­Trường Đại học Bách Khoa Hà Nội

Viện Công nghệ Thông Tin và Truyền Thông



Đồ án Tốt nghiệp Đại học

Xây dựng ứng dụng tư vấn và quản lý cửa hàng thời trang

Đỗ Đức Thanh

Hà Nội, 06/2018

Trường Đại học Bách Khoa Hà Nội

Viện Công nghệ Thông Tin và Truyền Thông



Đồ án Tốt nghiệp Đại học

Xây dựng ứng dụng tư vấn và quản lý cửa hàng thời trang

[Sau khi in quyển, sinh viên xin chữ ký của giáo viên hướng dẫn ở đây. Chữ ký là xác nhận đồng ý cho phép bảo vệ của giáo viên hướng dẫn]

|  |  |
| --- | --- |
| Sinh viên thực hiện | Đỗ Đức Thanh |
| Người hướng dẫn | PGS-TS. Cao Tuấn Dũng |

­­

Hà Nội, 06/2018

Họ và tên sinh viên: Đỗ Đức Thanh

Điện thoại liên lạc: 01655871797 . Email: doducthanh95@gmail.com

Lớp: CNTT2.03 K58. Hệ đào tạo: đại học

Tôi Đỗ Đức Thanh cam kết Đồ án Tốt nghiệp (ĐATN) là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của PGS-TS Cao Tuấn Dũng. Các kết quả nêu trong ĐATN là trung thực, là thành quả của riêng tôi, không sao chép theo bất kỳ công trình nào khác. Mọi tham khảo trong ĐATN – bao gồm hình ảnh, bảng biểu, số liệu, và các câu từ trích dẫn – đều được ghi rõ ràng và đầy đủ nguồn gốc trong danh mục tài liệu tham khảo. Tôi xin hoàn toàn chịu trách nhiệm với dù chỉ một sao chép vi phạm quy chế của nhà trường.

|  |  |
| --- | --- |
|  | *Hà Nội, ngày 24 tháng 5 năm 2018*  Tác giả ĐATN  *Họ và tên sinh viên* |

# Lời cảm ơn

Em xin chân thành cảm ơn các thầy cô trong viện đã truyền đạt cho em những kiến thức và kinh nghiệm quý giá trong suốt 5 năm học qua để em có đủ kiến thức để hoàn thành đồ án này.

Em xin cảm ơn bộ môn Công nghệ Phần mềm, viện Công nghệ thông tin và Truyền thông đã tạo điều kiện để em được thực hiện đồ án này.

Em xin gửi lời cảm ơn sâu sắc đến thầy Cao Tuấn Dũng đã hướng dẫn, chỉ dạy chúng em tận tình trong suốt quá trình thực hiện đồ án này.

Cuối cùng, em xin được gửi lời cảm ơn chân thành đến gia đình bạn bè đã luôn tận tâm giúp đỡ, động viên, đóng góp ý kiến giúp em hoàn thành đồ án tốt nghiệp thành công.

# Tóm tắt

Ngày nay với sự phát triển nhanh chóng về kinh tế, các lĩnh vực kinh doanh cũng có sự phát triển vượt bậc. Thời trang vốn từ lâu đã là một trong những thành phần chủ đạo trong các nền kinh tế cũng không nằm trong vùng ngoại lệ. Ngoài những công ty, những cửa hàng lớn đã có từ trước, các đơn vị kinh doanh nhỏ lẻ, kinh doanh online cũng ngày một phát triển, nâng tầm về quy mô và số lượng. Thị trường kinh doanh thời trang vốn đã sôi động nay càng trở thành mảnh đất giàu tiềm năng cho các nhà đầu tư hơn.

Khi mở rộng quy mô kinh doanh, không chỉ các công ty, cửa hàng lớn mà ngay cả những hộ tiểu thương, kinh doanh online cũng đã và đang gặp những khó khăn trong việc quản lí, nắm bắt thông tin về tất cả sản phẩm trong cửa hàng của mình. Có nhiều cửa hàng, nhãn hiệu thời trang lớn đã tự xây Các chủ cửa hiệu cũng mất rất nhiều thời gian quản lí, cập nhật tình hình kinh doanh của các cửa hàng. Mặc dù một số nhãn hàng thời trang đã tự mình sản xuất các ứng dụng tra cứu và quản lý cho mình, song không phải cửa hàng nào cũng có điều kiện như vậy. Vì thế nhu cầu về quản lý và tư vấn vẫn còn rất lớn, và còn một lượng lớn khách hàng tiềm năng.

Nắm bắt được các vướng mắc đó, em đã đề xuất thực hiện ứng dụng tư vấn và quản lí bán hàng nhằm giúp các nhân viên tư vấn sản phẩm một cách hiệu quả, các chủ cửa hiệu cập nhật tình hình kinh doanh nhanh chóng. Các nhân viên bán hàng có thể tra cứu thông tin sản phẩm qua màu sắc, tên gọi, mã vạch, các sản phẩm bán chạy, khuyến mại,... Chủ cửa hiệu có thể nắm bắt được tình hình kinh doanh, doanh thu của các cửa hàng, phân tích xu hướng từ những tìm kiếm lựa chọn của người dùng để đưa ra chiến lược kinh doanh và nhập hàng hợp lí.

Bố cục đồ án chia thành 6 chương:

* Chương 1: Giới thiệu đề tài
* Chương 2: Khảo sát và phân tích yêu cầu
* Chương 3: Công nghệ sử dụng
* Chương 4: Phát triển và triển khai ứng dụng
* Chương 5: Các giải pháp và đóng góp nổi bật
* Chương 6: Kết luận và hướng phát triển

# Mục lục

[Danh mục hình vẽ…………………………………………………………...x](#_Toc514482580)

[Danh mục bảng……………………………………………………………xi](#_Toc514482581)

[Danh mục các từ viết tắt………………………………………………….xiii](#_Toc514482582)

[Danh mục thuật ngữ……………………………………………………….xiv](#_Toc514482583)

[Chương 1 Giới thiệu đề tài 1](#_Toc514482584)

[1.1 Đặt vấn đề 1](#_Toc514482585)

[1.2 Mục tiêu và phạm vi đề tài 1](#_Toc514482586)

[1.3 Định hướng giải pháp 2](#_Toc514482587)

[1.4 Bố cục đồ án 2](#_Toc514482588)

[Chương 2 Khảo sát và phân tích yêu cầu 4](#_Toc514482589)

[2.1 Khảo sát hiện trạng 4](#_Toc514482590)

[2.2 Tổng quan chức năng 5](#_Toc514482591)

[2.2.1 Biểu đồ use case tổng quan 5](#_Toc514482592)

[2.3 Đặc tả Usecase chức năng 7](#_Toc514482593)

[2.3.1 Đặc tả use case đăng nhập 7](#_Toc514482594)

[2.3.2 Đặc tả use case tìm kiếm sản phẩm theo tên 8](#_Toc514482595)

[Hình 3: Đặc tả usecase tìm kiếm sản phẩm theo tên 9](#_Toc514482596)

[3.3.6 Đặc tả use case tra cứu hàng mới về 9](#_Toc514482597)

[2.3.5 Đặc tả use case tra cứu hàng bán chạy 10](#_Toc514482598)

[Hình 5: Đặc tả use case tra cứu hàng bán chạy 11](#_Toc514482599)

[2.3.6 Đặc tả use case tra cứu hàng khuyến mại 11](#_Toc514482600)

[2.3.7 Đặc tả use case tra cứu hàng hoá tìm kiếm nhiều 12](#_Toc514482601)

[2.3.8 Đặc tả use case quản lý doanh thu cửa hàng 13](#_Toc514482602)

[2.3.9 Đặc tả use case quản lý sản phẩm 13](#_Toc514482603)

[2.4 Yêu cầu phi chức năng 14](#_Toc514482604)

[Chương 3 Công nghệ sử dụng 15](#_Toc514482605)

[3.1 Microservice 15](#_Toc514482606)

[3.2 Nodejs………………………………………………………………17](#_Toc514482607)

[3.3 Swift………………………………………………………………...18](#_Toc514482608)

[3.4 MySQL……………………………………………………………..19](#_Toc514482609)

[Chương 4 Phát triển và triển khai ứng dụng 22](#_Toc514482610)

[4.1 Thiết kế kiến trúc 22](#_Toc514482611)

[4.1.1 Lựa chọn kiến trúc phần mềm 22](#_Toc514482612)

[4.1.2 Thiết kế tổng quan 26](#_Toc514482613)

[4.2 Thiết kế chi tiết 27](#_Toc514482614)

[4.2.1 Thiết kế giao diện 27](#_Toc514482615)

[4.2.2 Thiết kế lớp 30](#_Toc514482616)

[4.2.3 Thiết kế cơ sở dữ liệu 34](#_Toc514482617)

[4.3 Xây dựng ứng dụng 36](#_Toc514482618)

[4.3.1 Thư viện và công cụ sử dụng 36](#_Toc514482619)

[4.3.2 Kết quả đạt được 37](#_Toc514482620)

[4.3.3 Minh hoạ các chức năng chính 38](#_Toc514482621)

[4.4 Kiểm thử 38](#_Toc514482622)

[4.5 Triển khai 38](#_Toc514482623)

[Chương 5 Các giải pháp và đóng góp nổi bật 39](#_Toc514482624)

[5.1 Những đóng góp của đề tài trong thực tế. 39](#_Toc514482625)

[5.2 Những khó khăn và giải pháp đặt ra trong ĐATN 42](#_Toc514482626)

[5.2.1 Giải quyết vấn đề tối ưu truy vấn. 42](#_Toc514482627)

[5.2.2 giải quyết vấn đề bảo mật 43](#_Toc514482628)

[Chương 6 Kết luận và hướng phát triển 46](#_Toc514482629)

[6.1 Kết luận 46](#_Toc514482630)

[6.2 Hướng phát triển 47](#_Toc514482631)

[Tài liệu tham khảo 48](#_Toc514482632)

# Danh mục hình vẽ

[**Hình 1.1** Xu hướng chi tiêu trong tháng của người tiêu dùng Việt Nam 2](#_Toc515021947)

[**Hình 1.2** Mật độ mua sắm thời trang của người Việt Nam 3](#_Toc515021948)

[**Hình 1.3** hành vi mua hàng của người dung 4](#_Toc515021949)

[**Hình 2.1** Biều đồ usecase tổng quan 12](#_Toc515021950)

[**Hình 2.2** đặc tả usecase đăng nhập 14](#_Toc515021951)

[**Hình 2.3** đặc tả usecase tìm kiếm sản phẩm theo tên 15](#_Toc515021952)

[**Hình 2.4** đặc tả usecase tra cứu hàng mới về 16](#_Toc515021953)

[**Hình 2.5** đặc tả usecase tra cứu hàng bán chạy 17](#_Toc515021954)

[**Hình 2.6** đặc tả usecase tra cứu hàng khuyến mại 18](#_Toc515021955)

[**Hình 2.7** đặc tả usecase tra cứu hàng hoá tìm kiếm nhiều 19](#_Toc515021956)

[**Hình 2.8** Đặc tả use case quản lý doanh thu cửa hàng 20](#_Toc515021957)

[**Hình 2.9** đặc tả usecase quản lý sản phẩm 21](#_Toc515021958)

[**Hình 3.1** Kiến trúc microservice 23](#_Toc515021959)

[**Hình 3.2** Mô hình triển khai hệ thống ở mức cao 24](#_Toc515021960)

[**Hình 3.3** Cơ chế hoạt động của Node JS 26](#_Toc515021961)

[**Hình 4.1** Mô hình kiến trúc tổng quan ứng dụng 31](#_Toc515021962)

[**Hình 4.2** Mô hình hoạt động MVC 32](#_Toc515021963)

[**Hình 4.3** Mô hình MVVM 32](#_Toc515021964)

[**Hình 4.4** View Login 33](#_Toc515021965)

[**Hình 4.5** File Account.swift 34](#_Toc515021966)

[**Hình 4.6** ViewModel trong Login 34](#_Toc515021967)

[**Hình 4.7** Ví dụ biểu đồ phụ thuộc gói 35](#_Toc515021968)

[**Hình 4.8** Các package tham gia vào chức năng đăng nhập 36](#_Toc515021969)

[**Hình 4.9** Các package tham gia vào hiển thị thông tin model 37](#_Toc515021970)

[**Hình 4.10** Các package tham gia vào nhiệm vụ hiển thị tình hình doanhh thu 38](#_Toc515021971)

[**Hình 4.11** Minh hoạ các thành phần của Login 40](#_Toc515021972)

[**Hình 4.12** Minh hoạ cho chức năng tra cứu 41](#_Toc515021973)

[Hình 4.13 Minh hoạ cho chức năng xem chi tiết sản phẩm 42](#_Toc515021974)

[**Hình 4.14** Minh hoạ cho chức năng quản lí doanh thu 43](#_Toc515021975)

[**Hình 4.15** Minh hoạ cho chức năng quản lí sản phẩm 44](#_Toc515021976)

[**Hình 4.16** Thiết kế lớp Model\_SQL 45](#_Toc515021977)

[**Hình 4.17** Thiết kế lớp DataModel 46](#_Toc515021978)

[**Hình 4.18** Thiết kế chi tiết cho class RequestAPI 47](#_Toc515021979)

[**Hình 4.19** Biểu đồ tuần tự cho chức năng đăng nhập 48](#_Toc515021980)

[**Hình 4.20** Biểu đồ trình tự cho chức năng tra cứu theo tên 49](#_Toc515021981)

[**Hình 4.21** Biều đồ tuần tự thống kê doanh thu 50](#_Toc515021982)

[**Hình 4.22** Biểu đồ tuần tự cho chức năng hiển thị top tìm kiếm sản phẩm 51](#_Toc515021983)

[**Hình 4.23** Các thực thể liên kết 51](#_Toc515021984)

[**Hình 4.24** Mô hình thực thể liên kết 52](#_Toc515021985)

[**Hình 4.25** Các bảng cơ sở dữ liệu 53](#_Toc515021986)

[**Hình 4.26** Chức năng đăng nhập 58](#_Toc515021987)

[**Hình 4.27** Chức năng xem sản phẩm 59](#_Toc515021988)

[**Hình 4.28** Chức năng xem chi tiết sản phẩm 60](#_Toc515021989)

[**Hình 5.1** Mô tả chức năng của index trong cơ sở dữ liệu 66](#_Toc515021990)

[**Hình 5.2** Cách thức hoạt động của JWT 67](#_Toc515021991)

[**Hình 5.3** Token trả về của phương thức đăng nhập 68](#_Toc515021992)

# Danh mục bảng

[**Bảng 4.1** Quy chuẩn thiết kế giao diện ứng dụng 39](#_Toc515020756)

[**Bảng 4. 2** Giải thích phương thức trong class Model\_SQL 45](#_Toc515020757)

[Bảng 4. 3 giải thích các phương thức trong class DataModel 46](#_Toc515020758)

[**Bảng 4. 4** Giải thích các phương thức trong class RequestAPI 47](#_Toc515020759)

[**Bảng 4. 5** Đặc tả thông tin bảng Shop 53](#_Toc515020760)

[**Bảng 4. 6** đặc tả thông tin bảng Model 54](#_Toc515020761)

[**Bảng 4. 7** Đặc tả thông tin bảng Item 55](#_Toc515020762)

[**Bảng 4.8** Danh sách thư viện và công cụ sử dụng 56](#_Toc515020763)

[**Bảng 4. 9** Thống kê project 57](#_Toc515020764)

[**Bảng 4. 10** Kiểm thử cho chức năng đăng nhập 61](#_Toc515020765)

[**Bảng 4. 11** Kiểm thử chức năng đổi mật khẩu 61](#_Toc515020766)

[**Bảng 4. 12** Kiểm thử chức năng chọn hiển thị hàng hoá theo tên 62](#_Toc515020767)

[**Bảng 4. 13** Kiểm thử chức năng hiển thị chi tiết sản phẩm 63](#_Toc515020768)

# Danh mục các từ viết tắt

|  |  |
| --- | --- |
| API | Application Programming Interface  Giao diện lập trình ứng dụng |
| **CNTT** | Công nghệ thông tin |
| **ĐATN** | Đồ án tốt nghiệp |
| **SV**  **JWT** | Sinh viên  Json web token |

# Danh mục thuật ngữ

|  |  |
| --- | --- |
| Compiler  Client  Server | Trình biên dịch  Khách hàng, người sử dụng |

# Giới thiệu đề tài

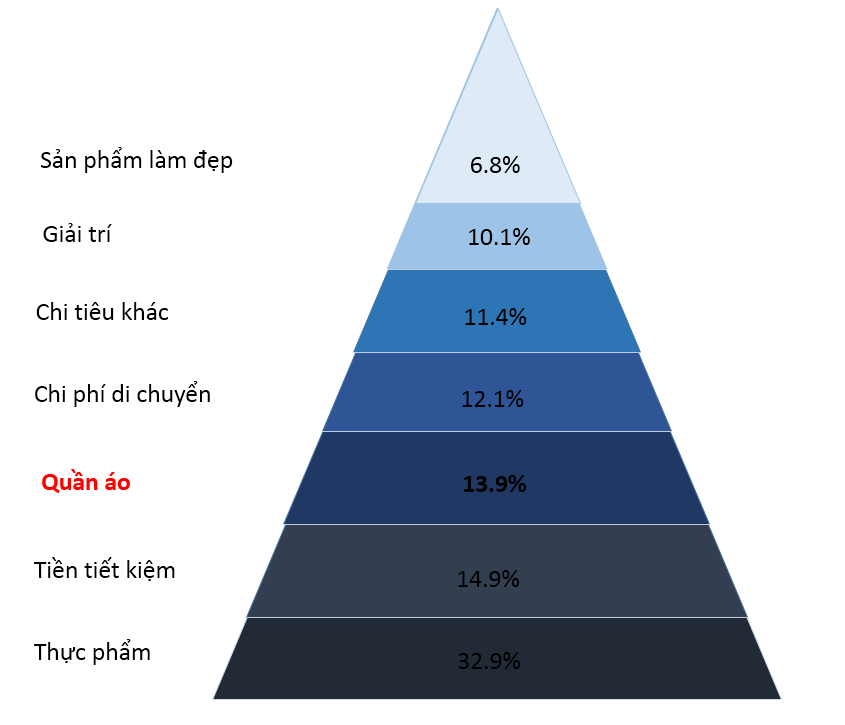
## Đặt vấn đề

Từ xưa đến nay, thời trang hay buôn bán quần áo vẫn luôn chiếm một tỉ trọng lớn trong ngành công thương của mỗi quốc gia, mỗi thời kì. Quần áo luôn là một mặt hàng thiết yếu không thể thiếu được trong đời sống xã hội con người.

Theo thống kê cuối năm 2017 của trang web maybanhang.net thì Việt Nam là một trong số những nước đứng đầu khu vực về tiêu thụ sản phẩm thời trang. Ở thị trường Việt Nam hiện nay có rất nhiều sản phẩm thời trang với đa dạng xuất xứ và chủng loại khác nhau. Ở phân khúc thị trường cao cấp, thị trường thời trang Việt chịu nhiều ảnh hưởng từ những sản phẩm đến từ các thương hiệu lớn nước ngoài như Banana Republic, Zara, Mango, Levis, … Ở phân khúc tầm trung là sự du nhập các sản phẩm đến từ Trung Quốc, Thái Lan, Hồng Kông, … Ngoài ra chưa kể đến một loạt những thương hiệu sản xuất có tiếng trong nước. Tất cả những điều đó đã tạo nên một thị trường thời trang đa dạng, phong phú và là lĩnh vực giàu tiềm năng cho các nhà đầu tư.

Theo khảo sát được thực hiện bởi The Conference Board® và Nielsen - một công ty đo lường hiệu quả kinh doanh toàn cầu và công bố ngày 29/3/2018, gần phân nửa số người khảo sát dùng tiền nhàn rỗi để mua sắm quần áo và du lịch. Con số này chiếm 49%, một con số khá ấn tượng và phản ánh đúng nhu cầu hiện nay của người dân Việt Nam.

Một số liệu phân tích khác về tỉ lệ các mục chi tiêu hàng tháng cũng cho thấy người Việt chúng ta dành khá nhiều cho lĩnh vực thời trang và mua sắm



**Hình 1.1** Xu hướng chi tiêu trong tháng của người tiêu dùng Việt Nam

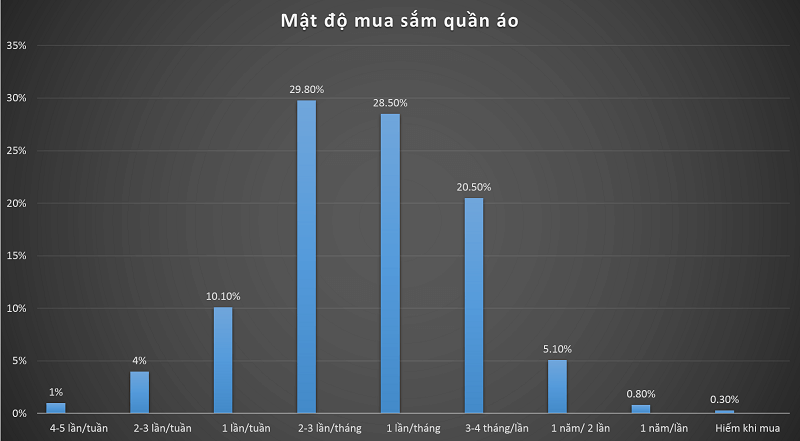
Từ những phân tích thực tế trên có thể thấy lĩnh vực kinh doanh thời trang ở Việt Nam đang rất có tiềm năng phát triển và hứa hẹn sẽ đem lại doanh thu lớn. Ở Việt Nam hiện nay tồn tại khá nhiều hình thức kinh doanh thời trang. Thứ nhất là những nhãn hiệu, cửa hàng thời trang lớn và có tiếng. Thứ hai là những tiểu thương, hộ kinh doanh vừa và nhỏ. Cuối cùng là những các nhân, đơn vị kinh doanh online. Đối với những nhãn hiệu cửa hàng lớn, họ đã tự xây dựng cho mình những hệ thống hỗ trợ và quản lý bán hàng. Có thể kể đến một vài sản phẩm như Lazada, Geek, Zalora, Shopee, … Những đối tượng còn lại mặc dù vẫn có nhu cầu quản lý và bán hàng xong họ không có đủ chi phí để đầu tư riêng cho mình những sản phẩm đắt đỏ như vậy.

Đi từ thực trạng của những vấn đề hiện nay, trong ĐATN của mình em đã lựa chọn đề tài **xây dựng ứng dụng hỗ trợ tư vấn, tìm kiếm và quản lý bán hàng**. Ứng dụng này sẽ đảm bảo giải quyết các vấn đề tồn đọng hiện nay của người dùng, đem lại thêm những lựa chọn cho các đơn vị kinh doanh thời trang.

## Mục tiêu và phạm vi đề tài

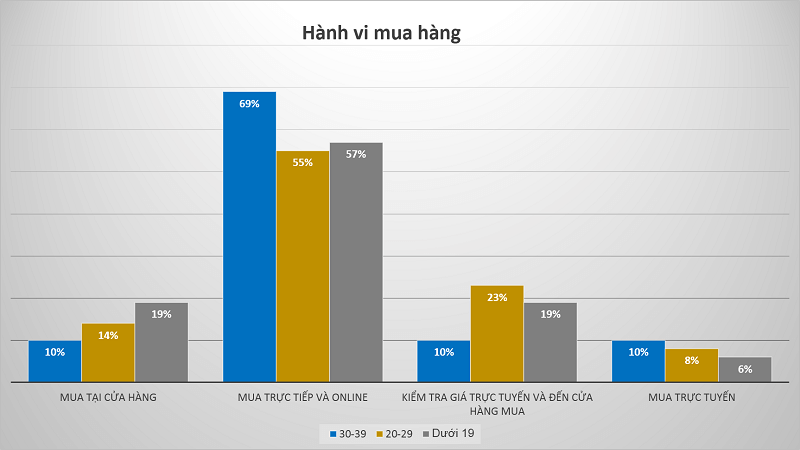
Với các vấn đề đã nêu ra ở mục 1.1 em đã xác định những mục tiêu cụ thể cho bài toán của mình. Đề tài hướng đến giải quyết những khó khăn trong công việc kinh doanh thời trang. Cụ thể đó là những vấn đề về việc nắm bắt thông tin của các sản phẩm và tinh hình doanh thu đối với từng cửa hàng cụ thể.

Theo một con số thống kê, trung bình mật độ mua sắm quần áo của người Việt là khoảng 1 lần/tháng.



**Hình 1.2** Mật độ mua sắm thời trang của người Việt Nam

Ngoài ra một số liệu khác đáng quan tâm đó là thói quen mua sắm của khách hàng đó là chủ yếu tại các cửa hàng và một phần là mua sắm online. Hình 3 dưới mô tả hành vi mua hàng của của người dân Việt Nam.



**Hình 1.3** hành vi mua hàng của người dung

Khách hàng tập trung mua hàng cao nhất vào những khung giờ từ 18h – 22h, và thường có xu hướng tăng cao vào các dịp lễ tết và cuối tuần. Chính vì thế mà áp lực sẽ càng tăng cao hơn đối với những cửa hàng vừa và nhỏ với số lượng nhân viên ít ỏi.

Hiện nay trên thị trường cũng đã có những sản phẩm với mục đích tương tự, xong nó còn những hạn chế riêng. Các sản phẩm này chủ yếu do những nhãn hiệu, cửa hàng lớn tự xây dựng nhằm phục vụ cho cửa hàng của họ. Chính vì thế mà những hộ kinh doanh vừa và nhỏ rất khó để tiếp cận với những sản phẩm này.

Với ứng dụng mà ĐATN đề xuất, mục tiêu mà ứng dụng hướng đến sẽ là:

* Giúp nhân viên bán hàng tại các cửa hàng hoặc buôn bán online nâng cao được hiệu suất làm việc, từ đó đáp ứng được nhiều khách hàng hơn.
* Giúp nhân viên không phải ghi nhớ hết thông tin về sản phẩm mà vẫn có thể tư vấn tốt cho khách hàng của mình. Các sản phẩm sẽ được tư vấn chi tiết đến từng mục cụ thể như màu sắc, kích thước và vị trí hiện tại của nó.
* Chủ cửa hàng sẽ nắm bắt tốt hơn tình hình kinh doanh các sản phẩm của cửa hàng.

Phạm vi mà đề tài hướng đến đó là những đơn vị kinh doanh vừa và nhỏ. Đây là những khách hàng khá đa dạng và tiềm năng. Mặc dù doanh thu từ những đối tượng này không lớn song ngược lại số lượng và tỉ trọng của các nhóm đối tượng này lại chiếm khá cao trong thị phần thời trang hiện nay. Ngoài ra, với chi phí sản phẩm không quá cao nên rất dễ dàng tiếp cận và thuyết phục nhóm khách hàng này sử dụng.

## Định hướng giải pháp

Với yêu cầu cần xây dựng một ứng dụng tư vấn và quản lý cửa hàng thời trang do đó cần thiết kế một hệ thống giúp hiển thị thông tin phù hợp với yêu cầu của người sử dụng. Những thông tin này có thể bao gồm: thông tin về hàng hoá, thông tin về tình trạng sản phẩm, thông tin về doanh thu, … Do đó mô hình phù hợp nhất trong bài toán này là Client – Server.

Mô hình Client – Server là một mô hình nổi tiếng và được áp dụng rộng rãi. Ý tưởng của mô hình này là máy con (đóng vài trò là máy khách) gửi một yêu cầu (request) để máy chủ (đóng vai trò người cung ứng dịch vụ), máy chủ sẽ xử lý và trả kết quả về cho máy khách. Áp dụng cho bài toán trên vì lí do thứ nhất là ứng dụng phục vụ cho nhiều cửa hàng, do đó cần một nơi lưu trữ tập trung. Thứ hai dữ liệu bán hàng có sự thay đổi nhanh vì thế cần có giải pháp đảm bảo rằng khi dữ liệu thay đổi các khách hàng cũng phải cập nhật được. Mô hình Client – Server sẽ đáp ứng đủ những yêu cầu trên.

### Giải pháp định hướng xây dựng server

Trong bài toán này em cần xây dựng một server với nhiệm vụ lưu trữ dữ liệu và truyền nhận với client. Hiện nay có nhiều công nghệ hỗ trợ để xây dựng server với chức năng tương tự, tuy nhiên để phù hợp hơn với bài toán này em đã lựa chọn công nghệ WebAPI vì những lí do sau:

* Đây là một trong những framework mới sẽ giúp ích trong việc xây dựng các HTTP service một cách rất đơn giản và nhanh chóng.
* Mã nguồn mở nên có thể được sử dụng bởi bất kì một client nào hỗ trợ XML, JSON.
* Nó cũng có khả năng hỗ trợ đầy đủ các thành phần HTTP: URL, request/response headers, caching, versioning, content forma.
* Một kiểu kiến trúc vô cùng phù hợp dành cho các thiết bị trang bị băng thông giới hạn như smartphone, tablet.
* Hỗ trợ đầy đủ các thành phần MVC như: routing, controller, action result, filter, model binder, IoC container, dependency injection, unit test.
* Dễ dàng triển khai và sử dụng.

### Giải pháp định hướng xây dựng client

Nhiệm vụ chính của client là hiển thị dữ liệu từ server theo những nội dung phù hợp với yêu cầu bài toán. Đối tượng người sử dụng chính mà mục tiêu bài toán hướng đến gồm nhân viên cửa hàng và chủ cửa hàng. Đây là những đối tượng với đặc điểm cần hoạt động di chuyển linh hoạt, ít ngồi một chỗ. Do đó việc chọn phát triển ứng dụng phía client lên destop là không hợp lý. Ngoài ra hiện nay với sự bùng nổ của các thiết bị di động thông minh với hiệu năng sử dụng không thua kém gì các sản phẩm destop mà giá cả đầu tư ít nên em lựa chọn phát triển ứng dụng để chạy trên nền tảng mobile.

Với các thiết bị mobile, người dùng chỉ cần có internet là có thể sử dụng được. Các dữ liệu sẽ được cập nhật nhanh chóng nếu có thay đổi. Việc sử dụng mobile sẽ đảm bảo được tính linh hoạt hơn, tiện dụng hơn. Ngoài ra ứng dụng mobile còn có những trải nghiệm sử dụng tốt hơn so với các ứng dụng destop.

## Bố cục đồ án

Đồ án tốt nghiệp này được tổ chức như sau:

* Chương 1 trình bày giới thiệu đề tài cũng như các vấn đề tổng quan. Trong chương này, em trình bày nội dung chính về tên đề tài, lí do chọn đề tài cũng như một số giải pháp định hướng cho đề tài được chọn.
* Chương 2 trình bày về khảo sát và phân tích các yêu cầu của khách hàng, tìm hiểu về quy trình nghiệp vụ bán hang. Trong chương này, nội dung khảo sát và phân tích yêu cầu khách hàng được làm rõ. Các biểu đồ tổng quan về mặt chức năng cũng được đề cập đến trong chương này.
* Chương 3 trình bày về các công nghệ và lý thuyết để sử dụng giải quyết bài toán đặt ra. Trong chương này em sẽ trình bày chủ yếu lý thuyết tổng quan của các công nghệ được, thư viện hỗ trợ được sử dụng trong ĐATN.
* Chương 4 trình bày về quy trình phát triển và triển khai ứng dụng.
* Chương 5 em xin trình bày về giải pháp và các đóng góp cụ thể của bài toán. Chương này sẽ tập trung làm rõ tác dụng của đề tài với người sử dụng thực tế, các khó khăn và hướng giải quyết trong đề tài.
* Chương 6 sẽ trình bày những kết luận và một số định hướng phát triển sản phẩm trong tương lai

# ­­Khảo sát và phân tích yêu cầu

## Khảo sát hiện trạng

Trên thị trường hiện nay có khá nhiều ứng dụng được phát triển để phục vụ cho các cửa hàng kinh doanh thời trang. Đặc điểm chính của những ứng dụng này là được phát triển tập trung cho 1 cửa hàng, nhãn hàng thời trang riêng biệt. Ngoài các ứng dụng cho mobile, nhiều đơn vị kinh doanh cũng lựa chọn web làm nền tảng phát triển. Bảng 1 dưới đây là kết quả so sánh những tính năng của một số sản phẩm hiện có trên thị trường.

**Bảng 2.1** so sánh các ứng dụng dành cho cửa hàng thời trang

|  |  |  |
| --- | --- | --- |
| STT | Sản phẩm | Chức năng cung cấp |
| 1 | Lazada, Shopee | * Bán hàng * Tra cứu hàng hoá theo đầu mục * Đặt hàng * Hỗ trợ vận chuyển hàng * Xem chi tiết hàng hoá tìm kiếm |
| 2 | Stylebook | * Tạo tủ đồ quần áo online * Gợi ý phối hợp trang phục |
| 3 | The Hunt | * Tìm kiếm sản phẩm theo ảnh * Gợi ý địa chỉ mua hàng |
| 4 | The Net Set | Ứng dụng này cho phép bạn chụp lại hình ảnh một đôi giày mà bạn để ý, tại một buổi hòa nhạc, trên đường đi hay ở một nhà hàng nào đó. Sau đó nó sẽ giúp bạn tìm kiếm hàng loạt những mặt hàng tương tự. |
| 5 | Sapo | * Bán hàng * Quản lí đặt hàng * Quản lý giao hàng * Trả hàng |

Những sản phẩm hiện có trên thị trường đa phần hỗ trợ cho việc bán hàng và vận chuyển. Mặc dù có những sản phẩm hỗ trợ khá tốt cho việc kinh doanh như lazada, shopee,.. tuy nhiên điểm hạn chế của những sản phẩm đó là việc chỉ áp dụng cho 1 đơn vị cụ thể. Chính vì thế, loại sản phẩm đó rất khó để có thể mở rộng và áp dụng một cách rộng rãi. Ngoài ra chi phí để bỏ ra để mua và sử dụng cho các đơn vị kinh doanh vừa và nhỏ cũng là gần như không thể. Đối với một số sản phẩm quốc tế, do đặc thù thị trường và thói quen của người dùng nên các sản phẩm này thường cũng có những chức năng khác tập trung vào đối tượng phục vụ khách mua hàng nhiều hơn.

Ở thị trường thời trang Việt Nam hiện nay có sự phân chia khá rõ giữa những đơn vị kinh doanh. Đầu tiên là những thương hiệu cửa hàng nổi tiếng như Owen, Ninomax, Blue Exchange, Viettien, May10, … Đây là những cửa hàng lớn với hàng loạt chi nhánh lớn nhỏ khắp cả nước. Có thể coi họ là những ông lớn trong ngành kinh doanh thời trang cao cấp. Đa phần những cửa hàng này thường phát triển riêng cho mình những công cụ hỗ trợ kinh doanh. Thứ hai là những cửa hàng thời trang có quy mô vừa và nhỏ, tuy nhiên bù lại số lượng đơn vị này lại rất lớn và chiếm một tỉ trọng khá cao trên thị trường. Đặc điểm chung của những đơn vị kinh doanh này là kinh doanh sản phẩm đa dạng, diện tích cửa hàng vừa phải và không có quá nhiều chi nhánh. Thứ ba đó là các cá nhân kinh doanh online. Hình thức kinh doanh này mới phát triển vài năm trở lại đây, song tốc độ tăng trưởng lại khá cao. Hai đối tượng kể trên có thể coi là những khách hàng giàu tiềm năng nhất để sử dụng các sản phẩm hỗ trợ kinh doanh. Đây cũng chính là những khách hàng mà ứng dụng trong ĐATN của em hướng đến.

Về mặt yêu cầu chức năng, từ những tìm hiểu thực tế của mình đối với những đối tượng kể trên em đã thu nhận được những yêu cầu chính như sau:

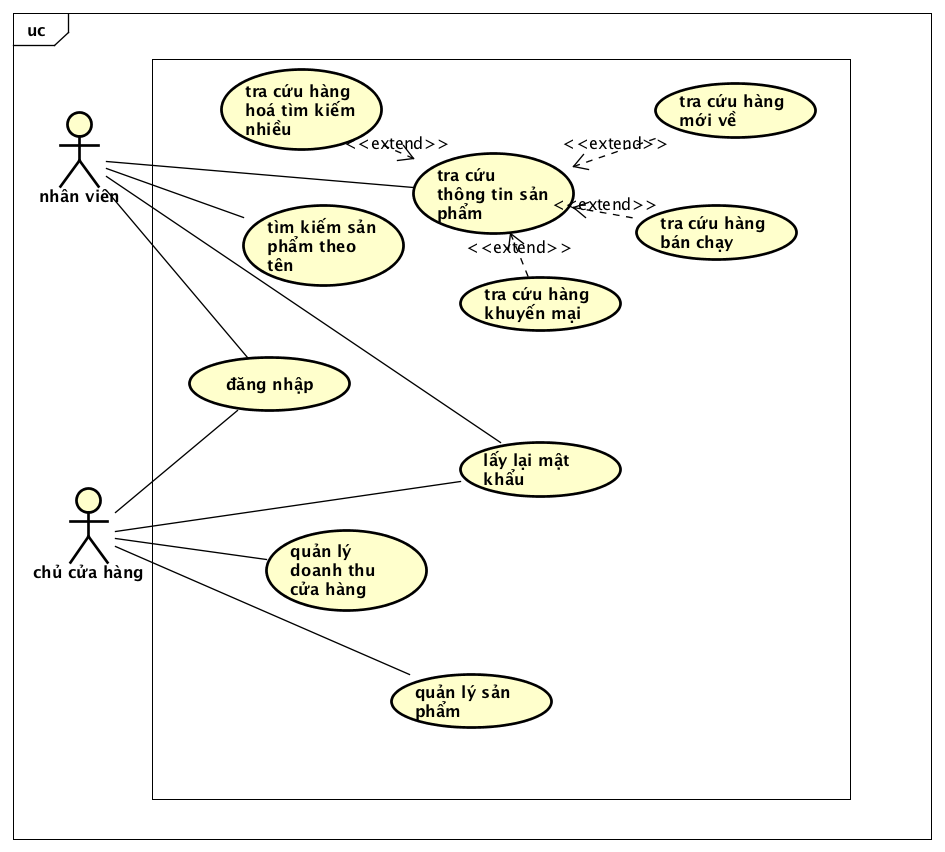
**Bảng 2.2** yêu cầu chức năng nhiệm vụ

|  |  |  |
| --- | --- | --- |
| STT | Chức năng | Chú thích |
| 1 | Hiển thị các sản phẩm kinh doanh | Các chức năng hỗ trợ nhân viên kinh doanh, nhân viên bán hàng tại các cửa hàng |
| 2 | Cho phép hiển thị đầu mục tên các sản phẩm |
| 3 | Tìm kiếm nhanh sản phẩm |
| 4 | Lọc sản phẩm mới về |
| 5 | Lọc sản phẩm khuyến mại |
| 6 | Lọc sản phẩm bán chạy |
| 7 | Xem chi tiết sản phẩm, gồm size, color, giá, vị trí theo cửa hàng, chi nhánh |
| 8 | Thống kê doanh thu tại các cửa hàng | Các chức năng phục vụ việc quản lý thông tin của chủ cửa hàng. |
| 9 | Thống kê top doanh thu theo các chi nhánh |
| 10 | Thống kê danh sách sản phẩm đang bán chạy |
| 11 | Thống kê danh sách xu hướng tìm kiếm sản phẩm của khách hàng |
| 12 | Tạo event khuyến mại cho sản phẩm |

## Tổng quan chức năng

### Biểu đồ use case tổng quan

Hình 1 dưới đây mô tả usecase tổng quan cho toàn bộ ứng dụng.



**Hình 2.1** Biều đồ usecase tổng quan

Trong ứng dụng có 2 tác nhân sử dụng chính gồm: nhân viên cửa hàng và chủ cửa hàng.

* Nhân viên cửa hàng: các nhân viên tại mỗi cửa hàng sẽ được cấp 1 tài khoản dùng để truy cập vào ứng dụng tư vấn. Tại ứng dụng này, nhân viên có thể tra cứu nhanh thông tin về sản phẩm của cửa hàng để tư vấn cho khách hàng của mình.
* Chủ cửa hàng: mỗi chủ cửa hàng cũng được cấp 1 tài khoản riêng để sử dụng ứng dụng quản lý của mình. Chủ cửa hàng có thể xem tình hình doanh thu của cả cửa hàng hoặc của từng chi nhánh theo thời gian. Ngoài ra chủ cửa hàng còn có thể xem được top doanh thu của các cửa hàng, top sản phẩm bán chạy và top tìm kiếm của người dung. Từ những thông tin đó, chủ cửa hàng có thể đưa ra các phân tích để có chiến lược kinh doanh hiệu quả

Các usecase chính bao gồm:

* Usecase đăng nhập: cho phép người dùng đăng nhập và sử dụng ứng dụng
* Usecase tra cứu thông tin sản phẩm: cho phép các nhân viên cửa hàng tìm kiếm nhanh thông tin sản phẩm theo size, màu sắc, …
* Usecase tra cứu hàng mới về là một usecase mở rộng của usecase tra cứu thông tin sản phẩm. Usecase này cho phép hiển thị các sản phẩm mới nhất
* Usecase tra cứu hàng bán chạy cho phép hiển thị tất cả các sản phẩm đang bán chạy nhất của cửa hàng đó.
* Usecase tra cứu hang khuyến mại cho phép hiển thị danh sách sản phẩm khuyến mại của cửa hàng.
* Usecase tra cứu hàng hoá tìm kiếm nhiều cho phép hiển thị danh sách các sản phẩm được khách hàng quan tâm, tìm kiếm nhiều nhất.
* Usecase tìm kiếm sản phẩm: cho phép nhân viên tìm kiếm nhanh thông tin về sản thông qua tên.
* Usecase quản lý doanh thu: cho phép chủ cửa hàng xem chi tiết doanh thu của từng cửa hàng theo thời gian
* Usecase quản lý sản phẩm: cho phép chủ cửa hàng xem top doanh thu, top mặt hàng bán chạy cũng như top tìm kiếm của người hàng.

## Đặc tả Usecase chức năng

### Đặc tả use case đăng nhập

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã usecase** | 01 | **Tên usecase** | Đăng nhập |
| **Tác nhân** | Nhân viên cửa hàng, chủ cửa hàng | | |
| **Tiền điều kiện** | Không | | |
| **Luồng sự kiện chính** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | | 1. | Hệ thống | Hiển thị giao diện đăng nhập | | 2. | Người dùng | Nhập thông tin đăng nhập | | 3. | Hệ thống | Kiểm tra các trường dữ liệu nhập | | 4. | Hệ thống | Gửi thông tin đăng nhập lên server | | 5. | Hệ thống | Trả lại token đăng nhập | | 6. | Hệ thống | Lưu thông tin đăng nhập và token, đồng  thời chuyển sang màn hình chính | | | |
| **Luồng sự kiện thay thế** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | | 3a | Hệ thống | Thông báo trường dữ liệu sai | | 5a | Hệ thống | Thông báo sai thông tin đăng nhập | | | |
| **Hậu điều kiện** | Không | | |

**Hình 2.2** đặc tả usecase đăng nhập

### Đặc tả use case tìm kiếm sản phẩm theo tên

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã usecase** | 02 | **Tên usecase** | Tìm kiếm sản phẩm theo tên |
| **Tác nhân** | Nhân viên cửa hàng | | |
| **Tiền điều kiện** | Đăng nhập với tài khoản nhân viên | | |
| **Luồng sự kiện chính** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | | 1. | Nhân viên | Nhập tên tìm kiếm | | 2. | Hệ thống | Kiểm tra trường dữ liệu nhập có rỗng  hay không | | 3. | Hệ thống | Gửi yêu cầu lên server | | 4. | Hệ thống | Nhận dữ liệu từ server và hiển thị | | | |
| **Luồng sự kiện thay thế** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | | 2a | Hệ thống | Thông báo tên tìm kiếm không được để trống | | | |
| **Hậu điều kiện** | Không | | |

**Hình 2.3** đặc tả usecase tìm kiếm sản phẩm theo tên

### Đặc tả use case tra cứu hàng mới về

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã usecase** | 03 | **Tên usecase** | Tra cứu hang mới về |
| **Tác nhân** | Nhân viên cửa hàng | | |
| **Tiền điều kiện** | Đăng nhập với tài khoản nhân viên | | |
| **Luồng sự kiện chính** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | | 1. | Nhân viên | Chọn option tra cứu ang mới về | | 2. | Hệ thống | Gửi yêu cầu lên server | | 3. | Hệ thống | Nhận dữ liệu trả về và hiển thị | | | |
| **Luồng sự kiện thay thế** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | | 2a | Hệ thống | Thông báo mất kết nối hoặc không  kết nối được | | | |
| **Hậu điều kiện** | Không | | |

**Hình 2.4** đặc tả usecase tra cứu hàng mới về

### 2.3.5 Đặc tả use case tra cứu hàng bán chạy

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã usecase** | 04 | **Tên usecase** | Tra cứu hang bán chạy |
| **Tác nhân** | Nhân viên cửa hàng | | |
| **Tiền điều kiện** | Đăng nhập với tài khoản nhân viên | | |
| **Luồng sự kiện chính** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | | 1. | Nhân viên | Chọn option tra cứu ang bán chạy | | 2. | Hệ thống | Gửi yêu cầu lên server | | 3. | Hệ thống | Nhận dữ liệu trả về và hiển thị | | | |
| **Luồng sự kiện thay thế** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | | 2a | Hệ thống | Thông báo mất kết nối hoặc không  kết nối được | | | |
| **Hậu điều kiện** | Không | | |

**Hình 2.5** đặc tả usecase tra cứu hàng bán chạy

### 2.3.6 Đặc tả use case tra cứu hàng khuyến mại

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã usecase** | 05 | **Tên usecase** | Tra cứu hàng khuyến  mại |
| **Tác nhân** | Nhân viên cửa hàng | | |
| **Tiền điều kiện** | Đăng nhập với tài khoản nhân viên | | |
| **Luồng sự kiện chính** | |  |  |  |  | | --- | --- | --- | --- | | **STT** | **Thực hiện bởi** | | **Hành động** | | 1. | Nhân viên | Chọn option tra cứu hang khuyến mại | | | 2. | Hệ thống | Gửi yêu cầu lên server | | | 3. | Hệ thống | Nhận dữ liệu trả về và hiển thị | | | | |
| **Luồng sự kiện thay thế** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | | 2a | Hệ thống | Thông báo mất kết nối hoặc không  kết nối được | | | |
| **Hậu điều kiện** | Không | | |

**Hình 2.6** đặc tả usecase tra cứu hàng khuyến mại

### Đặc tả use case tra cứu hàng hoá tìm kiếm nhiều

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã usecase** | 06 | **Tên usecase** | Tra cứu hàng hoá tìm kiếm nhiều |
| **Tác nhân** | Nhân viên cửa hàng | | |
| **Tiền điều kiện** | Đăng nhập với tài khoản nhân viên | | |
| **Luồng sự kiện chính** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | | 1. | Nhân viên | Chọn option tra cứu hàng hoá tìm kiếm nhiều | | 2. | Hệ thống | Gửi yêu cầu lên server | | 3. | Hệ thống | Nhận dữ liệu trả về và hiển thị | | | |
| **Luồng sự kiện thay thế** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | | 2a | Hệ thống | Thông báo mất kết nối hoặc không  kết nối được | | | |
| **Hậu điều kiện** | Không | | |

**Hình 2.7** đặc tả usecase tra cứu hàng hoá tìm kiếm nhiều

### 2.3.8 Đặc tả use case quản lý doanh thu cửa hàng

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã usecase** | 07 | **Tên usecase** | Quản lý doanh thu cửa hang |
| **Tác nhân** | Chủ cửa hàng | | |
| **Tiền điều kiện** | Đăng nhập với tài khoản chủ cửa hang | | |
| **Luồng sự kiện chính** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | | 1. | Nhân viên | Chọn chức năng quản lý doanh thu cửa hàng | | 2. | Hệ thống | Gửi yêu cầu lên server | | 3. | Hệ thống | Nhận dữ liệu trả về và vẽ đồ thị | | | |
| **Luồng sự kiện thay thế** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | | 2a | Hệ thống | Thông báo mất kết nối hoặc không  kết nối được | | | |
| **Hậu điều kiện** | Không | | |

**Hình 2.8** Đặc tả use case quản lý doanh thu cửa hàng

### 2.3.9 Đặc tả use case quản lý sản phẩm

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã usecase** | 08 | **Tên usecase** | Quản lý sản phẩm |
| **Tác nhân** | Chủ cửa hàng | | |
| **Tiền điều kiện** | Đăng nhập với tài khoản chủ cửa hang | | |
| **Luồng sự kiện chính** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | | 1. | Nhân viên | Chọn chức năng quản lý sản phẩm | | 2. | Hệ thống | Gửi yêu cầu lên server | | 3. | Hệ thống | Nhận dữ liệu trả về và vẽ đồ thị | | | |
| **Luồng sự kiện thay thế** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | | 2a | Hệ thống | Thông báo mất kết nối hoặc không  kết nối được | | | |
| **Hậu điều kiện** | Không | | |

**Hình 2.9** đặc tả usecase quản lý sản phẩm

## Yêu cầu phi chức năng

Ứng dụng được sử dụng bởi cả nhân viên và chủ cửa hàng, do đó phải đảm bảo tính dễ dùng. Ngoài ra ứng dụng phải đáp ứng được những yêu cầu về bảo mật cơ bản trong việc gửi nhận dữ liệu.

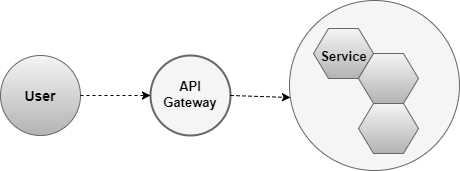
Ứng dụng tra cứu phục vụ cho nhân viên bán hàng nên tốc độ truy vấn dữ liệu phải nhanh hơn để đáp ứng yêu cầu phục vụ.

# Công nghệ sử dụng

### 3.1 Microservice

Microservice là một mô hình hóa và thực thi hóa cho mô-hình-hướng-dịch-vụ[[1]](#footnote-1), được sử dụng để xây dựng các hệ thống phần mềm linh hoạt và dễ triển khai.

Hình 3.1 là cấu trúc đơn giản nhất của một kiến trúc microservice. Tùy vào từng hệ thống, hướng phát triển mà Kiến trúc microservice có ít nhiều thay đổi. Nhưng nguyên lý hoạt động thì gần như không thay đổi.



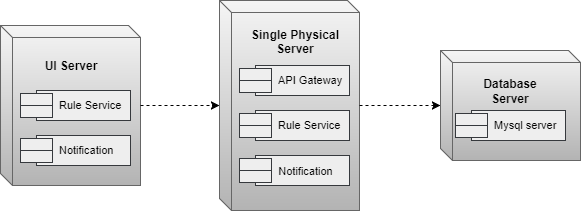
**Hình 3.1** Kiến trúc microservice

Hệ thống được chia nhỏ thành các Service, mỗi service có thể giao tiếp với một Cơ sở dữ liệu khác nhau. Người dùng giao tiếp với các service thông qua API Gateway. Khi bất cứ một service nào ngừng hoạt động do một nguyên nhân nào đó thì các service khác có thể hoạt động mà không bị ảnh hưởng.

Ưu điểm của kiến trúc microservice:

* Hiệu năng tốt
* Tái sử dụng được
* Dễ phát triển
* Dễ bảo trì
* Bảo mật tốt
* Khả năng chịu lỗi
* Khả năng tích hợp, mở rộng

Hình 3.2 dưới đây là một ví dụ cho mô hình triển khai hệ thống theo kiến trúc microservice.



**Hình 3.2** Mô hình triển khai hệ thống ở mức cao

API Gateway: là một component[[2]](#footnote-2) của mô hình Microservice. *API Gateway* là “lối vào duy nhất” cho tất cả các client. *API Gateway* xử lý các yêu cầu (request) bằng một trong hai cách, với các yêu cầu đơn giản chỉ là được định tuyến hoặc được ủy quyền cho service nhất định thì nó sẽ gửi đến các dịch vụ (service) phù hợp. Với các request khác nó xử lý bằng cách đưa ra nhiều dịch vụ (multiple services).

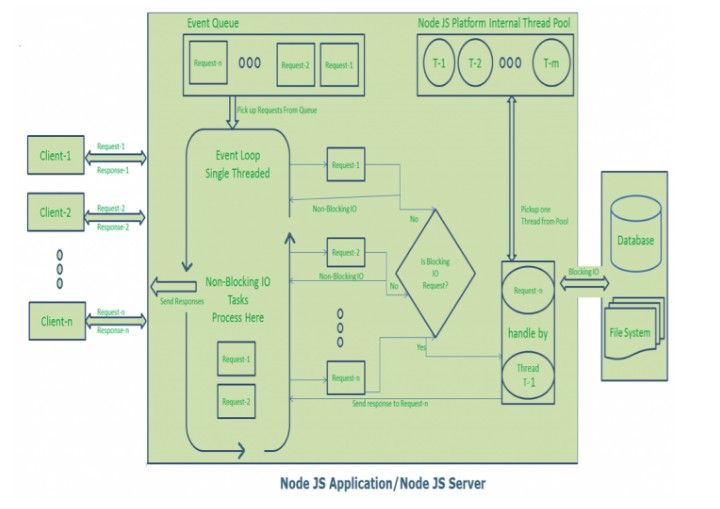
Notification: là một component của hệ thống. Notification là một microservice, nhiệm vụ của nó là gửi các thông báo như SMS, EMAIL, PUSH NOTIFICATION, v.v. Notification cũng quản lý mẫu cho nội dung của thông báo.

Registry: Registry là một "dịch vụ đăng ký"[[3]](#footnote-3), là một component của kiến trúc microservice, nó có thể được coi là cơ sở dữ liệu của các dịch vụ khác. Các cá thể dịch vụ (Service instances) được đăng ký các quyền, chức năng bởi service registry khi hệ thống khởi động, và được dỡ đăng ký khi hệ thống bị tắt. Khách hàng (client) của một Service nào đó truy vấn service registry để tìm kiếm đối tượng có sẵn của dịch vụ. Service registry có thể gọi một API để xác nhận xem một service có thể xử lý các yêu cầu không.

### 3.2 Nodejs

Tổng quan về Nodejs:   
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.

Các ứng dụng Node JS sử dụng kiến trúc "*Single Threaded Event Loop Model*" để xử lý đồng thời rất nhiều clients. Dươi đây là hình ảnh minh hoạ cho cơ chế hoạt động của Node JS.



**Hình 3.3** Cơ chế hoạt động của Node JS

Giải thích các bước xử lí trong mô hình đơn Threaded.

* Client gưi yêu cầu đến Web Server
* Node JS Web Server duy trì một Limited Thread Pool để cung cấp dịch vụ cho các yêu cầu của clients.
* Node JS Web Server nhận được các yêu cầu đó và đưa chúng vào hàng đợi. Nó được hiểu như là là "Event Queue".
* Node JS Web Server nội bộ có một Components, gọi là "Event Loop". Tại sao nó có tên này là vì nó sử dụng vòng lặp vô hạn để nhận yêu cầu và xử lý chúng.
* Event Loop sử dụng một Single Thread duy nhất. Đó là trọng tâm của mô hình Node JS Platform Processing Model.
* Event Loop kiểm tra bất kỳ client request nào được đặt trong Event Queue. Nếu không, nó sẽ chờ các request đến.
* Node JS Webserver bắt đầu chọn 1 yêu cầu từ Event Queue và bắt đầu quá trình xử lý yêu cầu.
* Nếu client request không yêu cầu bất kỳ Blocking IO Operations, thì xử lý mọi thứ và chuẩn bị phản hồi và gửi lại cho client.
* Nếu client request yêu cầu một số Blocking IO Operations như tương tác với cơ sở dữ liệu, file systems, extenal services thì nó sẽ theo cách tiếp cận khác.
* Kiểm tra các Thread có sẵn từ Internal Thread Pool
* Chọn một Thread và chỉ định request client này cho Thread đó.
* Thread này chịu trách nhiệm lấy request đó, xử lý nó, thực hiện các Blocking IO Operations, chuẩn bị phản hồi và gửi nó trở lại Event Loop.
* Event Loop lần lượt gửi phản hồi cho client tương ứng.

Nhờ sử dụng cơ chế Single Threaded Event Loop nên Node JS có thể xử lí nhiều yêu cầu của client một các dễ dàng. Node JS application sử dụng ít Threads để nó có thể sử dụng ít tài nguyên hoặc bộ nhớ hơn nhờ đó mà nâng cao hiệu năng.

### 3.3 Swift

Swift là ngôn ngữ chính thức được Apple thừa nhận để viết các sản phẩm mobile hoặc PC chạy trên hệ điều hành IOS. So với những ngôn ngữ khác, Swift có tuổi đời còn khá trẻ song không thể vì thế mà phủ nhận những ưu điểm vượt trội của ngôn ngữ này. Sau đây là một số đặc điểm của ngôn ngữ Swift mà em sử đụng để viết ứng dụng này:

* Swift là một ngôn ngữ lập trình hướng đối tượng. Sự ra đời của Swift có thể giúp các lập trình viên dễ dàng tiếp cận bởi nó sử dụng mã nguồn mở.
* Các dòng code của Swift ngắn và dễ đọc… giúp đẩy nhanh quá trình phát triển ứng dụng và tăng hiệu quả cho quá trình bảo trì trong tương lai.
* Swift tạo ra các ứng dụng có sức ép phần cứng thấp hơn Objective-C, người dùng sẽ được trải nghiệm game ấn tượng hơn trên các thiết bị iOS.
* Swift là một ngôn ngữ lập trình di động an toàn hơn sơ với người tiềm nhiệm của nó.

### 3.4 MySQL

MySQL[[4]](#footnote-4) là hệ quản trị cơ sở dữ liệu SQL nguồn mở phổ biến nhất, được phát triển, phân phối và hỗ trợ bởi Oracle Corporation.

**MySQL là một hệ thống quản lý cơ sở dữ liệu.**

Cơ sở dữ liệu là tập hợp có cấu trúc của dữ liệu. Nó có thể là bất cứ thứ gì từ một danh sách mua sắm đơn giản đến một thư viện hình ảnh hoặc một số lượng lớn thông tin trong một mạng lưới công ty. Để thêm, truy cập và xử lý dữ liệu được lưu trữ trong cơ sở dữ liệu, bạn cần một hệ thống quản lý cơ sở dữ liệu như MySQL Server. Vì các máy tính rất giỏi xử lý lượng lớn dữ liệu, các hệ thống quản lý cơ sở dữ liệu đóng vai trò trung tâm trong việc tính toán, như các tiện ích độc lập, hoặc như các phần của các ứng dụng khác.

**MySQL là Cơ sở dữ liệu quan hệ.**

Một cơ sở dữ liệu quan hệ lưu trữ dữ liệu trong các bảng riêng biệt thay vì đặt tất cả dữ liệu vào một kho lưu trữ lớn. Các cấu trúc cơ sở dữ liệu được tổ chức thành các tệp vật lý được tối ưu hóa cho tốc độ. Mô hình lôgic, với các đối tượng như cơ sở dữ liệu, bảng biểu, khung nhìn, hàng và cột, cung cấp môi trường lập trình linh hoạt. Bạn thiết lập các quy tắc điều chỉnh mối quan hệ giữa các trường dữ liệu khác nhau, chẳng hạn như một-một, một-nhiều, duy nhất, bắt buộc hoặc tùy chọn, và “các con trỏ” giữa các bảng khác nhau. Cơ sở dữ liệu thực thi các quy tắc này, do đó, với một cơ sở dữ liệu được thiết kế tốt, ứng dụng của bạn không bao giờ thấy dữ liệu không nhất quán, trùng lặp, đơn độc, đã lỗi thời hoặc thiếu.

Phần SQL của “MySQL” là viết tắt của “Structured Query Language” . SQL là ngôn ngữ tiêu chuẩn phổ biến nhất được sử dụng để truy cập cơ sở dữ liệu. Tùy thuộc vào môi trường lập trình của bạn, bạn có thể nhập SQL trực tiếp (ví dụ: để tạo báo cáo), nhúng các câu lệnh SQL vào mã được viết bằng ngôn ngữ khác hoặc sử dụng API cụ thể cho ngôn ngữ để ẩn cú pháp SQL.

**Phần mềm MySQL là mã nguồn mở.**

Mã nguồn mở có nghĩa là mọi người có thể sử dụng và sửa đổi phần mềm. Bất kỳ ai cũng có thể tải về phần mềm MySQL từ Internet và sử dụng nó mà không phải trả bất cứ điều gì. Nếu bạn muốn, bạn có thể nghiên cứu mã nguồn và thay đổi nó cho phù hợp với nhu cầu của bạn. Phần mềm MySQL sử dụng GPL (GNU General Public License), để xác định những gì bạn có thể và không thể làm với phần mềm trong các tình huống khác nhau. Nếu bạn cảm thấy không thoải mái với GPL hoặc cần phải nhúng mã MySQL vào một ứng dụng thương mại, bạn có thể mua một phiên bản thương mại được cấp phép.

Máy chủ cơ sở dữ liệu MySQL rất nhanh, đáng tin cậy, khả năng mở rộng và dễ sử dụng.

MySQL Server có thể chạy thoải mái trên máy tính để bàn hoặc máy tính xách tay, cùng với các ứng dụng khác của bạn, máy chủ web, v.v ... cần ít hoặc không có sự chú ý. Nếu bạn dành toàn bộ một máy cho MySQL, bạn có thể điều chỉnh các thiết lập để tận dụng tất cả bộ nhớ, sức mạnh của CPU và dung lượng I/O. MySQL cũng có thể quy mô lên đến các cụm máy móc, kết nối với nhau.

MySQL Server ban đầu được phát triển để xử lý cơ sở dữ liệu lớn nhanh hơn nhiều so với các giải pháp hiện tại và đã được sử dụng thành công trong các môi trường sản xuất đòi hỏi khắt khe trong nhiều năm. Mặc dù dưới sự phát triển liên tục, MySQL Server hôm nay cung cấp một bộ các chức năng phong phú và hữu ích. Khả năng kết nối, tốc độ và bảo mật của nó làm cho MySQL Server rất phù hợp với việc truy cập cơ sở dữ liệu trên Internet.

**MySQL Server làm việc trong máy khách/máy chủ hoặc các hệ thống nhúng.**

Phần mềm Cơ sở dữ liệu MySQL là một hệ thống máy khách/máy chủ bao gồm một máy chủ SQL đa luồng hỗ trợ các kết thúc ngược lại, một số chương trình khách hàng khác nhau và thư viện, các công cụ hành chính và một loạt các giao diện lập trình ứng dụng (API).

Chúng tôi cũng cung cấp MySQL Server như một thư viện đa luồng nhúng mà bạn có thể liên kết với ứng dụng của bạn để có được sản phẩm độc lập nhỏ hơn, nhanh hơn và dễ quản lý hơn.

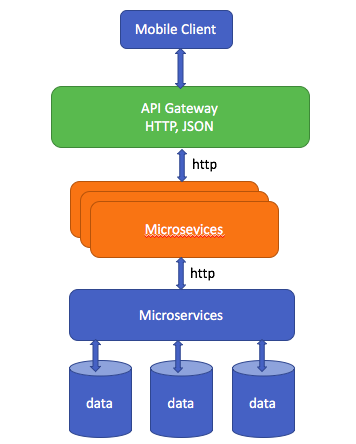
MySQL Server có một bộ tính năng thiết thực được phát triển với sự hợp tác chặt chẽ với người dùng của chúng tôi. Rất có thể ứng dụng hoặc ngôn ngữ yêu thích của bạn hỗ trợ máy chủ cơ sở dữ liệu MySQL.

# Phát triển và triển khai ứng dụng

## Thiết kế kiến trúc

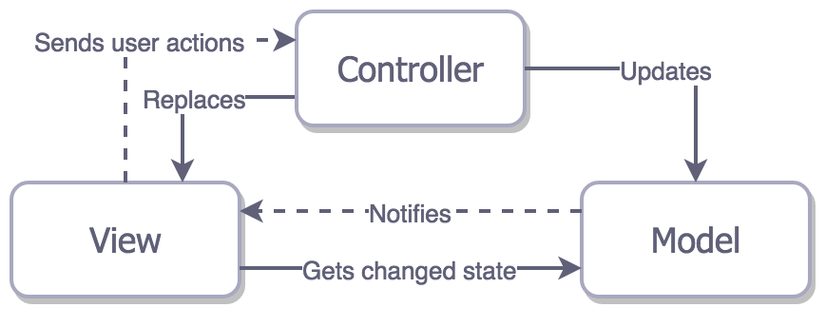
### Lựa chọn kiến trúc phần mềm

Mô hình kiến trúc tổng quan của toàn ứng dụng được mô tả theo hình



**Hình 4.1** Mô hình kiến trúc tổng quan ứng dụng

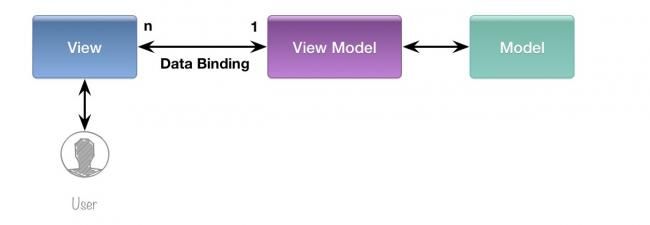
Tại client em đã lựa chọn mô hình MVVM để xây dựng phía mobile. Với các ứng dụng mobile trước đây, chúng ta đã khá quen thuộc với mô hình kinh điển MVC (Module-View-Controller). Hình dưới đây mô tả cách thức hoạt động của mô hình MVC.



**Hình 4.2** Mô hình hoạt động MVC

Đây là mô hình có từ lâu và rất được phổ biến trong quy trình phát triển phần mềm. Tuy nhiên một trong số những nhược điểm lớn nhất của mô hình khi áp dụng để triển khai ứng dụng mobile đó là việc phân tách nhiệm vụ giữa View và Controller không được rõ ràng. Do còn tồn tại sự nhập nhằng giữa View và Controller dẫn đến việc test chức năng cho các ứng dụng trở lên khó khăn hơn.

MVVM là một mô hình được sinh ra nhằm khắc phục những khuyết điểm tồn tại của mô hình MVC. Hình dưới đây mô tả cách thức hoạt động của mô hình MVVM



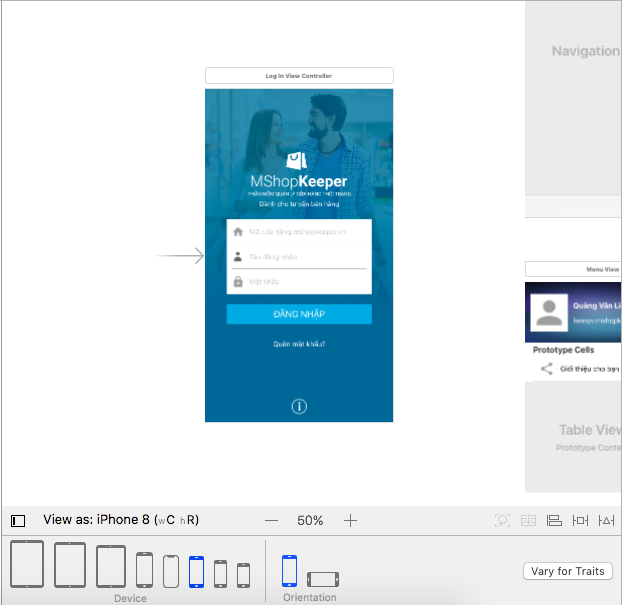
**Hình 4.3** Mô hình MVVM

Thành phần ViewModel được sinh ra nhằm giúp điều hướng giữa View và Model. Từ đó giúp cho View không bị phình to khi project phát triển.

ViewModel có tính chất tái sử dụng trong code, giúp phân tách rõ ràng viewController và view. Trong ngôn ngữ Swift, thành phần View bao gồm các file storyboad, file xib, các file viewController

Ví dụ về mô hình MVVM được sử dụng trong ĐATN.

Màn hình LoginViewController được thiết kế trong file MainStoryboard và đóng vai trò là View trong kiến trúc



**Hình 4.4** View Login

Tiếp đến là file Account đóng vai trò Model (hình 15)



**Hình 4.5** File Account.swift

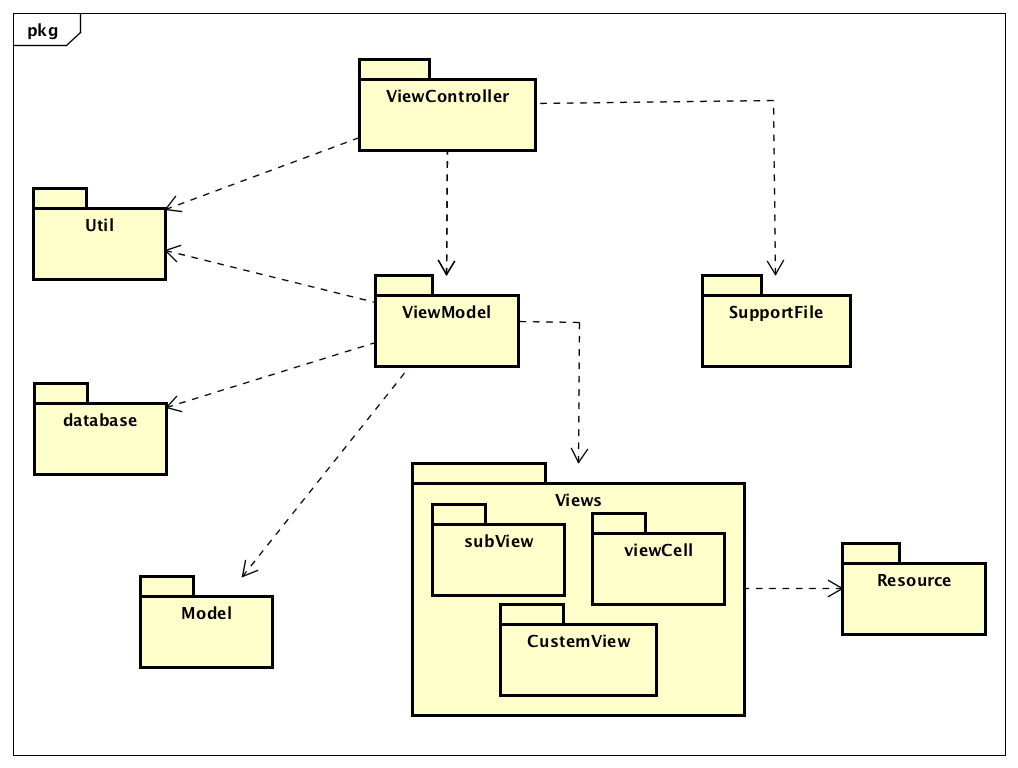
Cuối cùng là thành phần ViewModel trung gian điều khiển việc lưu dữ liệu và hiển thị. Trong file LoginViewController sử dụng 2 ViewModel là UserModel dùng để gọi phương thức lưu lại thông tin đăng nhập người dùng và ViewModel RequestAPI chứa các phương thức tương tác với API – hình



**Hình 4.6** ViewModel trong Login

Khi người dùng nhấm vào button đăng nhập view login sẽ gọi đến ViewModel RequestAPI thực thi phương thức login. Sau khi thực hiện xong, RequestAPI sẽ trả lại một số thông tin thông qua callback của hàm để view có thể tiếp tục thực hiện công việc của mình. Các luồng thực hiện logic sẽ không được viết trực tiếp trong các lớp View.

### Thiết kế tổng quan

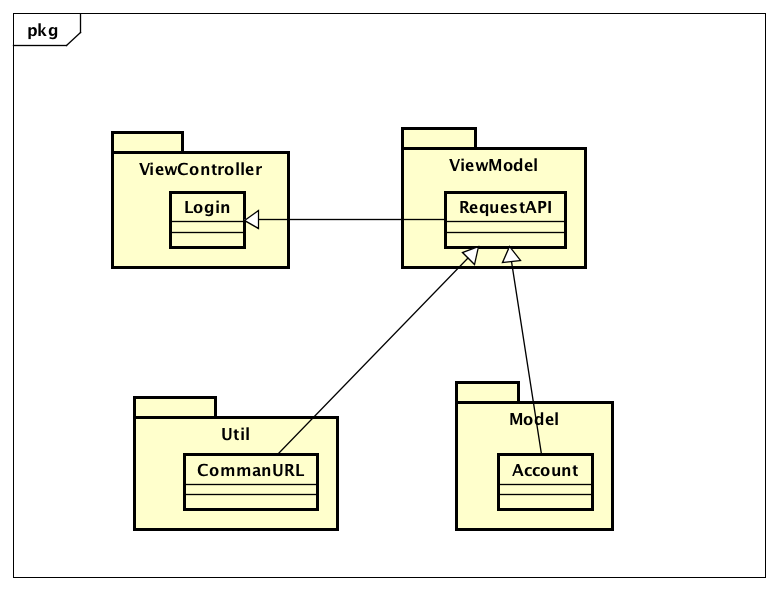


**Hình 4.7** Ví dụ biểu đồ phụ thuộc gói

Mô tả chức năng các package:

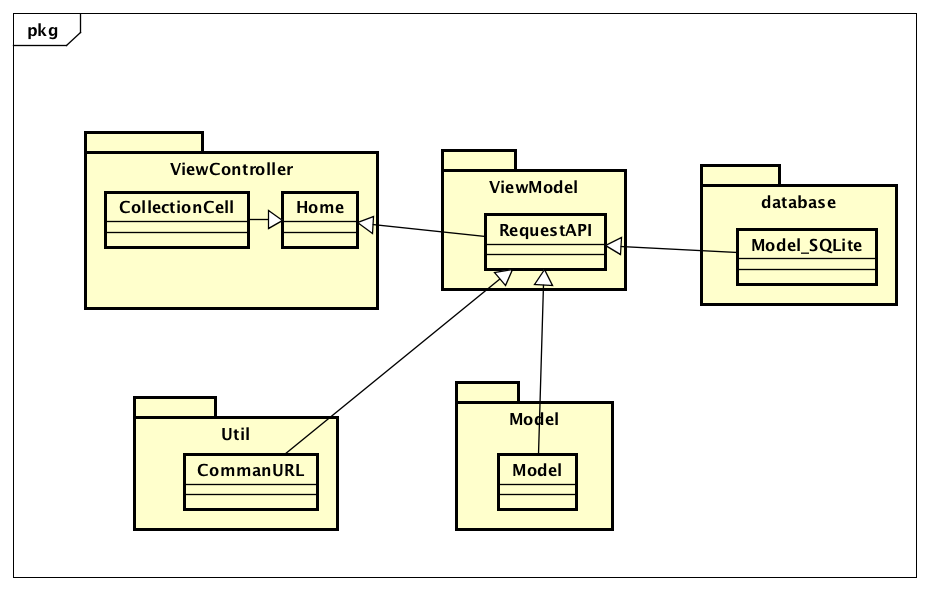
* Package ViewController: Bao gồm các file Swift để quản lý các màn hình giao diện.
* Package Model: chứa các mô tả thực thể, đối tượng trong project.
* Package ViewModel: Gồm các file điểu khiển việc hiển thị thành phần cho view.
* Package Util: Gồm các file định nghĩa các hàm tiện ích để các lớp trong dự án có thể gọi sử dụng.
* Package Resource: Gồm các file ảnh được sử dụng
* Package SupportFile: bao gồm các file quản lý của Xcode, file giao diện tổng quan.
* Package database: bao gồm các file khởi tạo và thực thi liên quan đến data local.

### Thiết kế chi tiết gói



**Hình 4.8** Các package tham gia vào chức năng đăng nhập

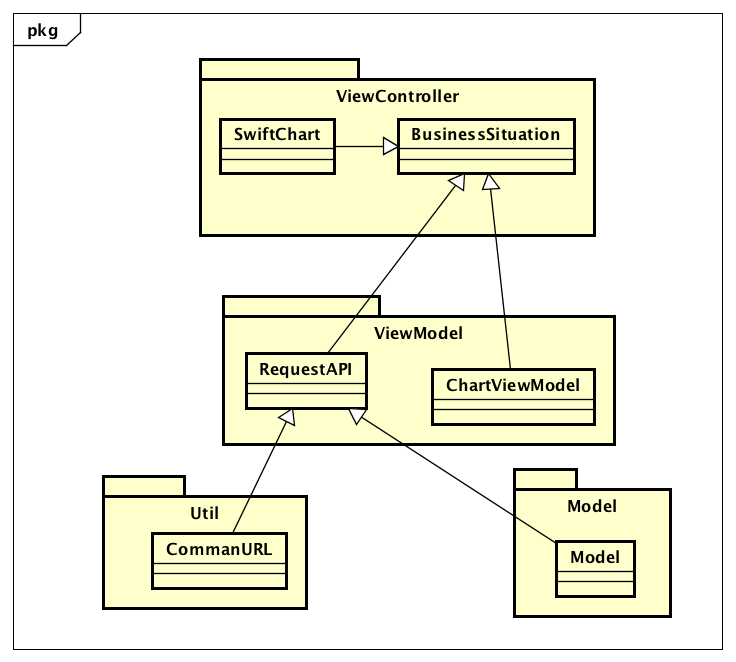
Khi người dùng nhấn vào nút đăng nhập trên màn hình login, view login sau khi kiểm tra trạng thái các trường dữ liệu đầu vào sẽ gọi đến ViewModel RequestAPI để gửi thông tin lên server. Tại hàm *requestLogin()* sẽ xử lý bằng cách truyền các tham số đường dẫn lấy từ class CommanURL trong package Util cùng với các giá trị từ view Login gửi sang. Sau khi truy vấn dữ liệu ViewModel RequestAPI sẽ lưu lại thông tin account và gửi kết quả lại cho view Login tiếp tục xử lí.



**Hình 4.9** Các package tham gia vào hiển thị thông tin model

Sau khi đăng nhập thành công với tài khoản nhân viên, ứng dụng sẽ chuyển sang màn hình Home để hiển thị danh sách các model đang có tại cửa hàng. Màn hình này sẽ gọi đến VM RequestAPI để lấy thông tin về danh sách sản phẩm. Package database sẽ lưu lại thông tin vào data local của thiết bị nhằm mục đích sử dụng cho trường hợp không có internet.

Mỗi một phiên làm việc hoặc một lần reload dữ liệu từ người dùng, dữ liệu lưu trữ trong local sẽ ghi lại những thay đổi mới nhất thông qua hàm update. Vì thế người dùng hoàn toàn có thể sử dụng những thông tin này để tư vấn kịp thời trong quãng thời gian ngắn nếu mất internet.



**Hình 4.10** Các package tham gia vào nhiệm vụ hiển thị tình hình doanhh thu

Ở màn hình hiển thị doanh thu theo từng cửa hàng ViewModel RequestAPI tiếp tục được gọi để lấy dữ liệu trả về màn hình chính. Tại đây ViewModel ChartViewModel sẽ lấy dữ liệu đã được bóc tách để thực hiện nhiệm vụ vẽ biểu đồ của mình.

## Thiết kế chi tiết

### Thiết kế giao diện

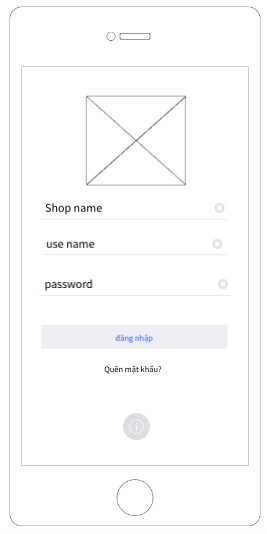
Giao diện client được thiết kế để chạy tốt nhất trên các thiết bị iphone 6, 7, 8 với các độ phân giải màn hình như 750 x 1336 pixels và 1080 x 1920 pixels. Màn hình hiển thị với kích thước 4.7 inch và 5.5 inch cho phép hiển thị đầy đủ màu sắc.

Về các chuẩn thống nhất được mô tả như bảng sau:

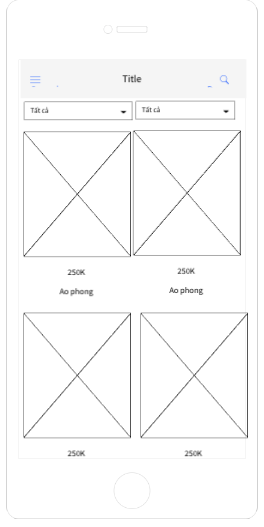
**Bảng 4.1** Quy chuẩn thiết kế giao diện ứng dụng

|  |  |  |
| --- | --- | --- |
| STT | Tên thành phần | Mô tả |
| 1 | Màu sắc | Ứng chọn chọn phông màu xanh với mã màu **#006799** làm màu chủ đạo của ứng dụng, thống nhất màu sắc hiển thị giữa các màn hình |
| 2 | Phông chữ | Quy định phông chữ 14 cho các nôi dung dạng text bình thường Phông chữ 17 cho các tiêu đề  Phông chứ 12 cho các phần chú thích |
| 3 | Vị trí hiển thị thông báo | Đối với các thông báo, ứng dụng dùng stype chuẩn của Apple, hiển thị ở vị trí chính giữa màn hình thiết bị |
| 4 | Một số quy định thiết kế khác | Đối với các nút bấm trên màn hình phải được bo tròn và quy định chung một loại phông chữ hiển thị cũng như hiệu ứng khi tương tác |

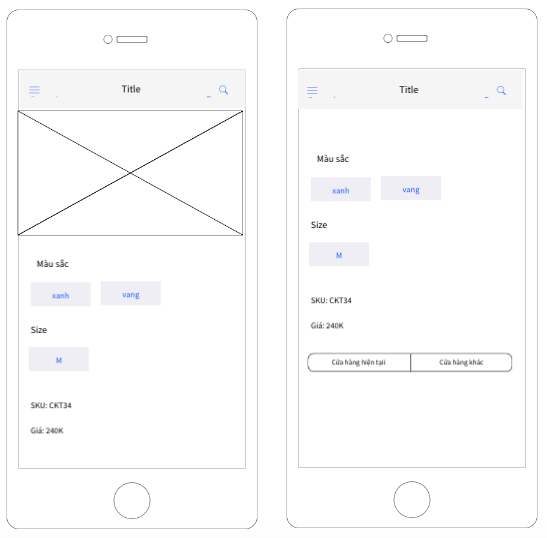
Một số hình ảnh thiết kế minh hoạ cho chức năng của sản phẩm



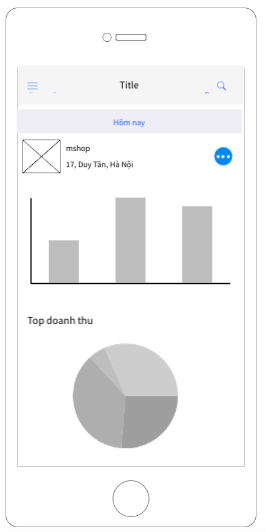
**Hình 4.11** Minh hoạ các thành phần của Login



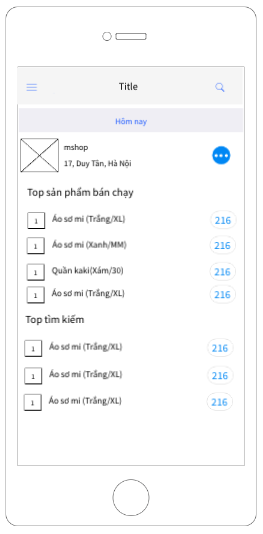
**Hình 4.12** Minh hoạ cho chức năng tra cứu



Hình . Minh hoạ cho chức năng xem chi tiết sản phẩm



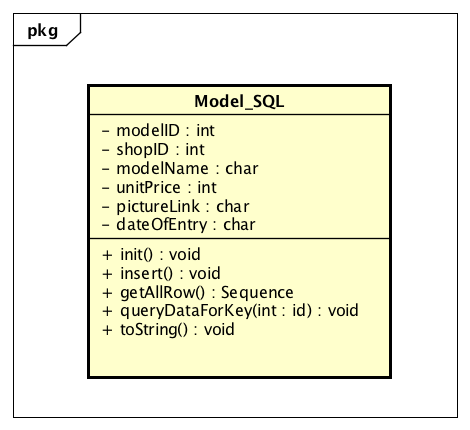
**Hình 4.14** Minh hoạ cho chức năng quản lí doanh thu



**Hình 4.15** Minh hoạ cho chức năng quản lí sản phẩm

### Thiết kế lớp

Thiết kế chi tiết cho một số lớp quan trọng của project.

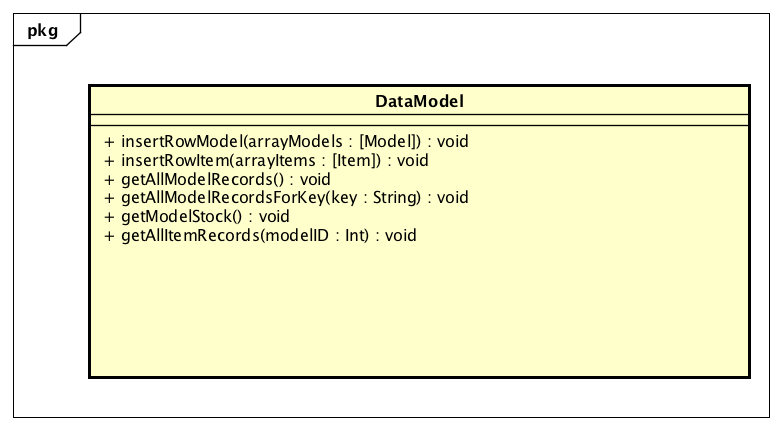


**Hình 4.16** Thiết kế lớp Model\_SQL

Class Model\_SQL có nhiệm vụ tạo bảng Model lưu trữ những mẫu sản phẩm kinh doanh của cửa hàng.

**Bảng 4. 2** Giải thích phương thức trong class Model\_SQL

|  |  |
| --- | --- |
| **Tên phương thức** | **Nhiệm vụ** |
| Init() | Khởi tạo bảng |
| Insert() | Thêm bản ghi vào bảng |
| getAllRow() | Lấy ra thông tin các bản ghi trong bảng |
| getDataForKey() | Lấy dữ liệu theo điều kiện nhập |
| toString | Chuyển dữ liệu trong bảng thành chuỗi kí tự |

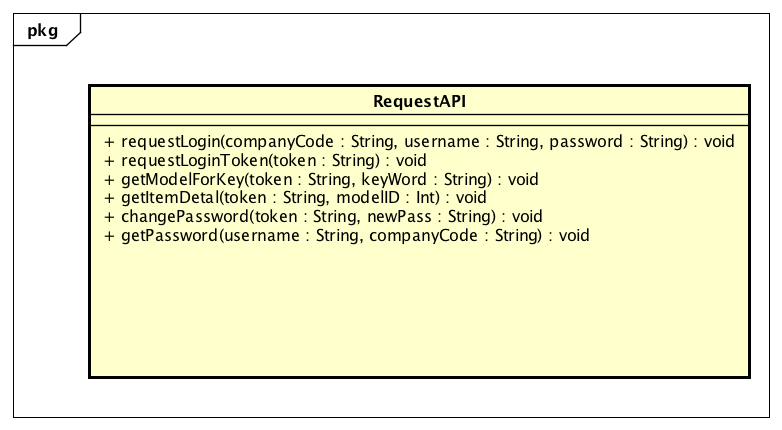


**Hình 4.17** Thiết kế lớp DataModel

Class DataModel là một class quan trọng trong project. Class này có nhiệm vụ thực hiện các hàm liên quan đến cơ sở dữ liệu local.

Bảng 4. giải thích các phương thức trong class DataModel

|  |  |
| --- | --- |
| **Tên phương thức** | **Nhiệm vụ** |
| insertRowModel | Lưu 1 bản ghi model vào bảng Model |
| insertRowItem | Lưu 1 bản ghi item vào bảng Item |
| getAllModelRecord | Lấy mảng các model đã lưu |
| getAllModelRecordForKey | Lấy mảng model đã lưu theo điều kiện |
| getModelStock | Lấy mảng model trong kho |
| getAllItemRecord | Lấy mảng sản phẩm đã lưu |



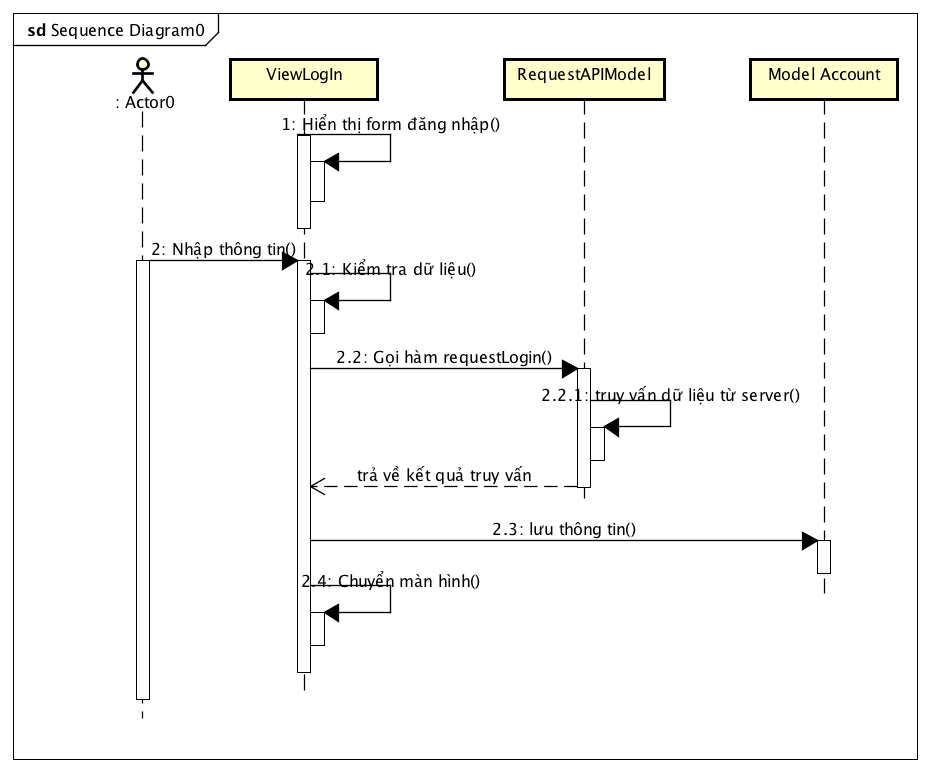
**Hình 4.18** Thiết kế chi tiết cho class RequestAPI

Class RequestAPI bao gồm tất cả các phương thức để giao tiếp và lấy dữ liệu tử server để phục vụ cho các màn hình khác nhau trong chương trình.

**Bảng 4. 4** Giải thích các phương thức trong class RequestAPI

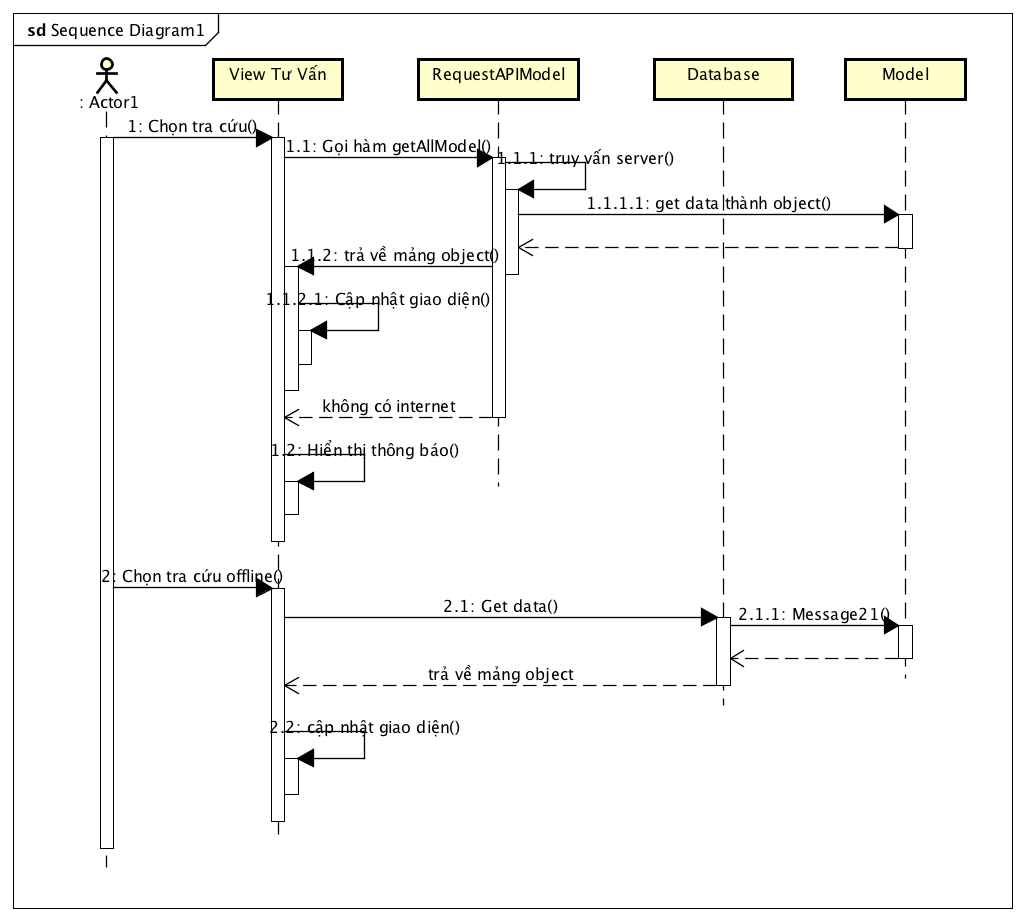
|  |  |
| --- | --- |
| **Tên phương thức** | **Nhiệm vụ** |
| requestLogin() | Yêu cầu đăng nhập vào ứng dung |
| requestLoginToken | Yêu cầu đăng nhập vào ứng dụng với token của phiên làm việc. |
| getModelForKey | Lấy về model từ server theo điều kiện |
| getItemData | Lấy về các item của cửa hàng |
| changePassword | Yêu cầu đổi mật khẩu tài khoản |
| getPassword | Lấy lại mật khẩu |

Minh hoạ vai trò của các class thông qua các biểu đồ tuần tự sau.



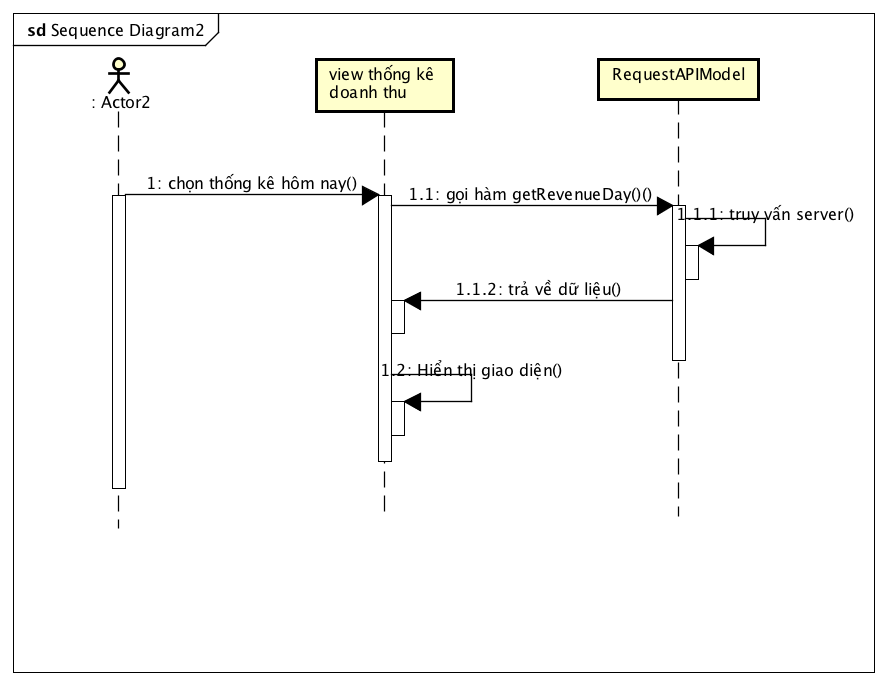
**Hình 4.19** Biểu đồ tuần tự cho chức năng đăng nhập

Hình 4.20 mô tả chức năng hiển thị danh sách các model trong cửa hàng. Tuỳ vào lựa chọn của nhân viên tra cứu theo tiêu chí gì ứng dụng sẽ hiển thị danh sách tương ứng.



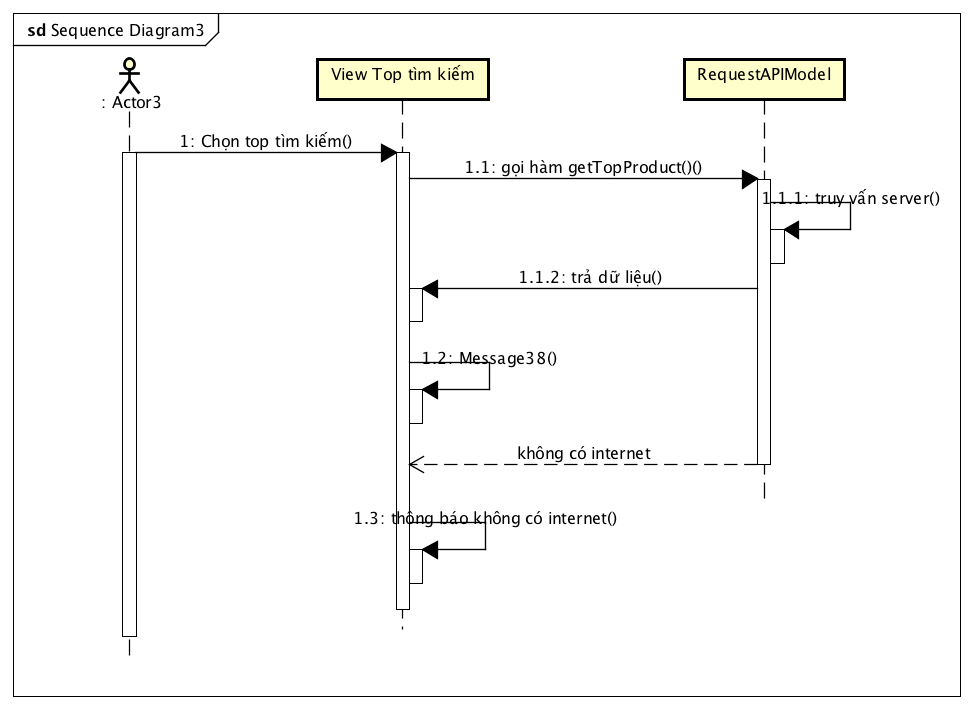
**Hình 4.20** Biểu đồ trình tự cho chức năng tra cứu theo tên

Hình 4.21 mô tả trình tự cho chức năng thống kê doanh thu của cửa hàng. Tại màn hình thống kê, chủ cửa sẽ có nhiều lựa chọn thống kê theo ngày, tuần hay tháng. Ngoài ra ứng dụng còn hỗ trợ cho phép hiển thị thống kê doanh thu theo từng chi nhánh trong toàn bộ cửa hàng. Với chức năng này sẽ giúp chủ cửa hàng có những đánh giá chính xác nhất về tình hình kinh doanh của mỗi chi nhánh, qua đó sẽ có những chiến lược kinh doanh và nhập hàng phù hợp.



**Hình 4.21** Biều đồ tuần tự thống kê doanh thu

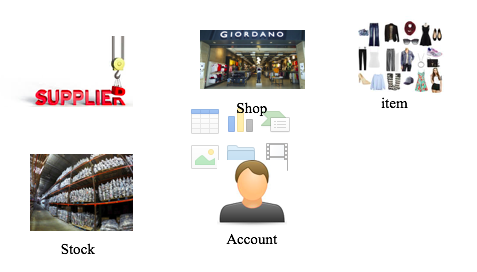
Hình 4.22 mô tả trình tự cho chức năng hiển thị top sản phẩm tìm kiếm nhiều nhất tại cửa hàng. Nắm bắt được xu hướng tìm kiếm sản phẩm của khách hàng là một trong những yếu tố cần thiết nhất để chủ cửa hàng có thể đưa ra được những chiến lược kinh doanh hợp lý. Từ những con số thống kê cho từng cửa hàng, chủ cửa hàng cũng dễ dàng hơn trong việc cung cấp các mẫu sản phẩm cho từng cửa hàng đó. Không chỉ có thế, chỉ số thống kê còn là nguồn tham khảo để chủ cửa hàng nắm bắt được nhu cầu của khách hàng, xu hướng thời trang theo mùa để kịp thời bổ sung vào những thời gian sau.



**Hình 4.22** Biểu đồ tuần tự cho chức năng hiển thị top tìm kiếm sản phẩm

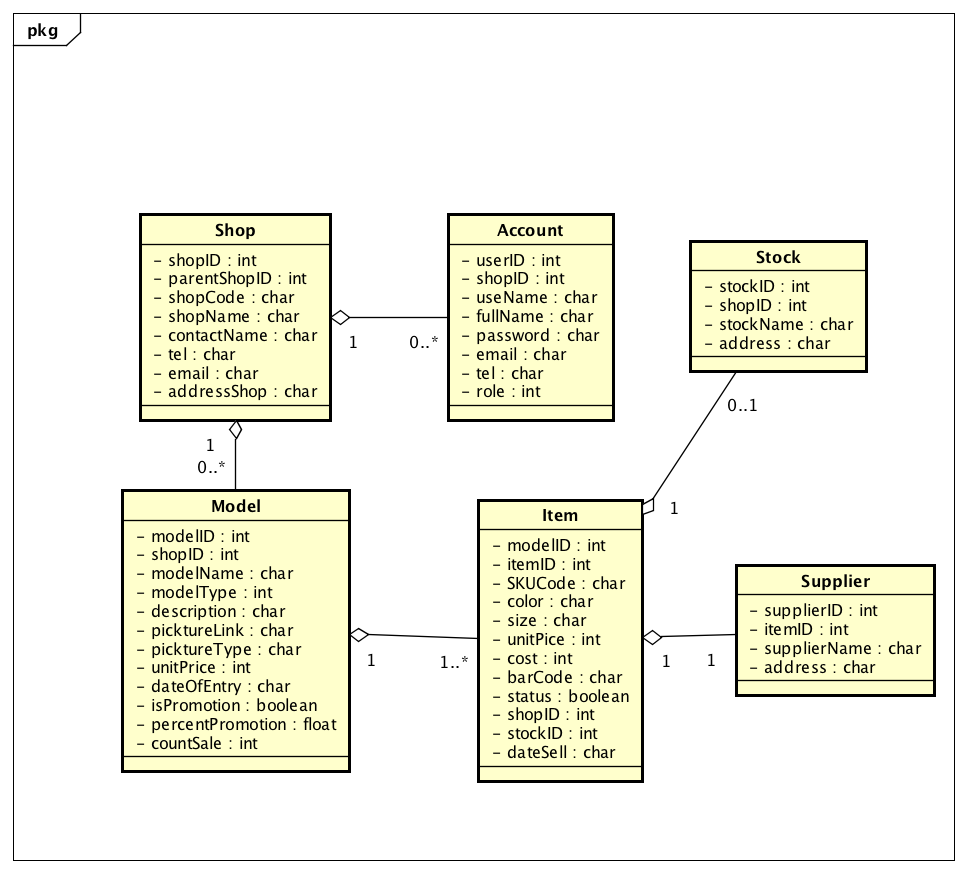
### Thiết kế cơ sở dữ liệu

Phân tích bài toán tra cứu tư vấn và quản lý doanh thu cửa hàng như sau:



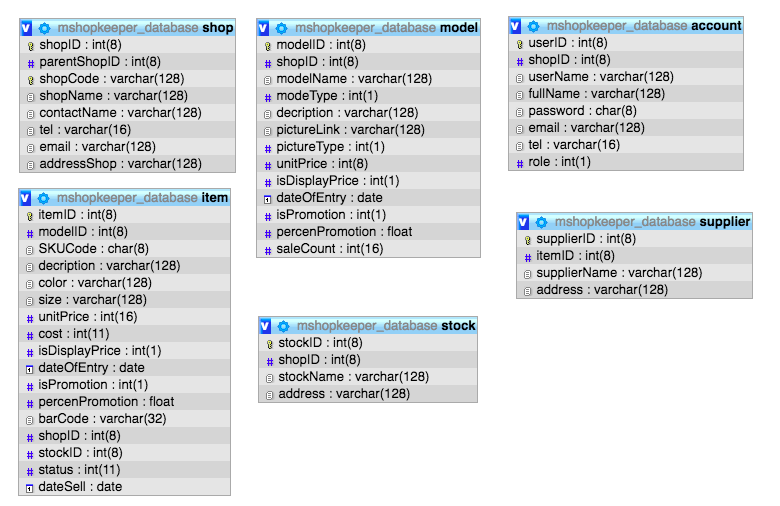
**Hình 4.23** Các thực thể liên kết

Hình 4.24 dưới đây mô tả các thuộc tính của thực thể và mối quan hệ liên kết giữa các thực thể trong bài toán.



**Hình 4.24** Mô hình thực thể liên kết

Hinh 4.25 mô tả thiết kế các bảng cơ sở dữ liệu



**Hình 4.25** Các bảng cơ sở dữ liệu

Đặc tả các bảng cơ sở dữ liệu

**Bảng 4. 5** Đặc tả thông tin bảng Shop

|  |  |  |
| --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Giải thích |
| shopID | Int | Mã của cửa hàng |
| parentShopID | Int | Mã của cửa hàng cha nếu cửa hàng đó là chi nhánh của một cửa hàng khác |
| shopCode | Char | Mã code của cửa hàng |
| shopName | Char | Tên của cửa hàng |
| contactName | Char | Tên liên hệ của cửa hàng, thường dùng nếu đó là cửa hàng chi nhánh |
| Tel | Char | Số điện thoại liên hệ của cửa hàng |
| Email | Char | Email của cửa hàng |
| Address | Char | Địa chỉ cửa hàng |

**Bảng 4. 6** đặc tả thông tin bảng Model

|  |  |  |
| --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Giải thích |
| modelID | Int | Mã của mẫu sản phẩm |
| shopID | Int | Mã của cửa hàng chứa mẫu sản phẩm |
| modelName | Char | Tên của mẫu sản phẩm |
| modelType | Char | Loại mẫu sản phẩm, sản phẩm đơn hoặc combo. |
| Desciption | Char | Thông tin mô tả |
| picktureLink | Char | Link ảnh đại diện của mẫu |
| picktureType | Int | Loại ảnh đại diện, do hệ thống tự sinh hoặc từ nguồn khác |
| unitPrice | Int | Giá bán chung của sản phẩm |
| isDisplayPrice | Int | Có cho phép hiển thị giá cho mẫu đó không |
| dateOfEntry | Date | Ngày nhập hàng |
| isPromotion | Int | Sản phẩm có khuyến mại hay không |
| percentPromotion | Float | Phần trăm khuyến mại |
| saleCount | Int | Số lượng sản phẩm đã bán trong mẫu |

**Bảng 4. 7** Đặc tả thông tin bảng Item

|  |  |  |
| --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Giải thích |
| itemID | Int | Mã ID sản phẩm |
| modelID | Int | Mã ID của mẫu sản phẩm |
| SKUCode | Int | Mã SKU sản phẩm |
| Description | Char | Thông tin mô tả sản phẩm |
| Color | Char | Màu sản phẩm |
| Size | Char | Kích thước sản phẩm |
| unitPrice | Int | Giá bán sản phẩm |
| Cost | Int | Giá gốc sản phẩm |
| isDisplayPrice | int | Có cho phép hiển thị giá không |
| dateSell | Date | Ngày bán |
| Status | Int | Trạng thái sản phẩm, bán hay chưa bán |
| barCode | Char | Mã barCode của sản phẩm |
| shopID | Int | Mã ID cửa hàng |
| stockID | Int | Mã ID kho hàng |

## Xây dựng ứng dụng

### Thư viện và công cụ sử dụng

Trong quá trình thực hiện đề tài của mình, em có tham khảo và sử dụng một số công cụ lập trình và các thư viện hỗ trợ như sau:

**Bảng 4.8** Danh sách thư viện và công cụ sử dụng

|  |  |  |
| --- | --- | --- |
| **Mục đích** | **Công cụ** | **Địa chỉ URL** |
| IDE lập trình | Xcode | <http://www.eclipse.org/> |
|  | Atom | https://atom.io/ |
| Thư viện | Alamofire | https://github.com/Alamofire/Alamofire |
|  | Kingfisher | https://github.com/onevcat/Kingfisher |
|  | SlideMenuSwift | https://github.com/dekatotoro/SlideMenuControllerSwift |
|  | Express | https://expressjs.com/ |
|  | JWT | https://github.com/auth0/node-jsonwebtoken |

### Kết quả đạt được

Sản phẩm sau khi hoàn thành bao gồm các thành phần sau:

* File .zip của project Mshopkeeper tư vấn: đây là file nén của ứng dụng tư vấn dành cho nhân viên cửa hàng.
* File .zip của project Mshopkeeper thống kê: đây là file nén của ứng dụng quản lý dành cho chủ cửa hàng.
* File .zip của project server và API: đây là file nén toàn bộ project của phần server

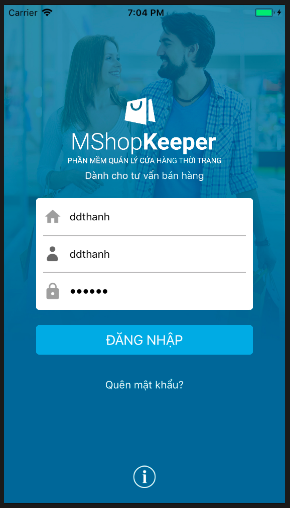
Thống kê khác

**Bảng 4. 9** Thống kê project

|  |  |  |
| --- | --- | --- |
| Stt | Tên thành phần | Kích thước |
| 1 | Tổng số dòng code |  |
| 2 | Số lớp |  |
| 3 | Số gói |  |
| 4 | Dung lượng app mshopkeeper tư vấn |  |
| 5 | Dung lượng app mshopkeeper quản lý |  |
| 6 | Dung lượng project server |  |

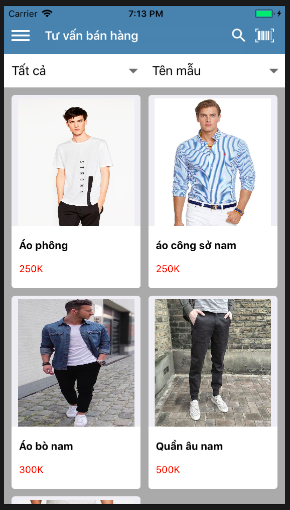
### Minh hoạ các chức năng chính

Sản phẩm được chia ra làm 2 ứng dụng nhỏ cho chủ cửa hàng và nhân viên bán hàng sử dụng. Dưới đây là minh hoạ cho một số chức năng chính.



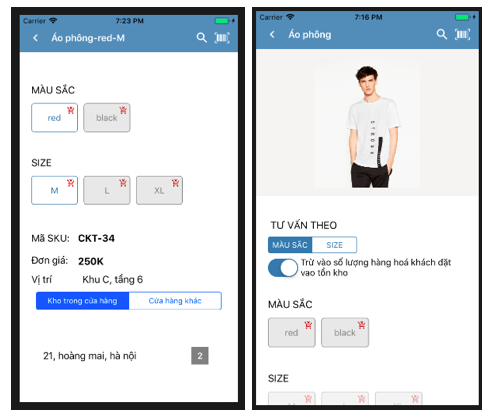
**Hình 4.26** Chức năng đăng nhập

Với màn hình đăng nhập, người dùng có thể dùng tải khoản của mình để đăng nhập và sử dụng. Nếu người dùng quên mật khẩu, có thể chọn chức năng lấy lại mật khẩu.



**Hình 4.27** Chức năng xem sản phẩm

Tại màn hình này có nhiều lựa chọn cho nhân viên bán hàng. Nhân viên có thể tra cứu danh sách các sản phẩm mới nhập, sản phẩm bán chạy, hay sản phẩm khuyến mại. Ngoài ra ứng dụng còn hỗ trợ tra cứu nhanh thông qua tên cho người dùng nhập.



**Hình 4.28** Chức năng xem chi tiết sản phẩm

Ở màn hình này, nhân viên cửa hàng sẽ biết được chính xác sản phẩm đang tra cứu còn những màu và kích cỡ nào. Ngoài ra ứng dụng còn cho biết được đơn giá, số lượng sản phẩm tại các chi nhánh cửa hàng.

## Kiểm thử

Kiểm thử sản phẩm là một trong những khâu quan trọng trong việc đảm bảo chất lượng dự án. Dưới đây em xin trình bày các bảng kiểm thử cho một vài chức năng quan trọng

**Bảng 4. 10** Kiểm thử cho chức năng đăng nhập

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Trạng thái | Hành động | Kết quả mong đợi | Kết quả test |
| 1 | Có thông tin useName, password, shopCode | Click vào button đăng nhập | Chuyển sang màn hình tiếp theo | Pass |
| 2 | Có thông tin usename, shopCode, thiếu trường pass | Click vào button đăng nhập | Con trỏ chỉ vào trường password | Pass |
| 3 | Thiếu thông tin trường shopCode | Click vào đăng nhập | Con trỏ chỉ vào trường shopCode | Pass |
| 4 | Sai thông tin usename | Click vào đăng nhập | Có thông báo | Pass |

**Bảng 4. 11** Kiểm thử chức năng đổi mật khẩu

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Trạng thái | Hành động | Kết quả mong đợi | Kết quả test |
| 1 | Chưa nhập mật khẩu cũ | Click vào button đổi mật khâu | Con trỏ chi vào trường nhập mật khẩu cũ, có thông báo | Pass |
| 2 | Nhập mật khầu mới nhỏ hơn 6 kí tự | Click vào button đổi mật khẩu | Có thông báo mật khẩu phải lớn hơn 6 kí tự | Pass |
| 3 | Thiếu trường nhập lại password | Click vào button đổi mật khẩu | Con trỏ chỉ vào trường nhập lại password | Pass |
| 4 | Password mới nhập lại không đúng | Click vào đăng nhập | Thông báo nhập password chưa đúng | Pass |
| 5 | Nhập đầy đủ và đúng thông tin | Click vào đăng nhập | Thông báo đôi mật khẩu thành công | Pass |

**Bảng 4. 12** Kiểm thử chức năng chọn hiển thị hàng hoá theo tên

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Trạng thái | Hành động | Kết quả mong đợi | Kết quả test |
| 1 | Không có internet | Chọn hiển thị theo tên | Hiện thị thông báo mất internet | Pass |
| 2 | Không có internet, chọn chức năng offline | Chọn hiển thị theo tên | Màn hình hiển thị danh sách hàng hoá | Pass |
| 3 | Có internet | Chọn hiển thị theo tên | Màn hình hiển thị được danh sách hàng hoá | Pass |

**Bảng 4. 13** Kiểm thử chức năng hiển thị chi tiết sản phẩm

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Trạng thái | Hành động | Kết quả mong đợi | Kết quả test |
| 1 | Không có internet | Chọn sản phẩm | Chuyển sang màn hình chi tiết | Pass |
| 2 | Có internet, sản phẩm có số liệu | Chọn sản phẩm | Chuyển sang màn hình chi tiết, hiển thị đủ size và color sản phẩm | Pass |
| 3 | Có internet, sản phẩm có số liệu | Chọn sản phẩm | Chuyển sang màn hình chi tiết, hiển thị kết quả tra cứu không còn hàng | Pass |
| 4 | Chọn size, không chọn color | Chọn sản phâm | Không hiển thị ra giá, mã SKU, thông tin hàng tại các chi nhanh | Pass |
| 5 | Chọn size, color | Chọn sản phẩm | Hiển thị đầy đủ thông tin về sản phẩm | Pass |

## Triển khai

Sinh viên trình bày mô hình và/hoặc cách thức triển khai thử nghiệm/thực tế. Ứng dụng của sinh viên được triển khai trên server/thiết bị gì, cấu hình như thế nào. Kết quả triển khai thử nghiệm nếu có (số lượng người dùng, số lượng truy cập, thời gian phản hồi, phản hồi người dùng, khả năng chịu tải, các thống kê, v.v.)

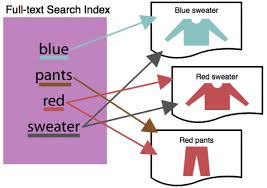
# Các giải pháp và đóng góp nổi bật

## Tăng hiệu năng ứng dụng bằng cách tối ưu hoá truy vấn.

Trong quá trình thiết kế và xây dựng cơ sở dữ liệu cho ứng dụng, một trong những vấn đề quan trọng nhất là đảm bảo tốc độ truy vấn dữ liệu. Cơ sở dữ liệu của ứng dụng phải đáp ứng cho nhiều cửa hàng khác nhau, mỗi cửa hàng lại có rất nhiều sản phẩm khác nhau. Chính vì thế số lượng bản ghi trong cơ sở dữ liệu là rất lớn và ngày càng được mở rộng hơn. Khi số lượng cửa hàng đăng ký tăng lên cũng là lúc bài toán tối ưu truy vấn được đặt ra. Một ứng dụng có phải là sự lựa chọn tốt nhất cho người dùng không thì ngoài việc phải đáp ứng những chức năng thiết yếu còn phải đáp ứng hiệu năng và trải nghiệm người dùng. Sẽ thật là tồi tệ nếu như bắt người dùng phải chờ đợi. Theo số liệu thống kê hiện nay, thời gian trung bình người dùng có thể chấp nhận được đối với 1 thao tác là không quá 3s. Nếu không xử lý tốt vấn đề truy vấn thì 1 request dữ liệu từ cơ sở mà em đã thiết kế sẽ không thể đáp ứng được nhu cầu của khách hàng.

Từ bản chất của ứng dụng chủ yếu là truy vấn và lấy dữ liệu từ server trả về để hiển thị **theo** mục đích của người dùng do đó em lựa chọn giải pháp đặt index cho một số trường trong cơ sở dữ liệu.

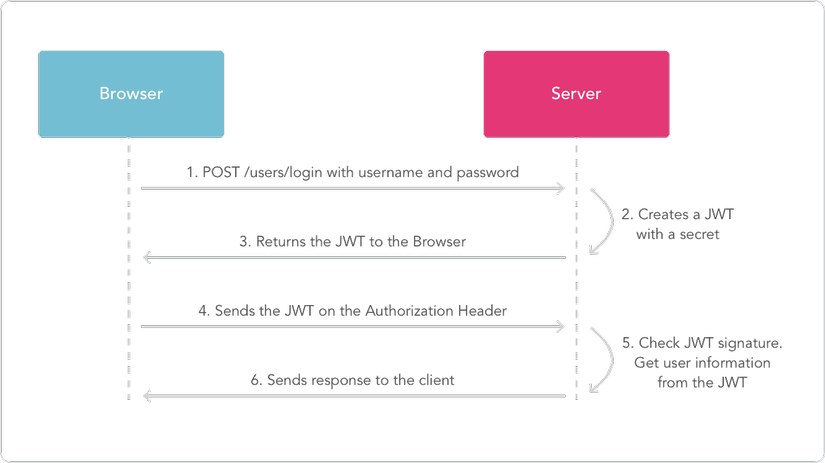
Index trong các cơ sở dữ liệu thực chất là cũng là 1 loại bảng mà giữ khoá chính hoặc trường index và một loại con trỏ đến từng bản ghi vào 1 bảng thực tế. Có thể coi index giống như một bảng tra cứu phụ lục trong các cuốn sách. Người sử dụng không thể nhìn thấy các index, chúng chỉ được sử dụng để tăng tốc độ truy vấn và sẽ được sử dụng bởi cơ sở dữ liệu tìm kiếm để xác định vị trí bản ghi rất nhanh. Hình dưới đây mô tả khái quát chức năng của index trong các cơ sở dữ liệu.



**Hình 5.1** Mô tả chức năng của index trong cơ sở dữ liệu

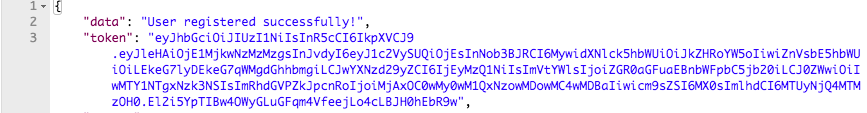
## Đảm bảo an toàn thông tin dữ liệu

Bảo mật là một trong những yêu cầu quan trọng đối với các ứng dụng di động nói riêng và các sản phẩm phần mềm nói chung. Các dữ liệu có thể coi như một tài sản riêng của từng cửa hàng, từng cá nhân. Vì thế việc bảo mật dữ liệu là điều cần thiết. Một ứng dụng tốt bắt buộc phải đáp ứng những yêu cầu về bảo mật dữ liệu người dùng. Trong đề tài của mình, vấn đề bảo mật cũng được em chú ý đến khi thiết kế và xây dựng chương trình.  
Do sử dụng phương thức truyền nhận dữ liệu từ server tới client qua các API nên em đã lựa chọn phương thức bảo mật sử dụng thư viện JSON WEB TOKEN (JWT). JWT được biết đến rộng rãi như một giải pháp hỗ trợ bảo mật tốt nhất cho API. Cùng tìm hiểu qua về cách thức hoạt động của JWT để thấy được những ưu nhược điểm của nó.  
Như chúng ta đã biết trong API thì Client giao tiếp và gửi request (yêu cầu) đến Server thông qua URL trên giao thức HTTP. Hình ảnh dưới đây minh hoạ cho cách thức hoạt động của API.



**Hình 5.2** Cách thức hoạt động của JWT

Giải thích hoạt động:   
Bước 1: Khi người dùng gửi username và password lên server sẽ truy vấn và trả về thông tin ứng với các tham số đó.   
Bước 2, 3: Server sẽ tạo ra một token tương ứng với thông tin request trả về (hình )



**Hình 5.3** Token trả về của phương thức đăng nhập

Bước 4: Với những lần truy vấn sau, client chỉ cần gửi token kèm theo phần header để truy vấn dữ liệu mà không cần gửi lại thông tin cá nhân khác. Bước 5: Server sau khi bóc tách token sẽ xác thực được người dùng để lấy đúng thông tin yêu cầu thông qua hàm verify().  
Bước 6: Server gửi lại thông tin cho client.  
  
Việc gửi nhận yêu cầu dữ liệu thông qua xác thực token sẽ giúp đảm bảo an toàn hơn cho dữ liệu người dùng, tránh tối đa việc thất thoát trong quá trình truyền tin. Ngoài ra việc gửi token kèm theo header sẽ giảm số câu truy vấn lặp lại trong database, từ đó nâng cao được thời gian xử lý các yêu cầu.  
  
Thông thường mỗi 1 token sẽ có một khoảng thời hạn nhất định. Trong ứng dụng của mình em đang cấu hình khoảng thời gian cho 1 phiên làm việc là 24h. Sau khi token hết hiệu lực, người dùng phải đăng nhập lại ứng dụng để bắt đầu một phiên làm việc mới.

# Kết luận và hướng phát triển

## Kết luận

Tổng kết sau quá trình thực hiện đề tài và đánh giá chung với một số ứng dụng cùng loại trên thị trường, em rút ra một số kết luận sau:

* Ứng dụng tra cứu tư vấn và quản lý cửa hàng Mshopkeeper đã đáp ứng tốt các yêu cầu đề ra của bài toán.
* Ứng dụng tuy không được tích hợp tính năng bán hàng như một vài sản phẩm cùng loại trên thị trường xong cũng phần nào mang lại lợi ích thực sự cho các hộ kinh doanh và cửa hàng.
* Ứng dụng có khả năng triển khai với nhiều tập người dùng, đáp ứng tốt các trải nghiệm cơ bản.
* Với mục tiêu và đối tượng ban đầu hướng đến là các hộ kinh doanh vừa và nhỏ, ứng dụng này thực sự có tiềm năng phát triển.
* Hiện tại ứng dụng còn một vài điểm hạn chế, chưa tích hợp được tính năng hỗ trợ bán hàng, đặt hàng. Ứng dụng còn phải cải tiến nhiều hơn về mặt hiệu năng nếu tập người dùng lớn.

Qua quá trình tìm hiểu và thực hiện đề tài em cũng phần nào rút ra cho mình những bài học từ việc áp dụng các lý thuyết cho đến quá trình tạo ra một sản phẩm thực tế hữu ích với người dùng.

## Hướng phát triển

Để sản phẩm có thể hoàn thiện và tồn tại được trong tương lai, em đã có một số định hướng nhằm cải thiện và phát triển ứng dụng như sau.

Trước hết cần hoàn thiện hơn một số yêu cầu quản lý, tập trung thống kê chi tiết hơn thói quen của người dùng.

Đánh giá được các thông số khác như tỉ lệ mua hàng của người dùng theo từng giờ trong ngày, các khung giờ cao điểm.

Trong tương lai sản phẩm cần tích hợp và có thể mở rộng đối tượng sử dụng đến cả khách mua hàng chứ không chỉ dừng lại cho nhân viên và chủ cửa hàng. Khi đó cần thêm cả tính năng hỗ trợ tự động tư vấn dựa vào thông tin mà khách hàng nhập.

# Tài liệu tham khảo

1. Hovy E. H., Automated Discourse Generation Using Discourse Structure Relations, Artificial Intelligence, Elsevier Science Publishers, 63: 341-385, 1993.

\* **Sách**: Tên tác giả, tên sách, volume (nếu có), lần tái bản (nếu có), nhà xuất bản, năm xuất bản.

1. Peterson L. L. and Davie B. S. , Computer Networks: A Systems Approach, 2nd ed., Mogran-Kaufmann, 1999.
2. Nguyễn Thúc Hải, Mạng máy tính và các hệ thống mở, Nhà xuất bản giáo dục, 1999.

\* **Tập san Báo cáo Hội nghị Khoa học**: Tên tác giả, tên báo cáo, tên hội nghị, ngày (nếu có), địa điểm hội nghị, năm xuất bản.

1. Poesio M. and Di Eugenio B., Discourse Structure and Anaphoric Accessibility, In Proc. of the ESSLLI Workshop on Information Structure, Discourse Structure and Discourse Semantics, Helsinki, 2001.

\* **Đồ án tốt nghiệp, Luận văn Thạc sĩ, Tiến sĩ**: Tên tác giả, tên đồ án/luận văn, loại đồ án/luận văn, tên trường, địa điểm, năm xuất bản.

1. Knott D., A Data-Driven Methodology for Motivating a Set of Coherence Relations, Ph.D. Thesis, University of Edinburgh, UK, 1996.

\* **Tài liệu tham khảo từ Internet** : Tên tác giả (nếu có), tựa đề, cơ quan (nếu có), địa chỉ trang web, thời gian lần cuối truy cập trang web.

1. Berners-Lee T., Hypertext Transfer Protocol (HTTP), CERN, ftp:/info.cern.ch/pub/www/doc/http-spec.txt.Z, last visited May 2010.
2. Princeton University, WordNet, http://www.cogsci.princeton.edu/~wn/index.shtml, last visited May 2010.

1. Mô-hình-hướng-dịch-vụ: service-oriented architectures (SOA) [↑](#footnote-ref-1)
2. Component: chỉ một thành phần cấu thành nên một đối tượng nào đó. [↑](#footnote-ref-2)
3. dịch vụ đăng ký : Service Registry [↑](#footnote-ref-3)
4. https://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html [↑](#footnote-ref-4)