# **TÓM TẮT**

*Em xin chân thành cảm ơn!*

|  |
| --- |
| *Hà Nội, ngày .. tháng .. năm 2018* |
| Sinh viên thực hiện |
|  |
|  |
| **Trần Thị Hương** |

# 

# **LỜI CẢM ƠN**

Trong thời gian nghiên cứu, học tập tại khoa Công nghệ thông tin trường Đại học Công nghệ Giao thông vận tải, được sự giúp đỡ quý báu của quý thầy cô giáo, các bạn sinh viên cùng các doanh nghiệp là đối tác của nhà trường đã giúp em hoàn thiện đồ án tốt nghiệp. Trước tiên cho phép em được bày tỏ lời cảm ơn sâu sắc nhất tới các thầy cô trong khoa Công nghệ thông tin. Đặc biệt là thầy TS. **Lê Chí Luận**, giảng viên khoa Công nghệ thông tin, trường Đại học Công nghệ Giao thông vận tải, người trực tiếp hướng dẫn, nhận xét, giúp đỡ em trong suốt quá trình thực hiện và hoàn thành đồ án.

Đồng thời em cũng xin chân thành cảm ơn Ban giám hiệu nhà trường, các thầy cô trong khoa Công nghệ thông tin cùng các phòng ban nhà trường đã tạo điều kiện tốt nhất cho em trong suốt thời gian học tập và làm đồ án tốt nghiệp này.

Tiếp theo, em xin gửi lời cảm ơn đến các thành viên lớp 65DCHT21, những người bạn đã luôn ở bên cạnh động viên, tạo điều kiện thuận lợi giúp em hoàn thành tốt đồ án, bên cạnh đó em xin cảm ơn gia đình đã tạo điều kiện về vật chất cũng như tinh thần, động viên giúp em hoàn thành đồ án này.

Trong quá trình thực hiện làm đồ án còn có rất nhiều sai sót, em mong nhận được sự giúp đỡ nhiệt tình của thầy cô cũng như các bạn để hoàn thiện một cách tốt nhất.

*Em xin chân thành cảm ơn!*

|  |
| --- |
| *Hà Nội, ngày .. tháng .. năm 2018* |
| Sinh viên thực hiện |
|  |
|  |
| **Trần Thị Hương** |

# **LỜI CAM ĐOAN**

|  |
| --- |
| *Hà Nội, ngày .. tháng .. năm 2018* |
| Sinh viên thực hiện |
|  |
|  |
| **Trần Thị Hương** |

**DANH MỤC HÌNH VẼ**

**DANH MỤC BẢNG**

**BẢNG KÝ HIỆU CÁC CỤM TỪ VIẾT TẮT**

# **CHƯƠNG 1. GIỚI THIỆU**

## **1.1. Đặt vấn đề**

## **1.2. Các đóng góp của đồ án**

## **1.3. Bố cục của đồ án**

# **CHƯƠNG 2. KIẾN THỨC NỀN TẢNG**

Qua quá trình học trên ghế nhà trường cùng với thời gian nghiên cứu tài liệu tham khảo, em đã tóm tắt được những kiến thức nền tảng chính được sử dụng trong xây dựng chương trình và thiết kế cơ sở dữ liệu và xây dựng chương trình như sau.

## **2.1. Cơ sở lý thuyết**

Để có thể xây dựng một website thì phân tích thiết kế , cơ sở dữ liệu và ngôn ngữ lập trình và mô hình lập trình là những phần không thể thiếu, vì vậy em đã nghiên cứu và tóm tắt những tài liệu và trọng điểm chính sau đây.

### 2.1.1. Phân tích thiết kế hướng đối tượng

*Định nghĩa*: Phân tích thiết kế hướng đối tượng là giai đọan phát triển một mô hình chính xác và súc tích của vấn đề, có thành phần là các đối tượng và khái niệm đời thực, dễ hiểu đối với người sử dụng.

Ưu điểm: Một trong những ưu điểm quan trọng bậc nhất của phương pháp phân tích và thiết kế hướng đối tượng là tính tái sử dụng: Có thể tạo các thành phần (đối tượng) một lần và dùng chúng nhiều lần sau đó.

### 2.1.2. Cơ sở dữ liệu quan hệ

Cơ sở dữ liệu là một hệ thống các thông tin có cấu trúc, được lưu trữ trên các thiết bị lưu trữ nhằm thỏa mãn yêu cầu khai thác thông tin đồng thời của nhiều người sử dụng hay nhiều chương trình ứng dụng chạy cùng một lúc với những mục đích khác nhau.

*Định nghĩa cơ sở dữ liệu quan hệ*: Cơ sở dữ liệu quan hệ dựa trên mô hình dữ liệu quan hệ. Trong mô hình này, dữ liệu được tổ chức thành một hoặc nhiều bảng (hoặc "quan hệ") có các cột và hàng. Mỗi hàng được xác định bằng cách sử dụng một khóa duy nhất.

Những hàng này được gọi là bản ghi hoặc bộ dữ liệu. Các cột được gọi là các thuộc tính. Mỗi bảng/quan hệ biểu thị một "loại thực thể" (chẳng hạn như sản phẩm hoặc cá nhân). Các hàng biểu thị một cá thể của một loại thực thể. Các cột biểu thị các thuộc tính được gán cho cá thể đó.

Nguyên lý ACID của cơ sở dữ liệu quan hệ rất nổi tiếng, khiến nó tạo ra một sự khác biệt so với cơ sở dữ liệu truyền thống. Do đó, em quyết định sử dụng mô hình này trong việc thiết kế CSDL. ACID là từ viết tắt các chữ cái đầu của bốn từ tiếng Anh atomicity, consistency, isolation, và durability. Chúng được coi là bốn thuộc tính quan trọng của một hệ quản trị cơ sở dữ liệu khi xử lý bất kỳ giao dịch nào. Nếu thiếu một trong những thuộc tính này thì tính toàn vẹn của cơ sở dữ liệu khó có thể được đảm bảo. Các tính chất ACID trong trường hợp này sẽ đảm bảo các giao dịch được thực hiện một cách đáng tin cậy:

* Tính nguyên tố (Atomicity). Một giao dịch có nhiều thao tác khác biệt thì hoặc là toàn bộ các thao tác hoặc là không một thao tác nào được hoàn thành..
* Tính nhất quán (Consistency). Một giao dịch hoặc là sẽ tạo ra một trạng thái mới và hợp lệ cho dữ liệu, hoặc trong trường hợp có lỗi sẽ chuyển toàn bộ dữ liệu về trạng thái trước khi thực thi giao dịch.
* Tính độc lập (Isolation). Một giao dịch đang thực thi và chưa được xác nhận phải bảo đảm tách biệt khỏi các giao dịch khác.
* Tính bền vững (Durability). Dữ liệu được xác nhận sẽ được hệ thống lưu lại sao cho ngay cả trong trường hợp hỏng hóc hoặc có lỗi hệ thống, dữ liệu vẫn đảm bảo trong trạng thái chuẩn xác.

Các cơ sở dữ liệu quan hệ phổ biến bao gồm Microsoft SQL Server, Oracle Database, MySQL và IBM DB2. Ngoài ra còn có các cơ sở dữ liệu quan hệ dựa trên đám mây như Dịch vụ cơ sở dữ liệu quan hệ của Amazon (RDS), Google Cloud SQL, IBM DB2, Cơ sở dữ liệu Microsoft Azure SQL và Dịch vụ đám mây cơ sở dữ liệu Oracle. Tuy nhiên, trong đề tài này, em sẽ sử dụng hệ quản trị Microsoft SQL Server.

### 2.1.3. Ngôn ngữ được sử dụng

#### a. HTML (HyperText Markup Language)

HTML (tiếng Anh, viết tắt cho HyperText Markup Language, hay là "Ngôn ngữ Đánh dấu Siêu văn bản", là một ngôn ngữ đánh dấu được thiết kế ra để tạo nên các trang web với các mẩu thông tin được trình bày trên World Wide Web. Cùng với CSS và JavaScript, HTML tạo ra bộ ba nền tảng kỹ thuật cho World Wide Web.

HTML được định nghĩa như là một ứng dụng đơn giản của SGML và được sử dụng trong các tổ chức cần đến các yêu cầu xuất bản phức tạp. HTML đã trở thành một chuẩn Internet do tổ chức World Wide Web Consortium (W3C) duy trì. Phiên bản chính thức mới nhất của HTML là HTML 4.01 (1999). Sau đó, các nhà phát triển đã thay thế nó bằng XHTML. Hiện nay, HTML đang được phát triển tiếp với phiên bản HTML5 hứa hẹn mang lại diện mạo mới cho Web.

#### b. CSS

*CSS* (Cascading Style Sheets) là một ngôn ngữ dùng để định dạng cho các phần tử HTML, hạn chế tối thiểu việc làm rối mã HTML của trang Web bằng các thẻ quy định kiểu dáng (chữ đậm, chữ in nghiêng, chữ có gạch chân, chữ màu), khiến mã nguồn của trang Web được gọn gàng hơn, tách nội dung của trang Web và định dạng hiển thị, dễ dàng cho việc cập nhật nội dung. Do đó, CSS tạo ra các kiểu dáng có thể áp dụng cho nhiều trang Web, giúp tránh phải lặp lại việc định dạng cho các trang Web giống nhau.

#### c. Javascript

JavaScript, theo phiên bản hiện hành, là một ngôn ngữ lập trình thông dịch được phát triển từ các ý niệm nguyên mẫu. Ngôn ngữ này được dùng rộng rãi cho các trang web (phía người dùng) cũng như phía máy chủ (với Nodejs). Nó vốn được phát triển bởi Brendan Eich tại Hãng truyền thông Netscape với cái tên đầu tiên Mocha, rồi sau đó đổi tên thành LiveScript, và cuối cùng thành JavaScript. Giống Java, JavaScript có cú pháp tương tự C, nhưng nó gần với Self hơn Java. .js là phần mở rộng thường được dùng cho tập tin mã nguồn JavaScript.

Trên trình duyệt, rất nhiều trang web sử dụng JavaScript để thiết kế trang web động và một số hiệu ứng hình ảnh thông qua DOM. JavaScript được dùng để thực hiện một số tác vụ không thể thực hiện được với chỉ HTML như kiểm tra thông tin nhập vào, tự động thay đổi hình ảnh,... Ở Việt Nam, JavaScript còn được ứng dụng để làm bộ gõ tiếng Việt giống như bộ gõ hiện đang sử dụng trên trang Wikipedia tiếng Việt. Tuy nhiên, mỗi trình duyệt áp dụng JavaScript khác nhau và không tuân theo chuẩn W3C DOM, do đó trong rất nhiều trường hợp lập trình viên phải viết nhiều phiên bản của cùng một đoạn mã nguồn để có thể hoạt động trên nhiều trình duyệt. Một số công nghệ nổi bật dùng JavaScript để tương tác với DOM bao gồm Ajax và SPA.

#### d. C#

C# là một ngôn ngữ lập trình hướng đối tượng hiện đại được phát triển vào năm 2000 bởi Anders Hejlsberg tại Microsoft, với vị thế như là một đối thủ của Java. Java được tạo ra bởi một nhóm kỹ sư tại Sun (sau đó được mua lại bởi Oracle) vì họ không muốn Microsoft thay đổi Java, do đó Microsoft chỉ còn cách tạo ra ngôn ngữ của riêng họ. C# đã phát triển nhanh chóng kể từ khi nó được ra mắt lần đầu tiên, với sự hỗ trợ to lớn từ Microsoft giúp nó đạt tạo được tiếng vang lớn. Và giờ đây nó trở thành một trong những ngôn ngữ lập trình phổ biến nhất trên thế giới. C# có rất nhiều ưu điểm nổi bật như:

* Gần gũi với các ngôn ngữ lập trình thông dụng như C++, Java, Pascal.
* Xây dựng dựa trên nền tảng của các ngôn ngữ lập trình mạnh nên thừa hưởng những ưu điểm của chúng.
* Cải tiến các khuyết điểm của C/C++ như con trỏ, hiệu ứng phụ,...
* Dễ dàng tiếp cận, dễ phát triển.

Nhược điểm lớn nhất của C# là chỉ chạy trên nền Windows và có cài .NET Framework. Thao tác đối với phần cứng yếu hơn so với ngôn ngữ khác, hầu hết phải dựa vào windows.

#### e. UML

UML là viết tắt của “Unifiled Modeling Language” – là ngôn ngữ mô hình hóa thống nhất, dùng để đặc tả, trực quan hóa và tư liệu hóa phần mềm hướng đối tượng, có phần chính bao gồm những ký hiệu hình học, được các phương pháp hướng đối tượng sử dụng để thể hiện và miêu tả các thiết kế của một hệ thống. Nó là một ngôn ngữ để đặc tả, trực quan hoá.

*Diagram (Biểu đồ):* Đồ thị biểu diễn đồ họa về tập các phần tử trong mô hình và mối quan hệ của chúng. Trong phiên bản mới nhất có 13 loại biểu đồ trong đó có 9 loại biểu đồ chính được sử dụng.

* Use Case Case Diagram (Biểu đồ ca sử dụng): Mô tả sự tương tác giữa các tác nhân ngoài và hệ thống thông qua các ca sử dụng.
* Class Diagram (Biểu đồ lớp): Mô tả cấu trúc tĩnh, mô hình khái niệm bao gồm các lớp đối tượng và các mối quan hệ của chúng trong hệ thống hướng đối tượng.
* *Object Diagram (Biểu đồ đối tượng):* Là một phiên bản của biểu đồ lớp và thường cũng sử dụng các ký hiệu như biểu đồ lớp.
* Sequence diagram (Biểu đồ trình tự): Thể hiện sự tương tác của các đối tượng với nhau.
* Collaboration Diagram (Biểu đồ cộng tác): Tương tự như biểu đồ trình tự nhưng nhấn mạnh vào sự tương tác của các đối tượng trên cơ sở cộng tác với nhau bằng cách trao đổi các thông điệp để thực hiện các yêu cầu theo ngữ cảnh công việc.
* State Diagram (Biểu đồ trạng thái): Thể hiện chu kỳ hoạt động của các đối tượng, của các hệ thống con và của cả hệ thống.
* *Activity Diagram* (Biểu đồ hành động): Chỉ ra dòng hoạt động của hệ thống.
* *Component diagram* (Biểu đồ thành phần*):* Chỉ ra cấu trúc vật lý của các thành phần trong hệ thống.
* *Deployment Diagram (Biểu đồ triển khai):* Chỉ ra cách bố trí vật lý các thành phần theo kiến trúc được thiết kế của hệ thống.

### 2.1.3. Công nghệ được sử dụng để phát triển hệ thống bên server

#### a. ASP.NET Framework

ASP.NET MVC là một phần mềm mã nguồn mở của Microsoft, kết hợp các tính năng của kiến trúc MVC (Model-View-Controller). Hầu hết các ý tưởng và kỹ thuật được phát triển từ nền tảng Agile và các phần tốt nhất của nền tảng ASP.NET hiện có.

ASP.NET MVC là một framework tuyệt vời hỗ trợ pattern MVC cho ASP.NET. Nếu bạn muốn hiểu ASP.NET MVC làm việc như thế nào, bạn cần phải có một sự hiểu biết rõ ràng về mô hình MVC. MVC là cụm từ viết tắt của Model-View-Controller, nó phân chia pattern của ứng dụng thành 3 phần - model, controller và view.

* Model giúp lưu trữ dữ liệu của ứng dụng và trạng thái của nó. Nó là một cơ sở dữ liệu hoặc cũng có thể chỉ là một tập tin XML.
* View được coi là một giao diện người dùng được sử dụng bởi khách truy cập trang web của bạn để nhìn thấy các dữ liệu. Các trang ASPX thường được sử dụng để hiển thị view trong các ứng dụng ASP.NET MVC.
* Controller chịu trách nhiệm xử lý các tương tác của người dùng với trang web. Nó được sử dụng để xác định loại view nào cần phải được hiển thị. Controller cũng được sử dụng cho mục đích giao tiếp với model.

Framework này là khá nhẹ và cung cấp khả năng kiểm thử, trong đó tích hợp với các tính năng hiện có của ASP.NET như xác thực (authentication) dựa trên membership và cả các master page.

#### b. Mô hình Repository kết hợp MVC

3-Layers là một kiến trúc kiểu client/server mà trong đó giao diện người dùng (UI-user interface), các quy tắc xử lý (BR-business rule hay BL-business logic), và việc lưu trữ dữ liệu được phát triển như những module độc lập, và hầu hết là được duy trì trên các nền tảng độc lập, và mô hình 3 tầng (3-tiers) được coi là một kiến trúc phần mềm và là một mẫu thiết kế.

* Tầng Presentation: hiển thị các thành phần giao diện để tương tác với người dùng như tiếp nhận thông tin, thông báo lỗi, …
* Tầng Business Logic: thực hiện các hành động nghiệp vụ của phần mềm như tính toán, đánh giá tính hợp lệ của thông tin, … Tầng này còn di chuyển, xử lí thông tin giữa 2 tầng trên dưới.
* Tầng Data: nơi lưu trữ và trích xuất dữ liệu từ các hệ quản trị CSDL hay các file trong hệ thống. Cho phép tầng Business logic thực hiện các truy vấn dữ liệu .

Dựa trên cơ sở của kiến trúc 3-layers, em tiến hành triển khai mô hình Repository kết hợp với MVC như sau:

* Tầng Presentation sẽ là project web ASP.NET. Trong đó bao gồm mô hình MVC. Mục đích của việc thiết kế này là giảm tải cho controller, tăng tốc độ request từ client. Ngoài ra, việc controller gắn chặt với view trong project web ASP.NET sẽ gây khó khăn trong việc Unit Test, nên khi đẩy phần code nghiệp vụ trong controller lên tầng Bussiness sẽ có thể Unit Test một cách dễ dàng.
* Bổ sung Tầng Repository: Là lớp trung gian giữa tầng Bussiness và tầng Data, giúp cho việc truy cập dữ liệu chặt chẽ và bảo mật hơn. Thông thường thì các phần truy xuất, giao tiếp với database năm rải rác ở trong code, khi bạn muốn thực hiện một thao tác lên database thì phải tìm trong code cũng như tìm các thuộc tính trong bảng để xử lý. Điều này gây lãng phí thời gian và công sức rất nhiều.Với Repository design pattern, thì việc thay đổi ở code sẽ không ảnh hưởng quá nhiều công sức chúng ra chỉnh sửa.

#### c. Entity Framework

Entity Framework (EF) là một framework ánh xạ quan hệ đối tượng (ORM) dành cho ADO.NET, là 1 phần của .NET Framework. EF cho phép các nhà phát triển Web tương tác với dữ liệu quan hệ theo phương pháp hướng đối tượng đặc trưng. Lợi ích lớn nhất của EF là giúp lập trình viên giảm thiểu việc lập trình mã nguồn cần thiết để truy cập và tương tác với cơ sở dữ liệu. EF được Microsoft hỗ trợ phát triển lâu dài và bền vững, vì vậy EF là 1 framework mạnh nhất hiện nay để phát triển ứng dụng Web với sự hỗ trợ đông đảo của các nhà phát triển Web.



Trong hình trên, EF dường như có vị trí rất tự do, xuất hiện ở cả 3 tầng Data Access Layer, Business Layer, và cả Presentation Layer. Mô hình này xuất phát có thể là sự cẩu thả của lập trình viên khi không thích tuân theo tiêu chuẩn quan hệ giữa các tầng hoặc cách giải thích khác là sự lỏng lẻo giúp cho việc lập trình thuận tiện hơn. Tuy nhiên lại gặp vấn đề khi muốn mở rộng dự án và sử dụng Framework khác thay cho EF sẽ khiến mất rất nhiều thời gian để sửa code. Lúc này, mô hình Repository phát huy tác dụng của nó. Code của EF sẽ được viết trong tầng Repository và khi thay đổi framework khác ta chỉ việc sửa code trong tầng Repository.



#### d. Identity Sample 2.0

Identity Sample 2.0 là một pakage được cung cấp bởi Microsoft nhằm hỗ trợ quá trình xác thực và phần quyền người dùng trong nhiều loại project web sử dụng .NET Framework khác nhau như ASP.NET MVC, Web Forms, Web Pages, Web API, và SignalR. Identity Sample hỗ trợ:

* Xác định danh tính người dùng khi client sử dụng website
* Kiểm soát thông tin người dùng website
* Sử dụng Entity Framework để lưu trữ dữ liệu người dùng
* Có thể unit test
* Quản lý phân quyền
* Hỗ trợ claims-based authentication
* Cung cấp các phương thức đăng nhập bằng các tài khoản từ mạng xã hội khác
* Sử dụng OWIN CookieAuthentication để tạo cookie lưu trữ thông tin đăng nhập.

### 2.1.3. Công nghệ được sử dụng để phát triển web application phía Client

#### a. Bootstrap

Bootstrap là một framework cho phép thiết kế website reponsive nhanh hơn và dễ dàng hơn

Bootstrap là bao gồm các HTML templates, CSS templates và Javascript tao ra những cái cơ bản có sẵn như: typography, forms, buttons, tables, navigation, modals, image carousels và nhiều thứ khác. Trong bootstrap có thêm các plugin Javascript trong nó. Giúp cho việc thiết kế reponsive của bạn dễ dàng hơn và nhanh chóng hơn.

*b. Jquery*

jQuery là một thư viện được xây dựng từ Javascript nhằm giúp lập trình viên xây dựng những chức năng có sử dụng Javascript trở nên dễ dàng hơn. jQuery được tích hợp nhiều module khác nhau từ module hiệu ứng cho đến module truy vấn selector. jQuery được sử dụng đến 99% trên tổng số website trên thế giới (trừ những website sử dụng JS Framework).

* Các module chính của jQuery bao gồm:
* Ajax - xử lý Ajax
* Atributes - Xử lý các thuộc tính của đối tượng HTML
* Effect - xử lý hiệu ứng
* Event - xử lý sự kiện
* Form - xử lý sự kiện liên quan tới form
* DOM - xử lý Data Object Model
* Selector - xử lý luồng lách giữa các đối tượng HTML

## **2.2. Công cụ sử dụng**

### 2.2.1. Visual Studio 2017

Visual studio là một trong những công cụ hỗ trợ lập trình rất nổi tiếng nhất hiện nay của Mcrosoft và chưa có một phần mềm nào có thể thay thế được nó. Visual Studio được viết bằng 2 ngôn ngữ đó chính là C# và VB+. Đây là 2 ngôn ngữ lập trình giúp người dùng có thể lập trình được hệ thống một các dễ dàng và nhanh chóng nhất thông qua Visual Studio.

Visual Studio là một phần mềm lập trình hệ thống được sản xuất trực tiếp từ Microsoft. Từ khi ra đời đến nay, Visual Studio đã có rất nhiều các phiên bản sử dụng khác nhau. Điều đó, giúp cho người dùng có thể lựa chọn được phiên bản tương thích với dòng máy của mình cũng như cấu hình sử dụng phù hợp nhất.

Bên cạnh đó, Visual Studio còn cho phép người dùng có thể tự chọn lựa giao diện chính cho máy của mình tùy thuộc vào nhu cầu sử dụng.

### 2.2.2. SQL Server 2014

SQL Server là một hệ thống quản lý cơ sở dữ liệu quan hệ, (relational database management system viết tắt là RDBMS), hỗ trợ một số lượng lớn các quy trình xử lý giao dịch, ứng dụng doanh nghiệp và ứng dụng phân tích trong các công ty IT. SQL Server là một trong 3 công nghệ dữ liệu dẫn đầu hiện này cùng với Oracle Database và IBM's DB2.

Cũng giống như các phần mềm RDBMS khác, Microsoft SQL Server được xây dựng bên trên lớp SQL - ngôn ngữ lập trình tiêu chuẩn hóa mà quản trị viên cơ sở dữ liệu (DBAs) và các chuyên gia CNTT sử dụng để quản lý cơ sở dữ liệu và truy vấn dữ liệu nằm bên trong. SQL Server thường gắn với Transact-SQL (T-SQL), một cài đặt SQL của Microsoft bổ sung một bộ chương trình mở rộng ngôn ngữ lập trình chuẩn.

# **CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG**

Trong chương này em đã khảo sát hệ thống để từ đó biết được quy trình đăng ký thực tập của khoa CNTT Trường Đại học CNGTVT xây dựng và phát triển các chức năng cần có, vẽ được biểu đồ usecase tổng quát, phân rã, các bản đặc tả yêu cầu chi tiết, biểu đồ hoạt động, trình tự, trạng thái và biểu đồ lớp.

## **3.1. Khảo sát hệ thống**

Để tìm hiểu và khảo sát cụ thể cho website, em đã trực tiếp đi khảo sát tại khoa công nghệ thông tin Trường Đại học Công nghệ GTVT tại 54 Phố Triều Khúc – Thanh Xuân – Hà Nội, với những bạn sinh viên không sắp xếp cuộc gặp gỡ và trao đổi được nên em có chuẩn bị biểu mẫu bảng hỏi thông qua google sheet.

### 3.1.1. Tìm hiểu những đặc trưng của dự án thuộc lĩnh vực công nghệ thông tin

Dự án công nghệ thông tin không giống như một dự án xây dựng bắt đầu bằng việc hình thành nên bức tranh toàn cảnh về mục tiêu cũng như sản phẩm được lên kế hoạch để hoàn thành. Dự án công nghệ thông tin giống như một con đường có nhiều ngã rẽ, chỉ một biến động nhỏ về cập nhật công nghệ hay thậm chí có tình huống công nghệ bị lạc hậu ngay trong quá trình triển khai dự án, các nhà quản lý bắt buộc phải thay đổi phạm vi, thêm mới các chức năng hay thậm chí thay đổi mục tiêu của mình đồng thời đảm bảo thời gian - tài chính đáp ứng duy trì hệ thống được đồng bộ.

**a. Đặc điểm**

* Xuất hiện cùng sự ra đời của công nghệ thông tin như một điều tất yếu.
* Tuân theo quy trình PMI: Khởi tạo - Lập kế hoạch - Thực hiện - Theo dõi & Kiểm soát - Kết thúc Dự án
* Coi trọng Truyền thông: Tại sao cho rằng truyền thông là yếu tố cần được chú trong trong quản lý dự án công nghệ thông tin? Bên cạnh việc kiểm soát thời gian, quản lý ngân sách, ứng biến rủi ro thì linh hoạt trong truyền thông cũng là điều mà bất cứ dự án nào cần coi trọng, đặc biệt với lĩnh vực CNTT. Tuy việc công nghệ thay đổi hàng ngày, hàng giờ đang mang lại những giá trị mới cho nhân loại nhưng chúng vốn tiềm ẩn nhiều rủi ro khiến các dự án CNTT gặp khó khăn, phải đổi hướng hay thậm chí là thất bại. Khi một yêu cầu thay đổi nhỏ về phần mềm hay phần cứng không được truyền đạt nhanh – đủ tới các đối tượng liên quan thì hậu quả sẽ vô cùng nghiêm trọng. Trong khi những rắc rối vô hình từ vấn đề công nghệ gây ra thường khó kiểm soát thì yếu tố con người luôn ổn định hơn cả. Sự phối hợp truyền thông và xử lý tình huống của người quản lý luôn đóng vai trò giải quyết đáng kể các rủi ro mà đặc thù lĩnh vực mang lại
* Vòng đời triển khai ngắn: Không giống như những sản phẩm của dự án xây dựng cơ bản có thời gian sử dụng đến vài chục năm, sản phẩm của các dự án CNTT thường có vòng đời ngắn hơn khi va phải yếu tố lạc hậu về công nghệ, yêu cầu người dùng thay đổi. Như vậy một dự án CNTT có thể quay vòng nhiều lần để nâng cấp theo kịp sự phát triển
* Kết hợp đào tạo: Sau khi hoàn thiện sản phẩm cũng như đi vào bàn giao, những người thực hiện dự án phần mềm hay ứng dụng công nghệ thông tin còn phải lên kế hoạch, chuẩn bị tài liệu cũng như các phương pháp đào tạo người dùng. Một sản phẩm CNTT sẽ trở nên vô dụng khi nó không được sử dụng hiệu quả.

**b. Phân loại quản lý dự án CNTT**

* Dự án phần cứng: Triển khai lắp đặt, đưa trang thiết bị công nghệ, phần cứng vào hoạt động (VD: Dự án lắp đặt hệ thống xử lý nước thải.v.v.)
* Dự án phần mềm: Triển khai ứng dụng phần mềm vào hoạt động quản lý, an ninh, hoạt động sản xuất (VD: Dự án an ninh mạng, xây dựng cơ sở dữ liệu…..).

**c. Quy trình của dự án CNTT**

* Khởi tạo dự án (Initiating): Giai đoạn này thực hiện việc định nghĩa một dự án mới hoặc một phát sinh (hoặc trộn lẫn) mới của một dự án có sẵn như: Xác định yêu cầu của dự án, mức độ ưu tiên của dự án, phân tích các yêu cầu đầu tư, phân công trách nhiệm cho các bộ phận triển khai.
* Lập kế hoạch dự án (Planning): Giai đoạn này yêu cầu thiết lập phạm vi công viêc của dự án, điều chỉnh lại mục tiêu và xác định đường đi tới mục tiêu đó.
* Triển khai (Executing): Giai đoạn này thực hiện hoàn thành các công việc được xác định trong phần lập kế hoạch để đảm bảo các yêu cầu của dự án.
* Giám sát và kiểm soát (Monitoring & Control): Giai đoạn này yêu cầu việc theo dõi, rà soát và điều chỉnh lại tiến độ và khả năng thực hiện của dự án. Theo dõi các rủi ro, thay đổi, phát sinh trong quá trình thực hiện và có những đề xuất điều chỉnh kịp thời.
* Kết thúc (Closing): Giai đoạn này thực hiện để kết thúc tất cả các hoạt động của dự án để chính thức đóng lại dự án.



### 3.1.2. Các hệ thống trên thị trường trong và ngoài nước

**a. Trello**

Trello là công cụ quản lý dự án vận hành theo phương pháp Kanban. Nguyên lí của phương pháp này là trực quan hoá công việc thành một bảng thông tin, gồm các cột tương ứng với trạng thái công việc To do, Doing, hoặc Done. Mục đích của việc phân loại là để giới hạn số “công việc đang tiến hành”, điều này giúp cho cả nhóm có thể làm việc tập trung hơn, tránh lãng phí thời gian khi phải di chuyển qua lại giữa các công việc khác nhau.

Ở Trello, mỗi “thẻ” đại diện cho một nhiệm vụ. Các thẻ được phân loại thành các danh sách, mỗi danh sách đại diện cho một tình trạng công việc hoặc một tính chất khác biệt (cần hoàn thành, đã hoàn thành…). Các cột được xếp trong một “bảng”, mỗi “bảng” là một dự án, hoặc một luồng công việc cụ thể.

Trello quản lý công việc dựa trên thẻ (card-based), các bình luận, trao đổi, ghi chú sẽ xảy ra xung quanh "thẻ" này. Trello hoạt động rất tốt nếu bạn tổ chức dự án theo chiều dọc (theo các giai đoạn khác nhau của dự án), bạn có thể di chuyển một thẻ (công việc) từ giai đoạn này sang giai đoạn tiếp theo.

Ví dụ: Bạn có thể sử dụng Trello cho việc quản lý các công việc của phòng Đi tư các "Công việc cần làm" đến "Công việc đang làm" và cuối cùng là "Công việc đã làm"

Qua thực tế sử dụng, Trello được khuyến nghị sử dụng cho các team từ 3 - 10 người, với đặc thù công việc theo dạng tuần tự, cần tập trung vào sự đơn giản, cộng tác hiệu quả, nhanh gọn thay vì quản trị.

*Ưu điểm của Trello*

- Dễ sử dụng

Giao diện làm việc của Trello giống hệt với các tờ giấy note được dán trên màn hình. Do vậy, bất kì ai nhìn vào cũng có thể dễ dàng nắm bắt và sử dụng. Các thao tác sử dụng cũng rất đơn giản.

- Theo dõi trực quan

Trello được thiết kế dựa trên phương pháp quản lý dự án Kanban, nên các giai đoạn công việc sẽ được phân chia thành các danh sách như các to-do list. Và chỉ cần nhìn vào giao diện, là nhà quản lý đã nắm bắt ngay được tiến độ dự án một cách trực quan nhất.

- Khả năng tích hợp lớn

Trello còn có ưu điểm đáng kể đó là khả năng tích hợp lớn khi sử dụng bản trả phí. Nếu bạn sử dụng gói Free, dung lượng file đính kèm chỉ có 10MB nhưng nếu bạn sử dụng gói Business, con số đó lên tới 250MB

*Nhược điểm của Trello:*

- Không lý tưởng cho các dự án có hàng trăm nhiệm vụ

- Khi số lượng thẻ bắt đầu gia tăng, Trello trở nên khó sử dụng, bởi người dùng bắt đầu mất tầm nhìn vào những gì đã làm, và cần phải tìm kiếm thẻ nhiều hơn và thường xuyên hơn.

- Không phân cấp thành viên quản trị

- Tất cả các thành viên tham gia dự án trên Trello đều có quyền chỉnh sửa, giao việc, xóa bỏ phần đã đánh dấu trước. Sự phân quyền lỏng lẻo hoàn toàn không phù hợp nếu có nhu cầu áp dụng Trello cho cấp độ doanh nghiệp.

- Môi trường giao tiếp kém

Mặc dù các thành viên có thể trao đổi trong các thẻ, tuy nhiên lại thiếu một giao diện cho các bình luận chung về toàn bộ dự án. Trello không có tính năng chat, vì vậy cần tích hợp với một công cụ giao tiếp khác như Slack.

- Không phù hợp cho quản lí thời gian

Với duy nhất một giao diện trải theo chiều ngang, Trello gây khó khăn cho người dùng trong việc quản lý thời gian chính xác của các công việc. Các thẻ được thiết kế độc lập, cản trở việc quản lý mối quan hệ giữa các đầu việc

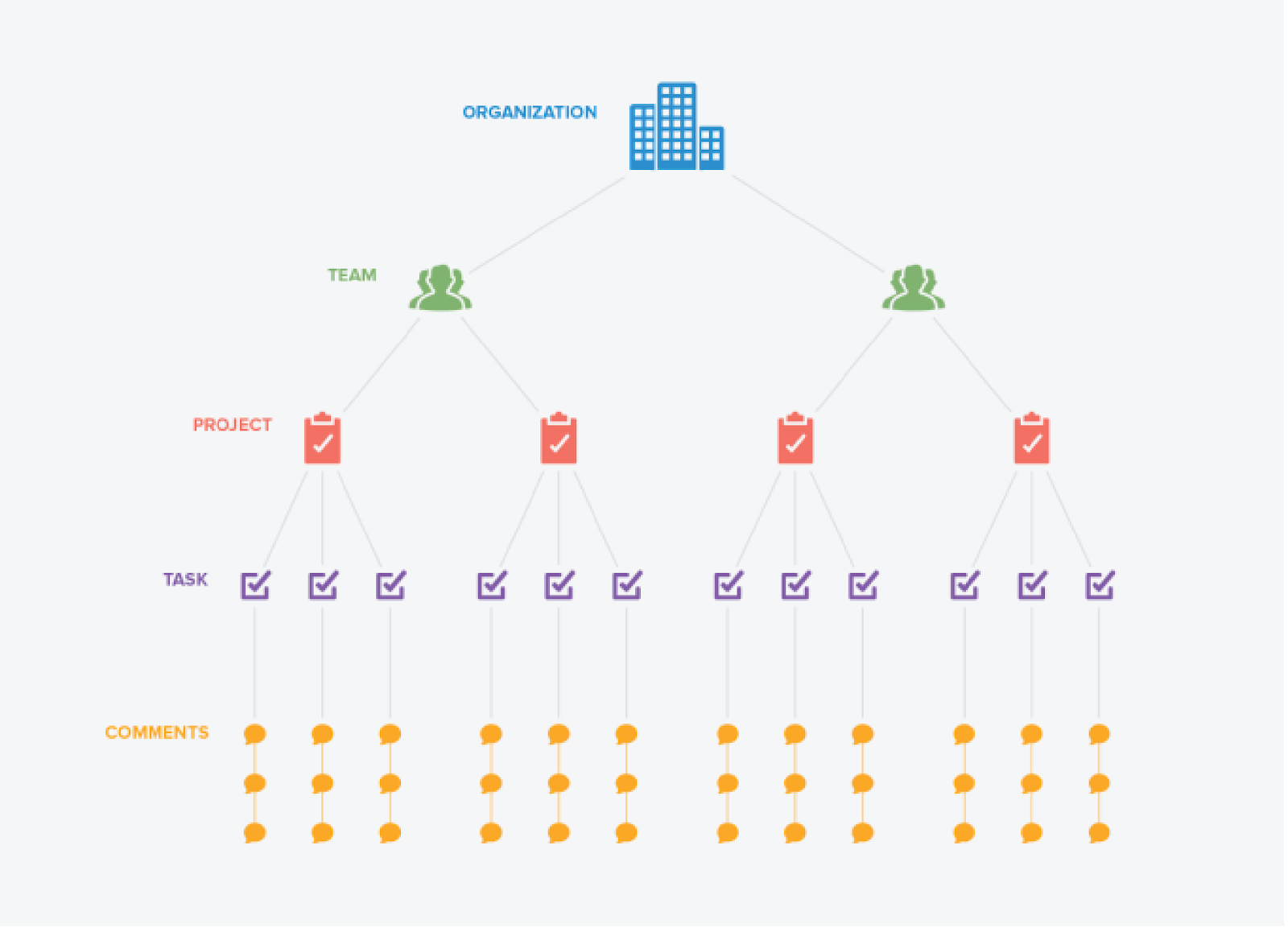
- Thiếu báo cáo công việc

Trello có thể là một công cụ tuyệt vời cho làm việc nhóm, nhưng lại thiếu đi nhiều tính năng thiết yếu đối với vai trò của một người Quản lý (Project/Team manager); trong đó phải kể đến việc báo báo. Trello không có một giao diện cho phép người quản lý theo dõi ngay được công việc đã hoàn thành được bao nhiêu % so với dự kiến, những cá nhân nào đang đảm bảo được tiến độ công việc được giao,...).

**b. Asana**

Khác với Trello, Asana không trực quan dòng công việc (work flow), nó trông giống như một danh sách việc phải làm (to-do list) của cả công ty, nơi tất cả mọi người có thể tạo nhiệm vụ và giao việc cho người khác.

Cách quản lý của Asana được minh hoạ qua hình sau:



Ở Asana, mỗi thành viên có một tài khoản riêng và có thể tham gia nhiều “Team” khác nhau. Ví dụ, bạn có thể đồng thời ở cả team Quản lý thi công, Quản lý dự án, Tài chính kế toán… Một người có thể là thành viên của nhiều hơn một team.

Trong mỗi team, “Dự án" được tạo ra và giúp xác định các khối công việc chính mà team cần làm. Bên trong các dự án này, bạn có danh sách "Nhiệm vụ- Task" và "Nhiệm vụ phụ- Subtask" được phân bổ cho các thành viên khác nhau. Những nhiệm vụ này chia nhỏ công việc trong một dự án và xác định rõ ràng ai đang làm gì và vào ngày nào (Trả lời được 2 W: Who + When). Bên trong mỗi công việc, bạn cũng có phần bình luận để giao tiếp với nhau về nhiệm vụ đang diễn ra.

Công cụ Asana quản lý dự án dựa trên các nhiệm vụ (task-based). Asana thích hợp cho các công ty có hệ thống công việc phức tạp, một người cần phải tham gia nhiều dự án/ phòng ban khác nhau. Khi đó, Asana không chỉ giúp team cộng tác hiệu quả, mà còn giúp người quản lý theo dõi tổng thể công việc ở tất cả các phòng ban và dự án.

*Ưu điểm của Asana:*

- Giao tiếp tốt

Asana có thể giúp loại bỏ việc dùng email hoặc ứng dụng thứ ba để giao tiếp, bởi người dùng được tự động nhận thông báo trong hộp thư đến của họ khi một nhiệm vụ được giao hoặc thay đổi. Tính năng bình luận ở từng nhiệm vụ có thể thay thế các email trao đổi công việc. Giao tiếp theo thời gian thực cho phép các thành viên phản hồi nhanh chóng, giảm thiểu các cuộc họp tốn thời gian.

- Theo dõi tổng quan dự án

Giao diện dạng danh sách, phân chia công việc theo thứ bậc từ lớn đến nhỏ, giúp người quản lý theo dõi được tổng thể dự án.

- Báo cáo công việc

Asana có báo cáo trực quan, cập nhật theo thời gian thực giúp nhà quản lý ngay lập tức đánh giá được hiệu quả của dự án, tuy nhiên tính năng này chỉ có ở phiên bản trả phí.

- Tuỳ chỉnh quyền riêng tư cho dự án

Ở phiên bản trả phí, Asana có thể tạo dự án với tuỳ chỉnh quyền riêng tư (dự án chỉ hiển thị với một số thành viên cho phép), tạo một không gian làm việc an toàn cho những công việc mang tính bảo mật cao.

*Nhược điểm của Asana:*

- Không phù hợp với các dự án có tính chất công việc theo tuần tự

Giao diện to-do list thiếu khả năng sắp xếp theo thứ tự chính là điểm yếu của Asana khi quản lý loại dự án này.

- Không có bản tiếng Việt

Hiện tại Asana chỉ có duy nhất một phiên bản tiếng Anh mà chưa mở rộng sang các ngôn ngữ khác.

- Không có biểu đồ Gantt cho quản lý dự án

Đối với các dự án lớn, có nhiều công việc phải hoàn thành cùng lúc, trình tự công việc chồng chéo nhau, trực quan hoá công việc bằng biểu đồ Gantt là điều cực kì cần thiết. Tuy nhiên hiện tại Asana vẫn chưa có tính năng này.

Từ những phân tích ở trên có thể bạn sẽ khó phân định được Trello hay Asana tối ưu hơn, bởi mỗi công cụ quản lý chỉ phục vụ tốt nhất cho những công việc mang tính đặc thù nhất định. Tùy theo tính chất công việc, quy mô dự án, quy mô đội ngũ, bạn có thể sử dụng Trello hoặc Asana kết hợp với một ứng dụng khác để đáp ứng những nhu cầu còn thiếu trong quản lý.

### 3.1.3. Kết luận

Với mỗi lĩnh vực khác nhau các dự án đều có những đặc thù riêng biệt. Để không mất nhiều thời gian, lãng phí nguồn lực cũng như đảm bảo ngân sách, các nhà quản lý cần nắm được các yếu tố tác động chủ yếu tới dự án của mình, nhanh chóng nhận biết và xử lý kịp thời. Dự án công nghệ thông tin cũng vậy, nhận biết được rủi ro công nghệ, rủi ro truyền thông, tập trung vào yếu tố con người sẽ giúp dự án khắc phục được các bất ổn tiêu cực mà đặc thù ngành mang lại.

Dựa trên những thông tin về đặc trưng của dự án CNTT, em nhận thấy đề tài “Thiết kế và xây dựng hệ thống quản lý dự án CNTT” sẽ đóng vai trò không nhỏ trong việc giải quyết được bài toán về rủi ro truyền thông trong việc quản lý dự án CNTT.

Dễ thấy, các hệ thống quản lý trên thế giới hiện nay đều không thể hỗ trợ sâu trong việc quản lý các dự án CNTT bởi vì họ muốn mở rộng phạm vi sang dự án của nhiều lĩnh vực khác nên chỉ quan tâm đến việc phát triển về mặt chiều rộng. Các công ty CNTT lớn trong nước thì sẽ tạo ra hệ thống kín cho riêng họ để quản lý nhưng các công ty khởi nghiệp, vừa vả nhỏ không thể có đủ kinh phí và kinh nghiệm để xây dựng một hệ thống quản lý dự án thuộc riêng lĩnh vực CNTT.

## **3.2. Yêu cầu cho hệ thống**

Phân tích làm rõ quy trình nghiệp vụ của hệ thống quản lý dự án CNTT bao gồm các chức năng :

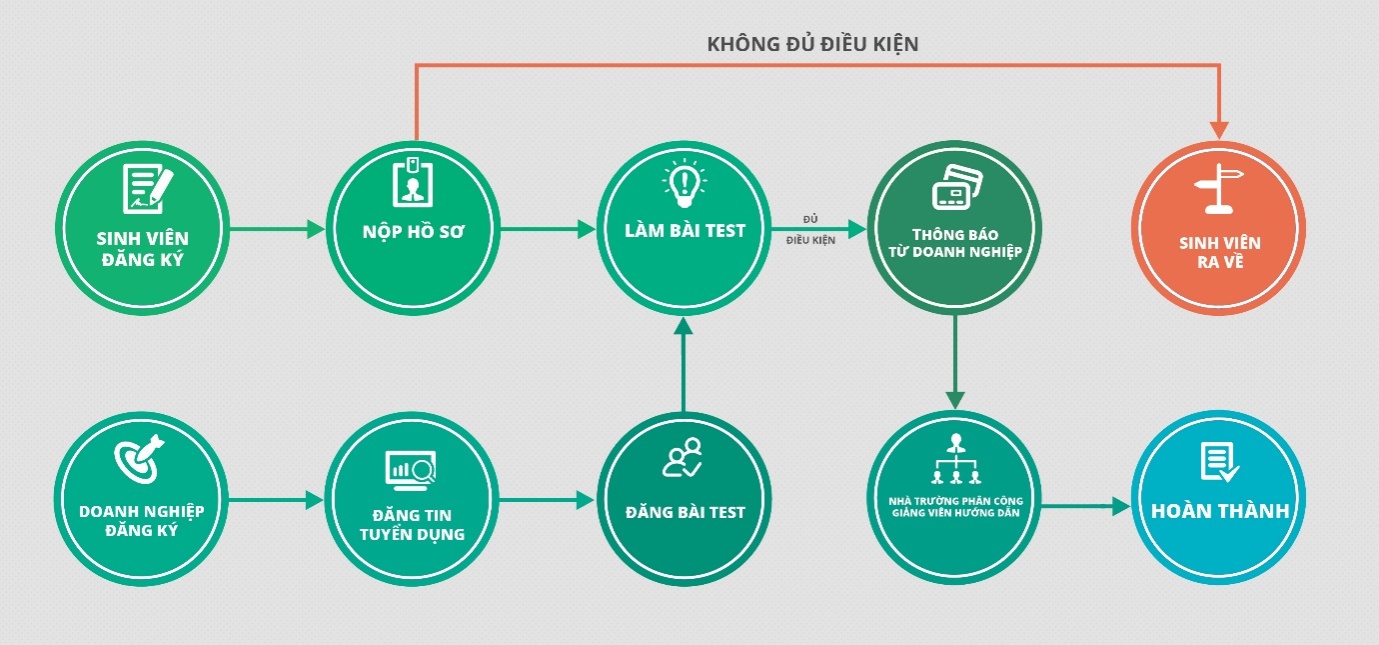
* Quản lý tài khoản
* Đăng nhập
* Đăng xuất
* Đổi mật khẩu
* Đăng ký tài khoản, xác nhận email
* Sửa tài khoản
* Xem thông tin của người sở hữu tài khoản
* Thống kê dự án tài khoản này đang tham gia
* Xóa tài khoản
* Quản lý thông tin dự án
* Tạo dự án
* Chỉnh sửa thông tin dự án
* Đóng dự án
* Xóa dự án
* Thống kê dự án
* Quản chức vụ công việc trong ngành công nghệ thông tin
* Thêm chức vụ
* Sửa chức vụ
* Phân quyền cho chức vụ trong mỗi dự án
* Xóa chức vụ
* Quản lý thành viên trong dự án
* Thêm thành viên vào dự án
* Gán chức vụ cho thành viên trong dự án
* Loại bỏ thành viên khỏi dự án
* Quản lý module của dự án
* Thêm module
* Sửa module
* Xóa module
* Quản lý loại công việc trong dự án
* Thêm loại công việc
* Sửa loại công việc
* Xóa loại công việc
* Quản lý công việc trong dự án
* Thêm công việc trong dự án
* Chỉnh sửa thông tin công việc
* Xóa công việc.
* Thống kê công việc trong dự án
* Thống kê tỷ lệ phần trăm vai trò của các thành viên trong dự án
* Thống kê nhật ký dự án
* Trích xuất nhật ký dự án ra file Excel
* Quản lý yêu cầu từ các thành viên trong dự án
* Thêm yêu cầu
* Chỉnh sửa yêu cầu
* Duyệt yêu cầu
* Tự động gửi mail khi có một yêu cầu cần duyệt
* Thống kê tiến độ dự án

## **3.3. Mô tả bài toán**

Website đăng ký thực tập doanh nghiệp cho sinh viên khoa CNTT Trường Đại học Công nghệ GTVT, với mục tiêu hỗ trợ sinh viên trong việc nộp hồ sơ thực tập đến các doanh nghiệp, tiết kiệm thời gian đối với doanh nghiệp và nhà trường một cách nhanh chóng và chính xác. Hệ thống được mô tả như sau: Người dùng bao gồm sinh viên có nhu cầu tìm kiếm việc làm thực tập và doanh nghiệp có nhu cầu đăng tin tuyển dụng có thể truy cập vào website, sau đó *đăng ký* một tài khoản tại website ứng với từng vai trò, dùng tài khoản vừa tạo mới đó *đăng nhập* vào hệ thống để thực hiện chức năng của mình. Trước tiên, để sinh viên có thể nhìn thấy tin tuyển dụng từ doanh nghiệp hiển thị trên trang chủthì doanh nghiệp đó phải vào hệ thống để *thêm mới tin tuyển dụng*, khi thêm mới hệ thống sẽ hỏi có tạo câu hỏi để sinh viên làm bài kiểm tra hay không. Nếu doanh nghiệp chọn có hệ thống sẽ có mục *thêm mới bài test* cho doanh nghiệp nhập, trường hợp câu hỏi hoặc tin tuyển dụng đó cần được thay đổi, cập nhật hoặc không cần thiết nữa doanh nghiệp có thể trực tiếp sửa hoặc xóa câu hỏi và tin tuyển dụng đó.

Sinh viên khi tìm thấy tin tuyển dụng phù hợp với chuyên môn, khả năng của bản thân thì thực hiện nộp hồ sơ bằng cách Xem chi tiết bài đăng đó, hệ thống sẽ dẫn đến màn hình thông tin chi tiết của tin tuyển dụng đó, lúc này sinh viên bấm nút *Nộp hồ sơ.* Màn hình yêu cầu nhập đầy đủ các thông tin chung và bắt buộc bao gồm họ tên, email và mã sinh viên cùng với option chọn muốn *Nộp hồ sơ bằng cách đính kèm file pdf* hay *Nộp hồ sơ bằng cách điền vào form mẫu*, nếu chọn nộp hồ sơ bằng đính kèm file sẽ xuất hiện một mẫu hồ sơ do nhà trường tạo ra yêu cầu sinh viên tải về điền vào và thực hiện upload file đó lên bằng bản pdf, nếu sinh viên chọn nộp hồ sơ bằng cách điền vào form mẫu màn hình hiển thị các thông tin chi tiết bắt buộc sinh viên phải điền vào như: hình thức mong muốn làm việc là partime hay fulltime, chuyên môn, kỹ năng, giải thưởng hay trình độ học vấn. v.v. Khi hồ sơ đã được nộp thành công đối với những tin tuyển dụng không có bài test hệ thống sẽ thông báo nộp hồ sơ thành công, với những tin tuyển dụng có bài test sẽ được chuyển hướng đến luôn màn hình *làm bài kiếm tra*. Số câu hỏi của doanh nghiệp là không cố định, một câu hỏi có 4 đáp án với 1 đáp án đúng và 3 đáp án sai, bằng hoặc trên 60% số câu đúng trên tổng số câu thì được coi là qua bài test, sinh viện thực hiện chọn phương án xong thì bấm nút Nộp bài. Sinh viên qua bài test và không qua bài test thì đềuđược thông báo số câu đúng trên tổng số câu, sau đó có thể về trang chủ và thực hiện nộp hồ sơ với tin tuyển dụng khác, ví dụ một đề tài yêu cầu số lượng là 10 sinh viên thì cho phép nộp 15 hồ sơ. Sinh viên có thể *sửa đổi nội dung hồ sơ sinh viên* trong trường hợp tất cả các doanh nghiệp mà nó đã nộp hồ sơ chưa thực hiện nhận xét và đánh giá. Ngược lại, khi đã có một doanh nghiệp nào đó đã nhận xét và đánh giá thì sinh viên đó không thể sửa đổi hồ sơ nữa.

Đối với những sinh viên qua bài kiểm tra, doanh nghiệp sẽ *thống kê* và nhìn thấy được, lúc này doanh nghiệp sẽ trực tiếp liên hệ qua mail hoặc số điện thoại có trong hồ sơ để gặp gỡ, trao đổi cũng như phỏng vấn. Sau buổi phỏng vấn đó, doanh nghiệp sẽ có *ý kiến nhận xét* về sinh viên đó và quyết định chọn trạng thái đạt hoặc không đạt. Sinh viên nào được doanh nghiệp chọn đạt thì sinh viên đó không được phép nộp hồ sơ cho các tin tuyển dụng khác nữa. Nhà trường sẽ có nhiệm vụ *thêm mới các giảng viên hướng dẫn,* trường hợp giảng viên đó cần sửa đổi nội dung thông tin hoặc giảng viên đó không có nhiệm vụ thì nhà trường có thể *sửa đổi hoặc xóa đi giảng viên*, sau khi doanh nghiệp nhận xét và đánh giá, nhà trường sẽ nhìn thấy tất cả các sinh viên đã trúng tuyển lúc này thực hiện *phân công giảng viên*, giảng viên hướng dẫn sẽ không được phân công vượt quá số lượng cho phép đã nhập, danh sách sinh viên đã được phân công sẽ được *thống kê* riêng cho nhà trường quản lý. Về mặt trang chủ, có hiển thị các bài viết tin tức mang tính chất giải trí, thư giãn có nội dung về chuyên ngành, giáo dục sẽ được *thêm sửa xóa tin tức* bởi nhà trường. Tất cả các đối tượng đều có thể truy cập vào trang chủ của website để thực hiện *tìm kiếm* thông qua chuyên môn, lĩnh vực.

****

#### Hình 3.. Quy trình đăng ký thực tập doanh nghiệp.

## **3.4. Phân tích thiết kế**

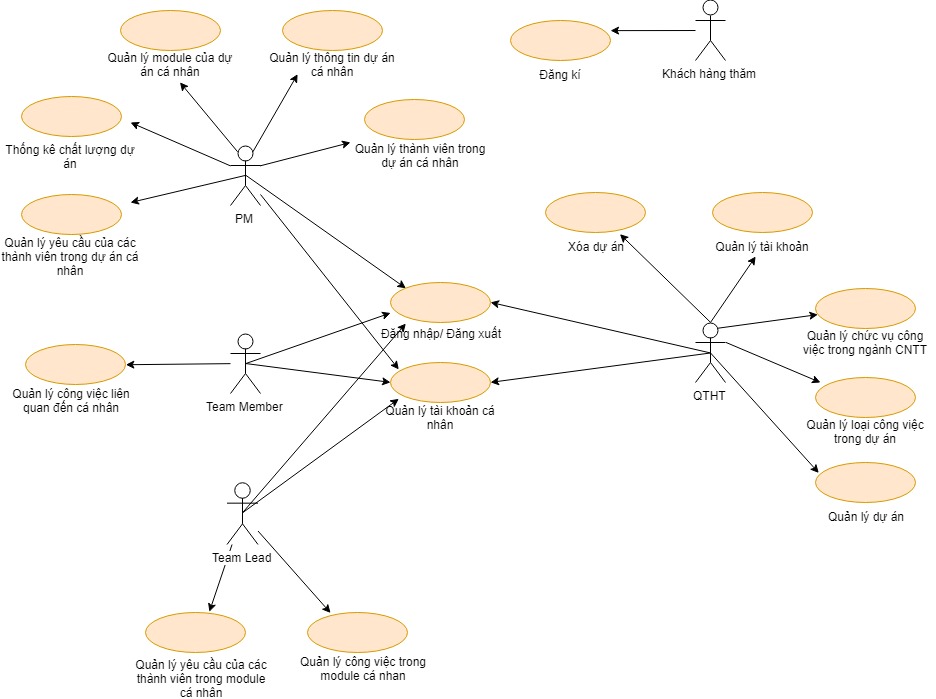
### 3.4.1. Xác định các tác nhân (Actor) và chức năng (Usecase)

Bảng 3.3. Xác định Actor và Usecase.

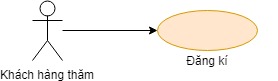
|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tác nhân** | **Mô tả tác nhân** | **Chức năng** |
| 1 | PM | Quản lý dự án | * Đăng nhập/ đăng xuất * Quản lý tài khoản cá nhân * Quản lý thông tin dự án, đóng dự án * Quản lý module của dự án * Quản lý thành viên trong dự án được tạo * Quản lý công việc trong dự án được tạo * Thống kê tỷ lệ phần trăm vai trò của các thành viên trong dự án được tạo * Thống kê nhật ký dự án được tạo * Trích xuất nhật ký của dự án ra file được tạo excel * Quản lý yêu cầu của các thành viên trong dự án được tạo * Thống kê tiến độ dự án được tạo * Thống kê danh sách dự án được tạo |
| 2 | Team Lead | Quản lý một module trong dự án | * Đăng nhập/ đăng xuất * Quản lý tài khoản cá nhân * Quản lý công việc trong module được tạo * Quản lý yêu cầu của các thành viên trong team module của mình |
| 3 | Team member | Thành viên trong dự án (Developer, Tester, BA,..) | * Đăng nhập/ đăng xuất * Quản lý tài khoản cá nhân * Tạo công việc, chuyển trạng thái công việc. |
| 4 | Khách hàng thăm | Những user không đăng nhập | * Đăng kí tài khoản * Đăng nhập * Xem giới thiệu website |
| 5 | QTHT | Quản trị hệ thống | * Đăng nhập/đăng xuất * Quản lý dự án: Xóa dự án * Quản lý chức vụ công việc trong ngành CNTT * Quản lý loại công việc trong dự án * Quản lý tài khoản |

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

#### **3.4.2.1. Usecase tổng quát**

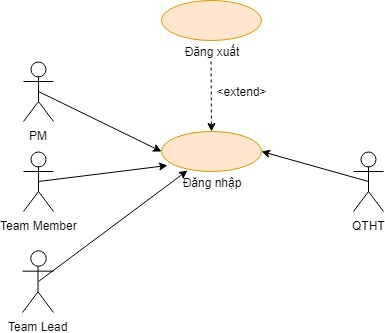


#### **3.4.2.5. Biểu đồ chi tiết Usecase Đăng kí**

******

|  |  |  |
| --- | --- | --- |
| 1 | Tên UC | Đăng kí |
| 2 | Tác nhân | Khách hàng thăm |
| 3 | Mục đích | Đăng kí hệ thống |
| 4 | Mô tả | Người dùng đăng ký làm thành viên của hệthống |
| 5 | Luồng sự kiện chính | 1. Người sử dụng chọn chức năng “Đăngký”2. Hệ thống hiển thị form “Đăng ký”3. Người sử dụng nhập thông tin4. Hệ thống lưu thông tin người dùng vàoCSDL và chuyển sang form “Đăng nhập” |
| 6 | Luồng sự kiện phụ | 1. Nhập sai thông tin2. Nhập thiếu thông tin |

#### **3.4.2.5. Biểu đồ chi tiết Usecase Đăng nhập**



|  |  |  |
| --- | --- | --- |
| 1 | Tên UC | Đăng kí |
| 2 | Tác nhân | QTHT, PM, Team Lead, Team Member |
| 3 | Mục đích | Đăng nhập hệ thống |
| 4 | Mô tả | Người sử dụng đăng nhập vào hệ thống thông qua “Email” và “Password” |
| 5 | Luồng sự kiện chính | 1. Người sử dụng chọn chức năng “Đăngnhập”2. Hệ thống hiển thị form “Đăng nhập”3. Người sử dụng nhập “Email” và “Password”, gửi thông tin đến hệ thống4. Hệ thống kiểm tra. Nếu đúng thì gửi mail xác nhận email5. Nếu người dùng sử dụng đúng mail để đăng kí ấn xác nhận trong hòm thư thì hệ thống sẽ cho phép người dùng truy cập, và sử dụng các chức năng theo đúng quyền hạn. Nếu không hợp lệ, hệ thống thông báo lỗi và yêu cầu người dùng nhập lại thông tin.6. Nếu người dùng ấn đăng xuất thì sẽ đăng xuất ra khỏi hệ thống và sẽ có quyền hạn của một khách hàng thăm. |
| 6 | Luồng sự kiện phụ | 1. Nhập sai định dạng “Email” hoặc“Password”2. Nhập thiếu thông tin |