TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH

KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN CƠ SỞ

*Tên đề tài:*

PHÂN TÍCH FLUTTER VÀ THIẾT KẾ

ỨNG DỤNG DI DỘNG BÁN ĐIỆN THOẠI

|  |  |
| --- | --- |
| Giảng viên hướng dẫn : | ThS. ĐẶNG THẾ HÂN |
| Sinh viên thực hiện: | NGUYỄN LÂM DUY |
| MSSV: | 2200006652 |
| Chuyên ngành: | Kỹ thuật phần mềm |
| Môn học: | Đồ án cơ sở |
| Khoá: | 2022 |

Tp HCM, tháng 9 năm 2024

TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH

KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN CƠ SỞ

*Tên đề tài:*

PHÂN TÍCH FLUTTER VÀ THIẾT KẾ

ỨNG DỤNG DI DỘNG BÁN ĐIỆN THOẠI

|  |  |
| --- | --- |
| Giảng viên hướng dẫn : | ThS. ĐẶNG THẾ HÂN |
| Sinh viên thực hiện : | 1. NGUYỄN LÂM DUY 2. NGUYỄN HÀ LAN ANH |
| MSSV : | 2200006652 - 2200006286 |
| Chuyên ngành : | Kỹ thuật phần mềm |
| Môn học : | Đồ án cơ sở |
| Khoá : | 2022 |

Tp HCM, tháng 9 năm 2024

|  |  |
| --- | --- |
| Trường Đại học Nguyễn Tất Thành      Khoa Công Nghệ Thông Tin  🙜  🙜  🙝  🙝 | CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM    Độc lập – Tự do – Hạnh phúc  🙜  🙜  🙝  🙝 |
|  |  |

NHIỆM VỤ ĐỒ ÁN CƠ SỞ

|  |  |
| --- | --- |
| Họ và tên: NGUYỄN LÂM DUY | MSSV: 2200006652 |
| Chuyên ngành: KỸ THUẬT PHẦN MỀM | Lớp: 22DKTPM1D |
| Email: [2200006652@nttu.edu.vn](mailto:2200006652@nttu.edu.vn) | SĐT: 0398845375 |
| Họ và tên: NGUYỄN HÀ LAN ANH | MSSV: 2200006286 |
| Chuyên ngành: KỸ THUẬT PHẦN MỀM | Lớp: 22DKTPM1D |
| Email: 2200006286@nttu.edu.vn | SĐT: 0375663540 |

1. Tên đề tài

PHÂN TÍCH FLUTTER VÀ THIẾT KẾ ỨNG DỤNG DI ĐỘNG BÁN ĐIỆN THOẠI

1. Giảng viên hướng dẫn: ThS. Đặng Thế Hân
2. Thời gian thực hiện: 22/6/2024 đến 21/9/2024
3. Nhiệm vụ / nội dung ( mô tả chi nội dung, yêu cầu, phương pháp, …)

* Xây dựng ứng dụng đa nền tảng Flutter dựa trên ngôn ngữ lập trình Dart.

LỜI CẢM ƠN

Tôi chân thành cám ơn Thầy hướng dẫn Đặng Thế Hân, người đã hướng dẫn tận tình, đóng góp các ý kiến chuyên môn đồ án. Thầy đã động viên tinh thần tôi cố gắng hoàn thành những nghiên cứu đặt ra. Thầy cũng cung cấp một số tài liệu liên quan đến đồ án mà tôi đang nghiên cứu và ân cần nhắc nhở tôi đến tiến độ thực hiện đồ án này.

Tôi cũng chân thành gởi lời cám ơn đến các Thầy Cô khoa Công nghệ thông tin trường Đại học Nguyễn Tất Thành đã giảng dạy, hướng dẫn, trang bị các kiến thức cho tôi trong những năm học vừa qua, từ các kiến thức cơ bản đến các vấn đề chuyên sâu.

Tôi xin gởi lời cám ơn đến anh, chị, bạn bè, đồng nghiệp bằng nhiều hình thức khác nhau đã giúp đỡ tôi trong quá trình học tập tại trường cũng như trong thời gian hoàn thành đồ án.

Đặc biệt xin gởi lời cám ơn đến cha mẹ và anh chị em trong gia đình đã động viên tinh thần cũng như chia sẻ những khó khăn trong những năm qua.

LỜI MỞ ĐẦU

Trong thời đại kỹ thuật số hiện nay, “Công nghệ thông tin” không chỉ là thuật ngữ thông dụng mà còn trở thành nền tảng quan trọng trong mọi khía cạnh của đời sống kinh tế và xã hội. Công nghệ thông tin đã mang lại những biến đổi đáng kể, từ cách thức hoạt động của các doanh nghiệp lớn đến những thói quen hàng ngày của chúng ta, đồng thời mở ra hướng đi mới cho sự sáng tạo và phát triển.

Xu hướng việc làm trong nhóm ngành Công nghệ thông tin (Computer Science) hiện nay đang tập trung vào một số lĩnh vực chính như phát triển phần mềm, khoa học dữ liệu, chuyên gia trí tuệ nhân tạo và học máy, quản lý IT, phân tích an ninh thông tin, kiến trúc sư hệ thống máy tính, kỹ sư độ tin cậy của trang web, và kỹ sư DevOps. Những vị trí này đều có mức lương trung bình cao và đều dự kiến sẽ có sự tăng trưởng mạnh mẽ về nhu cầu trong những năm tới. Điều này phản ánh sự phát triển không ngừng của công nghệ và nhu cầu ngày càng cao về chuyên môn trong lĩnh vực IT​.

Ứng dụng di động là một lĩnh vực chuyên biệt trong ngành công nghệ phần mềm, chú trọng vào việc thiết kế, tạo lập và duy trì các ứng dụng được tối ưu hóa cho các thiết bị di động như điện thoại thông minh và máy tính bảng. Quy trình này sử dụng nhiều ngôn ngữ lập trình, khung và công cụ khác nhau. Java và Kotlin là những ngôn ngữ phổ biến nhất được sử dụng để phát triển ứng dụng Android (với các công cụ như Android Studio), trong khi Swift và Objective-C được sử dụng để phát triển ứng dụng iOS.

Khảo sát hiện trạng này nhằm mục đích cung cấp cái nhìn tổng quan về sự phát triển của Flutter trong lĩnh vực ứng dụng di động và thương mại điện tử, đồng thời nhấn mạnh tính cạnh tranh và tiềm năng mà nền tảng công nghệ này mang lại. Qua đó, chúng ta sẽ hiểu rõ hơn về lợi ích và cơ hội của việc nghiên cứu và phát triển dự án trên nền tảng Flutter trong bối cảnh công nghệ thông tin hiện đại.

|  |  |
| --- | --- |
| TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH  **TRUNG TÂM KHẢO THÍ** | **KỲ THI KẾT THÚC HỌC PHẦN**  **HỌC KỲ 3 NĂM HỌC 2023 - 2024** |

**PHIẾU CHẤM THI TIỂU LUẬN/ĐỒ ÁN**

BM-ChT-11

Môn thi: Đồ án cơ sở kỹ thuật phần mềm Lớp học phần: 22DKTPM1D

Nhóm sinh viên thực hiện :

1. Nguyễn Lâm Duy Tham gia đóng góp: Tìm hiểu, code chiếm 50%.

2. Nguyễn Hà Lan Anh Tham gia đóng góp: Tìm hiểu, thiết kế MongoDB chiếm 50%.

Ngày thi: 30/09/2024 Phòng thi: L.503

Đề tài tiểu luận/báo cáo của sinh viên :

**PHÂN TÍCH FLUTTER VÀ THIẾT KẾ ỨNG DỤNG DI ĐỘNG BÁN ĐIỆN THOẠI**

Phần đánh giá của giảng viên (căn cứ trên thang rubrics của môn học):

|  |  |  |  |
| --- | --- | --- | --- |
| Tiêu chí (theo CĐR HP) | Đánh giá của GV | Điểm tối đa | Điểm đạt được |
| Cấu trúc của báo cáo |  |  |  |
| Nội dung |  |  |  |
| * Các nội dung thành phần |  |  |  |
| * Lập luận |  |  |  |
| * Kết luận |  |  |  |
| - Trình bày |  |  |  |
| TỔNG ĐIỂM |  |  |  |

|  |  |
| --- | --- |
|  | Giảng viên chấm thi  *(ký, ghi rõ họ tên)* |

MỤC LỤC

[NHIỆM VỤ ĐỒ ÁN CƠ SỞ I](#_Toc177656357)

[LỜI CẢM ƠN II](#_Toc177656358)

[LỜI MỞ ĐẦU III](#_Toc177656359)

[MỤC LỤC V](#_Toc177656360)

[DANH MỤC BẢNG VIII](#_Toc177656361)

[DANH MỤC HÌNH IX](#_Toc177656362)

[KÍ HIỆU CÁC CỤM TỪ VIẾT TẮT X](#_Toc177656363)

[CHƯƠNG 1. TỔNG QUAN ĐỀ TÀI 1](#_Toc177656364)

[1.1. Giới thiệu 1](#_Toc177656365)

[1.2. Khảo sát hiện trạng 1](#_Toc177656366)

[1.3. Lí do chọn đề tài 2](#_Toc177656367)

[1.4. Mục tiêu đề tài 3](#_Toc177656368)

[1.5. Phạm vi đề tài 4](#_Toc177656369)

[1.6. Mô tả các đối tượng 5](#_Toc177656370)

[CHƯƠNG 2. CƠ SỞ LÍ THUYẾT 6](#_Toc177656371)

[2.1. Mô tả về ngôn ngữ Dart 6](#_Toc177656372)

[2.1.1. Giới thiệu về ngôn ngữ Dart 6](#_Toc177656373)

[2.1.2. Đặc điểm nổi bật của Dart 7](#_Toc177656374)

[2.1.3. Các nền tảng (platforms) thực thi: 7](#_Toc177656375)

[2.1.4. Các tính năng của Flutter 7](#_Toc177656376)

[2.1.5. Những kiểu dữ liệu phổ biến trong Dart 9](#_Toc177656377)

[2.1.6. Hướng đối tượng (OOP) trong Dart 10](#_Toc177656378)

[2.2. Mô tả về Framework Flutter 11](#_Toc177656379)

[2.2.1. Giới thiệu về Flutter 11](#_Toc177656380)

[2.2.2. Kiến trúc Flutter 11](#_Toc177656381)

[2.3. Mô tả cơ sở dữ liệu trong Flutter 13](#_Toc177656382)

[2.3.1. Server và API sử dụng Node.js. 13](#_Toc177656383)

[2.3.2. Cơ sở dữ liệu MongoDB 13](#_Toc177656384)

[2.4. Lựa chọn công cụ cho đề tài 14](#_Toc177656385)

[CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 14](#_Toc177656386)

[3.1. Yêu cầu chức năng 14](#_Toc177656387)

[3.1.1 Chức năng đăng nhập/đăng kí 14](#_Toc177656388)

[3.1.2 Chức năng tìm kiếm 14](#_Toc177656389)

[3.1.3 Chức năng quản lý giỏ hàng 15](#_Toc177656390)

[3.1.4 Chức năng thanh toán 15](#_Toc177656391)

[3.1.5 Chức năng đánh giá sản phẩm 15](#_Toc177656392)

[3.2. Yêu cầu phi chức năng 17](#_Toc177656393)

[3.3. Mô tả tổng quan hệ thống 18](#_Toc177656394)

[3.3.1. Biểu đồ Use – Case tổng quát 18](#_Toc177656395)

[3.3.2. Danh sách các tác nhân ( Actor) 18](#_Toc177656396)

[3.3.3. Danh sách chức năng và mô tả 19](#_Toc177656397)

[3.4. Đặc tả các yêu cầu chức năng 20](#_Toc177656398)

[3.4.1. Chức năng đăng kí, đăng nhập 20](#_Toc177656399)

[3.4.1.1. Biểu đồ tuần tự Đăng nhập 20](#_Toc177656400)

[3.4.1.2. Biểu đồ tuần tự Đăng kí 22](#_Toc177656401)

[3.4.2. Biểu đồ tuần tự tìm kiếm 23](#_Toc177656402)

[3.4.3. Biểu đồ tuần tự thêm sản phẩm vào giỏ hàng 23](#_Toc177656403)

[3.4.4. Biểu đồ tuần tự quản lý giỏ hàng 24](#_Toc177656404)

[3.4.5. Biểu đồ tuần tự chức năng đánh giá 25](#_Toc177656405)

[3.4.6. Biểu đồ tuần tự chức năng quản lý sản phẩm 25](#_Toc177656406)

[3.4.7. Biểu đồ tuần tự chức năng quản lý khách hàng 27](#_Toc177656407)

[3.5. Thiết kế Cơ sở dự liệu 27](#_Toc177656408)

[3.5.1 Mô hình thực thể chi tiết 28](#_Toc177656409)

[CHƯƠNG 4. CÀI ĐẶT CHƯƠNG TRÌNH VÀ TRIỂN KHAI 30](#_Toc177656410)

[4.1. Mô tả về hệ thống 30](#_Toc177656411)

[4.1.1. Xây dựng hệ thống 30](#_Toc177656412)

[4.1.2. Một số mã nguồn quan trọng của hệ thống 32](#_Toc177656413)

[4.2. Mô tả ứng dụng 34](#_Toc177656414)

[4.2.1 Website quản lý 34](#_Toc177656415)

[4.2.2 Ứng dụng di động 34](#_Toc177656416)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 39](#_Toc177656417)

[1. Kết quả đạt được 39](#_Toc177656418)

[1.1. Website quản lý 39](#_Toc177656419)

[2. Hướng phát triển 39](#_Toc177656420)

[PHỤ LỤC 42](#_Toc177656421)

[1. Link source code 42](#_Toc177656422)

[2. Code mẫu 42](#_Toc177656423)

[TÀI LIỆU THAM KHẢO 43](#_Toc177656424)

DANH MỤC BẢNG

[Bảng 1 Bảng khách thể 6](#_Toc177659599)

[Bảng 2 Bảng tác nhân 19](#_Toc177659600)

[Bảng 3 Bảng danh sách chức năng 19](#_Toc177659601)

[Bảng 4 Bảng tác nhân Đăng kí 22](#_Toc177659602)

[Bảng 5 Tác nhân đăng nhập 23](#_Toc177659603)

[Bảng 6 Bảng cơ sở dữ liệu 29](#_Toc177659604)

DANH MỤC HÌNH

[Hình 1 Các nền tảng (platforms) 7](#_Toc177659169)

[Hình 2 Các tính năng của Flutter 8](#_Toc177659170)

[Hình 3 Các dữ liệu cơ bản trong Dart 9](#_Toc177659171)

[Hình 4 Kiến trúc Flutter 12](#_Toc177659172)

[Hình 5 Server và API sử dụng node.js 13](#_Toc177659173)

[Hình 6 MongoDB 14](#_Toc177659174)

[Hình 7 Sơ đồ phân cấp nhóm chức năng mua hàng 16](#_Toc177659175)

[Hình 8 Sơ đồ phân cấp nhóm chức năng quản lí 17](#_Toc177659176)

[Hình 9 Biểu đồ Use – Case tổng quát 18](#_Toc177659177)

[Hình 10 Biểu đồ tuần tự chức năng đăng nhập 20](#_Toc177659178)

[Hình 11 Biểu đồ tuần tự chức năng đăng kí 22](#_Toc177659179)

[Hình 12 Biểu đồ tuần tự chức năng tìm kiếm 23](#_Toc177659180)

[Hình 13 Biểu đồ tuần tự thêm sản phẩm vào giỏ hàng 24](#_Toc177659181)

[Hình 14 Biểu đồ tuần tự chức năng quản lý giỏ hàng 24](#_Toc177659182)

[Hình 15 Biểu đồ tuần tự chức năng đánh giá 25](#_Toc177659183)

[Hình 16 Biểu đồ tuần tự chức năng quản lý sản phẩm 26](#_Toc177659184)

[Hình 17 Biểu đồ tuần tự chức năng quản lý khách hàng 27](#_Toc177659185)

[Hình 18 Cơ sở dữ liệu 28](#_Toc177659186)

[Hình 19 Class diagram 30](#_Toc177659187)

[Hình 20 Lớp data\_provider 31](#_Toc177659188)

[Hình 21 Lớp presentation 31](#_Toc177659189)

[Hình 22 Lớp Service 32](#_Toc177659190)

[Hình 23 Lớp API 32](#_Toc177659191)

[Hình 24 Giao diện website trang quản lý 35](#_Toc177659192)

[Hình 25 Giao diện đăng nhập và đăng kí 36](#_Toc177659193)

[Hình 26 Giao diện màn hình chính và yêu thích 37](#_Toc177659194)

[Hình 27 Giao diện chi tiết sản phẩm và giỏ hàng 38](#_Toc177659195)

[Hình 28 Giao diện trang cá nhân và thông tin 39](#_Toc177659196)

KÍ HIỆU CÁC CỤM TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| SDK | Software Development Kit | Bộ công cụ phát triển phần mềm |
| API | Application Programming Interface | Giao diện lập trình ứng dụng |
| JIT | Just In Time |  |
| AOT | Ahead Of Time | Khi chạy chương trình nó sẽ biên dịch từ đầu đến cuối |
| OOP | Object Oriented Programming | Lập trình hướng đối tượng |

CHƯƠNG 1. TỔNG QUAN ĐỀ TÀI

* 1. Giới thiệu

Khi chúng ta bước vào năm 2024, ngành phát triển ứng dụng di động nhanh chóng phát triển với những xu hướng mới đang nổi lên. Phát triển ứng dụng di động là một lĩnh vực hot nhất trong ngành công nghệ thông tin. Hiện nay, có rất nhiều các công cụ giúp xây dựng một ứng dụng di động tạo trải nghiệm tốt. Trong đó, Flutter là một trong những công cụ phổ biến. Với các đặc tính nổi bật và tốc độ phát triển nhanh chóng, Flutter là lựa chọn hàng đầu cho các Mobile development.

Các ứng dụng công nghệ không chỉ đơn thuần là công cụ hỗ trợ mà còn là yếu tố quyết định sự thành công của các tổ chức và doanh nghiệp. Chúng mở ra cánh cửa cho khả năng hoàn thành công việc mà trước đây có vẻ như không thể. Và trong tất cả những ứng dụng này, điện thoại thông minh đã nổi lên như một trong những biểu tượng rõ nét nhất của sự tiến bộ.

Điện thoại thông minh không chỉ đơn giản là một công cụ liên lạc. Nó là trung tâm của cuộc sống hàng ngày, giúp chúng ta kết nối, học tập và giải trí một cách tiện lợi và linh hoạt. Chức năng liên lạc trên điện thoại đã trải qua một sự chuyển đổi lớn, từ việc tốn kém thời gian và tiền bạc đến việc trở nên miễn phí và dễ dàng hơn bao giờ hết.

Trong bối cảnh này, việc xây dựng một ứng dụng mua sắm trên thiết bị di động không chỉ đơn thuần là một ý tưởng. Đó là cầu nối để kết nối con người với nhau, góp phần gắn kết mọi người lại gần nhau hơn, tạo ra thu thập tài chính. Điều này không chỉ tạo ra một ứng dụng mua sắm, mà còn là một cách để xây dựng cộng đồng, tạo nên một môi trường thu thập tài chính thuận lợi và gần gũi.

Với sự tiến bộ không ngừng của công nghệ, việc xây dựng những ứng dụng như vậy không chỉ là một bước đi quan trọng trong ngành công nghệ mà còn góp phần làm cho xã hội trở nên tiến bộ và gắn kết hơn.

* 1. Khảo sát hiện trạng

Hiện nay, việc phát triển ứng dụng di động đang trở thành một xu hướng không thể phủ nhận trong cả thế giới và đặc biệt là tại Việt Nam. Cùng với sự gia tăng mạnh mẽ của thị trường mua sắm trực tuyến, các ứng dụng thương mại điện tử đang ngày càng được ưa chuộng và trở thành một phần không thể thiếu trong cuộc sống hàng ngày của người tiêu dùng. Trong bối cảnh này, Flutter - một framework phát triển ứng dụng di động đa nền tảng của Google, đang thu hút sự quan tâm đặc biệt nhờ vào khả năng mang lại trải nghiệm người dùng mượt mà và hiệu quả cao.

Khảo sát hiện trạng này nhằm mục đích cung cấp cái nhìn tổng quan về sự phát triển của Flutter trong lĩnh vực ứng dụng di động và thương mại điện tử, đồng thời nhấn mạnh tính cạnh tranh và tiềm năng mà nền tảng công nghệ này mang lại. Qua đó, chúng ta sẽ hiểu rõ hơn về lợi ích và cơ hội của việc nghiên cứu và phát triển dự án trên nền tảng Flutter trong bối cảnh công nghệ thông tin hiện đại.

* 1. Lí do chọn đề tài

Việc lựa chọn đề tài "Tìm hiểu Flutter và xây dựng ứng dụng di động thương mại điện tử" là một quyết định có cơ sở và hợp lý. Flutter, với khả năng phát triển ứng dụng di động đa nền tảng vượt trội, đem lại cơ hội nghiên cứu sâu hơn về công nghệ mới trong lĩnh vực phát triển phần mềm di động. Việc áp dụng Flutter vào dự án thương mại điện tử không chỉ là việc khám phá một công nghệ tiên tiến mà còn là cơ hội để áp dụng những kiến thức lý thuyết thành hệ thống thực tế, từ thiết kế giao diện đến quản lý dữ liệu và tối ưu hóa hiệu suất ứng dụng.

Đặc biệt, đề tài này mang lại khả năng ứng dụng rộng rãi, từ việc xây dựng một ứng dụng có tính năng mua bán trực tuyến đến việc tạo ra một sản phẩm có thể cung cấp giá trị thực cho người dùng và tiềm năng thương mại cao. Việc nghiên cứu và phát triển trên nền tảng Flutter không chỉ giúp nắm bắt các kỹ năng kỹ thuật mới mà còn mở ra cơ hội khám phá và thử thách bản thân trong một dự án có tính ứng dụng cao và môi trường phát triển đầy thử thách.

Khi hoàn thành, ứng dụng thương mại điện tử sử dụng Flutter có thể không chỉ là một dự án học thuật mà còn là một sản phẩm thực tế, sẵn sàng tiếp cận thị trường và tạo ra giá trị thương mại. Qua quá trình này, không chỉ là cơ hội để nghiên cứu và áp dụng công nghệ mới mà còn là bước đầu tiên trong việc xây dựng một nền tảng kỹ thuật vững chắc và phát triển nghề nghiệp trong lĩnh vực công nghệ thông tin, đặc biệt là trong lĩnh vực phát triển phần mềm di động và thương mại điện tử.

* 1. Mục tiêu đề tài

Xây dựng một nền tảng nghiên cứu chuyên sâu về Flutter và một ứng dụng thực tiễn trong phát triển ứng dụng di động thương mại điện tử, từ đó mang lại những đóng góp quan trọng cho cả lĩnh vực công nghệ thông tin và thương mại điện tử. Gồm các phần như:

Nghiên cứu sâu về Flutter: Mục tiêu chính của đề tài là khám phá và nghiên cứu sâu về Flutter - một framework phát triển ứng dụng di động đa nền tảng từ Google. Điều này bao gồm việc tìm hiểu về kiến trúc của Flutter, lịch sử phát triển và các tính năng nổi bật mà nó mang lại. Chúng tôi sẽ so sánh Flutter với các công nghệ khác như Native và React Native để đánh giá ưu nhược điểm và tính phù hợp trong việc phát triển ứng dụng thương mại điện tử.

Xây dựng ứng dụng thương mại điện tử: Một trong những mục tiêu quan trọng của đề tài là phát triển một ứng dụng di động thương mại điện tử hoàn chỉnh sử dụng Flutter. Chúng tôi sẽ thiết kế và triển khai các tính năng quan trọng như đăng nhập, hiển thị danh sách sản phẩm, tìm kiếm, quản lý giỏ hàng và thanh toán. Mục đích là cung cấp cho người dùng trải nghiệm mua sắm trực tuyến mượt mà và thuận tiện.

Đánh giá hiệu suất và trải nghiệm người dùng: Chúng tôi sẽ tiến hành đánh giá và tối ưu hóa hiệu suất của ứng dụng sử dụng Flutter. Điều này bao gồm thử nghiệm và đánh giá thời gian phản hồi của ứng dụng trên các thiết bị khác nhau, đồng thời đo lường trải nghiệm người dùng để cải thiện và điều chỉnh các tính năng và giao diện của ứng dụng.

Nghiên cứu thực tiễn và ứng dụng thương mại: Mục tiêu tiếp theo là áp dụng các kết quả nghiên cứu vào thực tiễn bằng việc xây dựng một ứng dụng thương mại điện tử thực tế. Chúng tôi sẽ đánh giá khả năng cạnh tranh và tiềm năng phát triển của ứng dụng này trên thị trường, từ đó cung cấp những thông tin quý giá cho các doanh nghiệp và nhà phát triển quan tâm đến việc áp dụng Flutter vào lĩnh vực thương mại điện tử.

Chia sẻ kết quả và kinh nghiệm học hỏi: Cuối cùng, mục tiêu của chúng tôi là tổng kết kinh nghiệm và chia sẻ kết quả nghiên cứu để giúp cộng đồng phát triển ứng dụng di động và thương mại điện tử hiểu rõ hơn về tiềm năng và thách thức của việc sử dụng Flutter. Chúng tôi hy vọng rằng những thông tin này sẽ cung cấp những đóng góp ý nghĩa và giúp thúc đẩy sự phát triển bền vững của ngành công nghiệp công nghệ thông tin và thương mại điện tử.

* 1. Phạm vi đề tài

Nghiên cứu về Flutter:

* Phân tích và nghiên cứu sâu về Flutter như là một framework phát triển ứng dụng di động đa nền tảng của Google.
* Điều tra về kiến trúc, cơ chế hoạt động, và các thành phần chính của Flutter.
* So sánh Flutter với các công nghệ khác như Native và React Native để đánh giá ưu nhược điểm và tính phù hợp trong các trường hợp sử dụng khác nhau.

Xây dựng ứng dụng thương mại điện tử:

* Phát triển một ứng dụng di động thương mại điện tử hoàn chỉnh sử dụng Flutter làm nền tảng chính.
* Tập trung vào thiết kế và triển khai các tính năng cơ bản như đăng nhập, hiển thị danh sách sản phẩm, tìm kiếm, quản lý giỏ hàng và thanh toán.
* Đảm bảo ứng dụng có khả năng đáp ứng được yêu cầu về trải nghiệm người dùng và hiệu suất.

Đánh giá hiệu suất và tương thích:

* Tiến hành các bài kiểm tra và đánh giá hiệu suất của ứng dụng trên nhiều nền tảng và thiết bị khác nhau.
* Đo lường và phân tích khả năng tương thích của ứng dụng trên các phiên bản hệ điều hành và màn hình khác nhau.

Nghiên cứu thực tiễn và áp dụng thương mại:

* Áp dụng kiến thức và kết quả nghiên cứu vào việc xây dựng một ứng dụng thương mại điện tử thực tế.
* Đánh giá khả năng cạnh tranh và tiềm năng phát triển của ứng dụng trên thị trường thương mại điện tử hiện nay.

Chia sẻ kết quả và đề xuất:

* Tổng hợp kết quả nghiên cứu và trình bày những phát hiện, những hạn chế cũng như đề xuất cải tiến cho cộng đồng phát triển ứng dụng di động và thương mại điện tử.
* Chia sẻ kinh nghiệm và những bài học hữu ích từ quá trình nghiên cứu để khuyến khích sự phát triển bền vững của ngành công nghiệp công nghệ thông tin và thương mại điện tử.
  1. Mô tả các đối tượng

Đối tượng nghiên cứu: Flutter, Dart.

Khách thể nghiên cứu:

|  |  |  |  |
| --- | --- | --- | --- |
| Đối tượng | Mục đích | Hành vi | Yêu cầu |
| Người dùng | Là những người sử dụng ứng dụng để duyệt và mua các sản phẩm, dịch vụ trực tuyến | Tìm kiếm sản phẩm, đặt hàng, thanh toán và quản lý thông tin cá nhân. | Cần một giao diện người dùng thân thiện, tính năng mua hàng dễ dàng và an toàn. |
| Quản trị viên | Là người quản lý hệ thống và dữ liệu của ứng dụng. | Quản lý người dùng, sản phẩm, đơn hàng, quản lý nội dung và báo cáo hoạt động. | Cần một giao diện quản trị mạnh mẽ, bảo mật cao và tính năng báo cáo chi tiết. |
| Payment Gateway | Cung cấp dịch vụ thanh toán trực tuyến an toàn và hiệu quả | Xử lý các giao dịch thanh toán, bảo vệ thông tin tài khoản và hỗ trợ các phương thức thanh toán đa dạng. | Bảo mật cao, tích hợp dễ dàng với các hình thức thanh toán phổ biến và báo cáo giao dịch chi tiết. |
| Chatbox | Cung cấp hỗ trợ và giải quyết các vấn đề cho người dùng. | Hỗ trợ qua điện thoại, email, chat trực tuyến hoặc các kênh liên lạc khác, xử lý yêu cầu bảo hành, đổi trả. | Phản hồi nhanh chóng, hiệu quả và hỗ trợ tốt cho người dùng. |

Bảng 1 Bảng khách thể

CHƯƠNG 2. CƠ SỞ LÍ THUYẾT

* 1. Mô tả về ngôn ngữ Dart
     1. Giới thiệu về ngôn ngữ Dart

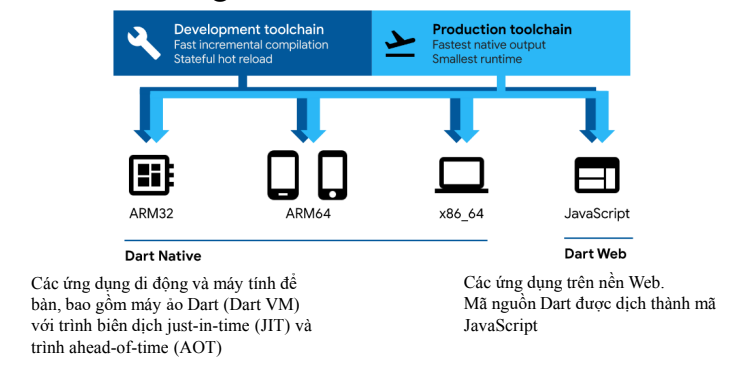
Dart là một ngôn ngữ lập trình mã nguồn mở (open source), đa năng (general purpose) do Google phát triển

Dart là một ngôn ngữ đa nền tảng (cross-platform) có thể được sử dụng để phát triển các ứng dụng web, máy tính bàn, máy chủ và thiết bị di động. Sự linh hoạt này giúp các lập trình viên dễ dàng xây dựng và triển khai ứng dụng trên nhiều nền tảng khác nhau từ một mã nguồn duy nhất.

Các ứng dụng Flutter được viết bằng Dart, mang lại trải nghiệm tốt nhất cho nhà phát triển. Dart cung cấp các công cụ, thư viện mạnh mẽ, hộ trợ biên dịch Ahead Of Time (AOT) và Just In Time (JIT) , giúp việc viết mã trở nên dễ dàng và hiệu quả, từ đó tạo ra các ứng dụng mượt mà và có hiệu suất cao.

* + 1. Đặc điểm nổi bật của Dart
* Công cụ hiệu quả (productive tooling)
* Thu gom rác (garbage collection)
* Chú thích kiểu (type annotations)
* Kiểu tĩnh (statically typing)
* Tính di động (portability)
  + 1. Các nền tảng (platforms) thực thi:

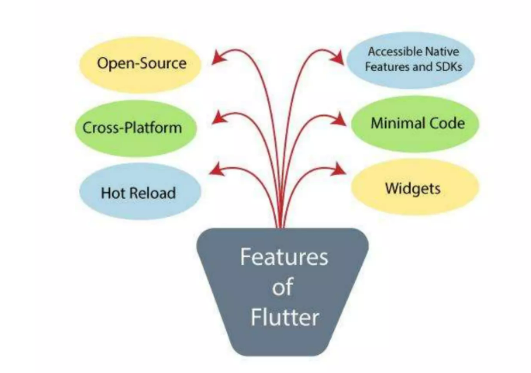
Hai nền tảng: Dart Native và Dart Web



Hình 1 Các nền tảng (platforms)

* + 1. Các tính năng của Flutter

Flutter cung cấp các phương pháp dễ dàng và đơn giản để xây dụng các ứng dụng đẹp mắt. Các tính năng chính để phát triển framwork di dộng:



Hình 2 Các tính năng của Flutter

Code nguồn mở (Open-Source): Flutter là một framwork code nguồn mở và miễn phí để phát triển các ứng dụng di dộng.

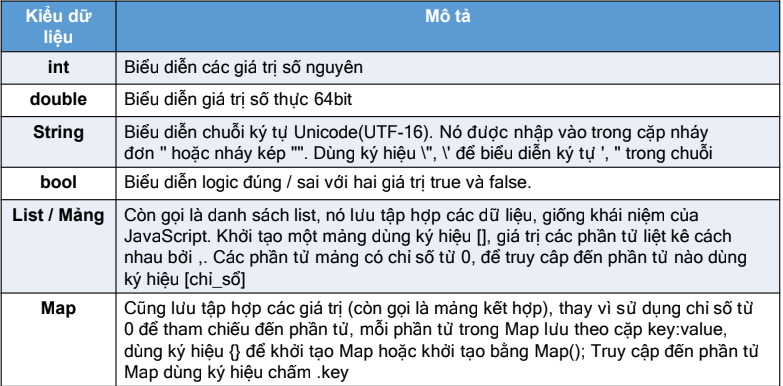
Đa nền tảng (Cross-platform): Tính năng này cho phép Flutter viết code một lần, duy trì và có thể chạy trên các nền tảng khác nhau như iOS, Android, web và desktop, tiết kiệm thời gian, công sức và chi phí cho các doanh nghiệp phát triển ứng dụng.

Tải lại nóng (Hot Reload): Bất cứ khi nào nhà phát triển thực hiện thay đổi trong code, thì khả năng tái tạo lại ứng dụng ngay lập tức trong chính ứng dụng mà không cần khởi động lại ứng dụng khi thay đổi mã nguồn, giúp tiết kiệm thời gian phát triển và thử nghiệm ứng dụng.

Các tính năng và SDK gốc có thể truy cập (Accessible Native Features and SDKs): Tính năng này cho phép quá trình phát triển ứng dụng dễ dàng và thú vị thông qua code gốc của Flutter, tích hợp bên thứ ba và các API nền tảng. Do đó, chúng tôi có thê dễ dàng truy cập SDK trên cả hai nền tảng.

Code tối thiểu (Minimal code): Ứng dụng Flutter được phát triển bởi ngôn ngữ lập trình Dart, sử dụng biên dịch JIT và AOT để cải thiện thời gian khởi động tổng thể, hoạt động và tăng tốc hiệu suất. JIT nâng cao hệ thống phát triển và làm mới giao diện người dùng mà không cần nỗ lực thêm vào việc xây dựng hệ thống mới.

Widget: framework công tác Flutter cung cấp các widget có khả năng phát triển các thiết kế giao diện tương tác với người dùng đẹp và linh hoạt, với nhiều lựa chọn và kiểu dáng phù hợp với nhiều thiết bị khác nhau có thể tùy chỉnh. Quan trọng nhất, Flutter có hai bộ công cụ tiện ích tập trung vào: Material Design (Android) và Cupertino Style (iOS) giúp mang lại trải nghiệm không có trục trặc trên tất cả các nền tảng.

* + 1. Những kiểu dữ liệu phổ biến trong Dart

Hình 3 Các dữ liệu cơ bản trong Dart

* Một số khai báo dữ liệu khác:
* Khai báo biến: sử dụng từ khóa var, không cần chỉ rõ kiểu dữ liệu, kiểu dữ liệu của biến phụ thuộc vào đối tượng gán cho biến.

|  |
| --- |
| var a = “Learn Dart” //a có kiểu String, nó chỉ lưu chuỗi |

* Khai báo biến: sử dụng từ khóa dynamic không cần chỉ rõ kiểu dữ liệu, kiểu dữ liệu của biến khôngcphụ thuộc vào đối tượng gán cho biến.

|  |  |
| --- | --- |
| Dynamic dyn = 123’ | // khởi tạo là số int |
| dyn = “Learn Dart” | // Gán chuỗi |
| dyn = 1.546 | // Gán double |

* + 1. Hướng đối tượng (OOP) trong Dart

Lập trình hướng đối tượng (OOP) trong Dart là một phương pháp lập trình dựa trên các đối tượng và lớp, giúp tổ chức mã nguồn theo cách dễ bảo trì và mở rộng.

OOP trong Dart giúp tổ chức mã nguồn theo cách rõ ràng, dễ bảo trì, tái sử dụng và mở rộng.

Dart hỗ trợ đầy đủ các tính năng của OOP, giúp lập trình viên xây dựng các ứng dụng phức tạp một cách dễ dàng và hiệu quả.

Các khái niệm chính trong OOP với Dart:

Lớp ( Class)

Class đơn giản là một khuôn mẫu dùng để chứa các biến và các hàm.

Trong biểu diễn một đối tượng, class dùng để mô tả đối tượng, bao gồm các biến ( các thuộc tính của đối tượng ) là các trường dữ liệu và các hàm (các phương thức của đối tượng).

Tính kế thừa ( Inheritance)

Kế thừa cho phép một lớp sử dụng các thuộc tính và phương thức của lớp khác

Trong Dart sử dụng từ khóa extends để kế thừa từ một lớp cha.

Tính trừ tượng (Abstraction)

Tính đóng gói (Encapsulation)

Giấu chi tiết triển khai của lớp, chỉ cung cấp các phương thức để tương tác.

Tính đa hình ( Polymorphism)

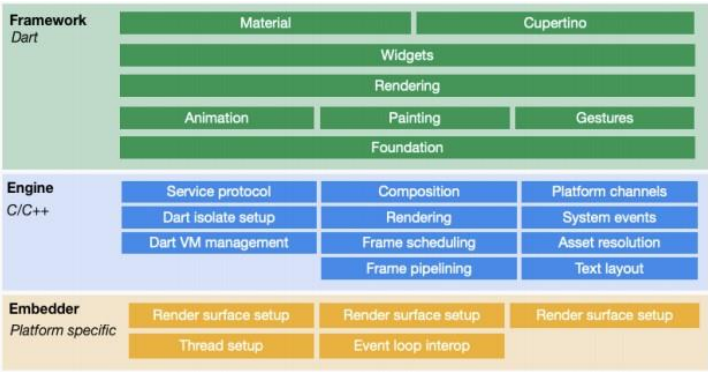
Cho phép các đối tượng thuộc các lớp khác nhau được xử lý thông qua cùng một giao diện.

* 1. Mô tả về Framework Flutter
     1. Giới thiệu về Flutter

Flutter là một bộ SDK đa nền tảng có thể hoạt động trên iOS và Android do Google phát triển được sử dụng để tạo ra các ứng dụng dành cho di động (native app).

Flutter gồm 2 thành phần quan trọng:

* Một SDK (Software Development Kit): Một tập hợp các công cụ sẽ giúp bạn phát triển các ứng dụng của mình. Nó bao gồm các công cụ để biên dịch mã nguồn của bạn thành các mã máy native (mã sử dụng cho IOS và Android).
* Một Framework (UI Library based on widgets): Một tập hợp các thành phần giao diện người dùng (UI) có thể tái sử dụng (button, text inputs, slider, v.v.) giúp bạn có thể cá nhân hóa tùy theo nhu cầu của riêng mình.
  + 1. Kiến trúc Flutter



Hình 4 Kiến trúc Flutter

Tầng Framwork (Dart): này được viết bằng ngôn ngữ lập trình Dart và cung cấp một loạt các thành phần giao diện người dùng, từ các widget cơ bản như nút và văn bản đến các cấu trúc phức tạp hơn.

Tầng Engine (C/C++):

* Lớp này được viết bằng C/C++ và chịu trách nhiệm cung cấp các dịch vụ cốt lõi cho ứng dụng Flutter.
* Công cụ này bao gồm các: thư viện đồ họa (thông qua Skia) để vẽ giao diện người dùng, bố cục văn bản, các API hộ trợ khả năng truy cập và thao tác các tệp và hệ thống mạng (network I/O), kiến trúc plugin, hộ trợ trợ năng.
* Thời gian chạy dart (Dart runtime) cung cấp môi trường để phát triển, thực thi các đoạn mã đã được thông dịch hoặc biên dịch cũng như cung cấp các runtime system bao gồm garbage collector, các thư viện cần có của ngôn ngữ.
* Thông dịch các đoạn mã Dart theo phương thức JIT hoặc AOT
* Đóng vai trò trung gian giữa mã nguồn được viết bởi Dart và thiết bị phần cứng (hoặc phần mềm nằm ngoài ứng dụng).

Tầng Embedder: Embedder này đóng vai trò quan trọng trong việc cung cấp dịch vụ tích hợp ứng dụng Flutter với các nền tảng khác nhau như iOS, Android, Windows, macOS và Linux. Embedder trong Flutter đảm nhận vai trò then chốt này, giúp ứng dụng của bạn hoạt động mượt mà và tương thích trên nhiều hệ điều hành.

* 1. Mô tả cơ sở dữ liệu trong Flutter

Cơ sở dữ liệu (Database) là một tập hợp dữ liệu có tổ chức, hỗ trợ việc lưu trữ và thao tác dữ liệu và được truy cập điện tử từ hệ thống máy tính. Chúng ta có thể tô chức dữ liệu thành các hàng, cột, bảng và chỉ mục. Nó giúp cho việc quản lý đừ liệu trở nên dê dàng. Chúng tôi có thê lưu trữ nhiêu thứ trong cơ sở dữ liệu, như tên, tuổi, hình ảnh, hình ảnh, tệp, pdf, v.v.

* + 1. Server và API sử dụng Node.js.

Backend của dự án được xây dựng bằng Node.js, một nền tảng lập trình dựa trên JavaScript cho việc xây dựng các ứng dụng mạng có khả năng mở rộng. Node.js được chọn vì tính linh hoạt và khả năng xử lý yêu cầu đồng thời cao, đáp ứng tốt yêu cầu của ứng dụng di động.

Hình 5 Server và API sử dụng node.js

* + 1. Cơ sở dữ liệu MongoDB

MongoDB được sử dụng làm hệ quản trị cơ sở dữ liệu cho dự án. MongoDB là một cơ sở dữ liệu NoSQL phổ biến, cung cấp khả năng lưu trữ và truy xuất dữ liệu linh hoạt, đặc biệt phù hợp cho các ứng dụng có cấu trúc dữ liệu phức tạp hoặc yêu cầu mở rộng dễ dàng.

Hình 6 MongoDB

* 1. Lựa chọn công cụ cho đề tài

Flutter, dart, node.js, MongoDB.

Ngôn ngữ lập trình Androi và công cụ lập trình Visual Studio Code, Android Studio.

CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

3.1. Yêu cầu chức năng

3.1.1 Chức năng đăng nhập/đăng kí

Chức năng cho phép khách hàng có thêm xem chi tiết sản, thêm sản phẩm vào giỏ hàng, đặt hàng, thanh toán và đánh giá sản phẩm phẩm đồng thời xem sản phẩm gợi ý.

* + 1. Chức năng tìm kiếm

Khi khách hàng cần tìm một sản phẩm nào đó thì có thể nhập thông tin sản phẩm vào ô tìm kiếm để thực hiện chức năng tìm kiếm. Nếu sản phẩm đó có tồn tại trong dữ liệu thì sẽ hiển thị thông tin sản phẩm được tìm thấy

Chức năng bao gồm: tìm kiếm theo tên, tìm kiếm theo tên nhà sản xuất và theo giá sản phẩm và lịch sử tìm kiếm.

* + 1. Chức năng quản lý giỏ hàng

Khách hàng thêm sản phẩm phù hợp vào giỏ hàng sau khi thấy sản phẩm đáp ứng nhu cầu của mình thì có thể chọn thêm vào giỏ hàng.

Khách hàng cũng có thể xóa sản phẩm đã thêm trong giỏ hàng và thêm vào một sản phẩm khác. Ngoài ra, khi cần khách hàng có thể cập nhật số lượng sản phẩm cần mua trong giỏ hàng.

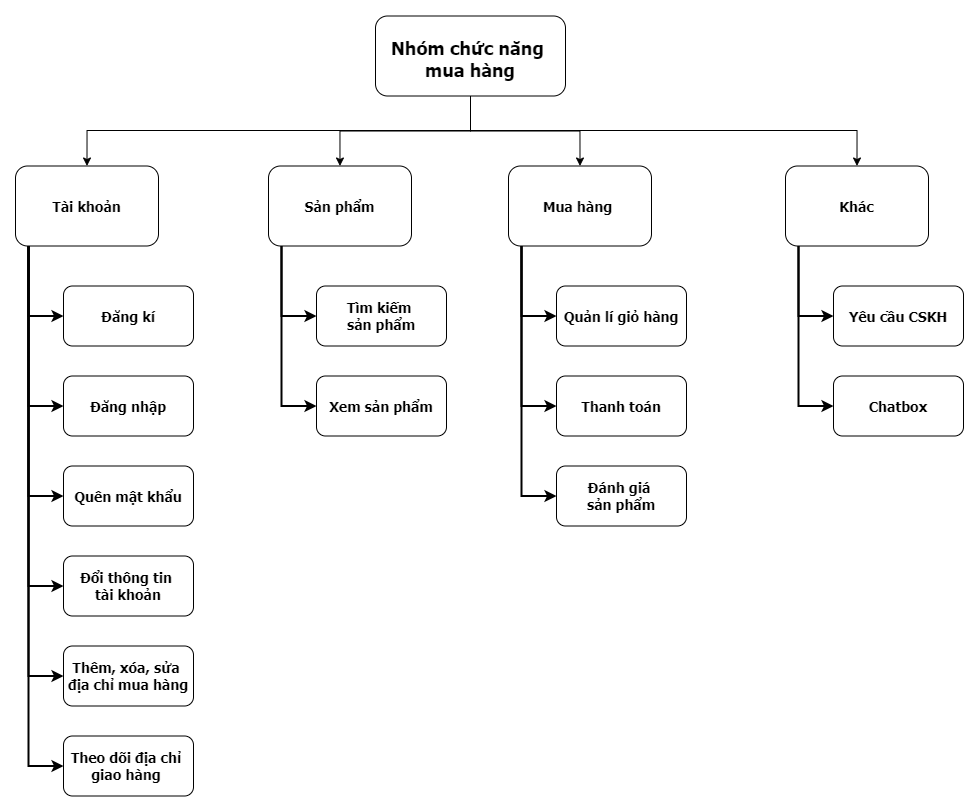
* + 1. Chức năng thanh toán

Khách hàng sau khi xem xem sản phẩm có thể chọn chức năng thanh toán để thực hiện xác nhận mua hàng. Muốn thực hiện thanh toán thì yêu cầu người dùng phải đăng nhập vào hệ thống.

* + 1. Chức năng đánh giá sản phẩm

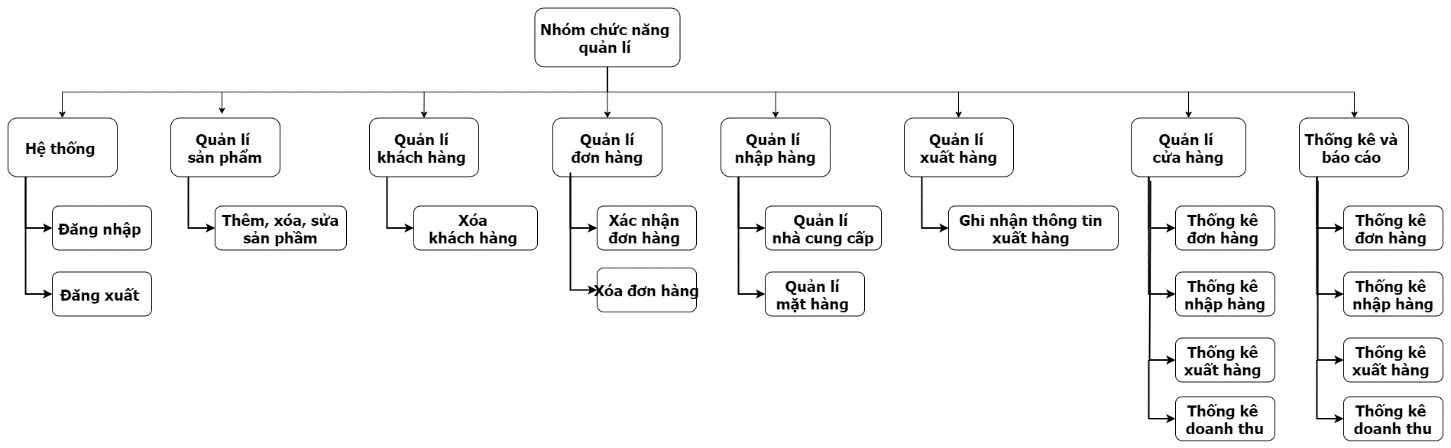
Khách hàng sau khi xem chi tiết sản phẩm, khách hàng có thể để lại những đánh giá về sản phẩm. Muốn thực hiện đánh giá sản phẩm yêu cầu khách hàng cần đăng nhập vào hệ thống.

* Nhóm chức năng mua hàng



Hình 7 Sơ đồ phân cấp nhóm chức năng mua hàng

* Nhóm chức năng quản lí



Hình 8 Sơ đồ phân cấp nhóm chức năng quản lí

3.2. Yêu cầu phi chức năng

* **Hiệu suất**

Hiệu suất là yếu tố then chốt; thời gian tải trang nên dưới 3 giây để giữ chân người dùng, và hệ thống cần có khả năng xử lý tối thiểu 1000 người dùng đồng thời mà không bị gián đoạn, giúp đảm bảo trải nghiệm mượt mà ngay cả trong giờ cao điểm.

* **Bảo mật**

Về bảo mật, việc bảo vệ thông tin cá nhân của người dùng là cực kỳ quan trọng. Dữ liệu như thông tin thẻ tín dụng và địa chỉ giao hàng phải được mã hóa để ngăn chặn truy cập trái phép. Ngoài ra, xác thực hai yếu tố cho tài khoản người dùng là một biện pháp bổ sung cần thiết để tăng cường an toàn.

* **Khả năng mở rộng**

Khả năng mở rộng cũng là một yêu cầu không thể thiếu, khi hệ thống cần có khả năng xử lý lượng giao dịch dự kiến tăng lên 50% trong năm tới. Điều này đảm bảo rằng ứng dụng có thể phát triển cùng với nhu cầu của thị trường.

* **Khả năng sử dụng**

Cùng với đó, tính khả dụng yêu cầu ứng dụng đạt được thời gian hoạt động (uptime) ít nhất 99.9%, và hỗ trợ đa nền tảng (web, iOS, Android) cũng như nhiều trình duyệt phổ biến để phục vụ đa dạng người dùng..

* **Bảo trì**

Hơn nữa, tính bảo trì là một yếu tố quan trọng, hệ thống cần dễ dàng được cập nhật mà không gây gián đoạn cho người dùng, đảm bảo luôn hoạt động ổn định.

* **Tương thích**

Khả năng tương thích với các phiên bản mới nhất của hệ điều hành di động và trình duyệt web phổ biến cũng cần được chú trọng.

* **Độ tin cậy**

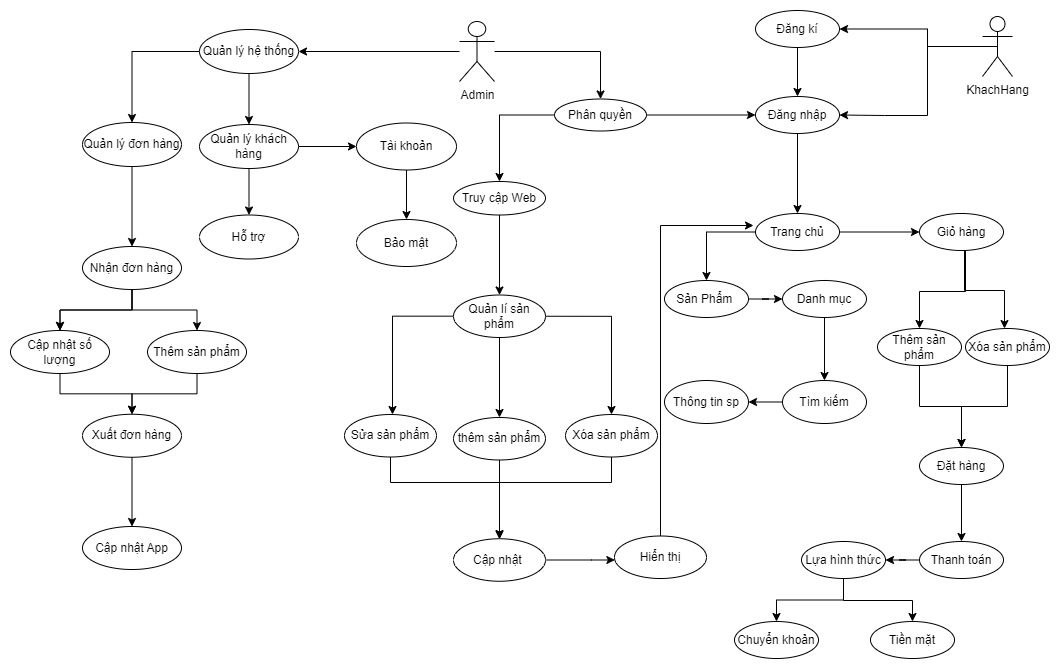
Việc tự động sao lưu dữ liệu giao dịch hàng ngày là một yêu cầu cần thiết để đảm bảo không mất mát thông tin trong trường hợp có sự cố.

* **Khả năng truy cập**

Ứng dụng cần đáp ứng đầy đủ các quy định về bảo vệ dữ liệu như GDPR và luật thương mại điện tử, cũng như đảm bảo tính khả năng truy cập cho người dùng khuyết tật, chẳng hạn như hỗ trợ cho màn hình đọc. Tất cả những yếu tố này kết hợp lại sẽ tạo nên một ứng dụng bán hàng vừa hiệu quả, vừa an toàn, và thân thiện với người dùng.

3.3. Mô tả tổng quan hệ thống

3.3.1. Biểu đồ Use – Case tổng quát



Hình 9 Biểu đồ Use – Case tổng quát

3.3.2. Danh sách các tác nhân ( Actor)

|  |  |  |
| --- | --- | --- |
| **Tác nhân** | **Mô tả** | **Ghi chú** |
| Khách hàng | Người dùng ứng dụng, sử dụng các chức năng của App |  |
| Người quản lí | Là người nắm tình hình mua bán, doanh thu của cửa hàng và chịu trách nhiệm tới phiếu nhập hàng khi hàng hóa được nhập về và tạo phiếu xuất khi kho hàn hàng. | Nhân viên hệ thống |

Bảng 2 Bảng tác nhân

3.3.3. Danh sách chức năng và mô tả

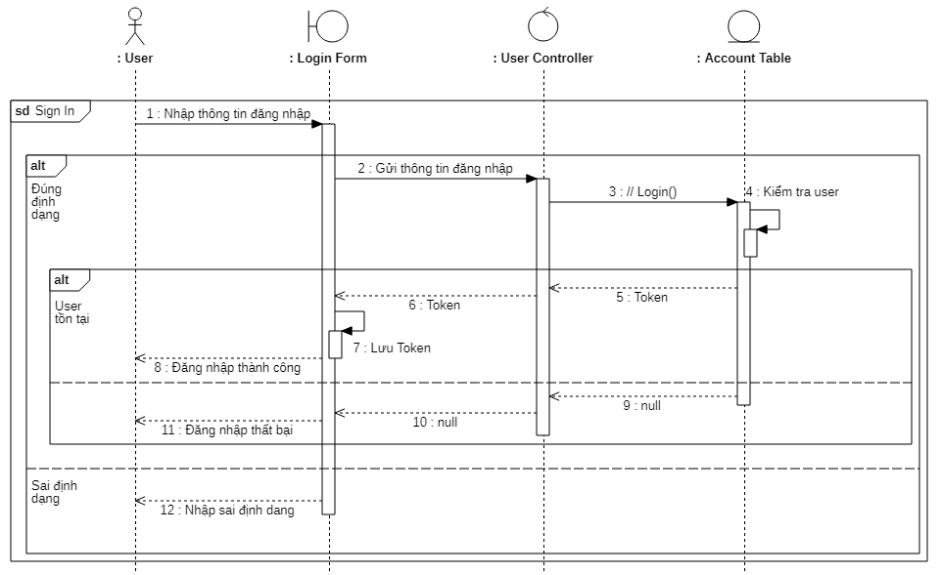
|  |  |  |
| --- | --- | --- |
| **Tên Chức năng** | **Mô tả** | Chức năng |
| TT\_Đăng nhập | Mỗi người sử dụng có 1 use name và password để đăng nhập vào hệ thống, người quản trị sẽ cấp phát, thay đổi, xóa quyền và tên người dùng, nếu người dùng có tên và mật khẩu đúng thì thao tác đăng nhập sẽ thành công. | Đăng nhập |
| TT\_Đăng kí | Đăng kí tài khoản để truy cập vào ứng dụng | Đăng kí |
| TT\_Quản lí tài khoản | Khách hàng có thể quản lí thông tin cá nhân, theo dõi đơn hàng và hộ trợ khi cần thiết | Nhóm chức năng mua hàng ( tài khoản, khác) |
| TT\_Mua hàng | Giúp khách hàng có nhữn trải nghiệm khi mua hàng | Nhóm chức năng mua hàng (sản phẩm, mua hàng) |
| TT\_Hệ thống | Giúp người hệ quản trị hệ thống có thể kiểm soát mọi thứ như doanh thu, mặt hàng từ các cửa hàng | Nhóm chức năng hệ thống |

Bảng 3 Bảng danh sách chức năng

3.4. Đặc tả các yêu cầu chức năng

3.4.1. Chức năng đăng kí, đăng nhập

3.4.1.1. Biểu đồ tuần tự Đăng nhập

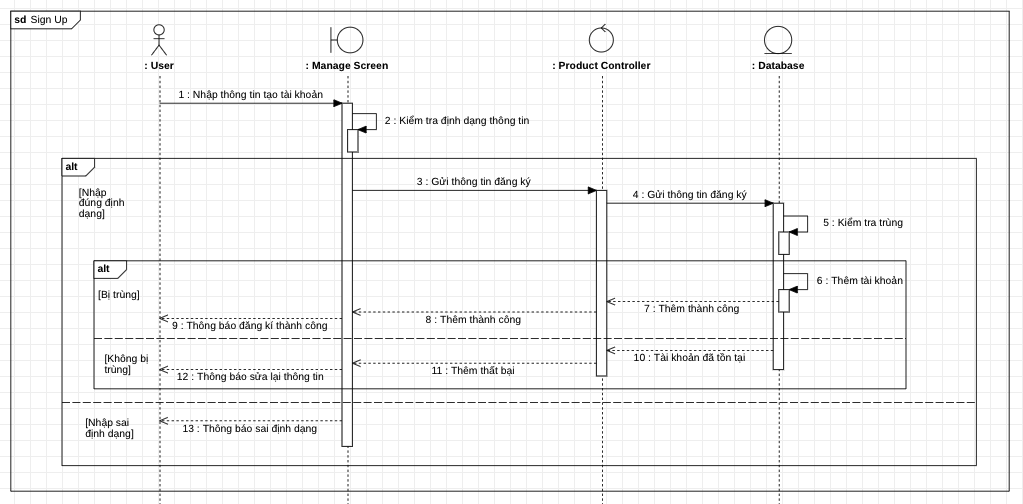


Hình 10 Biểu đồ tuần tự chức năng đăng nhập

|  |  |
| --- | --- |
| **Tuần tự** | **Nội dung** |
| Tên | Đăng nhập |
| Mô tả | Chức năng cho phép thành viên hoặc khách hàng đăng nhập vào hệ thống để thực hiện những chức năng đang có trên hệ thống. |
| Actor | Khách hàng, Nhân viên hệ thống |
| Sự kiện kích hoạt | Khi thành viên chọn chức năng đăng nhập trên trang công khai hoặc trang quản trị. |
| Tên điều kiện | Thành viên phải có tài khoản trên hệ thống. |
| Hậu điều kiện | Thành viên đăng nhập thành công. |
| Luồng sự kiện chính | 1. Hệ thống hiển thị màn hình đăng nhập.  2. Thành viên nhập tên đăng nhập và mật khẩu.  3. Hệ thống kiểm tra thông tin đăng nhập.  4. Nếu thành công hệ thống hiển thị màn hình đăng nhập thành  công.  5. Kết thúc |
| Luồng điều kiện phụ | A1 – Mật khẩu không hợp lệ: Khi thành viên nhập sai tên đăng nhập  và mật khẩu.  1. Hệ thống hiển thị lại màn hình đăng nhập để thành viên nhập  lại thông tin kèm theo thông báo tên đăng nhập và mật khẩu  bị sai.  2. Quay lại bước 2 trong luồng sự kiện chính.  A2 – Quên mật khẩu: Khi thành viên chọn chức năng quên mật khẩu  trên màn hình đăng nhập.  1. Hệ thống hiện thị màn hình để thành viên nhập email.  2. Thành viên nhập email và chọn nút chức năng “Lấy lại mật  khẩu”.  3. Hệ thống kiểm tra email hợp lệ và gửi liên kết để reset mật  khẩu cho khách hàng qua email.  4. Hệ thống hiển thị màn hình thông báo thành công.  5. Kết thúc. |

Bảng 4 Bảng tác nhân Đăng kí

3.4.1.2. Biểu đồ tuần tự Đăng kí

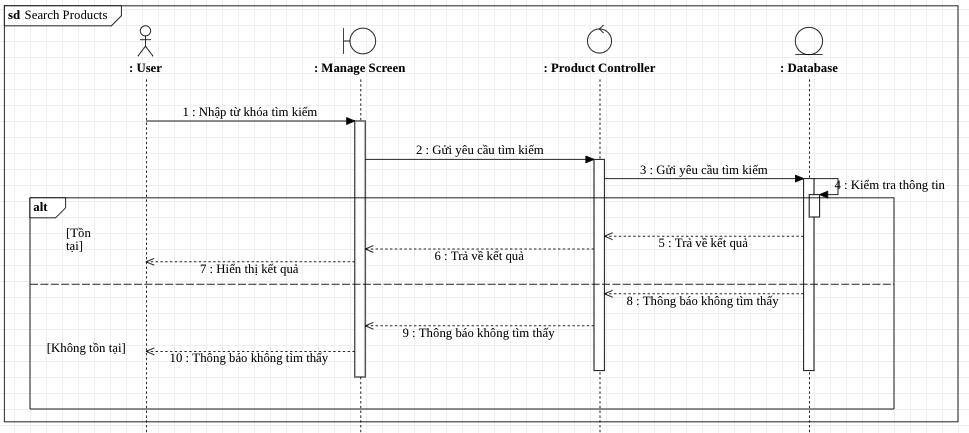


Hình 11 Biểu đồ tuần tự chức năng đăng kí

|  |  |
| --- | --- |
| **Tuần tự** | **Nội dung** |
| Tên | Đăng kí tài khoản |
| Mô tả | Chức năng cho phép khách hàng đăng ký làm thành viên trên hệ thống. |
| Actor | Khách hàng |
| Sự kiện kích hoạt | Khách hàng chọn nút đăng ký trong phần đăng nhập. |
| Tên điều kiện | Không có |
| Hậu điều kiện | Khách hàng trở thành thành viên của hệ thống. |
| Luồng sự kiện chính | 1. Khách hàng chọn đăng ký trong phần đăng nhập.  2. Nhập thông tin vào biểu mẫu hiển thị.  3. Nhấn nút đăng ký.  4. Nếu thành công thì hệ thống sẽ chuyển đến phần đăng nhập  và thông báo đăng ký thành công.  5. Hệ thống cập nhập thông tin khách hàng vào cơ sở dữ liệu. |
| Luồng sự kiện phụ | A1 – Thông tin nhập không chính xác: Khách hàng nhập sai thông tin trên biểu mẫu.  1. Nếu thông tin trùng với thông tin tài khoản đã tồn tại trong cơ sở dữ liệu, hệ thống thông báo tài khoản đã tồn tại và yêu cầu tạo lại thông tin khác.  2. Hệ thống yêu cầu khách hàng nhập lại thông tin.  3. Quay lại bước hai của luồng sự kiện chính. |

Bảng 5 Tác nhân đăng nhập

3.4.2. Biểu đồ tuần tự tìm kiếm



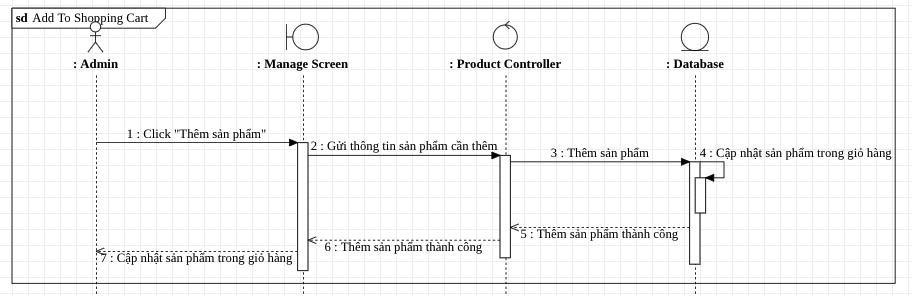
Hình 12 Biểu đồ tuần tự chức năng tìm kiếm

Mô tả tóm tắt:

* Khách hàng đăng nhập hệ thống trang chủ.
* Khách hàng chọn tìm kiếm trên giao diện trang chủ.
* Khách hàng nhập thông tin tìm kiếm.
* Hệ thống sẽ kiểm tra thông tin mà khách hàng tìm kiếm trong cơ sở dữ liệu.
  + Hệ thống trả về kết quả tìm kiếm và hiển thị lên giao diện.
  + Nếu không có kết quả nào trùng khớp, hệ thống gửi về thông báo không tìm thấy sản phẩm.

Nếu nhập không đủ thông tin, hệ thống hiển thị thống báo lỗi.

3.4.3. Biểu đồ tuần tự thêm sản phẩm vào giỏ hàng

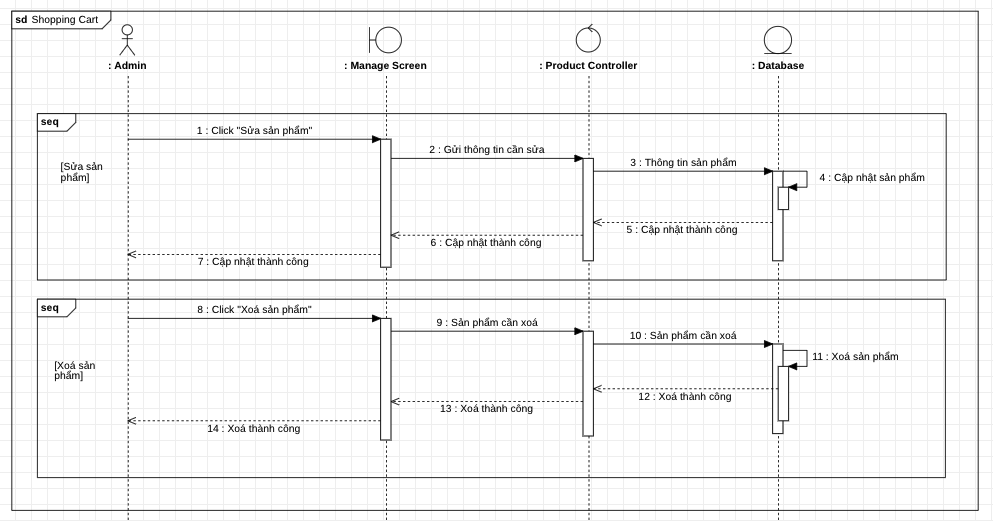


Hình 13 Biểu đồ tuần tự thêm sản phẩm vào giỏ hàng

Mô tả tóm tắt:

* Tại màn hình danh sách sản phẩm.
* Chọn sản phẩm cần mua.
* Hiển thị màn hình chi tiết sản phẩm.
* Chọn nút thêm vào giỏ hàng
* Hệ thống lưu vào danh sách sản phẩm trong giỏ hàng.

3.4.4. Biểu đồ tuần tự quản lý giỏ hàng

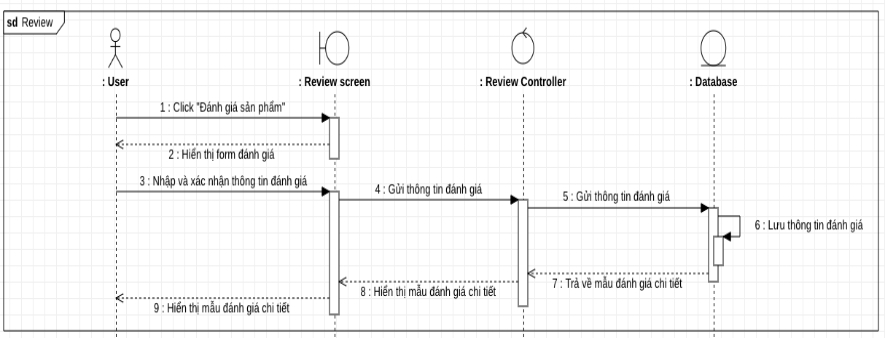


Hình 14 Biểu đồ tuần tự chức năng quản lý giỏ hàng

Mô tả tóm tắt:

* Chức năng xóa sản phẩm:
  + Chọn sản phẩm cần xóa.
  + Chọn nút xóa.
  + Hệ thống xác nhận và cập nhật lại danh sách sản phẩm trong giỏ hàng.
* Chức năng thay đổi số lượng sản phẩm:
  + Chọn sản phẩm.
  + Chọn nút “+” hoặc “-” để thay đổi số lượng sản phẩm như mong muốn.
  + Hệ thống xác nhận và cập nhật lại danh sách sản phẩm trong giỏ hàng.

3.4.5. Biểu đồ tuần tự chức năng đánh giá

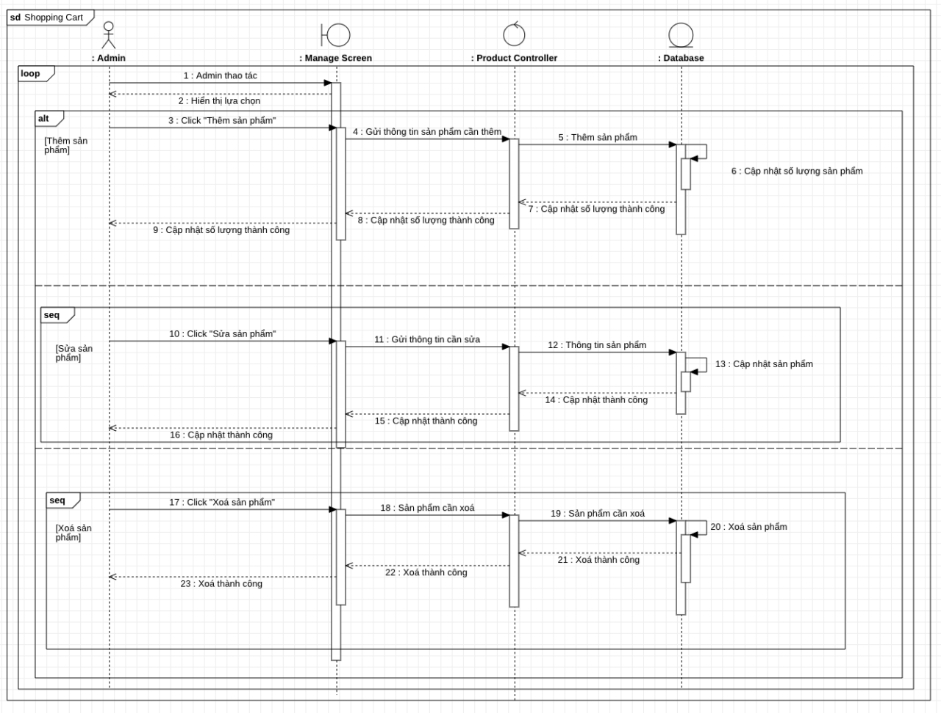


Hình 15 Biểu đồ tuần tự chức năng đánh giá

Mô tả tóm tắt:

* Khách hàng đăng nhập vào hệ thống.
* Khách hàng chọn sản phẩm và chọn đánh giá sản phẩm.
* Hiển thị form đánh giá sản phẩm.
* Gửi thông tin đánh giá sản phẩm.
* Hệ thông lưu thông tin đánh giá sản phẩm vào cơ sở dữ liệu.

3.4.6. Biểu đồ tuần tự chức năng quản lý sản phẩm

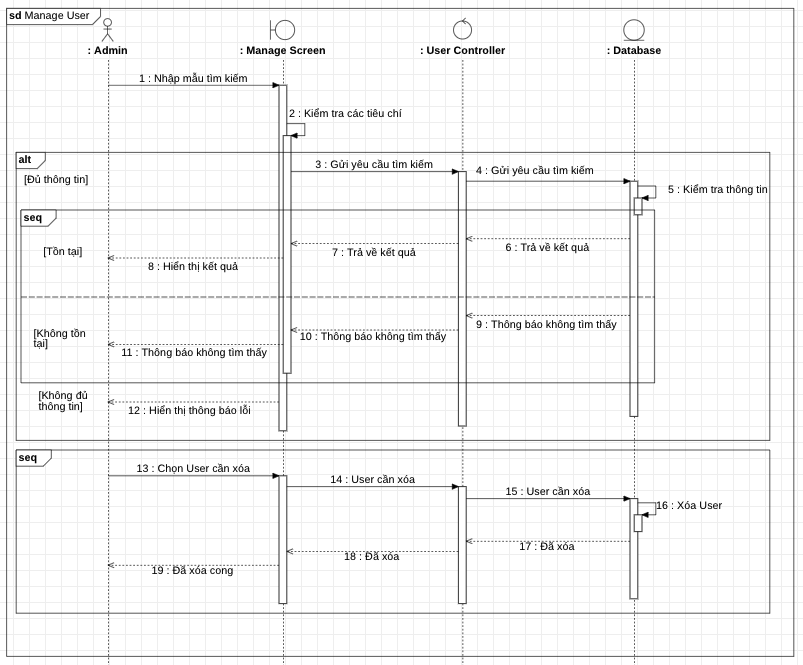


Hình 16 Biểu đồ tuần tự chức năng quản lý sản phẩm

Mô tả tóm tắt:

* Chức năng thêm sản phẩm:
  + Tại màn hình danh sách sản phẩm.
  + Ấn nút thêm sản phẩm.
  + Hiển thị màn hình nhập thông tin sản phẩm mới.
  + Chọn nút Lưu
  + Hệ thống lưu vào danh sách sản phẩm.
* Chức năng xóa sản phẩm:
  + Chọn sản phẩm cần xóa.
  + Chọn nút xóa.
  + Hệ thống xác nhận và cập nhật lại danh sách sản phẩm.
* Chức năng sửa sản phẩm sản phẩm:
  + Chọn sản phẩm cần xóa.
  + Chọn sửa thông tin sản phẩm.
  + Nhập thông tin cần chỉnh sửa.
  + Hệ thống xác nhận và cập nhật lại danh sách sản phẩm.

3.4.7. Biểu đồ tuần tự chức năng quản lý khách hàng



Hình 17 Biểu đồ tuần tự chức năng quản lý khách hàng

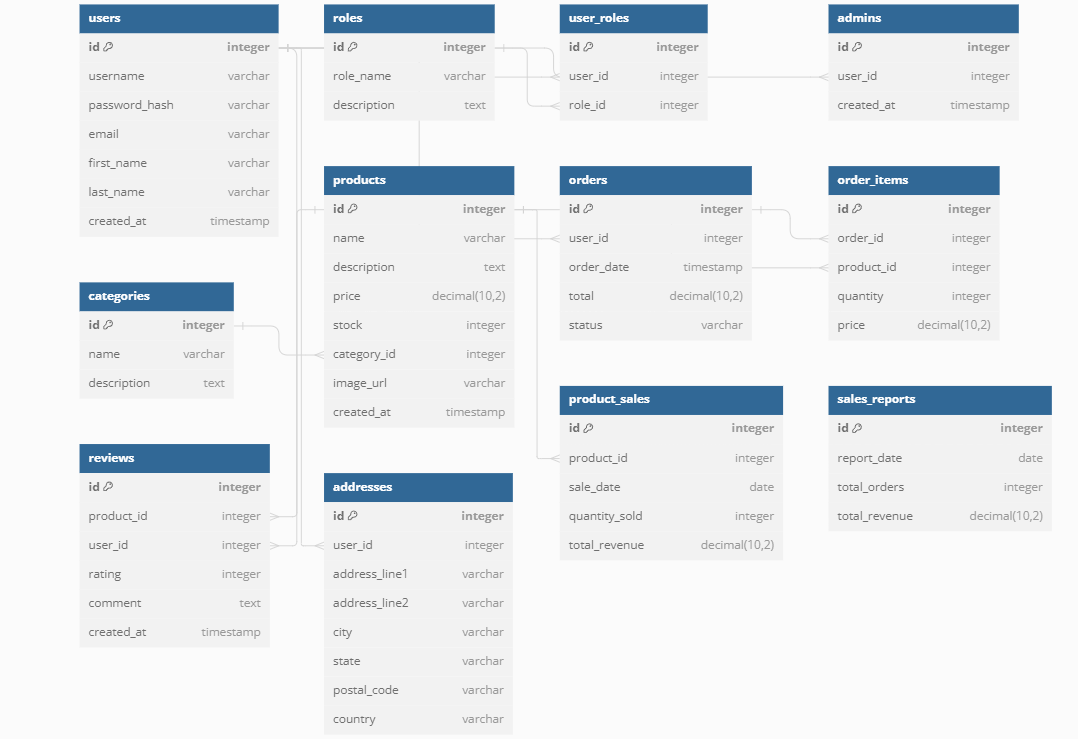
Mô tả tóm tắt:

* Admin đăng nhập vào trang quản trị hệ thống.
* Chức năng tìm kiếm sản phẩm:
  + Tại màn hình danh sách sản phẩm.
  + Chọn tìm kiếm.
  + Nhập từ khóa cần tìm.
  + Hệ thống kiểm tra thông tin trong CSDL và hiển thị lên màn hình danh sách tài khoản khách hàng.
* Chức năng xóa tài khoản:
  + Chọn tài khoản cần xóa.
  + Chọn nút xóa.

Hệ thống xác nhận và cập nhật lại danh sách tài khoản.

3.5. Thiết kế Cơ sở dự liệu

3.5.1 Mô hình thực thể chi tiết



Hình 18 Cơ sở dữ liệu

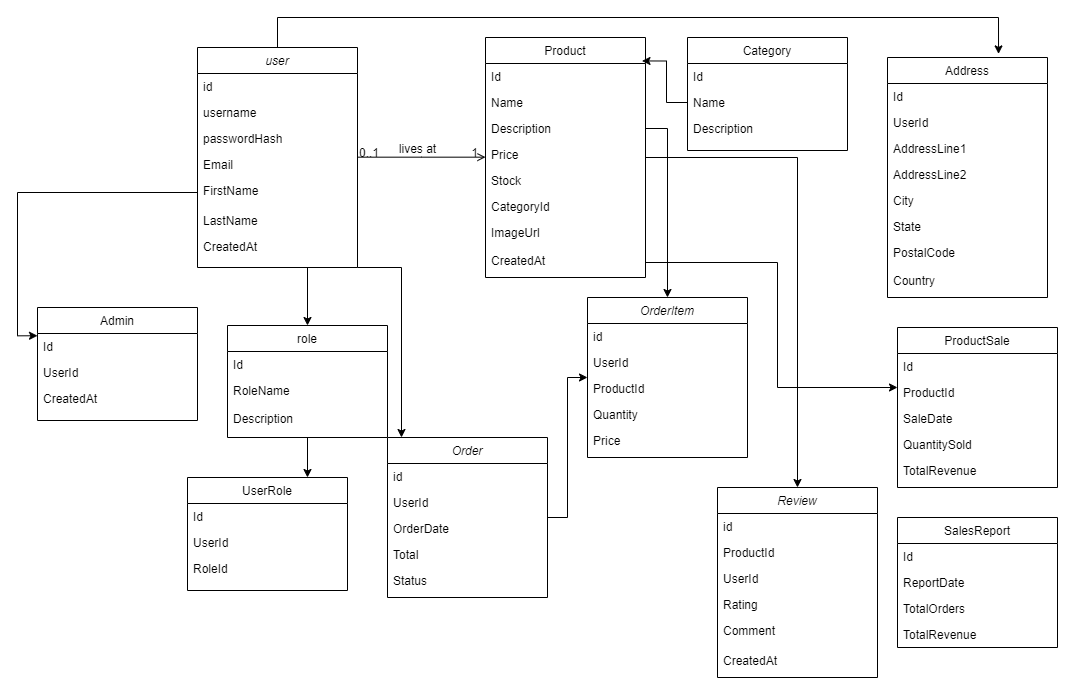
|  |  |  |
| --- | --- | --- |
| **STT** | **Tên Class** | **Mô tả** |
| 1 | users | Lưu thông tin người dùng, bao gồm tên đăng nhập, mật khẩu (đã mã hóa), email, tên, họ và ngày tạo tài khoản. |
| 2 | roles | Quản lý các vai trò trong hệ thống. |
| 3 | user\_roles | Liên kết người dùng với các vai trò của họ. |
| 4 | admins | Lưu thông tin quản trị viên, liên kết với bảng người dùng. |
| 5 | categories | Lưu thông tin về các danh mục sản phẩm. |
| 6 | products | Lưu thông tin sản phẩm, bao gồm tên, mô tả, giá, số lượng tồn kho, danh mục, và URL hình ảnh. |
| 7 | orders | Lưu thông tin đơn hàng, bao gồm người dùng đặt hàng, ngày đặt hàng, tổng giá và trạng thái. |
| 8 | order\_items | Chi tiết mặt hàng trong mỗi đơn hàng, bao gồm mã đơn hàng, mã sản phẩm, số lượng và giá. |
| 9 | reviews | Đánh giá sản phẩm từ người dùng, bao gồm đánh giá, bình luận và ngày tạo. |
| 10 | addresses | Địa chỉ giao hàng của người dùng |
| 11 | product\_sales | Thống kê doanh số bán hàng theo sản phẩm. |
| 12 | sales\_reports | Báo cáo tổng hợp doanh số và đơn hàng theo ngày. |

Bảng 6 Bảng cơ sở dữ liệu

Bổ sung các thuộc tính cho các class:

* Users (**Id**, Username, PasswordHash, Email, FirstName, LastName, CreatedAt)
* Roles (**Id**, RoleName, Description)
* UserRoles (**Id**, UserId, RoleId)
* Admins (**Id**, UserId, CreatedAt)
* Categories (**Id**, Name, Description)
* Products (**Id**, Name, Description, Price, Stock, CategoryId, ImageUrl, CreatedAt)
* Orders (**Id**, *UserId*, OrderDate, Total, Status)
* OrderItems (**Id**, *OrderId*, *ProductId*, Quantity, Price)
* Reviews (**Id**, *ProductId*, *UserId*, Rating, Comment, CreatedAt)
* Addresses (**Id**, *UserId*, AddressLine1, AddressLine2, City, State, PostalCode, Country)
* ProductSales (**Id**, *ProductId*, SaleDate, QuantitySold, TotalRevenue)
* SalesReports (**Id**, ReportDate, TotalOrders, TotalRevenue)

**Class diagram:**



Hình 19 Class diagram

CHƯƠNG 4. CÀI ĐẶT CHƯƠNG TRÌNH VÀ TRIỂN KHAI

* 1. Mô tả về hệ thống

4.1.1. Xây dựng hệ thống

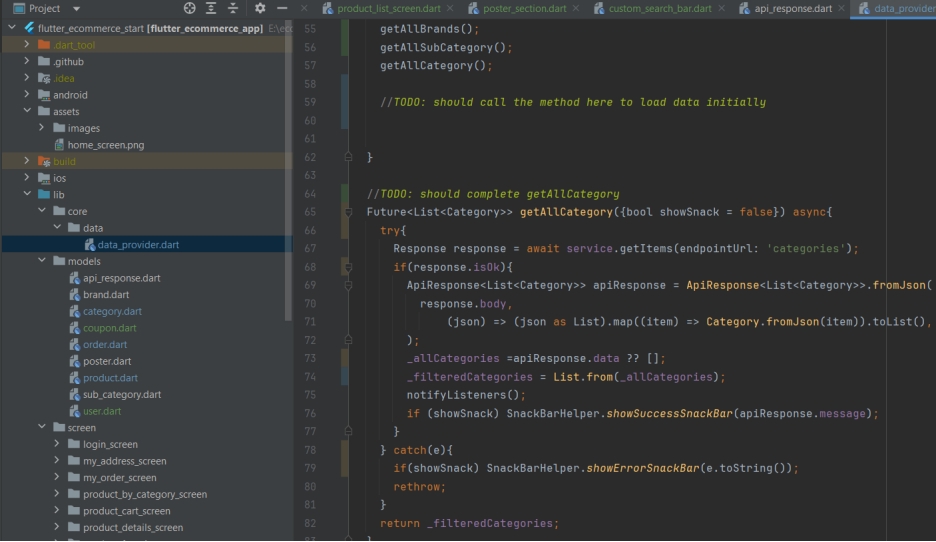
Trong sản phẩm em đã sử dụng **Layered Architecture** để phân chia thành các lớp chính sau:

1. Data Layer (Lớp dữ liệu)
2. Presentation Layer (Lớp trình bày)
3. Service Layer (Lớp dịch vụ)
4. Model Layer (Lớp mô hình)
5. Utility Layer (Lớp tiện ích)
6. Widget Layer (Lớp giao diện thành phần)

1. Data Layer

Thư mục: core/data/dataprovider

Chức năng: Chứa các thành phần liên quan đến việc truy cập và quản lý dữ liệu, bao gồm việc giao tiếp với cơ sở dữ liệu hoặc các nguồn dữ liệu bên ngoài.

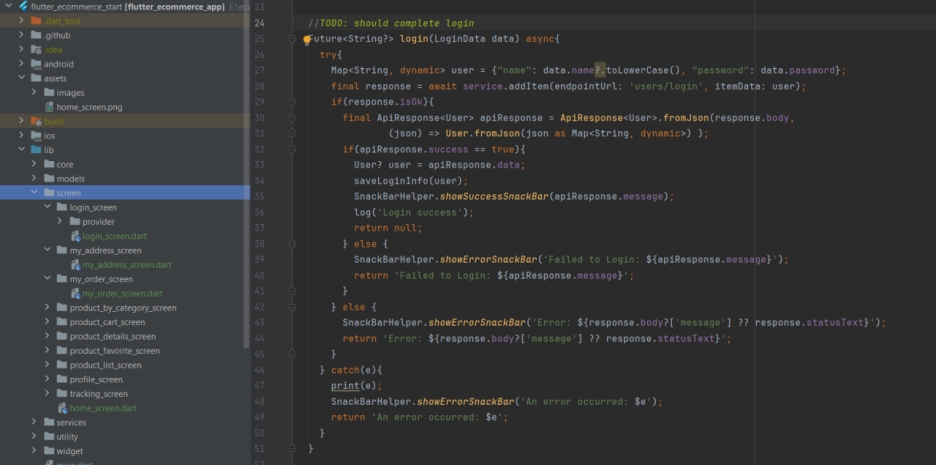


Hình 20 Lớp data\_provider

2. Presentation Layer

Thư mục: screen/…

Chức năng: Định nghĩa giao diện người dùng và quản lý tương tác với người dùng. Thư mục này có thể được chia nhỏ thành các thành phần như provider.

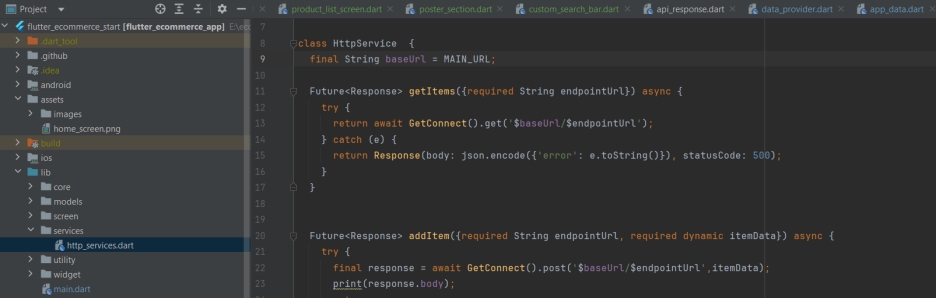


Hình 21 Lớp presentation

3. Service Layer

Thư mục: services/http\_services

Chức năng: Chứa các dịch vụ kết nối với các hệ thống hoặc dịch vụ bên ngoài, thực hiện các tác vụ như gọi API, xử lý logic nghiệp vụ.

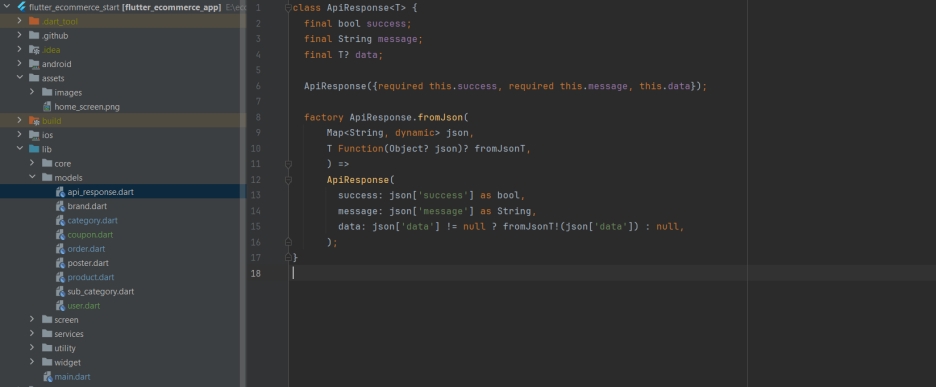


Hình 22 Lớp Service

4. Model Layer

Thư mục: models/…

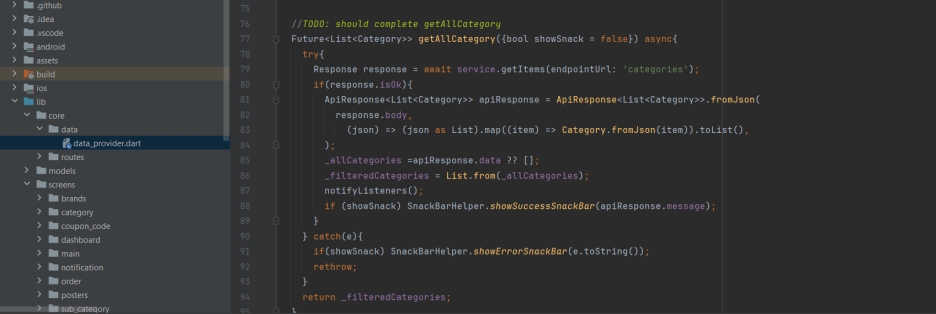
Chức năng: Chứa các định nghĩa về cấu trúc dữ liệu và các mô hình trong ứng dụng. Các mô hình này thường đại diện cho các thực thể trong cơ sở dữ liệu.

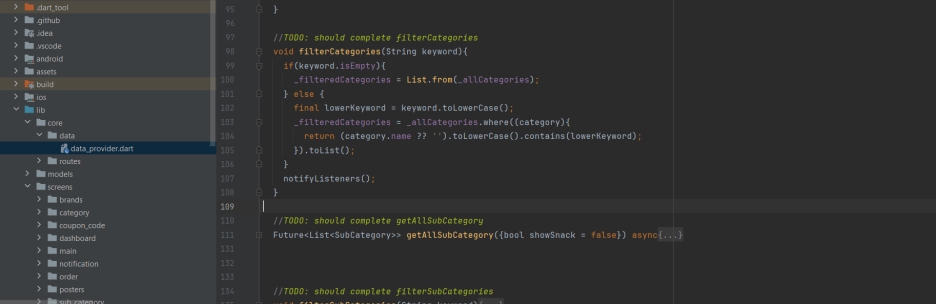


Hình 23 Lớp API

4.1.2. Một số mã nguồn quan trọng của hệ thống

1. **Lớp DataProvider** đóng vai trò là trung tâm quản lý dữ liệu của ứng dụng, cung cấp dữ liệu từ backend, lưu trữ và quản lý dữ liệu cục bộ, và cung cấp các phương thức để lọc và tính toán dữ liệu theo yêu cầu. Sử dụng ChangeNotifier, lớp này có thể thông báo cho các widget khác khi dữ liệu thay đổi, giúp cập nhật giao diện người dùng một cách linh hoạt và hiệu quả.

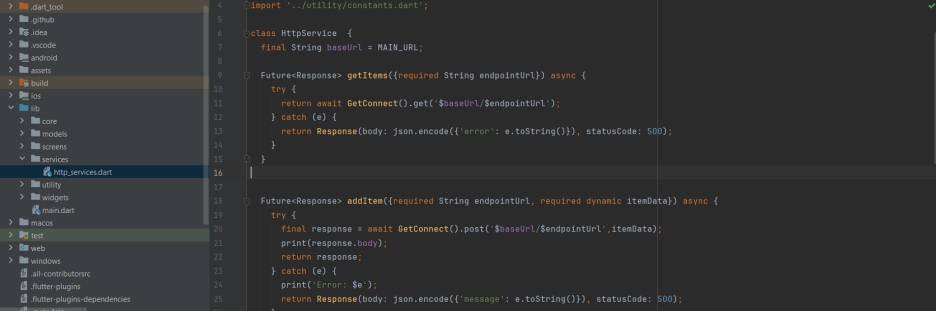




*Hình 25 Lớp Dataprovider*

2. **Lớp services** cung cấp các phương thức để tương tác với backend API thông qua giao thức RESTful. Cụ thể:

1. Khởi tạo và thiết lập baseUrl: Xác định URL cơ sở của backend API.
2. Phương thức getItems: Thực hiện yêu cầu HTTP GET để lấy dữ liệu từ một tài nguyên cụ thể trên API.
3. Phương thức addItem: Thực hiện yêu cầu HTTP POST để thêm một mục mới vào backend.
4. Phương thức updateItem: Thực hiện yêu cầu HTTP PUT để cập nhật một mục trong backend.
5. Phương thức deleteItem: Thực hiện yêu cầu HTTP DELETE để xóa một mục từ backend.

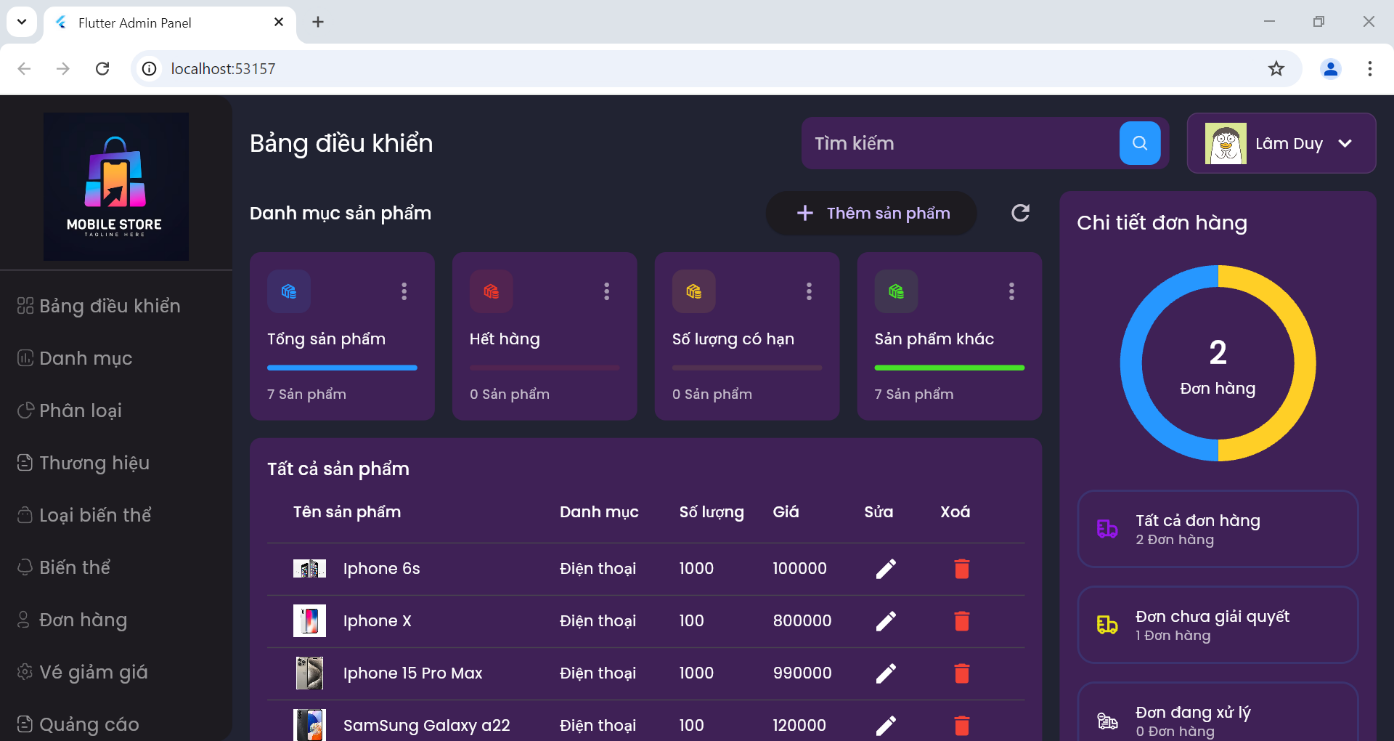


*Hình 26 Lớp Services*

* 1. Mô tả ứng dụng

Các chức năng mà sản phẩm đạt được:

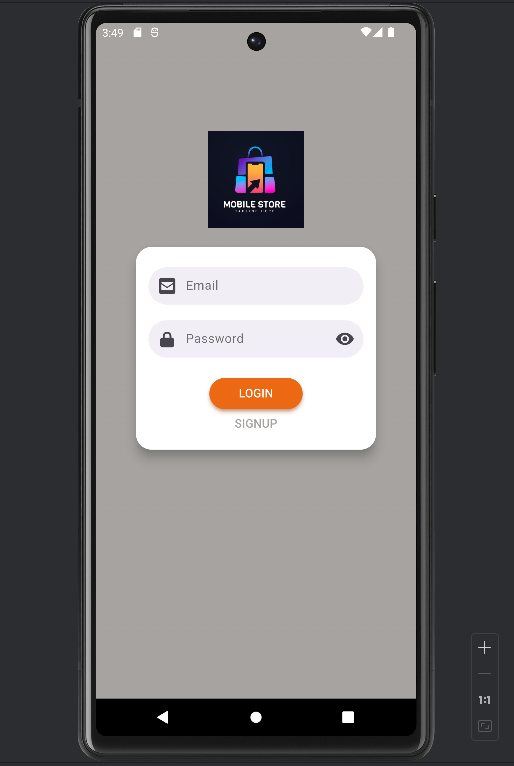
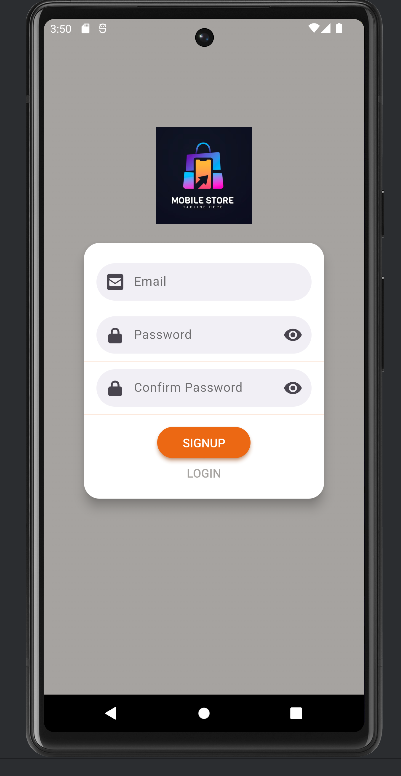
* 1. Website quản lý
     + 1. **Thêm sản phẩm**: Cho phép quản trị viên thêm sản phẩm mới vào hệ thống.
       2. **Sửa sản phẩm**: Cho phép quản trị viên chỉnh sửa thông tin của sản phẩm hiện có.
       3. **Xóa sản phẩm**: Cho phép quản trị viên xóa sản phẩm khỏi hệ thống.
       4. **Quản lý người dùng**: Cho phép quản trị viên thêm, sửa và xóa thông tin người dùng.
  2. **Ứng dụng di động**
     1. **Đăng nhập và đăng ký**: Cho phép người dùng tạo tài khoản mới và đăng nhập vào hệ thống.
     2. **Tìm kiếm sản phẩm**: Cho phép người dùng tìm kiếm sản phẩm trong hệ thống.
     3. **Thêm vào giỏ hàng**: Cho phép người dùng thêm sản phẩm vào giỏ hàng của họ.
     4. **Xem đơn hàng**: Cho phép người dùng xem các đơn hàng của họ.
     5. **Thêm sản phẩm yêu thích**: Cho phép người dùng thêm sản phẩm vào danh sách yêu thích của họ.
     6. **Thanh toán**: Cho phép người dùng thanh toán giỏ hàng
     7. Website quản lý

Giao diện trang website quản lý:

Hình 24 Giao diện website trang quản lý

Trong trang web người quản lý có thể thêm sửa xoá sản phẩm và người dùng, ngoài ra có thể tạo mã giảm giá, theo dõi đơn hàng và chỉnh sửa poster trong cửa hàng để thay đổi chương trình khuyến mãi.

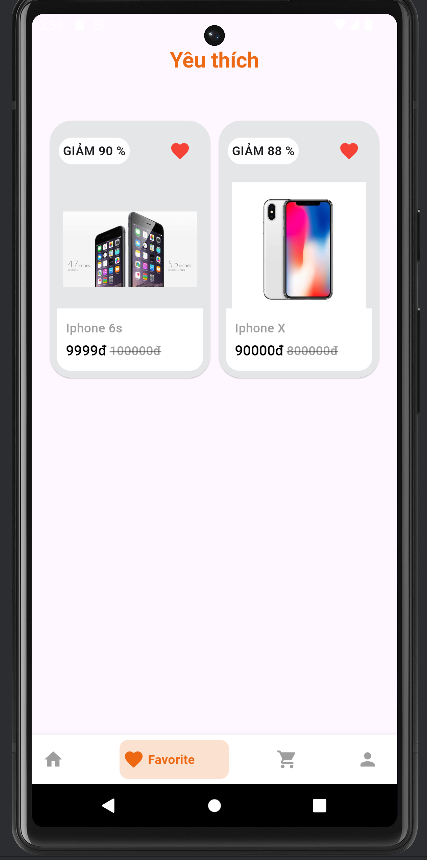
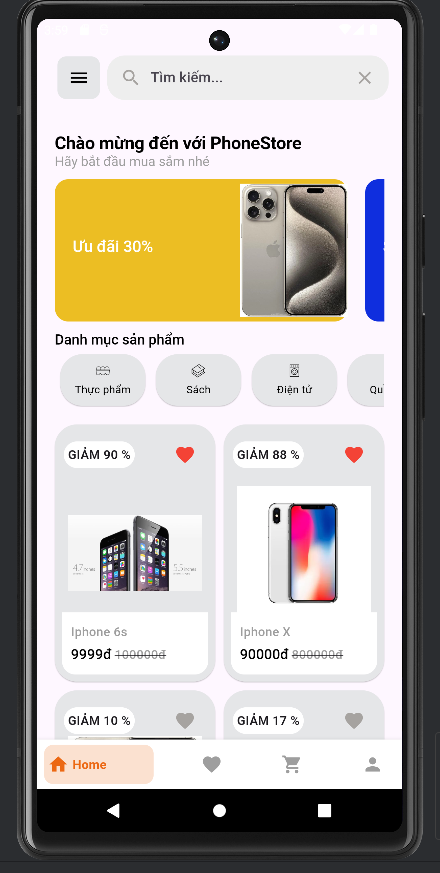
* + 1. Ứng dụng di động

Giao diện màn hình đăng nhập và đăng kí:

Hình 25 Giao diện đăng nhập và đăng kí

Giao diện đăng nhập và đăng kí cho phép người dùng đăng kí tài khoản và đăng nhập trực tiếp vào ứng dụng.

Giao diện màn hình chính và danh sách sản phẩm yêu thích:

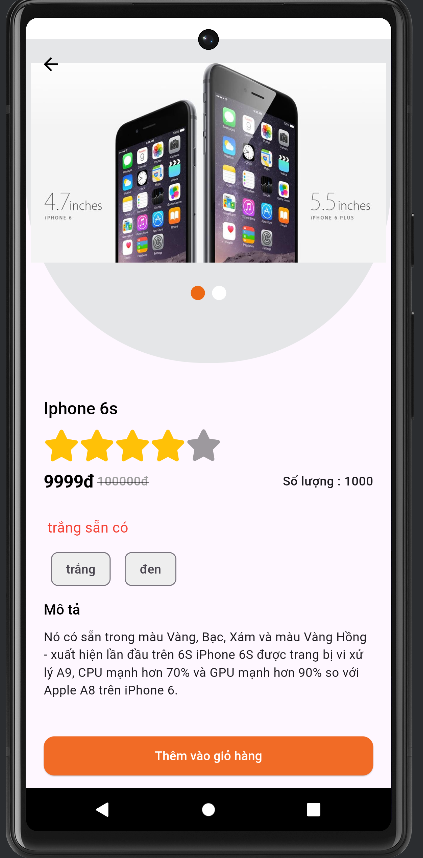
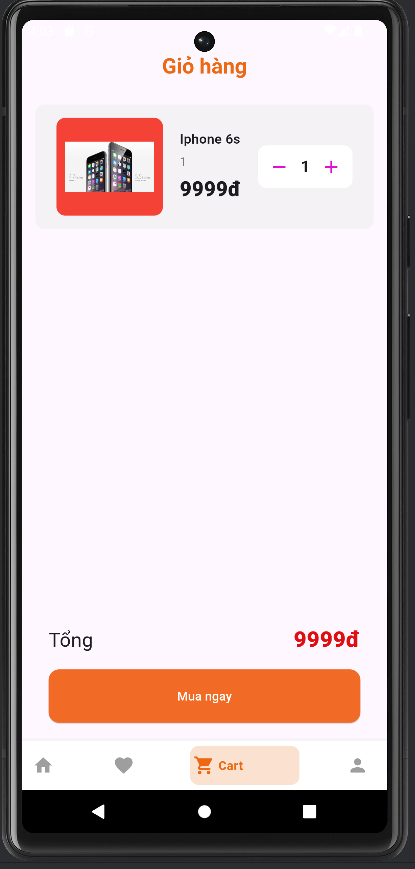


Hình 26 Giao diện màn hình chính và yêu thích

Giao diện chính của sản phẩm và danh sách sản phẩm yêu thích.

Khách hàng ấn yêu thích sản phẩm, sản phẩm sẽ được lưu trong mục danh sách yêu thích.

Giao diện chi tiết sản phẩm và giỏ hàng:

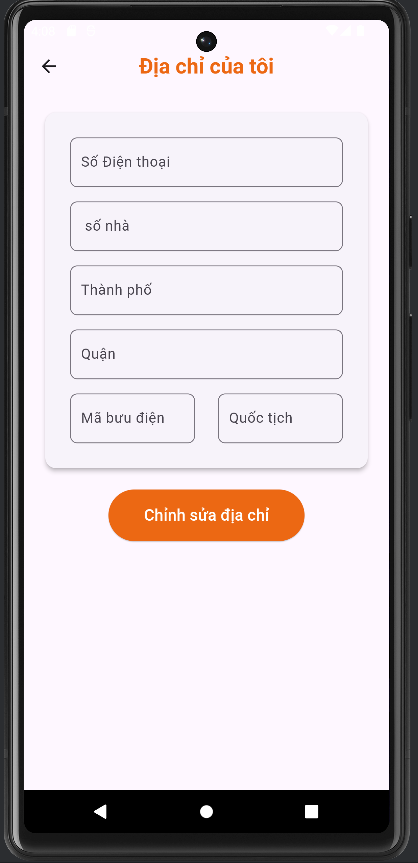
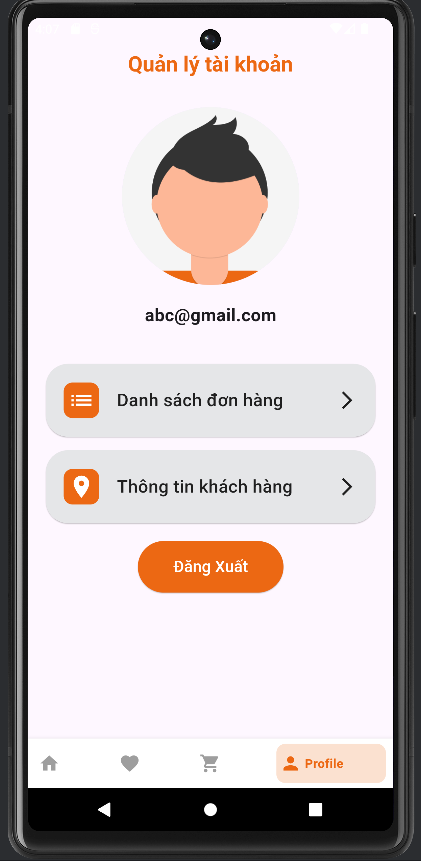


Hình 27 Giao diện chi tiết sản phẩm và giỏ hàng

Khách hàng ấn vào sản phẩm có thể xem chi tiết sản phẩm và thêm sản phẩm vào giỏ hàng.

Khách hàng ấn vào giỏ hàng có thể chỉnh sửa sản phẩm đã thêm và tiến hành thanh toán.

Giao diện trang cá nhân và thông tin:



Hình 28 Giao diện trang cá nhân và thông tin

Hiển thị tài khoản của người dùng và thông tin địa chỉ của người dùng.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. Kết quả đạt được

Sau thời gian tìm tòi và phát triển, em đã cơ bản hoàn thành được các yêu cầu bài toán đã đặt ra. Một số kết quả nhóm thu được như sau:

1. Về mặt lý thuyết: Tìm hiểu và vận dụng thành công 2 mô hình kiến trúc là Layered Architecture và Monolithic Architecture.
2. Về mặt thực nghiệm: Xây dựng thành công phần mềm bán điện thoại dựa trên những yêu cầu đặt ra như: đăng nhập, đăng ký, thêm sản phẩm…

* Hạn chế:

1. Đề tài mới chỉ được phát triển ở mức cơ bản nên không tránh khỏi việc các lỗi phát sinh trong quá trình hoạt động sau này.

2. Giao diện tương tác với người sử dụng chưa đạt được tính thẩm mĩ ở mức độ cao.

1. Hướng phát triển
2. Để đề tài có thể phong phú hơn và tính ứng dụng cao hơn thì trong tương lai cần phát triển hơn nữa về một số yêu cầu sau:
3. Hoàn thiện tính năng và triển khai thêm những chức năng khác nữa phù hợp với đề tài.
4. Nghiên cứu và phát triển vận dụng thêm các mô hình kiến trúc khác.

PHỤ LỤC

1. Link source code

https://github.com/lamduy04/App\_Android.git

1. Code mẫu

TÀI LIỆU THAM KHẢO

[1]. **Dart Programming Language** by Kathy Walrath, Seth Ladd - Đây là một tài liệu cơ bản về ngôn ngữ Dart, bao gồm các khái niệm cơ bản và các tính năng nâng cao của Dart.

[2]. **Flutter for Beginners: An introductory guide to building cross-platform mobile applications with Flutter and Dart 2** by Alessandro Biessek - Sách này dành cho người mới bắt đầu với Flutter, giải thích cách sử dụng Flutter và Dart để xây dựng các ứng dụng di động.

[3]. **MongoDB Applied Design Patterns** by Rick Copeland - Sách này tập trung vào các mẫu thiết kế được áp dụng cho MongoDB, giúp tối ưu hóa hiệu suất và sử dụng hiệu quả cơ sở dữ liệu NoSQL này.

[4]. **Node.js Design Patterns** by Mario Casciaro - Sách này cung cấp cái nhìn sâu sắc vào Node.js và các mẫu thiết kế phổ biến được áp dụng trong việc phát triển ứng dụng Node.js.