

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ - ĐHQGHN



BÁO CÁO BÀI TẬP LỚN

Đề tài: Tìm hiểu Framework Laravel 5.2. Xây dựng hệ thống quản lí nhân viên công ty.

Môn: Dự án

Lớp: INT 3509 1

Giáo viên hướng dẫn: TS. Vũ Diệu Hương

Sinh viên: Bế Văn Khánh

MSV: 12020639

Mục lục

Phần 1: Giới thiệu Framework Laravel	5
1. Laravel là gì?	5
2. Sơ lược về tính năng cơ bản của Laravel Framework	5
3. Một số ưu điểm của laravel.....	6
4. Cài đặt và tạo project Laravel 5.2.....	7
5. Cấu trúc thư mục laravel.....	8
6. Kết nối cơ sở dữ liệu trong Laravel.....	10
7. Artisan trong laravel	12
8. Route, Controller, View trong laravel.....	14
9. Mô hình MVC trong laravel (Model – View – Controller).....	19
Phần 2: Triển khai hệ thống quản lí nhân sự	22
I. Giới thiệu bài toán	22
1. Mô tả tóm tắt bài toán	22
2. Yêu cầu hệ thống.....	22
II. Phân tích, thiết kế hệ thống	24
1. Sơ đồ USE-CASE.....	24
2. Đặc tả Use-case	25
3. Biểu đồ tuần tự	35
4. Biểu đồ lớp	39
5. Kiến trúc hệ thống.....	43
III. Thiết kế giao diện	44
IV. Kết quả thực nghiệm	49
1. Môi trường phát triển:	49
2. Kết quả đạt được:.....	49
3. Hướng phát triển:.....	49
V. Tài liệu tham khảo	49

Danh mục bảng

Bảng 1: danh sách Actor.....	24
Bảng 2: danh sách các Use-case	25
Bảng 3: Danh sách lớp đối tượng và quan hệ.....	39
Bảng 4: Mô tả chi tiết từng lớp đối tượng	40
Bảng 5: Mô tả chi tiết các kiểu dữ liệu.....	40

LỜI CẢM ƠN

Để hoàn thành môn dự án này, em xin chân thành cảm ơn các thầy, cô giáo đã tận tình hướng dẫn, giảng dạy trong suốt quá trình học tập, nghiên cứu và rèn luyện ở Trường Đại học Công nghệ - ĐHQGHN.

Em xin chân thành cảm ơn giảng viên hướng dẫn TS. Vũ Diệu Hương đã tận tình hướng dẫn để em thực hiện tốt môn dự án này.

Trong suốt quá trình thực hiện, do còn hạn chế về kiến thức cũng như kinh nghiệm nên không thể tránh khỏi những thiếu sót nhất định. Em rất mong được sự góp ý của quý Thầy, Cô giáo để đề tài được hoàn chỉnh hơn.

Cuối cùng em xin kính chúc quý Thầy, Cô dồi dào sức khỏe và thành công trong sự nghiệp cao quý.

Em xin chân thành cảm ơn!

Phần 1: Giới thiệu Framework Laravel

1. Laravel là gì?

Laravel là một PHP framework mã nguồn mở và miễn phí, được phát triển bởi Taylor Otwell và nhằm vào mục tiêu hỗ trợ phát triển các ứng dụng web theo kiến trúc model-view-controller (MVC). Những tính năng nổi bật của Laravel bao gồm cú pháp dễ hiểu – rõ ràng, một 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.

Laravel phiên bản đầu tiên (Laravel 1) được phát hành vào tháng 6/2011. Đến nay, Laravel đã có 5 phiên bản, phiên bản mới nhất hiện nay là Laravel 5.2.

2. Sơ lược về tính năng cơ bản của Laravel Framework

- **Bundles:** Ở laravel phiên bản 3.x, cung cấp một hệ thống đóng gói các module, với rất nhiều tính năng đi kèm.
- **Composer:** Ở laravel phiên bản 4.x, được sử dụng như một công cụ quản lý với tính năng như thêm các gói cài đặt, các chức năng PHP phụ trợ cho Laravel có trong kho Packagist.
- **Eloquent ORM** (object relation mapping): ánh xạ các đối tượng và quan hệ cơ sở dữ liệu, cung cấp các phương thức nội bộ để thực thi đồng thời cũng bổ sung các tính năng hạn chế về mỗi

quan hệ giữa các đối tượng cơ sở dữ liệu. Eloquent ORM trình bày các bảng trong cơ sở dữ liệu dưới dạng các lớp, cung cấp thêm lựa chọn truy cập cơ sở dữ liệu trực tiếp mới mẻ hơn, chuyên nghiệp hơn.

- **Application logic:** Là một phần của phát triển ứng dụng, được sử dụng bởi bộ điều khiển controllers.
- **Routes:** Định nghĩa mối quan hệ giữa các đường dẫn (url), các liên kết (link) . Khi một liên kết được tạo ra bằng cách sử dụng tên của routes, thì một định danh liên kết thống nhất sẽ được tạo ra bởi laravel.
- **Restful Controller:** cung cấp các tùy chọn để tách các logic phía sau các request HTTP POST, GET
- **Class auto loading:** cung cấp việc tải tự động các class trong PHP, mà không cần include các class vào. Tùy thuộc vào yêu cầu các class cần thiết sẽ được nạp vào, hạn chế các class không cần thiết.
- **View:** chứa các mã html, hiển thị dữ liệu được chỉ định bởi controller
- **Migrations:** cung cấp một hệ thống kiểm soát các phiên bản lược đồ cơ sở dữ liệu (database cheme), làm cho web ứng dụng có khả năng tương tác phù hợp những thay đổi logic, các đoạn mã code của ứng dụng và những thay đổi cần thiết trong việc bố trí cơ sở dữ liệu, triển khai nói lỏng và cập nhật các ứng dụng.
- **Unit Testing:** đóng một vai trò quan trọng trong Laravel, Unit testing chứa rất nhiều các hệ thống unit testing, giúp phát hiện và ngăn chặn lỗi trong khuôn khổ nhất định. Unit Testing có thể được chạy thông qua tiện ích command-line.
- **Automatic pagination:** Tính năng tự động phân trang được tích hợp vào Laravel giúp đơn giản hóa các nhiệm vụ thực hiện phân trang so với các phương pháp thông thường.

3. Một số ưu điểm của laravel

- Laravel ra đời khá muộn nên được thừa hưởng những ưu điểm và thế mạnh của các framework khác.
- Có số lượng người sử dụng nhiều nhất.
- Document rõ ràng, dễ học.

- Autoload theo namespace.
- Sử dụng mô hình ORM rất đơn giản khi thao tác với Database.
- Các lệnh tương tác với cơ sở dữ liệu cực kỳ ngắn gọn và thân thiện.
- Việc quản lý layout đơn giản với Blade Templating.
- Dễ dàng tích hợp các thư viện khác vào dự án, và được quản lý với Composer.

4. Cài đặt và tạo project Laravel 5.2

❖ Yêu cầu:

- PHP \geq 5.5.9
- OpenSSL PHP Extension
- PDO PHP Extension
- Mbstring PHP Extension
- Tokenizer PHP Extension

❖ Kiến nghị trước khi cài đặt

- Cài đặt Composer [Composer-Setup.exe](#)
- Cài đặt webserver trên window có thể cài đặt một số phần mềm như Openserver, Wamp, Xampp, Ampps, vertrigo ...

❖ Cài đặt laravel 5.2

Mở cmd chạy lệnh *composer global require "laravel/installer"* . Sau khi máy load các thư viện của laravel 5.2, hoàn tất việc cài đặt.

❖ Tạo folder chứa project laravel

Mở cmd chỉ đường dẫn tới nơi chứa folder chạy lệnh: *composer create-project --prefer-dist laravel/laravel **project-name*** . với **project-name** là tên folder chứa project laravel.

```
composer create-project --prefer-dist laravel/laravel project-name

E:\Laravel>composer create-project --prefer-dist laravel/laravel project-name
Installing laravel/laravel (v5.2.29)
- Installing laravel/laravel (v5.2.29)
  Loading from cache

Created project in project-name
> php -r "copy('.env.example', '.env');"
Loading composer repositories with package information
Installing dependencies (including require-dev)
- Installing vlucas/phpdotenv (v2.2.1)
  Loading from cache

- Installing symfony/polyfill-mbstring (v1.1.1)
  Loading from cache

- Installing symfony/var-dumper (v3.0.4)
  Loading from cache

- Installing symfony/translation (v3.0.4)
  Loading from cache

- Installing symfony/routing (v3.0.4)
  Loading from cache
```

5. Cấu trúc thư mục laravel

Name	Date modified	Type	Size
app	24/04/2016 6:25 PM	File folder	
bootstrap	24/04/2016 6:25 PM	File folder	
config	24/04/2016 6:25 PM	File folder	
database	24/04/2016 6:25 PM	File folder	
public	24/04/2016 6:25 PM	File folder	
resources	24/04/2016 6:25 PM	File folder	
storage	24/04/2016 6:25 PM	File folder	
tests	24/04/2016 6:25 PM	File folder	
vendor	24/04/2016 6:28 PM	File folder	
.env	24/04/2016 6:28 PM	ENV File	1 KB
.env.example	24/04/2016 6:25 PM	EXAMPLE File	1 KB
.gitattributes	24/04/2016 6:25 PM	Text Document	1 KB
.gitignore	24/04/2016 6:25 PM	Text Document	1 KB
artisan	24/04/2016 6:25 PM	File	2 KB
composer.json	24/04/2016 6:25 PM	JSON File	2 KB
composer.lock	24/04/2016 6:27 PM	LOCK File	106 KB
gulpfile.js	24/04/2016 6:25 PM	JavaScript File	1 KB
package.json	24/04/2016 6:25 PM	JSON File	1 KB
phpunit.xml	24/04/2016 6:25 PM	XML File	2 KB
readme.md	24/04/2016 6:25 PM	MD File	2 KB
server.php	24/04/2016 6:25 PM	PHP File	1 KB

- **app:** Chứa code ứng dụng
 - Thư mục App\Http\Controllers chứa các controller.
 - Tập tin App\Http/routes.php: tập tin này lưu trữ toàn bộ route trong ứng dụng để giúp Laravel biết cách kết nối khi có yêu cầu, theo hàm khai sẵn, controller và action...
- **bootstrap:** Chứa một số file của bootstrap framework và một số file cấu hình nạp tự động.
- **Config:** chứa tất cả các file cấu hình của hệ thống.
- **Database:** Chứa *migration* và *seed* cơ sở dữ liệu.
 - Thư mục *migration* chứa những lớp PHP mà cho phép Laravel cập nhật Schema cơ sở dữ liệu mà vẫn giữ được toàn bộ versions cơ sở dữ liệu trong khi đồng bộ. Những file migration được tạo bởi tool Artisan.
 - Thư mục *seeds* chứa những file PHP cho phép Artisan đưa vào bảng cơ sở dữ liệu với dữ liệu được đề xuất.
- **public:** Chứa các front controller và các tài nguyên khác (hình ảnh, javascript, css,...)
- **resources:** Chứa các view, các raw asset và các file ngôn ngữ.
 - Thư mục *views* chứa những file mẫu HTML mà được sử dụng bởi controller hay route. Chú ý rằng chỉ nên đặt những file mẫu view tại vị trí này vì những file css, javascript đã được đặt trong thư mục public.
 - Thư mục *lang* chứa những file PHP chứa mảng của những chuỗi cho phép dễ dàng xác định vị trí của ứng dụng. Mặc định thư mục này chứa những dòng ngôn

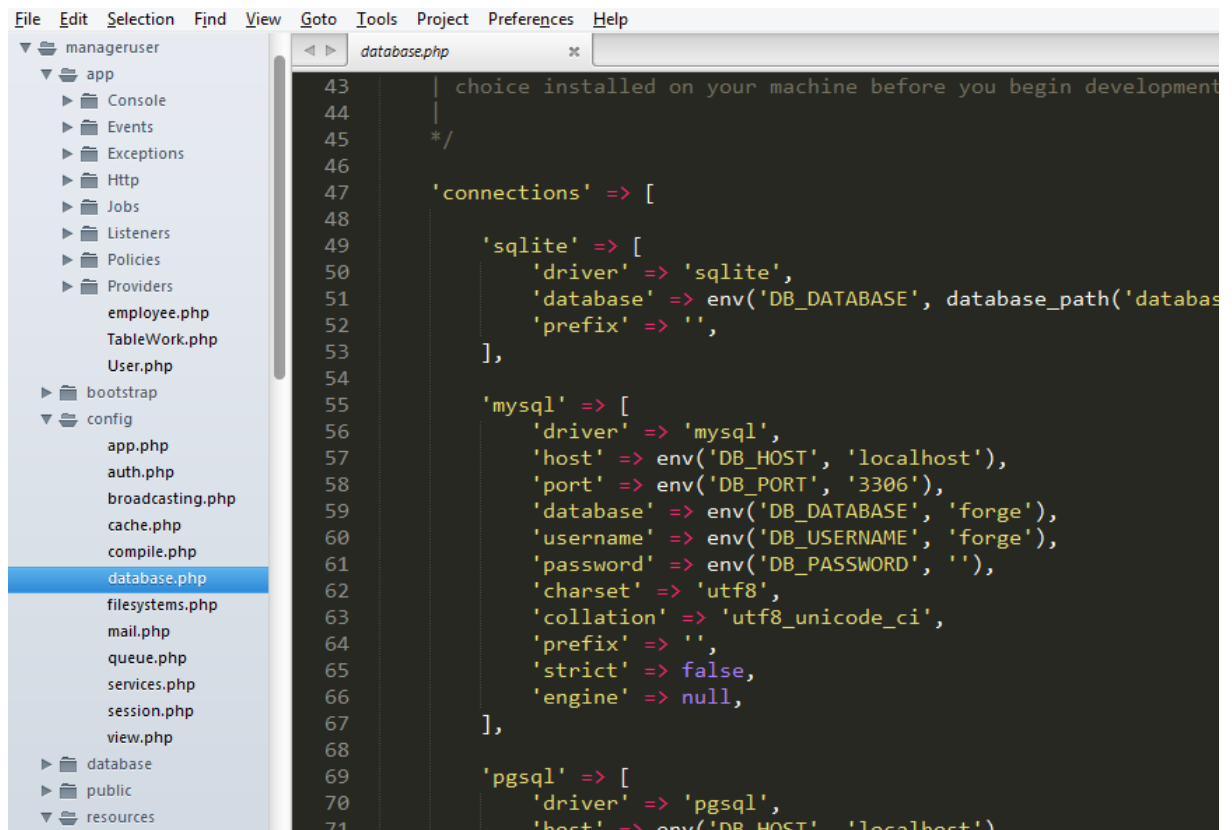
ngữ cho việc phân trang và chứng thực form người dùng với ngôn ngữ là tiếng Anh.

- **Storage:** Chứa các khuôn mẫu Blade, các file session, các file cache và các file khác được sinh ra bởi framework.
- **Tests:** Chứa các test tự động.
- **Vendor:** Chứa các composer package sử dụng bởi ứng dụng.

6. Kết nối cơ sở dữ liệu trong Laravel

Laravel kết nối tới cơ sở dữ liệu và chạy truy vấn cực kỳ đơn giản. Cấu hình nằm tại file **config/database.php**. Trong file này định nghĩa tất cả những loại kết nối cơ sở dữ liệu.

Hiện tại thì Laravel hỗ trợ những hệ quản trị cơ sở dữ liệu: MySQL, SQLite, pgSQL. Trong đó cấu hình mặc định sử dụng là MySQL.



Để kết nối tới cơ sở dữ liệu trên webserver cần cấu hình lại file **.env** trong folder project. Với:

DB_CONNECTION=tên hệ quản trị cơ sở dữ liệu

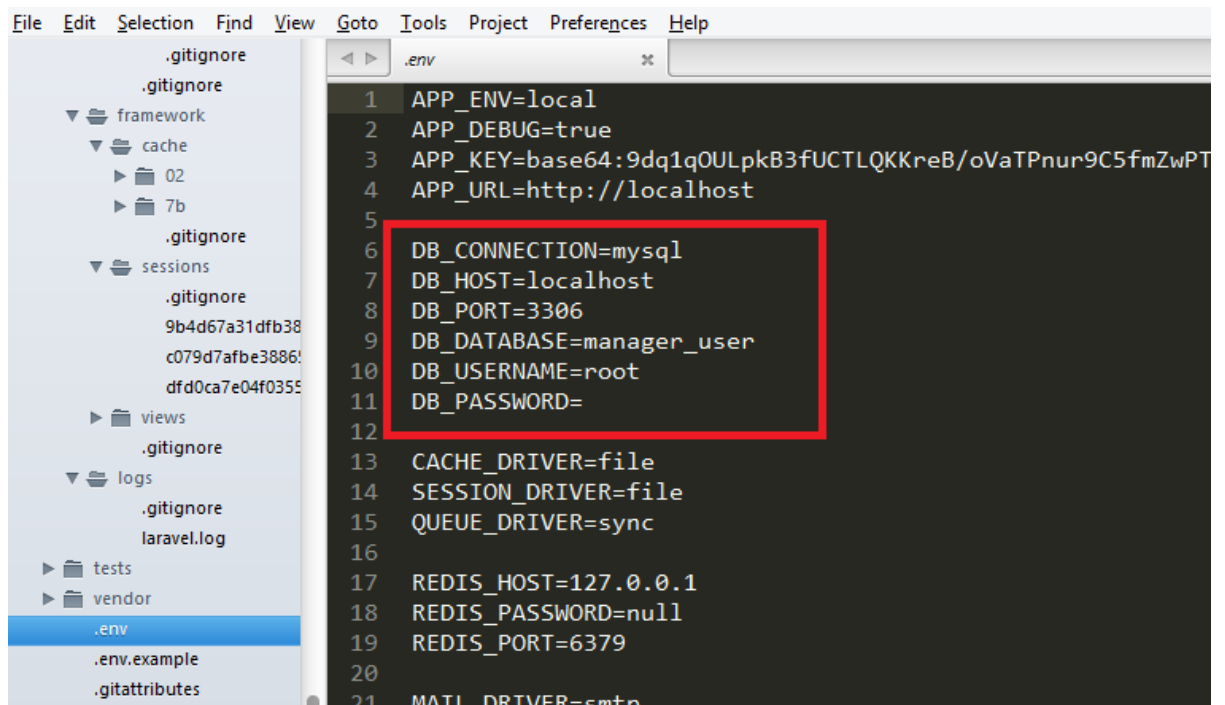
DB_HOST=tên host

DB_PORT=tên cổng

DB_DATABASE=tên database

DB_USERNAME=tên username

DB_PASSWORD=mật khẩu



7. Artisan trong laravel

Artisan là một CLI được tích hợp trong Laravel, cung cấp một số lệnh (command) hữu dụng trong khi phát triển ứng dụng. Để xem tất cả các lệnh của Artisan các bạn gõ lệnh này trong cmd:

php artisan

```
C:\Windows\system32\cmd.exe

E:\Laravel\project-name>php artisan
Laravel Framework version 5.2.30

Usage:
  command [options] [arguments]

Options:
  -h, --help            Display this help message
  -q, --quiet            Do not output any message
  -V, --version          Display this application version
      --ansi             Force ANSI output
      --no-ansi          Disable ANSI output
  -n, --no-interaction  Do not ask any interactive question
      --env[=ENV]        The environment the command should run under.
  -vvvvvv, --verbose    Increase the verbosity of messages: 1 for normal output,
                        2 for more verbose output and 3 for debug

Available commands:
  clear-compiled  Remove the compiled class file
  down           Put the application into maintenance mode
  env            Display the current framework environment
  help           Displays help for a command
  list           Lists commands
  migrate        Run the database migrations
  optimize       Optimize the framework for better performance
  serve         Serve the application on the PHP development server
  tinker         Interact with your application
  up            Bring the application out of maintenance mode
  app
  app:name       Set the application namespace
  auth
  auth:clear-resets  Flush expired password reset tokens
  cache
  cache:clear     Flush the application cache
  cache:table     Create a migration for the cache database table
  config
  config:cache    Create a cache file for faster configuration loading
  config:clear    Remove the configuration cache file
  db
  db:seed        Seed the database with records
  event
  event:generate  Generate the missing events and listeners based on registr
ation
  key
  key:generate    Set the application key
  make
  make:auth       Scaffold basic login and registration views and routes
  make:console    Create a new Artisan command
  make:controller Create a new controller class
  make:event      Create a new event class
  make:job        Create a new job class
  make:listener   Create a new event listener class
  make:middleware Create a new middleware class
  make:migration  Create a new migration file
  make:model      Create a new Eloquent model class
  make:policy     Create a new policy class
```

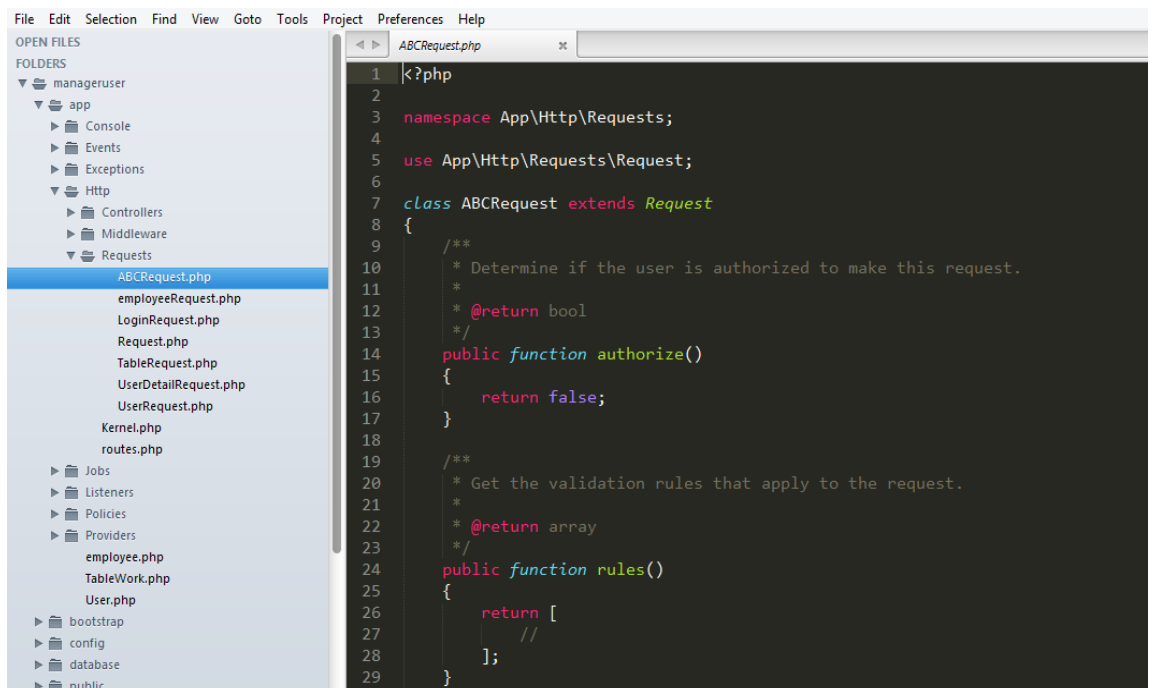
Ví dụ: Tạo một Request

Tại thư mục project giữ Shift kích chuột phải chọn *Open command window here* sau đó chạy dòng lệnh.

php artisan make :request ABCrequest

Kết quả : tạo ra 1 request có đường dẫn app/http/request/ABCrequest.php tự động import các thư viện cần thiết của Laravel và một số phương thức mặc định dùng

trong request giúp cho việc làm việc với request trở nên nhanh chóng và dễ dàng hơn.



8. Route, Controller, View trong laravel

- ❖ **Route:** có vai trò chỉ đường cho yêu cầu (request). Khi web nhận được request từ người dùng, nó sẽ đưa request này cho route, từ route sẽ chỉ dẫn tiếp cho request này đi đến đâu hoặc nó có thể trả lời ngay tại route.

Ví dụ khi người dùng truy cập vào website với đường dẫn là domain.com/login, thì lúc này Route sẽ nhận được một phần trong URL là /login và nó sẽ xử lý request đó gọi đến Controller Login hoặc là một thao tác khác (xử lý thế nào thì tùy vào coder).

Hầu hết các route của laravel định nghĩa trong file app/routes.php. Danh sách các loại Route bao gồm các loại cơ bản và thường được sử dụng sau:

- `Route::get`: phương thức GET.
- `Route::post`: phương thức POST.
- `Route::match`: kết hợp nhiều phương thức.
- `Route::any`: nhận tất cả các phương thức.
- `Route::filter`: tạo ra bộ lọc nhằm mục đích nào đó, ví dụ tạo bộ lọc kiểm tra đã đăng nhập hay chưa.
- `Route::group`: gom các route lại với nhau thành một nhóm.
- `Route::controller`: gọi đến controller tương ứng.
- `Route::resource`: sử dụng với resource controller.

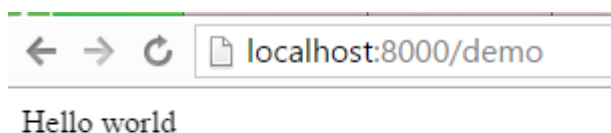
Cú pháp: `Route::get($uri, $action);`

trong đó `$action` có thể là 1 mảng, hàm hoặc 1 chuỗi

Ví dụ: mở file `app/http/route.php` gõ đoạn code sau:

```
18 Route::get('demo', function(){
19     return "Hello world";
20 });
21
```

Kết quả:



Khi action là một mảng: có 2 tham số được dùng trong mảng là **name** và **uses**. Trong đó name là tên của route mình sẽ đặt, còn uses sẽ gọi đến phương thức của Controller.

Ví dụ: mở file `app/http/route.php` gõ đoạn code sau:

```
Route::get('/array',['name'=>'get.array','uses'=>'HomeController@showWelcome']);
```

Trong đó

/array là uri

‘get.array’ là tên của route này

uses route này sẽ gọi đến phương thức showWelcome của HomeController

❖ Controller

Controller chứa các phương thức, các câu lệnh xử lý dữ liệu trước khi đưa ra view.

Tất cả Controller trong laravel được đặt trong thư mục
app/http/Controllers

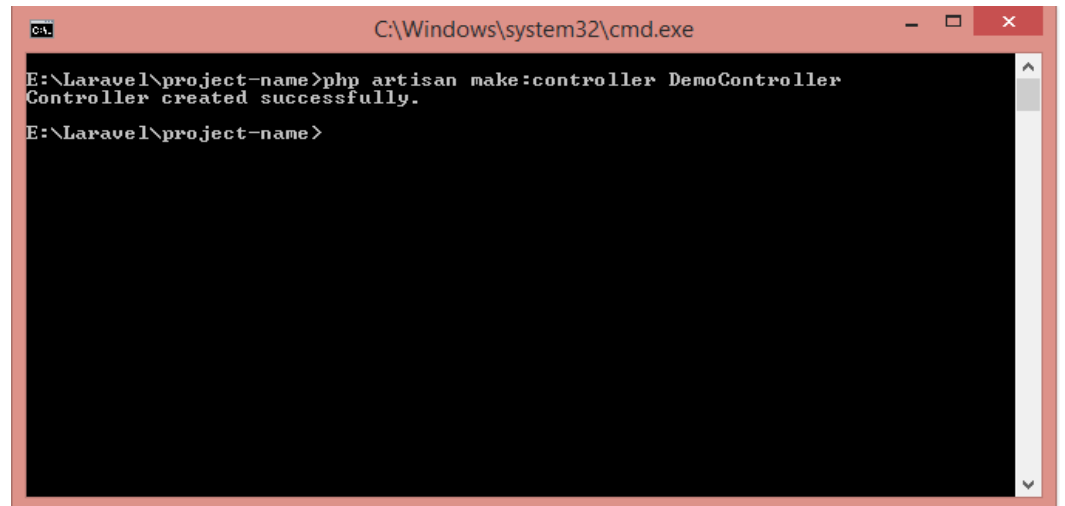
Việc tạo tên file và tên controller phải tuân theo nguyên tắc:

- Trong app/http/Controllers/ có một controller.php được tạo sẵn, đây là một lớp xử lý controller chính của laravel nên tất cả các Controller mới phải kế thừa (extends) từ lớp này.
- Nếu muốn kế thừa từ một Controller khác thì lớp đó phải kế thừa từ lớp Controller của Laravel.
- Tên Controller phải trùng với tên file. Ví dụ: Controller tên DemoController thì file phải tên là DemoController.php.

Ngoài cách tạo controller thủ công Laravel hỗ trợ tạo controller với Command line mở cmd dẫn tới thư mục chứa project chạy lệnh sau:

```
php artisan make:controller Tên-controller
```

Ví dụ: mở cmd chỉ đường dẫn tới project

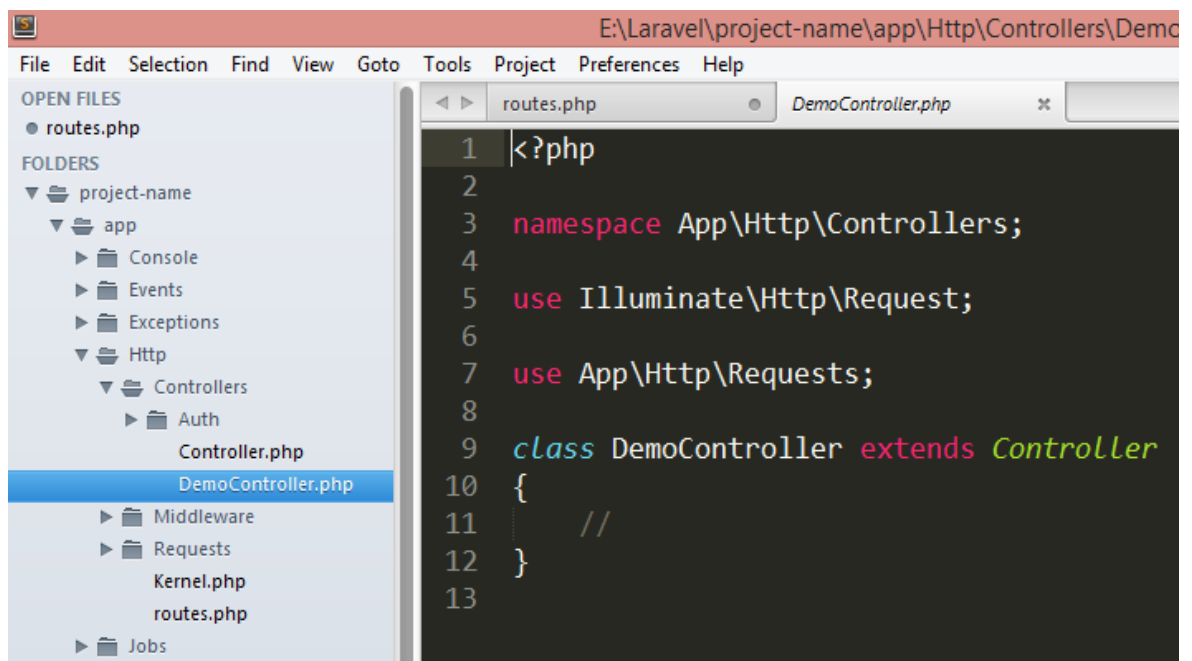


```
C:\Windows\system32\cmd.exe

E:\Laravel\project-name>php artisan make:controller DemoController
Controller created successfully.

E:\Laravel\project-name>
```

file DemoController được tạo ra

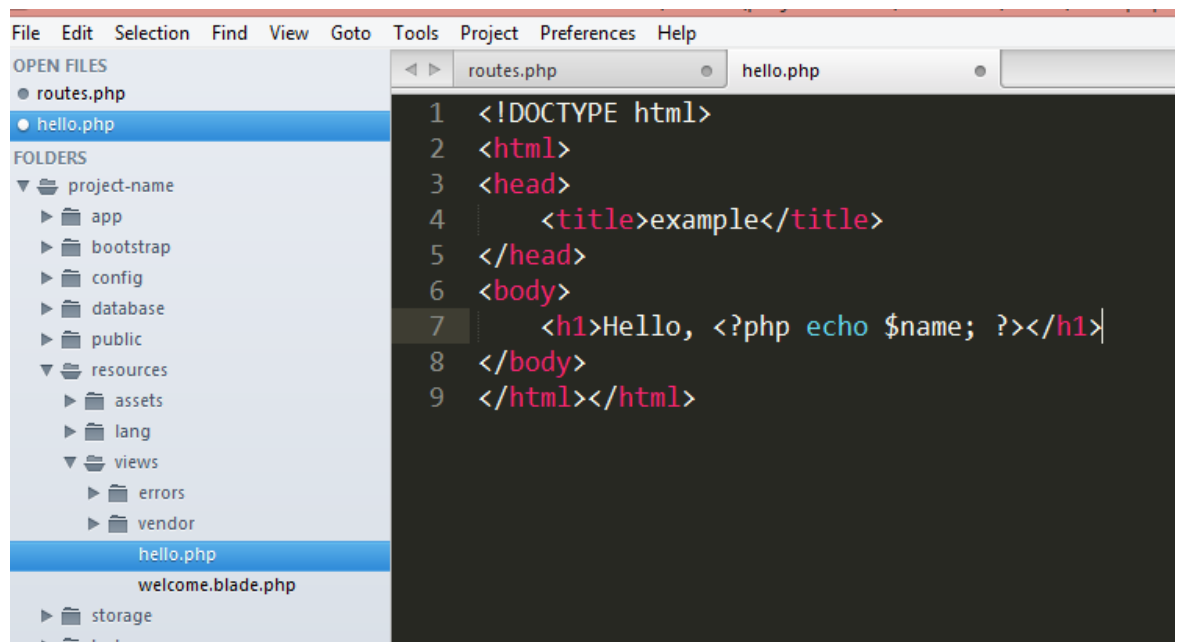


❖ View

View chịu trách nhiệm nhận dữ liệu đầu, cuối và hiển thị ra giao diện.

Tất cả view trong laravel mặc định lưu trong **resources/view**

Ví dụ: tạo file hello.php có đường dẫn `resources/view/hello.php`



Sau đó trong app/http/route.php thực gõ code sau:

```
25
26 Route::get('/hello', function () {
27     return view('hello', ['name' => 'Khánh']);
28 });
29
```

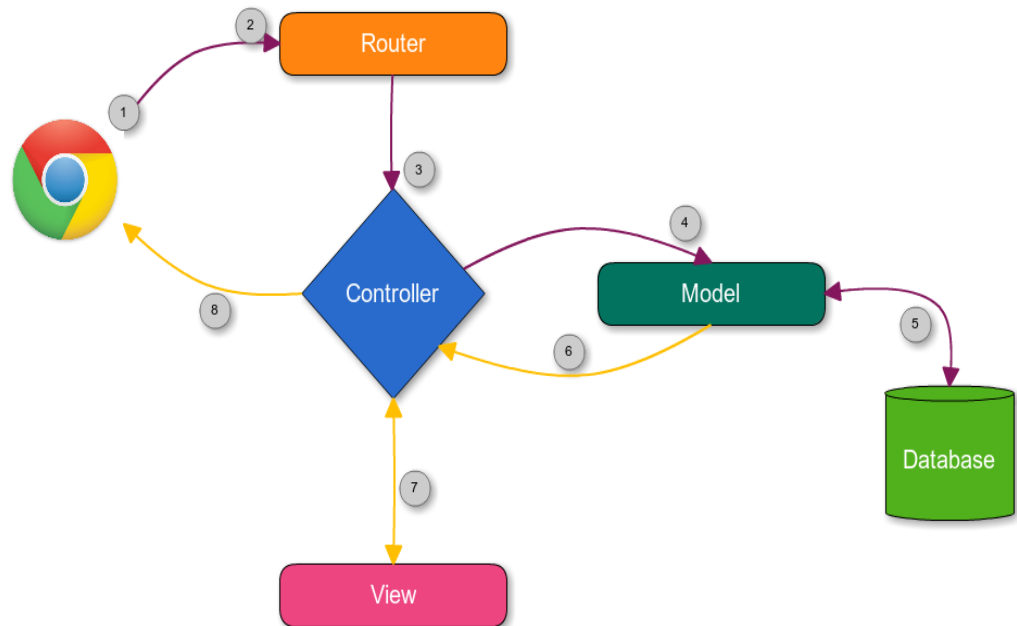
Kết quả:



Hello, Khánh

Như đã thấy, hàm `view()` nhận vào hai tham số. Tham số thứ nhất chính là tên của view, chỉ cần gọi tên, không cần thêm vào đuôi “.php”. Tham số thứ hai là một mảng để truyền dữ liệu đã “xử lý” vào view; nó là một mảng, với key là tên của biến sẽ sử dụng ở view và value là giá trị của biến đó. Như ví dụ trên, biến `$name` ở view được gán giá trị “Khánh” ở route.

9. Mô hình MVC trong laravel (Model – View – Controller)



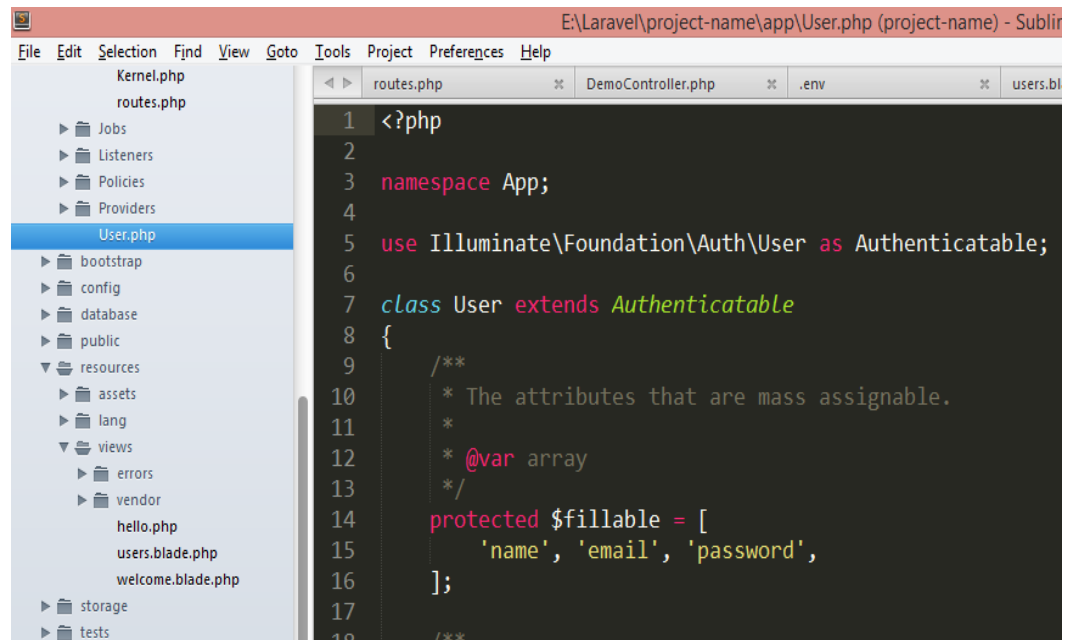
Hình 2: Mô hình MVC trong laravel

Nội dung:

- Người dùng sử dụng trình duyệt web để truy cập vào địa chỉ website.
- Route sẽ phân tích xem đường dẫn truy cập đến Controller nào.
- Controller sẽ truy cập vào Model để lấy dữ liệu từ Database, sau đó trả về controller.
- Controller đổ dữ liệu ra View hiện thị ra giao diện cho người dùng.

Demo:

Sử dụng model **app/User.php** có sẵn trong project laravel



Trong file app/http/route.php thực hiện đoạn code tạo uri là user và truyền vào id. Sử dụng controller là DemoController với phương thức showview.

```
2
3 Route::get('/user/{id}', 'DemoController@showview');
4
```

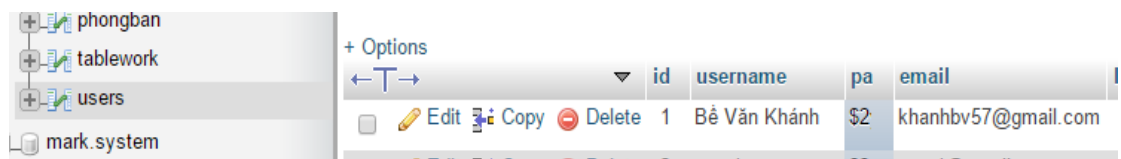
Trong controller app/http/Controllers/DemoController.php tạo phương thức showview với id được truyền vào
Tạo biến \$user = User::find(\$id);
Users ở đây là tên file trong resources/view/users.blade.php

```
10
11 class DemoController extends Controller
12 {
13     public function showview($id){
14         $user = User::find($id);
15         return view('users')->with('user',$user);
16     }
17 }
18
```

Trong file users.blade.php thực hiện gõ đoạn code để hiện thị thông tin ra màn hình như sau:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title></title>
5 </head>
6 <body>
7     <ul>
8         <li>Name: <strong>{{ $user->username }}</strong></li>
9         <li>Email: <strong>{{ $user->email }}</strong></li>
10    </ul>
11 </body>
12 </html>
```

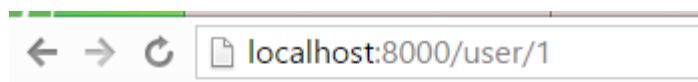
Trên database insert dữ liệu để test thử.



The screenshot shows a database interface with a table named 'users'. The table has four columns: 'id', 'username', 'password', and 'email'. There is one record in the table with the following values: id=1, username='Bê Văn Khánh', password='\$2', and email='khanhbv57@gmail.com'.

id	username	password	email
1	Bê Văn Khánh	\$2	khanhbv57@gmail.com

Tiến hành test thử:



- Name: **Bê Văn Khánh**
- Email: **khanhbv57@gmail.com**

Phần 2: Triển khai hệ thống quản lí nhân sự

I. Giới thiệu bài toán

1. Mô tả tóm tắt bài toán

Hiện nay ở công ty A có hơn 300 nhân viên, làm việc tại 3 tầng của tòa nhà. Mỗi bàn làm việc đều có một máy điện thoại bàn để liên lạc nội bộ trong công ty. Tuy nhiên hiện nay việc quản lý chỗ ngồi và số điện thoại nội bộ của công ty mới chỉ thực hiện được thủ công trên Excel. Mỗi khi có nhân viên nào đó đổi bàn làm việc hoặc có nhân viên mới vào công ty, việc cập nhật thông tin về chỗ ngồi và số điện thoại đều phải thực hiện bằng tay. Ngoài ra nếu một khách hàng hoặc một nhân viên muốn tìm số máy bàn của một nhân viên khác thì sẽ phải tìm lần lượt trong các sheet trong file Excel, điều này làm mất thời gian và khó khăn cho người tìm kiếm. Hệ thống quản lí số điện thoại nội bộ của công ty sẽ giúp khắc phục các vấn đề này.

2. Yêu cầu hệ thống

❖ Chức năng chính

- Cập nhật thông tin nhân viên:

- *Quản trị viên: quản trị viên có quyền tạo, thay đổi thay đổi thông tin cá nhân của các nhân viên dưới quyền trong công ty như: Họ tên, chức danh, mã nhân viên, phòng, số điện thoại nội bộ, địa chỉ email...*

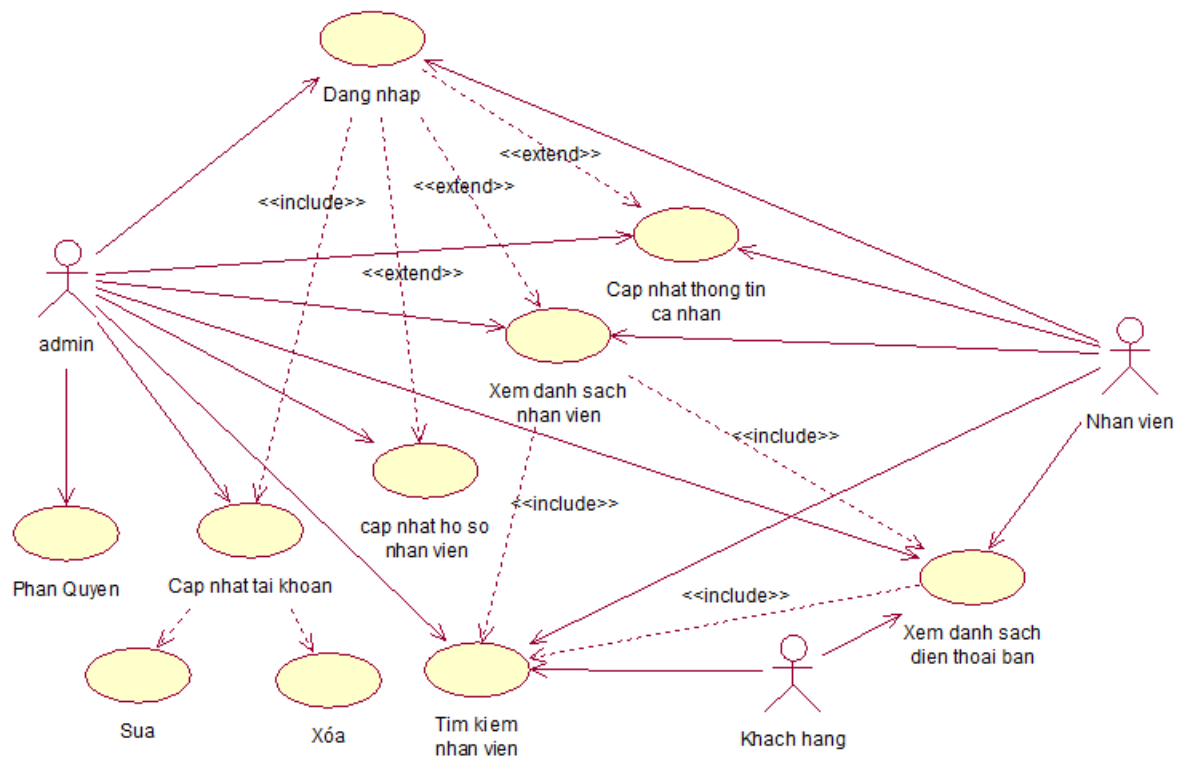
- *Nhân viên: nhân viên trong công ty có quyền đăng nhập thay đổi thông tin cá nhân của mình như email, số điện thoại cá nhân, địa chỉ ...*
- *Tìm kiếm: Hệ thống cho phép người sử dụng có thể tìm kiếm nhân viên theo mã nhân viên, tên nhân viên, chức danh.*
- *Xem danh sách điện thoại bàn: Hệ thống hiển thị danh sách điện thoại bàn nội bộ của công ty bao gồm: Mã bàn làm việc, số điện thoại bàn, Mã nhân viên, tên nhân viên, chức vụ.*
- *Xem Thông tin nhân viên: Hệ thống có chức năng hiển thị danh sách thông tin nhân viên như: mã nhân viên, họ tên nhân viên, phòng ban, chức vụ, số điện bàn phụ trách, số điện thoại cá nhân, email...*
- *Xem sơ đồ công ty: Hệ thống có chức năng hiển thị sơ đồ công ty dưới dạng file ảnh.*

❖ Yêu cầu

- *Giao diện trực quan, tiện dụng.*
- *Có thể sử dụng 24/24, đáp ứng nhiều lượt truy cập một lúc.*
- *Hệ thống chạy trên nền web, người dùng truy cập thông qua trình duyệt.*
- *Có khả năng bảo mật, phân quyền truy cập tốt, mỗi đối tượng người dùng chỉ có thể sử dụng một số chức năng riêng.*

II. Phân tích, thiết kế hệ thống

1. Sơ đồ USE-CASE



Bảng 1: danh sách Actor

STT	Tên Actor	Ý nghĩa
1	Admin	Người có quyền cao nhất trong hệ thống
3	Nhân viên	Nhân viên làm việc trong công ty
4	Khách hàng	Khách hàng làm việc với công ty

Bảng 2: danh sách các Use-case

STT	Use case	Ý nghĩa
1	Đăng nhập	Đăng nhập vào hệ thống
2	Tạo tài khoản	Tạo tài khoản nhân viên để có thể đăng nhập vào hệ thống và phân quyền truy cập.
3	Cập nhật tài khoản	Sửa, xóa thông tin tài khoản.
2	Cập nhật hồ sơ nhân viên	Tạo, sửa, cập nhật thông tin của các nhân viên trong công ty
3	Cập nhật thông tin cá nhân	Sửa đổi thông tin cá nhân của chính nhân viên đang đăng nhập trong hệ thống
4	Xem danh sách nhân viên	Xem danh sách nhân viên
5	Xem danh sách điện thoại bàn	Xem danh sách điện thoại bàn
6	Tìm kiếm nhân viên	Tìm kiếm nhân viên
7	Phân quyền	Phân quyền các User trong hệ thống để đảm bảo các User chỉ thực hiện được một số chức năng riêng của mình.

2. Đặc tả Use-case

2.1 Use case “ Đăng nhập”

❖ Mô tả tóm tắt:

- *Admin, Quản lí, nhân viên sử dụng use-case.*
- *Dùng use-case để đăng nhập vào hệ thống.*
- *Use-case dùng để đăng nhập vào hệ thống và phân quyền.*

❖ Dòng sự kiện chính:

- *Use-case bắt đầu khi có một User đăng nhập vào hệ thống.*

- Hệ thống yêu cầu người dùng nhập Username và password.
- Người dùng nhập Username và Password.
- Hệ thống kiểm tra username và password đăng nhập và cho phép người dùng đăng nhập vào hệ thống.

❖ Dòng sự kiện khác

- Tên hoặc mật khẩu sai: Nếu người dùng đăng nhập sai username hoặc password thì hệ thống sẽ thông báo lỗi. Người dùng có thể quay lại đăng nhập hoặc kết thúc đăng nhập.

❖ Yêu cầu đặc biệt

- Yêu cầu khi đăng nhập phải thực hiện nhập đúng tên và mật khẩu
- Để đảm bảo tính an toàn cho hệ thống, mỗi user chỉ được nhập lại tối đa 3 lần. Nếu sai tiếp hệ thống sẽ tự kết thúc đăng nhập.

❖ Trạng thái hệ thống trước khi thực hiện use-case

- Không có trạng thái nào.

❖ Trạng thái hệ thống sau khi thực hiện use-case

- Khi đăng nhập thành công, hệ thống sẽ chuyển đến trang chủ, các chức năng của người dùng sẽ sử dụng được.
- Khi đăng nhập thất bại, không đăng nhập được vào hệ thống sẽ không thực hiện được các chức năng.

❖ Phần mở rộng:

- Không có

2.2 Use-case “Tạo tài khoản”

❖ Mô tả tóm tắt

- Admin sử dụng use-case.
- Dùng use-case tạo tài khoản người dùng để đăng nhập và phân quyền user khi truy cập vào hệ thống.

❖ Dòng sự kiện chính:

- Khi admin chọn chọn tạo tài khoản hệ thống hiển thị form tạo tài khoản, admin tiến hành nhập thông tin cần thiết và chọn quyền đăng nhập cho tài khoản.
- Sau khi nhập tên, mật khẩu và cấp quyền xong, người dùng chọn submit hệ thống sẽ lưu thông tin về user vừa nhập vào database.

❖ Dòng sự kiện khác

- Nếu thông tin nhập là chính xác khi submit hệ thống sẽ đưa ra thông báo là đã tạo thành công. Ngược lại nếu thông tin nhập là sai hệ thống sẽ đưa ra thông báo và hiển thị tên lỗi.

❖ Yêu cầu đặc biệt

- Không có.

❖ Trạng thái hệ thống trước khi bắt đầu use-case

- Admin đăng nhập thành công vào hệ thống.

❖ Trạng thái hệ thống sau khi thực hiện use-case

- Sau khi phân quyền thành công hệ thống sẽ lưu thông tin vào cơ sở dữ liệu.

❖ Phân mở rộng

- Không có.

2.3 Ca sử dụng “Sửa tài khoản”

❖ Mô tả tóm tắt

- *Admin sử dụng use-case.*
- *Khi người dùng (nhân viên) quên mật khẩu, tài khoản đăng nhập yêu cầu xin cấp lại, admin sẽ sử dụng use-case để cập nhật lại tài khoản.*

❖ Dòng sự kiện chính

- *Admin chọn button “Sửa” hệ thống sẽ hiển thị form sửa tài khoản.*
- *Admin tiến hành nhập thông tin sau đó chọn submit để xác nhận.*

❖ Dòng sự kiện khác

- *Nếu thông tin nhập là chính xác hệ thống sẽ hiển thị thông báo sửa thành công và chuyển đến trang danh sách tài khoản để người dùng có thể kiểm tra.*
- *Nếu thông tin nhập không chính xác hệ thống sẽ hiển thị thông báo lỗi và yêu cầu người dùng nhập lại hoặc chọn kết thúc use-case.*

❖ Yêu cầu đặc biệt

- *Không có.*

❖ Trạng thái hệ thống trước khi bắt đầu use-case

- *Admin phải đăng nhập thành công vào hệ thống.*

❖ Trạng thái hệ thống sau khi kết thúc use-case

- *Hệ thống sẽ cập nhật lại tài khoản và chuyển đến danh sách tài khoản để người dùng có thể kiểm tra.*

2.4 Ca sử dụng “Xóa tài khoản”

❖ Mô tả tóm tắt

- *Admin sử dụng use-case.*
- *Khi người dùng không còn là nhân viên của công ty, admin tiến hành xóa bỏ tài khoản của nhân viên.*

❖ Dòng sự kiện chính

- *Admin chọn button “Xóa” hệ thống sẽ hiển thị thông báo để người dùng xác nhận. Chọn “Có” hệ thống sẽ tiến hành xóa tài khoản và tất cả các thông tin liên quan. Chọn “không” để hủy việc xóa.*

❖ Dòng sự kiện khác

- *Không có.*

❖ Yêu cầu đặc biệt

- *Không có.*

❖ Trạng thái hệ thống trước khi bắt đầu use-case

- *Admin phải đăng nhập thành công vào hệ thống.*

❖ Trạng thái hệ thống sau khi kết thúc use-case

- *Thông tin tài khoản sẽ được xóa khỏi cơ sở dữ liệu,*

2.5 Use-case “ Cập nhật hồ sơ nhân viên”

❖ Mô tả tóm tắt

- *admin sử dụng use-case.*
- *Use-case này mô tả việc người dùng tạo một nhân viên mới, thay đổi thông tin của nhân viên khi thông tin sai hay nhân viên thay đổi vị trí, chức vụ công tác hoặc xóa hồ sơ nhân viên khi nhân viên đó nghỉ việc.*

❖ Dòng sự kiện chính

- *Use-case này bắt đầu khi người dùng thêm một nhân viên mới hoặc sửa thông tin nhân viên.*

- Hệ thống hiển thị danh sách nhân viên công ty, người dùng tiến hành chọn chức năng tương ứng như tạo mới, sửa, xóa nhân viên.
- Sau khi điền đầy đủ thông tin cần thiết, người dùng tiến hành chọn “save” để hoàn tất thao tác.
- Hệ thống sẽ kiểm tra tính hợp lệ của các thông tin.
- Nếu không hợp lệ, hệ thống sẽ báo lỗi và người dùng sẽ tiến hành chỉnh lại.
- Nếu thông tin hợp lệ, hệ thống sẽ tiến hành cập nhật thông tin nhân viên. Nhân viên này sẽ được tự động sắp xếp tăng dần theo bảng chữ cái.

❖ Dòng sự kiện khác

- Thông tin về nhân viên không đầy đủ: Nếu các thông tin nhập vào không đầy đủ, hệ thống sẽ thông báo lỗi, thiếu các thông tin cần thiết và yêu cầu bổ sung đầy đủ thông tin. Người sử dụng có thể bổ sung tiếp thông tin hoặc hủy bỏ thao tác thực hiện, lúc này use-case sẽ kết thúc.
- Thông tin về nhân viên không hợp lệ: Nếu các thông tin nhập vào không hợp lệ, hệ thống sẽ thông báo lỗi và yêu cầu chỉnh sửa lại thông tin. Người sử dụng hệ thống có thể chỉnh sửa hoặc tiến hành hủy thao tác thực hiện, lúc này use-case sẽ kết thúc.

❖ Yêu cầu đặc biệt

- Không có.

- ❖ Trạng thái hệ thống trước khi thực hiện Use-case
 - *Người sử dụng use-case phải thực hiện đăng nhập thành công với vai trò là quản lí hoặc admin.*
- ❖ Trạng thái hệ thống sau khi thực hiện Use-case
 - *Nếu Use-case thành công thì nhân viên sẽ được cập nhật mới vào hệ thống. Ngược lại trạng thái của hệ thống sẽ không thay đổi.*
- ❖ Phần mở rộng
 - *Không có.*

2.6 Use-case “ Cập nhật thông tin cá nhân”

- ❖ Mô tả tóm tắt
 - *Use-case được sử dụng bởi các User đăng nhập vào hệ thống.*
 - *Use-case dùng để thay đổi một số thông tin cá nhân không cố định của chính nhân viên đang đăng nhập như email, số điện thoại, địa chỉ, ngày sinh...*
- ❖ Dòng sự kiện chính
 - *Use-case này bắt đầu khi người dùng muốn thay đổi thông tin cá nhân của mình.*
 - *Người dùng chọn chức năng thay đổi thông tin cá nhân.*
 - *Sau khi điền các thông tin chỉnh sửa, người dùng tiến hành chọn nút “save”.*
 - *Hệ thống sẽ kiểm tra tính hợp lệ của thông tin.*
 - *Nếu không hợp lệ hệ thống sẽ báo lỗi và yêu cầu nhập lại.*

- Nếu hợp lệ thông tin về nhân viên sẽ được cập nhật vào hệ thống. Sau đó hệ thống sẽ quay lại màn hình chính.

❖ Dòng sự kiện khác

- Thông tin chỉnh sửa về nhân viên không hợp lệ: hệ thống sẽ thông báo lỗi và yêu cầu người dùng chỉnh sửa lại các thông tin không hợp lệ. Người dùng có thể tiến hành chỉnh sửa lại hoặc hủy bỏ thao tác thực hiện, lúc này use-case sẽ kết thúc.

❖ Yêu cầu đặc biệt

- Người sử dụng use-case chỉ có thể chỉnh sửa một số thông tin không cố định mà hệ thống cho phép như email, số điện thoại cá nhân, địa chỉ, tôn giáo, dân tộc ..

❖ Trạng thái hệ thống trước khi bắt đầu thực hiện use-case

- Người sử dụng use-case này phải đăng nhập vào hệ thống với vai trò là nhân viên của công ty.

❖ Trạng thái hệ thống sau khi thực hiện use-case

- Nếu use-case thực hiện thành công thì thông tin cá nhân của nhân viên sẽ được cập nhật vào hệ thống và giao diện sẽ quay lại màn hình chính của use-case. Ngược lại, trạng thái của hệ thống sẽ không đổi.

❖ Phần mở rộng

- Không có.

2.7 Use-case “ Xem danh sách điện thoại bàn ”

❖ Mô tả tóm tắt

- Admin, quản lí, nhân viên sử dụng use-case.
- Use-case dùng để xem danh sách điện thoại bàn trong công ty.

❖ Dòng sự kiện chính

- Use-case này bắt đầu khi người dùng muốn xem danh sách điện thoại bàn nội bộ trong công ty.
- Hệ thống liệt kê bảng danh sách nhân viên mới với đầy đủ các thông tin như số điện thoại bàn, tên nhân viên phụ trách, mã nhân viên phụ trách, chức danh nhân viên, phòng ban.

❖ Dòng sự kiện khác

- Không có.

❖ Yêu cầu đặc biệt

- Không có.

❖ Trạng thái hệ thống trước khi bắt đầu use-case

- Người sử dụng use-case phải đăng nhập vào hệ thống với vai trò nhất định.

❖ Trạng thái hệ thống sau khi thực hiện use-case

- Nếu use-case thành công người thì người dùng có thể xem được bảng danh sách điện thoại bàn của công ty. Ngược lại trạng thái của hệ thống sẽ không đổi.

❖ Phần mở rộng

- Không có.

2.8 Use-case “Tìm kiếm nhân viên”

❖ Mô tả tóm tắt

- *Tất cả người dùng truy cập vào hệ thống đều có thể sử dụng use-case mà không cần đăng nhập.*
- *Dùng use-case để thuận tiện cho việc tìm kiếm nhân viên.*
- *Use-case cho phép người dùng tìm kiếm nhân theo tên nhân viên, mã nhân viên, chức danh.*

❖ Dòng sự kiện chính

- *Use-case bắt đầu khi có người dùng muốn tìm kiếm nhân viên.*
- *Người dùng tiến hành nhập thông tin vào ô tìm kiếm nhân viên có thể theo mã nhân viên, tên nhân viên, chức danh.*
- *Sau khi nhập xong thông tin, người dùng chọn nút tìm kiếm hệ thống sẽ hiển thị danh sách nhân viên tương ứng.*

❖ Dòng sự kiện khác

- *Nếu thông tin người dùng nhập không chính xác hoặc không tồn tại nhân viên với thông tin tìm kiếm, hệ thống sẽ đưa ra thông báo. Người dùng có thể nhập lại thông tin tìm kiếm hoặc kết thúc use-case.*

❖ Yêu cầu đặc biệt

- *Không có.*

❖ Trạng thái hệ thống trước khi bắt đầu use-case

- *Hệ thống đang ở trạng thái homepage.*

❖ Trạng thái hệ thống sau khi thực hiện use-case

- Nếu use-case thành công, hệ thống sẽ hiển thị danh sách nhân viên tương ứng với thông tin tìm kiếm. Ngược lại, hệ thống sẽ đưa ra thông báo và quay lại trạng thái ban đầu.

❖ Phần mở rộng

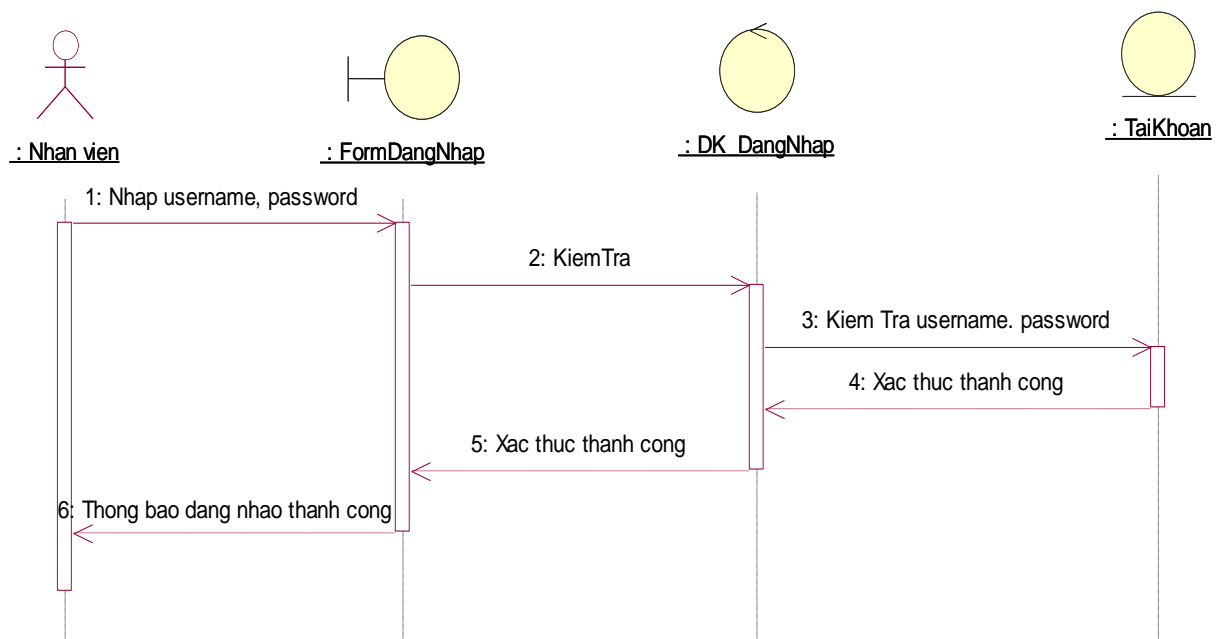
- Không có.

2.9 Xem sơ đồ công ty

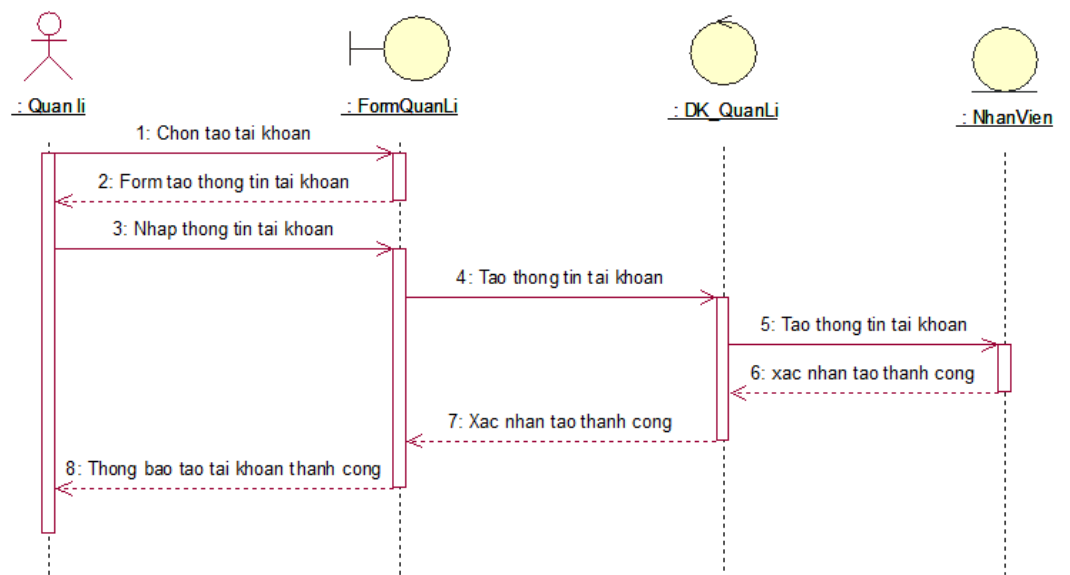


3. Biểu đồ tuần tự

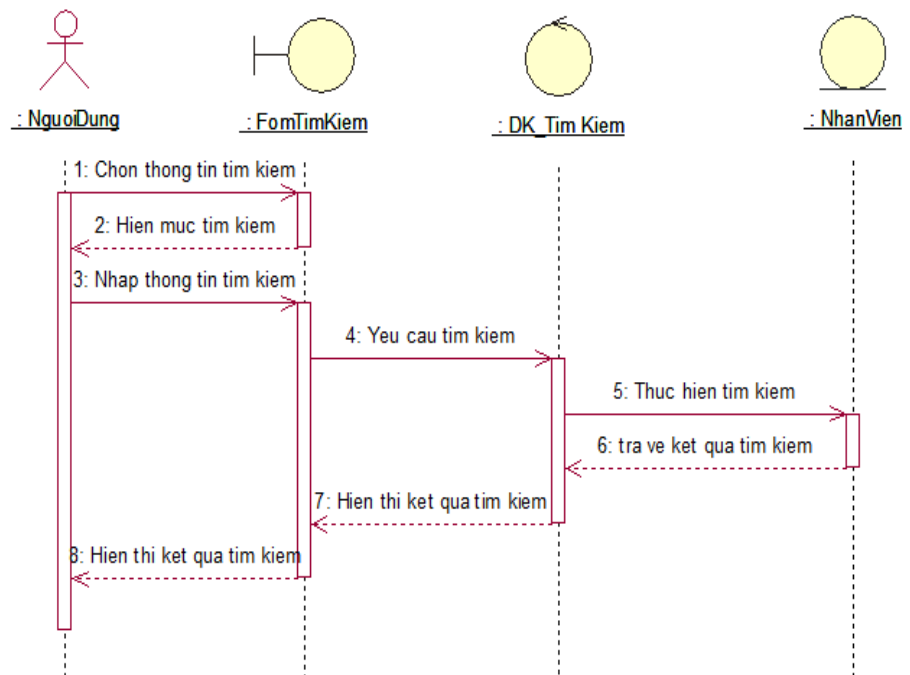
3.1 Ca sử dụng “Đăng nhập”



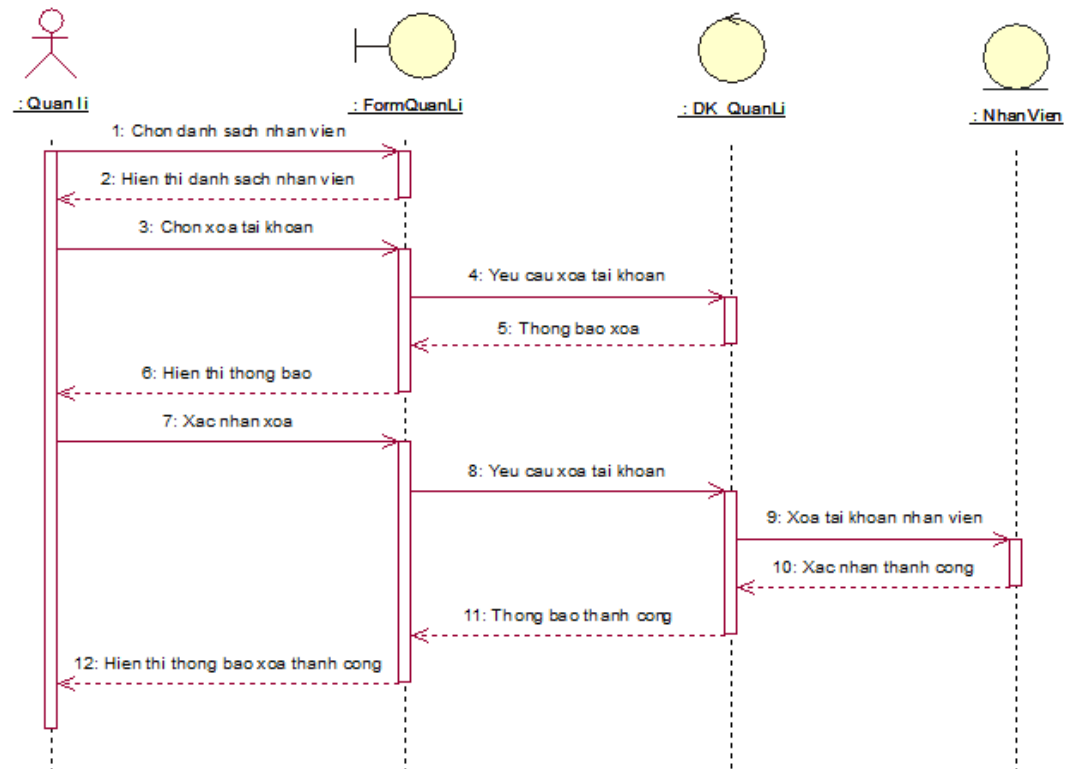
3.2 Ca sử dụng “Tạo tài khoản nhân viên”



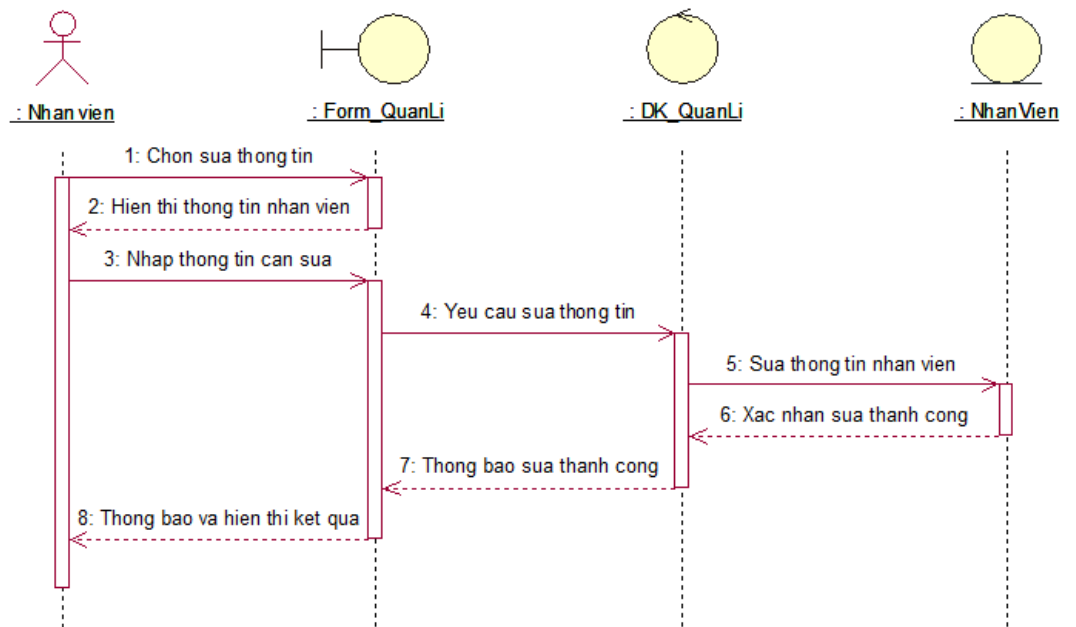
3.3 Ca sử dụng “Tìm kiếm nhân viên”



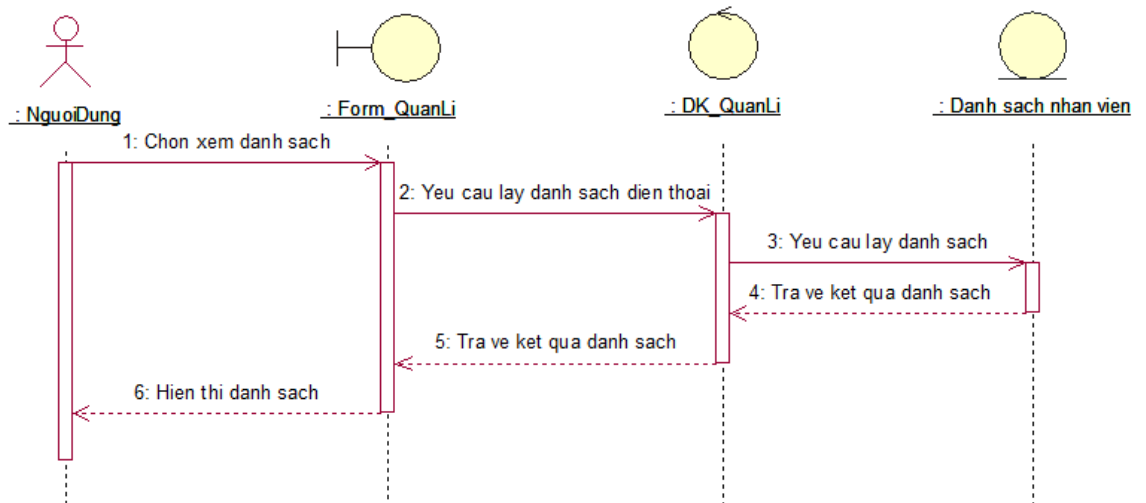
3.4 Ca sử dụng “ xóa hồ sơ nhân viên ”



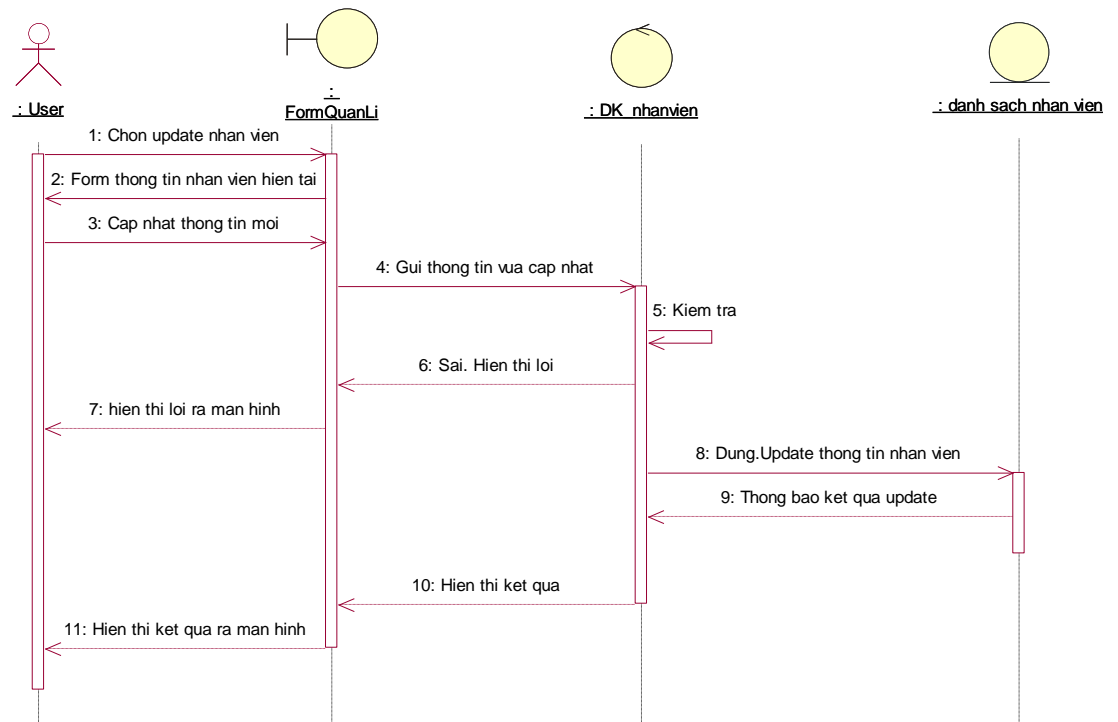
3.5 Ca sử dụng sửa “ Hồ sơ nhân viên ”



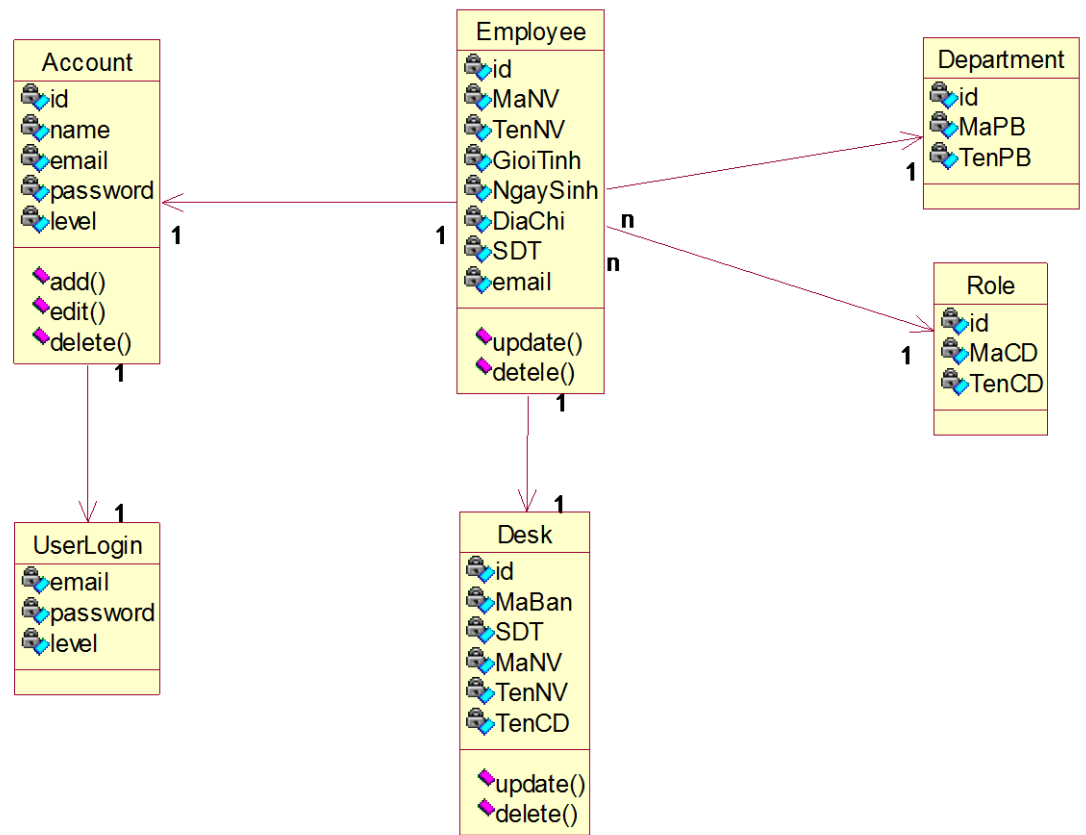
3.6 Ca sử dụng “ xem danh sách nhân viên ”



3.7 Ca sử dụng “ update thông tin ”



4. Biểu đồ lớp



Bảng 3: Danh sách lớp đối tượng và quan hệ

STT	Tên lớp / Quan hệ	Loại	Ý nghĩa/Ghi chú
1	Account	Public	Tài khoản đăng nhập của nhân viên
2	Employee	Public	Lưu thông tin của nhân viên
3	Department	Public	Lưu thông tin phòng ban
4	Role	Public	Lưu thông tin chức danh
5	Desk	Public	Lưu thông tin bàn làm việc
6	UserLogin	Public	Lưu thông tin người truy cập

Bảng 4: Mô tả chi tiết từng lớp đối tượng

STT	Phương thức	Tên thuộc tính	Loại	Ràng buộc class	Ý nghĩa/Ghi chú
1	Add, edit, delete	Name, email, password, level	Public	Employee, UserLogin	Cho biết thông tin tài khoản
2	Update, delete	MaNV, TenNV, GioiTinh, NgaySinh, DiaChi, SDT, email	Public	Account, Department, Role, Desk	Cho biết thông tin của nhân viên
3	Update, delete	MaBan, SDT_ban, MaNV, TenNV, TenCD	Public	Employee, Role	Cho biết thông tin của bàn làm việc

Bảng 5: Mô tả chi tiết các kiểu dữ liệu

Account			
STT	Thuộc tính	Kiểu	Ghi chú
1	Id	int	Thêm tự động
2	name	Varchar	Không được null
3	email	Varchar	Không được null
4	password	Varchar	Không được null
5	level	tinyint	Không được null

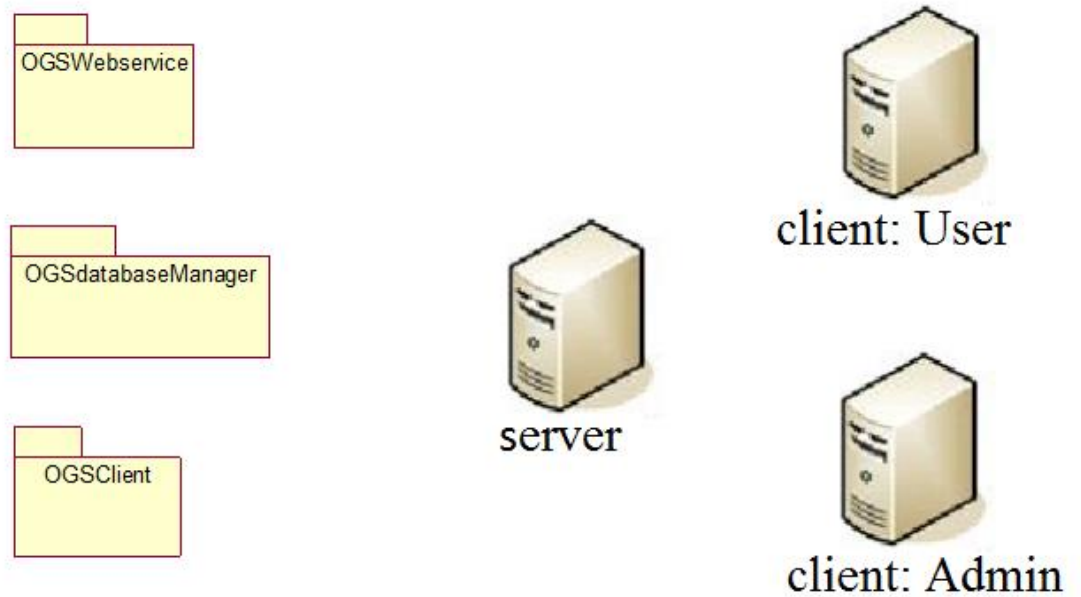
Employee			
STT	Thuộc tính	Kiểu	Ghi chú
1	id	int	Thêm tự động
2	MaNV	Varchar	Không được null
3	TenNV	Varchar	Không được null
4	GioiTinh	tinyint	Có thể null
5	NgaySinh	date	Có thể null
6	DiaChi	Varchar	Có thể null
7	SĐT	int	Có thể null
8	email	Varchar	Có thể null

Role			
STT	Thuộc tính	Kiểu	Ghi chú
1	id	int	Thêm tự động
2	MaCD	Char	Không được null
3	TênCD	Varchar	Không được null

Department			
STT	Thuộc tính	Kiểu	Ghi chú
1	id	int	Thêm tự động
2	MaPB	Char	Không được null
3	TênPB	Varchar	Không được null

Desk			
STT	Thuộc tính	Kiểu	Ghi chú
1	id	int	Thêm tự động
2	MaBan	Char	Không được null
3	SĐT_ban	int	Không được null
4	MaNV	Varchar	Không được null
5	TenVN	Varchar	Có thể null
6	TenCD	Varchar	Có thể null

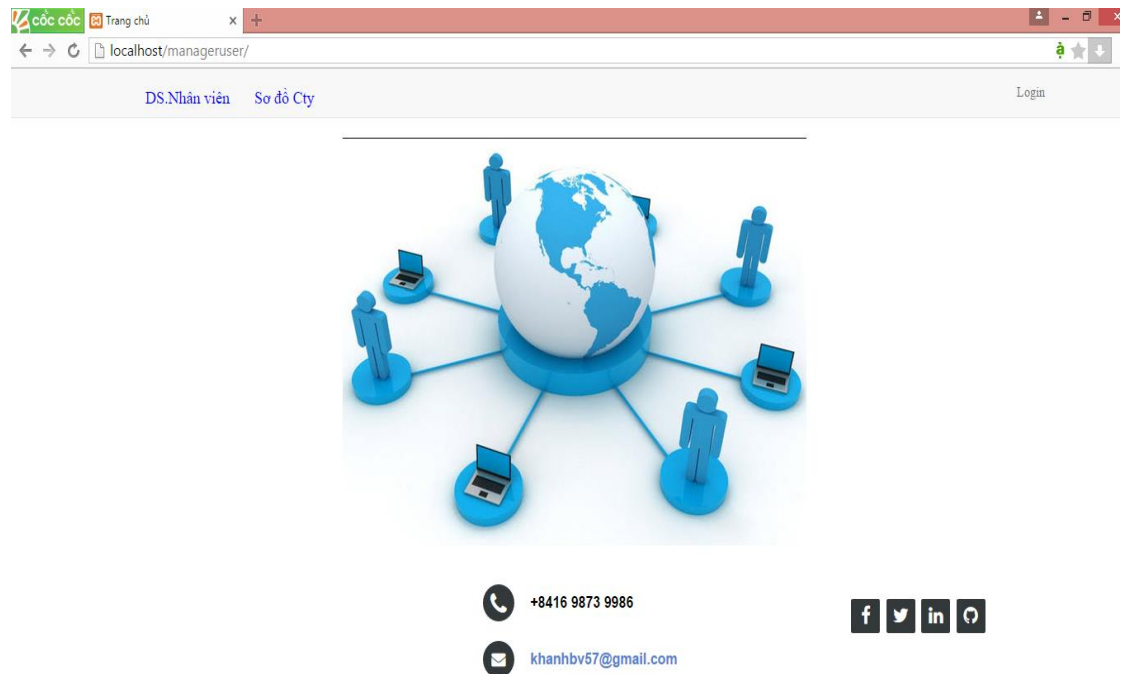
5. Kiến trúc hệ thống



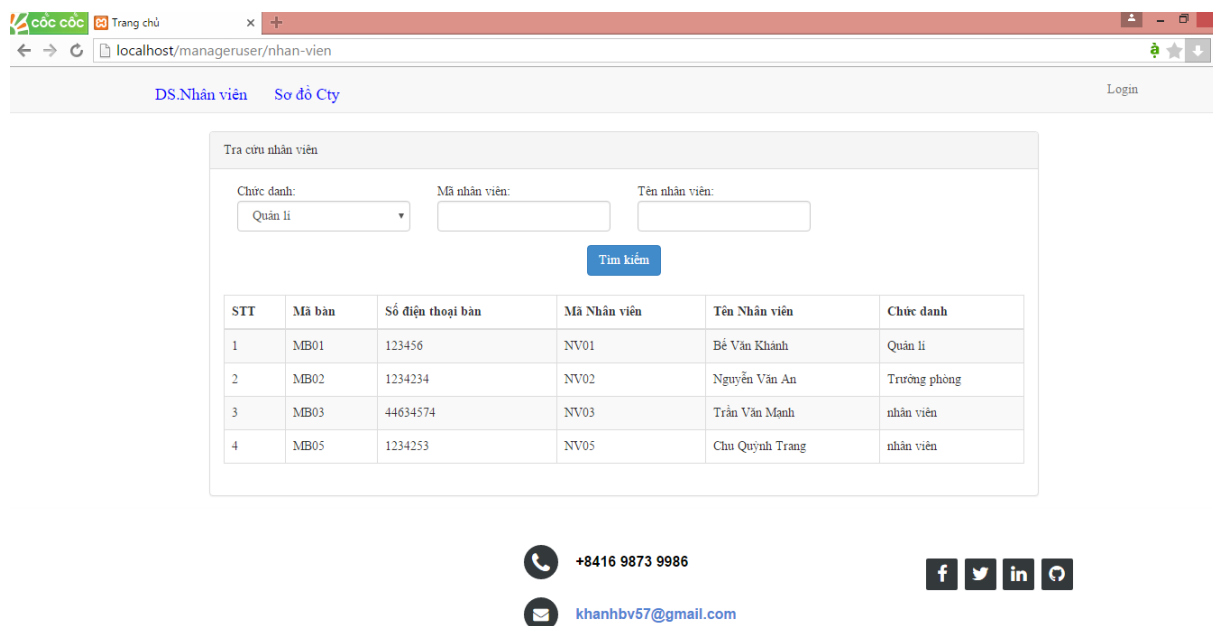
Thành phần	Ý nghĩa
OGSCient	Client
OGSWebService	Web Service
OGSDatabaseManager	Quản lí database
Client: User	Người sử dụng hệ thống
Client: Admin	Người quản lí hệ thống

III. Thiết kế giao diện

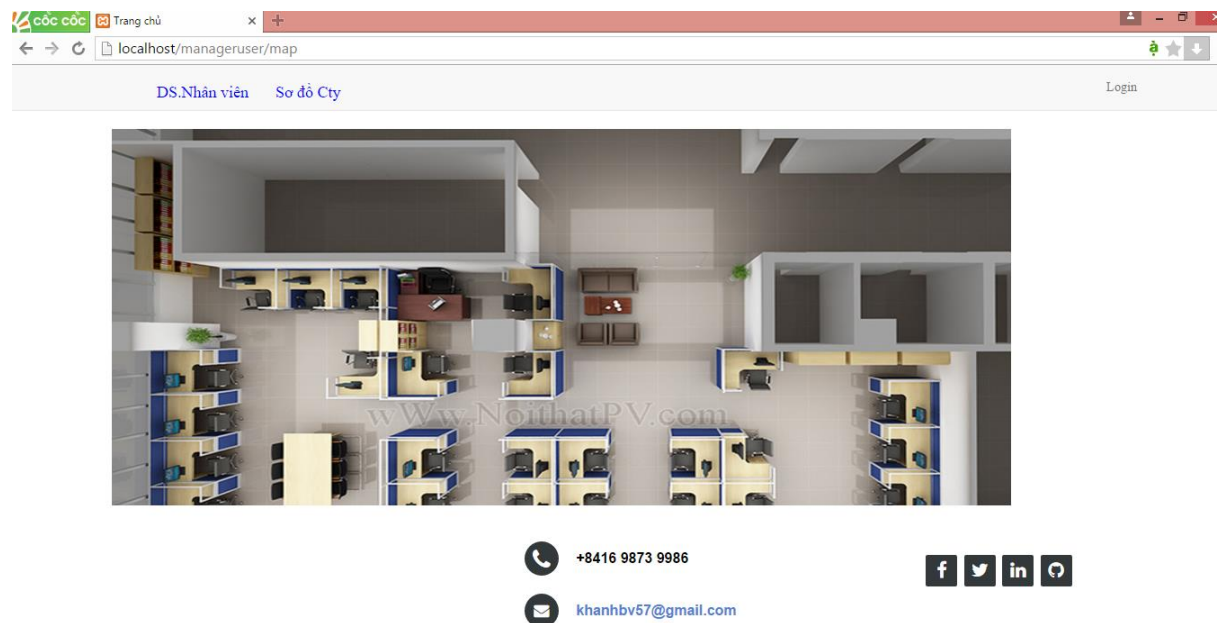
5.1 Giao diện màn hình chính



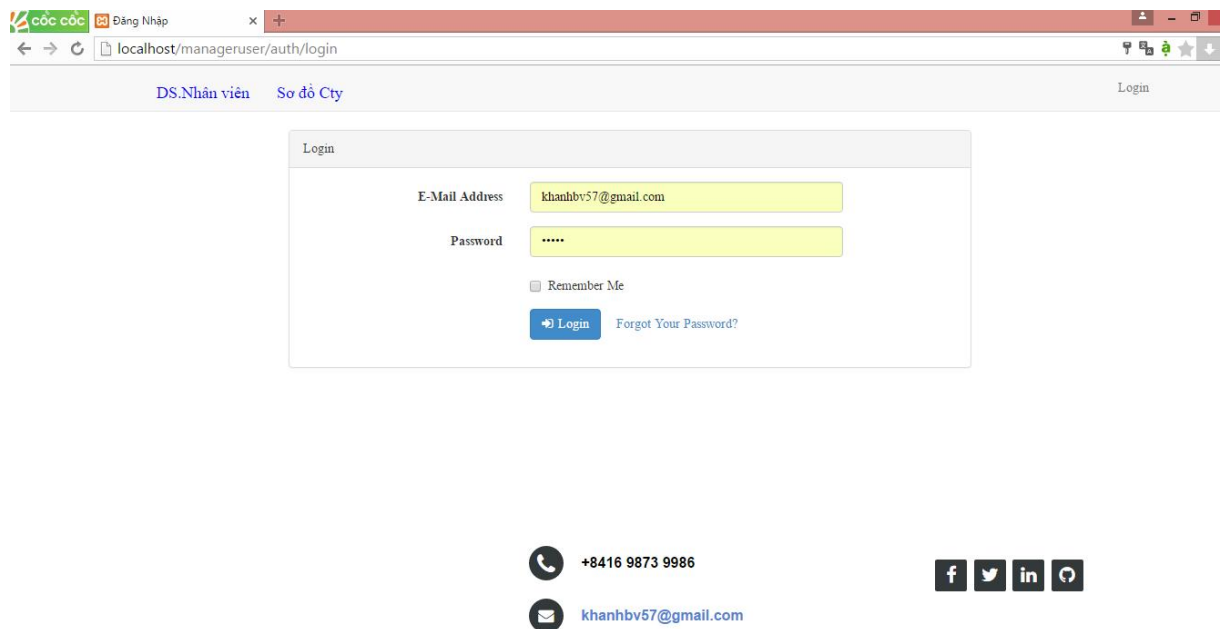
5.2 Giao diện tìm kiếm ngoài màn hình chính



5.3 Xem sơ đồ công ty



5.4 Màn hình đăng nhập



5.5 Trang quản lí tài khoản

Quản lí nhân sự

Search...

Dashboard

Tài khoản

Danh sách tài khoản

Thêm tài khoản

Nhân viên

Bản làm việc

Danh sách tài khoản

Show 10 entries Search:

ID	Họ Tên	Email	Level	Delete	Edit
1	Bê Văn Khánh	khanhbv57@gmail.com	SuperAdmin	Delete	Edit
2	Nguyễn Văn An	nguyenvanan@gmail.com	Admin	Delete	Edit
3	Trần Văn Mạnh	tranvanmanh@gmail.com	Member	Delete	Edit
5	Chu Quỳnh Trang	chuquynhtrang@gmail.com	Member	Delete	Edit
7	a	aa@gmail.com	Admin	Delete	Edit

Showing 1 to 5 of 5 entries

Previous 1 Next

5.6 Sửa tài khoản

Quản lí nhân sự

Search...

Dashboard

Tài khoản

Nhân viên

Bản làm việc

Sửa thông tin

Username

Bê Văn Khánh

Password

Please Enter Password

RePassword

Please Enter RePassword

Email

khanhbv57@gmail.com

User Level ☒ Admin ☐ Member

User Edit Reset

5.7 Trang thêm tài khoản

Quản lý nhân sự

Search...

Dashboard

Tài khoản

Danh sách tài khoản

Thêm tài khoản

Nhân viên

Bàn làm việc

Thêm tài khoản

Username

Please Enter Username

Password

Please Enter Password

RePassword

Please Enter RePassword

Email

Please Enter Email

User Level ☒ Admin ☐ Member

User Add Reset

5.8 Danh sách bàn làm việc

Quản lý nhân sự

Search...

Dashboard

Tài khoản

Nhân viên

Bàn làm việc

Danh sách bàn

Thêm bàn

Danh sách bàn làm việc

Show 10 entries Search:

STT	Mã bàn	SDT Bàn	Mã nhân viên	Tên Nhân viên	Chức Danh
1	MB01	0123456	NV01	Bê Văn Khánh	Quản lý
2	MB02	01234234	NV02	Nguyễn Văn An	Trưởng phòng
3	MB03	044634574	NV03	Trần Văn Mạnh	nhân viên
4	MB05	01234253	NV05	Chu Quỳnh Trang	nhân viên

Showing 1 to 4 of 4 entries

Previous 1 Next

5.9 Thêm bàn làm việc

The screenshot shows a web application interface for 'Quản lý nhân sự' (HR Management). The left sidebar contains a search bar and a menu with items: Dashboard, Tài khoản, Nhân viên, Bàn làm việc (expanded), Danh sách bàn, and Thêm bàn. The main content area is titled 'Thêm bàn làm việc' and contains a form with the following fields: Table_code, Table_phone, User_id, Role_id, and Room_id. Each field has a placeholder text 'Please Enter [field name]'. At the bottom of the form are 'Add' and 'Reset' buttons.

5.10 Giao diện profile

The screenshot shows a web application interface for 'Quản lý nhân sự' (HR Management). The left sidebar contains a search bar and a menu with items: Dashboard, Tài khoản, Nhân viên (expanded), Danh sách nhân viên, Thêm nhân viên, and Bàn làm việc. The main content area is titled 'Thêm nhân viên' and contains a form for adding a new employee. The form includes a profile picture placeholder, a 'Họ tên:' field, a 'Mã nhân viên:' field, a 'User_ID:' field, a 'Role_id:' field, a 'Phòng ban:' field, a 'Ngày sinh:' field with a date picker, an 'Email:' field, a 'Giới tính:' field with radio buttons for 'Nam' and 'Nữ', and a 'Điện thoại:' field. At the bottom of the form are 'Add' and 'Reset' buttons.

IV. Kết quả thực nghiệm

1. Môi trường phát triển:

- Hệ điều hành: Microsoft Windows
- Hệ quản trị cơ sở dữ liệu: MySQL
- Công cụ phân tích thiết kế: Rational Rose
- Framework: Laravel 5.2

2. Kết quả đạt được:

- Đã thiết kế được hết các chức năng đã đề ra.
- Giao diện đơn giản, dễ nhìn, dễ dàng thao tác.

3. Hướng phát triển:

- Tiếp tục nghiên cứu để phát triển ứng dụng hoàn thiện hơn, sửa chữa sai sót các sai sót.
- Tiếp tục nghiên cứu để làm cho ứng dụng thêm nhiều tính năng hơn.

V. Tài liệu tham khảo

- Tài liệu Laravel tham khảo tại: <https://laravel.com/> kungfuphp.com
- Tài liệu IBM Rational Rose:
- Tài liệu phân tích thiết kế hướng đối tượng: *Phân tích thiết kế hệ thống quản lý nhân sự - Luận văn, đồ án, đề tài tốt nghiệp*
<http://thuvienluanvan.info/>