TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

──────── \* ───────



ĐỒ ÁN

**TỐT NGHIỆP ĐẠI HỌC**

NGÀNH CÔNG NGHỆ THÔNG TIN

**Xây dựng ứng dụng tư vấn bán hàng MShopkeeper**

Sinh viên thực hiện: **Đỗ Đức Thanh**

Lớp CNTT 2.03 – K58

Giáo viên hướng dẫn: [PGS-TS] **Cao Tuấn Dũng**

HÀ NỘI 6-2018

**PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP**

1. Thông tin về sinh viên

Họ và tên sinh viên: Đỗ Đức Thanh

Điện thoại liên lạc: 01655871797 Email: doducthanh95@gmail.com

Lớp: CNTT2.03 K58 Hệ đào tạo: Đại học

Đồ án tốt nghiệp được thực hiện tại: Viện Công nghệ thông tin và truyền thông, trường Đại học Bách Khoa Hà Nội

Thời gian làm ĐATN: Từ ngày 10/01/2018 đến 15/05/2018

2. Mục đích nội dung của ĐATN

Ứng dụng tư vấn bán hàng là một phần trong hệ thống quản lí cửa hang thời trang. Ứng dụng cho phép nhân viên bán hàng tư vấn thông tin sản phẩm một cách nhanh chóng và hiệu quả.

3. Các nhiệm vụ cụ thể của ĐATN

* Phát hiện và Phân tích yêu cầu của phần mềm
* Thiết kế phần mềm về mặt chức năng
* Xây dựng server API, thiết kế database
* Xây dựng giao diện phía client
* Tìm hiểu kiến trúc MVVM

4. Lời cam đoan của sinh viên:

Tôi – Đỗ Đức Thanh- cam kết ĐATN là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *PGS*-*TS* Cao Tuấn Dũng, kết hợp với công ty MISA.

Các kết quả nêu trong ĐATN là trung thực, không phải là sao chép toàn văn của bất kỳ công trình nào khác.

|  |  |
| --- | --- |
|  | *Hà Nội, ngày 22 tháng 05 năm 2018*  Tác giả ĐATN  *Đỗ Đức Thanh* |

5. Xác nhận của giáo viên hướng dẫn về mức độ hoàn thành của ĐATN và cho phép bảo vệ:

|  |  |
| --- | --- |
|  | *Hà Nội, ngày tháng năm 2018*  Giáo viên hướng dẫn  *PGS-TS Cao Tuấn Dũng* |

**Mục Lục**

[Danh Sách Bảng Biểu 5](#_Toc503479013)

[Danh Sách Hình Vẽ 6](#_Toc503479014)

[Danh Sách Thuật Ngữ Sử Dụng 8](#_Toc503479015)

[Danh Sách Từ Viết Tắt Sử Dụng 10](#_Toc503479016)

[Tóm Tắt Nội Dung Đồ Án Tốt Nghiệp 11](#_Toc503479017)

[Lời Cảm Ơn 13](#_Toc503479018)

[Chương 1 Đặt Vấn Đề Và Tổng Quan Về Bài Toán 14](#_Toc503479019)

[1 Đặt Vấn Đề 14](#_Toc503479020)

[1.1 Bối cảnh Bài Toán 14](#_Toc503479022)

[1.2 Mục đích phát triển đề tài 14](#_Toc503479023)

[2 Tổng quan về bài toán 14](#_Toc503479024)

[Chương 2 Tổng Quan Về Hệ Thống Và Công Nghệ Sử Dụng 16](#_Toc503479025)

[1 Tổng quan về hệ thống 16](#_Toc503479026)

[2 Công nghệ sử dụng 16](#_Toc503479027)

[2.1 Microservice 16](#_Toc503479028)

[2.2 Angular 18](#_Toc503479029)

[2.3 Spring boot 19](#_Toc503479030)

[2.4 Jhipster 20](#_Toc503479031)

[2.5 MySQL 23](#_Toc503479032)

[Chương 3 Phân Tích Và Thiết Kế Hệ Thống 25](#_Toc503479033)

[1 Các tác nhân của hệ thống 25](#_Toc503479034)

[2 Phân tích hệ thống về mặt chức năng 26](#_Toc503479035)

[2.1 Biểu đồ use case Tổng quan 26](#_Toc503479036)

[2.2 Biểu đồ use case phân rã 27](#_Toc503479037)

[2.3 Quy trình nghiệp vụ 29](#_Toc503479038)

[Chương 4 Xây Dựng Và Phát Triển Hệ Thống 47](#_Toc503479039)

[1 Thiết kế tổng quan 47](#_Toc503479040)

[2 Thiết kế Cơ sở dữ liệu 47](#_Toc503479041)

[2.1 Cơ sở dữ liệu 47](#_Toc503479042)

[2.2 Chi tiết các bảng cơ sở dữ liệu 49](#_Toc503479043)

[3 Thiết kế giao diện 61](#_Toc503479044)

[3.1 Giao diện chức năng thêm Customer 61](#_Toc503479045)

[3.2 Giao diện chức năng tạo Department 62](#_Toc503479046)

[3.3 Giao diện chức năng thêm Requester 64](#_Toc503479047)

[3.4 Giao diện cho chức năng tạo Interpreter 66](#_Toc503479048)

[3.5 Giao diện cho chức năng thêm Bill rate 67](#_Toc503479049)

[3.6 Giao diện cho chức năng tạo lịch hẹn 69](#_Toc503479050)

[3.7 Giao diện cho quản lý lịch hẹn 71](#_Toc503479051)

[3.8 Giao diện cho quản lý lịch hẹn của Interpreter 73](#_Toc503479052)

[Chương 5 Triển Khai, Cài Đặt Và Kiểm Thử 75](#_Toc503479053)

[1 Triển khai và Cài đặt hệ thống 75](#_Toc503479054)

[Triển khai và cài đặt 75](#_Toc503479055)

[Yêu cầu về dữ liệu 75](#_Toc503479056)

[2 Kiểm thử 75](#_Toc503479057)

[Chương 6 Tổng Kết Và Hướng Phát Triển 77](#_Toc503479058)

[1 Kết quả mà sản phẩm đạt được 77](#_Toc503479059)

[2 Kết quả mà bản thân đạt được 77](#_Toc503479060)

[3 Hướng phát triển 78](#_Toc503479061)

[Tài Liệu Tham Khảo 79](#_Toc503479062)

# Danh Sách Bảng Biểu

[Bảng 1 Cấu trúc dữ liệu bảng “Intake” 50](#_Toc501681060)

[Bảng 2 Cấu trúc dữ liệu bảng “Customer” 52](#_Toc501681061)

[Bảng 3 Cấu trúc dữ liệu Bảng “Requester” 53](#_Toc501681062)

[Bảng 4 Cấu trúc dữ liệu bảng “Interpreter” 54](#_Toc501681063)

[Bảng 5 Cấu trúc dữ liệu bảng “User” 55](#_Toc501681064)

[Bảng 6 Cấu trúc dữ liệu bảng “Sup-department” 56](#_Toc501681065)

[Bảng 7 Cấu trúc dữ liệu bảng “Department” 57](#_Toc501681066)

[Bảng 8 Cấu trúc dữ liệu bảng “Pay Bill Rate” 58](#_Toc501681067)

[Bảng 9 Cấu trúc dữ liệu bảng “Invoice Interpreter’ 58](#_Toc501681068)

[Bảng 10 Cấu trúc dữ liệu bảng “Invoice process Interpreter” 59](#_Toc501681069)

[Bảng 11 Cấu trúc dữ liệu bảng “Invoice Customer” 59](#_Toc501681070)

[Bảng 12 Cấu trúc dữ liệu bảng “Language” 59](#_Toc501681071)

[Bảng 13 Cấu trúc dữ liệu bảng “Permission” 60](#_Toc501681072)

[Bảng 14 Cấu trúc dữ liệu bảng “Authority” 60](#_Toc501681073)

# Danh Sách Hình Vẽ

[Hình 1 Kiến trúc microservice 17](#_Toc501681150)

[Hình 2 Mô hình triển khai hệ thống ở mức cao 17](#_Toc501681151)

[Hình 3 Kiến trúc của Angular project 19](#_Toc501681152)

[Hình 4 Cấu trúc của ứng dụng Spring Boot 20](#_Toc501681153)

[Hình 5 Biểu đồ use case Tổng quan 26](#_Toc501681154)

[Hình 6 Use case phân rã của “Quản lý lịch hẹn” 27](#_Toc501681155)

[Hình 7 Use case phân rã cho “Quản lý user” 28](#_Toc501681156)

[Hình 8 Use case phân rã cho “Quản lý hóa đơn” 28](#_Toc501681157)

[Hình 9 Biểu đồ hoạt động use case “Admin tạo lịch hẹn cho requester” 31](#_Toc501681158)

[Hình 10 Biểu đồ hoạt động use case “Requester tạo lịch hẹn” 33](#_Toc501681159)

[Hình 11 Biểu đồ hoạt động use case “Admin kết thúc lịch hẹn” 34](#_Toc501681160)

[Hình 12 Biểu đồ hoạt động use case “Interpreter kế thúc lịch hẹn” 36](#_Toc501681161)

[Hình 13 Biểu đồ hoạt động use case “Requester hủy lịch hẹn” 37](#_Toc501681162)

[Hình 14 Biểu đồ hoạt động use case “Interpreter hủy lịch hẹn” 39](#_Toc501681163)

[Hình 15 Biểu đồ hoạt động use case “Tạo mới Customer” 40](#_Toc501681164)

[Hình 16 Biểu đồ hoạt động use case “Tạo mới Department cho một Customer” 42](#_Toc501681165)

[Hình 17 Biểu đồ hoạt động use case “Tạo mới Requester” 44](#_Toc501681166)

[Hình 18 Biểu đồ hoạt động use case “Tạo mới Interpreter” 45](#_Toc501681167)

[Hình 19 Kiến trúc triển khai của hệ thống 47](#_Toc501681168)

[Hình 20 Mô hình các bảng cơ sở dữ liệu thu gọn 48](#_Toc501681169)

[Hình 21 Mô hình bảng Cơ sở dữ liệu rút gọn (tiếp) 48](#_Toc501681170)

[Hình 22 Các bản ghi của bảng “Authority” 60](#_Toc501681171)

[Hình 23 Giao diện thêm mới Customer 61](#_Toc501681172)

[Hình 24 Giao diện chọn Customer để thêm Department 62](#_Toc501681173)

[Hình 25 Giao diện thêm mới Department 63](#_Toc501681174)

[Hình 26 Giao diện chọn Customer và Department để thêm Requester 64](#_Toc501681175)

[Hình 27 Giao diện cho thêm mới Requester 65](#_Toc501681176)

[Hình 28 Giao diện thêm mới Interpreter 67](#_Toc501681177)

[Hình 29 Giao diện thêm mới Bill Rate 68](#_Toc501681178)

[Hình 30 Giao diện tạo lịch hẹn thoại 69](#_Toc501681179)

[Hình 31 Giao diện tạo yêu cầu dịch tài liệu 70](#_Toc501681180)

[Hình 32 Giao diện chung cho quản lý lịch hẹn 71](#_Toc501681181)

[Hình 33 Giao diện cho Interpreter hủy lịch hẹn 71](#_Toc501681182)

[Hình 34 Giao diện cho Requester hủy lịch hẹn 72](#_Toc501681183)

[Hình 35 Giao diện cho kết thúc lịch hẹn 72](#_Toc501681184)

[Hình 36 Giao diện thông tin hóa đơn của lịch hẹn 73](#_Toc501681185)

[Hình 37 Giao diện cho Interpreter thực hiện một yêu cầu dịch tài liệu 73](#_Toc501681186)

[Hình 38 Giao diện kết thúc lịch hẹn của Interpreter 74](#_Toc501681187)

[Hình 39 Giao diện sau đăng nhập của Admin 77](#_Toc501681188)

# Danh Sách Thuật Ngữ Sử Dụng

|  |  |  |
| --- | --- | --- |
| STT | Thuật ngữ | Ý nghĩa |
| 1 | Customer | Chỉ khách hàng của Công ty sử dụng hệ thống này. Khách hàng có thể là cá nhân hoặc tổ chức/công ty/doanh nghiệp. Hệ thống sẽ lưu lại thông tin của Customer |
| 2 | Department | Khi khách hàng không phải là cá nhân, mà là công ty hay tổ chức. Hệ thống quản lý thông tin các phòng ban của khách hàng. Department chỉ các phòng ban của khách hàng. |
| 3 | Sub-department | Sub-department chỉ phòng ban của phòng ban.  Một ví dụ: Khách hàng là công ty NCCSoft. Công ty có các phòng: phòng dev, phòng test, phòng nhân sự. Trong phòng dev có đội android, đội .NET, đội java. Vậy trong hệ thống sẽ có thông tin một Customer là công ty NCCSoft, ba Department của Customer đó là phòng dev, phòng test, phòng nhân sự, và Department là phòng dev có ba Sub-department là đội Android, đội .NET và đội Java |
| 4 | Super-requester | Chỉ các đại diện cho Customer (là tổ chức/doanh nghiệp) để đặt các lịch hẹn, yêu cầu dịch vụ trên hệ thống. Một Customer có thể có nhiều Super-requester. Super-requester có quyền xem các lịch hẹn mà requester của nó đã đặt, cũng như đặt lịch hẹn cho requester của nó. |
| 5 | Requester | Chỉ các đại diện cho phòng ban của Customer khi Customer là tổ chức/công ty. Khi Customer là cá nhân thì Requester cũng chính là Customer. Một phòng ban có thể có nhiều requester. |
| 6 | Interpreter | Chỉ các người phiên dịch viên làm việc cho hệ thống. Họ có thể là nhân viên của công ty, cộng tác viên của công ty, một công ty cũng lĩnh vực phiên dịch khác, v.v. |
| 7 | Invoice | Chỉ hóa đơn. Hóa đơn tiền khách hàng phải trả cho công ty. Hóa đơn tiền công ty phải trả cho các phiên dịch viên. |
| 8 | Pay rate | Chỉ một mức tiền phải trả cho phiên dịch viên. Có nhiều kiểu Pay rate, theo loại ngôn ngữ, theo loại dịch vụ là dịch tài liệu, dịch họp thoại hay phiên dịch, theo người phiên dịch, v.v. |
| 9 | Bill rate | Chỉ một mức tiền Khách hàng phải trả cho công ty. Có nhiều kiểu Bill rate, tùy thuộc vào kiểu dịch vụ, kiểu khách hàng |
| 10 | Intake | Chỉ một yêu cầu dịch vụ từ phía khách hàng. Ví dụ một khách hàng yêu cầu dịch một tài liệu Tiếng Anh sang Tiếng Nhật, trước 9h00 ngày 22/12/2017 phải xong. Thì yêu cầu đó của khách hàng gọi là một intake. |
| 11 | Assigned | Chỉ một trang thái của dịch vụ. Trạng thái này là khi dịch vụ đã có một phiên dịch viên nhận thực hiện. |
| 12 | Uncover | Chỉ một trạng thái của dịch vụ. Trạng thái này là khi chưa có phiên dịch viên nhận dịch vụ |
| 13 | Cancelled | Chỉ một trạng thái của dịch vụ. Trạng thái này là khi khách hàng hủy dịch vụ. |
| 14 | Lịch hẹn | Cũng giống như intake. Lịch hẹn chỉ một yêu cầu dịch vụ của khách hàng. |
| 15 | Tính tiền | Chỉ hành động tính toán hóa đơn cho khách hàng và phiên dịch viên khi dịch vụ kết thúc. |
| 16 | Lịch hẹn kiểu on-site | Chỉ kiểu lịch hẹn mà khách hàng muốn phiên dịch viên đến trực tiếp đến một địa điểm để thực hiện công việc phiên dịch. |
| 17 | Lịch hẹn kiểu thoại | Chỉ kiểu dịch vụ là dịch hội thoại là video, video call, thư thoại, v.v. |
| 18 | Lịch hẹn kiểu translation | Chỉ yêu cầu dịch tài liệu, văn bản, v.v. |

# Danh Sách Từ Viết Tắt Sử Dụng

|  |  |  |
| --- | --- | --- |
| STT | Từ viết tắt | Ý nghĩa |
| 1 | CRUD | Create, Read, Update, và Delete là bốn chức năng cơ bản của lưu trữ dữ liệu |
| 2 | API | Application Programming Interface |
| 3 | UI | User Interface |
| 4 | ĐATN | Đồ Án Tốt Nghiệp |

# Tóm Tắt Nội Dung Đồ Án Tốt Nghiệp

Ngày nay với sự phát triển nhanh chóng về kinh tế, các lĩnh vực kinh doanh cũng có sự phát triển vượt bậc. Thời trang vốn từ lâu đã là một trong những thành phần chủ đạo trong các nền kinh tế cũng không nằm trong vùng ngoại lệ. Ngoài những công ty, những cửa hàng lớn đã có từ trước, các đơn vị kinh doanh nhỏ lẻ, kinh doanh online cũng ngày một phát triển, nâng tầm về quy mô và số lượng. Thị trường kinh doanh thời trang vốn đã sôi động nay càng trở thành mảnh đất giàu tiềm năng cho các nhà đầu tư hơn.

Khi mở rộng quy mô kinh doanh, không chỉ các công ty, cửa hàng lớn mà ngay cả những hộ tiểu thương, kinh doanh online cũng đã và đang gặp những khó khăn trong việc quản lí, nắm bắt thông tin về tất cả sản phẩm trong cửa hàng của mình. Các chủ cửa hiệu cũng mất rất nhiều thời gian quản lí, cập nhật tình hình kinh doanh của các cửa hàng.

Nắm bắt được các vướng mắc đó, em đã đề xuất thực hiện ứng dụng tư vấn và quản lí bán hàng nhằm giúp các nhân viên tư vấn sản phẩm một cách hiệu quả, các chủ cửa hiệu cập nhật tình hình kinh doanh nhanh chóng. Các nhân viên bán hàng có thể tra cứu thông tin sản phẩm qua màu sắc, tên gọi, mã vạch, các sản phẩm bán chạy, khuyến mại,... Chủ cửa hiệu có thể nắm bắt được tình hình kinh doanh, doanh thu của các cửa hàng, phân tích xu hướng từ những tìm kiếm lựa chọn của người dùng để đưa ra chiến lược kinh doanh và nhập hàng hợp lí.

Đồ án chia thành 6 chương:

* Chương 1: Đặt vấn đề và tổng quan về bài toán
* Chương 2: Kiến trúc tổng quan của hệ thống và công nghệ sử dụng
* Chương 3: Phân tích và thiết kế hệ thống
* Chương 4: Xây dựng và phát triển hệ thống
* Chương 5: Triển Khai, Cài Đặt và Kiểm Thử
* Chương 6: Tổng kết và hướng phát triển

# Lời Cảm Ơn

Em xin chân thành cảm ơn các thầy cô trong viện đã truyền đạt cho em những kiến thức và kinh nghiệm quý giá trong suốt 5 năm học qua để em có đủ kiến thức để hoàn thành đồ án này.

Em xin cảm ơn bộ môn Công nghệ Phần mềm, viện Công nghệ thông tin và Truyền thông đã tạo điều kiện để em được thực hiện đồ án này.

Em xin gửi lời cảm ơn sâu sắc đến thầy Cao Tuấn Dũng đã hướng dẫn, chỉ dạy chúng em tận tình trong suốt quá trình thực hiện đồ án này.

Cuối cùng, em xin được gửi lời cảm ơn chân thành đến gia đình bạn bè đã luôn tận tâm giúp đỡ, động viên, đóng góp ý kiến giúp em hoàn thành đồ án tốt nghiệp thành công.

# Chương 1 Đặt Vấn Đề Và Tổng Quan Về Bài Toán

*Nội dung chương này sẽ trình bày hai vấn đề chính sau đây. Thứ nhất, bối cảnh ra đời của bài toán, tầm quan trọng của việc giải quyết bài toán. Thứ hai, giải pháp cho bài toán, và các yêu cầu mà giải pháp đó phải giải quyết được.*

## Đặt Vấn Đề



### Bối cảnh Bài Toán

Từ xưa đến nay, thời trang hay buôn bán quần áo vẫn luôn chiếm một tỉ trọng lớn trong ngành công thương của mỗi quốc gia, mỗi thời kì. Quần áo luôn là một mặt hàng thiết yếu không thể thiếu.

Ngày nay khi đời sống xã hội con người được cải thiện, nhu cầu ăn uống ăn mặc cũng tăng cao. Không chỉ có thế, các yêu cầu về sản phẩm của người tiêu dùng cũng khắt khe hơn trước. Đáp ứng lại sự thay đổi đó, các công ty chuyên về sản xuất thời trang cũng luôn luôn đưa ra thị trường rất nhiều sản phẩm đa dạng cả về mẫu mã và giá cả. Thị trường ngày nay không còn là sân chơi của những công ty kinh doanh lớn hay những cửa hàng có tiếng trước đó nữa mà còn có sự góp mặt của rất nhiều hộ kinh doanh nhỏ lẻ, kinh doanh online. Các cửa hàng chi nhánh thời trang cũng đua nhau mọc lên phủ kín những con phố.

Số lượng cửa hàng, sản phẩm tăng lên kéo theo bài toán về nắm bắt và quản lí thông tin. Việc nắm bắt tình trạng của tất cả các mặt hàng gần như là không thể và là vấn đề lớn đối với các nhân viên của cửa hàng.

### Mục đích phát triển đề tài

Với sự phát triển mạnh mẽ của công nghệ thông tin và truyền thông, sự bùng nổ của các thiết bị di động, cùng với bài toán đặt ra như trên, chúng ta nghĩ ngay đến xây dựng một ứng dụng chạy trên nền tảng mobile.

Ứng dụng này phải đảm bảo hai tiêu chí chính. Thứ nhất là giải quyết được hết vấn đề của bài toán trên. Thứ hai là phải giúp giảm chi phí cho dịch vụ, mang lại lợi ích cho cả công ty và khách hàng của họ. Mặt khác, chi phí cho xây dựng ứng dụng, cũng như triển khai, bảo trì phải thấp.

## Tổng quan về bài toán

Nắm bắt những khó khăn bất cập của thị trường, công ty MISA đã đưa ra một giải pháp tổng thể cho việc quản lí các cửa hàng thời trang trong và ngoài nước. Một hệ thống mang tên Mshopkeeper giúp quản lí từ việc nhập hàng, bán hàng, tra cứu đến quản lí nhân viên, doanh thu. Ứng dụng tra cứu thông tin và quản lí cửa hàng Mshopkeeper là một phần trong toàn bộ hệ thống đó.

Ứng dụng này phải giải quyết được các vấn đề nghiệp vụ sau:

* Giúp nhân viên tra cứu thông tin sản phẩm theo màu sắc, size, tên gọi, mã vạch.
* Tra cứu thông tin về các loại sản phẩm bán chạy, sản phẩm khuyến mại hay sản phẩm mới nhập về.
* Tra cứu chính xác địa chỉ sản phẩm theo từng chi nhánh cửa hàng nếu sản phẩm đó hiện không có có chi nhánh hiện tại.
* Quản lí doanh thu của từng cửa hàng trong chuỗi, các biểu đồ về tình hình thu chi trong cửa hàng.
* Biểu đồ phân tích xu hướng tìm kiếm của người dùng đối với mặt hàng trong cửa hàng.

**KẾT CHƯƠNG:**

*Trong**chương này, người viết đồ án đã trình bày tổng quan nhu cầu ra đời và quản lý của các cửa hàng thời trang. Đồng thời đưa ra giải pháp cho vấn đề trên, và các yêu cầu mà giải pháp phải đáp ứng.*

*Ở chương tiếp theo, người viết đồ án sẽ nêu lý thuyết liên quan đến giải quyết bài toán, cũng như đưa ra các công nghệ và thuật toán sử dụng.*

# Chương 2 Tổng Quan Về Ứng Dụng Và Công Nghệ Sử Dụng

*Nội dung chương này sẽ trình bày 2 vấn đề chính sau đây. Thứ nhất, tổng quan về ứng dụng. Thứ hai là trình bày kiến thức mà người thực hiện đồ án đã tìm hiểu được từ các công nghệ sử dụng trong ứng dụng.*

## Tổng quan về hệ thống

Nhà phá triển quyết định chọn hệ quản trị cơ sở dữ liệu MySQL để quản lý cơ sở dữ liệu. Bởi MySQL là hệ quản trị cơ sở dữ liệu nguồn mở. MySQL là cơ sở dữ liệu truy cập tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành, cung cấp một hệ thống lớn các hàm tiện ích rất mạnh. Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng có truy cập cơ sở dữ liệu trên internet.

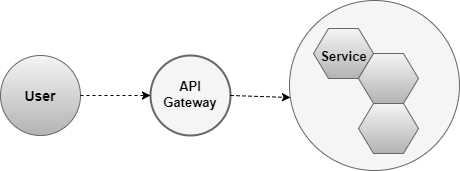
Với hy vọng hệ thống sẽ phát triển mạnh mẽ, có khả năng tích hợp mở rộng, dễ bảo trì, bảo mật tốt và khả năng chịu lỗi cao. Em quyết đinh chọn kiến trúc microservice. Với kiến trúc microservice, phía client ứng dụng chọn nền tảng IOS để triển khai. Phía server hệ thống chọn sử dụng nodejs để phát triển. Client và server sẽ giao tiếp với nhau qua API Gateway.

## Công nghệ và thư viện sử dụng

### 2.1 Microservice

Microservice là một mô hình hóa và thực thi hóa cho mô-hình-hướng-dịch-vụ[[1]](#footnote-1), được sử dụng để xây dựng các hệ thống phần mềm linh hoạt và dễ triển khai.

Hình 1 là cấu trúc đơn giản nhất của một kiến trúc microservice. Tùy vào từng hệ thống, hướng phát triển mà Kiến trúc microservice có ít nhiều thay đổi. Nhưng nguyên lý hoạt động thì gần như không thay đổi.



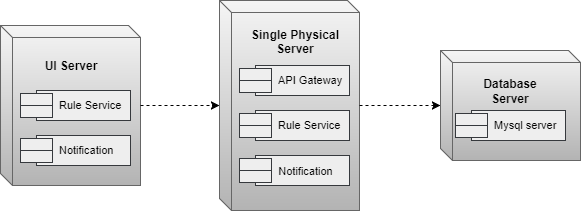
Hình 1 Kiến trúc microservice

Hệ thống được chia nhỏ thành các Service, mỗi service có thể giao tiếp với một Cơ sở dữ liệu khác nhau. Người dùng giao tiếp với các service thông qua API Gateway. Khi bất cứ một service nào ngừng hoạt động do một nguyên nhân nào đó thì các service khác có thể hoạt động mà không bị ảnh hưởng.

Ưu điểm của kiến trúc microservice:

* Hiệu năng tốt
* Tái sử dụng được
* Dễ phát triển
* Dễ bảo trì
* Bảo mật tốt
* Khả năng chịu lỗi
* Khả năng tích hợp, mở rộng

Hình 2 dưới đây là một ví dụ cho mô hình triển khai hệ thống theo kiến trúc microservice.



Hình 2 Mô hình triển khai hệ thống ở mức cao

API Gateway: là một component[[2]](#footnote-2) của mô hình Microservice. *API Gateway* là “lối vào duy nhất” cho tất cả các client. *API Gateway* xử lý các yêu cầu (request) bằng một trong hai cách, với các yêu cầu đơn giản chỉ là được định tuyến hoặc được ủy quyền cho service nhất định thì nó sẽ gửi đến các dịch vụ (service) phù hợp. Với các request khác nó xử lý bằng cách đưa ra nhiều dịch vụ (multiple services).

Notification: là một component của hệ thống. Notification là một microservice, nhiệm vụ của nó là gửi các thông báo như SMS, EMAIL, PUSH NOTIFICATION, v.v. Notification cũng quản lý mẫu cho nội dung của thông báo.

Registry: Registry là một "dịch vụ đăng ký"[[3]](#footnote-3), là một component của kiến trúc microservice, nó có thể được coi là cơ sở dữ liệu của các dịch vụ khác. Các cá thể dịch vụ (Service instances) được đăng ký các quyền, chức năng bởi service registry khi hệ thống khởi động, và được dỡ đăng ký khi hệ thống bị tắt. Khách hàng (client) của một Service nào đó truy vấn service registry để tìm kiếm đối tượng có sẵn của dịch vụ. Service registry có thể gọi một API để xác nhận xem một service có thể xử lý các yêu cầu không.

### 2.2 Nodejs

Tổng quan về Nodejs:   
NodeJS là một mã nguồn được xây dựng dựa trên nền tảng Javascript V8 Engine, nó được sử dụng để xây dựng các ứng dụng web như các trang video clip, các forum và đặc biệt là trang mạng xã hội phạm vi hẹp. NodeJS là một mã nguồn mở được sử dụng rộng bởi hàng ngàn lập trình viên trên toàn thế giới. NodeJS có thể chạy trên nhiều nền tảng hệ điều hành khác nhau từ WIndow cho tới Linux, OS X nên đó cũng là một lợi thế. NodeJS cung cấp các thư viện phong phú ở dạng Javascript Module khác nhau giúp đơn giản hóa việc lập trình và giảm thời gian ở mức thấp nhất.

Đặc điểm và cách thức hoạt động của Nodejs:

* Tính không đồng bộ:

### 2.3 Express framework

Express là một module hay còn được nhắc đến là một framework trong nodejs. Express framework cung cấp cho chúng ta đầy đủ tiện ích để có thể xây dựng một ứng dụng web hay một API

### 2.4 Json web token (JWT)

JSON Web Token (JWT) là 1 tiêu chuẩn mở ([RFC 7519](https://tools.ietf.org/html/rfc7519)) định nghĩa cách thức truyền tin an toàn giữa các thành viên bằng 1 đối tượng JSON. Thông tin này có thể được xác thực và đánh dấu tin cậy nhờ vào "chữ ký" của nó. Phần chữ ký của JWT sẽ được mã hóa lại bằng [**HMAC**](https://en.wikipedia.org/wiki/Hash-based_message_authentication_code) hoặc [**RSA**](https://en.wikipedia.org/wiki/RSA_(cryptosystem))

### 2.5 MySQL

MySQL[[4]](#footnote-4) là hệ quản trị cơ sở dữ liệu SQL nguồn mở phổ biến nhất, được phát triển, phân phối và hỗ trợ bởi Oracle Corporation.

**MySQL là một hệ thống quản lý cơ sở dữ liệu.**

Cơ sở dữ liệu là tập hợp có cấu trúc của dữ liệu. Nó có thể là bất cứ thứ gì từ một danh sách mua sắm đơn giản đến một thư viện hình ảnh hoặc một số lượng lớn thông tin trong một mạng lưới công ty. Để thêm, truy cập và xử lý dữ liệu được lưu trữ trong cơ sở dữ liệu, bạn cần một hệ thống quản lý cơ sở dữ liệu như MySQL Server. Vì các máy tính rất giỏi xử lý lượng lớn dữ liệu, các hệ thống quản lý cơ sở dữ liệu đóng vai trò trung tâm trong việc tính toán, như các tiện ích độc lập, hoặc như các phần của các ứng dụng khác.

**MySQL là Cơ sở dữ liệu quan hệ.**

Một cơ sở dữ liệu quan hệ lưu trữ dữ liệu trong các bảng riêng biệt thay vì đặt tất cả dữ liệu vào một kho lưu trữ lớn. Các cấu trúc cơ sở dữ liệu được tổ chức thành các tệp vật lý được tối ưu hóa cho tốc độ. Mô hình lôgic, với các đối tượng như cơ sở dữ liệu, bảng biểu, khung nhìn, hàng và cột, cung cấp môi trường lập trình linh hoạt. Bạn thiết lập các quy tắc điều chỉnh mối quan hệ giữa các trường dữ liệu khác nhau, chẳng hạn như một-một, một-nhiều, duy nhất, bắt buộc hoặc tùy chọn, và “các con trỏ” giữa các bảng khác nhau. Cơ sở dữ liệu thực thi các quy tắc này, do đó, với một cơ sở dữ liệu được thiết kế tốt, ứng dụng của bạn không bao giờ thấy dữ liệu không nhất quán, trùng lặp, đơn độc, đã lỗi thời hoặc thiếu.

Phần SQL của “MySQL” là viết tắt của “Structured Query Language” . SQL là ngôn ngữ tiêu chuẩn phổ biến nhất được sử dụng để truy cập cơ sở dữ liệu. Tùy thuộc vào môi trường lập trình của bạn, bạn có thể nhập SQL trực tiếp (ví dụ: để tạo báo cáo), nhúng các câu lệnh SQL vào mã được viết bằng ngôn ngữ khác hoặc sử dụng API cụ thể cho ngôn ngữ để ẩn cú pháp SQL.

**Phần mềm MySQL là mã nguồn mở.**

Mã nguồn mở có nghĩa là mọi người có thể sử dụng và sửa đổi phần mềm. Bất kỳ ai cũng có thể tải về phần mềm MySQL từ Internet và sử dụng nó mà không phải trả bất cứ điều gì. Nếu bạn muốn, bạn có thể nghiên cứu mã nguồn và thay đổi nó cho phù hợp với nhu cầu của bạn. Phần mềm MySQL sử dụng GPL (GNU General Public License), để xác định những gì bạn có thể và không thể làm với phần mềm trong các tình huống khác nhau. Nếu bạn cảm thấy không thoải mái với GPL hoặc cần phải nhúng mã MySQL vào một ứng dụng thương mại, bạn có thể mua một phiên bản thương mại được cấp phép.

Máy chủ cơ sở dữ liệu MySQL rất nhanh, đáng tin cậy, khả năng mở rộng và dễ sử dụng.

MySQL Server có thể chạy thoải mái trên máy tính để bàn hoặc máy tính xách tay, cùng với các ứng dụng khác của bạn, máy chủ web, v.v ... cần ít hoặc không có sự chú ý. Nếu bạn dành toàn bộ một máy cho MySQL, bạn có thể điều chỉnh các thiết lập để tận dụng tất cả bộ nhớ, sức mạnh của CPU và dung lượng I/O. MySQL cũng có thể quy mô lên đến các cụm máy móc, kết nối với nhau.

MySQL Server ban đầu được phát triển để xử lý cơ sở dữ liệu lớn nhanh hơn nhiều so với các giải pháp hiện tại và đã được sử dụng thành công trong các môi trường sản xuất đòi hỏi khắt khe trong nhiều năm. Mặc dù dưới sự phát triển liên tục, MySQL Server hôm nay cung cấp một bộ các chức năng phong phú và hữu ích. Khả năng kết nối, tốc độ và bảo mật của nó làm cho MySQL Server rất phù hợp với việc truy cập cơ sở dữ liệu trên Internet.

**MySQL Server làm việc trong máy khách/máy chủ hoặc các hệ thống nhúng.**

Phần mềm Cơ sở dữ liệu MySQL là một hệ thống máy khách/máy chủ bao gồm một máy chủ SQL đa luồng hỗ trợ các kết thúc ngược lại, một số chương trình khách hàng khác nhau và thư viện, các công cụ hành chính và một loạt các giao diện lập trình ứng dụng (API).

Chúng tôi cũng cung cấp MySQL Server như một thư viện đa luồng nhúng mà bạn có thể liên kết với ứng dụng của bạn để có được sản phẩm độc lập nhỏ hơn, nhanh hơn và dễ quản lý hơn.

MySQL Server có một bộ tính năng thiết thực được phát triển với sự hợp tác chặt chẽ với người dùng của chúng tôi. Rất có thể ứng dụng hoặc ngôn ngữ yêu thích của bạn hỗ trợ máy chủ cơ sở dữ liệu MySQL.

**KẾT CHƯƠNG:**

*Kết thúc chương này, người viết đồ án đã trình bày các công nghệ sử dụng và lý do sử dụng các công nghệ đó.*

*Ở chương tiếp theo, người viết sẽ trình bày phân tích và thiết kế hệ thống.*

# Chương 3 Phân Tích Và Thiết Kế Hệ Thống

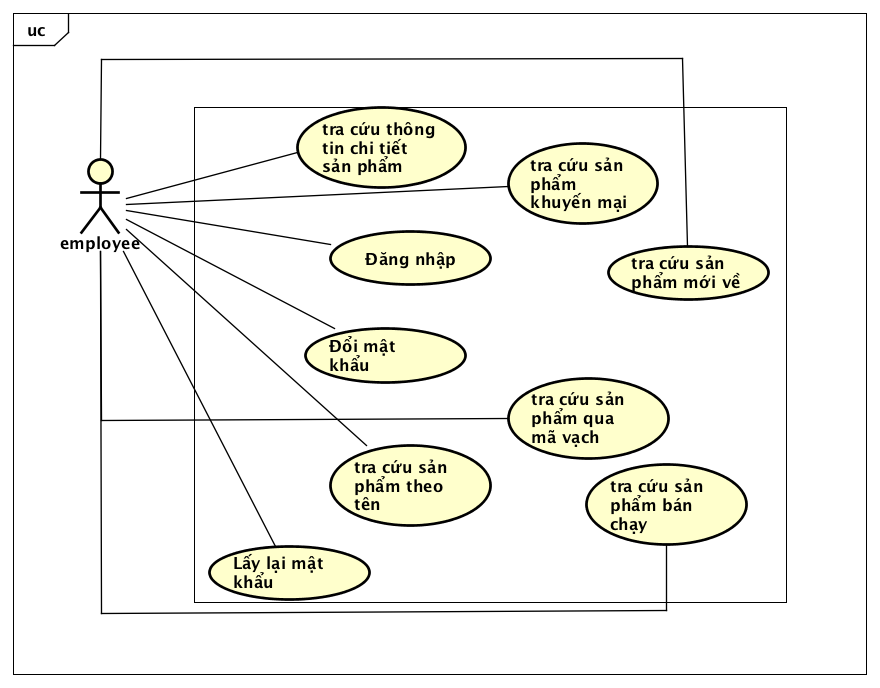
*Chương này sẽ trình bày phân tích và thiết kế hệ thống. Cụ thể là, các tác nhân của hệ thống sẽ sử dụng các chức năng của hệ thống như thế nào. Các use case sẽ được đặc tả bằng lời và biểu đồ hoạt động.*

## Các tác nhân của hệ thống

Tác nhân duy nhất sử dụng ứng dụng là nhân viên tư vấn bán hàng.

## Phân tích hệ thống về mặt chức năng

### 2.1 Biểu đồ use case Tổng quan



Hình 5 Biểu đồ use case Tổng quan

### 2.2 Quy trình nghiệp vụ

Trong phần này, sẽ trình các quy trình nghiệp vụ chính: quy trình sử dụng phần mềm, quy trình quản lý cuộc hẹn, quy trình quản lý user, v.v.

Chi tiết về hành động trong các quy trình này được mô hình hoá trong các mục con của

từng quy trình.

2.3.1 Quy trình sử dụng phần mềm

Khách được tạo tài khoản từ admin hay người dùng khác có thẩm quyền tạo tài khoản. Sau khi tài khoản được tạo, hệ thống sẽ gửi mail cho người dùng vừa tạo. Email gồm tên đăng nhập, một đường link chứa token để người dùng thiết lập mật khẩu. Người dùng phải vào liên kết đó để thực hiện việc thiết lập lại mật khẩu.

Sau khi thiết lập mật khẩu, khách mới có thể đăng nhập vào hệ thống, bằngtên đăng nhập và mật khẩu đã thiết lập.

Sau khi đăng nhập thành công vào hệ thống, người dùng có thể sử dụng các chức năng được phân quyền. Cụ thể như sau:

Admin có toàn quyền sử dụng các chức năng của hệ thống:

* Tạo Pay rate, Bill rate, sửa thông tin Rate đã tạo, vô hiệu hóa/kích hoạt Rate.
* Tạo Customer, sửa thông tin Customer đã tạo, vô hiệu hóa/kích hoạt Customer.
* Tạo Department, Sub-department thuộc một Customer, sửa thông tin, vô hiệu hóa/kích hoạt Department/Sub-department
* Tạo Requester đại diện cho Customer để đặt lịch hẹn, sửa thông tin, vô hiệu hóa/kích hoạt Requester.
* Tạo Interpreter, sửa thông tin, vô hiệu hóa/kích hoạt Interpreter
* Quản lý thông tin người dùng
* Quản lý các lịch hẹn, bao gồm việc tạo lịch hẹn cho khách, sửa lịch hẹn, hủy lịch hẹn, xem lịch sử cập nhật cuộc hẹn, kết thúc cuộc hẹn
* Xem, xuất các loại báo cáo
* Xem, xuất hóa đơn của khách (customer) và nhân viên (interpreter)
* Thêm Ngôn ngữ, vô hiệu hóa/kích hoạt ngôn ngữ

Requester có quyền sử dụng các chức năng sau:

* Tạo lịch hẹn, sửa thông tin lịch hẹn, hủy lịch hẹn, xem lịch sử cập nhật lịch hẹn
* Xem một số loại báo cáo: báo cáo hóa đơn phải thanh toán cho công ty, báo cáo thống kê cuộc hẹn đã đặt theo ngôn ngữ.
* Đánh giá chất lượng phiên dịch

Super-requester có các quyền tương tự như Requester, bởi nó cũng chính là một Requester nhưng nó là đại diện của nhiều requester khác. Ngoài ra, nó còn có quyền thêm các phòng ban của chính công ty mà nó làm chủ.

Ở đây ta hiểu Super-requester chính là Customer (kiểu một công ty, tổ chức), Requester là Department, Sub-department (các phòng ban của công ty đó).

Interpreter có quyền sử dụng các chức năng sau:

* Xem thông tin lịch hẹn đã được giao, từ chối nhận lịch hẹn được giao
* Nhận một lịch hẹn có thể nhận.
* Kết thúc lịch hẹn
* Xem báo cáo các lịch hẹn đã thực hiện kèm số tiền được trả.

2.3.2 Quy trình Quản lý lịch hẹn

Tất cả các tác nhân đều có thể tham gia vào quy trình quản lý lịch hẹn

Các use case có trong quản lý lịch hẹn:

* Admin tạo lịch hẹn cho Requester
* Requester tạo lịch hẹn cho mình
* Sửa thông tin lịch hẹn
* Xem lịch sử cập nhật lịch hẹn
* Admin giao lịch hẹn cho một Interpreter
* Interpreter nhận một lịch hẹn
* Interpreter hủy lịch hẹn
* Requester hủy lịch hẹn
* Admin kết thúc lịch hẹn
* Interpreter kết thúc lịch hẹn

Người dùng có thể tạo lịch hẹn là admin, super-requester, requester.

Một lịch hẹn phải thuộc về một Requester.

Sau khi lịch hẹn được tạo, nếu lịch hẹn được chọn broadcast, thì nó sẽ xuất hiện trong my folder của tất cả các interpreter có thể thực hiện nó. Và interpreter có thể nhận lịch hẹn đó. Khi Interpreter nhận lịch hẹn, lịch hẹn đó sẽ vào trong my schedule của interpreter đó. Và lịch hẹn có trạng thái là “assigned”.

Hoặc admin cũng có quyền giao lịch hẹn cho một interpreter nhất định. Interpreter có thể nhận thực hiện cuộc hẹn đó hoặc từ chối. Nếu interpreter từ chối thì lịch hẹn đó chuyển sang trạng thái “uncover” là trạng thái chưa có ai nhận cuộc hẹn.

Nếu Requetser hủy lịch hẹn, trạng thái của lịch hẹn chuyển sang “cancelled”. Và trạng thái của lịch hẹn không thể thay đổi được nữa.

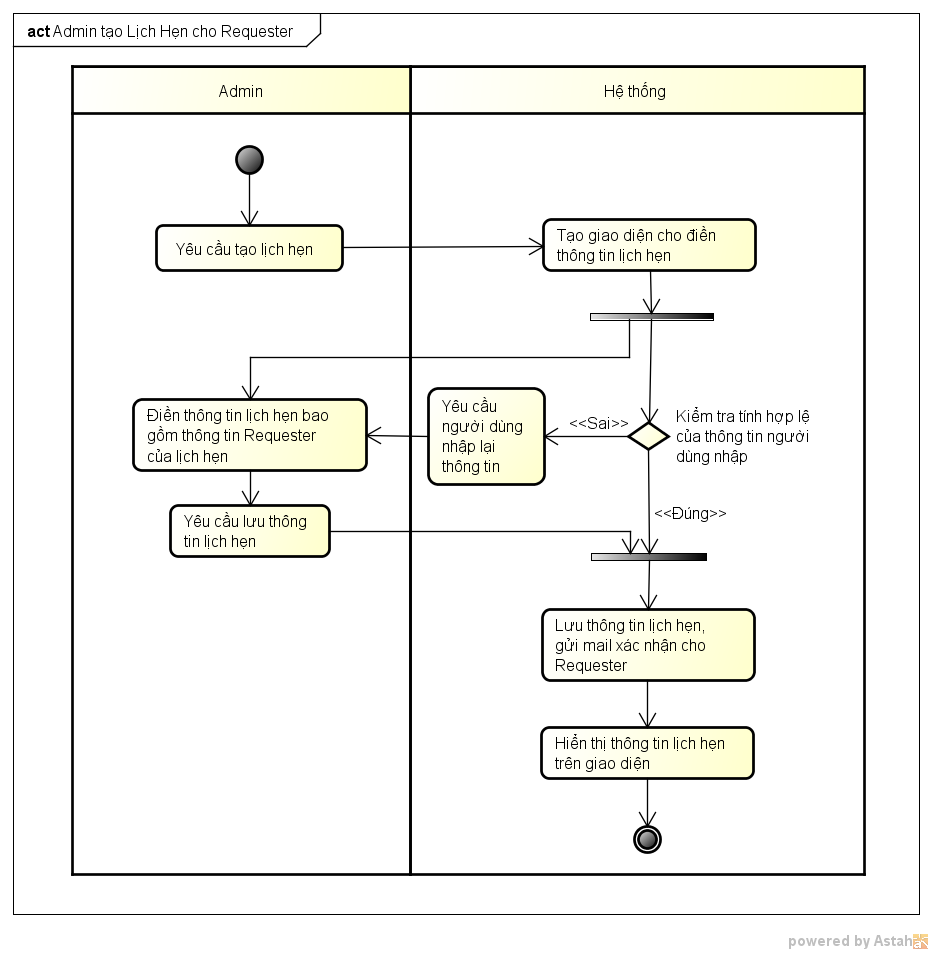
Nếu lịch hẹn đã được thực hiện bởi một interpreter, và nó đã có Bill rate, Pay rate phù hợp thì có thể tiến hành kết thúc lịch hẹn. Lịch hẹn kết thúc đồng thời hệ thống sẽ tính số tiền Customer phải trả và số tiền phải trả cho Interpreter (gọi tắt là Tính tiền).

Quá trình tính tiền kết thúc nghĩa là của Lịch hẹn cũng kết thúc.

Sau đây, người viết đồ án sẽ trình bày chi tiết hoạt động tạo lịch hẹn, kết thúc lịch hẹn, hủy lịch hẹn.

2.3.2.1 Chi tiết hoạt động của use case “Admin tạo lịch hẹn cho requester”

Chi tiết hoạt động của use case “Admin tạo lịch hẹn cho Requester” được mô hình hóa trong Hình 9, và mô tả chi tiết bên dưới.



Hình 9 Biểu đồ hoạt động use case “Admin tạo lịch hẹn cho requester”

Lịch hẹn do Admin tạo phải là được tạo cho một requester (đại diện cho một customer, hoặc phòng ban của customer).

Trình tự thực hiện :

* Tác nhân: chọn loại lịch hẹn muốn tạo, có các loại lịch hẹn đã được trình bày trong phần 2.2.3.
* Hệ thống: hiển thị giao diện cho người dùng tạo lịch hẹn
* Tác nhân: điền thông tin cho lịch hẹn, đối với mỗi loại lịch hẹn thông tin cần điền có thể khác nhau. Ví dụ lịch hẹn là Translation/transcription thì cần đính kèm file tài liệu muốn dịch. Các thông tin chung phải điền là thông tin của người tạo lịch hẹn, của requester, loại ngôn ngữ đích muốn dịch, địa điểm lịch hẹn, người trực tiếp nhận lịch hẹn, v.v.
* Hệ thống: trong quá trình người dùng nhập dữ liệu, hệ thống kiểm tra tính hợp lệ của dữ liệu nhập vào, ví dụ mail phải ở dạng [sample@abc.xyz](mailto:sample@abc.xyz). Nếu thông tin nhập vào không hợp lệ, hệ thống thông báo trên giao diện chi tiết lỗi sai của người nhập.
* Tác nhân: Sau khi điền xong thông tin, tác nhân yêu cầu lưu thông tin lịch hẹn.
* Hệ thống: kiểm tra các trường thông tin đã đúng, lưu lịch hẹn lại và gửi email xác nhận cho Requester, admin.
* Hệ thống: hiển thị thông tin lịch hẹn đã lưu trên giao diện

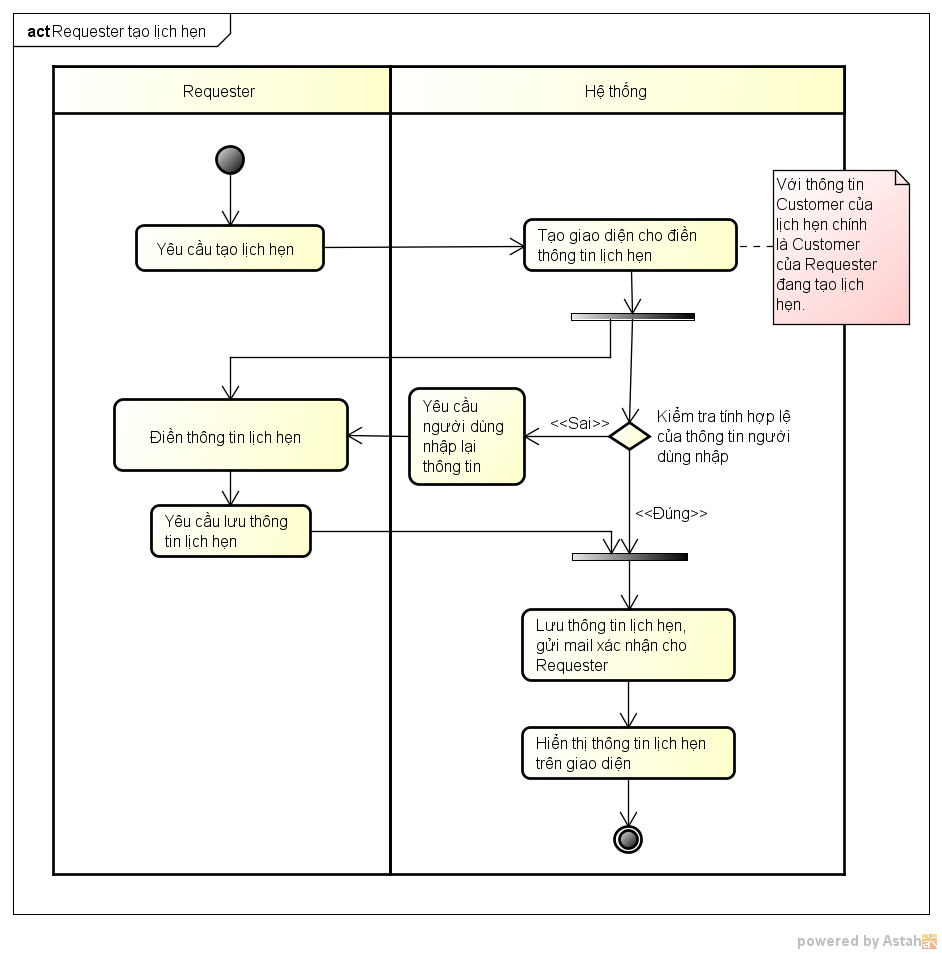
2.3.2.2 Chi tiết hoạt động của use case Requester tạo lịch hẹn

Chi tiết hoạt động của use case “Requester tạo lịch hẹn” được mô hình hóa trong Hình 10, và mô tả chi tiết bên dưới.

Khi Requester tạo lịch hẹn thì lịch hẹn đó là được tạo cho chính Requester đó.

Trình tự thực hiện :

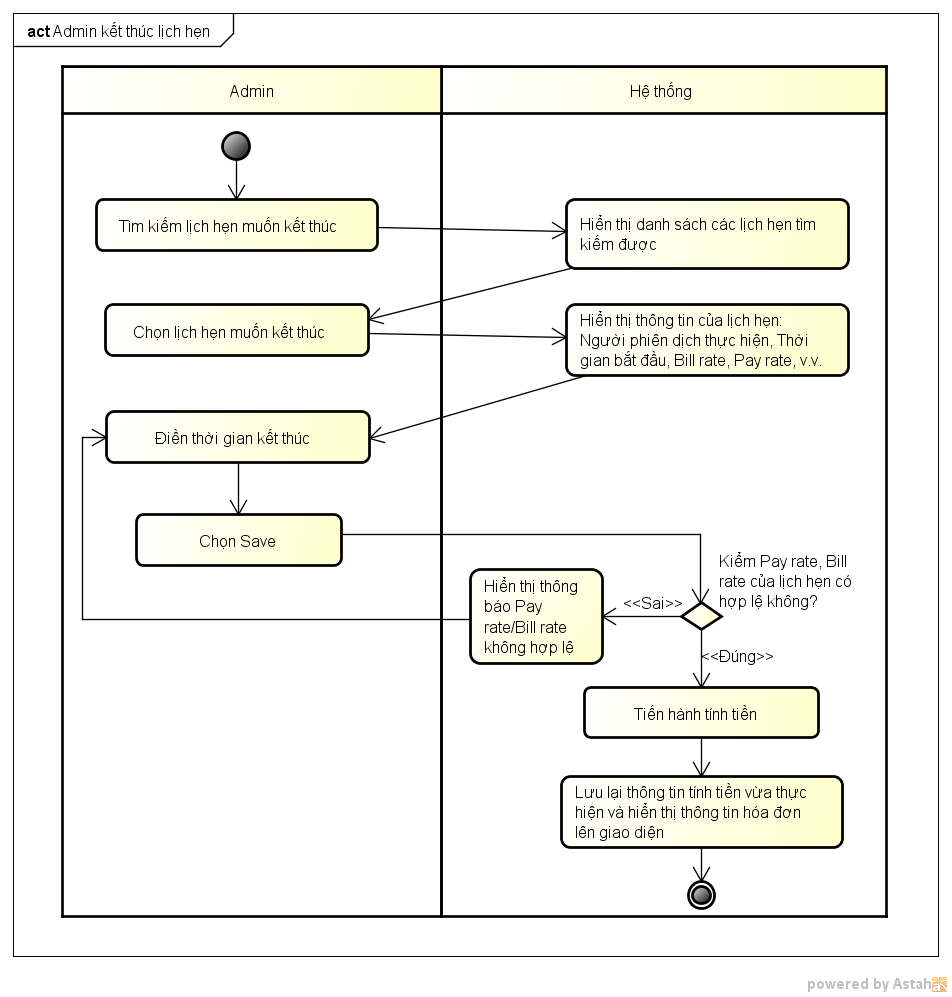
* Tác nhân: chọn loại lịch hẹn muốn tạo, có các loại lịch hẹn đã được trình bày trong phần 2.2.3.
* Hệ thống: hiển thị giao diện cho người dùng tạo lịch hẹn. Với thông tin customer là customer của Requester.
* Tác nhân: điền thông tin cho lịch hẹn, đối với mỗi loại lịch hẹn thông tin cần điền có thể khác nhau. Ví dụ lịch hẹn là Translation/transcription thì cần đính kèm file tài liệu muốn dịch. Các thông tin chung phải điền là thông tin của người tạo lịch hẹn, của requester, loại ngôn ngữ đích muốn dịch, địa điểm lịch hẹn, người trực tiếp nhận lịch hẹn, v.v.
* Hệ thống: trong quá trình người dùng nhập dữ liệu, hệ thống kiểm tra tính hợp lệ của dữ liệu nhập vào, ví dụ mail phải ở dạng [sample@abc.xyz](mailto:sample@abc.xyz). Nếu thông tin nhập vào không hợp lệ, hệ thống thông báo trên giao diện chi tiết lỗi sai của người nhập.
* Tác nhân: Sau khi điền xong thông tin, tác nhân yêu cầu lưu thông tin lịch hẹn.
* Hệ thống: kiểm tra các trường thông tin đã đúng, lưu lịch hẹn lại và gửi email xác nhận cho Requester, admin.
* Hệ thống: hiển thị thông tin lịch hẹn đã lưu trên giao diện



Hình 10 Biểu đồ hoạt động use case “Requester tạo lịch hẹn”

2.3.2.3 Chi tiết hoạt động của use case “Admin kết thúc lịch hẹn”

Chi tiết hoạt động của use case “Admin kết thúc lịch hẹn” được mô hình hóa trong Hình 11, và mô tả chi tiết bên dưới.



Hình 11 Biểu đồ hoạt động use case “Admin kết thúc lịch hẹn”

Tiền điều kiện để có thể kết thúc lịch hẹn là lịch hẹn đã được thực hiện bởi một Interpreter (nghĩa là status của nó là “assigned”), có Bill rate, Pay rate phù hợp cho lịch hẹn đó.

Trình tự thực hiện:

* Tác nhân: Chọn lịch hẹn muốn kết thúc
* Hệ thống: Hiển thị chi tiết lịch hẹn
* Tác nhân: Lựa chọn sửa thông tin lịch hẹn nếu cần thiết (ví dụ: thêm Bill rate, Pay rate cho lịch hẹn), và chọn lưu
* Hệ thống: Lưu lại thông tin lịch hẹn khi người dùng chọn sửa lịch hẹn
* Tác nhân: Điền thời gian kết thúc lịch hẹn, và chọn kết thúc lịch hẹn.
* Hệ thống: Kiểm tra các tiền điều kiện: đã được thực hiện bởi một Interpreter, đã có Bill rate, Pay rate hợp lệ. Nếu không thỏa mãn, hệ thống sẽ hiển thị thông báo lỗi chi tiết, và không tiến hành tính tiền. Nếu thỏa mãn, hệ thống tiến hành tính số tiền Customer phải trả, và số tiền phải trả cho Interpreter (gọi tắt quá trình này là tính tiền). Và lưu lại, hiển thị nên giao diện thong tin hóa đơn.

2.3.2.4 Chi tiết hoạt động của use case “Interpreter kết thúc lịch hẹn”

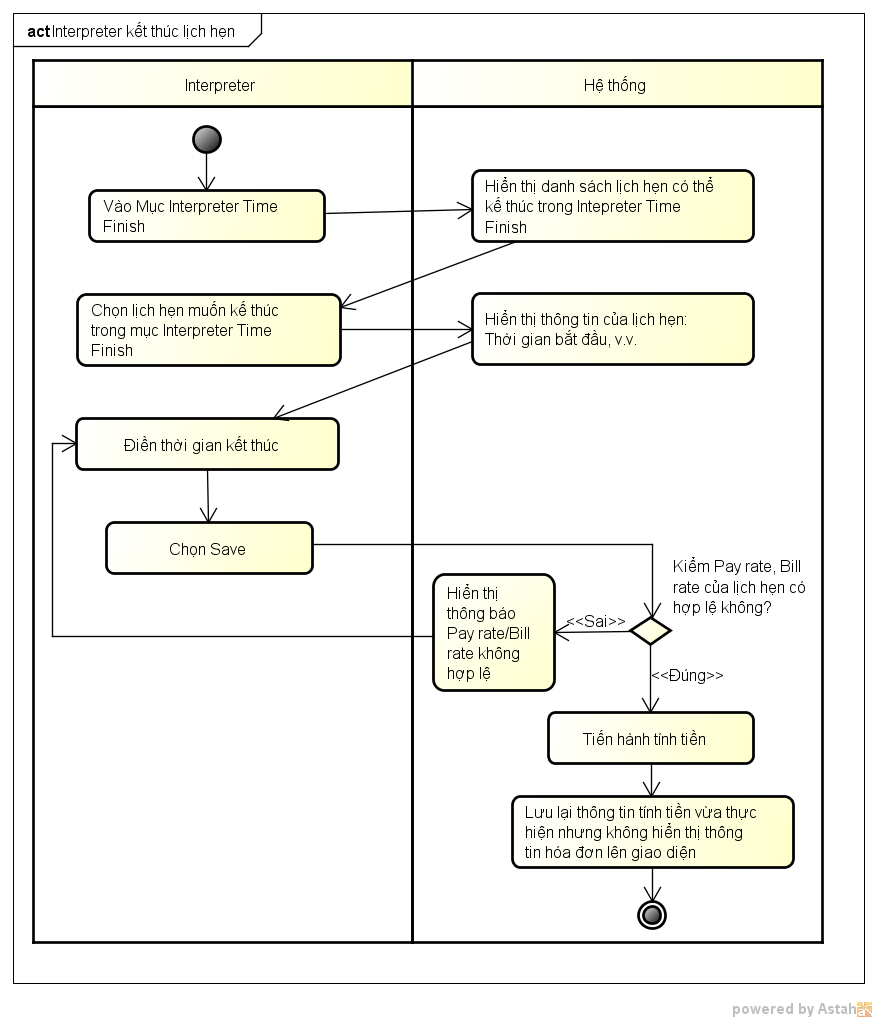
Chi tiết hoạt động của use case “Admin kết thúc lịch hẹn” được mô hình hóa trong Hình 12, và mô tả chi tiết bên dưới.

Tiền điều kiện để có thể kết thúc lịch hẹn là lịch hẹn đã được thực hiện bởi một Interpreter (nghĩa là status của nó là “assigned”), có Bill rate, Pay rate phù hợp cho lịch hẹn đó.

Trình tự thực hiện:

* Tác nhân: Chọn lịch hẹn muốn kết thúc trong Interpreter Time Finish
* Hệ thống: Hiển thị chi tiết lịch hẹn
* Tác nhân: Điền thời gian kết thúc lịch hẹn, và chọn kết thúc lịch hẹn.
* Hệ thống: Kiểm tra các tiền điều kiện: đã được thực hiện bởi một Interpreter, đã có Bill rate, Pay rate hợp lệ. Nếu không thỏa mãn, hệ thống sẽ hiển thị thông báo lỗi chi tiết, và không tiến hành tính tiền. Nếu thỏa mãn, hệ thống tiến hành tính số tiền Customer phải trả, và số tiền phải trả cho Interpreter (gọi tắt quá trình này là tính tiền). Và lưu lại, nhưng không hiển thị nên giao diện thong tin hóa đơn.

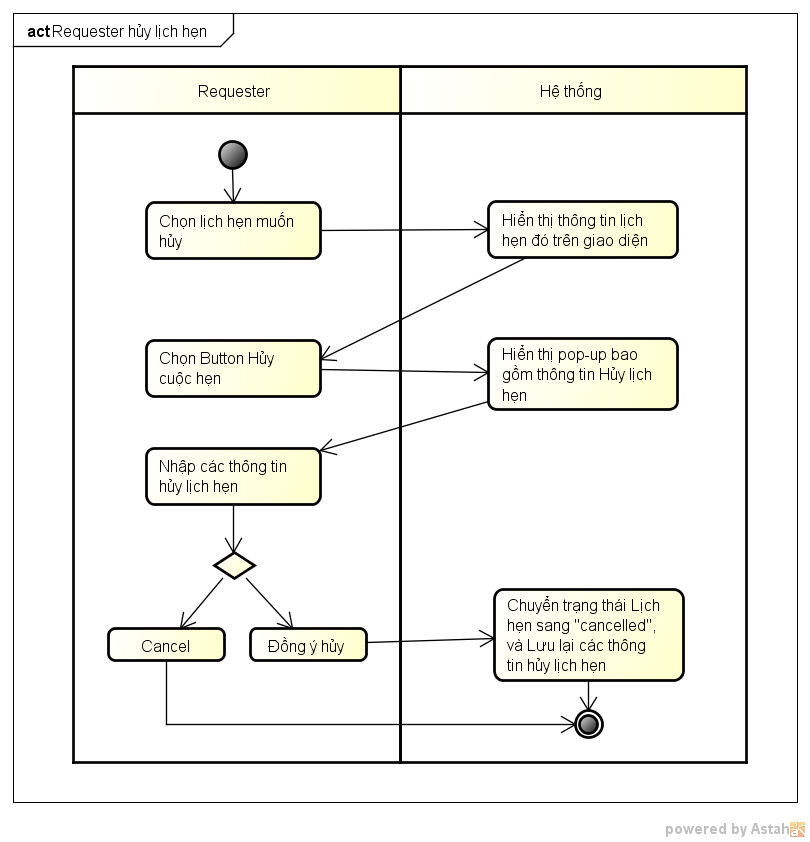
Sau khi kết thúc lịch hẹn thì lịch hẹn đó không xuất hiện trong mục “Interpreter time Finish” nữa.



Hình 12 Biểu đồ hoạt động use case “Interpreter kế thúc lịch hẹn”

2.3.2.5 Chi tiết hoạt động của use case “Requester lịch hẹn”

Chi tiết hoạt động của use case “Requester hủy lịch hẹn” được mô hình hóa trong Hình 13, và mô tả chi tiết bên dưới.



Hình 13 Biểu đồ hoạt động use case “Requester hủy lịch hẹn”

Khi Requester hủy lịch hẹn, trạng thái của lịch hẹn chuyển sang “cancelled” và lịch hẹn không tiếp tục thực hiện được (nghĩa là bị hủy bỏ).

Tiền điều kiện để một tác nhân có thể thực hiện use case này là: Requester/Super-requester phải là chủ nhân của lịch hẹn đó

Trình tự thực hiện:

Tác nhân: Chọn lịch hẹn muốn hủy

Hệ thống: Hiển thị thông tin lịch hẹn trên giao diện.

Tác nhân: Chọn button Hủy lịch hẹn.

Hệ thống: Hiển thị pop-up bao gồm thông tin cho Hủy lịch hẹn.

Tác nhân: Chọn nút “Hủy” để Hủy bỏ ý muốn hủy cuộc hẹn. Ngược lại, điền thông tin hủy lịch hẹn bao gồm: lý do hủy lịch hẹn, ngày hủy lịch hẹn, tên người hủy lịch hẹn và chọn nút “Save” để lưu lại thông tin hủy lịch hẹn.

Hệ thống: Nếu người dùng chọn hủy lịch hẹn, hệ thống thay đổi trạng thái của lịch hẹn sang “cancelled” và lưu lại thông tin hủy lịch hẹn.

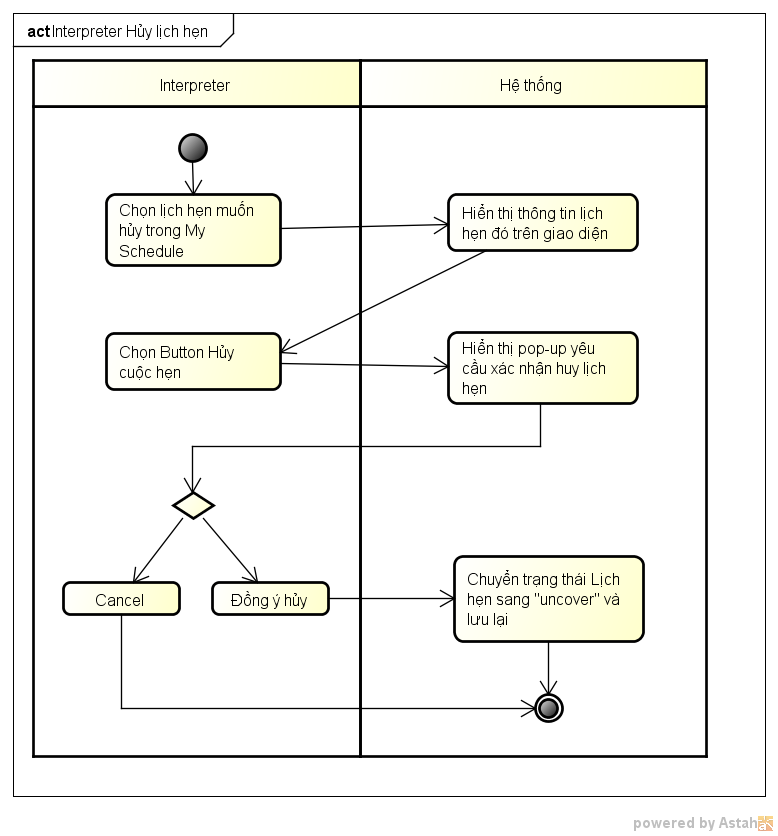
2.3.2.6 Chi tiết hoạt động của use case “Interpreter hủy lich hẹn”

Đối với kiểu Interpreter hủy lịch hẹn, Trạng thái của lịch hẹn chuyển sang “not cover” và lịch hẹn quay lại giai đoạn sau khi tạo lịch hẹn trong vòng đời của lịch hẹn đã trình bày ở trên.

Tiền điều kiện để Interpreter có thể hủy lịch hẹn là lịch hẹn đó được admin giao cho Interpreter.

Trình tự thực hiện:

* Tác nhân: Chọn lịch hẹn muốn hủy trong My schedule
* Hệ thống: Hiển thị thông tin lịch hẹn trên giao diện.
* Tác nhân: Chọn button Hủy lịch hẹn.
* Hệ thống: Hiển thị pop-up xác nhận hủy lịch hẹn
* Tác nhân: Chọn nút “Hủy” để Hủy bỏ ý muốn hủy cuộc hẹn. Ngược lại, chọn nút “Save” để hủy lịch hẹn.
* Hệ thống: Nếu người dùng chọn hủy lịch hẹn, hệ thống thay đổi trạng thái của lịch hẹn sang “uncover”.



Hình 14 Biểu đồ hoạt động use case “Interpreter hủy lịch hẹn”

2.3.3 Quy trình quản lý người dùng của hệ thống

Người dùng của hệ thống có bốn loại:

* Admin: có vai trò như quản trị viên của hệ thống.
* Interpreter: là phiên dịch viên. Có thể là nhân viên công ty, có thể là free-lancer, có thể là tình nguyện viên, hoặc một nhóm/tổ chức phiên dịch.
* Requester: đại diện cho một khách hàng của công ty. Một khách hàng có thể có nhiều requester.
* Super-requester: cũng chính là một requester. Nhưng có sự khác biệt như sau: giả sử Customer là một công ty có nhiều phòng ban, khi đó Super-requester là đại diện cho cả công ty đó, và có các requester đại diện cho các phòng ban. Khi đó Super-requester có thể đặt lịch hẹn nhìn thấy các lịch hẹn các phòng ban của mình.

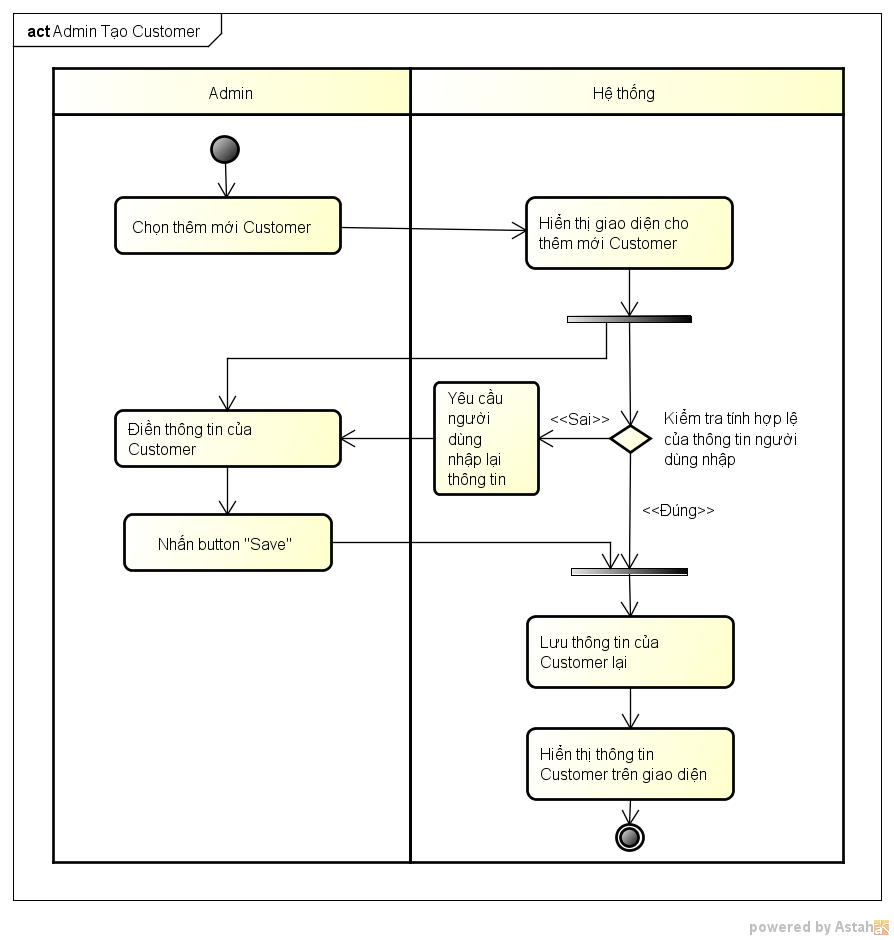
Chỉ admin có quyền quản lý người dùng của hệ thống.

Quản lý user bao gồm các use case:

* Thêm người dùng
* Sửa thông tin người dùng
* Vô hiệu hóa tài khoản người dùng
* Kích hoạt tài khoản người dùng

2.3.3.1 Chi tiết hoạt động use case “Admin tạo mới Customer”

Hình 15 mô hình hóa hoạt động của use case “Admin tạo mới Customer”. Chi tiết hoạt động sẽ được mô tả bên dưới.



Hình 15 Biểu đồ hoạt động use case “Tạo mới Customer”

Tác nhân tham gia vào use case là Admin.

Trình tự thực hiện:

* Tác nhân: Chọn thêm mới Customer
* Hệ thống: Hiển thị giao diện cho thêm mới Customer
* Tác nhân: Nhập vào các thông tin cần thiết của Customer
* Hệ thống: Kiểm tra tính hợp lệ của thông tin nhập vào
* Tác nhân: Nhấn button “Save” để lưu thông tin Customer
* Hệ thống: Nếu thông tin nhập vào hợp lệ. Hệ thống lưu mới một Customer. Nếu thông tin nhập vào không hợp lệ. Yêu cầu người dùng nhập lại.
* Hệ thống: Hiển thị thông tin Customer vừa lưu lên giao diện.

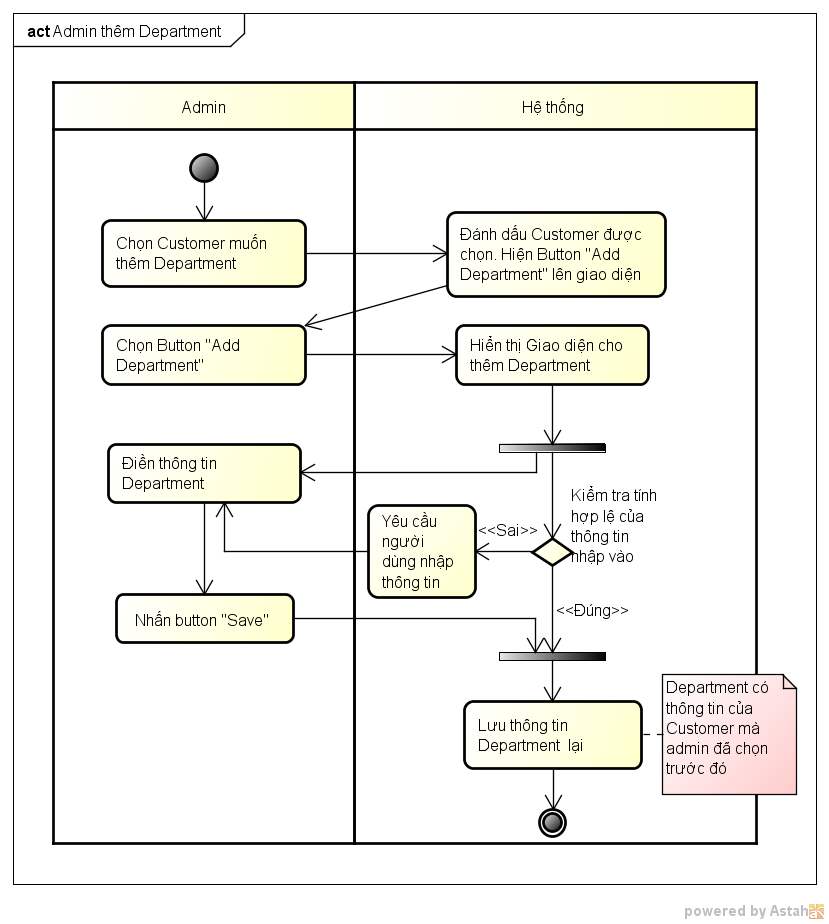
2.3.3.2 Chi tiết use case “Tạo mới Department cho một Customer”

Hình 16 bên dưới mô hình hóa hoạt động của use case “Tạo mới Department”

Tác nhân được tham gia vào use case là Admin.

Trình tự thực hiện:

* Tác nhân: Chọn Customer muốn tạo Department
* Hệ thống: Đánh dấu Customer được chọn. Hiển thị button “Add Department” lên giao diện.
* Tác nhân: Nhấn button “Add Department”
* Hệ thống: Hiển thị giao diện cho nhập thông tin Department.
* Tác nhân: Nhập thông tin của Department
* Hệ thống: Kiểm tra tính hợp lệ của thông tin người dùng nhập vào
* Tác nhân: Nhấn button “Save” để lưu Department
* Hệ thống: Nếu kiểm tra thông tin nhập vào hợp lệ thì tạo mới một Department với thông tin đã nhập vào và có thông tin Customer là Customer đã chọn trước đó. Ngược lại, nếu thông tin người dùng nhập vào không hợp lệ thì yêu cầu người dùng nhập lại.
* Hệ thống: Hiển thị thông tin Department mới tạo lên giao diện.



Hình 16 Biểu đồ hoạt động use case “Tạo mới Department cho một Customer”

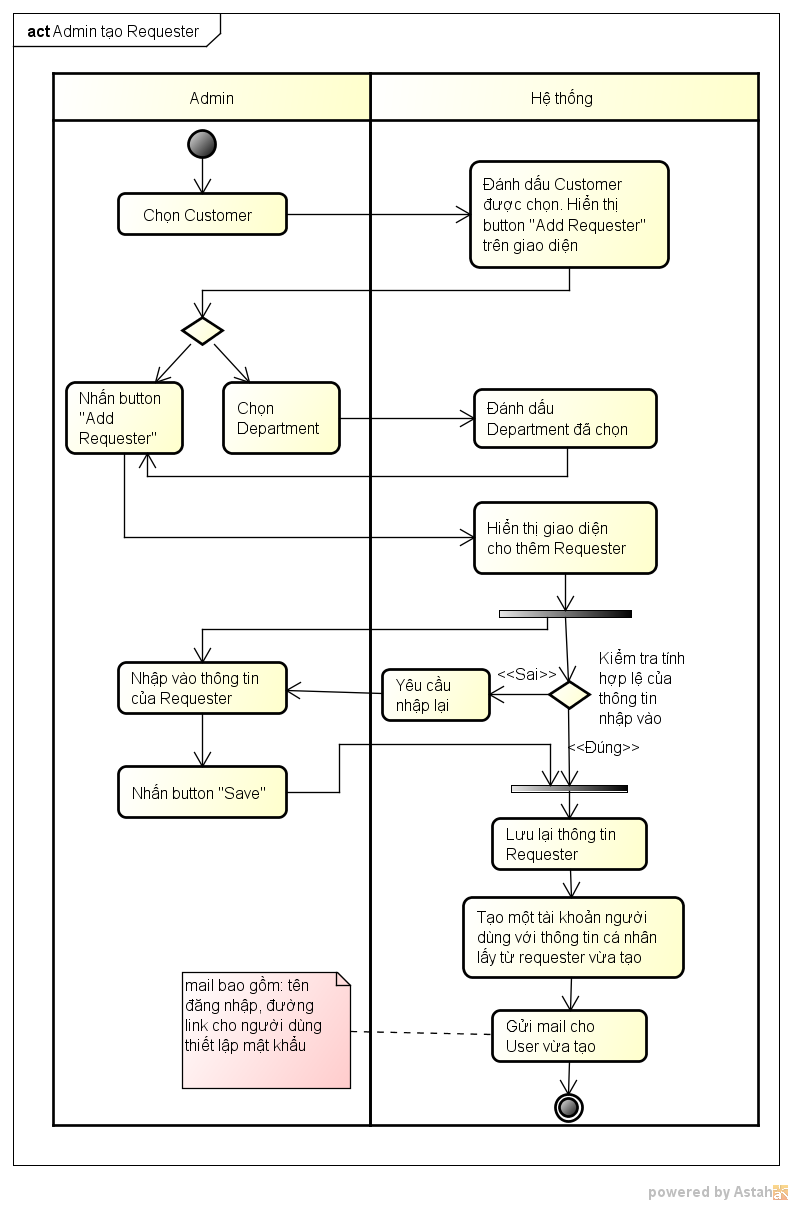
2.3.3.3 Chi tiết use case “Tạo mới Requester”

Hình 17 bên dưới là mô hình hóa của use case “Tạo mới Requester”.

Tác nhân tham gia vào use case là Admin.

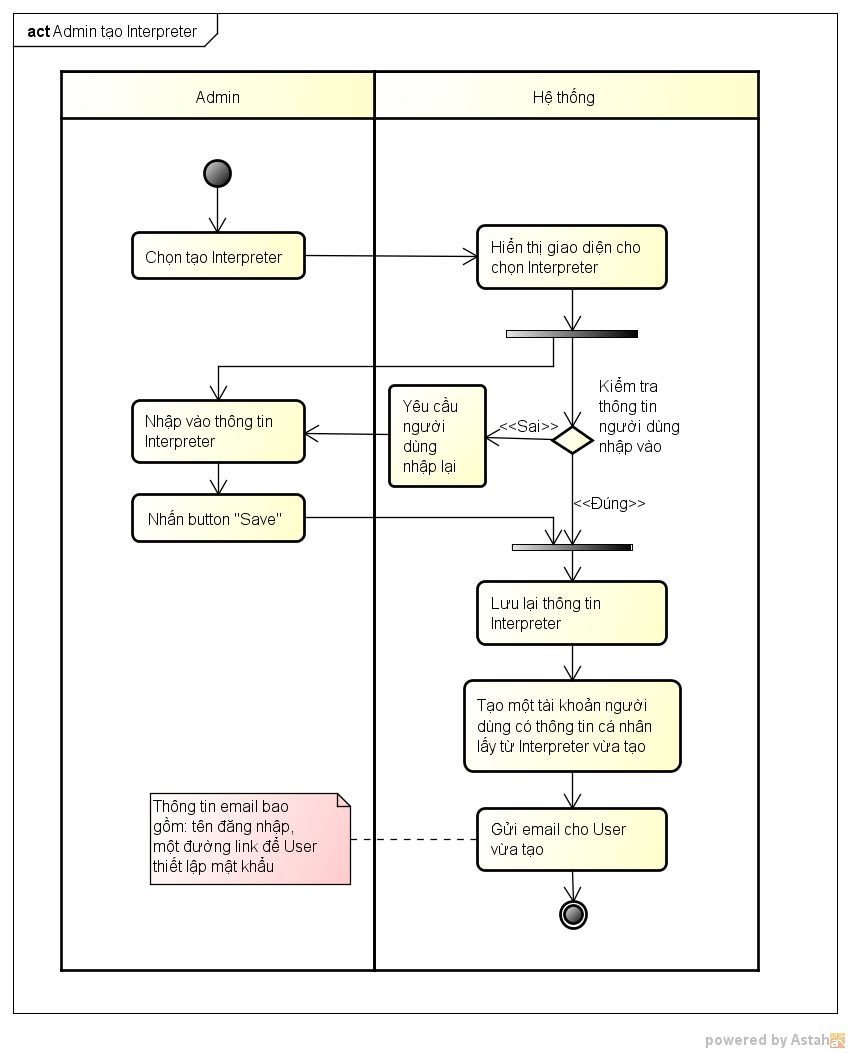
Trình tự thực hiện:

* Tác nhân: Chọn Customer muốn tạo Requester.
* Hệ thống: Đánh dấu Customer đã chọn. Hiển thị button “Add Requester” lên giao diện.
* Tác nhân: Chọn Department của Customer muốn tạo Requester (Hành động này có thể có hoặc không. Tùy thuộc vào ngưới dùng muốn tạo Requester cho Customer hay Department).
* Hệ thống: Đánh dấu Department người dùng đã chọn
* Tác nhân: Nhấn button “Add Requester”
* Hệ thống: Hiển thị giao diên cho nhập thông tin requester
* Tác nhân: Nhập vào thông tin của Requester
* Hệ thống: Kiểm tra tính hợp lệ của thông tin người dùng nhập vào
* Tác nhân: Nhấn button “Save” để lưu thông tin Requester.
* Hệ thống: Nếu kiểm tra thấy thông tin nhập vào không hơp lệ thì yêu cầu người dùng nhập lại. Nếu thông tin đúng thì tạo mới một Requester có thông tin Customer/Department như đã chọn trước đó và lưu lại. Đồng thời, hệ thống tạo mới một tài khoản người dùng (User) chính là tài khoản để Requester này đăng nhập vào hệ thống. Hệ thống sẽ gửi mail cho User với thông tin bao gồm: tên đăng nhập, một đường link để người dùng thiết lập mật khẩu. Sau khi thiết lập mật khẩu, User có thể đăng nhập vào hệ thống với tên đăng nhập và mật khẩu mới thiết lập. Và có quyền của một Requester.



Hình 17 Biểu đồ hoạt động use case “Tạo mới Requester”

2.3.3.4 Chi tiết use case “Tạo mới Interpreter”



Hình 18 Biểu đồ hoạt động use case “Tạo mới Interpreter”

Hình 18 trên mô hình hóa hoạt động use case “Tạo mới Interpreter”.

Tác nhân tham gia vào use case là Admin.

Trình tự thực hiện:

* Tác nhân: Chọn tạo mới Interpreter
* Hệ thống: Hiển thị giao diên cho nhập thông tin tạo mới Interpreter
* Tác nhân: Nhập vào thông tin của Interpreter
* Hệ thống: Kiểm tra tính hợp lệ của thông tin người dùng nhập vào
* Tác nhân: Nhấn button “Save” để lưu thông tin Interpreter.
* Hệ thống: Nếu kiểm tra thấy thông tin nhập vào không hơp lệ thì yêu cầu người dùng nhập lại. Nếu thông tin đúng thì tạo mới một và lưu lại. Đồng thời, hệ thống tạo mới một tài khoản người dùng (User) chính là tài khoản để Interpreter này đăng nhập vào hệ thống. Hệ thống sẽ gửi mail cho User với thông tin bao gồm: tên đăng nhập, một đường link để người dùng thiết lập mật khẩu. Sau khi thiết lập mật khẩu, User có thể đăng nhập vào hệ thống với tên đăng nhập và mật khẩu mới thiết lập. Và có quyền của một Interpreter.

**KẾT CHƯƠNG:**

*Chương vừa rồi người viết đồ án đã trình bày chi tiết phân tích và thiết kế hệ thống về mặt chức năng. Đồng thời đặc tả một số use case chính của hệ thống.*

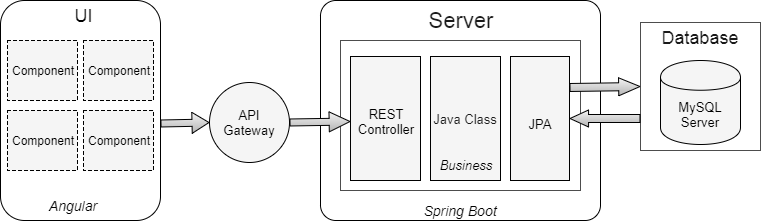
*Chương sau, người viết đồ án sẽ trình bày quá trình xây dựng và phát triển hệ thống.*

# Chương 4 Xây Dựng Và Phát Triển Hệ Thống

*Trong chương này người thực hiện đồ án sẽ trình bày chi tiết xây dựng và phát triển hệ thống. Thứ nhất là tổng quan hệ thống. Thứ hai là xây dựng cơ sở dữ liệu. Thứ ba là thiết kế giao diện người dùng.*

## Thiết kế tổng quan

Hình 19 dưới đây là mô hình hóa kến trúc tổng quan của hệ thống.



Hình 19 Kiến trúc triển khai của hệ thống

Phía giao diện người dùng hệ thống sử dụng Angular framework để phát triển. Các chức năng được thiết kế theo các component.

Phía server hệ thống sử dụng Spring Boot framework để phát triển. Đầu ra của server là các API cung cấp các service.

Khi người dùng yêu cầu một dịch vụ. Phía UI – ta gọi là phía client nhận biết đó là yêu cầu dịch vụ nào. Thông qua API Gateway mà gọi được api cần thiết. Khi phía client gọi đến api đó, server sẽ biết phải làm gì và trả gì về cho phía client.

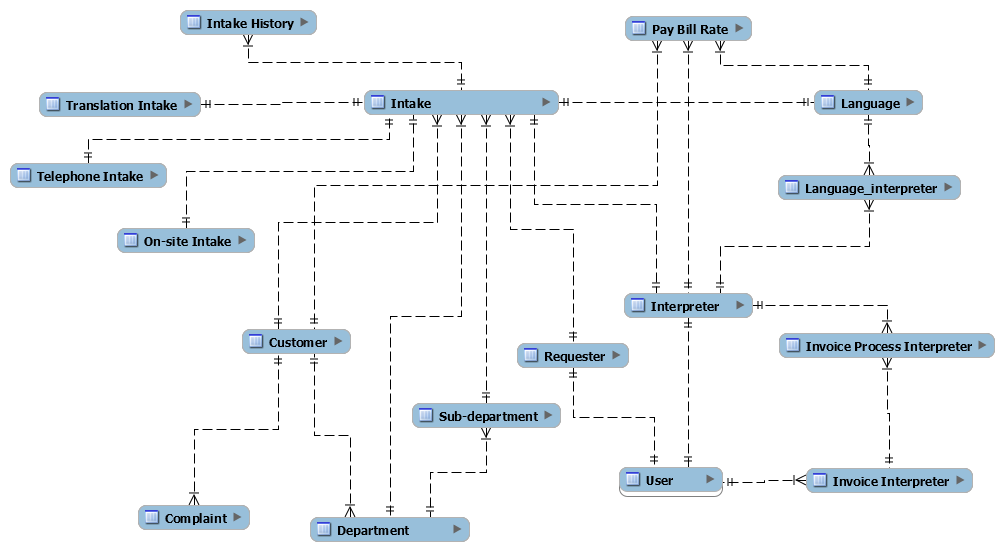
Database của hệ thống được quản lý bởi Hệ quản trị cơ sở dữ liệu MySQL. Server giao tiếp với Database thông qua JPA (Java Persistence API). JPA là một phần của framework Spring Boot.

Tiếp sau đây, người viết đồ án sẽ trình bày chi tiết về thiết kế Database vad thiết kế Giao diện người dùng.

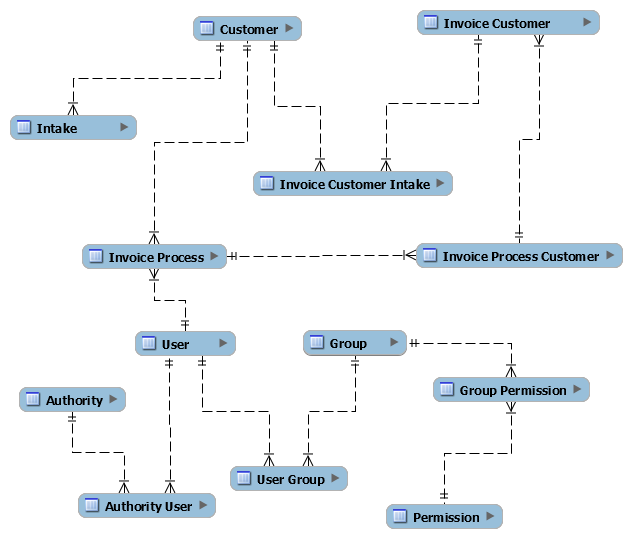
## Thiết kế Cơ sở dữ liệu

### Cơ sở dữ liệu

Hình 20 Hình 21 dưới đây là mô hình rút gọn của cỡ sử dữ liệu và quan hệ giữa các bảng. Do các bảng dữ liệu quá nhiều nên em đã tách ra làm hai mô hình cho dễ quan sát. Theo đó, bảng Intake và User ở hai hình là một.



Hình 20 Mô hình các bảng cơ sở dữ liệu thu gọn



Hình 21 Mô hình bảng Cơ sở dữ liệu rút gọn (tiếp)

Sau đây, người viết đồ án sẽ mô tả chi tiết cấu trúc các bảng cơ sở dữ liệu

### Chi tiết các bảng cơ sở dữ liệu

2.2.1 Bảng “Intake”

|  |  |  |  |
| --- | --- | --- | --- |
| Tên thuộc tính | Kiểu dữ liệu | Mô tả | Giá trị minh họa |
| id | Long | Khóa chính | 1001 |
| intake\_type | Integer | Kiểu của lịch hẹn  Có 6 kiểu tương ứng với giá trị từ 0 đến 5. | Giá trị 1 là kiểu lịch hẹn thoại. |
| status | Integer | Trạng thái của lịch hẹn | Giá trị 5 là “uncover” |
| app\_nbr | String | Một chuỗi kí tự, được hiểu như id giúp phân biệt lịch hẹn này với lịch hẹn khác | DOC17-6, IP17-22 |
| time\_of\_service | Datetime | Ngày lịch hẹn diễn ra | 2017-12-21T09:00:00Z |
| service\_recipients | String | Thông tin của người trực tiếp nhận cuộc hẹn phía khách hàng |  |
| pay\_items | String | Pay rate của lịch hẹn | 120.0 |
| bill\_items | String | Bill rate của lịch hẹn | 200.0 |
| total\_pay | Double | Tổng tiền phải trả cho Interpreter | 240.0 |
| total\_bill | Double | Tổng tiền khách hàng phải trả | 400.0 |
| duration | Double | Khoảng thời gian của lịch hẹn | 120 đơn vị phút |
| start\_time | Datetime | Thời gian bắt đầu lịch hẹn | 2017-12-21T09:00:00Z |
| finish\_time | Datetime | Thời gian kết thúc lịch hẹn | 2017-12-21T11:00:00Z |
| canceler | Integer | Id của người hủy lịch hẹn | 1 |
| cancel\_reason | String | Lý do hủy lịch hẹn (Đối với Requester hủy lịch hẹn) | Lỗi của khách hàng |
| total\_bill\_pre\_tax\_amount | Double | Tổng hóa đơn chưa tính thuế | 200.0 |
| total\_bill\_taxed | Double | Tổng hóa đơn đã tính thuế | 210.0 |
| cover\_by\_id | Long | Id của Interpreter nhận lịch hẹn | 1 |
| appointment\_intake\_id | Long | Nếu lịch hẹn là kiểu on-site, nó sẽ có appointment\_intake\_id | 1 |
| telephone\_intake\_id | Long | Nếu lịch hẹn là kiểu thoại, nó sẽ có telephone\_intake\_id | 1 |
| translation\_intake\_id | Long | Nếu lịch hẹn kiểu translation, nó sẽ có translation\_intake\_id | 1 |
| demand\_telephone\_intake\_id | Long | Nếu lịch hẹn kiểu on-demand telephone nó sẽ có demand\_telephone\_intake\_id | 1 |
| language\_id | Long | Id của ngôn ngữ của lịch hẹn | 1 |
| requester\_id | Long | Id của Requster yêu cầu lịch hẹn | 1 |
| customer\_id | Long | Id của khách hàng có Requester là đại diện | 1 |
| Sup\_department\_id | Long | Id của sub-department đặt lịch hẹn | 1 |
| department\_id | Long | Id của department đặt lịch hẹn | 1 |
| process\_pay | Boolean | Biểu thị Lịch hẹn đã được tính tiền chưa | true |
| process\_bill | Boolean | Biểu thị Lịch hẹn đã được tính tiền chưa | true |
| cancel\_time | Datetime | Thời gian hủy lịch hẹn(đối với Requester hủy) |  |
| canceler\_name | String | Tên người hủy lịch hẹn(đối với Requster hủy) |  |
| service\_type | Integer | Kiểu dịch vụ  Có 4 giá trị tương ứng với 4 kiểu | Giá trị 4 là kiểu phiên dịch thoại |

Bảng 1 Cấu trúc dữ liệu bảng “Intake”

2.2.2 Chi tiết bảng “Customer”

|  |  |  |  |
| --- | --- | --- | --- |
| Tên thuộc tính | Kiểu dữ liệu | Mô tả | Giá trị minh họa |
| id | Long | Khóa chính | 1001 |
| active | Boolean | Biểu thị trạng thái kích hoạt hay vô hiệu hóa của Customer | true |
| name | String | Tên của customer | Ly Nguyen |
| contact\_name | String | Tên gọi khi liên hệ của Customer | Ly Nguyen |
| bus\_phone | String | Số điện thoại Làm việc | 01658190669 |
| cell\_phone | String | Số điện thoại cá nhân | 01658190669 |
| fax\_number | String | Số fax |  |
| email | String | mail của customer | abc@sample.com |
| website | String | Trang web của customer |  |
| access\_code | String | Mã của Customer giống như id. Nếu Customer Department, Sub-department thì các Department, Sub-department này cũng có Access code giống với Customer | 221217 |
| invoice\_style | String | Kiểu hóa đơn | 0 là kiểu Account Unit |
| add\_1 | String | Địa chỉ của Customer | HUST, số 1 Đại Cồ việt |
| add\_2 | String | Địa chỉ của Customer |  |
| add\_3 | String | Địa chỉ của Customer |  |
| city | String | Thành phố | Hà Nội |
| zip\_code | String | Mã bưu chính của thành phố | 100000 |
| broadcast | Boolean | Nếu broadcast ở trạng thái true, khi Customer đặt lịch hẹn thì tất cả các interpreter có thể thực hiện lịch hẹn có thể nhìn thấy và nhận | true |
| tax\_rate | Double | Mức thuế | 0.1 |
| email\_noti\_app\_submit | Boolean | Có thông báo khi lịch hẹn của Customer được xác nhận không | true |
| email\_noti\_app\_cancel | Boolean | Có thông báo khi lịch hẹn của Customer bị hủy không | true |
| email\_noti\_app\_decline | Boolean | Có thông báo khi lịch hẹn của Customer bị hoãn không | true |
| min\_minute\_bill | Double | Thời gian ngắn nhất để tính phí cho cuộc hẹn(phút) kiểu on-site | Ví dụ giá trị cho thuộc tính này là 120.  Một cuộc hẹn mà customer đặt kéo dài 100 phút thì vẫn tính phí cho khoảng thời gian cuộc hẹn là 120 |
| next\_bill\_incr | Double | Mốc thời gian để làm tròn cho tổng thời gian cuộc hẹn(phút) kiểu on-site | Ví dụ cuộc hẹn kéo dài 2h15 phút sẽ làm tròn thành 2h30 phút |
| min\_minute\_bill\_tel | Double | Thời gian ngắn nhất để tính phí cho cuộc hẹn(phút) kiểu hội thoại |  |
| next\_bill\_incr\_tel | Double | Mốc thời gian để làm tròn cho tổng thời gian cuộc hẹn(phút) kiểu hội thoại |  |

Bảng 2 Cấu trúc dữ liệu bảng “Customer”

2.2.3 Chi tiết bảng “Requester”

|  |  |  |  |
| --- | --- | --- | --- |
| Thuộc tính | Kiểu dữ liệu | Mô tả | Giá trị minh họa |
| id | Long | Khóa chính | 1521 |
| supervisor\_name | String | Tên người giám sát | Trịnh Tuấn Đạt |
| `noti\_cover | Boolean | Thông báo khi lịch hẹn được cover không | true |
| noti\_decline | Boolean | Thông báo khi lịch hẹn bị hủy không | true |
| sup\_department\_id | Long | Id của Sub-department mà Requester đại diện cho | 1001 |
| department\_id | Long | Id của Department mà Requester đại diện cho | 1003 |
| customer\_id | Long | Id của Customer mà Requster đại diện cho | 1021 |
| user\_id | Long | Id của user cũng chính là requester đó | 1051 |

Bảng 3 Cấu trúc dữ liệu Bảng “Requester”

2.2.4 Chi tiết bảng “Interpreter”

|  |  |  |  |
| --- | --- | --- | --- |
| Thuộc tính | Kiểu dữ liệu | Mô tả | Giá trị minh họa |
| id | Long | Khóa chính | 1521 |
| user\_id | Long | Id của user cũng chính là Interpreter đó | 1001 |
| int\_type | Integer | Kiểu của Interpreter |  |
| bus\_name | String | Họ tên muốn gọi | Ly Nguyen |
| ssn | String | Social security number |  |
| dob | Datetime | Ngày sinh | 13/03/1995 |
| driver\_license | String | Số bằng lái xe |  |
| start\_date | Datetime | Ngày bắt đầu làm việc | 22/12/2017 |
| gender | integer | Giới tính của interpreter | 1 là Nam, 2 là nữ |
| working\_cities | String | Thành phố mà Interpreter làm việc | Hà Nội |
| notes | String | Ghi chú |  |
| phones | String | Số điện thoại của interpreter | 01658190669 |
| accreditation | String | Chứng chỉ, bằng cấp của interpreter | Chứng chỉ TOEFL |
| min\_minute\_paid | Double | Thời gian ngắn nhất để tính phí cho một cuộc hẹn là kiểu on-site |  |
| next\_pay\_incr | Double | Mốc thời gian làm tròn cho cuộc hẹn kiểu on-site |  |
| min\_minute\_paid\_tel | Double | Thời gian ngắn nhất để tính phí cho một cuộc hẹn là kiểu telephone |  |
| next\_pay\_incr\_tel | Double | Mốc thời gian làm tròn cho cuộc hẹn kiểu telephone |  |
| payment\_type | Integer | Kiểu thanh toán | Vi dụ giá trị 1 kiểu thanh toán là Pay |
| text\_msg\_number | String | Số điện thoại của interpreter để gứi tin nhắn | 01658190669 |
| text\_noti\_app\_sent\_inbox | Boolean | Gửi tin nhắn thông báo khi có lịch hẹn vào folder của interpreter | true |
| text\_noti\_app\_cancel | Boolean | Gửi tin nhắn thông báo khi lịch hẹn bị hủy | true |
| text\_noti\_app\_sent\_schedule | Boolean | Gửi tin nhắn thông báo khi có lịch hẹn vào schedule của interpreter | true |
| in\_person\_pay\_rate | Double | Mức phí cho interpreter với lịch hẹn là on-site | 120.0 |
| tel\_pay\_rate | Double | Mức phí cho interpreter với lịch hẹn là kiểu telephone | 100.0 |
| trans\_min\_word\_paid | Integer | Số từ tối thiểu phải trả phí cho một văn bản dịch | 100 |
| trans\_next\_pay\_inc\_word | Integer | Số từ tối thiểu được làm tròn | 1 |

Bảng 4 Cấu trúc dữ liệu bảng “Interpreter”

2.2.5 Chi tiết bảng “User”

|  |  |  |  |
| --- | --- | --- | --- |
| Thuộc tính | Kiểu dữ liệu | Mô tả | Giá trị minh họa |
| id | Long | Khóa chính | 1251 |
| login | String | Tên đăng nhập của user | Lynt |
| pass\_hash | String | Password của user sau khi được mã hóa |  |
| first\_name | String | Tên đầu của user | ly |
| last\_name | String | Tên sau của user | nguyen |
| email | String | Mail của user | sample@abc.xyz |
| phone\_number | String | Số điện thoại của user | 01688190669 |
| image\_url | String | Link ảnh của user |  |
| activated | Boolean | Trạng thái tài khoản | True: có hiệu lực  False: không hiệu lực |
| reset\_key | String | Một chuỗi string để user reset mật khẩu khi lần đầu đăng nhập vào tài khoản |  |
| created\_by | String | Danh xưng người tạo tài khoản user | admin |
| created\_date | Datetime | Ngày tạo tài khoản | 2017-12-01T09:00:00Z |
| reset\_date | Datetime | Ngày tài khoản được tạo mật khẩu lần đầu | 2017-12-11T09:00:00Z |
| last\_modified\_by | String | Danh xưng người cập nhật thông tin user lần cuối | admin |
| last\_modified\_date | String | Ngày nhật thông tin user lần cuối | 2017-12-21T09:00:00Z |
| customer\_id | Long | Id của Customer chính là user này(trường hợp user là customer) | 1 |

Bảng 5 Cấu trúc dữ liệu bảng “User”

2.2.6 Chi tiết bảng “sub-department”

|  |  |  |  |
| --- | --- | --- | --- |
| Thuộc tính | Kiểu dữ liệu | Mô tả | Giá trị minh họa |
| id | Long | Khóa chính | 2521 |
| name | String | Tên của Sup-department | Phòng tài vụ |
| location | String | Địa điểm của Sup-department | Số 1 Đại Cồ Việt, Hà Nội, Việt Nam |
| active | Boolean | Sup-department đang có hiệu lực | true |
| access\_code | String | Là giá trị access code của customer | 123abc |
| contact\_name | String | Tên liên hệ | Phòng tài vụ |
| street\_address | String | Địa chỉ | Số 1 Đại Cồ Việt, Hà Nội, Việt Nam |
| email | String | Địa chỉ email của Sub-department |  |
| suite\_number | String | Số của tòa nhà/khu mà sup-department đó đặt | B1 |
| phone\_number | String | Số điện thoại của phòng ban | 01658190669 |
| ext | String | Phần mở rộng của số điện thoại | +84 |
| city | String | Thành phố nơi sup-department đặt | Hà Nội |
| zip\_postal\_code | String | Mã bưu điện/mã vùng | 100000 |
| department\_id | Long | Id của department | 1251 |

Bảng 6 Cấu trúc dữ liệu bảng “Sup-department”

2.2.7 Chi tiết bảng “Department”

|  |  |  |  |
| --- | --- | --- | --- |
| Thuộc tính | Kiểu dữ liệu | Mô tả | Giá trị minh họa |
| id | Long | Khóa chính | 1251 |
| name | String | Tên của department | Phòng nhân sự |
| active | Boolean | department đang có hiệu lực | true |
| access\_code | String | Là giá trị access code của customer | 123abc |
| contact\_name | String | Tên liên hệ | Phòng nhân sự |
| street\_address | String | Địa chỉ | Số 1 Đại Cồ Việt, Hà Nội, Việt Nam |
| email | String | Địa chỉ email của  Sub-department | Nguyenthily13031995@gmail.com |
| suite\_number | String | Số của tòa nhà/khu mà department đó đặt | B1 |
| phone\_number | String | Số điện thoại của phòng ban | 01658190669 |
| ext | String | Phần mở rộng của số điện thoại | +84 |
| city | String | Thành phố nơi department đặt | Hà Nội |
| zip\_postal\_code | String | Mã bưu điện/mã vùng | 100000 |
| customer\_id | Long | Id của customer | 1251 |

Bảng 7 Cấu trúc dữ liệu bảng “Department”

2.2.8 Chi tiết bảng “Pay Bill Rate”

|  |  |  |  |
| --- | --- | --- | --- |
| Thuộc tính | Kiểu dữ liệu | Mô tả | Giá trị minh họa |
| id | Long | Khóa chính | 1251 |
| name | String | Tên của rate | Pay rate item 1 |
| tel\_type |  |  |  |
| effective\_date | Datetime | Ngày rate có hiệu lực | 2017-12-21T09:00:00Z |
| pay\_rate | Double | Mức tiền thanh toán cho interpreter | 200.0 |
| bill\_rate | Double | Mức tiền customer phải trả | 220.0 |
| weekend\_rate | Boolean | Là Mức phí cho ngày cuối tuần? | true |
| discount\_rate | Boolean | Là mức giảm giá? | false |
| active | Boolean | Có hoạt động hay không | true |
| rate\_type | Integer | Kiểu rate | Ví dụ: giá trị 3 là standard bill pate |
| language\_id | Long | Id của language | 1001 |
| customer\_id | Long | Id của customer | 1051 |
| interpreter\_id | Long | Id của interpreter | 1551 |

Bảng 8 Cấu trúc dữ liệu bảng “Pay Bill Rate”

2.2.9 Chi tiết bảng “Invoice Interpreter”

|  |  |  |  |
| --- | --- | --- | --- |
| Thuộc tính | Kiểu dữ liệu | Mô tả | Giá trị minh họa |
| id | Long | Khóa chính | 1251 |
| process\_number | String | Số của hóa đơn, để hiển thị trên giao diện | 1 |
| from\_date | Datetime | Hóa đơn từ ngày | 2017-12-01T09:00:00Z |
| to\_date | Datetime | Hóa đơn đến ngày | 2017-12-21T09:00:00Z |
| create\_date | Datetime | Ngày tạo hóa đơn | 2017-12-21T09:00:00Z |
| user\_id | Long | Id của user tạo hóa đơn | 1201 |

Bảng 9 Cấu trúc dữ liệu bảng “Invoice Interpreter’

2.2.10 Chi tiết bảng “Invoice Process Interpreter”

|  |  |  |  |
| --- | --- | --- | --- |
| Thuộc tính | Kiểu dữ liệu | Mô tả | Giá trị minh họa |
| id | Long | Khóa chính | 1251 |
| total\_pay | Double | Tổng tiền được trả | 12000 |
| intake\_id\_list | String | Danh sách id của lịch hẹn tương ứng | 1001, 1003, 1021 |
| invoice\_intepreter\_id | Long | Id của hóa đơn của Interpreter | 1551 |
| interpreter\_id | Long | Id của interpreter | 3 |

Bảng 10 Cấu trúc dữ liệu bảng “Invoice process Interpreter”

2.2.11 Chi tiết bảng “Invoice Customer”

|  |  |  |  |
| --- | --- | --- | --- |
| Thuộc tính | Kiểu dữ liệu | Mô tả | Giá trị minh họa |
| id | Long | Khóa chính | 1251 |
| total\_bill | Double | Tổng tiền khách hàng phải thanh toán | 120000 |
| invoice\_nbr | String | Mã số hóa đơn | 1 |
| invoice\_process\_customer\_id | Long | Id của Invoce Process Customer | 1201 |

Bảng 11 Cấu trúc dữ liệu bảng “Invoice Customer”

2.2.12 Chi tiết bảng “Language”

|  |  |  |  |
| --- | --- | --- | --- |
| Thuộc tính | Kiểu dữ liệu | Mô tả | Giá trị minh họa |
| id | Long | Khóa chính | 1251 |
| name | String | Tên ngôn ngữ | Vietnamese |
| active | Boolean | Có hiệu lực? | true |

Bảng 12 Cấu trúc dữ liệu bảng “Language”

2.2.13 Chi tiết bảng “Permission”

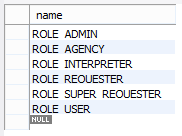
|  |  |  |  |
| --- | --- | --- | --- |
| Thuộc tính | Kiểu dữ liệu | Mô tả | Giá trị minh họa |
| id | Long | Khóa chính | 1251 |
| name | String | Tên của Permission | Tạo lịch hẹn |
| level | Integer | Cấp của Permission | 1 |
| parent | Long | Cha của Permision, chính là id của một Permission khác | 1001 |
| path | String | Đường dẫn component | /intake/appointment-intake |

Bảng 13 Cấu trúc dữ liệu bảng “Permission”

2.2.14 Chi tiết bảng “Authority”

|  |  |  |  |
| --- | --- | --- | --- |
| Thuộc tính | Kiểu dữ liệu | Mô tả | Giá trị minh họa |
| name | String | Tên của Authority. Một tên là duy nhất |  |

Bảng 14 Cấu trúc dữ liệu bảng “Authority”



Hình 22 Các bản ghi của bảng “Authority”

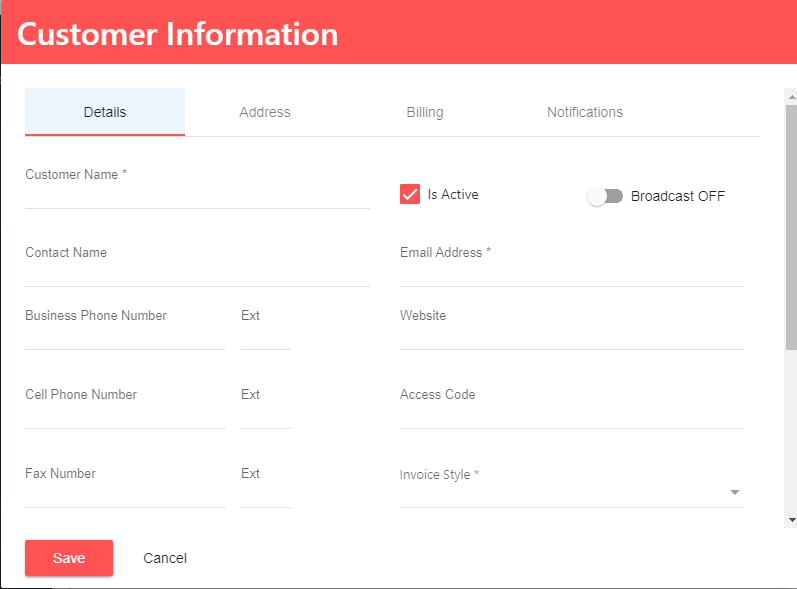
## Thiết kế giao diện

### Giao diện chức năng thêm Customer

Hình 23 là giao diện của chức năng thêm mới Customer.

Người dùng có thể sử dụng chức năng này là Admin.

Người dùng nhập đầy đủ các thông tin bắt buộc phải điền và các chức năng cần thiết khác và nhấn nút “Save” để lưu, hoặc nhấn nút “Cancel” để hủy bỏ việc tạo Customer. Chú ý có bốn menu phải điền thông tin: Details, Address, Billing, Notifications



Hình 23 Giao diện thêm mới Customer

### Giao diện chức năng tạo Department

Người dùng có thể sử dụng chức năng này là admin, customer.

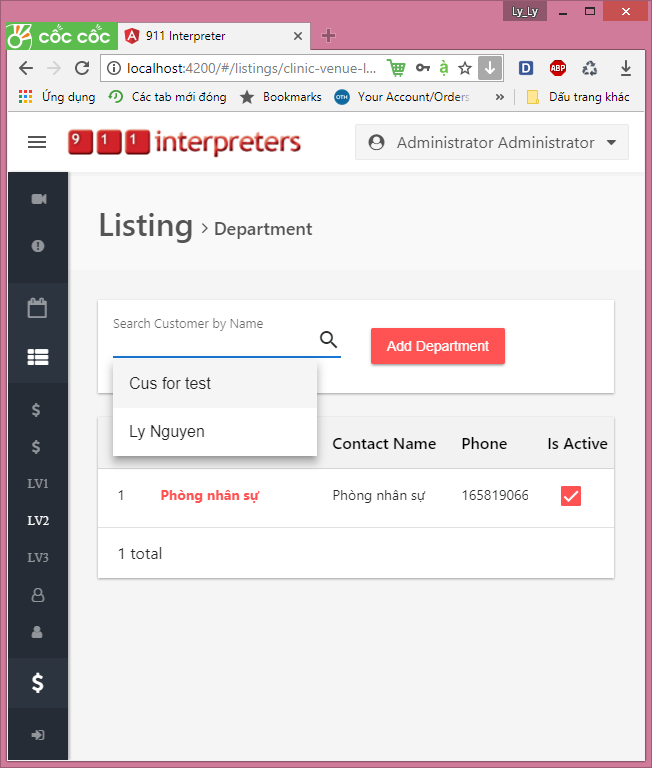
Admin có thể tạo Department cho bất cứ Customer nào.

Customer chỉ có thể tạo Department cho chính mình

Admin khi muốn tạo mới một Department cho một Customer, người dùng phải chọn Customer đó trước thì button “Add Department” mới hiện ra.

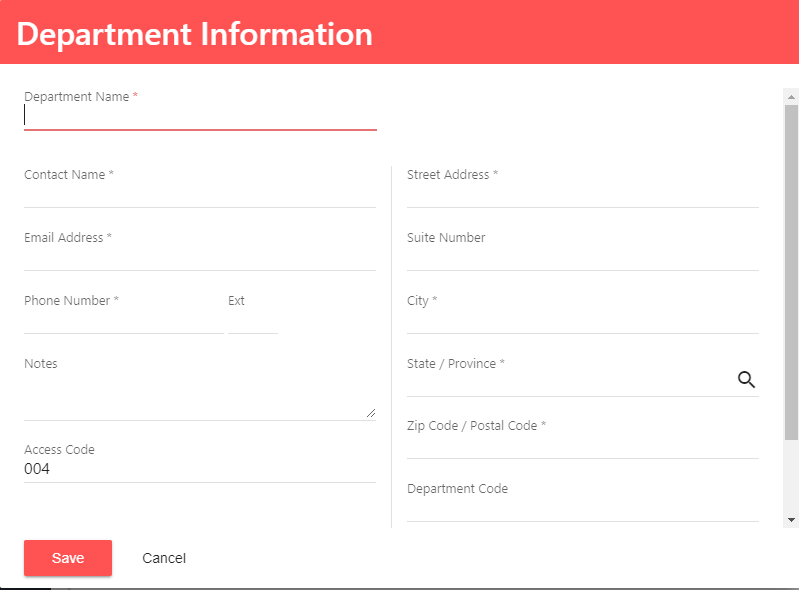
Hình 24 bên dưới là giao diện chọn Customer trước khi tạo Department cho Customer đó.

Sau đó, Người dùng nhấn vào button “Add Department” như trên Hình 24 để tiến hành thêm Department.



Hình 24 Giao diện chọn Customer để thêm Department

Hình 25 ở trang bên cạnh là giao diện thêm Department



Hình 25 Giao diện thêm mới Department

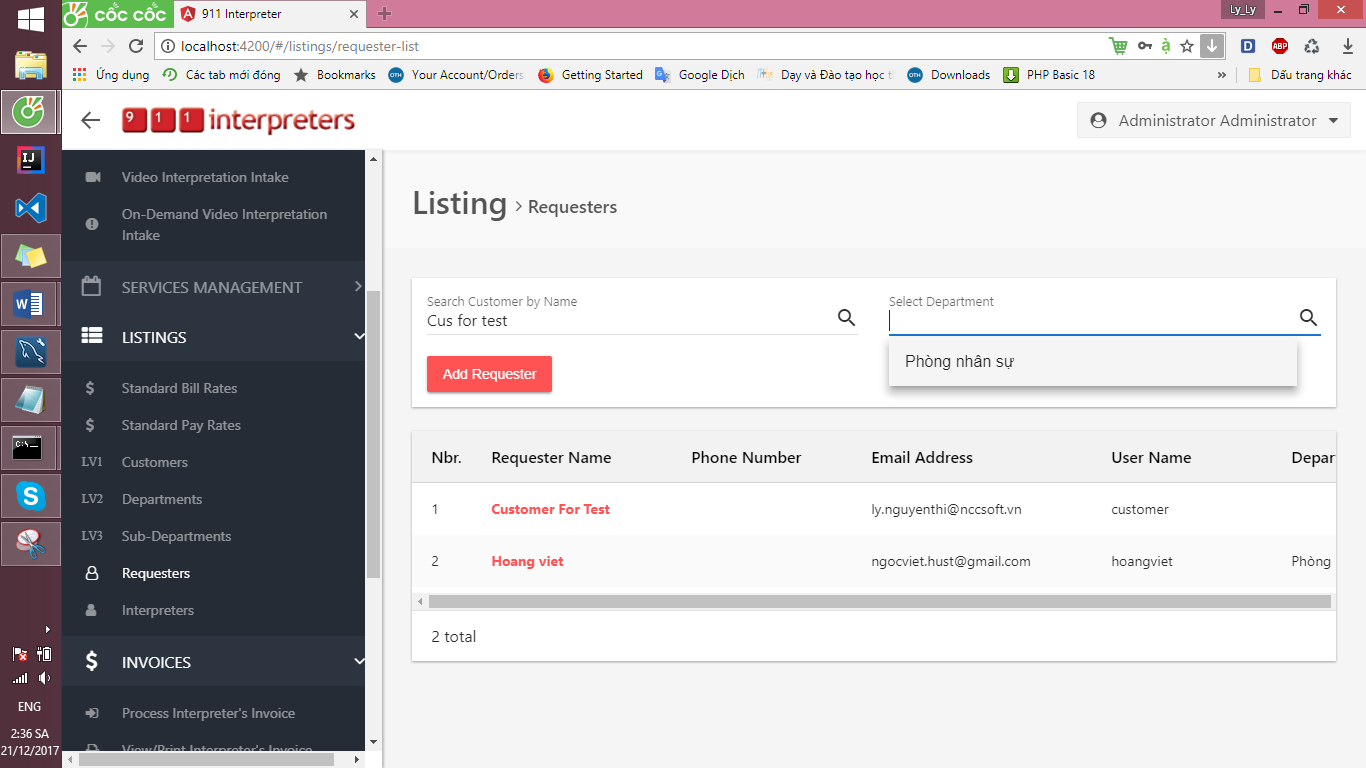
### Giao diện chức năng thêm Requester

Người dùng có thể thực hiện chức năng này chỉ có Admin.

Khi muốn thêm mới Requester cho Customer/Department, người dùng phải chọn Customer, Department đó trước thì button “Add Requester” mới hiện ra.

Hình 26 bên dưới là giao diện cho chọn Customer và Department để thêm mới Requester cho Customer/Department đó.

Sau khi nhấn button “Add Requester”, giao diện thêm Requester sẽ hiện ra.

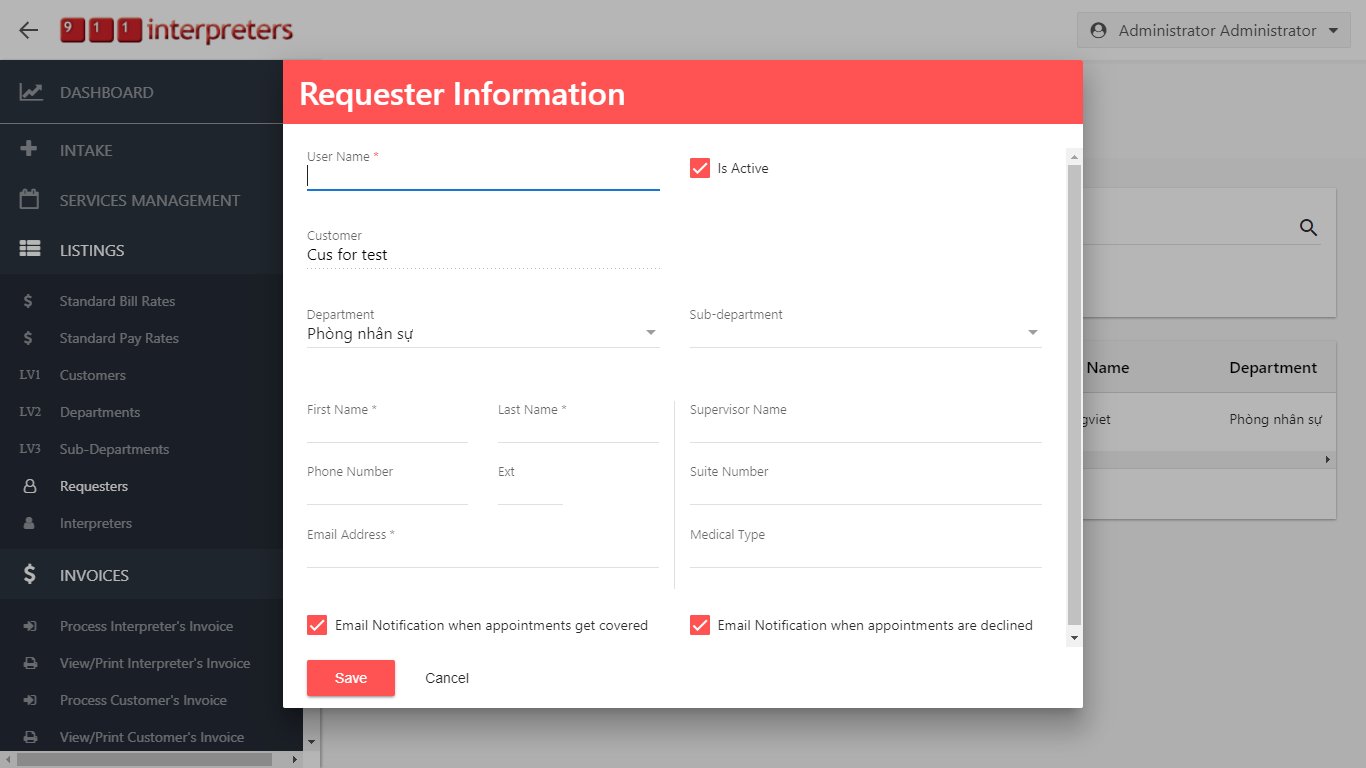


Hình 26 Giao diện chọn Customer và Department để thêm Requester

Hình 27 ở trang bên là giao diện cho thêm mới Requester. Sau khi điền đầy đủ các thông tin bắt buộc và cần thiết, người dùng nhấn nút “Save” để lưu. Hoặc nhấn nút “Cancel” để hủy.

Hành động tạo một Requester đồng thời cũng tạo một người dùng của hệ thống, Người dùng này chính là Requester vừa tạo và được phân các quyền của Requester.

Sau khi tài khoản được tạo, hệ thống sẽ gửi mail cho người dùng vừa tạo bao gồm tên đăng nhập, và một đường link để thiết lập mật khẩu. Người dùng vào đường link đó, thiết lập mật khẩu và sau đó có thể đăng nhập vào hệ thống.



Hình 27 Giao diện cho thêm mới Requester

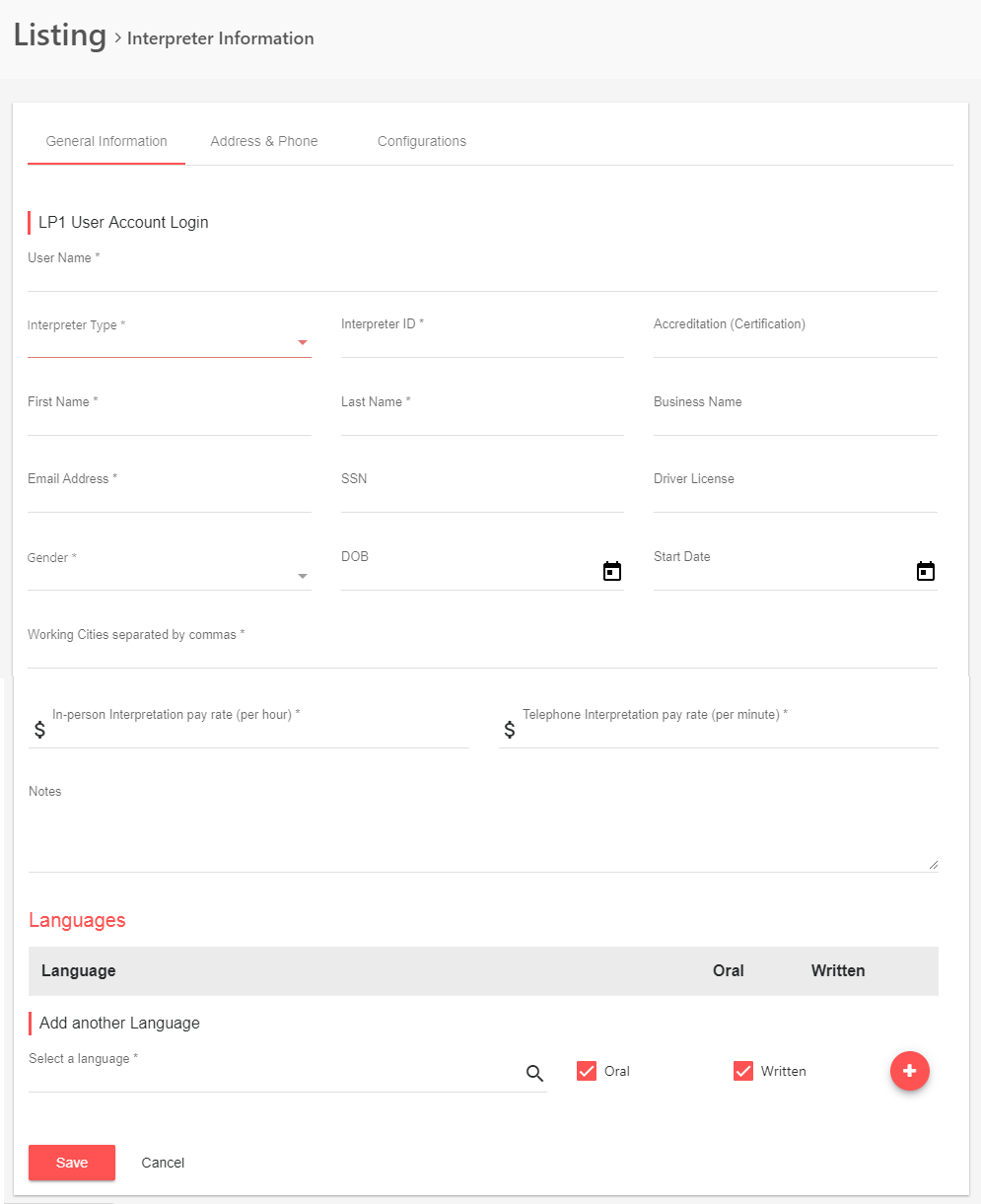
### Giao diện cho chức năng tạo Interpreter

Chỉ Admin có quyền thêm mới Requester

Hình 28 bên dưới là giao diện cho tạo mới một Interpreter. Người dùng nhập đầy đủ các thông tin cần thiết. Sau đó nhấn nút “Save” để lưu, hoặc nhấn nút “Cancel” để hủy.

Khi một Interpreter được tạo, đồng thời một tài khoản người dùng của hệ thống cũng được tạo. User này chính là Interpreter vừa được tạo, có các quyền của một Interpreter.

Sau khi người dùng được tạo, hệ thống sẽ gửi cho người dùng một email bao gồm: tên đăng nhập, một đường link để thiết lập mật khẩu. Sau khi người dùng thiết lập mật khẩu thì có thể đăng nhập vào hệ thống bằng tên đăng nhập và mật khẩu.

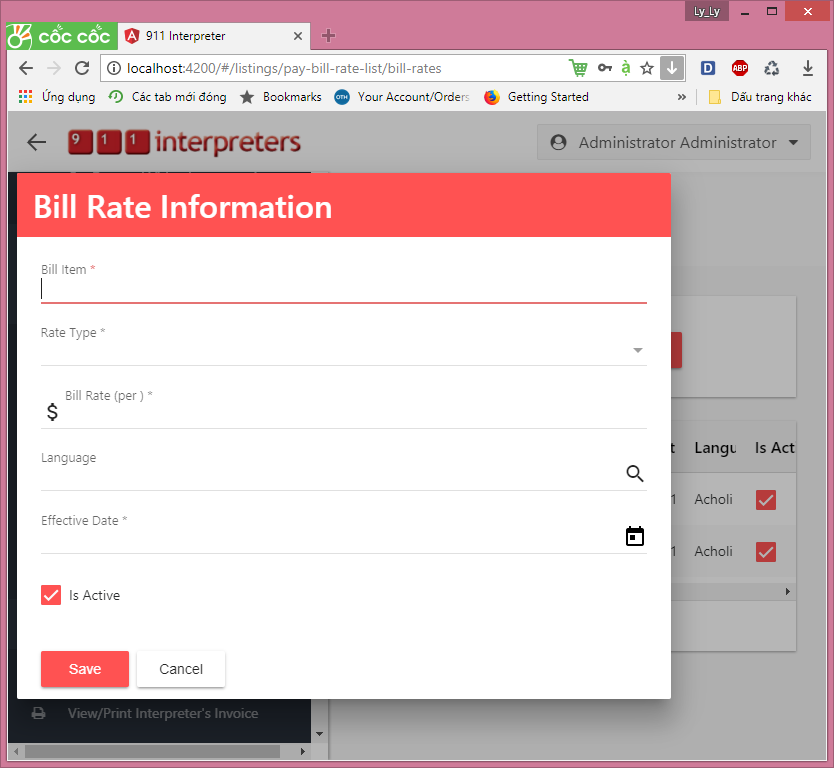


Hình 28 Giao diện thêm mới Interpreter

### Giao diện cho chức năng thêm Bill rate

Chỉ Admin có quyền thực hiện chức năng này.

Hình 29 là giao diện cho thêm mới một Bill rate. Giao diện cho thêm mới Pay rate cũng tương tự. Người dùng nhập vào các thông tin cần thiết và nhấn nút”Save” để lưu. Hoặc nhấn nút “Cancel” để hủy.



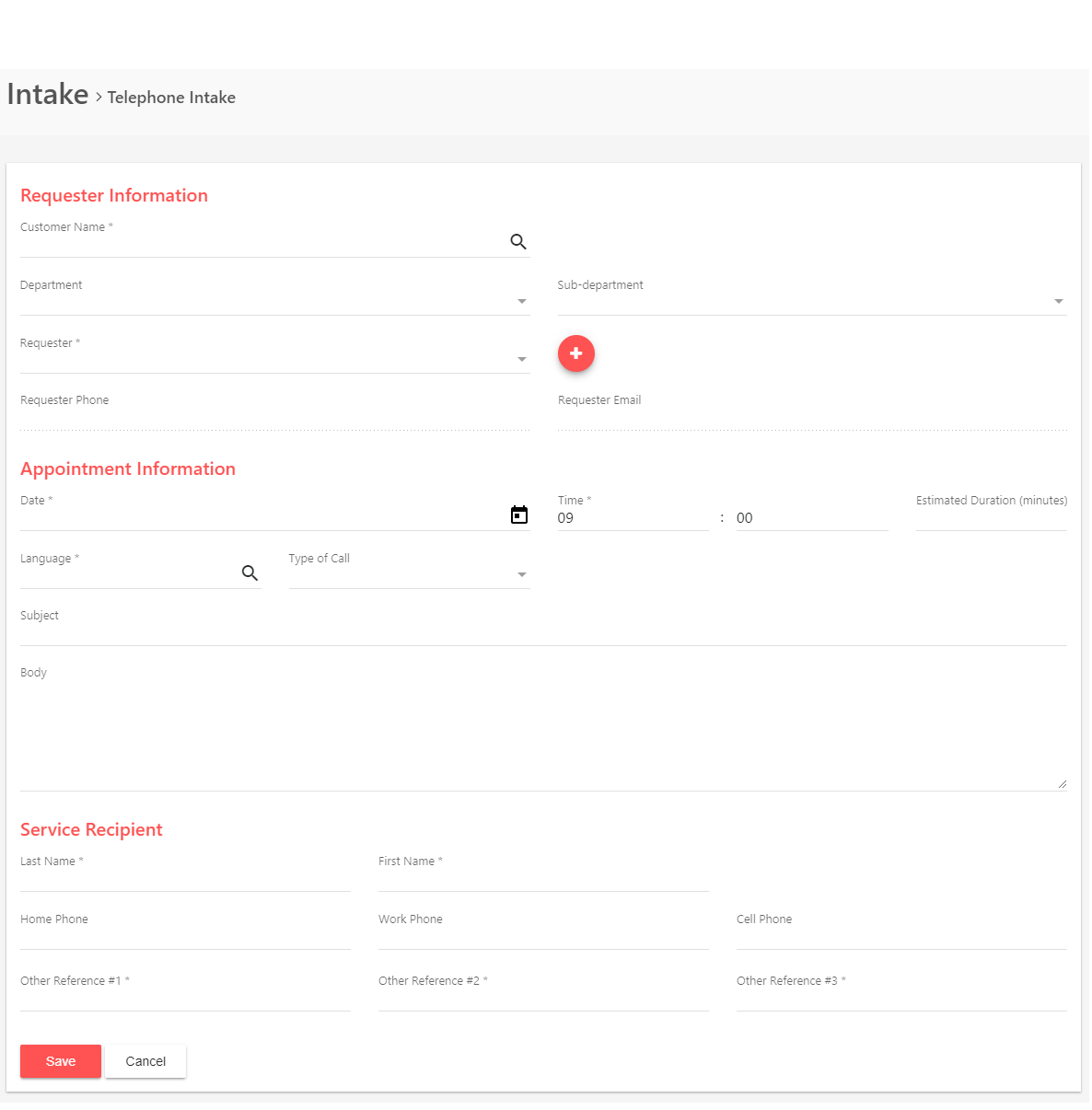
Hình 29 Giao diện thêm mới Bill Rate

### 3.6 Giao diện cho chức năng tạo lịch hẹn

Hình 30 bên dưới là giao diện cho tạo một lịch hẹn kiểu cuộc thoại.

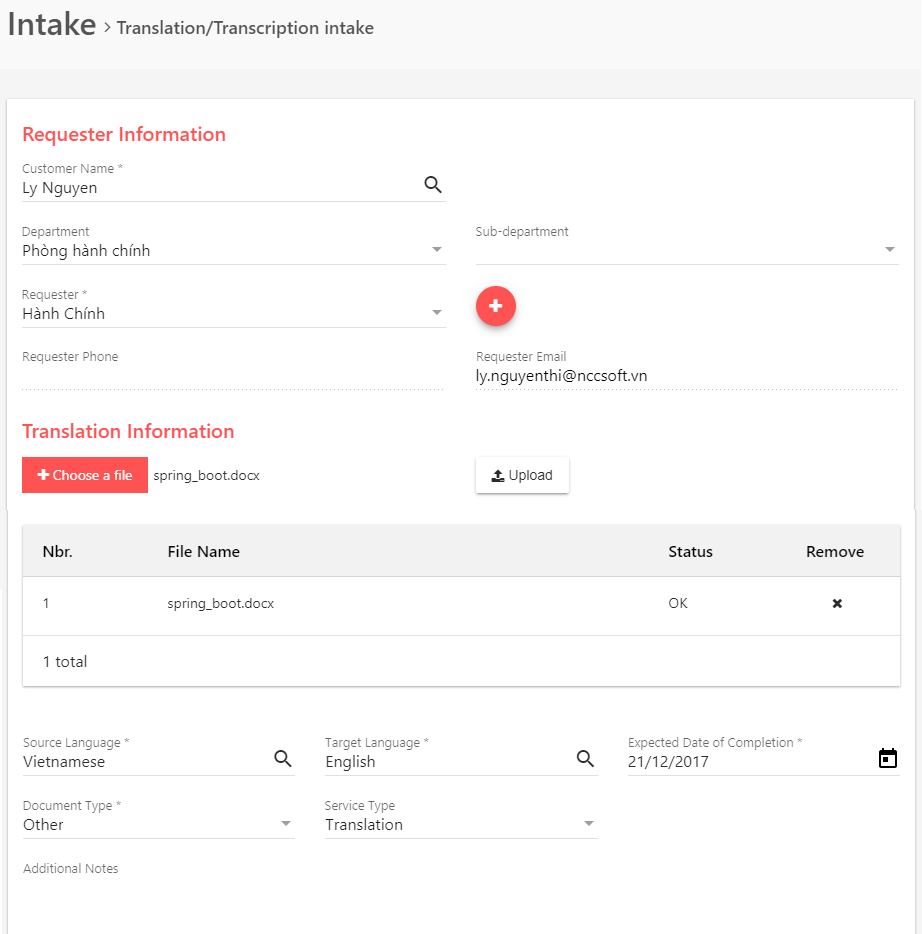
Người dùng nhập đầy đủ các thông tin cần thiết như hình và nhấn nút “Save” để lưu. Hoặc nút “Cancel” để hủy.

Các thông tin phải chú ý điền là: thông tin Requester đặt, đặt cho Customer/Department nào. Thứ hai là thông tin lịch hẹn bao gồm ngày hẹn, thời gian, khoảng thời gian ước tính của cuộc hẹn. Ngôn ngữ cần phiên dịch, kiểu cuộc thoại và các thông tin khác. Và cuối cùng là thông tin người nhận dịch vụ.



Hình 30 Giao diện tạo lịch hẹn thoại

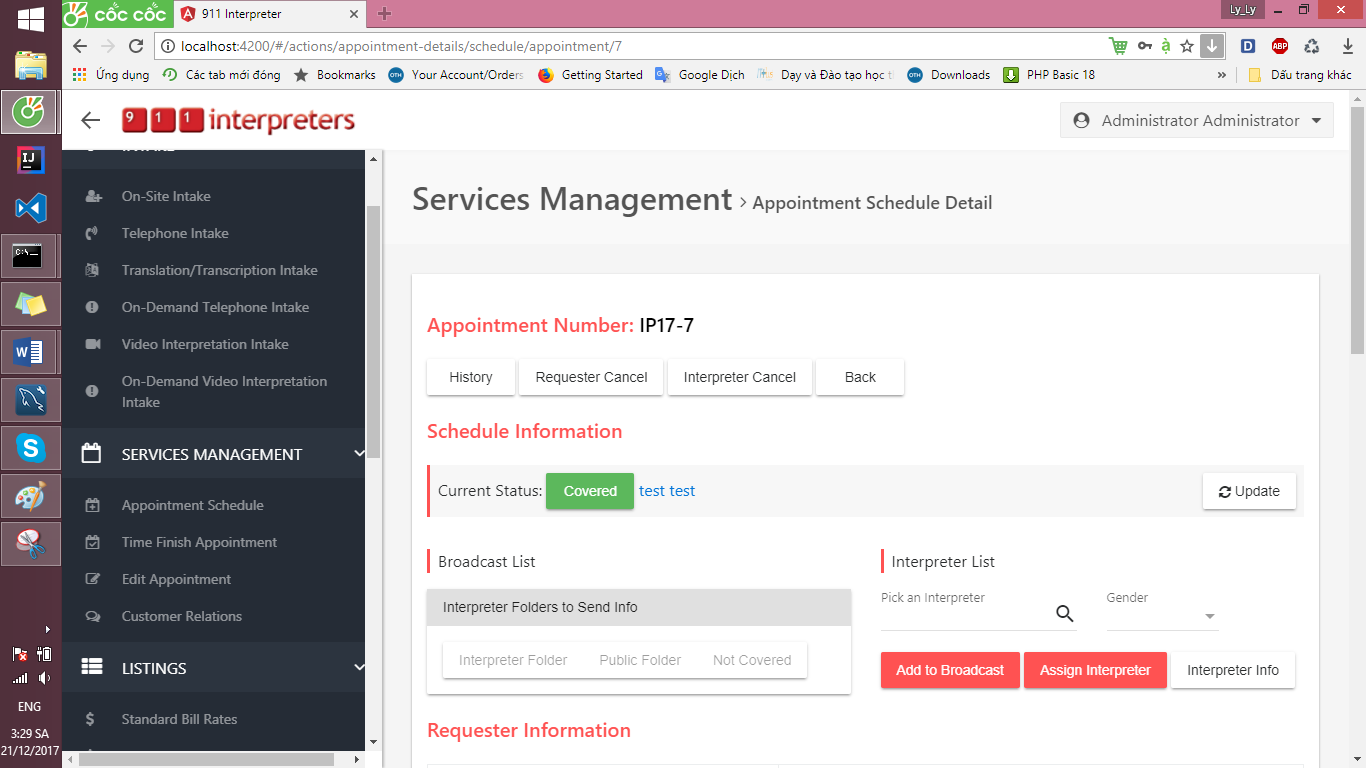
Hình 31 là giao diện cho tạo mới yêu cầu dịch tài liệu. Người dùng nhập đầy đủ các thông tin cần thiết. Tải lên file tài liệu cần dịch. Chọn ngôn ngữ cần dịch của tài liệu, này hy vọng tài liệu dịch xong. Và nhấn nút “Save” để lưu lại, hoặc nhấn nút”Cancel để hủy.



Hình 31 Giao diện tạo yêu cầu dịch tài liệu

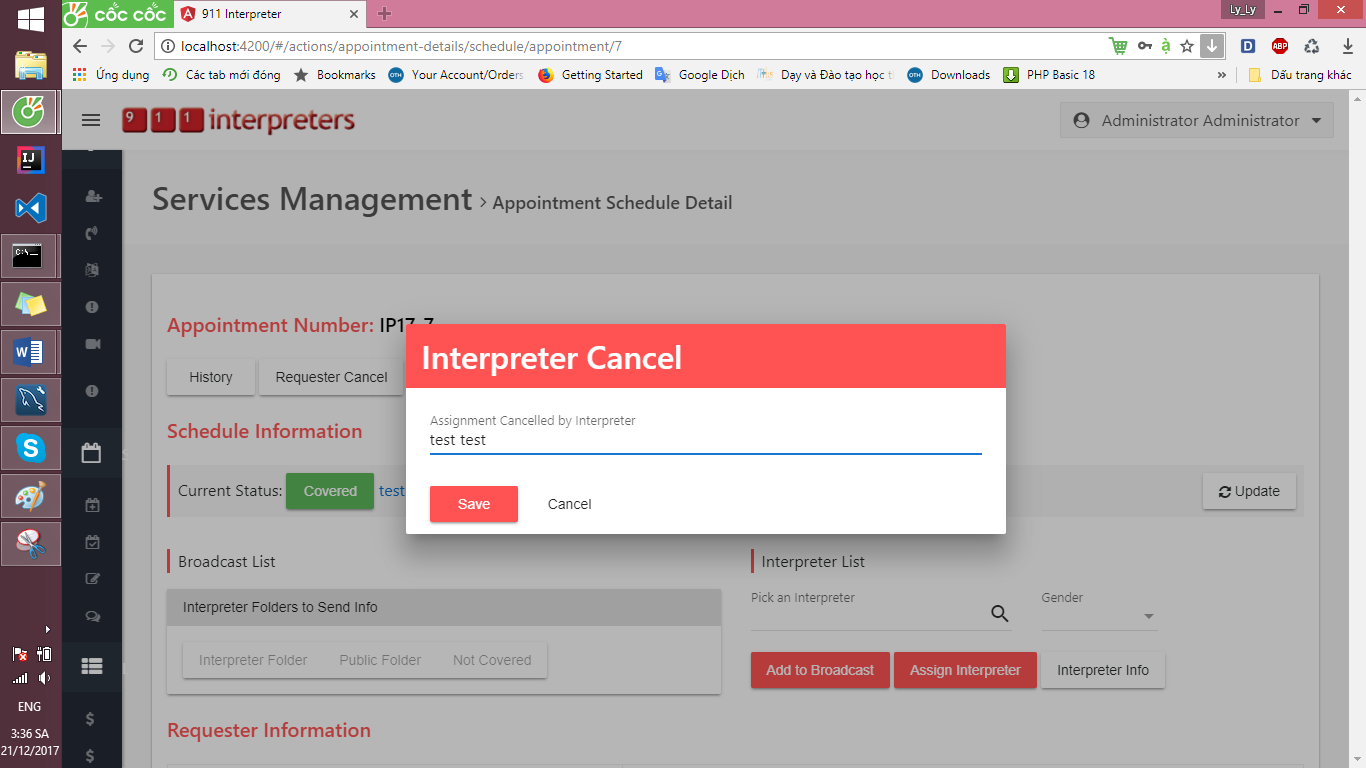
### 3.7 Giao diện cho quản lý lịch hẹn

Hình 32 là giao diện của quản lý lịch hẹn.



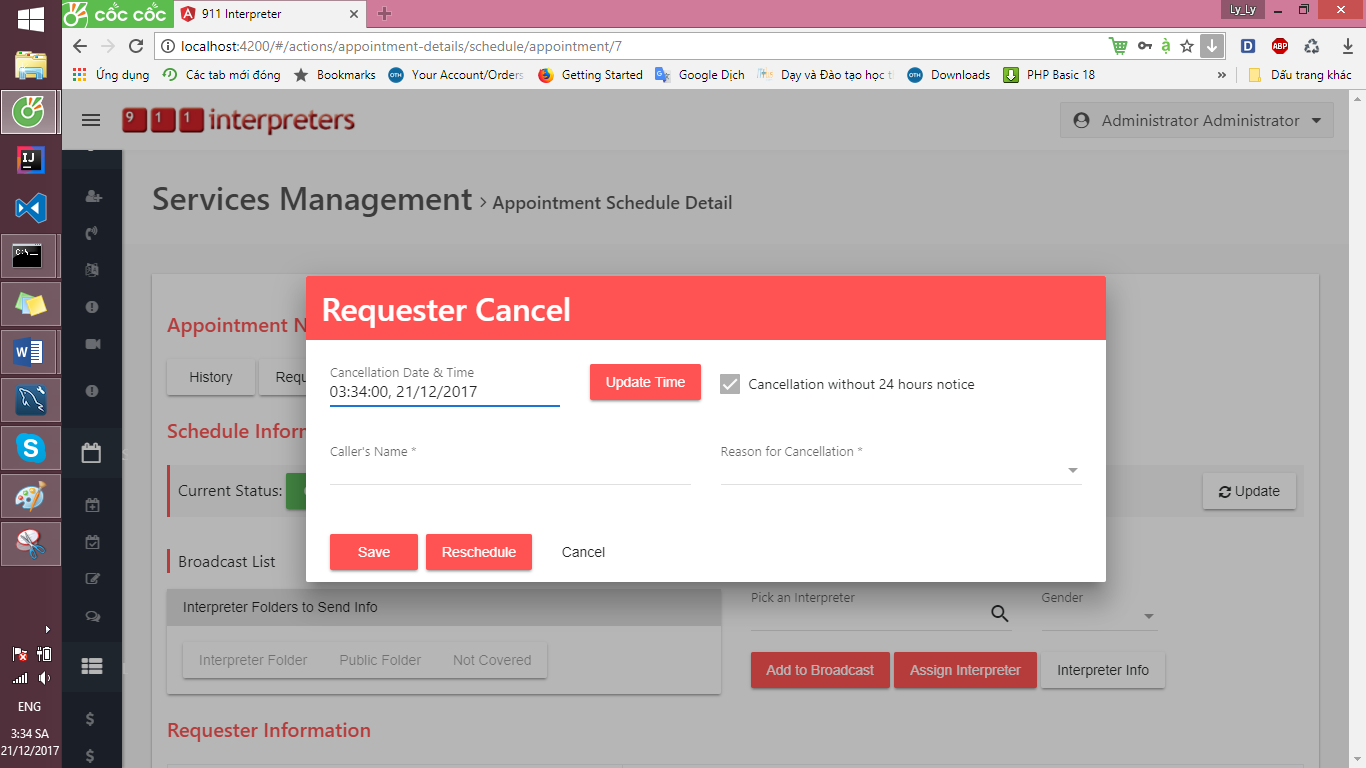
Hình 32 Giao diện chung cho quản lý lịch hẹn

Khi Interpreter hủy lịch hẹn sẽ hiện ra pop-up xác nhận hủy như Hình 33



Hình 33 Giao diện cho Interpreter hủy lịch hẹn

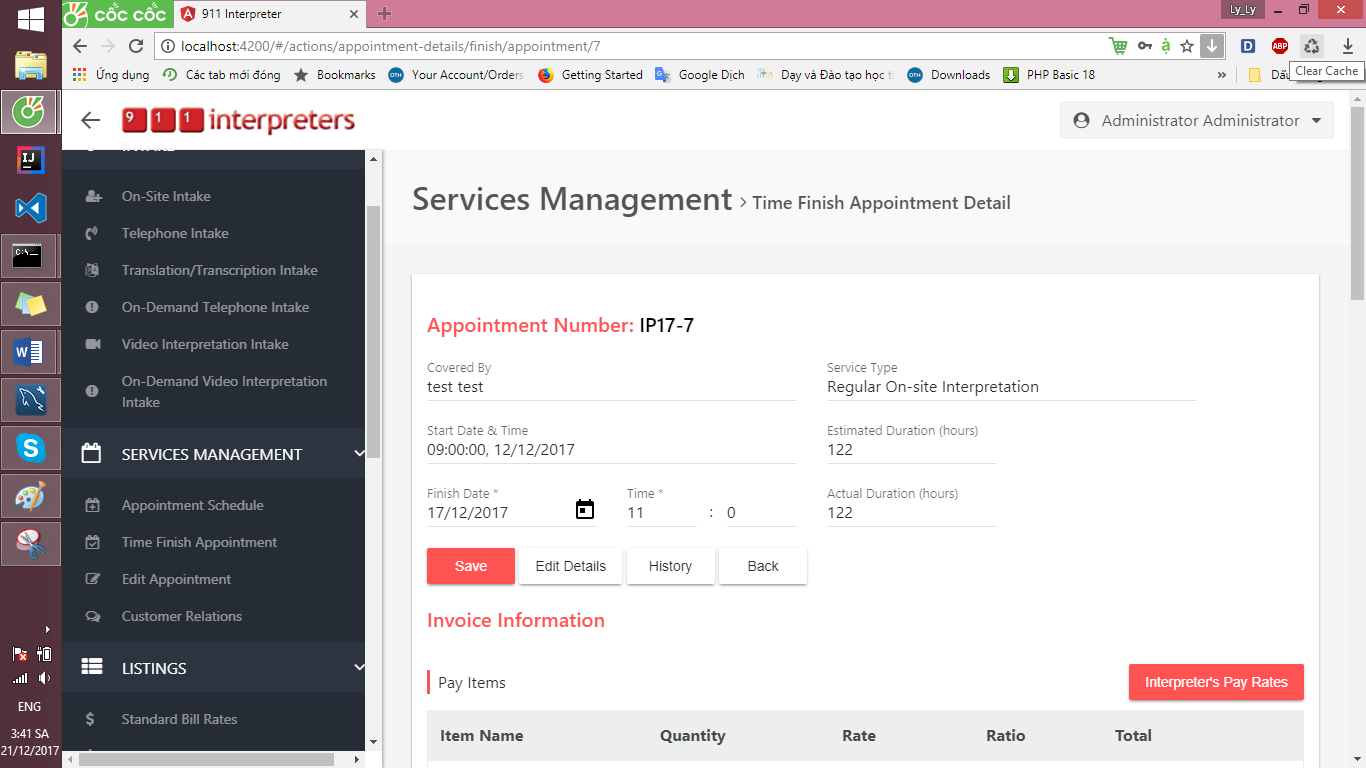
Khi Requester hủy lịch hẹn thì sẽ hiện ra giao diện như Hình 34 bên dưới.



Hình 34 Giao diện cho Requester hủy lịch hẹn

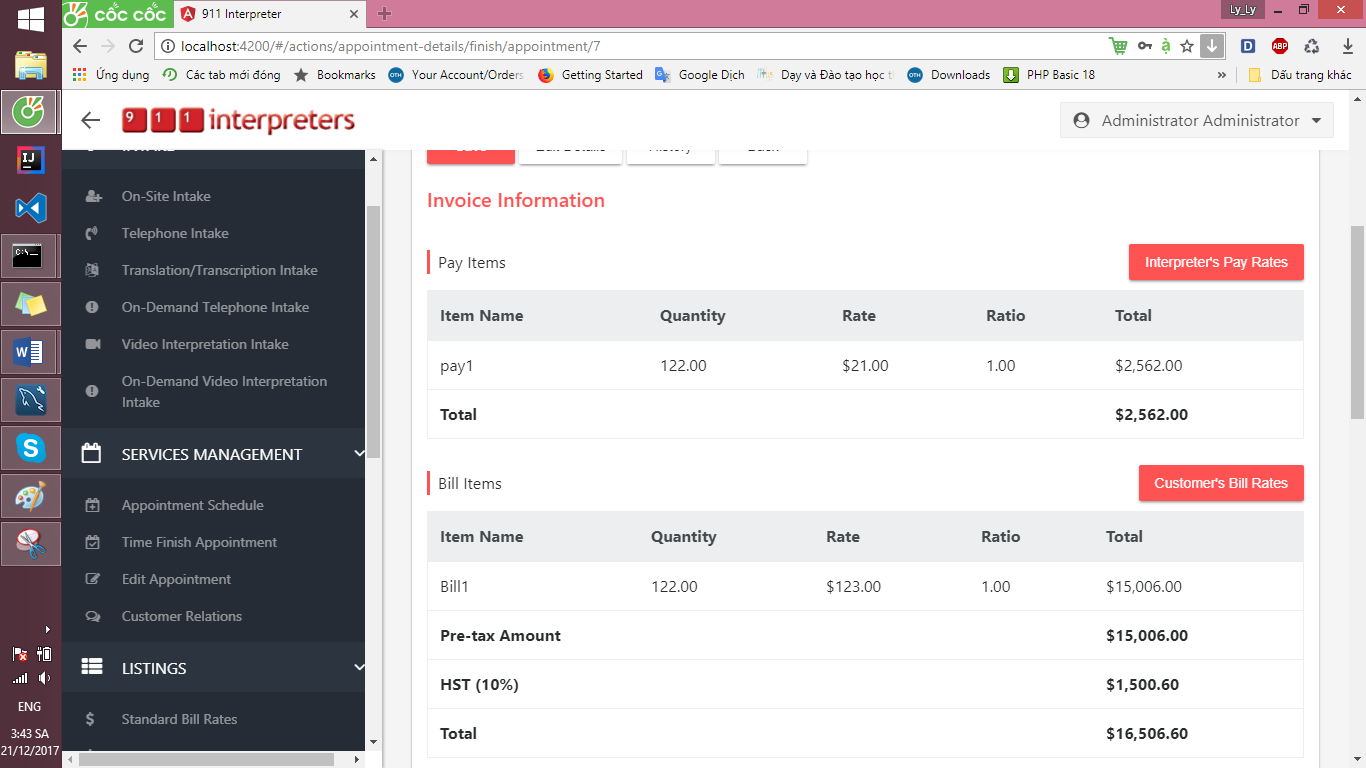
Hình 35 bên dưới là giao diện cho kết thúc lịch hẹn. Người dùng nhập vào ngày và thời gian kết thúc lịch hẹn và nhấn nút “Save” để kế thúc lịch hẹn.

Trước khi kết thúc lịch hẹn, người dùng có thể nhấn nút “Edit Details” để sửa một số thông tin của lịch hẹn, thêm Pay rate, bill rate nếu cần thiết. Hoặc xem lịch sử cập nhật lịch hẹn.



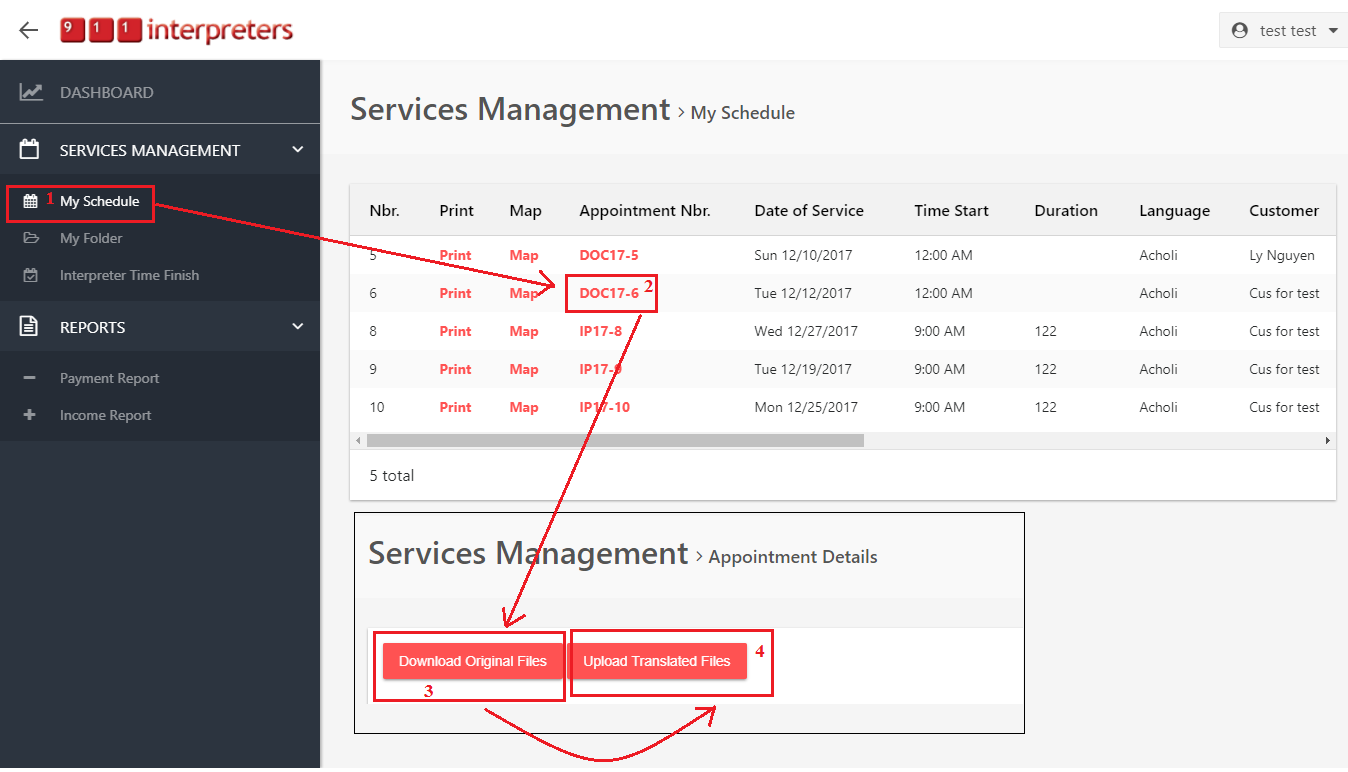
Hình 35 Giao diện cho kết thúc lịch hẹn

Sau khi kết thúc lịch hẹn, thông tin hóa đơn của lịch hẹn sẽ xuất hiện trên màn hình như Hình 36 bên dưới.



Hình 36 Giao diện thông tin hóa đơn của lịch hẹn

### Giao diện cho quản lý lịch hẹn của Interpreter



Hình 37 Giao diện cho Interpreter thực hiện một yêu cầu dịch tài liệu

Hình 37 biểu hiện một trường hợp sử dụng của Interpreter.

Mục đích của luồng này là Interpreter lấy được file tài liệu cần tải và sau đó tải lên file tài liệu đã được dịch.

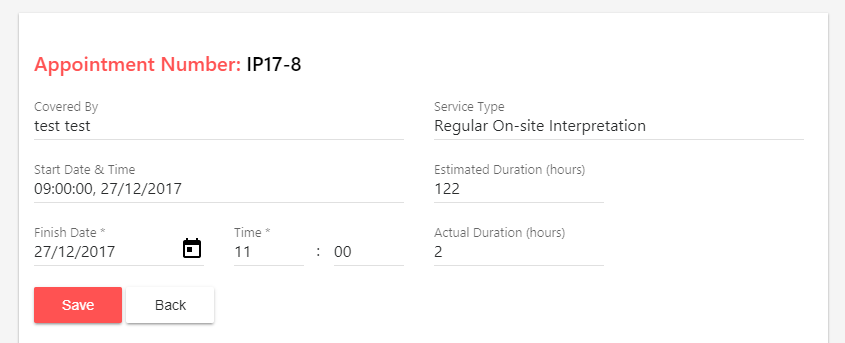
Bước một, interpreter vào My Schedule của mình như hình trên là nút kí hiệu 1. Sau đó danh sách các lịch hẹn/yêu cầu của interpreter sẽ hiện ra.

Bước 2, interpreter chọn lịch hẹn muốn xem, bằng cách nhấp vào mã cuộc hẹn như nút kí hiệu 2. Sau đó chi tiết của lịch hẹn sẽ hiện ra.

Ở hình trên, loại lịch hẹn là yêu cầu dịch tài liệu. Interpreter sẽ nhấp vào nút “Download Original files” để tải tài liệu cần dịch, như nút kí hiệu 3.

Sau khi dịch tài liệu xong, Interpreter sẽ tải lên tài liệu đã được dịch bằng cách nhấp vào nút “Upload Translated Files”.

Hình 38 bên dưới là giao diện kết thúc lịch hẹn của Interpreter. Khác với admin, Interpreter không được phép sửa lịch hẹn.



Hình 38 Giao diện kết thúc lịch hẹn của Interpreter

**KẾT CHƯƠNG:**

*Chương này, người thực hiện đồ án đã trình bày tổng quan hệ thống, thiết kế database, thiết kế giao diện người dùng.*

*Ở chương tiếp theo, người thực hiện đồ án sẽ trình bày triển khai và cài đặt hệ thống và kiểm thử.*

# Chương 5 Triển Khai, Cài Đặt Và Kiểm Thử

*Ở chương này, người thực hiện đồ án sẽ trình bày quá trình triển khai, cài đặt và kiểm thử hệ thống.*

## Triển khai và Cài đặt hệ thống

### Triển khai và cài đặt

Yêu cầu với computer để build client và server:

* Java environment
* Install Node.js[[5]](#footnote-5)
* Install Yarn[[6]](#footnote-6)

Đầu tiên, build client được một loạt file .html, .js, .css

build server thì được 1 file ..-SNAPSHOT.WAR

Do Spring Boot tích hợp sẵn tomcat nên chúng ta chỉ cần một server apache để deploy ứng dụng.

Đôi chút về yarn :

Yarn là trình quản lý package cho mã nguồn. Nó cho phép sử dụng và chia sẻ mã nguồn với các nhà phát triển khác từ khắp nơi trên thế giới. Yarn thực hiện việc này một cách nhanh chóng, an toàn và đáng tin cậy.

Yarn cho phép sử dụng các giải pháp của các nhà phát triển khác cho các vấn đề khác nhau, giúp bạn phát triển phần mềm của mình dễ dàng hơn.

Mã nguồn được chia sẻ thông qua **package** (đôi khi được gọi là module). Package chứa tất cả các mã được chia sẻ cũng như một package.json mô tả package đó.

### Yêu cầu về dữ liệu

Khách hàng cần import một lượng dữ liệu vào cơ sở dữ liệu. Đó là dữ liệu của Customer, department, requester, interpreter dưới dang file excel. Yêu cầu cần có một service thực thi yêu cầu này. Sau khi import data vào cơ sở dữ liệu, hệ thống đã có một số lượng lớn người dùng.

## 2 Kiểm thử

Em đã tiến hành xây dựng tài liệu test case để test chức năng cho hệ thống.

Kết quả test:

Tổng số test case: 100

Số test case pass: 97

Số test case fail: 3

Kiểm tra tốc độ pending của ứng dụng trên nền web: tốc độ pending của hệ thống khoảng 5000ms đến 2000ms

Kiểm tra tính Responsive trên các thiết bị màn hình PC, laptop, smartphone, tablet: ứng dựng có khả năng responsive trên tất cả các thiết bị trên.

# Chương 6 Tổng Kết Và Hướng Phát Triển

## Kết quả mà sản phẩm đạt được



Hình 39 Giao diện sau đăng nhập của Admin

Sản phẩm tạo ra đáp ứng được các nhu cầu quản lý khách hàng, nhân viên, dịch vụ của doanh nghiệp. Giúp giảm rất nhiều chi phí cho dịch vụ, mang lại lợi ích cho cả khách hàng và doanh nghiệp.

**Nhược điểm:**

Một số chức năng của người dùng chưa hoàn chỉnh như thay đổi password, cập nhật profile.

Chức năng gửi mail còn hạn chế là nếu quá trình gửi mail thất bại do timeout hay yếu tố nào đó thì mail không được gửi lại.

Do hệ thống còn nhỏ nên chưa lột tả được hết kiến trúc cũng như lợi ích của kiến trúc microservice mang lại.

## Kết quả mà bản thân đạt được

Về công nghệ:

Tìm hiểu được nhiều kiến thức về Spring Boot framework, các công cụ phát triển web kèm theo.

Hiểu biết về kiến trúc microservice, và kiến trúc khác như kiến trúc monolithic.

Về kĩ năng:

Rèn luyện kĩ năng xây dựng và phát triển phần mềm, làm chủ công nghệ và dự án.

## Hướng phát triển

Với sự phát triển mạnh mẽ của ngành công nghiệp di động, hầu hết mọi người đều sử dụng smartphone. Với mong muốn người dùng có thể dễ dàng sử dụng ứng dụng mọi lúc mọi nơi.

Hướng phát triển mong muốn là phát triển một ứng dụng di động cung cấp các dịch vụ như ứng dụng web hiện tại. Ứng dụng di động còn giúp người dùng linh hoạt có thể đặt nhắc nhở, hoặc tự động nhắc nhở khi đến thời gian thực hiện dịch vụ, v.v.

Thứ hai, mong muốn hệ thống có thể sử dụng với nhiều người dùng hơn nữa. Ví dụ cũng như Customer, Interpreter có các cấp quản lý khi Interpreter là một tổ chức/công ty phiên dịch khác.

Thứ ba là mong muốn hệ thống hoạt động linh hoạt hơn trong việc tạo tài khoản người dùng cho Khách hàng. Dụ như cho khách hàng tự đăng kí tài khoản người dùng. Nhưng phải khắc phục được nhược điểm đảm bảo tính tin cậy của thông tin người dùng đăng kí.

# Tài Liệu Tham Khảo

1. Java 8 in Action, tác giả Alan Mycroft, Mario Fusco, xuất bản năm 2014
2. Learning Spring Boot, tác giả Greg L. Turnquist, xuất bản năm 2014
3. Spring Boot In Action, tác giả Craig Walls, xuất bản năm 2016
4. The Jhipster Mini-Book, tác giả Matt Raible, xuất bản năm 2016
5. <http://www.jhipster.tech>
6. <https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/>

1. Mô-hình-hướng-dịch-vụ: service-oriented architectures (SOA) [↑](#footnote-ref-1)
2. Component: chỉ một thành phần cấu thành nên một đối tượng nào đó. [↑](#footnote-ref-2)
3. dịch vụ đăng ký : Service Registry [↑](#footnote-ref-3)
4. https://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html [↑](#footnote-ref-4)
5. https://nodejs.org/en/ [↑](#footnote-ref-5)
6. https://yarnpkg.com/en/docs/getting-started [↑](#footnote-ref-6)