cập lớn cần mở rộng nhanh, cần đổi mới công nghệ, hoặc tạo ra các dự án Startup nhanh nhất có thể.

**2.1. Tại sao lại chọn Nodejs**

Các ứng dụng NodeJs được viết bằng JavaScript, ngôn ngữ này là một ngôn ngữ khá thông dụng. Theo tác giả của ngôn ngữ JavaScript, Ryan Dahl:

“JavaScript có những đặc tính mà làm cho nó rất khác biệt so với các ngôn ngữ lập trình động còn lại, cụ thể là nó không có khái niệm về đa luồng, tất cả là đơn luồn và hướng sự kiện.

Nodejs chạy đa nền tảng phía server, sử dụng kiến trúc hướng sự kiện Event-driven, cơ chế non-blocking I/O làm cho nó nhẹ và hiệu quả.

Có thể chạy ứng dụng Nodejs ở bất kỳ đâu trên máy Mac – Window – Linux, hơn nữa cộng đồng Nodejs rất lớn và hoàn toàn miễn phí.

Các ứng dụng NodeJs đáp ứng tốt thời gian thực và chạy đa nền tảng, đa thiết bị.

**3. ExpressJs là gì ?**

Expressjs hay còn được viết là Express js, Express.js. Đây là một framework mã nguồn mở miễn phí cho Node.js. Express.js được sử dụng trong thiết kế và xây dựng các ứng dụng web một cách đơn giản và nhanh chóng.

Vì Express js chỉ yêu cầu ngôn ngữ lập trình Javascript nên việc xây dựng các ứng dụng web và API trở nên đơn giản hơn với các lập trình viên và nhà phát triển.Expressjs cũng là một khuôn khổ của Node.js do đó hầu hết các mã code đã được viết sẵn cho các lập trình viên có thể làm việc.

Nhờ có Expressjs mà các nhà lập trình có thể dễ dàng tạo các ứng dụng 1 web, nhiều web hoặc kết hợp. Do có dung lượng khá nhẹ, Expressjs giúp cho việc tổ chức các ứng dụng web thành một kiến trúc MVC có tổ chức hơn.Để có thể sử dụng được mã nguồn này, chúng ta cần phải biết về Javascript và HTML.

Expressjs cũng là một phần của công nghệ giúp quản lý các ứng dụng web một cách dễ dàng hơn hay còn được gọi là ngăn xếp phần mềm MEAN.Nhờ có thư viện Javascript của Express js đã giúp cho các nhà lập trình xây dựng nên các ứng dụng web hiệu quả và nhanh chóng hơn. Expressjs cũng được sử dụng để nâng cao các chức năng của Node.js.

Trên thực tế, nếu không sử dụng Express.js, bạn sẽ phải thực hiện rất nhiều bước lập trình phức tạp để xây dựng nên một API hiệu quả. Express js đã giúp cho việc lập trình trong Node.js trở nên dễ dàng hơn và có nhiều tính năng mới bổ sung.

**3.1. Tại sao chọn ExpressJs**

1. ExpressJs hoàn toàn miễn phí.
2. Hỗ trợ xây dựng website một cách nhanh chóng.
3. Phổ biến được nhiều người sử dụng, cũng như có cộng đồng support lớn.

**II. HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU NOSQL MONGODB**

**1. Cơ sở dữ liệu NoSQL là gì ?**

Non-Relational SQL Là loại Cơ sở dữ liệu mà không truy vấn bằng ngôn ngữ SQL. Các Hệ quản trị cơ sở dữ liệu NoSQL sử dụng các kỹ thuật lưu trữ khác nhau: cặp khóa – giá trị (key-value pair) trên RAM, cặp khóa – giá trị (key-value pair) trên ổ cứng, tài liệu/tập tin (document). NoSQL không có các chuẩn chung thống nhất như SQL. Các cơ sở dữ liệu NoSQL đang được sử dụng ngày càng nhiều trong các ứng dụng dữ liệu lớn và độ trễ thấp và ứng dụng nền web thời gian thực.

Thuật ngữ NoSQL được giới thiệu lần đầu vào năm 1998 sử dụng làm tên gọi chung cho các lightweight open source relational database (cơ sở dữ liệu quan hệ nguồn mở nhỏ) nhưng không sử dụng SQL cho truy vấn. Vào năm 2009, Eric Evans, nhân viên của Rackspace giới thiệu lại thuật ngữ NoSQL trong một hội thảo về cơ sở dữ liệu nguồn mở phân tán. Thuật ngữ NoSQL đánh dấu bước phát triển của thế hệ database mới: distributed (phân tán) + non-relational (không ràng buộc). Đây là 2 đặc tính quan trọng nhất.

**1.1. Tại sao cần phải có NoSQl**

Sở dĩ người ta phát triển NoSQL suất phát từ yêu cầu cần những database có khả năng lưu trữ dữ liệu với lượng cực lớn, truy vấn dữ liệu với tốc độ cao mà không đòi hỏi quá nhiều về năng lực phần cứng cũng như tài nguyên hệ thống và tăng khả năng chịu lỗi. Đây là những vấn đề mà các relational database không thể giải quyết được.

**1.2. Một số đặc điểm chung**

1. High Scalability: Gần như không có một giới hạn cho dữ liệu và người dùng trên hệ thống.
2. High Availability: Do chấp nhận sự trùng lặp trong lưu trữ nên nếu một node (commodity machine) nào đó bị chết cũng không ảnh hưởng tới toàn bộ hệ thống.
3. Atomicity: Độc lập data state trong các operation.
4. Consistency: chấp nhận tính nhất quán yếu, có thể không thấy ngay được sự thay đổi mặc dù đã cập nhật dữ liệu.
5. Durability: dữ liệu có thể tồn tại trong bộ nhớ máy tính nhưng đồng thời cũng được lưu trữ lại đĩa cứng.

**2. MongoDB là gì ?**

MongoDB là một hệ quản trị cơ sở dữ liệu mã nguồn mở, là CSDL thuộc NoSql và được hàng triệu người sử dụng, MongoDB là một database hướng tài liệu (document), các dữ liệu được lưu trữ trong document kiểu JSON. MongoDB được phát triển bởi MongoDB Inc. và được cấp phép theo Giấy phép Công cộng phía Máy chủ (SSPL).

Với CSDL quan hệ chúng ta có khái niệm bảng, các cơ sở dữ liệu quan hệ (như MySQL hay SQL Server…) sử dụng các bảng để lưu dữ liệu thì với MongoDB chúng ta sẽ dùng khái niệm là collection thay vì bảng.

So với RDBMS thì trong MongoDB collection ứng với table, còn document sẽ ứng với row , MongoDB sẽ dùng các document thay cho row trong RDBMS.

Các collection trong MongoDB được cấu trúc rất linh hoạt, cho phép các dữ liệu lưu trữ không cần tuân theo một cấu trúc nhất định.

**2.1 Tại sao chọn MongoDB**

1. Lưu trữ dữ liệu dưới dạng JSON, mỗi một collection sẽ có các kích cỡ và các document khác nhau nên sẽ rất linh hoạt cho việc lưu trữ dữ liệu.
2. Khác với RDBMS dữ liệu trong đây không có sự ràng buộc và không có yêu cầu tuân theo khuôn khổ nhất định nên các thao tác được thực hiện rất nhanh => tối ưu performance.
3. Tiếp đến cũng là về hiệu suất, khi có trư vấn dữ liệu thì các record được cached lên bộ nhớ Ram, để lượt truy vấn sau diễn ra nhanh hơn mà không cần đọc từ ổ cứng.
4. Rất dễ mở rộng, MongoDB dễ dàng mở rộng hệ thống bằng cách thêm node vào cluster – cụm các node chứa dữ liệu giao tiếp với nhau.

# **CHƯƠNG 4. THIẾT KẾ CƠ SỞ DỮ LIỆU**

### Bảng dữ liệu người dùng (users)

Bảng 5: Bảng dữ liệu người dùng

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| stt | Tên trường | Kiểu dữ liệu | Mô tả | Ghi chú |
| 1 | \_id | ObjectId | Id | Primary key |
| 2 | username | String | Tên người dùng |  |
| 3 | email | String | Email |  |
| 4 | isAdmin | String | Quyền |  |
| 5 | password | String | Mật khẩu |  |
| 6 | avatar | String | Ảnh đại diện |  |
| 8 | phone | String | Số điện thoại |  |
| 9 | createdAt | String | Thời gian tạo bản ghi |  |
| 10 | updatedAt | String | Thời gian cập nhật bản ghi |  |

### Bảng dữ liệu sản phẩm (products)

Bảng 6: Bảng dữ liệu bài viết

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| stt | Tên trường | Kiểu dữ liệu | Mô tả | Ghi chú |
| 1 | \_id | ObjectId | Id | Primary key |
| 2 | category | ObjectId | Id category | Ref collection category |
| 3 | name | String | Tên sản phẩm |  |
| 4 | image | String | Ảnh sản phẩm |  |
| 5 | description | String | Mô tả |  |
| 6 | rating | Number | Đánh giá |  |
| 7 | numberReviews | Number | Số lượng đánh giá |  |
| 8 | price | Number | Giá |  |
| 9 | countInStock | Number | Tồn |  |
| 10 | reviews | Array | Tất cả đánh giá | Array of reviews schema |
| 11 | createdAt | String | Thời gian tạo bản ghi |  |
| 12 | updatedAt | String | Thời gian cập nhật bản ghi |  |

### Bảng dữ liệu đánh giá(reviews)

Bảng 7: Bảng dữ liệu bình luận

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| stt | Tên trường | Kiểu dữ liệu | Mô tả | Ghi chú |
| 1 | \_id | ObjectId | Id | Primary key |
| 2 | name | String | Tên tài khoản |  |
| 3 | rating | Number | Mức độ đánh giá |  |
| 4 | comment | String | Nội dung đánh giá |  |
| 5 | user | ObjectId | Id user | Ref collection user |
| 6 | created\_at | String | Thời gian tạo bản ghi |  |
| 7 | updated\_at | String | Thời gian cập nhật bản ghi |  |

### Bảng dữ liệu danh mục (categorys)

Bảng 8: Bảng dữ liệu danh mục

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| stt | Tên trường | Kiểu dữ liệu | Mô tả | Ghi chú |
| 1 | \_id | ObjectId | Id | Primary key |
| 2 | name | String | Tên danh mục |  |
| 4 | description | String | Mô tả danh mục |  |
| 5 | image | String | Ảnh danh mục |  |
| 6 | created\_at | String | Thời gian tạo bản ghi |  |
| 7 | updated\_at | String | Thời gian cập nhật bản ghi |  |

### 5 Bảng dữ liệu mua hàng (orders)

Bảng 8: Bảng dữ liệu mua hàng

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| stt | Tên trường | Kiểu dữ liệu | Mô tả | Ghi chú |
| 1 | \_id | ObjectId | Id | Primary key |
| 2 | user | ObjectId | Id user | Ref collection user |
| 3 | orderItems | Array | Món hàng đặt |  |
| 3.1 | name | String | Tên hàng | Child of orderItems |
| 3.2 | qty | Number | Số lượng | Child of orderItems |
| 3.3 | image | String | Ảnh món hàng | Child of orderItems |
| 3.4 | price | Number | Đơn giá | Child of orderItems |
| 3.5 | product | ObjectId | Id product | Ref collection product |
| 4 | ShippingAddress | Object | Địa chỉ giao hàng |  |
| 4.1 | address | String | Địa chỉ nhà | Child of shippingAddress |
| 4.2 | city | String | Địa chỉ thành phố | Child of shippingAddress |
| 4.3 | country | String | Địa chỉ quốc gia | Child of shippingAddress |
| 4.4 | postalCode | String | Mã bưu chính | Child of shippingAddress |
| 5 | paymentMethod | Object | Phương thức thanh toán |  |
| 6 | paymentResult | Object | Kết quả thanh toán |  |
| 6.1 | Id | ObjectId | Id | Child of paymentResult |
| 6.2 | status | String | Trạng thái | Child of paymentResult |
| 6.3 | update\_time | String | Thời gian trạng thái | Child of paymentResult |
| 6.3 | email\_address | String | Email | Child of paymentResult |
| 7 | taxPrice | Number | Thuế hàng hóa |  |
| 8 | shippingPrice | Number | Thuế vận chuyển |  |
| 9 | isPaid | Boolean | Trạng thái trả |  |
| 10 | paidAt | Date | Thời gian trả |  |
| 11 | isDelivered | Boolean | Trạng thái vận chuyển |  |
| 12 | deliveredAt | Date | Thời gian vận chuyển |  |
| 13 | created\_at | String | Thời gian tạo bản ghi |  |
| 14 | updated\_at | String | Thời gian cập nhật bản ghi |  |

# **CHƯƠNG 5. THIẾT KẾ CHƯƠNG TRÌNH**

## I. Thiết kế chức năng:

**CHỨC NĂNG HỆ THỐNG**

* Đăng nhập :chức năng để người sử dụng xác nhận quyền được truy cập.

**CHỨC NĂNG QUẢN LÝ CỬA HÀNG ĐIỆN THOẠI**

* Nhập thông tin: giúp người sử dụng nhập các thông tin về điện thoại, hóa đơn, khách hàng, …
* Tra cứu : giúp người sử dụng tra cứu về số lượng tồn kho, số hóa đơn, thành tiền tiền thoại hoặc thông tin khách hàng
* Bán hàng: lấy thông tin từ hệ thống để xử lý
* In hóa đơn**:** bao gồm thông tin các mặt hàng và tổng tiển
* Cập nhật :chức năng thực hiện khi xuất , nhập kho hay bán hàng,… cần cập nhật lại các thông tin để cửa hàng được quản lý một cách hiệu qủa

**CHỨC NĂNG THỐNG KÊ , BÁO CÁO**

* Thông kê : nhân viên cần thống kê số lượng điện thoại bán ra.
* Báo cáo : nhân viên báo cáo số lượng hàng tồn kho.
* Tính doanh thu **:** được thực hiện khi chủ cửa hàng yêu cầu nhân viên tính doanh thu tháng.

## II. Thiết kế giao diện :

**1.CLIENT:**

**1.1 Login**

**Graphical user interface

Description automatically generated**

**1.2: Dashbroad**

Graphical user interface, application, website

Description automatically generated

**1.3 Detail product**

Graphical user interface, text, website

Description automatically generated

**1.4 Checkout cart**

Graphical user interface, application, website

Description automatically generated

**1.5 Place order screen**

Graphical user interface, application

Description automatically generated

**1.6 Profile tab**

Graphical user interface, application

Description automatically generated

**1.7 Payment screen**

Graphical user interface, application, website

Description automatically generated

**2.ADMIN**

**2.1 Dashbroad**

**Graphical user interface

Description automatically generated 2.2 Product list**

**A screenshot of a computer

Description automatically generated with medium confidence**

**2.3 Category list**

**Graphical user interface, application, Teams

Description automatically generated**

**2.4 Category detail**

**A screenshot of a computer

Description automatically generated**

**2.5 Order list**

**Graphical user interface

Description automatically generated with medium confidence**

**2.5 Order detail**

**Graphical user interface, text, application

Description automatically generated**

**2.6 User list**

**Graphical user interface, text, application, email

Description automatically generated**

**2.7 Import and export excel**

**Graphical user interface, text, application

Description automatically generated**

**2.8 Send Mail**

**Graphical user interface, text, application

Description automatically generated**

# **CHƯƠNG 6 : HƯỚNG PHÁT TRIỂN**

## I. Nhận xét

**Ưu điểm:**

* Phần mềm có những chức năng cơ bản , đáp ứng phần nào nhu cầu người sử dụng.
* Phần mềm dễ hiểu , dễ sử dụng.
* Giao diện trực quan.
* Không yêu cầu người sử dụng có trình độ cao

**Nhược điểm:**

* Phần mềm chưa giải quyết những vấn đề cấp thiết.
* Chức năng còn đơn giản, chưa đầy đủ.

## Hướng phát triển:

Chương trình chỉ xây dựng ở mức sơ bộ nên hệ thống chỉ thể hiên chức năng cơ bản đáp ứng một phần nhu cầu sử dụng, từ đó có thể phát triển thêm các chức năng khác, xây dựng hệ thống chắc chắn và phát triển hơn.

## Tài liệu tham khảo:

1. React document: <https://reactjs.org/docs>
2. Redux document: <https://redux.js.org/>
3. ExpressJS: <https://expressjs.com/en/guide/routing.html>
4. Cellphone ecommerce. Link: <https://cellphones.com.vn/>