**ĐẠI HỌC ĐÀ NẴNG**

**ĐẠI HỌC BÁCH KHOA**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO MÔN HỌC**

**CÔNG NGHỆ PHẦN MỀM**

ĐẶC TẢ YÊU CẦU PHẦN MỀM

HỆ THỐNG QUẢN LÝ NHÓM

**SINH VIÊN THỰC HIỆN:**

|  |  |  |
| --- | --- | --- |
| HỌ TÊN SINH VIÊN | MÃ SINH VIÊN | NHÓM HP |
| Nguyễn Quốc Anh | 102220003 | 22Nh15 |
| Bùi Chí Cường | 102220008 | 22Nh15 |

**Đà Nẵng 2024-2025**

QUÁ TRÌNH THAY ĐỔI

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ngày | Phiên bản | \* A, M, D | Mô tả | Tác giả |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

\* A; Thêm mới; M: Sửa đổi; D: Xóa

|  |  |
| --- | --- |
| Tác giả | Tên: Bùi Chí Cường Ngày: 22/04/2024  Vai trò: Thiết kế hệ thống.  Tên: Nguyễn Quốc Anh Ngày: 22/04/2024  Vai trò: Thiết kế UI |

MỤC LỤC

[1](#_Toc166868744)

[1. Giới thiệu 6](#_Toc166868745)

[1.1. Tổng quan 6](#_Toc166868746)

[1.2. Mục đích 6](#_Toc166868747)

[1.3. Phạm vi 6](#_Toc166868748)

[1.4. Thuật ngữ viết tắt 6](#_Toc166868749)

[1.5. Tài liệu tham khảo 7](#_Toc166868750)

[2. Mô tả tổng quan 7](#_Toc166868751)

[2.1. Mô hình các tình huống sử dụng tổng quan 7](#_Toc166868752)

[2.2. Mô tả sơ đồ tổng quan 7](#_Toc166868753)

[3. Đặc tả yêu cầu chức năng 8](#_Toc166868754)

[3.1. Các tác nhân 8](#_Toc166868755)

[3.2. Các chức năng của hệ thống: 8](#_Toc166868756)

[3.3. Biểu đồ use case tổng quan: 9](#_Toc166868757)

[3.4. Biểu đồ use case phân rã: 10](#_Toc166868758)

[3.4.1 Phân rã use case “Khách hàng”: 10](#_Toc166868759)

[3.4.2 Phân rã use case “Trưởng nhóm”: 12](#_Toc166868760)

[3.4.3 Phân rã use case “Người sử dụng” 14](#_Toc166868761)

[3.4.4 Phân rã use case “Thành viên nhóm”: 16](#_Toc166868762)

[3.4.5 Phân rã use case “Admin”: 18](#_Toc166868763)

[3.4.6 Phân rã use case “Người quản lý dự án”: 19](#_Toc166868764)

[3.5. Sơ đồ lớp ( Class Diagram): 21](#_Toc166868765)

[3.6. Quy trình nghiệp vụ: 22](#_Toc166868766)

[3.7. Sơ đồ tuần tự: 23](#_Toc166868767)

[4. Yêu cầu phi chức năng. 23](#_Toc166868768)

[4.1. Yêu cầu mức độ tin cậy. 23](#_Toc166868769)

[4.2. Yêu cầu bảo mật 23](#_Toc166868770)

[4.3. Yêu cầu giao diện 23](#_Toc166868771)

[4.4. Yêu cầu hiệu suất 23](#_Toc166868772)

[4.5. Yêu cầu khả năng hỗ trợ và bảo trì 23](#_Toc166868773)

[4.6. Yêu cầu về môi trường 23](#_Toc166868774)

[4.7. Yêu cầu hệ thống tài liệu trực tuyến và hệ thống trợ giúp 23](#_Toc166868775)

[4.8. Thành phần mua ngoài 24](#_Toc166868776)

[4.9. Yêu cầu pháp lý, bản quyền và các lưu ý khác 24](#_Toc166868777)

[4.10. Tiêu chuẩn áp dụng 24](#_Toc166868778)

[5. Đặc tả giao diện UI/UX. 24](#_Toc166868779)

[6. UNIT TEST. 29](#_Toc166868780)

[6.1. Khái niệm Unit Test. 29](#_Toc166868781)

[6.2. Khái niệm Unit. 29](#_Toc166868782)

[6.3. Công cụ thực hiện Unit test. 29](#_Toc166868783)

[6.4. Các thuật ngữ khi sử dụng Unit Test. 30](#_Toc166868784)

[6.5. Vòng đời của Unit Test. 30](#_Toc166868785)

[6.6. Ưu điểm và nhược điểm của Unit Test. 30](#_Toc166868786)

[6.7. Thiết kế Unit test 31](#_Toc166868787)

[7. CODE STYLE JAVA 33](#_Toc166868788)

[7.1. Khái niệm: 33](#_Toc166868789)

[7.2. Mục đích của viết code theo quy chuẩn quy định. 34](#_Toc166868790)

[7.3. Chuẩn hình thức và ngữ nghĩa. 34](#_Toc166868791)

[7.3.1 Chuẩn hình thức . 34](#_Toc166868792)

[7.3.2 Chuẩn ngữ nghĩa. 35](#_Toc166868793)

# Giới thiệu

## Tổng quan

Hệ thống quản lý dự án nhóm là một phần mềm hoặc nền tảng công nghệ được sử dụng để quản lý và tổ chức các dự án trong một nhóm làm việc. Mục tiêu chính của hệ thống này là hỗ trợ các thành viên trong nhóm để họ có thể làm việc hiệu quả hơn, tăng cường sự hợp tác và tạo điều kiện cho việc quản lý dự án trở nên dễ dàng và hiệu quả hơn.

## Mục đích

Cung cấp cái nhìn tổng quan về dự án: Tài liệu này giúp cung cấp một cái nhìn tổng quan, dễ hiểu về các yêu cầu, thành phần của dự án. Điều này giúp cho tất cả các thành viên trong nhóm hiểu rõ về phạm vi, yêu cầu và mục tiêu của dự án.

Tài liệu tham khảo cho sinh viên: Tài liệu này được cung cấp như một tài liệu tham khảo cho sinh viên trực tiếp tham gia phát triển dự án, phục vụ cho việc hoàn thành đồ án tốt nghiệp. Nó giúp các sinh viên hiểu rõ hơn về công việc của họ, các yêu cầu và quy trình trong dự án.

Phục vụ cho môi trường thực tế: Ngoài ra, trong môi trường thực tế bên ngoài, tài liệu này còn phục vụ cho những nhà phát triển phần mềm, kiểm thử viên, nhà quản lý dự án cũng như các bên liên quan. Điều này giúp đảm bảo rằng mọi người liên quan đều hiểu rõ về dự án, giúp tăng tính hiệu quả và chính xác trong quá trình phát triển.

## Phạm vi

Tài liệu đặc tả yêu cầu phần mềm này được xây dựng nhằm phục vụ cho dự án Phát triển hệ thống Group Management System phục vụ công việc quản lý tiến độ công việc. Với việc phát triển hệ thống cung cấp cho khách hàng, tổ chức cá nhân một công cụ đi kèm với nhiều lợi ích về kinh tế với chi phí thấp nhưng vẫn đảm bảo được tiến độ cũng như chất lượng của dự án.

## Thuật ngữ viết tắt

|  |  |
| --- | --- |
| Group Management System | GMS |
| Web server | Phần mềm máy chủ cung cấp các chức năng, tài nguyên cho máy khác |
| Use Case(s) | Biểu đồ mô tả những yêu cầu của hệ thống |
| ACTIVITY DIAGRAM | Hiển thị quy trình làm việc từ điểm bắt đầu đến điểm kết thúc. |
| SEQUENCE DIAGRAM | Biểu đồ mô tả cách thức—và thứ tự—một nhóm đồ vật hoạt động cùng nhau |

## Tài liệu tham khảo

[1] IViettech.(2014). Classs Diagarm: Bản vẽ về Class(lớp).Từ: <https://iviettech.vn/blog/813-classs-diagarm-ban-ve-ve-classlop.html>

[2] VIBLO.(2016). Phân tích thiết kế hệ thống thông tin sử dụng biểu đồ UML.Từ: <https://viblo.asia/p/phan-tich-thiet-ke-he-thong-thong-tin-su-dung-bieu-do-uml-phan-2-0bDM6wpAG2X4>

[3] CODESTAR.(2015).Tài liệu đặc tả SRS trong phân tích yêu cầu.Từ: <https://codestar.vn/tai-lieu-dac-ta-srs-trong-phan-tich-yeu-cau/>

[4]CODELEARN.(2020).Các chuẩn coding style trong javascript.Từ: <https://codelearn.io/sharing/cac-chuan-coding-style-trong-javascript>.

[5]TOPDEV.(2017). Unit Test là gì? Khái niệm và vai trò.Từ: <https://topdev.vn/blog/unit-test-la-gi/>.

# Mô tả tổng quan

Tài liệu này được viết dựa theo chuẩn của Tài liệu đặc tả yêu cầu phần mềm (Group Management System - GMS) .

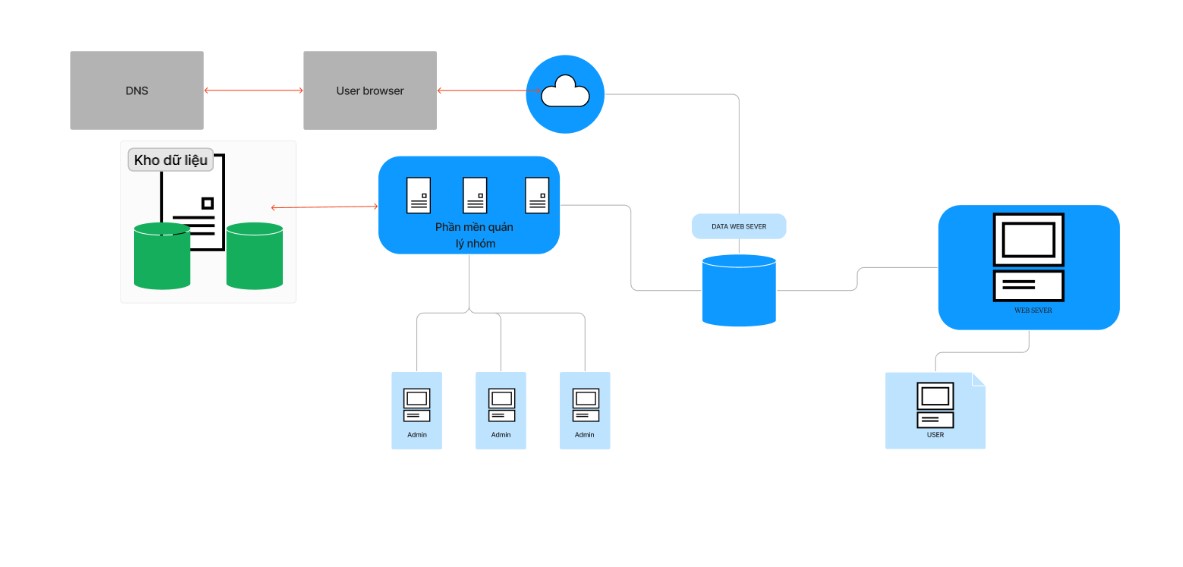
Với cấu trúc được chia làm ba phần:

Phần 1: Cung cấp cái nhìn tổng quan về các thành phần của GMS.

Phần 2: Mô tả tổng quan các nhân tố, ràng buộc, đặc điểm người dùng, môi trường thực thi tác động lên hệ thống và các yêu cầu của nó. Cung cấp thông tin chi tiết các yêu cầu chức năng, cung cấp cho các nhà phát triển phần mềm thông tin để phát triên phần mềm đáp ứng được các yêu cầu đó.

Phần 3: Các yêu cầu phi chức năng

## Mô hình các tình huống sử dụng tổng quan



## Mô tả sơ đồ tổng quan

1. Lớp dữ liệu:

* Data Source: Dùng để lưu trữ dữ liệu thực của người dùng.
* Kho dữ liệu : Dùng để lưu dữ liệu hệ thống quản lý

1. Lớp bảo mật:

* Máy chủ(admin):Xác thực người dùng và mã hóa dữ liệu.

1. Lớp giao diện:

* Trang chủ:Hiện thị thông tin tổng thể,thông tin các nhóm và tiến độ làm việc
* Tạo dự án:From để tạo dự án mới và tạo nhóm thực hiện
* Liên hệ khách hàng: Đặt lịch hẹn với khách.
* Xem tiến độ làm việc chi tiết các thành viên trong nhóm.

# Đặc tả yêu cầu chức năng

## Các tác nhân

- Khách hàng

- Quản lý

- Admin

- Thành viên nhóm

- Trưởng nhóm

- Người quản lý dự án

## Các chức năng của hệ thống:

Đăng nhập: Chức năng này nhằm mục đích xác thực người dùng khi tương tác với hệ thống nhằm cung cấp quyền cũng như phạm vi truy cập hệ thống.

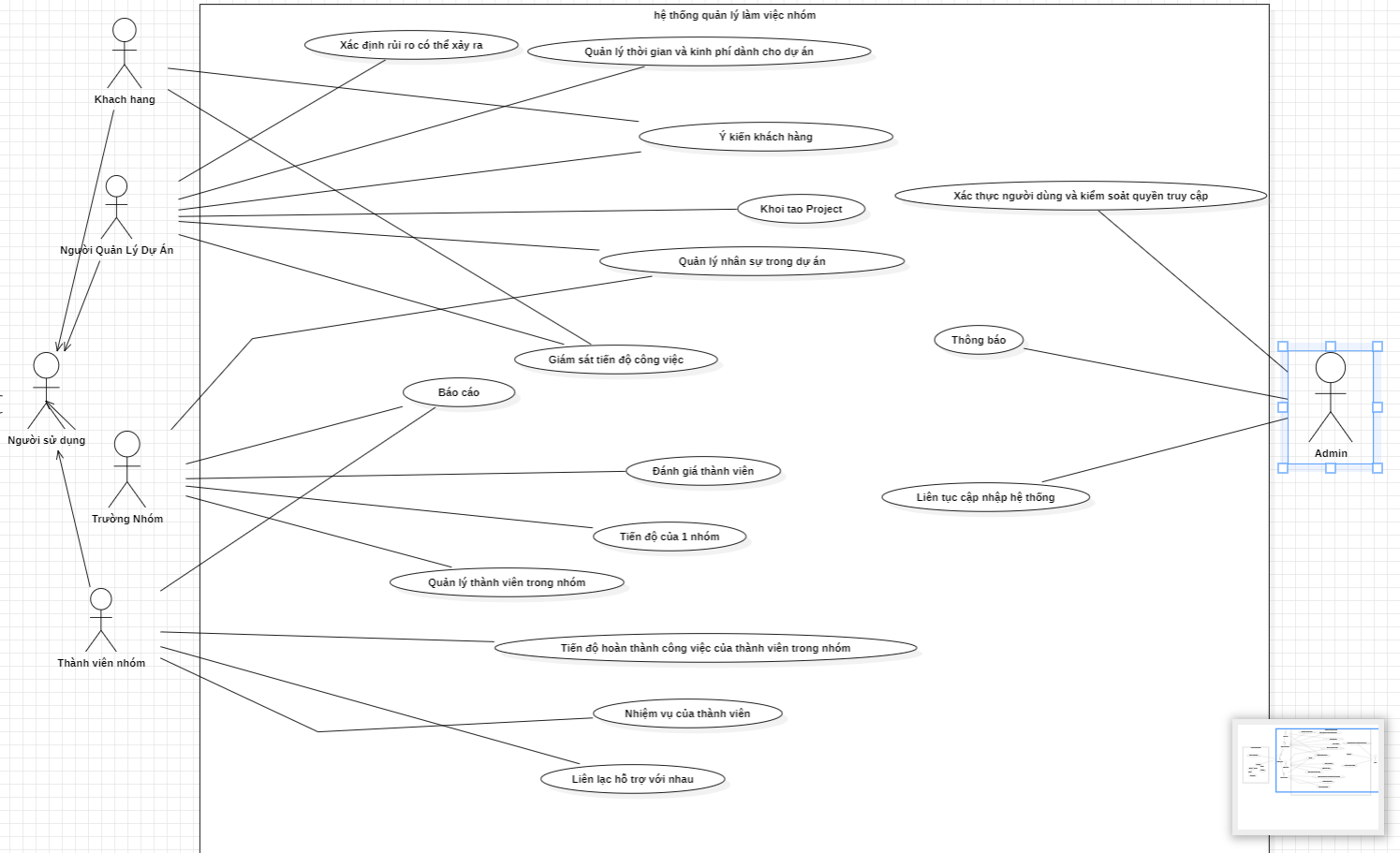
Đăng ký: Để truy cập sử dụng hệ thống thì Người dùng trước hết cần đăng ký tài khoản.

Admin: Có vai trò quản trị những người dùng trong hệ thống.

Các nhóm chức năng quản lý dự án, tiến độ: Công việc quản trị khóa học của Giảng viên.

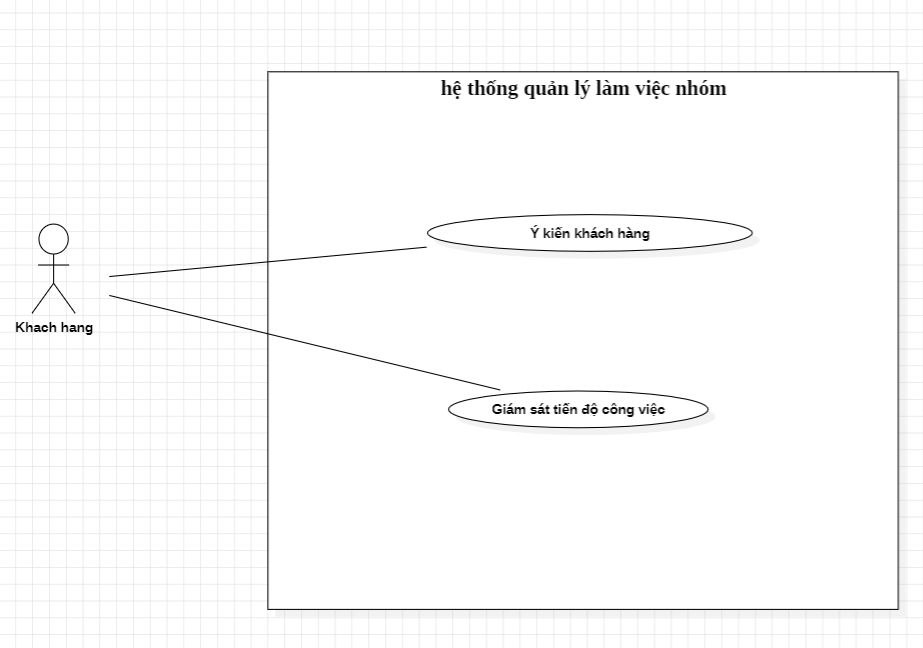
Để có thể hình dung rõ hơn về các tác nhân cũng như yêu cầu chức năng của hệ thống bằng cách mô hình hóa chúng dưới các sơ đồ use cases, các sơ đồ sẽ được trình bày phía sau.

## Biểu đồ use case tổng quan:



## Biểu đồ use case phân rã:

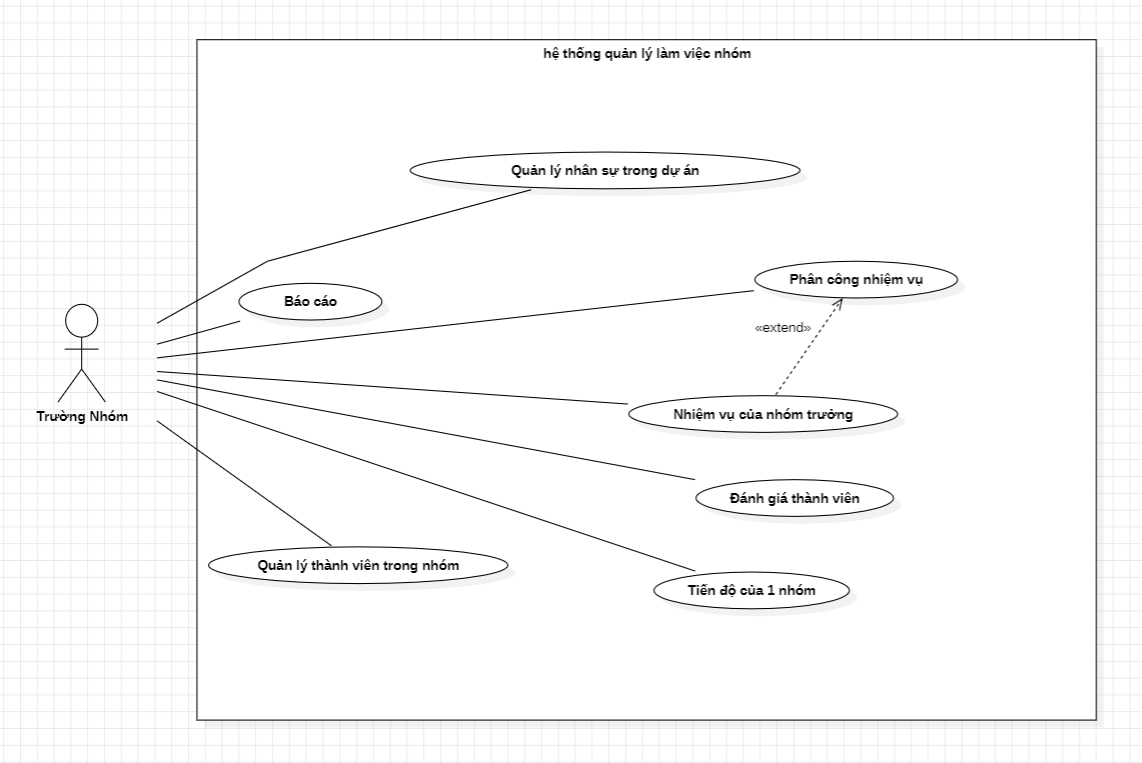
### Phân rã use case “Khách hàng”:



|  |  |
| --- | --- |
| Tên Use Case:Ý kiến và giám sát tiến độ công việc của khách hàng. | |
| Tác nhân | Khách hàng. |
| Mô tả | Khách hàng có thể đóng góp ý kiến và giám sát tiến độ |
| Sự kiện kích hoạt | Tác nhân kích hoạt các thao tác tương ứng với use case |
| Tiền điều kiện | Tác nhân đã đăng nhập vào hệ thống |
| Luồng sự kiện chính(Thành công) | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | Khách hàng | Chọn chức năng xem quá trình làm việc của nhóm. | | 2 | Hệ thống | Hiển thị tiến trình công việc của các thành viên nhóm | | 3 | Khách hàng | Đóng góp ý kiến | | 4 | Hệ thống | Hiển thị ý kiến đến các nhóm tương ứng | |
| Luồng sự kiện thay thế | |  |  |  |  | | --- | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | | 2.1 | Hệ thống | | Thông báo lỗi:chưa có nhóm thực hiện | | 2 |  | |  | |
| Hậu điều kiện | Khách xem được tiến độ công việc và đóng góp ý kiến |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Trường dữ liệu** | **Mô tả** | **Bắt buộc** | **Điều kiện hợp lệ** | **Ví dụ** |
| 1 | ID | Nhập mã khách hàng | Có | Trùng với căn cước | 20100405 |
| 2 | SĐT | Nhập số điện thoại | Có | Trùng với số điện thoại khách hàng | 0375049213 |
| 3 | Maduan | Nhập mã dự án | Có | Trùng với mã dự án được giao | 001 |

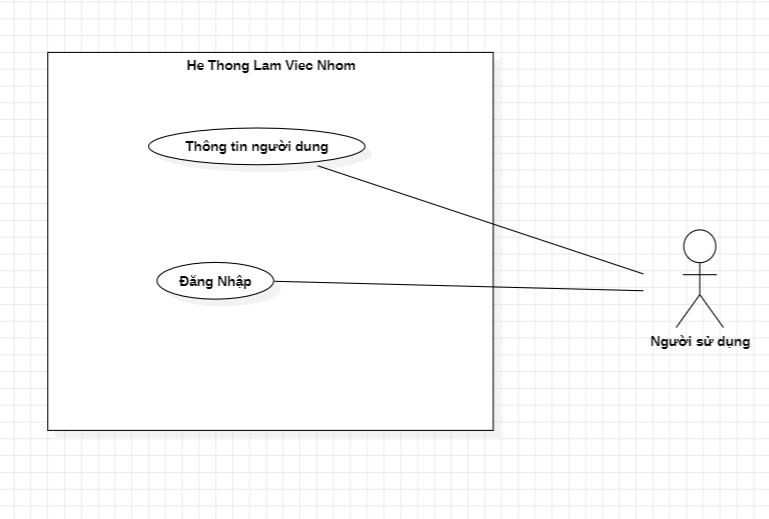
### Phân rã use case “Trưởng nhóm”:



|  |  |
| --- | --- |
| Tên Use Case:Quản lý,báo cáo nhân sự trong dự án | |
| Tác nhân | Trưởng nhóm. |
| Mô tả | Trưởng nhóm có thể xem tiến độ công việc, báo cáo và giám sát thành viên nhóm |
| Sự kiện kích hoạt | Tác nhân kích hoạt các thao tác tương ứng với use case |
| Tiền điều kiện | Tác nhân đã đăng nhập vào hệ thống |
| Luồng sự kiện chính(Thành công) | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | Trưởng nhóm | Chọn chức năng xem quá trình làm việc của nhóm. | | 2 | Hệ thống | Hiển thị tiến trình công việc của các thành viên nhóm | | 3 | Trưởng nhóm | Chọn chức năng quản lý nhân sự nhóm | | 4 | Hệ thống | Hiển thị thông tin các nhân sự | | 5 | Trưởng nhóm | Cấp quyền truy cập cho nhân sự mới | | 6 | Hệ thống | Cập nhật và thông báo thành công | | 7 | Trưởng nhóm | Phân chia nhiệm vụ cho các thành viên nhóm | | 8 | Hệ thống | Thông báo nhiệm vụ đến các thành viên | |
| Luồng sự kiện thay thế | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 2.1 | Hệ thống | Thông báo lỗi:chưa có nhóm thực hiện | | 4.1 | Hệ thống | Thông báo lỗi:chưa có thành viên nhóm | | 6.1 | Hệ thống | Thông báo lỗi: cập nhập thất bại yêu cầu cấp quyền lại | | 8.1 | Hệ thống | Thông báo lỗi: chưa có thành viên nhóm | |
| Hậu điều kiện | Khách xem được tiến độ công việc và đóng góp ý kiến |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Trường dữ liệu** | **Mô tả** | **Bắt buộc** | **Điều kiện hợp lệ** | **Ví dụ** |
| 1 | Manhom | Nhập mã nhóm | Có | Trùng với id nhóm được tạo | 1415 |
| 2 | ID | Nhập mã trưởng nhóm | Có | Trùng với số căn cước | 204156897 |
| 3 | Thoigian | Nhập thời gian làm dự án | Có | Trùng với thời gian dự án | 02/07/2024 12h00 |

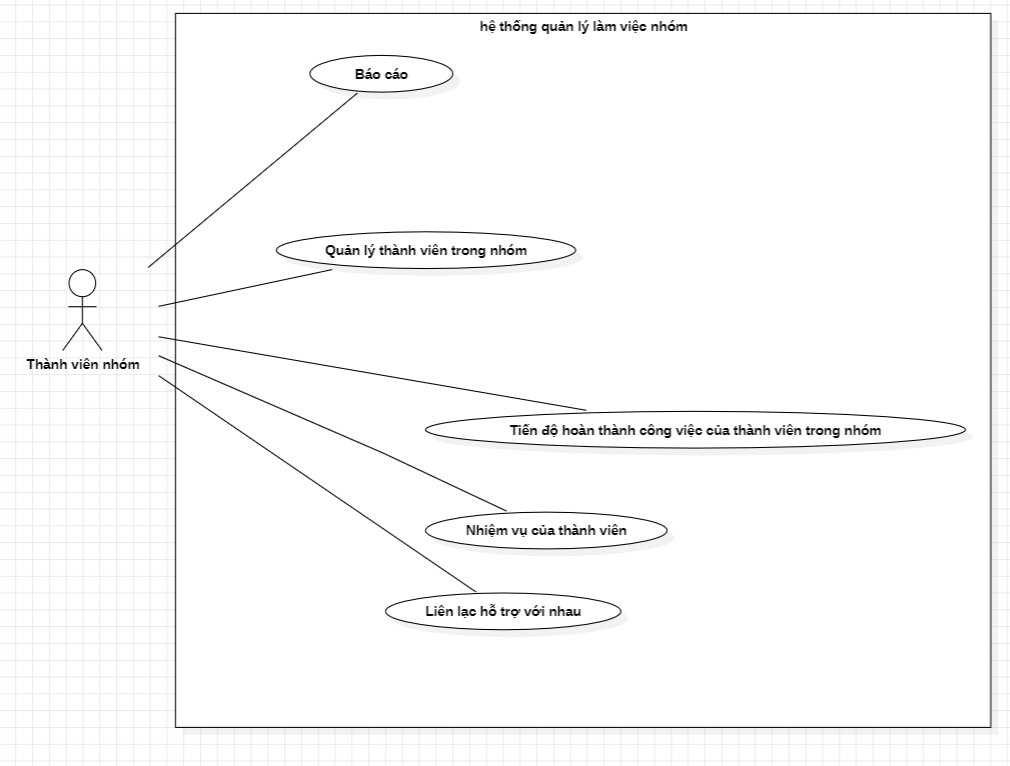
### Phân rã use case “Người sử dụng”



|  |  |
| --- | --- |
| Tên Use Case: Thông tin người dùng và đăng nhập | |
| Tác nhân | Tất cả người sử dụng hệ thống. |
| Mô tả | Để sử dụng hệ thống thì người dùng phải nhập thông tin của mình và khởi tạo tài khoản, mật khẩu để đăng nhập vào sử dụng. |
| Sự kiện kích hoạt | Giao diện người dùng sẽ được hiện thị sau khi đăng nhập thành công. |
| Tiền điều kiện | Tác nhân đã có tài khoản |
| Luồng sự kiện chính(Thành công) | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | Người dùng | Nhập thông tin của mình và khởi tạo tài khoản. | | 2 | Hệ thống | Lưu trữ thông tin người dùng vào CSDL của hệ thống. | |
| Luồng sự kiện thay thế | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 2.1 | Hệ thống | Thông báo lỗi:chưa có nhóm thực hiện | | 2 |  |  | |
| Hậu điều kiện | Đăng nhập thành công |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Trường dữ liệu** | **Mô tả** | **Bắt buộc** | **Điều kiện hợp lệ** | **Ví dụ** |
| 1 | Taikhoan | Nhập mã tài khoản | Có | Trùng với id người dùng | 204156897 |
| 2 | Matkhau | Nhập mật khẩu | Có | Trùng với mật khẩu người dùng | Kho1546 |

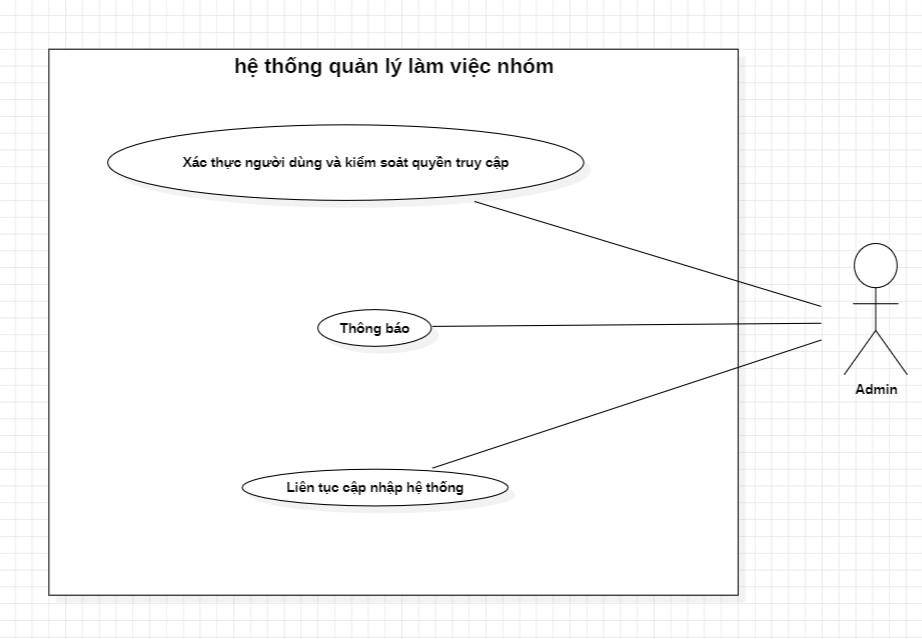
### Phân rã use case “Thành viên nhóm”:



|  |  |
| --- | --- |
| Tên Use Case: Báo cáo, tiến độ công việc, nhiệm vụ, hỗ trợ. | |
| Tác nhân | Thành viên nhóm. |
| Mô tả | Mỗi thành viên trong nhóm sẽ được nhóm trưởng phân chia nhiệm vụ khác nhau, mỗi nhiệm vụ sẽ có thời hạn và phải hoàn thành trong thời gian đó, trong thời gian làm phải liên tục báo cáo lại tiến độ, mọi người trong nhóm hỗ trợ lẫn nhau. |
| Sự kiện kích hoạt | Giao diện người dùng sẽ được hiện thị sau khi đăng nhập thành công. |
| Tiền điều kiện | Tác nhân đã đăng nhập vào hệ thống |
| Luồng sự kiện chính(Thành công) | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | Thành viên nhóm | Nhận nhiệm vụ từ trưởng nhóm sau đó thực hiện nhiện vụ. | | 2 | Hệ thống | Cung cấp dữ liệu nhiệm vụ từ trường nhóm xuống cho các thành viên nhóm. | |
| Luồng sự kiện thay thế | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 2.1 | Hệ thống | Hệ báo lôi: không tìm thấy thành viên | | 2 |  |  | |
| Hậu điều kiện | Trường nhóm và người quản lý dự án có thể giám sát công việc của từng thành viên trong dự án. |

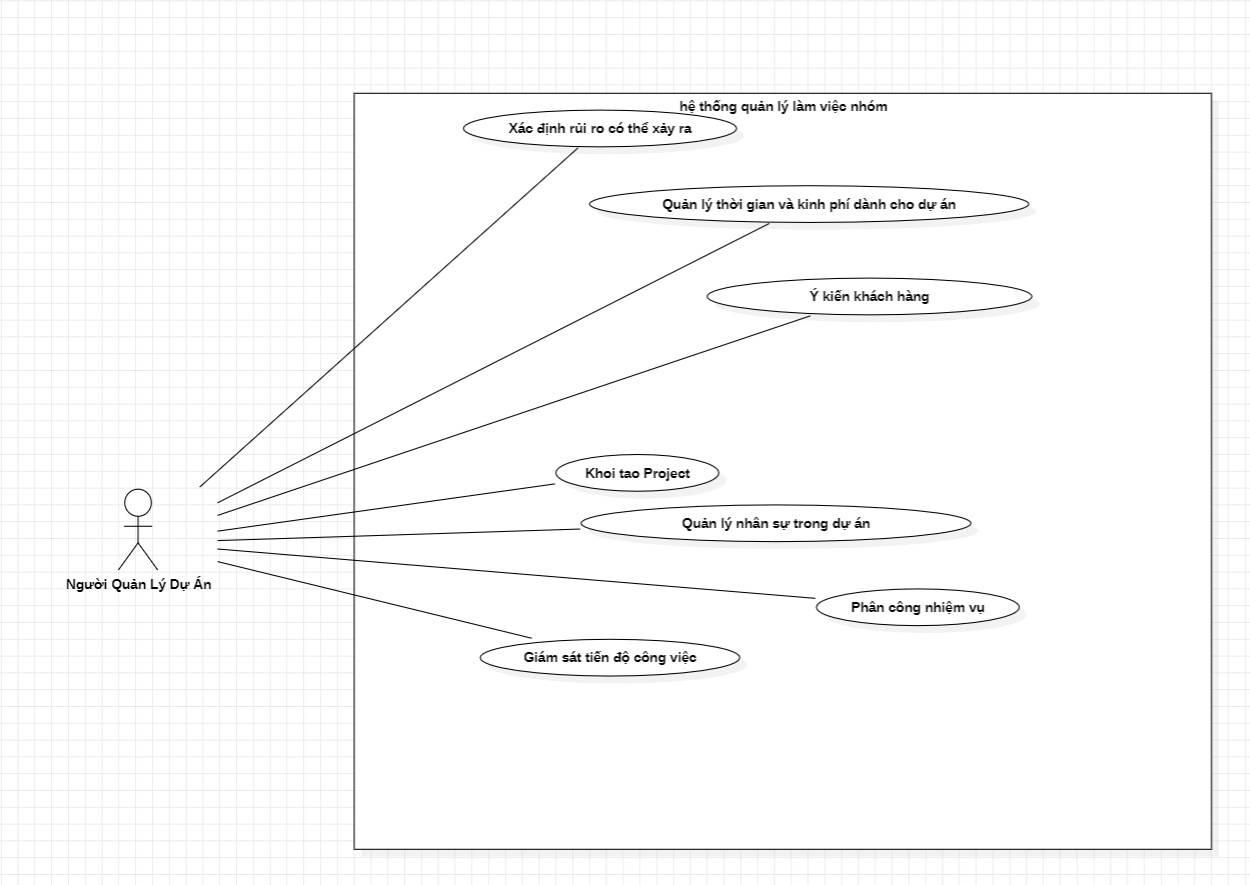
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Trường dữ liệu** | **Mô tả** | **Bắt buộc** | **Điều kiện hợp lệ** | **Ví dụ** |
| 1 | ID | Nhập mã tài khoản | Có | Trùng với id người dùng | 204156897 |
| 2 | SĐT | Nhập mật khẩu | Có | Trùng với mật khẩu người dùng | Kho1546 |
| 3 | Manhom | Nhập mã nhóm | Có | Trùng với mã nhóm được tạo | 1415 |

### Phân rã use case “Admin”:



|  |  |
| --- | --- |
| Tên Use Case: Xác thực người dùng kiểm soát truy cập, thông báo, cập nhập hệ thống. | |
| Tác nhân | Admin(hệ thống) |
| Mô tả | Hệ thống sẽ có nhiệm vụ xác thực người sử dụng hệ thống và kiểm soát quyền, số lượng truy cập vào hệ thống để tránh xảy ra quá tải hoặc tấn công. Thông báo đến người dùng về tiến độ công việc, cảnh báo nếu đang cố gắng phá huỷ hệ thống. Liên tục đưa ra những cập nhập nhằm nâng cấp hệ thống. |
| Sự kiện kích hoạt | Hệ thống cần cập nhập và người dùng tạo tài khoản để sử dụng hệ thống. |
| Tiền điều kiện | Tác nhân đã đăng nhập vào hệ thống |
| Luồng sự kiện chính(Thành công) | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | Hệ thống | Xác nhận người dùng từ cơ sỡ dữ liệu người dùng | | 2 | Hệ thống | Cập nhập, nâng cấp hệ thống. | | 3 | Hệ thống | Thông báo tiến độ công việc | |
| Luồng sự kiện thay thế | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1.1 | Hệ thống | Hệ thống báo lỗi: không tìm thấy người dùng | | 2.1 | Hệ thống | Hệ thống báo lỗi: cập nhập không thành công, tải lại hệ thống | |
| Hậu điều kiện | Người dùng nhận được tiến độ công việc |

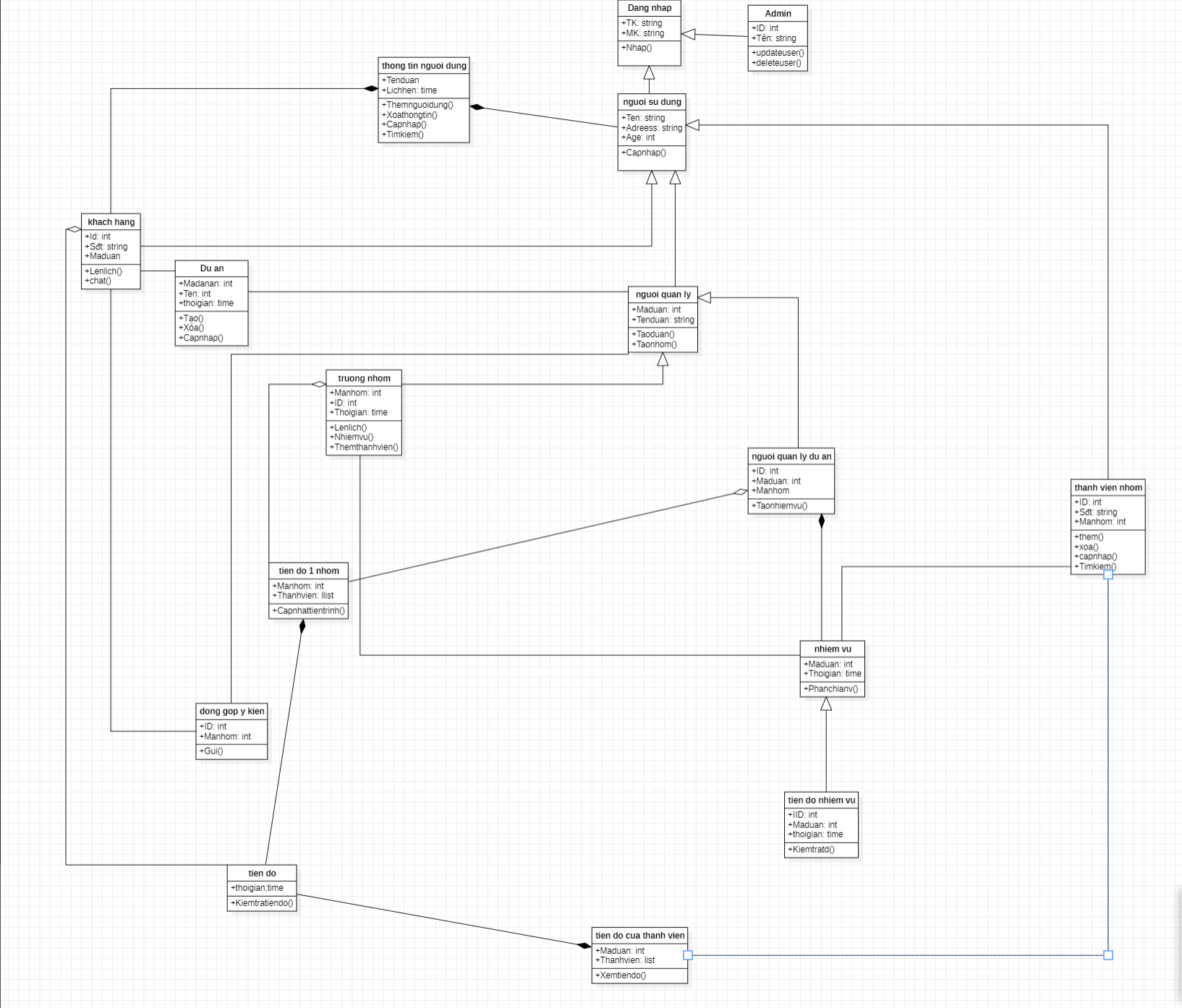
### Phân rã use case “Người quản lý dự án”:



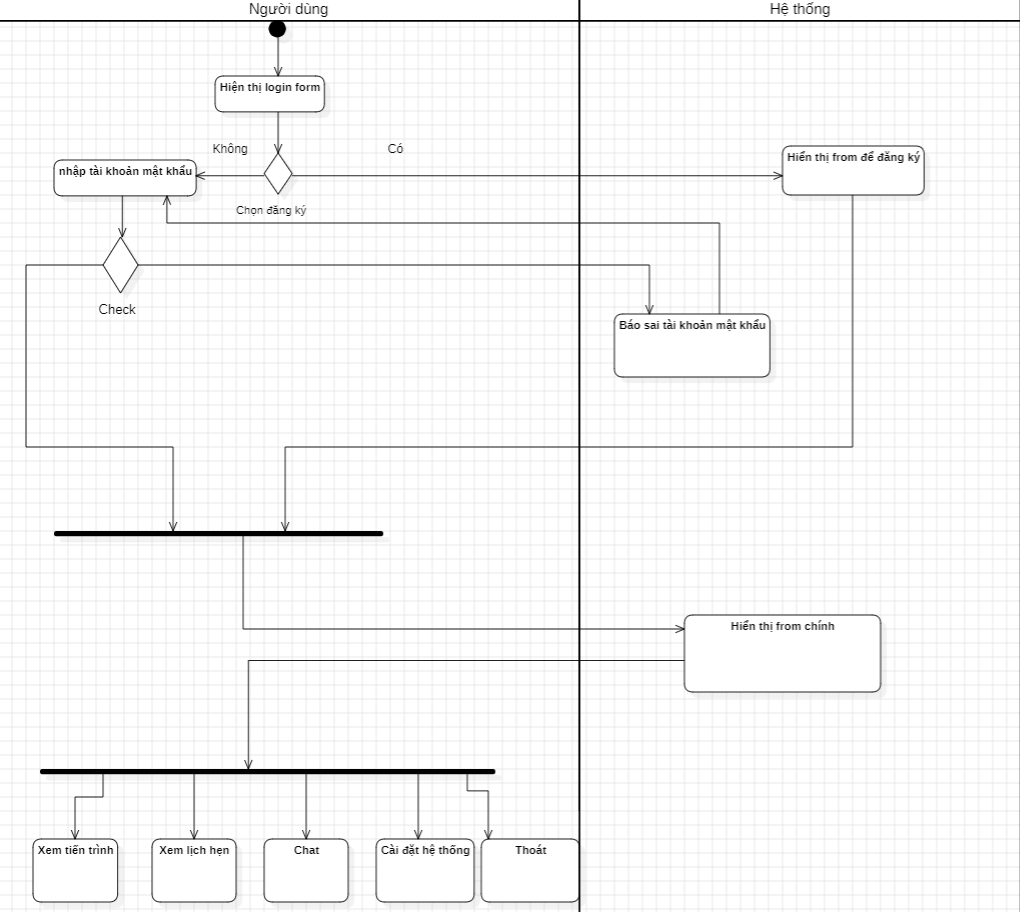
|  |  |
| --- | --- |
| Tên Use Case: Xác nhận rủi ro,khởi tạo dự án và giám sát tiến độ công việc, nhân sự và kinh phí | |
| Tác nhân | Người quản lý |
| Mô tả | Người quản lý dự án có thể khởi tạo dự án, giám sát tiến độ của các nhóm và ghi nhận ý kiến khách hàng |
| Sự kiện kích hoạt | Tác nhân kích hoạt các thao tác tương ứng với use case |
| Tiền điều kiện | Tác nhân đã đăng nhập vào hệ thống. |
| Luồng sự kiện chính(Thành công) | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | Người quản lý | Khởi tạo project | | 2 | Hệ thống | Hiển thị workspace của project và mã dự án. | | 3 | Người quản lý | Chọn giám sát tiến độ công việc | | 4 | Hệ thống | Hiển thị tiến độ các nhóm | | 5 | Người quản lý | Chọn phân chia nhiệm vụ | | 6 | Hệ thống | Gửi nhiệm vụ đến các nhóm trưởng | | 7 | Người quản lý | Chọn ý kiến khách hàng | | 8 | Hệ thống | Hiện thị phản hồi của khách | |
| Luồng sự kiện thay thế | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 2.1 | Hệ thống | Hệ thống báo lỗi: Chưa có project được tạo | | 4.1 | Hệ thống | Hệ thống báo lỗi: Chưa có nhóm làm dự án | | 6.1 | Hệ thống | Hệ thống báo lỗi: Dự án chưa có nhóm trưởng | |
| Hậu điều kiện | Người quản lý nhận được tiến độ của các nhóm cũng như phản hồi khách hàng |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Trường dữ liệu** | **Mô tả** | **Bắt buộc** | **Điều kiện hợp lệ** | **Ví dụ** |
| 1 | ID | Nhập mã của người quản lý | Có | Trùng với id người dùng | 204156897 |
| 2 | Maduan | Nhập mã dự án | Có | Trùng với mã dự án được tạo | 004 |
| 3 | Manhom | Nhập mã nhóm | Có | Trùng với mã nhóm được tạo | 1415 |

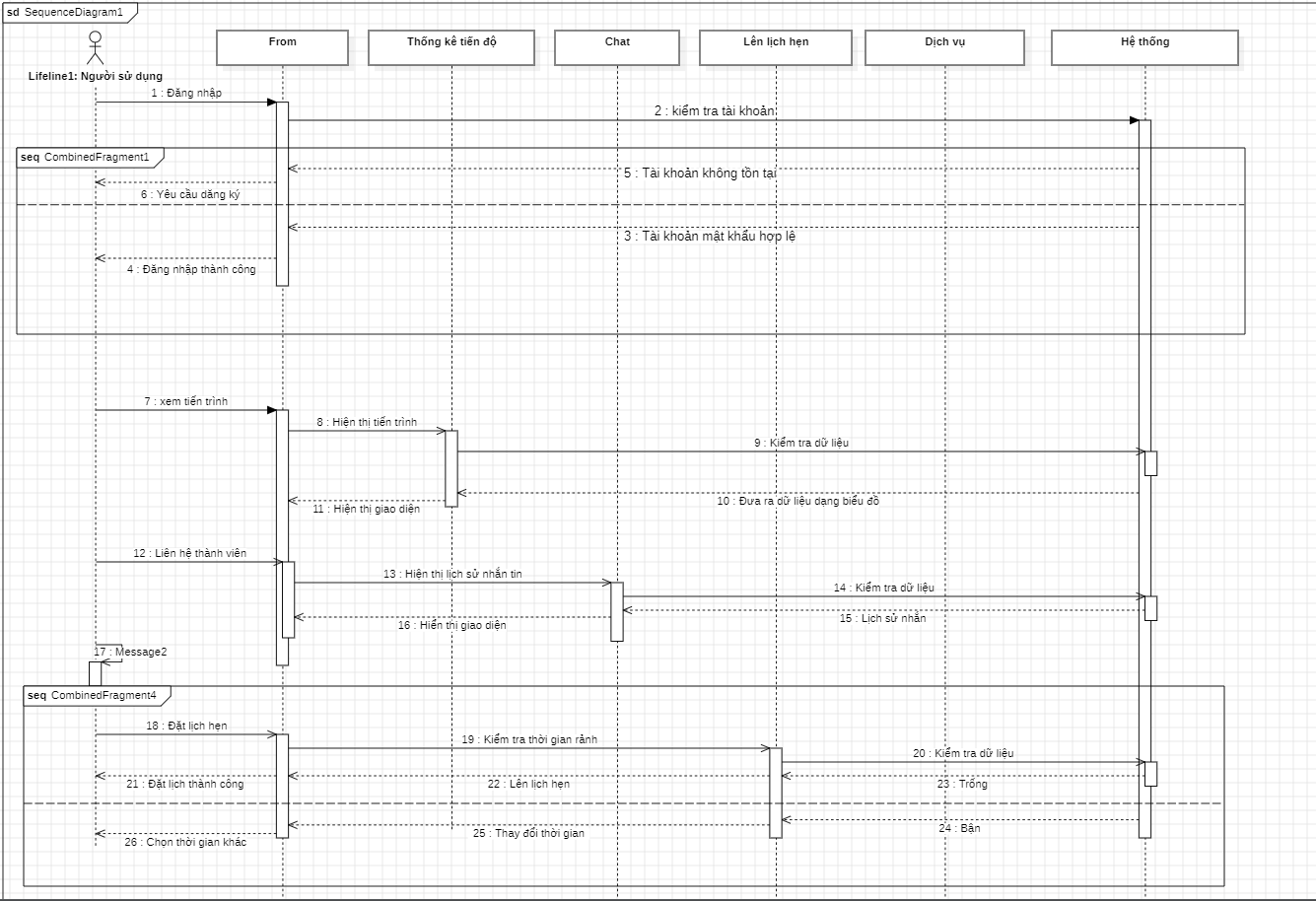
## Sơ đồ lớp ( Class Diagram):



## Quy trình nghiệp vụ:



## Sơ đồ tuần tự:



# Yêu cầu phi chức năng.

## Yêu cầu mức độ tin cậy.

Hệ thống phải hoạt động ổn định và đáng tin cây.

## Yêu cầu bảo mật

Dữ liệu người dùng phải bảo vệ chặt chẽ.

## Yêu cầu giao diện

Giao diện người dùng thân thiện và dễ sử dụng.

## Yêu cầu hiệu suất

Hệ thống phải đạt hiệu suất cao, đáp ứng nhanh chóng mọi tình huống

## Yêu cầu khả năng hỗ trợ và bảo trì

Cần có các tiêu chuẩn về lập trình, quy ước đặt tên và tài liệu đẻ bảo trì hệ thống này.

## Yêu cầu về môi trường

Hệ thống phải hoạt động tốt trong mọi môi trường, bảo trì và khôi phục dễ dàng

## Yêu cầu hệ thống tài liệu trực tuyến và hệ thống trợ giúp

Cần có hệ thống trực tuyến và trợ giúp người dùng dễ dàng tìm kiếm thông tin và hỗ trợ khi cần

## Thành phần mua ngoài

Cần xác đinh rõ thành phần mua ngoài được sử dụng và các giới hạn, giấy phép và tiêu chuẩn tương thích.

## Yêu cầu pháp lý, bản quyền và các lưu ý khác

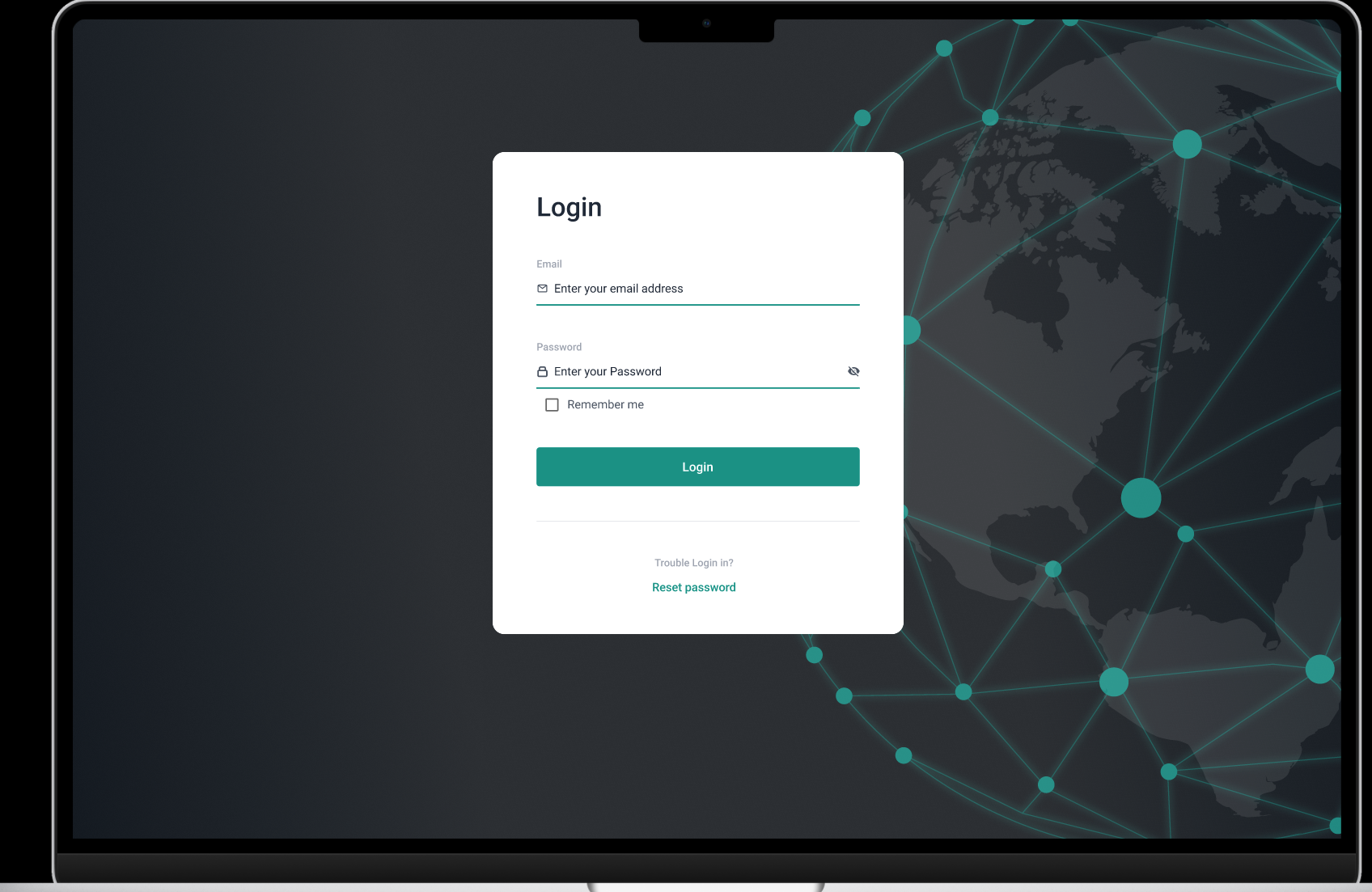
Cần tuân thủ các quy định pháp lý, bản quyền và các lưu ý khác liên quan đến hệ thống phần mền

## Tiêu chuẩn áp dụng

* ISO/IEC 27001:2013: Đảm bảo an ninh thông tin trong quá trình phát triển và vận hành hệ thống, bao gồm quản lý rủi ro, bảo vệ dữ liệu và hệ thống.
* OWASP Top 10: Đối với các ứng dụng web, tuân thủ các tiêu chuẩn để bảo vệ khỏi các lỗ hổng bảo mật phổ biến như injection, cross-site scripting (XSS),…
* GDPR (General Data Protection Regulation): Bảo vệ dữ liệu cá nhân của người dùng và tuân thủ các quy định về việc thu thập, xử lý và bảo mật thông tin cá nhân.

# Đặc tả giao diện UI/UX.

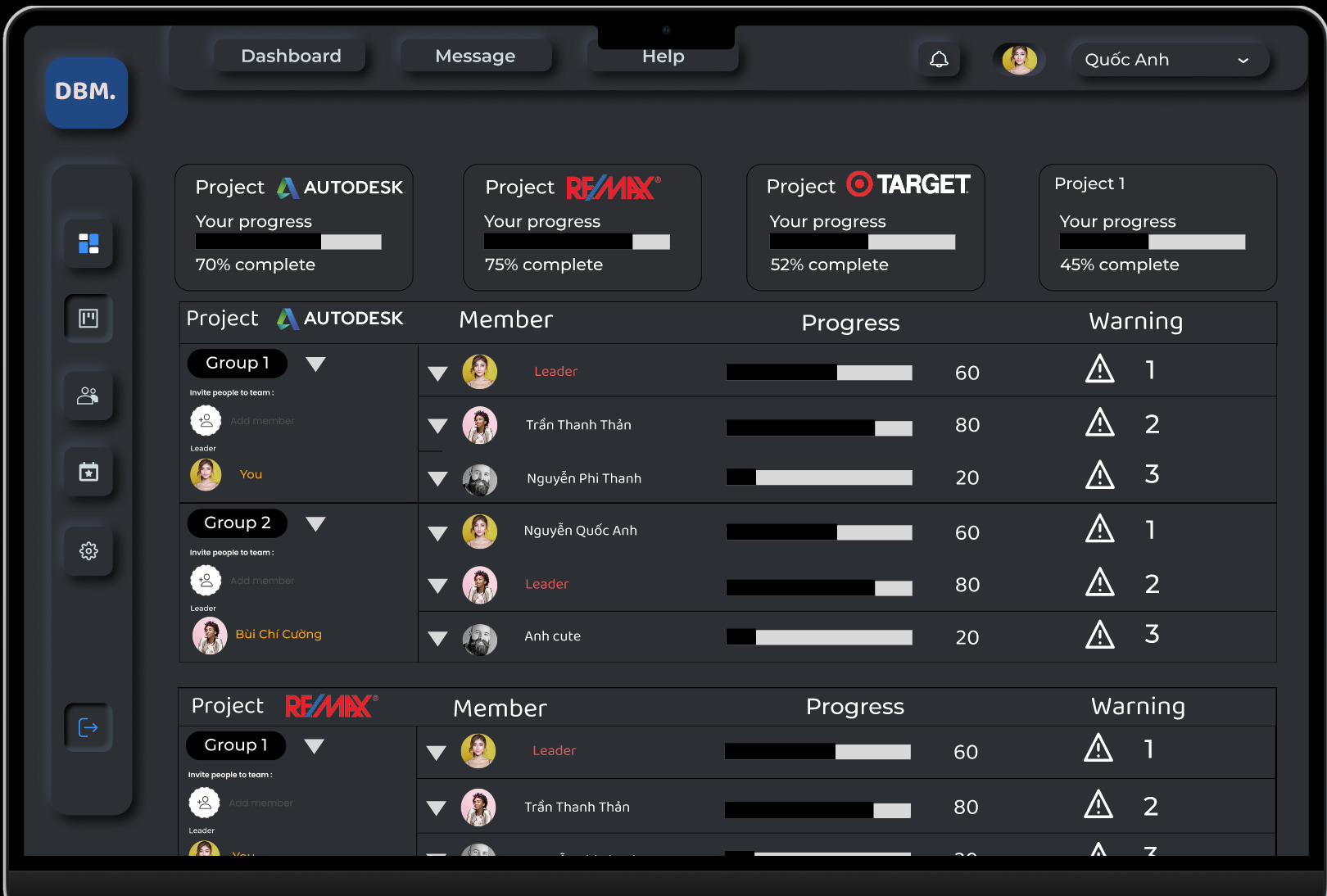
* Màn hình hiển thị From login: Dùng để xác thực người dùng.



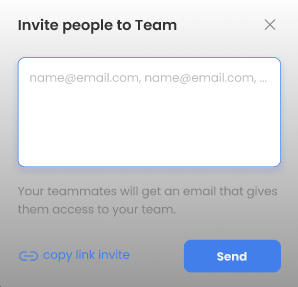
* Màn hình giao diện chính của chương trình:
* Tạo dự án (Add Project): Thêm dự án mới và thêm nhóm thực hiện.
* Xem tiến độ làm việc của các nhóm.
* Xem thông tin hoạt động của các trưởng nhóm.
* Thực hiện các tác vụ khác.



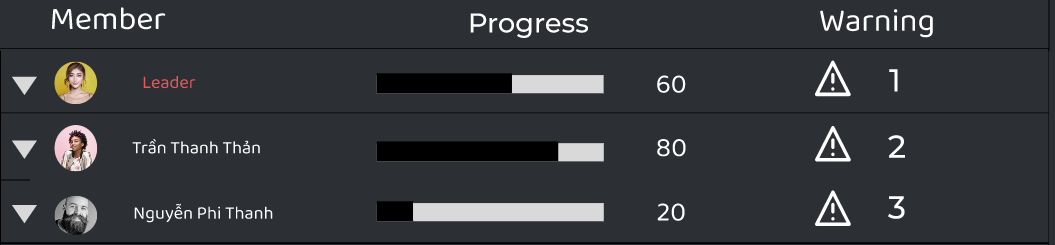
* Màn hình xem tiến độ dự án và cảnh báo tiến độ các thành viên của các nhóm:



* Thêm thành viên vào nhóm bằng cách nhập email người dùng.



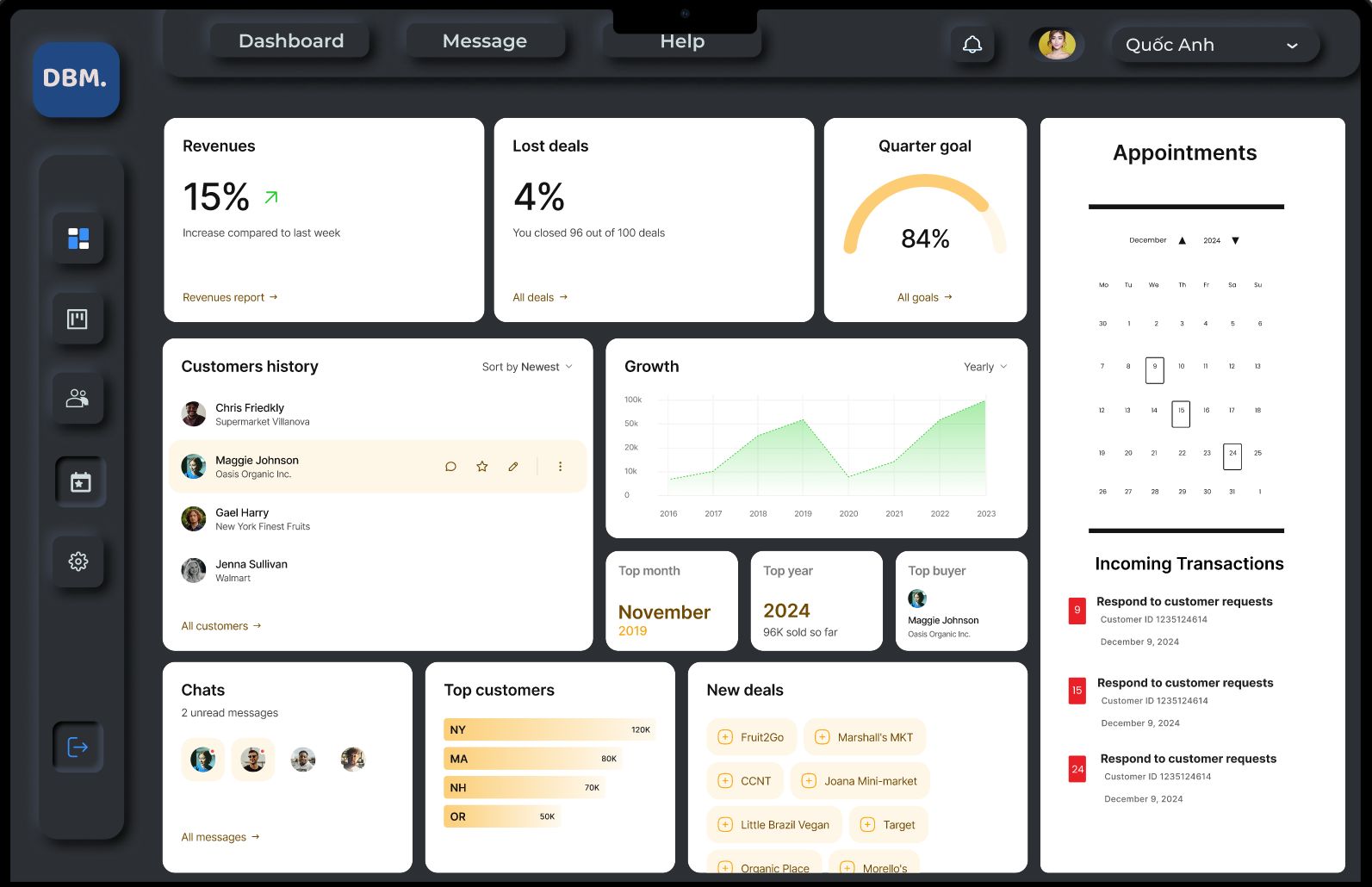
* Thông tin tiến độ chi tiết các thành viên trong nhóm và cảnh báo tiến độ hoàn thành chi tiết của các thành viên.



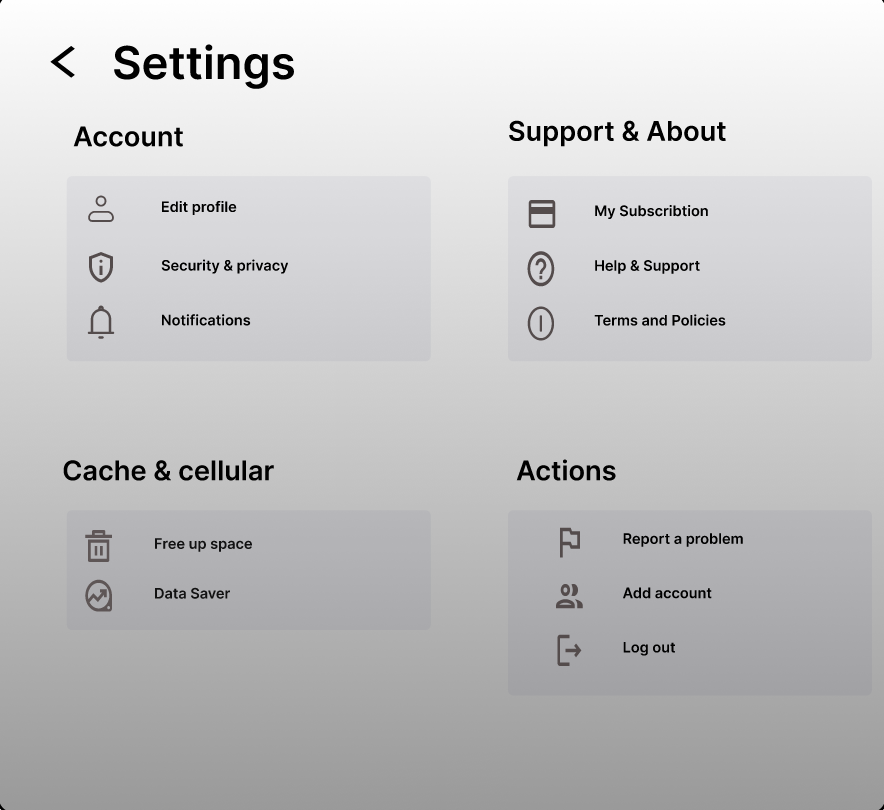
* Thông tin phần trăm hoàn thành công việc của các nhóm và từng thành viên, thông tin doanh thu cá nhân tần xuất làm việc.



* Đặt lịch hẹn với khách hàng



* Cài đặt chi tiết hệ thống.



# UNIT TEST.

## Khái niệm Unit Test.

- Unit Test là một loại kiểm thử phần mềm trong đó các đơn vị hay thành phần riêng lẻ của phần mềm được kiểm thử. Kiểm thử đơn vị được thực hiện trong quá trình phát triển ứng dụng. Mục tiêu của Kiểm thử đơn vị là cô lập một phần code và xác minh tính chính xác của đơn vị đó.

## Khái niệm Unit.

- Một Unit là một thành phần phần mềm nhỏ nhất mà ta có thể kiểm tra được như các hàm (Function), thủ tục (Procedure), lớp (Class), hoặc các phương thức (Method).

- Vì Unit được chọn để kiểm tra thường có kích thước nhỏ và chức năng hoạt động đơn giản, chúng ta không khó khăn gì trong việc tổ chức, kiểm tra, ghi nhận và phân tích kết quả kiểm tra nên việc phát hiện lỗi sẽ dễ dàng xác định nguyên nhân và khắc phục cũng tương đối dễ dàng vì chỉ khoanh vùng trong một Unit đang kiểm tra.

Mỗi UT sẽ gửi đi một thông điệp và kiểm tra câu trả lời nhận được đúng hay không, bao gồm:

* Các kết quả trả về mong muốn
* Các lỗi ngoại lệ mong muốn

- Các đoạn mã UT hoạt động liên tục hoặc định kỳ để thăm dò và phát hiện các lỗi kỹ thuật trong suốt quá trình phát triển, do đó UT còn được gọi là kỹ thuật kiểm nghiệm tự động. UT có các đặc điểm sau:

* Đóng vai trò như những người sử dụng đầu tiên của hệ thống.
* Chỉ có giá trị khi chúng có thể phát hiện các vấn đề tiềm ẩn hoặc lỗi kỹ thuật.

## Công cụ thực hiện Unit test.

- Có nhiều công cụ thực hiện Unit Test như:

1. [Junit](https://junit.org/junit5/?ref=200lab.io): Junit là công cụ miễn phí được sử dụng cho ngôn ngữ lập trình Java. Nó cung cấp assertions để xác định phương pháp kiểm tra. Đầu tiên, công cụ này kiểm tra dữ liệu và sau đó chèn vào đoạn code.
2. [NUnit](https://nunit.org/?ref=200lab.io): NUnit là framework unit-testing được sử dụng rộng rãi cho tất cả các ngôn ngữ .NET. Nó là công cụ mã nguồn mở, có thể viết script theo cách thủ công. Ngoài ra, nó còn hỗ trợ các bài kiểm tra data-driven có thể chạy song song.
3. [JMockit](https://jmockit.github.io/index.html?ref=200lab.io): JMockit là công cụ Unit testing mã nguồn mở. Nó là công cụ code coverage với các số liệu đường (line) và đường dẫn (path). Công cụ này cung cấp Line coverage, Path Coverage, và Data Coverage.
4. [EMMA](http://emma.sourceforge.net/?ref=200lab.io): EMMA là một toolkit mã nguồn mở để phân tích và báo cáo code được viết bằng ngôn ngữ Java. Emma hỗ trợ các loại coverage như method, line, basic block. Nó dựa trên Java nên không có dependencies thư viện bên ngoài và có thể truy cập source code.
5. [PHPUnit](https://phpunit.de/?ref=200lab.io): PHPUnit là công cụ unit testing dành cho lập trình viên PHP. Nó sử dụng các phần nhỏ của code được gọi là các đơn vị và kiểm tra từng phần trong số chúng một cách riêng biệt. Công cụ này cũng cho phép các dev sử dụng các phương pháp pre-define assertion để khẳng định hệ thống hoạt động theo cách nhất định.

## Các thuật ngữ khi sử dụng Unit Test.

1. **Assertion**: Là một phát biểu mô tả các công việc kiểm tra cần tiến hành, thí dụ: AreEqual(), IsTrue(), IsNotNull()… Mỗi một UT gồm nhiều assertion kiểm tra dữ liệu đầu ra, tính chính xác của các lỗi ngoại lệ ra và các vấn đề phức tạp khác như: – Sự tồn tại của một đối tượng – Điều kiện biên: Các giá trị có vượt ra ngoài giới hạn hay không – Thứ tự thực hiện của các luồng dữ liệu …
2. **Test Point**: Là một đơn vị kiểm tra nhỏ nhất, chỉ chứa đơn giản một assertion nhằm khẳng định tính đúng đắn của một chi tiết mã nào đó. Mọi thành viên dự án đều có thể viết một test point. Test Case: Là một tập hợp các test point nhằm kiểm tra một đặc điểm chức năng cụ thể, thí dụ toàn bộ giai đoạn người dùng nhập dữ liệu cho đến khi thông tin được nhập vào cơ sở dữ liệu. Trong nhiều trường hợp kiểm tra đặc biệt và khẩn cấp có thể không cần đến test case.
3. **Test Suite**: Là một tập hợp các test case định nghĩa cho từng module hoặc hệ thống con.
4. **Regression Testing (hoặc Automated Testing)**: Là phương pháp kiểm nghiệm tự động sử dụng một phần mềm đặc biệt. Cùng một loại dữ liệu kiểm tra giống nhau nhưng được tiến hành nhiều lần lặp lại tự động nhằm ngăn chặn các lỗi cũ phát sinh trở lại. Kết hợp Regression Testing với Unit Testing sẽ đảm bảo các đoạn mã mới vẫn đáp ứng yêu cầu thay đổi và các đoạn mã cũ sẽ không bị ảnh hưởng bởi các hoạt động bảo trì.
5. **Production Code**: Phần mã chính của ứng dụng được chuyển giao cho khách hàng.
6. **Unit Testing Code**: Phần mã phụ để kiểm tra mã ứng dụng chính, không được chuyển giao cho khách hàng.

## Vòng đời của Unit Test.

UT test có 3 trạng thái cơ bản

* Fail (trạng thái lỗi)
* Ignore (tạm ngừng thực hiện)
* Pass (trạng thái làm việc)
* Toàn bộ UT được vận hành trong một hệ thống tách biệt. Thông thường, trạng thái của UT được biểu hiện bằng các màu khác nhau: màu xanh (pass), màu vàng (ignore) và màu đỏ (fail).

UT hoạt động hiệu quả khi

* Được vận hành lặp nhiều lần.
* Tự động hoàn toàn.
* Độc lập với UT khác.
* Test case đơn giản dễ đọc, dễ bảo trì.
* Sử dụng data dễ hiểu, dễ đọc.

## Ưu điểm và nhược điểm của Unit Test.

*Ưu điểm của Unit Testing bao gồm:*

* Chúng ta có thể kiểm thử từng phần riêng rẽ mà không phải đợi các phần khác hoàn thành.
* Vấn đề được phát hiện càng sớm thì càng ít lỗi phức hợp xảy ra.
* Quá trình debug được thực hiện dễ dàng hơn.
* Các dev có thể nhanh chóng thực hiện những thay đổi đối với code base.
* Các dev cũng có thể sử dụng lại code, chuyển nó sang các dự án mới.
* Các lỗi được phát hiện sớm sẽ giúp giảm chi phí sửa lỗi. Bởi vì chi phí sửa lỗi trong giai đoạn **unit test**sẽ ít hơn so với những giai đoạn sau.

*Nhược điểm của Unit Testing gồm:*

* Các bài test sẽ không phát hiện ra tất cả các bug.
* Unit tests chỉ kiểm tra các tập dữ liệu và chức năng của nó. Chứ unit tests không tìm ra các lỗi tích hợp.
* Có thể cần viết nhiều dòng test code hơn để kiểm tra một dòng code.
* Unit testing cần đầu tư thời gian, công sức để học tập. Bởi vì đôi khi bạn cần học cách sử dụng các công cụ phần mềm tự động.

## Thiết kế Unit test

Mỗi UT đều được tiết kế theo trình tự sau:

* Thiết lập các điều kiện cần thiết: khởi tạo các đối tượng, xác định tài nguyên cần thiết, xây dựng các dữ liệu giả…
* Triệu gọi các phương thức cần kiểm tra.
* Kiểm tra sự hoạt động đúng đắn của các phương thức.
* Dọn dẹp tài nguyên sau khi kết thúc kiểm tra.

# CODE STYLE JAVA

## Khái niệm:

Tiêu chuẩn coding (Coding Standards) là một bộ quy tắc quy định cách viết code của một chương trình mà lập trình viên phải tuân theo khi tham gia vào dự án phát triển chương trình đó. Tùy theo mỗi dự án mà sẽ có những tiêu chuẩn khác nhau, bộ quy tắc đó bao gồm:

* Đặt tên lớp, interface, tên biến, phương thức, …
* Khoảng trắng, tab
* Khai báo và sử dụng biến
* Comment mã nguồn: tên người tạo, phiên bản, ngày tạo file, lớp, phương thức, người thay đổi, nội dung thay đổi, …
* Độ dài tối đa mỗi dòng code, mỗi file, …

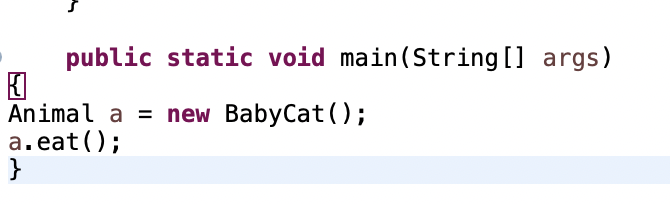
## Mục đích của viết code theo quy chuẩn quy định.

* Dễ fix bug, sửa code.
* Để người đến sau có thể kế thừa và phát triển code của chính bản thân.
* Thống nhất được code của những thành viên trong cùng 1 dự án.
* Nếu là người duy nhất hiểu được code mình thì khi gặp trở ngại người khác sẽ khó để giúp đỡ.

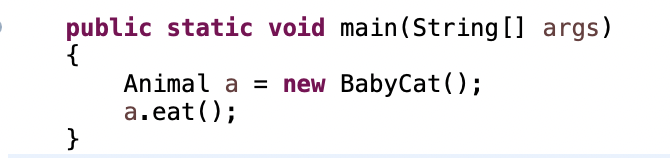
## Chuẩn hình thức và ngữ nghĩa.

### Chuẩn hình thức .

* Thụt đầu dòng.



Sai



Đúng

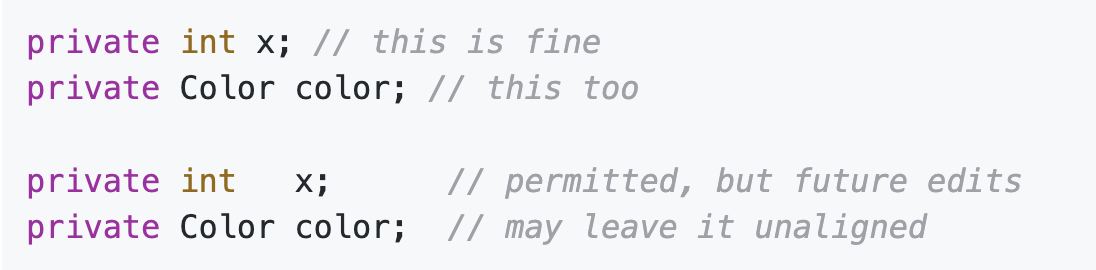
* Sử dụng khoảng trắng.



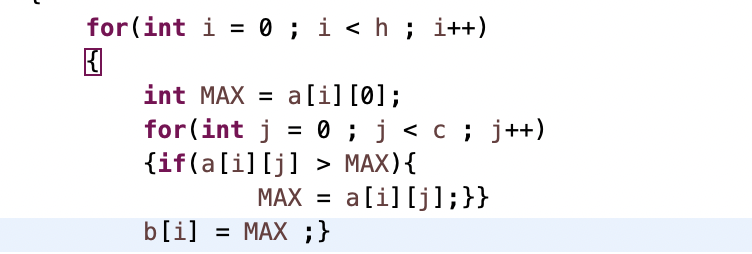
Sai

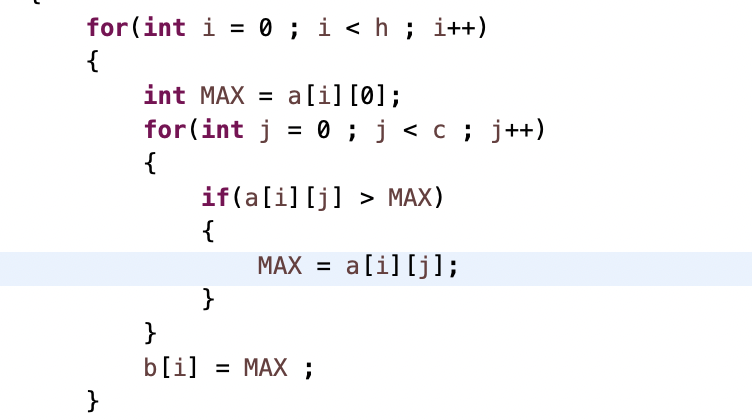


Đúng

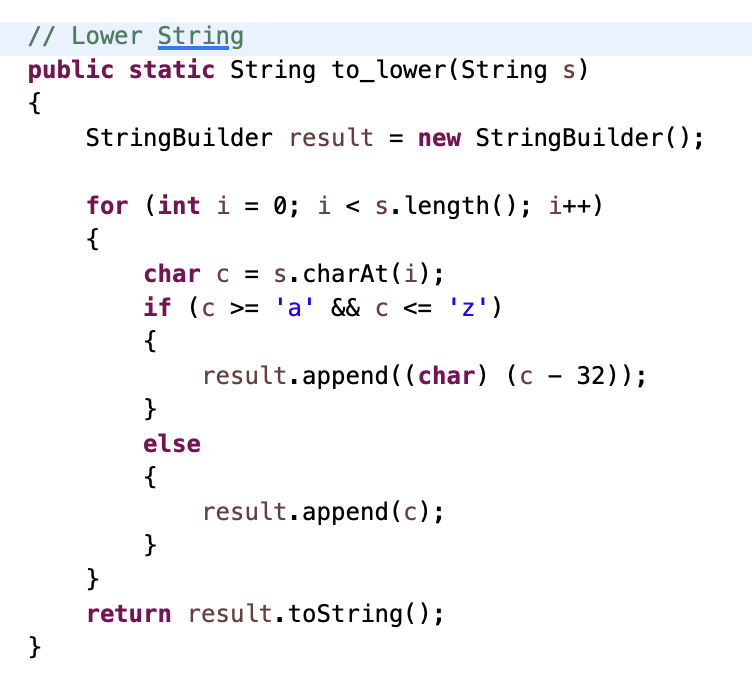


* Đóng ngoặc, mở ngoặc.

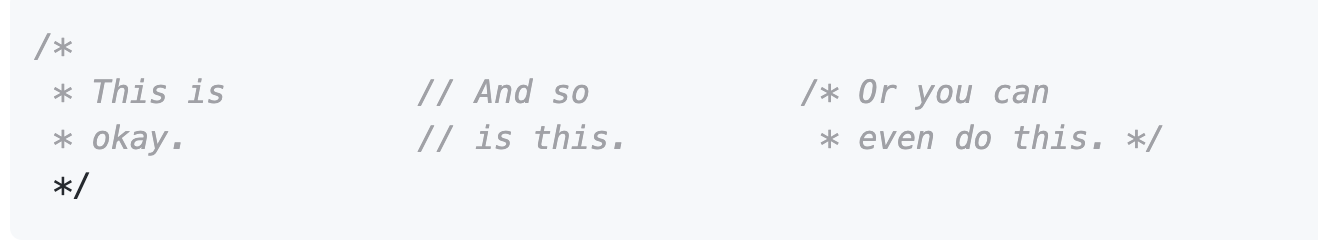




* Đặt tên lớp, thuộc tính, phương thức (đặt tên phụ hợp với ngữ cảnh ngữ nghĩa).
* Phụ đề cho hàm.



* While Space (dòng trống) : Các dòng code có quan hệ với nhau gom lại thành 1 khối, 2 dòng code có mối quan hệ khác nhau thì có 1 While Space để dễ phân biệt.
* Quy ước đặt tên : các chữ cái mỗi từ sẽ ghi hoa ( Flower not flower).
* Các khối comments: Các khối comment được thụt lề cùng cấp với code quanh nó. Có thể là dạng /\* ... \* */, // ....,* và các dòng tiếp theo phải bắt đầu bằng dấu \* thẳng hành với dấy \* ở dòng trước.



* Đặt tên class, interface, abstract class: hạn chế viết tắt và đặt tên khó hiểu, tên package đặt hoàn toàn bằng chữ thường.
* Hạn chế 1 dòng code quá nhiều kí tự (tối đa 80 là hợp lý nhất).

### Chuẩn ngữ nghĩa.

* Khai báo và sử dụng biến:
* Một biến cho mỗi khai báo
  + Mọi khai báo biến (trường hoạc cục bộ) chỉ khai báo một biến (từng biến một, dù rằng chung kiểu dữ liệu). Không sử dụng int a, b; (mặc dù không sai).
  + Ngoại trừ nhiều khai báo biến được chấp nhận trong vòng lặp for
* Khai báo khi cần đến
  + Các biến cục bộ không được khai báo theo thói quen đầu khối chứa chúng. Thay vào đó, chúng được khai báo gần với điểm chúng được sử dụng, để giảm thiểu phạm vi của nó.
* Cài đặt phương thức:
  + Khi một đoạn code xuất hiện ở nhiều nơi trong chương trình ta gom các đoạn code đó thành một phương thức: Tiết kiệm thời gian bảo trì, sửa lỗi.
  + Khi trong một phương thức có các đoạn code xử lý phức tạp thì ta nên tách đoạn code phức tạp đó ra thành một phương thức riêng biệt: Dễ dàng theo dõi, debug.
  + Khai báo tham số truyền vào vừa đủ, tránh tình trạng khai báo tham số truyền vào nhưng không sử dụng.
  + Mỗi phương thức chỉ thực hiện một chức năng.
  + Kích thước của một phương thức: Nhiều thí nghiệm cho thấy một phương thức có khoảng từ 50 đến 150 dòng code là hợp lý.