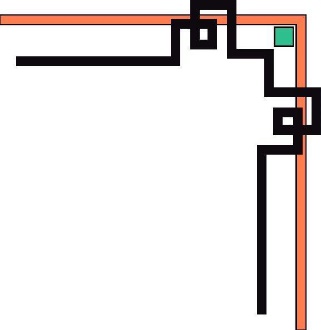
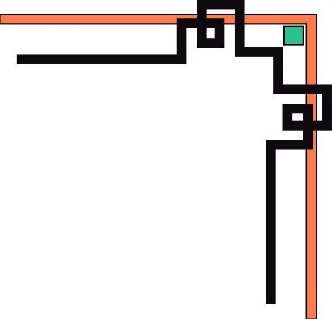
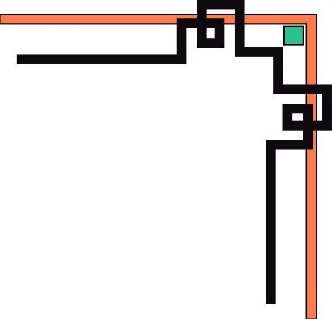
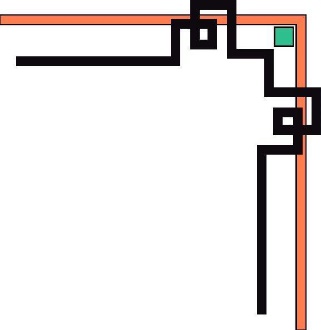
**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM**



**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**🕯✡🕮🕮✡🕯**

**MÔN HỌC: LẬP TRÌNH WEB**

**BÁO CÁO ĐỒ ÁN MÔN HỌC**

**HỆ THỐNG QUẢN LÝ**

**WEBSITE BẤT ĐỘNG SẢN**

**GVHD:** ThS. Nguyễn Minh Đạo

**SVTH: MSSV**

Đào Xuân Thủy 16110544

Ngô Công An 16110002

Nguyễn Út Thiện 16110573

**Tp. Hồ Chí Minh, tháng 12 năm 2018**

**MỤC LỤC**

[GIỚI THIỆU ĐỀ TÀI 1](#_Toc533369131)

[1. Lý do chọn đề tài và mục tiêu hướng tới 1](#_Toc533369132)

[2. Phương pháp nghiên cứu 1](#_Toc533369133)

[3. Chức năng cơ bản 1](#_Toc533369134)

[3.1 Người quản lý 1](#_Toc533369135)

[4. Use case diagram 1](#_Toc533369136)

[CƠ SỞ DỮ LIỆU 4](#_Toc533369137)

[1. Mô hình ERD 4](#_Toc533369138)

[2. Database diagram 5](#_Toc533369139)

[GIAO DIỆN 6](#_Toc533369140)

[1. Giao diện quản lí 6](#_Toc533369141)

[MÔ HÌNH - FRAMEWORK THỰC HIỆN ĐỀ TÀI 13](#_Toc533369142)

[1. Mô hình xây dựng dự án 13](#_Toc533369143)

[1) Spring boot 13](#_Toc533369144)

[2) Spring data JPA và Hibernate 14](#_Toc533369145)

[3) Spring security 16](#_Toc533369146)

[4) My SQL 16](#_Toc533369147)

[ĐÁNH GIÁ 18](#_Toc533369148)

[1. Quá trình thực hiện ứng dụng 18](#_Toc533369149)

[2. Ưu – Nhược điểm khi ứng dụng 18](#_Toc533369150)

[3. Hướng phát triển của đề tài 18](#_Toc533369151)

# 

# GIỚI THIỆU ĐỀ TÀI

## Lý do chọn đề tài và mục tiêu hướng tới

Hiện nay việc mua bán bất động sản ngày càng tăng và với thời đại công nghệ 4.0 hiện nay việc sử dụng điện thoại, laptop ngày càng tăng cao… nắm bắt được nhu cầu đó nhóm chúng em quyết định thực hiện đồ án web là web bất động sảnvới các chức năng chính như sau: Quản lý các bài đăng, tạo bài đăng, quản lý nhân viên, tạo nhân viên, quản lý khách hàng và tạo khách hàng.

## Phương pháp nghiên cứu

* Công cụ lập trình: IntelliJ IDEA 2018, Git.
* Ngôn ngữ lập trình: java.
* Hệ quản trị cơ sở dữ liệu: My SQL.

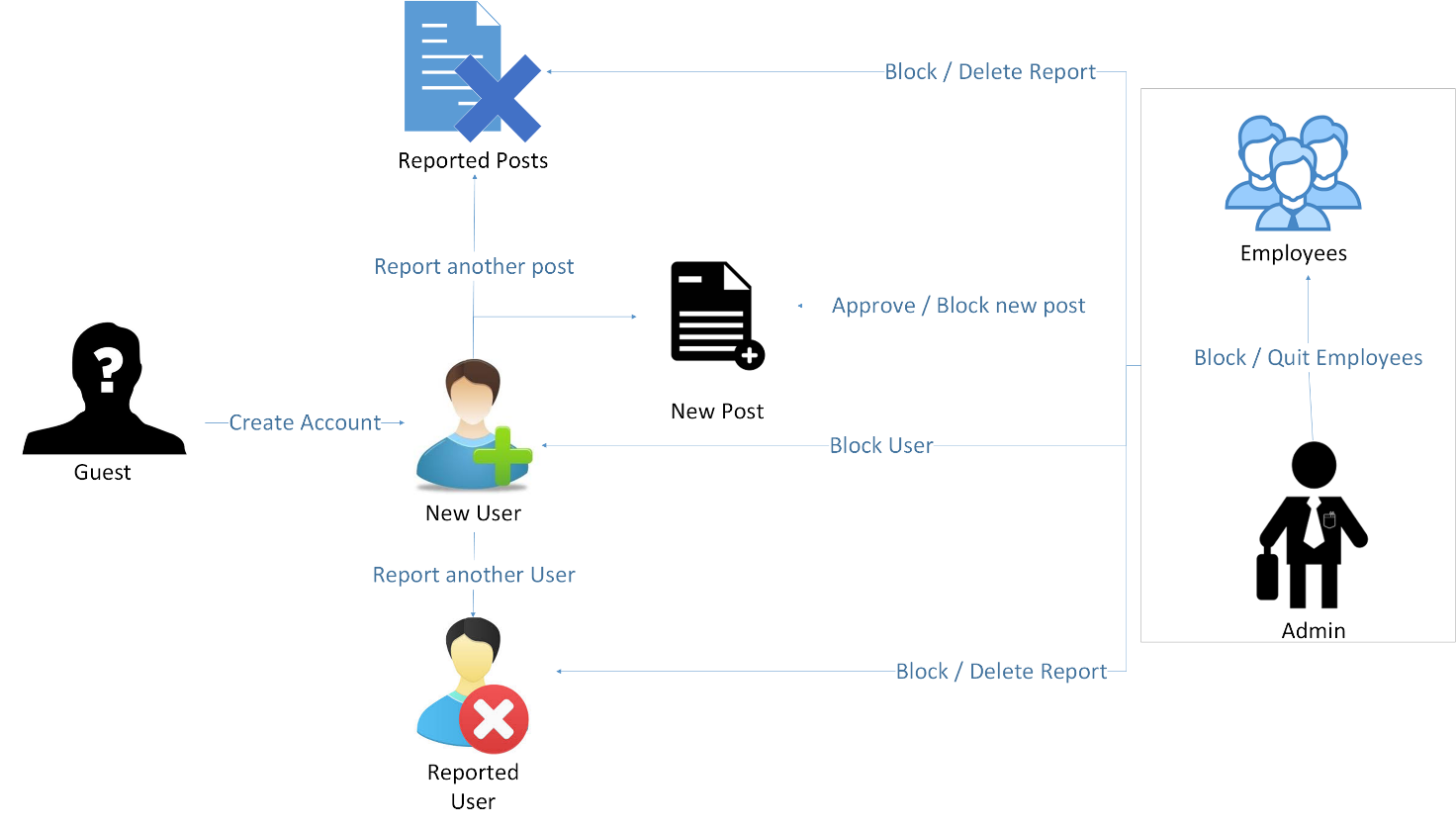
## Chức năng cơ bản

Các chức năng của ứng dụng được phân làm ba nhóm chính dựa vào ba đối tượng là người dùng, nhân viên và bài đăng, cụ thể như sau:

### Người quản lý

* Quản lý bài đăng của người dùng
  + Thêm bài đăng.
  + Khóa bài đăng.
  + Duyệt bài đăng mới.
  + Quản lý những bài đăng bị người dùng báo cáo.
* Quản lý nhân viên
* Tạo nhân viên mới.
* Quản lý tài khoản nhân viên.
* Xem/Chỉnh sửa thông tin chi tiết nhân viên.
* Quản lý người dùng
* Tạo người dùng mới.
* Xem/Chỉnh sửa thông tin chi tiết người dùng.

## Use case diagram



Hình 1. Use case diagram

Bảng 1. Bảng mô tả các actor trong use case diagram

|  |  |  |
| --- | --- | --- |
| TT | Tên Actor | Ý nghĩa |
| 1 | Guest | Người dùng truy cập trang web, sử dụng trang web để tra cứu về bất động sản nhưng chưa có tài khoản. |
| 2 | New User | Người dùng đã đăng ký tài khoản trên hệ thống và có thể đăng bài. |
| 3 | New Post | Bài đăng mới được người dùng tạo. |
| 4 | Reported Posts | Danh sách bài đăng bị người dùng báo cáo sai phạm. |
| 5 | Reported Users | Danh sách người dùng bị người dùng khác báo cáo sai phạm. |
| 6 | Employees | Nhân viên kiểm duyệt bài đăng, quản lý người dùng. |

Bảng 2. Bảng mô tả các use case trong use case diagram

|  |  |  |
| --- | --- | --- |
| TT | Tên use case | Ý nghĩa |
| 1 | Create Account | Người dùng truy cập trang web có thể đăng ký tài khoản với hệ thống để sử dụng các chức năng đặc biệt: tạo bài đăng, báo cáo bài đăng, báo cáo người dùng khác. |
| 2 | Report another post | Người dùng có thể báo cáo bài đăng của người khác nếu bài đăng đó sai phạm hoặc lừa đảo. |
| 4 | Block Report | Admin có thể xóa báo cáo hay khóa tạm thời hoặc vĩnh viễn người dùng/ bài đăng bị báo cáo. |
| 5 | Approve / Block new post | Admin có thể duyệt hoặc khóa bài đăng mới của người dùng. |

# CƠ SỞ DỮ LIỆU

## Mô hình ERD

## 

Hình 1. Mô hình ERD

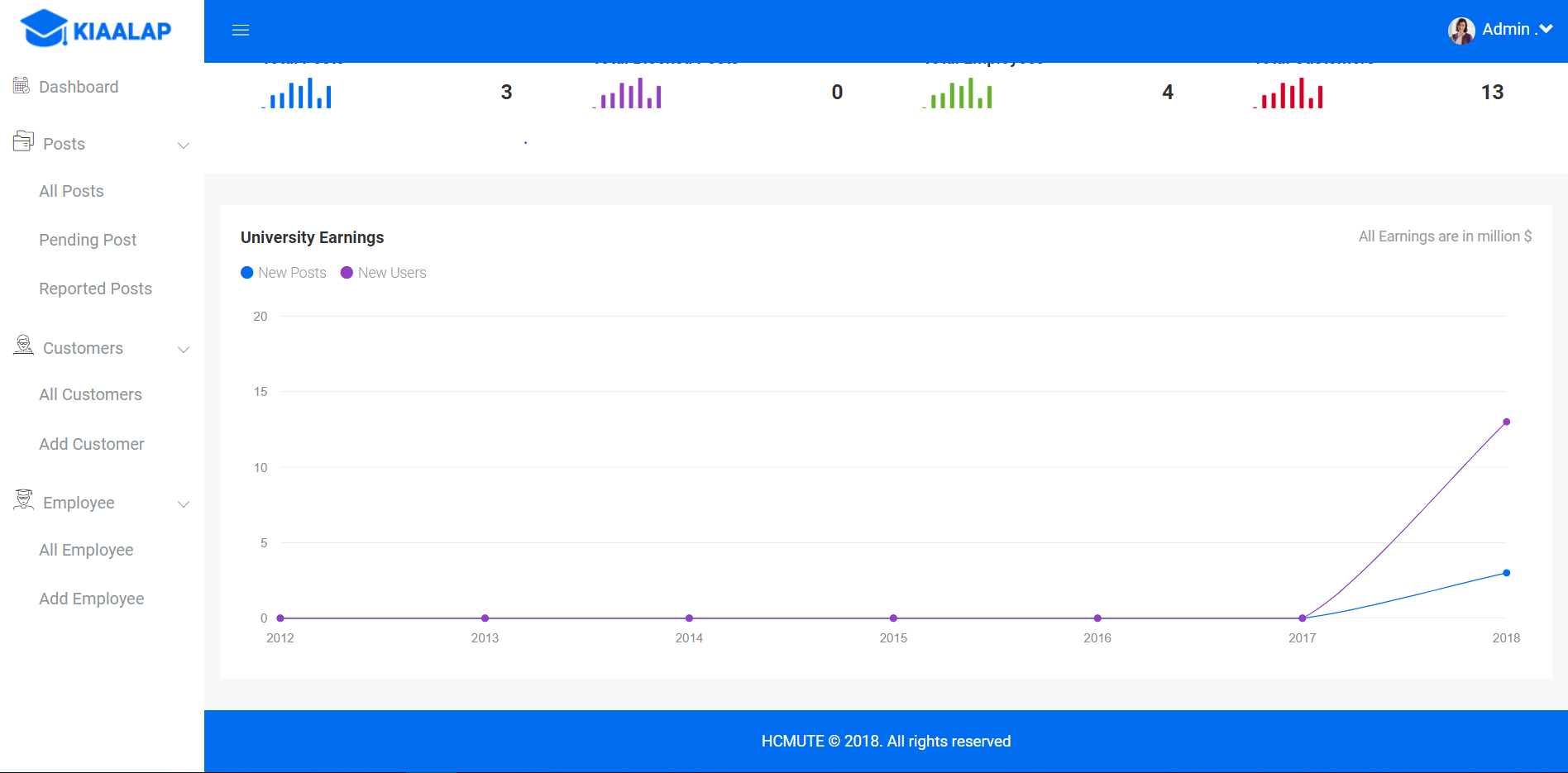
## Database diagram

Hình 2. Mô hình cơ sở dữ liệu

# 

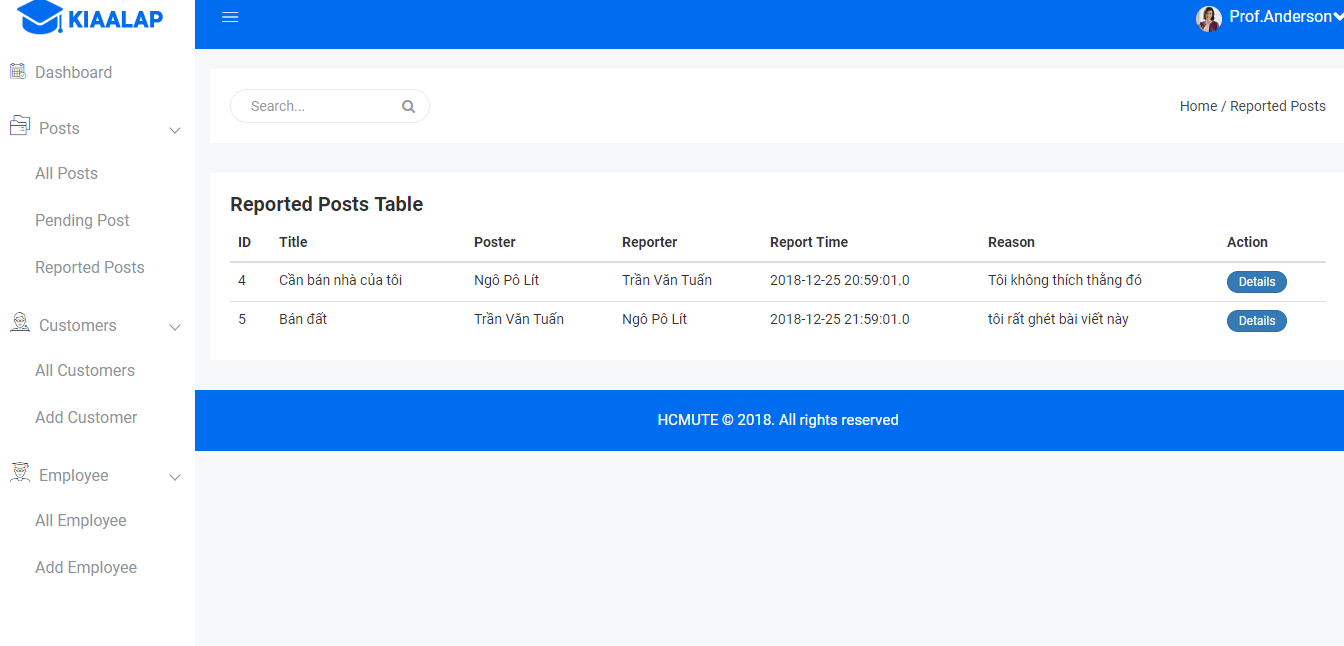
# GIAO DIỆN

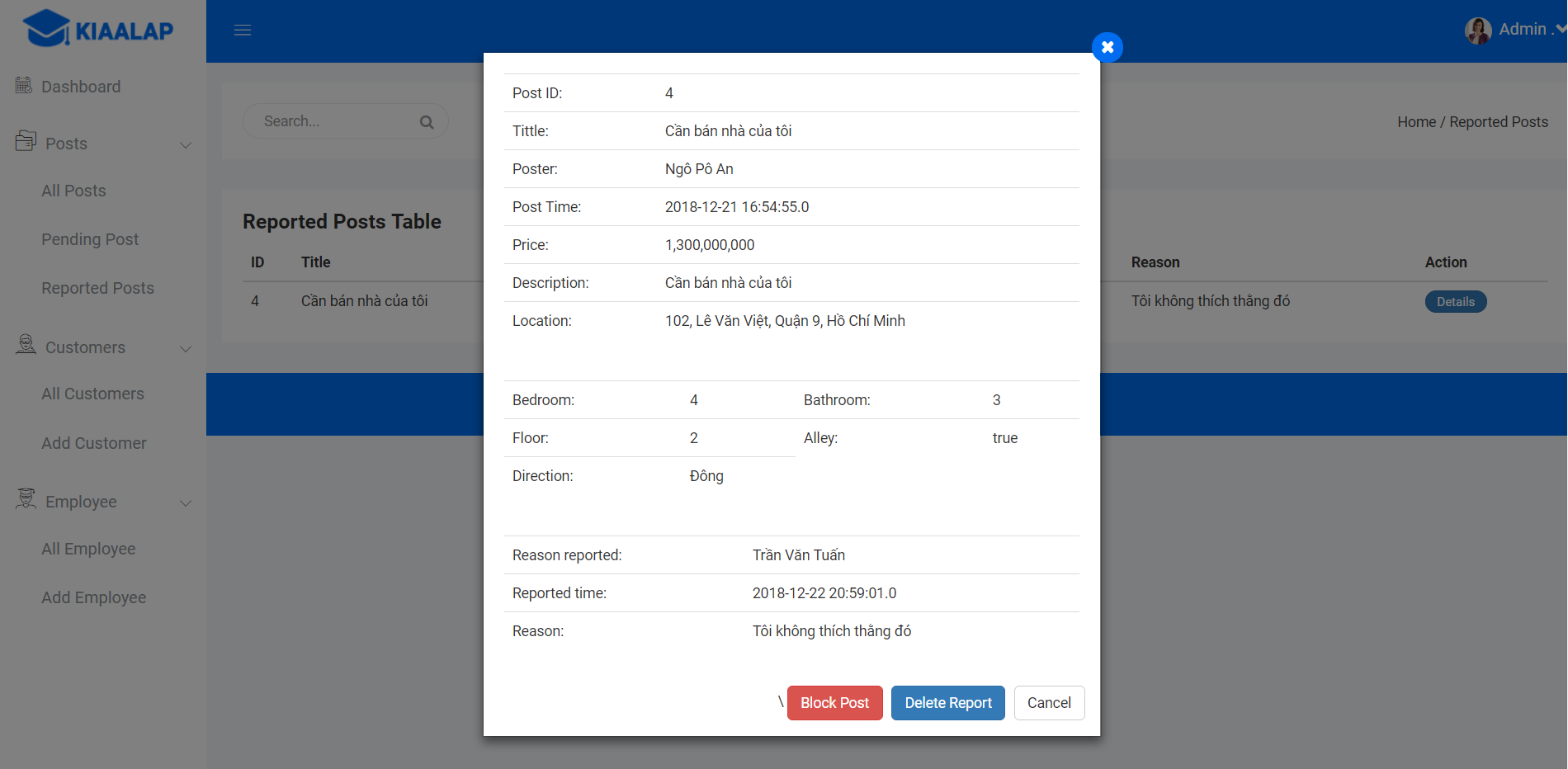
## Giao diện quản lí



Hình 3. Giao diện trang quản lí

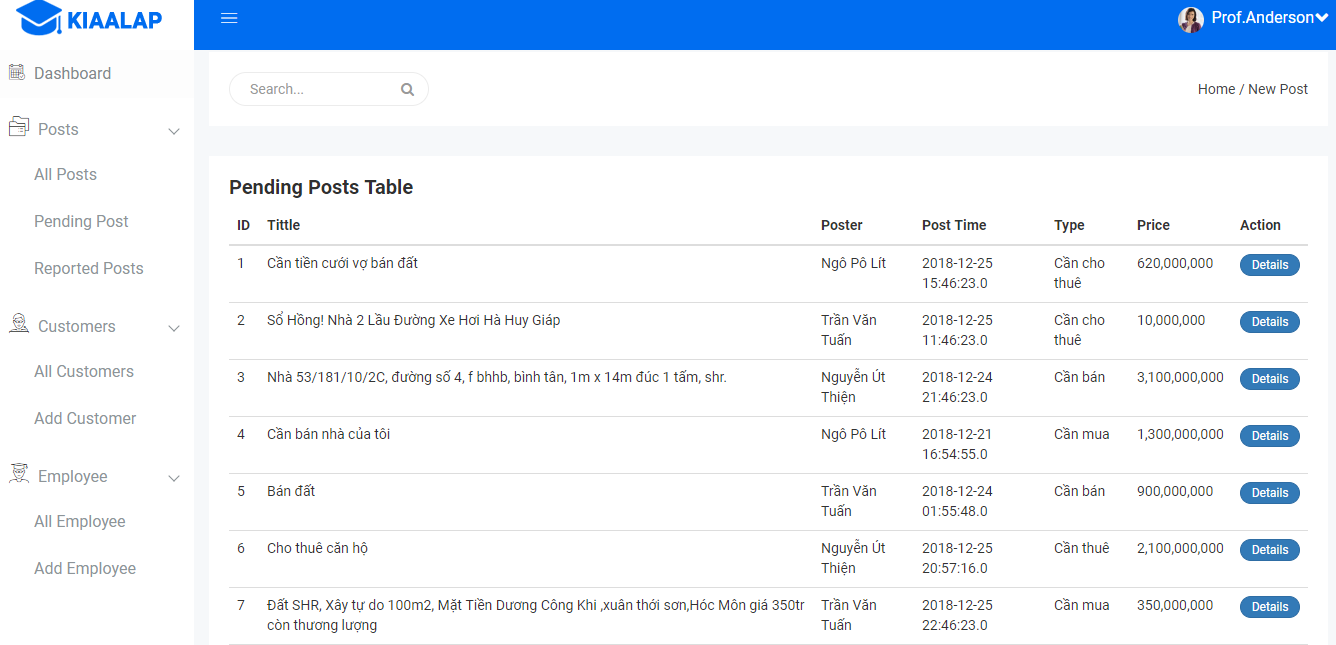
* Giải thích: trong giao diện này sẽ giúp ta có thể xem chi tiết trên lược đồ có bao nhiêu bài đăng mới và user mới

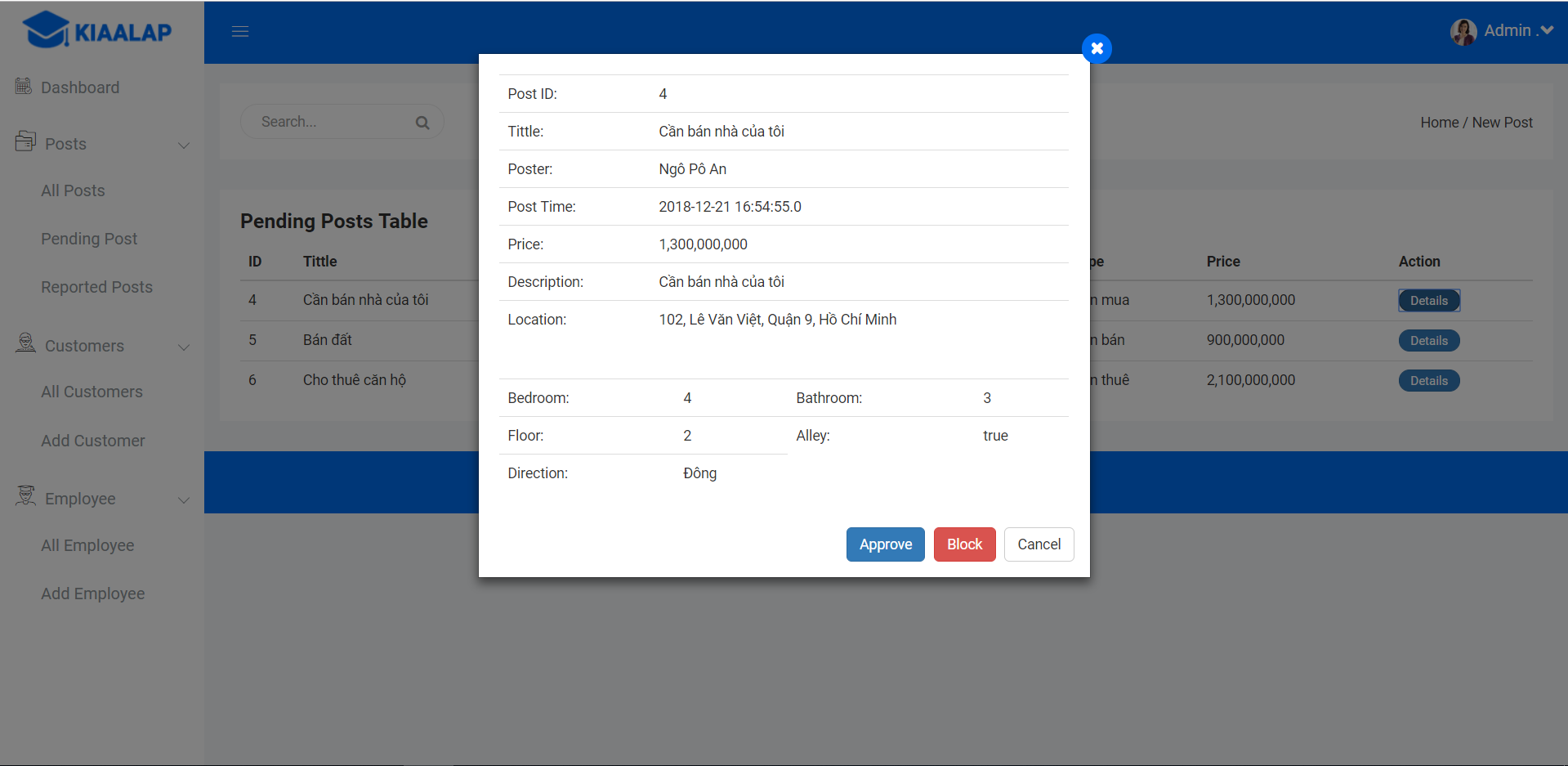
 Hình 4. Giao diện reported post (1)



Hình 4.1. Giao diện reported post (2)

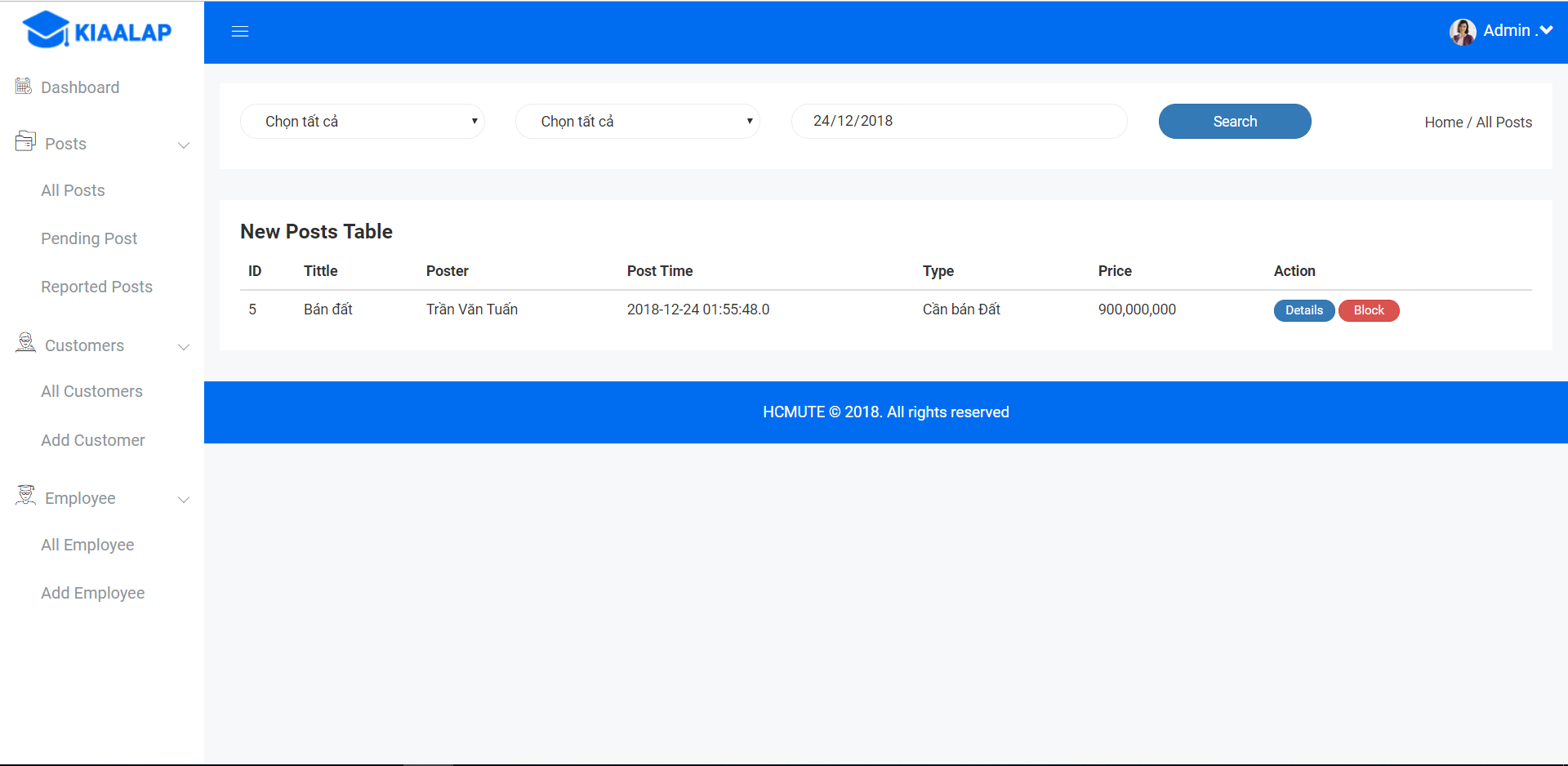
* Giải thích: có thể nhấn vô để xem chi tiết bài đăng, ta có thể block hoặc xóa bài đăng khi ta cần

Hình 5. Giao diện pending post (1)

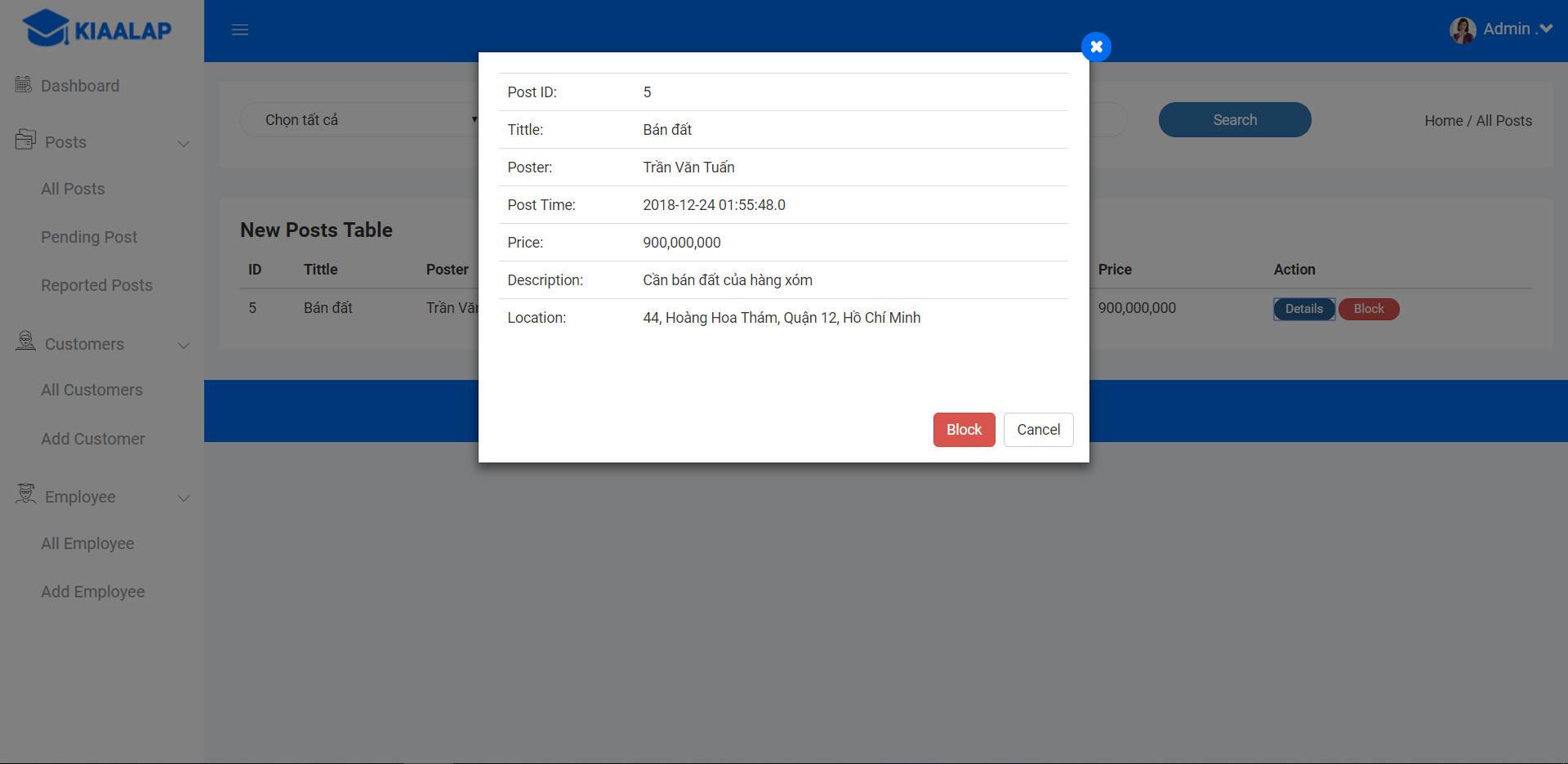


Hình 5.1. Giao diện pending post (2)

* Giải thích: ta có thể xem chi tiết, có thể duyệt và block bài đăng

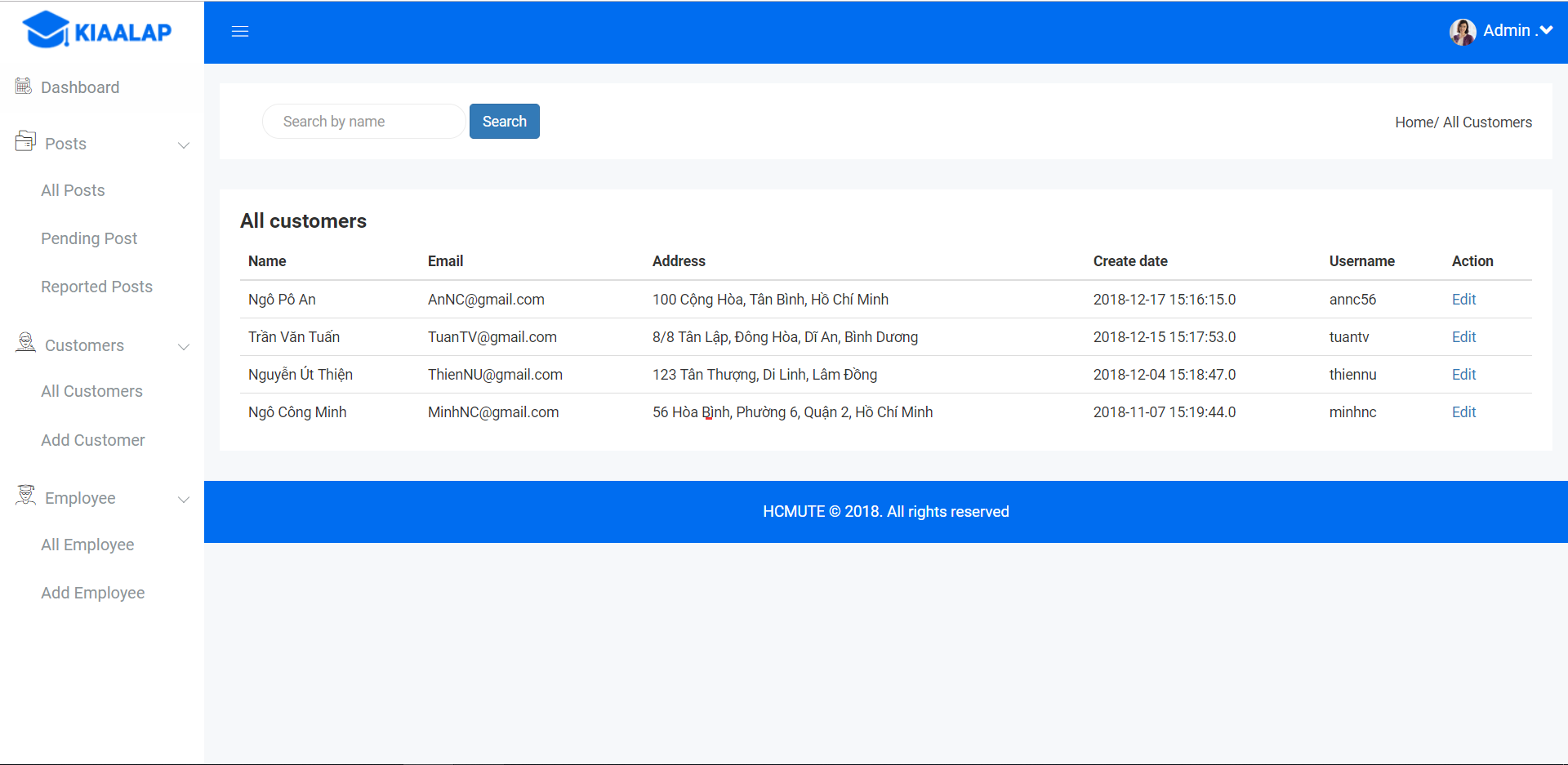


Hình 6. Giao diện all post (1)



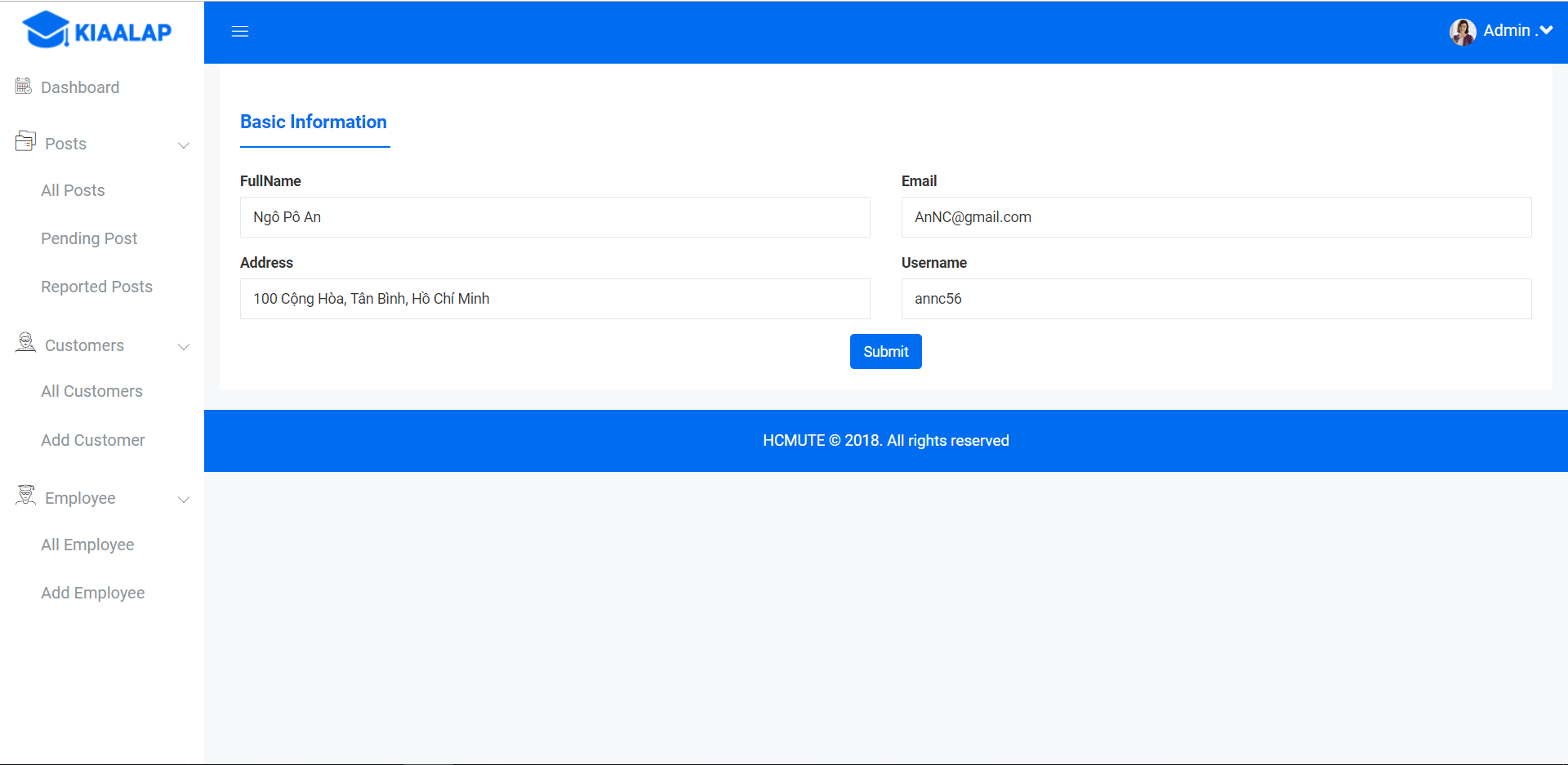
Hình 6.1. Giao diện all post (2)

* Giải thích: quản lí các bài đăng có thể xóa, block, duyệt bài đăng



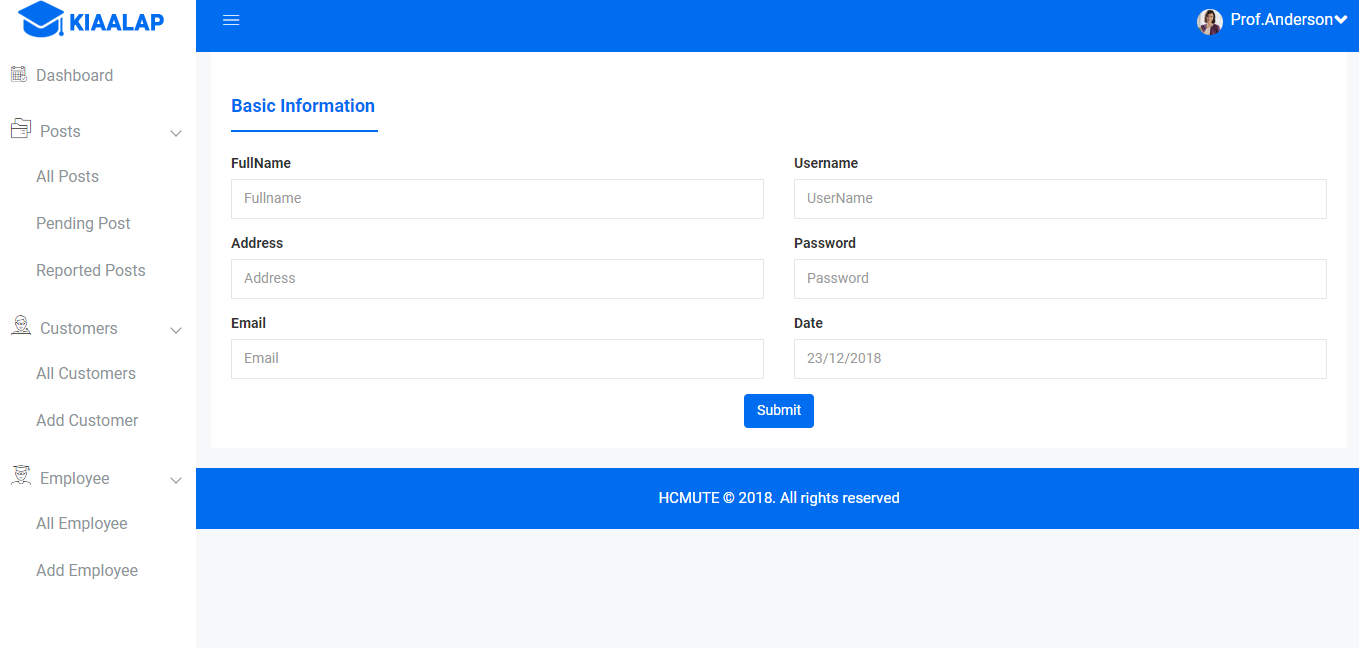
Hình 7. Giao diện all customers

* Giải thích: quản lí xem thông tin customers



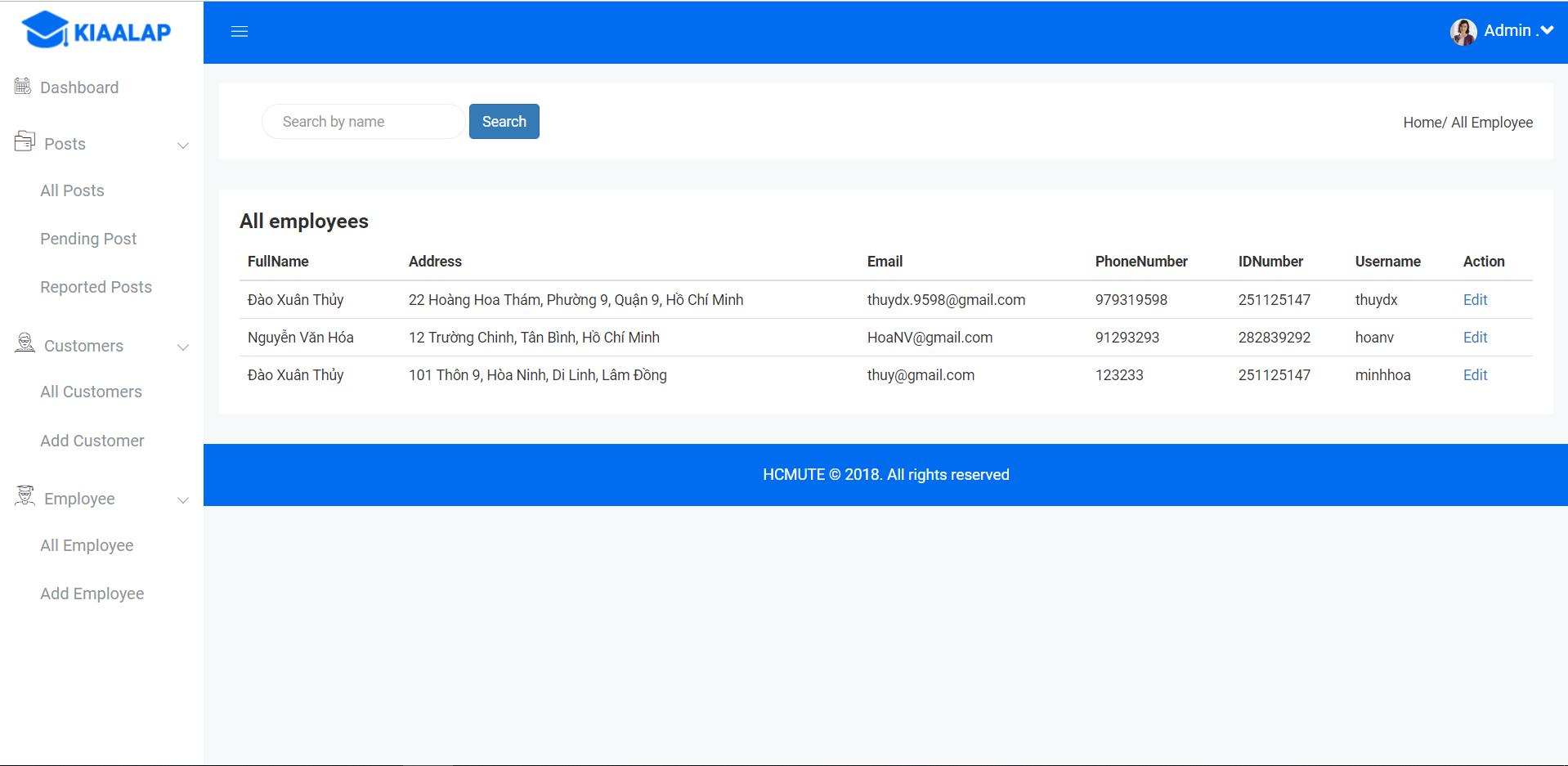
Hình 8. Giao diện edit customers

* Giải thích: chỉnh sửa thông tin customers



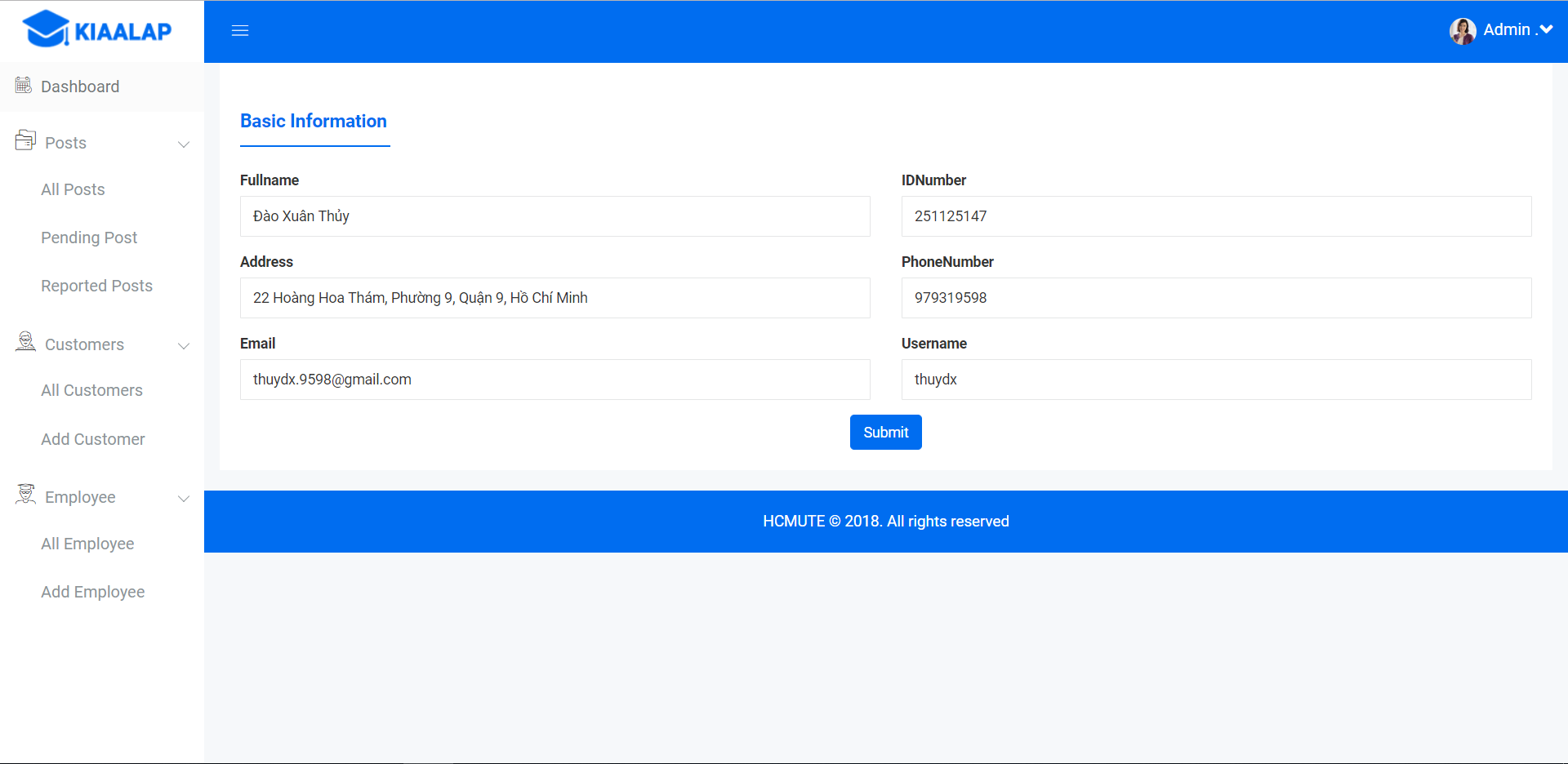
Hình 9. Giao diện add customer

* Giải thích: tạo mới customers



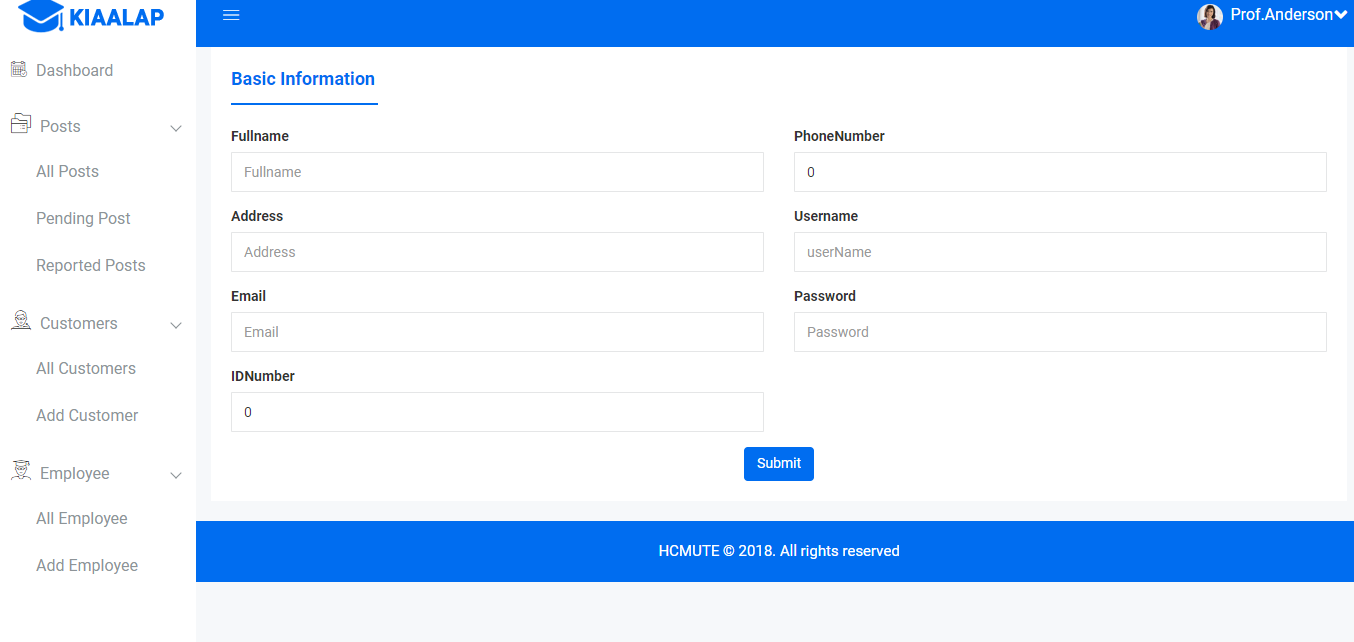
Hình 10. Giao diện all employee

* Giải thích: quản lí xem thông tin employee

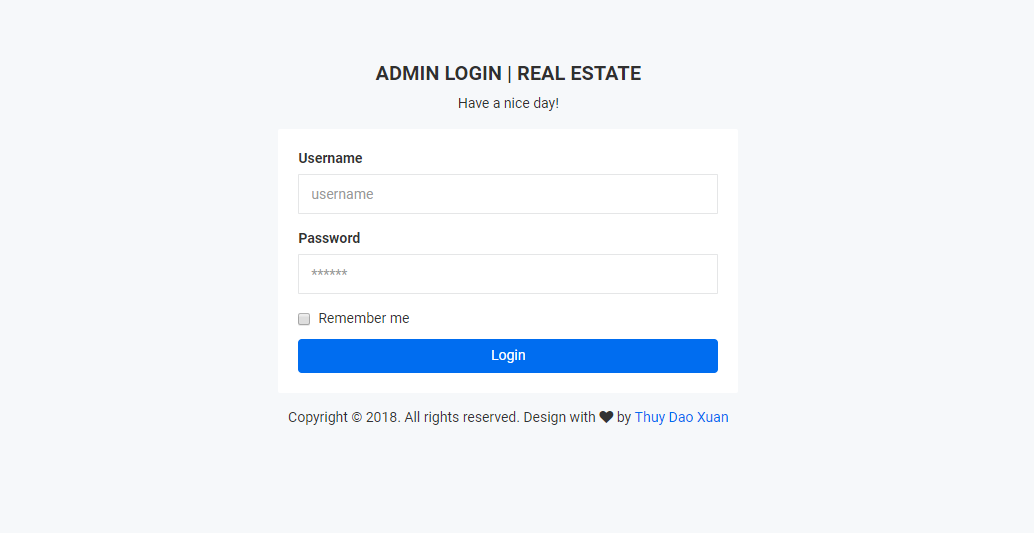


Hình 11. Giao diện edit employee

* Giải thích: chỉnh sửa thông tin employee

 Hình 12. Giao diện add employee

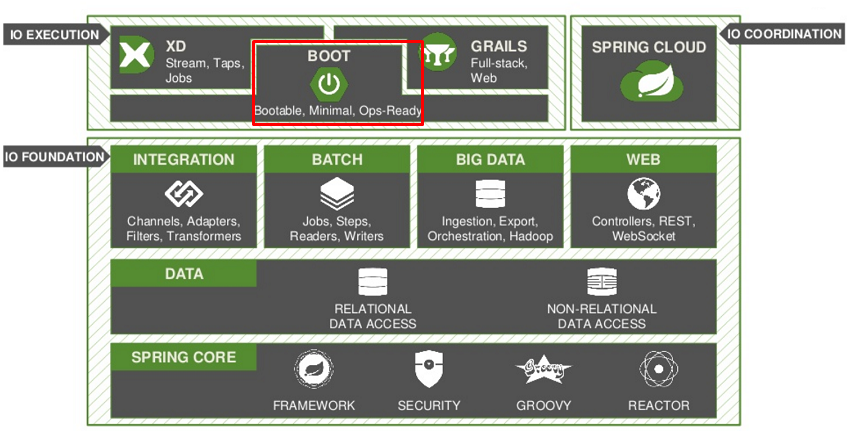
* Giải thích: tạo mới employee

Hình 13. Giao diện trang login

# MÔ HÌNH - FRAMEWORK THỰC HIỆN ĐỀ TÀI

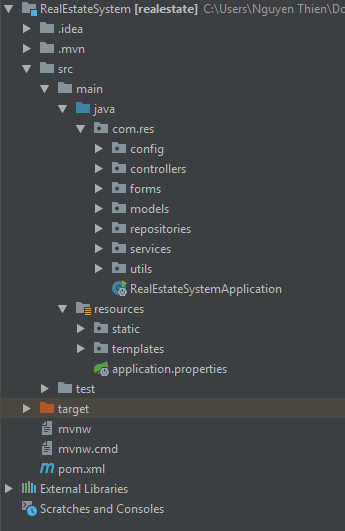
## Mô hình xây dựng dự án

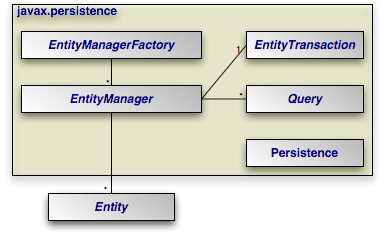
* Dự án được xây dựng dưới mô hình spring boot

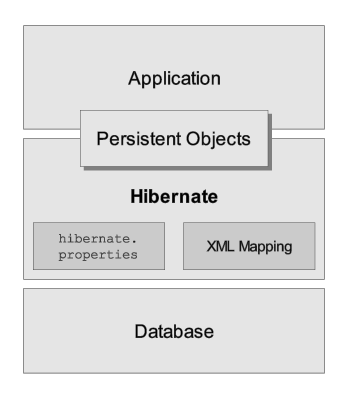
1. Spring boot

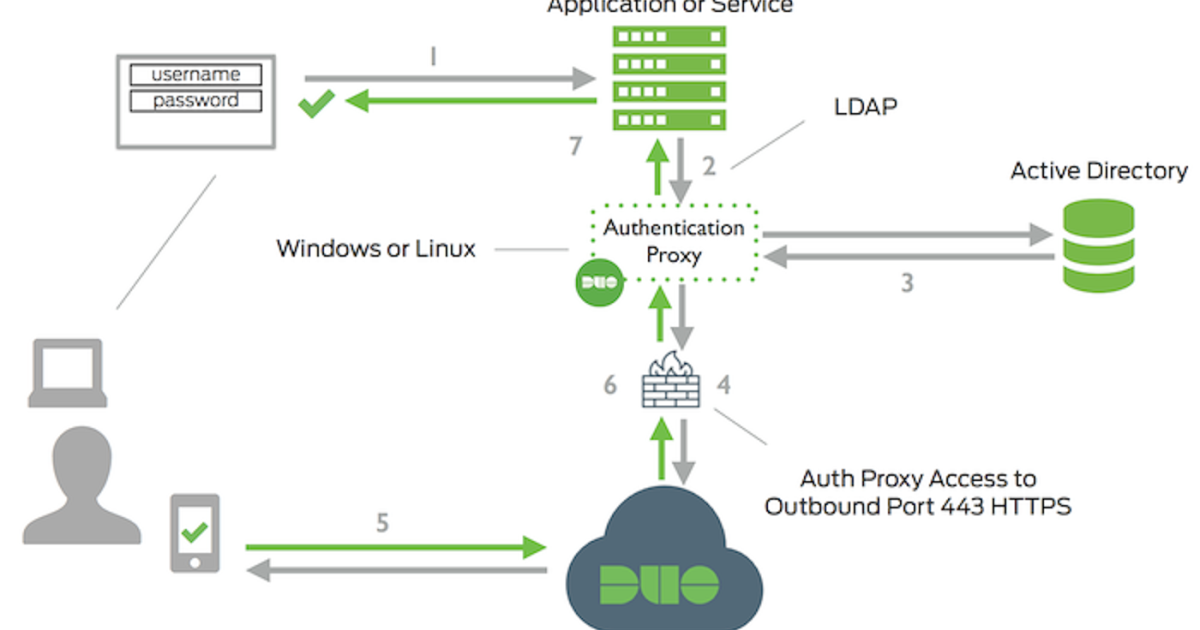
Hình 14. Mô hình spring boot

* Spring Boot là một module của Spring Framework, cung cấp tính năng RAD (Rapid Application Development) – Phát triển ứng dụng nhanh.
* Spring Boot được dùng để tạo các ứng dụng độc lập dựa trên Spring.
* Spring Boot không yêu cầu cấu hình XML
* Nó là một chuẩn cho cấu hình thiết kế phần mềm, tăng cao năng suất cho developer.

 Hình 15. Cấu trúc cây thư mục của dự án

1. Spring data JPA và Hibernate

* JPA (Java Persistence API) là 1 giao diện lập trình ứng dụng Java, nó mô tả cách quản lý các mối quan hệ dữ liệu trong ứng dụng sử dụng Java Platform.
* JPA cung cấp một mô hình POJO persistence cho phép ánh xạ các table/các mối quan hệ giữa các table trong database sang các class/mối quan hệ giữa các object.
* Ví dụ: table Users với các column (Id, name, age…) sẽ tương ứng với class Users.java với các field Id, name, age… từ đó mỗi khi truy vấn table hay các column ta sẽ truy vấn trực tiếp trên các class, các field của class mà không cần quan tâm tới việc đang dùng loại database nào, dữ liệu database ra sao…
* [Hibernate](https://stackjava.com/framework/hibenrate-framework/series-hibernate-phan-1-gioi-thieu-ve-hibernate-framework.html) là 1 ORM (Object Relational Mapping) framework cho phép người lập trình thao tác với database một cách hoàn toàn tự nhiên thông qua các đối tượng. Lập trình viên hoàn toàn không cần quan tâm đến loại database sử dụng, SQL…
* Hay nói cách khác, hibernate chính là cài đặt của JPA (JPA là 1 tập các interface, còn Hibernate implements các interface ấy 1 cách chi tiết).

1. Spring security

* Spring Security là một framework tập trung vào việc cung cấp khả năng xác thực và phân quyền cho ứng dụng java.
* Cơ chế hoạt động của Spring Security.
* Ở phần này mình sẽ nói về cơ chế làm việc của spring mvc security.
* Spring security sẽ tự động tạo form đăng nhập nếu ta không định nghĩa cho nó, sau khi đăng nhập một đối tượng user sẽ được lưu trong session, đối tượng user này sẽ gồm các thông tin như username, password, các quyền…
* Lợi ích của Spring Security
* Lợi ích lớn nhất của Spring Security là giúp bạn tích hợp tính năng xác thực và phân quyền một cách dễ dàng vào ứng dụng của mình.
* Ngoài ra Spring Security còn giúp:
  + Chống lại CSRF attack Bảo vệ Session Fixation Mã hóa mật khẩu. Cache control X-XSS-Protection …
  + Ở loạt bài này mình sẽ làm các ví dụ Spring Security với **Spring MVC - Security sử dụng XML config**

1. My SQL

* MySQL là [hệ quản trị cơ sở dữ liệu](https://vi.wikipedia.org/wiki/H%E1%BB%87_qu%E1%BA%A3n_tr%E1%BB%8B_c%C6%A1_s%E1%BB%9F_d%E1%BB%AF_li%E1%BB%87u) [tự do nguồn mở](https://vi.wikipedia.org/wiki/Ph%E1%BA%A7n_m%E1%BB%81m_ngu%E1%BB%93n_m%E1%BB%9F) phổ biến nhất thế giới và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng. Vì MySQL là 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](https://vi.wikipedia.org/wiki/Microsoft_Windows), [Linux](https://vi.wikipedia.org/wiki/Linux), [Mac OS X](https://vi.wikipedia.org/wiki/Mac_OS_X), [Unix](https://vi.wikipedia.org/wiki/Unix), [FreeBSD](https://vi.wikipedia.org/wiki/FreeBSD), [NetBSD](https://vi.wikipedia.org/w/index.php?title=NetBSD&action=edit&redlink=1), [Novell NetWare](https://vi.wikipedia.org/w/index.php?title=Novell_NetWare&action=edit&redlink=1), [SGI Irix](https://vi.wikipedia.org/w/index.php?title=SGI_Irix&action=edit&redlink=1), [Solaris](https://vi.wikipedia.org/wiki/Solaris), [SunOS](https://vi.wikipedia.org/w/index.php?title=SunOS&action=edit&redlink=1),..
* 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 Java, NodeJs, [PHP](https://vi.wikipedia.org/wiki/PHP), [Perl](https://vi.wikipedia.org/wiki/Perl), và nhiều ngôn ngữ khác, làm nơi lưu trữ những thông tin trên các trang web viết bằng Java, NodeJs, PHP hay Perl,...

# ĐÁNH GIÁ

## Quá trình thực hiện ứng dụng

Thuận lợi:

* Nhiều công nghệ hỗ trợ cho việc phát triển ứng dụng.
* Nhiều thư viện tiện ích.
* Mô hình quản lý thực tế.
* Nguồn tài liệu phong phú.
* Bản thân sinh viên có kiến thức cơ bản về lập trình Java.
* Thời gian thiết kế và phát triển ứng dụng nhiều.

Khó khăn:

* Tìm hiểu về công nghệ và hiện thực trong thời gian ngắn.
* Chưa tận dụng, tối ưu chức năng các thư viện, công nghệ.

## Ưu – Nhược điểm của ứng dụng

Ưu điểm:

* Có những chức năng cơ bản.
* Chạy trên nhiều nền tảng.

Nhược điểm:

* Cơ sở dữ liệu chưa ràng buộc chặt chẽ.
* Load dữ liệu và giao diện còn chậm.

## Hướng phát triển của đề tài

* Thêm một số bảng mở rộng dự án.
* Ràng buộc chặt chẽ hơn.
* Mở rộng chức năng cho admin và người dùng.
* Cải tiến giao diện.