|  |  |
| --- | --- |
| LÊ HƯƠNG GIANG |  |
| **BỘ CÔNG THƯƠNG**  **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**  **---------------------------------** |
| Nhận diện thương hiệu Logo HaUI |
| **ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC**  NGÀNH HỆ THỐNG THÔNG TIN |
| **ĐỀ TÀI: XÂY DỰNG WEBSITE QUẢN LÝ THIẾT BỊ CÔNG NGHỆ CHO DOANH NGHIỆP LDM** |
|  |
| **CBHD: ThS. Nguyễn Đức Lưu** |
| HỆ THỐNG THÔNG TIN | **Sinh viên: Lê Hương Giang** |
| **Mã sinh viên: 2020601404**  **Lớp: 2020DHHTTT01**  **Khóa: K15** |
|  |
|  |
|  |
|  |
| *Hà Nội - Năm 2024* |
|  |

# 

# MỤC LỤC

[MỤC LỤC i](#_Toc167056351)

[DANH MỤC HÌNH ẢNH iv](#_Toc167056352)

[LỜI CẢM ƠN vi](#_Toc167056353)

[MÔ TẢ TÓM TẮT ĐỀ TÀI vii](#_Toc167056354)

[MỞ ĐẦU 1](#_Toc167056355)

[CHƯƠNG 1. TỔNG QUAN VỀ CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ SỬ DỤNG 4](#_Toc167056356)

[1.1 Laravel PHP framework và mô hình MVC 4](#_Toc167056357)

[1.1.1 Giới thiệu về Laravel PHP Framework 4](#_Toc167056358)

[1.1.2 Lịch sử phát triển của Laravel PHP Framework 4](#_Toc167056359)

[1.1.3 Giới thiệu mô hình MVC 7](#_Toc167056360)

[1.1.4 Lợi ích khi xây dựng ứng dụng web dựa trên mô hình MVC 8](#_Toc167056361)

[1.2 MySQL 9](#_Toc167056362)

[1.2.1 Giới thiệu về MySQL 9](#_Toc167056363)

[1.2.2 Ưu điểm của MySQL 10](#_Toc167056364)

[1.3 PhpMyAdmin 10](#_Toc167056365)

[1.3.1 Các tính năng của phpMyAdmin 11](#_Toc167056366)

[1.3.2 Ưu điểm của phpMyAdmin 11](#_Toc167056367)

[1.4 Restful API 12](#_Toc167056368)

[1.4.1 Giới thiệu về Resful API 12](#_Toc167056369)

[1.4.2 Các thành phần của Restful API 13](#_Toc167056370)

[1.5 Một số công cụ sử dụng 13](#_Toc167056371)

[1.5.1 Github 13](#_Toc167056372)

[1.5.2 Case Studio 14](#_Toc167056373)

[1.5.3 Rational Rose 14](#_Toc167056374)

[1.5.4 Visual Studio Code 14](#_Toc167056375)

[1.5.5 Laragon 17](#_Toc167056376)

[CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG VÀ THIẾT KẾ CƠ SỞ DỮ LIỆU 18](#_Toc167056377)

[2.1 Tổng quan hệ thống 18](#_Toc167056378)

[2.2 Xác định các yêu cầu chính của hệ thống 19](#_Toc167056379)

[2.2.1 Phân tích các yêu cầu 19](#_Toc167056380)

[2.2.1.1 Các yêu cầu chức năng 19](#_Toc167056381)

[2.2.1.2 Các yêu cầu phi chức năng 20](#_Toc167056382)

[2.2.2 Xây dựng và phân tích các biểu đồ use case 20](#_Toc167056383)

[2.2.2.1 Các tác nhân 20](#_Toc167056384)

[2.2.2.2 Các use case của hệ thống 20](#_Toc167056385)

[2.2.2.3 Biểu đồ use case 22](#_Toc167056386)

[2.2.3 Phân tích use case 23](#_Toc167056387)

[2.2.3.1 Use case đăng nhập 23](#_Toc167056388)

[2.2.3.2 Use case quên mật khẩu 26](#_Toc167056389)

[2.2.3.3 Use case cấp thiết bị 30](#_Toc167056390)

[2.2.3.4 Use case xác nhận thiết bị đã sửa xong 34](#_Toc167056391)

[2.2.3.5 Use case xem thông sửa chữa hoặc bảo hành của thiết bị 37](#_Toc167056392)

[2.2.4 Xây dựng cơ cở dữ liệu 41](#_Toc167056393)

[2.2.4.1 Biểu đồ thực thể liên kết (Entity Relationship Diagram) 41](#_Toc167056394)

[2.2.4.2 Các bảng trong cơ sở dữ liệu 42](#_Toc167056395)

[CHƯƠNG 3. KẾT QUẢ THỬ NGHIỆM 46](#_Toc167056396)

[3.1 Giao diện chương trình 46](#_Toc167056397)

[3.1.1 Màn hình đăng nhập 46](#_Toc167056398)

[3.1.2 Màn hình trang chủ 46](#_Toc167056399)

[3.1.3 Màn hình danh sách thiết bị 47](#_Toc167056400)

[3.1.4 Màn hình thêm mới thiết bị 47](#_Toc167056401)

[3.1.5 Màn hình danh sách thiết bị đang cho mượn 48](#_Toc167056402)

[3.1.6 Màn hình danh sách thiết bị đang hỏng 48](#_Toc167056403)

[3.1.7 Màn hình danh sách thiết bị đã sửa chữa hoặc bảo hành 49](#_Toc167056404)

[3.1.8 Màn hình cấp thiết bị 49](#_Toc167056405)

[3.1.9 Màn hình quên mật khẩu 50](#_Toc167056406)

[CHƯƠNG 4. KIỂM THỬ WEBSITE 51](#_Toc167056407)

[4.1 Thiết kế và tạo testcase 51](#_Toc167056408)

[4.2 Thực hiện kiểm thử 51](#_Toc167056409)

[4.3 Kĩ thuật kiểm thử 51](#_Toc167056410)

[4.4 Kiểm thử giao diện hệ thống 52](#_Toc167056411)

[KẾT LUẬN 63](#_Toc167056412)

[TÀI LIỆU THAM KHẢO 64](#_Toc167056413)

# DANH MỤC HÌNH ẢNH

[Hình 1. 1 Laravel framework PHP 4](#_Toc167056414)

[Hình 1. 2 Mô hình MVC 7](#_Toc167056415)

[Hình 1. 3 MySQL 10](#_Toc167056416)

[Hình 1. 4 Res tAPI 12](#_Toc167056417)

[Hình 2. 1 Biểu đồ use case tổng quát 22](#_Toc167056418)

[Hình 2. 2 Biểu đồ phân rã use case quản lý thiết bị 22](#_Toc167056419)

[Hình 2. 3 Biểu đồ lớp tham gia ca sử dụng use case Đăng nhập 24](#_Toc167056420)

[Hình 2. 4 Biểu đồ trình tự use case Đăng nhập 25](#_Toc167056421)

[Hình 2. 5 Biểu đồ hoạt động use case đăng nhập 26](#_Toc167056422)

[Hình 2. 6 Biểu đồ lớp tham gia ca sử dụng use case quên mật khẩu 28](#_Toc167056423)

[Hình 2. 7 Biểu đồ trình tự use case quên mật khẩu 29](#_Toc167056424)

[Hình 2. 8 Biểu đồ hoạt động use case quên mật khẩu 30](#_Toc167056425)

[Hình 2. 9 Biểu đồ lớp tham gia ca sử dụng use case cấp thiết bị 32](#_Toc167056426)

[Hình 2. 10 Biểu đồ trình tự use case cấp thiết bị 33](#_Toc167056427)

[Hình 2. 11 Biểu đồ hoạt động use case cấp thiết bị 34](#_Toc167056428)

[Hình 2. 12 Biểu đồ lớp tham gia ca sử dụng use case xác nhận thiết bị đã sửa xong 36](#_Toc167056429)

[Hình 2. 13 Biểu đồ trình tự use case xác nhận thiết bị đã sửa xong 36](#_Toc167056430)

[Hình 2. 14 Biểu đồ hoạt động use case xác nhận thiết bị đã sửa xong 37](#_Toc167056431)

[Hình 2. 15 Biểu đồ lớp tham gia ca sử dụng use case xem thông sửa chữa hoặc bảo hành của thiết bị 39](#_Toc167056432)

[Hình 2. 16 Biểu đồ trình tự use case xem thông sửa chữa hoặc bảo hành của thiết bị 40](#_Toc167056433)

[Hình 2. 17 Biểu đồ hoạt động use case xem thông sửa chữa hoặc bảo hành của thiết bị 41](#_Toc167056434)

[Hình 2. 18 Biểu đồ ERD thiết kế cơ sở dữ liệu 41](#_Toc167056435)

[Hình 2. 19 Bảng users 42](#_Toc167056436)

[Hình 2. 20 Bảng departments 42](#_Toc167056437)

[Hình 2. 21 Bảng categories 43](#_Toc167056438)

[Hình 2. 22 Bảng requests 43](#_Toc167056439)

[Hình 2. 23 Bảng devices 44](#_Toc167056440)

[Hình 2. 24 24 Bảng repairs 44](#_Toc167056441)

[Hình 2. 25 Bảng repair\_details 45](#_Toc167056442)

[Hình 3. 1 Màn hình đăng nhập 46](#_Toc167056459)

[Hình 3. 2 Màn hình trang chủ 47](#_Toc167056460)

[Hình 3. 3 Màn hình danh sách thiết bị 47](#_Toc167056461)

[Hình 3. 4 Màn hình thêm mới thiết bị 48](#_Toc167056462)

[Hình 3. 5 Màn hình danh sách thiết bị đang cho mượn 48](#_Toc167056463)

[Hình 3. 6 Màn hình danh sách thiết bị đang hỏng 49](#_Toc167056464)

[Hình 3. 7 Màn hình danh sách thiết bị đã sửa chữa hoặc bảo hành 49](#_Toc167056465)

[Hình 3. 8 Màn hình cấp thiết bị 50](#_Toc167056466)

[Hình 3. 9 Màn hình gửi link xác nhận đặt lại mật khẩu 50](#_Toc167056467)

# LỜI CẢM ƠN

*Đề tài “Xây dựng Website quản lý thiết bị công nghệ cho doanh nghiệp LDM” là đồ án tốt nghiệp của em.*

*Trong quá trình nghiên cứu và hoàn thiện đồ án, em đã nhận được nhiều sự quan tâm, giúp đỡ từ quý thầy cô, gia đình và bạn bè.*

*Trước tiên, em xin cảm ơn Khoa Công nghệ thông tin Trường đại học công nghiệp Hà Nội, lãnh đạo và lời tri ân sâu sắc nhất tới thầy ThS. Nguyễn Đức Lưu đã luôn giúp đỡ, hướng dẫn, chỉ bảo tận tình để em hoàn thành tốt đồ án của mình.*

*Em cũng xin cảm ơn những người thân, bạn bè và thầy cô giảng dạy đã động viên và hỗ trợ em hoàn thành đồ án này.*

*Em xin chân thành cảm ơn!*

*Sinh viên thực hiện*

*Lê Hương Giang*

# MÔ TẢ TÓM TẮT ĐỀ TÀI

**Họ và tên** : Lê Hương Giang

**Mã sinh viên**: 2020601404

**Đề tài** : Xây dựng Website quản lý thiết bị công nghệ cho doanh nghiệp LDM.

**Tóm tắt đề tài:**

***Giới thiệu đề tài:*** Hiện nay, dưới sự bùng nổ của công nghệ thông tin, các doanh nghiệp ngày càng nhiều kéo theo đó là nhu cầu quản lý các thiết bị công nghệ tăng cao. Để thuận tiện cho việc quản lý các thiết bị một cách hiệu quả và đáng tin cậy. Vậy nên, em chọn đề tài ***“Xây dựng Website quản lý thiết bị công nghệ cho doanh nghiệp LDM”.***

***Mục tiêu đề tài:***

* Nắm được kiến thức về công nghệ sử dụng.
* Phân tích, hiểu nghiệp vụ thực tế của Website quản lý thiết bị công nghệ.
* Xây dựng được một trang web quản lý thiết bị chuyên nghiệp, có khả năng giúp công ty quản lý tài sản thiết bị hiệu quả hơn, tăng tính đáng tin cậy và hiệu suất của hệ thống quản lý.
* Cài đặt và triển khai Website quản lý thiết bị công nghệ.

***Kết quả đạt được:***

* Xây dựng được các chức năng của Website.
* Hoàn thành báo cáo, cài đặt và triển khai Website quản lý thiết bị công nghệ*.*

**TOPIC SUMMARY DESCRIPTION**

**Full name:** Le Huong Giang

**Student code:** 2020601404

**Topic:** Building a technology equipment management website for LDM businesses.

**Topic summary:**

***Topic introduction*:** Currently, under the explosion of information technology, there are more and more businesses, leading to an increased need to manage technological devices. To facilitate efficient and reliable management of devices. Therefore, I chose the topic "**Building a website to manage technology equipment for LDM businesses**".

***Subject objective:***

- Grasp knowledge of used technology.

- Analyze and understand the actual operations of the technology equipment management website.

- Build a professional equipment management website, capable of helping the company manage equipment assets more effectively, increasing the reliability and performance of the management system

- Install and deploy technology equipment management website.

***Result:***

- Build website functions.

- Complete reports, install and deploy technology equipment management website.

# MỞ ĐẦU

1. **Lý do chọn đề tài**

Hiện nay công nghệ thông tin không ngừng phát triển mạnh mẽ và hiện đại, các công ty lập trình, thiết kế gia công phần mềm ngày càng nhiều. Các công ty này sử dụng nhiều thiết bị như máy tính cây, laptop, máy chủ, thiết bị mạng và nhiều thiết bị khác để thực hiện công việc của mình. Điều này đặt ra nhu cầu quản lý các thiết bị này một cách hiệu quả và đáng tin cậy.

Dưới đây là một số lý do tại sao doanh nghiệp cần một website quản lý thiết bị:

* Quản lý thiết bị dễ dàng: trang web quản lý thiết bị giúp công ty lập trình quản lý các thiết bị dễ dàng hơn. Nó cho phép nhân viên quản lý thông tin về thiết bị như thông số kỹ thuật, trạng thái và lịch sử bảo trì. Điều này giúp nhân viên nắm bắt được thông tin về trạng thái của thiết bị và lên kế hoạch bảo trì, sửa chữa kịp thời.
* Tăng tính đáng tin cậy của hệ thống: trang web quản lý thiết bị giúp công ty lập trình đưa ra quyết định kịp thời và giải quyết sự cố nhanh chóng. Điều này tăng tính đáng tin cậy của hệ thống và giảm thiểu tác động tiêu cực đến hoạt động của công ty.
* Nâng cao hiệu suất của nhân viên: trang web quản lý thiết bị cho phép nhân viên truy cập thông tin về thiết bị một cách dễ dàng. Điều này giúp nhân viên thực hiện công việc của mình một cách hiệu quả hơn và tiết kiệm thời gian.
* Tăng tính đồng bộ trong công ty: trang web quản lý thiết bị giúp tạo ra một kênh giao tiếp hiệu quả giữa các bộ phận trong công ty. Điều này tăng tính đồng bộ trong công ty và giúp cải thiện quy trình làm việc.
* Giảm chi phí: trang web quản lý thiết bị giúp giảm chi phí cho công ty lập trình. Bằng cách quản lý và bảo trì thiết bị một cách đúng đắn, công ty có thể giảm thiểu chi phí bảo trì và sửa chữa thiết bị.
* Dễ dàng mở rộng quy mô hoạt động: khi công ty lập trình mở rộng quy mô hoạt động, việc quản lý thiết bị càng trở nên phức tạp hơn. Trang web quản lý thiết bị giúp công ty dễ dàng mở rộng quy mô hoạt động mà không gặp khó khăn trong việc quản lý thiết bị.
* Quản lý thiết bị từ xa: nếu công ty lập trình có nhiều chi nhánh, trang web quản lý thiết bị giúp quản lý thiết bị từ xa. Nhân viên có thể truy cập vào trang web từ bất kỳ đâu và kiểm tra trạng thái của thiết bị một cách dễ dàng.
* Thống kê báo cáo: trang web quản lý thiết bị cung cấp các báo cáo thống kê cho công ty lập trình, giúp công ty phân tích và đưa ra quyết định dựa trên dữ liệu. Thông tin về trạng thái của thiết bị và lịch sử bảo trì sẽ được phân tích để đưa ra kế hoạch phù hợp.

Tóm lại, việc thiết kế một trang web quản lý thiết bị cho doanh nghiệp là rất quan trọng và đáng được đầu tư. Nó giúp công ty quản lý thiết bị một cách hiệu quả hơn, tăng tính đáng tin cậy của hệ thống, nâng cao hiệu suất của nhân viên, tăng tính đồng bộ và giảm chi phí. Với mong muốn củng cố và nâng cao kiến thức về lập trình Website, áp dụng kiến thức để xây dựng một hệ thống giúp người dùng có thể dễ dàng sử dụng, sử dụng hệ thống với công nghệ mới nhanh và mượt mà nên em đã chọn đề tài “**Xây dựng Website quản lý thiết bị công nghệ cho doanh nghiệp LDM**”.

**2. Mục tiêu của đề tài**

Xây dựng được một trang web quản lý thiết bị chuyên nghiệp, có khả năng giúp công ty quản lý tài sản thiết bị hiệu quả hơn, tăng tính đáng tin cậy và hiệu suất của hệ thống quản lý, cải thiện quy trình làm việc của nhân viên và giảm thiểu các sai sót không đáng có trong quá trình quản lý thiết bị.

**3. Nội dung nghiên cứu**

* Khảo sát và phân tích yêu cầu quy trình quản lý thiết bị của công ty IT.
* Phân tích thiết kế hệ thống.
* Thiết kế cơ sở dữ liệu.
* Lập trình backend API, MVC.
* Lập trình giao diện theo mô hình MVC kết hợp sử dụng API.

**4.** **Bố cục đề tài**

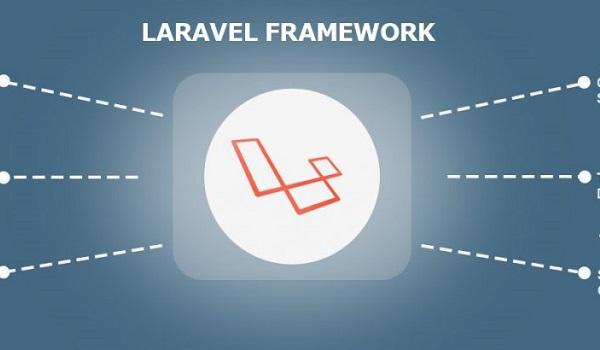
* **Chương 1**: Cơ sở lý thuyết và công nghệ sử dụng
* **Chương 2**: Phân tích thiết kế hệ thống và thiết kế cơ sở dữ liệu.
* **Chương 3**: Kết quả thử nghiệm
* **Chương 4**: Kiểm thử Website

# CHƯƠNG 1. TỔNG QUAN VỀ CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ SỬ DỤNG

## 1.1 Laravel PHP framework và mô hình MVC

### 1.1.1 Giới thiệu về Laravel PHP Framework

**Laravel** là một PHP framework, có mã nguồn mở và miễn phí, được xây dựng nhằm hỗ trợ phát triển các phần mềm, ứng dụng, theo kiến trúc MVC**.**Hiện nay, Laravel đang là PHP framework phổ biến nhất và tốt nhất.



Hình 1. Laravel framework PHP

Những lý do khiến Laravel trở nên rộng rãi:

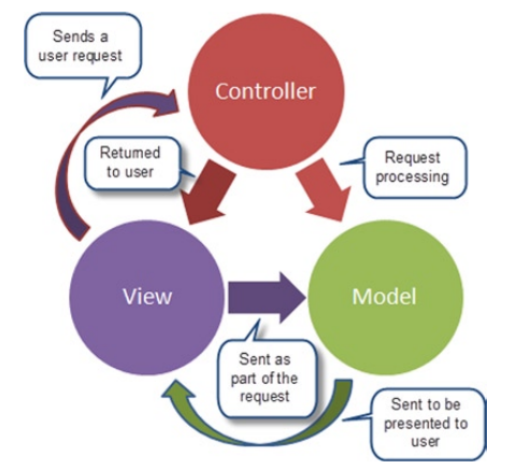
* Cú pháp dễ hiểu – rõ ràng
* Hệ thống đóng gói modular và quản lý gói phụ thuộc
* Nhiều cách khác nhau để truy cập vào các cơ sở dữ liệu quan hệ
* Nhiều tiện ích khác nhau hỗ trợ việc triển khai vào bảo trì ứng dụng.

### 1.1.2 Lịch sử phát triển của Laravel PHP Framework

Bản Laravel beta đầu tiên được phát hành vào ngày 9/6/2011 do Taylor Otwell tạo ra như một giải pháp thay thế cho **CodeIgniter**, tiếp đó là:

* **Laravel 1:** phát hành trong cùng tháng. Laravel 1 bao gồm các tính năng như xác thực, bản địa hóa, model, view, session, định tuyến và các cơ cấu khác, nhưng vẫn còn thiếu controller, điều này làm nó chưa thật sự là một MVC framework đúng nghĩa.
* **Laravel 2:** được phát hành vào tháng 9 năm 2011, mang đến nhiều cài tiến từ tác giả và cộng đồng. Tính năng đáng kể bao gồm hỗ trợ controller, điều này thực sự biến Laravel 2 thành một MVC framework hoàn chỉnh, hỗ trợ Inversion of Control ([IoC](https://en.wikipedia.org/wiki/Inversion_of_control)), hệ thống template Blade. Bên cạnh đó, có một nhược điểm là hỗ trợ cho các gói của nhà phát triển bên thứ 3 bị gỡ bỏ.
* **Laravel 3:** được phát hành vào tháng 2 năm 2012, với một tấn tính năng mới bao gồm giao diện dòng lệnh (CLI) tên “Artisan”, hỗ trợ nhiều hơn cho hệ thống quản trị cơ sở dữ liệu, chức năng ánh xạ cơ sở dữ liệu Migration, hỗ trợ “bắt sự kiện” trong ứng dụng,  và hệ thống quản lý gói gọi là “Bundles”. Lượng người dùng và sự phổ biến tăng trưởng mạnh kể từ phiên bản Laravel 3.
* **Laravel 4:**, tên mã “Illuminate”, được phát hành vào tháng 5 năm 2013. Lần này thực sự là sự lột xác của Laravel framework, di chuyển và tái cấu trúc các gói hỗ trợ vào một tập được phân phối thông qua Composer, một chương trình quản lý gói thư viện phụ thuộc độc lập của PHP. Bố trí mới như vậy giúp khả năng mở rộng của Laravel 4 tốt hơn nhiều so với các phiên bản trước. Ra mắt lịch phát hành chính thức mỗi sáu tháng một phiên bản nâng cấp nhỏ. các tính năng khác trong Laravel 4 bao gồm tạo và thêm dữ liệu mẫu (database seeding), hỗ trợ hàng đợi, các kiểu gởi mail, và hỗ trợ “xóa mềm”  (soft-delete: record bị lọc khỏi các truy vấn từ Eloquent mà không thực sự xóa hẳn khỏi DB).
* **Laravel 5:** được phát hành trong tháng 2 năm 2015, như một kết quả thay đổi đáng kể cho việc kết thúc vòng đời nâng cấp Laravel lên 4.3. Bên cạnh một loạt tính năng mới và các cải tiến như hiện tại, Laravel 5 cũng giới thiệu cấu trúc cây thư mục nội bộ cho phát triển ứng dụng mới. Những tính năng mới của Laravel 5 bao gồm hộ trợ lập lịch định kỳ thực hiện nhiệm vụ thông qua một gói tên là “Scheduler”, một lớp trừu tượng gọi là “Flysystem” cho phép điều khiển việc lưu trữ từ xa đơn giản như lưu trữ trên máy local – dễ thấy nhất là mặc định hỗ trợ dịch vụ Amazone S3, cải tiến quản lý assets thông qua “Elixir”, cũng như đơn giản hóa quản lý xác thực với các dịch vụ bên ngoài bằng gói “Socialite”.
* **Laravel 5.1:** phát hành vào tháng 6 năm 2015, là bản phát hành đầu tiên nhận được hỗ trợ dài hạng (LTS) với một kết hoạch fix bug lên tới 2 năm vào hỗ trợ vá lỗi bảo mật lên tới 3 năm. Các bản phát hành LTS của Laravel được lên kế hoạch theo mỗi 2 năm.
* **Laravel 5.3**: được phát hành vào ngày 23 tháng 8 năm 2016. Các tính năng mới trong 5.3 tập trung vào việc cải thiện tốc độ phát triển bằng cách bổ sung thêm các cải tiến cho các tác vụ phổ biến.
* **Laravel 5.4:** Phiên bản này có nhiều tính năng mới, như Laravel Dusk, Laravel Mix, Blade Components và Slots, Markdown Emails, Automatic Facades, Route Improvements, Higher Order Messaging cho Collections, và nhiều thứ khác.
* **Laravel 5.5**: phát hành vào ngày 30 tháng 8 năm 2017 là phiên bản LTS thứ 2
* **Laravel 5.6**: phát hành vào ngày 7 tháng 2 năm 2018.
* **Laravel 5.7**: phát hành vào ngày 4 tháng 9 năm 2018 với những cập nhật.
* Cải thiện thông báo lỗi
* Callable Action URLs
* Email Verification
* Bổ sung phương thức mới cho tùy chỉnh phân trang
* Thay đổi cấu trúc thư mục Resource
* **Laravel 6:** phát hành vào ngày 3 tháng 9 năm 2019. Đây là version LTS.
* Đổi versioning scheme sang Semantic Versioning
* Cải thiện Exceptions thông qua Ignition
* Cải thiện Authorization Responses
* Job Middleware
* Lazy Collections
* Cải tiến Eloquent subquery
* **Laravel 7:**ra mắt ngày 3 tháng 3 năm 2020 cung cấp nhiều tính năng mới như Laravel Airlock, HTTP Client, Route Caching...
* **Laravel 8:** Ra mắt vào tháng 9 năm 2020, Laravel 8 cung cấp tính năng Jetstream, Laravel Breeze, Migration Squashing và nhiều tính năng khác để tăng tốc độ phát triển ứng dụng web.

### 1.1.3 Giới thiệu mô hình MVC



Hình 1. Mô hình MVC

Các đặc trưng chính của Model, View và Controller trong mẫu thiết kế chuẩn MVC.

* **Model:** đại diện cho hình dạng của dữ liệu. Nó duy trì dữ liệu của ứng dụng. Các đối tượng model lấy và lưu trữ trạng thái trong cơ sở dữ liệu.
* **View:** là giao diện người dùng. View hiển thị dữ liệu của model cho người dùng và cũng cho phép họ sửa đổi dữ liệu.
* **Controller:** xử lý yêu cầu của người dùng. Thông thường, người dùng tương tác với view, những tương tác này sẽ tạo ra yêu cầu tương ứng và sẽ được xử lý bởi controller. Controller trả về view thích hợp kèm theo dữ liệu model dưới dạng phản hồi.

### 1.1.4 Lợi ích khi xây dựng ứng dụng web dựa trên mô hình MVC

Nền tảng Laravel framework PHP theo mô hình MVC mang lại những lợi ích sau:

* Dễ dàng quản lý sự phức tạp của ứng dụng bằng cách chia ứng dụng thành ba thành phần model, view, controller.
* Hỗ trợ tốt cho các ứng dụng được xây dựng bởi những đội có nhiều lập trình viên và thiết kế mà vẫn quản lý được tính năng của ứng dụng.
* Sử dụng các tính năng mới nhất của PHP:Sử dụng Laravel 8 giúp các lập trình viên tiếp cận những tính năng mới nhất mà PHP cung cấp, nhất là đối với Namespaces, Interfaces, Overloading, Anonymous functions và Shorter array syntax.
* Nguồn tài nguyên vô cùng lớn và sẵn có:Nguồn tài nguyên của**Laravel**rất thân thiện với các lập trình viên với đa dạng tài liệu khác nhau để tham khảo. Các phiên bản được phát hành đều có nguồn tài liệu phù hợp với ứng dụng của mình.
* Tích hợp với dịch vụ mail:Lavarel là framework được trang bị API sạch trên thư viện **SwiftMailer**, do đó, bạn có thể gửi thư qua các dịch vụ dựa trên nền tảng đám mây hoặc local.
* Tốc độ xử lý nhanh:Laravel hỗ trợ hiệu quả cho việc tạo lập website hay các dự án lớn trong thời gian ngắn. Vì vậy, nó được các công ty công nghệ và lập trình viên sử dụng rộng rãi để phát triển các sản phẩm của họ.
* Dễ sử dụng:**Laravel**được đón nhận và trở nên sử dụng phổ biến vì nó rất dễ sử dụng. Thường chỉ mất vài giờ, bạn có thể bắt đầu thực hiện một dự án nhỏ với vốn [kiến thức](https://wiki.matbao.net/kien-thuc/) cơ bản nhất về lập trình với PHP.
* Tính bảo mật cao*:* Laravel cung cấp sẵn cho người dùng các tính năng bảo mật mạnh mẽ để người dùng hoàn toàn tập trung vào việc phát triển sản phẩm của mình

## 1.2 MySQL

### 1.2.1 Giới thiệu về MySQL

**MySQL** là hệ quản trị cơ sở dữ liệu tự do nguồn mở phổ biến nhất thế giới và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng. Vì MySQL là hệ quản trị cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh. Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng có truy cập CSDL trên internet. Người dùng có thể tải về MySQL miễn phí từ trang chủ. MySQL có nhiều phiên bản cho các hệ điều hành khác nhau: phiên bản Win32 cho các hệ điều hành dòng Windows, Linux, Mac OS X, Unix, FreeBSD, NetBSD, Novell NetWare, SGI Irix, Solaris, SunOS,..  
MySQL là một trong những ví dụ rất cơ bản về Hệ Quản trị Cơ sở dữ liệu quan hệ sử dụng Ngôn ngữ truy vấn có cấu trúc (SQL).  
MySQL được sử dụng cho việc bổ trợ NodeJs, PHP, Perl, và nhiều ngôn ngữ khác, làm nơi lưu trữ những thông tin trên các trang web viết bằng NodeJs, PHP hay Perl,...Lịch sử phát triển

### 1.2.2 Ưu điểm của MySQL



Hình 1. MySQL

* **Sự ổn định:** MySQL là một hệ quản trị cơ sở dữ liệu rất ổn định và được sử dụng rộng rãi trong các ứng dụng web và doanh nghiệp lớn.
* **Tốc độ và hiệu suất:** MySQL được xây dựng để xử lý lượng dữ liệu lớn với tốc độ xử lý nhanh, đồng thời hỗ trợ nhiều ngôn ngữ lập trình khác nhau.
* **Độ tin cậy cao:** MySQL hỗ trợ nhiều tính năng đảm bảo tính toàn vẹn dữ liệu, bao gồm kiểm soát truy cập, bảo mật và sao lưu dữ liệu.
* **Khả năng mở rộng:** MySQL có khả năng mở rộng tốt với khả năng xử lý dữ liệu lớn và cung cấp tính năng nhân rộng (scalability) cho phép tăng hiệu suất và khả năng lưu trữ khi cần thiết.
* **Phát triển mạnh mẽ:** MySQL được cộng đồng mã nguồn mở đông đảo phát triển và bảo trì, đồng thời cung cấp các công cụ và thư viện phát triển mạnh mẽ cho phép phát triển ứng dụng nhanh chóng và tiện lợi.
* **Tính khả dụng cao:** MySQL có sẵn trên nhiều nền tảng và hệ điều hành khác nhau, đồng thời có thể tích hợp với các công cụ phát triển phổ biến như PHP, Java, Python, .NET, và nhiều ngôn ngữ lập trình khác.

## 1.3 PhpMyAdmin

PhpMyAdmin là phần mềm mã nguồn mở được viết bằng ngôn ngữ PHP giúp quản trị cở sở dữ liệu MySQL thông qua giao diện web. Tính đến nay, phpMyAdmin đã có đến hàng triệu lượt sử dụng và vẫn không ngừng tăng.

MySQL là một hệ thống quản trị cơ sở dữ liệu mã nguồn mở (gọi tắt là RDBMS) hoạt động theo mô hình client-server. Với RDBMS là viết tắt của Relational Database Management System. MySQL được tích hợp apache, PHP. MySQL quản lý dữ liệu thông qua các cơ sở dữ liệu. Mỗi cơ sở dữ liệu có thể có nhiều bảng quan hệ chứa dữ liệu. MySQL cũng có cùng một cách truy xuất và mã lệnh tương tự với ngôn ngữ SQL. MySQL được phát hành từ thập niên 90s.

### 1.3.1 Các tính năng của phpMyAdmin

Quản lý cơ sở dữ liệu: PhpMyAdmin cho phép quản lý cơ sở dữ liệu của bạn bằng cách tạo và xóa cơ sở dữ liệu, bảng, chỉnh sửa cấu trúc bảng, thêm, sửa và xóa dữ liệu.

Sao lưu và khôi phục: PhpMyAdmin cho phép sao lưu và khôi phục cơ sở dữ liệu của bạn với các tùy chọn như tạo một tệp SQL hoặc tạo bản sao của cơ sở dữ liệu.

Tìm kiếm và thay thế: PhpMyAdmin cho phép tìm kiếm và thay thế dữ liệu trong cơ sở dữ liệu của bạn.

Quản lý người dùng: PhpMyAdmin cho phép quản lý người dùng và các quyền truy cập vào cơ sở dữ liệu.

Các công cụ khác: PhpMyAdmin còn cung cấp các tính năng như phân tích cú pháp SQL, tạo truy vấn SQL, tạo chỉ mục và ràng buộc và quản lý các phiên làm việc.

### 1.3.2 Ưu điểm của phpMyAdmin

- Dễ sử dụng: PhpMyAdmin cung cấp một giao diện đồ họa dễ sử dụng để quản lý cơ sở dữ liệu, đặc biệt là cho những người không có nhiều kinh nghiệm với lệnh SQL.

- Tiện ích và tính năng phong phú: PhpMyAdmin cung cấp nhiều tính năng và tiện ích để quản lý cơ sở dữ liệu, bao gồm tạo, xóa, sửa, sao lưu và khôi phục cơ sở dữ liệu, tìm kiếm và thay thế, quản lý người dùng và phân quyền, phân tích cú pháp SQL và nhiều hơn nữa.

- Mã nguồn mở: PhpMyAdmin là một phần mềm mã nguồn mở, có nghĩa là nó miễn phí và có thể tùy chỉnh để phù hợp với nhu cầu của bạn.

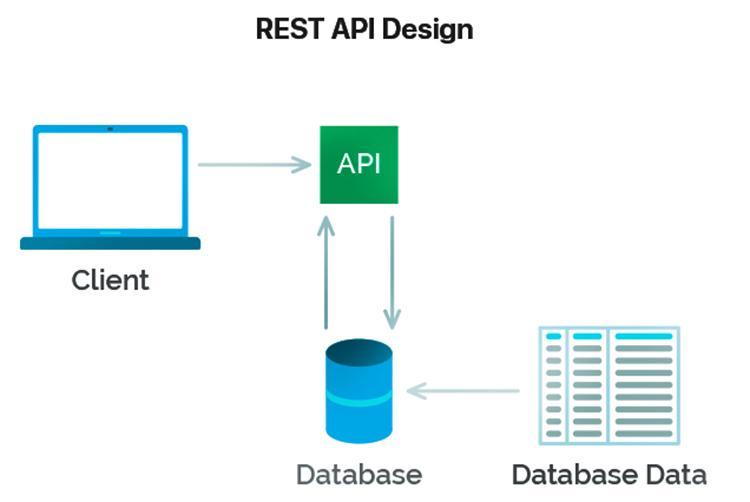
- Tương thích với nhiều hệ thống quản trị cơ sở dữ liệu: PhpMyAdmin được thiết kế để tương thích với nhiều hệ thống quản trị cơ sở dữ liệu khác nhau, bao gồm MySQL và MariaDB.

- Tích hợp với nhiều hệ thống quản lý máy chủ web: PhpMyAdmin có thể được tích hợp với nhiều hệ thống quản lý máy chủ web khác nhau như Apache, Nginx và Lighttpd.

## 1.4 Restful API

### 1.4.1 Giới thiệu về Resful API

**Rest API** là một ứng dụng cho phép chuyển đổi cấu trúc dữ liệu có những phương thức kết nối với các thư viện và ứng dụng khác. Đây là một giải pháp giúp xây dựng lên những ứng dụng web chuyên dụng thay thế cho những giải pháp cũ.



Hình 1. Res tAPI

**Restful API** là một trong những tiêu chuẩn được sử dụng để thiết kế API để tiện hơn cho việc quản lý các resource. Dữ liệu được truyền tải thông qua HTTP.

### 1.4.2 Các thành phần của Restful API

* **API** (Application Programming Interface) là giao diện lập trình ứng dụng giúp đưa ra các phương thức kết nối với các thư viện và ứng dụng khác.
* **REST** (Reprosemtational State Transfer) là một đại diện cho sự chuyển đổi dữ liệu. Nghĩa là client và server hoàn toàn độc lập, các yêu cầu gửi tới Rest API đều không mang theo trạng thái của yêu cầu trước đó.

## 1.5 Một số công cụ sử dụng

### 1.5.1 Github

- GitHub là dịch vụ máy chủ lưu trữ mã nguồn lập trình lớn nhất thế giới, giúp nhiều người có thể cùng hợp tác và giám sát những thay đổi của dự án. Khi sử dụng GitHub, ngoài các công việc chính như tạo Branch, tạo Pull Request và Fork một Repository, bạn có thể theo dõi, tương tác với nhũng người khác như một mạng xã hội thông thường.

- GitHub được viết bằng Ruby on Rails và Erlang do Tom Preston-Werner, Chris Wanstrath, và PJ Hyett phát triển trang web được đưa ra và chạy chính thức vào tháng 4 năm 2008.

- Tính đến thời điểm tháng 3 năm 2018 Github đang là dịch vụ máy chủ lưu trữ các mã nguồn lập trình lớn nhất thế giới với hơn 25 triệu người dùng và hơn 80 triệu mã nguồn dự án Github đã trở thành một phần không thể thiêu đối với cộng đồng phát triển mã nguồn mở và cộng đồng lập trình viên trên toàn thế giới.

**Tính năng của Github**

GitHub được coi là một mạng xã hội dành cho lập trình viên lớn nhất và dễ dùng nhất với các tính năng cốt lõi như:

• Wiki, issue, thống kê, đổi tên project, project được đặt vào namespace là user

• Watch project: theo dõi hoạt động của project của người khác. Xem quá trình người ta phát triển phầm mềm thế nào, project phát triển ra sao

• Follow user: theo dõi hoạt động của người khác.

### 1.5.2 Case Studio

CASE Studio 2 là công cụ vẽ mô hình dữ liệu chuyên nghiệp và có thể tùy chỉnh hỗ trợ những lập trình viên hay nhân viên thiết kế dữ liệu trong việc vẽ mô hình Entity Relationship Diagrams (ERD) và Data Flow Diagrams (DFD) cũng như tạo script SQL cho nhiều cơ sở dự liệu một cách tự động. Phần mềm hỗ trợ đầy đủ cho 20 cơ sở dữ liệu như Oracle, DB2, MS SQL, Sybase, MySQL, Firebird, PostgreSQL vvv. Chức năng chính của CASE Studio 2 : tạo mô hình ERD từ mã script SQL (DDL), Đảo mã từ file HTML cụ thể hay file dữ liệu RTF, xuất mô hình Data Flow Diagrams thành file quản lý định đạng XML, Templater editor và còn nữa. Phiên bản 2.23.1 hỗ trợ thêm cho: Informix 10, Oracle 9i / 10g, MS SQL 2005 / 2000, DB2 version 8, PostgreSQL 8.1 / 8.0 , báo cáo định dạng HTML/RTF

### 1.5.3 Rational Rose

Rational Rose là một công cụ lập mô hình trực quan mạnh trợ giúp bạn phân tích và thiết kế các hệ thống phần mềm hướng đối tượng. Nó được dùng để lập mô hình hệ thống trước khi bạn viết mã (code). Dùng mô hình, bạn có thể bắt kịp những thiếu sót về thiết kế, trong khi việc chỉnh sửa chúng vẫn chưa tốn kém. − Mô hình Rose là bức tranh về một hệ thống từ nhiều góc nhìn khác nhau. Nó bao gồm tất cả các sơ đồ UML, các actor, các use case, các đối tượng, các lớp, các thành phần… Nó mô tả chi tiết nội dung mà hệ thống sẽ gộp và cách nó sẽ làm việc.

### 1.5.4 Visual Studio Code

Visual Studio Code Là một trình biên tập lập trình code miễn phí dành cho Windows, Linux và macOS, Visual Studio Code được phát triển bởi Microsoft. Nó được xem là một sự kết hợp hoàn hảo giữa IDE và Code Editor.

Visual Studio Code hỗ trợ chức năng debug, đi kèm với Git, có syntax highlighting, tự hoàn thành mã thông minh, snippets, và cải tiến mã nguồn. Nhờ tính năng tùy chỉnh, Visual Studio Code cũng cho phép người dùng thay đổi theme, phím tắt, và các tùy chọn khác.

**Các tính năng**

* **Hỗ trợ nhiều ngôn ngữ lập trình**

Visual Studio Code hỗ trợ nhiều ngôn ngữ lập trình như C/C++, C#, F#, Visual Basic, HTML, CSS, JavaScript, … Vì vậy, nó dễ dàng phát hiện và đưa ra thông báo nếu chương chương trình có lỗi.

* **Hỗ trợ đa nền tảng**

Các trình viết code thông thường chỉ được sử dụng hoặc cho Windows hoặc Linux hoặc Mac Systems. Nhưng Visual Studio Code có thể hoạt động tốt trên cả ba nền tảng trên.

* **Cung cấp kho tiện ích mở rộng**

Trong trường hợp lập trình viên muốn sử dụng một ngôn ngữ lập trình không nằm trong số các ngôn ngữ Visual Studio hỗ trợ, họ có thể tải xuống tiện ích mở rộng. Điều này vẫn sẽ không làm giảm hiệu năng của phần mềm, bởi vì phần mở rộng này hoạt động như một chương trình độc lập.

* **Kho lưu trữ an toàn**

Đi kèm với sự phát triển của lập trình là nhu cầu về lưu trữ an toàn. Với Visual Studio Code, người dùng có thể hoàn toàn yên tâm vì nó dễ dàng kết nối với Git hoặc bất kỳ kho lưu trữ hiện có nào.

* **Hỗ trợ web**

Visual Studio Code hỗ trợ nhiều ứng dụng web. Ngoài ra, nó cũng có một trình soạn thảo và thiết kế website.

* **Lưu trữ dữ liệu dạng phân cấp**

Phần lớn tệp lưu trữ đoạn mã đều được đặt trong các thư mục tương tự nhau. Ngoài ra, Visual Studio Code còn cung cấp các thư mục cho một số tệp đặc biệt quan trọng.

* **Hỗ trợ viết Code**

Một số đoạn code có thể thay đổi chút ít để thuận tiện cho người dùng. Visual Studio Code sẽ đề xuất cho lập trình viên các tùy chọn thay thế nếu có.

* **Hỗ trợ thiết bị đầu cuối**

Visual Studio Code có tích hợp thiết bị đầu cuối, giúp người dùng khỏi phải chuyển đổi giữa hai màn hình hoặc trở về thư mục gốc khi thực hiện các thao tác.

* **Màn hình đa nhiệm**

Người dùng Visual Studio Code có thể mở cùng lúc nhiều tệp tin và thư mục – mặc dù chúng không hề liên quan với nhau.

* **Intellisense**

Hầu hết các trình viết mã đều có tính năng nhắc mã Intellisense, nhưng ít chương trình nào chuyên nghiệp bằng Visual Studio Code. Nó có thể phát hiện nếu bất kỳ đoạn mã nào không đầy đủ. Thậm chí, khi lập trình viên quên không khai báo biến, Intellisense sẽ tự động giúp họ bổ sung các cú pháp còn thiếu.

* **Hỗ trợ Git/Github**

Visual Studio Code hỗ trợ kéo hoặc sao chép mã trực tiếp từ GitHub. Mã này sau đó có thể được thay đổi và lưu lại trên phần mềm.

* **Bình luận**

Việc để lại nhận xét giúp người dùng dễ dàng nhớ công việc cần hoàn thành.

### 1.5.5 Laragon

Laragon là phần mềm cung cấp môi trường phát triển trên Windows bao gồm Mysql, PHP, Redis, Apache (WAMP Stack) một cách dễ dàng và nhanh chóng. Ngoài ra ta còn có thể thêm một số ngôn ngữ khác như Python, Ruby, Java, ... hoặc cơ sở dữ liệu khác như MongoDB, PostgreSQL vào Laragon chỉ với vài bước đơn giản. Laragon rất nhẹ, dễ dàng mở rộng và chỉ sử dụng dưới 4MB ram khi chạy. Nó không sử dụng bất cứ services nào của Windows mà có sẵn services của mình.

**Một số tính năng nổi bật**

* Pretty URLs: ta có thể dễ dàng tạo tên miền riêng như **viblo.local** thay vì **localhost**.
* **Portable**: dễ dàng di chuyển thư mục Laragon đến bất cứ đâu như USP, máy tính khác, ... mà không cần lo lắng liệu có lỗi hay project có không chạy được không.
* **Isolated**: Laragon là một môi trường độc lập với hệ điều hành giúp giữ cho hệ điều hành của bạn "clean".
* **Easy Operation**: Laragon tự động config tất cả mọi thứ giúp ta có thể dễ dàng thêm hoặc thay đổi phiên bản của PHP, Python, Java, Go, Apache, ...
* **Modern & Powerful**: Laragon giúp ta hoạt động với hai webserive Apache và Nginx và quản lý chúng. Hỗ trợ cài đặt SSL, cmd, các thao tác với cơ sở dữ liệu, public project lên với khách hàng, gửi mail từ local, ...

# CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG VÀ THIẾT KẾ CƠ SỞ DỮ LIỆU

## 2.1 Tổng quan hệ thống

Hệ thống Quản lý thiết bị công ty là phần mềm quản lý thiết bị của các nhân viên trong doanh nghiệp hoặc công ty. Phần mềm cho phép thêm mới, chỉnh sửa, và xem các thiết bị, quản lí thiết bị của nhân viên. Ngoài ra phần mềm còn có thể quản lý thông tin của các nhân viên trong công ty.

Người sử dụng hệ thống:

|  |  |
| --- | --- |
| **Người sử dụng** | **Mô tả** |
| Nhân viên | - Đăng nhập, Cập nhật thông tin cá nhân, Quên mất khẩu, Thay đổi mật khẩu, Gửi yêu cầu, Xem các thiết bị đã mượn. |
| Người quản lý thiết bị | - Đăng nhập, Cập nhật thông tin cá nhân, Quên mất khẩu, Thay đổi mật khẩu, Bảo trì danh mục,Quản lý, bảo trì thiết bị. |
| Giám đốc | -Đăng nhập, Cập nhật thông tin cá nhân, Quên mất khẩu, Thay đổi mật khẩu, Bảo trì danh mục, Quản lý, bảo trì thiết bị, Quản lý thông tin phòng ban, Quản lý thông tin người dùng. |

## 2.2 Xác định các yêu cầu chính của hệ thống

### 2.2.1 Phân tích các yêu cầu

#### 2.2.1.1 Các yêu cầu chức năng

**a) Những quyền lợi đối với nhân viên**

* Đăng nhập: nhân viên phải đăng nhập để sử dụng Website.
* Cập nhật thông tin cá nhân: nhân viên có thể cập nhật thông tin cá nhân.
* Đổi mật khẩu: nhân viên có thể đổi mật khẩu.
* Quên mật khẩu: nhân viên có thể reset mật khẩu khi quên mất khẩu
* Xem danh sách thiết bị đang mượn: nhân viên có xem danh sách các thiết bị mà mình đang mượn.

**b) Những quyền lợi đối với người quản lý thiết bị**

* Đăng nhập: người quản lý thiết bị phải đăng nhập để sử dụng Website.
* Cập nhật thông tin cá nhân: người quản lý thiết bị có thể cập nhật thông tin cá nhân.
* Đổi mật khẩu: người quản lý thiết bị có thể đổi mật khẩu.
* Quên mật khẩu: người quản lý thiết bị có thể reset mật khẩu khi quên mất khẩu.
* Quản lý danh mục: người quản lý thiết bị có thể xem, thêm, sửa, xóa các danh mục.
* Quản lý thiết bị: người quản lý thiết bị có thể xem, thêm, sửa, xóa thiết bị: xem danh sách các thiết bị có sẵn, thiết bị đang cho mượn, đang hỏng, đang bảo hành, đang sửa chữa, thiết bị đã bảo hành / sửa chữa,.

**c) Những quyền lợi đối với giám đốc**

* Đăng nhập: giám đốc phải đăng nhập để sử dụng Website.
* Cập nhật thông tin cá nhân: giám đốc có thể cập nhật thông tin cá nhân.
* Đổi mật khẩu: giám đốc có thể đổi mật khẩu.
* Quên mật khẩu: giám đốc có thể reset mật khẩu khi quên mất khẩu.
* Quản lý danh mục: giám đốc có thể xem, thêm, sửa, xóa các danh mục.
* Quản lý thiết bị: giám đốc có thể xem, thêm, sửa, xóa thiết bị: xem danh sách các thiết bị có sẵn, thiết bị đang cho mượn, đang hỏng, đang bảo hành, đang sửa chữa, thiết bị đã bảo hành / sửa chữa,.
* Quản lý phòng ban: giám đốc có thể xem, thêm, sửa, xóa phòng ban.
* Quản lý người dùng: giám đốc xem, thêm, sửa xóa người dùng.

#### 2.2.1.2 Các yêu cầu phi chức năng

* Hiệu năng cao, trải nghiệm mượt mà, ổn định.
* Giao diện thân thiện.
* Đáp ứng được lượng lớn truy cập đồng thời.
* Dễ dàng bảo trì và mở rộng.

## 2.2.2 Xây dựng và phân tích các biểu đồ use case

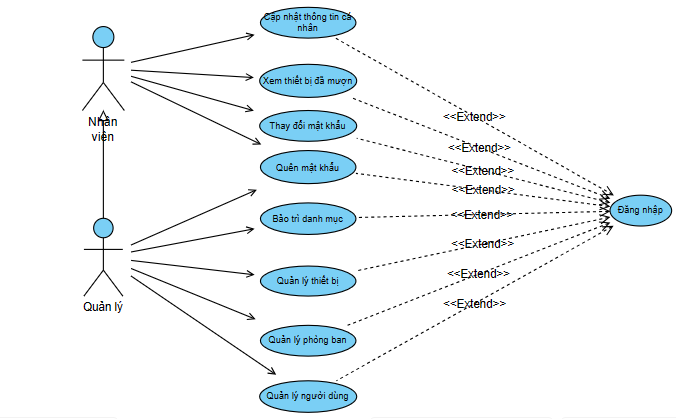
### 2.2.2.1 Các tác nhân

* **Giám đốc:** Quản lý nội dung website, xử lý các yêu cầu từ nhân viên, quản lý thông tin phòng ban, quản lý thông tin người dùng.
* **Người quản lý thiết bị:** Quản lý nội dung website, xử lý các yêu cầu từ nhân viên.
* **Nhân viên:** xem thông tin thiết bị đã mượn, gửi các yêu cầu.

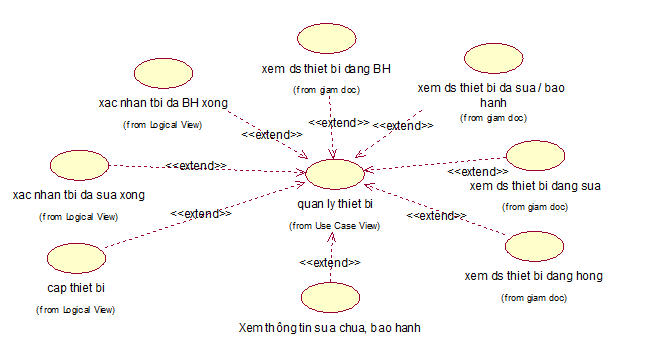
### 2.2.2.2 Các use case của hệ thống

|  |  |
| --- | --- |
| **Tác nhân** | **Use case** |
| Nhân viên | Đăng nhập  Cập nhật thông tin cá nhân  Quên mất khẩu  Thay đổi mật khẩu  Gửi yêu cầu  Xem các thiết bị đã mượn  Xem các yêu cầu đã gửi |
| Người quản lý thiết bị | Đăng nhập  Cập nhật thông tin cá nhân  Quên mất khẩu  Thay đổi mật khẩu  Bảo trì danh mục  Quản lý, bảo trì thiết bị  Quản lý yêu cầu  Quản lý phầm mềm |
| Giám đốc | Đăng nhập  Cập nhật thông tin cá nhân  Quên mất khẩu  Thay đổi mật khẩu  Bảo trì danh mục  Quản lý, bảo trì thiết bị  Quản lý yêu cầu  Quản lý thông tin phòng ban  Quản lý thông tin người dùng |

### 2.2.2.3 Biểu đồ use case



Hình 2. Biểu đồ use case tổng quát



Hình 2. Biểu đồ phân rã use case quản lý thiết bị

### 2.2.3 Phân tích use case

#### 2.2.3.1 Use case đăng nhập

**a) Mô tả chi tiết use case**

* **Mô tả vắn tắt**: Use case này cho phép người dùng đăng nhập vào hệ thống để sử dụng Website.
* **Luồng sự kiện**:
  + **Luồng cơ bản**:

1. Use ase này bắt đầu khi người dùng truy cập vào Website, hệ thống sẽ hiển thị ra form đăng nhập.
2. Người dùng nhập email và mật khẩu sau đó nhấn vào nút “đăng nhập”. Hệ thống sẽ kiểm tra email và mật khẩu người dùng nhập với thông tin trong bảng users, nếu khớp sẽ kiểm tra quyền truy cập của tài khoản và chuyển hướng đến trang web dành phù hợp với quyền của tài khoản. Use case kết thúc.

* **Luồng rẽ nhánh**

1. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.
2. Tại bước 2 trong luồng cơ bản, khi người dùng nhập thiếu thông tin hoặc thông tin không hợp lệ hoặc không tồn tại bản ghi trong bảng users. Hệ thống sẽ hiển thị thông báo lỗi và yêu cầu nhập lại. Người dùng có thể nhập lại để tiếp tục hoặc kết thúc thao tác. Use case kết thúc.

* **Các yêu cầu đặc biệt:**

Không có.

* **Tiền điều kiện:**

Không có.

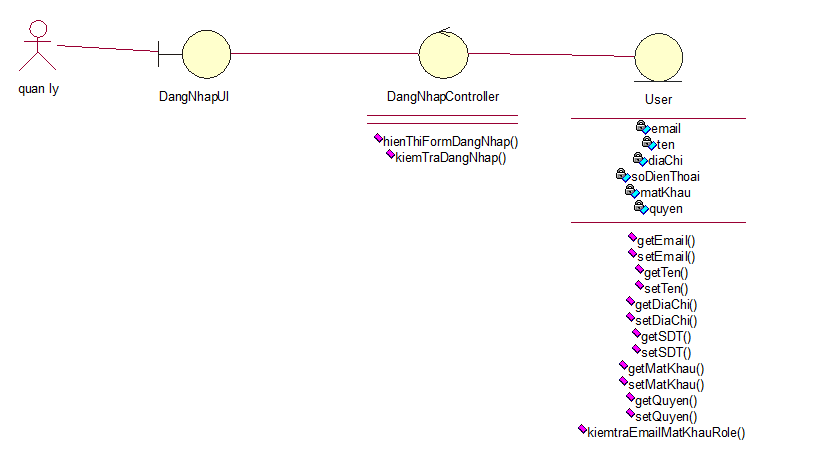
* **Hậu điều kiện:**

Không có.

* **Điểm mở rộng:**

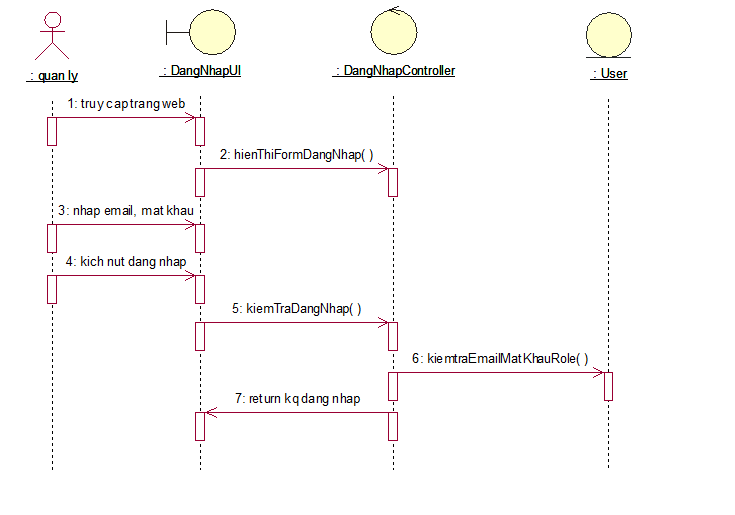
Không có.

**a) Biểu đồ lớp tham gia ca sử dụng (VOPC)**



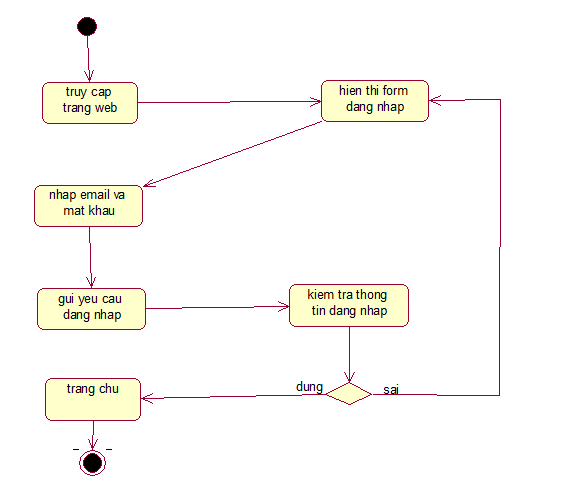
Hình 2. Biểu đồ lớp tham gia ca sử dụng use case Đăng nhập

**b) Biểu đồ trình tự (Sequence diagram)**



Hình 2. Biểu đồ trình tự use case Đăng nhập

**c) Biểu đồ hoạt động**

****

Hình 2. Biểu đồ hoạt động use case đăng nhập

#### 2.2.3.2 Use case quên mật khẩu

**a) Mô tả chi tiết use case**

* **Mô tả vắn tắt**: Use case này cho phép người dùng đặt lại mật khẩu khi quên mật khẩu.
* **Luồng sự kiện**:
  + **Luồng cơ bản**:

1. Use ase này bắt đầu khi người dùng ấn vào “quên mật khẩu” trên form đăng nhập. Hệ thống sẽ hiển thị form nhập email để xác nhận đặt lại mật khẩu.

2. Người dùng nhập email và ấn nút gửi, hệ thống sẽ gửi về email người dùng đường link để xác nhận đặt lại mật khẩu.

3. Người dùng ấn vào link xác nhận, hệ thống lưu thông tin email và 1 đoạn mã ngẫu nhiên vào bảng password\_resets và hiển thị form đặt lại mật khẩu.

4. Người dùng nhập mật khẩu mới và ấn đồng ý, hệ thống sẽ cập nhật mật khẩu người dùng vào bảng users. Use case kết thúc.

* **Luồng rẽ nhánh**

1. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.
2. Tại bước 2 trong luồng cơ bản, khi người dùng nhập email không tồn tại trong bảng users, hệ thống sẽ hiển thị thông báo lỗi. Người dùng có thể nhập lại để tiếp tục hoặc kết thúc thao tác. Use case kết thúc.
3. Tại bước 3 của luồng cơ bản, nếu người dùng không ấn vào link xác nhận thì use case kết thúc.
4. Tại bước 4 của luồng cơ bản, nếu người dùng không nhập mật khẩu hoặc nhập không hợp lệ thì hệ thống sẽ hiển thị thông báo lỗi Người dùng có thể nhập lại để tiếp tục hoặc kết thúc thao tác. Use case kết thúc.

* **Các yêu cầu đặc biệt:**

Không có.

* **Tiền điều kiện:**

Không có.

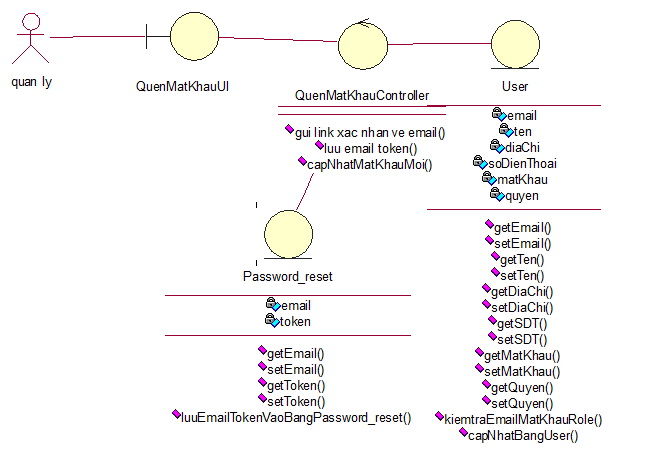
* **Hậu điều kiện:**

Không có.

* **Điểm mở rộng:**

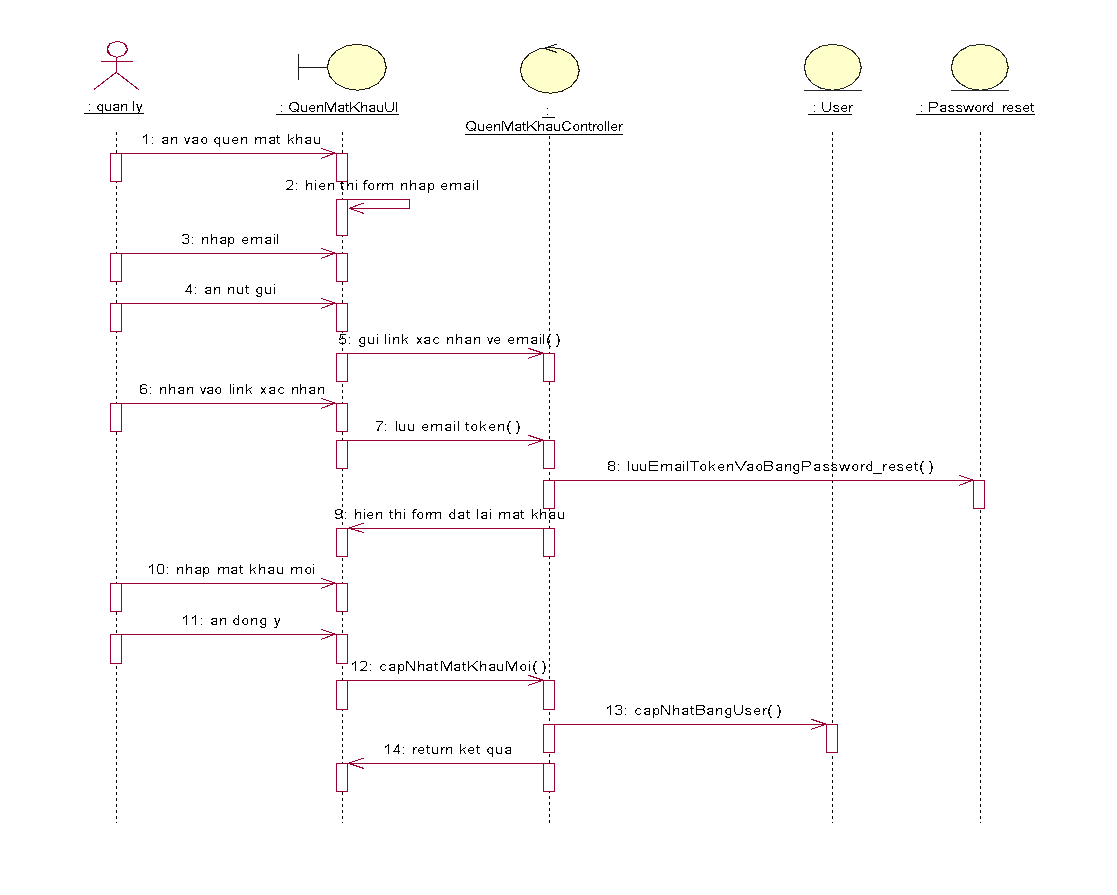
Không có.

**b) Biểu đồ lớp tham gia ca sử dụng (VOPC)**

****

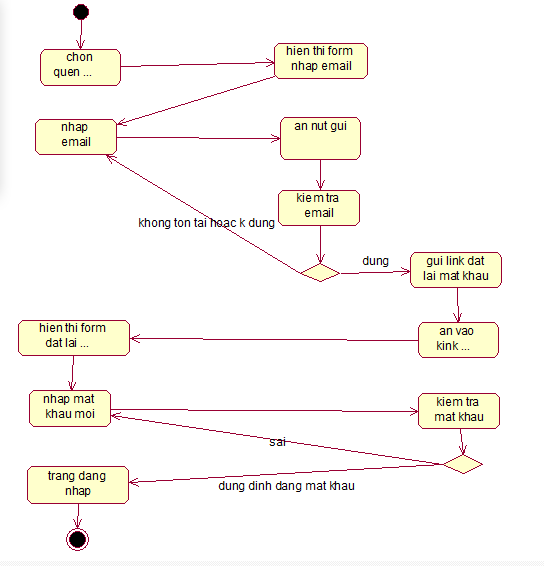
Hình 2. Biểu đồ lớp tham gia ca sử dụng use case quên mật khẩu

**c) Biểu đồ trình tự (Sequence diagram)**

****

Hình 2. Biểu đồ trình tự use case quên mật khẩu

**d) Biểu đồ hoạt động**

****

Hình 2. Biểu đồ hoạt động use case quên mật khẩu

#### 2.2.3.3 Use case cấp thiết bị

**a) Mô tả chi tiết use case**

* **Mô tả vắn tắt:** Use case này cho phép người quản trị cấp thiết bị cho nhân viên.
* **Luồng sự kiện:**
  + **Luồng cơ bản:**

1. Use case này bắt đầu khi người quản trị ấn vào “cấp thiết bị” trong trang danh sách yêu cầu cấp thiết bị. Hệ thống sẽ hiển thị trang cấp thiết bị.
2. Hệ thống sẽ lấy danh sách thiết bị có sẵn, người quản trị chọn thiết bị, chọn người nhận sau đó ấn “cấp” và “OK”. Cột “status” trong bảng devices và các bản ghi sẽ được cập nhật vào bảng requests và chuyển hướng về trang danh sách yêu cầu cấp thiết bị. Use case kết thúc.
   * **Luồng rẽ nhánh:**
3. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case, nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi. Use case kết thúc.
4. Tại bước 2 của luồng cơ bản, nếu không có thiết bị nào có sẵn, hệ thông sẽ hiển thị thông báo “không có thiết bị nào có sẵn”. Use case kết thúc.
5. Tại bước 2 trong luồng cơ bản, khi người quản trị nhập thiếu thông tin hoặc nhập thông tin không hợp lệ thì hệ thống sẽ hiển thị thông báo lỗi và yêu cầu nhập lại. người quản trị có thể nhập lại để tiếp tục. use case kết thúc.
6. Tại bước 2 của luồng cơ bản, khi người quản trị ấn nút “cancel” thao tác sẽ bị hủy. Use case kết thúc

* **Các yêu cầu đặc biệt**

Người dùng phải đăng nhập thành công với vai trò quản lý hoặc giám đốc thì mới sử dụng được chức năng này.

* **Tiền điều kiện**

Người dùng phải đăng nhập thành công thì mới sử dụng được chức năng này.

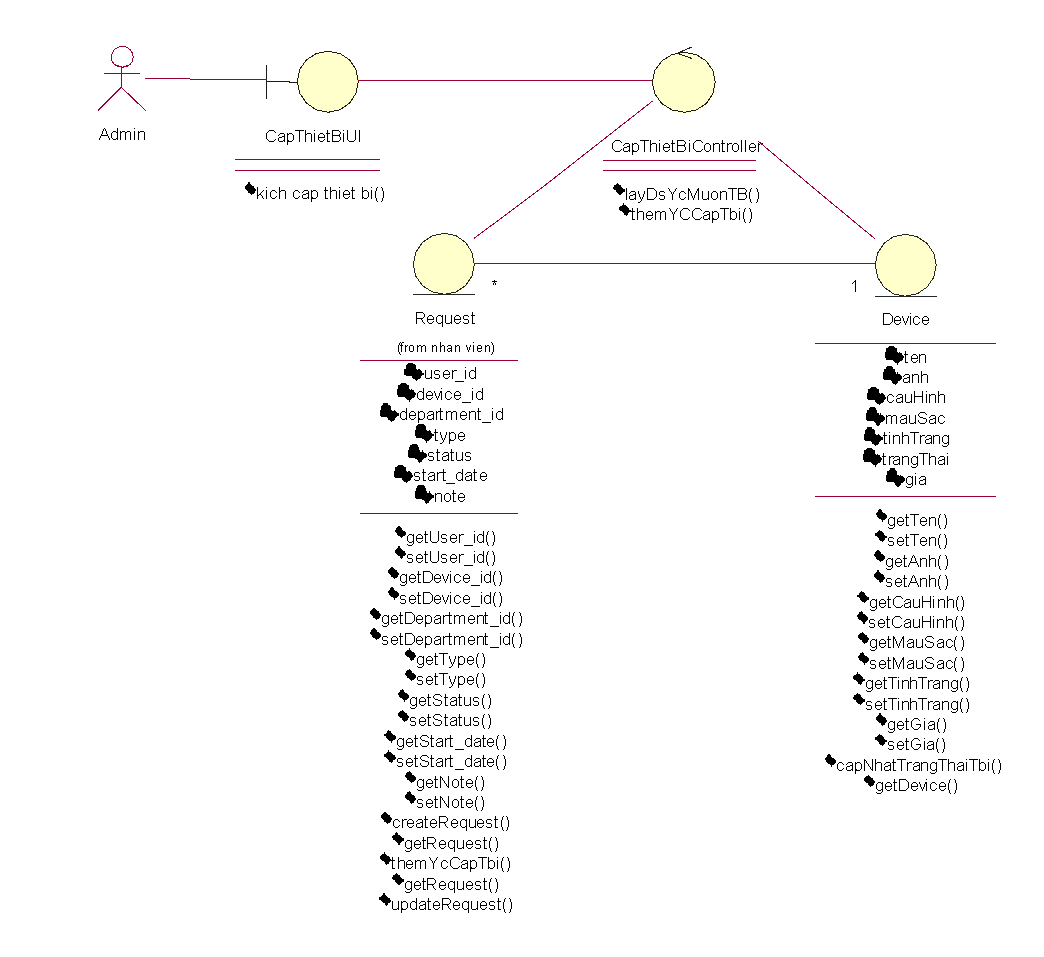
* **Hậu điều kiện**

Khi chức năng này được thực hiện thành công, các thông tin sẽ được cập nhật vào bảng requests và devices.

* **Điểm mở rộng:**

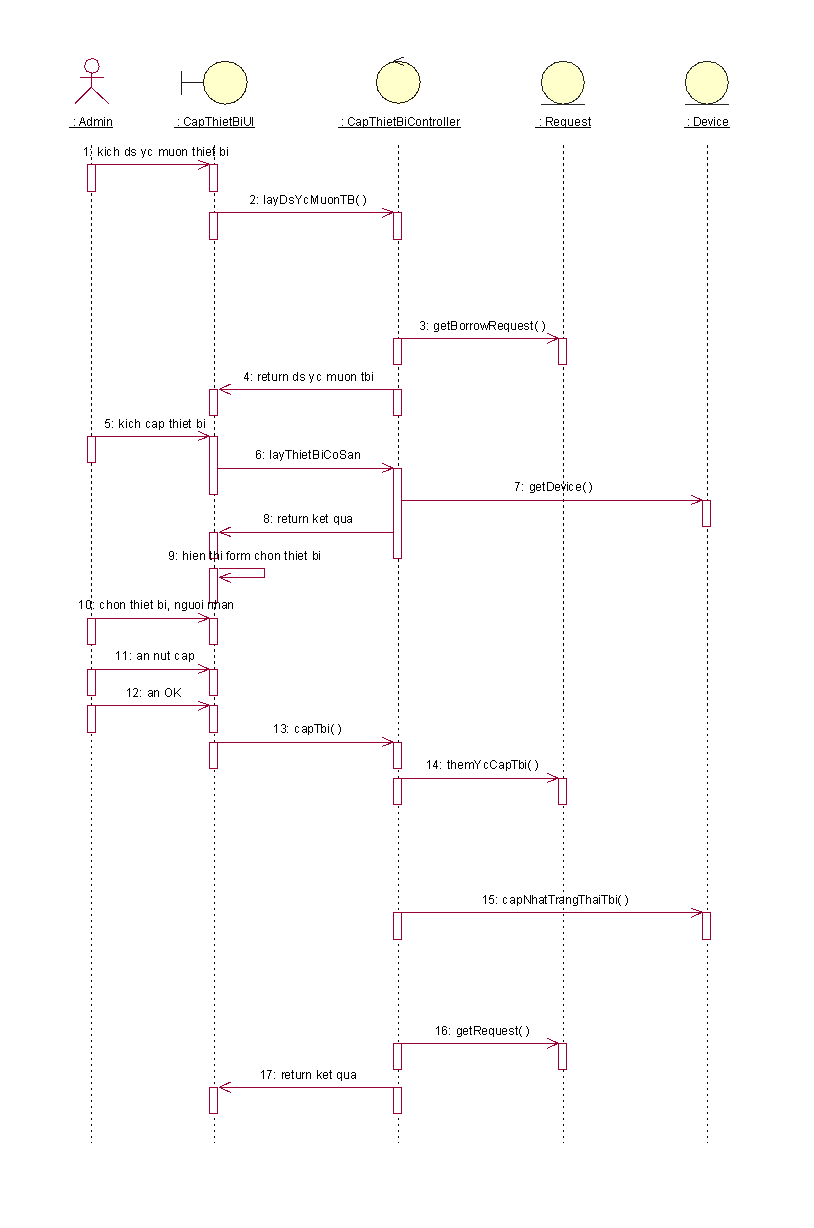
Không có

**b) Biểu đồ lớp tham gia ca sử dụng (VOPC)**

****

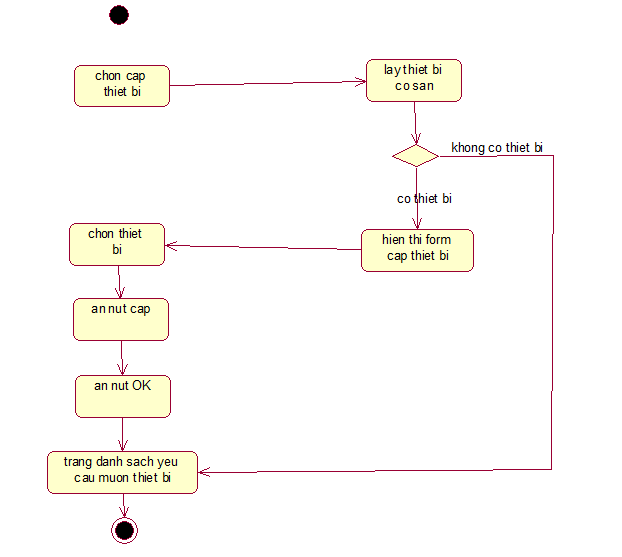
Hình 2. Biểu đồ lớp tham gia ca sử dụng use case cấp thiết bị

**c) Biểu đồ trình tự (Sequence diagram)**

****

Hình 2. Biểu đồ trình tự use case cấp thiết bị

**d) Biểu đồ hoạt động**

****

Hình 2. Biểu đồ hoạt động use case cấp thiết bị

#### 2.2.3.4 Use case xác nhận thiết bị đã sửa xong

**a) Mô tả chi tiết use case**

* **Mô tả vắn tắt:** Use case này cho phép người quản trị xác nhận thiết bị đã sửa xong và lưu thông tin chi tiết lần sửa chữa thiết bị đó.
* **Luồng sự kiện**
  + **Luồng cơ bản:**

1. Use case này bắt đầu khi người quản trị nhấn vào “xác nhận thiết bị đã sửa xong” trên trang danh sách thiết bị đang sửa chữa. Hệ thống sẽ hiển thị ra màn hình nhập thông tin chi tiết cho lần sửa đó.
2. Người quản trị nhập thông vào form đó rồi ấn “xác nhận” và “OK”, cột condition trong bảng devices, cột type trong bảng type\_repairs sẽ được cập nhật và bản ghi mới sẽ được tạo trong bảng repair\_details và chuyển hướng về trang danh sách thiết bị đang sửa chữa. Use case kết thúc.

* **Luồng rẽ nhánh**

1. Tại thười điểm bất kỳ trong quá trình thực hiện use case, nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi. Use case kết thúc
2. Tại bước 2 trong luồng cơ bản, nếu người quản trị nhập thiếu thông tin hoặc nhập thông tin không hợp lệ thì hệ thống sẽ hiển thị thông báo lỗi, người quản trị có thể nhập lại để tiếp tục.
3. Tại bước 2 của luồng cơ bản, khi người quản trị ấn nút “cancel” thao tác trên sẽ bị hủy. Use case kết thúc

* **Các yêu cầu đặc biệt**

Người dùng phải đăng nhập thành công với vai trò quản lý hoặc giám đốc thì mới sử dụng được chức năng này.

* **Tiền điều kiện**

Người dùng phải đăng nhập thành công thì mới sử dụng được chức năng này.

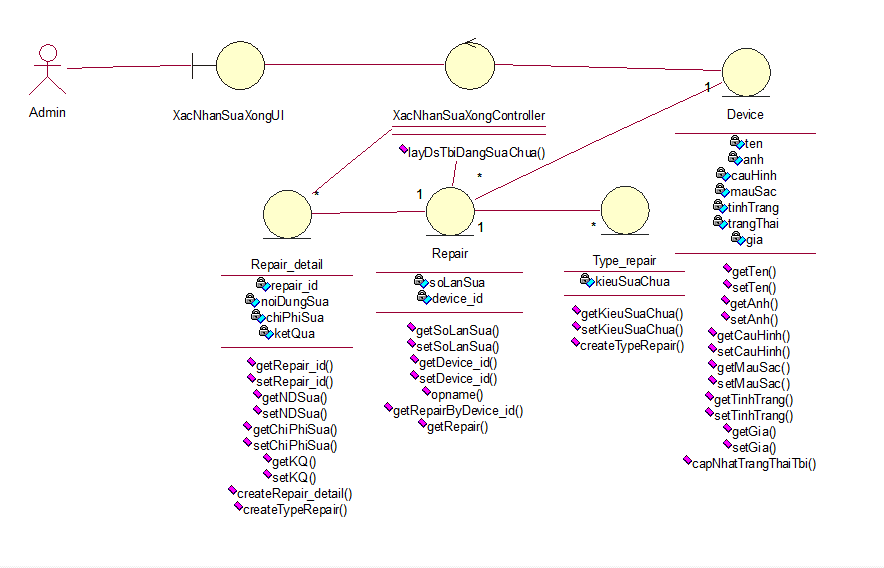
* **Hậu điều kiện**

Khi chức năng này được thực hiện thành công, các thông tin sẽ được cập nhật vào bảng type\_repairs, devices và repair\_details.

* **Điểm mở rộng:**

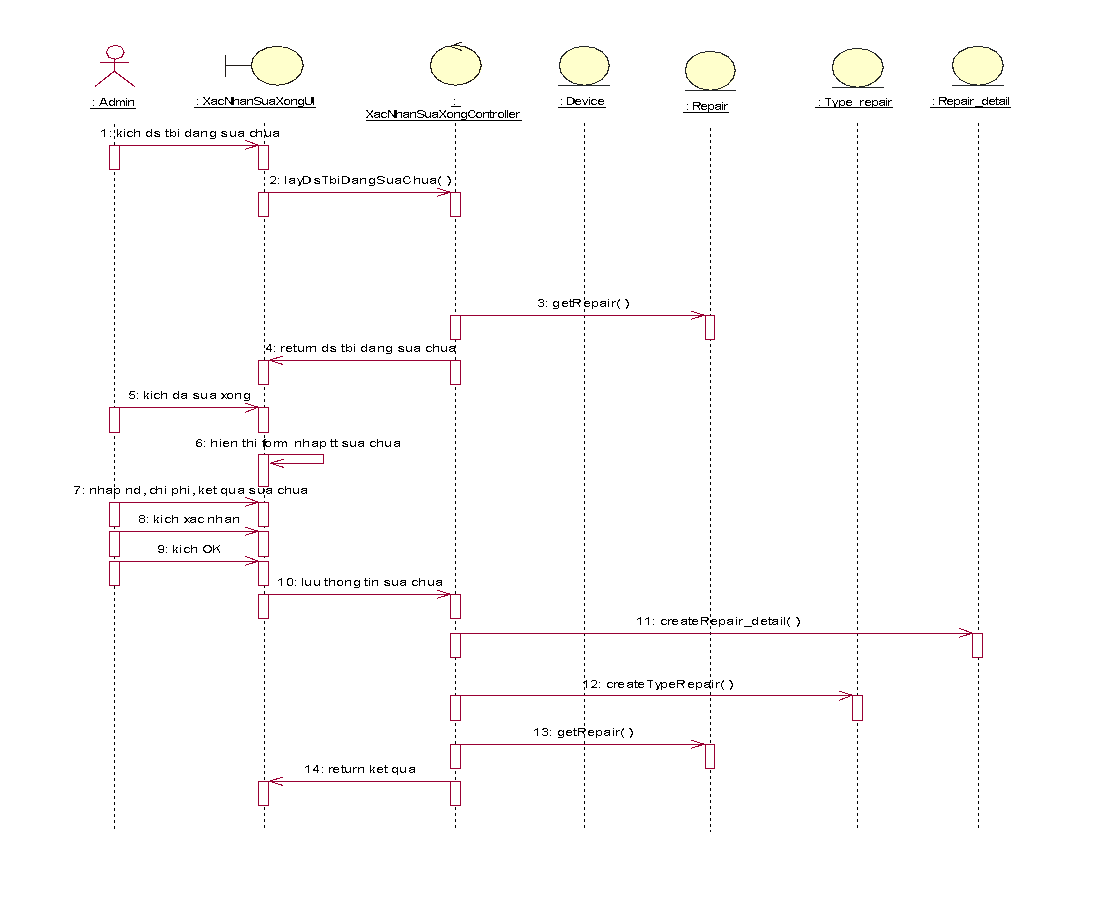
Không có

**b) Biểu đồ lớp tham gia ca sử dụng (VOPC)**

****

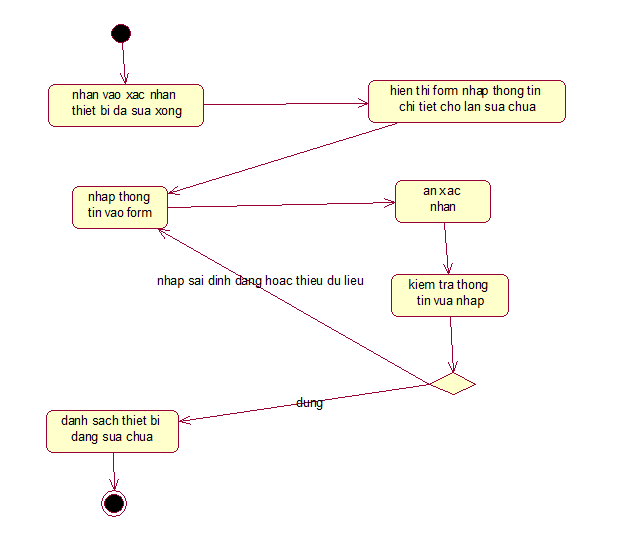
Hình 2. Biểu đồ lớp tham gia ca sử dụng use case xác nhận thiết bị đã sửa xong

**c) Biểu đồ trình tự (Sequence diagram)**

****

Hình 2. Biểu đồ trình tự use case xác nhận thiết bị đã sửa xong

**d) Biểu đồ hoạt động**

****

Hình 2. Biểu đồ hoạt động use case xác nhận thiết bị đã sửa xong

#### 2.2.3.5 Use case xem thông sửa chữa hoặc bảo hành của thiết bị

**a) Mô tả chi tiết use case**

* **Mô tả vắn tắt:** Use case này cho phép người quản trị xem danh sách các thiết bị đã từng sửa chữa hoặc bảo hành và thông tin các lần sửa chữa, bảo hành của thiết bị.
* **Luồng sự kiện:**
  + **Luồng cơ bản:**

1. Use case này bắt đầu khi người quản trị ấn vào nut “danh sách thiết bị đã sửa chữa hoặc bảo hành”, hệ thống sẽ lấy tất cả các thiết bị có cột repair\_count lớn hơn 0 trong bảng repairs hoặc cột warranty\_count lớn hơn 0 trong bảng warranties hiển thị lên màn hình.
2. Người dùng ấn vào “xem chi tiết”, hệ thống sẽ lấy thông tin chi tiết bảo hành từ bảng warranty\_details, thông tin chi tiết sửa chữa và kiểu sửa chữa từ bảng type\_repairs, repair\_details hiển thị lên màn hình. Use case kết thúc.

* **Luồng rẽ nhánh**

1. Tại thời điểm bất kỳ trong quá trình thực hiện use case, nếu không kết nối được tới cơ sở dữ liệu thì sẽ hiển thị thông báo lỗi. Use case kết thúc.
2. Tại bước 1 trong luồng cơ bản, nếu trong cơ sở dữ liệu không có thiết bị nào thỏa mãn điều kiện thì hệ thống sẽ hiển thị thông bảo chưa có thiết bị nào từng sửa chữa hoặc bảo hành. Use case kết thúc.

* **Các yêu cầu đặc biệt**

Người dùng phải đăng nhập thành công với vai trò quản lý hoặc giám đốc thì mới sử dụng được chức năng này

* **Tiền điều kiện**

Người dùng phải đăng nhập thành công thì mới sử dụng được chức năng này.

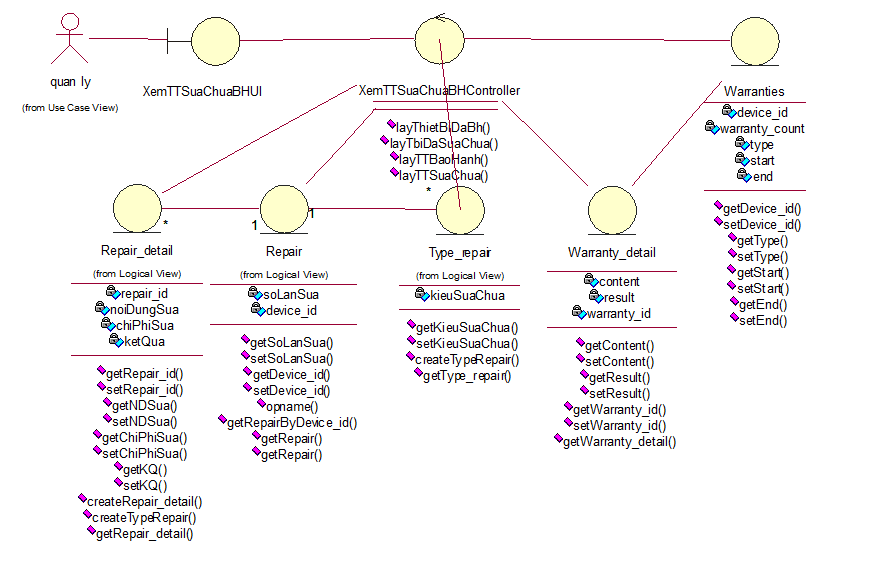
* **Hậu điều kiện**

Không có

* **Điểm mở rộng:**

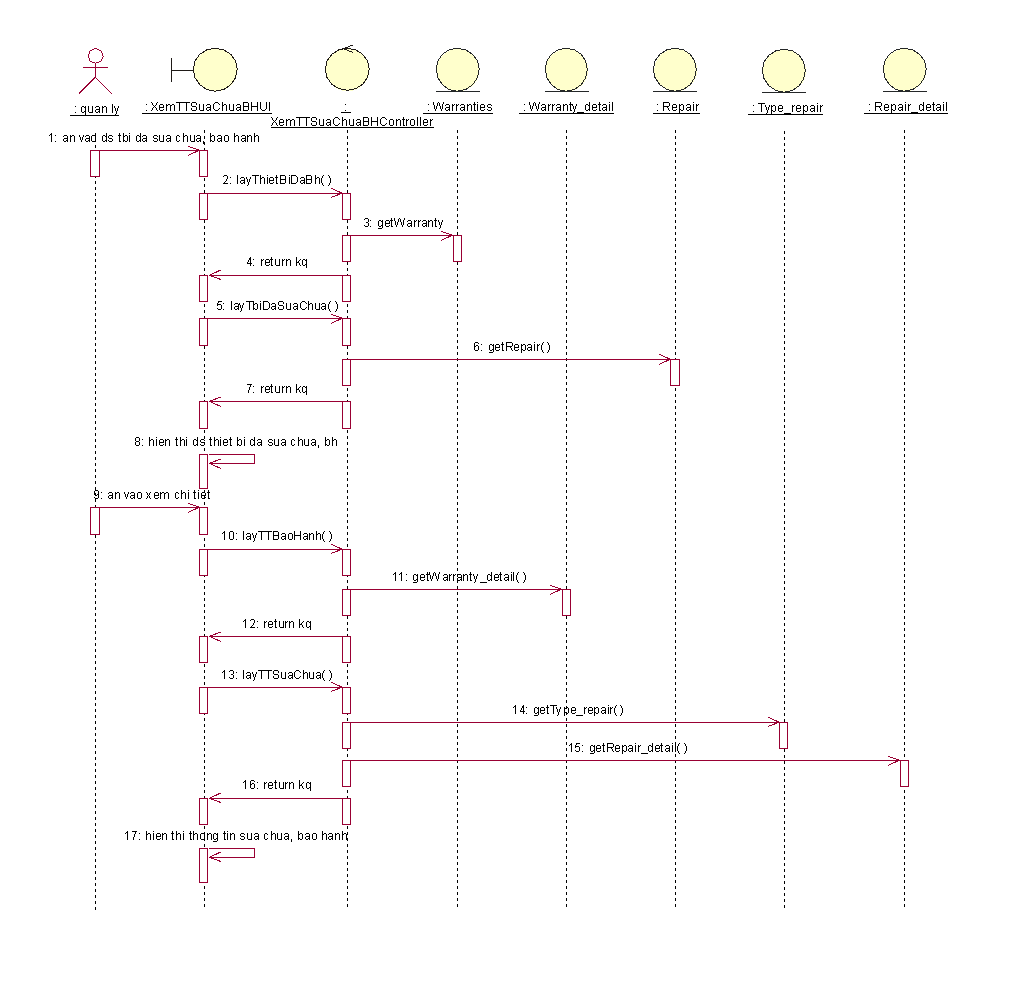
Không có

**b) Biểu đồ lớp tham gia ca sử dụng (VOPC)**

****

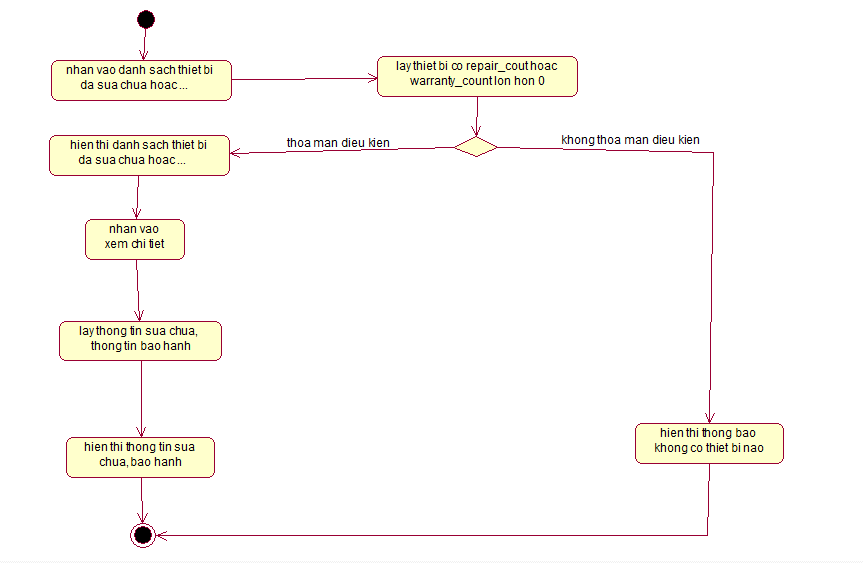
Hình 2. Biểu đồ lớp tham gia ca sử dụng use case xem thông sửa chữa hoặc bảo hành của thiết bị

**c) Biểu đồ trình tự (Sequence diagram)**

****

Hình 2. Biểu đồ trình tự use case xem thông sửa chữa hoặc bảo hành của thiết bị

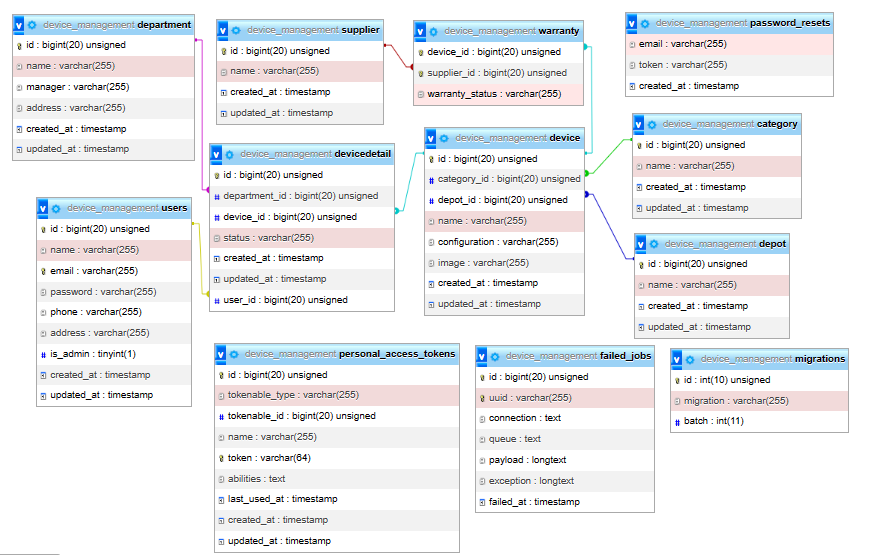
**d) Biểu đồ hoạt động**

****

Hình 2. Biểu đồ hoạt động use case xem thông sửa chữa hoặc bảo hành của thiết bị

## 2.2.4 Xây dựng cơ cở dữ liệu

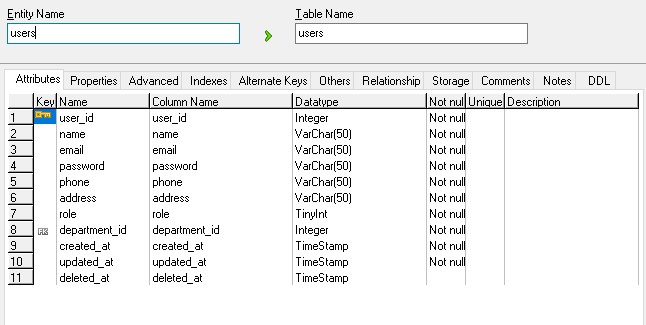
### 2.2.4.1 Biểu đồ thực thể liên kết (Entity Relationship Diagram)



Hình 2. Biểu đồ ERD thiết kế cơ sở dữ liệu

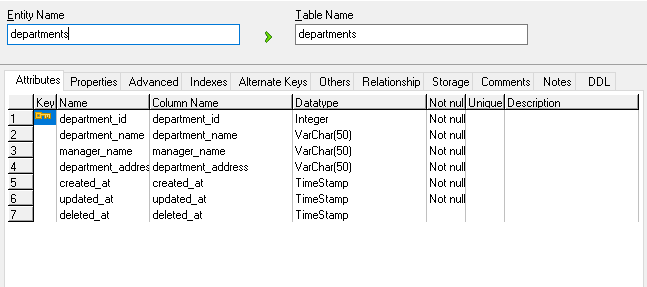
### 2.2.4.2 Các bảng trong cơ sở dữ liệu

**Bảng users** (Lưu thông tin tài khoản người dùng)



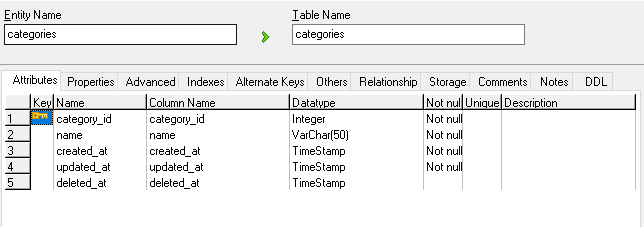
Hình 2. Bảng users

**Bảng departments** (Lưu thông tin phòng ban)



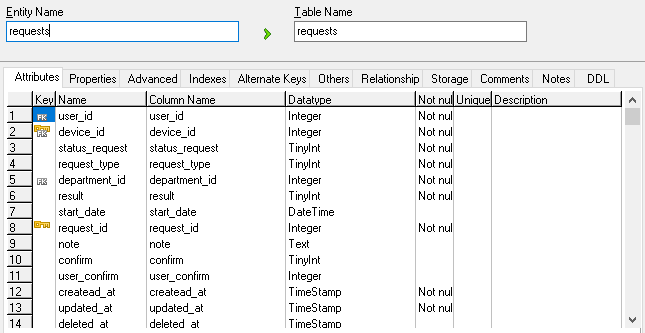
Hình 2. Bảng departments

**Bảng categories** (Lưu thông tin danh mục thiết bị)



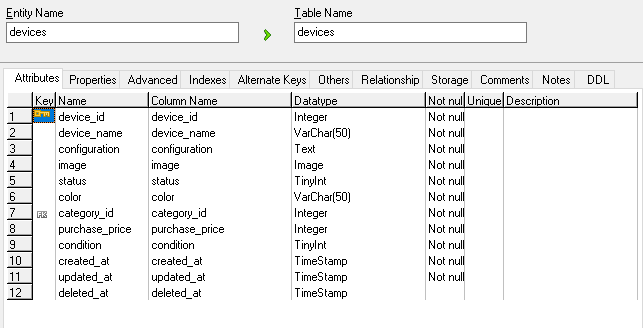
Hình 2. Bảng categories

**Bảng requests** (Lưu thông tin yêu cầu)



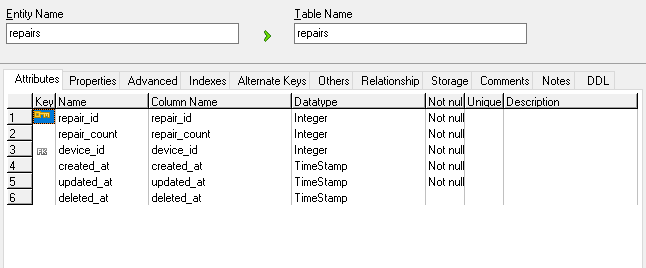
Hình 2. Bảng requests

**Bảng devices** (Lưu thông tin thiết bị)



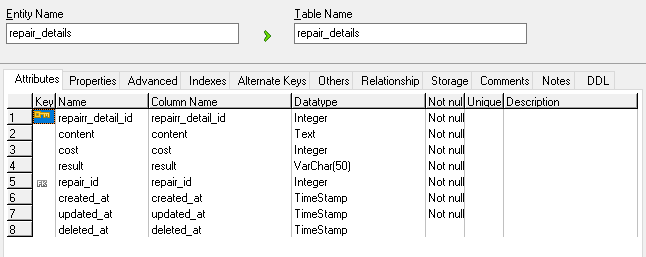
Hình 2. Bảng devices

**Bảng repairs** (Lưu thông tin sửa chữa thiết bị)



Hình 2. 24 Bảng repairs

**Bảng repair\_details** (Lưu thông tin chi tiết các lần sửa chữa)



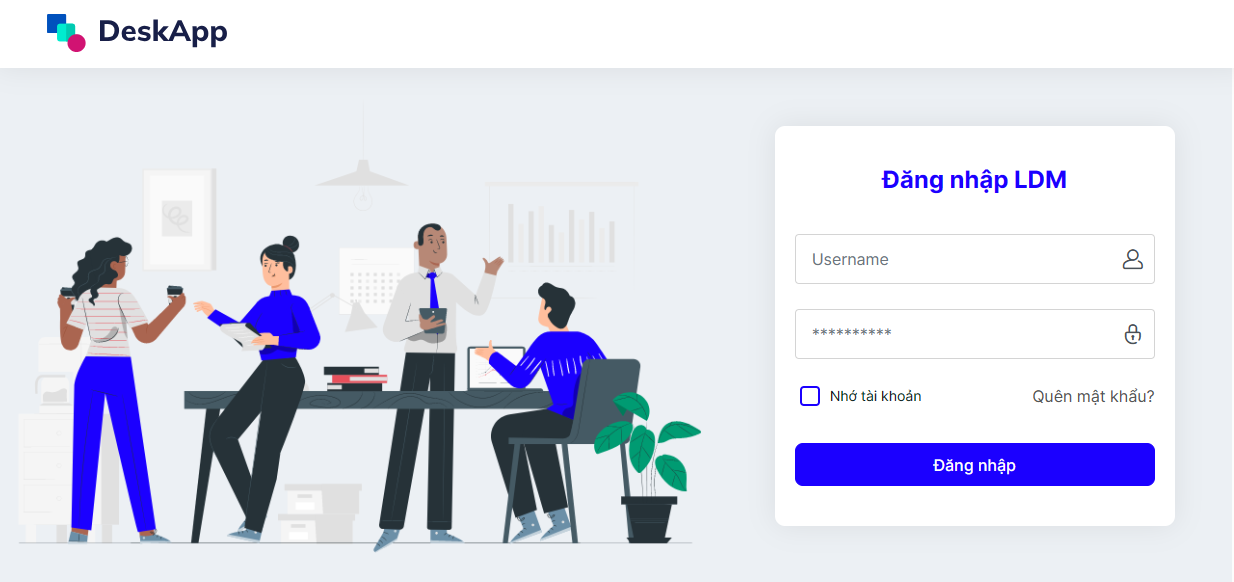
Hình 2. Bảng repair\_details

# CHƯƠNG 3. KẾT QUẢ THỬ NGHIỆM

## 3.1 Giao diện chương trình

#### 3.1.1 Màn hình đăng nhập

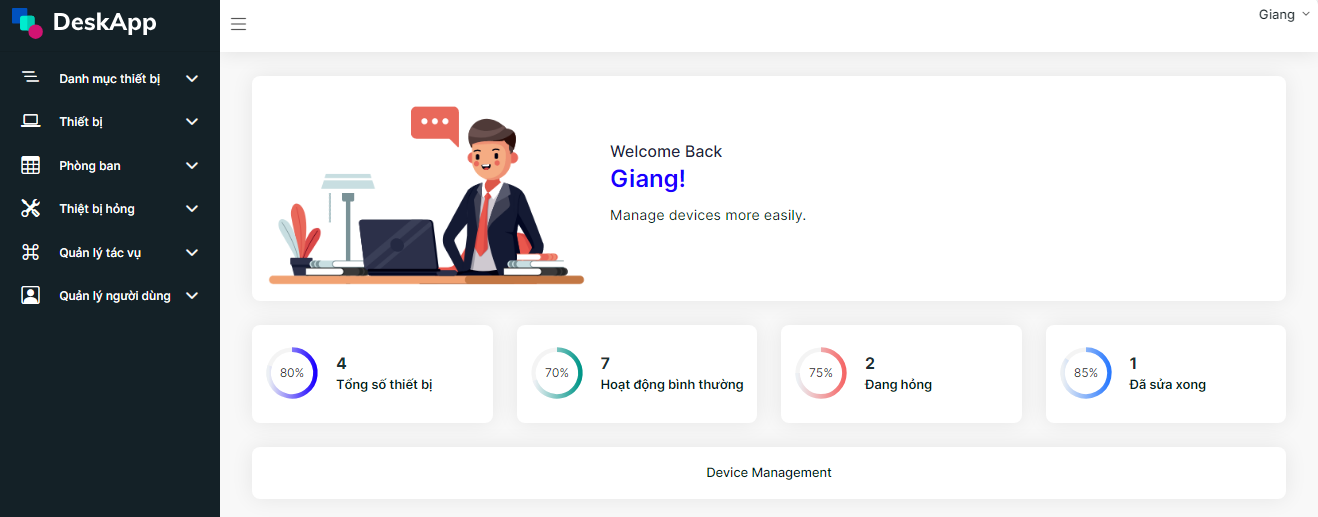
Màn hình này hiển thị biểu mẫu đăng nhập, người dùng bắt buộc phải nhập đầy đủ các trường email và mật khẩu để đăng nhập vào hệ thông.



Hình 3. Màn hình đăng nhập

#### 3.1.2 Màn hình trang chủ

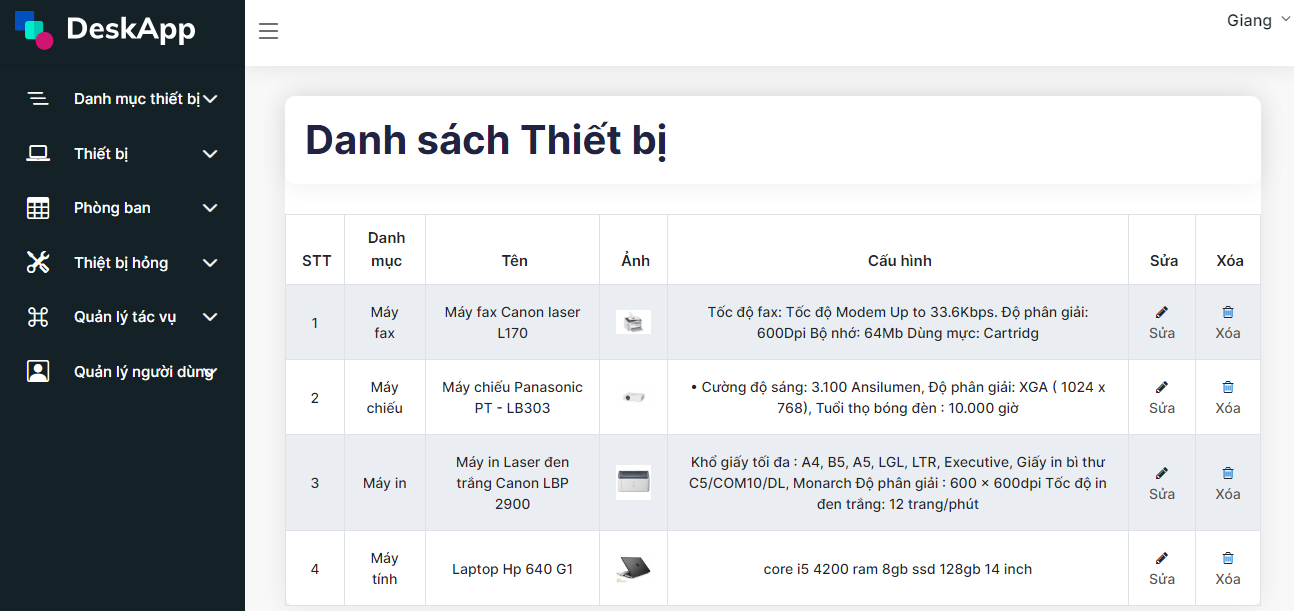
Khi người dùng đăng nhập thành công vào hệ thống sẽ hiển thị màn hình trang chủ. Màn hình này hiển thị bảng thống kê tổng số thiết bị, số lượng thiết bị theo danh mục, thiết bị đang hỏng và thiết bị đã sửa.



Hình 3. Màn hình trang chủ

#### 3.1.3 Màn hình danh sách thiết bị

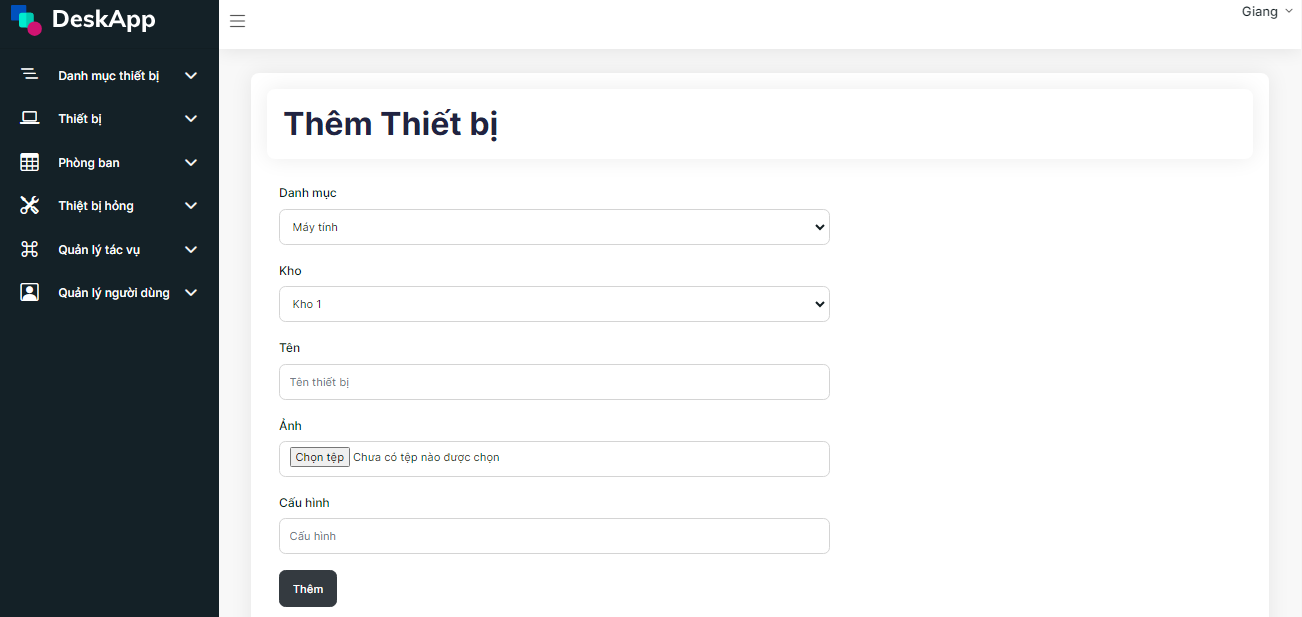
Màn hình này hiển thị danh sách và thông tin của tất cả thiết bị.



Hình 3. Màn hình danh sách thiết bị

#### 3.1.4 Màn hình thêm mới thiết bị

Màn hình này hiện thị giao diện chức năng thêm mới thiết bị. Khi thêm mới một thiết bị, người dùng cần chọn danh mục thiết bị, kho, hình ảnh, nhập tên, cấu hình, sau đó ấn nút thêm để thêm mới thiết bị.



Hình 3. Màn hình thêm mới thiết bị

#### 3.1.5 Màn hình danh sách thiết bị đang cho mượn

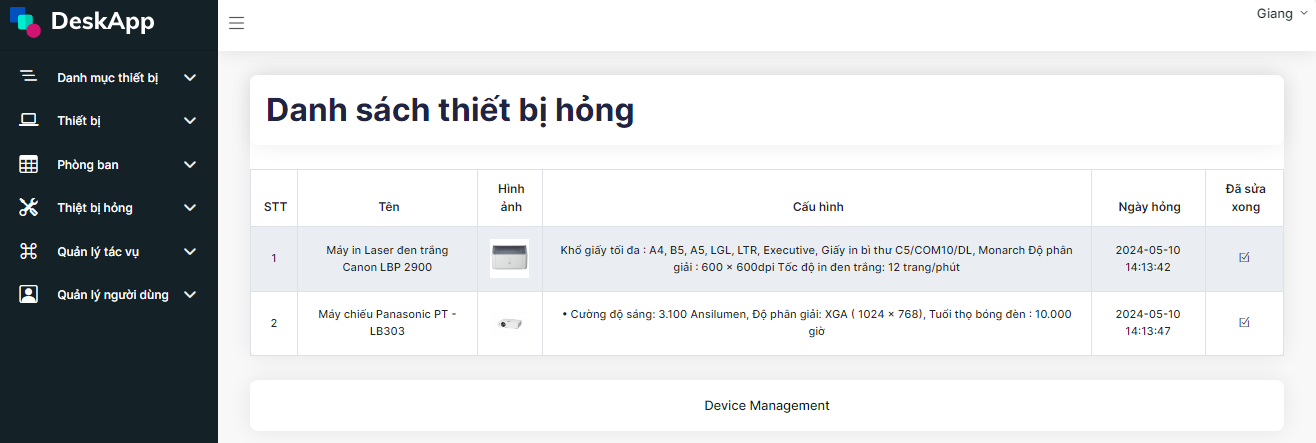
Màn hình này hiển thị danh sách thiết bị đang cho mượn.



Hình 3. Màn hình danh sách thiết bị đang cho mượn

#### 3.1.6 Màn hình danh sách thiết bị đang hỏng

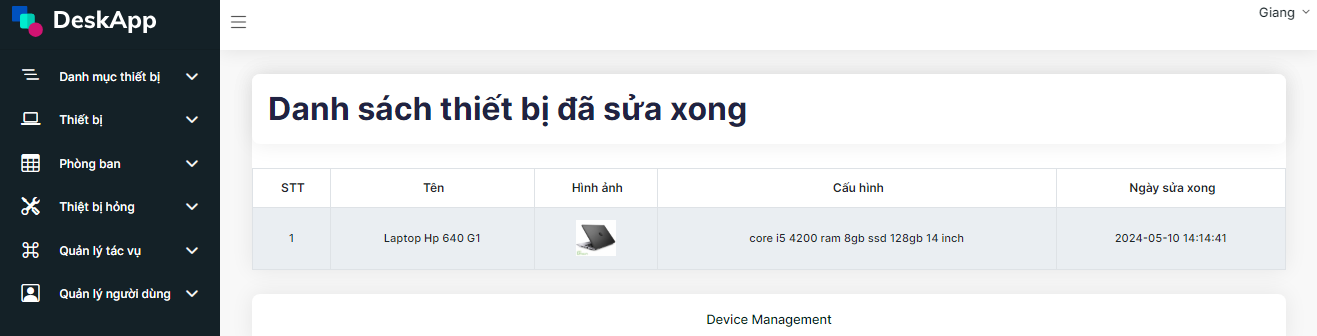
Màn hình này hiển thị danh sách thiết bị đang hỏng.



Hình 3. Màn hình danh sách thiết bị đang hỏng

#### 3.1.7 Màn hình danh sách thiết bị đã sửa chữa hoặc bảo hành

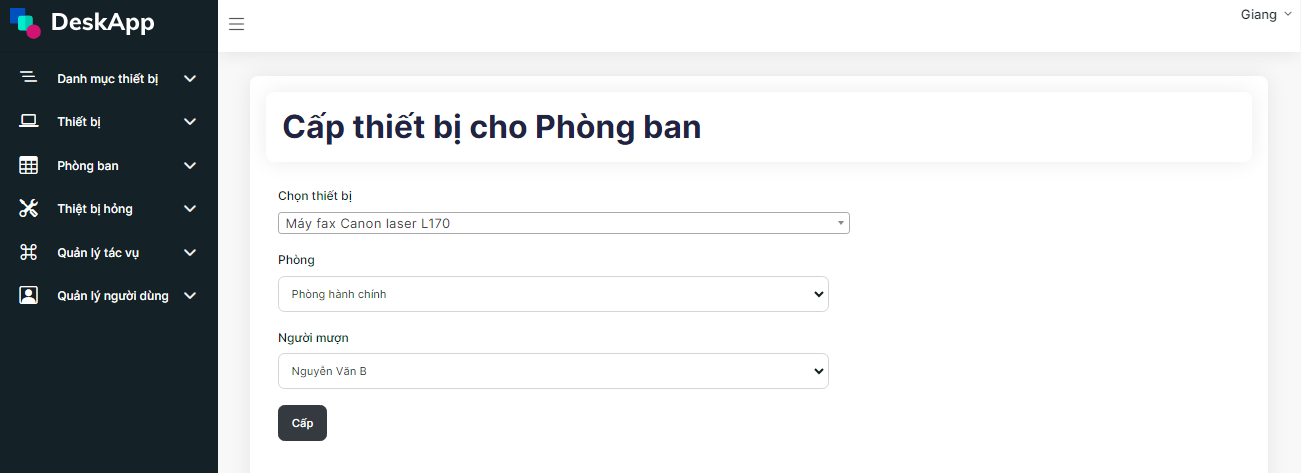
Màn hình này hiển thị danh sách thiết bị đã sửa chữa hoặc bảo hành.



Hình 3. Màn hình danh sách thiết bị đã sửa chữa hoặc bảo hành

#### 3.1.8 Màn hình cấp thiết bị

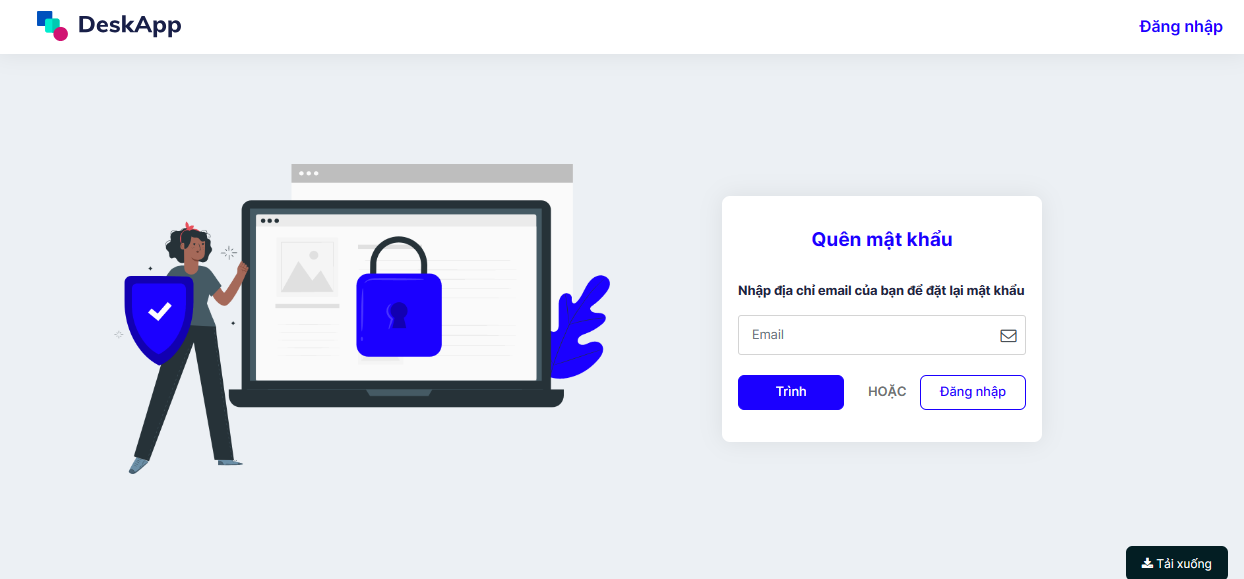
Màn hình này cho phép người quản trị cấp thiết bị cho nhân viên. Người quản trị cần chọn thiết bị để thực hiện chức năng này.



Hình 3. Màn hình cấp thiết bị

#### 3.1.9 Màn hình quên mật khẩu

Khi người dùng nhấn vào “quên mật khẩu”, hệ thống sẽ hiển thị màn hình yêu cầu nhập email để nhận link xác nhận đặt lại mật khẩu.



Hình 3. Màn hình gửi link xác nhận đặt lại mật khẩu

Sau khi người dùng nhập email và gửi link, hệ thống sẽ gửi một đường link xác nhận đặt lại mật khẩu vào email đó. Người dùng ấn vào link sẽ hiển thị màn hình đặt lại mật khẩu và nhập mật khẩu mới, xác nhận mật khẩu để hoàn thành chức năng này.

# CHƯƠNG 4. KIỂM THỬ WEBSITE

## 4.1 Thiết kế và tạo testcase

Trong giai đoạn này, sẽ thiết kế testcase / test checklist ( với test manual) và test script ( với automation test) bằng cách sử dụng kỹ thuật kiểm thử hộp đen. Dữ liệu kiểm thử cũng được chuẩn bị từ giai đoạn này.

* Tạo test checklist/testcase hoặc test script
* Review lại các testcase và test script xem đã sát với đặc tả yêu cầu hay chưa.
* Tạo dữ liệu test.

## 4.2 Thực hiện kiểm thử

Trong giai đoạn này, sẽ thực hiện kiểm thử dựa trên bản kế hoạch kiểm thử và kịch bản kiểm thử đã được chuẩn bị. Các bug được tìm thấy sẽ được note lại để sửa chữa.

* Thực hiện kiểm thử theo kế hoạch
* Log bug và ghi chú lại khi có các testcase thực hiện không thành công.
* Kiểm tra lại ( Retest ) sau khi các bug đã được fix.
* Theo dõi và đóng bug.

## 4.3 Kĩ thuật kiểm thử

Kỹ thuật kiểm thử hộp đen (black box testing) là một phương pháp kiểm thử phần mềm tập trung vào kiểm tra các chức năng và yêu cầu bên ngoài của một ứng dụng mà không cần biết chi tiết về cấu trúc nội bộ hay mã nguồn của phần mềm.

Các bước chính trong kỹ thuật kiểm thử hộp đen bao gồm:

* Phân tích yêu cầu: Xác định và hiểu rõ các yêu cầu và chức năng của phần mềm cần kiểm thử. Điều này giúp xác định phạm vi kiểm thử và xây dựng kịch bản kiểm thử.
* Thiết kế kịch bản kiểm thử: Dựa trên yêu cầu, thiết kế các kịch bản kiểm thử mà sẽ được thực hiện trên phần mềm. Kịch bản này bao gồm việc xác định đầu vào, thực hiện các tác vụ và đánh giá kết quả đầu ra.
* Thực hiện kiểm thử: Thực hiện các kịch bản kiểm thử theo yêu cầu đã thiết kế. Đầu vào được cung cấp và kết quả đầu ra được so sánh với kết quả mong đợi để xác định sự hoạt động đúng đắn của phần mềm.
* Phân tích kết quả: Kiểm tra và phân tích kết quả kiểm thử để xác định xem phần mềm đã đáp ứng yêu cầu và chức năng hay không. Nếu phát hiện lỗi, thông báo và báo cáo các vấn đề để có thể được khắc phục.

Các ưu điểm của kỹ thuật kiểm thử hộp đen bao gồm:

* Không cần biết chi tiết về cấu trúc nội bộ và mã nguồn của phần mềm.
* Tập trung vào giao diện và chức năng bên ngoài của phần mềm.
* Dễ áp dụng và thực hiện, không đòi hỏi kiến thức chuyên sâu về lập trình.
* Giúp xác định được các lỗi mà người dùng cuối có thể gặp phải.

## 4.4 Kiểm thử giao diện hệ thống

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Mã trường hợp kiểm thử |  | Tên chức năng | | Mục đích kiểm thử | Điều kiện kiểm thử | Các bước thực hiện | Kết quả mong đợi | Kết quả thực tế |
| TC\_01 | Đăng nhập | | Check UI | Kiểm tra giao diện đăng nhập của hệ thống | User có tài khoản để đăng nhập vào hệ thống | 1. Truy cập hệ thống 2. Quan sát màn hình | 1. Đủ số lượng các Item 2. Vị trí, font chữ, size chữ, màu sắc đúng theo yêu cầu 3. Trạng thái các Item hiển thị đúng thiết kế | pass |
| TC\_02 | Đăng nhập | | Check button Đăng nhập | Kiểm tra chức năng đăng nhập của hệ thống | User có tài khoản để đăng nhập vào hệ thống | 1. Truy cập hệ thống 2. Điền thông tin tên tài khoản, mật khẩu 3. Chọn "Đăng nhập" | Đăng nhập thành công vào hệ thống | pass |
| TC\_03 | Quản lí Nhân viên | | Thêm mới | Kiểm tra chức năng thêm mới Nhân viên | User đăng nhập thành công vào hệ thống | 1. Truy cập hệ thống thành công 2. Chọn "Quản lí người dùng" 3. Click " Thêm mới" 4. Điền các trường thông tin 5. Click " Lưu lại" | 1. Thêm mới User thành công 2. Dữ liệu được lưu vào Database | pass |
| TC\_04 | Quản lí Nhân viên | | Cập nhật | Kiểm tra chức năng cập nhật User | User đăng nhập thành công vào hệ thống | 1. Truy cập hệ thống thành công 2. Chọn "Quản lí người dùng" 3. Click "Sửa" 4. Điền các trường thông tin 5. Click " Lưu lại" | 1. Cập nhật User thành công 2. Dữ liệu được lưu vào Database | pass |
| TC\_05 | Quản lí Nhân viên | | Xóa | Kiểm tra chức năng xóa Nhân viên | User đăng nhập thành công vào hệ thống | 1. Truy cập hệ thống thành công 2. Chọn "Quản lí người dùng" 3. Click "Xóa" 4. Click " Lưu lại" | 1. Xóa User thành công 2. Dữ liệu được lưu vào Database | pass |
| TC\_06 | Quản lí danh mục | | Thêm mới | Kiểm tra chức năng thêm danh mục | User đăng nhập thành công vào hệ thống | 1. Truy cập hệ thống thành công 2. Chọn "Thêm danh mục mới" 3. Điền các thông tin 4. Click " Thêm" | 1. Thêm danh mục thành công 2. Dữ liệu được lưu vào Database | pass |
| TC\_07 | Quản lí danh mục | | Sửa | Kiểm tra chức năng sửa danh mục | User đăng nhập thành công vào hệ thống | 1. Truy cập hệ thống thành công 2. Chọn " danh sách danh mục "  3.chọn nút"sửa" 4. Điền các thông tin 5. Click " Cập nhật" | 1. Thêm danh mục thành công 2. Dữ liệu được lưu vào Database | pass |
| TC\_08 | Quản lí danh mục | | Xóa | Kiểm tra chức năng Xóa danh mục | User đăng nhập thành công vào hệ thống | 1. Truy cập hệ thống thành công 2. Chọn "danh sách danh mục" 3. Chọn "xóa" | 1. Xóa danh mục thành công 2. Dữ liệu được lưu vào Database | pass |
| TC\_09 | Quản lí thiết bị | | Thêm | Kiểm tra chức năng thêm thiết bị | User đăng nhập thành công vào hệ thống | 1. Truy cập hệ thống thành công 2. Chọn "Thêm thiết bị mới" 3. Điền các thông tin  4.Chọn "Thêm " | 1. Thêm thiết bị thành công 2. Dữ liệu được lưu vào Database | pass |
| TC\_10 | Quản lí thiết bị | | Sửa | Kiểm tra chức năng sửa thiết bị | User đăng nhập thành công vào hệ thống | 1. Truy cập hệ thống thành công 2. Chọn "tất cả thiết"  3.Chọn"Sửa"  4. Điền các thông tin  5.Chọn "cập nhật " | 1.Sửa thiết bị thành công 2. Dữ liệu được lưu vào Database | pass |
| TC\_11 | Quản lí thiết bị | | Xóa | Kiểm tra chức năng xóa thiết bị | User đăng nhập thành công vào hệ thống | 1. Truy cập hệ thống thành công 2. Chọn "tất cả thiết bị"  3.Chọn "Xóa " | 1. Xóa thiết bị thành công 2. Dữ liệu được lưu vào Database | pass |
| TC\_12 | Quản lí thêm thiết bị cho phòng ban | | Thêm | Kiểm tra chức năng thêm thiết bị cho phòng ban | User đăng nhập thành công vào hệ thống | 1. Truy cập hệ thống thành công 2. Chọn "danh sách phòng ban"  3.Chọn "Thêm thiết bị "  4.Điền các thông tin  5. Chọn "Thêm " | 1. Thêm thiết bị thành công 2. Dữ liệu được lưu vào Database | pass |
| TC\_13 | Trả thiết bị về kho | | Trả thiết bị | Kiểm tra chức năng trả thiết bị về kho | User đăng nhập thành công vào hệ thống | 1. Truy cập hệ thống thành công 2. Chọn "danh sách phòng ban"  3.Chọn "Xem "  4. Chọn "trả về kho " | 1. Trả thiết bị thành công 2. Dữ liệu được lưu vào Database | pass |
| TC\_14 | Báo hỏng thiết bị | | Báo hỏng | Kiểm tra chức báo hỏng thiết bị | User đăng nhập thành công vào hệ thống | 1. Truy cập hệ thống thành công 2. Chọn "danh sách phòng ban"  3.Chọn "Xem "  4. Chọn "báo hỏng" | 1. Trả thiết bị thành công 2. Dữ liệu được lưu vào Database | pass |
| TC\_15 | Thiết bị đã sửa xong | | Đã sửa | Kiểm tra chức Đã sửa thiết bị | User đăng nhập thành công vào hệ thống | 1. Truy cập hệ thống thành công 2. Chọn "danh sách thiết bị hỏng"  3.Chọn "đã sửa xong " | 1. Trả thiết bị thành công 2. Dữ liệu được lưu vào Database | pass |
| TC\_16 | Quản lí phòng ban | | Thêm | Kiểm tra chức năng Thêm phòng ban | User đăng nhập thành công vào hệ thống | 1. Truy cập hệ thống thành công 2. Chọn "quản lí thông tin phòng ban"  3.Chọn "Thêm "  4.Điền các thông tin  5. Chọn "Thêm " | 1. Thêm phòng ban thành công 2. Dữ liệu được lưu vào Database | pass |
| TC\_17 | Quản lí phòng ban | | Sửa | Kiểm tra chức năng Sửa phòng ban | User đăng nhập thành công vào hệ thống | 1. Truy cập hệ thống thành công 2. Chọn "quản lí thông tin phòng ban"  3.Chọn "Sửa "  4.Điền các thông tin  5. Chọn "Cập nhật " | 1. Sửa phòng ban thành công 2. Dữ liệu được lưu vào Database | pass |
| TC\_18 | Quản lí phòng ban | | Xóa | Kiểm tra chức năng Xóa phòng ban | User đăng nhập thành công vào hệ thống | 1. Truy cập hệ thống thành công 2. Chọn "quản lí thông tin phòng ban"  3.Chọn "Xóa " | 1. Xóa phòng ban thành công 2. Dữ liệu được lưu vào Database | pass |
| TC\_19 | Quản lí kho | | Thêm | Kiểm tra chức năng Thêm kho | User đăng nhập thành công vào hệ thống | 1. Truy cập hệ thống thành công 2. Chọn "quản lí thông tin kho"  3.Chọn "Thêm "  4.Điền các thông tin  5. Chọn "Thêm " | 1. Thêm kho thành công 2. Dữ liệu được lưu vào Database | pass |
| TC\_20 | Quản lí kho | | Sửa | Kiểm tra chức năng Sửa kho | User đăng nhập thành công vào hệ thống | 1. Truy cập hệ thống thành công 2. Chọn "quản lí thông tin kho"  3.Chọn "Sửa "  4.Điền các thông tin  5. Chọn "Cập nhật " | 1. Sửa kho thành công 2. Dữ liệu được lưu vào Database | Pass |
| TC\_21 | Quản lí kho | | Xóa | Kiểm tra chức năng Xóa kho | User đăng nhập thành công vào hệ thống | 1. Truy cập hệ thống thành công 2. Chọn "quản lí thông tin kho"  3.Chọn "Xóa " | 1. Xóa kho thành công 2. Dữ liệu được lưu vào Database | pass |
| TC\_22 | Quản lí NCC | | Thêm | Kiểm tra chức năng Thêm NCC | User đăng nhập thành công vào hệ thống | 1. Truy cập hệ thống thành công 2. Chọn "quản lí thông tin NCC"  3.Chọn "Thêm "  4.Điền các thông tin  5. Chọn "Thêm " | 1. Thêm NCC thành công 2. Dữ liệu được lưu vào Database | pass |
| TC\_23 | Quản lí NCC | | Sửa | Kiểm tra chức năng Sửa NCC | User đăng nhập thành công vào hệ thống | 1. Truy cập hệ thống thành công 2. Chọn "quản lí thông tin NCC"  3.Chọn "Sửa "  4.Điền các thông tin  5. Chọn "Cập nhật " | 1. Sửa NCC thành công 2. Dữ liệu được lưu vào Database | Pass |
| TC\_24 | Quản lí NCC | | Xóa | Kiểm tra chức năng Xóa NCC | User đăng nhập thành công vào hệ thống | 1. Truy cập hệ thống thành công  2. Chọn "quản lí thông tin NCC"  3.Chọn "Xóa " | 1. Xóa NCCthành công 2. Dữ liệu được lưu vào Database | Pass |
| TC\_25 | Quên mật khẩu | | Check IU | Kiểm tra chức năng quên mật khẩu | User có tài khoản để đăng nhập vào hệ thống | 1.Click "Quên mật khẩu trên giao diện trang đăng nhập "  2.Điền các thông tin  3.Chọn "Trình " hoặc "Đăng nhập " | 1. Thay đổi mật khẩu thành công 2. Dữ liệu được lưu vào Database | Pass |

# 

# KẾT LUẬN

Trong quá trình thực hiện đồ án **“Xây dựng website quản lý thiết bị công nghệ cho doanh nghiệp LDM“**, em đã vận dụng được kiến thức từ những môn học bổ ích trên trường đồng thời vận dụng kiến thức tại Trường Đại học Công nghiệp Hà Nội – nơi mà em đã trải qua quá trình học và thực tập để có cơ hội áp dụng những kiến thức đã học được vào thực tiễn: framework laravel của PHP, các kiến thức về thuật toán, cách xây dựng hệ thống Website, rèn luyện các kiến thức và kỹ năng phát triển hệ thống phần mềm và thái độ làm việc chuyên nghiệp.

Hệ thống được xây dựng với mục đích giúp công ty quản lý tài sản thiết bị hiệu quả hơn, cải thiện quy trình làm việc của nhân viên và giảm thiểu các sai sót không đáng có trong quá trình quản lý thiết bị.

Kết quả đạt được: xây dựng được một Website quản lý các thiết bị như: laptop, pc, các thiết bị mạng… của công ty lập trình, gia công phần mềm với các chức năng đề ra.

Những hạn chế của hệ thống: Website vận hành theo mô hình server-side rendering khiến cho Trang web phải xử lý lại hoàn toàn và load lại từ đầu nếu chỉ có một thay đổi nhỏ trong nội dung. Việc xử lý nội dung HTML khiến hao tốn tài nguyên server, gây chậm trễ khi xử lý các request khác.

Hướng phát triển tiếp theo của hệ thống: nâng cấp các tính năng đang có cho hoàn thiện tối ưu nhất; thêm một số tính năng, nâng cao trải nghiệm người dùng như: gợi ý thiết bị dựa vào hành vi của người dùng trên website, tối ưu tốc độ trang web.

|  |  |
| --- | --- |
|  |  |

# TÀI LIỆU THAM KHẢO

1. **Trần Phương Nhung, Nguyễn Trung Phú**, *Giáo trình Lập trình ứng dụng cơ sở dữ liệu trên web*, Trường Đại Học Công nghiệp Hà Nội, 2019.
2. **Đỗ Ngọc Sơn, Phan Văn Viên, Nguyễn Phương Nga**, *Giáo trình Hệ quản trị cơ sở dữ liệu,* Tổ Khoa học và Kỹ thuật, Trường Đại học Công nghiệp Hà Nội, 2015.
3. **Nguyễn Tấn Trường**, *Hướng Dẫn Thực Hành PHP Và MySQL*, Nhà xuất bản thanh niên, 2015.
4. **Nguyễn Thị Thanh Huyền**, **Ngô Thị Bích Thúy, Phạm Thị Kim Phượng**, *Giáo trình phân tích thiết kế hệ thống*, NXB Giáo dục Việt Nam, 2011.
5. ***Vũ Thị Dương, Phùng Đức Hòa, Nguyễn Thị Hương Lan****, Giáo trình Phân tích thiết kế hướng đối tượng,* NXB Khoa học và Kỹ thuật, 2015.