

ỨNG DỤNG QUẢN LÝ DỰ ÁN

TÓM TẮT

Trong bối cảnh thị trường lao động hiện đại, việc theo dõi tiến độ công việc và quản lý nhân viên trong dự án, ngày càng trở nên quan trọng. Để duy trì sự linh hoạt và hiệu suất cao trong tổ chức, thực sự cần có một giải pháp cho phép giám sát các dự án và nhân sự đã thúc đẩy, nhằm đáp ứng nhu cầu phát triển một ứng dụng quản lý và kiểm tra tiến độ công việc. Điều này không chỉ giúp cải thiện sự minh bạch và giao tiếp trong tổ chức mà còn tăng cường khả năng theo dõi và đánh giá hiệu quả làm việc của từng cá nhân. Đặc biệt, với sự gia tăng của làm việc từ xa và linh hoạt, việc triển khai một ứng dụng quản lý nhân viên và tiến độ công việc hiệu quả là chìa khóa giúp các tổ chức duy trì và tăng cường sự liên kết, đồng thời đảm bảo tiến độ và chất lượng công việc.

Đề tài "**Xây dựng app quản lý nhân viên và tiến độ công việc**" tập trung nghiên cứu chủ yếu đến thiết kế và phát triển, giúp giám sát công việc, quản lý tiến độ dự án và tối ưu quy trình làm việc trong tổ chức.

- **Vấn đề nghiên cứu:**

- Tập trung nghiên cứu chủ yếu đến thiết kế và phát triển giải pháp nhằm mục đích quản lý nhân viên của dự án, giám sát công việc,
- Theo dõi tiến độ công việc và cải thiện hiệu suất làm việc trong tổ chức.
- Đảm bảo dữ liệu của nhân viên được bảo mật và quản lý hiệu quả.

- **Hướng tiếp cận:**

- **Phân tích nhu cầu:** Xử lý các yêu cầu từ các nhân viên, người quản lý để thiết lập các chức năng, giao diện của ứng dụng.
- **Phát triển:** Triển khai lập trình các chức năng thiết yếu của ứng dụng bao gồm quản lý thông tin nhân viên, thông tin nghỉ phép, làm việc ngoài giờ, các công việc được giao.

- **Kiểm thử:** Thực hiện kiểm thử để đảm bảo ứng dụng hoạt động chính xác và hiệu quả.
- **Cách giải quyết vấn đề**
 - Sử dụng front end bằng React native, một framework có thể giúp các lập trình viên chỉ cần code 1 lần nhưng có thể cung cấp cho cả ứng dụng IOS và Android
 - Sử dụng backend bằng C#: một ngôn ngữ cung cấp framework ASP.NET phục vụ viết API theo chuẩn RESTful API dễ dàng.
 - Sử dụng PostgreSQL để lưu trữ thông tin của hệ thống
 - Thực hiện xuất ứng dụng giao diện bằng file APK và publish API bằng IIS.
- **Kết quả và phát hiện cơ bản**
 - **Kết quả:**
 - ◆ Cải thiện hiệu suất làm việc và quản lý thời gian của nhân viên
 - ◆ Tăng sự minh bạch và hợp tác giữa các nhóm làm việc trong công ty.
 - **Phát hiện cơ bản:**
 - ◆ Khả năng mở rộng và tích hợp với các hệ thống khác là cần thiết để tăng cường hiệu quả quản lý
 - ◆ Bảo mật và quyền hạn của nhân viên là một trong những yếu tố cần thiết và quan trọng.

PROJECT MANAGEMENT APPLICATION SUMMARY

In the context of the modern labor market, managing employees and tracking work progress has become increasingly important to maintain flexibility and high productivity within organizations. The necessity for a comprehensive solution that allows for tight project and personnel management has driven the need for developing an employee and work progress management application. This not only helps improve transparency and communication within the organization but also enhances the ability to track and evaluate the effectiveness of individuals' work. Especially with the rise of remote work and flexibility, deploying an effective employee and work progress management application is key to helping organizations maintain and enhance cohesion, while ensuring progress and work quality.

The topic of "**Building an Employee and Work Progress Management App**" focuses mainly on designing and developing solutions aimed at managing employee information, task assignment, project progress management, and optimizing workflow within the organization.

- **Research Issues:**

- Focus on designing and developing solutions aimed at managing employee information, task assignment.
- Project progress management and workflow optimization within the organization.
- Ensuring the security and effective management of employee data.

- **Approach:**

- **Needs analysis:** Gather information, requirements from employees, managers to determine the necessary functions, interfaces of the app.

- **System design:** Based on the requirements, analyze to design user interfaces and functions.
- **Development:** Implement programming essential functions of the application including managing employee information, leave information, overtime work, assigned tasks.
- **Testing:** Perform testing to ensure the application works accurately and efficiently.
- **Problem-solving approach:**
 - Use React Native for the front end, a framework that allows developers to code once but can provide applications for both IOS and Android.
 - Use C# for the backend, a language providing the ASP.NET framework to write RESTful API easily.
 - Use PostgreSQL to store system information.
 - Implement the application interface export by APK file and publish API by IIS.
- **Results and Key Findings:**
 - **Results:**
 - ◆ Improved work efficiency and time management of employees.
 - ◆ Increased transparency and collaboration among work teams in the company.
 - ◆ Help team leaders make more accurate decisions based on the information provided by the application.
 - **Key Findings:**
 - ◆ Employee information and work progress management apps are essential for every company.
 - ◆ The ability to scale and integrate with other systems is essential to enhance management efficiency.
 - ◆ Data security and employee privacy are among the most essential and important factors in the application development process.

MỤC LỤC

DANH MỤC HÌNH VẼ	x
DANH MỤC BẢNG BIỂU	xii
DANH MỤC CÁC CHỮ VIẾT TẮT	xiii
CHƯƠNG 1. MỞ ĐẦU VÀ TỔNG QUAN ĐỀ TÀI	1
1.1 Lý do chọn đề tài	1
1.2 Mục tiêu thực hiện đề tài	1
1.3 Phương pháp nghiên cứu	1
1.4 Mục tiêu kết quả	2
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT	3
2.1 Tổng quan về web api	3
2.2 Tổng quan về react native	4
CHƯƠNG 3. PHÂN TÍCH HỆ THỐNG	6
3.1 Đặc tả yêu cầu	8
3.1.1 Nhóm chức năng quản lý thông tin người dùng	8
3.1.2 Nhóm chức năng quản lý thời gian, công việc của người dùng	9
3.1.3 Nhóm chức năng quản lý các cuộc họp, thảo luận	9
3.2 Yêu cầu chức năng	9
3.3 Yêu cầu phi chức năng	10
3.4 Danh mục các use-case của hệ thống:	10
3.5 Mô hình ERD (Entity Relationship Diagram)	11
3.6 Sơ đồ lớp (Class diagram)	12
3.7 Use case tổng quát	13

3.8 Chi tiết Use case	14
3.8.1 Use case đăng nhập	14
3.8.2 Use case tìm kiếm dự án	17
3.8.3 Use case chấm công	19
3.8.4 Use case đăng ký OT (làm thêm giờ)	23
3.8.5 Use case đăng ký nghỉ phép/công tác	26
3.8.6 Use case xem thống kê nghỉ phép	29
3.8.7 Use case bình luận về dự án	31
3.8.8 Use case quản lý tài khoản user	33
3.8.9 Use case quản lý nhiệm vụ	37
CHƯƠNG 4. THIẾT KẾ GIAO DIỆN	41
4.1 Màn hình đăng nhập	41
4.2 Màn hình trang chủ	41
4.3 Màn hình xem hồ sơ nhân sự	42
4.4 Màn hình đăng ký phép	43
4.5 Màn hình thống kê nghỉ phép	43
4.6 Màn hình đăng ký OT	44
4.7 Màn hình chấm công	44
4.8 Màn hình thống kê chấm công	45
4.9 Màn hình xem nhiệm vụ được giao	45
4.10 Màn hình chi tiết nhiệm vụ	46
TÀI LIỆU THAM KHẢO	47

DANH MỤC HÌNH VẼ

Hình 2.1	Web api.....	3
Hình 2.2	React native.....	5
Hình 3.1	Mô hình quan hệ.....	12
Hình 3.2	Sơ đồ lớp.....	12
Hình 3.3	Use case tổng quát.....	13
Hình 3.4	Use case đăng nhập.....	14
Hình 3.5	Sơ đồ tuần tự use case đăng nhập.....	16
Hình 3.6	Sơ đồ hoạt động use case đăng nhập.....	16
Hình 3.7	Use case tìm kiếm dự án.....	17
Hình 3.8	Sơ đồ tuần tự use case tìm kiếm dự án.....	19
Hình 3.9	Sơ đồ hoạt động Use case tìm kiếm dự án.....	19
Hình 3.10	Use case chấm công.....	20
Hình 3.11	Sơ đồ tuần tự Use case chấm công.....	22
Hình 3.12	Sơ đồ hoạt động Use case chấm công.....	23
Hình 3.13	Use case đăng ký OT.....	23
Hình 3.14	Sơ đồ tuần tự Use case đăng ký OT.....	25
Hình 3.15	Sơ đồ hoạt động Use case đăng ký OT.....	26
Hình 3.16	Use case đăng ký nghỉ phép/công tác.....	26
Hình 3.17	Sơ đồ tuần tự Use case đăng ký nghỉ phép/công tác.....	28
Hình 3.18	Sơ đồ hoạt động Use case đăng ký nghỉ phép/công tác.....	29
Hình 3.19	Use case xem thông kê nghỉ phép.....	29
Hình 3.20	Sơ đồ tuần tự Use case xem thông kê nghỉ phép.....	30

Hình 3.21	Sơ đồ hoạt động Use case xem thống kê nghỉ phép	31
Hình 3.22	Use case bình luận về dự án	31
Hình 3.23	Sơ đồ tuần tự Use case bình luận về dự án	33
Hình 3.24	Sơ đồ hoạt động Use case bình luận về dự án	33
Hình 3.25	Use case quản lý tài khoản user	34
Hình 3.26	Sơ đồ tuần tự Use case quản lý tài khoản user	36
Hình 3.27	Sơ đồ hoạt động Use case quản lý tài khoản user	37
Hình 3.28	Use case quản lý nhiệm vụ	37
Hình 3.29	Sơ đồ tuần tự Use case quản lý nhiệm vụ	40
Hình 3.30	Sơ đồ hoạt động Use case quản lý nhiệm vụ	40
Hình 4.1	Màn hình đăng nhập	41
Hình 4.2	Màn hình trang chủ	41
Hình 4.3	Màn hình xem hồ sơ nhân sự	42
Hình 4.4	4.4Màn hình đăng ký phép	43
Hình 4.5	Màn hình thống kê nghỉ phép	43
Hình 4.6	Màn hình đăng ký OT	44
Hình 4.7	Màn hình chấm công	44
Hình 4.8	Màn hình thống kê chấm công	45
Hình 4.9	Màn hình xem nhiệm vụ được giao	45
Hình 4.10	Màn hình chi tiết nhiệm vụ	46

DANH MỤC BẢNG BIỂU

Bảng 3.1	Danh mục các use case của hệ thống	10
Bảng 3.2	Đặc tả use case đăng nhập	14
Bảng 3.3	Đặc tả use case tìm kiếm dự án	17
Bảng 3.4	Đặc tả use case chấm công	20
Bảng 3.5	Đặc tả use case đăng ký làm thêm giờ	24
Bảng 3.6	Đặc tả use case đăng ký nghỉ phép	27
Bảng 3.7	Đặc tả use case xem thông kê nghỉ phép	29
Bảng 3.8	Đặc tả use case bình luận dự án	31
Bảng 3.9	Đặc tả use case quản lý tài khoản user	34
Bảng 3.10	Đặc tả use case quản lý nhiệm vụ	38

DANH MỤC CÁC CHỮ VIẾT TẮT

Admin	Người quản lý ứng dụng
App	Ứng dụng
OT	Làm thêm giờ
User	Người sử dụng ứng dụng
Task	Nhiệm vụ được giao

CHƯƠNG 1. MỞ ĐẦU VÀ TỔNG QUAN ĐỀ TÀI

1.1 Lý do chọn đề tài

- **Nhu cầu thị trường:** Trong môi trường kinh doanh ngày nay, việc giám sát dự án và tối ưu thời gian làm việc rất cần thiết. Các doanh nghiệp luôn cần có các công cụ hiệu quả để giám sát dự án, tài nguyên của họ.
- **Nâng cao hiệu suất:** Việc sử dụng một app giám sát dự án giúp tăng cường hiệu suất làm việc, giảm thời gian lãng phí và tối ưu hóa tài nguyên của công ty.

1.2 Mục tiêu thực hiện đề tài

- **Tạo ra một công cụ hiệu quả cho quản lý dự án:** Mục đích chính của nghiên cứu có thể là phát triển một ứng dụng hoặc phần mềm, giúp giám sát dự án một cách tối ưu. Bao gồm việc quản lý tài nguyên, phân công công việc, theo dõi tiến độ, và tối ưu hóa quy trình làm việc.
- **Tăng hiệu quả làm việc** thông qua việc tối ưu hóa quản lý thời gian, giao tiếp và phân công công việc.
- **Nâng cao quản lý tài nguyên:** cải thiện quản lý tài nguyên, bao gồm nguồn nhân lực, thời gian để tối ưu hóa sự phối hợp và tối ưu các tài nguyên này một cách thông minh.

1.3 Phương pháp nghiên cứu

- **Xác định yêu cầu và tính năng:** Xác định yêu cầu, chức năng ứng dụng.
- **Thiết kế ứng dụng:** Tạo ra các bản thiết kế sơ bộ, sau đó chuyển thiết kế giao diện người dùng và trải nghiệm người dùng tốt nhất có thể
- **Phát triển và kiểm thử:** Tiến hành xây dựng các chức năng của ứng dụng sau đó thực hiện kiểm thử để tối ưu, bảo mật và khả năng hoạt động của ứng dụng.
- **Triển khai và đánh giá:** Đưa ra thị trường và thu nhập phản hồi của người dùng.

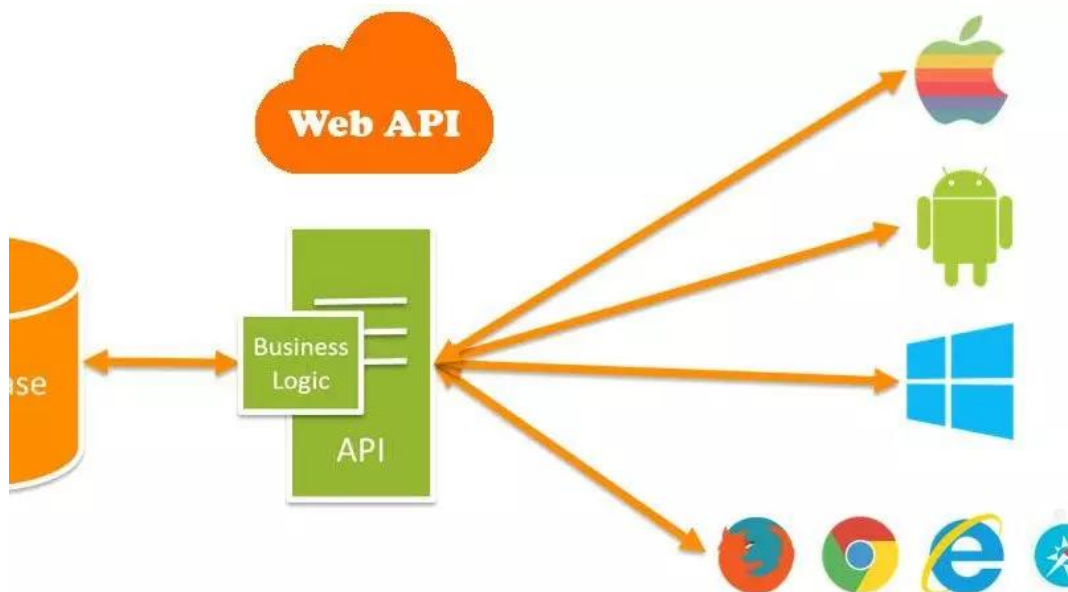
1.4 Mục tiêu kết quả

- **Xây dựng một app hoạt động ổn định, linh hoạt:** tạo ra app có khả năng hoạt động ổn định, linh hoạt và tuân thủ các tiêu chuẩn công nghệ.
- **Tối ưu hóa quản lý công việc và hiệu suất làm việc:** Mục tiêu là cải thiện quá trình quản lý và tiết kiệm thời gian làm việc thông qua việc cung cấp các tính năng giúp quản lý công việc hiệu quả hơn.
- **Tạo ra giá trị thực tế cho doanh nghiệp:** Mục tiêu là cung cấp giá trị thực tế cho doanh nghiệp, giúp họ quản lý dự án hiệu quả và tối ưu hóa thời gian làm việc.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1 Tổng quan về web api

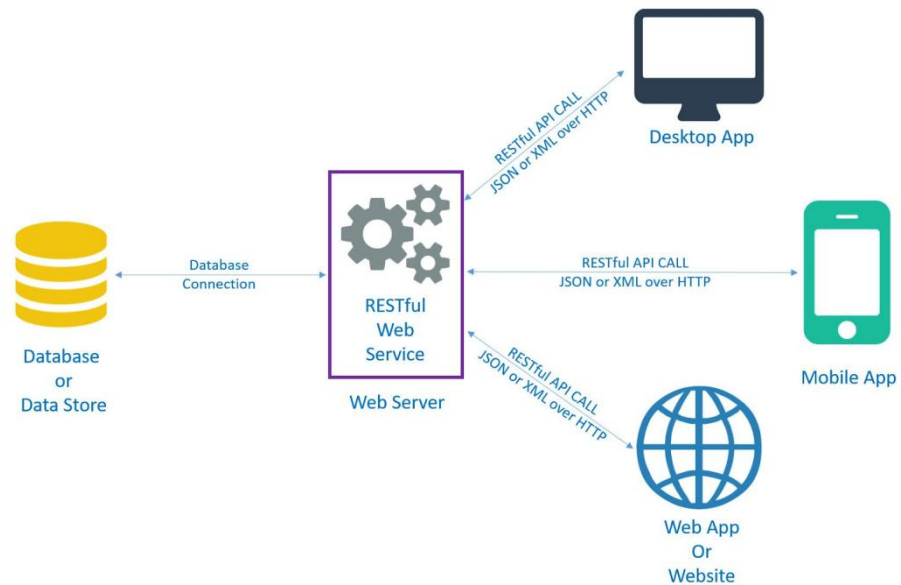
- **Khái niệm:** Web API là một công cụ cho phép ứng dụng trao đổi dữ liệu với nhau. Dữ liệu trả về thường ở dạng JSON hoặc XML và được truyền đi qua giao thức HTTP/HTTPS.



Hình 2.1 Web api

- **Cách thức hoạt động:**
 - Quá trình bắt đầu khi một URL API được tạo ra, cho phép các ứng dụng bên ngoài yêu cầu dữ liệu qua HTTP/HTTPS.
 - Tại máy chủ, các ứng dụng xác thực yêu cầu (nếu cần) và xác định dữ liệu cần phản hồi.
 - Dữ liệu sau đó được gửi về dưới dạng JSON hoặc XML.
 - Người nhận (có thể là một ứng dụng web hoặc di động) sẽ phân tích dữ liệu này.
- **Ưu điểm**
 - **Tích hợp Linh hoạt:** Web API mở cửa cho việc trao đổi dữ liệu giữa các ứng dụng và dự án khác nhau, thúc đẩy sự hợp tác và chia sẻ thông tin.

- **Quản lý Dữ liệu Theo Cấp Độ:** Cung cấp khả năng truy cập dữ liệu an toàn, cho phép phân loại thông tin theo quyền truy cập.
- **Phục vụ Đa Người dùng:** Cho phép mở rộng dịch vụ và chức năng đến người dùng toàn cầu.



● Nhược điểm

- **An ninh:** Các vấn đề như xác thực và quản lý quyền có thể làm tăng nguy cơ bảo mật, bao gồm nguy cơ bị tấn công
- **Hiệu Suất:** Phụ thuộc vào Web API, mà API có thể không được thiết kế để xử lý lượng truy cập cao.
- **Tài liệu và Hỗ trợ:** Thiếu tài liệu chi tiết hoặc hỗ trợ từ cộng đồng có thể khiến việc triển khai và bảo trì trở nên khó khăn

2.2 Giới thiệu React native

- **Khái niệm:** React Native là một mã nguồn JavaScript khung, được thiết kế để xây dựng ứng dụng trên nhiều nền tảng như iOS, Android và web ứng dụng, sử dụng cùng một cơ sở mã, và nó dựa trên React



Hình 2.2 React native

- **React và React Native:**

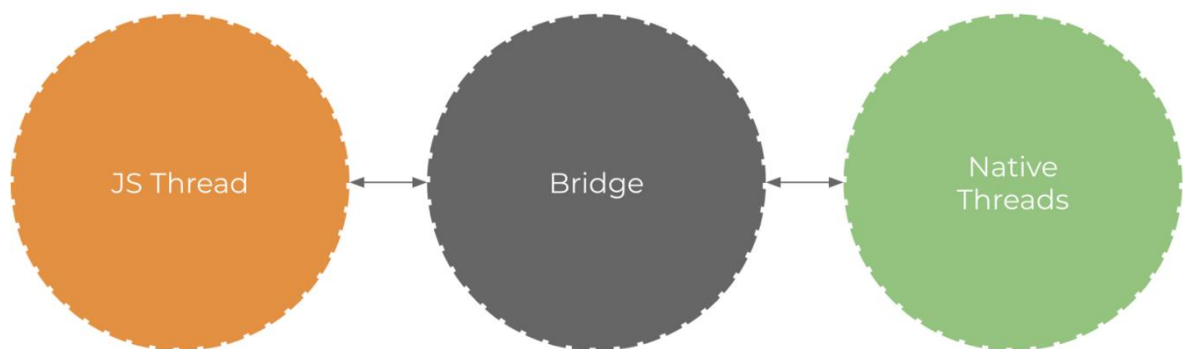
- React là một thư viện JavaScript, mã nguồn mở, được sử dụng bên phía giao diện người dùng, để thiết kế giao diện nhanh chóng, và đáng tin cậy cho các ứng dụng web.
- React Native, cung cấp bởi React và phát triển bởi Facebook, là một kết quả của React. Đó là một framework di động, cho phép xây dựng các ứng dụng gần như native bằng cách sử dụng JavaScript.
- Tóm lại, cả hai framework đều có mối liên hệ chặt chẽ với nhau, nhưng chúng đều được dùng cho một mục tiêu hoàn toàn khác. React dùng cho phát triển web, và React Native (không sử dụng HTML) được sử dụng cho phát triển di động.

- **Cách hoạt động:**

- Nó tích hợp trực tiếp các giao diện và các thành phần native. Nhờ đó, nó có thể sử dụng mã được viết bằng ngôn ngữ native, và cho phép API truy cập vào các tính năng dành riêng cho hệ điều hành trong ứng dụng
- Ban đầu, React Native sử dụng khái niệm "bridge", cho phép giao tiếp không đồng bộ giữa các phần tử JavaScript và Native - khái niệm bridge

nằm ở trung tâm của tính linh hoạt của React Native. Cách đây không lâu, thì React Native giới thiệu khái niệm Kiến trúc mới.

- Mục tiêu hiện tại là loại bỏ hoàn toàn bridge. Kiến trúc Mới chuyển từ "Bridge" sang một phương thức giao tiếp khác: Giao diện JavaScript (JSI). JSI cho phép khả năng lưu giữ các tham chiếu đến các đối tượng C++ bên trong các đối tượng JavaScript. Điều này cho phép giao tiếp đồng bộ giữa JavaScript và các phần tử native, dẫn đến hiệu suất được cải thiện.



Hình 2.2 Cầu nối của React Native

● Ưu điểm:

- **Dùng lại mã:** Việc phát triển hiệu quả nhiều nền tảng cùng lúc là ưu điểm lớn nhất và mạnh nhất của React Native. Việc sử dụng cùng một code base cho các nền tảng khác nhau mang lại những lợi ích như: phát triển ứng dụng nhanh hơn và rút ngắn thời gian đưa sản phẩm ra thị trường nhanh hơn, dễ dàng bảo trì và có giá thành rẻ hơn (chỉ cần quản lý một mã nguồn, thay vì phải tốn thời gian để xử lý nhiều mã nguồn),.... Mã code tương tự có thể được dùng với website và android/ios nếu cả hai đều sử dụng React Native.
- **Xem các thay đổi trong code:** React native cho phép các nhà phát triển xem các thay đổi trong mã của họ trong thời gian thực, mà không cần phải làm mới bất cứ điều gì. Điều chỉnh nhỏ này có vẻ như không đáng kể, nhưng có thể cải thiện rõ rệt quá trình phát triển, bởi vì nó cung cấp phản hồi theo thời gian thực, và về bất kỳ điều gì được thay đổi bên trong mã,

dẫn đến tăng năng suất thực hiện. Ngoài ra, có thể chỉ tải lại các phần cụ thể của mã, tiết kiệm thời gian cần thiết cho việc biên dịch đầy đủ.

- **Về hiệu suất:** So sánh với các giải pháp phát triển đa nền tảng khác, khái niệm "bridge" của React Native có thể được coi như là một cuộc cách mạng. Vì ứng dụng React Native cho phép sử dụng mã được viết bằng ngôn ngữ native, nên nó không bị lag như các giải pháp đa nền tảng dựa trên web.
- **Chi phí rẻ:** Hiệu quả về chi phí chính là cốt lõi, và là lý do chính cho việc phát triển đa nền tảng. Nhờ việc sử dụng lại đoạn mã trên nhiều nền tảng, mà ta chỉ cần một đội nhóm nhỏ hơn để triển khai dự án. Ngược lại với việc phát triển native, nơi bạn cần hai đội riêng biệt để cung cấp và triển khai hai sản phẩm hoạt động tương tự nhau, thay vì chỉ một nhóm như React native.
- **Về cộng đồng:** React Native là một framework mã nguồn mở và hiện tại cộng đồng của nó đang trở nên mạnh mẽ và không ngừng mở rộng.
- **Nhược điểm:**
 - React native vẫn còn là một ngôn ngữ tương đối mới, nên vẫn có khả năng có một số hạn chế, lỗi,....
 - Hiệu suất của React Native là vượt trội so với các giải pháp giúp xây dựng trên nhiều nền tảng khác. Tuy nhiên, nó vẫn chậm hơn so với việc phát triển ứng dụng gốc.

CHƯƠNG 3. PHÂN TÍCH HỆ THỐNG

3.1 Đặc tả yêu cầu

Đề tài xây dựng ứng dụng quản lý dự án đặt ra một số yêu cầu cụ thể để được tổ chức, theo dõi và giám sát các dự án một cách tối ưu.

Các yêu cầu chức năng được phân chia theo nhóm dựa trên tính chất của các chương trình để thuận tiện cho việc giám sát. Các nhóm chức năng bao gồm:

- Nhóm chức năng quản lý thông tin người dùng
- Nhóm chức năng quản lý thời gian, công việc của người dùng.
- Nhóm chức năng quản lý các cuộc họp, thảo luận giữa các người dùng.

3.1.1 Nhóm chức năng quản lý thông tin người dùng

- **Mô tả chung:** Nhóm chức năng này giúp quản lý thông tin của nhân viên. Vì đối tượng người sử dụng là các nhân viên của doanh nghiệp nên hệ thống sẽ đòi hỏi các thông tin cá nhân cho mỗi đối tượng người dùng. Chi tiết về các thông tin cần quản lý như sau:

- Ngày sinh
- Giới tính
- Chứng minh nhân dân
- Ngày cấp
- Nơi cấp
- Quốc tịch
- Dân tộc
- Tôn giáo
- Hôn nhân
- Trình độ học vấn
- Số điện thoại
- Email cá nhân
- Email doanh nghiệp
- Địa chỉ liên hệ

- Đối với nhân viên đã ký hợp đồng với công ty sẽ cần có quản lý thêm các thông tin trong hợp đồng

3.1.2 Nhóm chức năng quản lý thời gian, công việc của người dùng

- **Mô tả chung:** Nhóm chức năng này giúp cho người sử dụng quản lý thời gian làm việc, công việc được giao một cách hiệu quả. Bên cạnh đó ở nhóm chức năng này người dùng cũng có thể đăng ký lịch nghỉ phép, đăng ký làm thêm giờ và có thể xem được tiền lương của mình.

3.1.3 Nhóm chức năng quản lý các cuộc họp, thảo luận

- **Mô tả chung:** Trao đổi nội bộ giữa các thành viên trong team, giữa các team khác nhau trong một doanh nghiệp là một yếu tố không quan trọng không thể thiếu. Ở nhóm chức năng này cung cấp cho người dùng quản lý được các cuộc họp và có nơi trao đổi giữa các thành viên trong công ty với nhau.

3.2 Yêu cầu chức năng

- **User có thể:**
 - Đăng nhập
 - Tìm kiếm dự án
 - Chấm công
 - Đăng kí OT
 - Đăng ký nghỉ phép/công tác
 - Xem thống kê nghỉ phép
 - Bình luận về dự án
- **Admin có thể:**
 - Đăng nhập
 - Quản lý người dùng của dự án
 - Quản lý tài khoản user (*Xem/Thêm/Sửa*)
 - Quản lý nhiệm vụ (*Xem/Thêm/Sửa*)
 - Xem thống kê nghỉ phép

3.3 Yêu cầu phi chức năng

- **Hiệu suất:**

- **Đáp ứng thời gian:** Hệ thống phải có thể xử lý các yêu cầu người dùng và tải lớn mà vẫn giữ được thời gian phản hồi hợp lý.
- **Tải đồng thời (concurrent load):** Hệ thống cần hỗ trợ số lượng người dùng đồng thời mà không làm giảm hiệu suất.
- **Thời gian đáp ứng:** Đảm bảo thời gian đáp ứng của hệ thống nhanh chóng và ổn định

- **Bảo mật:**

- **Quản lý quyền truy cập:** Hệ thống cần hỗ trợ quản lý quyền truy cập, để chắc chắn rằng: chỉ những người được cấp quyền, mới được truy cập vào các chức năng quan trọng.
- **Bảo vệ dữ liệu:** Bảo vệ thông tin cá nhân và dữ liệu quan trọng tránh khỏi truy cập trái phép hoặc thay đổi.

- **Duy trì và hỗ trợ:** dễ duy trì, giao diện thân thiện

- **Tương thích với các hệ thống khác:** Hỗ trợ tích hợp với các hệ thống khác thông qua API hoặc giao thức tiêu chuẩn.

- **Tương thích trình duyệt:** Hỗ trợ nhiều trình duyệt phổ biến để đảm bảo mọi người dùng có thể truy cập.
- **Tương thích thiết bị:** Hỗ trợ nhiều thiết bị khác nhau như máy tính, điện thoại di động và máy tính bảng.

- **Hiệu suất Điều hành:**

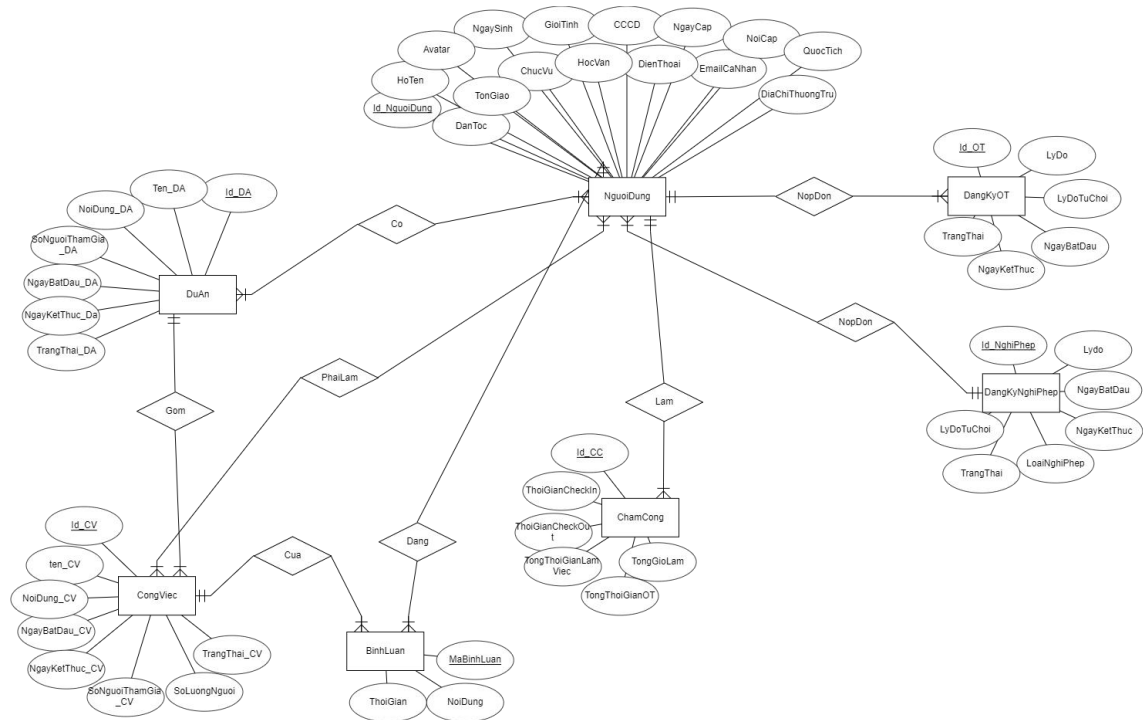
- **Dung lượng hệ thống:** Đảm bảo rằng hệ thống không chiếm quá nhiều tài nguyên hệ thống.
- **Tiêu thụ băng thông:** Kiểm soát lượng băng thông mà hệ thống tiêu thụ để tránh ảnh hưởng đến hiệu suất của mạng...

3.4 Danh mục các use-case của hệ thống:

Bảng 3.1 Danh mục các use case của hệ thống

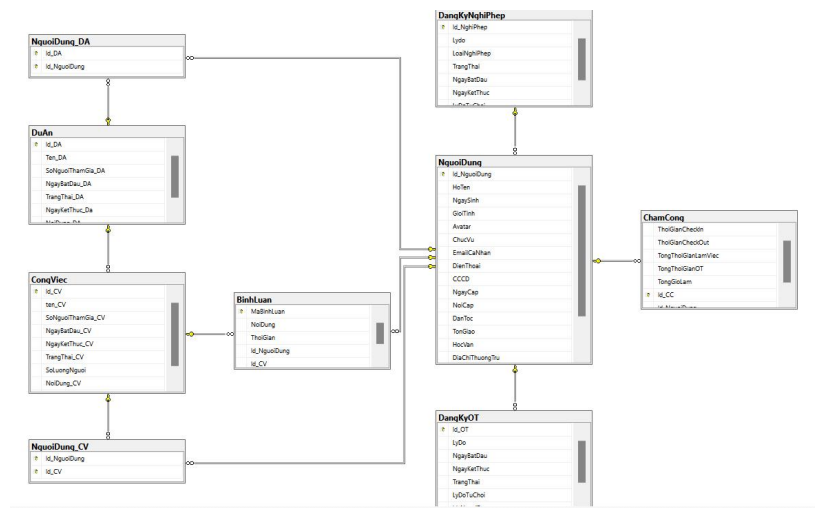
ID	Tên Use-case	Mô tả	Actor
UC01	Đăng nhập	Cho phép người dùng có thể đăng nhập vào hệ thống.	User, Admin
UC02	Tìm kiếm dự án	Người dùng có thể tìm kiếm tên dự án mình đang làm	User, Admin
UC03	Chấm công	Người dùng có thể chấm công	User
UC04	Đăng ký OT	Người dùng đăng ký làm ngoài giờ	User
UC05	Đăng ký nghỉ phép	Người dùng đăng ký nghỉ phép	User
UC06	Xem thống kê nghỉ phép	Người dùng xem số ngày nghỉ phép trong tháng	User
UC07	Bình luận về dự án	Thảo luận và chia sẻ nhiệm vụ (<i>Trao đổi nội bộ giữa thành viên trong team, trong công ty</i>)	User, Admin
UC08	Quản lý tài khoản user của dự án	Thêm/sửa tài khoản user của dự án	Admin
UC09	Quản lý nhiệm vụ	Tạo nhiệm vụ, giao nhiệm vụ	Admin

3.5 Mô hình ERD (*Entity Relationship Diagram*)



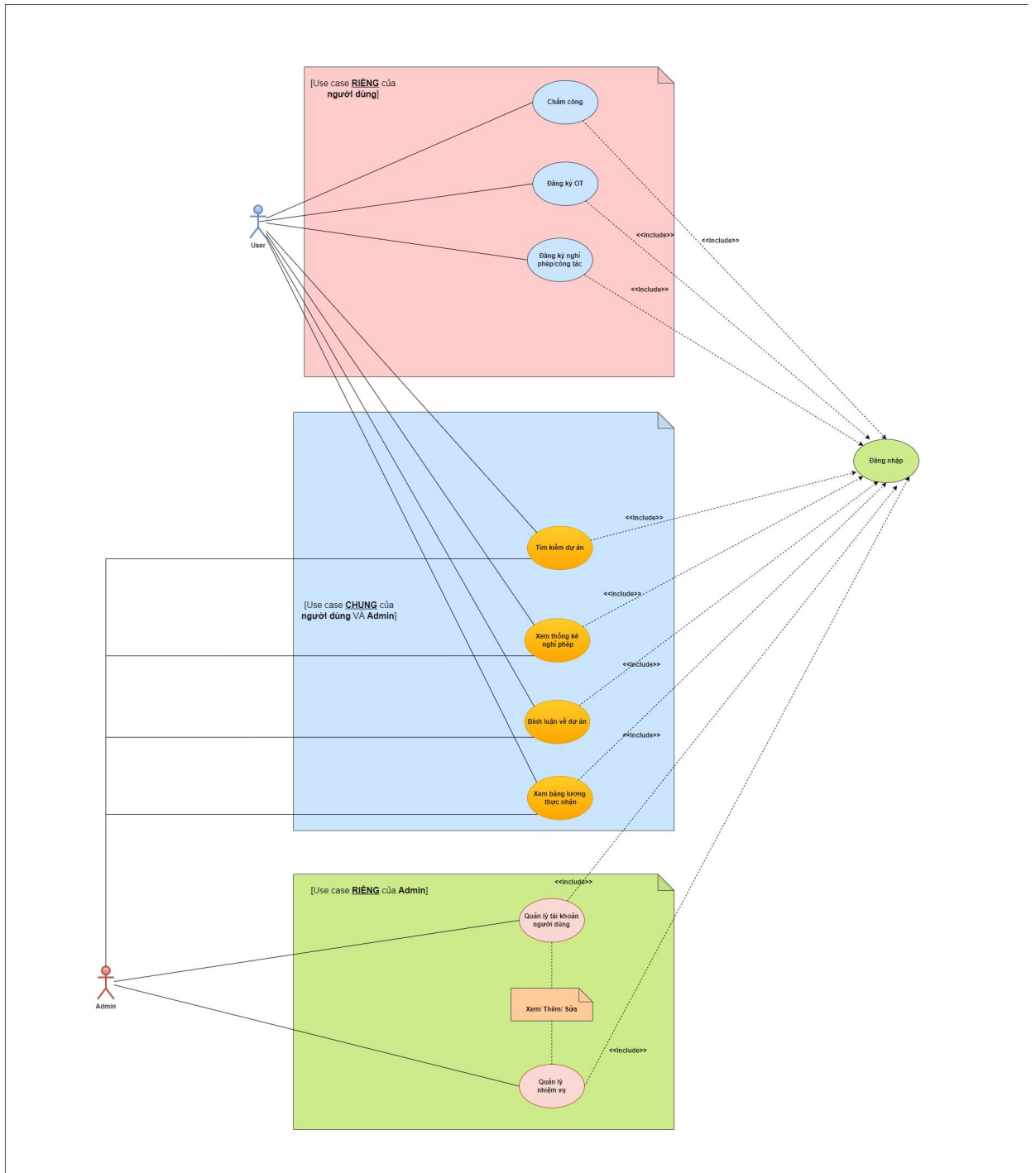
Hình 3.1 Mô hình quan hệ

3.6 Sơ đồ lớp (Class diagram)



Hình 3.2 Sơ đồ lớp

3.7 Use case tổng quát

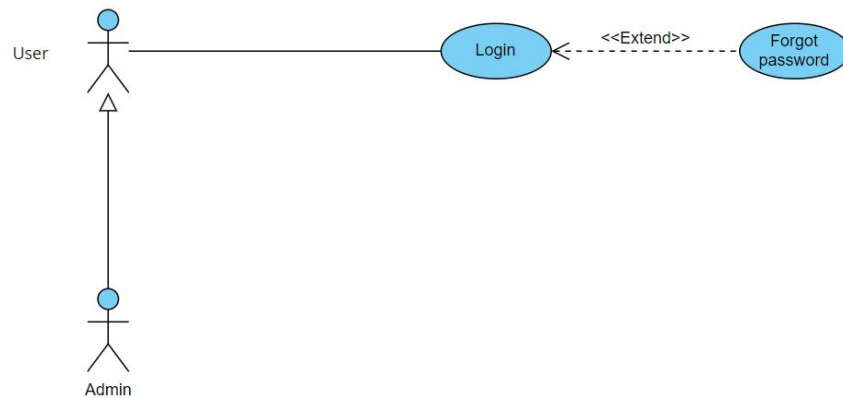


Hình 3.3 Use case tổng quát

3.8 Chi tiết Use case

3.8.1 Use case đăng nhập

3.8.1.1 Sơ đồ use case



Hình 3.4 Use case đăng nhập

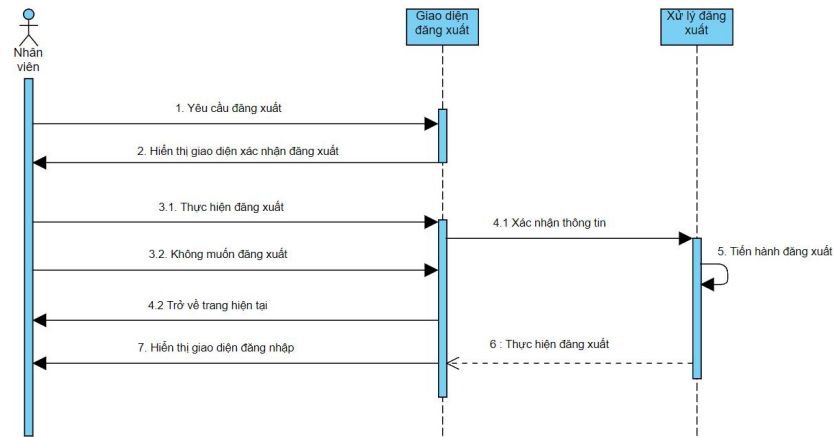
3.8.1.2 Đặc tả use case

Bảng 3.2 Đặc tả use case đăng nhập

Use Case ID	UC01
Use case	Đăng nhập
Description	User, admin đăng nhập vào hệ thống
Actor	User, admin
Trigger	Người dùng đăng nhập vào hệ thống mà chưa đăng nhập trước đó.

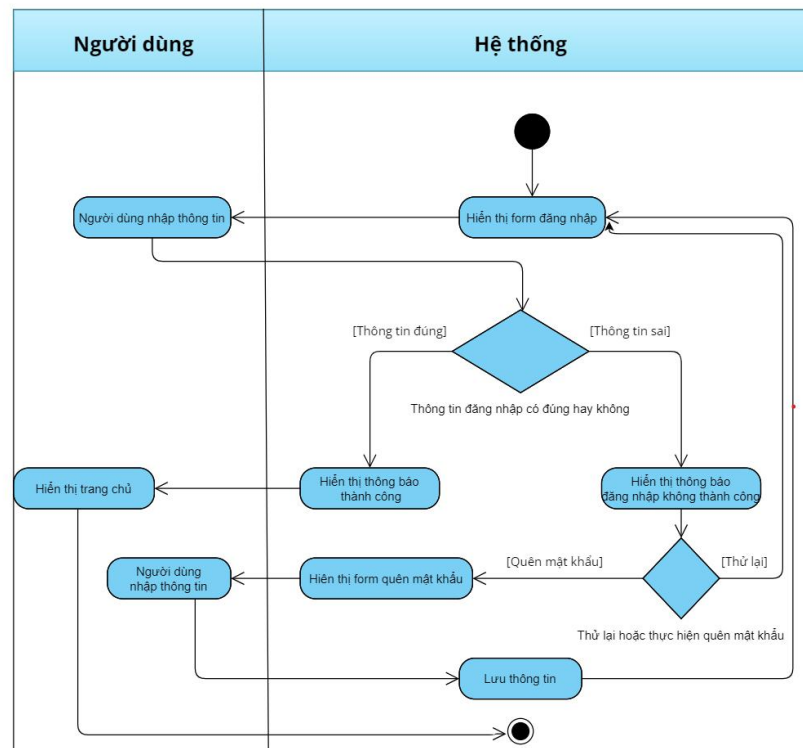
Pre - condition	- Người dùng phải có tài khoản trước đó. - Thiết bị người dùng phải truy cập được Internet	
Post - condition	Người dùng đăng nhập vào được hệ thống	
Basic flow	ACTOR	SYSTEM
	1. User, admin click vào icon của app trên màn hình 2. User, admin nhập thông tin đăng nhập bao gồm email và mật khẩu	1.1. Hệ thống hiển thị giao diện đăng nhập 2.1. Hệ thống ghi nhận người dùng đăng nhập thành công. .
Exception flow	2.1.1 Hệ thống ghi nhận thông tin đăng nhập không đúng và hiển thị thông báo. 2.1.1.1 Người dùng không tiếp tục đăng nhập <i>Use case dừng lại</i> 2.1.1.2. Người dùng chọn quên mật khẩu	

3.8.1.3 Sơ đồ tuần tự



Hình 3.5 Sơ đồ tuần tự use case đăng nhập

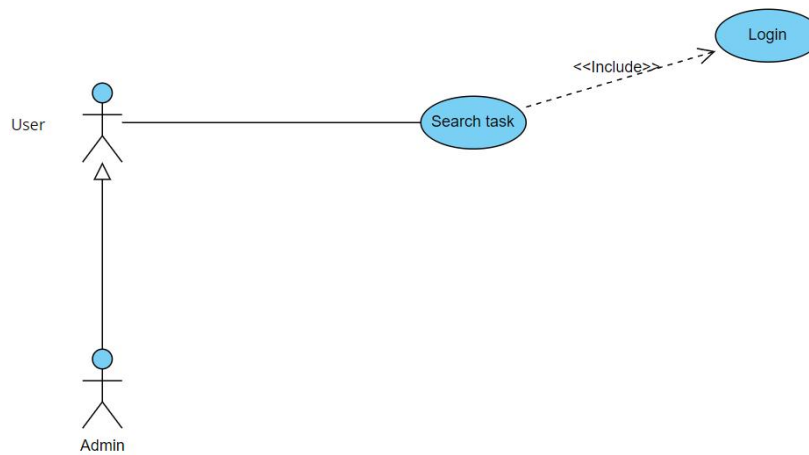
3.8.1.4 Sơ đồ hoạt động



Hình 3.6 Sơ đồ hoạt động use case đăng nhập

3.8.2 Use case tìm kiếm dự án

3.8.2.1 Sơ đồ use case



Hình 3.7 Use case tìm kiếm dự án

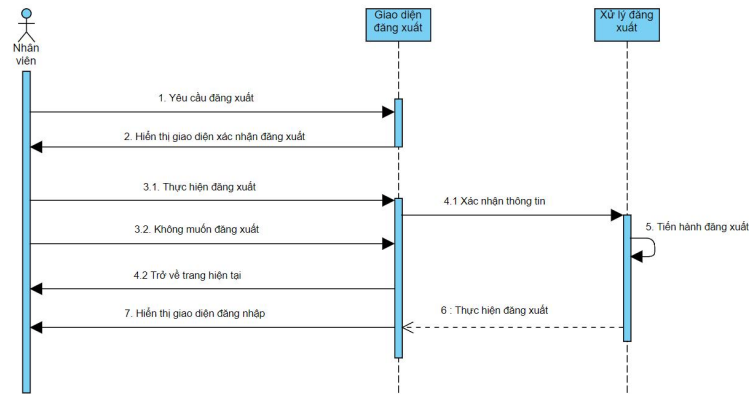
3.8.2.2 Đặc tả use case

Bảng 3.3 Đặc tả use case tìm kiếm dự án

Use Case ID	UC02
Use case	Tìm kiếm dự án
Description	Người dùng tìm kiếm dự án
Actor	User, admin
Trigger	Người dùng nhập thông tin cần tìm kiếm vào thanh tìm

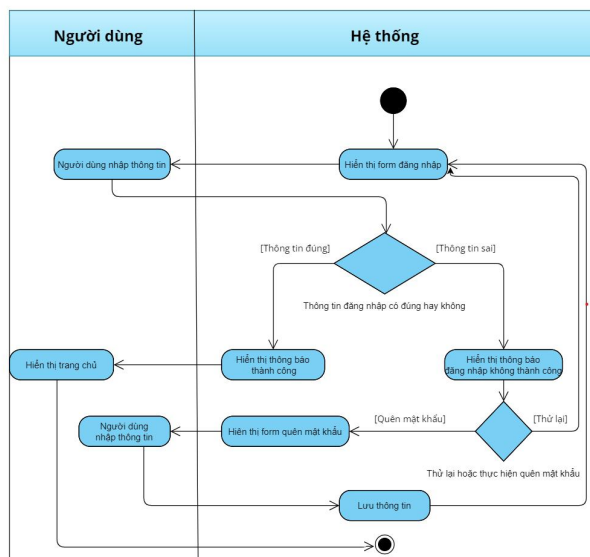
	kiểm	
Post - condition	Người dùng nhận được kết quả tìm kiếm và có thể xem chi tiết nhiệm vụ.	
Basic flow	ACTOR	SYSTEM
	<ol style="list-style-type: none"> 1. Người dùng truy cập vào tính năng “Tìm kiếm nhiệm vụ” trong giao diện ứng dụng 2. Người dùng nhập từ khóa tìm kiếm và nhấn vào Enter. 3. Người dùng nhấp vào nhiệm vụ để xem chi tiết 	<ol style="list-style-type: none"> 1.1. Hệ thống hiển thị ô tìm kiếm và yêu cầu người dùng nhập từ khóa tìm kiếm 2.1. Hệ thống tiến hành tìm kiếm nhiệm vụ dựa trên từ khóa nhập vào 2.2. Hệ thống hiển thị danh sách các nhiệm vụ phù hợp với từ khóa. 3.1. Hệ thống hiển thị các thông tin liên quan đến nhiệm vụ
Exception flow	2.2.1 Nếu không có nhiệm vụ nào phù hợp với từ khóa tìm kiếm, hệ thống thông báo cho người dùng không tìm thấy kết quả	

3.8.2.3 Sơ đồ tuần tự



Hình 3.8 Sơ đồ tuần tự use case tìm kiếm dự án

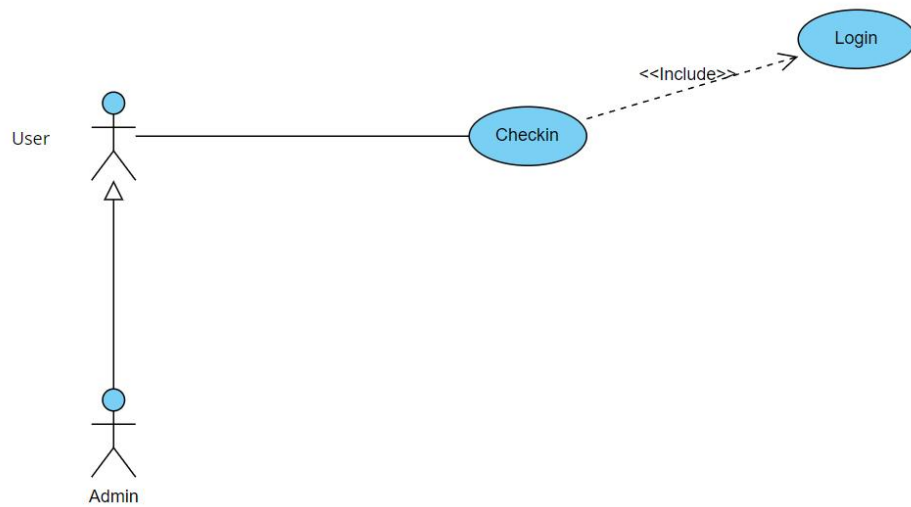
3.8.2.4 Sơ đồ hoạt động



Hình 3.9 Sơ đồ hoạt động Use case tìm kiếm dự án

3.8.3 Use case chấm công

3.8.3.1 Sơ đồ use case



Hình 3.10 Use case chấm công

3.8.3.2 Đặc tả use case

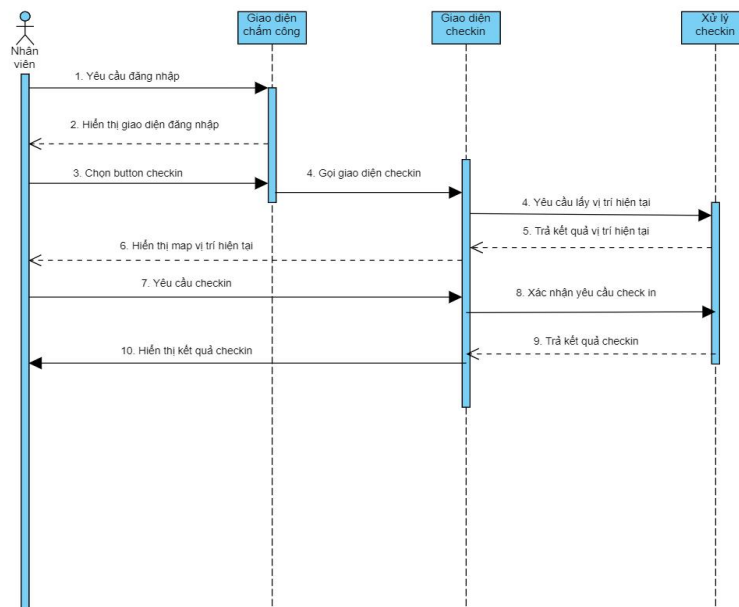
Bảng 3.4 Đặc tả use case chấm công

Use Case ID	UC03	
Use case	Chấm công	
Description	Người dùng chấm công khi đến giờ và hết giờ làm việc	
Actor	User	
Trigger	Người dùng thực hiện check in, check out	

Basic flow	ACTOR	SYSTEM
	<p>1. Người dùng truy cập vào tính năng “Chấm công” trong giao diện ứng dụng</p> <p>2. Người dùng chọn “Check in” để bắt đầu ngày làm việc</p> <p>3. User chọn “Check in” để tiến hành điểm danh</p> <p>4. Người dùng chọn “Check out” để kết thúc giờ làm việc</p> <p>5. User chọn “Check out” để tiến hành điểm danh</p>	<p>1.1. Hệ thống hiển thị giao diện chấm công với hai tùy chọn: ”Check in” và “Check out”</p> <p>3.1. Hệ thống ghi nhận lại thời gian chấm công của User và hiển thị thời gian checkin của User.</p> <p>4.1. Hệ thống hiển thị thời gian check out của nhân viên đó.</p> <p>5.1. Hệ thống ghi nhận lại thời gian, địa điểm chấm công của User và hiển thị thời gian check out của</p>

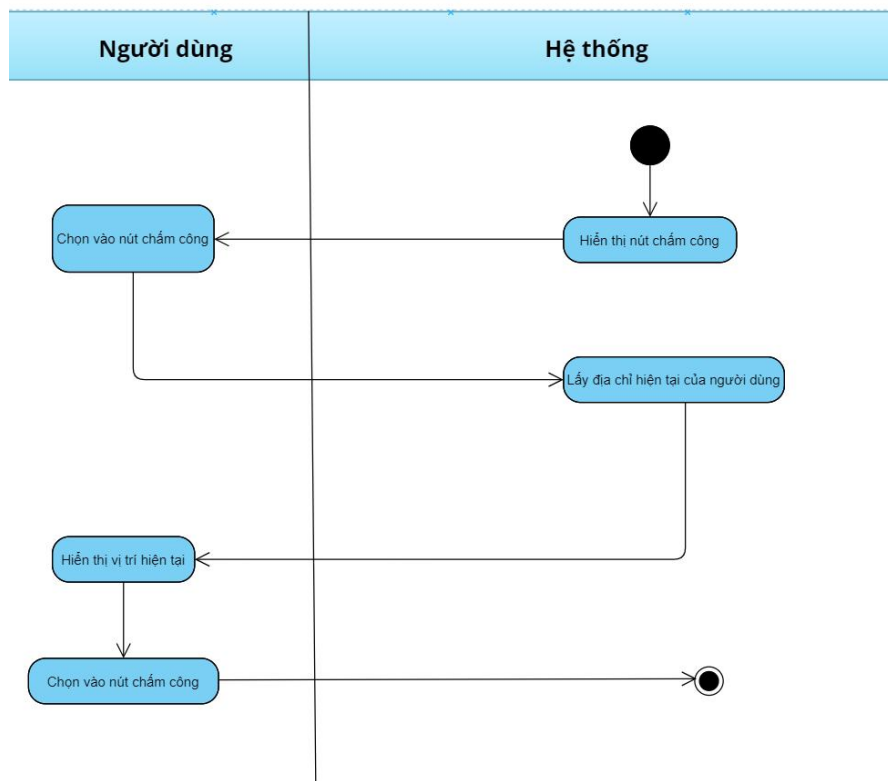
		User.
Exception flow	<p>Ở bước 3 và bước 5, nếu ghi nhận thời gian chấm công của User bị lỗi hoặc không thành công(Ví dụ: do lỗi kết nối), hệ thống sẽ hiển thị thông báo yêu cầu người dùng thử lại)</p>	

3.8.3.3 Sơ đồ tuần tự



Hình 3.11 Sơ đồ tuần tự Use case chấm công

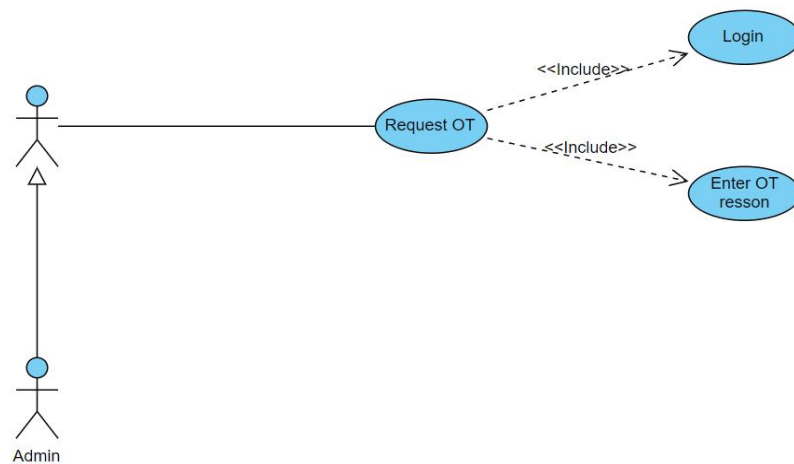
3.8.3.4 Sơ đồ hoạt động



Hình 3.12 Sơ đồ hoạt động Use case chấm công

3.8.4 Use case đăng ký OT (làm thêm giờ)

3.8.4.1 Sơ đồ use case



Hình 3.13 Use case đăng ký OT

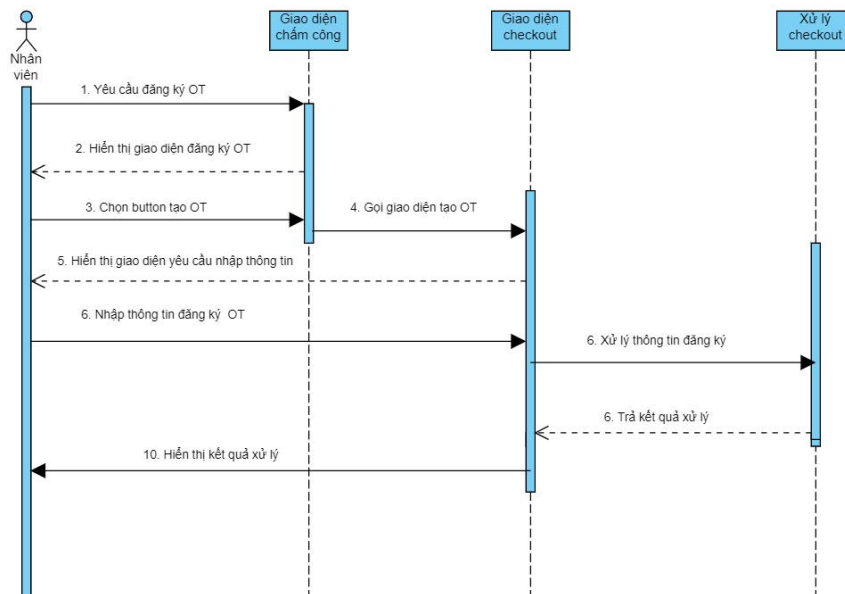
3.8.4.2 Đặc tả use case

Bảng 3.5 Đặc tả use case đăng ký làm thêm giờ

Use Case ID	UC04	
Use case	Đăng ký làm việc ngoài giờ	
Description	Người dùng yêu cầu làm việc ngoài giờ	
Actor	User	
Trigger	Người dùng thực hiện đăng ký làm việc ngoài giờ	
Post - condition	Yêu cầu đăng ký làm việc ngoài giờ của User đã được ghi nhận trong hệ thống và cần được duyệt bởi Admin	
Basic flow	ACTOR	SYSTEM
	<ol style="list-style-type: none"> 1. Người dùng truy cập tính năng "Đăng ký OT" trong giao diện của ứng dụng 2. User nhập thông tin đăng ký làm việc ngoài giờ và sau đó 	<ol style="list-style-type: none"> 2.1. Hệ thống ghi nhận yêu cầu đăng ký làm việc ngoài giờ của User

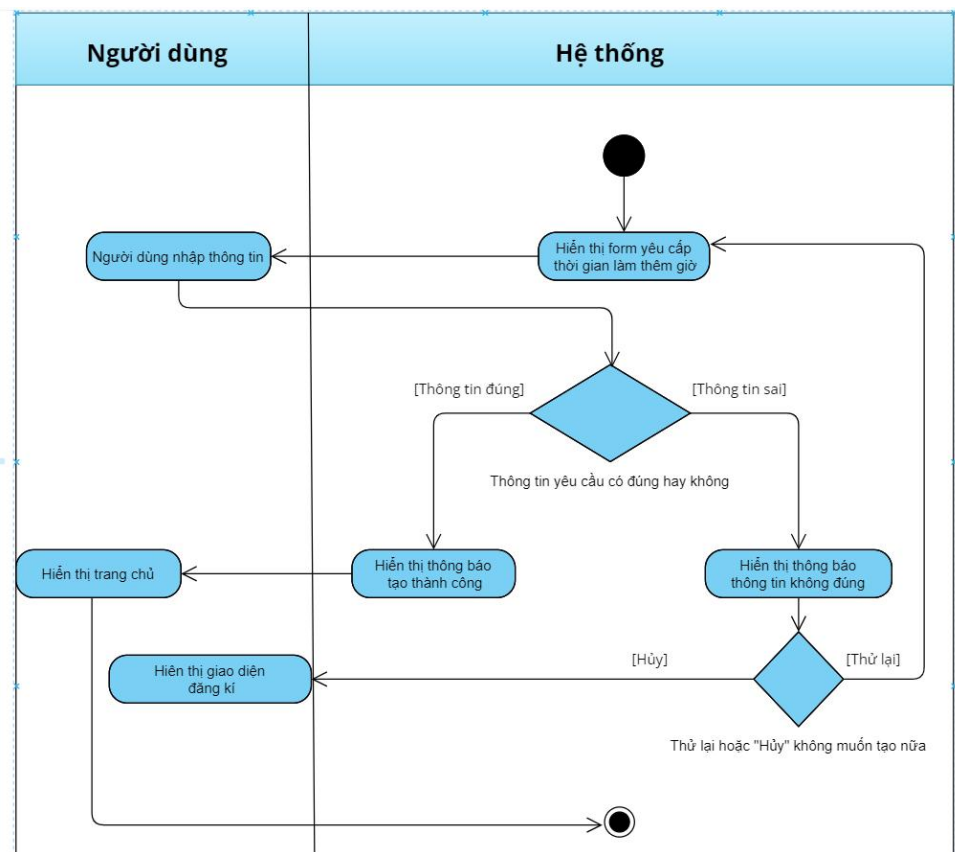
	chọn button “Đăng ký”	2.2. Hệ thống hiển thị thông báo xác nhận rằng yêu cầu đã tạo thành công và đang chờ phê duyệt từ Admin
Exception flow	<p>2a.</p> <p>Nếu thông tin đăng ký ở bước 2 không hợp lệ, hoặc thiếu thông tin bắt buộc hệ thống sẽ yêu cầu User nhập đủ thông tin cần thiết</p>	

3.8.4.3 Sơ đồ tuần tự



Hình 3.14 Sơ đồ tuần tự Use case đăng ký OT

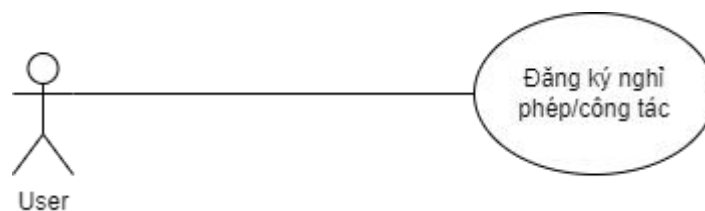
3.8.4.4 Sơ đồ hoạt động



Hình 3.15 Sơ đồ hoạt động Use case đăng ký OT

3.8.5 Use case đăng ký nghỉ phép/công tác

3.8.5.1 Sơ đồ use case



Hình 3.16 Use case đăng ký nghỉ phép/công tác

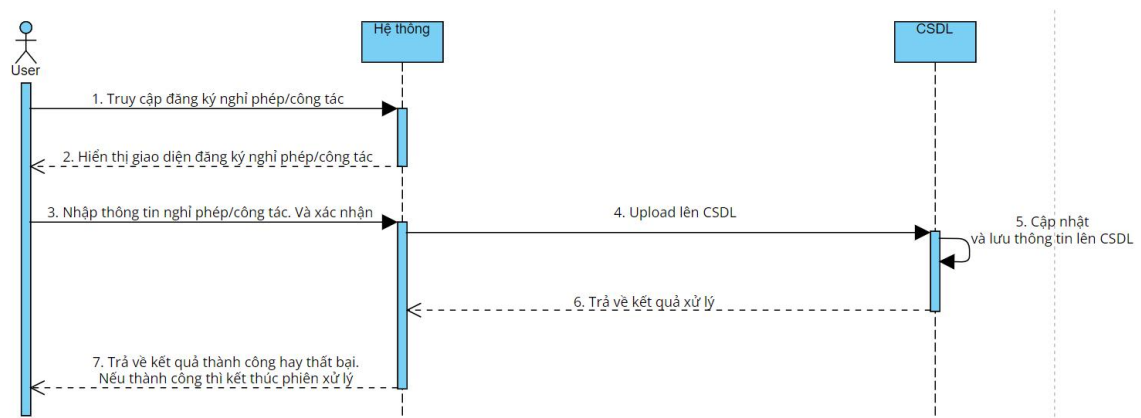
3.8.5.2 Đặc tả use case

Bảng 3.6 Đặc tả use case đăng ký nghỉ phép

Use Case ID	UC05	
Use case	Đăng ký nghỉ phép/công tác	
Description	Use case đăng ký nghỉ phép/công tác	
Actor	User	
Trigger	Người dùng thực hiện đăng ký nghỉ phép/công tác	
Pre - condition	Người dùng phải đăng nhập vào được hệ thống Thiết bị của người dùng phải truy cập vào Internet.	
Post - condition	Yêu cầu đăng ký nghỉ phép/công tác của người dùng đã được ghi nhận trong hệ thống và cần được duyệt bởi Admin	
Basic flow	ACTOR	SYSTEM
	1. Người dùng truy cập tính năng "Đăng ký nghỉ phép" trong giao diện của ứng dụng 2. Người dùng chọn	1.1. Hệ thống hiển thị giao diện đăng ký nghỉ phép. 2.1. Hệ thống ghi nhận yêu

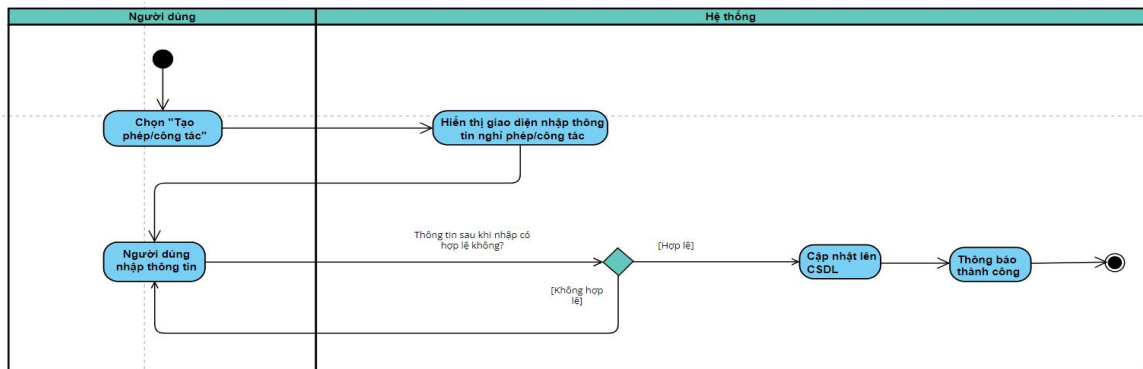
	“Tạo phép” và nhập thông tin đăng ký phép/công tác và sau đó chọn button “Tạo phép/công tác”	cầu đăng ký phép/công tác và hiển thị thông báo xác nhận tạo thành công và đang chờ phê duyệt từ admin
Exception flow	2a. Nếu thông tin đăng ký ở bước 2 không hợp lệ, hoặc thiếu thông tin bắt buộc hệ thống sẽ yêu cầu User nhập đủ thông tin cần thiết	

3.8.5.3 Sơ đồ tuần tự



Hình 3.17 Sơ đồ tuần tự Use case đăng ký nghỉ phép/công tác

3.8.5.4 Sơ đồ hoạt động



Hình 3.18 Sơ đồ hoạt động Use case đăng ký nghỉ phép/công tác

3.8.6 Use case xem thống kê nghỉ phép

3.8.6.1 Sơ đồ use case



Hình 3.19 Use case xem thống kê nghỉ phép

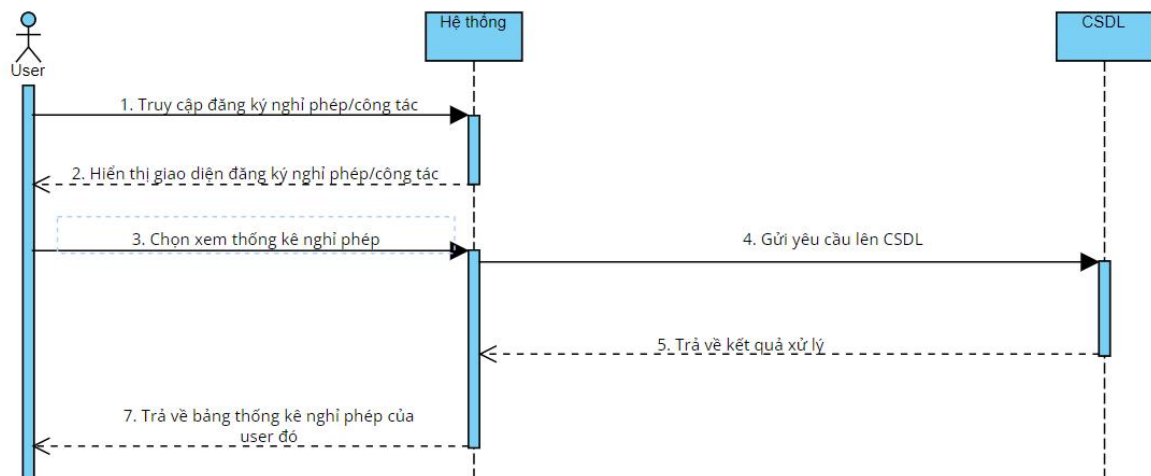
3.8.6.2 Đặc tả use case

Bảng 3.7 Đặc tả use case xem thống kê nghỉ phép

Use Case ID	UC06
Use case	Xem thống kê nghỉ phép
Description	Usecase xem thống kê nghỉ phép
Actor	User
Trigger	Người dùng thực hiện xem thống kê nghỉ phép

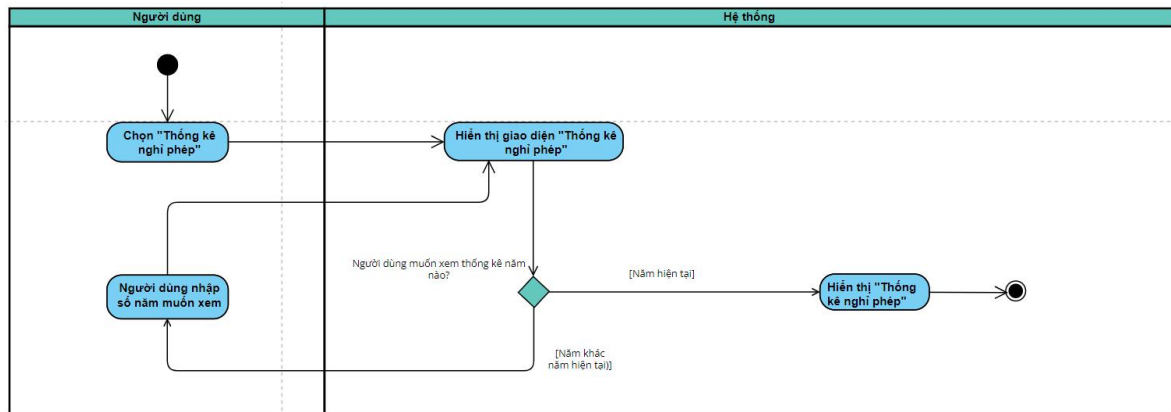
Post - condition	Hệ thống hiển thị giao diện “Thống kê nghỉ phép”	
Basic flow	ACTOR	SYSTEM
	1. Người dùng truy cập tính năng "Xem thống kê nghỉ phép" trong giao diện của ứng dụng	1.1. Hệ thống hiển thị giao diện “Thống kê nghỉ phép”.

3.8.6.3 Sơ đồ tuần tự



Hình 3.20 Sơ đồ tuần tự Use case xem thống kê nghỉ phép

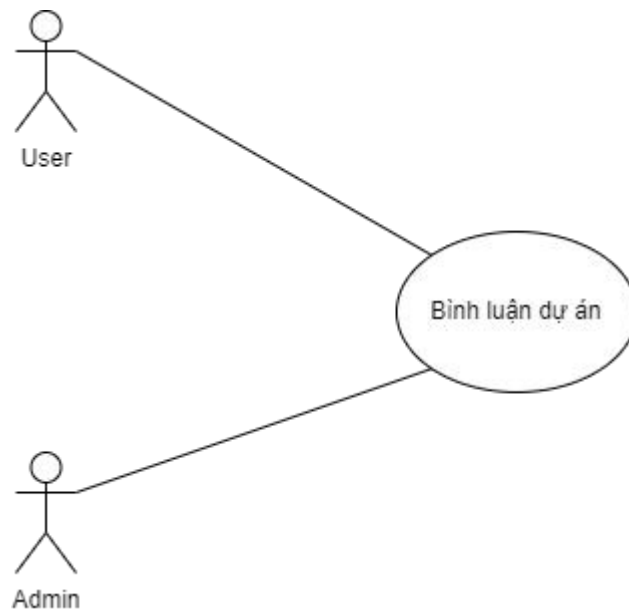
3.8.6.4 Sơ đồ hoạt động



Hình 3.21 Sơ đồ hoạt động Use case xem thống kê nghỉ phép

3.8.7 Use case bình luận về dự án

3.8.7.1 Sơ đồ use case



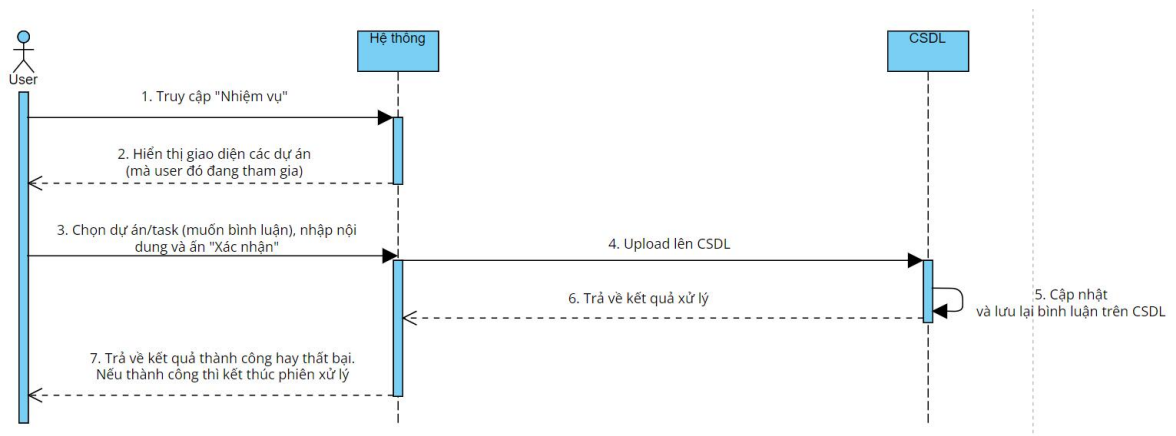
Hình 3.22 Use case bình luận về dự án

3.8.7.2 Đặc tả use case

Bảng 3.8 Đặc tả use case bình luận dự án

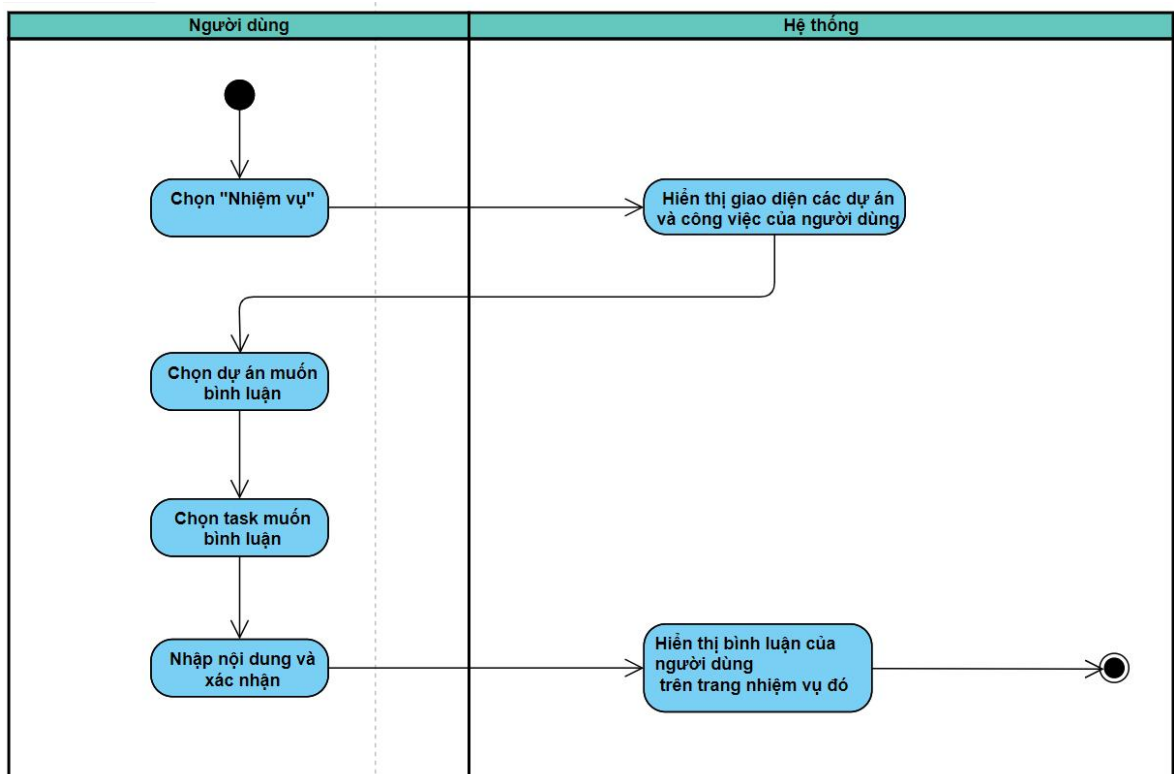
Use Case ID	UC07
Use case	Bình luận về dự án
Description	Người dùng chia sẻ ý kiến của mình về dự án
Actor	User, Admin
Trigger	Khách hàng bình luận dự án
Pre - condition	Người dùng đăng nhập ứng dụng thành công
Post - condition	Hệ thống sẽ cập nhật lên giao diện, bình luận của khách hàng về nhiệm vụ
Basic flow	<ol style="list-style-type: none"> 1. Người dùng chọn dự án muốn bình luận 2. Người dùng chọn ô bình luận, nhập nội dung và ấn “Bình luận”

3.8.7.3 Sơ đồ tuần tự



Hình 3.23 Sơ đồ tuần tự Use case bình luận về dự án

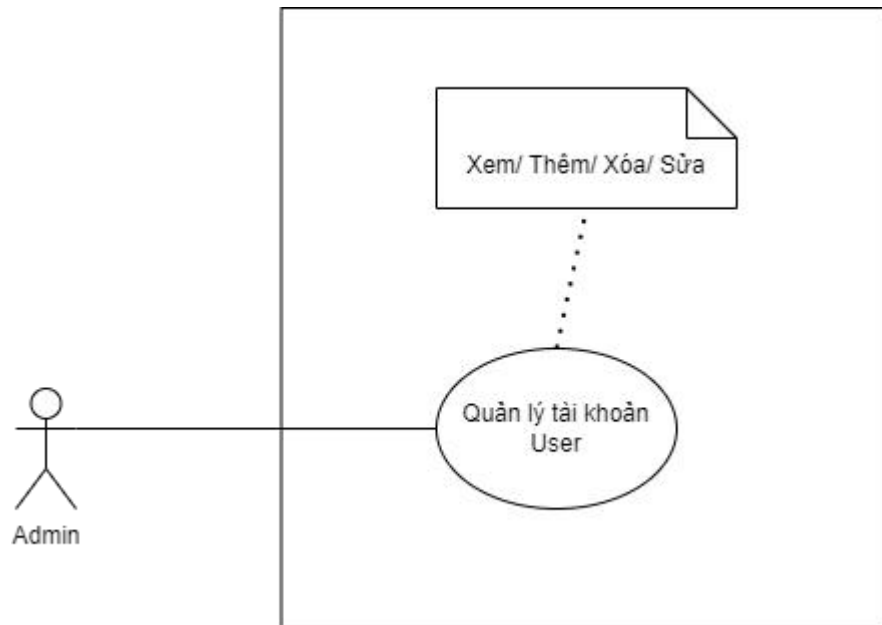
3.8.7.4 Sơ đồ hoạt động



Hình 3.24 Sơ đồ hoạt động Use case bình luận về dự án

3.8.8 Use case quản lý tài khoản user

3.8.8.1 Sơ đồ use case



Hình 3.25 Use case quản lý tài khoản user

3.8.8.2 Đặc tả use case

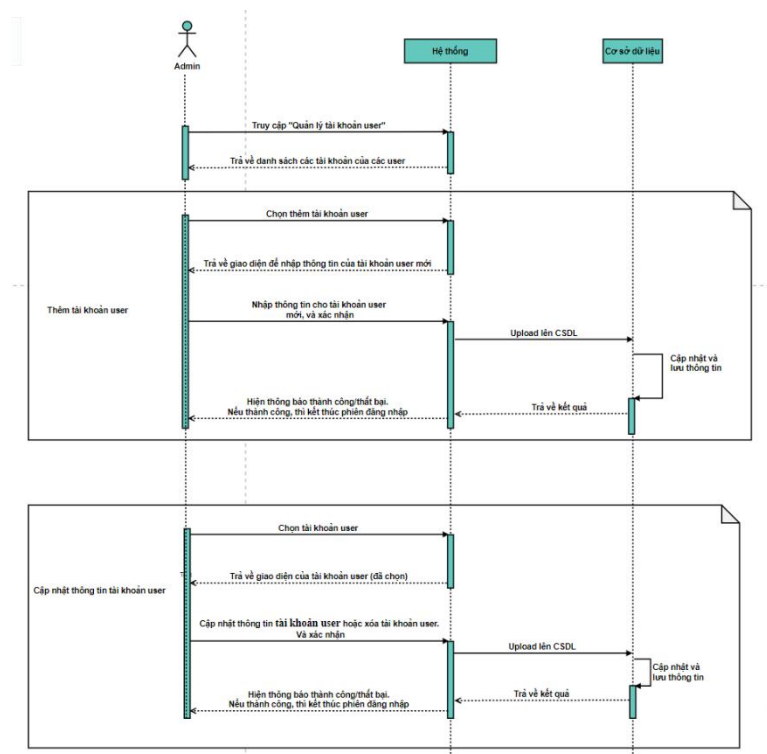
Bảng 3.9 Đặc tả use case quản lý tài khoản user

Use Case ID	UC08
Use case	Quản lý tài khoản user
Description	Người dùng có thể thêm và sửa thông tin tài khoản user
Actor	Admin
Trigger	Admin thêm/sửa tài khoản user

Post - condition	<p>Người dùng chỉnh sửa tài khoản user thành công</p> <p>Hệ thống sẽ hiện lên thông báo chỉnh sửa tài khoản user thành công</p>	
Basic flow	ACTOR	SYSTEM
	<ol style="list-style-type: none"> 1. chọn “quản lý tài khoản user” 2. Sau đó, người dùng chọn mục thêm tài khoản user 3. Nhập thông tin user cần thêm 4. Người dùng ấn xác nhận thêm 	<p>3.1. Hệ thống hỏi người dùng có chắc chắn muốn thêm tài khoản user vào hệ thống hay không?</p> <p>4.1. Hệ thống sẽ thêm tài khoản user vào database và hiển thị thông báo đã thêm tài khoản user thành công trên màn hình</p>
Alternative flow	<p>2a. Người dùng chọn chỉnh sửa tài khoản user</p> <p>3a. Người dùng nhập thông tin tài khoản user cần sửa</p> <p>4a. Người dùng ấn xác</p>	<p>3a1. Hỏi có chắc chắn muốn sửa không?</p> <p>4a1. Hệ thống sẽ sửa thông tin tài khoản user trong database và hiển thị thông báo đã sửa thông tin tài khoản user thành</p>

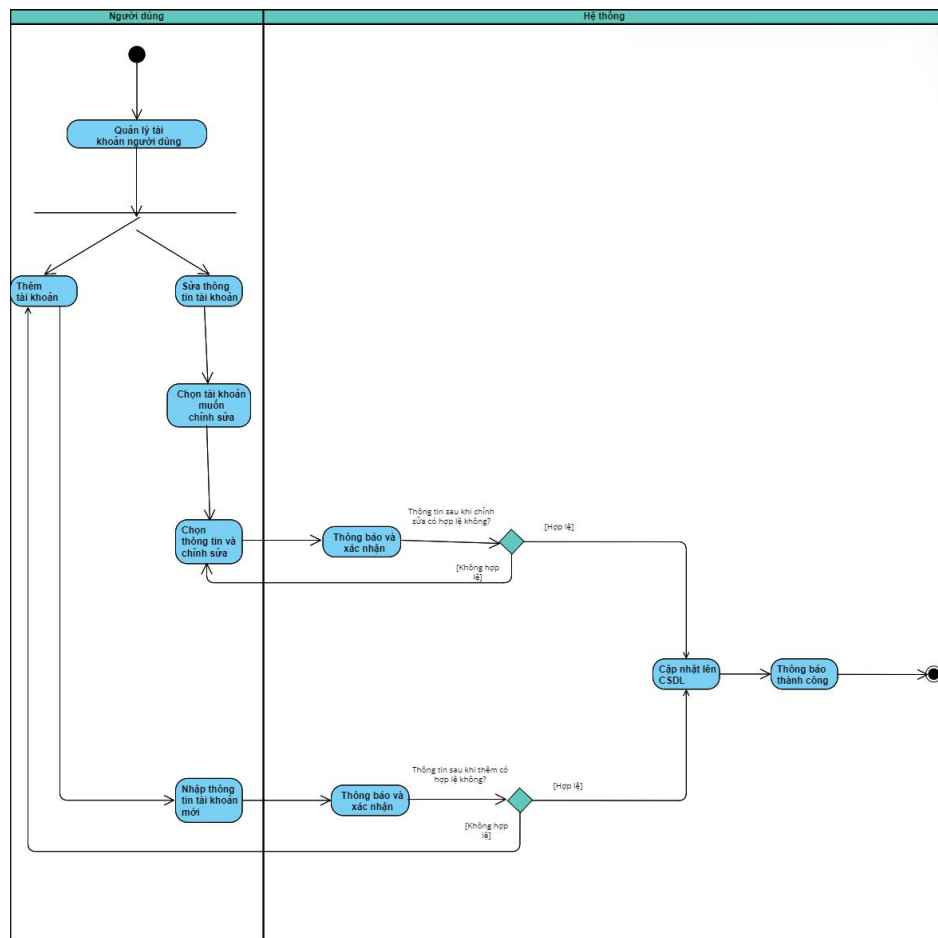
	nhận chỉnh sửa	công trên màn hình
Exception flow	4b. Người dùng ấn từ chối <i>Use case tiếp tục ở bước 1</i>	

3.8.8.3 Sơ đồ tuần tự



Hình 3.26 Sơ đồ tuần tự Use case quản lý tài khoản user

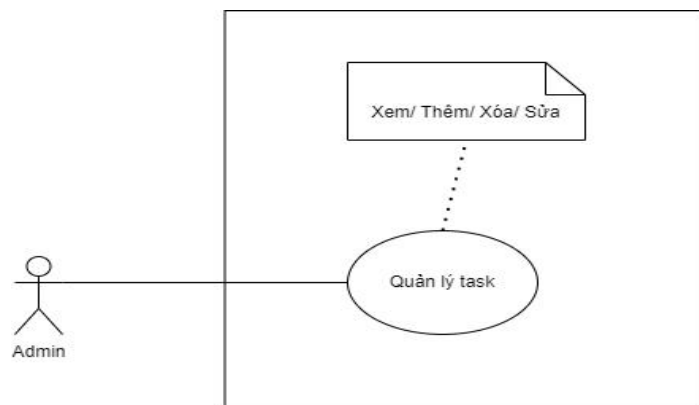
3.8.8.4 Sơ đồ hoạt động



Hình 3.27 Sơ đồ hoạt động Use case quản lý tài khoản user

3.8.9 Use case quản lý nhiệm vụ

3.8.9.1 Sơ đồ use case



Hình 3.28 Use case quản lý nhiệm vụ

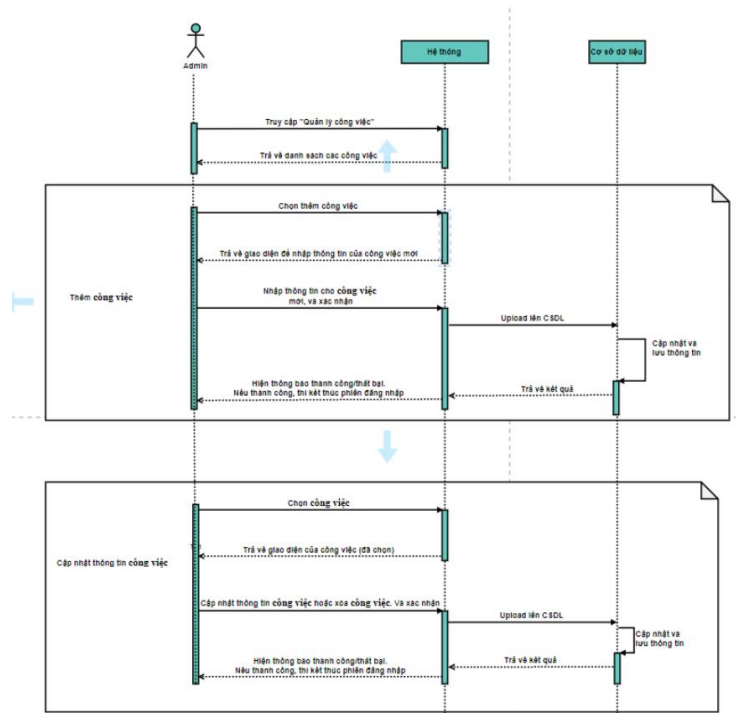
3.8.9.2 Đặc tả use case

Bảng 3.10 Đặc tả use case quản lý nhiệm vụ

Use Case ID	UC14	
Use case	Quản lý nhiệm vụ	
Description	Người dùng có thể thêm và sửa thông tin nhiệm vụ	
Actor	Admin	
Trigger	Admin thêm/sửa nhiệm vụ	
Post - condition	<p>Người dùng chỉnh sửa nhiệm vụ thành công</p> <p>Hệ thống sẽ hiện lên thông báo chỉnh sửa nhiệm vụ thành công</p>	
Basic flow	ACTOR	SYSTEM
	<ol style="list-style-type: none"> 1. Người dùng chọn vào mục quản lý nhiệm vụ 2. Sau đó, người dùng chọn mục thêm nhiệm vụ 3. Người dùng nhập thông tin nhiệm vụ cần thêm 4. Người dùng ấn xác nhận thêm 	<ol style="list-style-type: none"> 1.1. Hệ thống hỏi người dùng có chắc chắn muốn thêm nhiệm vụ vào hệ thống hay không? 2.1. Hệ thống sẽ thêm nhiệm vụ vào database và hiển thị thông báo đã thêm nhiệm vụ thành công trên màn hình

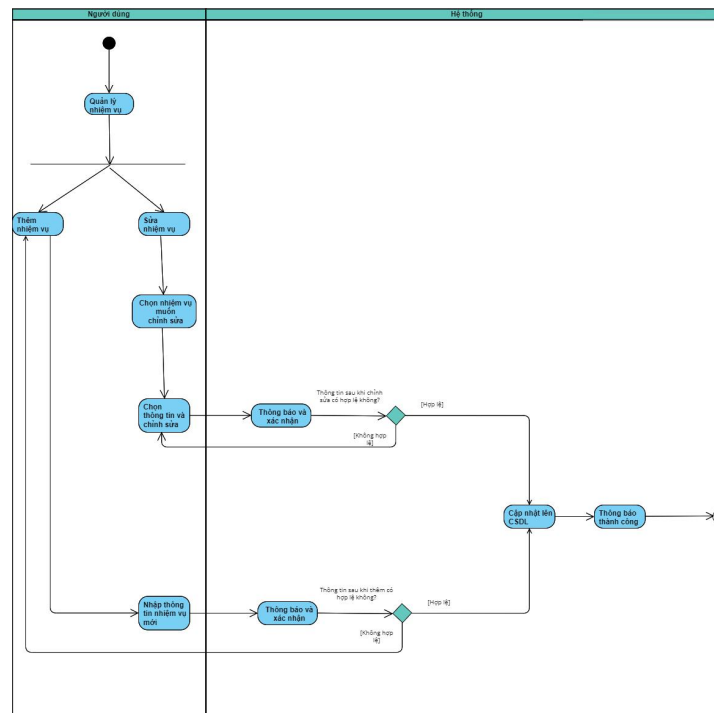
Alternative flow	<p>2a. Người dùng chọn chỉnh sửa nhiệm vụ</p> <p>3a. Người dùng nhập thông tin nhiệm vụ cần sửa</p> <p>4a. Người dùng ấn xác nhận chỉnh sửa</p>	<p>3a1. Hệ thống hỏi người dùng có chắc chắn muốn sửa thông tin nhiệm vụ này hay không?</p> <p>4a1. Hệ thống sẽ sửa thông tin nhiệm vụ trong database và hiển thị thông báo đã sửa thông tin nhiệm vụ thành công trên màn hình</p>
Exception flow	<p>4b. Người dùng ấn từ chối</p> <p><i>Use case tiếp tục ở bước 1</i></p>	

3.8.9.3 Sơ đồ tuần tự



Hình 3.29 Sơ đồ tuần tự Use case quản lý nhiệm vụ

3.8.9.4 Sơ đồ hoạt động



Hình 3.30 Sơ đồ hoạt động Use case quản lý nhiệm vụ

CHƯƠNG 4. THIẾT KẾ GIAO DIỆN

4.1 Màn hình đăng nhập

Email

Mật khẩu

☐ Ghi nhớ mật khẩu [Quên mật khẩu](#)

Đăng nhập

Hình 4.1 Màn hình đăng nhập

4.2 Màn hình trang chủ

24:50

THÁNG 12/2023 **13** Thứ ba

Ngày làm việc:
Giờ làm việc: 08:00-17:00
Check in: 7:50
Check out: 12:00

HỒ SƠ NHÂN SỰ

Hồ sơ Hợp đồng

CÔNG - PHÉP - LƯƠNG

Điểm danh Chăm công Bảng lương

Nghỉ phép Nhiệm vụ ĐK OT

Đk nghỉ phép

CUỘC HỌP - TRAO ĐỔI

Phòng họp Chat

Hình 4.2 Màn hình trang chủ

- Gồm các chức năng:

- Xem hồ sơ
- Xem hợp đồng
- Điểm danh
- Xem dữ liệu điểm danh
- Đăng ký nghỉ phép
- Xem các nhiệm vụ
- Đăng ký OT
- Workspace

4.3 Màn hình xem hồ sơ nhân sự

← Tiêu Viết Khánh

Ngày sinh: Giới tính:

CCCD:

Ngày cấp:

Nơi cấp:

Quốc tịch: Dân tộc:

Tôn giáo: Chức vụ:

Học vấn: Điện thoại:

Email cá nhân:

Email công ty:

Địa chỉ thường trú:

Hình 4.3 Màn hình xem hồ sơ nhân sự

- Ở màn hình này người dùng sẽ xem được các thông tin trong hồ sơ xin việc.

4.4 Màn hình đăng ký phép

Hình 4.4 4.4Màn hình đăng ký phép

- Ở màn hình này người dùng có thể xem các yêu cầu đăng ký nghỉ phép và tạo nghỉ phép mới

4.5 Màn hình thống kê nghỉ phép

STT	Tháng	Số ngày nghỉ
1	Tháng 1	0
2	Tháng 2	0
3	Tháng 3	0
4	Tháng 4	0
5	Tháng 5	0
6	Tháng 6	0
7	Tháng 7	0
8	Tháng 8	0
9	Tháng 9	0
10	Tháng 10	0
11	Tháng 11	0
12	Tháng 12	0

Hình 4.5 Màn hình thống kê nghỉ phép

- Ở màn hình này, người dùng có thể xem số lượng ngày nghỉ phép của cá nhân trong 12 tháng.

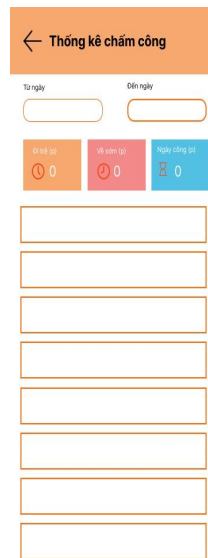
4.6 Màn hình đăng ký OT

Hình 4.6 Màn hình đăng ký OT

4.7 Màn hình chấm công

Hình 4.7 Màn hình chấm công

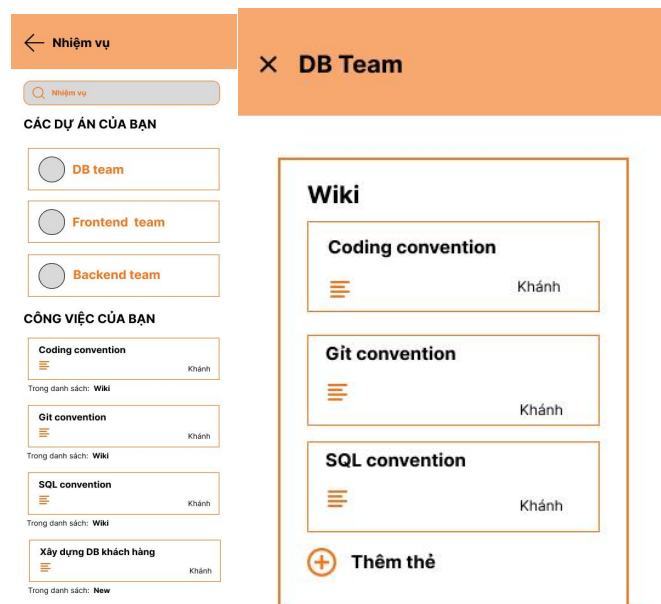
4.8 Màn hình thống kê chấm công



Hình 4.8 Màn hình thống kê chấm công

- Ở màn hình này người dùng sẽ xem được thống kê chấm công trong tháng. Và có thể filter được dữ liệu điểm danh từ ngày - đến ngày

4.9 Màn hình xem nhiệm vụ được giao



Hình 4.9 Màn hình xem nhiệm vụ được giao

- Ở màn hình này, dùng để xem các dự án, nhiệm vụ được giao cần phải hoàn thành.

4.10 Màn hình chi tiết nhiệm vụ

X
Xây dựng DB khách hàng

Trong danh sách: **New**

Xây dựng db về người dùng

Ngày bắt đầu

Ngày hết hạn

Thành viên

Hoạt động

Vũ đã di chuyển thẻ này từ danh sách **New** tới danh sách **Doing**.
19:32 19 thg 12

Khánh đã thêm thẻ này vào danh sách **New**.
19:32 19 thg 12

Bình luận

Hình 4.10 Màn hình chi tiết nhiệm vụ

- Ở màn hình này người dùng có thể xem được chi tiết nội dung của nhiệm vụ: ngày bắt đầu, ngày kết thúc, thành viên tham gia cùng, và có thể trao đổi về các nhiệm vụ với người khác.

TÀI LIỆU THAM KHẢO

Viettel Cyberspace Center. (2023). *Web API là gì? Ưu và nhược điểm nổi bật Của web API*. Truy cập từ 2/2024, từ website VTCC - Công Thông Tin Công Nghệ Mới Nhất: <https://vtcc.vn/web-api-la-gi/>

Maciej Budziński. (2023). *What is react native? Complex guide for 2023*. Digital Acceleration Company.(2/2024), website Netguru: <https://www.netguru.com/glossary/react-native>