

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC BÁCH KHOA  
KHOA KHOA HỌC & KỸ THUẬT MÁY TÍNH



**BÁO CÁO ĐỀ CƯƠNG**  
**ĐỒ ÁN CHUYÊN NGÀNH**

---

**BUSINESS PROCESS EVALUATION**

---

<b>Giảng viên hướng dẫn:</b>	Trương Thị Thái Minh	
<b>Sinh viên thực hiện:</b>	Lê Văn Vỹ	2010805
	Nguyễn Trí Hiếu	2013153

## Mục lục

<b>1</b>	<b>Cơ sở lý thuyết</b>	<b>3</b>
1.1	Business Project Evaluation	3
1.1.1	Thời gian chu kỳ	3
1.1.2	Chi phí	4
1.1.3	Độ minh bạch	4
1.1.4	Xử lý ngoại lệ	4
1.1.5	Độ linh hoạt	4
1.1.6	Chất lượng	5
1.2	Process portfolio	5
1.3	Process selection	6
1.3.1	Độ quan trọng chiến lược	6
1.3.2	Sức khỏe	6
1.3.3	Tính khả thi	8
1.4	Survey	8
1.4.1	Nguyên tắc thiết kế bảng khảo sát	9
1.4.2	Yếu tố khảo sát	10
1.4.3	Phương pháp MUSA	12
1.4.4	Tính toán kết quả khảo sát	13
1.4.5	Tính toán giá trị chất lượng của quy trình nghiệp vụ	14
1.4.6	Đề xuất bảng khảo sát	15
<b>2</b>	<b>Phân tích và thiết kế hệ thống</b>	<b>23</b>
2.1	Giới thiệu hệ thống	23
2.2	Kiến trúc phần mềm	25
2.3	Use-case	27
2.3.1	Lược đồ use-case	27
2.3.2	Đặc tả use-case	28
2.3.3	Create new workspace	28
2.3.4	Join workspace through invitation	29
2.3.5	View user's list of workspaces	30
2.3.6	Access to projects' content in selected workspace	31
2.3.7	Send request for inviting other user	32
2.3.8	Send request for adjust permission	33
2.3.9	Delete member in Workspace	34
2.3.10	Manage requests from member	35
2.3.11	Create new project	36
2.3.12	Delete project in Workspace	37
2.3.13	Send invitation to other users	38
2.4	Sơ đồ hoạt động	39
2.4.1	Tạo mới workspace	39
2.4.2	Xoá workspace	40
2.4.3	Xoá thành viên khỏi Workspace	41
2.4.4	Quản lý requests trong Workspace	42
2.4.5	Thay đổi quyền của thành viên trong Workspace	43
2.4.6	Tạo mới project trong Workspace	44
2.4.7	Xoá project trong Workspace	45
2.4.8	Gửi lời mời trực tiếp đến cho user	46
2.5	Sơ đồ lớp	47
2.5.1	Tổng quan	47
2.5.2	Presentation Layer	47
2.5.3	Business Layer	53
2.5.4	Persistence Layer	56
2.6	Thiết kế cơ sở dữ liệu	60
2.6.1	Lược đồ ERD	60
2.6.2	Ảnh xạ ERD và mô tả chi tiết thực thể	60

<b>3</b>	<b>Hiện thực hệ thống</b>	<b>65</b>
3.1	Công nghệ sử dụng . . . . .	65
3.1.1	ReactJS - Javascript . . . . .	65
3.1.2	Python . . . . .	65
3.1.3	Websocket và Socket.IO . . . . .	66
3.1.4	PostgreSQL . . . . .	67
3.2	Hiện thực giao diện người dùng . . . . .	68
3.3	Kết luận . . . . .	69
3.3.1	Kết quả đạt được . . . . .	69
3.3.2	Những điểm hạn chế . . . . .	69
3.3.3	Hướng phát triển cho giai đoạn luận văn . . . . .	69
<b>4</b>	<b>Tham khảo</b>	<b>70</b>

# 1 Cơ sở lý thuyết

## 1.1 Business Project Evaluation

### 1.1.1 Thời gian chu kỳ

Thời gian chu kỳ của một quy trình được tính toán dựa trên thời gian chu kỳ của từng hoạt động trong quy trình đó. Thời gian được sử dụng ở đây là thời gian lý thuyết, không tính thời gian thực tế. Quy trình nghiệp vụ được chia thành các khối nhỏ hơn. Khi tính toán được thời gian chu kỳ của từng khối, có thể dễ dàng tính toán được chu kỳ của cả quy trình.

- Khối tuần tự (sequential fragment): thời gian chu kỳ của 1 khối tuần tự bằng với thời gian chu kỳ của các hoạt động trong khối đó:

$$CT = \sum_{i=1}^n T_i$$

- Khối XOR (Exclusive gateway): khối được bắt đầu và kết thúc bởi 2 cổng XOR, trong trường hợp này cần quan tâm tới xác suất rẽ nhánh của từng nhánh đi ra từ cổng XOR, ta ký hiệu là  $p$ :

$$CT = \sum_{i=1}^n p_i \times T_i$$

- Khối AND (parallel gateway): Khối được bắt đầu và kết thúc bởi 2 cổng AND, ta cần biết thời gian lớn nhất mà các nhánh cần để thực thi khối AND này.

$$CT = \max(T_1, T_2, \dots, T_n)$$

- Vòng lặp (rework pattern): Một chuỗi các hoạt động hoặc một hoạt động được lặp đi lặp lại với tần suất  $r$  nào đó, ta có công thức tính thời gian chu kỳ của vòng lặp:

$$CT = \frac{T}{1 - r}$$

Với  $T$  là tổng thời gian chu kỳ của khối lặp này

- Khối OR (inclusive gateway): Khối được bắt đầu và kết thúc bởi 2 cổng OR, cho phép ta chọn được một hoặc nhiều hơn một nhánh được rẽ ra từ cổng này. Công thức tổng quát cho khối OR:

$$CT = \max(T_1, T_2, \dots, T_i)$$

Với  $T_1, T_2, \dots, T_i$  là những nhánh được xác định sẽ được chạy trong khối OR này và  $i \leq n$  với  $n$  là tất cả các nhánh trong khối OR. Các nhánh được chạy sẽ được xác định ngay từ đầu.

### 1.1.2 Chi phí

Trong đề tài này, tương tự như đề tài trước, chúng tôi vẫn dựa trên *Time-driven activity-based costing model* (chi phí được tính dựa trên thời gian cần để thực hiện tất cả các task có trong quy trình đó và dựa trên chi phí đơn vị (unit cost) của khả năng cung ứng (supplying capacity)) để tính toán chi phí của quy trình dựa vào yếu tố thời gian. Công thức tính như sau:

$$C = UC \times T$$

Trong đó C là tổng chi phí của cả quy trình, T là thời gian chu kỳ của quy trình và UC (unit cost) là chi phí đơn vị của quy trình đó. Chi phí đơn vị được tính như sau:

$$UC = \frac{CTT}{PC}$$

Trong đó CT T là tổng số thời gian mà một đơn vị (phòng ban) trong tổ chức có thể bỏ ra trong một đơn vị thời gian. P C là chi phí có thể cung cấp cho một đơn vị của tổ chức trong một đơn vị thời gian.

### 1.1.3 Độ minh bạch

Tính minh bạch của một quy trình đối với một lane cụ thể được xác định với số lượng hoạt động minh bạch của một lane bất kỳ so với tổng số lượng hoạt động minh bạch hiện có trong quy trình đó. Một hoạt động được xem như là minh bạch khi nó không phải là một *sub - process* hay một *call - activity*:

$$\text{Transparent Level (specific view)} = \frac{\text{Num of explicit tasks (specific view)}}{\text{Num of explicit tasks (full view)}}$$

### 1.1.4 Xử lý ngoại lệ

Ngoại lệ là những sự kiện khiến cho quy trình đi từ luồng thông thường sang tình huống không mong đợi và không thường xảy ra. Có thể đó là những lỗi trong quy trình nghiệp vụ hoặc lỗi về mặt kỹ thuật. Công thức tính độ xử lý ngoại lệ như sau:

$$\text{Exception handling level} = \frac{\text{Num of handled exceptions}}{\text{Num of handled exceptions} + \text{Num of unhandled exceptions}}$$

Chúng tôi cùng với đề tài trước quy định để biểu diễn một ngoại lệ, người dùng phải biểu diễn bằng error event, nếu như event đó dẫn ra một hoạt động khác thì ngoại lệ đó được xem như là đã được xử lý. Còn lại các ngoại lệ khác sẽ được xem như chưa được xử lý.

### 1.1.5 Độ linh hoạt

Độ linh hoạt được biểu diễn bằng số lượng biến thể có thể có của một quy trình. Biến thể được tạo ra vì chúng có các khối hay các nhánh không phải thực thi. Độ linh hoạt của một quy trình được tính như

sau:

$$F = \frac{\text{Num of optional tasks}}{\text{Num of total tasks}}$$

Với *Num of optional tasks* là số lượng tasks có trong khối XOR, và *Num of total tasks* là tổng các tasks có trong quy trình đó.

### 1.1.6 Chất lượng

Chất lượng là một khái niệm trừu tượng và khó để mô hình hoá cụ thể. Ở đề tài trước, chất lượng của quy trình được đo lường dựa trên xác suất lặp lại  $r$  của các khối lặp, với công thức tính như sau:

$$Q = 1 - \frac{(r_1 + r_2 + \dots + r_n)}{n}$$

trong đó  $r_1, r_2, \dots, r_n$  là xác suất lặp lại của các vòng lặp trong quy trình,  $n$  là tổng số vòng lặp có trong quy trình.

Tuy nhiên, ở đề tài này, chúng tôi nhận thấy chất lượng của một quy trình không chỉ đến từ xác suất của các vòng lặp trong quy trình, mà nó còn có thể bị tác động bởi nhiều yếu tố khác. Chất lượng của một quy trình có thể được xem xét ở hai góc nhìn khác nhau: góc nhìn từ phía người sở hữu quy trình hay người thiết kế quy trình, và góc nhìn của người tham gia sử dụng, vận hành quy trình, sử dụng sản phẩm, dịch vụ của quy trình đó trên thực tế. Đây cũng có thể được xem lần lượt là *internal quality* (chất lượng bên trong) và *external quality* (chất lượng bên ngoài).

*Internal quality* của một quy trình nghiệp vụ được xem như chất lượng quy trình trong quá trình thiết kế. Ở đây, chúng tôi cho rằng *internal quality* có thể được xem như chất lượng gắn liền với số lượng và xác suất của các vòng lặp có trong quy trình, được biểu diễn và tính toán như trên. Mặt khác, *external quality* lại được đánh giá bởi sự hài lòng và trải nghiệm của những người sử dụng, tham gia vận hành hay sử dụng sản phẩm, dịch vụ của quy trình nghiệp vụ đó trong thực tế. Sự hài lòng đối với một quy trình có thể được diễn tả như mức độ mà người dùng (khách hàng) cảm nhận liệu quy trình nghiệp vụ có đáp ứng được nhu cầu của họ hay không. Sự hài lòng này không chỉ đơn thuần là cảm nhận trừu tượng, mà nó có thể được đo đạc với những giá trị cụ thể biểu thị cho mức độ của nó. Việc đo lường này giúp phản ánh được thái độ và kỳ vọng của khách hàng một cách hiệu quả, trực tiếp, có ý nghĩa, có thể tham khảo và so sánh.

## 1.2 Process portfolio

Nhìn vào thực tế, quy trình nghiệp vụ thường gắn liền với một tổ chức hoặc công ty, vì vậy việc quản lý quy trình nghiệp vụ liên quan đến một nhóm những quy trình có ảnh hưởng tới nhau.

Tuy nhiên, không phải tất cả những bên liên quan (stakeholders) phụ trách quản lý quy trình nghiệp vụ đều có được cái nhìn tổng quát về toàn bộ quy trình trong tổ chức của họ.

Và nhu cầu xác định những quy trình tạo ra tổn thất, nguy cơ đến doanh nghiệp trở nên cấp thiết hơn, để doanh nghiệp tập trung thực hiện giám sát, thay đổi hoặc loại trừ nhóm quy trình đó.

Từ đó, mong muốn hỗ trợ những người quản lý quy trình nghiệp vụ có được góc nhìn chính xác về việc xác định những quy trình cần được cải tiến đã thúc đẩy sự hình thành của Danh mục quy trình (trong ngữ cảnh của hệ thống thì chúng ta sẽ thống nhất dùng từ Process portfolio). Process portfolio đề cập tới một lược đồ có nhiệm vụ trực quan hóa những quy trình nghiệp vụ thông qua những tiêu chí được đề ra.

Process portfolio tạo ra mức độ ưu tiên giữa những quy trình nghiệp vụ, đây là cơ sở để hiện thực những hoạt động khác (hiện thực, phân tích, đánh giá hiệu suất, tái thiết kế, quản lý,...) liên quan tới lược đồ quy trình nghiệp vụ - cũng là đối tượng chính của hệ thống BKSky 2.0.

### 1.3 Process selection

Quá trình phân loại mức độ ưu tiên của quy trình nghiệp vụ sẽ dựa trên 3 tiêu chí: Độ quan trọng chiến lược (Strategic importance), Sức khỏe (Health), Tính khả thi (Feasibility).

#### 1.3.1 Độ quan trọng chiến lược

Độ quan trọng chiến lược được hiểu là mức độ ảnh hưởng của quy trình nghiệp vụ tới mục tiêu chiến lược của tổ chức; ví dụ như quy trình mang tính độc quyền thương hiệu, quy trình đóng góp vào lợi thế cạnh tranh trong thị trường, quy trình mang lại lợi nhuận chính cho tổ chức.

#### 1.3.2 Sức khỏe

Sức khỏe là tiêu chí nhằm vào việc đánh giá mức độ đạt được hiệu suất hoạt động mong muốn do tổ chức đề ra cho quy trình nghiệp vụ. Việc đánh giá tiêu chí này sẽ dễ dàng với những tổ chức hoạt động hướng quy trình - xác định nghiệp vụ dưới dạng những quy trình nghiệp vụ có thể mô hình hóa được; đây là một lợi thế cho phép doanh nghiệp thu thập được những thông tin cần thiết cho quá trình đánh giá hiệu suất. Cụ thể hơn, việc đánh giá hiệu suất quy trình nghiệp vụ sẽ được mô tả thông qua 4 tiêu chí: thời gian (time), chi phí (cost), chất lượng (quality) và độ linh hoạt (flexibility).

##### (a) Thời gian

Thời gian là yếu tố phổ biến trong đánh giá hiệu suất quy trình nghiệp vụ. Cụ thể hơn thì đại lượng được tính toán phổ biến là thời gian chu kỳ (cycle time) - khoảng thời gian mà quy trình nghiệp vụ bắt đầu cho đến khi kết thúc; đây là đại lượng được tập trung tính toán ở hệ thống tiền nhiệm (BKSky 1.0) nhằm mục đích đánh giá quy trình nghiệp vụ.

Mục tiêu của đa phần những quy trình nghiệp vụ là giảm thời gian chu kỳ thực tế, điều này có thể hiểu là giảm thời gian chờ đợi (waiting time) - khoảng thời gian mà quy trình chờ để nhận được phản hồi từ người dùng; hoặc thời gian phục vụ (serving time) - khoảng thời gian thực sự tiêu tốn để hoàn thành tác vụ quy định trong quy trình nghiệp vụ.

Trong ngữ cảnh của hệ thống hiện tại, chúng ta vẫn sẽ tập trung vào đánh giá hiệu suất của quy trình dựa vào đại lượng thời gian chu kỳ.

(b) Chi phí

Khía cạnh về chi phí là khía cạnh phổ biến được cân nhắc trong giai đoạn phân tích, đánh giá và tái thiết kế quy trình. Khi chúng ta đề cập tới chi phí, mục tiêu đi kèm là giảm thiểu chi phí được sử dụng để vận hành quy trình nghiệp vụ. Chi phí này có thể là chi phí cố định (fixed cost) - chi phí không bị ảnh hưởng bởi cường độ và tần suất vận hành của quy trình nghiệp vụ, thường là chi phí sử dụng cơ sở vật chất, chi phí bảo trì hệ thống,...; hoặc chi phí biến đổi (variable cost) - chi phí liên quan tới "lượng", ví dụ phụ thuộc vào số lượng sản phẩm, số lượng nhân viên,...

Kế thừa ngữ cảnh của hệ thống cũ, hệ thống BKSky 2.0 sẽ xem xét khía cạnh chi phí theo mô hình Chi phí dựa trên hoạt động theo thời gian (Time-driven activity-based costing model); trong mô hình này chúng ta sẽ quan tâm tới chi phí đơn vị (unit cost) mà tổ chức chi trả cho quy trình nghiệp vụ để thực thi, vận hành trong một đơn vị thời gian.

(c) Chất lượng

Chất lượng của quy trình có thể được nhìn nhận dưới hai góc nhìn chính: bên trong (internal quality) và bên ngoài (external quality). Chất lượng quy trình là khái niệm có phần trừu tượng, định nghĩa chất lượng có thể được tổng hợp từ nhiều yếu tố nhỏ liên quan; tuy nhiên trong ngữ cảnh hệ thống hiện tại, chúng ta sẽ tập trung phản ánh chất lượng của quy trình thông qua tính dễ hiểu của quy trình đối với người tham gia quy trình (process's participants - phân biệt với khách hàng sử dụng sản phẩm/dịch vụ cung cấp bởi đầu ra của quy trình).

Kế thừa ngữ cảnh của hệ thống BKSky 1.0, thời gian của các tác vụ trong khối lập trong quy trình ảnh hưởng theo một trọng số lớn đến tổng thời gian của cả quy trình. Vì vậy xác suất lập của khối lập càng lớn thì khả năng nó tác động đến tổng thời gian của quy trình càng cao, chất lượng bên trong của quy trình sẽ bị ảnh hưởng bởi xác suất lập của các khối lập. Yếu tố này sẽ được kế thừa trong hệ thống hiện tại.

Bên cạnh đó, phản hồi từ phía người tham gia quy trình nghiệp vụ cũng đóng góp vào việc đánh giá chất lượng của quy trình. Phản hồi có thể được thu thập thông qua khảo sát hoặc những lời góp ý từ phía người tham gia quy trình. Hệ thống hiện tại sẽ tập trung vào việc khai thác phản hồi từ phía người dùng để hoàn thiện hơn quá trình đánh giá chất lượng của quy trình nghiệp vụ.

(d) Độ linh hoạt

Độ linh hoạt của quy trình thể hiện thông qua khả năng thay đổi để thích nghi dưới những điều kiện khác nhau của quy trình nghiệp vụ. Tổ chức thường có mong muốn khiến quy trình nghiệp vụ của họ nhanh hơn, rẻ hơn và tốt hơn mà không chú ý đến yếu tố thay đổi của quy trình. Có thể dưới những hoàn cảnh khác nhau, tính ổn định vốn có của quy trình sẽ đánh mất hoàn toàn; khi đó, để so sánh hai quy trình với nhau thì quy trình có thể hoạt động ổn định trong đa số hoàn cảnh lại chiếm ưu thế hơn quy trình hoạt động tốt khi tải bình thường nhưng lại đình trệ khi tải vượt quá cao.

Mục tiêu là chúng ta cần nâng cao được tính linh hoạt của quy trình để có thể đáp ứng nhiều điều kiện khác nhau, tính thích ứng có thể thể hiện thông qua:



- Run-time flexibility - Khả năng sinh ra luồng thực thi ứng với những điều kiện đầu vào khác nhau: Luồng thực thi khi này sẽ được xem xét như là một biến thể (variation) của quy trình nghiệp vụ.
- Build-time flexibility - Khả năng thay đổi cấu trúc của quy trình ứng với những hoàn cảnh khác nhau: Quy trình khi này sẽ sinh ra những phiên bản (version), có thể khác biệt về cấu trúc quy trình, nhằm đảm bảo được đầu ra ổn định dưới những điều kiện khác nhau.

Trong ngữ cảnh của hệ thống BKSky 1.0, chúng ta đã xem xét tính linh hoạt của quy trình nghiệp vụ bị ảnh hưởng bởi số lượng biến thể sinh ra khi luồng thực thi rẽ nhánh. Ở hệ thống hiện tại, chúng ta sẽ kế thừa đồng thời phát triển thêm về tính linh hoạt của quy trình khi hoàn cảnh khác nhau sẽ yêu cầu cấu trúc khác nhau nhằm đảm bảo tính ổn định của hiệu suất quy trình.

### 1.3.3 Tính khả thi

Tính khả thi là tiêu chí thứ ba được cân nhắc trong process portfolio; trong phạm vi ngữ cảnh của hệ thống, tính khả thi của quy trình nghiệp vụ sẽ liên quan tới câu hỏi "Quy trình có khả năng mở rộng không?". Thay vì tập trung vào câu hỏi "Quy trình có khả năng hiện thực không?" thì khả năng mở rộng của quy trình trong tương lai sẽ ảnh hưởng nhiều hơn tới quyết định đầu tư tài nguyên và nhân lực của tổ chức. Tính khả thi thể hiện rõ ràng hơn trên con đường phát triển dài về sau; khi mà mục tiêu hiệu suất trở nên ngày càng cao thì doanh nghiệp sẽ phải nỗ lực hơn để liên tục thử nghiệm và thay đổi quy trình nghiệp vụ, quá trình này có thể kéo dài theo năm. Vì vậy để có được điểm khởi đầu tốt thì tính khả thi của quy trình nghiệp vụ là đáng được cân nhắc.

## 1.4 Survey

Đánh giá *external quality* đồng nghĩa với việc đánh giá được sự hài lòng của người dùng hoặc khách hàng đối với một quy trình nghiệp vụ. Để làm được như vậy, cần phải thu thập những ý kiến của người sử dụng hoặc theo dõi và tự động đánh giá quá trình người dùng thực thi quy trình nghiệp vụ. Ngày nay, có một số công cụ dưới dạng các ứng dụng mở rộng (add-on extensions) trên các trình duyệt web, cho phép tích hợp vào hệ thống để theo dõi các luồng thực thi của khách hàng trên một quy trình cụ thể, chẳng hạn như khách hàng có hoàn thành phiên hoạt động của mình hay không, khách hàng có rơi vào những trường hợp ngoại lệ không, có khách hàng nào dừng quy trình giữa chừng hay không. Ứng dụng các công cụ trên vào đánh giá chất lượng bên ngoài của quy trình có thể đảm bảo tính chính xác, và được thực thi một cách tự động dựa trên việc theo dõi hành vi của người dùng, không tốn nhiều thời gian, công sức của người dùng. Tuy nhiên, không phải quy trình nghiệp vụ nào cũng có thể được tích hợp công cụ trên. Một số quy trình nghiệp vụ cần được thực thi ngoài thực tế, không phải trên các trang web hay hệ thống máy tính, dẫn tới không thể ứng dụng các công cụ theo dõi hành vi người dùng vào những quy trình nghiệp vụ này. Chúng tôi nhận thấy rằng, đối với những loại quy trình này nói riêng, hay cả những quy trình khác nói chung, đều có thể thu thập được sự hài lòng và trải nghiệm của người dùng thông qua các cuộc khảo sát (survey).

Khảo sát đã được ứng dụng từ lâu để lấy ý kiến khách hàng về một dịch vụ, sản phẩm,... hay thậm chí là cả một tổ chức, một doanh nghiệp. Ngày nay, khảo sát vẫn là một công cụ phổ biến và hiệu quả để thu thập thông tin từ người dùng, cho nên ta có thể sử dụng khảo sát, với các câu hỏi khai thác hợp lý, có thể đánh giá được thái độ của khách hàng đối với quy trình nghiệp vụ cụ thể như thế nào, từ đó có thể rút ra được *external quality* của quy trình. Khảo sát có thể được thực thi trên nhiều nền tảng khác nhau với nhiều hình thức khác nhau, chẳng hạn như có thể khảo sát khách hàng thông qua chatbot - là một phần mềm ứng dụng trí tuệ nhân tạo để mô phỏng lại các cuộc trò chuyện với người dùng, thông qua cuộc trò chuyện đó có thể lấy được thông tin từ người dùng. Hoặc có thể là một bảng khảo sát trực tuyến với các câu hỏi đa dạng hình thức được thiết kế sẵn để người dùng trả lời và gửi phản hồi về quy trình đó. Khảo sát trực tuyến có một số lợi ích như sau:

- Người dùng có thể thực hiện khảo sát bất cứ khi nào họ muốn.
- Người dùng có thể dành nhiều thời gian hơn để trả lời các câu hỏi, nói cách khác, họ không bị ràng buộc thời gian.
- Khảo sát trực tuyến cho phép người thiết kế đưa ra nhiều hình thức câu hỏi khác nhau để đánh giá nhiều khía cạnh khác nhau của vấn đề cần được khảo sát (câu hỏi hai đáp án, nhiều đáp án, câu hỏi dạng thang điểm,...).

Vì một số ưu điểm nêu trên, chúng tôi quyết định lựa chọn khảo sát trực tuyến như là phương tiện chính để đánh giá sự hài lòng của khách hàng đối với quy trình nghiệp vụ.

#### **1.4.1 Nguyên tắc thiết kế bảng khảo sát**

##### **1.4.1.1 Nguyên tắc**

Một số nguyên tắc khi thiết kế bảng khảo sát được tổng hợp bên dưới như sau:

- Sử dụng từ ngữ đơn giản, quen thuộc, tránh sử dụng các từ ngữ chuyên môn, tiếng lóng, nhiều tầng lớp nghĩa, mơ hồ.
- Sử dụng cấu trúc câu đơn giản.
- Đưa ra các đáp án lựa chọn đầy đủ, toàn diện, và riêng biệt với nhau.
- Tránh những câu hỏi đôi, những câu hỏi phủ định hay phủ định của phủ định.
- Tránh những câu hỏi định hướng người dùng buộc phải chọn một đáp án một cách chủ ý.
- Những câu hỏi mở đầu nên là những câu hỏi dễ, từ đó xây dựng được sự kết nối giữa người thiết kế câu hỏi và người trả lời câu hỏi.
- Các câu hỏi thuộc cùng một chủ đề nên được nhóm lại với nhau.
- Các câu hỏi thuộc cùng một chủ đề nên đi từ chung đến riêng.

#### 1.4.1.2 Thang điểm

Việc chọn lựa thang điểm đánh giá cũng là một yếu tố cần được xem xét khi thiết kế bảng câu hỏi khảo sát. Likert (1932) sử dụng thang điểm 5. Osgood, Suci và Tannenbaum (1957) sử dụng thang điểm 7, hay như Thurstone (1928) lại sử dụng thang điểm 11. Không có một quy chuẩn nào cho việc sử dụng thang điểm nào trong khảo sát, dẫu vậy, chúng tôi cho rằng một số thang điểm cụ thể nên được sử dụng để tối ưu hoá dữ liệu thu được từ người dùng. Người dùng khi quan sát thang điểm sẽ thực hiện việc đối chiếu các mức điểm với câu trả lời đưa ra và cố gắng tìm kiếm mức điểm trùng khớp nhất có thể. Vì vậy, có một số điều kiện nhất định cần phải được thỏa mãn.

- Thứ nhất, thang đo cần bao phủ các mức độ nhiều nhất có thể, không có ngoại lệ nào.
- Thứ hai, các mức điểm trong thang đo cần có sự khác biệt với nhau, ý nghĩa của chúng không đan xen trùng lặp nhau.
- Thứ ba, người dùng và người thiết kế câu hỏi phải hiểu được ý nghĩa của mỗi mức điểm và có cách hiểu giống nhau.

Nếu một vài trong những điều kiện trên không được đáp ứng có thể ảnh hưởng đến chất lượng của dữ liệu. Ví dụ, nếu câu trả lời người dùng rơi vào một trường hợp nào đó chưa được liệt kê trong thang điểm, nói cách khác, độ chi tiết của thang đo chưa cao dẫn tới chưa có đáp án thực sự phù hợp với câu trả lời của người dùng lúc đó, dẫn tới họ có thể chọn các đáp án khác nhau ở các thời điểm thực hiện khảo sát khác nhau. Ngoài ra, những đáp án có ý nghĩa tương tự nhau (chẳng hạn như “đôi khi” - “thỉnh thoảng”) sẽ khiến người dùng bối rối khi chọn lựa. Khi đó, giả sử như có nhiều người cùng thực hiện bài khảo sát này, họ có thể chọn những đáp án khác nhau vì họ có cách hiểu khác nhau đối với mỗi đáp án mặc dù câu trả lời của họ đưa ra trong đầu là giống nhau.

#### 1.4.2 Yếu tố khảo sát

Một số độ đo độ hài lòng của khách hàng đối với một quy trình nghiệp vụ nói riêng và một dịch vụ, sản phẩm nói chung được áp dụng rộng rãi hiện nay có thể kể đến như: Customer Satisfaction Score (CSAT), Customer Effort Score (CES), Net Promoter Score (NPS).

##### 1.4.2.1 CES

CES được Corporate Executive Board (CEB) giới thiệu vào năm 2010. CES là dạng câu hỏi đánh giá sự hài lòng của người dùng bằng việc đo đặc nỗ lực của người dùng trong việc tương tác với sản phẩm hay dịch vụ, yêu cầu người dùng đánh giá nỗ lực của họ trên thang điểm từ 1 - 7, với 1 là giá trị cao nhất cho sự không đồng thuận với câu hỏi được đặt ra.

Một số công thức tính CES đã được đưa ra, tuy nhiên mỗi công thức sẽ phù hợp với một số những thang điểm nhất định. Ở đây, chúng tôi chọn thang điểm từ 1 - 7, công thức tính thường gặp sẽ lấy số lượng người đồng ý chia cho tổng số phản hồi từ người dùng và nhân với 10 hoặc 100. Người dùng được xem như là đồng ý khi số điểm họ đánh giá cho câu hỏi này nằm trong tập giá trị {5, 6, 7}. Miền giá trị

của điểm CES là  $[0, 1]$ .

$$CES (\%) = \frac{\text{Number of positive results}}{\text{Total of respondents}} \times 100$$

Trong đó *Number of positive results* là số lượng người dùng đồng ý với câu hỏi, *Total of respondents* là số lượng người dùng tham gia trả lời câu hỏi. Ví dụ, nếu chúng tôi nhận được 100 phản hồi và 70 trong số đó là phản hồi tích cực, thì điểm CES sẽ là 70%.

#### 1.4.2.2 CSAT

CSAT là một yếu tố đo lường mức độ hài lòng của khách hàng đối với một trải nghiệm cụ thể. CSAT thường được đo trên thang điểm từ 1 - 5, với 1 là rất không hài lòng và 5 là tuyệt đối hài lòng. Điểm CSAT nằm trong khoảng từ 75% đến 85% được xem như là một giá trị tốt. Ta có thể tính điểm CSAT bằng cách lấy số lượng phản hồi hài lòng chia cho tổng số phản hồi từ người dùng và nhân với 100. Người dùng được xem như là hài lòng khi họ chọn đáp án thuộc tập giá trị  $\{4, 5\}$ . Miền giá trị của điểm CSAT là  $[0, 1]$ .

$$CSAT (\%) = \frac{\text{Number of satisfied responses}}{\text{Total of respondents}} \times 100$$

Trong đó, *Number of satisfied responses* là số phản hồi hài lòng từ khách hàng, *Total of respondents* là tổng số phản hồi khảo sát từ người dùng. Ví dụ, nếu chúng tôi nhận được 100 phản hồi và 80 trong số đó là đánh giá hài lòng, điểm CSAT sẽ là 80%.

#### 1.4.2.3 NPS

NPS là câu hỏi được phát triển bởi Reichheld vào năm 2003, đánh giá tỉ lệ người dùng sẽ đề xuất sản phẩm hay dịch vụ cho người khác, trên thang điểm từ 0 - 10, với 0 là giá trị cho biết sản phẩm hay dịch vụ chắc chắn sẽ không được đề xuất, và 10 là giá trị đảm bảo rằng sản phẩm hay dịch vụ sẽ được đề xuất cho người khác bởi người dùng.

Những phản hồi từ khảo sát NPS được phân loại thành ba nhóm chính: Promoters, Passives và Detractors. *Promoters* là những người dùng có lòng trung thành cao đã đánh giá trải nghiệm của họ từ 9 đến 10. *Passives* là những người trung lập và đánh giá ở mức 7 và 8. Trong khi đó, *Detractors* là những người không hài lòng với sản phẩm và dịch vụ, và chỉ cho số điểm từ 0 đến 6. Điểm NPS được tính bằng tỉ lệ giữa hiệu của số người thuộc nhóm Promoters trừ đi số người thuộc nhóm Detractors trên tổng số người dùng tham gia khảo sát.

$$NPS = \frac{\text{Promoters} - \text{Detractors}}{\text{Total of respondents}} \times 100$$

Trong đó, *Promoters* là số người thuộc nhóm Promoters, *Detractors* là số người thuộc nhóm Detractors, *Total of respondents* là tổng số phản hồi khảo sát từ người dùng.

Như vậy có thể thấy, điểm NPS sẽ nằm trong khoảng giá trị từ -100% đến 100%, hay  $NPS \leq |1|$ . Theo Reichheld và Markey, giá trị NPS từ -100% đến 0 sẽ là không hài lòng, với -100% đến -50% là *deficient*, và từ -49% đến 0 là *insufficient*. *Deficient* nghĩa là chất lượng của dịch vụ được đánh giá là cực kỳ tệ. *Insufficient* cho thấy người dùng đánh giá chất lượng không tốt lắm. Ngược lại, kết quả rơi vào khoảng

giá trị từ 0 đến 100% sẽ là mức độ hài lòng, với 0 đến 49% và *sufficient* và 50% đến 100% là *excellent*. *Sufficient* được hiểu rằng chất lượng của dịch vụ nhận được phản hồi khá tích cực, trong khi đó, *excellent* cho thấy người dùng đánh giá rất cao.

### 1.4.3 Phương pháp MUSA

Một vấn đề đặt ra, đó bảng khảo sát đo đạc giá trị CES, CSAT, NPS không chỉ trên một yếu tố của quy trình, mà còn trên nhiều yếu tố khác, chẳng hạn như khả năng xử lý các ngoại lệ và độ linh hoạt của quy trình, cần phải tổng hợp các giá trị đó thành một giá trị đo lường chung. Bên cạnh đó, tùy vào mỗi loại quy trình nghiệp vụ, các yếu tố được đánh giá có thể có những trọng số khác nhau, phụ thuộc vào người thiết kế quy trình ưu tiên yếu tố nào hơn. Chẳng hạn, đối với một quy trình A, người thiết kế xem trọng việc xử lý những ngoại lệ hơn hẳn tính linh hoạt của quy trình, dẫn đến việc trong bảng khảo sát, một điều tất yếu là điểm CES của xử lý ngoại lệ có độ ưu tiên cao hơn điểm CES của tính linh hoạt của quy trình. Điều này cũng sẽ ảnh hưởng đến điểm tổng CES của cả cuộc khảo sát.

Một bài nghiên cứu về đánh giá sự hài lòng của khách hàng trong lĩnh vực ngân hàng tư nhân của ngân hàng Thương mại Hy Lạp vào năm 1999, đánh giá trên nhiều tiêu chí khác nhau:

- Bộ phận nhân sự: Bao gồm các đặc trưng như: kỹ năng và kiến thức chuyên môn, tính trách nhiệm, khả năng giao tiếp và làm việc với khách hàng, sự thân thiện,...
- Sản phẩm: Tiêu chí này tập trung chủ yếu vào đặc điểm các sản phẩm cung cấp cho khách hàng, như độ đa dạng, khả năng bồi thường, giá cả, các dịch vụ đặc biệt,...
- Hình ảnh thương hiệu: Bao gồm tên tuổi, danh tiếng của ngân hàng, khả năng ứng dụng công nghệ và đáp ứng nhu cầu khách hàng trong tương lai.
- Dịch vụ: Liên quan tới những dịch vụ cung cấp cho khách hàng, như thời gian chờ đợi để được xử lý, độ phức tạp của các quy trình, thông tin cung cấp cho khách hàng,...
- Khả năng truy cập: Khả năng mở rộng mạng lưới của ngân hàng, vị trí các chi nhánh, khả năng xử lý những vấn đề có thể xảy ra với hệ thống (ATM bị lỗi).

Để tính toán được giá trị hài lòng cuối cùng (global satisfaction), trước đó cần biết được mức độ hài lòng của khách hàng trên các tiêu chí được liệt kê ở trên (partial satisfaction). Và mỗi tiêu chí lại phụ thuộc vào những đặc trưng bên trong chúng. Các tiêu chí hay đặc trưng của chúng có độ quan trọng khác nhau, đòi hỏi cần phải kết hợp các giá trị này lại với nhau thành một giá trị tổng hoà duy nhất. Tác giả đã đề xuất việc sử dụng phương pháp MUSA để xử lý vấn đề trên, và khi đối chiếu ngược trở về vấn đề xây dựng khảo sát đánh giá sự hài lòng của khách hàng đối với quy trình nghiệp vụ của chúng tôi, chúng tôi nhận thấy có sự tương đồng, bởi chúng tôi cũng khảo sát sự hài lòng trên nhiều yếu tố liên quan đến quy trình nghiệp vụ, chẳng hạn như khả năng xử lý ngoại lệ, thời gian, chi phí thực thi quy trình, sản phẩm đầu ra của quy trình. Chính vì thế, chúng tôi quyết định lựa chọn ứng dụng phương pháp MUSA vào việc tính toán mức độ hài lòng của khách hàng trên nhiều khía cạnh khác nhau của quy trình nghiệp vụ.

Multicriteria Satisfaction Analysis (MUSA) là phương pháp để đo lường và phân tích sự hài lòng của khách hàng. Phương pháp MUSA là một mô hình phân tích mức độ ưu tiên theo những nguyên tắc của phân tích hồi quy thứ tự. Phương pháp luận này sẽ đánh giá mức độ hài lòng của một tập những người dùng dựa trên giá trị của họ và những mức độ ưu tiên. Quá trình kết hợp - phân tích này sẽ được thực thi với ít khả năng xảy ra lỗi nhất. Ưu điểm của phương pháp MUSA là nó hoàn toàn xem xét chất lượng mức độ ưu tiên và đánh giá của người dùng.

Mục đích chính của phương pháp MUSA là kết hợp các đánh giá độc lập thành một hàm thu thập giá trị, giả sử như độ hài lòng tổng thể của người dùng sẽ phụ thuộc vào tập  $n$  tiêu chí hay biến số đại diện cho các yếu tố khác nhau của dịch vụ. Tập các tiêu chí này được ký hiệu  $X = (X_1, X_2, \dots, X_n)$  với mỗi tiêu chí cụ thể  $i$  được đại diện bằng một biến đơn  $X_i$ . Bằng cách này, việc đánh giá sự hài lòng của người dùng có thể được xem như một bài toán phân tích nhiều tiêu chí.

Phương pháp MUSA đánh giá độ hài lòng tổng thể và độ hài lòng ở từng yếu tố lần lượt  $Y^*$  và  $X_i^*$ . Cần chú ý rằng phương pháp này tuân theo những nguyên tắc của phân tích hồi quy tuần tự với một số ràng buộc, sử dụng các kỹ thuật quy hoạch tuyến tính (Jacquet-Lagrezze and Siskos, 1982; Siskos and Yannacopoulos, 1985; Siskos, 1985). Công thức của phân tích hồi quy tuần tự trình bày như sau:

$$Y^* = \sum_{i=1}^n b_i X_i^*$$

$$\sum_{i=1}^n b_i = 1$$

Với  $b_i$  là trọng số của tiêu chí thứ  $i$  và giá trị của  $Y^*$  và  $X_i^*$  đã được chuẩn hoá về miền giá trị  $[0, 1]$ . Các hàm dự đoán giá trị là những kết quả quan trọng nhất của phương pháp MUSA, chúng cho thấy giá trị thực trong miền giá trị  $[0, 1]$  là giá trị mà người dùng đánh giá cho mỗi mức độ hài lòng, từng yếu tố hay tổng thể.

#### 1.4.4 Tính toán kết quả khảo sát

Như đã đề cập, bài khảo sát sẽ có những câu hỏi để tính toán các giá trị để đánh giá mức độ hài lòng của người dùng đối với một quy trình nghiệp vụ. Các độ đo ở những thang điểm khác nhau, cụ thể CES và CSAT tính trên thang điểm 1 - 7, NPS trên thang điểm 0 - 10. Vì thế, chúng tôi nhận thấy cần phải chuẩn hóa các độ đo về một miền giá trị duy nhất.

Hiện nay có nhiều phương pháp sử dụng để chuẩn hoá dữ liệu, và *Min-Max* là một trong số đó. Chuẩn hoá *Min-Max* biểu diễn sự biến đổi tuyến tính trên dữ liệu ban đầu. Biết rằng  $\min_a$  và  $\max_a$  lần lượt là giá trị nhỏ nhất và giá trị lớn nhất của thuộc tính  $A$ . Phương pháp chuẩn hoá *Min-Max* sẽ ánh xạ một giá trị  $v$  của  $A$  thành  $v'$  trong khoảng  $(\text{new} - \min_a, \text{new} - \max_a)$  bằng việc tính toán như công thức dưới đây:

$$v' = \left( \frac{v - \min_a}{\max_a - \min_a} \right) \times ((\text{new} - \max_a) - (\text{new} - \min_a)) + \text{new} - \min_a$$

Ở đây chúng tôi chọn một miền giá trị phổ biến là  $[0, 1]$ , và đưa ra công thức chuẩn hoá cụ thể cho

từng độ đo như sau, vận dụng phương pháp chuẩn hoá *Min-Max* được trình bày ở trên:

**Độ đo CES:**

$$\text{Normalized CES} = \left( \frac{CES - CES_{Min}}{CES_{Max} - CES_{Min}} \right) \times 100$$

Trong đó, *CES* là giá trị *CES* tính được từ kết quả khảo sát,  $CES_{Min}$  và  $CES_{Max}$  lần lượt là giá trị nhỏ nhất và giá trị lớn nhất của điểm *CES*, hay cận dưới và cận trên của thang điểm của độ đo *CES*. Ví dụ, trên miền giá trị của điểm *CES* là  $[0, 1]$ , ta có  $CES_{Min} = 0$  và  $CES_{Max} = 1$

**Độ đo CSAT:**

$$\text{Normalized CSAT} = \left( \frac{CSAT - CSAT_{Min}}{CSAT_{Max} - CSAT_{Min}} \right) \times 100$$

Trong đó, *CSAT* là giá trị *CSAT* tính được từ kết quả khảo sát,  $CSAT_{Min}$  và  $CSAT_{Max}$  lần lượt là giá trị nhỏ nhất và giá trị lớn nhất của điểm *CSAT*, hay cận dưới và cận trên của thang điểm của độ đo *CSAT*. Ví dụ, trên miền giá trị của điểm *CSAT* là  $[0, 1]$ , ta có  $CSAT_{Min} = 0$  và  $CSAT_{Max} = 1$

**Độ đo NPS:**

$$\text{Normalized NPS} = \left( \frac{NPS - NPS_{Min}}{NPS_{Max} - NPS_{Min}} \right) \times 100$$

Trong đó, *NPS* là giá trị *NPS* tính được từ kết quả khảo sát,  $NPS_{Min}$  và  $NPS_{Max}$  lần lượt là giá trị nhỏ nhất và giá trị lớn nhất của điểm *NPS*, hay cận dưới và cận trên của thang điểm của độ đo *NPS*. Ví dụ, trên miền giá trị của điểm *NPS* là  $[-1, 1]$ , ta có  $NPS_{Min} = -1$  và  $NPS_{Max} = 1$

Tuỳ vào mỗi loại quy trình nghiệp vụ mà người thiết kế sẽ ưu tiên xem xét độ đo nào chiếm tỉ trọng lớn hơn trong giá trị chung của bảng khảo sát. Vì thế, ứng dụng phương pháp MUSA đã được trình bày ở trên, chúng tôi đề xuất công thức tính giá trị chung như sau:

$$\text{Survey Score} = (w_{CES} \times \text{Normalized CES}) + (w_{NPS} \times \text{Normalized NPS}) + (w_{CSAT} \times \text{Normalized CSAT})$$

$$w_{CES} + w_{CSAT} + w_{NPS} = 1$$

Trong đó,  $w_{CES}$ ,  $w_{CSAT}$ ,  $w_{NPS}$  lần lượt là trọng số của giá trị *CES*, *NPS* và *CSAT* đã chuẩn hóa. Trọng số này có thể được quy định bởi người thiết kế bảng khảo sát, tùy thuộc vào độ ưu tiên của họ đối với độ đo nào cho bảng khảo sát hay quy trình nghiệp vụ.

#### 1.4.5 Tính toán giá trị chất lượng của quy trình nghiệp vụ

Như đã đề cập, chất lượng của một quy trình nghiệp vụ bao gồm chất lượng bên ngoài và bên trong. Chúng tôi đã đề xuất phương pháp và công thức tính chất lượng bên ngoài chính là giá trị chung của bài khảo sát, và chất lượng bên trong chúng tôi giữ nguyên cách tính toán của đề tài trước. Người thiết kế quy trình nghiệp vụ vẫn có thể thiết lập mức độ quan trọng của từng loại chất lượng, và với phương pháp MUSA, chúng tôi đề xuất công thức tính giá trị *Quality* của cả quy trình nghiệp vụ như sau:

$$Q = w_{eq} \times \text{Survey Score} + w_{iq} \times \text{Internal Quality}$$



$$w_{eq} + w_{iq} = 1$$

Trong đó,  $w_{eq}$ ,  $w_{iq}$  lần lượt là trọng số của *external quality* và *internal quality*; Survey Score là điểm của bài khảo sát cũng như là giá trị của *external quality*. Vì 2 giá trị này có chung một miền giá trị  $[0, 1]$  nên không cần phải chuẩn hóa chúng trước khi đưa vào tính toán nữa. Trọng số này có thể được quy định bởi người thiết kế quy trình nghiệp vụ, tùy thuộc vào độ ưu tiên của họ đối với yếu tố nào của chất lượng để đánh giá chất lượng tổng quan của quy trình nghiệp vụ.

#### 1.4.6 Đề xuất bảng khảo sát

##### 1.4.6.1 Đối tượng khảo sát

Đối tượng được nhắm đến để thực hiện khảo sát bao gồm những người tham gia vào vận hành, thực thi quy trình nghiệp vụ ở những vai trò khác nhau trong quy trình; và những người là khách hàng của quy trình, chỉ tương tác với quy trình thông qua một phần của quy trình. Như vậy, kết quả chất lượng của quy trình trở nên chính xác và khách quan hơn, vì không chỉ được đánh giá thông qua việc thực thi quy trình trên thực tế, mà còn được đánh giá qua sản phẩm đầu ra của quy trình đó.

##### 1.4.6.2 Nội dung khảo sát

Để đánh giá được toàn diện chất lượng bên ngoài của một quy trình nghiệp vụ, cần phải xem xét đến ba độ đo CES, CSAT, NPS đã được trình bày ở trên. Vì thế, các câu hỏi dùng để thu thập giá trị của những độ đo này là bắt buộc và người thiết kế khảo sát không thể loại trừ chúng ra khỏi bộ câu hỏi. Bài khảo sát tập trung đánh giá trải nghiệm, sự hài lòng của người dùng nên ở đây, nội dung câu hỏi sẽ xoay quanh các yếu tố liên quan đến trải nghiệm của người dùng, đó là thời gian, chi phí, khả năng xử lý ngoại lệ và độ linh hoạt của quy trình.

Tuy nhiên, tùy vào loại quy trình mà có sự xuất hiện của đối tượng khách hàng của quy trình hay không. Chẳng hạn như, những quy trình đóng, phục vụ nội bộ thì có thể không xuất hiện đối tượng khách hàng, hay không có một sản phẩm đầu ra cụ thể nào phục vụ cho khách hàng bên ngoài. Vì vậy, người thiết kế bảng khảo sát cần xác định rõ đối tượng khảo sát để tùy chỉnh nội dung bảng khảo sát phù hợp, tránh xảy ra tình trạng khảo sát không đúng đối tượng, gây nhầm lẫn và sai lệch kết quả thu được.

Dựa vào những nguyên tắc thiết kế bảng câu hỏi đã được trình bày ở trên, chúng tôi quyết định chọn trình tự các câu hỏi như sau:

- Mở đầu bằng những câu hỏi rẽ nhánh, lấy thông tin cơ bản của người dùng, với mục đích lấy kết quả trả lời của người dùng ở những câu hỏi này để điều hướng tới các câu hỏi phù hợp. Thông tin cơ bản của người dùng chỉ bao gồm vai trò cụ thể của người dùng trong quy trình nghiệp vụ đó là gì, là một dạng thông tin công khai.
- Nhóm các câu hỏi chung chủ đề lại với nhau, bao gồm các nhóm câu hỏi về CES, CSAT, NPS tương ứng với ba độ đo đã đề cập, với thứ tự như trên, với mục đích sau khi đã tìm hiểu những nỗ lực của người dùng khi thực thi quy trình, sẽ đánh giá sự hài lòng tổng quan đối với quy trình, và cuối



cùng là liệu người dùng có muốn đề xuất quy trình cho người khác hay không. Ở trong mỗi nhóm câu hỏi, sẽ đi từ những câu hỏi đánh giá tổng quan trước (được dùng để tính điểm độ đo), sau đó là những câu hỏi để lấy thêm thông tin xoay quanh lựa chọn của người dùng ở những câu hỏi đánh giá tổng quan đó.

#### 1.4.6.3 Hình thức câu hỏi

Chúng tôi sẽ sử dụng một số hình thức câu hỏi khác nhau. Điều này cho phép chúng tôi có thể thu thập nhiều hơn thông tin từ người dùng. Dưới đây là bảng các hình thức và tên câu hỏi được sử dụng trong bài khảo sát:

Hình thức	Mô tả	Ký hiệu
Multiple choice (single select) question	Nhiều lựa chọn, nhưng một lúc chỉ chọn nhiều nhất một đáp án	MC - SS
Multiple choice (multiple select) question	Nhiều lựa chọn, cùng một lúc có thể chọn nhiều đáp án khác nhau	MC - MS
Open question	Cho phép nhập câu trả lời tùy ý, dưới dạng một đoạn văn ngắn hoặc câu trả lời ngắn	OP
Likert Scale question	Dùng để đánh giá CES, NPS, CSAT với các mức độ khác nhau	LS
Dichotomous	Hai lựa chọn, cùng lúc chỉ chọn một đáp án	DM

Chúng tôi cũng đặt tên các câu hỏi theo mục đích của chúng.

Tên	Mô tả	Ký hiệu
CES question	Câu hỏi đánh giá điểm CES	CES
CSAT question	Câu hỏi đánh giá điểm CSAT	CSAT
CES insight question	Câu hỏi thu thập thêm thông tin từ việc đánh giá CES của người dùng	CES - IN
CSAT insight question	Câu hỏi thu thập thêm thông tin từ việc đánh giá CSAT của người dùng	CSAT - IN
NPS question	Câu hỏi đánh giá điểm NPS	NPS
User-information-collecting question	Câu hỏi thu thập thông tin cơ bản của người dùng	UIC
Branching question	Câu hỏi điều kiện, mục đích điều hướng người dùng tới những câu hỏi khác phù hợp với câu trả lời của họ ở câu hỏi này	BR

#### 1.4.6.4 Nội dung câu hỏi

##### Câu hỏi điều kiện:

- Do you use any products or services provided by this process?
- Are there any handled exceptions in your process?
- Variations refer to process' workflows in different conditions. Are there any variations in your process?

Các câu hỏi điều kiện được sử dụng để xác định chi tiết đối tượng của bài khảo sát, liệu trong phần quy trình của họ có xảy ra những ngoại lệ nào đã được xử lý hay không, hay có bất kỳ những biến thể của quy trình hay không. Với những thông tin cung cấp từ người dùng sẽ điều hướng họ tới những câu hỏi sau đó phù hợp với câu trả lời ở những câu hỏi này.

##### Câu hỏi lấy thông tin cơ bản của người dùng:

- What is your role in this process?

Câu hỏi này dùng để xác định xem vai trò cụ thể của người dùng (lane) trong quy trình nghiệp vụ là gì, sau khi người dùng được phân loại thuộc nhóm những người tham gia vận hành, thực thi quy trình.

##### Câu hỏi CES

- To what extent do you agree with this statement: "It is easy to handle exceptions in your process".
- To what extent do you agree with this statement: "It is easy to execute your process".

Hai câu hỏi CES tập trung đánh giá nỗ lực của người dùng trong việc thực thi quy trình nói chung, và xử lý những ngoại lệ xảy ra trong quá trình thực thi đó. Người dùng đồng ý với mệnh đề ở mức độ nào sẽ chọn giá trị tương ứng trong những lựa chọn được cung cấp sẵn. Ngoài ra, để thu thập thêm thông tin xoay quanh những ngoại lệ xảy ra trong quy trình, chúng tôi cung cấp thêm một số câu hỏi theo sau (follow-up question), nhưng những câu hỏi này không được sử dụng để tính điểm khảo sát:

- How frequently do exceptions happen in practice?
- Which handled exceptions do you think the way to handle them needs optimizing? Can you suggest some solutions to those?
- Can you suggest any exceptions that need handling and the way to handle them?

Các câu hỏi trên tìm hiểu nhiều hơn về việc người dùng nhận thấy những ngoại lệ trên thực tế xảy ra với tần suất như thế nào. Trong số những ngoại lệ đã được xử lý bởi người thiết kế quy trình, người dùng có cảm thấy cách xử lý nào chưa được tối ưu hay vẫn còn một số vấn đề như quá phức tạp, tốn nhiều thời gian không, và họ có thể đề xuất những giải pháp khác. Bên cạnh đó, người dùng có gặp phải những ngoại lệ mà chưa được xử lý không, và trong những trường hợp đó, người dùng đã làm gì để giải quyết tạm thời. Sau khi thu thập được những thông tin này, người thiết kế quy trình nghiệp vụ có thể đưa ra được các phương án để tối ưu quy trình và xử lý các ngoại lệ có thể có.

#### **Câu hỏi CSAT**

- How satisfied are you with the way exceptions are handled in your process?
- A flexible process means it has suitable variations for different conditions. How satisfied are you with those variations of the process?
- How satisfied are you with the amount of time you have to spend on this process?
- How satisfied are you with the cost of products or services of this process?
- How satisfied are you with the products or services provided by this process?
- How satisfied are you with the total time you have to spend on waiting for the output of this process?

Các câu hỏi CSAT tập trung đo lường mức độ hài lòng của người dùng ở cách các ngoại lệ đã được xử lý trong quy trình, việc quy trình có những biến thể để linh hoạt đáp ứng các điều kiện khác nhau xảy ra trong quá trình thực thi, thời gian, chi phí mà người dùng phải bỏ ra để thực thi quy trình, cũng như kết quả đầu ra của quy trình. Người dùng cảm thấy hài lòng tới mức độ nào thì có thể chọn giá trị tương ứng trong những lựa chọn được cung cấp sẵn. Ngoài ra, chúng tôi đề xuất thêm câu hỏi theo sau để tìm hiểu xem liệu người dùng có gặp vấn đề khi sử dụng sản phẩm hoặc dịch vụ mà quy trình nghiệp vụ cung cấp hay không. Từ đó, người thiết kế quy trình nghiệp vụ có thể đưa ra những giải pháp để nâng cao chất lượng sản phẩm, dịch vụ của quy trình. Và câu hỏi này không được dùng để tính điểm CSAT:

- Do you encounter any problems while using products or services provided by this process?

#### **Câu hỏi NPS**

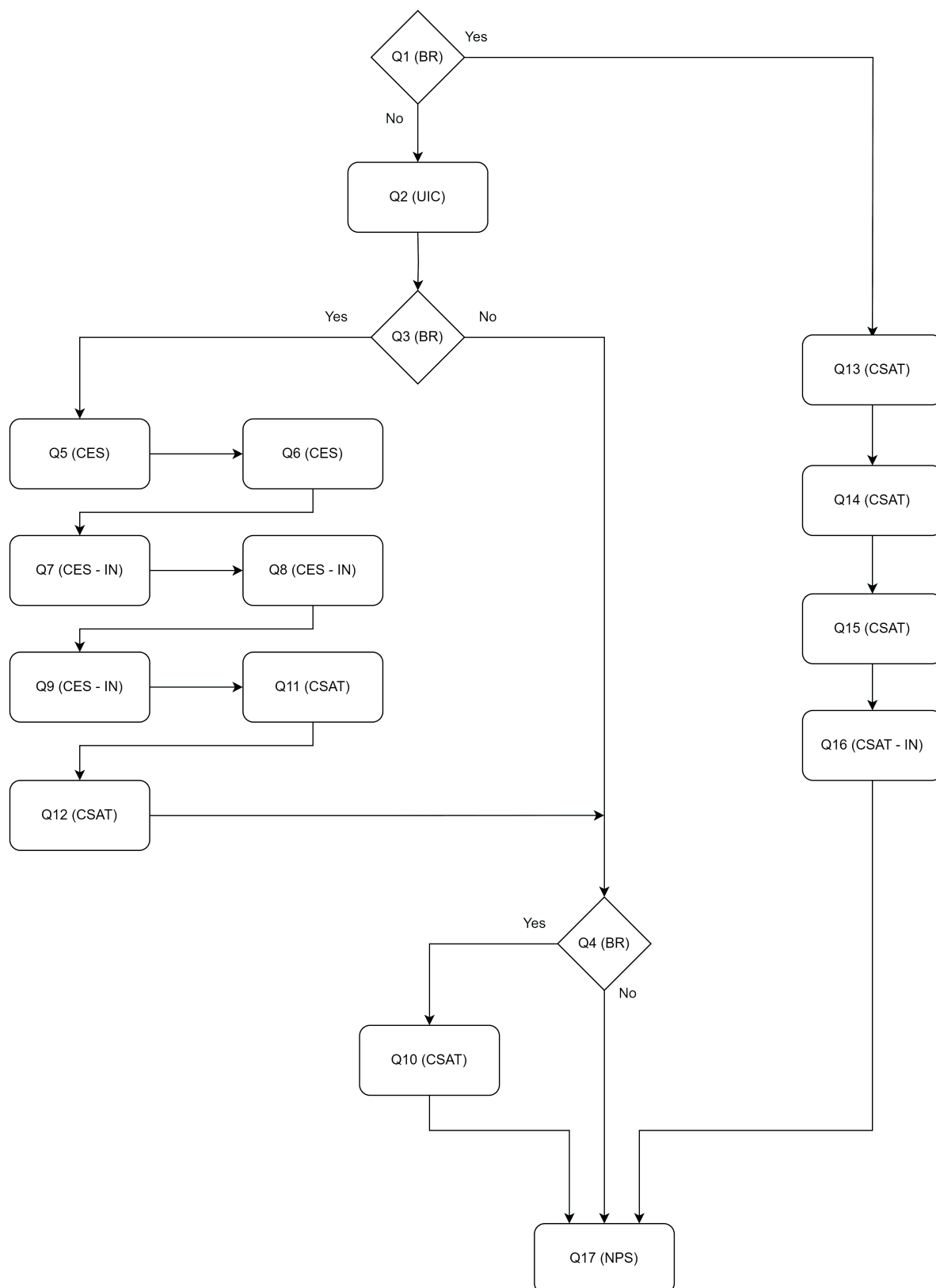
- How likely is it that you would recommend this process to others?

Để chốt lại bảng khảo sát, câu hỏi NPS sẽ được sử dụng để đánh giá liệu người dùng có cảm thấy quy trình nghiệp vụ xứng đáng để được giới thiệu cho những người khác hay không. Người dùng có thể chọn câu trả lời phù hợp với đánh giá và thái độ của họ.

Để có cái nhìn tổng quan nhất về những câu hỏi trong bảng khảo sát, chúng tôi xin cung cấp bảng tổng hợp các câu hỏi cũng như trình tự các câu hỏi được thể hiện dưới dạng biểu đồ như sau:

Câu hỏi	Hình thức	Thang điểm	Tên	Nội dung	Bản dịch (tiếng Việt)
Q1	DM		BR	Do you use any products or services provided by this process?	Bạn có sử dụng sản phẩm hay dịch vụ nào được cung cấp bởi quy trình này không?
Q2	MC - SS		UIC	What is your role in this process?	Vai trò của bạn trong quy trình này là gì?
Q3	DM		BR	Are there any handled exceptions in your process?	Trong quy trình của bạn có xảy ra ngoại lệ nào đã được xử lý không?
Q4	DM		BR	Variations refer to process' workflows in different conditions. Are there any variations in your process?	Các biến thể là các luồng thực thi ở những điều kiện khác nhau của quy trình. Trong quy trình của bạn, có xảy ra biến thể nào của quy trình hay không?
Q5	LS	1 - 7	CES	To what extent do you agree with this statement: "It is easy to handle exceptions in your process".	Bạn đồng ý với mệnh đề này tới mức độ nào: "Bạn cảm thấy dễ dàng để xử lý các ngoại lệ trong quy trình nghiệp vụ".
Q6	LS	1 - 7	CES	To what extent do you agree with this statement: "It is easy to execute your process."	Bạn đồng ý với mệnh đề này tới mức độ nào: "Bạn cảm thấy dễ dàng thực thi quy trình nghiệp vụ".
Q7	MC - SS		CES - IN	How frequently do exceptions happen in practice?	Hãy chọn mức độ thường xuyên xảy ra ngoại lệ trên thực tế theo đánh giá của bạn.
Q8	OP		CES - IN	Which handled exceptions do you think the way to handle them needs optimizing? Can you suggest some solutions to those?	Trong số những ngoại lệ đã được xử lý, bạn nghĩ cái nào cách xử lý của nó vẫn chưa được tối ưu? Bạn có thể đề xuất một số cách tối ưu của bạn cho ngoại lệ đó được không?
Q9	OP		CES - IN	Can you suggest any exceptions that need handling and the way to handle them?	Bạn có thể đề xuất một số những ngoại lệ cần phải được xử lý và cách xử lý chúng không?

Q10	LS	1 - 7	CSAT	A flexible process means it has variations for different conditions. How satisfied are you with those variations of the process?	Một quy trình được xem là linh hoạt khi nó có nhiều biến thể cho nhiều điều kiện khác nhau. Bạn hài lòng như thế nào với những biến thể này của quy trình?
Q11	LS	1 - 7	CSAT	How satisfied are you with the way exceptions are handled in your process?	Bạn hài lòng như thế nào với cách mà quy trình nghiệp vụ xử lý các ngoại lệ?
Q12	LS	1 - 7	CSAT	How satisfied are you with the amount of time you have to spend on this process?	Bạn hài lòng như thế nào với tổng thời gian bạn dành cho quy trình?
Q13	LS	1 - 7	CSAT	How satisfied are you with the cost of products or services of this process?	Bạn hài lòng như thế nào với chi phí của sản phẩm, dịch vụ của quy trình?
Q14	LS	1 - 7	CSAT	How satisfied are you with the products or services provided by this process?	Bạn hài lòng như thế nào với sản phẩm, dịch vụ quy trình cung cấp?
Q15	LS	1 - 7	CSAT	How satisfied are you with the total time you have to spend on waiting for the output of this process?	Bạn hài lòng như thế nào với thời gian chờ sản phẩm hay dịch vụ của quy trình?
Q16	OP		CSAT - IN	Do you encounter any problems while using products or services provided by this process?	Bạn có gặp vấn đề gì trong quá trình sử dụng sản phẩm, dịch vụ của quy trình không?
Q17	LS	0 - 10	NPS	How likely is it that you would recommend this process to others?	Bạn có muốn đề xuất quy trình nghiệp vụ này cho người khác không?



**Hình 1:** Trình tự câu hỏi trong bảng khảo sát

## 2 Phân tích và thiết kế hệ thống

### 2.1 Giới thiệu hệ thống

Ở đề tài trước, hệ thống (tạm gọi là phiên bản 1.0) đã được hiện thực với một số tính năng quan trọng, chẳng hạn như: thiết kế, so sánh, mô phỏng và đánh giá mô hình BPMN. Ngoài việc cung cấp bộ công cụ đầy đủ các thành phần BPMN 2.0, phiên bản 1.0 này của hệ thống còn hỗ trợ nhập xuất file BPMN, ghi chú trên các thành phần và mô hình, các tính năng undo, redo, zoom-in, zoom-out,... Trong quá trình thiết kế, người dùng cũng có thể xác thực mô hình BPMN về mặt ngữ nghĩa cũng như so sánh các phiên bản khác nhau của mô hình.

Nhóm chúng tôi nhận ra rằng tính năng quản lý mô hình BPMN của người dùng vẫn chưa được chú trọng. Các dự án của người dùng chỉ mới được đóng gói trong các đơn vị gọi là Project, và chỉ mới thuộc về riêng một cá nhân sở hữu project đó. Tuy nhiên, chúng tôi nhận thấy rằng, khi hệ thống được mở rộng, đồng nghĩa với việc số lượng người dùng tăng lên, người ta sẽ có xu hướng nhóm các dự án lại với nhau thành từng cụm lớn hơn. Trong những cụm đó, những người dùng có thể làm việc với nhau trên các project bên trong mỗi cụm, giúp phân tách và quản lý dự án hiệu quả hơn. Chúng tôi gọi đó là tính năng Workspace.

Workspace được cụ thể hoá bởi các tính năng sau đây:

- Quản lý thành viên (members). Một Workspace có thể có một hoặc nhiều người cùng tham gia. Mỗi người dùng đều có một Workspace cá nhân của bản thân, gọi là Personal Workspace. Personal Workspace là riêng tư, nên không thể mời người khác tham gia vào Personal Workspace của bản thân. Người dùng có thể tự tạo Workspace khác (tạm gọi là Regular Workspace) để có thể làm việc với nhiều người cho các dự án ở trong Workspace đó. Chủ sở hữu Workspace có thể mời người khác tham gia vào Workspace của mình, cũng như thiết lập một số quyền truy cập của thành viên, hay loại bỏ thành viên khỏi Workspace.
- Quản lý dự án (projects). Các Projects sẽ được tạo trong Workspace, cho phép các thành viên trong Workspace có thể truy cập các project liên quan tới Workspace đó. Bất kỳ thành viên nào của Workspace với bất kỳ quyền hạn nào cũng có thể tạo project của riêng mình trong các Workspace mà mình tham gia, và mời các thành viên khác vào project được tạo để cùng nhau phát triển. Các projects cũng có thể bị xoá khỏi Workspace tùy theo mục đích và nhu cầu.
- Quản lý yêu cầu (requests). Khi hiện thực chức năng Workspace, nhóm chúng tôi để tâm tới tính đóng mở của Workspace. Chúng tôi cho rằng Workspace nên đủ đóng để bảo vệ sự toàn vẹn dữ liệu bên trong và đảm bảo sự riêng tư. Tuy nhiên Workspace nên đủ mở để nhiều thành viên khác trong hệ thống có thể cùng tham gia và phát triển, xây dựng hệ sinh thái trong đó. Chính vì vậy, quản lý requests là nhóm các tính năng cho phép chủ sở hữu Workspace có thể quản lý được những luồng tham gia vào Workspace hay những yêu cầu muốn thay đổi quyền hạn của bản thân thành viên để có thêm một số quyền hạn đối với các dữ liệu trong Workspace. Chủ sở hữu có thể chấp thuận, từ chối hay xoá bỏ những yêu cầu nào từ phía thành viên. Khi một request được chấp thuận



hay từ chối, sẽ có một thông báo gửi về cho đối tượng của request đó. Bên cạnh đó, chủ sở hữu Workspace cũng sẽ nhận được các requests theo thời gian thực, ngay khi thành viên gửi đi bất kỳ yêu cầu nào.

- Cá nhân hoá Workspace. Hệ thống cho phép người dùng linh hoạt tùy chỉnh một số thuộc tính của Workspace như tên, mô tả, logo biểu trưng của Workspace nhằm tăng trải nghiệm cá nhân và phù hợp với mục đích của Workspace đó.

Bên cạnh tính năng Workspace, nhóm chúng tôi còn phát triển thêm tính năng thông báo:

- Quản lý các thông báo cá nhân. Mỗi người dùng sẽ có một hộp thông báo của riêng mình. Ở đó, các thông báo sẽ được tổng hợp để người dùng có thể theo dõi. Các loại thông báo bao gồm: thông báo khi có thành viên mới tham gia Workspace, thông báo khi yêu cầu thay đổi quyền hạn của bản thân trong Workspace tham gia được chấp thuận hay từ chối, thông báo là lời mời tham gia Workspace từ một thành viên trong Workspace đó. Người dùng có thể chấp thuận hoặc từ chối lời mời này. Bên cạnh đó, người dùng có thể thực thi các thao tác xoá, lọc, đánh dấu những thông báo quan trọng.
- Nhận, gửi thông báo theo thời gian thực. Thông thường, để luôn nhận được thông báo mới nhất, người dùng cần phải tải lại trang định kỳ để hệ thống trả về các thông báo mới. Tuy nhiên, nhóm chúng tôi đã hiện thực tính năng thông báo này theo thời gian thực, nghĩa là khi có một thông báo mới, nó sẽ được hiển thị đồng thời, ngay lập tức cho người dùng, để họ dù đang làm gì trên hệ thống vẫn có thể theo dõi được các yêu cầu của mình có được chấp thuận hay chưa, hay nhận lời mời tham gia Workspace ngay lập tức. Việc hiện thực theo thời gian thực giúp tăng khả năng phản hồi và trải nghiệm của người dùng khi sử dụng hệ thống.

Trong thời gian tới, nhóm chúng tôi sẽ tập trung phát triển tính năng Template Suggestions. Chúng tôi đang hướng tới việc mở hoá hệ thống, nghĩa là cho phép người dùng có thể công khai một số project đến cho cộng đồng để được đóng góp và phát triển thành những sản phẩm tiềm năng, tối ưu. Những người dùng khác có thể tham gia phát triển, tạo bản sao đem về project cá nhân của họ và đánh giá những dự án chất lượng cao nhằm xây dựng tài nguyên các mô hình BPMN và đưa ra các đề xuất phù hợp với nhu cầu của người dùng khi họ cần tìm kiếm hay khởi tạo dự án nào trong hệ thống, họ có thể tham khảo tới những mẫu mô hình được đánh giá cao đã có sẵn.

Các yêu cầu phi chức năng cũng là yếu tố quan trọng cần được đánh giá, xem xét. Các yêu cầu phi chức năng dưới đây được chúng tôi đảm bảo hệ thống có thể thực hiện được nhằm tối ưu hoá trải nghiệm của người dùng, bên cạnh các yêu cầu phi chức năng kế thừa từ đề tài trước đó:

Performance	<ul style="list-style-type: none"><li>– Đối với màn hình input: tối đa 5 trường dữ liệu, không tính toán dữ liệu phức tạp, không tương tác với hệ thống ngoài, có thể lưu trữ dữ liệu trực tiếp ngay xuống database, không lưu trữ các tệp nội dung lớn như hình ảnh, video, tập tin quá 3 MB.</li><li>– Đối với màn hình output: dữ liệu được truy vấn trực tiếp từ database, hạn chế những câu lệnh truy vấn phức tạp, những truy vấn từ hệ thống ngoài. Hiển thị tối đa 10 dòng dữ liệu, có độ dài nhỏ hơn 100 ký tự</li></ul>
Usability	<ul style="list-style-type: none"><li>– Tất cả thông tin quan trọng phải được hiển thị trong 1 màn hình (không cần phải thực hiện thêm thao tác cuộn).</li><li>– Giao diện của hệ thống cần có sự nhất quán, về mặt hình ảnh biểu tượng cũng như vị trí các đối tượng trên màn hình để người dùng làm quen dễ dàng hơn.</li><li>– Người dùng có thể thành thạo các thao tác trên màn hình trong 15 phút sử dụng.</li></ul>
Supportability	<ul style="list-style-type: none"><li>– Hệ thống có thể được sử dụng hiệu quả trên các trình duyệt web (Opera, UC Browser, Safari, Microsoft Edge, Google Chrome, Samsung Internet Browser).</li><li>– Hệ thống không chạy trên các phiên bản trình duyệt quá cũ.</li></ul>
Scalability	Có khả năng tách database trên một server riêng và backend trên một server riêng

## 2.2 Kiến trúc phần mềm

Với những nghiệp vụ được đề cập ở trên, chúng tôi quyết định lựa chọn hướng tiếp cận của kiến trúc đa lớp cho hệ thống của chúng tôi. Kiến trúc đa lớp là một trong những kiến trúc phổ biến trong các loại kiến trúc phần mềm. Kiến trúc này ra đời nhằm phân chia các thành phần trong hệ thống, các thành phần cùng chức năng sẽ được nhóm lại với nhau và phân chia công việc cho từng nhóm để dữ liệu không bị chồng chéo và lộn xộn. Kiến trúc này phát huy hiệu quả nhất ở các hệ thống từ nhỏ đến lớn, giúp cho việc quản lý code và xử lý dữ liệu lỗi dễ dàng hơn.

Mặc dù không có quy định cụ thể về số lượng hay kiểu của các lớp, ở những hệ thống phức tạp hơn có thể có nhiều lớp hơn, đa số các kiến trúc đa lớp gồm có ba lớp chuẩn (3-tier): *Presentation Layer*, *Business*

*Layer, Data Layer. Presentation layer* có nhiệm vụ chính giao tiếp với người dùng, gồm các thành phần giao diện và thực hiện các công việc như nhập liệu, hiển thị dữ liệu, kiểm tra tính đúng đắn của dữ liệu trước khi gọi lớp *Business layer*. *Business layer* phân ra thành hai nhiệm vụ: thứ nhất, đây là nơi đáp ứng các yêu cầu thao tác dữ liệu của *Presentation layer*, xử lý chính nguồn dữ liệu từ *Presentation layer* trước khi truyền xuống *Data layer* và lưu xuống hệ quản trị cơ sở dữ liệu. *Data layer* có chức năng giao tiếp với hệ quản trị CSDL như thực hiện các công việc liên quan đến lưu trữ và truy vấn dữ liệu ( tìm kiếm, thêm, xóa, sửa,... ).

Kiến trúc này có một số ưu điểm như sau:

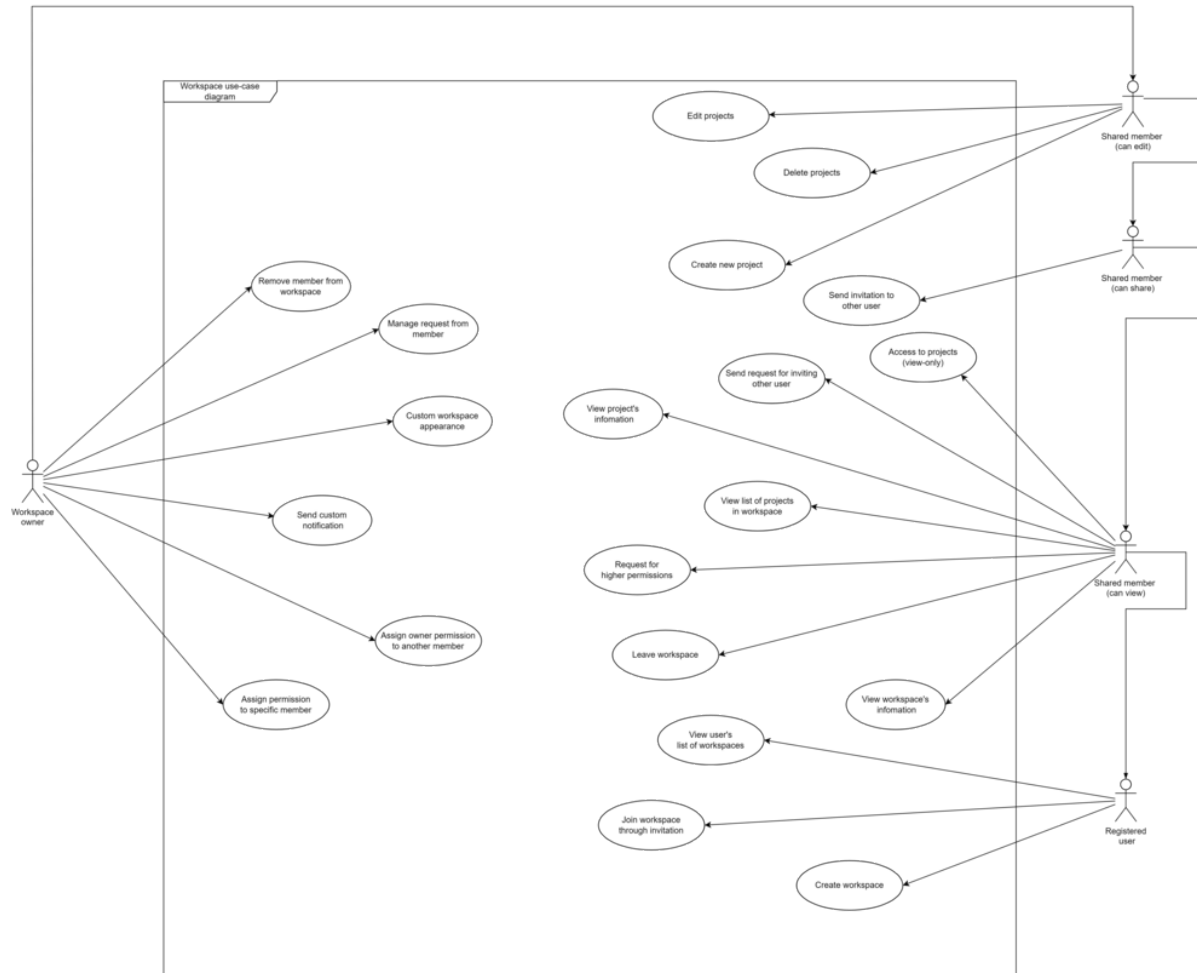
- Việc phân chia thành từng lớp giúp cho code được tường minh hơn. Nhờ vào việc chia ra từng lớp đảm nhận các chức năng khác nhau và riêng biệt như giao diện, xử lý, truy vấn thay vì để tất cả lại một chỗ, nhằm giảm sự kết dính.
- Dễ bảo trì khi được phân chia, thì một thành phần của hệ thống sẽ dễ thay đổi. Việc thay đổi này có thể được cô lập trong 1 lớp, hoặc ảnh hưởng đến lớp gần nhất mà không ảnh hưởng đến cả chương trình.
- Dễ phát triển, tái sử dụng: khi chúng ta muốn thêm một chức năng nào đó thì việc lập trình theo một mô hình sẽ dễ dàng hơn vì chúng ta đã có chuẩn để tuân theo. Và việc sử dụng lại khi có sự thay đổi giữa hai môi trường thì chỉ việc thay đổi lại *Presentation layer*.

Ngoài kiến trúc đa lớp này, chúng tôi còn tìm hiểu thêm một loại kiến trúc cũng rất phổ biến hiện nay, đó là *Microservices*. Theo kiến trúc này, một ứng dụng được chia thành một bộ các *microservice*, mỗi *microservice* thực chất là một service có thể được triển khai và chạy độc lập. Chúng tách biệt về mặt mã nguồn, về hoạt động và dữ liệu. Mỗi *microservice* có nơi chứa dữ liệu của riêng của nó và chỉ có nó có quyền truy cập vào vùng dữ liệu này. Do các *microservice* là độc lập, chúng không giao tiếp trực tiếp với nhau mà qua một thành phần trung gian được gọi là *API gateway*. Có thể thấy vai trò của *API gateway* rất quan trọng trong mô hình *microservice*. Nó là điểm đến và đi của mọi yêu cầu hay phản hồi.

Tính phân tán là đặc trưng của kiến trúc này, vì vậy việc xác định mức độ chi tiết của từng dịch vụ là chìa khóa để xây dựng một hệ thống tốt. Đó là điều khó đạt được đối với hệ thống đang phát triển của chúng tôi. Một vấn đề khác với hướng tiếp cận theo kiến trúc này đó là việc các nghiệp vụ trong hệ thống chúng tôi liên quan khá chặt chẽ, vì vậy kiến trúc *Microservices* tỏ ra không phù hợp với hệ thống của chúng tôi.

## 2.3 Use-case

### 2.3.1 Lược đồ use-case



### 2.3.2 Đặc tả use-case

#### 2.3.3 Create new workspace

<b>Use-case name</b>	Create new workspace
<b>Actor</b>	Registered user
<b>Description</b>	Người dùng tạo mới workspace
<b>Trigger</b>	Không
<b>Pre-conditions</b>	Người dùng đã đăng ký vào hệ thống và đăng nhập thành công, hiện tại đang ở giao diện Default homepage
<b>Post-conditions</b>	Người dùng tạo mới workspace thành công, và trở thành workspace owner của workspace đó
<b>Normal flow</b>	<ol style="list-style-type: none"><li>1. Người dùng ở giao diện Default homepage, hiện đang hiển thị danh sách workspace của người dùng đó</li><li>2. Người dùng chọn nút "Create workspace"</li><li>3. Hệ thống mở modal yêu cầu thông tin của workspace: Workspace name, workspace description</li><li>4. Người dùng nhập tên (Name) của workspace</li><li>5. Người dùng nhập mô tả (Description) của workspace</li><li>6. Người dùng chọn nút "Save" để lưu thông tin và hoàn tất tạo mới workspace</li></ol>
<b>Alternative flows</b>	Không
<b>Exceptions</b>	<ol style="list-style-type: none"><li>2a. Người dùng chọn nút "Cancel" để đóng modal nhập thông tin workspace</li><li>3a. Người dùng sử dụng tên của workspace đã tồn tại<ol style="list-style-type: none"><li>3a.1. Hệ thống yêu cầu người dùng sửa lại tên phù hợp</li><li>3a.2. Tiếp tục bước 4</li></ol></li><li>3b. Người dùng không nhập tên của workspace<ol style="list-style-type: none"><li>3b.1. Hệ thống yêu cầu người dùng phải điền thông tin bắt buộc</li><li>3b.2. Tiếp tục bước 4</li></ol></li></ol>

### 2.3.4 Join workspace through invitation

<b>Use-case name</b>	Join workspace through invitation
<b>Actor</b>	Registered user
<b>Description</b>	Tham gia workspace thông qua “Lời mời tham gia workspace”
<b>Trigger</b>	<ol style="list-style-type: none"><li>1. Thành viên trong workspace với quyền Chia sẻ hoặc cao hơn gửi lời mời trực tiếp đến người dùng trên hệ thống và chưa tồn tại trong workspace</li><li>2. Workspace owner đồng ý xét duyệt yêu cầu chia sẻ của thành viên không có quyền Chia sẻ trong workspace đến người dùng trên hệ thống</li><li>3. Workspace owner gửi lời mời tham gia trực tiếp đến người dùng trên hệ thống</li></ol>
<b>Pre-conditions</b>	Người dùng đã đăng ký vào hệ thống và đăng nhập thành công
<b>Post-conditions</b>	Người dùng thành công tham gia vào workspace
<b>Normal flow</b>	<ol style="list-style-type: none"><li>1. Người dùng chọn icon “Notification” trên thanh điều hướng (navigation bar)</li><li>2. Hệ thống chuyển hướng đến trang Notification (thông báo) của người dùng</li><li>3. Hệ thống hiển thị danh sách những thông báo của người dùng</li><li>4. Người dùng chọn vào thông báo “Lời mời tham gia workspace” để mở modal hiển thị thông tin chi tiết của Lời mời</li><li>5. Người dùng chọn “Accept” để chấp nhận lời mời tham gia vào workspace</li></ol>
<b>Alternative flows</b>	6a. Người dùng chọn “Decline” để từ chối lời mời tham gia workspace
<b>Exceptions</b>	<p>6a. Người dùng chọn “Accept” với những lời mời đã hết hiệu lực</p> <p>6a.1. Hệ thống hiển thị lỗi khi xử lý yêu cầu của người dùng</p> <p>6a.2. Hệ thống đóng modal hiển thị thông tin Lời mời vào workspace</p>

### 2.3.5 View user's list of workspaces

<b>Use-case name</b>	View user's list of workspaces
<b>Actor</b>	Registered user
<b>Description</b>	Xem danh sách những workspaces đã tham gia của người dùng
<b>Trigger</b>	Không
<b>Pre-conditions</b>	Người dùng đã đăng ký vào hệ thống và đăng nhập thành công
<b>Post-conditions</b>	Người dùng xem được danh sách những workspaces đã tham gia
<b>Normal flow</b>	<ol style="list-style-type: none"><li>1. Người dùng hiện tại đang ở giao diện Default homepage</li><li>2. Người dùng chọn tab "Recently opened" để xem danh sách những workspace đã tham gia được sắp xếp theo thời gian mở gần nhất</li></ol>
<b>Alternative flows</b>	2a. Người dùng chọn tab "Pin workspace" để xem danh sách những workspace được ghim
<b>Exceptions</b>	Không

### 2.3.6 Access to projects' content in selected workspace

<b>Use-case name</b>	Access to projects' content in selected workspace
<b>Actor</b>	Workspace owner, workspace member
<b>Description</b>	Truy cập vào danh sách project bên trong workspace
<b>Trigger</b>	Không
<b>Pre-conditions</b>	Người dùng đã đăng ký vào hệ thống và đăng nhập thành công, hiện tại đang ở giao diện Workspace của workspace được chọn
<b>Post-conditions</b>	Người dùng xem nội dung project bên trong workspace được chọn
<b>Normal flow</b>	<ol style="list-style-type: none"><li>1. Người dùng ở giao diện Workspace của workspace được chọn</li><li>2. Hệ thống hiển thị cho người dùng xem được danh sách những project bên trong workspace</li><li>3. Người dùng chọn project tương ứng để xem nội dung bên trong dưới dạng danh sách xổ xuống (files, processes, documents,...)</li></ol>
<b>Alternative flows</b>	Không
<b>Exceptions</b>	Không



### 2.3.7 Send request for inviting other user

<b>Use-case name</b>	Send request for inviting other user
<b>Actor</b>	Workspace member
<b>Description</b>	Gửi yêu cầu “Chia sẻ workspace đến người khác” tới workspace owner để được xét duyệt
<b>Trigger</b>	Không
<b>Pre-conditions</b>	Người dùng đã đăng ký vào hệ thống và đăng nhập thành công
<b>Post-conditions</b>	Người dùng gửi thành công yêu cầu “Chia sẻ workspace đến người khác” tới workspace owner để được xét duyệt
<b>Normal flow</b>	<ol style="list-style-type: none"> <li>1. Người dùng ở giao diện Default homepage, hệ thống hiển thị danh sách những workspace mà người dùng tham gia/sở hữu</li> <li>2. Người dùng chọn icon "Menu" ở workspace mà người dùng tham gia với quyền hạn của viewer để hiển thị dropdown menu</li> <li>3. Người dùng chọn nút “Share” để hiện modal chia sẻ workspace: <ol style="list-style-type: none"> <li>(a) Những người hiện có trong workspace</li> <li>(b) Quyền hạn của những người hiện có trong workspace</li> </ol> </li> <li>4. Người dùng nhập email của người được mời để tìm kiếm người dùng trong hệ thống</li> <li>5. Người dùng chọn quyền hạn của người dùng được mời vào workspace</li> <li>6. Người dùng chọn “Send” để gửi yêu cầu đến workspace owner</li> </ol>
<b>Alternative flows</b>	<p>4a. Email của người được mời không tồn tại trên hệ thống</p> <p>4a.1. Hệ thống hiển thị không tìm thấy kết quả phù hợp</p> <p>5a. Người dùng gán quyền hạn cho người được mời vào workspace vượt quá quyền hạn hiện có trong workspace</p> <p>5a.1. Hệ thống thông báo lỗi yêu cầu người dùng gán quyền cho người được mời không vượt quá quyền hạn hiện có (Editor &gt; Sharer &gt; Viewer)</p> <p>5a.2. Tiếp tục ở bước 5</p>
<b>Exceptions</b>	5a. Người dùng chọn “Cancel” để hủy yêu cầu và đóng modal

### 2.3.8 Send request for adjust permission

<b>Use-case name</b>	Send request for adjust permission
<b>Actor</b>	Workspace member
<b>Description</b>	Gửi yêu cầu “Cung cấp thêm quyền” tới workspace owner để được xét duyệt
<b>Trigger</b>	Không
<b>Pre-conditions</b>	Người dùng đã đăng ký vào hệ thống và đăng nhập thành công
<b>Post-conditions</b>	Người dùng gửi thành công yêu cầu “Cung cấp điều chỉnh quyền” tới workspace owner để được xét duyệt
<b>Normal flow</b>	<ol style="list-style-type: none"><li>1. Người dùng ở giao diện của workspace đã chọn và đang tham gia vào workspace dưới quyền hạn của viewer hoặc sharer</li><li>2. Người dùng chọn nút “Create new project”, hệ thống hiện modal thông báo:<ol style="list-style-type: none"><li>(a) Người dùng không có quyền hạn để chỉnh sửa nội dung workspace</li><li>(b) Gửi yêu cầu đến workspace owner để điều chỉnh quyền trong workspace</li></ol></li><li>3. Người dùng chọn nút “Send” để gửi yêu cầu</li></ol>
<b>Alternative flows</b>	Không
<b>Exceptions</b>	3a. Người dùng chọn “Cancel” để đóng modal

### 2.3.9 Delete member in Workspace

<b>Use-case name</b>	Delete member in Workspace
<b>Actor</b>	Workspace owner
<b>Description</b>	Workspace owner xóa thành viên ra khỏi Workspace
<b>Trigger</b>	Không
<b>Pre-conditions</b>	Người dùng đăng nhập vào hệ thống, người dùng hiện tại đang ở giao diện workspace mà người dùng sở hữu
<b>Post-conditions</b>	Thành công xóa người dùng khỏi workspace
<b>Normal Flow</b>	<ol style="list-style-type: none"><li>1. Người dùng chọn icon "Management" bên cạnh tiêu đề workspace, hệ thống chuyển hướng đến trang quản lý workspace. Mặc định ở giao diện "Members management"</li><li>2. Hệ thống hiển thị danh sách thành viên trong workspace.</li><li>3. Người dùng chọn một hoặc nhiều thành viên. Hệ thống hiển thị nút "Delete"</li><li>4. Người dùng chọn nút "Delete", hệ thống hiển thị modal xác nhận thao tác xóa thành viên đã chọn khỏi workspace</li><li>5. Người dùng chọn "Delete" để loại trừ thành viên khỏi workspace</li></ol>
<b>Alternative Flow</b>	3a. Danh sách thành viên chỉ có workspace owner, người dùng không thể lựa chọn thành viên để thao tác
<b>Exceptions</b>	5a. Người dùng chọn "Cancel" để tắt modal

### 2.3.10 Manage requests from member

<b>Use-case name</b>	Manage requests from member
<b>Actor</b>	Workspace owner
<b>Description</b>	Workspace owner xét duyệt các yêu cầu từ thành viên trong Workspace
<b>Trigger</b>	Không
<b>Pre-conditions</b>	Người dùng đã đăng nhập vào hệ thống, người dùng hiện tại đang ở giao diện workspace mà người dùng sở hữu
<b>Post-conditions</b>	Thành công xét duyệt các yêu cầu từ người dùng trong workspace
<b>Normal Flow</b>	<ol style="list-style-type: none"><li>1. Người dùng chọn icon "Management" bên cạnh tiêu đề workspace, hệ thống chuyển hướng đến trang quản lý workspace. Mặc định ở giao diện "Members management"</li><li>2. Người dùng chọn "Requests management" ở thanh sidebar để hệ thống chuyển hướng tới trang quản lý yêu cầu của workspace.</li><li>3. Hệ thống hiển thị danh sách những yêu cầu từ người dùng trong workspace.</li><li>4. Người dùng chọn vào yêu cầu để mở modal hiển thị những thông tin của yêu cầu.</li><li>5. Người dùng chọn nút "Approve" trong modal để chấp nhận yêu cầu tương ứng.</li></ol>
<b>Alternative Flow</b>	5a. Người dùng chọn "Decline" trong modal để từ chối yêu cầu tương ứng.
<b>Exceptions</b>	5a. Người dùng chọn "Cancel" để hủy thao tác, hệ thống tắt modal xác nhận

### 2.3.11 Create new project

<b>Use-case name</b>	Create new project
<b>Actor</b>	Workspace owner, workspace member
<b>Description</b>	Workspace owner có thể tạo project trong Workspace
<b>Trigger</b>	Không
<b>Pre-conditions</b>	Workspace owner đã đăng nhập vào hệ thống, hiện tại đang ở giao diện workspace người dùng muốn tạo project. Người dùng có quyền hạn của editor hoặc owner.
<b>Post-conditions</b>	Tạo project thành công
<b>Normal Flow</b>	<ol style="list-style-type: none"><li>1. Người dùng chọn nút "Create project"</li><li>2. Hệ thống mở modal yêu cầu thông tin của project: project name, project description</li><li>3. Người dùng nhập tên (Name) của project</li><li>4. Người dùng nhập mô tả (Description) của project</li><li>5. Người dùng chọn nút "Save" để lưu thông tin và hoàn tất tạo mới project</li></ol>
<b>Alternative Flow</b>	Không
<b>Exceptions</b>	<ol style="list-style-type: none"><li>3a. Người dùng chọn nút "Cancel" để đóng modal nhập thông tin workspace</li><li>4a. Người dùng sử dụng tên của workspace đã tồn tại<ol style="list-style-type: none"><li>4a.1. Hệ thống yêu cầu người dùng sửa lại tên phù hợp</li><li>4a.2. Tiếp tục bước 4</li></ol></li><li>4b. Người dùng không nhập tên của workspace<ol style="list-style-type: none"><li>4b.1. Hệ thống yêu cầu người dùng phải điền thông tin bắt buộc</li><li>4b.2. Tiếp tục bước 4</li></ol></li></ol>

### 2.3.12 Delete project in Workspace

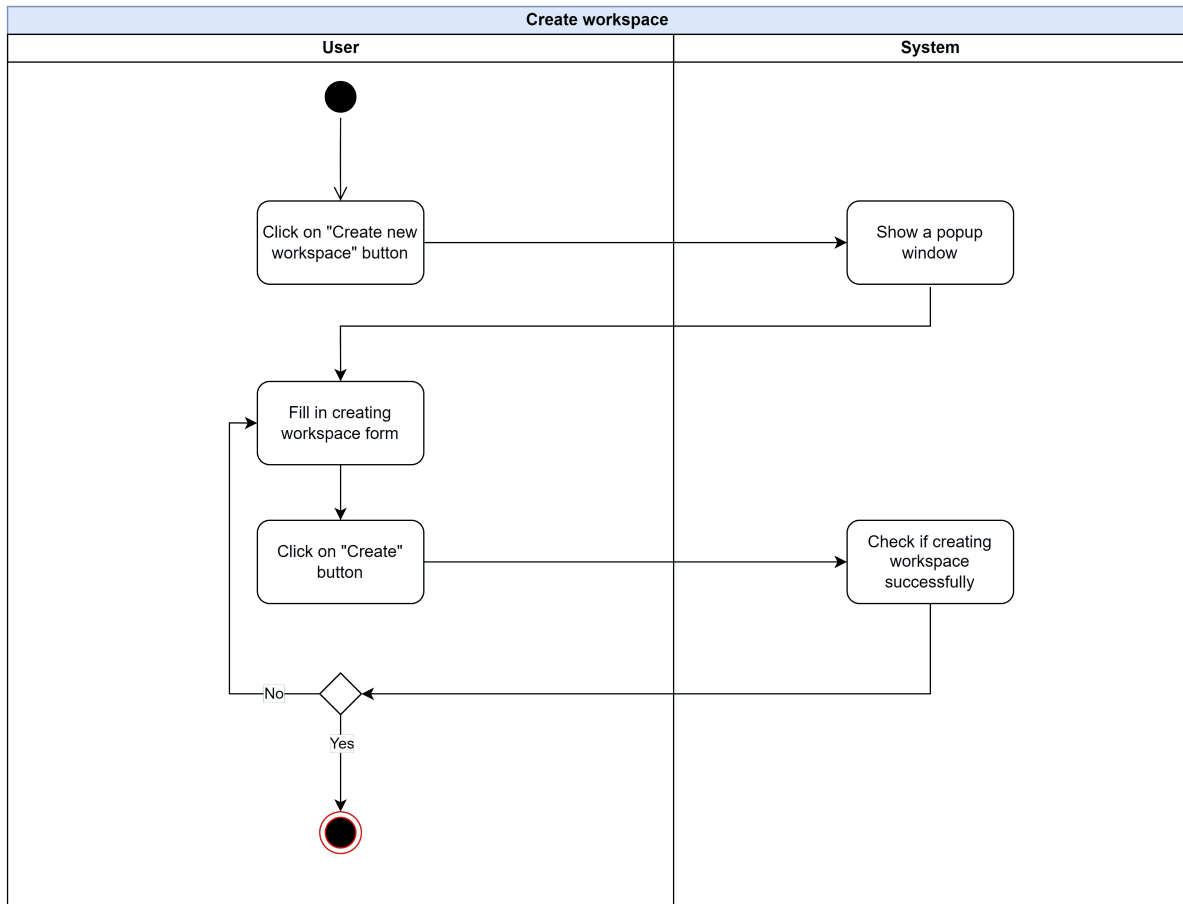
<b>Use-case name</b>	Delete project in Workspace.
<b>Actor</b>	Workspace owner, workspace member
<b>Description</b>	Người dùng có thể xóa project trong Workspace
<b>Trigger</b>	Không
<b>Pre-conditions</b>	Người dùng đã đăng nhập vào hệ thống và đang ở giao diện workspace đã chọn, người dùng có quyền hạn của editor hoặc owner
<b>Post-conditions</b>	Xóa project thành công
<b>Normal Flow</b>	<ol style="list-style-type: none"><li>1. Người dùng hiện tại đang ở giao diện workspace và hệ thống hiển thị danh sách project.</li><li>2. Người dùng chọn vào icon "Menu" của item project, hệ thống hiển thị danh sách menu thao tác với project</li><li>3. Người dùng chọn nút "Delete" trong menu</li><li>4. Hệ thống hiển thị modal yêu cầu người dùng xác nhận xóa project</li><li>5. Người dùng nhấn nút "Delete" để xác nhận xóa project</li></ol>
<b>Alternative Flow</b>	Không
<b>Exceptions</b>	5a. Người dùng chọn "Cancel" để hủy thao tác xóa project

### 2.3.13 Send invitation to other users

<b>Use-case name</b>	Send invitation to other users
<b>Actor</b>	Workspace member
<b>Description</b>	Người dùng có thể gửi lời mời tham gia workspace cho người dùng trong hệ thống.
<b>Trigger</b>	Không
<b>Pre-conditions</b>	Người dùng đã đăng nhập vào hệ thống thành công
<b>Post-conditions</b>	Gửi lời mời người dùng trong hệ thống vào workspace thành công.
<b>Normal Flow</b>	<ol style="list-style-type: none"><li>1. Người dùng ở giao diện Default homepage, hệ thống hiển thị danh sách những workspace mà người dùng tham gia/sở hữu</li><li>2. Người dùng chọn icon "Menu" ở workspace mà người dùng tham gia với quyền hạn của viewer để hiển thị dropdown menu</li><li>3. Người dùng chọn nút "Share" để hiện modal chia sẻ workspace:<ol style="list-style-type: none"><li>(a) Những người hiện có trong workspace</li><li>(b) Quyền hạn của những người hiện có trong workspace</li></ol></li><li>4. Người dùng nhập email của người được mời để tìm kiếm người dùng trong hệ thống</li><li>5. Người dùng chọn quyền hạn của người dùng được mời vào workspace</li><li>6. Người dùng chọn "Send" để gửi lời mời tới người dùng</li></ol>
<b>Alternative flows</b>	<p>4a. Email của người được mời không tồn tại trên hệ thống</p> <p>4a.1. Hệ thống hiển thị không tìm thấy kết quả phù hợp</p> <p>5a. Người dùng gán quyền hạn cho người được mời vào workspace vượt quá quyền hạn hiện có trong workspace</p> <p>5a.1. Hệ thống thông báo lỗi yêu cầu người dùng gán quyền cho người được mời không vượt quá quyền hạn hiện có (Editor &gt; Sharer &gt; Viewer)</p> <p>5a.2. Tiếp tục ở bước 5</p>
<b>Exceptions</b>	5a. Người dùng chọn "Cancel" để hủy yêu cầu và đóng modal

## 2.4 Sơ đồ hoạt động

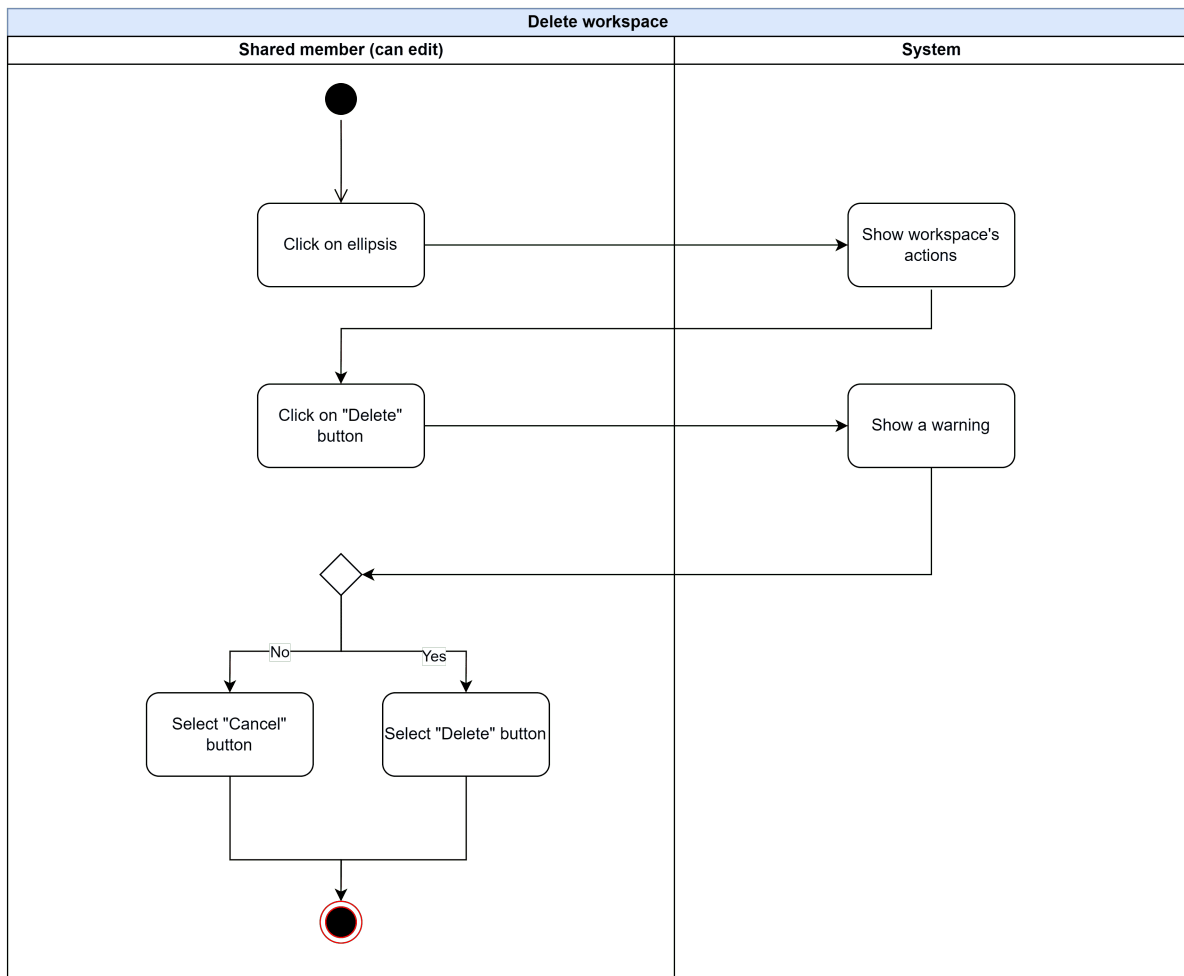
### 2.4.1 Tạo mới workspace



Hình 2: Tạo mới workspace

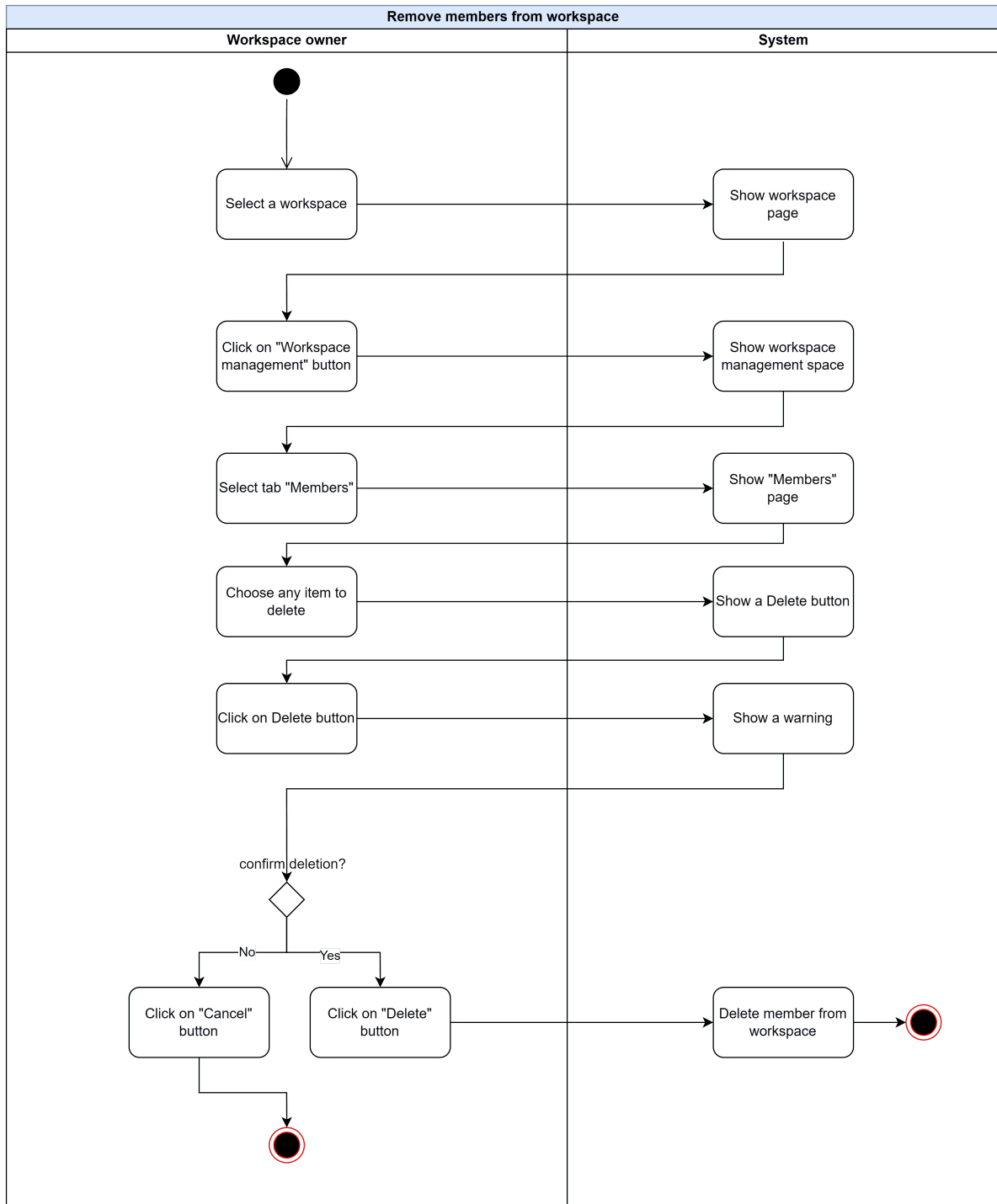


## 2.4.2 Xoá workspace



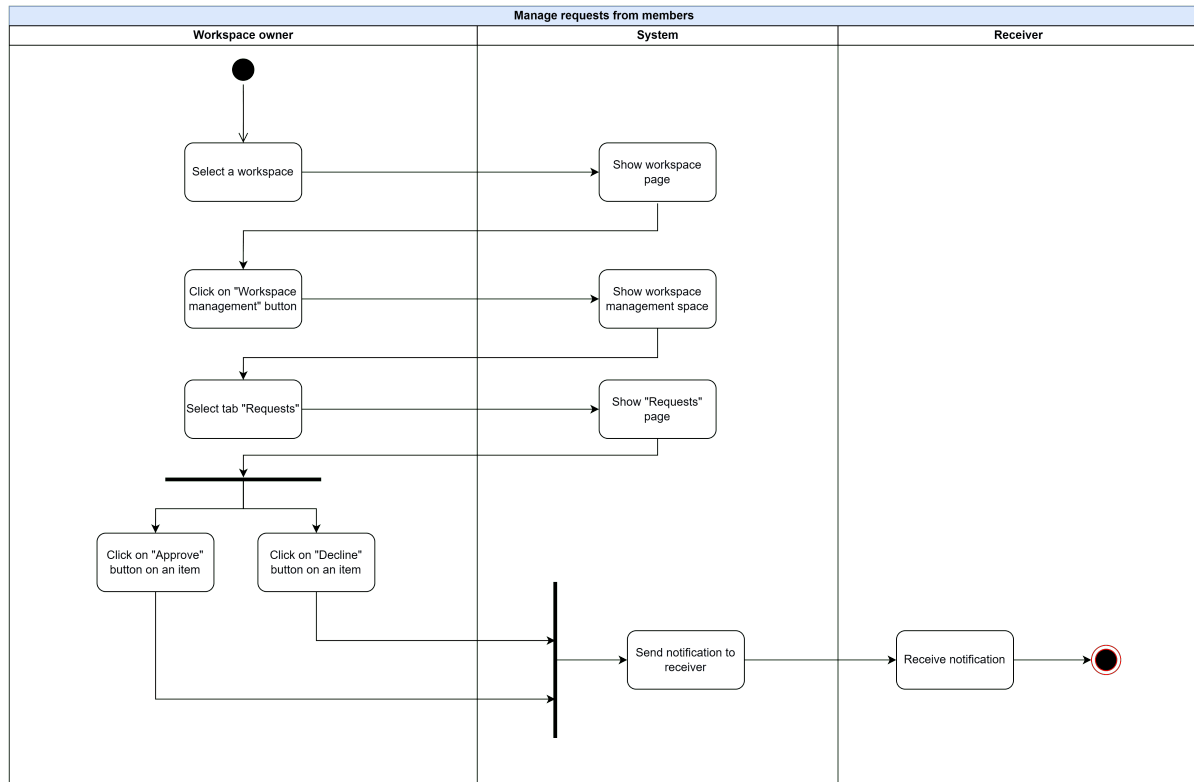
**Hình 3:** Xoá workspace

### 2.4.3 Xoá thành viên khỏi Workspace



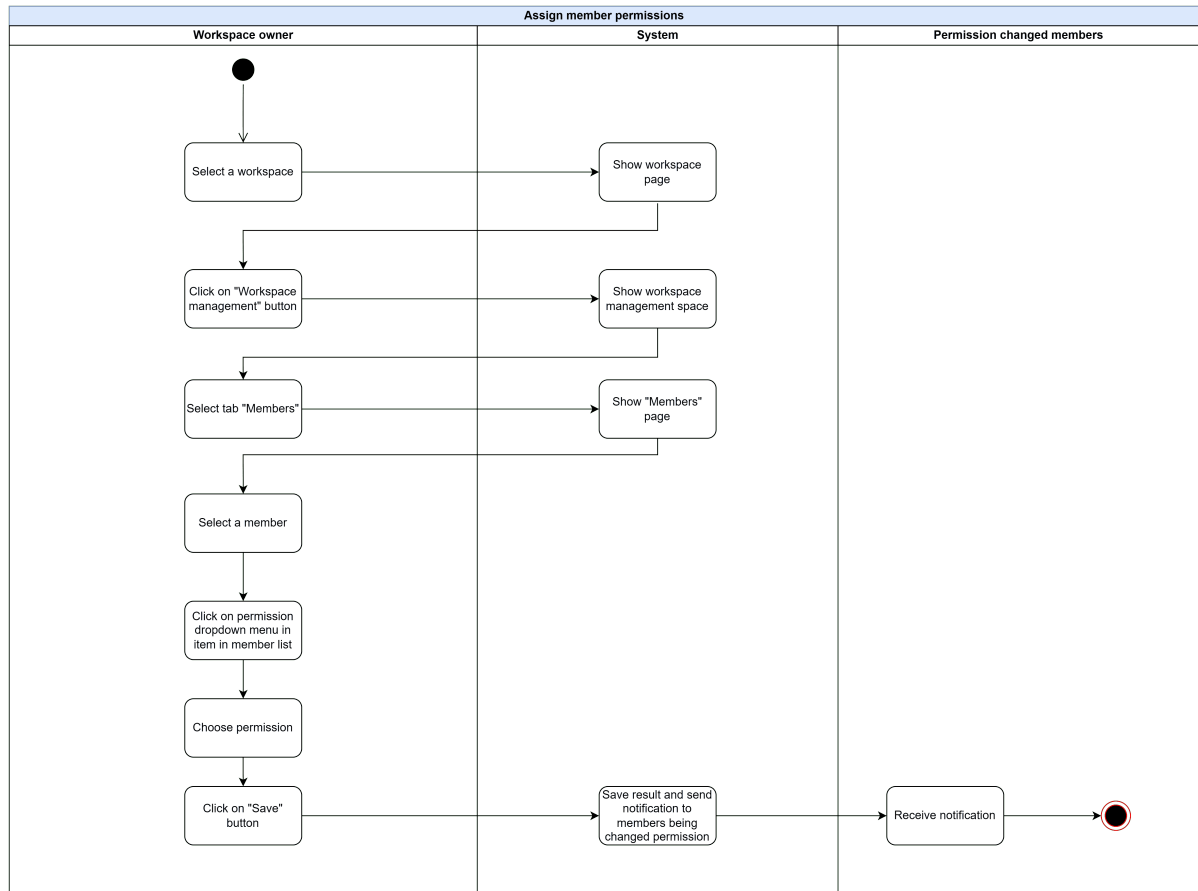
**Hình 4:** Xoá thành viên khỏi Workspace

## 2.4.4 Quản lý requests trong Workspace



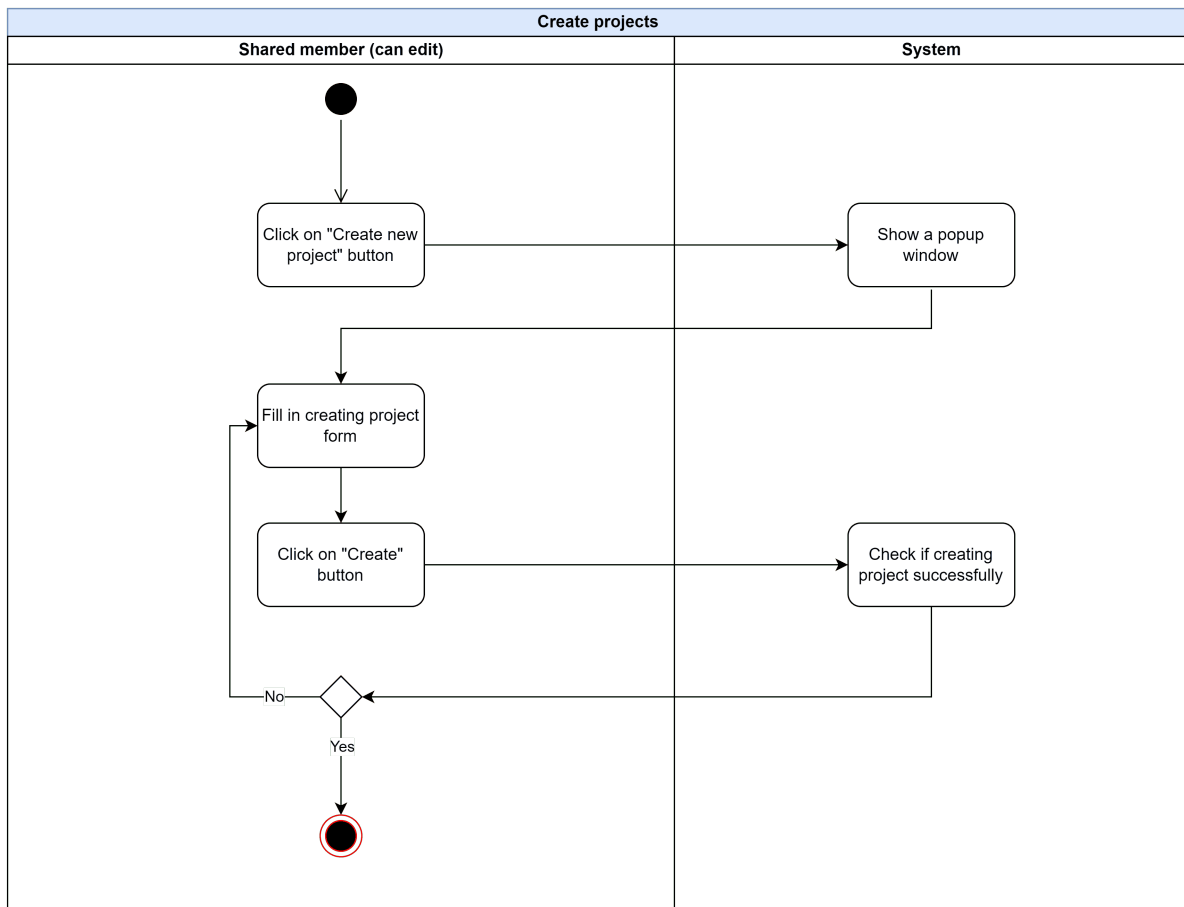
**Hình 5:** Quản lý requests trong Workspace

## 2.4.5 Thay đổi quyền của thành viên trong Workspace



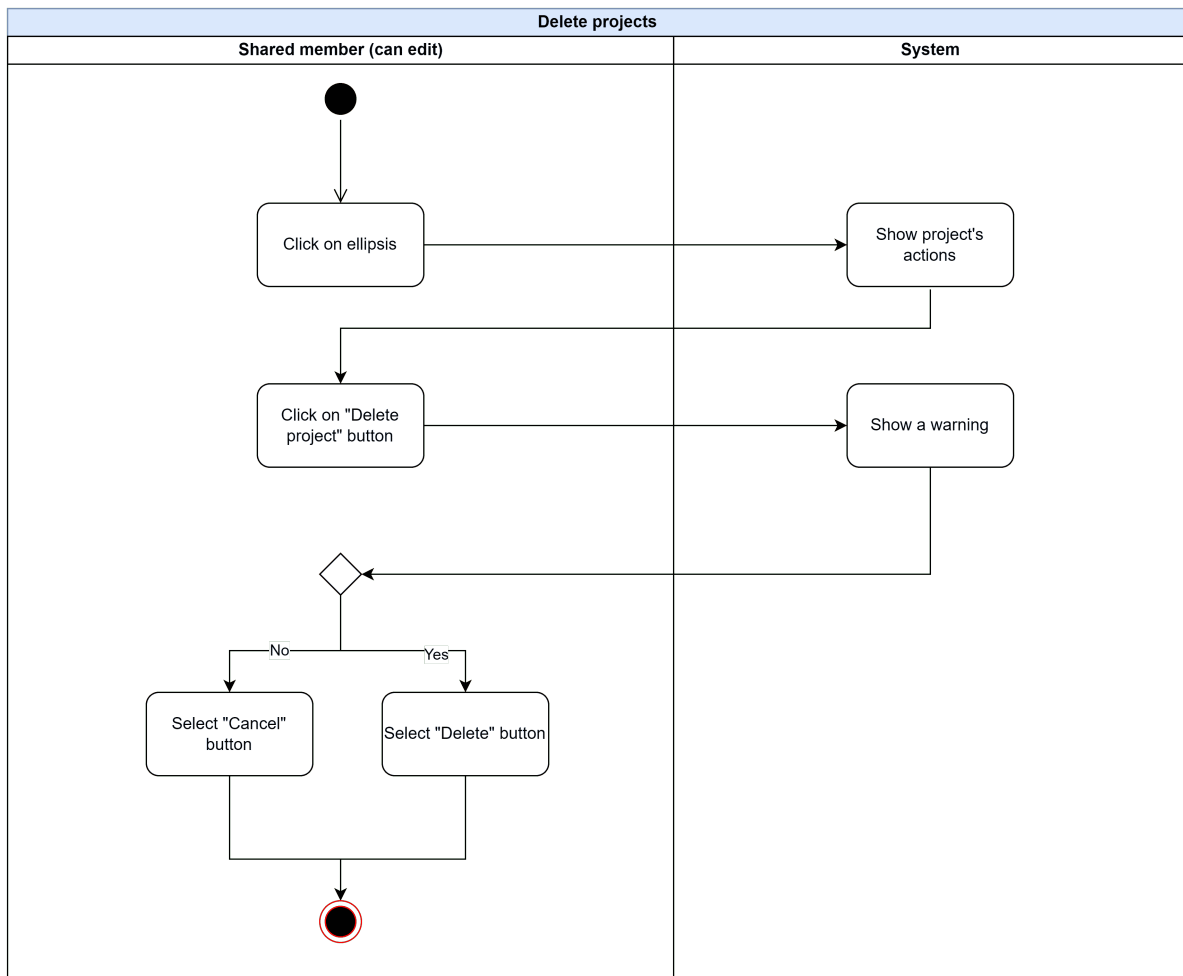
**Hình 6:** Thay đổi quyền của thành viên trong Workspace

## 2.4.6 Tạo mới project trong Workspace



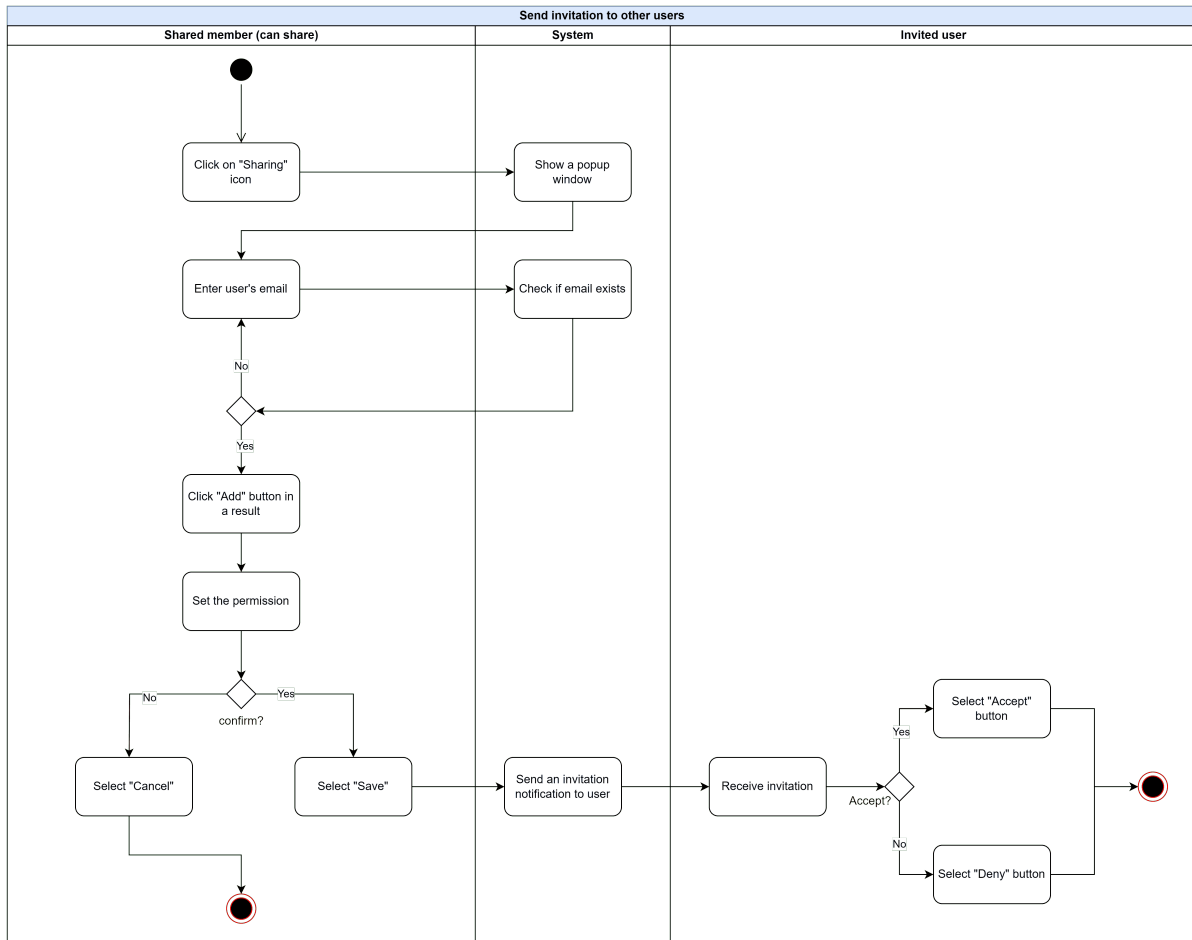
Hình 7: Tạo mới project trong Workspace

## 2.4.7 Xoá project trong Workspace



**Hình 8:** Xoá project trong Workspace

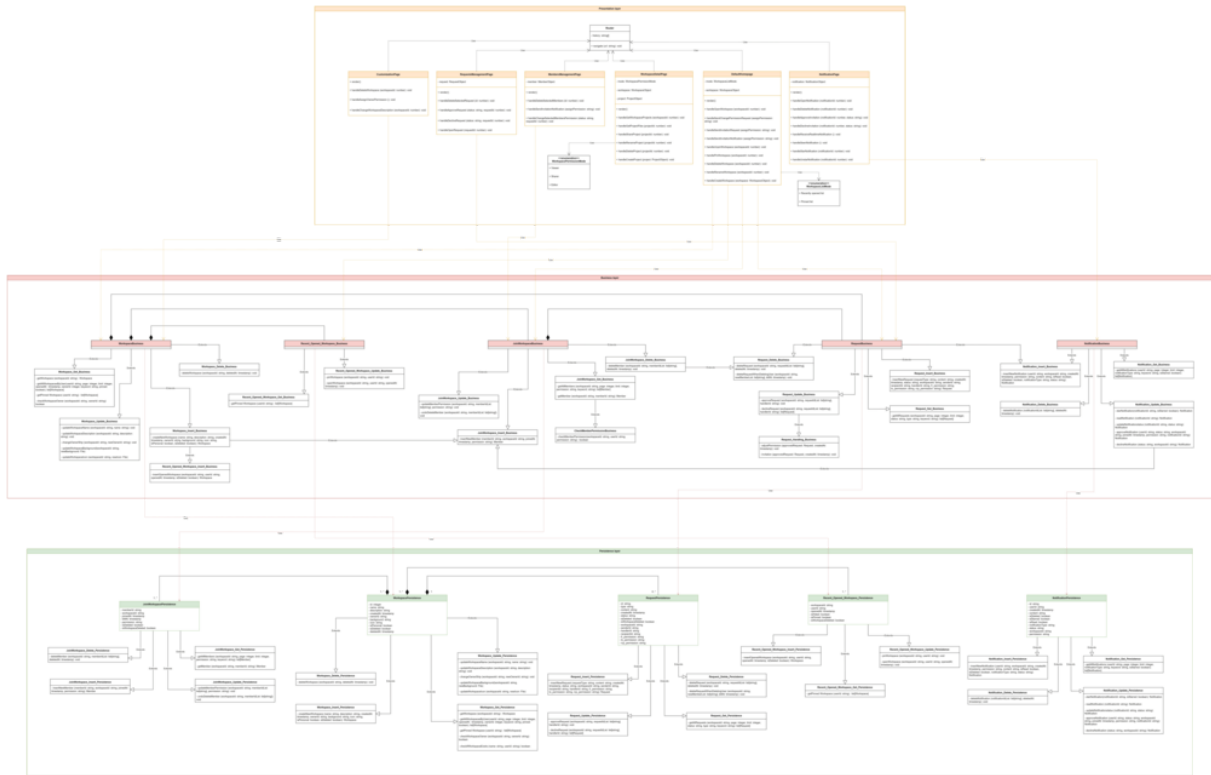
## 2.4.8 Gửi lời mời trực tiếp đến cho user



Hình 9: Gửi lời mời trực tiếp đến cho user

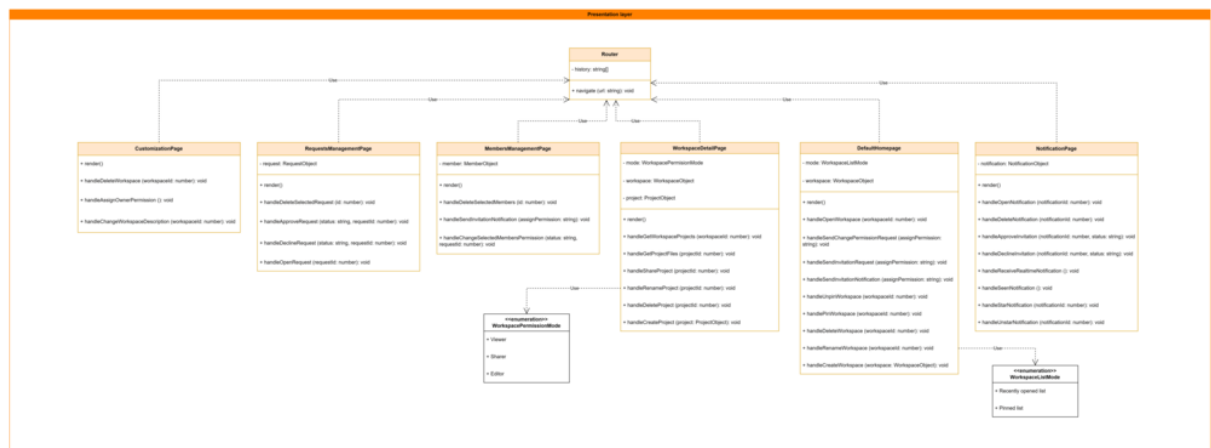
## 2.5 Sơ đồ lớp

### 2.5.1 Tổng quan



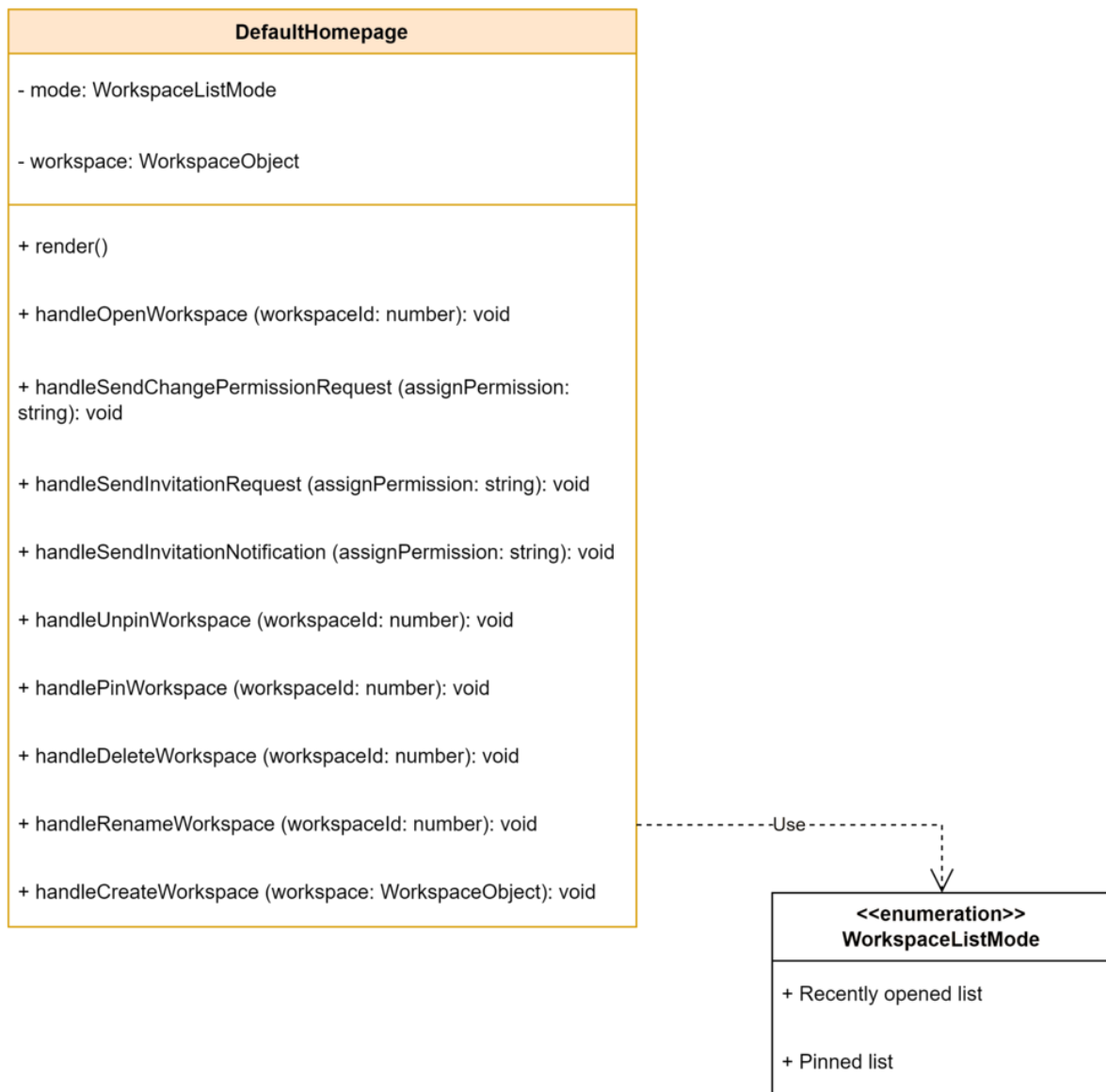
Hình 10: Sơ đồ lớp cho toàn bộ hệ thống

### 2.5.2 Presentation Layer

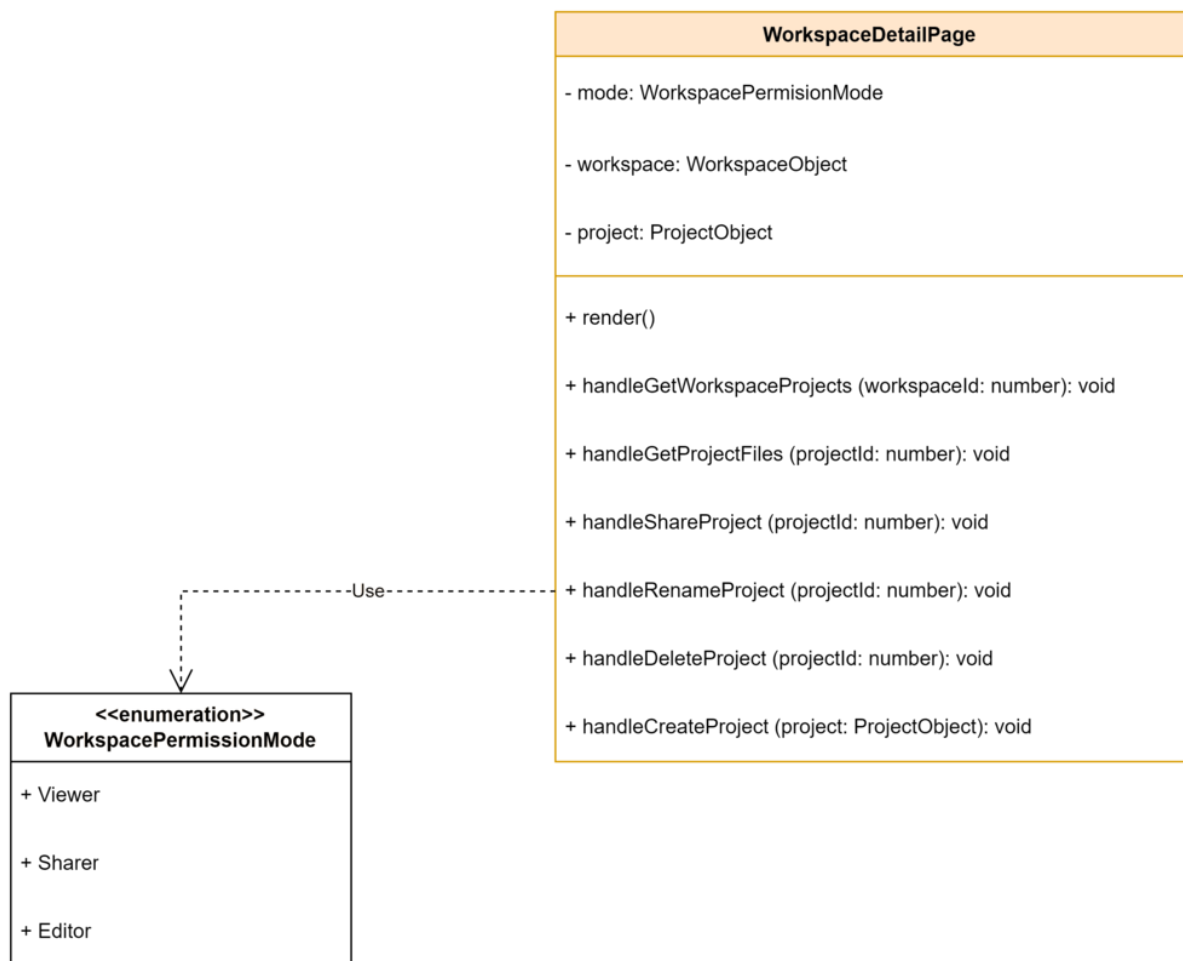


Hình 11: Presentation Layer





**Hình 12:** Default homepage trong Presentation layer



**Hình 13:** *Workspace detail page trong Presentation layer*

MembersManagementPage
- member: MemberObject
<div>+ render()</div> <div>+ handleDeleteSelectedMembers (id: number): void</div> <div>+ handleSendInvitationNotification (assignPermission: string): void</div> <div>+ handleChangeSelectedMembersPermission (status: string, requestId: number): void</div>

**Hình 14:** *Member management page trong Presentation layer*

NotificationPage
- notification: NotificationObject
+ render()  + handleOpenNotification (notificationId: number): void  + handleDeleteNotification (notificationId: number): void  + handleApproveInvitation (notificationId: number, status: string): void  + handleDeclineInvitation (notificationId: number, status: string): void  + handleReceiveRealtimeNotification (): void  + handleSeenNotification (): void  + handleStarNotification (notificationId: number): void  + handleUnstarNotification (notificationId: number): void

**Hình 15:** *Notification page trong Presentation layer*

## CustomizationPage

```
+ render()  
  
+ handleDeleteWorkspace (workspaceId: number): void  
  
+ handleAssignOwnerPermission (): void  
  
+ handleChangeWorkspaceDescription (workspaceId: number): void
```

**Hình 16:** *Customization page trong Presentation layer*

## RequestsManagementPage

- request: RequestObject

```
+ render()
```

```
+ handleDeleteSelectedRequest (id: number): void
```

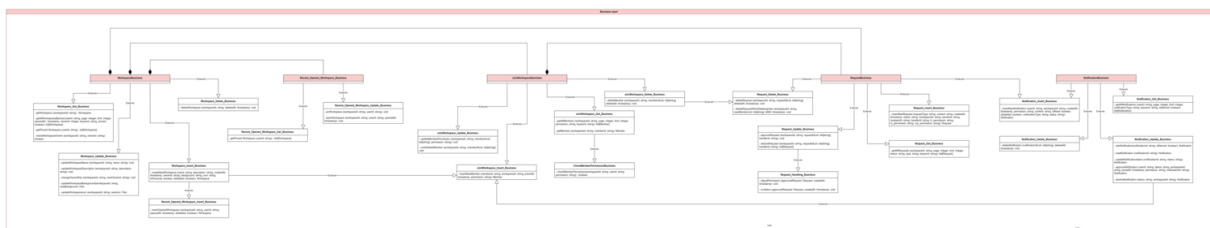
```
+ handleApproveRequest (status: string, requestId: number): void
```

```
+ handleDeclineRequest (status: string, requestId: number): void
```

```
+ handleOpenRequest (requestId: number): void
```

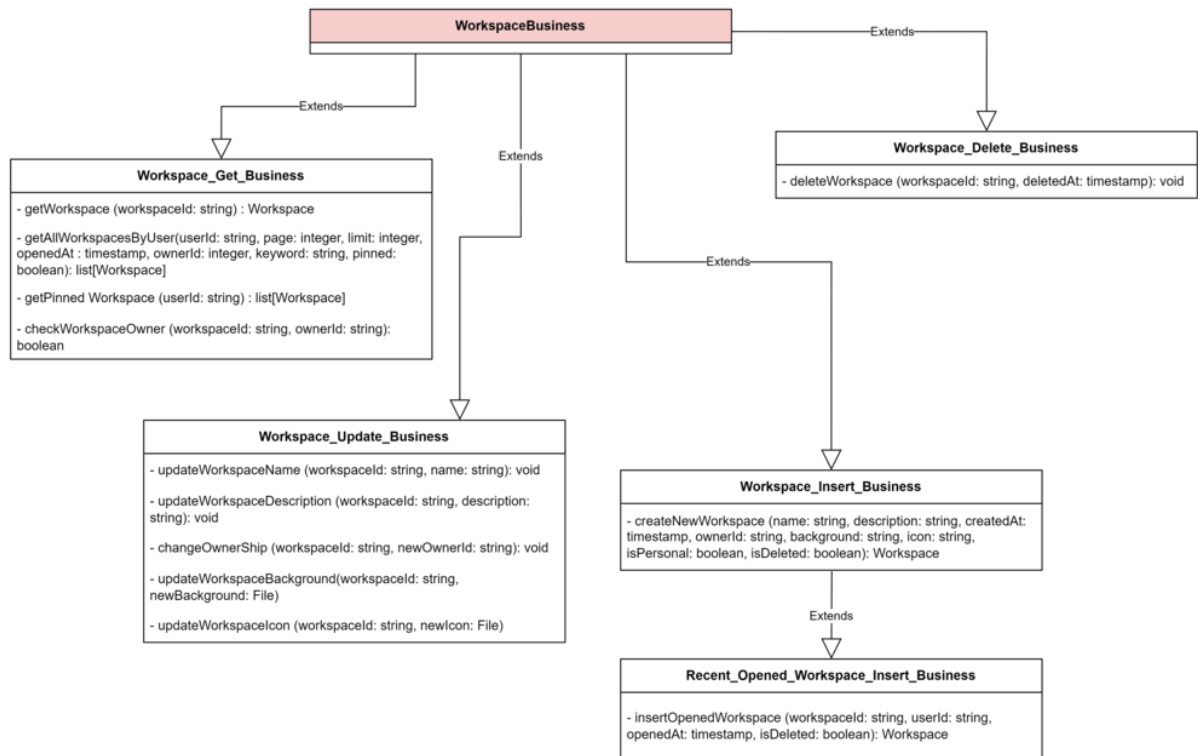
**Hình 17:** *Request management page trong Presentation layer*

### 2.5.3 Business Layer

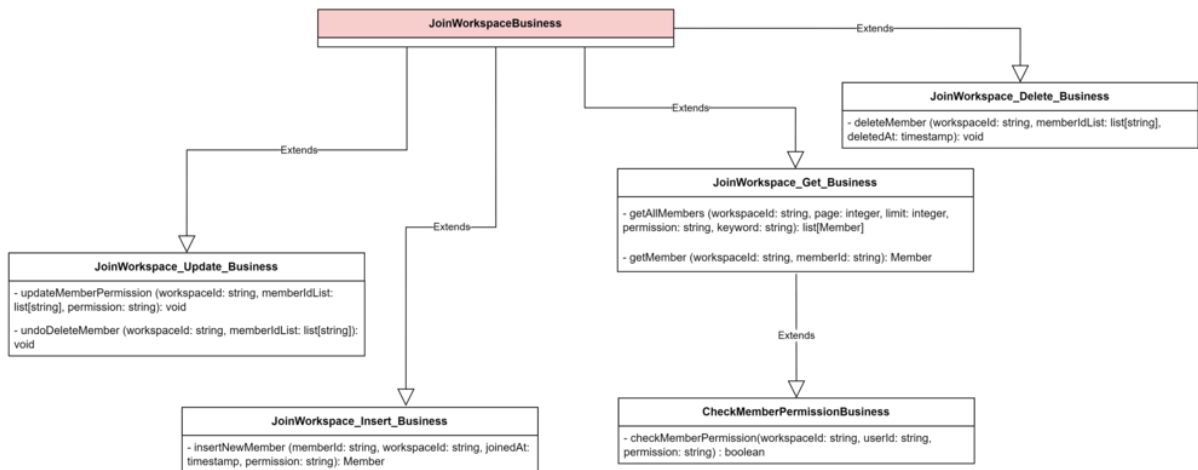


**Hình 18:** *Business Layer*

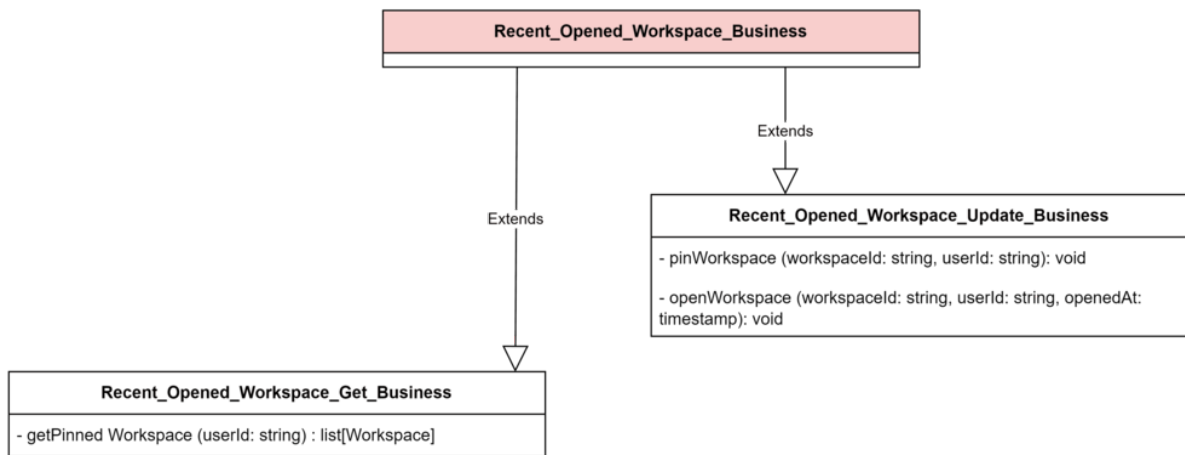
Nhiệm vụ của tầng Business là phân tích những yêu cầu, xử lý logic nghiệp vụ và gọi tới các đối tượng ở tầng Persistence để tương tác với cơ sở dữ liệu.



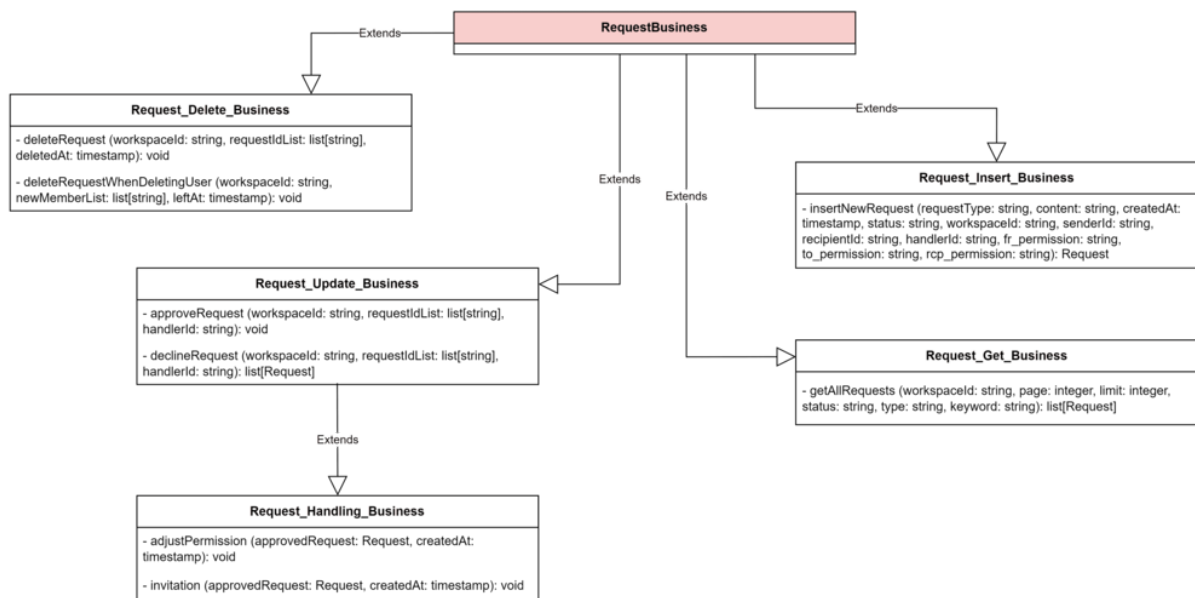
Hình 19: *Workspace Business*



Hình 20: *Join Workspace Business*

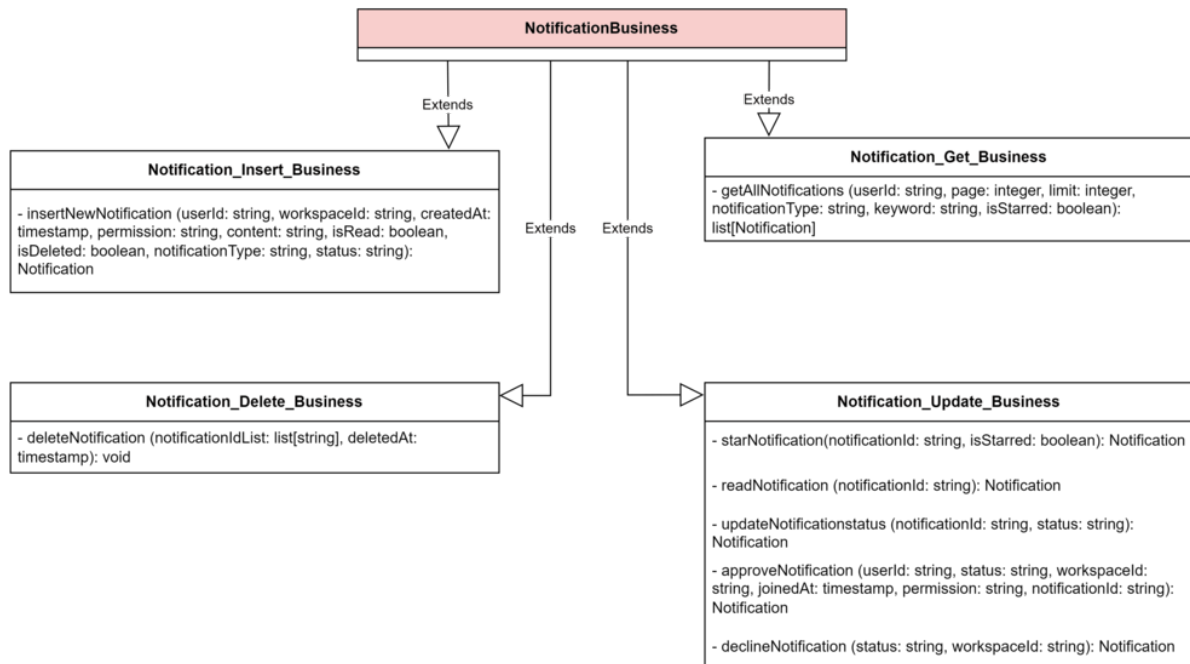


Hình 21: Recent Opened Workspace Business



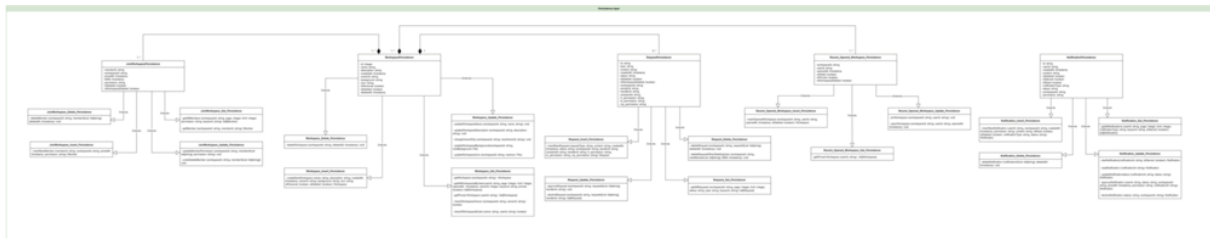
Hình 22: Request Business





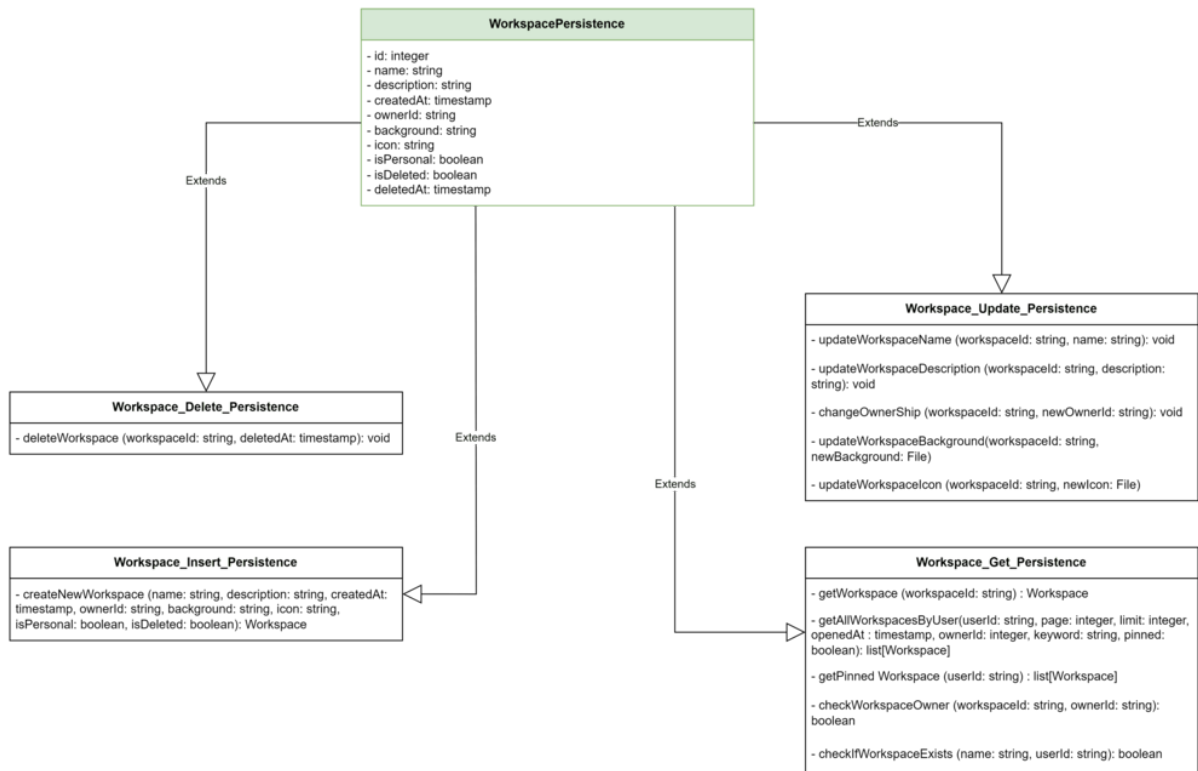
Hình 23: Notification Business

## 2.5.4 Persistence Layer

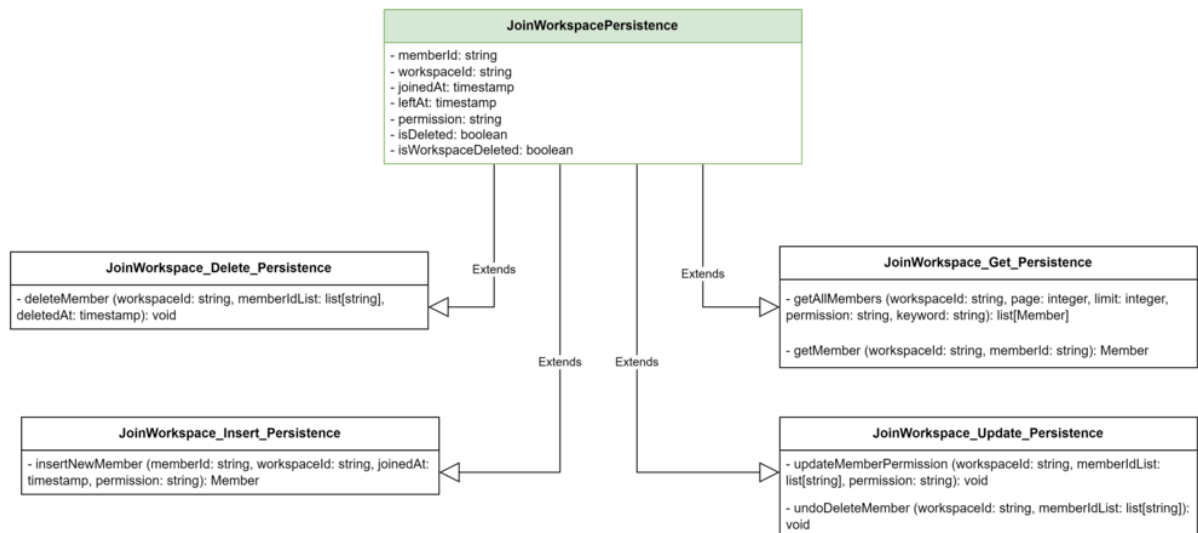


Hình 24: Persistence Layer

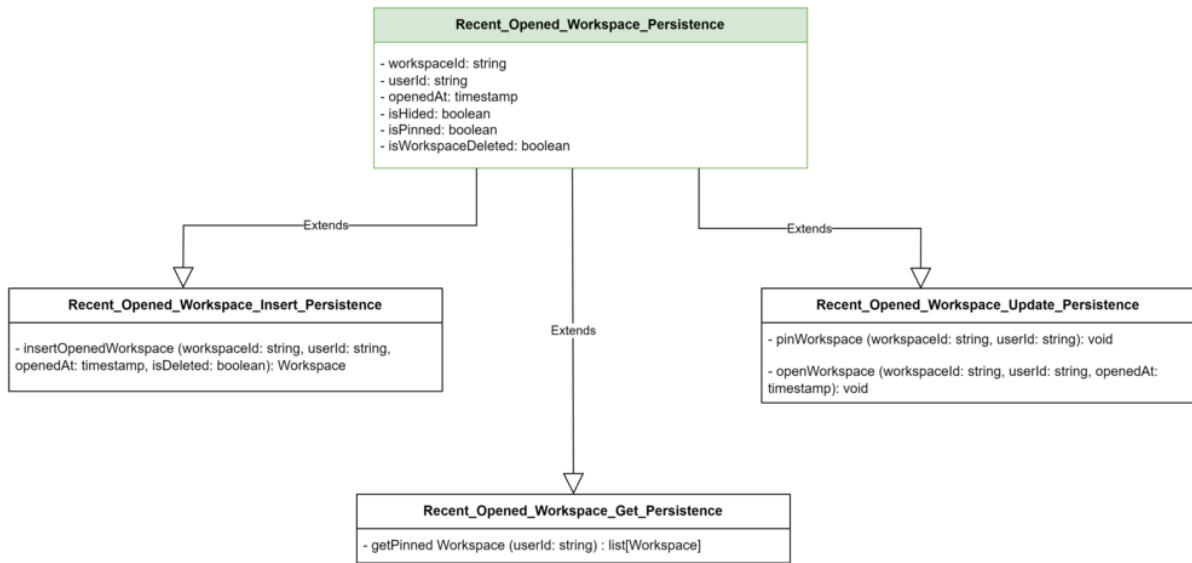
Tầng Persistence sẽ nhận dữ liệu, yêu cầu từ tầng Business để thực hiện các thao tác tương tác với cơ sở dữ liệu, sau đó trả về kết quả cho tầng Business.



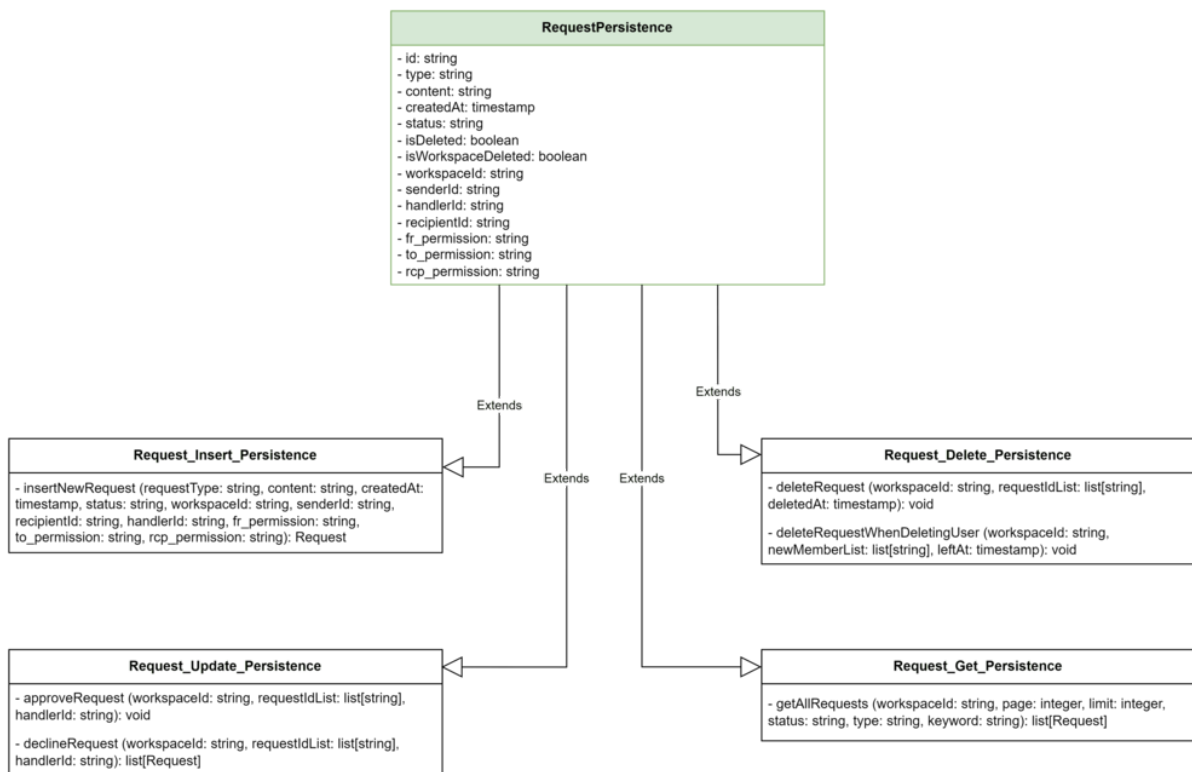
Hình 25: Workspace Persistence



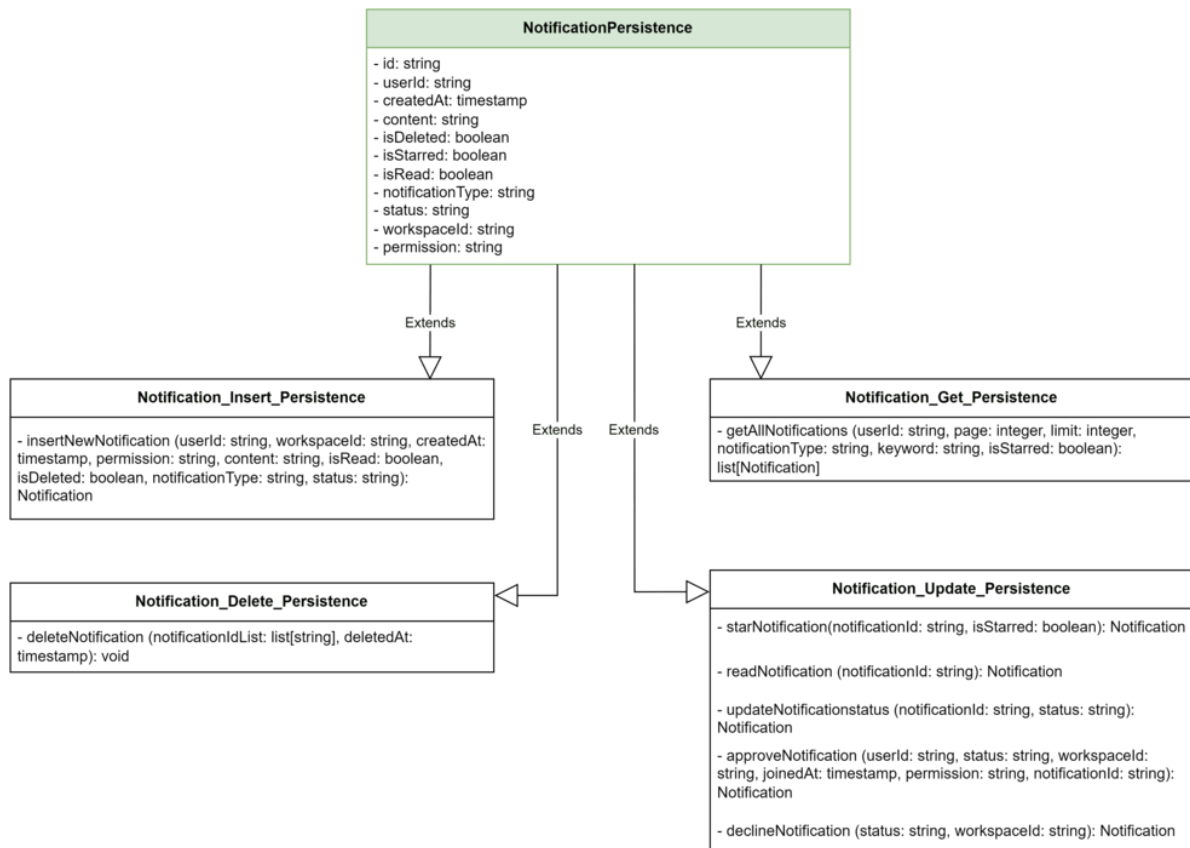
Hình 26: Join Workspace Persistence



Hình 27: Recent Opened Workspace Persistence



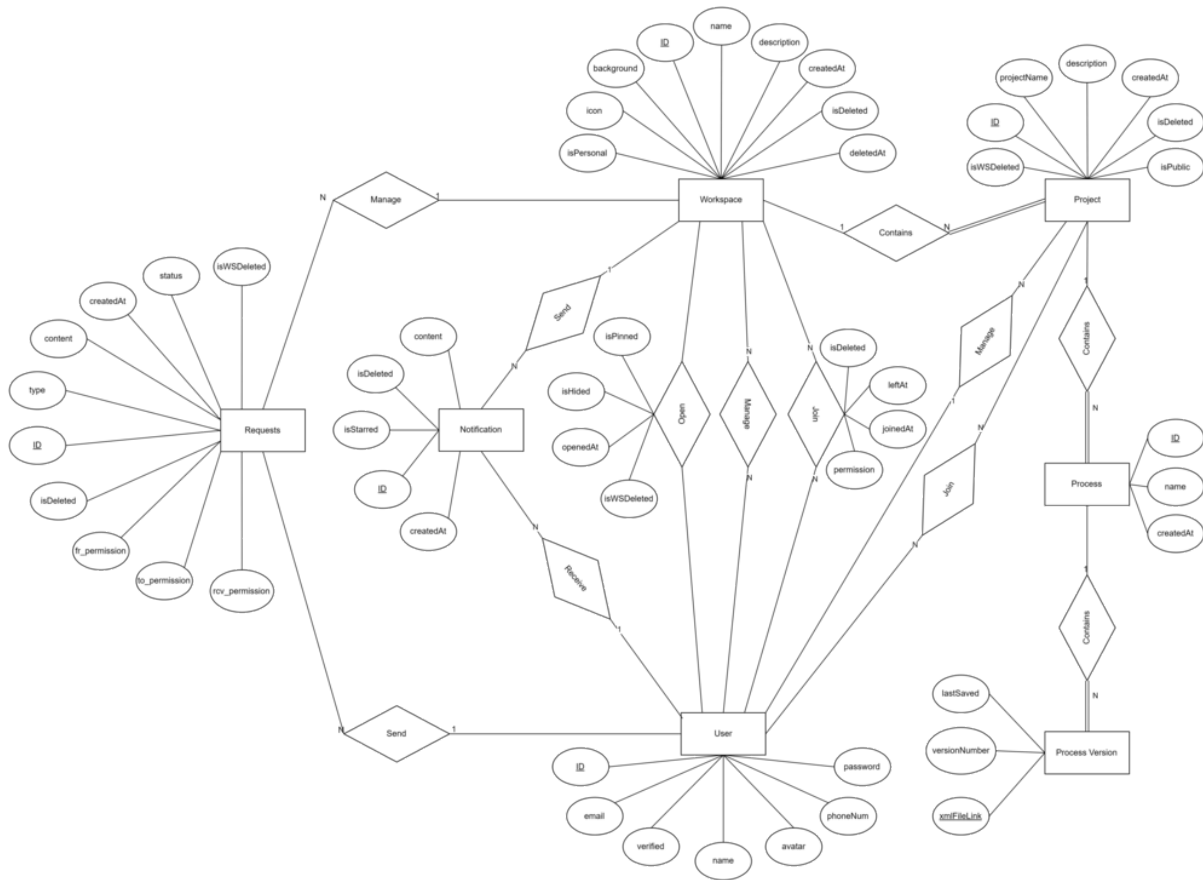
Hình 28: Request Persistence



Hình 29: Notification Persistence

## 2.6 Thiết kế cơ sở dữ liệu

### 2.6.1 Lược đồ ERD



### 2.6.2 Ánh xạ ERD và mô tả chi tiết thực thể

#### 2.6.2.1 User: Người dùng của hệ thống

Thuộc tính	Kiểu	Mô tả
<u>ID</u>	integer	ID của người dùng. Khoá chính.
email	varchar	Email của người dùng
password	varchar	Mật khẩu của người dùng
phoneNumber	varchar	Số điện thoại của người dùng
name	text	tên người dùng
avatar	text	Đường dẫn tới ảnh đại diện của người dùng
verified	boolean	True nếu tài khoản đã xác thực, False nếu tài khoản chưa xác thực

#### 2.6.2.2 Project: Dự án trong workspace

<u>ID</u>	email	verified	name	avatar	phoneNum	password
-----------	-------	----------	------	--------	----------	----------

Thuộc tính	Kiểu	Mô tả
<u>ID</u>	integer	ID của project. Khoá chính.
projectName	text	Tên của project
description	text	Mô tả của project
isDelete	boolean	Đánh dấu project này đã bị xoá hay chưa
createAt	timestamp	Thời gian project được tạo
ownerId	integer	ID của người sở hữu project. Khoá ngoại tham chiếu tới trường Id trong bảng User
workspaceId	integer	ID của workspace nơi mà project được tạo. Khoá ngoại tham chiếu tới trường Id trong bảng Workspace
deletedAt	timestamp	Thời gian project được xoá khỏi Workspace
isWorkspaceDeleted	boolean	Đánh dấu workspace đã bị xoá hay chưa

<u>ID</u>	projectName	description	createdAt	ownerId	isDeleted	isWorkspaceDeleted	workspaceId	deletedAt
-----------	-------------	-------------	-----------	---------	-----------	--------------------	-------------	-----------

### 2.6.2.3 Workspace: Không gian làm việc trong hệ thống

Thuộc tính	Kiểu	Mô tả
<u>ID</u>	integer	ID của workspace. Khoá chính.
name	text	Tên của workspace
description	text	Mô tả của workspace
isDeleted	boolean	Đánh dấu workspace đã bị xoá hay chưa
createAt	timestamp	Thời gian workspace được tạo
ownerId	integer	ID của người sở hữu workspace. Khoá ngoại tham chiếu tới trường Id trong bảng User
background	text	Đường link dẫn tới background của workspace
icon	text	Đường link dẫn tới icon của workspace
deletedAt	timestamp	Thời gian Workspace được xoá khỏi hệ thống
isPersonal	boolean	True nếu đây là workspace cá nhân. False nếu là workspace khác

<u>memberID</u>	<u>workspaceID</u>	joinedAt	permission	isDeleted	isWorkspaceDeleted	leftAt
-----------------	--------------------	----------	------------	-----------	--------------------	--------

#### 2.6.2.4 Join Workspace: Quan hệ tham gia giữa người dùng và workspace

Thuộc tính	Kiểu	Mô tả
<u>memberId</u>	integer	ID của member tham gia Workspace. Khoá ngoại tham chiếu tới trường Id trong User.
<u>workspaceId</u>	integer	ID của Workspace. Khoá ngoại tham chiếu tới trường Id trong Workspace.
joinedAt	timestamp	Thời gian người dùng tham gia Workspace
isDeleted	boolean	Đánh dấu người dùng đã rời khỏi Workspace hay chưa
permission	text	Quyền hạn của người dùng khi tham gia Workspace
leftAt	timestamp	Thời gian người dùng rời khỏi Workspace
isWorkspaceDeleted	boolean	Đánh dấu workspace đã bị xoá hay chưa

<u>memberID</u>	<u>workspaceID</u>	joinedAt	permission	isDeleted	isWorkspaceDeleted	leftAt
-----------------	--------------------	----------	------------	-----------	--------------------	--------

#### 2.6.2.5 Recent Opened Workspace: Quan hệ chỉnh sửa giữa người dùng và workspace

Thuộc tính	Kiểu	Mô tả
<u>userId</u>	integer	ID của member tham gia Workspace. Khoá ngoại tham chiếu tới trường Id trong User.
<u>workspaceId</u>	integer	ID của Workspace. Khoá ngoại tham chiếu tới trường Id trong Workspace.
openedAt	timestamp	Thời gian người dùng mở Workspace lần cuối.
isHided	boolean	Đánh dấu người dùng đã ẩn Workspace khỏi danh sách của mình hay chưa
isPinned	boolean	Đánh dấu người dùng đã pin Workspace vào danh sách ưu tiên hay chưa
isWorkspaceDeleted	boolean	Đánh dấu workspace đã bị xoá hay chưa

#### 2.6.2.6 Request: Các yêu cầu trong hệ thống

<u>workspaceID</u>	<u>userID</u>	openedAt	isHided	isPinned	isWorkspaceDeleted
--------------------	---------------	----------	---------	----------	--------------------

Thuộc tính	Kiểu	Mô tả
<u>ID</u>	integer	ID của request. Khoá chính
type	varchar	Kiểu request
createdAt	timestamp	Thời gian người dùng tạo request.
status	text	Trạng thái của request, đang được xử lý hay đã được chấp thuận, từ chối.
isDeleted	boolean	Đánh dấu request bị xoá hay chưa
isWorkspaceDeleted	boolean	Đánh dấu workspace đã bị xoá hay chưa
workspaceId	integer	ID của workspace chứa request đó. Khoá ngoại tham chiếu tới trường Id trong Workspace
senderId	integer	ID của người gửi request. Khoá ngoại tham chiếu tới trường Id trong User
recipientId	integer	ID của người nhận request đã xử lý. Khoá ngoại tham chiếu tới trường Id trong User
handlerId	integer	ID của người xử lý request. Khoá ngoại tham chiếu tới trường Id trong User
frPermission	text	permission hiện tại của người gửi.
toPermission	text	permission mong muốn của người gửi.
rcpPermission	text	permission của người sẽ nhận được lời mời tham gia workspace.
deletedAt	timestamp	Thời gian xoá request.

ID	type	content	createdAt	status	isDeleted	isWorkspaceDeleted	workspaceID	senderID	handlerID	recipientId	fr_permission	to_permission	rcp_permission	deletedAt
----	------	---------	-----------	--------	-----------	--------------------	-------------	----------	-----------	-------------	---------------	---------------	----------------	-----------

#### 2.6.2.7 Notification: Các thông báo của người dùng



Thuộc tính	Kiểu	Mô tả
<u>ID</u>	integer	ID của thông báo. Khoá chính
notificationType	varchar	Kiểu thông báo
createdAt	timestamp	Thời gian người dùng nhận được thông báo.
status	text	Trạng thái của thông báo, đã được chấp thuận hay đồng ý hoặc từ chối lời mời.
isDeleted	boolean	Đánh dấu thông báo bị xoá hay chưa
isStarred	boolean	Đánh dấu thông báo đã được đánh dấu ưu tiên hay chưa
workspaceId	integer	ID của workspace gửi thông báo đó. Khoá ngoại tham chiếu tới trường Id trong Workspace
isRead	boolean	Đánh dấu thông báo đã được đọc hay chưa
permission	text	permission của người nhận.
deletedAt	timestamp	Thời gian xoá request.

<u>ID</u>	<u>userId</u>	createdAt	content	isDeleted	isStarred	isRead	deletedAt	workspaceId	permission	notificationType	status
-----------	---------------	-----------	---------	-----------	-----------	--------	-----------	-------------	------------	------------------	--------

## 3 Hiện thực hệ thống

### 3.1 Công nghệ sử dụng

#### 3.1.1 ReactJS - Javascript

React là một thư viện Javascript mã nguồn mở, được phát triển và hỗ trợ bởi Meta (trước đây là Facebook), được ra mắt vào năm 2013 với mục đích xây dựng giao diện người dùng. React được sử dụng rộng rãi để xây dựng các trang web SPA (Single Page Application) và các ứng dụng trên nền tảng di động. React tổ chức các thành phần của trang web thành các component, cho phép chúng ta nhóm các logic xử lý của các phần nhỏ của giao diện lại với nhau, từ đó có thể tái sử dụng nhiều lần và mở rộng thêm chức năng cho mỗi thành phần đó. Mỗi component có thể tự quản lý trạng thái (state) của riêng mình, cũng như nhận vào các thuộc tính (props) được truyền từ component cha. React tối ưu việc render trang web bằng cách chỉ cho phép render khi phát hiện có sự thay đổi trong state và props, từ đó giảm thiểu số lần phải re - render. React còn nổi tiếng với việc sử dụng Virtual DOM để cải thiện hiệu năng, khi so sánh DOM gốc của trang web và Virtual DOM để tìm ra những sự khác nhau, và chỉ render lại những điểm khác nhau đó, thay vì re - render cả cây DOM, điều mà sẽ làm hiệu năng của trang web giảm đi đáng kể.

#### Một số ưu điểm của React

- Cập nhật hiệu quả. Các thuật toán của ReactJS đảm bảo rằng các cập nhật, thay đổi UI được thực thi hiệu quả. Chỉ những components bị ảnh hưởng bởi một thay đổi nào đó mới được re - render, giúp tối ưu hoá tổng quan hiệu năng của ứng dụng.
- Luồng dữ liệu một chiều. ReactJS chỉ cung cấp luồng dữ liệu đi theo một chiều, giúp cho lập trình viên dễ dàng hiểu cách dữ liệu hoạt động, thay đổi như thế nào trong cây phân cấp component, cũng như dự đoán trước trạng thái của component.
- Cộng đồng lớn. ReactJS là thư viện front-end Javascript nổi tiếng, do đó sự hỗ trợ từ cộng đồng là rất tốt khi ta có vấn đề trong việc lập trình. Ngoài ra, các thư viện, components của bên thứ ba, các hooks được phát triển và duy trì bởi hàng triệu lập trình viên trên toàn thế giới, giúp ta có thể tìm thấy thư viện mong muốn, phục vụ cho nhiều chức năng của hệ thống.

#### 3.1.2 Python

Tiếp nối với phần Backend của đề tài trước, ở đề tài này, nhóm tiếp tục sử dụng Python làm ngôn ngữ lập trình chính cho phần server của hệ thống, áp dụng framework Flask. Hiện nay có rất nhiều framework để hỗ trợ người dùng lập trình web với ngôn ngữ Python, và Flask là một trong những framework phổ biến đó.

Ở Flask có một số ưu điểm nổi bật để chúng tôi quyết định lựa chọn nó:

- Một micro web framework. Flask là một micro web framework được viết bằng Python, không yêu cầu tool hay thư viện cụ thể nào. “Micro” không có nghĩa là thiếu chức năng mà “micro” theo triết

lý thiết kế là cung cấp một lõi chức năng “súc tích” nhất cho ứng dụng web nhưng người dùng có thể mở rộng bất cứ lúc nào. Flask luôn hỗ trợ các thành phần tiện ích mở rộng cho ứng dụng như tích hợp cơ sở dữ liệu, xác thực biểu mẫu, xử lý upload, các công nghệ xác thực, template, email, RESTful... Người dùng có thể tập trung xây dựng web application ngay từ đầu trong một khoảng thời gian rất ngắn và có thể phát triển quy mô của ứng dụng tùy theo yêu cầu.

- Khả năng mở rộng. Chính nguyên lý và áp dụng microframework đã giúp cho Flask Python có thể dễ dàng mở rộng nếu có nhu cầu từ Web Application. Do phần core chạy độc lập và ít phụ thuộc, nên cho dù mở rộng ở mức quy mô thì web application sử dụng Flask Python vẫn đáp ứng được.
- Sự linh hoạt. Sự linh hoạt là tính năng cốt lõi và cũng là ưu điểm của Flask Python. Chính vì giữ cho core và các thành phần khác đơn giản nên ít bị phụ thuộc vào nhau. Sự đơn giản này giúp ta có thể chuyển hướng web application theo business owner dễ dàng hơn. Ngoài ra do ít bị phụ thuộc lẫn nhau nên một thành phần nào đó bị sập cũng khó mà kéo theo cả hệ thống bị sập.

### 3.1.3 Websocket và Socket.IO

Websocket là một công nghệ theo thời gian thực, cung cấp một giao tiếp hai chiều, song công toàn phần giữa máy khách và máy chủ trên một kết nối đơn socket, bền vững. Trong chế độ truyền song công toàn phần, việc giao tiếp giữa bên gửi và bên nhận có thể diễn ra đồng thời. Bên gửi và bên nhận có thể truyền và nhận tín hiệu cùng một lúc. Chế độ truyền song công toàn phần giống như con đường hai chiều, trong đó các phương tiện có thể lưu chuyển theo cả hai hướng cùng một lúc. Ví dụ, trong một cuộc trò chuyện qua điện thoại, hai người giao tiếp và cả hai có thể tự do nói và nghe cùng một lúc. Một kết nối Websocket sẽ bắt đầu với một HTTP request/response handshake. Nếu handshake khởi tạo thành công, máy chủ và máy khách đồng ý sử dụng kết nối TCP có sẵn đã được thiết lập cho một kết nối Websocket. Kết nối này sẽ tồn tại lâu nhất có thể (theo lý thuyết có thể tồn tại mãi mãi), cho phép máy chủ và máy khách có thể độc lập gửi dữ liệu cho nhau. Một số ưu điểm của Websocket:

- Trước khi có Websocket, các kỹ thuật HTTP như AJAX long polling và Comet là những chuẩn để xây dựng các ứng dụng thời gian thực. Tuy nhiên, ở Websocet đã lược bỏ đi việc thiết lập kết nối mỗi khi có request, và giảm kích thước của mỗi message vì không còn HTTP headers. Điều này giúp tiết kiệm băng thông, cải thiện độ trễ.
- Độ linh hoạt đã ăn sâu vào thiết kế của công nghệ Websocket, cho phép việc hiện thực các giao thức ở tầng ứng dụng và các mở rộng cho những chức năng khác (ví dụ như pub/sub messaging).
- Là một công nghệ hướng sự kiện (event - driven), Websocket cho phép dữ liệu được truyền đi mà không cần máy khách phải gửi request. Cơ chế này sẽ cực kỳ hữu ích trong viễn cảnh máy khách cần phản ứng nhanh chóng với một sự kiện nào đó (đặc biệt là những sự kiện không thể dự đoán trước).

Dẫu vậy, Websocket vẫn có một số những nhược điểm như sau:

- Không như HTTP, Websocket là có trạng thái (stateful). Trong một số trường hợp sẽ khá khó khăn khi phải xử lý điều này, đặc biệt là lúc mở rộng hệ thống, bởi vì nó yêu cầu phía server phải theo dõi mỗi kết nối Websocket đơn lẻ và duy trì các thông tin trạng thái.
- Websocket không tự động khôi phục khi các kết nối bị huỷ - và để tự động khôi phục ta cần phải tự hiện thực lấy. Đây cũng là lý do vì sao có nhiều thư viện hỗ trợ Websocket phía máy khách.
- Một số môi trường (ví dụ như môi trường trong doanh nghiệp với các proxy servers) sẽ chặn các kết nối Websocket.

Chính vì vậy, chúng tôi sử dụng thêm một thư viện để hỗ trợ trong việc trao đổi dữ liệu qua giao thức Websocket. Socket.IO là một thư viện theo thời gian thực, giúp giảm thiểu độ trễ và giao tiếp hai chiều giữa máy khách và máy chủ. Socket.IO được xây dựng trên nền tảng giao thức Websocket, và cung cấp thêm một số tính năng khác như tự động kết nối lại, hỗ trợ broadcast,... Một số ưu điểm của Socket.IO:

- Socket.IO hỗ trợ multiplexing thông qua namespaces. Việc tận dụng namespaces giúp ta có thể tối giản hoá số lượng kết nối TCP đã sử dụng, và lưu các cổng socket trên máy chủ.
- Socket.IO cho phép phía máy chủ có thể linh hoạt broadcast các sự kiện đến tất cả máy khách đã kết nối. Ta cũng có thể broadcast các sự kiện tới một tập nhỏ các máy khách thông qua tính năng room.
- Socket.IO hỗ trợ HTTP long polling như một phương án dự phòng, sẽ hữu ích trong các môi trường không hỗ trợ Websocket.
- Socket.IO cung cấp cơ chế Ping/Pong, cho phép phát hiện một kết nối có đang hoạt động hay không. Ngoài ra, nếu một máy khách bị ngắt kết nối, nó sẽ tự động kết nối lại.

Tuy nhiên, thư viện này cũng bộc lộ một số nhược điểm như sau:

- Socket.IO có giới hạn trong các tính năng bảo mật. Chẳng hạn như, nó không cung cấp mã hoá đầu cuối, hay cơ chế nào để tạo ra và làm mới các tokens phục vụ cho việc xác thực người dùng.
- Không tương thích với những hiện thực Websocket khác. Ta không thể sử dụng một máy khách Websocket cơ bản với một máy chủ Socket.IO.
- Socket.IO được thiết kế để làm việc trong một vùng đơn, hơn là kiểu kiến trúc đa vùng. Điều này có thể làm gia tăng độ trễ nếu các người dùng ở các vùng khác nhau.

### 3.1.4 PostgreSQL

Để thiết kế database cho mục đích lưu trữ dữ liệu, nhóm quyết định chọn cơ sở dữ liệu quan hệ (relational database) để hiện thực, và hệ quản trị cơ sở dữ liệu được chọn là PostgreSQL. Những lợi ích của việc sử dụng hệ cơ sở dữ liệu có quan hệ như sau:

- Bảo đảm tính toàn vẹn dữ liệu. Cơ sở dữ liệu quan hệ bắt buộc cần phải bảo đảm tính toàn vẹn dữ liệu, bằng việc sử dụng các ràng buộc như khoá chính, khoá ngoại, các ràng buộc duy nhất. Những ràng buộc này giúp cho dữ liệu được chính xác, nhất quán và tuân theo các quy định đã được định nghĩa từ trước.
- Nhờ việc sử dụng kỹ thuật chuẩn hoá dữ liệu, mà cơ sở dữ liệu quan hệ có thể tái tổ chức lại dữ liệu, lược bỏ những dư thừa hay các vấn đề liên quan đến nhất quán dữ liệu khi tiến hành chỉnh sửa, thêm, xoá dữ liệu.
- Hỗ trợ các công cụ cũng như tính năng nâng cao giúp người dùng dễ dàng xử lý trên dữ liệu một cách hiệu quả như indexing, transaction, cơ chế query optimization.
- Hệ quản trị cơ sở dữ liệu quan hệ cho phép lưu trữ và phục hồi nhờ việc cung cấp các cơ chế có sẵn cho việc phục hồi. Điều này đảm bảo dữ liệu có thể được lưu trữ một cách an toàn, giải quyết các vấn đề liên quan tới mất mát dữ liệu.

So sánh với các loại cơ sở dữ liệu không quan hệ phổ biến hiện nay, ưu điểm lớn nhất của việc sử dụng cơ sở dữ liệu quan hệ so với cơ sở dữ liệu không quan hệ trong phạm vi đề tài đó là việc sử dụng cấu trúc có quan hệ giúp quản lý dữ liệu tốt hơn, tính nhất quán dữ liệu được đảm bảo hơn. Ngoài ra, do đề tài này được phát triển dựa trên đề tài gốc vốn sử dụng cơ sở dữ liệu quan hệ, để đảm bảo tính thống nhất về mặt chức năng của hệ thống, nhóm nhận thấy tiếp tục loại hình cơ sở dữ liệu này là một điều đúng đắn.

### **3.2 Hiện thực giao diện người dùng**

### 3.3 Kết luận

#### 3.3.1 Kết quả đạt được

Đối với đề tài này, ở giai đoạn đồ án chuyên ngành, chúng tôi đã đạt được một số kết quả cơ bản như sau:

- Hiện thực không gian quản lý các dự án và các quy trình nghiệp vụ, cho phép người dùng chia sẻ không gian để mở rộng và cộng tác với nhiều thành viên khác trong hệ thống.
- Xây dựng lý thuyết đánh giá chất lượng và độ linh hoạt của quy trình nghiệp vụ.

#### 3.3.2 Những điểm hạn chế

Bên cạnh những kết quả đạt được trình bày ở trên, hệ thống của chúng tôi vẫn còn tồn đọng một số hạn chế:

–

#### 3.3.3 Hướng phát triển cho giai đoạn luận văn

- Hoàn thiện tính năng thiết kế bảng khảo sát, chia sẻ và thu nhận kết quả khảo sát.
- Hoàn thiện chức năng đánh giá chất lượng và độ linh hoạt của quy trình nghiệp vụ.
- Hiện thực tính năng đề xuất bản thiết kế quy trình, giúp người dùng chọn lựa quy trình nghiệp vụ mẫu có sẵn phù hợp với nhu cầu của mình.
- Mở rộng hệ thống, hướng đến cho phép người dùng đóng góp, chia sẻ thiết kế, đánh giá của quy trình nghiệp vụ cho những người dùng khác trong hệ thống.



## 4 Tham khảo

1.