

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**Khoa Hệ Thống Thông Tin**



**Sinh viên thực hiện:**  
Nguyễn Ngọc Công - 15520069

Nguyễn Xuân Hội – 15520268

Nguyễn Xuân Phóng - 1552

**Giảng viên hướng dẫn:**  
ThS. Đỗ Thị Minh Phụng

**ĐỒ ÁN MÔN   
HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**

ĐỀ TÀI: ỨNG DỤNG QUẢN LÝ KHO CUNG CẤP VẬT LIỆU XÂY DỰNG

**TP. Hồ Chí Minh, Tháng 06 năm 2020**

**LỜI CẢM ƠN**

Đầu tiên, em xin gửi lời cảm ơn chân thành đến quý Thầy cô giảng viên Trường Đại học Công nghệ thông tin – Đại học Quốc gia TP.HCM và quý thầy cô khoa Hệ thống Thông tin đã giúp cho em có những kiến thức cơ bản làm nền tảng để thực hiện đồ án này.

Đặc biệt, em xin gửi lời cảm ơn và lòng biết ơn sâu sắc nhất tới cô – ThS. Đỗ Thị Minh Phụng, người đã hướng dẫn cho em trong suốt thời gian làm đồ án. Cô đã trực tiếp hướng dẫn tận tình, sửa chữa và đóng góp nhiều ý kiến quý báu giúp nhóm chúng em hoàn thành tốt báo cáo môn học của mình. Một lần nữa em chân thành cảm ơn cô và chúc cô dồi dào sức khoẻ.

Trong thời gian một học kỳ thực hiện đồ án, chúng em đã vận dụng những kiến thức nền tảng đã tích lũy đồng thời kết hợp với việc học hỏi và nghiên cứu những kiến thức mới từ thầy cô, bạn bè cũng như nhiều nguồn tài liệu tham khảo. Từ đó, nhóm chúng em đã vận dụng tối đa những gì đã thu thập được để hoàn thành một báo cáo đồ án tốt nhất. Tuy nhiên, vì kiến thức chuyên môn còn hạn chế và bản thân còn thiếu nhiều kinh nghiệm thực tiễn nên nội dung của báo cáo không tránh khỏi những thiếu sót, em rất mong nhận được sự góp ý, chỉ bảo thêm của quý thầy cô nhằm hoàn thiện những kiến thức của mình để em có thể dùng làm hành trang thực hiện tiếp các đề tài khác trong tương lai cũng như là trong việc học tập và làm việc sau này.

Một lần nữa xin gửi đến thầy cô, bạn bè lời cảm ơn chân thành và tốt đẹp nhất.

Thành phố Hồ Chí Minh, tháng 06 năm 2020

Nhóm sinh viên thực hiện

Nguyễn Ngọc Công

Nguyễn Xuân Hội

**NHẬN XÉT CỦA GIẢNG VIÊN**

**……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………**

# TỔNG QUAN ĐỀ TÀI

## Đặt vấn đề

Ngày nay Công nghệ thông tin đã phát triển với tốc độ nhanh chóng. CNTT đã ứng dụng hầu như vào mọi hoạt động của con người từ nguyên cứu khoa học, phát triển kinh tế, quân sự, nghệ thuật, kinh doanh… Cùng với sự phát triển không ngừng về kỹ thuật máy tính và mạng điện tử, công nghệ thông tin cũng được những công nghệ có đẳng cấp cao và lần lượt chinh phục hết đỉnh cao này đến đỉnh cao khác. Mạng Internet là một trong những sản phẩm có giá trị hết sức lớn lao và ngày càng trở nên một công cụ không thể thiếu, là nền tảng chính cho sự truyền tải, trao đổi thông tin trên toàn cầu.

Giờ đây, mọi việc liên quan đến thông tin trở nên thật dễ dàng cho người sử dụng: chỉ cần có một máy tính kết nối Internet và một dòng dữ liệu truy tìm thì gần như lập tức… cả thế giới về vấn đề mà bạn đang quan tâm sẽ hiện ra, có đầy đủ thông tin, hình ảnh và thậm chí đôi lúc có cả những âm thanh nếu bạn cần…

Bằng Internet, chúng ta đã thực hiện được nhiều công việc với tốc độ nhanh hơn và chi phí thấp hơn nhiều so với cách thức truyền thống. Chính điều này, đã thúc đẩy sự khai sinh và phát triển của thương mại điện tử và chính phủ điện tử trên khắp thế giới, làm biến đổi đáng kể bộ mặt văn hóa, nâng cao chất lượng cuộc sống con người.

Trong hoạt động sản xuất, kinh doanh, giờ đây, việc công nghệ thông tin hóa quy trình kinh doanh đã khẳng định được vai trò xúc tiến và thúc đẩy sự phát triển của doanh nghiệp. Quản lý kho là một công việc rất vất vả cho người chủ shop và nhân viên bán hàng. Khi cửa hàng, doanh nghiệp ngày càng phát triển thì số lượng hàng hóa trong kho cũng ngày càng tăng cao. Quản lý kho thế nào để hàng hóa bán ra – nhập vào không bị nhầm lẫn, thất thoát là bài toán nan giải được các doanh nghiệp đặt ra. Nhận thấy việc quản lý kho bãi thủ công bằng giấy tờ hiện tại gặp nhiều bất cập và sai sót, gây ảnh hưởng lớn đến doanh nghiệp. Nhóm em quyết định lên ý tưởng thực hiện đề tài Quản trị cơ sở dữ liệu Quản lý kho cho doanh nghiệp. Với đề tài này, ứng dụng Quản lý kho có thể áp dụng cho những doanh nghiệp có hình thức kinh doanh có kho hàng. Đối với phạm vi đề tài, nhóm chúng em chọn triển khai ví dụ cho doanh nghiệp Kinh doanh Vật liệu Xây Dựng.

## Mục tiêu

Dùng những kiến thức đã học ở trường, đặc biệt là kiến thức môn Hệ quản trị cơ sở dữ liệu để xây dựng một ứng dụng Quản lý kho. Ứng dụng được xây dựng đáp ứng đầy đủ các nhu cầu cơ bản cho nghiệp vụ quản lý kho hàng.

## Công cụ sử dụng

Trong quá trình thực hiện, nhóm sử dụng một số công cụ sau:

* SQL Developer
* Oracle SQL Developer Data Modeler
* Draw.io
* IntelliJ IDEA
* Github

# PHÂN TÍCH VÀ THIẾT KẾ CƠ SỞ DỮ LIỆU

## Phân tích nghiệp vụ:

Lĩnh vực kinh doanh: Kinh doanh vật liệu xây dựng

### Những nghiệp vụ chung

* Nghiệp vụ quản lý sản phẩm: Mục đích quản lý các sản phẩm, nhà sản cung cấp, giá cả, doanh số bán hàng của từng loại sản phẩm.
* Nghiệp vụ quản lý tồn kho: Mục đích quản lý tồn kho sản phẩm, xem xét số lượng tồn, để lên phương án giảm giá thúc đẩy bán sản phẩm tồn hoặc liên hệ nhà cung cấp để mua hàng mới.
* Nghiệp vụ nhập hàng: Mục đích quản lý nhập hàng, giá cả, nhà cung cấp, quản lý các chừng từ liên quan.
* Nghiệp vụ xuất hàng: Mục đích quản lý hàng hóa xuất đi, khách hàng, quản lý các chứng từ liên quan.
* Nghiệp vụ báo cáo, thống kê: Mục đích báo cáo/ thống kê tổng quát hiệu suất công việc bản hàng, kho bãi, lượng hàng tồn, để người quản lý có các chiến lược phương án báng hàng, nhập hàng phù hợp.

### Nghiệp vụ riêng của người quản lý

* Nghiệp vụ quản lý nhân viên: Mục đích quản lý danh sách nhân viên đang làm việc, tính lương, tính hiệu xuất cũng như đánh giá nhân viên.

## Phân tích yêu cầu

### Yêu cầu chức năng

Chức năng phải đáp ứng tốt những nhu cầu cơ bản của một doanh nghiệp kinh doanh có kho hàng. Người dùng bao gồm: Người quản lý và nhân viên, trong đó người quản lý có toàn quyền với mọi chức năng của ứng dụng, nhân viên bị hạn chế một số chức năng nhất định, từ đây đến cuối đồ án, đề cập đến tác nhân người dùng có nghĩa là bao gồm cả nhân viên và người quản lý.

* **Chức năng hệ thống**

1. Chức năng đăng nhập / đăng xuất: Người dùng sau khi đăng nhập, dựa vào cơ sở dữ liệu hệ thống sẽ xác định quyền của người dùng là quyền nhân viên hay quản lý. Từ đó danh sách menu trong bảng điều khiển sẽ hiển thị những menu thích hợp.

* **Những chức năng chung**

1. Chức năng quản lý sản phẩm: Chức năng này quản lý thông tin về sản phẩm. Người dùng có thể xem danh sách sản phẩm, thực hiện các chức năng thêm, sửa, xóa, tìm kiếm sản phẩm.
2. Chức năng quản lý loại sản phẩm: Chức năng này quản lý thông tin về loại sản phẩm. Người dùng có thể xem danh sách loại sản phẩm, thực hiện các chức năng thêm, sửa, xóa, tìm kiếm loại sản phẩm.
3. Chức năng quản lý hàng tồn kho: Chức năng này quản lý số lượng hàng đang tồn trong kho. Người dùng có thể xem chi tiết số lượng còn tồn trong kho với mỗi mặt hàng.
4. Chức năng quản lý nhập hàng: Chức năng này quản lý quá trình nhập hàng của người dùng. Ở chức năng này, hóa đơn ghi lại thông tin, chi tiết nhập hàng.
5. Chức năng xuất hàng: Chức năng này quản lý quá trình xuất hàng của người dùng. Ở chức năng này, hóa đơn ghi lại thông tin, chi tiết xuất hàng.
6. Chức năng báo cáo / thống kê: Chức năng hỗ trợ người dùng có thể xuất báo cáo, thống kê doanh thu về các mặt hàng riêng biệt, cũng như doanh số xuất / nhập theo thời gian.

* **Những chức năng riêng cho người quản lý**

1. Chức năng quản lý nhân viên: Chức năng này quản lý thông tin về nhân viên. Người quản có thể xem danh sách loại sản phẩm, thực hiện các chức năng thêm, sửa, xóa, tìm kiếm nhân viên.
2. Chức năng quản lý phân quyền: Chức năng này giúp người quản lý có thể phân quyền quân viên của họ. Cho phép cấp và thu hồi quyền của mỗi nhân viên, từ đó nâng cao hiệu suất, gia tăng tính bảo mật và hiệu quả của hệ thống.

### Yêu cầu phi chức năng

**Yêu cầu về giao diện:** giao diện ứng dụng gần gũi, thân thiện. Người dùng có thể sử dụng dễ dàng, thuận tiện và có tính tương tác cao. Thao tác trên ứng dụng nhanh gọn và dễ sử dụng.

**Yêu cầu về chất lượng:**

Tính tiến hóa: Dễ dàng trong việc nâng cấp hay thêm các modules tiện ích khác.

**Tính tiện dụng:**

* Ứng dụng có giao diện trực quan, thân thiện và dễ sử dụng.
* Các đặc tả và hướng dẫn sử dụng rõ ràng.
* Thao tác đơn giản và nhanh gọn, đáp ứng nhu cầu của học sinh.

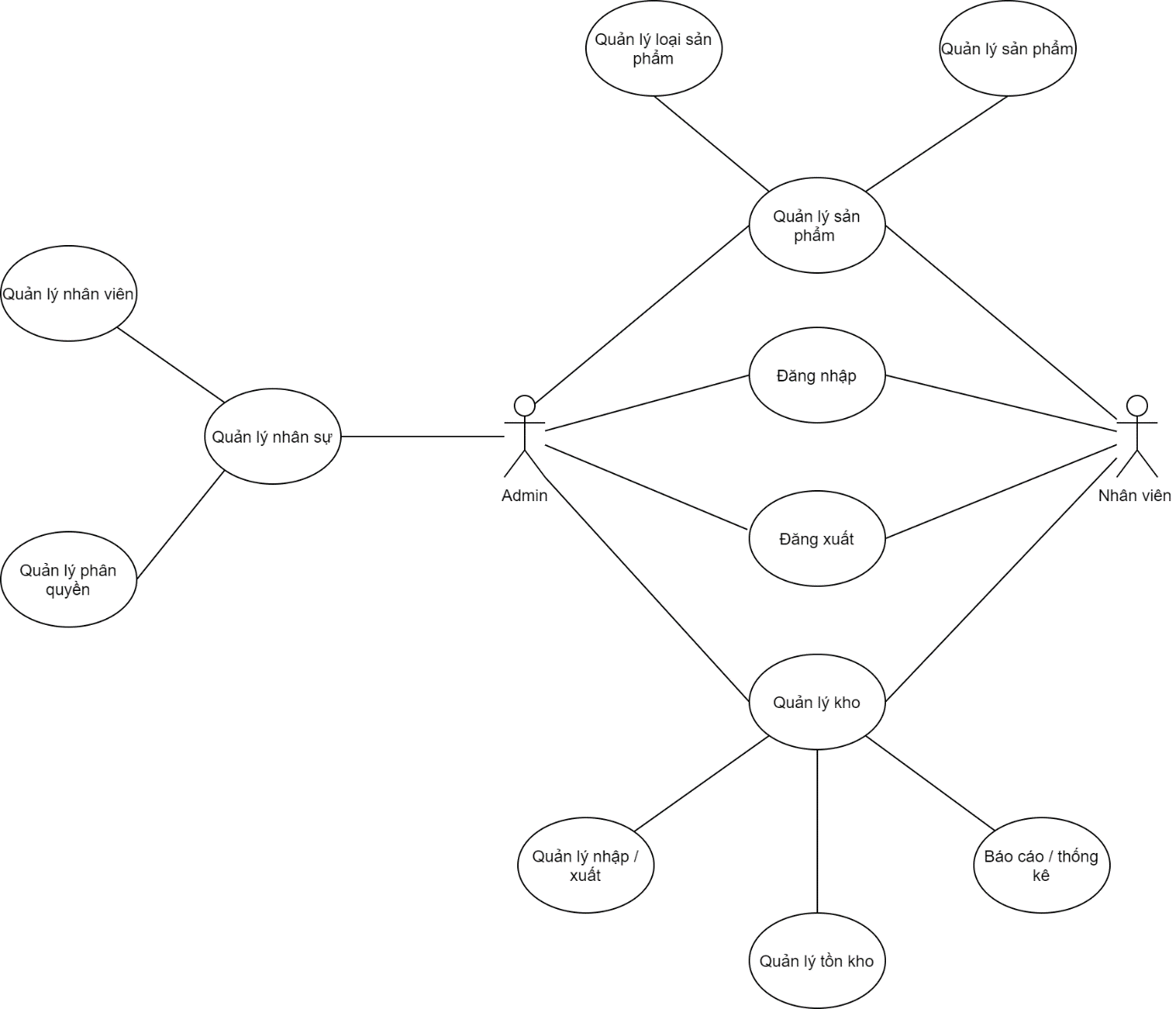
**Tính tương thích**: Hoạt động tốt trên mọi nền tảng, trên ứng dụng, mobile, tablet.

**Tính hiệu quả:**

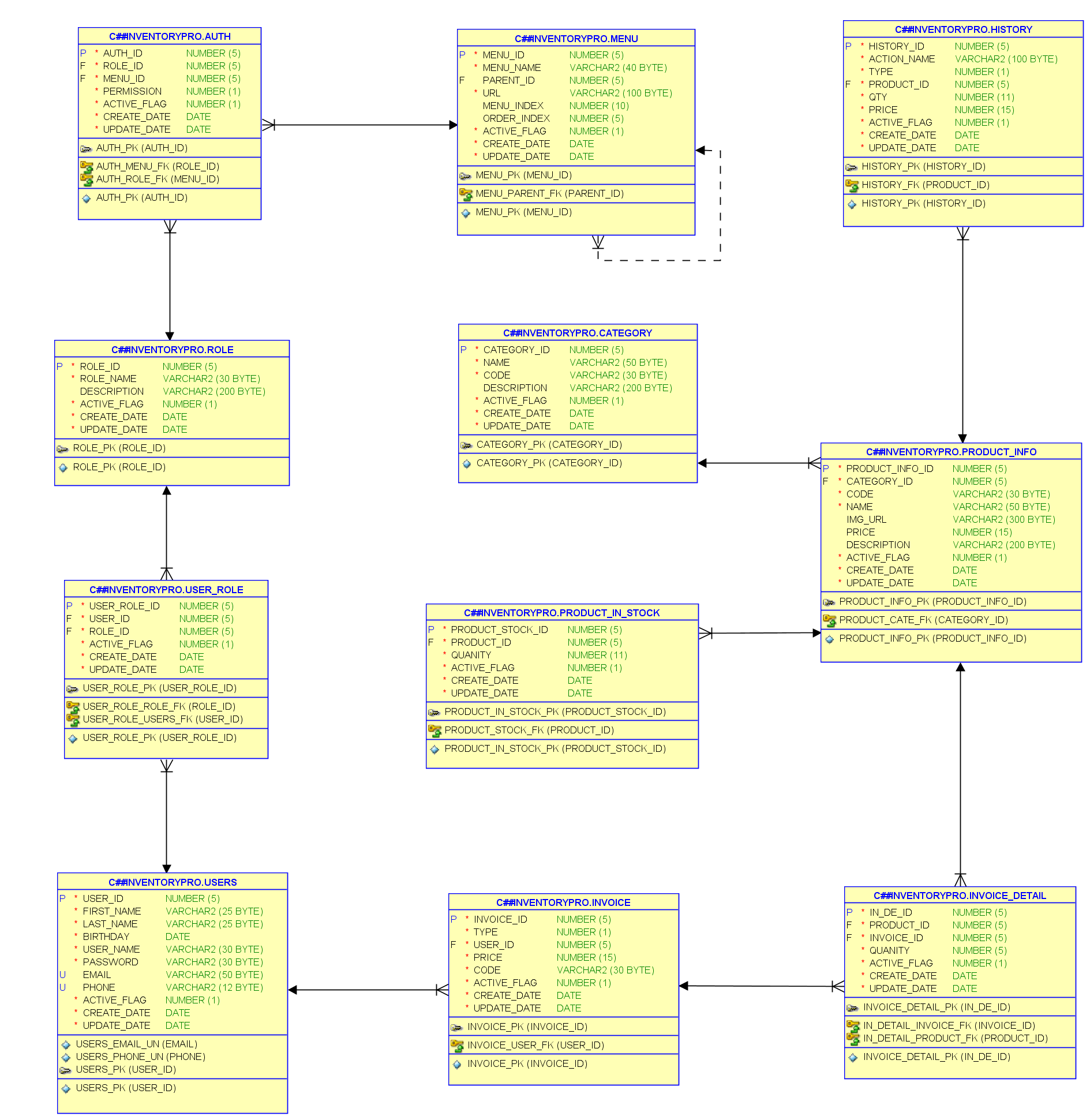
* Ứng dụng hoạt động tốt và đầy đủ những yêu cầu chức năng cơ bản của nghiệp vụ quản lý kho.
* Trang ứng dụng hoạt động ổn định, và đáng tin cậy, có thể thể truy cập và xử lý đồng thời nhiều hành động

### Thiết kế mô hình quan hệ

1. Sơ đồ Use-case



1. Mô hình dữ liệu quan hệ



1. Mô tả thành phần dữ liệu

Bảng AUTH:

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Diễn giải** |
| 1 | AUTH\_ID | NUMBER(5,0) |  |
| 2 | ROLE\_ID | NUMBER(5,0) |  |
| 3 | MENU\_ID | NUMBER(5,0) |  |
| 4 | PERMISSION | NUMBER(1,0) |  |
| 5 | ACTIVE\_FLAG | NUMBER(1,0) |  |
| 6 | CREATE\_DATE | DATE |  |
| 7 | UPDATE\_DATE | DATE |  |

Bảng CATEGORY

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Diễn giải** |
| 1 | CATEGORY\_ID | NUMBER(5,0) |  |
| 2 | NAME | VARCHAR2(50 BYTE) |  |
| 3 | CODE | VARCHAR2(30 BYTE) |  |
| 4 | DESCRIPTION | VARCHAR2(200 BYTE) |  |
| 5 | ACTIVE\_FLAG | NUMBER(1,0) |  |
| 6 | CREATE\_DATE | DATE |  |
| 7 | UPDATE\_DATE | DATE |  |

Bảng HISTORY

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Diễn giải** |
| 1 | HISTORY\_ID | NUMBER(5,0) |  |
| 2 | ACTION\_NAME | VARCHAR2(50 BYTE) |  |
| 3 | TYPE | NUMBER(1,0) |  |
| 4 | PRODUCT\_ID | NUMBER(5,0) |  |
| 5 | QTY | NUMBER(11,0) |  |
| 6 | PRICE | NUMBER(15,0) |  |
| 7 | ACTIVE\_FLAG | NUMBER(1,0) |  |
| 8 | CREATE\_DATE | DATE |  |
| 9 | UPDATE\_DATE | DATE |  |

Bảng INVOICE

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Diễn giải** |
| 1 | INVOICE\_ID | NUMBER(5,0) |  |
| 2 | TYPE | NUMBER(1,0) |  |
| 3 | USER\_ID | NUMBER(5,0) |  |
| 4 | PRICE | NUMBER(15,0) |  |
| 5 | CODE | VARCHAR2(30 BYTE) |  |
| 6 | ACTIVE\_FLAG | NUMBER(1,0) |  |
| 7 | CREATE\_DATE | DATE |  |
| 8 | UPDATE\_DATE | DATE |  |

Bảng INVOICE\_DETAIL

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Diễn giải** |
| 1 | IN\_DE\_ID | NUMBER(5,0) |  |
| 2 | PRODUCT\_ID | NUMBER(5,0) |  |
| 3 | INVOICE\_ID | NUMBER(5,0) |  |
| 4 | QUANITY | NUMBER(5,0) |  |
| 5 | ACTIVE\_FLAG | NUMBER(1,0) |  |
| 6 | CREATE\_DATE | DATE |  |
| 7 | UPDATE\_DATE | DATE |  |

Bảng MENU

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Diễn giải** |
| 1 | MENU\_ID | NUMBER(5,0) | Mã số duy nhất của menu |
| 2 | MENU\_NAME | VARCHAR2(40 BYTE) | Tên menu |
| 3 | PARENT\_ID | NUMBER(5,0) | Menu cha |
| 4 | URL | VARCHAR2(100 BYTE) | Link menu |
| 5 | ORDER\_INDEX | NUMBER(5,0) | Thứ tự sắp xếp menu |
| 6 | ACTIVE\_FLAG | NUMBER(1,0) | Trạng thái hoạt động: 0 – inactive, 1 – active |
| 7 | CREATE\_DATE | DATE | Ngày tạo |
| 8 | UPDATE\_DATE | DATE | Ngày cập nhật |

Bảng PRODUCT\_INFO

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Diễn giải** |
| 1 | PRODUCT\_INFO\_ID | NUMBER(5,0) | Mã số duy nhất của sản phẩm |
| 2 | CATEGORY\_ID | NUMBER(5,0) | Mã số thể loại |
| 3 | CODE | VARCHAR2(30 BYTE) | Mã định danh sản phẩm |
| 4 | NAME | VARCHAR2(50 BYTE) | Tên sản phẩm |
| 5 | IMG\_URL | VARCHAR2(300 BYTE) | Link hình ảnh sản phẩm |
| 6 | PRICE | NUMBER(15,0) | Giá sản phẩm |
| 7 | DESCRIPTION | VARCHAR2(200 BYTE) | Mô tả |
| 8 | ACTIVE\_FLAG | NUMBER(1,0) | Trạng thái hoạt động: 0 – inactive, 1 – active |
| 9 | CREATE\_DATE | DATE | Ngày tạo |
| 10 | UPDATE\_DATE | DATE | Ngày cập nhật |

Bảng PRODUCT\_IN\_STOCK

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Diễn giải** |
| 1 | PRODUCT\_STOCK\_ID | NUMBER(5,0) | Mã số duy nhất của bảng |
| 2 | PRODUCT\_ID | NUMBER(5,0) | Mã số sản phẩm |
| 3 | QUANITY | NUMBER(11,0) | Số lượng |
| 4 | ACTIVE\_FLAG | NUMBER(1,0) | Trạng thái hoạt động: 0 – inactive, 1 – active |
| 5 | CREATE\_DATE | DATE | Ngày tạo |
| 6 | UPDATE\_DATE | DATE | Ngày cập nhật |

Bảng ROLE

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Diễn giải** |
| 1 | ROLE\_ID | NUMBER(5,0) | Mã số duy nhất của phân quyền |
| 2 | ROLE\_NAME | VARCHAR2(30 BYTE) | Tên quyền |
| 3 | DESCRIPTION | VARCHAR2(200 BYTE) | Mô tả |
| 4 | ACTIVE\_FLAG | NUMBER(1,0) | Trạng thái hoạt động: 0 – inactive, 1 – active |
| 5 | CREATE\_DATE | DATE | Ngày tạo |
| 6 | UPDATE\_DATE | DATE | Ngày cập nhật |

Bảng USERS

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Diễn giải** |
| 1 | USER\_ID | NUMBER(5,0) | Mã số duy nhất của nhân viên |
| 2 | FIRST\_NAME | VARCHAR2(25 BYTE) | Tên người dùng |
| 3 | LAST\_NAME | VARCHAR2(25 BYTE) | Họ người dùng |
| 4 | BIRTHDAY | DATE | Ngày sinh |
| 5 | USER\_NAME | VARCHAR2(30 BYTE) | Tên đăng nhập |
| 6 | PASSWORD | VARCHAR2(30 BYTE) | Mật khẩu |
| 7 | EMAIL | VARCHAR2(50 BYTE) | Địa chỉ email |
| 8 | PHONE | VARCHAR2(12 BYTE) | Điện thoại |
| 9 | ACTIVE\_FLAG | NUMBER(1,0) | Trạng thái hoạt động: 0 – inactive, 1 – active |
| 10 | CREATE\_DATE | DATE | Ngày tạo |
| 11 | UPDATE\_DATE | DATE | Ngày cập nhật |

Bảng USER\_ROLE

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Diễn giải** |
| 1 | USER\_ROLE\_ID | NUMBER(5,0) | Mã số duy nhất của bảng |
| 2 | USER\_ID | NUMBER(5,0) | Mã người dùng |
| 3 | ROLE\_ID | NUMBER(5,0) | Mã phân quyền |
| 4 | ACTIVE\_FLAG | NUMBER(1,0) | Trạng thái hoạt động: 0 – inactive, 1 – active |
| 5 | CREATE\_DATE | DATE | Ngày tạo |
| 6 | UPDATE\_DATE | DATE | Ngày cập nhật |

# CÔNG NGHỆ SỬ DỤNG

## Oracle

* + 1. Giới thiệu Oracle

Oracle Database hay còn gọi là Oracle RDBMS hoặc đơn giản là Oracle (do đây có lẽ là sản phẩm nổi tiếng nhất của hãng), là 1 hệ quản trị cơ sở dữ liệu quan hệ, được phát triển và phân phối bởi tập đoàn Oracle.

Là hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mang tính mềm dẻo, linh động, thích ứng cao với các quy mô xử lý giao dịch, an toàn hệ thống. Cung cấp các công cụ xây dựng và quản lý cơ sở dữ liệu.

Giống như các phần mềm RDBMS khác, Oracle Database được xây dựng dựa trên ngôn ngữ SQL. Phần mềm Oracle được bổ sung PL/SQL, được Oracle phát triển nhằm bổ sung một số extension độc quyền cho SQL chuẩn – khá phổ biến trong các nhà cung cấp RDBMS.

Cơ sở dữ liệu Oracle cũng hỗ trợ lập trình bằng Java, tích hợp web (kết nối ứng dụng với công nghệ Web được tích hợp trong Oracle WebServer)

* + 1. Các phiên bản

- Phiên bản 1 (năm 1977), Phiên bản 2 (năm 1979)

- Phiên bản 3 (năm 1983), Phiên bản 4 (1984)

- Phiên bản 5 phát hành năm 1985 (SQLNet: hệ thống khách/chủ (client/server)).

- Phiên bản 6 phát hành năm 1988 (Sequence, thao tác ghi trễ).

- Oracle7 được phát hành năm 1992 (SQL\*DBA).

- Năm 1999 Oracle giới thiệu Oracle8i (i:internet).

- Năm 2001-2002: 2 phiên bản Oracle9i (Release 1&2).

- Năm 2004-2005: 2 phiên bản Oracle10g (g:Grid) (Release 1&2).

- Năm 2008 – 2009: Phiên bản 11g

- Năm 2013: Phiên bản 12c (c: cloud)

- Năm 2018: Phiên bản 18c, Cơ sở Dữ liệu Tự động của Oracle (Oracle Autonomous Databas, cloud database có khả năng tự quản lý, tự bảo vệ và tự sửa chữa nhờ công nghệ học máy (machine learning) giúp cung cấp hiệu suất cao, khả năng bảo mật mà không cần sự can thiệp của con người, với chi phí thấp hơn một nửa so với Amazon Web Services.

* + 1. Oracle 12c

Để theo kịp xu hướng Cloud Computing, PaaS (Platform as a Service), từ phiên bản Oracle 12c thì Oracle đã giới thiệu tính năng mới là Pluggable Database. Tính năng này cho phép 1 Database độc lập có thể được di chuyển dễ dàng qua các hệ thống khác nhau hoặc qua các nền tảng khác nhau hoặc giữa các bản phân phối (release) Database khác nhau.

Multitenant là 1 khái niệm mới được sử dụng trong phiên bản Oracle 12c và là 1 bước thay đổi lớn trong lịch sử kiến trúc của Oracle. Multitenant bao gồm 2 khái niệm nhỏ là Container Database (CDB) and Pluggable Database (PDB).

Khái niệm database trong 11g tương ứng với khái niệm Container Database (CDB) trong 12c. Cụ thể sau khi cài đặt phần mềmOracle 12c, có thể tạo 1 hoặc nhiều Container Database (CDB). (Thực tế là chỉ cần 1).

Trong Oracle 12C có một khái niệm mới là CDB$ROOT (Hoặc gọi là CDB Root) là một đối tượng nằm trong CDB. Các SCHEMA có thể gắn vào trên CDB$ROOT. Hoặc có các Pluggable Database (PDB) thông thường khác có gắn vào CDB$ROOT. Mỗi Pluggable Database chứa 0 hoặc nhiều SCHEMA.

PDB$SEED là một Pluggable database mẫu (Template) nó được sử dụng để làm mẫu (mặc định) để tạo ra một Pluggable Database mới. Tất nhiên có thể lấy một Pluggable Database bất kỳ nào đó làm mẫu để tạo ra một Pluggable Database mới.

Mối quan hệ giữa CDB và PDB là mối quan hệ 1-nhiều. Trong phiên bản Oracle 12c R1 thì 1 CDB có thể chứa tối đa là 250 PDB.

## Hibernate

* + 1. ORM framework

ORM (Object Relational Mapping) framework là một cơ chế cho phép người lập trình thao tác với database một cách hoàn toàn tự nhiên thông qua các đối tượng. Lập trình viên hoàn toàn không quan tâm đến loại database sử dụng SQL Server, Oracle, MySQL, PostgreSQL, …

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.

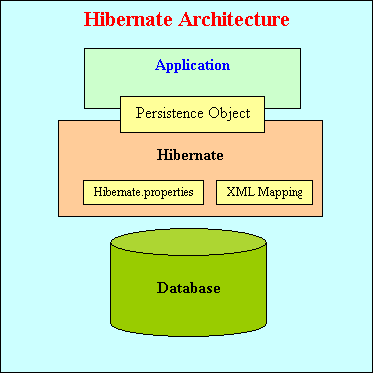
* + 1. Hibernate framework

Hibernate là một trong những ORM Framework. Hibernate framework là một framework cho persistence layer. Như vậy, nhờ có Hibernate framework mà giờ đây khi bạn phát triển ứng dụng bạn chỉ còn chú tâm vào những layer khác mà không phải bận tâm nhiều về persistence layer nữa.

Hibernate giúp lập trình viên viết ứng dụng Java có thể map các object (POJO) với hệ quản trị cơ sở dữ liệu quan hệ (database), và hỗ trợ thực hiện các khái niệm lập trình hướng đối tượng với cơ dữ liệu quan hệ.

Hibernate giúp lưu trữ và truy vấn dữ liệu quan hệ mạnh mẽ và nhanh. Hibernate cho phép bạn truy vẫn dữ liệu thông qua Java Persistence API (JPA) hoặc bằng ngôn ngữ SQL mở rộng của Hibernate (HQL) hoặc bằng SQL thuần (Native SQL).

Kiến trúc Hibernate bao gồm nhiều đối tượng như đối tượng persistent D, session factory, transaction factory, connection factory, session, transaction, …



## JPA

JPA là viết tắt của Java Persistence API, nó là một đặc tả Java cho việc ánh xạ giữa các đối tượng Java với cơ sở dữ liệu quan hệ sử dụng công nghệ phổ biến là ORM (Object Relational Mapping).

JPA cung cấp đầy đủ các công cụ cho phép chúng ta có thể thao tác với cơ sở dữ liệu một cách đơn giản và nhanh chóng. JPA có thể dùng để persist một đối tượng Java (POJO – Plain Old Java Object) vào trong cơ sở dữ liệu hoặc lấy dữ liệu từ cơ sở dữ liệu và ánh xạ (mapping) ra các đối tượng Java một cách đơn giản.

JPA hoạt động như một cầu nối giữa các table/ các mối quan hệ giữa các table trong database và các class/ mối quan hệ giữa các object. Ví dụ: table USER với các column (Id, username, password) sẽ tương ứng với class User.java với các field Id, username, password. Từ đó mỗi khi truy vấn table hay các column ta sẽ gọi trực tiếp các phương thức trên các class, các field của class mà không cần quan tâm tới việc đang dùng loại database nào, kiểu dữ liệu database ra sao, …

## Java Spring

Spring là framework phát triển ứng dụng phổ biến nhất dành cho Java Enterprise. Ban đầu nó được viết bởi Rod Johnson và lần đầu tiên được phát hành theo giấy phép Apache 2.0 vào tháng 6 năm 2003. Spring có kích thướng nhẹ, phiên bản cơ bản của Spring framework có kích thước khoảng 2MB.

Spring framework là một Java Platform mã nguồn mở, một giải pháp gọn nhẹ dành cho Java Enterprise. Với Spring Framework các nhà phát triển có thể tạo ra các mã có hiệu suất cao, dễ kiểm thử và có thể sử dụng lại được.

Các tính năng core của Spring Framework có thể được sử dụng trong việc phát triển bất kỳ ứng dụng Java nào. Bên cạnh đó, phần mở rộng được sử dụng để xây dựng các ứng dụng web trên nền tảng Java EE. Mục tiêu của Spring Framework là làm cho việc phát triển ứng dụng J2EE dễ dàng hơn và thúc đẩy việc lập trình tốt hơn bằng mô hình POJO-based.



## Github

* + 1. Git

Git là tên gọi là một hệ thống quản lý phiên bản phân tán phổ biến nhất hiện nay (Distributed Version Control System – DVCS). Git sẽ giúp người dùng lưu lại các phiên bản của những lần thay đổi vào mã nguồn và sẽ dễ dàng cho việc khôi phục lại mà không cần phải thủ công copy rồi paste vào đâu đó, phiên bản đó đã được sao lưu. Khi chúng ta phát hiện ra lỗi ở đâu đó và muốn backup lại phiên làm việc trước khi bị lỗi xảy ra thì sẽ thật đơn giản khi chúng ta sử dụng Git. Một điểm đặc biệt nữa là một thành viên trong cùng một team khi làm việc với nhau hoàn toàn có thể theo dõi online được các thay đổi của các thành viên khác ở từng phiên bản làm việc mà không nhất thiết phải ngồi ngay cạnh nhau, họ cũng có thể đối chiếu được những thay đổi đó để rồi gộp phiên bản của thành viên khác vào phiên bản của họ. Cuối cùng là tất cả có thể đưa các thay đổi vào mã nguồn của mình lên một kho chứa mã nguồn.

* + 1. Github

Github được cho ra đời với mục đích cung cấp dịch vụ Git server miễn phí. Ngoài ra trang này cũng cung cấp các giao diện để người dùng có thể dễ dàng theo dõi các sự thay đổi trên trình duyệt mà không cần phải cài phần mềm Git trên máy tính.

## IntelliJ IDEA

IntelliJ IDEA là một IDE Java thông minh cung cấp một sự kết hợp mạnh mẽ của các công cụ phát triển phần mềm. IntelliJ IDEA là công cụ nhằm tạo ra những dự án lập trình cho điện thoại hoặc cho khả năng mã hóa sâu sắc và điều hướng nhanh, phần mềm này còn cung cấp cho người dùng một danh sách các biểu tượng và ký hiệu phục vụ trong công việc lập trình của người dùng. Chức năng của IntelliJ IDEA được tiếp tục mở rộng bởi người dùng và bên thứ ba thông qua các plugin. IntelliJ IDEA cung cấp hỗ trợ cho Java EE, Spring / Hibernate và các ngăn xếp công nghệ khác.

# XÂY DỰNG VÀ QUẢN LÝ GIAO TÁC

# XỬ LÝ TRUY XUẤT ĐỒNG THỜI

# THIẾT KẾ GIAO DIỆN

# KẾT LUẬN

# Phụ lục

## Phụ lục 1: Bảng phân chia công việc

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Đầu việc** | **Tỷ lệ** | |
| **Công** | **Hội** |
| 1 | Phân tích bài toán, xác định chức năng |  |  |
| 2 | Thiết kế & xây dựng CSDL |  |  |
| 3 | Thiết kế giao diện |  |  |
| 4 | Triển khai ứng dụng bằng Java Spring |  |  |
| 5 | Xử lý vấn đề truy xuất đồng thời |  |  |
| 6 | Kiểm tra chức năng |  |  |
| 7 | Soạn và chỉnh sửa báo cáo |  |  |

## Phụ lục 2: Tài liệu tham khảo

[1] Oracle Database Database Concepts, 12c Release 2 (12.2) - Primary Authors: Lance Ashdown, Tom Kyte

[2] Slide môn Hệ quan trị cơ sở dữ liệu – ThS. Đỗ Thị Minh Phụng, trường ĐH Công Nghệ Thông Tin

[3] Quản lý kho là gì?: <https://www.sapo.vn/blog/kinh-nghiem-quan-ly-kho-dat-hieu-qua/>

[4] Quy trình quản lý kho: <https://sec-warehouse.vn/quy-trinh-quan-ly-kho-hang.html>

[5] Kiến thức Oracle : <https://csc.edu.vn/lap-trinh-va-csdl/tin-tuc/kien-thuc-lap-trinh/hoc-oracle--kien-thuc-co-ban-cho-nguoi-moi-bat-dau-802>

[6] Hibernate framework: <https://hibernate.org/>

[7] Hibernate tutorial: <https://www.tutorialspoint.com/hibernate/hibernate_architecture.htm>

[8] Tổng quan Spring framework: <https://viblo.asia/p/tong-quan-ve-spring-framework-YWOZryEyKQ0>

[9] Spring framework: <https://spring.io/>

[10]Github: <https://github.blog/>

[11]IntelliJ IDEA: <https://www.jetbrains.com/idea/>

[12]

[13]

# 