

GVHD: Thầy NGUYỄN THANH PHƯỚC

Lớp UD14306 – Nhóm SG-Rose

|  |  |  |
| --- | --- | --- |
| Mã | Họ và tên | Vai trò |
| 1 | Doanh Ngọc Diễm | Trưởng nhóm |
| 2 | Trần Công Truyền | Thành viên |
| 3 | Nguyễn Đức Hòa | Thành viên |
| 4 | Hồ Phi Long | Thành viên |
| 5 | Trần Đại Hưng | Thành viên |

**BÁO CÁO DỰ ÁN TỐT NGHIỆP**

**KINH DOANH HOA SG-ROSE**

Ngành: Ứng dụng phần mềm

TP.HCM 24-12-2020

TRƯỜNG CAO ĐẲNG THỰC HÀNH FPT

----- 🕮 -----

**MỤC LỤC**

[DANH MỤC VIẾT TẮT 4](#_Toc58418777)

[LỜI MỞ ĐẦU 5](#_Toc58418778)

[NHẬN XÉT CỦA GV HƯỚNG DẪN 6](#_Toc58418779)

[NHẬN XÉT CỦA GV PHẢN BIỆN 7](#_Toc58418780)

[I. GIỚI THIỆU ĐỀ TÀI 8](#_Toc58418781)

[**1.** **Tên đề tài:** 8](#_Toc58418782)

[**2.** **Lý do chọn đề tài:** 8](#_Toc58418783)

[**a.** **Phân tích nhu cầu Website:** 8](#_Toc58418784)

[**b.** **Tầm ảnh hưởng của Website:** 8](#_Toc58418785)

[**c.** **Nghiên cứu tính khả thi của Website:** 8](#_Toc58418786)

[II. KHẢO SÁT YÊU CẦU 11](#_Toc58418787)

[**1.** **Mục tiêu – Phạm vi đồ án:** 11](#_Toc58418788)

[**a.** **Mục tiêu:** 11](#_Toc58418789)

[**b.** **Phạm vi đồ án:** 11](#_Toc58418790)

[**2.** **Mô tả yêu cầu đặt ra:** 11](#_Toc58418791)

[**a.** **Mô tả yêu cầu:** 11](#_Toc58418792)

[**b.** **Nhiệm vụ cơ bản:** 11](#_Toc58418793)

[**c.** **Yêu cầu đặt ra:** 12](#_Toc58418794)

[III. PHÂN TÍCH HIỆN TRẠNG 13](#_Toc58418795)

[**1.** **Thiết kế Use Case Diagram:** 13](#_Toc58418796)

[**2.** **Mô tả Actor:** 17](#_Toc58418797)

[**3.** **Đặt tả Use Case:** 17](#_Toc58418798)

[IV. THIẾT KẾ HỆ THỐNG 43](#_Toc58418799)

[**1.** **Phác thảo mô hình ứng dụng:** 43](#_Toc58418800)

[**a.** **Phác thảo mô hình tổng quan:** 43](#_Toc58418801)

[**b.** **Phác thảo mô hình ứng dụng web theo MVC:** 44](#_Toc58418802)

[**2.** **Phác thảo giao diện:** 45](#_Toc58418803)

[**3.** **Phác thảo ERD:** 50](#_Toc58418804)

[V. THỰC HIỆN DỰ ÁN 51](#_Toc58418805)

[**1.** **Thiết kế CSDL:** 51](#_Toc58418806)

[**2.** **Thiết kế layout, thiết kế giao diện chi cho các chức năng:** 54](#_Toc58418807)

[**3.** **Thiết kế thư viện cần thiết cho dự án:** 66](#_Toc58418808)

[**4.** **Thiết kế mô hình triển khai theo mô hình MVC:** 67](#_Toc58418809)

[VI. KIỂM THỬ 82](#_Toc58418810)

[**1.** **Thiết kế CSDL:** 82](#_Toc58418811)

[**a.** **Khảo sát rủi ro:** 82](#_Toc58418812)

[**b.** **Phạm vi:** 82](#_Toc58418813)

[**c.** **Nguồn lực:** 82](#_Toc58418814)

[**2.** **Test case và kết quả test:** 82](#_Toc58418815)

[VII. HƯỚNG DẪN TRIỂN KHAI VÀ SỬ DỤNG 89](#_Toc58418816)

[**1.** **Hướng dẫn cài đặt:** 89](#_Toc58418817)

[**2.** **Hướng dẫn sử dụng:** 89](#_Toc58418818)

[VIII. KẾT LUẬN 91](#_Toc58418819)

[**1.** **Kết luận:** 91](#_Toc58418820)

[**2.** **Khó khăn và thuận lợi:** 91](#_Toc58418821)

[1. Thuận lợi: 91](#_Toc58418822)

[2. Khó khăn: 91](#_Toc58418823)

[**3.** **Hướng phát triển:** 92](#_Toc58418824)

[LỜI CẢM ƠN 93](#_Toc58418825)

[TÀI LIỆU THAM KHẢO 94](#_Toc58418826)

[**1.** **Tiếng việt:** 94](#_Toc58418827)

[**2.** **Tiếng anh:** 94](#_Toc58418828)

# DANH MỤC VIẾT TẮT

* SP: Sản phẩm
* CSDL: Cơ sở dữ liệu
* HTML: HyperText Markup Language
* CSS: Cascading Style Sheets
* XML: Extensible Markup Language
* UC: Use Case
* UI: User Interface
* JSON: JavaScript Object Noattion
* N/A: No Answer
* MVC: Model – View – Controller

# LỜI MỞ ĐẦU

Bán hoa online, mở tiệm hoa online đang là hình thức kinh doanh mới mẻ được nhiều người quan tâm. Kinh doanh hoa online mang lại hiệu quả rất tốt cùng với sự phát triển của công nghệ giúp mua hoa online trở nên dễ dàng và nhanh chóng.

Hiện nay, bán hoa chủ yếu đều dựa vào hình thức online. Thực tế cho thấy bán hoa offline doanh thu kém hơn online. Đối với đối tượng chị em phụ nữ hoặc các sự kiện khai trương thì họ tự tìm đến tiệm hoa để xem loại đẹp và tươi, và cũng có khi phải lên web tìm đến loại hoa nào đó vì tiệm hoa offline không có. Đối với những loại hoa dành cho đám tang thì họ thường có nhu cầu mua hoa trên mạng. Do đó, chúng ta có thể thấy hình thức kinh doanh online vô cùng hiệu quả, dễ tiếp cận được nhiều khách hàng hơn chúng ta vẫn nghĩ.

Để thực thi kế hoạch kinh doanh hoa online cần phải có kênh để bán hàng online. Hiện nay có 2 kênh phù hợp cho việc bán hoa online là bán trên website và bán trên mạng xã hội. Thiết kế website là cách tốt nhất để bán hàng online đối với bất cứ mặt hàng nào. Quan trọng, website đó đòi hỏi phải đẹp mắt, có công cụ hỗ trợ mua hàng, thanh toán tiện lợi, tốc độ tải nhanh, màu sắc đẹp mắt, logo thương hiệu ấn tượng dễ nhận biết, nội dung hình ảnh hấp dẫn,… sẽ nâng tầm chuyên nghiệp cho thương hiệu.

Từ những ý tưởng và vấn đề được đặt ra, cùng với vận dụng vốn kiến thức và đầu tư công sức của cả nhóm. “Website bán hoa” được tạo ra để giải quyết những vấn đề trên. Với những chức năng mà website có, hi vọng rằng website sẽ đạt được thành công nhất định.

# NHẬN XÉT CỦA GV HƯỚNG DẪN

……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………........................................................................................................

# NHẬN XÉT CỦA GV PHẢN BIỆN

……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………........................................................................................................

# GIỚI THIỆU ĐỀ TÀI

## **Tên đề tài:**

“Xây dựng Website kinh doanh hoa sử dụng Spring MVC và Bootstrap”.

## **Lý do chọn đề tài:**

### **Phân tích nhu cầu Website:**

* Trong thời đại công nghệ thông tin như hiện nay, khi mà Internet trở nên thân quen và dần trở thành một công cụ không thể thiếu trong cuộc sống thì lợi ích của Website đối với việc quảng bá sản phẩm hoa và thương hiệu của một công ty thật là to lớn.
* Tùy từng lĩnh vực kinh doanh của từng doanh nghiệp mà Website mang lại những lợi ích khác nhau. Trang Web trở thành một cửa ngõ để doanh nghiệp tiếp thị sản phẩm của mình đến khách hàng khắp nơi trên toàn thế giới.
* Dự án “Xậy dựng Website kinh doanh hoa SG-Rose” tạo ra Website kinh doanh hoa toàn diện, chi phí thiết kế thấp, giao diện khách hàng đẹp mắt, việc quản lý trở nên dễ dàng, tối ưu hóa cơ sở dữ liệu.

### **Tầm ảnh hưởng của Website:**

* Website là nơi doanh nghiệp và khách hàng gặp nhau. Khi họ truy cập cập vào Website của bạn và tìm thấy những thông tin về sản phẩm của bạn cung cấp, khi đó, Website trở thành một kênh tư vấn khách hàng trực tuyến hữu hiệu.
* Website là một phương tiện để giới thiệu doanh nghiệp, quảng bá, marketing và PR. Cho nên, việc chọn hướng kinh doanh hoa online trên Website là hoàn toàn đúng đắn và phù hợp với hướng phát triển của cả nhóm.

### **Nghiên cứu tính khả thi của Website:**

* Khả thi về tổ chức:
  + Giúp cơ hội kinh doanh mở rộng: Cơ hội quảng bá không giới hạn, công việc kinh doanh mở cửa 24h/1ngày. Website của bạn chính là một văn phòng giao dịch kinh doanh toàn cầu luôn mở cửa 24h/ngày, 7 ngày/tuần và 365 ngày/năm. Bất kỳ ai, dù ở đâu đều có thể ghé thăm văn phòng của bạn.
  + Giảm tối thiểu thời gian quản lý, tối ưu hóa việc vận hành hệ thống.
* Khả thi về kinh tế:
  + Hãy tưởng tượng, thay vì phải gửi Catalogue bằng giấy hoặc đĩa CDROM cho khách hàng. Bạn có thể mời khách hàng đến thăm Website của mình và những thông tin còn cặn kẽ và đầy đủ hơn ở Catalogue.
  + Bạn chẳng phải trả lương cho ai cả, mọi thứ ở văn phòng giao dịch toàn cầu của bạn đều thực hiện một cách chính xác và hoàn toàn tự động. Chi phí để có một Website thật là quá nhỏ bé so với những chi phí mà bạn bỏ ra để quảng bá các hoạt động của bạn bằng những phương tiện truyền thống.
* Khả thi về kỹ thuật:
  + Sử dụng ngôn ngữ lập trình Java/ SQLServer thiết kế Website, ngôn ngữ dễ sử dụng và được rất nhiều doanh nghiệp lựa chọn.
  + Ngôn ngữ lập trình Java đã được phát triển bởi công ty Sun Microsystem vào năm 1995. Năm 2010, Sun Microsystem đã được Oracle mua lại, vì thế hiện nay Java thuộc sở hữu của Oracle. Trong hơn 24 năm hình thành và phát triển, Java đã trở thành nền tảng của hầu hết các ứng dụng ngày nay. Nó đã chứng minh được giá trị của mình trong việc phát triển các phần mềm doanh nghiệp, ứng dụng di động, phát triển website, game và cả hệ thống nhúng. Java đã được sử dụng rất nhiều trong các ứng dụng ngân hàng / tài chính trên khắp thế giới. Nó cũng được các doanh nghiệp lớn như Amazon, Google và Facebook coi là một trong những ngôn ngữ tốt nhất để duy trì và phát triển website dành cho doanh nghiệp.
* SQL là ngôn ngữ phi thủ tục, không yêu cầu cách thức truy cập cơ sở dữ liệu như thế nào. Tất cả các thông báo của SQL rất dễ dàng sử dụng và ít mắc lỗi. Vì tầm quan trọng của database ngày càng tăng lên, Relational Database Management Systems (RDBMS) cũng trở nên quan trọng hơn. Đối tượng của SQL là các bảng dữ liệu với các cột và các hàng. Cột được gọi là trường dữ liệu và hàng là bản ghi của bảng. Cột dữ liệu và kiểu dữ liệu xác định tạo nên cấu trúc của bảng. Khi bảng được tổ chức thành một hệ thống cho một mục đích sử dụng cụ thể vào công việc nào đó sẽ trở thành một cơ sở dữ liệu. SQL Server, cũng được gọi là Microsoft SQL Server, đã tồn tại thậm chí còn lâu hơn MySQL. Microsoft phát triển SQL Server từ giữa thập niên 80s, với lời hứa RDBMS cung cấp giải pháp đáng tin cậy và có thể mở rộng. Việc này trở thành đặc tính đáng giá của SQL Server cho tới tận bây giời, vì nó là một nền tảng được chọn cho những phần mềm doanh nghiệp cần mở rộng lớn theo thời gian.
* **Bootstrap** là một bộ sưu tập miễn phí của các **mã nguồn mở** và công cụ dùng để tạo ra một mẫu webiste hoàn chỉnh. Với các thuộc tính về giao diện được quy định sẵn như kích thước, màu sắc, độ cao, độ rộng…, các **designer** có thể sáng tạo nhiều sản phẩm mới mẻ nhưng vẫn tiết kiệm thời gian khi làm việc với **framework** này trong quá **trình thiết kế giao diện website.**
* Vào năm 2002, Spring Framework phát hành phiên bản đầu tiên bởi Rod Johnson. Việc xây dựng các ứng dụng doanh nghiệp trở nên đơn giản và dễ dàng hơn. Rod Johnson đã cho xuất bản một cuốn sách được phổ biến rộng rãi có nhan đề "J2EE Development without EJB". Điều này đã cho Spring Framework rất nhiều động lực để thay thế EJB. Theo đà phát triển đó, Spring đã trở thành framework mã nguồn mở phổ biến nhất để xây dựng các ứng dụng doanh nghiệp. Cách tiếp cận thực tế ban đầu của Rod Johnson tiếp tục được phát triển và hướng tới một bộ công cụ hoàn chỉnh dành cho xây dựng các ứng dụng doanh nghiệp. Theo một số nguồn, trên 50% các ứng dụng web Java hiện nay đang sử dụng Spring.
* Hibernate framework là một giải pháp ORM (Object Relational Mapping) mã nguồn mở, gọn nhẹ. Hibernate giúp đơn giản hoá sự phát triển của ứng dụng java để tương tác với cơ sở dữ liệu. Tool ORM giúp đơn giản hoá việc tạo ra dữ liệu, thao tác dữ liệu và truy cập dữ liệu. Đó là một kỹ thuật lập trình để ánh xạ đối tượng vào dữ liệu được lưu trữ trong cơ sở dữ liệu.
* JDBC (Java Database Connectivity) là một chuẩn API (Application Program Interface) cho phép kết nối các chương trình viết bởi Java với các hệ quản trị cơ sở dữ liệu (MySQL, MS SQL, Postgre SQL, Oracle, DB2…)

# KHẢO SÁT YÊU CẦU

## **Mục tiêu – Phạm vi đồ án:**

### **Mục tiêu:**

* Nội dung trang bán hoa hấp dẫn, thông tin đầy đủ, phù hợp với mọi đối tượng sử dụng. Hoa và thông tin hoa được sắp xếp, bố cục một cách hợp lý, tạo điều kiện cho người đọc dễ định hướng trong Website.
* Hình thức đẹp phù hợp với chủ đề, nội dung, thân thiện với người dùng, trang Web phải được thiết kế responsive để phù hợp với tất cả các thiết bị, mọi độ phần giải màn hình.
* Nội dung trang quản lý của Admin và nhân viên phải bố cục gọn gòn, trực quan, hiển thị đầy đủ thông tin, dữ liệu cần thiết, các chức năng cần thực hiện chính xác và mượt mà.

### **Phạm vi đồ án:**

* Phạm vi thực hiện đồ án dựa trên những kiến thức thu được từ những môn học trong quá trình học tập tại trường và tham khảo thêm một số tài liệu bên ngoài để bổ sung kiến thức nhằm hoàn thiện đồ án tốt hơn, đáp ứng được mục tiêu đồ án.
* Ứng dụng này khi phát triển hoàn chỉnh sẽ tăng hiệu quả kinh doanh, giảm chi phí quảng cáo, quản lý dễ dàng.

## **Mô tả yêu cầu đặt ra:**

### **Mô tả yêu cầu:**

* Hệ thống Website kinh doanh hoa là một hệ thống web nhằm giới thiệu và quảng bá về hoa – thông tin hoa, cũng như đáp nhu cầu mua hoa của khách hàng. Giúp quảng bá hình ảnh của doanh nghiệp thông qua Website.
* Hệ thống sẽ giúp khách hàng chọn lựa những loại hoa ưng ý nhất, tạo tài khoản để tương tác. Giúp nội bộ có thể quản lý mọi việc một cách dễ dàng và chi tiết từ khâu vận hành cho tới giao dịch.
* Hệ thống website phải có những tính năng cơ bản của một website thông thường.

### **Nhiệm vụ cơ bản:**

* Giới thiệu thương hiệu và hình ảnh của doanh nghiệp bán hoa.
* Giới thiệu thông tin về hoa và bán hoa.
* Tìm kiếm loại hoa theo yêu cầu khách hàng.
* Hỗ trợ, tư vấn khách hàng khi mua hoa.
* Giúp quản lý thành viên và thống kê cho Admin
* Giúp quản lý và thống kê hoa cho nhân viên

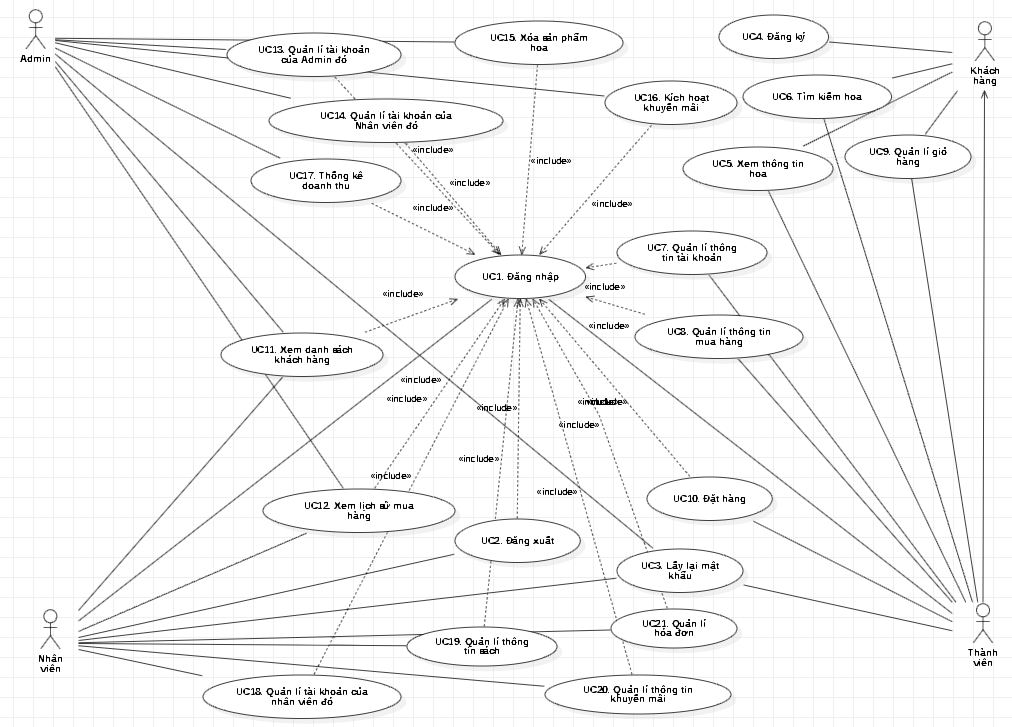
### **Yêu cầu đặt ra:**

* Yêu cầu về thiết bị và phầm mềm của nhóm làm dự án:
  + Laptop hoặc pc: Sử dụng hệ điều hành.
  + Hệ quản trị cơ sở dữ liệu: SQLServer.
  + Phần mềm lập trình: NetBeans IDE 8.2 RC
  + Ngôn ngữ lập trình: Back-end: Java, Font-end: Bootstrap.
  + Các phần mềm phụ trợ: Mircrosoft Office, Zalo,...
* Yêu cầu chứng năng nghiệp vụ của hệ:
  + Người dùng chưa đăng nhập:
    - Xem tất cả danh mục, loại sách, khuyến mãi, tin tức.
    - Xem thông tin hoa.
    - Tìm kiếm hoa.
    - Quản lý giỏ hàng.
    - Đăng ký thành viên.
  + Thành viên:
    - Các chức năng của người dùng chưa đăng nhập.
    - Quản lý thông tin tài khoản.
    - Quản lý giỏ hàng.
    - Đặt hàng.
    - Đăng xuất.
  + Nhân viên:
    - Đăng nhập.
    - Quản lý thông tin tài khoản của nhân viên đó.
    - Xem danh sách khách hàng.
    - Xem lịch sử mua hoa.
    - Quản lý thông tin hoa.
    - Quản lý thông tin khuyến mãi.
    - Quản lý đơn hàng.
    - Đăng xuất.
  + Admin:
    - Đăng nhập.
    - Quản lý thông tin tài khoản của admin đó.
    - Quản lý tài khoản nhân viên.
    - Xem danh sách khách hàng.
    - Xem lịch sử mua hàng.
    - Xem thông tin hoa.
    - Kích hoạt khuyến mãi.
    - Thống kê doanh thu theo ngày, tháng, năm.
    - Đăng xuất.

# PHÂN TÍCH HIỆN TRẠNG

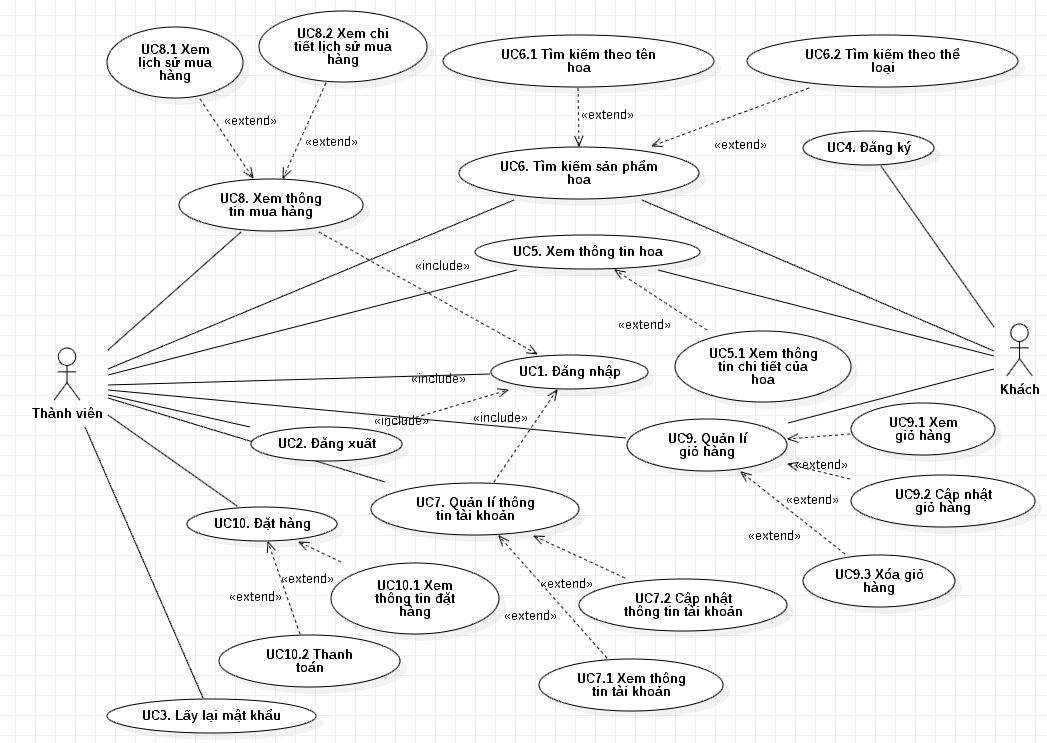
## **Thiết kế Use Case Diagram:**

1. Use Case tổng quan hệ thống:



1. Chi tiết Use Case:

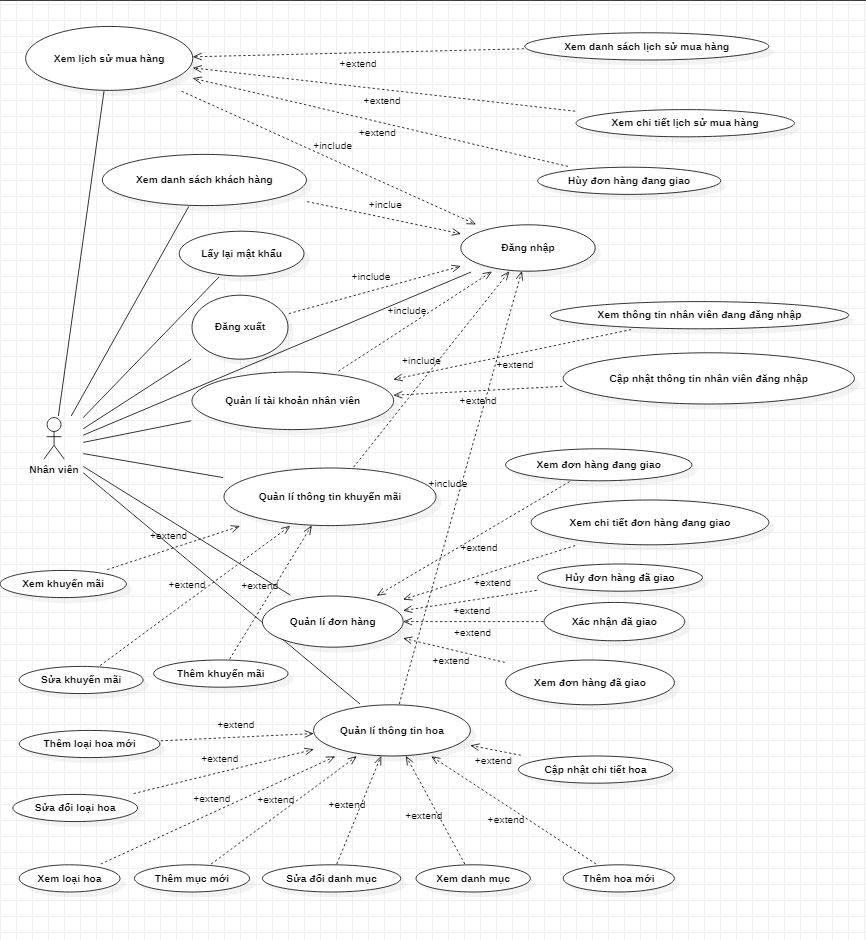
* Chi tiết Use Case cho Khách và Thành viên:



* Chi tiết Use Case dành cho Khách và Thành viên:



* Use Case dành cho nhân viên, admin:



## **Mô tả Actor:**

|  |  |  |
| --- | --- | --- |
| **Mã** | **Actor** | **Mô tả** |
| AT01 | Khách | Người dùng chưa đăng nhập. |
| AT02 | Thành viên | Khách hàng đăng nhập vào Website. |
| AT03 | Nhân viên | Nhân viên quản lý của Website. |
| AT04 | Admin | Quản trị viên của Website. |

## **Đặt tả Use Case:**

1. Bảng chú thích:

|  |  |
| --- | --- |
| **Name** | Tên Use Case |
| **ID** | Mã Use Case |
| **Description** | Tóm gọn sự tương tác được thể hiện trong Use Case |
| **Actor** | Những đối tượng thực hiện sự tương tác Use Case |
| **Pre-condition** | Điều kiện cần để Use Case thực hiện thành công |
| **Post-condition** | Những thứ sẽ xuất hiện sau khi Use Case được thưc hiện |
| **Trigger** | Điều kiện kích hoạt Use Case xảy ra |
| **Basic flow** | Luồng tương tác **chính** giữa Actor và System để Use Case được **thực hiện thành công**. |
| **Alternative flow** | Luồng tương tác **thay thế** giữa các Actor và System để Use Case **thực hiện thành công**. |
| **Exception flow** | Luông tương tác **ngoại lệ** giữa Actor và System mà Use Case **thực hiện thất bại**. |

1. Đặc tả chi tiết Use Case:

* Use Case 1 – Đăng nhập

|  |  |
| --- | --- |
| **Name** | Đăng nhập |
| **ID** | UC1 |
| **Description** | Chức năng cho phép actor đăng nhập tài khoản. |
| **Actor** | Thành viên, Nhân viên, Admin. |
| **Pre-condition** | Actor đã có tài khoản tạo sẵn |
| **Post-condition** | Nếu đăng nhập thành công – truy cập vào Website. |
| **Trigger** | Actor nhấn [Đăng nhập] |
| **Basic flow** | 1. Bấm vào nút [Đăng nhập] – Thành viên, vào trang Login Admin – Admin, trang Login Nhanvien – Nhân viên.  2. Hiển thị màng hình đăng nhập.  3. Actor nhập email, password.  4. Nhấn nút [Login] hoặc nhấn Enter. |
| **Alternative flow** | N/A |
| **Exception flow** | Nếu thất bại – thông báo lỗi. |

* Use Case 2 – Đăng xuất

|  |  |
| --- | --- |
| **Name** | Đăng xuất. |
| **ID** | UC2 |
| **Description** | Cho phép actor đăng xuất khỏi tài khoản. |
| **Actor** | Thành viên, Nhân viên, Admin. |
| **Pre-condition** | Đăng nhập vào Website. |
| **Post-condition** | Đăng xuất tài khoản – Quay lại trang Login. |
| **Trigger** | Actor nhấn [Đăng xuất] |
| **Basic flow** | 1. Actor bấm vào nút [Đăng xuất].  2. Xác nhận đăng xuất. |
| **Alternative folow** | N/A |
| **Exception flow** | Hủy xác nhận đăng xuất – không đăng xuất. |

* Use Case 3 – Lấy lại mật khẩu

|  |  |
| --- | --- |
| **Name** | Lấy lại mật khẩu |
| **ID** | UC3 |
| **Description** | Gửi lại mật khẩu tài khoản hiện tại cho actor. |
| **Actor** | Thành viên, Nhân viên, Admin |
| **Pre-condition** | Actor có tài khoản tạo sẵn.  Actor dùng email tài khoản để nhân mật khẩu. |
| **Post-condition** | Mật khẩu được gửi vào email đăng ký tài khoản của actor.  Nếu không phải email đăng ký thì sẽ không gửi. |
| **Trigger** | Actor nhấn [Gửi mật khẩu] |
| **Basic flow** | 1. Nhấn vào nút [Quên mật khẩu] hoặc [Thay đổi mật khẩu].  2. Điền địa chỉ email tài khoản.  3. Nhấn [Gửi]  3. Kiểm tra email để xem mật khẩu. |
| **Alternative flow** | N/A |
| **Exception flow** | Nếu không đúng email – thông báo lỗi. |

* Use Case 4 – Đăng ký

|  |  |
| --- | --- |
| **Name** | Đăng ký |
| **ID** | UC4 |
| **Description** | Cho phép Khách hàng tạo tài khoản thành viên. |
| **Actor** | Khách. |
| **Pre-condition** | N/A |
| **Post-condition** | Tài khoản thành viên mới được tạo. |
| **Trigger** | Khách nhấn [Đăng ký] |
| **Basic flow** | 1. Khách bấm vào nút [Đăng ký] trên header.  2. Nhập thông tin tài khoản.  3. Bấm nút [Đăng ký].  4. Thông báo kết quả đăng ký |
| **Alternative flow** | N/A |
| **Exception** | Nhập thông tin không đúng định dạng – thông báo lỗi  Đăng ký không thành công – thông báo lỗi |

* Use Case 5 – Xem thông tin hoa

|  |  |
| --- | --- |
| **Name** | Xem thông tin hoa. |
| **ID** | UC5 |
| **Description** | Cho actor xem thông tin hoa. |
| **Actor** | Khách, Thành viên. |
| **Pre-condition** | N/A |
| **Post-condition** | Hiển thị thông tin hoa và thông tin trang web |
| **Trigger** | Actor truy cập vào trang bán hoa |
| **Basic flow** | 1. Truy cập vào trang bán hoa. |
| **Alternative flow** | N/A |
| **Exception** | Không load thì hiển thị “không có hoa nào” |

* Use Case 5.1 – Xem chi tiết thông tin sách

|  |  |
| --- | --- |
| **Name** | Xem chi tiết thông tin hoa |
| **ID** | UC5.1 |
| **Description** | Cho Khách hàng xem chi tiết thông tin hoa. |
| **Actor** | Khách, Thành viên. |
| **Pre-condition** | N/A |
| **Post-condition** | Hiển thị tiết thông tin hoa được chọn |
| **Trigger** | Actor nhấn [Xem chi tiết] |
| **Basic flow** | 1. Nhấn vào nút [Xem chi tiết].  2. Trang chi tiết sách hiển thị hoa được chọn. |
| **Alternative flow** | N/A |
| **Exception** | Dữ liệu không load – dùng image thấy thế |

* Use Case 6 – Tìm kiếm hoa
* Use Case 6.1 – Tìm kiếm theo tên hoa

|  |  |
| --- | --- |
| **Name** | Tìm kiếm sách theo tên hoa. |
| **ID** | UC6.1 |
| **Description** | Cho Khách hàng tìm kiếm sách theo tên hoa |
| **Actor** | Khách, Thành viên. |
| **Pre-condition** | N/A |
| **Post-condition** | Hoa tìm kiếm tồn tại sẽ hiển thị lên trang. |
| **Trigger** | Actor nhấn [Tìm kiếm] |
| **Basic flow** | 1. Click vào ô [Tìm kiếm].  2. Nhập tên hoa cần tìm.  3. Nhấn nút [Tìm kiếm] hoặc Enter. |
| **Alternative flow** | N/A |

* Use Case 6.2 – Tìm kiếm theo thể loại

|  |  |
| --- | --- |
| **Name** | Tìm kiếm hoa theo thể loại |
| **ID** | UC6.2 |
| **Description** | Cho Khách hàng tìm kiếm hoa theo thể loại. |
| **Actor** | Khách, Thành viên. |
| **Pre-condition** | N/A |
| **Post-condition** | Hiển thị sách theo thể loại. |
| **Trigger** | Actor chọn [Thể loại hoa] cần tìm |
| **Basic flow** | 1. Chọn thể loại hoa trên phần nav.  2. Trang hiển thị hoa theo loại hoa. |
| **Alternative flow** | N/A |
| **Exception** | Nếu loại hoa không tồn tại sẽ thông báo “Trống”. |

* Use Case 7 – Quản lý thông tin tài khoản
* Use Case 7.1 – Xem thông tin tài khoản thành viên

|  |  |
| --- | --- |
| **Name** | Xem thông tin tài khoản. |
| **ID** | UC7.1 |
| **Description** | Xem thông tin tài khoản của Thành viên đang đăng nhập. |
| **Actor** | Thành viên |
| **Pre-condition** | Đăng nhập vào Website |
| **Post-condition** | Hiển thị danh sách thông tin tài khoản của Thành viên |
| **Trigger** | Actor nhấn vào [tên thành viên] |
| **Basic flow** | 1. Nhấn vào [Thông tin tài khoản] hoặc [Tên Thành viên] trên thanh header.  2. Thông tin tài khoản được tải lên trang. |
| **Alternative flow** | N/A |
| **Exception** | Không thể hiện thị thông tin tài khoản – thông báo lỗi |

* Use Case 7.2 – Cập nhật thông tin tài khoản

|  |  |
| --- | --- |
| **Name** | Cập nhập thông tin tài khoản |
| **ID** | UC7.2 |
| **Description** | Cho Thành viên sửa đổi và cập nhập lại thông tin tài khoản. |
| **Actor** | Thành viên. |
| **Pre-condition** | Đăng nhập vào Website. |
| **Post-condition** | Cập nhập thành công – lưu lại thông tin vào database.  Cập nhập thất bại – thông báo lỗi. |
| **Trigger** | Actor nhấn [Cập nhập tài khoản] |
| **Basic flow** | 1. Vào trang thông tin tài khoản.  2. Sửa đổi thông tin tài khoản.  3. Nhấn [Cập nhập tài khoản] để lưu. |
| **Alternative flow** | N/A |
| **Exception** | Thông tin sửa không đúng định dạng – thông báo lỗi.  Không cập nhập được – thông báo lỗi. |

* Use Case 8 – Xem thông tin mua hàng:
* Use Case 8.1 – Xem lịch sử mua hàng

|  |  |
| --- | --- |
| **Name** | Xem lịch sử mua hàng |
| **ID** | UC8.1 |
| **Description** | Cho Thành viên xem lịch sử mua hàng |
| **Actor** | Thành viên |
| **Pre-condition** | Đăng nhập vào Website |
| **Post-condition** | Hiển thị danh sách lịch sử mua hàng khi có lịch sử.  Hiển thị “không tìm thấy” nếu không có lịch sử mua hàng. |
| **Trigger** | Actor nhấn [Lịch sử mua hàng] |
| **Basic flow** | 1. Nhấn [Lịch sử mua hàng]  2. Hiển thị danh sách lịch sử mua hàng. |
| **Alternative flow** | N/A |
| **Exception** | Chưa mua hàng – thông báo không có lịch sử mua hàng |

* Use Case 8.2 – Xem chi tiết lịch sử mua hàng

|  |  |
| --- | --- |
| **Name** | Xem chi tiết lịch sử mua hàng |
| **ID** | UC8.2 |
| **Description** | Cho Thành viên chi tiết lịch sử đơn hàng |
| **Actor** | Thành viên |
| **Pre-condition** | Đăng nhập Website |
| **Post-condition** | Hiển thị chi tiết lịch sử đơn hàng nếu có lịch sử.  Hiển thị “không tìm thấy” nếu không có lịch sử. |
| **Trigger** | Actor nhấn xem [Chi tiết đơn hàng] |
| **Basic flow** | 1. Nhấn [Chi tiết đơn hàng] trong trang lịch sử mua hàng.  2. Hiển thị chi tiết đơn hàng đã chọn. |
| **Alternative flow** | N/A |
| **Exception** | Không load được dữ liệu – thông báo lỗi |

* Use Case 9 – Quản lý giỏ hàng
* Use Case 9.1 – Xem giỏ hàng

|  |  |
| --- | --- |
| **Name** | Xem giỏ hàng |
| **ID** | UC9.1 |
| **Description** | Cho Khách, Thành viên xem hoa trong giỏ hàng. |
| **Actor** | Khách, Thành viên |
| **Pre-condition** | N/A |
| **Post-condition** | Hiển thị sản phẩm trong giỏ hàng |
| **Trigger** | Actor chọn biểu tượng [Giỏ hàng] |
| **Basic flow** | 1. Nhấn vào [Giỏ hàng]  2. Giỏ hàng được load – hiển thị giỏ hàng |
| **Alternative flow** | N/A |
| **Exception** | Không có sản phẩm trong giỏ hàng – báo rỗng.  Không load được dữ liệu – báo lỗi. |

* Use Case 9.2 – Cập nhật giỏ hàng

|  |  |
| --- | --- |
| **Name** | Cập nhập giỏ hàng |
| **ID** | UC9.2 |
| **Description** | Cập nhập số lượng, tổng tiền. |
| **Actor** | Khách, Thành viên |
| **Pre-condition** | Sách đã được đưa vào giỏ hàng |
| **Post-condition** | Cập nhập giỏ hàng – load lại giỏ hàng. |
| **Trigger** | Actor nhấn [Cập nhập giỏ hàng] |
| **Basic flow** | 1. Thay đổi số lượng hoặc thêm hoa mới vào giỏ hàng.  2. Nhấn [Cập nhập giỏ hàng]. |

* Use Case 9.3 – Xóa giỏ hàng

|  |  |
| --- | --- |
| **Name** | Xóa giỏ hàng |
| **ID** | UC9.3 |
| **Description** | Cho phép Khách, Thành viên xóa hoa trong giỏ hàng |
| **Actor** | Khách, Thành viên |
| **Pre-condition** | Thêm hoa vào giỏ hàng. |
| **Post-condition** | Hoa bị xóa khỏi giỏ hàng |
| **Trigger** | Actor nhấn [Xóa] |
| **Basic flow** | 1. Chọn sách cần xóa.  2. Nhấn [X] hoặc [Xóa] để xóa hoa khỏi giỏ hàng  3. Giỏ hàng được cập nhập lại. |
| **Alternative flow** | N/A |
| **Exception** | Không thể nhấn xóa – thông báo lỗi  Giỏ hàng không được cập nhập lại |

* Use Case 10 – Đặt hàng
* Use Case 10.1 – Xem thông tin đặt hàng

|  |  |
| --- | --- |
| **Name** | Xem thông tin đặt hàng |
| **ID** | UC10.1 |
| **Description** | Thành viên xem thông tin đặt hàng. |
| **Actor** | Thành viên |
| **Pre-condition** | Đăng nhập vào Website.  Trong giỏ hàng có hoa. |
| **Post-condition** | Hiển thị thông tin đặt hàng |
| **Trigger** | Actor nhấn [Đặt hàng] |
| **Basic flow** | 1. Nhấn [Đặt hàng]  2. Load thông tin đặt hàng |
| **Alternative flow** | N/A |
| **Exception** | Không thể đặt hàng – thông báo lỗi  Hết hàng – thông báo hết hàng  Không thể load thông tin đặt hàng – thông báo lỗi |

* Use Case 10.2 – Thanh toán

|  |  |
| --- | --- |
| **Name** | Thành toán |
| **ID** | UC10.2 |
| **Description** | Thanh toán và xác nhận đơn hàng của Thành viên. |
| **Actor** | Thành viên |
| **Pre-condition** | Đăng nhập.  Giỏ hàng có hoa. |
| **Post-condition** | Đơn hàng được xác nhận và sẽ giao hàng.  Email thông báo về đơn hàng cho Thành viên. |
| **Trigger** | N/A |
| **Basic flow** | 1. Nhấn [Thanh toán]  2. Gửi thông tin đặt hàng đến khách hàng (email)  3. Đơn hàng chuyển vào trạng thái “Đang giao” |
| **Alternative flow** | N/A |
| **Exception** | Không thể thanh toán – thông báo lỗi  Email không được gửi – thông báo email  Đơn hàng không được duyệt – thông báo email |

* Use Case 11 – Xem danh sách khách hàng

|  |  |
| --- | --- |
| **Name** | Xem danh sách khách hàng |
| **ID** | UC11 |
| **Description** | Hiển thị dánh sách tất cả Thành viên và thông tin tài khoản của Thành viên. |
| **Actor** | Admin, Nhân viên |
| **Pre-condition** | Đăng nhập |
| **Post-condition** | Hiển thị danh sách khách hàng |
| **Trigger** | Actor nhấn [Danh sách khách hàng] |
| **Basic flow** | 1. Nhấn [Tài khoản khách hàng]  2. Nhấn [Danh sách khách hàng]  3. Danh sách khách hàng hiển thị |
| **Alternative flow** | N/A |
| **Exception** | Không thể load dữ liệu – thông báo lỗi database |

* Use Case 12 – Xem lịch sử mua hàng
* Use Case 12.1 – Xem danh sách lịch sử mua hàng

|  |  |
| --- | --- |
| **Name** | Xem danh sách lịch sử mua hàng |
| **ID** | UC12.1 |
| **Description** | Hiển lịch sử mua hàng của khách hàng. |
| **Actor** | Admin, Nhân viên. |
| **Pre-condition** | Đăng nhập. |
| **Post-condition** | Danh sách lịch sử mua hàng của khách hàng hiển thị |
| **Trigger** | Actor chọn khách hàng và nhấn [Lịch sử] |
| **Basic flow** | 1. Chọn khách hàng cần xem.  2. Nhấn [Lịch sử].  3. Danh sách lịch sử mua hàng của khách được load. |
| **Alternative flow** | Chọn [Đơn hàng đang giao] – xem đơn hàng đang giao của khách hàng.  Chọn [Đơn hàng đã giao] – xem đơn hang đã giao của khách hàng.  Admin và Nhân viên phải tự tổng hợp. |
| **Exception** | Không load được dữ liệu – thông báo lỗi database |

* Use Case 12.2 – Xem chi tiết lịch sử mua hàng

|  |  |
| --- | --- |
| **Name** | Xem chi tiết lịch sử mua hàng |
| **ID** | UC12.2 |
| **Description** | Xem chi tiết đơn hàng của Khách hàng |
| **Actor** | Admin, Nhân viên. |
| **Pre-condition** | Đăng nhập admin, nhân viên |
| **Post-condition** | Danh sách chi tiết đơn hàng của khách hàng |
| **Trigger** | Actor nhấn [Chi tiết] |
| **Basic flow** | 1. Chọn đơn hàng cần xem.  2. Nhấn [Chi tiết].  3. Dữ liệu được load – hiển thị danh sách |
| **Alternative flow** | 1. Vào danh sách đơn hàng.  2. Tìm kiếm khách hàng từ đơn hàng.  3. Nhấn [Chi tiết] |
| **Exception** | Không load được dữ liệu – thông bao lỗi dữ liệu |

* Use Case 13 – Quản lý tài khoản admin, nhân viên đang đăng nhập
* Use Case 13.1 – Xem thông tin tài khoản admin, nhân viên

|  |  |
| --- | --- |
| **Name** | Xem thông tin tài khoản admin đăng nhập |
| **ID** | UC13.1 |
| **Description** | Hiển thị thông tin tài khoản của admin |
| **Actor** | Admin |
| **Pre-condition** | Đăng nhập admin |
| **Post-condition** | Hiển thị thông tin tài khoản của admin đó |
| **Trigger** | Actor nhấn [Xem thông tin] |
| **Basic flow** | 1. Nhấn [Tài khoản của bạn] trong [Quản lý tài khoản].  2. Nhấn [Xem thông tin].  3. Load thông tin tài khoản đưa lên trang. |
| **Alternative flow** | 1. Nhấn vào [Tên Admin]  2. Chọn [Thông tin tài khoản]  3. Load dữ liệu. |
| **Exception** | Dữ liệu không load được – báo lỗi database. |

* Use Case 13.2 – Sửa đổi thông tin tài khoản admin, nhân viên

|  |  |
| --- | --- |
| **Name** | Sửa đổi thông tin tài khoản Admin |
| **ID** | UC13.2 |
| **Description** | Cho phép admin thay đổi thông tin tài khoản của mình |
| **Actor** | Admin |
| **Pre-condition** | Đăng nhập admin |
| **Post-condition** | Thông tin tài khoản được cập nhập và lưu lại |
| **Trigger** | Actor nhấn [Cập nhập thông tin] |
| **Basic flow** | 1. Nhấn [Tài khoản của bạn]  2. Chọn [Sửa đổi thông tin]  3. Sửa đổi thông tin tài khoản  4. Nhấn [Sửa đổi tài khoản]  5. Dữ liệu được lưu và load trở lại. |
| **Alternative flow** | N/A |
| **Exception** | Không thể load thông tin tài khoản – lỗi dữ liệu  Admin nhập thông tin sai định dạng – báo lỗi  Admin đổi ý nhấn [Quay trở lại] – thông tin giữ nguyên. |

* Use Case 14 – Quản lý tài khoản của nhân viên
* Use Case 14.1 – Xem danh sách tài khoản nhân viên

|  |  |
| --- | --- |
| **Name** | Xem danh sách tài khoản nhân viên |
| **ID** | UC14.1 |
| **Description** | Cho phép admin xem tất cả thông tin của nhân viên |
| **Actor** | Admin |
| **Pre-condition** | Đăng nhập admin |
| **Post-condition** | Hiển thị danh sách đầy đủ thông tin nhân viên |
| **Trigger** | Actor nhấn [Danh sách nhân viên] |
| **Basic flow** | 1. Nhấn [Khách hàng và nhân viên]  2. Chọn [Danh sách nhân viên]  3. Load danh sách và hiển thị |
| **Alternative flow** | N/A |
| **Exception** | Không load được danh sách – lỗi dữ liệu |

* Use Case 14.2 – Cập nhật thông tin tài khoản của nhân viên

|  |  |
| --- | --- |
| **Name** | Cập nhập thông tin tài khoản của nhân viên |
| **ID** | UC14.2 |
| **Description** | Cho phép Admin sửa đổi tài khoản của nhân viên |
| **Actor** | Admin |
| **Pre-condition** | Đăng nhập admin  Load danh sách nhân viên |
| **Post-condition** | Cập nhập thông tin nhân viên |
| **Trigger** | Actor nhấn [Cập nhập] |
| **Basic flow** | 1. Chọn nhân viên cần cập nhập  2. Sửa đổi thông tin  3. Nhấn [Cập nhập thông tin]  4. Thông tin được lưu và load lại. |
| **Alternative flow** | N/A |
| **Exception** | Không load được dữ liệu – lỗi dữ liệu  Thay đổi thông tin không đúng định dạng – thông báo lỗi  Lưu không được – thông báo |

* Use Case 14.3 – Xóa nhân viên

|  |  |
| --- | --- |
| **Name** | Xóa nhân viên |
| **ID** | UC14.3 |
| **Description** | Cho phép Admin xóa nhân viên |
| **Actor** | Admin |
| **Pre-condition** | Đăng nhập Admin |
| **Post-condition** | Nhân viên bị xóa và cập nhập lại danh sách nhân viên |
| **Trigger** | Admin nhấn [Xóa] |
| **Basic flow** | 1. Chọn nhân viên cần xóa  2. Nhấn [Xóa]  3. Nhân viên bị xóa  4. Cập nhập lại danh sách |
| **Alternative flow** | N/A |
| **Exception** | Danh sách nhân viên không được load  Không xóa được – thông báo lỗi  Không load lại danh sách |

* Use Case 14.4 – Thêm nhân viên mới:

|  |  |
| --- | --- |
| **Name** | Thêm nhân viên mới |
| **ID** | UC14.4 |
| **Description** | Admin thêm nhân viên mới |
| **Actor** | Admin |
| **Pre-condition** | Đăng nhập Admin |
| **Post-condition** | Thêm nhân viên mới và cập nhập danh sách |
| **Trigger** | Admin nhấn [Thêm] |
| **Basic flow** | 1. Nhấn [Khách hàng và nhân viên]  2. Chọn [Thêm nhân viên]  3. Điển thông tin nhân viên vào form  4. Nhấn [Thêm]  5. Nhân viên được thêm mới và load lại danh sách |
| **Alternative flow** | N/A |
| **Exception** | Thông tin nhân viên không đúng địng dạng – thông báo  Không thêm được nhân viên – thông báo lỗi |

* Use Case 15 – Thống kê hoa

|  |  |
| --- | --- |
| **Name** | Thống kế hoa |
| **ID** | UC15 |
| **Description** | Thống kê tất cả thông tin về hoa có trong dữ liệu |
| **Actor** | Admin |
| **Pre-condition** | Đăng nhập admin |
| **Post-condition** | Hiển thị thông tin hoa, danh mục hoa, loại hoa. |
| **Trigger** | Admin nhấn [Thông tin hoa] |
| **Basic flow** | 1. Chọn [Hoa]  2. Nhấn [Xem thông tin hoa]  3. Hiển thị thông tin hoa. |
| **Alternative flow** | N/A |
| **Exception** | Dữ liệu không được load – thông báo lỗi. |

* Use Case 16 – Kích hoạt khuyến mãi
* Use Case 16.1 – Xóa khuyến mãi

|  |  |
| --- | --- |
| **Name** | Xóa khuyến mãi |
| **ID** | UC16.1 |
| **Description** | Cho phép admin xóa khuyến mãi được chọn |
| **Actor** | Admin |
| **Pre-condition** | Đăng nhập admin |
| **Post-condition** | Khuyến mãi được xóa – load lại danh sách khuyến mãi |
| **Trigger** | Admin chọn [Xóa] |
| **Basic flow** | 1. Vào danh sách khuyến mãi  2. Chọn khuyến mãi  3. Nhấn [Xóa] |
| **Alternative flow** | N/A |
| **Exception** | Xóa không được thực hiện – thông báo lỗi  Dữ liệu không load lại – tự động load. |

* Use Case 16.2 – Bật, tắt khuyến mãi

|  |  |
| --- | --- |
| **Name** | Bật, tắt khuyến mãi |
| **ID** | UC16.2 |
| **Description** | Cho phép admin bật, tắt khuyến mãi. |
| **Actor** | Admin |
| **Pre-condition** | Đăng nhập Admin |
| **Post-condition** | Khuyến mãi bật, tắt theo ý muốn của admin |
| **Trigger** | Admin kích hoạt [Bật] hoặc [Tắt] |
| **Basic flow** | 1. Vào danh sách khuyến mãi  2. Chọn khuyến mãi cần kích hoạt  3. Nhấn [Bật] hoặc [Tắt]  4. Load lại dữ liệu |
| **Alternative flow** | N/A |
| **Exception** | Kích hoạt không thành công – thông báo lỗi |

* Use Case 17 – Thống kê doanh thu
* Use Case 17.1 – Thống kê đơn hàng đã giao

|  |  |
| --- | --- |
| **Name** | Thống kê đơn hàng đã giao |
| **ID** | UC17.1 |
| **Description** | Xem tất cả đơn hàng đã giao và tổng doanh thu |
| **Actor** | Admin |
| **Pre-condition** | Đăng nhập Admin |
| **Post-condition** | Hiển thị danh sách đơn hàng đã giao và tổng thu nhập |
| **Trigger** | Admin nhấn [Đơn hàng đã hoàn thành] |
| **Basic flow** | 1. Nhấn [Đơn hàng đã giao]  2. Load và hiển thị danh sách |
| **Alternative flow** | N/A |
| **Exception** | Không load được dữ liệu – thông báo lỗi dữ liệu |

* Use Case 17.2 – Thống kê doanh thu theo năm

|  |  |
| --- | --- |
| **Name** | Thống kê doanh thu theo năm |
| **ID** | UC17.2 |
| **Description** | Thống kê doanh thu theo năm cho admin xem |
| **Actor** | Admin |
| **Pre-condition** | Đăng nhập admin |
| **Post-condition** | Hiển thị về doanh thu, thống kê doanh thu |
| **Trigger** | Admin vào [Thống kê theo năm] |
| **Basic flow** | 1. Chọn [Thống kê doanh thu]  2. Chọn [Doanh thu theo năm]  3. Danh sách được load và hiển thị. |
| **Alternative flow** | N/A |
| **Exception** | Dữ liệu không được load – thông báo lỗi dữ liệu |

* Use Case 17.3 – Thống kê doanh thu theo tháng

|  |  |
| --- | --- |
| **Name** | Thống kê doanh thu theo tháng |
| **ID** | UC17.3 |
| **Description** | Thống kê doanh thu theo tháng cho admin xem |
| **Actor** | Admin |
| **Pre-condition** | Đăng nhập admin |
| **Post-condition** | Hiển thị về doanh thu, thống kê doanh thu |
| **Trigger** | Admin vào [Thống kê theo tháng] |
| **Basic flow** | 1. Chọn [Thống kê doanh thu]  2. Chọn [Doanh thu theo năm]  3. Chọn [Doanh thu theo tháng]  3. Danh sách được load và hiển thị. |
| **Alternative flow** | N/A |
| **Exception** | Không load được dữ liệu – thông báo lỗi. |

* Use Case 17.4 – Thống kê doanh thu theo ngày

|  |  |
| --- | --- |
| **Name** | Thống kê doanh thu theo ngày |
| **ID** | UC17.4 |
| **Description** | Thống kê doanh thu theo ngày cho admin xem |
| **Actor** | Admin |
| **Pre-condition** | Đăng nhập admin |
| **Post-condition** | Hiển thị doanh thu, thống kê doanh thu |
| **Trigger** | Admin vào [Thống kê theo ngày] |
| **Basic flow** | 1. Chọn [Thống kê doanh thu]  2. Chọn [Doanh thu theo năm]  3. Chọn [Doanh thu theo tháng]  4. Chọn [Doanh thu theo ngày]  5. Danh sách được load và hiển thị. |
| **Alternative flow** | N/A |
| **Exception** | Không load được dữ liệu – thông báo lỗi. |

* Use Case 18 – Quản lý tài khoản nhân viên đang đăng nhập
* Use Case 18.1 – Xem thông tin tài khoản

|  |  |
| --- | --- |
| **Name** | Xem thông tin tài khoản |
| **ID** | UC18.1 |
| **Description** | Cho phép Nhân viên xem thông tin tài khoản của mình |
| **Actor** | Nhân viên |
| **Pre-condition** | Đăng nhập Nhân viên |
| **Post-condition** | Hiển thị thông tin tài khoản nhân viên đang đăng nhập |
| **Trigger** | Nhân viên chọn [Xem thông tin] |
| **Basic flow** | 1. Chọn [Tài khoản của bạn]  2. Chọn [Xem thông tin]  3. Hiển thị thông tin tài khoản. |
| **Alternative flow** | N/A |
| **Exception** | Không load được dữ liệu – thông báo lỗi dữ liệu |

* Use Case 18.2 – Cập nhật thông tin tài khoản

|  |  |
| --- | --- |
| **Name** | Cập nhập thông tin tài khoản |
| **ID** | UC18.2 |
| **Description** | Cập nhập thông tin tài khoản của Nhân viên |
| **Actor** | Nhân viên |
| **Pre-condition** | Đăng nhập Nhân viên |
| **Post-condition** | Cập nhập lại thông tin Nhân viên |
| **Trigger** | Nhân viên nhấn [Cập nhập thông tin] |
| **Basic flow** | 1. Chọn [Tài khoản của bạn]  2. Chọn [Sửa đổi thông tin]  3. Sửa đổi thông tin.  4. Cập nhập và load lại danh sách. |
| **Alternative flow** | 1. Nhấn chuột vào [Tên Nhân viên] trên header.  2. Chọn [Sửa đổi thông tin]  3. Sửa đổi lại thông tin.  4. Cập nhập lại. |
| **Exception** | Không vào được trang Cập nhập thông tin – thông báo lỗi  Điền thông tin sai đinh dạng – thông báo và focus  Không thể cập nhập – thông báo lỗi dữ liệu. |

* Use Case 19 – Quản lý thông tin sách
* Use Case 19.1/ 19.3/ 19.6 – Xem danh sách thông tin hoa/ Thể loại hoa/ Loại hoa

|  |  |
| --- | --- |
| **Name** | Xem danh sách thông tin hoa/ Thể loại hoa/ Loại hoa |
| **ID** | UC19.1/ UC19.4/ UC19.7 |
| **Description** | Xem tất cả thông tin về hoa/ Thể loại/ Loại hoa |
| **Actor** | Nhân viên |
| **Pre-condition** | Đăng nhập Admin  Kết nối với csdl |
| **Post-condition** | Hiển thị danh sách thông tin về hoa/ Thể loại/ Loại hoa |
| **Trigger** | Nhân viên chọn [Xem thông tin hoa]/ [Thể loại hoa]/ [Loại hoa] |
| **Basic flow** | 1. Chọn [Thống kế hoa]  2. Chọn [Xem thông tin hoa]/ [Thể loại hoa]/ [Loại hoa].  3. Danh sách được load và hiện thị theo yêu cầu. |
| **Alternative flow** | N/A |
| **Exception** | Dữ liệu không thể load – thông báo dữ liệu không load |

* Use Case 19.2/ 19.5/ 19.8 – Cập nhật thông tin hoa/ Thể loại hoa/ Loại hoa

|  |  |
| --- | --- |
| **Name** | Cập nhập thông tin hoa/ Thể loại hoa/ Loại hoa |
| **ID** | UC19.2/ UC19.5/ UC19.8 |
| **Description** | Cho phép cập nhập thông tin hoa/ Thể loại hoa/ Loại hoa. |
| **Actor** | Nhân viên |
| **Pre-condition** | Đăng nhập Nhân viên |
| **Post-condition** | Cập nhập thành công |
| **Trigger** | Nhân viên nhấn [Lưu] |
| **Basic flow** | 1. Vào [Thống kê hoa]  2. Load danh sách  3. Chọn hoa / thể loại/ loại hoa cần cập nhật  4. Nhấn cập nhật  5. Lưu lại thông tin |
| **Alternative flow** | N/A |
| **Exception** | Nếu dữ liệu cập nhật sai định dạng – báo lỗi dữ liệu. |

* Use Case 19.3/ 19.6/ 19.9 – Thêm hoa mới/ Thể loại mới/ Loại hoa mới

|  |  |
| --- | --- |
| **Name** | Thêm hoa mới/ Thể loại mới/ Loại hoa mới |
| **ID** | UC19.3/ UC19.6/ UC19.9 |
| **Description** | Cho phép Nhân viên thêm hoa mới/ Thể loại mới/ Loại hoa mới. |
| **Actor** | Nhân viên. |
| **Pre-condition** | Đăng nhập admin |
| **Post-condition** | Tạo thêm hoa mới/ Thể loại mới/ Loại mới |
| **Trigger** | Nhân viên nhấn nút [Thêm] |
| **Basic flow** | 1. Thống kế sách  2. Chọn [Thêm hoa mới]/ [Thể loại hoa]/ [Loại hoa]  3. Thêm hoa mới và cập nhật lại danh sách. |
| **Alternative flow** | N/A |
| **Exception** | Nếu dữ liệu không được thêm – thông báo lỗi. |

* Use Case 20 – Quản lý thông tin khuyến mãi
* Use Case 20.1 – Xem khuyến mãi

|  |  |
| --- | --- |
| **Name** | Xem khuyến mãi |
| **ID** | UC20.1 |
| **Description** | Cho phép nhân viên xem danh sách khuyến mãi |
| **Actor** | Nhân viên |
| **Pre-condition** | Đăng nhập vào Nhân viên |
| **Post-condition** | Hiển thị danh sách tất cả khuyến mãi. |
| **Trigger** | Nhân viên nhấn [Danh sách khuyến mãi] |
| **Basic flow** | 1. Chọn [Quản lý khuyến mãi]  2. Nhấn [Danh sách khuyến mãi]  3. Load và hiển thị danh sách. |
| **Alternative flow** | N/A |
| **Exception** | Không load được dữ liệu – thông báo lỗi |

* Use Case 20.2 – Sửa đổi khuyến mãi

|  |  |
| --- | --- |
| **Name** | Sửa đổi khuyến mãi |
| **ID** | UC20.2 |
| **Description** | Cho phép nhân viên sửa đổi khuyến mãi |
| **Actor** | Nhân viên |
| **Pre-condition** | Đăng nhập Nhân viên |
| **Post-condition** | Khuyến mãi được cập nhập – load lại danh sách |
| **Trigger** | Nhân viên chọn [Sửa đổi] |
| **Basic flow** | 1. Chọn khuyến mãi  2. Nhấn [Sửa đổi]  3. Chỉnh sửa  4. Nhấn [Cập nhập] và load lại danh sách |
| **Alternative flow** | N/A |
| **Exception** | Không load được dữ liệu – thông báo lỗi  Không lưu được – thông báo lỗi dữ liệu. |

* Use Case 20.3 – Thêm khuyến mãi mới

|  |  |
| --- | --- |
| **Name** | Thêm khuyến mãi |
| **ID** | UC20.3 |
| **Description** | Cho phép Nhân viên thêm khuyến mãi |
| **Actor** | Nhân viên |
| **Pre-condition** | Đăng nhập Nhân viên |
| **Post-condition** | Khuyến mãi mới được thêm – load lại danh sách |
| **Trigger** | Nhân viên nhấn [Thêm mới] |
| **Basic flow** | 1. Vào [Quản lý khuyến mãi]  2. Chọn [Thêm khuyến mãi]  3. Điền thông tin khuyến mãi.  4. Lưu và load dư liệu. |
| **Alternative flow** | N/A |
| **Exception** | Điền thông tin sai định dàng – thông báo và focus |

* Use Case 21 – Quản lý đơn hàng
* Use Case 21.1 – Xem đơn hàng đang giao

|  |  |
| --- | --- |
| **Name** | Xem đơn hàng đang giao |
| **ID** | UC21.1 |
| **Description** | Để nhân viên xem thông tin đơn hàng đang giao |
| **Actor** | Nhân viên |
| **Pre-condition** | Đăng nhập Nhân viên |
| **Post-condition** | Hiển thị danh sách đơn hàng đang giao |
| **Trigger** | Nhân viên nhấn [Xem đơn hàng đang giao] |
| **Basic flow** | 1. Chọn [Đơn hàng đang giao]  2. Hiển thị danh sách đơn hàng đang giao |
| **Alternative flow** | N/A |
| **Exception** | Không load được dữ liệu – thông báo lỗi |

* Use Case 21.2 – Xem chi tiết đơn hàng đang giao

|  |  |
| --- | --- |
| **Name** | Xem chi tiết đơn hàng đang giao |
| **ID** | UC21.2 |
| **Description** | Để nhân viên xem chi tiết đơn hàng đang giao |
| **Actor** | Nhân viên |
| **Pre-condition** | Đăng nhập Nhân viên |
| **Post-condition** | Hiển thị chi tiết danh sách đơn hàng |
| **Trigger** | Nhân viên nhấn [Chi tiết đơn hàng] |
| **Basic flow** | 1. Chọn [Chi tiết]  2. Hiển thị chi tiết |
| **Alternative flow** | N/A |
| **Exception** | Không load được dữ liệu – thông báo lỗi |

* Use Case 21.3 – Hủy đơn hàng đang giao

|  |  |
| --- | --- |
| **Name** | Hủy đơn hàng đang giao |
| **ID** | UC21.3 |
| **Description** | Cho phép Nhân viên hủy đơn hàng đang giao |
| **Actor** | Nhân viên |
| **Pre-condition** | Đăng nhập Nhân viên |
| **Post-condition** | Xóa đơn hàng đang giao, đưa sách về lại kho |
| **Trigger** | Nhân viên nhấn [Hủy] |
| **Basic flow** | 1. Chọn đơn hàng  2. Nhấn [Hủy]  3. Hủy đơn hàng và cập nhập lại sách vào kho |
| **Alternative flow** | N/A |
| **Exception** | Không xóa được – thông báo lỗi  Không đưa sách về kho – thông báo lỗi |

* Use Case 21.4 – Xác nhận đã giao

|  |  |
| --- | --- |
| **Name** | Xác nhận đã giao |
| **ID** | UC21.4 |
| **Description** | Cho phép Nhân viên xác nhận đã giao hàng |
| **Actor** | Nhân viên |
| **Pre-condition** | Đăng nhập Nhân viên |
| **Post-condition** | Đơn hàng được xác nhận giao và chuyển sang đã giao |
| **Trigger** | Nhân viên nhấn [Đã giao] |
| **Basic flow** | 1. Vào dánh sách đơn hàng đang giao.  2. Nhấn [Đã giao]  3. Đơn hàng được giao và cập nhập danh sách |
| **Alternative flow** | N/A |
| **Exception** | Không thể xác nhận – báo lỗi hệ thống. |

* Use Case 21.5 – Xem đơn hàng đã giao

|  |  |
| --- | --- |
| **Name** | Xem đơn hàng đã giao |
| **ID** | UC21.5 |
| **Description** | Xem những đơn hàng đã giao |
| **Actor** | Nhân viên. |
| **Pre-condition** | Đăng nhập Nhân viên |
| **Post-condition** | Hiển thị danh sách đơn hàng đã giao |
| **Trigger** | Nhân viên vào [Đơn hàng đã giao] |
| **Basic flow** | 1. Vào [Đơn hàng đã giao]  2. Load và hiển thị danh sách |
| **Alternative flow** | N/A |
| **Exception** | Không load được dữ liệu – báo lỗi “trống” |

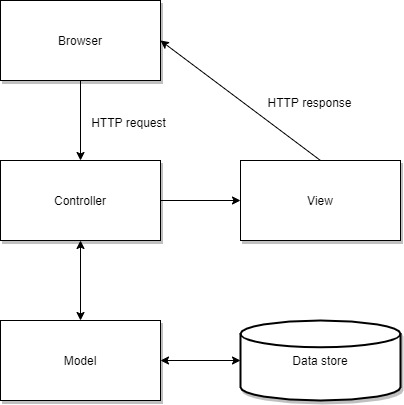
# THIẾT KẾ HỆ THỐNG

## **Phác thảo mô hình ứng dụng:**

### **Phác thảo mô hình tổng quan:**

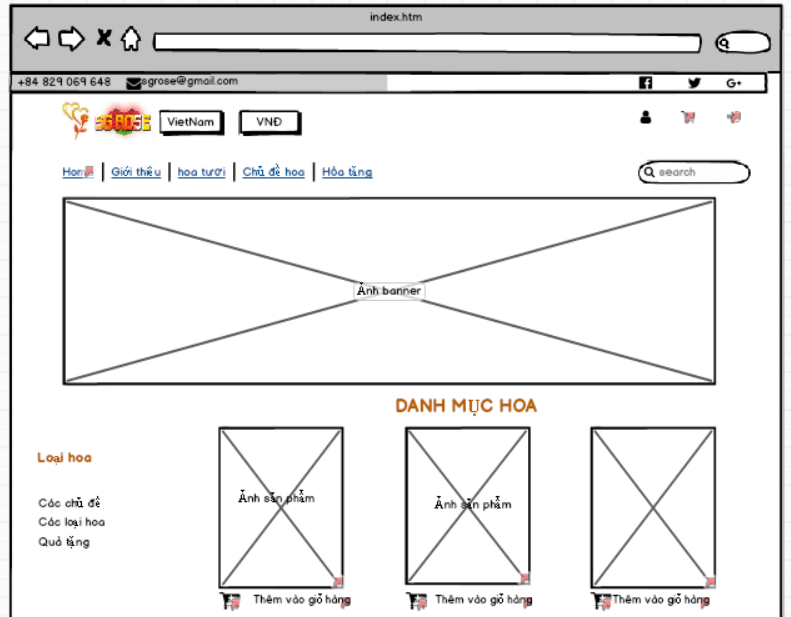


### **Phác thảo mô hình ứng dụng web theo MVC:**

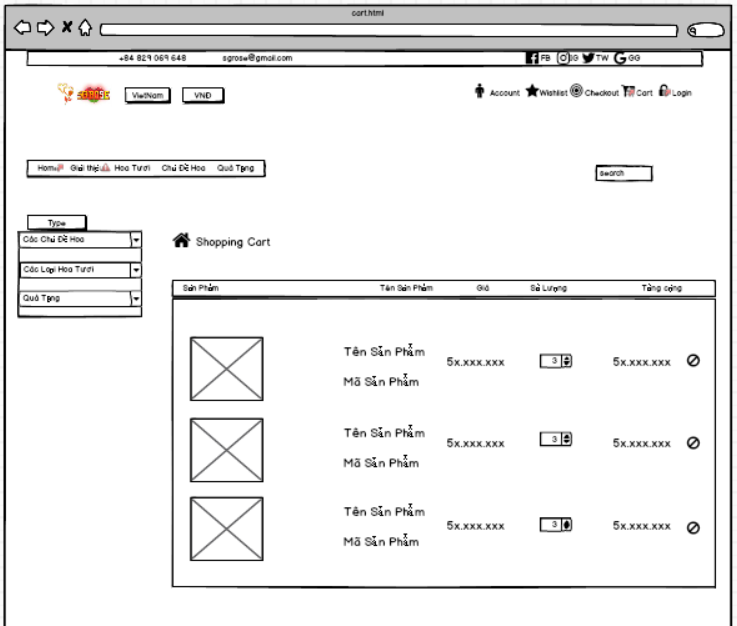


## **Phác thảo giao diện:**

* Trang phác thảo index dành cho khách hàng



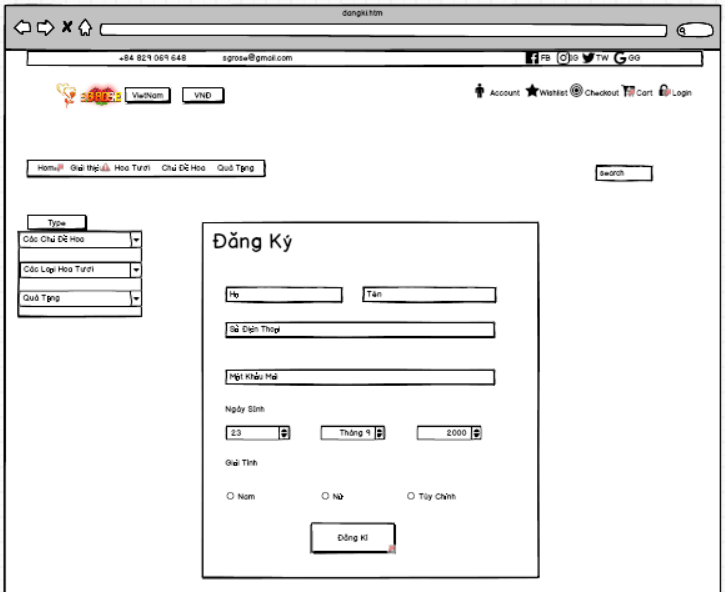
* Trang phác thảo shopping cart dành cho khách hàng



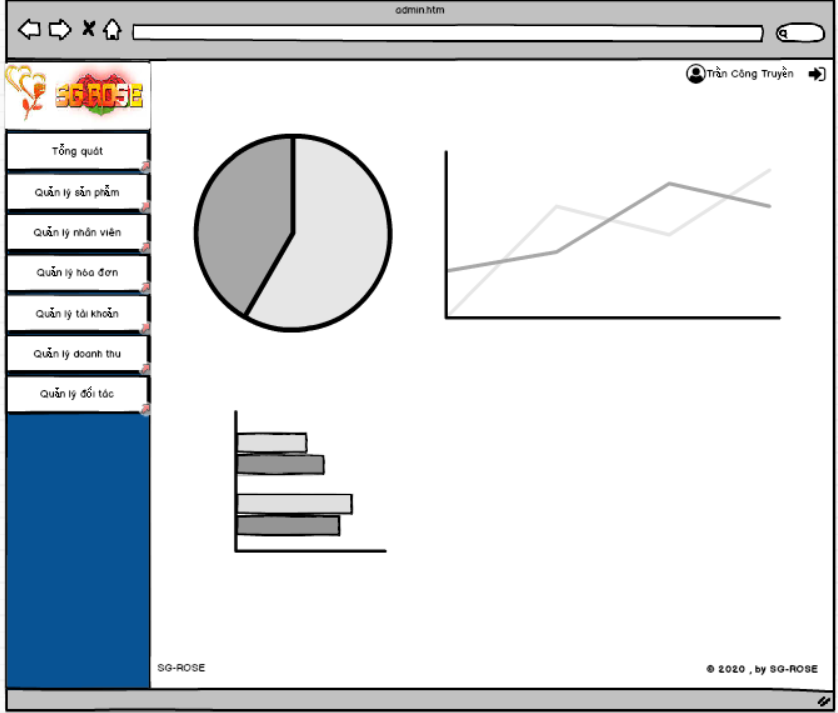
* Trang phác thảo đăng nhập dành cho khách hàng



* Trang phác thảo đăng kí dành cho khách hàng



* Trang phác thảo admin và nhân viên



## **Phác thảo ERD:**



# THỰC HIỆN DỰ ÁN

## **Thiết kế CSDL:**

* Bảng Actives

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Khóa** | **Ràng buộc** | **Mô tả** |
| Id | Int | PK | Not null | Mã tài khoản |
| Name | Varchar(10) |  | Not null | Tên tài khoản |

* Bảng Companys

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Khóa** | **Ràng buộc** | **Mô tả** |
| Name | Nvarchar(500) |  | Not null | Tên của hàng |
| Address | Nvarchar(500) |  | Not null | Địa chỉ của hàng |
| Fax | Varchar(10) |  |  | SĐT Của hàng |
| Link | Varchar(500) |  |  | Web cửa hang |
| Id | int | PK | Not null | Mã cửa hàng |
| Image | Varchar(100) |  |  | Hình ảnh |

* Contacts

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Khóa** | **Ràng buộc** | **Mô tả** |
| Id | Int | PK | Not null | Mã liên hệ |
| UserId | Int |  |  | Mã tài khoản liên hệ |
| Topic | Nvarchar(100) |  |  | Tiêu đề |
| Message | Nvarchar(4000) |  | Not null | Tin nhắn |
| Phone | Varchar(20) |  | Not null | Số điện thoại |
| Map | Varchar(500) |  |  | Bản đồ |

* DiscountPrograms

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Khóa** | **Ràng buộc** | **Mô tả** |
| Id | Int | PK | Not null | Mã sản phẩm |
| Name | Nvarchar(100) |  | Not null | Tên sản phẩm |
| DateStart | Date |  | Not null | Ngày bắt đầu |
| DateEnd | Date |  | Not null | Ngày kết thúc |

* FlowerPromotionPrograms

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Khóa** | **Ràng buộc** | **Mô tả** |
| Id | Int | PK | Not null | Mã |
| DiscountId | Int |  | Not null | Mã giảm giá |
| FlowerID | Int |  | Not null | Mã hoa |
| TypePercent | Int |  | Not null | Loại phần tram |
| TypePrice | Int |  | Not null | Loại giá |
| Amount | int |  | Not null | Số lượng |

* Flowers

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Khóa** | **Ràng buộc** | **Mô tả** |
| Id | Int | PK | Not null | Mã hoa |
| Name | Nvarchar(50) |  | Not null | Tên hoa |
| TypeId | Int |  |  | Loại mã hoa |
| Amount | Int |  | Not null | Số lượng |
| Price | Int |  |  | Giá |
| Image | Nvarchar(100) |  |  | Hình ảnh |
| Notes | Nvarchar(4000) |  |  | Ghi chú |
| Createdtime | Date |  |  | Ngày tạo |
| Updatedtime | Date |  |  | Ngày cập nhật |

* Genders

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Khóa** | **Ràng buộc** | **Mô tả** |
| Id | Int | PK | Not null | Mã giới tính |
| Name | Nvarchar(10) |  | Not null | Giới tính |

* Ispaids

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Khóa** | **Ràng buộc** | **Mô tả** |
| Id | Int | PK | Not null | Mã thanh toán |
| Name | Nvarchar(50) |  | Not null | Thanh toán |

* Orders

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Khóa** | **Ràng buộc** | **Mô tả** |
| Id | Int | PK | Not null | Mã hóa đơn |
| UserId | Int |  |  | Mã tài khoản |
| Address | Nvarchar(500) |  |  | Địa chỉ |
| TotalMoney | Int |  |  | Tổng giá tiền |
| Status | Int |  |  | Hiện tại |
| Ispaid | Int |  |  | Tình trạng |
| Createdtime | Date |  |  | Ngày tạo |
| Updatedtime | Date |  |  | Ngày cập nhật |
| Name | Nvarchar(50) |  |  | Tên |
| NumberPhone | Varchar(10) |  |  | Số điện thoại |
| Email | Varchar(100) |  |  | Địa chỉ mail |

* OrdersDetails

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Khóa** | **Ràng buộc** | **Mô tả** |
| Id | Int | PK | Not null | Mã hóa đơn |
| OrderId | Int |  | Not null | Mã thanh toán |
| FlowerId | Int |  | Not null | Mã hoa |
| Price | Int |  | Not null | Giá |
| Amount | Int |  | Not null | Số lượng |
| Totalprice | Int |  | Not null | Tổng giá tiền |
| Createdtime | Date |  |  | Ngày tạo |
| Updatedtime | Date |  |  | Ngày cập nhật |

* OrderStatus

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Khóa** | **Ràng buộc** | **Mô tả** |
| Id | Int | PK | Not null | Mã hóa đơn |
| Name | Nvarchar(50) |  | Not null | Tình trạng của hóa đơn |

* Roles

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Khóa** | **Ràng buộc** | **Mô tả** |
| Id | Int | PK | Not null | Mã vai trò |
| Name | Varchar(255) |  | Not null | Tên vai trò |

* Staffs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Khóa** | **Ràng buộc** | **Mô tả** |
| Id | Int | PK | Not null | Mã nhân viên |
| Name | Nvarchar(50) |  | Not null | Tên nhân viên |
| Email | Nvarchar(100) |  | Not null | Địa chỉ mail |
| Password | Nvarchar(100) |  | Not null | Mật khẩu |
| Birthday | Date |  | Not null | Ngày sinh |
| Address | Nvarchar(200) |  | Not null | Địa chỉ |
| Avatar | Varchar(200) |  |  | Hình avatar |
| Gender | Int |  |  | Giới tính |
| Phone | Varchar(20) |  | Not null | Số điện thoại |
| Role | Int |  | Not null | Vai trò |
| Createdtime | Date |  |  | Ngày tạo |
| Updatedtime | Date |  |  | Ngày cập nhật |

* TypesOfFlowers

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Khóa** | **Ràng buộc** | **Mô tả** |
| Id | Int | PK | Not null | Mã loại hoa |
| Name | Nvarchar(50) |  | Not null | Tên loại hoa |
| Image | Nvarchar(50) |  |  | Hình ảnh |

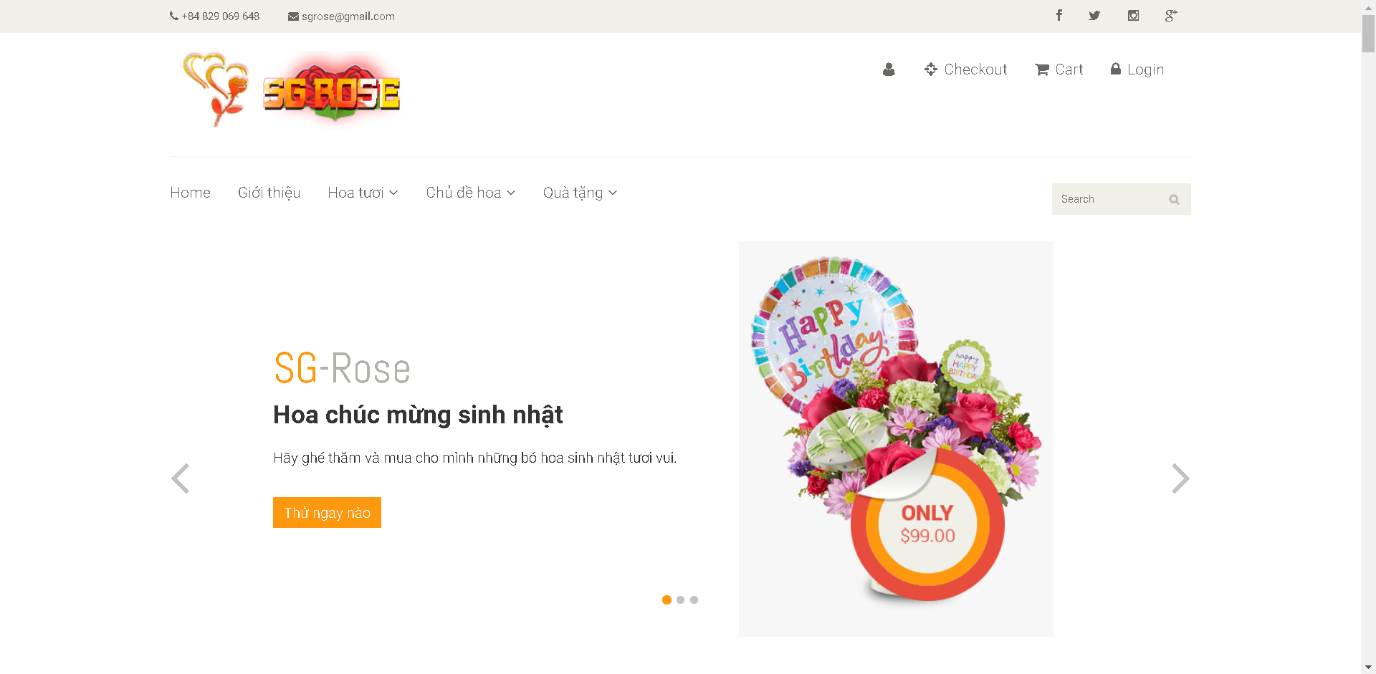
* Users

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Khóa** | **Ràng buộc** | **Mô tả** |
| Id | Int | PK | Not null | Mã tài khoản |
| Name | Nvarchar(50) |  |  | Tên tài khoản |
| Email | Nvarchar(200) |  |  | Địa chỉ mail |
| Address | Nvarchar(200) |  |  | Địa chỉ |
| Phone | Nvarchar(100) |  |  | Số điện thoại |
| Password | Nvarchar(100) |  |  | Mật khẩu |
| Avatar | Varchar(100) |  |  | Hình avatar |
| Gender | Int |  |  | Giới tính |
| IsActive | Int |  |  | Tình trạng |
| Createdtime | Date |  |  | Ngày tạo |
| Updatedtime | Date |  |  | Ngày cập nhật |
| Birthday | Date |  |  | Ngày sinh |

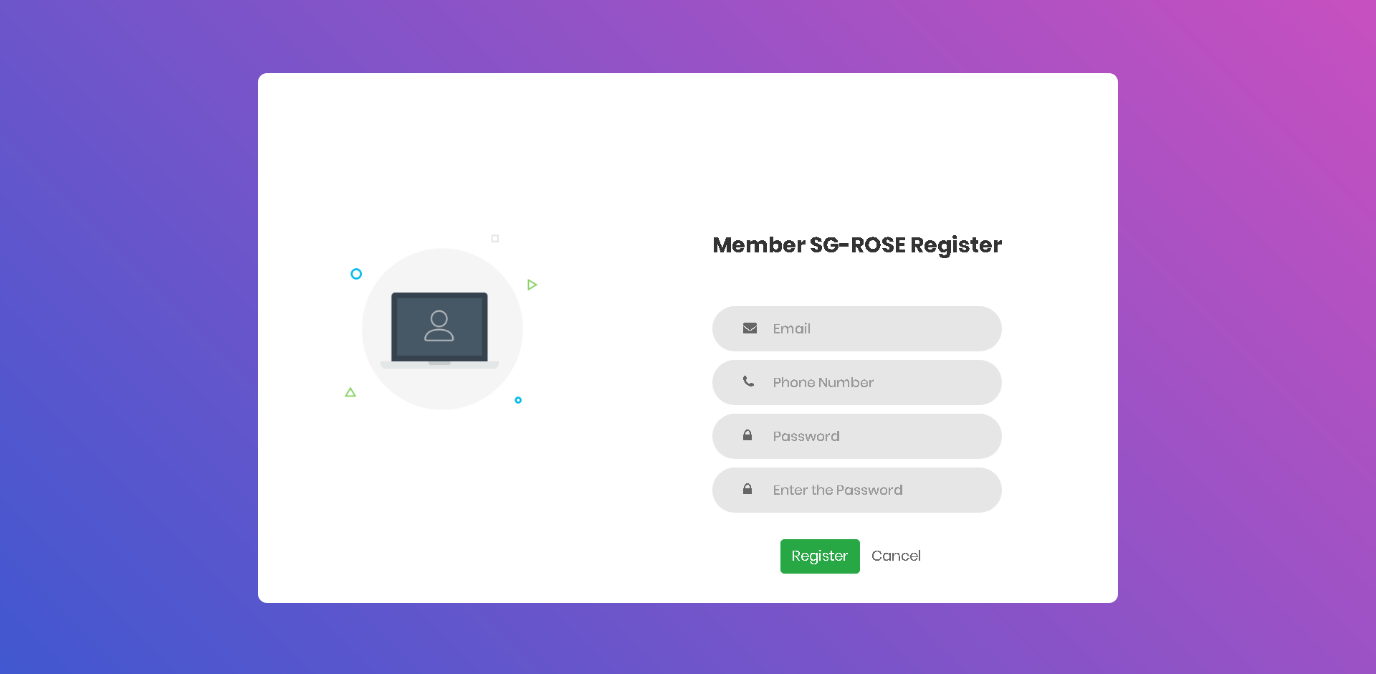
## **Thiết kế layout, thiết kế giao diện chi cho các chức năng:**

1. Giao diện dành cho khách hàng:

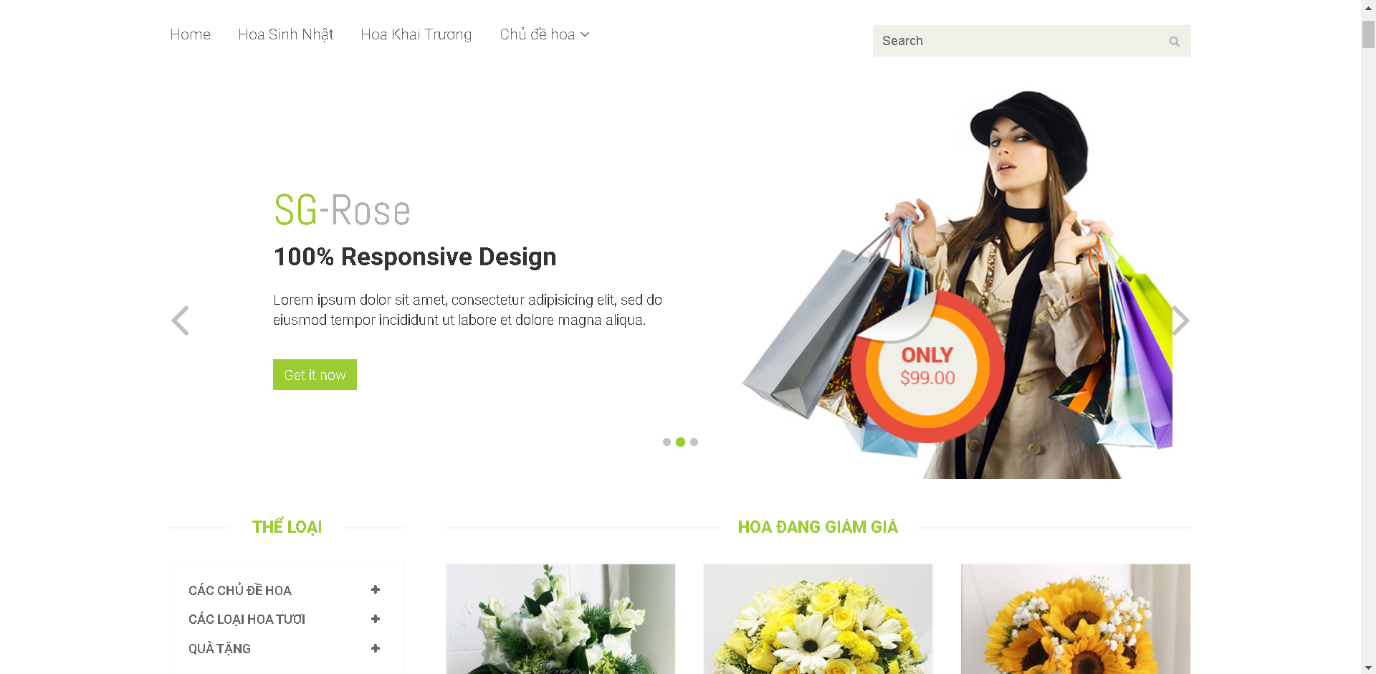
* Trang index:



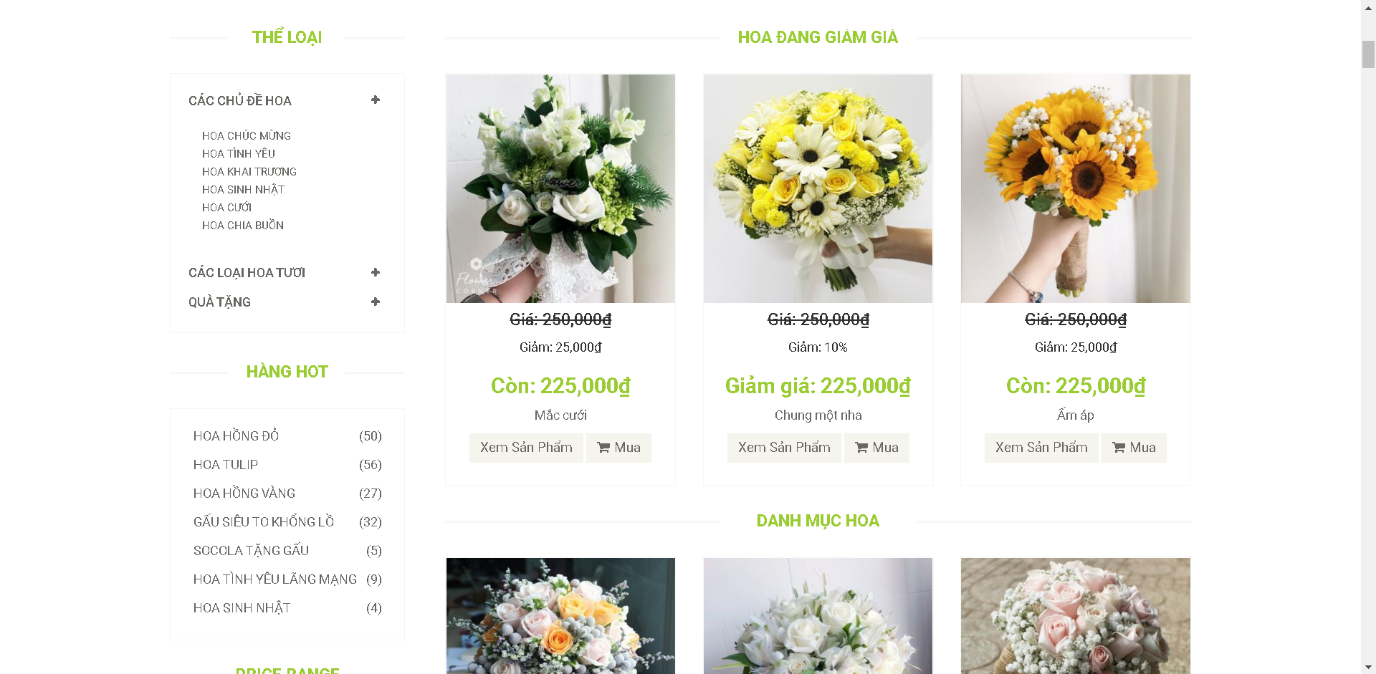
* + Mô tả: Hiển thị giao diện trang chính tương tác với khách hang
  + Hoạt động:
* Tìm kiếm hoa theo tên
* Tìm kiếm hoa theo thể loại
* Xem quà tặng
* Vào giỏ hàng
* Đăng ký
* Trang đăng ký:



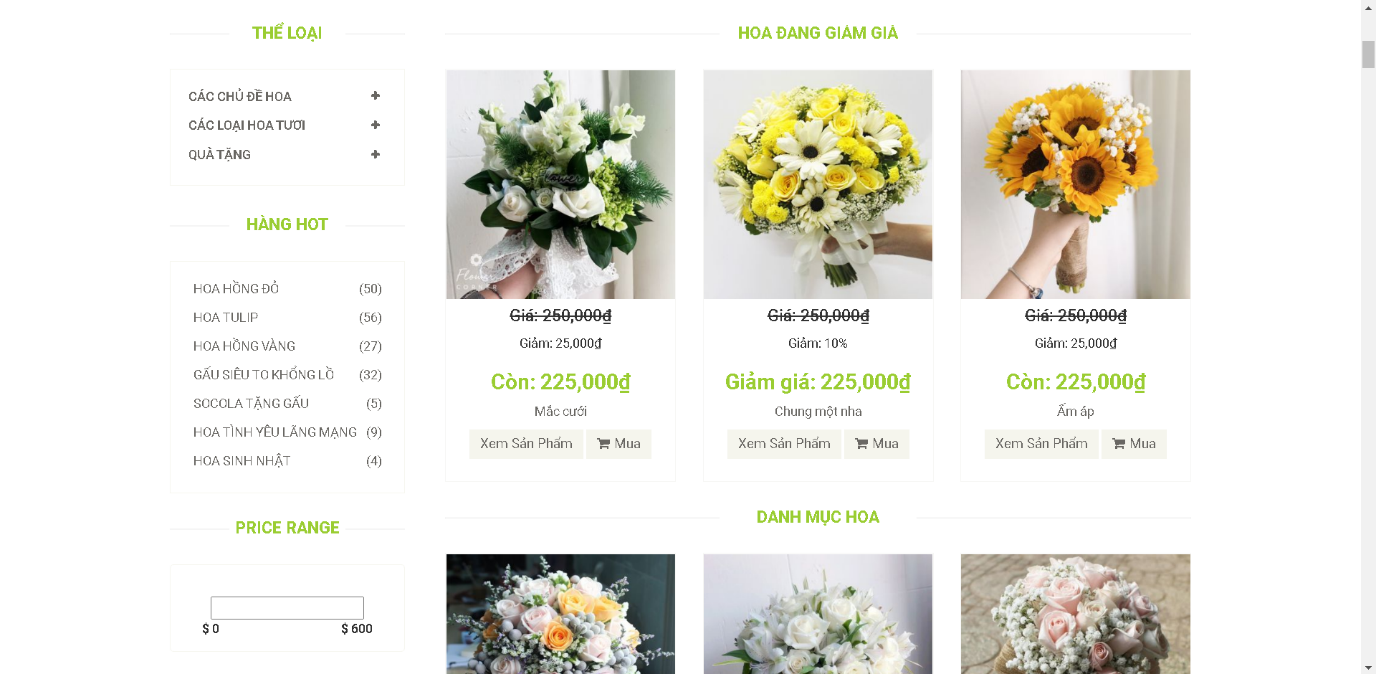
* + Mô tả: Hiển thị form đăng ký cho khách hàng
  + Hoạt động:
* Cho khách đăng ký điền vào thông tin
* [Đăng ký] tạo tài khoản khách hàng
* Tìm kiếm hoa theo tên:



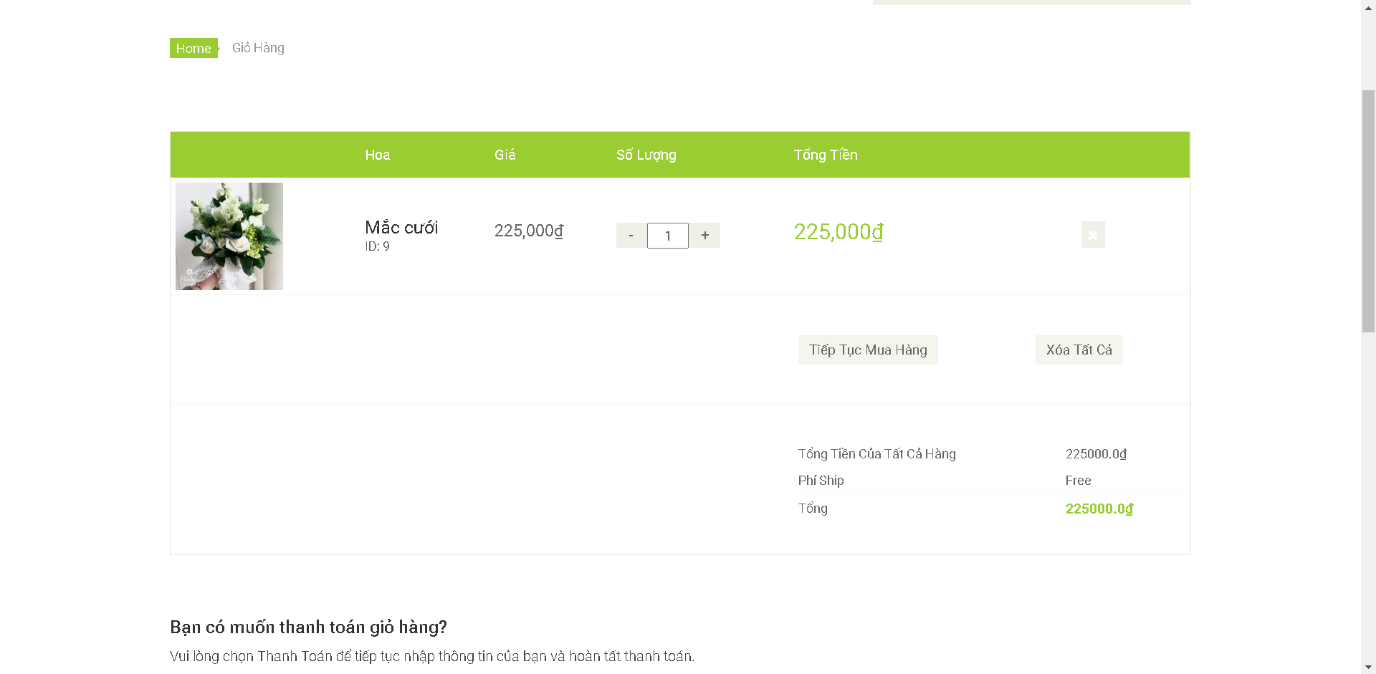
* + Mô tả: Tìm kiếm và hiển thị sách theo tên
  + Hoạt động:
* Nhấn từ khóa từ [Tìm kiếm]
* Tìm kiếm hoa phù hợp và load danh sách
* Tìm kiếm hoa theo thể loại:



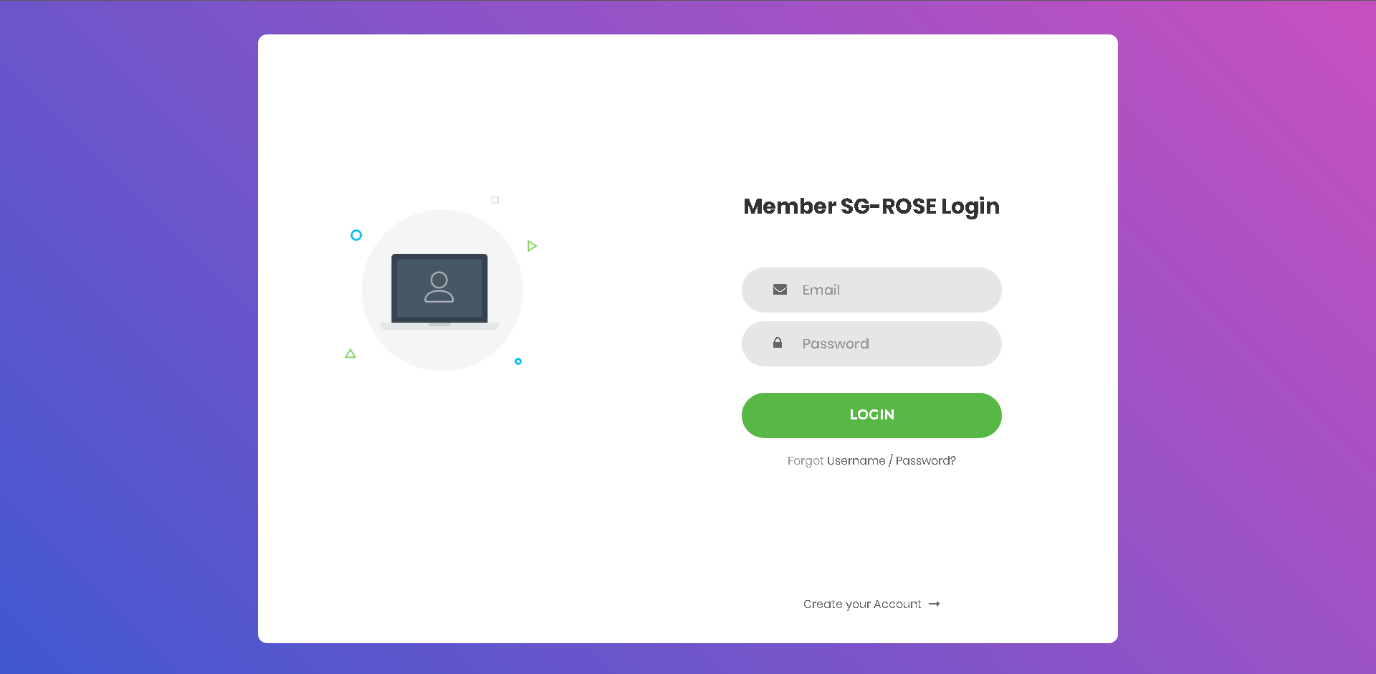
* + Mô tả: Tìm kiếm và hiển thị sách theo thể loại
  + Hoạt động:
* Nhấn vào các mục ở trong mục Thể loại
* Tìm kiếm hoa phù hợp
* Hàng hot:



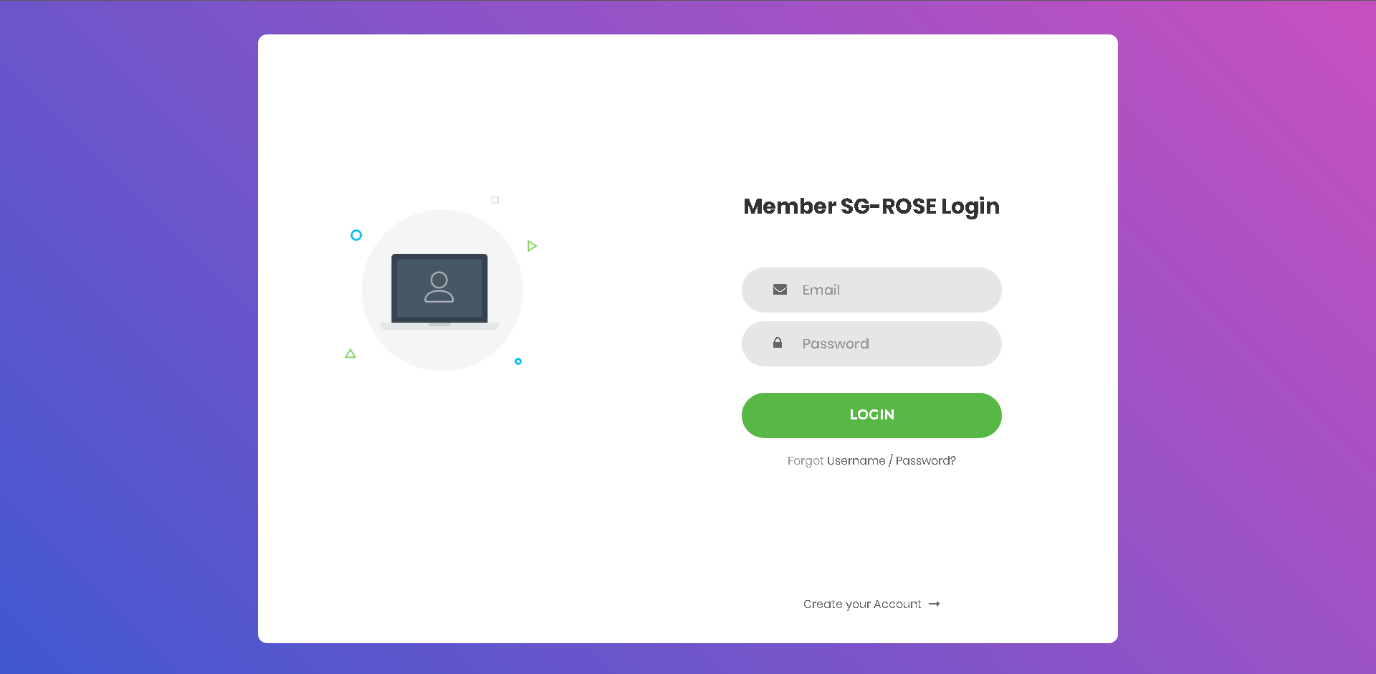
* + Mô tả: Tìm kiếm và hiển thị hoa theo hàng hot
  + Hoạt động:
* Nhấn vào các mục ở trong mục Hàng hot
* Tìm kiếm hoa phù hợp
* Giỏ hàng:



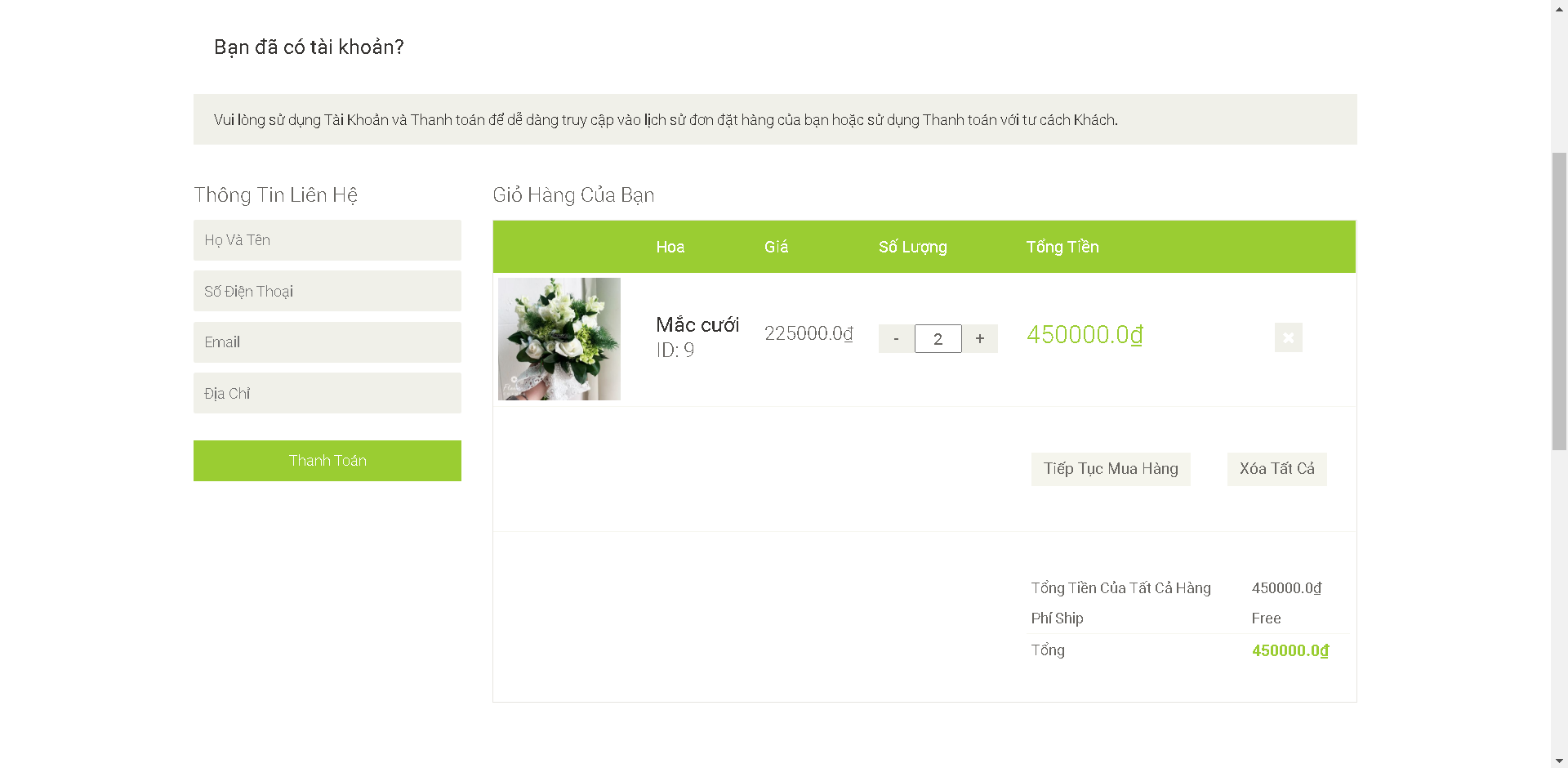
* + Mô tả: Hiển thị tất cả hoa có trong giỏ hàng
  + Hoạt động:
* Load hoa được đưa vào giỏ hàng
* Click [X]: xóa hoa khỏi giỏ hàng
* Cập nhật giỏ hoa
* Đặt hàng => Thanh toán
* Trang đăng nhập cho khách hàng:



* + Mô tả: Hiển thị form đăng nhập
  + Hoạt động:
* Input email & password
* [Login] đăng nhập tài khoản
* Quên mật khẩu:



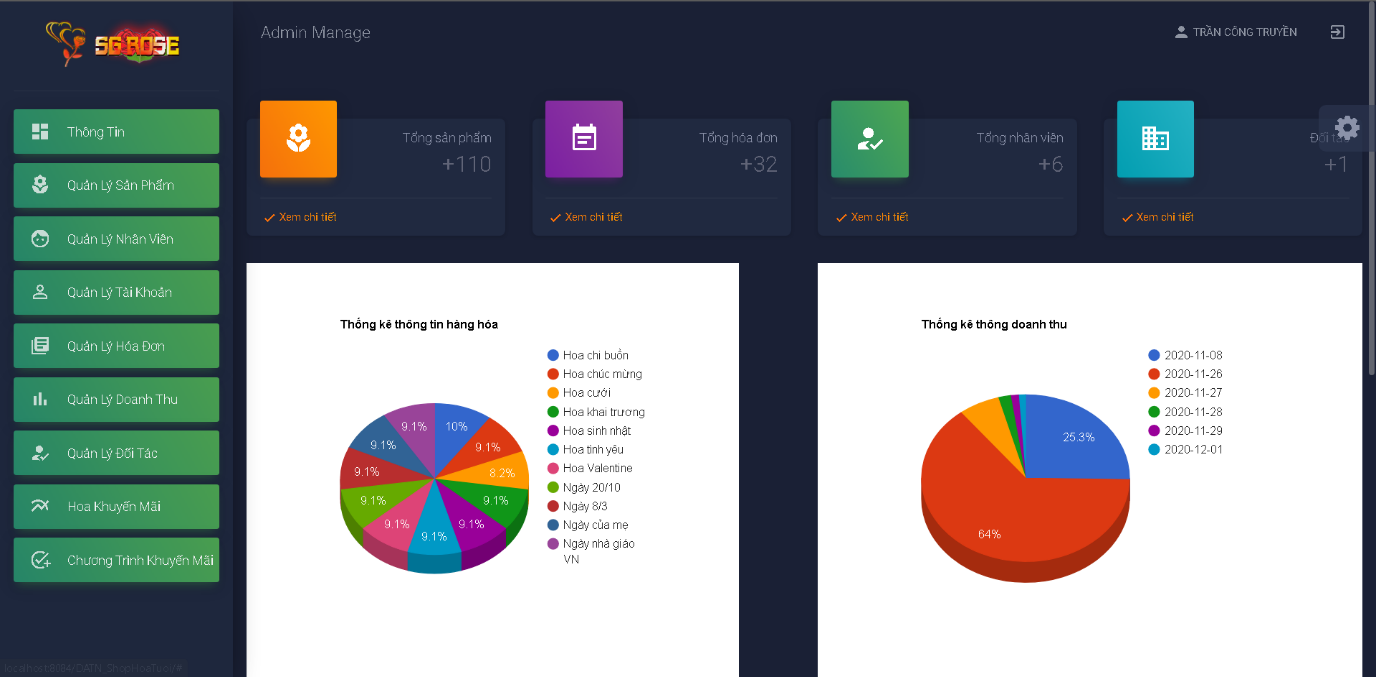
* + Mô tả: Hiển thị form điển email để lấy lại mật khẩu
  + Hoạt động:
    - Input email
    - [Gửi mật khẩu] gửi mật khẩu vào email
* Thanh toán:



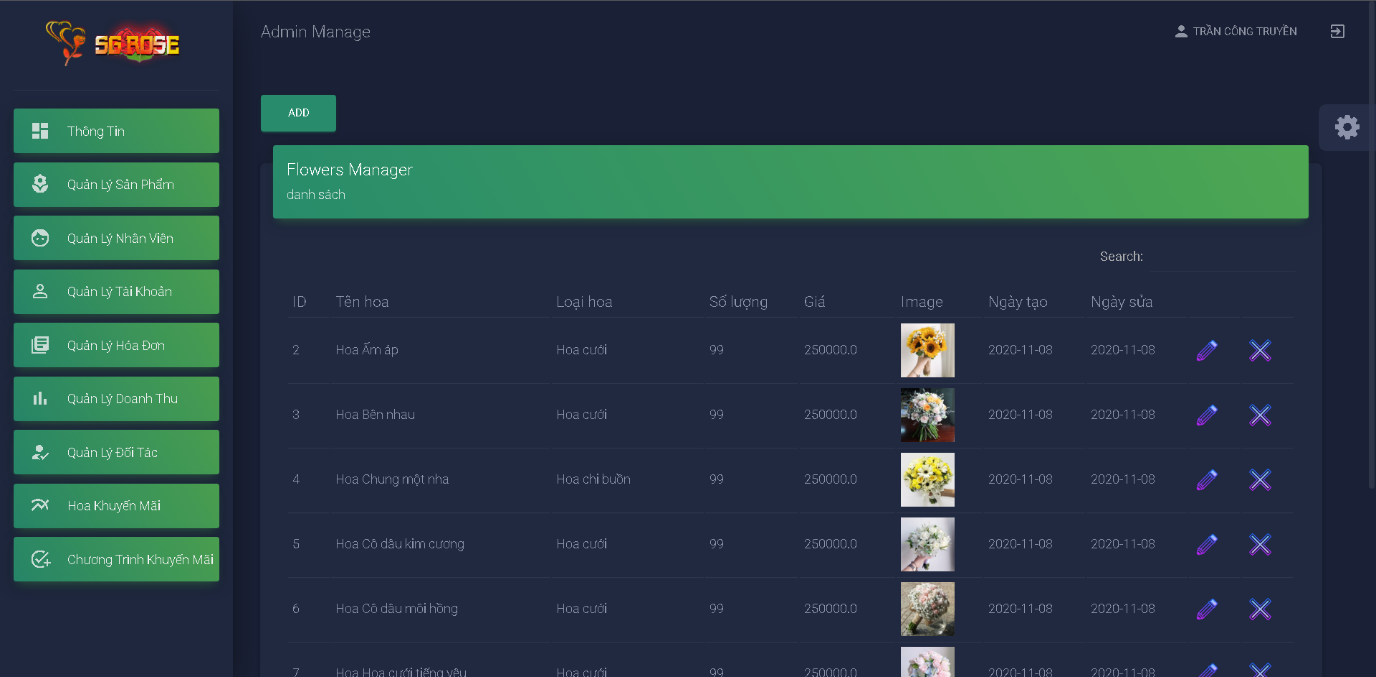
* + Mô tả: Hiển thị thông tin liên hệ và hóa đơn
  + Hoạt động:
* Nhập các thông tin vào trong mục Thông tin liên hệ
* Load thông tin đơn hàng
* [Thanh toán] Xác nhận đặt hàng và gửi mail xác nhận

1. Giao diện dành cho admin, nhân viên:

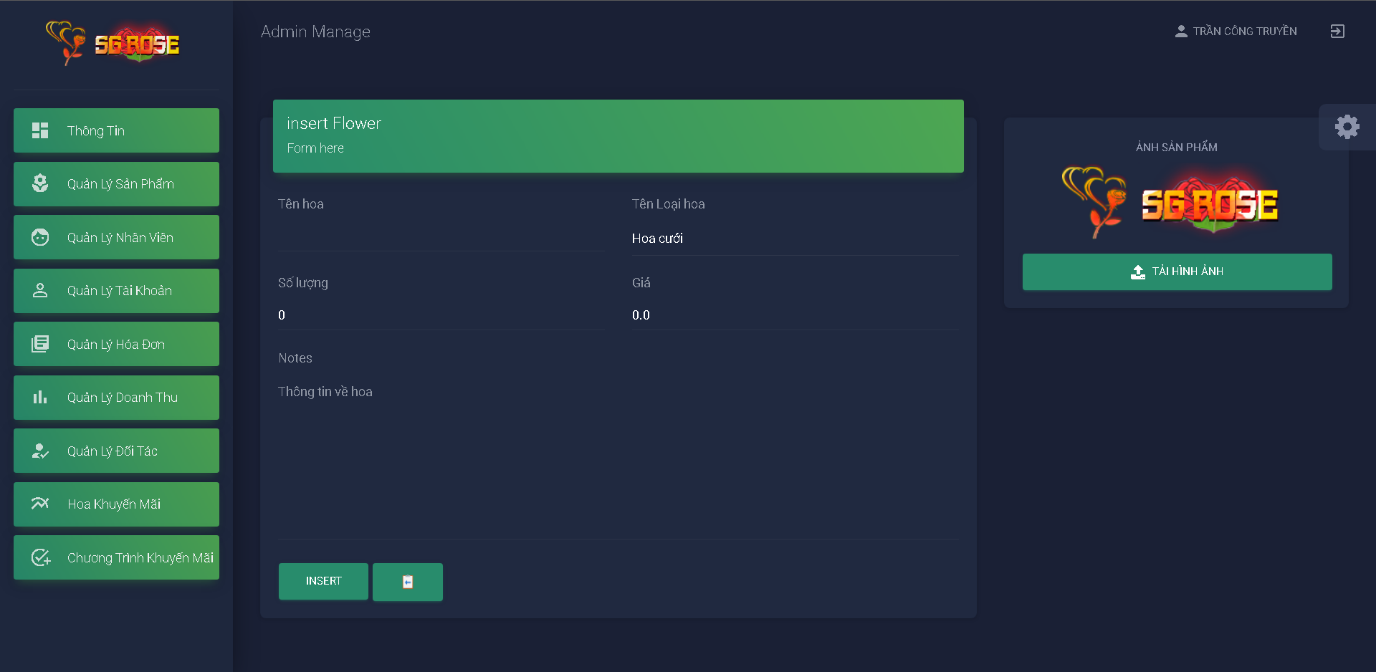
* Trang index admin:



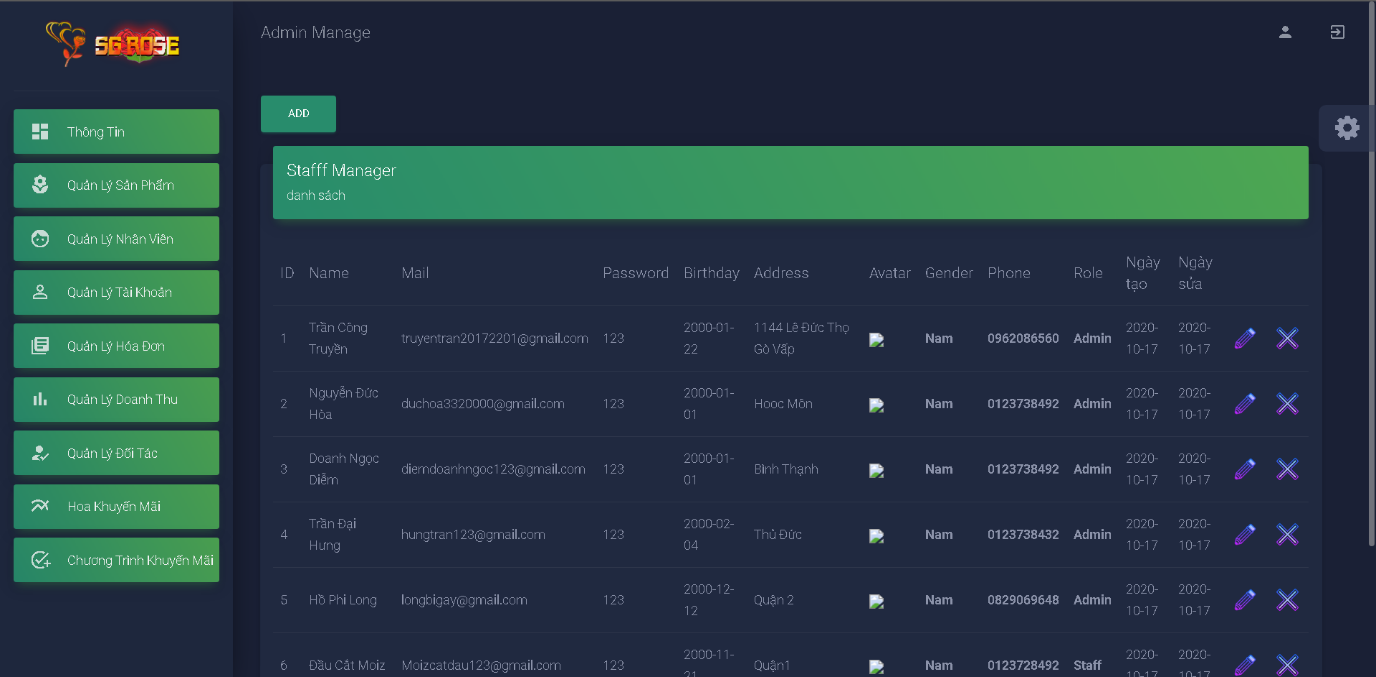
* + Mô tả: Thống kê doanh thu và chức năng quản lý cho admin, nhân viên
  + Hoạt động:
* Sidebar bên trái: Các chức năng của admin, nhân viên
* Load dữ liệu thống kê
* Quản lý sản phẩm:



* + Mô tả: Hiển thị danh sách sản phẩm hoa
  + Hoạt động:
* Nhấn từ khóa từ [Search] để tìm hoa
* Chỉnh sửa hoặc xóa sản phẩm hoa
* Nhấn nút Add để thêm sản phẩm hoa



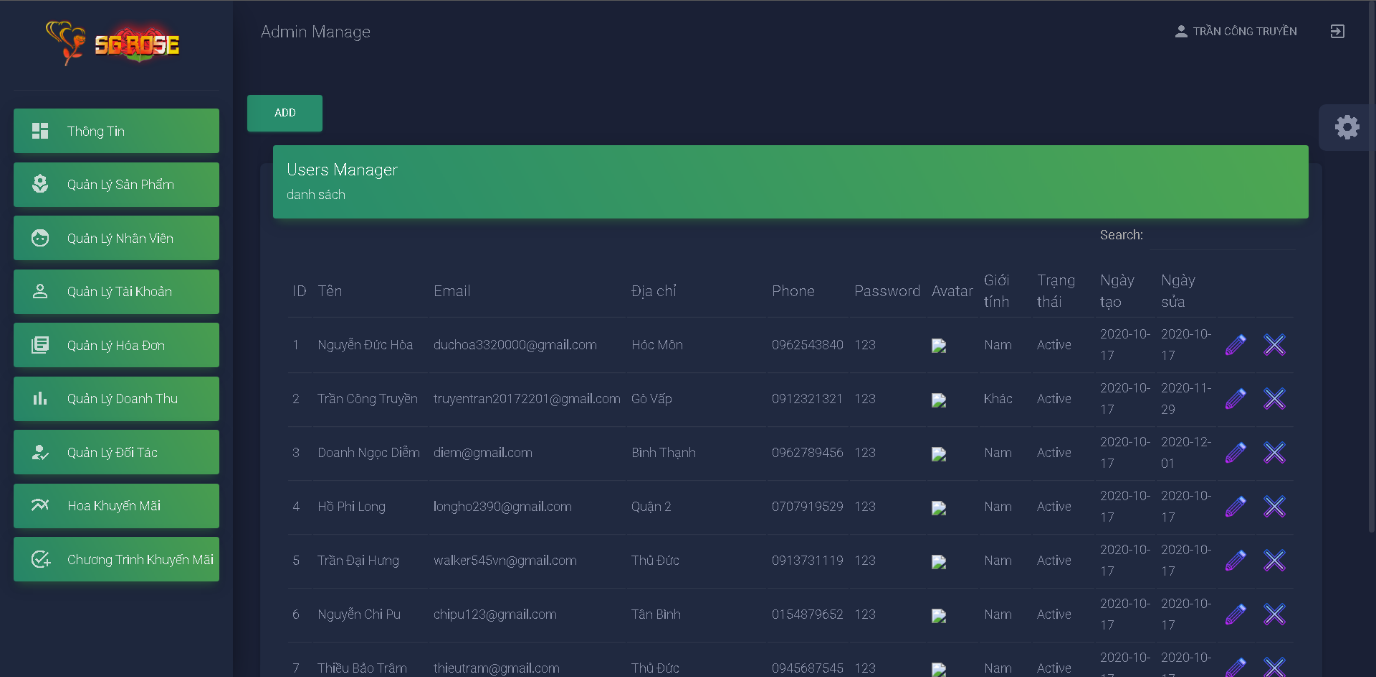
* + Mô tả: Thêm sản phẩm hoa
  + Hoạt động:
* Thêm thông tin sản phẩm hoa
* Chọn hình ảnh hoa cho phù hợp với sản phẩm hoa
* Nhấn [Insert]
* Quản lý nhân viên:



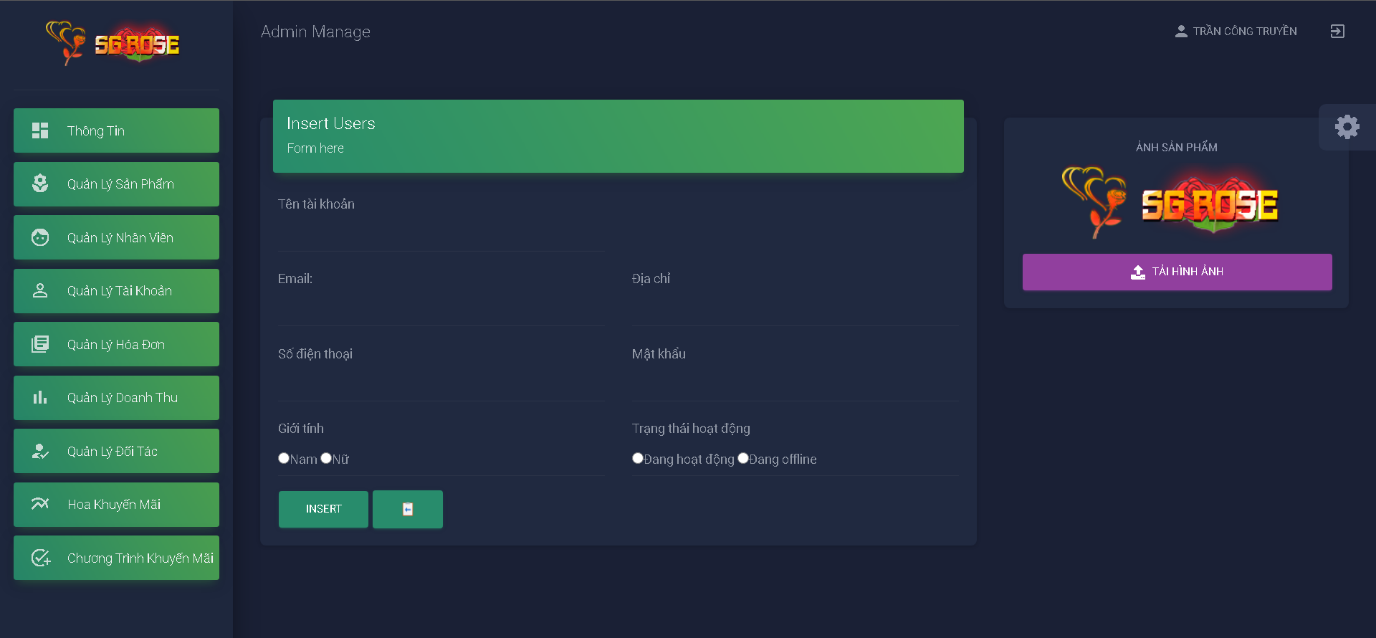
* + Mô tả: Hiển thị danh sách nhân viên
  + Hoạt động:
* Nhấn từ khóa từ [Search] để tìm nhân viên
* Chỉnh sửa hoặc xóa nhân viên
* Nhấn nút Add để thêm nhân viên



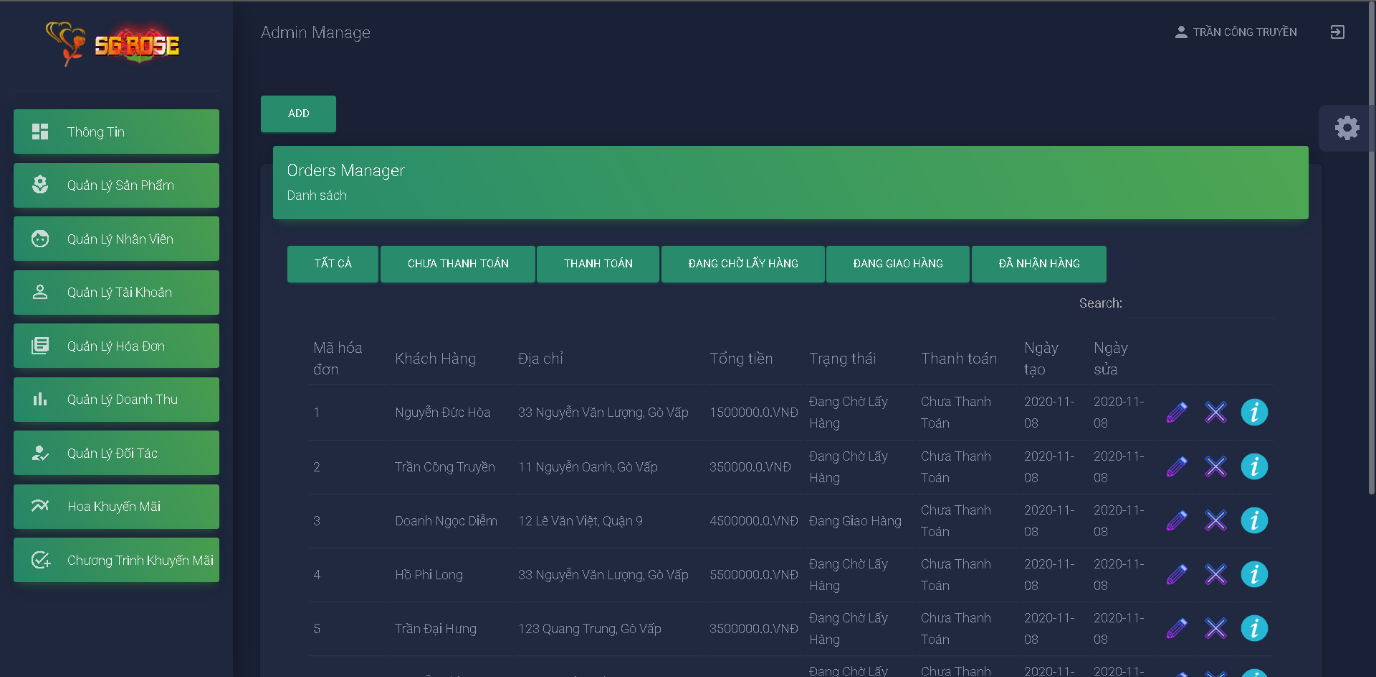
* + Mô tả: Thêm nhân viên
  + Hoạt động:
* Thêm thông tin nhân viên
* Chọn hình ảnh nhân viên sao cho phù hợp
* Nhấn [Insert]
* Quản lý tài khoản:



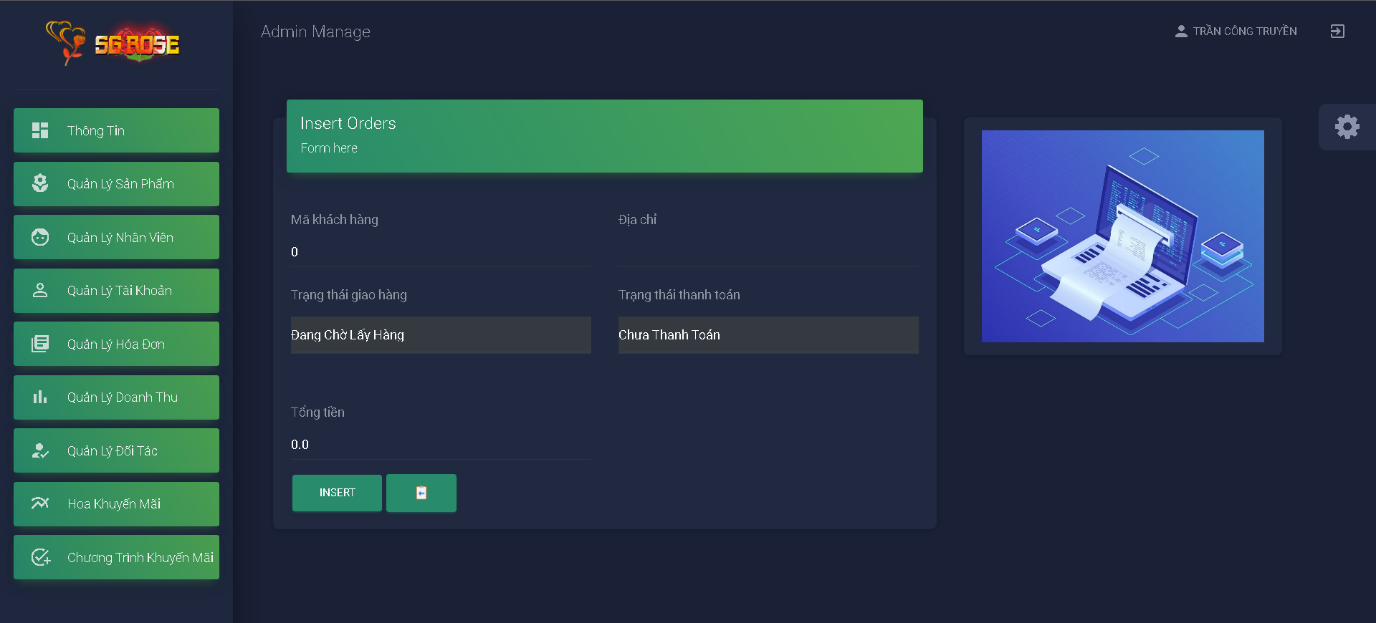
* + Mô tả: Hiển thị danh sách tài khoản
  + Hoạt động:
* Nhấn từ khóa từ [Search] để tìm tài khoản
* Chỉnh sửa hoặc xóa tài khoản
* Nhấn nút Add để thêm tài khoản



* + Mô tả: Thêm tài khoản
  + Hoạt động:
* Thêm thông tin tài khoản
* Chọn hình ảnh tài khoản
* Nhấn [Insert]
* Quản lý hóa đơn:



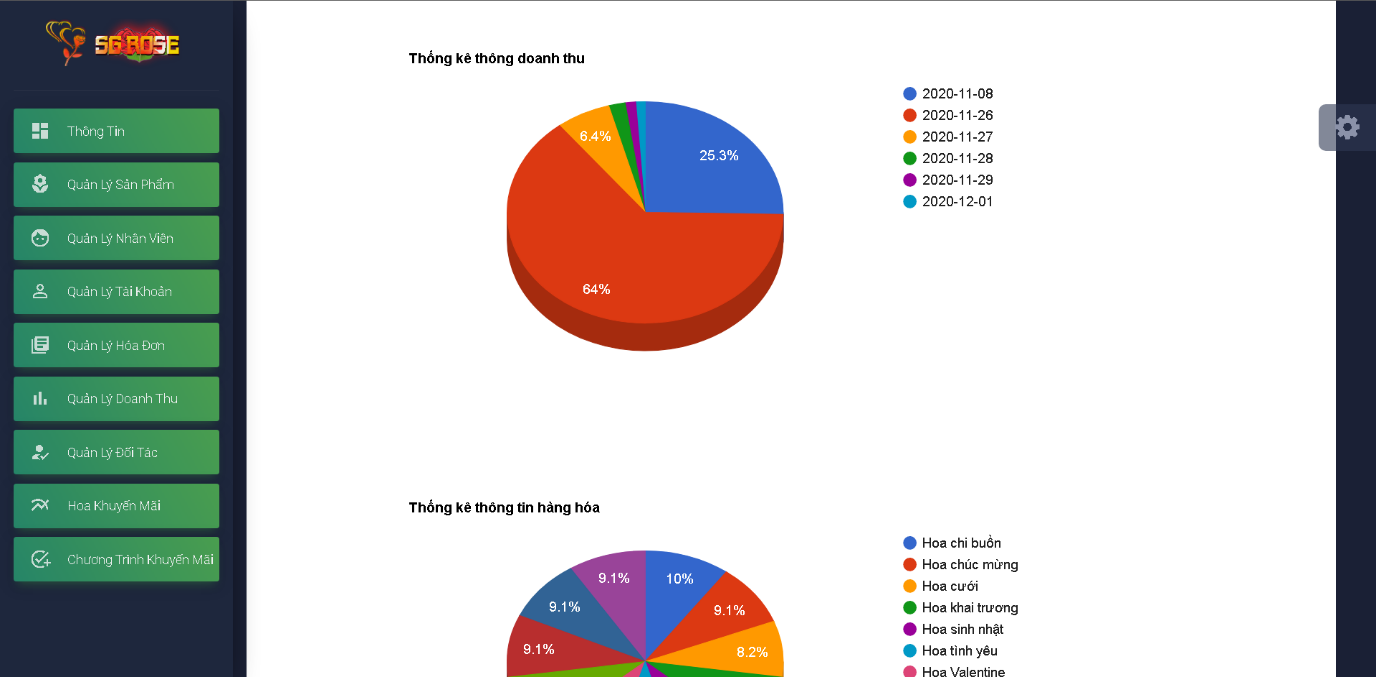
* + Mô tả: Hiển thị danh sách hóa đơn
  + Hoạt động:
* Nhấn từ khóa từ [Search] để tìm mã hóa đơn
* Chỉnh sửa hoặc xóa hóa đơn
* Nhấn nút Add để thêm hóa đơn



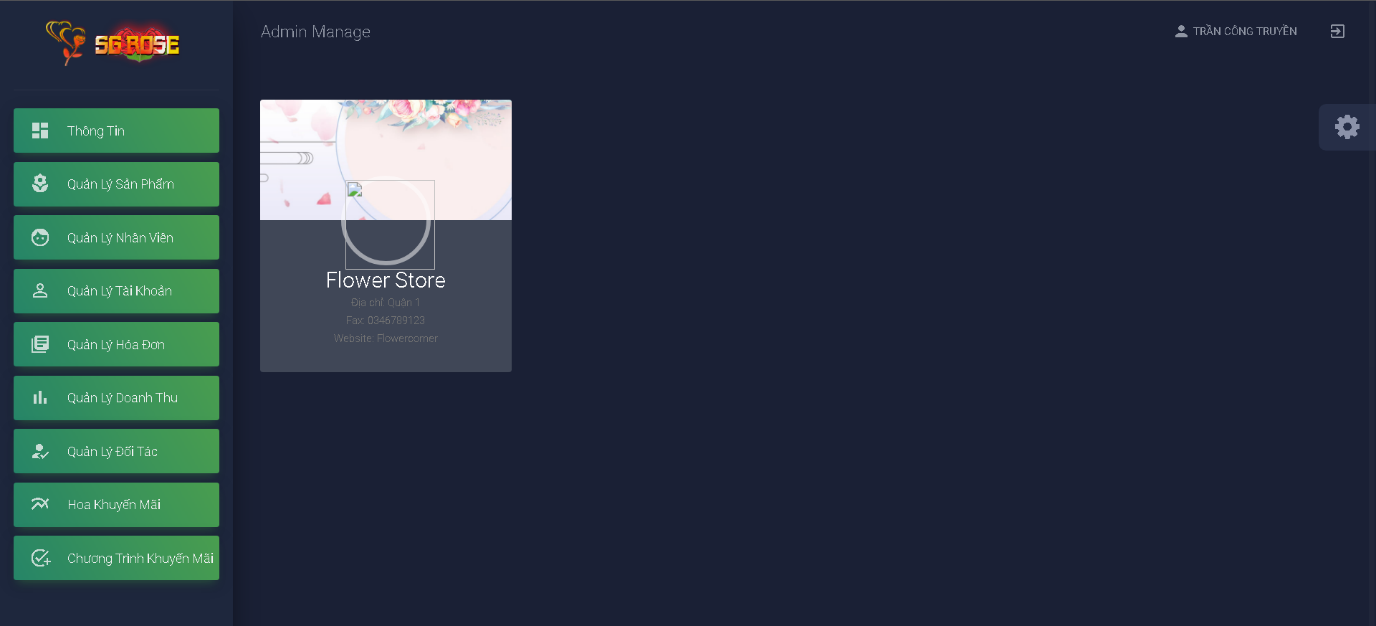
* + Mô tả: Thêm hóa đơn
  + Hoạt động:
* Thêm thông tin hóa đơn
* Nhấn [Insert]
* Quản lý doanh thu:



* + Mô tả: Hiển thị danh sách doanh thu
  + Hoạt động:
* Thống kê sản phẩm



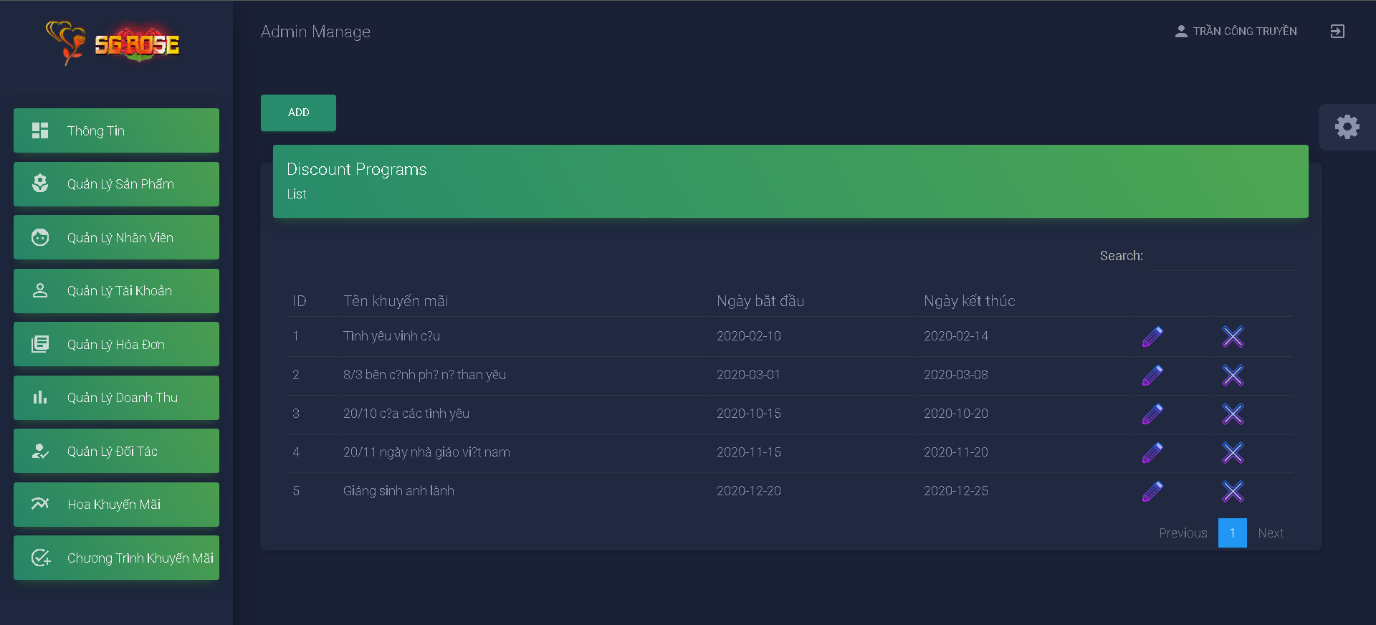
* + Mô tả: Hiển thị biểu đồ
  + Hoạt động:
* Hiển thị thống kê doanh thu và thông tin hàng hóa
* Quản lý đối tác:



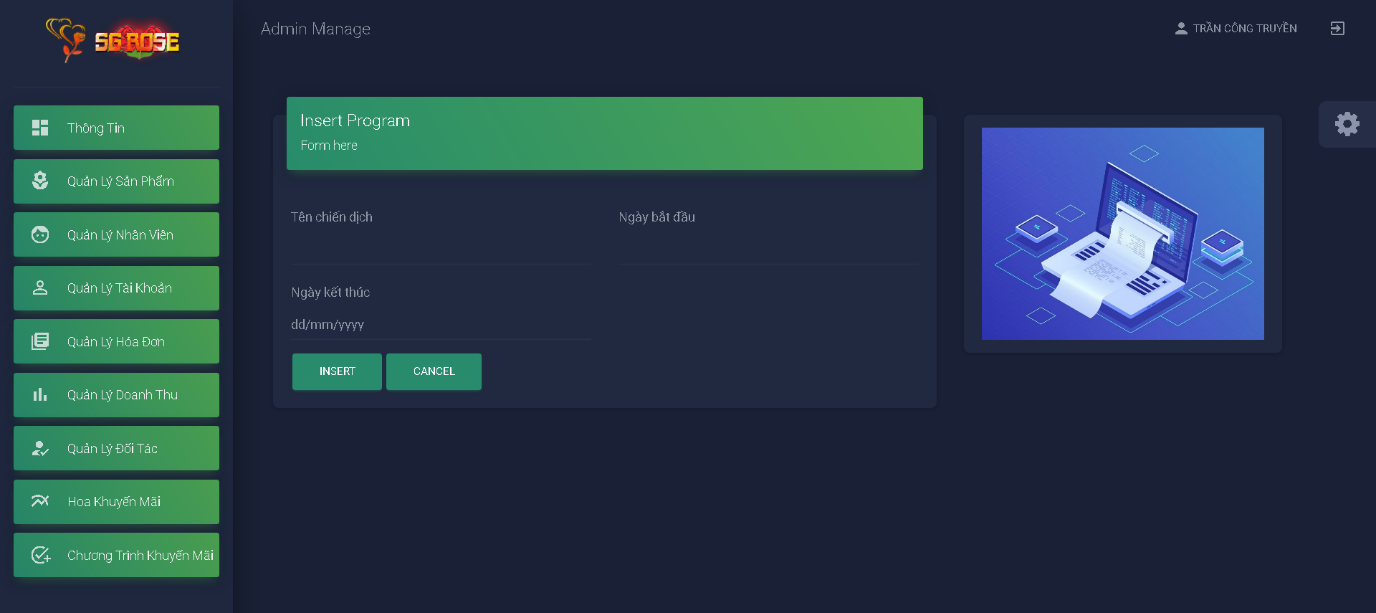
* + Mô tả: Hiển thị đối tác
  + Hoạt động:
* Hiển thị thông tin đối tác
* Hoa khuyến mãi:



* + Mô tả: Hiển thị hoa khuyến mãi
  + Hoạt động:
* Hiển thị thông tin hoa khuyến mãi
* Chương trình khuyến mãi:



* + Mô tả: Hiển thị chương trình khuyến mãi
  + Hoạt động:
* Hiển thị thông tin chương trình khuyến mãi
* Chỉnh sửa và xóa chương trình khuyến mãi
* Nhấn nút Add để thêm chương trình khuyến mãi



* + Mô tả: Thêm chương trình khuyến mãi
  + Hoạt động:
* Thêm thông tin chương trình khuyến mãi
* Nhấn [Insert]

## **Thiết kế thư viện cần thiết cho dự án:**

1. Giới thiệu JavaEE:

* "Java EE" tên đầy đủ là "Java Enterprise Edition" dịch sang tiếng việt có nghĩa là là "Java Phiên Bản Doanh Nghiệp", nó là một nền tảng (platform) dành cho việc xây dựng các ứng dụng (application) cấp doanh nghiệp.
* Java EE tập hợp nhiều công nghệ kỹ thuật và phối hợp chúng với nhau, việc này sẽ làm giảm đáng kể chi phí và độ phức tạp cho việc xây dựng, phát triển, triển khai, quản lý các tầng làm việc (managing multi-tier), cho các ứng dụng máy chủ trung tâm.
* Java EE được xây dựng dưa trên nền tảng Java SE và có thêm các tập APIs (application programming interfaces) để phát triển các ứng dựng phái máy chủ một cách nhanh chóng(running), linh động(portable), mạnh mẽ(rob), khả năng mở rộng cao, đáng tinh cậy và bảo mật.

1. Cách JavaEE hoạt động:

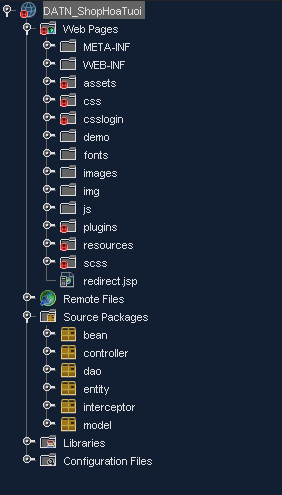
* **HTTP client technology**: Công nghệ này sử dụng để xử lý các máy khách dựa trên HTTP, J2EE cũng bao gồm API Java cho lập trình WebSocket, API để xử lý JSON, API JSF và Servlet và JSP Standard Tag Library (JSTL).
* **Công nghệ truy cập CSDL và tài nguyên**: Các công nghệ này sử dụng để tương tác với các hệ thống bên ngoài và  hệ thống Back end. Java EE bao gồm JavaMail, Standard conenctor Architecture, Java Message Service (JMS) API và Java Transaction API (JTA).
* **Công nghệ REST và Web Service**: Để giúp lập trình và triển khai các dịch vụ web dựa trên REST-, SOAP-, XML và JSON, các API Java cho Dịch vụ web RESTful (JAX-RS) và các dịch vụ web dựa trên XML (JAX-WS), cùng với các API cho thông báo XML và đăng ký XML (JAXR).
* **Java EE security and container management**: Để triển khai bảo mật Java EE tùy chỉnh và quản lý các thùng chứa Java EE, các lập trình viên phần mềm có quyền truy cập vào Java Authorization Contract cho các Container và Java Authentication Service Provider Interface cho các container.

## **Thiết kế mô hình triển khai theo mô hình MVC:**

1. Giới thiệu mô hình MVC:

* MVC là từ viết tắt bởi 3 từ Model – View – Controller. Đây là mô hình thiết kế sử dụng trong kỹ thuật phần mềm. Mô hình source code thành 3 phần, tương ứng mỗi từ. Mỗi từ tương ứng với một hoạt động tách biệt trong một mô hình:
* Model (M): Là bộ phận có chức năng lưu trữ toàn bộ dữ liệu của ứng dụng. Bộ phận này là một cầu nối giữa 2 thành phần bên dưới là View và Controller. Model thể hiện dưới hình thức là một cơ sở dữ liệu hoặc có khi chỉ đơn giản là một file XML bình thường. Model thể hiện rõ các thao tác với cơ sở dữ liệu như cho phép xem, truy xuất, xử lý dữ liệu,…
* View (V): Đây là phần giao diện (theme) dành cho người sử dụng. Nơi mà người dùng có thể lấy được thông tin dữ liệu của MVC thông qua các thao tác truy vấn như tìm kiếm hoặc sử dụng thông qua các website.
* Controller (C): Bộ phận có nhiệm vụ xử lý các yêu cầu người dùng đưa đến thông qua view. Từ đó, C đưa ra dữ liệu phù hợp với người dùng. Bên cạnh đó, Controller còn có chức năng kết nối với model.

1. Tổ chức code theo mô hình MVC trong code:



1. Viết mã cho phần Model:

* Employee

|  |
| --- |
| package bean;  public class Employee {  private int id;  private String tagName;  public Employee(){}  public Employee(int id, String tagName) {  this.id = id;  this.tagName = tagName;  }  public int getId() {  return id;  }  public void setId(int id) {  this.id = id;  }  public String getTagName() {  return tagName;  }  public void setTagName(String tagName) {  this.tagName = tagName;  }      } |

* ItemBean

|  |
| --- |
| package bean;  import entity.Flower;  public class ItemBean {  private Flower product;  private int quantity;  public ItemBean() {  }  public ItemBean(Flower product, int quantity) {  this.product = product;  this.quantity = quantity;  }  public Flower getProduct() {  return product;  }  public void setProduct(Flower product) {  this.product = product;  }  public int getQuantity() {  return quantity;  }  public void setQuantity(int quantity) {  this.quantity = quantity;  }  } |

* OrdersDetails

|  |
| --- |
| package bean;  import java.util.Date;  public class OrdersDetails {    private int id;    private int orderid;    private int flowerId;    private int price;  private int amount;  private int totalprice;    private Date createdtime;    private Date updatedtime;      public int getId() {  return id;  }  public void setId(int Id) {  this.id = Id;  }  public int getOrderid() {  return orderid;  }  public void setOrderid(int OrderId) {  this.orderid = OrderId;  }  public int getFlowerId() {  return flowerId;  }  public void setFlowerId(int FlowerId) {  this.flowerId = FlowerId;  }    public int getPrice() {  return price;  }  public void setPrice(int Price) {  this.price = Price;  }  public int getAmount() {  return amount;  }  public void setAmount(int Amount) {  this.amount = Amount;  }  public int getTotalprice() {  return totalprice;  }  public void setTotalprice(int Totalprice) {  this.totalprice = Totalprice;  }  public Date getCreatedtime() {  return createdtime;  }  public void setCreatedtime(Date Createdtime) {  this.createdtime = Createdtime;  }  public Date getUpdatedtime() {  return updatedtime;  }  public void setUpdatedtime(Date Updatedtime) {  this.updatedtime = Updatedtime;  }    } |

* Report

|  |
| --- |
| package bean;  public class Report {  String group;  Double sum, avg, min, max;  Integer count;  public Report() {  }  public Report(String group, Double sum, Double avg, Double min, Double max, Integer count) {  this.group = group;  this.sum = sum;  this.avg = avg;  this.min = min;  this.max = max;  this.count = count;  }      public String getGroup() {  return group;  }  public void setGroup(String group) {  this.group = group;  }  public Double getSum() {  return sum;  }  public void setSum(Double sum) {  this.sum = sum;  }  public Double getAvg() {  return avg;  }  public void setAvg(Double avg) {  this.avg = avg;  }  public Double getMin() {  return min;  }  public void setMin(Double min) {  this.min = min;  }  public Double getMax() {  return max;  }  public void setMax(Double max) {  this.max = max;  }  public Integer getCount() {  return count;  }  public void setCount(Integer count) {  this.count = count;  }  } |

* CartModel

|  |
| --- |
| package model;  import bean.ItemBean;  import entity.Flower;  import java.util.ArrayList;  import java.util.HashMap;  public class CartModel {    HashMap<Integer, ItemBean> cart;  public CartModel() {  cart = new HashMap<>();  }  public void addProduct(int productid){  if(cart.containsKey(productid)){  ItemBean item = cart.get(productid);  item.setQuantity(item.getQuantity() + 1);    } else {  Flower product = new FlowerModel().getProductByProductid(productid);  ItemBean item = new ItemBean(product,1);  cart.put(productid, item);  }  }    public ArrayList<ItemBean> getListItems() {  ArrayList<ItemBean> listItems = new ArrayList<>();  for (ItemBean i : cart.values()) {  listItems.add(i);  }  return listItems;  }    public void plusProduct(int productid){  ItemBean item = cart.get(productid);  item.setQuantity(item.getQuantity() + 1);  }    public void minusProduct(int productid){  ItemBean item = cart.get(productid);  item.setQuantity(item.getQuantity() - 1);  if(item.getQuantity() == 0){  cart.remove(productid);  }  }    public void removeProduct(int productid) {  cart.remove(productid);  }  public void removeAllProduct() {  cart.clear();  }  public double getTotal() {  double total = 0;  for (ItemBean i : cart.values()) {  total = total + i.getQuantity()\* i.getProduct().getPrice();  }  return total;  }  } |

* DateHelper

|  |
| --- |
| package model;  import java.text.ParseException;  import java.text.SimpleDateFormat;  import java.util.Date;  public class DateHelper {  static final SimpleDateFormat DATE\_FORMATER = new SimpleDateFormat("MM/dd/yyyy");  public static Date toDate(String date, String...pattern) {  try {  if(pattern.length > 0){  DATE\_FORMATER.applyPattern(pattern[0]);  }  if(date == null){  return DateHelper.now();  }  return DATE\_FORMATER.parse(date);  }  catch (ParseException ex) {  throw new RuntimeException(ex);  }  }  public static String toString(Date date, String...pattern) {  if(pattern.length > 0){  DATE\_FORMATER.applyPattern(pattern[0]);  }  if(date == null){  date = DateHelper.now();  }  return DATE\_FORMATER.format(date);  }  public static Date now() {  return new Date();  }  public static Date addDays(Date date, int days) {  date.setTime(date.getTime() + days\*24\*60\*60\*1000);  return date;  }  public static Date add(int days) {  Date now = DateHelper.now();  now.setTime(now.getTime() + days\*24\*60\*60\*1000);  return now;  }  } |

* FlowerModel

|  |
| --- |
| package model;  import entity.Flower;  import entity.TypesOfFlower;  import java.sql.Connection;  import java.sql.PreparedStatement;  import java.sql.ResultSet;  public class FlowerModel {    public FlowerModel(){    }  public Flower getProductByProductid(int productid) {  Flower product = null;    Connection cn = new MyConnect().getcn();  if (cn == null) {  return null;  }  try {  String sql = "select \* from Flowers where id not in (SELECT flowerID FROM FlowerPromotionPrograms) and id = ?";  String sqlpromotion = "select f.id, f.name,f.typeid, f.amount, ((f.price - p.TypePrice) \* ((100 - p.typePercent)/100)) as Price, f.image, f.Notes, f.Createdtime, f.updatedtime From FlowerPromotionPrograms p inner join Flowers f on f.id=p.FlowerID where f.id = ?";  PreparedStatement ps = cn.prepareStatement(sql);  PreparedStatement ps2 = cn.prepareStatement(sqlpromotion);  ps.setInt(1, productid);  ps2.setInt(1, productid);  ResultSet rs = ps.executeQuery();    ResultSet rs2 = ps2.executeQuery();  TypesOfFlower type = new TypesOfFlower();  if (rs.next()) {  product = new Flower(rs.getInt(1), rs.getString(2), type, rs.getInt(4), rs.getInt(5), rs.getString(6), rs.getString(7), rs.getDate(8), rs.getDate(9));  } else if(rs2.next()) {  product = new Flower(rs2.getInt(1), rs2.getString(2), type, rs2.getInt(4), rs2.getInt(5), rs2.getString(6), rs2.getString(7), rs2.getDate(8), rs2.getDate(9));  }  } catch (Exception e) {  e.printStackTrace();  }  return product;  }    } |

* MyConnect

|  |
| --- |
| package model;  import java.sql.Connection;  import java.sql.DriverManager;  import java.sql.SQLException;  public class MyConnect {  String servername;  String port;  String db\_name;  String db\_user;  String db\_pass;  public MyConnect() {  servername = "127.0.0.1";  port = "1433";  db\_name = "SG\_Rose";  db\_user = "sa";  db\_pass = "1234";  }  public Connection getcn() {  Connection cn = null;  try {  String db\_url = "jdbc:sqlserver://" + servername + ":" + port + ";databasename=" + db\_name;  Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");  cn = DriverManager.getConnection(db\_url, db\_user, db\_pass);  } catch (ClassNotFoundException | SQLException ex) {  System.out.println(ex.getMessage());  }  return cn;  }  } |

* OrderModel

|  |
| --- |
| package model;  import java.sql.\*;  import java.text.SimpleDateFormat;  import java.util.Calendar;  import org.hibernate.SessionFactory;  import org.springframework.beans.factory.annotation.Autowired;  import org.springframework.transaction.annotation.Transactional;  @Transactional  public class OrderModel {    @Autowired  SessionFactory factory;    CartModel cartmodel = new CartModel();    public OrderModel(){}    public int buyFlower(int userid, String address, int totalmoney, String name, String numberphone){  int kq = 0;  int status = 1;  int ispaid = 1;  SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");  Calendar cal = Calendar.getInstance();  String createdtime = dateFormat.format(cal.getTime());  Connection cn = new MyConnect().getcn();  if (cn == null) {  return -1;  }  try {  String sql = "insert into Orders(UserId, Address, TotalMoney, Status, Ispaid,Createdtime, Name, NumberPhone) values(?,?,?,?,?,?,?,?)";  PreparedStatement ps = cn.prepareStatement(sql);  ps.setInt(1, userid);  ps.setString(2, address);  ps.setInt(3, totalmoney);  ps.setInt(4, status);  ps.setInt(5, ispaid);  ps.setDate(6, Date.valueOf(createdtime));  ps.setString(7, name);  ps.setString(8, numberphone);    kq = ps.executeUpdate();  } catch (Exception e) {  e.printStackTrace();  }  return kq;  }    public int buyFlowerGuest(String address, int totalmoney, String name, String numberphone){  int kq = 0;  int status = 1;  int ispaid = 1;  SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");  Calendar cal = Calendar.getInstance();  String createdtime = dateFormat.format(cal.getTime());  Connection cn = new MyConnect().getcn();  if (cn == null) {  return -1;  }  try {  String sql = "insert into Orders(Address, TotalMoney, Status, Ispaid,Createdtime, Name, NumberPhone) values(?,?,?,?,?,?,?)";  PreparedStatement ps = cn.prepareStatement(sql);  ps.setString(1, address);  ps.setInt(2, totalmoney);  ps.setInt(3, status);  ps.setInt(4, ispaid);  ps.setDate(5, Date.valueOf(createdtime));  ps.setString(6, name);  ps.setString(7, numberphone);    kq = ps.executeUpdate();  } catch (Exception e) {  e.printStackTrace();  }  return kq;  }    public int getId\_Lastest() {  int idLastest = 0;  Connection cn = new MyConnect().getcn();  if (cn == null) {  return -1;  }  try {  String sql = "Select max(Id) from Orders";  PreparedStatement ps = cn.prepareStatement(sql);  ResultSet rs = ps.executeQuery();  if (rs.next()) {  idLastest = rs.getInt(1);  }  ps.close();  cn.close();  } catch (Exception e) {  e.printStackTrace();  }  return idLastest;  }  } |

# KIỂM THỬ

## **Thiết kế CSDL:**

### **Khảo sát rủi ro:**

* Công cụ test không tối ưu, test thủ công
* Test case còn thiếu sót
* Nguồn nhân lực ít, thiếu thời gian
* Bảo mật kém

### **Phạm vi:**

* Form Đăng nhập, Đăng ký, Quên mật khẩu
* Form Khách hàng
* Form Giỏ hàng
* Form quản lý của nhân viên, Admin

### **Nguồn lực:**

|  |  |  |
| --- | --- | --- |
| **Thành viên** | **Chức vụ** | **Nhiệm vụ** |
| Trần Đại Hưng | Thành viên | Test Design |
| Hồ Phi Long | Thành viên | Test Case |

## **Test case và kết quả test:**

1. Test Đăng nhập:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TC ID | Kịch bản | Các bước | Dữ liệu | Công cụ kiểm thử | Kết quả dự kiến | Kết quả thực tế | Pass/Fail |
| TC 01 | Đăng nhập tài khoản dành cho Khách hàng | Truy cập:  Localhost:8084/ DATN\_ShopHoaTuoi/login.htm  Nhấn nút đăng nhập.  Nhập email và password. | Email:  walker545vn@gmail.com  Password:  123 | Bằng tay | Đăng nhập thành công | Như mong đợi | Pass |
| TC 02 | Đăng nhập dành cho nhân viên, admin truy cập | Truy cập:  Localhost:8084/ DATN\_ShopHoaTuoi/login.htm  Nhấn nút đăng nhập  Nhập SĐT và password | SĐT:  0962086560  Password:  123 | Bằng tay | Đăng nhập thành công | Như mong đợi | Pass |
| TC 03 | Đăng nhập sai tài khoản cho Khách hàng | Truy cập:  Localhost:8084/ DATN\_ShopHoaTuoi/login.htm  Nhấn nút đăng nhập.  Nhập email và password | Email:  walker545vn@ymail.com  Password:  123 | Bằng tay | Đăng nhập thất bại | Như mong đợi | Pass |
| TC 04 | Đăng nhập sai tài khoản cho nhân viên, admin | Truy cập:  Localhost:8084/ DATN\_ShopHoaTuoi/login.htm  Nhấn nút đăng nhập.  Nhập SĐT và password | SĐT:  0962086561  Password:  123 | Bằng tay | Đăng nhập thất bại | Như mong đợi | Pass |
| TC 05 | Đăng nhập sai thông tin cho Khách hàng | Truy cập:  Localhost:8084/ DATN\_ShopHoaTuoi/login.htm  Nhấn nút đăng nhập.  Nhập email và password | Email:  walker545vn@gmail.com  Password:  1234 | Bằng tay | Đăng nhập thất bại | Như mong đợi | Pass |

1. Test Đăng ký:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TC ID | Kịch bản | Các bước | Dữ liệu | Công cụ kiểm thử | Kết quả dự kiến | Kết quả thực tế | Pass/Fail |
| TC 06 | Đăng ký tài khoản, người dùng nhập đầy đủ thông tin yêu cầu. | Truy cập:  Localhost:8084/ DATN\_ShopHoaTuoi/Register.htm  Nhập các thông tin yêu cầu.  Nhấn nút Đăng ký. |  | Bằng tay | Đăng ký thành công | Như mong đợi | Pass |
| TC 07 | Đăng ký tài khoản, người dùng để trống 1 hoặc nhiều trường bắt buộc | Truy cập:  Localhost:8084/ DATN\_ShopHoaTuoi/Register.htm  Nhập các thông tin không yêu cầu.  Nhấn nút Đăng ký. |  | Bằng tay | Đăng ký thất bại | Như mong đợi | Pass |
| TC 08 | Đăng ký tài khoản, người dùng nhập đầy đủ thông tin yêu cầu. Tuy nhiên email đã tồn tại. | Truy cập:  Localhost:8084/ DATN\_ShopHoaTuoi/Register.htm  Nhập các thông tin không yêu cầu.  Nhấn nút Đăng ký. |  | Bằng tay | Đăng ký thất bại | Như mong đợi | Pass |

1. Test Đăng xuất:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TC ID | Kịch bản | Các bước | Dữ liệu | Công cụ kiểm thử | Kết quả dự kiến | Kết quả thực tế | Pass/Fail |
| TC 09 | Đăng xuất tài khoản | Localhost:8084/ DATN\_ShopHoaTuoi/user/home.htm  Nhập từ khóa từ tên tài khoản  Chọn từ khóa Đăng xuất |  | Bằng tay | Đăng xuất thất bại | Không như mong đợi | Fail |

1. Test Giỏ hàng

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TC ID | Kịch bản | Các bước | Dữ liệu | Công cụ kiểm thử | Kết quả dự kiến | Kết quả thực tế | Pass/Fail |
| TC 10 | Kiểm tra việc đăng nhập trước khi khách hàng đặt hàng không có thông tin.. | Truy cập  localhost:8084/  khachhang/cart | Email: dattq201098@gmail.com  Password: coko23 | Bằng tay | Đúng như yêu đặt ra. | Như mong đợi. | Pass |
| TC 11 | Kiểm tra việc không đăng nhập trước khi khách hàng đặt hàng không có thông tin. | Truy cập  localhost:8084/  DATN\_ShopHoaTuoi/user/checkouts.htm | Email: walker545vn@gmail.com  Password: 123 | Bằng tay | Đúng như yêu đặt ra. | Như mong đợi. | Pass |
| TC 12 | Kiểm tra việc thêm hoa vào giỏ hàng. | Truy cập:  localhost:8084/DATN\_ShopHoaTuoi/user/buy/79.htm?yeucau=muasp&txtmasp=79 |  | Bằng tay | Đúng như yêu đặt ra. | Như mong đợi. | Pass |
| TC 13 | Kiểm tra  Việc tăng giảm, xóa sản phẩm trong giỏ hàng. | Truy cập  localhost:8084/DATN\_ShopHoaTuoi/user/buy/5.htm?yeucau=plus&txtmasp=5 |  | Bằng tay | Đúng như yêu đặt ra. | Như mong đợi. | Pass |

1. Test Quản lý nhân viên, Admin

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TC ID | Kịch bản | Các bước | Dữ liệu | Công cụ kiểm thử | Kết quả dự kiến | Kết quả thực tế | Pass/Fail |
| TC 14 | Tạo tài khoản mới, nhập đầy đủ thông tin,. | Truy cập localhost:8084/DATN\_ShopHoaTuoi/admin/staff/index.htm  Đăng nhập bằng tài khoản admin.  Vào trang nhân viên .  Nhấn nút Tạo người dùng.  Nhập đầy đủ thông tin.  Nhấn nút Tạo mới. |  | Bằng tay | Tạo tài khoản mới thành công. | Như mong đợi. | Pass |
| TC 15 | Tạo tài khoản mới, nhập thiếu thông tin họ tên. | Truy cập localhost:8084/DATN\_ShopHoaTuoi/admin/staff/index.htm  Đăng nhập bằng tài khoản admin.  Vào trang nhân viên .  Nhấn nút Tạo người dùng.  Nhập đầy đủ thông tin.  Nhấn nút Tạo mới. |  | Bằng tay | Tạo tài khoản mới thành công | Như mong đợi. | Pass |
| TC 16 | Kiểm thử các điều khiển | Truy cập  localhost:8084/  DATN\_ShopHoaTuoi/admin/dashboardmanage.htm |  | Bằng tay | Đúng như yêu đặt ra. | Như mong đợi. | Pass |
| TC 17 | Kiêm thử các nội dụng trên form ( màu sắc, các tab hiện thị …) |  |  | Bằng tay | Đúng như yêu đặt ra. | Như mong đợi. | Pass |
| TC 18 | Kiểm thử  Chức năng hiện thị thông tin và chỉnh sửa thông tin của nhân viên… |  |  |  | Đúng như yêu đặt ra. | Như mong đợi. | Pass |
| TC 19 | Kiểm thử chúc năng Thống kê hoa trong form Nhân viên có hiện thị Thông tin hoa | Truy Cập:  Localhost:8084/DATN\_ShopHoaTuoi/admin/report.htm |  | Bằng tay | Đúng như yêu đặt ra. | Như mong đợi. | Pass |
| TC 20 | Kiểm thử chúc năng thêm hoa mới trong form thống kê của Admin | Truy Cập:  Localhost:8084/DATN\_ShopHoaTuoi/admin/report.htm |  | Bằng tay | Đúng như yêu đặt ra. | Như mong đợi. | Pass |
| TC 21 | Kiểm thử chúc năng Thống kê hoa trong fom Admin có hiện thị danh mục hoa | Truy Cập:  Localhost:8084/DATN\_ShopHoaTuoi/admin/reportChart.htm |  | Bằng tay | Đúng như yêu đặt ra. | Như mong đợi. | Pass |

1. Test Khách hàng

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TC ID | Kịch bản | Các bước | Dữ liệu | Công cụ kiểm thử | Kết quả dự kiến | Kết quả thực tế | Pass/Fail |
| TC 22 | Kiểm tra giao diện trang web | Truy cập  localhost:8084/  DATN\_ShopHoaTuoi/home.htm |  | Bằng tay | Đúng như yêu cầu đặt ra. | Như mong đợi. | Pass |
| TC 23 | Kiểm tra các điều khiển |  |  | Bằng tay | Đúng như yêu đặt ra. | Như mong đợi. | Pass |

# HƯỚNG DẪN TRIỂN KHAI VÀ SỬ DỤNG

## **Hướng dẫn cài đặt:**

1. Cài đặt Netbeans 8.2 with JDK 8u111:

* [JDK 8u111 with NetBeans 8.2 (oracle.com)](https://www.oracle.com/technetwork/java/javase/downloads/jdk-netbeans-jsp-3413139-esa.html)
* Cài đặt và cấu hình Netbeans

1. Cài đặt SQL Server:

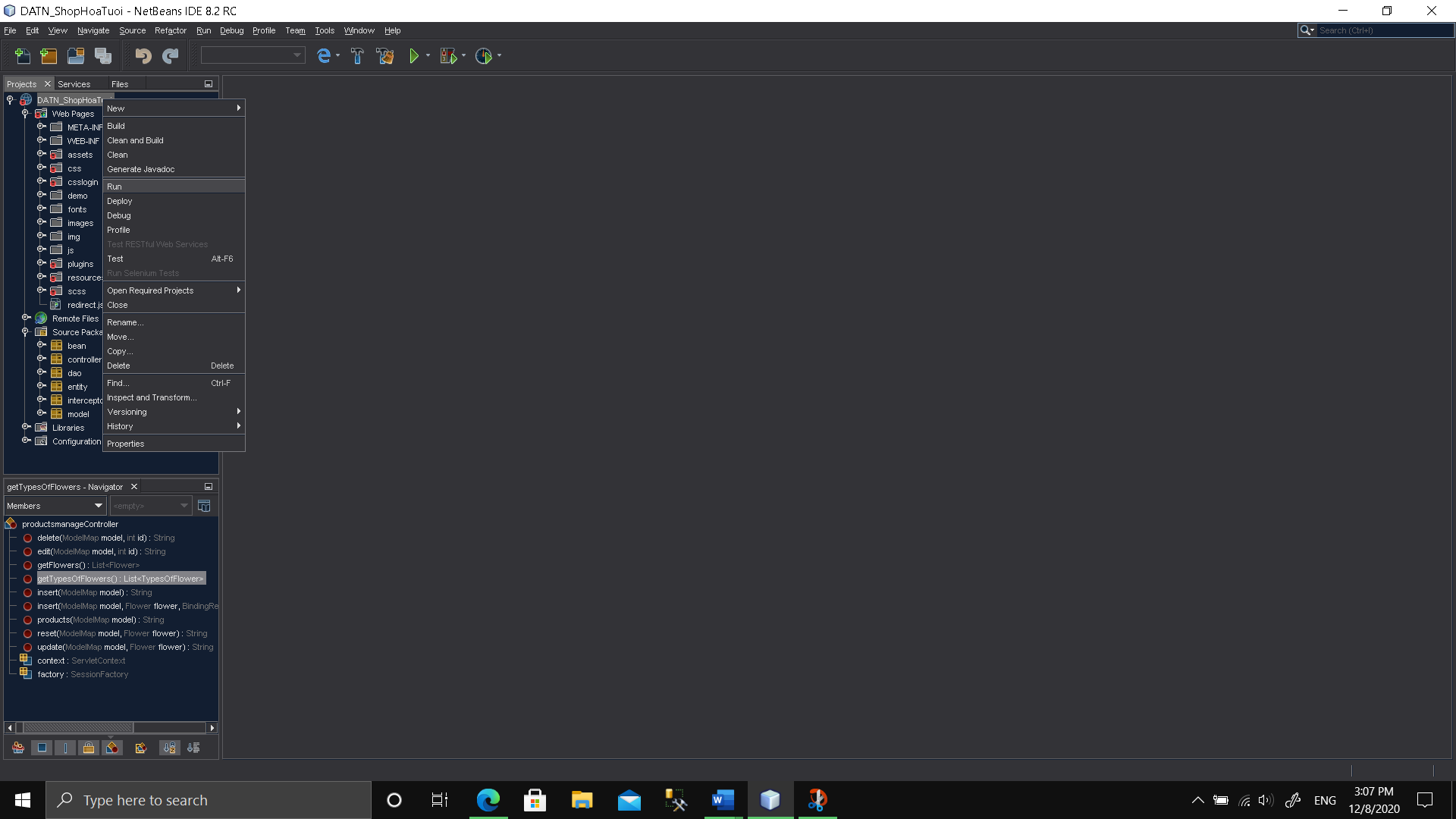
* [Download Microsoft® SQL Server® 2014 Express from Official Microsoft Download Center](https://www.microsoft.com/en-us/download/details.aspx?id=42299)
* Cài đặt và cấu hình SQL Server:
  + Login: sa
  + Password: 1234

1. Tạo CSDL từ file SQL
2. Add project vào Eclipse:

* Open => Open Projects from File System…
* Import source: Đường dẫn đến file chứa project
* Finish.

## **Hướng dẫn sử dụng:**

1. Chạy dự án bằng Java EE:



1. Sử dụng trình duyệt web truy cập vào đường dẫn bên dưới:

* Trang Khách hàng: http://localhost:8084/DATN\_ShopHoaTuoi/home.htm
* Trang Admin, nhân viên:

http://localhost:8084/DATN\_ShopHoaTuoi/admin/dashboardmanage.htm

# KẾT LUẬN

## **Kết luận:**

Hệ thống Website “Kinh doanh hoa SG-Rose” được xây dựng nhằm mục đích giúp doanh nghiệp vừa và nhỏ có thể: Quản lý, vận hành và thống kê hoạt động kinh doanh hoa online. Tạo ra nơi cung cấp thông tin hoa và mua hoa online, đồng thời việc tìm kiếm, thanh toán nhanh gọn.

Trong quá trình nghiên cứu và hoàn thiện dự án “Website kinh doanh hoa SG-Rose” thì nhóm đã nỗ lực hoàn thiện hầu hết những chức năng cơ bản như: Đẩy dữ liệu từ database; Thêm, xóa, sửa các Bảng; Phân quyền đăng nhập; Thống kê doanh thu; Gởi mail tự động.

Thông qua đề tài, chúng tôi đạt được nhiều mục tiêu đặt ra và bổ sung rất nhiều kiến thức mới.

## **Khó khăn và thuận lợi:**

### Thuận lợi:

Hệ thống Website “Kinh doanh hoa SG-Rose” được xây dựng bằng những kiến thức từ các chương trình được học trong quá trình học tập tại trường Cao đẳng thực hành FPT Polytechnic HCM.

Được sự giúp đỡ, chỉ đạo tới từ GVHD và giảng viên của trường, cung cấp cho chúng tôi những kiến thức cần thiết cũng như tinh thần học tập độc lập và sáng tạo trong môn học. Được sự hợp tác và làm việc chăm chỉ của các thành viên trong nhóm để hoàn thành sản phẩm này.

### Khó khăn:

Trong quá trình thực hiện dự án “Website kinh doanh hoa SG-Rose”, nhóm chúng gặp một vài khó khăn như:

* Làm việc nhóm chưa hiệu quả và không có nhiều thời gian do các thành viên đa phần đi thực tập ở nhiều nơi và ảnh hưởng dịch “Covid 19”.
* Chưa khảo sát thực tế bên khách hàng thật sự cần gì về Website để bổ sung dự án. Đa phần, các chức năng lấy từ việc khảo sát Website đối thủ và nhóm đặt ra nên thiếu thực tế.
* Việc áp dụng những công nghệ và xu hướng lập trình cũng như việc tối ưu hóa hệ thống chưa được tốt do việc hạn chế về kiến thức cũng như kinh nghiệm lập trình.

## **Hướng phát triển:**

Từ những thuận lợi và khó khăn nêu bên trên, nhóm chúng tôi đã đưa ra nhiều giải pháp và hướng phát triển cho sản phẩm của mình:

* Về việc làm nhóm:
  + Các thành viên phải bổ sung những thiếu sót về kiến thức và chuyên môn.
  + Đầu tư thời gian và rèn luyện khả năng làm việc nhóm hiệu quả.
* Về hướng phát triển sản phẩm:
  + Chỉnh sửa giao diện đẹp mắt, trực quan, tối ưu hóa hiển thị và độ mượt mà của các trang người dùng.
  + Rà soát và bổ sung cơ sở dữ liệu, code, vận hành hệ thống,…
  + Bổ sung nhiều tính năng và công nghệ mới cho Website như:
    - Đăng nhập Website bằng tài khoản Google, Facebook,…
    - Lọc sản phẩm và gợi ý hoa.
    - Cho khách hàng bình luận và đánh giả.
    - Tạo nơi hỗ trợ giữa khách hàng.
    - Nâng cấp tài khoản để nhận ưu đãi cho Thành viên.
    - Thêm tính năng và các biểu đồ thống kê khác cho trang quản lý.
    - Bảo mật dữ liệu.
    - Thanh toán thông qua liên kết ngân hàng, ví điện tử.

# LỜI CẢM ƠN

Đầu tiên, chúng tôi xin chân thành cảm ơn đến các thầy cô, giảng viên trường cao đẳng Fpoly Polytechnic HCM nói chung, và khoa Công Nghệ Thông Tin, chuyên ngành Ứng Dụng Phần Mềm nói riêng vì đã hướng dẫn, cung cấp những kiến thức quan trọng về cả khía cạnh chuyên ngành và xã hội, giúp chúng tôi có được kinh nghiệm, kỹ năng trong công việc, học tập, cách sống thật đúng đắn.

Đặc biệt, chúng tôi xin chân thành cảm ơn sự hướng dẫn và chỉ bảo tận tình của thầy Nguyễn Thành Phước - người đã luôn hướng dẫn, quan tâm, động viên, thông cảm, tạo điều kiện thuận lợi cho nhóm rất nhiều trong quá trình thực hiện đề tài. Các định hướng và sự hiểu biết về khoa học của thầy chính là tiền đề để chúng tôi hoàn thành được đề tài này.

Bài báo cáo và ứng dụng có thể còn nhiều hạn chế. Kính mong các thầy cô góp ý để đề tài chúng tôi hoàn thiện hơn. Cho chúng tôi những lời khuyên để có thêm kinh nghiệm bước vào tương lai.

Chúng tôi xin chân thành cảm ơn!

# TÀI LIỆU THAM KHẢO

## **Tiếng việt:**

* Các hướng dẫn sử dụng Bootstrap: <https://o7planning.org/vi/11745/bootstrap>
* Hướng dẫn lập trình Spring MVC cho người mới bắt đầu:

[Hướng dẫn lập trình Spring MVC cho người mới bắt đầu - Hello Spring 4 MVC (o7planning.org)](https://o7planning.org/vi/10129/huong-dan-lap-trinh-spring-mvc-cho-nguoi-moi-bat-dau)

## **Tiếng anh:**

* <https://getbootstrap.com/>
* Spring Framework: <https://spring.io/>
* [JavaScript Tutorial (w3schools.com)](https://www.w3schools.com/js/DEFAULT.asp)
* Template khách hàng: [E Shopper Free Website Template | Free CSS Templates | Free CSS (free-css.com)](https://www.free-css.com/free-css-templates/page203/e-shopper)
* Template admin, nhân viên: [Material Dashboard Dark Edition: Free Bootstrap 4 Material Admin @ Creative Tim (creative-tim.com)](https://www.creative-tim.com/product/material-dashboard-dark)