

ĐỒ ÁN CHUYÊN NGÀNH

XÂY DỰNG HỆ THỐNG QUẢN LÝ CƠ SỞ VẬT CHẤT THÔNG QUA QR CODE

Ngành: **Công Nghệ Thông Tin**

Chuyên ngành: **Công Nghệ Phần Mềm**

Giảng viên hướng dẫn : Hoàng Văn Hiếu

Sinh viên thực hiện : Trần Phương Nam

MSSV: 1711060218 Lớp: 17DTHB4

TP. Hồ Chí Minh, 2021

ĐỒ ÁN CHUYÊN NGÀNH

XÂY DỰNG HỆ THỐNG QUẢN LÝ CƠ SỞ VẬT CHẤT THÔNG QUA QR CODE

Ngành: Công Nghệ Thông Tin

Chuyên ngành: Công Nghệ Phần Mềm

Giảng viên hướng dẫn : Hoàng Văn Hiếu

Sinh viên thực hiện : Trần Phương Nam

MSSV: 1711060218 Lớp: 17DTHB4

TP. Hồ Chí Minh, 2020

THÔNG TIN VỀ ĐỀ TÀI

1. Mô tả đề tài:

a. Ý tưởng:

- Tạo ra một hệ thống để xử lý các công việc liên quan đến quản lý cơ sở vật chất. Giúp tối ưu hoá trong các công việc như lưu trữ, chỉnh sửa thông tin tài sản, lưu trữ các phiếu mượn, phiếu trả để thống kê, ...

b. Chức năng:

- Phân quyền trong hệ thống.
- Thêm xoá sửa các thông tin của tài sản.
- Tạo các phiếu mượn phiếu trả.
- Thêm tài sản bằng Excel.

2. Lý do thực hiện đề tài:

- Hiện nay hầu hết các phòng, ban của các trường đều đã ứng dụng các phần mềm để quản lý các quy trình mượn trả các thiết bị như là Excel, ... Tuy nhiên các dữ liệu sau khi tổng hợp lại thì đa số là riêng lẻ ví dụ như sheet mượn trả sẽ nằm 1 file còn sheet quản lý sẽ nằm ở 1 file khác, dẫn tới việc khó đồng bộ trong công tác quản lý mượn trả cũng như xử lý, sửa chữa các thiết bị cơ sở vật chất nhà trường.
- Hiện nay ở Việt Nam ta có rất nhiều hệ thống quản lý trường học hay quản lý tài sản đã được triển khai và hệ thống nào cũng vận hành rất tốt nhưng chưa đáp ứng hết được nhu cầu quản lý của nhà trường và chi phí bỏ ra để thực hiện nó là không hề nhỏ không phù hợp với khả năng tài chính của nhà trường. Chính vì lý do trên nên em chọn đề tài: “Xây dựng hệ thống quản lý cơ sở vật chất thông qua QR CODE”. Nhằm giải quyết những khó khăn về vấn đề quản lý cơ sở vật chất nhà trường.

3. Mục tiêu đề tài:

- Mục tiêu của đề tài này là ứng dụng dịch vụ web , app để xây dựng hệ thống quản lý cơ sở vật chất của nhà trường. Hệ thống này cho phép quản lý toàn bộ cơ sở vật chất nhà trường như bàn, ghế, các thiết bị âm thanh, các thiết bị trình chiếu thông qua 1 mã QR CODE tự sinh khi nhập vào hệ thống. Thay vì chọn hệ thống mạng cục bộ thì em chọn làm trên web với mục đích cho phép giảng viên, sinh viên có thể đăng ký

mượn thông qua mạng internet. Hệ thống có thể quản lý được các thiết bị cơ sở vật chất nhà trường, thống kê tính hình mượn trả cũng như tình trạng thiết bị của nhà trường.

4. Nội dung thực hiện:

Gồm 5 phần: Phân tích yêu cầu, thiết kế Cơ sở dữ liệu, thiết kế giao diện, kiểm thử, vận hành và sử dụng.

5. Phương pháp thực hiện:

- Phần 1: Phân tích từng nhu cầu một cách chi tiết nhất, tham khảo các hệ thống có chức năng tương tự với các chức năng của hệ thống đang thực hiện.
- Phần 2: Thiết kế cơ sở dữ liệu một cách chi tiết. Tham khảo thêm phân tích hệ thống thông tin có liên quan đến việc quản lý cơ sở vật chất để hoàn thiện tốt hơn trong vấn đề phân tích và thiết kế cơ sở dữ liệu. Hoàn thành bản vẽ cơ sở dữ liệu.
- Phần 3: Dùng phần mềm Visual Studio Code cùng với ngôn ngữ NodeJS (back-end) và ReactJS (front-end) với cơ sở dữ liệu là MySQL. Và dùng phần Android Studio để lập trình ứng dụng android.
- Phần 4: Sau khi hoàn thành xong, tiến hành dùng thử ở phương diện người dùng, tiến hành dò xét các lỗi kỹ thuật. Từ đó đưa ra nhận định và tìm ra phương pháp khắc phục những lỗi xuất hiện trong quá trình chạy thử.
- Phần 5: Kiểm tra tổng quát lần cuối sau đó bàn giao.

6. Dự kiến kết quả đạt được:

Hoàn thành trang web, ứng dụng android đáp ứng được những mục tiêu đã đặt ra, có thể phát triển và cải tiến sau này.

LỜI CẢM ƠN

Sau hơn ba tháng học tập và nghiên cứu đề tài, em đã được mở mang kiến thức và được hiểu biết nhiều hơn về lĩnh vực công nghệ thông tin qua sự quan tâm, hướng dẫn và chia sẻ tận tình của Thầy Hoàng Văn Hiếu.

Với tấm lòng cảm ơn sâu sắc, em xin gửi tới Quý Thầy/ Cô trường Đại Học Công Nghệ Thành Phố Hồ Chí Minh nói chung và khoa Công Nghệ Thông Tin nói riêng, những người đã trang bị hành trang kiến thức nền tảng cho đề tài này. Đặc biệt là Thầy Hoàng Văn Hiếu đã nhiệt tình mô tả yêu cầu ở phương diện người dùng một cách chi tiết và cụ thể nhất, đóng góp những ý kiến quý báu cho việc hoàn thành đề tài này, đồng thời gửi lời cảm ơn chân thành nhất đến các bạn học chung trong lớp đã chia sẻ kinh nghiệm và kiến thức giúp em hoàn thành đồ án này. Xin Thầy nhận nơi em lời cảm ơn chân thành nhất.

Em xin chân thành cảm ơn!

Mục Lục

| | |
|---|----|
| CHƯƠNG I: TỔNG QUAN..... | 1 |
| 1.1. Tổng quan về vấn đề được nghiên cứu: | 1 |
| 1.1.1. Khảo sát thực trạng: | 1 |
| 1.1.2. Đánh giá: | 1 |
| 1.2. Phân tích đề án:..... | 1 |
| 1.2.1. Đối tượng và phạm vi trang web: | 1 |
| 1.2.2. Mô tả phương án tổng quan: | 2 |
| CHƯƠNG II: CƠ SỞ LÝ THUYẾT | 4 |
| 2.1. Các khái niệm và cơ chế hoạt động: | 4 |
| 2.1.1. Khái niệm lập trình hướng đối tượng: | 4 |
| 2.1.2. Lợi ích của lập trình hướng đối tượng: | 6 |
| 2.2. Các công nghệ sử dụng: | 7 |
| 2.2.1. JavaScript:..... | 7 |
| 2.2.2. NodeJS: | 8 |
| 2.2.3. ReactJS: | 8 |
| 2.2.4. HTML: | 9 |
| 2.2.5. CSS: | 10 |
| 2.2.5.1. Express FrameWork: | 10 |
| 2.2.6. MySQL: | 11 |
| 2.2.7. Android (JAVA): | 11 |
| 2.2.8. SQLite:..... | 13 |
| CHƯƠNG III: KẾT QUẢ THỰC NGHIỆM | 15 |
| 3.1. Các thuộc tính của các thực thể: | 15 |
| 3.2. Sơ đồ mối quan hệ giữa các bảng: | 18 |
| 3.3. UseCase:..... | 18 |
| 3.4. Sơ đồ hoạt động: | 29 |
| 3.5. Sơ đồ lớp: | 30 |
| 3.6. Mô tả giao diện của website và ứng dụng android: | 31 |
| 3.6.1. Mô tả giao diện website:..... | 31 |
| 3.6.2. Mô tả giao diện ứng dụng android:..... | 41 |
| CHƯƠNG IV: KẾT LUẬN..... | 43 |

| | |
|---------------------------------|----|
| 4.1. Kết quả thực nghiệm: | 43 |
| 4.2. Đánh giá: | 43 |
| 4.2.1. Ưu điểm: | 43 |
| 4.2.2. Nhược điểm: | 43 |
| 4.3. Hướng phát triển: | 44 |
| TÀI LIỆU THAM KHẢO | 45 |

Mục Lục Bảng

| | |
|--|----|
| Bảng 1 Bảng DanhMucQuyen..... | 15 |
| Bảng 2: Bảng TaiKhoan | 15 |
| Bảng 3: Bảng NhanVien..... | 15 |
| Bảng 4: Bảng DonViQuanLy | 16 |
| Bảng 5: Bảng TaiSan..... | 16 |
| Bảng 6: Bảng LoaiTaiSan | 16 |
| Bảng 7: Bảng DonViTinh..... | 17 |
| Bảng 8: Bảng MuonTaiSan | 17 |
| Bảng 9: Bảng ChiTietMuon | 17 |
| Bảng 10: Danh sách Actor..... | 18 |
| Bảng 11: Danh Sách UseCase | 19 |
| Bảng 12: Đặc tả UseCase "Đăng nhập" | 20 |
| Bảng 13: Đặc tả UseCase “Quản lý tài sản”..... | 21 |
| Bảng 14: Đặc tả UseCase “Thêm tài sản” | 21 |
| Bảng 15: Đặc tả UseCase “Xóa tài sản” | 22 |
| Bảng 16: Đặc tả UseCase “Sửa thông tin tài sản” | 23 |
| Bảng 17: Đặc tả UseCase “Quản lý loại tài sản” | 23 |
| Bảng 18: Đặc tả UseCase “Thêm loại tài sản” | 24 |
| Bảng 19: Đặc tả UseCase “Xóa loại tài sản” | 25 |
| Bảng 20: Đặc tả UseCase “Sửa thông tin loại tài sản” | 25 |
| Bảng 21: Đặc tả UseCase “Quản lý đơn vị quản lý” | 26 |
| Bảng 22: Đặc tả UseCase “Thêm đơn vị” | 27 |
| Bảng 23: Đặc tả UseCase “Xóa đơn vị” | 28 |
| Bảng 24: Đặc tả UseCase “Sửa thông tin đơn vị” | 28 |

Mục Lục Hình Ảnh

| | |
|---|----|
| Hình 1: Sơ đồ mối quan hệ giữa các bảng..... | 18 |
| Hình 2: Sơ đồ UseCase tổng quan..... | 19 |
| Hình 3: UseCase Quản lý tài sản..... | 20 |
| Hình 4: UseCase Quản lý loại tài sản..... | 23 |
| Hình 5: UseCase Quản lý đơn vị quản lý | 26 |
| Hình 6: Sơ đồ hoạt động của chức năng Đăng nhập | 29 |
| Hình 7: Sơ đồ hoạt động của chức năng thêm tài sản | 29 |
| Hình 8: Sơ đồ hoạt động của chức năng sửa tài sản..... | 30 |
| Hình 9: Sơ đồ lớp | 30 |
| Hình 10: Giao diện trang đăng nhập..... | 31 |
| Hình 11: Giao diện trang chủ | 31 |
| Hình 12: Giao diện trang quản lý loại tài sản..... | 32 |
| Hình 13: : Giao diện trang thêm loại tài sản..... | 32 |
| Hình 14: Giao diện trang sửa loại tài sản | 33 |
| Hình 15: Giao diện trang quản lý tài sản..... | 33 |
| Hình 16: Giao diện trang thêm tài sản..... | 34 |
| Hình 17: Giao diện trang thêm tài sản bằng file excel | 34 |
| Hình 18: Giao diện trang chi tiết tài sản..... | 35 |
| Hình 19: Giao diện trang sửa thông tin tài sản..... | 35 |
| Hình 20: Giao diện trang tài sản mới | 36 |
| Hình 21: Giao diện trang đơn vị tính..... | 36 |
| Hình 22: Giao diện trang thêm mới đơn vị tính | 37 |
| Hình 23: Giao diện trang đơn vị quản lý | 37 |
| Hình 24: Giao diện trang quản lý tài khoản | 38 |
| Hình 25: Giao diện trang thêm mới tài khoản | 38 |
| Hình 26: Giao diện trang sửa thông tin tài khoản | 39 |
| Hình 27: Giao diện trang quản lý nhân viên | 39 |
| Hình 28: Giao diện trang thêm mới nhân viên | 40 |
| Hình 29: Giao diện trang đăng nhập ứng dụng android | 41 |

| | |
|--|----|
| Hình 30: Giao diện trang hiển thị lịch sử quét | 41 |
| Hình 31: Giao diện trang quét mã QRCODE..... | 42 |
| Hình 32: Giao diện trang hiển thị chi tiết thông tin tài sản | 42 |

CHƯƠNG I: TỔNG QUAN

1.1. Tổng quan về vấn đề được nghiên cứu:

1.1.1. Khảo sát thực trạng:

Cuộc cách mạng khoa học kỹ thuật đã và đang mang lại những chuyển biến mạnh mẽ trên toàn thế giới. Với sự phát triển nhanh như vũ bão. Việc áp dụng công nghệ thông tin trong hoạt động kinh tế đem lại lợi ích to lớn cho toàn xã hội. Cùng với sự phát triển của cuộc cách mạng công nghệ thì nhu cầu của con người về các phần mềm giúp ích cho các công việc quản lý và lưu trữ dữ liệu thay vì lưu trữ theo cách truyền thống là lưu trên các giấy tờ, sổ sách, ... dẫn tới việc kiểm tra lại các số liệu, thông tin gặp nhiều khó khăn.

1.1.2. Đánh giá:

1.1.2.1. Ưu điểm:

- Cách sử dụng đơn giản
- Giao diện thân thiện với người dùng
- Chỉ cần laptop có kết nối internet và tài khoản đăng nhập hệ thống là có thể xem thông tin, quản lý tài sản, ...

1.1.2.2. Nhược điểm:

- Có một điều mà ai cũng thấy khi gia nhập cộng đồng mạng đó là SỰ BẢO MẬT. Sự bảo mật ở internet không được đảm bảo, chưa có một luận hay chính sách nào quy định rõ về an ninh mạng
- Còn một vấn đề nữa không kém vấn đề bảo mật của trang web chính là tốc độ xử lý của trang web, tốc độ phản hồi của trang web cho người dùng.

1.2. Phân tích đề án:

1.2.1. Đối tượng và phạm vi trang web:

1.2.1.1. Đối tượng:

- Chức năng của trang web được xây dựng hướng đến các công ty, các trường học có nhu cầu cần quản lý cơ sở vật chất.

1.2.1.2. Phạm vi:

- ❖ Thông tin tổng quan:

- Tên website: Facility Management.

❖ Mục tiêu:

- Website cho phép người dùng quản lý thông tin về tài sản, loại tài sản, tài khoản, đơn vị tính, đơn vị quản lý,... và các trang phục vụ cho các chức năng thêm, xoá, sửa các thông tin.
- Đảm bảo cơ sở dữ liệu có độ bảo mật tốt và tin cậy cao.

❖ Lợi ích mang lại:

- Tạo sự tiện lợi, nhanh chóng cho người dùng.
- Tiết kiệm được thời gian và chi phí.
- Tạo thương hiệu, tiếng vang và lan rộng cho mọi người trong xã hội.

❖ Các bước thực hiện để hoàn thành dự án:

- Lập kế hoạch phát triển.
- Phân tích trang web.
- Thiết kế database.
- Thiết kế UI.
- Xử lý logic.
- Kiểm tra và xử lý lỗi.

1.2.2. *Mô tả phương án tổng quan:*

1.2.2.1. Phương án lưu trữ:

❖ Cơ sở dữ liệu tập trung:

- Là phương án đưa dữ liệu về một nơi.
- Giúp quản lý dữ liệu chặt chẽ hơn, tăng tính bảo mật vì mọi thao tác trên dữ liệu được thực hiện ở một nơi.
- Tốc độ thao tác dữ liệu hạn chế do nhiều thao tác cùng một lúc vào dữ liệu ở mọi nơi.

❖ Cơ sở dữ liệu phân tán:

- Ngược lại với cơ sở dữ liệu tập trung.
- Tốc độ thao tác dữ liệu nhanh hơn cơ sở dữ liệu tập trung.
- Chi phí đầu tư cao.

- Thiết kế dữ liệu đối tượng tương đối khó khăn, không chặt chẽ, có thể bị lỗi không cập nhật cho tất cả các nơi lưu trữ.
- Chỉ phù hợp cho cơ sở dữ liệu lớn, có khoảng cách địa lý.

❖ Các hệ cơ sở dữ liệu:

- Microsoft Acces, Microsoft SQL Server, MySQL, Oracle...

1.2.2.2. Phương án khả thi:

Với các mô hình dữ liệu trên, mô hình dữ liệu được áp dụng cho hệ thống là mô hình dữ liệu tập trung vì những lợi ích sau:

- Với sự phát triển của công nghệ hiện nay thì tốc độ đường truyền, dung lượng bộ nhớ không là vấn đề lớn, cơ sở dữ liệu tập trung giúp chúng ta dễ dàng sao lưu, phục hồi, bảo đảm an toàn dữ liệu.
- Về mặt phần cứng thì chi phí đầu tư cho mô hình này không cao.
- Về mặt bảo mật dữ liệu, cần phân quyền đối với người sử dụng hệ thống, mặt khác giúp việc quản lý được chặt chẽ hơn.

CHƯƠNG II: CƠ SỞ LÝ THUYẾT

2.1. Các khái niệm và cơ chế hoạt động:

2.1.1. Khái niệm lập trình hướng đối tượng:

Lập trình hướng đối tượng (gọi tắt là OOP, từ chữ Anh ngữ object-oriented programming), hay còn gọi là lập trình định hướng đối tượng, là kỹ thuật lập trình hỗ trợ công nghệ đối tượng. OOP được xem là giúp tăng năng suất, đơn giản hóa độ phức tạp khi bảo trì cũng như mở rộng phần mềm bằng cách cho phép lập trình viên tập trung vào các đối tượng phần mềm ở bậc cao hơn. Ngoài ra, nhiều người còn cho rằng OOP dễ tiếp thu hơn cho những người mới học về lập trình hơn là các phương pháp trước đó.

Một cách giản lược, đây là khái niệm và là một nỗ lực nhằm giảm nhẹ các thao tác viết mã cho người lập trình, cho phép họ tạo ra các ứng dụng mà các yếu tố bên ngoài có thể tương tác với các chương trình đó giống như là tương tác với các đối tượng vật lý.

Những đối tượng trong một ngôn ngữ OOP là các kết hợp giữa mã và dữ liệu mà chúng được nhìn nhận như là một đơn vị duy nhất. Mỗi đối tượng có một tên riêng biệt và tất cả các tham chiếu đến đối tượng đó được tiến hành qua tên của nó. Như vậy, mỗi đối tượng có khả năng nhận vào các thông báo, xử lý dữ liệu (bên trong của nó), và gửi ra hay trả lời đến các đối tượng khác hay đến môi trường.

Đối tượng (object): Các dữ liệu và chỉ thị được kết hợp vào một đơn vị đầy đủ tạo nên một đối tượng. Đơn vị này tương đương với một chương trình con và vì thế các đối tượng sẽ được chia thành hai bộ phận chính: phần các phương thức (method) và phần các thuộc tính (attribute / Properties). Trong thực tế, các phương thức của đối tượng là các hàm và các thuộc tính của nó là các biến, các tham số hay hằng nội tại của một đối tượng (hay nói cách khác tập hợp các dữ liệu nội tại tạo thành thuộc tính của đối tượng).

Các phương thức là phương tiện để sử dụng một đối tượng trong khi các thuộc tính sẽ mô tả đối tượng có những tính chất gì.

Các phương thức và các thuộc tính thường gắn chặt với thực thể các đặc tính và sử dụng của một đối tượng.

Trong thực tế, các đối tượng thường được trừu tượng hóa qua việc định nghĩa của các lớp (class). Tập hợp các giá trị hiện có của các thuộc tính tạo nên trạng thái của một đối tượng.

Mỗi phương thức hay mỗi dữ liệu nội tại cùng với các tính chất được định nghĩa (bởi người lập trình) được xem là một đặc tính riêng của đối tượng. Nếu không có gì làm lẫn thì tập hợp các đặc tính này gọi chung là đặc tính của đối tượng.

Lập trình hướng đối tượng là một phương pháp lập trình có 4 tính chất chính như sau:

- Tính trừu tượng (abstraction): Đây là khả năng của chương trình bỏ qua hay không chú ý đến một số khía cạnh của thông tin mà nó đang trực tiếp làm việc lên, nghĩa là nó có khả năng tập trung vào những cốt lõi cần thiết. Mỗi đối tượng phục vụ như một “động tử” có thể hoàn tất các công việc một cách nội bộ, báo cáo, thay đổi trạng thái của nó và liên lạc với các đối tượng khác mà không cần cho biết làm cách nào đối tượng tiến hành thao tác. Tính chất này thường được gọi là sự trừu tượng của dữ liệu. Tính trừu tượng còn thể hiện qua việc một đối tượng ban đầu có thể có một số đặc điểm chung cho nhiều đối tượng khác như sự mở rộng của nó nhưng bản thân đối tượng ban đầu này có thể không có biện pháp thi hành. Tính trừu tượng này thường được xác định trong khái niệm gọi là lớp trừu tượng hay lớp cơ sở trừu tượng.
- Tính đóng gói (encapsulation) và che giấu thông tin (information hiding): Tính chất này không cho phép người sử dụng các đối tượng thay đổi trạng thái nội tại của một đối tượng. Chỉ có các phương thức nội tại của đối tượng cho phép thay đổi trạng thái của nó. Việc cho phép môi trường bên ngoài tác động lên các dữ liệu nội tại của một đối tượng theo cách nào là hoàn toàn tùy thuộc vào người viết mã. Đây là tính chất đảm bảo sự toàn vẹn của đối tượng.
- Tính đa hình (polymorphism): Thể hiện thông qua việc gửi các thông điệp (message). Việc gửi các thông điệp này có thể so sánh như việc gọi các hàm bên trong của một đối tượng. Các phương thức dùng trả lời cho một thông điệp sẽ tùy theo đối tượng mà thông điệp đó được gửi tới sẽ có phản ứng khác nhau. Người lập trình có thể định nghĩa một đặc tính (chẳng hạn việc thông qua tên các phương thức) cho một loạt các đối tượng gần nhau nhưng khi thi hành thì dùng cùng một tên gọi là sự thi hành của mỗi đối tượng

sẽ tự động xảy ra tương ứng theo đặc tính của từng đối tượng sẽ tự động xảy ra tương ứng theo đặc tính của từng đối tượng mà không bị nhầm lẫn.

- Tính kế thừa (inheritance): Đặc tính này cho phép một đối tượng có thể có sẵn các đặc tính mà đối tượng khác đã có thông qua kế thừa. Điều này cho phép các đối tượng chia sẻ hay mở rộng các đặc tính sẵn có mà không phải tiến hành định nghĩa lại. Tuy nhiên, không phải ngôn ngữ định hướng đối tượng nào cũng có tính chất này.

2.1.2. *Lợi ích của lập trình hướng đối tượng:*

Tại sao lập trình hướng đối tượng lại được sử dụng rộng rãi để giải quyết các vấn đề khi xây dựng phần mềm ngày nay ? Trong thập niên 70 và 80, ngôn ngữ lập trình hướng thủ tục như C, Pascal và Fortran được sử dụng phổ biến để xây dựng hệ thống phần mềm. Ngôn ngữ thủ tục tổ chức theo hướng chạy trình tự các dòng từ trên xuống. Nói cách khác, chương trình là một chuỗi các bước nối tiếp nhau sau khi bước trước đó đã hoàn thành. Kiểu lập trình này chỉ hoạt động tốt với chương trình nhỏ chỉ gồm khoảng vài trăm dòng lệnh, nhưng các chương trình ngày càng lớn dần và chúng trở nên khó quản lý và sửa lỗi.

Trong một nỗ lực để quản lý kích thước không ngừng lớn dần của các chương trình, lập trình cấu trúc (structured programming) được giới thiệu để chia nhỏ mã ra thành những đoạn nhỏ được gọi là hàm (function) hoặc thủ tục (procedure). Đây là một sự cải tiến lớn, nhưng các chương trình thi hành những chức năng phức tạp hơn và tương tác với nhiều hệ thống khác, nó bắt đầu để lộ những khuyết điểm dưới đây:

- Khó bảo trì.
- Tồn tại những chức năng rất khó chỉnh sửa mà không gây ảnh hưởng đến các chức năng khác của hệ thống.
- Các chương trình mới về cơ bản phải xây dựng lại từ đầu.
- Lập trình không có lợi cho nhóm phát triển. Các lập trình viên phải biết mọi khía cạnh cách chương trình làm việc và không thể tách riêng họ vào một khía cạnh nào đó của hệ thống.
- Khó chuyển đổi từ mô hình thực tế sang mô hình lập trình
- Làm việc độc lập tốt nhưng không thích hợp tốt vào các hệ thống khác.

Ngoài các khuyết điểm, vài sự tiến triển của hệ thống máy tính tạo thêm khó khăn cho việc tiếp cận các ngôn ngữ cấu trúc, chẳng hạn như:

- Không lập trình viên nào được yêu cầu và được truy cập trực tiếp chương trình thông qua giao diện đồ họa người dùng và máy tính của họ.
- Người dùng yêu cầu trực quan hơn, ít cấu trúc hơn khi tương tác với chương trình.
- Hệ thống máy tính phát triển theo mô hình phân tán, nơi mà giao diện người dùng, cơ sở dữ liệu phụ trợ được liên kết lỏng lẻo (loosely coupled) và có thể truy cập từ mạng internet.

Và như một điều tất yếu, các công ty phần mềm đã sử dụng phương pháp lập trình hướng đối tượng để giải quyết các vấn đề của họ. Những lợi ích mà họ có được như sau:

- Tích hợp tốt với các hệ thống máy tính có sẵn.
- Khả năng bảo trì và thay đổi chương trình nhanh chóng và hiệu quả.
- Khả năng tạo giao diện người dùng trực quan.

2.2. Các công nghệ sử dụng:

2.2.1. JavaScript:

2.2.1.1. JavaScript là gì:

JavaScript là một ngôn ngữ lập trình hoặc ngôn ngữ kịch bản cho phép triển khai những chức năng phức tạp trên trang web như hiển thị các cập nhật nội dung kịp thời, tương tác với bản đồ, hoạt cảnh 2D/3D vv... - điều có sự hỗ trợ của JavaScript. Nó là lớp thứ ba của chiếc bánh tiêu chuẩn của các công nghệ web, hai trong số chúng (HTML và CSS) đã được chúng tôi trình bày rất chi tiết trong các phần khác của Learning Area.

2.2.1.2. Ưu điểm:

- JavaScript là ngôn ngữ dễ học
- Nó được phát triển bởi Netscape, và đang được dùng trên 92% webstie.
- JS có thể được gắn vào một element của trang web hoặc sự kiện của trang web như cú click chuột.
- Hoạt động trên đa trình duyệt và đa thiết bị.
- Nhanh và nhẹ hơn các ngôn ngữ lập trình khác.

- Bạn có thể thêm JavaScript trực tiếp vào HTML hoặc bạn có thể lưu nó trên files riêng biệt và gọi lên khi cần.
- JS code snippets lớn.

2.2.1.3. Nhược điểm:

- Dễ bị khai thác.
- Có thể được dùng để thực thi mã độc trên máy tính của người dùng.
- Nhiều khi không được hỗ trợ trên mọi trình duyệt.
- Có thể bị triển khai khác nhau tùy từng thiết bị dẫn đến việc không đồng nhất.

2.2.2. *NodeJS:*

2.2.2.1. NodeJs là gì:

Node.js là một nền tảng chạy trên môi trường V8 JavaScript runtime - một trình thông dịch JavaScript cực nhanh chạy trên trình duyệt Chrome. Bình thường thì bạn cũng có thể tải bộ V8 và nhúng nó vào bất cứ thứ gì; Node.js làm điều đó đối với các web server. JavaScript suy cho cùng cũng chỉ là một ngôn ngữ - vậy thì không có lý do gì để nói nó không thể sử dụng trên môi trường server tốt như là trong trình duyệt của người dùng được.

2.2.2.2. Tại sao sử dụng NodeJS:

Đầu tiên là ưu điểm về tốc độ thực thi và khả năng mở rộng. Node.js có tốc độ rất nhanh. Đó là một yêu cầu khá quan trọng khi bạn là một startup đang cố gắng tạo ra một sản phẩm lớn và muốn đảm bảo có thể mở rộng nhanh chóng, đáp ứng được một lượng lớn người dùng khi trang web của bạn phát triển lên.

Node.js có thể xử lý hàng ngàn kết nối đồng thời trong khi PHP sẽ chỉ có nước sụp đổ. Bên cạnh các lợi ích về tốc độ thực thi và khả năng mở rộng, có thể bạn cũng đã biết một chút về JavaScript, vì vậy tại sao lại phải phiền toái để học thêm về một ngôn ngữ lập trình hoàn toàn mới như PHP? Và sau đó bạn sẽ có một sự phấn khích khi học về một cái gì đó mới mẻ và gần như chưa được khám phá. Bạn còn nhớ cái cảm giác khi mà một cái gì đó mới xuất hiện và sau đó trở thành phổ biến khắp mọi nơi mà bạn hối tiếc đã không học về nó sớm hơn, và mãi mãi chỉ là người đến sau? Đừng phạm phải sai lầm như vậy lần nữa. Node.js đang ngày càng trở nên lớn mạnh hơn.

2.2.3. *ReactJS:*

2.2.3.1. ReactJS là gì:

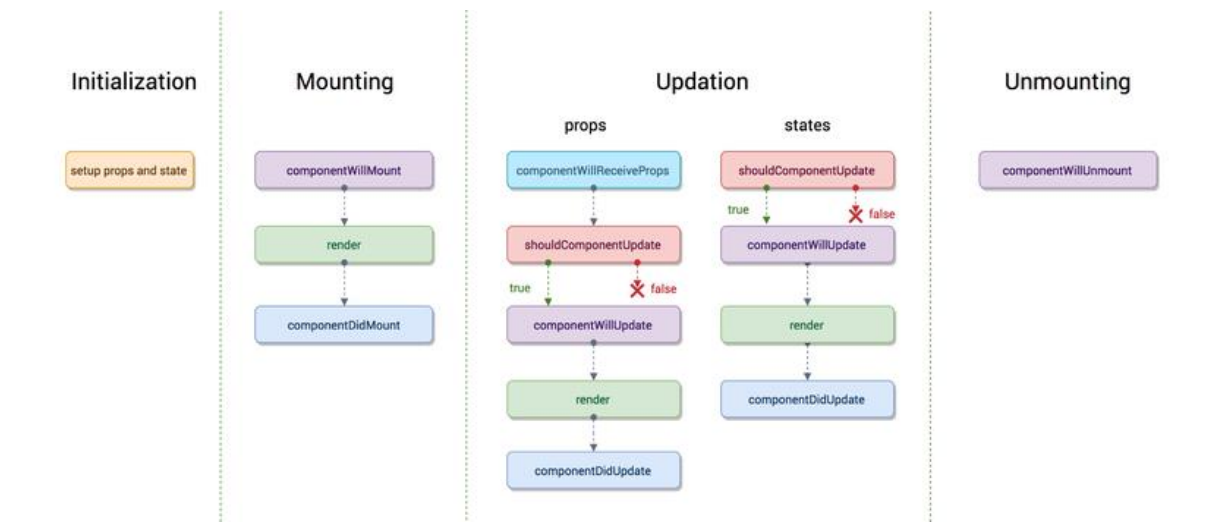
React là thư viện JavaScript phổ biến nhất để xây dựng giao diện người dùng (UI). Nó cho tốc độ phản hồi tuyệt vời khi user nhập liệu bằng cách sử dụng phương pháp mới để render trang web.

Components của công cụ này được phát triển bởi Facebook. Nó được ra mắt như một công cụ JavaScript mã nguồn mở vào năm 2013. Hiện tại, nó đã đi trước các đối thủ chính như Angular và Bootstrap, hai thư viện JavaScript bán chạy nhất thời bấy giờ.

2.2.3.2. Tại sao sử dụng ReactJS:

- Dễ sử dụng.
- Nó hỗ trợ Reusable Component trong Java.
- Viết component dễ dàng hơn.
- Hiệu suất tốt hơn với Virtual DOM.
- Thân thiện với SEO

2.2.3.3. ReactJS hoạt động như nào:



2.2.4. HTML:

HTML là chữ viết tắt của Hypertext Markup Language. Nó giúp người dùng tạo và cấu trúc các thành phần trong trang web hoặc ứng dụng, phân chia các đoạn văn, heading, links, blockquotes, ...

HTML không phải là ngôn ngữ lập trình, đồng nghĩa với việc nó không thể tạo ra các chức năng “động” được. Nó chỉ giống như Microsoft Word, dùng để bố cục và định dạng trang web.

Khi làm việc với HTML, chúng ta sẽ sử dụng cấu trúc code đơn giản (tags và attributes) để đánh dấu lên trang web. Ví dụ, chúng ta có thể tạo một đoạn văn bằng cách đặt văn bản vào trong cặp tag mở và đóng văn bản.

Tổng quan, HTML là ngôn ngữ markup, nó rất trực tiếp dễ hiểu, dễ học, và tất cả mọi người mới đều có thể bắt đầu học nó để xây dựng website.

2.2.5. CSS:

CSS là ngôn ngữ tạo phong cách cho trang web – Cascading Style Sheet language. Nó dùng để tạo phong cách và định kiểu cho những yếu tố được viết dưới dạng ngôn ngữ đánh dấu, như là HTML. Nó có thể điều khiển định dạng của nhiều trang web cùng lúc để tiết kiệm công sức cho người viết web. Nó phân biệt cách hiển thị của trang web với nội dung chính của trang bằng cách điều khiển bố cục, màu sắc, và font chữ.

CSS được phát triển bởi W3C (World Wide Web Consortium) vào năm 1996, vì một lý do đơn giản. HTML không được thiết kế để gắn tag để giúp định dạng trang web. Bạn chỉ có thể dùng nó để “đánh dấu” lên site.

Những tag như được ra mắt trong HTML phiên bản 3.2, nó gây rất nhiều rắc rối cho lập trình viên. Vì website có nhiều font khác nhau, màu nền và phong cách khác nhau. Để viết lại code cho trang web là cả một quá trình dài, cực nhọc. Vì vậy, CSS được tạo bởi W3C là để giải quyết vấn đề này.

Mối tương quan giữa HTML và CSS rất mật thiết. HTML là ngôn ngữ markup (nền tảng của site) và CSS định hình phong cách (tất cả những gì tạo nên giao diện website), chúng là không thể tách rời.

CSS về lý thuyết không có cũng được, nhưng khi đó website sẽ không chỉ là một trang chứa văn bản mà không có gì khác.

2.2.5.1. Express Framework:

Express là một framework giành cho nodejs. Nó cung cấp cho chúng ta rất nhiều tính năng mạnh mẽ trên nền tảng web cũng như trên các ứng dụng di động. Express hỗ trợ các

phương thức HTTP và middleware tạo ra một API vô cùng mạnh mẽ và dễ sử dụng. Có thể tổng hợp một số chức năng chính của express như sau:

- Thiết lập các lớp trung gian để trả về các HTTP request.
- Định nghĩa router cho phép sử dụng với các hành động khác nhau dựa trên phương thức HTTP và URL.
- Cho phép trả về các trang HTML dựa vào các tham số.

Khi cài đặt xong Express trong project của mình thì chúng ta cần cài đặt thêm một số module quan trọng đi cùng express như sau:

- body-parser: Đây là lớp trung gian, xử lý JSON, text và mã hóa URL.
- cookie-parser : Chuyển đổi header của cookie và phân bố đến các req.cookies
- multer - Xử lý phần multipart/form-data.

2.2.6. *MySQL*:

2.2.6.1. MySQL là gì:

- MySQL là một hệ quản trị cơ sở dữ liệu phổ biến nhất thế giới, được các nhà phát triển ưa chuộng.
- MySQL có tốc độ cao, ổn định, dễ sử dụng, hoạt động trên nhiều hệ điều hành khác nhau.
- MySQL rất thích hợp cho các ứng dụng có truy cập cơ sở dữ liệu trên internet.
- MySQL hỗ trợ nhiều ngôn ngữ lập trình như PHP, Perl,... trong việc phát triển website.

2.2.6.2. Mô hình Client – Server:

Máy tính cài đặt và chạy phần mềm RDBMS được gọi là client (máy khách). Mỗi khi chúng cần truy cập dữ liệu, chúng kết nối tới máy chủ (server) RDBMS. Cách thức này chính là mô hình “client-server”.

2.2.7. *Android (JAVA)*:

2.2.7.1. Lập trình android là gì:

Lập trình Android là nghề sử dụng các ngôn ngữ lập trình để viết và phát triển các phần mềm. Thế gia tăng tiện ích cho di động thiết bị, đặc biệt là di động sử dụng hệ điều hành.

Với sự phát triển của di động thông minh của mình như hiện tại, kết quả của Android thiết lập là một ngành rất nóng. Theo thống kê, hiện nay Android chính là hệ điều hành được sử dụng nhiều nhất. Và nếu bạn đang lắng nghe về phần lớn đối thủ của Android là iOS, thì bạn đừng lo lắng. Bởi từng nền tảng mới cho người dùng trải nghiệm riêng. If as iOS app back for people too many khó khăn với độ bảo mật cao. Thì đặc biệt Android lại được người dùng yêu thích sử dụng vì tùy biến dễ dàng và đa dạng hơn. Chính vì vậy mà hiện nay nhu cầu tuyển dụng lập trình viên Android tăng cao.

Có rất nhiều bạn đang quản lý rằng không biết nên chọn thiết lập web hoặc thiết lập Android. Bởi cả hai nghề này đều quá hot và đang khao khát nguồn nhân lực. Mỗi nghề sẽ có những điểm riêng và dựa vào những điểm đặc biệt đó bạn có thể chọn một ngành nghề phù hợp với mình. Tốt nhất bạn nên học chuyên sâu vào một nghề. Không nên học nửa vời. Bởi hiện nay cả ngành nghề này đều đang khao khát những lập trình viên chuyên nghiệp.

2.2.7.2. Nền tảng android:

Android là một hệ điều hành dựa trên nền tảng Linux. Nó được thiết kế ra để dành cho các thiết bị di động có màn hình cảm ứng như điện thoại thông minh hay máy tính bảng.

Android có mã nguồn mở và Google phát hành mã nguồn theo Giấy phép Apache. Chính mã nguồn mở và những giấy phép không có quá nhiều ràng buộc. Đã tạo điều kiện thuận lợi để Android phát triển. Các nhà phát triển thiết bị, mạng di động và các lập trình viên nhiệt huyết được điều chỉnh và phân phối Android một cách tự do. Thêm vào đó Android còn có một cộng đồng đông đảo các lập trình viên và các chuyên gia có thể chuyên viết các ứng dụng để mở rộng chức năng của thiết bị.

2.2.7.3. Sự phát triển của android:

Theo thống kê vào năm 2012, sau bốn năm kể từ khi ra mắt có khoảng 700.000 ứng dụng trên Android và số lượt tải ứng dụng từ Google Play. Cửa hàng ứng dụng chính của

Android, ước tính khoảng 25 tỷ lượt. Chỉ trong vòng hơn 10 năm android đã trở thành một hệ điều hành phổ biến trên thế giới. Nó đã phát triển bất chấp sự bùng nổ mạnh mẽ của hệ điều hành iOS của Apple.

Trong khi các thiết bị của Apple phát triển theo chiều hướng mức giá ngày càng tăng. Thì các thiết bị Android lại tăng cường mở rộng thị trường toàn cầu. Chính điều này đã tạo nên sự khác biệt về các con số. Vào thời điểm quý 2 năm 2017, Android chiếm 87,7% thị phần điện thoại thông minh trên toàn thế giới. Với tổng cộng 2 tỷ thiết bị đã được kích hoạt và 1,3 triệu lượt kích hoạt mỗi ngày.

Với sự phát triển đó, Android không thu mình trong những chiếc smartphone đó. Mà nó còn tìm đường đi vào nhiều các thiết bị khác. Ví dụ như các thiết bị về TV, máy chiếu, ô tô và thậm chí cả phương tiện giải trí. Bạn muốn đèn nhà mình bật sáng trong ngôi nhà thông minh của mình? Bạn có thể sử dụng giao diện màn hình cảm ứng dựa trên Android để thao tác điều khiển. Android đã mang tới nhưng thay đổi không thể ngờ tới trong cuộc sống con người. Nó làm cho thế giới trở nên tốt đẹp hơn bao giờ hết.

2.2.8. *SQLite:*

2.2.8.1. SQLite là gì:

SQLite là một thư viện phần mềm mà triển khai một SQL Database Engine, không cần máy chủ, không cần cấu hình, khép kín và nhỏ gọn. Nó là một cơ sở dữ liệu, không cần cấu hình, có nghĩa là giống như các cơ sở dữ liệu khác mà bạn không cần phải cấu hình nó trong hệ thống của mình.

SQLite engine không phải là một quy trình độc lập (standalone process) như các cơ sở dữ liệu khác, bạn có thể liên kết nó một cách tĩnh hoặc động tùy theo yêu cầu của bạn với ứng dụng của bạn. SQLite truy cập trực tiếp các file lưu trữ (storage files) của nó.

2.2.8.2. Tại sao sử dụng SQLite:

- SQLite không yêu cầu một quy trình hoặc hệ thống máy chủ riêng biệt để hoạt động.
- SQLite không cần cấu hình, có nghĩa là không cần thiết lập hoặc quản trị.
- Một cơ sở dữ liệu SQLite hoàn chỉnh được lưu trữ trong một file disk đa nền tảng (cross-platform disk file).

- SQLite rất nhỏ và trọng lượng nhẹ, dưới 400KiB được cấu hình đầy đủ hoặc dưới 250KiB với các tính năng tùy chọn bị bỏ qua.
- SQLite là khép kín (self-contained), có nghĩa là không có phụ thuộc bên ngoài.
- Các transaction trong SQLite hoàn toàn tuân thủ ACID, cho phép truy cập an toàn từ nhiều tiến trình (process) hoặc luồng (thread).
- SQLite hỗ trợ hầu hết các tính năng ngôn ngữ truy vấn (query language) được tìm thấy trong tiêu chuẩn SQL92 (SQL2).
- SQLite được viết bằng ANSI-C và cung cấp API đơn giản và dễ sử dụng.
- SQLite có sẵn trên UNIX (Linux, Mac OS-X, Android, iOS) và Windows (Win32, WinCE, WinRT).

CHƯƠNG III: KẾT QUẢ THỰC NGHIỆM

3.1. Các thuộc tính của các thực thể:

Cơ sở dữ liệu được mô phỏng bằng hệ quản trị cơ sở dữ liệu Microsoft SQL SERVER.
Các thuộc tính của các thực thể trong mỗi bảng như sau:

➤ Bảng DanhMucQuyen:

| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc |
|-----|----------------|---------------|-------------|
| 1 | Id | Int | Primary key |
| 2 | TenQuyen | Nvarchar(50) | |
| 3 | MoTa | Nvarchar(100) | |

Bảng 1 Bảng DanhMucQuyen

➤ Bảng TaiKhoan:

| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc |
|-----|----------------|--------------|-------------|
| 1 | Id | Int | Primary key |
| 2 | Username | Varchar(50) | |
| 3 | MoTa | Varchar(50) | |
| 4 | IdQuyên | Int | |

Bảng 2: Bảng TaiKhoan

➤ Bảng NhanVien:

| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc |
|-----|----------------|---------------|-------------|
| 1 | Id | Int | Primary key |
| 2 | TenNV | Nvarchar(50) | |
| 3 | SoDienThoai | Varchar(11) | |
| 4 | DiaChi | Nvarchar(100) | |
| 5 | IdAccount | Int | |

Bảng 3: Bảng NhanVien

➤ Bảng DonViQuanLy:

| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc |
|-----|----------------|--------------|-------------|
| 1 | Id | Int | Primary key |
| 2 | TenDonVi | Nvarchar(50) | |
| 3 | SoDienThoai | Varchar(11) | |
| 4 | Email | Varchar(100) | |

Bảng 4: Bảng DonViQuanLy

➤ Bảng TaiSan:

| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc |
|-----|----------------|---------------|-------------|
| 1 | Id | Varchar(50) | Primary key |
| 2 | TenTaiSan | Nvarchar(50) | |
| 3 | QRCODE | Varchar(20) | |
| 4 | IdLoai | Int | |
| 5 | IdDonViTinh | Int | |
| 6 | NgayMua | Date | |
| 7 | HanSuDung | Int | |
| 8 | GiaTien | Decimal | |
| 9 | IdDonViQuanLy | Int | |
| 10 | MoTa | Nvarchar(100) | |

Bảng 5: Bảng TaiSan

➤ Bảng LoaiTaiSan:

| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc |
|-----|----------------|---------------|-------------|
| 1 | Id | Int | Primary key |
| 2 | TenLoai | Nvarchar(50) | |
| 3 | MoTa | Nvarchar(100) | |

Bảng 6: Bảng LoaiTaiSan

➤ Bảng DonViTinh:

| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc |
|-----|----------------|--------------|-------------|
| 1 | Id | Int | Primary key |
| 2 | TenDonViTinh | Nvarchar(50) | |

Bảng 7: Bảng DonViTinh

➤ BảngMuonTaiSan:

| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc |
|-----|----------------|--------------|-------------|
| 1 | Id | Int | Primary key |
| 2 | IdNv | Int | |
| 3 | SoLuong | Int | |
| 4 | HoTenNguoiMuon | Nvarchar(50) | |
| 5 | SoDienThoai | Varchar(11) | |
| 6 | Email | Varchar(50) | |
| 7 | NgayMuon | Date | |
| 8 | NgayTra | Date | |

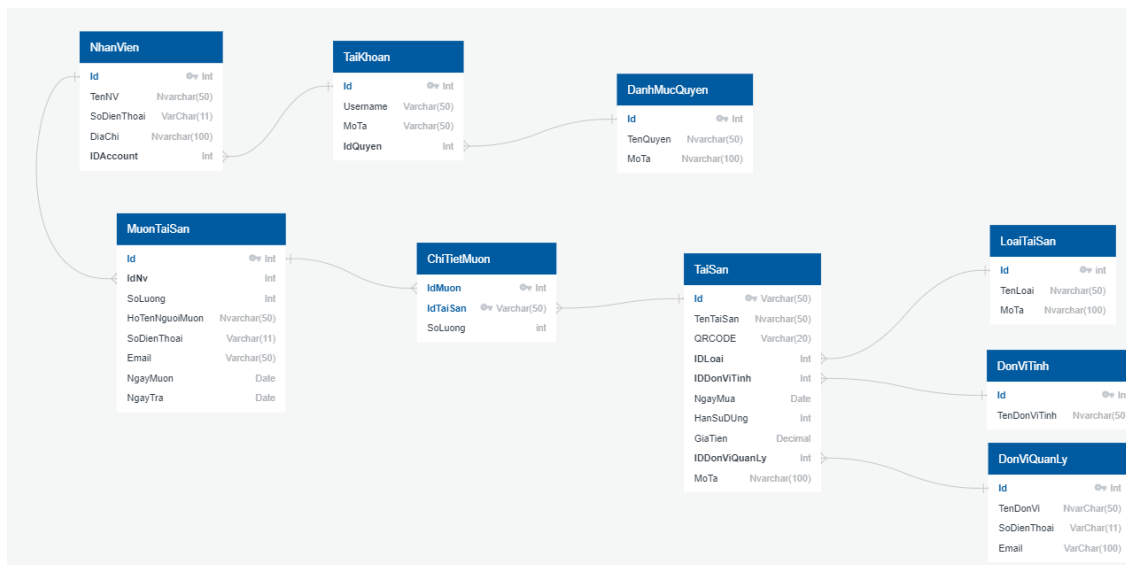
Bảng 8: Bảng MuonTaiSan

➤ BảngChiTietMuon:

| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc |
|-----|----------------|--------------|-------------|
| 1 | IdMuon | Int | Primary key |
| 2 | IdTaiSan | Varchar(50) | Primary key |
| 3 | SoLuong | Int | |

Bảng 9: Bảng ChiTietMuon

3.2. Sơ đồ mối quan hệ giữa các bảng:



Hình 1: Sơ đồ mối quan hệ giữa các bảng

3.3. UseCase:

Danh sách Actor:

| STT | Tên actor | Ý nghĩa |
|-----|-----------|---|
| 1 | Quản trị | Người có quyền làm mọi thao tác trên phần mềm |
| 2 | Nhân viên | Người được quản trị chia 1 quyền nhỏ trong hệ thống |
| 4 | Server | Nơi lưu trữ thông tin |

Bảng 10: Danh sách Actor

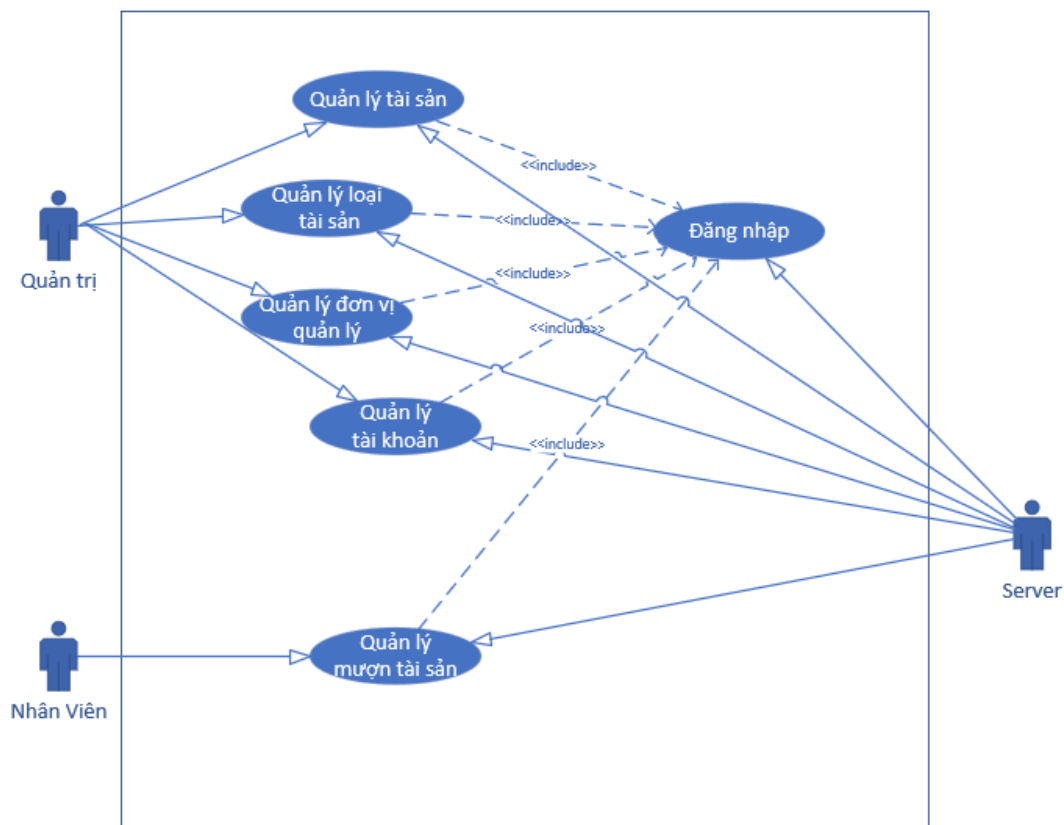
Danh sách UseCase:

| STT | Tên usecase | Usecase phân rã |
|-----|----------------------|-----------------|
| 1 | Đăng nhập | |
| 2 | Quản lý tài sản | Thêm mới |
| | | Sửa thông tin |
| | | Xoá |
| 3 | Quản lý loại tài sản | Thêm mới |

| | | |
|---|------------------------|---------------|
| | | Sửa thông tin |
| | | Xoá |
| 4 | Quản lý đơn vị quản lý | Thêm mới |
| | | Sửa thông tin |
| | | Xoá |
| 5 | Quản lý tài khoản | Thêm mới |
| | | Sửa thông tin |
| | | Xoá |
| 6 | Quản lý mượn tài sản | |

Bảng 11: Danh Sách UseCase

Sơ đồ UseCase tổng quan:

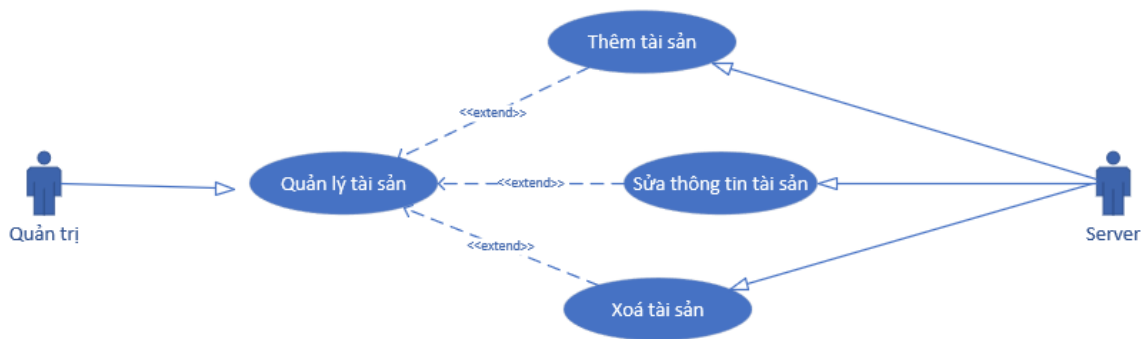


Hình 2: Sơ đồ UseCase tổng quan

3.3.1. Đặc tả UseCase “Đăng nhập”:

| | |
|----------------------|---|
| Tác nhân | Quản trị. |
| Điều kiện tiên quyết | Quản trị truy cập vào trang web. |
| Điểm mở rộng | Không có. |
| Điều kiện kết thúc | Quản trị đăng nhập thành công. |
| Dòng sự kiện chính | <p>Use-Case này được bắt đầu khi người dùng truy cập vào trang web.</p> <p>Khi người dùng để trống thì sẽ thông báo lại cho người dùng.</p> <p>Khi người dùng nhập tài khoản và mật khẩu thì hệ thống sẽ kiểm tra tài khoản đó có hợp lệ hay không. Nếu hợp lệ thì sẽ vào được trang quản lý của cửa hàng. Còn không thì sẽ thông báo lại cho người dùng.</p> |
| Điều kiện thoát | <p>Khi người dùng thực hiện thành công</p> <p>Khi người dùng chọn chức năng thoát</p> |

Bảng 12: Đặc tả UseCase “Đăng nhập”



Hình 3: UseCase Quản lý tài sản

3.3.2. Đặc tả UseCase “Quản lý tài sản”:

| | |
|----------------------|--|
| Tác nhân | Quản trị. |
| Điều kiện tiên quyết | Quản trị đã đăng nhập vào hệ thống thành công và chọn chức năng “Quản lý tài sản”. |
| Điểm mở rộng | Không có. |

| | |
|--------------------|---|
| Điều kiện kết thúc | Khi quản trị thực hiện chọn chức năng khác. |
| Dòng sự kiện chính | <ol style="list-style-type: none"> 1. Use-Case này được bắt đầu khi người dùng chọn yêu cầu “Quản lý sản phẩm”. 2. Hệ thống sẽ hiển thị màn hình có giao diện bảng có dữ liệu là các sản phẩm có trong cơ sở dữ liệu. 3. Thêm tài sản. 4. Xóa tài sản. 5. Sửa thông tin tài sản. |
| Điều kiện thoát | <ol style="list-style-type: none"> 1. Khi người dùng thực hiện thành công 2. Khi người dùng chọn chức năng thoát |

Bảng 13: Đặc tả UseCase “Quản lý tài sản”

3.3.3. Đặc tả UseCase “Thêm tài sản”:

| | |
|----------------------|--|
| Tác nhân | Quản trị. |
| Điều kiện tiên quyết | Quản trị đã đăng nhập vào hệ thống và loại tài khoản phải có chức năng vào được trang “Thêm tài sản”. |
| Điểm mở rộng | Không có |
| Điều kiện kết thúc | Tài sản mới được thêm vào database |
| Dòng sự kiện chính | <ol style="list-style-type: none"> 1. Use-Case này được bắt đầu khi người dùng chọn yêu cầu “Thêm tài sản” trong trang “Quản lý tài sản”. 2. Hệ thống sẽ hiển thị giao diện nhập thông tin tài sản và nút “Chấp nhận”. |
| Điều kiện thoát | <ol style="list-style-type: none"> 1. Khi người dùng thực hiện thành công 2. Khi người dùng chọn chức năng thoát |

Bảng 14: Đặc tả UseCase “Thêm tài sản”

3.3.4. Đặc tả UseCase “Xoá tài sản”:

| | |
|----------------------|--|
| Tác nhân | Quản trị. |
| Điều kiện tiên quyết | Quản trị nhấn vào nút có biểu tượng “X” trên bảng hiển thị sản phẩm trong trang “Quản lý sản phẩm”. |
| Điểm mở rộng | Không có. |
| Điều kiện kết thúc | Dữ liệu đã được xoá khỏi database |
| Dòng sự kiện chính | <ol style="list-style-type: none"> 1. Use-Case này được bắt đầu khi người dùng nhấn vào nút có biểu tượng “X”. 2. Hệ thống sẽ hiển thị thông báo hỏi người dùng có chắc muốn xoá hay là không. 3. Nếu người dùng chọn “OK” thì hệ thống sẽ xoá dòng dữ liệu mà người dùng chọn đó ra khỏi database. |
| Điều kiện thoát | <ol style="list-style-type: none"> 1. Khi người dùng thực hiện thành công 2. Khi người dùng chọn chức năng thoát |

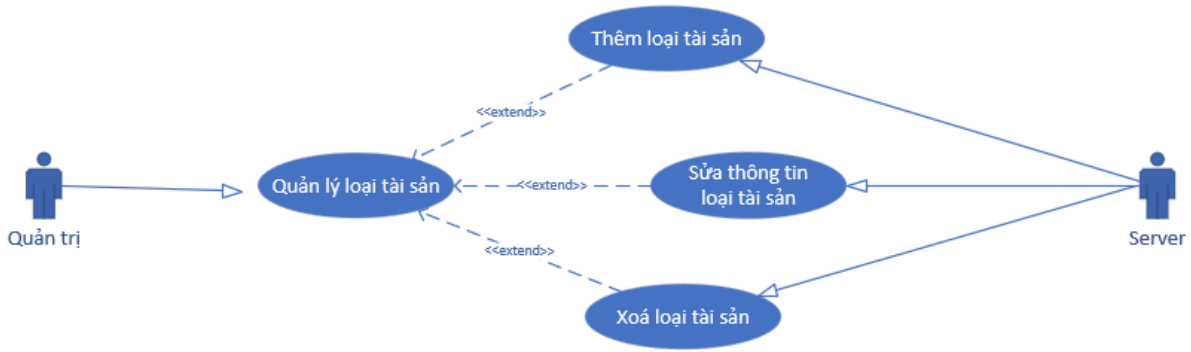
Bảng 15: Đặc tả UseCase “Xoá tài sản”

3.3.5. Đặc tả UseCase “Sửa thông tin tài sản”:

| | |
|----------------------|---|
| Tác nhân | Quản trị. |
| Điều kiện tiên quyết | Quản trị đã đăng nhập vào hệ thống và loại tài khoản phải có chức năng vào được trang “Sửa tài sản”. |
| Điểm mở rộng | Không có |
| Điều kiện kết thúc | Sản phẩm mới được sửa lại rồi lưu lại thông tin sửa vào database |
| Dòng sự kiện chính | <ol style="list-style-type: none"> 1. Use-Case này được bắt đầu khi người dùng chọn yêu cầu “Sửa tài sản” trong trang “Quản lý tài sản”. 2. Hệ thống sẽ hiển thị giao diện nhập tài sản vào database có hiển thị dữ liệu là thông tin sản |

| | |
|-----------------|--|
| | phẩm mà người dùng chọn và nút “Chấp nhận”. |
| Điều kiện thoát | <ol style="list-style-type: none"> 1. Khi người dùng thực hiện thành công 2. Khi người dùng chọn chức năng thoát |

Bảng 16: Đặc tả UseCase “Sửa thông tin tài sản”



Hình 4: UseCase Quản lý loại tài sản

3.3.6. Đặc tả UseCase “Quản lý loại tài sản”:

| | |
|----------------------|--|
| Tác nhân | Quản trị. |
| Điều kiện tiên quyết | Quản trị đã đăng nhập vào hệ thống thành công và chọn chức năng “Quản lý loại tài sản”. |
| Điểm mở rộng | Không có. |
| Điều kiện kết thúc | Khi quản trị thực hiện chọn chức năng khác. |
| Dòng sự kiện chính | <ol style="list-style-type: none"> 1. Use-Case này được bắt đầu khi người dùng chọn yêu cầu “Quản lý loại tài sản”. 2. Hệ thống sẽ hiển thị màn hình có giao diện bảng có dữ liệu là các loại tài sản có trong cơ sở dữ liệu. 3. Thêm loại tài sản. 4. Sửa thông tin loại tài sản. 5. Xoá loại tài sản. |
| Điều kiện thoát | <ol style="list-style-type: none"> 1. Khi người dùng thực hiện thành công 2. Khi người dùng chọn chức năng thoát |

Bảng 17: Đặc tả UseCase “Quản lý loại tài sản”

3.3.7. Đặc tả UseCase “Thêm loại tài sản”:

| | |
|----------------------|--|
| Tác nhân | Quản trị. |
| Điều kiện tiên quyết | Quản trị đã đăng nhập vào hệ thống và loại tài khoản phải có chức năng vào được trang “Thêm loại tài sản”. |
| Điểm mở rộng | Không có. |
| Điều kiện kết thúc | Loại tài sản mới được thêm vào database. |
| Dòng sự kiện chính | <ol style="list-style-type: none"> 1. Use-Case này được bắt đầu khi người dùng chọn yêu cầu “Thêm loại tài sản” trong trang “Quản lý loại tài sản”. 2. Hệ thống sẽ hiển thị giao diện nhập loại tài sản vào database và nút “Chấp nhận”. |
| Điều kiện thoát | <ol style="list-style-type: none"> 1. Khi người dùng thực hiện thành công 2. Khi người dùng chọn chức năng thoát |

Bảng 18: Đặc tả UseCase “Thêm loại tài sản”

3.3.8. Đặc tả UseCase “Xoá loại tài sản”:

| | |
|----------------------|--|
| Tác nhân | Quản trị. |
| Điều kiện tiên quyết | Quản trị nhấn vào nút có biểu tượng “X” trên bảng hiển thị tài sản trong trang “Quản lý loại tài sản”. |
| Điểm mở rộng | Không có. |
| Điều kiện kết thúc | Dữ liệu đã được xoá khỏi database |
| Dòng sự kiện chính | <ol style="list-style-type: none"> 1. Use-Case này được bắt đầu khi người dùng nhấn vào nút có biểu tượng “X”. 2. Hệ thống sẽ hiển thị thông báo hỏi người dùng có chắc muốn xoá hay là không. 3. Nếu người dùng chọn “OK” thì hệ thống sẽ xoá dòng dữ liệu mà người dùng chọn đó ra khỏi database. |

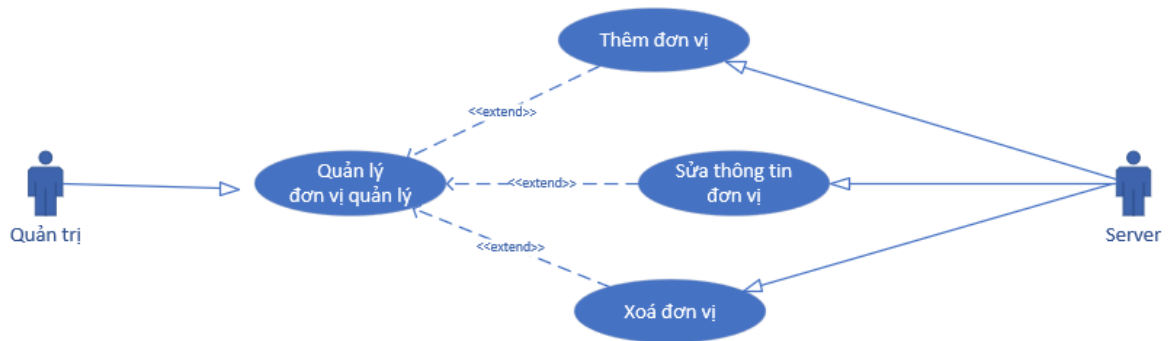
| | |
|-----------------|--|
| Điều kiện thoát | <ol style="list-style-type: none"> 1. Khi người dùng thực hiện thành công 2. Khi người dùng chọn chức năng thoát |
|-----------------|--|

Bảng 19: Đặc tả UseCase “Xoá loại tài sản”

3.3.9. Đặc tả UseCase “Sửa thông tin loại tài sản”:

| | |
|----------------------|--|
| Tác nhân | Quản trị. |
| Điều kiện tiên quyết | Quản trị đã đăng nhập vào hệ thống và loại tài khoản phải có chức năng vào được trang “Sửa loại tài sản”. |
| Điểm mở rộng | Không có |
| Điều kiện kết thúc | Loại tài sản mới được sửa lại rồi lưu lại thông tin sửa vào database |
| Dòng sự kiện chính | <ol style="list-style-type: none"> 1. Use-Case này được bắt đầu khi người dùng chọn yêu cầu “Sửa loại tài sản” trong trang “Quản lý loại tài sản”. 2. Hệ thống sẽ hiển thị giao diện nhập loại tài sản vào database có hiển thị dữ liệu là thông tin loại tài sản mà người dùng chọn và nút “Chấp nhận”. |
| Điều kiện thoát | <ol style="list-style-type: none"> 1. Khi người dùng thực hiện thành công 2. Khi người dùng chọn chức năng thoát |

Bảng 20: Đặc tả UseCase “Sửa thông tin loại tài sản”



Hình 5: UseCase Quản lý đơn vị quản lý

3.3.10. Đặc tả UseCase “Quản lý đơn vị quản lý”:

| | |
|----------------------|--|
| Tác nhân | Quản trị. |
| Điều kiện tiên quyết | Quản trị đã đăng nhập vào hệ thống thành công và chọn chức năng “Quản lý đơn vị quản lý”. |
| Điểm mở rộng | Không có. |
| Điều kiện kết thúc | Khi quản trị thực hiện chọn chức năng khác. |
| Dòng sự kiện chính | <ol style="list-style-type: none"> 1. Use-Case này được bắt đầu khi người dùng chọn yêu cầu “Quản lý đơn vị quản lý”. 2. Hệ thống sẽ hiển thị màn hình có giao diện bảng có dữ liệu là các đơn vị quản lý có trong cơ sở dữ liệu. 3. Thêm đơn vị. 4. Sửa thông tin đơn vị. 5. Xoá đơn vị. |
| Điều kiện thoát | <ol style="list-style-type: none"> 1. Khi người dùng thực hiện thành công 2. Khi người dùng chọn chức năng thoát |

Bảng 21: Đặc tả UseCase “Quản lý đơn vị quản lý”

3.3.11. Đặc tả UseCase “Thêm đơn vị”:

| | |
|----------------------|--|
| Tác nhân | Quản trị. |
| Điều kiện tiên quyết | Quản trị đã đăng nhập vào hệ thống và loại tài khoản phải có chức năng vào được trang “Thêm đơn vị quản lý”. |
| Điểm mở rộng | Không có. |
| Điều kiện kết thúc | Đơn vị mới được thêm vào database. |
| Dòng sự kiện chính | <ol style="list-style-type: none"> 1. Use-Case này được bắt đầu khi người dùng chọn yêu cầu “Thêm đơn vị quản lý” trong trang “Quản lý đơn vị quản lý”. 2. Hệ thống sẽ hiển thị giao diện nhập đơn vị quản lý vào database và nút “Chấp nhận”. |
| Điều kiện thoát | <ol style="list-style-type: none"> 1. Khi người dùng thực hiện thành công 2. Khi người dùng chọn chức năng thoát |

Bảng 22: Đặc tả UseCase “Thêm đơn vị”

3.3.12. Đặc tả UseCase “Xoá đơn vị”:

| | |
|----------------------|--|
| Tác nhân | Quản trị. |
| Điều kiện tiên quyết | Quản trị nhấn vào nút có biểu tượng “X” trên bảng hiển thị đơn vị quản lý trong trang “Quản lý đơn vị quản lý”. |
| Điều kiện kết thúc | Dữ liệu đã được xoá khỏi database |
| Dòng sự kiện chính | <ol style="list-style-type: none"> 1. Use-Case này được bắt đầu khi người dùng nhấn vào nút có biểu tượng “X”. 2. Hệ thống sẽ hiển thị thông báo hỏi người dùng có chắc muốn xoá hay là không. 3. Nếu người dùng chọn “OK” thì hệ thống sẽ xoá dòng dữ liệu mà người dùng chọn đó ra khỏi database. |

| | |
|-----------------|--|
| Điều kiện thoát | <ol style="list-style-type: none"> 1. Khi người dùng thực hiện thành công 2. Khi người dùng chọn chức năng thoát |
|-----------------|--|

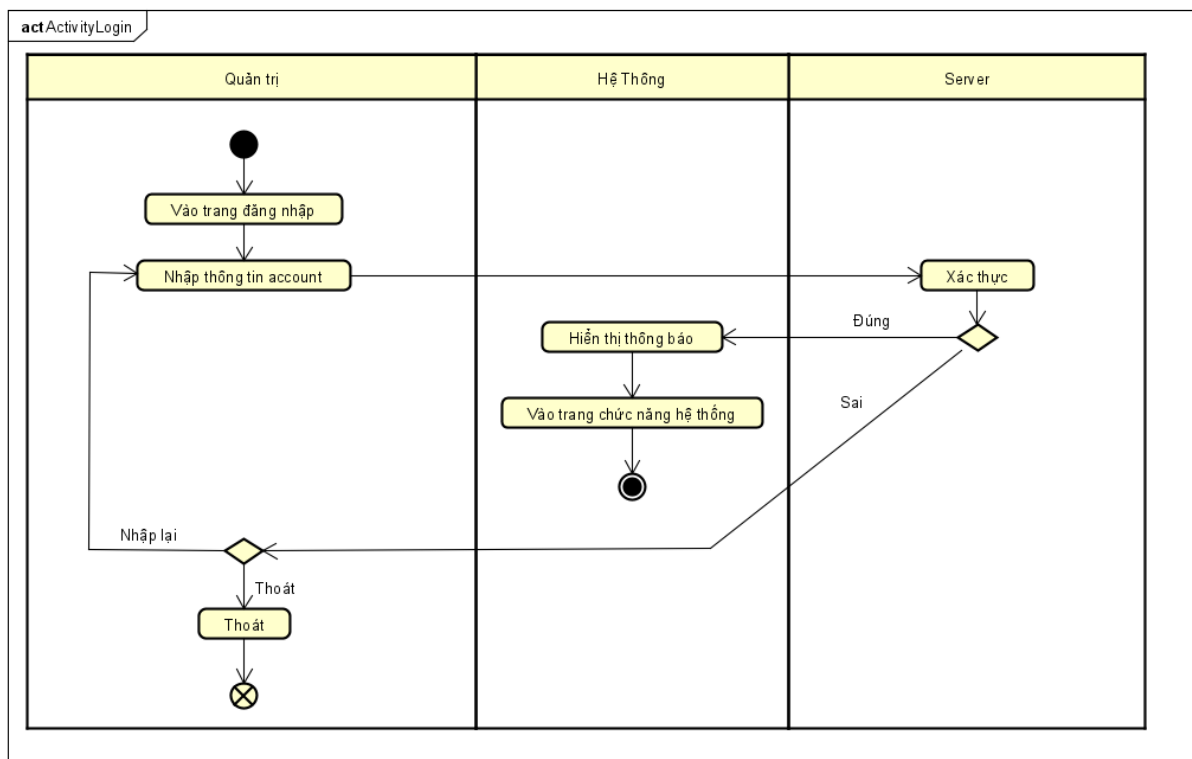
Bảng 23: Đặc tả UseCase “Xóa đơn vị”

3.3.13. Đặc tả UseCase “Sửa thông tin đơn vị”:

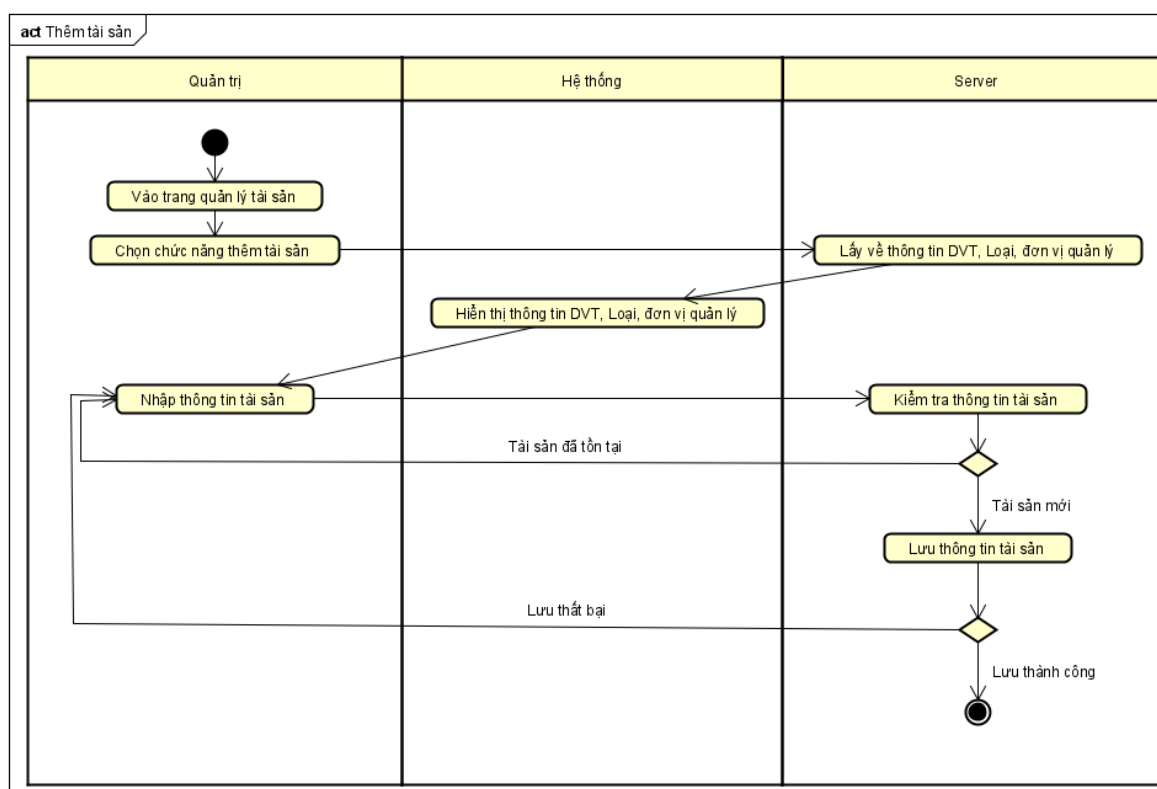
| | |
|----------------------|--|
| Tác nhân | Quản trị. |
| Điều kiện tiên quyết | Quản trị đã đăng nhập vào hệ thống và loại tài khoản phải có chức năng vào được trang “đơn vị quản lý”. |
| Điểm mở rộng | Không có |
| Điều kiện kết thúc | Đơn vị mới được sửa lại rồi lưu lại thông tin sửa vào database |
| Dòng sự kiện chính | <ol style="list-style-type: none"> 1. Use-Case này được bắt đầu khi người dùng chọn yêu cầu “Sửa đơn vị quản lý” trong trang “Quản lý đơn vị quản lý”. 2. Hệ thống sẽ hiển thị giao diện nhập nhà sản xuất vào database có hiển thị dữ liệu là thông tin nhà sản xuất mà người dùng chọn và nút “Chấp nhận”. |
| Điều kiện thoát | <ol style="list-style-type: none"> 1. Khi người dùng thực hiện thành công 2. Khi người dùng chọn chức năng thoát |

Bảng 24: Đặc tả UseCase “Sửa thông tin đơn vị”

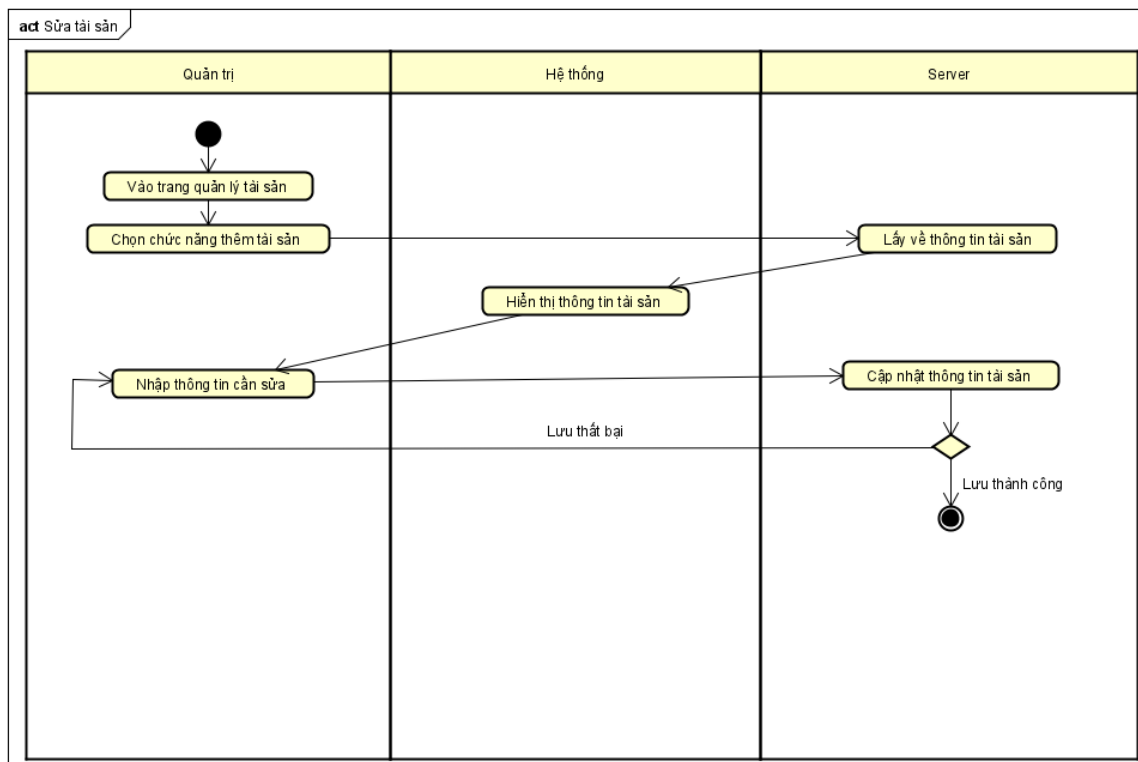
3.4. Sơ đồ hoạt động:



Hình 6: Sơ đồ hoạt động của chức năng Đăng nhập

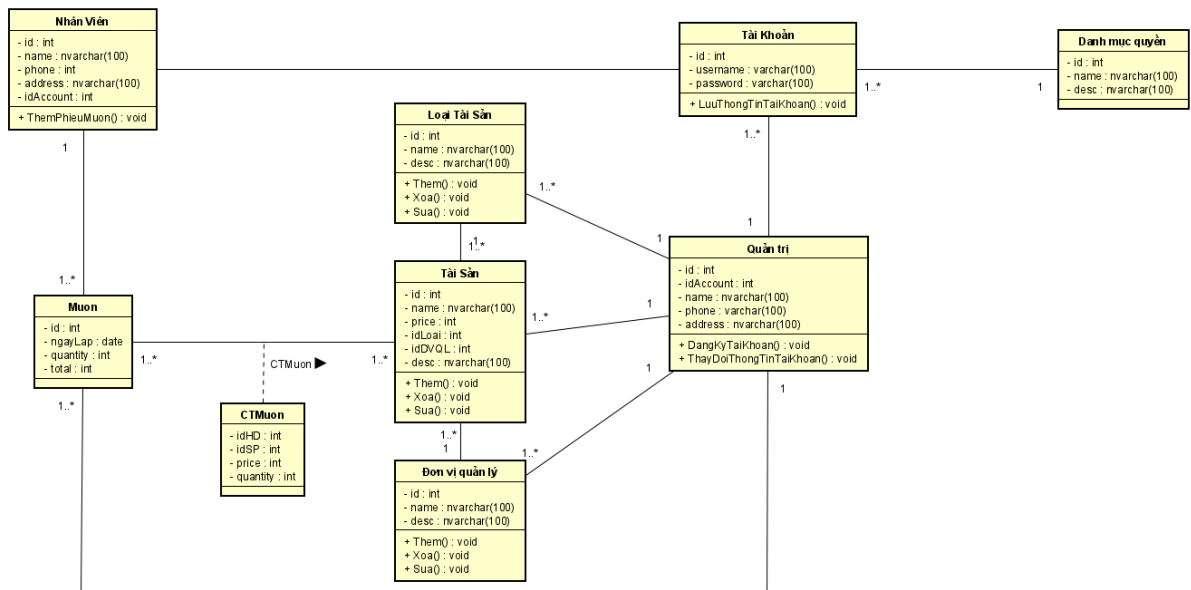


Hình 7: Sơ đồ hoạt động của chức năng thêm tài sản



Hình 8: Sơ đồ hoạt động của chức năng sửa tài sản

3.5. Sơ đồ lớp:

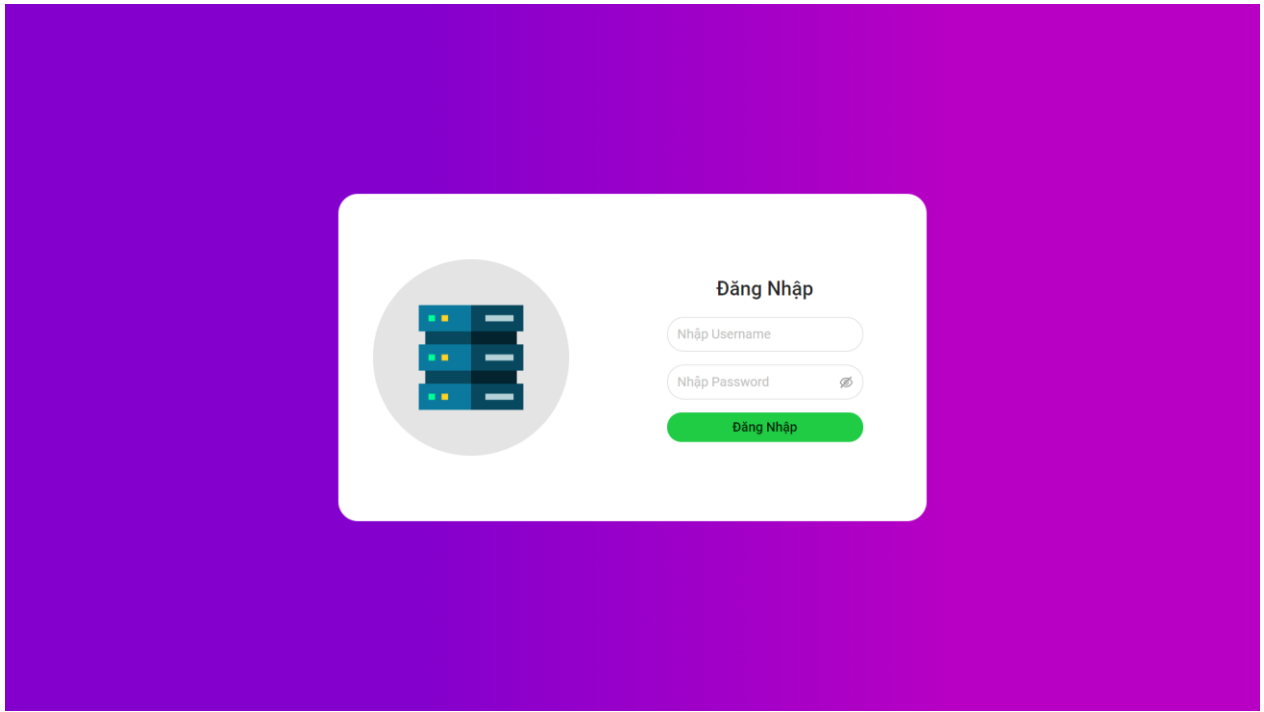


Hình 9: Sơ đồ lớp

3.6. Mô tả giao diện của website và ứng dụng android:

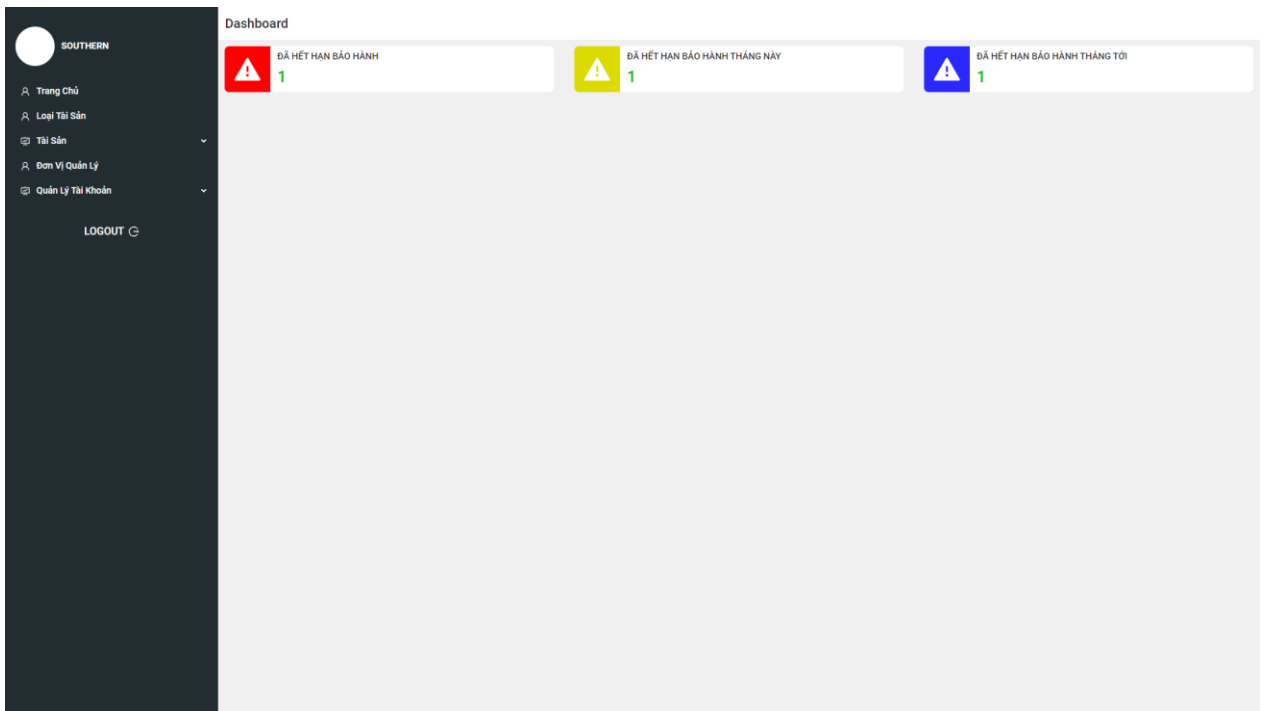
3.6.1. Mô tả giao diện website:

3.6.1.1. Giao diện trang Đăng nhập:



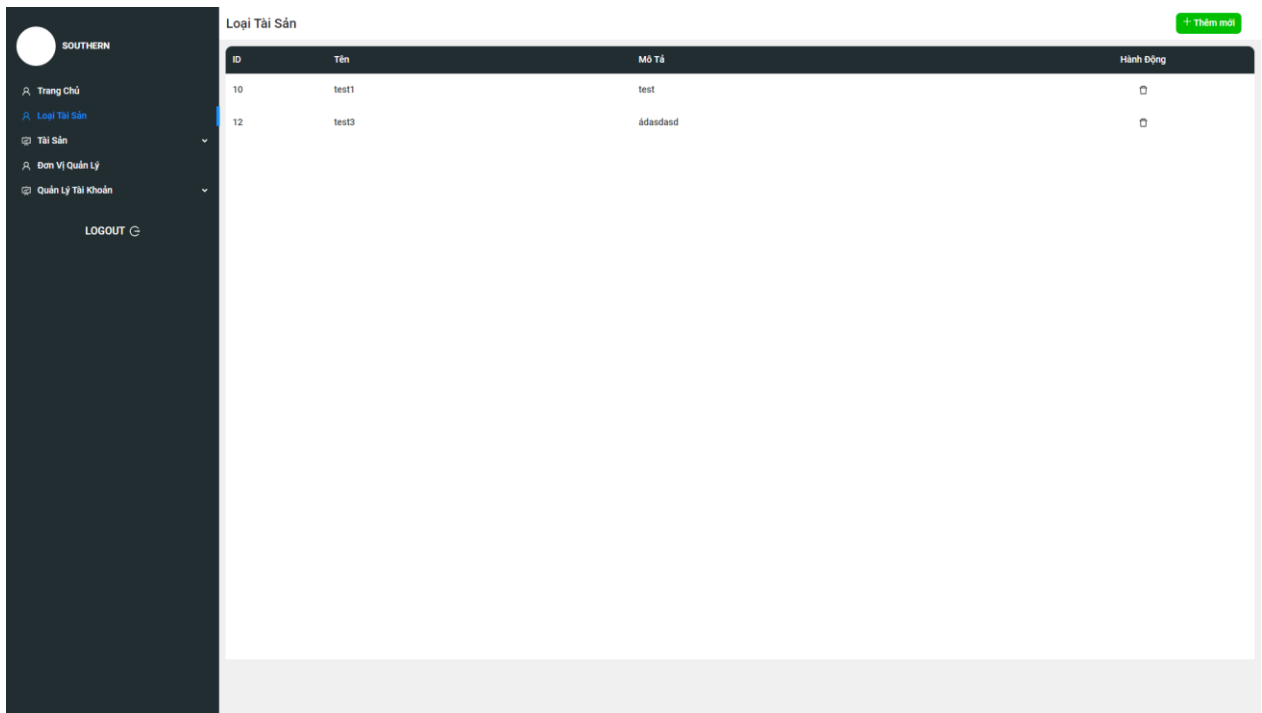
Hình 10: Giao diện trang đăng nhập



3.6.1.2. Giao diện trang chủ:



Hình 11: Giao diện trang chủ

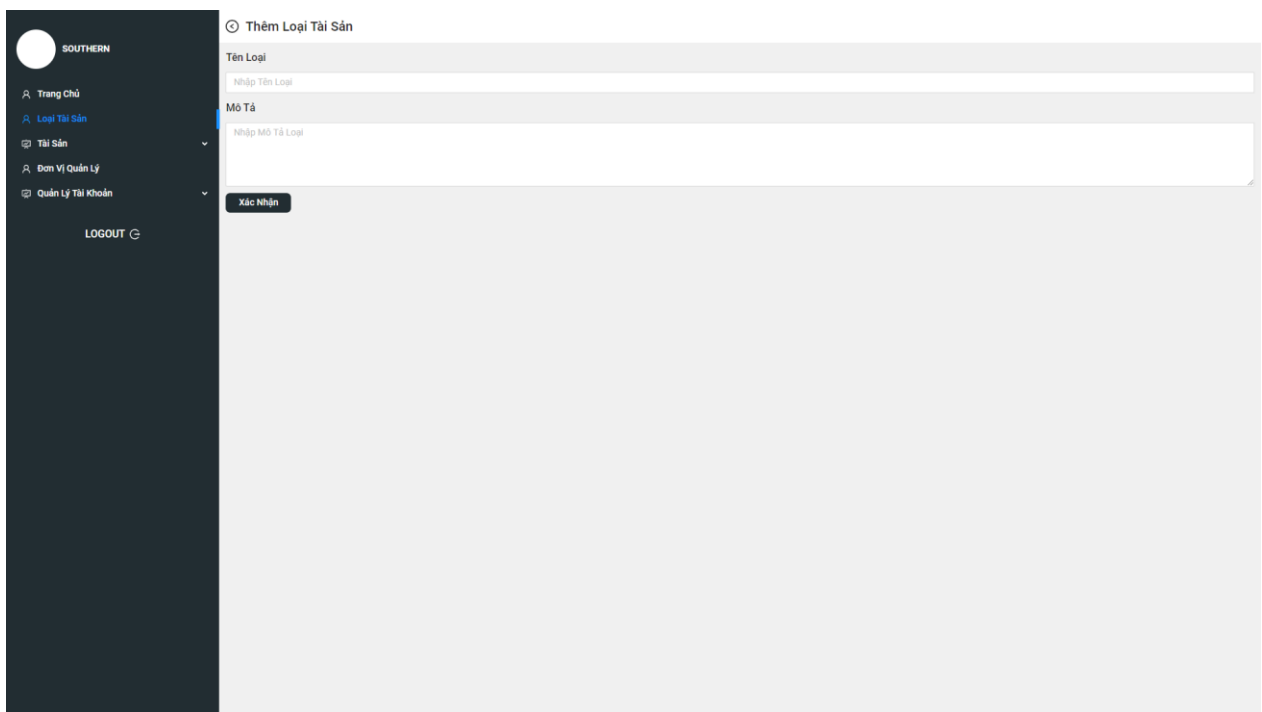
3.6.1.3. Giao diện trang quản lý loại tài sản:



| ID | Tên | Mô Tả | Hành Động |
|----|-------|---------|---|
| 10 | test1 | test |  |
| 12 | test3 | adasdad |  |

Hình 12: Giao diện trang quản lý loại tài sản

3.6.1.4. Giao diện trang thêm loại tài sản:



Thêm Loại Tài Sản

Tên Loại
Nhập Tên Loại

Mô Tả
Nhập Mô Tả Loại

Xác Nhận

Hình 13: : Giao diện trang thêm loại tài sản

3.6.1.5. Giao diện trang sửa loại tài sản:

SOUTHERN

- Trang Chủ
- Loại Tài Sản
- Tài Sản
- Đơn Vị Quản Lý
- Quản Lý Tài Khoản

LOGOUT

Sửa Loại Tài Sản

Tên Loại: test1

Mô Tả: test

Xác Nhận

Hình 14: Giao diện trang sửa loại tài sản

3.6.1.6. Giao diện trang quản lý tài sản:

SOUTHERN

- Trang Chủ
- Loại Tài Sản
- Tài Sản
- Tài Sản Mới
- Tài Sản Sử Dụng
- Đơn Vị Quản Lý
- Quản Lý Tài Khoản

LOGOUT

Quản Lý Tài Sản

+ Thêm mới + Thêm mới bằng Excel

| Tên Tài Sản | QR CODE | Tên Loại | Đơn Vị Tính | Ngày Mua | Hạn Sử Dụng | Giá Tiền | Đơn Vị Qu |
|-------------|---------|----------|-------------|------------|-------------|----------|-----------|
| TEST | | test1 | Cây | 2020-11-25 | 6 | 100.000 | Khoa CNTT |
| asdx | | test1 | Cây | 2020-12-02 | 6 | 12.523 | Khoa CNTT |
| asdx | | test1 | Cây | 2020-12-02 | 6 | 12.523 | Khoa CNTT |
| asdx | | test1 | Cây | 2020-12-02 | 6 | 12.523 | Khoa CNTT |
| asdx | | test1 | Cây | 2020-12-02 | 6 | 12.523 | Khoa CNTT |
| asdx | | test1 | Cây | 2020-12-02 | 6 | 12.523 | Khoa CNTT |
| asdx | | test1 | Cây | 2020-12-02 | 6 | 12.523 | Khoa CNTT |

1 / 2

Hình 15: Giao diện trang quản lý tài sản

3.6.1.7. Giao diện trang thêm tài sản:

The screenshot shows the 'Thêm Tài Sản' (Add Asset) form. On the left is a dark sidebar with the SOUTHERN logo and a menu containing: Trang Chủ, Loại Tài Sản, Tài Sản (highlighted), Tài Cả Tài Sản, Tài Sản Mới, Tài Sản Dừng, Đơn Vị Tính, Đơn Vị Quản Lý, and Quản Lý Tài Khoản. Below the menu is a 'LOGOUT' button. The main content area is titled 'Thêm Tài Sản' and contains several input fields: 'Tên Tài Sản' (with a sub-field 'Nhập Tên Loại'), 'Hạn Sử Dụng' (with a sub-field 'Nhập Hạn Sử Dụng'), 'Loại Tài Sản' (a dropdown menu with 'Chọn Loại Tài Sản'), 'Giá Tiền' (with a sub-field 'Nhập Giá Tiền'), 'Đơn Vị Tính' (a dropdown menu with 'Chọn Đơn Vị Tính'), 'Ngày Mua' (with a sub-field 'Chọn Ngày Mua'), 'Đơn Vị Quản Lý' (a dropdown menu with 'Chọn Đơn Vị Quản Lý'), and 'Mô Tả' (with a sub-field 'Nhập Mô Tả'). A 'Xác Nhận' (Confirm) button is located at the bottom right of the form.

Hình 16: Giao diện trang thêm tài sản

3.6.1.8. Giao diện trang thêm tài bằng excel:

The screenshot shows the 'Thêm Tài Sản Bằng File Excel' (Add Asset by Excel File) form. It features a dark sidebar identical to the one in Figure 16. The main content area is titled 'Thêm Tài Sản Bằng File Excel' and includes a 'Chọn tệp' (Select file) button with a message 'không có tệp nào được chọn' (no files selected). Below this is a 'Xác Nhận' (Confirm) button. The form also displays a table with the following headers: 'Tên Tài Sản', 'Ngày Mua', 'Hạn Sử Dụng', 'Giá Tiền', and 'Mô Tả'. A 'Submit' button is located at the bottom right of the form.

Hình 17: Giao diện trang thêm tài sản bằng file excel

3.6.1.9. Giao diện trang chi tiết tài sản:

SOUTHERN

- Trang Chủ
- Loại Tài Sản
- Tài Sản
- Tất Cả Tài Sản
- Tài Sản Mới
- Tôi Sử Dụng
- Đơn Vị Tính
- Đơn Vị Quản Lý
- Quản Lý Tài Khoản
- LOGOUT

TEST

THÔNG TIN TÀI SẢN

| | | | |
|-------------------|-----------|--------------|--|
| Mã Tài Sản: | 123456 | Đơn Vị Tính: | Cây |
| Tên Tài Sản: | TEST | Giá Tiền: | 100.000 |
| Mã QRCode: | 123456 | Ngày Mua: | 2020-11-25 |
| Tên Loại Tài Sản: | test1 | Hạn Sử Dụng: | 6 Tháng |
| Đơn Vị Quản Lý: | Khoa CNTT | Mô Tả: | Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled |

LỊCH SỬ MƯỢN TÀI SẢN

Hình 18: Giao diện trang chi tiết tài sản

3.6.1.10. Giao diện trang sửa thông tin tài sản:

SOUTHERN

- Trang Chủ
- Loại Tài Sản
- Tài Sản
- Tất Cả Tài Sản
- Tài Sản Mới
- Tôi Sử Dụng
- Đơn Vị Tính
- Đơn Vị Quản Lý
- Quản Lý Tài Khoản
- LOGOUT

Sửa Thông Tin Tài Sản

| | |
|----------------|--|
| Tên Tài Sản | Hạn Sử Dụng |
| TEST | 6 |
| Loại Tài Sản | Giá Tiền |
| test1 | 100000 |
| Đơn Vị Tính | Mô Tả |
| Cây | Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled |
| Ngày Mua | |
| 2020-11-25 | |
| Đơn Vị Quản Lý | |
| Khoa CNTT | |

Xác Nhận

Hình 19: Giao diện trang sửa thông tin tài sản

3.6.1.11. Giao diện trang tài sản mới:

| Tên Tài Sản | QR CODE | Loại | Đơn Vị Tính | Ngày Mua | Hạn Sử Dụng | Giá Tiền | Đơn Vị Quĩ |
|-------------|---------|------|-------------|------------|-------------|----------|------------|
| asdx | | 10 | 1 | 2020-12-02 | 6 | 12.523 | null |

Hình 20: Giao diện trang tài sản mới

3.6.1.12. Giao diện trang đơn vị tính:

| ID | Tên | Hành Động |
|----|-------|-----------|
| 1 | Cây | |
| 2 | Cái | |
| 3 | Thùng | |

Hình 21: Giao diện trang đơn vị tính

3.6.1.13. Giao diện trang thêm mới đơn vị tính:

Thêm Đơn Vị Tính

Tên Đơn Vị Tính

Nhập Tên Đơn Vị Tính

Xác Nhận

Hình 22: Giao diện trang thêm mới đơn vị tính

3.6.1.14. Giao diện trang đơn vị quản lý:

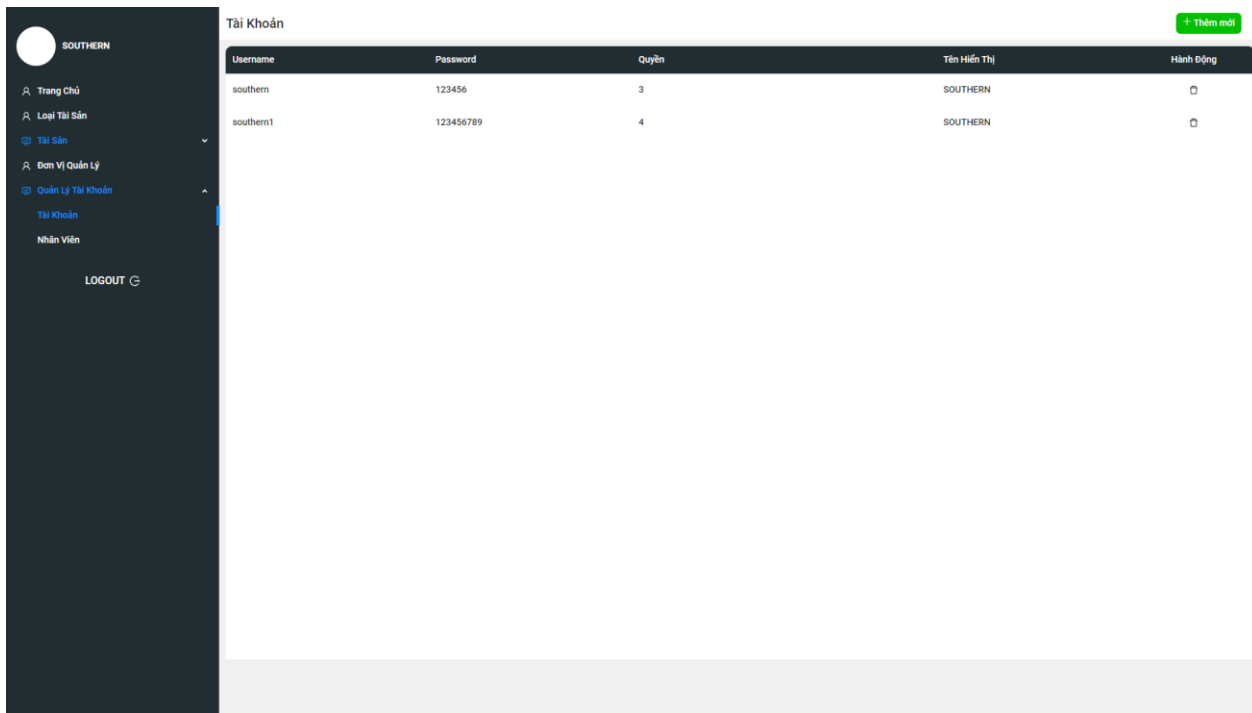
Quản Lý Đơn Vị Quản Lý

+ Thêm mới

| Tên Đơn Vị | Số Điện Thoại | Email | Hành Động |
|------------|---------------|-------------------------|-----------|
| Khoa CNTT | 123456789 | khoacntt@tech@gmail.com | |

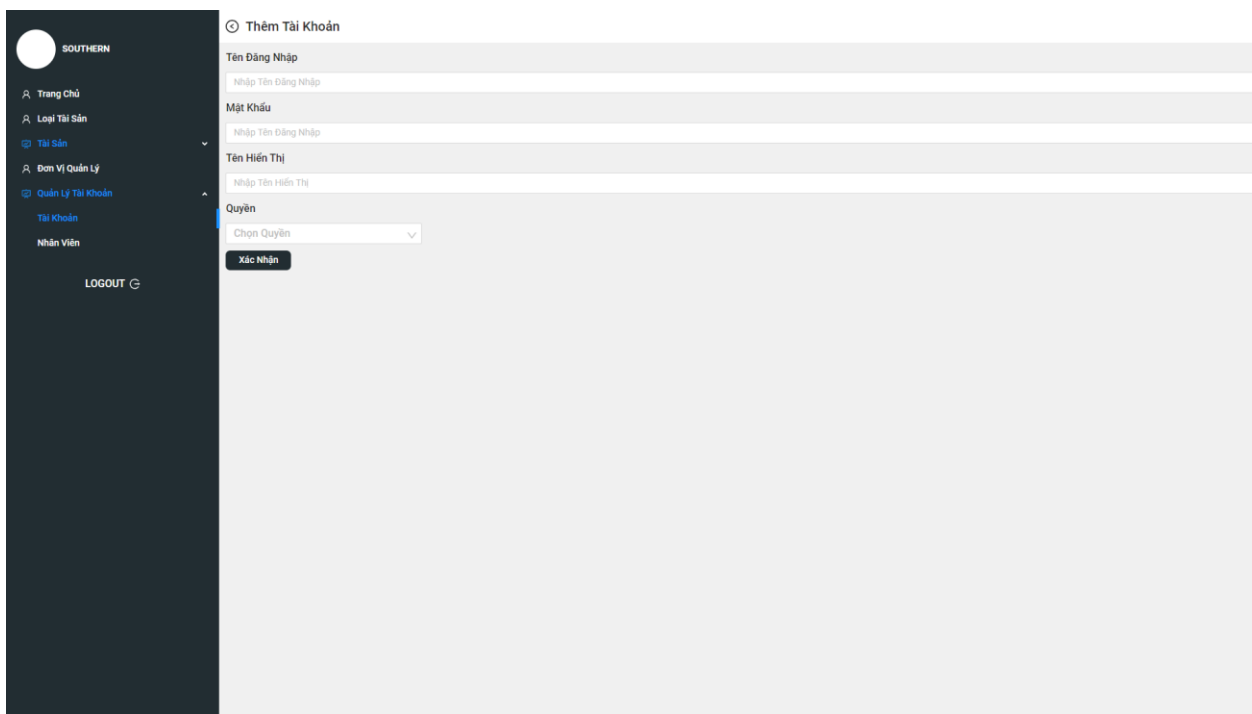
Hình 23: Giao diện trang đơn vị quản lý

3.6.1.15. Giao diện trang quản lý tài khoản:



Hình 24: Giao diện trang quản lý tài khoản

3.6.1.16. Giao diện trang thêm mới tài khoản:



Hình 25: Giao diện trang thêm mới tài khoản

3.6.1.17. Giao diện trang sửa thông tin tài khoản:

SOUTHERN

- Trang Chủ
- Loại Tài Sản
- Tài Sản
- Đơn Vị Quản Lý
- Quản Lý Tài Khoản
 - Tài Khoản
 - Nhân Viên

LOGOUT

Thêm Tài Khoản

Tên Đăng Nhập: southern1

Mật Khẩu: 123456789

Tên Hiển Thị: SOUTHERN

Quyền: nhân Viên

Xác Nhận

Hình 26: Giao diện trang sửa thông tin tài khoản

3.6.1.18. Giao diện trang quản lý nhân viên:

SOUTHERN

- Trang Chủ
- Loại Tài Sản
- Tài Sản
- Đơn Vị Quản Lý
- Quản Lý Tài Khoản
 - Tài Khoản
 - Nhân Viên

LOGOUT

Quản Lý Nhân Viên + Thêm mới

| Tên Nhân Viên | Số Điện Thoại | Địa Chỉ | Mã Tài Khoản | Hành Động |
|---------------|---------------|----------|--------------|-----------|
| a | 123456 | 23123423 | 8 | |

Hình 27: Giao diện trang quản lý nhân viên

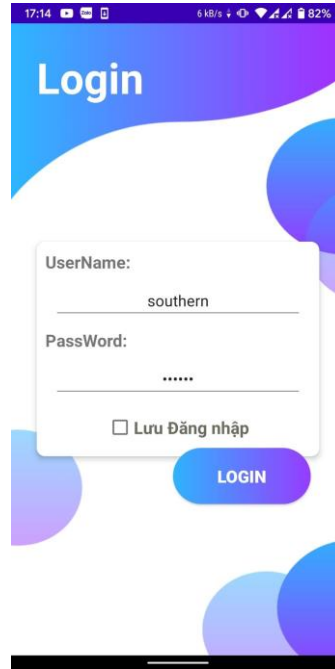
3.6.1.19. Giao diện trang thêm mới nhân viên:

The screenshot displays the 'Thêm Nhân Viên' (Add New Employee) form within the SOUTHERN application. The interface features a dark sidebar on the left with navigation links: Trang Chủ, Loại Tài Sản, Tài Sản, Đơn Vị Quản Lý, Quản Lý Tài Khoản, Tài Khoản, Nhân Viên, and a LOGOUT button. The main content area is titled 'Thêm Nhân Viên' and contains several input fields: 'Tên Nhân Viên' (Employee Name), 'Số Điện Thoại' (Phone Number), and 'Địa Chỉ' (Address). Below these is a 'Quyền' (Rights) section with a dropdown menu labeled 'Chọn Tài Khoản' (Select Account) and a 'Xác Nhận' (Confirm) button. The form is designed with a clean, modern aesthetic using light gray backgrounds and white text.

Hình 28: Giao diện trang thêm mới nhân viên

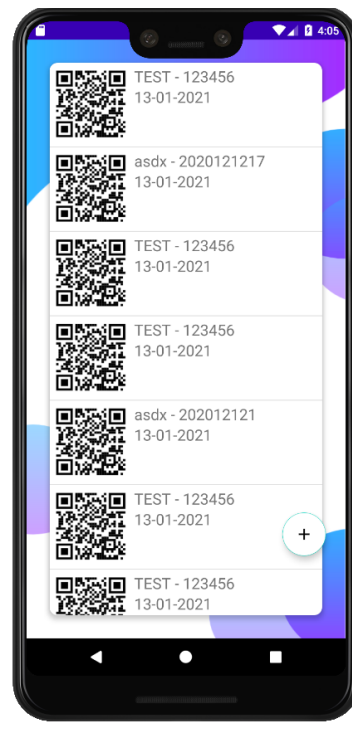
3.6.2. Mô tả giao diện ứng dụng android:

3.6.2.1. Giao diện trang đăng nhập:



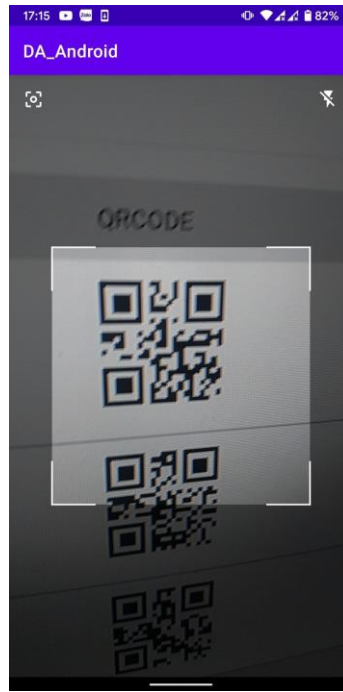
Hình 29: Giao diện trang đăng nhập ứng dụng android

3.6.2.2. Giao diện trang chủ:



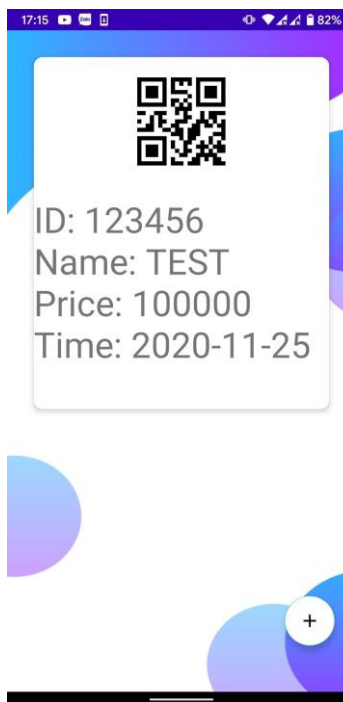
Hình 30: Giao diện trang hiển thị lịch sử quét

3.6.2.3. Giao diện trang quét mã QR CODE:



Hình 31: Giao diện trang quét mã QR CODE

3.6.2.4. Giao diện trang hiển thị chi tiết thông tin tài sản:



Hình 32: Giao diện trang hiển thị chi tiết thông tin tài sản

CHƯƠNG IV: KẾT LUẬN

4.1. Kết quả thực nghiệm:

Đề tài quản lý cơ sở vật chất thông qua QRCODE không phải là đề tài quá mới mẻ, nhưng nó mang lại tính thực tế cao trong quá trình sử dụng, nhất là trong giai đoạn ngày nay, giai đoạn áp dụng những công nghệ để phục vụ đời sống của con người. Dưới sự hướng dẫn của thầy Hoàng Văn Hiếu đã giúp chúng em giải quyết các vấn đề khó khăn trong lúc làm đồ án.

Tuy nhiên với mục tiêu đã đặt ra trước đó em đã hoàn thành các yêu cầu bao gồm:

- Giao diện: thiết kế dễ nhìn, thu hút người dùng.
- Phân quyền quản lý: tùy thuộc vào loại tài khoản mà sẽ có những quyền hạn khác nhau chẳng hạn như nhân viên cashier thì sẽ không vào được trang quản lý sản phẩm.
- Xử lý luồng data một cách hợp lý để tối ưu tốc độ của trang web khi người dùng tra cứu thông tin sản phẩm và quản trị trong việc quản lý hệ thống.
- Phân loại các chức năng và các trang con để dễ quản lý và sử dụng.
- Các quyền bao gồm:
 - Admin: toàn quyền sử dụng các chức năng của hệ thống.
 - Nhân viên: được admin giao cho 1 số các chức năng trong hệ thống.
- Về phần giao diện:
 - Layout đăng nhập
 - Layout quét mã QRCode
 - Layout chi tiết thông tin tài sản
 - Layout hiển thị lịch sử đã quét

4.2. Đánh giá:

4.2.1. Ưu điểm:

- Phần giao diện thân thiện với người dùng.
- Truy xuất dữ liệu từ server nhanh.
- Tài khoản đăng nhập được lưu lại cho lần sau sử dụng.

4.2.2. Nhược điểm:

- Trình bày cấu trúc form còn chưa được hài hoà
- Chưa có phần hỗ trợ cho phần quên mật khẩu.

4.3. Hướng phát triển:

- Bổ sung và hoàn thiện một số chức năng còn thiếu trong phần hệ thống.
- Tăng cường bảo mật cho hệ thống.
- Kiểm soát các lỗi chặt chẽ hơn.
- Thêm các chức năng còn thiếu trong lần này.

TÀI LIỆU THAM KHẢO

- [1]. Viblo [online]. Truy cập khoảng 10 lần/ ngày. Từ: <https://viblo.asia/newest>
- [2]. NodeJS [online]. Truy cập khoảng 5 lần/ ngày. Từ: <https://nodejs.org/en/docs/>
- [3]. ReactJS [online]. Truy cập khoảng 7 lần/ ngày.
Từ: <https://reactjs.org/docs/getting-started.html>
- [4]. React Router [online]. Truy cập khoảng 5 lần/ ngày.
Từ: <https://reacttraining.com/react-router/web/example/basic>
- [5]. W3School [online]. Truy cập khoảng 11 lần/ ngày.
Từ: <https://www.w3schools.com/>
- [6]. MDN web docs [online]. Truy cập khoảng 20 lần/ ngày.
Từ: <https://developer.mozilla.org/en-US/>
- [7]. Lập trình web [offline]. Truy cập khoảng 5 lần/ ngày. Từ: “Giáo trình Lập Trình Web của Đại Học Công Nghệ Thành Phố Hồ Chí Minh (HUTECH)”
- [8]. Công Nghệ Phần Mềm [offline]. Truy cập khoảng 5 lần/ ngày. Từ: “Giáo trình Công Nghệ Phần Mềm của Đại Học Công Nghệ Thành Phố Hồ Chí Minh (HUTECH)”
- [9]. Lập Trình Hướng Đối Tượng [offline]. Truy cập khoảng 5 lần/ ngày. Từ: “Giáo trình Lập Trình Hướng Đối Tượng của Đại Học Công Nghệ Thành Phố Hồ Chí Minh (HUTECH)”
- [10]. Phân tích và thiết kế hướng đối tượng [offline]. Truy cập khoảng 10 lần/ ngày. Từ “Giáo trình Phân tích và thiết kế hướng đối tượng của Đại Học Công Nghệ Thành Phố Hồ Chí Minh (HUTECH)”.