

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

**ĐỒ ÁN TỐT NGHIỆP**  
**NGÀNH: CÔNG NGHỆ THÔNG TIN**  
**CHUYÊN NGÀNH: CÔNG NGHỆ PHẦN MỀM**

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

Người hướng dẫn: **PGS. TS. NGUYỄN TẤN KHÔI**  
Sinh viên thực hiện: **TRẦN NGỌC QUỐC**  
Số thẻ sinh viên: **102150127**  
Lớp: **15T2**

**Đà Nẵng, 12/2019**

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

# **ĐỒ ÁN TỐT NGHIỆP**

**NGÀNH: CÔNG NGHỆ THÔNG TIN**  
**CHUYÊN NGÀNH: CÔNG NGHỆ PHẦN MỀM**

**ĐỀ TÀI:**

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

Người hướng dẫn: **PGS. TS. NGUYỄN TẤN KHÔI**  
Sinh viên thực hiện: **TRẦN NGỌC QUỐC**  
Số thẻ sinh viên: **102150127**  
Lớp: **15T2**

**Đà Nẵng, 12/2019**

## 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: Trần Ngọc Quốc

Số thẻ SV: 102150127      Lớp: 15T2

Người hướng dẫn: PGS.TS. Nguyễn Tấn Khôi

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: Trần Ngọc Quốc

Số thẻ sinh viên: 102150127

Lớp:15T2

Khoa:CNTT

Ngành: CNPM

1. Tên đề tài đồ án:

Xây dựng hệ thống quản lí, 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: PGS.TS. Nguyễn Tấn Khôi

7. Ngày giao nhiệm vụ đồ án: 05/09/2019.

8. Ngày hoàn thành đồ án: 20/12/2019.

Đà Nẵng, ngày 20 tháng 12 năm 2019

**Trưởng Bộ môn:** Lê Thị Mỹ Hạnh

**Người hướng dẫn**

PGS.TS. Nguyễn Tấn Khôi

## 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 Nguyễn Tấn Khôi – giảng viên ngành 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 20 tháng 12 năm 2019*

Sinh viên thực hiện

**Trần Ngọc Quốc**

## **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 Nguyễn Tấn Khôi.
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

**Trần Ngọc Quốc**



# 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..... viii

**MỞ ĐẦU.....1**

**1. Tổng quan về đề tài .....1**

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

2.1. Mục đích đề tài: .....2

2.2. Nhiệm vụ chính của đề tài: .....2

**3. Công nghệ sử dụng.....2**

**4. Công cụ hỗ trợ .....2**

**5. Cấu trúc đồ án .....2**

**CHƯƠNG 1. CƠ SỞ LÝ THUYẾT.....4**

**1.1. React Js .....4**

1.1.1. Giới thiệu.....4

1.1.2. Virtual DOM .....4

1.1.3. One-way data binding .....5

1.1.4. Giới thiệu về JSX.....5

1.1.5. Giới thiệu về Components .....5

1.1.6. Props và State .....6

1.1.7. Lifecycle .....6

**1.2. Redux-Saga.....7**

1.2.1. Giới thiệu.....7

1.2.2. Side effect .....7

1.2.3.	<i>Generator function</i> .....	7
1.2.4.	<i>Cách hoạt động</i> .....	7
<b>1.3.</b>	<b>WEB API</b> .....	8
1.3.1.	<i>ASP.NET CORE</i> .....	8
1.3.2.	<i>Entity Framework</i> .....	8
1.3.3.	<i>Phương thức HTTP</i> .....	9
<b>1.4.</b>	<b>Docker</b> .....	9
1.4.1.	<i>Khái niệm</i> .....	9
1.4.2.	<i>Lợi ích của Docker</i> .....	10
<b>CHƯƠNG 2.</b>	<b>PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG</b> .....	11
<b>2.1.</b>	<b>Đặt vấn đề</b> .....	11
<b>2.2.</b>	<b>Cách chức năng chính</b> .....	11
2.2.1.	<i>Quản lý thông tin nhân viên</i> .....	11
2.2.2.	<i>Quản lý dự án</i> .....	12
2.2.3.	<i>Quản lý công việc</i> .....	13
2.2.4.	<i>Quản lý thời gian làm việc</i> .....	14
<b>2.3.</b>	<b>Các module chính của hệ thống</b> .....	15
<b>2.4.</b>	<b>Sơ đồ phân rã chức năng</b> .....	16
<b>2.5.</b>	<b>Sơ đồ hoạt động</b> .....	17
<b>2.6.</b>	<b>Luồng dữ liệu</b> .....	21
<b>2.7.</b>	<b>Sơ đồ quan hệ</b> .....	22
<b>2.8.</b>	<b>Thiết kế cơ sở dữ liệu</b> .....	23
2.8.1.	<i>Sơ đồ quan hệ, thực thể dữ liệu</i> .....	23
2.8.2.	<i>Đặc tả các bảng dữ liệu</i> .....	24
<b>CHƯƠNG 3.</b>	<b>TRIỂN KHAI VÀ KẾT QUẢ</b> .....	32
<b>3.1.</b>	<b>Môi trường triển khai</b> .....	32
<b>1.2.</b>	<b>Kịch bản thử nghiệm</b> .....	32
<b>3.3.</b>	<b>Kết quả thực hiện</b> .....	33
<b>KẾT LUẬN</b> .....		40
<b>TÀI LIỆU THAM KHẢO</b> .....		1

## DANH SÁCH CÁC HÌNH VẼ

Hình 1.1: Đặc trưng Virtual DOM.....	4
Hình 1.2 Các luồng dữ liệu và sự kiện trong React .....	5
Hình 1.3 Các luồng dữ liệu và sự kiện trong React .....	6
Hình 1.4 Mô tả cách thức hoạt động của Redux - Saga.....	8
Hình 1.5 Mô tả cách thức hoạt động của Entity Framework.....	9
Hình 1.6 Mô tả cách thức hoạt động của Docker .....	10
Hình 2.1 Sơ đồ use-case cho module quản lý nhân viên.....	12
Hình 2.2 Sơ đồ use-case cho module quản lý dự án .....	13
Hình 2.3 Sơ đồ use-case cho module quản lý tiến độ công việc trong dự án .....	14
Hình 2.4 Sơ đồ use-case module quản lý thời gian làm việc .....	15
Hình 2.5 Sơ đồ hoạt động của module quản lý nhân viên .....	17
Hình 2.6 Sơ đồ hoạt động của module quản lý dự án.....	18
Hình 2.7 Sơ đồ hoạt động của module quản lý tiến độ công việc.....	19
Hình 2.8 Sơ đồ hoạt động của module quản lý thời gian biểu, chấm công .....	20
Hình 2.9 Mô tả luồng đi của dữ liệu .....	21
Hình 2.10 Sơ đồ lớp cho dự án.....	22
Hình 2.11 Sơ đồ quan hệ, thực thể dữ liệu.....	23
Hình 3.1: Màn hình Login.....	33
Hình 3.2 Quản lý nhân viên .....	33
Hình 3.3 Xem thông tin cá nhân của nhân viên .....	34
Hình 3.4 Chỉnh sửa thông tin cá nhân .....	34
Hình 3.5 Lọc và tìm kiếm nhân viên dựa trên kỹ năng, vị trí, tên, ... ..	35
Hình 3.6 Màn hình quản lý các dự án.....	35
Hình 3.7 Tạo một dự án mới.....	36
Hình 3.8 Thông tin chi tiết của một công việc.....	36

<i>Hình 3.9 Tạo một công việc mới.....</i>	<i>37</i>
<i>Hình 3.10 Giao việc cho thành viên trong dự án .....</i>	<i>37</i>
<i>Hình 3.11 Thêm nhãn cho công việc để dễ quản lí .....</i>	<i>38</i>
<i>Hình 3.12 Xóa một công việc .....</i>	<i>38</i>
<i>Hình 3.13 Các tùy chọn cho một bản .....</i>	<i>38</i>
<i>Hình 3.14 Quản lí thời gian.....</i>	<i>39</i>
<i>Hình 3.15 Ghi lại thời gian đã dành cho một công việc .....</i>	<i>39</i>

## DANH SÁCH CÁC BẢNG

<i>Bảng 2.1 Bảng User .....</i>	<i>24</i>
<i>Bảng 2.3 Bảng Position.....</i>	<i>25</i>
<i>Bảng 2.4 Bảng Skill .....</i>	<i>25</i>
<i>Bảng 2.5 Bảng Skill .....</i>	<i>25</i>
<i>Bảng 2.6 Bảng Role.....</i>	<i>26</i>
<i>Bảng 2.7 Bảng UserRole .....</i>	<i>26</i>
<i>Bảng 2.8 Bảng TimeLineEvent.....</i>	<i>26</i>
<i>Bảng 2.9 Bảng Board .....</i>	<i>27</i>
<i>Bảng 2.10 Bảng Phase .....</i>	<i>27</i>
<i>Bảng 2.11 Bảng BoardUser .....</i>	<i>28</i>
<i>Bảng 2.12 Bảng Task.....</i>	<i>28</i>
<i>Bảng 2.13 Bảng Attachment.....</i>	<i>29</i>
<i>Bảng 2.14 Bảng Todo .....</i>	<i>29</i>
<i>Bảng 2.15 Bảng Label .....</i>	<i>29</i>
<i>Bảng 2.16 Bảng TaskLabel.....</i>	<i>30</i>
<i>Bảng 2.17 Bảng WorkLog .....</i>	<i>30</i>
<i>Bảng 2.18 Bảng Comment.....</i>	<i>30</i>
<i>Bảng 2.19 Bảng TaskAssignee .....</i>	<i>31</i>
<i>Bảng 2.20 Bảng TaskAction .....</i>	<i>31</i>

## 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	Asynchronous 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
- Docker
- Nginx

## **4. Công cụ hỗ trợ**

- Visual Studio 2019
- Visual Studio Code
- SQL Server Management System
- Postman
- Git Kraken
- Filezilla

## **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, nên 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. React Js

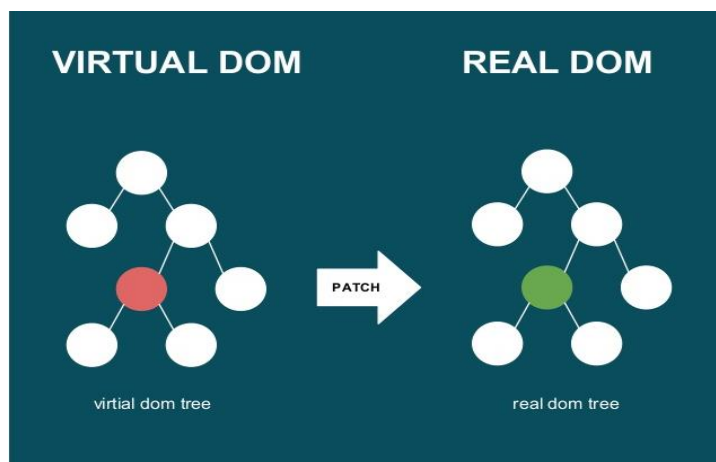
#### 1.1.1. Giới thiệu

React là một thư viện UI phát triển tại Facebook để hỗ trợ việc xây dựng những thành phần (components) UI có tính tương tác cao, có trạng thái và có thể sử dụng lại được. React được sử dụng tại Facebook trong production, và đồng thời ứng dụng instagram cũng được viết hoàn toàn trên React.

Một trong những điểm hấp dẫn của React là thư viện này không chỉ hoạt động trên phía client, mà còn được render trên server và có thể kết nối với nhau. React so sánh sự thay đổi giữa các giá trị của lần render này với lần render trước và cập nhật ít thay đổi nhất trên **DOM**. Trước khi đến cài đặt và cấu hình, chúng ta sẽ đi đến một số khái niệm cơ bản:

#### 1.1.2. Virtual DOM

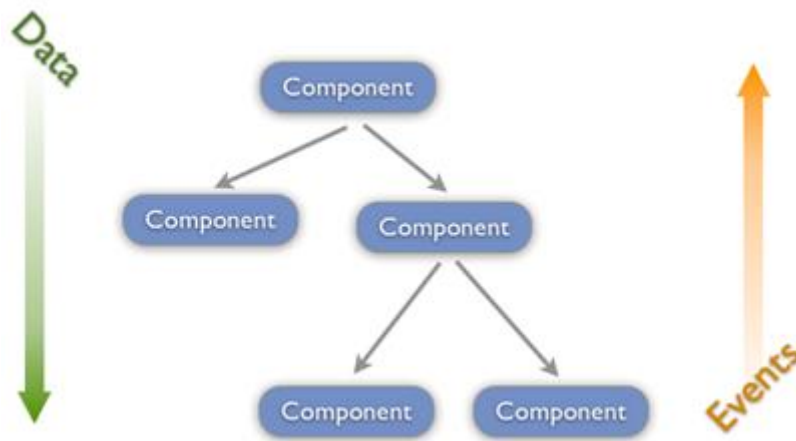
Công nghệ DOM ảo giúp tăng hiệu năng cho ứng dụng. Việc chỉ node gốc mới có trạng thái và khi nó thay đổi sẽ tái cấu trúc lại toàn bộ, đồng nghĩa với việc DOM tree cũng sẽ phải thay đổi một phần, điều này sẽ ảnh hưởng đến tốc độ xử lý. React JS sử dụng Virtual DOM (DOM ảo) để cải thiện vấn đề này. Virtual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



*Hình 1.1 Đặc trưng Virtual DOM*

### **1.1.3. One-way data binding**

React sử dụng cơ chế one-way data binding – luồng dữ liệu 1 chiều. Dữ liệu được truyền từ parent đến child thông qua props. Luồng dữ liệu đơn giản giúp chúng ta dễ dàng kiểm soát cũng như sửa lỗi.



*Hình 1.2 Các luồng dữ liệu và sự kiện trong React*

Với các đặc điểm ở trên, React dùng để xây dựng các ứng dụng lớn mà dữ liệu của chúng thay đổi liên tục theo thời gian. Dữ liệu thay đổi thì hầu hết kèm theo sự thay đổi về giao diện. Ví dụ như Facebook: trên Newsfeed của bạn cùng lúc sẽ có các status khác nhau và mỗi status lại có số like, share, comment liên tục thay đổi. Khi đó React sẽ rất hữu ích để sử dụng.

### **1.1.4. Giới thiệu về JSX**

JSX là một dạng ngôn ngữ cho phép viết các mã HTML trong Javascript. Đặc điểm: Faster: Nhanh hơn. JSX thực hiện tối ưu hóa trong khi biên dịch sang mã Javascript. Các mã này cho thời gian thực hiện nhanh hơn nhiều so với một mã tương đương viết trực tiếp bằng Javascript. Safer: an toàn hơn. Ngược với Javascript, JSX là kiểu statically-typed, nghĩa là nó được biên dịch trước khi chạy, giống như Java, C++. Vì thế các lỗi sẽ được phát hiện ngay trong quá trình biên dịch. Ngoài ra, nó cũng cung cấp tính năng gỡ lỗi khi biên dịch rất tốt. Easier: Dễ dàng hơn. JSX kế thừa dựa trên Javascript, vì vậy rất dễ dàng để cho các lập trình viên Javascripts có thể sử dụng

### **1.1.5. Giới thiệu về Components**

React được xây dựng xung quanh các component, chứ không dùng template như các framework khác. Trong React, chúng ta xây dựng trang web sử dụng những thành phần (component) nhỏ. Chúng ta có thể tái sử dụng một component ở nhiều nơi, với các trạng thái hoặc các thuộc tính khác nhau, trong một component lại có thể chứa thành

phần khác. Mỗi component trong React có một trạng thái riêng, có thể thay đổi, và React sẽ thực hiện cập nhật component dựa trên những thay đổi của trạng thái. Mọi thứ React đều là component. Chúng giúp bảo trì mã code khi làm việc với các dự án lớn. Một react component đơn giản chỉ cần một method render. Có rất nhiều methods khả dụng khác, nhưng render là method chủ đạo.

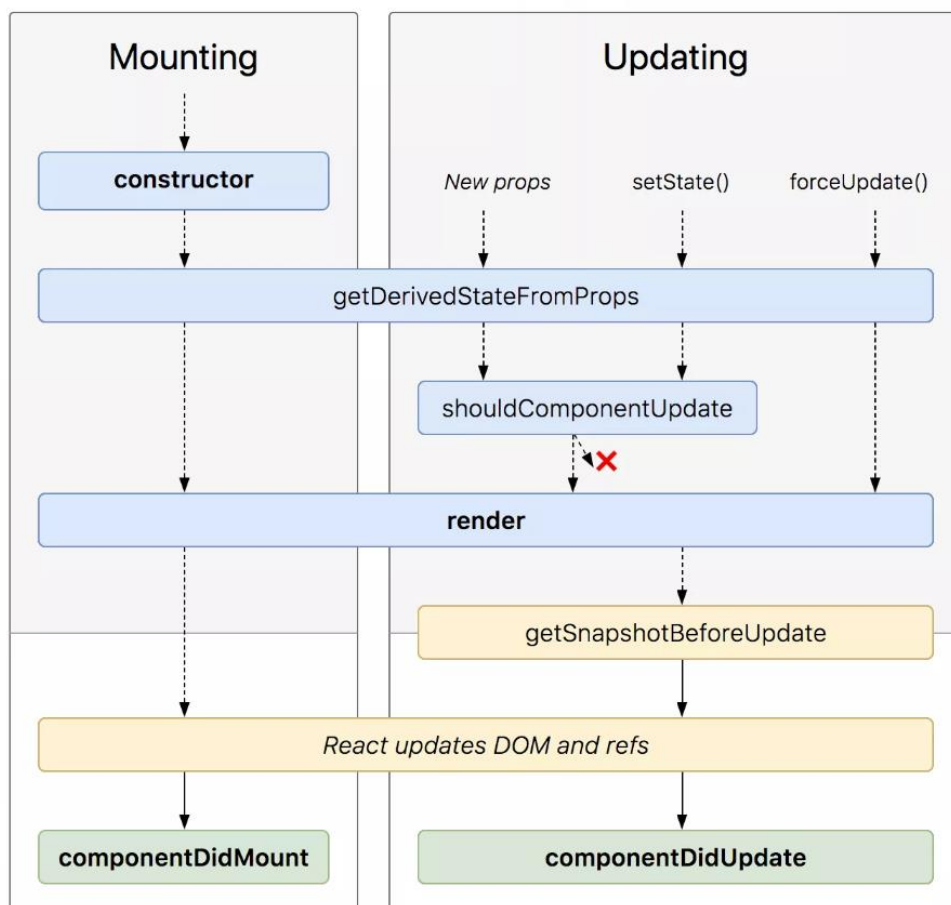
#### **1.1.6. Props và State**

Props: giúp các component tương tác với nhau, component nhận input gọi là props, và trả thuộc tính mô tả những gì component con sẽ render. Prop là bất biến.

State: thể hiện trạng thái của ứng dụng, khi state thay đổi thì component đồng thời render lại để cập nhật UI.

#### **1.1.7. Lifecycle**

*Lifecycle methods* là những method được gọi tại một thời điểm nào đó trong vòng đời của một component. Chúng ta có thể viết một *lifecycle methods* được gọi trước khi component được render trong lần đầu tiên. Hoặc được gọi sau khi component được render trong những lần sau.



*Hình 1.3 Các luồng dữ liệu và sự kiện trong React*

## **1.2. Redux-Saga**

### **1.2.1. Giới thiệu**

Redux-Saga là một thư viện redux middleware, giúp quản lý những side effect trong ứng dụng redux trở nên đơn giản hơn. Bằng việc sử dụng tối đa tính năng Generators (function\*) của ES6, nó cho phép ta viết async code nhìn giống như là synchronos.

### **1.2.2. Side effect**

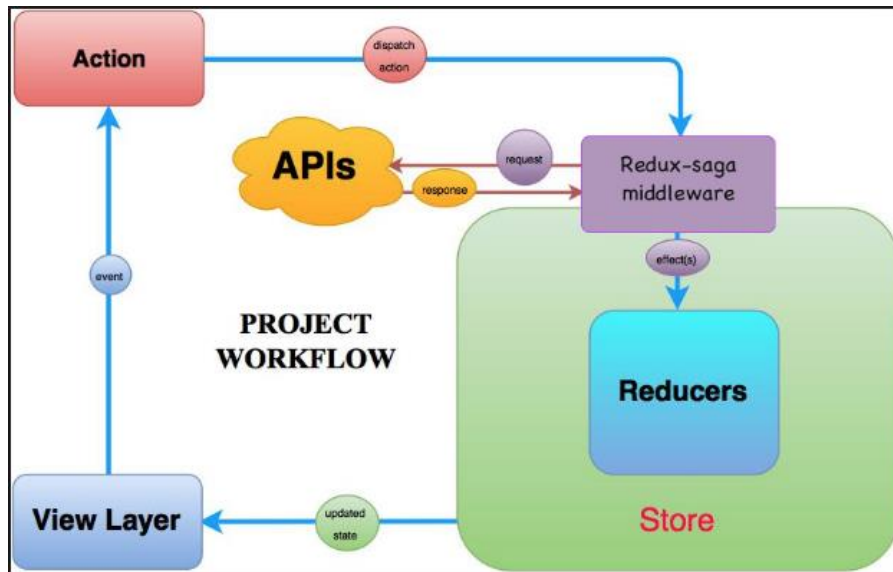
Ta đã biết tất cả những xử lý ở REDUCER đều phải là synchronous và pure tức chỉ là xử lý đồng bộ. Nhưng trong ứng dụng thực tế thì cần nhiều hơn vậy ví dụ như asynchronous (thực hiện một số việc như gọi một hàm AJAX để fetch dữ liệu về nhưng cần đợi kết quả chứ kết quả không trả về ngay được) hoặc là impure (thực hiện lưu, đọc dữ liệu ra bên ngoài như lưu dữ liệu ra ổ cứng hay đọc cookie từ trình duyệt... đều cần đợi kết quả). Những việc như thế trong lập trình hàm gọi nó là side effects.

### **1.2.3. Generator function**

Khác với function bình thường là thực thi và trả về kết quả, thì Generator function có thể thực thi, tạm dừng trả về kết quả và thực thi bằng tiếp. Từ khóa để làm được việc đây là “YIELD”. Generator được đưa ra cách đây mấy chục năm nhưng đến ES2015 mới được bổ sung, các ngôn ngữ khác đã được bổ sung tính năng này như C#, PHP, Ruby, C++, R....

### **1.2.4. Cách hoạt động**

Đối với logic của saga, ta cung cấp một hàm cho saga, chính hàm này là hàm đứng ra xem xét các action trước khi vào store, nếu là action quan tâm thì nó sẽ thực thi hàm sẽ được thực thi, nếu bạn biết khái niệm hook thì hàm cung cấp cho saga chính là hàm hook. Điều đặc biệt của hàm hook này nó là một generator function, trong generator function này có yield và mỗi khi yield ta sẽ trả về một plain object. Object trả về đó được gọi Effect object. effect object này đơn giản chỉ là một object bình thường nhưng chứa thông tin đặc biệt dùng để chỉ dẫn middleware của Redux thực thi các hoạt động khác ví dụ như gọi một hàm async khác hay put một action tới store. Để tạo ra effect object đề cập ở trên thì ta gọi hàm từ thư viện của saga là redux-saga/effects.



*Hình 1.4 Mô tả cách thức hoạt động của Redux - Saga*

### **1.3. WEB API**

#### **1.3.1. 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)

#### **1.3.2. Entity Framework**

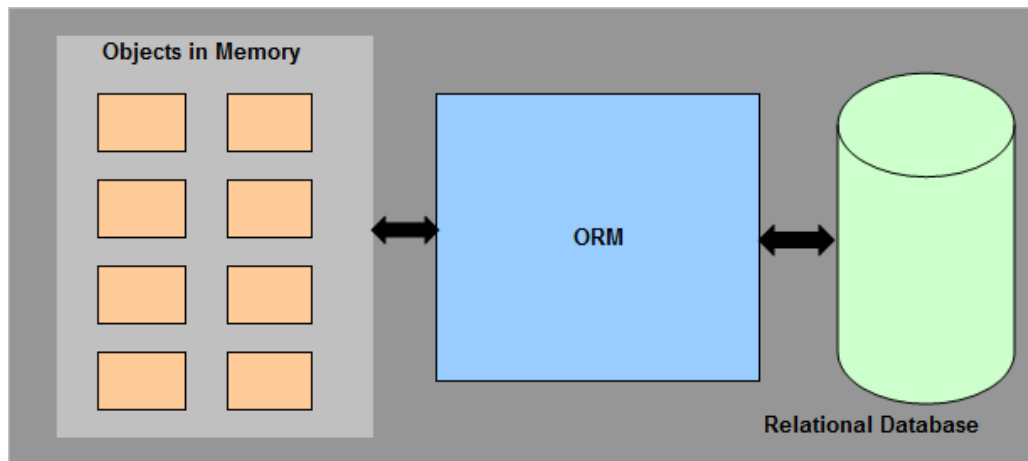
Entity Framework được ra mắt lần đầu tiên vào năm 2008, giúp cho việc truy vấn cơ sở dữ liệu quan hệ trở nên đơn giản, dễ học, giúp giảm bớt thời gian phát triển dự án xuống đáng kể.

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 1.5 Mô tả cách thức hoạt động của Entity Framework*

### **1.3.3. Phương thức HTTP**

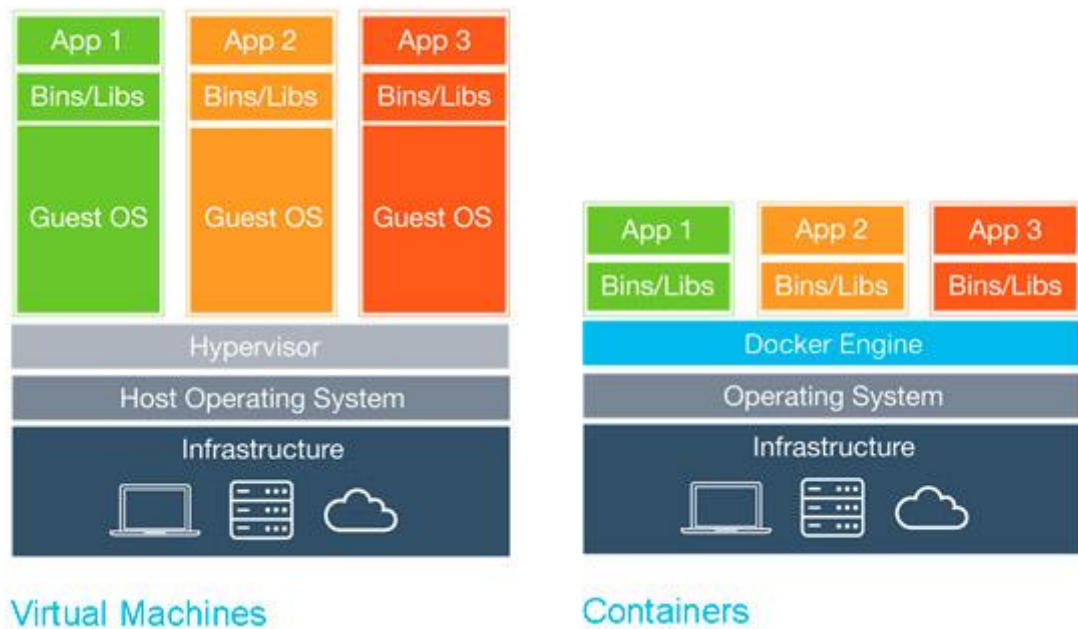
**HTTP** (Tiếng Anh: **HyperText Transfer Protocol** - Giao thức truyền tải siêu văn bản) là một trong năm giao thức chuẩn của mạng Internet, được dùng để liên hệ thông tin giữa Máy cung cấp dịch vụ (Web server) và Máy sử dụng dịch vụ (Web client) trong mô hình Client/Server dùng cho World Wide Web-WWW,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

## **1.4. Docker**

### **1.4.1. Khái niệm**

Docker - đây là một công cụ tạo môi trường được packaging (Container) trên máy tính độc lập mà không làm tác động tới môi máy, môi trường trong Docker sẽ chạy độc lập. Mỗi containers chứa ứng dụng gồm chứa các thư viện riêng, nhưng kernel được chia sẻ với các containers khác. Mỗi kernel được chạy trong các môi trường độc lập với nhau. Docker infrastructure chạy được trên mọi máy tính, đám mây hay nền tảng nào. Vì vậy bạn có thể vận chuyển ứng dụng đến bất cứ đâu và không cần phải quan tâm đến môi trường phát triển, thiếu thư viện, ... Tựu chung lại thì Docker rất tốt tại việc xây dựng và chia sẻ Disk Image qua hệ thống Docker Index, là một phần mềm quản lý cơ sở hạ tầng, làm việc tốt với với các công cụ quản lý file config (vd: Chef, Puppet), sử dụng btrfs để giám sát các file hệ thống và có thể được chia sẻ với user khác. (Như cách

hoạt động của Git) và có một bộ kho trung tâm của các Disk Images (có thể được public hoặc private), điều này cho phép bạn dễ dàng chạy trên nhiều hệ điều hành khác nhau (Ubuntu, Centos, Fedora, Gentoo).



*Hình 1.6 Mô tả cách thức hoạt động của Docker*

#### **1.4.2. Lợi ích của Docker**

- Thời gian khởi động nhanh
- Gói gọn mọi environment cần thiết trong một container, có thể chạy container trên mọi môi trường win, linux, macos, ...
- Tốc độ build nhanh hơn.
- Dễ dàng thiết lập môi trường làm việc khi có vấn đề, sự cố với server.
- Giữ cho workspace sạch sẽ hơn
- Các container có thể chạy độc lập
- Có thể chạy nhiều container trên cùng một máy



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

### **2.1. Đặt vấn đề**

Hệ thống cần giải quyết được bài toán tối ưu quá trình quản lý dự án của công ty. Hiện tại công ty đang vận hành các dự án theo mô hình scrum-agile nên 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. Các tiến trình có thể tùy chỉnh và thay đổi để phù hợp với từng dự án. Đồng thời việc tạo mới, quản lý các công việc phải hợp lý, logic.

Đồng thời, hệ thống cũng phải giải quyết được bài toán quản lý thời gian biểu, chấm công của công ty. Nhân viên có thể quản lý được thời gian biểu của mình qua việc lưu lại thời gian làm việc trong ngày, tuần, tháng của mình. Admin có thể quan sát, theo dõi được lịch làm việc và thời gian biểu của toàn bộ nhân viên trong công ty.

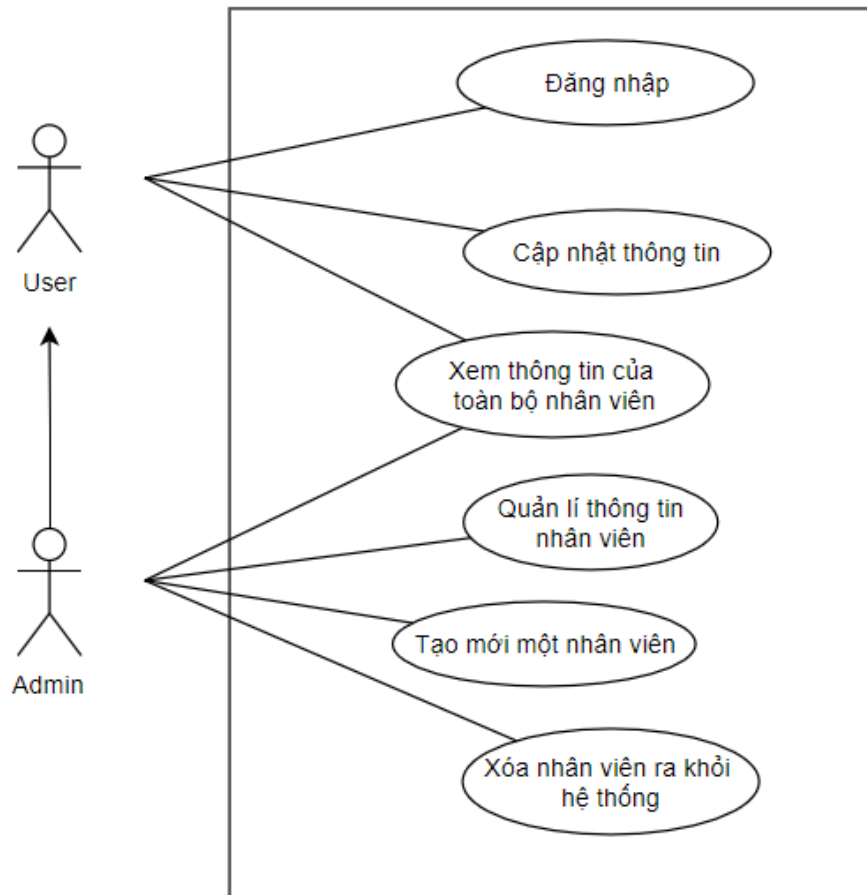
Việc xem xét thông tin cá nhân của nhân viên cũng phải dễ dàng, dễ tìm kiếm, lọc,... để thuận lợi hơn cho qui trình quản lý.

### **2.2. Cách chức năng chính**

#### **2.2.1. Quản lý thông tin nhân viên**

Để thuận lợi cho việc truy xuất cũng như quản lý thông tin các nhân viên hiện thời của công ty, hệ thống sẽ cho phép:

- Nhân viên quản lý, cập nhật thông tin cá nhân của mình.
- Xem thông tin của những nhân viên hiện có trong công ty.
- Tìm kiếm nhân viên theo tên, chức vụ hiện thời, hoặc bằng các kỹ năng hiện có của nhân viên.
- Admin có thể tạo mới tài khoản cho nhân viên mới, chỉnh sửa, khóa hoặc xóa tài khoản của nhân viên ra khỏi hệ thống

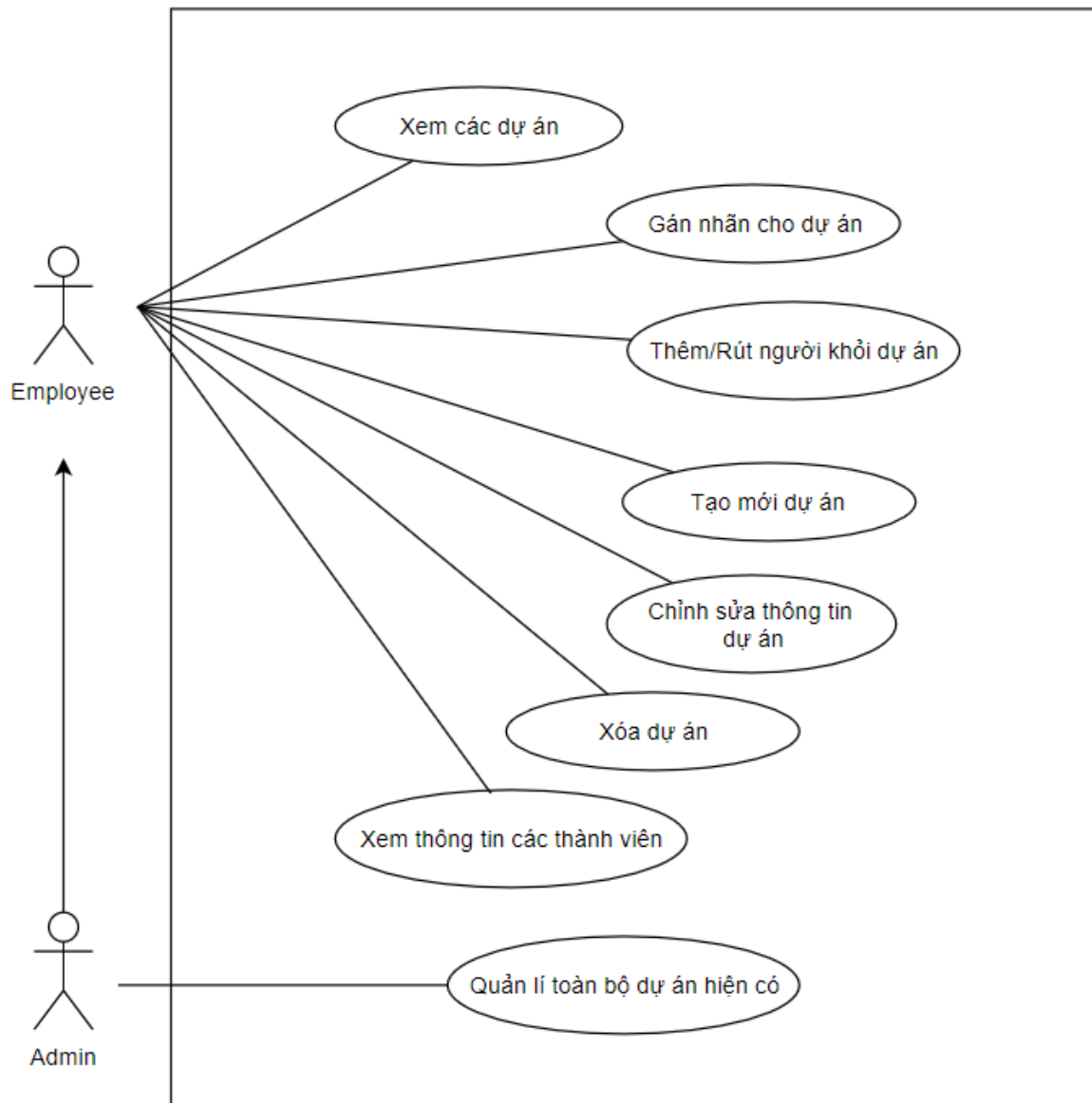


Hình 2.1 Sơ đồ use-case cho module quản lý nhân viên

### 2.2.2. Quản lý dự án

Hệ thống sẽ cung cấp công cụ để quản lý các dự án của công ty:

- Nhân viên có thể xem các dự án mà mình đang tham gia
- Nhân viên có thể tạo mới một dự án
- Nhân viên có thể thêm người vào dự án
- Nhân viên có thể thêm các nhãn màu cho dự án để dễ phân biệt
- Nhân viên có thể chỉnh sửa thông tin của dự án
- Nhân viên có thể xem hồ sơ của các thành viên trong dự án
- Giám đốc / trưởng bộ phận (Admin) có thể quản lý, giám sát toàn bộ dự án của công ty

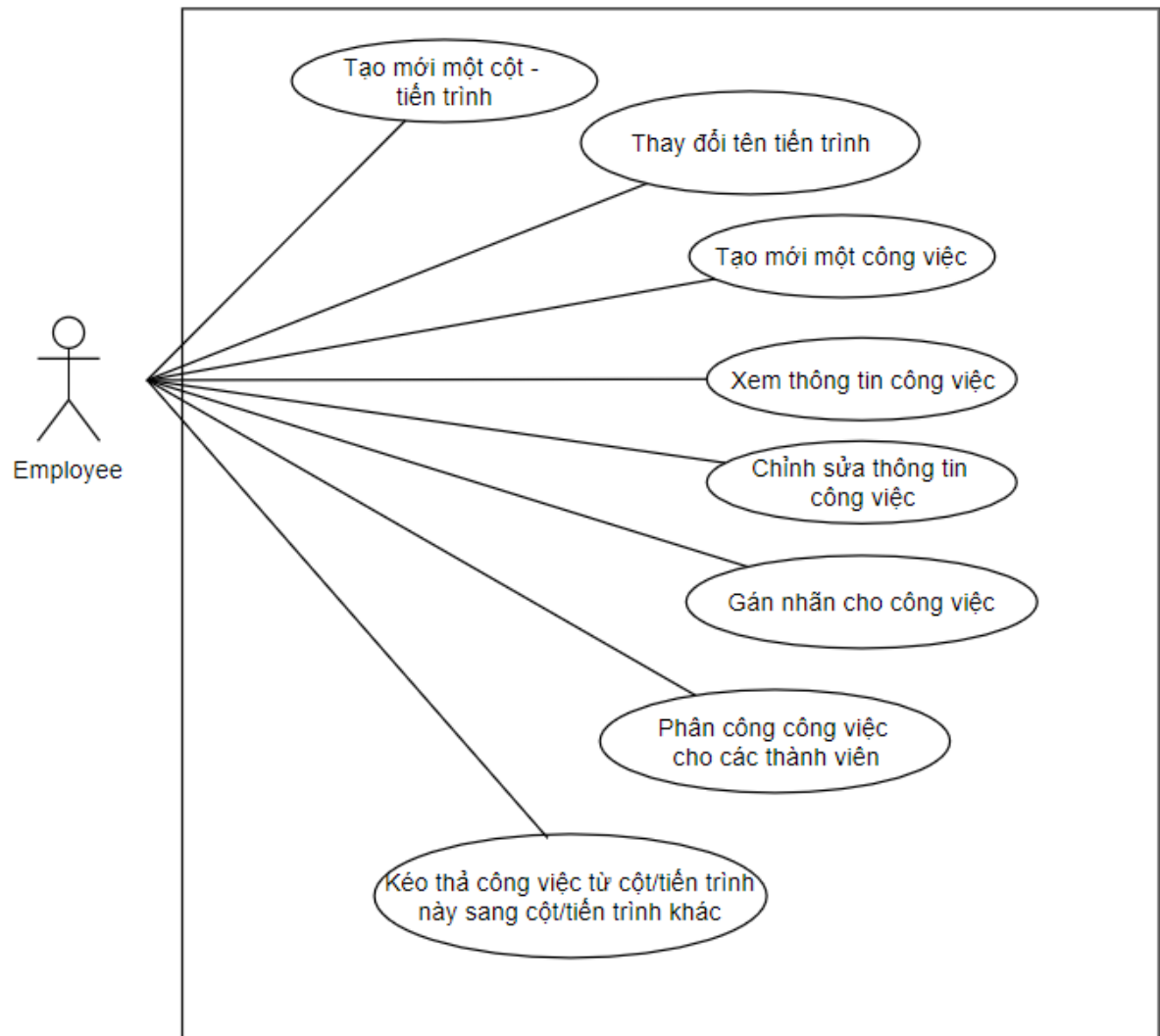


*Hình 2.2 Sơ đồ use-case cho module quản lý dự án*

### **2.2.3. Quản lý công việc**

Hệ thống sẽ cung cấp công cụ để quản lý tiến độ công việc các dự án của công ty theo dạng scrum board:

- Mọi thành viên trong nhóm có quản lý tiến độ công việc của dự án bằng cách thêm các luồng thực hiện công việc, thêm mới và quản lý các nhiệm vụ, công việc.
- Thành viên có thể giao, nhận các công việc.
- Thành viên có thể quan sát tiến độ của dự án thông qua tổng thời gian ước lượng của toàn bộ công việc và lượng thời gian thực tế đã tiêu tốn cho những công việc trong dự án.

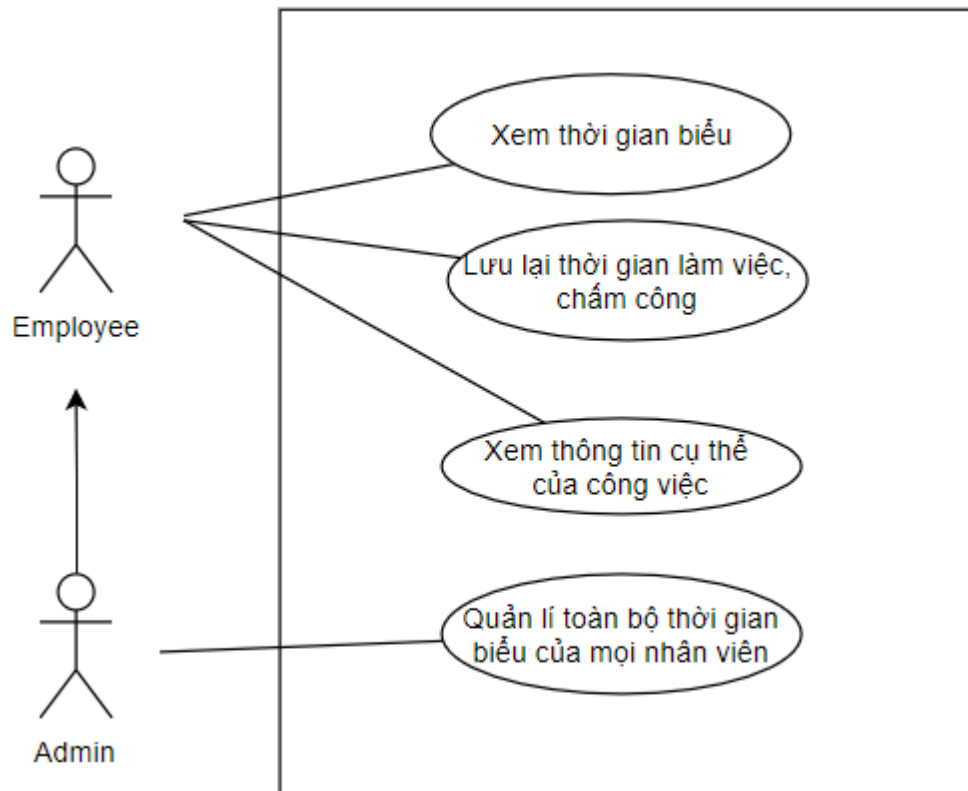


*Hình 2.3 Sơ đồ use-case cho module quản lý tiến độ công việc trong dự án*

#### **2.2.4. Quản lý thời gian làm việc**

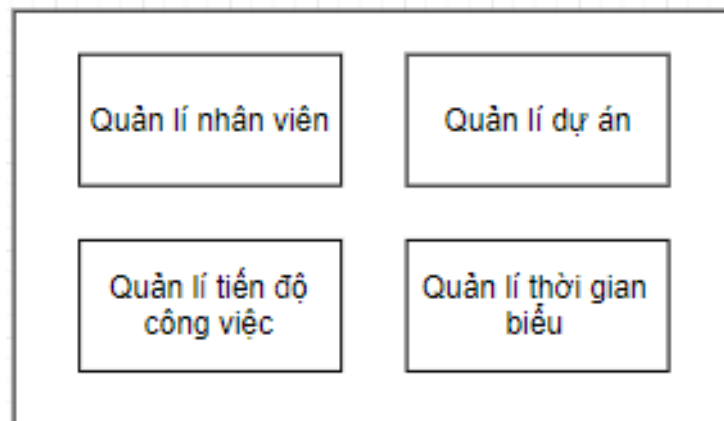
Để thuận lợi hơn trong việc đánh giá hiệu suất làm việc của nhân viên, hệ thống sẽ cho phép:

- 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.
- Nhân viên có thể xem toàn bộ thời gian làm việc của mình
- Nhân viên có thể xem thông tin của công việc đã làm ngay tại thời gian biểu
- Giám đốc / trưởng bộ phận có thể quan sát thời gian làm việc của mọi nhân viên.



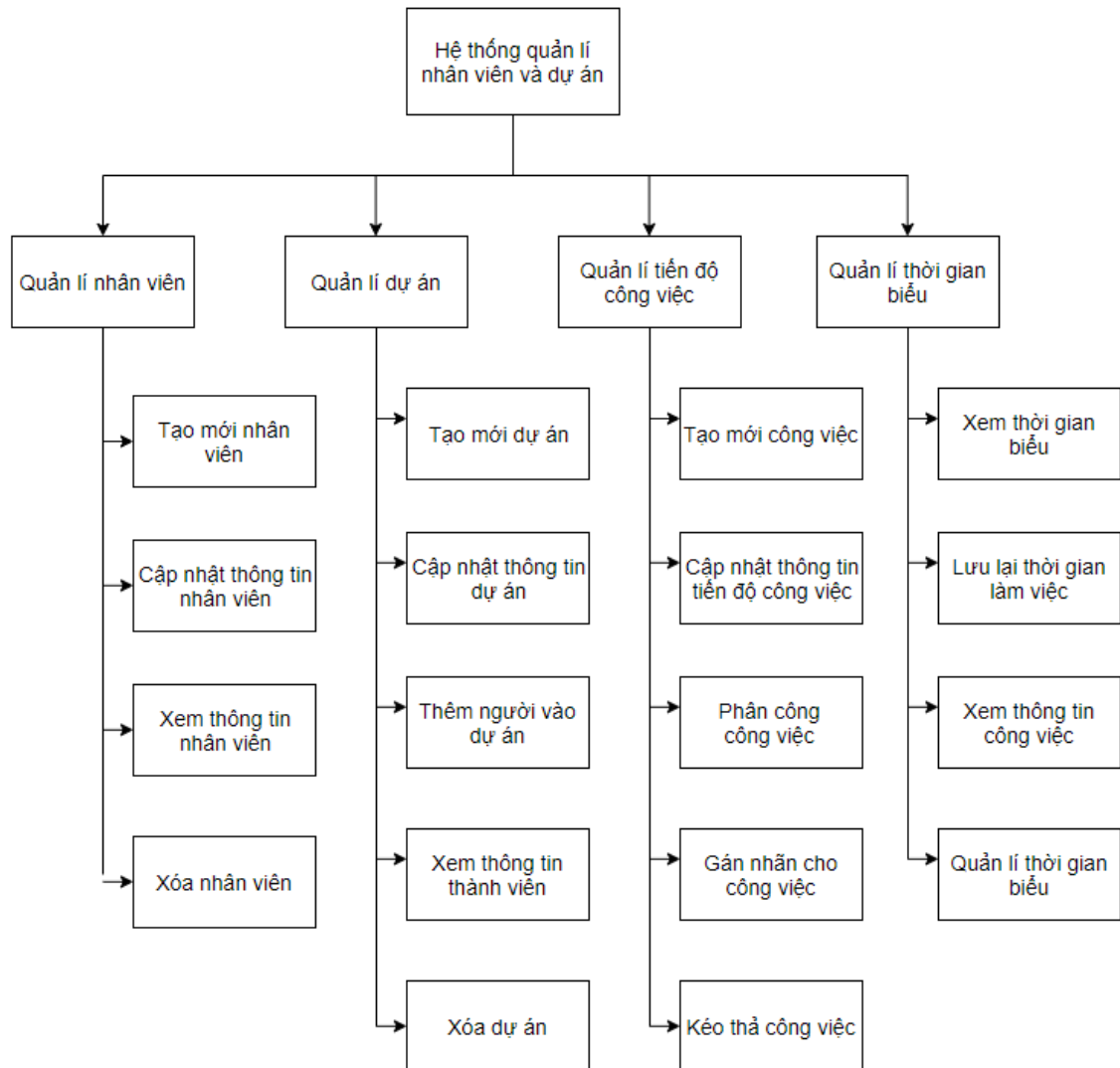
*Hình 2.4 Sơ đồ use-case module quản lý thời gian làm việc*

### **2.3. Các module chính của hệ thống**



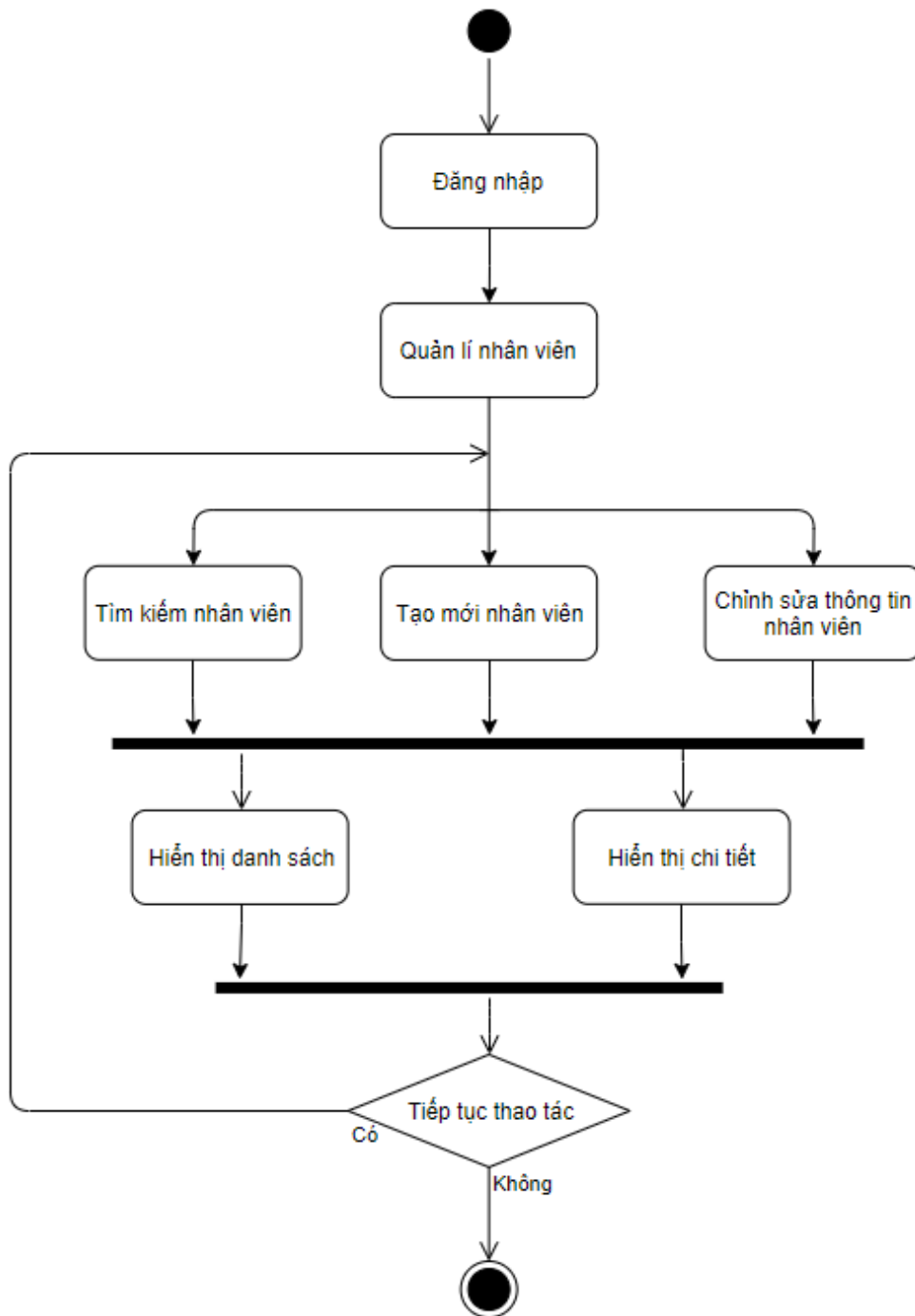
*Hình 2.5 Các module chính của dự án*

## 2.4. Sơ đồ phân rã chức năng

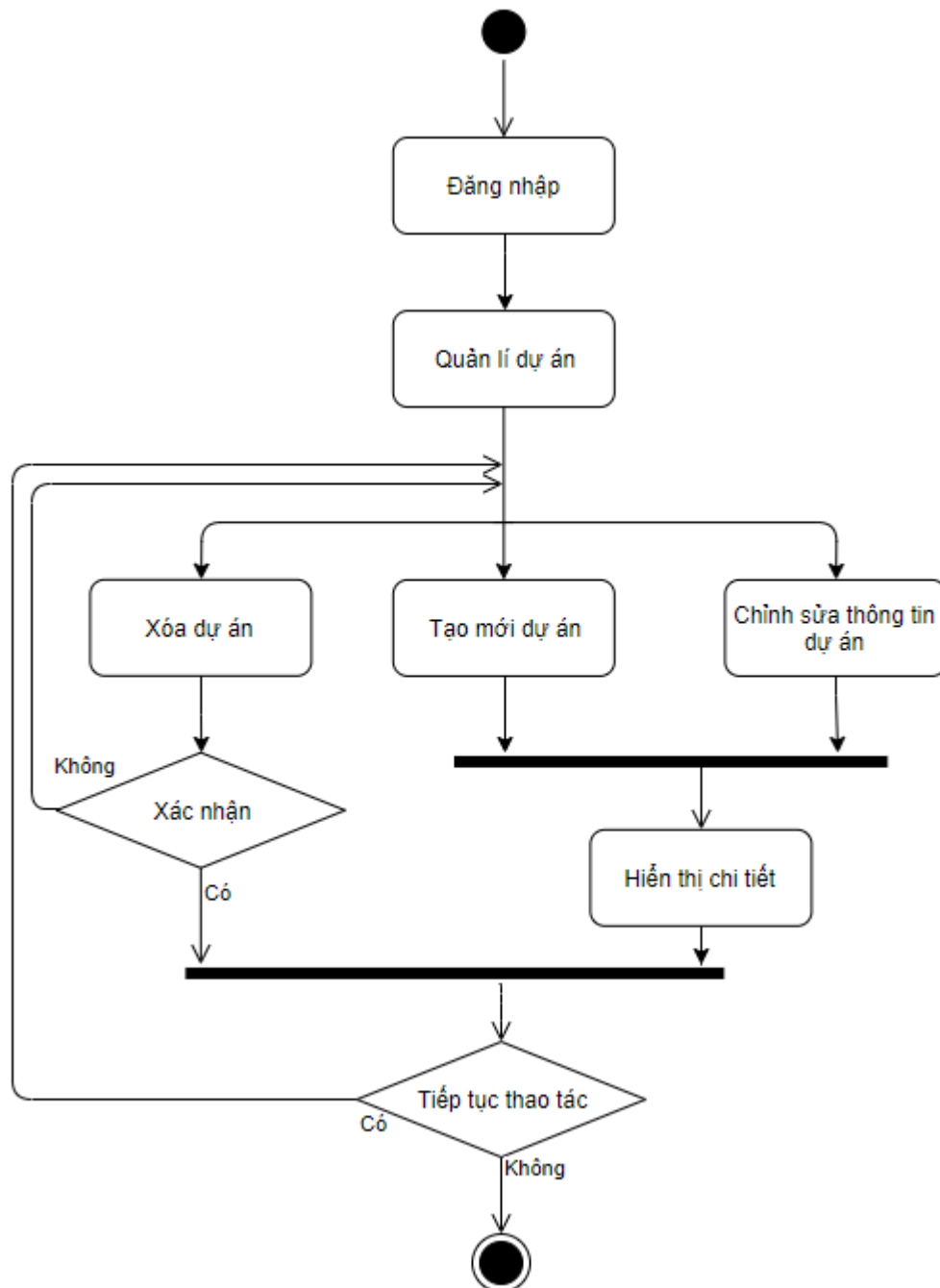


*Hình 2.6 Sơ đồ phân rã chức năng*

## 2.5. Sơ đồ hoạt động

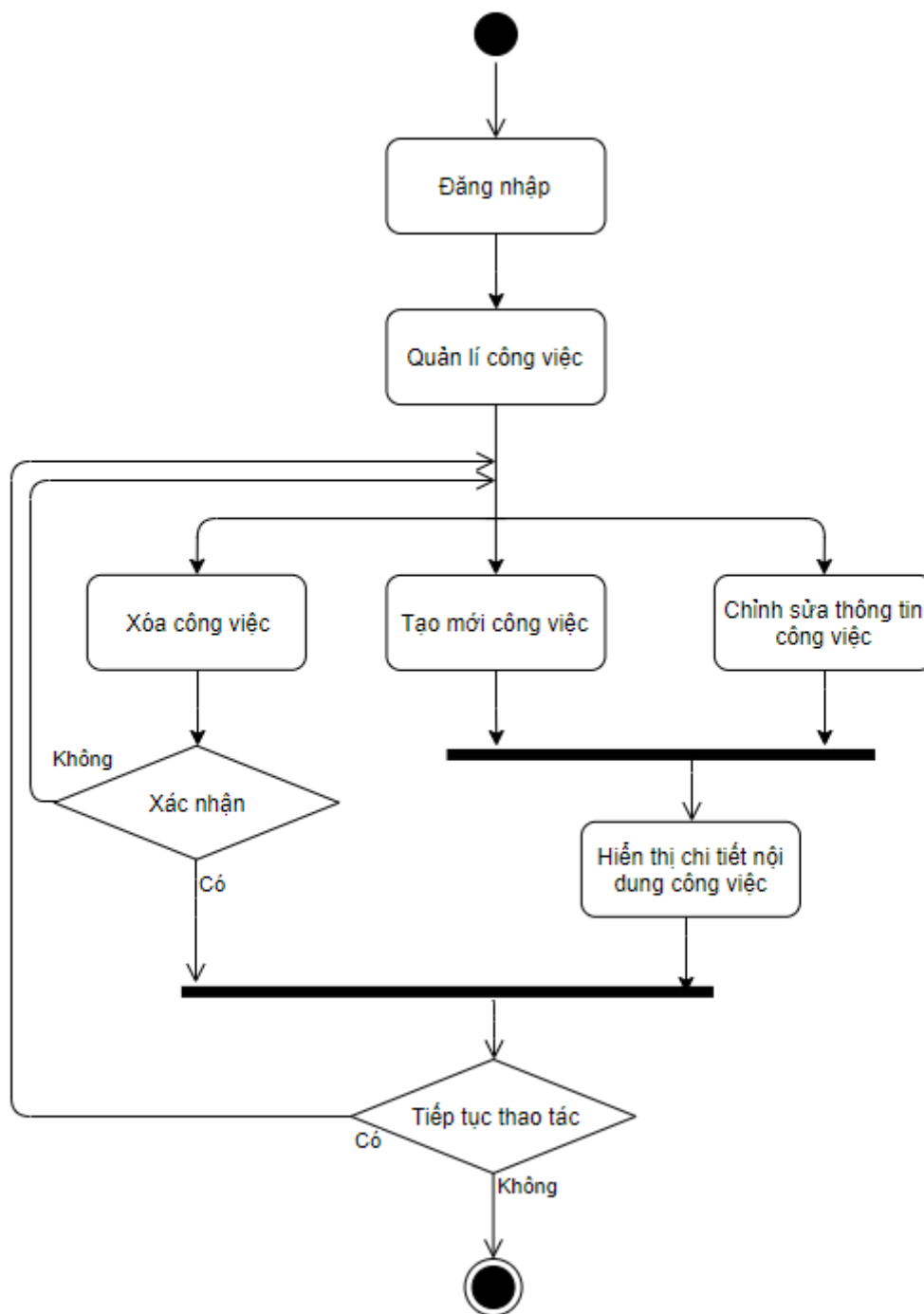


*Hình 2.7 Sơ đồ hoạt động của module quản lý nhân viên*

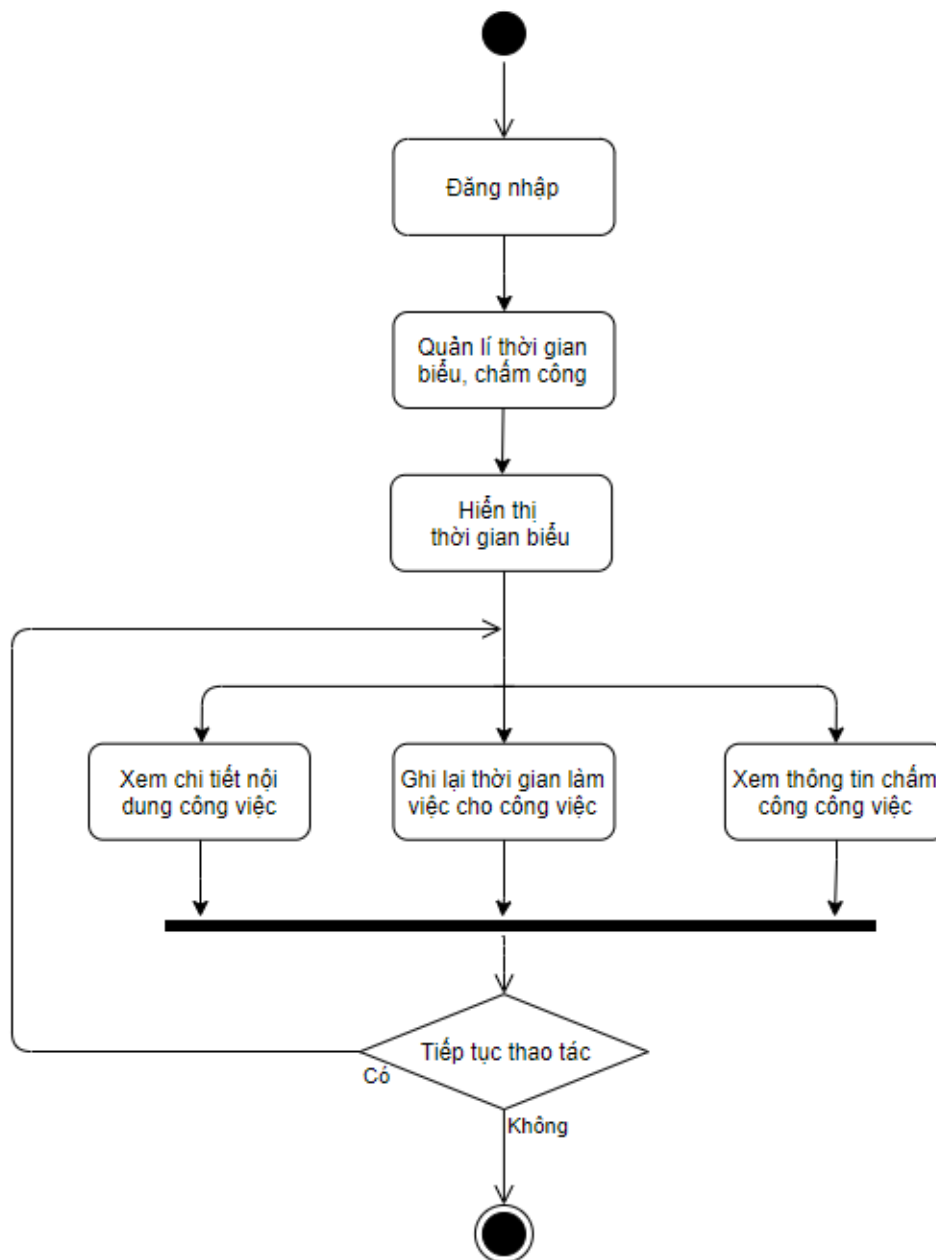


Hình 2.8 Sơ đồ hoạt động của module quản lí dự án



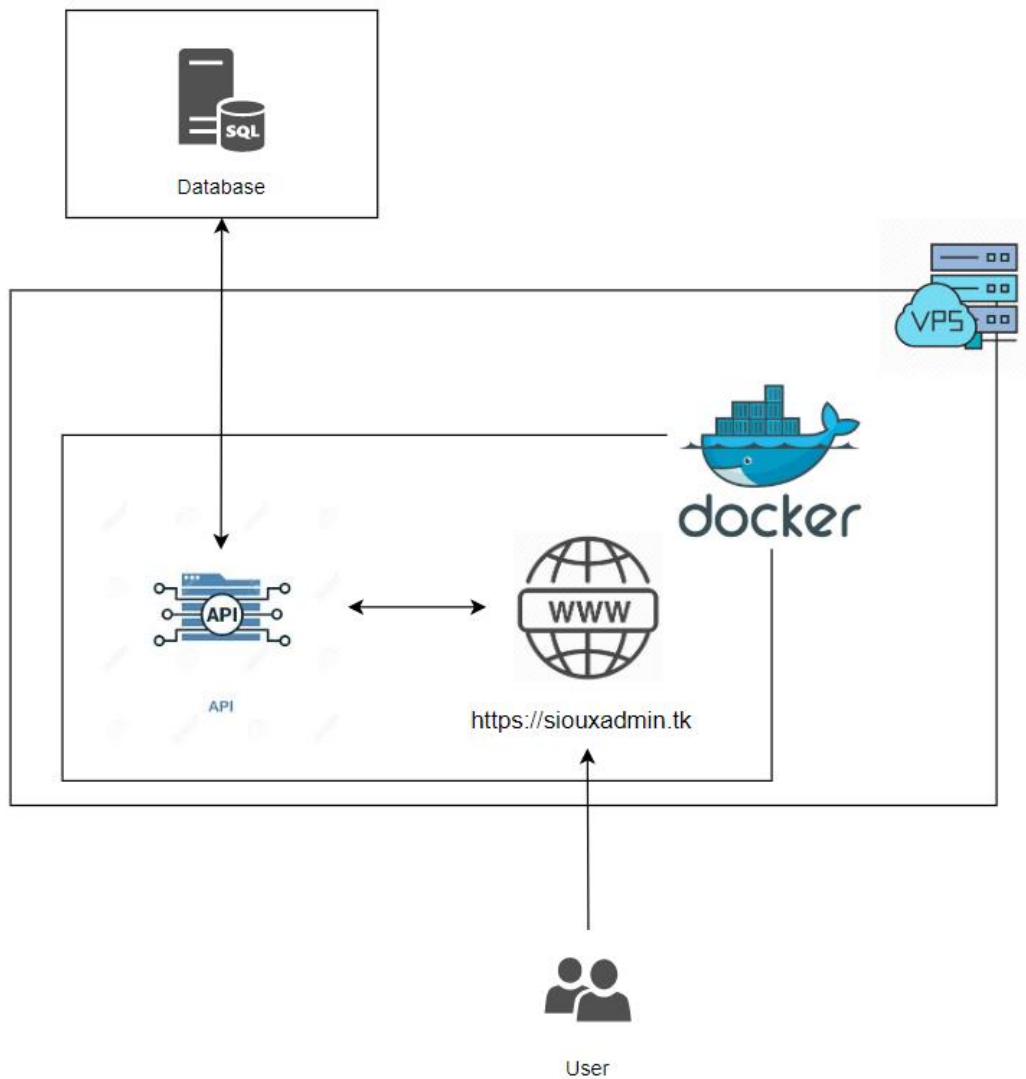


Hình 2.9 Sơ đồ hoạt động của module quản lý tiến độ công việc



Hình 2.10 Sơ đồ hoạt động của module quản lý thời gian biểu, chấm công

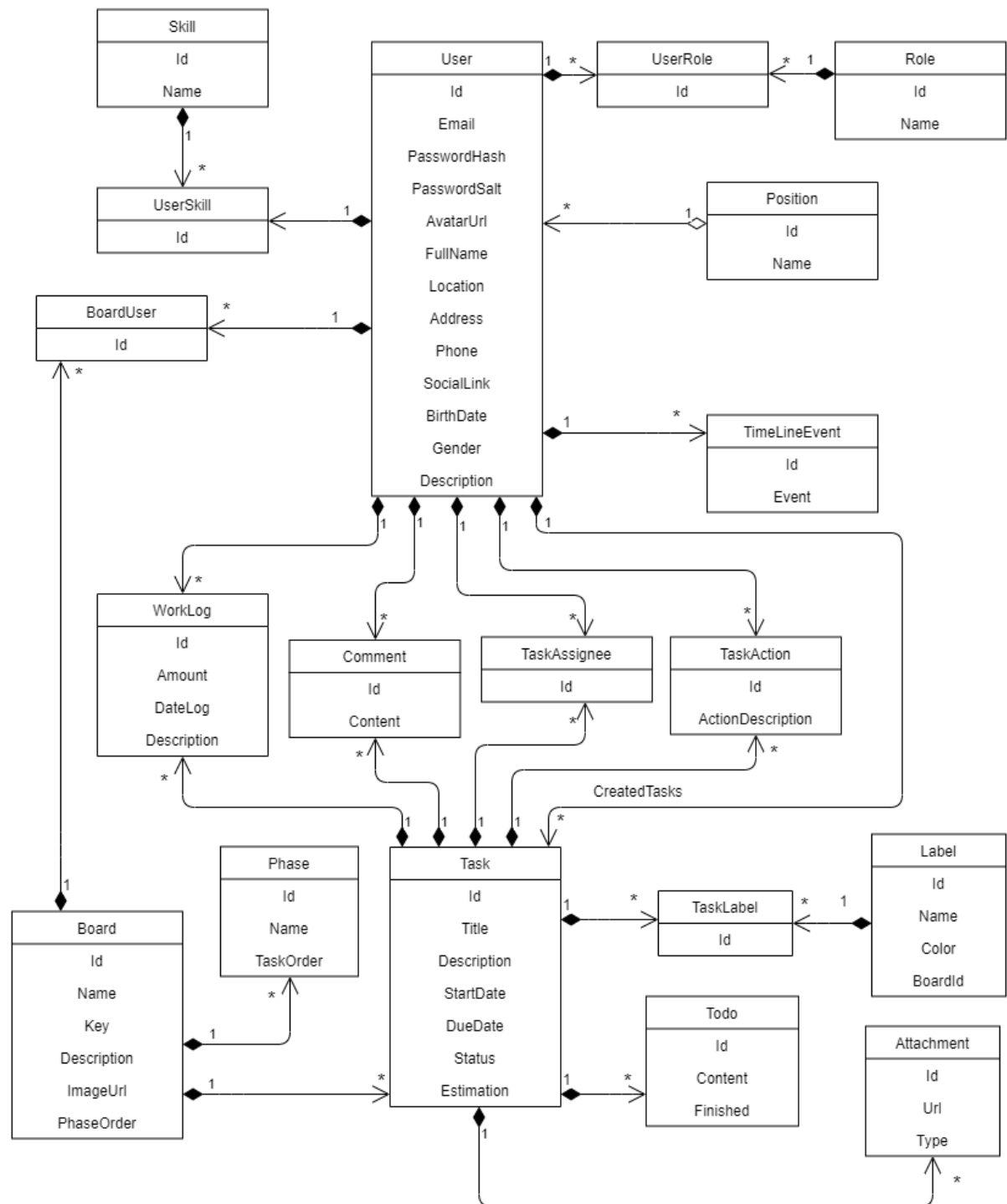
## 2.6. Luồng dữ liệu



*Hình 2.11 Mô tả luồng đi của dữ liệu*

Thông thường, nhân viên sẽ đăng nhập vào hệ thống quản lý qua website. Các thao tác trên website sẽ thông qua API để truy cập đến cơ sở dữ liệu. Sau khi lấy dữ liệu từ cơ sở dữ liệu, API server sẽ trả tín hiệu về cho trình duyệt để hiển thị.

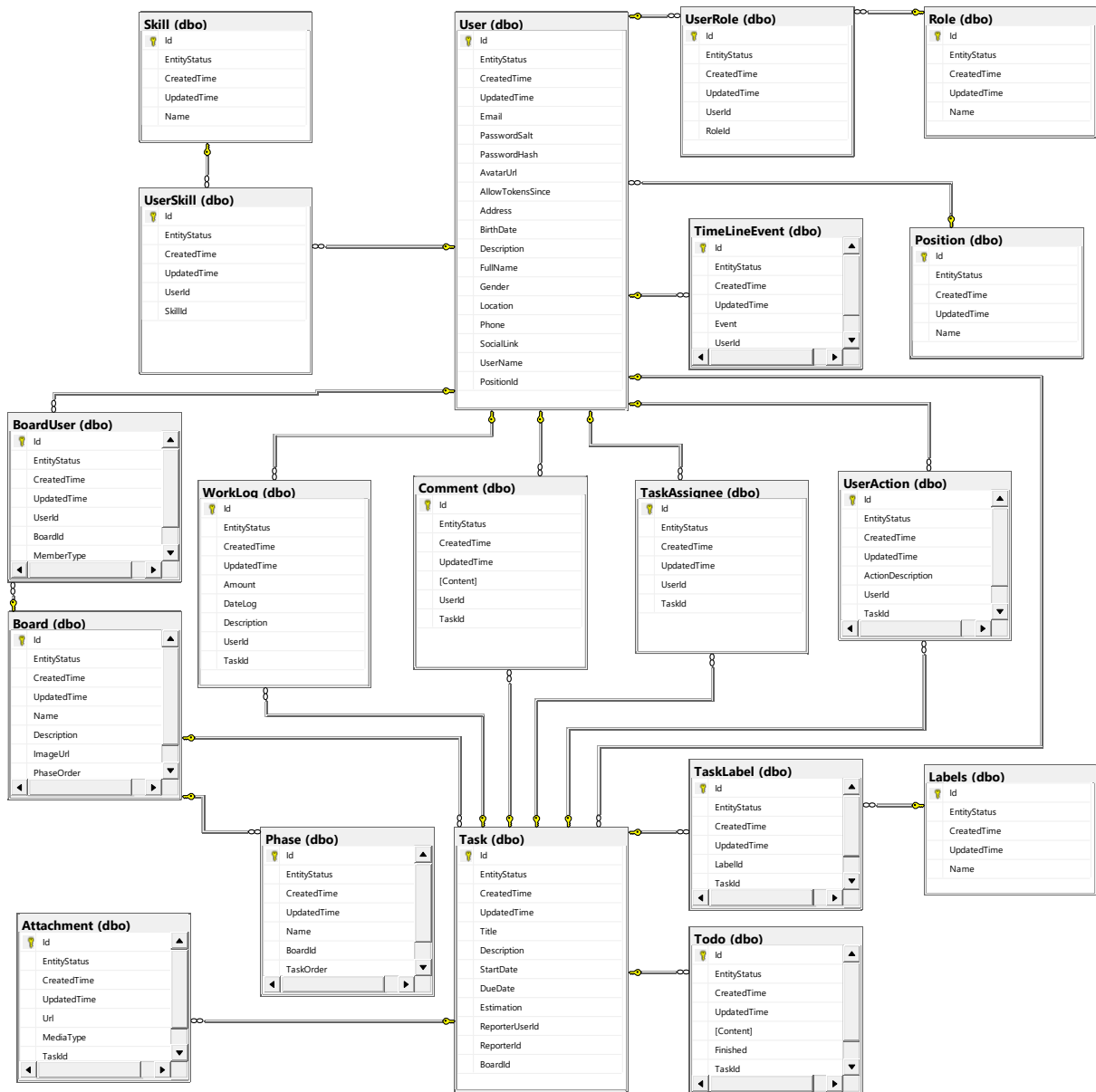
## 2.7. Sơ đồ quan hệ



Hình 2.12 Sơ đồ lớp cho dự án

## 2.8. Thiết kế cơ sở dữ liệu

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



Hình 2.13 Sơ đồ quan hệ, thực thể dữ liệu

## **2.8.2. Đặc tả các bảng dữ liệu**

*Bảng 2.1 Bảng User*

<b>Tên</b>	<b>Mô tả</b>	<b>Data Type</b>	<b>Note</b>
<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 2.2 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 2.3 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 2.4 Bảng Skill*

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 2.5 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 2.6 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 2.7 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 2.8 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 2.9 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 2.10 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 2.11 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 2.12 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 2.13 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 2.14 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 2.15 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 2.16 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 2.17 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 2.18 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 2.19 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

## **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 Pattern: Repository Pattern
- Hệ quản trị cơ sở dữ liệu: SQL Server
- Hệ điều hành: Docker container trên môi trường linux
- Địa chỉ document của website: <https://api.sioxadmin.tk/swagger>

*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: Docker container trên môi trường linux
- Địa chỉ website: <https://sioxadmin.tk>

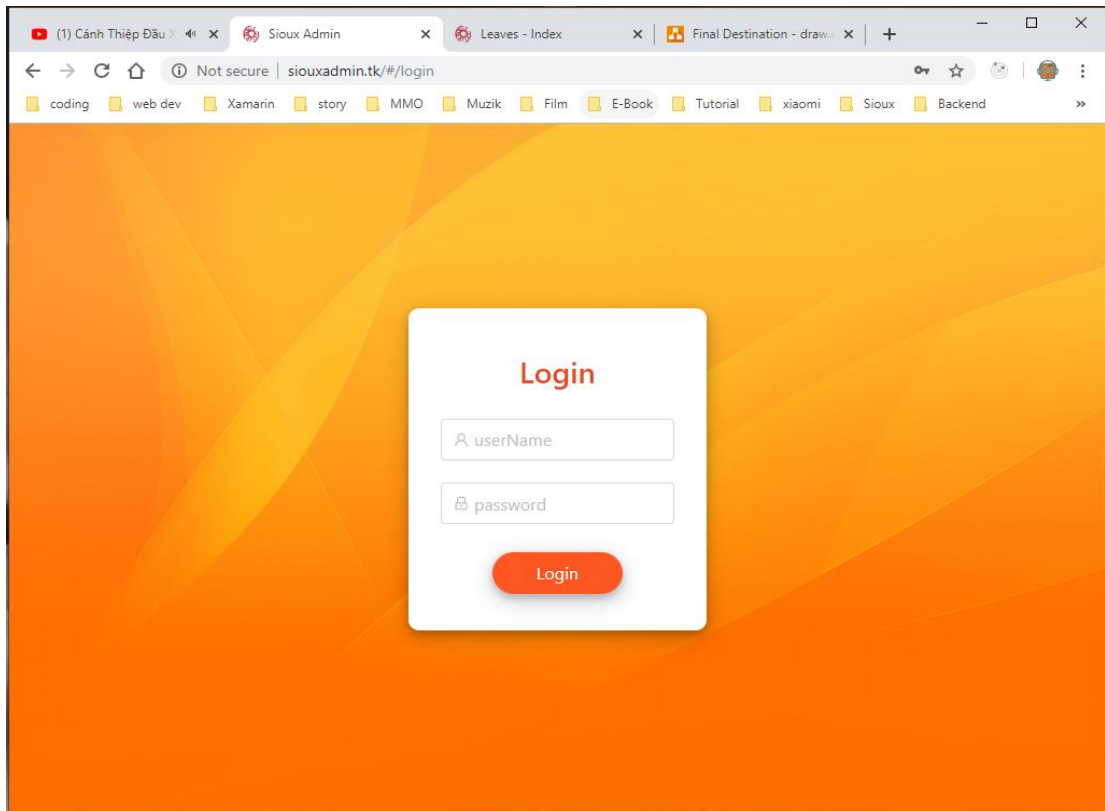
*Môi trường phát triển:*

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

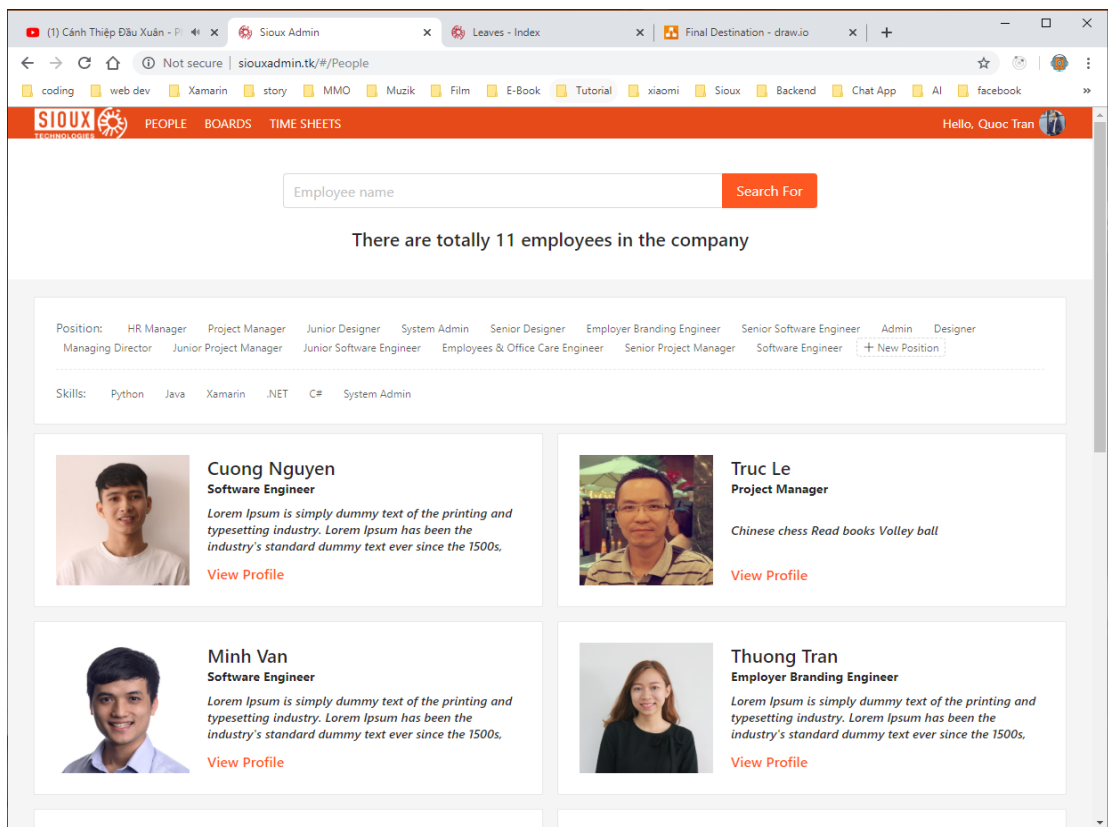
### **1.2. Kịch bản thử nghiệm**

- 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

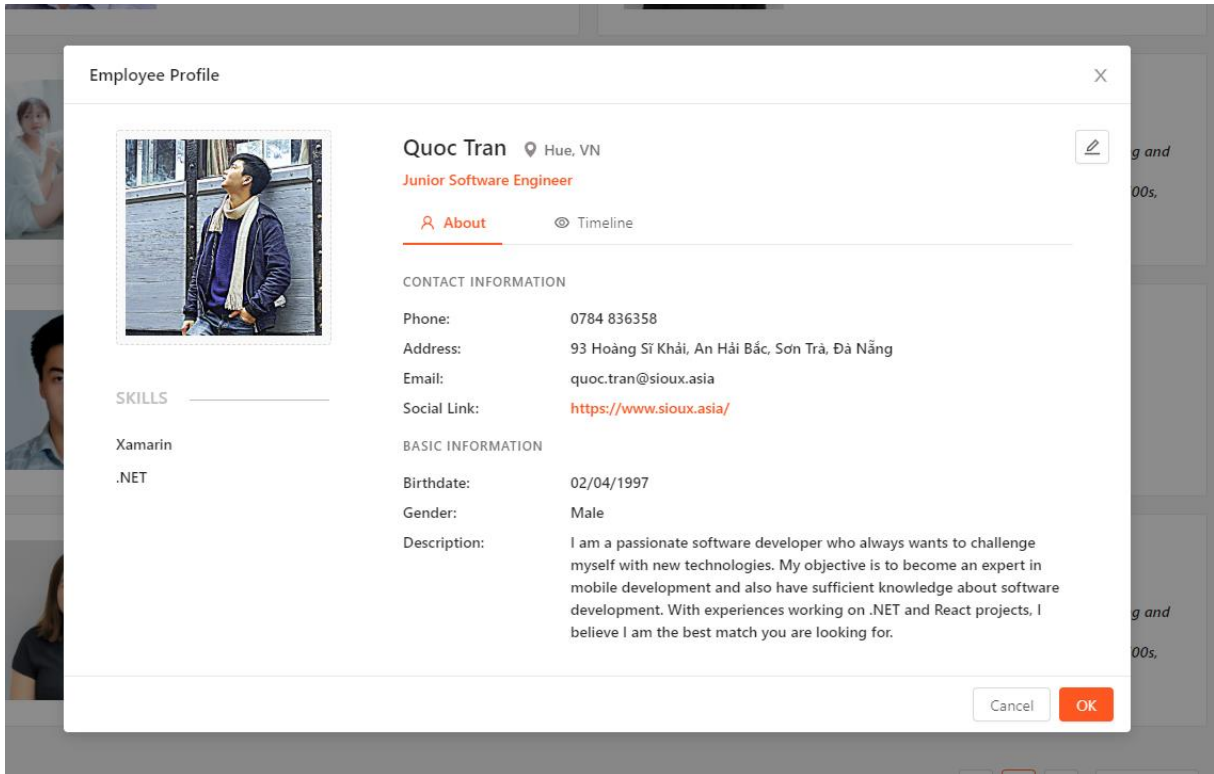
### 3.3. Kết quả thực hiện



*Hình 3.1 Màn hình Login*



*Hình 3.2 Quản lý nhân viên*

A screenshot of an 'Employee Profile' window. On the left, there is a profile picture of a man and a list of skills: 'Xamarin' and '.NET'. The main area on the right displays the employee's details: 'Quoc Tran' with a location pin for 'Hue, VN' and the title 'Junior Software Engineer'. Below this are tabs for 'About' (selected) and 'Timeline'. The 'About' tab shows 'CONTACT INFORMATION' (Phone: 0784 836358, Address: 93 Hoàng Sĩ Khải, An Hải Bắc, Sơn Trà, Đà Nẵng, Email: quoc.tran@sioux.asia, Social Link: https://www.siox.asia/) and 'BASIC INFORMATION' (Birthdate: 02/04/1997, Gender: Male, Description: I am a passionate software developer who always wants to challenge myself with new technologies. My objective is to become an expert in mobile development and also have sufficient knowledge about software development. With experiences working on .NET and React projects, I believe I am the best match you are looking for.). At the bottom right are 'Cancel' and 'OK' buttons.

Employee Profile

Quoc Tran Hue, VN

Junior Software Engineer

About Timeline

CONTACT INFORMATION

Phone: 0784 836358

Address: 93 Hoàng Sĩ Khải, An Hải Bắc, Sơn Trà, Đà Nẵng

Email: quoc.tran@sioux.asia

Social Link: <https://www.siox.asia/>

BASIC INFORMATION

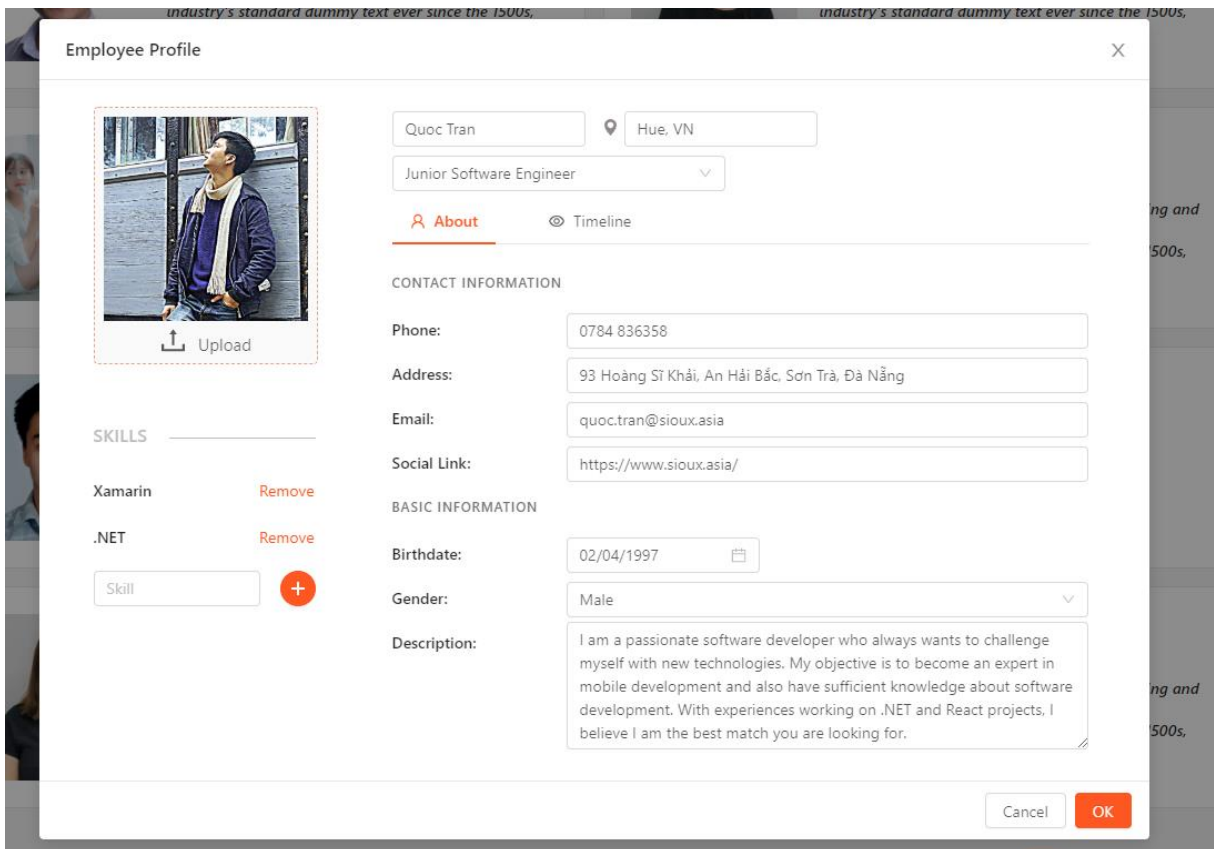
Birthdate: 02/04/1997

Gender: Male

Description: I am a passionate software developer who always wants to challenge myself with new technologies. My objective is to become an expert in mobile development and also have sufficient knowledge about software development. With experiences working on .NET and React projects, I believe I am the best match you are looking for.

Cancel OK

Hình 3.3 Xem thông tin cá nhân của nhân viên

A screenshot of the 'Employee Profile' window in edit mode. The layout is similar to the view mode, but all text fields are now input boxes. The profile picture has an 'Upload' button. The skills list has 'Remove' buttons next to 'Xamarin' and '.NET', and a '+ Skill' button at the bottom. The 'About' tab is selected, showing the same contact and basic information fields as input boxes. The description field is a larger text area. At the bottom right are 'Cancel' and 'OK' buttons.

Employee Profile

Quoc Tran Hue, VN

Junior Software Engineer

About Timeline

CONTACT INFORMATION

Phone: 0784 836358

Address: 93 Hoàng Sĩ Khải, An Hải Bắc, Sơn Trà, Đà Nẵng

Email: quoc.tran@sioux.asia

Social Link: <https://www.siox.asia/>

BASIC INFORMATION

Birthdate: 02/04/1997

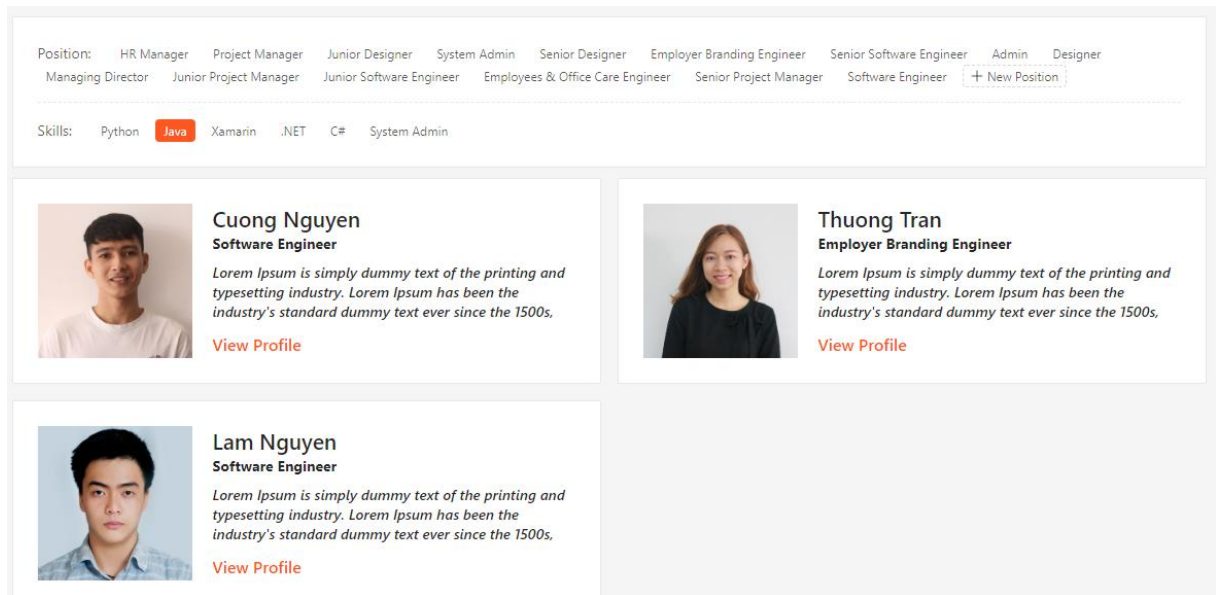
Gender: Male

Description: I am a passionate software developer who always wants to challenge myself with new technologies. My objective is to become an expert in mobile development and also have sufficient knowledge about software development. With experiences working on .NET and React projects, I believe I am the best match you are looking for.

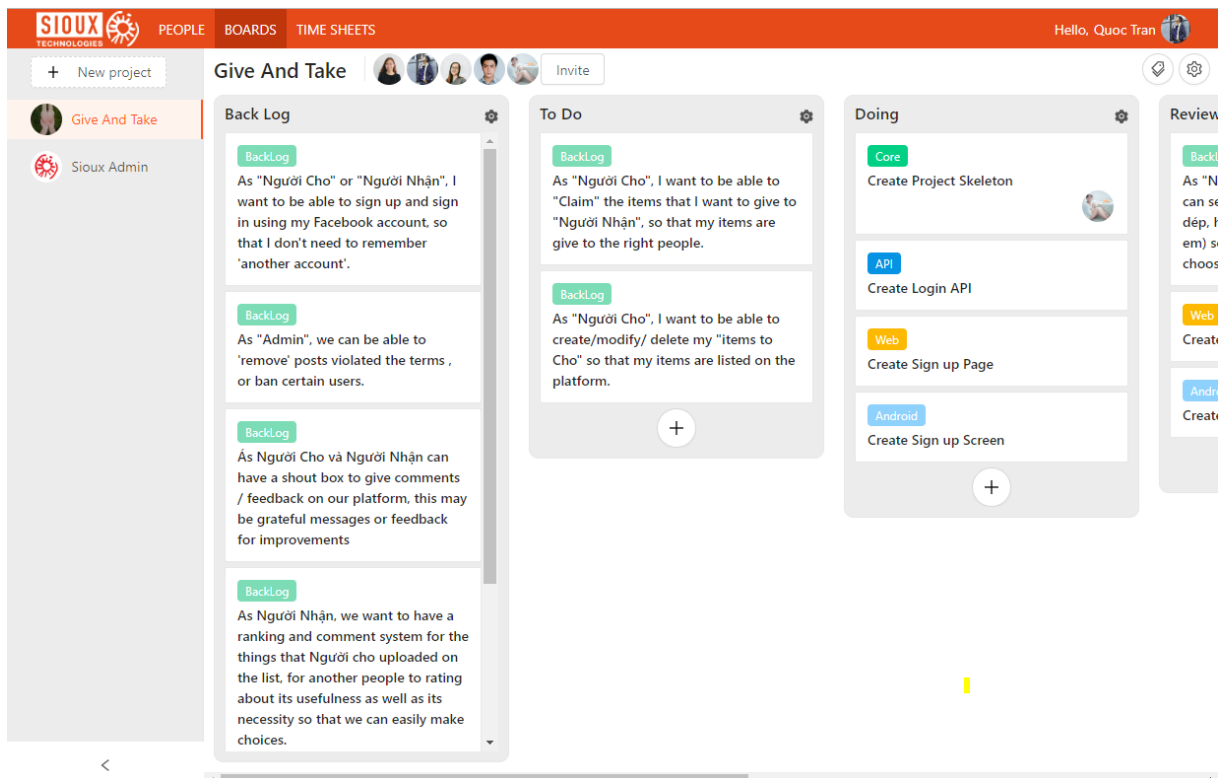
Cancel OK

Hình 3.4 Chỉnh sửa thông tin cá nhân

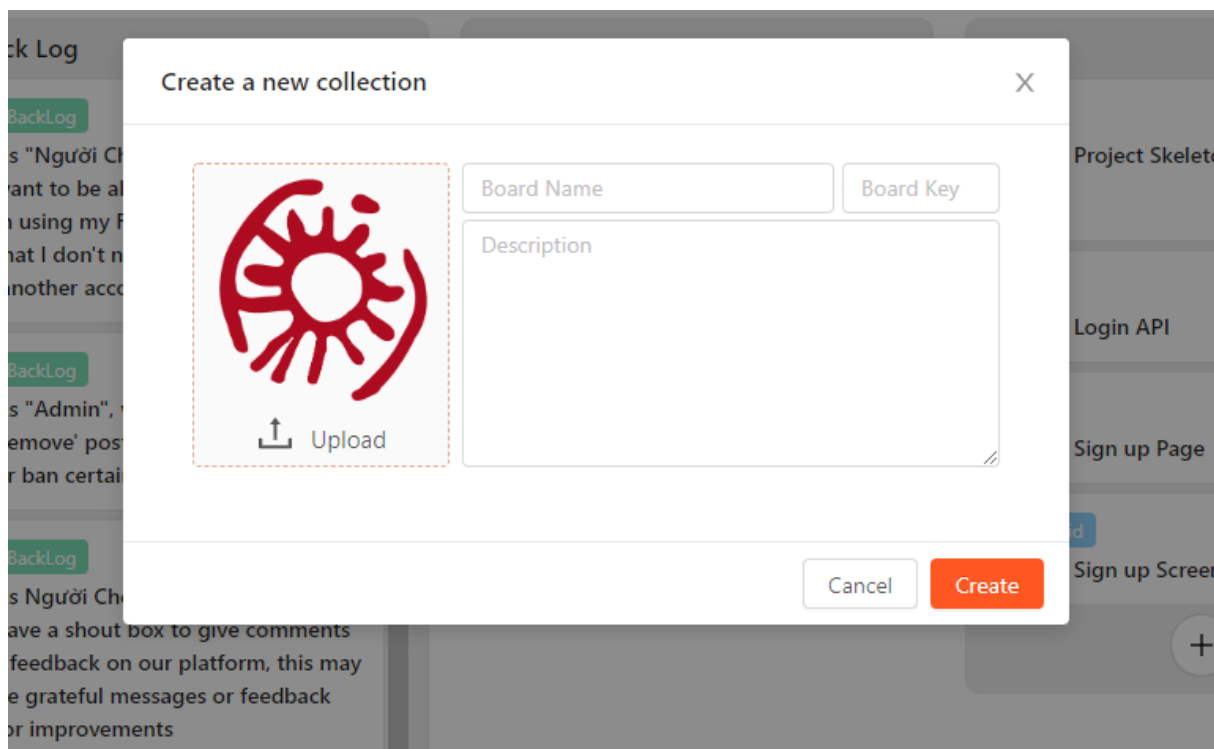




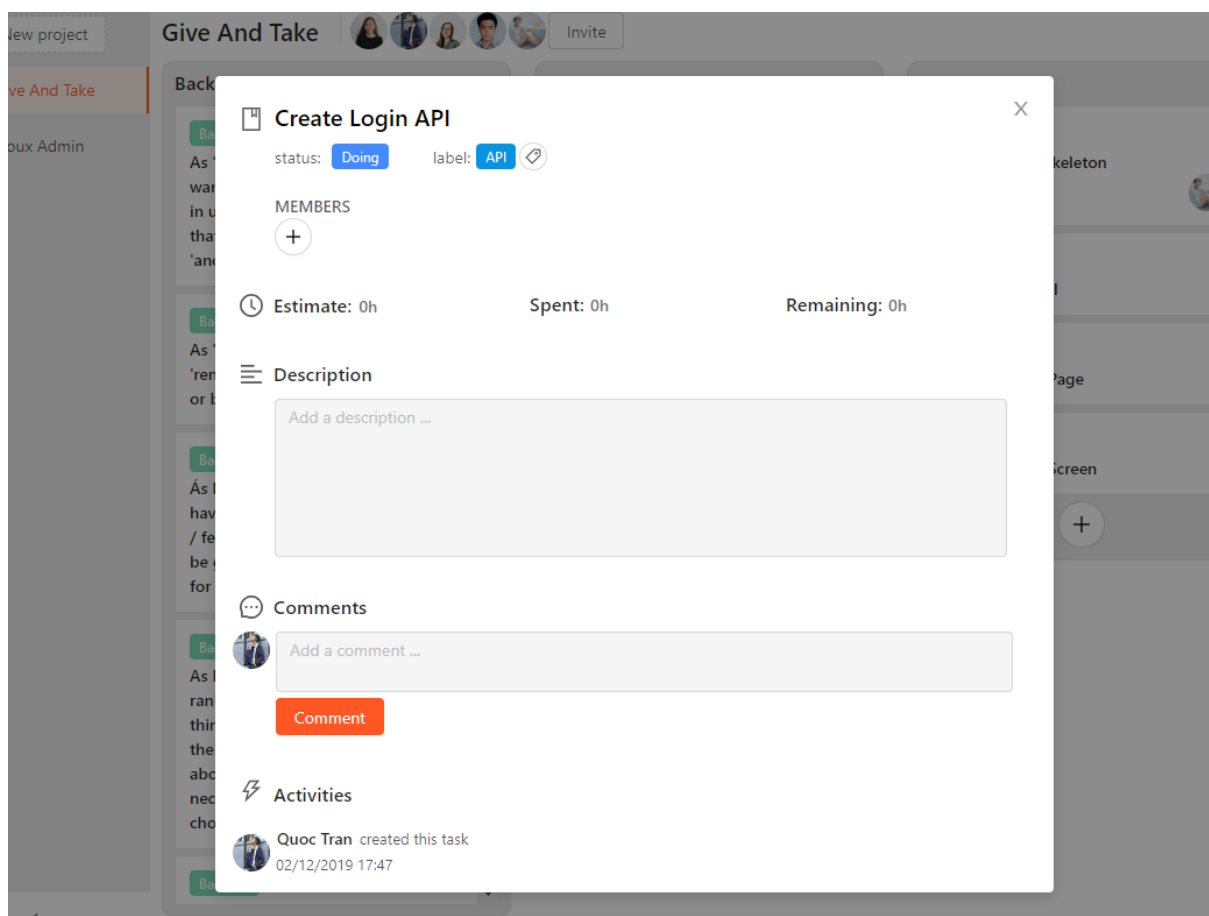
*Hình 3.5 Lọc và tìm kiếm nhân viên dựa trên kỹ năng, vị trí, tên, ...*



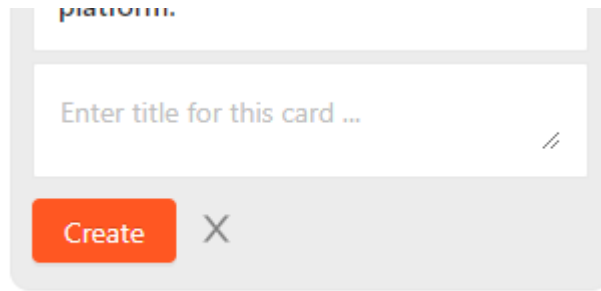
*Hình 3.6 Màn hình quản lý các dự án*



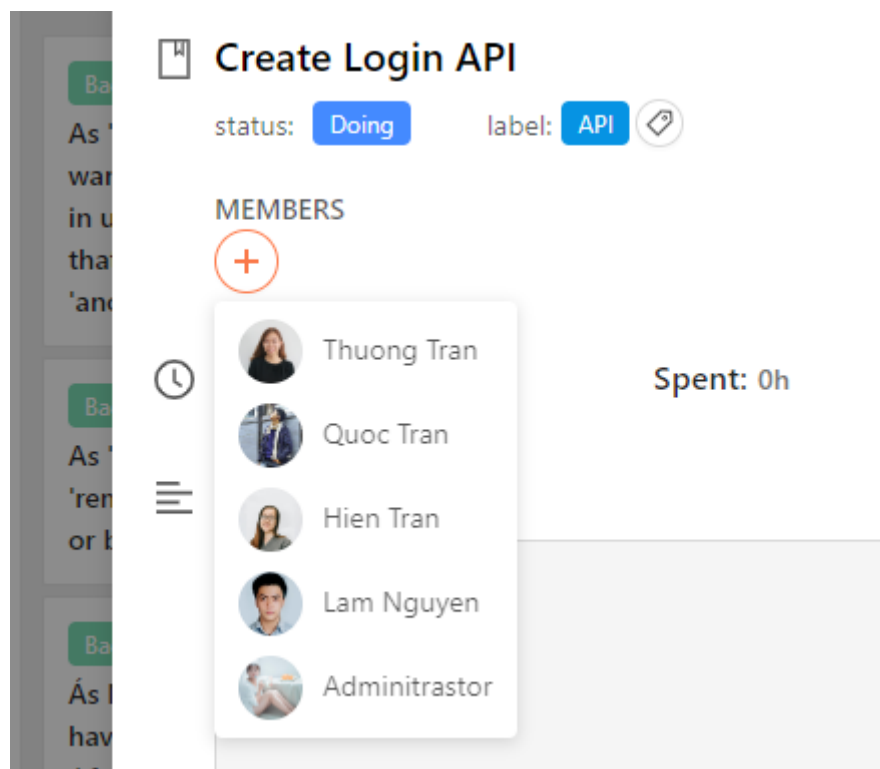
*Hình 3.7 Tạo một dự án mới*



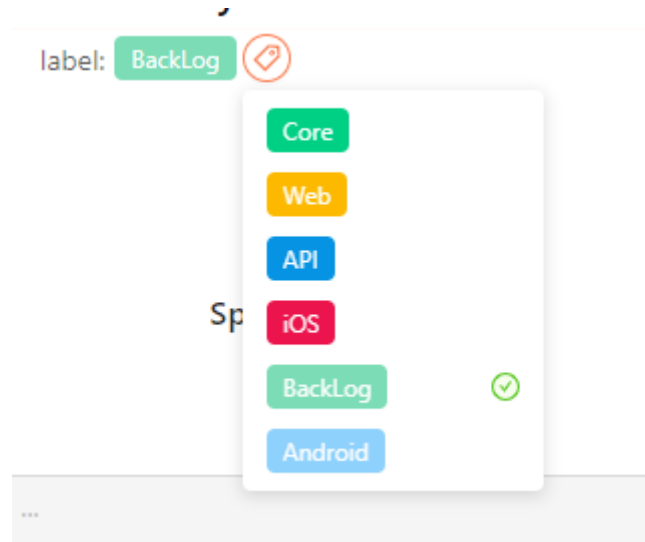
*Hình 3.8 Thông tin chi tiết của một công việc*



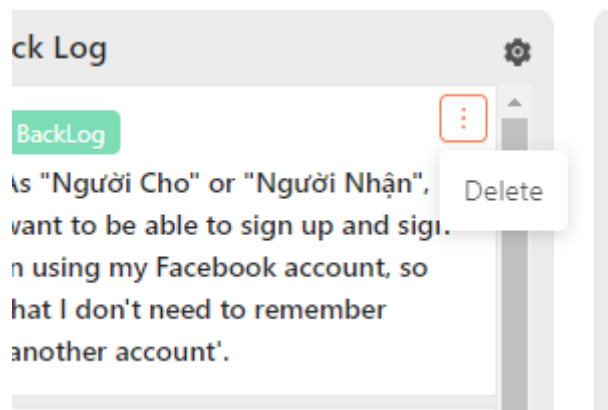
Hình 3.9 Tạo một công việc mới



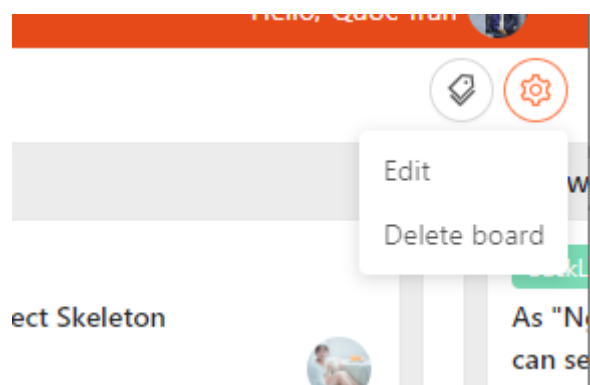
Hình 3.10 Giao việc cho thành viên trong dự án



Hình 3.11 Thêm nhãn cho công việc để dễ quản lý



Hình 3.12 Xóa một công việc



Hình 3.13 Các tùy chọn cho một bản

The screenshot shows the 'Time Sheets' section of the 'Sioux Admin' application. The interface includes a navigation bar with 'PEOPLE', 'BOARDS', and 'TIME SHEETS'. The main content is a table with columns for days of the week (Dec 1 to Dec 10) and a total column. The table lists various task actions with their corresponding time allocations.

Key	Title	Dec 1 Sun	Dec 2 Mon	Dec 3 Tue	Dec 4 Wed	Dec 5 Thu	Dec 6 Fri	Dec 7 Sat	Dec 8 Sun	Dec 9 Mon	Dec 10 Tue	Σ
SA#21	Create task action api	0	0	0	0	0	0	0	0	0	0	0
SA#22	Implement task actions list in task modal	0	0	0	0	0	0	0	0	0	0	0
SA#23	Attach task action when creating a new task	0	0	0	0	0	0	0	0	0	0	0
SA#24	Attach task action when moving a task between...	0	0	0	0	0	0	0	0	0	0	0
SA#25	Attach an action when assign members to a task	0	0	0	0	0	0	0	0	0	0	0
SA#26	Delete task action	0	0	0	0	0	0	0	0	0	0	0
SA#29	Update Document for the 3rd Report	0	0	0	0	0	0	0	0	0	0	0
#	Total	0	0	0	0	0	0	0	0	0	0	0

*Hình 3.14 Quản lý thời gian*

The screenshot shows a modal form for recording time allocation. The form is overlaid on the 'Time Sheets' table. It includes a time input field set to '00:00:00', a description text area, and two buttons: 'Save and close' and 'Cancel'.

	Dec 1 Sun	Dec 2 Mon	Dec 3 Tue	Dec 4 Wed	Dec 5 Thu	Dec 6 Fri
	0	0	0	0	0	0
modal	0					0
a new task	0					0
task between...	0					0
nbers to a task	0					0
ort	0					0
	0					0

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

## 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] Tài liệu lập trình React JS: <https://reactjs.org/docs/getting-started.html>
- [2] Tài liệu lập trình ASP.NET CORE:  
<https://docs.microsoft.com/en-us/aspnet/core/?view=aspnetcore-3.1>
- [3] Tài liệu lập trình docker: <https://docs.docker.com/get-started/>