# MỤC LỤC

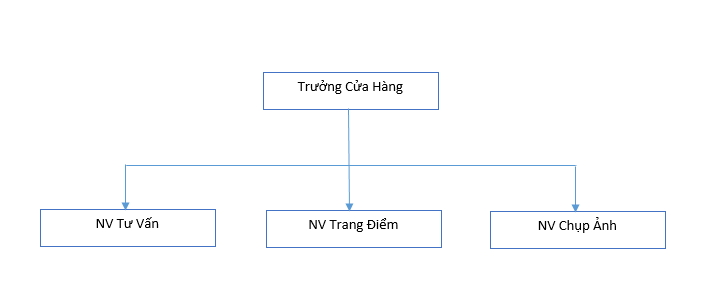
# CHƯƠNG 1: TỔNG QUAN

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

* 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.
* Vì sao ngôn ngữ này lại được đặt tên là Java: 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.
* Ứng dụng của Java:
* 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. Giới thiệu về Spring Framework
* 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).
* 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.
* 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). 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.
* 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. Giới thiệu về Spring Boot
* Giới thiệu SpringBoot:
* 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 câu config XML…
* Cung cấp nhiều plugin.
* Chuẩn cho Microservices (Cloud support; giảm việc setup, config; các thư viện hỗ trợ…).
* Các tính năng nối 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...

# 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. Đặc tả usecase
3. Xem trang chủ
4. Xem danh sách sản phẩm
5. Đăng kí lịch hẹn
6. Thêm mới khách hàng
7. Cập nhật khách hàng
8. Thêm mới sản phẩm
9. Cập nhật sản phẩm
10. Thêm mới giao dịch
11. Cập nhật giao dịch
12. Thêm mới người dùng
13. Cập nhật người dùng
14. Biểu đồ lớp
15. Cở sở dữ liệu
16. Mô hình cở sở dữ liệu
17. Chi tiết bảng dữ liệu

# 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

# TÀI LIỆU THAM KHẢO