ĐẶC TẢ YÊU CẦU PHẦN MỀM

HỆ THỐNG QUẢN LÝ NHÀ Ở CHO SINH VIÊN

**Phiên bản: 1.0**

**Mã dự án: 01**

**Mã tài liệu: 01\_SRS\_1.0.doc**

**Sinh viên thực hiện:**

**Phan Thanh Sum MSSV: 102220039**

**Nguyễn Hữu Minh Vũ MSSV: 102220049**

**MỤC LỤC**

[**1.** **Giới thiệu** 5](#_Toc165948826)

[**1.1** **Tổng quan** 5](#_Toc165948827)

[**1.2** **Mục đích** 5](#_Toc165948828)

[**1.3** **Phạm vi** 5](#_Toc165948829)

[**1.4** **Thuật ngữ viết tắt** 5](#_Toc165948830)

[**1.5** **Tài liệu tham khảo** 5](#_Toc165948831)

[**2.** **Mô tả tổng quan** 6](#_Toc165948832)

[**2.1.** **Mô hình các tình huống sử dụng tổng quan** 6](#_Toc165948833)

[**2.2.** **Kiến trúc tổng thể phần mềm** 6](#_Toc165948834)

[**3.** **Đặc tả yêu cầu chức năng** 8](#_Toc165948835)

[**3.1** **Các tác nhân** 8](#_Toc165948836)

[**3.2** **Các chức năng của hệ thống** 8](#_Toc165948837)

[**3.3** **Biểu đồ use case tổng quan** 8](#_Toc165948838)

[**3.4** **Biểu đồ use case phân rã** 9](#_Toc165948839)

[**3.5** **Quy trình nghiệp vụ** 10](#_Toc165948840)

[**3.6** **Đặc tả các use case** 12](#_Toc165948841)

[**4.** **Các yêu cầu phi chức năng** 20](#_Toc165948842)

[**4.1** **Giao diện người dùng** 20](#_Toc165948843)

[**4.2** **Tính bảo mật** 20](#_Toc165948844)

[**4.3** **Ràng buộc** 20](#_Toc165948845)

[**5.** **Code style** 21](#_Toc165948846)

[**5.1** **White space** 21](#_Toc165948847)

[**5.2** **Comment** 21](#_Toc165948848)

[**5.3** **Quy tắc đặt tên** 21](#_Toc165948849)

[**6.** **Unit test** 24](#_Toc165948850)

[**6.1** **Khái niệm về unit test** 24](#_Toc165948851)

[**6.2** **Vòng đời Unit Test** 24](#_Toc165948852)

[**6.3** **Thiết kế Unit Test** 25](#_Toc165948853)

[**6.4** **Ứng dụng Unit Test** 25](#_Toc165948854)

[**6.5** **Lợi ích của việc áp dụng Unit Test** 25](#_Toc165948855)

[**6.6** **Cách code hiệu quả với Unit Test** 26](#_Toc165948856)

**DANH MỤC HÌNH ẢNH, BẢNG BIỂU**

[**Hình 1:Kiến trúc 3 lớp phần mềm** 5](#_Toc167031064)

[**Hình 2: Kiến trúc Client Server phần mềm** 5](#_Toc167031065)

[**Hình 3: Biểu đồ use case tổng quan** 7](#_Toc167031066)

[**Hình 4: Biểu đồ use case phân rã Sinh viên** 8](#_Toc167031067)

[**Hình 5: Biểu đồ use case phân rã chủ trọ** 8](#_Toc167031068)

[**Hình 6: Biểu đồ use case phân rã quản trị viên** 9](#_Toc167031069)

[**Hình 7: Biểu đồ hoạt động sử dụng phần mềm** 9](#_Toc167031070)

[**Hình 8: Biểu đồ hoạt động thuê trọ** 10](#_Toc167031071)

[**Hình 9: Biều đồ hoạt động quản lý nhà trọ** 11](#_Toc167031072)

[**Hình 10: Biểu đồ hoạt động quản lý phòng trọ đang thuê** 12](#_Toc167031073)

[**Hình 11: Biểu đồ hoạt động sử dụng phần mềm của sinh viên** 13](#_Toc167031074)

[**Hình 12: Biểu đồ hoạt động quản lý câu hỏi thường gặp** 14](#_Toc167031075)

[**Hình 13: Sơ đồ lớp phân tích** 23](#_Toc167031076)

[**Hình 14:Sơ đồ lớp thiết kế** 23](#_Toc167031077)

# **Giới thiệu**

## **Tổng quan**

Tài liệu này được viết dựa theo chuẩn của Tài liệu đặc tả yêu cầu phần mềm (Software Requirements Specifications - SRS) được giải thích trong "IEEE Recommended Practice for Software Requirements Specifications" và " IEEE Guide for Developing System Requirements Specifications".

Với cấu trúc được chia làm ba phần:

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

2. 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 triển phần mềm đáp ứng được các yêu cầu đó.

3. Phần 3: Các yêu cầu phi chức năng.

## **Mục đích**

Mục đích của tài liệu đặc tả yêu cầu phần mềm này là cung cấp một cái nhìn tổng quan, dễ hiểu về các yêu cầu, thành phần của dự án, phục vụ cho những nhà phát triển phần mềm, kiểm thử viên, nhà quản lý dự án cũng như các bên liên quan.

## **Phạm vi**

Tài liệu đặc tả yêu cầu phần mềm này được xây dựng nhằm phục vụ cho dự án Hệ thống quản lý nhà ở cho sinh viên phục vụ cho các bạn sinh viên tìm kiếm nhà trọ và các chủ trọ có thể quảng bá và quản lý nhà trọ của mình.

Với việc phát triển hệ thống phần mềm như thế này cung cấp cho sinh viên và các chủ trọ dễ dàng quản lý nhà ở hơn so với cách truyền thống. Mang lại nhiều lợi ích về kinh tế với chi phí thấp nhưng vẫn đảm bảo được nhu cầu cũng như chất lượng.

## **Thuật ngữ viết tắt**

## **Tài liệu tham khảo**

# **Mô tả tổng quan**

• Tổng quan sản phẩm: Sản phẩm là một nền tảng mạng xã hội dành cho việc quản lý nhà ở của sinh viên và chủ trọ. Là một cộng đồng nơi mà các sinh viên và chủ trọ có thể kết nối, trao đổi thông tin và tìm kiếm cơ hội thuê nhà hoặc cho thuê nhà một cách dễ dàng.

• Các chức năng của sản phẩm: Thuê trọ và cho thuê trọ, quản lý nhà trọ, phòng trọ.

• Các đặc điểm của người dùng: Các cá nhân và các tổ chức doanh nghiệp.

• Các ràng buộc: không.

• Giả thiết và các phụ thuộc: không.

## **Kiến trúc tổng thể phần mềm**

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

**Hình 1:Kiến trúc 3 lớp phần mềm**

Ảnh có chứa biểu đồ, hàng, thiết kế

Mô tả được tạo tự động

**Hình 2: Kiến trúc Client Server phần mềm**

# **Đặc tả yêu cầu chức năng**

## **Các tác nhân**

Hệ thống gồm có các tác nhân là Khách, Sinh viên, Chủ trọ và Quản trị viên. Khách có vai trò là người dùng khi chưa đăng nhập vào hệ thống. Sinh viên khi đăng nhập vào hệ thống. Chủ trọ được cung cấp các chức năng quản lý nhà trọ của mình. Quản trị viên có vai trò quản trị hoạt động của hệ thống.

## **Các chức năng của hệ thống**

1 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.

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

3. Thuê trọ: Chức năng giúp các sinh viên đăng kí thuê nhà trọ theo nhu cầu.

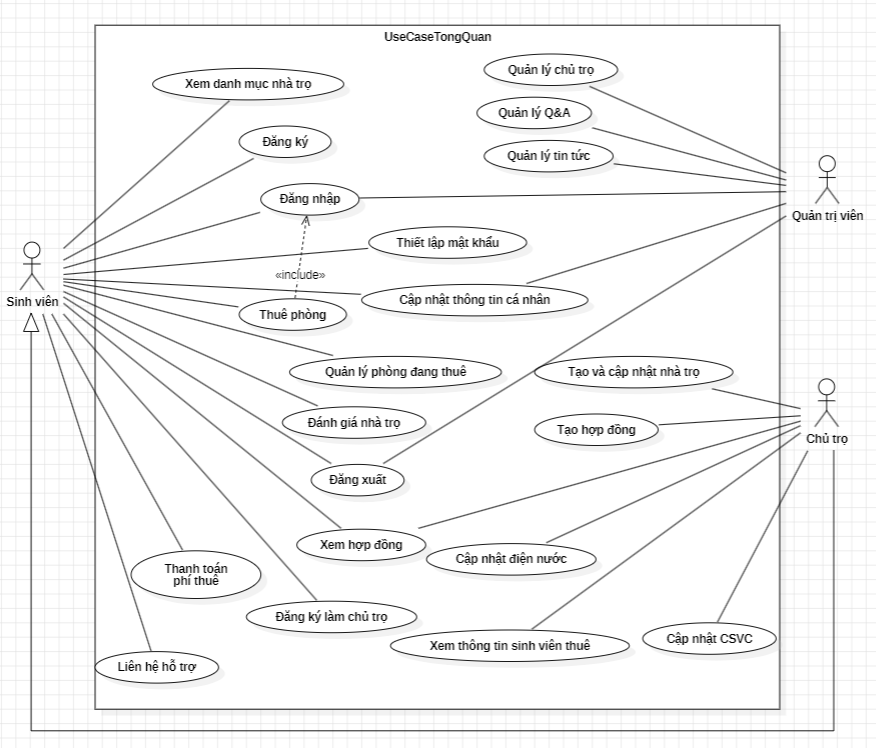
4. Cho thuê trọ: Chức năng giúp các chủ nhà trọ có nhu cầu cho thuê đăng bài quảng bá.

5. Quản lý nhà trọ: Chức năng giúp chủ nhà trọ quản lý hoạt động nhà trọ của mình, và sinh viên theo dõi tình hình của phòng mình đang thuê.

6. Quản trị người dùng: Quản trị viên có vai trò quản trị những người dùng trong hệ thống và phê duyệt các đăng kí làm chủ trọ.

7. Các nhóm chức năng khác: xử lý đánh giá bình luận, xử lý quy trình thanh toán,…

## **Biểu đồ use case tổng quan**



**Hình 3: Biểu đồ use case tổng quan**

## **Biểu đồ use case phân rã**

* + 1. **Biểu đồ use case phân rã sinh viên**

Ảnh có chứa văn bản, biểu đồ, hàng, mẫu

Mô tả được tạo tự động

**Hình 4: Biểu đồ use case phân rã Sinh viên**

* + 1. **Biểu đồ use case phân rã chủ trọ**

Ảnh có chứa văn bản, biểu đồ, hàng, ảnh chụp màn hình

Mô tả được tạo tự động

**Hình 5: Biểu đồ use case phân rã chủ trọ**

* + 1. **Biểu đồ use case phân rã quản trị viên**

Ảnh có chứa văn bản, biểu đồ, ảnh chụp màn hình, hàng

Mô tả được tạo tự động

**Hình 6: Biểu đồ use case phân rã quản trị viên**

## **Quy trình nghiệp vụ**

* + 1. **Quy trình sử dụng phần mềm**

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, số

Mô tả được tạo tự động

**Hình 7: Biểu đồ hoạt động sử dụng phần mềm**

* + 1. **Quy trình thuê trọ**

Ảnh có chứa văn bản, biểu đồ, Kế hoạch, bản đồ

Mô tả được tạo tự động

**Hình 8: Biểu đồ hoạt động thuê trọ**

* + 1. **Quy trình quản lý cho thuê trọ**

**Ảnh có chứa văn bản, biểu đồ, Kế hoạch, sơ đồ

Mô tả được tạo tự động**

**Hình 9: Biều đồ hoạt động quản lý nhà trọ**

* + 1. **Quy trình quản lý phòng đang thuê**

**Ảnh có chứa biểu đồ, văn bản, hàng, Kế hoạch

Mô tả được tạo tự động**

**Hình 10: Biểu đồ hoạt động quản lý phòng trọ đang thuê**

* + 1. **Quy trình sử dụng phần mềm của sinh viên**

**Ảnh có chứa văn bản, biểu đồ, Kế hoạch, hàng

Mô tả được tạo tự động**

**Hình 11: Biểu đồ hoạt động sử dụng phần mềm của sinh viên**

* + 1. **Quy trình quản lý câu hỏi thường gặp**

**Ảnh có chứa văn bản, biểu đồ, Song song, hàng

Mô tả được tạo tự động**

**Hình 12: Biểu đồ hoạt động quản lý câu hỏi thường gặp**

## **Đặc tả các use case**

* + 1. **Đăng nhập**

|  |  |
| --- | --- |
| Tên Use Case: Đăng nhập | |
| Tác nhân | Khách |
| Mô tả | Tác nhân đăng nhập vào hệ thống để sử dụng các chức năng |
| Sự kiện kích hoạt | Click vào nút đăng nhập trên hệ thống |
| Tiền điều kiện | Tác nhân đã có tài khoản trên hệ thống |
| Luồng sự kiện chính  (Thành công | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | Khách | Chọn chức năng đăng nhập | | 2 | Hệ thống | Hiển thị giao diện đăng nhập | | 3 | Khách | Nhập tài khoản và mật khẩu | | 4 | Khách | Yêu cầu đăng nhập | | 5 | Hệ thống | Kiểm tra thông tin khách nhập đã đầy đủ và có trong hệ thống hay chưa | | 6 | Hệ thống | HIển thị chức năng tương ứng với Người dùng | |
| Luồng sự kiện thay thế | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 5a | Hệ thống | Thông báo lỗi: Cần nhập đầy đủ tài khoản mật khẩu nếu khách nhập thiếu | | 5b | Hệ thống | Thông báo lỗi: Không tìm thấy tài khoản của khách nếu không tìm thấy tài khoản khách trong hệ thống | |
| Hậu điều kiện | Tác nhân đăng nhập được vào hệ thống |

\*Dữ liệu đầu vào

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Trường dữ liệu | Mô tả | Bắt buộc | Điều kiện | Ví dụ |
| 1 | Email | Email Field | Có | Đúng định dạng email | test@gmail.com |
| 2 | Mật khẩu | Password Field | Có | Tối thiểu 6 kí tự | 123456 |

* + 1. **Đăng kí**

|  |  |
| --- | --- |
| Tên Use Case: Đăng ký | |
| Tác nhân | Khách |
| Mô tả | Tác nhân đăng ký vào hệ thống để sử dụng các chức năng |
| Sự kiện kích hoạt | Click vào nút đăng ký trên hệ thống |
| Tiền điều kiện | Tác nhân chưa có tài khoản |
| Luồng sự kiện chính  (Thành công | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | Khách | Chọn chức năng đăng ký | | 2 | Hệ thống | Hiển thị giao diện đăng ký | | 3 | Khách | Nhập thông tin theo yêu cầu hệ thống | | 4 | Khách | Yêu cầu đăng ký | | 5 | Hệ thống | Kiểm tra thông tin khách nhập đã đầy đủ các trường bắt buộc chưa | | 6 | Hệ thống | Kiểm tra địa chỉ email có hợp lệ không | | 7 | Hệ thống | Kiểm tra mật khẩu nhập lại có trùng mật khẩu trước hay không | | 8 | Hệ thống | Kiểm tra mức độ bảo mật của mật khẩu | | 9 | Hệ thống | Gửi xác thực qua email đăng ký | | 10 | Khách | Xác thực tài khoản qua email | | 11 | Hệ thống | Thông báo đăng ký thành công và lưu tài khoản vào hệ thống | |
| Luồng sự kiện thay thế | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 5a | Hệ thống | Thông báo lỗi: Cần nhập đầy đủ các trường bắt buộc | | 6a | Hệ thống | Thông báo lỗi: Nhập địa chỉ email không hợp lệ | | 7a | Hệ thống | Thông báo lỗi: Mật khẩu xác thực không chính xác | | 8a | Hệ thống | Thông báo lỗi: Đặt lại mật khẩu do mật khẩu không đủ mức độ an toàn | | 11.a | Hệ thống | Không thông báo đăng ký thành công nếu khách không xác nhận tài khoản qua email | |
| Hậu điều kiện | Tác nhân đăng ký được tài khoản và được hệ thống lưu lại |

\*Dữ liệu đầu vào

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Trường dữ liệu | Mô tả | Bắt buộc | Điều kiện | Ví dụ |
| 1 | Email | Email Field | Có | Đúng định dạng email | test@gmail.com |
| 2 | Mật khẩu | Password Field | Có | Tối thiểu 6 kí tự | 123456@ |
| 3 | Xác thực mật khẩu | Password Field | Có | Trùng với mật khẩu | 123456@ |
| 4 | Điện thoại | TextField | Có | Kí tự số | 0987666554 |

* + 1. **Cập nhật thông tin cá nhân**

|  |  |
| --- | --- |
| Tên Use Case: Cập nhật thông tin cá nhân | |
| Tác nhân | Sinh viên, Chủ trọ (Người dùng) |
| Mô tả | Cập nhật thông tin cá nhân của người dùng |
| Sự kiện kích hoạt | Click thông tin cá nhân ở danh mục |
| Tiền điều kiện | Tác nhân đã đăng nhập thành công |
| Luồng sự kiện chính  (Thành công | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | Người dùng | Chọn vào cá nhân (account) trên danh mục | | 2 | Hệ thống | Hiển thị giao diện thông tin cá nhân | | 3 | Người dùng | Chọn thay đổi thông tin (change information) | | 4 | Hệ thống | Hiển thị giao diện cập nhật thông tin | | 5 | Người dùng | Điền thông tin vào và xác nhận lưu | | 6 | Hệ thống | Cập nhật và thông báo thành công | |
| Luồng sự kiện thay thế | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 6a | Hệ thống | Thông báo lỗi: Nếu thông tin cập nhật không đúng định dạng | | 6b | Hệ thống | Thông báo lỗi: Nếu hệ thống không thể cập nhận thông tin | |
| Hậu điều kiện | Cập nhật thành công và thông tin được lưu vào hệ thống |

\*Dữ liệu đầu vào

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Trường dữ liệu | Mô tả | Bắt buộc | Điều kiện | Ví dụ |
| 1 | Mật khẩu cũ | Password Field | Không | Tối thiểu 6 kí tự | 123456@ |
| 2 | Mật khẩu mới | Password Field | Không | Tối thiểu 6 kí tự | 123456@H |
| 3 | Ảnh | Ảnh đại diện | Không | Định dạng JPG, PNG |  |
| 4 | Họ tên | Name Field | Không | Đúng cú pháp tiếng Việt | Nguyễn Văn A |
| 5 | Năm sinh | DatePicker | Không | Ngày tháng hợp lệ | 10/10/2001 |
| 6 | Giới tính | Male, Female | Không | Chọn 1 trong 2 | Male |
| 7 | Địa chỉ | Address ComboBox | Có |  | Đường …. |
| 8 | Điện thoại | TextField | Có | Kí tự số | 0987666554 |

* + 1. **Tìm kiếm nhà trọ**

|  |  |
| --- | --- |
| Tên Use Case: Tìm kiếm nhà trọ | |
| Tác nhân | Sinh viên, Chủ trọ (Người dùng) |
| Mô tả | Tác nhân sử dụng thanh tìm kiếm ở giao diện người dùng |
| Sự kiện kích hoạt | Nhập thông tin tìm kiếm vào textfield |
| Tiền điều kiện | Tác nhân đã đăng nhập thành công |
| Luồng sự kiện chính  (Thành công | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | Người dùng | Thực hiện nhập dữ liệu vào thanh tìm kiếm hoặc lướt trên trang chủ (dashboard) | | 2 | Người dùng | Yêu cầu tìm kiếm | | 3 | Hệ thống | Tìm và lấy về thông tin những nhà trọ có ký tự thỏa mãn tiêu chí tìm kiếm | | 4 | Hệ thống | Hiển thị các nhà trọ hợp lệ | |
| Luồng sự kiện thay thế | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 3a | Hệ thống | Thông báo : Không tìm thấy thông tin thỏa mãn yêu cầu người dùng | |
| Hậu điều kiện | Hiển thị những nhà trọ tương ứng cần tìm kiếm |

\*Dữ liệu đầu vào

* + 1. **Quản lý phòng trọ (Chủ trọ): Tạo, sửa đổi thông tin, xóa, cập nhật.**

|  |  |
| --- | --- |
| Tác nhân | Chủ trọ |
| Mô tả | Tác nhân sử dụng các tác vụ như: Sửa, Xóa, Cập nhật, Tạo hợp đồng |
| Sự kiện kích hoạt | Click vào quản lý nhà trọ cá nhân và lựa chọn các tác vụ |
| Tiền điều kiện | Tác nhân đã đăng nhập thành công |
| Tạo phòng trọ   |  |  | | --- | --- | | Use Case: Tạo phòng trọ | | | Luồng sự kiện chính | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | Chủ trọ | Chọn My House ở danh mục | | 2 | Hệ thống | Hiển thị giao diện phòng trọ của chủ trọ | | 3 | Chủ trọ | Chọn Create House | | 4 | Chủ trọ | Nhập thông tin nhà trọ mới theo yêu cầu | | 5 | Hệ thống | Kiểm tra các trường bắt buộc đã được nhập chưa và thông báo thêm phòng trọ mới thành công | | 6 | Hệ thống | Thêm phòng trọ vào danh sách trọ cá nhân | | | Luồng sự kiện thay thế | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 5a | Hệ thống | Thông báo : Nhập đầy đủ thông tin nếu các trường bắt buộc chưa được nhập | | 6a | Hệ thống | Thông báo lỗi: Nếu nhà trọ mới thêm vào danh sách chưa thành công | |   Xóa phòng trọ   |  |  | | --- | --- | | Use Case: Xóa phòng trọ | | | Luồng sự kiện chính | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | Chủ trọ | Click vào My House ở danh mục | | 2 | Chủ trọ | Chọn phòng trọ cần xóa | | 3 | Chủ trọ | Chọn Delete | | 4 | Hệ thống | Hiển thị xác nhận xóa trọ | | 5 | Hệ thống | Xóa và cập nhật lại danh sách trọ | | | Luồng sự kiện thay thế | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 5a | Hệ thống | Thông báo lỗi: Nếu không xóa được phòng trọ | | | |
| Sửa đổi thông tin trọ   |  |  | | --- | --- | | Use Case: Sửa đổi thông tin trọ | | | Luồng sự kiện chính | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | Chủ trọ | Click vào My House ở danh mục | | 2 | Chủ trọ | Chọn phòng trọ cần chỉnh sửa | | 3 | Hệ thống | Hiển thị thông tin trọ trên giao diện | | 4 | Chủ trọ | Nhập thông tin trọ theo các trường bắc buộc | | 4 | Chủ trọ | Xác nhận thay đổi thông tin | | 5 | Hệ thống | Thông báo thay đổi thông tin trọ thành công và lưu dữ liệu mới vào hệ thống | | | Luồng sự kiện thay thế | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 5a | Hệ thống | Thông báo lỗi: Nếu các trường bắt buộc chưa được nhập liệu chính xác | | | |
| Tạo hợp đồng   |  |  | | --- | --- | | Use Case: Tạo hợp đồng | | | Luồng sự kiện chính | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | Chủ trọ | Click vào Customer trên danh mục | | 2 | Hệ thống | HIển thị giao diện chứa các khách hàng | | 3 | Chủ trọ | Chọn View để xem khách hàng | | 4 | Chủ trọ | Chọn contact để tạo hợp đồng | | 5 | Hệ thống | Hiển thị hợp đồng cho thuê mẫu đã được cung cấp sẵn | | 6 | Chủ trọ | Nhập thông tin hợp đồng | | 7 | Chủ trọ | Xác nhận tạo hợp đồng | | 8 | Hệ thống | Thông báo tạo hợp đồng thành công và gửi file hợp đồng về tài khoản sinh viên | | 9 | Sinh viên | Xác nhận nhận hợp đồng | | | Luồng sự kiện thay thế | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 8a | Hệ thống | Thông báo lỗi nếu hợp đồng tạo không hợp lệ | | | |
| Cập nhật điện nước, CSVC   |  |  | | --- | --- | | Use Case: Cập nhật điện nước CSVC | | | Luồng sự kiện chính | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | Chủ trọ | Click vào Control trên danh mục | | 2 | Hệ thống | Hiển thị bảng điều khiển | | 3 | Chủ trọ | Tìm kiếm và cập nhật theo dõi điện nước | | 4 | Hệ thống | Thông báo cập nhật thành công và lưu dữ liệu vào hệ thống | | | Luồng sự kiện thay thế | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 4a | Hệ thống | Thông báo lỗi: Nếu hệ thống chưa cập nhật được dữ liệu | | | |
| Hậu điều kiện | Các tác vụ được hoàn thành và lưu lại, hiển thị tương ứng với những thao tác người dùng |

\*Dữ liệu đầu vào khi tạo/ sửa

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Trường dữ liệu | Mô tả | Bắt buộc | Điều kiện | Ví dụ |
| 1 | Tên trọ | TextField | Có |  |  |
| 2 | Địa chỉ | TextField | Có |  |  |
| 3 | Ảnh | Ảnh đại diện | Có | Định dạng JPG, PNG |  |
| 4 | CSVC | TextField | Có |  |  |
| 5 | Diện tích | TextField | Có | Đúng đơn vị đo đạc | 20 m2 |
| 6 | Mô tả | TextField | Có | Kí tự số |  |

\*Dữ liệu đầu vào khi tạo hợp đồng

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Trường dữ liệu | Mô tả | Bắt buộc | Điều kiện | Ví dụ |
| 1 | Tiền thuê | Money | Có | Đúng mệnh giá | 2.000.000VND |
| 2 | Hạn thuê | DatePicker | Có | Ngày tháng hợp lệ | 6 tháng |
| 3 | Ngày tháng bắt đầu thuê | DatePicker | Có | Ngày tháng hợp lệ | 22/2/2001 |

* + 1. **Quản lý sinh viên – nhà trọ(Chủ trọ):**

|  |  |
| --- | --- |
| Tác nhân | Chủ trọ |
| Mô tả | Tác nhân sử dụng các chức năng tương ứng để quan sát xử lý và quản lý đối tượng |
| Sự kiện kích hoạt | Click vào các nút tương ứng trên hệ thống |
| Tiền điều kiện | Tác nhân đã đăng nhập thành công |
| Tìm kiếm và xem sinh viên thuê trọ   |  |  | | --- | --- | | Use Case: Quản lý sinh viên đang thuê | | | Luồng sự kiện chính | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | Chủ trọ | Click vào Customer trên danh mục | | 2 | Hệ thống | HIển thị giao diện chứa các khách hàng đã thuê trọ | | 3 | Chủ trọ | Tìm, sửa với khách hàng tùy chọn | | | Luồng sự kiện thay thế | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 2a | Hệ thống | Thông báo: Hiển thị lỗi | | | |
| Liên hệ sinh viên   |  |  | | --- | --- | | Use Case: Liên hệ sinh viên | | | Luồng sự kiện chính | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | Chủ trọ | Chọn sinh viên cần liên hệ sau khi tìm kiếm | | 2 | Hệ thống | Hiển thị thông tin sinh viên và gửi thông báo đến tài khoản sinh viên | | 3 | Chủ trọ | Chọn Contact để liên hệ sinh viên | | | Luồng sự kiện thay thế | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 2a | Hệ thống | Thông báo lỗi: Nếu không gửi được thông báo đến tài khoản sinh viên | | | |
| Hậu điều kiện | Hiển thị những sinh viên cần tìm kiếm và gửi thành công thông báo liên hệ đến sinh viên |

# **Các yêu cầu phi chức năng**

## **Giao diện người dùng**

Giao diện hiển thị tốt trên các thiết bị khác nhau.

Đối với Người dùng khi truy cập sử dụng hệ thống, hệ thống sẽ nhận yêu cầu truy cập từ phía máy khách và gửi trả về cho các menu chức năng tương ứng với phạm vi của từng người dùng.

Trong trường hợp này Người dùng có thể tương tác với hệ thống thông qua cửa sổ màn hình với cấu trúc gồm có.

## **Tính bảo mật**

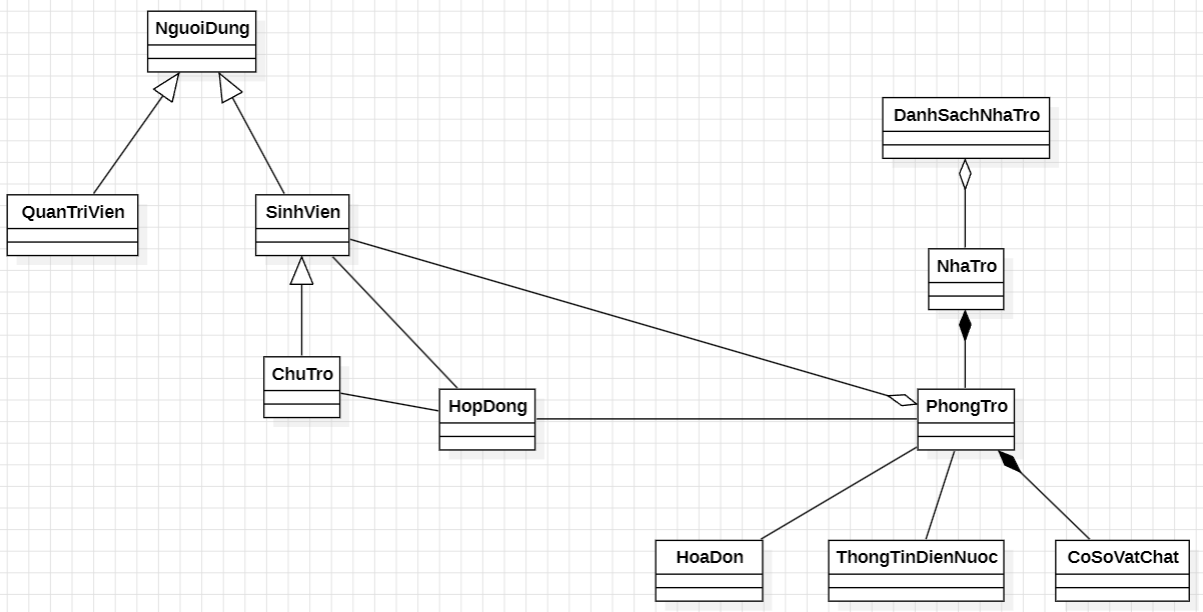
Người dùng chỉ có thể sử dụng các chức năng và truy cập các dữ liệu phù hợp với vai trò của người dùng đó.

## **Ràng buộc**

Hệ thống quản lý nhà ở cho sinh viên là hệ thống app desktop có kết nối mạng do vậy người dùng cần có các thiết bị có kết nối với internet như máy vi tính và được cung cấp các dịch vụ như thư điện tử, hình ảnh, văn bản, đa phương tiện, giao thức truyền thông siêu văn bản để có thể gửi và nhận thông tin, dữ liệu giữa máy khách và server.

Bên phía máy khách – người dùng cần cài đặt “Hệ thống quản lý nhà ở cho sinh viên”.

# **Sơ đồ lớp (phân tích)**



**Hình 13: Sơ đồ lớp phân tích**

# **Sơ đồ lớp thiết kế**

**Ảnh có chứa văn bản, biểu đồ, Song song, Phông chữ

Mô tả được tạo tự động**

**Hình 14:Sơ đồ lớp thiết kế**

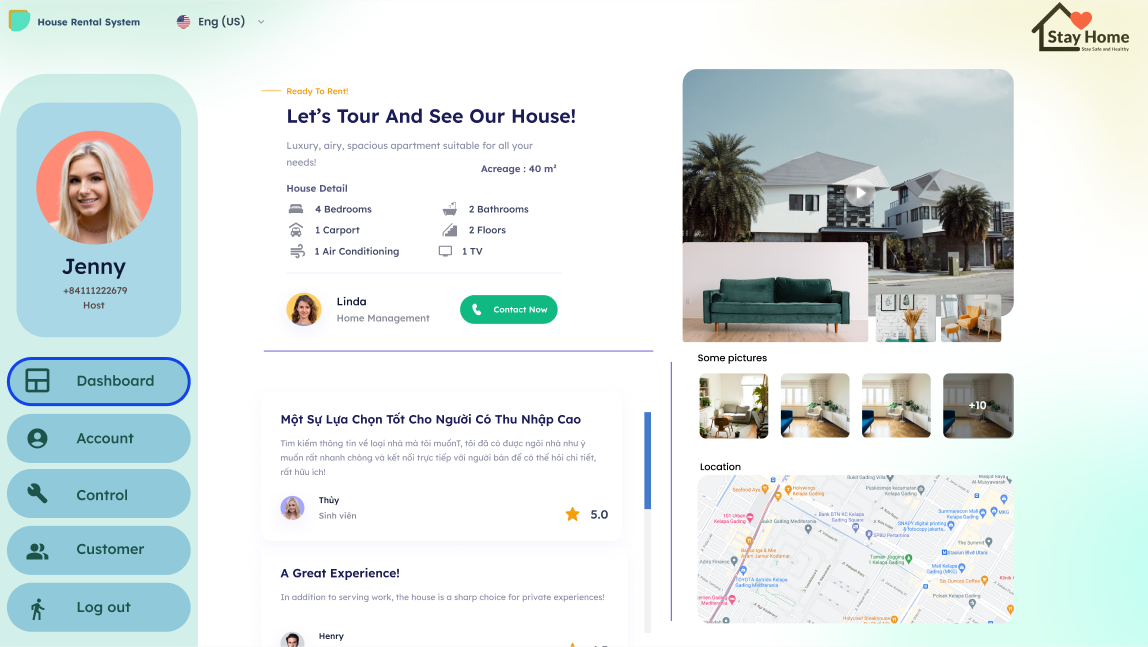
# **Thiết kế giao diện**

## **Giao diện chính**

Ảnh có chứa văn bản, ảnh chụp màn hình, Website, phần mềm

Mô tả được tạo tự động

## **Giao diện xem phòng**



## **Giao diện quản lý phòng đang được thuê của chủ trọ**

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Biểu tượng máy tính

Mô tả được tạo tự động

## **Giao diện quản lý của chủ trọ**

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, Hệ điều hành

Mô tả được tạo tự động

# **Code style**

## **White space**

**White space – thụt đầu dòng:**

* 1 đơn vị thụt đầu dòng = 1 tab
* 2 dòng code cách nhau một bậc thì cách nhau 1 tab

**White space – Dòng trống:**

* Những dòng code có quan hệ với nhau (cùng thực hiện một công việc) thì gom lại thành một block
* Hai block code thì cách nhau ít nhất một dòng trống
* Đặt khoảng trắng sau dấu phẩy và dấu chấm phẩy
* Đặt khoảng trắng xung quanh các toán tử

## **Comment**

* Viết các comment không cầu kì; càng đơn giản càng tốt.
* Khi dùng nhiều endline comment trên các dòng code liên tiếp nhau thì các comment này phải được canh lề như nhau.
* Nên vừa code vừa viết comment. Tránh trường hợp viết code xong rồi mới viết comment.
* Chỉ viết comment khi bạn cảm nhận là đoạn code của mình quá phức tạp.

## **Quy tắc đặt tên**

* + 1. **Quy tắc viết hoa**

**Pascal case**

* Các chữ cái đầu mỗi từ được viết hoa.
* Các chữ còn lại được viết thường.
* Ví dụ: MyProvider, StringBuilder

**Camel case**

* Giống với Pascal case nhưng chữ cái đầu của từ đầu tiên viết thường
* Ví dụ: myProvider, stringBuilder
  + 1. **Đặt tên class,  interface, abstract class**
* Sử dụng danh từ hay cụm danh từ : SinhVien, FormSinhVien,…
* Dùng **Pascal case** : SinhVien, FormSinhVien,…
* Hạn chế viết tắt gây khó hiểu :
* Sai: FormSV
* Đúng:FormSinhVien
* Không dùng tiền tố khi đặt tên lớp:
* Sai : ISinhVien
* Đúng: SinhVien
  + 1. **Phương thức**
* Sử dụng **Camel case** để đặt tên phương thức. Ví dụ: xepLoai.
* Tên phương thức thể hiện được chức năng của phương thức đó. tinhDiemTrungBinh.
* Tránh đặt tên gây cảm giác mơ hồ, không rõ nghĩa. Ví dụ: hienThi, tinh.
* Không phân biệt tên các phương thức bằng số. Ví dụ: tinhDiem1, tinhDiem2.
  + 1. **Biến**
* Sử dụng Camel case để đặt tên biến. Ví dụ: int diemTrungBinh, String hoTen
* Không dùng tiền tố. Ví dụ:
* Đúng: String address
* Sai: String strAddress
* Tên biến gợi nhớ, tránh viết tắt gây khó hiểu. Ví dụ:
* Đúng: String address
* Sai: String addr
* Không đặt tên biến chỉ bằng 1 chữ cái như x, y , z,… trừ trường hợp các biến đếm i, j, k.
* Không nên đặt tên biến quá dài, hay quá ngắn vì có thể làm rối chương trình hoặc cũng dẫn đến ý nghĩa biến mơ hồ (quá ngắn).
  + 1. **Biến static, enum**
* Tất cả các từ được viết hoa và phân cách bằng dấu gạch dưới (\_).
* Ví dụ:
* static float PI = 3.14f
* static int MIN\_WIDTH = 4
  + 1. **Biến final**
* Đối với biến final toàn cục: đặt tên biết giống như biến static. Tất cả các từ được viết hoa và phân cách bằng dấu gạch dưới (\_).
* Đối với biến fianl cục bộ: đặt tên biến giống như biến thông thường.
  + 1. **Đặt tên package**
* Tên package: tất cả đều là chữ thường.
  + 1. **Viết một phương thức hiệu quả**
* 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: khoảng từ 50 đến 150 dòng code.
  + 1. **Sử dụng biến (variables)**
* Tránh tình trạng khai báo biến mà không sử dụng: nhiều trình biên dịch warning khi complie (Eclipse IDE).
* Các lệnh if, while, for không nên lồng nhau hơn 3 bậc.
  + 1. **Import thư viện sử dụng**
* Chỉ import thư viện sử dụng cần thiết. Không sử dụng import tất cả.
* Ví dụ: sử dụng import java.util**.List**; thay cho import java.util**.\***;

# **Unit test**

## **Khái niệm về unit test**

Unit test là một phương pháp kiểm thử phần mềm tập trung vào việc kiểm tra các đơn vị nhỏ nhất của mã nguồn, thường là các hàm, phương thức, hoặc lớp, để đảm bảo rằng chúng hoạt động đúng cách. Mục tiêu của unit test là phát hiện và khắc phục lỗi ở giai đoạn phát triển sớm nhất có thể, giúp tiết kiệm thời gian và công sức về sau.

Khi làm Unit test chúng ta thường thấy các khái niệm sau:

* **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 …
* **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.
* **Test Suite:** Là một tập hợp các test case định nghĩa cho từng module hoặc hệ thống con.
* **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.
* **Production Code:** Phần mã chính của ứng dụng được chuyển giao cho khách hàng.
* **Unit Testing Code:** Phần mã phụ để kiểm tra mã ứng dụng chính, không được chuyển giao cho khách hàng.

## **Vòng đời Unit Test**

UT 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. Có rất nhiều PM hỗ trợ thực thi UT với giao diện trực quan. 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 chỉ thực sự đem lại hiệu quả khi:

* Được vận hành lặp lại nhiều lần
* Tự động hoàn toàn
* Độc lập với các UT khác.

## **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.

## **Ứng dụng Unit Test**

* Kiểm tra mọi đơn vị nhỏ nhất là các thuộc tính, sự kiện, thủ tục và hàm.
* Kiểm tra các trạng thái và ràng buộc của đối tượng ở các mức sâu hơn mà thông thường chúng ta không thể truy cập được.
* Kiểm tra các quy trình (process) và mở rộng hơn là các khung làm việc(workflow – tập hợp của nhiều quy trình)

## **Lợi ích của việc áp dụng Unit Test**

Thời gian đầu, người ta thường do dự khi phải viết UT thay vì tập trung vào code cho các chức năng nghiệp vụ. Công việc viết Unit Test có thể mất nhiều thời gian hơn code rất nhiều nhưng lại có lợi ích sau:

* Tạo ra môi trường lý tưởng để kiểm tra bất kỳ đoạn code nào, có khả năng thăm dò và phát hiện lỗi chính xác, duy trì sự ổn định của toàn bộ PM và giúp tiết kiệm thời gian so với công việc gỡ rối truyền thống.
* Phát hiện các thuật toán thực thi không hiệu quả, các thủ tục chạy vượt quá giới hạn thời gian.
* Phát hiện các vấn đề về thiết kế, xử lý hệ thống, thậm chí các mô hình thiết kế.
* Phát hiện các lỗi nghiêm trọng có thể xảy ra trong những tình huống rất hẹp.
* Tạo hàng rào an toàn cho các khối mã: Bất kỳ sự thay đổi nào cũng có thể tác động đến hàng rào này và thông báo những nguy hiểm tiềm tàng.

Trong môi trường làm việc Unit Test còn có tác dụng rất lớn đến năng suất làm việc:

* Giải phóng chuyên viên QA khỏi các công việc kiểm tra phức tạp.
* Tăng sự tự tin khi hoàn thành một công việc. Chúng ta thường có cảm giác không chắc chắn về các đoạn mã của mình như liệu các lỗi có quay lại không, hoạt động của module hiện hành có bị tác động không, hoặc liệu công việc hiệu chỉnh mã có gây hư hỏng đâu đó…
* Là công cụ đánh giá năng lực của bạn. Số lượng các tình huống kiểm tra (test case) chuyển trạng thái “pass” sẽ thể hiện tốc độ làm việc, năng suất của bạn.

## **Cách code hiệu quả với Unit Test**

Phân tích các tình huống có thể xảy ra đối với mã. Đừng bỏ qua các tình huống tồi tệ nhất có thể xảy ra, thí dụ dữ liệu nhập làm một kết nối cơ sở dữ liệu thất bại, ứng dụng bị treo vì một phép toán chia cho không, các thủ tục đưa ra lỗi ngoại lệ sai có thể phá hỏng ứng dụng một cách bí ẩn…

Mọi UT phải bắt đầu với trạng thái “fail” và chuyển trạng thái “pass” sau một số thay đổi hợp lý đối với mã chính.

Mỗi khi viết một đoạn mã quan trọng, hãy viết các UT tương ứng cho đến khi bạn không thể nghĩ thêm tình huống nào nữa.

Nhập một số lượng đủ lớn các giá trị đầu vào để phát hiện điểm yếu của mã theo nguyên tắc:

* Nếu nhập giá trị đầu vào hợp lệ thì kết quả trả về cũng phải hợp lệ
* Nếu nhập giá trị đầu vào không hợp lệ thì kết quả trả về phải không hợp lệ
* Sớm nhận biết các đoạn mã không ổn định và có nguy cơ gây lỗi cao, viết UT tương ứng để khống chế.

Ứng với mỗi đối tượng nghiệp vụ (business object) hoặc đối tượng truy cập dữ liệu (data access object), nên tạo ra một lớp kiểm tra riêng vì những lỗi nghiêm trọng có thể phát sinh từ các đối tượng này.

Để ngăn chặn các lỗi có thể phát sinh trở lại thực thi tự động tất cả UT mỗi khi có một sự thay đổi quan trọng, hãy làm công việc này mỗi ngày. Các UT lỗi cho chúng ta biết thay đổi nào là nguyên nhân gây lỗi.

Để tăng hiệu quả và giảm rủi ro khi viết các UT, cần sử dụng nhiều phương thức kiểm tra khác nhau. Hãy viết càng đơn giản càng tốt.

Cuối cùng, viết UT cũng đòi hỏi sự nỗ lực, kinh nghiệm và sự sáng tạo như viết PM.

**PHỤ LỤC**

**Link github:** [Vucore/CNPM---12 (github.com)](https://github.com/Vucore/CNPM---12?fbclid=IwZXh0bgNhZW0CMTAAAR2emsQbjvXO4HCsZvJM_W3uaoVQN7ZdyxZnyzm1_ElKHaunQoRilTKV9sA_aem_AYF9AHz-PdC3WQ3ntzJDsM2no9FRreEOMYsQPhODgqj4gUs_uQgL62BCXQszlGBhbxwGZoq-eW6dn3H9Ye_kPc0D)

**Link figma:** [CNPM Figma - Nhom 12 – Figma](https://www.figma.com/design/DbOTsZWfDbWD76a0M8l5dw/CNPM-Figma---Nhom-12?node-id=40-2512&t=GFas9I6VCZqrs3lM-0)