

TRƯỜNG ĐẠI HỌC TIỀN GIANG

**KHOA CÔNG NGHỆ THÔNG TIN**

-----🙞🙜🕮🙞🙜-----

****

**ĐỒ ÁN NGÀNH**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

**ĐỀ TÀI**

**XÂY DỰNG ỨNG DỤNG LẤY Ý KIẾN**

**CÁC BÊN LIÊN QUAN VỀ CTĐT CỦA TRƯỜNG ĐẠI HỌC TIỀN GIANG**

Người hướng dẫn: **Nguyễn Văn Thuận**

Sinh viên thực hiện:

Đặng Hữu Lộc 018101027

**LỜI NÓI ĐẦU**

**LỜI CẢM ƠN**

🙞🙞🕮🙜🙜

Đầu tiên tôi 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 chúng tôi những kiến thức quý báu để tôi có thể hoàn thành tốt đồ án này.

Mặc dù đã rất cố gắng trong quá trình làm đồ án, nhưng không thể tránh khỏi những thiếu sót. Rất mong nhận được sự thông cảm và góp ý tận tình của quý Thầy Cô và các bạn.

Cuối cùng chúng tôi kính chúc quý Thầy/Cô dồi dào sức khỏe và thành công trong công việc.

Xin chân thành cảm ơn!

Tiền Giang, ngày…tháng…năm…

Sinh viên thực hiện

Đặng Hữu Lộc

**MỤC LỤC**

[CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI 7](#_Toc57750752)

[1.1. Giới thiệu đề tài 7](#_Toc57750753)

[1.1.1. Đặc tả đề tài 7](#_Toc57750754)

[1.1.2. Mục tiêu đề tài 7](#_Toc57750755)

[1.1.3. Nhiệm vụ 8](#_Toc57750756)

[1.2. Các vấn đề tập trung giải quyết 8](#_Toc57750757)

[Chương 2: CƠ SỞ LÝ THUYẾT VÀ THỰC TIỄN 9](#_Toc57750758)

[2.1. Cơ sở lý thuyết 9](#_Toc57750759)

[2.1.1. Giả thuyết nghiên cứu 9](#_Toc57750760)

[2.1.2. Phương pháp nghiên cứu 9](#_Toc57750761)

[2.1.3. Công nghệ sử dụng 10](#_Toc57750762)

[2.1.4. Cơ sở lý thuyết 10](#_Toc57750763)

[2.2. Cơ sở thực tiễn và ứng dụng của khóa luận 18](#_Toc57750764)

[2.2.1. Bối cảnh thực tiễn 18](#_Toc57750765)

[2.2.2. Ứng dụng 18](#_Toc57750766)

[CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ 18](#_Toc57750767)

[3.1. Phân tích yêu cầu 18](#_Toc57750768)

[3.1.1. Phân tích yâu cầu chức năng 18](#_Toc57750769)

[3.1.2. Phân tích yêu cầu phi chức năng 18](#_Toc57750770)

[3.1.3. Phân tích quy trình 18](#_Toc57750771)

[3.2. Phân tích thiết kế hệ thống 18](#_Toc57750772)

[TÀI LIỆU THAM KHẢO 19](#_Toc57750773)

**DANH MỤC HÌNH**

[Hình 1. Kiến thúc hệ thống quản lý khảo sát 7](#_Toc57752978)

[Hình 2. Mô hình ứng dụng ví dụ 15](#_Toc57752979)

[Hình 2. Định nghĩa các ứng dụng và thực thể trong ví dụ với JDL 16](#_Toc57752980)

[Hình 3. Khởi tạo dự án thành công 17](#_Toc57752981)

[Hình 4. Các thư mục tương ứng với các ứng dụng đã định nghĩa trong ví dụ trên 17](#_Toc57752982)

[Hình 5. Usecase tổng quát ứng dụng 19](#_Toc57752983)

**DANH MỤC BẢNG**

[Bảng 1 Usecase đăng ký 19](#_Toc57752984)

[Bảng 2 Usecase đăng nhập 20](#_Toc57752985)

[Bảng 3 Usecase thay đổi mật khẩu 20](#_Toc57752986)

[Bảng 4 Usecase thêm tài khoản người dùng 21](#_Toc57752987)

[Bảng 5 Usecase thay đổi thông tin cá nhân 21](#_Toc57752988)

[Bảng 6 Usecase xem thông tin chi tiết môn học 22](#_Toc57752989)

[Bảng 7 Usecase thêm môn học 22](#_Toc57752990)

[Bảng 8 Usecase cập nhật thông tin môn học 23](#_Toc57752991)

[Bảng 9 Usecase xem thông tin chi tiết CTĐT 23](#_Toc57752992)

[Bảng 10 Usecase thêm CTĐT 23](#_Toc57752993)

[Bảng 11 Cập nhật CTĐT 24](#_Toc57752994)

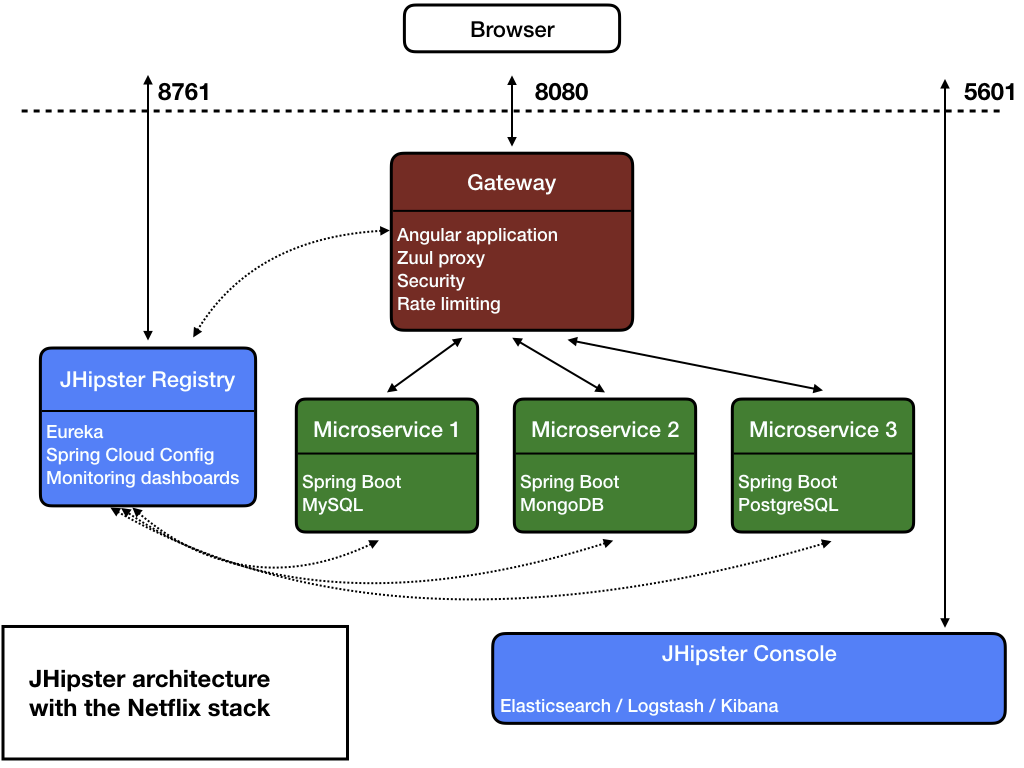
[Bảng 12 Tạo phiếu khảo sát chuẩn đầu ra của CTĐT 25](#_Toc57752995)

# CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI

## **1.1. Giới thiệu đề tài**

### 1.1.1. **Đặc tả đề tài**

Để đáp bắt kịp với sự tiến bộ của các công nghệ mới và đáp ứng nhu cầu sử dụng của người dùng. Hệ thống quản lý khảo sát sẽ kết hợp những công nghệ mới theo kiến trúc như sau:



Hình 1. Kiến thúc hệ thống quản lý khảo sát

### Mục tiêu đề tài

Tìm hiểu lý thuyết:

* Tìm hiểu về Microservice (Jhipster Framework).
* Tìm hiểu về các công nghệ Frontend (Angular X, TypeScript).
* Tìm hiểu về các công nghệ Backend (Java 11, Spring Framework, Jasper Report).
* Tìm hiểu Docker
* Nắm được cấu trúc CTĐT, cấu trúc phiếu khảo sát chuẩn đầu ra cấp 3 theo CDIO.

Ứng dụng thực nghiệm:

* Xây dựng được hệ thống quản lý khảo sát: Quản lý thông tin phiếu khảo sát theo CTĐT,quản lý thông tin các bên liên quan (giảng viên, sinh viên, cựu sinh viên, doanh nghiệp), thông kê kết quả khảo sát.
* Quản lý, lưu trữ được các thông tin các thông tin các CTĐT theo năm học.
* Quản lý, lưu trữ được thông tin các môn học.
* Liên kết giao diện, thông tin với service Quản lý khảo sát được chính xác, nhanh chóng và hiệu quả.
* Ứng dụng có đáp ứng đủ cho việc lấy ý kiến các bên liên quan về CTĐT và có khả năng áp dụng vào thực tế.
* Ứng dụng hoạt động hiệu quả, nhanh chóng.
* Giao diện trực quan và dễ sử dụng.

### Nhiệm vụ

Tìm hiểu lý thuyết: Tìm hiểu nguyên lý hoạt động của Microservices (Jhipster), nắm được các kiến thức, định nghĩa cốt lõi của Jhipster Framework, Angular, TypeScript, HighChart, JasperReposts và MySql …

Ứng dụng 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, cài đặt giải thuật, hoàn thiện các chức năng đã liệt kê.

## **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, kết nối truyền dữ liệu một cách chính xác tới các service liên quan.
* Giao diện đơn giản, dễ sử dụng : có bố cục hợp lí, các menu chức năng rõ ràng.
* Ứng dụng có thể chạy được trên các trình duyệt web trên máy tính như (Google Chrome, IE,…)

# 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 cho quá trình quản lý, thống kê thông tin ý kiến các bên liên quan về CTĐT được chính xác, dễ dàng và hiệu quả 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:

* Chuẩn đầu ra của DH CNTT khóa 2017 và của DH CNTT khóa 2020.
* Mẫu phiếu khảo sát “cựu sinh viên về chuẩn đầu ra của ngành công nghệ thực phẩm”
* Mẫu phiếu khảo sát “doanh nghiệp về chuẩn đầu ra của ngành công nghệ thực phẩm”
* Mẫu phiếu khảo sát “sinh viên về chuẩn đầu ra của ngành công nghệ thực phẩm”
* Mẫu phiếu khảo sát “giảng viên về chuẩn đầu ra của ngành công nghệ thực phẩm”

Phương pháp quan sát khoa học: Quan sát khách quan các vấn đề liên quan đến quy trình quản lý thông tin khảo sát, tài liệu chuẩn đầu ra của CTĐT.

Phương pháp thực nghiệm: Áp dụng lấy dữ liệu từ hệ thống thực tế.

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 tài liệu do nhà trường cung cấp và các tài liệu liên quan.
* 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, 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 sửa 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 để 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 quá trình quản lý khảo sát ý kiến các bên liên quan về CTĐT của nhà trường.

### 2.1.3. Công nghệ sử dụng

* Hệ điều hành: Windows 10
* Ngôn ngữ lập trình chính: TypeScript, Java 11
* Hệ quản trị cở sở dữ liệu: MySQL
* Môi trường lập trình: Trình soạn thảo mã nguồn: Visual Studio Code, IntelliJ IDEA 2020, Docker
* Trình duyệt web: Google Chrome, Microsoft Edge, Mozila Firefox.

### 2.1.4. Cơ sở lý thuyết

1. Microservices

* 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, ứng dụng web đã có thể chia nhỏ hơn cho từng nhóm đối tượng người dùng. 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. Điều này giúp hạn chế sự dư thừa dữ liệu và phát huy được vai trò riêng biệt của từng dịch vụ.

Ưu điểm của kiến trúc Microservices:

* Cho phép phân phối và triển khai một các liên tục các ứng dụng lớn và phức tạp.
* Cải thiện khả năng bảo trì
* Cải thiện khả năng kiểm thử (các service nhỏ nên việc kiểm thử dễ dàng và nhanh chóng hơn)
* Khả năng triển khai tốt hơn (các service có khả năng triển khai độc lập).
* Kiến trúc Microservices cho phép tổ chức các nhóm tự chủ cho một hoặc một nhóm service. Các nhóm tự chủ này có thể phát triển, thử nghiệm, triển khai, mở rộng quy mô một các độc lập với các nhóm khác.
* Mỗi service tương đối nhỏ.
* Lập trình viên dễ hiểu hơn.
* Các công cụ mà lập trình viên sử dụng cũng hoạt động hiệu quả hơn.
* Các ứng dụng khởi động nhanh hơn và giúp lập trình viên hoạt động hiệu quả hơn.
* Có thể cách ly được lỗi.
* Loại bỏ khả năng phụ thuộc công nghệ nào đó một cách lâu dài. Khi phát triển một service mới chúng ta có thể chọn công nghệ mới cho service mới hoặc có thể lại service cũ bằng một công nghệ khác mà ít ảnh hưởng tới các service khác.

Nhược điểm của Microservices:

* Việc xử lý các request trên nhiều service trở nên khó khăn hơn, phải phối hợp cẩn thận với nhiều nhóm nếu mỗi nhóm đảm nhiệm một Service khác nhau.
* Khó khăn trong việc kiểm thử vì phải tương tác với nhiều service.
* Tốn nhiều bộ nhớ hơn (do chia ra nhiều Service con).
* Việc triển khai cũng trở nên phức tạp khi phải giám sát một số lượng lớn các service.
* Tuy có những khuyết điểm nhưng đó là những hạn chế có thể tránh khỏi nếu lập trình viên làm việc có kỉ luật, đã quen với hệ thống và hoàn toàn không liên quan đến những hạn chế về yếu tố kĩ thuật (không thể thay đổi trong thời gian ngắn được).

1. Angular X, Typescript, JasperReports

Giới thiệu về Angular X:

Angular là một framework, một ngôn ngữ chức năng hoạt động trên tất cả các trình duyệt hiện đại nhất ngày nay

Angular có cấu trúc dễ hiểu và dễ áp dụng phổ biến, tương thích mạnh mẽ với MVC Core (MVC là viết tắt của Model View Controller - một mẫu cấu trúc thiết kế rất hiệu quả cho việc mở rộng ứng dụng). Thông thường những framework khác yêu cầu lập trình viên dành rất nhiều công sức để chia ứng dụng thành những phần MVC nhỏ, sau đó cần phải viết code để nối lại các thành phần đó. Tuy nhiên, khi dử dụng Angular, công việc của lập trình viên sẽ đơn giản hơn rất nhiều do chỉ cần tạo thành phần và Angular sẽ hoàn thành cho bạn tất cả những phần còn lại bao gồm kết nối và quản lý. Không chỉ vậy, framework này sẽ mang lại các dòng code “sạch”, rất dễ đọc và dễ để sử dụng phổ biến.

Angular hoạt động dựa chủ yếu vào mã HTML để xây dựng và phát triển web, giúp lập trình viên bỏ qua thủ tục viết JavaScript cho các phần từ giao diện, khiến cho việc xây dựng web dễ dàng và nhẹ nhàng hơn rất nhiều. Nhờ vào sử dụng HTML, lập trình viên sẽ mất ít thời gian hơn trong việc định hình cấu trúc thẩm mỹ của giao diện và thao tác xử lý, hơn nữa còn có thể giúp truy cập và sửa code trực tiếp trên file HTML mà không cần cài đặt thêm bất kì phần mềm nào.

Angular đã cập nhật và khắc phục được nhiều những lỗ hổng kĩ thuật mà những nhà phát triển trên toàn thế giới phải đối mặt trong quá trình xây dựng ứng dụng có tính năng mạnh mẽ, mở rộng dựa trên trình duyệt suốt nhiều năm. Framewrok này cung cấp cho người dùng những tính năng phát triển cô cùng tuyệt vời như tính năng tự động kết nối trình duyệt theo giao diện người dùng, các bộ lọc, bản mẫu động hay là các chỉ thị.

Nhờ vào tính năng mã nguồn mở, Angular đã được lập trình viên khắp thế giới quan tâm và ưa chuộng sử dụng. Chính vì vậy, lực lượng hỗ trợ cho Angular cũng vô cùng đông đảo. Kết hợp với việc Google đang hỗ trợ rất mạnh cho framework này đã khiến cho chất lượng mã nguồn của Angular rất cao và bền vững.

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.

Giới thiệu JasperReports:

* JasperReports là một công cụ tạo báo cáo mã nguồn mở phổ biến 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,…, hỗ trợ các nhà phát triển với nhiệm vụ bổ sung khả năng báo cáo cho các ứng dụng Java Application.
* 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.

1. Giới thiệu về Spring

Giới thiệu về Spring trong Java:

* Spring là một Framework phát triển các ứng dụng. Nó giúp tạo các ứng dụng có hiệu năng cao, dễ kiểm thử, sử dụng lại code…
* Spring là một mã nguồn mở, được phát triển, chia sẻ và có cộng đồng người dùng rất lớn.
* Spring được chia làm nhiều module khác nhau, tùy theo mục đích phát triển ứng dụng mà ta dùng 1 trong các module đó.
* Trong số các module của Spring, chúng tôi sẽ sử dụng module Spring Boot để xây dựng kiến trúc ứng dụng

1. Giới thiệu về Spring Boot:

* Spring Boot là một module của Spring Framework, cung cấp tính năng RAD (Rapid Application Development) – Phát triển ứng dụng nhanh.
* Spring Boot được dùng để tạo các ứng dụng độc lập dựa trên Spring.
* Spring Boot không yêu cầu cấu hình XML
* Nó là một chuẩn cho cấu hình thiết kế phần mềm, tăng cao năng suất cho developer.
* Ưu điểm của SpringBoot:

+ Có các tính năng của Spring Framework.

+ Tạo ứng dụng độc lập, có thể chạy bằng java -jar (cho cả java web)

+ Nhúng trực tiếp các ứng dụng server (Tomcat, Jetty…) do đó không cần phải triển khai file WAR

+ Cấu hình ít, tự động cậu hình bất kì khi nào có thể (Giảm thời gian viết code, tăng năng suất)

+ Không yêu cầu XML config…

+ Cung cấp nhiều plugin

+ Chuẩn cho Microservices (Cloud support; giảm việc setup, config; các thư viện hỗ trợ…)

1. Giới thiệu Spring Data JPA

* Là một phần của Spring Data, nó làm cho việc truy cập dữ liệu của các ứng dụng dựa trên Spring một các dễ dàng.
* Các tính năng của Spring Data JPA
  + Hỗ trợ xây dựng kho lưu trữ dựa trên Spring và JPA
  + Kiểm tra minh bạch của lớp miền.
  + Hỗ trợ phân trang, thực thi truy vấn động, khả năng tích hợp mã truy cập dữ liệu tùy chỉnh.
  + Xác thực các câu truy vấn có chú thích @Query tại thời gian khởi động.
  + Hỗ trợ ánh xạ thực thể dựa trên XML
  + Cấu hình kho lưu trữ dựa trên JavaConfig

1. Sử dụng JHipster để tạo nhanh mô hình Microservice

Giới thiệu về JHipster:

* JHipster là một framework được phát triển bởi Yeoman và được sử dụng để phát triển các dự án Spring Boot + Angular/ReactJS/Vue.
* Jhipster sẽ giúp chúng ta dễ dàng có được một project đủ mạnh mẽ đầy đủ những thứ cơ bản để bắt đầu với thời gian nhanh nhất.

Cài đặt thư viện cho Jhipster:

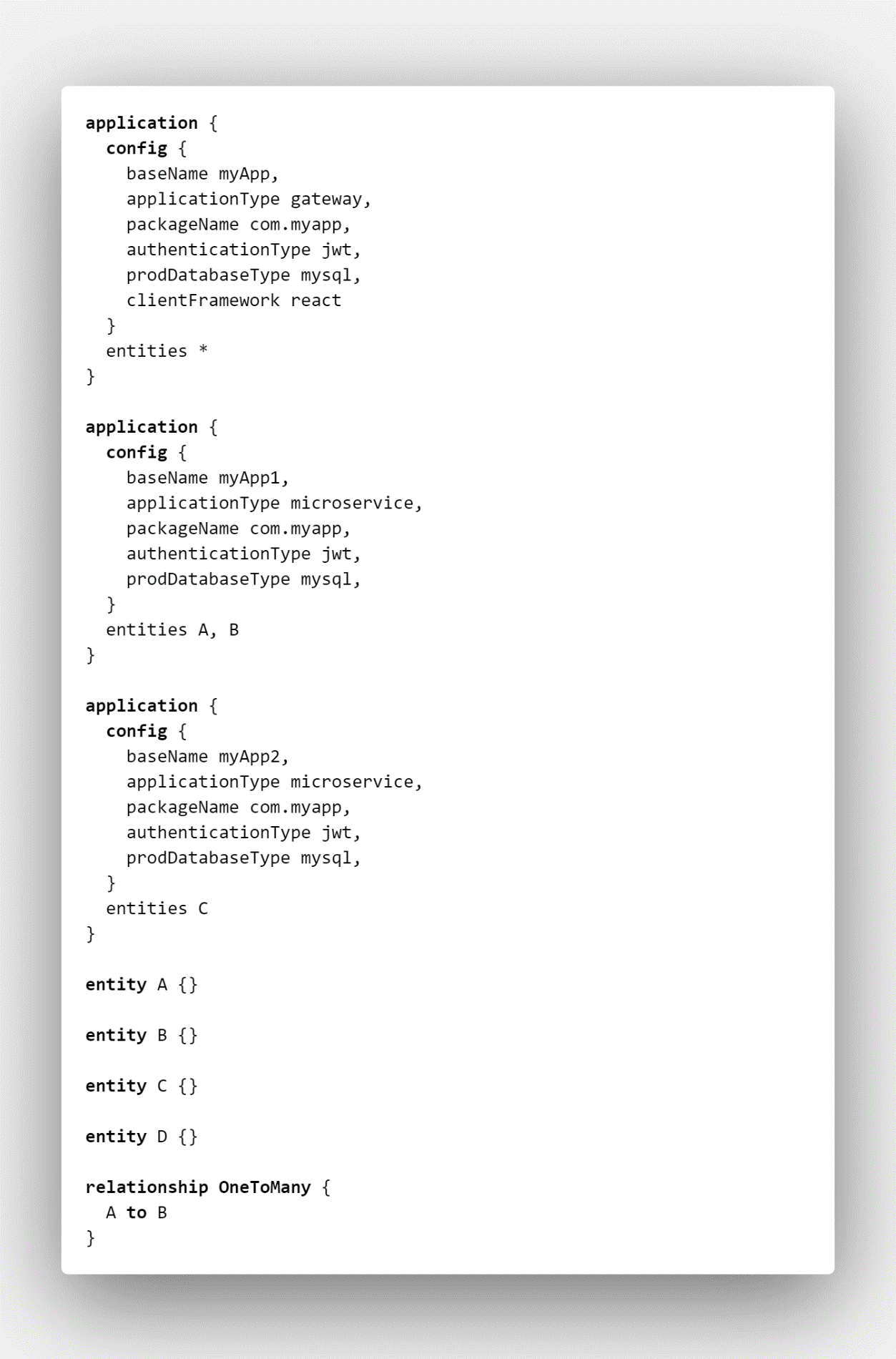
* Cài đặt và cấu hình Java 11 trên máy tính
* Cài đặt Node-js từ trang chủ
* Cài đặt Yeoman sử dụng lệnh: npm install -g yo
* Cài Jhipster Generator sử dụng lệnh: npm install -g generator -jhipster

Khởi tạo dự án microservice:

* Ta định nghĩa mô hình ứng dụng với các ứng dụng, thực thể và mối quan hệ giữa các thực thể với JDL – Jhipster Domain Language.

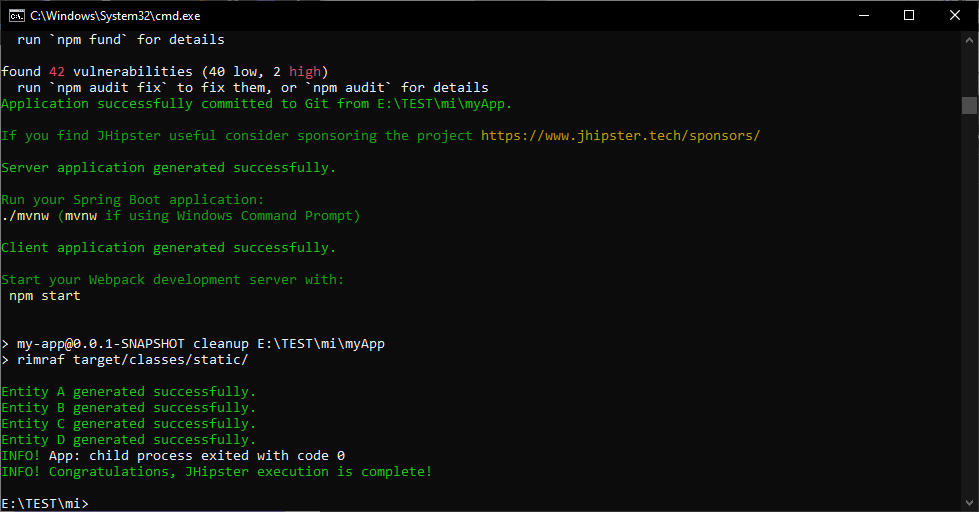
Ví dụ sau đây, ta sẽ tạo một ứng dụng web microservice với 1 gateway và 2 microservice với 4 thực thể A, B, C, D. Gateway quản lý hiển thị thông tin tất cả các thực thể, microservice1 sẽ quản lý thực thể A và B, microservice 2 sẽ quản lý thực thể C, 2 thực thể A vả B có mối quan hệ 1 nhiều.

Hình 2. Mô hình ứng dụng ví dụ

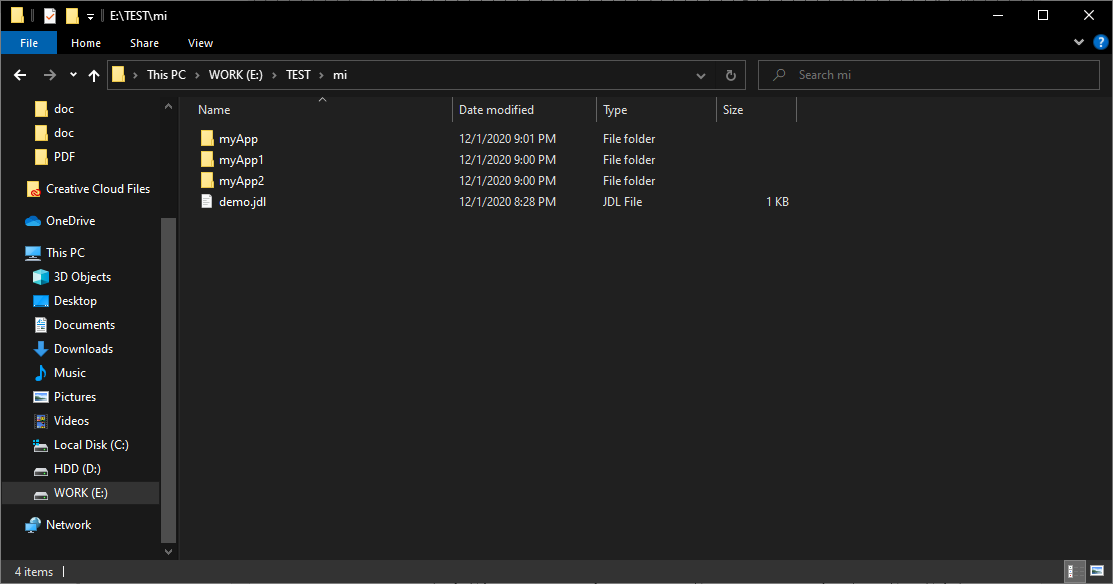


Hình 2. Định nghĩa các ứng dụng và thực thể trong ví dụ với JDL

* Sau đó ta tạo một thư mục chứa dự án và sao chép file JDL trên vào thư mục và chạy câu lệnh: “jhipster import-jdl <tên file jdl>.jdl” và nhấn enter.



Hình 3. Khởi tạo dự án thành công

* Sau khi khởi tạo dự án thành công, ta sẽ thấy 3 thư mục tương ứng với 3 ứng dụng đã được định nghĩa trong file JDL.

Hình 4. Các thư mục tương ứng với các ứng dụng đã định nghĩa trong ví dụ trên

* Bây giờ thì ta đã có một dự án microservice với:
* Backend: Spring Boot + Spring Security + Spring Data JPA
* Database: MySQL
* Frontend: Bootstrap + SASS + React (hoặc thấp hơn tùy vào phiên bản JHipster người dùng cài đặt ở trên).
* Chạy ứng dụng Spring Boot bằng cách nhập: ./mvnw

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

### Bối cảnh thực tiễn

Trường Đại học Tiền Giang có rất nhiều CTĐT và cần lấy ý kiến các bên liên quan về CTĐT.

Nhu cầu thực tế:

* Xây dựng phần mềm tin học lưu trữ thông tin các bên liên quan, thông tin góp ý về CTĐT
* Có thể sử dụng trên các thiết bị kết nối mạng
* Đảm bảo hoạt động ổn định

### Ứng dụng

* Sản phẩm của phần mềm này có thể dùng để thu thập ý kiến của các bên liên quan về CTĐT.

# CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ

## **3.1. Phân tích yêu cầu**

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

* Giao diện: đơn giản, dễ sử dụng, phù hợp với các nhân viên, bác sĩ của bệnh viện.
* Bảo mật: Không bị khai thác lỗi.
* Có thể hoạt động trên các trình duyệt cơ bản trên máy tính.
* Chạy nhanh, ổn định.

### 3.1.2. Phân tích yêu cầu chức năng

a. Các chức năng của hệ thống

* Cán bộ quản lý hệ thống:
  + Đăng nhập hệ thống
  + Quản lý thông tin tài khoản
    - Thêm tài khoản
    - Cập nhật thông tin tài khoản
    - Xóa tài khoản
    - Kích hoạt/khóa tài khoản
  + Quản lý thông tin về học phần
    - Thêm học phần
    - Cập nhật thông tin học phần
    - Xóa học phần
  + Quản lý thông tin về CTĐT
    - Thêm CTĐT
    - Cập nhật CTĐT
    - Xóa CTĐT
  + Quản lý thông tin khảo sát
    - Tạo phiếu khảo sát chuẩn đầu ra của một CTĐT
    - Tạo cuộc khảo sát (cho phép các bên liên quan tiến hành làm khảo sát)
    - Thống kê dữ liệu khảo sát ( hiển thị ra sơ đồ, xuất ra báo cáo)
* Bên liên quan (giảng viên, sinh viên, cựu sinh viên, doanh nghiệp)
  + Đăng ký tài khoản
  + Thay đổi mật khẩu
  + Cập nhật thông tin cá nhân
  + Xem thông tin chi tiết môn học
  + Xem thông tin chi tiết của CTĐT
  + Làm khảo sát

b. Các chức năng sẽ thực hiện trong đồ án này

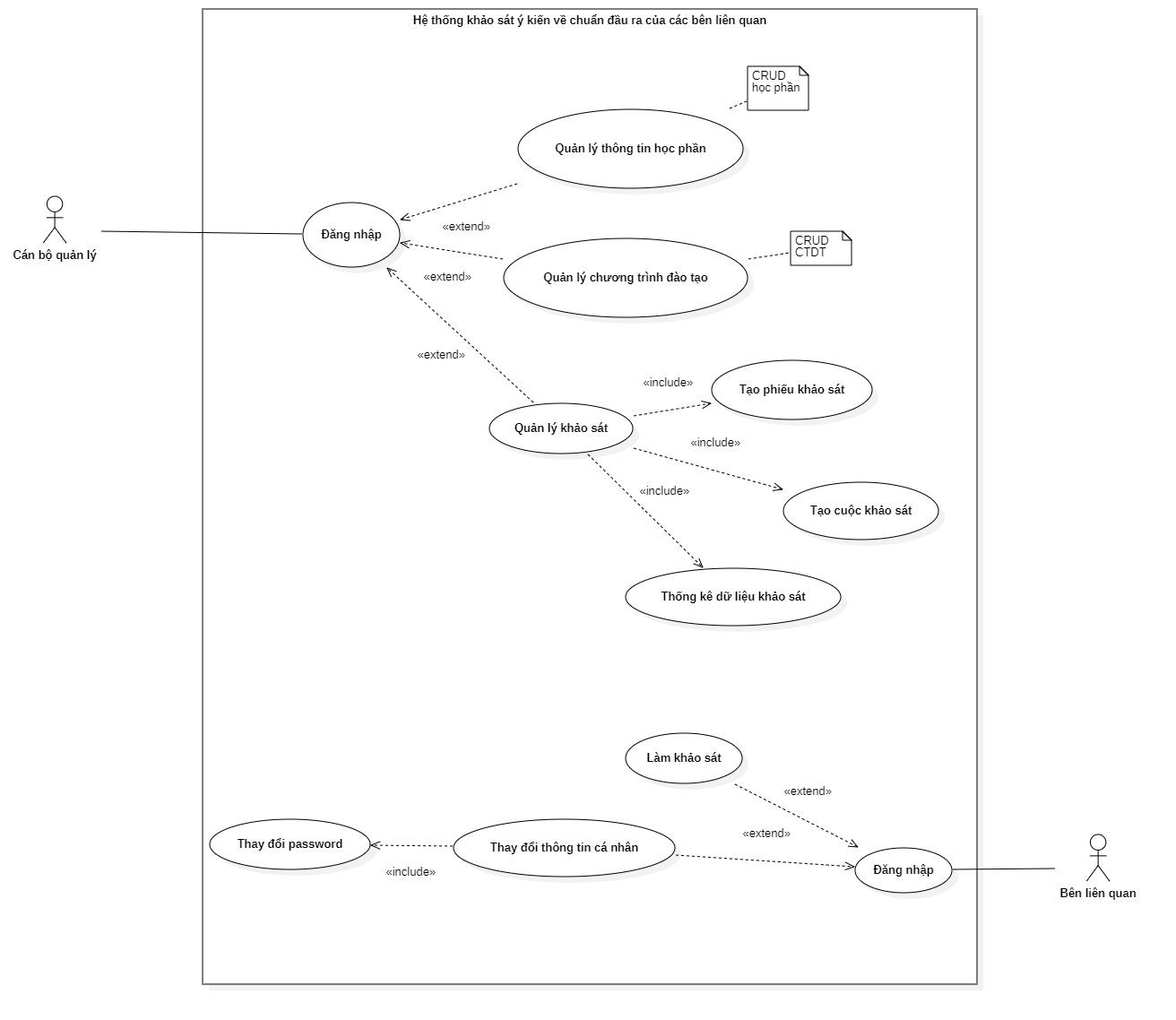
1. Đăng ký
2. Đăng nhập
3. Thay đổi mật khẩu
4. Quản lý tài khoản người dùng
5. Thay đổi thông tin cá nhân
6. Xem thông tin chi tiết môn học
7. Thêm môn học
8. Cập nhật thông tin môn học
9. Xem thông tin chi tiết CTĐT
10. Thêm CTĐT
11. Cập nhật CTĐT
12. Tạo phiếu khảo sát chuẩn đầu ra của CTĐT

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

### 3.2.1. Hệ thống lưu trữ thông tin người dùng

### 3.2.2. Hệ thống lưu trữ thông tin khảo sát

a. Usecase tổng quát của ứng dụng



Hình 5. Usecase tổng quát ứng dụng

b. Các chức năng sẽ thực hiện trong đồ án này

1. Đăng ký
2. Đăng nhập
3. Thay đổi mật khẩu
4. Quản lý tài khoản người dùng
5. Thay đổi thông tin cá nhân
6. Xem thông tin chi tiết môn học
7. Thêm môn học
8. Cập nhật thông tin môn học
9. Xem thông tin chi tiết CTĐT
10. Thêm CTĐT
11. Cập nhật CTĐT
12. Tạo phiếu khảo sát chuẩn đầu ra của CTĐT

c. Kịch bản cho các chức năng đã chọn

#### Bảng Usecase đăng ký

|  |  |
| --- | --- |
| Tên | Đăng ký |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Không có |
| Hậu điều kiện | Đăng ký thành công vào hệ thống |
| Điều kiện tối thiểu | Yêu cầu nhập lại mật khẩu |
| Sự kiện chính | 1. Người dùng nhấn vào nút “tài khoản → đăng ký”. 2. Người dùng nhập thông tin tài khoản (tên đăng nhập, email, mật khẩu, xác nhận lại mật khẩu). 3. Nếu mật khẩu có đủ độ mạnh (có ít nhất 4 kí tự, có số) thì sẽ hiện thông báo đăng ký thành công. 4. Người dùng chờ người quản lý “kích hoạt” tài khoản để đăng nhập |
| Ngoại lệ | 3.1 Nếu mật khẩu không đủ độ mạnh thì không cho phép nhấn vào nút đăng ký. |

#### Bảng Usecase đăng nhập

|  |  |
| --- | --- |
| Tên | Đăng nhập |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Tài khoản người dùng đã tồn tại trong hệ thống |
| Hậu điều kiện | Đăng nhập thành công vào hệ thống |
| Điều kiện tối thiểu | Yêu cầu đăng nhập lại |
| Sự kiện chính | 1. Hệ thống hiển thị màn hình đăng nhập 2. Người dùng nhập thông tin tài khoản và mật khẩu 3. 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 nhập lại thông tin tài khoản, mật khẩu. |

#### Bảng Usecase thay đổi mật khẩu

|  |  |
| --- | --- |
| Tên | Thay đổi mật khẩu |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Có tài khoản và đã đăng nhập |
| Hậu điều kiện | Thay đổi mật khẩu thành công |
| Điều kiện tối thiểu | Nhập đúng email |
| Sự kiện chính | 1. Khi người dùng đã đăng nhập nhấn vào nút “tài khoản → mật khẩu”, nếu chưa đăng nhập thì nhấn “quên mật khẩu”. 2. Người dùng nhập nhập khẩu hiện tại, mật khẩu mới và xác nhận lại mật khẩu mới. 3. Hệ thống gửi thông báo về email của người dùng. 4. Hệ thống cập nhật thay đổi mật khẩu của người dùng. |
| Ngoại lệ | 2.1. Người dùng nhập sai mật khẩu hiện tại.  2.2. Người dùng nhập mật khẩu mới không đủ độ mạnh.  2.3. Người dùng nhập xác thực mật khẩu mới không trùng với mật khẩu mới. |

#### Bảng Usecase thêm tài khoản người dùng

|  |  |
| --- | --- |
| Tên | Thêm tài khoản người dùng |
| Tác nhân chính | Cán bộ quản lý |
| Tiền điều kiện | Tài khoản quản lý có quyền admin |
| Hậu điều kiện | Thêm thành công tài khoản người dùng |
| Điều kiện tối thiểu | Người quản lý nhập đầy đủ thông tin của tài khoản người dùng. |
| Sự kiện chính | 1. Hệ thống hiển thị giao diện tạo quản lý tài khoản. 2. Người quản lý nhấn nút “tạo tài khoản mới” trong giao diện hiện hành để thêm tài khoản mới. 3. Người quản lý cần nhập vào tên đăng nhập, tên, họ, email, tình trạng kích hoạt, các quyền (admin, user) và bấm nút lưu. 4. Hệ thống sẽ lưu thông tin tài khoản. 5. Hệ thống gửi password về email của người dùng. |
| Ngoại lệ | * 1. Người quản lý tạo tài khoản có tên đăng nhập trùng với tài khoản hiện có.   2. Người quản lý nhập email cho tài khoản mới trùng với email của tài khoản hiện có.   3. Người quản lý nhập không đầy đủ các thông tin. |

#### Bảng Usecase thay đổi thông tin cá nhân

|  |  |
| --- | --- |
| Tên | Thay đổi thông tin cá nhân |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Tài khoản người dùng đã tồn tại trong hệ thống |
| Hậu điều kiện | Cập nhật thành công thông tin cá nhân |
| Điều kiện tối thiểu | Yêu cầu nhập lại đầy đủ thông tin cá nhân |
| Sự kiện chính | 1. Hiển thị giao diện thay đổi thông tin cá nhân. 2. Người dùng nhập các thông tin cá nhân 3. Hệ thống kiểm tra 4. Hệ thống lưu lại kết quả thay đổi tài khoản |
| Ngoại lệ | 3.1. Người dùng thay đổi địa chị email trùng với tài khoản khác.  3.2. Người dùng nhập thiếu thông tin |

#### Bảng Usecase xem thông tin chi tiết môn học

|  |  |
| --- | --- |
| Tên | Xem thông tin chi tiết môn học |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Tài khoản người dùng đã tồn tại trong hệ thống  Thông tin môn học đã tồn tại trong hệ thống |
| Hậu điều kiện | Hiển thị thông tin chi tiết môn học |
| Điều kiện tối thiểu | Yêu cầu nhập lại tên môn học cần tìm kiếm |
| Sự kiện chính | 1. Hiển thị giao diện danh sách tất cả các môn học hoặc giao diện các môn học trong CTĐT. 2. Người dùng nhấn vào môn học cần xem thông tin chi tiết. 3. Hệ thống hiển thị thông tin chi tiết môn học |
| Ngoại lệ | Không tìm thấy thông tin chi tiết môn học. |

#### Bảng Usecase thêm môn học

|  |  |
| --- | --- |
| Tên | Thêm môn học |
| Tác nhân chính | Cán bộ quản lý |
| Tiền điều kiện | Thông tin môn học chưa tồn tại trong hệ thống |
| Hậu điều kiện | Thêm thông tin chi tiết môn học thành công |
| Điều kiện tối thiểu | Yêu cầu nhập đầy đủ thông tin môn học |
| Sự kiện chính | 1. Hệ thống hiển thị giao diện thêm môn học 2. Người quản lý nhập thông tin chi tiết môn học 3. Nhấn nút “Lưu” 4. Hệ thống kiểm tra 5. Hệ thống lưu lại thông tin môn học |
| Ngoại lệ | * 1. Hệ thống lưu thất bại, mã môn học bị trùng với môn học khác.   2. Người quản lý không nhập đầy đủ các thông tin chi tiết môn học |

#### Bảng Usecase cập nhật thông tin môn học

|  |  |
| --- | --- |
| Tên | Cập nhật thông tin môn học |
| Tác nhân chính | Cán bộ quản lý |
| Tiền điều kiện | Thông tin môn học đã tồn tại trong hệ thống |
| Hậu điều kiện | Cập nhật thông tin chi tiết môn học thành công |
| Điều kiện tối thiểu | Yêu cầu nhập đầy đủ thông tin môn học |
| Sự kiện chính | 1. Hệ thống hiển thị giao diện cập nhật môn học 2. Người quản lý nhập thông tin chi tiết môn học 3. Nhấn nút “Lưu” 4. Hệ thống kiểm tra 5. Hệ thống lưu lại thông tin môn học |
| Ngoại lệ | * 1. Hệ thống lưu thất bại, mã môn học bị trùng với môn học khác.   2. Người quản lý không nhập đầy đủ các thông tin chi tiết môn học |

#### Bảng Usecase xem thông tin chi tiết CTĐT

|  |  |
| --- | --- |
| Tên | Xem thông tin chi tiết CTĐT |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Thông tin chi tiết CTĐT đã tồn tại trong hệ thống |
| Hậu điều kiện | Xem thông tin chi tiết CTĐT thành công |
| Điều kiện tối thiểu | Yêu cầu nhập lại tên CTĐT cần tìm kiếm |
| Sự kiện chính | 1. Hệ thống hiển thị giao diện danh sách các CTĐT 2. Người dùng chọn một CTĐT cần xem 3. Hệ thống hiển thị các thông tin chi tiết của CTĐT |
| Ngoại lệ |  |

#### Bảng Usecase thêm CTĐT

|  |  |
| --- | --- |
| Tên | Thêm CTĐT |
| Tác nhân chính | Người quản lý |
| Tiền điều kiện | Thông tin các môn học thuộc CTĐT đã tồn tại trong hệ thống |
| Hậu điều kiện | Thêm thông tin chi tiết CTĐT thành công |
| Điều kiện tối thiểu | Yêu cầu nhập lại các thông tin CTĐT |
| Sự kiện chính | 1. Hệ thống hiển thị giao diện quản lý CTĐT 2. Người quản lý nhấn “Tạo mới CTĐT” 3. Nhập đầy đủ các thông tin chi tiết của CTĐT 4. Nhấn “Lưu” 5. Thêm các môn học vào CTĐT 6. Nhấn “Hoàn thành” 7. Hệ thống kiểm tra 8. Hệ thống lưu lại CTĐT |
| Ngoại lệ | * 1. Người dùng thêm một môn học 2 lần vào CTĐT   2. Người dùng nhập thiếu thông tin chi tiết CTĐT   3. Hệ thống lưu thất bại, mã CTĐT trùng với một CTĐT khác.   4. Hệ thông lưu thất bại, đã tồn tại CTĐT cùng ngành và cùng năm học. |

#### Bảng Cập nhật CTĐT

|  |  |
| --- | --- |
| Tên | Cập nhật CTĐT |
| Tác nhân chính | Người quản lý |
| Tiền điều kiện | Thông tin CTĐT đã tồn tại trong hệ thống  Thông tin các môn học thuộc CTĐT đã tồn tại trong hệ thống |
| Hậu điều kiện | Thêm thông tin chi tiết CTĐT thành công |
| Điều kiện tối thiểu | Yêu cầu nhập lại các thông tin CTĐT |
| Sự kiện chính | 1. Hệ thống hiển thị giao diện quản lý CTĐT 2. Người quản lý chọn một CTĐT cần cập nhật 3. Thay đổi các thông tin chi tiết cần thay đổi của CTĐT 4. Nhấn “Lưu” 5. Thay đổi các môn học của CTĐT 6. Nhấn “Hoàn thành” 7. Hệ thống kiểm tra 8. Hệ thống lưu lại CTĐT |
| Ngoại lệ | * 1. Người dùng thêm một môn học 2 lần vào CTĐT   2. Người dùng nhập thiếu thông tin chi tiết CTĐT   3. Hệ thống lưu thất bại, mã CTĐT trùng với một CTĐT khác.   4. Hệ thông lưu thất bại, đã tồn tại CTĐT cùng ngành và cùng năm học. |

#### Bảng Tạo phiếu khảo sát chuẩn đầu ra của CTĐT

|  |  |
| --- | --- |
| Tên | Tạo phiếu khảo sát |
| Tác nhân chính | Người quản lý |
| Tiền điều kiện | Thông tin CTĐT đã tồn tại trong hệ thống |
| Hậu điều kiện | Tạo thành công phiếu khảo sát |
| Điều kiện tối thiểu | Yêu cầu nhập lại các thông tin phiếu |
| Sự kiện chính | 1. Hệ thống hiển thị giao diện quản lý phiếu khảo sát 2. Người quản lý nhập các thông tin cần thiết của phiếu 3. Người quản lý chọn một CTĐT để làm khảo sát 4. Nhấn “Lưu” 5. Nhập các câu hỏi khảo sát 6. Nhấn “Hoàn thành” 7. Hệ thống kiểm tra 8. Hế thống lưu lại phiếu khảo sát |
| Ngoại lệ | * 1. Máy chủ quá tải, hệ thống lưu thất bại   2. Đường truyền mạng không ổn định, hệ thống lưu thất bại |

# TÀI LIỆU THAM KHẢO