|  |
| --- |
| TRƯỜNG ĐẠI HỌC TIỀN GIANG  **KHOA CÔNG NGHỆ THÔNG TIN**  LÊ LÂM KHÁNH DUY - 015101007  **NGHIÊN CỨU CÔNG NGHỆ MICROSERVICES**  **ỨNG DỤNG LẬP TRÌNH HỆ THỐNG CHỈ ĐẠO TUYẾN**  **VÀ NGHIÊN CỨU KHOA HỌC CHO NGÀNH Y TẾ**  KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC  NGÀNH CÔNG NGHỆ THÔNG TIN  NGƯỜI HƯỚNG DẪN:  LÊ ĐẶNG ĐĂNG KHOA  DƯƠNG VĂN HIẾU  KHÓA 2015 – 2019 |

**LỜI NÓI ĐẦU**

Để hỗ trợ quy trình chuyển chỉ đạo tuyến từ các bệnh viện tuyến trên xuống các bệnh viện tuyến dưới và hỗ trợ quy trình quản lý NCKH cho ngành y tế, em chọn đề tài “Nghiên cứu công nghệ Microservices ứng dụng lập trình hệ thống chỉ đạo tuyến và nghiên cứu khoa học cho ngành y tế”.

**LỜI CẢM ƠN**

🕮

Đầu tiên em xin chân thành cảm ơn Ban Giám Hiệu cùng quý thầy cô trường Đại học Tiền Giang đặc biệt là toàn thể quý thầy cô trong khoa Công nghệ thông tin đã tận tình giảng dạy, trang bị cho em những kiến thức quý báu trong quá trình học tập tại trường để hoàn thành tốt khóa luận tốt nghiệp.

Tiếp theo, em xin chân thành cảm ơn sự giúp đỡ tận tình của các Anh/Chị trong công ty VNPT, hỗ trợ, giúp em tiếp cận với công nghệ Microservices, hướng dẫn em các quy trình của chỉ đạo tuyến và nghiên cứu khoa học cho ngành y tế.

Để hoàn thành tốt khóa luận tốt nghiệp này, em xin tỏ lòng biết ơn sâu sắc và lời cám ơn chân thành đến thầy Dương Văn Hiếu – người đã tạo điều kiện để em có thể làm khóa luận theo đề tài của công ty VNPT. Bên cạnh đó là những ý kiến đóng góp của bạn bè đã cho em nguồn động viên lớn để hoàn thành nhiệm vụ của khóa luận. Qua đó, em đã đạt được nhiều tiến bộ về kiến thức cũng như những kĩ năng làm việc bổ ích.

Mặc dù đã rất cố gắng trong quá trình làm khóa luận tốt nghiệp , nhưng chắc chắn cũng không thể tránh khỏi những thiếu sót. Em rất mong nhận được sự thông cảm và góp ý tận tình của quý thầy cô và các bạn. Em xin chân thành cảm ơn!

Cuối cùng em kính chúc quý Thầy/Cô và quý Anh/Chị trong công ty VNPT dồi dào sức khỏe và thành công trong công việc.

Em xin chân thành cảm ơn!

Tiền Giang , ngày 21 tháng 6 năm 2019

Sinh viên thực hiện

Lê Lâm Khánh Duy

**ĐỀ CƯƠNG CHI TIẾT KHÓA LUẬN TỐT NGHIỆP**

**Tên đề tài**: Nghiên cứu công nghệ Microservices ứng dụng lập trình hệ thống chỉ đạo tuyến và NCKH cho ngành y tế.

**Người hướng dẫn:** Lê Đặng Đăng Khoa

Dương Văn Hiếu

**Thời gian thực hiện:** 14/03/2019 – 21/06/2019.

**Sinh viên thực hiện:** Lê Lâm Khánh Duy MSSV: 015101007

**Loại đề tài:** Xây dựng ứng dụng

**Nội dung đề tài:**

* **Mục tiêu đề tài:**
  + Tìm hiểu về Microservices.
  + Tìm hiểu về ReactJS, Typescript, Redux, JasperReports…và các ngôn ngữ liên quan.
  + Tìm hiểu quy trình chỉ đạo tuyến, quy trình đăng ký, xét duyệt đề tài nghiên cứu khoa học cho ngành y tế.
  + Rèn luyện kỹ năng phân tích và giải quyết vấn đề.
  + Ứng dụng những kiến thức đã học và kiến thức đã tìm hiểu để xây dựng một hệ thống.
* **Kết quả cần đạt được:**
  + Công dụng: Hệ thống quản lý các thông tin cần lưu trữ của chỉ đạo tuyến và đề tài NCKH ngành y tế
  + Chức năng:
    - Đăng nhập dựa trên mô hình Microservices
    - Quản lý thông tin của chỉ đạo tuyến, thống kê danh sách văn bản chỉ đạo tuyến, kế hoạch chỉ đạo tuyến, thông tin chỉ đạo tuyến hàng năm.
    - Quản lý thông tin của đề tài NCKH, thống kê danh sách đề tài đã đăng ký, đề tài được công nhận đề tài năm, đề tài được xét duyệt ứng dụng vào hoạt động của đơn vị hàng năm ở đơn vị.
  + Đặc điểm chính:
    - Hệ thống tích hợp hai phân hệ là chỉ đạo tuyến và NCKH cho ngành y tế.
    - Hệ thống hoạt động hiệu quả và chính xác.
* **Nhiệm vụ đề tài:**
  + **Nghiên cứu thực tế:** Ứng dụng công nghệ Microservices để thiết kế hệ thống, kết hợp sử dụng ReactJS, Redux, Typescript, JasperReports giúp người lập trình có thêm kiến thức mới hỗ trợ lập trình web mạnh mẽ hơn.
  + **Xác định phạm vi đề tài:** Đề tài để phục vụ các bệnh viện về quy trình chỉ đạo tuyến và NCKH ở bệnh viện.
  + **Nghiên cứu tài liệu chuyên môn:** Trong quá trình thực hiện đề tài đã tham khảo một số tài liệu về lập trình Microservices, JavaScript, CSS, ReactJS, Redux, Typescript, JasperReports…
* **Phân tích, thiết kế, cài đặt (hiện thực), thử nghiệm:**
  + Phân tích quy trình chỉ đạo tuyến, quy trình đăng ký, xét duyệt đề tài NCKH.
  + Thiết kế cơ sở dữ liệu.
  + Tìm hiểu về ReactJS, Redux, Typescript, JasperReports....
  + Lập trình Font-end.
  + Lập trình Back-end.
  + Thiết kế các báo cáo, thống kê.
  + Chạy thử và sửa lỗi.
  + Báo cáo và rút kinh nghiệm.
* **Yêu cầu khoa học công nghệ:**
  + Hệ điều hành: Windows
  + Trình soạn thảo mã nguồn: IntelliJ IDEA 2018
  + Trình quản lý CSDL: MySQL Workbench 8.0 CE
  + Trình duyệt web: Google Chrome, IE,....

**Kế hoạch thực hiện**

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Các nội dung,công việc chủ yếu cần thực hiện**  (các mốc đánh giá chủ yếu) | **Từ ngày** | **Đến ngày** |
| 1 | Phân tích đề tài | 14/03/2019 | 21/03/2019 |
| 2 | Thiết kế CSDL | 22/03/2019 | 29/03/2019 |
| 3 | Tìm hiểu về ReactJS, Redux, Typescript, JasperReports.... | 30/03/2019 | 13/04/2019 |
| 4 | Lập trình Front-end | 14/04/2019 | 05/05/2019 |
| 5 | Lập trình Back-end | 06/05/2019 | 06/06/2019 |
| 6 | Thiết kế các báo cáo, thống kê | 07/06/2019 | 10/06/2019 |
| 7 | Chạy thử và sửa lỗi | 11/06/2019 | 15/06/2019 |
| 8 | Viết báo cáo và sửa lỗi đề tài | 15/06/2019 | 20/06/2019 |

**Xác nhận của người hướng dẫn Ngày……tháng……năm 2019**

**Sinh viên thực hiện**

**Lê Lâm Khánh Duy**

**MỤC LỤC**

[Chương 1. TỔNG QUAN 1](#_Toc11881428)

[1.1. Giới thiệu đề tài 1](#_Toc11881429)

[1.1.1. Lý do chọn đề tài 1](#_Toc11881430)

[1.1.2. Mục tiêu đề tài 1](#_Toc11881431)

[1.1.3. Nhiệm vụ 1](#_Toc11881432)

[1.1.4. Phạm vi nghiên cứu của đề tài 2](#_Toc11881433)

[1.3.Các vấn đề tập trung giải quyết 2](#_Toc11881434)

[Chương 2: CƠ SỞ LÝ THUYẾT VÀ THỰC TIỄN 4](#_Toc11881435)

[2.1. Cơ sở lý thuyết 4](#_Toc11881436)

[2.1.1. Giả thuyết nghiên cứu 4](#_Toc11881437)

[2.1.2. Phương pháp nghiên cứu 4](#_Toc11881438)

[2.1.3. Công nghệ sử dụng 5](#_Toc11881439)

[2.1.4. Cơ sở lý thuyết 5](#_Toc11881440)

[2.2. Cơ sở thực tiễn và ứng dụng của khóa luận 25](#_Toc11881441)

[2.2.1. Bối cảnh thực tiễn 25](#_Toc11881442)

[2.2.2. Ứng dụng 26](#_Toc11881443)

[Chương 3: PHÂN TÍCH, THIẾT KẾ, CÀI ĐẶT GIẢI PHÁP 27](#_Toc11881444)

[3.1. Phân tích yêu cầu 27](#_Toc11881445)

[3.1.1. Phân tích yêu cầu chức năng 27](#_Toc11881446)

[3.1.2. Phân tích yêu cầu phi chức năng 28](#_Toc11881447)

[3.1.3. Phân tích quy trình 28](#_Toc11881448)

[3.3. Phân tích thiết kế hệ thống 31](#_Toc11881449)

[3.3.1. Hệ thống chỉ đạo tuyến 31](#_Toc11881450)

[3.3.2. Hệ thống nghiên cứu khoa học 41](#_Toc11881451)

[3.4. Cài đặt giải pháp 55](#_Toc11881452)

[3.4.1. Xây dựng cấu trúc Microservice cho hệ thống chỉ đạo tuyến và hệ thống NCKH 55](#_Toc11881453)

[3.4.2. Cài đặt hệ thống 59](#_Toc11881454)

[3.4.3 Một số hình ảnh giao diện hệ thống 62](#_Toc11881455)

[Chương 4: KẾT QUẢ VÀ HƯỚNG PHÁT TRIỂN 66](#_Toc11881456)

[4.1. Kết quả đạt được 66](#_Toc11881457)

[4.2. Hạn chế 66](#_Toc11881458)

[4.3. Hướng phát triển 66](#_Toc11881459)

[DANH MỤC TÀI LIỆU THAM KHẢO 67](#_Toc11881460)

**DANH MỤC HÌNH**

[Hình 2.1. Minh họa mô hình kiến trúc nguyên khối 6](#_Toc11879996)

[Hình 2.2. Minh họa mô hình Microservices 7](#_Toc11879997)

[Hình 2.3. Mô hình Virtual DOM 10](#_Toc11879998)

[Hình 2.4. Giới thiệu về Props và State 12](#_Toc11879999)

[Hình 2.5. Mô phỏng Redux cho ứng dụng 12](#_Toc11880000)

[Hình 2.6. Mô hình hoạt động của Redux 14](#_Toc11880001)

[Hình 2.7. Minh họa TypeScript 15](#_Toc11880002)

[Hình 2.8. Các bước cài đặt JasperSoft Studio 16](#_Toc11880003)

[Hình 2.9. Các bước cài đặt JasperSoft Studio 16](#_Toc11880004)

[Hình 2.10. Các bước cài đặt JasperSoft Studio 17](#_Toc11880005)

[Hình 2.11. Các bước tạo báo cáo bằng JasperSoft Studio 18](#_Toc11880006)

[Hình 2.12. Các bước tạo báo cáo bằng JasperSoft Studio 18](#_Toc11880007)

[Hình 2.13. Các bước tạo báo cáo bằng JasperSoft Studio 18](#_Toc11880008)

[Hình 2.14. Các bước tạo báo cáo bằng JasperSoft Studio 19](#_Toc11880009)

[Hình 2.15. Các bước tạo báo cáo bằng JasperSoft Studio 20](#_Toc11880010)

[Hình 2.16. Các bước tạo báo cáo bằng JasperSoft Studio 20](#_Toc11880011)

[Hình 2.17. Các bước tạo báo cáo bằng JasperSoft Studio 21](#_Toc11880012)

[Hình 2.18. Các bước tạo báo cáo bằng JasperSoft Studio 21](#_Toc11880013)

[Hình 2.19. Các bước tạo báo cáo bằng JasperSoft Studio 22](#_Toc11880014)

[Hình 2.20. Các bước tạo báo cáo bằng JasperSoft Studio 22](#_Toc11880015)

[Hình 2.21. Các bước tạo báo cáo bằng JasperSoft Studio 23](#_Toc11880016)

[Hình 2.22. Các bước tạo báo cáo bằng JasperSoft Studio 24](#_Toc11880017)

[Hình 2.23. Minh họa JHipster 24](#_Toc11880018)

[Hình 2.24. Minh họa tạo dự án JHipster 25](#_Toc11880019)

[Hình 3.1. Sơ đồ Use case tổng quát (cán bộ phụ trách chỉ đạo tuyến) 32](#_Toc11880030)

[Hình 3.2. Sơ đồ Use case tổng quát (nhân viên đơnvị tiếp nhận chỉ đạo tuyến) 32](#_Toc11880031)

[Hình 3.3. Sơ đồ lớp 40](#_Toc11880032)

[Hình 3.4. Mô hình cơ sở dữ liệu 41](#_Toc11880033)

[Hình 3.5. Sơ đồ Use case tổng quát (Nhân viên quản lý NCKH) 41](#_Toc11880035)

[Hình 3.6. Sơ đồ Use case tổng quát (Nhân viên đơn vị) 42](#_Toc11880036)

[Hình 3.7. Sơ đồ Use case tổng quát (Cán bộ hội đồng NCKH) 42](#_Toc11880037)

[Hình 3.8. Sơ đồ Use case tổng quát (Cán bộ đơn vị NCKH) 43](#_Toc11880038)

[Hình 3.9. Sơ đồ lớp 54](#_Toc11880039)

[Hình 3.10. Mô hình CSDL 55](#_Toc11880040)

[Hình 3.11. Xây dựng cấu trúc cho hệ thống 55](#_Toc11880043)

[Hình 3.12. Xây dựng cấu trúc cho hệ thống 56](#_Toc11880044)

[Hình 3.13. Xây dụng cấu trúc cho hệ thống 56](#_Toc11880045)

[Hình 3.14. Xây dựng cấu trúc cho hệ thống 57](#_Toc11880046)

[Hình 3.15. Xây dụng cấu trúc hệ thống 58](#_Toc11880047)

[Hình 3.16. Xây dựng cấu trúc hệ thống 58](#_Toc11880048)

[Hình 3.17. Xây dụng cấu trúc hệ thống 59](#_Toc11880049)

[Hình 3.18. Thêm dịch vụ vào IntteliJ IDEA 2018 59](#_Toc11880051)

[Hình 3.19. Thêm dịch vụ vào IntteliJ IDEA 2018 60](#_Toc11880052)

[Hình 3.20. Thêm dịch vụ vào IntteliJ IDEA 2018 61](#_Toc11880053)

[Hình 3.21. Chạy dịch vụ trong IntteliJ IDEA 2018 61](#_Toc11880054)

[Hình 3.22. Chạy dịch vụ trong IntteliJ IDEA 2018 62](#_Toc11880055)

[Hình 3.23. Chạy dịch vụ trong IntteliJ IDEA 2018 62](#_Toc11880056)

[Hình 3.24. Dịch vụ đã được chạy 62](#_Toc11880057)

[Hình 3.25.Giao diện đăng nhập 63](#_Toc11880059)

[Hình 3.26. Giao diện chính 63](#_Toc11880060)

[Hình 3.27. Giao diện Quản lý nội dung chỉ đạo tuyến 63](#_Toc11880061)

[Hình 3.28. Giao diện thêm mới nội dung chỉ đạo tuyến 64](#_Toc11880062)

[Hình 3.29. Giao diện Xem thông tin nội dung chỉ đạo tuyến 64](#_Toc11880063)

[Hình 3.30. Giao diện Cập nhật thông tin nội dung chỉ đạo tuyến 64](#_Toc11880064)

[Hình 3.31. Giao diện Xóa nội dung chỉ đạo tuyến 65](#_Toc11880065)

[Hình 3.32. Giao diện Thống kê chỉ đạo tuyến 65](#_Toc11880066)

[Hình 3.33. Giao diện Xuất báo cáo chỉ đạo tuyến 65](#_Toc11880067)

**BẢNG CÁC TỪ VIẾT TẮT**

|  |  |
| --- | --- |
| **Từ viết tắt** | **Mô tả** |
| CSDL | Cơ sở dữ liệu |
| NCKH | Nghiên cứu khoa học |
| API | Application Programming Interface (giao diện chương trình ứng dụng) |
| Microservices | Mô hình chia nhỏ dịch vụ |

**TÓM TẮT KHÓA LUẬN**

**1. Mục tiêu**

Trong tìm hiểu: Hiểu được cách hoạt động của Microservices, nắm được các kiến thức nền tảng của lập trình sử dụng ReactJS, Redux, TypeScript,…Nắm rõ quy trình chỉ đạo tuyến và quy trình đăng ký, xét duyệt đề tài NCKH cho ngành y tế.

Trong thực nghiệm: Xây dựng thành công hệ thống Microservices cho hệ thống chỉ đạo tuyến và nghiên cứu khoa học.

**2. Nhiệm vụ**

Trong tìm hiểu: Tìm hiểu cách hoạt động của Microservices, tìm hiểu lập trình sử dụng ReactJS, Redux, TypeScript,…Tìm hiểu quy trình chỉ đạo tuyến và quy trình đăng ký, xét duyệt đề tài NCKH cho ngành y tế.

Trong thực nghiệm: Đặt ra và phân tích yêu cầu, liệt kê các chức năng cần thiết, thiết kế cơ sở dữ liệu, xây dựng cấu trúc ứng dụng, cấu hình ứng dụng, thiết kế giao diện, thiết kế các báo cáo, thống kê, cài đặt giải thuật và hiện thực các chức năng đã liệt kê.

**3. Phương pháp giải quyết**

Trong tìm hiểu: Phương pháp nghiên cứu lý thuyết, mô hình hóa, sơ đồ.

Trong thực nghiệm: Phương pháp thực nghiệm khoa học, phân tích và tổng kết kinh nghiệm.

**4. Kết quả đạt được**

Trong tìm hiểu: Hiểu được cách hoạt động của Microservices, nắm được các kiến thức nền tảng của lập trình sử dụng ReactJS, Redux, TypeScript,…Nắm rõ quy trình chỉ đạo tuyến và quy trình đăng ký, xét duyệt đề tài NCKH cho ngành y tế.

Trong thực nghiệm: Xây dựng thành công hệ thống đáp ứng các chức năng đã đặt ra.

# 

# Chương 1. TỔNG QUAN

## **1.1. Giới thiệu đề tài**

### **1.1.1. Lý do chọn đề tài**

Hiện tại với tốc độ phát triển vượt trội của thông tin truyền thông và Internet, việc sử dụng công nghệ mới để thiết kế web cũng được các doanh nghiệp quan tâm. Trong số những công nghệ lập trình web hiện nay, Microservices được quan tâm đáng kể với nhiều ưu điểm về bảo mật và ổn định. Chính vì có ưu điểm về bảo mật cao nên được sử dụng nhiều trong y tế, chỉ đạo tuyến và NCKH là một bộ phận trong y tế, có vai trò quan trọng, cần phải được quản lý một cách bào mật. Xuất phát từ đó, em chọn đề tài “Nghiên cứu công nghệ Microservices ứng dụng lập trình hệ thống chỉ đạo tuyến và NCKH cho ngành y tế”.

### **1.1.2. Mục tiêu đề tài**

Trong tìm hiểu:

* Hiểu rõ mô hình Microservices, cũng cố kiến thức và thành thạo kỹ năng lập trình ReactJS, Redux, Typescript,…
* Học cách viết Microservices, cách cấu trúc chương trình một cách hiệu quả.
* Nắm được quy trình chỉ đạo tuyến và đăng ký, xét duyệt đề tài NCKH cho ngành y tế.

Trong thực nghiệm:

* Phát triển hệ thống cho phân hệ chỉ đạo tuyến và NCKH cho ngành y tế.
* Ứng dụng hoạt động hiệu quả, nhanh chóng.
* Giao diện đơn giản, dễ sử dụng.
* Xây dụng ứng dụng với các phân quyền cho từng nhóm người dùng.

### **1.1.3. Nhiệm vụ**

Trong tìm hiểu:

* Tìm hiểu về mô hình Microservices, ôn tập lại các kiến thức và thực hành kỹ năng lập trình.
* Thử nghiệm thành lập một mô hình Microservices nhỏ để hiểu cách thức hoạt động.
* Tìm hiểu cách cấu hình, các thư viện hỗ trợ Microservices.
* Tìm hiểu các quy trình chỉ đạo tuyến, quy trình đăng ký, xét duyệt đề tài NCKH.

Trong thực nghiệm:

* Đặt ra và phân tích các yêu cầu dựa trên các quy trình.
* Liệt kê các chức năng cần thiết.
* Thiết kế cơ sở dữ liệu.
* Xây dựng cấu trúc ứng dụng.
* Cấu hình ứng dụng.
* Thiết kế giao diện.
* Cài đặt giải thuật và hiện thực các chức năng đã liệt kê.
* Sửa chữa, cải tiến, hoàn thiện và triển khai ứng dụng.

### **1.1.4. Phạm vi nghiên cứu của đề tài**

Phạm vi trong tìm hiểu: Để có thể sử dụng thành thạo Microservices thì cần phải tìm hiểu rất nhiều, nên trong phạm vị khóa luận này, em chỉ tập trung vào tìm hiểu những kỹ thuật cần thiết cho hệ thống ở mức độ biết cách sử dụng.

Phạm vi trong thực nghiệm: Trong phạm vi khóa luận này, ứng dụng web của em chỉ sử dụng CSDL MySQL.

**1.2. Các kết quả nghiên cứu liên quan tới đề tài**

Hệ thống chỉ đạo tuyến cũ:

* + Sử dụng công nghệ trước đây để thiết kế, chưa đảm bảo bảo mật
  + Chưa đáp ứng nhu cầu hiện tại

Hệ thống quản lý NCKH cũ:

* + Sử dụng công nghệ trước đây để thiết kế, chưa đảm bảo bảo mật
  + Quy trình đăng ký, xét duyệt được cập nhật nên hệ thống chưa đáp ứng kịp

## **1.3.Các vấn đề tập trung giải quyết**

Giải quyết các vấn đề sau:

* + Thiết kế theo kiến trúc Microservice, đảm bảo tính bảo mật, hoạt động ổn định.
  + Giao diện đơn giản, dễ sử dụng.
  + Bố cục hợp lí, các menu rõ ràng.
  + Có thể chạy trên các trình duyệt web trên máy tính.

**Chương 2: CƠ SỞ LÝ THUYẾT VÀ THỰC TIỄN**

**2.1. Cơ sở lý thuyết**

**2.1.1. Giả thuyết nghiên cứu**

Nếu xây dựng thành công hệ thống sẽ giúp việc chỉ đảo tuyến và quản lý đăng ký, xét duyệt đề tài NCKH nhanh chóng, dễ dàng hơn.

### **2.1.2. Phương pháp nghiên cứu**

Phương pháp nghiên cứu lý thuyết: Nghiên cứu lý thuyết các vấn đề liên quan như cơ sở dữ liệu, ngôn ngữ lập trình, tài liệu. Nghiên cứu tài liệu:

* Tài liệu: “THÔNG TƯ QUY ĐỊNH VỀ CHỈ ĐẠO TUYẾN TRONG LĨNH VỰC KHÁM BỆNH, CHỮA BỆNH” của bộ Y tế năm 2016.
* Tài liệu: “QUY TRÌNH NGHIỆM THU ĐỀ TÀI NGHIÊN CỨU KHOA HỌC CẤP CƠ SỞ QT.33.HT” của bệnh viện Bạch Mai.

Phương pháp quan sát khoa học: Quan sát khách quan các vấn đề liên quan đến quy trình chỉ đạo tuyến, quy trình đăng ký, xét duyệt đề tài NCKH của ngành y tế.

Phương pháp thực nghiệm: Chạy thử nghiệm phần mềm.

Phương pháp nghiên cứu dựa theo quy trình: Quy trình thiết kế gồm các bước sau đây:

* Xác định yêu cầu: Thu thập thông tin từ các văn bản pháp luật.
* Phân tích: Phân tích, nghiên cứu để giải quyết vấn đề từ các yêu cầu thu thập được từ bước xác định yêu cầu , vẽ các mô hình, lớp xử lý dữ liệu của bài toán.
* Thiết kế: Thiết kế giao diện và các giải thuật, các lớp xử lí từ việc phân tích yêu cầu để có thể hoàn thành hệ thống.
* Cài đặt: Sau khi hoàn thành cơ bản chức năng của hệ thống, cần cho chạy thử nghiệm để kiểm tra, tìm lỗi và sữa lỗi...
* Kiểm chứng: thu thập, lấy ý kiến đánh giá của người sử dụng. Từ đó, tổng kết để thấy được những cái hoàn thiện, thiếu sót của hệ thống có thể chỉnh sửa kịp thời.
* Phát triển thực tế: Nếu đề tài được ứng dụng sẽ triển khai và cài đặt trên host để có thể chính thức hoạt động sẽ hỗ trợ cho quy trình chỉ đạo tuyến và đăng ký, xét duyệt đề tài NCKH.

### **2.1.3. Công nghệ sử dụng**

|  |  |
| --- | --- |
| Hệ điều hành | Windows 10 |
| Ngôn ngữ lập trình chính | ReactJS |
| Hệ quản trị cở sở dữ liệu | MySQL Workbench 8.0 CE |
| Môi trường lập trình | Trình soạn thảo mã nguồn: IntelliJ IDEA 2018 |
| Trình duyệt web | Google Chrome, Microsoft Edge |

### **2.1.4. Cơ sở lý thuyết**

**❖ Giới thiệu về Microservices**

**a. Tổng quan về Microservices**

Microservices - còn được gọi là kiến ​​trúc Microservices - là một kiểu kiến ​​trúc cấu trúc một ứng dụng như một tập hợp các dịch vụ khác nhau.

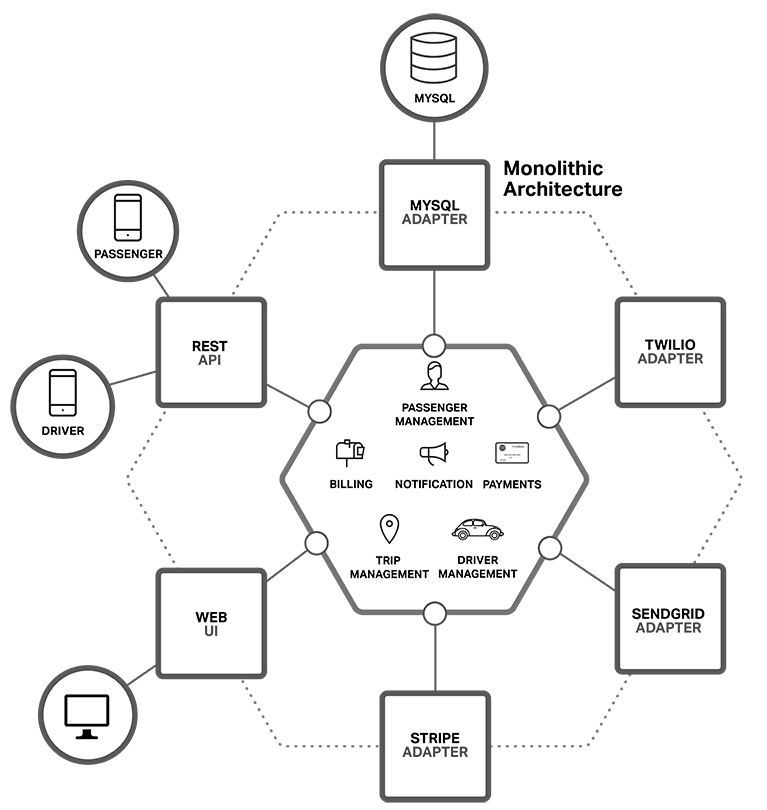
Đặc điểm của Microsevice: Có khả năng duy trì và kiểm tra cao, kết nối các thành phần riêng lẻ lại với nhau, có thể triển khai độc lập, tổ chức xung quanh khả năng kinh doanh.

Kiến trúc microservice cho phép phân phối / triển khai liên tục các ứng dụng lớn, có độ phức tạp cao.

**b. Phân biệt giữa kiên trúc nguyên khối và kiến trúc Microservices**

**Kiến trúc nguyên khối**

Người lập trình xây dựng hệ thống chỉ trong một khối cung cấp các tính năng đáp ứng nhu cầu của người sử dụng. Khi hệ thống đưa vào sử dụng, và thành công của hệ thống sẽ thu hút nhiều người sử dụng, từ đó, nhu cầu về tính năng mới của hệ thống sẽ tăng cao. Khi phải thêm nhiều tính năng mới vào hệ thống, dữ liệu sẽ tăng lên, dẫn đến hệ thống sẽ càng ngày càng phức tạp, việc bảo trỉ, nâng cấp cho hệ thống cũng sẽ gặp rất nhiều khó khăn, thậm chí là không thể, hoặc chi phí bỏ ra để xây dựng hệ thống khác sẽ ít hơn chi phí sửa chữa cho hệ thống cũ. Khi hệ thống càng lớn, mỗi lần cho ra phiên bản mới cũng sẽ rất khó khăn, có thể làm trễ tiến độ của người sử dụng. Ngoài ra, hệ thống lớn sẽ ảnh hưởng đến khả năng làm việc của các lập trình viên tham gia phát triển hệ thống.



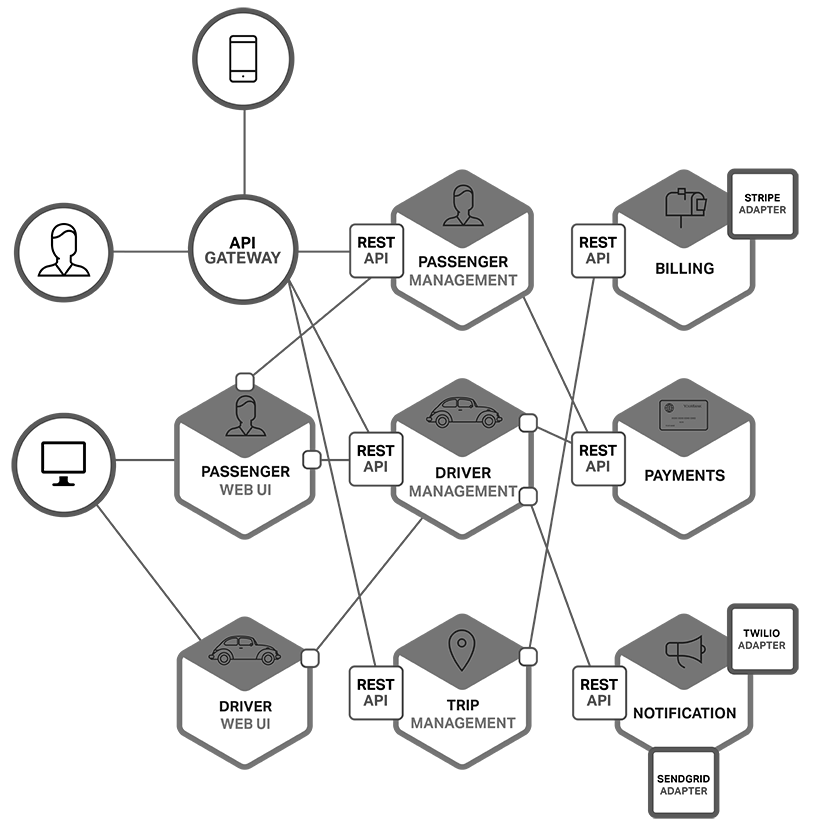
###### Hình 2.1. Minh họa mô hình kiến trúc nguyên khối

**Kiến trúc Microservices**

Để giải quyết những vấn đề của kiến trúc nguyên khối, ý tưởng được đề xuất là chia nhỏ hệ thống lớn ra thành những dịch vụ nhỏ kết nối với nhau. Mỗi dịch vụ nhỏ thực hiện một tập các chức năng chuyên biệt như quản lý đơn hàng, quản lý khách hàng,…Mỗi dịch vụ là một ứng dụng nhỏ kết nối thông qua bộ chuyển đổi (Adapter) khác nhau. Một số dịch vụ có nhiệm vụ làm API cho phép giao tiếp với các dịch vụ nhỏ khác hay các ứng dụng khác gọi tới. Khi vận hành, mỗi dịch vụ nhỏ được chạy trong một máy ảo (Virtual Machine) hoặc Docker Container (ảo hóa tầng ứng dụng).

Như vậy, mỗi vùng chức năng giờ đã được thực thi bởi một dịch vụ nhỏ, ứng dụng web cũng có thể chia nhỏ hơn cho từng nhóm đối tượng người dùng (quản lý, khách hàng,..). Thiết kế giao diện cho từng nhóm đối tượng người dùng giúp tối ưu hóa trải nghiệm, tốc độ nhanh hơn, dễ tương thích hơn trong khi cấu trúc chức năng đơn giản hơn.

Kiến trúc microservices chú trọng nhiều đến quan hệ giữa ứng dụng và CSDL. Thay vì dùng chung một cơ sở dữ liệu giữa các dịch vụ, mỗi dịch vụ sẽ có CSDL riêng. Cách này khác so với cơ sở dữ liệu tập trung (Toàn bộ ứng dụng sử dụng chung một CSDL thống nhất). Hệ quả là sẽ có sự dư thừa dữ liệu, cơ chế ràng buộc khóa ngoại cho quan hệ dữ liệu không thể áp dụng với bảng ở 2 cơ sở dữ liệu tách biệt.



###### Hình 2.2. Minh họa mô hình Microservices

**Kết luận phân biệt giữa hai kiến trúc**

Kiến trúc một khối sẽ hữu hiệu đối với ứng dụng đơn giản, ít chức năng. Nó bộc lộ nhiều nhược điểm khi ứng dụng phát triển lớn nhiều chức năng.

Kiến trúc Microservices chia nhỏ kiến trúc một khối ra các dịch vụ nhỏ. Microservices sẽ hiệu quả, phù hợp cho những ứng dụng phức tạp, liên tục phát triển nếu được thiết kế đúng và tận dụng các công nghệ quản lý, vận hành tự động.

**c. Ưu điểm của kiến trúc Microservices**

Đơn giản hóa hệ thống lớn: Vì hệ thống được phân chia ra thành nhiều dịch vụ nhỏ hơn giao tiếp với nhau nên hệ thống lớn sẽ giảm sự phức tạp của hệ thống đi rất nhiều.

Chia nhỏ ứng dụng một khối cồng kềnh thành các dịch vụ nhỏ dễ quản lý, bảo trì nâng cấp, tự do chọn, nâng cấp công nghệ mới: Hệ thống được chia ra thành nhiều dịch vụ nhỏ hơn nên khi xảy ra lỗi ở một chức năng nào đó trong hệ thống, chỉ cần tìm đến dịch vụ thực hiện chức năng đó để bảo trì sửa chữa, tương tự quá trình nâng cấp, phát triển hệ thống cũng sẽ đơn giản hơn rất nhiều.

Microservices thúc đẩy tách rạch ròi các khối chức năng : Điều rất khó thực hiện với ứng dụng một khối. Nếu muốn tạo sự rạch ròi trong ứng dụng một khối, sẽ phải thiết kế theo mô hình kiến trúc đặc biệt và phải liên tục tái cấu trúc hệ thống.

Mỗi dịch vụ nhỏ sẽ phát triển dễ hơn, nhanh hơn, dễ viết mã kiểm thử tự động cho từng dịch vụ nhỏ, điều mà kiến trúc nguyên khối có thể gặp khó khăn.

Một số dịch vụ nhỏ có thuê ngoài phát triển mà vẫn bảo mật hệ thống và mã nguồn phần dịch vụ còn lại. Đội phát triển có nhiều lựa chọn công nghệ mới, khung phần mềm (framework), CSDL mới, đa dạng để nâng cấp từng dịch vụ nhỏ, chọn môi trường tối ưu nhất để chạy. Các dịch vụ có thể bật tắt để kiểm nghiệm so sánh, tăng tốc quá trình cải tiến giao diện. Việc riển khai phần mềm đều đặn khả thi với Microservices. Dịch vụ nhỏ đóng gói có thể chuyển từ môi trường phát triển sang môi trường chạy thật không phải cấu hình thủ công lại, không phải copy tập tin quá lớn.

**d. Nhược điểm của Microservices**

Nhược điểm đầu tiên của Microservicess là Microservices nhấn mạnh kích thước nhỏ gọn của dịch vụ. Chia quá nhiều sẽ dẫn đến rời rạc, vụn vặt, khó kiểm soát. Việc lưu dữ liệu cục bộ bên trong những dịch vụ quá nhỏ sẽ khiến dữ liệu phân tán quá mức cần thiết.

Do hệ thống được chia thành những dịch vụ nhỏ nên phải xử lý sự cố khi kết nối chậm, lỗi khi thông điệp không gửi được hoặc thông điệp gửi đến nhiều đích đến vào các thời điểm khác nhau.

Đảm bảo giao dịch phân tán cập nhật dữ liệu đúng đắn vào nhiều dịch vụ nhỏ khác nhau rất khó, đôi khi là không thể so với đảm bảo giao dịch cập nhật vào nhiều bảng trong một cơ sở dữ liệu trung tâm.

Kiểm thử tự động một dịch vụ trong kiến trúc Microservices đôi khi yêu cầu phải chạy cả các dịch vụ nhỏ khác mà nó phụ thuộc. Do đó khi phân rã ứng dụng một khối thành Microservices cần luôn kiểm tra mức độ ràng buộc giữa các dịch vụ một cách cẩn thận.

Nếu các dịch vụ nhỏ thiết kế phục thuộc vào nhau theo chuỗi công việc A gọi B, B gọi C, C gọi D. Nếu một mắt xích có giao tiếp API thay đổi, các mắc xích khác có thể sẽ bị ảnh hưởng. Vì vậy, việc thiết kế dịch vụ tốt sẽ giảm tối đa ảnh hưởng lan truyền đến các dịch vụ khác, và cần phải có quy tắc phân luồng để trách tình trạng tắc nghẽn ở một dịch vụ nào đó.

**❖ Giới thiệu về ReactJS, Redux, Typescript, JasperReports**

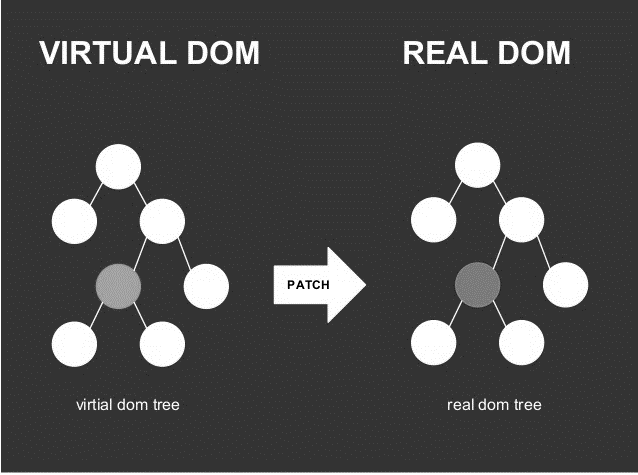
**a. Giới thiệu về ReactJS**

ReactJS là một thư viện giao diện người dùng, được Facebook phát triển để hỗ trợ việc xây dựng những thành phần giao diện người dùng có tính tương tác cao, có trạng thái và có thể sử dụng lại được. ReactJS được sử dụng tại Facebook trong sản xuất, và cả Instagram cũng được viết hoàn toàn trên ReactJS.

Một trong những điểm hấp dẫn của ReactJS là thư viện này không chỉ hoạt động trên phía người dùng, mà còn được kết xuất (render) trên máy chủ và có thể kết nối với nhau. ReactJS so sánh sự thay đổi giữa các giá trị của lần kết xuất này với lần kết xuất trước và cập nhật ít thay đổi nhất trên DOM (Document Object Model – Mô hình đối tượng tài liệu).

Mô hình đối tượng tài liệu ảo (Virtual DOM – DOM ảo):

Công nghệ DOM ảo giúp tăng hiệu năng cho ứng dụng. Việc chỉ nút gốc mới có trạng thái và khi nó thay đổi sẽ tái cấu trúc lại toàn bộ, đồng nghĩa với việc cây DOM cũng sẽ phải thay đổi một phần, điều này sẽ ảnh hưởng đến tốc độ xử lý. ReactJS sử dụng Virtual DOM để cải thiện vấn đề này.Virtual DOM là một đối tượng trong Javascript, mỗi đối tượng chứa đầy đủ thông tin cần thiết để tạo ra một DOM, khi dữ liệu thay đổi nó sẽ tính toán sự thay đổi giữa đối tượng và cấu trúc cây thật, điều này sẽ giúp tối ưu hoá việc kết xuất lại cây DOM thật.



###### Hình 2.3. Mô hình Virtual DOM

ReactJS sử dụng cơ chế luồng dữ liệu một chiều. Dữ liệu được truyền từ thành phần cha đến thành phần con thông qua ‘props’. Luồng dữ liệu đơn giản giúp chúng ta dễ dàng kiểm soát cũng như sửa lỗi. Với các đặc điểm ở trên, ReactJS dùng để xây dựng các ứng dụng lớn mà dữ liệu của chúng thay đổi liên tục theo thời gian. Dữ liệu thay đổi thì hầu hết kèm theo sự thay đổi về giao diện. Ví dụ như Facebook: trên mục bảng tin của người dùng cùng lúc sẽ có các bài đăng khác nhau và mỗi bài đăng lại có số lượt thích, chia sẻ, bình luận liên tục thay đổi. Khi đó ReactJS sẽ rất hữu ích để sử dụng.

**Giới thiệu về JSX**

JSX là một dạng ngôn ngữ cho phép viết các mã HTML trong Javascript. JSX thực hiện tối ưu hóa trong khi biên dịch sang mã Javacsript. Các mã này cho thời gian thực hiện nhanh hơn nhiều so với một mã tương đương viết trực tiếp bằng Javascript.

Ngược với Javascript, JSX là kiểu được biên dịch trước khi chạy, giống như Java, C++. Vì thế các lỗi sẽ được phát hiện ngay trong quá trình biên dịch. Ngoài ra, nó cũng cung cấp tính năng gỡ lỗi khi biên dịch rất tốt.

JSX kế thừa dựa trên Javascript, vì vậy rất dễ dàng để cho các lập trình viên Javascripts có thể sử dụng.

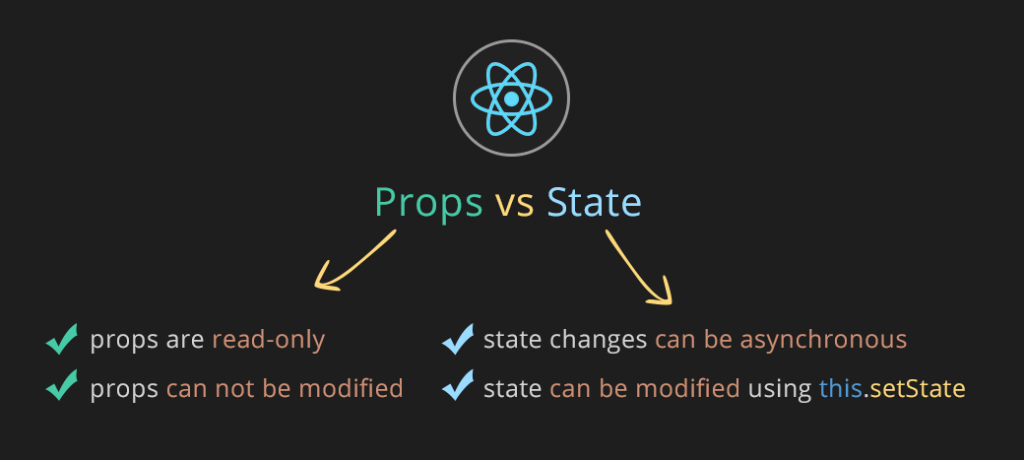
**Giới thiệu về Components (thành phần)**

ReactJS được xây dựng xung quanh các thành phần, gọi là Component, chứ không dùng khung mẫu như các framework khác. Trong ReactJS, chúng ta xây dựng trang web sử dụng những Component nhỏ. Chúng ta có thể tái sử dụng một Component ở nhiều nơi, với các trạng thái hoặc các thuộc tính khác nhau, trong một Component lại có thể chứa Component khác. Mỗi Component trong React có một trạng thái riêng, có thể thay đổi, và ReactJS sẽ thực hiện cập nhật Component dựa trên những thay đổi của trạng thái. Mọi thứ trong React đều là Component. Chúng giúp bảo trì mã lệnh khi làm việc với các dự án lớn. Một React Component đơn giản chỉ cần một phương thức kết xuất, gọi là Render Method. Có rất nhiều phương thức khả dụng khác, nhưng Render Method là phương thức chủ yếu.

**Giới thiệu về Props và States**

Props: Giúp các Component tương tác với nhau, Component nhận giá trị trả về gọi là props, và trả thuộc tính mô tả những gì Component con sẽ kết xuất. Props là phân tử không thay đổi.

States: Thể hiện trạng thái của ứng dụng, khi state thay đồi thì Component đồng thời kết xuất lại để cập nhật giao diện người dùng. States có thể thay đổi bằng cách sử dụng this.setState();



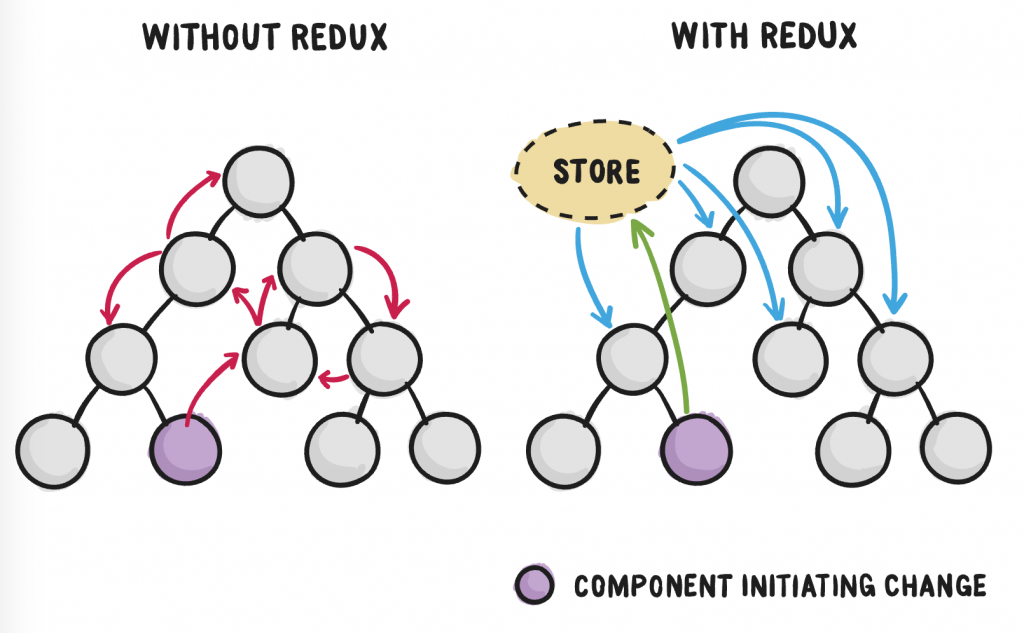
###### Hình 2.4. Giới thiệu về Props và State

**b. Giới thiệu về Redux**

Redux là một công cụ quản lý cho các ứng dụng Javascript, có thể gọi Redux là một kho chứa. Nó giúp bạn viết các ứng dụng hoạt động một cách nhất quán, chạy trong các môi trường khác nhau (client, server, and native).

Mặc dù chủ yếu được sử dụng với React, nhưng nó có thể được sử dụng với bất kỳ thư viện hoặc Framework JavaScript nào khác.

Với Redux, States của ứng dụng được giữ trong một nơi gọi là Store và mỗi Component đều có thể truy cập bất kỳ States nào mà chúng muốn từ Store này.



###### Hình 2.5. Mô phỏng Redux cho ứng dụng

**Cách hoạt động của Redux:**

Redux có một Store lưu trữ toàn bộ State của của. Mỗi Component có thể truy cập trực tiếp đến State được lưu trữ thay vì phải gửi Props từ Component này đến Component khác. Có 3 thành phần của Redux: Actions, Store, Reducers.

**Action:**

Actions đơn giản là các sự kiện. Chúng là cách mà chúng ta gửi dữ liệu từ ứng dụng đến Store. Những dữ liệu này có thể là từ sự tương tác của người dùng với ứng dụng, gọi từ API hoặc cũng có thể là từ gửi dữ liệu của bảng.

Actions được gửi bằng cách sử dụng phương thức Store.dispatch(), chúng phải có một kiểu để biểu lộ loại sự kiện để thực hiện. Chúng cũng phải có một khối dữ liệu, gọi là payload chứa thông tin.

**Reducers:**

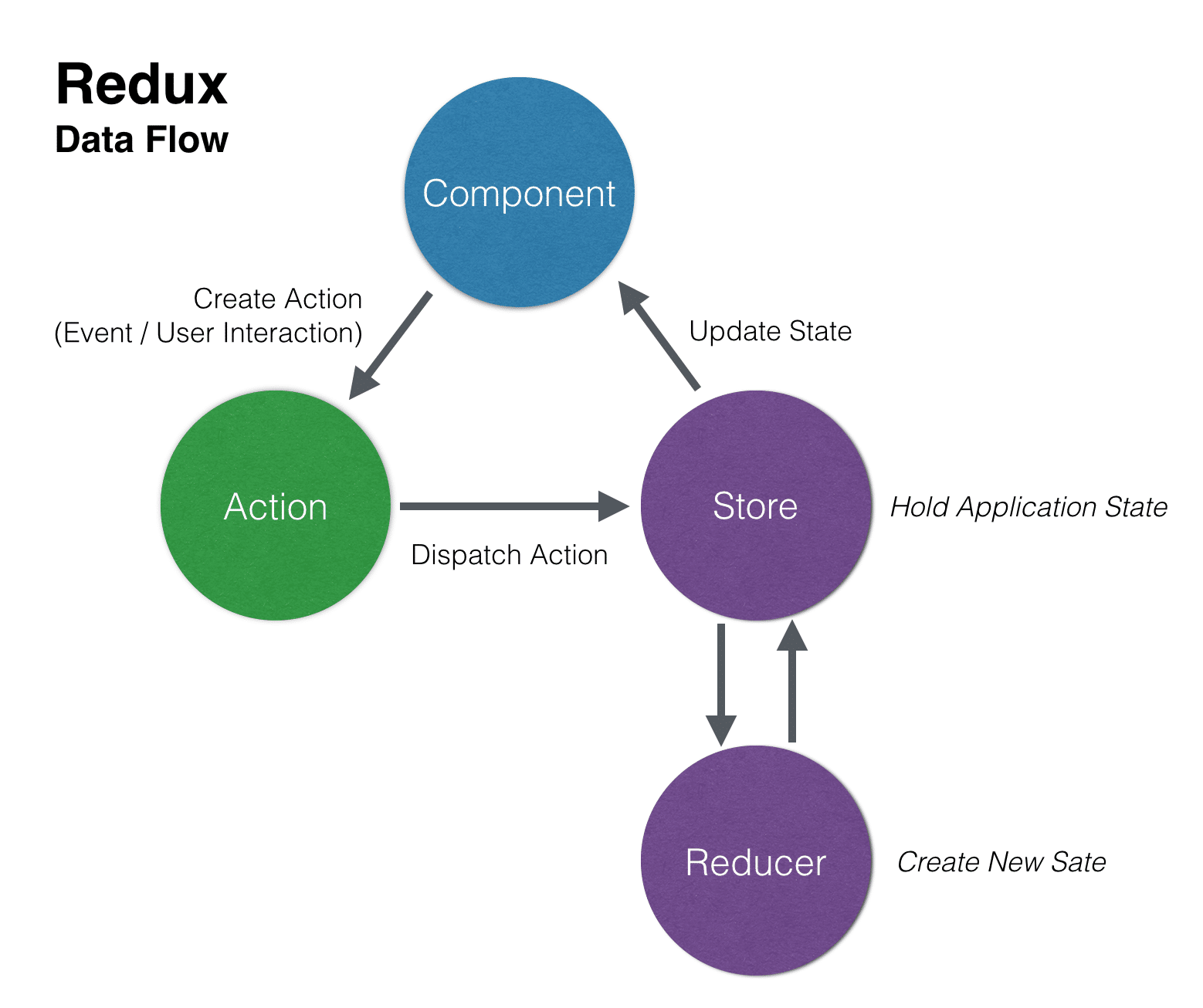
Reducers là các phương thức nguyên thủy (Pure Funtioncs) chúng lấy State hiện tại của ứng dụng, thực hiện một Action và trả về một state mới. Những State này được lưu như những đối tượng và chúng định rõ cách State của một ứng dụng thay đổi trong việc phản hồi một Action được gửi đến Store.

Phương thức nguyên thủy: Một hàm thuần tuý (pure) chỉ là một hàm bình thường với hai ràng buộc bổ sung mà nó buộc thỏa mãn:

* Với một tập hợp các dữ liệu đầu vào, hàm luôn luôn trả về cùng một kết quả.
* Hàm này không sinh ra sự kiện phụ.

**Store:**

Store lưu trạng thái ứng dụng và nó là duy nhất trong bất kỳ một ứng dụng Redux nào. Người lập trình có thể truy các State được lưu, cập nhật State.



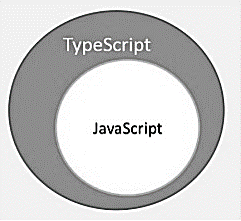
###### Hình 2.6. Mô hình hoạt động của Redux

**c. Giới thiệu về TypeScript**

TypeScript là một dự án mã nguồn mở được phát triển bởi Microsoft, nó có thể được coi là một phiên bản nâng cao của Javascript bởi việc bổ sung tùy chọn kiểu tĩnh và lớp hướng đối tượng mà điều này không có ở Javascript. TypeScript có thể sử dụng để phát triển các ứng dụng chạy ở Client-side (Angular2) và Server-side (NodeJS). TypeScript sử dụng tất cả các tính năng của của ECMAScript 2015 (viết tắt là ES6) như classes, modules.

Lí do nên sử dụng Typescript:

* Dễ phát triển dự án lớn: Với việc sử dụng các kỹ thuật mới nhất và lập trình hướng đối tượng nên TypeScript giúp chúng ta phát triển các dự án lớn một cách dễ dàng.
* Nhiều Framework lựa chọn: Hiện nay các Javascript Framework đã dần khuyến khích nên sử dụng TypeScript để phát triển.
* Hỗ trợ các tính năng của Javascript phiên bản mới nhất: TypeScript luôn đảm bảo việc sử dụng đầy đủ các kỹ thuật mới nhất của Javascript, ví dụ như version ECMAScript 2015 (ES6).
* Là mã nguồn mở: TypeScript là một mã nguồn mở nên người dùng hoàn toàn có thể sử dụng mà không mất phí, bên cạnh đó còn được cộng đồng hỗ trợ.
* TypeScript là Javscript: Bản chất của TypeScript là biên dịch tạo ra các đoạn mã Javascript nên người dùng có thê chạy bất kì ở đâu miễn ở đó có hỗ trợ biên dịch Javascript. Ngoài ra bạn có thể sử dụng trộn lẫn cú pháp của Javascript vào bên trong TypeScript, điều này giúp các lập trình viên tiếp cận TypeScript dễ dàng hơn.



Hình 2.7. Minh họa TypeScript

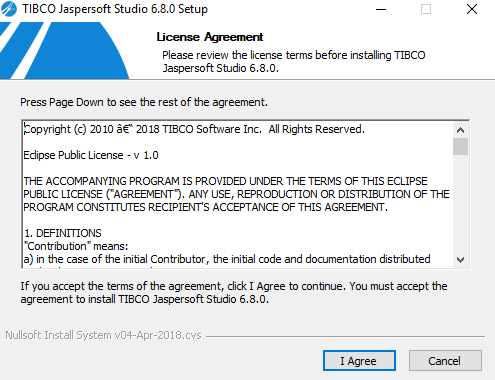
**❖ Giới thiệu về JasperReports**

JasperReports là một công cụ tạo báo cáo mã nguồn mở phổ biế nhất hiện tại. Nó được viết hoàn toàn bằng Java và được cho phép sử dụng với nhiều loại nguồn dữ liệu và tạo ra những báo cáo cho phép hiển thị, in ấn hoặc xuất ra những định dạng như HTML, PDF, Excel, Word,…

Hầu hết các ứng dụng trong thực tế cần phải xuất các báo cáo theo nghiệp vụ cũng như theo yêu cầu của khách hàng. Vì vậy, JasperReports sẽ là một công cụ tạo báo cáo đơn giản, mạnh mẽ sẽ giúp tiết kiệm được nhiều công sức, thời gian. Các báo cáo được thiết kế bằng một phần mềm cung cấp riêng để tạo báo cáo có tên JasperSoft Studio.

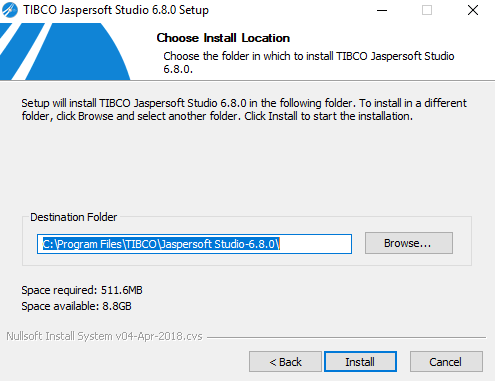
**a. Cài đặt JasperSoft Studio**

Bước 1: Chọn I Agree.



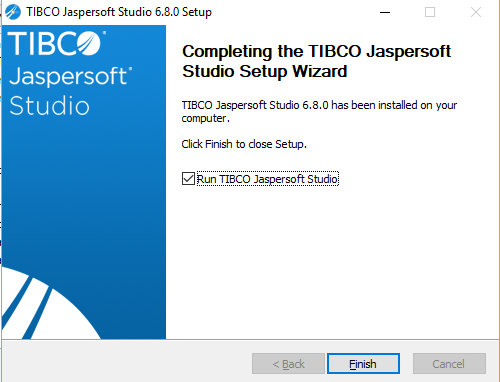
###### Hình 2.8. Các bước cài đặt JasperSoft Studio

Bước 2: Chọn thư mục để cài phần mềm 🡪 Nhấn Install.



###### Hình 2.9. Các bước cài đặt JasperSoft Studio

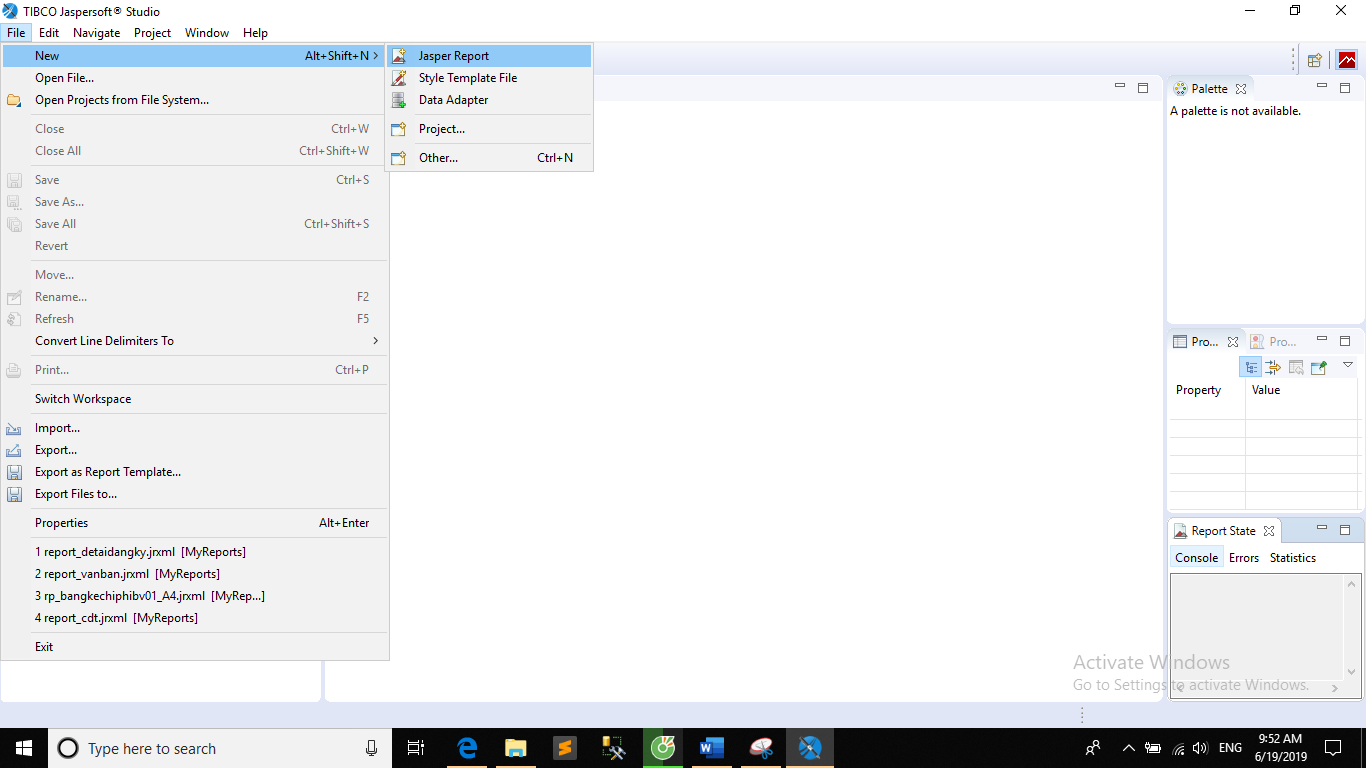
Bước 3: Sau khi phần mềm cài đặt xong 🡪 Chọn Finish.



###### Hình 2.10. Các bước cài đặt JasperSoft Studio

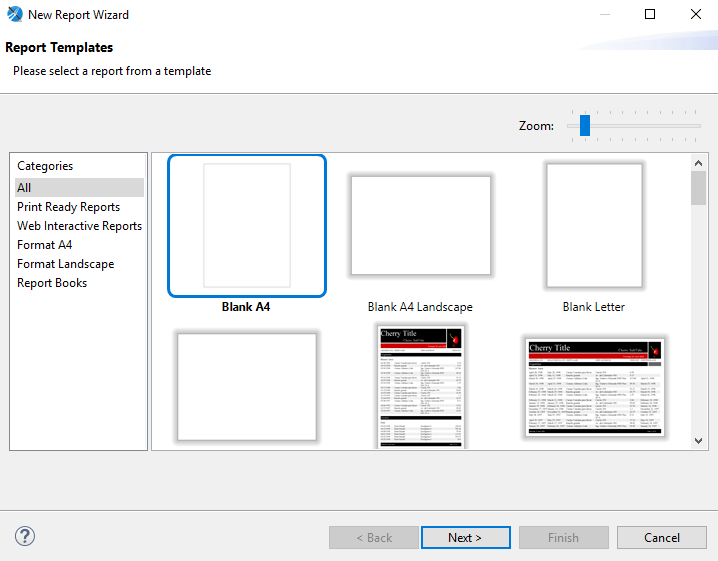
**b. Tạo báo cáo bằng JasperSoft Studio**

Bước 1: Chọn File 🡪 New 🡪 Jasper Report.



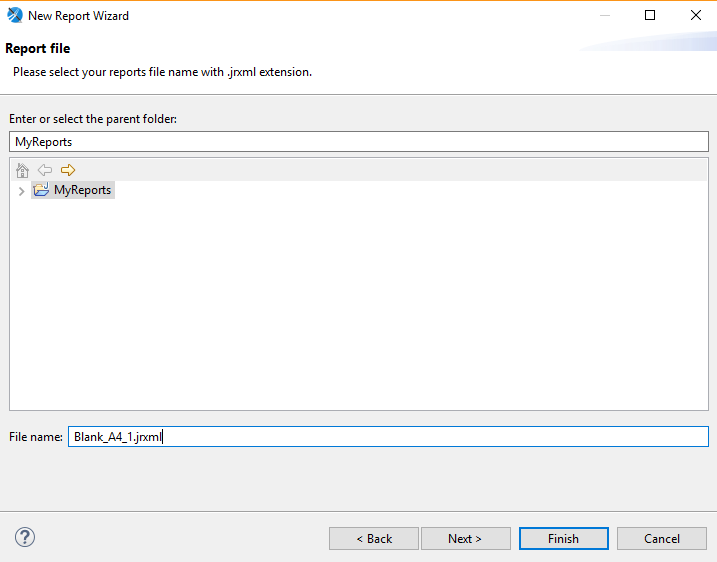
###### Hình 2.11. Các bước tạo báo cáo bằng JasperSoft Studio

Bước 2: Chọn vào loại khung cho Report. Jasper hỗ trợ rất nhiều loại, ở đây chọn loại khổ A4 giấy đứng 🡪 Nhấn Next.



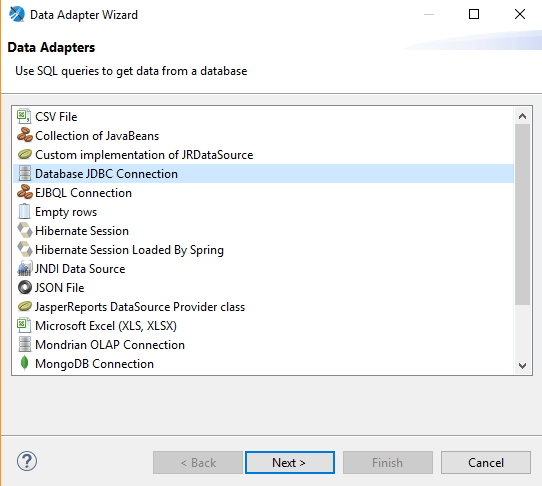
###### Hình 2.12. Các bước tạo báo cáo bằng JasperSoft Studio

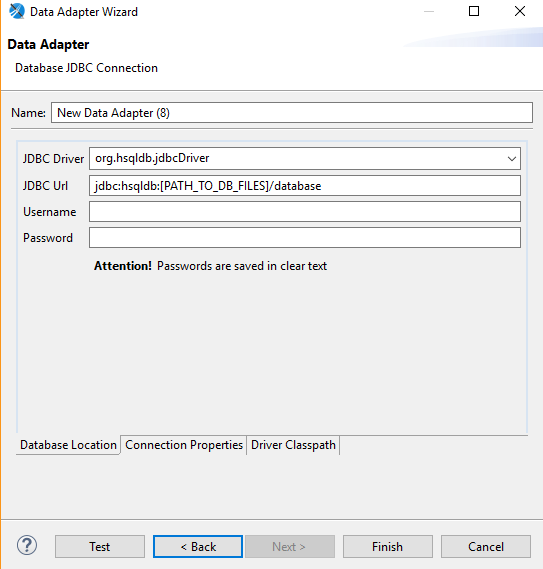
Bước 3: Chọn thư mục lưu trữ và đặt tên cho báo cáo 🡪 Nhấn Next.



###### Hình 2.13. Các bước tạo báo cáo bằng JasperSoft Studio

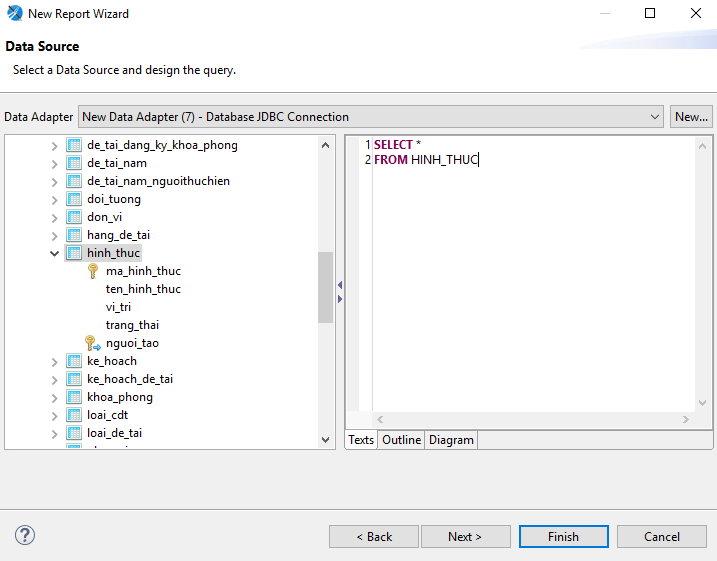
Bước 4: Chọn kết nối tới CSDL bằng cách: Chọn New 🡪 Chọn loại kết nối CSDL. Jasper hỗ trợ rất nhiều kết nối đến CSDL, ở đây chọn Database JDBC Connection 🡪 Nhấn Next.

Hình 2.14. Các bước tạo báo cáo bằng JasperSoft Studio

Bước 5: Nhập các thông tin kết nối đến CSDL, nhấn Test để kiểm tra kết nối CSDL 🡪 Nhấn Finish.

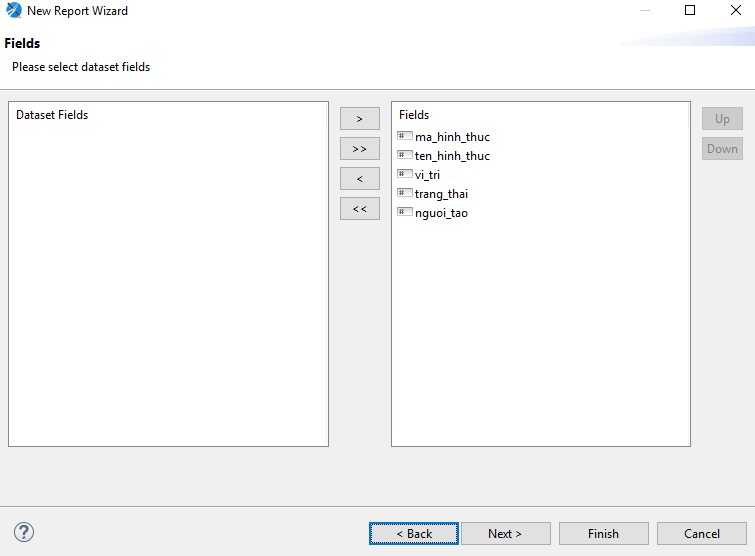
###### Hình 2.15. Các bước tạo báo cáo bằng JasperSoft Studio

Bước 6: Nhập câu lệnh truy vấn 🡪 Nhấn Next.



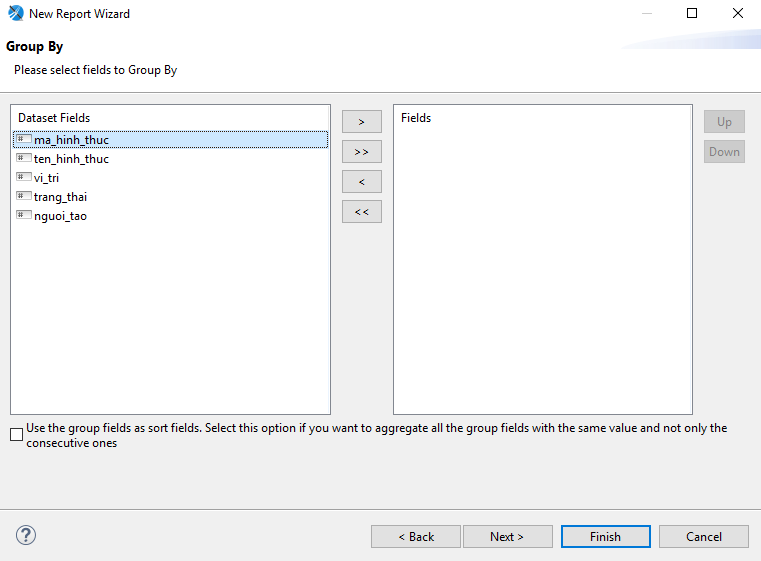
###### Hình 2.16. Các bước tạo báo cáo bằng JasperSoft Studio

Bước 7: Chọn các trường dữ liệu sẽ tạo báo cáo 🡪 Nhấn > 🡪 Nhấn Next.



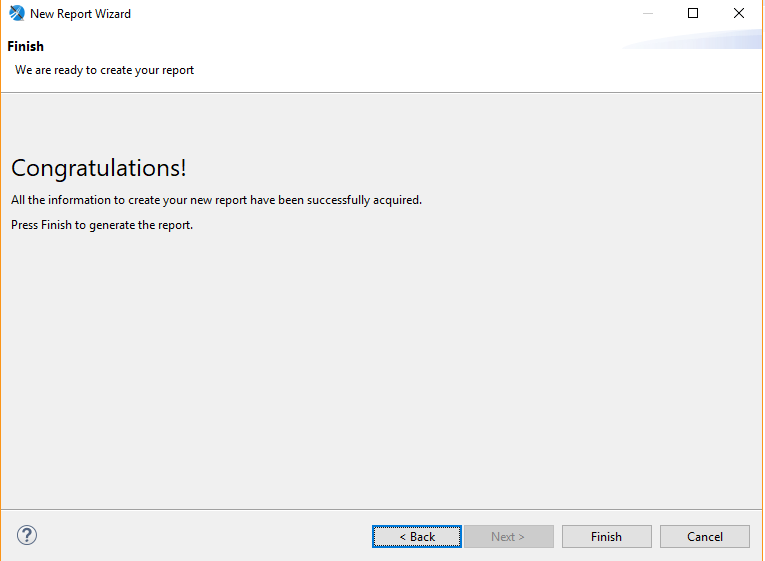
###### Hình 2.17. Các bước tạo báo cáo bằng JasperSoft Studio

Bước 8: Chọn phân nhóm cho báo cáo, nếu không có phân nhóm thì bỏ qua 🡪 Nhấn Next.



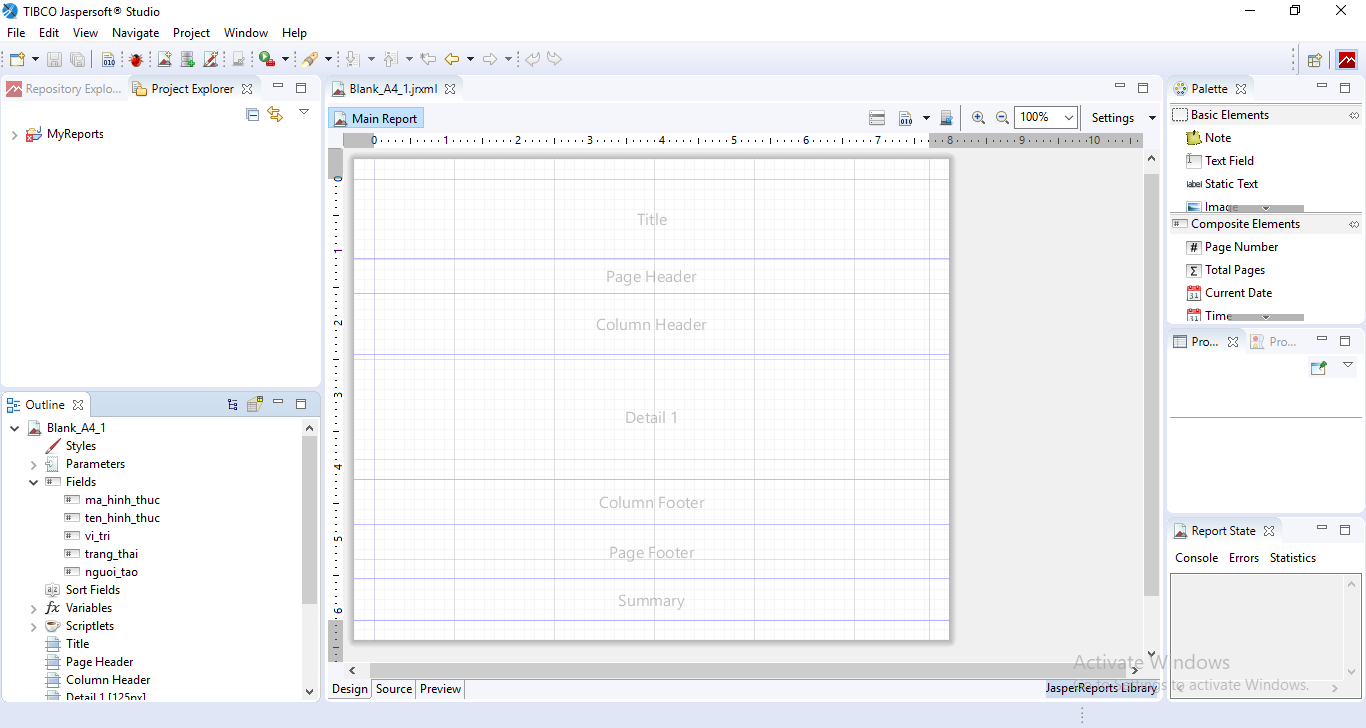
###### Hình 2.18. Các bước tạo báo cáo bằng JasperSoft Studio

Bước 9: Nhấn Finish để hoàn tất tạo báo cáo



###### Hình 2.19. Các bước tạo báo cáo bằng JasperSoft Studio

Bước 10: Sau khi xong các bước trên, giao diện sẽ như bên dưới, bắt đầu tiến hành thiết kế Report bằng cách chọn các điều khiển ở khung Palette



###### Hình 2.20. Các bước tạo báo cáo bằng JasperSoft Studio

Các điều khiển cơ bản:

Static Text: Văn bản thuần túy, không thể sử dụng mã lệnh trong văn bản.

Text Field: Trường văn bản, chứa văn bản có thể sử dụng mã lệnh đối với văn bản.

Image: Hình ảnh, dùng để hiển thị hình ảnh trong báo cáo.

Note: Ghi chú thích của người thiết kế.

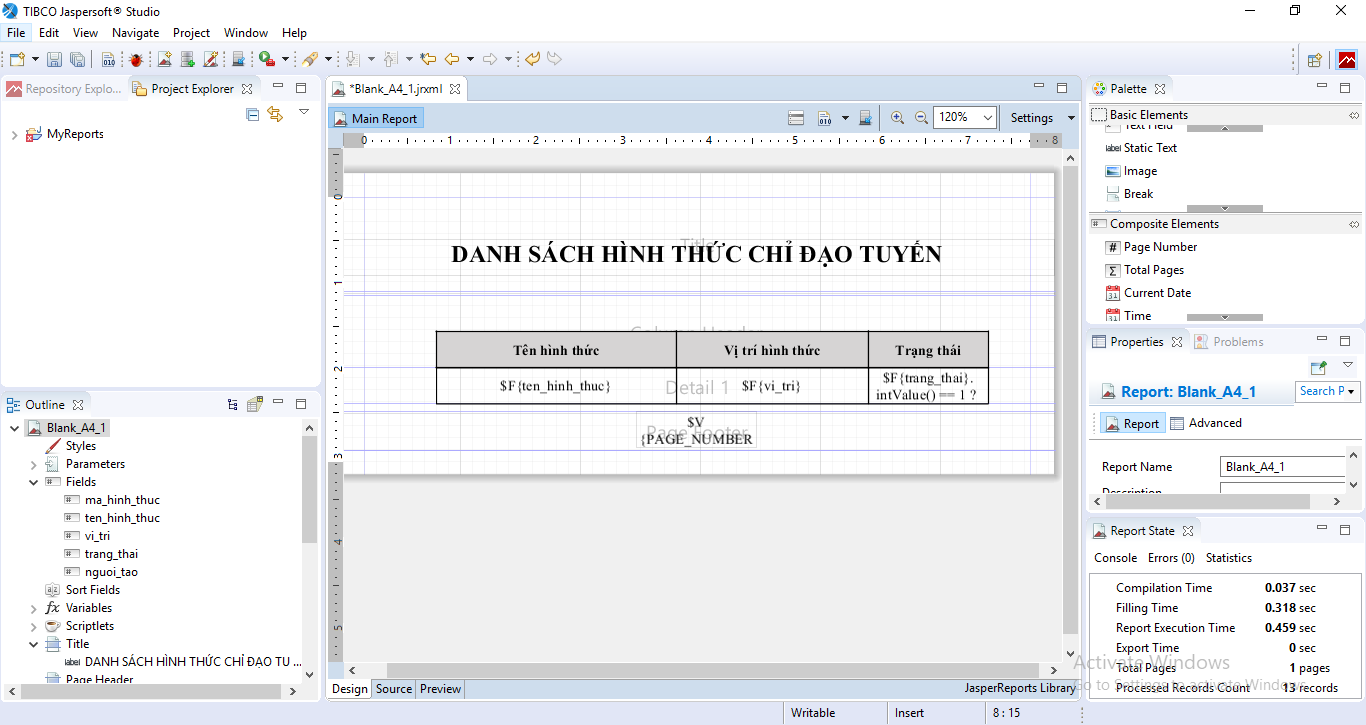
Page Number: Là một Text Field có sẵn mã lệnh hiển thị trang hiện tại.

Total Page: Là một Text Field có sẵn mã lệnh hiển thị tổng số trang của báo cáo.

Current Date: Là một Text Field có sẵn mã lệnh hiển thị thời gian hiện tại.

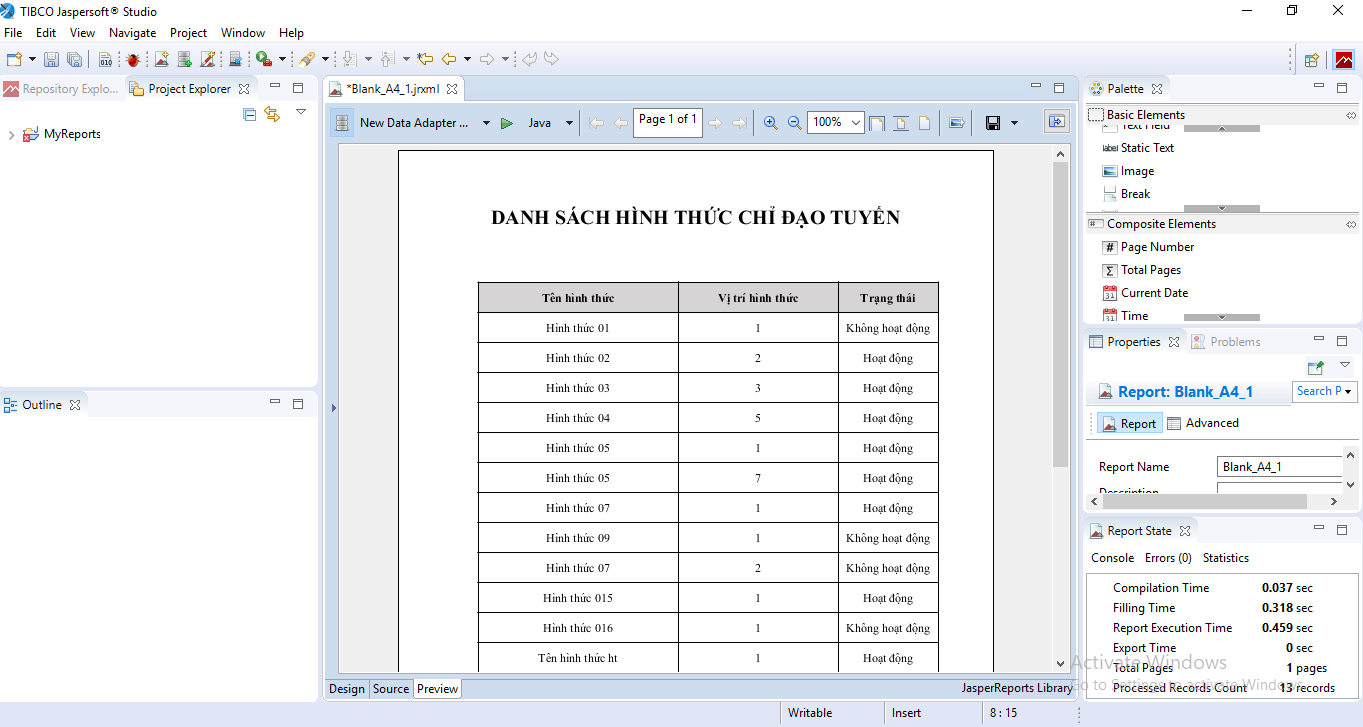
Ngoài những điều khiển cơ bản để thiết kế báo cáo, Jasper Reports cũng cho phép người dùng tạo ra các biến giá trị(Variables), các tham số truyền vào(Params), Fields(Các trường dữ liệu, mặc định sẽ là những trường dữ liệu mà người dùng đã lấy từ câu truy vấn ở bước 6.

Bước 11: Sau khi thiết kế xong, giao diện sẽ như bên dưới 🡪 Nhấn Preview để xem kết quả.



###### Hình 2.21. Các bước tạo báo cáo bằng JasperSoft Studio

Kết quả:

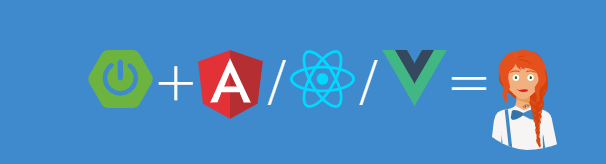


###### Hình 2.22. Các bước tạo báo cáo bằng JasperSoft Studio

**❖ Sử dụng JHipster để tạo nhanh mô hình Microservice**

**a. Giới thiệu JHipster**

JHipster, viết tắt của Java Hipster là một framework được phát triển bởi Yeoman và được sử dụng để phát triển các dự án Spring Boot + ReactJS hoặc AngularJS.



###### Hình 2.23. Minh họa JHipster

**b. Yêu cầu hệ thống**

Cài đặt đầy đủ thư viện cho JHipster, bao gồm:

Cài đặt và cấu hình Java 8 trên máy tính

Cài đặt NodeJS từ trang chủ

Cài Yeoman bằng cách vào command prompt và nhập npm install -g yo

**c. Khởi tạo dự án**

Tất cả câu lệnh đều nhập thủ công trên command prompt của máy tính. Để tạo dự án JHipter, thực hiện :

* Tạo một thư mục trống là nơi sẽ chứa dự án bằng lệnh: mkdir myapplication
* Chuyển đến thư mục vừa tạo bằng lệnh: cd myapplication
* Tạo tự động ứng dụng JHipster bằng lệnh: jhipster
* Sau đó, lựa chọn những thứ phù hợp với dự án của người dùng.

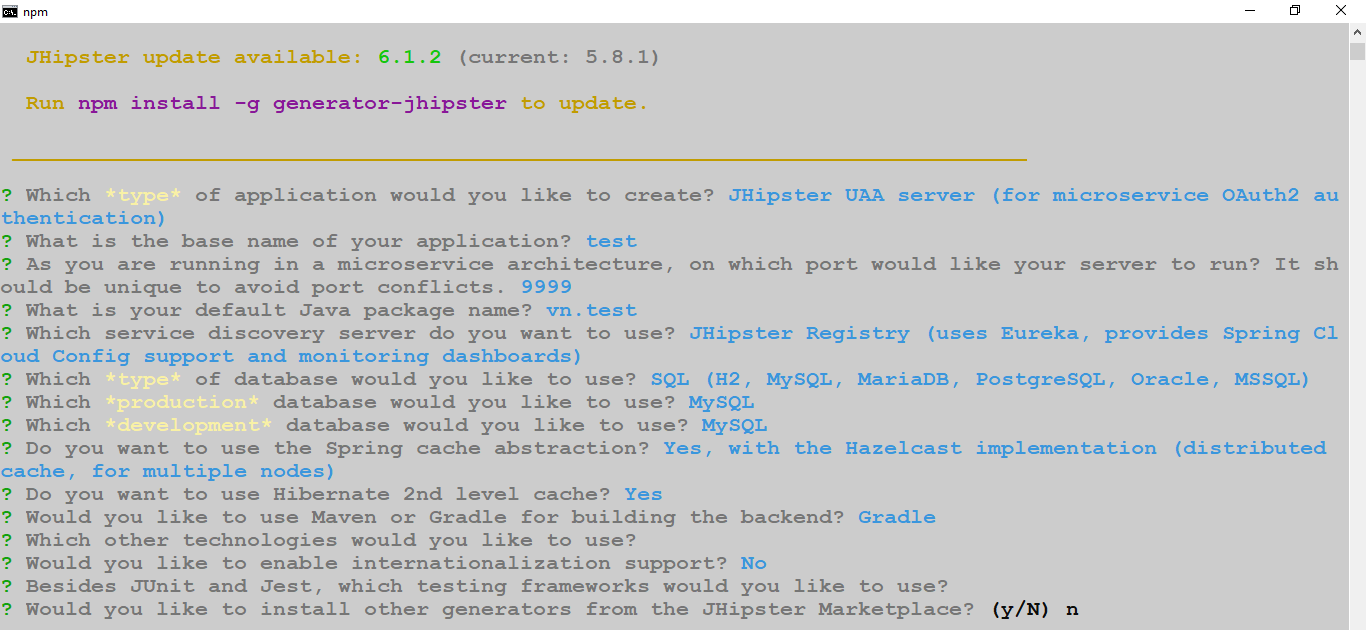
Bây giờ người dùng đã có một dự án với:

* Backend: Spring Boot + Spring Security
* Database:
  + Mysql (production)
  + H2 with disk-based (development)
* Frontend: Bootstrap + SASS + React (hoặc thấp hơn tùy vào phiên bản JHipster người dùng cài đặt ở trên).

Sau khi chạy xong người dùng sẽ nhận được kết quả như bên dưới. Đó cũng là hướng dẫn để người dùng xây dựng dự án của mình trên local.

Chạy ứng dụng Spring Boot bằng cách nhập: ./mvnw

Sau đó dùng lệnh để chạy: npmstart



###### Hình 2.24. Minh họa tạo dự án JHipster

## **2.2. Cơ sở thực tiễn và ứng dụng của khóa luận**

### **2.2.1. Bối cảnh thực tiễn**

**❖ Bối cảnh thực tiễn**

Internet được phát triển rộng khắp trên thế giới, và ngày 19 tháng 11 năm 1997 là ngày đầu Việt Nam được hòa vào mạng Internet toàn cầu. Internet xuất hiện cũng đã làm xuất hiện nhu cầu về lập trình web, các nền tảng lập trình web ban đầu cũng được hình thành, như HTML, CSS,... Qua nhiều năm phát triển không ngừng, lập trình web đã có nhiều bước tiến đáng kể và được sử dụng ngày càng nhiều và yêu cầu đặt ra cũng tăng cao, các công nghệ lập trình web cũ dần trở nên lỗi thời hoặc không còn đảm bảo nữa. Việc sử dụng công nghệ cũ trong lập trình web tiềm ẩn nhiều rủi ro, tính bảo mật chưa cao, nên việc sử dụng công nghệ mới trong lập trình web là rất cần thiết. Chính vì vậy, việc sử dụng công nghệ mới, trong đó có Microservice sẽ mang lại rất nhiều lợi ích.

**❖ Nhu cầu thực tế**

Cần phải có một hệ chỉ đạo tuyến cho ngành để kịp thời chuyển tuyến các chỉ đạo tuyến từ các bệnh viện tuyến trên xuống tuyến dưới một cách an toàn, bảo mật.

Cần phải có một hệ thống quản lý việc đăng ký, xét duyệt đề tài NCKH trong các đơn vị ngành y tế.

### **2.2.2. Ứng dụng**

Khóa luận này là một tài liệu tham khảo có ích cho các sinh viên khi tìm hiểu về mô hình Microservices.

Sản phẩm phần mềm của khóa luận này có thể ứng dụng vào sử dụng ở một đơn vị ngành y tế.

# Chương 3: PHÂN TÍCH, THIẾT KẾ, CÀI ĐẶT GIẢI PHÁP

## **3.1. Phân tích yêu cầu**

### **3.1.1. Phân tích yêu cầu chức năng**

Đầy đủ các thuộc tính của một chỉ đạo tuyến và một đề tài NCKH

Có phân quyền cho nhiều nhóm người sử dụng:

Phân hệ chỉ đạo tuyến:

* Cán bộ phụ trách chỉ đạo tuyến:
* Quản lý thông tin về hình thức chỉ đạo tuyến, loại chỉ đạo tuyến, nội dung chỉ đạo tuyến, đối tượng tiếp nhận chỉ đạo tuyến, đánh giá chỉ đạo tuyến, văn bản chỉ đạo tuyến, kế hoạch chỉ đạo tuyến và chỉ đạo tuyến.
* Thống kê các văn bản chỉ đạo tuyến, kế hoạch chỉ đạo tuyến, và chỉ đạo tuyến hàng năm.
* Nhân viên của đơn vị tuyến dưới:
  + Xem thông tin chỉ đạo tuyến tiếp nhận của đơn vị.
* Phân hệ NCKH:
  + Nhân viên quản lý NCKH
    - Quản lý loại đề tài, hạng đề tài, chức vụ hội đồng, chủ thể đề tài, nhóm đề tài, kế hoạch đề tài.
* Nhân viên phòng NCKH của đơn vị
  + Quản lý đề tài đăng ký tại đơn vị.
  + Xét duyệt tên đề tài đã đăng ký để trình lên hội đồng.
  + Thống kê đề tài đăng ký hàng năm.
* Cán bộ hội đồng NCKH
  + Xét duyệt đề cương, xét duyệt báo cáo tổng kết, xét duyệt công nhận đề tài năm, xét duyệt đề tài ứng dụng tại đơn vị.
  + Thống kê đề tài đăng ký, thống kê đề tài công nhận đề tài năm, đề tài ứng dụng tại đơn vị hàng năm.
* Cán bộ đơn vị NCKH
  + Cập nhật biên bản ứng dụng, cập nhật báo cáo quý năm, cập nhật văn bản áp dụng đề tài tại đơn vị.
  + . Thống kê đề tài đăng ký, thống kê đề tài công nhận đề tài năm, đề tài ứng dụng tại đơn vị hàng năm.

### **3.1.2. Phân tích yêu cầu phi chức năng**

Giao diện: trực quan, đơn giản, dễ sử dụng, phù hợp với người không rành về Công nghệ thông tin.

Bảo mật: Không bị khai thác lỗi.

Có thể hoạt động trên các trình duyệt cơ bản trên máy tính.

Chạy nhanh, ổn định.

**3.1.3. Phân tích quy trình**

**❖ Phân tích quy trình chỉ đạo tuyến**

Quy trình chỉ đạo tuyến được mô tả thông qua bảng bên dưới.

|  |  |  |
| --- | --- | --- |
| **Trách nhiệm** | **Nội dung quy trình** | **Mô tả, biểu mẫu** |
| Trưởng bộ phận thực hiện chỉ đạo tuyến |  | Theo chỉ đạo của giám đốc bệnh viện, các trưởng khoa, trưởng phòng và trưởng phòng kế hoạch tổng hợp xây dựng kế hoạch chỉ đạo tuyến. |
| Bộ phận thực hiện chỉ đạo tuyến | Xây dựng chỉ đạo tuyến | Các cán bộ được phân công xây dựng chỉ đạo tuyến, bao gồm: hình thức, loại, nội dung, đối tượng tiếp nhận, đánh giá, văn bản chỉ đạo tuyến. |
| Bộ phận thực hiện chỉ đạo tuyến | Chuyển tuyến chỉ đạo tuyến xuống tuyến dưới | Các cán bộ được phân công thực hiện chuyển tuyến chỉ đạo tuyến xuống các tuyến dưới. |
| Cán bộ bệnh viện tiếp nhận được phân công | Tiếp nhận chỉ đạo tuyến từ tuyến trên | Các cán bộ được phân công thực hiện nhận chỉ đạo tuyến từ tuyến trên và thực hiện theo nội dung chỉ đạo tuyến. |
| Cán bộ bệnh viện tiếp nhận được phân công |  | Các cán bộ được phân công lưu trữ hồ sơ theo từng năm |

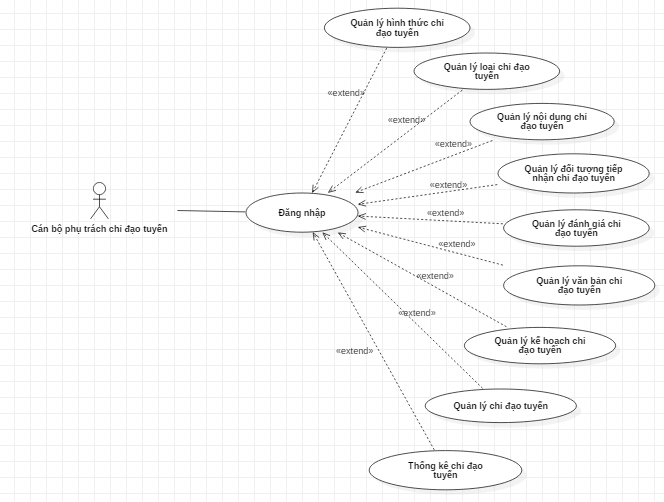
**❖ Phân tích quy trình đăng ký, xét duyệt đề tài NCKH**

|  |  |  |
| --- | --- | --- |
| **Trách nhiệm** | **Nội dung quy trình** | **Mô tả, biểu mẫu** |
| Hội đồng NCKH |  | - Gửi công văn cho các đơn vị về nội dung kế hoạch đăng ký NCKH, thông báo thời hạn đăng ký;  - Cung cấp các biểu mẫu trên trang web của bệnh viện. |
| Thư ký phòng NCKH các đơn vị | Đăng ký và kiểm tra thủ tục đăng ký | - Các đơn vị đăng ký đề tài qua phần mềm quản lý NCKH tại đơn vị;  - Cán bộ kiểm tra thông tin đăng ký đề tài của các đơn vị. |
| Hội đồng NCKH | Xét duyệt đề cương đề tài | - Cán bộ xét duyệt đề cương của đề tài đã đăng ký của các đơn vị.  - Nếu xét duyệt đề cương là không duyệt thì đề tài sẽ không được xét báo cáo tổng kết. |
| Hội đồng NCKH | Xét duyệt báo cáo tổng kết | - Cán bộ xét duyệt báo cáo tổng kết của đề tài đã đăng ký của các đơn vị.  - Nếu xét duyệt báo cáo tổng kết là không duyệt thì đề tài sẽ không được xét tiếp duyệt để công nhận đề tài năm. |
| Hội đồng NCKH | Xét duyệt công nhận đề tài năm | - Cán bộ xét duyệt đề tài để công nhận đề tài năm;  - Nếu xét duyệt đề tài là không duyệt thì đề tài không được không được xét để ứng dụng tại đơn vị. |
| Chủ tịch hội đồng NCKH đề xuất theo yêu cầu của đơn vị | Xét duyệt đề tài ứng dụng cho đơn vị | - Cán bộ xét duyệt đề tài để ứng dụng vào hoạt động của các đơn vị;  - Nếu xét duyệt của đề tài là không cho phép thì đề tài không được ứng dụng vào hoạt động của đơn vị |
| Hội đồng NCKH;  Bộ phận thụ hưởng đề tài | Cập nhật các biên bản cần thiết | - Cán bộ cập nhật các biên bản cần thiết, bao gồm: Biên bản kế hoạch ứng dụng đề tài, báo cáo quý năm, văn bản áp dụng đề tài lâu dài. |
| Thư ký hội đồng NCKH |  | Các cán bộ được phân công lưu trữ hồ sơ theo từng năm |

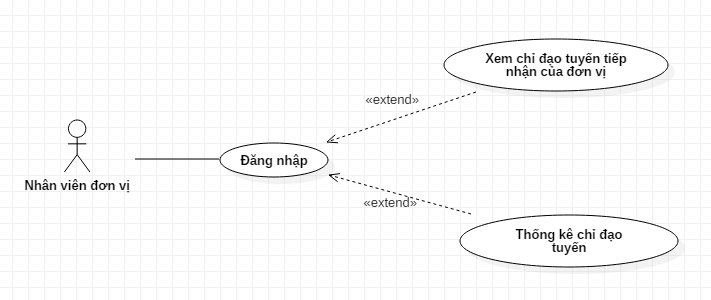
## **3.3. Phân tích thiết kế hệ thống**

### **3.3.1. Hệ thống chỉ đạo tuyến**

**❖ Sơ đồ Use case tổng quát**



###### Hình 3.1. Sơ đồ Use case tổng quát (cán bộ phụ trách chỉ đạo tuyến)



###### Hình 3.2. Sơ đồ Use case tổng quát (nhân viên đơnvị tiếp nhận chỉ đạo tuyến)

**❖ Đặc tả Use case**

**Use case Đăng nhập**

|  |  |
| --- | --- |
| Mã use case | UC01 |
| Tên use case | Đăng nhập |
| Tác nhân chính | Cán bộ phụ trách chỉ đạo tuyến, nhân viên đơn vị |
| Tiền điều kiện | Không có |
| Hậu điều kiện | Đăng nhập thành công vào hệ thống |
| Điều kiện tối thiểu | Yêu cầu nhân viên quản lý đăng nhập lại |
| Sự kiện chính | 1. Hệ thống hiển thị màn hình đăng nhập  2. Nhân viên quản lý nhập thông tin tài khoản và mật khẩu  3. Nhân viên quản lý nhấn “Đăng nhập”  4. Hệ thống chứng thực tài khoản  5. Hệ thống thông báo đăng nhập thành công |
| Ngoại lệ | 4.1. Hệ thống chứng thực thất bại, yêu cầu nhân viên nhập lại thông tin tài khoản, mật khẩu. |

**Use case Quản lý hình thức chỉ đạo tuyến**

|  |  |
| --- | --- |
| Mã use case | UC02 |
| Tên use case | Quản lý hình thức chỉ đạo tuyến. |
| Tác nhân chính | Cán bộ phụ trách chỉ đạo tuyến. |
| Tiền điều kiện | Đăng nhập thành công vào hệ thống. |
| Hậu điều kiện | Cập nhật thành công hình thức chỉ đạo tuyến. |
| Điều kiện tối thiểu | Loại bỏ thông tin đã nhập và quay lại bước trước. |
| Sự kiện chính | 1. Hệ thống hiển thị màn hình quản lý hình thức chỉ đạo tuyến.  2. Cán bộ phụ trách nhấn vào nút chức năng muốn thực hiện (Thêm, sửa, xóa).  3. Cán bộ phụ trách nhập thông tin cần thiết của một hình thức chỉ đạo tuyến.  4. Cán bộ phụ trách xác nhận lại thông tin và cập nhật dữ liệu.  5. Hệ thống chứng thực thông tin hình thức chỉ đạo tuyến.  6. Hệ thống thông báo cập nhật thành công. |
| Ngoại lệ | 5.1. Hệ thống chứng thực thất bại, quay lại màn hình quản lý hình thức chỉ đạo tuyến. |

**Use case Quản lý loại chỉ đạo tuyến**

|  |  |
| --- | --- |
| Mã use case | UC03 |
| Tên use case | Quản lý loại chỉ đạo tuyến. |
| Tác nhân chính | Cán bộ phụ trách chỉ đạo tuyến. |
| Tiền điều kiện | Đăng nhập thành công vào hệ thống. |
| Hậu điều kiện | Cập nhật thành công loại chỉ đạo tuyến. |
| Điều kiện tối thiểu | Loại bỏ thông tin đã nhập và quay lại bước trước. |
| Sự kiện chính | 1. Hệ thống hiển thị màn hình quản lý loại chỉ đạo tuyến.  2. Cán bộ phụ trách nhấn vào nút chức năng muốn thực hiện (Thêm, sửa, xóa).  3. Cán bộ phụ trách nhập thông tin cần thiết của một loại chỉ đạo tuyến.  4. Cán bộ phụ trách xác nhận lại thông tin và cập nhật dữ liệu.  5. Hệ thống chứng thực thông tin loại chỉ đạo tuyến.  6. Hệ thống thông báo cập nhật thành công. |
| Ngoại lệ | 4.1. Hệ thống chứng thực thất bại, quay lại màn hình quản lý loại chỉ đạo tuyến. |

**Use case Quản lý nội dung chỉ đạo tuyến**

|  |  |
| --- | --- |
| Mã use case | UC04 |
| Tên use case | Quản lý nội dung chỉ đạo tuyến. |
| Tác nhân chính | Cán bộ phụ trách chỉ đạo tuyến. |
| Tiền điều kiện | Đăng nhập thành công vào hệ thống. |
| Hậu điều kiện | Cập nhật thành công nội dung chỉ đạo tuyến. |
| Điều kiện tối thiểu | Loại bỏ thông tin đã nhập và quay lại bước trước. |
| Sự kiện chính | 1. Hệ thống hiển thị màn hình quản lý nội dung chỉ đạo tuyến.  2. Cán bộ phụ trách nhấn vào nút chức năng muốn thực hiện (Thêm, sửa, xóa).  3. Cán bộ phụ trách nhập thông tin cần thiết của một nội dung chỉ đạo tuyến.  4. Cán bộ phụ trách xác nhận lại thông tin và cập nhật dữ liệu.  5. Hệ thống chứng thực thông tin nội dung chỉ đạo tuyến.  6. Hệ thống thông báo cập nhật thành công. |
| Ngoại lệ | 5.1. Hệ thống chứng thực thất bại, quay lại màn hình quản lý nội dung chỉ đạo tuyến. |

**Use case Quản lý đối tượng tiếp nhận chỉ đạo tuyến**

|  |  |
| --- | --- |
| Mã use case | UC05 |
| Tên use case | Quản lý đối tượng tiếp nhận chỉ đạo tuyến. |
| Tác nhân chính | Cán bộ phụ trách chỉ đạo tuyến. |
| Tiền điều kiện | Đăng nhập thành công vào hệ thống. |
| Hậu điều kiện | Cập nhật thành công đối tượng tiếp nhận chỉ đạo tuyến. |
| Điều kiện tối thiểu | Loại bỏ thông tin đã nhập và quay lại bước trước. |
| Sự kiện chính | 1. Hệ thống hiển thị màn hình quản lý đối tượng tiếp nhận chỉ đạo tuyến.  2. Cán bộ phụ trách nhấn vào nút chức năng muốn thực hiện (Thêm, sửa, xóa).  3. Cán bộ phụ trách nhập thông tin cần thiết của một đối tượng tiếp nhận chỉ đạo tuyến.  4. Cán bộ phụ trách xác nhận lại thông tin và cập nhật dữ liệu.  5. Hệ thống chứng thực thông tin đối tượng tiếp nhận chỉ đạo tuyến.  6. Hệ thống thông báo cập nhật thành công. |
| Ngoại lệ | 4.1. Hệ thống chứng thực thất bại, quay lại màn hình quản lý đối tượng tiếp nhận chỉ đạo tuyến. |

**Use case Quản lý đánh giá chỉ đạo tuyến**

|  |  |
| --- | --- |
| Mã use case | UC06 |
| Tên use case | Quản lý đánh giá chỉ đạo tuyến. |
| Tác nhân chính | Cán bộ phụ trách chỉ đạo tuyến. |
| Tiền điều kiện | Đăng nhập thành công vào hệ thống. |
| Hậu điều kiện | Cập nhật thành công đánh giá chỉ đạo tuyến. |
| Điều kiện tối thiểu | Loại bỏ thông tin đã nhập và quay lại bước trước. |
| Sự kiện chính | 1. Hệ thống hiển thị màn hình quản lý đánh giá chỉ đạo tuyến.  2. Cán bộ phụ trách nhấn vào nút chức năng muốn thực hiện (Thêm, sửa, xóa).  3. Cán bộ phụ trách nhập thông tin cần thiết của một đánh giá chỉ đạo tuyến.  4. Cán bộ phụ trách xác nhận lại thông tin và cập nhật dữ liệu.  5. Hệ thống chứng thực thông tin đánh giá chỉ đạo tuyến.  6. Hệ thống thông báo cập nhật thành công. |
| Ngoại lệ | 4.1. Hệ thống chứng thực thất bại, quay lại màn hình quản lý đánh giá chỉ đạo tuyến. |

**Use case Quản lý văn bản chỉ đạo tuyến**

|  |  |
| --- | --- |
| Mã use case | UC07 |
| Tên use case | Quản lý văn bản chỉ đạo tuyến. |
| Tác nhân chính | Cán bộ phụ trách chỉ đạo tuyến. |
| Tiền điều kiện | Đăng nhập thành công vào hệ thống. |
| Hậu điều kiện | Cập nhật thành công văn bản chỉ đạo tuyến. |
| Điều kiện tối thiểu | Loại bỏ thông tin đã nhập và quay lại bước trước. |
| Sự kiện chính | 1. Hệ thống hiển thị màn hình quản lý văn bản chỉ đạo tuyến.  2. Cán bộ phụ trách nhấn vào nút chức năng muốn thực hiện (Thêm, sửa, xóa).  3. Cán bộ phụ trách nhập thông tin cần thiết của một văn bản chỉ đạo tuyến.  4. Cán bộ phụ trách xác nhận lại thông tin và cập nhật dữ liệu.  5. Hệ thống chứng thực thông tin văn bản chỉ đạo tuyến.  6. Hệ thống thông báo cập nhật thành công. |
| Ngoại lệ | 4.1. Hệ thống chứng thực thất bại, quay lại màn hình quản lý văn bản chỉ đạo tuyến. |

**Use case Quản lý kế hoạch chỉ đạo tuyến**

|  |  |
| --- | --- |
| Mã use case | UC08 |
| Tên use case | Quản lý kế hoạch chỉ đạo tuyến. |
| Tác nhân chính | Cán bộ phụ trách chỉ đạo tuyến. |
| Tiền điều kiện | Đăng nhập thành công vào hệ thống. |
| Hậu điều kiện | Cập nhật thành công kế hoạch chỉ đạo tuyến. |
| Điều kiện tối thiểu | Loại bỏ thông tin đã nhập và quay lại bước trước. |
| Sự kiện chính | 1. Hệ thống hiển thị màn hình quản lý kế hoạch chỉ đạo tuyến.  2. Cán bộ phụ trách nhấn vào nút chức năng muốn thực hiện (Thêm, sửa, xóa).  3. Cán bộ phụ trách nhập thông tin cần thiết của một kế hoạch chỉ đạo tuyến.  4. Cán bộ phụ trách xác nhận lại thông tin và cập nhật dữ liệu.  5. Hệ thống chứng thực thông tin kế hoạch chỉ đạo tuyến.  6. Hệ thống thông báo cập nhật thành công. |
| Ngoại lệ | 4.1. Hệ thống chứng thực thất bại, quay lại màn hình quản lý kế hoạch chỉ đạo tuyến. |

**Use case Quản lý chỉ đạo tuyến**

|  |  |
| --- | --- |
| Mã use case | UC09 |
| Tên use case | Quản lý chỉ đạo tuyến. |
| Tác nhân chính | Cán bộ phụ trách chỉ đạo tuyến. |
| Tiền điều kiện | Đăng nhập thành công vào hệ thống. |
| Hậu điều kiện | Cập nhật thành công chỉ đạo tuyến. |
| Điều kiện tối thiểu | Loại bỏ thông tin đã nhập và quay lại bước trước. |
| Sự kiện chính | 1. Hệ thống hiển thị màn hình quản lý chỉ đạo tuyến.  2. Cán bộ phụ trách nhấn vào nút chức năng muốn thực hiện (Thêm, sửa, xóa).  3. Cán bộ phụ trách nhập thông tin cần thiết của một chỉ đạo tuyến.  4. Cán bộ phụ trách xác nhận lại thông tin và cập nhật dữ liệu.  5. Hệ thống chứng thực thông tin chỉ đạo tuyến.  6. Hệ thống thông báo cập nhật thành công. |
| Ngoại lệ | 4.1. Hệ thống chứng thực thất bại, quay lại màn hình quản lý chỉ đạo tuyến. |

**Use case Thống kê chỉ đạo tuyến**

|  |  |
| --- | --- |
| Mã use case | UC10 |
| Tên use case | Thống kê chỉ đạo tuyến. |
| Tác nhân chính | Cán bộ phụ trách chỉ đạo tuyến. |
| Tiền điều kiện | Đăng nhập thành công vào hệ thống. |
| Hậu điều kiện | Xuất thống kê chỉ đạo tuyến |
| Điều kiện tối thiểu | Không có dữ liệu để xuất |
| Sự kiện chính | 1. Hệ thống hiển thị màn hình thống kê chỉ đạo tuyến.  2. Cán bộ phụ trách nhấn vào nút chức năng muốn thống kê  3. Hệ thống hiển thị kết quả thống kê |
| Ngoại lệ | Không có |

**❖ Các lớp**

**Hình thức**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | maHinhThuc | Kiểu số nguyên dương | Mỗi hình thức chỉ đạo tuyến có một mã riêng để phân biệt |
| 2 | tenHinhThuc | Kiểu chuỗi kí tự | Tên của hình thức chỉ đạo tuyến |
| 3 | viTri | Kiểu số nguyên dương | Vị trí ưu tiên của hình thức chỉ đạo tuyến |
| 4 | trangThai | Kiểu số nguyên | Trạng thái của hình thức chỉ đạo tuyến |

**Loại**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | maLoai | Kiểu số nguyên dương | Mỗi loại chỉ đạo tuyến có một mã riêng để phân biệt |
| 2 | tenLoai | Kiểu chuỗi kí tự | Tên của loại chỉ đạo tuyến |

**Nội dung**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | maNoiDung | Kiểu số nguyên dương | Mỗi nội dung chỉ đạo tuyến có một mã riêng để phân biệt |
| 2 | tenNoiDung | Kiểu chuỗi kí tự | Tên của nội dung chỉ đạo tuyến |
| 3 | viTri | Kiểu số nguyên dương | Vị trí ưu tiên của nội dung chỉ đạo tuyến |
| 4 | trangThai | Kiểu số nguyên | Trạng thái của nội dung chỉ đạo tuyến |

**Đối tượng**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | maDoiTuong | Kiểu số nguyên dương | Mỗi đối tượng tiếp nhận chỉ đạo tuyến có một mã riêng để phân biệt |
| 2 | tenDoiTuong | Kiểu chuỗi kí tự | Tên của đối tượng tiếp nhận chỉ đạo tuyến |
| 3 | viTri | Kiểu số nguyên dương | Vị trí ưu tiên của đối tượng tiếp nhận chỉ đạo tuyến |
| 4 | trangThai | Kiểu số nguyên | Trạng thái của đối tượng tiếp nhận chỉ đạo tuyến |

**Đánh giá**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | maDanhGia | Kiểu số nguyên dương | Mỗi đánh giá chỉ đạo tuyến có một mã riêng để phân biệt |
| 2 | tenDanhGia | Kiểu chuỗi kí tự | Tên của đánh giá chỉ đạo tuyến |
| 3 | viTri | Kiểu số nguyên dương | Vị trí ưu tiên của đánh giá chỉ đạo tuyến |
| 4 | trangThai | Kiểu số nguyên | Trạng thái của đánh giá chỉ đạo tuyến |

**Đánh giá**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | maDanhGia | Kiểu số nguyên dương | Mỗi đánh giá chỉ đạo tuyến có một mã riêng để phân biệt |
| 2 | tenDanhGia | Kiểu chuỗi kí tự | Tên của đánh giá chỉ đạo tuyến |
| 3 | viTri | Kiểu số nguyên dương | Vị trí ưu tiên của đánh giá chỉ đạo tuyến |
| 4 | trangThai | Kiểu số nguyên | Trạng thái của đánh giá chỉ đạo tuyến |

**Văn bản**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | maVanBan | Kiểu số nguyên dương | Mỗi văn bản chỉ đạo tuyến có một mã riêng để phân biệt |
| 2 | noiDungTrichYeu | Kiểu chuỗi kí tự | Nội dung trích yếu của văn bản chỉ đạo tuyến |
| 3 | ngayBanHanh | Kiểu ngày tháng | Ngày ban hành văn bản chỉ đạo tuyến |
| 4 | soCongVan | Kiểu chuỗi kí tự | Số công văn của văn bản chỉ đạo tuyến |
| 5 | tapTin | Kiểu chuỗi kí tự | Tập tin văn bản chỉ đạo tuyến |
| 6 | trangThai | Kiểu số nguyên | Trạng thái của văn bản chỉ đạo tuyến |

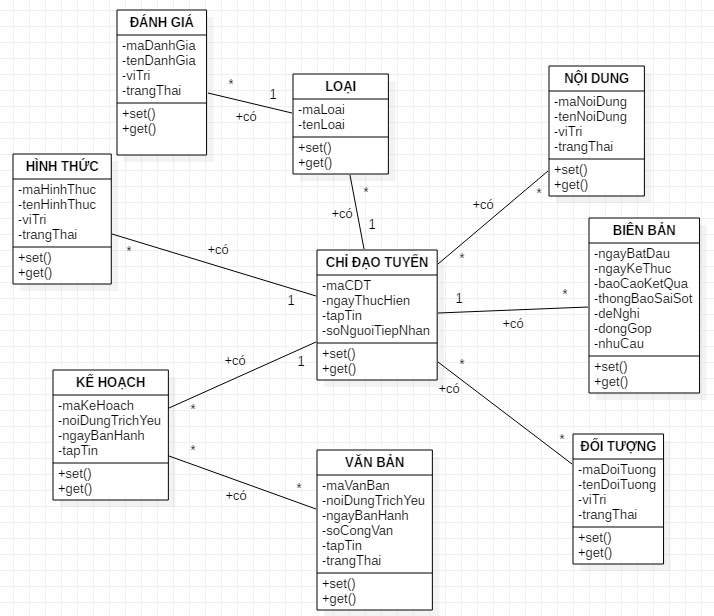
**Kế hoạch**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | maKeHoach | Kiểu số nguyên dương | Mỗi kế hoạch chỉ đạo tuyến có một mã riêng để phân biệt |
| 2 | noiDungTrichYeu | Kiểu chuỗi kí tự | Nội dung trích yếu của kế hoạch chỉ đạo tuyến |
| 3 | ngayBanHanh | Kiểu ngày tháng | Ngày ban hành kế hoạch chỉ đạo tuyến |
| 4 | tapTin | Kiểu chuỗi kí tự | Tập tin kế hoạch chỉ đạo tuyến |

**Chỉ đạo tuyến**

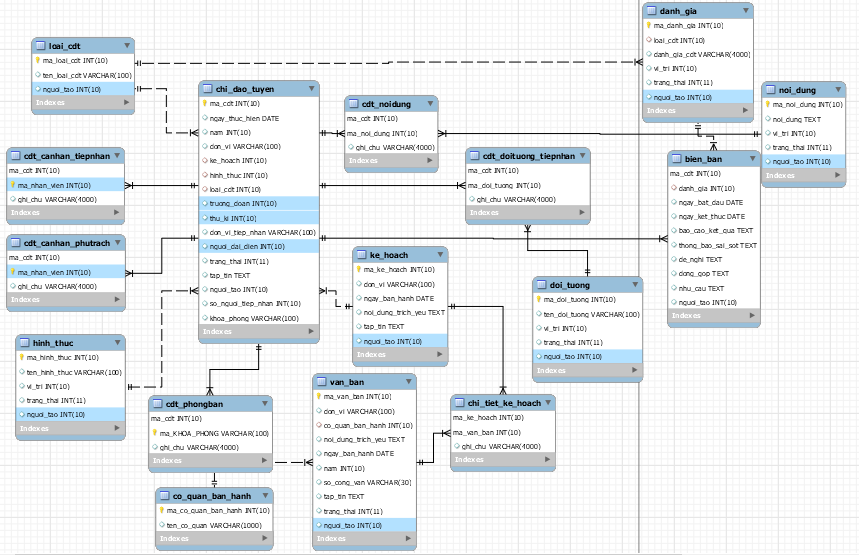
|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | maCDT | Kiểu số nguyên dương | Mỗi chỉ đạo tuyến có một mã riêng để phân biệt |
| 3 | ngayThucHien | Kiểu ngày tháng | Ngày thực hiện chỉ đạo tuyến |
| 4 | tapTin | Kiểu chuỗi kí tự | Tập tin chỉ đạo tuyến |
| 5 | soNguoiTiepNhan | Kiểu số nguyên dương | Số người tiếp nhận chỉ đạo tuyến ở đơn vị |

**❖ Sơ đồ lớp**



###### Hình 3.3. Sơ đồ lớp

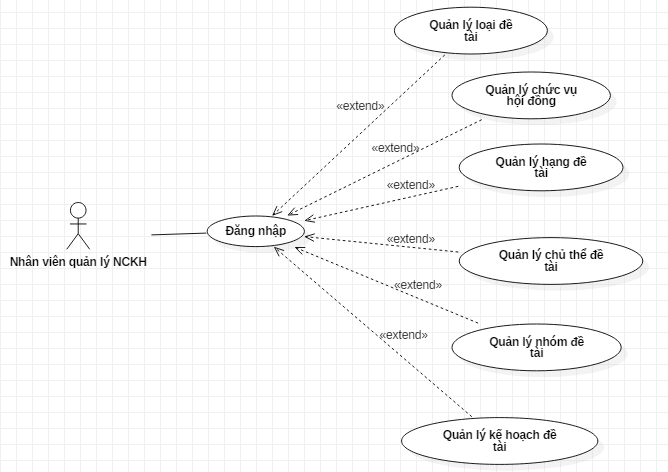
**❖ Mô hình cơ sở dữ liệu**

****

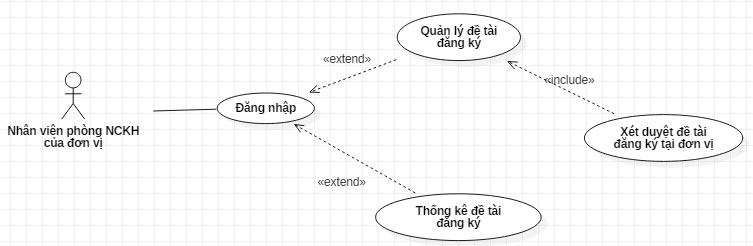
###### Hình 3.4. Mô hình cơ sở dữ liệu

### **3.3.2. Hệ thống nghiên cứu khoa học**

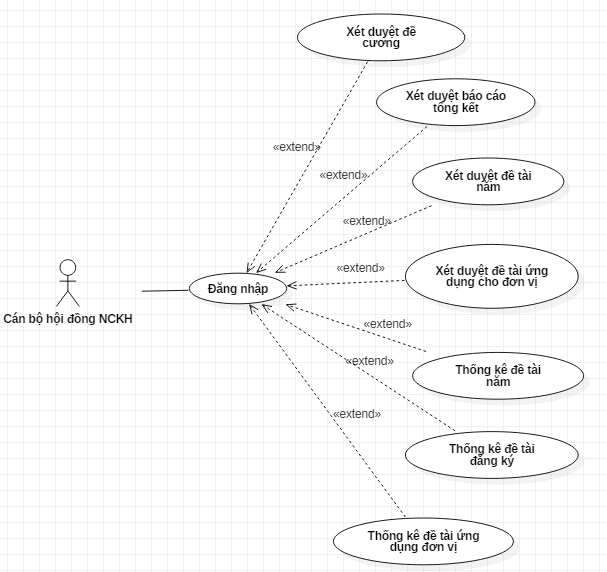
**❖ Sơ đồ Use case tổng quát**



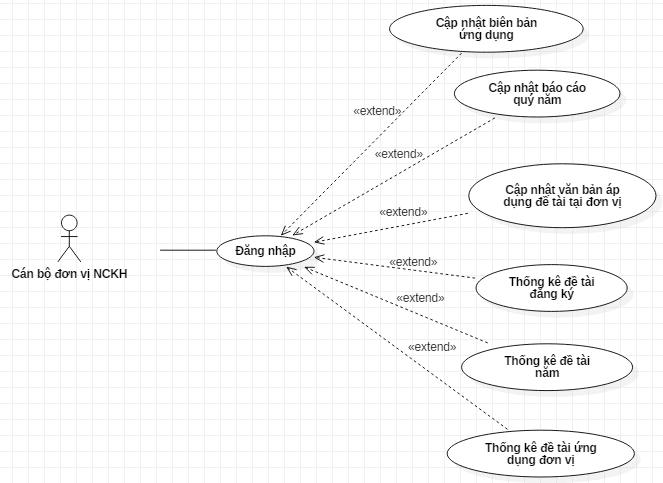
###### Hình 3.5. Sơ đồ Use case tổng quát (Nhân viên quản lý NCKH)



###### Hình 3.6. Sơ đồ Use case tổng quát (Nhân viên đơn vị)



###### Hình 3.7. Sơ đồ Use case tổng quát (Cán bộ hội đồng NCKH)



###### Hình 3.8. Sơ đồ Use case tổng quát (Cán bộ đơn vị NCKH)

**❖ Đặc tả Use case**

**Use case Đăng nhập**

|  |  |
| --- | --- |
| Mã use case | UC11 |
| Tên use case | Đăng nhập |
| Tác nhân chính | Nhân viên đơn vị, cán bộ hội đồng, cán bộ hội đồng NCKH, nhân viên quản lý NCKH |
| Tiền điều kiện | Không có |
| Hậu điều kiện | Đăng nhập thành công vào hệ thống |
| Điều kiện tối thiểu | Yêu cầu nhân viên quản lý đăng nhập lại |
| Sự kiện chính | 1. Hệ thống hiển thị màn hình đăng nhập.  2. Nhân viên quản lý nhập thông tin tài khoản và mật khẩu.  3. Nhân viên quản lý nhấn “Đăng nhập”.  4. Hệ thống chứng thực tài khoản.  5. Hệ thống thông báo đăng nhập thành công |
| Ngoại lệ | 4.1. Hệ thống chứng thực thất bại, yêu cầu nhân viên nhập lại thông tin tài khoản, mật khẩu. |

**Use case Quản lý loại đề tài**

|  |  |
| --- | --- |
| Mã use case | UC12 |
| Tên use case | Quản lý loại đề tài |
| Tác nhân chính | Nhân viên quản lý NCKH |
| Tiền điều kiện | Đăng nhập thành công vào hệ thống |
| Hậu điều kiện | Cập nhật thành công loại đề tài |
| Điều kiện tối thiểu | Loại bỏ thông tin đã nhập và quay lại bước trước |
| Sự kiện chính | 1. Hệ thống hiển thị màn hình quản lý loại đề tài.  2. Nhân viên nhấn vào nút chức năng muốn thực hiện (Thêm, sửa, xóa).  3. Nhân viên nhập thông tin cần thiết của loại đề tài.  4. Nhân viên xác nhận lại thông tin và cập nhật dữ liệu.  5. Hệ thống chứng thực thông tin loại đề tài.  6. Hệ thống thông báo cập nhật thành công. |
| Ngoại lệ | 5.1. Hệ thống chứng thực thất bại, quay lại màn hình quản lý loại đề tài. |

**Use case Quản lý chức vụ hội đồng**

|  |  |
| --- | --- |
| Mã use case | UC13 |
| Tên use case | Quản lý chức vụ hội đồng |
| Tác nhân chính | Nhân viên quản lý NCKH |
| Tiền điều kiện | Đăng nhập thành công vào hệ thống |
| Hậu điều kiện | Cập nhật thành công chức vụ hội đồng |
| Điều kiện tối thiểu | Loại bỏ thông tin đã nhập và quay lại bước trước |
| Sự kiện chính | 1. Hệ thống hiển thị màn hình quản lý chức vụ hội đồng.  2. Nhân viên nhấn vào nút chức năng muốn thực hiện (Thêm, sửa, xóa).  3. Nhân viên nhập thông tin cần thiết của chức vụ hội đồng.  4. Nhân viên xác nhận lại thông tin và cập nhật dữ liệu.  5. Hệ thống chứng thực thông tin chức vụ hội đồng.  6. Hệ thống thông báo cập nhật thành công. |
| Ngoại lệ | 5.1. Hệ thống chứng thực thất bại, quay lại màn hình quản lý chức vụ hội đồng. |

**Use case Quản lý hạng đề tài**

|  |  |
| --- | --- |
| Mã use case | UC14 |
| Tên use case | Quản lý hạng đề tài |
| Tác nhân chính | Nhân viên quản lý NCKH |
| Tiền điều kiện | Đăng nhập thành công vào hệ thống |
| Hậu điều kiện | Cập nhật thành công hạng đề tài |
| Điều kiện tối thiểu | Loại bỏ thông tin đã nhập và quay lại bước trước |
| Sự kiện chính | 1. Hệ thống hiển thị màn hình quản lý hạng đề tài.  2. Nhân viên nhấn vào nút chức năng muốn thực hiện (Thêm, sửa, xóa).  3. Nhân viên nhập thông tin cần thiết của hạng đề tài.  4. Nhân viên xác nhận lại thông tin và cập nhật dữ liệu.  5. Hệ thống chứng thực thông tin hạng đề tài.  6. Hệ thống thông báo cập nhật thành công. |
| Ngoại lệ | 5.1. Hệ thống chứng thực thất bại, quay lại màn hình quản lý hạng đề tài. |

**Use case Quản lý chủ thể đề tài**

|  |  |
| --- | --- |
| Mã use case | UC15 |
| Tên use case | Quản lý chủ thể đề tài |
| Tác nhân chính | Nhân viên quản lý NCKH |
| Tiền điều kiện | Đăng nhập thành công vào hệ thống |
| Hậu điều kiện | Cập nhật thành công hạng đề tài |
| Điều kiện tối thiểu | Loại bỏ thông tin đã nhập và quay lại bước trước |
| Sự kiện chính | 1. Hệ thống hiển thị màn hình quản lý hạng đề tài.  2. Nhân viên nhấn vào nút chức năng muốn thực hiện (Thêm, sửa, xóa).  3. Nhân viên nhập thông tin cần thiết của hạng đề tài.  4. Nhân viên xác nhận lại thông tin và cập nhật dữ liệu.  5. Hệ thống chứng thực thông tin hạng đề tài.  6. Hệ thống thông báo cập nhật thành công. |
| Ngoại lệ | 5.1. Hệ thống chứng thực thất bại, quay lại màn hình quản lý hạng đề tài. |

**Use case Quản lý nhóm đề tài**

|  |  |
| --- | --- |
| Mã use case | UC16 |
| Tên use case | Quản lý nhóm đề tài |
| Tác nhân chính | Nhân viên quản lý NCKH |
| Tiền điều kiện | Đăng nhập thành công vào hệ thống |
| Hậu điều kiện | Cập nhật thành công nhóm đề tài |
| Điều kiện tối thiểu | Loại bỏ thông tin đã nhập và quay lại bước trước |
| Sự kiện chính | 1. Hệ thống hiển thị màn hình quản lý nhóm đề tài.  2. Nhân viên nhấn vào nút chức năng muốn thực hiện (Thêm, sửa, xóa).  3. Nhân viên nhập thông tin cần thiết của nhóm đề tài.  4. Nhân viên xác nhận lại thông tin và cập nhật dữ liệu.  5. Hệ thống chứng thực thông tin nhóm đề tài.  6. Hệ thống thông báo cập nhật thành công. |
| Ngoại lệ | 5.1. Hệ thống chứng thực thất bại, quay lại màn hình quản lý nhóm đề tài. |

**Use case Quản lý kế hoạch đề tài**

|  |  |
| --- | --- |
| Mã use case | UC17 |
| Tên use case | Quản lý kế hoạch đề tài |
| Tác nhân chính | Nhân viên quản lý NCKH |
| Tiền điều kiện | Đăng nhập thành công vào hệ thống |
| Hậu điều kiện | Cập nhật thành công kế hoạch đề tài |
| Điều kiện tối thiểu | Loại bỏ thông tin đã nhập và quay lại bước trước |
| Sự kiện chính | 1. Hệ thống hiển thị màn hình quản lý kế hoạch đề tài.  2. Nhân viên nhấn vào nút chức năng muốn thực hiện (Thêm, sửa, xóa).  3. Nhân viên nhập thông tin cần thiết của kế hoạch đề tài.  4. Nhân viên xác nhận lại thông tin và cập nhật dữ liệu.  5. Hệ thống chứng thực thông tin kế hoạch đề tài.  6. Hệ thống thông báo cập nhật thành công. |
| Ngoại lệ | 5.1. Hệ thống chứng thực thất bại, quay lại màn hình quản lý kế hoạch đề tài. |

**Use case Quản lý đề tài đăng ký**

|  |  |
| --- | --- |
| Mã use case | UC18 |
| Tên use case | Quản lý đề tài đăng ký |
| Tác nhân chính | Nhân viên phòng NCKH của đơn vị |
| Tiền điều kiện | Đăng nhập thành công vào hệ thống |
| Hậu điều kiện | Cập nhật thành công đề tài đăng ký |
| Điều kiện tối thiểu | Loại bỏ thông tin đã nhập và quay lại bước trước |
| Sự kiện chính | 1. Hệ thống hiển thị màn hình quản lý đề tài đăng ký.  2. Nhân viên nhấn vào nút chức năng muốn thực hiện (Thêm, sửa, xóa).  3. Nhân viên nhập thông tin cần thiết của đề tài đăng ký.  4. Nhân viên xác nhận lại thông tin và cập nhật dữ liệu.  5. Hệ thống chứng thực thông tin đề tài đăng ký.  6. Hệ thống thông báo cập nhật thành công. |
| Ngoại lệ | 5.1. Hệ thống chứng thực thất bại, quay lại màn hình quản lý đề tài đăng ký. |

**Use case Xét duyệt đề tài đăng ký tại đơn vị**

|  |  |
| --- | --- |
| Mã use case | UC19 |
| Tên use case | Xét duyệt đề tài đăng ký tại đơn vị |
| Tác nhân chính | Nhân viên phòng NCKH của đơn vị |
| Tiền điều kiện | Đăng nhập thành công vào hệ thống |
| Hậu điều kiện | Xét duyệt thành công đề tài đăng ký tại đơn vị |
| Điều kiện tối thiểu | Loại bỏ thông tin đã nhập và quay lại bước trước |
| Sự kiện chính | 1. Hệ thống hiển thị màn hình xét duyệt đề tài đăng ký tại đơn vị.  2. Nhân viên nhấn chọn đề tài cần xét duyệt và nhấn nút “Xét duyệt”.  3. Nhân viên nhập thông tin xét duyệt cần thiết của đề tài.  4. Nhân viên nhấn nút “Lưu” để lưu thông tin xét duyệt.  5. Hệ thống chứng thực thông tin xét duyệt.  6. Hệ thống thông báo xét duyệt thành công. |
| Ngoại lệ | 5.1. Hệ thống chứng thực thất bại, quay lại màn hình xét duyệt đề tài. |

**Use case Thống kê đề tài đăng ký**

|  |  |
| --- | --- |
| Mã use case | UC20 |
| Tên use case | Thống kê đề tài đăng ký |
| Tác nhân chính | Nhân viên phòng NCKH của đơn vị, cán bộ Hội đồng NCKH, cán bộ đơn vị NCKH |
| Tiền điều kiện | Đăng nhập thành công vào hệ thống. |
| Hậu điều kiện | Xuất thống kê đề tài đăng ký |
| Điều kiện tối thiểu | Không có dữ liệu để xuất |
| Sự kiện chính | 1. Hệ thống hiển thị màn hình thống kê đề tài đăng ký.  2. Nhân viên chọn các thông tin đề tài cần thống kê và nhấn nút In.  3. Hệ thống hiển thị kết quả thống kê |
| Ngoại lệ | Không có |

**Use case Xét duyệt đề cương**

|  |  |
| --- | --- |
| Mã use case | UC21 |
| Tên use case | Xét duyệt đề cương |
| Tác nhân chính | Cán bộ Hội đồng NCKH |
| Tiền điều kiện | Đăng nhập thành công vào hệ thống |
| Hậu điều kiện | Xét duyệt thành công đề tài cương của đề tài |
| Điều kiện tối thiểu | Loại bỏ thông tin đã nhập và quay lại bước trước |
| Sự kiện chính | 1. Hệ thống hiển thị màn hình xét duyệt đề cương của đề tài.  2. Nhân viên nhấn chọn đề tài cần xét duyệt và nhấn nút “Xét duyệt”.  3. Nhân viên nhập thông tin xét duyệt cần thiết của đề tài.  4. Nhân viên nhấn nút “Lưu” để lưu thông tin xét duyệt.  5. Hệ thống chứng thực thông tin xét duyệt.  6. Hệ thống thông báo xét duyệt thành công. |
| Ngoại lệ | 5.1. Hệ thống chứng thực thất bại, quay lại màn hình xét duyệt đề cương. |

**Use case Xét duyệt đề cương**

|  |  |
| --- | --- |
| Mã use case | UC22 |
| Tên use case | Xét duyệt báo cáo tổng kết |
| Tác nhân chính | Cán bộ Hội đồng NCKH |
| Tiền điều kiện | Đăng nhập thành công vào hệ thống |
| Hậu điều kiện | Xét duyệt thành công đề tài cương của đề tài |
| Điều kiện tối thiểu | Loại bỏ thông tin đã nhập và quay lại bước trước |
| Sự kiện chính | 1. Hệ thống hiển thị màn hình xét duyệt báo cáo tổng kết của đề tài.  2. Nhân viên nhấn chọn đề tài cần xét duyệt và nhấn nút “Xét duyệt”.  3. Nhân viên nhập thông tin xét duyệt cần thiết của đề tài.  4. Nhân viên nhấn nút “Lưu” để lưu thông tin xét duyệt.  5. Hệ thống chứng thực thông tin xét duyệt.  6. Hệ thống thông báo xét duyệt thành công. |
| Ngoại lệ | 5.1. Hệ thống chứng thực thất bại, quay lại màn hình xét duyệt báo cáo tổng kết. |

**Use case Xét duyệt đề tài năm**

|  |  |
| --- | --- |
| Mã use case | UC23 |
| Tên use case | Xét duyệt đề tài năm |
| Tác nhân chính | Cán bộ Hội đồng NCKH |
| Tiền điều kiện | Đăng nhập thành công vào hệ thống |
| Hậu điều kiện | Xét duyệt thành công đề tài năm |
| Điều kiện tối thiểu | Loại bỏ thông tin đã nhập và quay lại bước trước |
| Sự kiện chính | 1. Hệ thống hiển thị màn hình xét duyệt đề tài năm.  2. Nhân viên nhấn chọn đề tài cần xét duyệt và nhấn nút “Xét duyệt”.  3. Nhân viên nhập thông tin xét duyệt cần thiết của đề tài.  4. Nhân viên nhấn nút “Lưu” để lưu thông tin xét duyệt.  5. Hệ thống chứng thực thông tin xét duyệt.  6. Hệ thống thông báo xét duyệt thành công. |
| Ngoại lệ | 5.1. Hệ thống chứng thực thất bại, quay lại màn hình xét duyệt đề tài năm. |

**Use case Xét duyệt đề tài ứng dụng cho đơn vị**

|  |  |
| --- | --- |
| Mã use case | UC24 |
| Tên use case | Xét duyệt đề tài ứng dụng cho đơn vị |
| Tác nhân chính | Cán bộ Hội đồng NCKH |
| Tiền điều kiện | Đăng nhập thành công vào hệ thống |
| Hậu điều kiện | Xét duyệt thành công đề tài ứng dụng cho đơn vị |
| Điều kiện tối thiểu | Loại bỏ thông tin đã nhập và quay lại bước trước |
| Sự kiện chính | 1. Hệ thống hiển thị màn hình xét duyệt đề tài ứng dụng cho đơn vị.  2. Nhân viên nhấn chọn đề tài cần xét duyệt và nhấn nút “Xét duyệt”.  3. Nhân viên nhập thông tin xét duyệt cần thiết của đề tài.  4. Nhân viên nhấn nút “Lưu” để lưu thông tin xét duyệt.  5. Hệ thống chứng thực thông tin xét duyệt.  6. Hệ thống thông báo xét duyệt thành công. |
| Ngoại lệ | 5.1. Hệ thống chứng thực thất bại, quay lại màn hình xét duyệt đề tài ứng dụng cho đơn vị. |

**Use case Thống kê đề tài năm**

|  |  |
| --- | --- |
| Mã use case | UC25 |
| Tên use case | Thống kê đề tài năm |
| Tác nhân chính | Nhân viên phòng NCKH của đơn vị, cán bộ đơn vị NCKH |
| Tiền điều kiện | Đăng nhập thành công vào hệ thống. |
| Hậu điều kiện | Xuất thống kê đề tài năm |
| Điều kiện tối thiểu | Không có dữ liệu để xuất |
| Sự kiện chính | 1. Hệ thống hiển thị màn hình thống kê đề tài năm.  2. Nhân viên chọn các thông tin đề tài cần thống kê và nhấn nút In.  3. Hệ thống hiển thị kết quả thống kê |
| Ngoại lệ | Không có |

**Use case Thống kê đề tài ứng dụng tại đơn vị**

|  |  |
| --- | --- |
| Mã use case | UC26 |
| Tên use case | Thống kê đề tài ứng dụng tại đơn vị |
| Tác nhân chính | Nhân viên phòng NCKH của đơn vị, cán bộ đơn vị NCKH |
| Tiền điều kiện | Đăng nhập thành công vào hệ thống. |
| Hậu điều kiện | Xuất thống kê đề tài ứng dụng tại đơn vị |
| Điều kiện tối thiểu | Không có dữ liệu để xuất |
| Sự kiện chính | 1. Hệ thống hiển thị màn hình thống kê đề tài ứng dụng tại đơn vị.  2. Nhân viên chọn các thông tin đề tài cần thống kê và nhấn nút In.  3. Hệ thống hiển thị kết quả thống kê |
| Ngoại lệ | Không có |

**Use case Cập nhật biên bản ứng dụng**

|  |  |
| --- | --- |
| Mã use case | UC27 |
| Tên use case | Cập nhật biên bản ứng dụng |
| Tác nhân chính | Cán bộ đơn vị NCKH (đơn vị đã áp dụng đề tài) |
| Tiền điều kiện | Đăng nhập thành công vào hệ thống |
| Hậu điều kiện | Cập nhật thành công biên bản ứng dụng cho đề tài |
| Điều kiện tối thiểu | Loại bỏ thông tin đã nhập và quay lại bước trước |
| Sự kiện chính | 1. Hệ thống hiển thị màn hình cập nhật biên bản ứng dụng.  2. Nhân viên nhấn chọn đề tài cần cập nhật biên bản và nhấn nút “Cập nhật biên bản”.  3. Nhân viên nhập thông tin cập nhật biên bản cần thiết của đề tài.  4. Nhân viên nhấn nút “Lưu” để lưu thông tin cập nhật.  5. Hệ thống chứng thực thông tin cập nhật.  6. Hệ thống thông báo cập nhật biên bản thành công. |
| Ngoại lệ | 5.1. Hệ thống chứng thực thất bại, quay lại màn hình cập nhật biên bản ứng dụng. |

**Use case Cập nhật báo cáo quý năm**

|  |  |
| --- | --- |
| Mã use case | UC28 |
| Tên use case | Cập nhật báo cáo quý năm |
| Tác nhân chính | Cán bộ đơn vị NCKH (đơn vị đã áp dụng đề tài) |
| Tiền điều kiện | Đăng nhập thành công vào hệ thống |
| Hậu điều kiện | Cập nhật thành công báo cáo quý năm cho đề tài |
| Điều kiện tối thiểu | Loại bỏ thông tin đã nhập và quay lại bước trước |
| Sự kiện chính | 1. Hệ thống hiển thị màn hình cập nhật báo cáo quý năm.  2. Nhân viên nhấn chọn đề tài cần cập nhật và nhấn nút “Cập nhật”.  3. Nhân viên nhập thông tin cập nhật biên bản cần thiết của đề tài.  4. Nhân viên nhấn nút “Lưu” để lưu thông tin cập nhật.  5. Hệ thống chứng thực thông tin cập nhật.  6. Hệ thống thông báo cập nhật biên bản thành công. |
| Ngoại lệ | 5.1. Hệ thống chứng thực thất bại, quay lại màn hình cập nhật báo cáo quý năm. |

**❖ Các lớp**

**Loại đề tài**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | maLoai | Kiểu số nguyên dương | Mỗi loại đề tài có một mã riêng để phân biệt |
| 2 | tenLoai | Kiểu chuỗi kí tự | Tên của loại đề tài |
| 3 | viTri | Kiểu số nguyên dương | Vị trí ưu tiên của loại đề tài |
| 4 | trangThai | Kiểu số nguyên | Trạng thái loại đề tài |

**Hạng đề tài**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | maHang | Kiểu số nguyên dương | Mỗi hạng đề tài có một mã riêng để phân biệt |
| 2 | tenHang | Kiểu chuỗi kí tự | Tên của hạng đề tài |
| 3 | viTri | Kiểu số nguyên dương | Vị trí ưu tiên của hạng đề tài |
| 4 | trangThai | Kiểu số nguyên | Trạng thái hạng đề tài |

**Chủ thể đề tài**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | maChuThe | Kiểu số nguyên dương | Mỗi chủ thể đề tài có một mã riêng để phân biệt |
| 2 | tenChuThe | Kiểu chuỗi kí tự | Tên của chủ thể đề tài |
| 3 | viTri | Kiểu số nguyên dương | Vị trí ưu tiên của chủ thể đề tài |
| 4 | trangThai | Kiểu số nguyên | Trạng thái chủ thể đề tài |

**Nhóm đề tài**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | maNhom | Kiểu số nguyên dương | Mỗi nhóm đề tài có một mã riêng để phân biệt |
| 2 | tenNhom | Kiểu chuỗi kí tự | Tên của nhóm đề tài |
| 3 | viTri | Kiểu số nguyên dương | Vị trí ưu tiên của nhóm đề tài |
| 4 | trangThai | Kiểu số nguyên | Trạng thái nhóm đề tài |

**Văn bản áp dụng**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | trangThai | Kiểu số nguyên | Trạng thái của đề tài đã cập nhật văn bản hay chưa |
| 2 | tapTin | Kiếu chuỗi kí tự | Tập tin văn bản áp dụng |

**Văn bản chấp thuận**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | trangThai | Kiểu số nguyên | Trạng thái của đề tài đã cập nhật văn bản chấp thuận hay chưa |
| 2 | tapTin | Kiếu chuỗi kí tự | Tập tin văn bản chấp thuận |

**Biên bản kế hoạch**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | trangThai | Kiểu số nguyên | Trạng thái của đề tài đã cập nhật biên bản kế hoạch hay chưa |
| 2 | tapTin | Kiếu chuỗi kí tự | Tập tin biên bản kế hoạch |

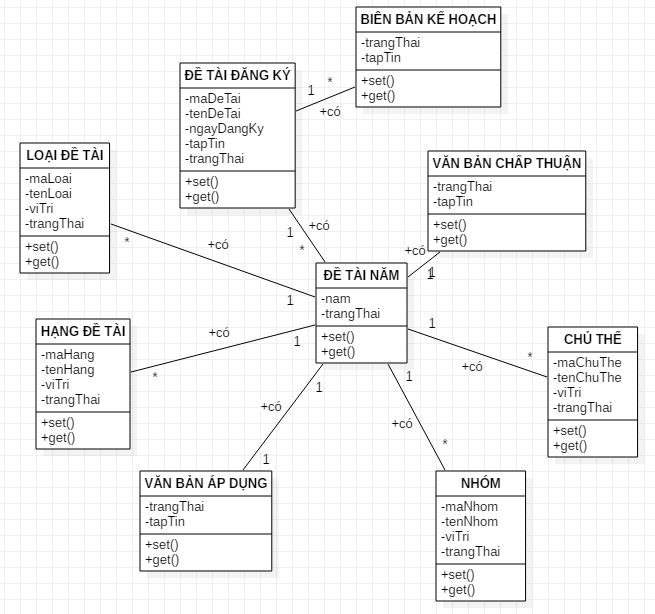
**Đề tài đăng ký**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | maDeTai | Kiểu số nguyên dương | Mỗi đề tài có một mã riêng để phân biệt |
| 2 | tenDeTai | Kiếu chuỗi kí tự | Tên đề tài đăng ký |
| 3 | ngayDangKy | Kiểu ngày tháng | Ngày đăng ký đề tài |
| 4 | tapTin | Kiểu chuỗi kí tự | Tập tin của đề tài |
| 5 | trangThai | Kiểu số nguyên | Trạng thái của đề tài |

**Đề tài năm**

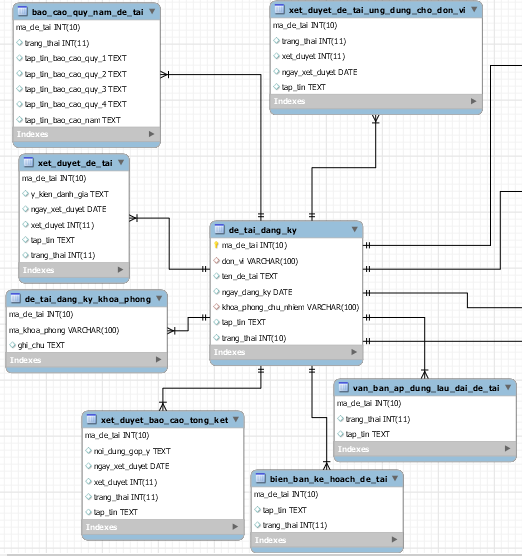
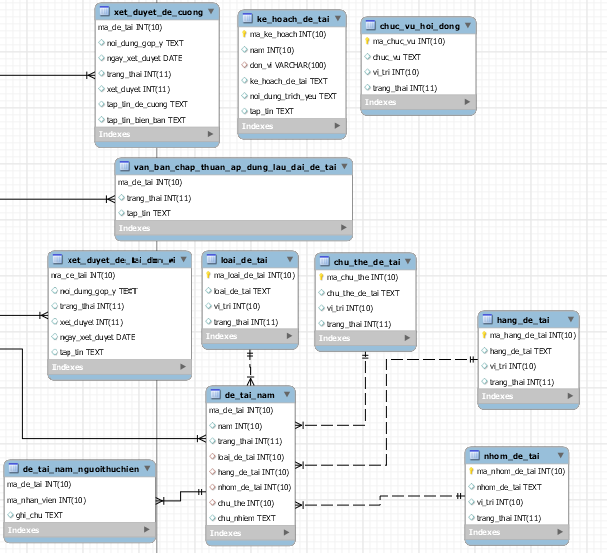
|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | nam | Kiểu số nguyên dương | Năm mà đề tài được công nhận đề tài năm |
| 2 | trangThai | Kiểu số nguyên | Trạng thái của đề tài |

**❖ Sơ đồ lớp**



###### Hình 3.9. Sơ đồ lớp

**❖ Mô hình cơ sở dữ liệu**



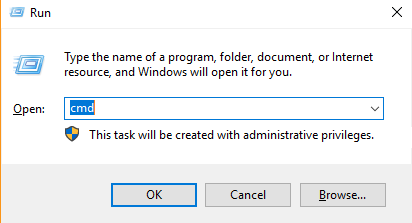
###### Hình 3.10. Mô hình CSDL

## **3.4. Cài đặt giải pháp**

### **3.4.1. Xây dựng cấu trúc Microservice cho hệ thống chỉ đạo tuyến và hệ thống NCKH**

Sử dụng khởi tạo ứng dụng tự động với JHipster:

Bước 1: Mở Command Prompt trên mát tính bằng phím tắt Windows + R, nhập cmd và nhấn Enter.

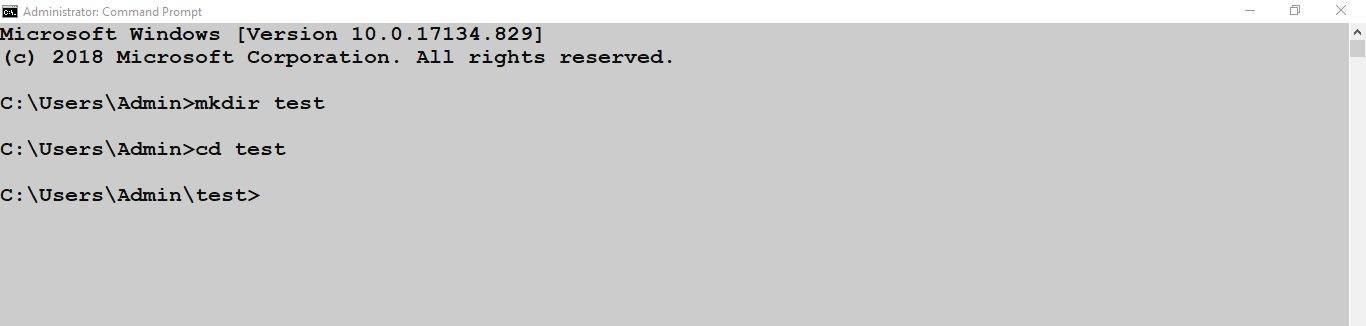


###### Hình 3.11. Xây dựng cấu trúc cho hệ thống

Bước 2: Tạo thư mục mới và di chuyển vào thư mục đó bằng cách gõ:

mkdir ten\_thu\_muc

cd ten\_thu\_muc



###### Hình 3.12. Xây dựng cấu trúc cho hệ thống

Bước 3: Tạo ứng dụng ban đầu: Gõ jhipster và nhấn Enter.



###### Hình 3.13. Xây dụng cấu trúc cho hệ thống

Sau đó chon dịch vụ nhỏ cho hệ thống. Ở đây, em sẽ chọn để tạo 4 service:

1. JHipster UAA server (for microsevice Oauth2 authencation): Dịch vụ này sẽ làm nhiệm vụ là trung tâm quản trị đăng nhập cho hệ thống.

2. Micro service gateway: Dịch vụ này sẽ hiển thị giao diện người dùng, sử dụng ReactJS để thiết kế.

3. Microservice application: Dịch vụ này sẽ là máy chủ cho dịch vụ chỉ đạo tuyến, dùng để kết nối tới CSDL.

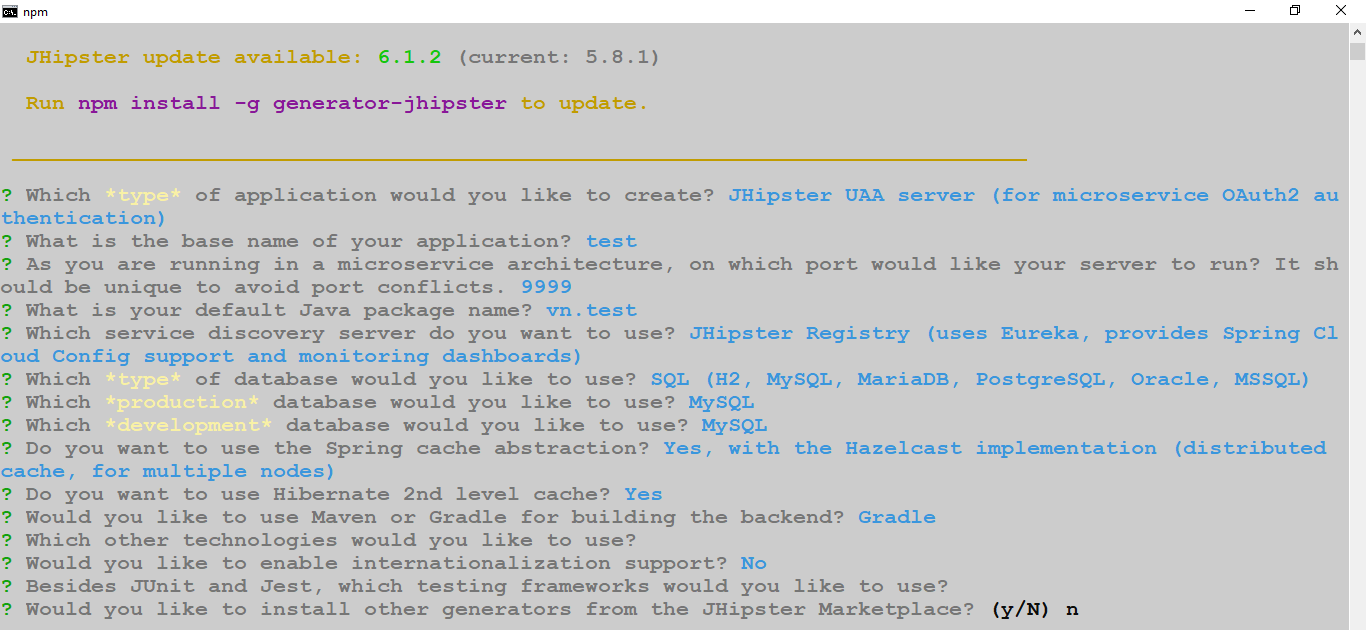
4. Microservice application: Dịch vụ này sẽ là máy chủ cho dịch vụ NCKH, dùng để kết nối tới cơ sở dữ liệu.

5. Ngoài ra cần phải có 1 dịch vụ đặc biệt của JHipster tên là JHipster Registry, có thể tải dịch vụ đó tại: https://www.jhipster.tech/jhipster-registry/

Để dễ cho việc lập trình, em sẽ gộp dịch vụ 3 và 4 lại để kết nối CSDL, và sử dụng chung một dịch vụ 2 để hiển thị giao diện.

Bắt đầu tạo dịch vụ đầu tiên (dịch vụ UAA), các dịch vụ còn lại tương tự:

Bước 1: Nhập như hình bên dưới, mỗi lần nhập nhấn Enter một lần. Sau khi nhập xong nhấn Enter và JHipster sẽ tiến hành cài đặt cấu trúc cần thiết cho dịch vụ UAA.

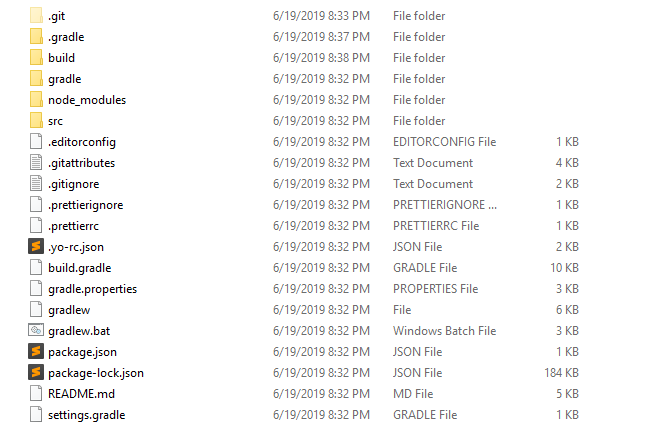


###### Hình 3.14. Xây dựng cấu trúc cho hệ thống

Sau khi cài đặt xong:

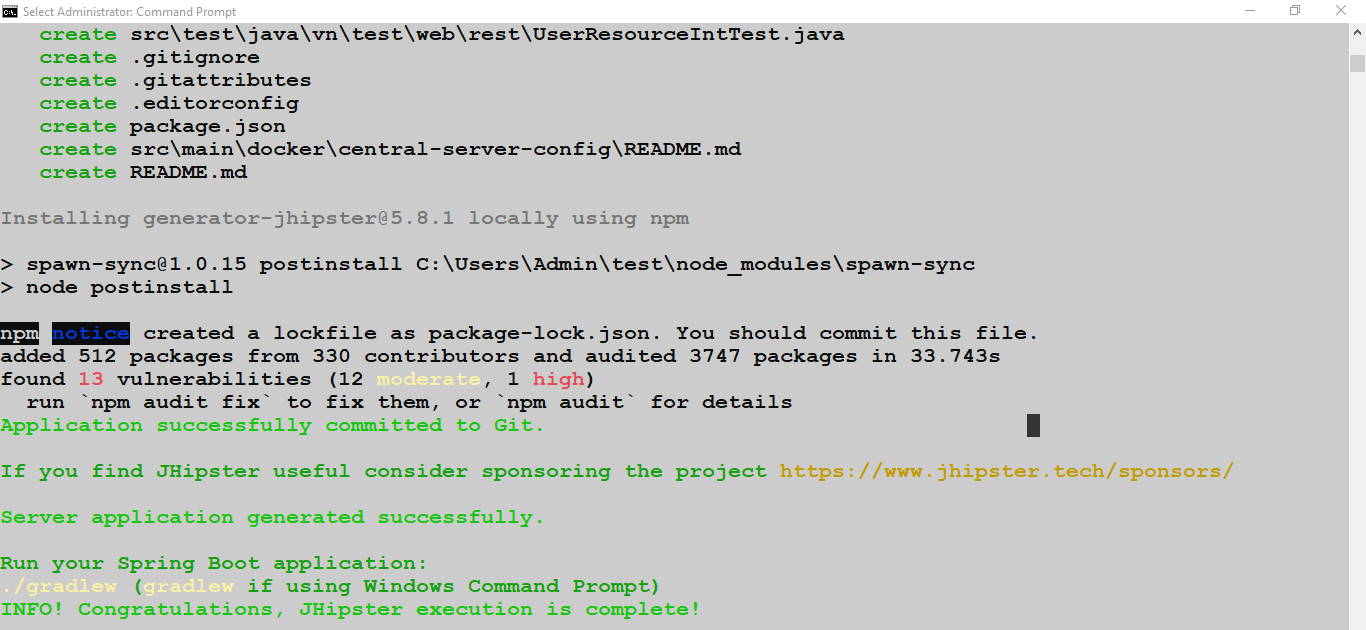
Cấu trúc thư mục sẽ có dạng như sau:

Trong File Explorer:



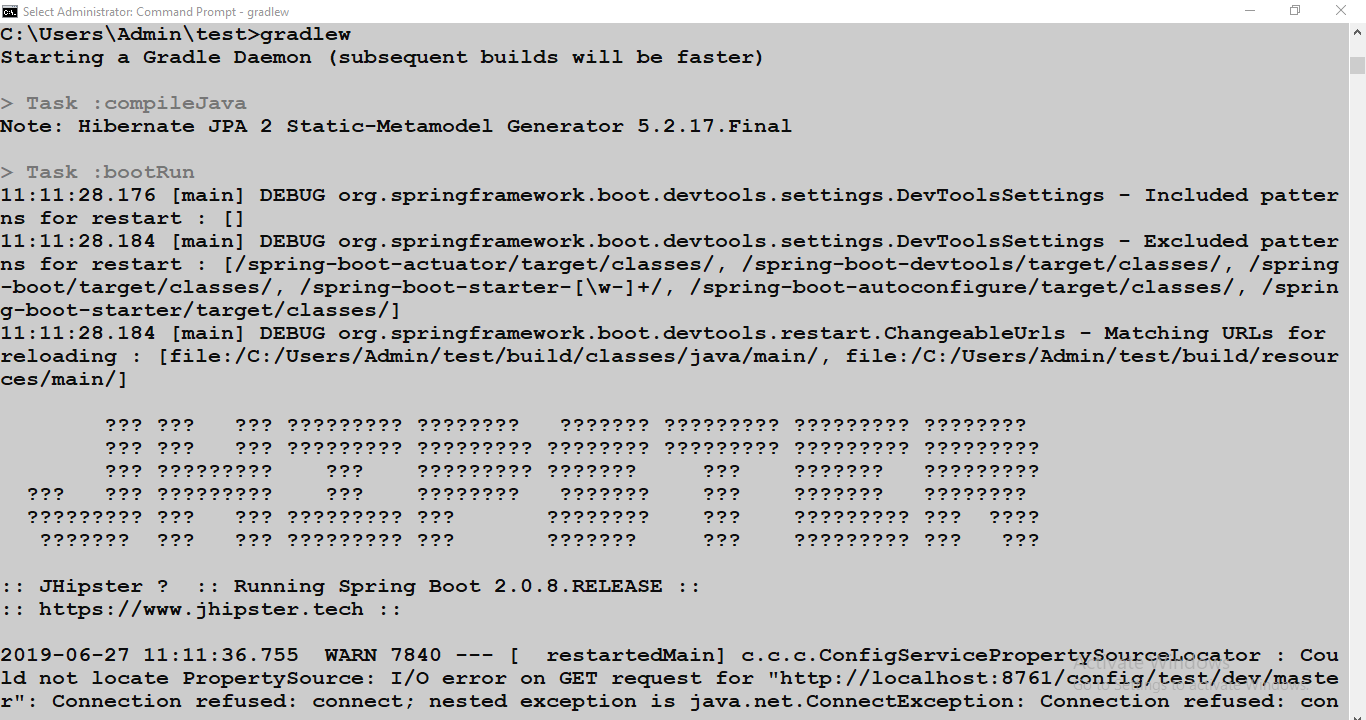
###### Hình 3.15. Xây dụng cấu trúc hệ thống

Trong màn hình Command Prompt:



###### Hình 3.16. Xây dựng cấu trúc hệ thống

Bước 2: Nhập gradlew.bat hoặc nhập gradlew, và nhấn Enter để chạy dịch vụ. Lần đầu chạy dịch vụ sẽ rất lâu. Riêng với dịch vụ Gateway và Application, cần phải chạy thêm lệnh sau: npm install (để cài đặt các gói cần thiết), npm start (để bắt đầu chạy dịch vụ).



###### Hình 3.17. Xây dụng cấu trúc hệ thống

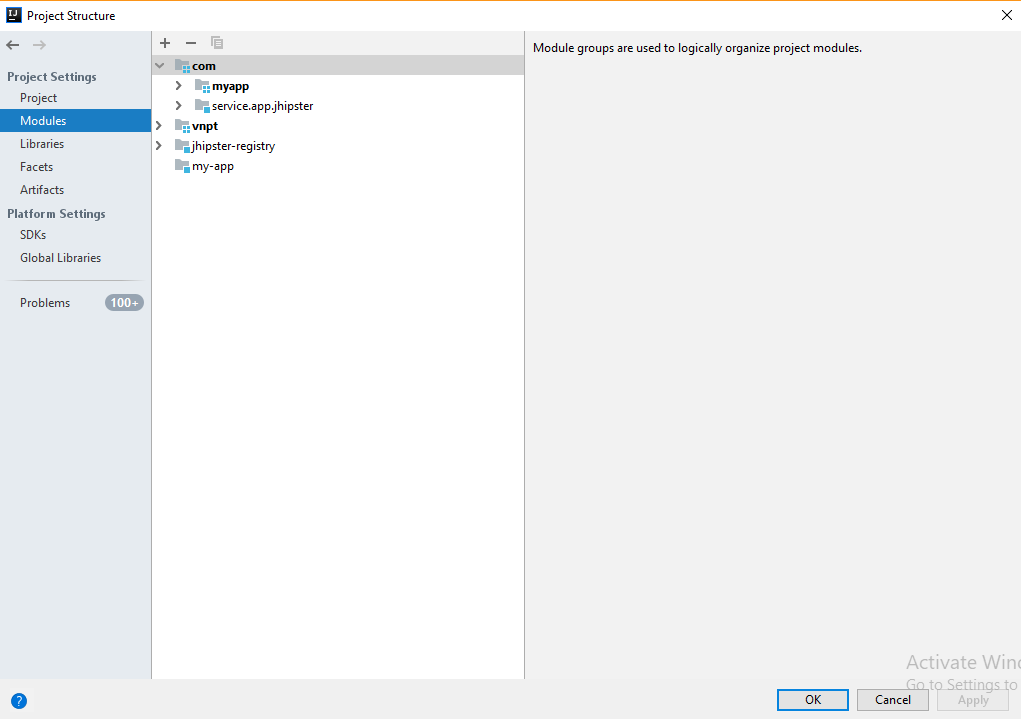
### **3.4.2. Cài đặt hệ thống**

Để chạy hệ thống, cần phải chạy tất cả dịch vụ đã nói ở mục 3.4.1, theo thứ tự sau: JHipster Registry 🡪 UAA 🡪 Application 🡪 Gateway.

Để có thể chạy hệ thống nhanh hơn, em sử dụng trình soạn thảo mã code là IntelliJ IDEA 2018.

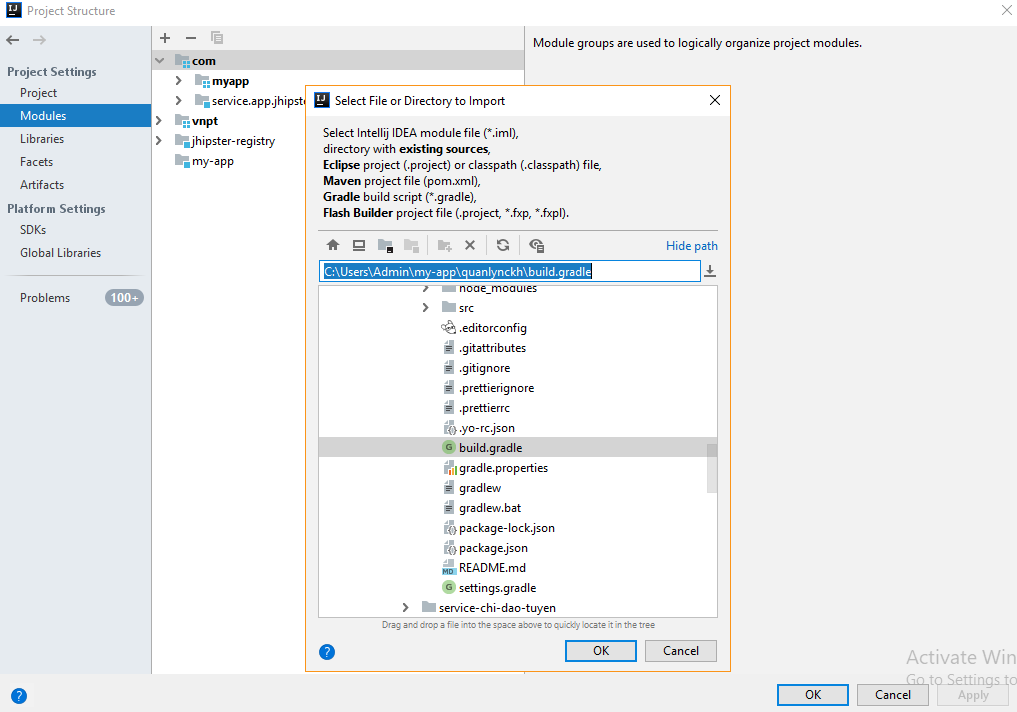
Các bước thêm dịch vụ vào IntelliJ IDEAD 2018:

Bước 1: Chọn File 🡪 Project Structure 🡪 Modules.



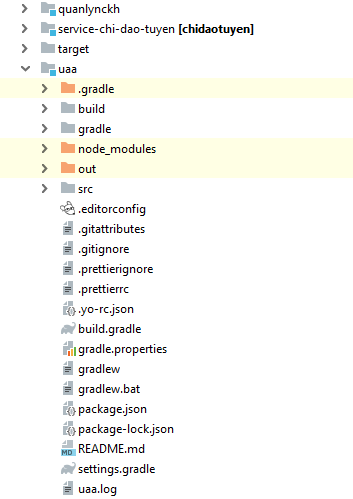
###### Hình 3.18. Thêm dịch vụ vào IntteliJ IDEA 2018

Bước 2: Nhấn vào dấu + 🡪 Chọn Import Modules 🡪 Chọn đường dẫn đến các dịch vụ đã tạo ở mục 3.4.1, chọn tập tin build.gradle 🡪 Nhấn OK.



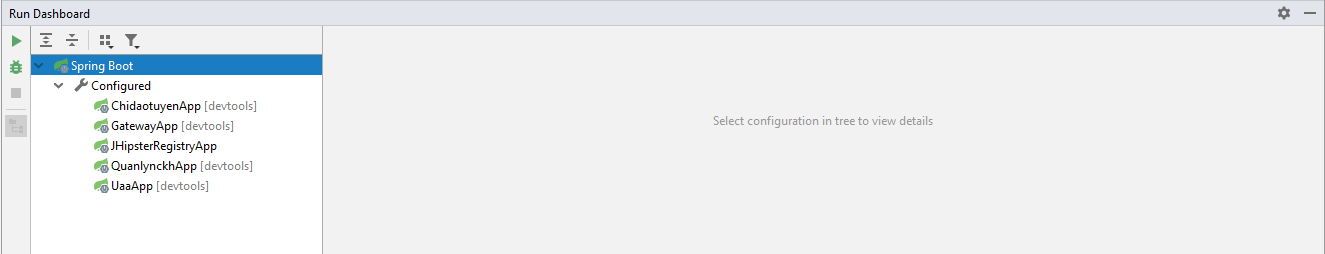
###### Hình 3.19. Thêm dịch vụ vào IntteliJ IDEA 2018

Cấu trúc thư mục sẽ như hình sau:



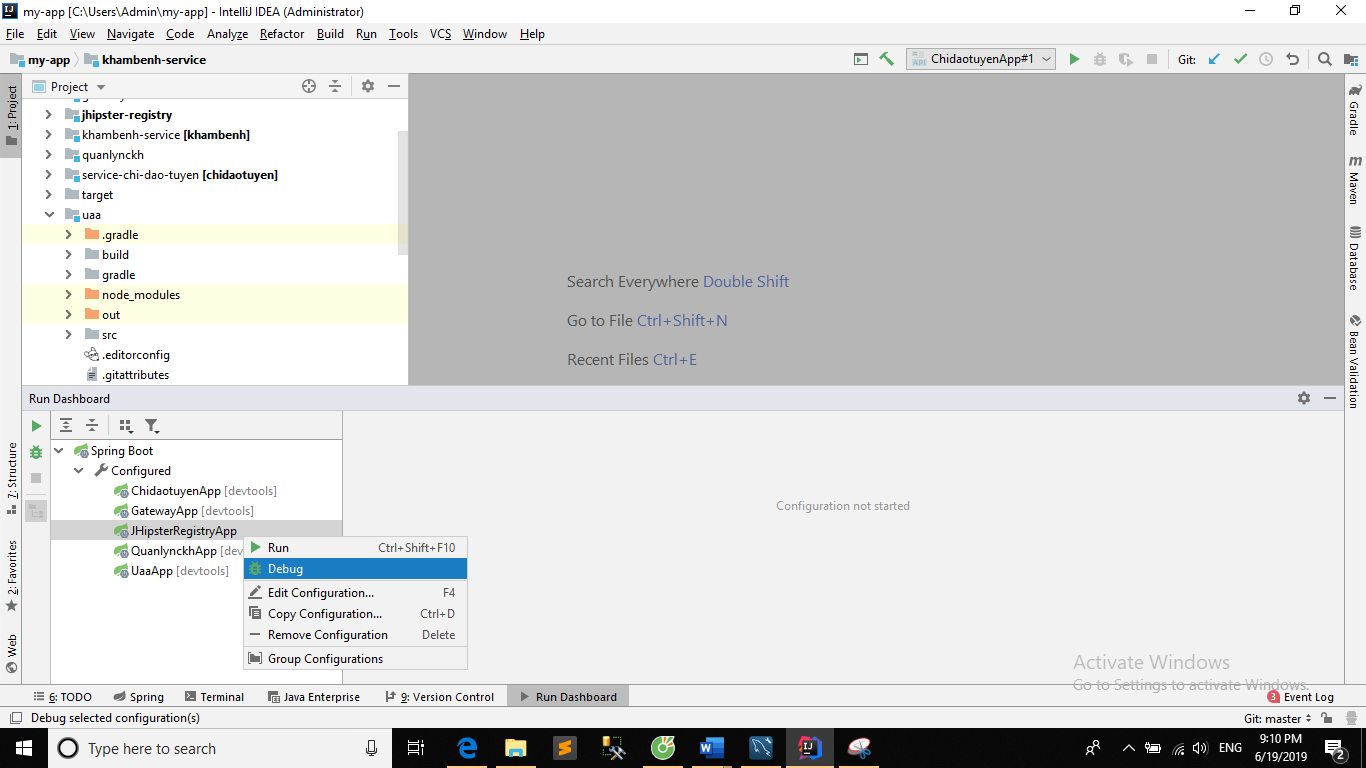
###### Hình 3.20. Thêm dịch vụ vào IntteliJ IDEA 2018

Bước 4: Chọn mục Run Dashboard



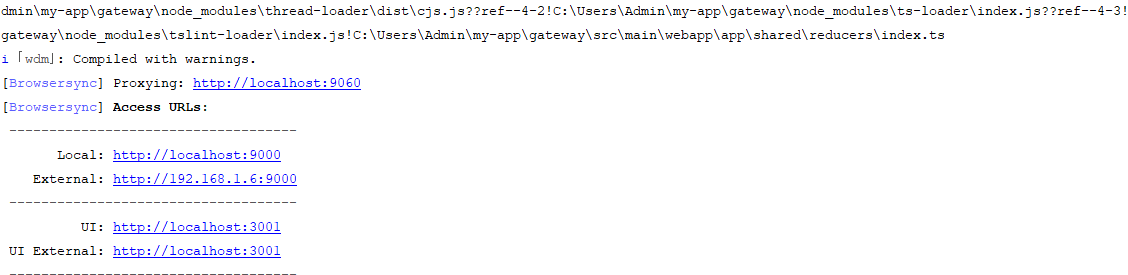
###### Hình 3.21. Chạy dịch vụ trong IntteliJ IDEA 2018

Bước 5: Nhấn chuột phải vào dịch vụ 🡪 Chọn Debug (Gỡ lỗi) hoặc Run (Chạy), ở đây sẽ chọn Debug. Làm tương tự với cái dịch vụ còn lại.



###### Hình 3.22. Chạy dịch vụ trong IntteliJ IDEA 2018

Bước 6: Tại trang Terminal 🡪 Chọn đường dẫn tới dịch vụ gateway 🡪 nhập npm start sau đó đợi dịch vụ chạy.



###### Hình 3.23. Chạy dịch vụ trong IntteliJ IDEA 2018

Bước 7: Mở một trình duyệt web và nhập địa chỉ: <http://localhost:9000/#/>



###### Hình 3.24. Dịch vụ đã được chạy

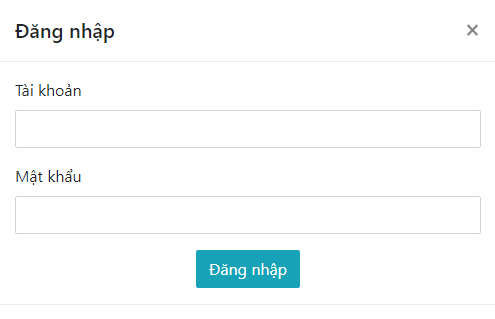
### **3.4.3 Một số hình ảnh giao diện hệ thống**

**❖ Giao diện đăng nhập**

Đăng nhập với tài khoản, mật khẩu đã được cấp trước đó, tùy vào phân quyền mà hệ thống sẽ hiển thị chức năng.

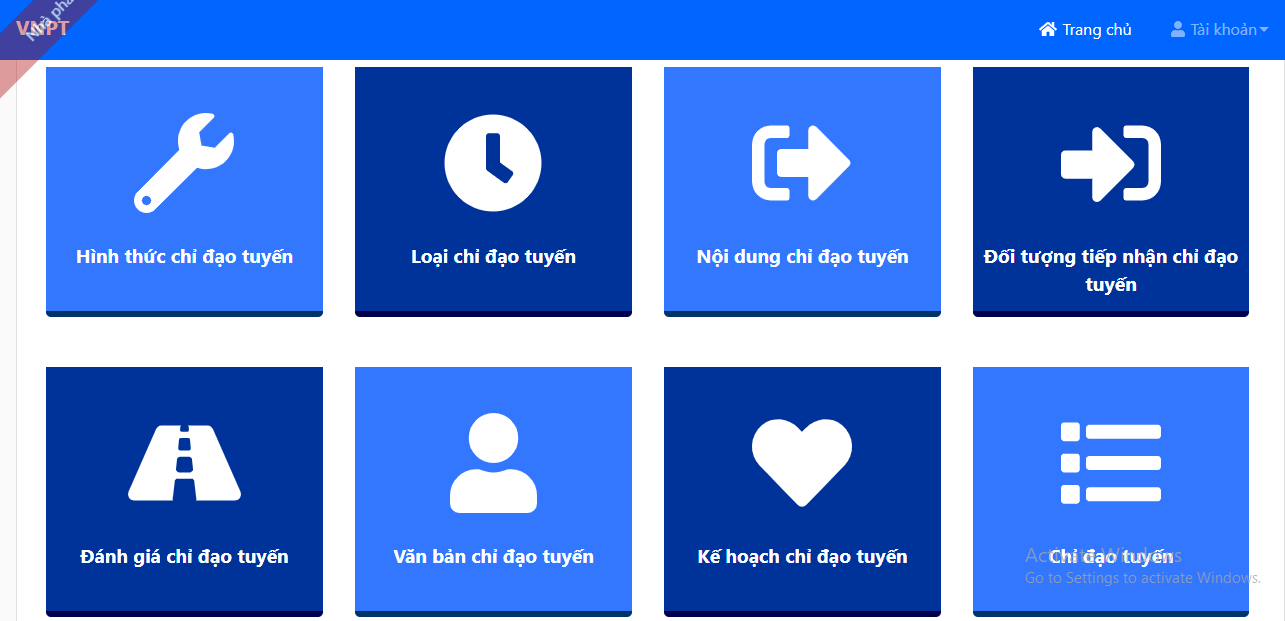
Tài khoản: admin1

Mật khẩu: admin1



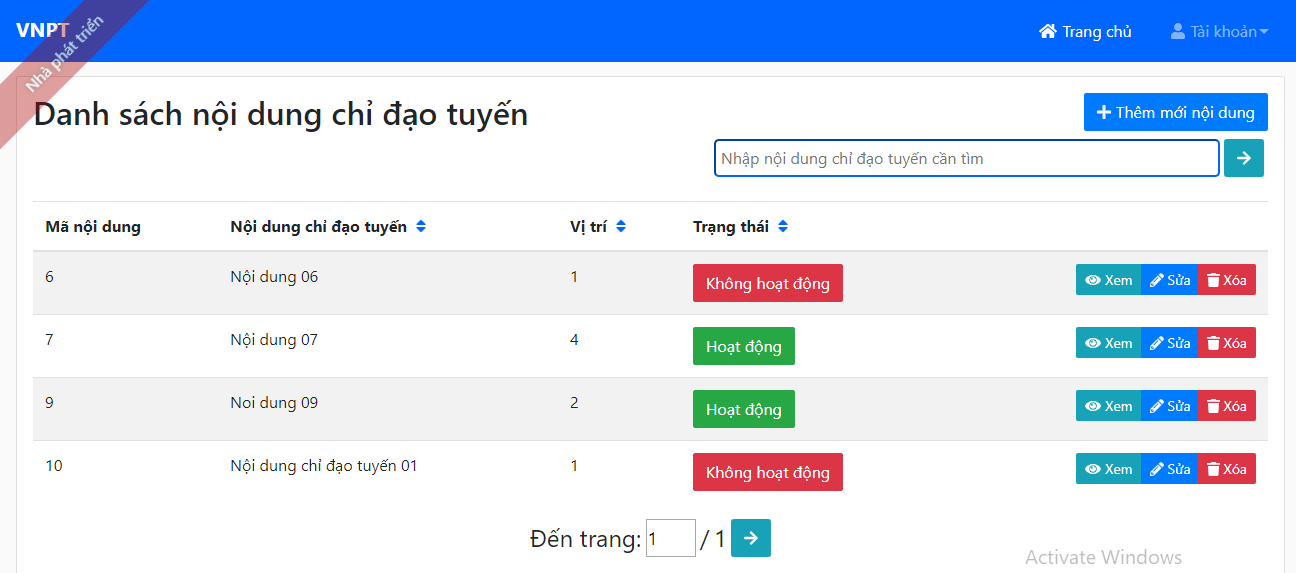
###### Hình 3.25.Giao diện đăng nhập

**❖ Giao diện màn hình chính với phân quyền Cán bộ quản lý chỉ đạo tuyến**



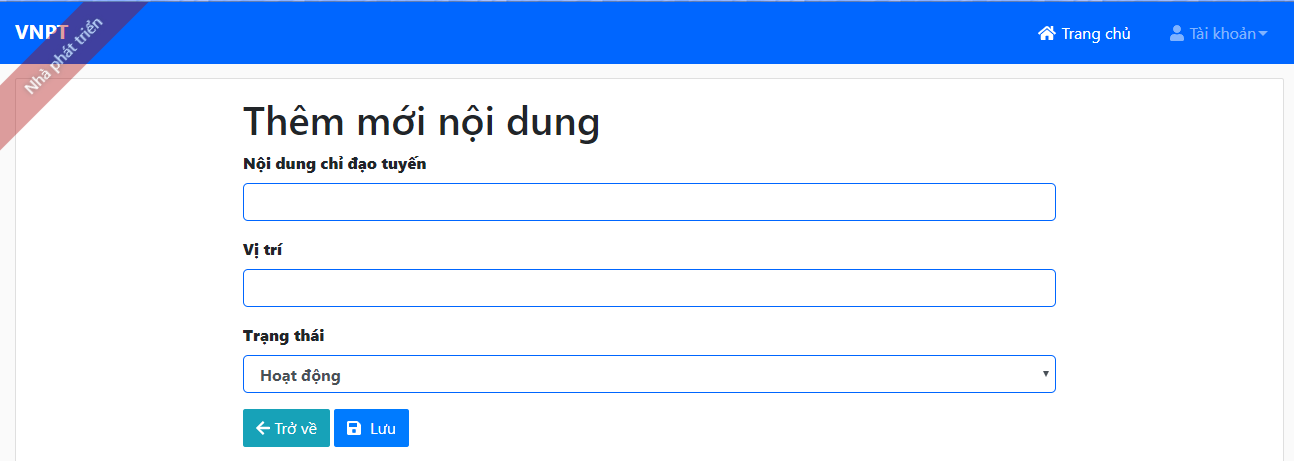
###### Hình 3.26. Giao diện chính

**❖ Giao diện Quản lý nội dung chỉ đạo tuyến**



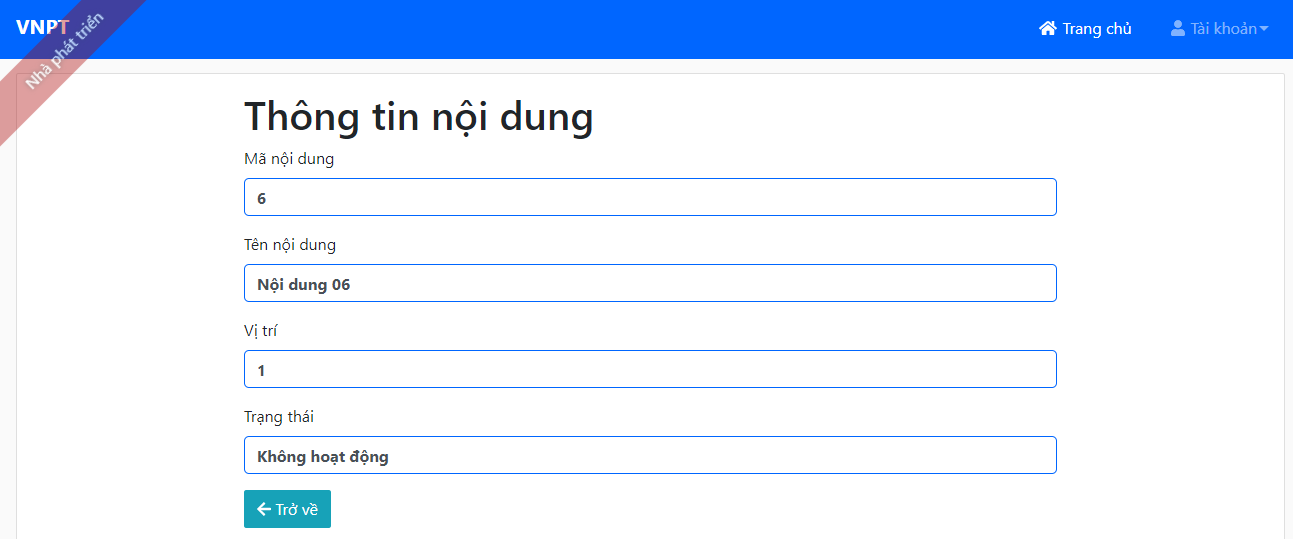
###### Hình 3.27. Giao diện Quản lý nội dung chỉ đạo tuyến

**❖ Giao diện Thêm mới nội dung chỉ đạo tuyến**



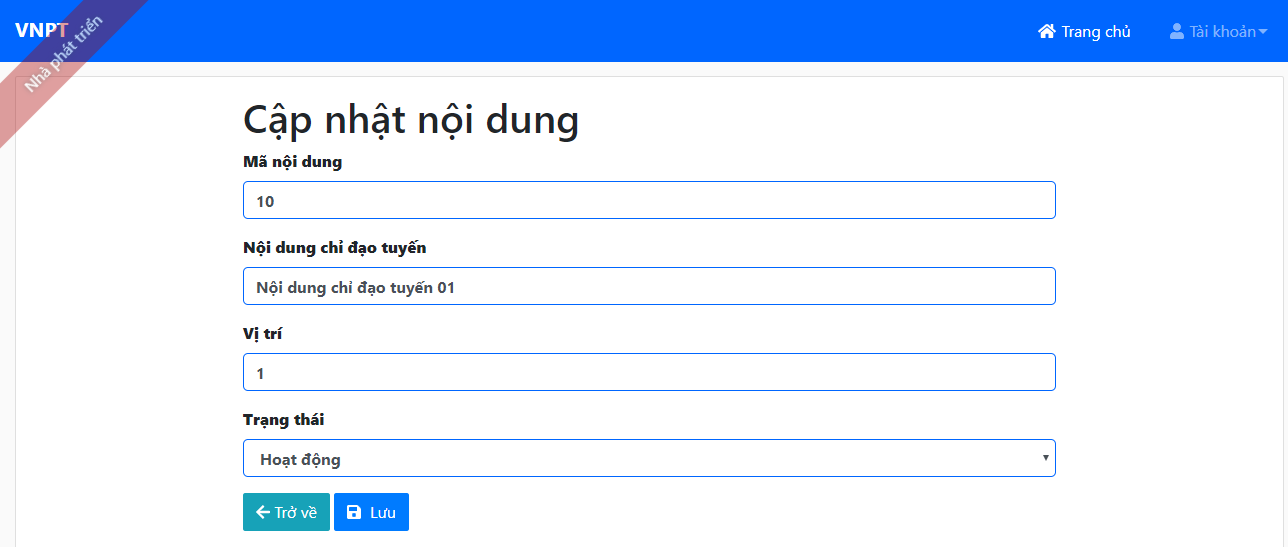
###### Hình 3.28. Giao diện thêm mới nội dung chỉ đạo tuyến

**❖ Giao diện Xem thông tin nội dung chỉ đạo tuyến**



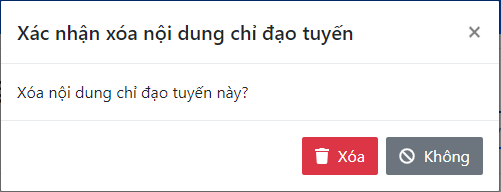
###### Hình 3.29. Giao diện Xem thông tin nội dung chỉ đạo tuyến

**❖ Giao diện Cập nhật nội dung chỉ đạo tuyến**



###### Hình 3.30. Giao diện Cập nhật thông tin nội dung chỉ đạo tuyến

**❖ Giao diện Xóa nội dung chỉ đạo tuyến**



###### Hình 3.31. Giao diện Xóa nội dung chỉ đạo tuyến

**❖ Giao diện Thống kê chỉ đạo tuyến**



###### Hình 3.32. Giao diện Thống kê chỉ đạo tuyến

**❖ Giao diện Xuất báo cáo chỉ đạo tuyến**



###### Hình 3.33. Giao diện Xuất báo cáo chỉ đạo tuyến

# Chương 4: KẾT QUẢ VÀ HƯỚNG PHÁT TRIỂN

## **4.1. Kết quả đạt được**

Kết quả trong tìm hiểu:

* Đã hiểu rõ mô hình Microservice, củng cố được kiến thức và nâng cao khả năng lập trình.
* Có thêm nhiều kiến thức về ReactJS, Redux, TypeScript, JasperReports,…. Biết cách sử dụng nhiều thư viện.
* Học tập được tính phân tích, thiết kế hệ thống theo yêu cầu.
* Biết được quy trình chỉ đạo tuyến, quy trình đăng ký, xét duyệt đề tài trong ngành y tế.

Kết quả trong thực nghiệm:

* Xây dựng được phần mềm thông qua quá trình nghiên cứu, khảo sát, phân tích thiết kế hệ thống, phân tích thiết kế cơ sở dữ liệu, đặc tả và thiết kế giao diện.
* Thực hiện thành công chức năng cơ bản đã đặt ra.
* Được chạy thử nghiệm, kiểm tra.

## **4.2. Hạn chế**

Cấu hình hệ thống khá phức tạp.

Giao diện chưa được bắt mắt.

Chưa tận dụng được hết tối ưu của kiến trúc Microservice

## **4.3. Hướng phát triển**

Xây dựng hệ thống tương thích với nhiều thiết bị sử dụng trình duyệt khác nhau.

# DANH MỤC TÀI LIỆU THAM KHẢO

[1] JasperReports, <https://community.jaspersoft.com/>.

[2] JHipster, https://www.jhipster.tech/.

[3] Microservice, <https://microservices.io/>.

[4] Redux, <https://redux.js.org/>.

[5] TypeScript, <https://www.typescriptlang.org/>.