**NGHIÊM XUÂN HIẾU - NGUYỄN VĂN VƯƠNG**

***Đà Lạt, 2019***

**KHOÁ LUẬN TỐT NGHIỆP KỸ SƯ CÔNG NGHỆ THÔNG TIN**

**XÂY DỰNG HỆ THỐNG QUẢN LÝ KHOA CNTT TRƯỜNG ĐẠI HỌC ĐÀ LẠT**

**(phân hệ số hoá và rút trích công văn)**

**TRƯỜNG ĐẠI HỌC ĐÀ LẠT**

**KHOA CÔNG NGHỆ THÔNG TIN**

**XÂY DỰNG HỆ THỐNG QUẢN LÝ KHOA CNTT TRƯỜNG ĐẠI HỌC ĐÀ LẠT**

**(phân hệ số hoá và rút trích công văn)**

**NGHIÊM XUÂN HIẾU – 1510198**

**NGUYỄN VĂN VƯƠNG – 1510289**

**TRƯỜNG ĐẠI HỌC ĐÀ LẠT**

**KHOA CÔNG NGHỆ THÔNG TIN**

***Đà Lạt, 2019***

**GIÁO VIÊN HƯỚNG DẪN**

**THS. NGUYỄN MINH HIỆP**

**KHOÁ LUẬN TỐT NGHIỆP KỸ SƯ CÔNG NGHỆ THÔNG TIN**

**XÂY DỰNG HỆ THỐNG QUẢN LÝ KHOA CNTT, TRƯỜNG ĐẠI HỌC ĐÀ LẠT**

**(phân hệ số hoá và rút trích công văn)**

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

Đà Lạt, ngày 06 tháng 12 năm 2019

Giáo viên hướng dẫn

**NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN 1**

Khoá luận đáp ứng yêu cầu của Khoá luận kỹ sư Công nghệ thông tin.

Đà Lạt, ngày 06 tháng 12 năm 2019

Giáo viên hướng dẫn

**NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN 2**

Khoá luận đáp ứng yêu cầu của Khoá luận kỹ sư Công nghệ thông tin.

Đà Lạt, ngày 06 tháng 12 năm 2019

Giáo viên hướng dẫn

**LỜI CAM ĐOAN**

Chúng em xin cam đoan đây là công trình nghiên cứu của riêng chúng em với sự hỗ trợ của một số thư viện và công cụ mã nguồn mở như: Tesseract của Google (trong việc số hoá hình ảnh thành ký tự số), Jwts (tạo token xác thực người dùng). Các nội dung, tài liệu và kết quả nêu trong luận văn là trung thực và chưa được công bố trong các công trình khác. Nếu không đúng như đã nếu trên, chúng em xin hoàn toàn chịu trách nhiệm về đề tài của mình.

Người cam đoan

Nghiêm Xuân Hiếu Nguyễn Văn Vương

**LỜI CẢM ƠN**

Khoá luận là cơ hội rất tốt để cho chúng em có thể vận dụng các kiến thức chuyên ngành vào dự án thực tế. Qua việc thực hiện đề tài, chúng em có thể ôn lại những kiến thức mà bản thân có được về chuyên ngành. Đồng thời, chúng em cũng nhận ra những lỗ hổng về kiến thức chuyên ngành để biết bản thân đang ở đâu và phải đầu tư cái gì cho tương lai.

Cho đến thời điểm này, chúng em đã hoàn thành khoá luận chuyên ngành cách tốt đẹp. Để có được thành quả này, chúng em đã đầu tư mọi khả năng và thời giờ để thực hiện, Tuy nhiên, chúng em không thể phủ nhận những sự giúp đỡ đến từ Khoa Công nghệ Thông tin, trường Đại học Đà Lạt, các bạn bè và cách đặc biệt là thầy Nguyễn Minh Hiệp – giảng viên hướng dẫn của chúng em.

Trước tiên, chúng em xin dành lời cảm ơn tới Ban lãnh đạo và các thầy cô trong Khoa Công nghệ Thông tin. Các thầy cô đã tạo điều kiện cho chúng em có được cơ hội đem kiến thức vào thực tế qua các đề tài khoá luận thiết thực và giá trị. Cũng không quên các thầy cô đã xây dựng vốn kiến thức nền tảng để chúng em có thể vận dụng và thực hiện khoá luận một cách dễ dàng và thành công.

Với lòng biết ơn sâu sắc, chúng em xin dành lời cảm ơn chân thành nhất cho thầy Nguyễn Minh Hiệp – giảng viên hướng dẫn và các quý thầy cô phản biện. Qua thầy và các thầy cô phản biện, chúng em đã có được những kiến thức mới và các hướng giải quyết tối ưu cho đề tài. Nhờ thầy và các thầy phản biện, chúng em dễ dàng xây dựng và phát triển đề tài theo hướng tích cực và ngày càng hoàn thiện hơn.

Cuối cùng, chúng em một lần nữa xin chân thành cảm ơn Ban lãnh đạo, các thầy cô giáo Khoa Công nghệ Thông tin, trường Đại học Đà Lạt, giảng viên hướng dẫn – thầy Nguyễn Minh Hiệp, các thầy cô phản biện và mọi người. Cầu chúc ban lãnh đạo, các thầy cô và mọi người luôn bình an, vui vẻ và hạnh phúc.

Đà Lạt, ngày 06 tháng 12 năm 2019

Trường Đại học Đà Lạt

**Khoa Công Nghệ Thông Tin**

**ĐỀ CƯƠNG NGHIÊN CỨU KHOÁ LUẬN TỐT NGHIỆP**

* **Tên Đề Tài:** Xây dựng hệ thống quản lý khoa CNTT, Trường Đại Học Đà Lạt (phân hệ số hoá và rút trích công văn).
* **Chuyên ngành:** kỹ thuật phần mềm.
* **Sinh viên thực hiện:** Nghiêm Xuân Hiếu – 1510198.

Nguyễn Văn Vương – 1510289.

* **Khoá:** 2015 - 2020
* **Giáo viên hướng dẫn:** Nguyễn Minh Hiệp.

1. **Mục tiêu đề tài:**

* Đảm bảo tính toàn vẹn của dữ liệu.
* Phục vụ cho công tác quản lý, khai thác, thống kê, phân tích công văn.
* Giao diện thân thiện cho người sử dụng.
* Sử dụng RESTful API cho back-end, tạo sự linh hoạt khi triển khai và phát triển lâu dài.
* Bám sát với yêu cầu thực tiễn.
* Loại bỏ tình trạng công văn trùng lặp, giải quyết chồng chéo.
* Nâng cao được các kỹ năng tìm kiếm, tổng hợp tài liệu; nghiên cứu, sử dụng các công nghệ, kỹ thuật; trao dồi và nâng cao khả năng trình bày…
* Đóng góp cho khoa một ứng dụng mang tính thực tiễn cao. Có thể áp dụng triển khai tại Khoa Công nghệ Thông tin.
* Ứng dụng các kiến thức đã tích lũy được trong quá trình học tập.
* Giúp sinh viên hiểu rõ hơn về các công nghệ mới như Spring Framework, React…

1. **Nội dung đề tài:** gồm 4 phần chính

* Phân tích và thiết kế hệ thống
* Giới thiệu công nghệ
* Cài đặt công cụ và phần mềm
* Triển khai hệ thống

1. **Phần mềm và công cụ sử dụng:**

Phần mềm:

* Intellij
* Visual Studio Code
* Chrome

Công cụ:

* Tesseract
* Microsoft SQL Server
* Postman

Công nghệ:

* Spring Framework
* Swagger
* React
* Ant Design
* Anxious

1. **Dự kiến kết quả:**

* Hoàn thành đúng thời gian quy định.
* Hệ thống có đầy đủ chức năng cơ bản và nâng cao, hoạt động khá tốt.

1. **Tài liệu tham khảo chính:**

* Chủ yếu là tài liệu trên mạng chính gốc bằng tiếng anh. (thường được lấy từ trang chủ của công nghệ) mà chúng em sử dụng.
* Các dự án mẫu

Đà Lạt, ngày 06 tháng 12 năm 2019

**Giáo viên hướng dẫn SV Thực hiện**

**Trưởng khoa Tổ trưởng Bộ môn**

**MỤC LỤC**

[**MỞ ĐẦU** 16](#_Toc26519457)

[**Chương 1: TỔNG QUAN** 18](#_Toc26519458)

[**1.1.** **Giới thiệu kỹ thuật nhận dạng ký tự quang học** 18](#_Toc26519459)

[**1.2.** **Đánh giá về việc nhận dạng ký tự quang học hiện nay (OCR[1])** 20](#_Toc26519460)

[**Chương 2: NỘI DUNG** 22](#_Toc26519461)

[**2.1.** **Phân tích và thiết kế hệ thống** 22](#_Toc26519462)

[**2.1.1.** **Phân tích** 22](#_Toc26519463)

[**2.1.2.** **Thiết kế** 25](#_Toc26519464)

[**2.2.** **Giới thiệu công nghệ** 30](#_Toc26519465)

[**2.2.1.** **Spring Framework** 30](#_Toc26519466)

[**2.2.2.** **Swagger** 40](#_Toc26519467)

[**2.2.3.** **React** 43](#_Toc26519468)

[**2.2.4.** **Ant Design** 50](#_Toc26519469)

[**2.2.5.** **Axious** 51](#_Toc26519470)

[**2.3.** **Cài đặt các công cụ và phần mềm** 52](#_Toc26519471)

[**2.3.1.** **Yêu cầu tối thiểu** 52](#_Toc26519472)

[**2.3.2.** **Thư viện và môi trường cho java** 53](#_Toc26519473)

[**2.3.3.** **Git** 57](#_Toc26519474)

[**2.3.4.** **Microsoft SQL Server** 60](#_Toc26519475)

[**2.3.5.** **Node.js** 66](#_Toc26519476)

[**2.3.6.** **Yarn** 71](#_Toc26519477)

[**2.3.7.** **Intellij** 73](#_Toc26519478)

[**2.3.8.** **Visual Studio Code** 78](#_Toc26519479)

[**2.4.** **Triển khai hệ thống** 80](#_Toc26519480)

[**2.4.1.** **Triển khai trên từng module nhỏ** 80](#_Toc26519481)

[**2.4.2.** **Giao diện hệ thống** 82](#_Toc26519482)

[**2.4.3.** **Các chức năng chính** 83](#_Toc26519483)

[**Chương 3: KẾT QUẢ** 85](#_Toc26519484)

[**KẾT LUẬN** 86](#_Toc26519485)

[**HƯỚNG PHÁT TRIỂN** 88](#_Toc26519486)

[**DANH MỤC TÀI LIỆU THAM KHẢO** 89](#_Toc26519487)

**DANH MỤC HÌNH VẼ**

[Hình 1. Các bảng dữ liệu cho thực thể đối tượng 33](#_Toc26523663)

[Hình 2. Sơ đồ mô tả tổng quát chức năng hệ thống 33](#_Toc26523664)

[Hình 3. Cách thức hoạt động mô hình Front Controller 36](#_Toc26523665)

[Hình 4. Cách thức hoạt động mô hình MVC 37](#_Toc26523666)

[Hình 5. Luồng hoạt động Server 38](#_Toc26523667)

[Hình 6. Cách thức hoạt động Filter 40](#_Toc26523668)

[Hình 7. Ví dụ metadata - Swagger 45](#_Toc26523669)

[Hình 8. Ví dụ Servers - Swagger 46](#_Toc26523670)

[Hình 9. Ví dụ Paths - Swagger 46](#_Toc26523671)

[Hình 10. Ví dụ Schema - Swagger 47](#_Toc26523672)

[Hình 11. Mô tả cách bắt sư kiện trong ReactJS 49](#_Toc26523673)

[Hình 12. Sơ đồ cơ bản một website ReactJS cơ bản 50](#_Toc26523674)

[Hình 13. Ứng dụng trước và sau khi sử dụng Redux 51](#_Toc26523675)

[Hình 14. Mô tả cách hoạt động của Redux 52](#_Toc26523676)

[Hình 15. Cách sơ đồ hoạt động của redux sau khi tích hợp middleware 53](#_Toc26523677)

[Hình 16. Trang đăng nhập được tạo bằng Ant Design 55](#_Toc26523678)

[Hình 17. Một đoạn code mô tả cách thức sử dụng của thư viện Anxious 56](#_Toc26523679)

[Hình 18. Mở DirectX Diagnostic Tool để xem cấu hình mấy tính 57](#_Toc26523680)

[Hình 19. Thông số hệ điều hành 57](#_Toc26523681)

[Hình 20. Chọn file cài đặt JDK tương ứng với hệ điều hành 58](#_Toc26523682)

[Hình 21. Cửa sổ cài đặt 58](#_Toc26523683)

[Hình 22. Kiểm tra tính đầy đủ của packed 59](#_Toc26523684)

[Hình 23. Quá trình cài đặt 59](#_Toc26523685)

[Hình 24. Xác nhận đường dẫn cài đặt 60](#_Toc26523686)

[Hình 25. Quá trình cài đặt 60](#_Toc26523687)

[Hình 26. Hoàn cài đặt JDK 61](#_Toc26523688)

[Hình 27. Chọn file cài đặt tương ứng với hệ điều hành 61](#_Toc26523689)

[Hình 28. Xác nhận bản quyền 62](#_Toc26523690)

[Hình 29. Đường dẫn lưu trữ phần mềm cài đặt 62](#_Toc26523691)

[Hình 30. Các tính năng của Git 63](#_Toc26523692)

[Hình 31. Quá trình cài đặt Git 63](#_Toc26523693)

[Hình 32. Hoàn tất cài đặt Git 64](#_Toc26523694)

[Hình 33. Cài đặt Microsoft SQL Server 64](#_Toc26523695)

[Hình 34. Chọn Basic để tiến hành cài đặt 65](#_Toc26523696)

[Hình 35. Chấp nhận điều khoản 65](#_Toc26523697)

[Hình 36. Chọn “Install” để cài đặt 66](#_Toc26523698)

[Hình 38. Hoàn tất quá trình cài đặt SQL Server 67](#_Toc26523700)

[Hình 39. Cài đặt bộ quản lý cho SQL Server 67](#_Toc26523701)

[Hình 40. Chọn cài đặt 68](#_Toc26523702)

[Hình 41. Quá trình cài đặt diễn ra 68](#_Toc26523703)

[Hình 42. Quá trình cài đặt Bộ quản lý SQL Server hoàn tất 69](#_Toc26523704)

[Hình 43. Chọn Database.sql để tiến hành khôi phục cơ sở dữ liệu 69](#_Toc26523705)

[Hình 44. Quá trình tạo Schema và dữ liệu 70](#_Toc26523706)

[Hình 45. Chọn file cài đặt tương ứng với hệ điều hành 70](#_Toc26523707)

[Hình 46. Cửa sổ cài đặt 71](#_Toc26523708)

[Hình 47. Đọc và đồng ý với các điều khoản của nhà phát triển 72](#_Toc26523709)

[Hình 48. Xác nhận đường dẫn lưu phần mềm 73](#_Toc26523710)

[Hình 49. Quá trình cài đặt Node.js 74](#_Toc26523711)

[Hình 50. Hoàn tất quá trình cài đặt Node.js 74](#_Toc26523712)

[Hình 51. Chọn file cài đặt yarn 75](#_Toc26523713)

[Hình 52. Chọn Next để cài đặt 75](#_Toc26523714)

[Hình 53. Đọc và chấp nhận điều khoản của nhà phát triển Yarn 76](#_Toc26523715)

[Hình 54. Xác nhận nơi lưu trữ Yarn trong máy 77](#_Toc26523716)

[Hình 55. Hoàn tất quá trình cài đặt Yarn 77](#_Toc26523717)

[Hình 57. Chọn Next để tiếp tục cài đặt 78](#_Toc26523719)

[Hình 58. Chọn Next để tiếp tục cài đặt Intellij 79](#_Toc26523720)

[Hình 59. Chọn đường dẫn cài đặt Intellij 80](#_Toc26523721)

[Hình 60. Tuỳ chọn cài đặt 81](#_Toc26523722)

[Hình 61. Quá trình cài đặt diễn ra 81](#_Toc26523723)

[Hình 62. Hoàn tất quá trình cài đặt Intellij 82](#_Toc26523724)

[Hình 63. Chọn file cài đặt VSCode để tiến hành cài đặt 82](#_Toc26523725)

[Hình 64. Tuỳ chọn cài đặt cho VSCode 83](#_Toc26523726)

[Hình 65. Hoàn tất quá trình cài đặt VSCode 84](#_Toc26523727)

[Hình 66. Giao diện quản lý API - Swagger 85](#_Toc26523728)

[Hình 67. Kiểm tra API - Endpoint lấy danh sách công văn 85](#_Toc26523729)

[Hình 68. Giao diện hệ thốn 86](#_Toc26523730)

[Hình 69.Giao diện quản lý người dùng 86](#_Toc26523731)

[Hình 70.Giao diện quản lý công văn 86](#_Toc26523732)

[Hình 71.Giao diện quản lý loại công văn 87](#_Toc26523733)

[Hình 72.Giao diện quản lý lĩnh vực 87](#_Toc26523734)

[Hình 73.Giao diện chức năng số hoá công văn 87](#_Toc26523735)

[Hình 74. Kết quả tìm kiếm từ khoá “a” 88](#_Toc26523736)

[Hình 75. Giao diện tìm kiếm 88](#_Toc26523737)

**DANH MỤC CHỮ VIẾT TẮT**

[1] OCR -Optical Character Recognition: kỹ thuật nhận dạng ký tự quang học, chuyển đổi hình ảnh của chữ đánh máy hoặc chữ viết tay thành ký tự số.

[2] BCN – Ban Chủ Nhiệm: là trưởng khoa công nghệ thông tin.

[3] MVC – Model View Controller: là mô hình xây dựng hệ thống thường thấy hiện nay.

[4] POJO – Plan Old Java Object: nói tới các đối tượng Java thuần khiết ( chỉ bao gồm các trường, các phương thức get(), set())

[5] API – Application Programming Interface: giao diện lập trình ứng dụng tạo nên các phương thức kết nối tới thư viện và ứng dụng.

[6] HTTP – HyperText Transfer Protocol: giao thức truyền tải siêu văn bản được sử dụng để truyển dư liệu giữa Server và Browser.

[7] JDK – Java Development Kit: Bộ cài đặt lỗi cho Java, cho phép lâp trình viên tạo ra các chương trình Java

[8] SQL – Structured Query Language: ngôn ngư truy vấn để truy cập hệ thống cơ sở dữ liệu quan hệ.

**TÓM TẮT KHOÁ LUẬN**

* Đề tài mà chúng em nghiên cứu gồm các vấn đề chính sau:
  + Số hoá công văn (chuyển công văn dạng hình ảnh thành văn bản)
  + Phân loại
  + Rút trích thông tin
  + Tìm kiếm từ khoá nâng cao
* Hướng tiếp cận:
  + Chúng em tìm kiếm các công cụ “nhận dạng ký tự quang học” thì hai hãng lớn trong lĩnh vực này là ABBYY và Tesseract của Google. Vì Tesseract là công cụ mã nguồn mở và miễn phí nên chúng em đã quyết định ứng dụng nó vào đề tài khoá luận.
  + Với sự hướng dẫn nhiệt tình của thầy ThS. Nguyễn Minh Hiệp, chúng em đã tìm ra giải pháp cho các vấn đề còn lại bằng cách sử dụng kiến thức đã học tại trường và mô hình thuật toán Naïve Bayes. Em xin trình bày chi tiết các giải pháp đó ở phần nội dung.
* Cách giải quyết
  + Dùng công cụ mã nguồn mở Tesseract của Google để giải quyết bài toán về số hoá.
  + Dùng mô hình thuật toán Naive Bayes tính phân phối xác suất để giải quyết bài toán phân loại
  + Dùng biểu thức chính quy (Regex) để rút trích tự động thông tin
  + Tách từ ở mỗi công văn khi thêm vào, sau khi loại bỏ các từ phổ biến, chúng em tạo ra một từ điển với từ khoá là khoá (key), số dòng và công văn là giá trị (value) để tạo ra hệ thống tìm kiếm theo chỉ mục.
* Kết quả đạt được

Hệ thống chúng em đã gần như hoàn thiện và đã thử nghiệm trên khoá, các chức năng hoạt động tốt. Sau báo cáo tiến độ lần 2, thầy cô đã hướng dẫn chúng em để cải thiện chức năng tìm kiếm mà em trình bày như trên.

# **MỞ ĐẦU**

Trải qua ba cuộc cách mạng công nghiệp, nhân loại đã thu được những thành tựu rực rỡ thay đổi hoàn toàn cuộc sống của con người. Khoa học kỹ thuật được áp dụng dần dần thay thế cho sức lao động của con người. Tới nay, thời đại của nền **Công nghiệp 4.0** lên ngôi, công nghệ thông tin đang ngày càng gần gũi hơn với con người. Trí tuệ nhân tạo (AI), vạn vật kết nối (IOT) và dữ liệu lớn (BIGDATA) là những yêu tố cốt lõi và đang là những mối quan tâm hàng đầu. Tới thời điểm hiện nay nhân loại cũng đã gặt hái được rất nhiều thành công trong nền công nghiệp 4.0 bác sỹ AI, trợ lý ảo thông minh, nhà máy không nhân công, nhà thông mình… Chúng ta dễ dàng nhận thấy công nghệ hỗ trợ đắc lực cho con người, giúp công việc trở nên đơn giản, nhanh chóng và thuận lợi hơn, mọi phần mềm ứng dụng tương ứng với từng chức năng cụ thể cũng ra đời theo nhu cầu của từng ngành nghề. Việc quản lý công văn ở khoa “Công nghệ Thông tin” mà chúng em đang học còn khá thủ công, văn thư nhận công văn đến qua hộp thư điện tử và lưu chúng vào các thư mục trong máy tính, giả sử khi văn thư muốn tìm một công văn theo một từ khoá nào đó, chúng sẽ mất rất nhiều thời gian. Nhu cầu về việc lấy văn bản từ hình ảnh đang ngày càng phát triển,công nghệ nhận dạng ký tự quanghọc (Optical Character Recognition) hay còn được gọi tắt là OCR. Đây là một côngnghệ giúp chuyển đổi hình ảnh của chữ viết tay hoặc đánh máy thành các ký tự đãđược mã hóa trong máy tính.

Việc chọn đề tài này là do chúng em muốn áp dụng công nghệ nhận dạng ký tự quang học nêu trên để chuyển công văn từ dạng hình ảnh sang dạng chữ, tự động phân loại công văn đó và rút trích các thông tin cần thiết để lưu trữ. Chúng em muốn tạo ra một hệ thống giúp khoa “Công nghệ Thông tin” nói riêng và các khoa khác nói chung đơn giản hoá việc quản lý công văn bằng cách ứng dụng công nghệ hiện có.

Hiện nay ở Việt Nam hầu như việc lưu trữ văn bản giấy tờ vẫn còn dùng cách thủ công. Việc này gây rất nhiều khó khăn trong việc lưu trữ, quản lý và bảo quản các văn bản này vì thường các văn bản giấy tờ thường rất nhiều, có thể lên tới hàng ngàn bản chỉ trong vòng một tháng. Hơn nữa những đơn vị được giao việc quản lý các văn bản giấy tờ thường rất ít nhân lực càng gây khó khăn hơn, nếu nhiều nhân lực thì lại tốn nhiều chi phí hơn. Khó khăn càng chồng khó khăn. Chính vì thế hiện tại không còn gì tuyệt vời hơn ngoài việc áp dụng khoa học kỹ thuật vào việc quản lý công văn bằng cách lưu chúng thành các văn kiện điện tử lưu vào trong máy tính đem lại rất nhiều lợi ích: vừa giảm chi phí, vừa dễ quản lý và cũng có thể triển khai trên diện rộng giúp đỡ rất nhiều cho con người. Chính vì lí do trên, chúng em được các thầy cô phân công đề tài này với mục đích là muốn giúp các thầy cô trong và ngoài khoa dễ dàng hơn trong việc thao tác đối với các công văn cũng như là tìm kiếm công văn. Triển khai giải pháp số hoá kho dữ liệu, chuyển đổi từ tài liệu dạng giấy sang tài liệu dạng số lưu trữ trong máy tính, máy chủ hoặc các thiết bị lưu trữ nội dung số khác để đảm bảo tính toàn vẹn của dữ liệu, phục vụ cho công tác khai thác, thống kê, phân tích và dự báo trong hoạt động của khoa.

Đề tài “Quản lý công việc khoa CNTT (phân hệ) số hoá và rút trích công văn” nhằm tìm hiểu kỹ thuật số hoá thông tin từ một công văn tiếng Việt, vốn là một hình chụp từ máy Scan. Kỹ thuật số hoá bao gồm tìm hiểu các phần mềm VietOCR, ABBYY, Tesseract. Từ đó nghiên cứu cách phân loại công văn và rút trích văn bản, sau cùng là xây dựng một hệ thống tìm kiếm công văn cơ bản và nâng cao. Ở giai đoạn đầu, chúng em sẽ thử triển khai hệ thống để thử nghiệm tại khoa Công nghệ Thông tin, giai đoạn tiếp theo chúng em sẽ triển khai hệ thống trên tất cả các khoa. Đối tượng người sử dụng ở đây chủ yếu là ban chủ nhiệm khoa, văn thư và giảng viên.

Bài báo cáo chúng em sẽ chia làm ba chương để mô tả một cách chi tiết nhất về hệ thống.Trong phần đầu chúng em sẽ đưa ra một cách nhìn tổng quan nhất về đề tài mà chúng em đang tiến hành, phần tiếp theo sẽ là phần nội dung, chúng em sẽ phân tích sâu vào việc thiết kế, phân tích, các công nghệ và thư viện hỗ trợ hiện này để tìm hiểu xem chúng là gì và cách thức hoạt động như thế nào? Chương cuối, chúng em xin nói về những kết quả đã đạt được trong suốt thời gian thực hiện đề tài và cuối cùng chúng em sẽ đưa ra kết luận, cũng như đặt ra hướng phát triển cho hệ thống trong tương lai.

# **Chương 1: TỔNG QUAN**

* 1. **Giới thiệu kỹ thuật nhận dạng ký tự quang học**

Nhận dạng ký tự quang học OCR[1] là loại phần mềm máy tính được tạo ra để chuyển các hình ảnh của chữ viết tay hoặc chữ đánh máy (thường được quét bằng máy scanner) thành các văn bản tài liệu. OCR[1] được hình thành từ một lĩnh vực nghiên cứu về nhận dạng mẫu, trí tuệ nhận tạo và machine vision. Mặc dù công việc nghiên cứu học thuật vẫn tiếp tục, một phần công việc của OCR[1] đã chuyển sang ứng dụng trong thực tế với các kỹ thuật đã được chứng minh.

Nhận dạng ký tự quang học (dùng các kỹ thuật quang học chẳng hạn như gương và ống kính) và nhận dạng ký tự số (sử dụng máy quét và các thuật toán máy tính) lúc đầu được xem xét như hai lĩnh vực khác nhau. Bởi vì chỉ có rất ít các ứng dụng tồn tại với các kỹ thuật quang học thực sự, bởi vậy thuật ngữ Nhận dạng ký tự quang học được mở rộng và bao gồm luôn ý nghĩa nhận dạng ký tự số.

Đầu tiên hệ thống nhận dạng yêu cầu phải được huấn luyện với các mẫu của các ký tự cụ thể. Các hệ thống "thông minh" với độ chính xác nhận dạng cao đối với hầu hết các phông chữ hiện nay đã trở nên phổ biến. Một số hệ thống còn có khả năng tái tạo lại các định dạng của tài liệu gần giống với bản gốc bao gồm: hình ảnh, các cột, bảng biểu, các thành phần không phải là văn bản.

Hiện nay, với chữ Việt, phần mềm nhận dạng chữ Việt in VnDOCR 4.0 có khả năng nhận dạng trực tiếp các loại tài liệu được quét qua máy quét, không cần lưu trữ dưới dạng tệp ảnh trung gian. Các trang tài liệu có thể được quét và lưu trữ dưới dạng tệp tin nhiều trang. Kết quả nhận dạng được lưu trữ sang định dạng của Microsoft Word, Excel... phục vụ rất tốt nhu cầu số hóa dữ liệu.

Ngoài ra, còn có một dự án OCR[1] Tiếng Việt có tên VietOCR, được phát triển dựa trên nền tảng mã nguồn mở tesseract-ocr do Google tài trợ. VietOCR có khả năng nhận dạng chữ Việt rất tốt. Đây là một chương trình nguồn mở Java/.NET, hỗ trợ nhận dạng cho các dạng ảnh PDF, TIFF, JPEG, GIF, PNG, và BMP.

ABBYY - một hãng công nghệ hàng đầu trên thế giới về lĩnh vực Nhận dạng ký tự quang học đã tiến hành nghiên cứu và triển khai công nghệ nhận dạng Tiếng Việt vào tháng 4 năm 2009. Với công nghệ này độ chính xác trong việc nhận dạng tài liệu chữ in Tiếng Việt lên tới hơn 99% (cứ nhận dạng 100 ký tự thì có chưa đến 1 ký tự sai). Công nghệ của ABBYY chấp nhận hầu hết các định dạng ảnh đầu vào như: PDF, TIFF, JPEG, GIF, PNG, BMP, PCX, DCX,... Kết quả nhận dạng được lưu trữ dưới các định dạng MS Word, MS Excel, HTML, TXT, XML, PDF, PDF 2 lớp, trong đó định dạng PDF 2 lớp là một định dạng hoàn hảo cho việc lưu trữ và khai thác tài liệu. Với định dạng này, người đọc có thể đọc trung thực ảnh gốc nhờ lớp ảnh bên trên, các công cụ tìm kiếm có thể tìm kiếm toàn văn trên văn bản nhờ lớp text nhận dạng được bên dưới.

Tesseract là một dự án nghiên cứu tiến sĩ tại phòng thí nghiệm HP ở Bristol và đã được tích hợp vào trong các dòng máy quét dạng phẳng của hãng dưới dạng các add-on phần cứng hoặc phần mềm. Nhưng thực tế dự án này đã thất bại ngay từ trong trứng nước vì nó chỉ làm việc hiệu quả trên các tài liệu in có chất lượng tốt.

Sau đó, dự án này cùng với sự cộng tác của bộ phận máy quét HP ở bang Colorado đã đạt được một bước tiến quan trọng về độ chuẩn xác khi nhận dạng và vượt lên nhiều bộ nhận dạng OCR[1] thời đó nhưng dự án đã không thể trở thành sản phẩm hoàn chỉnh vì độ cồng kềnh và phức tạp. Sau đó, dự án được đưa về phòng thí nghiệm của HP để nghiên cứu về cách thức nén và tối ưu mã nguồn.

Dự án tập trung cải thiện hiệu năng làm việc của Tesseract dựa trên độ chính xác đã có. Dự án này được hoàn tất vào cuối năm 1994 và sau đó vào năm 1995 bộ Tesseract được gửi đi tham dự hội nghị thường niên về độ chính xác của OCR[1], vượt trội hơn hẳn so với các phần mềm OCR[1] lúc bấy giờ. Tuy nhiên, Tesseract đã không thể trở thành một sản phẩm thương mại hoàn chỉnh được và vào năm 2005, HP đã chuyển Tesseract sang mã nguồn mở và được hãng Google tài trợ.

Tesseract cho đến nay vẫn được nhiều nhà phát triển cộng tác và tiếp tục hoàn thiện. Phiên bản mới nhất của bộ nhận dạng Tesseract là phiên bản 5.0.

* 1. **Đánh giá về việc nhận dạng ký tự quang học hiện nay (OCR[1])**

Hiện nay, giải pháp nhận dạng ký tự quang học đang phát triển mạnh mẽ về mặt ứng dụng và liên tục có nhiều cải tiến mới để tăng tính ứng dụng và hiệu quả ở sản phẩm đầu ra. Đi đầu trong công nghệ OCR[1] phải kể đến hai hãng phát triển và cải tiến phần mềm nhận dạng ký tự đó là Google và ABBYY.

Google trên nền tảng Tesseract được phát triển bởi HP cho tới hiện tại là phiên bản Tesseract 5.0, sử dụng mã nguồn mở, có chất lượng nhận dạng chính xác cao, với nhiều định dạng file ảnh và có thể nhận dạng hơn 60 ngôn ngữ khác nhau.

Trong khi, ABBYY được coi là hãng tiên phong trong lĩnh vực OCR. ABBYY cho ra đời phần mềm nhận dạng ký tự quang học với tên gọi ABBYY có khả năng nhận dạng 190 ngôn ngữ. Đặc biệt, đối với ký tự La-tinh và tiếng Nga, công nghệ OCR của ABBYY có thể đạt hiệu quả nhận dạng đến 99% cho một file ảnh chất lượng tốt.

Tuy nhiên, việc nhận dạng ký tự tiếng Việt (loại hình ngôn ngữ có “dấu”) vẫn là thách thức đối với sự phát triển của công nghệ OCR trên thế giới. Hiện tại, ABBYY đang tiến hành nghiên cứu và triển khai công nghệ nhận dạng tiếng Việt với độ chính xác trên 90% cho một file ảnh có chất lượng tốt. Song mới chỉ dừng lại ở việc nhận dạng ký tự tiếng Việt được soạn thảo bằng vi tính hoặc công nghiệp in ấn. Mà chưa thể vươn tới các loại hình chữ viết tay, thậm chí hiểu quả đạt được còn rất khiêm tốn đối với các loại tài liệu cũ, đa dạng về phông chữ hoặc sử dụng kỹ thuật in lạc hậu. Ngoài ra, giải pháp OCR của ABBYY còn thiếu tính cạnh tranh bởi giá thành cao, là sản phẩm đóng gói của nước ngoài, chưa thực sự phù hợp với các loại hình tài liệu của Việt Nam, cũng như khó khăn trong việc tích hợp vào các công nghệ khác.

|  |  |
| --- | --- |
| **ABBYY** | **Tesseract** |
| Hỗ trợ hơn 192 ngôn ngữ | Hỗ trợ trên 116 ngôn ngữ và đang tăng dần |
| Có giao diện đồ họa | Không hỗ trợ giao diện đồ họa (dùng  Command Line để gõ lệnh) |
| Độ chính xác cao. | Độ chính xác cao. |
| Chi phí khá cao 130$ - 500 $ | Hoàn toàn miễn phí (mã nguồn mở), được tài trợ bởi Google. |
| Không có phiên bản dành cho lập trình viên, có API nhưng phải trả phí | Phù hợp với lập trình viên vì có thể nhúng vào các ngôn ngữ lập trình |

Chúng em sẽ chọn Tesseract vì đó là công cụ miễn phí và đủ để đáp ứng chức năng của hệ thống, vì sau này Tesseract được chuyển giao cho Google nên chúng em tin rằng nó sẽ rất manh trong tương lai.

# **Chương 2: GIỚI THIỆU CÁC CÔNG NGHỆ SỬ DỤNG**

* 1. **Phân tích và thiết kế hệ thống**
     1. **Phân tích**
        1. **Quản lý phân quyền tiếp cận công văn:**

**BCN[2] Khoa:**

* + - Người có thể nắm được hết thông tin của các loại công văn.
    - Xem thông tin chi tiết công văn.
    - Xem được các báo cáo, thống kê hàng tháng của văn thư.

**Văn thư:**

* + - Người có trách nhiệm nhiều nhất trong việc chuyển giao, quản lý, lưu trữ công văn.

**Giảng viên:**

* + - Người có quyền truy cập công văn theo quyền cho phép như : tìm kiếm, hiển thị, in ấn…
    - Ta bắt đầu phân loại người dùng bằng hai quyền chính:
    - Quyền quản trị viên (Admin)
      * Thêm người dùng.
      * Sửa thông tin người dùng.
      * Xóa người dùng.
      * Sao lưu, phục hồi dữ liệu.
    - Quyền hội viên (Member)
      * Xem công văn
      * Tạo công văn mới
      1. **Quản lý công văn**
         1. **Trình tự công văn**
  + Công văn đến từ bất kỳ nguồn nào đều phải được tập trung tại văn thư cơ quan, tổ
  + chức để làm thủ tục tiếp nhận, đăng ký. Những công văn đến không được đăng ký tại
  + văn thư, các đơn vị, cá nhân không có trách nhiệm giải quyết.
  + Văn thư tiếp nhận công văn và tiến hành lấy thông tin và nhập vào hệ thống, lưu lại.
  + Khi có phát sinh cần chỉnh sửa sẽ được thực hiện.
    - * + Kiểm tra thể thức, hình thức và kỹ thuật trình bày; ghi số, ký hiệu và ngày, tháng của công văn.
        + Đóng dấu cơ quan và dấu mức độ khẩn, mật (nếu có).
        + Đăng ký công văn đi.
        + Làm thủ tục, chuyển phát và theo dõi việc chuyển phát công văn đi.
        + Lưu công văn đi.
        1. **Dữ liệu công văn**
  + Số/Kí hiệu.
  + Ngày tháng công văn (ngày ký).
  + Ngày đi.
  + Ngày nhập sổ.
  + Tên cơ quan ban hành.
  + Tên đơn vị nhận.
  + Tên loại văn bản.
  + Trích yếu nội dung.
  + Mức độ mật.
  + Mức độ khẩn.
  + Số tờ.
  + Số bản.
  + Người ký duyệt.
  + Ghi chú.
    - * 1. **Công văn đến**
  + Phê duyệt và tiếp nhận công văn đến.
  + Phân loại công văn đến theo:
* Ngày, tháng, năm.
* Số hiệu
* Loại
* Cơ quan, phòng ban.
* Mức độ bảo mật.
* Nơi xuất phát.
  + Lưu trữ công văn.
* Sau khi phân loại công văn tiến hành lưu trữ theo các bộ tập tài liệu.
* Phân biệt bằng các màu và nhãn dán khác nhau.
* Mỗi tầng của kệ hoặc màu tập sẽ phân biệt được mức độ bảo mật của công văn đó.
  + Chuyển giao công văn
* Văn thư sẽ chuyển giao công văn tới các đơn vị phòng ban chịu trách nhiệm xử lý công văn đó.
* Văn thư có thể chuyển công văn lên BCN Khoa nếu thấy có sai sót.
  + - * 1. **Công văn đi**
  + Soạn thảo công văn đi
* Viết công văn theo mẫu loại.
* Gửi lên BCN Khoa để đóng dấu.
  + Kiểm tra thể thứ, hình thức và kỹ thuật trình bày ( ghi số, ký hiệu và ngày, tháng của công văn )
  + Đóng dấu và dấu mức độ khẩn, mật ( nếu có)
  + Đăng ký công văn đi:
* Là một hình thức lưu lại công văn đi để phòng trường hợp tra cứu, sửa đổi.
  + Trưởng khoa ký duyệt công văn đi
  + Làm thủ tục, chuyển phát và theo dõi việc chuyển công văn đi
  + Lưu công văn đi
  + Xác nhận công văn đã tới nơi gửi.
    - * 1. **Công văn nội bộ**
  + Soạn công văn nội bộ (thường do văn thư soạn thảo)
  + Đưa lên BCN[2] Khoa và các cơ quan ban ngành để bổ sung ý kiến
  + BCN[2] Khoa ký xác nhận, duyệt công văn.
  + Văn thư tiến hành lưu trữ công văn.
    - 1. **Quản lý loại công văn**
* Tự động phân loại công văn
* Quản lý công văn: tìm kiếm, sắp xếp…
* Tìm kiếm văn bản, công văn trên hệ thống để đáp ứng nhu cầu người sử dụng. Có thể thực hiện công việc tìm kiếm theo nhiều tiêu chí khác nhau.
* Tìm kiếm công văn đến, công văn đi. Cho phép tìm nhanh với từ khóa:
  + Số công văn
  + Ngày công văn.
  + Ngày nhận/gửi công văn
  + Trích yếu
  + Đơn vị người nhận/gửi
    - 1. **Quản lý lưu trữ**
  + Lưu trữ công văn đến (Công văn được lưu trữ trong các kẹp giấy.)
    - * + Lưu trữ theo ngày, tháng, năm ban hành.
        + Lưu trữ theo số hiệu.
        + Lưu trữ theo từng loại công văn.
        + Lưu trữ theo nơi gửi, nhận.
  + Lưu trữ công văn đi (Công văn được lưu trữ trong các kẹp giấy.)
    - * + (Công văn đi được lưu ít nhất 2 bản; một bản lưu tại văn thư khoa, một bản lưu trong hồ sơ.)
        + Lưu trữ trong các kẹp giấy.
        + Lưu trữ theo ngày tháng năm ban hành.
        + Lưu trữ theo số hiệu.
        + Lưu trữ theo từng loại công văn.
        + Lưu trữ theo nơi gửi, nhận.
    1. **Thiết kế**
       1. **Yêu cầu hệ thống**
  + Tính nhất quán
    - * + Cơ chế dự phòng và hồi phục dữ liệu
        + Tránh truy cập nặc danh, sử dụng quyền quản trị để quản lý và phân quyền người dùng
  + Tính sẵn sàng
    - * + Hệ thống máy chủ được duy trì hoạt động liên tục.
        + Hệ thống mạng luôn luôn ổn định
  + Tính bảo mật
    - * + Việc truy cập của người sử dụng phải được kiểm soát theo định danh và mật khẩu riêng của từng người sử dụng, người sử dụng phải được kiểm soát theo nhóm tác nghiệp đã hoạch định, quá trình truy cập vào hệ thống của người sử dụng phải được kiểm soát.
        + Hệ thống phải tạo ra nhóm tác nghiệp theo chức năng và nhiệm vụ được giao trên cơ sở của việc hạn chế mức truy cập của hệ thống.
        + 4. Đánh giá hiệu quả hệ thống
        + Hệ thống đáp ứng tốt khi số lượng người dùng và dữ liệu tăng; chỉ cần bổ sung nâng cấp server.
  + Tính khả thi của hệ thống:
    - * + Hệ thống triển khai làm 02 giai đoạn bảo đảm việc thử nghiệm thành công trên phạm vi nhỏ sau đó triển khai trên toàn bộ hệ thống.
        + Hai Giai đoạn bao gồm: Beta và Stable.

Dựa trên việc sử dụng thành công các phần mềm hỗ trợ chức năng chính của hệ thống.

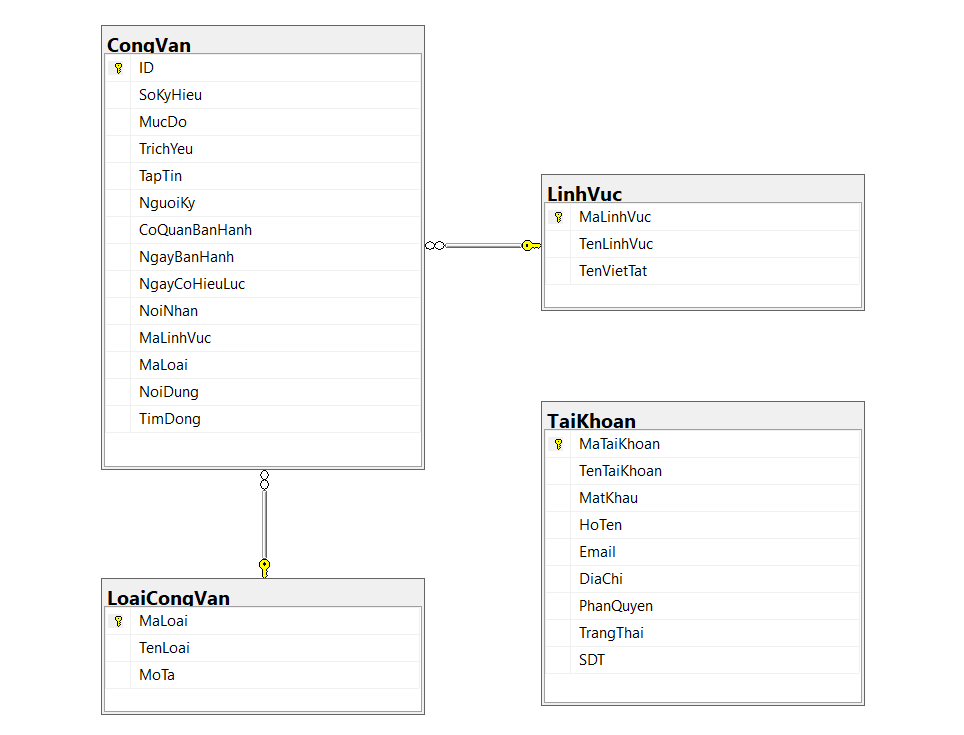
* + - 1. **Chức năng**

Đây là một hệ thống số hoá và rút trích công văn. Từ những gì đã phân tích ở trên, hệ thống yêu cầu những chức năng như sau:

* + - Đối với công văn:
      * Hiển thị danh sách công văn, chi tiết công văn được sắp xếp theo các trường dữ liệu.
      * Cho phép số hoá công văn từ hình ảnh sang dạng văn bản lưu trong máy.
      * Rút trích thông tin cần thiết của công văn
      * Tìm kiếm công văn theo loại, lĩnh vực, các trường dữ liệu trong công văn
    - Đối với loại công văn:
      * Phân loại công văn là công văn đến, công văn đi hay công văn nội bộ.
    - Đối với lĩnh vực:
      * Phân loại lĩnh vực mà công văn đó thuộc về
    - Đối với người dùng:
      * Quản lý thông tin người dùng hệ thống (quản lý tài khoản).
      * Phân quyền
      1. **Quản lý tài khoản**
  + Admin:
    - * + Thêm người dùng.
        + Sửa thông tin người dùng.
        + Xóa người dùng.
        + Phân quyền cho người dùng.
        + Sao lưu, phục hồi dữ liệu.
  + User (người dùng):
    - * + Thao tác nghiệp vụ.
        + Đổi mật khẩu.
  + Hệ thống sẽ phân cấp người dùng tìm kiếm các công văn cần thiết khi có nhu cầu:
    - * + Chỉ có admin, văn thư, lãnh đạo của cơ quan (BCN khoa) mới có thể nắm được hết thông tin
        + của các loại công văn. Khi xem thông tin chi tiết công văn cũng sẽ biết được người
        + dùng nào đã nhập liệu, chỉnh sửa cuối cùng thông tin của các công văn.
        + Khi có sự cố lỗi nhập liệu sai, admin sẽ tìm được người chỉnh sửa cuối cùng thông qua lịch sử đăng nhập.

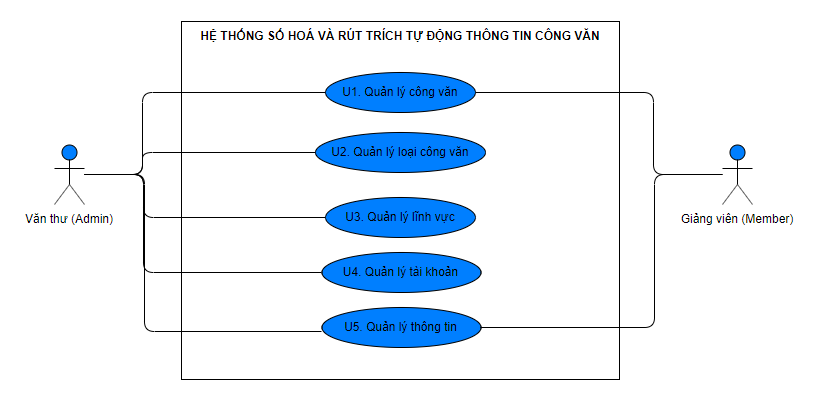
Người dùng bình thường không thể thấy được các công văn mật, công văn nội bộ trong cơ quan.

* + - 1. **Tìm kiếm**
  + Tìm kiếm văn bản, công văn trên hệ thống để đáp ứng nhu cầu người sử dụng. Có thể thực hiện công việc tìm kiếm theo nhiều tiêu chí khác nhau.
  + Tìm kiếm công văn đến, công văn đi. Cho phép tìm nhanh với từ khóa:
    - * + Số công văn
        + Ngày công văn.
        + Ngày nhận/gửi công văn
        + Trích yếu
        + Đơn vị người nhận/gửi
        + Theo loại
        + Sau khi tìm kiếm xong thì hiển thị công văn bằng giấy tờ.
  + In công văn
  + Đăng ký công văn
  + Tìm kiếm công văn
  + Thêm công văn
  + Xóa công văn
  + Sửa công văn
  + Hiển thị nội dung
    - 1. **Xác định thực thể để thiết lập bảng dữ liệu**



Hình 1. Các bảng dữ liệu cho thực thể đối tượng

* + - 1. **Sơ đồ usecase tổng quát**

****

Hình 2. Sơ đồ mô tả tổng quát chức năng hệ thống

* 1. **Giới thiệu công nghệ**
     1. **Spring Framework**
        1. **Spring Core**

- Ở phần Spring Core này đã giúp em có một kiến thức vững chắc hơn về convention code. Em bắt đầu tìm hiểu về Loose Coupling, Inversion of Control (IoC) và Dependency Injection (DI). Em xin báo cáo phần tìm hiểu của em như sau:

- Ta nói tới Dependency Inversion nghĩa là đảo ngược sự phụ thuộc. Đây là nguyên lý cuối cùng của SOLID priciples (Những nguyên lý khi viết và thiết kế code). Dependency Inversion gồm 2 nguyên tắc:

* Module cấp cao không nên phụ thuộc vào module cấp thấp mà cả hai nên phụ thuộc vào abstraction.
* Interface (abstraction) không nên phụ thuộc vào chi tiết và ngược lại (nghĩa là các class giao tiếp với nhau thông qua Interface chứ không phải thông qua implementation)

- Trong lập trình Java, khi ta muốn sử dụng một phương thức a() của một class A thì trong class B ta muốn sử dụng, đương nhiên ta phải khởi tạo class A() kia thì mới dùng được phương thức a() của nó. Nếu giả sử ta muốn thay phương thức b() thì ta sẽ phải thay đổi class B (ở đây ta xoá class A cùng phương thức a() rồi thay cái khác). Như vậy nếu trong class B, ta dùng hàng trăm phương thức từ các class khác nhau và giả sử ta phải thay đổi phương thức thì nó sẽ ảnh hưởng nghiêm trọng tới class B. Đó là lý do sinh ra các khái niệm như LooseCoupling, Dependency Injection và Inversion of Control.

- LooseCoupling: Như đã nói ở trên, việc module cấp cao không phụ thuộc vào module cấp thấp mà phụ thuộc vào abstraction. LooseCoupling nói tới việc làm nới lỏng quan hệ, phụ thuộc giữa các class với nhau thông qua việc định nghĩa một interface để mà module cấp cao sử dụng interface đó để chuyển đổi các module cấp thấp có cùng bản chất nhưng khác về mặt logic.

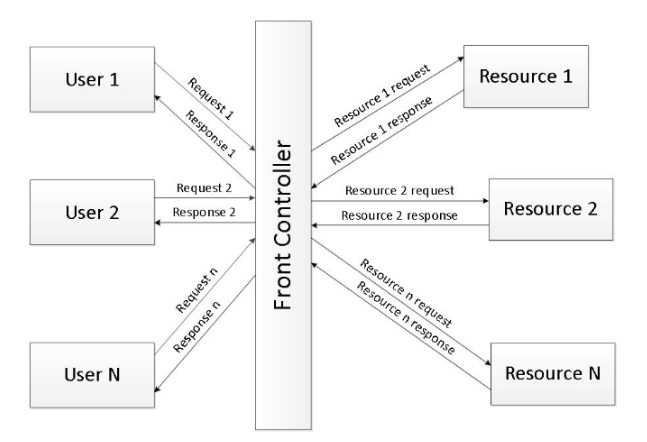
- Inversion of Control: Để hiện thực hoá nguyên lý Dependency Inversion thì ngta làm ra IoC. Đó chính xác là một design partern, một bản thiết kế mẫu. Hướng dẫn bạn phải tuân thủ theo nguyên lý trên. Và để thực hiện IoC này, chúng ta đi tiếp tới phần Dependency Injection.

- Có nhiều cách để thực hiện IoC như Service Locator, Event, Delegate và Dependency Injection là một trong những cách để thực hiện IoC. Theo đó các module cấp thấp sẽ được inject (tiêm) vào module cấp cao. Đại khái thằng DI này sử dụng interface và tiêm interface đó vào constructor của module cấp cao, hoặc tiêm vào phương thức setter(). Và nó sử dụng framework để có thể thực thi IoC (đảo ngược điều khiển) nghĩa là nó dùng một container để lưu các module cấp thấp, ứng dụng design pattern singleton để không phải khởi tạo nhiều ở module cấp cao.

* + - 1. **Spring MVC[3]**

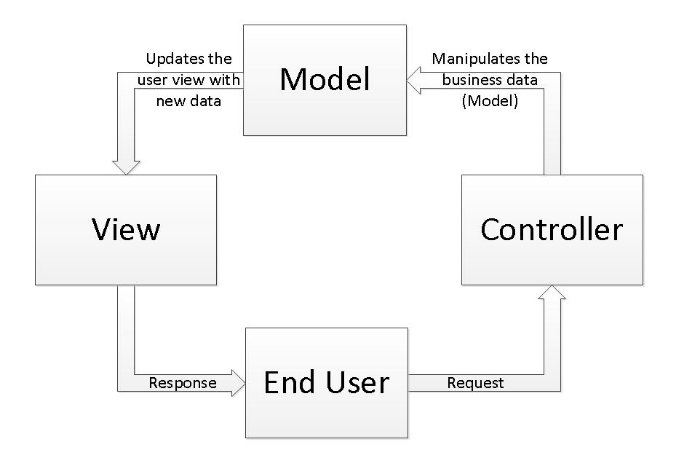
Trong Spring framework thì Spring MVC[3] này được thiết kế dựa trên hai mẫu thiết kế web khá nổi là Front Controller và MVC[3]. Nên em sẽ nói sơ về hai mẫu Front Controller và MVC trước như sau:

* **Front Controller**: đối với mẫu thiết kế này thì web được vận hành sẽ cung cấp một điểm truy cập cho tất cả các yêu cầu (request) từ client (người dùng hay khách hàng). Các yêu cầu này sẽ được bộ xử lý trung tâm, sau đó bộ xử lý này dựa trên yêu cầu (request) để ánh xạ tới tài nguyên (resource) thích hợp. Sau khi tìm được tài nguyên thì bộ xử lý trung tâm sẽ gửi lại hồi đáp (response) về lại client (người dùng hoặc khách hàng). Design pattern này cụ thể được minh hoạ như hình dưới đây.



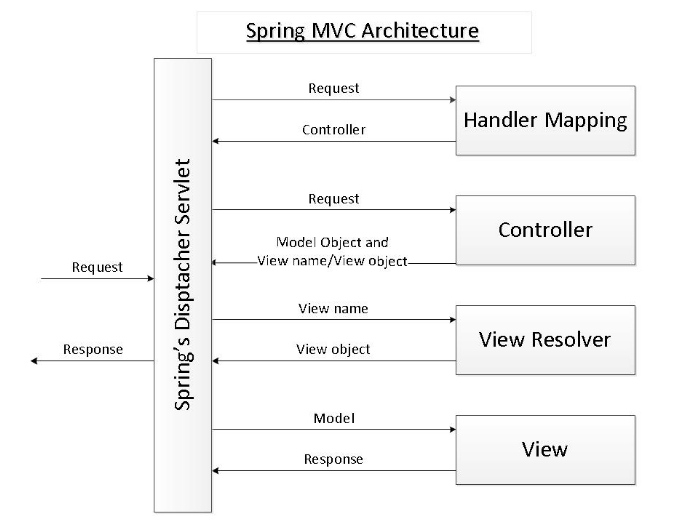
Hình 3. Cách thức hoạt động mô hình Front Controller

* **MVC[3]**: Mẫu này giúp ta xây dựng mô hình tầng ứng dụng (representation application layer) thành ba phần với nhiệm vụ khác nhau: Model, View và Controller. Khi người dùng gửi request url vào hệ thống thì sẽ qua Controller, Controller có nhiệm vụ điều hướng dựa vào thông tin trong url. Nó sẽ vào model lấy dữ liệu và gửi view cho người dùng theo yêu cầu.



Hình 4. Cách thức hoạt động mô hình MVC

* Spring MVC[3] được thiết kế và xây dựng dựa trên hai mẫu thiết kế trên. Nghĩa là tất cả các request sẽ được gửi tới một servlet có tên là DispatcherServlet – đóng vai trò như là một Front Cotroller. Khi nhận request thì DispatcherServlet này sẽ đưa tới HandlerMapping để giúp nó tìm đối tượng phù hợp. Khi tìm được đối tượng phù hợp thì DispatcherServlet sẽ yêu cầu Controller (trình điều khiển) để nó trả về đối tượng thuộc sự quản lý của nó. Đối tượng nằm trong Model và Controller sẽ lấy dữ liệu của model và trả về ModelView. Lúc này ta cần ta cần một thứ lấy ModelView và lọc trả về View. Thứ ta cần chính là View Resolver.



Hình 5. Luồng hoạt động Server

Các đối tượng:

**DispatcherServlet** trong tài liệu của Spring được xem như là Front Controller (Nghĩa là trước khi tới Controller để điều hướng) thì các request của người dùng trước trên phải qua DispatcherServlet trước. Và trong DispatcherServlet này có giao tiếp với một thành phần gọi là HandlerMapping và hỏi rằng yêu cầu của khách hàng là như vậy và yêu cầu HandlerMapping tìm ra Controller phù hợp để xử lý tiếp yêu cầu đó. Thường các yêu cầu của khách hàng là các HTTP Request.

**Model**: thường trong Spring MVC[3], ta tạo thư mục model và có khi nó còn gọi và domain (tuỳ vào “convention” của khách hàng). Mỗi model này sẽ chứa các POJO[4] (Plain Old Java Object) hay nói đơn giản hơn là các đối tượng với các thuộc tính và phương thức cơ bản và hàm khởi tạo – POJO[4] là một đối tượng persistence object được java nói đến. Khi tạo một đối tượng POJO[4] trong thư mục model, ta sẽ tạo class, tên thuộc tính trong class, hàm khởi tạo và các phương thức get, set.

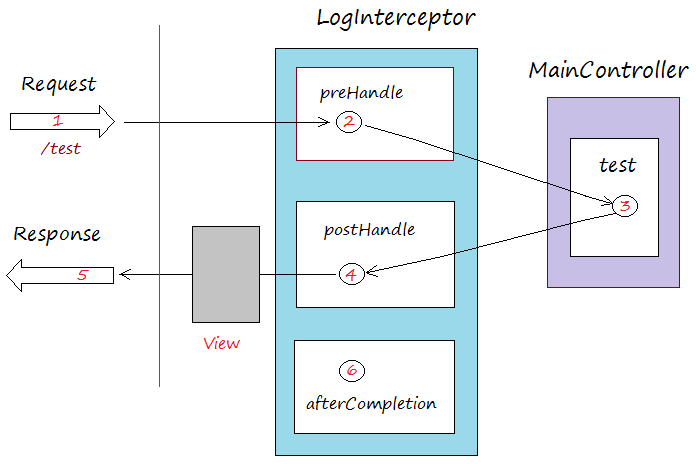
**View:** tượng trưng cho các trang giao diện người dùng, thứ mà người dùng sau khi gửi yêu cầu sẽ hiện ra màn hình kết quả. Tuỳ vào yêu cầu người dùng mà có các View khác nhau. View ở đây trong thực tế thường là các trang với định dạng là html đã được nhúng css, javascript, jquery … để tạo nên một giao diện đẹp mắt hiển thị cho người dùng.

**Controller**: đối với mỗi model thì ta sẽ tạo riêng một controller để quản lý nó và trả về ModelView thích hợp cho DispatcherServlet. Ta có thể nói Control là đóng vai trò và người điều khiển luồng đi phụ (sau Dispatcher) để ánh xạ yêu cầu người dùng tới đúng model cần lấy dữ liệu và trả về kết quả hiển thị cùng dữ liệu (ModelView) tới người quản lý chính (DispatcherServlet). Mỗi một model sẽ có một controller phụ trách.

**ViewResolver:** Sau khi controller trả về ModelView cho DispatcherServlet thì nó sẽ được chuyển ngày tới bộ phận gọi là ViewResolver. Sau khi nhận được tên của View trong ModelView thì ViewResolver tiến hành ánh xạ tới thư mục có chứa view và trả về cho DispatchetServlet. Thường trong thực tế, ta sẽ cấu hình ViewResolver này trong file xml để nó trỏ tới đúng đường dẫn thư mục và đọc đúng tập tin view mà ta muốn.

**Filter (Trong Servlet người ta dùng Filter, còn trong Spring nó được đổi tên thành Interceptor):** trong Spring MVC[3] thì khái niệm Filter được đổi thành Interceptor. Nó sẽ phụ trách xử lý trước và sau khi yêu cầu được gửi tới Controller. Thông thường thì Interceptor dùng để ghi log như ghi lại ngày tháng năm của request, tên request… Ngoài ra Interceptor này có tác dụng chuyển yêu cầu người dùng tới một trang khác, yêu cầu quyền người dùng thích hợp và có thể chặn yêu cầu nếu người dùng không có quyền. Nó được dùng để phân quyền người dùng.

**Ví dụ:** khi ta muốn vào trang quản lý thì ta được trỏ tới một trang đăng nhập tài khoản. Ta phải điền đúng tài khoản có quyền admin thì mới vào được trang quản lý mà ta muốn.



Hình 6. Cách thức hoạt động Filter

* + - 1. **Spring Boot**
* Là một dự án phát triển bởi ngôn ngữ Java trong hệ sinh thái Spring Framework.
* Nó giúp cho các lập trình viên chúng ta đơn giản hóa quá trình lập trình một ứng dụng với Spring, chỉ tập trung vào việc phát triển business cho ứng dụng.
* Để phát triển một ứng dụng web cơ bản HelloWorld sử dụng Spring framework chúng ta sẽ cần ít nhất 5 công đoạn sau:
* Tạo một project sử dụng Gradle hoặc Maven với các dependency cần thiết của Spring MVC[3] và Servlet API[5].
* Một tập tin web.xml để khai báo DispatcherServlet của Spring MVC[3].
* Một tập tin cấu hình của Spring MVC[3].
* Một class Controller trả về một trang “Hello World” khi có request đến.
* Cuối cùng là phải có một web server dùng để triển khai ứng dụng lên chạy.
* Trong các công đoạn này, chỉ có công đoạn tạo một class Controller thì có thể khác cho các ứng dụng khác nhau vì mỗi ứng dụng có một yêu cầu khác nhau.
* Giờ đây với Spring Boot, chúng ta có thể tạo dự án Spring một cách nhanh chóng và cấu hình cũng đơn giản.
* Dưới đây là một số tính năng nổi bật của Spring Boot:
* Tạo các ứng dụng Spring độc lập
* Nhúng trực tiếp Tomcat, Jetty hoặc Undertow (không cần phải deploy ra file WAR)
* Các starter dependency giúp việc cấu hình Maven đơn giản hơn
* Tự động cấu hình Spring khi cần thiết
* Không sinh code cấu hình và không yêu cầu phải cấu hình bằng XML
* Ứng dụng Spring Boot để xây dựng nhanh REST API[5]:
* REST (Representation State Transfer) là một kiến trúc định ra cấu trúc cho việc chuyển đổi dữ liệu. Bằng một cách tổ chức nào đó mà các dữ liệu được truy cập thông qua các hệ thống phân phối nền tảng khác nhau như PC, Mobile…
* REST này sẽ định nghĩa các quy tắc kiến trúc để chúng ta thiết kế ra các API[5] hướng tới tài nguyên hệ thống, bao gồm việc trạng thái tài nguyên được chuyển đổi như thế nào và được chuyển qua các phương thức HTTP nào dưới các ngôn ngữ khác nhau như thế nào.
* REST API[5] là nói tới việc tạo các API[5] (application programming interface) – giúp người dùng có thể thao tác với các tài nguyên hệ thống thông qua các phương thức HTTP dưới các nền tảng viết bằng ngôn ngữ khác nhau.
* Thường các tài nguyên của REST API[5] được định danh dựa vào Uri trong request của client, thường chuẩn để biểu diễn tài nguyên là dạng JSON.
  + - 1. **Restful Web Services**

- Web: khi chúng ta gõ vào trình duyệt web một URL, hay còn gọi là đường dẫn để truy cập một thông tin hoặc dữ liệu nào đó thì chúng ta thông thường sẽ nhận được một thứ gọi là trang web. Nội dung của trang web đã được thiết kế lại dễ nhìn và đẹp mắt.

- Web Service không phải là web, nó là một dịch vụ hoặc một khái niệm rộng hơn, nó cung cấp thông tin thô và khó hiểu hơn với người dùng bình thường, chính vì vậy mà nó được sử dụng bởi các ứng dụng để chế biến các thông tin thô này và trả về cho người dùng cuối cùng.

VD: Khi bạn vào trang “vnexpress.net” để xem các thông tin liên quan đến các lĩnh vực đời sống và các tài liệu thì trang web đó sẽ hiển thị lên với giao diện người dùng đẹp mắt và dữ liệu cho phép người dùng đầu cuối thu thập.

Để có được các dữ liệu thông qua các số liệu thì “vnexpress.net” sẽ cung cấp các dịch vụ web gọi là webservice giúp lấy các số liệu, các dữ liệu sau khi lấy từ các dịch vụ sẽ được chế biến lại và trả về người dùng một trang web hoàn chỉnh

- Restful web service là thuật ngữ ám chữ xây dựng các web service tuần thủ theo các quy tắc cấu trúc của rest. Nếu phạm một trong những quy tắc đó thì người ta chỉ gọi là rest mà thôi. Các quy tắc ấy được nêu như sau:

Client – Server: máy server và máy chủ sẽ tách biệt và không biết gì sâu của nhau. Client chỉ thực hiện request và server nhận request và trả về response. Và client sẽ không biết server chứa cái gì và ngược lại.Vì client – server là mô hình tách biệt nên ta sẽ dễ dàng mở rộng khi cần mà chúng không ảnh hưởng gì tới nhau.

Stateless: mỗi request từ client thì tách biệt với các request khác. Dù người dùng có gửi 2 request giống nhau thì server chỉ trả lời theo 2 request mà không quan tâm hai request đó giống nhau hay không.

Cacheable: để tăng hiệu suất thì giữa client và server sẽ có một bộ phận là cache có nhiệm vụ lưu trữ request và response. Khi request giống nhau thì cache này sẽ đẩy response tới client mà không cần tới server.

Uniform interface: nói tới việc thống nhất interface. Để đảm bảo rang buộc này thì hệ thống sẽ tập trung xử lý tài nguyên. Mỗi tài nguyên sẽ được định danh bằng một URI riêng biệt.

Layered System: hệ thống được chia thành nhiều lớp, việc giao tiếp của một lớp được tiến hành với lớp trên hoặc lớp ở dưới. Điều này giúp phân rõ nhiệm vụ logic và cân bằng nhiệm vụ cho mỗi lớp.

- Các thực hành tốt nhất trong việc thiết kế Restful Web Service bắt nguồn từ kinh nghiệm trong việc làm các dự án thực tế và từ đó họ đauw ra các quy ước cũng như các chuẩn để ta thiết kế một Restful Web Service hoàn chỉnh:

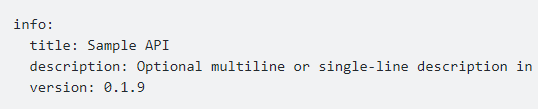
* + - Áp dụng phương thức HTTP[6] đúng với yêu cầu. Ta sẽ chủ yếu dụng 4 phương thức thông dụng nhất: GET (dùng để lấy dữ liệu), POST (dùng để tạo dữ liệu), PUT & PATCH (dùng để cập nhật dữ liệu), DELETE (dùng để xoá dữ liệu). Rest sẽ hướng tới tài nguyên đối tượng và nó được thể hiện trên URI: /object/. Endpoint là thuật ngữ chỉ sự kết hợp giữa phương thức HTTP[6] và URI. (GET /object/ )
    - Không trả về văn bản thô mà phải theo một định dạng nhất định: phần lớn JSON là định dạng phổ biến hay dùng trong REST API. Để trả về một chuối JSON thì chúng ta cần định nghĩa kiểu dữ liệu cho header là Content-Type và giá trị là application/json.
    - Tránh sử dụng động từ trong URI: vì trước kia người ta chỉ dùng hai phương thức HTTP[6] là GET và POST nên mới phải dùng động từ ở URI để phân biệt các hành động theo yêu cầu. Bây giờ khi ta sử dụng các phương thức PUT và DELETE thì theo chuẩn RestFul, phải phải đặt URI là danh từ.
    - Sử dụng danh từ số nhiều trong URI để tường mình hơn.
    - Trả về lỗi cụ thể cho request người dùng nếu request vượt ngoài tầm dữ liệu.
    - Chú ý tới mã tình trạng request. VD: request thành công mã sẽ là 200
    - Việc sử dụng mã tình trạng phải nhất quán (phải có theo quy ước hoặc tuân thủ luật) như GET, PUT, PATCH dùng mã 200 (thành công). POST dùng mã 201 (Đã tạo) và DELETE dùng mã 204 (Không có nội dung).
    - Không dùng URI lồng tài nguyên: nếu ta muốn yêu cầu tài nguyên đang nằm trong tài nguyên khác thì ta sẽ sử dụng chuỗi truy vấn trong URI. Nó sẽ giúp URI đọc vào rõ ràng và dễ hiểu.
      * VD: GET: /tests/?result\_id=2: ta lấy kết quả số 2 trong các bài kiểm tra.
    - Sử lý dấu “/” trong URI để điều hướng trang mong muốn và tránh tình trạng lỗi.
      * VD: /tests/ và /tests đều phải trả về tài nguyên của đối tượng test.
    1. **Swagger**
       1. **Giới thiệu**

Swagger là một công cụ mã nguồn mở để xây dựng OpenAPI Specifications (Định dạng mô tả API) giúp chúng em có thể thiết kế, xây dựng tài liệu và sử dụng REST API.

Swagger có thể được viết bằng dạng cấu hình trong application.yml hoặc dùng @Annotation mà nó cung cấp để tạo tài liệu cho REST API.

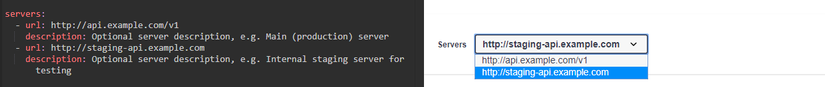
Thông thường thì mọi người hay sử dụng dạng viết file cấu hình nên em sẽ trình bày sơ cách cấu hình bên dưới

* + - 1. **Cấu trúc**
         1. **Metadata**
  + Mỗi OpenAPI specifications sẽ bắt đầu với từ khoá openapi để khai báo phiên bản (VD: openapi: 3.0.0). Phiên bản này sẽ định nghĩa toàn bộ cấu trúc của API. Phần Info sẽ chứa những thông tin của API như: title, description, version
* Title: là tên API
* Description: là thông tin mở rộng, mô tả về API
* Version: chỉ phiên bản API hiện tại



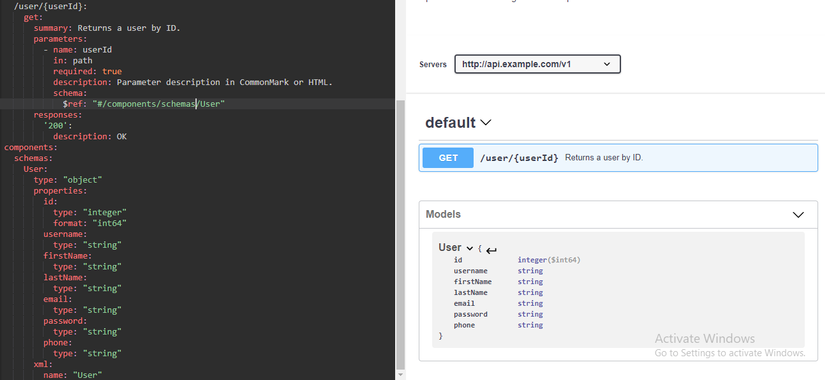
Hình 7. Ví dụ metadata - Swagger

* + - * 1. **Servers**
  + Đây là phần chỉ định đường dẫn của server để chúng em có thể test API. Chúng ta có thể định nghĩa một hoặc nhiều server.
  + Tất cả đường dẫn API là đường dẫn tương đối của URL mà chúng ta định nghĩa



Hình 8. Ví dụ Servers - Swagger

* + - * 1. **Paths**
* Đây là phần trọng tâm chính của API[5]. Ở phần này chúng em sẽ định nghĩa các đường dẫn url cho api, các tham số, ví dụ, kiểu dữ liệu…



Hình 9. Ví dụ Paths - Swagger

* Phần này sẽ bắt đầu bằng từ khoá paths
* Tiếp đó là các phương thức API[5] (GET, POST, DELETE, PUT)
* **Sumary** ở đây là phần tóm tắt cho một API[5]
* **Parameter** là các tham số sẽ truyền vào API[5]
* **Response** là phần trả về của server. Chúng ta có thể định nghĩa những HTTP[6] code:
  + 200: OK
  + 400: BAD REQUEST
  + 404: NOT FOUND
  + 500: SERVER ERROR
    - * 1. **Schema**
* Ta định nghĩa ra các **Models** cho API
* Phần này được khai báo qua từ khoá **component** & **schemas**.
* Chúng ta có thể gọi tới chema này bằng các đường dẫn tương đối (VD: $ref: “#/component/schemas/CongVan”)



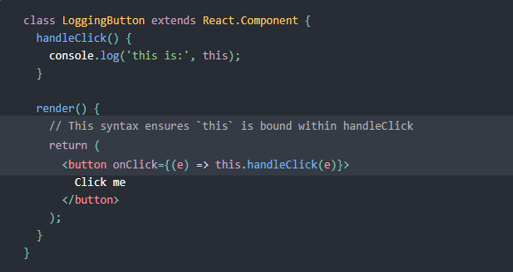
Hình 10. Ví dụ Schema - Swagger

* + 1. **React**
       1. **React JS**

ReactJS là một thư viện JavaScript được dùng để xây dựng giao diện người dùng (user interfaces). ReactJS hỗ trợ người dùng trong việc thiết kế giao diện người dùng (UI) giống như là phần View trong mô hình MVC[3] tương tác với người dùng. ReactJS được duy trì bởi Facebook và một cộng đồng gồm các nhà phát triển và công ty cá nhân.

ReactJS có thể được sử dụng làm nền tảng trong việc phát triển các ứng dụng web SPA-Single Page Application hoặc các ứng dụng di động. Khác với các bộ thư viện khác, ReactJS sẽ tiến hành tạo ra giao diện người dùng (render) từ mã JavaScript kết hợp với bộ cú pháp mở rộng JSX thành các bộ mã mà trình duyệt có thể hiểu và phân giải (encode) được để từ đó hiển thị được cho người dùng sử dụng và tương tác. ReactJS cũng hỗ trợ tối ưu SEO rất tốt đạt chuẩn tìm kiếm của Google với các thuật toán và các plugin có sẵn.

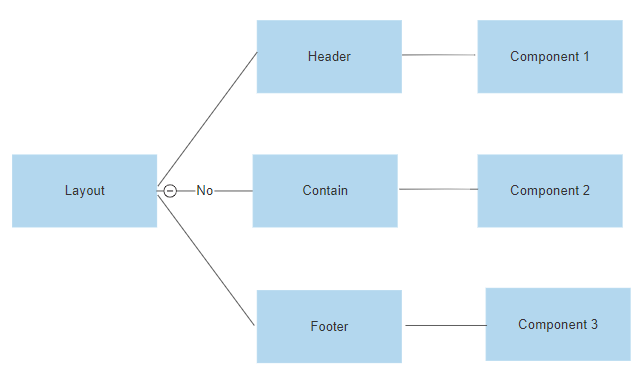
Khi tìm hiểu về ReactJS, một số khái niệm sau cần phải chú ý như sau:

* Component: là một thành phần của website, hay nói cách khác với một trang web cơ bản được chia ra làm 3 phần độc lập là: header, main và footer thì 3 phần này chính là component trong ReactJS. Component thực hiện phương thức render() cùng với dữ liệu đầu vào và trả về (return) mã html thuần để trình duyệt có thể phân giải và hiển thị cho người dùng.Việc phân tách trang web thành các component giúp việc quản lý dễ dàng hơn, viết một lần dùng ở mọi nơi khi chỉ cần nhúng (import) component đó vào nơi cần sử dụng.
* Stateful component: là nơi lưu trạng thái của component. Đây là nơi khai báo và lưu trữ các giá trị riêng của component đó( ví dụ danh sách sinh viên. Khi danh sách này bị thay đổi (nghĩa là state bị thay đổi) như thêm hoặc xoá một sinh viên, phương thức render() sẽ được gọi để in lại (re-render) danh sách sinh viên với thông tin đã được thay đổi. Hay nói một cách khác, component sẽ được render lại khi trạng thái của component đó có sự thay đổi.
* Props (Properties): là các thuộc tính liên kết giữa tương tác, lưu trữ và truyền dữ liệu giữa các component có mối quan hệ với nhau. Ví dụ như bảng danh sách sinh viên (component cha) và từng dòng trong bảng đó (component con). Tương tự statefull component, component sẽ được render lại khi props thay đổi. Có một lưu ý nhỏ ở đây là props không thể thay đổi giá trị trong component con. Component con chỉ nhận props từ component cha. Vì vậy việc thay đổi giá trị của props chỉ xảy ra ở component cha.
* Life cycle: là vòng đời của component, người dùng có thể chỉ định, truy cập và sử dụng tuỳ biến thông qua các hàm: componentDidMount(), componentWillUnmount() … Các life cycle cũng đảm bảo tính logic cho React, chúng quyết định xem component nào sẽ render trước và component nào sẽ sau, những điều sẽ làm trước khi render, những việc sẽ làm sau khi render …
* Handling Events: tạm gọi là việc bắt sự kiện, một công việc vô cùng quan trọng khi tương tác với websit. Cũng gần giống với JavaScript khi bắt các sự kiện như onClick, onLoad, onChange… nhưng điểm khác biệt là vì các component kế thừa một lớp tượng hình là lớp Component nên các phương thức (function) muốn được thực hiện phải được “bind” bằng nhiêu cách. Cách phổ biến nhất là dùng arrow function như ví dụ sau, khi click vào nút “Click me”, phương thức handleClick sẽ được gọi và in ra toàn bộ các thông số của nút.  
  

Hình 11. Mô tả cách bắt sư kiện trong ReactJS

* Nhìn lại một cách tổng quát, ReactJS sẽ phân chia một trang web thành các component nhỏ, mỗi component sẽ có một state để lưu trữ giá trị riêng cho component đó và có một lifecycle để quản lý vòng đời (life cycle) của nó. Các component có thể giao tiếp với nhau thông qua các thuộc tính mang tên props.
  + - 1. **Redux**

Việc mỗi componnet có một state (trạng thái) riêng dẫn tới việc quản lý các state này vô cùng khó khăn, hơn thế khi làm một dự án lớn có tới hàng ngàn component sẽ kéo theo có một số lượng state rất lớn cần quản lý. Việc truy vấn dữ liệu trao đổi dữ liệu giữa các component cũng mất rất nhiều thời gian và chi phí. Xem xét ví dụ sau:

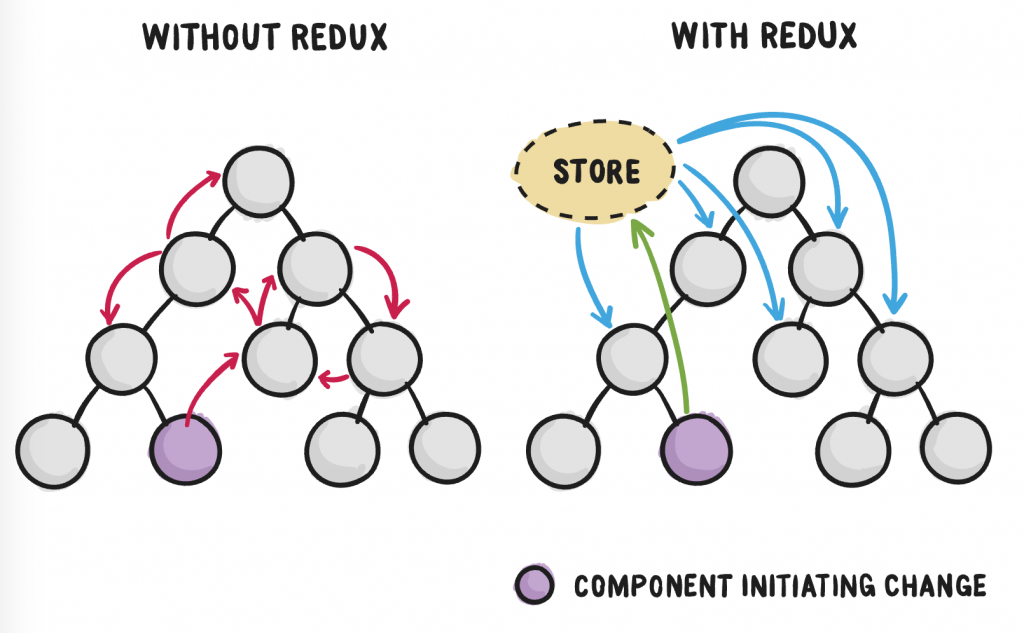


Hình 12. Sơ đồ cơ bản một website ReactJS cơ bản

Xét ví dụ trên, nếu đứng ở component 1 và muốn truy vấn tới dữ liệu tại Component 3, ta phải đi qua 3 component khác hình thành một đường đi như sau: Component 1 🡪 Header 🡪 Layout 🡪 Footer 🡪 Component 3. Cũng đồng nghĩa ngược lại khi bắt đầu tại component 3 và đích đến là componet 1. Việc này sẽ tạo nên một hệ thống mạng nhện liên kết dày đặc không nhất quán và làm mất rất nhiều thời gian, chi phí để quản lý. Lúc này chúng ta cần một kho lưu trữ state chung để khi bất kỳ component nào cần dữ liệu từ state thì chỉ cần vào kho này để lấy và cũng chính vì vậy Redux ra đời để giải quyết vấn đề này.

Như đã đề cập ở trên, Redux là một thư viện JavaScript giúp tạo ra một kho chứa các state của ứng dụng. Được dựa trên nền tảng tư tưởng của ngôn ngữ Elm kiến trúc Flux do Facebook giới thiệu, do vậy Redux thường là bộ đôi kết hợp hoàn hảo với React.

Redux sẽ tạo quản lý toàn bộ các state và props, các component khi cần truy vấn cơ sở dữ liệu không cần thông qua nhau nữa mà gọi thẳng lên Redux để xử lý và lấy dữ liệu khi cần. Nói một cách khác redux giống như một máy chủ, các component là client, khi cần dữ liệu client không cần hỏi lẫn nhau mà gọi thẳng lên Redux để xử lý và lấy dữ liệu cho mình. Việc này không những giúp dễ quản lý các state, giúp giao tiếp các component linh hoạt mà còn tiết kiệm được rất nhiều thời gian chi phí cho việc quản lý, vận hành và bảo trì website. Bên cạnh đó, Redux cũng hỗ trợ quản lý các thư viện thứ bên thứ ba nên source càng thêm “sạch” và nhất quán hơn.

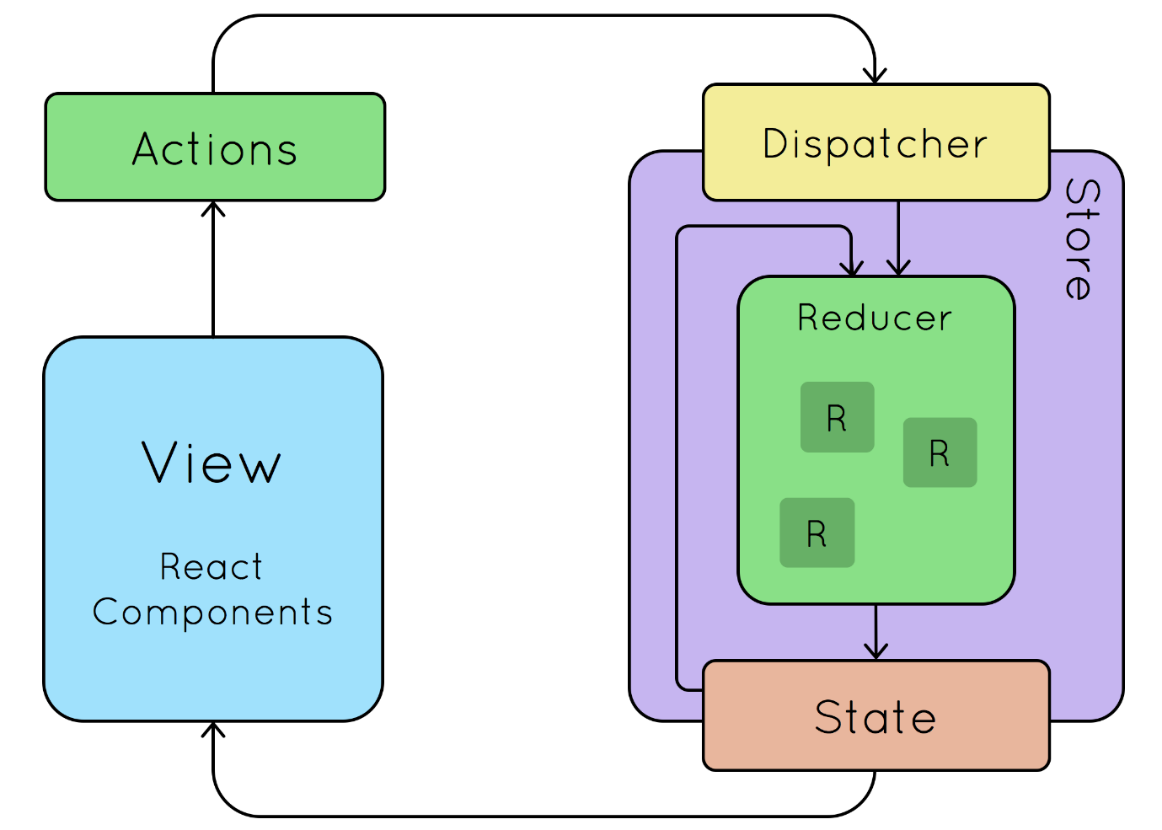


Hình 13. Ứng dụng trước và sau khi sử dụng Redux

Các thành phần của redux gồm có:

* View: phần hiển thị và tương tác với người sử dụng
* Actions: hành động mà người sử dụng tạo nên khi gọi các sự kiện.
* Store: giống như bộ não, nó lưu trữ state và xử lý các thông tin dựa trên các state ấy. Store gồm có 3 phần chính.
  + Dispatcher: Thực thi các actions (là các phương thức được định nghĩa để thực thi một công việc cụ thể nào đó).
  + Reducer: nơi tạo ra các state và quản lý chúng.
  + State: nơi lưu tập trung các state của các component.

Đầu tiên, reducer sẽ khởi tạo state cho các component và lưu chúng vào block State (xem hình dưới). Sau đó, khi người dùng yêu cầu một yêu cầu nào đó từ view, các view sẽ gửi các Actions này tới cho Dispatcher. Lúc này các Dispatcher thực thi và gửi kết quả tới reducer để tiếp tục xử lý, sau khi xử lý xong Reducer sẽ tạo ra một sstate mới, lưu nó vào block State và cuối cùng là gửi state đó cho View.

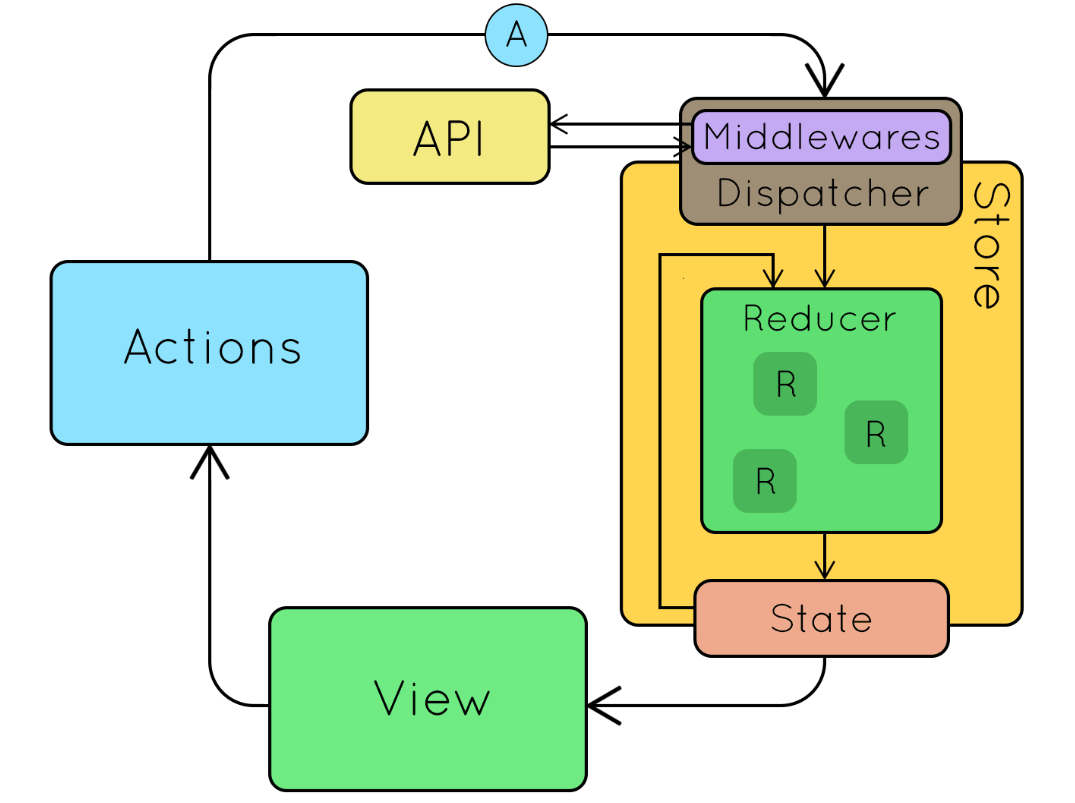


Hình 14. Mô tả cách hoạt động của Redux

* + - 1. **Middleware**

Có thêm một bài toán cần xử lý để làm một ứng dụng hoàn chỉnh: xử lý bất đồng bộ. Xử lý đồng bộ có thể hiểu đơn giản là khả năng thực thi các tác vụ độc lập nhau, có nghĩa là chúng không nhất thiết phải chạy một cách tuần tự (có thể chạy song song với nhau) hoặc xử lý này không phải đợi xử lý khác giúp cho việc cải thiện hiệu suất hoạt động của ứng dụng Theo thông thường ứng dụng sẽ chạy đồng bộ, điều này hoàn toàn bình thường cho tới khi có một vài trường hợp đặc biệt đòi hỏi xử lý bất đồng bộ. Trên thực tế sử dụng lập trình bất đồng bộ (Asynchronous programing) đã trở nên khá phổ biến. Lấy ví dụ cụ thể, action yêu cầu danh lấy danh sách sinh viên từ server, lưu vào state và xuất ra ngoài màn hình để người dùng có thể thấy và tương tác. Nhưng vấn đề đã sảy ra, khi gọi API, danh sách sinh viên không thể trả về ngay lập tức như khi lấy dữ liệu từ các biến trong hàm được, nó mất thời gian gửi dữ liệu lên cho server, đợi server xử lý, nhận dữ liệu về rồi xử lý dữ liệu ấy thành danh sách sinh viên rồi thực hiện các công việc tiếp theo. Công việc ấy mất một khoảng thời gian nhưng reducer không hiểu việc đó. Nó chỉ thực hiện tuần tự: gọi API, xử lý và xuất ra một danh sách trống ra màn hình. Nó không đợi dữ liệu trả về từ server rồi mới xuất ra màn hình như mong đợi. Như chính vì vậy, lúc này cần có sự can thiệp của middleware.

Nói một cách ngắn gọn, middleware sẽ đứng giữa Actions và Store.

Khi gọi API để lấy danh sách sinh viên, yêu cầu sẽ được gửi đi và middleware sẽ ngừng luồng công việc tại đây cho tới khi nào nhận được response từ server. Hiện nay có rất nhiều thư viện middleware với cùng hàng ngàn chức năng nhưng phổ biến nhất vẫn là middleware “Thunk” với tính các tính năng cơ bản nhưng đủ để sử dụng cho một dự án vừa và nhỏ.

Hình 15. Cách sơ đồ hoạt động của redux sau khi tích hợp middleware

* + 1. **Ant Design**

Gần giống như Bootstrap nhưng cao cấp hơn, là một thư viện React UI cung cấp rất nhiều component đã được thiết kế giao diện sẵn cùng các api để tuỳ chỉnh component đó như: Modal, Menu, Confirm…

Ant Design được tạo bởi tập đoàn Alibaba của Trung Quốc và được sử dụng bởi một số tập đoàn tên tuổi lớn như : Alibaba, Tencent, Baidu…Ant Design không những hỗ trợ ReactJS mà hiện tại đã có nhiều phiên bản tuỳ chọn như Ant Design Pro với nhiều tính năng đặc quyền, Ant Design Mobile dùng cho các thiết bị di động, NG-ZORRO và NG-ZORRO-MOBILE là phiên bản hỗ trợ cho Angular, Ant Design cho Vue….

Ngoài việc cung cấp các component, Ant Design cũng cung cấp thêm các Icon, themes, template, các bộ font khiến cho việc tạo ra một giao diện hoàn chỉnh cho ứng dụng càng dễ dàng hơn bao giờ hết. Tất cả công việc cần làm là chỉ cần nhúng thư viện Ant Design vào dự án, sau đó lên trang chủ của Ant Design tìm, chọn và tìm hiểu cách sử dụng nó cùng các API[5] tuỳ biến component đó cho phù hợp với yêu cầu của dự án. Ngoài các API[5], Ant Design cũng cho phép tuỳ biến hiển thị bằng cách ghi đè (override) lại phương thức render của component đó. Hầu hết các component đều có các API[5] để handle các hành động của component đó như: onSubmit, onChange, onClick Khiến cho việc thiết kế giao diện đã dễ lại càng dễ hơn.



Hình 16. Trang đăng nhập được tạo bằng Ant Design

* + 1. **Axious**

Làm việc với API[5] hiện nay không còn xa lạ gì với người lập trình. Tạo một server hỗ trợ API[5] sẽ vừa dễ phát triển vừa dễ bảo trì sau này. Chỉ cần làm một lần có thể mang lên mọi nền tảng có hỗ trợ API và hiện nay hầu hết mọi ngôn ngữ lập trình đều đã làm rất tốt việc này. Mặc định ReactJS không chính thức hỗ trợ gọi API[5] nhưng có rất nhiều thư viện bổ mở rộng bổ sung tính năng này và nổi bật nhất và phổ biến nhất hiện nay là Axios.

Axios là một thư viện HTTP[6] Client dựa trên Promise. Cơ bản thì nó cung cấp một API[5] cho việc xử lý XHR (XMLHttpRequests). Axios có các đặc điểm như là: tạo XMLHttpRequests từ trình duyệt, thực hiện các http request, hỗ trợ Promise API[5], chặn request và response, Chuyển đổi dữ liệu request và response, hủy request, tự động chuyển đổi về dữ liệu JSON và hỗ trợ phía client để chống lại tấn công CSRF. Axios cũng hỗ trợ rất nhiều trình duyệt web phổ biến hiện nay như: Chrome, FireFox, Safari, Opera Mini, Internet Edge, Internet Explorer. Cách sử dụng của Axios cũng rất đơn giản,

chỉ cần nhúng (import) thư viện Axios vào dự án hành tuỳ chỉnh theo ý mình.

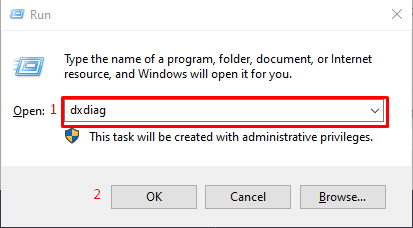


Hình 17. Một đoạn code mô tả cách thức sử dụng của thư viện Anxious

* 1. **Cài đặt các công cụ và phần mềm**
     1. **Yêu cầu tối thiểu**
* Môi trường: máy tính chạy windows 7 trở lên.
* Ổ cứng: trống 2GB.
* RAM: tối thiểu 2GB.
* Trình duyệt: khuyến nghị dùng Chrome.
* Bộ cài đặt Resource.

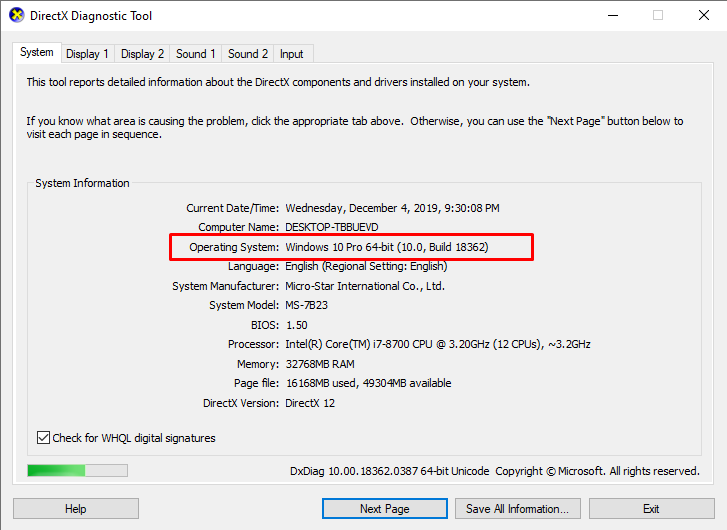
Lưu ý: Trước khi cài đặt phần mềm và công cụ, ta cần kiểm tra cấu hình máy tính, hệ điều hảnh bằng cách:

* Ấn tổ hợp phím + R, một cửa sổ “Run” sẽ hiện ra, nhập vào khung dữ liệu “dxdiag” rồi ấn chọn OK.



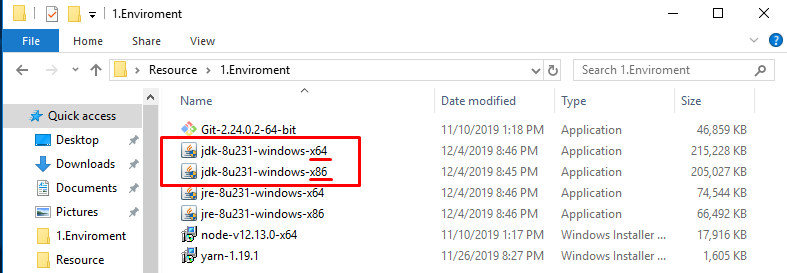
Hình 18. Mở DirectX Diagnostic Tool để xem cấu hình mấy tính

Một cửa số Diagnostic Tool hiện ta, trên cửa sổ này cũng sẽ hiện thông số của hệ điều hành như hình dưới. Lưu ý phải chú ý hệ điều hành đang sử dụng là 32bit hay 64bit để chuẩn bị cho bước kế tiếp:

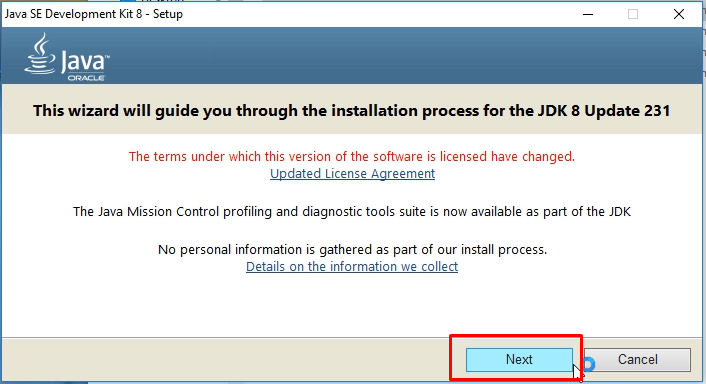


Hình 19. Thông số hệ điều hành

* + 1. **Thư viện và môi trường cho java**
* Vào thư mục SETUP sẽ có 2 file jdk như hình dưới. Chọn file cài đặt tương ứng với hệ điều hành đã xác định ở bước trên và tiến hành cài đặt

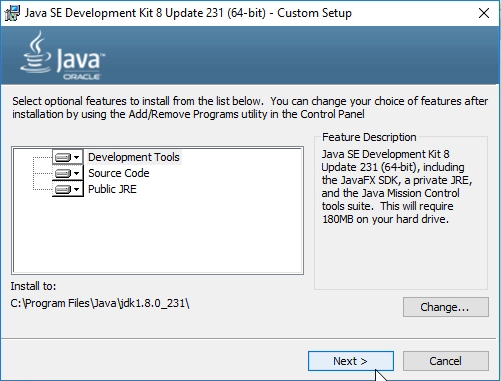


Hình 20. Chọn file cài đặt JDK[7] tương ứng với hệ điều hành

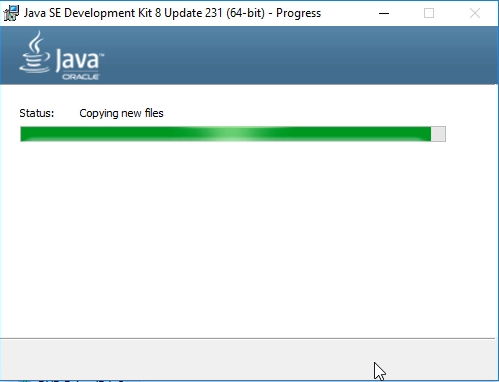


Hình 21. Cửa sổ cài đặt

* Kiểm tra các packed xem có bị thiếu gói nào không, lưu ý ở bước này không thay đổi thư mục cài đặt để tránh các lỗi không đáng có

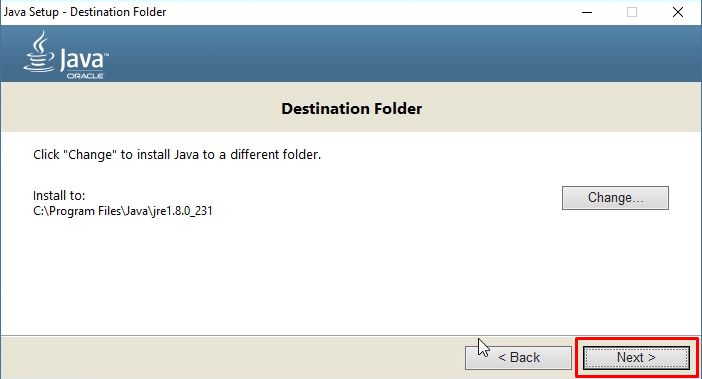


Hình 22. Kiểm tra tính đầy đủ của packed



Hình 23. Quá trình cài đặt

* Cửa sổ thay đổi thư mục cài đặt hiện ra, tại bước này cũng không nên thay đổi đường dẫn thư mục cài đặt, hãy để mặc định

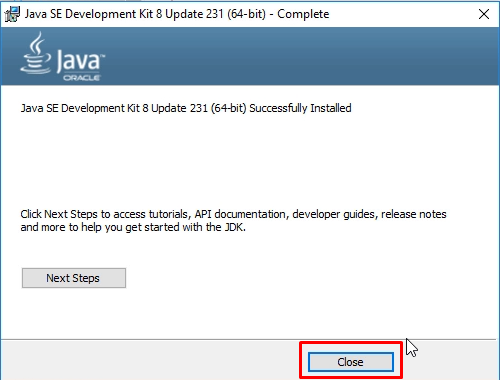


Hình 24. Xác nhận đường dẫn cài đặt

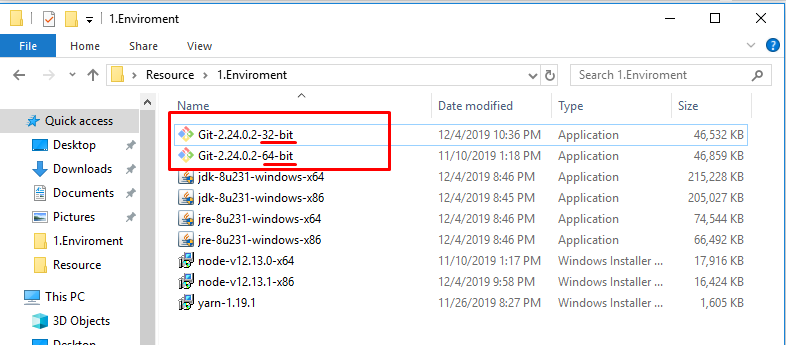


Hình 25. Quá trình cài đặt

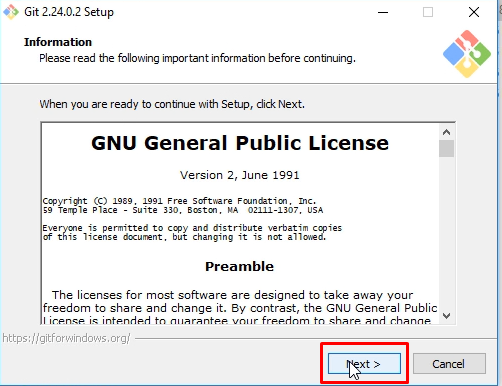
* Quá trình cài đặt đã xong, chọn “Close” để đóng cửa sổ này, kết thúc quá trình cài đặt.



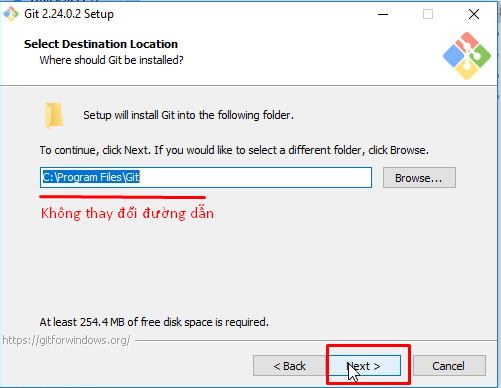
Hình 26. Hoàn cài đặt JDK[7]

* Vào thư mục SETUP, sẽ có 2 file jre như hình dưới. Chọn file cài đặt tương ứng với hệ điều hành đã xác định ở bước trên và tiến hành cài đặt tương tự như JDK[7].
  + 1. **Git**
* Vào thư mục SETUP, sẽ có 2 file jre như hình dưới. Chọn file cài đặt tương ứng với hệ điều hành đã xác định ở bước trên và tiến hành cài đặt

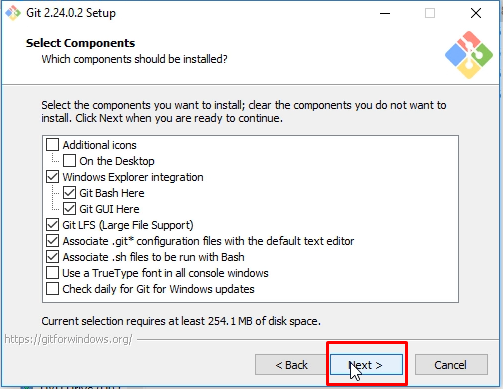
Hình 27. Chọn file cài đặt tương ứng với hệ điều hành



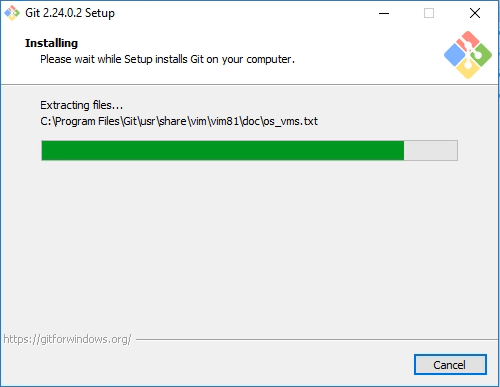
Hình 28. Xác nhận bản quyền



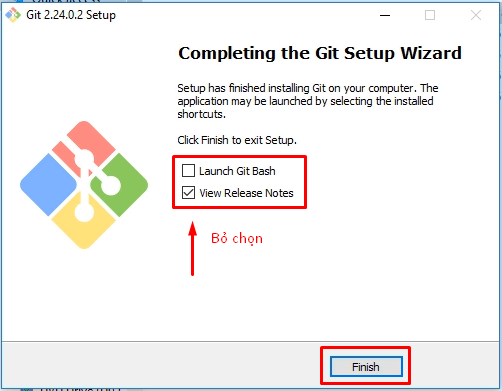
Hình 29. Đường dẫn lưu trữ phần mềm cài đặt



Hình 30. Các tính năng của Git

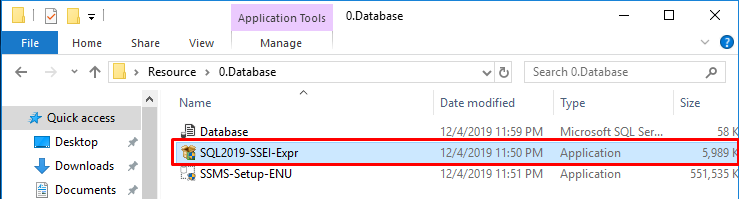


Hình 31. Quá trình cài đặt Git

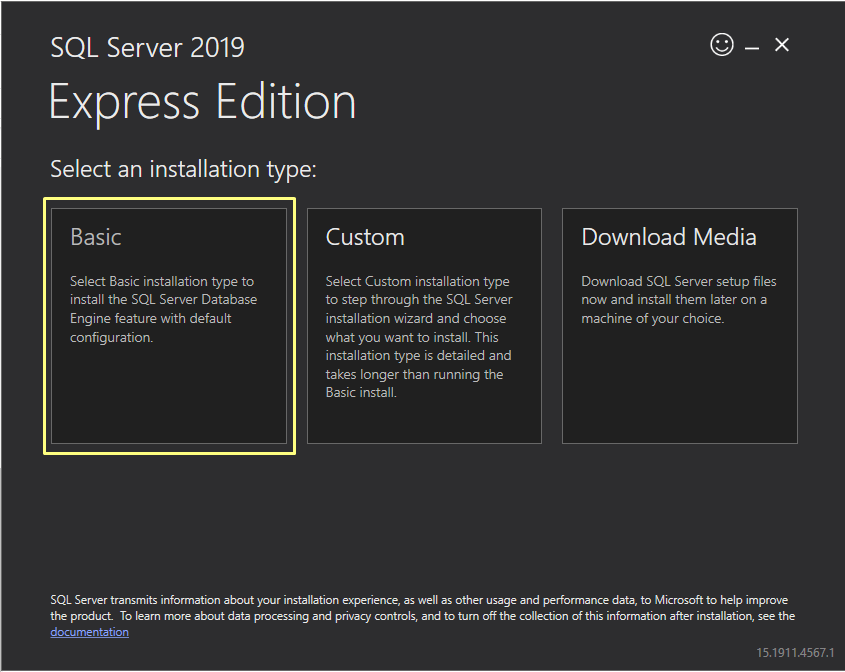


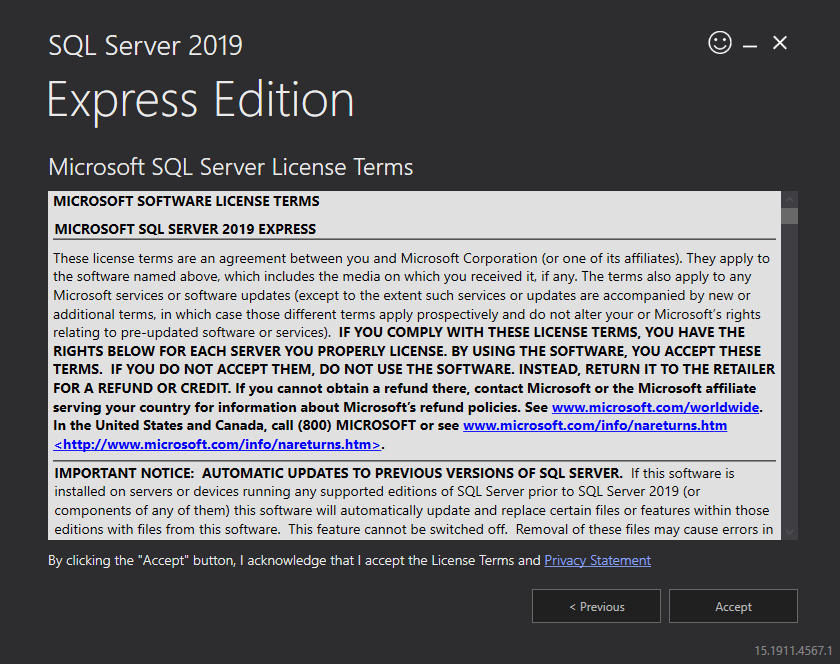
Hình 32. Hoàn tất cài đặt Git

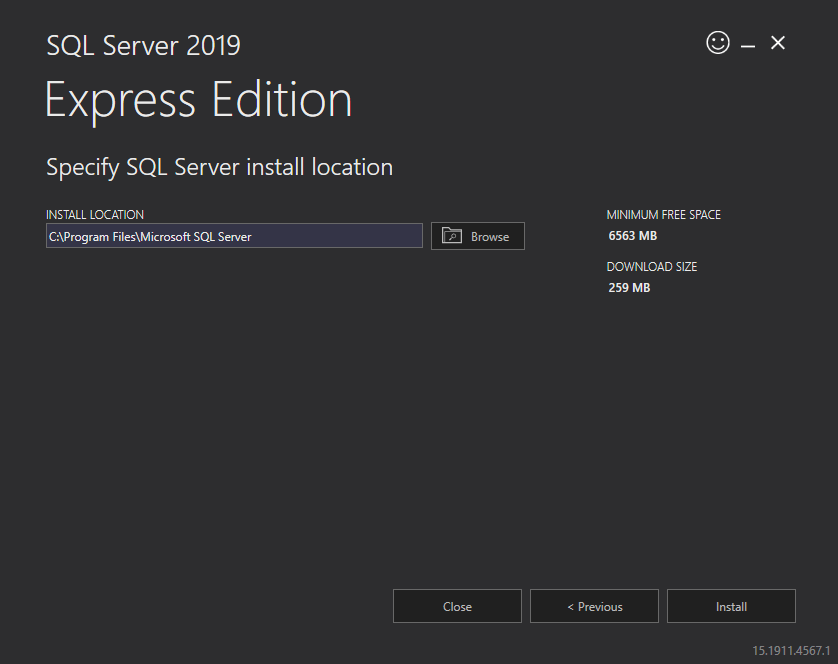
* + 1. **Microsoft SQL Server**
* Vào thư mục SETUP chạy file SQL[8] 2019-SSEI-Expr.exe để tiến hành cài đặt Microsoft SQL[8] Server.



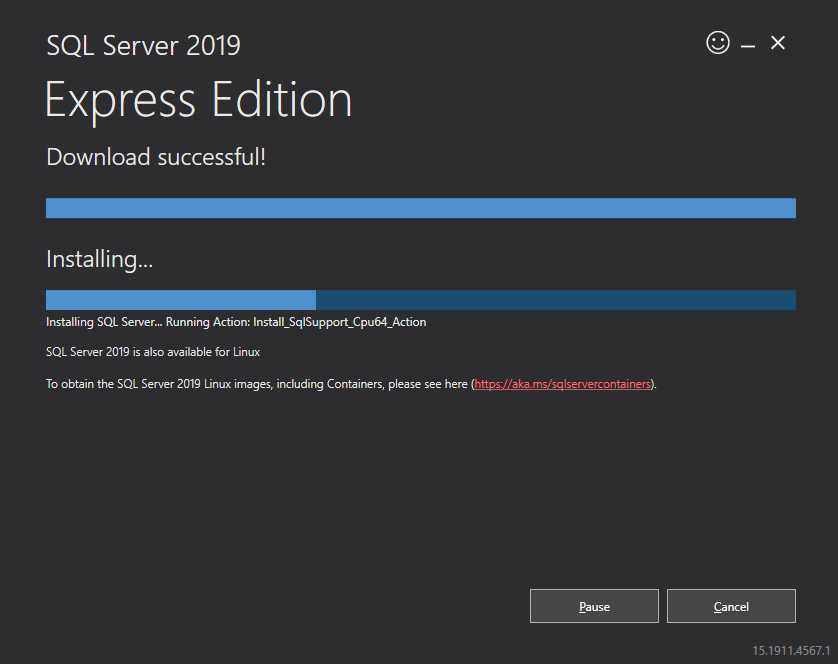
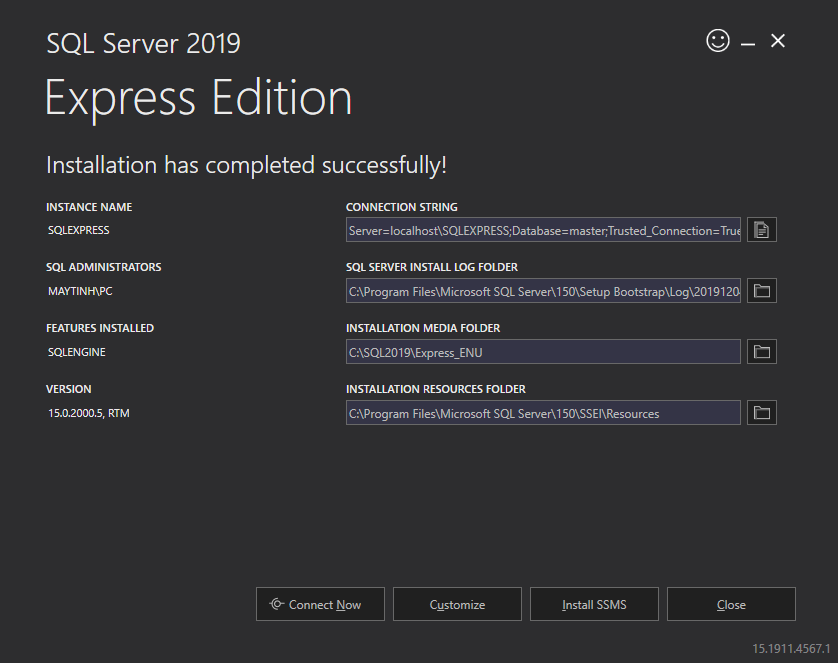
Hình 33. Cài đặt Microsoft SQL[8] Server



Hình 34. Chọn Basic để tiến hành cài đặt

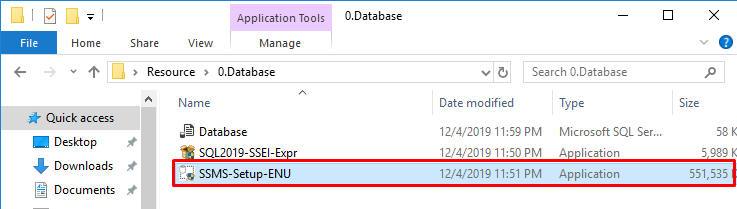
Hình 35. Chấp nhận điều khoản

Hình 36. Chọn “Install” để cài đặt

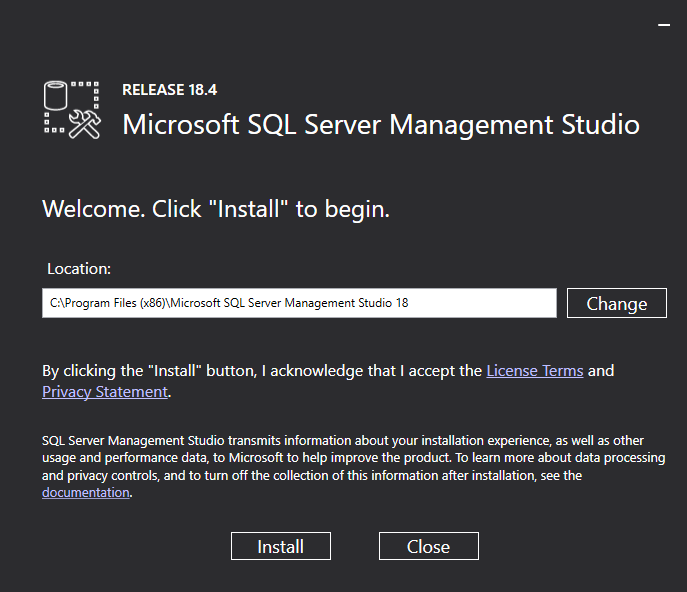
Hình 37. Quá trình down load cài đặt diễn ra

Hình 38. Hoàn tất quá trình cài đặt SQL Server

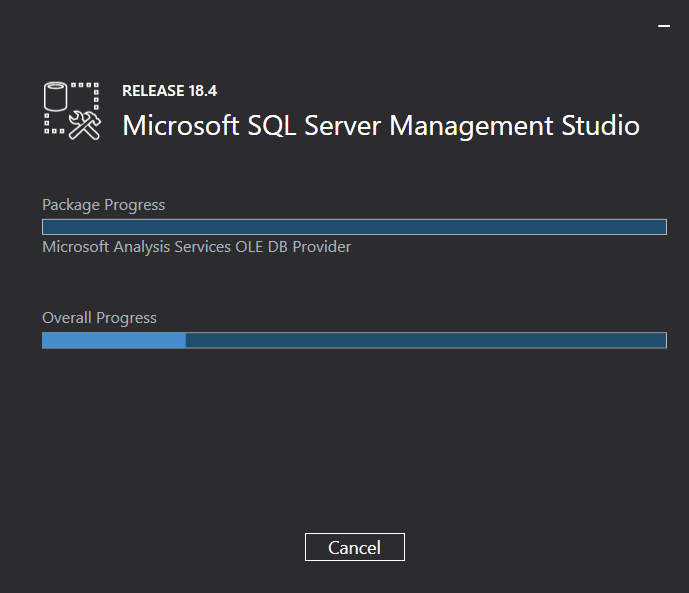
* Vào thư mục SETUP chạy file SSMS-Setup-ENU.exe để tiến hành cài đặt Microsoft SQL[8] Server Management Studio 18.



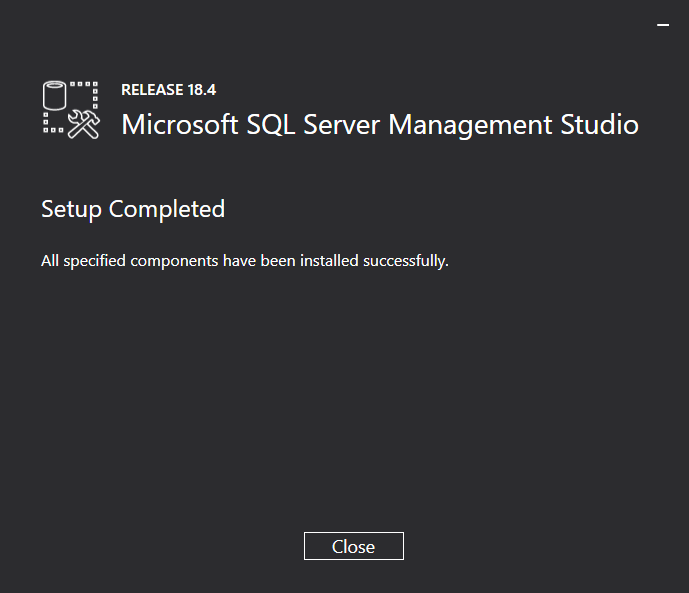
Hình 39. Cài đặt bộ quản lý cho SQL Server



Hình 40. Chọn cài đặt

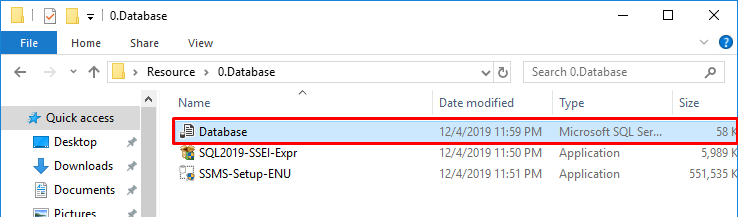


Hình 41. Quá trình cài đặt diễn ra

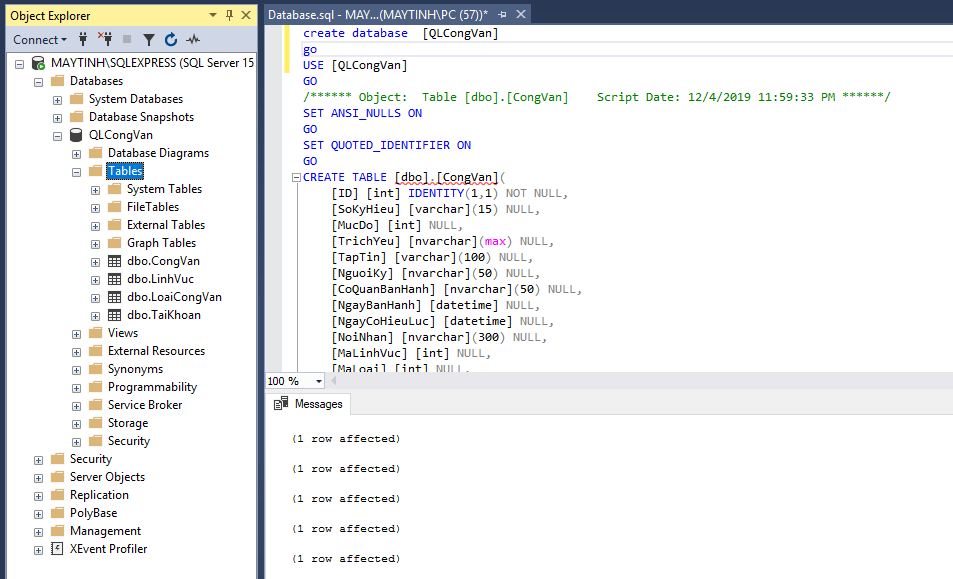


Hình 42. Quá trình cài đặt Bộ quản lý SQL Server hoàn tất

* Vào thư mục DATABASE chạy file Database.sql để tiến hành khôi phục cơ sở dữ liệu

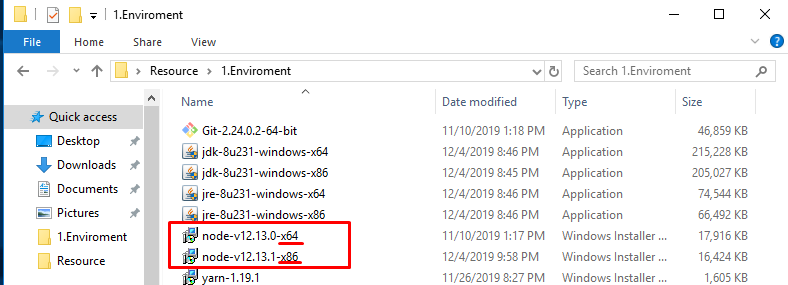


Hình 43. Chọn Database.sql để tiến hành khôi phục cơ sở dữ liệu

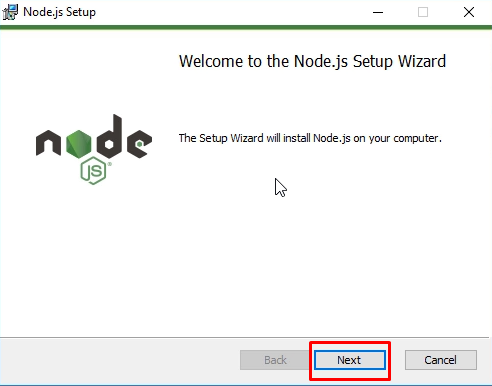


Hình 44. Quá trình tạo Schema và dữ liệu

* + 1. **Node.js**
* Vào thư mục SETUP, sẽ có 2 file jre như hình dưới. Chọn file cài đặt tương ứng với hệ điều hành đã xác định ở bước trên và tiến hành cài đặt.

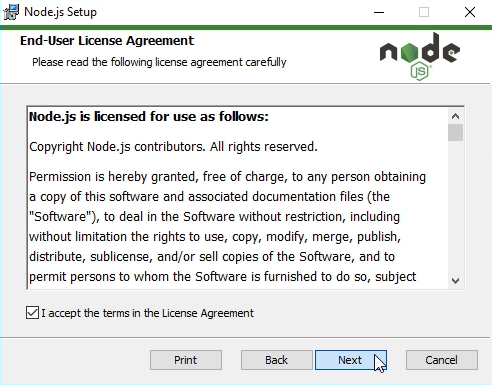


Hình 45. Chọn file cài đặt tương ứng với hệ điều hành



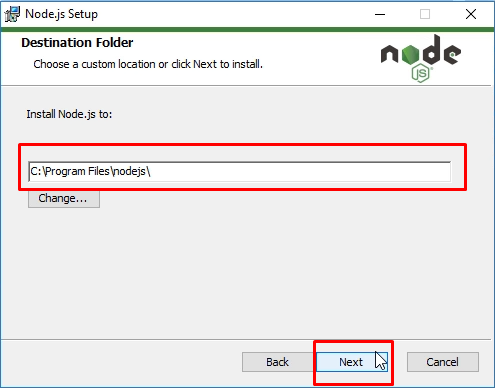
Hình 46. Cửa sổ cài đặt

* Đọc và chọn đồng ý với các điều khoản của nhà phát triển, sau đó chọn “Next” để chuyển sang bước kế tiếp.



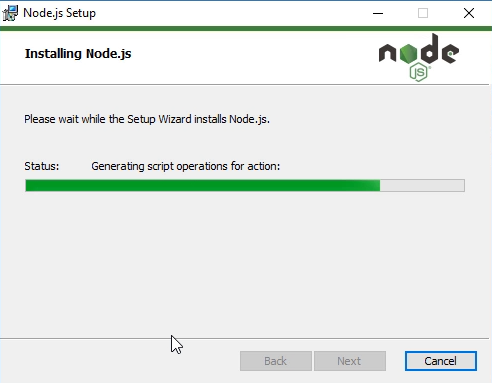
Hình 47. Đọc và đồng ý với các điều khoản của nhà phát triển

* Bước tiếp theo, vẫn không thay đổi đường dẫn cài đặt để tránh lỗi không đáng có, sau đó nhấn “Next” để chuyển sang bước kế tiếp.

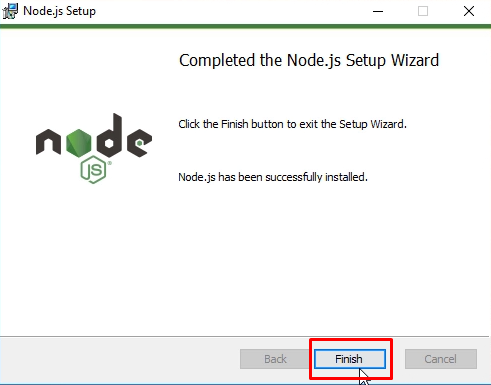


Hình 48. Xác nhận đường dẫn lưu phần mềm

* Sau đó chọn “Next” liên tục cho tới khi quá trình cài đặt được diễn ra và hoàn tất.

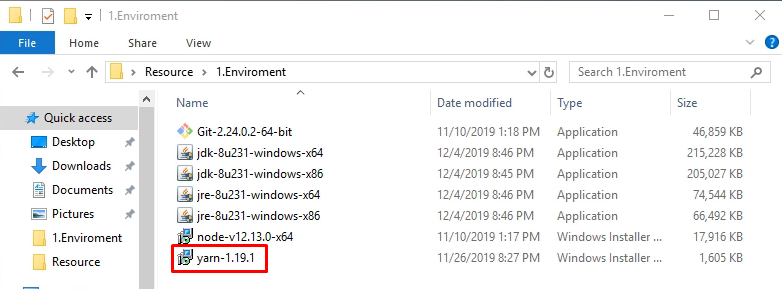


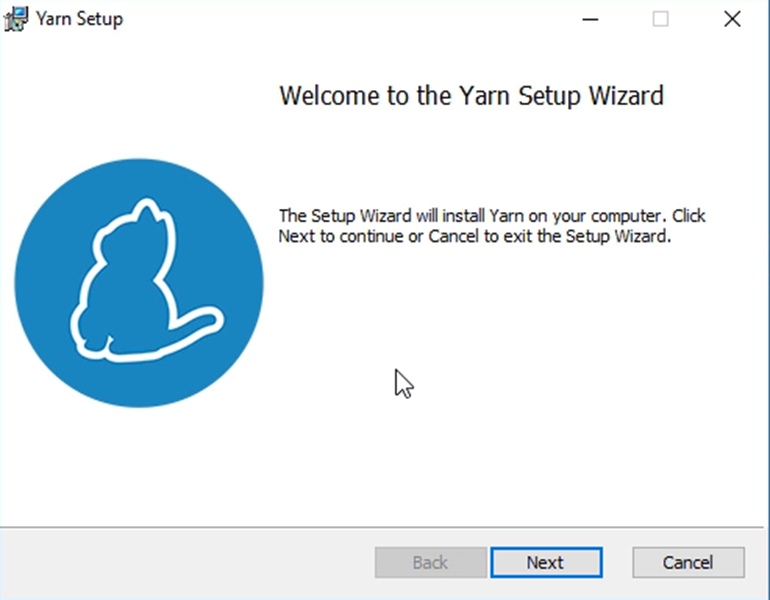
Hình 49. Quá trình cài đặt Node.js



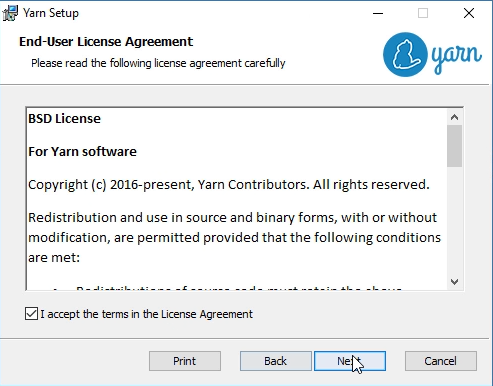
Hình 50. Hoàn tất quá trình cài đặt Node.js

* + 1. **Yarn**
* Vào thư mục SETUP, chọn file cài đặt yarn như hình dưới. Chọn file cài đặt tương ứng với hệ điều hành đã xác định ở bước trên và tiến hành cài đặt.

****

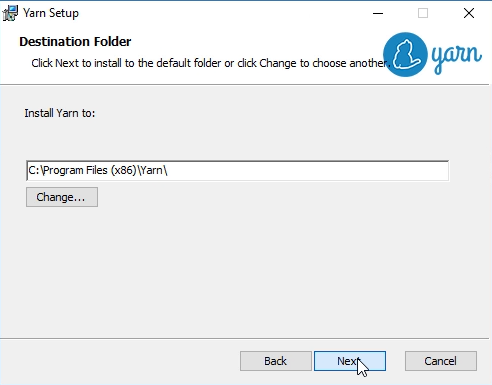
Hình 51. Chọn file cài đặt yarn

Hình 52. Chọn Next để cài đặt

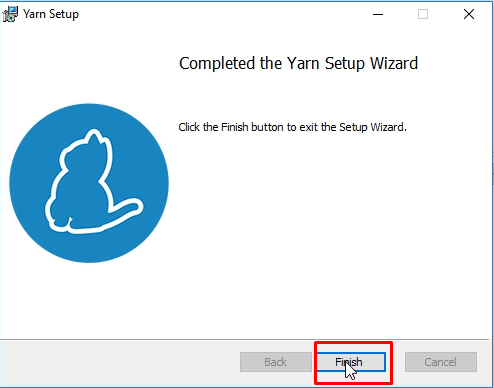


Hình 53. Đọc và chấp nhận điều khoản của nhà phát triển Yarn

* Sang bước kế tiếp chọn đường dẫn cài đặt, tại bước này không nên thay đổi đường dẫn này tránh các lỗi vặt không đáng có.

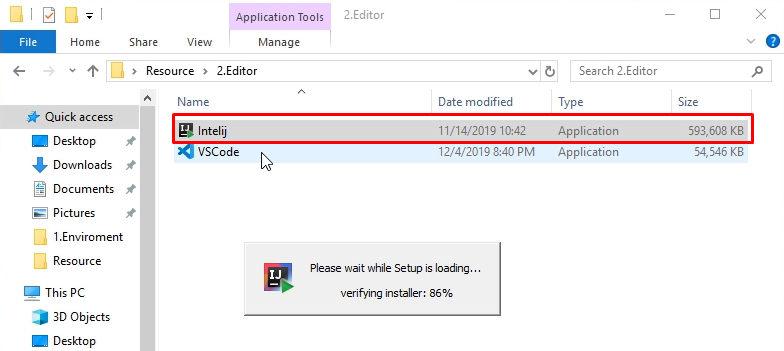


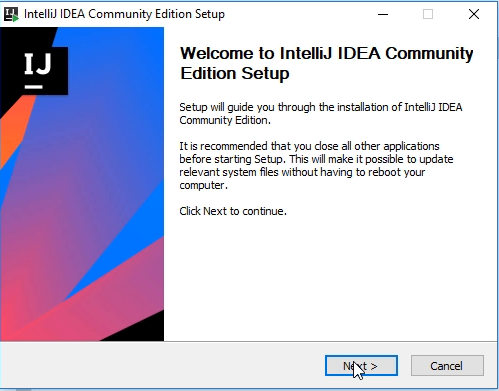
Hình 54. Xác nhận nơi lưu trữ Yarn trong máy



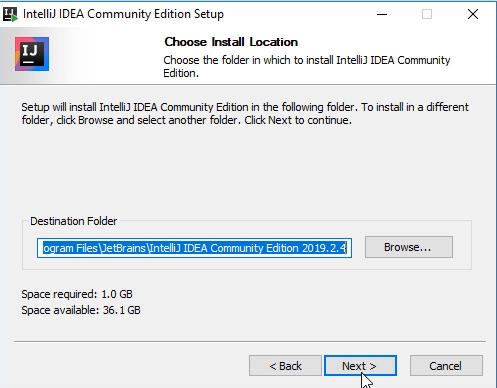
Hình 55. Hoàn tất quá trình cài đặt Yarn

* + 1. **Intellij**
* Vào thư mục SETUP, chọn file cài đặt Intellij như hình dưới. Chọn file cài đặt tương ứng với hệ điều hành đã xác định ở bước trên và tiến hành cài đặt.



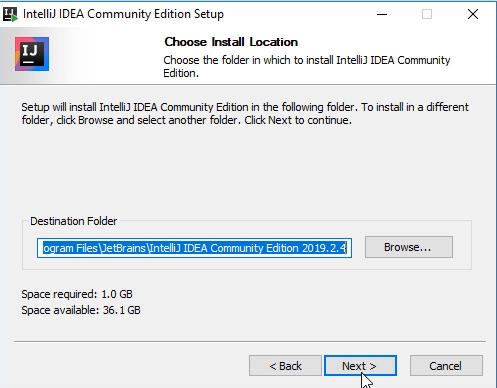
Hình 56. Chạy file cài đặt Intellij

Hình 57. Chọn Next để tiếp tục cài đặt

* Sang bước chọn đường dẫn cài đặt, không thay đổi để tránh lỗi không đáng có, sau đó chọn “Next” để chuyển sang bước kế tiếp.

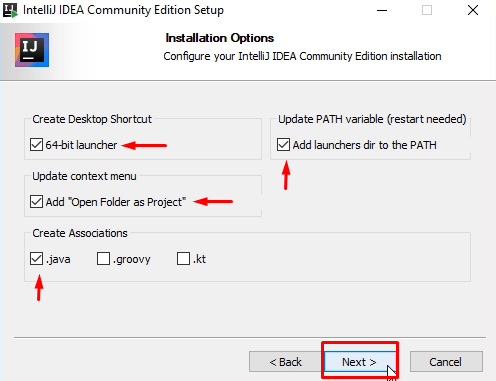
Hình 58. Chọn Next để tiếp tục cài đặt Intellij

* Chọn thông số cài đặt như hình, có thể chọn thêm groovy và kt nếu có dùng thêm để code ngôn ngữ Groovy và Kotlin. Bắt buộc chọn thêm “Add launchers dir to the PATH” để cài đặt biến môi trường. Sau đó chọn “Next” để tiếp tục

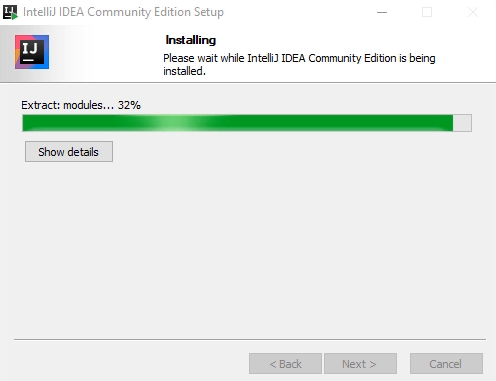


Hình 59. Chọn đường dẫn cài đặt Intellij

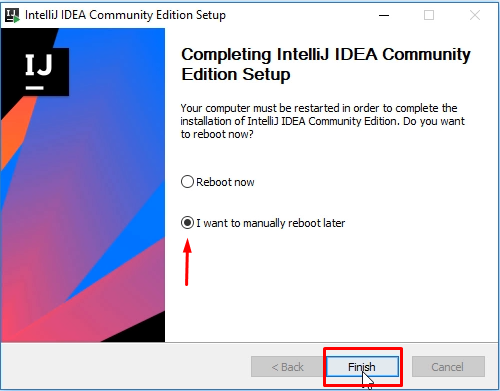
* Sau đó liên tục chọn “Next” cho tới khi quá trình cài đặt diễn ra và hoàn tất.



Hình 60. Tuỳ chọn cài đặt

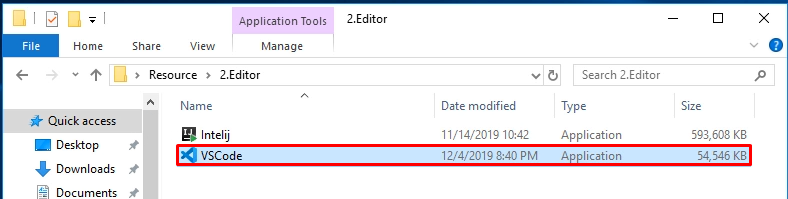


Hình 61. Quá trình cài đặt diễn ra



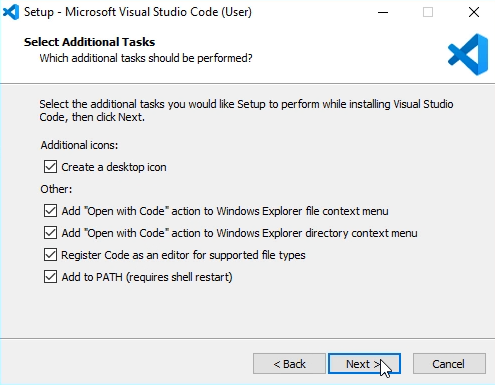
Hình 62. Hoàn tất quá trình cài đặt Intellij

* + 1. **Visual Studio Code**
* Vào thư mục SETUP, chọn file cài đặt Visual Code như hình dưới. Chọn file cài đặt tương ứng với hệ điều hành đã xác định ở bước trên và tiến hành cài đặt.



Hình 63. Chọn file cài đặt VSCode để tiến hành cài đặt

* Sau đó liên tục chọn “Next”, tới đó tới bước này thì chọn như hình để thêm tuỳ chọn mở dự án vào menu chuột phải, cài đặt biến môi trường và đưa icon ra ngoài màn hình desktop.



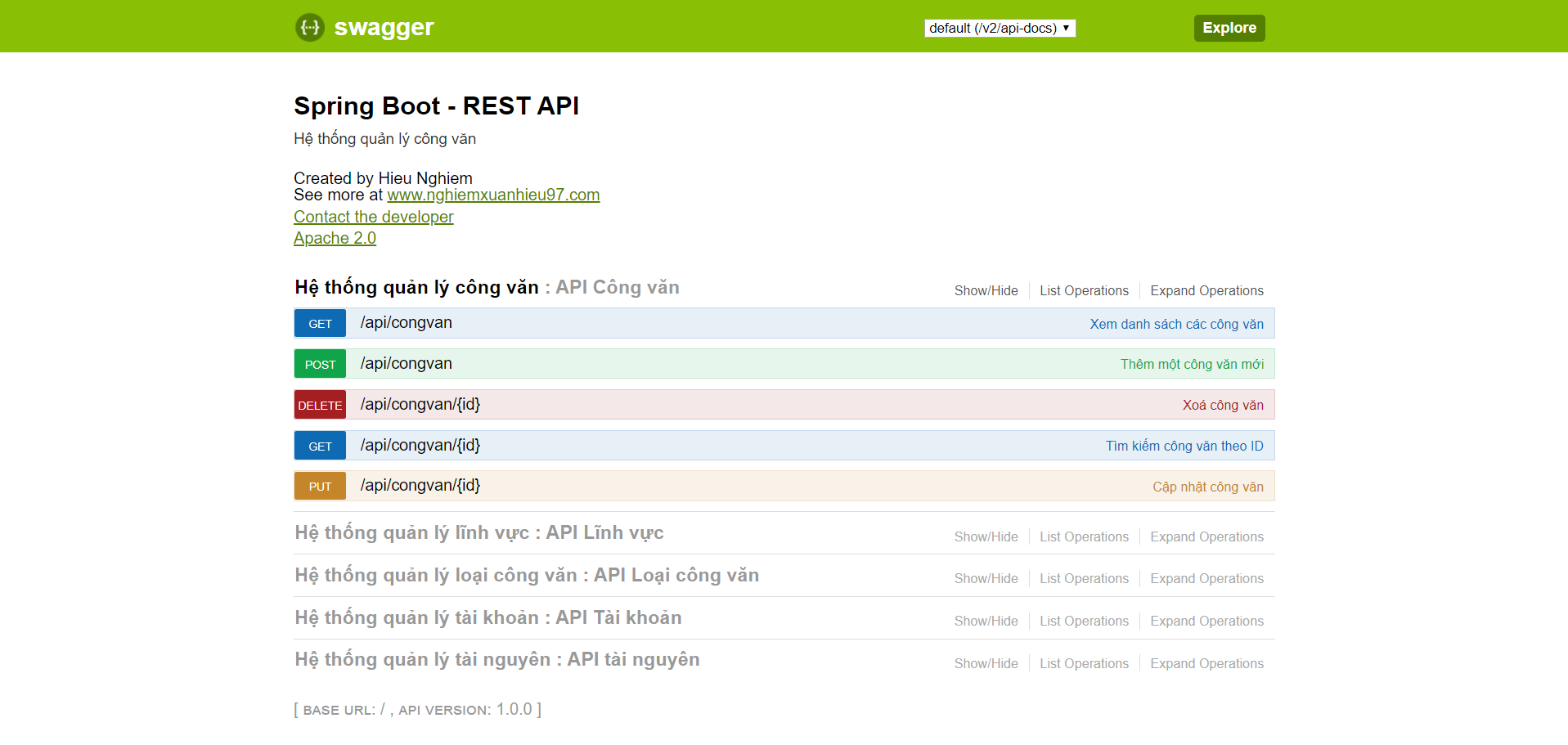
Hình 64. Tuỳ chọn cài đặt cho VSCode

* Sau đó tiếp tục nhấn “Next” cho tới khi quá trình cài đặt diễn ra và hoàn tất.

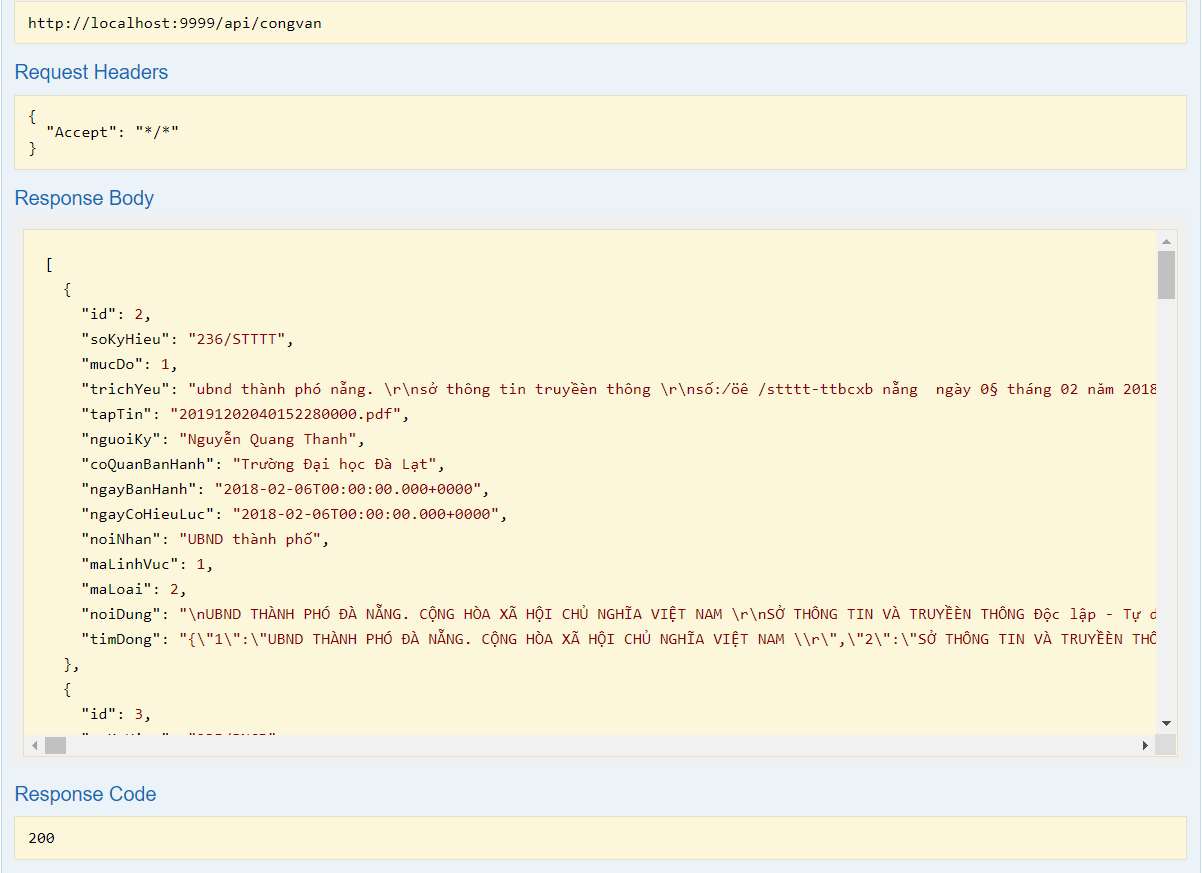


Hình 65. Hoàn tất quá trình cài đặt VSCode

* 1. **Triển khai hệ thống**
     1. **Triển khai trên từng module nhỏ**
* Triển khai các module với các chức năng: thêm, xoá, sửa, lấy danh sách, lấy một đối tượng.
* Chúng ta bắt đầu khởi động server cho hệ thống về phía backend và frontend.
* Chúng ta mở giao diện người dùng Swagger bên phía backend để xem và kiểm tra các API[5] của hệ thống

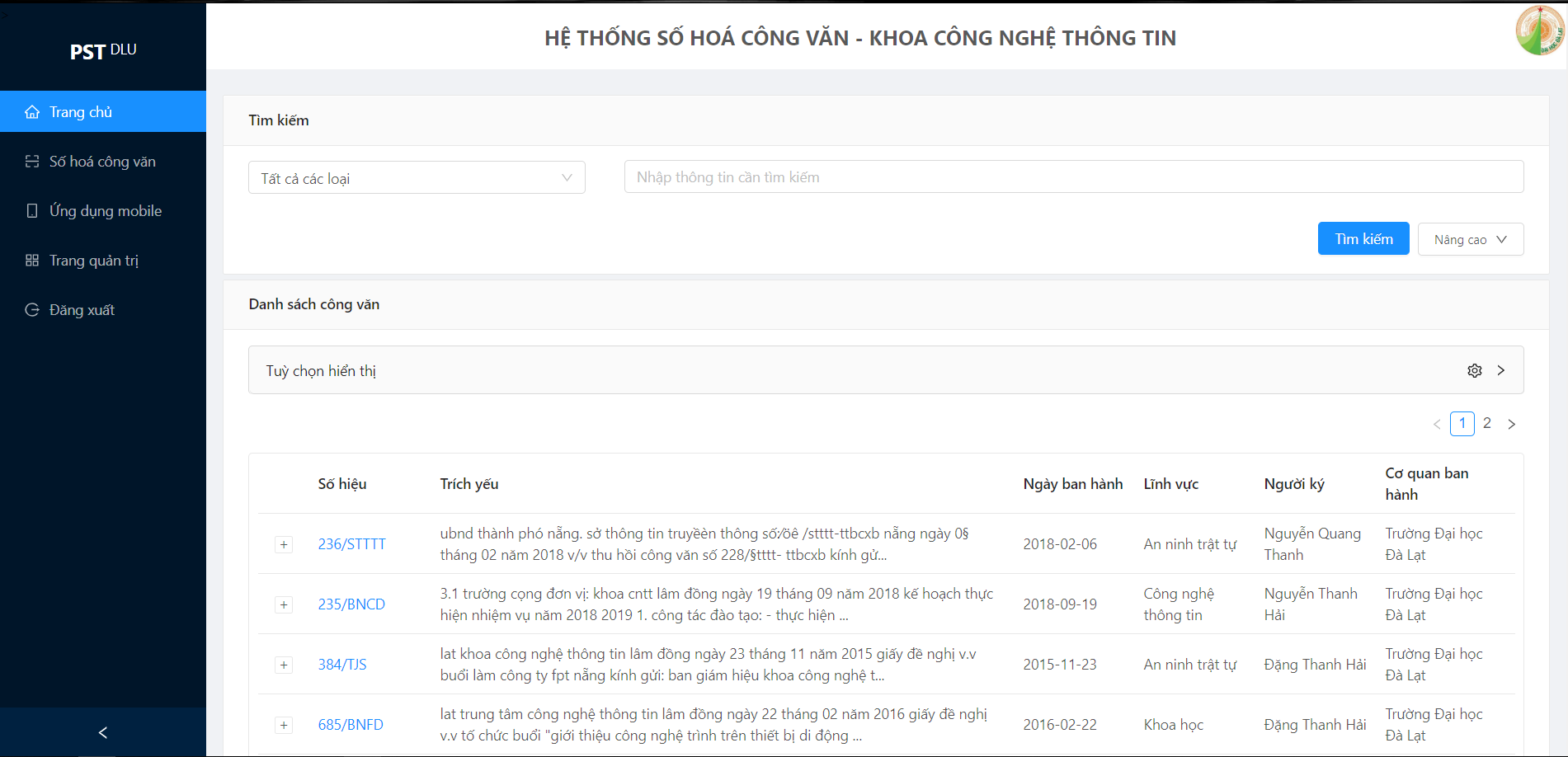


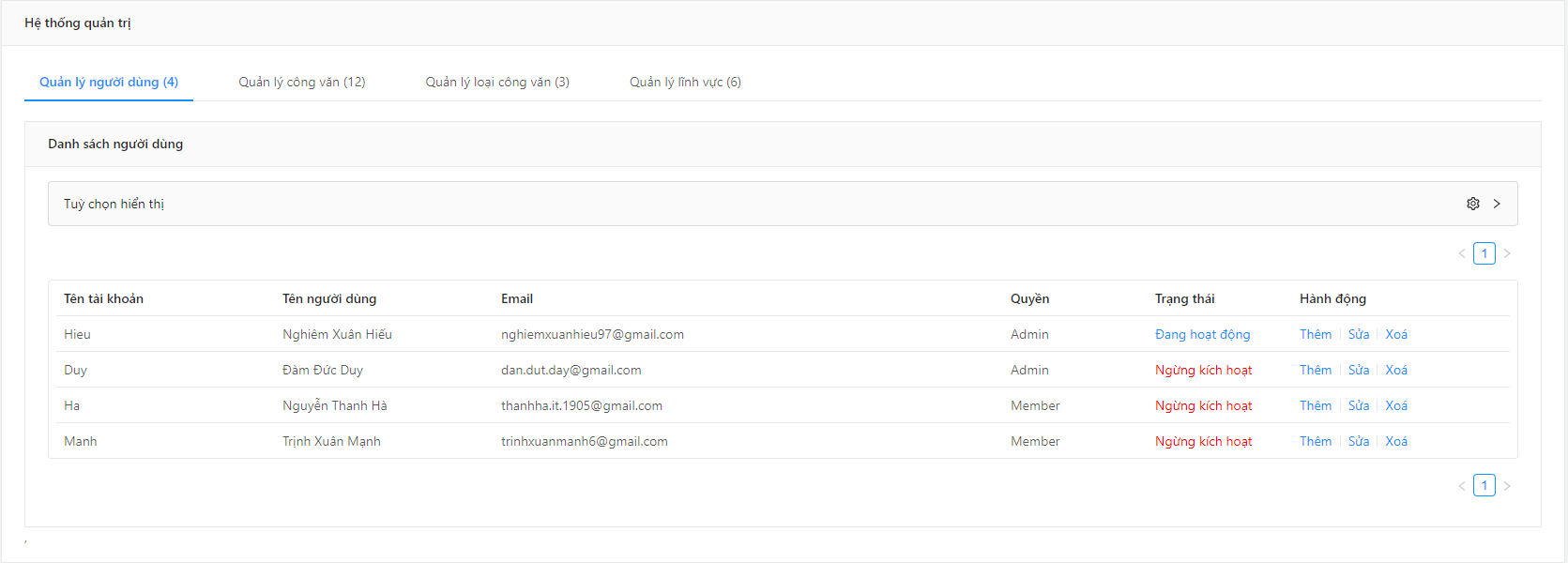
Hình 66. Giao diện quản lý API - Swagger



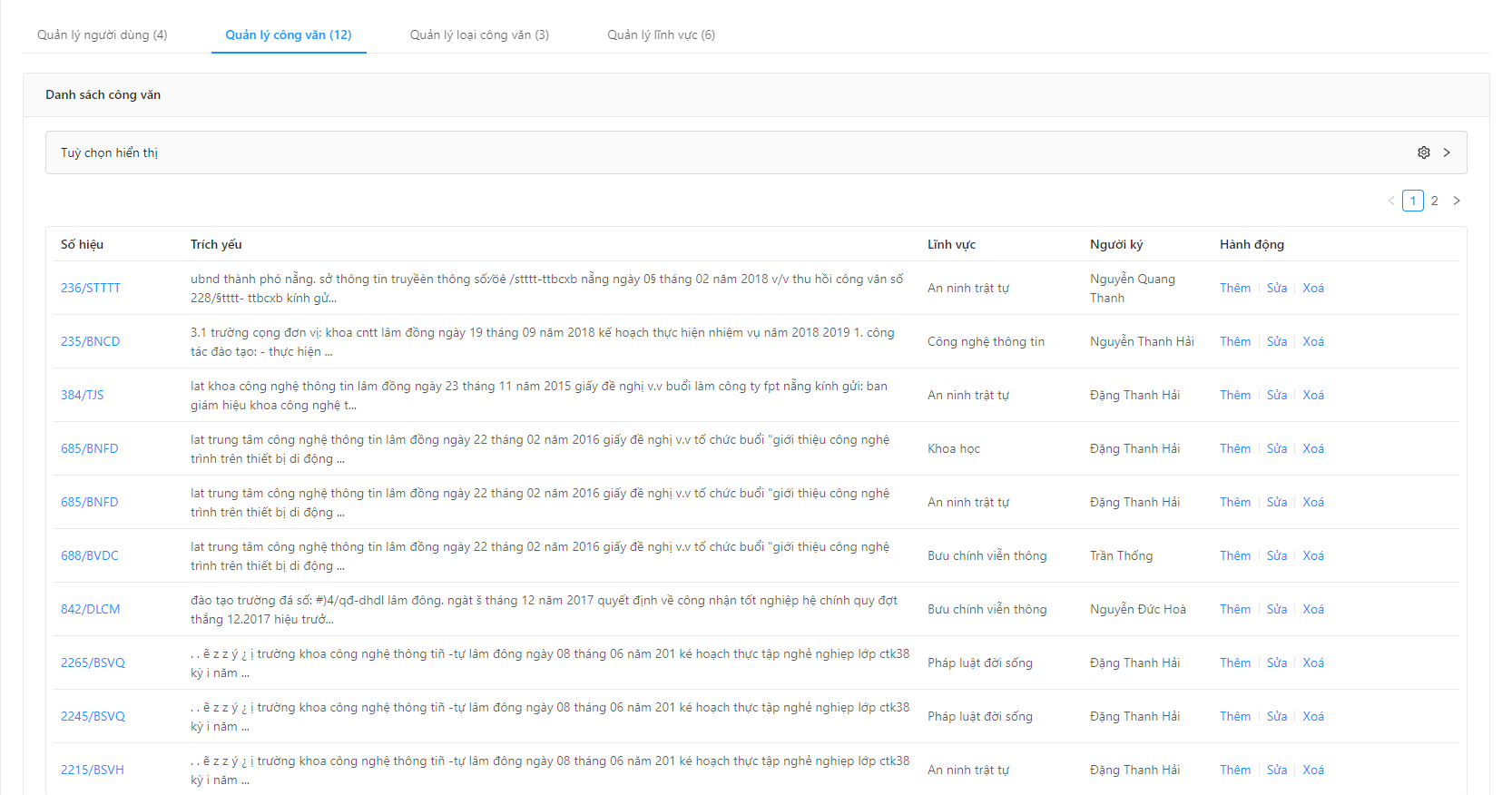
Hình 67. Kiểm tra API - Endpoint lấy danh sách công văn

* + 1. **Giao diện hệ thống**



Hình 68. Giao diện hệ thống

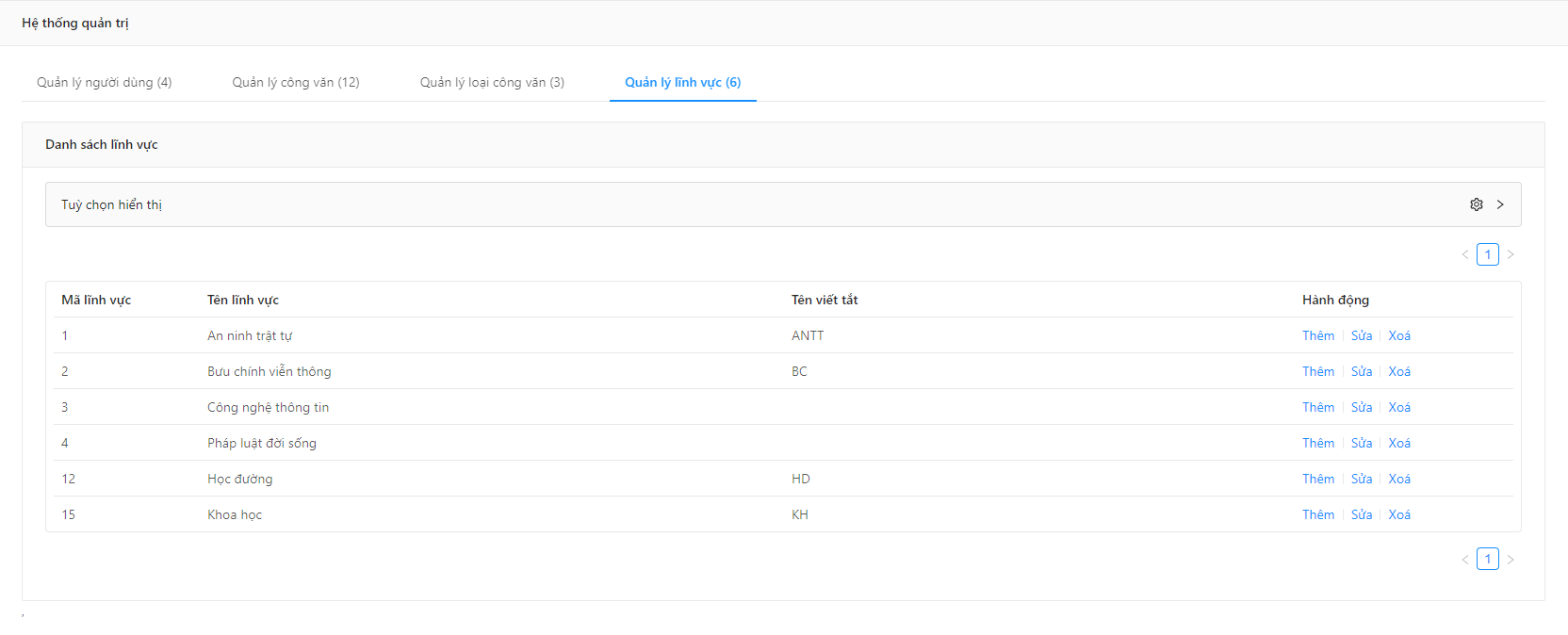
Hình 69.Giao diện quản lý người dùng



Hình 70.Giao diện quản lý công văn

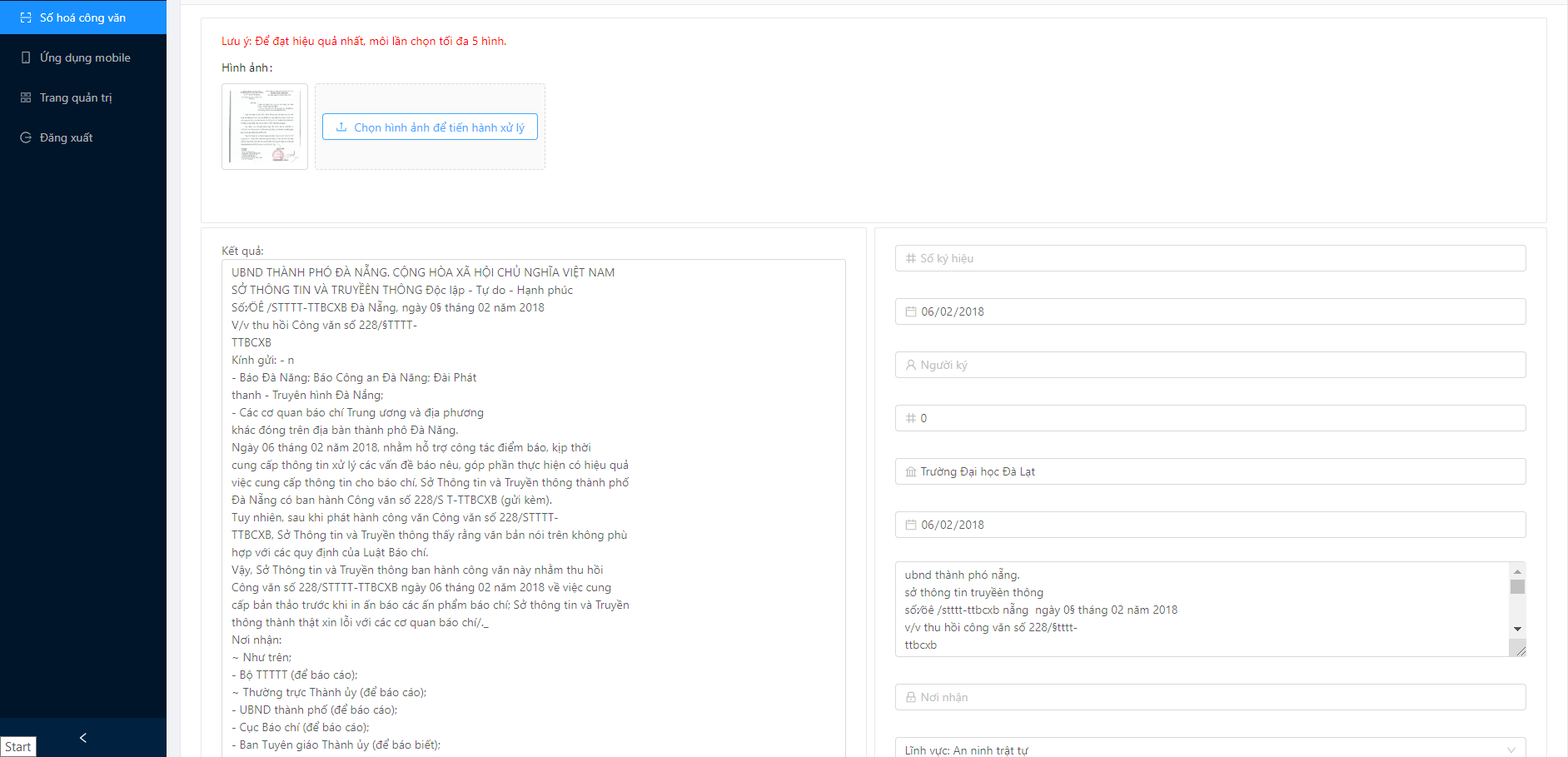


Hình 71.Giao diện quản lý loại công văn



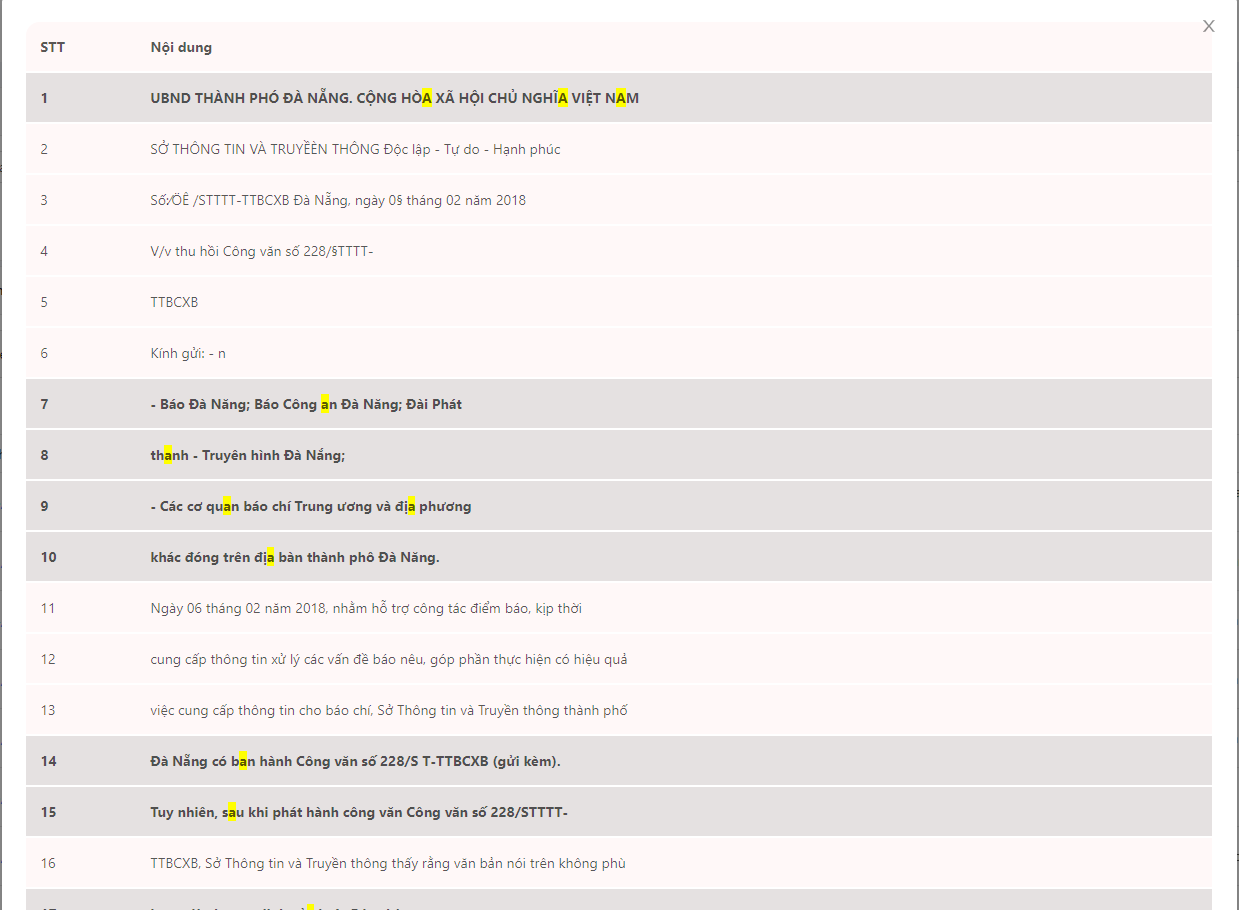
Hình 72.Giao diện quản lý lĩnh vực

* + 1. **Các chức năng chính**
       1. **Số hoá công văn**

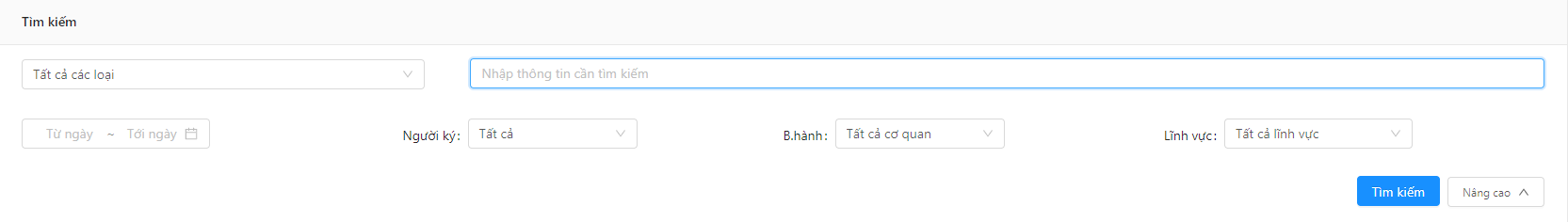


Hình 73.Giao diện chức năng số hoá công văn

* + - 1. **Tìm kiếm từ khoá**



Hình 74. Kết quả tìm kiếm từ khoá “a”



Hình 75. Giao diện tìm kiếm

# **Chương 3: XÂY DỰNG ỨNG DỤNG**

Sau một thời gian dài nghiên cứu và thực hiện, hệ thống của chúng em đã đạt được những kết quả đáng mong đợi như sau:

* + Hoàn thiện các chức năng cơ bản như: thêm, xoá, sửa, hiển thị
  + Giao diện hiện đại, đẹp mắt và đơn giản, gần gũi với người sử dụng.
  + Hoàn thiện các chức năng nâng cao như: nhận dạng ký tự quang học, phân loại tự động, rút trích, tìm kiếm
  + Xây dựng hệ thống độc lập về backend và frontend khiến cho việc bảo trì hệ thống cũng như nâng cấp sau này dễ dàng hơn.

Ngoài ra, đề tài này còn giúp chúng em ôn lại kiến thức đã học ở trường và áp dụng nó vào thực tế cũng như bước đầu giúp chúng em có thêm nhiều kinh nghiệm cho việc làm sau này.

# **KẾT LUẬN**

Trong thời kỳ mà ngành công nghệ thông tin đang phát triển mạnh mẽ như hiện nay thì việc áp dụng tiến bộ khoa học kỹ thuật vào ứng dụng thực tế luôn được chú trọng và quan tâm hàng đầu. Không những phát huy các điểm tốt, khắc phục, sửa chữa những khuyết điểm như: giúp tiết kiệm thời gian, công sức, chi phí và đạt được nhiều công dụng khác mà các phương pháp truyền thống sử dụng sức người không thể có được.

Trong chương 1, chúng em đã tập trung vào nghiên cứu các vấn đề thực tiễn, công việc quản lý công văn hiện tại của khoa, thu thập và phân tích các mẫu công văn, liệt kê ra các danh sách công việc cần phải làm. Bên cạnh đó chúng em cũng nghiên cứu tìm hiểu về công nghệ nhận dạng ký tự quang học để giải quyết bài toán số hoá công văn từ công văn vật lý thành công văn điện tử có thể lưu trữ vào các hệ thống số hoá.

Trong chương 2, chúng em bám sát các các bài toán đã tìm được ở chương đầu để phân tích và thiết kế hệ thống. Qua đó chúng em cũng trình bày về các công nghệ mà chúng em đã tìm hiểu nghiên cứu và cách cài đặt chúng để áp dụng vào triển khai phát triển dự án.

Cuối cùng, sau khoảng thời gian hơn ba tháng, với sự giúp đỡ tận tình từ giáo viên hướng dẫn cùng các giáo viên bộ môn. Chúng em phát triển thành công hệ thống số hoá và rút trích thông tin công văn với mong muốn giúp thầy cô quản lý công văn dễ dàng hơn và tạo ra hệ thống số hoá, rút trích, phân loại và tìm kiếm công văn.

Ưu điểm:

* Giúp người sử dụng có thể kiểm soát toàn bộ hệ thống: quản lý công văn, quản lý loại công văn, quản lý người dùng…
* Xây dựng hệ thống back-end hỗ trợ API[5] giúp dễ dàng triển khai hệ thống trên nền tảng khác: mobile, windows, website.
* Xây dựng hệ thống theo hướng mở: back-end và front-end được thiết kế hoàn toàn riêng biệt rất ít giảm sự phụ thuộc lẫn nhau.
* Tách hệ thống thành các module nhỏ: giúp dễ dàng phát triển và bảo trì hệ thống.
* Tối ưu hoá hiệu suất: sử dụng các module phiên bản mới nhất, tối ưu hoá code, tìm kiếm theo chỉ mục…
* Giao diện thân thiện dễ sử dụng: thiết kế giao diện tối giản thân thiện dễ dàng làm quen và sử dụng.

Nhược điểm

* Hệ thống đã phát triển rất tốt song vẫn chưa thể đáp ứng được một số nhu cầu thực tế do nhiều yếu tố khách quan: lỗi chính tả, ngữ nghĩa, không nhận dạng được chữ viết tay, chữ ký…
* Gặp khó khăn khi scan hình ảnh công văn chất lượng kém: hình bị mờ, nhoè màu, nét chữ không rõ…

Trên đây là những bộ kết quả đạt được thông qua nghiên cứu đề tài. Mặc dù đã cố gắng rất nhiều nhưng do kiến thức và kinh nghiệm hạn hẹp của cá nhân, trình độ hiểu biết nên bài báo cáo này không thể tránh khỏi những điều sai sót. Mong các quý thầy cô chỉ dạy để khoá luận của chúng em đạt kết quả tốt hơn.

# **HƯỚNG PHÁT TRIỂN**

Với những kết quả đạt được, đề tài chúng em dự kiến sẽ tiếp tục với những định hướng sau:

* Triển khai hệ thống vào thực tế áp dụng tại Khoa Công nghệ Thông tin.
* Phát triển trên nền tảng ứng dụng di động.
* Phát triển các chức năng bảo mật.
* Thêm vào các chức năng tiện dụng như: báo cáo, thống kê tự động, dự đoán quản lý…

Mặc dù chúng em đã cố gắng hoàn chỉnh các yêu cầu của bài báo cáo nhưng đề tài khoá luận vẫn còn nhiều thiếu sót, mong nhận được sự chỉ bảo tận tình cũng như hướng dẫn của các thầy cô, đề xuất thêm ý kiến cũng như bổ sung thêm các vấn đề cần khắc phục của hệ thống để việc thực hiện khoá luận được hoàn thiện hơn. Chúng em xin chân thành cảm ơn quý thầy cô!

# **DANH MỤC TÀI LIỆU THAM KHẢO**

**Ebook**

[1] Timothy C. NeedHam, Learning Java – A Crash Course Guide to Learn Java in 1 Week

[2] Christian Bauer Gavin King, Java Persistence With Hibernate, Second Edition

**Nguồn tài nguyên mạng**

* Wikipedia
* Youtube
* StackOverFlow
* Spring (Homepage)
* React (Homepage)
* Swagger (Homepage)