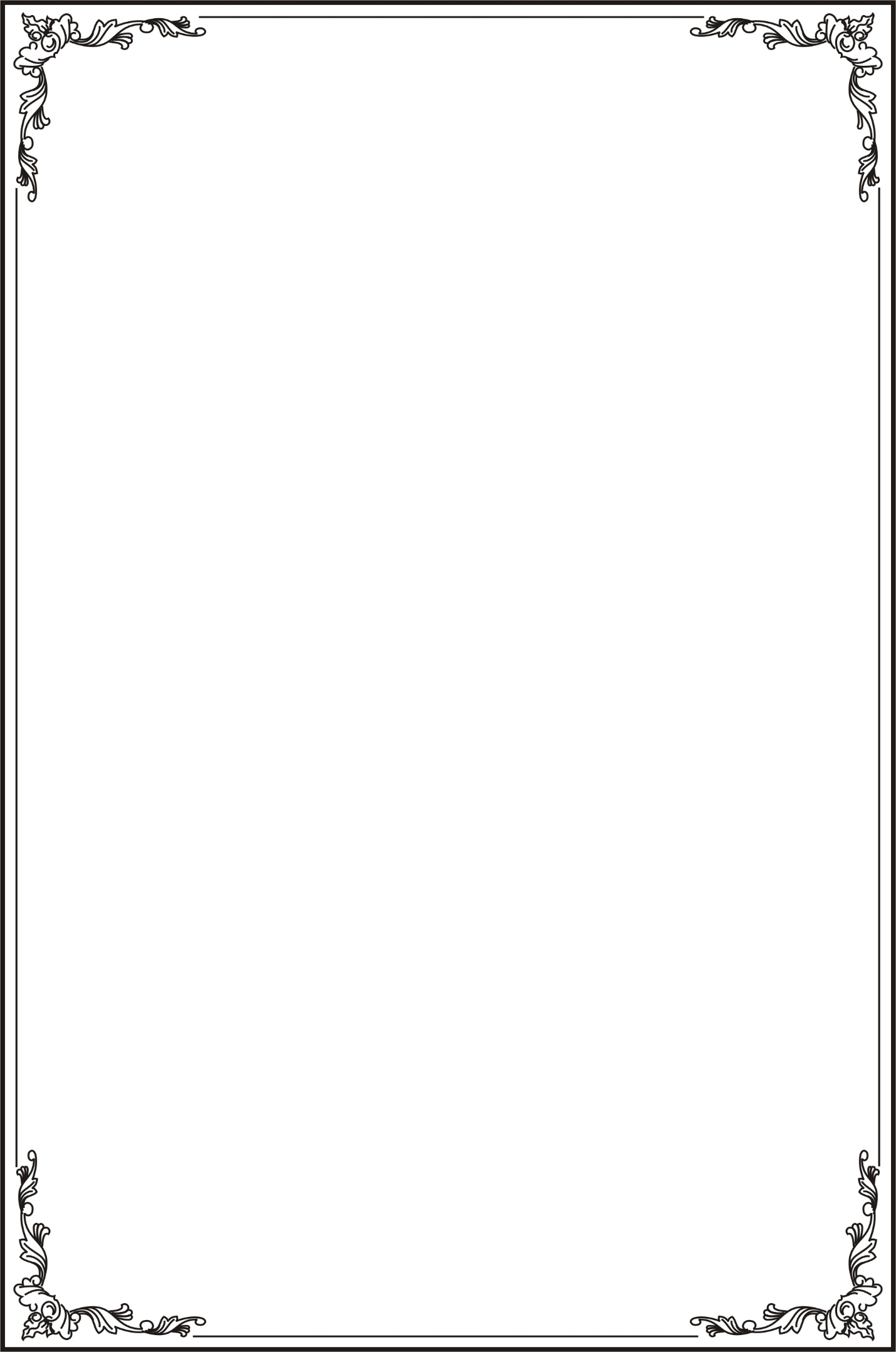
**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

-----🙞🙜🕮🙞🙜-----



**TIỂU LUẬN CHUYÊN NGÀNH**

**ĐỀ TÀI:** **WEBSITE BÁN THỨC ĂN NHANH SỬ DỤNG MERN STACK**

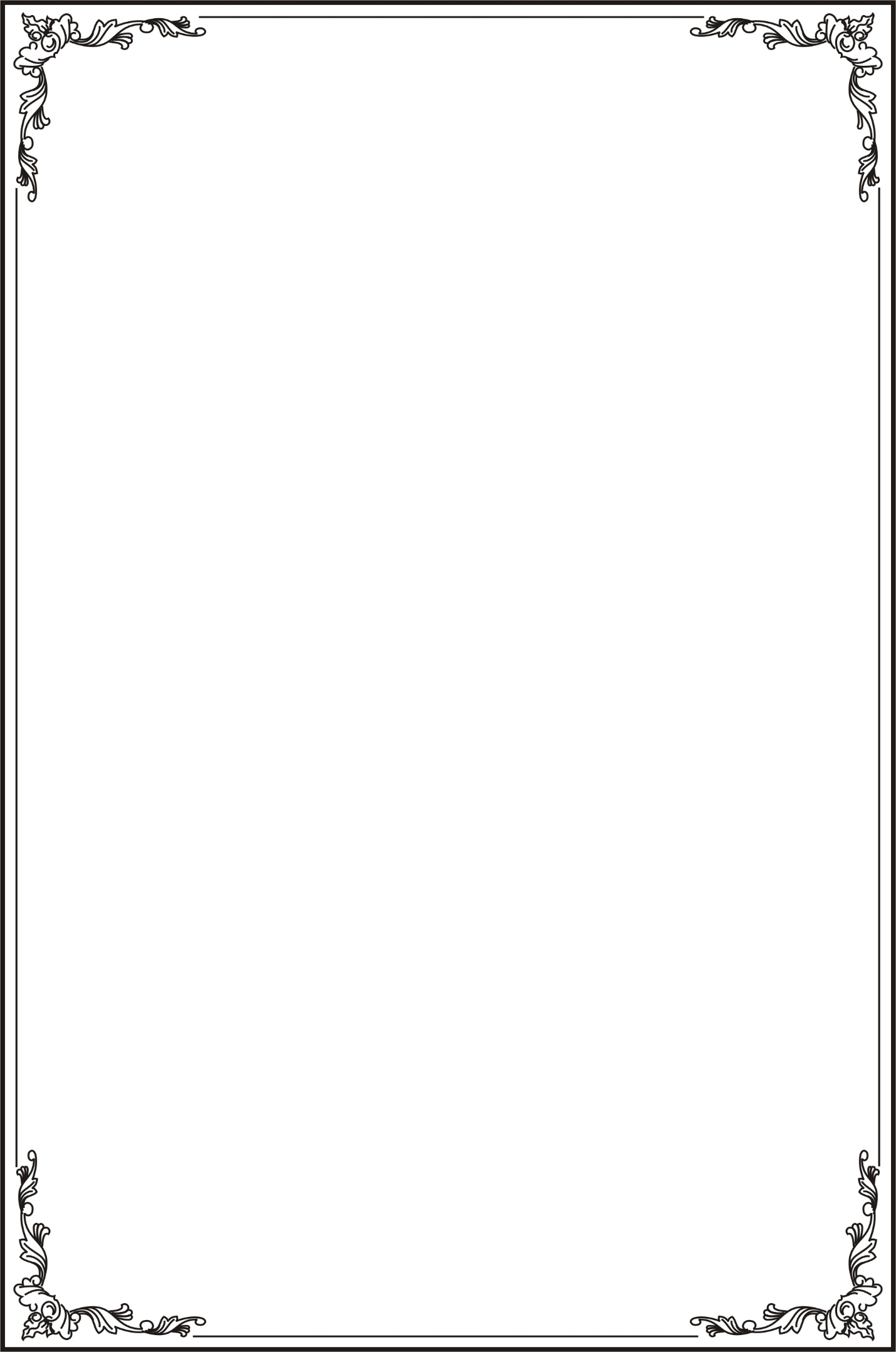
**Giảng viên hướng dẫn: Thầy Lê Vĩnh Thịnh**

**Sinh viên thực hiện: Lê Nguyễn Thanh Nhân 18110165**

**Phan Cao Cường 18110087**

**Đinh Tấn Tú 18119211**

*Hồ Chí Minh, ngày 26 tháng 12 năm 2021*

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

-----🙞🙜🕮🙞🙜-----

**TIỂU LUẬN CHUYÊN NGÀNH**

**ĐỀ TÀI: WEBSITE BÁN THỨC ĂN NHANH SỬ DỤNG MERN STACK**

**Giảng viên hướng dẫn: Thầy Lê Vĩnh Thịnh**

**Sinh viên thực hiện: Lê Nguyễn Thanh Nhân 18110165**

**Phan Cao Cường 18110087**

**Đinh Tấn Tú 18119211**

*Hồ Chí Minh, ngày 26 tháng 12 năm 2021*

***Nhận xét cúa giáo viên hướng dẫn:***

*...............................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................*

***Điểm Chữ ký của giảng viên***

***Nhận xét cúa giáo viên phản biện:***

*............................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................*

***Điểm Chữ ký của giảng viên***

**MỤC LỤC**

Trang

[DANH MỤC BẢNG BIỂU I](#_Toc91523980)

[DANH MỤC HÌNH ẢNH IV](#_Toc91523981)

[Chương I. KHẢO SÁT HIỆN TRẠNG VÀ XÁC ĐỊNH YÊU CẦU 3](#_Toc91523982)

[I.1. Mô tả về bài toán 3](#_Toc91523983)

[I.2. Ý nghĩa thực tiễn và ý nghĩa khoa học 3](#_Toc91523984)

[I.2.1. Ý nghĩa thực tiễn 3](#_Toc91523985)

[I.2.2. Ý nghĩa khoa học 4](#_Toc91523986)

[I.2.3. Mục tiêu dự án 4](#_Toc91523987)

[I.2.4. Công nghệ sử dụng 4](#_Toc91523988)

[I.2.5. Phạm vi dự án 4](#_Toc91523989)

[Chương II. CƠ SỞ LÝ THUYẾT 6](#_Toc91523990)

[II.1. Tổng quan về ReactJS 6](#_Toc91523991)

[II.1.1. ReactJS là gì Mô tả 6](#_Toc91523992)

[II.1.2. Thành phần cơ bản của ReactJS 6](#_Toc91523993)

[II.1.3. Lợi ích khi sử dụng ReactJS 7](#_Toc91523994)

[II.2. Tổng quan về MongoDB 7](#_Toc91523995)

[II.2.1. Định nghĩa về MongoDB 7](#_Toc91523996)

[II.2.2. Cơ chế hoạng động của MongoDB là gì ? 8](#_Toc91523997)

[II.2.3. Ưu điểm của MongoDB 8](#_Toc91523998)

[II.2.4. Nhược điểm của MongoDB 9](#_Toc91523999)

[II.3. Tổng quan về NodeJS 9](#_Toc91524000)

[II.3.1. NodeJS là gì ? 9](#_Toc91524001)

[II.3.2. Các đặc tính của NodeJS ? 10](#_Toc91524002)

[II.3.3. Ưu nhược điểm của NodeJS 11](#_Toc91524003)

[II.4. Tổng quan về Express 12](#_Toc91524004)

[II.4.1. Express là gì ? 12](#_Toc91524005)

[II.4.2. Tạo sao nên sử dụng Express ? 12](#_Toc91524006)

[II.4.3. Những tính năng của Express 14](#_Toc91524007)

[II.4.4. Phân biệt Express và NodeJS 14](#_Toc91524008)

[Chương III. ĐẶC TẢ YÊU CẦU HỆ THỐNG 16](#_Toc91524009)

[III.1. Mô tả hệ thống 16](#_Toc91524010)

[III.2. Yêu cầu chức năng 16](#_Toc91524011)

[III.2.1. Yêu cầu người dùng 16](#_Toc91524012)

[III.3. Yêu cầu phi chức năng 30](#_Toc91524013)

[III.3.1. Khả năng sử dụng 30](#_Toc91524014)

[III.3.2. Tính xác thực 30](#_Toc91524015)

[III.3.3. Hiệu suất 31](#_Toc91524016)

[III.3.4. Công cụ thực hiện 31](#_Toc91524017)

[III.3.5. Giao diện sử dụng 31](#_Toc91524018)

[III.4. Yêu cầu APIs 32](#_Toc91524019)

[III.4.1. Chức năng login với POST method 32](#_Toc91524020)

[III.4.2. Chức năng hiển thị danh sách địa chỉ của người dùng với GET method 32](#_Toc91524021)

[III.4.3. Chức năng đặt địa chỉ mặc định với PUT method 33](#_Toc91524022)

[III.4.4. Chức năng thêm mới địa chỉ với POST method 34](#_Toc91524023)

[Chương IV. THIẾT KẾ HỆ THỐNG 35](#_Toc91524024)

[IV.1. Kiến trúc hệ thống 35](#_Toc91524025)

[IV.1.1. Kiến trúc Front-end 35](#_Toc91524026)

[IV.1.2. Kiến trúc Back-end 36](#_Toc91524027)

[IV.2. ER Diagram 39](#_Toc91524028)

[IV.3. Activity Diagram 40](#_Toc91524029)

[IV.3.1. Đăng ký Activity Diagram 40](#_Toc91524030)

[IV.3.2. Đăng nhập Acitvity Diagram 41](#_Toc91524031)

[IV.3.3. Thêm sản phẩm vào giỏ hàng Activity diagram 42](#_Toc91524032)

[IV.3.4. Thanh toán Activity Diagram 43](#_Toc91524033)

[IV.4. Component Diagram 44](#_Toc91524034)

[IV.4.1. Backend 44](#_Toc91524035)

[IV.4.2. Frontend 44](#_Toc91524036)

[IV.5. Database Diagram 45](#_Toc91524037)

[IV.5.1. Mô tả các bảng 45](#_Toc91524038)

[IV.6. Thiết kế giao diện 50](#_Toc91524039)

[IV.6.1. Giao diện đăng nhập 50](#_Toc91524040)

[IV.6.2. Giao diện đăng ký 51](#_Toc91524041)

[IV.6.3. Giao diện trang chủ 51](#_Toc91524042)

[IV.6.4. Giao diện quản lý thông tin 52](#_Toc91524043)

[IV.6.5. Giao diện danh sách sản phẩm 52](#_Toc91524044)

[IV.6.6. Giao diện giỏ hàng 53](#_Toc91524045)

[IV.6.7. Giao diện checkout 54](#_Toc91524046)

[IV.6.8. Giao diện thanh toán VNPAY 55](#_Toc91524047)

[IV.6.9. Giao diện quản lý địa chỉ 55](#_Toc91524048)

[Chương V. THỰC THI HỆ THỐNG 56](#_Toc91524049)

[V.1. Công cụ và môi trường phát triển 56](#_Toc91524050)

[V.2. Công nghệ sử dụng 56](#_Toc91524051)

[V.3. Quản lý source code 57](#_Toc91524052)

[Chương VI. THỰC NGHIỆM, ĐÁNH GIÁ, PHÂN TÍCH KẾT QUẢ 58](#_Toc91524053)

[VI.1. Kế hoạch kiểm thử 58](#_Toc91524054)

[VI.2. Các mẫu Testcase 59](#_Toc91524055)

[VI.3. Kiểm thử WhiteBox 85](#_Toc91524056)

[VI.3.1. Chức năng Login 85](#_Toc91524057)

[VI.3.2. Chức năng UpdateUser 91](#_Toc91524058)

[VI.3.3. Chức năng Active Email 95](#_Toc91524059)

[VI.4. Đánh giá 101](#_Toc91524060)

[Chương VII. KẾT LUẬN 102](#_Toc91524061)

[VII.1. Đánh giá những kết quả đã thực hiện được 102](#_Toc91524062)

[VII.2. Đánh giá quá trình thực hiện và phân tích kết quả 102](#_Toc91524063)

[VII.2.1. Ưu điểm 102](#_Toc91524064)

[VII.2.2. Nhược điểm 103](#_Toc91524065)

[VII.3. Định hướng phát triển 103](#_Toc91524066)

# DANH MỤC BẢNG BIỂU

Trang

[Bảng 3. 1 Khách vãng lai 17](#_Toc91524081)

[Bảng 3. 2 Khách hàng 20](#_Toc91524082)

[Bảng 3. 3 Usecase đăng ký 21](#_Toc91524083)

[Bảng 3. 4 Usecase đăng nhập email 22](#_Toc91524084)

[Bảng 3. 5 Usecase đăng nhập bằng Gmail 23](#_Toc91524085)

[Bảng 3. 6 Usecase đăng nhập bằng Facebook 24](#_Toc91524086)

[Bảng 3. 7 Usecase Thêm sản phẩm vào giỏ hàng 25](#_Toc91524087)

[Bảng 3. 8 Usecase Quên mật khẩu 26](#_Toc91524088)

[Bảng 3. 9 Usecase Đăng xuất 27](#_Toc91524089)

[Bảng 3. 10 Usecase Cập nhật sản phẩm trong giỏ hàng 27](#_Toc91524090)

[Bảng 3. 11 Usecase Xóa sản phẩm trong giỏ hàng 28](#_Toc91524091)

[Bảng 3. 12 Usecase Order sản phẩm 29](#_Toc91524092)

[Bảng 3. 13 Usecase Group order 30](#_Toc91524093)

[Bảng 4. 1 Bảng user 46](#_Toc91450577)

[Bảng 4. 2 Bảng userDetail 46](#_Toc91450578)

[Bảng 4. 3 Bảng userRole 46](#_Toc91450579)

[Bảng 4. 4 Bảng product 47](#_Toc91450580)

[Bảng 4. 5 Bảng category 47](#_Toc91450581)

[Bảng 4. 6 Bảng cart 47](#_Toc91450582)

[Bảng 4. 7 Bảng cartItem 48](#_Toc91450583)

[Bảng 4. 8 Bảng order 48](#_Toc91450584)

[Bảng 4. 9 Bảng orderItem 49](#_Toc91450585)

[Bảng 4. 10 Bảng size 49](#_Toc91450586)

[Bảng 4. 11 Bảng topping 49](#_Toc91450587)

[Bảng 4. 12 Bảng token 50](#_Toc91450588)

[Bảng 6. 1 Testcase đăng nhập email 60](#_Toc91450608)

[Bảng 6. 2 Testcase đăng nhập bằng Google 62](#_Toc91450609)

[Bảng 6. 3 Testcase đăng nhập bằng Facebook 64](#_Toc91450610)

[Bảng 6. 4 Testcase thêm sản phẩm vào giỏ hàng 67](#_Toc91450611)

[Bảng 6. 5 Testcase hoạt động của giỏ hàng 70](#_Toc91450612)

[Bảng 6. 6 Testcase xóa sản phẩm ra khỏi giỏ hàng 73](#_Toc91450613)

[Bảng 6. 7 Testcase lưu đơn hàng sau đăng xuất 76](#_Toc91450614)

[Bảng 6. 8 Testcase cập nhật đơn hàng cũ với thông tin sản phẩm không đổi 81](#_Toc91450615)

[Bảng 6. 9 Testcase cập nhật lại đơn hàng cũ với cùng sản phẩm nhưng khác thông tin đi kèm 85](#_Toc91450616)

[Bảng 6. 10 TestCase cho từng nhánh 88](#_Toc91450617)

[Bảng 6. 11 TestCase cho từng nhánh 93](#_Toc91450618)

[Bảng 6. 12 Đồ thị dòng dữ liệu 93](#_Toc91450619)

[Bảng 6. 13 TestCase cho từng nhánh 97](#_Toc91450620)

[Bảng 6. 14 Đồ thị dòng dữ liệu 98](#_Toc91450621)

# DANH MỤC HÌNH ẢNH

Trang

[Hình 3. 1 Usecase tổng quát 20](#_Toc91524117)

[Hình 3. 2 Chức năng login với POST method 32](#_Toc91524118)

[Hình 3. 3 Chức năng hiển thị danh sách địa chỉ của người dùng với GET method 33](#_Toc91524119)

[Hình 3. 4 Chức năng đặt địa chỉ mặc định với PUT method 33](#_Toc91524120)

[Hình 3. 5 Chức năng thêm mới địa chỉ với POST method 34](#_Toc91524121)

[Hình 4. 1 Tổ chức folder Front-end 35](#_Toc91524122)

[Hình 4. 2 Mô hình Reac-Redux 36](#_Toc91524123)

[Hình 4. 3 Tổ chức folder backend 37](file:///F:\HKI_Nam4\TLCN\BaoCaoTLCN.docx#_Toc91524124)

[Hình 4. 4 Mô hình kiến trúc backend 38](file:///F:\HKI_Nam4\TLCN\BaoCaoTLCN.docx#_Toc91524125)

[Hình 4. 5 Mô hình ERD 39](#_Toc91524126)

[Hình 4. 6 Thanh toán activity diagram 40](#_Toc91524127)

[Hình 4. 7 Đăng nhập activity diagram 41](#_Toc91524128)

[Hình 4. 8 Thêm sản phẩm vào giỏ hàng activity diagram 42](#_Toc91524129)

[Hình 4. 9 Thanh toán activity diagram 43](#_Toc91524130)

[Hình 4. 10 Backend component diagram 44](#_Toc91524131)

[Hình 4. 11 Frontend component diagram 44](#_Toc91524132)

[Hình 4. 12 Mô hình Database Diagram 45](#_Toc91524133)

[Hình 4. 13 Giao diện đăng nhập 50](#_Toc91524134)

[Hình 4. 14 Giao diện đăng ký 51](#_Toc91524135)

[Hình 4. 15 Giao diện trang chủ 51](#_Toc91524136)

[Hình 4. 16 Giao diện quản lý thông tin 52](#_Toc91524137)

[Hình 4. 17 Giao diện danh sách sản phẩm 52](#_Toc91524138)

[Hình 4. 18 Giao diện giỏ hàng 53](file:///F:\HKI_Nam4\TLCN\BaoCaoTLCN.docx#_Toc91524139)

[Hình 4. 19 Giao diện checkout 54](#_Toc91524140)

[Hình 4. 20 Giao diện thanh toán VNPAY 55](#_Toc91524141)

[Hình 4. 21 Giao diện quản lý địa chỉ 55](#_Toc91524142)

[Hình 6. 1 Đồ thị luồng dữ liệu đăng ký 87](#_Toc91524143)

[Hình 6. 2 Đồ thị dòng dữ liệu 88](#_Toc91524144)

[Hình 6. 3 Kiểm tra vòng đời biến email 89](#_Toc91524145)

[Hình 6. 4 Kiểm tra vòng đời biến password 89](#_Toc91524146)

[Hình 6. 5 Kiểm tra vòng đời biến user 90](#_Toc91524147)

[Hình 6. 6 Kiểm tra vòng đời biến match 90](#_Toc91524148)

[Hình 6. 7 Đồ thị luồng dữ liệu update user 92](#_Toc91524149)

[Hình 6. 8 Kiểm tra vòng đời biến name 94](#_Toc91524150)

[Hình 6. 9 Kiểm tra vòng đời biến avatar 94](#_Toc91524151)

[Hình 6. 10 Kiểm tra vòng đời biến user 95](#_Toc91524152)

[Hình 6. 11 Đồ thị luồng dữ liệu active mail 96](#_Toc91524153)

[Hình 6. 12 Kiểm tra vòng đời biến data 99](#_Toc91524154)

[Hình 6. 13 Kiểm tra vòng đời biến userId 99](#_Toc91524155)

[Hình 6. 14 Kiểm tra vòng đời biến sub 100](#_Toc91524156)

[Hình 6. 15 Kiểm tra vòng đời biến result 100](#_Toc91524157)

**LỜI NÓI ĐẦU**

---------------------------------oOo----------------------------------

Ngày nay cuộc sống đang phát triển, mọi thứ cần phải khắc phục và sửa đổi để đáp ứng nhu cầu cuộc sống để ngày một hoàn thiện hơn.

Hiện nay với tình hình dịch bệnh đang diễn ra phức tạp, thì người dân các thành phố càng trở nên khó khăn trong việc nấu ăn hàng ngày, ví thế rất cần có một website hỗ trợ đặt hàng, giao thức ăn nhanh ngay trong ngày.

Tuy nhiên, theo tìm hiểu của nhóm tác giả thì hiện tại người dân còn khá khó trong việc tìm được website thực sự dễ sử dụng là khá khó khăn. Với những hiểu biết và kĩ năng có được sau khi học môn *Công nghệ phần mềm mới* thì nhóm tác giả quyết định chọn đề tài ***WEBSITE BÁN THỨC ĂN NHANH SỬ DỤNG MERN STACK*** mong muốn có thể phần nào giúp đỡ người dân trong các thành phố dễ dàng tiếp cận với các món ăn nhanh ngay sau đặt hàng.

Với sự hướng dẫn của Thầy Lê Vĩnh Thịnh, nhóm tác giả đã cố gắng nỗ lực để hoàn thành đồ án. Tuy nhiên sai xót là điều không thể tránh khỏi, nhóm mong muốn nhận được sự đóng góp ý kiến của thầy cô bạn bè để đồ án được hoàn chỉnh hơn.

**Nhóm sinh viên thực hiện**

# KHẢO SÁT HIỆN TRẠNG VÀ XÁC ĐỊNH YÊU CẦU

## Mô tả về bài toán

Trong thời đại hiện nay, cùng với sự phát triển của công nghệ thông tin thì nhu cầu của con người cũng ngày càng gia tăng. Trong thời điểm hiện tại, khi mà dịch bệnh Covid-19 đang tàn phá, thì việc một người có thể ra đường mua đồ ăn rất là khó khăn. Đó là lí do TNC – Ý tưởng về một website bán thức ăn nhanh được hình thành.

Trang web được xây dựng trên nền tảng web – nền tảng thân thiện với người dùng nhất, giúp người dùng có thể truy cập một cách dễ dàng và thực hiện đặt đồ ăn một cách thuận tiện và nhanh chóng, giúp tiết kiệm thời gian và kinh phí. Nhất là trong thời kì dịch bệnh Covid-19 đang diễn biến phức tạp, một ứng dụng giúp người dùng không phải ra đường vẫn có thể mua được đồ ăn ưa thích giúp phần nào giải quyết được vấn đề giãn cách xã hội, góp phần đẩy lùi dịch bệnh.

Thời gian gần đây, các công nghệ liên quan đến JavaScript đang phát triển một cách mạnh mẽ. Giờ ta hoàn toàn có thể xây dựng một trang web thuần bằng JavaScript. Nắm bắt được xu hướng này, đề tài mà nhóm đã chọn là Xây dựng một trang web bán thức ăn nhanh sử dụng MERN Stack.

## Ý nghĩa thực tiễn và ý nghĩa khoa học

### Ý nghĩa thực tiễn

• Dự án giúp người dùng có thể đặt hàng qua internet một cách dễ dàng và nhanh chóng mà không phải ra khỏi nhà

• Xây dựng mô hình bán hàng qua mạng là mô hình được áp dụng nhiều trong thực tế giúp nhóm thực hiện có them kiến thức thực tế về nghiệp vụ của đề tài này

### Ý nghĩa khoa học

• Vì MERN STACK là công nghệ của JavaScript – một công nghệ đang phát triển gần đây nên có những công cụ hỗ trợ việc xây dựng một trang web một cách nhanh chóng và tiện lợi, giúp nhóm thực hiện học hỏi thêm về các công nghệ mới đang được ưa chuộng hiện nay

### Mục tiêu dự án

• Nghiên cứu về MERN STACK (MongoDB – Express – ReactJS -NodeJs) để phát triển website

• Nghiên cứu về nghiệp vụ của một mô hình bán hàng và đặt hàng qua mạng và nghiệp vụ quản lý chung

• Phát triển website bán thức ăn nhanh TNC bằng MERN STACK

* Phát triển web API để vận chuyển thông tin giữa client và Database

### Công nghệ sử dụng

• Front-end: ReactJS

• Back-end: NodeJS, Express framework

• Database: MongoDB

### Phạm vi dự án

* Tìm hiểu ReactJS để phát triển giao diện cho website
* Tìm hiểu lifecycle, cách thức hoạt động và các component và framework được ưa thích của ReactJS
* Tìm hiểu về nghiệp vụ bán hàng, giao hàng online và nghiệp vụ quản lý chung
* Tìm hiểu và phân tích các chức năng trong thực tế của một hệ thống bán thức ăn nhanh
* Đánh giá và ước lượng những chức năng khả thi để đưa vào project. Từ đó triển khai các bước sau:
* Thiết kế sơ đồ usecase
* Thiết kế kiến trúc cho chương trình
* Thiết kế database
* Phát triển các website đặt hàng và trang quản lý chung sử dụng MERN STACK:
* Phát triển các chức năng được phân theo module như:
* Header
* Footer
* Cart
* Checkout
* Talke
* Payment
* Những chức năng cho user:
* Login/logout
* Xem sản phẩm
* Quản lý giỏ hàng
* Checkout
* Các chức năng nâng cao khác:
* Thanh toán bằng VNPay, Paypal
* Đặt hàng chung
* Phát triển web API để vận chuyển dữ liệu giữa client và web database
* Tạo API sử dụng NodeJS-Express framework
* Tạo các function xử lý các câu lệnh phức tạp
* Những phạm vi khác:
* Deploy database lên VPS
* Deploy Backend lên VPS

# CƠ SỞ LÝ THUYẾT

## Tổng quan về ReactJS

### ReactJS là gì Mô tả

Reactjs là một thư viện Javascript mã nguồn mở hỗ trợ xây dựng các thành phần giao diện nhanh gọn và tiện lợi. Bình thường các lập trình viên sẽ nhúng javascript vào code HTML thông qua các attribute như AngularJS nhưng với Reactjs làm việc như một thư viện cho phép nhúng HTML vào javascript thông qua JSX.Qua đó bạn có thể dễ dàng lồng các đoạn HTML vào trong JSX làm cho các component dễ hiểu và dễ sử dụng hơn.

Trong Reactjs thường dùng javascript để thiết kế bố cục cho trang web, nhưng nhược điểm là cấu trúc khá là khó. Thay vào đó sử dụng JSX và nhúng các đoạn HTML vào javascript, ta thấy cú pháp dễ hiểu hơn và JSX cũng có thể tối ưu code khi biên soạn. Vừa dễ cho người lập trình mà vừa tiện cho việc biên dịch. Trong ví dụ trên các bạn sẽ thấy người ta viết HTML trong JSX, khi đọc code cũng dễ hiểu hơn là đọc Javascript.Yêu cầu chức năng

### Thành phần cơ bản của ReactJS

Thành phần cơ bản của React được gọi là **components**. Syntax để viết HTML sử dụng Javascript để render. Bạn có thể tạo ra một component bằng các gọi phương thức createClass của đối tượng React, điểm bắt đầu khi tiếp cận với thư viện này. Có thể lồng nhiều component vào nhau thông qua lệnh return của phương thức render.

Trong một chương trình thì có rất nhiều các component, để đơn giản việc quản lý các component đó người ta sử dụng redux, **redux giống như 1 cái kho chứa các component** và khi dùng component nào thì chỉ cần gọi nó ra. Virtual DOM không được tạo ra bởi Reactjs nhưng lại được sử dụng rất nhiều. Đây là một chuẩn của W3C được dùng để truy xuất code HTML hoặc XML.

Các **Virtual DOM** sẽ được tạo ra khi chạy chương trình, đó là nơi chưa các component. Sử dụng DOM sẽ tiết kiệm được hiệu suất làm việc, khi có thay đổi gì Reactjs đều tính toán trước và việc còn lại chỉ là thực hiện chúng lên DOM.

### Lợi ích khi sử dụng ReactJS

Làm gì cũng phải có nguyên nhân, người dùng sử dụng Reactjs ngày càng nhiều thì Reactjs phải mang lại những lợi ích tốt.

* ReactJS giúp cho việc viết các đoạn code Javascript sẽ trở nên dễ dàng hơn vì nó sử dụng một cú pháp đặc biệt đó chính là cú pháp JSX. Thông qua JSX cho phép nhúng code HTML và Javascript.
* ReactJS cho phép Developer phá vỡ những cấu tạo UI phức tạp thành những component độc lập. Dev sẽ không phải lo lắng về tổng thể ứng dụng web, giờ đây Developer dễ dàng chia nhỏ các cấu trúc UI/UX phức tạp thành từng component đơn giản hơn.
* Đi kèm với ReactJS là rất nhiều các công cụ phát triển giúp cho việc  debug code một cách dễ dàng hơn.
* Một trong những ưu điểm nữa của ReactJS đó là sự thân thiện với SEO. Hầu như các JS [Frameworks](https://blog.itnavi.com.vn/framework-java/) không thân thiện với các tìm kiếm mặc dù đã được cải thiện nhiều nhưng dưới sự hỗ trợ của các render dữ liệu trả về dưới dạng web page giúp cho SEO chuẩn hơn.

## Tổng quan về MongoDB

### Định nghĩa về MongoDB

MongoDB là một dạng phần mềm cơ sở dữ liệu sử dụng mã nguồn mở NoSQL. Nó có thể hỗ trợ trên nhiều nền tảng khác nhau và được thiết kế với mục đích hướng đến đối tượng. MongoDB hoạt động dựa vào các khái niệm Collection và Document. Đồng thời, nó có hiệu suất cao cùng với tính khả dụng tốt và dễ dàng mở rộng.

Các Collection trong MongoDB có cấu trúc cực kỳ linh hoạt. Điều này cho phép dữ liệu không cần thiết phải tuân theo bất kỳ một dạng cấu trúc nào. Vì thế, MongoDB có thể lưu trữ những dữ liệu có cấu trúc đa dạng và phức tạp. Dữ liệu trong MongoDB được lưu bằng định dạng kiểu JSON.

### Cơ chế hoạng động của MongoDB là gì ?

Nguyên tắc hoạt động của MongoDB là dưới một tiến trình dịch vụ ngầm và mở một cổng (mặc định là cổng 27017), để có thể tiếp nhận các yêu cầu truy vấn, thao tác; sau đó tiến hành xử lý.

Mỗi bản ghi của MongoDB (document) được gắn một trường có tên “\_id” nhằm xác định tính duy nhất của bản ghi. Có thể hiểu id này như tên gọi của một bản ghi và dùng phân biệt chúng với các bản ghi khác. Đồng thời, nó còn được sử dụng cho mục đích truy vấn hoặc tìm kiếm thông tin. Trường dữ liệu “\_id” được tự động đánh chỉ mục (index) để đảm bảo tốc độ truy vấn đạt hiệu suất tối ưu.

Mỗi truy vấn dữ liệu đều được ghi đệm lên bộ nhớ RAM nên các truy vấn sau đó sẽ diễn ra nhanh hơn. Bởi nó không cần đọc dữ liệu từ ổ cứng.

Khi thực hiện thêm, xóa hay sửa bản ghi thì MongoDB đều mất 60s để ghi các dữ liệu được thay đổi từ RAM xuống ổ cứng. Điều này nhằm mục đích đảm bảo hiệu suất mặc định của chương trình.

### Ưu điểm của MongoDB

MongoDB mang đến cho người dùng khá nhiều lợi ích:

* Linh hoạt trong lưu trữ các kích cỡ dữ liệu khác nhau. Nhờ chúng được lưu dưới dạng JSON nên bạn thoải mái chèn bất kỳ thông tin nào tùy theo nhu cầu sử dụng.
* Tiết kiệm thời gian trong việc kiểm tra sự tương thích về cấu trúc khi thêm, xóa hoặc cập nhật dữ liệu. Nhờ MongoDB không có sự ràng buộc trong một khuôn khổ, quy tắc nhất định nào.
* Bạn dễ dàng mở rộng hệ thống thông qua việc thêm node vào cluster. Cụm các node này đóng vai trò như thư viện chứa các dữ liệu giao tiếp với nhau.
* Tốc độ truy vấn của MongoDB nhanh hơn so với RDBMS do toàn bộ dữ liệu truy vấn đã được ghi đệm lên bộ nhớ RAM. Nhờ thế, những lượt truy vấn sau sẽ được rút ngắn thời gian vì chúng không cần đọc từ ổ cứng.
* Trường dữ liệu “\_id” (đại diện cho giá trị duy nhất trong mỗi document) được tự động đánh chỉ mục nên hiệu suất luôn đạt mức cao nhất.

### Nhược điểm của MongoDB

Bên cạnh các ưu điểm, MongoDB vẫn còn tồn tại một số điểm hạn chế mà bạn cần chú ý khi cài đặt và sử dụng:

* Vì dữ liệu không bị ràng buộc nên trong quá trình sử dụng, bạn cần cẩn thận trong mọi thao tác nhằm tránh xảy ra những điều không mong muốn, làm ảnh hưởng đến dữ liệu.
* Chương trình MongoDB tiêu tốn khá nhiều dung lượng bộ nhớ do dữ liệu được lưu dưới dạng key và value. Bên cạnh đó, một số collection chỉ có sự khác biệt về value nên việc lặp lại key là điều khó tránh khỏi. Điều này dẫn đến thừa dữ liệu.
* Thông thường, thời gian để dữ liệu chuyển đổi từ RAM xuống ổ cứng khoảng 60s nên nguy cơ bị mất dữ liệu nếu xảy ra mất điện là điều có thể xảy ra.

## Tổng quan về NodeJS

### NodeJS là gì ?

NodeJS là một mã nguồn được xây dựng dựa trên nền tảng Javascript V8 Engine, nó được sử dụng để xây dựng các ứng dụng web như các trang video clip, các forum và đặc biệt là trang mạng xã hội phạm vi hẹp. NodeJS là một mã nguồn mở được sử dụng rộng bởi hàng ngàn lập trình viên trên toàn thế giới.

NodeJS có thể chạy trên nhiều nền tảng hệ điều hành khác nhau từ WIndow cho tới Linux, OS X nên đó cũng là một lợi thế. 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.

Khi nói đến NodeJS thì phải nghĩ tới vấn đề Realtime. Realtime ở đây chính là xử lý giao tiếp từ client tới máy chủ theo thời gian thực. Giống như khi bạn lướt Facebook thì mỗi khi bạn comment hay like một topic nào đó thì ngay lập tức chủ topic và những người đã comment trên đó sẽ nhận được thông báo là bạn đã comment.

Nếu ban nghĩ Facebook đang sử dụng [Ajax](https://freetuts.net/hoc-php/hoc-ajax) thì bạn đã sai rồi nhé, nếu họ sử dụng Ajax thì Server họ sẽ chết ngay lập tức bởi các request gửi lên với số lượng tính theo tỉ :3. Còn nếu bạn hỏi Facebook có phải sử dụng NodeJS không thì mình xin trả lời là mình không biết nhé :D Vì mình không phải là nhà phát triển Facebook.

### Các đặc tính của NodeJS ?

Qua phần tìm hiểu **NodeJS là gì** mình có giới thiệu một đặc tính rất quan trọng đó là Realtime, tuy nhiên vẫn còn khá nhiều đặc tính mà bạn cần phải biết trước khi học NodeJS.

* **Không đồng bộ**: Tất cả các API của NodeJS đều không đồng bộ (none-blocking), nó chủ yếu dựa trên nền của NodeJS Server và chờ đợi Server trả dữ liệu về. Việc di chuyển máy chủ đến các API tiếp theo sau khi gọi và cơ chế thông báo các sự kiện của Node.js giúp máy chủ để có được một phản ứng từ các cuộc gọi API trước (Realtime).
* **Chạy rất nhanh**: NodeJ được xây dựng dựa vào nền tảng V8 Javascript Engine nên việc thực thi chương trình rất nhanh.
* **Đơn luồng nhưng khả năng mở rộng cao**: Node.js sử dụng một mô hình luồng duy nhất với sự kiện lặp. cơ chế tổ chức sự kiện giúp các máy chủ để đáp ứng một cách không ngăn chặn và làm cho máy chủ cao khả năng mở rộng như trái ngược với các máy chủ truyền thống mà tạo đề hạn chế để xử lý yêu cầu. Node.js sử dụng một chương trình đơn luồng và các chương trình tương tự có thể cung cấp dịch vụ cho một số lượng lớn hơn nhiều so với yêu cầu máy chủ truyền thống như Apache HTTP Server.
* **Không đệm**: NodeJS không đệm bất kì một dữ liệu nào và các ứng dụng này chủ yếu là đầu ra dữ liệu.
* **Có giấy phép**: NodeJS đã được cấp giấy phép bởi [MIT License](https://freetuts.net/nodejs-la-gi-584.html).

### Ưu nhược điểm của NodeJS

* Ưu điểm:
* Có tốc độ xử lý nhanh nhờ cơ chế xử lý bất đồng bộ (non-blocking). Bạn có thể dễ dàng xử lý hàng ngàn kết nối trong khoảng thời gian ngắn nhất.
* Giúp bạn dễ dàng mở rộng khi có nhu cầu phát triển website.
* Nhận và xử lý nhiều kết nối chỉ với một single-thread. Nhờ đó, hệ thống xử lý sẽ sử dụng ít lượng RAM nhất và giúp quá trình xử Nodejs lý nhanh hơn rất nhiều.
* Có khả năng xử lý nhiều Request/s cùng một lúc trong thời gian ngắn nhất.
* Có khả năng xử lý hàng ngàn Process cho hiệu suất đạt mức tối ưu nhất.
* Phù hợp để xây dựng những ứng dụng thời gian thực như các ứng dụng chat, mạng xã hội …
* Nhược điểm:
* Nodejs gây hao tốn tài nguyên và thời gian. Nodejs được viết bằng C++ và JavaScript nên khi xử lý cần phải trải qua một quá trình biên dịch. Nếu bạn cần xử lý những ứng dụng tốn tài nguyên CPU thì không nên sử dụng Nodejs.
* Nodejs so với các ngôn ngữ khác như PHP, Ruby và Python sẽ không có sự chênh lệch quá nhiều. Nodejs có thể sẽ phù hợp với việc phát triển ứng dụng mới. Tuy nhiên khi xây dựng và triển khai dự án quan trọng thì Nodejs không phải là sự lựa chọn hoàn hảo nhất.

## Tổng quan về Express

### Express 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](https://itnavi.com.vn/blog/api-la-gi/) 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.

### Tạo sao nên sử dụng Express ?

#### Sự phổ biến của Javascript

Javascript là một ngôn ngữ lập trình được sử dụng rộng rãi hiện nay, dễ đọc và được hỗ trợ ở mọi nơi. Mà Expressjs lại hỗ trợ Javascript, do đó nếu bạn đã biết đến Javascript thì chắc chắn việc lập trình bằng Express.js là vô cùng đơn giản. Ngay cả những người mới bắt đầu tham gia vào lĩnh vực phát triển web này cũng có thể sử dụng Expressjs.

Javascript là ngôn ngữ lập trình dễ học với cả những người không có bất kỳ kiến thức gì về ngôn ngữ lập trình khác. Chính vì tính phổ biến, dễ học và dễ sử dụng này mà Express.js cho phép các tài năng trẻ tham gia và đạt được nhiều thành công trong phát triển ứng dụng web.

#### Hỗ trợ xây dựng website một cách nhanh chóng

Thời gian chính là tài sản quý giá của bất kỳ doanh nghiệp nào. Hơn thế, nhiều lập trình viên còn phải chịu áp lực xây dựng các ứng dụng web một cách hiệu quả trong thời gian ngắn. Nhưng để thực hiện tốt công việc này cần rất nhiều thời gian và Express.js chính là công cụ cứu cánh cho các nhà lập trình.

Express.js có thể giúp làm giảm một nửa thời gian viết mã mà vẫn xây dựng lên các ứng dụng web hiệu quả. Không chỉ trợ giúp về mặt thời gian Expressjs còn làm giảm những áp lực cần thiết để xây dựng với sự trợ giúp của các tính năng khác nhau của nó. Express js còn cung cấp một phần mềm trung gian đảm nhận nhiệm vụ đưa ra các quyết định để phản hồi chính xác những yêu cầu của khách hàng.

Nếu không có Express.js, các lập trình viên phải viết mã code riêng để xây dựng nên thành phần định tuyến. Đó là một công việc vô cùng tẻ nhạt và tốn thời gian. Express.js đã giúp cho công việc này trở nên đơn giản và hiệu quả hơn rất nhiều.

#### Express.js hoàn toàn miễn phí

Một trong những yếu tố để tổ chức, doanh nghiệp nên sử dụng công cụ, ứng dụng nào chính là chi phí. Với một ứng dụng web với nhiều tính năng tuyệt vời như vậy mà lại không tốn một chút kinh phí nào thì không có lý do gì để không sử dụng nó.

### Những tính năng của Express

Phát triển máy chủ nhanh chóng: Expressjs cung cấp nhiều tính năng dưới dạng các hàm để dễ dàng sử dụng ở bất kỳ đâu trong chương trình. Điều này đã loại bỏ nhu cầu viết mã từ đó tiết kiệm được thời gian.

Phần mềm trung gian Middleware: Đây là phần mềm trung gian có quyền truy cập vào cơ sở dữ liệu, yêu cầu của khách hàng và những phần mềm trung gian khác. Phần mềm Middleware này chịu trách nhiệm chính cho việc tổ chức có hệ thống các chức năng của Express.js.

Định tuyến - Routing: Express js cung cấp cơ chế định tuyến giúp duy trì trạng thái của website với sự trợ giúp của URL.

Tạo mẫu - Templating: Các công cụ tạo khuôn mẫu được Express.js cung cấp cho phép các nhà xây dựng nội dung động trên các website bằng cách tạo dựng các mẫu HTML ở phía máy chủ.

Gỡ lỗi - Debugging: Để phát triển thành công các ứng dụng web không thể thiết đi việc gỡ lỗi. Giờ đây với Expressjs việc gỡ lỗi đã trở nên dễ dàng hơn nhờ khả năng xác định chính xác các phần ứng dụng web có lỗi.

### Phân biệt Express và NodeJS

|  |  |
| --- | --- |
| **Express.js** | **Node.js** |
| Là framework của Node.js, sử dụng để xây dựng phần phụ trợ của ứng dụng web. | Được sử dụng để xây dựng cả frontend và backend của ứng dụng web. |
| Được viết bằng một ngôn ngữ lập trình duy nhất là Javascript.,... | Được viết bằng nhiều ngôn ngữ lập trình khác nhau như C/C++, |
| Là một framework | Không phải là framework |
| Để sử dụng Expressjs các lập trình viên cần cài đặt Expressjs cùng Node.js. | Các lập trình viên chỉ cần cài đặt Node.js trên thiết bị của mình để có thể sử dụng |
| Được sử dụng để xây dựng nên các ứng dụng phía máy chủ trên Node.js. | Được sử dụng để phát triển lên các ứng dụng mạng và phía máy chủ. |
| Express js phù hợp với các dự án có quy mô nhỏ. | Nodejs được sử dụng cho các dự án có quy mô lớn. |
| Express js chỉ được sử dụng ở phía máy chủ. | Node.js có thể sử dụng được cả phía máy chủ và máy khách. |
| Express.js tương thích với tất cả các hệ điều hành tương thích với Node.js. | Node.js tương thích với tất cả các hệ điều hành chính. |
| Cung cấp các thành phần định tuyến và có phần mềm trung gian để hỗ trợ giúp phát triển ứng dụng web một cách dễ dàng hơn. | Cung cấp nhiều tính năng cho các nhà phát triển trong xây dựng một ứng dụng web. |
| Express.js chỉ hỗ trợ ngôn ngữ JavaScript. | Node.js hỗ trợ nhiều ngôn ngữ khác nhau như: TypeScript, CoffeeScript và Ruby. |
| Nó được sử dụng bởi IBM, PayPal, Fox Sports,… | Nó được sử dụng bởi LinkedIn, PayPal, Walmart, Uber,… |

Bảng 2. Phân biệt Express và NodeJS

# ĐẶC TẢ YÊU CẦU HỆ THỐNG

## Mô tả hệ thống

Một hệ thống bán thức ăn nhanh online sẽ có cấu trúc như sau:

* Khách hàng có thể xem và tìm kiếm các món hàng trên trang web, xem được các chi tiết sản phẩm của món hàng đó, xem các đánh giá về món hàng đó sau đó mới quyết định mua sản phẩm nào
* Khi khách hàng muốn mua sản phẩm nào hay ưa thích sản phẩm nào thì cần phải đăng nhập với tài khoản của mình, sau đó mới có thể thực hiện chức năng them vào giỏ hàng để lần đăng nhập sau có thể nhìn thấy các lựa chọn đã chọn
* Khách hàng có thể quản lý giỏ hàng của riêng mình
* Khách hàng có thể quản lý các thông tin cá nhân và các địa chỉ mà mình hay sử dụng để thuận tiện cho việc đặt hàng
* Khách hàng có thể thanh toán các món hàng đã chọn trong giỏ hàng bằng các phương thức khác nhau
* Khách hàng có thể đặt đơn online để shipper giao hàng hoặc có thể đặt đơn và đến cửa hàng để lấy đơn
* Khách hàng có thể liên hệ với nhân viên khi gặp khó khăn cần hỗ trợ
* Khách hàng có thể đặt hàng chung với nhau bằng tính năng Group Order
* Khách hàng có thể theo dõi các đơn hàng đã mua, hủy đơn hàng đó
* Khách hàng có thể theo dõi các chương trình khuyến mãi, tiến trình đơn hàng quan phần thông báo của ứng dụng
* Back end

## Yêu cầu chức năng

### Yêu cầu người dùng

Với các yêu cầu đã thu thập được, nhóm thực hiện đã xây dựng ra một website hướng tới các đối tượng sử dụng như đã tìm hiểu như trên.

#### Tác nhân của hệ thống

Dựa vào mô tả vấn đề có thể xác định được các tác nhân của hệ thống như sau:

* **Khách vãng lai**: người dùng này có thể sử dụng các tính năng cơ bản nhất bao gồm xem danh sách các sản phẩm của cửa hàng, tìm kiếm sản phầm, lọc sản phẩm theo các điều kiện,…
* **Khách hàng**: người dùng này bắt buộc phải có tài khoản để có thể các sử dụng thêm các tính năng nâng cao hơn như quản lý giỏ hàng, đặt hàng, quản lý thông tin cá nhân, quản lý đơn đã mua

**Chức năng chính:**

* **Khách vãng lai:**

|  |  |  |
| --- | --- | --- |
|  | Tên chức năng | Mô tả |
| 1 | Xem danh sách sản phẩm | Xem danh sách các sản phẩm có thể mua cũng như các thông tin chi tiết của sản phẩm |
| 2 | Tìm kiếm sản phẩm | Khách có thể tìm kiếm các sản phẩm theo tên của sản phẩm đó |
| 3 | Lọc sản phẩm theo điều kiện | Khách hàng có thể lọc các kết quả tìm kiếm dựa theo các điều kiện đã cho |
| 4 | Chat với quản trị viên | Khách hàng có thể tương tác với quản tị viên thông quan Messenger |

Bảng 3. Khách vãng lai

* **Khách hàng:**

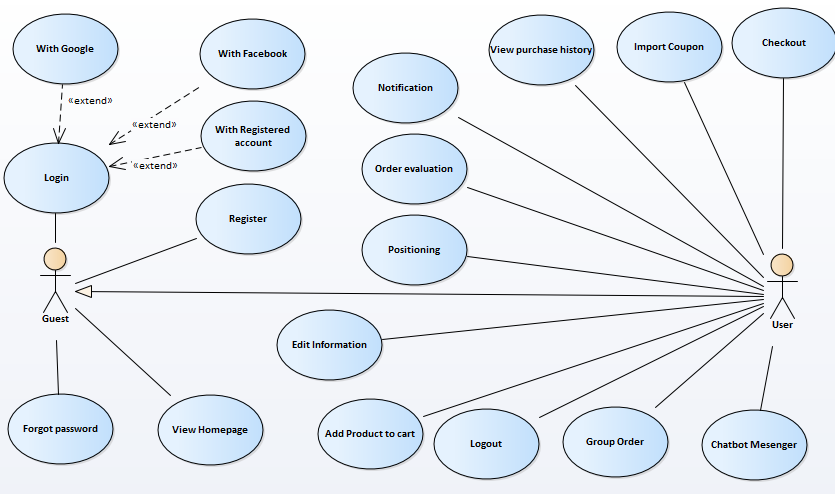
Khách hàng có những chức năng như khách vãng lai và các chức năng khác như sau:

|  |  |  |
| --- | --- | --- |
|  | Tên chức năng | Mô tả |
| 1 | Đăng nhập | Đăng nhập vào hệ thống bằng tài khoản đã đăng ký |
| 2 | Đăng nhập bằng Facebook | Đăng nhập vào hệ thống bằng tài khoản Facebook |
| 3 | Đăng nhập bằng Google | Đăng nhập vào hệ thống bằng tài khoản Google |
| 4 | Quản lý địa chỉ | Khách hàng có thể thêm địa chỉ mới, chỉnh sửa địa chỉ cũ, xóa các địa chỉ không cần thiết. Ngoài ra khách hàng có thể đặt cho 1 địa chỉ bất kì là mặc định cho tất cả các đơn hàng được thanh toán sau đó.  Khách hàng có thể tìm địa chỉ dựa vào việc cung cấp các thông tin như tỉnh, huyện, thị trấn, số nhà hoặc có thể chọn trên bản đồ |
| 5 | Quản lý giỏ hàng | Khách hàng sau khi chọn được món hàng ưng ý có thể thêm vào giỏ hàng để lưu lại món hàng đó. Ngoài ra có thể xóa các món hàng không cần thiết và chỉnh sửa thông tin của sản phẩm bên trong giỏ hàng tùy thích  Có 2 loại đơn hàng mà khách có thể đặt:   * **Đơn hàng online**: Đơn hàng này sẽ được cửa hàng tiếp nhận và sẽ ship đến địa chỉ của khách hàng. Với lại đơn hàng này khách hàng phải cung cấp địa chỉ nhận hàng và phải chịu phí vận chyển * **Đơn hàng take-way**: Đơn hàng này được cửa hàng tiếp nhận và thông báo cho khách hàng khi đơn được hoàn thành và khách hàng sẽ đến cửa hàng chỉ cần cung cấp mã QR tương ứng để nhận hàng. Với loại đơn hàng này, khách hàng không phải cung cấp địa chỉ nhận hàng và không phải trả phí vận chuyển |
| 6 | Đặt hàng chung | Ngoài việc đặt hàng thông thường, khách hàng hoàn toàn có thể đặt hàng chung với các khách hàng khác bằng cách chia sẻ mã mời. Mỗi khi khách hàng khác thêm sản phầm mới thì sản phẩm đó sẽ được quản lý bởi người chia sẻ và có thể thanh toán chung. Có thể đặt cả 2 loại đơn hàng như đã nêu trên |
| 7 | Thanh toán | Khách hàng có thể lựa chọn một trong 3 cách thanh toán sau đầy:   * **Thanh toán khi nhận hàng**: Khách hàng sẽ thanh toán giá trị đơn hàng các phí phát sinh khi nhận hàng * **Thanh toán bằng VNPAY**: Khách hàng sẽ thanh toán thông qua dịch vụ VNPay * **Thanh toán bằng PayPal**: Khách hàng sẽ thanh toán thông qua PayPal |
| 8 | Quản lý thông tin cá nhân | Khách hàng quản lý các thông tin cá nhân như tên, số điện thoại, email,…. Khách hàng cần phải cung cấp thông tin thật để đảm bảo giao dịch thành công |
| 9 | Quản lý các đơn hàng đã mua | Khách hàng có thể theo dõi lịch sử mua hàng và lọc đơn hàng theo các trạng thái đơn hàng.  Các trạng thái đơn hàng có thể có:   * Đợi xác nhận: Đây là trạng thái đơn hàng đã được khởi tạo nhưng chưa được quản trị viên nhà hàng xác nhận là đã nhận được * Đang thực hiện: Đây là trạng thái đơn hàng đã được xác nhận và đang được thực hiện * Đang giao: Đây là trạng thái đơn hàng đã được hoàn thành và đang được shipper giao đến khách hàng * Thành công: Đây là trạng thái thái đơn hàng được hoàn thành khi đã giao thành công với đơn hàng online và khách đã lấy hàng với đơn takeaway * Đã hủy: Đây là trạng thái đơn hàng bị khách hàng hủy thành công |
| 10 | Theo dõi đơn hàng | Khách hàng có thể theo dõi các đơn hàng đã được khởi tạo, mỗi khi trạng thái đơn hàng thay đổi sẽ có thông báo gửi đến khách hàng |
| 11 | Hủy đơn hàng | Khách hàng có thể hủy đơn hàng đã được tạo. Đơn hàng có thể hủy chỉ khi đơn hàng đó chưa được xác nhận |
| 12 | Quản lý thông báo | Khách hàng có thể nhận được các thông báo về trạng thái đơn hàng, khuyến mãi,… |

Bảng 3. Khách hàng

#### Yêu cầu hệ thống

##### Usecase tổng quát



Hình 3. Usecase tổng quát

##### Usecase chi tiết

###### Usecase Đăng ký tài khoản

|  |  |
| --- | --- |
| **Name** | Đăng ký tài khoản |
| **Brief description** | User đăng ký một tài khoản cá nhân để đăng nhập vào trang web |
| **Actor(s)** | User |
| **Pre-conditions** | Người dùng đã truy cập vào hệ thống dưới tư cách User |
| **Post-conditions** | Nếu thành công: hệ thống trả về tài khoản đăng nhập vào trang web cho User đó.  Nếu thất bại: hệ thống thông báo lỗi về thông tin tài khoản và tài khoản không được tạo. |
| **Flow of events** |  |
| Basic flow  (Thành công) | Use case bắt đầu khi người dùng chọn chức năng đăng ký tài khoản   1. User chọn button đăng ký tài khoản 2. Nhập các thông tin cần thiết để tạo tài khoản 3. Hệ thống kiểm tra thông tin nhập vào và so sánh với cơ sở dữ liệu 4. Thêm tài khoản vào cơ sở dữ liệu 5. Vào email xác thực đăng ký 6. Cơ sở dữ liệu cập nhật trạng thái đã thực 7. Hiển thị thông báo khi đăng ký tài khoản thành công |
| Alternative flow  (Thất bại) | Nếu người dùng nhập sai thông tin. Những công việc sau đây được thực hiện:   * + - 1. Hệ thống thông báo sang ràng buộc và yêu cầu nhập lại thông tin       2. Người dùng nhập lại thông tin       3. Quay lại bước 3 ở Basic flow sẽ được thực hiện |
| **Extension point** | Không có |

Bảng 3. Usecase đăng ký

###### Usecase Đăng nhập

|  |  |
| --- | --- |
| **Name** | Đăng nhập bằng email |
| **Brief description** | User đăng nhập vào tài khoản cá nhân để tạo đơn hàng và thanh toán |
| **Actor(s)** | User |
| **Pre-conditions** | Người dùng truy cập vào hệ thống |
| **Post-conditions** | Nếu thành công: hệ thống cho phép User truy cập vào hệ thống với những quyền của tài khoản đó.  Nếu thất bại: hệ thống thông báo lỗi về thông tin tài khoản và không đăng nhập được vào hệ thống. |
| **Flow of events** |  |
| Basic flow  (Thành công) | Use case bắt đầu khi người dùng chọn chức năng đăng nhập tài khoản   1. User chọn button đăng nhập tài khoản 2. Nhập username và password 3. Hệ thống kiểm tra thông tin đăng nhập vào và so sánh với cơ sở dữ liệu 4. Hiển thị thông báo khi đăng nhập tài khoản thành công |
| Alternative flow  (Thất bại) | Nếu người dùng nhập sai thông tin. Những công việc sau đây được thực hiện:   * + - 1. Hệ thống thông báo sang ràng buộc và yêu cầu nhập lại thông tin       2. Người dùng nhập lại thông tin       3. Quay lại bước 3 ở Basic flow sẽ được thực hiện |
| **Extension point** | Không có |

Bảng 3. Usecase đăng nhập email

|  |  |
| --- | --- |
| **Name** | Đăng nhập bằng Google |
| **Brief description** | User đăng nhập vào tài khoản cá nhân để tạo đơn hàng và thanh toán |
| **Actor(s)** | User |
| **Pre-conditions** | Người dùng truy cập vào hệ thống |
| **Post-conditions** | Nếu thành công: hệ thống cho phép User truy cập vào hệ thống với những quyền của tài khoản đó.  Nếu thất bại: hệ thống thông báo lỗi về thông tin tài khoản và không đăng nhập được vào hệ thống. |
| **Flow of events** |  |
| Basic flow  (Thành công) | Use case bắt đầu khi người dùng chọn chức năng đăng nhập Google   1. User chọn button Google 2. Nhập username và password 3. Hệ thống kiểm tra thông tin đăng nhập vào và so sánh với cơ sở dữ liệu 4. Hiển thị thông báo khi đăng nhập tài khoản thành công |
| Alternative flow  (Thất bại) | Nếu người dùng nhập sai thông tin. Những công việc sau đây được thực hiện:   * + - 1. Hệ thống thông báo sang ràng buộc và yêu cầu nhập lại thông tin       2. Người dùng nhập lại thông tin       3. Quay lại bước 3 ở Basic flow sẽ được thực hiện |
| **Extension point** | Không có |

Bảng 3. Usecase đăng nhập bằng Gmail

|  |  |
| --- | --- |
| **Name** | Đăng nhập Facebook |
| **Brief description** | User đăng nhập vào tài khoản cá nhân để tạo đơn hàng và thanh toán |
| **Actor(s)** | User |
| **Pre-conditions** | Người dùng truy cập vào hệ thống |
| **Post-conditions** | Nếu thành công: hệ thống cho phép User truy cập vào hệ thống với những quyền của tài khoản đó.  Nếu thất bại: hệ thống thông báo lỗi về thông tin tài khoản và không đăng nhập được vào hệ thống. |
| **Flow of events** |  |
| Basic flow  (Thành công) | Use case bắt đầu khi người dùng chọn chức năng đăng nhập Facebook   1. User chọn button đăng nhập tài khoản 2. Nhập username và password 3. Hệ thống kiểm tra thông tin đăng nhập vào và so sánh với cơ sở dữ liệu 4. Hiển thị thông báo khi đăng nhập tài khoản thành công |
| Alternative flow  (Thất bại) | Nếu người dùng nhập sai thông tin. Những công việc sau đây được thực hiện:   * + - 1. Hệ thống thông báo sang ràng buộc và yêu cầu nhập lại thông tin       2. Người dùng nhập lại thông tin       3. Quay lại bước 3 ở Basic flow sẽ được thực hiện |
| **Extension point** | Không có |

Bảng 3. Usecase đăng nhập bằng Facebook

###### Usecase Thêm sản phẩm vào giỏ hàng

|  |  |
| --- | --- |
| **Name** | Thêm sản phẩm vào giỏ hàng |
| **Brief description** | User đăng nhập vào tài khoản cá nhân để thêm sản phẩm vào giỏ hàng |
| **Actor(s)** | User |
| **Pre-conditions** | Người dùng truy cập vào hệ thống |
| **Post-conditions** | Nếu thành công: hệ thống cho phép User truy cập vào hệ thống với những quyền của tài khoản đó.  Nếu thất bại: hệ thống thông báo lỗi về thông tin tài khoản và không đăng nhập được vào hệ thống. |
| **Flow of events** |  |
| Basic flow  (Thành công) | Use case bắt đầu khi người dùng chọn chức năng thêm sản phẩm vào giỏ hàng   1. User chọn sản phẩm muốn mua 2. Chọn số lượng sản phẩm 3. Click button thêm vào giỏ hàng 4. Hiện thị thông báo thêm vào giỏ hàng thành công |
| Alternative flow  (Thất bại) | Nếu người dùng chọn sản phẩm đã hết hàng. Những công việc sau đây được thực hiện:   * + - 1. Hệ thống thông báo sản phẩm đã hết hàng       2. Quay lại bước 1 ở Basic flow sẽ được thực hiện |
| **Extension point** | Không có |

Bảng 3. Usecase Thêm sản phẩm vào giỏ hàng

###### Usecase Quên mật khẩu

|  |  |
| --- | --- |
| **Name** | Quên mật khẩu |
| **Brief description** | User quên mật khẩu khi đăng nhập |
| **Actor(s)** | User |
| **Pre-conditions** | Người dùng cập nhật lạ mật khẩu mới |
| **Post-conditions** | Nếu thành công: hệ thống cho phép User đổi mật khẩu mới  Nếu thất bại: hệ thống thông báo lỗi về đổi mật khẩu thất bại |
| **Flow of events** |  |
| Basic flow  (Thành công) | Use case bắt đầu khi người dùng chọn chức năng quên mật khẩu   1. User chọn “Quên mật khẩu” 2. Hệ thống thông báo “Vui lòng check email để đổi mật khẩu mới” 3. Đăng nhập email đăng kí trước đó 4. Click vào link gửi đến 5. Điền mật khẩu mới và mật khẩu xác nhận 6. Click button xác nhận 7. Hệ thống thông báo cập nhật mật khẩu mới thành công |
| Alternative flow  (Thất bại) | Nếu người dùng nhập sai thông tin. Những công việc sau đây được thực hiện:   * + - 1. Hệ thống thông báo sang ràng buộc và yêu cầu nhập lại mật khẩu       2. Người dùng nhập lại mật khẩu       3. Quay lại bước 5 ở Basic flow sẽ được thực hiện |
| **Extension point** | Không có |

Bảng 3. Usecase Quên mật khẩu

###### Usecase Đăng xuất

|  |  |
| --- | --- |
| **Name** | Đăng xuất |
| **Brief description** | User đăng xuất tài khoản khỏi hệ thống |
| **Actor(s)** | User |
| **Pre-conditions** | Người dùng đăng xuất khỏi hệ thống |
| **Post-conditions** | Nếu thành công: hệ thống cho phép User thoát khỏi chức năng user trên hệ thống nhưng vẫn xem được thông tin trên hiện cơ bản  Nếu thất bại: hệ thống thông báo lỗi về đăng xuất thất bại |
| **Flow of events** |  |
| Basic flow  (Thành công) | Use case bắt đầu khi người dùng chọn chức năng đăng xuất tài khoản   1. User chọn biểu tượng ảnh đại diện 2. Chọn đăng xuất 3. Hệ thống trở lại trang đăng nhập |
| Alternative flow  (Thất bại) |  |
| **Extension point** | Không có |

Bảng 3. Usecase Đăng xuất

###### Usecase Cập nhật sản phẩm trong giỏ hàng

|  |  |
| --- | --- |
| **Name** | Cập nhật sản phẩm trong giỏ hàng |
| **Brief description** | User đăng nhập vào tài khoản cá nhân để cập nhật sản phẩm trong giỏ hàng |
| **Actor(s)** | User |
| **Pre-conditions** | Người dùng truy cập vào hệ thống |
| **Post-conditions** | Nếu thành công: hệ thống cho phép User truy cập vào hệ thống với những quyền của tài khoản đó.  Nếu thất bại: hệ thống thông báo lỗi về thông tin tài khoản và không đăng nhập được vào hệ thống. |
| **Flow of events** |  |
| Basic flow  (Thành công) | Use case bắt đầu khi người dùng chọn chức năng cập nhật sản phẩm trong giỏ hàng   1. User chọn sản phẩm muốn cập nhật 2. Chọn số lượng sản phẩm 3. Chọn topping mong muốn |
| Alternative flow  (Thất bại) | Nếu người dùng chọn sản phẩm đã hết hàng. Những công việc sau đây được thực hiện:   * + - 1. Hệ thống thông báo sản phẩm không đủ số lượng       2. Quay lại bước 1 ở Basic flow sẽ được thực hiện |
| **Extension point** | Không có |

Bảng 3. Usecase Cập nhật sản phẩm trong giỏ hàng

###### Usecase Xóa sản phẩm trong giỏ hàng

|  |  |
| --- | --- |
| **Name** | Xóa sản phẩm trong giỏ hàng |
| **Brief description** | User đăng nhập vào tài khoản cá nhân để xóa sản phẩm trong giỏ hàng |
| **Actor(s)** | User |
| **Pre-conditions** | Người dùng truy cập vào hệ thống |
| **Post-conditions** | Nếu thành công: hệ thống cho phép User truy cập vào hệ thống với những quyền của tài khoản đó.  Nếu thất bại: hệ thống thông báo lỗi về thông tin tài khoản và không đăng nhập được vào hệ thống. |
| **Flow of events** |  |
| Basic flow  (Thành công) | Use case bắt đầu khi người dùng chọn chức năng cập nhật sản phẩm trong giỏ hàng   1. User chọn sản phẩm muốn xóa 2. Click button xóa 3. Hệ thống thông báo sản phẩm đã được xóa |
| Alternative flow  (Thất bại) | Nếu người dùng chọn xóa thất bại. Những công việc sau đây được thực hiện:   * + - 1. Hệ thống thông báo xóa sản phẩm thất bại |
| **Extension point** | Không có |

Bảng 3. Usecase Xóa sản phẩm trong giỏ hàng

###### Usecase Order sản phẩm

|  |  |
| --- | --- |
| **Name** | Order sản phẩm |
| **Brief description** | User đăng nhập vào tài khoản cá nhân để tiến hành thanh toán |
| **Actor(s)** | User |
| **Pre-conditions** | Người dùng truy cập vào hệ thống |
| **Post-conditions** | Nếu thành công: hệ thống cho phép User truy cập vào hệ thống với những quyền của tài khoản đó.  Nếu thất bại: hệ thống thông báo lỗi về thông tin tài khoản và không đăng nhập được vào hệ thống. |
| **Flow of events** |  |
| Basic flow  (Thành công) | Use case bắt đầu khi người dùng chọn chức năng thanh toán   1. Click button “Xem thông tin giỏ hàng” 2. Chọn phương thức đặt hàng 3. Chọn chi nhánh 4. Chọn địa chỉ nhận hàng 5. Chọn sản phẩm cần mua 6. Chọn phương thức thanh toán 7. Click button “Mua hàng” 8. Nếu bước 6 thanh toán online thì điền thông tin thẻ thanh toán. |
| Alternative flow  (Thất bại) | Nếu người dùng thanh toán thất bại. Những công việc sau đây được thực hiện:   * + - 1. Hệ thống thông báo vui lòng chọn sản phẩm cần mua       2. Trường hợp thanh toán online vui lòng giá trị đơn hàng lớn hơn 5000đ và nhỏ hơn 1 tỷ đồng |
| **Extension point** | Không có |

Bảng 3. Usecase Order sản phẩm

###### Usecase Group order

|  |  |
| --- | --- |
| **Name** | Group order |
| **Brief description** | User đăng nhập vào tài khoản cá nhân để tiến hành thanh toán |
| **Actor(s)** | User |
| **Pre-conditions** | Người dùng truy cập vào hệ thống |
| **Post-conditions** | Nếu thành công: hệ thống cho phép User truy cập vào hệ thống với những quyền của tài khoản đó.  Nếu thất bại: hệ thống thông báo lỗi về thông tin tài khoản và không đăng nhập được vào hệ thống. |
| **Flow of events** |  |
| Basic flow  (Thành công) | Use case bắt đầu khi người dùng chọn chức năng group order   1. Click button “invite” 2. Click vào link để sao chép 3. Gửi link cho người muốn order chung 4. Nếu bước 2 không gửi link thì có thể quét mã Qr 5. Người được mời nhấp vào link hoặc quét mã Qr được gửi. 6. Tiến hành đăng nhập 7. Chọn sản phẩm muốn mua 8. Hệ thống thông báo “Bạn đã thêm sản phẩm vào giỏ hàng” 9. Click button “Tôi đã xong” |
| Alternative flow  (Thất bại) | Nếu người dùng thanh order chung thất bại. Những công việc sau đây được thực hiện:   * + - 1. Hệ thống thông báo link không còn tồn tại |
| **Extension point** | Không có |

Bảng 3. Usecase Group order

## Yêu cầu phi chức năng

### Khả năng sử dụng

Khách hàng và khách vãng lai có thể sử dụng ứng dụng dễ dàng

Khách vãng lai chưa đăng nhập không thể sử dụng một số chức năng

### Tính xác thực

* Bảo mật: Mỗi khách hàng có một tài khoản để sử dụng các chức năng nâng cao. Nếu chưa đăng nhập sẽ không thể sử dụng được
* Vấn đề hệ thống
* Hệ thống sẽ được sử dụng và duy trì 2 tuần/lần
* Nếu hệ thống có lỗi nghiêm trọng sẽ được sửa chữa ngay lập tức
* Chỉ được xuất hiện các lỗi nhỏ

### Hiệu suất

* Thời gian thực hiện đặt hàng chung dưới 30s
* Các giao dịch được thực hiện dưới 10s
* Khi thêm một chức năng mới đảm bảo các chức năng khách hoạt động bình thường

### Công cụ thực hiện

Các công cụ sử dụng là IDE VSCode, MongoDB Compass hay các IDE hỗ trợ code JavaScript

### Giao diện sử dụng

#### Giao diện người dùng

* Giao diện phải thân thiện và dễ sử dụng
* Hình ảnh không được vỡ
* Các button dễ bấm
* Font chữ dễ đọc
* Giao diện không bị vỡ khi sử dụng trên các màn hình khác nhau

#### Yêu cầu phần cứng

Tất cả các máy tính sử dụng được trình duyệt web đều có thể truy cập và sử dụng trang web bình thường

#### Giao diện ứng dụng

Ứng dụng sử dụng các thư viện hỗ trợ của ReactJS

## Yêu cầu APIs

### Chức năng login với POST method

Graphical user interface

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Hình 3. Chức năng login với POST method

### Chức năng hiển thị danh sách địa chỉ của người dùng với GET method

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Hình 3. Chức năng hiển thị danh sách địa chỉ của người dùng với GET method

### Chức năng đặt địa chỉ mặc định với PUT method

Graphical user interface, application

Description automatically generatedGraphical user interface, text, application

Description automatically generated

Hình 3. Chức năng đặt địa chỉ mặc định với PUT method

### Chức năng thêm mới địa chỉ với POST method

Graphical user interface, text, application, email

Description automatically generated

Text

Description automatically generated

Hình 3. Chức năng thêm mới địa chỉ với POST method

# THIẾT KẾ HỆ THỐNG

## Kiến trúc hệ thống

### Kiến trúc Front-end

* Kiến trúc fron-end được xây dựng bởi ReactJS- là thư viện được ưa chuộng để xây dựng một SPA(Single-Page-Application) giúp hỗ trợ người dùng xây dựng một trang web dễ dàng
* Bên cạnh đó, React cũng có một số lượng thư viện hỗ trợ rất lớn để giúp người dùng xây dựng giao diện theo ý thích cũng như các thao tác gọi APIs,… một cách dễ dàng và nhanh chóng
* Kiến trúc front-end được xây dựng trong dự án này theo dạng Component hóa chương tình, tức là sẽ chia chương trình thành các component nhỏ. Điều này giúp duy trì code mà quản lý code một cách dễ dàng

Graphical user interface

Description automatically generated with medium confidence

Hình 4. Tổ chức folder Front-end

* Thư mục **base** chứa các component của chương tình
* Thư mục **core** chứa các hàm xử lý với APIs
* Thư mục **feature** chứa các màn hình của chương tình
* Thư mục **store** chứa các state của chương trình cho Redux quản lý
* Thư mục **translation** chứa các cấu hình để làm multi-language
* Thư mục **ultilities** chứa các component, functions,.. có thể tái sử dụng ở các dự án khác

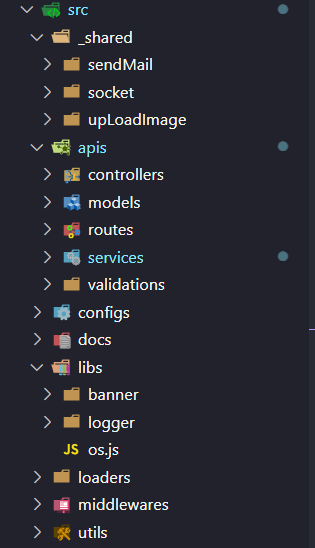
Diagram

Description automatically generated

Hình 4. Mô hình Reac-Redux

* Component sẽ đóng vai trò hiển thị giao diện, hiển thị giữ liệu và là nơi người dùng có thể tương tác. Nhận gửi đi request đến back-end và nhận response khi back-end thực hiện xong
* Sau khi nhận được response từ back-end, component sẽ dispatch một action tương ứng với mỗi chức năng đến store để tương tác với dữ liệu bên trong
* Store được xây dựng bởi Redux – quản lý các state của chương trình. Ứng với mỗi action nhận vào, store sẽ xử lý các state tương ứng với action đó và cập nhất giá trị state và thông báo sự thay đổi state đến component
* Component nhận thấy có sự thay đối state sẽ thực hiện các tác vụ tương ứng và gửi request đến back-end

### Kiến trúc Back-end

* Các framework nodejs phổ biết như Express cho phép chúng ta dễ dàng tạo ra Resful API xử lí các request từ phía client một cách nhanh chóng và linh hoạt.
* Thư mục \_share chứa các tính năng có thể dùng chung ở nhiều nơi

Hình 4. 3 Tổ chức folder backend

* Thư mục Api nơi nhận các request từ clien và response về cho người dùng

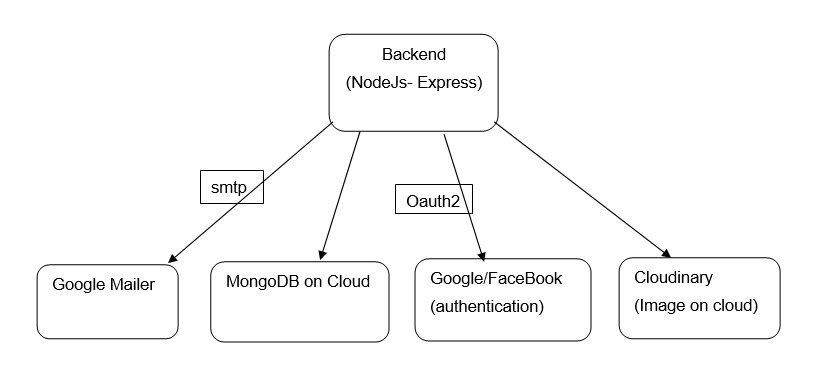
+ Thự mục Controller có vai trò điều hướng dữ liệu

+ Thư mục Model định nghĩa các model, schema cho database

+ Thư mục Route phương thức khai báo để đáp lại requrest từ client.

+ Thư mục service nơi xử lý các logic và tương tác với model.

+ Thư mục validation xác thực dữ liệu

* Thư mục configs cấu hình kết nói đến database
* Thư mục doc viết tài liệu đặt tả nếu cần
* Thư mục libs chứa các banner, logger thuận tiện cho việc theo dõi api được gửi đến.
* Thư mục loaders: Chia tách các module nhỏ thay vì tập trung ở một nơi
* Thư mục middleware tiền xử lý trước khi được đưa đến controller
* Thư mục Utils define các giá trị hay được sử dụng

Hình 4. 4 Mô hình kiến trúc backend

Quy trình xây dựng một enpoind:

* Bước 1: Xác định mục đích của endpoint.
* Bước 2: Lựa chọn phương thức phù hợp.
* Bươc 3: Định tuyến đường đi.
* Bước 4: Điều hướng dữ liệu.
* Bước 5: Xử lý logic dữ liệu.
* Bước 6: Tương tác database
* Bước 7: Gửi Resquest ra bên ngoài.

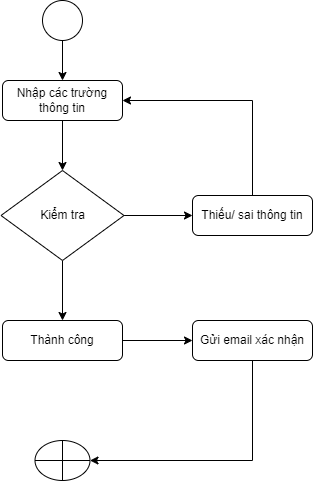
## ER Diagram



Hình 4. Mô hình ERD

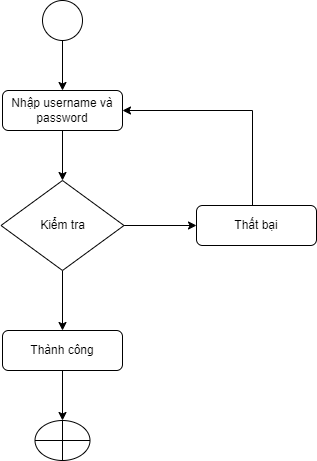
## Activity Diagram

### Đăng ký Activity Diagram



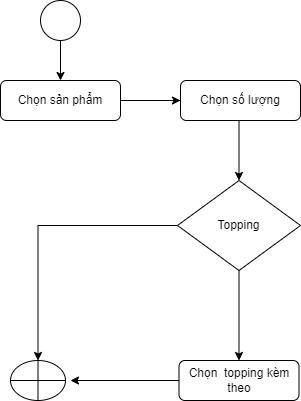
Hình 4. Thanh toán activity diagram

### Đăng nhập Acitvity Diagram



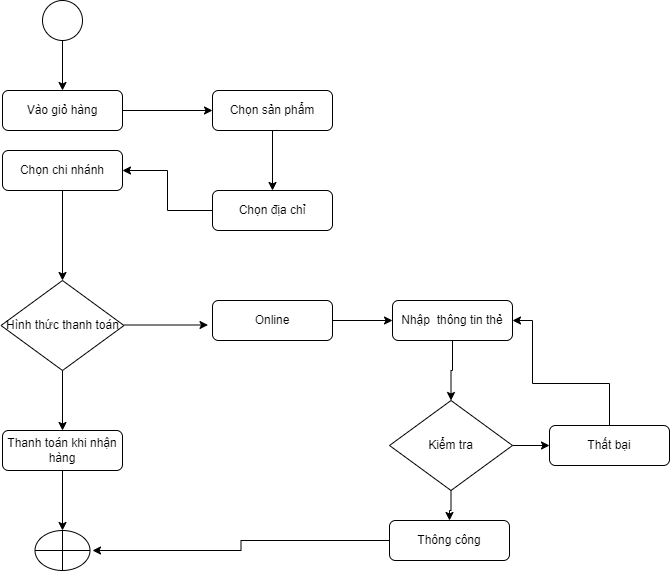
Hình 4. Đăng nhập activity diagram

### Thêm sản phẩm vào giỏ hàng Activity diagram



Hình 4. Thêm sản phẩm vào giỏ hàng activity diagram

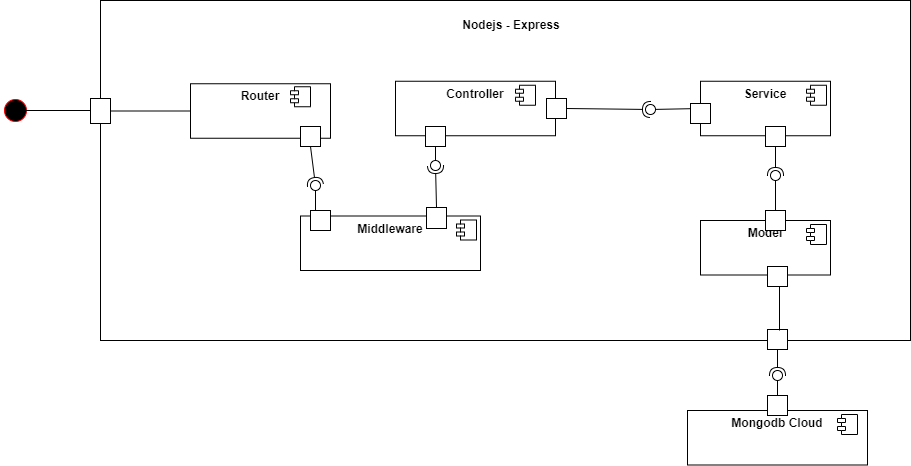
### Thanh toán Activity Diagram



Hình 4. Thanh toán activity diagram

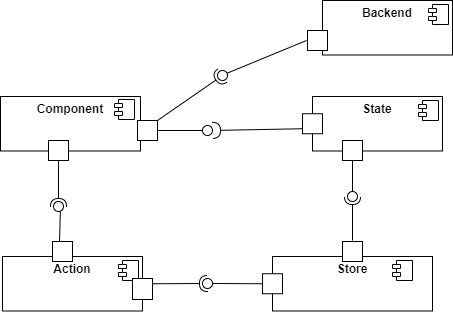
## Component Diagram

### Backend



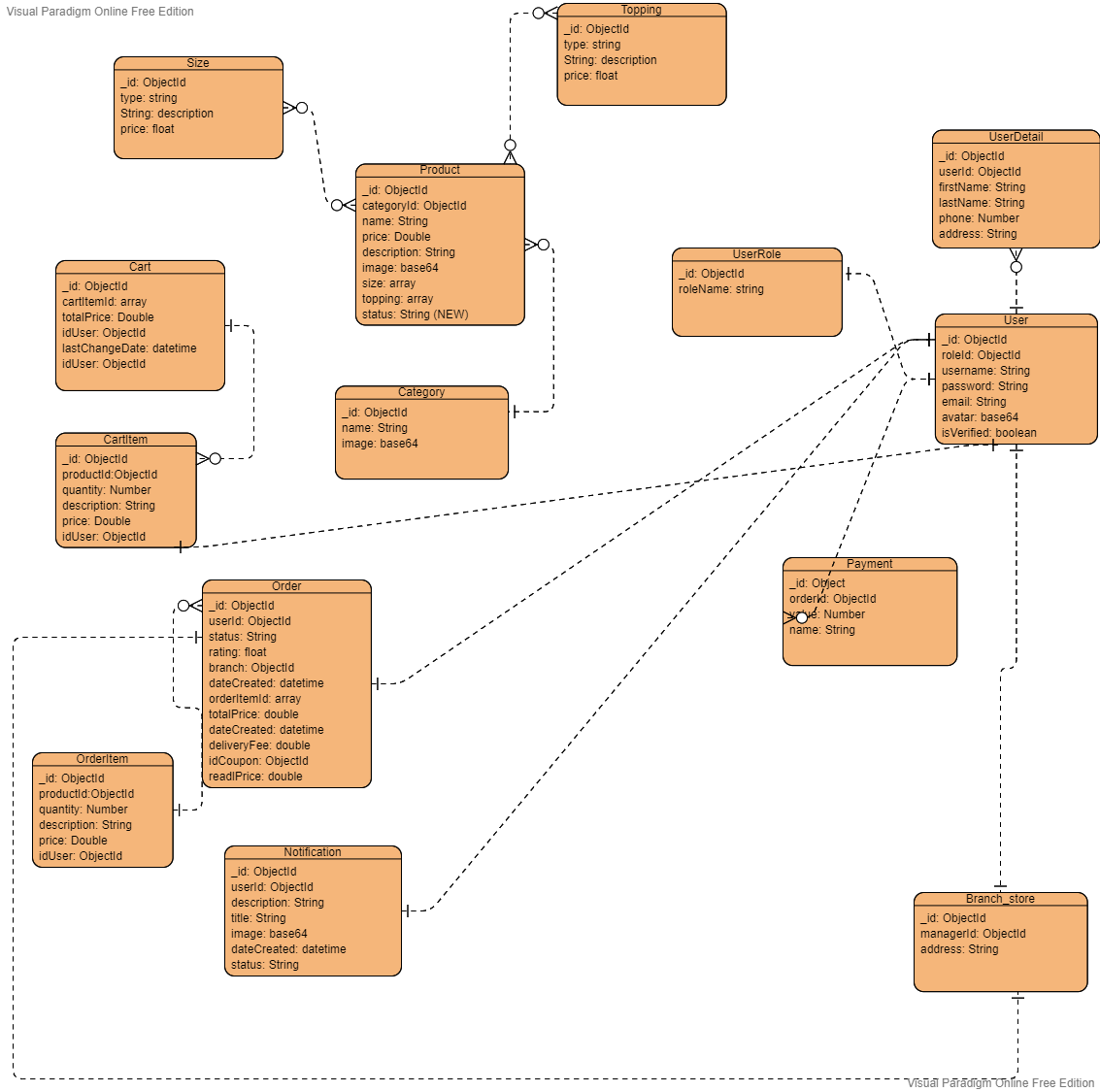
Hình 4. Backend component diagram

### Frontend



Hình 4. Frontend component diagram

## Database Diagram



Hình 4. Mô hình Database Diagram

### Mô tả các bảng

#### Bảng User

|  |  |  |  |
| --- | --- | --- | --- |
| Thuộc tính | Mô tả | Kiểu dữ liệu | Khóa |
| id | Id người dùng | Object | Khóa chính |
| roleId |  | String |  |
| userDetail | Thông tin chi tiết người dùng | Array |  |
| name | Tên người dùng | String |  |
| email | Email người dùng | String |  |
| password | Mật khẩu người dùng | String |  |
| avatar | Ảnh đại diện người dùng | String |  |
| sex | Giới tính người dùng | String |  |
| isEmailVerified | Xác thực tài thực đã đăng ký thành công | Boolean |  |

Bảng 4. 1 Bảng user

#### Bảng UserDetail

|  |  |  |  |
| --- | --- | --- | --- |
| Thuộc tính | Mô tả | Kiểu dữ liệu | Khóa |
| id | Id thông tin chi tiết của người dùng | Object | Khóa chính |
| userId | Id người dùng | String | Khóa ngoại |
| name | Tên người dùng | String |  |
| dateOfBirth | Ngày/tháng/năm sinh người dùng | Date |  |
| phone | Số điện thoại người dùng | Number |  |
| type | Kiểu địa chỉ | String |  |
| fullAddress | Địa chỉ người dùng | String |  |
| city | Thành phố | String |  |
| province | Quận/Huyện | String |  |
| street | Số nhà/tên đường | String |  |
| latitude | Vĩ độ | String |  |
| longitude | Kinh độ | String |  |
| isDefault | Xét kiểu mặc định | Boolean |  |

Bảng 4. 2 Bảng userDetail

#### Bảng UserRole

|  |  |  |  |
| --- | --- | --- | --- |
| Thuộc tính | Mô tả | Kiểu dữ liệu | Khóa |
| id | Id của role người dùng | Object | Khóa chính |
| roleName | Tên quyền | String |  |

Bảng 4. 3 Bảng userRole

#### Bảng Product

|  |  |  |  |
| --- | --- | --- | --- |
| Thuộc tính | Mô tả | Kiểu dữ liệu | Khóa |
| id | Id sản phẩm | Object | Khóa chính |
| categoryId | Id danh mục sản phẩm | String | Khóa ngoại |
| initPrice | Giá sản phẩm sau khi giảm | Number |  |
| idCoupon | Id mã giảm giá | String | Khóa ngoại |
| price | Giá sản phẩm | Number |  |
| name | Tên sản phẩm | String |  |
| description | Mô tả san phẩm | String |  |
| shortDescription | Mô tả ngắn gọn sản phẩm | int |  |
| image | Hình ảnh minh họa cho sản phẩm | String |  |
| size | Kích thước sản phẩm | Array | Khóa ngoại |
| topping | Các topping kèm theo | Array | Khóa ngoại |
| status | Trạng thái sản phẩm | String |  |

Bảng 4. 4 Bảng product

#### Bảng Category

|  |  |  |  |
| --- | --- | --- | --- |
| Thuộc tính | Mô tả | Kiểu dữ liệu | Khóa |
| id | Id danh mục | Object | Khóa chính |
| name | Tên danh mục | String |  |
| image | Hình ảnh minh họa | String |  |

Bảng 4. 5 Bảng category

#### Bảng Cart

|  |  |  |  |
| --- | --- | --- | --- |
| Thuộc tính | Mô tả | Kiểu dữ liệu | Khóa |
| id | Id của giỏ hàng | long | Khóa chính |
| CartItemId | Id của item có trong giỏ hàng | int | Khóa ngoại |
| userId | Id của người dùng | String | Khóa ngoại |
| mount | Số lượng sản phẩm có trong giỏ hàng | String |  |
| totalPrice | Tổng giá thành của giỏ hàng | int |  |

Bảng 4. 6 Bảng cart

#### Bảng CartItem

|  |  |  |  |
| --- | --- | --- | --- |
| Thuộc tính | Mô tả | Kiểu dữ liệu | Khóa |
| id | Id của cartItem | Object | Khóa chính |
| userid | Id người mua hàng | String | Khóa ngoại |
| quantity | Só lượng sản phẩm có trong cartItem | Number |  |
| description | Mô tả sản phẩm có trong cartItem | String |  |
| price | Giá thành một sản phẩm | Number |  |
| total | Giá tổng của một cartItem | Number |  |
| sizeChooose | Kích cỡ sản phẩm được chọn | String | Khóa ngoại |
| toppingChoose | Topping kèm theo | Array | Khóa ngoại |
| note | Ghi chú cho cartItem | String |  |

Bảng 4. 7 Bảng cartItem

#### Bảng Order

|  |  |  |  |
| --- | --- | --- | --- |
| Thuộc tính | Mô tả | Kiểu dữ liệu | Khóa |
| id | Id của cartItem | Object | Khóa chính |
| userid | Id người mua hàng | String | Khóa ngoại |
| branchId | Id chi nhánh | String | Khóa ngoại |
| orderItemId | Id orderItem chứa thông tin một sản phẩm | Array | Khóa ngoại |
| couponId | Id các mã giảm giá | Array | Khóa ngoại |
| status | Trạng thái đơn hàng | String |  |
| note | Ghi chú cho đơn hàng | String |  |
| rating | Đánh giá đơn | Number |  |
| totalPrice | Tổng giá thành đơn hàng | Number |  |
| deliveryFee | Phí giao hàng | Number |  |
| realPrice | Giá thành sau khi giảm giá | Number |  |
| mount | Số lượng sản phẩm có trong đơn | Number |  |
| ispay | Kiểm tra đơn đã thanh toán hay chưa | Boolean |  |
| typePayment | Kiểu thanh toán | String |  |

Bảng 4. 8 Bảng order

#### Bảng OrderItem

|  |  |  |  |
| --- | --- | --- | --- |
| Thuộc tính | Mô tả | Kiểu dữ liệu | Khóa |
| id | Id của orderItem | Object | Khóa chính |
| productId | Id sản phẩm | String | Khóa ngoại |
| userId | Id người mua hàng | String | Khóa ngoại |
| quantity | Số lượng sản phẩm | Number |  |
| description | Mô tả cho đơn hàng | String |  |
| price | Giá thành cho một sản phẩm | Number |  |
| total | Giá thành tổng thể theo số lượng | Number |  |
| sizeChooose | Kích cỡ cho sản phẩm | String |  |
| topppingChoose | Các topping kèm theo | Array |  |
| note | Ghi chú cho shipper | String |  |

Bảng 4. 9 Bảng orderItem

#### Bảng size

|  |  |  |  |
| --- | --- | --- | --- |
| Thuộc tính | Mô tả | Kiểu dữ liệu | Khóa |
| id | Id của kích cỡ sản phẩm | Object | Khóa chính |
| type | Kích cỡ sản phẩm | String |  |
| description | Mô tả cho kích cỡ | String |  |
| price | Giá thành | Number |  |

Bảng 4. 10 Bảng size

#### Bảng Topping

|  |  |  |  |
| --- | --- | --- | --- |
| Thuộc tính | Mô tả | Kiểu dữ liệu | Khóa |
| id | Id của kích cỡ sản phẩm | Object | Khóa chính |
| type | Kích cỡ sản phẩm | String |  |
| description | Mô tả cho kích cỡ | String |  |
| price | Giá thành | Number |  |
| image | Hình ảnh cho topping | String |  |

Bảng 4. 11 Bảng topping

#### Bảng Token

|  |  |  |  |
| --- | --- | --- | --- |
| Thuộc tính | Mô tả | Kiểu dữ liệu | Khóa |
| id | Id của token | Object | Khóa chính |
| token | Token xác thực người dùng | String |  |
| user | Id của người dùng | String | Khóa ngoại |
| type | Token sau khi refresh | String |  |
| expires | Ngày hết hạn | Date |  |

Bảng 4. 12 Bảng token

## Thiết kế giao diện

### Giao diện đăng nhập

A screenshot of a cell phone

Description automatically generated with low confidence

Hình 4. 13 Giao diện đăng nhập

### Giao diện đăng ký

A screenshot of a cell phone

Description automatically generated with low confidence

Hình 4. 14 Giao diện đăng ký

### Giao diện trang chủ

Graphical user interface, website

Description automatically generated

Hình 4. 15 Giao diện trang chủ

### Giao diện quản lý thông tin

Graphical user interface, application

Description automatically generated

Hình 4. 16 Giao diện quản lý thông tin

### Giao diện danh sách sản phẩm

A picture containing food, dish, screenshot

Description automatically generated

Hình 4. 17 Giao diện danh sách sản phẩm

### Graphical user interface, application Description automatically generatedGiao diện giỏ hàng

Hình 4. 18 Giao diện giỏ hàng

### Giao diện checkout

Graphical user interface, text, application, email, website

Description automatically generated

Hình 4. 19 Giao diện checkout

### Giao diện thanh toán VNPAY

Graphical user interface

Description automatically generated with medium confidence

Hình 4. 20 Giao diện thanh toán VNPAY

### Giao diện quản lý địa chỉ

Graphical user interface, text, application

Description automatically generated

Hình 4. 21 Giao diện quản lý địa chỉ

# THỰC THI HỆ THỐNG

## Công cụ và môi trường phát triển

|  |  |
| --- | --- |
| Tool | Name |
| Environment | NodeJS |
| IDE | VSCode |
| Dbms | MongoDB |
| Source control | Gitlab, Git bash |
| Modelling tools | Draw.io, Enterprise Architect |

* Back-end
* Phát triển với IDE VSCode
* Chạy với NodeJS Sever
* Sử dụng Postman để kiểm tra APIs
* Front-end
* Phát triển với IDE VSCode

## Công nghệ sử dụng

Trong dự án, các công nghệ được sử dụng được liệt kê dưới đây

|  |  |
| --- | --- |
| Transmission Protocol | REST API |
| JSON Web Token |
| User Interface | React v17.0.2 |
| System Technologies | NodeJS v14.17.3 |
| Express framework 4.17.1 |
| Databases | MongoDB |
| Services | NodeMailer (Gửi email) |
| Cloudinary (Lưu trữ hình ảnh) |
| AWS-SDK |
| Socket (Thực hiện realtime đặt hàng chung) |
| VNPAY Sandbox (Thanh toán VNPAY) |
| Firebase (Gửi thông báo) |
| Thirdly Party | GoogleAPIs (Đăng nhập facebook, google) |
| Mapbox.com ( Get Location) |

## Quản lý source code

* Quản lý source code thông quan GitLab
* Sử dụng Git bash để quản lý source code
* Source Frontend: https://gitlab.com/fast-food-management/front-end.git
* Source Backend: https://gitlab.com/fast-food-management/back-end.git

# THỰC NGHIỆM, ĐÁNH GIÁ, PHÂN TÍCH KẾT QUẢ

## Kế hoạch kiểm thử

Trong dự án, chúng tôi sử dụng Blackbox Tesing và Whitebox Testing để kiểm tra các chức năng của chương trình. Ưu điểm lớn nhất của BlackBox Testing là phù hợp với việc kiểm thử 1 số lượng lớn chức năng của hệ thống. Kiểm thử Whitebox sẽ giúp sửa những lỗi do code phát sinh trong quá trình phát triển.

Danh sách các chức năng được kiểm thử:

* Testcase:
* Kiểm thử khả năng hoạt động của chức năng với tài khoản đã đăng kí và đã Verify Email
* Kiểm thử khả năng hoạt động của chức năng với tài khoản Google
* Kiểm thử khả năng hoạt động của chức năng với tài khoản Facebook
* Kiểm tra khả năng hoạt động của tính năng chọn Option cho sản phẩm và thêm vào giỏ hàng của người dùng
* Kiểm tra khả năng hoạt động của giỏ hàng
* Kiểm tra khả năng loại bỏ sản phẩm ra khỏi giỏ hàng của người dùng
* Kiểm tra khả năng lưu lại đơn hàng sau khi đăng xuất
* Kiểm tra khả năngcập nhật lại đơn hàng cũ với thông tin sản phẩm không đổi
* Kiểm tra khả năng cập nhật lại đơn hàng cũ với cùng sản phẩm nhưng khác thông tin đi kèm
* Kiểm thử Whitebox:
* Chức năng login
* Chức năng UpdateUser
* Chức năng Active Email

## Các mẫu Testcase

* Kiểm thử khả năng hoạt động của chức năng với tài khoản đã đăng kí và đã Verify Email

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | GR4\_Login\_TestCase1 | **Mô tả TestCase** | | Kiểm thử khả năng hoạt động của chức năng với tài khoản đã đăng kí và đã Verify Email | | | | | |
| **Người tạo :** | | Hoàng Ngọc Doanh | **Người đánh giá:** | | Đinh Tấn Tú | | **Phiên bản** | | 1.0 | |
|  |  |  |  |  |  |  |  |  |  |  |
| **QA Tester’s Log** | |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Tên người Test** | | Doanh | **Ngày Test** | | November 15, 2021 | | **Kết quả TestCase(Pass/Fail/Not Executed)** | | Pass | |
|  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Điều kiện tiên quyết** | | |  | **S #** | **Dữ liệu thử** | | | | |
| 1 | FrontEnd đã có thể chạy được | | |  | 1 | "Email": 18110089@student.hcmute.edu.vn | | | | |
| 2 | BackEnd đã có thể chạy được | | |  | 2 | "Mật khẩu" : doanh31220 | | | | |
| 3 | Port 3000 và Port 5000 đang để trống. | | |  | 3 |  | | | | |
| 4 |  | | |  | 4 |  | | | | |
|  |  |  |  |  | 5 |  | | | | |
| **Kịch bản Test** |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Các bước** | **Các bước chi tiết** | | **Kết quả mong đợi** | | **Kết quả thực tế** | | | **Pass / Fail / Not executed / Suspended** | | |
|  |
| 1 | Khởi động chương trình | | Trang chủ của Website sẽ hiện lên với những thông tin về trang chủ. | | Như kết quả mong đợi | | | Pass | | |  |
| 2 | Nhấn chuột vào dòng chữ đăng nhập bên góc tay phải màn hình, cạnh biểu tượng avartar | | Giao diện đăng nhậptài khoản của Website sẽ được hiện ra | | Như kết quả mong đợi | | | Pass | | |  |
| 3 | Nhập thông tin tương ứng của Dữ liệu Test vào từng box phù hợp. | | Dữ liệu sẽ được nhập đúng (Mật khẩu và Xác nhận phải được ẩn) | | Như kết quả mong đợi | | | Pass | | |  |
| 4 | Nhấn nút đăng nhập | | Vòng waiting sẽ xuất hiện và trả về trang chủ | | Như kết quả mong đợi | | | Pass | | |  |
| 5 | Tại màn hình trang chủ có thay đổi | | Tại màn hình User thay vì dòng chữ đăng nhập nay đổi thành hình đại diện và thông tin người dùng được xổ xuống bên tay phải | | Như kết quả mong đợi | | | Pass | | |  |

Bảng 6. 1 Testcase đăng nhập email

* Kiểm thử khả năng hoạt động của chức năng với tài khoản Google

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | GR4\_Login\_TestCase2 | **Mô tả TestCase** | | Kiểm thử khả năng hoạt động của chức năng với tài khoản Google | | | | | |
| **Người tạo :** | | Hoàng Ngọc Doanh | **Người đánh giá:** | | Đinh Tấn Tú | | **Phiên bản** | | 1.0 | |
|  |  |  |  |  |  |  |  |  |  |  |
| **QA Tester’s Log** | |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Tên người Test** | | Doanh | **Ngày Test** | | November 15, 2021 | | **Kết quả TestCase(Pass/Fail/Not Executed)** | | Pass | |
|  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Điều kiện tiên quyết** | | |  | **S #** | **Dữ liệu thử** | | | | |
| 1 | FrontEnd đã có thể chạy được | | |  | 1 | "Email": gacon.chuot999@gmail.com | | | | |
| 2 | BackEnd đã có thể chạy được | | |  | 2 |  | | | | |
| 3 | Port 3000 và Port 5000 đang để trống. | | |  | 3 |  | | | | |
| 4 |  | | |  | 4 |  | | | | |
|  |  |  |  |  | 5 |  | | | | |
| **Kịch bản Test** |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Các bước** | **Các bước chi tiết** | | **Kết quả mong đợi** | | **Kết quả thực tế** | | | **Pass / Fail / Not executed / Suspended** | | |
|  |
| 1 | Khởi động chương trình | | Trang chủ của Website sẽ hiện lên với những thông tin về trang chủ. | | Như kết quả mong đợi | | | Pass | | |  |
| 2 | Nhấn chuột vào dòng chữ đăng nhập bên góc tay phải màn hình, cạnh biểu tượng avartar | | Giao diện đăng nhập tài khoản của Website sẽ được hiện ra | | Như kết quả mong đợi | | | Pass | | |  |
| 3 | Nhấn vào biểu tượng Google ở bên dưới button Đăng nhập | | Một trang nhỏ của Google sẽ mở ra cho phép bạn chọn lựa tài khoản sử dụng cho mục đích đăng nhập | | Như kết quả mong đợi | | | Pass | | |  |
| 4 | Chọn tài khoản theo địa chỉ Email trong dữ liệu thử | | Sau khi chọn tài khoản google sẽ trả về lại trang chủ | | Như kết quả mong đợi | | | Pass | | |  |
| 5 | Tại màn hình trang chủ có thay đổi | | Tại màn hình User thay vì dòng chữ đăng nhập nay đổi thành hình đại diện và thông tin người dùng được xổ xuống bên tay phải | | Như kết quả mong đợi | | | Pass | | |  |

Bảng 6. 2 Testcase đăng nhập bằng Google

* Kiểm thử khả năng hoạt động của chức năng với tài khoản Facebook

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | GR4\_Login\_TestCase3 | **Mô tả TestCase** | | Kiểm thử khả năng hoạt động của chức năng với tài khoản Facebook | | | | | |
| **Người tạo :** | | Hoàng Ngọc Doanh | **Người đánh giá:** | | Đinh Tấn Tú | | **Phiên bản** | | 1.0 | |
|  |  |  |  |  |  |  |  |  |  |  |
| **QA Tester’s Log** | |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Tên người Test** | | Doanh | **Ngày Test** | | November 15, 2021 | | **Kết quả TestCase(Pass/Fail/Not Executed)** | | Fail | |
|  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Điều kiện tiên quyết** | | |  | **S #** | **Dữ liệu thử** | | | | |
| 1 | FrontEnd đã có thể chạy được | | |  | 1 |  | | | | |
| 2 | BackEnd đã có thể chạy được | | |  | 2 |  | | | | |
| 3 | Port 3000 và Port 5000 đang để trống. | | |  | 3 |  | | | | |
| 4 |  | | |  | 4 |  | | | | |
|  |  |  |  |  | 5 |  | | | | |
| **Kịch bản Test** |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Các bước** | **Các bước chi tiết** | | **Kết quả mong đợi** | | **Kết quả thực tế** | | | **Pass / Fail / Not executed / Suspended** | | |
|  |
| 1 | Khởi động chương trình | | Trang chủ của Website sẽ hiện lên với những thông tin về trang chủ. | | Như kết quả mong đợi | | | Pass | | |  |
| 2 | Nhấn chuột vào dòng chữ đăng nhập bên góc tay phải màn hình, cạnh biểu tượng avartar | | Giao diện đăng nhập tài khoản của Website sẽ được hiện ra | | Như kết quả mong đợi | | | Pass | | |  |
| 3 | Nhấn vào biểu tượng Facebook ở bên dưới button Đăng nhập | | Một trang nhỏ của Facebook sẽ mở ra cho phép bạn chọn lựa tài khoản sử dụng cho mục đích đăng nhập | | Kết quả trả về thông tin như sau :" Ứng dụng này chưa được cấp phép sử dụng tài khoản facebook." Sau khi nhấn okey trả về màn hình "Please waiting a second ..." | | | Fail | | |  |
| 4 | Chọn liên kết địa chỉ Facebook | | Sau khi chọn tài khoản Facebook sẽ trả về lại trang chủ | |  | | |  | | |  |
| 5 | Tại màn hình trang chủ có thay đổi | | Tại màn hình User thay vì dòng chữ đăng nhập nay đổi thành hình đại diện và thông tin người dùng được xổ xuống bên tay phải | |  | | |  | | |  |

Bảng 6. 3 Testcase đăng nhập bằng Facebook

* Kiểm tra khả năng hoạt động của tính năng chọn Option cho sản phẩm và thêm vào giỏ hàng của người dùng

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | GR4\_User\_AddToCart\_TestCase30 | **Mô tả TestCase** | | Kiểm tra khả năng hoạt động của tính năng chọn Option cho sản phẩm và thêm vào giỏ hàng của người dùng | | | | | |
| **Người tạo :** | | Phan Hoàng Phúc | **Người đánh giá:** | | Lê Nguyễn Thanh Nhân | | **Phiên bản** | | 1.0 | |
|  |  |  |  |  |  |  |  |  |  |  |
| **QA Tester’s Log** | |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Tên người Test** | | Phúc | **Ngày Test** | | November 15, 2021 | | **Kết quả TestCase(Pass/Fail/Not Executed)** | | Pass | |
|  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Điều kiện tiên quyết** | | |  | **S #** | **Dữ liệu thử** | | | | |
| 1 | FrontEnd đã có thể chạy được | | |  | 1 | "Email": phancaocuong0000@gmail.com | | | | |
| 2 | BackEnd đã có thể chạy được | | |  | 2 | "Mật khẩu" : cuong123 | | | | |
| 3 | Port 3000 và Port 5000 đang để trống. | | |  | 3 | "Size" : S | | | | |
| 4 |  | | |  | 4 | "Số lượng" :2 | | | | |
|  |  |  |  |  | 5 |  | | | | |
| **Kịch bản Test** |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Các bước** | **Các bước chi tiết** | | **Kết quả mong đợi** | | **Kết quả thực tế** | | | **Pass / Fail / Not executed / Suspended** | | |
|  |
| 1 | Khởi động chương trình | | Trang chủ của Website sẽ hiện lên với những thông tin về trang chủ. | | Như kết quả mong đợi | | | Pass | | |  |
| 2 | Nhấn chuột vào dòng chữ đăng nhập bên góc tay phải màn hình, cạnh biểu tượng avartar | | Giao diện đăng nhậptài khoản của Website sẽ được hiện ra | | Như kết quả mong đợi | | | Pass | | |  |
| 3 | Nhập thông tin tương ứng của Dữ liệu Test vào từng box phù hợp. | | Dữ liệu sẽ được nhập đúng (Mật khẩu và Xác nhận phải được ẩn) | | Như kết quả mong đợi | | | Pass | | |  |
| 4 | Nhấn nút đăng nhập | | Vòng waiting sẽ xuất hiện và trả về trang chủ | | Như kết quả mong đợi | | | Pass | | |  |
| 5 | Tại màn hình trang chủ có thay đổi | | Tại màn hình User thay vì dòng chữ đăng nhập nay đổi thành hình đại diện và thông tin người dùng được xổ xuống bên tay phải | | Như kết quả mong đợi | | | Pass | | |  |
| 6 | Nhấn vào "ReadMore" trên thanh Slide giới thiệu sản phẩm | | Trang chủ sẽ nhảy sang trang Product và load lên màn hình toàn bộ sản phẩm có thể mua | | Như kết quả mong đợi | | | Pass | | |  |
| 7 | Nhấn vào bất kỳ một sản phẩm để xem thông tin sản phẩm | | Xuất hiện 1 modal chứa thông tin sản phẩm lên màn hình người dùng để họ xem thông tin chi tiết | | Như kết quả mong đợi | | | Pass | | |  |
| 8 | Thay đổi Option Size và Option số lượng | | Giá sản phẩm thay đổi đúng theo những Option mà người dùng lựa chọn | | Như kết quả mong đợi | | | Pass | | |  |
| 9 | Nhấn "Add to Cart" | | Thêm sản phẩm đã chọn vào giỏ hàng, modal lựa sản phẩm tắt, giỏ hàng thay đổi giá tiền bằng số tiền sản phẩm có trong giỏ | | Như kết quả mong đợi | | | Pass | | |  |

Bảng 6. 4 Testcase thêm sản phẩm vào giỏ hàng

* Kiểm tra khả năng hoạt động của giỏ hàng

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | GR4\_User\_CheckCart\_TestCase32 | **Mô tả TestCase** | | Kiểm tra khả năng hoạt động của giỏ hàng | | | | | |
| **Người tạo :** | | Phan Hoàng Phúc | **Người đánh giá:** | | Lê Nguyễn Thanh Nhân | | **Phiên bản** | | 1.0 | |
|  |  |  |  |  |  |  |  |  |  |  |
| **QA Tester’s Log** | |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Tên người Test** | | Phúc | **Ngày Test** | | November 15, 2021 | | **Kết quả TestCase(Pass/Fail/Not Executed)** | | Pass | |
|  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Điều kiện tiên quyết** | | |  | **S #** | **Dữ liệu thử** | | | | |
| 1 | FrontEnd đã có thể chạy được | | |  | 1 | "Email": phancaocuong0000@gmail.com | | | | |
| 2 | BackEnd đã có thể chạy được | | |  | 2 | "Mật khẩu" : cuong123 | | | | |
| 3 | Port 3000 và Port 5000 đang để trống. | | |  | 3 | "Size" : S | | | | |
| 4 |  | | |  | 4 | "Số lượng" :2 | | | | |
|  |  |  |  |  | 5 | "Số lượng": 10 | | | | |
| **Kịch bản Test** |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Các bước** | **Các bước chi tiết** | | **Kết quả mong đợi** | | **Kết quả thực tế** | | | **Pass / Fail / Not executed / Suspended** | | |
|  |
| 1 | Khởi động chương trình | | Trang chủ của Website sẽ hiện lên với những thông tin về trang chủ. | | Như kết quả mong đợi | | | Pass | | |  |
| 2 | Nhấn chuột vào dòng chữ đăng nhập bên góc tay phải màn hình, cạnh biểu tượng avartar | | Giao diện đăng nhậptài khoản của Website sẽ được hiện ra | | Như kết quả mong đợi | | | Pass | | |  |
| 3 | Nhập thông tin tương ứng của Dữ liệu Test vào từng box phù hợp. | | Dữ liệu sẽ được nhập đúng (Mật khẩu và Xác nhận phải được ẩn) | | Như kết quả mong đợi | | | Pass | | |  |
| 4 | Nhấn nút đăng nhập | | Vòng waiting sẽ xuất hiện và trả về trang chủ | | Như kết quả mong đợi | | | Pass | | |  |
| 5 | Tại màn hình trang chủ có thay đổi | | Tại màn hình User thay vì dòng chữ đăng nhập nay đổi thành hình đại diện và thông tin người dùng được xổ xuống bên tay phải | | Như kết quả mong đợi | | | Pass | | |  |
| 6 | Nhấn vào "ReadMore" trên thanh Slide giới thiệu sản phẩm | | Trang chủ sẽ nhảy sang trang Product và load lên màn hình toàn bộ sản phẩm có thể mua | | Như kết quả mong đợi | | | Pass | | |  |
| 7 | Nhấn vào bất kỳ một sản phẩm để xem thông tin sản phẩm | | Xuất hiện 1 modal chứa thông tin sản phẩm lên màn hình người dùng để họ xem thông tin chi tiết | | Như kết quả mong đợi | | | Pass | | |  |
| 8 | Thay đổi Option Size và Option số lượng | | Giá sản phẩm thay đổi đúng theo những Option mà người dùng lựa chọn | | Như kết quả mong đợi | | | Pass | | |  |
| 9 | Nhấn "Add to Cart" | | Thêm sản phẩm đã chọn vào giỏ hàng, modal lựa sản phẩm tắt, giỏ hàng thay đổi giá tiền bằng số tiền sản phẩm có trong giỏ | | Như kết quả mong đợi | | | Pass | | |  |
| 10 | Lựa chọn lại sản phẩm cũ với size cũ, thay đổi số lượng thêm 10 như dữ liệu Test, sau đó nhấn "Add to Cart" | | Giá trị giỏ hàng được thay đổi theo số tiền thêm | | Như kết quả mong đợi | | | Pass | | |  |
| 11 | Nhấn vào biểu tượng giỏ hàng bên góc trên tay phải | | Thông tin về sản phẩm trong giỏ hàng sẽ được hiển thị bên trên màn hình người dùng | | Như kết quả mong đợi | | | Pass | | |  |

Bảng 6. 5 Testcase hoạt động của giỏ hàng

* Kiểm tra khả năng loại bỏ sản phẩm ra khỏi giỏ hàng của người dùng

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | GR4\_User\_CheckCart\_TestCase33 | **Mô tả TestCase** | | Kiểm tra khả năng loại bỏ sản phẩm ra khỏi giỏ hàng của người dùng | | | | | |
| **Người tạo :** | | Phan Hoàng Phúc | **Người đánh giá:** | | Lê Nguyễn Thanh Nhân | | **Phiên bản** | | 1.0 | |
|  |  |  |  |  |  |  |  |  |  |  |
| **QA Tester’s Log** | |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Tên người Test** | | Phúc | **Ngày Test** | | November 15, 2021 | | **Kết quả TestCase(Pass/Fail/Not Executed)** | | Fail | |
|  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Điều kiện tiên quyết** | | |  | **S #** | **Dữ liệu thử** | | | | |
| 1 | FrontEnd đã có thể chạy được | | |  | 1 | "Email": phancaocuong0000@gmail.com | | | | |
| 2 | BackEnd đã có thể chạy được | | |  | 2 | "Mật khẩu" : cuong123 | | | | |
| 3 | Port 3000 và Port 5000 đang để trống. | | |  | 3 | "Size" : S | | | | |
| 4 |  | | |  | 4 | "Số lượng" :10 | | | | |
|  |  |  |  |  |  |  |  |  |  |  |
| **Kịch bản Test** |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Các bước** | **Các bước chi tiết** | | **Kết quả mong đợi** | | **Kết quả thực tế** | | | **Pass / Fail / Not executed / Suspended** | | |
|  |
| 1 | Khởi động chương trình | | Trang chủ của Website sẽ hiện lên với những thông tin về trang chủ. | | Như kết quả mong đợi | | | Pass | | |  |
| 2 | Nhấn chuột vào dòng chữ đăng nhập bên góc tay phải màn hình, cạnh biểu tượng avartar | | Giao diện đăng nhậptài khoản của Website sẽ được hiện ra | | Như kết quả mong đợi | | | Pass | | |  |
| 3 | Nhập thông tin tương ứng của Dữ liệu Test vào từng box phù hợp. | | Dữ liệu sẽ được nhập đúng (Mật khẩu và Xác nhận phải được ẩn) | | Như kết quả mong đợi | | | Pass | | |  |
| 4 | Nhấn nút đăng nhập | | Vòng waiting sẽ xuất hiện và trả về trang chủ | | Như kết quả mong đợi | | | Pass | | |  |
| 5 | Tại màn hình trang chủ có thay đổi | | Tại màn hình User thay vì dòng chữ đăng nhập nay đổi thành hình đại diện và thông tin người dùng được xổ xuống bên tay phải | | Như kết quả mong đợi | | | Pass | | |  |
| 6 | Nhấn vào "ReadMore" trên thanh Slide giới thiệu sản phẩm | | Trang chủ sẽ nhảy sang trang Product và load lên màn hình toàn bộ sản phẩm có thể mua | | Như kết quả mong đợi | | | Pass | | |  |
| 7 | Nhấn vào bất kỳ một sản phẩm để xem thông tin sản phẩm | | Xuất hiện 1 modal chứa thông tin sản phẩm lên màn hình người dùng để họ xem thông tin chi tiết | | Như kết quả mong đợi | | | Pass | | |  |
| 8 | Thay đổi Option Size và Option số lượng | | Giá sản phẩm thay đổi đúng theo những Option mà người dùng lựa chọn | | Như kết quả mong đợi | | | Pass | | |  |
| 9 | Nhấn "Add to Cart" | | Thêm sản phẩm đã chọn vào giỏ hàng, modal lựa sản phẩm tắt, giỏ hàng thay đổi giá tiền bằng số tiền sản phẩm có trong giỏ | | Như kết quả mong đợi | | | Pass | | |  |
| 10 | Nhấn vào biểu tượng giỏ hàng bên góc trên tay phải | | Giá trị giỏ hàng được thay đổi theo số tiền thêm | | Như kết quả mong đợi | | | Pass | | |  |
| 11 | Nhấn dấu x để bỏ toàn bộ sản phẩm ra khỏi giỏ hàng | | Giá trị giỏ hàng giảm về 0 | | Giá trị giỏ hàng không đổi | | | Fail | | |  |

Bảng 6. 6 Testcase xóa sản phẩm ra khỏi giỏ hàng

* Kiểm tra khả năng lưu lại đơn hàng sau khi đăng xuất

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | GR4\_User\_AddToCart\_TestCase34 | **Mô tả TestCase** | | Kiểm tra khả năng lưu lại đơn hàng sau khi đăng xuất | | | | | |
| **Người tạo :** | | Phan Hoàng Phúc | **Người đánh giá:** | | Lê Nguyễn Thanh Nhân | | **Phiên bản** | | 1.0 | |
|  |  |  |  |  |  |  |  |  |  |  |
| **QA Tester’s Log** | |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Tên người Test** | | Phúc | **Ngày Test** | | November 15, 2021 | | **Kết quả TestCase(Pass/Fail/Not Executed)** | | Pass | |
|  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Điều kiện tiên quyết** | | |  | **S #** | **Dữ liệu thử** | | | | |
| 1 | FrontEnd đã có thể chạy được | | |  | 1 | "Email": phancaocuong0000@gmail.com | | | | |
| 2 | BackEnd đã có thể chạy được | | |  | 2 | "Mật khẩu" : cuong123 | | | | |
| 3 | Port 3000 và Port 5000 đang để trống. | | |  | 3 | "Size" : S | | | | |
| 4 |  | | |  | 4 | "Số lượng" :2 | | | | |
|  |  |  |  |  | 5 | "Nhân" :nhân 1 | | | | |
| **Kịch bản Test** |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Các bước** | **Các bước chi tiết** | | **Kết quả mong đợi** | | **Kết quả thực tế** | | | **Pass / Fail / Not executed / Suspended** | | |
|  |
| 1 | Khởi động chương trình | | Trang chủ của Website sẽ hiện lên với những thông tin về trang chủ. | | Như kết quả mong đợi | | | Pass | | |  |
| 2 | Nhấn chuột vào dòng chữ đăng nhập bên góc tay phải màn hình, cạnh biểu tượng avartar | | Giao diện đăng nhậptài khoản của Website sẽ được hiện ra | | Như kết quả mong đợi | | | Pass | | |  |
| 3 | Nhập thông tin tương ứng của Dữ liệu Test vào từng box phù hợp. | | Dữ liệu sẽ được nhập đúng (Mật khẩu và Xác nhận phải được ẩn) | | Như kết quả mong đợi | | | Pass | | |  |
| 4 | Nhấn nút đăng nhập | | Vòng waiting sẽ xuất hiện và trả về trang chủ | | Như kết quả mong đợi | | | Pass | | |  |
| 5 | Tại màn hình trang chủ có thay đổi | | Tại màn hình User thay vì dòng chữ đăng nhập nay đổi thành hình đại diện và thông tin người dùng được xổ xuống bên tay phải | | Như kết quả mong đợi | | | Pass | | |  |
| 6 | Nhấn vào "OrderNow" trên thanh Slide giới thiệu sản phẩm | | Trang chủ sẽ nhảy sang trang Product và load lên màn hình toàn bộ sản phẩm có thể mua | | Như kết quả mong đợi | | | Pass | | |  |
| 7 | Nhấn vào bất kỳ một sản phẩm để xem thông tin sản phẩm | | Xuất hiện 1 modal chứa thông tin sản phẩm lên màn hình người dùng để họ xem thông tin chi tiết | | Như kết quả mong đợi | | | Pass | | |  |
| 8 | Thay đổi Option Size và Option số lượng | | Giá sản phẩm thay đổi đúng theo những Option mà người dùng lựa chọn | | Như kết quả mong đợi | | | Pass | | |  |
| 9 | Nhấn "Add to Cart" | | Thêm sản phẩm đã chọn vào giỏ hàng, modal lựa sản phẩm tắt, giỏ hàng thay đổi giá tiền bằng số tiền sản phẩm có trong giỏ | | Như kết quả mong đợi | | | Pass | | |  |
| 10 | Nhấn vào biểu tượng giỏ hàng bên góc trên tay phải | | Giá trị giỏ hàng được thay đổi theo số tiền thêm | | Như kết quả mong đợi | | | Pass | | |  |
| 11 | Nhấn vào avartar, chọn logout để đăng xuất | | Trang trở về trang đăng nhập cũ | | Như kết quả mong đợi | | | Pass | | |  |
| 12 | Thực hiện lại các bước đăng nhập 3 -4, kiểm tra lại giá trị giỏ hàng | | Giá trị giỏ hàng giữ nguyên với bước 10 | | Như kết quả mong đợi | | | Pass | | |  |

Bảng 6. 7 Testcase lưu đơn hàng sau đăng xuất

* Kiểm tra khả năng cập nhật lại đơn hàng cũ với thông tin sản phẩm không đổi

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| * **Test Case ID** | | GR4\_User\_AddToCart\_TestCase35 | **Mô tả TestCase** | | Kiểm tra khả năngcập nhật lại đơn hàng cũ với thông tin sản phẩm không đổi | | | | | |
| **Người tạo :** | | Phan Hoàng Phúc | **Người đánh giá:** | | Lê Nguyễn Thanh Nhân | | **Phiên bản** | | 1.0 | |
|  |  |  |  |  |  |  |  |  |  |  |
| **QA Tester’s Log** | |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Tên người Test** | | Phúc | **Ngày Test** | | November 15, 2021 | | **Kết quả TestCase(Pass/Fail/Not Executed)** | | Pass | |
|  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Điều kiện tiên quyết** | | |  | **S #** | **Dữ liệu thử** | | | | |
| 1 | FrontEnd đã có thể chạy được | | |  | 1 | "Email": phancaocuong0000@gmail.com | | | | |
| 2 | BackEnd đã có thể chạy được | | |  | 2 | "Mật khẩu" : cuong123 | | | | |
| 3 | Port 3000 và Port 5000 đang để trống. | | |  | 3 | "Size" : S | | | | |
| 4 |  | | |  | 4 | "Số lượng" :1 | | | | |
|  |  |  |  |  | 5 | "Size" : S | | | | |
| **Kịch bản Test** |  |  |  |  | 6 | "Số lượng" :10 | | | | |
|  |  |  |  |  |  |  |  |  |  |  |
| **Các bước** | **Các bước chi tiết** | | **Kết quả mong đợi** | | **Kết quả thực tế** | | | **Pass / Fail / Not executed / Suspended** | | |
|  |
| 1 | Khởi động chương trình | | Trang chủ của Website sẽ hiện lên với những thông tin về trang chủ. | | Như kết quả mong đợi | | | Pass | | |  |
| 2 | Nhấn chuột vào dòng chữ đăng nhập bên góc tay phải màn hình, cạnh biểu tượng avartar | | Giao diện đăng nhậptài khoản của Website sẽ được hiện ra | | Như kết quả mong đợi | | | Pass | | |  |
| 3 | Nhập thông tin tương ứng của Dữ liệu Test vào từng box phù hợp. | | Dữ liệu sẽ được nhập đúng (Mật khẩu và Xác nhận phải được ẩn) | | Như kết quả mong đợi | | | Pass | | |  |
| 4 | Nhấn nút đăng nhập | | Vòng waiting sẽ xuất hiện và trả về trang chủ | | Như kết quả mong đợi | | | Pass | | |  |
| 5 | Tại màn hình trang chủ có thay đổi | | Tại màn hình User thay vì dòng chữ đăng nhập nay đổi thành hình đại diện và thông tin người dùng được xổ xuống bên tay phải | | Như kết quả mong đợi | | | Pass | | |  |
| 6 | Nhấn vào "OrderNow" trên thanh Slide giới thiệu sản phẩm | | Trang chủ sẽ nhảy sang trang Product và load lên màn hình toàn bộ sản phẩm có thể mua | | Như kết quả mong đợi | | | Pass | | |  |
| 7 | Nhấn vào bất kỳ một sản phẩm để xem thông tin sản phẩm | | Xuất hiện 1 modal chứa thông tin sản phẩm lên màn hình người dùng để họ xem thông tin chi tiết | | Như kết quả mong đợi | | | Pass | | |  |
| 8 | Thay đổi Option Size và Option số lượng ờ ô dữ liệu 3-4 | | Giá sản phẩm thay đổi đúng theo những Option mà người dùng lựa chọn | | Như kết quả mong đợi | | | Pass | | |  |
| 9 | Nhấn "Add to Cart" | | Thêm sản phẩm đã chọn vào giỏ hàng, modal lựa sản phẩm tắt, giỏ hàng thay đổi giá tiền bằng số tiền sản phẩm có trong giỏ. | | Như kết quả mong đợi | | | Pass | | |  |
| 10 | Kiểm tra giá tiền | | Nếu dùng dữ liệu thử, số tiền sẽ là 90k Size S, số lượng 1 | | Như kết quả mong đợi | | | Pass | | |  |
| 11 | Nhấn vào thêm sản phẩm đã lựa chọntại bước 7 | | Xuất hiện 1 modal chứa thông tin sản phẩm lên màn hình người dùng để họ xem thông tin chi tiết | | Như kết quả mong đợi | | | Pass | | |  |
| 12 | Thay đổi Option Size và Option số lượng ờ ô dữ liệu 5-6 | | Giá sản phẩm thay đổi đúng theo những Option mà người dùng lựa chọn. | | Như kết quả mong đợi | | | Pass | | |  |
| 13 | Nhấn "Add to Cart" | | Thêm sản phẩm đã chọn vào giỏ hàng, modal lựa sản phẩm tắt, giỏ hàng thay đổi giá tiền bằng số tiền sản phẩm có trong giỏ. | | Như kết quả mong đợi | | | Pass | | |  |
| 14 | Kiểm tra giá tiền | | Giá tiền sẽ được cộng thêm 10 sản phẩm, giá tiền được tăng lên 990k | | Như kết quả mong đợi | | | Pass | | |  |

Bảng 6. 8 Testcase cập nhật đơn hàng cũ với thông tin sản phẩm không đổi

* Kiểm tra khả năng cập nhật lại đơn hàng cũ với cùng sản phẩm nhưng khác thông tin đi kèm

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | GR4\_User\_AddToCart\_TestCase36 | **Mô tả TestCase** | | Kiểm tra khả năng cập nhật lại đơn hàng cũ với cùng sản phẩm nhưng khác thông tin đi kèm | | | | | |
| **Người tạo :** | | Phan Hoàng Phúc | **Người đánh giá:** | | Lê Nguyễn Thanh Nhân | | **Phiên bản** | | 1.0 | |
|  |  |  |  |  |  |  |  |  |  |  |
| **QA Tester’s Log** | |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Tên người Test** | | Phúc | **Ngày Test** | | November 15, 2021 | | **Kết quả TestCase(Pass/Fail/Not Executed)** | | Pass | |
|  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Điều kiện tiên quyết** | | |  | **S #** | **Dữ liệu thử** | | | | |
| 1 | FrontEnd đã có thể chạy được | | |  | 1 | "Email": phancaocuong0000@gmail.com | | | | |
| 2 | BackEnd đã có thể chạy được | | |  | 2 | "Mật khẩu" : cuong123 | | | | |
| 3 | Port 3000 và Port 5000 đang để trống. | | |  | 3 | "Size" : S | | | | |
| 4 |  | | |  | 4 | "Số lượng" :1 | | | | |
|  |  |  |  |  | 5 | "Size" : L | | | | |
| **Kịch bản Test** |  |  |  |  | 6 | "Số lượng" :10 | | | | |
|  |  |  |  |  |  |  |  |  |  |  |
| **Các bước** | **Các bước chi tiết** | | **Kết quả mong đợi** | | **Kết quả thực tế** | | | **Pass / Fail / Not executed / Suspended** | | |
|  |
| 1 | Khởi động chương trình | | Trang chủ của Website sẽ hiện lên với những thông tin về trang chủ. | | Như kết quả mong đợi | | | Pass | | |  |
| 2 | Nhấn chuột vào dòng chữ đăng nhập bên góc tay phải màn hình, cạnh biểu tượng avartar | | Giao diện đăng nhậptài khoản của Website sẽ được hiện ra | | Như kết quả mong đợi | | | Pass | | |  |
| 3 | Nhập thông tin tương ứng của Dữ liệu Test vào từng box phù hợp. | | Dữ liệu sẽ được nhập đúng (Mật khẩu và Xác nhận phải được ẩn) | | Như kết quả mong đợi | | | Pass | | |  |
| 4 | Nhấn nút đăng nhập | | Vòng waiting sẽ xuất hiện và trả về trang chủ | | Như kết quả mong đợi | | | Pass | | |  |
| 5 | Tại màn hình trang chủ có thay đổi | | Tại màn hình User thay vì dòng chữ đăng nhập nay đổi thành hình đại diện và thông tin người dùng được xổ xuống bên tay phải | | Như kết quả mong đợi | | | Pass | | |  |
| 6 | Nhấn vào "OrderNow" trên thanh Slide giới thiệu sản phẩm | | Trang chủ sẽ nhảy sang trang Product và load lên màn hình toàn bộ sản phẩm có thể mua | | Như kết quả mong đợi | | | Pass | | |  |
| 7 | Nhấn vào bất kỳ một sản phẩm để xem thông tin sản phẩm | | Xuất hiện 1 modal chứa thông tin sản phẩm lên màn hình người dùng để họ xem thông tin chi tiết | | Như kết quả mong đợi | | | Pass | | |  |
| 8 | Thay đổi Option Size và Option số lượng ờ ô dữ liệu 3-4 | | Giá sản phẩm thay đổi đúng theo những Option mà người dùng lựa chọn | | Như kết quả mong đợi | | | Pass | | |  |
| 9 | Nhấn "Add to Cart" | | Thêm sản phẩm đã chọn vào giỏ hàng, modal lựa sản phẩm tắt, giỏ hàng thay đổi giá tiền bằng số tiền sản phẩm có trong giỏ. | | Như kết quả mong đợi | | | Pass | | |  |
| 10 | Kiểm tra giá tiền | | Nếu dùng dữ liệu thử, số tiền sẽ là 90k Size S, số lượng 1 | | Như kết quả mong đợi | | | Pass | | |  |
| 11 | Nhấn vào thêm sản phẩm đã lựa chọn tại bước 7 | | Xuất hiện 1 modal chứa thông tin sản phẩm lên màn hình người dùng để họ xem thông tin chi tiết | | Như kết quả mong đợi | | | Pass | | |  |
| 12 | Thay đổi Option Size và Option số lượng ờ ô dữ liệu 5-6 | | Giá sản phẩm thay đổi đúng theo những Option mà người dùng lựa chọn. | | Như kết quả mong đợi | | | Pass | | |  |
| 13 | Nhấn "Add to Cart" | | Thêm sản phẩm đã chọn vào giỏ hàng, modal lựa sản phẩm tắt, giỏ hàng thay đổi giá tiền bằng số tiền sản phẩm có trong giỏ. | | Như kết quả mong đợi | | | Pass | | |  |
| 14 | Kiểm tra giá tiền | | Trong giỏ hàng sẽ có 2 sản phẩm 1 nhưng 2 option khác nhau, số tiền được tính tổng các sản phẩm trong giỏ hàng | | Như kết quả mong đợi | | | Pass | | |  |

Bảng 6. 9 Testcase cập nhật lại đơn hàng cũ với cùng sản phẩm nhưng khác thông tin đi kèm

## Kiểm thử WhiteBox

### Chức năng Login

* ***Xác định node trong source code***

const login = async ({email, password}) => {

const user = await Users.findOne({ email }) (1)

const isMatch = await bcrypt.compare(password, user.password) (2)

if (!isMatch || !user.email) { (3)

return {

result: false,

msg: 'Email or password incorrect ',

data: '',

} (4)

} else if (!user.isEmailVerified) { (5)

return {

result: false,

msg: 'Email is not verify ',

data: '',

} (6)}

const token = await generateAuthTokens(user) (7)

return {

result: true,

msg: 'Verify your email address',

data: 'Done',

} (8)

}

* ***Vẽ đồ thị luồng dữ liệu***

8

7

6

4

5

3

1

2

3

5

4

6

7

8

Hình 6. 1 Đồ thị luồng dữ liệu đăng ký

* ***Xác định độ phức tạp***: Đồ thị bên có 2 nút quyết định nhị phân nên có độ phức tạp C = 2+1 = 3.Vậy có ít nhất là 3 test case có thể bao phủ 100% các nhánh
* ***Dựng bảng*** ***TestCase cho từng nhánh***

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **TestCase** | **Dữ liệu**  **(email;password)** | **Kết quả kỳ vọng** |
| **1** | 1 🡪 2 🡪 3 🡪 5🡪 7🡪 8 | nhanle1862@gmail.com;nhan123 | {result: true,  msg: 'Verify your email address',  data: 'Done', } |
| **2** | 1 🡪 2 🡪 3 🡪 4 | doanhhoanggmail.com;doho123 | {result: false,  msg: 'Email or password incorrect ',  data: '',} |
| **3** | 1 🡪 2 🡪 3 🡪 5 🡪 6 | tudu@gmail.com;tudu123 | { result: false,  msg: 'Email is not verify ',  data: '',} |

Bảng 6. 10 TestCase cho từng nhánh

* ***Đồ thị dòng dữ liệu***

k

d(token), u(token)

d(user), u(user), d(email), d(password)

,

d(isMatch), u(isMatch)

1

2

3

5

4

6

7

8

Hình 6. 2 Đồ thị dòng dữ liệu

* ***Kiểm tra vòng đời từng biến:***
* Biến Email:

Diagram

Description automatically generated

Hình 6. 3 Kiểm tra vòng đời biến email

* Biến Password:

Diagram

Description automatically generated

Hình 6. 4 Kiểm tra vòng đời biến password

* Biến User

Diagram

Description automatically generated

Hình 6. 5 Kiểm tra vòng đời biến user

* Biến isMatch

Diagram

Description automatically generated

Hình 6. 6 Kiểm tra vòng đời biến match

### Chức năng UpdateUser

* ***Xác định node trong source code***

const updateUser = async ({ name, avatar }) => {

if (!name || !avatar) (1)

return {

result: false,

data: ‘No data’

} (2)

const user = await Users.findOneAndUpdate(

{ \_id: data.user.id },

{

name,

avatar,

}

) (3)

return {

result: true,

user: user,

data: ‘Changed Successfully’

} (4)

* **Vẽ đồ thị luồng dữ liệu**

**A picture containing diagram

Description automatically generated**

Hình 6. 7 Đồ thị luồng dữ liệu update user

* ***Xác định độ phức tạp***: Đồ thị bên có 1 nút quyết định nhị phân nên có độ phức tạp C = 1+1 = 2.Vậy có ít nhất là 2 test case có thể bao phủ 100% các nhánh
* ***Dựng bảng*** ***TestCase cho từng nhánh***

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **TestCase** | **Dữ liệu**  **(name;avatar;)** | **Kết quả kỳ vọng** |
| **1** | 1 🡪 2 | null;https://upload.wikimedia.org/wikipedia/  commons/thumb/b/b6/  Image\_created\_with\_a\_mobile\_phone.png/  1200px-Image\_created\_with\_a\_mobile\_phone.png | {  result: false,  data: ‘No data’  } |
| **2** | 1 🡪 3 🡪 4 | Tu Du;https://www.w3schools.com/  w3css/img\_lights.jpg | {  result: true,  user: user,  data: ‘Changed Successfully’  } |

Bảng 6. 11 TestCase cho từng nhánh

* ***Đồ thị dòng dữ liệu***

Diagram

Description automatically generated

Bảng 6. 12 Đồ thị dòng dữ liệu

* ***Kiểm tra vòng đời từng biến:***
* Biến Name:

Diagram

Description automatically generated

Hình 6. 8 Kiểm tra vòng đời biến name

* Biến Avatar:

Diagram

Description automatically generated

Hình 6. 9 Kiểm tra vòng đời biến avatar

* Biến User:

Diagram, schematic

Description automatically generated

Hình 6. 10 Kiểm tra vòng đời biến user

### Chức năng Active Email

* ***Xác định node trong source code***

const activateEmail = async (data) => {

const userId = jwt.verify(data, process.env.PASSPORT\_JWT) (1)

const sub = userId.sub (2)

const result = await Users.findOneAndUpdate(

{

\_id: sub,

},

{

isEmailVerified: true,

}

) (3)

if (result) { (4)

return {

result: true,

msg: 'Successfully'

} (5)

}

return {

result: false,

msg: 'Failed'

} (6)

}

* **Vẽ đồ thị luồng dữ liệu**

**Diagram

Description automatically generated**

Hình 6. 11 Đồ thị luồng dữ liệu active mail

* ***Xác định độ phức tạp***: Đồ thị bên có 1 nút quyết định nhị phân nên có độ phức tạp C = 1+1 = 2.Vậy có ít nhất là 2 test case có thể bao phủ 100% các nhánh
* ***Dựng bảng TestCase cho từng nhánh***

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **TestCase** | **Dữ liệu**  **(data)** | **Kết quả kỳ vọng** |
| **1** | 1 🡪 2 🡪 3 🡪 4 🡪 5 | eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.  eyJzdWIiOiIxMjM0NTY3ODkwIn0.  P9Klv\_0x0vk32T1z4PFqZaKDeF2DFacQJiOxqjXGS48 | {  result: true,  msg: 'Successfully'  } |
| **2** | 1 🡪 3 🡪 4 🡪 6 | eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.  eyJzdWIiOiIwMTIzMTIzMzIifQ.  Jx21bcjHZjPR9E4H1mH3h\_-Fl3PX853OinFpbUWPsFY | {  result: false,  msg: 'Failed'  } |

Bảng 6. 13 TestCase cho từng nhánh

* ***Đồ thị dòng dữ liệu***

Diagram

Description automatically generated

Bảng 6. 14 Đồ thị dòng dữ liệu

* ***Kiểm tra vòng đời từng biến:***
* Biến Data:

Diagram

Description automatically generated

Hình 6. 12 Kiểm tra vòng đời biến data

* Biến UserId:

Diagram

Description automatically generated

Hình 6. 13 Kiểm tra vòng đời biến userId

* Biến Sub:

Diagram

Description automatically generated

Hình 6. 14 Kiểm tra vòng đời biến sub

* Biến result:

Diagram

Description automatically generated

Hình 6. 15 Kiểm tra vòng đời biến result

## Đánh giá

* Back-end
* Back-end luôn khả dụng
* Cung cấp đủ các APIs để sử dụng cho các chức năng đã được liệt kê trong usecase
* Chưa cung cấp được các APIs cho phát triển cho tương lai
* Request và response với tốc độ nhanh
* Front-end
* Giao diện tương đối đẹp, dễ nhìn sử dụng được các thư viện tốt nhất hiện nay như Boostrap, Material
* Nhìn chung các chức năng tương đối hoàn thiện.
* Vẫn còn một số chức năng chưa hoàn thành: Chỉnh sửa thông tin cá nhân, quản lý đơn hàng
* Khả năng bảo mật chưa tốt

# KẾT LUẬN

## Đánh giá những kết quả đã thực hiện được

* Database:

Tạo database bằng code first sử dụng MongoDB Clound và connect tới back-end

* Back-end
* Xây dựng back-end theo mô hình Model-Controller-Route
* Tạo những APIs cần sử dụng cho dự án
* Front-end
* Hiểu được cách vận hành của ReactJS
* Sử dụng được ReactJS để thiết kế giao diện chương tình
* Hoàn thành một số chức năng chính của chương trình
* Kinh nghiệm nhận được
* Hiểu được cách khởi tạo, xây dựng và vận hành một chương trình
* Hiểu được mô hình workflow giữa Front-end với Back-end và database
* Dựng được khung chương trình
* Học thêm được kiến thức mới và các cách khắc phục sự cố khi code

## Đánh giá quá trình thực hiện và phân tích kết quả

### Ưu điểm

* Hiểu biết thêm về công nghệ mới MERN STACK
* Hiệu suất chương trình khá tốt
* Cấu trúc chương trình được dựng chỉn chu, phân biệt các thành phần với nhau. Từ đó dễ bảo trì và fix bugs khi có sự cố xảy ra
* Chương trình có thông báo realtimes đảm bảo cập nhật trạng thái mới nhất đến khách hàng
* Có chức năng GroupOrder – chức năng mới được mà ít mô hình thực tiễn áp dụng được

### Nhược điểm

* Chương trình vẫn chưa hoàn thiện và trong quá trình phát triển
* Chưa xử lý được trình trạng nếu không có internet thì dữ liệu chưa đồng bộ tốt
* Lần đầu sử dụng công nghệ mới nên còn vướng nhiều vấn đề khi tiếp cận
* Nghiệp vụ chương trình chưa đầy đủ lắm. Còn thiếu phần quản lý

## Định hướng phát triển

* Phát triển dự án them phần quản lý và phát triển dự án thành hệ thống bán thức ăn nhanh đa chi nhánh
* Phân cấp user chi tiết hơn
* Nâng cao bảo mật và hiệu năng cho chương trình
* Cập nhật lại tính năng chọn địa chỉ và tích hợp AI vào việc phân tích xu hướng người dùng
* Phát triển các tính năng mới về phục vụ nghiệp vụ quản lý nhiều hơn
* Chỉnh sửa UI cho phù hợp hơn

**BẢNG PHÂN CÔNG**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Công việc** | **Thành viên** | **Đánh giá** |
| 1 | Phát triển Front-end  Viết báo cáo | Đinh Tấn Tú | Tốt |
| 2 | Phát triển Back-end  Vẽ sơ đồ usecase, ERD,… | Phan Cao Cường | Tốt |
| 3 | Phát triển Back-end  Hỗ trợ fixbugs, testing  Xây dựng khung chương trình | Lê Nguyễn Thanh Nhân | Tốt |

Bảng 7. 1 Bảng phân công nhiệm vụ

**TÀI LIỆU THAM KHẢO**

---------------------------\*\*\*\*\*\*\*\*\*\*-------------------------

* + - 1. ReactJS là gì – Căn bản về ReactJS
      2. https://itnavi.com.vn/blog/reactjs-la-gi-can-ban-ve-reactjs
      3. MongoDB là gì ?
      4. https://hostingviet.vn/mongodb-la-gi
      5. Tìm hiểu về NodeJS và những ưu nhược điểm của NodejS
      6. https://vn.got-it.ai/blog/tim-hieu-ve-nodejs-va-nhung-uu-nhuoc-diem-cua-nodejs
      7. ExpressJS là gì
      8. https://itnavi.com.vn/blog/expressjs-la-gi/?amp