

**ĐẠI HỌC ĐÀ NẴNG**  
**TRƯỜNG ĐẠI HỌC BÁCH KHOA**  
**KHOA CÔNG NGHỆ THÔNG TIN**

# **ĐỒ ÁN TỐT NGHIỆP**

**NGÀNH: CÔNG NGHỆ THÔNG TIN**  
**CHƯƠNG TRÌNH CÔNG NGHỆ THÔNG TIN CHẤT LƯỢNG**  
**CAO CHUẨN NHẬT BẢN**

**ĐỀ TÀI:**

## **XÂY DỰNG HỆ THỐNG QUẢN LÝ TIẾN ĐỘ CÔNG VIỆC VÀ DỰ ÁN TẠI CÔNG TY**

Người hướng dẫn: **TS. ĐẶNG HOÀI PHƯƠNG**

Sinh viên thực hiện: **ĐÀM VĂN HOÀNG BỬU**

Số thẻ sinh viên: **102160234**

Lớp: **16TCLC3**

**Đà Nẵng, 06/2021**

**ĐẠI HỌC ĐÀ NẴNG**  
**TRƯỜNG ĐẠI HỌC BÁCH KHOA**  
**KHOA CÔNG NGHỆ THÔNG TIN**

# **ĐỒ ÁN TỐT NGHIỆP**

**NGÀNH: CÔNG NGHỆ THÔNG TIN**

**CHƯƠNG TRÌNH CÔNG NGHỆ THÔNG TIN CHẤT LƯỢNG  
CAO CHUẨN NHẬT BẢN**

**ĐỀ TÀI:**

**XÂY DỰNG HỆ THỐNG QUẢN LÝ TIẾN ĐỘ  
CÔNG VIỆC VÀ DỰ ÁN TẠI CÔNG TY**

Người hướng dẫn: **TS. ĐẶNG HOÀI PHƯƠNG**

Sinh viên thực hiện: **ĐÀM VĂN HOÀNG BỬU**

Số thẻ sinh viên: **102160234**

Lớp: **16TCLC3**

**Đà Nẵng, 06/2021**

## This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the width of the page, providing a guide for handwriting practice. There are no margins, text, or other markings on the page.

[illegible]

## TÓM TẮT

Tên đề tài: Xây dựng hệ thống quản lý, giám sát tiến độ công việc và dự án tại công ty

Sinh viên thực hiện: Đàm Văn Hoàng Bửu

Số thẻ SV: 102160234      Lớp: 16TCLC3

Người hướng dẫn: TS. Đặng Hoài Phương

Hệ thống bao gồm:

- Website quản lý, giám sát tiến độ công việc và dự án tại công ty

Các chức năng chính:

- Quản lý thông tin nhân viên hiện có trong công ty
- Quản lý dự án của công ty
- Quản lý tiến độ công việc của dự án theo mô hình scrum-agile
- Quản lý thời gian biểu, chấm công của nhân viên

## NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Họ tên sinh viên: Đàm Văn Hoàng Bửu

Số thẻ sinh viên: 102160234

Lớp: 16TCLC3

Khoa: CNTT

Ngành: CLC Tiếng Nhật

1. Tên đề tài đồ án:

Xây dựng hệ thống quản lý, giám sát tiến độ công việc và dự án tại công ty

2. Đề tài thuộc diện: ☐ Có ký kết thỏa thuận sở hữu trí tuệ đối với kết quả thực hiện

3. Các số liệu và dữ liệu ban đầu:

.....  
.....  
.....

4. Nội dung các phần thuyết minh và tính toán:

.....  
.....  
.....  
.....  
.....

5. Các bản vẽ, đồ thị ( ghi rõ các loại và kích thước bản vẽ ):

.....  
.....  
.....  
.....

6. Họ tên người hướng dẫn: TS. Đặng Hoài Phương

7. Ngày giao nhiệm vụ đồ án: 05/02/2021.

8. Ngày hoàn thành đồ án: 1/06/2021.

*Đà Nẵng, ngày 20 tháng 12 năm 2019*

**Người hướng dẫn**

TS. Đặng Hoài Phương

## LỜI NÓI ĐẦU

Để đạt được kết quả tốt cho đồ án lần này, chúng em đã nhận được sự giúp đỡ nhiệt tình từ rất nhiều người. Với tình cảm sâu sắc và sự chân thành, chúng em muốn bày tỏ lòng biết ơn đối với tất cả những cá nhân, cơ quan đã giúp đỡ chúng em trong quá trình học tập và nghiên cứu.

Đầu tiên chúng em xin chân thành cảm ơn các thầy cô của trường Đại học Bách Khoa Đà Nẵng nói chung và khoa Công Nghệ Thông Tin nói riêng đã truyền đạt những kiến thức quý báu thông qua các giờ học trên giảng đường trong suốt những năm đại học, tạo tiền đề cho chúng em thực hiện tốt đồ án tốt nghiệp lần này.

Đặc biệt, chúng em muốn gửi lời cảm ơn sâu sắc thầy Đặng Hoài Phương – giảng viên khoa Công Nghệ Thông Tin, đã nhiệt tình hướng dẫn, chia sẻ các kinh nghiệm quý báu, và tạo điều kiện về nhiều mặt để chúng em có thể hoàn thành tốt được đồ án lần này.

Chúng em cũng xin chân thành cảm ơn gia đình, bạn bè và nhiều người xung quanh cũng tạo điều kiện, hỗ trợ về vật chất và tinh thần trong thời gian làm đồ án.

Với kinh nghiệm giới hạn của sinh viên, bài báo cáo có thể sẽ xảy ra những sai sót không muốn, em rất mong sự góp ý của quý thầy cô và mọi người để em rút kinh nghiệm.

Em xin chân thành cảm ơn.

*Đà Nẵng, ngày 1 tháng 06 năm 2021*

Sinh viên thực hiện

**Đàm Văn Hoàng Bửu**



## **CAM ĐOAN**

Chúng em xin cam đoan:

1. Nội dung trong đồ án này là do em thực hiện dưới sự hướng dẫn trực tiếp của thầy Đặng Hoài Phương.
2. Các tham khảo dùng trong đồ án đều được trích dẫn rõ ràng tên tác giả, tên công trình, thời gian, địa điểm công bố.
3. Mọi hình ảnh, video demo trong báo cáo đều do bản thân nhóm tự làm, không sao chép toàn bộ từ bất cứ nguồn nào. Nếu có những sao chép không hợp lệ, vi phạm, em xin chịu hoàn toàn trách nhiệm.

*Đà Nẵng, ngày 20 tháng 12 năm 2019*

Sinh viên thực hiện

**Đàm Văn Hoàng Bửu**

# MỤC LỤC

## TÓM TẮT

## NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

LỜI NÓI ĐẦU .....	i
CAM ĐOAN .....	ii
MỤC LỤC .....	iii
DANH SÁCH CÁC HÌNH VẼ .....	v
DANH SÁCH CÁC BẢNG .....	vii
DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT .....	ix
MỞ ĐẦU .....	10
1. Tổng quan về đề tài .....	10
2. Mục đích và nhiệm vụ chính của đề tài .....	11
2.1. Mục đích đề tài: .....	11
2.2. Nhiệm vụ chính của đề tài: .....	11
3. Công nghệ sử dụng .....	11
4. Công cụ hỗ trợ .....	11
5. Cấu trúc đồ án .....	12
CHƯƠNG 1. CƠ SỞ LÝ THUYẾT .....	13
1.1. Nghiên cứu lý thuyết .....	13
1.1.1. Kiến thức cơ bản .....	13
1.1.2. Tổng quan về ASP.NET CORE .....	14
1.1.3. Tổng quan về RESTful API .....	15
1.1.4. Tổng quan về ReactJS .....	19
1.1.5. Tổng quan về Microsoft SQL Server .....	24
1.1.6. Các công cụ sử dụng .....	25
1.2. Kết chương .....	27
CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG .....	28
2.1. Phân tích yêu cầu .....	28

2.1.1.	<i>Bài toán thực tế</i> .....	28
2.1.2.	<i>Bài toán đề xuất</i> .....	28
<b>2.2.</b>	<b>Phân tích chức năng</b> .....	29
2.2.1.	<i>Các module chính của hệ thống</i> .....	29
2.2.2.	<i>Sơ đồ cây phân rã chức năng</i> .....	30
2.2.3.	<i>Biểu đồ ca sử dụng</i> .....	31
2.2.4.	<i>Biểu đồ lớp</i> .....	40
2.2.5.	<i>Biểu đồ hoạt động</i> .....	41
2.2.6.	<i>Biểu đồ tuần tự</i> .....	45
<b>2.3.</b>	<b>Thiết kế cơ sở dữ liệu</b> .....	51
2.3.1.	<i>Sơ đồ quan hệ, thực thể dữ liệu</i> .....	51
2.3.2.	<i>Đặc tả các bảng dữ liệu</i> .....	52
<b>2.4.</b>	<b>Kết chương</b> .....	61
<b>CHƯƠNG 3. TRIỂN KHAI VÀ KẾT QUẢ</b> .....		62
<b>3.1.</b>	<b>Môi trường triển khai</b> .....	62
<b>3.2.</b>	<b>Kịch bản thử nghiệm</b> .....	62
<b>3.3.</b>	<b>Kết quả thực hiện</b> .....	63
<b>KẾT LUẬN</b> .....		73
<b>TÀI LIỆU THAM KHẢO</b> .....		74

## DANH SÁCH CÁC HÌNH VẼ

Hình 1.1 Mô tả cách thức hoạt động của Entity Framework.....	15
Hình 1.2 Mô tả RESTful API .....	16
Hình 1.3 Các phương thức HTTP của RESTful API .....	17
Hình 1.4 Hoạt động của RESTful API .....	18
Hình 1.5 Jordan Walke - người sáng lập ra ReactJS .....	19
Hình 1.6 Component trong ReactJS .....	20
Hình 1.7 Sử dụng State trong Component.....	21
Hình 1.8 Sơ đồ vòng đời của Component trong ReactJS .....	22
Hình 1.9 Tổng quan về SQL Server .....	24
Hình 1.10 Giao diện chính của StarUML .....	25
Hình 1.11 Giao diện chính của Visual Studio 2019 .....	25
Hình 1.12 Giao diện VSCode .....	26
Hình 1.13 Giao diện của Postman .....	26
Hình 2.1 Các module chính của dự án.....	29
Hình 2.2 Sơ đồ phân rã chức năng.....	30
Hình 2.3 Biểu đồ usecase cho module quản lý nhân viên .....	31
Hình 2.4 Biểu đồ usecase cho module quản lý dự án.....	32
Hình 2.5 Biểu đồ usecase cho module quản lý tiến độ công việc trong dự án.....	33
Hình 2.6 Biểu đồ usecase module quản lý thời gian làm việc.....	34
Hình 2.7 Biểu đồ lớp của dự án .....	40
Hình 2.8 Biểu đồ hoạt động của module quản lý nhân viên.....	41
Hình 2.9 Biểu đồ hoạt động của module quản lý dự án .....	42
Hình 2.10 Biểu đồ hoạt động của module quản lý tiến độ công việc .....	43
Hình 2.11 Biểu đồ hoạt động của module quản lý thời gian biểu, chấm công.....	44

Hình 2.12 Biểu đồ tuần tự quá trình đăng nhập.....	45
Hình 2.13 Biểu đồ tuần tự quá trình cập nhật thông tin nhân viên.....	46
Hình 2.14 Biểu đồ tuần tự quá trình thêm nhân viên.....	47
Hình 2.15 Biểu đồ tuần tự quá trình thêm mới dự án.....	48
Hình 2.16 Biểu đồ tuần tự thêm mới tiến trình.....	49
Hình 2.17 Biểu đồ tuần tự thêm mới công việc.....	50
Hình 2.18 Sơ đồ quan hệ, thực thể dữ liệu .....	51
Hình 3.1 Màn hình Login .....	63
Hình 3.2 Quản lí nhân viên.....	63
Hình 3.3 Xem thông tin cá nhân của nhân viên.....	64
Hình 3.4 Chỉnh sửa thông tin cá nhân .....	65
Hình 3.5 Lọc và tìm kiếm nhân viên dựa trên kĩ năng, vị trí, tên, .....	65
Hình 3.6 Màn hình quản lí các dự án.....	66
Hình 3.7 Tạo một dự án mới.....	67
Hình 3.8 Thông tin chi tiết của một công việc .....	68
Hình 3.9 Tạo một công việc mới .....	68
Hình 3.10 Giao việc cho thành viên trong dự án.....	69
Hình 3.11 Thêm nhãn cho công việc để dễ quản lí .....	69
Hình 3.12 Xóa một công việc .....	70
Hình 3.13 Các tùy chọn cho một bản .....	70
Hình 3.14 Quản lí thời gian .....	71
Hình 3.15 Ghi lại thời gian đã dành cho một công việc .....	72

## DANH SÁCH CÁC BẢNG

Bảng 2.1 Đặc tả ca sử dụng đăng nhập.....	34
Bảng 2.2 Đặc tả ca sử dụng cập nhật thông tin .....	35
Bảng 2.3 Đặc tả ca sử dụng thêm mới nhân viên .....	36
Bảng 2.4 Đặc tả ca sử dụng thêm mới dự án .....	36
Bảng 2.5 Đặc tả ca sử dụng thêm mới tiến trình .....	37
Bảng 2.6 Đặc tả ca sử dụng thêm mới công việc .....	37
Bảng 2.7 Đặc tả ca sử dụng quản lý công việc .....	39
Bảng 2.8 Bảng User .....	52
Bảng 2.9 Bảng Position .....	53
Bảng 2.10 Bảng Skill .....	53
Bảng 2.11 Bảng UserSkill .....	53
Bảng 2.12 Bảng Role .....	54
Bảng 2.13 Bảng UserRole .....	54
Bảng 2.14 Bảng TimeLineEvent .....	55
Bảng 2.15 Bảng Board.....	55
Bảng 2.16 Bảng Phase .....	56
Bảng 2.17 Bảng BoardUser .....	56
Bảng 2.18 Bảng Task.....	56
Bảng 2.19 Bảng Attachment.....	58
Bảng 2.20 Bảng Todo .....	58
Bảng 2.21 Bảng Label .....	59
Bảng 2.22 Bảng TaskLabel.....	59
Bảng 2.23 Bảng WorkLog.....	60
Bảng 2.24 Bảng Comment.....	60

Bảng 2.25 Bảng TaskAssignee .....	61
Bảng 2.26 Bảng TaskAction.....	61

## DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT

API	Application Programming Interface
JS	Javascript
SQL	Structured Query Language
UI	User Interface
DOM	Document Object Model
ASP	Active Server Pages
OS	Operating System
HTTP	HyperText Transer Protocol
WWW	World Wide Web
App	Application
URL	Uniform Resource Locator
ID	Identifier
.NET	Dotnet
ORM	Object Relational Mapper
ES	ECMA Script
AJAX	Asynchrouous Javascript and XML
IoT	Internet of Things
TCP	Transmission Control Protocol
IP	Internet Protocol



## MỞ ĐẦU

### 1. Tổng quan về đề tài

Bài toán của các nhà quản trị là làm sao kiểm soát những công việc và dự án đang diễn ra, tối ưu nguồn lực có hạn và nâng cao năng suất doanh nghiệp. Để giải được bài toán này, các phần mềm quản lý công việc giải quyết 2 việc:

- Thứ nhất, minh bạch hóa quá trình giao việc - nhận việc giữa nhà quản lý và nhân viên, từ đó minh bạch hóa trách nhiệm;
- Thứ hai, giúp nhà quản lý lên kế hoạch và giúp nhân viên cộng tác, làm việc tập trung trên một nền tảng duy nhất.

Tóm lại, lợi ích của việc sử dụng một phần mềm quản lý công việc có thể quan sát ở 2 góc độ. Đối với nhân viên, phần mềm giúp họ:

- Nhìn thấy được tất cả công việc của mình, không sót việc.
- Biết đâu là những việc cần ưu tiên.
- Tính toán và sắp xếp được thời gian làm việc hiệu quả.
- Cộng tác với đồng đội để hoàn thành công việc với chất lượng tốt nhất.

Đối với nhà quản lý thì một phần mềm quản lý công việc đem lại những lợi ích như sau:

- Thấy được tổng quan công việc và dự án của tất cả bộ phận
- Ra quyết định và xử lý các vấn đề phát sinh một cách kịp thời
- Sắp xếp, phân bổ nguồn lực (nhân sự + thời gian) một cách hiệu quả
- Tất cả công việc, dự án đều được hoàn thành đúng thời hạn với chất lượng tốt nhất

Hiện nay, trên thị trường có khá nhiều cái tên nổi bật như Trello, Asana, Wrike, Jira,... Tuy nhiên, phần lớn những ứng dụng trên đều tồn tại một số vấn đề bất cập như:

- Không phù hợp với quy trình làm việc của công ty
- Khuyết thiếu một số chức năng cần thiết
- Chi phí cao.

Do đó, cần thiết xây dựng một hệ thống để quản lý, giám sát tiến độ công việc, dự án phù hợp cho quy trình làm việc của công ty, dễ tùy chỉnh, đồng thời tối thiểu chi phí phát sinh cho công ty.

## **2. Mục đích và nhiệm vụ chính của đề tài**

### **2.1. Mục đích đề tài:**

Xây dựng một hệ thống giúp cho công ty có thể dễ dàng thực hiện các công việc sau:

- Tìm kiếm và truy xuất thông tin nhân viên của công ty.
- Quản lý, giao việc và giám sát tiến độ công việc cho các dự án.
- Quản lý thời gian làm việc của nhân viên.

### **2.2. Nhiệm vụ chính của đề tài:**

Đề tài có nhiệm vụ chính:

- Xây dựng website để admin có thể quản lý thông tin nhân viên, dự án và thời gian làm việc của toàn bộ nhân viên
- Xây dựng website để nhân viên có thể cập nhật thông tin tiến độ của công việc được giao, quản lý thời gian làm việc của mình và quản lý thông tin cá nhân.

## **3. Công nghệ sử dụng**

Hệ thống sử dụng những công nghệ mới và phổ biến trên thị trường:

- React JS
- Redux & Saga
- ASP .NET CORE Web API
- SQL Server
- Azure services

## **4. Công cụ hỗ trợ**

- Visual Studio 2019
- Visual Studio Code
- SQL Server Management System
- Postman

## **5. Cấu trúc đồ án**

**MỞ ĐẦU-** Giới thiệu tổng quan về đồ án, lý do chọn đề tài. Trình bày về mục đích, nhiệm vụ và cách thức tiến hành để đạt được kết quả cao. Tóm tắt nội dung của những phần tiếp theo.

**Chương 1: LÝ THUYẾT VÀ CÔNG NGHỆ-** Chương này giới thiệu về tất cả cơ sở lý thuyết và công nghệ sử dụng trong đồ án.

**Chương 2: PHÂN TÍCH VÀ THIẾT KẾ-** Chương này trình bày về tổng quan hệ thống, việc phân tích, đặc tả yêu cầu, và thiết kế hệ thống. Trình bày về phương pháp phát hiện té ngã được lựa chọn.

**Chương 3: TRIỂN KHAI VÀ KẾT QUẢ-** Chương này trình bày về kết quả của việc lập trình của hệ thống, các hình ảnh thực tế và các bước cài đặt để sử dụng hệ thống.

**KẾT LUẬN-** Nêu lên kết luận của nhóm về đồ án, các bài học trong quá trình thực hiện, nêu lên nhận xét về các vấn đề còn tồn tại và các đề xuất.

**TÀI LIỆU THAM KHẢO-** Trình bày về các thông tin cần thiết, liên quan đến các nguồn trích dẫn trong đồ án..

## **CHƯƠNG 1. CƠ SỞ LÝ THUYẾT**

### **1.1. Nghiên cứu lý thuyết**

#### **1.1.1. Kiến thức cơ bản**

##### **1.1.1.1. HTML**

Là viết tắt cho HyperText Markup Language, hay là "Ngôn ngữ đánh dấu siêu văn bản", được thiết kế ra để tạo nên các trang web với các mẫu 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 [1]:

##### **1.1.1.2. CSS**

Là chữ viết tắt của Cascading Style Sheets, nó là một ngôn ngữ được sử dụng để tìm và định dạng lại các phần tử được tạo ra bởi các ngôn ngữ đánh dấu (ví dụ như HTML). Có thể hiểu đơn giản rằng, nếu HTML đóng vai trò định dạng các phần tử trên website như việc tạo ra các đoạn văn bản, các tiêu đề, bảng,...thì CSS [2] sẽ góp phần thêm một chút “phong cách” vào các phần tử HTML đó như đổi màu sắc trang, đổi màu chữ, thay đổi cấu trúc,... rất nhiều.

##### **1.1.1.3. JavaScript**

Là một ngôn ngữ lập trình kịch bản dựa trên đối tượng được phát triển từ các ý niệm nguyên mẫu. Ngôn ngữ này được dùng rộng rãi cho các trang web, nhưng cũng được dùng để tạo khả năng viết script sử dụng các đối tượng nằm sẵn trong các ứng dụng. 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 là Java. Nó là phần mở rộng thường được dùng cho tập tin mã nguồn JavaScript [3].

## **1.1.2. Tổng quan về ASP.NET CORE**

### **1.1.2.1. Giới thiệu về ngôn ngữ C#**

C# (hay còn gọi là C-Sharp) [4] là ngôn ngữ lập trình đa năng, mạnh mẽ được phát triển bởi Microsoft., C# là khởi đầu cho kế hoạch .NET của họ. Là một ngôn ngữ được dẫn xuất từ C và C++ nó kế thừa những đặc tính tốt của những ngôn ngữ này, và hơn thế nữa, C# trở nên đơn giản hơn trong việc tiếp cận, học tập và sử dụng khi đã loại bỏ những macro, template, đa kế thừa, lớp cơ sở ảo.

Đây là một ngôn ngữ khá đa năng khi được ứng dụng trong nhiều lĩnh vực như lập trình ứng dụng windows, lập trình web, lập trình game. Chính vì điều đó C# đã và đang trở thành một trong những ngôn ngữ lập trình được sử dụng phổ biến nhất trên thế giới.

### **1.1.2.2. Giới thiệu về ASP.NET Core**

ASP.NET Core là một open-source mới và framework đa nền tảng (cross-platform) cho việc xây dựng những ứng dụng hiện tại dựa trên kết nối đám mây, giống như web apps, IoT và backend cho mobile. Do là một framework đa nền tảng nên .NET hỗ trợ hầu hết các hệ điều hành như Windows, Linux, macOS.

Ứng dụng ASP.NET Core có thể chạy trên phiên bản đầy đủ của .NET Framework. Nó được thiết kế để cung cấp và tối ưu development framework cho những dụng cụ mà được triển khai trên đám mây (cloud) hoặc chạy on-promise.

DotNET Core framework bao gồm .NET Core, ASP.NET Core, Entity Framework Core. Các công nghệ này khác với .NET Framework ở chỗ chúng chạy CoreCLR runtime (used in Universal Windows Platform)

### **1.1.2.3. Entity Framework**

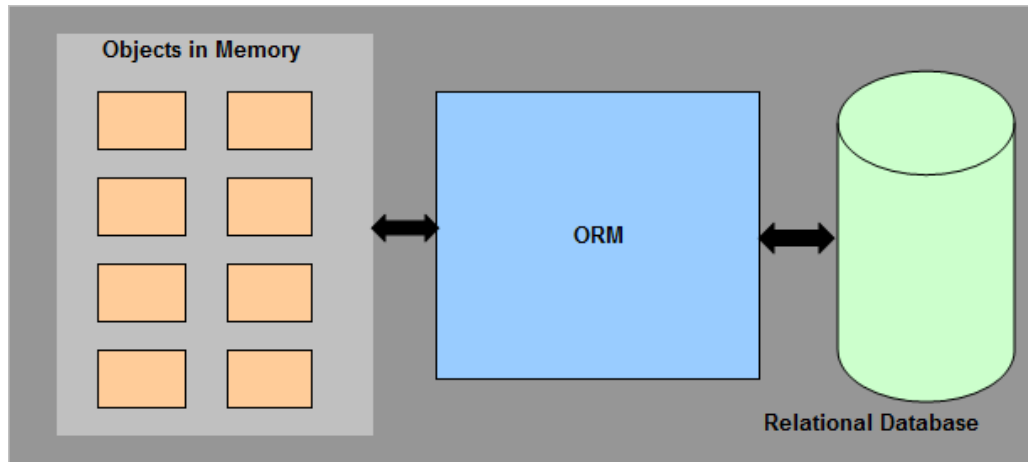
Entity Framework được ra mắt lần đầu tiên vào năm 2008, giúp cho việc truy vấn cơ sở dữ liệu quan hệ trở lên đơn giản, dễ học, giúp giảm bớt thời gian phát triển dự án xuống đáng kể.

Entity Framework là một Object Relational Mapper(ORM) một công cụ mapping giữa các objects trong ứng dụng với các tables và columns của cơ sở dữ liệu quan hệ:

- Entity Framework là một Open source ORM Framework.
- Một ORM có thể tạo kết nối đến một Relational database, thực thi câu lệnh và trả về kết quả là các objects trong ứng dụng.

- Một ORM sẽ theo dõi sự thay đổi dữ liệu, trạng thái của các objects trả về ở trên, mỗi khi có sự thay đổi dữ liệu từ những objects này ORM sẽ gửi những thay đổi đó đến Relational database.

Hiểu một cách đơn giản thì ORM là công cụ tự động tạo ra các Objects trong ứng dụng từ tables, views, store procedures của một Relational Database nào đó..



Hình 1.1 Mô tả cách thức hoạt động của Entity Framework

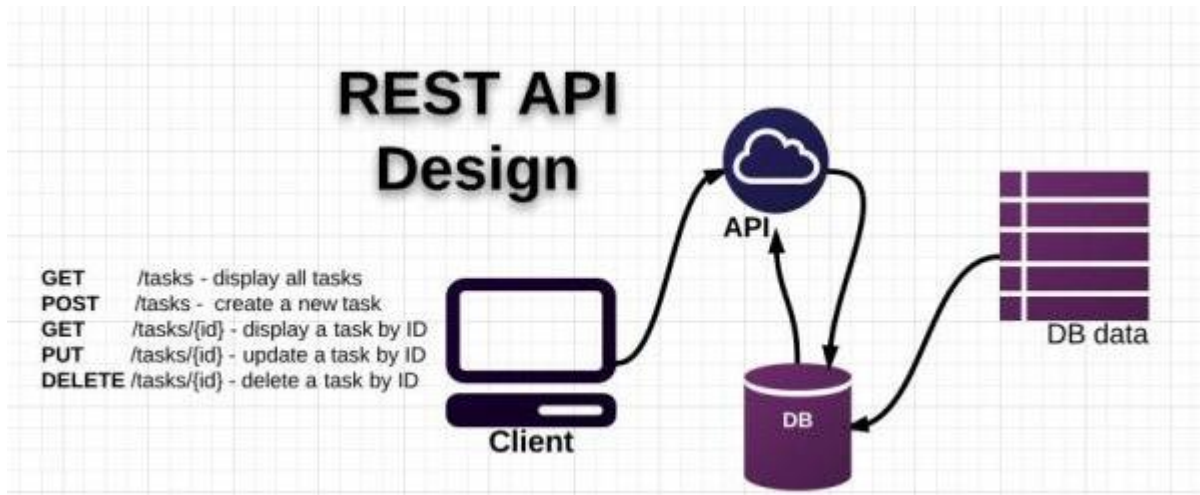
#### 1.1.2.4. Phương thức HTTP

**HTTP** (Tiếng Anh: **HyperText Transfer Protocol** - Giao thức truyền tải siêu văn bản) là một trong năm giao thức chuẩn của mạng Internet, được dùng để liên hệ thông tin giữa Máy cung cấp dịch vụ (Web server) và Máy sử dụng dịch vụ (Web client) trong mô hình Client/Server dùng cho World Wide Web-WWW, **HTTP** là một giao thức thuộc tầng ứng dụng, nằm trên cặp giao thức tầng giao vận & tầng mạng là TCP/IP.

### 1.1.3. Tổng quan về RESTful API

#### 1.1.3.1. Định nghĩa

RESTful API là một tiêu chuẩn dùng trong việc thiết kế API cho các ứng dụng web (thiết kế Web services) để tiện cho việc quản lý các resource. Nó chú trọng vào tài nguyên hệ thống (tệp văn bản, ảnh, âm thanh, video, hoặc dữ liệu động...), bao gồm các trạng thái tài nguyên được định dạng và được truyền tải qua HTTP [6].



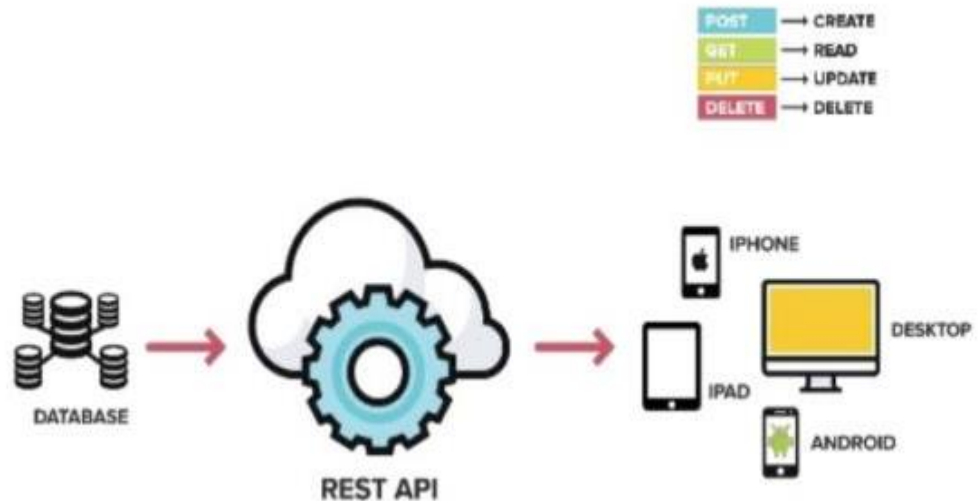
Hình 1.2 Mô tả RESTful API

**API** (Application Programming Interface) là một tập các quy tắc và cơ chế mà theo đó, một ứng dụng hay một thành phần sẽ tương tác với một ứng dụng hay thành phần khác. API có thể trả về dữ liệu mà cần cho ứng dụng của người lập trình ở những kiểu dữ liệu phổ biến như JSON hay XML.

**REST** (REpresentational State Transfer) là một dạng chuyển đổi cấu trúc dữ liệu, một kiểu kiến trúc để viết API. Nó sử dụng phương thức HTTP đơn giản để tạo cho giao tiếp giữa các máy. Vì vậy, thay vì sử dụng một URL cho việc xử lý một số thông tin người dùng, REST gửi một yêu cầu HTTP như GET, POST, PUT, DELETE, vv đến một URL để xử lý dữ liệu.

RESTful API là một tiêu chuẩn dùng trong việc thiết kế các API cho các ứng dụng web để quản lý các resource. RESTful là một trong những kiểu thiết kế API được sử dụng phổ biến ngày nay để cho các ứng dụng (web, mobile...) khác nhau giao tiếp với nhau.

Chức năng quan trọng nhất của REST là quy định cách sử dụng các HTTP method (như GET, POST, PUT, DELETE...) và cách định dạng các URL cho ứng dụng web để quản các resource. RESTful không quy định logic code ứng dụng và không giới hạn bởi ngôn ngữ lập trình ứng dụng, bất kỳ ngôn ngữ hoặc framework nào cũng có thể sử dụng để thiết kế một RESTful API.



Hình 1.3 Các phương thức HTTP của RESTful API

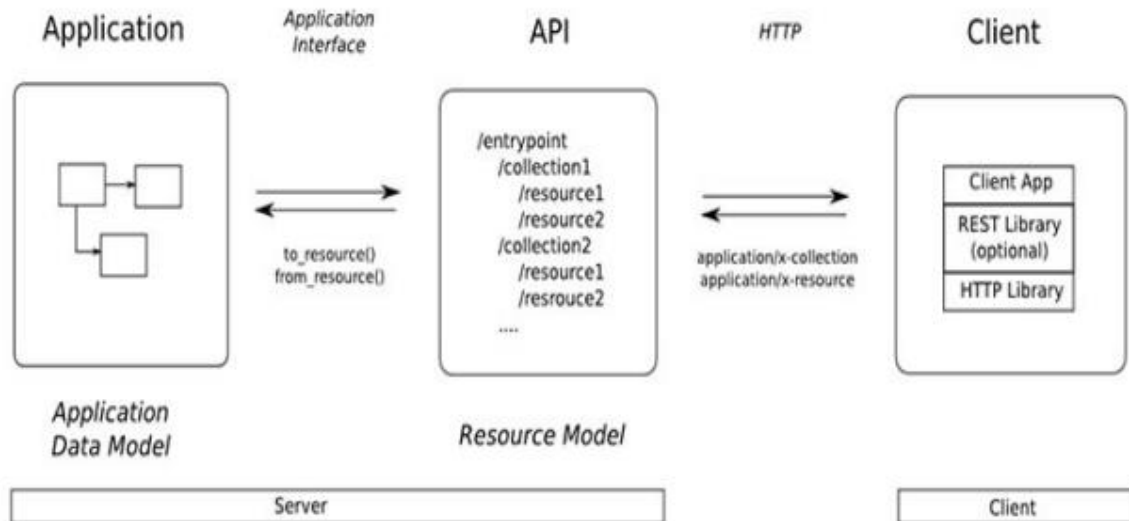
#### 1.1.3.2. Hoạt động

REST hoạt động chủ yếu dựa vào giao thức HTTP. Các hoạt động cơ bản nêu trên sẽ sử dụng những phương thức HTTP riêng:

- GET (SELECT): Trả về một Resource hoặc một danh sách Resource;
- POST (CREATE): Tạo mới một Resource; - PUT (UPDATE): Cập nhật thông tin cho Resource;
- DELETE (DELETE): Xóa một Resource.

Những phương thức hay hoạt động này thường được gọi là CRUD tương ứng với Create, Read, Update, Delete – Tạo, Đọc, Sửa, Xóa.





Hình 1.4 Hoạt động của RESTful API

Sau quá trình gọi API việc trả về những mã trạng thái cụ thể khi nhận và phiên dịch một HTTP Request, Server sẽ gửi tín hiệu phản hồi là một HTTP Response là một trong số các loại trạng thái sau:

- 200 OK – Trả về thành công cho những phương thức GET, PUT, PATCH hoặc DELETE;
- 201 Created – Trả về khi một Resource vừa được tạo thành công;
- 204 No Content – Trả về khi Resource xóa thành công;
- 304 Not Modified – Client có thể sử dụng dữ liệu cache;
- 400 Bad Request – Request không hợp lệ;
- 401 Unauthorized – Request cần có auth;
- 403 Forbidden – bị từ chối không cho phép;
- 404 Not Found – Không tìm thấy resource từ URI;
- 405 Method Not Allowed – Phương thức không cho phép với user hiện tại;
- 410 Gone – Resource không còn tồn tại, Version cũ đã không còn hỗ trợ;
- 415 Unsupported Media Type – Không hỗ trợ kiểu Resource này;
- 422 Unprocessable Entity – Dữ liệu không được xác thực;
- 429 Too Many Requests – Request bị từ chối do bị giới hạn

#### **1.1.4. Tổng quan về ReactJS**

##### **1.1.4.1. Giới thiệu**

React (hay ReactJS, React.js) là một thư viện Javascript mã nguồn mở để xây dựng các thành phần giao diện có thể tái sử dụng. Nó được tạo ra bởi *Jordan Walke*, một kỹ sư phần mềm tại Facebook. Người bị ảnh hưởng bởi XHP (Một nền tảng thành phần HTML cho PHP). React lần đầu tiên được triển khai cho ứng dụng Newsfeed của Facebook năm 2011, sau đó được triển khai cho Instagram.com năm 2012. Nó được mở mã nguồn (open-sourced) tại JSConf US tháng 5 năm 2013 [7].



Hình 1.5 Jordan Walke - người sáng lập ra ReactJS

Đặc tính của ReactJS:

- Tư tưởng của ReactJS là xây dựng lên các components có tính tái sử dụng, dễ dàng cho việc chia nhỏ vấn đề, testing. Nó giúp dễ dàng quản lý, mở rộng hệ thống, điều này nếu là Angular JS thì đòi hỏi cấu trúc, cách viết code phải thật tối ưu.
- ReactJS luôn giữ các components ở trạng thái stateless (nhiều nhất có thể) khiến ta dễ dàng quản lý bởi nó chẳng khác gì một trang static HTML. Bản thân các components này không có trạng thái (state) nó nhận đầu vào từ bên ngoài và chỉ hiển thị ra dựa vào các đầu vào đó, điều này giải thích tại sao nó mang tính tái sử dụng và dễ dàng cho việc test.

#### 1.1.4.2. Các đặc trưng của ReactJS

Component:

- Là một đoạn mã ngắn có ý nghĩa thể hiện một chức năng hay đối tượng nhất định, có thể tái sử dụng.
- Một ứng dụng React có thể sử dụng hàng chục, hoặc hàng trăm các component liên kết và tương tác với nhau.
- Ta có thể tạo ra một component bằng cách extends từ Component như sau:

```
1  import React, { Component } from "react";
2
3  class Welcome extends Component {
4    render() {
5      return (
6        <h1>Hello, My name is Trung!</h1>
7      );
8    }
9  }
10 export default Welcome;
```

Hình 1.6 Component trong ReactJS

- Phương thức quan trọng nhất là render(), phương thức này được trigger khi component đã sẵn sàng để được render lên trên trang. Trong hàm này, sẽ trả về mô tả cho việc muốn React render cái gì lên trên trang. Hàm render chính là mô tả cụ thể của giao diện người dùng tại bất cứ thời điểm nào. Vì thế nếu dữ liệu thay đổi, React sẽ quan tâm đến việc cập nhật lại giao diện với dữ liệu tương ứng.

Props:

- Thông tin được truyền từ component này sang component khác được gọi là props.
- Props của một component có thể là một object, function, string, number .... Nó chứa thông tin về thành phần đó.

State:

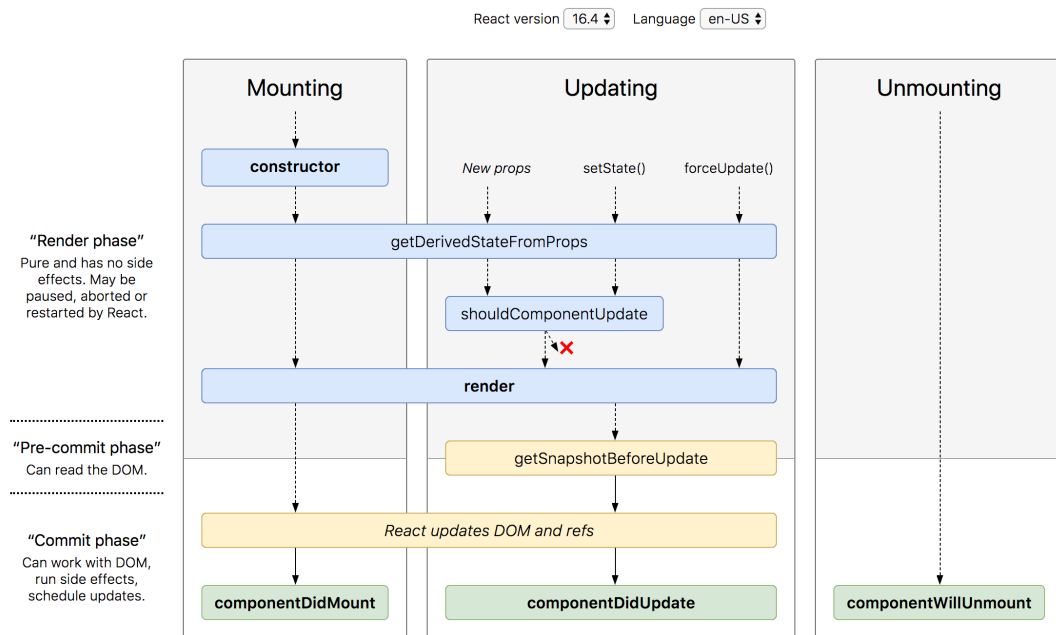
- Một component trong React có 2 cách để lấy thông tin props và state. Không như props, state của 1 component không được truyền từ bên ngoài vào. Một component sẽ tự quyết định state của chính nó. Để tạo ra state cho component, ta sẽ phải khai báo nó trong hàm constructor.

```
1  import React, { Component } from "react";
2
3  class Welcome extends Component {
4    constructor(props) {
5      super(props);
6      this.state = {
7        name: 'Tran Huu Trung'
8      };
9    }
10
11   render() {
12     return (
13       <h1>Hello, My name is {this.state.name}</h1>
14     );
15   }
16 }
17 export default Welcome;
```

Hình 1.7 Sử dụng State trong Component

- **This.state** ở đây là một đối tượng. Để truy cập đến các state ta sử dụng cú pháp: *this.state.name\_of\_property*. Và cách phổ biến để thay đổi state của một component là dùng hàm `setState()`.

## Lifecycle:



Hình 1.8 Sơ đồ vòng đời của Component trong ReactJS

- Một thành phần đặc trưng không kém Props và State của React đó là Lifecycle. Lifecycle là những phương thức được viết sẵn truy cập đến từng thời điểm trong một vòng đời của component, từ đó có thể triển khai ý tưởng một cách dễ dàng mà linh động.
- **constructor(props):** Hàm này là hàm khởi tạo, là hàm chạy ngay sau khi component được tạo ra, đầu vào của hàm này là props (tức là thuộc tính của component được dùng để truyền dữ liệu từ bên ngoài vào). Trong hàm này cũng có thể khởi tạo state cho component.
- **Render:** Là hàm bắt buộc, vì đây là đầu ra của component. Nó chịu trách nhiệm trả về một đối tượng JSX những gì muốn hiển thị ra bên ngoài.
- **componentDidMount:** Là hàm được gọi ngay sau khi giao diện được render xong, có nghĩa là khi giao diện hiện lên sẽ gọi hàm này. Trong hàm này có thể dùng để lấy dữ liệu từ api, server... và setState dữ liệu ra.
- **componentWillUnmount:** Hàm này được gọi khi component loại bỏ DOM, thực hiện các thao tác dọn dẹp, loại bỏ những phần tử dư thừa và hủy các timer...
- **componentDidUpdate(prevProps, prevState):** Hàm này được gọi ngay sau khi component được render lại.

#### *1.1.4.3. Ưu điểm và nhược điểm của ReactJS*

##### *Ưu điểm:*

- ReactJS cực kì hiệu quả: ReactJS tạo ra cho chính nó DOM ảo – nơi mà các component thực sự tồn tại trên đó. Điều này sẽ giúp cải thiện hiệu suất rất nhiều. ReactJS cũng tính toán những thay đổi nào cần cập nhật lên DOM và chỉ thực hiện chúng. Điều này giúp Reactjs tránh những thao tác cần trên DOM mà nhiều chi phí.
- ReactJS giúp việc viết các đoạn code JS dễ dàng hơn: Nó dùng cú pháp đặc biệt là JSX (Javascript mở rộng) cho phép ta trộn giữa code HTML và Javascript. Ta có thể thêm vào các đoạn HTML vào trong hàm render mà không cần phải nối chuỗi. Đây là đặc tính thú vị của ReactJS. Nó sẽ chuyển đổi các đoạn HTML thành các hàm khởi tạo đối tượng HTML bằng bộ biến đổi JSX.
- ReactJS có nhiều công cụ phát triển: Khi bắt đầu ReactJS, đừng quên cài đặt ứng dụng mở rộng của Chrome dành cho ReactJS. Nó giúp sửa lỗi chương trình dễ dàng hơn. Sau khi cài đặt ứng dụng này, người dùng sẽ có cái nhìn trực tiếp vào virtual DOM như thể đang xem cây DOM thông thường.
- Render tầng server: Một trong những vấn đề với các ứng dụng đơn trang là tối ưu SEO và thời gian tải trang. Nếu tất cả việc xây dựng và hiển thị trang đều thực hiện ở client, thì người dùng sẽ phải chờ cho trang được khởi tạo và hiển thị lên. Điều này thực tế là chậm. Hoặc nếu giả sử người dùng vô hiệu hóa Javascript thì sao? ReactJS là một thư viện component, nó có thể vừa render ở ngoài trình duyệt sử dụng DOM và cũng có thể render bằng các chuỗi HTML mà server trả về.
- Làm việc với vấn đề test giao diện: Nó cực kì dễ để viết các test case giao diện vì virtual DOM được cài đặt hoàn toàn bằng JS.
- Hiệu năng cao đối với các ứng dụng có dữ liệu thay đổi liên tục, dễ dàng cho bảo trì và sửa lỗi.

##### *Nhược điểm:*

- Reactjs chỉ phục vụ cho tầng View. React chỉ là View Library nó không phải là một MVC framework như những framework khác. Đây chỉ là thư viện của Facebook giúp render ra phần view. Vì thế React sẽ không có phần Model và Controller, mà phải kết hợp với các thư viện khác. React cũng sẽ không có 2-way binding hay là Ajax.
- Tích hợp Reactjs vào các framework MVC truyền thống yêu cầu cần phải cấu hình lại.

- React khá nặng nề so với các framework khác, React có kích thước tương đương với Angular (Khoảng 35kb so với 39kb của Angular). Trong khi đó Angular là một framework hoàn chỉnh.

#### **1.1.5. Tổng quan về Microsoft SQL Server**

SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) được phát triển và tiếp thị bởi Microsoft. Là một máy chủ cơ sở dữ liệu, chức năng chính của SQL Server là lưu trữ và truy xuất dữ liệu được sử dụng bởi các ứng dụng khác.

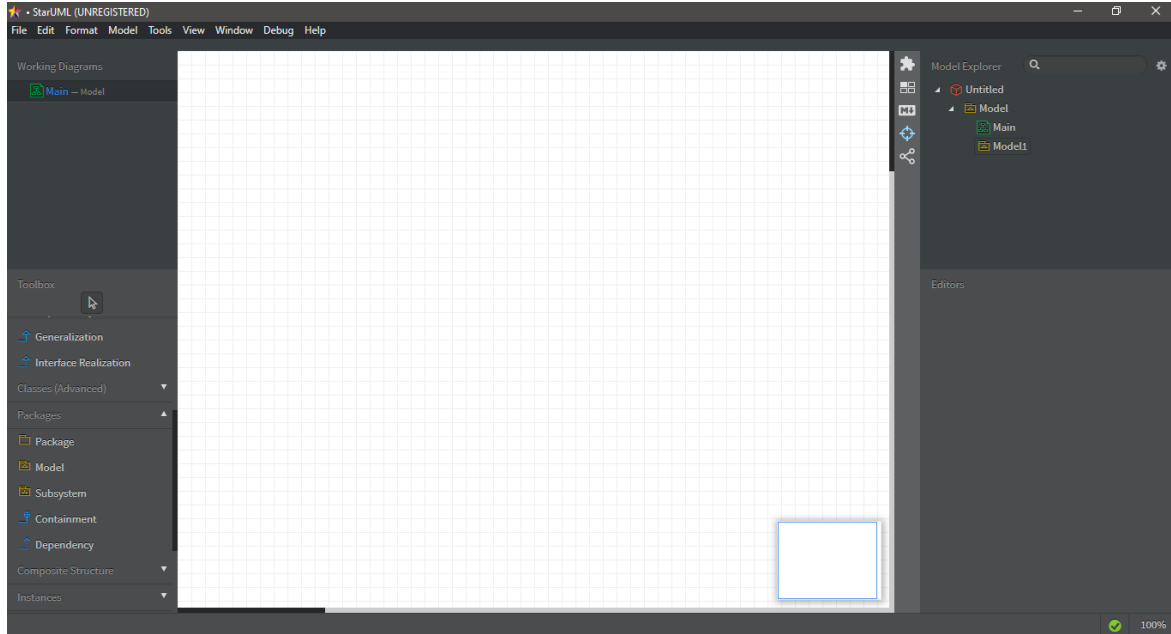
Cơ sở dữ liệu là tập hợp dữ liệu theo cùng một cấu trúc. Giống như việc chụp hình: nhấn nút chụp ảnh, hình ảnh là dữ liệu, thư viện lưu ảnh là cơ sở dữ liệu. Cơ sở dữ liệu, hay database, là nơi chứa và sắp đặt dữ liệu. Dữ liệu được đặt trong một bộ dữ liệu chung, dataset, được tổ chức sắp xếp giống như một bảng tính chung. Mỗi “bảng” có liên hệ với nhau theo cách nào đó. Vì vậy từ Relational (liên hệ) trong RDBMS có ý nghĩa như vậy. Nếu phần mềm không hỗ trợ mô hình dữ liệu quan hệ với nhau thì như vậy gọi là DBMS.



Hình 1.9 Tổng quan về SQL Server

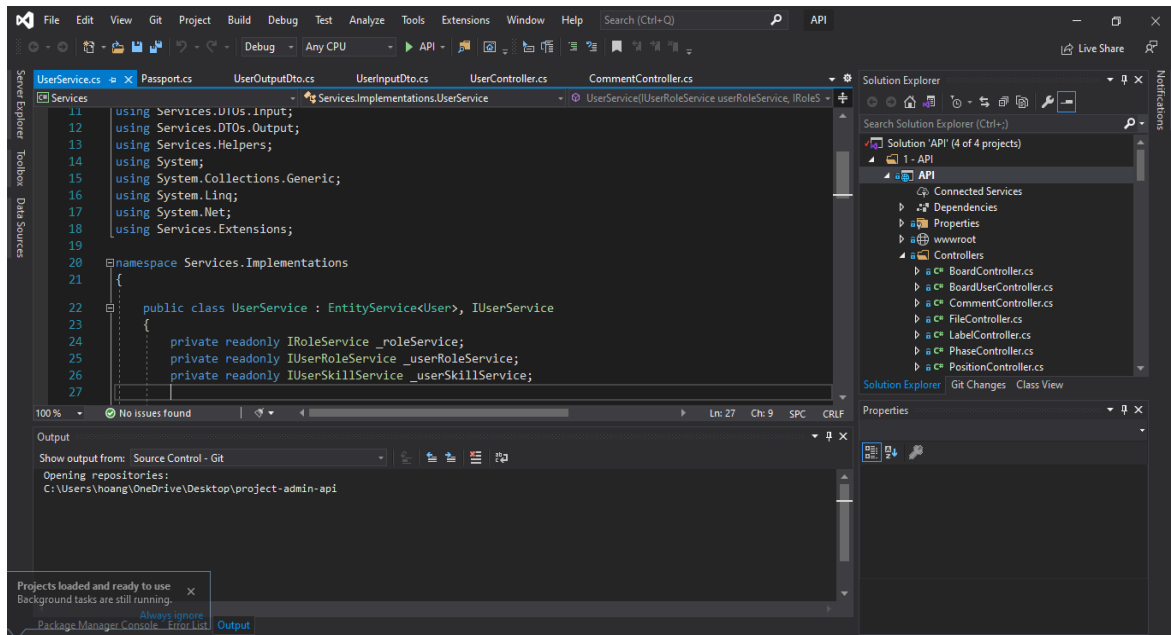
### 1.1.6. Các công cụ sử dụng

Quá trình phân tích và thiết kế UML thì công cụ **StarUML** được sử dụng. Công cụ này giúp có thể dễ dàng thao tác và sử dụng để tạo các sơ đồ ca sử dụng, sơ đồ hoạt động,....



Hình 1.10 Giao diện chính của StarUML

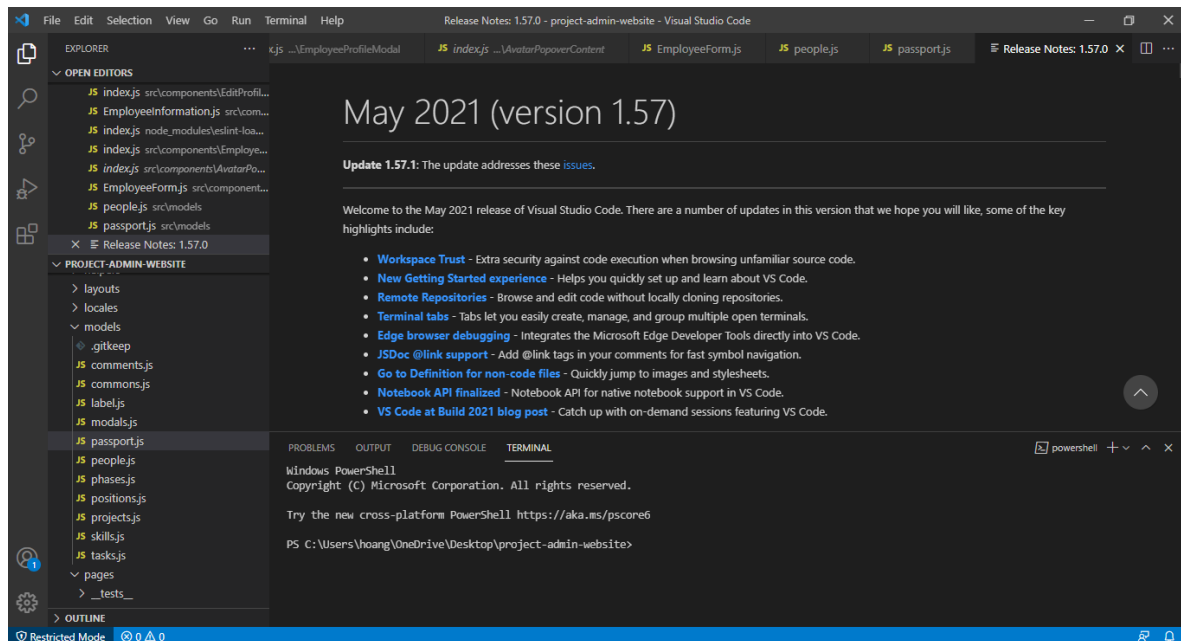
Quá trình xây dựng mã chương trình phía Back-end thì công cụ editor được dùng chính là Visual Studio 2019. Công cụ này do Microsoft phát triển hỗ trợ tối đa cho việc xây dựng các project ASP.NET.



Hình 1.11 Giao diện chính của Visual Studio 2019

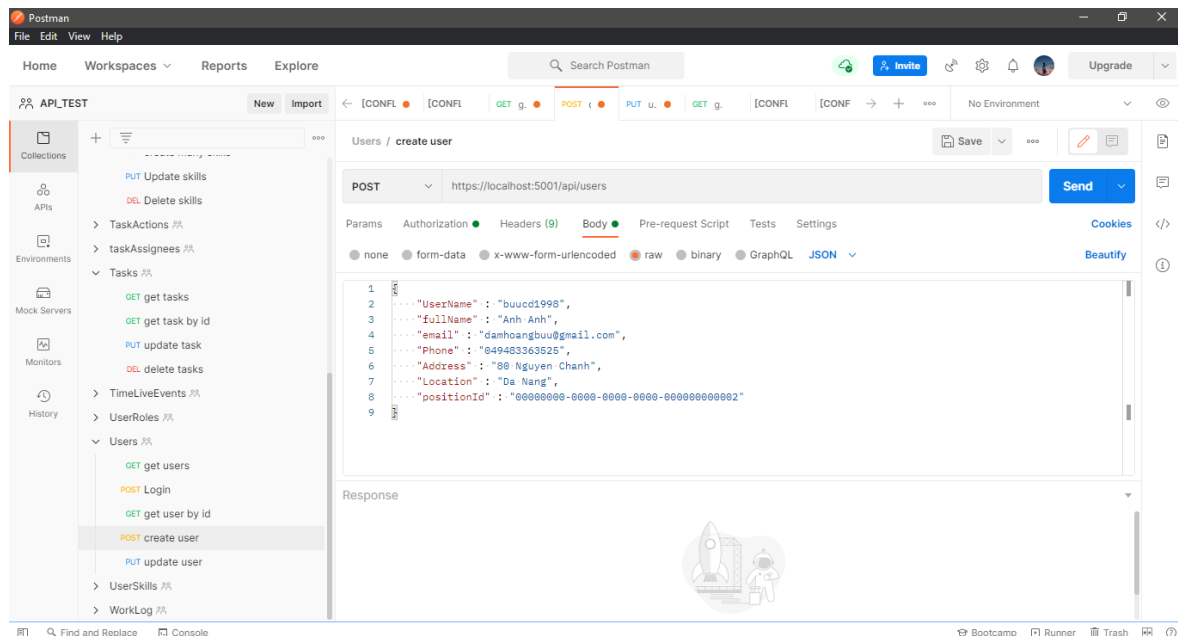


Quá trình xây dựng mã chương trình phía Front-end thì công cụ sử dụng là VSCode. Đặc điểm nổi bật là đơn giản, dễ dàng cài đặt, hỗ trợ nhiều hệ điều hành cũng như nhiều ngôn ngữ lập trình khác nhau.



Hình 1.12 Giao diện VSCode

Quá trình kiểm tra, test và chạy thử các API đã được viết chương trình trước khi áp dụng để gọi từ các request ở website, công cụ được sử dụng là Postman.



Hình 1.13 Giao diện của Postman

## **1.2. Kết chương**

Chương này cung cấp cho người đọc cái nhìn tổng quan về các công nghệ và công cụ để xây dựng hệ thống quản lí, giám sát tiến độ công việc và dự án tại công ty.

## CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

### 2.1. Phân tích yêu cầu

#### 2.1.1. Bài toán thực tế

Hiện nay bài toán tối ưu hóa quá trình quản lý dự án là vấn đề đang được công ty quan tâm nhiều. Hầu hết các dự án hiện tại của công ty đều đang vận hành theo mô hình scrum-agile nên yêu cầu một hệ thống cần đáp ứng việc tạo mới các dự án, quản lý các dự án theo mô hình này bằng các cột tiến trình. Trên thị trường hiện nay, tuy các hệ thống hỗ trợ cho việc quản lý dự án không còn xa lạ gì nhưng vẫn còn khá nhiều khuyết điểm như sau:

- Không phù hợp với qui trình làm việc của công ty.
- Khuyết thiếu một số chức năng cần thiết.
- Chi phí cao.

Để giải quyết bài toán này thì ý tưởng xây dựng hệ thống đã ra đời.

#### 2.1.2. Bài toán đề xuất

Hệ thống chia thành 2 nhóm người dung chính: Quản trị viên (admin) và nhân viên công ty (user).

Đối với nhân viên công ty (user) thì có những nhóm chức năng chính như sau:

- **Quản lý thông tin của bản thân:** user được quyền xem, sửa, thêm các thông tin cá nhân của mình.
- **Quản lý dự án:** user xem một dự án mà mình đã tham gia, tạo mới một dự án, thêm thành viên vào dự án mà mình đã tạo, chỉnh sửa thông tin trong dự án, thêm các nhãn màu trong dự án, xem thông tin của các thành viên trong dự án.
- **Quản lý thời gian biểu của bản thân:** nhân viên có thể quản lý, kiểm soát thời gian làm việc của mình thông qua việc lưu lại thời gian đã tiêu tốn cho từng công việc được giao, có thể xem toàn bộ thời gian làm việc của mình, có thể xem thông tin của công việc đã làm ngay tại thời gian biểu.
- **Quản lý công việc:** nhân viên có thể tạo mới các tiến trình trong dự án mình đã tham gia, giao và nhận công việc trong dự án, quan sát thời gian làm việc của mỗi tiến trình trong dự án.

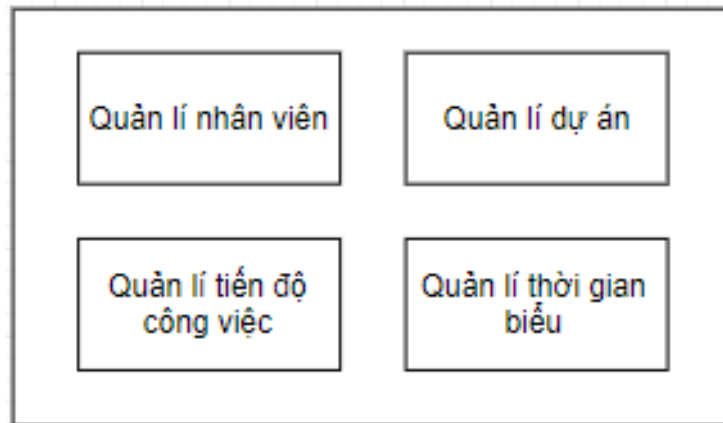
Đối với quản trị viên thì có những nhóm chức năng chính sau:

- **Quản lý thông tin của tất cả nhân viên:** có thể thêm tài khoản nhân viên mới, xem, chỉnh sửa thông tin của tất cả nhân viên.
- **Quản lý dự án:** được quyền giám sát tất cả các dự án.
- **Quản lý thời gian làm việc:** được quyền quan sát thời gian làm việc của mọi nhân viên.

## 2.2. Phân tích chức năng

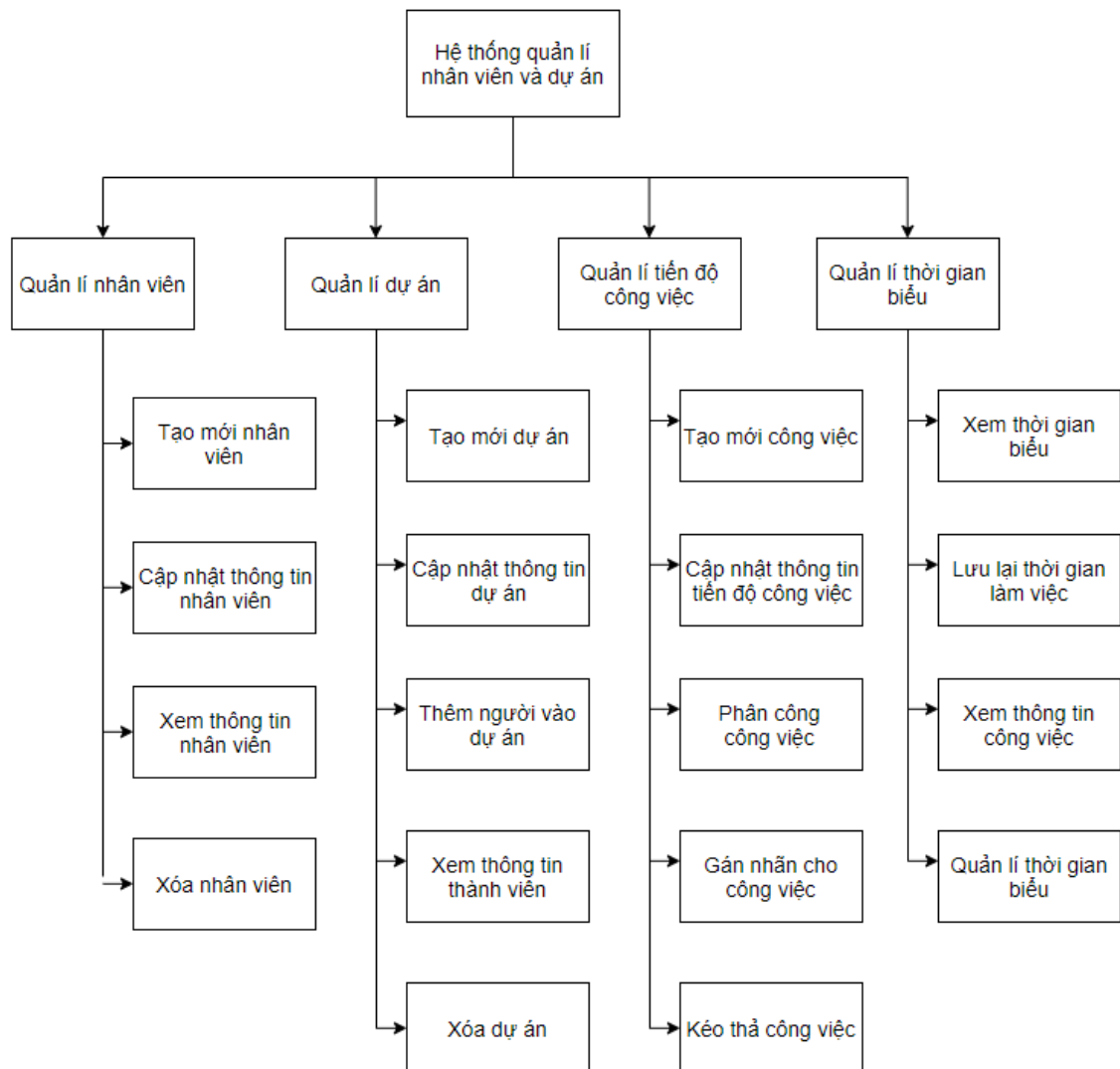
### 2.2.1. Các module chính của hệ thống

Hệ thống được xây dựng gồm 4 module chính: quản lý nhân viên, quản lý dự án, quản lý công việc và quản lý thời gian làm việc.



Hình 2.1 Các module chính của dự án

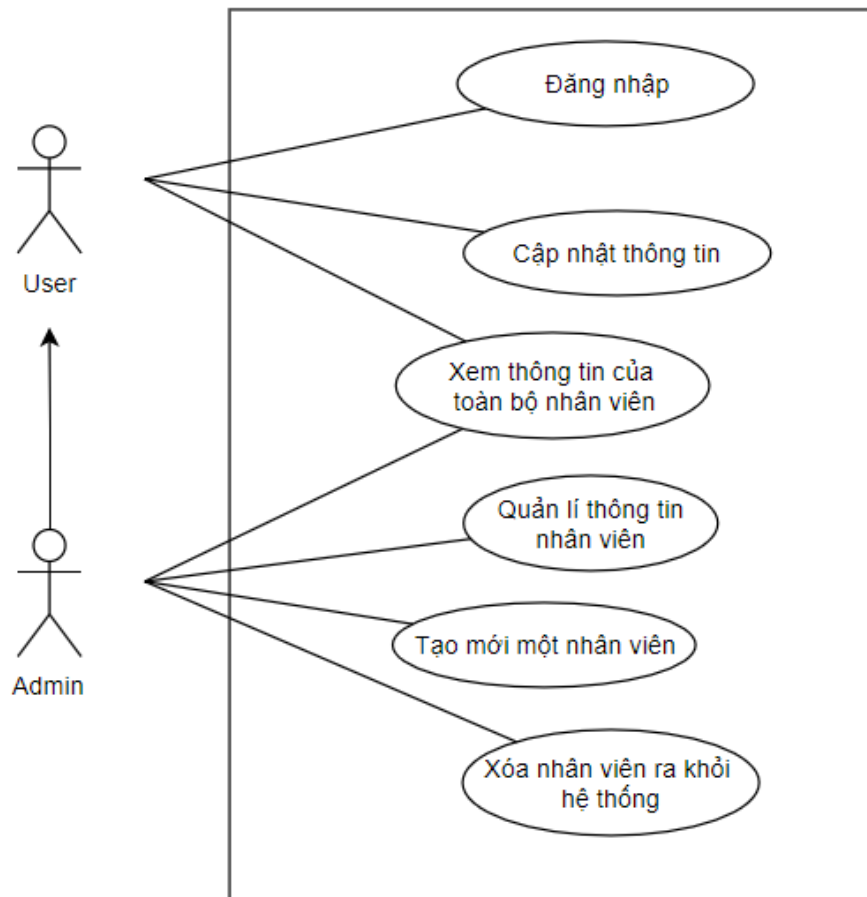
### 2.2.2. Sơ đồ cây phân rã chức năng



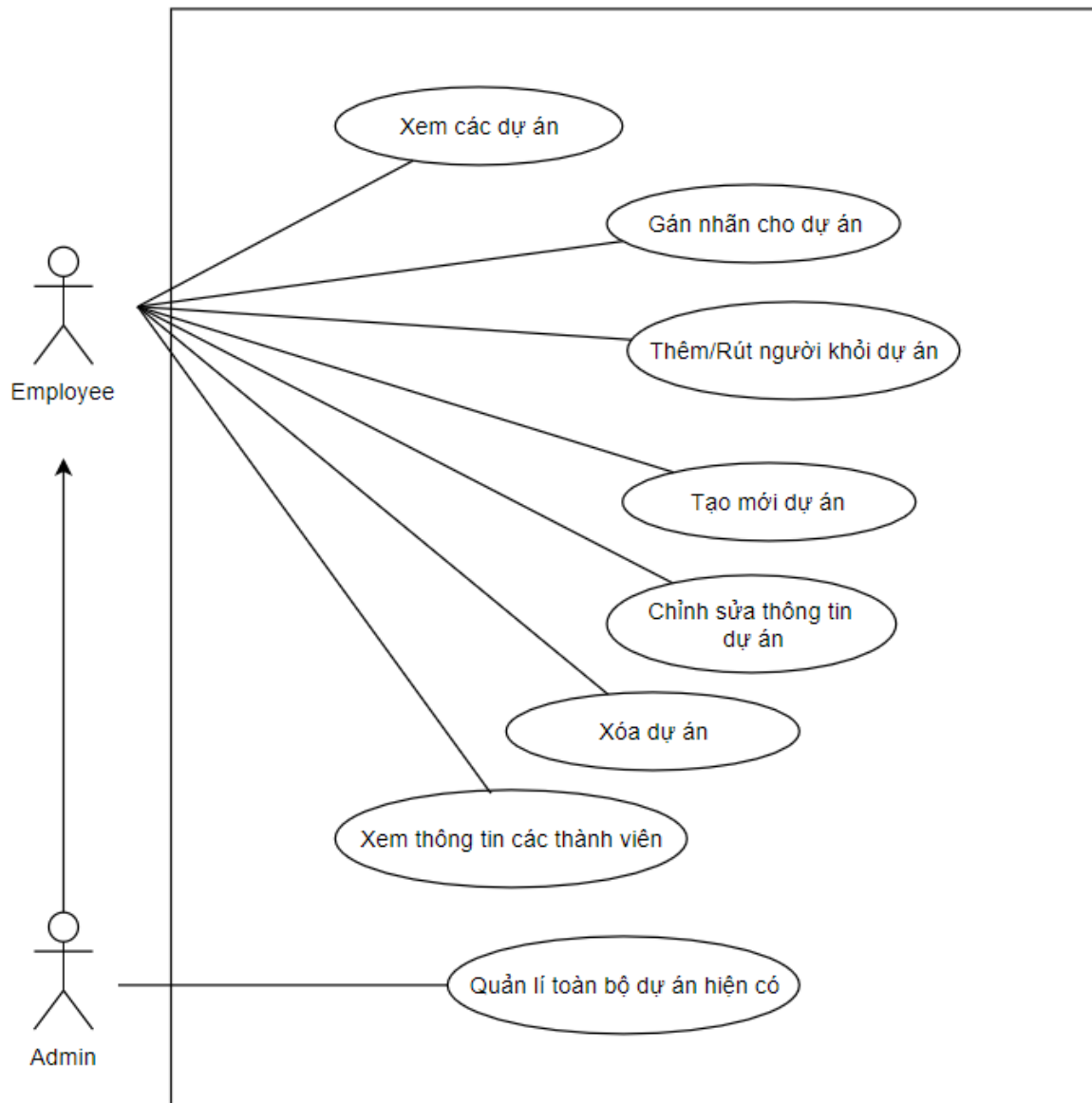
Hình 2.2 Sơ đồ phân rã chức năng

### 2.2.3. Biểu đồ ca sử dụng

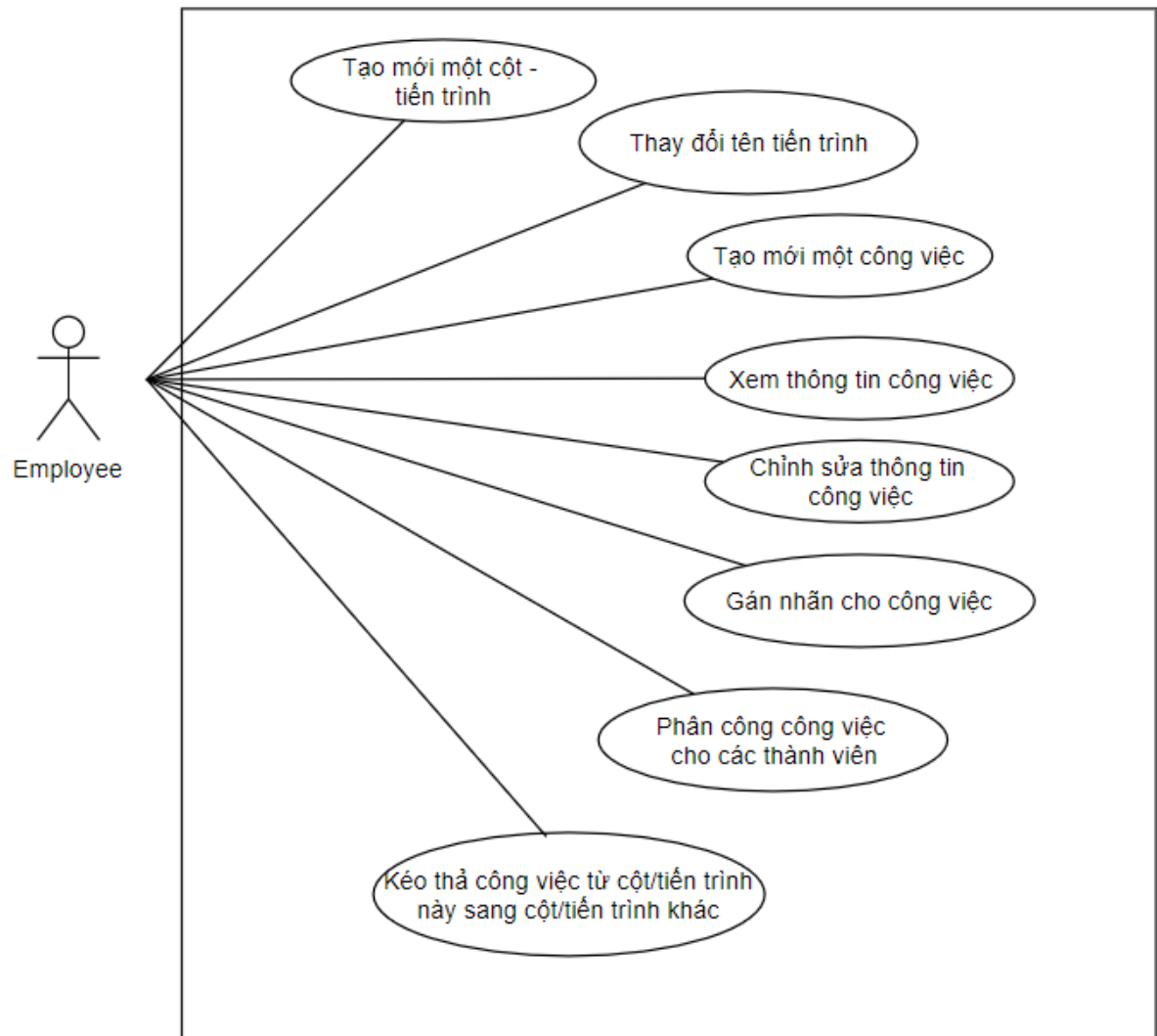
#### 2.2.3.1. Các biểu đồ ca sử dụng của hệ thống



Hình 2.3 Biểu đồ usecase cho module quản lý nhân viên

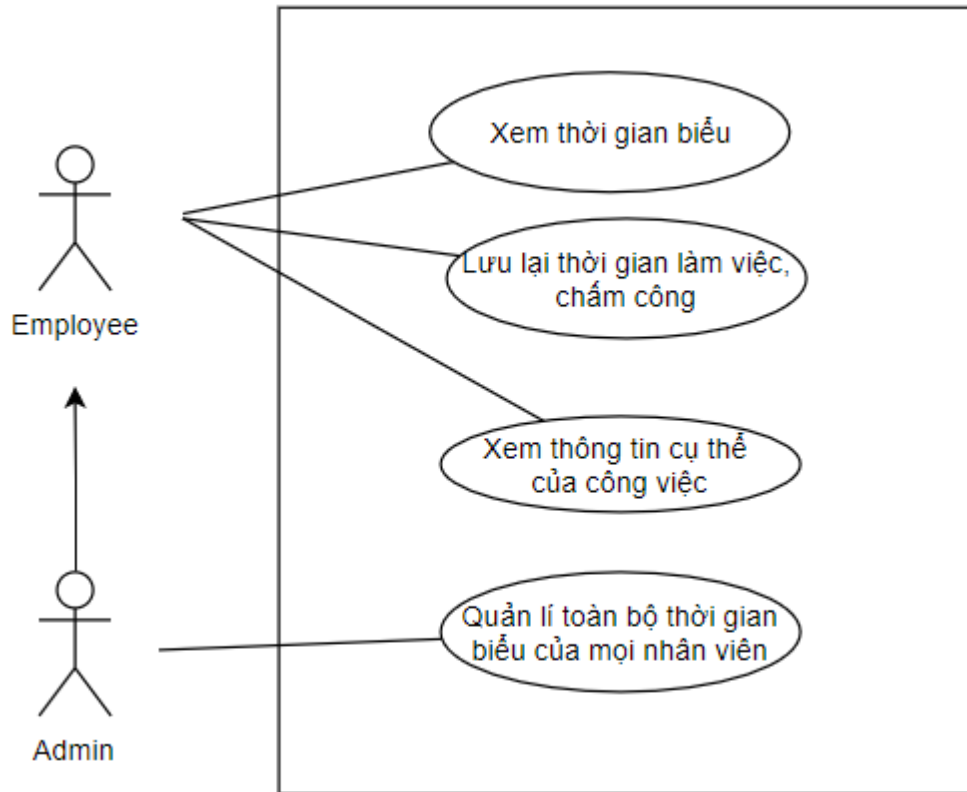


Hình 2.4 Biểu đồ usecase cho module quản lí dự án



Hình 2.5 Biểu đồ usecase cho module quản lý tiến độ công việc trong dự án





Hình 2.6 Biểu đồ usecase module quản lý thời gian làm việc

### 2.2.3.2. Đặc tả ca sử dụng

Bảng 2.1 Đặc tả ca sử dụng đăng nhập

<b>Mã ca sử dụng</b>	UC-01		
<b>Tác nhân</b>	Nhân viên		
<b>Mô tả</b>	Cho phép người dùng truy cập vào hệ thống		
<b>Điều kiện trước</b>	Vào website		
<b>Điều kiện sau</b>	Đăng nhập vào hệ thống		
<b>Kịch bản</b>	<b>STT</b>	<b>Tác nhân</b>	<b>Hệ thống phản hồi</b>
	1	Vào website	
	2		Hiển thị giao diện đăng nhập
	3	Nhập vào username và password	
	4	Nhấn nút đăng nhập hoặc enter	
	5		Hệ thống xác nhận thông tin gửi lên CSDL. Nếu hợp lệ,

			hệ thống tiếp nhận thông tin, và truy cập vào trang chủ
<b>Luồng thay thế và các ngoại lệ</b>			Nếu nhập dữ liệu không hợp lệ, hệ thống sẽ thông báo và yêu cầu nhập lại

**Bảng 2.2** Đặc tả ca sử dụng cập nhật thông tin

<b>Mã ca sử dụng</b>	UC-02		
<b>Tác nhân</b>	Nhân viên		
<b>Mô tả</b>	Cho phép chỉnh sửa thông tin cá nhân		
<b>Điều kiện trước</b>	Đã đăng nhập vào hệ thống		
<b>Điều kiện sau</b>	Cập nhật thông tin thành công		
<b>Kịch bản</b>	<b>STT</b>	<b>Tác nhân</b>	<b>Hệ thống phản hồi</b>
	1	Nhấn vào hình đại diện của người dùng ở góc trên phải	
	2		Hiện thị drop list
	3	Nhấn vào profile	
	4		Hiện thị profile của nhân viên
	5	Nhấn vào button biểu tượng edit trong trang profile	
	6		Hiện thị trang edit thông tin
	7	Nhập lại những dữ liệu cần cập nhật	
	8	Nhấn OK	
	9		Hệ thống xác nhận thông tin gửi lên CSDL
<b>Luồng thay thế và ngoại lệ</b>			Nếu nhập dữ liệu không hợp lệ, hệ thống sẽ thông báo lỗi và yêu cầu nhập lại

Bảng 2.3 Đặc tả ca sử dụng thêm mới nhân viên

<b>Mã ca sử dụng</b>	UC-03		
<b>Tác nhân</b>	Quản trị viên		
<b>Mô tả</b>	Đăng ký tài khoản mới (thêm mới nhân viên)		
<b>Điều kiện trước</b>	Đã đăng nhập vào hệ thống		
<b>Điều kiện sau</b>	Đăng ký tài khoản thành công		
<b>Kịch bản</b>	<b>STT</b>	<b>Tác nhân</b>	<b>Hệ thống phản hồi</b>
	1	Nhấn vào biểu tượng thêm mới ở góc dưới bên phải	
	2		Hiện thị giao diện thêm mới nhân viên
	3	Nhập thông tin nhân viên	
	4	Nhấn OK	
	5		Hệ thống xác nhận thông tin gửi lên CSDL
<b>Luồng thay thế và ngoại lệ</b>			Nếu nhập dữ liệu không hợp lệ, hệ thống sẽ thông báo lỗi và yêu cầu nhập lại

Bảng 2.4 Đặc tả ca sử dụng thêm mới dự án

<b>Mã ca sử dụng</b>	UC-04		
<b>Tác nhân</b>	Nhân viên		
<b>Mô tả</b>	Cho phép nhân viên tạo một dự án mới		
<b>Điều kiện trước</b>	Đã đăng nhập vào hệ thống và truy cập vào trang quản lý dự án		
<b>Điều kiện sau</b>	Thêm mới dự án thành công		
<b>Kịch bản</b>	<b>STT</b>	<b>Tác nhân</b>	<b>Hệ thống phản hồi</b>
	1	Nhấn vào button “new project” ở góc trên bên trái	
	2		Hiện thị giao diện thêm mới

			dự án
	3	Nhập thông tin dự án	
	4	Nhấn Create	
	5		Hệ thống xác nhận thông tin gửi lên CSDL
<b>Luồng thay thế và ngoại lệ</b>			Nếu nhập dữ liệu không hợp lệ, hệ thống sẽ thông báo lỗi và yêu cầu nhập lại

**Bảng 2.5** Đặc tả ca sử dụng thêm mới tiến trình

<b>Mã ca sử dụng</b>	UC-05		
<b>Tác nhân</b>	Nhân viên		
<b>Mô tả</b>	Cho phép nhân viên thêm mới một tiến trình		
<b>Điều kiện trước</b>	Đã đăng nhập vào hệ thống và truy cập vào trang quản lí dự án		
<b>Điều kiện sau</b>	Thêm mới tiến trình thành công		
<b>Kịch bản</b>	<b>STT</b>	<b>Tác nhân</b>	<b>Hệ thống phản hồi</b>
	1	Nhấn vào button “create another column”	
	2		Hiện thị textbox để nhập tên tiến trình
	3	Nhập tên tiến trình	
	4	Nhấn Enter	
	5		Hệ thống xác nhận thông tin gửi lên CSDL
<b>Luồng thay thế và ngoại lệ</b>			Nếu nhập dữ liệu không hợp lệ, hệ thống sẽ thông báo lỗi và yêu cầu nhập lại

**Bảng 2.6** Đặc tả ca sử dụng thêm mới công việc

<b>Mã ca sử dụng</b>	UC-06
<b>Tác nhân</b>	Nhân viên
<b>Mô tả</b>	Cho phép nhân viên thêm mới một công việc

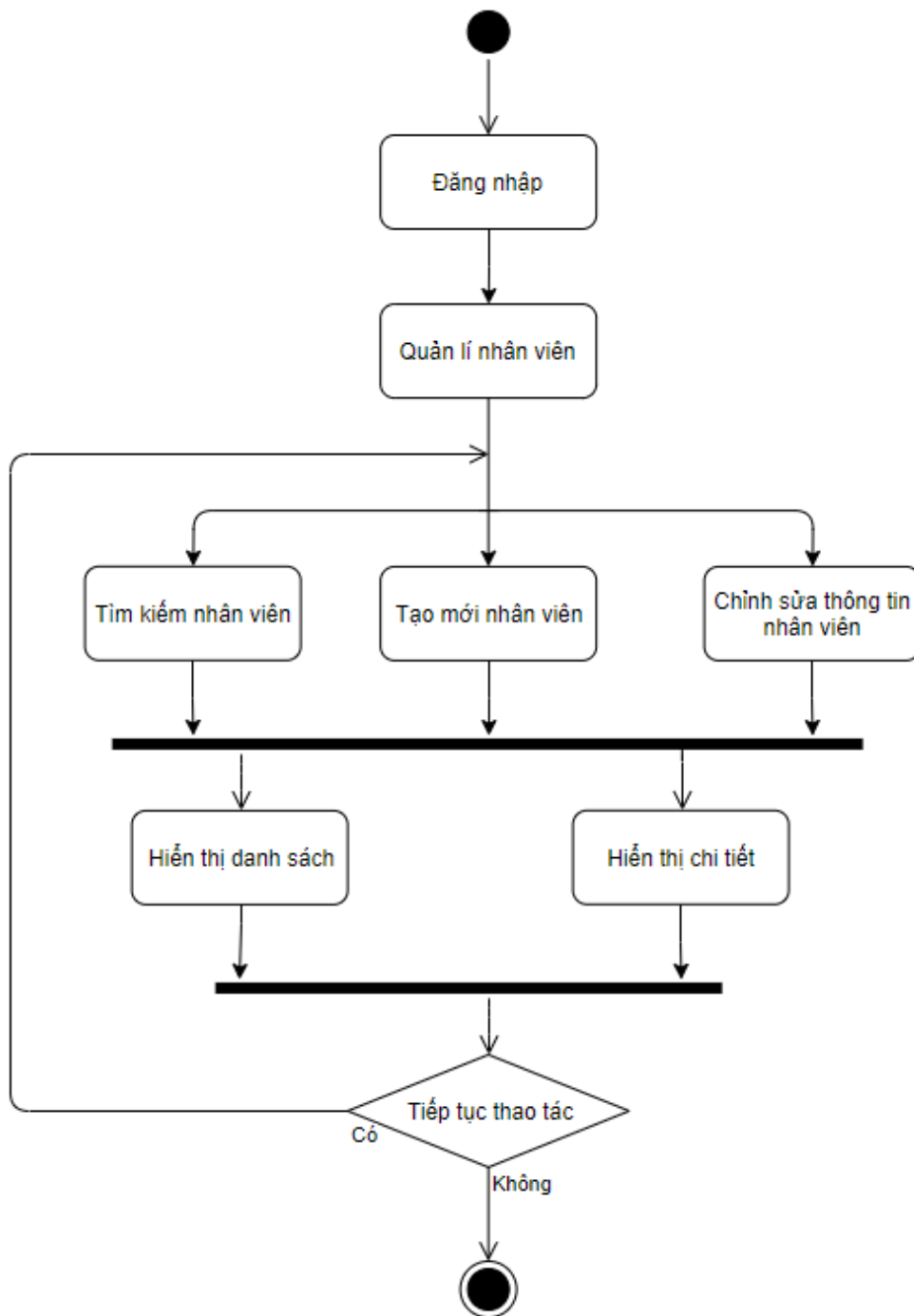
<b>Điều kiện trước</b>	Đã đăng nhập vào hệ thống và truy cập vào trang quản lý dự án		
<b>Điều kiện sau</b>	Thêm mới công việc thành công		
<b>Kịch bản</b>	<b>STT</b>	<b>Tác nhân</b>	<b>Hệ thống phản hồi</b>
	1	Nhấn vào button biểu tượng (+) ở mỗi tiến trình	
	2		Hiện thị textbox để nhập tên công việc
	3	Nhập tên công việc	
	4	Nhấn Create	
	5		Hệ thống xác nhận thông tin gửi lên CSDL
<b>Luồng thay thế và ngoại lệ</b>			Nếu nhập dữ liệu không hợp lệ, hệ thống sẽ thông báo lỗi và yêu cầu nhập lại

Bảng 2.7 Đặc tả ca sử dụng quản lý công việc

<b>Mã ca sử dụng</b>	UC-07		
<b>Tác nhân</b>	Nhân viên		
<b>Mô tả</b>	Cho phép nhân viên quản lý công việc		
<b>Điều kiện trước</b>	Đã đăng nhập vào hệ thống và truy cập vào trang quản lý dự án		
<b>Điều kiện sau</b>	Thêm mới công việc thành công		
<b>Kịch bản</b>	<b>STT</b>	<b>Tác nhân</b>	<b>Hệ thống phản hồi</b>
	1	Nhấn vào công việc cần quản lý	
	2		Hiển thị giao diện quản lý công việc
	3	Thêm mô tả, bình luận, gắn nhãn hoặc thành viên vào công việc	
	4	Nhấn button (x) để thoát khỏi giao diện	
	5		Hệ thống xác nhận thông tin gửi lên CSDL
<b>Luồng thay thế và ngoại lệ</b>			Nếu nhập dữ liệu không hợp lệ, hệ thống sẽ thông báo lỗi và yêu cầu nhập lại

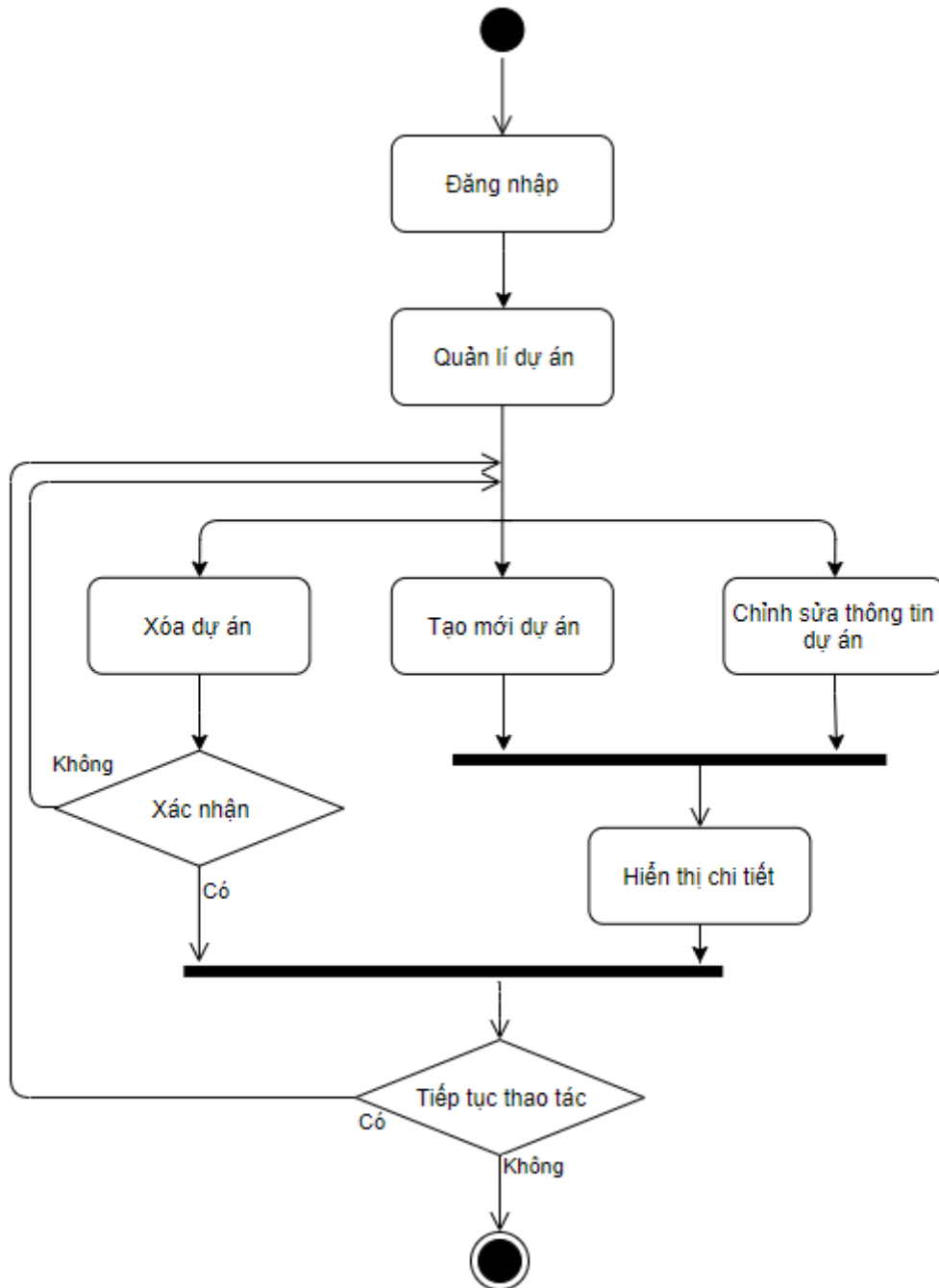


### 2.2.5. Biểu đồ hoạt động

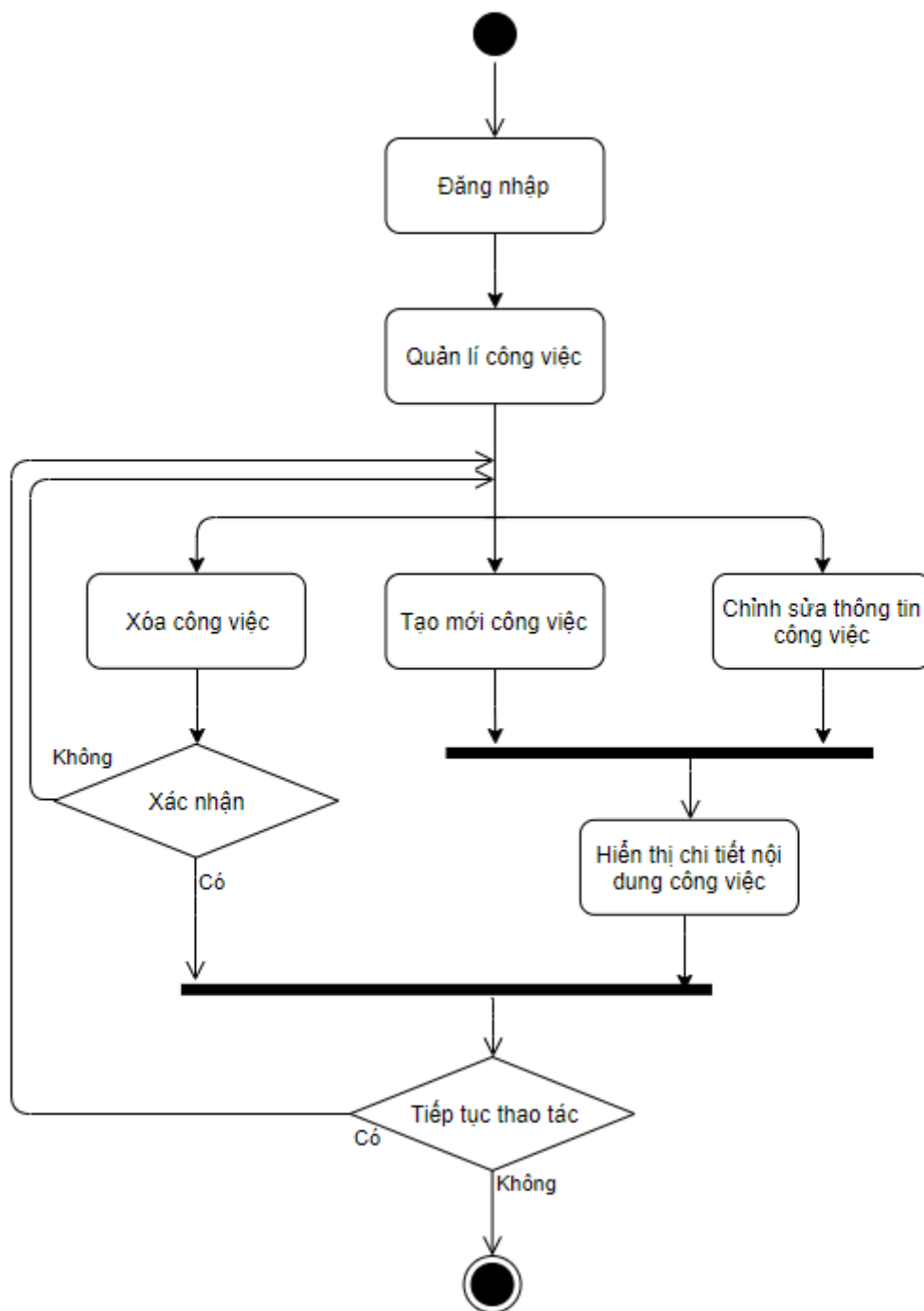


Hình 2.8 Biểu đồ hoạt động của module quản lý nhân viên

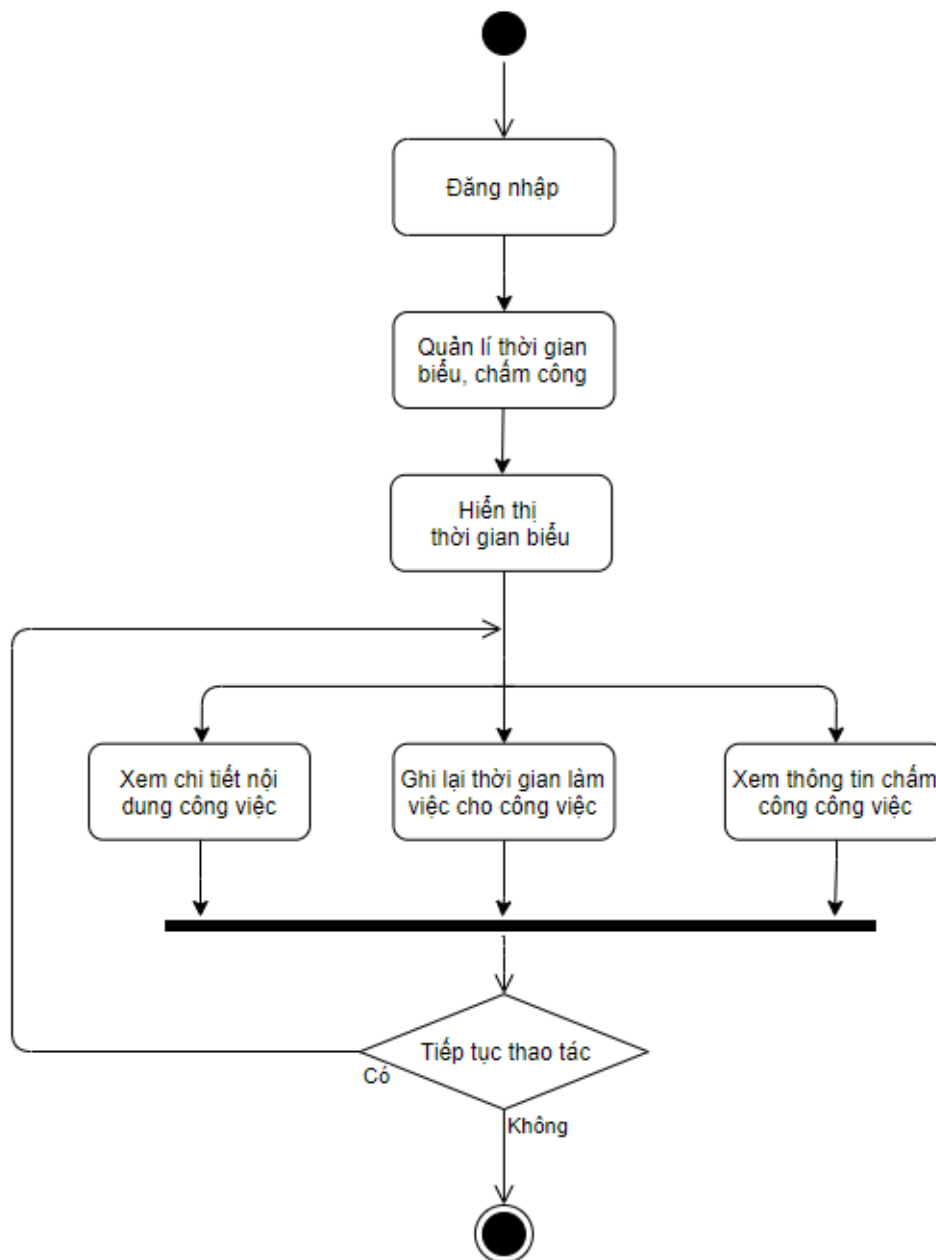




Hình 2.9 Biểu đồ hoạt động của module quản lí dự án



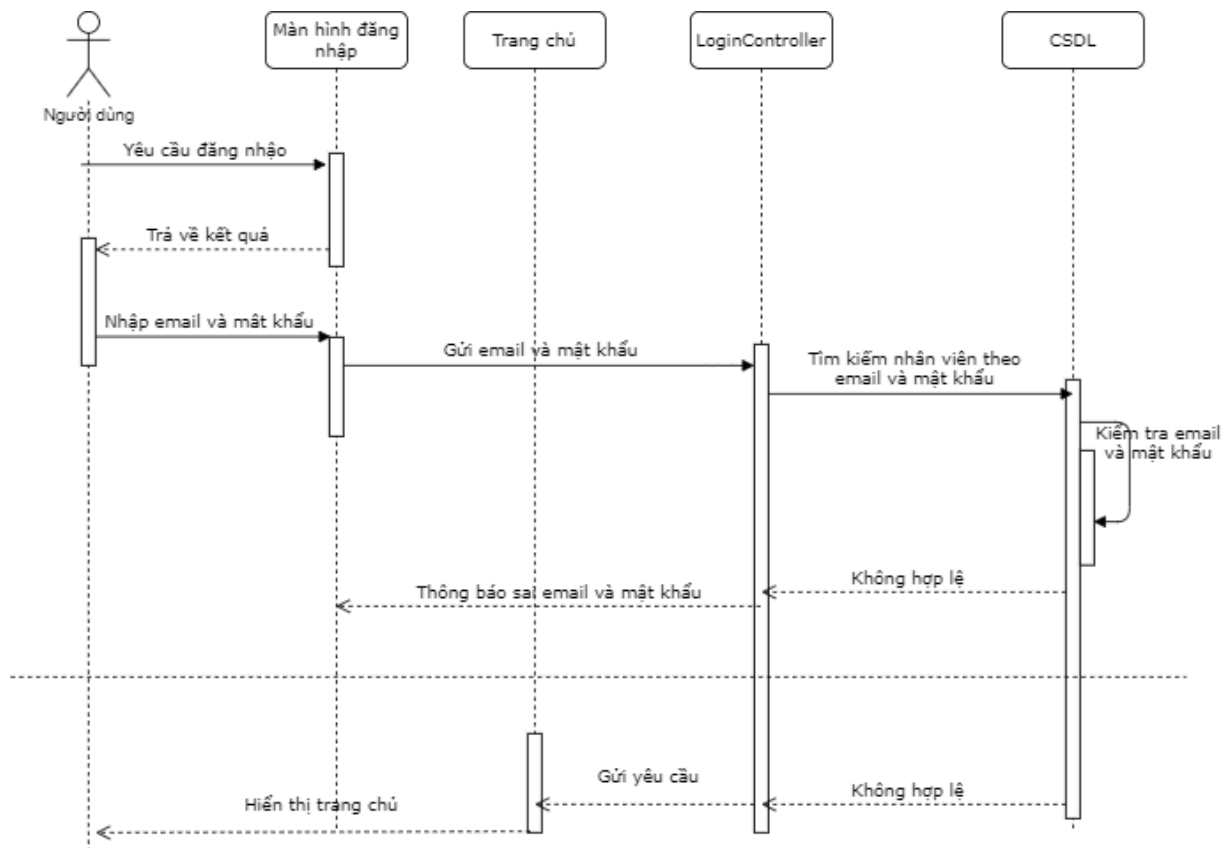
Hình 2.10 Biểu đồ hoạt động của module quản lý tiến độ công việc



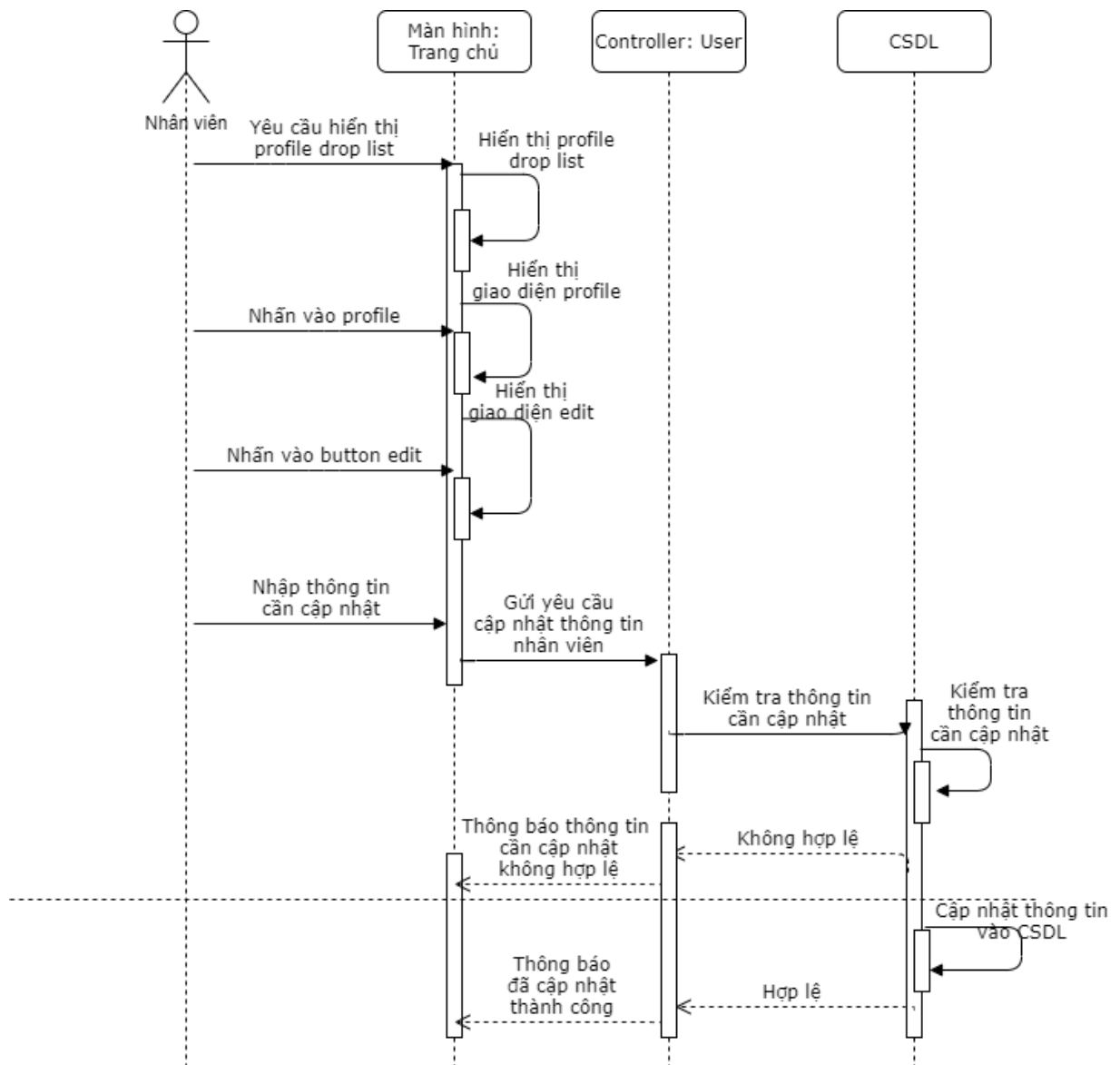
Hình 2.11 Biểu đồ hoạt động của module quản lý thời gian biểu, chấm công

### 2.2.6. Biểu đồ tuần tự

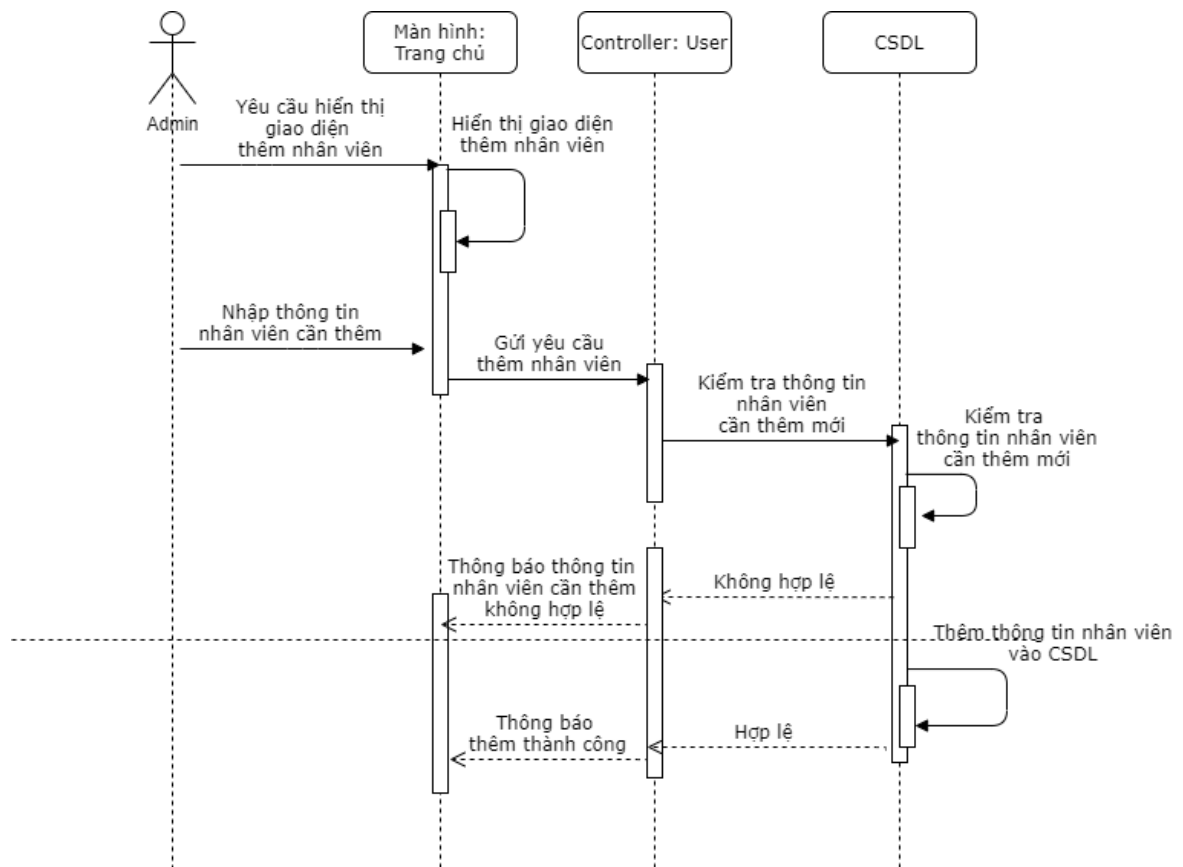
Một số biểu đồ tuần tự tổng quát sau sẽ cho thấy rõ hơn tự tương tác cơ bản giữa các thành phần của hệ thống trong một số chức năng quan trọng.



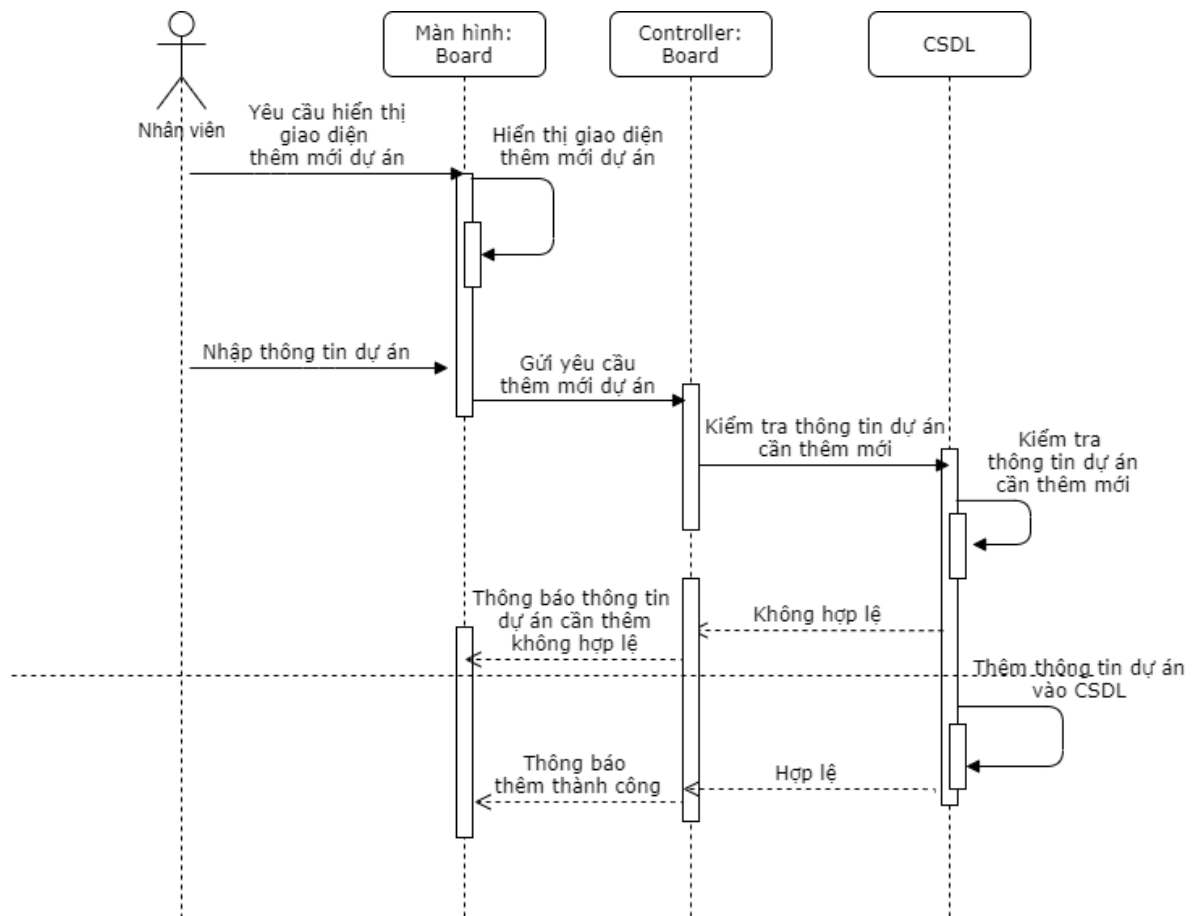
Hình 2.12 Biểu đồ tuần tự quá trình đăng nhập



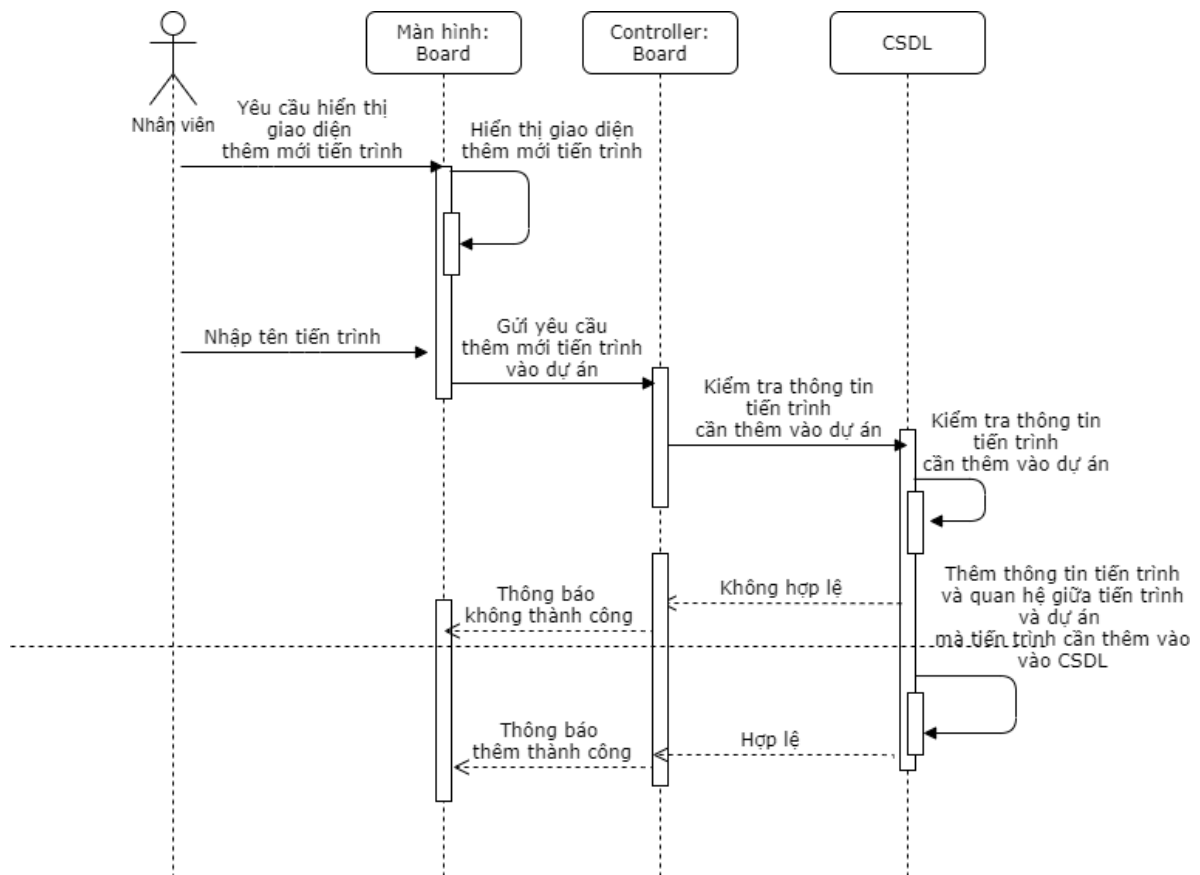
Hình 2.13 Biểu đồ tuần tự quá trình cập nhật thông tin nhân viên



Hình 2.14 Biểu đồ tuần tự quá trình thêm nhân viên

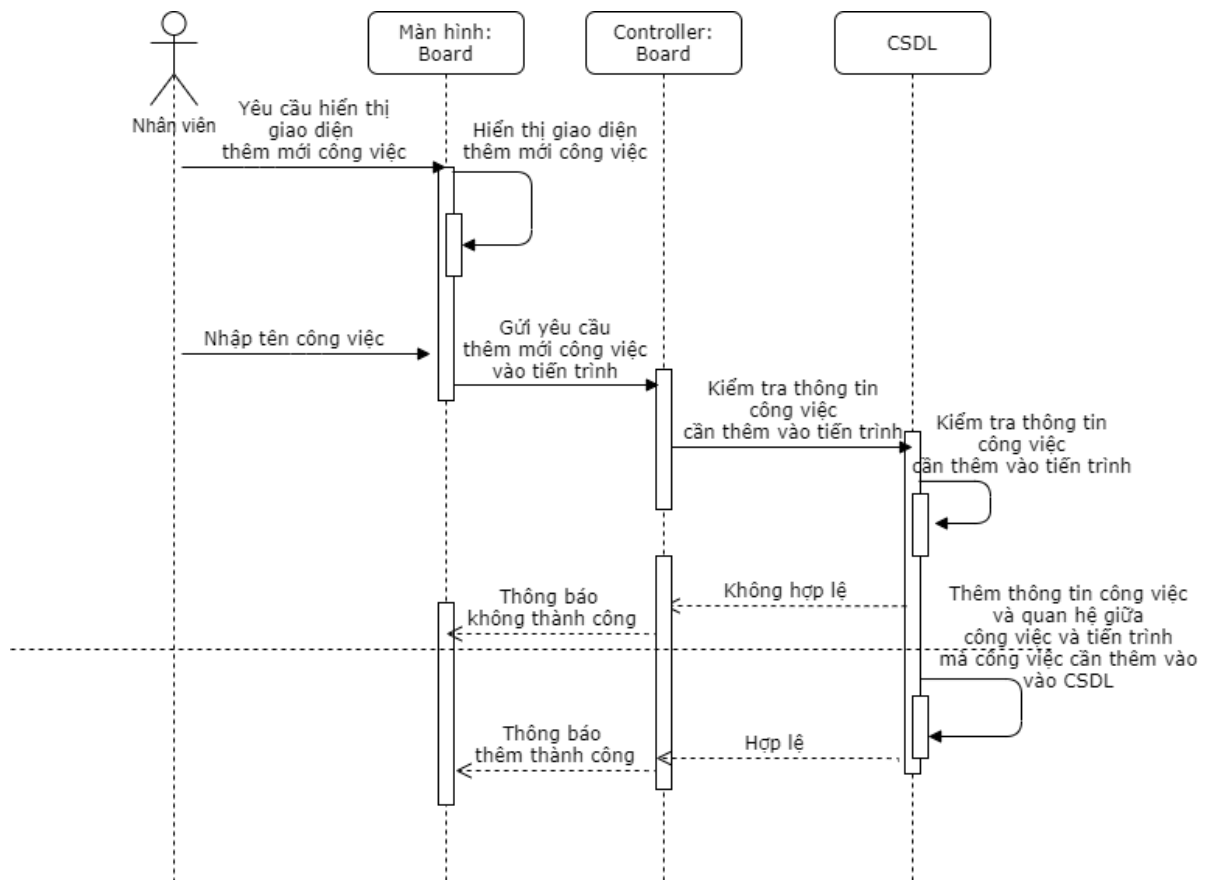


Hình 2.15 Biểu đồ tuần tự quá trình thêm mới dự án



Hình 2.16 Biểu đồ tuần tự thêm mới tiến trình

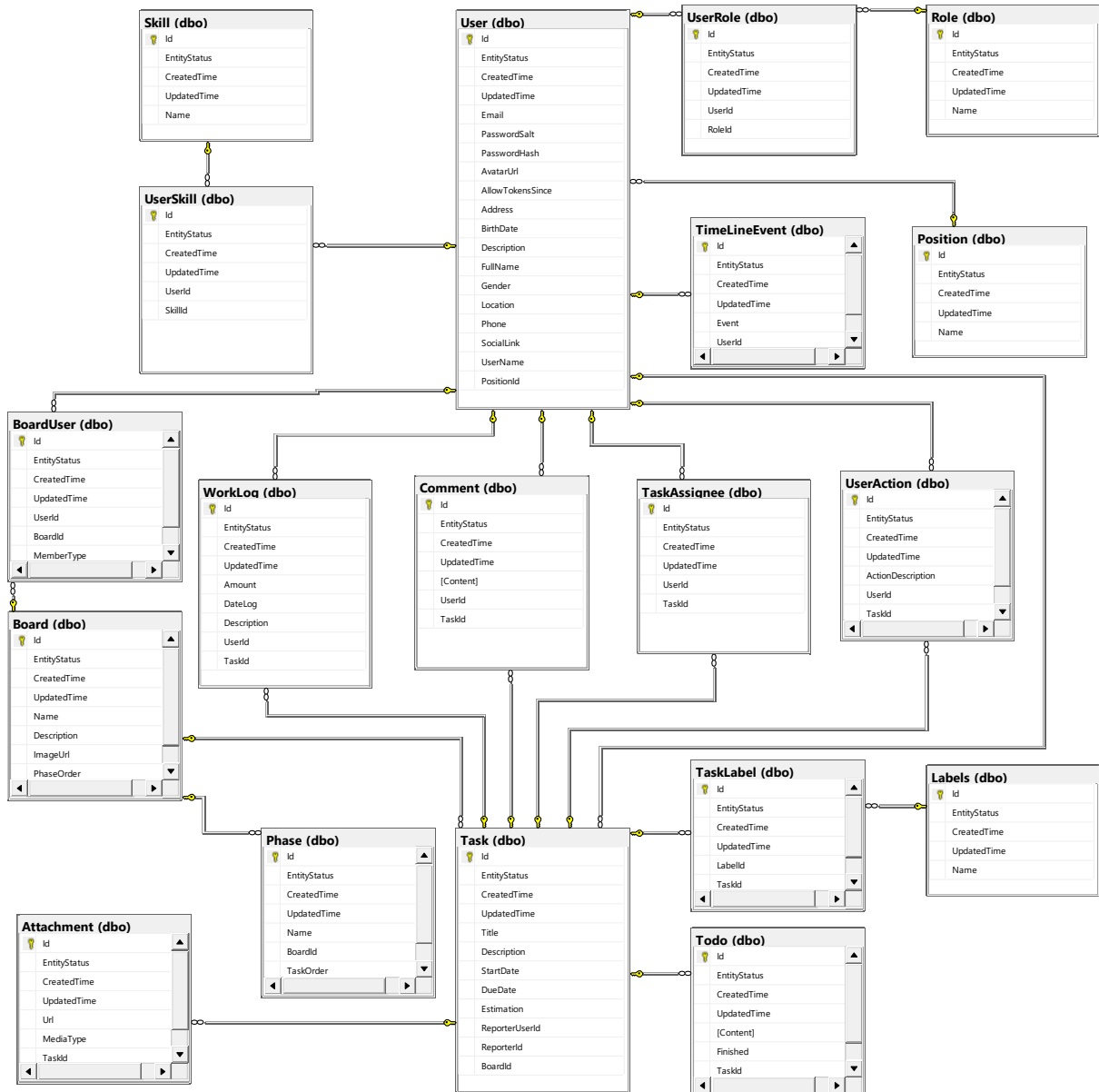




Hình 2.17 Biểu đồ tuần tự thêm mới công việc

## 2.3. Thiết kế cơ sở dữ liệu

### 2.3.1. Sơ đồ quan hệ, thực thể dữ liệu



Hình 2.18 Sơ đồ quan hệ, thực thể dữ liệu

### 2.3.2. Đặc tả các bảng dữ liệu

Bảng 2.8 Bảng User

Tên	Mô tả	Data Type	Note
<b>Id</b>	Id của bảng User	Char	Khóa chính
EntityStatus	Trạng thái của một thực thể	Integer	
CreatedTime	Ngày khởi tạo	Date Time	
UpdatedTime	Ngày chỉnh sửa cuối cùng	Date Time	
Email	Địa chỉ Email của nhân viên	Text	
PasswordHash	Mã hóa hash của password	Text	
PasswordSalt	Mã hóa salt của password	Text	
AvatarUrl	Link ảnh Avatar của nhân viên	Text	
FullName	Tên đầy đủ của nhân viên	Text	
Location	Tỉnh/thành phố hiện tại của nhân viên	Text	
Address	Địa chỉ nhà của nhân viên	Text	
Phone	Số điện thoại của nhân viên	Text	
SocialLink	Link trang cá nhân của nhân viên	Text	
BirthDate	Ngày sinh của nhân viên	DateTime	
Gender	Giới tính của nhân viên	Text	
Description	Sơ lược giới thiệu nhân viên	Text	
PositionId	Id chức vụ của nhân viên	Char	Khóa ngoại

Bảng 2.9 Bảng Position

Tên	Mô tả	Data Type	Note
Id	Id của bảng Position	Char	Khóa chính
EntityStatus	Trạng thái của một thực thể	Integer	
CreatedTime	Ngày khởi tạo	Date Time	
UpdatedTime	Ngày chỉnh sửa cuối cùng	Date Time	
Name	Tên chức vụ của nhân viên	Text	

Bảng 2.10 Bảng Skill

Tên	Mô tả	Data Type	Note
Id	Id của bảng Skill	Char	Khóa chính
EntityStatus	Trạng thái của một thực thể	Integer	
CreatedTime	Ngày khởi tạo	Date Time	
UpdatedTime	Ngày chỉnh sửa cuối cùng	Date Time	
Name	Tên kỹ năng của nhân viên	Text	

Bảng 2.11 Bảng UserSkill

Tên	Mô tả	Data Type	Note
Id	Id của bảng	Char	Khóa chính
EntityStatus	Trạng thái của một thực thể	Integer	

CreatedTime	Ngày khởi tạo	Date Time	
UpdatedTime	Ngày chỉnh sửa cuối cùng	Date Time	
UserId	Id của User	Char	Khóa ngoại
SkillId	Id của Skill	Char	Khóa ngoại

Bảng 2.12 Bảng Role

Tên	Mô tả	Data Type	Note
Id	Id của bảng Role	Char	Khóa chính
EntityStatus	Trạng thái của một thực thể	Integer	
CreatedTime	Ngày khởi tạo	Date Time	
UpdatedTime	Ngày chỉnh sửa cuối cùng	Date Time	
Name	Tên vai trò của nhân viên	Text	

Bảng 2.13 Bảng UserRole

Tên	Mô tả	Data Type	Note
Id	Id của bảng UserRole	Char	Khóa chính
EntityStatus	Trạng thái của một thực thể	Integer	
CreatedTime	Ngày khởi tạo	Date Time	
UpdatedTime	Ngày chỉnh sửa cuối cùng	Date Time	
UserId	Id của User	Char	Khóa ngoại
RoleId	Id của Role	Char	Khóa ngoại

Bảng 2.14 Bảng TimeLineEvent

Tên	Mô tả	Data Type	Note
Id	Id của bảng TimeLineEvent	Char	Khóa chính
EntityStatus	Trạng thái của một thực thể	Integer	
CreatedTime	Ngày khởi tạo	Date Time	
UpdatedTime	Ngày chỉnh sửa cuối cùng	Date Time	
Event	Tên sự kiện trong đời của nhân viên	Text	
UserId	Id của User	Char	Khóa ngoại

Bảng 2.15 Bảng Board

Tên	Mô tả	Data Type	Note
Id	Id của bảng Board	Char	Khóa chính
EntityStatus	Trạng thái của một thực thể	Integer	
CreatedTime	Ngày khởi tạo	Date Time	
UpdatedTime	Ngày chỉnh sửa cuối cùng	Date Time	
Name	Tên của dự án	Text	
Key	Key của dự án	Text	
Description	Giới thiệu sơ lược dự án	Text	
ImageUrl	Link ảnh đại diện của dự án	Text	
PhaseOrder	Thứ tự của các tiến trình trong dự án	Text	

Bảng 2.16 Bảng Phase

Tên	Mô tả	Data Type	Note
Id	Id của bảng	Char	Khóa chính
EntityStatus	Trạng thái của một thực thể	Integer	
CreatedTime	Ngày khởi tạo	Date Time	
UpdatedTime	Ngày chỉnh sửa cuối cùng	Date Time	
Name	Tên của tiến trình	Text	
TaskOrder	Thứ tự của các task trong cột	Text	
BoardId	Id của dự án	Char	Khóa ngoại

Bảng 2.17 Bảng BoardUser

Tên	Mô tả	Data Type	Note
Id	Id của bảng	Char	Khóa chính
EntityStatus	Trạng thái của một thực thể	Integer	
CreatedTime	Ngày khởi tạo	Date Time	
UpdatedTime	Ngày chỉnh sửa cuối cùng	Date Time	
UserId	Id của User	Char	Khóa ngoại
BoardIdId	Id của Board	Char	Khóa ngoại

Bảng 2.18 Bảng Task

Tên	Mô tả	Data Type	Note
Id	Id của bảng	Char	Khóa chính
EntityStatus	Trạng thái của một thực thể	Integer	
CreatedTime	Ngày khởi tạo	Date Time	
UpdatedTime	Ngày chỉnh sửa cuối cùng	Date Time	
Title	Title của công việc	Text	
Description	Sơ lược về công việc	Text	
StartDate	Ngày bắt đầu công việc	DateTime	
DueDate	Ngày kết thúc công việc	DateTime	
Status	Tình trạng công việc	Text	
Estimation	Ước lượng thời gian làm việc	TimeSpan	
UserId	Id của nhân viên tạo ra công việc	Char	Khóa ngoại
BoardId	Id của dự án	Char	Khóa ngoại



Bảng 2.19 Bảng Attachment

Tên	Mô tả	Data Type	Note
Id	Id của bảng Attachment	Char	Khóa chính
EntityStatus	Trạng thái của một thực thể	Integer	
CreatedTime	Ngày khởi tạo	Date Time	
UpdatedTime	Ngày chỉnh sửa cuối cùng	Date Time	
Url	Link Attachment	Text	
Type	Loại attachment	Int	
TaskId	Id của công việc	Char	Khóa ngoại

Bảng 2.20 Bảng Todo

Tên	Mô tả	Data Type	Note
Id	Id của bảng Todo	Char	Khóa chính
EntityStatus	Trạng thái của một thực thể	Integer	
CreatedTime	Ngày khởi tạo	Date Time	
UpdatedTime	Ngày chỉnh sửa cuối cùng	Date Time	
Content	Nội dung phải làm của công việc	Text	
Finished	Trạng thái công việc	Bool	
TaskId	Id của công việc	Char	Khóa ngoại

Bảng 2.21 Bảng Label

Tên	Mô tả	Data Type	Note
Id	Id của bảng Label	Char	Khóa chính
EntityStatus	Trạng thái của một thực thể	Integer	
CreatedTime	Ngày khởi tạo	Date Time	
UpdatedTime	Ngày chỉnh sửa cuối cùng	Date Time	
Name	Tên của nhãn dán cho công việc	Text	
Color	Màu của nhãn dán	Text	
BoardId	Id của dự án	Char	Khóa ngoại

Bảng 2.22 Bảng TaskLabel

Tên	Mô tả	Data Type	Note
Id	Id của bảng TaskLabel	Char	Khóa chính
EntityStatus	Trạng thái của một thực thể	Integer	
CreatedTime	Ngày khởi tạo	Date Time	
UpdatedTime	Ngày chỉnh sửa cuối cùng	Date Time	
TaskId	Id của công việc	Char	Khóa ngoại
LabelId	Id của nhãn dán	Char	Khóa ngoại

**Bảng 2.23 Bảng WorkLog**

Tên	Mô tả	Data Type	Note
Id	Id của bảng WorkLog	Char	Khóa chính
EntityStatus	Trạng thái của một thực thể	Integer	
CreatedTime	Ngày khởi tạo	Date Time	
UpdatedTime	Ngày chỉnh sửa cuối cùng	Date Time	
Amount	Khối lượng thời gian làm việc	TimeSpan	
DateLog	Ngày ghi lại	DateTime	
Description	Mô tả về worklog	Text	
TaskId	Id của công việc	Char	Khóa ngoại
UserId	Id của nhân viên	Char	Khóa ngoại

**Bảng 2.24 Bảng Comment**

Tên	Mô tả	Data Type	Note
Id	Id của bảng Comment	Char	Khóa chính
EntityStatus	Trạng thái của một thực thể	Integer	
CreatedTime	Ngày khởi tạo	Date Time	
UpdatedTime	Ngày chỉnh sửa cuối cùng	Date Time	
Content	Nội dung comment	Text	
TaskId	Id của công việc	Char	Khóa ngoại
UserId	Id của nhân viên	Char	Khóa ngoại

**Bảng 2.25 Bảng TaskAssignee**

Tên	Mô tả	Data Type	Note
Id	Id của bảng TaskAssignee	Char	Khóa chính
EntityStatus	Trạng thái của một thực thể	Integer	
CreatedTime	Ngày khởi tạo	Date Time	
UpdatedTime	Ngày chỉnh sửa cuối cùng	Date Time	
TaskId	Id của công việc	Char	Khóa ngoại
UserId	Id của nhân viên	Char	Khóa ngoại

**Bảng 2.26 Bảng TaskAction**

Tên	Mô tả	Data Type	Note
Id	Id của bảng Task Action	Char	Khóa chính
EntityStatus	Trạng thái của một thực thể	Integer	
CreatedTime	Ngày khởi tạo	Date Time	
UpdatedTime	Ngày chỉnh sửa cuối cùng	Date Time	
ActionDescription	Mô tả về action	Text	
TaskId	Id của công việc	Char	Khóa ngoại
UserId	Id của nhân viên	Char	Khóa ngoại

## **2.4. Kết chương**

Chương này cung cấp cho người đọc cái nhìn tổng quan về chuyên môn nghiệp vụ mà website cần phải đáp ứng. Quá trình phân tích và thiết kế hệ thống quản lý, giám sát tiến độ công việc và dự án tại công ty.

## **CHƯƠNG 3. TRIỂN KHAI VÀ KẾT QUẢ**

### **3.1. Môi trường triển khai**

*API Web Service:*

- Ngôn ngữ: C#
- Công nghệ: .NET CORE, Entity Framework
- Design Patern: Repository Patern
- Hệ quản trị cơ sở dữ liệu: SQL Server
- Hệ điều hành: Windows

*Web Client:*

- Ngôn ngữ: HTML, CSS, JavaScript
- Công nghệ: React JS, Ant Design, Umi, Node
- Tích hợp trên hầu hết các trình duyệt Google Chrome, Firefox, IE, Edge, ...
- Hệ điều hành: Windows

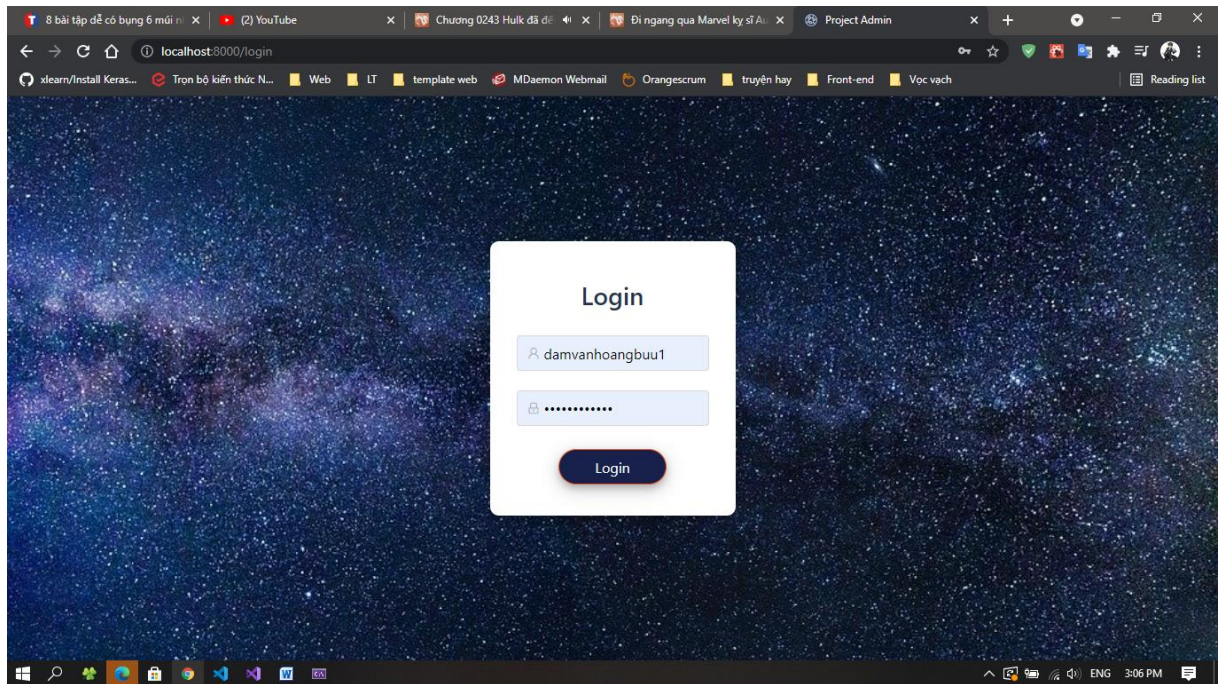
*Môi trường phát triển:*

- Visual Studio Code.
- Visual Studio 2019.
- Sql Server Management System

### **3.2. Kịch bản thử nghiệm**

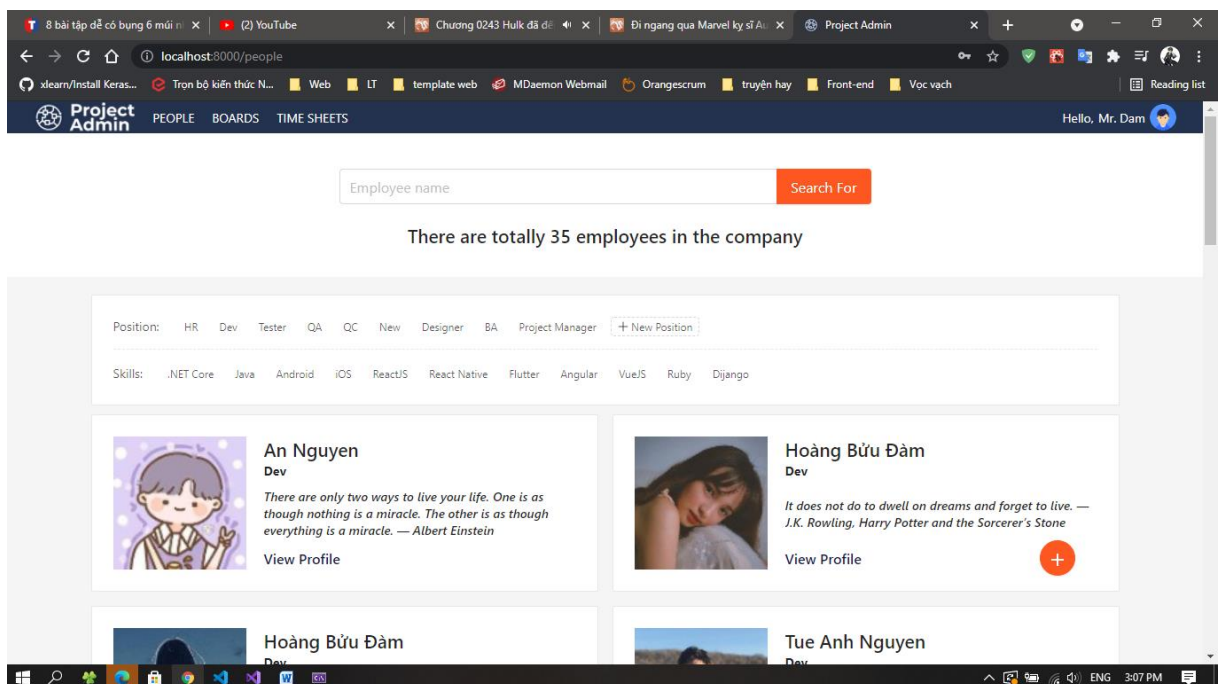
- Đăng nhập
- Tạo mới một dự án
- Điền đầy đủ thông tin dự án
- Thêm người vào dự án
- Tạo mới các qui trình làm việc
- Tạo mới các công việc
- Gán nhãn cho công việc
- Giao việc cho thành viên
- Lưu lại thời gian làm việc

### 3.3. Kết quả thực hiện



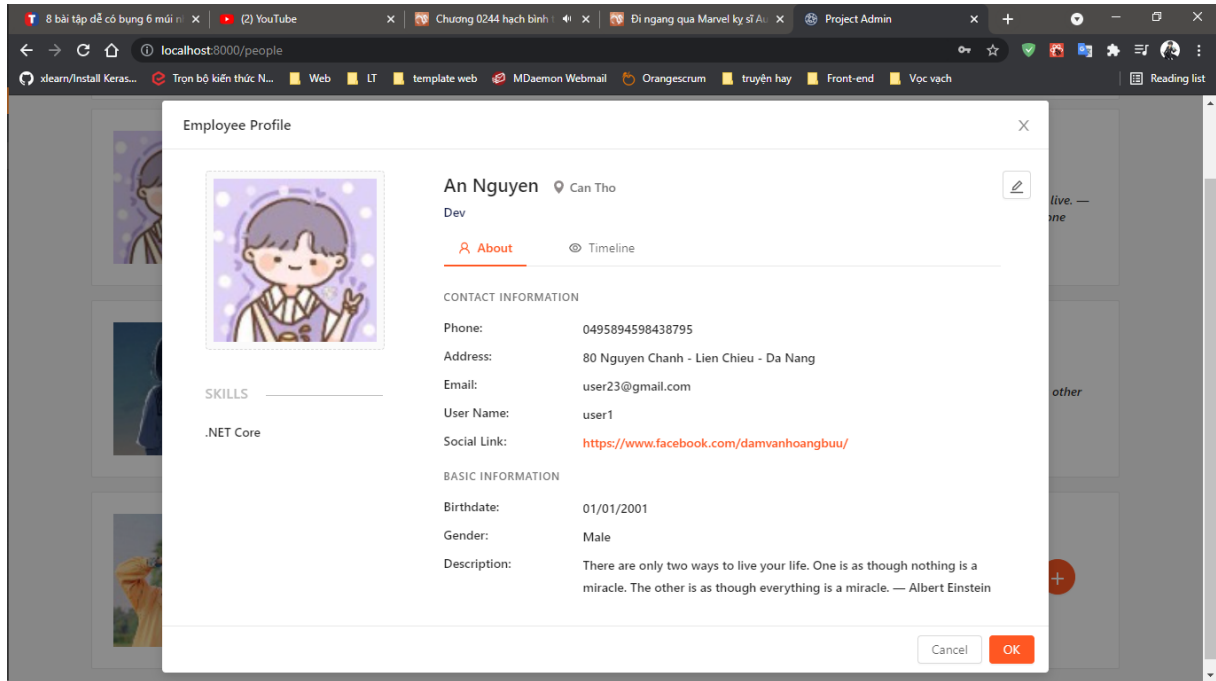
Hình 3.1 Màn hình Login

Nhân viên có thể đăng nhập để truy cập vào hệ thống bằng cách đăng nhập qua email và password mà admin đã tạo cho.



Hình 3.2 Quản lý nhân viên

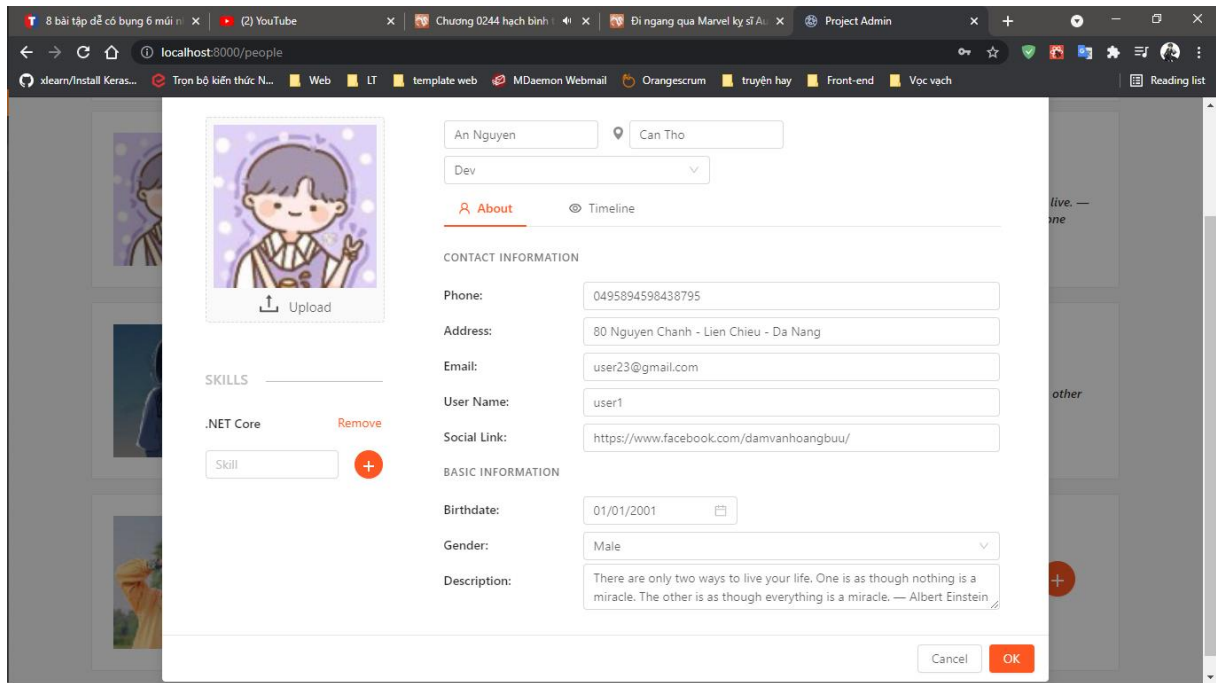
Sau khi đăng nhập vào hệ thống thì cả tài khoản admin và nhân viên đều được đưa vào màn hình quản lý nhân viên. Tuy nhiên, chỉ có admin mới nhìn thấy button thêm mới nhân viên ở góc phải bên dưới. Khi nhấn vào đó thì màn hình thêm mới nhân viên sẽ hiện ra. Ngoài ra, người dùng có thể tìm kiếm theo tên bằng cách dùng thanh tìm kiếm, hoặc có thể lọc danh sách người dùng dựa vào Position (chức vị) và Skill (kỹ năng).



Hình 3.3 Xem thông tin cá nhân của nhân viên

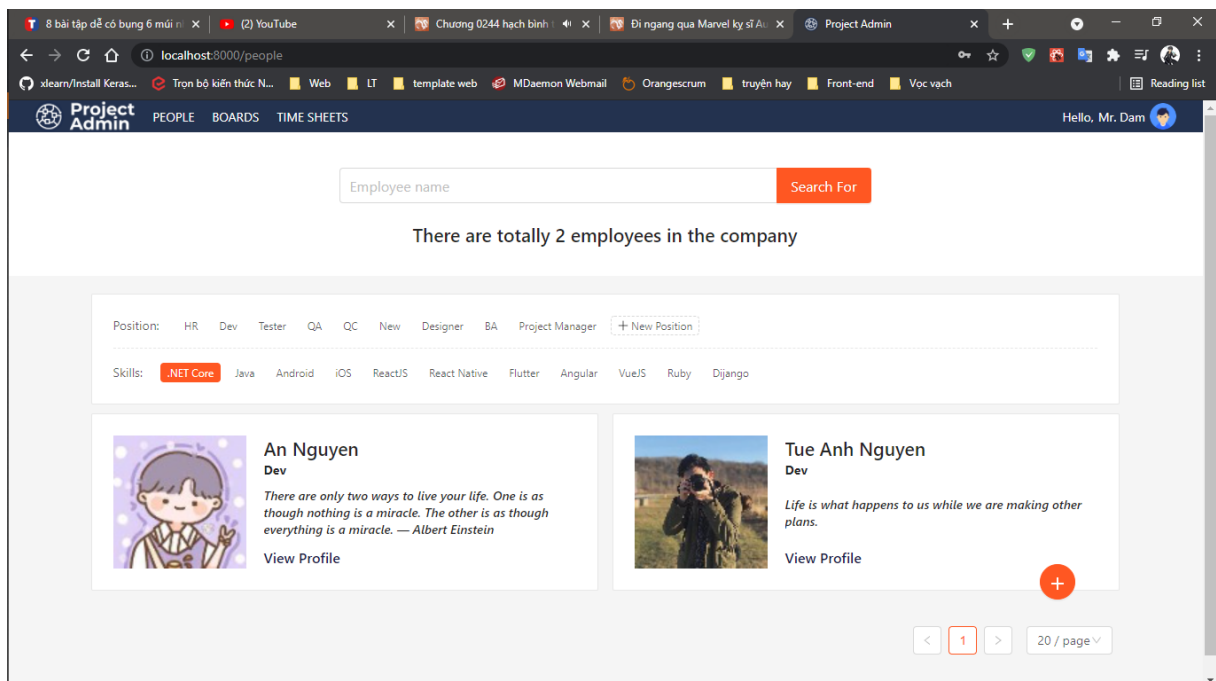
Sau khi nhấn vào “Wiew profile” thì giao diện thông tin nhân viên sẽ hiện lên với đầy đủ các thông tin của nhân viên.





Hình 3.4 Chỉnh sửa thông tin cá nhân

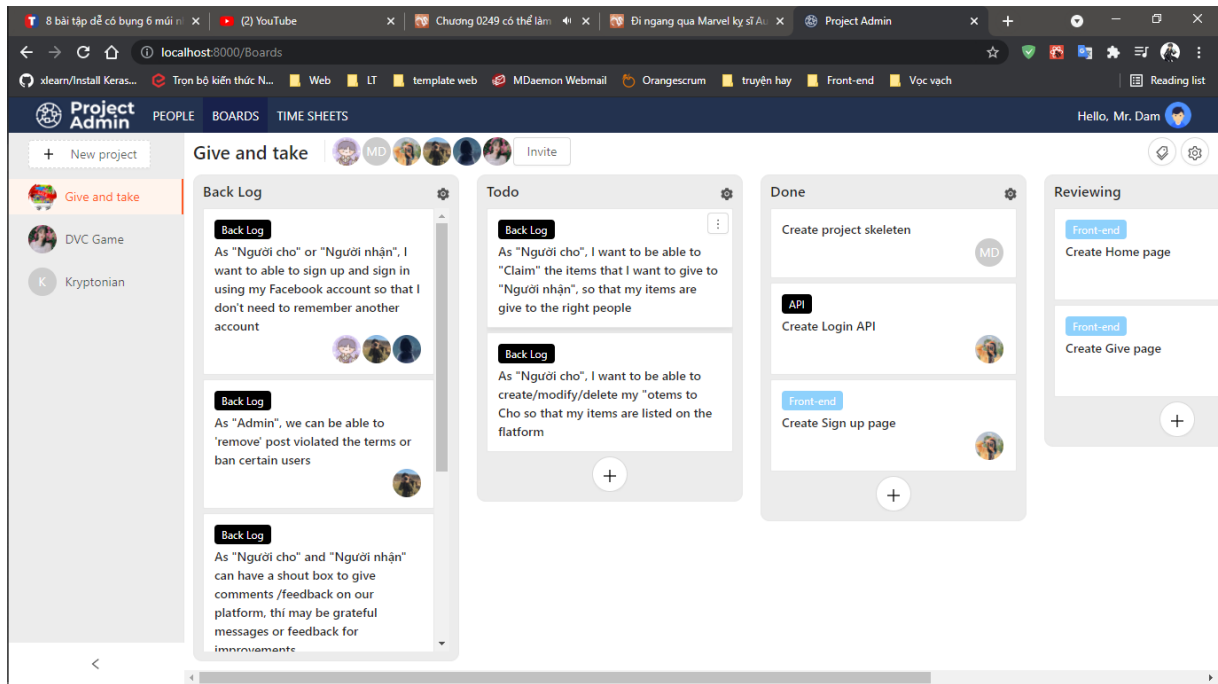
Màn hình chỉnh sửa hiện lên nếu ta nhấn button có biểu tượng edit trong giao diện profile nhân viên. Sau khi điền các thông tin cần chỉnh sửa thì nhấn “OK” để xác nhận, hoặc nhấn “Cancel” nếu không muốn chỉnh sửa.



Hình 3.5 Lọc và tìm kiếm nhân viên dựa trên kỹ năng, vị trí, tên, ...

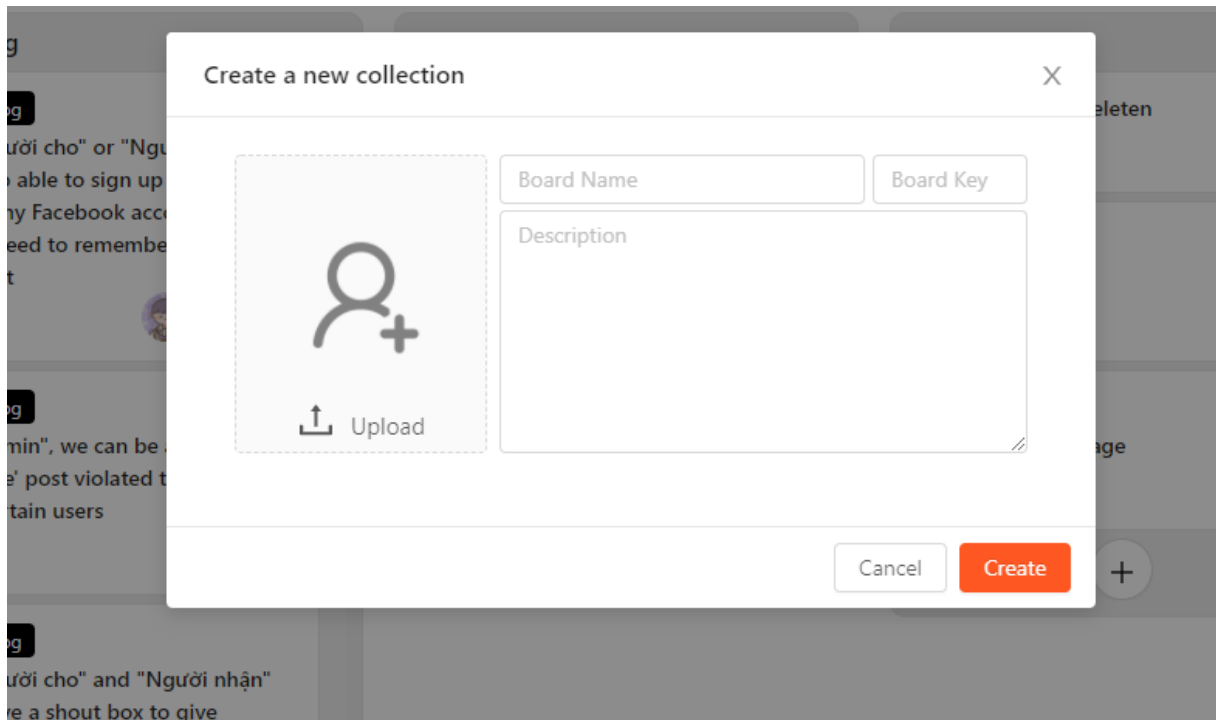
Như đã nói ở trên, ta có thể lọc nhân viên dựa trên kỹ năng và chức vụ trong công ty ở màn hình quản lý nhân viên





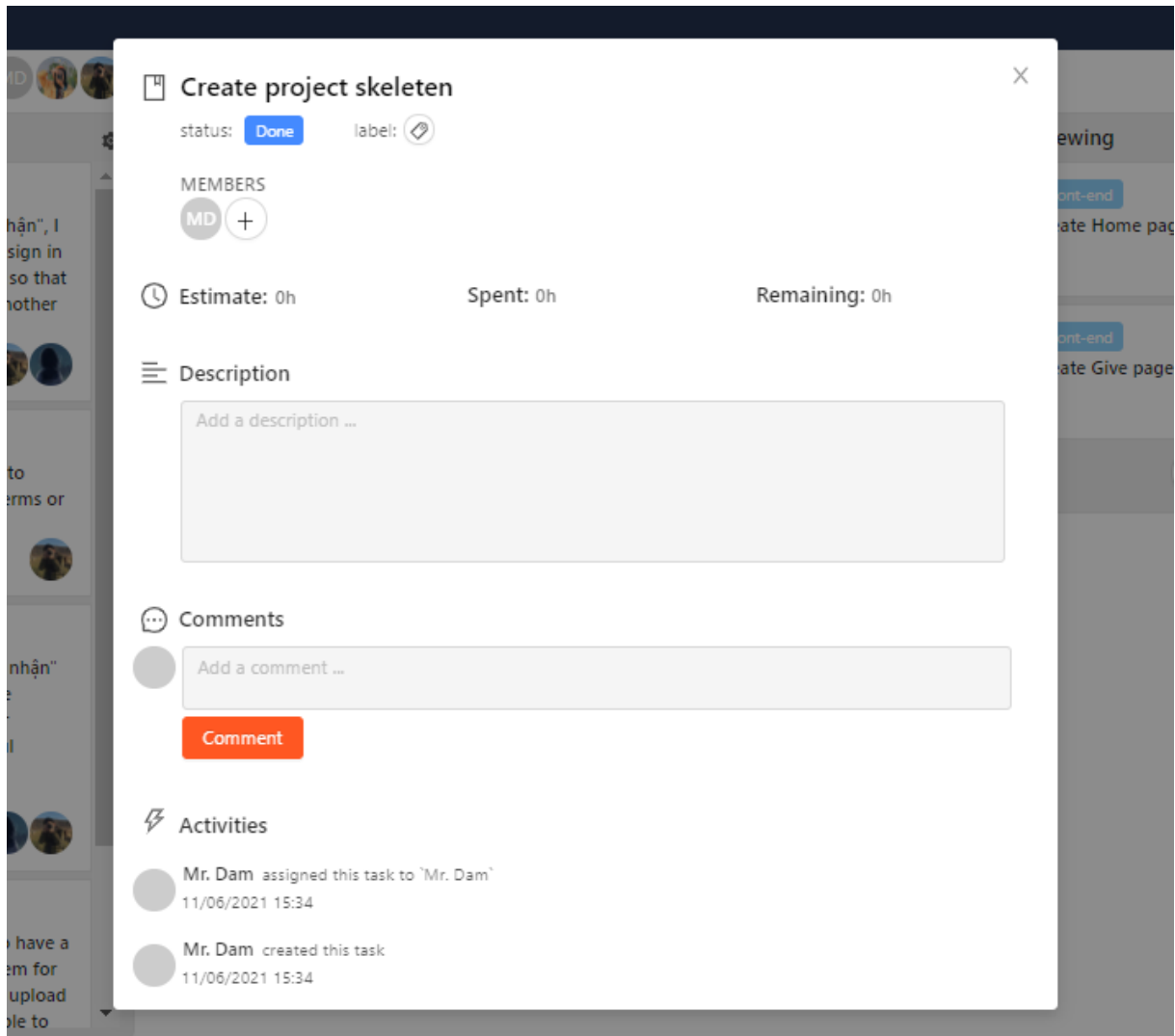
Hình 3.6 Màn hình quản lý các dự án

Ở màn hình này, nhân có thể thực hiện chỉnh sửa thông tin, thêm thành viên, thêm tiến trình, thêm công việc... trong những project mà họ đã tham gia. Riêng admin sẽ thấy toàn bộ project của công ty.

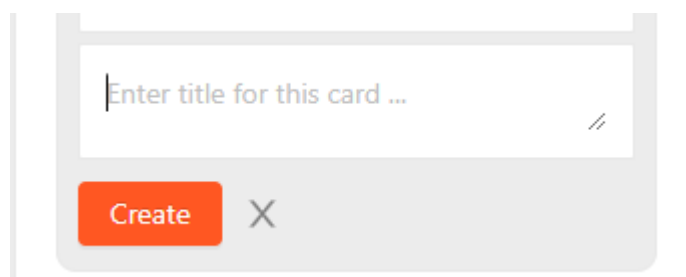


Hình 3.7 Tạo một dự án mới

Khi nhấn vào “New project” giao diện thêm mới dự án sẽ hiện ra. Chỉ cần nhập đầy đủ thông tin và nhấn “Create” nhân viên có thể tạo mới một dự án.

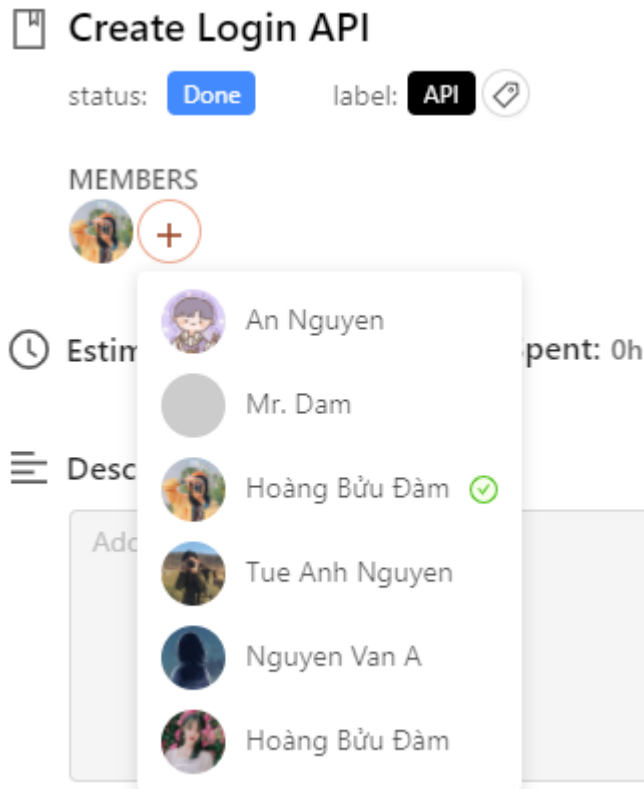


Hình 3.8 Thông tin chi tiết của một công việc



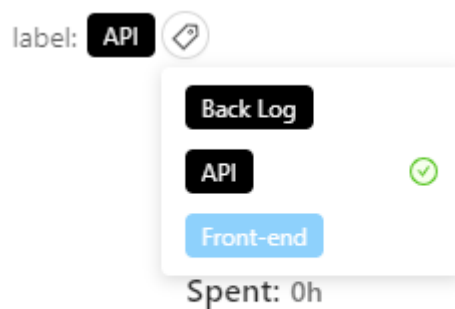
Hình 3.9 Tạo một công việc mới

Sẽ có button tạo mới công việc trong mỗi tiến trình của dự án, chỉ cần nhấn vào và điền tên công việc rồi nhấn “Create” là có thể tạo công việc mới. Có thể những thông tin của công việc nếu nhấn vào công việc đó.



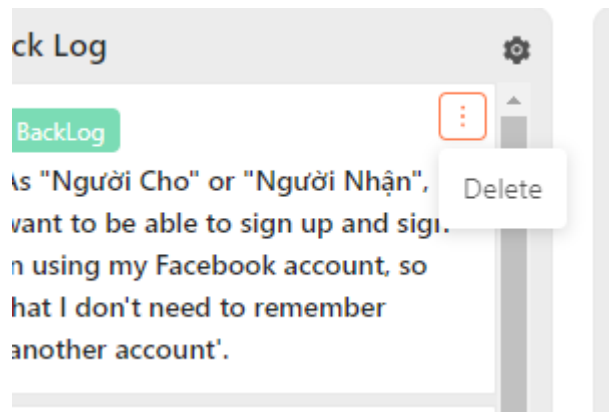
Hình 3.10 Giao việc cho thành viên trong dự án

Trong phần thông tin công việc, khi nhấn button (+) ở phần member sẽ hiển thị ra tên những thành viên trong dự án. Nếu muốn giao việc cho ai đó chỉ cần nhấn vào tên của họ và nhấn lần 2 nếu không muốn giao việc cho họ nữa.

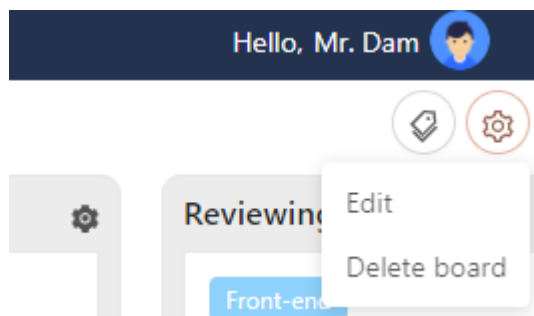


Hình 3.11 Thêm nhãn cho công việc để dễ quản lý

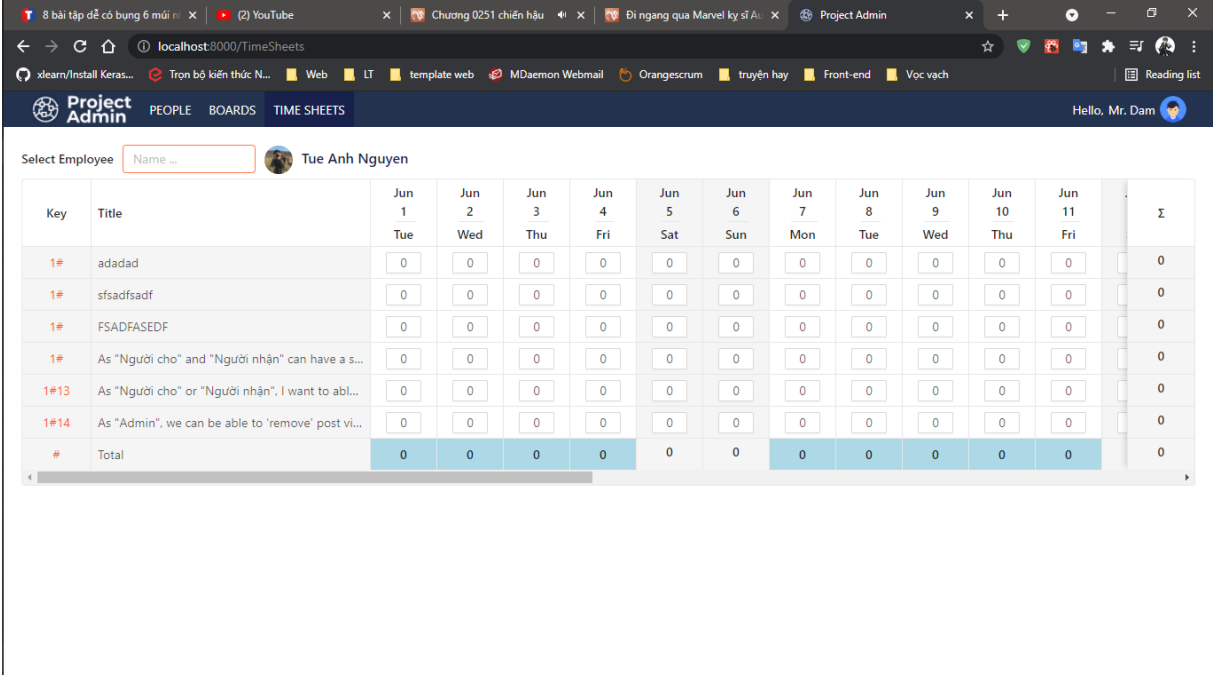
Khi nhấn vào biểu tượng nhãn thì có thể gắn nhãn cho công việc. Cách chọn nhãn tương tự như cách giao việc.



Hình 3.12 Xóa một công việc



Hình 3.13 Các tùy chọn cho một bản



The screenshot shows a web application interface for 'Project Admin'. The top navigation bar includes 'PEOPLE', 'BOARDS', and 'TIME SHEETS'. The 'TIME SHEETS' section is active, displaying a table for employee time tracking. The table has columns for days of the week (Jun 1 to Jun 11) and a total column (Σ). The rows list employees with their keys and titles. The table data is as follows:

Key	Title	Jun 1 Tue	Jun 2 Wed	Jun 3 Thu	Jun 4 Fri	Jun 5 Sat	Jun 6 Sun	Jun 7 Mon	Jun 8 Tue	Jun 9 Wed	Jun 10 Thu	Jun 11 Fri	Σ
1#	adadad	0	0	0	0	0	0	0	0	0	0	0	0
1#	sfsadfsadf	0	0	0	0	0	0	0	0	0	0	0	0
1#	FSADFASEDF	0	0	0	0	0	0	0	0	0	0	0	0
1#	As "Người cho" and "Người nhận" can have a s...	0	0	0	0	0	0	0	0	0	0	0	0
1#13	As "Người cho" or "Người nhận", I want to abl...	0	0	0	0	0	0	0	0	0	0	0	0
1#14	As "Admin", we can be able to 'remove' post vi...	0	0	0	0	0	0	0	0	0	0	0	0
#	Total	0	0	0	0	0	0	0	0	0	0	0	0

Hình 3.14 Quản lý thời gian

Trong màn hình này, nhân viên chỉ có thể thấy được thời gian làm việc của họ. Admin có thể xem được thời gian làm việc của tất cả nhân viên trong công ty.

	Jun 9 Wed	Jun 10 Thu	Jun 11 Fri	Jun 12 Sat	Jun 13 Sun	Jun 14 Mon
	0	0	0	0	0	0
	0					0
	0					0
	0					0
	0					0
	0					0
	0					0
	0					0
	0					0
	0					0

00:00:00

Description

✓ Save and close

Cancel

Hình 3.15 Ghi lại thời gian đã dành cho một công việc

Để ghi lại thời gian làm việc chỉ cần nhấn vào ô với công việc và ngày tương ứng rồi ghi lại thời gian làm việc, mô tả (nếu muốn) rồi nhấn “Save and close”.

## KẾT LUẬN

### 1. Kết quả đạt được

Trong thời gian tìm hiểu, nghiên cứu cơ sở lý thuyết và triển khai đồ án, em đã được những kết quả sau:

*Về mặt lý thuyết*, việc làm đồ án đã cũng có thêm được kiến thức lập trình, cơ sở dữ liệu và phân tích thiết kế hệ thống. Ngoài ra, con tìm hiểu thêm kiến thức mới về việc triển khai dự án lên internet, học được các công nghệ mới như docker, React, .NET, docker, .... Việc sử dụng thành thạo các phần mềm lập trình, công cụ hỗ trợ quản lý dự án đã hỗ trợ rất lớn cho việc nghiên cứu và kinh nghiệm trước khi đi làm.

*Về mặt thực tiễn ứng dụng*, hệ thống đã đáp ứng được cơ bản chức năng Xây dựng website giúp công ty quản lý công việc và thời gian cho nhân viên, phục vụ cho mục đích giảm thiểu chi phí vận hành cho công ty, bao gồm:

- Module quản lý nhân viên.
- Module quản lý dự án và tiến độ công việc.
- Module quản lý thời gian biểu.

Tuy nhiên, dù đã cố gắng và nỗ lực hết sức, đồ án còn tồn tại các vấn đề như sau:

- Chưa hỗ trợ nền tảng điện thoại di động.
- Chưa hoàn thiện quy trình làm việc
- Chưa phân quyền cho chi tiết cho admin, project manager, ...

### 2. Hướng phát triển

Một số hướng nghiên cứu và phát triển của đề tài như sau:

- Cải thiện giao diện ứng dụng
- Tạo ứng dụng cho nền tảng di động để báo tiến độ công việc về điện thoại của nhân viên
- Có thể tích hợp thêm quản lý lịch nghỉ phép, báo vắng, ốm của nhân viên để hoàn thiện quy trình làm việc của nhân viên
- Tích hợp các biểu đồ trong quy trình Scrum – Agile như Burn down chart, ... để tiện hơn cho việc kiểm soát tiến độ dự án



## **TÀI LIỆU THAM KHẢO**

- [1] HTML: <https://www.w3schools.com/html/>
- [2] CSS: <https://www.w3schools.com/css/>
- [3] JavaScript: <https://www.w3schools.com/js/DEFAULT.asp>
- [4] Giáo trình C# - Phạm Văn Việt, Trương Lập Vĩ
- [5] ASP.NET Core: <https://docs.microsoft.com/en-us/aspnet/core>
- [6] RESTful API: <https://www.restapitutorial.com/>
- [7] ReactJS: <https://reactjs.org/>
- [8] SQL Server: <https://www.sqlservertutorial.net/>
- [9] PGS. TS Phan Huy Khánh, Phân tích và Thiết kế Hệ thống Thông tin Quản lý, 2016.
- [10] TS. Lê Thị Mỹ Hạnh, Slide Công nghệ phần mềm, 2016.
- [11] Trần Nguyên Phong, Giáo trình SQL, 2004.
- [12] TS. Lê Thị Mỹ Hạnh, Giáo trình môn học Lập trình hướng đối tượng, 2002.

