**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**KHOA CÔNG NGHỆ THÔNG TIN**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

****

**ĐỒ ÁN TỐT NGHIỆP**

**Đề tài : Phát triển phần mềm quản lý dự án**

**Giảng viên hướng dẫn: PGS.TS. Nguyễn Văn Long**

**Sinh Viên: Lê Minh Nghĩa-160713708**

**Lớp: Công Nghệ Thông Tin 2 K-57**

**Hà Nội ngày 05 tháng 05 năm 2020**

**Lời cảm ơn**

Để hoàn thành đồ án tốt nghiệp này, em xin chân thành gửi lời cảm ơn đến PGS.TS Nguyễn Văn Long, giảng viên trực tiếp phụ trách hướng dẫn đề tài đồ án tốt nghiệp. Sự hướng dẫn tận tình của thầy đã giúp em có thêm nhiều kiến thức, khắc phục được những lỗi em thường mắc phải để đồ án được hoàn thiện tốt hơn. Qua những hướng dẫn tận tâm của thầy đã giúp chúng em hiểu sâu hơn về những kiến thức đã học, và trau dồi thêm những kiến thức mới bổ ích.

Em xin chân thành gửi lời cảm ơn quý thầy cô trong bộ môn cũng như quý thầy cô trong trường đã tận tình giúp đỡ và trang bị cho em những kiến thức cơ bản, cần thiết để em có thể hoàn thành tốt đồ án tốt nghiệp này.

Mặc dù có nhiều cố gắng để thực hiện đồ án tốt nghiệp một cách hoàn chỉnh nhất. Song do năng lực còn nhiều hạn chế nên không thể tránh khỏi những thiếu sót nhất định. Em rất mong nhận được sự đóng góp của quý thầy cô để đồ án tốt nghiệp của em được hoàn chỉnh hơn.

Một lần nữa, xin chân thành cảm ơn và luôn mong nhận được những tình cảm chân thành của tất cả mọi người.

Hà nội, ngày 05 tháng 05 năm 2020

Chương 1 Tổng quan về đề tài

1.1 Tính cấp thiết của đề tài

Hiện nay công nghệ thông tin đã đi vào đời sống, đi vào các doanh nghiệp với một phương thức hoạt động hoàn toàn mới mẻ, sáng tạo và nhanh chóng, tiết kiệm được nhiều thời gian, công sức mà không mất đi sự chính xác, còn làm cho công việc được thuận lợi và phát triển lên rất nhiều. Đặc biệt, nó đã đánh dấu một bước ngoặt trong việc áp dụng tin học vào trong hệ thống quản lý, doanh nghiệp có thể thu thập, xử lý, phổ biến thông tin, một cách nhanh chóng, chính xác và có hiệu quả.

Công ty HBLAB được thành lập vào năm 2015, hiện nay là một trong những công ty hàng đầu về mặt phát triển phầm mềm outSource cho Nhật Bản

Bên cạnh những thành công ban đầu, công ty cũng gặp phải không ít khó khăn do sự bất cập trong quản lý các dự án nội bộ công ty dẫn đến thất thoát tài nguyên dự án, thiếu hụt nguồn nhân lực, quản lý lỏng lẻo chưa hiệu quả.

Bài toán đặt ra cho công ty đó là làm sao để hoàn thiện và phát triển mô hình quản lý dự án, trên cơ sở đó xây dựng hệ thống thông tin quản lý dự án mang lại hiệu quả công việc cho công ty. Hiện nay trên thị trường có một số phần mềm quản lý dự án như Vinno, PMS, GRM và một số website quản lý dự án như Zoho Project, [Asana](https://app.asana.com/), [Redmine](http://demo.redmine.org/projects/), [ViewPath](http://www.viewpath.com/), [Collabtive](http://www.collabtive.com/), [Whodo](http://www.whodo.es/home)... Để sử dụng những công cụ này trong quản lý dự án công ty cần phải chi trả phí sử dụng, tuy nhiên những phần mềm/website này không thật sự phù hợp với hoạt động quản lý dự án của công ty HBlab, và có nhiều phần dư thừa và thiếu sót, thiếu tập trung và bảo mật thông tin trong công ty. Vì vậy khóa luận đề xuất việc phân tích thiết kế hệ thống thông tin quản lý dự án công ty Hblab sao cho phù hợp với hoạt động công ty, tận dụng cơ sở dữ liệu nhân sự và thiết lập phong cách làm việc khoa học tập trung đem lại hiệu quả cao.

1.2 Mục tiêu nghiên cứu của đề tài

Thứ nhất, hệ thống hóa kiến thức một số lý luận về phân tích và thiết kế hệ thống thông tin trong doanh nghiệp như: khái niệm, phân loại, phương pháp phân tích thiết kế hệ thống.

Thứ hai, hệ thống hóa kiến thức lý luận cơ bản về mô hình quản lý dự án nói chung của công ty HBlab nói riêng.

Thứ ba, trên cơ sở nghiên cứu thực trạng tình hình quản lý dự án tại công ty HBlab, từ đó đưa ra một số đề xuất, giải pháp có thể áp dụng, tiến hành phân tích thiết kế hệ thống thông tin quản lý dự án trên website công ty nhằm tạo ra một hệ thống thông tin quản lý dự án phù hợp và mang tính thiết thực đối với hoạt động quản lý dự án đem lại hiệu quả trong công việc.

1.3 Đối tượng nghiên cứu

Đối tượng nghiên cứu của đề tài bao gồm:

Thứ nhất, dự án và các thông tin về dự án, quy trình quản lý dự án, những người tham gian và có ảnh hưởng đến hoạt động quản lý dự án.

Thứ hai, các kiến thức phân tích thiết kế hướng đối tượng.

1.4 Phạm vi nghiên cứu

*Về không gian:* Tìm hiểu mô hình quản lý dự án trong môi trường sản xuất kinh doanh của công ty HBlab.

*Về thời gian:* Do điều kiện thời gian không cho phép nên em không thể thu thập được đầy đủ thông tin về tất cả hoạt động của doanh nghiệp, trong đề tài chỉ tập trung nghiên cứu những vấn đề cần thiết đặt ra trong doanh nghiệp liên quan tới quản lý dự án trong thời gian từ năm 2018 đến năm 2020.

1.5 Phương pháp nghiên cứu đề tài

Khóa luận được thực hiện trên cơ sở vận dụng tổng hợp các phương pháp nghiên cứu như phân tích, so sánh, thống kê, tổng hợp, việc phân tích thiết kế hệ thống thông tin tiến hành theo phương pháp hướng đối tượng

1.6 Kết cấu khóa luận

Khóa luận chia làm 4 phần chính

Chương 1. Tổng quan về đề tài.

Chương 2: Cơ sở lý thuyết và công nghệ sử dụng

Chương 3. Phân tích và thiết kế hệ thống.

Chương 4. Xây dựng cơ sở dữ liệu.

Chương 5. Thiết kế giao diện

Chương 2 Cơ sở lý thuyết

2.1 Phân loại web

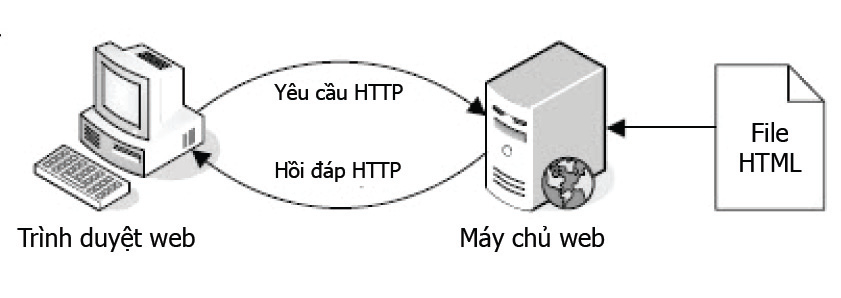
2.1.1 Web tĩnh

Web tĩnh là những website không có hệ thống quản lý nội dung và người dùng không thể chỉnh sửa được hoặc có thể chỉnh sửa, thay đổi dữ liệu được. Hiện nay web tĩnh hầu như ít tồn tại hoặc chỉ tồn tại với những công ty chuyên về thiết kế website, họ sử dụng web tĩnh bởi họ có kiến thức và dễ dàng chỉnh sửa nội dung khi cần.

Web tĩnh được làm từ gì?

Về kiến thức cơ bản thì web tĩnh thường được xây dựng từ CSS, HTML, JAVASCRIPT (DHTML), hiện nay có thêm công nghệ HTML5 & CSS3.

Nếu bạn chạy trên máy tính của bạn thì với web tĩnh bạn có thể đặt tại đâu cũng chạy được vì bản chất nó chỉ là một file bình thường.



#### Hình 1: Mô hình hoạt động của web tĩnh

Ưu điểm của web tĩnh:

* Về giao diện Designer có thẻ thiết kế theo kiểu mới lạ.
* Tốc độ truy cập nhanh bởi nó chỉ là những file HTML.
* Chi phí đầu tư thấp bởi bạn không phải trả tiền nhiều cho Coder.
* Thân thiện với bộ máy tìm kiếm bởi bạn có thể đặt tên file tùy ý (ten-file.html, tieu-de-tin-tuc.html).

Nhược điểm của web tĩnh:

* Khó quản lý nội dung.
* Khó nâng cấp bảo trì.
* Mỗi khi thay đổi phải vào file HTML, CSS hoặc JAVASCRIPT để chỉnh sửa.

Khi nào nên sử dụng web tĩnh?

* Nếu bạn là doanh nghiệp muốn tự mình làm website thì bạn có thể học các kiến thức căn bản và tự làm một Web tĩnh cho mình.
* Nội dung website ít khi cập nhật và ít nên bạn muốn tiết kiệm chi phí.
* Website bạn nhỏ và bạn thuê luôn người chuyên về web để quản trị.

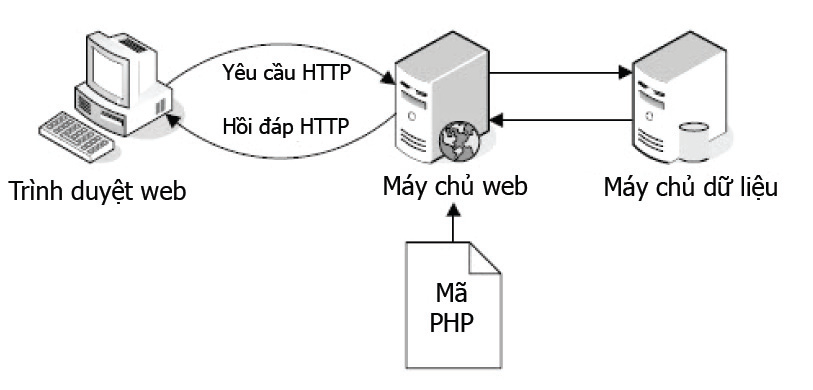
Trên thực tế thì hiện nay chỉ có các công ty thiết kế web họ mới làm web tĩnh bởi vì họ có khả năng chỉnh sửa được chứ khách hàng thì họ không có khả năng đó. Hiện nay có các trang web ONE PAGE có các hiệu ứng slide khá độc nên cũng có thể họ sử dụng web tĩnh hoặc tĩnh một phần trong đó.

1.1.2 Web động

Ngược lại với web tĩnh, web động là những website có hệ thống quản lý nội dung và người dùng có thể chỉnh sửa nội dung được. Ví dụ như trang vnexpress.net hay 24h.com.vn thì đó là những web động bởi vì họ có thể đăng tin mới, chỉnh sửa tin mới và chỉnh sửa danh mục menu, ...

Web động được làm từ gì?

Web động sử dụng các công nghệ như HTML, CSS, HTML5, CSS3, JAVASCRIPT, ... và điều đặc biệt là có sử dụng một ngôn ngữ lập trình server như PHP, một hệ quản trị cơ sở dữ liệu như MySQL, vì vậy web động phải chạy trong máy chủ. Nếu đặt trên máy tính của bạn thì bạn phải cài các Web server ảo như vertrigo, Xampp, wampp, ...



#### Hình 2: Mô hình hoạt động của web động

Ưu điểm của web động:

* Dễ dàng quản lý nội dung
* Dễ dàng nâng cấp và bảo trì
* Có thể xây dụng được web lớn
* Thường sử dụng tương tác với người dùng cao
* Hiện nay vẫn SEO tốt bởi ta có thể sử dụng chức năng Rewrite URL để chuyên URL thân thiện

Nhược điểm của web động:

* Chi phí xây dựng cao
* Nếu web lớn có thể cần thêm nhân sự chuyên ngành

Khi nào nên sử dụng web động?

* Bạn làm website thương mại điện tử bán hàng.
* Bạn làm web giới thiệu sản phảm công ty.
* Bạn làm web tin tức, blog cá nhân.
* Web bạn tầm cỡ lớn.

## 1.2 HTML

HTML (tiếng Anh, viết tắt cho HyperText Markup Language, hay là "Ngôn ngữ Đánh dấu Siêu văn bản") là một ngôn ngữ đánh dấu được thiết kế ra để tạo nên các trang web với các mẩu thông tin được trình bày trên World Wide Web. Cùng với CSS và JavaScript, HTML tạo ra bộ ba nền tảng kỹ thuật cho World Wide Web. HTML được định nghĩa như là một ứng dụng đơn giản của SGML và được sử dụng trong các tổ chức cần đến các yêu cầu xuất bản phức tạp. HTML đã trở thành một chuẩn Internet do tổ chức World Wide Web Consortium (W3C) duy trì. Phiên bản chính thức mới nhất của HTML là HTML 4.01 (1999). Sau đó, các nhà phát triển đã thay thế nó bằng XHTML. Hiện nay, HTML đang được phát triển tiếp với phiên bản HTML5 hứa hẹn mang lại diện mạo mới cho Web.

Bằng cách dùng HTML động hoặc Ajax, lập trình viên có thể được tạo ra và xử lý bởi số lượng lớn các công cụ, từ một chương trình soạn thảo văn bản đơn giản – có thể gõ vào ngay từ những dòng đầu tiên – cho đến những công cụ xuất bản WYSIWYG phức tạp. Hypertext là cách mà các trang Web (các tài liệu HTML) được kết nối với nhau. Và như thế, đường link có trên trang Web được gọi là Hypertext.Như tên gọi đã gợi ý, HTML là ngôn ngữ đánh dấu bằng thẻ (Markup Language), nghĩa là bạn sử dụng HTML để đánh dấu một tài liệu text bằng các thẻ (tag) để nói cho trình duyệt Web cách để cấu trúc nó để hiển thị ra màn hình.

## 1.3 CSS

CSS là chữ viết tắt của cụm từ tiếng Anh "Cascading Style Sheet", dịch ra tiếng Việt thì thực sự mình cũng không thể dịch cho chuẩn được blush, nhưng về ý nghĩa thì CSS dùng để trình bày bổ cục của một Website dựa vào các thẻ HTML. Nói nôm na thì CSS sẽ tác động thay đổi cách hiển thị mặc định của các thẻ HTML nhằm thân thiện với người dùng hơn hay nhằm một mục đích cụ thể nào đó.

Có một ví dụ trực quan để diễn tả nhiệm vụ của CSS như sau, giả sử bạn xây nhà và bạn sẽ sử dụng gạch để xây dựng ngôi nhà. Nhưng để trang trí cho ngôi nhà thì bạn sẽ sử dụng một số đồ nội thất khác như sơn màu, ... thì lúc này sơn màu sẽ tác động làm thay đổi color mặc định của gạch. Nói ví von thi màu sơn chính là CSS và gạch chính là HTML.

CSS đóng một vai trò rất quan trọng khi bạn xây dựng giao diện Website. Thông thường bạn sẽ sử dụng một chương trình thiết kế như Photoshop để tạo giao diện, sau đó các Frontend Developer sẽ dựa vào file đó kết hợp với HTML & CSS để convert thành một file có đuôi mở rộng là .html, file File này sẽ chứa code HTML và CSS.

## 1.4 Bootstrap

Bootstrap là gì?

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

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

Lịch sử Bootstrap

Bootstrap là dược phát triển bởi Mark Otto và Jacob Thornton tại Twitter. Nó được xuất bản như là một mã nguồn mở vào tháng 8 năm 2011 trên GitHub. Tính ra đến thời điểm mình viết bài viết này nó cũng đã phát triển được 3 năm rồi. Bản bootstrap mới nhất bây giờ là bootstrap 4.

Tại sao bạn nên sử dụng Bootstrap?

Những điểm thuận lợi khi bản sử dụng bootstrap:

Rất dễ để sử dụng: Nó đơn giản vì nó được base trên HTML, CSS và Javascript chỉ cẩn có kiến thức cơ bản về 3 cái đó là có thể sử dụng bootstrap tốt.

Tính năng Responsive: Bootstrap’s xây dựng sẵn reponsive css trên các thiết bị phones, tablets, và desktops

Mobile: Trong Bootstrap 3 mobile-first styles là một phần của core framework

Tương thích với trình duyệt: Nó tương thích với tất cả các trình duyệt (Chrome, Firefox, Internet Explorer, Safari, and Opera) nhưng lưu ý vì em IE vẫn rất kén với IE phiên bản cũ vì thế việc IE9 hay IE8 đổ xuống không support là chuyện bình thường. Các bạn có dùng nên luy ý điểm này theo như mình dùng làm dự án thì nó support tuyệt vời trên IE10 đổ lên nhé, thấp nhất có lẽ là IE9 thôi còn IE8 thì không nên sử dụng.

## 1.5 Javascript

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

1.6 Vuejs

Vue.js là một framework linh động (nguyên bản tiếng Anh: progressive – tiệm tiến) dùng để xây dựng giao diện người dùng (user interfaces). Khác với các framework nguyên khối (monolithic), Vue được thiết kế từ đầu theo hướng cho phép và khuyến khích việc phát triển ứng dụng theo từng bước. Khi phát triển lớp giao diện (view layer), người dùng chỉ cần dùng thư viện lõi (core library) của Vue, vốn rất dễ học và tích hợp với các thư viện hoặc dự án có sẵn. Cùng lúc đó, nếu kết hợp với những kĩ thuật hiện đại như SFC (single file components) và các thư viện hỗ trợ, Vue cũng đáp ứng được dễ dàng nhu cầu xây dựng những ứng dụng một trang (SPA - Single-Page Applications) với độ phức tạp cao hơn nhiều.

## 1.7 Laravel

Laravel là một PHP Framework mã nguồn mở và miễn phí, được phát triển bởi Taylor Otwell và nhắm mục tiêu hỗ trợ phát triển các ứng dụng web theo cấu trúc model- view- controller (MVC). Những tính năng nổi bật của Laravel bao gồm cú pháp dễ hiểu- rõ ràng, một hệ thống đóng gói Modular và quản lý gói phụ thuộc, nhiều cách khác nhau để truy cập vào các cơ sở dữ liệu quan hệ, nhiều tiện ích khác nhau hỗ trợ việc triển khai vào bảo trì ứng dụng.

Vào khoảng Tháng 3 năm 2015, các lập trình viên đã có một cuộc bình chọn PHP framework phổ biến nhất, Laravel đã giành vị trí quán quân cho PHP framework phổ biến nhất năm 2015, theo sau lần lượt là Symfony2, Nette, CodeIgniter, Yii2 vào một số khác. Trước đó, Tháng 8 năm 2014, Laravel đã trở thành project PHP phổ biến nhất và được theo dõi nhiều nhất trên Github.

Laravel được phát hành theo giấy phép MIT, với source code được lưu trữ tại Gitthub.

## 1.8 Mysql

MySQL là hệ quản trị cơ sở dữ liệu tự do nguồn mở phổ biến nhất thế giới và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng. Vì MySQL là cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh. Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng có truy cập CSDL trên internet. Người dùng có thể tải về MySQL miễn phí từ trang chủ. MySQL có nhiều phiên bản cho các hệ điều hành khác nhau: phiên bản Win32 cho các hệ điều hành dòng Windows, Linux, Mac OS X, Unix, FreeBSD, NetBSD, Novell NetWare, SGI Irix, Solaris, SunOS...

MySQL là một trong những ví dụ rất cơ bản về Hệ Quản trị Cơ sở dữ liệu quan hệ sử dụng Ngôn ngữ truy vấn có cấu trúc (SQL).

MySQL được sử dụng cho việc bổ trợ NodeJs, PHP, Perl, và nhiều ngôn ngữ khác, làm nơi lưu trữ những thông tin trên các trang web viết bằng NodeJs, PHP hay Perl…

## 1.9 Gitlab

GitLab là một phần mềm có nhiệm vụ quản lý kho code Git. Gitlab sở hữu các tính năng đơn giản, góp phần to lớn trong việc giúp các doanh nghiệp, cá nhân, tổ chức lưu trữ code một cách nhanh chóng vô cùng, người dùng hoàn toàn có thể truy cập mọi lúc mọi nơi miễn là có kết nối Internet.

Gitlab cũng có khá nhiều điểm tương đồng với GitHub nhưng GitHub đi theo hướng kinh doanh nhiều hơn, bởi vì nếu bạn sở hữu kho code riêng và muốn ẩn chúng khỏi cộng đồng hoặc mở rộng hơn nữa thì bạn sẽ phải trả phí cho dịch vụ này. Gitlab hoàn toàn ngược lại, bạn có thể ẩn kho code của mình, không công khai chúng cho bất kỳ ai, trong trường hợp vượt quá ngưỡng miễn phí thì bạn mới phải mất phí để mua thêm dịch vụ.

1.10 Docker

Docker là một nền tảng để cung cấp cách để building, deploying và running ứng dụng một cách dễ dàng trên nền tảng ảo hóa.

Các thành phần cơ bản của docker

-Image: Là file ảnh, file nền của một hệ điều hành, một nền tảng, một ngôn ngữ (vd: ubuntu image, ruby image, rails image, mysql image…)

-Container: Là một máy ảo, được cấu thành từ 1 image và được đắp thêm 1 lớp “trang trí” writable-file-layer. Các thay đổi trong máy ảo này (cài thêm phần mềm, tạo thêm file…) sẽ được lưu ở lớp trang trí này.

-DockerEngine: quản lý việc bạn tạo image, chạy container, dùng image có sẵn hay tải image chưa có về, kết nối vào container, thêm, sửa, xóa image và container.

Ưu điểm

-Tính đồng nhất

+ Khi nhiều người cùng phát triển trong cùng một dự án sẽ không bị sự sai khác về mặt môi trường

-Đóng Gói

+ Có thể ẩn môi trường bao gồm cả app vào trong một gói được gọi là container

+ Có thể test được các container

+ Việc bỏ hay tạo lại container rất dễ dàng

-Linh động (Tính nhất quán)

+ Có thể test container được dùng để phát triển bằng CI

+ Có thể deploy container đã được test bằng CI lên server

+ Có thể scale container đã được deploy

3. Phân tích và thiết kế hệ thống

3.1 Tổng quan về đề tài

3.1.1 Khảo sát:

Qua quá trình khoản sát tại công ty HBLab em có thể thu về kết quả như sau :

a, Các quyền trong hệ thống

Hệ thống sẽ gồm 3 quyền: Admin, Leader, Member

+Admin là bộ phận giám đốc của công ty gồm 2 thành viên là 2 giám đốc tại Nhật Bản và Viêt Nam

+Leader là toàn bộ những người đứng đầu của các Team trong công ty

+Member là toàn bộ các thành viên còn lại bên trong công ty

b, các khái niệm liên quan tới nghiệp vụ:

-Busy rate: Đánh giá mức độ bận rộn của 1 user trong 1 tháng

Max: 100%

Min: 0%

Busy rate của user trong 1 tháng được tính bằng công thức

Trong đó

n: Số dự án tham gia trong tháng đó

Số ngày làm việc: số ngày nhân viên đi làm trừ cuối tuần và ngày nghỉ

Tổng số ngày làm việc: Tổng số ngày nhân viên phải đi làm trong tháng

-Buget: Là số ManMonth làm dự án mà bên công ty đã thoản thuận cùng với khách hàng

Ví dụ: Buget = 4, dự án kéo dài 2 tháng. Có nghĩa là mỗi tháng cần 2 manmonth làm việc (2 user làm việc)

-UserMM: Là số ManMonth thực tế làm dự án

-EE: là tỉ lệ đánh giá việc sử dụng resource của dự án

-KPI: là thông số sau cùng để đánh giá mức độ hoàn thành của dự án , bao gồm:

+CSS: điểm đánh giá của khách hàng về thái độ và trình độ làm việc của Team

+Leakage: Số bug trong quá trình làm dự án sau khi đã bàn giao với khách hành

+Timeliness: Đánh giá xem Team làm dự án có bàn giao sản phẩm đúng hạn hay không

3.2 Phân tích đặc tả yêu cầu

3.2.1 Yêu cầu chức năng

-Quản lý người nhân sự công ty:

+Cho phép admin có thể tạo tài khoản cho bất kì nhân sự nào khi vào công ty và gán vai trò tương ứng cho họ

+Tài khoản đó là tài khoản đăng nhập của user vào hệ thống. Admin có thể thông qua tài khoản đó để xem được resource, busy rate của user trong 1 khoảng thời gian nhất định

+Admin có quyền update thông tin của user

-Quản lý các team của công ty:

+ Trong công ty sẽ chia thành các Team nhỏ đề dễ dàng quản lý, chức năng này cho phép admin, leader có thể quản lý cập nhật thông tin cho các Team

-Quản lý kì nghỉ:

+Cho phép admin cập nhật thông tin các ngày nghỉ trong tuần hoặc trong tháng.

-Quản lý dự án:

+Admin có thể quản lý toàn bộ các dự án công ty: Thêm, update

+Chức năng cho phép Admin có thể tạo ra 1 dự án

+Quản lý toàn bộ những thông tin xung quang dự án như:

.Phase của dự án

.Member của dự án

.Resource của dự án  
.KPI của dự án

-Xem Thống kê về MM và EE:

+ Chức năng này cho phép admin có thể xem được thống kê về số EE và MM của toàn từng Team và toàn bộ công ty

3.2.3 Yêu cầu phi chức năng

-Hệ thống sử dụng hệ quản trị cơ sở dữ liệu MySQL: Hệ quản trị cơ sở dữ liệu MySQL hoàn toàn miễn phí, ổn định và an toàn. MySQL đưa ra tính năng bảo mật đặc biệt chắc chắn dữ liệu sẽ được bảo mật tuyệt đối. Bên cạnh đó MySQL có khả năng chèn dữ liệu nhanh, và hỗ trợ mạnh cho các chức năng chuyên dụng của website như tìm kiếm văn bản nhanh. Do đó, MySQL là lựa chọn tốt nhất khi xây dựng một hệ thống thông tin trên website.

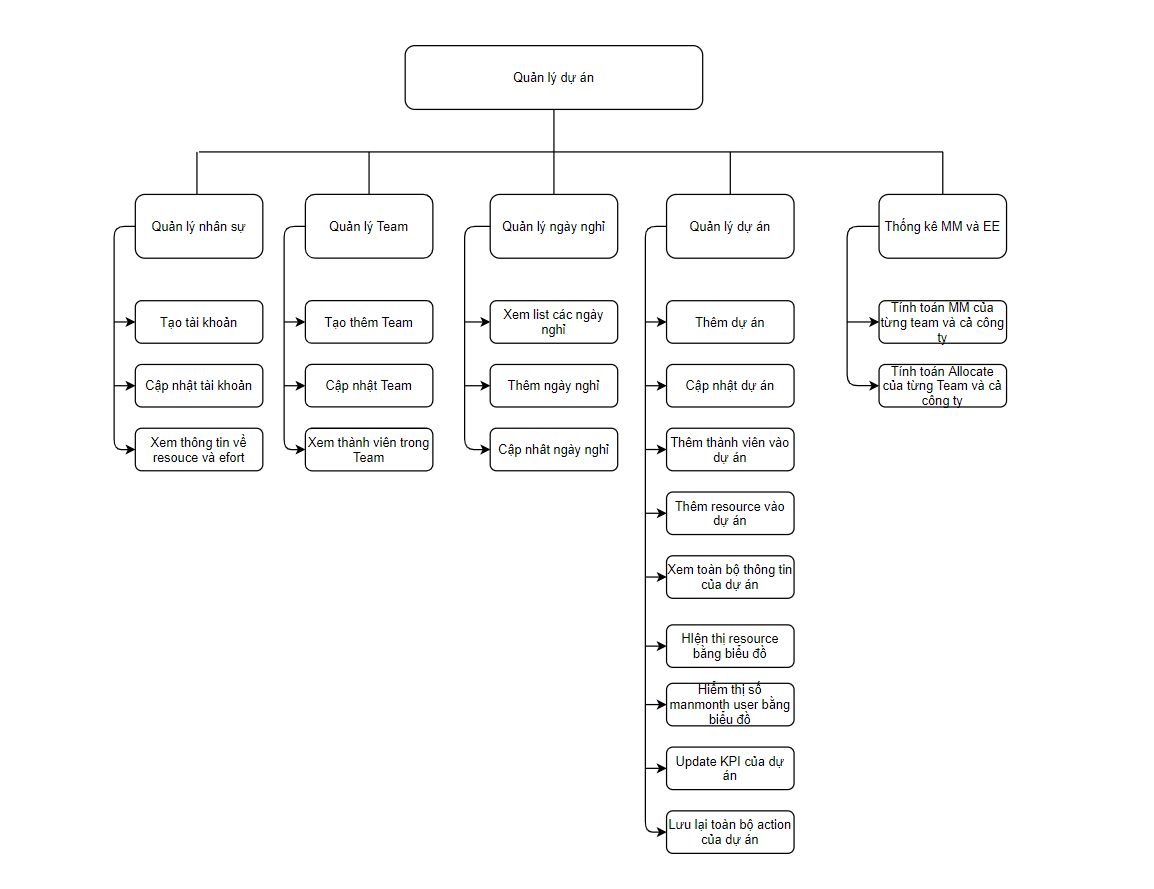
-Hệ thống sử dụng cơ sở dữ liệu tập trung: Cơ chế điều khiển tập trung, độc lập dữ liệu, giảm bớt dư thừa, cơ cấu vật lý phức tạp đối với khả năng truy cập, toàn vẹn, hồi phục, điều khiển tương tranh, biệt lập và an toàn dữ liệu.

-Chương trình có tính tái sử dụng, khả năng tiến hóa cao.

-Tốc độ truy xuất, xử lý nhanh, hiệu quả với lượng dữ liệu lớn.

-Giao diện đẹp, thân thiện với người dùng.

3.2.3 Sơ đồ phân giã chức năng



Hình 3.1 sơ đồ phân giã chức năng

3.3 Biểu đồ UseCase

3.3.1 Xác định các tác nhân

- Thông qua quá trình phân tích dự án thì các tác nhân của hệ thống là :

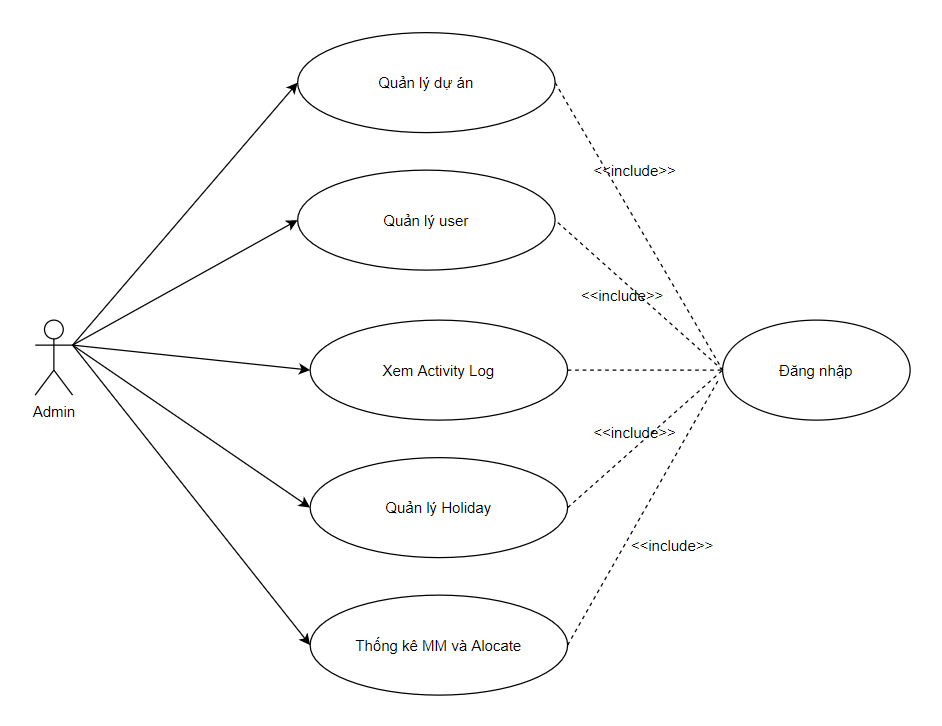
+ Admin

+Leader

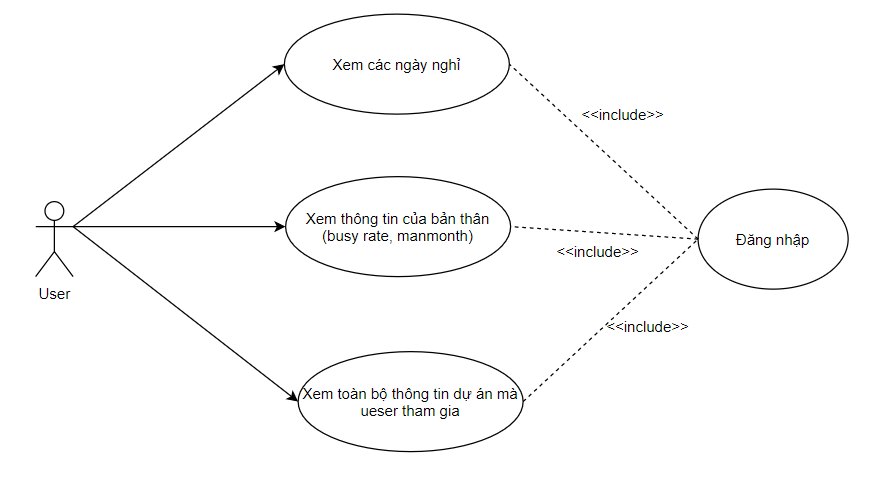
+Member

3.3.2 Biểu đồ UseCase

3.3.2.1 Sơ đồ usecase của hệ thống

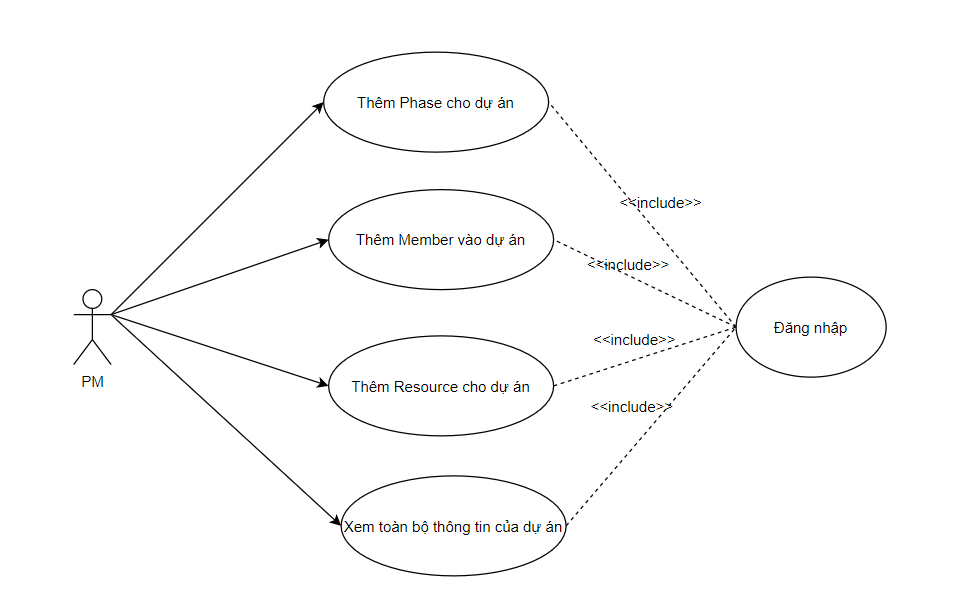


Sơ đồ usecase của Admin

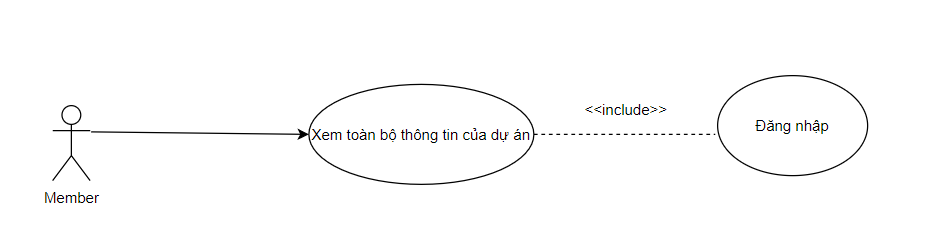


Sơ đồ usecase của Member

3.3.2.2 Sơ đồ usecase của dự án



Usecase của PM dự án



Sơ đồ usecase của member dự án

3.4 Đặc tả và thực hiện hóa Usecase

3.4.1 Usecase “Đăng nhập”

a) Tóm tắt

Usecase này mô tả người sử dụng đăng nhập vào hệ thống quản lý dự án. Khi một người sử dụng muốn dùng hệ thống phải đăng nhập vào hệ thống với một đặc quyền cho trước. Người sử dụng đăng nhập bằng tài khoản do Admin cung cấp khi vào công ty

b) Dòng sự kiện

b.1) Dòng sự kiện chính

Usecase này bắt đầu khi một người sử dụng yêu cầu đăng nhập vào hệ thống. Hệ thống hiển thị màn hình Đăng nhập của tài khoản. Người sử dụng nhập email cá nhân do công ty cấp, nhập mật khẩu. Nếu đăng nhập thành công, hệ thống hiển thị màn hình chính

b.2) Dòng sự kiện phụ

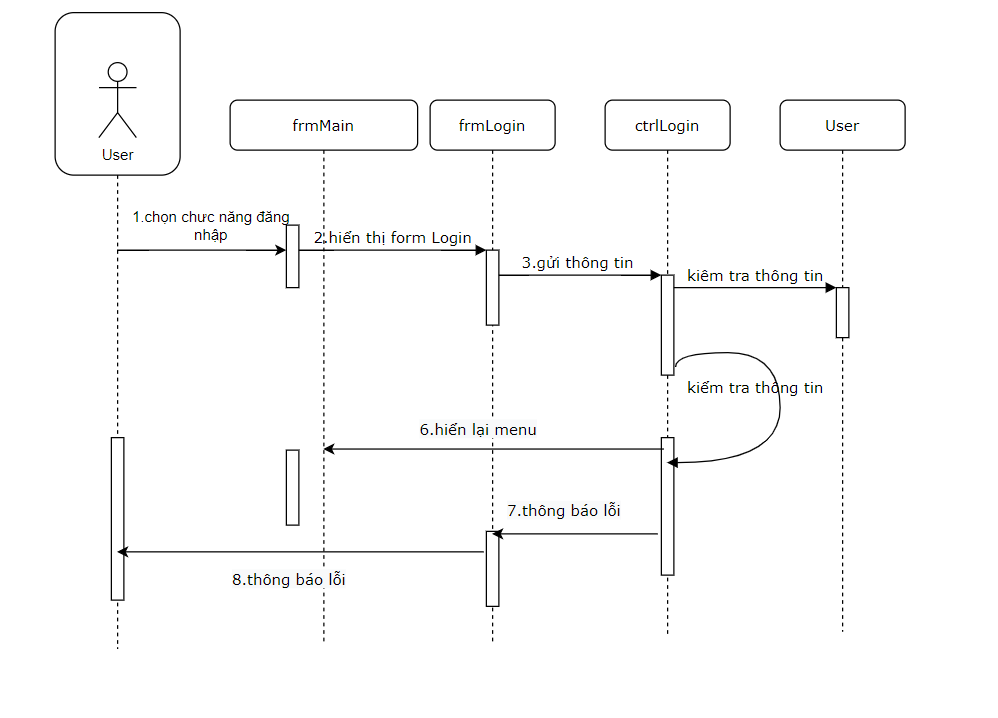
Nếu tài khoản hoặc mật khẩu sai thì hệ thống sẽ hiển thị một thông báo lỗi.Người sử dụng có thể chọn trở về đầu dòng sự kiện chính hoặc hủy bỏ đăng nhập, lúc này usecase kết thúc.

c) Điều kiện tiên quyết

Không có.

d) Post Condition

Nếu đăng nhập thành công, hệ thống sẽ tạo ra một phiên làm việc cho người sử dụng và mở màn hình chính của hệ thống. Ngược lại, trạng thái không đổi.



Biểu đồ trình tự login

3.4.2 Usecase “Đăng xuất”

a) Tóm tắt

Usecase này bắt đầu khi người sử dụng muốn thoát khỏi hệ, kết thúc phiên làm việc.

b) Dòng sự kiện

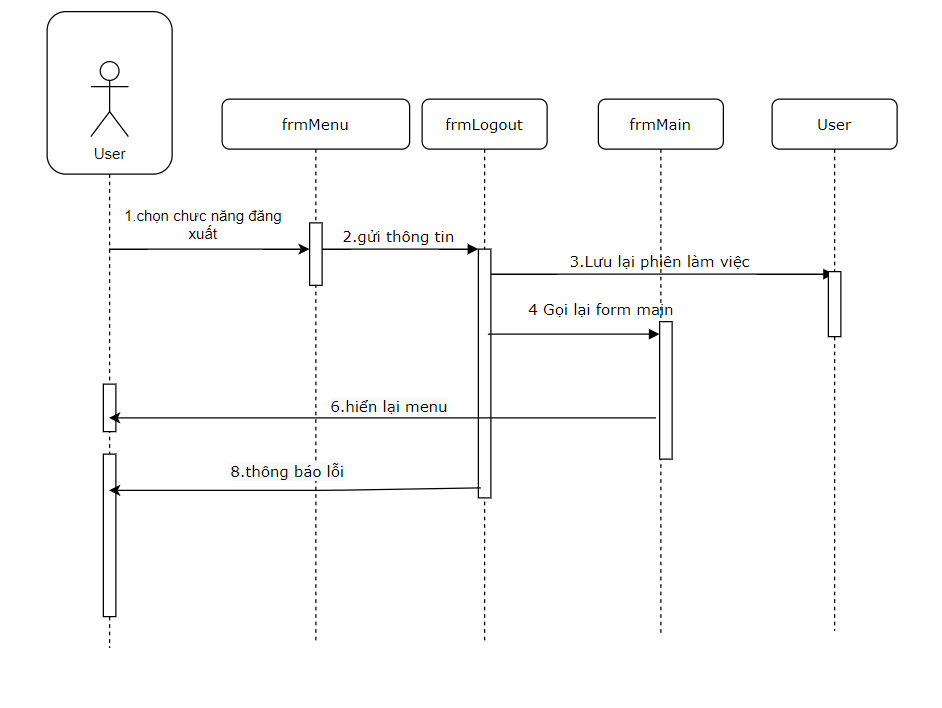
Người sử dụng chọn chức năng đăng xuất trên màn hình chính. Hệ thống lưu phiên làm việc của người sử dụng, đóng màn hình chính, trở về màn hình đăng nhập.

c) Điều kiện tiên quyết

Không có.

d) Post Condition

Nếu usecase thành công, quá trình đăng xuất được đóng. Ngược lại, trạng thái hệ thống giữ nguyên không đổi.



Biểu đồ trình tự LogOut

3.4.3 Usecase “Quản lý người dùng”

a) Tóm tắt

Usecase này bắt đầu khi Admin muốn quản lý thông tin người sử dụng. Usecase cho phép thực hiện các thao tác thêm, sửa đổi, xóa thông tin người sử dụng.

b) Dòng sự kiện

Usecase bắt đầu khi Admin truy cập vào trang chức năng quản lý người sử dụng.

Khi đó hệ thống sẽ trả ra danh sách toàn bộ member hiện tại đang làm việc trong công ty với đầy đủ các thông tin cơ bản như Fullname, email và quyền của member đó trong hệ thống. Admin có thể sử dụng thanh tìm kiếm bên trên để có thể lọc ra user muốn tìm kiếm một cách nhanh nhất

Sau khi Admin chọn chức năng thêm/xóa /sửa người sử dụng thì các dòng sự kiện phụ dưới đây được thực hiện:

Thêm người dùng:

Mỗi khi có một nhân viên mới đến làm việc tại công ty thì Admin chọn chức năng thêm mới user. Sau khi chọn thêm mới thì hệ thống sẽ hiện ra form để Admin nhập đầy đủ thông tin của người dùmg. Hệ thống kiểm tra tính hợp lệ của thông tin Admin nhập vào. Nếu hợp lệ hệ thống sẽ thêm một người dùng mới vào hệ thống.

Sửa thông tin người dùng:

Khi Admin lựa chọn người sử dụng và sửa đổi thông tin thì hệ thống sẽ hiển thị ra form đang chứa thông tin hiện tại của người dùng. Khi đó Admin chọn sửa lại những thông tin muốn sửa. Sau khi nhập xong, hệ thống sẽ kiểm tra tính đúng đắn của thông tin nhập vào. Nếu thông tin hợp lệ thì thông tin của người dùng sẽ được cập nhật trên hệ thống

Xóa người dùng: Admin lựa chọn người sử dụng cần xóa. Hệ thống chuyển thông tin người sử dụng cần xóa sang danh sách người sử dụng cũ, xóa quyền truy cập hệ thống.

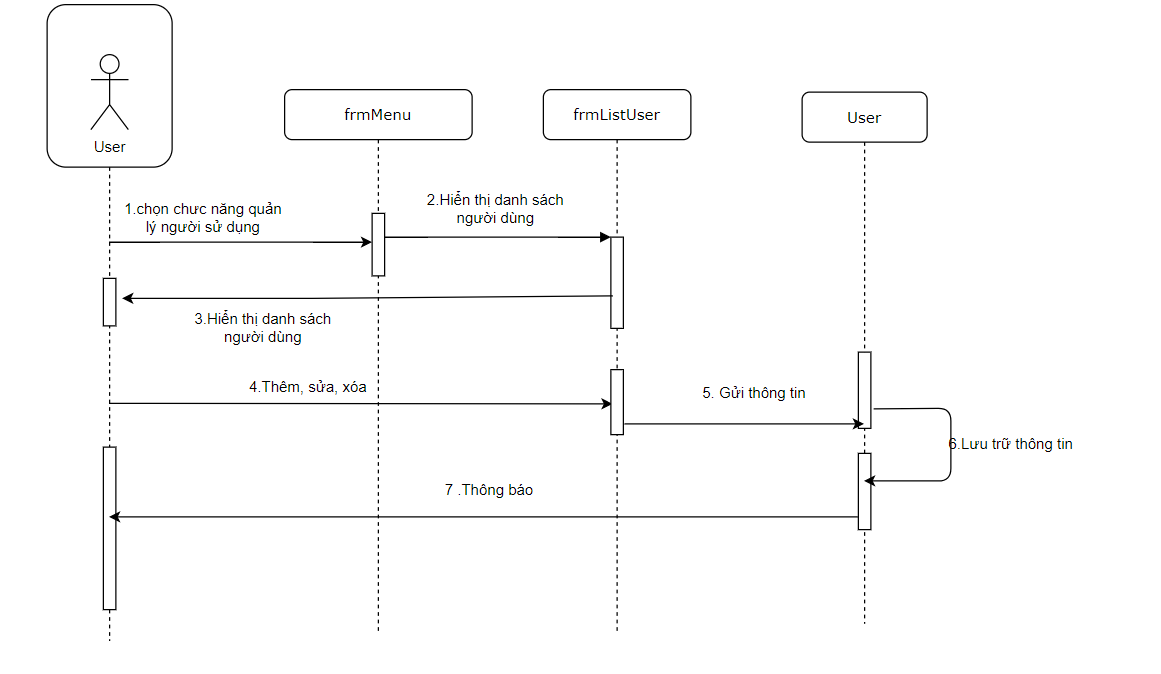
c) Điều kiện tiên quyết

Đăng nhập thành công

Role là Admin

d) Post Condition

Nếu usecase thành công, hệ thống cập nhật những thay đổi liên quan đến người dùng. Nếu usecase thất bại, hệ thống sẽ thông báo lỗi.



Bảng tuần tự quản lý User

3.4.3 Usecase “Chi tiết người dùng”

a) Tóm tắt

Usecase bắt đầu khi Admin click vào chi tiết người dùng từ màn hình quản lý người dùng hoặc member tự muốn xem thông tin cá nhân của bản thân

b) Dòng sự kiện

Sau khi User đến màn hình hiển thị thông tin chi tiết người dùng, hệ thống sẽ hiển thị các thông tin chi tiết sau:

Biểu đồ Busyrate: thể hiện mức độ bận rộn của user đó qua từng tháng bằng biểu đồ đường kẻ

Biểu đồ manmonth: biểu đồ hình cột biểu thị manmonth của user đó qua các tháng

Phase: hệ thống sẽ hiển thị thông tin phase mà user tham gia trong một khoảng thời gian nhất định

c) Điều kiện tiên quyết

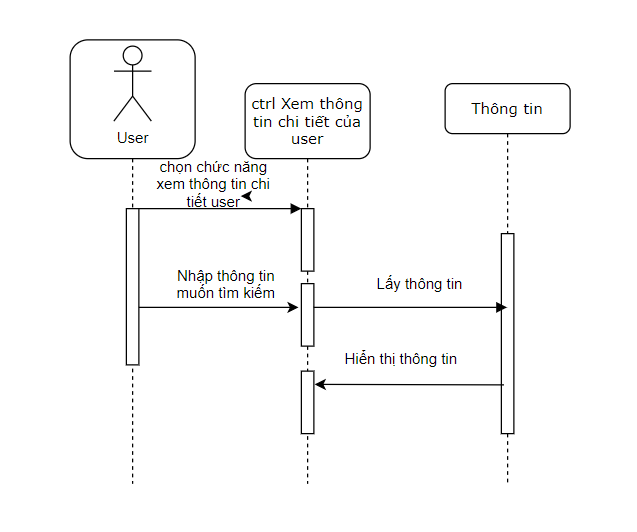
Đăng nhập thành công

Role là Admin: có thể xem thông tin toàn bộ nhân viên trong công ty

Role là member: xem thông tin của bản thân

e) Post Condition

Nếu usecase thành công, hệ thống sẽ hiển thị thông tin chi tiết của user. Nếu usecase thất bại, hệ thống sẽ thông báo lỗi.



Biểu đồ thông tin chi tiết user

3.4.3 Usecase “Quản lý Team”

a) Tóm tắt

Usecase này bắt đầu khi Admin muốn quản lý thông tin các Team trong công ty. Usecase cho phép thực hiện các thao tác thêm, sửa đổi, xóa các Team

b) Dòng sự kiện

Usecase bắt đầu khi Admin truy cập vào trang chức năng quản lý các Team trong công ty. Khi đó hệ thống sẽ trả lại toàn bộ thông tin của các Team đang hoạt động trong công ty với đầy đủ các thông tin cơ bản

Sau khi Admin chọn chức năng thêm/xóa /sửa Team thì các dòng sự kiện phụ dưới đây được thực hiện:

Thêm Team:

Khi có một Team mới được thành lập Admin chọn chức năng thêm Team và nhập các thông tin về Team đó. Hệ thống kiểm tra tính hợp lệ của thông tin Admin nhập vào. Nếu hợp lệ hệ thống sẽ thêm một Team mới vào hệ thống.

Sửa thông tin Team: Admin lựa chọn Team muốn sửa và cập nhật thông tin của Team đó. Hệ thống sẽ kiểm tra tính xác thực. Nếu hợp lệ sẽ cập nhật dữ liệu trong hệ thống

Xóa Team: Admin lựa chọn Team cần xóa. Hệ thống chuyển thông tin Team cần xóa sang trạng thái inActive.

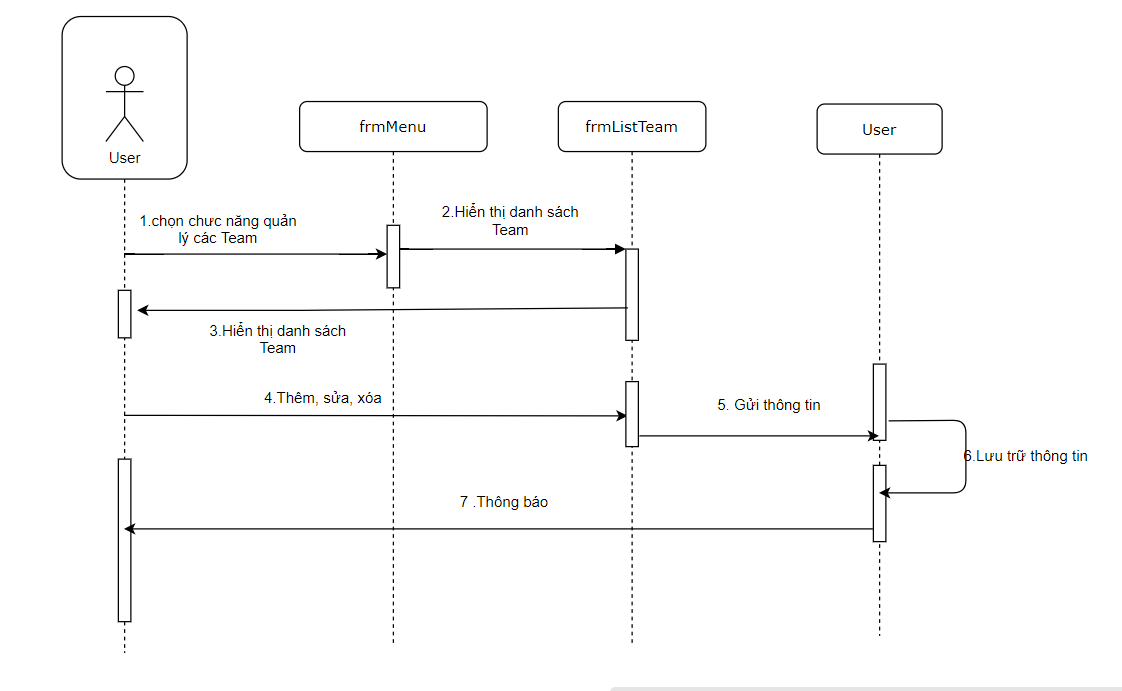
c) Điều kiện tiên quyết

Đăng nhập thành công

Role là Admin

d) Post Condition

Nếu usecase thành công, hệ thống cập nhật những thay đổi liên quan đến Team. Nếu usecase thất bại, hệ thống sẽ thông báo lỗi.



Biểu đồ tuần tự Team

3.4.4 Usecase “Quản lý Holiday”

a) Tóm tắt

Usecase này bắt đầu khi Admin muốn quản lý thông tin các ngày nghỉ trong công ty. Usecase cho phép thực hiện các thao tác thêm, sửa đổi, xóa các ngày nghỉ

b) Dòng sự kiện

Usecase bắt đầu khi Admin truy cập vào trang chức năng quản lý các ngày nghỉ của công ty. Khi đó sẽ hiển thị ra danh sách các ngày nghỉ dạng bảng hoặc dạng calendar. Trong đó đầy đủ thông tin cơ bản như kì nghỉ vì lý do gì, ngày bắt đầu và ngày kết thúc.

Sau khi Admin chọn chức năng thêm/xóa /sửa Team thì các dòng sự kiện phụ dưới đây được thực hiện:

Thêm ngày nghỉ:

Khi công ty có ngày nghỉ Admin nhập các thông tin về ngày nghỉ đó. Hệ thống kiểm tra tính hợp lệ của thông tin Admin nhập vào. Nếu hợp lệ hệ thống sẽ thêm các ngày nghỉ đó vào hệ thống.

Sửa thông tin ngày nghỉ: Admin lựa chọn ngày nghỉ muốn sửa và cập nhật thông tin của ngày nghỉ đó. Hệ thống sẽ kiểm tra tính xác thực. Nếu hợp lệ sẽ cập nhật dữ liệu trong hệ thống

Xóa ngày nghỉ: Admin lựa chọn Team cần xóa. Hệ thống sẽ xóa những thông tin cần xóa.

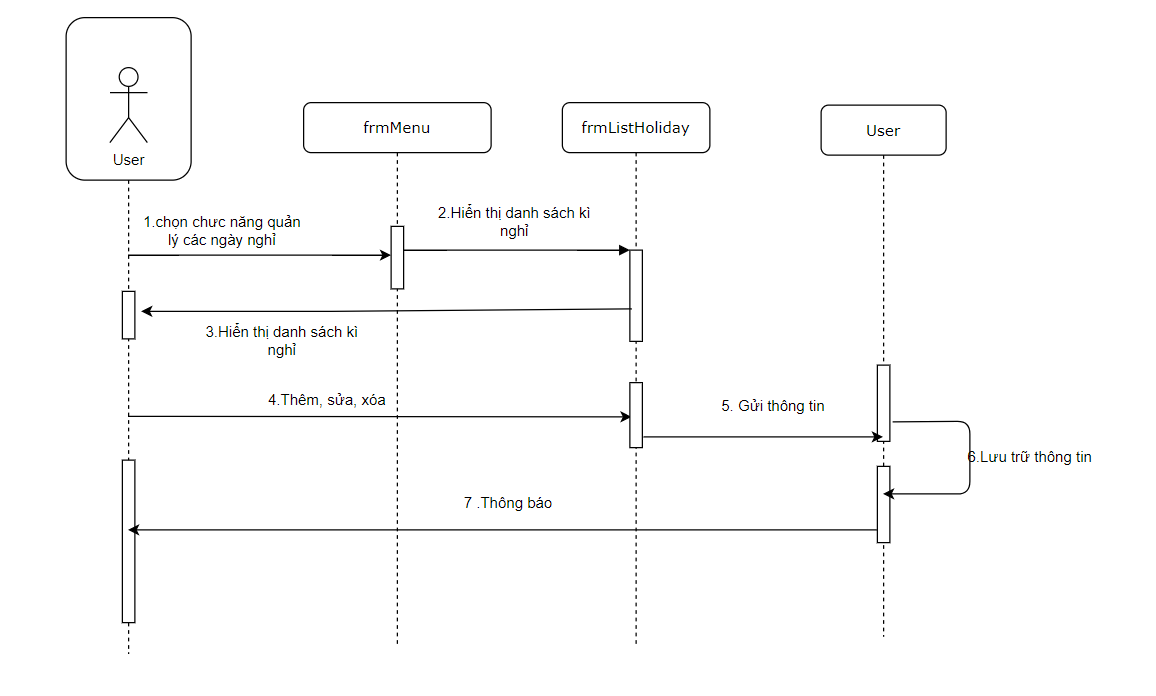
c) Điều kiện tiên quyết

Đăng nhập thành công

Role là Admin

d) Post Condition

Nếu usecase thành công, hệ thống cập nhật những thay đổi liên quan đến các ngày nghỉ. Nếu usecase thất bại, hệ thống sẽ thông báo lỗi.



Biểu đồ tuần tự ngày nghỉ

3.4.5 Usecase “Xem thông tin Alocate và Manmonth”

a) Tóm tắt

Usecase bắt đầu khi Admin chọn và chức năng xem thông tin về Alocate và Manmonth của từng Team riêng biệt hay của toàn bộ công ty. Admin có thể chọn thời gian cụ thể để xem thông tin trong khoảng thời gian đó

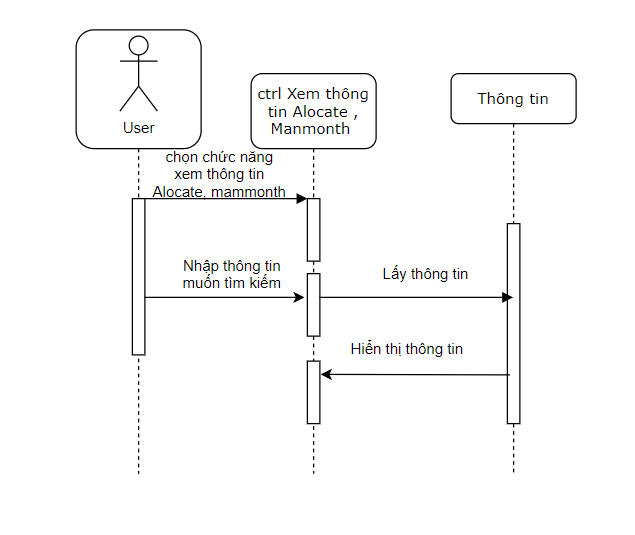
c) Điều kiện tiên quyết

Đăng nhập thành công

Role là Admin

d) Post Condition

Nếu usecase thành công, hệ thống sẽ hiển thị toàn bộ thông tin mà Admin muốn xem. Nếu thực hiện không thành công thì hệ thống sẽ báo lỗi



Biểu đồ tuần tự xem thông tin Alocate

3.4.6 Usecase “Xem thông tin Log của các dự án”

Usecase bắt đầu khi Admin chọn và chức năng xem thông tin Log của dự án. Bởi vì mỗi khi thực hiện một hành động với dự án thì hệ thống sẽ lưu hành động đó lại và khi Admin chọn chức năng xem thông tin đã được lưu thì sẽ hiện thị ra danh sách các Action đó. Admin cũng có thể lựa chọn tên của dự án để lọc ra từng action của từng dự án

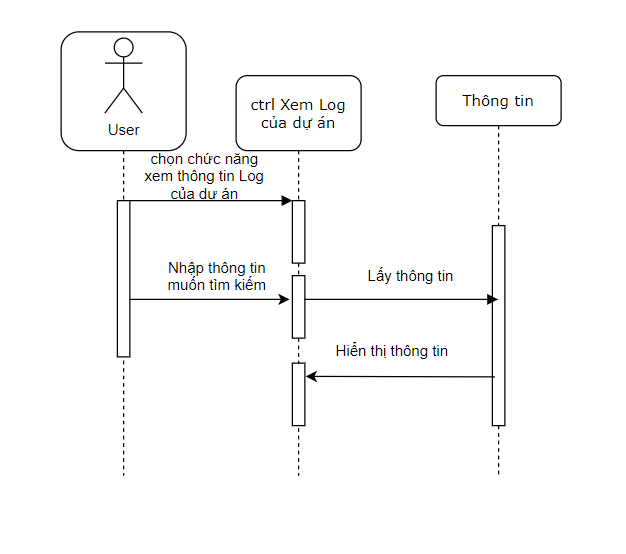
c) Điều kiện tiên quyết

Đăng nhập thành công

Role là Admin

d) Post Condition

Nếu usecase thành công, hệ thống sẽ hiển thị toàn bộ thông tin mà Admin muốn xem. Nếu thực hiện không thành công thì hệ thống sẽ báo lỗi



Biểu đồ tuần tự xem Log của dự án

3.4.7 Usecase “Quản lý dự án trong công ty”

a) Tóm tắt

Usecase này bắt đầu khi Admin muốn quản lý toàn bộ các dự án hiện đang có trong công ty. Usecase cho phép thực hiện các thao tác thêm, sửa đổi, xóa thông tin về các dự án.

b) Dòng sự kiện

Usecase bắt đầu khi Admin truy cập vào trang chức năng quản lý dự án.

Hệ thống sẽ cho hiển thị ra danh sách toàn bộ các dự hiện đang có. Kèm theo đó là một số thông tin cơ bản nhất của dự án như tên dự án, thời gian bắt đầu, thời gian kết thúc, team thực hiện và trạng thái của dự án đó.

Sau khi Admin chọn chức năng thêm/xóa /sửa dự án thì các dòng sự kiện phụ dưới đây được thực hiện:

Thêm dự án: Admin chọn chức năng tạo dự án. Khi đó sẽ có một form hiện ra và Admin cần phải nhập đầy đủ thông tin về dự án đó như: name , key, PM của dự án đó. Sau khi nhập thông tin đầy đủ. Hệ thống sẽ kiểm tra tính hợp lệ của dữ liệu đầu vào. Nếu dữ liệu hợp lệ thông tin sẽ cập nhật trong hệ thống

Sửa thông dự án: Admin lựa chọn dự án muốn thay đổi. Khi đó sẽ có form hiện ra và Admin nhập thông tin muốn thay đổi. Sau khi nhập xong, hệ thống sẽ kiểm tra tính đúng đắn của dữ liệu. Nếu dữ liệu nhập và hợp lệ thì thông tin dự án sẽ được cập nhật trên hệ thống

Xóa dự án: Admin lựa chọn dự án cần xóa sau đó chọn xóa. Hệ thống sẽ xác nhận yêu cầu. Nếu Admin chọn đồng ý thì hệ thống sẽ thực hiện xóa mềm dự án ra khỏi hệ thống

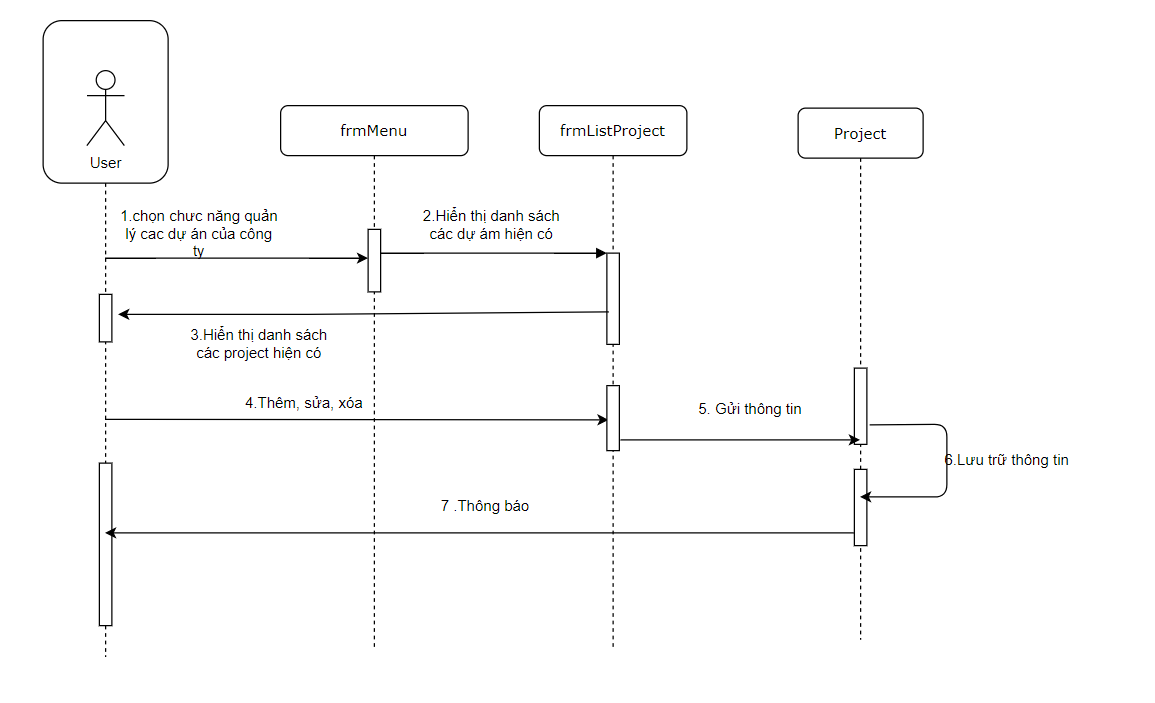
c) Điều kiện tiên quyết

Đăng nhập thành công

Role là Admin

e) Post Condition

Nếu usecase thành công, hệ thống cập nhật những thay đổi liên quan đến các dự án. Nếu usecase thất bại, hệ thống sẽ thông báo lỗi.



Biểu đồ tuần tự quản lý dự án trong công ty

3.4.8 Usecase “Tra cứu thông tin dự án”

a) Tóm tắt

Usecase bắt đầu khi người sử dụng hệ thống click vào xem chi tiết của một dự án

b) Dòng sự kiện

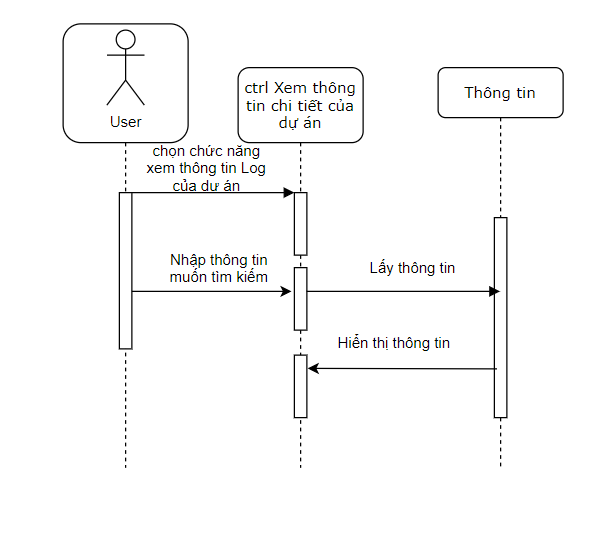
Khi người sử dụng click và một dự án thì hệ thống sẽ chuyển hướng đến màn hình thông tin chi tiết của dự án, Thông tin bao gồm:

-ProjectInfo: Hệ thống sẽ hiển thị đầy đủ thông tin về dự án mà người dùng chọn như: name, key, ngày bắt đầu, ngày kết thúc, Rank, trạng thái của dự án và team đang thực hiện dự án đó

-Phase của dự án: Một dự án sẽ chia thành các giai đoạn khác nhau, mỗi giai đoạn được gọi là một phase, Mỗi phase chứa đầy đủ thông tin như ngày bắt đầu, ngày kết thúc, buget của phase đó

-Member: Tại đây sẽ hiển thị đầy đủ danh sách các member đang được thêm vào để làm dự án cùng với thông tin của user đó như: Tên đầy đủ, email và quyền hạn của người dùng đó đối với dự án, bên cạnh đó là một biểu đồ hình tròn thể hiện manmonth của người đang có mặt trong dự án

-Rource: Tại đây sẽ hiển thị danh sách các member được thêm vào resource của dự án, bên cạnh đó là một biểu đồ hình cột thể hiện được resource của toàn bộ các thành viên trong team



Biểu đồ thông tin chi tiết của dự án

3.4.9 Usecase “Quản lý phase của dự án”

a) Tóm tắt

Usecase bắt đầu khi Admin hoặc PM của dự án thực hiện thêm dự án xong

b) Dòng sự kiện

Sau khi thêm dự án xong, Admin hoặc PM của dự án bấm vào chi tiết của dự án đó và bắt đầu tạo phase của dự án và tiến hành quản lý. Một dự án sẽ thực hiện chia ra thành nhiều phase để dễ tiến hành quản lý. Sau khi tạo phase xong sẽ hiển thị ra một danh sách các phase của dự án đó bao gồm đầy đủ các thông tin cơ bản như: tên phase, thời gian bắt đầu, thời gian kết thúc, trạng thái của phase…

Sau khi Admin chọn chức năng thêm/xóa /sửa phase của dự án thì các dòng sự kiện phụ dưới đây được thực hiện

Thêm Phase:

Usecase được thực hiện khi Admin hoặc PM của dự án tiến hành chọn chức năng thêm mới Phase. Khi đó sẽ có form hiểm thị lên màn hình để Admin tiến hành nhập đầy đủ thông tin vào. Khi chọn xong ngày bắt đầu và ngày kết thúc, hệ thống sẽ hiển thị các ô input để Admin nhập đầy đủ Buget từng tháng trong khoảng cách giữa ngày bắt đầu và ngày kết thúc. Khi đó Buget của phase là tổng Buget của tất cả các tháng cộng lại. Sau khi nhập xong hệ thống sẽ tiến hành kiểm tra tính hợp lệ của dữ liệu đầu vào. Nếu dữ liệu đúng đắn thì một phase mới sẽ được cập nhật vào hệ thống.

Cập nhật Phase:

Usecase được thực hiện khi Admin hoặc PM của dự án tiến hành cập nhật lại thông tin của phase. Khi đó sẽ có một form hiển thị ra và Admin cần nhập lại các thông tin muốn cập nhật sau đó hệ thống sẽ tiến hành kiểm tra dữ liệu. Nếu dữ liệu đầu vào là hợp lệ phase sẽ được hệ thống cập nhật

Sau khi hoàn thành dự án, Admin cần cập nhật lại các thông số KPI của dự án như: điểm CSS, timeline, leakage….

c) Điều kiện tiên quyết

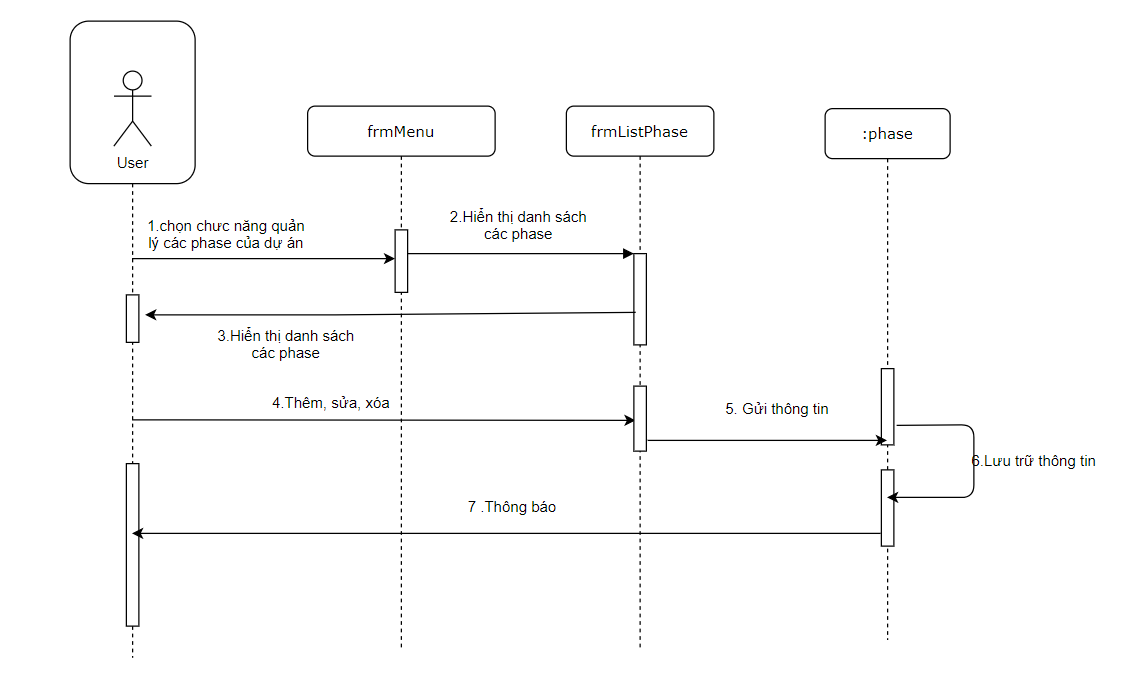
Đăng nhập thành công

Role là Admin hoặc là PM của dự án đó: xem thông tin phase, thêm phase và cập nhật phase

Role là member: xem thông tin của phase

d) Post Condition

Nếu usecase thành công, hệ thống cập nhật những thay đổi liên quan đến phase của dự án. Nếu usecase thất bại, hệ thống sẽ thông báo lỗi.



Biểu đồ tuần tự phase của dự án

3.4.10 Usecase “Cập nhật thành viên của dự án”

a) Tóm tắt

Usecase bắt đầu khi Admin hoặc PM của dự án muốn thêm thành viên hoặc cập nhật thành viên trong dự án. Hỗ trợ chức năng hiển thị vai trò của thành viên trong dự án và hiển thị danh sách nhân viên tham gia dự án

b) Dòng sự kiện

Khi Admin click vào chi tiết của dự án, hệ thống sẽ hiển thị danh sách các thành viên đang tham gia vào dự án với đầy đủ các thông tin cơ bản như: tên, email và vai trò của member đó trong dự án. Khi Admin chọn chức năng thêm thành viên hệ thống sẽ hiển thị một form để Admin nhập thông tin, Admin chọn member từ danh sách các member trong công ty và chỉ định vai trò của người đó trong dự án. Khi nhập đầy đủ thông tin và nhấn lưu, hệ thống sẽ kiểm tra tính đúng đắn của dữ liệu đầu vào. Nếu dữ liệu hợp lệ, hệ thống sẽ tiến hành cập nhật thành viên đó tham gia vào dự án

Khi muốn cập nhật lại vai trò của thành viên, Admin chọn chỉnh sửa và chọn lại vai trò sau đó nhấn lưu, hệ thống sẽ kiểm tra dữ liệu và cập nhật lại danh sách các member của dự án

Khi muốn xóa member ra khỏi dự án, Admin chỉ cần cập nhật lại trạng thái của member thành Inactive, hệ thống sẽ xóa thành viên đó ra khỏi dự án

c) Điều kiện tiên quyết

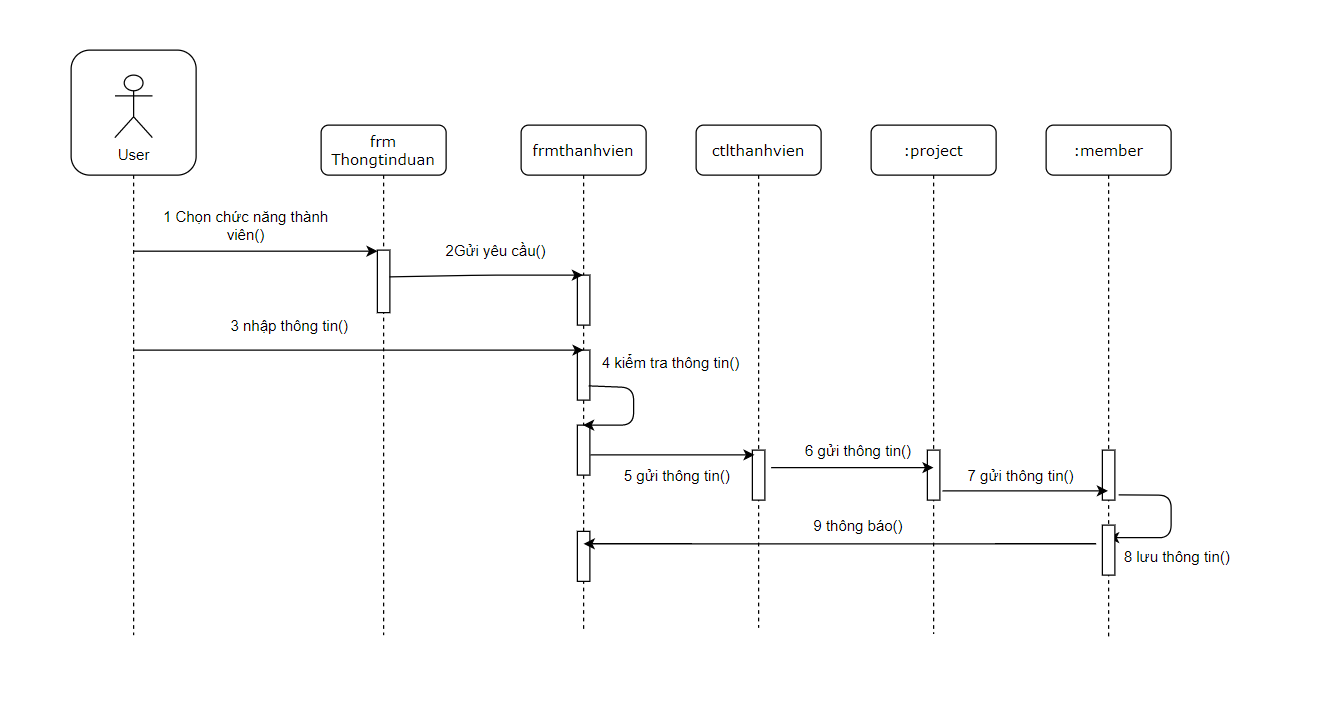
Đăng nhập thành công

Role là Admin hoặc là PM của dự án: xem danh sách các thành viên trong dự án, cập nhật thành viên dự án

Role là member: Xem danh sách các thành viên trong dự án

d) Post Condition

Nếu usecase thành công, hệ thống cập nhật những thay đổi liên quan đến member của dự án. Nếu usecase thất bại, hệ thống sẽ thông báo lỗi.



Biểu đồ trình tự thêm member vào dự án

3.4.11 Usecase “Thêm resource cho dự án”

a) Tóm tắt

Usecase bắt đầu sau khi Admin thêm member vào dự án và muốn thêm resource vào dự án cho member đó

b) Dòng sự kiện

Khi Admin click vào chi tiết của dự án, hệ thống sẽ hiển thị danh sách các resource đang có của dự án với đầy đủ các thông tin cơ bản như: tên, email và vai trò của member đó trong dự án và thời gian bắt đầu và thời gian kết thúc resource đó cùng với Alocate. Khi Admin chọn chức năng thêm resource hệ thống sẽ hiển thị một form để Admin nhập thông tin, Admin chọn member từ danh sách các member đang có trong dự án và chỉ định vai trò của người đó trong resource này, sau đó nhập thời gian bắt đầu và thời gian kết thúc và alocate của member trong giai đoạn đó. Khi nhập đầy đủ thông tin và nhấn lưu, hệ thống sẽ kiểm tra tính đúng đắn của dữ liệu đầu vào. Nếu dữ liệu hợp lệ, hệ thống sẽ tiến hành cập nhật resource đó vào dự án

Khi muốn cập nhật lại thông tin resource, Admin chọn chỉnh sửa và nhập lại thông tin sau đó nhấn lưu, hệ thống sẽ kiểm tra dữ liệu và cập nhật lại danh sách các resource của dự án

Khi muốn xóa resource ra khỏi dự án, Admin chỉ cần chọn hành động xóa, hệ thống sẽ xóa thành viên đó ra khỏi dự án

c) Điều kiện tiên quyết

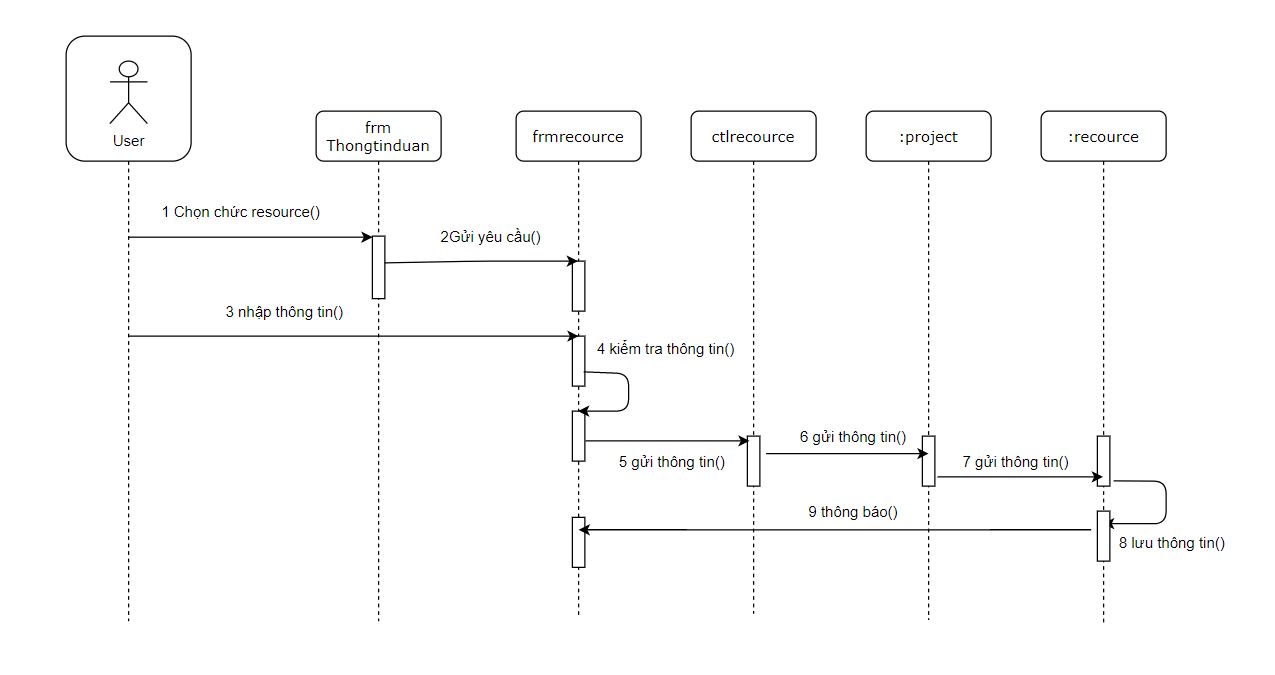
Đăng nhập thành công

Role là Admin hoặc là PM của dự án: xem thông tin resource của dự án và cập nhật resource của dự án

Role là member: xem thông tin resource của dự án

d) Post Condition

Nếu usecase thành công, hệ thống cập nhật những thay đổi liên quan đến resource của dự án. Nếu usecase thất bại, hệ thống sẽ thông báo lỗi.

 Biểu đồ Thêm resource cho dự án

## Chương 4: Kiến trúc và thiết kế

## 4.1. Kiến trúc hệ thống

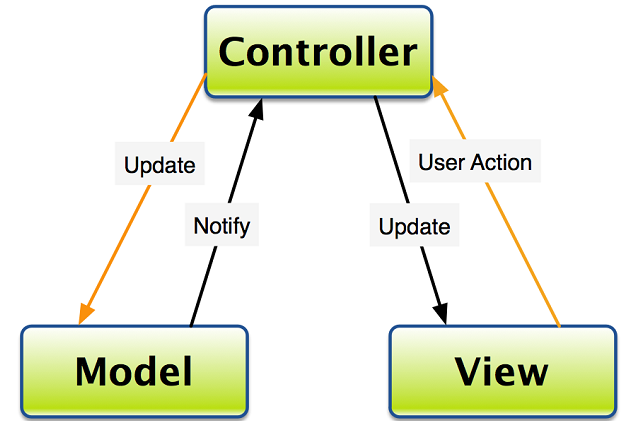
MVC là chữ viết tắt của Model - View - Controller, đây là một mô hình kiến phần mềm được tạo ra với mục đích quản lý và xây dựng dự án phần mềm có hệ thống hơn. Mô hình này được dùng khá rộng rãi và đặc biệt là trong các ngôn ngữ lập trình web. Trong PHP hiện tại có khá nhiều Framework và tất cả đều xây dựng từ mô hình MVC.

Trong mô hình này thì:

Model: có nhiệm vụ thao tác với cơ sở dữ liệu, nghĩa là nó sẽ chứa tất cả các hàm, các phương thức truy vấn trực tiếp với dữ liệu và controller sẽ thông qua các hàm, phương thức đó để lấy dữ liệu rồi gửi qua View.

View: có nhiệm vụ tiếp nhận dữ liệu từ controller và hiển thị nội dung sang các đoạn mã HTML, bạn có thể hiểu nôm na đây người ta còn gọi là thành phần giao diện.

Controller: đóng vài trò trung gian giữa Model và View. Nó có nhiệm vụ tiếp nhận yêu cầu từ client sau đó xử lý request, load model tương ứng và gửi data qua view tương ứng rồi trả kết quả về cho client.



#### Hình 26: Mô hình MVC

**Cách thức hoạt động của mô hình MVC**: Khi một máy client tương tác với view (bằng các trình duyệt trên PC hay Mobile). Gửi một yêu cầu đến phía server. Controller sẽ tiếp nhận yêu cầu, tiến hành xử lý yêu cầu đó, nếu thông tin liên quan đến CSDL thì Controller gọi đến Model để lấy dữ liệu. Sau đó trả kết quả sau khi xử lý cho View. View sẽ hiển thị thông tin dưới dạng các thẻ HTML cho người dùng.

Ưu điểm và nhược điểm của mô hình MVC:

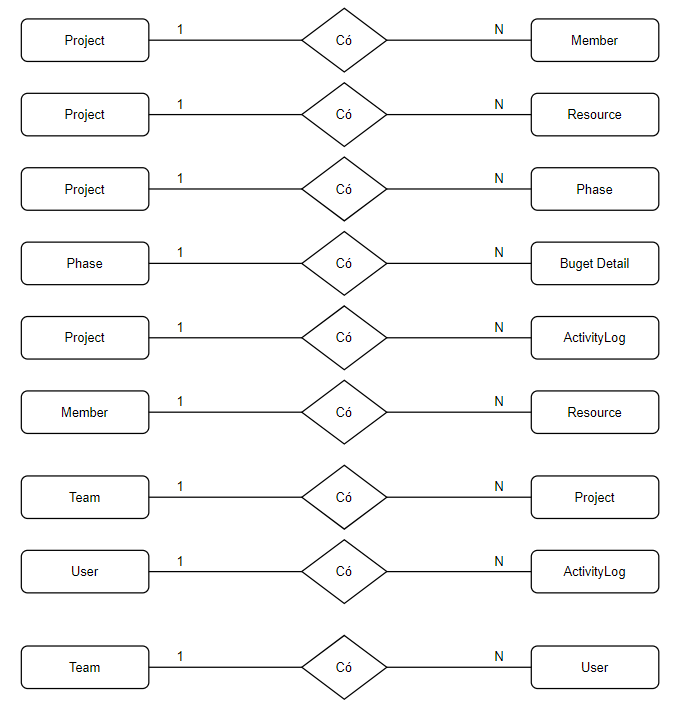
Ưu điểm:

* Hệ thống phân ra từng phần nên dễ dáng phát triển.
* Chia thành nhiều modun nhỏ nên nhiều người có thể làm chung dự án.
* Vấn đề bảo trì cũng tương đối ok, dễ nâng cấp.
* Dễ dàng debug trong quá trình xây dựng.

Nhược điểm:

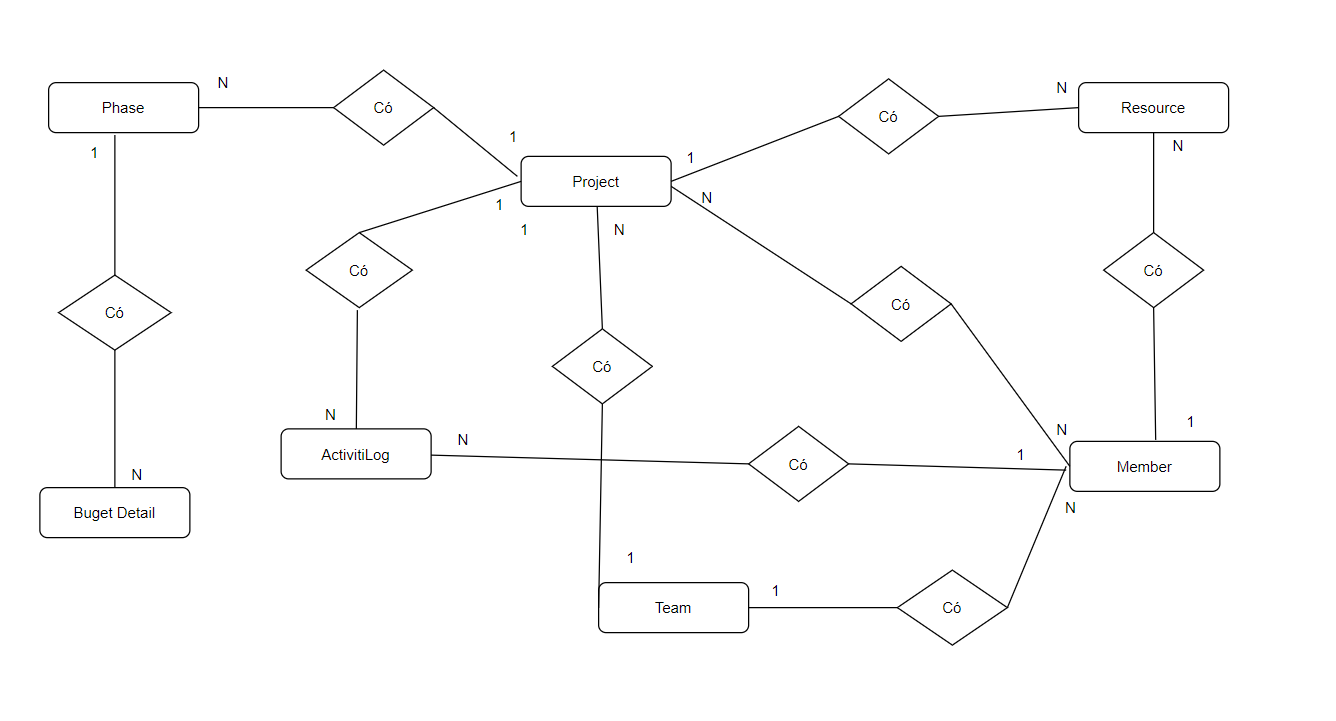
* Hệ thống sẽ chạy chậm hơn PHP thuần, tuy nhiên nó ko phải là vấn đề.
* Xây dựng cầu kì và mất thời gian để xây dựng thư viện, cấu trúc.

4.2 Xác định liên kết giữa các thực thể



Mô tả liên kết giữa các thực thể

Mô hình thực thể liên kết (E-R)



Hình: Biểu đồ liên kết thực thể

4.3 Chuyển mô hình ERD thành mô hình quan hệ dữ liệu

Với mỗi thực thể thông thường tạo một quan hệ tương ứng có các thuộc tính chính là các thuộc tính đơn của thực thể, đối với các thuộc tính phức hợp ta tách thành các thuộc tính đơn, khóa của quan hệ chính là khóa của thực thể. Ta thu được các quan hệ dữ liệu:

|  |
| --- |
| - PROJECT (id, name, key, from\_at, end\_at, buget, status, css, leakage, timeliness, created\_at, update\_at)  -RECOURCE (id, from\_at, to\_at, allocation, roleProject)  -MEMBER (id, roleProject, is\_member)  -ACTIVITILOG (id, content, type)  -USER (id, fullname, email, password, roleSystem)  -TEAM (id, title)  -PHASE (id, title, from\_at, to\_at, status, buget, note, css, leakage, timeliness)  -BUGET\_DETAIL: (id, month, buget)  -HOLIDAY: (id, title,from\_at, to\_at) |

4.4 Xử lý mô hình quản lý dữ liệu

Xử lý thực thể yếu: Trong bài toán đang xét không có thực thể yếu

Xử lý thuộc tính đa trị: Trong bài toán đang xét không có thuộc tính đa trị

Xử lý mối quan hệ 1-1: Trong bài toán đang xét không có mối quan hệ 1-1

Xử lý mối quan hệ 1-N: Bổ sung thêm khóa của quan hệ 1 làm thuộc tính quan hệ N.

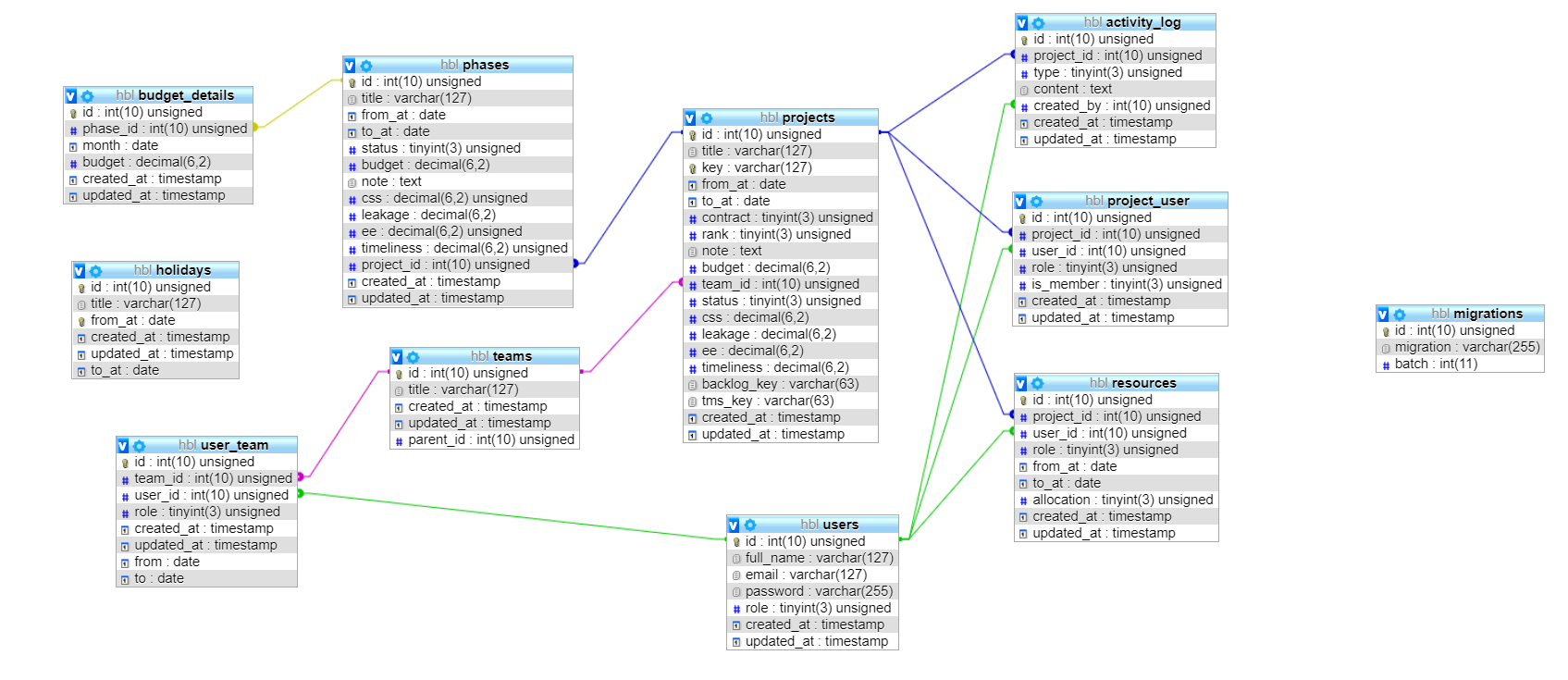
Xử lý mối quan hệ N-N: Tách thành 2 quan hệ 1-N (Tạo bảng MEMBER).

Chuẩn 1NF: Giải quyết 2 vấn đề chính là không chứa nhóm thuộc tính lặp và mỗi hàng có một thuộc tính khóa nhận dạng duy nhất. Xét mô hình quan hệ dữ liệu hiện tại, mọi quan hệ đều chỉ có một khóa và không chứa thuộc tính đa trị => Thỏa mãn 1 NF.

Chuẩn 2NF: Mỗi thuộc tính không khóa phải phụ thuộc đầy đủ vào khóa chính. Xét mô hình quan hệ dữ liệu hiện tại, không có quan hệ nào có 2 thuộc tính là khóa, tức là không có phụ thuộc hàm không đầy đủ vào khóa chính. => Thỏa mãn 2NF.

Chuẩn 3NF: Tất cả các thuộc tính không khóa được yêu cầu là phải phụ thuộc trực tiếp vào mỗi khóa của quan hệ. Xét mô hình quan hệ dữ liệu hiện tại, mọi quan hệ đều thỏa mãn tất cả các thuộc tính không khóa đều phụ thuộc trực tiếp vào mỗi khóa. => Thỏa mãn 3NF

4.5 Triển khai cơ sở dữ liệu trên MySql



Biểu đồ quan hệ

4.6 Mô tả cơ sở dữ liệu

4.6.1 Bảng User (Người dùng)

|  |  |  |
| --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Mô tả |
| Id | int | Khóa chính |
| Fullname | varchar | Tên đầy đủ của User |
| Email | varchar | Email của người dùng do công ty cung cấp |
| Password | varchar | Mật khẩu của người dùng |
| Role | varchar | Quyền hạn trong hệ thống |
| Created\_at | timestamp | Ngày tạo tài khoản |
| Updated\_at | timestamp | Ngày update tài khoản |

Bảng User

4.6.2 Bảng Team

|  |  |  |
| --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Mô tả |
| Id | int | Khóa chính |
| Title | varchar | Tên của Team |
| Pareant\_ID | varchar | Level của Team |
| Created\_at | timestamp | Ngày tạo Team |
| Updated\_at | timestamp | Ngày update Team |

Bảng Team

4.6.3 Bảng UserTeam

|  |  |  |
| --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Mô tả |
| Id | int | Khóa chính |
| Team\_ID | int | Là khó ngoại lấy từ bảng Team |
| User\_ID | int | Là khóa ngoại lấy từ bảng User |
| Role | int | Quyền hạn trong Team |
| From | varchar | Ngày bắt đầu join vào team |
| To | varchar | Ngày exit khỏi Team |
| Created\_at | timestamp | Ngày tạo User-team |
| Updated\_at | timestamp | Ngày update User-team |

Bảng User-Team

4.6.4 Bảng Holiday (Danh sách các ngày nghỉ)

|  |  |  |
| --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Mô tả |
| Id | int | Khóa chính |
| Title | varchar | Tên của kì nghỉ |
| From\_at | varchar | Thời gian kì nghỉ bắt đầu |
| To\_at | varchar | Thời gian kì nghỉ kết thúc |
| Created\_at | timestamp | Ngày tạo kì nghỉ |
| Updated\_at | timestamp | Ngày update kì nghỉ |

Bảng Holiday

4.6.5 Bảng Project (Dự án)

|  |  |  |
| --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Mô tả |
| Id | int | Khóa chính |
| Title | varchar | Tên của Project |
| Key | varchar | Key của dự án |
| From\_at | varchar | Thời gian bắt đầu dự án |
| To\_ | varchar | Thời gian kết thúc dự án |
| Rank | Int | Đánh giá độ lớn của dự án |
| Note | varchar | Ghi chú về dự án |
| Buget | Int | Tổng số Manmonth |
| Team\_id | Int | Khóa ngoại lấy từ bảng Team |
| Status | Int | Trạng thái của dự án |
| Css | Int | Điểm KPI đánh giá hiệu suất công việc |
| Leakage | Int | Điểm KPI số Bug có sau khi đã bàn giao sản phẩm cho khách hàng |
| Ee | Int | Điểm KPI đánh giá hiệu quả sử dụng nguồn lực trong dự án |
| Timeliness | Int | Điểm KPI đánh giá các mốc bàn giao cho khách hàng có chuẩn hay không |
| Backlog\_key | Int | Key của dự án trên BackLog |
| Tms\_key | Int | Key của dự án trên TMS |
| Created\_at | Timestamp | Ngày tạo dự án |
| Update\_at | Timestamp | Ngày Update dự án |

Bảng Project

4.6.6 Bảng Project-User (Member trong dự án)

|  |  |  |
| --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Mô tả |
| Id | int | Khóa chính |
| Project\_ID | int | Khóa ngoại lấy từ bảng Project |
| User\_ID | Int | Khóa ngoại lấy từ bảng User |
| Role | Int | Quyền hạn trong project |
| Is\_Member | Int | Trạng thái của người dùng trong Project |
| Created\_at | timestamp | Ngày tạo |
| Updated\_at | timestamp | Ngày update |

Bảng Project-User

4.6.7 Bảng Recource (Nguồn lực của dự án)

|  |  |  |
| --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Mô tả |
| Id | int | Khóa chính |
| Project\_ID | int | Khóa ngoại lấy từ bảng Project |
| User\_ID | Int | Khóa ngoại lấy từ bảng User |
| Role | Int | Quyền hạn trong project |
| From\_at | Varchar | Thời gian bắt đầu |
| To\_at | Varchar | Thời gian kết thúc |
| Allocate | Int | Nguồn lực tham gian bao nhiêu % time |
| Created\_at | timestamp | Ngày tạo |
| Updated\_at | timestamp | Ngày update |

Bảng Recources

4.6.8 Bảng Phase

|  |  |  |
| --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Mô tả |
| Id | int | Khóa chính |
| Title | varchar | Tên của Phase |
| Project\_ID | int | Khóa ngoại lấy từ bảng Project |
| From\_at | varchar | Thời gian bắt đầu phase |
| To\_at | varchar | Thời gian kết thúc phase |
| Note | varchar | Ghi chú về phase |
| Buget | Int | Tổng số Manmonth |
| Status | Int | Trạng thái của phase |
| Css | Int | Điểm KPI đánh giá hiệu suất công việc |
| Leakage | Int | Điểm KPI số Bug có sau khi đã bàn giao sản phẩm cho khách hàng |
| Ee | Int | Điểm KPI đánh giá hiệu quả sử dụng nguồn lực trong dự án |
| Timeliness | Int | Điểm KPI đánh giá các mốc bàn giao cho khách hàng có chuẩn hay không |
| Created\_at | Timestamp | Ngày tạo phase |
| Update\_at | Timestamp | Ngày Update phase |

Bảng Phase

4.6.9 Bảng Buget\_Detail

|  |  |  |
| --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Mô tả |
| Id | int | Khóa chính |
| Phase\_ID | int | Khóa ngoại lấy từ bảng Phase |
| Month | Varchar | Tháng (Mỗi tháng sẽ chứa 1 buget khác nhau) |
| Buget | Int | Nguồn lực của dự án |
| Created\_at | timestamp | Ngày tạo |
| Updated\_at | timestamp | Ngày update |

Bảng Buget\_Detail

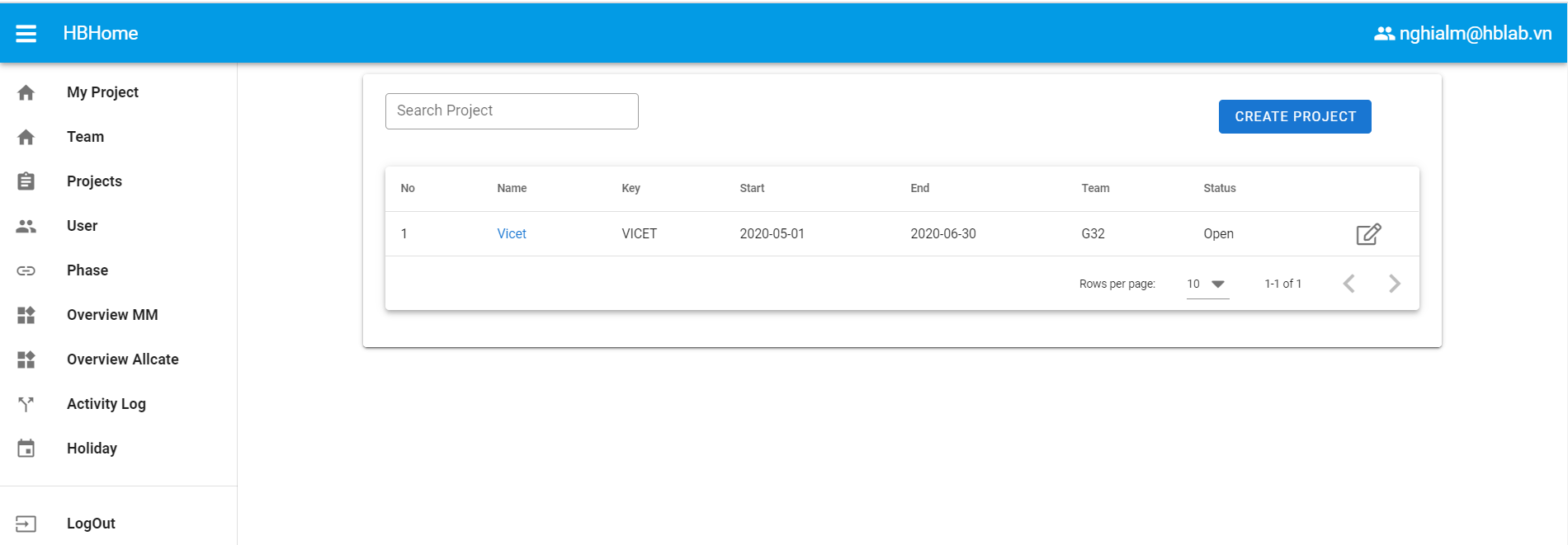
4.6.10 Bảng ActivityLog

|  |  |  |
| --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Mô tả |
| Id | int | Khóa chính |
| Project\_ID | int | Khóa ngoại lấy từ bảng Project |
| User\_ID | Int | Khóa ngoại lấy từ bảng User |
| Type | Int | Kiểu : insert, update, delete … |
| Content | Varchar | Nội dung của Log |
| Created\_at | timestamp | Ngày tạo |
| Updated\_at | timestamp | Ngày update |

Bảng Project-User

Chương 5 Thiết kế giao diện

5.1 Project



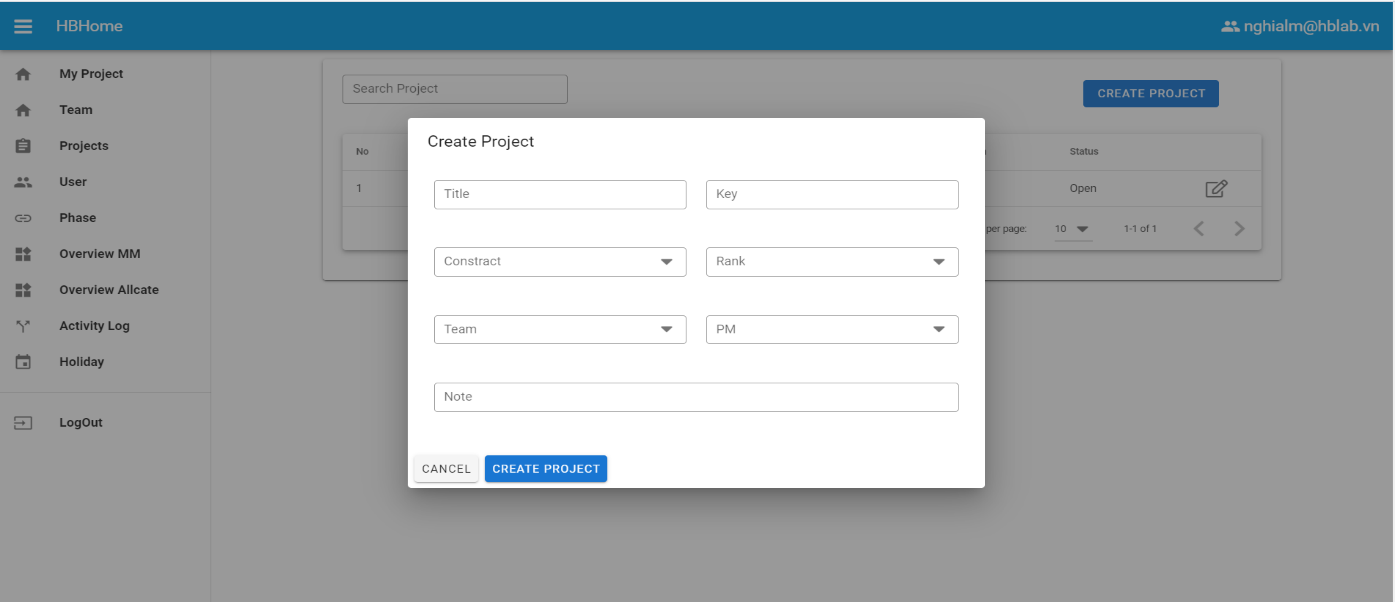
Giải thích:

-Sau khi đăng nhập thành công thì hệ thống sẽ chuyển đến trang danh sách các Project hiện có trong công ty

-Tại đây người dùng có thể thể thực hiện tìm kiếm theo tên bằng cách nhập vào ô tìm kiếm

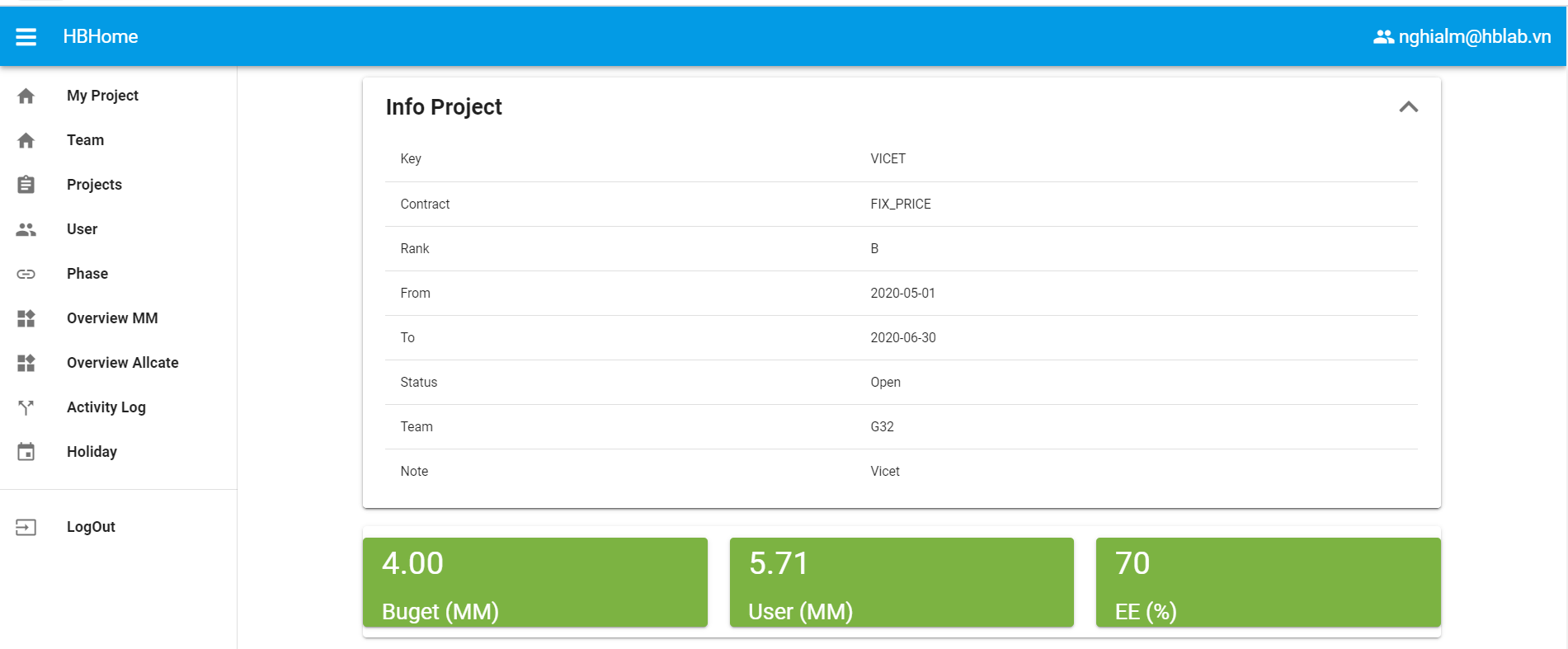
- Action:

+ Người dùng có thể thực hiện thêm mới Project bằng cách nhấn Button Project

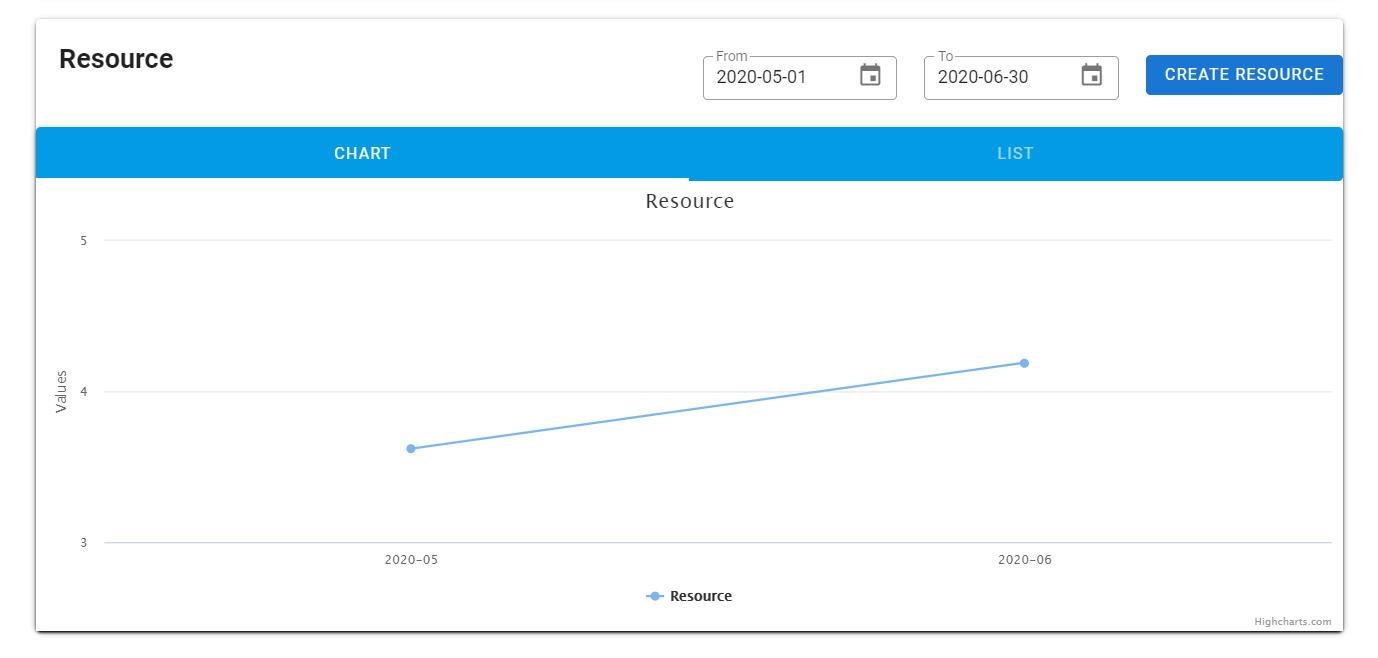


+ Khi người dùng nhấn thêm mới sẽ hiện ra Form để nhập thông tin. Thông tin sẽ được validate và gửi lên server.

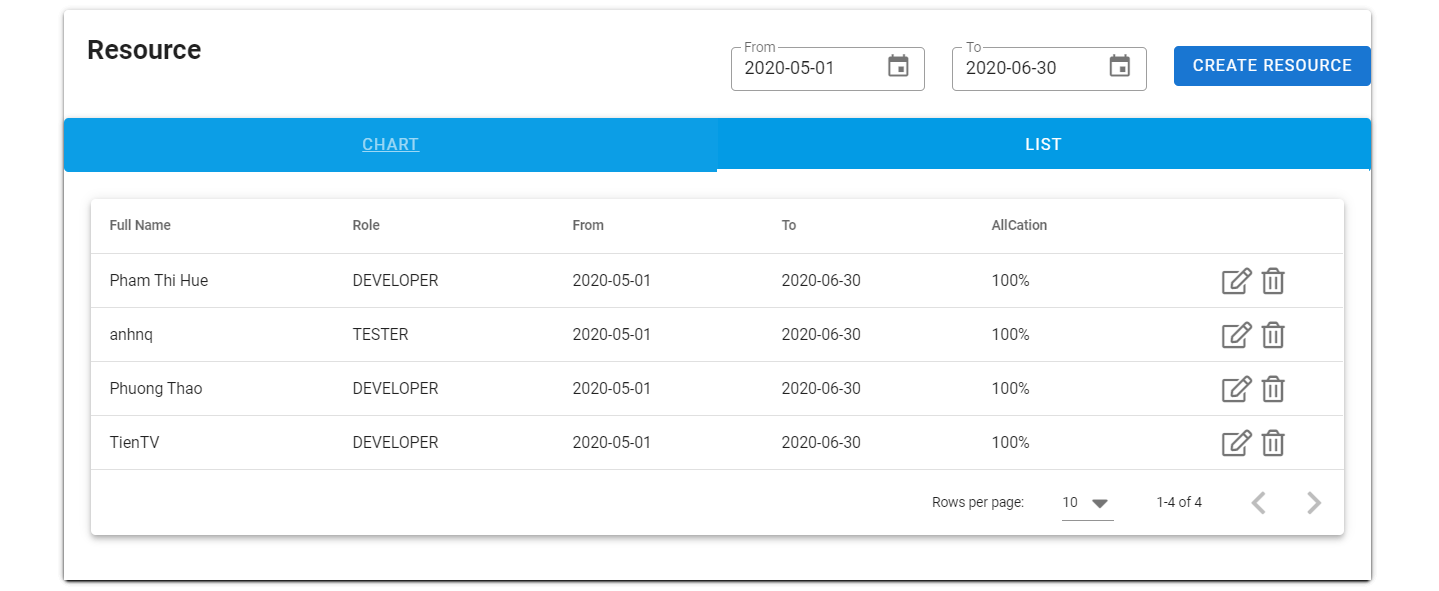
+Khi nhấn vào Chi Tiết Project



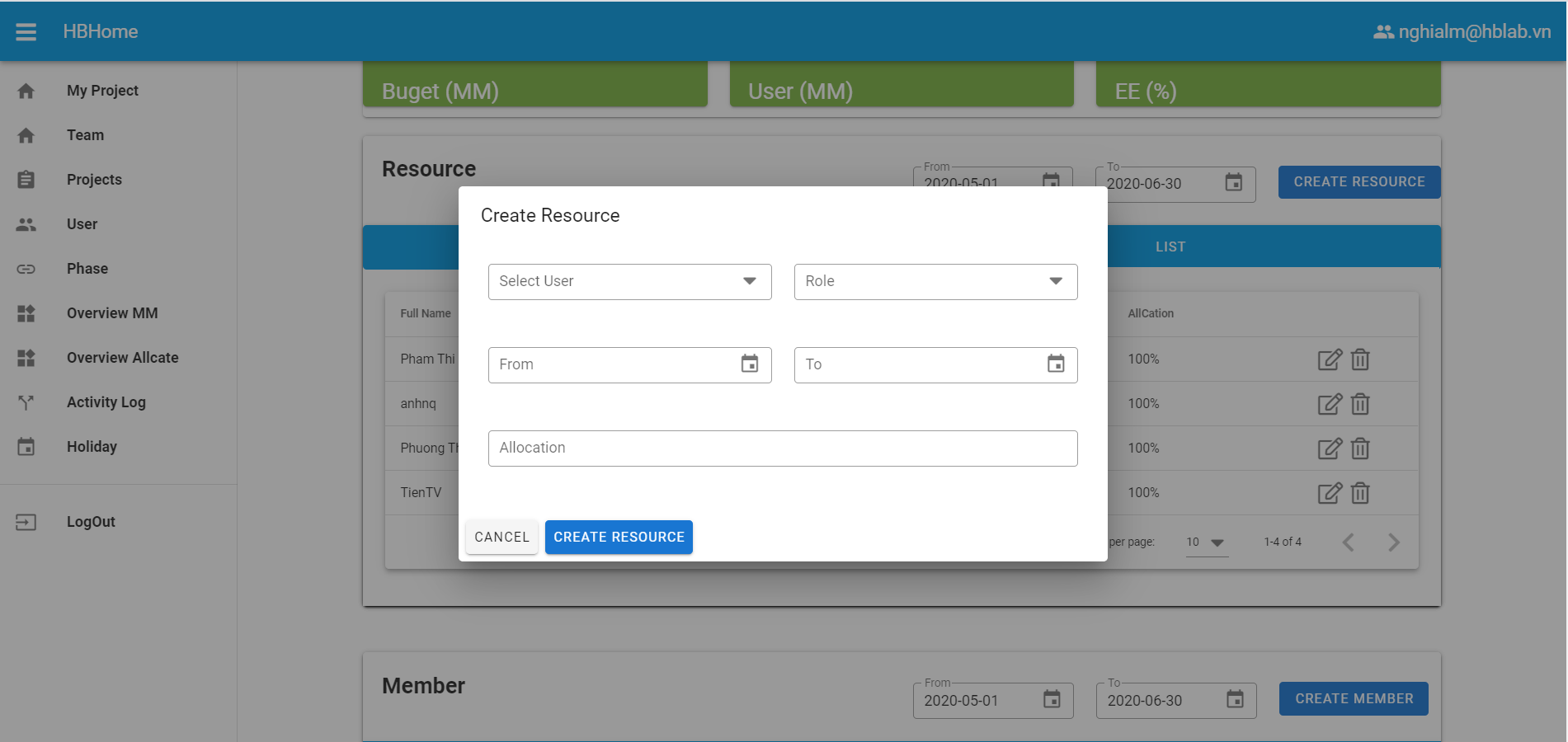
Sẽ hiển thị thông chi chi tiết của Project



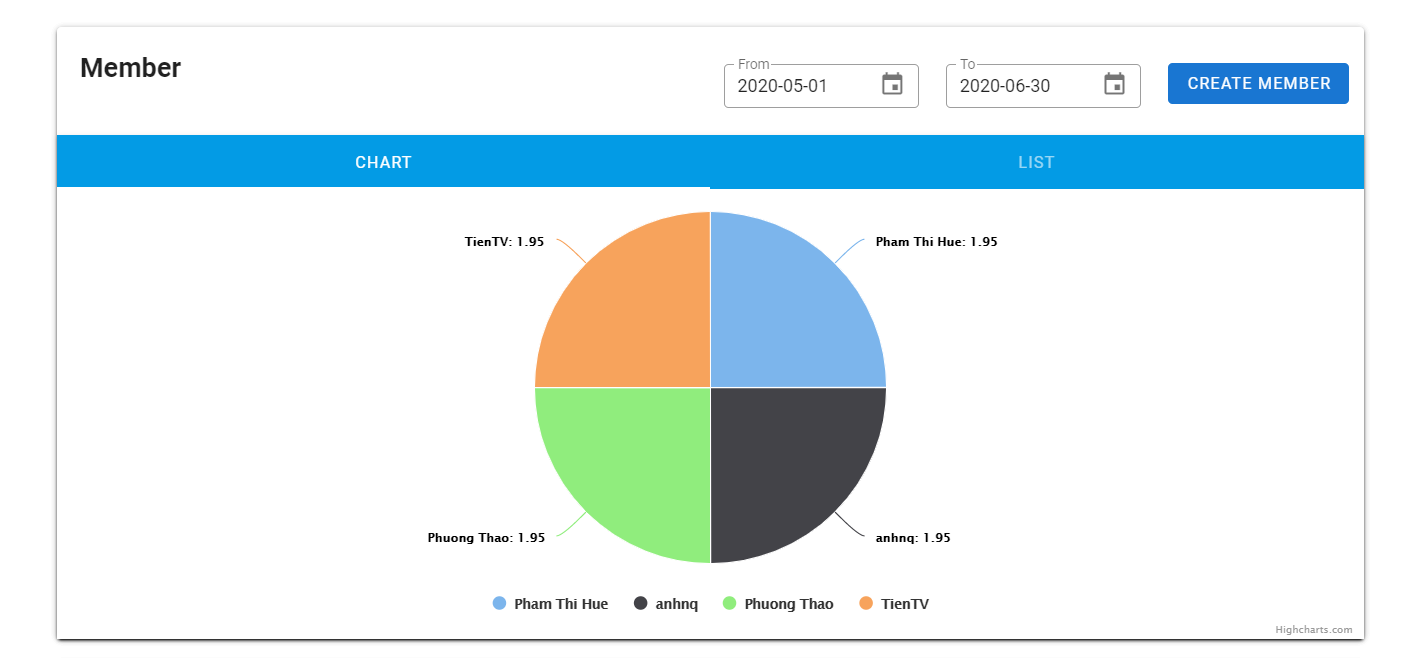
Thông tin resource của Project dạng Biểu đồ



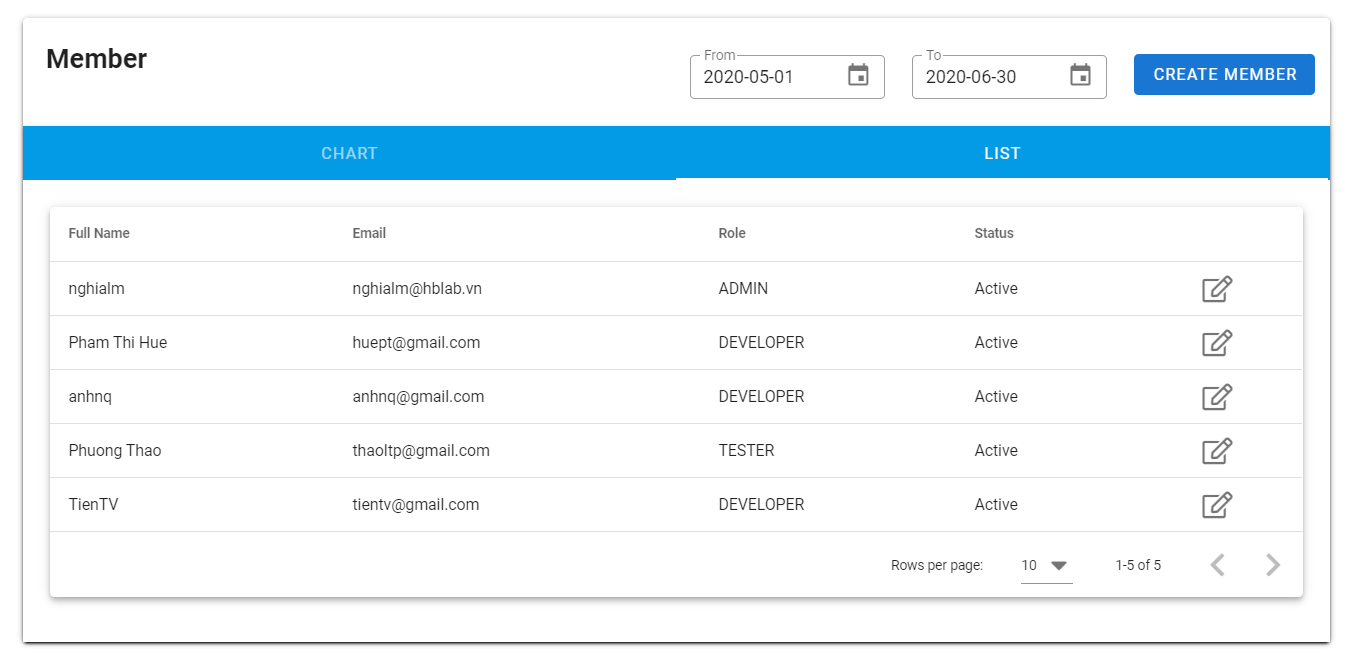
List Resource dạng bảng



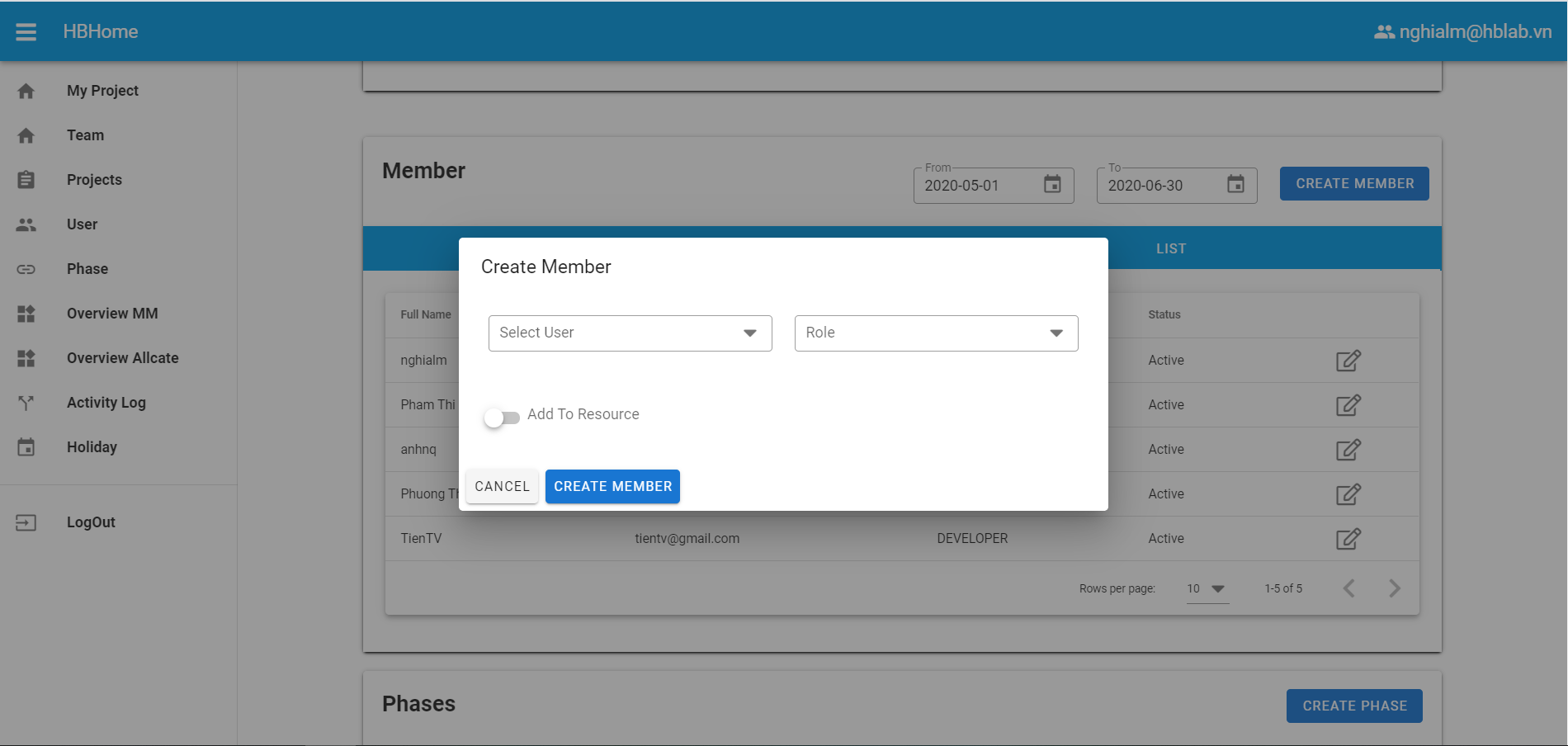
Tạo Resource cho dự án



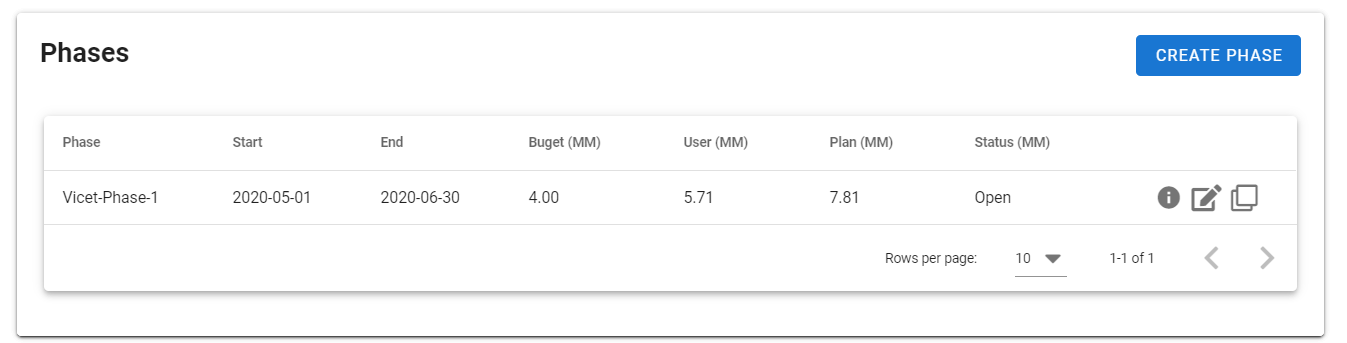
Thông tin manmonth Member Project dạng biểu đồ tròn



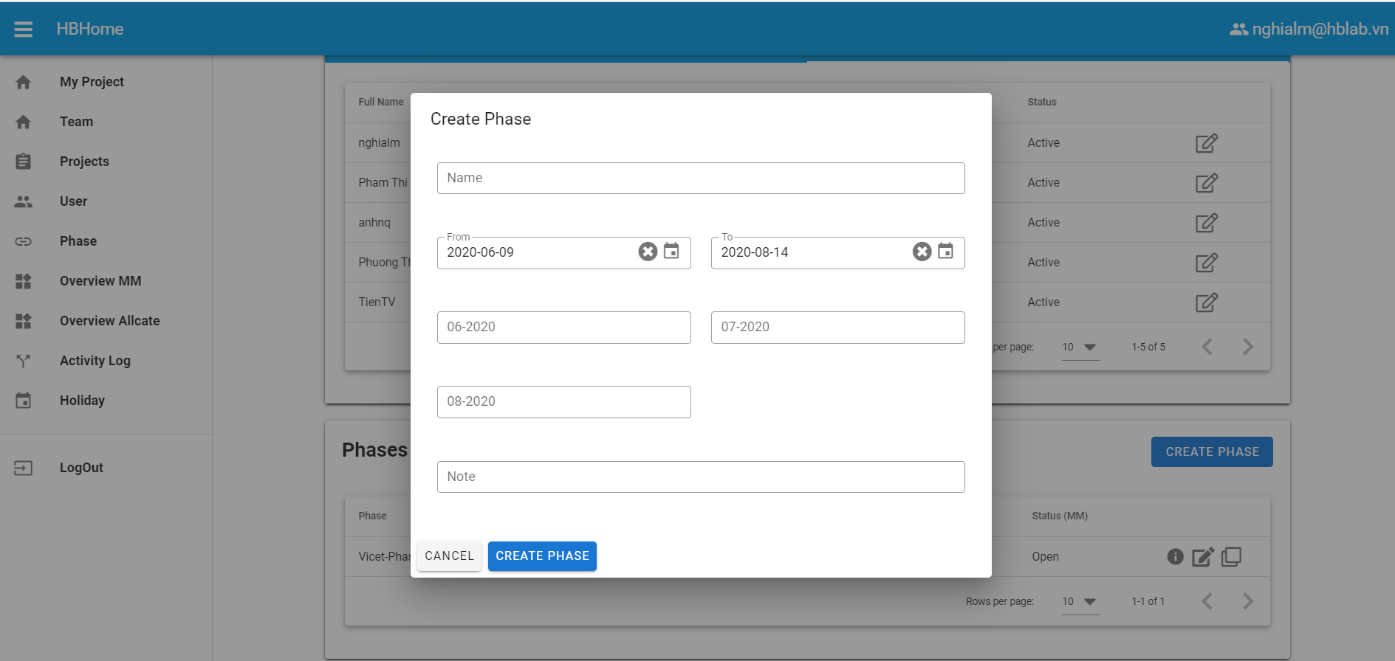
Thông tin Member của dự án



Thêm mới thành viên vào dự án

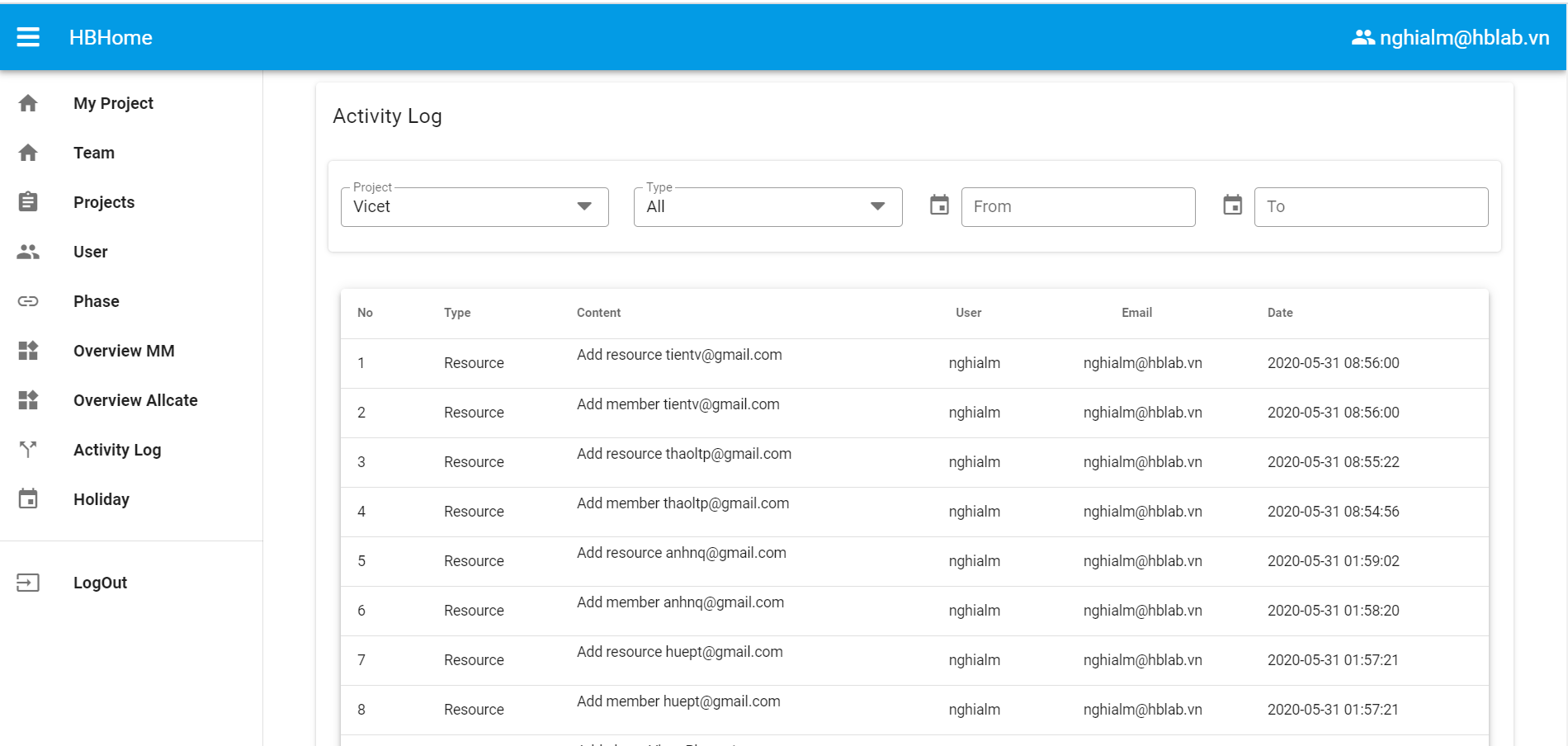


Danh sách các phase của dự án



Thêm phase cho dự án

5.2 Activity

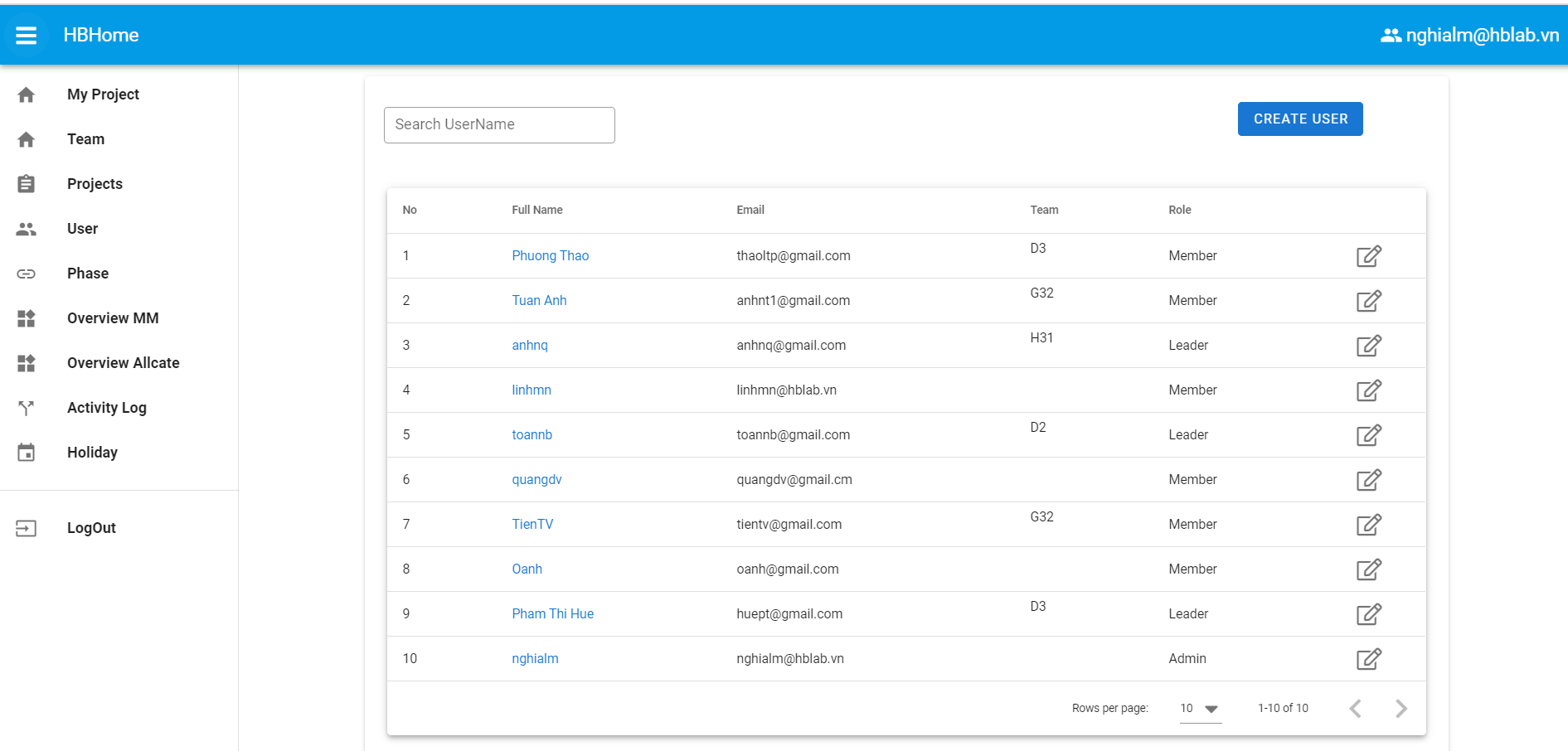


Giải thích:

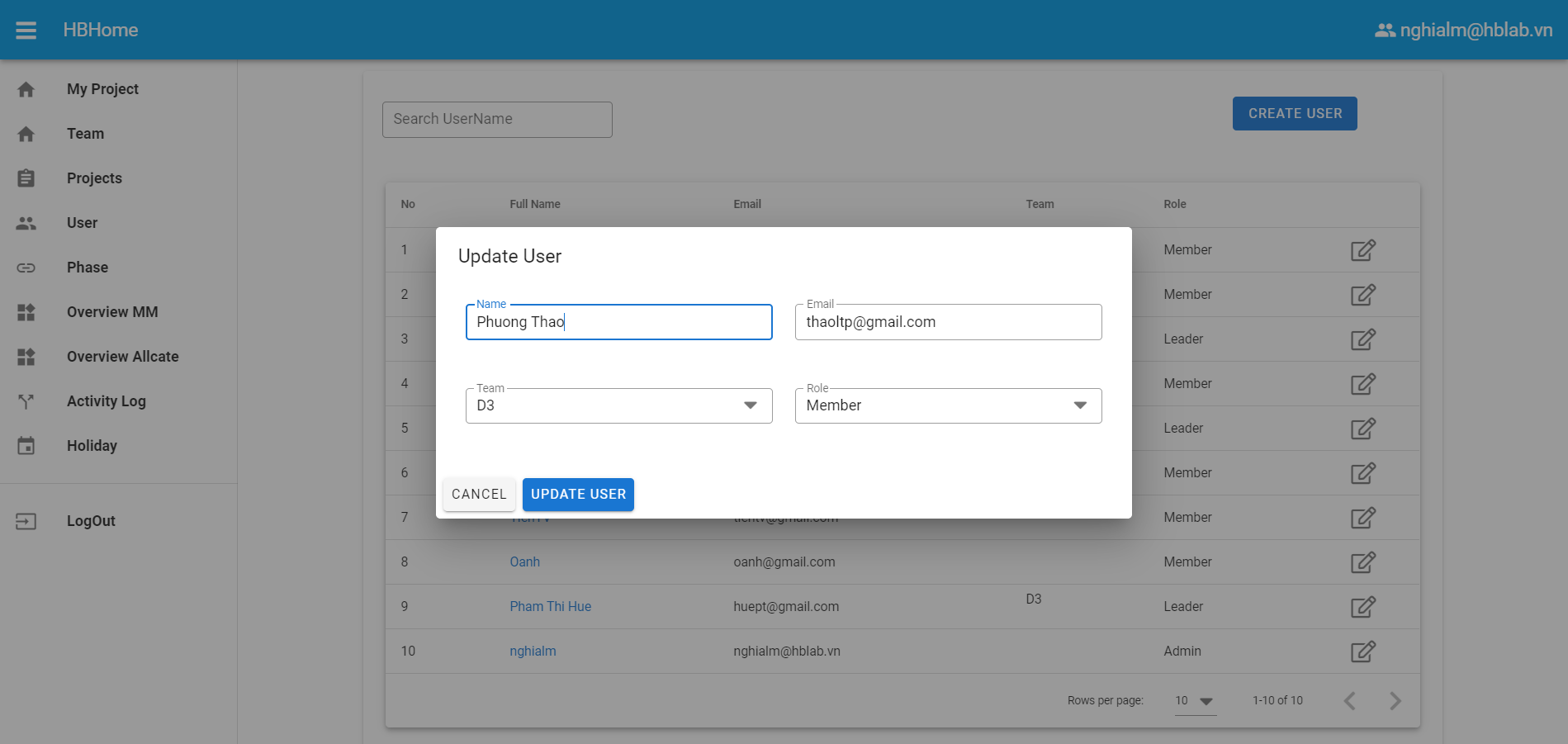
Mỗi khi thực hiện một hành động với Project thì hành động đó sẽ được lưu lại và hiển thị ở đây

+Người dùng có thể thực hiện tìm kiếm và lọc dữ liệu bằng cách chọn các thông số như trên hình

5.3 User



Danh sách các user



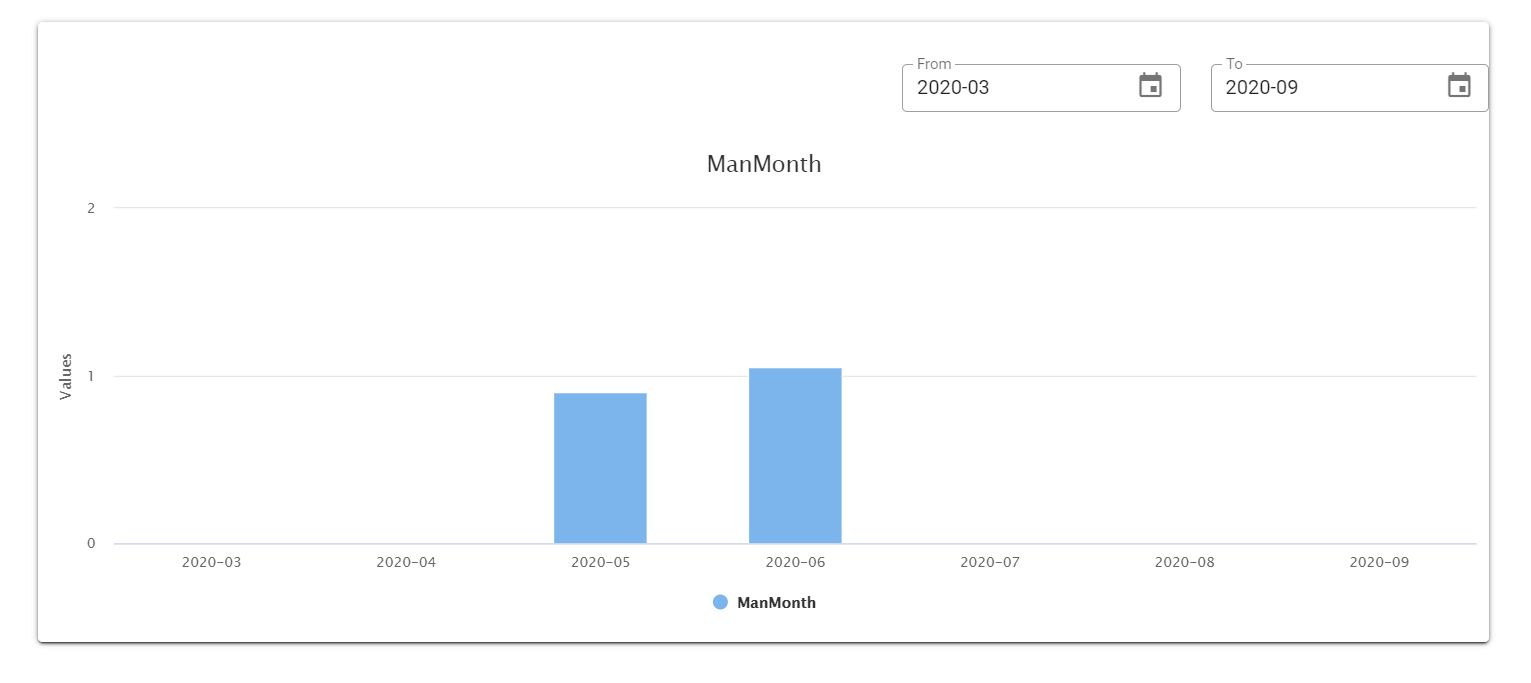
Thêm mới hoặc update User

Giải thích

-Khi vào trang quản lý User sẽ hiên thị ra danh sách thông tin User.

-Tại đây người dùng có thể thực hiện thêm mới hoặc update User

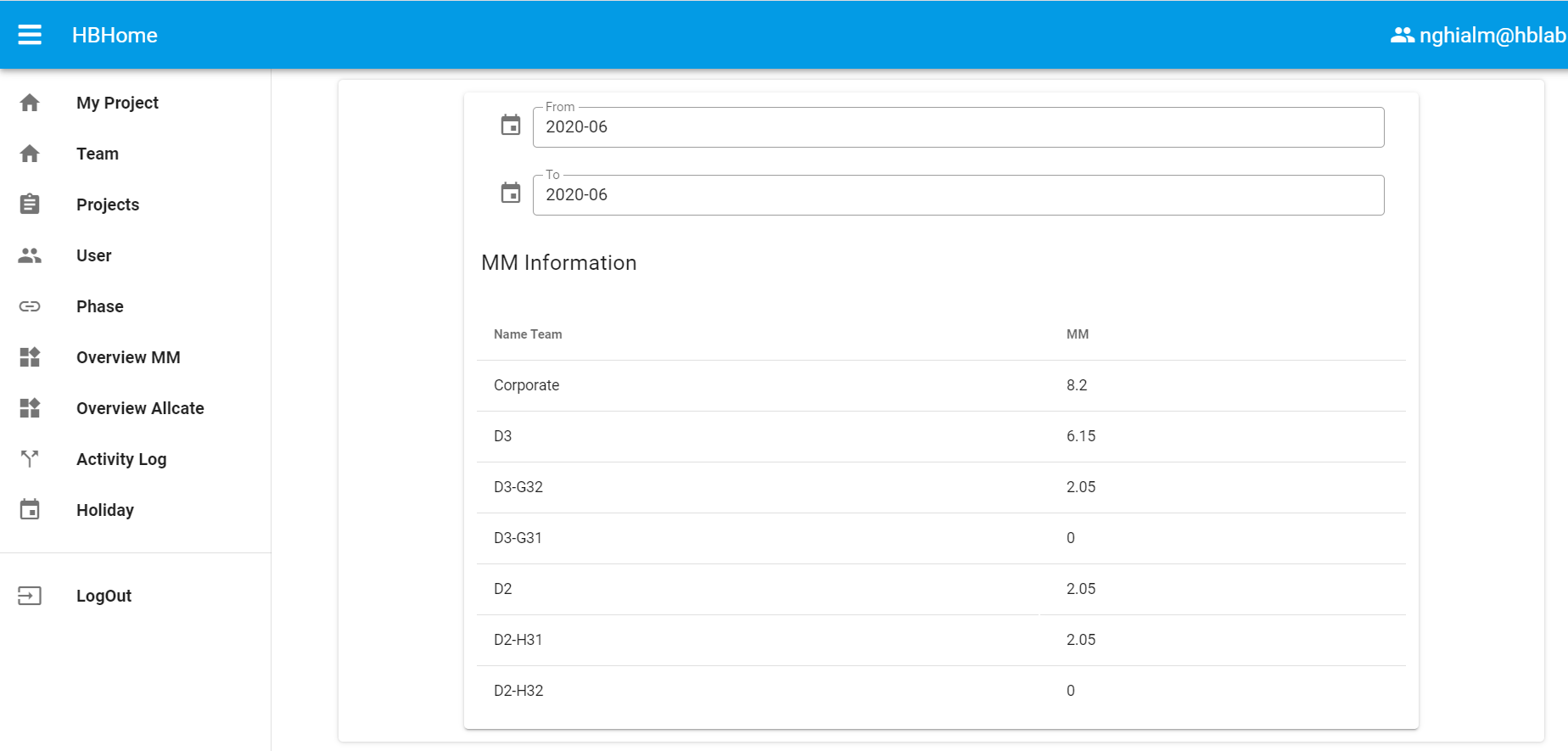
5.4 User Detail



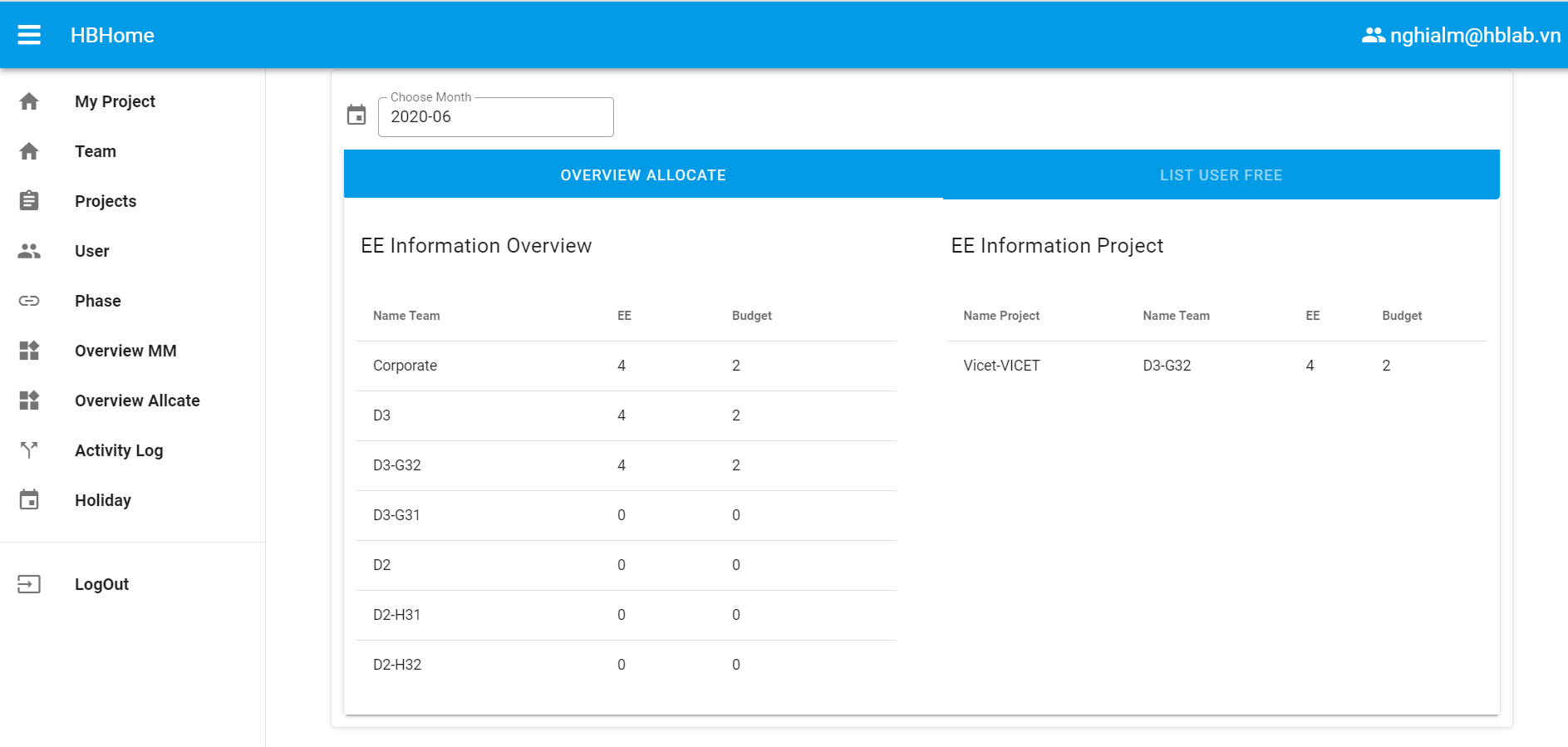
Giải thích:

Khi chọn vào user detail sẽ hiển thị thông tin dạng biểu đồ về mammonth và BusyRate của user đó

5.5 Alocate và MM



MM Infor

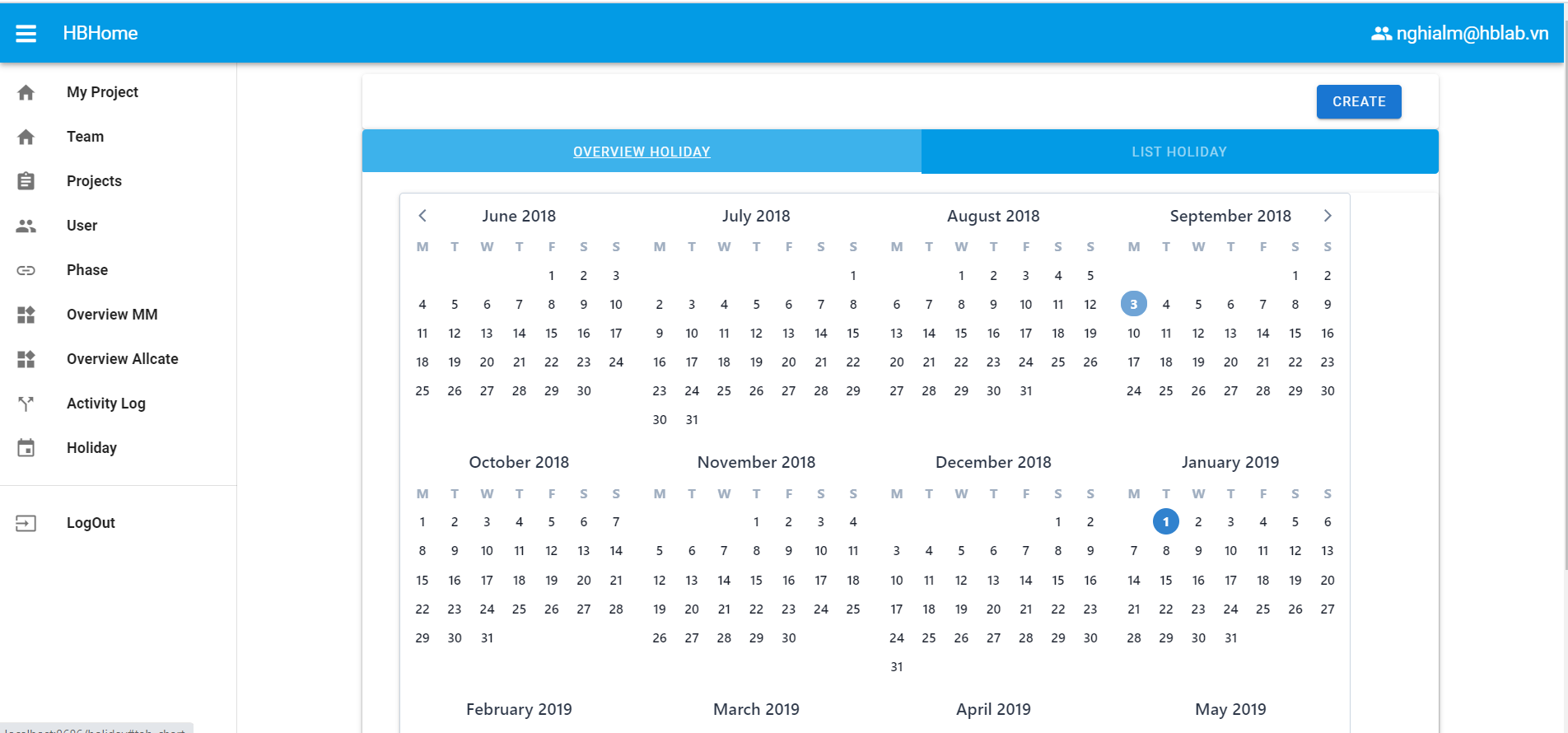


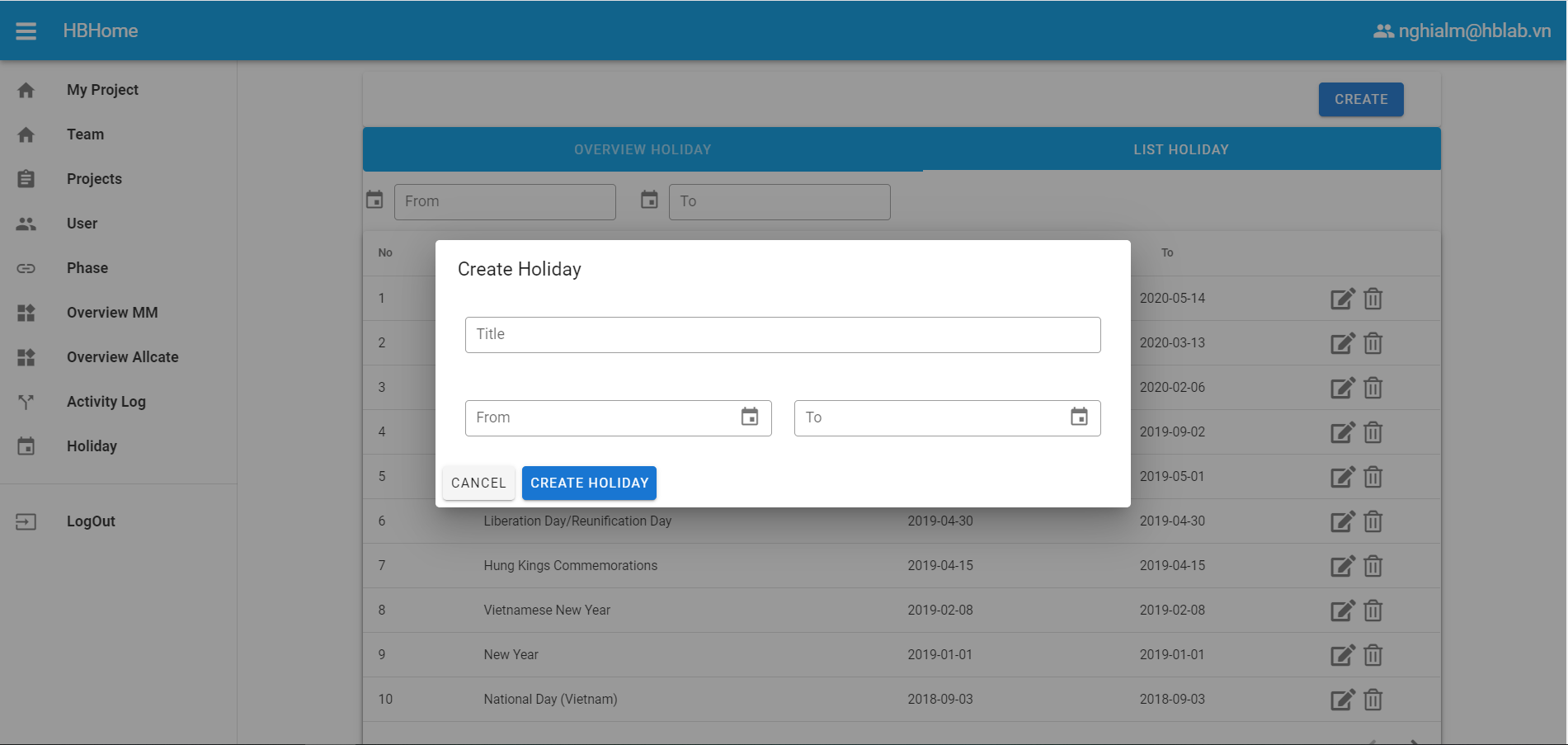
Alocate Infor

Giải thích:

Hiển thị thông tin manmonth và alocate của toàn bộ team và cả công ty

5.6 Holiday





Giải thích

Tại đây hiển thị toàn bộ ngày nghỉ của công ty trên calendar. Khi chỉ vào ngày đó sẽ hiển thị thông tin về kì nghỉ

Action:

-Người dùng có thể thực hiện thêm mới ngày nghỉ

# 

# KẾT LUẬN

Sau một thời gian nghiên cứu đề tài cùng với sự giúp đỡ của thầy hướng dẫn, em đã hoàn thành đồ án tốt nghiệp với đề tài:**’Xây dựng Website giới thiệu tour du lịch Việt’**. Trong phần này em sẽ trình bày những kết quả đạt được sau khi hoàn thành đề tài, những hạn chế của đề tài và hướng phát triển hệ thống.

* **Những kết quản đạt được**

Trong quá trình tìm hiểu và hoàn thành đề tài, em đã đạt được môt số kết quả sau:

* Biết được nhiều công nghệ, kiến trúc mới đang được áp dụng trong phát triển phần mềm hiện nay.
* Nâng cao khả năng tìm tự tìm hiểu các công nghệ mới và áp dụng trong phát triển phần mềm. Xây dựng được website giới thiệu các tour du lịch ở Việt Nam.
* **Hạn chế**
* Hệ thống chưa được chạy kiểm nghiệm trên một host thực tế và chưa được test bởi một phần mềm kiểm thử nào vì vậy chưa đánh giá được hiệu năng, tốc độ xử lý dữ liệu của hệ thống.
* Do thời gian có hạn cũng như hạn chế về kinh nghiệm của bản thân nên có một số chức năng chưa thể phát triển được theo mong muốn. Em rất mong nhận được những lời nhận xét góp ý để tiếp tục phát triển công cụ hoàn thiện hơn.
* **Hướng phát triển**
* Tích hợp thêm tính năng chat giữa người dùng trên website.
* Mở rộng thêm cả tour ở nước ngoài.
* Kết hợp với tính năng tìm vé máy bay đặt phòng đặt xe.
* Phát triển thêm tính năng đánh giá cho tour và bài đăng.
* Kết hợp với Machine Learning và Data Mining giúp việc đưa ra gợi ý lựa chọn tour cho phù hợp với khách hàng.

# TÀI LIỆU THAM KHẢO

[1] <https://laravel.com/>

[2]Bùi Minh Phụng, Phân tích thiết kế hệ thống hướng đối tượng bằng UML, <https://sites.google.com/site/buiminhphung/pttkhdt>

[3] <https://www.google.com.vn/>

[4] <https://youtube.com/>

[5] <https://www.w3schools.com/>

[6] <https://getbootstrap.com/>

[7] <https://jquery.com/>