

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



BÀI TẬP LỚN MÔN HỌC
CÔNG NGHỆ PHẦN MỀM (CO3001)

DỊCH VỤ
IN ẤN THÔNG MINH

GVHD: Ths. Lê Đình Thuận

SV thực hiện: Nguyễn Phạm Thiên Phúc – 2114445
Võ Tấn Hưng – 2113623
Nguyễn Đức An – 2112737
Trần Minh Hiếu – 2113363
Trương Thuận Hưng – 2113619



Danh sách thành viên & Tiến độ

Tuần 1 & 2:

Student ID	Full name	Assigned Task	% Work
2114445	Nguyễn Phạm Thiên Phúc	1.2. Động cơ nghiên cứu	100%
2113623	Võ Tấn Hưng	1.3. Mục tiêu	100%
2112737	Nguyễn Đức An	1.4. Yêu cầu chức năng	100%
2113363	Trần Minh Hiếu	1.5. Yêu cầu phi chức năng	100%
2113619	Trương Thuận Hưng	1.1. Mô tả dự án	100%

Tuần 3 & 4:

Student ID	Full name	Assigned Task	% Work
2114445	Nguyễn Phạm Thiên Phúc	Review tổng các UC Diagram, UC Scenario 3.1 UC: Whole System	100%
2113623	Võ Tấn Hưng	3.2 UC Diagram & UC Scenario: Printing	100%
2112737	Nguyễn Đức An	3.3 UC Diagram& UC Scenario: Modify Printing System Configuration	100%
2113363	Trần Minh Hiếu	3.4 UC Diagram & UC Scenario: Manage Printers	100%
2113619	Trương Thuận Hưng	3.5 UC Diagram& UC Scenario: Manage Local Printers	100%

Tuần 5-7:

Student ID	Full name	Assigned Task	% Work
2114445	Nguyễn Phạm Thiên Phúc	Thiết kế Class diagram Biểu đồ tuần tự: 5.4	100%
2113623	Võ Tấn Hưng	Thiết kế Class diagram.	100%
2112737	Nguyễn Đức An	Biểu đồ hoạt động: 4.2, 4.4 Biểu đồ tuần tự: 5.2	100%
2113363	Trần Minh Hiếu	Biểu đồ hoạt động: 4.3 Biểu đồ tuần tự: 5.3. MVP1	100%
2113619	Trương Thuận Hưng	Biểu đồ hoạt động: 4.1 Biểu đồ tuần tự: 5.1, 5.4	100%



Tuần 8-9:

Student ID	Full name	Assigned Task	% Work
2114445	Nguyễn Phạm Thiên Phúc	Thiết kế Deployment diagram	100%
2113623	Võ Tấn Hưng	Thiết kế Component diagram, Box and line diagram	100%
2112737	Nguyễn Đức An	Thiết kế Deployment diagram	100%
2113363	Trần Minh Hiếu	Thiết kế Component diagram	100%
2113619	Trương Thuận Hưng	Thiết kế Box and line diagram	100%

Tuần 10:

Student ID	Full name	Assigned Task	% Work
2114445	Nguyễn Phạm Thiên Phúc	Tạo kho lưu trữ (github), thêm các tài liệu, văn bản, thư mục cho giai đoạn 1	100%
2113623	Võ Tấn Hưng	Xác định nhiệm vụ, chiến lược kiểm thử	100%
2112737	Nguyễn Đức An	Tiến hành kiểm thử	100%
2113363	Trần Minh Hiếu	Viết báo cáo kiểm thử	100%
2113619	Trương Thuận Hưng	Tiến hành kiểm thử	100%

Tuần 11-12:

Student ID	Full name	Assigned Task	% Work
2114445	Nguyễn Phạm Thiên Phúc	Hiện thực phần Backend	100%
2113623	Võ Tấn Hưng	Hiện thực phần Backend	100%
2112737	Nguyễn Đức An	Hiện thực phần Frontend Tổng hợp và chỉnh sửa báo cáo	100%
2113363	Trần Minh Hiếu	Hiện thực phần Frontend Bổ sung tính năng Trò chuyện	100%
2113619	Trương Thuận Hưng	Hiện thực phần Frontend Bổ sung tính năng Thông báo	100%



Mục lục

1 Thu thập yêu cầu (Requirement elicitation)	5
1.1 Tổng quan	5
1.1.1 Mô tả dự án	5
1.1.1.a Bối cảnh áp dụng (Domain Context)	5
1.1.1.b Các bên liên quan (Stakeholders) - Nhu cầu và lợi ích	5
1.1.2 Động cơ nghiên cứu	6
1.1.3 Mục tiêu	7
1.2 Yêu cầu chức năng và phi chức năng	7
1.2.1 Cơ sở lý thuyết	7
1.2.2 Yêu cầu chức năng	8
1.2.2.a Góc độ người dùng (sinh viên)	8
1.2.2.b Góc độ người quản lý (SPSO)	8
1.2.2.c Góc độ nhân viên trực quầy (Staff)	9
1.2.3 Yêu cầu phi chức năng	9
1.2.3.a Yêu cầu phi chức năng cho toàn hệ thống	10
1.2.3.b Yêu cầu phi chức năng cho các tính năng	10
1.3 Biểu đồ use-case và kịch bản use-case	11
1.3.1 Cơ sở lý thuyết	11
1.3.2 Use-case: Whole system	12
1.3.3 Use-case: Printing	13
1.3.4 Use-case: Modify system configuration	15
1.3.5 Use-case: Manage printer	16
1.3.6 Use-case: Manage local printer	19
2 Mô hình hóa hệ thống (System modeling)	21
2.1 Biểu đồ hoạt động (Activity Diagram)	21
2.1.1 Cơ sở lý thuyết	21
2.1.2 Chức năng Printing	21
2.1.3 Chức năng Modify Printing System Configuration	22
2.1.4 Chức năng Manage Printer	23
2.1.5 Chức năng Manage Local Printer	26
2.2 Biểu đồ tuần tự (Sequence Diagram)	27
2.2.1 Cơ sở lý thuyết	27
2.2.2 Chức năng Printing	27
2.2.3 Chức năng Modify Printing System Configuration	29
2.2.4 Chức năng Manage Printer	30
2.2.5 Chức năng Manage Local Printer	37
2.3 Biểu đồ lớp (Class Diagram)	42
2.3.1 Cơ sở lý thuyết	42
2.3.2 Mô tả chi tiết	42
2.3.3 Hình vẽ	43
2.4 MVP 1 với giao diện người dùng	45
2.4.1 Sinh viên	45
2.4.2 SPSO	46
2.4.3 Staff	48
3 Thiết kế kiến trúc (Architecture design)	50
3.1 Biểu đồ Deployment	51
3.2 Biểu đồ Box - Line	53
3.3 Biểu đồ Component	57



4 Thực hiện - Giai đoạn 1 (Implementation - Sprint 1)	61
4.1 Thiết lập kho lưu trữ trực tuyến để kiểm soát phiên bản	61
4.2 Thêm tài liệu và thư mục cho phần "Thu thập yêu cầu", "Mô hình hóa hệ thống" và "Thiết kế kiến trúc"	61
4.3 Kiểm tra khả năng sử dụng với giao diện đã phát triển ở MVP 1	63
4.3.1 Xác định nhiệm vụ	63
4.3.1.a Giao diện Student	63
4.3.1.b Giao diện SPSO	65
4.3.2 Xác định chiến lược test	66
4.3.2.a Định lượng	66
4.3.2.b Từ xa	67
4.4 Phản hồi từ các tester	67
4.4.1 Giao diện Student	67
4.4.1.a Cấu hình và in ấn	67
4.4.1.b Thanh toán giấy in	69
4.4.1.c Dánh giá của tester về giao diện của Sinh viên nói chung	71
4.4.2 Giao diện SPSO	72
4.4.2.a Cấu hình hệ thống	72
4.4.2.b Xóa máy in	72
4.4.2.c Tùy chỉnh máy in	73
4.4.2.d Dánh giá của tester về giao diện của SPSO nói chung	73
5 Thực hiện - Giai đoạn 2 (Implementation - Sprint 2)	74
5.1 Phát triển MVP 2	74
5.1.1 Mô tả	74
5.1.2 Điểm khác biệt so với MVP 1	74
5.1.2.a Thay đổi giao diện	74
5.1.2.b Thêm chức năng Nhắn tin	77
5.1.2.c Thêm chức năng Thông báo	79
5.2 Trình bày dự án từ task 1 đến task 5	81
6 Công nghệ	81
6.1 UI/UX Design	81
6.1.1 Figma	81
6.2 Frontend	82
6.2.1 ReactJS	82
6.2.2 TailwindCSS	83
6.2.3 Kết hợp	84
6.3 Backend	84
6.3.1 NodeJS	84
6.3.2 ExpressJS	85
6.3.3 MongoDB	86
6.4 Realtime	87
6.4.1 Socket.IO	87
6.5 Deployment	88
6.5.1 Docker container	88
6.5.2 Web hosting service	89
6.6 Version Control	89
6.6.1 Git	89
6.6.2 Github	90
6.7 Project Management Tool	91
6.7.1 Trello	91
Tài liệu tham khảo	92



1 Thu thập yêu cầu (Requirement elicitation)

1.1 Tổng quan

1.1.1 Mô tả dự án

1.1.1.a Bối cảnh áp dụng (Domain Context)

Hiện nay, số lượng sinh viên ở trường Đại học Bách Khoa (HCMUT) ngày càng nhiều, nhu cầu in ấn cũng tăng lên đáng kể, trong khi các phòng in bên trong khuôn viên trường đang rất ít. Dẫn đến sinh viên phải in tại các cơ sở bên ngoài, rất bất tiện và thiếu tính bảo mật. Vì vậy, nhóm BigDaddy quyết định thực hiện dự án Student Smart Printing System (HCMUT_SSPS) - một hệ thống in ấn thông minh dành riêng cho sinh viên trường Đại học Bách Khoa (HCMUT).

Student Smart Printing System là một hệ thống nhiều máy in được kết nối qua website được bố trí rộng rãi khắp khuôn viên trường Đại học Bách Khoa. Sinh viên có thể chọn máy in qua website, tải tài liệu cần in và điều chỉnh các cấu hình in. Vào đầu mỗi học kỳ, sinh viên sẽ được cung cấp một số lượng trang in nhất định, sinh viên có thể mua thêm số lượng trang in thông qua BKPay.

Hệ thống các máy in được quản lý bởi Nhân viên quản lý in (SPSO). Nhân viên này có thể xem được nhật ký in của sinh viên và các máy in cũng như toàn quyền điều khiển máy in với các chức năng như thêm, kích hoạt và vô hiệu hóa. Ngoài ra, SPSO cũng là người quy định các định dạng file có thể được in và xem báo cáo về việc sử dụng hệ thống hàng tháng, hàng năm.

1.1.1.b Các bên liên quan (Stakeholders) - Nhu cầu và lợi ích

Stakeholder

Stakeholder, hay Bên Liên Quan, là những cá nhân, tổ chức, hoặc nhóm có ảnh hưởng hoặc quan tâm đến dự án hoặc hệ thống. Những cá nhân này thường có khả năng ảnh hưởng đến quyết định, kết quả, hoặc thành công của dự án. Stakeholder bao gồm nhiều loại, chẳng hạn như người sử dụng cuối cùng, nhà phát triển, quản lý dự án, và những bên liên quan từ các phòng ban khác. Mỗi stakeholder đều mang theo một tầm quan trọng đặc biệt và đóng góp vào quá trình phát triển và triển khai của hệ thống. Quản lý quan hệ với stakeholder đòi hỏi sự hiểu biết sâu sắc về mong muốn, nhu cầu, và lo ngại của họ để đảm bảo rằng dự án sẽ đáp ứng được mong đợi của tất cả các bên liên quan. Việc tương tác và giao tiếp mở cửa với stakeholder là quan trọng để duy trì sự hiểu biết và hỗ trợ liên tục trong suốt quá trình phát triển và vận hành hệ thống.

Dự án này bao gồm các stakeholder như:

- Sinh viên (Student)
- Nhân viên quản lý in (SPSO)
- Nhân viên trực quầy in (Staff)
- Đội ngũ phát triển dự án (Project Development Team)
- Trường Đại học Bách Khoa (University Administration)
- Nhà cung cấp thiết bị in ấn (Printing Equipment Supplier)

(a) Sinh viên

- Nhu cầu: hệ thống cần có một giao diện người dùng mạch lạc, dễ sử dụng và thân thiện với người dùng. Quan trọng nhất là cung cấp đủ tính năng để sinh viên có thể thực hiện các tác vụ cần thiết, bao gồm in ấn, xem lịch sử hoạt động, mua trang in và tiến hành thanh toán qua phương thức BKPay, xác thực thông tin qua hệ thống HCMUT_SSO và xem thông tin cá nhân. Ngoài ra việc gửi phản hồi feedback và gửi mail hoàn thành sau khi in xong là một điểm cộng
- Lợi ích: Tăng cường khả năng quản lý thời gian, tiện lợi và linh hoạt, tương tác và giao tiếp dễ dàng, nâng cao trải nghiệm của sinh viên, bảo mật thông tin cá nhân, tránh bị đánh cắp dữ liệu.



(b) Nhân viên quản lý in

- Nhu cầu: Ứng dụng phải đơn giản, thân thiện với người sử dụng. Song quan trọng nhất là tác vụ quản lý, tôi muốn xem nhật ký, thông tin liên quan đến việc in ấn của các sinh viên, hệ thống tự lập báo cáo hằng tháng, hằng năm nhằm phục vụ công tác đối soát khi có vấn đề phát sinh cũng như chỉnh sửa các cấu hình, tiêu chuẩn mặc định. Ngoài ra, có khả năng đăng nhập và đăng xuất cũng như, trang cá nhân trưng bày các thông tin cá nhân của tôi
- Lợi ích: Dễ kiểm soát truy cập, dễ quản lý máy in và tùy chỉnh hệ thống, các báo cáo vào đầu tháng, năm giúp SPSO dễ nắm bắt tài chính và số hóa quản lý.

(c) Nhân viên trực quầy in

- Nhu cầu: Ứng dụng dễ sử dụng, thân thiện. Chúng tôi cần có khả năng thay đổi trạng thái của máy in (kích hoạt và vô hiệu hóa) để xử lý khi có các tình huống xấu xảy ra với máy in. Có khả năng xem lịch sử in ấn và thông tin của máy in thuộc khu vực do tôi quản lý.
- Lợi ích: Tiện lợi và nhanh chóng trong việc cập nhật trạng thái máy in mà nhân viên trực.

(d) Đội ngũ phát triển dự án

- Nhu cầu: Các yêu cầu từ phía khách hàng (Đại học Bách Khoa) phải rõ ràng, cụ thể. Đặc biệt, nhà trường cần cấp phép chúng tôi khả năng truy cập tới API của BKPay hay HCMUT_SSO Authentication và các máy in ấn.
- Lợi ích: Tăng thêm nguồn thu nhập sau khi tham gia phát triển hệ thống, xây dựng hệ thống ý nghĩa với cộng đồng, cụ thể là sinh viên, học hỏi thêm về phương pháp xây dựng hệ thống ở trường học.

(e) Trường Đại học Bách Khoa

- Nhu cầu: Việc triển khai dự án phải đảm bảo tuân thủ quy định nhà trường và phải có bản báo cáo đầy đủ về dự án (tài chính, tiến độ,...). Nhà trường có khả năng kiểm soát hạ tầng trực tiếp và ứng dụng phải có khả năng tích hợp với hệ thống sẵn có của trường như BKPay và HCMUT_SSO.
- Lợi ích: Có thêm dịch vụ tiện ích, tối đa hóa lợi nhuận, giúp sinh viên có trải nghiệm tuyệt vời tại trường.

(f) Nhà cung cấp thiết bị in ấn

- Nhu cầu: Phía nhà trường cần phải cung cấp chi tiết số lượng máy in, mẫu mã cũng như số lượng trang giấy A3, A4,... cần cung cấp mỗi tháng. Nếu có trực trắc về mặt thiết bị, nhà trường cần liên hệ gấp để giải quyết.
- Lợi ích: Có quan hệ tốt và lâu dài với nhà trường. Dưới góc độ kinh tế, mang lại lợi nhuận tốt cho nhà cung cấp cũng như khả năng quảng bá sản phẩm máy in tới các bạn sinh viên.

1.1.2 Động cơ nghiên cứu

Hiện nay, nhu cầu in ấn tài liệu của sinh viên Bách Khoa ngày càng lớn và cũng gây ra một vài điểm bất cập và đó cũng là động lực nghiên cứu của nhóm BigDaddy để giải quyết các vấn đề sau:

- Quy trình phức tạp: Thông thường, quá trình in ấn tài liệu phải trải qua quy trình bắt tiên và phức tạp. Sinh viên phải tìm kiếm một tiệm in, chờ đợi lượt in, thông qua nhân viên để cài đặt các tùy chọn in và thậm chí phải đổi mặt với sự cố kỹ thuật hoặc hạn chế về số lượng trang in.
- Thiếu tiện lợi và linh hoạt: Hiện nay, việc in tài liệu thường yêu cầu sinh viên mang theo file cần in và tìm một cơ sở in gần đó. Điều này có thể gây ra khó khăn khi không có cơ sở nào gần khu vực sinh viên dẫn tới sinh viên phải in ở một địa điểm xa, gây tốn kém thời gian, đồng thời thiếu sự linh hoạt khi luôn phải thanh toán sau mỗi lần in.
- Hạn chế trong việc xem lịch sử in: Sinh viên hiện tại thường không thể xem lịch sử in của mình hoặc biết số lượng trang đã in trong một khoảng thời gian cụ thể. Điều này gây khó khăn trong việc theo dõi và đánh giá việc sử dụng tài liệu in của sinh viên.



- Thiếu tính bảo mật: nếu không có hệ thống quản lý in hiệu quả, việc bảo mật thông tin trong quá trình in có thể bị đe dọa. Các tài liệu quan trọng của sinh viên có thể bị mất hoặc bị lấy cắp, ngoài ra nhiều trường hợp sinh viên đăng nhập vào Zalo hay Drive để lấy file nhưng quên đăng xuất thiết bị.

Nắm bắt được yêu cầu đó, nhóm phác họa ý tưởng về một trang web hỗ trợ sinh viên trong việc in ấn tài liệu. Việc này không chỉ giúp sinh viên có thể in tài liệu nhanh chóng mà còn tạo ra một hệ thống ở phạm vi trường học, giúp quản lý tài liệu một cách hiệu quả và có tính bảo mật cao.

1.1.3 Mục tiêu

Với những bất cập đã nêu trên, mục tiêu của dự án Student Smart Printing System (HCMUT_SSPS) chính là mang lại lợi ích sau cho nhà trường và các bạn sinh viên:

- Quy trình đơn giản, dễ dàng: sinh viên chỉ cần truy cập vào Website thông qua máy tính hoặc điện thoại rồi xác thực qua HCMUT_SSO authentication service, sau đó sinh viên sử dụng trang giấy có sẵn (được cấp phát ở mỗi học kỳ hoặc mua thông qua BKPay) trong tài khoản rồi tải tài liệu lên, sử dụng các tính năng in ấn.
- Tiện lợi và linh hoạt: sinh viên không cần phải tiếp cận các dịch vụ in ấn bên ngoài mà chỉ cần truy cập vào Website và sử dụng số trang in đã thanh toán sẵn trước đó. Giúp sinh viên tiết kiệm được thời gian và sức lực, tăng tính trải nghiệm và sự hài lòng với các dịch vụ ở trường đại học.
- Số hóa quản lý: với việc áp dụng Student Smart Printing System, hệ thống giờ đây có thể lưu lại lịch sử sử dụng hệ thống của từng sinh viên cũng như tự động ghi lại báo cáo vào cuối tháng và cuối năm giúp cho SPSO quản lý trực quan, minh bạch và tiện lợi, từ đó có thể xác định được những cải thiện có thể triển khai lên hệ thống.
- Quản lý chặt chẽ: SPSO có toàn quyền điều khiển máy in ấn như thêm, kích hoạt và vô hiệu hóa máy in. Điều này giúp hệ thống có thể cập nhật trạng thái lập tức khi có một máy in được thêm vào hệ thống hay máy in hiện đang gặp lỗi và không thể sử dụng được cũng như được kích hoạt sử dụng.
- An toàn bảo mật: Đảm bảo các tài liệu và thông tin tải lên của sinh viên không bị đánh cắp hoặc rò rỉ ra bên ngoài. Nếu không có hệ thống quản lý in hiệu quả, việc bảo mật thông tin trong quá trình in có thể bị đe dọa và thông tin bị sử dụng trái phép.

1.2 Yêu cầu chức năng và phi chức năng

1.2.1 Cơ sở lý thuyết

Yêu cầu chức năng

Yêu cầu chức năng là những đặc điểm và khả năng cụ thể mà hệ thống phải có để thực hiện các chức năng hoặc nhiệm vụ cụ thể. Những yêu cầu này mô tả những hoạt động và tính năng cụ thể mà người sử dụng mong đợi từ hệ thống, thường được biểu diễn dưới dạng quy trình, đầu vào và đầu ra. Yêu cầu chức năng là hướng dẫn chi tiết về cách hệ thống nên hoạt động để đáp ứng mong đợi của người dùng.

Yêu cầu phi chức năng

Yêu cầu phi chức năng liên quan đến những thuộc tính và khía cạnh không trực tiếp liên quan đến chức năng cụ thể của hệ thống. Thường xuyên, những yêu cầu này đề cập đến các khía cạnh quan trọng như hiệu suất, bảo mật, sự linh hoạt, độ tin cậy, và trải nghiệm người dùng. Yêu cầu phi chức năng chịu trách nhiệm đảm bảo rằng hệ thống không chỉ thực hiện chức năng một cách hiệu quả mà còn đáp ứng được các tiêu chí quan trọng khác như an ninh và khả năng mở rộng.



1.2.2 Yêu cầu chức năng

1.2.2.a Góc độ người dùng (sinh viên)

Với tư cách là khách hàng, sinh viên chúng tôi muốn mình được tạo tài khoản, đăng nhập và đăng xuất, trang cá nhân trưng bày các thông tin cá nhân của tôi; chúng tôi muốn xem nhật ký in ấn, thông tin liên quan; thanh toán trực tuyến và có mail gửi về khi hoàn thành in ấn là một điểm cộng; song quan trọng nhất là phần mềm này **phải in được**.

Trên cơ sở những mong mỏi từ người dùng, nhóm tác giả thực hiện các tính năng sau:

Đăng nhập & Bảo mật (Login & Security)

- Để đảm bảo an toàn dữ liệu nhạy cảm, hệ thống có tích hợp tính năng bảo mật đó là xác thực người dùng. Việc xác thực được thực hiện thông qua dịch vụ xác thực HCMUT_SSO.
- Tự động đăng nhập vào hệ thống sau xác thực. Sinh viên không cần đăng ký tài khoản.
- Sinh viên chỉ cần dùng nút Đăng xuất để thoát.

In ấn và kiểm soát tiêu chuẩn (Printing & Standards controlling)

- Hệ thống có cài đặt và quản lý các thiết lập in ấn như màu sắc (in màu/trắng đen), kích thước giấy (A3, A4...), quy cách in và số lượng bản in, trang in.
- Sinh viên đặt lệnh in từ thiết bị (đã đăng nhập) của mình. Sau khi việc in ấn hoàn tất, hệ thống sẽ gửi mail hoàn thành in ấn về cho sinh viên. Sinh viên có thể hủy lệnh in và lượng trang giấy sẽ được cập nhật lại nếu như trạng thái in là đang xử lý.
- Hệ thống sẽ tính 2 trang giấy A4 cho mỗi lần in của 1 trang giấy A3 khi cập nhật lại số lượng giấy còn lại của sinh viên sau mỗi lần in.

Quản lý tài khoản (Profile management)

- Cho phép sinh viên xem thông tin và số lượng giấy in còn lại của tài khoản
- Đồng thời cho phép sinh viên theo dõi tình trạng lệnh in (đang xử lý, trong tiến trình, đã xong...).
- Xem các thông tin liên quan đến nhật ký in ấn.
- Cho phép sinh viên gửi phản hồi và đóng góp về hệ thống để nâng cấp và khắc phục sự cố kịp thời.

Thanh toán (Payment method)

- Hệ thống liên kết với hệ thống thanh toán trực tuyến như BKPay. Sau khi thanh toán qua BKPay, hệ thống sẽ cập nhật lượng giấy in cho tài khoản.
- Bên cạnh đó, mỗi sinh viên có thông tin lượng giấy in hiện tại cho mình, tất cả nhằm in ấn không dùng tiền mặt.

1.2.2.b Góc độ người quản lý (SPSO)

Với tư cách là nhân viên quản lý, tôi muốn mình được tạo tài khoản, đăng nhập và đăng xuất, trang cá nhân trưng bày các thông tin cá nhân của tôi; tôi muốn xem nhật ký, thông tin liên quan đến việc in ấn của các sinh viên, hệ thống tự lập báo cáo hàng tháng, hàng năm nhằm phục vụ công tác đối soát khi có vấn đề phát sinh, đồng thời tôi muốn có khả năng thêm, kích hoạt và vô hiệu hóa máy in; song quan trọng nhất vẫn là **tác vụ quản lý**.

Từ những nhu cầu của người quản lý, nhóm tác giả thực hiện các tính năng sau:

Đăng nhập & Bảo mật (Login & Security)

- Để đảm bảo an toàn dữ liệu nhạy cảm, hệ thống có tích hợp tính năng bảo mật đó là xác thực người dùng. Việc xác thực được thực hiện thông qua dịch vụ xác thực HCMUT_SSO..
- Tự động đăng nhập vào hệ thống sau xác thực. Nhân viên không cần đăng ký tài khoản.
- Nhân viên chỉ cần dùng nút Đăng xuất để thoát.



Quản lý 2 chiều (2-way Management)

- Quản lý thông minh: hệ thống cần có khả năng tự động nhận biết, giám sát các công việc in ấn, đặc biệt là quản lý hàng đợi công việc.
- Quản lý thủ công: người quản lý dịch vụ thao tác và ra lệnh cho tập các máy in qua hệ thống này. Anh ta/cô ta được quyền chỉnh sửa cấu hình, chẳng hạn: số trang in mặc định, loại tệp được phép in, và ngày hệ thống mặc định cung cấp lượng trang in miễn phí cho sinh viên. Đồng thời, cho phép nhân viên theo dõi tình trạng các lệnh in (đang xử lý, trong tiến trình, đã xong...) tại tất cả các máy cũng như xem thông tin liên quan đến nhật ký in ấn của tất cả sinh viên. Bên cạnh đó, họ có thể thêm, kích hoạt và vô hiệu hóa các máy in.

Thống kê và báo cáo (Statistics and reporting)

- Hệ thống cần cung cấp khả năng theo dõi, ghi nhật ký và tạo báo cáo hàng tháng, hàng năm về việc sử dụng máy in, bao gồm các thông tin như số trang đã in, ai đã sử dụng, sử dụng máy nào, số trang in của mỗi loại khổ giấy, tiêu đề tệp...

Quản lý tài khoản (Profile management)

- Cho phép nhân viên xem thông tin tài khoản.

1.2.2.c Góc độ nhân viên trực quầy (Staff)

Với tư cách là nhân viên trực quầy, tôi muốn có khả năng đăng nhập và đăng xuất, tôi muốn xem nhật ký, thông tin liên quan đến việc in ấn của các sinh viên tại khu vực máy in tôi quản lý, song quan trọng nhất là khả năng **thay đổi trạng thái máy in** khi có sự cố xáu xảy ra với máy in.

Từ những nhu cầu của nhân viên trực quầy, nhóm tác giả thực hiện các tính năng sau:

Dăng nhập & Bảo mật (Login & Security)

- Để đảm bảo an toàn dữ liệu nhạy cảm, hệ thống có tích hợp tính năng bảo mật đó là xác thực người dùng. Việc xác thực được thực hiện thông qua dịch vụ xác thực HCMUT_SSO..
- Tự động đăng nhập vào hệ thống sau xác thực. Nhân viên không cần đăng ký tài khoản.
- Nhân viên chỉ cần dùng nút đăng xuất để thoát.

Quản lý thủ công (Manual Management)

- Người nhân viên trực quầy dịch vụ thao tác với máy in qua hệ thống này. Anh ta/cô ta được quyền truy xuất lịch sử in của máy in mà trực đồng thời có khả năng cập nhật trạng thái máy in hiện tại lên hệ thống cũng như tình trạng các lệnh in (đang xử lý, trong tiến trình, đã xong...).

Quản lý tài khoản (Profile management)

- Cho phép nhân viên xem thông tin tài khoản.

1.2.3 Yêu cầu phi chức năng

Trong phần này, chúng ta sẽ xem xét yêu cầu phi chức năng của Dịch Vụ In Thông Minh cho Sinh viên HCMUT (HCMUT_SSPS). Yêu cầu phi chức năng bao gồm các khía cạnh quan trọng như hiệu suất, bảo mật, đáng tin cậy, khả năng sẵn có, khả năng mở rộng và tính sẵn sàng của hệ thống. Các yêu cầu này giúp đảm bảo rằng hệ thống không chỉ hoạt động một cách hiệu quả mà còn đảm bảo tính an toàn và tin cậy cho người dùng, cũng như khả năng mở rộng và sẵn sàng để đáp ứng các yêu cầu tương lai.



1.2.3.a Yêu cầu phi chức năng cho toàn hệ thống

Tính khả dụng

- Giao diện của hệ thống phải hiển thị đầy đủ các tính năng chính, tuân thủ quy định và đáp ứng nhu cầu từ phía nhà trường.
- Đối với sinh viên có thể sử dụng các chức năng in ấn cơ bản sau 10 phút bắt đầu trải nghiệm sản phẩm.
- Đối với người quản lý và nhà trường có thể sử dụng toàn bộ hệ thống sau 30 phút hướng dẫn.

Hiệu suất

- Thời gian phản hồi: Thời gian phản hồi tương tác với sinh viên tối đa 2 giây trong điều kiện kết nối bình thường, và thời gian khi tải tài liệu in lên hệ thống tối đa là 5 giây.
- Hệ thống hỗ trợ chạy real-time.

Tính bảo mật

- Thông tin của sinh viên sẽ được bảo mật sau khi đăng nhập vào hệ thống và không bị rò rỉ.
- Mật khẩu của người dùng phải được mã hóa.

Tính mở rộng

- Hệ thống có khả năng mở rộng thêm các tính năng trong tương lai nếu có thêm yêu cầu, thêm các module mà không ảnh hưởng đến cấu trúc hệ thống.

Tính tin cậy

- Thời gian trung bình giữa các sự cố lớn về hệ thống lớn hơn 6 tháng.

Tính sẵn sàng

- Downtime của website mỗi tháng không quá 1 giờ.
- Hệ thống có thể đáp ứng tối đa 200 truy cập cùng một lúc mà không bị ảnh hưởng đến trải nghiệm của sinh viên.

Tính tương thích

- Hệ thống có khả năng tương thích với các phiên bản trình duyệt phổ biến như Chrome, Firefox, Safari, và Edge.
- Responsive website với kích thước dao động từ điện thoại tới máy tính bảng và laptop.
- Hệ thống in ấn cần tương thích và có khả năng tích hợp vào các ứng dụng và cơ sở hạ tầng hệ thống của trường.

1.2.3.b Yêu cầu phi chức năng cho các tính năng

- Hệ thống hỗ trợ cập nhật thông tin in ấn của người dùng và tài liệu một cách real-time với thời gian phản hồi tối đa là 2s.
- Danh sách máy in tự động cập nhật theo địa điểm, trạng thái và các yêu cầu in ấn sẽ được cập nhật một cách tự động theo thời gian thực với thời gian cập nhật tối đa là 2s.
- Thông báo người dùng về tình trạng in ấn và cập nhật thông tin với thời gian phản hồi tối đa là 3 giây.



1.3 Biểu đồ use-case và kịch bản use-case

1.3.1 Cơ sở lý thuyết

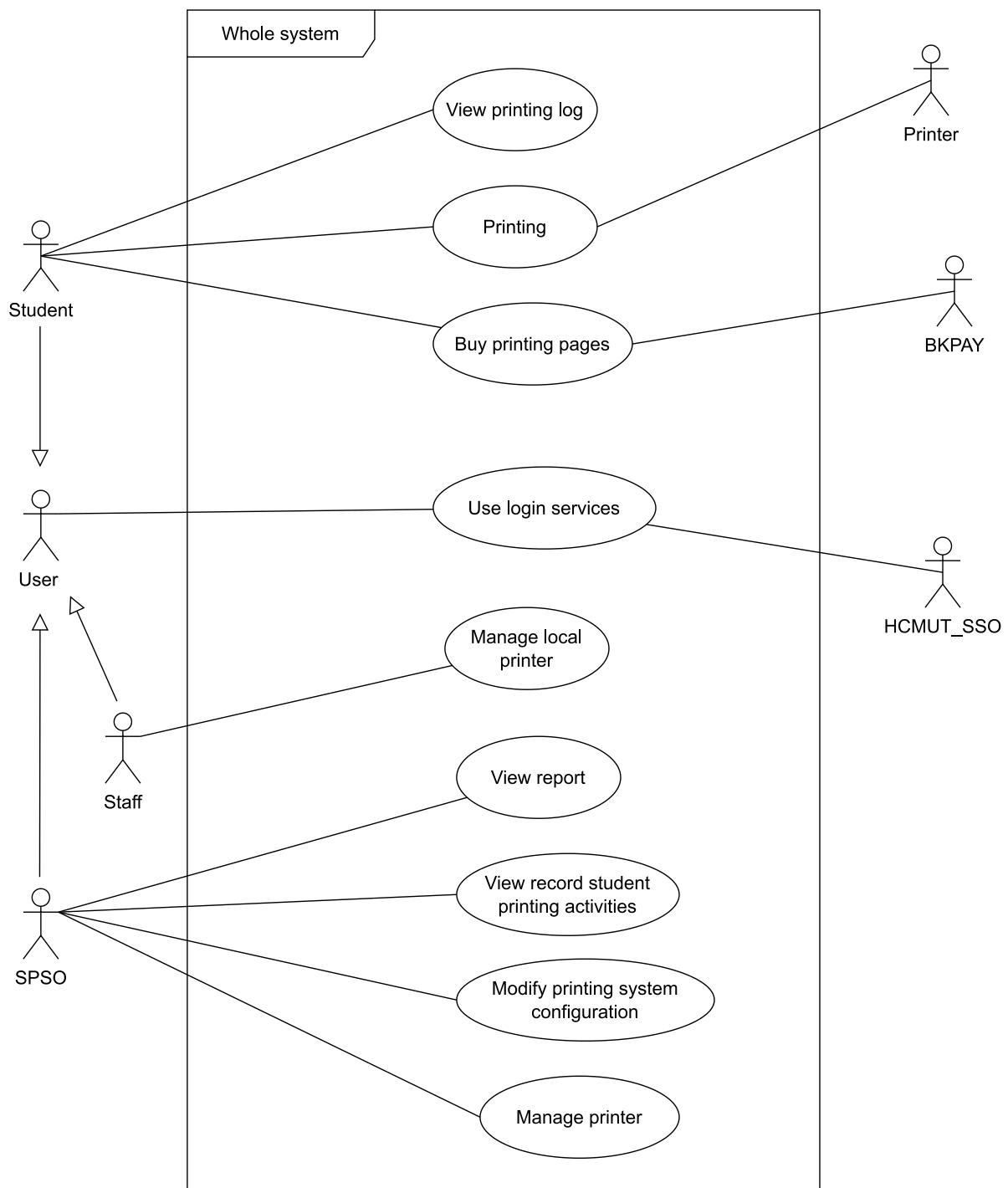
Biểu đồ use-case

Biểu đồ trường hợp - sử dụng (use-case diagram) thường được gọi là biểu đồ hành vi, được sử dụng để mô tả một tập hợp các hành động mà một số hệ thống hoặc hệ thống (chủ thể) nên hoặc có thể thực hiện với sự cộng tác của một hoặc nhiều người dùng bên ngoài của hệ thống (tác nhân - actor). Mỗi trường hợp sử dụng phải cung cấp một số kết quả có giá trị và có thể quan sát được cho các tác nhân hoặc các bên liên quan khác của hệ thống.

Biểu đồ use-case dùng để chỉ định:

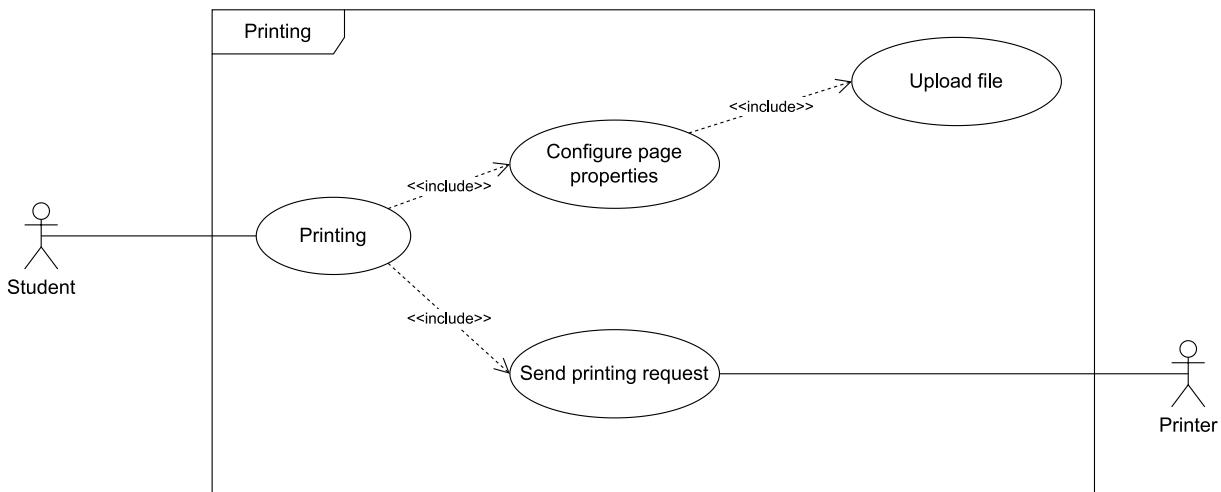
1. Các yêu cầu (bên ngoài), cách sử dụng bắt buộc của một hệ thống đang được thiết kế hoặc phân tích (chủ thể) - để nắm bắt những gì hệ thống phải làm.
2. Chức năng được cung cấp bởi một chủ thể – hệ thống có thể làm gì
3. Yêu cầu mà chủ thể được chỉ định đặt ra trên môi trường của nó - bằng cách xác định cách môi trường nên tương tác với chủ thể để nó có thể thực hiện các dịch vụ của mình.

1.3.2 Use-case: Whole system



1.3.3 Use-case: Printing

Use-case diagram



Use-case scenario: Printing

Tên Use-case	Printing
Actor	Primary: Sinh viên Secondary: Printer
Description	Sinh viên sử dụng dịch vụ in ấn của hệ thống SSPS
Trigger	Sau khi sinh viên ấn vào nút “In tài liệu” trên sidebar
Precondition	Sinh viên phải đăng nhập vào hệ thống thông qua HCMUT_SSO
Postcondition	Tài liệu của sinh viên được in thành công
Normal Flows	1. Hệ thống hiển thị giao diện “Chọn máy in” gồm danh sách các máy in và thông tin cho mỗi máy in 2. Sinh viên chọn máy in 3. Chuyển sang use-case "Configure page properties" 4. Chuyển sang use-case "Send printing request"
Alternative Flows	None
Exception Flows	None

Use-case scenario: Configure page properties

Tên Use-case	Configure page properties
Actor	Sinh viên
Description	Sinh viên cấu hình cho các file cần in
Trigger	Sau khi sinh viên nhấn vào máy in muốn chọn để in
Precondition	Sinh viên phải đăng nhập vào hệ thống thông qua HCMUT_SSO
Postcondition	Sinh viên cấu hình file in thành công
Normal Flows	1. Hệ thống hiển thị giao diện “Cấu hình file in” 2. Sinh viên chọn các cấu hình theo ý muốn.
Alternative Flows	None
Exception Flows	None

**Use-case scenario: Upload file**

Tên Use-case	Upload file
Actor	Sinh viên
Description	Sinh viên tải các file cần in lên hệ thống
Trigger	Sinh viên bấm vào ô “Tải tệp lên” trên giao diện “Cấu hình file in”
Precondition	Sinh viên tải thành công các file cần in lên hệ thống
Postcondition	Tài liệu của sinh viên được in thành công
Normal Flows	1. Hệ thống hiển thị cửa sổ file dialog cho phép sinh viên chọn các file từ máy của mình 2. Sinh viên chọn file muốn in
Alternative Flows	Tại bước 2, 2.1. Sinh viên chọn “Thêm tệp” để tải lên các file khác. Quay lại bước 1. Tại bước 2, 2.2. Sinh viên chọn “Xóa tệp” 3.2. Hệ thống hiển thị danh sách các file đã được tải lên 4.2. Sinh viên chọn tệp muốn xóa
Exception Flows	None

Use-case scenario: Send printing request

Tên Use-case	Send printing request
Actor	Primary: Sinh viên Secondary: Printer
Description	Sinh viên gửi yêu cầu in ấn
Trigger	Sinh viên nhấn nút “Gửi yêu cầu in” ở cuối trang “Cấu hình file in”
Precondition	Sinh viên phải đăng nhập vào hệ thống thông qua HCMUT_SSO
Postcondition	Sinh viên gửi yêu cầu in ấn thành công
Normal Flows	1. Hệ thống kiểm tra số dư trong tài khoản của sinh viên 2. Hệ thống xác nhận số dư đủ để thực hiện yêu cầu in ấn 3. Hệ thống cập nhật hàng đợi mới cho máy in 4. Hệ thống gửi email cho sinh viên sau khi in xong
Alternative Flows	None
Exception Flows	Tại bước 2, 2.1. Hệ thống xác nhận số dư không đủ để thực hiện yêu cầu in ấn 3.1 Hệ thống hiển thị thông báo in thất bại



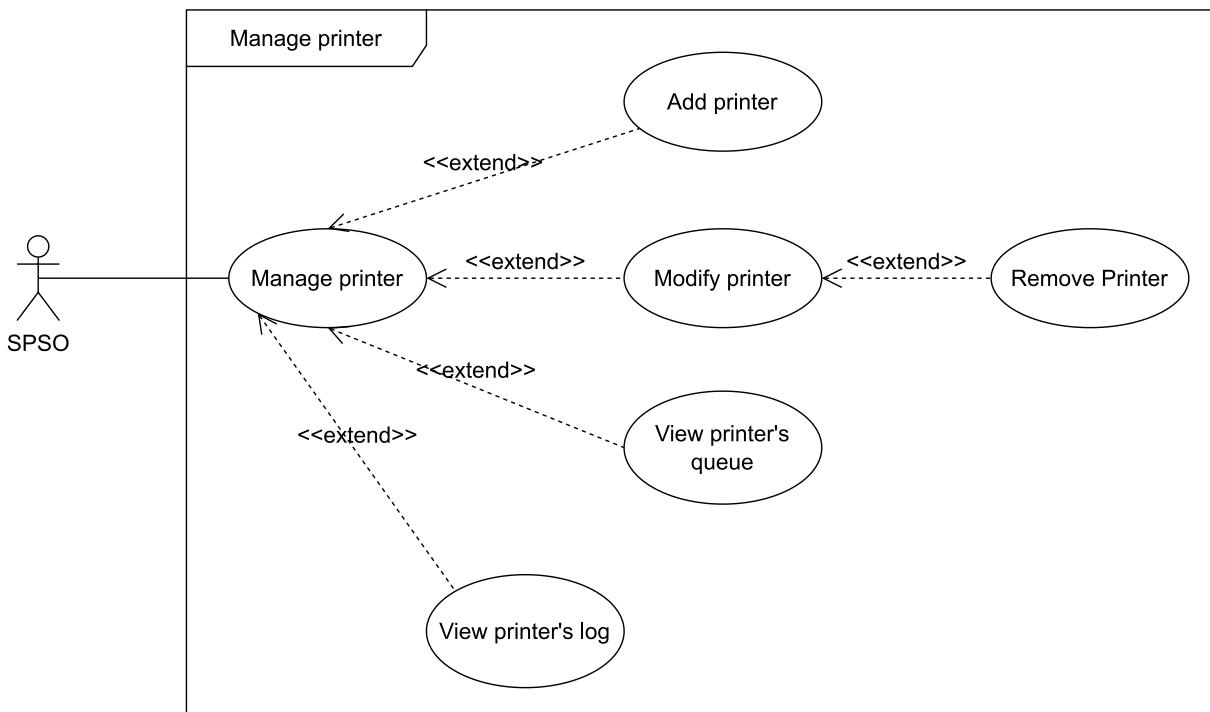
1.3.4 Use-case: Modify system configuration

Use-case scenario: Modify system configuration

Tên Use-case	Modify system configuration
Actor	SPSO
Description	SPSO muốn quản lý cấu hình hệ thống in ấn
Trigger	SPSO nhấn vào mục "Cấu hình hệ thống" trên sidebar
Precondition	SPSO phải đăng nhập vào hệ thống thông qua HCMUT_SSO
Postcondition	SPSO thay đổi thành công thông số cấu hình hệ thống
Normal Flows	1. Hệ thống hiển thị giao diện cấu hình hệ thống. 2. SPSO cập nhật cấu hình hệ thống mới và ấn "Áp dụng". 3. Hệ thống lưu lại cấu hình.
Alternative Flows	Alternative 1: Tại bước 2 2.1 SPSO chọn "Mặc định" rồi ấn "Áp dụng". UC tiếp tục bước 3.
Exception Flows	Exception 1: Tại bước 2 2.1 SPSO tắt giao diện cấu hình. UC dừng lại

1.3.5 Use-case: Manage printer

Use-case diagram



Use-case scenario: Manage Printer

Tên Use-case	Manage printer
Actor	SPSO
Description	SPSO muốn quản lý một máy in trong hệ thống
Trigger	SPSO nhấn vào mục "Hệ thống máy in" trên sidebar
Precondition	SPSO phải đăng nhập vào hệ thống thông qua HCMUT_SSO
Postcondition	Hệ thống hiển thị đầy đủ danh sách các máy in hiện tại
Normal Flows	<ol style="list-style-type: none"> Hệ thống hiển thị danh sách các máy in SPSO duyệt qua từng trang danh sách và không tương tác SPSO xem thông tin một máy in trên hàng giao diện ứng với máy đó SPSO xem cấu hình chi tiết bằng cách ấn vào nút "Cấu hình" trên hàng giao diện ứng với máy đó.
Alternative Flows	Alternative 1: Tại bước 2: SPSO có tương tác <u>Extended points:</u> UC Add printer, Modify printer, View printer's queue, View printer's log Alternative 2: Tại bước 2: 2.1 SPSO tìm kiếm một / một số máy in qua ô tìm kiếm / bộ lọc. UC tiếp tục bước 3.
Exception Flows	None

Use-case scenario: Add printer

Tên Use-case	Add printer
--------------	-------------



Actor	SPSO
Description	SPSO muốn thêm một máy in mới
Trigger	SPSO nhấn vào nút "Thêm máy in" trên giao diện hệ thống máy in
Precondition	SPSO phải đăng nhập vào hệ thống thông qua HCMUT_SSO
Postcondition	SPSO thêm máy in thành công
Normal Flows	1. Hệ thống hiển thị cửa sổ thêm máy in 2. SPSO nhập liệu thông tin máy in mới, đặt trạng thái và nhấn Hoàn tất. 3. Hệ thống lưu lại thông tin về máy in mới.
Alternative Flows	None
Exception Flows	Exception 1: Tại bước 2, SPSO quên / không nhấn hoàn tất hoặc nhấn nhầm ra ngoài cửa sổ. UC dừng lại.

Use-case scenario: Modify printer

Tên Use-case	Modify printer
Actor	SPSO
Description	SPSO muốn sửa đổi các thông tin, trạng thái của một máy in nào đó.
Trigger	SPSO nhấn vào nút "Cấu hình" trên hàng tương ứng với máy in đó.
Precondition	SPSO phải đăng nhập vào hệ thống thông qua HCMUT_SSO
Postcondition	SPSO sửa thông tin máy in thành công
Normal Flows	1. Hệ thống hiển thị cửa sổ Cấu hình máy in và tùy chỉnh 2. SPSO sửa đổi thông tin máy in và trạng thái. 3. SPSO ấn hoàn tất. 4. Hệ thống lưu lại thông tin, trạng thái mới của máy in.
Alternative Flows	Alternative 1: Tại bước 2, SPSO muốn xóa máy in. Chuyển đến UC Remove printer.
Exception Flows	Exception 1: Tại bước 3, SPSO quên / không nhấn hoàn tất hoặc nhấn nhầm ra ngoài cửa sổ. UC dừng lại.

Use-case scenario: Remove printer

Tên Use-case	Remove printer
Actor	SPSO
Description	SPSO muốn xóa một máy in nào đó.
Trigger	SPSO nhấn vào nút "Hiện tại" trên hàng tương ứng với máy in đó.
Precondition	SPSO phải đăng nhập vào hệ thống thông qua HCMUT_SSO
Postcondition	SPSO xem được hàng đợi của máy in
Normal Flows	1. Hệ thống hiển thị cửa sổ Cấu hình máy in và tùy chỉnh 2. SPSO ấn nút "Xóa máy in". 3. Hệ thống xóa dữ liệu máy in.



Alternative Flows	None
Exception Flows	None

Use-case scenario: View printer's queue

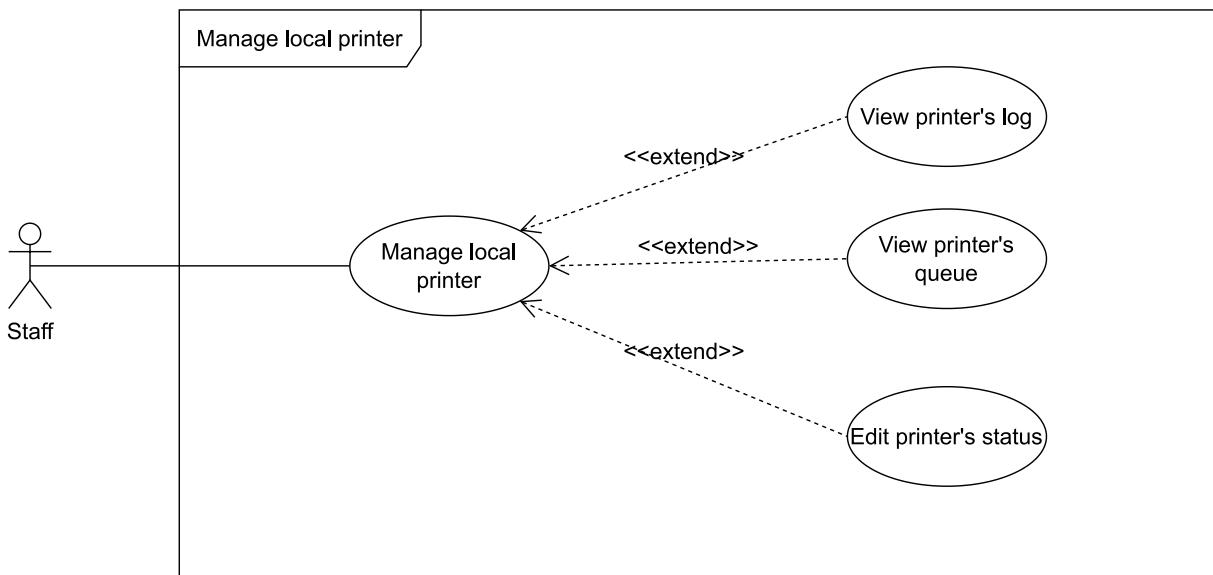
Tên Use-case	View printer's queue
Actor	SPSO
Description	SPSO muốn xem hàng đợi của một máy in nào đó.
Trigger	SPSO nhấn vào nút "Hiện tại" trên hàng tương ứng với máy in đó.
Precondition	SPSO phải đăng nhập vào hệ thống thông qua HCMUT_SSO
Postcondition	SPSO xem được hàng đợi của máy in
Normal Flows	1. Hệ thống hiển thị cửa sổ Hàng đợi 2. SPSO xem các thông tin về hàng đợi.
Alternative Flows	None
Exception Flows	None

Use-case scenario: View printer's log

Tên Use-case	View printer's log
Actor	SPSO
Description	SPSO muốn xem lịch sử in ấn của một máy in
Trigger	SPSO nhấn vào nút "Lịch sử" trên hàng giao diện ứng với của máy in muốn xem
Precondition	SPSO phải đăng nhập vào hệ thống thông qua HCMUT_SSO
Postcondition	SPSO xem lịch sử máy in thành công
Normal Flows	1. Hệ thống hiển thị cửa sổ lịch sử in ấn 2. SPSO xem lịch sử in của máy đó. 3. SPSO sử dụng bộ lọc (theo khoảng thời gian) để xem những mục mình cần.
Alternative Flows	None
Exception Flows	None

1.3.6 Use-case: Manage local printer

Use-case diagram



Use-case scenario: Manage local printer

Tên Use-case	Manage local printer
Actor	Staff
Description	Staff muốn xem danh sách các máy in trong nơi mình quản lý
Trigger	Staff nhấn vào mục “Hệ thống máy in” trên sidebar
Precondition	Staff phải đăng nhập vào hệ thống thông qua HCMUT_SSO
Postcondition	Hệ thống hiển thị đầy đủ danh sách các máy in hiện tại thuộc quản lý của staff
Normal Flows	1. Hệ thống hiển thị danh sách các máy in 2. Staff xem danh sách các máy in hiện tại 3. Staff xem thông tin của một máy in nào đó trên hàng giao diện tương ứng 4. Staff xem thông tin chi tiết bằng cách ấn vào nút "Cáu hình" trên hàng giao diện ứng với máy.
Alternative Flows	Alternative 1: Tại bước 3, Staff: <u>Extended points:</u> UC View printer's log, View printer's queue, Edit printer status Alternative 2: Sau bước 2, 3. Staff tìm kiếm một / một số máy in qua ô tìm kiếm. UC tiếp tục bước 3.
Exception Flows	None

Use-case scenario: View printer's log

Tên Use-case	View printer's log
Actor	Staff
Description	Staff muốn xem lịch sử in ấn của một máy in



Trigger	Staff nhấn vào nút "Lịch sử" trên hàng giao diện ứng với cửa máy in muốn xem
Precondition	Staff phải đăng nhập vào hệ thống thông qua HCMUT_SSO
Postcondition	Staff xem lịch sử máy in thành công
Normal Flows	1. Hệ thống hiển thị cửa sổ lịch sử in ấn 2. Staff xem lịch sử in của máy đó.
Alternative Flows	None
Exception Flows	None

Use-case scenario: View printer's queue

Tên Use-case	View printer's queue
Actor	Staff
Description	Staff muốn xem hàng đợi in ấn của một máy in
Trigger	Staff nhấn vào nút "Chi tiết" trên hàng giao diện của máy in muốn xem
Precondition	Staff phải đăng nhập vào hệ thống thông qua HCMUT_SSO
Postcondition	Staff xem được hàng đợi của máy in
Normal Flows	1. Hệ thống hiển thị cửa sổ hàng đợi in ấn 2. Staff xem hàng đợi in của máy đó.
Alternative Flows	None
Exception Flows	None

Use-case scenario: Edit printer's status

Tên Use-case	Edit printer's status
Actor	Staff
Description	Staff muốn thay đổi trạng thái máy in (vì lý do kỹ thuật v.v...)
Trigger	Staff nhấn vào nút "Cấu hình" trên hàng giao diện tương ứng với máy in đó.
Precondition	Staff phải đăng nhập vào hệ thống thông qua HCMUT_SSO
Postcondition	Staff xem được hàng đợi của máy in
Normal Flows	1. Hệ thống hiển thị cửa sổ Cấu hình máy in và tùy chỉnh 2. Staff thay đổi trạng thái của máy in và ấn Hoàn tất. 3. Hệ thống lưu lại thông tin về trạng thái máy in.
Alternative Flows	None
Exception Flows	Exception 1: Tại bước 2, SPSO quên / không nhấn hoàn tất hoặc nhấn nhầm ra ngoài cửa sổ. UC dừng lại.



2 Mô hình hóa hệ thống (System modeling)

2.1 Biểu đồ hoạt động (Activity Diagram)

2.1.1 Cơ sở lý thuyết

Biểu đồ hoạt động

Ngôn ngữ mô hình hóa thống nhất (UML) bao gồm một số tập hợp con biểu đồ, bao gồm biểu đồ cấu trúc, biểu đồ tương tác và biểu đồ hành vi. Biểu đồ hoạt động (activity diagram), cùng với biểu đồ trường hợp sử dụng và state-machine, được coi là biểu đồ hành vi vì chúng mô tả những gì phải xảy ra trong hệ thống được mô hình hóa. Biểu đồ hoạt động giúp mọi người ở khía cạnh kinh doanh và phát triển của một tổ chức cùng nhau giải quyết hiểu quả trình và hành vi tương tự. Một số thành phần phổ biến nhất của sơ đồ hoạt động bao gồm:

1. Hành động: Một bước trong hoạt động mà người dùng hoặc phần mềm thực hiện một tác vụ nhất định. Các hành động được biểu tượng bằng các hình chữ nhật bo góc.
2. Nút quyết định: Một nhánh có điều kiện trong luồng được biểu thị bằng hình thoi. Nó bao gồm một hoặc nhiều đầu vào và hai hoặc nhiều đầu ra.
3. Luồng điều khiển: Tên gọi khác của các đầu nối hiển thị luồng giữa các bước trong sơ đồ.
4. Nút bắt đầu: Tượng trưng cho sự bắt đầu của hoạt động. Nút bắt đầu được thể hiện bằng một vòng tròn màu đen.
5. Nút kết thúc: Biểu thị bước cuối cùng trong hoạt động. Nút cuối được biểu diễn bằng một vòng tròn màu đen có đường viền đỏ bên ngoài.

2.1.2 Chức năng Printing

Mô tả: Biểu đồ hoạt động của use-case: Printing bao gồm 4 swimlane là: Sinh viên, hệ thống SSPS, Printer, và Database.

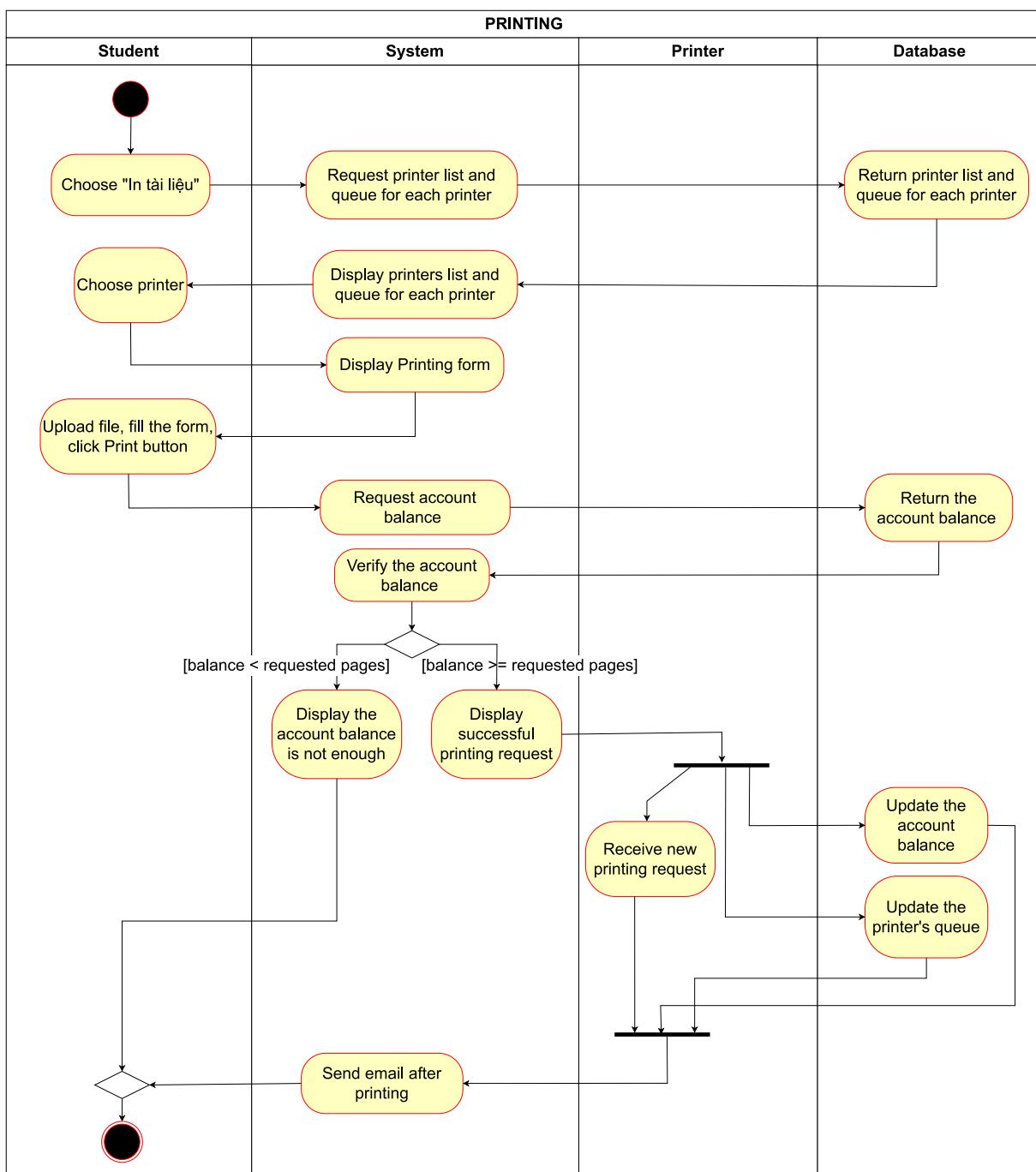
Dầu tiên, sinh viên chọn mục "In tài liệu" trên thanh sidebar, hệ thống sẽ yêu cầu database trả về danh sách máy in hiện tại, sau đó hiển thị ra màn hình qua giao diện "Chọn máy in".

Sau khi sinh viên chọn máy in, hệ thống hiển thị giao diện form "Cấu hình in ấn". Tại đây, sinh viên chọn tải lên các file cần in và tùy chỉnh các thông tin cấu hình cho các file in đó.

Tiếp theo, hệ thống sẽ kiểm tra số dư tài khoản của sinh viên.

Nếu số dư đủ, hệ thống hiển thị thông báo in thành công và cập nhật số dư mới trong database, cũng như cập nhật hàng đợi của máy in trong database.

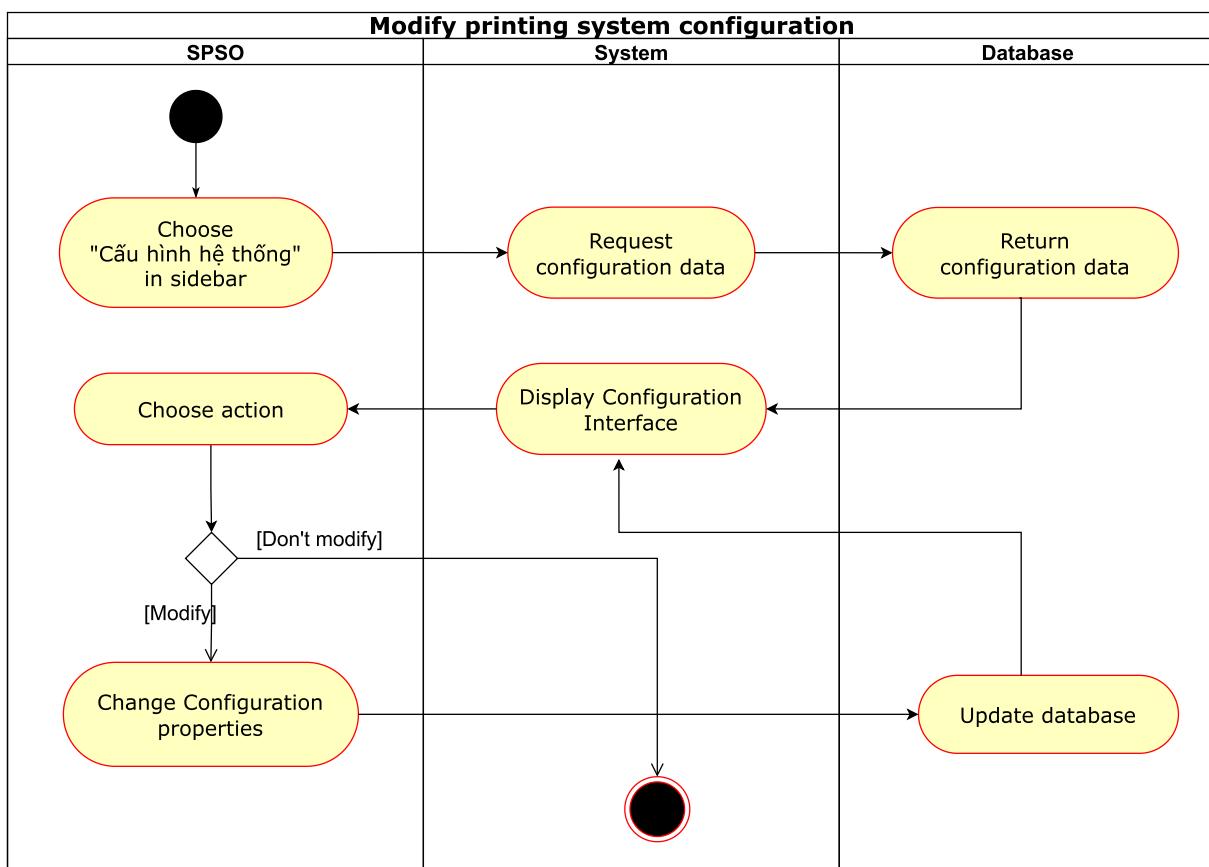
Nếu số dư không đủ, hệ thống hiển thị thông báo in thất bại và đề nghị sinh viên mua thêm giấy in.



2.1.3 Chức năng Modify Printing System Configuration

Mô tả: Biểu đồ này gồm 3 swimlane là: SPSO, hệ thống SSPS, database. Ban đầu SPSO chọn nút “Cấu hình hệ thống” ở thanh bên. Hệ thống cần hiển thị thông tin cấu hình đã lưu trước đó, do vậy yêu cầu database gửi dữ liệu về, sau đó các thông số được hiển thị ra màn hình.

Tiếp đến, nếu SPSO không có chỉnh sửa gì thêm, activity diagram kết thúc. Ngược lại, SPSO sẽ chỉnh sửa một hay nhiều thông số. Database được cập nhật. Các thông số mới được hiển thị ra màn hình.



2.1.4 Chức năng Manage Printer

Mô tả Activity diagram của Manage Printer

Activity diagram của Manage Printers sẽ gồm hai swimlane chính là: SPSO và hệ thống SSPS.

Khi SPSO chọn tùy chọn "Hệ Thống Máy In" trên Sidebar, hệ thống hiển thị giao diện Trang quản lý hệ thống máy in. Trên trang quản lý hệ thống máy in, SPSO có thể thực hiện các hoạt động:

- Chọn "Thêm máy in" ([Add printer]) để thêm máy mới cho hệ thống.
- Chọn "Hiện tại" ([View printer's queue]) trên một máy in để xem số yêu cầu in ấn hiện tại đang có trong máy in.
- Chọn "Lịch sử" ([View printer's log]) trên một máy in để xem lịch sử in ấn của máy đó.
- Chọn "Cấu hình" ([Modify printer]) trên một máy in để có thể thay đổi thông tin, thay đổi trạng thái, xóa máy in.

Đối với hoạt động chọn "Thêm máy in" ([Add printer]), hệ thống hiển thị giao diện form "Thêm máy in" để SPSO điền các thông tin cho máy in mới (Display Information Form). Lúc này SPSO có thể điền tất cả các thông tin nhấn "Hoàn tất" (Fill in the form and submit). Hệ thống cập nhật thông tin máy mới và hiển thị danh sách máy in đã được cập nhật.

Đối với hoạt động chọn xem "Hiện tại" ([View printer's queue]), hệ thống hiển thị một cửa sổ modal chứa danh sách yêu cầu in đang chờ trên máy in.

Đối với hoạt động chọn "Lịch sử" ([View printer's log]), hệ thống hiển thị một cửa sổ modal chứa danh sách yêu cầu in trong quá khứ trên máy in. Lúc này SPSO có thể thực hiện hoạt động lọc theo khoảng thời gian (ngày đến ngày). Khi lọc thì modal chỉ hiển thị các thông tin phù hợp với nhu cầu truy vấn.

Đối với hoạt động chọn "Cấu hình" trên một máy in ([Modify printer]) trên một máy in, hệ thống hiển thị cửa sổ modal, trong cửa sổ có tab thông tin máy in và tab tùy chỉnh cho máy in. Lúc này SPSO có thể thực hiện các hoạt động:

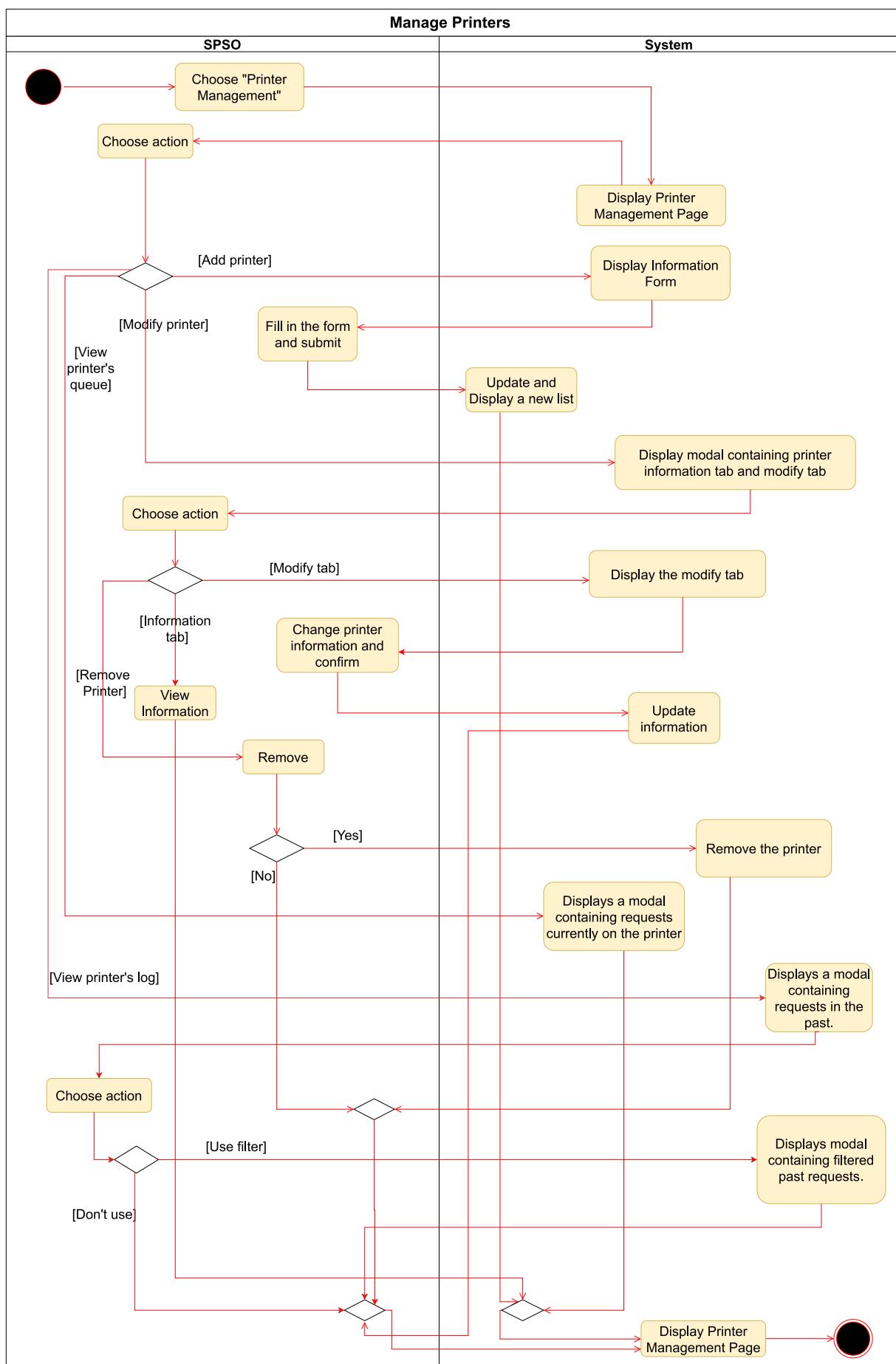


- Xem thông tin máy in (View Information)
- Chọn “Xóa máy in” ([Remove Printer])
- Chuyển sang tab “tùy chỉnh” ([Modify tab])

Dối với hoạt động xem thông tin máy in (View Information) thì SPSO vẫn tiếp tục ở tab thông tin máy in và không có thêm thao tác gì.

Dối với hoạt động chọn “Xóa máy in” ([Remove Printer]), khi SPSO chọn xóa máy in thì sẽ xuất hiện cửa sổ cảnh báo nhằm xác nhận quyết định này. Nếu được xác nhận ([Yes]), hệ thống xóa dữ liệu máy in. Nếu không được xác nhận ([No]), hệ thống không thực hiện thay đổi.

Dối với hoạt động chọn chuyển sang tab “tùy chỉnh” ([Modify tab]), hệ thống sẽ chuyển sang tab tùy chỉnh cho máy in. Trong tab tùy chỉnh cho máy in thì SPSO sẽ có thể thay đổi các thông tin máy in và trạng thái của nó. Sau khi thực hiện các thay đổi, SPSO ấn Hoàn tất để hệ thống lưu lại dữ liệu mới.

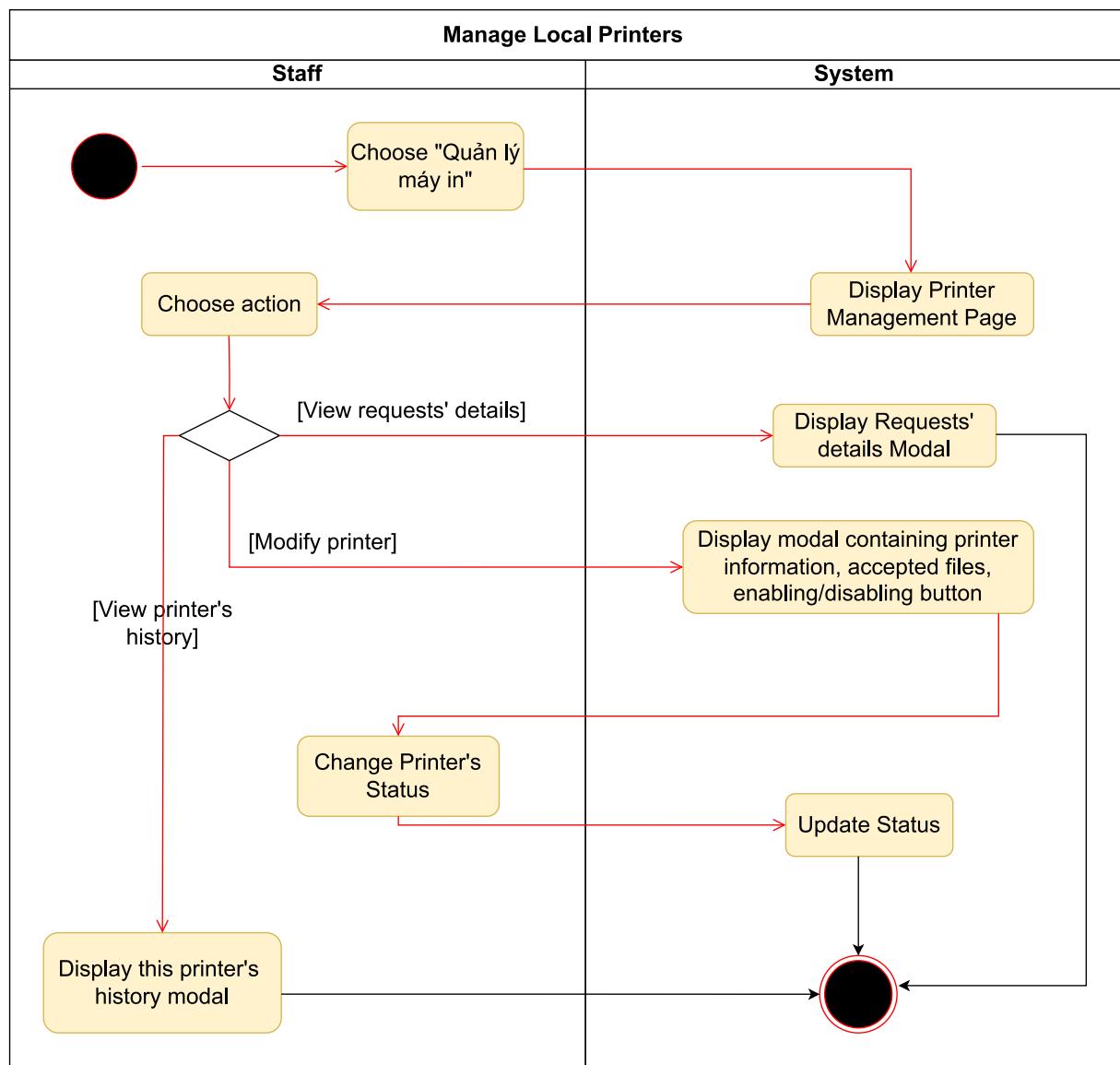


2.1.5 Chức năng Manage Local Printer

Mô tả: Biểu đồ này gồm 2 swimlane là: Staff, system. Ban đầu Staff chọn nút “Quản lý máy in” ở thanh bên. Hệ thống hiển thị thông tin của máy dưới dạng bảng gồm nhiều thuộc tính: ID máy in, Số yêu cầu in, Lịch sử in, Trạng thái, Thiết lập. Trừ ID máy in, trạng thái thì khi ấn vào các thuộc tính còn lại sẽ tiếp tục biểu đồ:

- Số yêu cầu in. Khi ấn vào, một modal hiện ra trình bày các yêu cầu in ấn hiện tại. Mỗi yêu cầu như vậy gồm các thông tin: tên sinh viên, mã số sinh viên, tên file, số bản in.
- Lịch sử in. Khi ấn vào, một modal hiện ra trình bày các yêu cầu in ấn đã hoàn thành. Mỗi yêu cầu như vậy gồm các thông tin: tên sinh viên, mã số sinh viên, tên file, số bản in, thời gian, lượng giấy in.
- Thiết lập. Khi ấn vào, modal "Cấu hình của máy in và tùy chỉnh" hiện ra. Trong này có thông tin máy in, tệp được phép, trạng thái. Tuy nhiên, với quyền hạn nhân viên, anh/cô ta chỉ được sửa trạng thái (hoạt động/vô hiệu hóa). Sau đó ấn "Hoàn tất" để hệ thống lưu lại.

Khi kết thúc mỗi chuỗi hoạt động trên thì biểu đồ hoạt động dừng lại.



2.2 Biểu đồ tuần tự (Sequence Diagram)

2.2.1 Cơ sở lý thuyết

Biểu đồ tuần tự

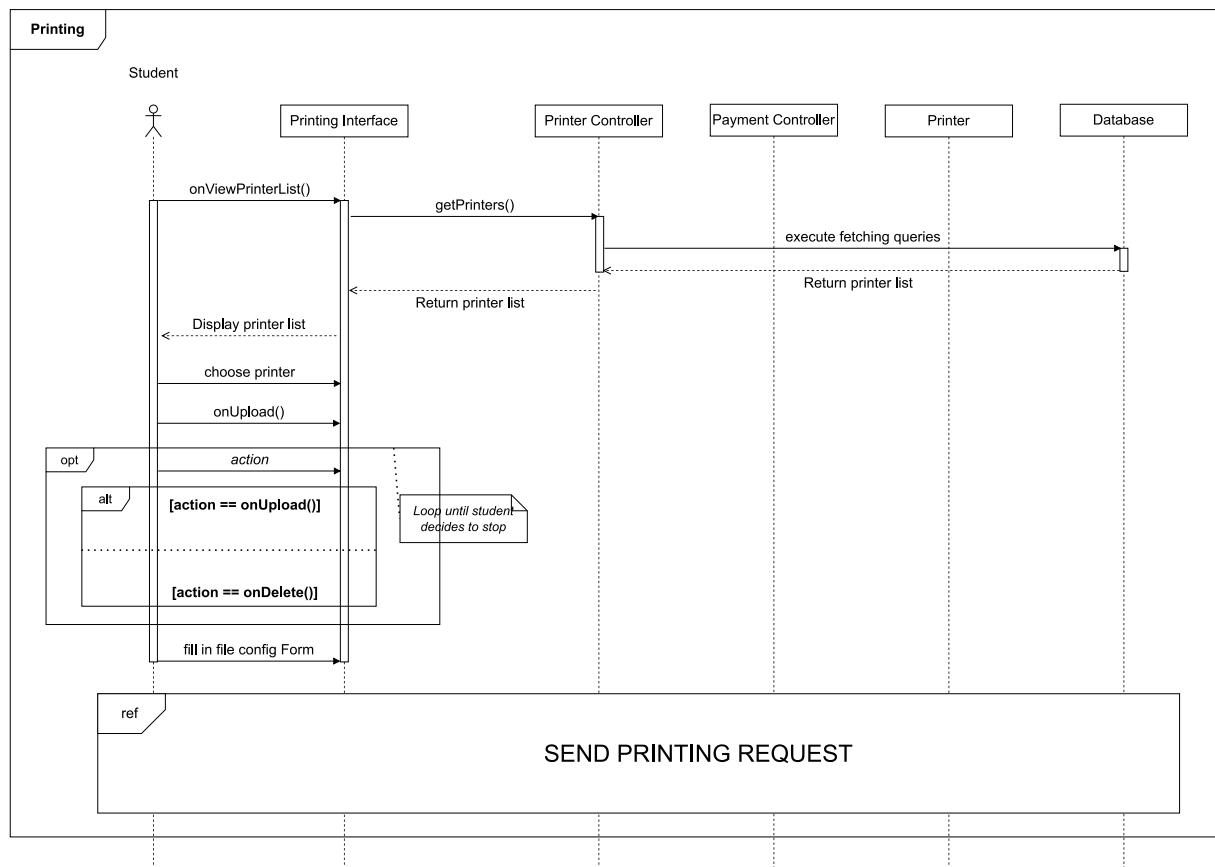
Biểu đồ tuần tự UML là biểu đồ tương tác trình bày chi tiết cách thực hiện các hoạt động. Chúng nắm bắt sự tương tác giữa các đối tượng trong bối cảnh hợp tác. Sơ đồ trình tự tập trung vào thời gian và chúng hiển thị thứ tự tương tác một cách trực quan bằng cách sử dụng trục đứng của sơ đồ để biểu thị thời gian những thông điệp nào được gửi và gửi khi nào. Biểu đồ tuần tự nắm bắt các vấn đề:

1. Sự tương tác diễn ra trong sự cộng tác nhằm hiện thực hóa một trường hợp sử dụng hoặc một hoạt động.
2. Tương tác cấp cao giữa người dùng hệ thống và hệ thống, giữa hệ thống và các hệ thống khác hoặc giữa các hệ thống con.

2.2.2 Chức năng Printing

Hai biểu đồ tuần tự của chức năng này, thầy / cô xem [tại đây](#) để có trải nghiệm tốt hơn.

Biểu đồ tuần tự: Printing



Mô tả: Biểu đồ tuần tự "Printing" mô tả quá trình tương tác giữa các thực thể sau:

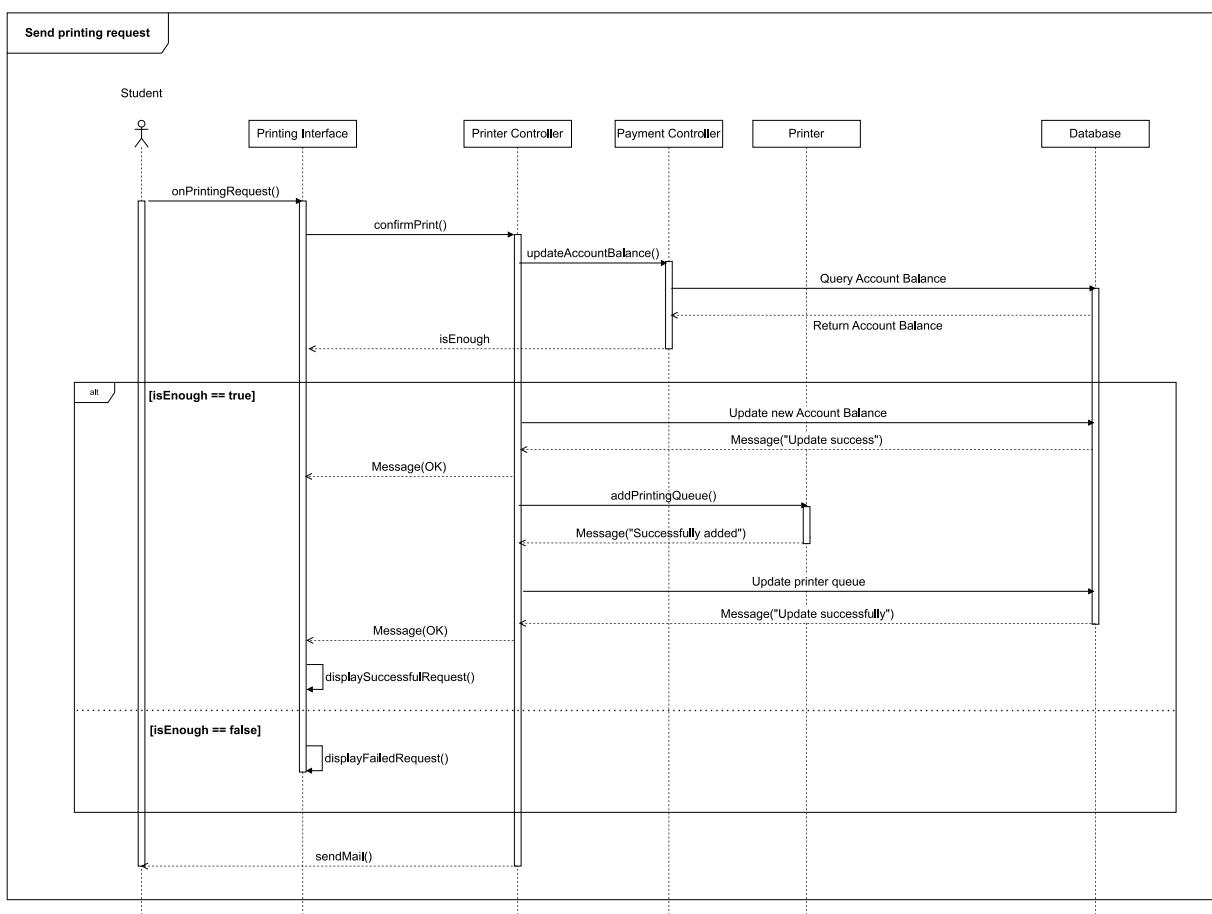
- Actor: Sinh viên.
- Printing Interface.
- Printer Controller.
- Payment Controller.

- Printer.
- Database.

Các bước trong quá trình tương tác như sau:

1. Sinh viên chọn "In tài liệu" trên thanh sidebar. (method onViewPrinterList() của Printing Interface được gọi).
2. Sinh viên chọn máy in.
3. Sinh viên tải file của mình (gọi method onUpload() của Printing Interface).
4. Sinh viên tiếp tục tải file lên (gọi tiếp onUpload())/ xóa file đã tải lên (gọi onDelete()). Lặp lại bước này cho đến khi sinh viên đã tải lên xong những file cần thiết của mình.
5. Sinh viên điền thông tin cấu hình cho các file in của mình.
6. Tham chiếu đến Biểu đồ tuần tự "Send Printing Request".

Biểu đồ tuần tự: Send Printing Request



Mô tả: Biểu đồ tuần tự "Send Printing Request" mô tả quá trình tương tác giữa các thực thể sau:

- Actor: Sinh viên.
- Printing Interface.
- Printer Controller.
- Payment Controller.
- Printer.



- Database.

Các bước trong quá trình tương tác như sau:

1. Sinh viên chọn nút "Gửi yêu cầu in". (SPSO gọi phương thức onPrintingRequest() của Printing Interface).
2. Printing Interface gọi method confirmPrint() của Printer Controller.
3. Printer Controller gọi method updateAccountBalance() để kiểm tra số dư tài khoản mới cho sinh viên.

Trường hợp số dư đủ:

- 4a. Method updateAccountBalance() đã gọi sẽ cập nhật số dư tài khoản mới cho sinh viên.
- 5a. Số dư tài khoản của sinh viên được cập nhật trong database.
- 6a. Printer controller gọi method addPrintingQueue() của Printer để thêm yêu cầu in vào hàng đợi, sau đó cập nhật hàng đợi trong database.
- 7a. Hệ thống hiển thị thông báo in thành công ra ngoài màn hình.

Trường hợp số dư không đủ:

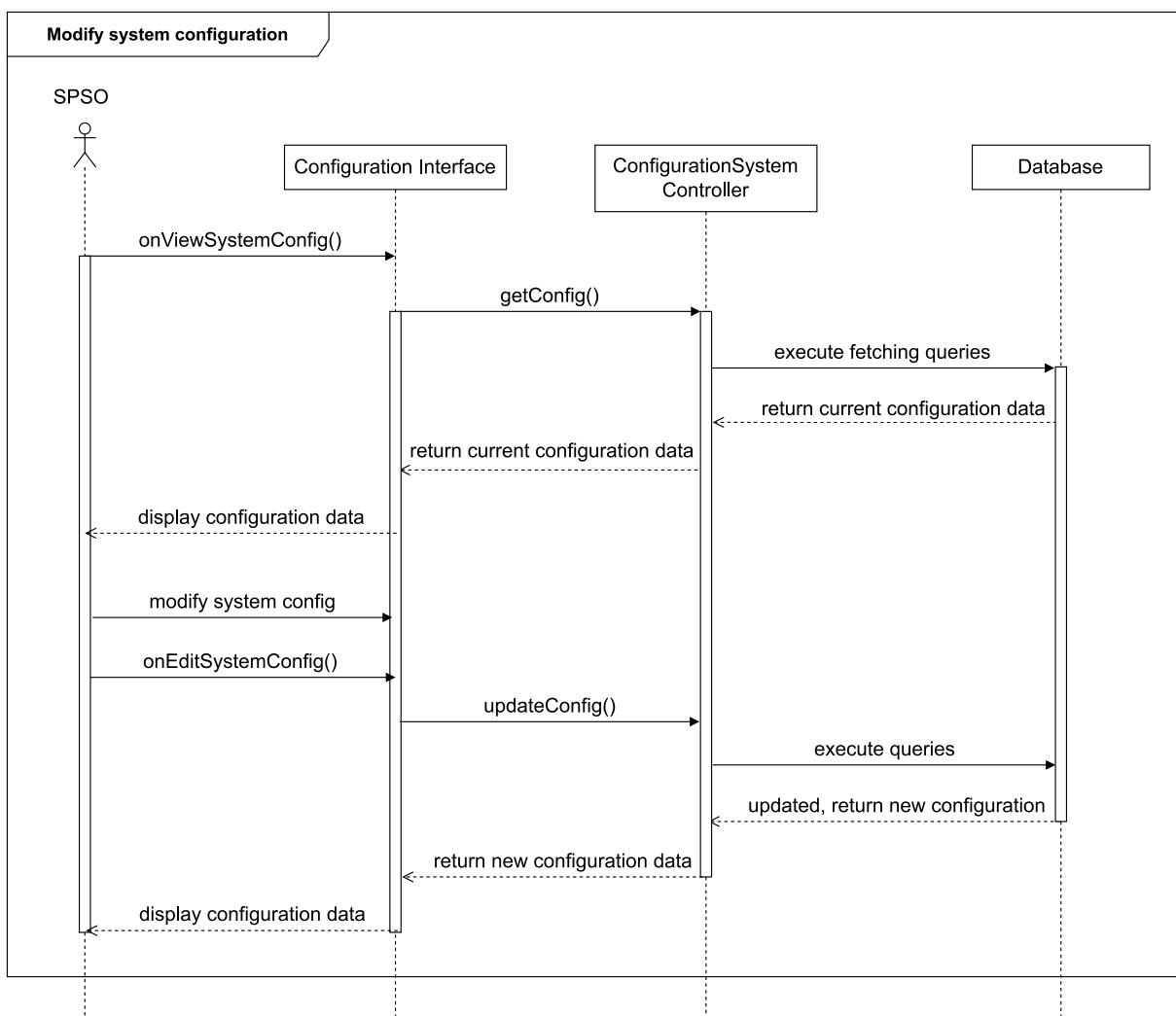
- 4b. Hệ thống hiển thị thông báo in thất bại do số dư tài khoản không đủ.

2.2.3 Chức năng Modify Printing System Configuration

Biểu đồ tuần tự: Modify Printing System Configuration

Mô tả: Biểu đồ này có sự tương tác giữa các thực thể: SPSO, Configuration Interface, ConfigurationSystem Controller, Database.

1. Ban đầu SPSO chọn nút “Cấu hình hệ thống” ở thanh bên (SPSO gọi onViewSystemConfig() của Configuration interface)
2. Configuration interface gọi phương thức getConfig() của ConfigurationSystem controller.
3. ConfigurationSystem controller sau đó truy vấn dữ liệu từ cơ sở dữ liệu.
4. Database trả về dữ liệu cho ConfigurationSystem controller. ConfigurationSystem Controller trả về dữ liệu cho Configuration Interface.
5. Configuration Interface hiển thị thông số hiện tại ra màn hình.
6. SPSO chỉnh sửa các thông số của hệ thống và ấn Áp dụng (gọi phương thức onEditSystemConfig() của Configuration interface).
7. Configuration interface gọi phương thức updateConfig() của ConfigurationSystem controller, thực hiện cập nhật dữ liệu từ database.
8. Database trả về dữ liệu cho ConfigurationSystem controller. Configuration Controller trả về dữ liệu cho Configuration Interface.
9. Configuration Interface hiển thị thông số mới ra màn hình.



2.2.4 Chức năng Manage Printer

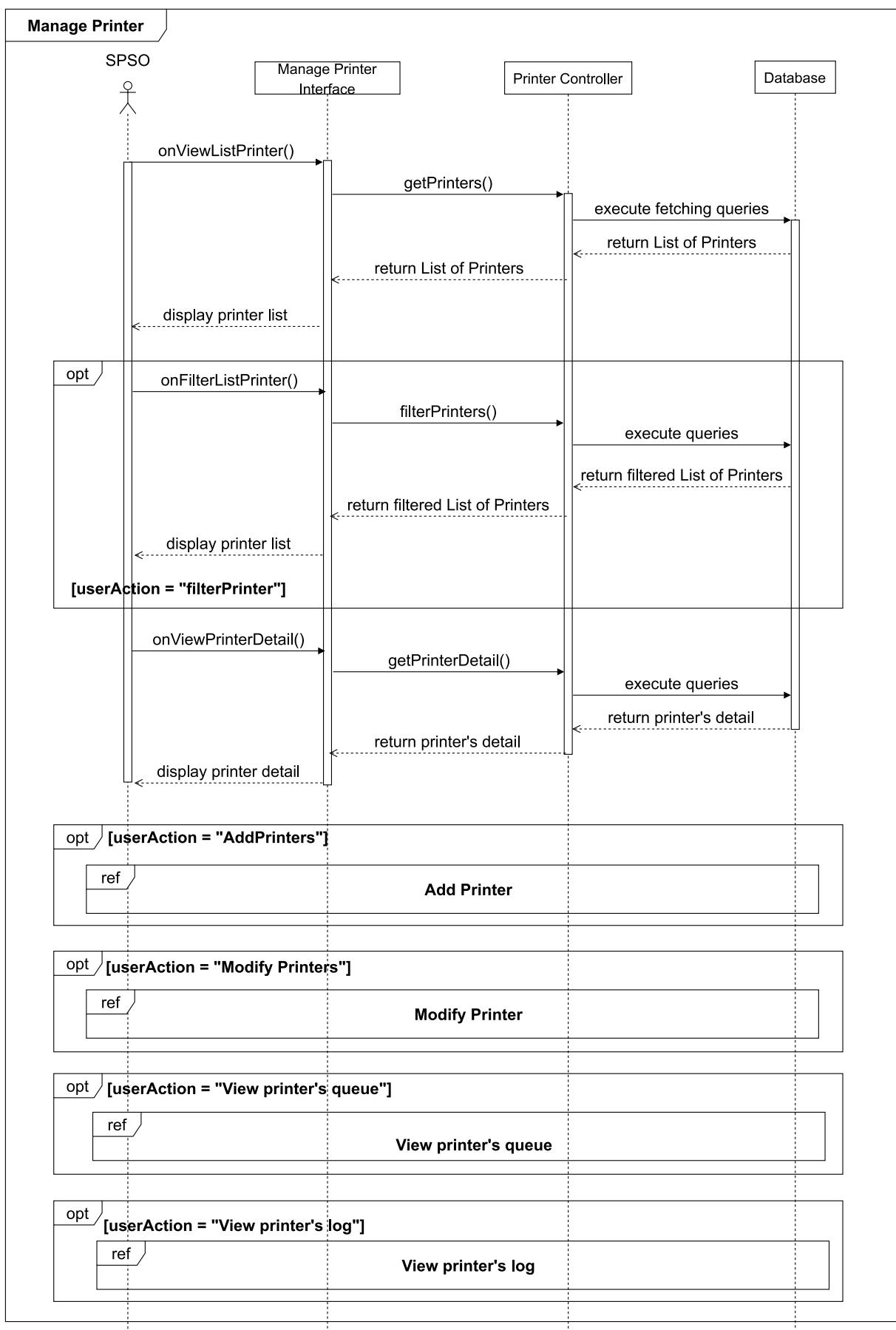
Biểu đồ tuần tự: Manage Printer

Biểu đồ tuần tự "Manage Printer" mô tả quá trình tương tác giữa các thực thể sau:

- Actor: SPSO
- Manage Printers Interface (MPI)
- Printer Controller
- Database

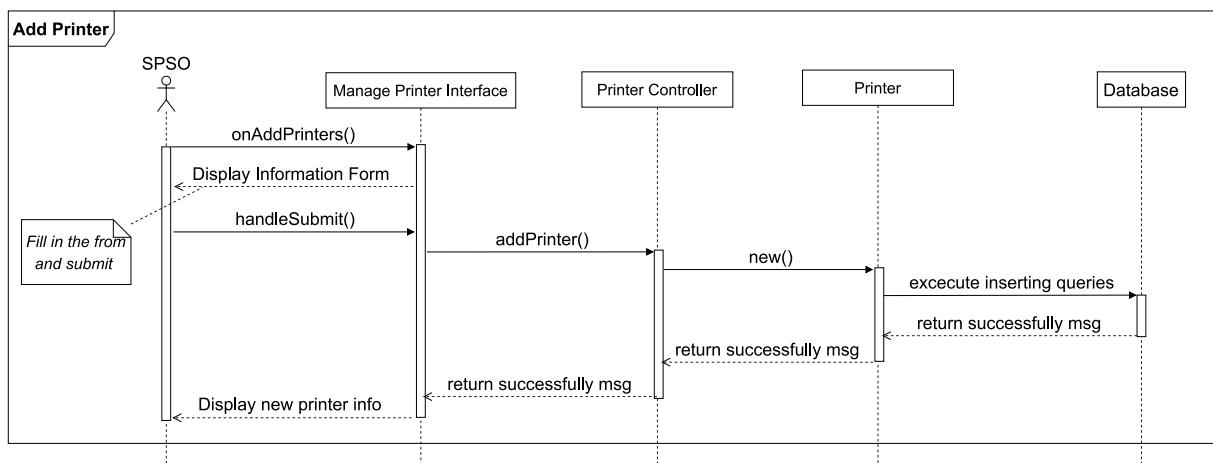
Mô tả các bước trong quá trình tương tác:

1. SPSO chọn nút "Hệ thống máy in" ở thanh bên (SPSO gọi phương thức `onViewListPrinter()` của MPI).
2. MPI tiếp tục gọi phương thức `getPrinters()` của Printer Controller.
3. Printer Controller sau đó truy vấn dữ liệu từ cơ sở dữ liệu.
4. Database trả về dữ liệu cho Printer Controller. Printer Controller trả về dữ liệu cho MPI.
5. MPI hiển thị danh sách máy in ra màn hình.
6. (Bước tùy chọn) SPSO sử dụng ô tìm kiếm / bộ lọc để tìm một / một số máy in mình muốn (SPSO gọi phương thức `onFilterListPrinter()` của MPI).



- MPI tiếp tục gọi phương thức filterPrinters() của Printer Controller.
 - Printer Controller sau đó truy vấn dữ liệu từ cơ sở dữ liệu.
 - Database trả về dữ liệu cho Printer Controller. Printer Controller trả về dữ liệu cho MPI.
 - MPI hiển thị danh sách máy in theo nhu cầu của SPSO ra màn hình.
- SPSO kích vào nút "Cấu hình" ở một máy in nào đó để xem (SPSO gọi phương thức onViewPrinterDetail() của MPI).
 - MPI tiếp tục gọi phương thức getPrinterDetail() của Printer Controller.
 - Printer Controller sau đó truy vấn dữ liệu từ cơ sở dữ liệu.
 - Database trả về dữ liệu về thông tin & trạng thái máy in cho Printer Controller. Printer Controller trả về dữ liệu cho MPI.
 - MPI hiển thị chi tiết về máy in ra màn hình cho SPSO.
 - (Bước tùy chọn) SPSO muốn thêm máy in. Chuyển đến biểu đồ tuần tự "Add Printer".
 - (Bước tùy chọn) SPSO xem hàng đợi máy in. Chuyển đến biểu đồ tuần tự "View Printer's Queue".
 - (Bước tùy chọn) SPSO sửa thông tin & trạng thái máy in. Chuyển đến biểu đồ tuần tự "Modify Printer".
 - (Bước tùy chọn) SPSO xem lịch sử của máy in. Chuyển đến biểu đồ tuần tự "View Printer's Log".

Biểu đồ tuần tự: Add Printer



Biểu đồ tuần tự "**Add Printer**" mô tả quá trình tương tác giữa các thực thể sau:

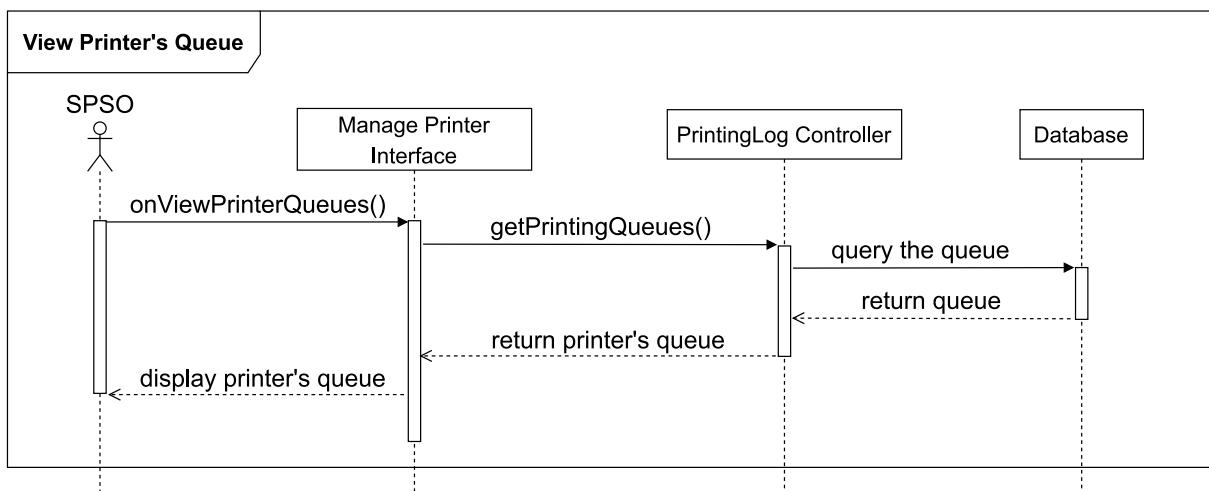
- Actor: SPSO
- Manage Printers Interface (MPI)
- Printer Controller
- Printer
- Database

Mô tả các bước trong quá trình tương tác:

- SPSO khởi tạo tương tác bằng cách ấn vào nút "Thêm máy in" trên giao diện (SPSO gọi phương thức onAddPrinters() của MPI).
- MPI nhận yêu cầu và hiển thị giao diện form để người dùng điền thông tin cho máy in mới.

3. SPSO sau khi điền đầy đủ thông tin trên giao diện và nhấn "Hoàn tất" (SPSO gọi phương thức handleSubmit() của MPI).
4. MPI nhận thông tin đã điền từ SPSO và gọi phương thức addPrinter() từ Printer Controller.
5. Printer Controller gọi phương thức new() của Printer để tạo đối tượng máy in mới.
6. Printer sau đó thực hiện các câu truy vấn để chèn dữ liệu mới vào cơ sở dữ liệu.
7. Database gửi thông điệp "Thành công" đến Printer. Sau đó, Printer gửi thông điệp "Thành công" đến Printer Controller. Printer Controller tiếp tục gửi thông điệp "Thành công" đến MPI.
8. MPI hiển thị thông tin máy in mới đến SPSO.

Biểu đồ tuần tự: View Printer's Queue



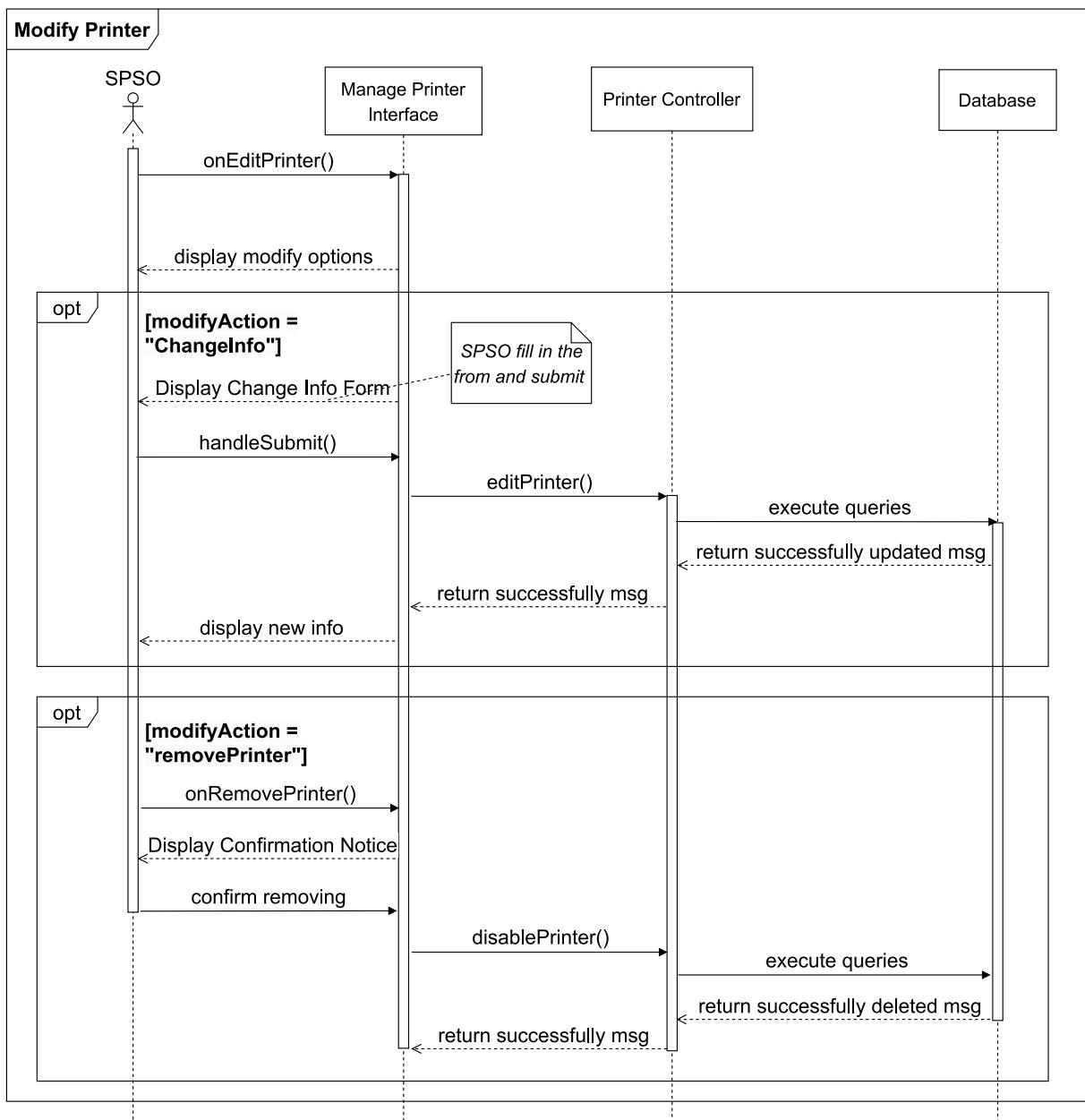
Biểu đồ tuần tự "View Printer's Queue" mô tả quá trình tương tác giữa các thực thể sau:

- Actor: SPSO
- Manage Printers Interface (MPI)
- PrintingLog Controller
- Database

Mô tả các bước trong quá trình tương tác:

1. SPSO bắt đầu tương tác bằng cách gửi ấn vào nút "Hiện tại" trên hàng giao diện của máy in tương ứng (SPSO gọi phương thức onViewPrinterQueue() của MPI).
2. MPI nhận yêu cầu và gửi yêu cầu lấy thông tin về các yêu cầu in đang chờ trên máy in đến PrintingLog Controller (MPI gọi phương thức getPrinterQueues() của PrintingLog Controller).
3. PrintingLog Controller thực hiện câu lệnh truy vấn đến cơ sở dữ liệu để lấy dữ liệu về hàng đợi in.
4. Dữ liệu về hàng đợi máy in sau khi được truy vấn từ Database được gửi về PrintingLog Controller.
5. PrintingLog Controller truyền tiếp dữ liệu này đến MPI.
6. MPI hiển thị dữ liệu hàng đợi máy in lên giao diện cho SPSO.

Biểu đồ tuần tự: Modify Printer



Biểu đồ tuần tự "**Modify Printer**" mô tả quá trình tương tác giữa các thực thể sau:

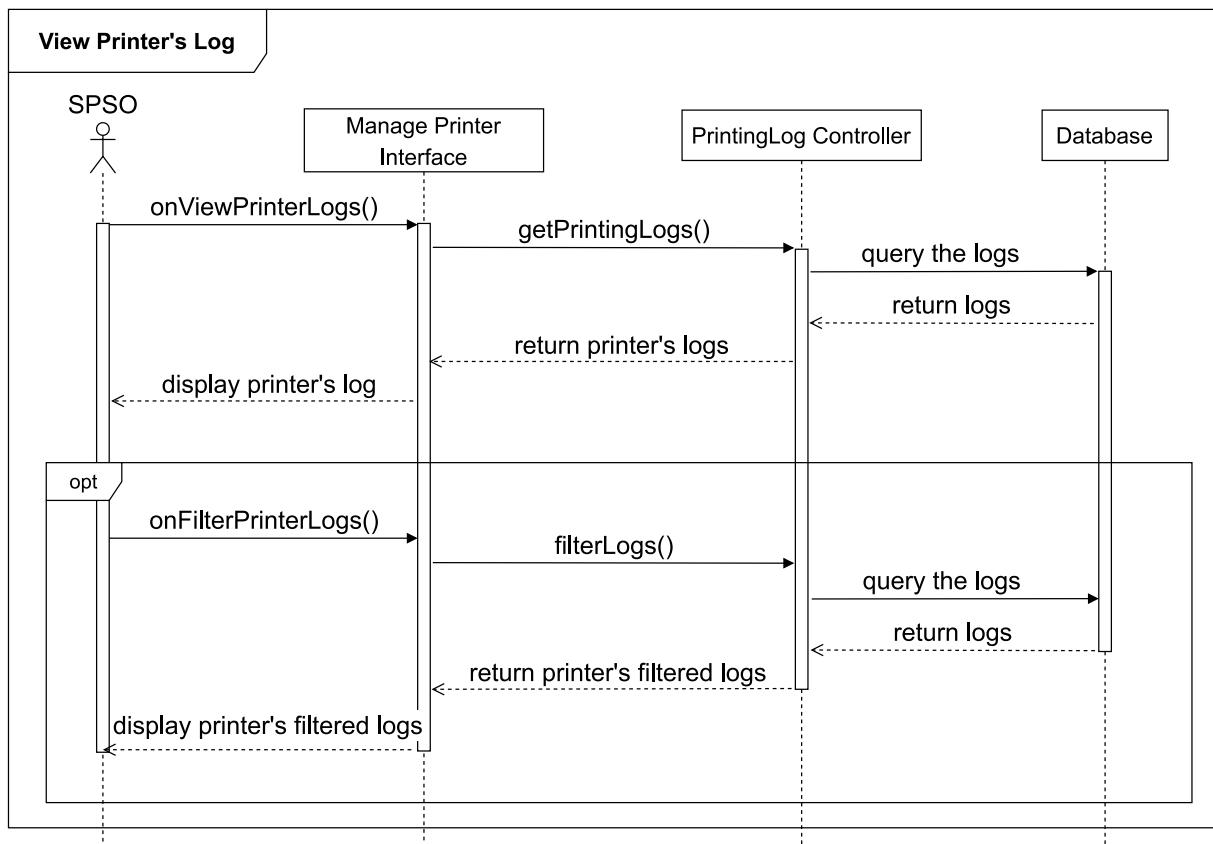
- Actor: SPSO
- Manage Printers Interface (MPI)
- Printer Controller
- Database

Các bước tương tác được mô tả như sau:

1. SPSO khởi tạo tương tác bằng cách ấn vào nút "Cáu hình" tương ứng với máy in. Cửa sổ hiện ra, chuyển sang tab tùy chỉnh máy in (SPSO gọi phương thức onEditPrinter()).
2. MPI hiển thị giao diện với các lựa chọn chỉnh sửa.
3. (Bước tùy chọn) SPSO thực hiện thay đổi các thông tin và trạng thái máy in.

- SPSO chuyển sang tab Tùy chỉnh, chỉnh sửa thông tin trong form giao diện và ấn Hoàn tất (SPSO gọi phương thức handleSubmit() của MPI).
 - MPI gọi phương thức editPrinter() của Printer Controller.
 - Printer Controller thực hiện các câu truy vấn chỉnh sửa tại cơ sở dữ liệu
 - Database gửi về thông điệp "Thành công" cho Printer Controller. Printer Controller gửi về thông điệp "Thành công" cho MPI.
 - MPI hiển thị thông tin và trạng thái mới của máy in.
4. (Bước tùy chọn) SPSO thực hiện xóa máy in.
- SPSO kích hoạt hành động bằng cách ấn nút "Xóa máy in" trên giao diện ở bước 2 (SPSO gọi phương thức removePrinter() của MPI).
 - MPI hiển thị cửa sổ cảnh báo rằng liệu SPSO có thực sự muốn xóa máy in.
 - SPSO xác nhận xóa.
 - MPI gọi hàm disablePrinter() của Printer Controller.
 - Printer Controller thực hiện các câu truy vấn xóa tại cơ sở dữ liệu.
 - Database gửi về thông điệp "Thành công" cho Printer Controller. Printer Controller gửi về thông điệp "Thành công" cho MPI.

Biểu đồ tuần tự: View Printer's Log



Biểu đồ tuần tự "View Printer's Log" mô tả quá trình tương tác giữa các thực thể sau:

- Actor: SPSO
- Manage Printers Interface (MPI)
- PrintingLog Controller



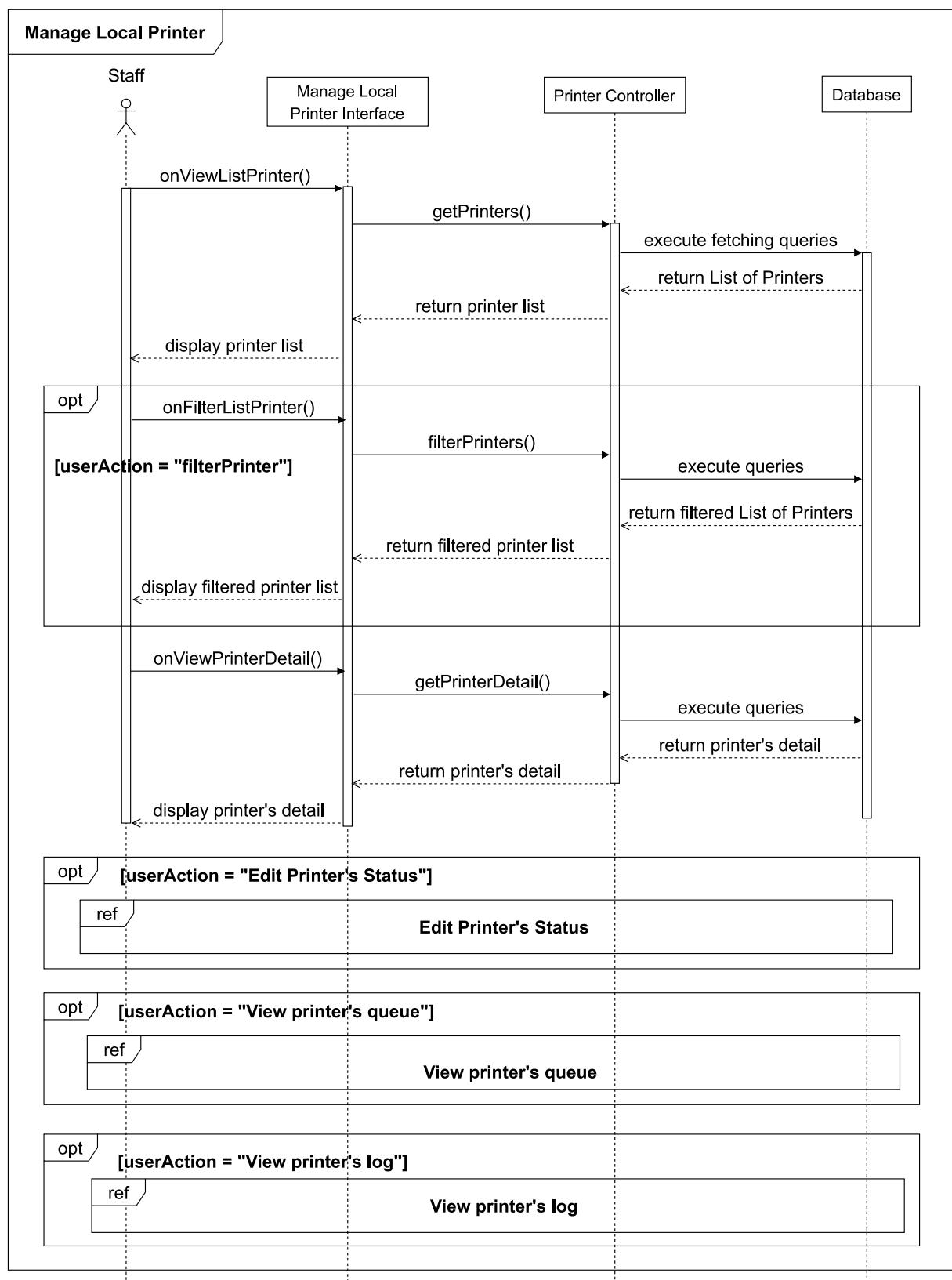
- Database

Mô tả các bước trong quá trình tương tác:

1. SPSO bắt đầu tương tác bằng cách gửi ấn vào nút "Lịch sử" trên hàng giao diện của máy in tương ứng (SPSO gọi phương thức onViewPrinterQueue() của MPI).
2. MPI nhận yêu cầu và gửi yêu cầu lấy thông tin về lịch sử trên máy in đến PrintingLog Controller (MPI gọi phương thức getPrintingLogs() của PrintingLog Controller).
3. PrintingLog Controller thực hiện câu lệnh truy vấn đến cơ sở dữ liệu để lấy dữ liệu về hàng đợi in.
4. Dữ liệu về lịch sử in ấn của máy in sau khi được truy vấn từ Database được gửi về PrintingLog Controller.
5. PrintingLog Controller truyền tiếp dữ liệu này đến MPI.
6. MPI hiển thị dữ liệu về lịch sử in ấn của máy in lên giao diện cho SPSO.
7. (Bước tùy chọn) SPSO muốn xem lịch sử của máy in trong một giai đoạn (ngày đến ngày)
 - SPSO khởi tạo tương tác bằng cách ấn vào biểu tượng Lọc trên giao diện ở bước 6.
 - MPI nhận yêu cầu và gửi yêu cầu lấy thông tin về lịch sử trên máy in đến PrintingLog Controller (MPI gọi phương thức getPrintingLogs() của PrintingLog Controller).
 - PrintingLog Controller thực hiện câu lệnh truy vấn đến cơ sở dữ liệu để lấy dữ liệu được yêu cầu.
 - Database gửi dữ liệu đã truy vấn được về PrintingLog Controller.
 - PrintingLog Controller truyền tiếp dữ liệu này đến MPI.
 - MPI hiển thị dữ liệu về lịch sử in ấn của máy in trong giai đoạn mà SPSO đã lọc lên giao diện.

2.2.5 Chức năng Manage Local Printer

Biểu đồ tuần tự: Manage Local Printer





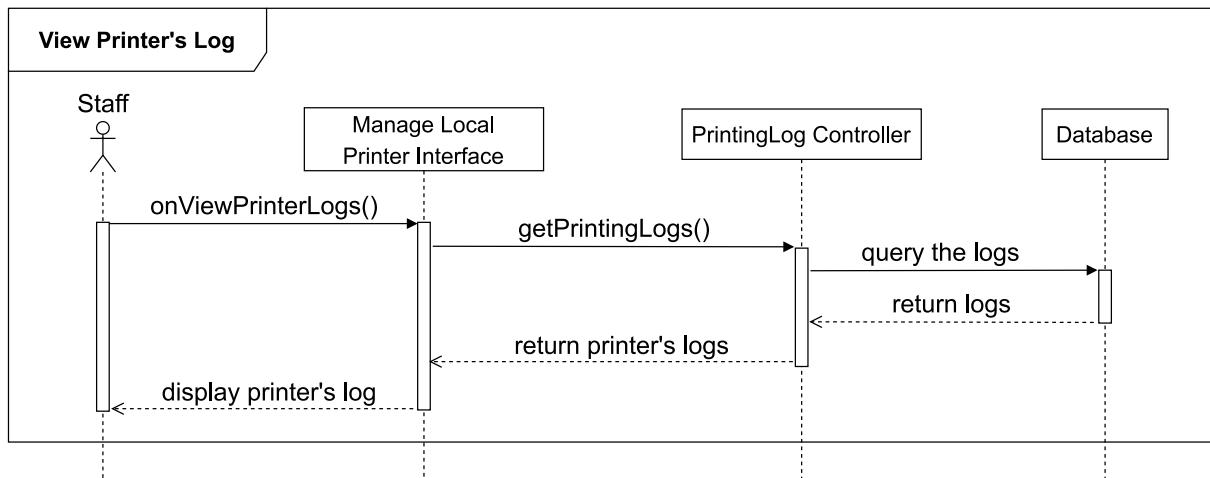
Biểu đồ tuần tự "Manage Local Printer" mô tả quá trình tương tác giữa các thực thể sau:

- Actor: Staff
- Manage Local Printers Interface (MLPI)
- Printer Controller
- Database

Mô tả các bước trong quá trình tương tác:

1. Staff chọn nút "Quản lý" ở thanh bên (SPSO gọi phương thức onViewListPrinter() của MLPI).
2. MLPI tiếp tục gọi phương thức getPrinters() của Printer Controller.
3. Printer Controller sau đó truy vấn dữ liệu từ cơ sở dữ liệu.
4. Database trả về dữ liệu cho Printer Controller. Printer Controller trả về dữ liệu cho MLPI.
5. MLPI hiển thị danh sách máy in ra màn hình.
6. (Bước tùy chọn) Staff sử dụng ô tìm kiếm để tìm một / một số máy in mình muốn (Staff gọi phương thức onFilterListPrinter() của MLPI).
 - MLPI gọi phương thức filterPrinters() của Printer Controller.
 - Printer Controller sau đó truy vấn dữ liệu từ cơ sở dữ liệu.
 - Database trả về dữ liệu cho Printer Controller. Printer Controller trả về dữ liệu cho MLPI.
 - MLPI hiển thị danh sách máy in theo nhu cầu của Staff ra màn hình.
7. Staff kích vào nút "Cấu hình" ở một máy in nào đó để xem (Staff gọi phương thức onViewPrinterDetail() của MPI).
8. MPI tiếp tục gọi phương thức getPrinterDetail() của Printer Controller.
9. Printer Controller sau đó truy vấn dữ liệu từ cơ sở dữ liệu.
10. Database trả về dữ liệu về thông tin & trạng thái máy in cho Printer Controller. Printer Controller trả về dữ liệu cho MLPI.
11. MLPI hiển thị chi tiết về máy in ra màn hình cho Staff.
12. (Bước tùy chọn) Staff xem hàng đợi máy in. Chuyển đến biểu đồ tuần tự "View Printer's Queue".
13. (Bước tùy chọn) Staff sửa trạng thái máy in. Chuyển đến biểu đồ tuần tự "Edit Printer's Status".
14. (Bước tùy chọn) Staff xem lịch sử của máy in. Chuyển đến biểu đồ tuần tự "View Printer's Log".

Biểu đồ tuần tự: View Printer's Log



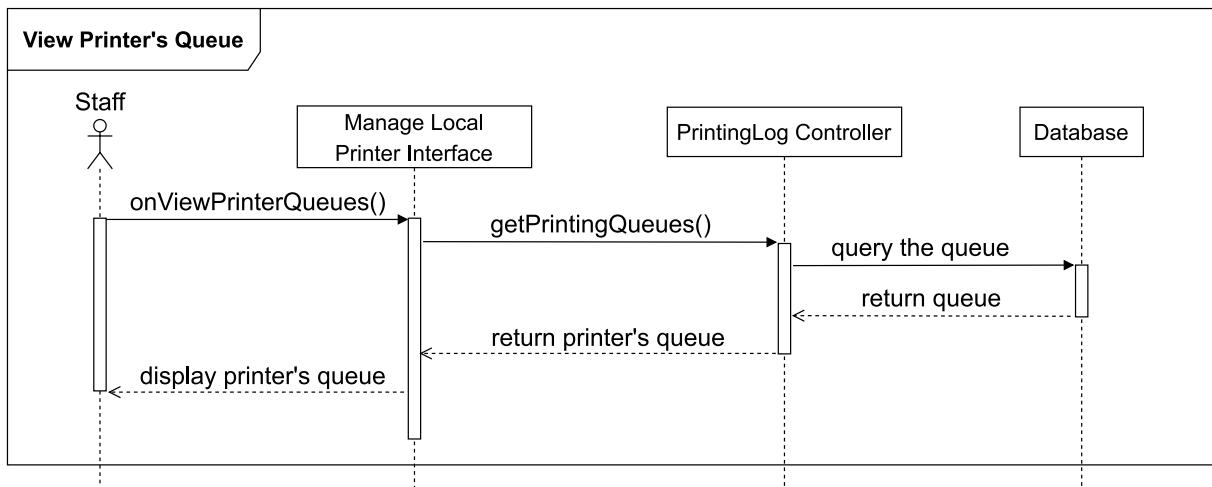
Biểu đồ tuần tự "View Printer's Log" mô tả quá trình tương tác giữa các thực thể sau:

- Actor: Staff
- Manage Printers Interface (MLPI)
- PrintingLog Controller
- Database

Mô tả các bước trong quá trình tương tác:

1. Staff bắt đầu tương tác bằng cách gửi yêu cầu vào nút "Lịch sử" trên hàng giao diện của máy in tương ứng (Staff gọi phương thức `onViewPrinterQueue()` của MLPI).
2. MLPI nhận yêu cầu và gửi yêu cầu lấy thông tin về lịch sử trên máy in đến PrintingLog Controller (MLPI gọi phương thức `getPrintingLogs()` của PrintingLog Controller).
3. PrintingLog Controller thực hiện câu lệnh truy vấn đến cơ sở dữ liệu để lấy dữ liệu về hàng đợi in.
4. Dữ liệu về lịch sử in ấn của máy in sau khi được truy vấn từ Database được gửi về PrintingLog Controller.
5. PrintingLog Controller truyền tiếp dữ liệu này đến MLPI.
6. MLPI hiển thị dữ liệu về lịch sử in ấn của máy in lên giao diện cho Staff.

Biểu đồ tuần tự: View Printer's Queue



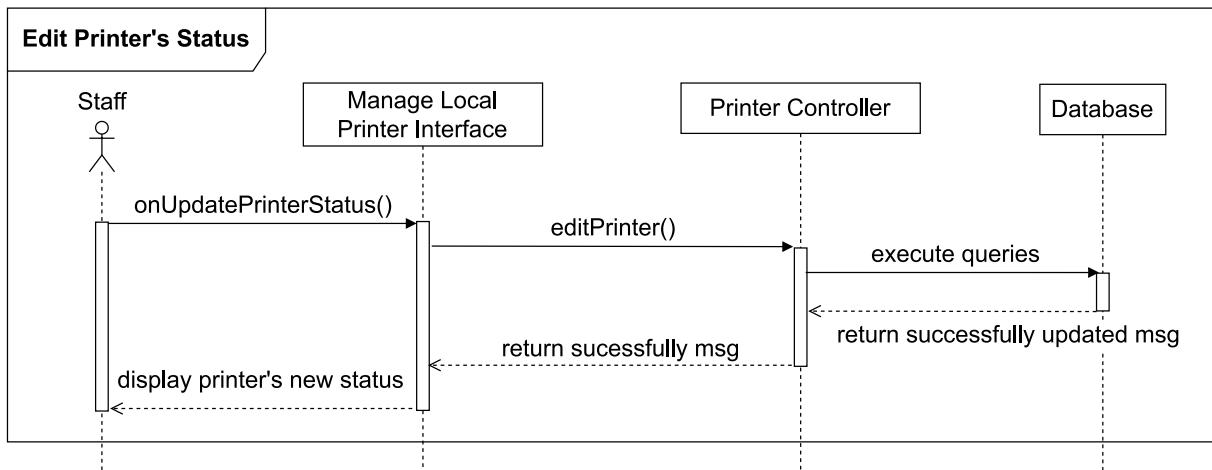
Biểu đồ tuần tự "View Printer's Queue" mô tả quá trình tương tác giữa các thực thể sau:

- Actor: Staff
- Manage Local Printer Interface (MLPI)
- PrintingLog Controller
- Database

Mô tả các bước trong quá trình tương tác:

1. Staff bắt đầu tương tác bằng cách gửi ấn vào nút "Hiện tại" trên hàng giao diện của máy in tương ứng (SPSO gọi phương thức `onViewPrinterQueue()` của MLPI).
2. MLPI nhận yêu cầu và gửi yêu cầu lấy thông tin về các yêu cầu in đang chờ trên máy in đến PrintingLog Controller (MLPI gọi phương thức `getPrinterQueue()` của PrintingLog Controller).
3. PrintingLog Controller thực hiện câu lệnh truy vấn đến cơ sở dữ liệu để lấy dữ liệu về hàng đợi in.
4. Dữ liệu về hàng đợi máy in sau khi được truy vấn từ Database được gửi về PrintingLog Controller.
5. PrintingLog Controller truyền tiếp dữ liệu này đến MLPI.
6. MLPI hiển thị dữ liệu hàng đợi máy in lên giao diện cho Staff.

Biểu đồ tuần tự: Edit Printer's Status



Biểu đồ tuần tự "Edit Printer's Status" mô tả quá trình tương tác giữa các thực thể sau:



- Actor: Staff
- Manage Local Printers Interface (MLPI)
- Printer Controller
- Database

Các bước tương tác được mô tả như sau:

1. Staff khởi tạo tương tác bằng cách ấn vào nút "Cấu hình" tương ứng với máy in, cửa sổ hiện ra (Staff gọi phương thức onUpdatePrinterStatus()).
2. Staff thực hiện thay đổi trạng thái máy in và ấn Hoàn tất (SPSO gọi phương thức handleSubmit() của MLPI).
3. MLPI gọi phương thức editPrinter() của Printer Controller.
4. Printer Controller thực hiện các câu truy vấn chỉnh sửa tại cơ sở dữ liệu.
5. Database gửi về thông điệp "Thành công" cho Printer Controller. Printer Controller gửi về thông điệp "Thành công" cho MLPI.
6. MLPI hiển thị trạng thái mới của máy in.



2.3 Biểu đồ lớp (Class Diagram)

2.3.1 Cơ sở lý thuyết

Class Diagram

Trong kỹ thuật phần mềm, một biểu đồ lớp (**Class Diagram**) trong Ngôn ngữ Mô hình Hợp nhất (UML) là một loại biểu đồ cấu trúc mô tả cấu trúc của hệ thống bằng cách hiển thị các lớp trong hệ thống, các thuộc tính của chúng, thao tác (hoặc phương thức), và mối quan hệ giữa các đối tượng.

- Mục đích của biểu đồ lớp là:
 1. Hiển thị cấu trúc tĩnh của các phân loại trong hệ thống.
 2. Biểu đồ cung cấp một ký hiệu cơ bản cho các biểu đồ cấu trúc khác do UML quy định.
 3. Hữu ích cho các nhà phát triển và các thành viên khác trong nhóm.
 4. Nhà phân tích kinh doanh có thể sử dụng biểu đồ lớp để mô hình hóa hệ thống từ góc độ kinh doanh.
- Một biểu đồ lớp UML được tạo thành từ:
 - Một tập hợp các lớp.
 - Một tập hợp các mối quan hệ giữa các lớp.

2.3.2 Mô tả chi tiết

Đối với biểu đồ này, chúng ta sẽ quan sát nó từ 2 góc độ:

- **Từ trên xuống dưới:** Ở góc độ này mang lại cái nhìn về lớp hệ thống. Có 5 lớp chính trong hệ thống:
 1. **Lớp Trình bày (Presentation layer):** Lớp trình bày, còn được biết đến là giao diện người dùng hoặc lớp UI, chịu trách nhiệm hiển thị dữ liệu và thông tin cho người dùng một cách dễ hiểu và tương tác. Nó có thể bao gồm các thành phần như biểu mẫu, menu, nút và các yếu tố hình ảnh khác cho phép người dùng tương tác với phần mềm. Trong hệ thống HCMUT_SSPPS, chúng tôi quyết định thiết kế ra một số giao diện người dùng cơ bản như:
 - Danh sách máy in (PrinterList View)
 - Cấu hình in ấn (PrintingConfig View)
 - Lịch sử in ấn của máy in (Printer's Logs View)
 - Lịch sử in ấn của sinh viên (Student History View)
 - Cấu hình hệ thống in ấn (SystemConfiguration View)
 - Báo cáo hệ thống (System Report View)
 - Hoạt động in ấn của sinh viên (Student Activities View)
 2. **Lớp Nghiệp vụ (Business layer):** Còn được biết đến là lớp miền hoặc lớp ứng dụng, đây là nơi logic nghiệp vụ cốt lõi của ứng dụng. Nó chứa các quy tắc và thuật toán của ứng dụng để xử lý và thao tác dữ liệu, thường bao gồm các thành phần như quy trình làm việc, thực thể nghiệp vụ và các hệ thống quy tắc.
 3. **Lớp Dịch vụ (Service layer):** Cung cấp một bộ giao diện hoặc API cấp cao để đóng gói logic nghiệp vụ và chức năng cốt lõi của ứng dụng. Nó làm nhiệm vụ làm cầu nối giữa lớp trình bày (Presentaton Layer) và lớp nghiệp vụ (Business Layer), cung cấp một giao diện định rõ có thể sử dụng được cả hai lớp.
 4. **Lớp Bền vững (Persistence layer):** Chịu trách nhiệm lưu trữ và truy xuất dữ liệu từ cơ sở dữ liệu hoặc hệ thống lưu trữ dữ liệu khác. Nó thực hiện các nhiệm vụ như tạo, đọc, cập nhật và xóa dữ liệu, thường bao gồm một framework ánh xạ đối tượng-quan hệ (ORM) để hỗ trợ ánh xạ giữa mô hình đối tượng của ứng dụng và mô hình quan hệ của cơ sở dữ liệu.



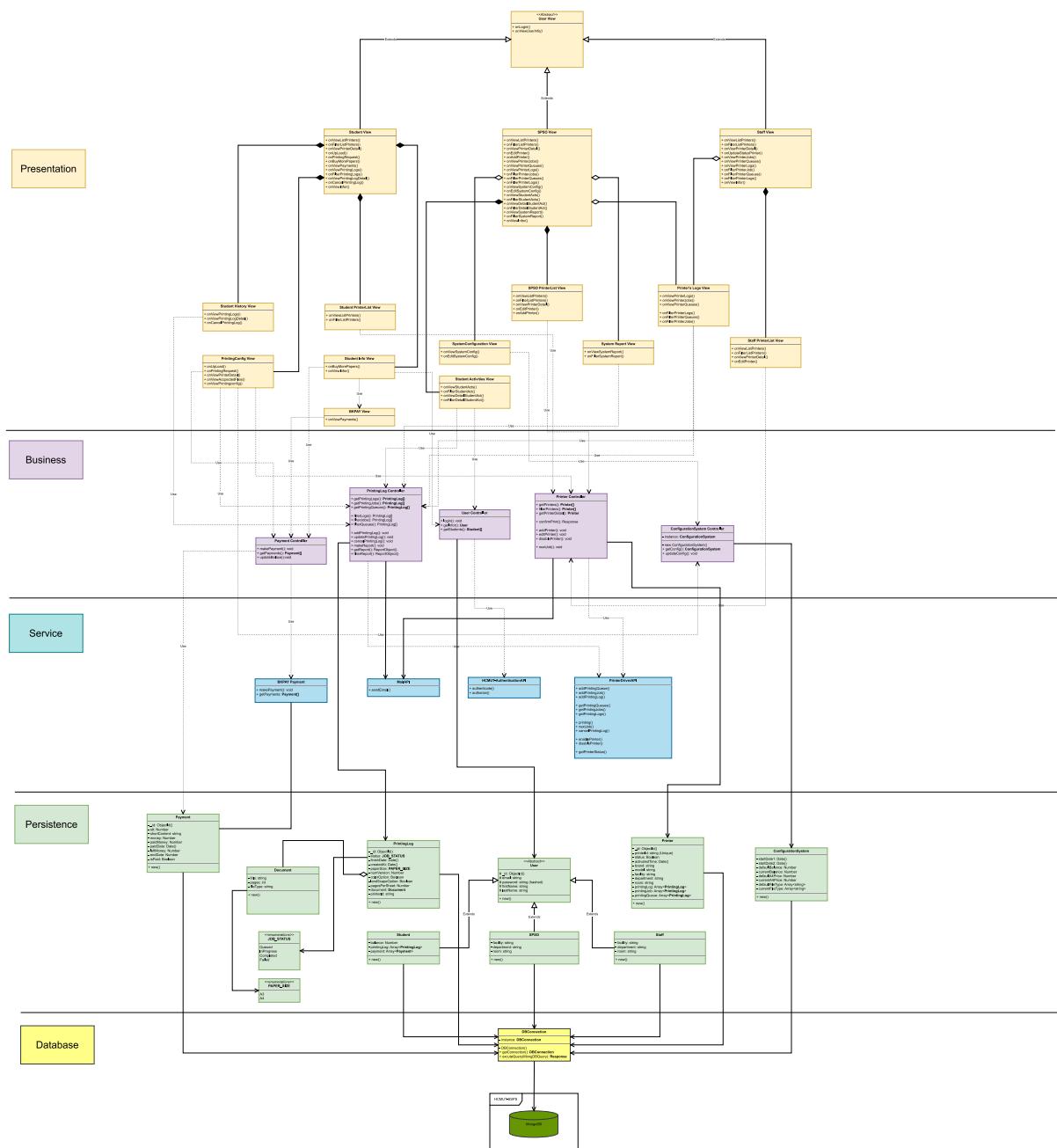
5. **Lớp Cơ sở dữ liệu (Database layer):** Chịu trách nhiệm quản lý nhu cầu lưu trữ và truy xuất dữ liệu của ứng dụng. Nó bao gồm các thành phần như hệ thống quản lý cơ sở dữ liệu (DBMS), lược đồ cơ sở dữ liệu và bất kỳ middleware hoặc trình điều khiển nào được sử dụng để truy cập cơ sở dữ liệu. Lớp cơ sở dữ liệu chịu trách nhiệm đảm bảo tính toàn vẹn dữ liệu, quản lý giao dịch, và cung cấp quyền truy cập dữ liệu hiệu quả và đáng tin cậy cho các lớp khác của ứng dụng. Lớp cơ sở dữ liệu có thể được xem như một phần của lớp bền vững, vì nó xử lý việc lưu trữ và truy xuất thực tế dữ liệu từ hệ thống cơ sở dữ liệu.

- **Từ trái sang phải:** Ở góc độ này vẽ ra một cái nhìn tổng quan về các mô-đun trong hệ thống.

1. **User Controller:** Quản lý tài khoản người dùng. Controller này sẽ xử lý các yêu cầu về thông tin cá nhân, lấy ra danh sách sinh viên (Chỉ có thông tin về tên và mã số sinh viên) để phục vụ cho việc theo dõi hoạt động in ấn của SPSO. Ngoài ra, controller này còn quản lý việc đăng nhập, quyền truy cập các tài nguyên trong hệ thống in ấn kết hợp với dịch vụ xác thực HCMUT_SSO.
2. **Printer Controller:** Controller này chịu trách nhiệm cho việc quản lý máy in. Cụ thể là quản lý việc truy vấn danh sách máy in, chỉnh sửa, kích hoạt, vô hiệu hóa máy in. Ngoài ra còn có thể thông qua Driver máy in để thực hiện các yêu cầu in ấn. Có liên hệ với PrintingLog Controller để lấy ra danh sách hàng đợi, lịch sử in của máy in.
3. **PrintingLog Controller:** Chịu trách nhiệm quản lý về các yêu cầu in ấn như: đưa vào hàng đợi, hủy yêu cầu in, sắp xếp hàng đợi tối ưu. Có thể xử lý yêu cầu về truy vấn thông tin chi tiết của một yêu cầu in (Đang đợi, Đang in, Đã in, Đã hủy) thông qua Driver của máy in. Ngoài ra còn có thể tạo ra các báo cáo hệ thống để phục vụ cho việc quan sát số liệu.
4. **ConfigurationSystem Controller:** Controller này chịu trách nhiệm về thiết lập các ràng buộc về in ấn như: loại file được cho phép in, số tờ mặc định cấp phát cho sinh viên theo từng học kỳ, giá một tờ in. Có thể giúp SPSO thiết lập cài đặt mặc định.
5. **Payment Controller:** Chịu trách nhiệm quản lý về các thông tin thanh toán của sinh viên thông qua dịch vụ thanh toán BKPAY và đồng thời cũng lưu trữ các thông tin thanh toán đó để đề phòng BKPAY bị mất dữ liệu thanh toán.
6. **APIs:** Bao gồm các dịch vụ trung gian ở Lớp Dịch vụ (Service Layer) như BKPAY Payment, HCMUT-AuthenticationAPI (HCMUT_SSO), MailAPI, PrinterDriverAPI.
7. **DBConnection:** Lớp này đóng vai trò trung gian giữa lớp bền vững (Persistence Layer) và lớp cơ sở dữ liệu (Database Layer). Nó nhận các yêu cầu từ các lớp ánh xạ và cung cấp các dịch vụ thích hợp để làm việc với cơ sở dữ liệu. Mục đích của hệ thống này là thiết lập một kết nối an toàn và hiệu quả giữa ứng dụng và cơ sở dữ liệu, cho phép ứng dụng truy xuất, lưu trữ và thao tác dữ liệu trong cơ sở dữ liệu.

2.3.3 Hình vẽ

Hình dưới đây là class diagram cho toàn bộ hệ thống HCMUT_SSPS. Do ảnh bị thu nhỏ khi chèn vào pdf, hãy / cô truy cập [tại đây](#) để xem bản rõ nét trên app.diagrams.net





2.4 MVP 1 với giao diện người dùng

Thầy/cô có thể xem bản đầy đủ cả ba biểu đồ [tại đây](#), vì ảnh khá lớn nên khi chèn vào file pdf sẽ không rõ.

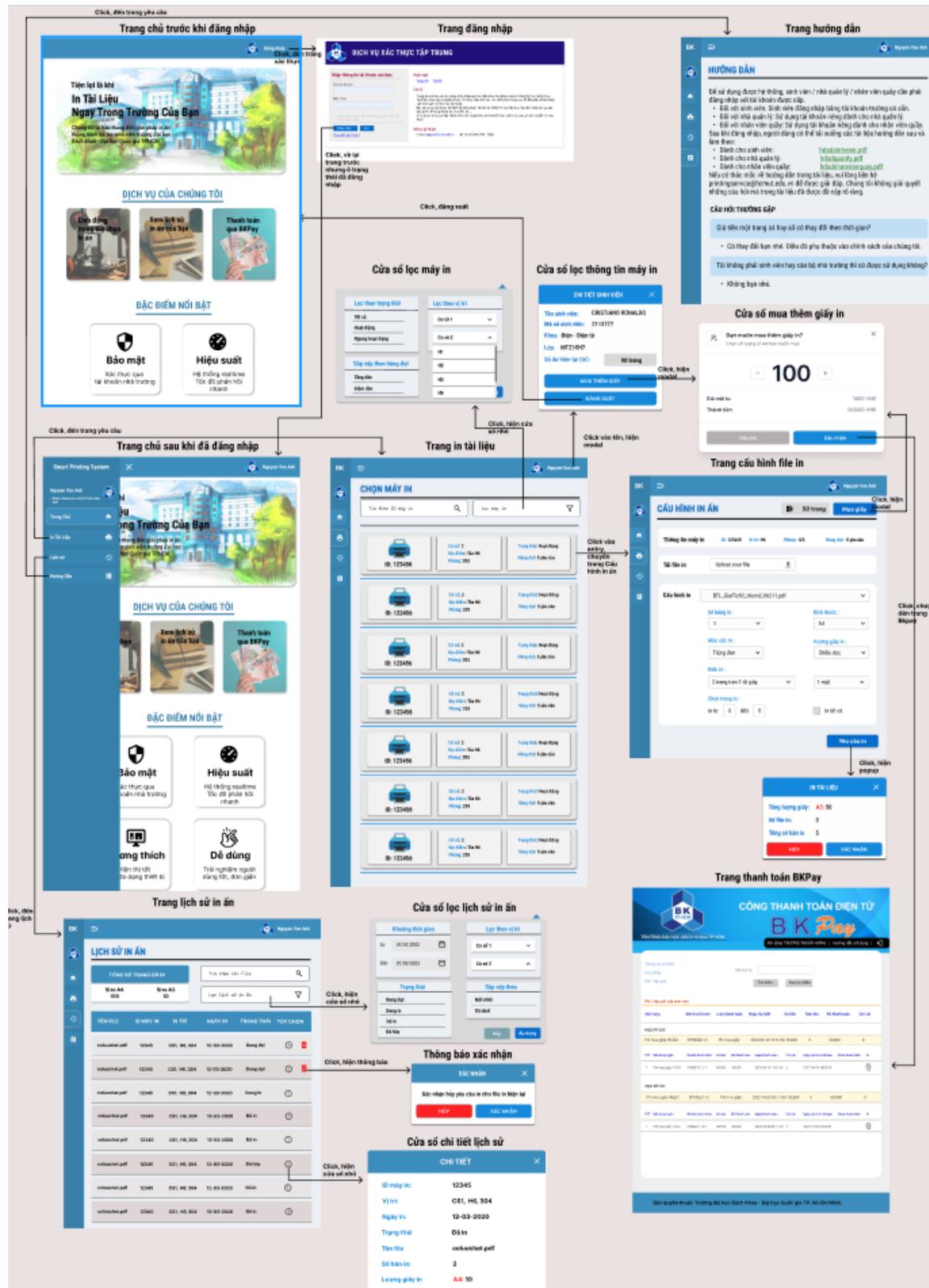
2.4.1 Sinh viên

Đối với sinh viên, để sử dụng được hệ thống SSPS, người dùng bắt buộc phải đăng nhập qua hệ thống HCMUT_SSO.

Sau khi đăng nhập thành công, giao diện trang chủ xuất hiện, sinh viên có thể nhìn thấy thanh sidebar ở bên trái màn hình, cung cấp các mục:

1. Trang chủ: Giao diện chính của trang web.
2. In tài liệu: Là chức năng quan trọng nhất của trang web đối với sinh viên. Tại đây, sinh viên có thể thực hiện các yêu cầu in ấn cũng như mua thêm trang in cho tài khoản của mình. Tính năng tự động chuyển hướng đến BKPay để thanh toán khi mua giấy cũng được hỗ trợ.
3. Lịch sử: Sinh viên có thể xem lịch sử các yêu cầu in ấn trước đó của mình, bao gồm các thông tin chi tiết về: File đã in, ID máy in, Vị trí máy in, Trạng thái in...
4. Hướng dẫn: Mục này cung cấp hướng dẫn sử dụng chi tiết về các tính năng của trang web in ấn. Sinh viên có thể tìm thấy thông tin về cách tải lên tài liệu, chọn tùy chọn in và quản lý các yêu cầu in ấn của mình bằng cách chọn mục này.

Các tính năng trên đã được mô tả cụ thể ở Thiết kế sau đây:



2.4.2 SPSO

Đối với SPSO, để sử dụng được hệ thống SSPS, người dùng bắt buộc phải đăng nhập qua hệ thống HCMUT_SSO.

Sau khi đăng nhập thành công, giao diện trang chủ xuất hiện, SPSO có thể nhìn thấy thanh sidebar ở bên trái màn hình, cung cấp các mục:



1. Trang chủ: Giao diện chính của trang web.
2. Hệ thống máy in: SPSO có thể quản lý danh sách bao gồm tất cả các máy in trong hệ thống, bao gồm các thông tin như: ID máy in, Vị trí, Trạng thái...
3. Cấu hình hệ thống: Là giao diện hỗ trợ SPSO trong việc thay đổi các cấu hình của hệ thống in ấn như: các định dạng file hợp lệ, số giấy miễn phí cho sinh viên mỗi học kỳ, giá tiền mỗi trang giấy khi mua thêm...
4. Hoạt động in ấn: Là giao diện hỗ trợ SPSO trong việc quản lý các hoạt động in ấn của sinh viên. Mục này chứa các thông tin về tất cả các lượt in bao gồm: ID và tên sinh viên, máy in thực hiện yêu cầu, thời gian in, file đã in...
5. Báo cáo: Là giao diện giúp SPSO xem lại các báo cáo về hệ thống bao gồm biểu đồ và các số liệu cụ thể, giúp SPSO quản lý hệ thống một cách bao quát, toàn diện hơn.
6. Hướng dẫn: Cung cấp hướng dẫn sử dụng chi tiết cho SPSO về hệ thống SSPS.

Các tính năng trên đã được mô tả cụ thể ở Thiết kế sau đây.

Click, đến trang yêu cầu

Trang chủ sau khi đã đăng nhập

Trang quản lý máy in của Staff

Click, hiện modal

Cửa sổ thông tin và tùy chỉnh máy in

Cửa sổ số yêu cầu in ấn hiện tại

Click, hiện modal

Cửa sổ số yêu cầu in ấn hiện tại

Cửa sổ lịch sử in ấn của máy in

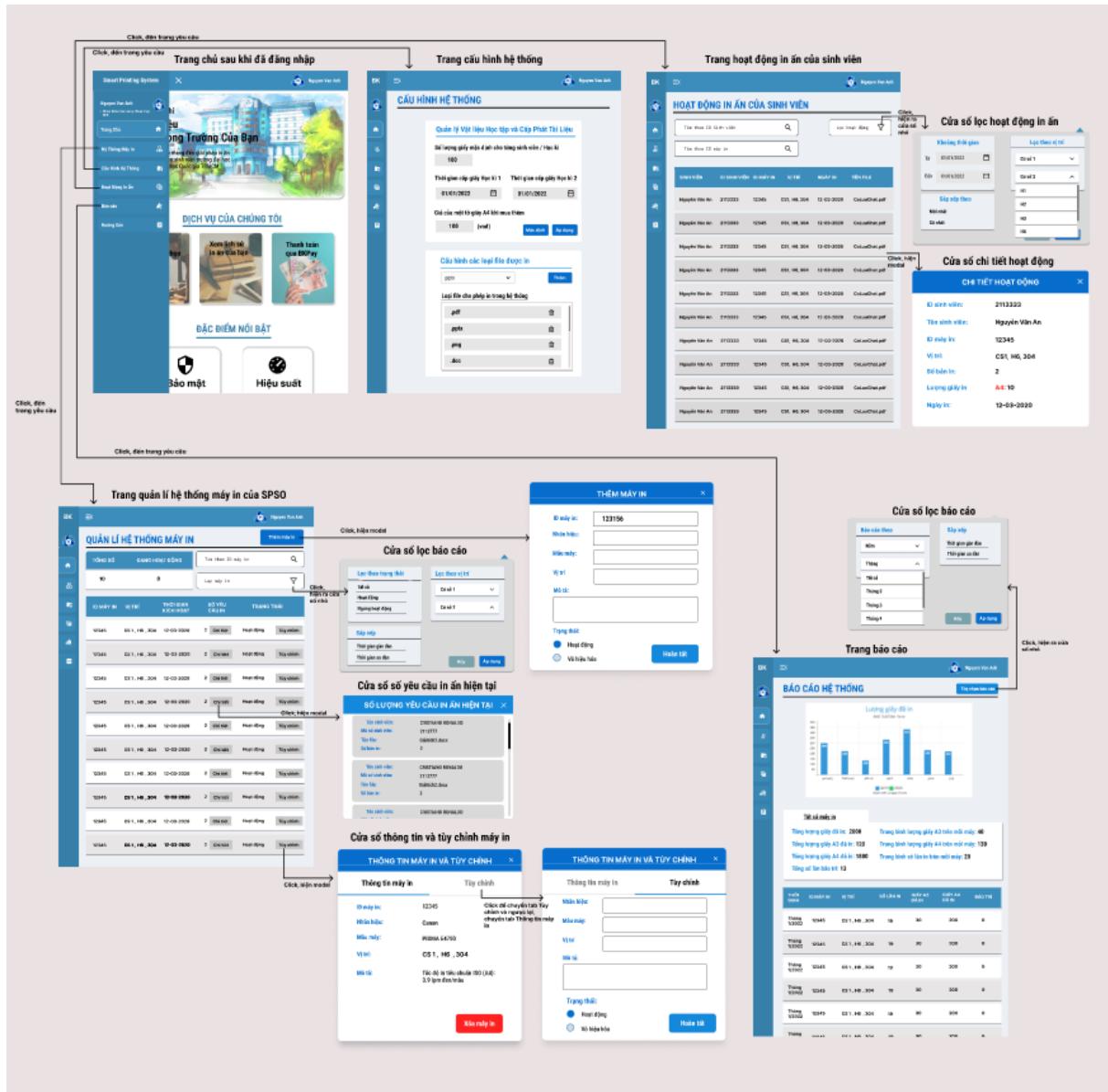
2.4.3 Staff

Dối với Staff, để sử dụng được hệ thống SSPS, người dùng bắt buộc phải đăng nhập qua hệ thống HCMUT_SSO.

Sau khi đăng nhập thành công, giao diện trang chủ xuất hiện, Staff có thể nhìn thấy thanh sidebar ở bên trái màn hình, cung cấp các mục:

1. Trang chủ: Giao diện chính của trang web.
2. Quản lý máy in: Tương tự đối với SPSO, tuy nhiên Staff chỉ được phép quản lý danh sách các máy in do mình phụ trách.
3. Hướng dẫn: Cung cấp hướng dẫn sử dụng chi tiết cho Staff về hệ thống SSPS.

Các tính năng trên đã được mô tả cụ thể ở Thiết kế sau đây.



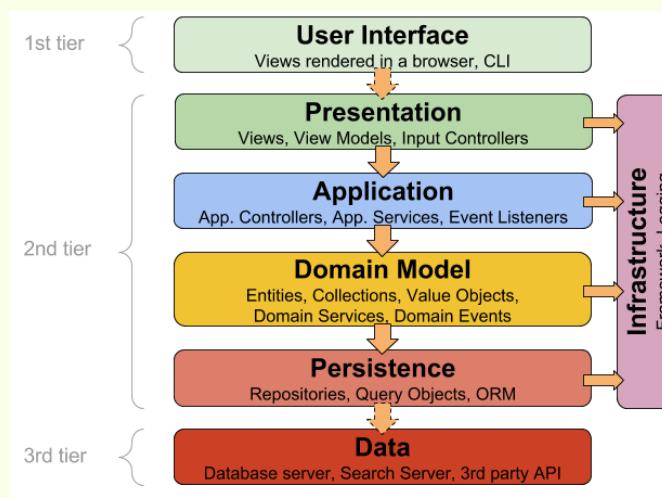
3 Thiết kế kiến trúc (Architecture design)

Kiến trúc lựa chọn

Layer Architecture

Kiến trúc lớp (**Layer Architecture**) được coi là kiến trúc phổ biến và rộng rãi sử dụng nhất trong phát triển phần mềm. Nó còn được biết đến là kiến trúc n-tier và mô tả một mô hình kiến trúc được tạo thành từ nhiều lớp ngang riêng biệt hoạt động cùng nhau như một đơn vị phần mềm duy nhất. Các thành phần có liên quan hoặc giống nhau thường được đặt trên cùng một lớp. Tuy nhiên, mỗi lớp khác nhau và đóng góp vào một phần khác nhau của hệ thống tổng thể. Số lượng lớp trong kiến trúc tầng không được đặt theo một số cụ thể và thường phụ thuộc vào nhà phát triển hoặc kiến trúc sư phần mềm. Quan trọng nhất là phải lưu ý rằng kiến trúc này thường luôn có một lớp tương tác người dùng, một lớp xử lý và một lớp xử lý dữ liệu.

- **Lớp Trình bày (Presentation Layer):** Chịu trách nhiệm cho tương tác của người dùng với hệ thống phần mềm.
- **Lớp Ứng dụng/Nghiệp vụ (Application/Business Layer):** Xử lý các khía cạnh liên quan đến việc đạt được các yêu cầu chức năng.
- **Lớp Miền (Domain Layer):** Chịu trách nhiệm về thuật toán và các thành phần lập trình.
- **Lớp Cơ sở hạ tầng/Bền vững/Cơ sở dữ liệu (Infrastructure/Persistence/-Database Layer):** Chịu trách nhiệm xử lý dữ liệu và cơ sở dữ liệu.



Hệ thống HCMUT_SSPS sẽ bao gồm 5 lớp. Mỗi lớp sẽ chứa một số mô-đun hỗ trợ chức năng của trang web. Mô tả chi tiết hơn về kiến trúc trên có thể được nhấn mạnh như sau:

1. **Lớp Cơ sở dữ liệu (Database Layer):** Lưu trữ dữ liệu và một lớp cơ bản để thiết lập kết nối giữa cơ sở dữ liệu và lớp bền vững. Do đó, nó không có mô-đun.
2. **Lớp Bền vững (Persistence Layer):** Chứa mô-đun Object Relational Mapping (ORM), cũng là một kỹ thuật được sử dụng để tạo ra "cầu nối" giữa chương trình hướng đối tượng và, trong hầu hết các trường hợp, cơ sở dữ liệu quan hệ. Nói một cách khác, chúng ta có thể xem ORM như là lớp kết nối giữa Lập trình Hướng đối tượng (OOP) và cơ sở dữ liệu quan hệ.
3. **Lớp Dịch vụ (Service Layer):** Lớp này sẽ đảm nhận việc bên hóa từ Lớp Nghiệp vụ sang các dịch vụ tương ứng.
4. **Lớp Nghiệp vụ (Business Layer):** Chứa 5 mô-đun chính:



- **Mô-đun quản lý máy in (Manage Printer Module):** Quản lý các máy in trong hệ thống dưới sự giám sát của SPSO và Staff
 - **Mô-đun in ấn (Printing Module):** Quản lý việc thực hiện các yêu cầu in ấn của sinh viên.
 - **Mô-đun nhật ký in ấn (PrintingLog Module):** Quản lý các lịch sử in ấn của sinh viên, tạo ra các báo cáo về hệ thống.
 - **Mô-đun Cấu hình hệ thống (System Configuration Module):** Chịu trách nhiệm về quản lý các thiết lập về hệ thống in ấn.
 - **Mô-đun Người dùng (User Module):** Quản lý các thông tin cá nhân của người dùng, xác nhận thông tin đăng nhập, quyền truy cập vào các tài nguyên của hệ thống in ấn.
5. **Lớp Giao diện Người dùng (User Interface Layer):** Sử dụng cho quá trình tương tác giữa người dùng và hệ thống, chứa một mô-đun xem (View Module).

Nguyên nhân lựa chọn kiến trúc

1. **Tính Tổ chức:** Kiến trúc lớp giúp tổ chức mã nguồn theo các tầng chức năng khác nhau, giúp dễ dàng theo dõi và quản lý.
2. **Tính Tính nhất quán:** Kiến trúc lớp giúp duy trì tính nhất quán trong mã nguồn, với các chức năng và trách nhiệm phân phối rõ ràng.
3. **Dễ Dàng Bảo trì và Mở rộng:** Việc chia thành các tầng riêng biệt giúp việc bảo trì trở nên thuận tiện hơn và mở rộng hệ thống trở nên linh hoạt hơn.
4. **Tính Tái sử dụng:** Các thành phần của từng tầng có thể được tái sử dụng trong các dự án khác, giảm công sức và tăng hiệu suất phát triển.
5. **Bảo mật và Quản lý Truy cập:** Việc tách biệt các lớp giúp kiểm soát quyền truy cập và bảo mật dữ liệu, đặc biệt là quan trọng trong hệ thống chứa thông tin quan trọng.
6. **Phân chia Công việc:** Các nhóm phát triển có thể chịu trách nhiệm cho từng tầng cụ thể, giúp phân chia công việc một cách hiệu quả.

3.1 Biểu đồ Deployment

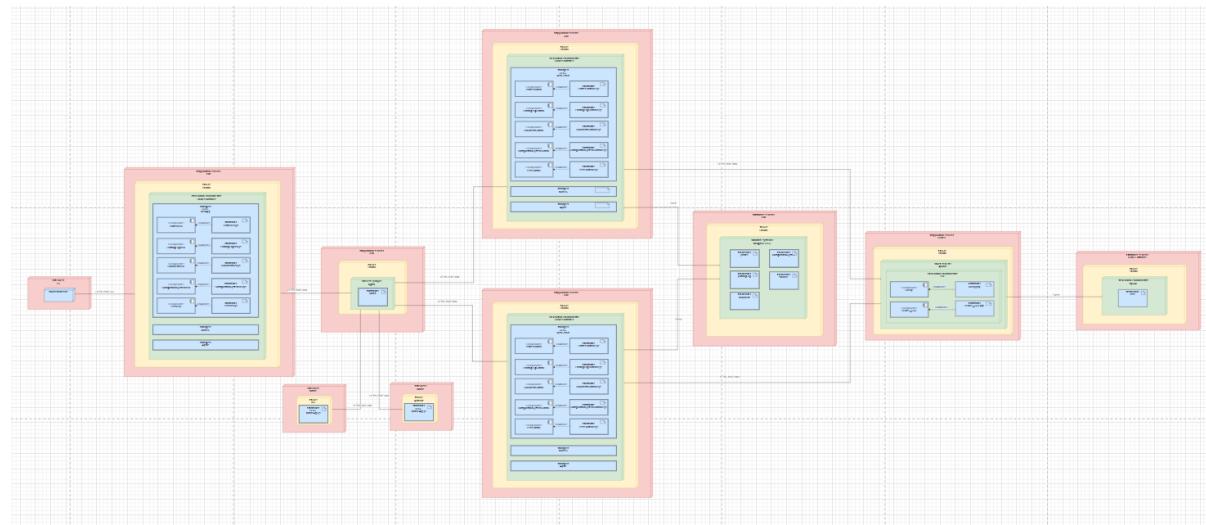
Cơ sở lý thuyết

Deployment Diagram

Trong UML, biểu đồ Deployment mô hình hóa kiến trúc vật lý của hệ thống. biểu đồ Deployment hiển thị mối quan hệ giữa các thành phần phần mềm và phần cứng trong hệ thống và sự phân bổ vật lý của quá trình xử lý. Các biểu đồ Deployment mà bạn thường chuẩn bị trong giai đoạn triển khai quá trình phát triển giúp hiển thị sự sắp xếp vật lý của các nodes trong hệ thống phân tán, các artifact được lưu trữ trên mỗi node cũng như các components và phần tử khác mà các artifact đó triển khai. Các nodes đại diện cho các thiết bị phần cứng như máy tính, cảm biến và máy in cũng như các thiết bị khác hỗ trợ môi trường thời gian chạy của hệ thống. Đường dẫn giao tiếp và mối quan hệ triển khai giúp mô hình hóa các kết nối trong hệ thống.

Biểu đồ

Thầy/cô có thể xem bản đầy đủ của cả ba biểu đồ [tại đây](#), vì ảnh khá lớn nên khi chèn vào file pdf sẽ không rõ.



Tổng thể biểu đồ Deployment cho hệ thống HCMUT_SSPS

- Web Browser: tượng trưng cho việc sử dụng browser của người dùng như Chrome, Firefox, Safari,... để kết nối tới trang web SSPS.
- Reverse Proxy: đóng vai trò quan trọng trong việc tăng cường bảo mật và caching. Đặc biệt, chức năng load balancing giúp phân chia tải đều các yêu cầu sang các máy chủ khác nhau, đồng thời giảm nguy cơ sự cố khi một máy chủ gặp vấn đề (one point failure).
- SSPS-WebApp image: tệp mã nguồn phía client mà trình duyệt của người dùng sẽ tải xuống khi họ truy cập trang web SSPS. Tệp này đã được đóng gói thành image sử dụng trong docker container
- SSPS-MobileApp.js: tệp mã nguồn mà client sẽ tải trên Appstore hay Google Play về thiết bị di động của mình và dùng nó để truy cập vào hệ thống SSPS.
- SSPS-WebService image: tệp này chạy ở phía server và chịu trách nhiệm xử lý các yêu cầu được gửi đến server. Khi server nhận được một request, tệp mã nguồn này sẽ xử lý request đó và trả về response tương ứng. Ở hình trên ta sẽ có 2 server để phân tải hiệu quả. Tệp này đã được đóng gói thành image sử dụng trong docker container
- MongoDB-SSPS: là hệ thống cơ sở dữ liệu sử dụng DBMS là MongoDB giúp ta scale database server dễ dàng, đơn giản và tiết kiệm hơn (horizontally scaling), đồng thời giai đoạn develop sẽ được tiết kiệm thời gian. Hệ thống chứa các dữ liệu về printer, printing log, valid file, document, configuration system.
- HCMUT-WebService: là service bên phía nhà trường HCMUT có một số tính năng như đăng nhập HCMUT_SSO và thanh toán BKPay. HCMUT cung cấp cho hệ thống SSPS các API liên quan để thực hiện
- HCMUT-Database: là hệ thống cơ sở dữ liệu sử dụng thuộc sở hữu của HCMUT. Hệ thống SSPS có thể thông qua API được cung cấp mà truy xuất dữ liệu của các users.

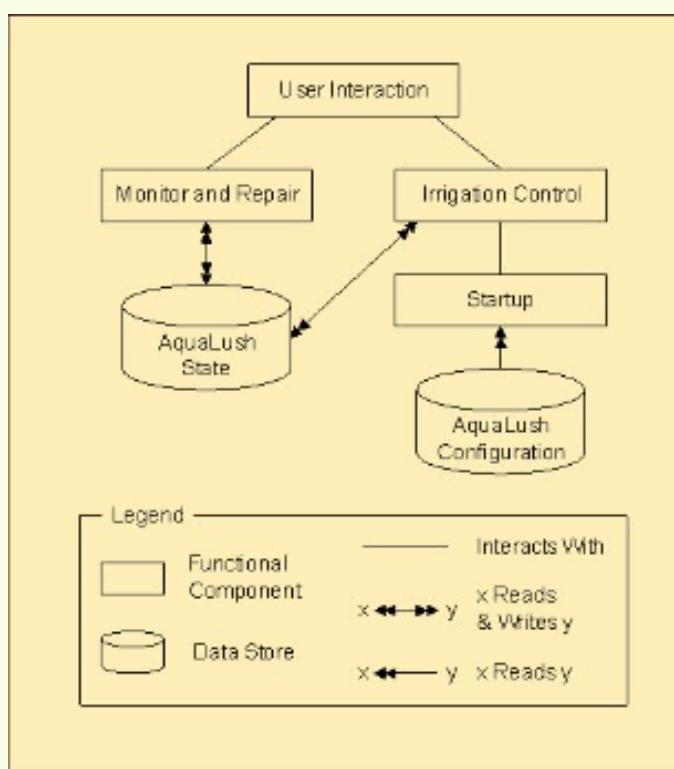
3.2 Biểu đồ Box - Line

Chúng ta sẽ mô tả rõ hơn về những thành phần trong kiến trúc lớp (Layer Architecture) của hệ thống HCMUT_SSPS thông qua Biểu đồ Box-Line

Cơ sở lý thuyết

Box-Line Diagram

Box-Line Diagram là một phần của UML (Unified Modeling Language), một ngôn ngữ mô hình hóa thống nhất được sử dụng để thiết kế các hệ thống phần mềm. Trong UML, Box-Line Diagram được sử dụng để mô tả các thành phần của một hệ thống phần mềm và các mối quan hệ giữa chúng. Các thành phần của hệ thống được biểu diễn bằng các hình chữ nhật (box), trong khi các mối quan hệ giữa các thành phần được biểu diễn bằng các đường nối (line) giữa các box. Các box thường được đặt tên để mô tả chức năng của thành phần tương ứng, trong khi các đường nối thường được đánh số để chỉ ra thứ tự hoặc quan hệ giữa các thành phần.



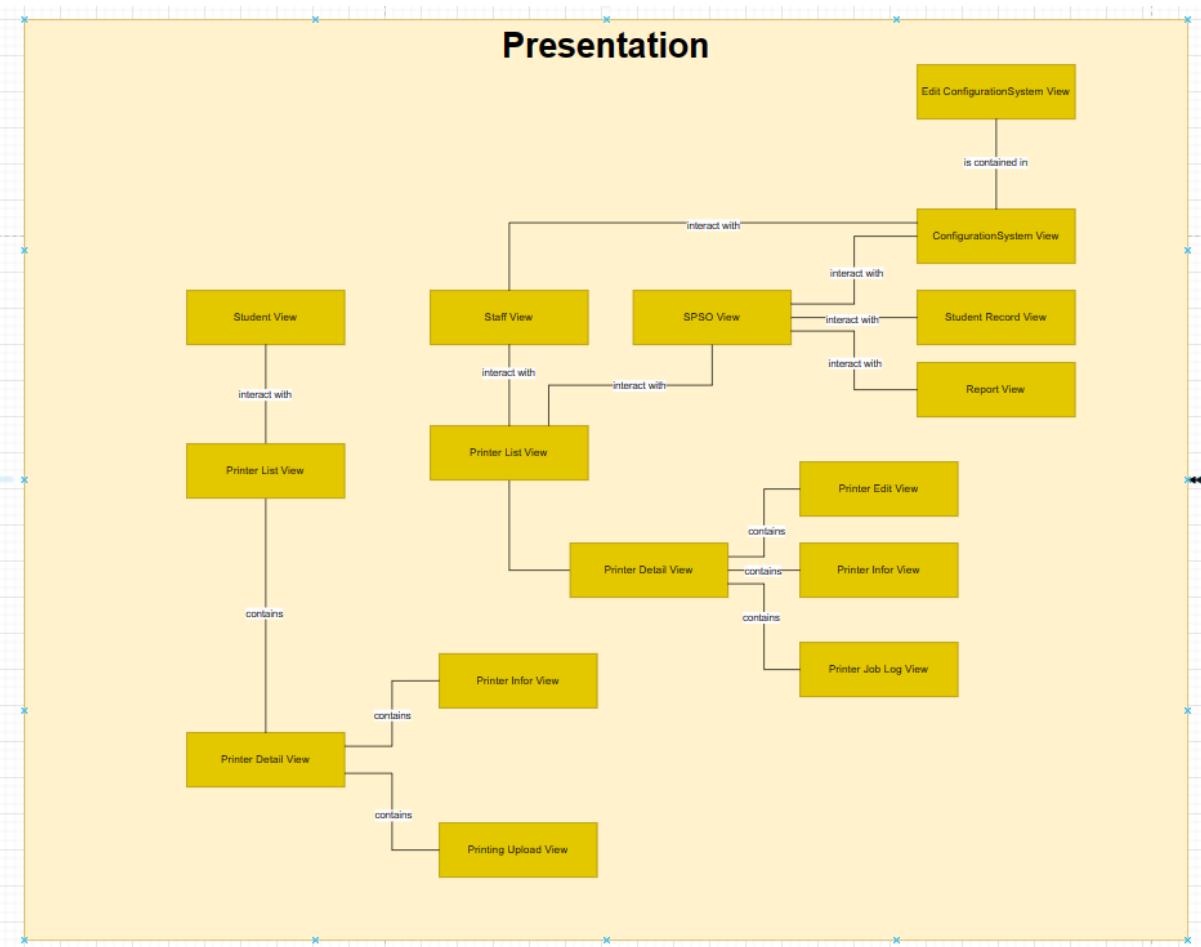
Biểu đồ

Có thể truy cập [tại đây](#) để có thể xem hình vẽ về biểu đồ Box-Line rõ ràng hơn.

Mô tả về biểu đồ Box-Line cho hệ thống HCMUT_SSPS

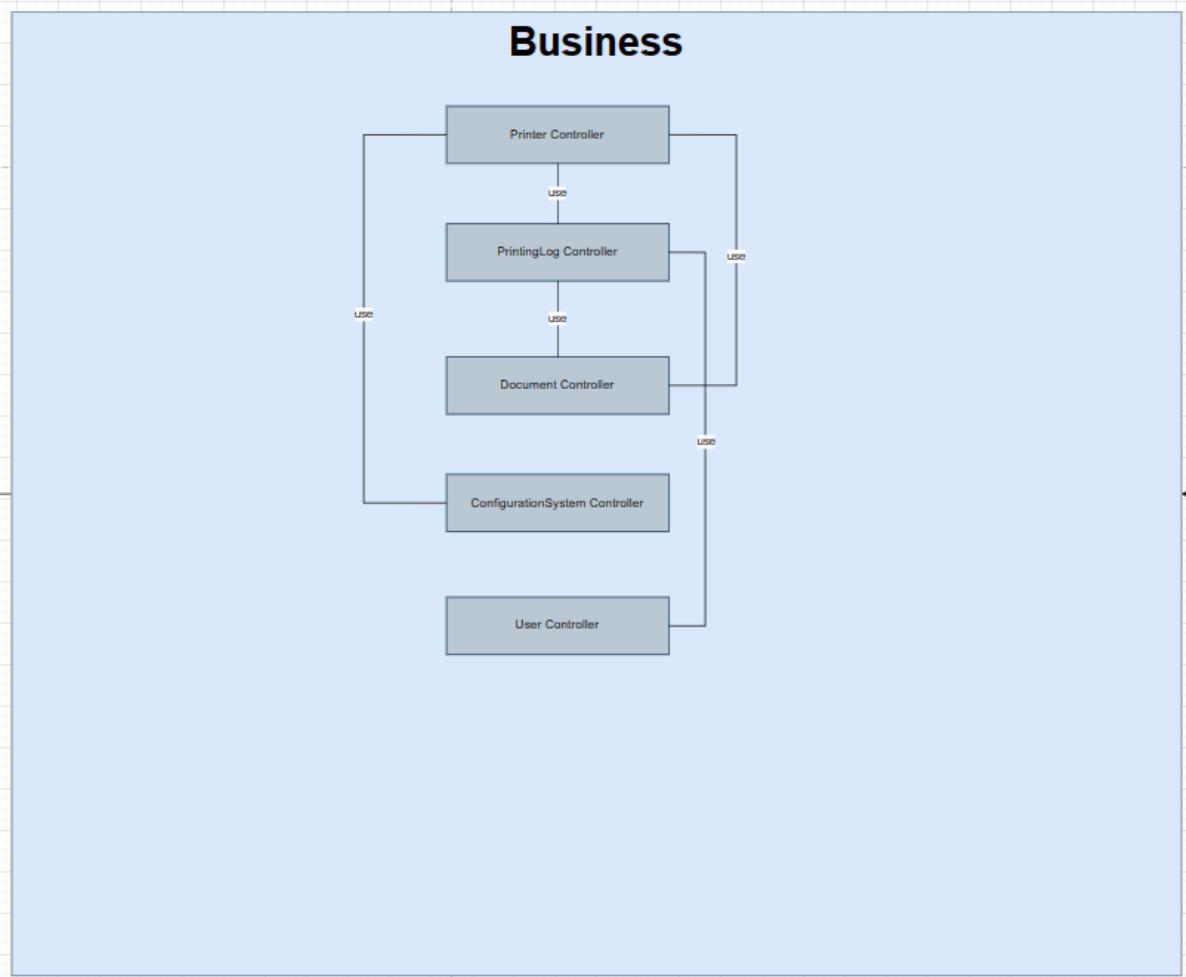
• Lớp trình bày (Presentation Layer)

- Student View tương tác với Printer List View, Printer Detail View (gồm Printer Infor View và Printer Upload View).
- Staff View và SPSO View tương tác với Printer List View, Printer Detail View (gồm Printer Edit View, Printer Infor View và Printer Job Log View), Configuration System View (nhưng Staff View không tương tác được Edit Configuration System View).
- Ngoài ra SPSO View có thể tương tác với Student Record View và Report View.

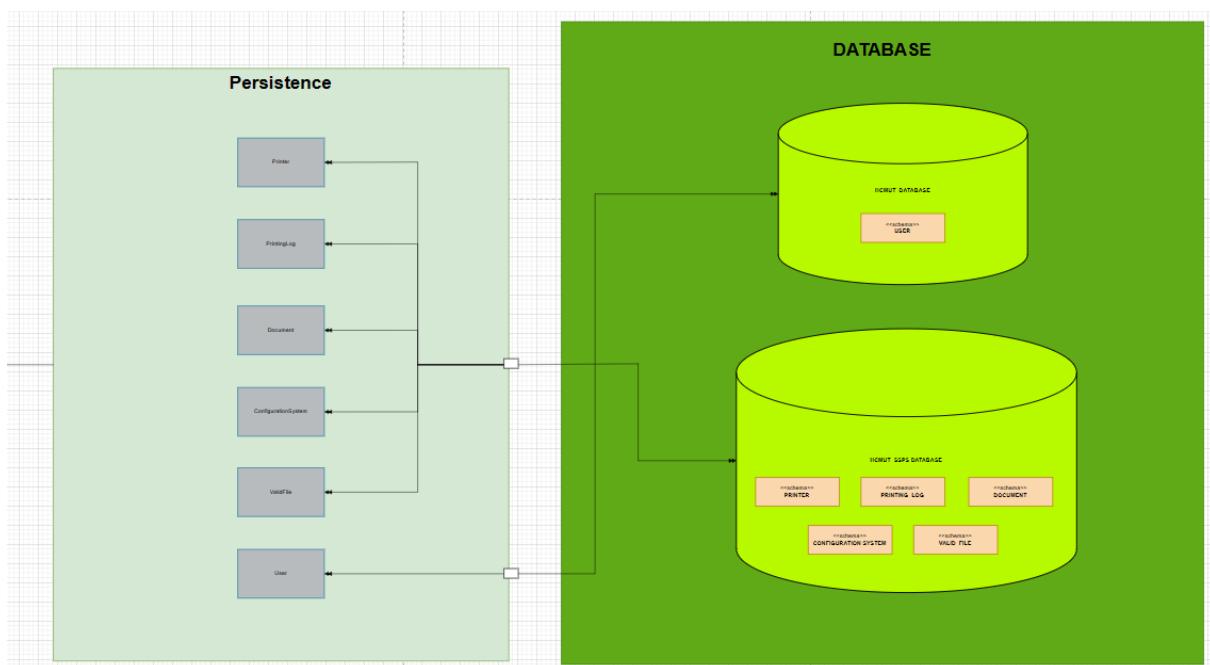


- **Lớp Nghệp vụ (Business)**

- Printer Controller: Xử lý yêu cầu từ các components Printer List View, Printer Detail View.
- PrintingLog Controller: xử lý gián tiếp yêu cầu từ các components Printer Job Log View, trả về cho Printer Controller và component Report View.
- Document Controller: xử lý gián tiếp yêu cầu từ component Printing Upload View trả về cho PrintingLog Controller và Printer Controller.
- ConfigurationSystem Controller: phục vụ cho component ConfigurationSystem View, Edit ConfigurationSystem View.
- User Controller: xử lý yêu cầu của component Student Record View và sử dụng bởi PrintingLog Controller.



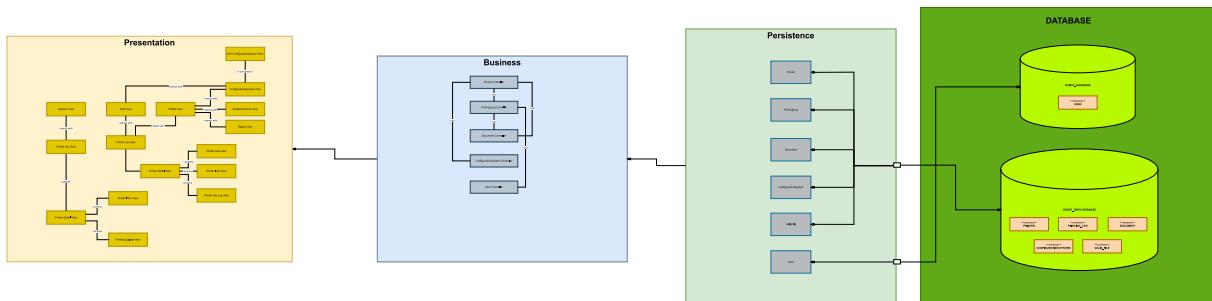
- Lớp Bền vững và Cơ sở dữ liệu (Persistence and Database Layer)



- Printer lưu các thông tin về máy in.

- PrintingLog lưu thông tin về hàng đợi in, lịch sử in.
- Document lưu thông tin về tài liệu in.
- ValidFile lưu thông tin về các file được phép in.
- ConfigurationSystem: thông tin về cấu hình hệ thống in.
- User: thông tin về người dùng, sử dụng external database của HCMUT_DATABASE.

- **Tổng thể về biểu đồ Box-Line (Overall Box-Line Diagram)**



Tương tác giữa các thành phần trong kiến trúc Layer Architecture của hệ thống HCMUT_SSNS được thực hiện thông qua giao thức HTTP và API, tạo ra sự phân tách rõ ràng giữa các lớp chức năng khác nhau. Quá trình này giúp quản lý dự án dễ dàng và thúc đẩy sự hiệu quả trong việc phát triển hệ thống.

Khi một người dùng tương tác với giao diện người dùng (Presentation Layer), ví dụ như thêm máy in mới hoặc yêu cầu thông tin máy in, Presentation Layer tạo một yêu cầu HTTP chứa thông tin tương tác và gửi yêu cầu này đến Business Layer (Controller). Business Layer, đóng vai trò như bộ điều khiển của ứng dụng, nhận yêu cầu và bắt đầu xử lý.

Business Layer có thể thực hiện các hoạt động sau:

1. Truy xuất dữ liệu từ Data Access Layer hoặc Database Layer: Để lấy thông tin về sinh viên, lịch sử in, tài liệu in hoặc bất kỳ thông tin nào cần thiết, Business Layer có thể gọi các phương thức trong Data Access Layer hoặc thực hiện truy vấn vào Database Layer.
2. Xử lý dữ liệu: Business Layer có nhiệm vụ xử lý dữ liệu dựa trên yêu cầu của người dùng và logic ứng dụng. Ví dụ, nếu SPSO yêu cầu xem danh sách hoạt động in ấn của sinh viên theo một tiêu chí nào đó, Business Layer sẽ sắp xếp hoặc lọc dữ liệu trước khi gửi nó đến Presentation Layer.
3. Trả kết quả về Presentation Layer: Khi đã hoàn thành xử lý, Business Layer trả về kết quả, thường thông qua một API endpoint, cho Presentation Layer. Kết quả này có thể chứa dữ liệu cần thiết để hiển thị lên giao diện người dùng.

Presentation Layer sau đó sử dụng kết quả này để hiển thị thông tin cho người dùng một cách thích hợp, đảm bảo một trải nghiệm người dùng tốt. Điều này có nghĩa là giao diện người dùng không trực tiếp tương tác với cơ sở dữ liệu hoặc Data Access Layer, mà thông qua sự trung gian của Business Layer. Điều này giúp đảm bảo tính phân tách giữa giao diện người dùng, logic ứng dụng và dữ liệu, giúp dễ dàng quản lý và bảo trì hệ thống cũng như phát triển một cách hiệu quả trong tương lai.

3.3 Biểu đồ Component

Cơ sở lý thuyết

Component Diagram

Biểu đồ thành phần (**Component Diagram**) được sử dụng để mô hình hóa các khía cạnh vật lý của các hệ thống hướng đối tượng được sử dụng để hiển thị, đặc tả và tài liệu hóa các hệ thống dựa trên thành phần cũng như để xây dựng các hệ thống có thể thực thi thông qua kỹ thuật kỹ thuật chuyển đổi và ngược lại. Biểu đồ thành phần về cơ bản là biểu đồ lớp tập trung vào các thành phần của hệ thống, thường được sử dụng để mô hình hóa góc nhìn triển khai tinh của một hệ thống. Một thành phần đại diện cho một phần modul của hệ thống bao gồm nội dung của nó và có thể thay thế trong môi trường của nó. Trong UML 2, một thành phần được vẽ như một hình chữ nhật với các ngăn xếp theo chiều dọc có thể thêm vào tùy chọn. Một cái nhìn trình bày trình độ cao cấp và trừu tượng của một thành phần trong UML 2 có thể được mô hình như sau:

1. Một hình chữ nhật với tên của thành phần.
2. Một hình chữ nhật với biểu tượng của thành phần.
3. Một hình chữ nhật với văn bản và/hoặc biểu tượng của kiểu phổ cập và/hoặc biểu tượng.

Có hai loại giao diện của thành phần bao gồm:

- Các biểu tượng giao diện đã được cung cấp với một vòng tròn đầy đủ ở cuối của chúng đại diện cho một giao diện mà thành phần cung cấp - biểu tượng "lollipop" này là viết tắt cho một mối quan hệ thực hiện của một giao diện phân loại.
- Các biểu tượng giao diện đã được yêu cầu chỉ với một nửa vòng tròn ở cuối (còn được gọi là ỏ cắm) đại diện cho một giao diện mà thành phần yêu cầu (trong cả hai trường hợp, tên của giao diện được đặt gần biểu tượng của giao diện).

Element	Symbol/Notation	Explanation
Component		Symbol for modules in a system (interaction and communication occur via interfaces).
Package		A package combines multiple elements in a system (e.g. classes, components, interfaces, etc.) into a group.
Artifact		Artifacts are physical units of information (e.g. source code, .exe files, scripts, documents) that are created or required during a system's development process or at runtime.
Provided interface		Symbol for one or more clearly defined interfaces that provide functions. (An open semi-circle is also called a 'socket').
Required interface		Symbol for a required interface that receives functions, services or data from the outside (the circle-with-stick notation is also referred to as a lollipop).
Port		This symbol indicates a separate point of interaction between a component and its environment.
Relationship		Lines act as connectors and show relationships between components.
Dependency		This special connector indicates a dependency between two parts of a system (not always explicitly shown).

Bộ phân loại hệ thống là một phiên bản chuyên biệt của bộ phân loại thành phần. Do đó, yếu tố đánh dấu của bộ phân loại hệ thống thừa hưởng tất cả các quy tắc giống nhau như yếu tố đánh dấu của thành phần. Sự khác biệt duy nhất là một yếu tố đánh dấu của bộ phân loại hệ thống có từ khóa là "subsystem" thay vì "component".

Cổng được đại diện bằng cách sử dụng một hình vuông dọc theo mép của hệ thống hoặc thành phần. Một cổng thường được sử dụng để giúp tiếp cận các giao diện được yêu cầu và được cung cấp của một thành phần.

Về mặt đồ họa, một biểu đồ thành phần là một tập hợp các đỉnh và cung và thường chứa các thành phần, giao diện và các mối quan hệ phụ thuộc, tổ hợp, ràng buộc, phổ cập, hiện thực và liên kết. Nó cũng có thể chứa chú thích và ràng buộc.

Biểu đồ

Có thể truy cập [tại đây](#) để có thể xem rõ hơn về các biểu đồ thành phần (Component Diagram) cho các mô-đun.



Mô tả về biểu đồ thành phần (Component Diagram) cho hệ thống HC-MUT_SSPPS

Module Printing

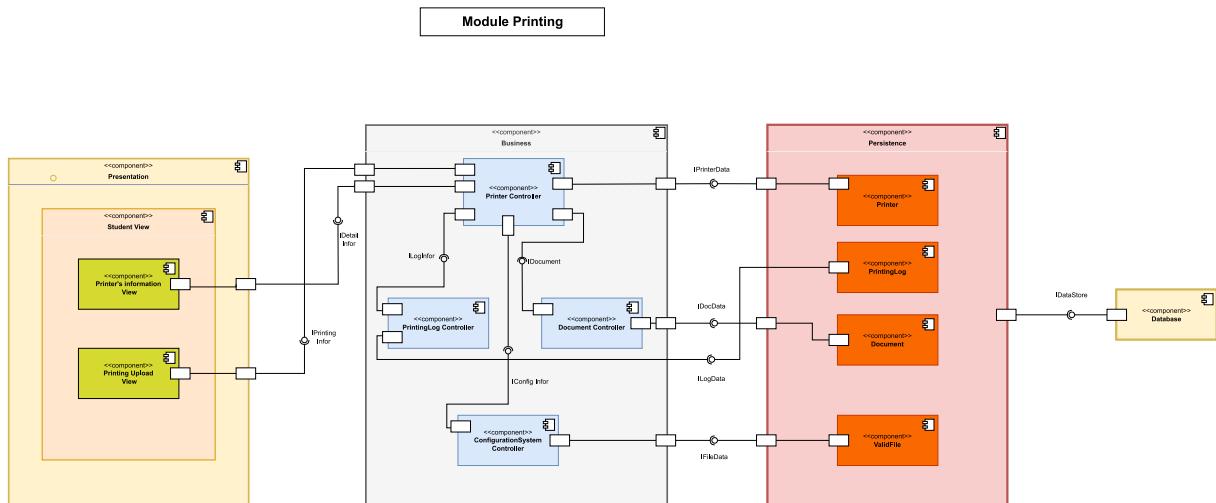
Các interface trong Module Printing thể hiện sự giao tiếp của các components qua các function được mô tả như sau:

- Mô tả các Interfaces:

Tên Interface	Các functions	Mô tả
IDetail Infor	+ getPrinterDetail() + getPrintingLog()	Cung cấp giao diện xử lý yêu cầu về thông tin chi tiết của máy in và lịch sử in từ giao diện người dùng
IPrinting Infor	+ getPrintingInfor() + verifyDocument() + uploadDocument()	Cung cấp giao diện xử lý yêu cầu về xem, cấu hình in ấn các tài liệu in từ giao diện người dùng
ILog Infor	+ getPrintingLog() + addPrintingLog() + removePrintingLog() + updatePrintingLog()	Cung cấp giao diện xử lý các tác vụ yêu cầu in ấn từ máy in
IDocument	+ getDocument() + AddDocument() + removeDocument() + updateDocument() + verifyDocument()	Cung giao diện xử lý yêu cầu tải lên, thêm, xóa, xác nhận tài liệu in từ máy in
IConfig Infor	+ getPrinterConfig() + getValidFile()	Cung cấp giao diện xem thông tin cấu hình in ấn hệ thống
IPrinterData	+ getPrinter() + updatePrinter()	Cung cấp giao diện truy vấn và chỉnh sửa dữ liệu về thông tin máy in từ lớp Database
ILogData	+ addPrintingLog() + removePrintingLog() + getPrintingLog() + updatePrintingLog()	Cung cấp giao diện truy vấn và cập nhật dữ liệu về thông tin nhật ký in ấn từ lớp Database
IDocData	+ addPrintingLog() + removePrintingLog() + getPrintingLog() + updatePrintingLog()	Cung cấp giao diện truy vấn và cập nhật dữ liệu về tài liệu in từ cơ sở dữ liệu
IFileDialog	+ getValidFile()	Cung cấp giao diện truy vấn các file in hợp lệ từ cơ sở dữ liệu

IDataStore	+ addData() + removeData() + getData() + updateData()	Cung cấp giao diện truy vấn, chỉnh sửa, kết nối với cơ sở dữ liệu
------------	--	---

- Hình vẽ cho Module Printing:



Module Manage Printer

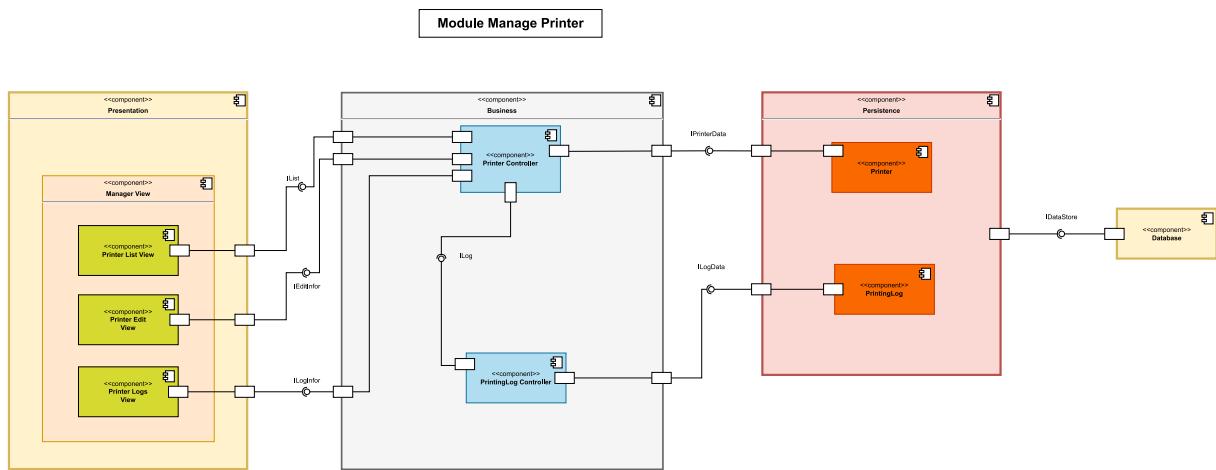
Các interface trong Module Manage Printer thể hiện sự giao tiếp của các components qua các function được mô tả như sau:

- Mô tả các Interfaces:

Tên Interface	Các functions	Mô tả
IList	+ getListPrinter() + addPrinter()	Cung cấp giao diện xử lý việc yêu cầu truy vấn, thêm máy in từ giao diện người dùng
IEditInfor	+ getPrinterInfor() + updatePrinterInfor() + disablePrinter() + enablePrinter() + removePrinter()	Cung cấp giao diện xử lý yêu cầu truy vấn, tùy chỉnh thông tin và trạng thái máy in từ giao diện người dùng
ILogInfor	+ getPrintingLog()	Cung cấp giao diện xử lý yêu cầu truy vấn nhặt ký in ấn của máy in từ giao diện người dùng
ILog	+ getPrintingLog() + addPrintingLog() + removePrintingLog() + updatePrintingLog()	Cung cấp giao diện xử lý yêu cầu truy vấn, tùy chỉnh, cập nhật và xóa nhặt ký in ấn của máy in

IPrinterData	+ addPrinter() + removePrinter() + getPrinter() + updatePrinter()	Cung cấp giao diện xử lý dữ liệu bằng cách truy vấn, tùy chỉnh, cập nhật và xóa máy in từ lớp Database
ILogData	+ addPrintingLog() + removePrintingLog() + getPrintingLog() + updatePrintingLog()	Cung cấp giao diện xử lý dữ liệu bằng cách truy vấn, tùy chỉnh, cập nhật và xóa nhật ký máy in từ lớp Database
IDataStore	+ addData() + removeData() + getData() + updateData()	Cung cấp giao diện truy vấn, chỉnh sửa, kết nối với cơ sở dữ liệu

- Hình vẽ cho Module Manage Printer:



Hình 1: Manager View ở đây bao gồm cả SPSO và Staff



4 Thực hiện - Giai đoạn 1 (Implementation - Sprint 1)

4.1 Thiết lập kho lưu trữ trực tuyến để kiểm soát phiên bản

Dường dẫn tới repo Github: <https://github.com/Phuc2003-vietnam/CNPM-Task4>

The screenshot shows the GitHub repository page for 'CNPM-Task4'. The repository was created by 'Phuc2003-vietnam' and has 1 branch and 0 tags. There are 3 commits from 'PhucNguyen-1807' adding materials to 'Architecture_Design', 'Requirement', and 'System_Modeling' folders, and an update to 'README.md'. The repository is described as 'This is the repo for Assignment of CNPM'. The sidebar shows 0 stars, 1 watching, and 0 forks.

4.2 Thêm tài liệu và thư mục cho phần "Thu thập yêu cầu", "Mô hình hóa hệ thống" và "Thiết kế kiến trúc"

1. Requirement

Thư mục này sẽ lưu trữ yêu cầu đề tài của báo cáo môn Công nghệ phần mềm

The screenshot shows the GitHub repository page for 'CNPM-Task4'. The 'Requirement' folder contains a file named 'Capstone_Project_hk231_2023_v3.pdf' added by 'PhucNguyen-1807' 21 minutes ago. The sidebar shows 0 stars, 1 watching, and 0 forks.

2. System modeling



Thư mục này sẽ lưu trữ các diagram sau: Activity Diagram, Class Diagram, Sequence Diagram, Usecase Diagram

The screenshot shows a GitHub repository named CNPM-Task4. The 'System_Modeling' directory contains several sub-directories and files:

- Activity_Diagram**: Contains ActivityManagePrinters.svg, activity_printing.svg, managelocal_act.svg, and sysconf_act.svg.
- Class_Diagram**: Contains class.png.
- Sequence_Diagram**: Contains SequenceAddPrinter.svg, SequenceManagePrinters.svg, local_1.svg, local_2.svg, local_3.svg, local_4.svg, local_seq.svg, and printing_sequence.svg.

A commit from PhucNguyen-1807 was made 23 minutes ago, adding materials to the Activity_Diagram, Class_Diagram, Sequence_Diagram, and UC_Diagram directories.

3. Architecture Design

Thư mục này sẽ lưu trữ các diagram sau: Boxline Diagram, Component Diagram, Deployment Diagram

The screenshot shows a GitHub repository named CNPM-Task4. The 'Architecture_Design' directory contains:

- Boxline_Diagram**: Contains Controller.png, model_data.png, overall (1).png, and view.png.
- Component_Diagram**: Contains module_Manage_Printer.png and module_printing.png.
- Deployment_Diagram**: Contains App.png, Service.png, and overall.png.

A commit from PhucNguyen-1807 was made 22 minutes ago, adding materials to the Boxline_Diagram, Component_Diagram, and Deployment_Diagram directories.



4.3 Kiểm tra khả năng sử dụng với giao diện đã phát triển ở MVP 1

Để thực hiện bài kiểm tra này, chúng tôi quyết định tạo một nền tảng và tiến hành cuộc khảo sát với những người sử dụng trang web của chúng tôi để xem mức độ đánh giá khả năng sử dụng của trang web. Nhóm cung cấp cho người dùng đường dẫn URL tới trang web để người dùng có thể đánh giá một cách trực quan và rõ ràng nhất. Các câu hỏi trong Google Form chủ yếu được sử dụng để thu thập những ý kiến quan trọng nhất và đánh giá khả năng sử dụng của giao diện người dùng trong quan điểm của họ. Phần lớn người dùng cảm thấy thoải mái với giao diện người dùng của MVP1, chỉ cần một chút cải tiến để trở nên hoàn thiện hơn ở MVP2. Sau cuộc khảo sát này, sẽ có các thay đổi được thực hiện để làm cho hệ thống trở nên thân thiện với người dùng hơn và cung cấp trải nghiệm người dùng tốt hơn.

- Có thể xem Google Form [tại đây](#)
- Trải nghiệm trang web: <https://minhhieu212.github.io/SPS-FE-Project/>

4.3.1 Xác định nhiệm vụ

Trong form được cung cấp, chúng tôi đã đưa ra toàn bộ các giao diện để người dùng trải nghiệm, nhưng trong báo cáo này chúng tôi chỉ tập trung test về các chức năng quan trọng.

4.3.1.a Giao diện Student

Môi trường Test:

- OS: Windows 11
- Browser: Microsoft Edge, Chrome
- URL: <https://minhhieu212.github.io/SPS-FE-Project/>

Thông tin đăng nhập:

- Tên đăng nhập: student1@gmail.com
- Mật khẩu: 1
- Vai trò tài khoản: Student

1. Cấu hình và in ấn

- Xác nhận rằng sinh viên có thể xem giao diện cấu hình và in ấn
 - Sinh viên nhập đường URL đã cung cấp.
 - Sinh viên thực hiện Login thành công với vai trò 'Student'.
 - Sinh viên click vào mục "In tài liệu" ở thanh "Side bar".
 - Xác nhận hệ thống hiển thị danh sách các máy in.
 - Sinh viên click vào bất kỳ vị trí nào trên một máy in mà người dùng muốn in.
 - Xác nhận hệ thống thành công chuyển sang giao diện "Cấu hình file in".
- Xác nhận rằng sinh viên có thể Upload file in lên
 - Từ giao diện "Cấu hình file in" đã hiển thị ở testcase (a)
 - Sinh viên nhấp vào biểu tượng upload ở mục "Tải tệp in".
 - Xác nhận cửa sổ hệ thống file của sinh viên hiện ra.
 - Sinh viên chọn một hoặc nhiều file rồi nhấn "OK".
 - Xác nhận các file đã chọn được hiển thị trên giao diện mục "Tải tệp in".
- Xác nhận rằng sinh viên có thể tiếp tục upload thêm nhiều file in đồng thời
 - Sinh viên nhấn vào nút "Thêm tệp" ở giao diện "Cấu hình file in" tại mục "Tải tệp in".
 - Xác nhận cửa sổ hệ thống file của sinh viên hiện ra.
 - Sinh viên chọn một hoặc nhiều file rồi nhấn "OK".
 - Xác nhận các file đã chọn được hiển thị trên giao diện mục "Tải tệp in" kèm với các file in đã tải lên trước đó.



- (d) Xác nhận rằng sinh viên có thể xóa một file in từ các file đã tải lên
- Sinh viên nhấp vào nút "Xóa tệp" ở giao diện "Cấu hình file in" tại mục "Tải tệp in".
 - Xác nhận cửa sổ chứa danh sách file đã chọn hiện lên.
 - Sinh viên nhấp vào một hoặc nhiều file cần xóa ở cửa sổ giao diện.
 - Xác nhận file được chọn lập tức biến mất ở cả cửa sổ chứa danh sách file in và trong mục "Tải tệp in".
- (e) Xác nhận rằng sinh viên có thể cấu hình cho từng file in từ các file đã tải lên
- Sinh viên nhấp vào "Dropdown list" ở mục "Cấu hình in".
 - Sinh viên chọn file cần cấu hình.
 - Sinh viên điều chỉnh các thông số như "Số bản in", "Kích thước giấy", "Màu sắc in", "Hướng giấy in", "Bố cục in", "Cách in" ở mục "Cấu hình in".
 - Sinh viên tiếp tục chọn các file khác để cấu hình.
 - Xác nhận hệ thống có thể tùy chỉnh cấu hình in
- (f) Xác nhận rằng sinh viên có thể gửi yêu cầu in từ các file in đã tải lên
- Sau khi chọn và cấu hình xong các file in ở testcase (e), sinh viên nhấp vào nút "Gửi yêu cầu in". Cửa sổ "Xác nhận in" hiện ra hiển thị các thông tin "Số file in", "Tổng số bản in".
 - Sinh viên nhấp vào nút "Xác nhận" trên giao diện cửa sổ "Xác nhận in" và quay trở lại giao diện trước đó.
 - Xác nhận hệ thống hoạt động thành công.
- (g) Xác nhận rằng sinh viên không thể gửi yêu cầu in từ các file in đã tải lên nếu chưa file không thuộc các loại file mà hệ thống chấp nhận
- Sau khi chọn và cấu hình xong các file in ở testcase (e), sinh viên nhấp vào nút "Gửi yêu cầu in". Cửa sổ "Xác nhận in" hiện ra hiển thị các thông tin "Số file in", "Tổng số bản in".
 - Sinh viên nhấp vào nút "Xác nhận" trên giao diện cửa sổ "Xác nhận in".
 - Cửa sổ thông báo lỗi có chứa file không hợp lệ hiển ra.
- (h) Xác nhận rằng sinh viên không thể gửi yêu cầu in từ các file in đã tải lên nếu tổng số dư không đủ
- Sau khi chọn và cấu hình xong các file in ở testcase (e), sinh viên nhấp vào nút "Gửi yêu cầu in". Cửa sổ "Xác nhận in" hiện ra hiển thị các thông tin "Số file in", "Tổng số bản in".
 - Sinh viên nhấp vào nút "Xác nhận" trên giao diện cửa sổ "Xác nhận in".
 - Xác nhận cửa sổ thông báo lỗi không đủ số dư hiện ra.

2. Thanh toán giấy in

- (a) Xác nhận rằng sinh viên có thể thấy được giao diện chọn số lượng giấy muốn mua
- Sinh viên nhập đường URL đã cung cấp.
 - Thực hiện Login thành công.
 - Nhấn vào tên của mình ở góc trên cùng bên phải của trang web. Giao diện chi tiết sinh viên hiện ra.
 - Sinh viên nhấp vào nút "Mua thêm giấy" trên giao diện "Chi tiết sinh viên".
 - Xác nhận cửa sổ điều chỉnh lượng giấy in muốn mua hiện ra.
- (b) Xác nhận rằng sinh viên có thể điều chỉnh lượng giấy muốn mua.
- Từ giao diện cửa sổ điều chỉnh lượng giấy in đã hiển thị ra ở testcase (b).
 - Sinh viên có thể tăng hoặc giảm lượng giấy muốn mua bằng cách click vào nút "+" hoặc "-" trên giao diện cửa sổ điều chỉnh lượng giấy in.
 - Xác nhận lượng giấy in trên cửa sổ thay đổi
- (c) Xác nhận rằng sinh viên có thể gửi yêu cầu thanh toán và chuyển sang tương tác với trang thanh toán BKPAY nếu đủ số dư



- Từ giao diện cửa sổ điều chỉnh lượng giấy in đã hiển thị ra ở testcase (c).
 - Sau khi điều chỉnh lượng giấy in, sinh viên nhấn nút "Xác nhận".
 - Xác nhận hệ thống chuyển sang trang BKPay nếu số dư sinh viên đủ.
- (d) Xác nhận rằng sinh viên không thể gửi yêu cầu thanh toán và chuyển sang tương tác với trang thanh toán BKPAY nếu không đủ số dư
- Từ giao diện cửa sổ điều chỉnh lượng giấy in đã hiển thị ra ở testcase trước.
 - Sau khi điều chỉnh lượng giấy in, sinh viên nhấn nút "Xác nhận".
 - Xác nhận hệ thống hiển thị thông báo không đủ số dư.

4.3.1.b Giao diện SPSO

Môi trường Test:

- OS: Windows 11
- Browser: Microsoft Edge, Chrome
- URL: <https://minhhieu212.github.io/SPS-FE-Project/>

Thông tin đăng nhập:

- Tên đăng nhập: spso1@gmail.com
- Mật khẩu: 1
- Vai trò tài khoản: SPSO

1. Cấu hình hệ thống

- (a) Xác nhận rằng SPSO có thể xem được giao diện "Cấu hình hệ thống"
- SPSO nhập đường URL đã cung cấp.
 - SPSO thực hiện Login thành công.
 - SPSO click vào mục "Cấu hình hệ thống" ở thanh "Side bar".
 - Xác nhận giao diện cấu hình của hệ thống hiển thị ra.
- (b) Xác nhận rằng SPSO có thể điều chỉnh các thông số ở mục "Quản lý Vật Liệu Học Tập và Cấp Phát Tài Liệu"
- Từ giao diện "Cấu hình hệ thống" ở testcase (a).
 - SPSO điều chỉnh các thông số "Số lượng giấy mặc định cho từng sinh viên / Học kì", "Thời gian cấp giấy Học kì 1", "Thời gian cấp giấy Học kì 2", "Giá của một tờ giấy A4 khi mua thêm" ở mục "Quản lý Vật Liệu Học Tập và Cấp Phát Tài Liệu".
 - SPSO nhấn nút "Áp dụng"
 - Xác nhận hệ thống cập nhật thành công.
- (c) Xác nhận rằng SPSO có thể điều chỉnh các thông số ở mục "Quản lý Vật Liệu Học Tập và Cấp Phát Tài Liệu" bằng chức năng mặc định và áp dụng
- Từ giao diện "Cấu hình hệ thống" ở testcase (a).
 - SPSO click vào nút "Mặc định".
 - Xác nhận các thông số mặc định được thiết lập sẵn hiện ra.
 - SPSO nhấn nút "Áp dụng"
 - Xác nhận hệ thống cập nhật thành công
- (d) Xác nhận rằng SPSO có thể thêm các loại file được phép in ở mục "Cấu hình các loại file được in" và áp dụng
- Từ giao diện "Cấu hình hệ thống" ở testcase (a).
 - SPSO click vào "Dropdown list" để chọn các loại file muốn thêm.
 - SPSO chọn các loại file được phép in và nhấn nút "Thêm". Các loại file được chọn hiển thị trong mục "Loại file được phép in trong hệ thống" nếu loại file này chưa được thêm.



- Xác nhận hệ thống cập nhật thành công
- (e) Xác nhận rằng SPSO có thể xóa các loại file được phép in ở mục "Cấu hình các loại file được in" và áp dụng
 - Từ giao diện "Cấu hình hệ thống" ở testcase (d).
 - SPSO click vào biểu tượng thùng rác ở một trong số loại file cần xóa ở trong mục "Loại file được phép in trong hệ thống".
 - Xác nhận các loại file được chọn ngay lập tức biến mất và đã bị xóa.

2. Tùy chỉnh máy in

- (a) Xác nhận rằng SPSO có thể xem được "Thông tin máy in" chi tiết
 - SPSO nhập đường URL đã cung cấp.
 - SPSO Thực hiện Login thành công.
 - SPSO click vào mục "Hệ thống máy in" ở thanh "Side bar".
 - Xác nhận Giao diện danh sách máy in hiển thị ra.
 - SPSO click vào nút "Cấu hình" ở một dòng của một máy in bất kì mà SPSO muốn xem trong danh sách máy in.
 - Xác nhận cửa sổ "Thông tin máy in và tùy chỉnh" hiển thị.
- (b) Xác nhận rằng SPSO có thể điều chỉnh được "Thông tin máy in" và thay đổi thành công.
 - Từ giao diện cửa sổ "Thông tin máy in và tùy chỉnh" ở testcase (a).
 - SPSO click vào tab "Tùy chỉnh".
 - SPSO thay đổi các thông số "Nhân hiệu", "Mẫu máy", "Vị trí", "Mô tả", "Trạng thái".
 - SPSO click vào nút "Hoàn tất".
 - Xác nhận hệ thống thay đổi thành công thông tin và trả lại giao diện trước đó

4.3.2 Xác định chiến lược test

Sau khi họp bàn, nhóm chúng tôi quyết định thực hiện bài kiểm thử qua hai hình thức: Định lượng (quantitative) và từ xa (remote).

4.3.2.a Định lượng

Nội dung của hình thức định lượng được chúng tôi sử dụng đó là sử dụng thang điểm đánh giá từ những người trải nghiệm, cụ thể là thang 10. Có 10 lựa chọn tương ứng với 10 mốc điểm hài lòng về trải nghiệm.

Trên thang điểm từ **1 đến 10**, bạn nghĩ sao về giao diện **Đăng nhập** của trang web? *

1 2 3 4 5 6 7 8 9 10

Không hài lòng Vô cùng hài lòng

Bên cạnh đó, người dùng nếu có nhu cầu mô tả chi tiết hơn có thể nhận xét ở một mục riêng:
Lợi thế của hình thức này:

- Một là, sự tiện lợi cho người đánh giá. Thời gian đánh giá nhanh và sự quyết định dựa trên con số cụ thể, không phải chỉ là mô tả hay diễn đạt mang tính chủ quan. Việc sử dụng phương thức trắc nghiệm khiến bài kiểm thử thu hút được nhiều đối tượng tham gia.



Xin vui lòng chia sẻ cảm nhận tổng quan của bạn về trang giao diện Student. Cảm *
thấy như thế nào khi sử dụng? Có những điểm mạnh nào và điểm cần cải thiện?
Hãy đề xuất những cải tiến mà bạn nghĩ sẽ làm cho trang web trở nên tốt hơn.
Điều gì có thể được thêm vào hoặc được cải thiện để nâng cao trải nghiệm người
dùng?

Câu trả lời của bạn

- Hai là, hình thức này giúp cho đội nhóm phát triển quan sát được bảng thống kê chi tiết về những điểm số đã được đưa ra, nhận xét mức độ hài lòng của người dùng đối với từng module của MVP1.
- Ba là, việc nâng cấp trải nghiệm hệ thống sẽ được thực hiện trên các module bị đánh giá thấp trước. Điều này giúp "đánh" vào trọng tâm, tiết kiệm thời gian cho đội nhóm phát triển so với khi kiểm tra định tính.

Tuy vậy hình thức này vẫn có hạn chế. Người dùng thường cảm thấy phiền khi đưa ra mô tả cụ thể. Do vậy, họ thường sẽ không điền vào mục chia sẻ cảm nhận, dẫn đến nhà phát triển có thể không thu được những nhận xét tiềm năng.

4.3.2.b Từ xa

Chúng tôi thực hiện bài kiểm thử từ xa bằng cách triển khai một form nhận xét.

Form được quảng bá trên các nền tảng mạng xã hội cũng như trong các môi trường xã hội mà chúng tôi có thể tiếp cận được (nhà trường, bạn bè...). Ưu điểm của hình thức này đó là phạm vi phổ biến lớn, dễ dàng thu hút được nhiều người tham gia trải nghiệm.

Khuyết điểm của nó so với hình thức in-person đó là thiếu sự tham vấn trực tiếp từ người tham gia.

4.4 Phản hồi từ các tester

4.4.1 Giao diện Student

4.4.1.a Cấu hình và in ấn

Trên thang điểm từ 1 đến 10, phản hồi của các tester về giao diện **Cấu hình và in ấn** như sau:

Khảo sát ứng dụng Dịch vụ in ấn thông minh cho sinh viên Đại học Bách Khoa - HCMUT_SSPS

an.nguyenduc2003qng@hcmut.edu.vn [Chuyển đổi tài khoản](#)



Email của bạn sẽ được ghi lại khi bạn gửi biểu mẫu này

* Biểu thị câu hỏi bắt buộc

Thông tin cá nhân của bạn

Hãy cho chúng mình biết một chút thông tin về bạn nhé <3

Họ và tên của bạn là gì ? *

1

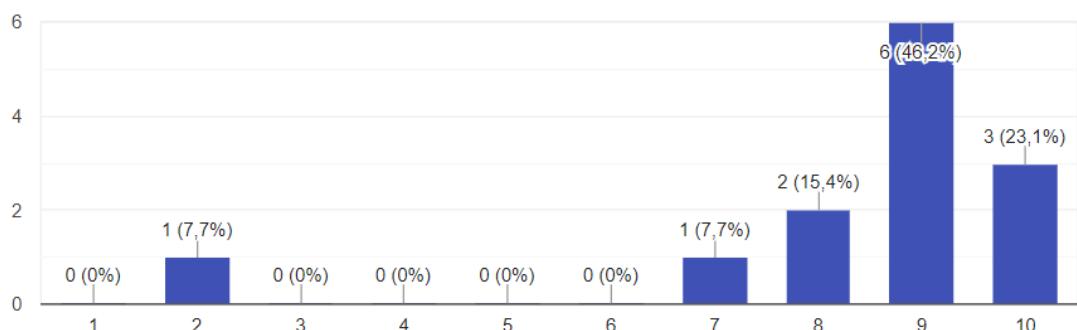
Ngôi trường bạn đang theo học ? *

Trên thang điểm từ **1 đến 10**, bạn nghĩ sao về giao diện **Cấu hình file in** của trang web?



Sao
chép

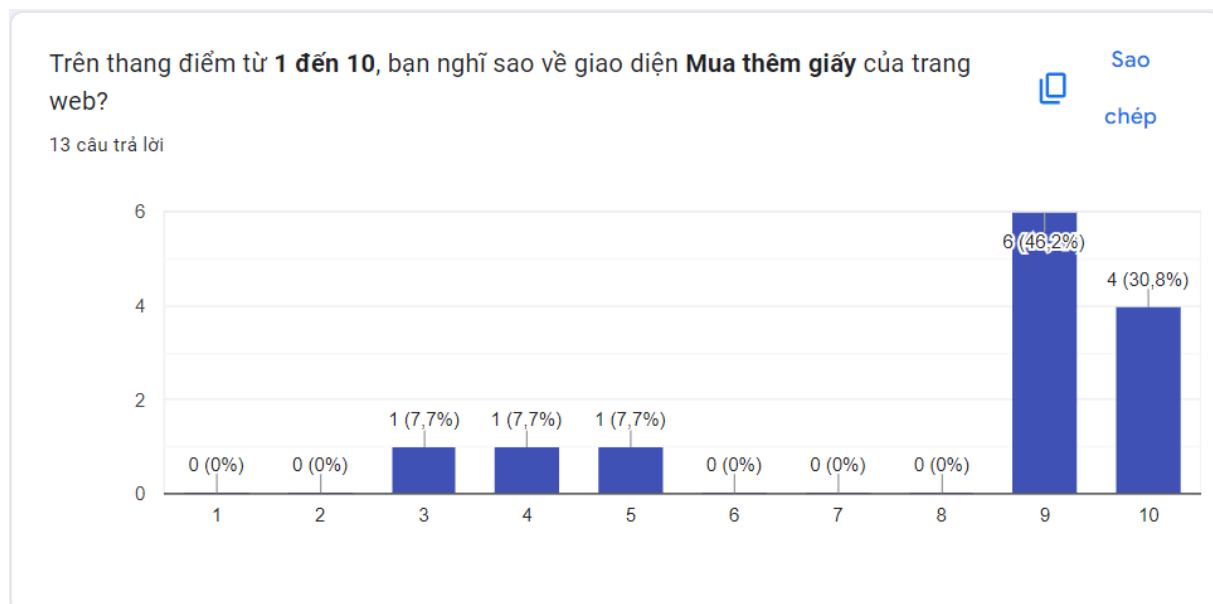
13 câu trả lời



Hình 2: Giao diện Cấu hình file in

4.4.1.b Thanh toán giấy in

Trên thang điểm từ 1 đến 10, phản hồi của các tester về giao diện **Thanh toán giấy in** như sau:

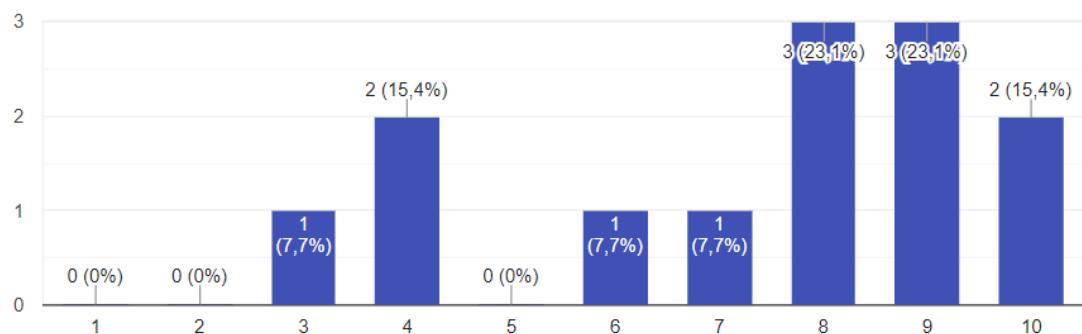


Hình 3: Dánh giá của tester về giao diện của modal **Mua giấy in**

Trên thang điểm từ **1 đến 10**, bạn nghĩ sao về giao diện **BKPay** của trang web?

Sao chép

13 câu trả lời



Hình 4: Dánh giá của tester về giao diện của trang thanh toán **BKPay**



4.4.1.c Đánh giá của tester về giao diện của Sinh viên nói chung

Xin vui lòng chia sẻ cảm nhận tổng quan của bạn về trang giao diện **Student**. Cảm thấy như thế nào khi sử dụng? Có những điểm mạnh nào và điểm cần cải thiện? Hãy đề xuất những cải tiến mà bạn nghĩ sẽ làm cho trang web trở nên tốt hơn. Điều gì có thể được thêm vào hoặc được cải thiện để nâng cao trải nghiệm người dùng?

13 câu trả lời

Ok rồi, mê
Không có đề xuất
Nâng cao cải thiện giao diện người dùng
Có đầy đủ tính năng cần thiết nhưng hơi bất tiện vì phải thanh toán qua BKPay
Không ý kiến
Mục máy in nên chia phụ lục cơ sở để dễ
Mình thấy giao diện khá dễ sử dụng, trông cũng khá bắt mắt
Tốt, font chữ đẹp
TÀI THỦY CHƯƠNG CÓ NHIỀU MÀU SẮC LẮM

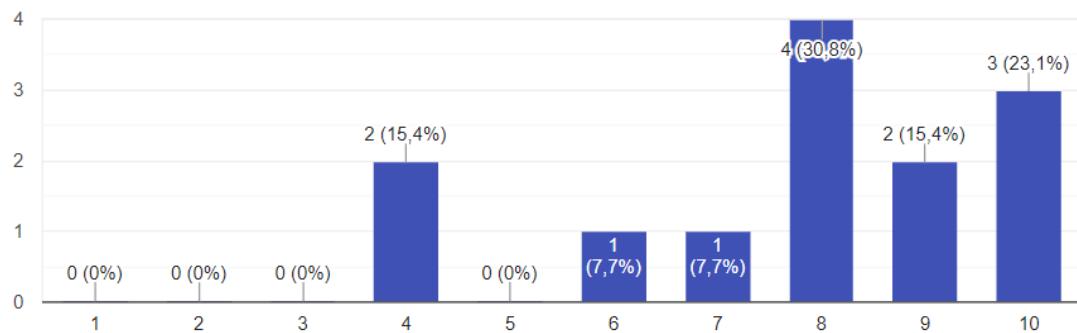
Hình 5: Đánh giá của tester về giao diện Student

4.4.2 Giao diện SPSO

4.4.2.a Cấu hình hệ thống

Trên thang điểm từ 1 đến 10, bạn nghĩ sao về thiết kế giao diện **Cấu hình hệ thống?**

13 câu trả lời

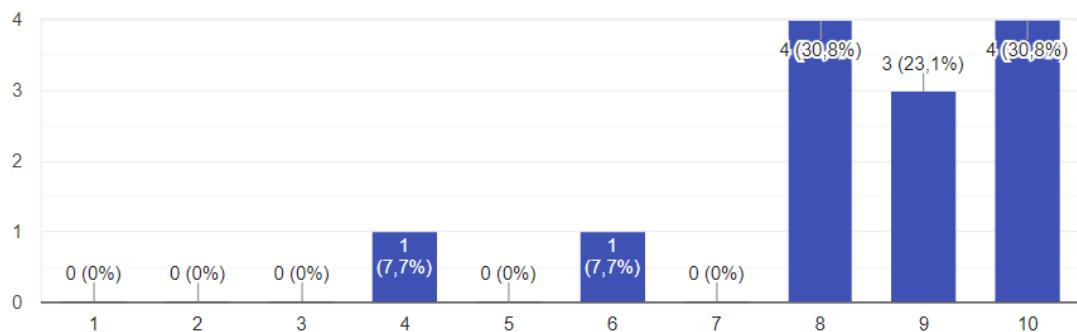


Hình 6: Dánh giá của tester về giao diện **Cấu hình hệ thống**

4.4.2.b Xóa máy in

Trên thang điểm từ 1 đến 10, bạn nghĩ sao về thiết kế giao diện **Xóa máy in?**

13 câu trả lời

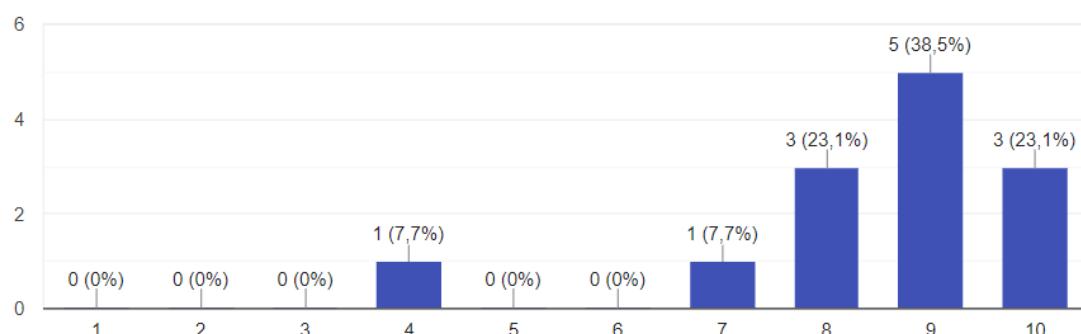


Hình 7: Dánh giá của tester về giao diện **Xóa máy in**

4.4.2.c Tùy chỉnh máy in

Trên thang điểm từ 1 đến 10, bạn nghĩ sao về thiết kế giao diện **Tùy chỉnh máy in**? Sao chép

13 câu trả lời



Hình 8: Dánh giá của tester về giao diện **Tùy chỉnh máy in**

4.4.2.d Đánh giá của tester về giao diện của SPSO nói chung

Xin vui lòng chia sẻ cảm nhận tổng quan của bạn về trang giao diện **SPSO**. Cảm thấy như thế nào khi sử dụng? Có những điểm mạnh nào và điểm cần cải thiện? Hãy đề xuất những cải tiến mà bạn nghĩ sẽ làm cho trang web trở nên tốt hơn. Điều gì có thể được thêm vào hoặc được cải thiện để nâng cao trải nghiệm người dùng?

13 câu trả lời

ok nè

Không có đề xuất

Cải thiện hình ảnh

Tính năng thống kê lượng giấy in có vẻ không cần thiết

Không ý kiến

báo lỗi nữa thì tuyệt

Mình thấy khá đầy đủ các thông tin

Tốt

Không có ý kiến

Hình 9: Dánh giá của tester về giao diện SPSO



5 Thực hiện - Giai đoạn 2 (Implementation - Sprint 2)

5.1 Phát triển MVP 2

5.1.1 Mô tả

Sau khi nhận được các phản hồi, nhóm tác giả đã có một số chỉnh sửa về giao diện, đồng thời thêm các chức năng mới.

Thầy cô và các bạn có thể ghé thăm MVP2 [tại đây](#). Để đăng nhập, mọi người sử dụng tài khoản, mật khẩu như sau:

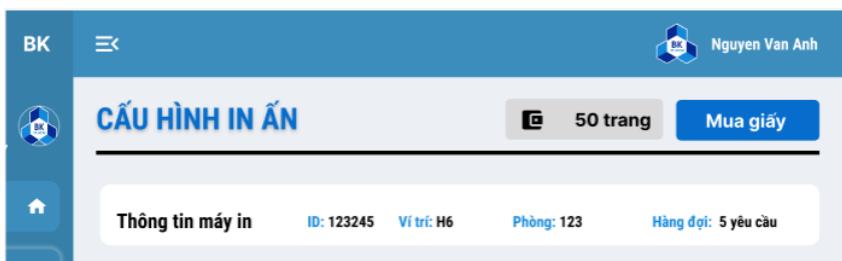
- Sinh viên: student1@gmail.com, mật khẩu: 1
- SPSO: sps01@gmail.com, mật khẩu: 1
- Staff: staff11@gmail.com, mật khẩu: 1

5.1.2 Điểm khác biệt so với MVP 1

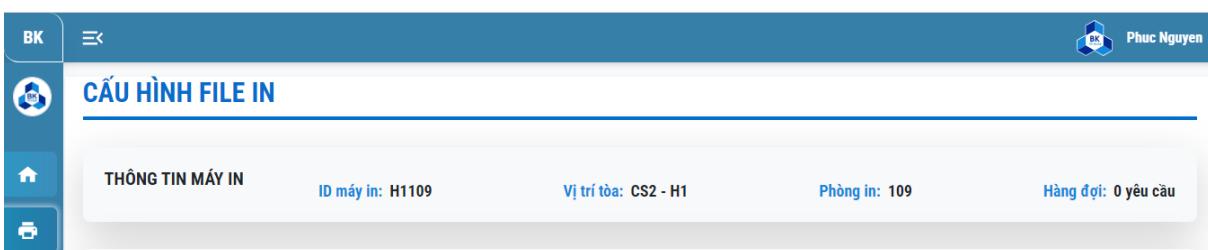
5.1.2.a Thay đổi giao diện

Chúng tôi liệt kê ở đây những điều chỉnh đáng chú ý. Các thay đổi khác nếu có, chỉ dừng lại ở mức độ làm đẹp giao diện, không có thay đổi trong chức năng và cách hoạt động.

Tại trang cấu hình file in thuộc Screenflow Sinh viên, chúng tôi đã xóa hai nút: một là nút hiển thị số dư tài khoản, hai là nút mua giấy. Khi ấn vào tên của sinh viên, anh/cô ấy đã có thể thực hiện việc xem, mua giấy từ bất kỳ trang nào. Do đó chúng tôi đã xóa tính năng này để tránh "thừa" giao diện.



Hình 10: Giao diện trước



Hình 11: Xóa nút số dư và nút mua giấy

Tại trang Quản lý máy in thuộc Screenflow Staff, có một thay đổi nhỏ đó là nút "Tùy chỉnh" đổi tên lại thành "Cấu hình" và được tách thành một cột riêng.



TỔNG SỐ	ĐANG HOẠT ĐỘNG
1	0

ID MÁY IN	SỐ YÊU CẦU IN	LỊCH SỬ IN	TRẠNG THÁI	THIẾT LẬP	
B4405	0	Chi tiết	Lịch sử	Tạm dừng	Cấu hình

Hình 12: *Đổi tên nút và tách cột*

Tại trang Cấu hình hệ thống thuộc Screenflow SPSO, chúng tôi đưa nút Hoàn tất và Mặc định xuống dưới. Hai nút này bây giờ có tác dụng trên toàn bộ form chứ không chỉ là phần nửa trên. Điều này tiện lợi cho việc nhận và gửi dữ liệu qua API.

Quản lý Vật liệu Học tập và Cấp Phát Tài Liệu

Số lượng giấy mặc định cho từng sinh viên / Học kì

100

Thời gian cấp giấy Học kì 1 Thời gian cấp giấy Học kì 2

01/01/2022 01/01/2022

Giá của một tờ giấy A4 khi mua thêm

100 (vnd)

Mặc định Áp dụng

Cấu hình các loại file được in

pptx

Thêm

Hình 13: *Giao diện trước*

Giá của một tờ giấy A4 khi mua thêm

1000 (vnd)

Cấu hình các loại file được in

Chọn loại tệp để thêm... Thêm

Loại file được phép in trong hệ thống

pdf

docx

Mặc định Áp dụng

Hình 14: *Giao diện sau*



Tại trang Quản lý hệ thống máy in của SPSO, chúng tôi thêm nút "Lịch sử" để người quản lý xem được lịch sử in của máy đó.

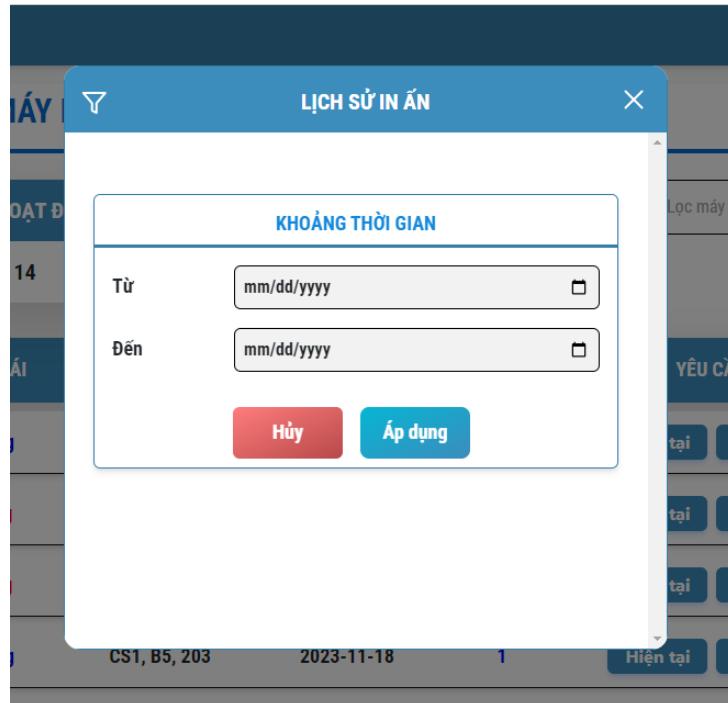
TỔNG SỐ	ĐANG HOẠT ĐỘNG
16	14

ID MÁY IN	TRẠNG THÁI	VỊ TRÍ	KÍCH HOẠT	ĐANG ĐỢI	YÊU CẦU IN	THIẾT LẬP
H3401	Hoạt động	CS2, H3, 401	2023-11-18	6	<button>Hiện tại</button> <button>Lịch sử</button>	<button>Cấu hình</button>
H6211	Tạm dừng	CS2, H6, 211	2023-11-18	0	<button>Hiện tại</button> <button>Lịch sử</button>	<button>Cấu hình</button>
B4405	Tạm dừng	CS1, B4, 405	2023-11-18	0	<button>Hiện tại</button> <button>Lịch sử</button>	<button>Cấu hình</button>
B5203	Hoạt động	CS1, B5, 203	2023-11-18	1	<button>Hiện tại</button> <button>Lịch sử</button>	<button>Cấu hình</button>

Hình 15: Thêm nút lịch sử

Khi ấn vào một modal hiện ra để SPSO xem lịch sử. SPSO có thể nhấn vào biểu tượng chiếc phễu để lọc lịch sử theo giai đoạn.

Hình 16: Modal lịch sử



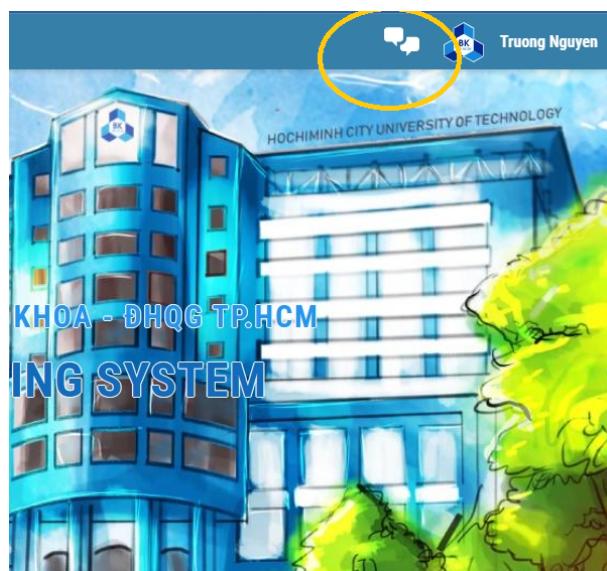
Hình 17: Chức năng lọc

5.1.2.b Thêm chức năng Nhắn tin

Chúng tôi đưa ra ý tưởng này hoàn toàn dựa trên thực tế. Chúng tôi muốn các nhân viên và quản lý có thể trao đổi thông tin trực tiếp qua tin nhắn trên hệ thống thay vì sử dụng ứng dụng bên thứ 3 như zalo, messenger. Công đoạn hiện thực sử dụng thư viện socket.io của javascript.

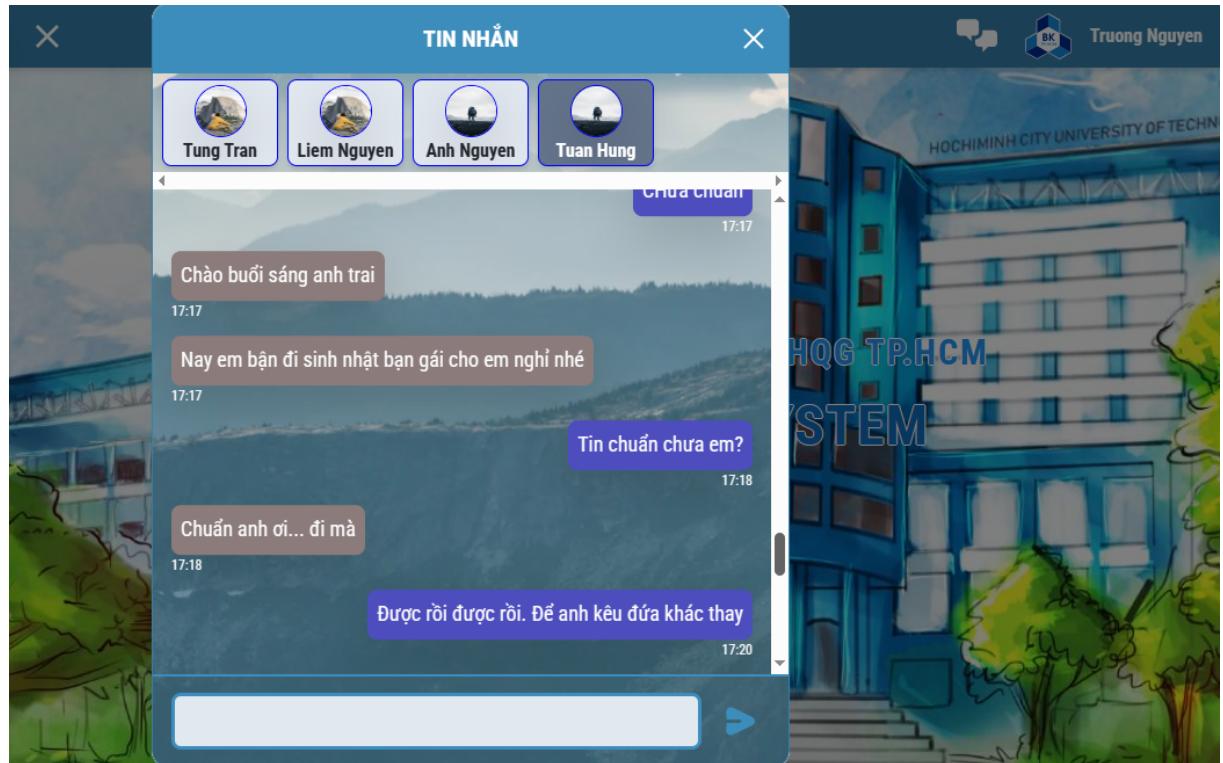
Trong ví dụ minh họa sau đây, SPSO (tên Truong Nguyen) đang nhắn tin với một SPSO (tên Tuan Hung).

Để truy cập mục tin nhắn, ấn vào biểu tượng tin nhắn ở góc trên bên phải:

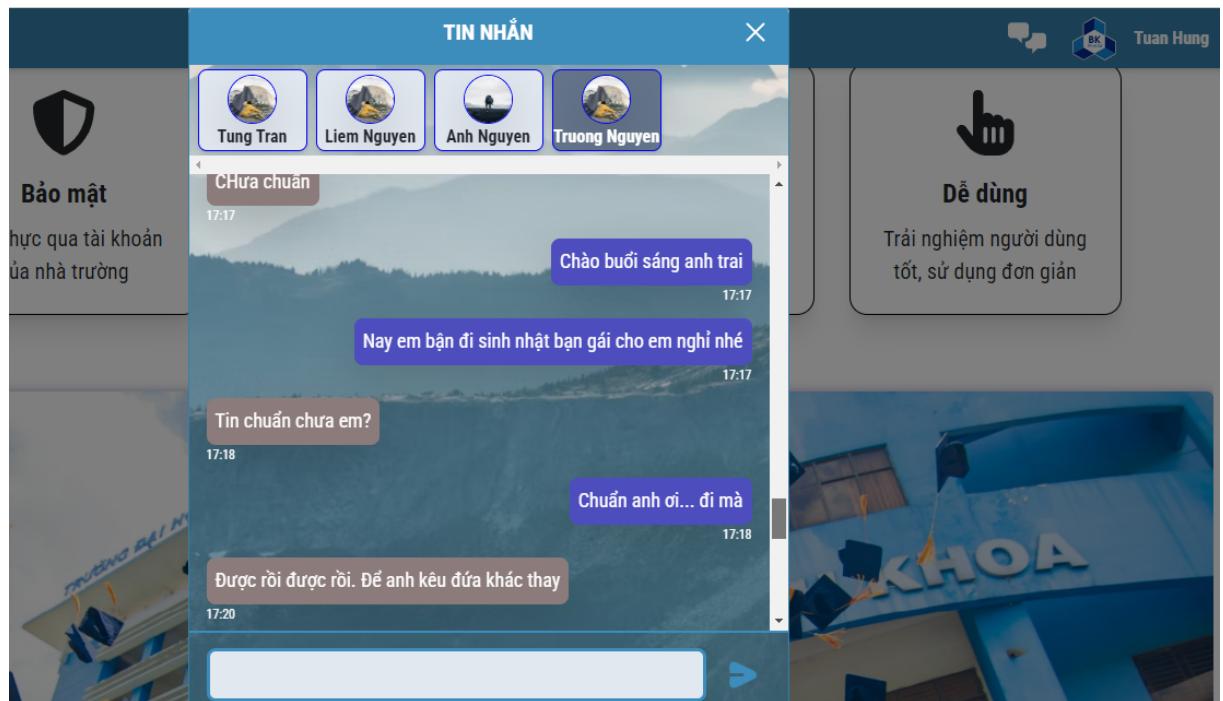




Giao diện SPSO hiện ra, chọn cuộc hội thoại tương ứng (Tuan Hung). Nhập tin nhắn vào khung chat sau đó Enter để gửi.



Tương tự, ở phía Staff chọn cuộc hội thoại tương ứng (Truong Nguyen)





5.1.2.c Thêm chức năng Thông báo

Việc các máy in gặp sự cố là không thể tránh khỏi trong quá trình vận hành. Chúng cần được "tạm dừng" để bảo trì, sửa chữa. Khi một nhân viên (staff) tạm dừng một máy in, hệ thống cần thông báo cho người quản lý (SPSO) rằng đã có một hành động xảy ra. SPSO sau đó sẽ liên lạc (sử dụng tính năng chat ở trên / gọi trực tiếp...) để xác nhận có phải máy đó bị hư thật hay không (để gọi nhân viên quản trị thiết bị) hay chỉ là nhân viên của anh/cô ấy đã nhầm (có biện pháp nhắc nhở, xử lý).

Một cách tương tự, SPSO cũng có thể vô hiệu hóa/kích hoạt một máy in. Thông báo này sẽ được gửi cho các nhân viên để họ biết sự kiện vừa xảy ra.

Để xem thông báo, vào mục Thông báo trên sidebar:



Giao diện của SPSO. Để đánh dấu tất cả là đã đọc, chọn nút "Xác nhận đã xem"

THỜI GIAN	HỌ TÊN	VAI TRÒ	NỘI DUNG	VỊ TRÍ
2023-12-06 04:08:53	Iname fname	staff	Iname Iname đã thực hiện thành công hành động " <i>disable</i> " printer với printerID: H1111, địa chỉ CS1, H1, 109	CS1, H1, 109
2023-12-06 04:08:28	Nguyen Truong	spso	Bạn đã thực hiện thành công hành động " <i>disable</i> " printer với printerID: H6211, địa chỉ CS1, B1, 305	CS1, B1, 305
2023-12-06 04:07:47	Nguyen Truong	spso	Bạn đã thực hiện thành công hành động " <i>disable</i> " printer với printerID: H3401, địa chỉ CS1, H1, 109	CS1, H1, 109
2023-12-06 03:58:01	Iname fname	staff	Iname Iname đã thực hiện thành công hành động " <i>enable</i> " printer với printerID: H3401, địa chỉ CS1, H1, 109	CS1, H1, 109
2023-12-06 03:57:51	Iname fname	staff	Iname Iname đã thực hiện thành công hành động " <i>disable</i> " printer với printerID: H3401, địa chỉ CS1, H1, 109	CS1, H1, 109

Tất cả các thông báo đã được xem.



Smart Printing System

X

Truong Nguyen

THÔNG BÁO HỆ THỐNG

Xác Nhận Đã Xem

THỜI GIAN	HỌ TÊN	VAI TRÒ	NỘI DUNG	VỊ TRÍ
2023-12-06 04:08:53	Iname fname	staff	Iname Iname đã thực hiện thành công hành động "disable" printer với printerID: H1111, địa chỉ CS1, H1, 109	CS1, H1, 109
2023-12-06 04:08:28	Nguyen Truong	spso	Bạn đã thực hiện thành công hành động "disable" printer với printerID: H6211, địa chỉ CS1, B1, 305	CS1, B1, 305
2023-12-06 04:07:47	Nguyen Truong	spso	Bạn đã thực hiện thành công hành động "disable" printer với printerID: H3401, địa chỉ CS1, H1, 109	CS1, H1, 109
2023-12-06 03:58:01	Iname fname	staff	Iname Iname đã thực hiện thành công hành động "enable" printer với printerID: H3401, địa chỉ CS1, H1, 109	CS1, H1, 109
2023-12-06 03:57:51	Iname fname	staff	Iname Iname đã thực hiện thành công hành động "disable" printer với printerID: H3401, địa chỉ CS1, H1, 109	CS1, H1, 109

Tương tự đối với nhân viên:

BK

☰

THÔNG BÁO HỆ THỐNG

Xác Nhận Đã Xem

THỜI GIAN	HỌ TÊN	VAI TRÒ	NỘI DUNG	VỊ TRÍ
2023-12-07 17:34:05	Nguyen Truong	spso	Nguyen Truong đã thực hiện thành công hành động "enable" printer với printerID: H6211, địa chỉ CS2, H6, 211	CS2, H6, 211
2023-12-07 17:33:51	Nguyen Truong	spso	Nguyen Truong đã thực hiện thành công hành động "disable" printer với printerID: H6211, địa chỉ CS2, H6, 211	CS2, H6, 211
2023-12-07 17:32:57	Hung Tuan	staff	Bạn đã thực hiện thành công hành động "enable" printer với printerID: B1306, địa chỉ CS1, H1, 109	CS1, H1, 109
2023-12-07 17:32:52	Hung Tuan	staff	Bạn đã thực hiện thành công hành động "disable" printer với printerID: B1306, địa chỉ CS1, H1, 109	CS1, H1, 109
2023-12-07 14:43:43	Nguyen Truong	spso	Nguyen Truong đã thực hiện thành công hành động "enable" printer với printerID: B4405, địa chỉ CS1, B4, 405	CS1, B4, 405

Tất cả thông báo đã được đọc.



The screenshot shows a system log titled "THÔNG BÁO HỆ THỐNG". It lists five entries with columns: Thời Gian (Time), Họ Tên (Name), Vai Trò (Role), Nội Dung (Content), and Vị Trí (Location). The log details actions like enabling and disabling printers at specific locations.

THỜI GIAN	HỌ TÊN	VAI TRÒ	NỘI DUNG	VỊ TRÍ
2023-12-07 17:37:06	Nguyen Truong	sps0	Nguyen Truong đã thực hiện thành công hành động "disable" printer với printerID: H6211, địa chỉ CS2, H6, 211	CS2, H6, 211
2023-12-07 17:34:05	Nguyen Truong	sps0	Nguyen Truong đã thực hiện thành công hành động "enable" printer với printerID: H6211, địa chỉ CS2, H6, 211	CS2, H6, 211
2023-12-07 17:33:51	Nguyen Truong	sps0	Nguyen Truong đã thực hiện thành công hành động "disable" printer với printerID: H6211, địa chỉ CS2, H6, 211	CS2, H6, 211
2023-12-07 17:32:57	Hung Tuan	staff	Bạn đã thực hiện thành công hành động "enable" printer với printerID: B1306, địa chỉ CS1, H1, 109	CS1, H1, 109
2023-12-07 17:32:52	Hung Tuan	staff	Bạn đã thực hiện thành công hành động "disable" printer với printerID: B1306, địa chỉ CS1, H1, 109	CS1, H1, 109

5.2 Trình bày dự án từ task 1 đến task 5

Slide trình bày về dự án SSPS có thể được xem [tại đây](#).

6 Công nghệ

6.1 UI/UX Design

6.1.1 Figma

Giới thiệu

Figma, một ứng dụng thiết kế giao diện web trực tuyến, hỗ trợ mạnh mẽ cho việc phát triển giao diện web. Cung cấp khả năng làm việc nhóm, làm việc trực tuyến. Figma cho phép nhóm thiết kế tạo ra giao diện tương tác, linh hoạt đáp ứng và đồng bộ thông qua thư viện và components. Figma giúp cho tính nhất quán trong quá trình phát triển giao diện giữa các thành viên được nâng cao và hiệu quả.

Ưu điểm công nghệ

Figma là một công cụ thiết kế và hợp tác trực tuyến phổ biến, nổi tiếng với nhiều ưu điểm đáng chú ý. Dưới đây là một số điểm mạnh của Figma:

- Đa Nền Tảng:** Figma hoạt động trực tuyến thông qua trình duyệt web, không yêu cầu cài đặt phần mềm nền tảng cụ thể, điều này giúp người dùng truy cập và làm việc từ mọi thiết bị và hệ điều hành.
- Hợp Tác Thời Gian Thực:** Công cụ này cho phép nhiều người dùng cùng làm việc trên cùng một bản thiết kế cùng một lúc, hỗ trợ tính thời gian thực và tương tác trực tiếp, giúp đội ngũ thiết kế làm việc hiệu quả hơn.
- Tích Hợp Dễ Dàng:** Figma tích hợp tốt với nhiều ứng dụng và dịch vụ khác, từ các công cụ quản lý dự án đến hệ thống kiểm soát phiên bản, tạo điều kiện thuận lợi cho quá trình làm việc và quản lý dự án.
- Tính Năng Prototyping:** Figma không chỉ giúp thiết kế giao diện người dùng mà còn hỗ trợ chức năng prototyping, giúp người dùng tạo và trải nghiệm các bản mô phỏng tương tác.
- Bảo Mật và Quản Lý Quyền:** Figma cung cấp các tính năng bảo mật mạnh mẽ và quản lý quyền truy cập, giúp bảo vệ thông tin nhạy cảm và kiểm soát quyền truy cập của từng thành viên trong đội.

Lí do sử dụng cho dự án

Dối với dự án lần này, Figma cung cấp cho nhóm một giao diện tương tác giữa các thành viên với nhau. Các thành viên có thể dựa vào Mockup đã vẽ sẵn và code theo, điều đó tạo nên sự thống nhất giữa các giai đoạn với nhau. Ngoài ra với tính năng prototyping, việc tạo các MVP cho thày nhận xét, trình bày sản phẩm vô cùng dễ dàng.



6.2 Frontend

6.2.1 ReactJS

Giới thiệu

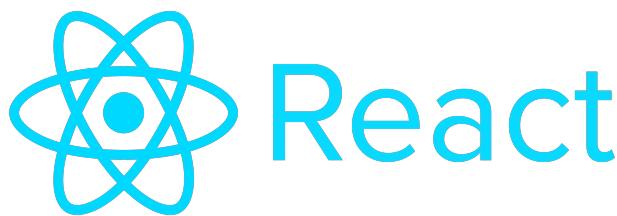
ReactJS là một thư viện JavaScript mã nguồn mở được phát triển bởi Facebook, ra mắt vào năm 2013 với mục đích để xây dựng giao diện người dùng. Nó được sử dụng rộng rãi để xây dựng các trang web SPA (Single Page Application), nói cách khác là ứng dụng sẽ được tải đúng một lần cho toàn bộ, lúc này người dùng sẽ mất thời gian lâu hơn cho việc tải lần đầu, song ở những lần truy cập tới, việc sử dụng trang web sẽ mượt mà hơn như đang thao tác với một ứng dụng di động, tăng trải nghiệm tương tác của người dùng. Nó rất dễ sử dụng và cho phép người dùng có thể tạo các component UI có thể tái sử dụng.

Ưu điểm công nghệ

- Thực Hiện Virtual DOM:** ReactJS sử dụng Virtual DOM để tối ưu hóa quá trình cập nhật giao diện người dùng. Thay vì cập nhật toàn bộ DOM, React chỉ cập nhật những phần thay đổi, giúp tăng hiệu suất ứng dụng.
- Components và Reusability:** ReactJS sử dụng khái niệm components, cho phép phân chia giao diện người dùng thành các thành phần nhỏ hơn. Điều này tạo điều kiện cho tái sử dụng mã nguồn và quản lý dễ dàng.
- Hỗ Trợ JSX:** JSX là một phần quan trọng của React, cho phép viết HTML trong JavaScript. JSX giúp tạo ra mã nguồn dễ đọc, linh hoạt và dễ bảo trì hơn.
- Tương Tác Thị Giác (Declarative):** React sử dụng phong cách lập trình tương tác thị giác, giúp mô tả được ý muốn của giao diện người dùng và giảm thiểu logic phức tạp.
- Cộng Đồng Lớn và Hỗ Trợ:** React có một cộng đồng lớn với nhiều tài nguyên, thư viện và hỗ trợ từ cộng đồng. Điều này giúp giải quyết vấn đề và tận dụng những cải tiến mới nhất.
- Hệ Sinh Thái Mạnh Mẽ:** React đi kèm với một hệ sinh thái mạnh mẽ, bao gồm React Router, Redux cho quản lý trạng thái, và nhiều thư viện khác giúp xây dựng ứng dụng đầy đủ chức năng.

Lí do sử dụng cho dự án

Với những ưu điểm đã liệt kê, ReactJS là lựa chọn phù hợp cho dự án lần này. Ngoài ra, React JS có một công ty lớn và một cộng đồng nhà phát triển lớn đằng sau nó. Tức là nó đã được thử nghiệm nhiều lần, cập nhật thường xuyên. Và quá trình chuyển đổi sang các phiên bản mới diễn ra suôn sẻ nhất có thể.



6.2.2 TailwindCSS

Giới thiệu

Tailwind CSS là một framework CSS mà không giấu điểm yếu hay giả tạo. Thay vì cung cấp các thành phần UI đã được thiết kế sẵn, Tailwind tập trung vào việc cung cấp một bộ công cụ mạnh mẽ để xây dựng và tùy chỉnh giao diện người dùng theo ý muốn. Với Tailwind, bạn sẽ không gặp phải bất kỳ lớp CSS nào đã được đặt tên từ trước. Thay vào đó, bạn sẽ có quyền kiểm soát tuyệt đối trên từng phần tử trong giao diện của mình bằng cách sử dụng các class utility. Điều này mang lại sự linh hoạt lớn cho việc tạo ra giao diện độc đáo và phức tạp.

Ưu điểm công nghệ

- Dễ Học và Sử Dụng:** Tailwind cung cấp một bộ class utility dễ hiểu, giúp giảm độ phức tạp khi xây dựng giao diện.
- Tùy Biến Cao:** Bạn có toàn quyền tùy chỉnh mọi khía cạnh của giao diện thông qua việc chỉnh sửa các giá trị trong file cấu hình.
- Hiệu Suất Tốt:** Tailwind tối ưu hóa mã nguồn đầu ra, giúp giảm kích thước file và tăng tốc độ tải trang.
- Thư Viện Mở Rộng:** Cộng đồng Tailwind đồng đội đa dạng và sáng tạo, mang lại nhiều plugin và thư viện mở rộng.
- Hỗ Trợ Responsive:** Tailwind tích hợp sẵn các class cho việc xây dựng giao diện responsive, giúp dễ dàng điều chỉnh kiểu dáng trên các kích thước màn hình khác nhau.
- Hệ Sinh Thái Mạnh Mẽ:** Tailwind kết hợp tốt với nhiều công cụ và frameworks khác như React, Vue, Laravel, giúp tích hợp linh hoạt vào các dự án.

Lí do sử dụng cho dự án

Thay vì sử dụng CSS thuần để design UI cho trang web (tốn khá nhiều thời gian và công sức thực hiện) thì Tailwind CSS - một framework của CSS cung cấp một thư viện khổng lồ để lập trình viên có thể tạo ra UI một cách dễ dàng. Ngoài ra Tailwind CSS còn giúp cho lập trình viên dễ dàng làm cho website mang tính responsive, mở rộng thêm về dự án, dễ dàng chia nhỏ các components để cho việc tái sử dụng.





6.2.3 Kết hợp

Kết hợp cả ReactJS và Tailwind CSS, nhóm có thể xây dựng trang web với phần Front-end nhanh chóng và tiện lợi. ReactJS cho phép nhóm xây dựng các thành phần UI động và quản lý trạng thái ứng dụng một cách dễ dàng, trong khi Tailwind CSS giúp tạo ra giao diện đẹp mà không cần viết nhiều CSS tùy chỉnh. Sự kết hợp này giúp tiết kiệm thời gian phát triển và nâng cao hiệu suất trang web của nhóm.



6.3 Backend

6.3.1 NodeJS

Giới thiệu

Node.js là một môi trường chạy mã JavaScript phía máy chủ (server-side) được xây dựng dựa trên JavaScript Engine V8 của Google Chrome. Nó cho phép bạn thực thi mã JavaScript trên máy chủ thay vì chỉ trên trình duyệt, mở ra nhiều cơ hội phát triển ứng dụng web và ứng dụng mạng nhanh chóng, mạnh mẽ.

Ưu điểm công nghệ

Node.js mang lại nhiều ưu điểm đáng chú ý, giúp nó trở thành một lựa chọn phổ biến cho việc phát triển ứng dụng web và mạng. Dưới đây là một mô tả chi tiết về những ưu điểm quan trọng của Node.js:

1. Hiệu Suất Cao:

- Môi Trường V8:** Sử dụng JavaScript Engine V8, Node.js cung cấp hiệu suất chạy mã nhanh và mạnh mẽ, giúp xử lý các yêu cầu với thời gian đáp ứng ngắn.

2. Hệ Sinh Thái Phong Phú:

- Nhiều Thư Viện và Module:** Node.js có một hệ sinh thái phong phú với nhiều thư viện và module hỗ trợ, giúp nhà phát triển giảm thời gian và công sức trong việc xây dựng ứng dụng.

3. Hỗ Trợ Ứng Dụng Thời Gian Thực:

- Kiến Trúc Non-blocking và Event-driven:** Node.js hỗ trợ kiến trúc non-blocking và event-driven, giúp xử lý nhanh chóng và đồng thời nhiều yêu cầu in ấn từ nhiều sinh viên mà không ảnh hưởng đến hiệu suất hệ thống.

4. Đa Nền Tảng và Hỗ Trợ Cộng Đồng:

- Khả Năng Phát Triển Đa Nền Tảng:** Node.js hỗ trợ phát triển đa nền tảng, giúp ứng dụng có thể chạy trên nhiều hệ điều hành khác nhau mà không cần thay đổi nhiều mã nguồn.
- Sự Hỗ Trợ Từ Cộng Đồng Lớn:** Node.js có một cộng đồng lớn và tích cực, cung cấp sự hỗ trợ, cập nhật và giải đáp thắc mắc liên tục.

Lí do sử dụng cho dự án

Dối với dự án xây dựng hệ thống máy in thông minh Student Smart Printing System, NodeJS giúp chúng ta xử lý đồng thời nhiều yêu cầu từ sinh viên in ấn mà không làm giảm hiệu suất hệ thống. Điều này quan trọng khi nhu cầu in ấn tăng lên đột ngột, như vào cuối kỳ học hoặc các sự kiện đặc biệt. nhờ kiến trúc Non-blocking và Event-driven đã kể trên. Ngoài ra, Node.js hỗ trợ phát triển đa nền tảng, giúp hệ thống có thể chạy mượt mà trên nhiều loại máy chủ và môi trường khác nhau. .



6.3.2 ExpressJS

Giới thiệu

Express.js là một khung (framework) ứng dụng web phía máy chủ được xây dựng trên Node.js, nó nhằm mục đích giúp đơn giản hóa và tăng cường khả năng phát triển ứng dụng web phía máy chủ bằng JavaScript. Với Express.js, bạn có thể nhanh chóng xây dựng các ứng dụng web mạnh mẽ và hiệu quả.

Ưu điểm công nghệ

Lợi ích chính của Express.js bao gồm sự linh hoạt trong xây dựng ứng dụng web, hỗ trợ middleware để xử lý các yêu cầu và phản hồi, tích hợp dễ dàng với các cơ sở dữ liệu, đơn giản hóa quản lý định tuyến (routing), và hệ sinh thái mở rộng đa dạng thông qua các gói mở rộng của cộng đồng. Điều này giúp giảm thời gian phát triển và làm cho việc xây dựng các ứng dụng web trở nên nhanh chóng và hiệu quả hơn.

Lí do sử dụng cho dự án

Trong dự án lần này, vì thời gian phát triển sản phẩm là vô cùng gấp rút (tầm 1 tháng) để phát triển server, chính vì vậy nhóm sử dụng ExpressJS vì ExpressJS tập trung vào tính đơn giản và linh hoạt, không đưa ra quá nhiều quy tắc cứng nhắc, cú pháp đơn giản và tài liệu chi tiết, ExpressJS là một framework dễ tiếp cận và học. Điều này giúp tối ưu hóa quá trình phát triển và giảm độ phức tạp, đặc biệt là quan trọng trong môi trường dự án nhanh và linh hoạt. Ngoài ra thì ExpressJS được xây dựng dựa trên kiến trúc non-blocking I/O, giúp tối ưu hóa hiệu suất và khả năng mở rộng của hệ thống.



6.3.3 MongoDB

Giới thiệu

MongoDB là một hệ quản trị cơ sở dữ liệu NoSQL (Not Only SQL), phát triển dựa trên kiến trúc hướng tài liệu. Được thiết kế để xử lý dữ liệu có cấu trúc linh hoạt và khả năng mở rộng tốt, MongoDB trở thành một lựa chọn phổ biến cho các ứng dụng web và dự án yêu cầu lưu trữ dữ liệu lớn và đa dạng.

Ưu điểm công nghệ

MongoDB, một hệ quản trị cơ sở dữ liệu NoSQL, mang lại nhiều lợi ích quan trọng cho các dự án:

1. Kiến Trúc Hướng Tài Liệu:

MongoDB lưu trữ dữ liệu dưới dạng BSON (Binary JSON), giúp linh hoạt trong việc lưu trữ dữ liệu có cấu trúc động và phong cách hướng đối tượng.

2. Tính Nhanh và Hiệu Quả:

Sử dụng mô hình lưu trữ tài liệu và khả năng mở rộng ngang giúp MongoDB xử lý lượng dữ liệu lớn và đảm bảo hiệu suất cao.

3. Cú Pháp Truy Vấn Mạnh Mẽ:

MongoDB sử dụng cú pháp truy vấn linh hoạt giúp lọc và trích xuất dữ liệu một cách dễ dàng.

4. Hỗ Trợ cho Index:

MongoDB hỗ trợ việc tạo các index, giúp tăng tốc độ truy vấn và tìm kiếm.

5. Mô Hình Lưu Trữ Phong Phú:

Hỗ trợ mô hình lưu trữ đa dạng, giúp phản ánh cấu trúc dữ liệu tự nhiên trong ứng dụng.

6. Khả Năng Mở Rộng:

MongoDB có khả năng mở rộng dữ liệu ngang, tối ưu hóa hiệu suất khi xử lý dữ liệu lớn.

7. Cộng Đồng Lớn và Hỗ Trợ Tốt:

Có cộng đồng người dùng lớn, cung cấp tài liệu phong phú và hỗ trợ chuyên nghiệp.

8. Tính Linh Hoạt và Tương Thích:

MongoDB linh hoạt và tương thích với nhiều ngôn ngữ lập trình, giúp tích hợp và phát triển dễ dàng.

Lí do sử dụng cho dự án

Nhóm sử dụng MongoDB cho dự án SSPS chính là vì khả năng mở rộng ngang (horizontally scale) do lưu trữ record dưới dạng BSON, mô hình dữ liệu linh hoạt và độc lập với cấu trúc, do đó nếu khi mở rộng database ta chỉ cần bổ sung thêm nhiều server host database(cluster). So với lưu data theo mô hình SQL thì chỉ có thể mở rộng dọc (vertically scale) vì các record trong SQL liên kết với nhau thông qua phép join table, do đó khi ta muốn nâng cấp database server thì chỉ có thể mua thêm tài nguyên về nâng cấp vào server hiện tại, tuy nhiên việc nâng cấp chỉ có thể chậm đến một mức nhất định mà không tăng



mãi được. Ngoài ra việc phát triển dự án trong thời gian ngắn với MongoDB là lựa chọn tốt vì MongoDB không có nhiều ràng buộc về dữ liệu như SQL.



6.4 Realtime

6.4.1 Socket.IO

Giới thiệu

Socket.IO là một thư viện JavaScript mã nguồn mở, cung cấp một cách thuận tiện để thực hiện giao tiếp thời gian thực hai chiều (bidirectional) giữa máy khách (client) và máy chủ (server). Nó hoạt động trên nền tảng Web và hỗ trợ cả các ứng dụng real-time như trò chơi trực tuyến, ứng dụng trò chuyện, cập nhật trạng thái thời gian thực, và nhiều ứng dụng khác.

Ưu điểm công nghệ

- Khả năng Giao Tiếp Thời Gian Thực:** Socket.IO cho phép giao tiếp thời gian thực giữa máy chủ và máy khách, tạo ra trải nghiệm người dùng mượt mà và hiệu quả.
- Dễ Dàng Sử Dụng và Linh Hoạt:** Socket.IO cung cấp một API dễ sử dụng, linh hoạt và đơn giản, giúp phát triển ứng dụng real-time một cách hiệu quả.
- Hỗ Trợ Nền Tảng Đa Dạng:** Socket.IO không chỉ hỗ trợ trên nền tảng Web mà còn có sẵn cho nhiều ngôn ngữ lập trình và môi trường như Node.js, Python, Java,...
- Chuyển Giao Tốt:** Socket.IO tự động xử lý chuyển đổi giữa các phương tiện truyền thông, đảm bảo khả năng tương thích trên nhiều trình duyệt và môi trường mạng.

Lí do sử dụng cho dự án

Trong dự án này, nhóm sử dụng Socket.IO giúp tăng trải nghiệm người dùng trong trường hợp: SPSO nhận được thông báo và trong chức năng chat, hay khi có yêu cầu in ấn được gửi đến máy in thì sinh viên, SPSO hay staff đều có thể cập nhật số lượng ngay lập tức. Giúp mọi người có trải nghiệm mượt mà, nhanh chóng mà không phải reload lại trang.



6.5 Deployment

6.5.1 Docker container

Giới thiệu

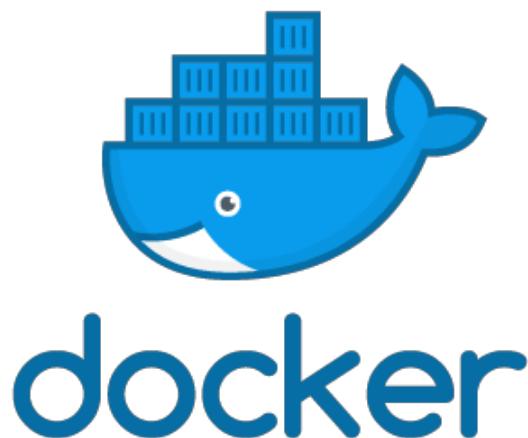
Docker container là một nền tảng ảo hóa tiên tiến giúp đóng gói, vận chuyển, và triển khai ứng dụng một cách nhẹ nhàng và đồng nhất trên môi trường máy chủ. Với Docker, bạn có thể tạo ra các container chứa ứng dụng và môi trường chạy độc lập, đảm bảo tính di động và đồng nhất giữa các máy chủ. Docker giảm bớt sự phức tạp của quá trình triển khai và quản lý hạ tầng, đồng thời cung cấp khả năng mở rộng linh hoạt, giảm tải và tăng tính ổn định của ứng dụng.

Ưu điểm công nghệ

- Đóng Gói Hiệu Quả:** Docker cho phép đóng gói ứng dụng và tất cả các dependencies của nó trong một container độc lập, giúp đảm bảo tính di động và tương thích.
- Khả Năng Chạy Mọi Nơi:** Container Docker có thể chạy trên bất kỳ môi trường nào hỗ trợ Docker, từ máy cá nhân đến máy chủ và đám mây.
- Tính Nhẹ Nhàng và Linh Hoạt:** Docker sử dụng công nghệ containerization, giúp giảm bớt tài nguyên hệ thống tiêu thụ và tạo ra môi trường cô lập.
- Quản Lý Tài Nguyên Hiệu Quả:** Docker cung cấp khả năng tự động mở rộng và quản lý tài nguyên một cách hiệu quả thông qua Docker Swarm hoặc Kubernetes.
- Tích Hợp Dễ Dàng:** Docker tích hợp tốt với nhiều công cụ và dịch vụ khác, tạo ra một môi trường phát triển linh hoạt và dễ quản lý.
- Tiết Kiệm Thời Gian và Chi Phí:** Docker giúp giảm thời gian triển khai và cấu hình, đồng thời tối ưu hóa sự chia sẻ tài nguyên, giảm chi phí hạ tầng.

Lí do sử dụng cho dự án

Trong dự án này, nhóm lựa chọn Docker với mong muốn triển khai sản phẩm một cách nhanh chóng, hiệu quả và chính xác từ môi trường development lên môi trường production mà không cần phải lo lắng về sự phức tạp của quá trình cài đặt trên máy chủ. Sự linh hoạt của Docker cung cấp sự thuận lợi khi mở rộng hệ thống, giảm gánh nặng của việc quản lý dependency, thư viện, và cấu hình cho máy chủ khi scale dự án. Docker không chỉ giúp tối ưu hóa quá trình phát triển mà còn đơn giản hóa quá trình quản lý và triển khai hệ thống.





6.5.2 Web hosting service

Client

Client được nhóm triển khai trên dịch vụ của Github. Client được deployed [url này](#).

Server

Server được nhóm triển khai trên dịch vụ của Render. Server được deployed tại [url này](#).

Mô phỏng máy in

Trang mô phỏng máy in có [tại đây](#).

6.6 Version Control

6.6.1 Git

Giới thiệu

Git là một hệ thống quản lý phiên bản phân tán (Distributed Version Control System - DVCS) giúp theo dõi sự thay đổi trong mã nguồn của dự án phần mềm. Git cho phép nhóm phát triển làm việc đồng thời trên cùng một dự án mà không gây xung đột. Mỗi thay đổi đều được ghi nhận, giúp theo dõi lịch sử và quản lý các phiên bản một cách hiệu quả. Do Linus Torvalds sáng lập và hiện đang được sử dụng rộng rãi trong cộng đồng phần mềm mã nguồn mở và công nghiệp.

Ưu điểm công nghệ

- Quản Lý Phiên Bản Hiệu Quả:** Git giúp theo dõi lịch sử thay đổi của mã nguồn, cho phép nhóm phát triển quay lại bất kỳ phiên bản nào khi cần.
- Phân Tán và Đồng Bộ:** Hệ thống phân tán giúp mỗi thành viên trong nhóm làm việc độc lập, sau đó hợp nhất (merge) lại mã nguồn một cách dễ dàng.
- Nhánh (Branch) Linh Hoạt:** Git cho phép tạo và quản lý nhiều nhánh, giúp phát triển đồng thời nhiều tính năng hoặc sửa lỗi mà không ảnh hưởng đến nhau.
- Hiệu Năng Cao:** Git được thiết kế để hoạt động nhanh chóng, thậm chí với các dự án lớn có lịch sử thay đổi dài.
- Hệ Thống An Toàn:** Mỗi thay đổi trong mã nguồn được xác định bằng mã hash SHA-1, đảm bảo tính toàn vẹn của dữ liệu.
- Cộng Đồng và Hỗ Trợ Mạnh Mẽ:** Git có một cộng đồng đông đảo, với sự hỗ trợ và đóng góp liên tục từ cộng đồng phần mềm mã nguồn mở.

Lí do sử dụng cho dự án

Lợi ích khi áp dụng Git trong dự án là không thể phủ nhận. Việc này giúp giảm thiểu xung đột giữa các phiên bản, tăng tính linh hoạt khi phát triển và giúp nhóm phát triển dễ dàng hợp tác từ xa. Git cũng cung cấp khả năng phục hồi dự án về trạng thái trước đó, giúp bảo vệ mã nguồn và dự án khỏi mất mát không mong muốn. Bằng việc sử dụng Git, nhóm đã làm việc một cách hiệu quả, nâng cao chất lượng sản phẩm và giảm thời gian phát triển, từ đó cải thiện hiệu suất đáng kể.





6.6.2 Github

Giới thiệu

GitHub là một dịch vụ nổi tiếng cung cấp kho lưu trữ mã nguồn Git cho các dự án phần mềm. GitHub có đầy đủ những tính năng của Git, ngoài ra nó còn bổ sung những tính năng về social để các developer tương tác với nhau.

Ưu điểm công nghệ

- Tích Hợp Quản Lý Phiên Bản:** GitHub kết hợp với Git, cung cấp một nền tảng mạnh mẽ cho quản lý phiên bản mã nguồn.
- Giao Diện Người Dùng Thân Thiện:** Giao diện web của GitHub dễ sử dụng, giúp người dùng dễ dàng theo dõi, đánh giá, và quản lý dự án.
- Quản Lý Nhánh Hiệu Quả:** GitHub cung cấp các công cụ mạnh mẽ để tạo, quản lý, và hợp nhất nhánh, giúp phát triển đồng thời nhiều tính năng và sửa lỗi.
- Hệ Thông Ghi Chú (Issues) và Pull Requests:** GitHub có hệ thống ghi chú tích hợp, giúp theo dõi công việc, vấn đề, và đề xuất thay đổi một cách có tổ chức.
- Tích Hợp CI/CD:** Hỗ trợ tích hợp liên tục (Continuous Integration) và phân phối liên tục (Continuous Deployment) thông qua tích hợp với nhiều dịch vụ khác nhau.
- Hỗ Trợ Cộng Đồng Mạnh Mẽ:** GitHub là nơi tập trung của nhiều dự án mã nguồn mở, thu hút một cộng đồng đa dạng và sáng tạo.
- Dịch Vụ Hosting Cho Git Repositories:** GitHub cung cấp dịch vụ lưu trữ và hosting cho các kho mã nguồn của bạn, giúp dễ dàng chia sẻ và làm việc cộng tác.
- Hỗ Trợ API và Tích Hợp Đa Dạng:** GitHub hỗ trợ API mạnh mẽ, cho phép tích hợp với các công cụ và dịch vụ khác.

Lí do sử dụng cho dự án

Sử dụng GitHub mang lại nhiều lợi ích cho nhóm trong việc teamwork cũng như lưu trữ source code. Nó tăng tính linh hoạt và sự hợp tác, cho phép các thành viên làm việc từ xa một cách hiệu quả. Tiếp theo, GitHub giúp giảm thiểu xung đột giữa các phiên bản, bằng cách cho phép các thành viên hợp nhất các thay đổi một cách mượt mà. Sử dụng GitHub không chỉ giúp dự án phát triển nhanh chóng mà còn tạo ra một môi trường làm việc tích cực và sáng tạo.



6.7 Project Management Tool

6.7.1 Trello

Giới thiệu

Trello là một công cụ quản lý dự án trực tuyến phổ biến, được thiết kế để giúp cá nhân và nhóm làm việc cùng nhau hiệu quả hơn trong việc theo dõi và quản lý công việc hàng ngày. Trello sử dụng một giao diện thẻ kéo và thả (drag-and-drop) để giúp bạn tạo, tổ chức và theo dõi danh sách công việc cũng như quản lý dự án.

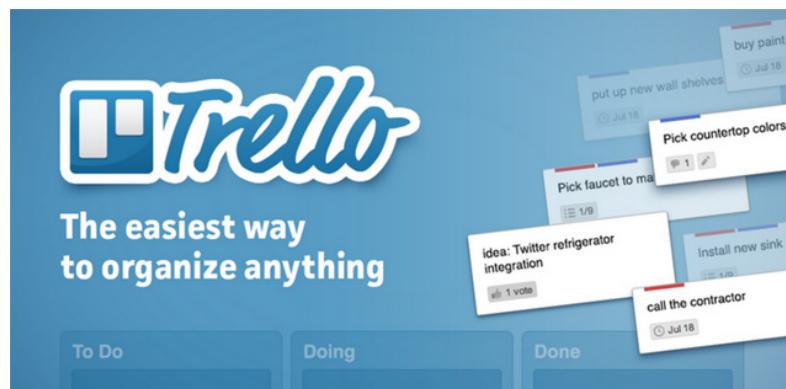
Ưu điểm công nghệ

Ưu điểm của Trello trong quản lý dự án và công việc bao gồm:

1. Tạo, tổ chức và quản lý thẻ: Bạn có thể tạo các thẻ tương ứng với các công việc, nhiệm vụ hoặc dự án cụ thể. Thẻ này có thể được kéo và thả vào các danh sách khác nhau, đại diện cho trạng thái hoặc giai đoạn khác nhau của dự án.
2. Tạo danh sách công việc: Trello cho phép bạn tạo các danh sách công việc, ví dụ: "Chưa bắt đầu," "Đang tiến hành," và "Hoàn thành." Điều này giúp bạn dễ dàng theo dõi tiến trình làm việc của từng thẻ.
3. Gắn kết tệp và thảo luận: Bạn có thể gắn kết tệp đính kèm, nhận xét và thảo luận trực tiếp trên thẻ, giúp giữ mọi thông tin liên quan đến công việc tập trung và dễ dàng truy cập.
4. Phân chia công việc và theo dõi tiến trình: Bạn có thể giao việc cho các thành viên trong nhóm, theo dõi tiến trình làm việc của họ và biết được ai đang làm gì.
5. Tương tác nhóm: Trello là một công cụ tương tác, cho phép các thành viên trong nhóm cùng làm việc, cập nhật thông tin và thấy được thay đổi tức thì.

Lí do sử dụng cho dự án

Với những ưu điểm trên, nhóm sử dụng Trello như một nền tảng phân chia công việc và theo dõi tiến độ làm việc của các thành viên. Trong đó, nhóm sử dụng mô hình Agile để phân chia công việc theo từng sprint. Đảm bảo thành viên phải hoàn thành và cập nhật tiến độ trước thời hạn. Từ đó hiệu quả công việc được tăng lên và mọi người luôn nắm bắt tình hình lẫn nhau.





Tài liệu

- [1] Ian Sommerville. *Software Engineering*. 10th ed. USA: Pearson.
- [2] Phạm Mạnh Cường. 2023. *Stakeholders là gì? Quản lý Stakeholders như thế nào?*. <https://pma.edu.vn/blogs/stakeholders-la-gi/>.
- [3] eTutorials.org. *UML*. <https://etutorials.org/Programming/UML/>