**ĐẠI HỌC QUỐC GIA HÀ NỘI**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**



**Nguyễn Hữu Kết**

**PHÁT TRIỂN PHÂN HỆ MỘT CỬA**

**THUỘC PHẦN MỀM DỊCH VỤ CÔNG**

**THEO MÔ HÌNH SAAS**

**KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY**

**Ngành: Công nghệ Thông tin**

**HÀ NỘI – 2014**

**ĐẠI HỌC QUỐC GIA HÀ NỘI**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**

**Nguyễn Hữu Kết**

**PHÁT TRIỂN PHÂN HỆ MỘT CỬA**

**THUỘC PHẦN MỀM DỊCH VỤ CÔNG**

**THEO MÔ HÌNH SAAS**

**KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY**

**Ngành: Công nghệ Thông tin**

**Cán bộ hướng dẫn: ThS. Lê Đình Thanh**

**Cán bộ đồng hướng dẫn: ThS. Nguyễn Việt Tân**

**HÀ NỘI – 2014**

**VIETNAM NATIONAL UNIVERSITY, HANOI**

**UNIVERSITY OF ENGINEERING AND TECHNOLOGY**

**Nguyen Huu Ket**

**DEVELOPMENT OF UNIGATE MODULE FOR THE SAAS-BASED PUBLIC SERVICE SYSTEM**

**Major: Information Technology**

**Supervisor: MSc. Le Dinh Thanh**

**Co**-**Supervisor: MSc. Nguyen Viet Tan**

**HA NOI - 2014**

**TÓM TẮT**

**Tóm tắt:** Khoá luận tốt nghiệp

***Từ khoá:*** SaaS

**ABSTRACT**

**Abstract:** Khoá luận tốt nghiệp

***Keywords:*** SaaS

**LỜI CAM ĐOAN**

Tên tôi là Nguyễn Hữu Kết

**MỤC LỤC**

[Chương 1. GIỚI THIỆU 10](#_Toc387585625)

[1.1. Giải quyết dịch vụ công theo cơ chế một cửa 10](#_Toc387585626)

[1.2. Phân hệ một cửa 11](#_Toc387585627)

[1.3. Mô hình phần mềm dịch vụ 12](#_Toc387585628)

[1.4. Nội dung đề tài 13](#_Toc387585629)

[1.5. Kết quả đề tài 13](#_Toc387585630)

[1.6. Bố cục của đề tài 13](#_Toc387585631)

[Chương 2. PHẦN MỀM DỊCH VỤ 15](#_Toc387585632)

[2.1. Khái niệm 15](#_Toc387585633)

[2.2. Mô hình 15](#_Toc387585634)

[2.3. Ưu, nhược điểm 17](#_Toc387585635)

[2.3.1. Ưu điểm 17](#_Toc387585636)

[2.3.2. Nhược điểm 18](#_Toc387585637)

[2.4. Một số công ty cung cấp phần mềm dịch vụ 19](#_Toc387585638)

[Chương 3. PHÂN TÍCH THIẾT KẾ HỆ THỐNG 20](#_Toc387585639)

[3.1.1. Mô hình nghiệp vụ 20](#_Toc387585640)

[3.1.2. Chức năng 23](#_Toc387585641)

[3.1.3. Tác nhân 27](#_Toc387585642)

[3.1.4. Ca sử dụng 28](#_Toc387585643)

[Chương 4. CÀI ĐẶT 33](#_Toc387585644)

[4.1. Khung phát triển ứng dụng Symfony2 33](#_Toc387585645)

[4.1.1. Giới thiệu 33](#_Toc387585646)

[4.1.2. Kiến trúc và hoạt động 35](#_Toc387585647)

[4.2. Cơ sở dữ liệu 38](#_Toc387585648)

[4.3. Mức đô hoàn thành 38](#_Toc387585649)

[4.4. Giới thiệu sử dụng 39](#_Toc387585650)

[4.4.1. Quản lý bên thuê 39](#_Toc387585651)

[4.4.2. Đăng ký đơn vị sử dụng hệ thống. 39](#_Toc387585652)

[4.4.3. Đăng nhập vào hệ thống. 40](#_Toc387585653)

[4.4.4. Niêm yết thủ tục hành chính 40](#_Toc387585654)

[4.4.5. Tiếp nhận hồ sơ 44](#_Toc387585655)

[Đến đây quá trình nộp hồ sơ đã hoàn tất. 46](#_Toc387585656)

[4.4.6. Chuyển thụ lý hồ sơ 46](#_Toc387585657)

[Chương 5. KẾT LUẬN 48](#_Toc387585658)

[5.1. Những việc đã làm được 48](#_Toc387585659)

[5.1.1. Tìm hiểu bài toán dịch vụ công 48](#_Toc387585660)

[5.1.2. Phân tích và thiết kế phân hệ một cửa 48](#_Toc387585661)

[5.1.3. Cài đặt phân hệ một cửa 48](#_Toc387585662)

[5.1.4. Tìm hiểu và sử dụng khung phát triển ứng dụng web Symfony2 48](#_Toc387585663)

[5.2. Hướng phát triển 48](#_Toc387585664)

[5.2.1. Hướng phát triển ngắn hạn 48](#_Toc387585665)

[5.2.2. Hướng phát triển dài hạn 48](#_Toc387585666)

**BẢNG CÁC KÍ HIỆU, CHỮ VIẾT TẮT**

# GIỚI THIỆU

## Giải quyết dịch vụ công theo cơ chế một cửa

Mặc dù có nhiều cách tiếp cận khái niệm, thuật ngữ dịch vụ công dưới các góc độ khác nhau, nhưng về cơ bản đều thống nhất tương đối ở các đặc điểm sau của dịch vụ công:

* Là một loại dịch vụ do Nhà nước (cơ quan hành chính, đơn vị sự nghiệp của Nhà nước) trực tiếp thực hiện hoặc uỷ quyền cho các tổ chức, đơn vị ngoài Nhà nước thực hiện dưới sự giám sát của Nhà nước;
* Nhằm đáp ứng nhu cầu xã hội, nhân dân (những nhu cầu tối thiểu, thiết yếu);
* Nhà nước là người chịu trách nhiệm đến cùng trước nhân dân, xã hội về chất lượng dịch vụ cũng như số lượng dịch vụ. Trách nhiệm ở đây thể hiện qua việc hoạch định chính sách, thể chế pháp luật, quy định tiêu chuẩn chất lượng, thanh tra kiểm tra giám sát việc thực hiện;
* Không nhằm mục tiêu lợi nhuận;
* Đối tượng thụ hưởng Dịch vụ công không trực tiếp trả tiền (đã trả qua hình thức thuế), tuy nhiên có những trường hợp phải trả lệ phí theo quy định chặt chẽ của pháp luật.

Vậy có thể hiểu một cách khái quát Dịch vụ công là những dịch vụ do Nhà nước chịu trách nhiệm, phục vụ các nhu cầu cơ bản, thiết yếu chung của người dân không vì mục đích lợi nhuận.

Xuất phát từ cơ sở nhận thức như trên, căn cứ vào điều kiện thực tế của Việt Nam, chúng ta có thể tạm chia dịch vụ công ở nước ta hiện nay thành các loại sau:

* Thứ nhất, những dịch vụ sự nghiệp công (có người gọi là hoạt động sự nghiệp công), phục vụ những nhu cầu thiết yếu cho xã hội, quyền và lợi ích công dân. Nhà nước trực tiếp (thông qua) các tổ chức, đơn vị sự nghiệp của Nhà nước hoặc uỷ quyền cho các tổ chức ngoài Nhà nước thực hiện, cụ thể như chăm sóc sức khoẻ, giáo dục, đào tạo, văn hoá, thể dục thể thao, khoa học, bảo hiểm an sinh xã hội, phòng cháy chữa cháy, bão lụt, thiên tai, dịch vụ tư vấn, hỗ trợ pháp lý cho người nghèo ....
* Thứ hai, những hoạt động mang tính dịch vụ công ích, đây là các hoạt động có một phần mang tính chất kinh tế, hàng hoá như cung cấp điện, cấp nước sạch, giao thông công cộng đô thị, viễn thông, vệ sinh môi trường, xây dựng kết cấu hạ tầng, vận tải công cộng, khuyến nông, khuyến ngư, khuyến diêm ...
* Bên cạnh đó, hiện nay còn có luồng ý kiến cho rằng có loại thứ ba của dịch vụ công, đó là dịch vụ hành chính công. Loại này liên quan đến hoạt động thực thi pháp luật của các cơ quan nhà nước, cụ thể như các hoạt động thẩm định hồ sơ, ký phê duyệt, tổ chức cho đăng ký, cấp giấy chứng nhận đăng ký kinh doanh, cấp giấy chứng nhận quyền sử dụng đất, cấp phép khai thác khoáng sản, trật tự an toàn xã hội, hải quan, chứng thực ....
* Cải cách thủ tục hành chính theo cơ chế "một cửa" chính là giảm bỏ các thủ tục phiền hà, không cần thiết khi thực hiện.

**Mục đích: của cơ chế một cửa**

* Tạo thuận lợi và giải quyết công việc nhanh chóng theo yêu cầu của khách hàng.
* Nâng cao trách nhiệm của công chức khi thực hiện công vụ.
* Nâng cao vai trò phối kết hợp của các phòng môn, giải quyết công việc có hiệu quả chất lượng.
* Tạo điều kiện cho công dân, tổ chức giám sát việc thực hiện công khai các thủ tục hành chính của Sở.
* Giúp lãnh đạo Sở nắm được các hoạt dộng cụ thể một cách có hệ thống để kiểm tra, đôn đốc và xử lý đúng các nội dung theo yêu cầu của khách hàng.

## Phân hệ một cửa

Hệ thống dịch vụ công gồm 3 phân hệ:

* Phân hệ hỗ trợ công dân hỗ trợ người dân đăng ký làm thủ tục hành chính,xem và cập nhật thông tin về thủ tục hành chính đã nộp.
* Phân hệ hỗ trợ lãnh đạo có nhiệm vụ trợ giúp lãnh đạo – những người có thẩm quyền về lĩnh vực của thủ tục hành chính đưa ra những quyết định nhanh chóng,giảm thời gian chờ đợi của người dân.
* Phân hệ một cửa có chức năng Tiếp nhận hồ sơ TTHC và trả kết quả TTHC, thông báo trạng thái giải quyết hồ sơ TTHC tới công dân/tổ chức đƣợc giải quyết. Kết nối với phân hệ điều hành tác nghiệp để thụ lý hồ sơ TTHC.

## Mô hình phần mềm dịch vụ

Khái niệm về phần mềm dịch vụ (Software as a Service - SaaS) hình thành đúng vào thời điểm khi các giám đốc CNTT đang càng ngày ngán ngẩm với những chi phí chóng mặt của các gói sản phẩm doanh nghiệp. Họ không chỉ phải tiêu tốn hàng ngàn đô chỉ để nhận được cái giấy phép sử dụng phần mềm đóng gói đó mà còn phải chi đến hàng chục ngàn thậm chí có thể tới hàng trăm ngàn đô để triển khai phần mềm đó. Chưa tính đến các chi phí khác như phí tư vấn và phí đào tạo. Ngoài ra việc bổ sung thêm cơ sở hạ tầng cần thiết để chạy phần mềm cũng làm tăng thêm các khoản phí bảo trì.

|  |
| --- |
| Vì vậy, SaaS nổi lên từ đống đổ nát đầy chắp vá của các dự án triển khai CRM và ERP trị giá hàng nhiều triệu đô la như một sự thay đổi toàn diện đối với mô hình cấp giấy phép sử dụng phần mềm truyền thống. Phần mềm dịch vụ SaaS hứa hẹn những ưu điểm như triển khai sớm hơn, tốc độ hơn, và rẻ hơn. Đề xuất có giá trị ấy rất khó làm ngơ nhưng cũng đầy rủi ro. Đến ngày hôm nay, SaaS vẫn là một sự lựa chọn cần phải cân nhắc kỹ đối với nhiều doanh.  **Sự khác biệt giữa SaaS và mô hình cung cấp phần mềm truyền thống**  Mô hình dịch cung cấp dịch vụ truyền thống – phần mềm đóng gói là mô hình bán đứt đoạn bản quyền phần mềm, điều này có nghĩa là người mua sẽ sở hữu bản quyền phần mềm từ nhà cung cấp. Hàng năm người mua phải trả thêm phí bảo trì và hỗ trợ (chiếm 15%-20% phí bản quyền). Ngược lại, với mô hình phần mềm dịch vụ SaaS, người dùng sẽ trả các khoản thuê bao theo tháng hoặc theo năm. Nói cách khác các khoản phí bản quyền, triển khai và duy trì hỗ trợ sẽ được thanh toán trải rộng ra theo thời gian. |

**Sự khác biệt giữa SaaS và mô hình nhà cung cấp dịch vụ phần mềm (ASP)**

Phần mềm dịch vụ SaaS phát triển từ mô hình nhà cung cấp dịch vụ phần mềm (ASP). Khi mô hình nhà cung cấp dịch vụ phần mềm ASP phát triển rầm rộ trong thập niên 90, họ đã đưa ra những tính năng tương tự như các nhà cung cấp sản phẩm phần mềm dịch vụ SaaS đưa ra hiện nay: sản phẩm cho thuê được cung cấp qua Internet. Vấn đề mà các mô hình nhà cung cấp dịch vụ phần mềm ASP gặp phải đó chính là việc họ cố làm mọi thứ để đáp ứng nhu cầu của tất cả người dùng và kết quả họ gặp phải sức ép về cơ sở hạ tầng của chính họ. Trong nỗ lực đáp ứng nhu cầu cụ thể của từng khách hàng, các nhà cung cấp sản phẩm ASP  đã mất nhiều chi phí cần thiết để cung cấp các dịch vụ theo cách thức tiết kiệm chi phí cho khách hàng.

Các nhà cung cấp phần mềm dịch vụ SaaS thành công hiện nay như Salesforce.com, LeanLogistics và Ketera đã giải quyết triệt để được các vấn đề về khả năng mở rộng và độ tin cậy của hệ thống phần mềm, việc làm này đã khiến cho sản phẩm SaaS bám sát và cuối cùng thì đánh bại được ASP. Thay vì việc cố gắng đáp ứng tất cả các yêu cầu của tất cả người dùng thì họ đưa ra các giải pháp “một cho tất cả”. Chính là việc tất cả các khách hàng của một nhà cung cấp sản phẩm phần mềm dịch vụ SaaS sẽ dùng chung một phần mềm. Code cơ bản dùng cho tất cả khách hàng là giống nhau và không thể tuỳ chỉnh. Bất cứ tính năng hay chức năng nào mà các nhà cung cấp SaaS thêm vào phần mềm này đều dựa trên những phản hồi của khách hàng nhằm cung cấp một phần mềm thích hợp nhất phục vụ cho số đông. Cách tiếp cận dựa trên nền tảng đa chiếm hữu (Multi-tency) khiến cho phần mềm dịch vụ SaaS khác biệt với mô hình cung cấp dịch vụ phần mềm ASP ban đầu và các nhà cung cấp sản phẩm cho thuê và sản phẩm theo yêu cầu khác đồng thời giúp cho các nhà cung cấp sản phẩm phần mềm dịch vụ có thể cung cấp cho khách hàng những sản phẩm thích hợp hơn với đơn giá thấp hơn, nhờ đó khách hàng có thể tiết kiệm chi phí hiệu quả và luôn đảm bảo chắc chắn sẽ nâng cấp sản phẩm của khách hàng lên những phiên bản mới nhất.

## Nội dung đề tài

Đề tài đưa ra một hướng giải quyết mới cho bài toán dịch vụ công là xây dựng một ứng dụng web cho phép người dân cũng như cán bộ thuộc phận một cửa trong mô hình dịch vụ công có thể giao tiếp dễ dàng hơn, từ đó giảm thời gian chờ đợi, giảm công sức đi lại, trao đổi thông tin giữa hai bên.

Đặc biệt hơn, ứng dụng web này sẽ được xây dựng trên mô hình phần mềm dịch vụ từ đó không chỉ đáp ứng nhu cầu của một cơ quan trên địa bàn nhất nhất định mà có phục vụ nhiều cơ quan tổ chức khác nhau.

## Kết quả đề tài

Tìm hiểu bài toán dịch vụ công

Phân tích và thiết kế phân hệ một cửa

Cài đặt phân hệ một cửa

Tìm hiểu và sử dụng khung phát triển ứng dụng web có tên symfony

## Bố cục của đề tài

Bố cục đề tài gồm 4 phần:

Phần 1. Giới thiệu đề tài.

Phần 2. Phân tích thiết kế hệ thống.

Phần 3. Phát triển khung ứng dụng SYMFONY2.

Phần 4. Cài đặt.

Phần 5. Kết luận.

# PHẦN MỀM DỊCH VỤ

## Khái niệm

Phần mềm dịch vụ là một mô hình cung cấp phần mềm tới khách hàng của các doanh nghiệp phát triển phần mềm [23]. Không giống như phần mềm đóng gói truyền thống, người sử dụng thường phải cài đặt vào hệ thống máy tính hoặc các máy chủ của họ, nhà cung cấp phần mềm dịch vụ SaaS làm chủ sở hữu phần mềm này và chạy phần mềm đó trên hệ thống máy tính ở trên trung tâm cơ sở dữ liệu. Khách hàng không sở hữu phần mềm này nhưng họ có thể thuê nó để tiết kiệm chi phí. Thường thì khách hàng sẽ trả phí thuê theo tháng. SaaS thỉnh thoảng vẫn được biết đến như một phần mềm cho thuê hay các sản phẩm phầm mềm theo yêu cầu “On-demand”.

Theo một định nghĩa khác, khái niệm phần mềm SaaS rất đơn giản là thay vì phải cấp phép sử dụng vĩnh viễn cho một phần mềm thì giờ đây các nhà cung cấp phần mềm cho phép khách hàng sử dụng phần mềm theo cách đóng phí định kì. Tất cả những vấn đề khác như bảo mật, nâng cấp tính năng đều do phía nhà cung cấp phần mềm SaaS thực hiện, doanh nghiệp sẽ được hưởng lợi từ những nâng cấp này mà không phải trả thêm một khoản phí nào.

## Mô hình

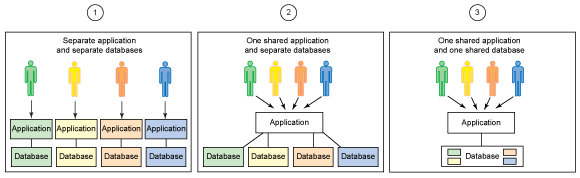
Các nhà cung cấp SaaS thường phân chia hệ thống thành nhiều ứng dụng nhỏ hơn. Mỗi ứng dụng này sẽ được cho thuê theo yêu cầu của doanh nghiệp. Như vậy, nhiều người dùng có thể thuê cùng lúc một ứng dụng. Đây là kiến trúc “một ứng dụng – nhiều thuê bao” (single-instance multi-tenant architecture) của hệ thống phần mềm dịch vụ. Theo quan điểm của các nhà thiết kế, ba đặc trưng để nhận biết một dịch vụ SaaS “đáng tin cậy” là:

1. Phạm vi năng lực của hệ thống

2. Khả năng đảm bảo hoạt động cho nhiều thuê bao

3. Khả năng cấu hình ứng dụng

Ba đặc trưng trên đóng vai trò quan trọng trong một kiến trúc SaaS. Tuy nhiên, tùy theo nhu cầu cụ thể của mình, doanh nghiệp có thể lựa chọn ba mức khai thác dịch vụ khác nhau [56]. Mỗi mức cung cấp những đặc trưng của kiến trúc SaaS, với tính đầy đủ được sắp tăng dần từ thấp lên cao.



*Hình 2.1. Ba mức mô hình dịch vụ đang được triển khai hiện nay [56]*

**Mức 1** Tương tự như việc triển khai phần mềm truyền thống. Mỗi doanh nghiệp có một ứng dụng chạy trên server của nhà cung cấp dịch vụ tuân theo kiến trúc khách-chủ truyền thống và khác với những doanh nghiệp còn lại.

Đặc điểm của hệ thống ở mức này là toàn bộ database và mã nguồn được sao chép từ hệ thống của nhà cung cấp dịch vụ và vận hành một cách độc lập. Với đặc điểm này,mức 1 của phần mềm dịch vụ gây ra sự phức tạp trong quản lý cũng như vận hành và bảo trì. Thay vì chỉ vận hành và bảo trì một hệ thống duy nhất,nhà cung cấp dịch vụ cần bảo trì số lượng hệ thống tương ứng với số lượng khách hàng sử dụng dịch vụ. Đôi khi điều này là không thể với những nhà cung cấp dịch vụ lớn như Google hay Saleforce.

**Mức 2:** Nhà phân phối cung cấp dịch vụ của mình dưới dạng những bản sao. Nhiều doanh nghiệp dùng chung sản phẩm phần mềm có mã lệnh như nhau. Việc sử dụng một cơ sở mã nguồn duy nhất làm nền tảng cho toàn bộ các ứng dụng sẽ giúp cho nhà cung cấp có thể dễ dàng sửa đổi, cập nhật, nâng cấp toàn bộ các ứng dụng cùng một lúc mà không cần phải thiết lập hoặc thao tác trên từng chương trình cụ thể với từng doanh nghiệp.

Hệ thống ở mức 2 đã cải tiến hơn so với mức một ở việc sử dụng chung mã nguồn cho toàn bộ khach hàng sử dụng dịch vụ. Điều này đảm vào việc nâng cấp và bảo trì hệ thống sẽ đơn giản hơn. Tuy nhiên khi có một lỗi hệ thống xảy ra,nó sẽ ảnh hưởng tới toàn bộ khách hàng đang sử dung. Ngoài ra,với khối lượng database lớn cần cập nhật và bảo trì thường xuyên,mô hình này cũng không thực sự là sự lựa chọn hợp lý cho các doanh nghiệp lớn hiện nay.

**Mức 3:** Ở mức này, nhà cung cấp sử dụng một hệ thống duy nhất để phục vụ tất cả mọi doanh nghiệp. Các chính sách bảo mật và xác thực quyền sử dụng đảm bảo dữ liệu được cung cấp một cách chính xác và không có sự xung đột lẫn nhau. Đồng thời, hệ thống cũng duy trì tính trong suốt đối với người sử dụng theo nghĩa họ không nhận thức được mình đang chia sẻ tài nguyên với những người khác.

## Ưu, nhược điểm

Có nhiều lợi ích mà khách hàng thu được trong quá trình triển khai phần mềm dịch vụ SaaS khác hẳn với các phần mềm cài đặt tại chỗ. Đầu tiên, việc triển khai SaaS thường tốn ít thời gian hơn việc triển khai các phần mềm cài đặt tại chỗ đơn giản là do người dùng sẽ không phải cài đặt phần mềm trên từng máy tính của họ. Các nhà cung cấp SaaS muốn vươn tới mục tiêu  trinh phục được khách hàng trong vòng 3 tháng hoặc ít hơn, nhưng thực tế việc triển khai có thể mất khoảng từ 3 đến 6 tháng (đôi khi còn nhiều hơn, điều này còn phụ thuộc vào quy mô và tính phức tạp của việc triển khai). Vì SaaS triển khai nhanh hơn và dễ hơn các phần mềm truyền thống nên doanh nghiệp cũng có thể thu nhận được chỉ số đầu tư hiệu quả (ROI) nhanh hơn, tất nhiên cung cấp cho người dùng một phần mềm thích hợp hơn.

Việc nâng cấp cũng có xu hướng khá ít chắp vá (bởi vì doanh nghiệp không tuỳ chỉnh phần mềm này) và doanh nghiệp luôn luôn thao tác với phiên bản mới nhất của phần mềm này, không giống như các phần mềm đóng gói truyền thống. Các nhà cung cấp phần mềm dịch vụ SaaS chủ yếu có tham vọng đẩy việc nâng cấp và cập nhất ra khỏi nền tảng của khách hàng.

Nói tóm lại, phần mềm dịch vụ ít lỗi hơn. Việc doanh nghiệp có một phiên bản của phần mềm dịch vụ này sẽ giảm độ phức tạp mà có thể là nguyên nhân gây lỗi chủ đạo của phần mềm.

### Ưu điểm

#### Lợi ích khách hàng

##### **Chi phí khởi tạo thấp**

Chi phí khởi tạo thấp. Tùy thuộc và tính năng yêu cầu và số lượng người dùng mà có cách tính phí khác nhau. Đây chính là lợi điểm dễ thấy nhất của ứng dụng SaaS. Thay vì doanh nghiệp phải bỏ một núi tiền ra mua cả một hệ thống khổng lồ để chỉ sử dụng và tính năng của chúng thì giờ đây doanh nghiệp có thể tiết kiệm chi phí kiểu đó.

##### **Chi phí hỗ trợ kĩ thuật thấp**

Thay vì doanh nghiệp phải bỏ tiền để thuê nhân viên kĩ thuật duy trì hệ thống của mình thì bây giờ các nhà cung cấp ứng dụng SaaS đã làm thay doanh nghiệp tất cả. Hầu như doanh nghiệp không phải đụng chạm gì đến vấn đề kĩ thuật. Chính vì thế chi phí thuê nhân viên kĩ thuật đã giảm đáng kể.

##### **Nâng cấp chương trình mà không tốn chi phí**

Thông thường những nhà cung cấp ứng dụng SaaS luôn tiến hành mở rộng tính năng ứng dụng của họ. Vì thế, khách hàng cũng được hưởng lợi từ việc này mà không phải trả thêm một đồng chi phí nào cả.

##### **Không giới hạn truy cập**

Doanh nghiệp có thể truy cập ứng dụng SaaS bất cứ đâu, bất cứ nơi miễn là nơi đó có kết nói internet. Đây cũng là một lợi điểm không hề nhỏ của các ứng dụng SaaS. Chính khả năng này mang lại cho những ứng dụng SaaS có nhiều cơ hội trong tương lai.

#### Lợi ích doanh nghiệp

**Bản quyền**

Nhà cung cấp dịch vụ không phải lo vấn đề vi phạm bản quyền vì chỉ có một phần mềm duy nhất được cài đặt và quản lý từ xa, hoàn hacker không thể nào lấy cắp được. Nếu càng nhiều người sử dụng thì nhà cung cấp dịch vụ càng có thể kiếm được nhiều tiền không bằng cách thu phí thì cũng bằng cách thu tiền quảng cáo, …

### Nhược điểm

Tuy nhiên ngoài những lợi ích kể trên thì cũng có khá nhiều hạn chế khi triển khai SaaS làm đau đầu các nhà cung cấp dịch vụ, một số hạn chế có thể liệt kê như sau:

*Không thể đáp ứng nhu cầu của tất cả mọi người*: Để xây dựng được một ứng dụng có khả năng đáp ứng được hết yêu cầu của mọi người là rất khó, mỗi cá nhân, công ty có một yêu cầu khác nhau, không ai giống ai, nên việc thống nhất tất cả đòi hỏi nhà cung cấp dịch vụ phải phân tích rất kỹ càng các nghiệp vụ trước khi triển khai SaaS.

*Khó khăn cho người dùng khi huyển đổi từ hệ thống cũ sang hệ thống mới*: Nếu khách hàng đang sử dụng các phần mềm hiện có, với cơ sở dữ liệu khá lớn, dùng trong một thời gian khá lâu thì sẽ rất khó để họ có thể chuyển qua dùng SaaS, ngay cả đối với nhà cung cấp dịch vụ cũng sẽ gặp rất nhiều khó khăn trong việc tích hợp hệ thống cũ của khách hàng với dịch vụ mình cung cấp.

*Không cung cấp được một số hệ thống đặc thù*: Có một số ứng dụng gần như không thể chuyển qua SaaS, ví dụ như các ứng dụng Business Intelligence, với khối lương dữ liệu rất lớn, không thể truyền tải qua mạng internet được, với lại dữ liệu này cần phải bảo mật cao, nên rất khó để khách hàng đồng ý đưa hết dữ liệu của mình lên internet. Bảo mật cũng là vấn đề lớn trong SaaS, nếu nhà cung cấp dịch vụ không có chính sách bảo mật tốt thì khách hàng không thể tin tưởng để có thể giao dữ liệu của mình cho người khác. Hiện tại thì các công ty lớn có lợi thế về vấn đề này hơn do thương hiệu của họ nổi tiếng thế giới, được người dùng tin tưởng hơn. Ví dụ một số ông lớn như: Google, Amazon, Microsoft, …

## Một số công ty cung cấp phần mềm dịch vụ

Hình dưới tổng hợp các công ty cung cấp dịch vụ Cloud Computing. Trong đó phần lớn các công ty đều cung cấp Software as a Service. Chúng ta có thể kể tên một số công ty tiêu biểu cung cấp SaaS như: Google, Amazon, Mocrosoft, Salesforce, Blogger, Flickr, …



*Ảnh 2.1 Các công ty lớn cung cấp dịch vụ SaaS [223]*

# PHÂN TÍCH THIẾT KẾ HỆ THỐNG

### Mô hình nghiệp vụ

Quy trình tại bộ phận một cưa như sau. Chuyên viên bộ phận một cửa tiếp nhận và kiểm tra hồ sơ của công dân sau đó lưu vào hệ thống. Chuyên viên thụ lý định kỳ đến nhận hồ sơ và chuyển về đơn vị thụ lý để giải quyết. Trong quá trình thụ lý,trang thái của hồ sơ liên tục được cập nhật và lưu vào hệ thống. Sau khi hoàn tất quá trình thụ lý,hồ sơ được giao lại cho chuyên viên bộ phận một cửa và được trả về phía công dân.

Ngoài ra còn một số quy trình phụ hỗ trợ quy trình trên,hình 3.1 thể hiện mối quan hệ giữa ba quy trình và với các quy trình phụ.



*Hình 3.1 Quy trình nghiệp vụ tại bộ phận một cửa*

#### Niêm yết thông tin thủ tục hành chính

*Bảng 3.1 Quy trình nghiệp vụ niêm yết thông tin thủ tục hành chính*

|  |  |  |
| --- | --- | --- |
| **Trách nhiệm** | **Trình tự công việc** | **Nghiệp vụ điện tử** |
| Cán bộ được phân quyền | Khai báo dịch vụ công  1 | * Chọn mục Khai báo TTHC |
| Tạo mới lĩnh vực | * Nhập tên lĩnh vực, mô tả * Chọn đơn vị thụ lý |
| 2  Tạo mới dịch vụ công  Thông tin mô tả  Giấy tờ hồ sơ  Văn bản liên quan  Phân quyền | - Nhập **thông tin mô tả**: tên dịch vụ, mã dịch vụ, lựa chọn có cho phép hiển thị trên website hay không? Lựa chọn có cho phép đăng ký hồ sơ qua mạng hay không?  - Nhập **giấy tờ hồ sơ**: tên giấy tờ, chọn loại giấy tờ, đính kèm mẫu (nếu có)  - Nhập **văn bản liên quan**: tên văn bản, đính kèm văn bản  - **Phân quyền** thao tác với dịch vụ: quyền tiếp nhận dịch vụ, trả lời câu hỏi công dân, nhập câu hỏi thường gặp, quyền cập nhật thông tin dịch vụ (bao gồm chức năng: Sửa, Xóa) |
| 3  Cập nhật câu hỏi thường gặp (FAQ) | * Chọn mục Câu hỏi thường gặp * Thêm mới câu hỏi: chọn dịch vụ, nhập câu hỏi, câu trả lời |
| 4  Dịch vụ công được công bố trên website |  |

#### Tiếp nhận giải quyết hồ sơ

*Bảng 3.2 Quy trình nghiệp vụ tiếp nhận giải quyết hồ sơ*

|  |  |  |
| --- | --- | --- |
| **Trách nhiệm** | **Trình tự công việc** | **Nghiệp vụ điện tử** |
| Công dân, tổ chức | 1  Nộp hồ sơ | * Nhập thông tin vào tờ khai điện tử * Đính kèm hồ sơ điện tử (nếu có) * In giấy biên nhận |
| CBCC làm việc tại BP1C | Kiểm tra, tiếp nhận  Y/c bổ sung  Trả lại hồ sơ nếu không hợp lệ | - Kiểm tra, tiếp nhận |
| CBCC làm việc tại BP1C | 2  Tổng hợp hồ sơ, chuyển cho phòng chuyên môn thụ lý | -Tìm kiến, lọc ra những hồ sơ mới tiếp nhận  - Chuyển thụ lý  - In phiếu giao nhận thụ lý hồ sơ (nếu cần) |
| Phòng chuyên môn | 3  Thụ lý hồ sơ | * Xử lý công việc: * Cho ý kiến xử lý * Thông báo cho công dân (nếu cần) |
| Phòng chuyên môn | 4  Trình, phê duyệt kết quả giải quyết | - Cho ý kiến xử lý, duyệt  - Kết thúc hồ sơ công việc |
| Phòng chuyên môn | 5  Chuyển kết quả giải quyết cho BP1C | * Thông báo trên website cho công dân |
| CBCC làm việc tại BP1C | 6  Trả kết quả cho công dân, tổ chức. Lưu hồ sơ | * Tìm kiếm, lọc ra những hồ sơ đã chuyển thụ lý * Gửi trả |

#### Hỏi đáp thông tin dịch vụ

*Bảng 3.3 Quy trình nghiệp vụ hỏi đáp thông tin dịch vụ*

|  |  |  |
| --- | --- | --- |
| **Trách nhiệm** | **Trình tự công việc** | **Nghiệp vụ điện tử** |
| Công dân, tổ chức | 1  Tạo, gửi câu hỏi | * Trên website TTHC, vào mục Hỏi đáp. * Chọn Tạo, gửi câu hỏi * Nhập thông tin: chọn gửi tới đơn vị nào, chọn hỏi về dịch vụ nào, nhập họ tên, câu hỏi,… |
| Cán bộ được phân quyền trả lời công dân về dịch vụ này | 2  Trả lời công dân | * Chọn mục Trả lời công dân * Chọn xem câu hỏi * Nhập thông tin: câu trả lời, lựa chọn thuộc tính hiển thị trên website |
| 3  Câu trả lời được công bố trên website |  |

### Chức năng

*Bảng 3.4 Chức năng*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Use Case** | **Actor chính** | **Actor phụ** | **Mức độ BMT** | **Mô tả trường hợp sử dụng** | **Phân loại Use Case** |
| 1 | Đăng ký làm TTHC | Công dân  Chuyênviên BP1C |  | B | Đăng ký làm TTHC | Phức tạp |
|  |  |  |  |  | Công dân có thể xem và lấy mẫu giấy tờ làm thủ tục hành chính |  |
|  |  |  |  |  | Công dân có thể đăng ký làm thủ tục hành chính trực tuyến |  |
|  |  |  |  |  | Hệ thống phải sinh mã đăng ký và hiện cho công dân mỗi khi công dân đăng ký |  |
|  |  |  |  |  | Công dân có thể điền form có sẵn để làm hồ sơ |  |
|  |  |  |  |  | Công dân có thể nhận được hướng dẫn làm đăng ký |  |
|  |  |  |  |  | Công dân có thể tìm lại hồ sơ đã đăng ký |  |
|  |  |  |  |  | Công dân phải xác thực trước khi thay đổi nội dung đăng ký |  |
|  |  |  |  |  | Công dân có thể hủy đăng ký trước đó |  |
|  |  |  |  |  | Công dân có thể bổ xung hay thay thế giấy tờ đăng ký |  |
| 2 | Xem các thủ tục được nhận/trả hồ sơ | Trưởng BP1C |  | B | Xem các thủ tục được nhận/trả hồ sơ | Đơn giản |
|  |  |  |  |  | Trưởng BP1C có thể xem danh sách các thủ tục được nhận/trả hồ sơ tại phòng |  |
| 3 | Phân ekip nhận/trả hồ sơ | Trưởng BP1C |  | B | Phân ekip nhận/trả hồ sơ | Đơn giản |
|  |  |  |  |  | Trưởng BP1C có thể xem danh sách các chuyên viên thuộc bộ phận |  |
|  |  |  |  |  | Trưởng BP1C có thể phân ekip nhận/trả hồ sơ |  |
| 4 | Log-in | Trưởng BP1C, chuyên viên BP1C |  | B | Đăng nhập hệ thống | Đơn giản |
|  |  |  |  |  | Người dùng phải nhập tên sử dụng và mật khẩu để vào hệ thống |  |
| 5 | Thống kê hồ sơ nhận/trả | Trưởng BP1C |  | B | Thống kê hồ sơ nhận/trả | Đơn giản |
|  |  |  |  |  | Trưởng B1C có thể xem số liệu thống kê các hồ sơ được nhận/ trả tại bộ phận |  |
|  |  |  |  |  | Trưởng BP1C có thể tạo báo cáo về số lượng hồ sơ được nhận/trả tại bộ phận |  |
| 6 | Xem thông tin ekip | Chuyên viên BP1C |  | B | Xem thông tin ekip | Đơn giản |
|  |  |  |  |  | Chuyên viên BP1C có thể biết thông tin về ekip nhận/trả |  |
| 7 | Theo dõi hồ sơ đăng ký | Chuyên viên BP1C |  | B | Theo dõi hồ sơ đăng ký | Trung bình |
|  |  |  |  |  | Chuyên viên BP1C phải biết danh sách các hồ sơ đăng ký thuộc trách nhiệm nhận/trả của ekip |  |
|  |  |  |  |  | Chuyên viên BP1C có thể xem nội dung hồ sơ đăng ký thuộc trách nhiệm nhận/trả |  |
|  |  |  |  |  | Chuyên viên BP1C có thể xóa các hồ sơ đăng ký (không đúng nội dung) |  |
|  |  |  |  |  | Chuyên viên BP1C có thể chuyển hồ sơ đăng ký quá hạn sang thư mục lưu tạm |  |
| 8 | Hướng dẫn làm đăng ký | Chuyên viên BP1C |  | B | Hướng dẫn làm đăng ký | Đơn giản |
|  |  |  |  |  | Chuyên viên BP1C có thể soạn tin hướng dẫn đăng ký TTHC và gửi cho công dân đã đăng ký |  |
| 9 | Tiếp nhận hồ sơ | Chuyên viên BP1C  Công dân |  | B | Tiếp nhận hồ sơ | Đơn giản |
|  |  |  |  |  | Chuyên viên BP1C có thể chọn hồ sơ đăng ký khi tiếp nhận |  |
|  |  |  |  |  | Chuyên viên BP1C có thể thêm hồ sơ được tiếp nhận |  |
|  |  |  |  |  | Hệ thống phải in ra phiếu biên nhận và hẹn ngày trả (kiêm hóa đơn thanh toán lệ phí) mỗi khi hoàn tất tiếp nhận |  |
| 10 | Chuyển thụ lý hồ sơ | Chuyên viên BP1C |  | B | Chuyển thụ lý hồ sơ | Đơn giản |
|  |  |  |  |  | Chuyên viên BP1C có thể xem hồ sơ |  |
|  |  |  |  |  | Chuyên viên BP1C có thể chọn hồ sơ để chuyển thụ lý |  |
|  |  |  |  |  | Hệ thống phải in phiếu bàn giao khi chuyên viên BP1C chọn chuyển thụ lý |  |
| 11 | Cập nhật và xử lý trạng thái hồ sơ | Chuyên viên thụ lý,  Chuyên viên BP1C |  | B | Cập nhật và xử lý trạng thái hồ sơ | Trung bình |
|  |  |  |  |  | Hệ thống phải giao tiếp được với phân hệ thụ lý hồ sơ để nhận thông tin về trạng thái thụ lý hồ sơ |  |
|  |  |  |  |  | Hệ thống phải nhận và lưu trữ các thông báo về trạng thái thụ lý hồ sơ |  |
|  |  |  |  |  | Chuyên viên BP1C có thể theo dõi trạng thái thụ lý hồ sơ được nhận/trả tại bộ phận |  |
|  |  |  |  |  | Chuyên viên BP1C có thể phân loại các hồ sơ theo trạng thái thụ lý |  |
| 12 | Bổ xung hồ sơ | Chuyên viên BP1C,  Công dân |  | B | Bổ xung hồ sơ | Đơn giản |
|  |  |  |  |  | Hệ thống cho phép bổ xung giấy tờ vào các hồ sơ đang thụ lý |  |
|  |  |  |  |  | Công dân có thể nạp giấy tờ bổ xung (bản điện tử) khi có yêu cầu |  |
| 13 | Nhận hồ sơ | Chuyên viên BP1C |  | B | Nhận hồ sơ | Đơn giản |
|  |  |  |  |  | Chuyên viên BP1C có thể chọn và đánh dấu một hồ sơ đã nhận lại từ bộ phận thụ lý (BPTL) |  |
|  |  |  |  |  | Hệ thống phải in ra phiếu biên nhận nhận lại hồ sơ khi BP1C nhận lại hồ sơ từ các BPTL |  |
| 14 | Trả hồ sơ | Chuyên viên BP1C,  Công dân |  | B | Trả hồ sơ | Đơn giản |
|  |  |  |  |  | Chuyên viên BP1C có thể chọn hồ sơ để trả cho công dân |  |
|  |  |  |  |  | Hệ thống phải thay đổi trạng thái hồ sơ khi hồ sơ được trả |  |
|  |  |  |  |  | Hệ thống phải in hóa đơn thanh toán các dịch vụ |  |
| 15 | Lưu trữ hồ sơ | Chuyên viên BP1C |  | B | Lưu trữ hồ sơ | Đơn giản |
|  |  |  |  |  | Hệ thống phải quản lý lưu trữ hồ sơ |  |
|  |  |  |  |  | Hệ thống có thể tìm lại hồ sơ đã giải quyết trước đó nếu hồ sơ này có yêu cầu giải quyết lại |  |

### Tác nhân

*Bảng 3.5 Tác nhân*

|  |  |  |
| --- | --- | --- |
| **#** | **Tác nhân** | **Mô tả** |
|  | Công dân | Công dân giao tiếp với phần mềm |
|  | Trưởng BP1C | Trưởng bộ phận một cửa |
|  | Chuyên viên BP1C | Chuyên viên tại bộ phận một cửa |
|  | Chuyên viên thụ lý | Chuyên viên thụ lý hồ sơ |
|  | Lãnh đạo | Lãnh đạo |

Giải thích:

* Công dân: Công dân và các tổ chức với tư cách là những cá nhân/tổ chức có nhu cầu giải

quyết thủ tục hành chính đại diện cho chính cá nhân hay cho các tổ chức yêu cầu giải quyết các thủ tục hành chính.

* Trưởng BP1C: Người đứng đầu bộ phận một cửa với chức năng chính là phân chia công việc (tiếp nhận,chuyển thụ lý …) cho các chuyên viên bộ phận một cửa.
* Chuyên viên bộ phận một cửa: Người có trách nhiệm giao tiếp với công dân để tiếp nhận và trả hồ sơ. Ngoài ra,chuyên viên bộ phận một cửa còn có trách nhiệm trả lời thắc mắc của công dân và niêm yết thủ tục hành chính.
* Chuyên viên thụ lý: Người của một đơn vị thụ lý nhất định,có trách nhiệm lấy hồ sơ từ bộ phận một cửa để giải quyết cho công dân.
* Lãnh đạo: Người đưa ra quyết đinh cuối cùng với hồ sơ thủ tục hành chính.

### Ca sử dụng

#### Biểu đồ ca sử dụng (Usecase Diagram)

****

*Hình 3.2 Biểu đồ ca sử dụng*

**Ca sử dụng: Đăng nhập hệ thống**

*Tác nhân tham gia*: Chuyên viên bộ phận một cửa,Trưởng bộ phận một cửa,Quản trị viên.

*Mục đích*: Chứng thực người dùng bằng tên đăng nhập và mật khẩu để đảm bảo người dùng có quyền đối với hệ thống.

*Luồng sự kiện*:

*Biều đồ hoạt động*:



*Hình 3.3 Biểu đồ luồng hoạt động của usecase Login*

**Ca sử dụng : Niêm yết thủ tục hành chính**

Actor tham gia: Chuyên viên bộ phận một cửa,Công dân.

Mục đích: Đưa thông tin về thủ tục hành chính mới được niêm yết tới với người dân,đảm bảo người dân nắm được thông tin cũng như quy trình thực hiện đối với thủ tục hành chính mới.

**Ca sử dung: Tiếp nhận hồ sơ**

Actor tham gia: Chuyên viên bộ phận một cửa,Công dân.

Mục đích: Tiếp nhận hồ sơ,yêu cầu của công dân đối với thủ tục hành chính,đảm bảo hồ sơ của công dân đến đúng nơi quy định,không làm thất lạc hồ sơ.

**Ca sử dung: Chuyển thụ lý hồ sơ**

Actor tham gia: Chuyên viên bộ phận một cửa,Chuyên viên thụ lý.

Mục đích: Chuyển hồ sơ của công dân tới cơ quan tổ chức có thẩm quyền xử lý.

**Ca sử dụng: Điều chỉnh trách nhiệm chuyên viên bộ phận một cửa**

Actor tham gia: Trưởng bộ phận một cửa,chuyên viên bộ phận một cửa.

Mục đích: để phù hợp với tình hình thực tế,trưởng bộ phận một cửa có thể phân bổ nhân lực cho phù hợp bằng việc điểu chình quyền tiếp nhận,chuyển thụ lý … của chuyên viện bộ phận một cửa dưới quyền.

**Ca sử dụng: Xem thống kê hồ sơ đến**

Actor tham gia: Trưởng bộ phận một cửa.

Mục đích: Thống kê số lượng hồ sơ đến để trưởng bộ phận một cửa có kế hoạch dài hạn cho cơ quan doanh nghiệp.

**Ca sử dụng: Khởi tạo dữ liệu cho hệ thống**

Actor tham gia: Quản trị viên

Mục đích: Hệ thống mới không có dữ liệu,quản trị viên khởi tạo dữ liệu phù hợp với cơ quan tổ chức để đảm bảo hệ thống vận hành hiệu quả.

#### Biểu đồ luồng hoạt động (Activity Diagram)

**Ca sử dung: Login** Ca sử dụng này tương đối đơn giản xin không giải thích gì thêm.

**Ca sử dụng: Niêm yết thủ tục hành chính**



# CÀI ĐẶT

## Khung phát triển ứng dụng Symfony2

### Giới thiệu

Symfony là một framework tổ chức việc phát triển ứng dụng bằng việc tự động hóa nhiều mẫu được dùng cho một mục đích xác định. Một framework tạo cấu trúc cho mã, thúc đẩy lập trình viên viết những mã tốt hơn, dễ đọc hơn và dễ bảo trì hơn. Cuối cùng, là 1 framework làm cho việc lập trình dễ hơn bởi vì nó bao hàm những tác vụ phức tạp trong những câu lệnh đơn giản.

Symfony là 1 framework đầy đủ được thiết kế để tối ưu việc phát triển ứng dụng web bằng một vài đặc điểm chính. Với những người mới bắt đầu, nó phân chia những qui tắc nghiệp vụ, server logic và phần hiển thị của ứng dụng web. Nó có những công cụ và class để thu ngắn thời gian phát triển của những ứng dụng web phức tạp. Thêm vào đó, nó tự động hóa những công việc thông thường để lập trình viên có thể tập trung hoàn toàn vào những đặc điểm của ứng dụng. Kết quả cuối cùng của những điểm lợi này là không cần phải lặp lại (reinvent the wheel) mỗi lần tạo ứng dụng web!

Symfony được viết hoàn toàn bằng PHP 5. Nó đã được kiểm tra 1 cách hoàn hảo trong nhiều dự án thực tế khác nhau và thực sự được dùng cho những website kinh doanh điện tử có yêu cầu cao. Nó tương thích với hầu hết các loại cơ sở dữ liệu, bao gồm MySQL, PostgreSQL, Oracle, và Microsoft SQL Server. Nó chạy trên hệ điều hành \*nix và Windows. Hãy bắt đầu xem xét những tính năng của nó.

**Tính năng của Symfony**

Symfony được xây dựng để thực hiện các yêu cầu sau:

* Dễ cài đặt và cấu hình trên hầu hết các hệđiều hành (và được bảo đảm làm việc trên các hệ điều hành chuẩn \*nix và Windows)
* Độc lập với hệ cơ sở dữ liệu
* Dễ dùng trong hầu hết các trường hợp nhưng vẫn đủ mềm dẻo để thích nghi với những trường hợp phức tạp
* Hoạt động theo cấu hìnhđịnh trước --lập trình viên chỉcần cấu hình trong những trường hợp riêng biệt
* Tuân theo những hoạt động và mẫu thiết kế tốt nhất
* Sẵn sàng cho môi trường xí nghiệp --thích nghi với những chính sách và kiến trúc công nghệ thông tin và đủ ổn định cho những dự án dài hạn.
* Mã rất dễ đọc, với những chú thích kiểu phpDocumentor, dễdàng bảo trì
* Dễ dàng mở rộng, cho phép tích hợp với những thư viện khác

Symfony2 được trang bị đầy đủ các tính năng của một dự án web tự động

* Lớp internationalization có sẵn cho phép phiên dịch giao diện, dữ liệu cũng như là content localization.
* Phần presentation sử dụng template và layout có thể được xây dựng bởi người thiết kế HTML mà không cần có kiến thức về framework. Các helper làm giảm số lượng mã presentation phải viết bằng cách tóm gọn các phần của mã vào những function đơn giản.
* Các form hỗ trợ kiểm tra và điền tự động, và điều này đảm bảo chất lượng tốt của dữ liệu trong cơ sở dữ liệu và kinh nghiệm người sử dụng tốt hơn.
* Việc escape output bảo vệ các ứng dụng khỏi những tấn công thông qua dữ liệu bị hỏng.
* Những tính năng quản lý cache giúp giảm lượng băng thông và tải của máy chủ.
* Những tính năng phân quyền và xác nhận giúp việc tạo các quản lý những mục hạn chế và bảo mật người dùng dễ dàng hơn.
* Việc định hướng và các URL thông minh làm cho phần địa chỉ của các trang trở nên thân thiện với máy tìm kiếm.
* Các tính năng quản lý API và email sẵn có cho phép ứng dụng web vượt qua những tương tác trình duyệt cổ điển.
* Các danh sách trở nên thân thiện với người dùng hơn nhờ việc phân trang, sắp xếp và tìm kiếm 1 cách tự động.
* Các factory, plugins và mixins cung cấp khả năng mở rộng ở mức cao.
* Các tương tác AJAX dễ hiện thực nhờ vào các helper 1 dòng tóm lượt những hiệu ứng javascript tương thích với các trình duyệt.

**Môi trường và công cụ phát triển**

Để đáp ứng những yêu cầu của những doanh nghiệp có các chuẩn viết mã riêng và các qui tắc quản lý dự án, symfony có thể hoàn toàn được tùy biến. Mặc nhiên nó cung cấp 1 vài môi trường phát triển và được gói gọn với nhiều công cụ để tự động hóa những nhiệm vụ công nghệ phần mềm thông thường:

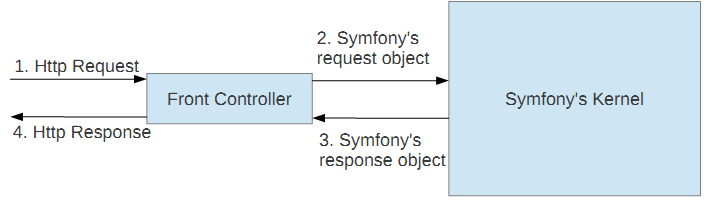
* Các công cụ sinh mã tự động rất tuyệt cho việc tạo bản prototype và phần quản lý backend chỉ với vài dòng lệnh.
* Unit và funtional testing framework sẵn có cung cấp những công cụ rất tốt cho phép test-driven development.
* Debug: giúp debug tất cả thông tin mà lập trình viên cần trên trang.
* Giao diện chế độ dòng lệnh tự động hóa triển khai các ứng dụng giữa hai server .
* Tính năng logging cung cấp cho adminnistrators những chi tiết đầy đủ về sự hoạt động của một ứng dụng

### Kiến trúc và hoạt động

#### Kiến trúc

Luồng hoạt động trong Symfony2 nhìn từ phía client

*Ảnh 4.4 Luồng giao tiếp giữa 3 thành phần chính trong Symfony2*



*Ảnh 4.5 Luồng giao tiếp giữa front Controller và Kernel*

1. Bộ điều khiển mặt trước (front controller) khởi động nhân ứng dụng (kernel), tiếp nhận Http Request, tạo đối tượng request và đưa Http Request vào đối tượng request được tạo, sau đó chuyển đối tượng request cho nhân
2. Nhân xử lý request, trả kết quả là đối tượng response cho bộ điều khiển mặt trước
3. Tại bộ điều khiển mặt trước, đối tượng response tạo và gửi Http Response cho client

*//Khởi động nhân ứng dụng*

***$kernel = new AppKernel('prod', false);***

***$kernel->loadClassCache();***

*//Tạo đối tượng request nắm giữ Http Request*

***$request = Request::createFromGlobals();***

*//Chuyển đối tượng request cho nhân xử lý, nhận kết quả xử lý là đối tượng response*

***$response = $kernel->handle($request);***

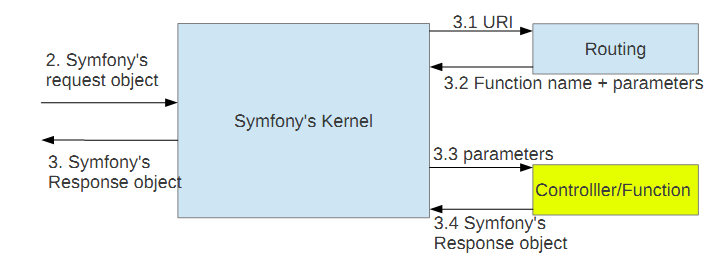
*//Tạo và gửi Http Response cho client*

***$response->send();***

//Tắt nhân

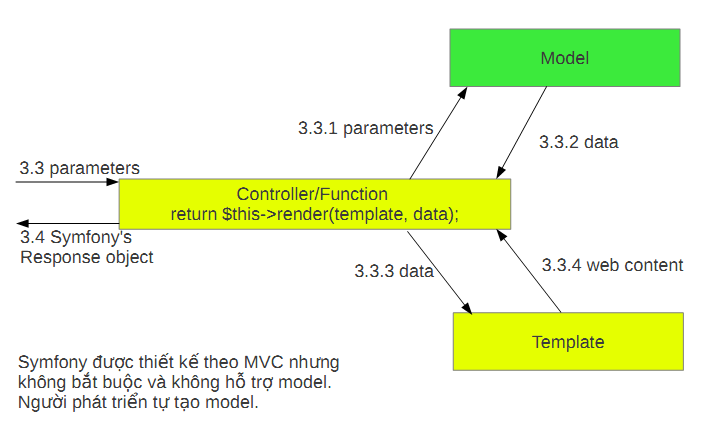
***$kernel->terminate($request, $response);***

Luồng hoạt động thực sự trong Symfony2

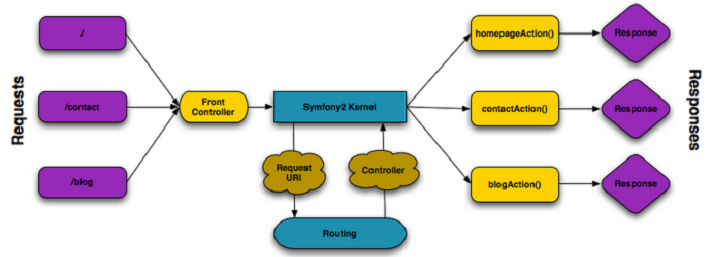


*Ảnh 4.6 Luống hoạt động trong Symfony2*

1. Symfony’s Kernel nhận Symfony’s Request Object,lọc URI và chuyển sang cho Routing.
2. Routing dựa vào thông tin từ URI và trả lại thông tin về router bao gồm cả function/controller nào sẽ được gọi.
3. Nhân Kernel sẽ thực hiện function/controller và trả về đối tượng Response.



Ta nhìn thấy rõ hơn điều này trong ví dụ dưới đây:



*Ảnh 4.7 Ví dụ về luồng hoạt động trong Symfony2*

1. Người dùng khởi tạo HTTP Request với URI “/” “/contact” và “/blog” tới server.
2. Front Controller tiếp nhận các request này và gửi tới Symfony2 Kernel dưới dạng các Symfony’s Request Object.
3. Symfony lọc các URI từ Symfony’s Request Object và gửi Request URI tới component Routing.
4. Routing dựa vào URI xác định controller và các tham số của nó và gửi lại về Symfony2 Kernel.
5. Symfony2 Kernel thực thi các controller được gửi về (homePageAction(),contcatAction() và blogAction()) và tạo Response.
6. Các đối tượng Response này sẽ được chuyển qua front controller và trả về client.

## Cơ sở dữ liệu

### Giải pháp cơ sở dữ liệu

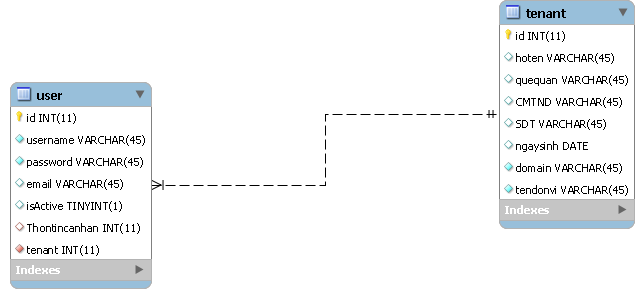
Như đã giới thiệu ở chương 2,mức 3 trong mô hình phần mềm dịch vụ SaaS đang được xem là tối ưu và được sử dụng nhiều nhất hiện nay. Vì vậy mô hình này được lựu chọn để triển khai phần mềm dịch vụ công.

Vì tất cả khách hàng sử dụng chung database lên việc phân biệt dữ liệu giữa các khách hàng với nhau là điều quan trọng nhất. Về giải pháp kỹ thuât, cơ sở dữ liệu cần có một bảng “khách hàng” để phân biệt các khách hàng với nhau. Tuy nhiên, điều này chỉ giúp phân biệt các khách hàng một cách đơn thuần mà chưa phân tách rõ ràng dữ liệu của chúng.

Một giải pháp được đưa ra là tất cả các bảng trong cơ sở dữ liệu tạo liên kết khóa ngoại tới bảng khách hàng. Giải pháp này có thể đảm bảo việc phân tách dữ liệu, tuy nhiên nó lại tạo ra một số lượng lớn ràng buộc dữ liệu không cần thiết,từ đó việc thêm xóa sửa sẽ phức tạp hơn.

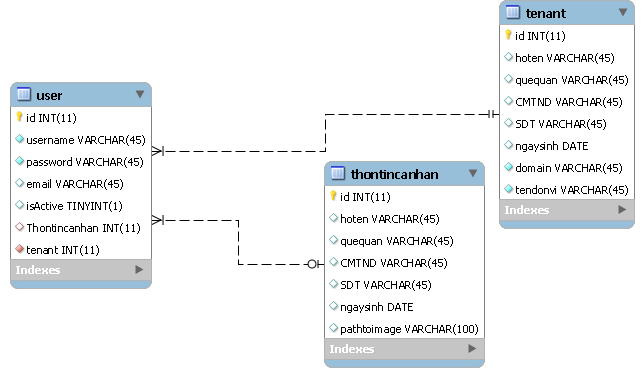
Giải pháp tối ưu cho bài toán này việc đưa khóa ngoại vào các bảng top-level (các bảng không chứa khóa ngoại mà chỉ làm khóa ngoại của bảng khác). Giải pháp này vẫn cho phép ta phân tách dữ liệu người dùng bằng việc truy vấn ghép lối các bảng mặt khác vẫn đảm bảo ràng buộc dữ liệu trong hệ thống đạt mức tối thiểu nhất.

Ví dụ,để phân biệt người dùng của hai khách hàng khác nhau chỉ cần kiểm tra “id” của khách hàng trong bảng người dùng. Việc này tương đối đơn giản.



*Hình 4.8 Ví dụ về phân tách dữ liệu*

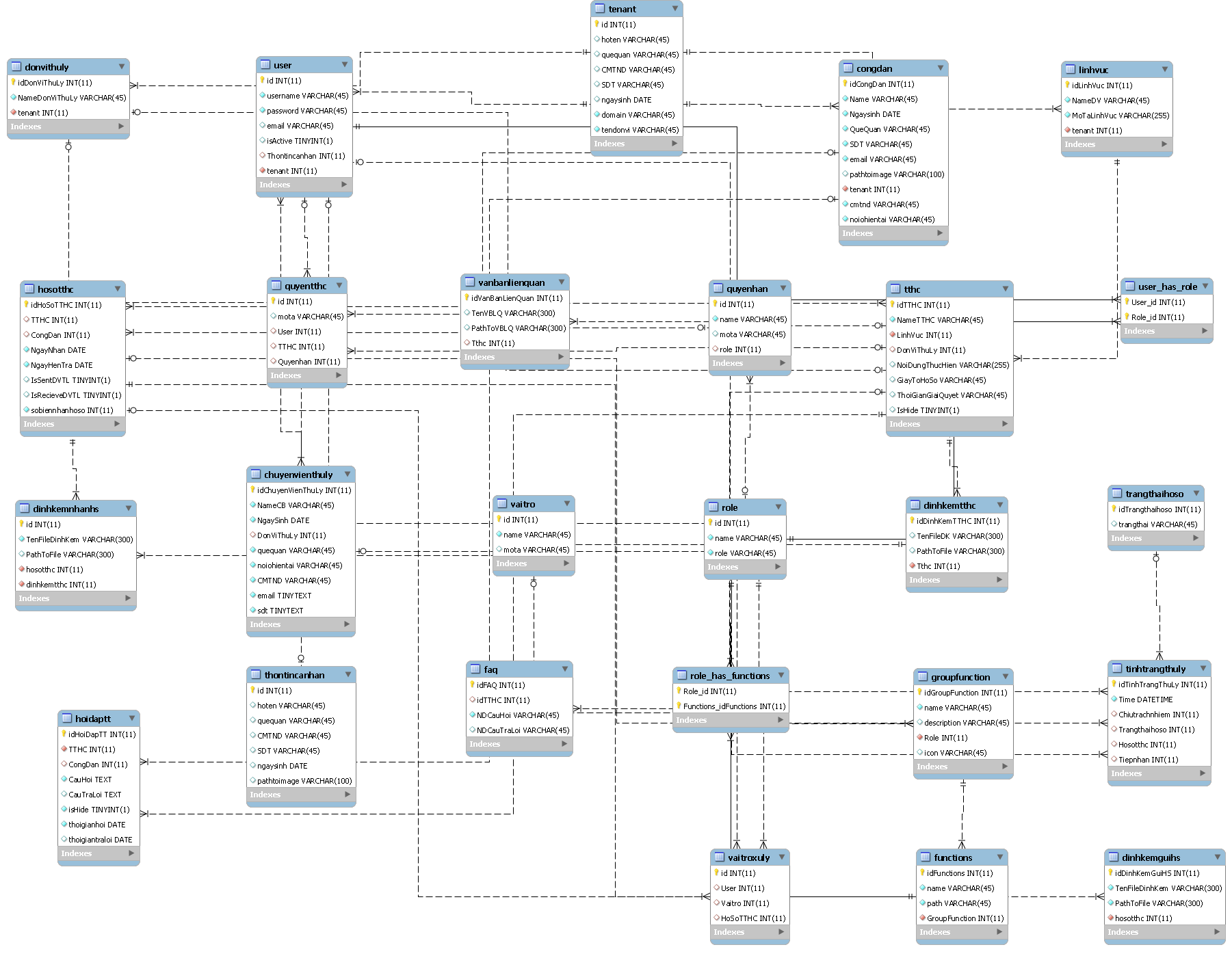
Ở mức khó hơn,khi cần kiểm tra thông tin khách hàng của bảng thông tin cá nhân (không thuộc top-level),việc cần làm là truy vấn ghép lối các bảng này với bảng người dùng để dựa vào định dang khánh hàng ở bảng này để phân biệt



*Hình 4.9 Ví dụ về phân tách dữ liệu*

### Cơ sở dữ liệu

Với giải pháp được đưa ra ở trên, dưới đây là cơ sở dữ liệu thực sự được cài đặt trong ứng dụng



*Hình 4.10 Cơ sở dữ liệu*

## Mức đô hoàn thành

1. Đăng ký làm TTHC.

Hoàn thành.

1. Xem các thủ tục được nhận/trả hồ sơ

Hoàn thành.

1. Phân ekip nhận/trả hồ sơ

Hoàn thành.

1. Log-in

Hoàn thành.

1. Thống kê hồ sơ nhận/trả

Hoàn thành.

1. Xem thông tin ekip

Hoàn thành.

1. Theo dõi hồ sơ đăng ký

Hoàn thành.

1. Hướng dẫn làm đăng ký

Hoàn thành.

1. Tiếp nhận hồ sơ

Hoàn thành.

1. Chuyển thụ lý hồ sơ

Hoàn thành.

1. Cập nhật và xử lý trạng thái hồ sơ

Hoàn thành.

1. Thông báo trạng thái thụ lý hồ sơ

Hoàn thành.

1. Bổ xung hồ sơ

Hoàn thành.

1. Nhận hồ sơ

Hoàn thành.

1. Trả hồ sơ

Hoàn thành.

1. Lưu trữ hồ sơ

Hoàn thành.

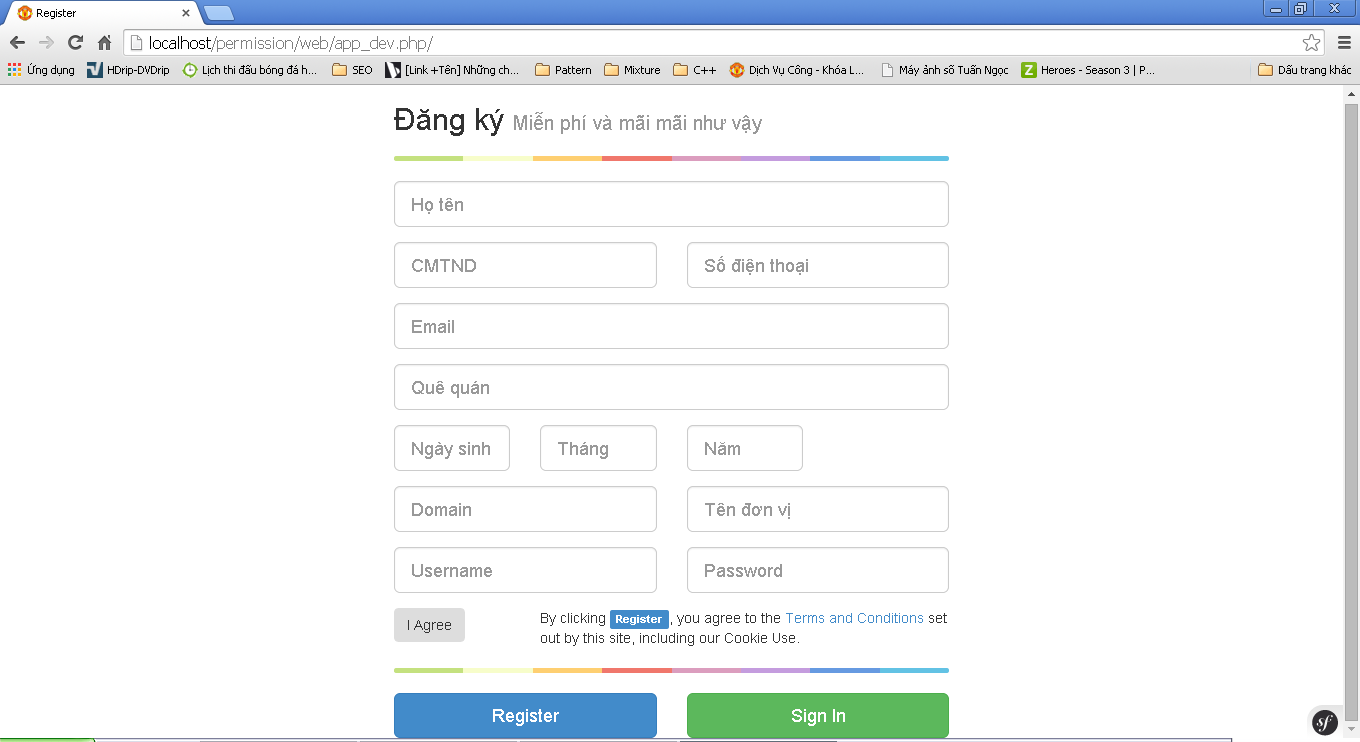
## Giới thiệu sử dụng

### Quản lý bên thuê

### Đăng ký đơn vị sử dụng hệ thống.

Người đăng ký – đại diện cho tổ chức,doanh nghiệp đăng ký sử dụng hệ thống.

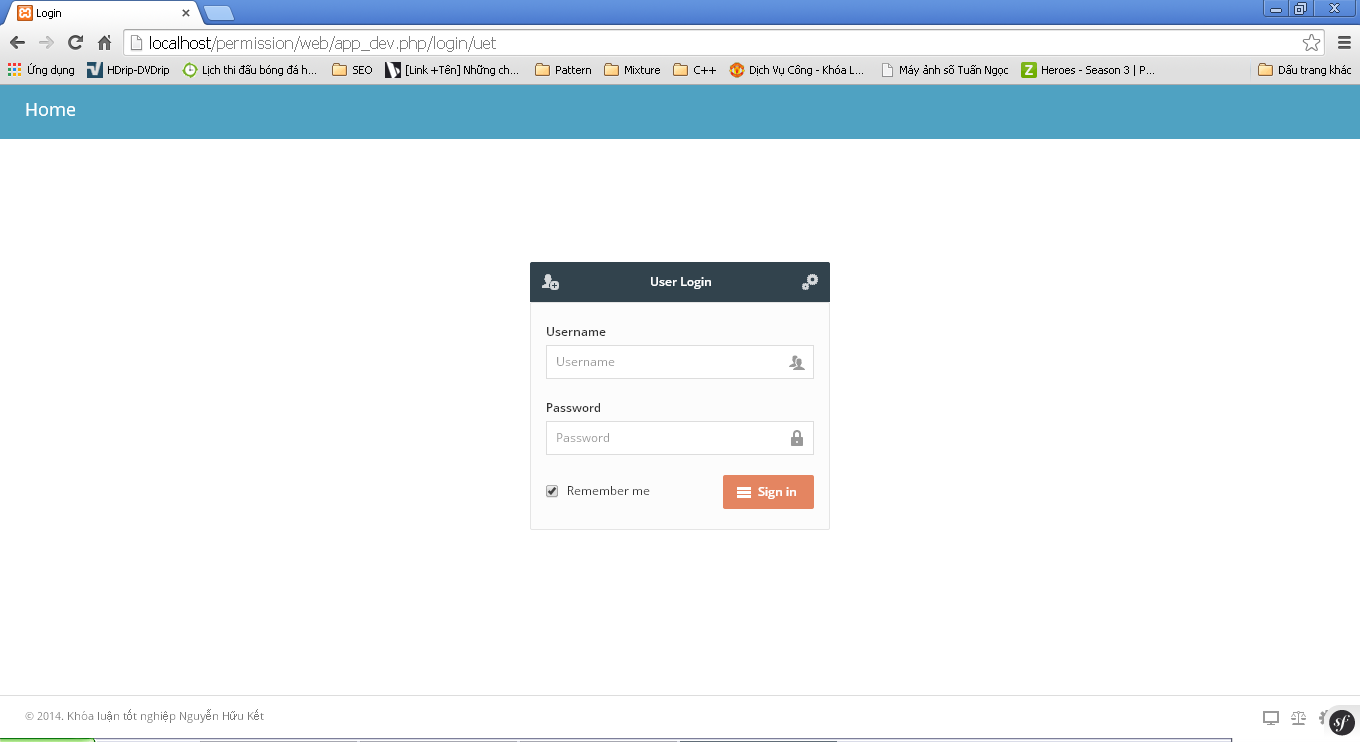
Người đăng ký nhập đầy đủ thông tin vào form để hệ thống có thể đưa vào và quản lý thành phần doanh nghiệp tổ chức của người đăng ký. Người đăng ký cần chú ý trường domain và tên đơn vị là 2 thành phần sẽ đại diện cho doanh nghiệp,tổ chức sau này.



***Hình*** *4.8 Giao diện đăng ký sử dụng dịch vụ*

### Đăng nhập vào hệ thống.

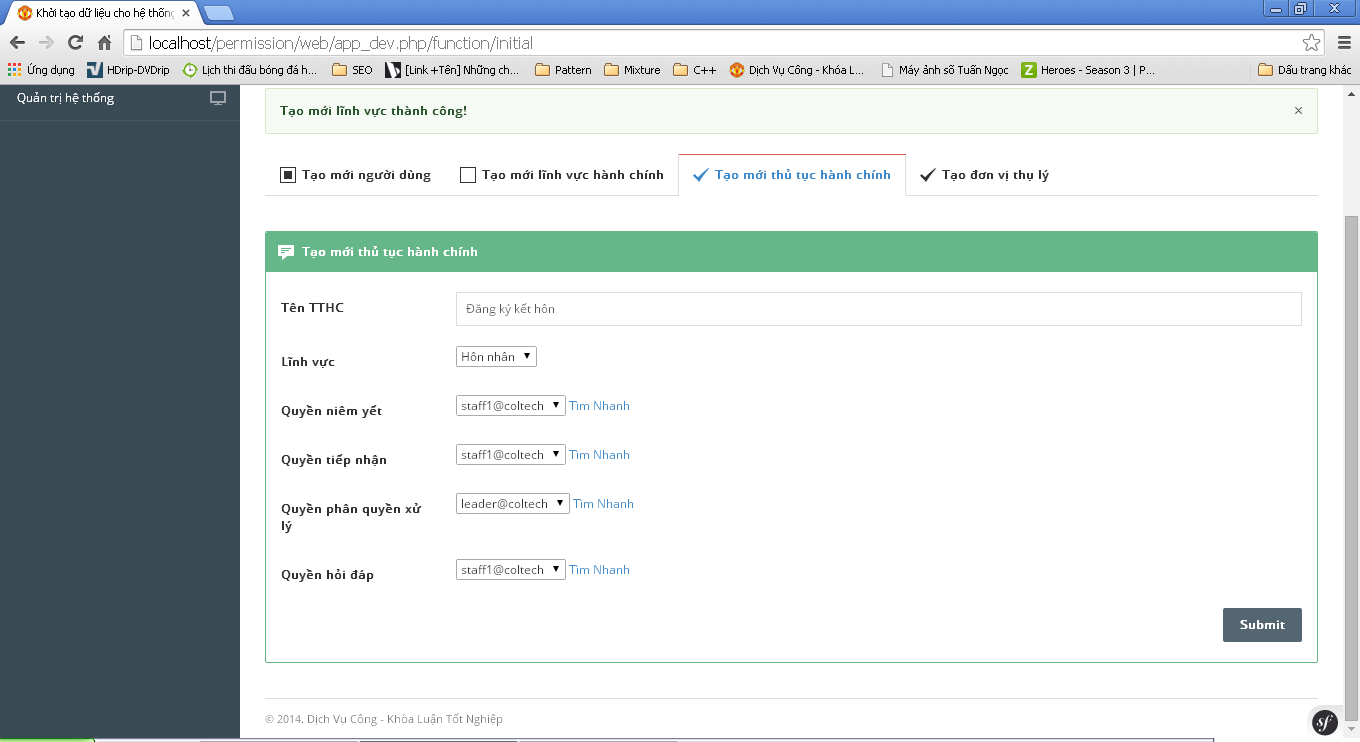
Sau khi đăng ký thành công,hệ thống tự động chuyển hướng người dùng tới trang đăng nhập thích hợp để người dùng có thể login vào hệ thống của mình.



Người dùng nhập username và password đã đăng ký ở trên để có thể login vào trang quản trị viên để khởi tạo dữ liệu cho hệ thống. Checkbox “Remmember me” được tích để đảm bào người dùng không cần nhập lại username và password cho lần đăng nhập tới.

### Niêm yết thủ tục hành chính

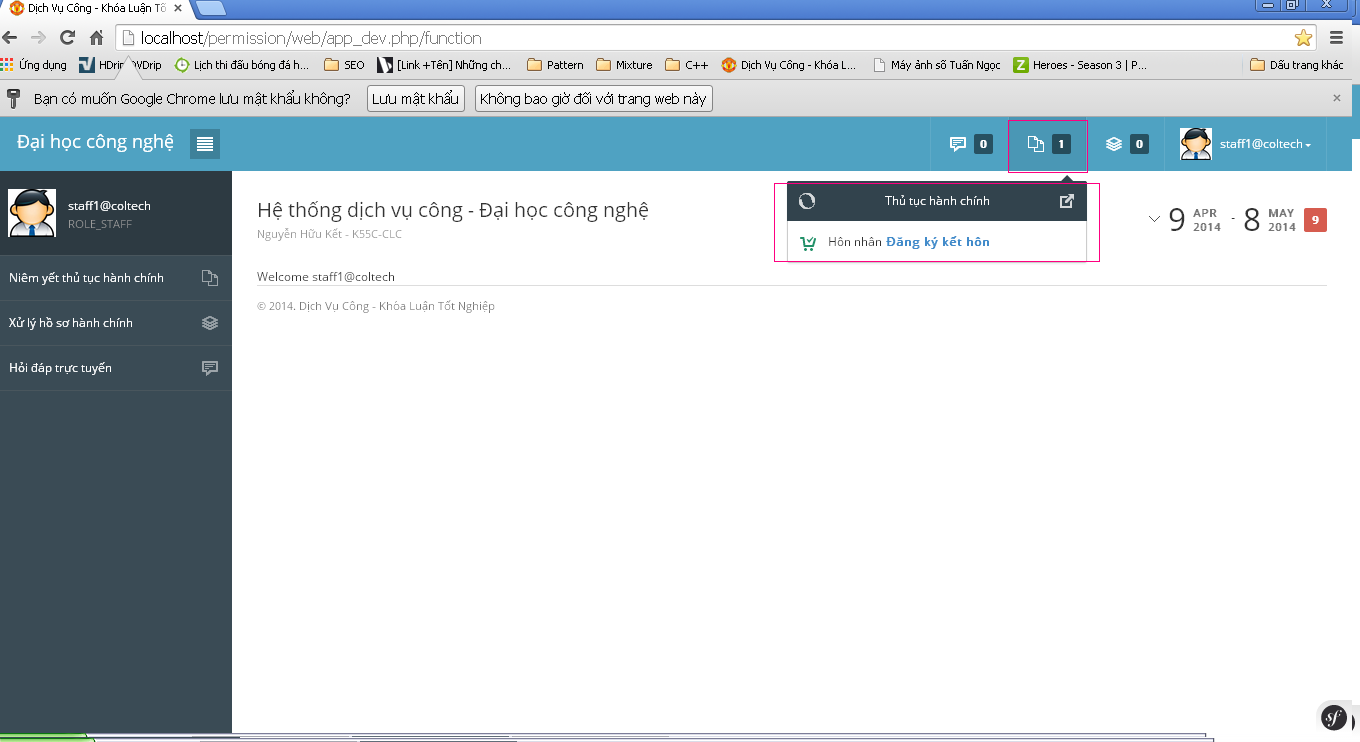
Quản trị viên vào chức năng khởi tạo dữ liệu và chọn tab tạo mới thủ tục hành chính



*Hình 4.9 Tạo mới thủ tục hành chính*

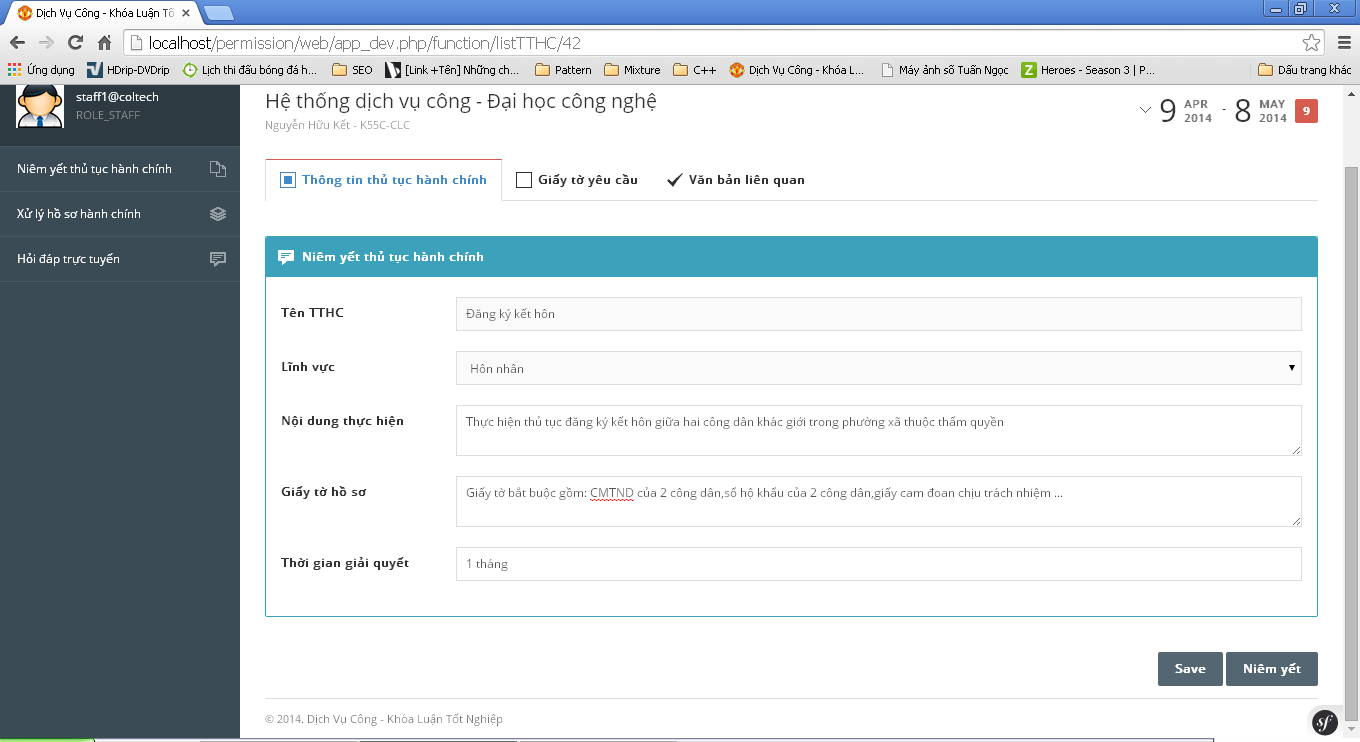
Ở đây,quản trị viên sẽ nhập tên thủ tục hành chính,chọn lĩnh vực của thủ tục hành chính và chọn người dùng có trách nhiệm với thủ tục hành chính này. Lưu ý,quyền niêm yết,tiếp nhận và hỏi đáp chỉ có hiệu lực với chuyên viên bộ phận một cửa. Quyền phân quyền xử lý chỉ có hiệu lực với lãnh đạo. Quản trị viên ấn nút Submit để hoàn tất việc khai báo thủ tục hành chính mới

Khi chuyên viên bô phận một cửa được phân quyền niêm yết đăng nhập vào hệ thống (Ở hình trên là staff1),hệ thống sẽ báo cho chuyên viên này biết có một thủ tục hành chính cần niêm yết.

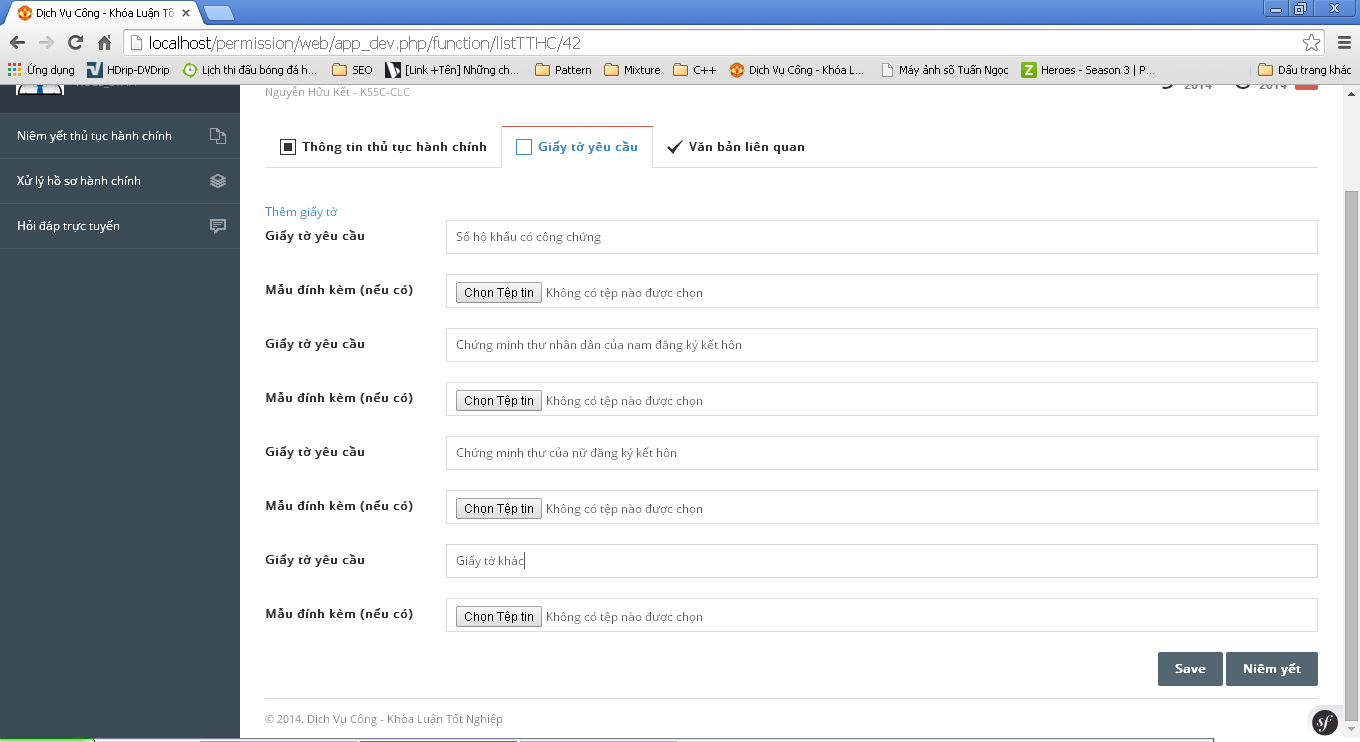


Chuyên viên bộ phận một cửa có thể truy cập nhanh tại thông báo hoặc vào menu Niêm yết thủ tục hành chính.

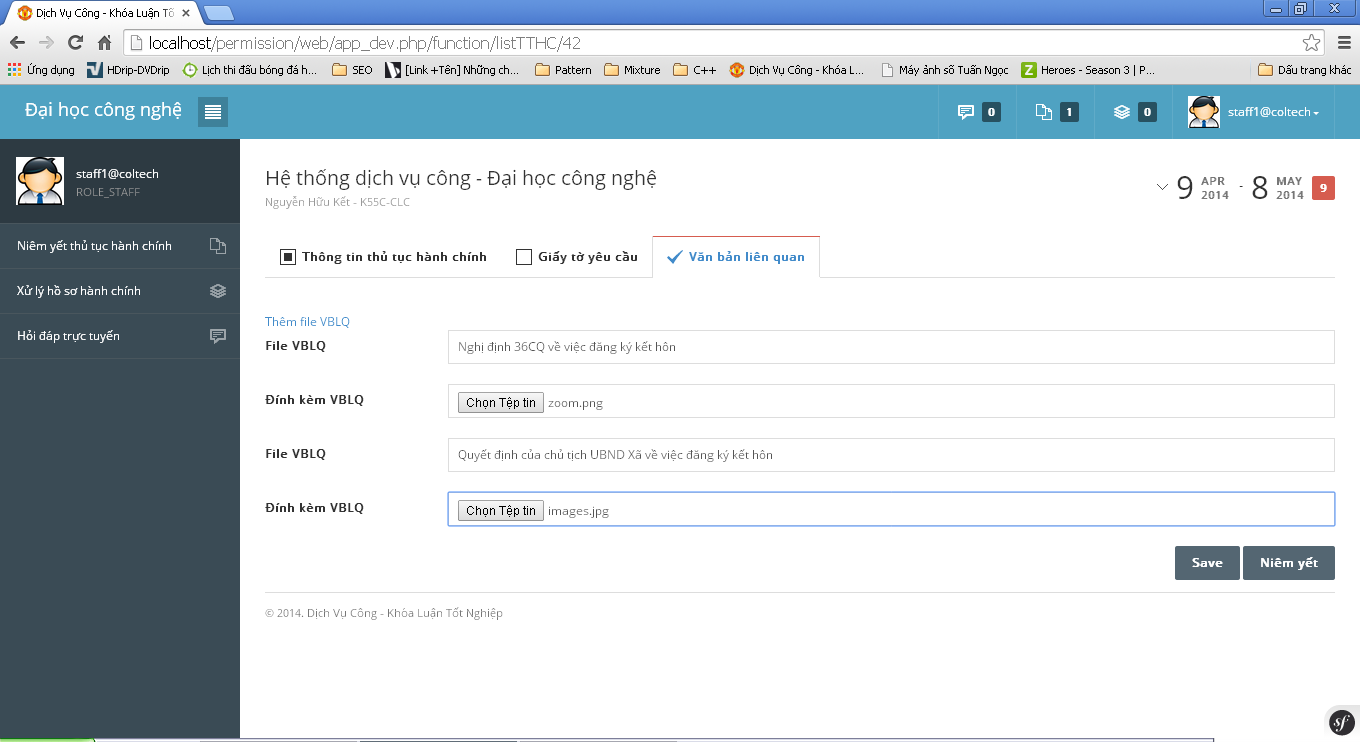
Ở tab đầu tiên,chuyên viên bộ phận một cửa điền đầy đủ thông tin về thủ tục hành chính cần niêm yết.



Ở tab thử hai,chuyên viên bộ phận một cửa đính kèm giấy tờ yêu cầu và mẫu của giấy tờ nếu có. Chọn thêm giấy tờ để có thể thêm được nhiều giấy tờ hơn nếu cần.



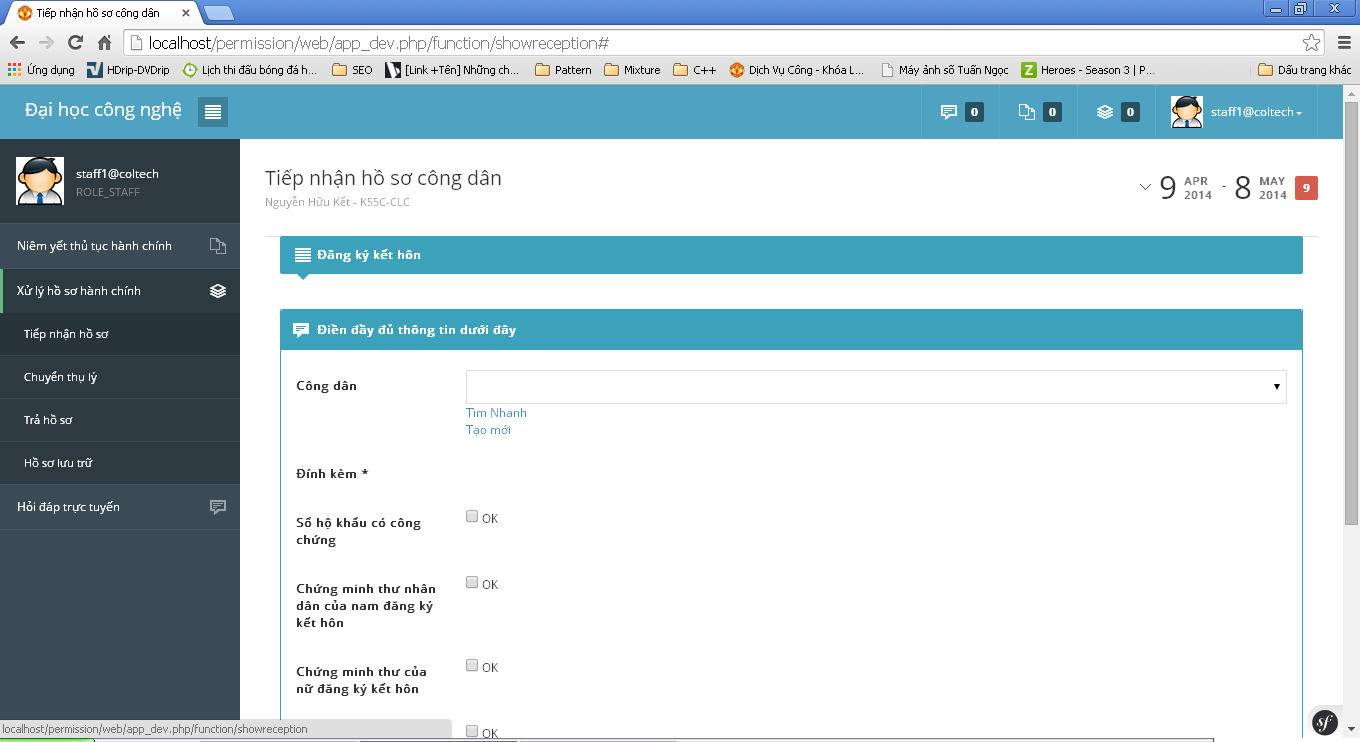
Ở tab cuối cùng,chuyên viên bộ phận một cửa nhập các văn bản liên quan,văn bản pháp lý của thủ tục hành chính. Chọn thêm văn bản,nếu cần.



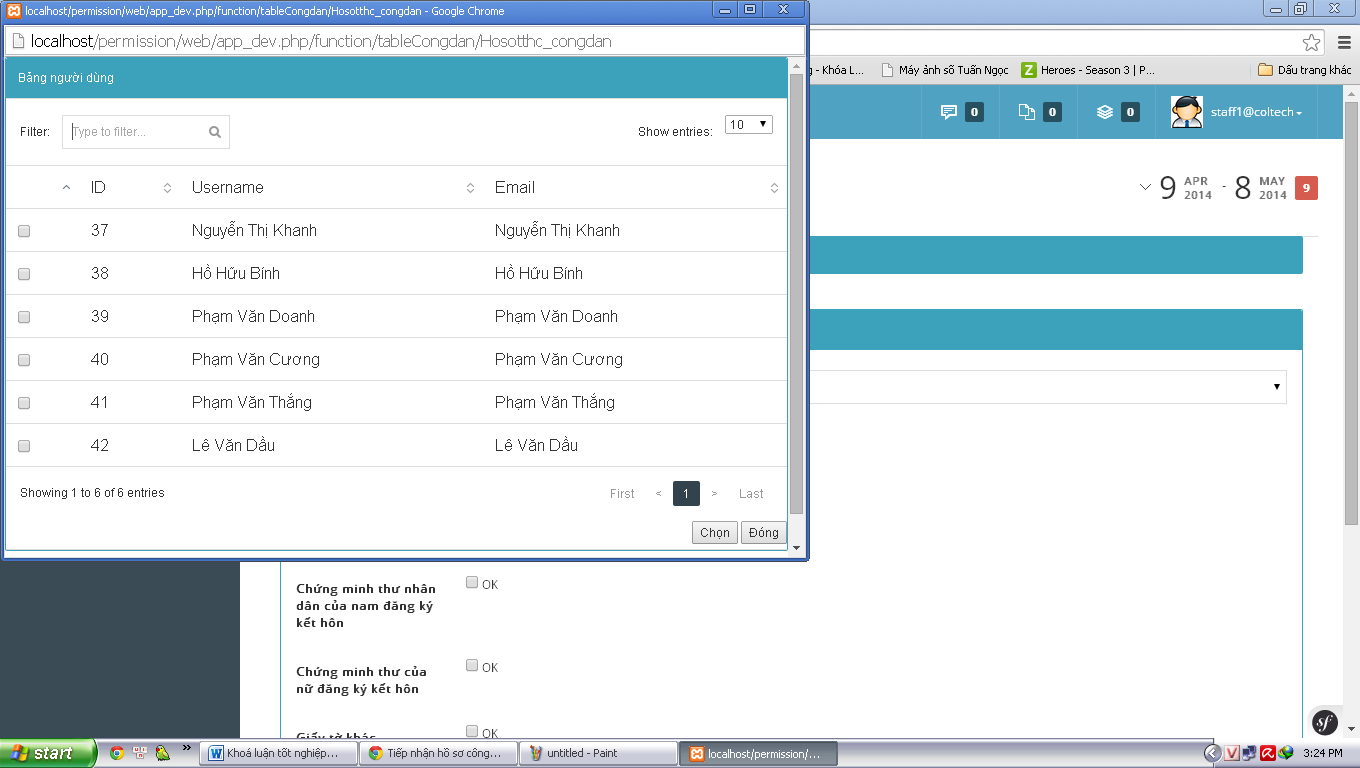
Sau khi điền đầy đủ thông tin,chuyên viên ấn nút “Niêm yết” để niêm yết thủ tục hành chính hoặc “Save” để lưu thông tin thủ tục đã nhập nhưng chưa muốn niêm yết. Đến đây,thủ tục hành chính đã được niêm yết thành công.

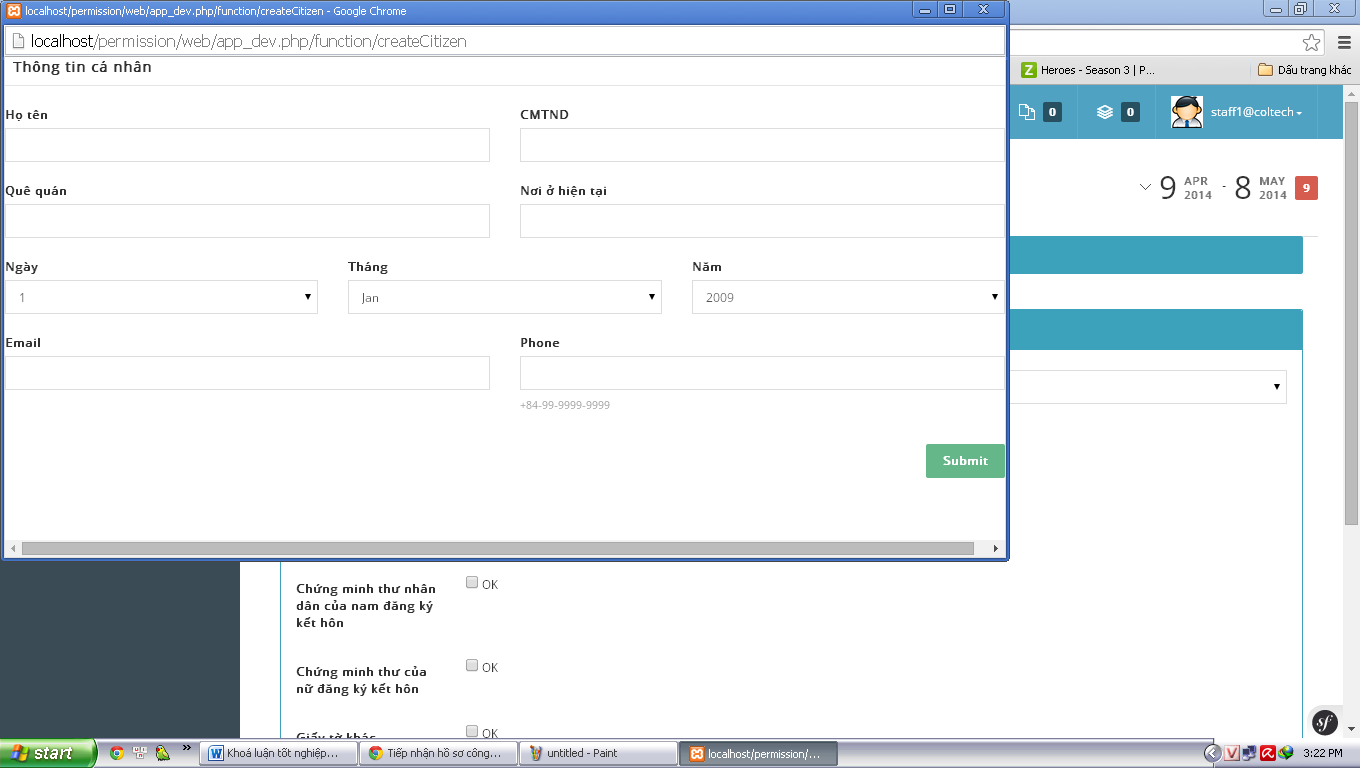
### Tiếp nhận hồ sơ

Ca sử dụng này bắt đầu khi có một công dân tới bộ phận một cửa đăng ký làm thủ tục hành chính. Chuyên viên bộ phận một cửa chọn menu Xử lý hồ sơ > Tiếp nhận hồ sơ.

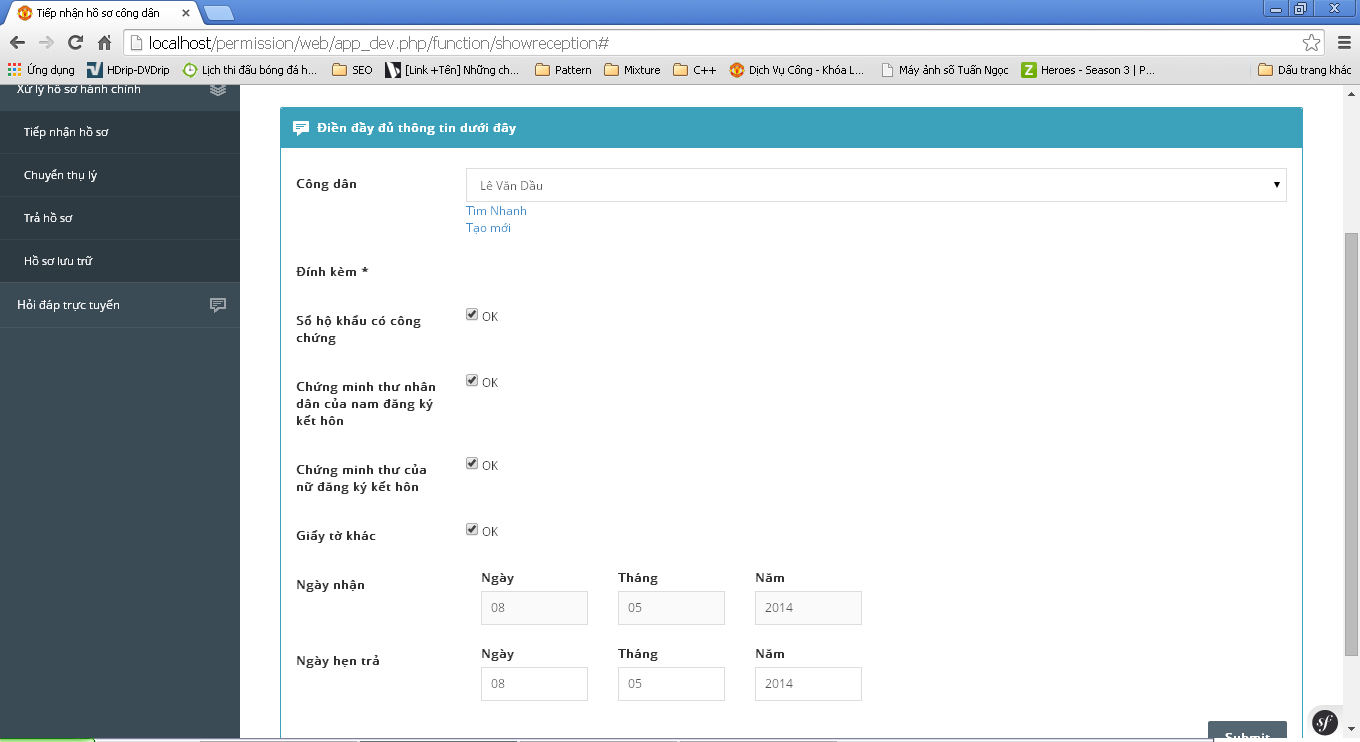


Nếu công dân đã đăng ký trên trang chủ hệ thống,chuyên viên bộ phận một cửa chọn tìm nhanh để lọc đúng thông tin người muốn nộp hồ sơ. Nếu công dân chưa đăng ký,chuyên viên chọn tạo mới.

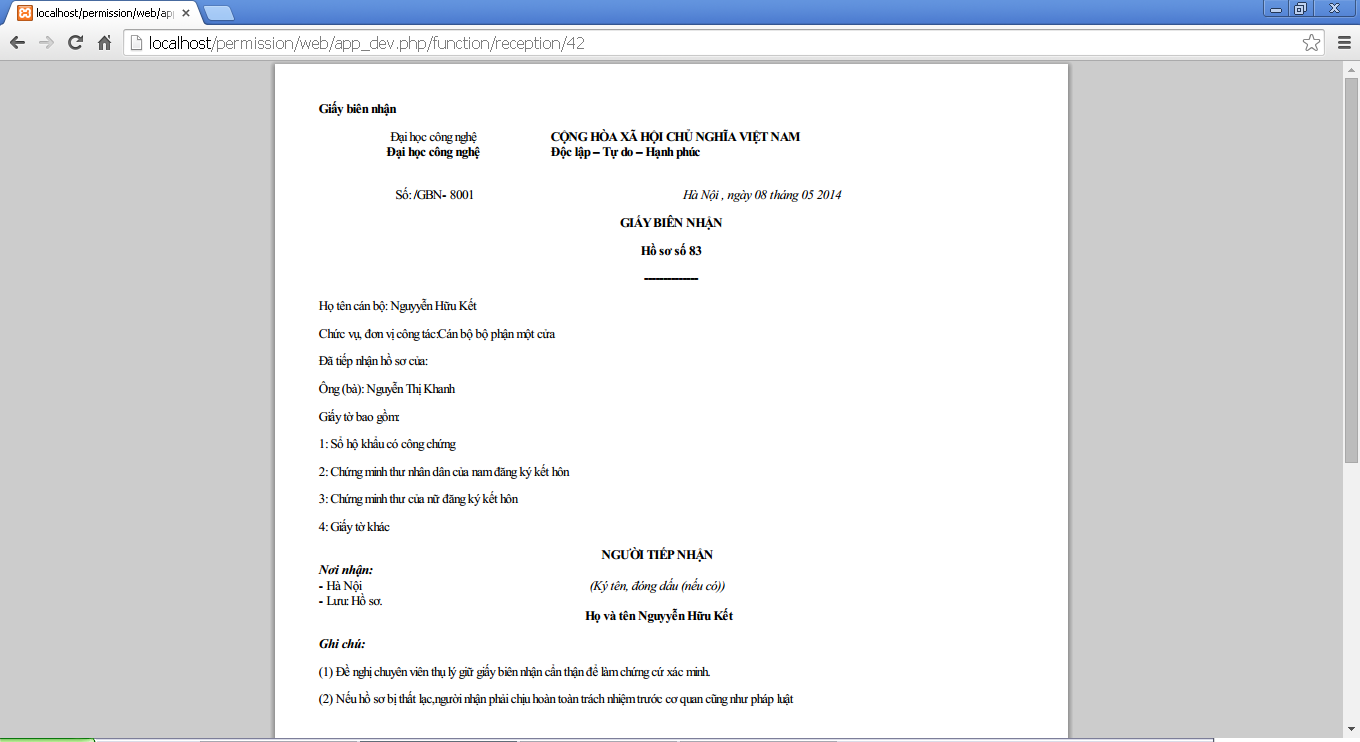




Sau chọn đương công dân đăng ký làm thủ tục hành chính,chuyên viên bộ phận một cửa kiểm tra giấy tờ của công dân có đầy đủ như yêu cầu của thủ tục hành chính không. Nếu thiếu,hồ sơ sẽ không thể được tiếp nhận.



Chuyên viên ấn nút Sutmit,hệ thống sẽ sinh ra một giấy biên nhận đã được điền đầy đủ thông tin cho chuyên viên có thể in ký và đóng dấu.

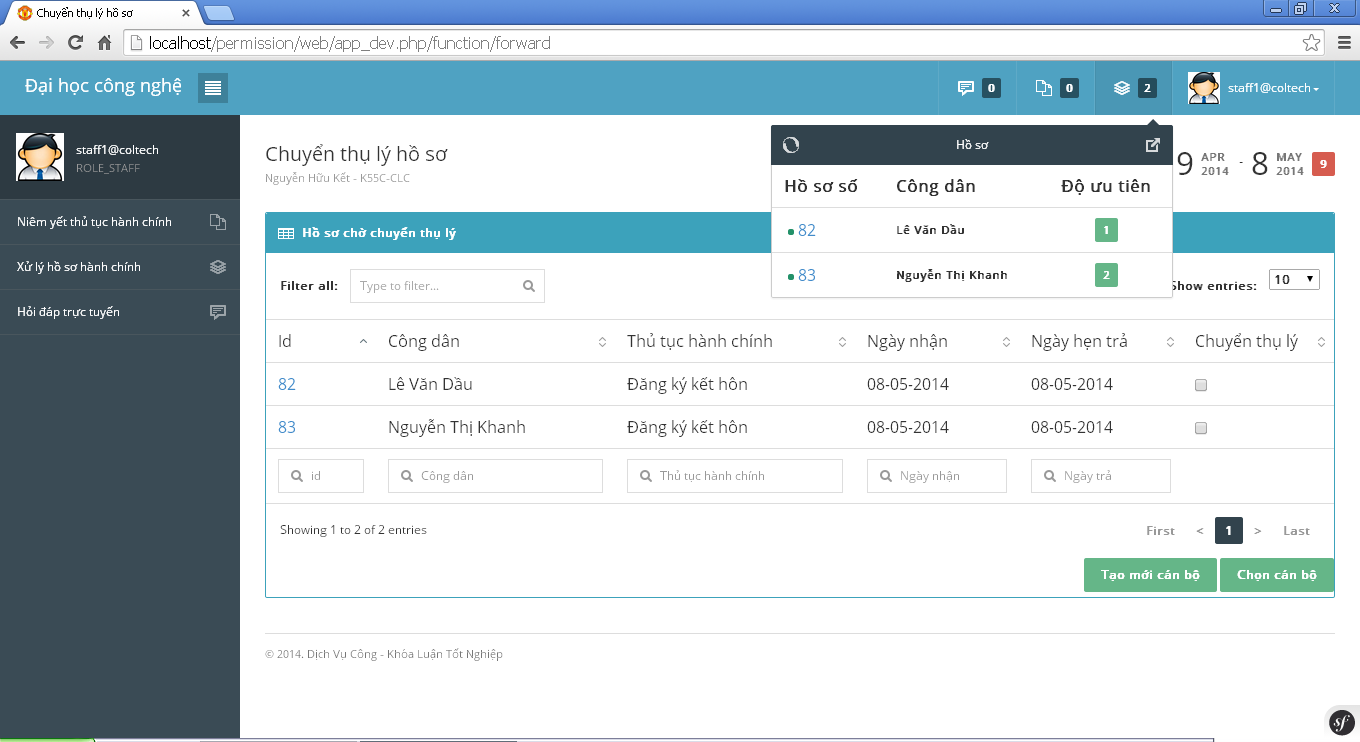


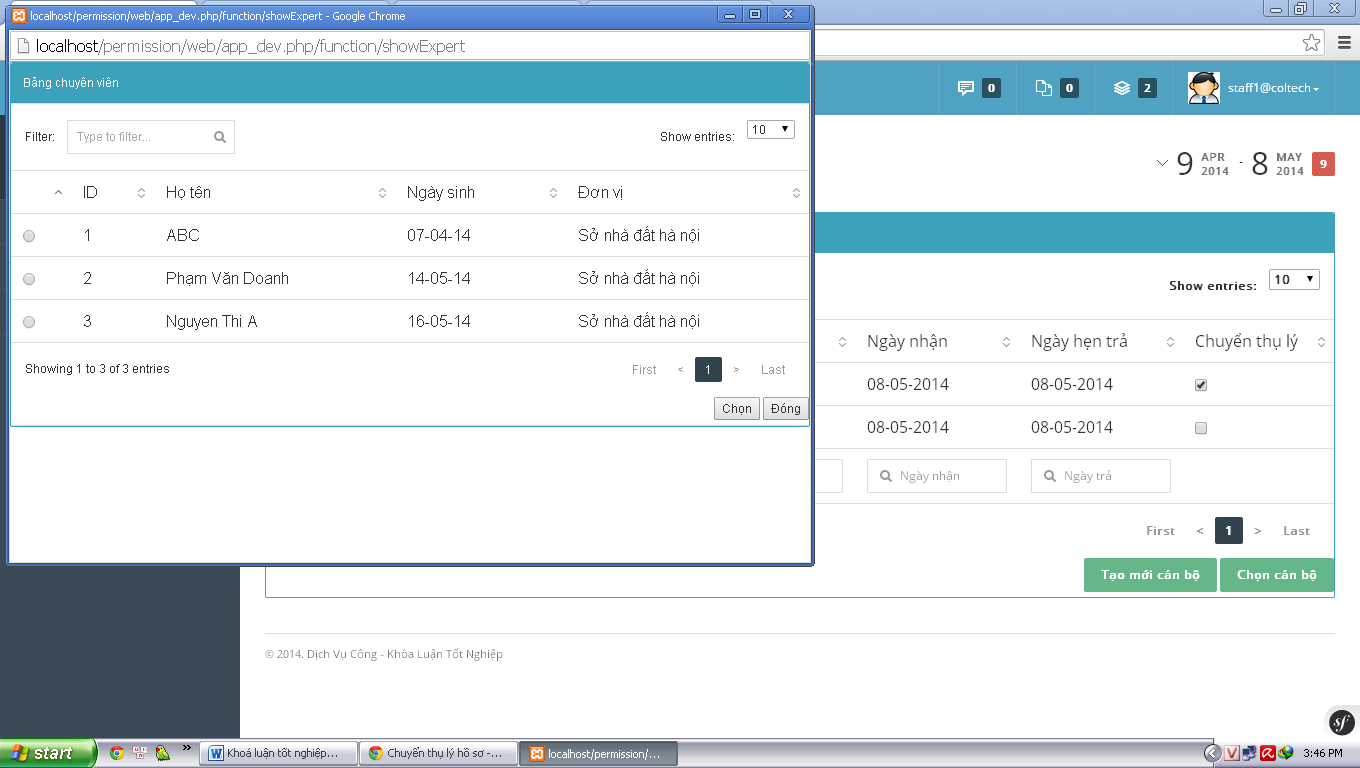
### Đến đây quá trình nộp hồ sơ đã hoàn tất.

### Chuyển thụ lý hồ sơ

Ca sử dụng bắt đầu khi chuyên viên thụ lý tới tiếp nhận hồ sơ. Chuyên viên bộ phận một cửa cần kiểm tra hồ sơ trước khi chuyển thụ lý. Khi nộp hồ sơ thành công,chuyên viên bộ phận một cửa sẽ nhận được notification thông báo có hồ sơ đến. Tại đây,chuyên viên có thể xem thông tin hồ sơ đã được nộp. Độ ưu tiên của hồ sơ được tính tăng dần từ thời điểm hiện tại tới thời gian cần trả hồ sơ. Hồ sơ cần trả sớm sẽ có độ ưu tiên cao hơn.

Sau khi kiểm tra thông tin hồ sơ,chuyên viên bộ phận một cửa tích chọn hồ sơ cần chuyển thụ lý. Nếu chuyên viên thụ lý đã có trong hệ thống,chuyên viên bộ phận một chọn “Chọn cán bộ”,nếu chưa chuyên viên chọn “Tạo mới cán bộ”.





# KẾT LUẬN

## Những việc đã làm được

### Tìm hiểu bài toán dịch vụ công

### Phân tích và thiết kế phân hệ một cửa

### Cài đặt phân hệ một cửa

### Tìm hiểu và sử dụng khung phát triển ứng dụng web Symfony2

## Hướng phát triển

Trong ngắn hạn,hướng phát triển chính của ứng dụng là cải thiện chức năng và hiệu suất sử dung. Thứ nhất, hệ thống cần bám sát hơn quy trình nghiệp vụ của bộ phận một cửa. Từ đó hệ thống cần hỗ trợ nhiều hơn công dân và chuyên viên bộ phận một cửa trong quá trình luân chuyển hồ sơ và nhận lại kết quả. Thứ hai,cải thiện một số chức năng hiện tại để hoàn thiện hơn,và có thể phát triển thêm một số chức năng mới.

Trong dài hạn,hệ thống sẽ cải thiện để tăng mức độ cá nhân hóa cho khách hàng. Khách hàng có thể điều chỉnh quy trình nghiệp vu, tùy biến chức năng, … từ đó làm cho ứng dụng phù hợp hơn với thực trạng của cơ quan, văn phòng khách hàng. Xa hơn nữa, hệ thống có thể được phân cấp để phục vụ trong phạm vi cả nước. Ví dụ, chính phủ mua lại toàn bộ hệ thống và đăng ký cho 64 tỉnh thành phố một site khách hàng. Sau đó, 64 tỉnh thành phố lại đăng ký cho các quận huyện trực thuộc một site con làm khách hàng của khách hàng. Tiếp đó, các quận huyện này lại phân xuống cấp làng,xã thị trấn … Từ đó tạo ra sự đồng bộ trong luân chuyển hồ sơ, tạo ra một chính quyền, chính phủ điện tử rất mạnh.