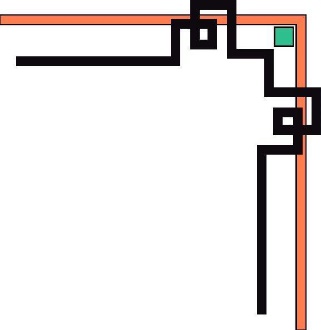
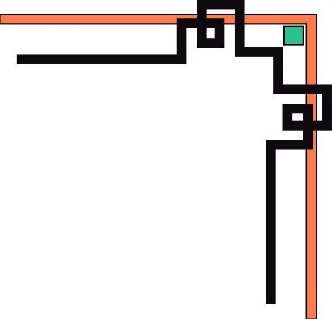
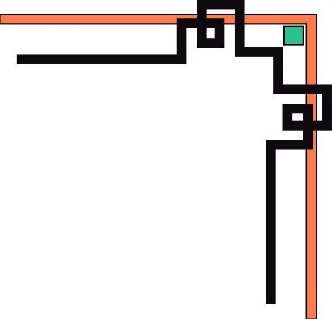
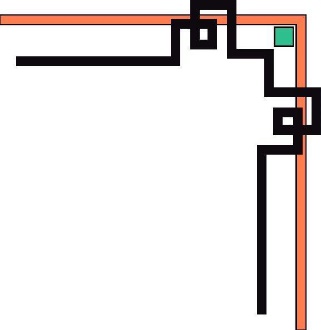
**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM**



**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**🕯✡🕮🕮✡🕯**

**MÔN HỌC: HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**

**BÁO CÁO ĐỒ ÁN MÔN HỌC**

**WEBSITE BẤT ĐỘNG SẢN**

**GVHD:** ThS. Hoàng Long

**SVTH: MSSV**

Đào Xuân Thủy 16110544

Ngô Công An 16110002

Nguyễn Út Thiện 16110573

Nguyễn Thiên Quốc 16110191

Phan Thanh Nam 16110162

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

**MỤC LỤC**

[DANH MỤC CÁC HÌNH 3](#_Toc532383210)

[DANH MỤC CÁC BẢNG 5](#_Toc532383211)

[GIỚI THIỆU ĐỀ TÀI 6](#_Toc532383212)

[1. Lý do chọn đề tài và mục tiêu hướng tới 6](#_Toc532383213)

[2. Phương pháp nghiên cứu 6](#_Toc532383214)

[3. Chức năng cơ bản 6](#_Toc532383215)

[3.1 Người dùng 6](#_Toc532383216)

[3.2 Nhân viên 6](#_Toc532383217)

[3.3 Người quản lý 6](#_Toc532383218)

[4. Use case diagram 7](#_Toc532383219)

[CƠ SỞ DỮ LIỆU 10](#_Toc532383220)

[1. Mô hình ERD 10](#_Toc532383221)

[2. Database diagram 10](#_Toc532383222)

*[3.](#_Toc532383223)* [Stored Procedures 10](#_Toc532383223)

[4. Một số trigger nổi bật 10](#_Toc532383224)

[5. Transaction 12](#_Toc532383225)

[6. Concurrency control 13](#_Toc532383226)

[7. Index 13](#_Toc532383227)

[GIAO DIỆN 15](#_Toc532383228)

[1. Giao diện người dùng 15](#_Toc532383229)

[2. Giao diện nhân viên kiểm duyệt 18](#_Toc532383230)

[3. Giao diện quản lý 21](#_Toc532383231)

[MÔ HÌNH - FRAMEWORK THỰC HIỆN ĐỀ TÀI 27](#_Toc532383232)

[1. Mô hình xây dựng dự án 27](#_Toc532383233)

[2. Framework sử dụng trong dự án 28](#_Toc532383234)

[ĐÁNH GIÁ 30](#_Toc532383235)

[1. Quá trình thực hiện ứng dụng 30](#_Toc532383236)

[2. Ưu – Nhược điểm khi ứng dụng 30](#_Toc532383237)

[3. Hướng phát triển của đề tài 30](#_Toc532383238)

# DANH MỤC CÁC HÌNH

[Hình 1. Use case diagram 6](#_Toc529307703)

[Hình 2. Giao diện trang chủ của người dùng 14](#_Toc529307703)

[Hình 3. Giao diện đăng nhập của người dùng 14](#_Toc529307704)

[Hình 4. Giao diện đăng ký của người dùng 15](#_Toc529307705)

[Hình 5. Giao diện đổi mật khẩu của người dùng 15](#_Toc529307706)

[Hình 6. Giao diện tìm kiếm bài đăng của người dùng 16](#_Toc529307706)

[Hình 7. Giao diện đăng bài của người dùng (ảnh 1) 16](#_Toc529307706)

[Hình 8. Giao diện đăng bài của người dùng (ảnh 2) 17](#_Toc529307706)

[Hình 9. Giao diện chính của nhân viên 17](#_Toc529307706)

[Hình 10. Giao diện quản lý bài đăng của nhân viên (ảnh 1) 18](#_Toc529307706)

[Hình 11. Giao diện quản lý bài đăng của nhân viên (ảnh 2) 18](#_Toc529307706)

[Hình 12. Giao diện tạo bài đăng mới của nhân viên 19](#_Toc529307706)

[Hình 13. Giao diện quản lý người dùng của nhân viên 19](#_Toc529307706)

[Hình 14. Giao diện tạo mới người dùng của nhân viên 20](#_Toc529307706)

[Hình 15. Giao diện đăng nhập 20](#_Toc529307706)

[Hình 16. Giao diện chính của người quản lý 21](#_Toc529307706)

[Hình 17. Giao diện quản lý nhân viên (ảnh 1) 21](#_Toc529307706)

[Hình 18. Giao diện quản lý nhân viên (ảnh 2) 22](#_Toc529307706)

[Hình 19. Giao diện quản lý nhân viên (ảnh 3) 22](#_Toc529307706)

[Hình 20. Giao diện quản lý nhân viên (ảnh 4) 23](#_Toc529307706)

[Hình 21. Giao diện quản lý nhân viên (ảnh 5 23](#_Toc529307706)

Hình 22. Giao diện chính sửa thông tin nhân viên [24](#_Toc529307707)

Hình 23. Giao diện tạo nhân viên mới [24](#_Toc529307707)

Hình 24. Giao diện quản lý tài khoản nhân viên [25](#_Toc529307707)

Hình 25. Giao diện quản lý khách hàng bị báo cáo [25](#_Toc529307707)

Hình 26. Mô hình MVC [26](#_Toc529307707)

Hình 27. Cấu trúc cây thư mục của dự án [27](#_Toc529307707)

Hình 28. Vị trí của Entity Framework trong mô hình ASP.NET MVC [28](#_Toc529307707)

# DANH MỤC CÁC BẢNG

[Bảng 1. Bảng mô tả các actor trong use case diagram 7](#_Toc532383401)

[Bảng 2. Bảng mô tả các use case trong use case diagram 8](#_Toc532383402)

[Bảng 3. Danh sách các thủ tục trong cơ sở dữ liệu 10](#_Toc532383403)

[Bảng 4. Một số trigger nổi bật 10](#_Toc532383404)

[Bảng 5. Transaction 12](#_Toc532383405)

[Bảng 6. Concurrency control 13](#_Toc532383406)

[Bảng 7. Index 13](#_Toc532383407)

# 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… nắm bắt được nhu cầu đó nhóm chúng em quyết định thực hiện đồ án hệ quản trị cơ sở dữ liệu là web bất động sảnvới các chức năng chính như sau: Đối với khách hàng giúp người dùng đăng bài hoặc tìm kiếm các loại bất động sản phù hợp với nhu cầu của họ.

## Phương pháp nghiên cứu

* Công cụ lập trình: Visual Studio Enterprise 2017, Git.
* Mô hình lập trình: ASP.NET MVC.
* Ngôn ngữ lập trình: C#.
* Thư viện (API): Entity Framework.
* Hệ quản trị cơ sở dữ liệu: SQL Server 2016.

## Chức năng cơ bản

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

### Người dùng

* Thêm bài đăng
* Chỉnh sửa tài khoản
* Đăng ký tài khoản
* Xem các bài đăng về các loại bất động sản và tìm kiếm các bài đăng theo ý muốn

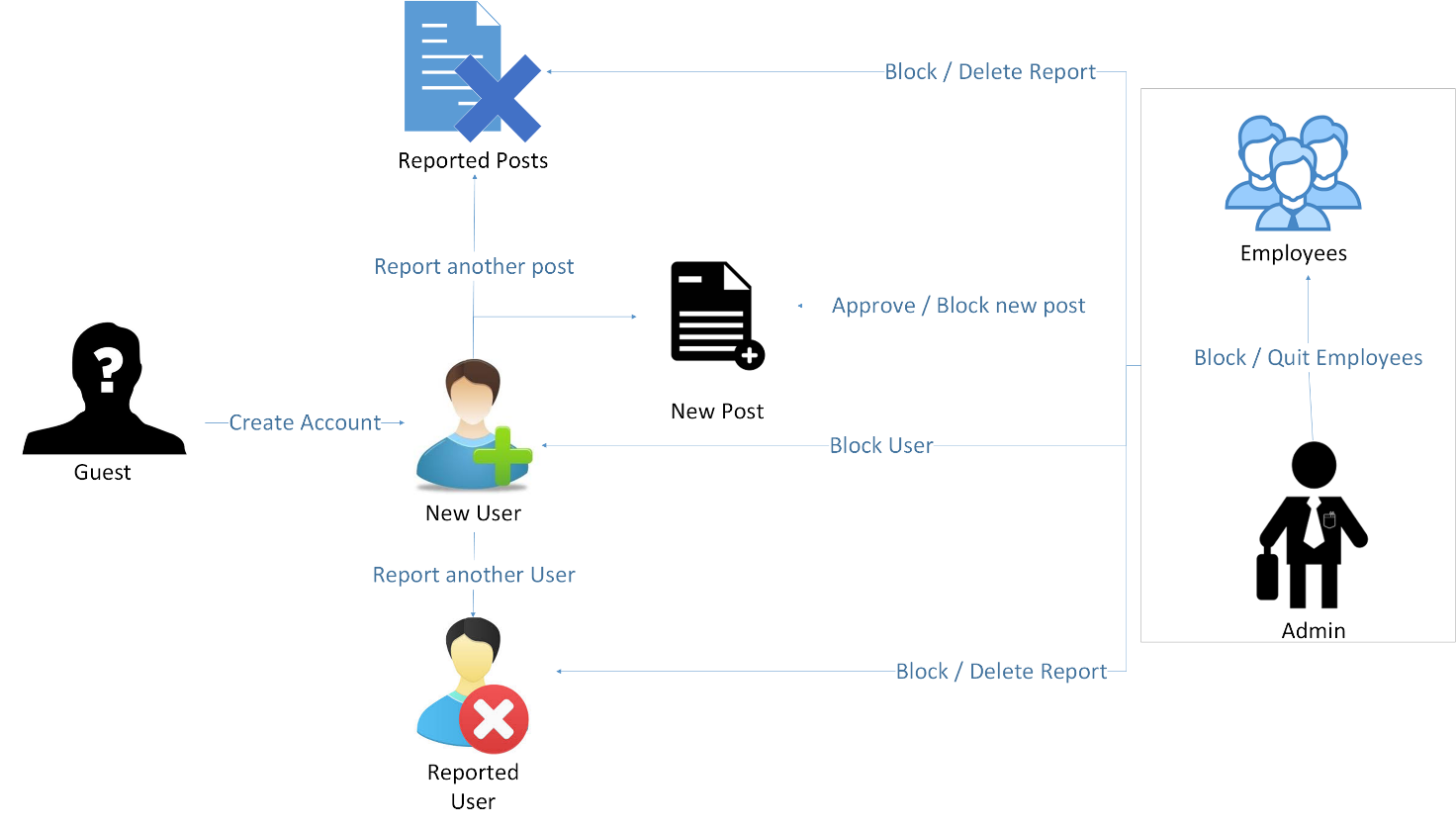
### Nhân viên

* 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 người dùng
  + Tạo người dùng mới.
  + Khóa người dùng.
  + Quản lý những người dùng bị người dùng khác báo cáo.

### Người quản lý

* Quản lý nhân viên
* Tạo nhân viên mới.
* Khoá/Mở Khoá nhân viên.
* Nghỉ việc nhân viên.
* Quản lý tài khoản nhân viên.
* Xem/Chỉnh sửa thông tin chi tiết nhân viên.
* Quản lý người dùng
* Tạo người dùng mới.
* Khoá/Mở Khoá người dùng.
* Quản lý những người dùng bị người dùng khác báo cáo.
* 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 và chưa được kiểm duyệt bởi kiểm duyệt viên. |
| 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. |
| 7 | Admin | Quản lý toàn bộ hệ thống. |

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. |
| 3 | Report another user | Người dùng có thể báo cáo người dùng khác nếu họ sai phạm hoặc lừa đảo. |
| 4 | Block / Delete Report | Employee hoặc 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 | Employee hoặc Admin có thể duyệt hoặc khóa bài đăng mới của người dùng. |
| 6 | Block User | Employee hoặc Admin có thể khóa tài khoản người dùng. |
| 7 | Block / Quit Employee | Admin có thể khóa tài khoản nhân viên hoặc cho nghỉ việc tài khoản nhân viên. |

# CƠ SỞ DỮ LIỆU

## Mô hình ERD

Vì mô hình ERD quá dài nên mô hình đã được đính kèm theo file báo cáo với tên “ERD – RealEstateSystem.pdf”.

## Database diagram

Vì Database diagram quá dài nên diagram đã được đính kèm theo file báo cáo với tên “DBDiagram – RealEstateSystem.pdf”.

## Stored Procedures

Bảng 3. Danh sách các thủ tục trong cơ sở dữ liệu

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TT** | **Tên stored procedure** | **Input** | **Output** | **Mục đích** |
| **1** | spUpdateAccount | Mật khẩu |  | Đổi mật khẩu người dùng. |
| **2** | spSearchKeyPrice | Từ khóa, giá nhỏ nhất, giá lớn nhất | Thông tin bài đăng | Tìm kiếm bài đăng theo giá và từ khóa. |
| **3** | uspLogError | Error Log ID |  | Lưu lại lỗi của SQL khi hệ thống bị lỗi. |
| **4** | uspPrintError |  |  | In ra lỗi giúp admin dễ hiểu. |
| **5** | USP\_GetPostCountBydate |  | Số bài đăng | Số bài đăng trong 10 ngày gần nhất. |
| **6** | USP\_UnBlockBlockCust | ID người dùng, ngày mở khoá |  | Mở khoá người dùng. |

## Một số trigger nổi bật

Bảng 4. Một số trigger nổi bật

|  |  |  |
| --- | --- | --- |
| **TT** | **Nội dung Trigger** | **Mục đích** |
| **1** | ALTER TRIGGER [Post].[tgr\_Post.Post\_Status\_InsertUpdate] ON [Post].[Post\_Status]  AFTER INSERT, UPDATE  AS  BEGIN  SET NOCOUNT ON;  DECLARE @sta\_id int;  SELECT @sta\_id = i.Post\_Status\_ID FROM inserted i;  UPDATE Post.Post\_Status  SET CensorshipTime = GETDATE()  WHERE Post\_Status\_ID = @sta\_id  DECLARE @post\_id int  DECLARE @stt int  SELECT @post\_id = i.Post\_ID, @stt = i.Status FROM inserted i;  UPDATE Post.Post  SET Status = @stt  WHERE Post\_ID = @post\_id  END | Tự động cập nhật thời gian insert dữ liệu vào bảng Post\_Status và thay đổi thuộc tính Status bên bảng Post. |
| **2** | ALTER TRIGGER [Customer].[tgr\_Customer.Block\_InsertUpdate] ON [Customer].[Block]  AFTER INSERT, UPDATE  AS  BEGIN  SET NOCOUNT ON;  DECLARE @block\_id int;  DECLARE @cus\_id int;  SELECT @block\_id = i.Block\_ID FROM inserted i;  SELECT @cus\_id = i.Cus\_ID FROM inserted i;  IF NOT EXISTS(SELECT \* FROM DELETED)  BEGIN -- Insert  UPDATE Customer.Block  SET ModifiedDate = GETDATE(), BlockDate = GETDATE()  WHERE Block\_ID = @block\_id  END  ELSE -- Update  BEGIN  UPDATE Customer.Block  SET ModifiedDate = GETDATE()  WHERE Block\_ID = @block\_id  END  END | Thực hiện lệnh sau khi Insert vào bảng Customer.Block   * Nếu là Insert Block thì đặt giá trị ModifiedDate và BlockDate là ngay lúc đó. * Nếu là chỉnh sửa thì thay đổi ModifiedDate. |
| **3** | ALTER TRIGGER [Customer].[tgr\_Customer.PhoneNumber\_InsertUpdate] ON [Customer].[PhoneNumber]  AFTER INSERT, UPDATE  AS  BEGIN  SET NOCOUNT ON;  DECLARE @cus\_id int;  DECLARE @pho\_id varchar(20);  SELECT @cus\_id = i.Customer\_ID FROM inserted i;  SELECT @pho\_id = i.PhoneNumber FROM inserted i;  UPDATE Customer.PhoneNumber  SET ModifiedDate = GETDATE()  WHERE Customer\_ID = @cus\_id AND PhoneNumber = @pho\_id  END | Chèn ngày chỉnh sửa vào bảng PhoneNumber sau khi Insert. |

* Ngoài ra còn rất nhiều Trigger tương tự.

## Transaction

Bảng 5. Transaction

|  |  |  |
| --- | --- | --- |
| TT | Nội dung | Mục đích |
| 1 | **using (var trans = db.Database.BeginTransaction())**  **{**  **try**  **{**  **// Create new post**  **db.Posts.Add(postData.post);**  **db.SaveChanges();**  **// Create new post\_status**  **postData.post.Post\_Status.Add(ps);**  **db.SaveChanges();**  **trans.Commit();**  **TempData["addNewPost"] = "OK";**  **return RedirectToAction("AddPost");**  **}**  **catch (Exception ex)**  **{**  **Console.WriteLine(ex.ToString());**  **trans.Rollback();**  **return View();**  **}**  **}** | - Nhân viên tạo mới bài đăng.  - Lưu bài đăng mới vào data để trigger tự sinh ra post\_status là đang chờ duyệt.  - Sau đó thêm post\_status là được duyệt bởi chính người đăng đó vì người đăng đó là nhân viên, không cần duyệt nữa. |
|  |  |  |

## Concurrency control

Bảng 6. Concurrency control

|  |  |  |
| --- | --- | --- |
| TT | Nội dung | Mục đích |
| 1 | Customer customerDB = db.Customers.Find(customer.Customer\_ID);  try  {  // Change info of customer  db.SaveChanges();  TempData["SaveOK"] = "OK";  return RedirectToAction("Index");  }  catch (OptimisticConcurrencyException)  {  db.Entry(customerDB).Reload();  if (db.Entry(customerDB).State == EntityState.Detached)  {  TempData["SaveError"] = "Customer has been deleted by another user";  }  else  {  TempData["SaveError"] = "Customer has been updated by another user";  }  }  catch (Exception ex)  {  TempData["SaveError"] = "Somethings was wrong. Please try again.";  }  return RedirectToAction("Index"); | - Kiểm soát truy cập đồng thời khi sửa thông tin khách hàng. |
|  |  |  |

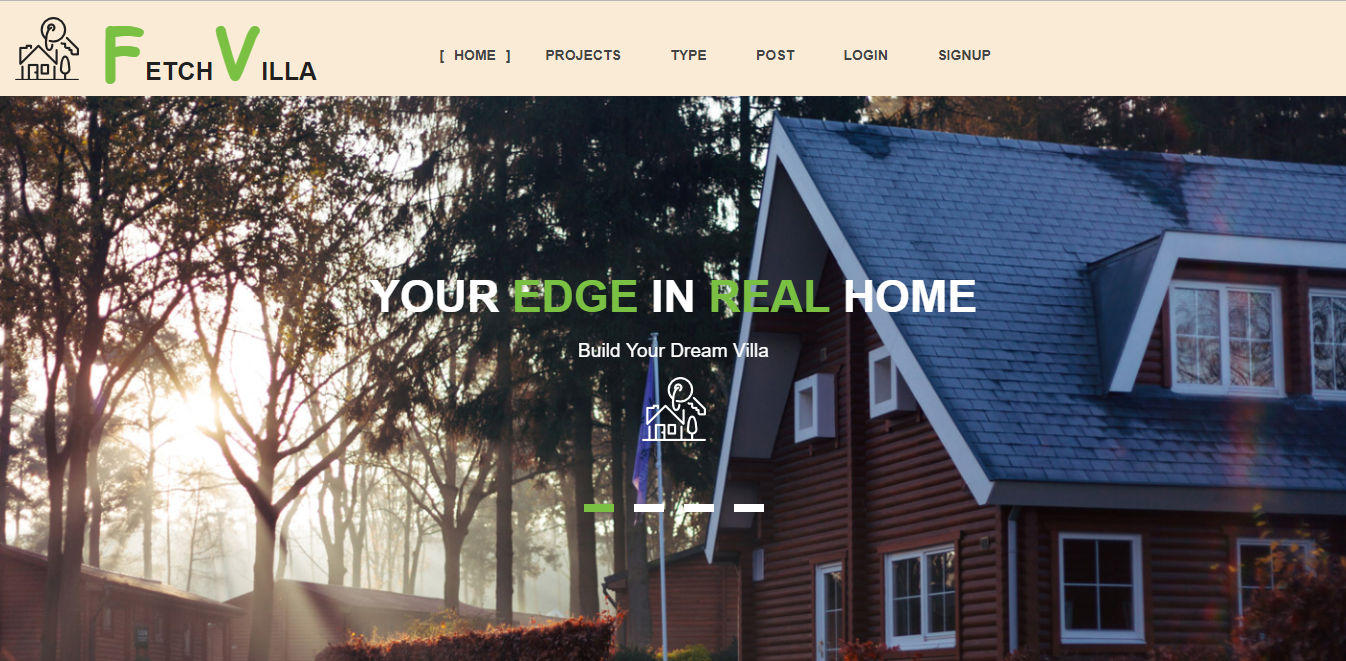
## Index

Bảng 7. Index

|  |  |  |
| --- | --- | --- |
| TT | Nội dung | Mục đích |
| 1 | CREATE INDEX Index\_ALog ON Account.AccountLog(ModifiedDate) | - Tạo nonclustered index cột “ngày insert” cho bảng AccountLog giúp tìm tổng số người đã đăng nhập trong 1 năm được nhanh hơn. |
| 2 | CREATE INDEX PK\_Account\_Username ON Account.Account(Username) | - Tạo nonclustered index cột “username” cho bảng Account giúp người dùng đăng nhập bằng account nhanh hơn. |
| 3 | CREATE INDEX PK\_Customer\_PhoneNumber ON Customer.PhoneNumber(PhoneNumber) | - Tạo nonclustered index cột “PhoneNumber” cho bảng Account giúp tìm người dùng theo số điện thoại được nhanh hơn. |

# GIAO DIỆN

## Giao diện người dùng



Hình 2. Giao diện trang chủ của người dùng

Hình 3. Giao diện đăng nhập của người dùng

Hình 4. Giao diện đăng ký của người dùng

Hình 5. Giao diện đổi mật khẩu của người dùng

Hình 6. Giao diện tìm kiếm bài đăng của người dùng

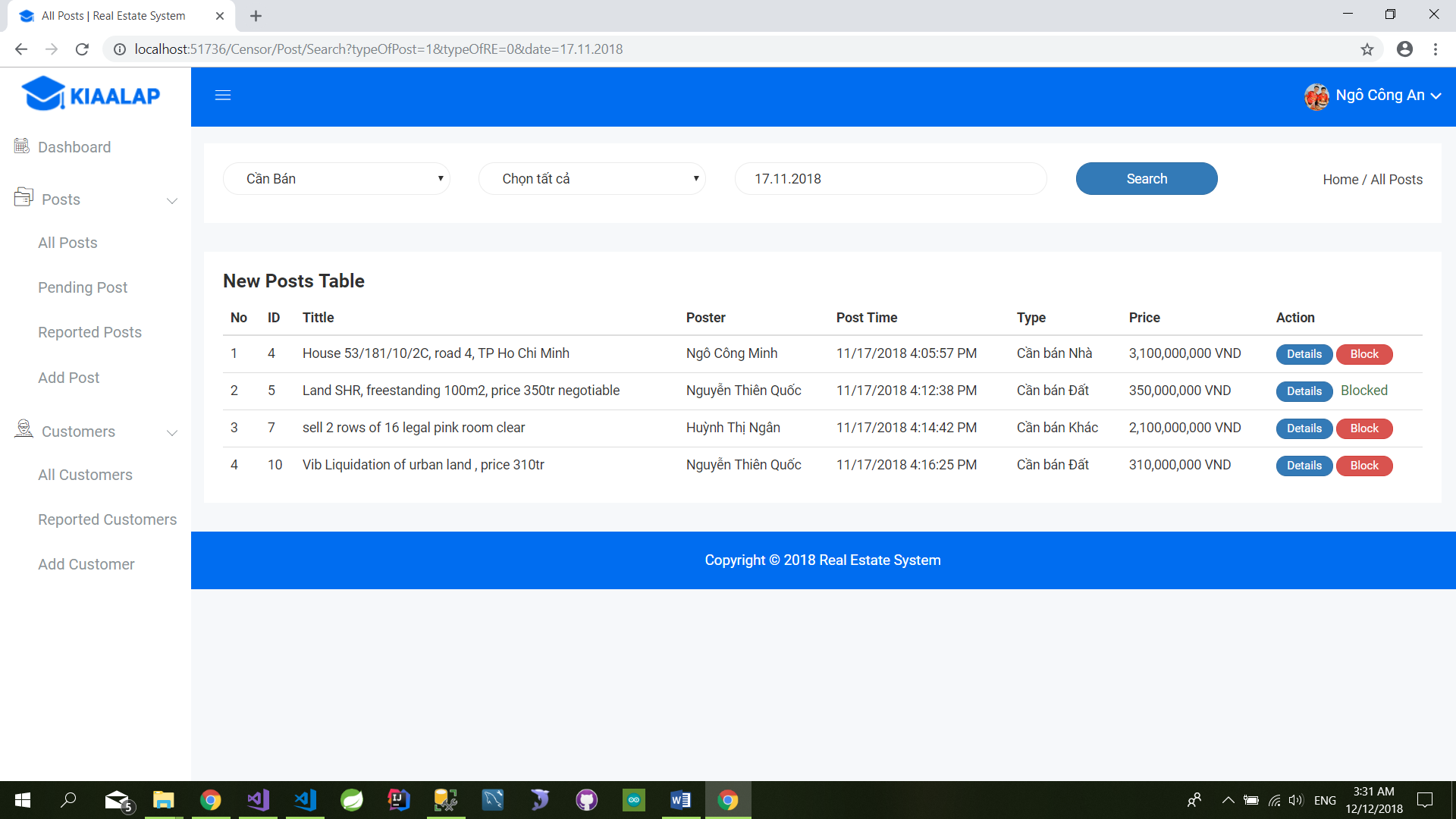
Hình 7. Giao diện đăng bài của người dùng (ảnh 1)

Hình 8. Giao diện đăng bài của người dùng (ảnh 2)

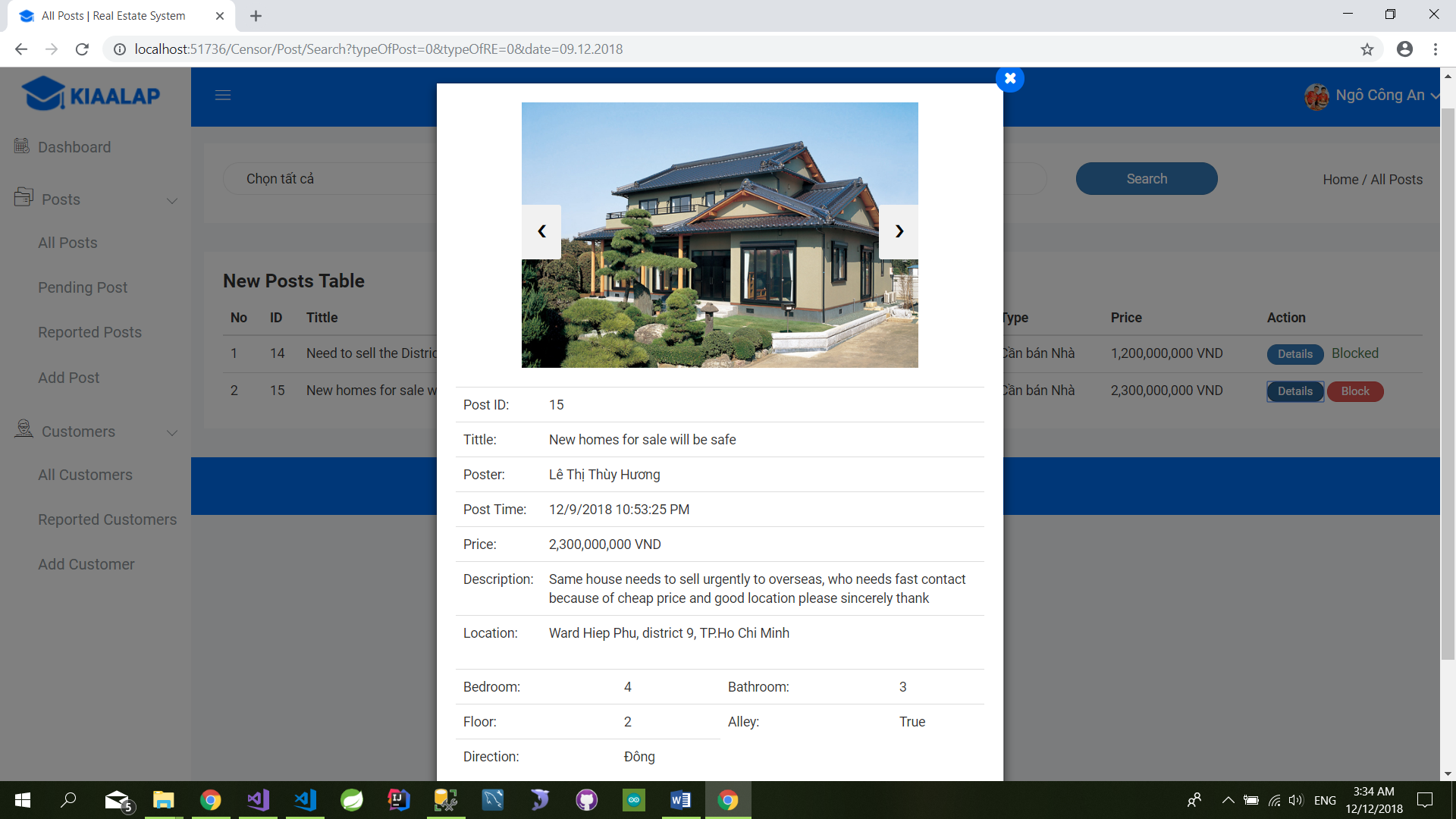
## Giao diện nhân viên kiểm duyệt



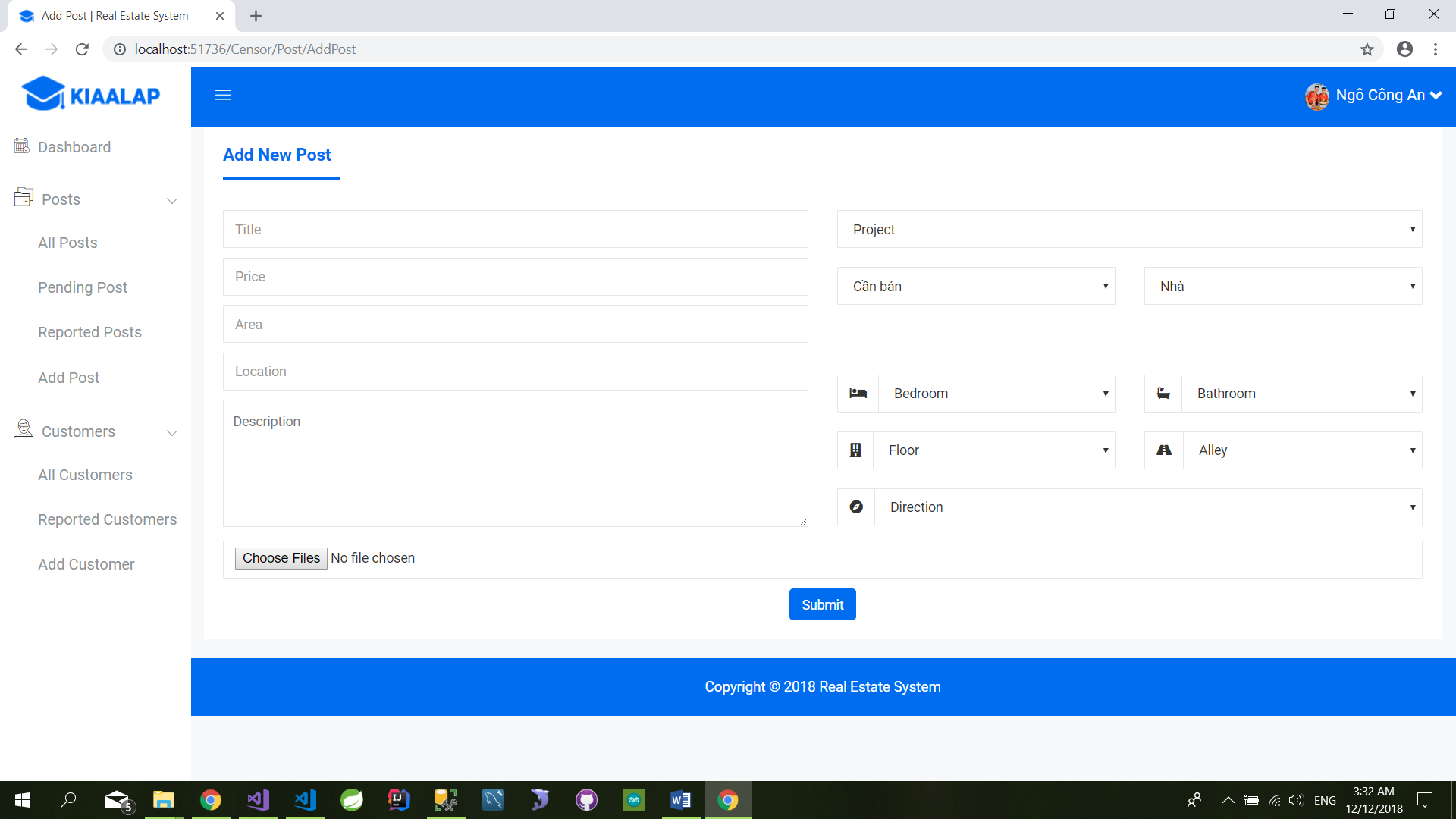
Hình 9. Giao diện chính của nhân viên



Hình 10. Giao diện quản lý bài đăng của nhân viên (ảnh 1)



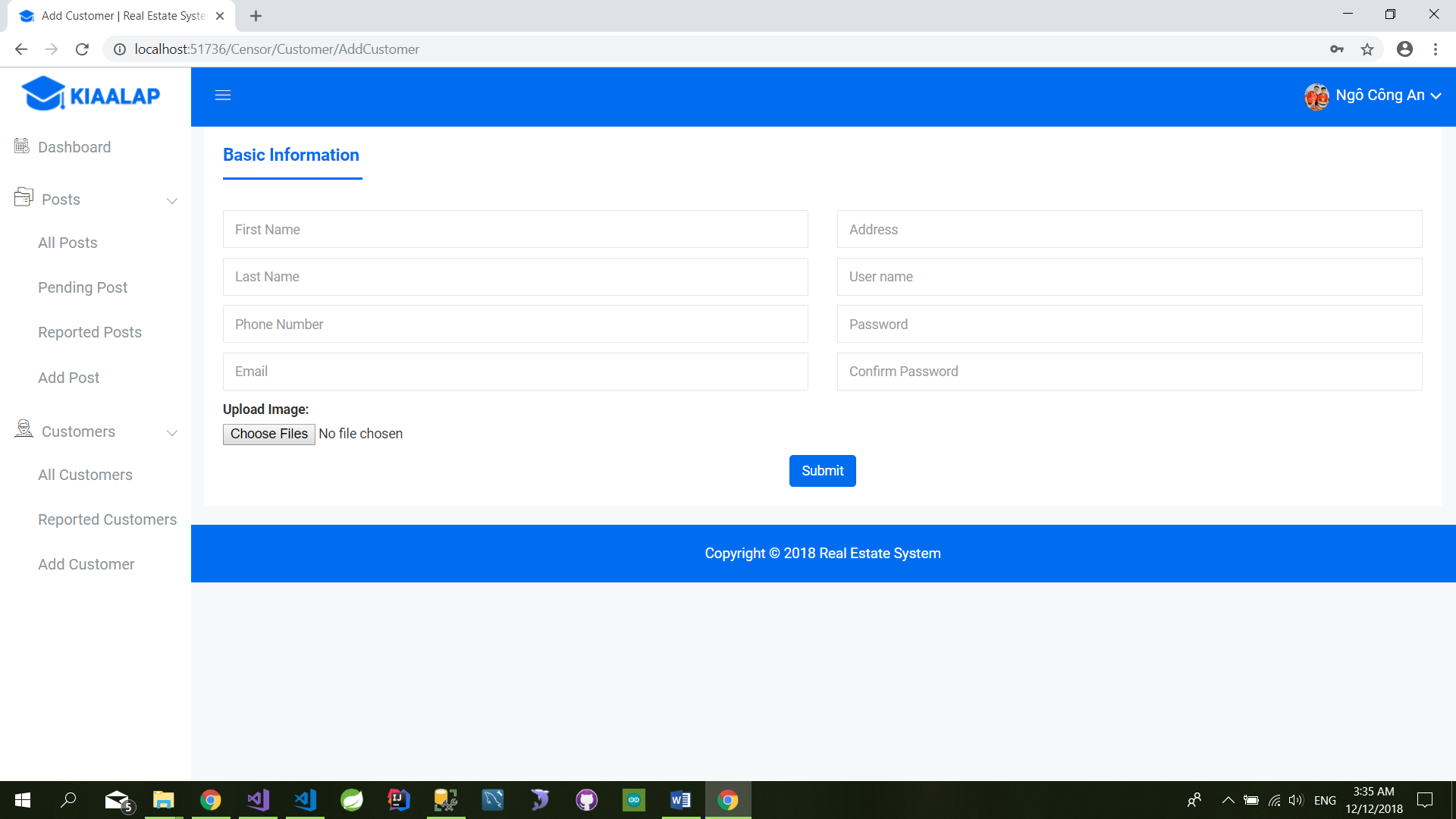
Hình 11. Giao diện quản lý bài đăng của nhân viên (ảnh 2)



Hình 12. Giao diện tạo bài đăng mới của nhân viên

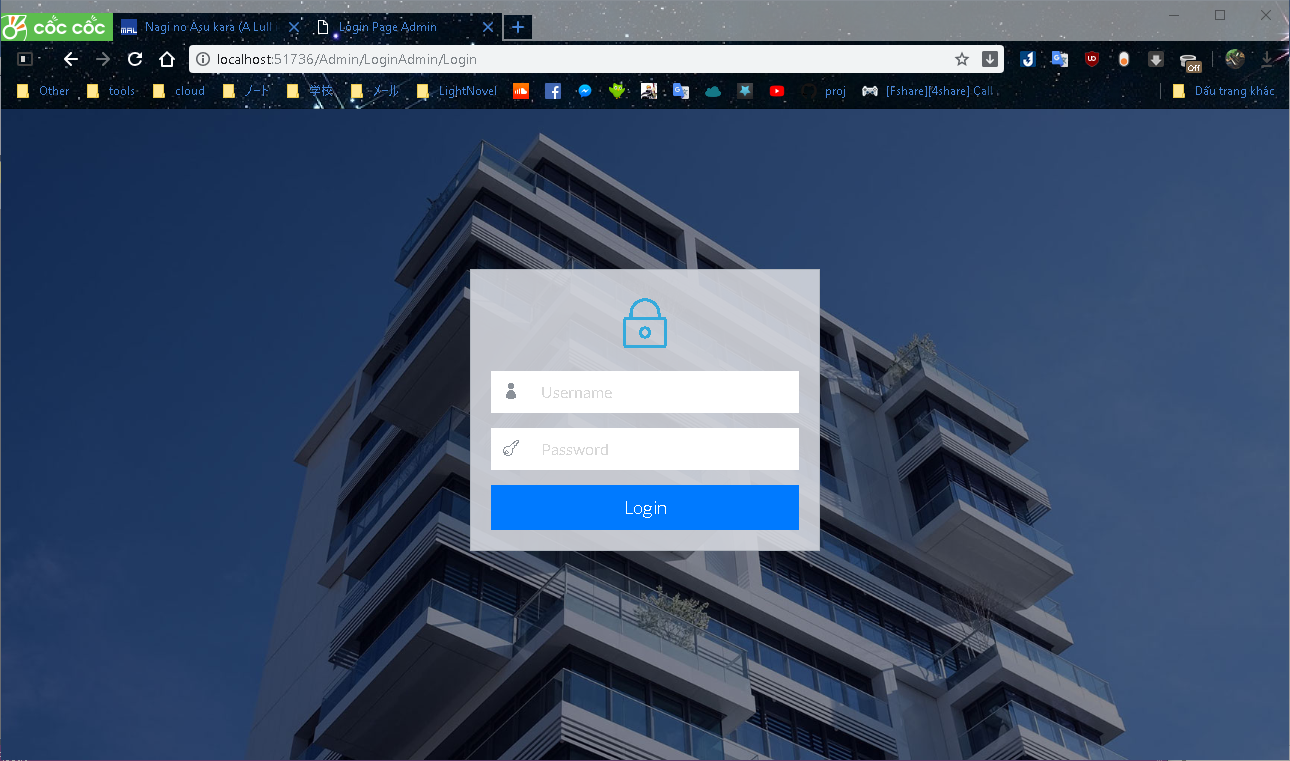


Hình 13. Giao diện quản lý người dùng của nhân viên

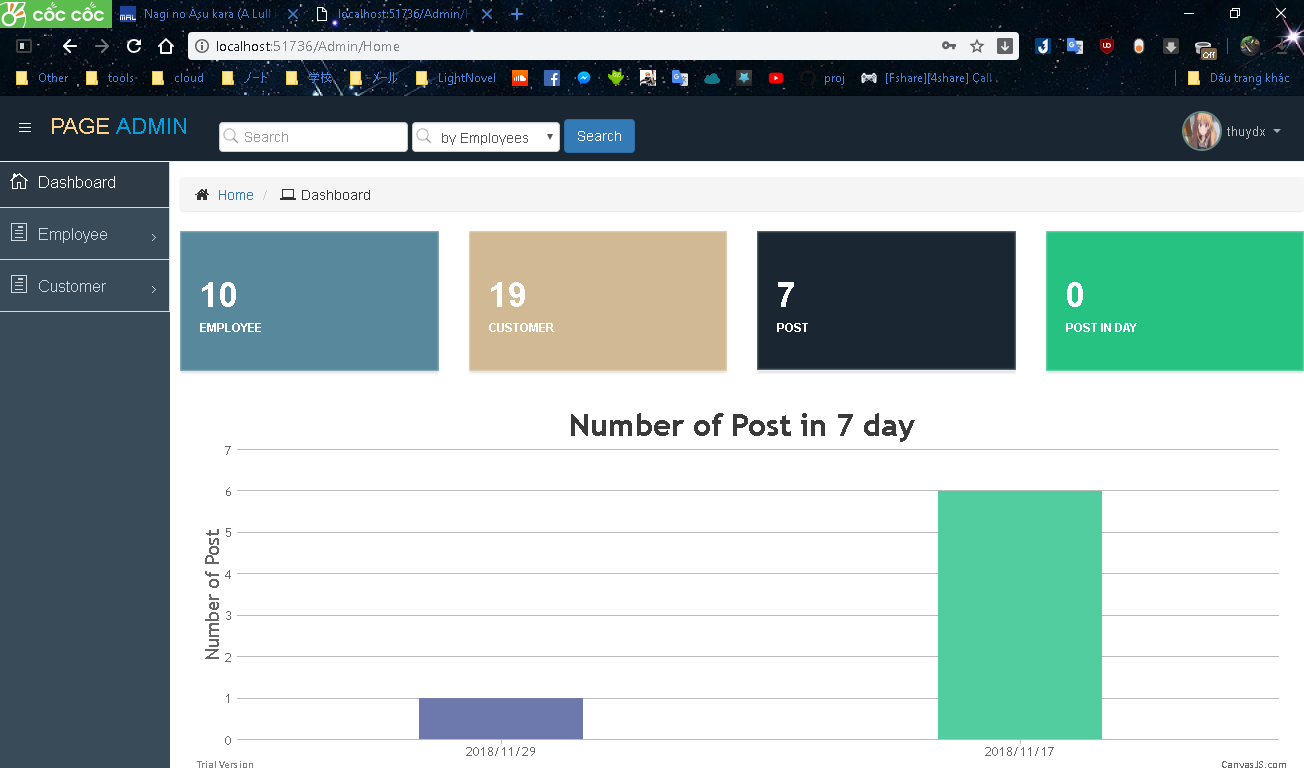


Hình 14. Giao diện tạo mới người dùng của nhân viên

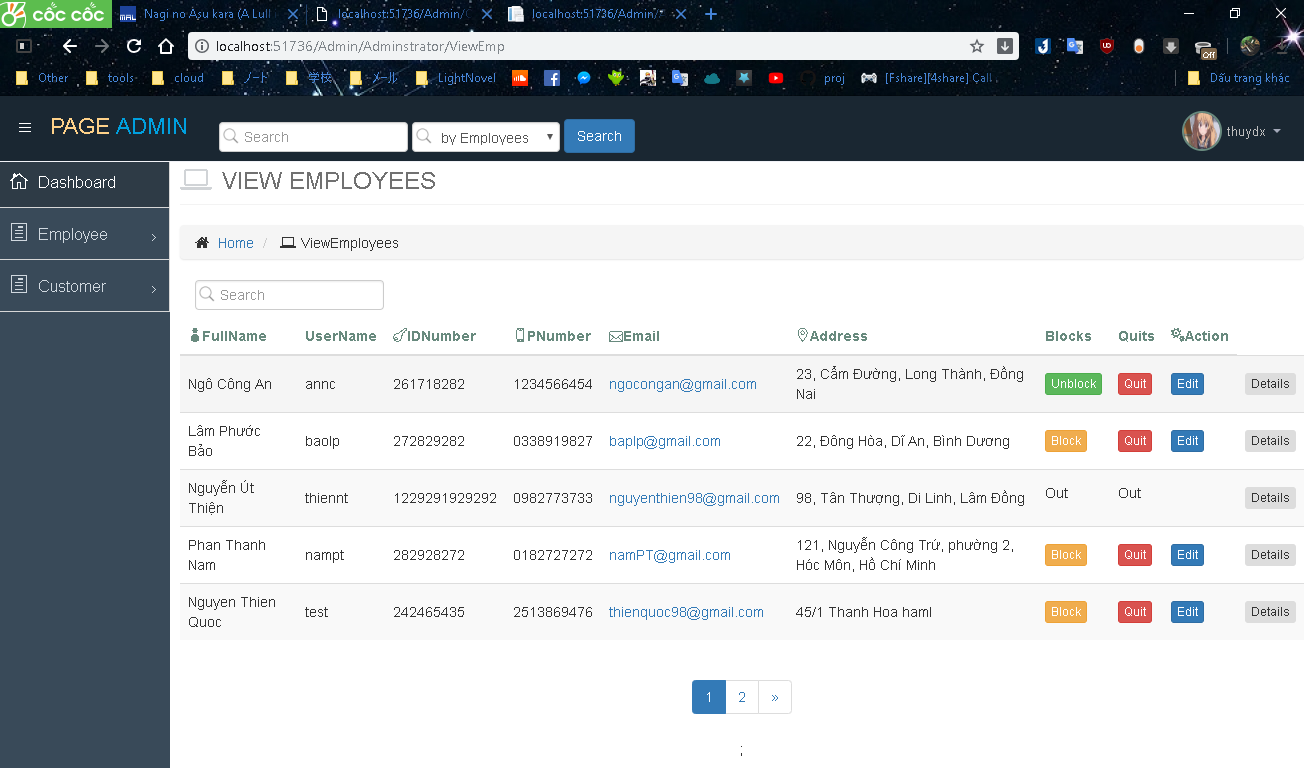
## Giao diện quản lý



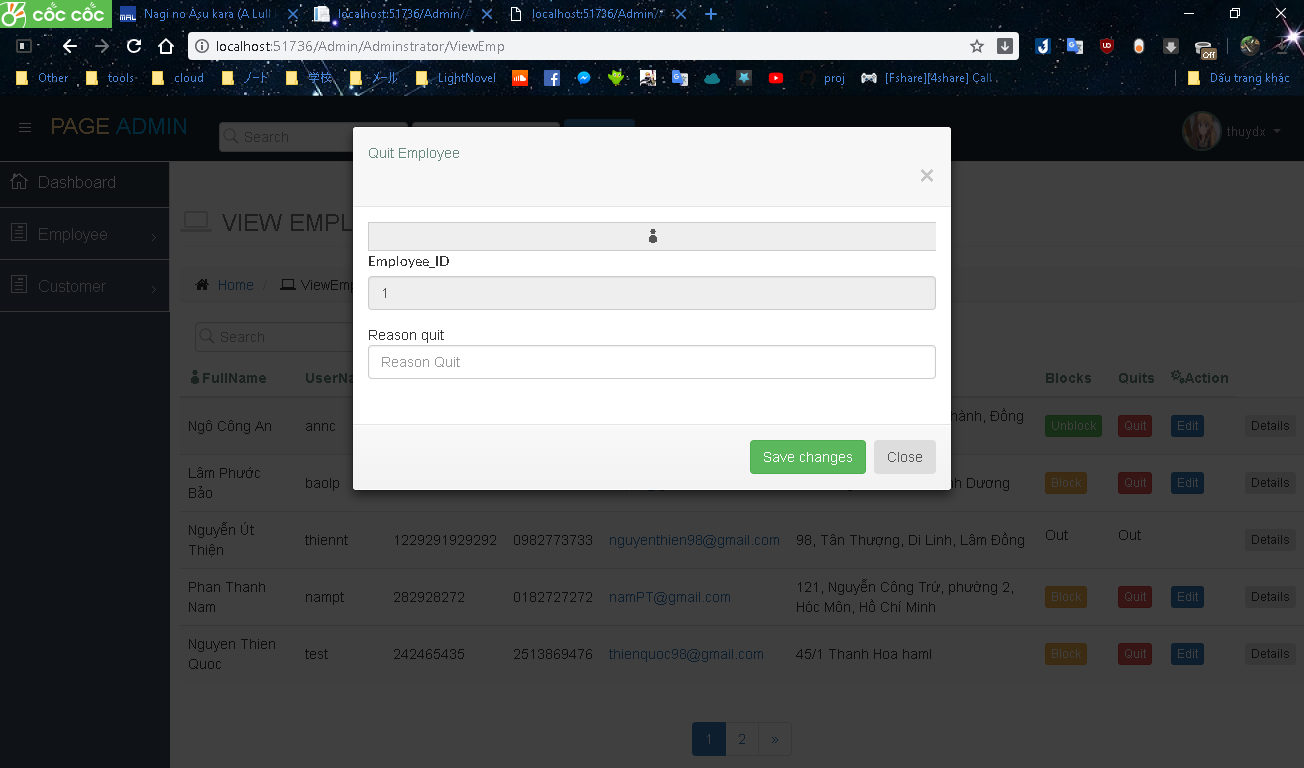
Hình 15. Giao diện đăng nhập



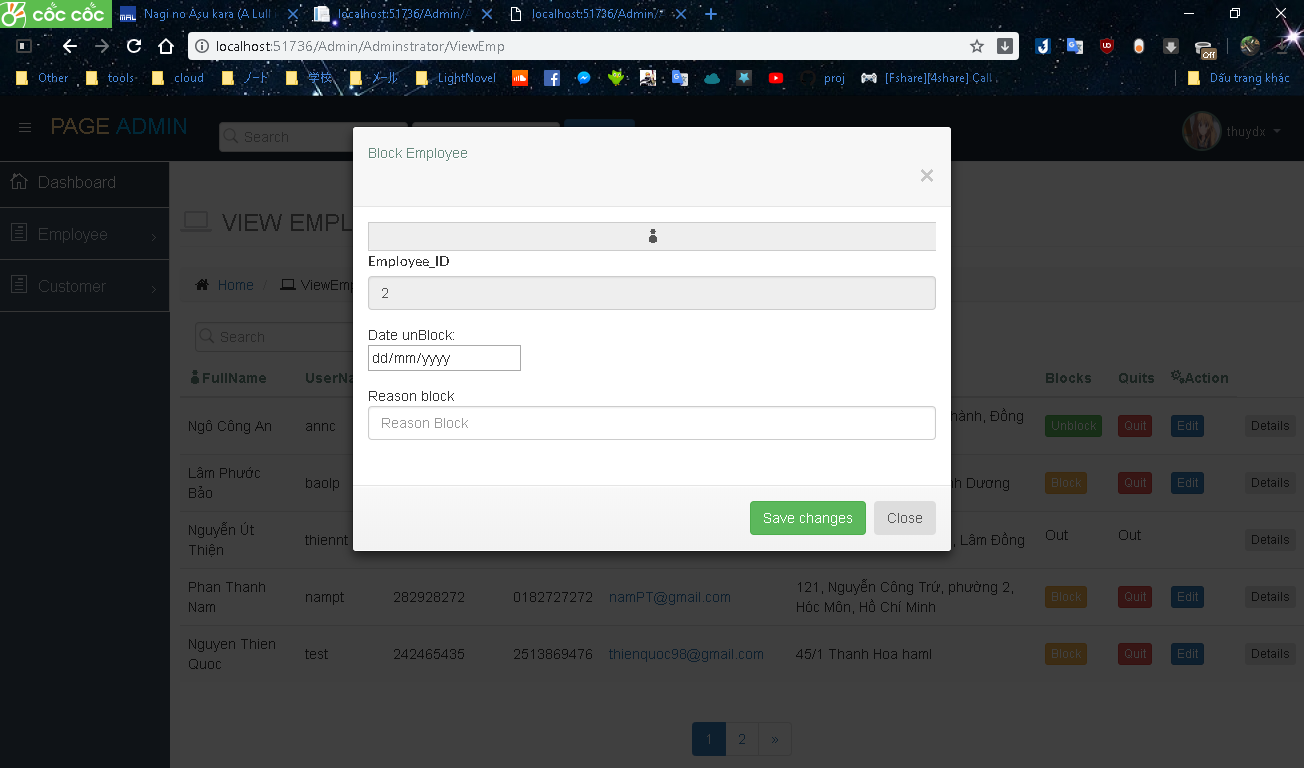
Hình 16. Giao diện chính của người quản lý



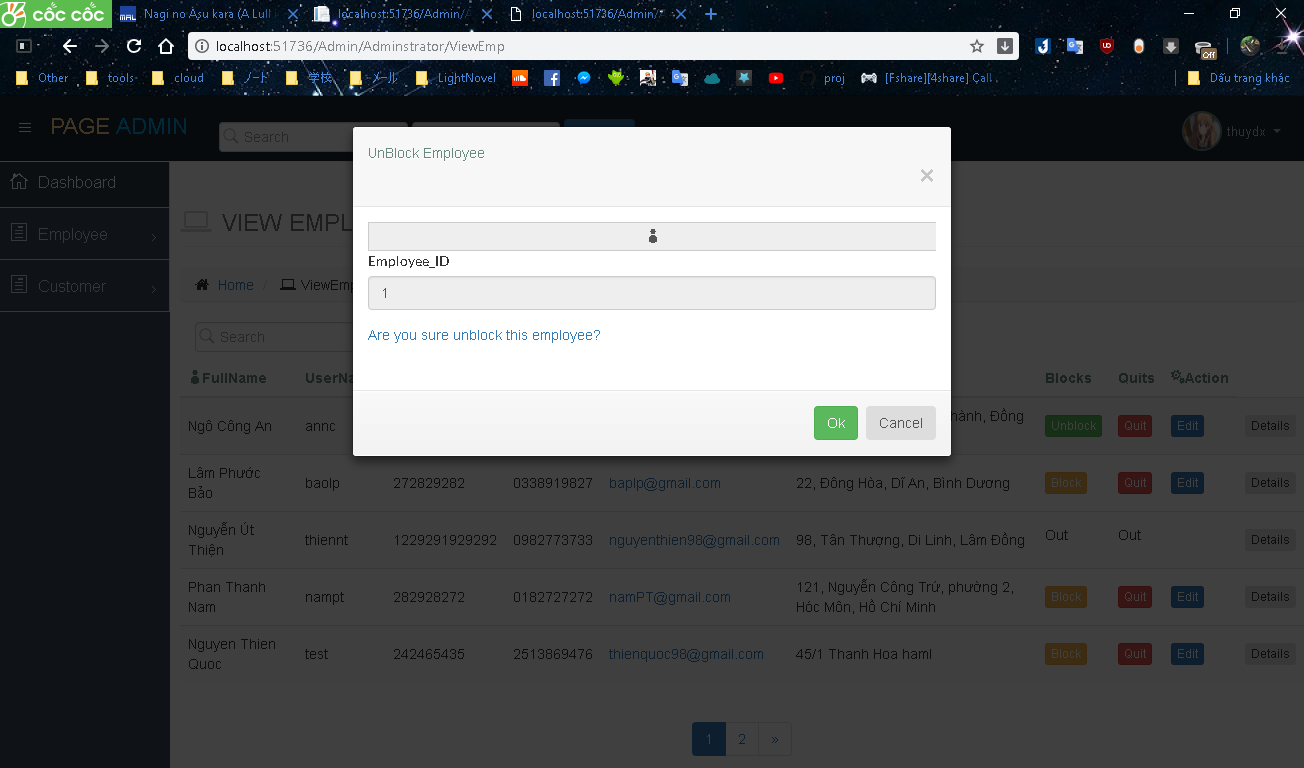
Hình 17. Giao diện quản lý nhân viên (ảnh 1)



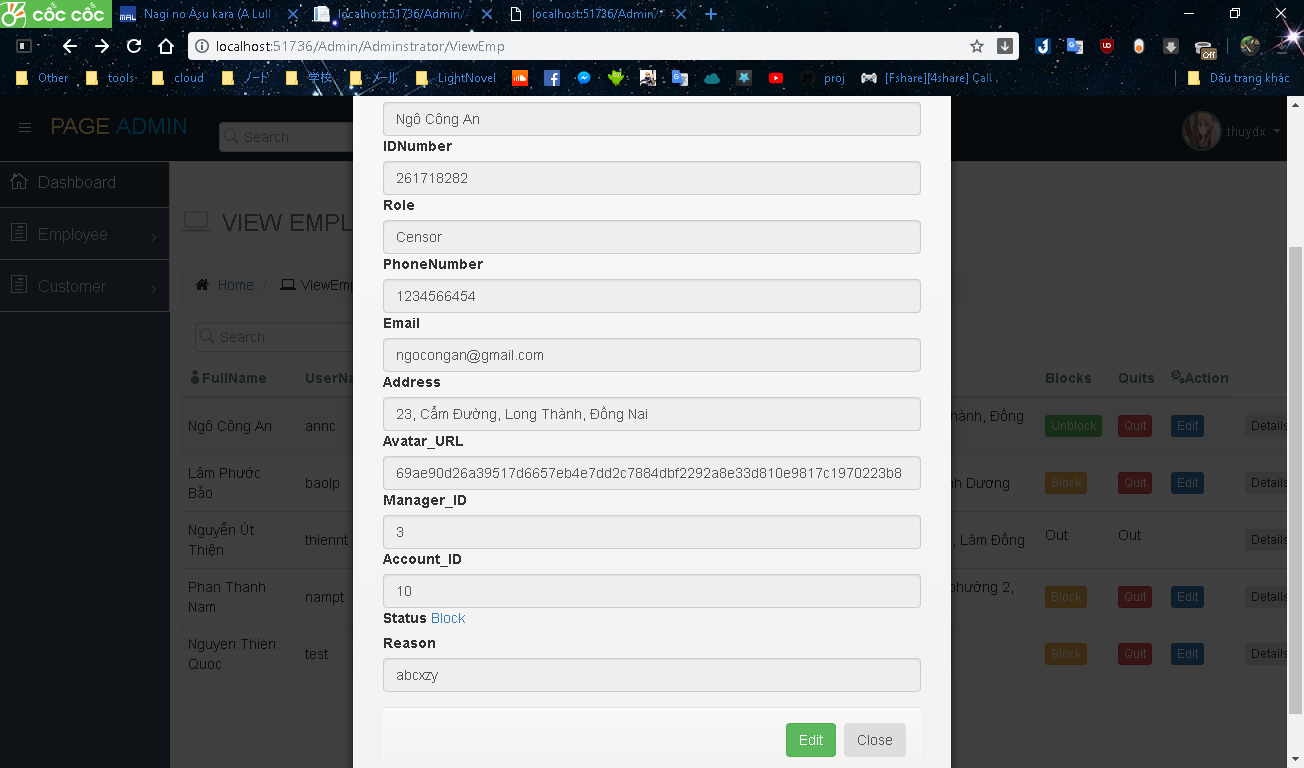
Hình 18. Giao diện quản lý nhân viên (ảnh 2)



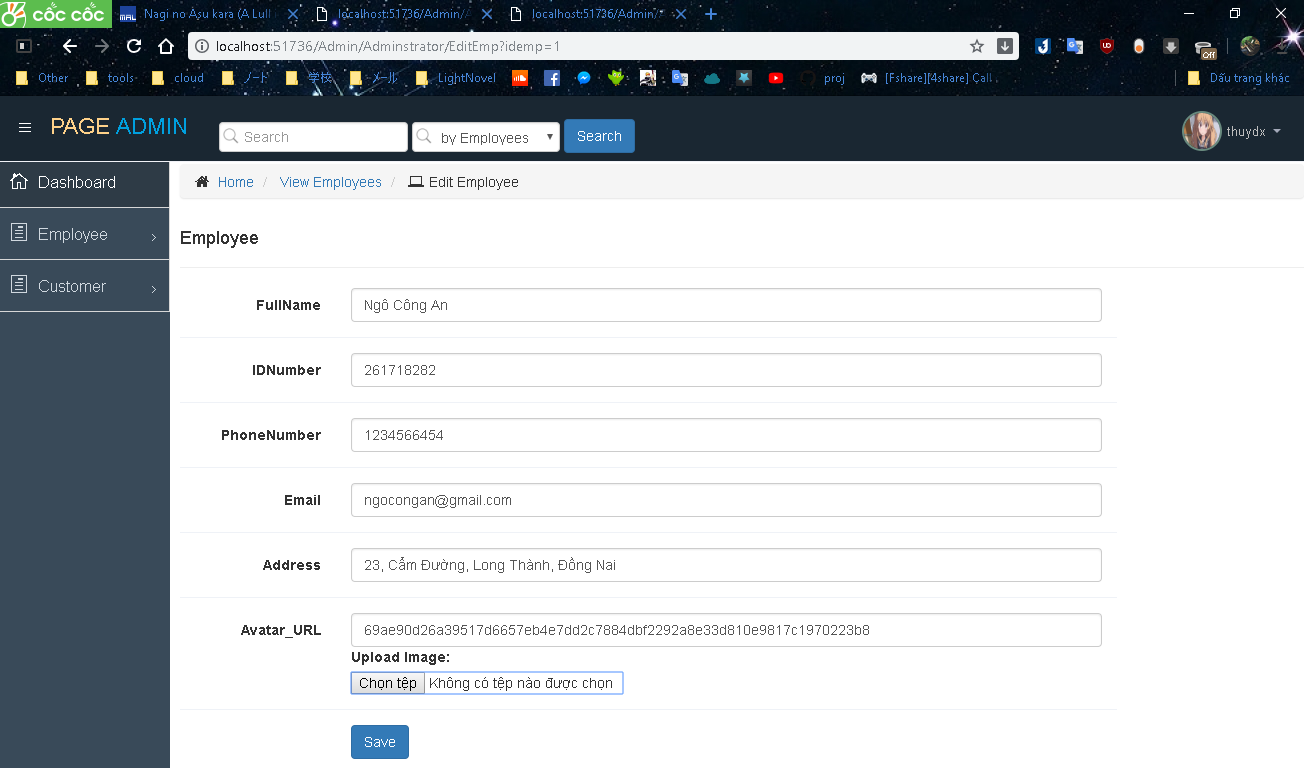
Hình 19. Giao diện quản lý nhân viên (ảnh 3)



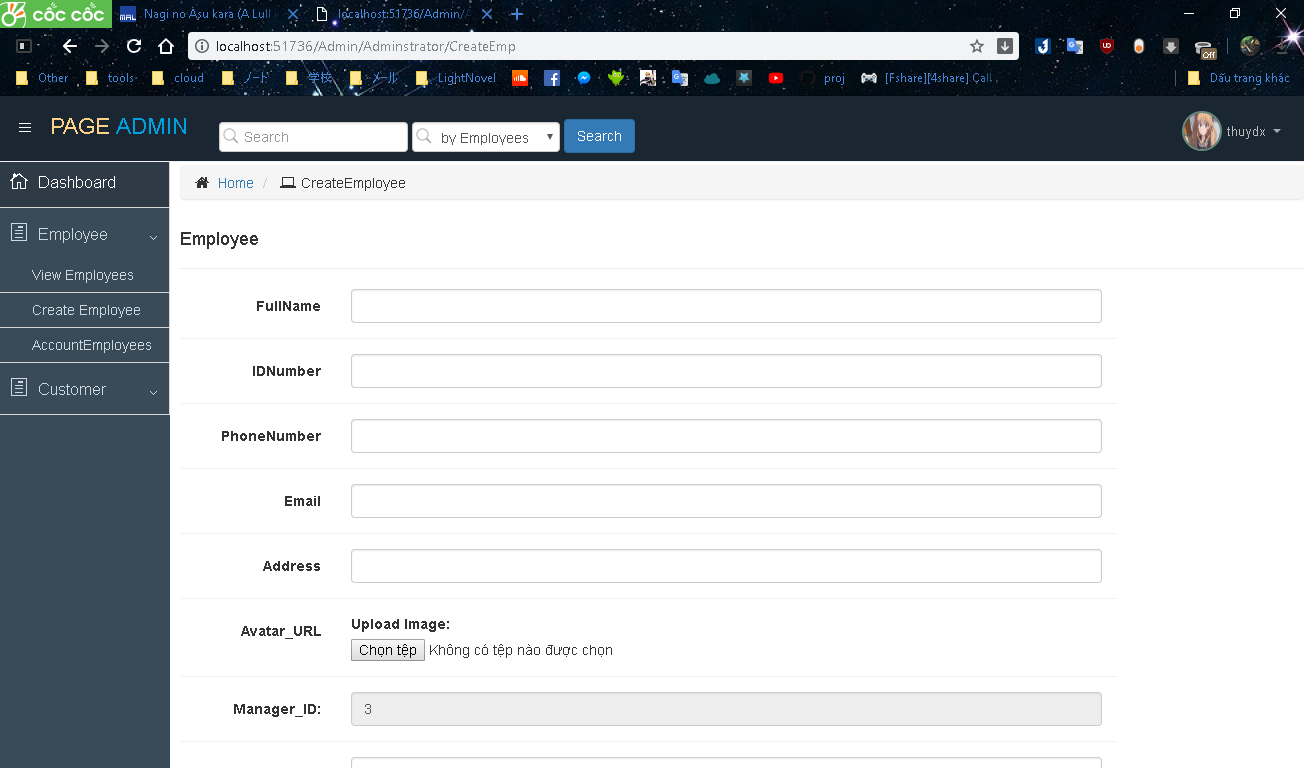
Hình 20. Giao diện quản lý nhân viên (ảnh 4)



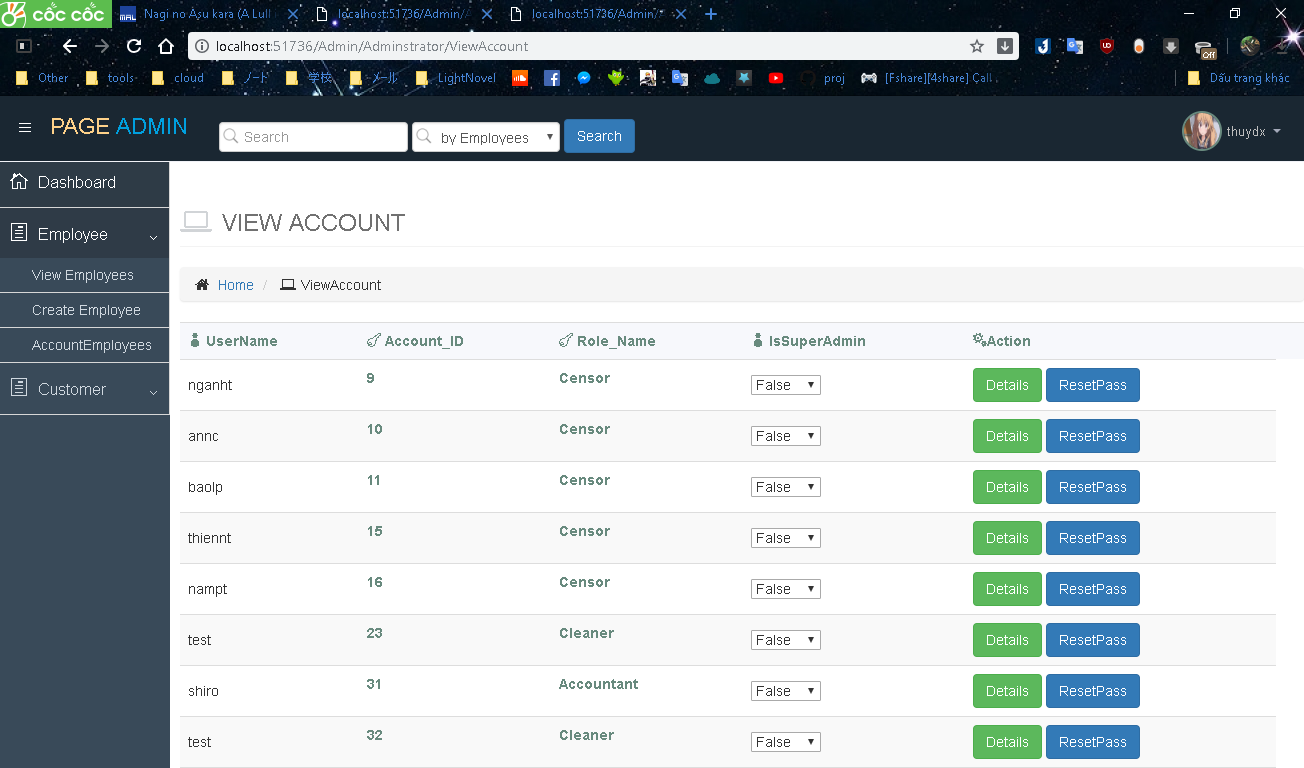
Hình 21. Giao diện quản lý nhân viên (ảnh 5)



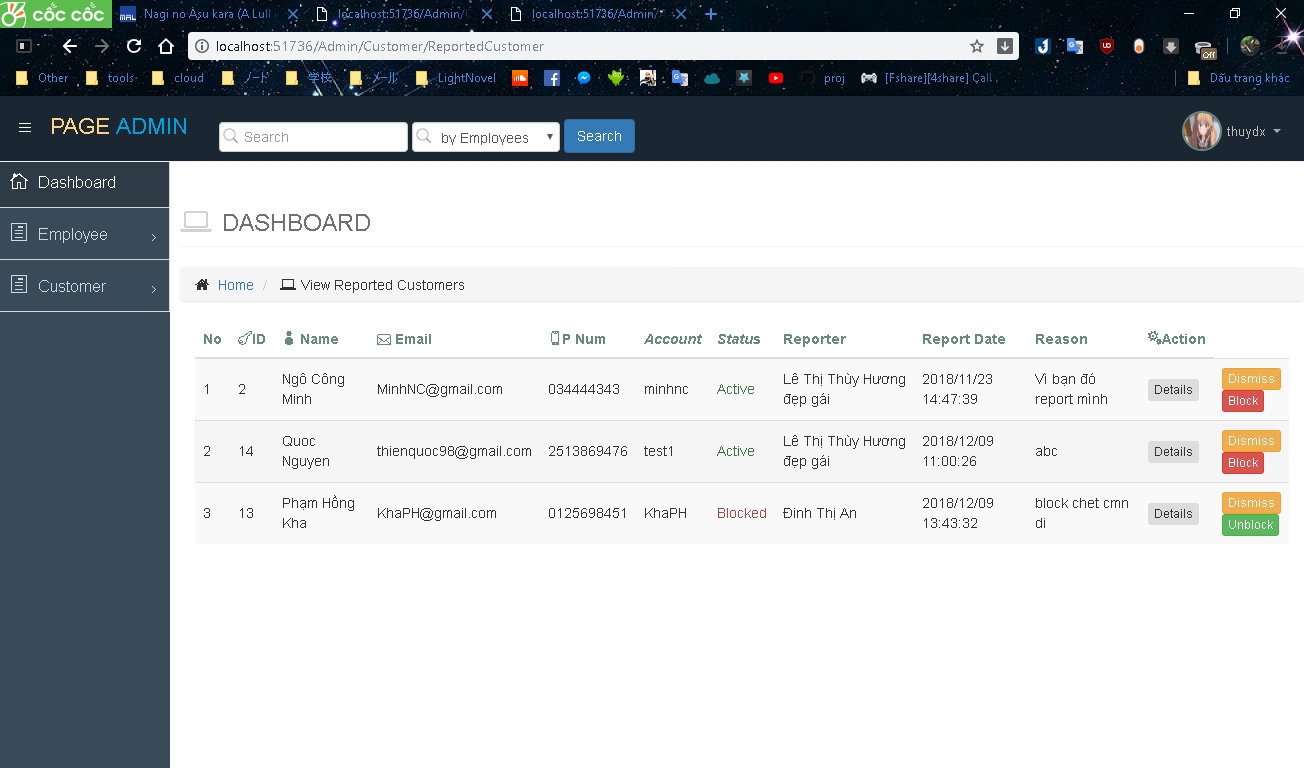
Hình 22. Giao diện chính sửa thông tin nhân viên



Hình 23. Giao diện tạo nhân viên mới



Hình 24. Giao diện quản lý tài khoản nhân viên



Hình 25. Giao diện quản lý khách hàng bị báo cáo

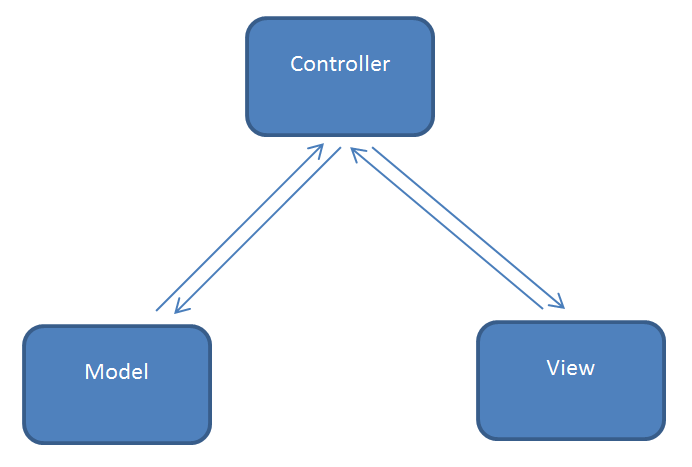
# MÔ HÌNH - FRAMEWORK THỰC HIỆN ĐỀ TÀI

## Mô hình xây dựng dự án

Dự án được xây dựng dưới mô hình MVC.

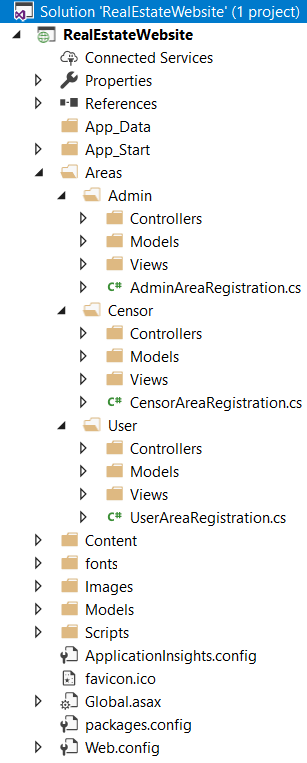
Mô hình MVC là viết tắt của Model – View – Controller. Trong đó Controller chính là trái tim của ứng dụng.

* Controller sẽ chịu trách nhiệm nhận các request được gửi lên từ người dùng, sau đó sẽ xử lý và điều hướng dữ liệu trả về.
* View đảm nhận công việc đơn giản hơn, nó chỉ có một nhiệm vụ duy nhất là chứa mã giao diện hoặc nhận dữ liệu trả về từ Controller.
* Model thì chịu trách nhiệm tương tác với CSDL, có thể là thực thi truy vấn thông thường hoặc trả về dữ liệu dạng đóng gói cho Controller xử lý và điều hướng.



Hình 26. Mô hình MVC

Cấu trúc cây thư mục của dự án:

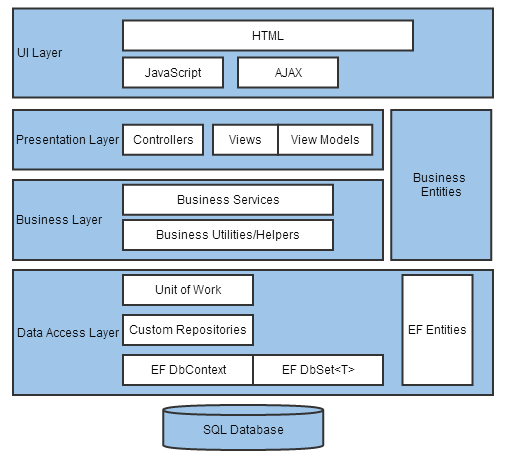


Hình 27. Cấu trúc cây thư mục của dự án

## Framework sử dụng trong dự án

Framework chính được sử dụng trong sự án là Entity Framework.

[**Entity Framework**](https://www.dammio.com/glossary/entity-framework) ([EF](https://www.dammio.com/glossary/ef)) là một framework ánh xạ quan hệ đối tượng (ORM) dành cho ADO.NET, là 1 phần của .NET Framework. [EF](https://www.dammio.com/glossary/ef) cho phép các nhà phát triển Web tương tác với dữ liệu quan hệ theo phương pháp hướng đối tượng đặc trưng. Lợi ích lớn nhất của [EF](https://www.dammio.com/glossary/ef) là giúp lập trình viên giảm thiểu việc lập trình mã nguồn cần thiết để truy cập và tương tác với cơ sở dữ liệu. [EF](https://www.dammio.com/glossary/ef) được Microsoft hỗ trợ phát triển lâu dài và bền vững, vì vậy [EF](https://www.dammio.com/glossary/ef) là 1 framework mạnh nhất hiện nay để phát triển ứng dụng Web với sự hỗ trợ đông đảo của các nhà phát triển Web.

[](https://dammio.files.wordpress.com/2016/11/ef_n_tier1.png)

Hình 28. Vị trí của Entity Framework trong mô hình lập trình và phát triển web ASP.NET MVC

# ĐÁNH GIÁ

## Quá trình thực hiện ứng dụng

Thuận lợi:

* Nhiều công nghệ hỗ trợ cho việc phát triển ứng dụng.
* Nhiều thư viện (API) 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:

* Ít kinh nghiệm trong việc thiết kế giao diện lẫn cơ sở dữ liệu.
* Chưa tận dụng, tối ưu chức năng các thư viện, công nghệ.

## Ưu – Nhược điểm khi ứng dụng

Ưu điểm:

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

Nhược điểm:

* Còn lỗi.
* 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.
* Phát triển ứng dụng trên mobile.