# MỤC LỤC

[MỤC LỤC 1](#_Toc110114482)

[CHƯƠNG 1: TỔNG QUAN 1](#_Toc110114483)

[1.1 Giới thiệu tổng quan 1](#_Toc110114484)

[1.2 Nghiên cứu hiện trạng và giải quyết vấn đề 1](#_Toc110114485)

[1.2.1 Hiện trạng cơ cấu tổ chức 1](#_Toc110114486)

[1.2.2 Cơ cấu tổ chức 2](#_Toc110114487)

[1.2.3 Mô tả hoạt động của các bộ phận 2](#_Toc110114488)

[1.2.4 Xác định yêu cầu 2](#_Toc110114489)

[1.3 Giới thiệu về công nghệ, ngôn ngữ sử dụng 3](#_Toc110114490)

[1.3.1 Giới thiệu về ngôn ngữ Java 3](#_Toc110114491)

[1.3.1.1 Khái niệm 3](#_Toc110114492)

[1.3.1.2 Ứng dụng của Java 4](#_Toc110114493)

[1.3.1.3 Những đặc điểm cơ bản của Java 4](#_Toc110114494)

[1.3.1.4 Các platform cơ bản của Java 5](#_Toc110114495)

[1.3.1.5 Tiêu chuẩn của một môi trường Java điển hình 6](#_Toc110114496)

[1.3.2 Giới thiệu về Spring Framework 8](#_Toc110114497)

[1.3.2.1 Giới thiệu Spring Framework 8](#_Toc110114498)

[1.3.2.2 Lịch sử ra đời của Spring Framework 9](#_Toc110114499)

[1.3.2.3 Tính năng của Spring Framework 10](#_Toc110114500)

[1.3.2.4 Các modun của Spring Framework 13](#_Toc110114501)

[1.3.3 Giới thiệu về Spring Boot 17](#_Toc110114502)

[1.2.3.1 Giới thiệu Spring Boot 17](#_Toc110114503)

[1.2.3.2 Các tính năng nổi bật 18](#_Toc110114504)

[1.2.3.3 Tạo một Project Spring Boot với Intelij Idea 18](#_Toc110114505)

[CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG 27](#_Toc110114506)

[2.1 Phân tích biểu đồ usecase 27](#_Toc110114507)

[2.1.1 Xác định các tác nhân 27](#_Toc110114508)

[2.1.2 Xác định usecase 27](#_Toc110114509)

[2.1.3 Biểu đồ usecase 28](#_Toc110114510)

[2.1.3.1 Usecase khách hàng 28](#_Toc110114511)

[2.1.3.2 Usecase trưởng cửa hàng 29](#_Toc110114512)

[2.1.3.3 Usecase nhân viên tư vấn 30](#_Toc110114513)

[2.2 Đặc tả usecase 30](#_Toc110114514)

[2.2.1 Xem trang chủ 30](#_Toc110114515)

[2.2.2 Xem danh sách sản phẩm 31](#_Toc110114516)

[2.2.3 Đăng kí lịch hẹn 32](#_Toc110114517)

[2.2.4 Thêm mới khách hàng 33](#_Toc110114518)

[2.2.5 Cập nhật khách hàng 34](#_Toc110114519)

[2.2.6 Thêm mới sản phẩm 35](#_Toc110114520)

[2.2.7 Cập nhật sản phẩm 36](#_Toc110114521)

[2.2.8 Thêm mới giao dịch 36](#_Toc110114522)

[2.2.9 Cập nhật giao dịch 37](#_Toc110114523)

[2.2.10 Thêm mới người dùng 38](#_Toc110114524)

[2.2.11 Cập nhật người dùng 39](#_Toc110114525)

[2.3 Biểu đồ lớp 41](#_Toc110114526)

[2.4 Cở sở dữ liệu 41](#_Toc110114527)

[2.4.1 Mô hình cở sở dữ liệu 42](#_Toc110114528)

[2.4.2 Chi tiết bảng dữ liệu 42](#_Toc110114529)

[2.4.2.1 Bảng sản phẩm (product) 42](#_Toc110114530)

[2.4.2.2 Bảng hình ảnh sản phẩm (product\_img) 43](#_Toc110114531)

[2.4.2.3 Bảng lịch hẹn (request\_txn) 43](#_Toc110114532)

[2.4.2.4 Bảng khách hàng (customer) 44](#_Toc110114533)

[2.4.2.5 Bảng giao dịch (payment\_txn) 44](#_Toc110114534)

[2.4.2.6 Bảng chi tiết giao dịch (payment\_txn\_detail) 45](#_Toc110114535)

[2.4.2.7 Bảng người dùng (user) 46](#_Toc110114536)

[2.4.2.8 Bảng nhóm quyền (group\_role) 47](#_Toc110114537)

[CHƯƠNG 3: CÀI ĐẶT VÀ TRIỂN KHAI 48](#_Toc110114538)

[3.1 Giới thiệu về công cụ và môi trường cài đặt 48](#_Toc110114539)

[3.2 Giao diện người dùng 48](#_Toc110114540)

[3.2.1 Màn hình trang chủ 48](#_Toc110114541)

[3.2.2 Màn hình danh sách sản phẩm 48](#_Toc110114542)

[3.2.3 Màn hình đặt lịch hẹn 48](#_Toc110114543)

[3.3 Giao diện quản trị viên 48](#_Toc110114544)

[3.3.1 Màn hình đăng nhập 48](#_Toc110114545)

[3.3.2 Màn hình danh sách người dùng 48](#_Toc110114546)

[3.3.3 Màn hình thêm mới người dùng 48](#_Toc110114547)

[3.3.4 Màn hình cập nhật người dùng 48](#_Toc110114548)

[3.3.5 Màn hình danh sách sản phẩm 48](#_Toc110114549)

[3.3.6 Màn hình thêm mới sản phẩm 48](#_Toc110114550)

[3.3.7 Màn hình cập nhật sản phẩm 48](#_Toc110114551)

[3.3.8 Màn hình danh sách khách hàng 48](#_Toc110114552)

[3.3.9 Màn hình thêm mới khách hàng 48](#_Toc110114553)

[3.3.10 Màn hình cập nhật khách hàng 48](#_Toc110114554)

[3.3.11 Màn hình danh sách đặt lịch hẹn 48](#_Toc110114555)

[3.3.12 Màn hình danh sách giao dịch 48](#_Toc110114556)

[3.3.13 Màn hình tạo mới giao dịch 48](#_Toc110114557)

[3.3.14 Màn hình cập nhật sản phẩm 48](#_Toc110114558)

[3.3.15 Báo cáo giao dịch thành công 48](#_Toc110114559)

[KẾT LUẬN 49](#_Toc110114560)

[TÀI LIỆU THAM KHẢO 50](#_Toc110114561)

# 

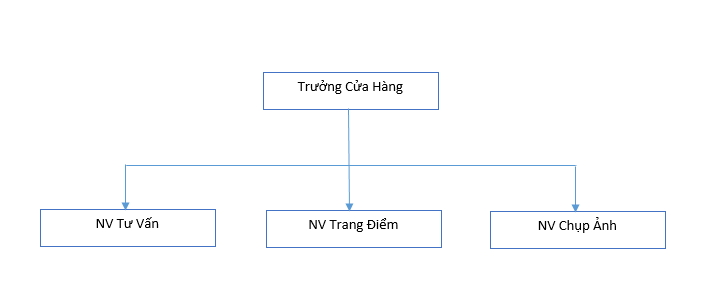
# CHƯƠNG 1: TỔNG QUAN

1. Giới thiệu tổng quan

Trong thời đại công nghệ đang phát triển bùng nổ như ngày nay, những phương pháp quản lý truyền thống đang dần được thay thế bởi các phần mền quản lý. Nhiều cửa hàng vừa và nhỏ lựa chọn phương pháp này bởi nó không chỉ giúp loại bỏ đi sự phức tạp, đem đến sự tiện lợi cho người dùng mà nó còn có thêm một loạt những lợi ích khác. Cửa hàng Áo Cưới Thủy Linh hiện nay đang có nhu cầu sử dụng phần mền quản lý để đơn giản hóa khâu quản lí và mở rộng khả năng tiếp cận tới khách hàng.

1. Nghiên cứu hiện trạng và giải quyết vấn đề
2. Hiện trạng cơ cấu tổ chức

* Cửa hàng Thủy Linh hiện đang có 2 cửa hàng kinh doanh:
* Số 10 Khu đô thị Phương Huy, Tiền Phong, Mê Linh, Hà Nội.
* Số 25 Phố Yên, Tiền Phong, Mê Linh, Hà Nội.
* Các sản phẩm, dịch vụ kinh doanh:
* Váy cưới nữ.
* Vest cưới nam.
* Dịch vụ trang điểm.
* Dịch vụ chụp ảnh cưới.
* Hình thức hoạt động: Khách hàng đến cửa hàng xem sản phẩm của cửa hàng. Nhân viên tư vấn của cửa hàng sẽ giới thiệu các sản phẩm, dịch vụ của cửa hàng cho khách hàng. Khách hàng lựa chọn sản phẩm, dịch vụ muốn sử dụng rồi báo lại cho nhân viên cửa hàng. Nhân viên cửa hàng sẽ thực hiện lưu thông tin sản phẩm, dịch vụ, thời gian mà khách hàng muốn sử dụng vào file excel để quản lý. Tiếp đó khách hàng sẽ gặp trưởng cửa hàng để đặt cọc trước 1 khoản chi phí đảm bảo với của hàng và thanh toán nốt phần chi phí dịch vụ còn lại sau khi sử dụng xong dịch vụ.

1. Cơ cấu tổ chức

* Trưởng cửa hàng: 1
* Nhân viên tư vấn: 2
* Nhân viên trang điểm: 1
* Nhân viên chụp ảnh: 1

1. Mô tả hoạt động của các bộ phận

* Trưởng cửa hàng: Quản lí công việc và nhân viên trong cửa hàng. Quản lí thu chi toàn bộ cửa hàng và thực hiện thu phí khách hàng khi họ sử dụng sản phẩm, dịch vụ của cửa hàng.
* Nhân viên tư vấn: Giới thiệu tới khách hàng các sản phẩm, dịch vụ của cửa hàng. Tư vấn cho khách hàng các sản phẩm, dịch vụ phù hợp, giải đáp thắc mắc khiếu nại của khách hàng.
* Nhân viên trang điểm: Trang điểm cho khách hàng.
* Nhân viên chụp ảnh: Chụp ảnh cho khách hàng.

1. Xác định yêu cầu

* Yêu cầu hệ thống:
* Khách hàng có thể xem thông tin các sản phẩm, dịch vụ,… của cửa hàng trên website.
* Khách hàng có thể đặt lịch hẹn trước trên website.
* Nhân viên cửa hàng có thể xem và duyệt thông tin đặt lịch hẹn của khách hàng.
* Hệ thống quản lí thông tin cá nhân, giao dịch của khách hàng
* Báo cáo thống kê doanh thu của cửa hàng theo tháng, năm…
* Yêu cầu công nghệ:
* Hệ thống hoạt động ổn định, không bị treo khi có số lượng truy cập lớn. Dễ dàng bảo trì khi gặp sự cố.
* Có khả năng phát triển linh hoạt khi có nhu cầu bổ xung thêm các tiện ích mới.
* Sử dụng tiện lợi trên các thiết bị: máy tính xách tay, máy tính để bàn, điện thoại thông minh.

1. Giới thiệu về công nghệ, ngôn ngữ sử dụng
   1. Giới thiệu về ngôn ngữ Java
2. Khái niệm

* Java là một ngôn ngữ lập lập trình, được phát triển bởi **Sun Microsystem** vào năm 1995, là ngôn ngữ kế thừa trực tiếp từ C/C++ và là một ngôn ngữ lập trình hướng đối tượng.
* Java là tên một hòn đảo ở Indonesia - hòn đảo nổi tiếng với loại coffee Peet và cũng là loại nước uống phổ biến của các kỹ sư Sun. Ban đầu Ngôn ngữ này được đặt tên là “oak” (có nghĩa là cây sồi 1991)*,* nhưng các luật sư của Sun xác định rằng tên đó đã được đăng ký nhãn hiệu nên các nhà phát triển đã phải thay thế bằng một tên mới -  và cũng vì lý do trên mà cái tên Java đã ra đời và trở thành tên gọi chính thức của Ngôn ngữ này - Ngôn ngữ Lập trình Java.

1. Ứng dụng của Java

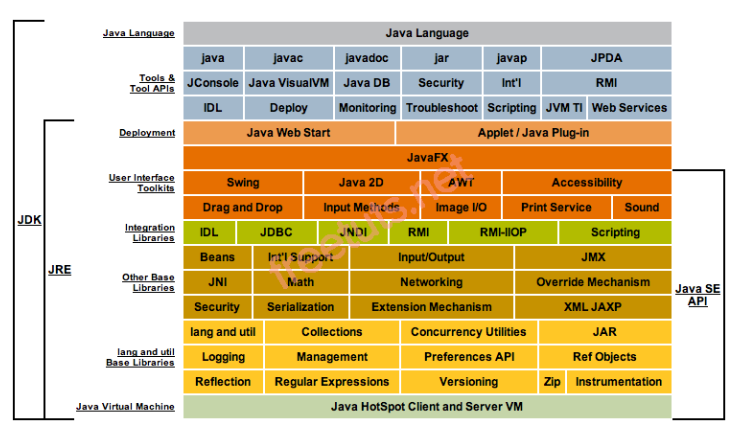
* Ngày nay Java được sử dụng với các mục đích sau:
* Phát triển ứng dụng cho các thiết bị điện tử thông minh, các ứng dụng cho doanh nghiệp với quy mô lớn.
* Tạo các trang web có nội dung động (web applet), nâng cao chức năng của server.
* Phát triển nhiều loại ứng dụng khác nhau: Cơ sở dữ liệu, mạng, Internet, viễn thông, giải trí,...

1. Những đặc điểm cơ bản của Java

* Tiêu chí hàng đầu của Ngôn ngữ Lập trình Java là **"Write Once, Run Anywhere"** (viết 1 lần chạy mọi nơi), nghĩa là Java cho phép chúng ta viết code một lần và thực thi được trên các hệ điều hành khác nhau. Ví dụ, chúng ta viết code trên Hệ điều hành Windows và nó có thể thực thi được trên các Hệ điều hành Linux và Mac OS...
* Với đặc điểm nổi bật đó, Java có những đặc điểm cơ bản như sau:
* Đơn giản và quen thuộc: Vì Java kế thừa trực tiếp từ C/C++ nên nó có những đặc điểm của ngôn ngữ này, Java đơn giản vì mặc dù dựa trên cơ sở C++ nhưng Sun đã cẩn thận lược bỏ các tính năng khó nhất của của C++ để làm cho ngôn ngữ này dễ sử dụng hơn.
* Hướng đối tượng và quen thuộc.
* Mạnh mẽ (thể hiện ở cơ chế tự động thu gom rác – garbage collection)và an toàn.
* Kiến trúc trung lập, độc lập nền tảng và có tính khả chuyển (Portabbility).
* Hiệu suất cao.
* Máy ảo (biên dịch và thông dịch).
* Phân tán.
* Đa nhiệm: Ngôn ngữ Java cho phép xâ dựng trình ứng dụng, trong đó nhiều quá trình có thể xảy ra đồng thời. Tính đa nhiệm cho phép các nhà lập trình có thể biên soạn phần mềm đáp ứng tốt hơn, tương tác tốt hơn và thực hiện theo thời gian thực.

1. Các platform cơ bản của Java

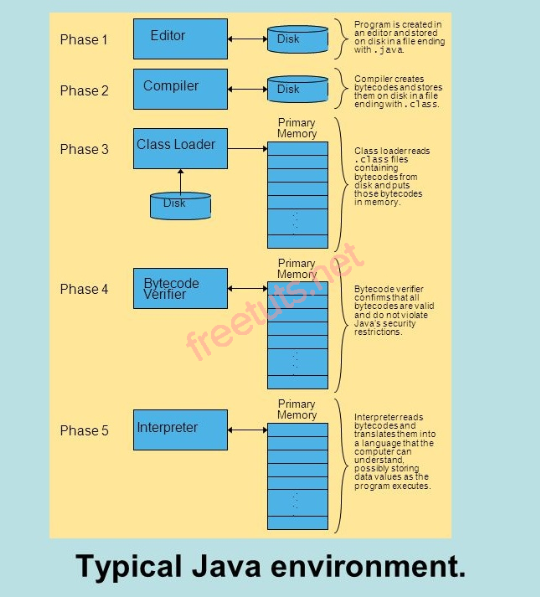
* Java Platform gồm có 3 thành phần chính:
* Java Virtual Machine (JVM): Máy ảo Java.
* Java Application Programming Interface(Java API).
* Java Development Kit (JDK) gồm trình biên dịch, thông dịch, trợ giúp, soạn tài liệu... và các thư viện chuẩn.



Hình 1. Các thành phần platform Java

1. Tiêu chuẩn của một môi trường Java điển hình

* Thông thường, các chương trình Java trải qua 5 giai đoạn chính:
* **Editor**: Lập trình viên viết chương trình và được lưu vào máy tính với định dạng .java.
* **Compiler**: Biên dịch chương trình thành bytecodes (ddingj dạng .class) - nhờ bước trung gian này mà Java được viết 1 lần và chạy trên các hệ điều hành khác nhau.
* **Class Loader**: Đọc file .class chứa mã bytecodes và lưu vào trong bộ nhớ.
* **Bytecode Verifier**: Đảm bảo rằng mã bytecodes là hợp lệ và không vi phạm các vấn đề về bảo mật của Java.
* **Intepreter**: Biên dịch bytecodes thành mã máy để máy tính có thể hiểu được và sau đó thực thi chương trình.



Hình 1. Môi trường Java điển hình

* 1. Giới thiệu về Spring Framework

1. Giới thiệu Spring Framework

* **Spring Framework** là một [bộ khung ứng dụng](https://vi.wikipedia.org/w/index.php?title=B%E1%BB%99_khung_%E1%BB%A9ng_d%E1%BB%A5ng&action=edit&redlink=1) và [bộ chứa](https://en.wikipedia.org/wiki/Servlet_container) [đảo ngược điều khiển](https://en.wikipedia.org/wiki/inversion_of_control) cho [nền tảng Java](https://vi.wikipedia.org/wiki/Java_(c%C3%B4ng_ngh%E1%BB%87)). Chức năng tính của bộ khung này có thể áp dụng cho bất kỳ ứng dụng Java nào, tuy vậy, nhiều bản mở rộng dành cho việc xây dựng ứng dụng nền web cũng được phát triển trên nền tảng [Java EE](https://vi.wikipedia.org/w/index.php?title=Java_EE&action=edit&redlink=1). Tuy bộ khung không ấn định lên một cụ thể nào, Spring Framework đã trở nên phổ biến trong cộng đồng [Java](https://vi.wikipedia.org/wiki/Java_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)) như một phương án song hành, thay thế, hoặc thậm chí bổ sung cho mô hình [Enterprise JavaBeans](https://vi.wikipedia.org/w/index.php?title=Enterprise_JavaBeans&action=edit&redlink=1) (EJB).

1. Lịch sử ra đời của Spring Framework

* Phần đầu tiên của Spring Framework ban đầu nguyên được [Rod Johnson](https://vi.wikipedia.org/w/index.php?title=Rod_Johnson&action=edit&redlink=1) viết vào năm 2000, trong khi ông đang làm một chuyên viên cố vấn cho các khách hàng trong thị trường [kinh tế tài chính](https://vi.wikipedia.org/w/index.php?title=Kinh_t%E1%BA%BF_t%C3%A0i_ch%C3%ADnh&action=edit&redlink=1) tại [Luân Đôn](https://vi.wikipedia.org/wiki/Lu%C3%A2n_%C4%90%C3%B4n). Trong lúc đang viết quyển Expert One-on-one J2EE Design And Development (Programmer to programmer) (Thiết kế và kiến tạo Java EE - tương giao cá nhân một đối một chuyên ngành (lập trình viên với lập trình viên), ông đã mở rộng thêm phần mã nguồn, hòng biểu đạt ý tưởng của mình về mối tương quan của các trình ứng dụng với những phần khác nhau của [J2EE](https://vi.wikipedia.org/wiki/J2EE) và cho rằng người ta có thể đơn giản hóa mối tương quan của chúng, và mối liên hệ ấy có thể trở nên cố định, bất biến hơn là những gì mà các lập trình viên và các công ty vốn quen sử dụng tại thời điểm đó.

1. Tính năng của Spring Framework

* Bao gồm bộ quản lý cài đặt rất có hiệu lực dựa trên JavaBeans, áp dụng lý thuyết [đảo ngược quyền khống chế](https://vi.wikipedia.org/w/index.php?title=%C4%90%E1%BA%A3o_ng%C6%B0%E1%BB%A3c_quy%E1%BB%81n_kh%E1%BB%91ng_ch%E1%BA%BF&action=edit&redlink=1) (IoC) làm cho việc mắc nối các chương trình ứng dụng trở nên nhanh chóng và dễ dàng hơn. Việc sử dụng (IoC) trong khung hình Spring Framework còn được ám chỉ đến với cái tên [Dependency Injection](https://vi.wikipedia.org/w/index.php?title=Dependency_Injection&action=edit&redlink=1) (DI) (Nghĩa đen: bơm hay tiếp thành phần phụ thuộc, từ bên ngoài vào, lúc cần thiết, thay vì phải tự tạo từ bên trong). Nguyên lý của IoC sử dụng trong Spring Framework là một kỹ thuật nhằm ngoại biên hóa (externalize) việc kiến tạo và quản lý những thành phần phụ thuộc. Lấy ví dụ trường hợp lớp "Foo" phụ thuộc vào một trường hợp cá biệt của lớp "Bar" để thi hình một nhiệm vụ nào đấy. Theo phương pháp cổ truyền, "Foo" phải tự kiến tạo trường hợp cá biệt của lớp "Bar", dùng hành động "new", hoặc phải tiếp nhận trường hợp cá biệt của lớp này từ một lớp chế xuất nào đấy (factory class). Nếu sử dụng tiếp cận của IoC thì một tiến trình ngoại biên sẽ phải cung cấp trường hợp của "Bar" (hoặc một tiểu lớp của nó) cho "Foo" trong thời gian chạy (runtime). Đây chính là nguyên nhân tại sao "DI" là một thành ngữ diễn tả tính năng của Spring Framework rõ hơn "IoC".
* Bao gồm một [mẫu hình chế xuất trừu tượng](https://vi.wikipedia.org/w/index.php?title=M%E1%BA%ABu_h%C3%ACnh_ch%E1%BA%BF_xu%E1%BA%A5t_tr%E1%BB%ABu_t%C6%B0%E1%BB%A3ng&action=edit&redlink=1) (abstract factory pattern) [Enterprise Java Beans](https://vi.wikipedia.org/w/index.php?title=Enterprise_Java_Beans&action=edit&redlink=1) (EJB) nòng cốt, có thể sử dụng trong bất cứ môi trường nào, từ những khung chứa (containers) [applet](https://vi.wikipedia.org/w/index.php?title=Applet&action=edit&redlink=1) cho đến [Java EE](https://vi.wikipedia.org/w/index.php?title=Java_EE&action=edit&redlink=1).
* Một lớp trừu tượng chung quản lý [giao dịch cơ sở dữ liệu](https://vi.wikipedia.org/wiki/Giao_d%E1%BB%8Bch_c%C6%A1_s%E1%BB%9F_d%E1%BB%AF_li%E1%BB%87u) (database transaction), cho phép các bộ phận quản lý giao dịch được bổ trợ theo lối lắp ráp (pluggable), và tạo điều kiện thuận lợi cho việc phân định danh giới các giao dịch (demarcate transactions) mà không cần phải xử lý với những vấn đề ở tầng dưới (low-level issues).
* Bao gồm một số các chiến lược phổ thông đã được cài sẵn bên trong (Built-in generic strategies) dành cho [JTA](https://vi.wikipedia.org/w/index.php?title=Java_Transaction_API&action=edit&redlink=1) và một đơn thể nguồn dữ liệu [JDBC](https://vi.wikipedia.org/w/index.php?title=Java_Database_Connectivity&action=edit&redlink=1) (DataSource). Đối lập với những giao dịch do khung chứa quản lý (Container-Managed Transactions - CMT) đơn thuần trong JTA hoặc [EJB](https://vi.wikipedia.org/w/index.php?title=EJB&action=edit&redlink=1), hỗ trợ giao dịch trong bộ khung Spring không đòi hỏi môi trường [Java EE](https://vi.wikipedia.org/w/index.php?title=Java_EE&action=edit&redlink=1).
* Tầng liên nối với cơ sở dữ liệu (JDBC abstraction layer ) cung cấp một cơ chế tầng cấp đầy ý nghĩa đối với những ngoại lệ (exception hierarchy) (không còn phải dựa vào việc nắn bóp mã nguồn của nhà sản xuất phần mềm như đối với SQLException nữa), đơn giản hóa việc xử lý lỗi, và cũng đồng thời thuyên giảm số lượng dòng lệnh mà lập trình viên cần phải viết. Lập trình viên sẽ không bao giờ phải viết khối 'finally' trong các dòng mã sử dụng JDBC nữa.
* Tích hợp với [Hibernate](https://vi.wikipedia.org/w/index.php?title=Hibernate_(Java)&action=edit&redlink=1), [JDO](https://vi.wikipedia.org/w/index.php?title=Java_Data_Objects&action=edit&redlink=1) và [iBATIS](https://vi.wikipedia.org/w/index.php?title=IBATIS&action=edit&redlink=1) [SQL Maps](https://vi.wikipedia.org/w/index.php?title=SQL_Maps&action=edit&redlink=1): trên phương diện cầm dữ tài nguyên (resource holders), hỗ trợ phần thực thi DAO (Data Access Object - Đối tượng truy cập dữ liệu), và các chiến lược giao dịch (transaction strategies). Đứng hàng đầu trong việc hỗ trợ Hibernate với rất nhiều tính năng IoC tiện dụng, giải quyết nhiều nan đề trong việc tích hợp với Hibernate. Tất cả những tính năng này đều phục tùng cơ chế tầng cấp giao dịch chung cùng tầng cấp ngoại lệ DAO của Spring.
* Khung hình [mô hình-giao diện-điều khiển](https://vi.wikipedia.org/w/index.php?title=M%C3%B4_h%C3%ACnh-giao_di%E1%BB%87n-%C4%91i%E1%BB%81u_khi%E1%BB%83n&action=edit&redlink=1) (Model-view-controller, hay MVC) linh hoạt dành cho [ứng dụng mạng](https://vi.wikipedia.org/w/index.php?title=%E1%BB%A8ng_d%E1%BB%A5ng_m%E1%BA%A1ng&action=edit&redlink=1) (web application), được gây dựng trên tính năng nền tảng của Spring. Các nhà xây dựng phần mềm (lập trình viên) có quyền khống chế cao độ trong khung hình này thông qua những giao diện chiến lược (strategy interfaces), đồng thời điều tiết với những kỹ thuật đa giao diện (multiple view technologies) như [JSP](https://vi.wikipedia.org/wiki/JSP), [FreeMarker](https://vi.wikipedia.org/wiki/FreeMarker), [Velocity](https://vi.wikipedia.org/w/index.php?title=Velocity_(software)&action=edit&redlink=1), [Tiles](https://vi.wikipedia.org/w/index.php?title=Tiles&action=edit&redlink=1), [iText](https://vi.wikipedia.org/w/index.php?title=IText&action=edit&redlink=1), và [POI](https://vi.wikipedia.org/w/index.php?title=Apache_Jakarta_POI&action=edit&redlink=1). Nên nhớ là [Tầng trung gian](https://vi.wikipedia.org/w/index.php?title=3_tier&action=edit&redlink=1) của Spring có thể dễ dàng kết hợp với một tầng mạng sử dụng bất cứ một khung hình MVC mạng nào đó (web MVC framework), như [Struts](https://vi.wikipedia.org/wiki/Struts), [WebWork](https://vi.wikipedia.org/w/index.php?title=WebWork&action=edit&redlink=1), hoặc [Tapestry](https://vi.wikipedia.org/w/index.php?title=Tapestry&action=edit&redlink=1).
* Khung hình [lập trình định dạng](https://vi.wikipedia.org/w/index.php?title=L%E1%BA%ADp_tr%C3%ACnh_%C4%91%E1%BB%8Bnh_d%E1%BA%A1ng&action=edit&redlink=1) (Aspect-oriented programming - hay AOP) nhằm cung cấp các dịch vụ như dịch cụ quản lý giao dịch (transaction management). AOP cung cấp khả năng thực thi lôgic [đan kết](https://vi.wikipedia.org/w/index.php?title=%C4%90an_k%E1%BA%BFt&action=edit&redlink=1) (crosscutting logic) - tức những lôgic tác động đến rất nhiều phần của chương trình ứng dụng - tại cùng một chỗ (nhóm lại) và để cho lôgic ấy áp dụng xuyên suốt chương trình ứng dụng của mình một cách tự động.

1. Các modun của Spring Framework

* Spring Framework được coi như là một tổ hợp của nhiều khung hình nhỏ, hay nói cách khác, nó là tổ hợp của khung hình trong khung hình. Đại bộ phận các khung hình này được thiết kế để hoạt động biệt lập với các khung hình khác, song lại tạo nên nhiều chức năng tốt hơn khi được phối hợp làm việc với nhau. Những khung hình này được phân loại theo các khối nền cơ bản mà những chương trình phần mềm phức tạp điển hình thường sử dụng:
* Khung chứa [đảo ngược khống chế](https://vi.wikipedia.org/w/index.php?title=%C4%90%E1%BA%A3o_ng%C6%B0%E1%BB%A3c_kh%E1%BB%91ng_ch%E1%BA%BF&action=edit&redlink=1) (Inversion of Control container): Sự cài đặt các thành phần của chương trình ứng dụng và quản lý chu trình của các đối tượng trong Java (lifecycle management of Java objects).
* Khung hình [lập trình định dạng](https://vi.wikipedia.org/w/index.php?title=L%E1%BA%ADp_tr%C3%ACnh_%C4%91%E1%BB%8Bnh_d%E1%BA%A1ng&action=edit&redlink=1) (Aspect-oriented programming framework): Spring AOP module tích hợp chức năng lập trình hướng khía cạnh vào Spring framework thông qua cấu hình của nó. Spring AOP module cung cấp các dịch vụ quản lý giao dịch cho các đối tượng trong bất kỳ ứng dụng nào sử dụng Spring. Với Spring AOP chúng ta có thể tích hợp declarative transaction management vào trong ứng dụng mà không cần dựa vào EJB component.
* Spring AOP module cũng đưa lập trình metadata vào trong Spring. Sử dụng cái này chúng ta có thể thêm annotation vào source code để hướng dẫn Spring nơi và làm thế nào để liên hệ với aspect..
* Khung hình [truy cập dữ liệu](https://vi.wikipedia.org/wiki/Truy_c%E1%BA%ADp_d%E1%BB%AF_li%E1%BB%87u) (Data access framework):Tầng JDBC và DAO đưa ra một cây phân cấp exception để quản lý kết nối đến database, điều khiển exception và thông báo lỗi được ném bởi vendor của database. Tầng exception đơn giản điều khiển lỗi và giảm khối lượng code mà chúng ta cần viết như mở và đóng kết nối. Module này cũng cung cấp các dịch vụ quản lý giao dịch cho các đối tượng trong ứng dụng Spring..
* Khung hình [quản lý giao dịch](https://vi.wikipedia.org/w/index.php?title=Qu%E1%BA%A3n_l%C3%BD_giao_d%E1%BB%8Bch&action=edit&redlink=1) (Transaction processing / Transaction management framework): Sự đồng hòa các API quản lý giao dịch và sự điều hợp quản lý giao dịch đối với các đối tượng Java dùng phương pháp cài đặt.
* Khung hình [mô hình-giao diện-điều khiển](https://vi.wikipedia.org/w/index.php?title=M%C3%B4_h%C3%ACnh-giao_di%E1%BB%87n-%C4%91i%E1%BB%81u_khi%E1%BB%83n&action=edit&redlink=1) (Model-view-controller framework): Khung hình dựa trên nền tảng [HTTP](https://vi.wikipedia.org/wiki/Hypertext_Transfer_Protocol) và [Servlet](https://vi.wikipedia.org/w/index.php?title=Java_Servlet_API&action=edit&redlink=1) cung cấp rất nhiều móc điểm có thể mở rộng và cải biên (extension and customization).
* Khung hình truy cập từ xa (Remote Access framework): Xuất nhập khẩu các đối tượng Java trên mạng lưới truyền thông hỗ trợ các giao thức dựa trên nền [HTTP](https://vi.wikipedia.org/wiki/Hypertext_Transfer_Protocol), sử dụng phong cách [RPC](https://vi.wikipedia.org/w/index.php?title=RPC&action=edit&redlink=1) (Remote procedure call) thông qua cài đặt, như [RMI](https://vi.wikipedia.org/w/index.php?title=RMI&action=edit&redlink=1) (Java remote method invocation - khởi động phương pháp từ xa trong Java), [CORBA](https://vi.wikipedia.org/w/index.php?title=CORBA&action=edit&redlink=1) và [dịch vụ mạng](https://vi.wikipedia.org/wiki/D%E1%BB%8Bch_v%E1%BB%A5_m%E1%BA%A1ng) (web services - [SOAP](https://vi.wikipedia.org/wiki/SOAP)).
* Khung hình [xác thực](https://vi.wikipedia.org/wiki/X%C3%A1c_th%E1%BB%B1c) và [ủy quyền](https://vi.wikipedia.org/w/index.php?title=%E1%BB%A6y_quy%E1%BB%81n&action=edit&redlink=1) (Authentication and authorization framework): Sự điều hợp các quá trình xác thực và ủy quyền thông qua cài đặt hỗ trợ nhiều tiêu chuẩn công nghiệp và những tiêu chuẩn được sử dụng phổ biến, cũng như các giao thức, công cụ và các thói quen thực hành phổ dụng.
* Khung hình quản lý từ xa (Remote Management framework): Sự quản lý và quảng bá các đối tượng Java đối với các bố trí địa phương và bố trí ngoại biên thông qua [JMX](https://vi.wikipedia.org/w/index.php?title=JMX&action=edit&redlink=1) (Java Management Extension - quản lý mở rộng của Java cho phép các phần mềm không trực thuộc Java có thể lắp ráp vào và sử dụng được trong nền tảng Java).
* Khung hình truyền thông điệp (Messaging framework): đăng ký các đối tượng lắng nghe thông điệp dùng để tiếp thu và sử dụng thông điệp từ các [hàng thông điệp](https://vi.wikipedia.org/w/index.php?title=H%C3%A0ng_th%C3%B4ng_%C4%91i%E1%BB%87p&action=edit&redlink=1) (message queue) thông qua [JMS](https://vi.wikipedia.org/w/index.php?title=JMS&action=edit&redlink=1) (Java Message Service - Dịch vụ Thông điệp của Java), cải tiến việc truyền gửi thông điệp dùng tiêu chuẩn trong [API](https://vi.wikipedia.org/wiki/Giao_di%E1%BB%87n_l%E1%BA%ADp_tr%C3%ACnh_%E1%BB%A9ng_d%E1%BB%A5ng) của [JMS](https://vi.wikipedia.org/w/index.php?title=JMS&action=edit&redlink=1).
* Khung hình [kiểm thử phần mềm](https://vi.wikipedia.org/w/index.php?title=Ki%E1%BB%83m_th%E1%BB%AD_(ph%E1%BA%A7n_m%E1%BB%81m)&action=edit&redlink=1) (Software testing framework): Những chủ thể (lớp) hỗ trợ việc viết các kiểm thử đơn vị và các kiểm thử tích hợp (unit tests and integration tests).
  1. Giới thiệu về Spring Boot

1. Giới thiệu Spring Boot

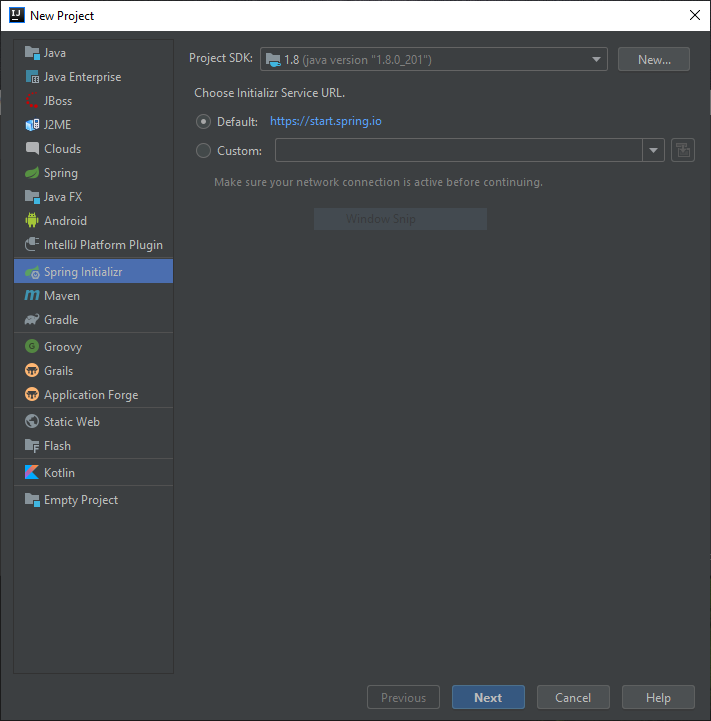
* **Spring Boot** là một dự án phát triển bởi **JAV**(ngôn ngữ java) trong hệ sinh thái Spring framework. Nó giúp cho các lập trình viên chúng ta đơn giản hóa quá trình lập trình một ứng dụng với Spring, chỉ tập trung vào việc phát triển business cho ứng dụng.
* Có các tính năng của Spring Framework.
* Tạo ứng dụng độc lập, có thể chạy bằng java -jar (cho cả java web)
* Nhúng trực tiếp các ứng dụng server (Tomcat, Jetty…) do đó không cần phải triển khai file WAR
* Cấu hình ít, tự động cậu hình bất kì khi nào có thể (Giảm thời gian viết code, tăng năng suất)
* Không yêu cầu XML config…
* Cung cấp nhiều plugin
* Chuẩn cho Microservices (Cloud support; giảm việc setup, config; các thư viện hỗ trợ…)

1. Các tính năng nổi bật

* Đóng gói ứng dụng Spring dưới dạng một file JAR (stand-alone application). Chúng ta có thể dễ dàng start ứng dụng Spring chỉ với câu lệnh quen thuộc java -jar.
* Tối ưu công đoạn cấu hình cho ứng dụng Spring, không sinh code cấu hình và không yêu cầu phải cấu hình bằng XML.
* Cung cấp một loạt các tính năng phi chức năng phổ biến cho các dự án lớn như nhúng trực tiếp web server như Tomcat, Jetty, ... vào ứng dụng, bảo mật, health check...

1. Tạo một Project Spring Boot với Intelij Idea

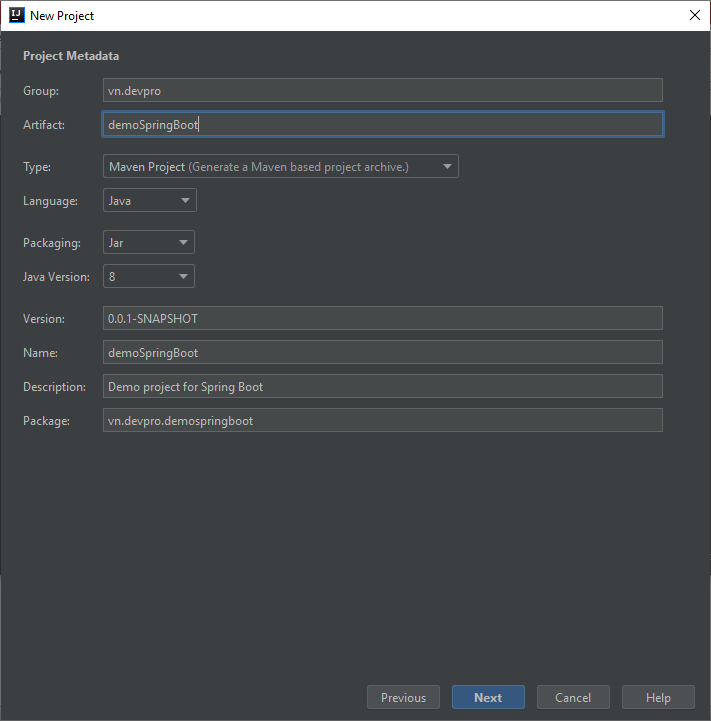
Bước 1: Chọn Spring Initializr và chọn phiên bản JDK của bạn. Sau đó chọn Next.



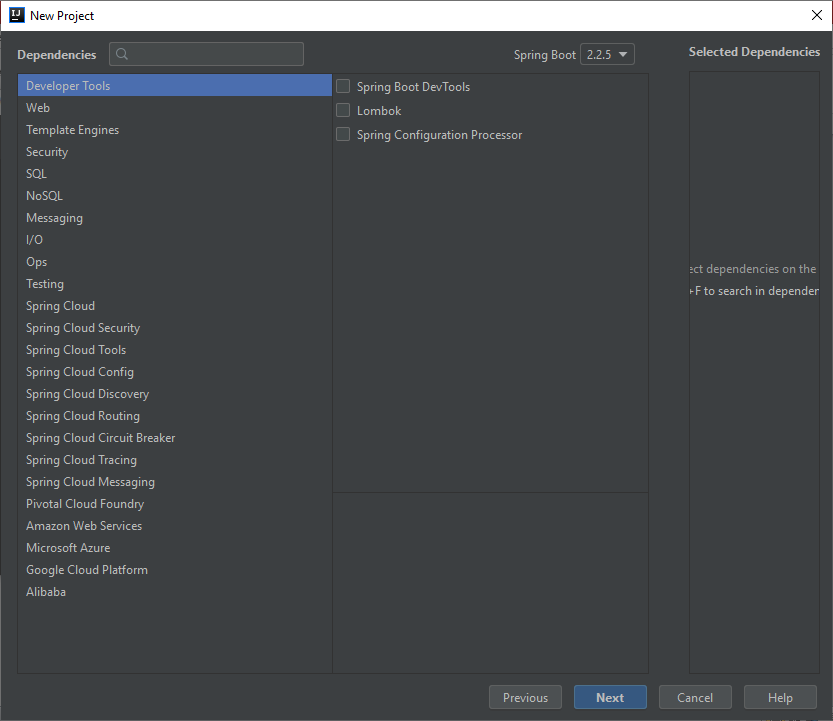
Bước 2: Thiết lập các thông tin cơ bản cho project, gồm các thông tin:

* Group: Tên tổ chức /công ty /cá nhân của dự án.
* Artifact: Tên package của dự án.
* Type: Cấu hình trình quản lí dự án.
* Language: Loại ngôn ngữ sử dụng cho dự án.
* Paking: Sản phẩm sau khi build dự án.
* Java Version: Phiên bản java sử dụng trong dự án.
* Version: Phiên bản dự án là lần thứ mấy.
* Name:Tên của dự án.
* Description:Mo tả dự án.
* Package: gói sẽ chứa các file hoạt động của dự án.

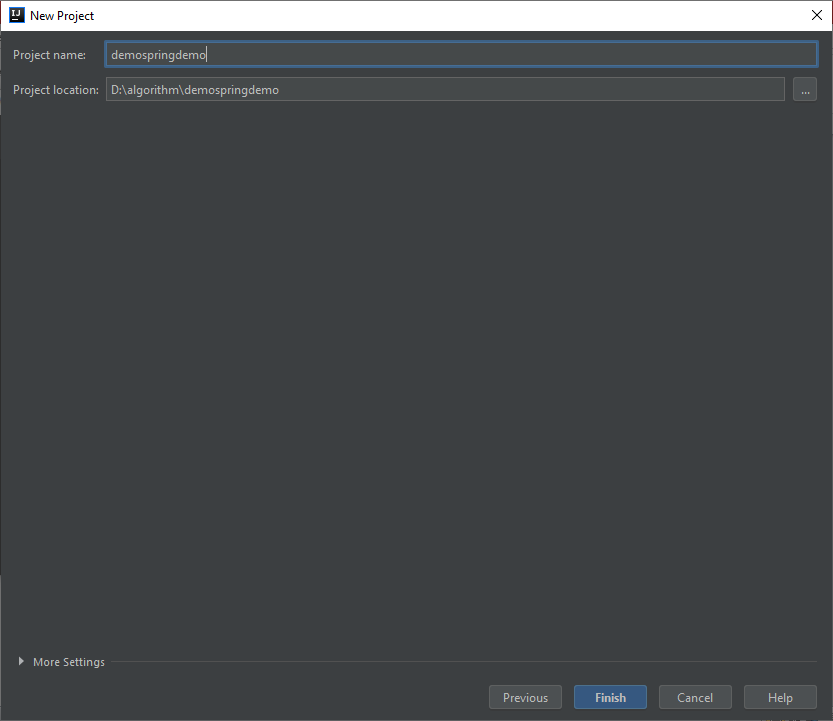
sau đó chọn Next.



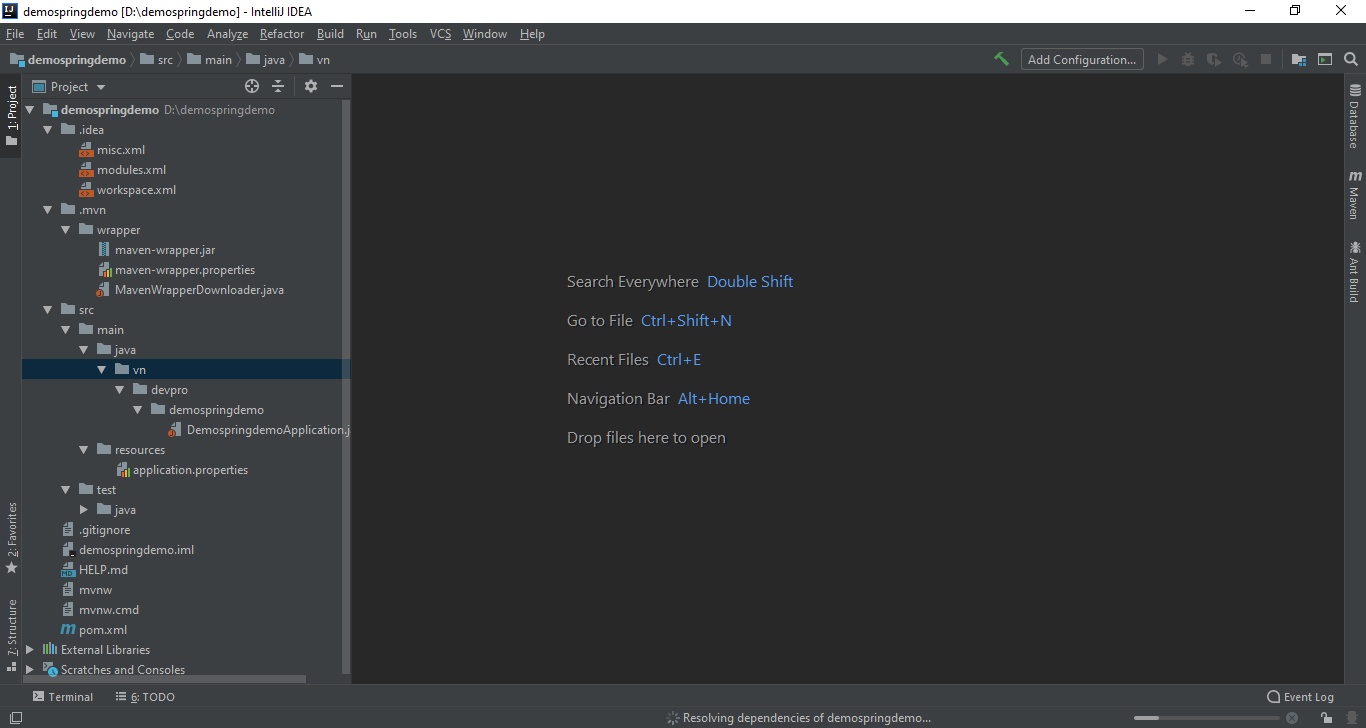
Bước 3: chọn các công cụ mà bạn sẽ sử dụng trong dự án của mình. Nếu không xác định trước thì để mặc định rồi chọn Next.



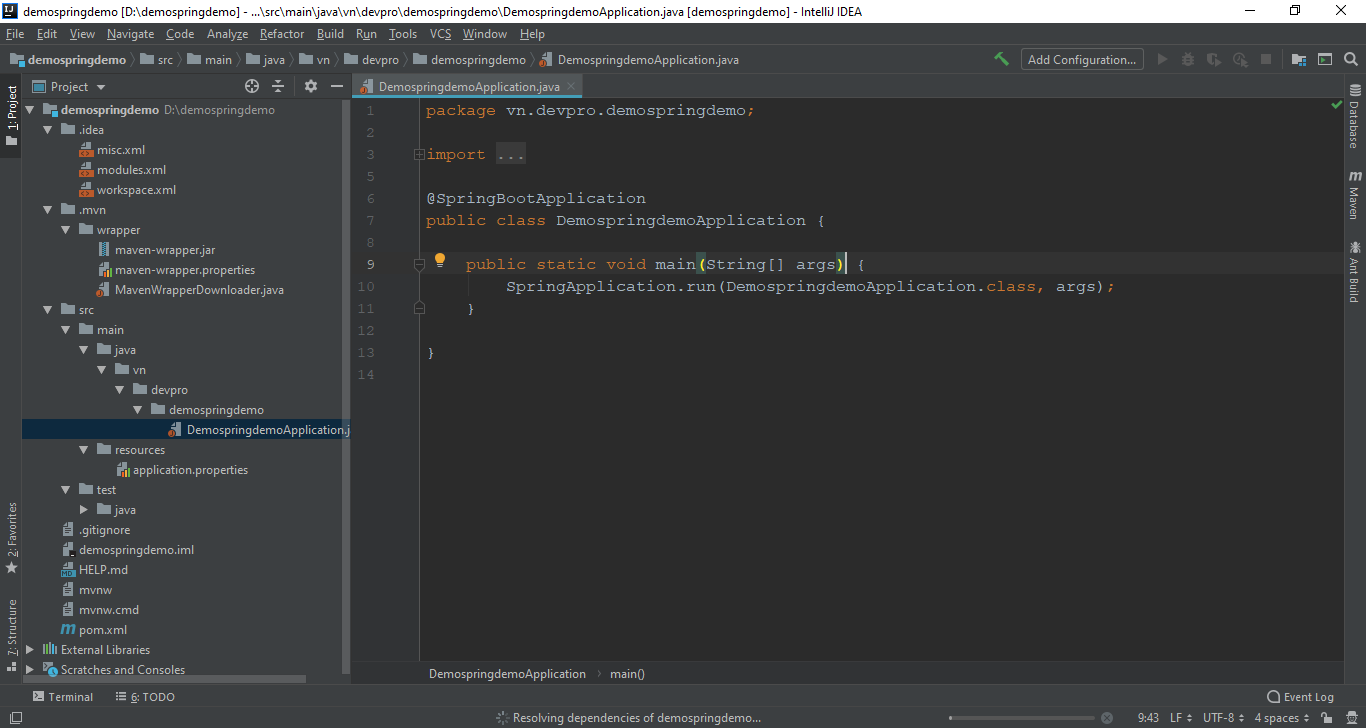
Bước 4:Đặt tên cho project và setup nơi lưu trữ dự án. Sau đó chọn Finish.



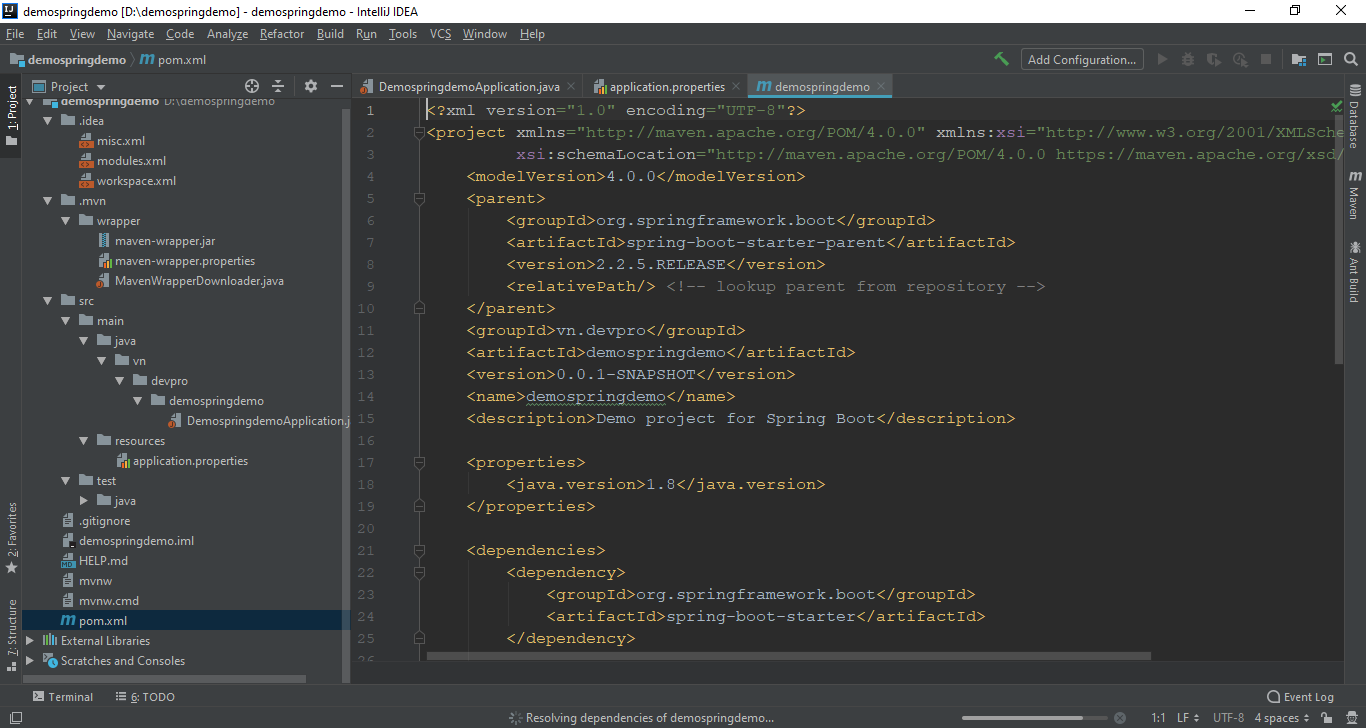
Sau khi hoàn tất chúng ta được một project đã được cấu hình sẵn.



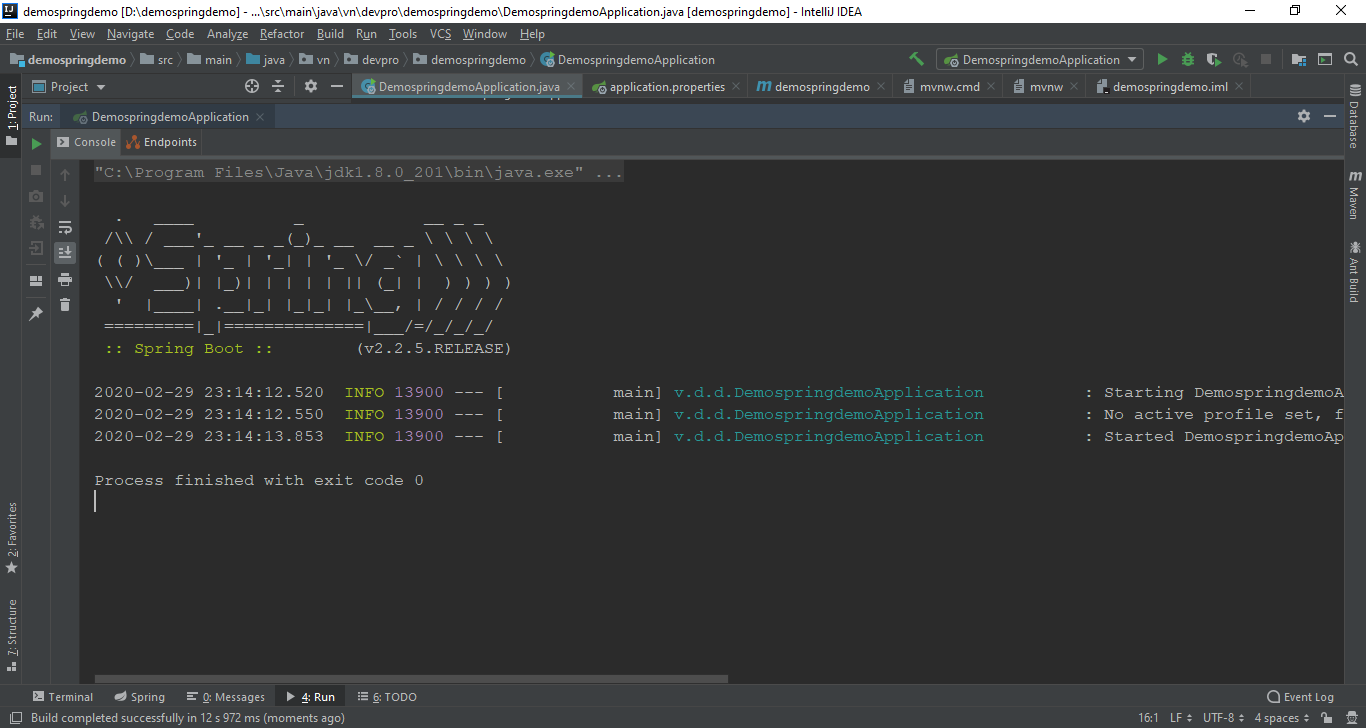
Đây là file chứa hàm main() project:



File pom.xml là nơi khai báo các thư viện mà chúng ta sử dụng.



Màn hình console khi chạy project.



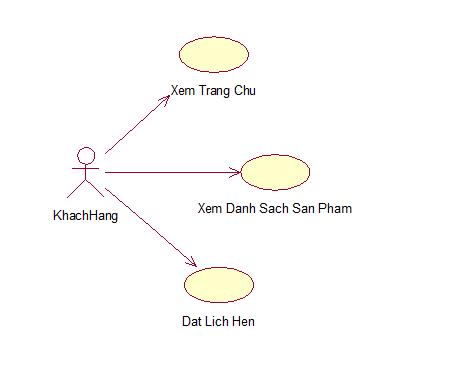
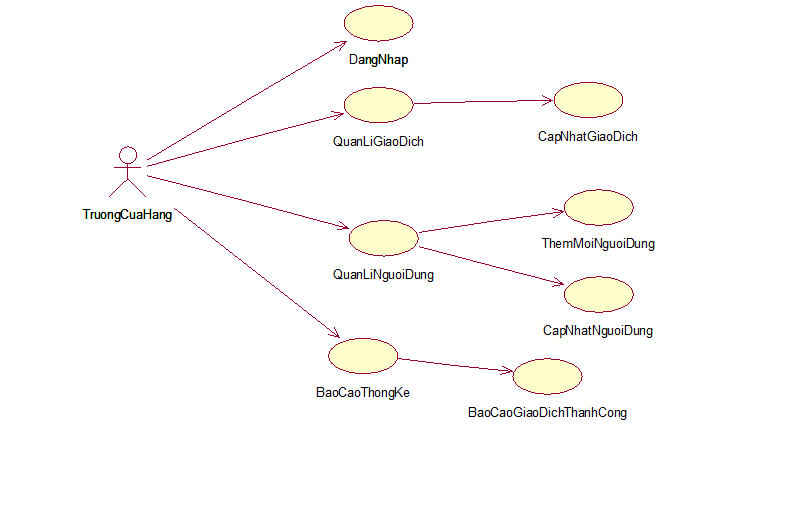
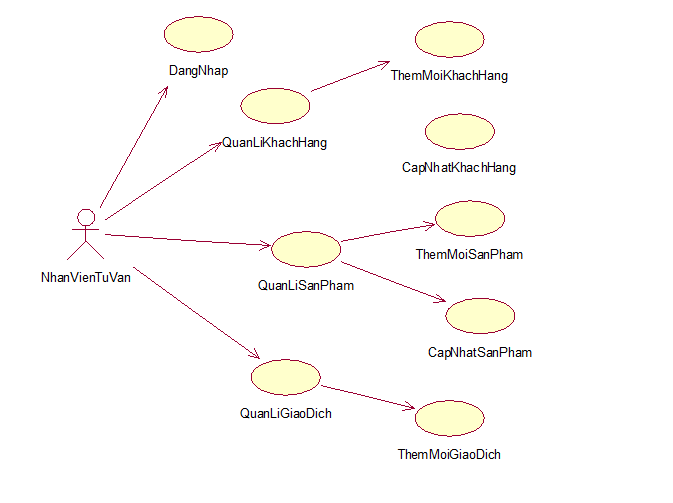
# CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

1. Phân tích biểu đồ usecase
2. Xác định các tác nhân

* Khách hàng
* Trưởng cửa hàng
* Nhân viên tư vấn

1. Xác định usecase

* Khách hàng
* Xem trang chủ
* Xem danh sách sản phẩm
* Đặt lịch hẹn
* Trưởng cửa hàng
* Thêm mới người dùng
* Cập nhật người dùng
* Cập nhật giao dịch
* Xem báo cáo thống kê
* Nhân viên tư vấn
* Thêm mới thông tin khách hàng
* Cập nhật thông tin khách hàng
* Tạo mới giao dịch
* Thêm mới sản phẩm
* Chỉnh sửa sản phẩm

1. Biểu đồ usecase
2. Usecase khách hàng
3. Usecase trưởng cửa hàng
4. Usecase nhân viên tư vấn
5. Đặc tả usecase
6. Xem trang chủ
7. Mô tả vắn tắt

* Tên ca sử dụng: Xem trang chủ.
* Mục đích: Hiển thị trang chủ cho khách hàng.
* Tác nhân: Khách hàng.

1. Các luồng sự kiện

* Ca sử dụng bắt đầu từ khi khách hàng muốn tìm hiểu thông tin về đồ cưới.
* Khách hàng tìm kiếm bằng từ khóa trên google,
* Danh sách các kết quả được trả về.
* Khách hàng click chọn link website của cửa hàng.
* Khách hàng được chuyển đến website trang chủ của cửa hàng.

1. Tiền điều kiện
2. Hậu điều kiện

* Khách hàng truy cập được vào trang chủ website của cửa hàng.

1. Xem danh sách sản phẩm
2. Mô tả vắn tắt

* Tên ca sử dụng: Xem danh sách sản phẩm.
* Mục đích: Giúp khách hàng xem thông tin các sản phẩm của cửa hàng.
* Tác nhân: Khách hàng.

1. Các luồng sự kiện

* Ca sử dụng bắt đầu khi khách hàng muốn xem danh sách các sản phẩm.
* Khách hàng click chuột chọn danh mục sản phẩm.
* Hệ thống lấy danh sách các sản phẩm hiển thị lên màn hình.
* Khách hàng chọn 1 sản phẩm và click chuột để xem chi tiết sản phẩm.
* Hệ thống lấy chi tiết sản phẩm được yêu cầu hiển thị lên màn hình.
* Nếu khách hàng muốn tìm sản phẩm khác, khách hàng nhập tên sản phẩm muốn tìm vào textbox tìm kiếm, rồi click chuột vào tìm kiếm.
* Hệ thống lấy danh sách sản phẩm có chứa từ khóa tìm kiếm và hiện thị lên màn hình.

Nếu không có kết quả phù hợp => A1

* Luồng sự kiện rẽ nhánh:
* A1: Không có kết nào được trả về, hệ thống hiển thị trang dánh sách trống.

1. Tiền điều kiện

* Khách hàng đang ở trang chủ.

1. Hậu điều kiện

* Hiển thị danh sách các sản phẩm lên màn hình.

1. Đăng kí lịch hẹn
2. Mô tả vắn tắt

* Tên ca sử dụng: Đăng kí lịch hẹn.
* Mục đích: Giúp khách hàng đăng kí lịch hẹn với cửa hàng.
* Tác nhân: Khách hàng.

1. Các luồng sự kiện

* Ca sử dụng bắt đầu khi khách hàng muốn đăng kí hẹn tại cửa hàng. Khách hàng click chuột chọn đăng kí hẹn.
* Hệ thống hiển thị trang đăng kí.
* Khách hàng nhập đầy đủ thông tin bắt buộc rồi chọn đăng kí.
* Hệ thống ghi nhận thông tin đăng kí của khách hàng.

Nêu không nhập đầy đủ thông tin bắt buộc => A1

* Luồng sự kiện rẽ nhánh:
* A1: Hiển thị lỗi không cho đăng kí, yêu cầu khách hàng nhập đầy đủ thông tin cần thiết.

1. Tiền điều kiện

* Khách hàng đang ở trang chủ, trang danh sách sản phẩm.

1. Hậu điều kiện

* Hiển thị thông báo.

1. Thêm mới khách hàng
2. Mô tả vắn tắt

* Tên ca sử dụng: Thêm mới khách hàng.
* Mục đích: Giúp nhân viên tư vấn thêm mới khách hàng.
* Tác nhân: Nhân viên tư vấn.

1. Các luồng sự kiện

* Ca sử dụng bắt đầu khi nhân viên tư vấn cần thêm một khách hàng mới.
* Nhân viên tư vấn chọn thêm mới.
* Hệ thống hiện thị form thêm mới khách hàng.
* Nhân viên tư vấn nhập thông tin khách hàng mới rồi click chọn thêm mới.

Nếu không nhập đầy đủ thông tin bắt buộc => A1

* Hệ thống thêm mới khách hàng vào danh sách khách hàng, cập nhập lại danh sách khách hàng và hiển thị lên màn hình.
* Luồng sự kiện rẽ nhánh:
* A1: Hiện thị thông báo lỗi: Thông tin khách hàng không hợp lệ.

1. Tiền điều kiện

* Nhân viên tư vấn đang ở trang quản lí khách hàng.

1. Hậu điều kiện

* Hiển thị thông báo và danh sách khách hàng sau khi thêm mới.

1. Cập nhật khách hàng
2. Mô tả vắn tắt

* Tên ca sử dụng: Cập nhật khách hàng.
* Mục đích: Giúp nhân viên tư vấn cập nhập thông tin khách hàng.
* Tác nhân: Nhân viên tư vấn.

1. Các luồng sự kiện

* Ca sử dụng bắt đầu khi nhân viên tư vấn cần sửa thông tin một khách hàng.
* Nhân viên tư vấn chọn khách hàng muốn sửa. Click chọn cập nhật.
* Hệ thống hiện thị form cập nhật khách hàng.
* Nhân viên tư vấn nhập thông tin mới rồi click chọn cập nhật.

Nếu không nhập đầy đủ thông tin => A1

* Hệ thống cập nhập thông tin mới của khách hàng, cập nhập lại danh sách khách hàng và hiển thị lên màn hình.
* Luồng sự kiện rẽ nhánh:
* A1: Thông tin khách hàng không hợp lệ.

1. Tiền điều kiện

* Nhân viên tư vấn đang ở trang quản lí khách hàng.

1. Hậu điều kiện

* Hiển thị thông báo và danh sách khách hàng sau khi cập nhập lại thông tin.

1. Thêm mới sản phẩm
2. Mô tả vắn tắt

* Tên ca sử dụng: Thêm mới sản phẩm.
* Mục đích: Giúp nhân viên tư vấn thêm mới sản phẩm.
* Tác nhân: Nhân viên tư vấn.

1. Các luồng sự kiện

* Ca sử dụng bắt đầu khi nhân viên tư vấn cần thêm mới 1 sản phẩm.
* Nhân viên tư vấn chọn thêm mới.
* Hệ thống hiện thị form thêm mới sản phẩm.
* Người dùng nhập thông tin sản phẩm mới rồi click chọn chọn thêm mới.

Nếu không nhập đầy đủ thông tin bắt buộc => A1

* Hệ thống thêm mới sản phẩm vào danh sách sản phẩm, cập nhập lại danh sách sản phẩm và hiển thị lên màn hình.
* Luồng sự kiện rẽ nhánh:
* A1: Hiển thị thông báo lỗi: Thông tin sản phẩm không hợp lệ.

1. Tiền điều kiện

* Nhân viên tư vấn đang ở trang quản lí sản phẩm.

1. Hậu điều kiện

* Hiển thị thông báo và danh sách sản phẩm sau khi thêm mới.

1. Cập nhật sản phẩm
2. Mô tả vắn tắt

* Tên ca sử dụng: Cập nhật sản phẩm.
* Mục đích: Giúp nhân viên tư vấn cập nhập thông tin sản phẩm.
* Tác nhân: Nhân viên tư vấn.

1. Các luồng sự kiện

* Ca sử dụng bắt đầu khi nhân viên tư vấn cần cập nhật thông tin một sản phẩm.
* Nhân viên tư vấn chọn khóa học cần cập nhật. Click chọn cập nhật.
* Hệ thống hiện thị form cập nhật sản phẩm.
* Nhân viên tư vấn nhập thông tin mới rồi click chọn cập nhật.

Nếu không nhập đầy đủ thông tin bắt buộc => A1

* Hệ thống cập nhập thông tin mới của sản phẩm, cập nhập lại danh sách khóa học và hiển thị lên màn hình.
* Luồng sự kiện rẽ nhánh:
* A1: Hiển thị thông báo: Thông tin sản phẩm không hợp lệ.

1. Tiền điều kiện

* Nhân viên tư vấn đang ở trang quản lí sản phẩm.

1. Hậu điều kiện

* Hiển thị thông báo và danh sách sản phẩm sau khi cập nhật.

1. Thêm mới giao dịch
2. Mô tả vắn tắt

* Tên ca sử dụng: Thêm mới giao dịch.
* Mục đích: Giúp nhân viên tư vấn thêm mới giao dịch.
* Tác nhân: Nhân viên tư vấn.

1. Các luồng sự kiện

* Ca sử dụng bắt đầu khi nhân viên tư vấn cần thêm mới 1 giao dịch.
* Nhân viên tư vấn chọn thêm mới.
* Hệ thống hiện thị form thêm mới giao dịch.
* Người dùng nhập thông tin giao dịch mới rồi click chọn thêm mới.

Nếu không nhập đầy đủ thông tin bắt buộc => A1

* Hệ thống thêm mới sản phẩm vào danh sách sản phẩm, cập nhập lại danh sách sản phẩm và hiển thị lên màn hình.
* Luồng sự kiện rẽ nhánh:
* A1: Hiện thị thông báo lỗi: Thông tin giao dịch không hợp lệ.

1. Tiền điều kiện

* Nhân viên tư vấn đang ở trang quản lí giao dịch.

1. Hậu điều kiện

* Hiển thị thông báo và danh sách giao dịch sau khi thêm mới.

1. Cập nhật giao dịch
2. Mô tả vắn tắt

* Tên ca sử dụng: Cập nhật giao dịch.
* Mục đích: Giúp trưởng cửa hàng cập nhập thông tin giao dịch.
* Tác nhân: Trưởng cửa hàng.

1. Các luồng sự kiện

* Ca sử dụng bắt đầu khi trưởng cửa hàng cần cập nhật thông tin một giao dịch.
* Trưởng cửa hàng chọn giao dịch cần cập nhật. Click chọn cập nhật.
* Hệ thống hiện thị form cập nhật giao dịch.
* Trưởng cửa hàng nhập thông tin mới rồi click chọn cập nhật.

Nếu không nhập nhật đầy đủ thông tin => A1

* Hệ thống cập nhập thông tin mới của giao dịch, cập nhập lại danh sách giao dịch và hiển thị lên màn hình.
* Luồng sự kiện rẽ nhánh:
* A1: Hiển thị thông báo lỗi: Thông tin giao dịch không hợp lệ.

1. Tiền điều kiện

* Trưởng cửa hàng đang ở trang quản lí giao dịch.

1. Hậu điều kiện

* Hiển thị thông báo và danh sách giao dịch sau khi cập nhật.

1. Thêm mới người dùng
2. Mô tả vắn tắt

* Tên ca sử dụng: Thêm mới người dùng.
* Mục đích: Giúp trưởng cửa hàng thêm mới người dùng.
* Tác nhân: Trưởng cửa hàng.

1. Các luồng sự kiện

* Ca sử dụng bắt đầu khi trưởng cửa hàng cần thêm mới một người dùng.
* Trưởng cửa hàng chọn thêm mới.
* Hệ thống hiện thị form thêm mới người dùng.
* Trưởng cửa hàng nhập thông tin người dùng mới rồi click chọn thêm mới.

Nếu không nhập đầy đủ thông tin bắt buộc => A1

* Hệ thống thêm mới sản phẩm vào danh sách sản phẩm, cập nhập lại danh sách sản phẩm và hiển thị lên màn hình.
* Luồng sự kiện rẽ nhánh:
* A1: Hiện thị thông báo lỗi: Thông tin người dùng không hợp lệ.

1. Tiền điều kiện

* Trưởng cửa hàng đang ở trang quản lí người dùng.

1. Hậu điều kiện

* Hiển thị thông báo và danh sách người dùng sau khi thêm mới.

1. Cập nhật người dùng
2. Mô tả vắn tắt

* Tên ca sử dụng: Cập nhật người dùng.
* Mục đích: Giúp trưởng cửa hàng cập nhập thông tin người dùng.
* Tác nhân: Trưởng cửa hàng.

1. Các luồng sự kiện

* Ca sử dụng bắt đầu khi trưởng cửa hàng cần cập nhật thông tin một người dùng.
* Trưởng cửa hàng chọn người dùng cần cập nhật thông tin. Click chọn cập nhật.
* Hệ thống hiện thị form cập nhật người dùng.
* Trưởng cửa hàng nhập thông tin mới rồi click chọn cập nhật.

Nếu không nhập đầy đủ thông tin bắt buộc => A1

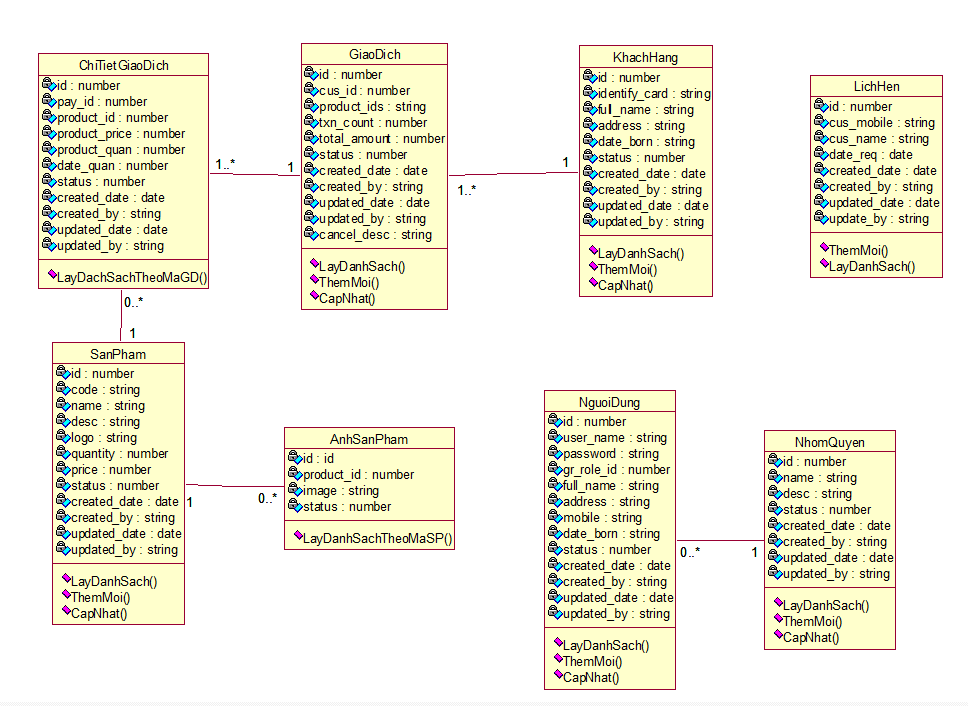
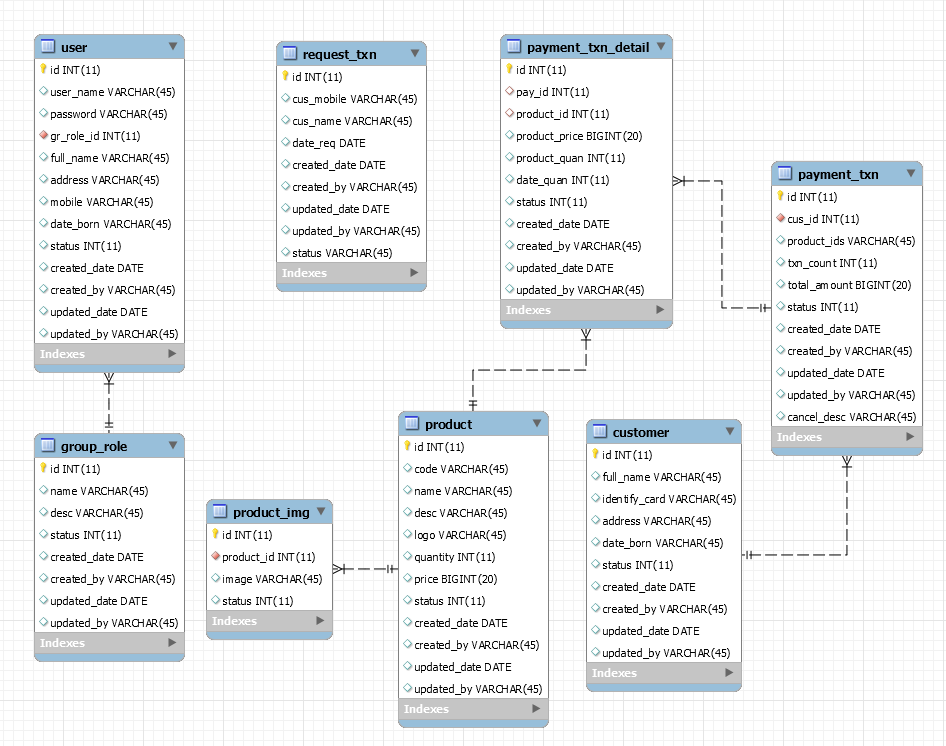
* Hệ thống cập nhập thông tin mới của người dùng, cập nhập lại danh sách người dùng và hiển thị lên màn hình.
* Luồng sự kiện rẽ nhánh:
* A1: Hiển thị thông báo: Thông tin người dùng không hợp lệ.

1. Tiền điều kiện

* Trưởng cửa hàng đang ở trang quản lí người dùng.

1. Hậu điều kiện

* Hiển thị thông báo và danh sách người dùng sau khi cập nhật.

1. Biểu đồ lớp
2. Cở sở dữ liệu
3. Mô hình cở sở dữ liệu­­­
4. Chi tiết bảng dữ liệu
5. Bảng sản phẩm (product)

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Mô tả |
| 1 | ID | NUMBER | Mã định danh |
| 2 | CODE | STRING | Mã sản phẩm |
| 3 | NAME | STRING | Tên sản phẩm |
| 4 | DESC | STRING | Mô tả sản phẩm |
| 5 | LOGO | STRING | Ảnh sản phẩm |
| 6 | QUANTITY | NUMBER | Số lượng sản phẩm |
| 7 | PRICE | NUMBER | Giá thuê / Ngày |
| 8 | STATUS | NUMBER | Trạng thái |
| 9 | CREATED\_DATE | DATE | Ngày tạo |
| 10 | CREATED\_BY | STRING | Người tạo |
| 11 | UPDATED\_DATE | DATE | Ngày cập nhật |
| 12 | UPDATED\_ BY | STRING | Người cập nhật |
| 13 | PRICE\_IMPORT | NUMBER | Giá nhập |
| 14 | PRICE\_EXPORT | NUMBER | Gía bán |

1. Bảng hình ảnh sản phẩm (product\_img)

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Mô tả |
| 1 | ID | NUMBER | Mã định danh |
| 2 | PRODUCT\_ID | NUMBER | Mã sản phẩm |
| 3 | IMAGE | STRING | Link ảnh |
| 4 | STATUS | NUMBER | Trạng thái |

1. Bảng lịch hẹn (request\_txn)

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Mô tả |
| 1 | ID | NUMBER | Mã định danh |
| 2 | CUS\_MOBILE | STRING | Số điện thoại khách hàng |
| 3 | CUS\_NAME | STRING | Tên khách hàng |
| 4 | DATE\_REQ | DATE | Thời gian hẹn |
| 5 | CREATED\_DATE | DATE | Ngày tạo |
| 6 | UPDATED\_DATE | DATE | Ngày cập nhật |
| 7 | UPDATE\_BY | STRING | Người cập nhật |
| 8 | STATUS | NUMBER | Trạng thái |

1. Bảng khách hàng (customer)

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Mô tả |
| 1 | ID | NUMBER | Mã định danh |
| 2 | IDENTIFY\_CARD | STRING | Số CMND, CCCD |
| 3 | MOBILE | STRING | Số điện thoại khách hàng |
| 4 | FULL\_NAME | STRING | Tên khách hàng |
| 5 | ADDRESS | STRING | Địa chỉ |
| 6 | DATE\_BORN | STRING | Ngày sinh |
| 7 | STATUS | NUMBER | Trạng thái |
| 8 | CREATED\_DATE | DATE | Ngày tạo |
| 9 | CREATED\_BY | STRING | Người tạo |
| 10 | UPDATED\_DATE | DATE | Ngày cập nhật |
| 11 | UPDATED\_ BY | STRING | Người cập nhật |

1. Bảng giao dịch (payment\_txn)

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Mô tả |
| 1 | ID | NUMBER | Mã định danh |
| 2 | CUS\_ID | NUMBER | Tham chiếu ID bàng khách hàng |
| 3 | PRODUCT\_IDS | STRING | Danh sách ID sản phẩm |
| 4 | TXN\_COUNT | NUMBER | Tổng số lượng sản phẩm |
| 6 | TOTAL\_AMOUNT | NUMBER | Tổng số tiền thanh toán |
| 7 | STATUS | NUMBER | Trạng thái  1: Khởi tạo  2: Chờ thanh toán  3: Đã thanh toán  4: Đã hủy |
| 8 | CREATED\_DATE | DATE | Ngày tạo |
| 9 | CREATED\_BY | STRING | Người tạo |
| 10 | UPDATED\_DATE | DATE | Ngày cập nhật |
| 11 | UPDATED\_ BY | STRING | Người cập nhật |
| 12 | CANCEL\_DESC | STRING | Lý do hủy |

1. Bảng chi tiết giao dịch (payment\_txn\_detail)

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Mô tả |
| 1 | ID | NUMBER | Mã định danh |
| 2 | PAY\_ID | NUMBER | Tham chiếu ID bảng giao dịch |
| 3 | PRODUCT\_ID | NUMBER | Tham chiếu ID bảng sản phẩm |
| 4 | PRODUCT\_PRICE | NUMBER | Giá tiền |
| 5 | PRODUCT\_QUAN | NUMBER | Số lượng sản phẩm |
| 6 | DATE\_QUAN | NUMBER | Số ngày thuê |
| 7 | STATUS | NUMBER | Trạng thái  1: use  2: cancel |
| 8 | CREATED\_DATE | DATE | Ngày tạo |
| 9 | CREATED\_BY | STRING | Người tạo |
| 10 | UPDATED\_DATE | DATE | Ngày cập nhật |
| 11 | UPDATED\_ BY | STRING | Người cập nhật |

1. Bảng người dùng (user)

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Mô tả |
| 1 | ID | NUMBER | Mã định danh |
| 2 | USER\_NAME | STRING | Số CMND, CCCD |
| 3 | PASSWORD | STRING | Mật khẩu |
| 4 | GR\_ROLE\_ID | NUMBER | Mã nhóm quyền |
| 5 | FULL\_NAME | STRING | Tên người dùng |
| 6 | MOBILE | STRING | Số điện thoại |
| 7 | ADDRESS | STRING | Địa chỉ |
| 8 | DATE\_BORN | STRING | Ngày sinh |
| 9 | STATUS | NUMBER | Trạng thái |
| 10 | CREATED\_DATE | DATE | Ngày tạo |
| 11 | CREATED\_BY | STRING | Người tạo |
| 12 | UPDATED\_DATE | DATE | Ngày cập nhật |
| 13 | UPDATED\_ BY | STRING | Người cập nhật |

1. Bảng nhóm quyền (group\_role)

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Mô tả |
| 1 | ID | NUMBER | Mã định danh |
| 2 | NAME | STRING | Tên nhóm quyền |
| 3 | DESC | STRING | Mô tả |
| 4 | STATUS | NUMBER | Trạng thái |
| 5 | CREATED\_DATE | DATE | Ngày tạo |
| 6 | CREATED\_BY | STRING | Người tạo |
| 7 | UPDATED\_DATE | DATE | Ngày cập nhật |
| 8 | UPDATED\_ BY | STRING | Người cập nhật |

# CHƯƠNG 3: CÀI ĐẶT VÀ TRIỂN KHAI

1. Giới thiệu về công cụ và môi trường cài đặt
2. Giao diện người dùng
3. Màn hình trang chủ
4. Màn hình danh sách sản phẩm
5. Màn hình đặt lịch hẹn
6. Giao diện quản trị viên
7. Màn hình đăng nhập
8. Màn hình danh sách người dùng
9. Màn hình thêm mới người dùng
10. Màn hình cập nhật người dùng
11. Màn hình danh sách sản phẩm
12. Màn hình thêm mới sản phẩm
13. Màn hình cập nhật sản phẩm
14. Màn hình danh sách khách hàng
15. Màn hình thêm mới khách hàng
16. Màn hình cập nhật khách hàng
17. Màn hình danh sách đặt lịch hẹn
18. Màn hình danh sách giao dịch
19. Màn hình tạo mới giao dịch
20. Màn hình cập nhật sản phẩm
21. Báo cáo giao dịch thành công

# KẾT LUẬN

Sau một thời gian nghiên cứu và xây dựng đề tài “ Xây dựng website quản lý cửa hàng Áo Cưới Thủy Linh”, em đã phân tích, thiết kế và hoàn thành về cơ bản theo đúng những yêu cầu về nội dung và thời gian đã định.

Trong quá trình nghiên cứu và thực hiện đề tài vẫn gặp nhiều hạn chế về kinh nghiệm và kiến thức nên đề tài của em không thể tránh khỏi những thiếu xót. Em rất mong nhận được ý kiến đóng góp từ quý thầy cô và các bạn để đề tài được hoàn thiện hơn.

**Kết quả đạt được**

* Về mặt công nghệ, em đã tìm hiểu và nắm bắt quy trình để xây dựng một website, tìm hiểu cách sử dụng intelij idea, Visual Code.
* Hoàn thiện khảo sát, phân tích, thiết kế website “Xây dựng website quản lý cửa hàng Áo Cưới Thủy Linh”
* Lập trình thành thạo với Spring Framework.

**Hạn chế của đề tài**

* Do thời gian làm có hạn lên em chưa thể hoàn thiện đầy đủ chức năng của chương trình, cũng như xử lý tất cả các ngoại lệ xảy ra trong quá trình xử dụng ứng dụng.
* Chưa kiểm soát được tất cả các lỗi có thể xảy ra trong quá trình sử dụng.

**Hướng phát triển**

* Hoàn thiện giao diện và các chức năng cho sản phẩm (chưa hoàn thành).
* Cải thiện tốc độ truy xuất dữ liệu khi có nhiều người cùng truy cập.
* Em rất mong tiếp tục nhận được sự giúp đỡ và tạo điều kiện của quý thầy cô và nhà trường để em có cơ hội phát triển, hoàn thiện ứng dụng tốt hơn trong thời gian tới.

# TÀI LIỆU THAM KHẢO

<https://freetuts.net/java-la-gi-gioi-thieu-ngon-ngu-java-1023.html>

<https://vi.wikipedia.org/wiki/Spring_Framework>

<http://laptrinhjavaweb.com/trang-chu>