

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



CÔNG NGHỆ PHẦN MỀM (CO3001)

Bài tập lớn

Smart printing service for students at HCMUT

Học kì 241 - Lớp L04

Instructor(s): Trần Trương Tuấn Phát

Group: sundaythekingplays

Student(s):	Lê Duy Khang	2211430 (<i>Lớp L04 - Leader</i>)
	Trương Nguyễn Minh Nhiên	2212452 (<i>Lớp L04</i>)
	Trần Thị Ngọc Huyền	2211311 (<i>Lớp L04</i>)
	Nguyễn Trường Giang	2210829 (<i>Lớp L04</i>)
	Bùi Thành Bách	2210178 (<i>Lớp L04</i>)
	Lý Nguyên Khang	2211437 (<i>Lớp L04</i>)
	Nguyễn Hữu Thắng	2213200 (<i>Lớp L04</i>)



Contents

1 Requirement elicitation - Tìm hiểu yêu cầu	6
1.1 Mô tả ngữ cảnh của Dịch vụ in ấn thông minh dành cho sinh viên tại HCMUT. Xác định các nhóm stakeholders? Nhu cầu và lợi ích của từng nhóm stakeholders?	6
1.1.1 Mô tả ngữ cảnh của Dịch vụ in ấn thông minh dành cho sinh viên tại HCMUT.	6
1.1.2 Xác định các nhóm stakeholders và nhu cầu của họ	7
1.1.3 Lợi ích HCMUT-SSPS mang lại cho từng nhóm stakeholders	7
1.2 Mô tả các yêu cầu chức năng và phi chức năng từ mô tả dự án	8
1.2.1 Yêu cầu chức năng từ mô tả dự án	8
1.2.2 Yêu cầu phi chức năng từ mô tả dự án	9
1.3 Vẽ biểu đồ use-case của toàn hệ thống và phân tích module chính của hệ thống	10
1.3.1 Use case diagram của toàn hệ thống	10
1.3.2 Phân tích use case diagram của module in ấn	10
2 System Modeling - Thiết kế mô hình hệ thống	15
2.1 Activity Diagram	15
2.1.1 Activity diagram cho chức năng đặt in tài liệu	15
2.1.2 Activity diagram cho chức năng xem lịch sử in	16
2.1.3 Activity diagram cho chức năng báo cáo sự cố	17
2.1.4 Activity diagram cho chức năng mua trang in	18
2.2 Sequence Diagram	18
2.2.1 Sequence diagram cho chức năng đặt in tài liệu	18
2.2.2 Sequence diagram cho chức năng xem lịch sử in	21
2.2.3 Sequence diagram cho chức năng báo cáo sự cố	22
2.2.4 Sequence diagram cho chức năng mua trang in	22
2.3 Class diagram	23
2.4 MVP 1 (User interface)	25
2.4.1 Giao diện đăng nhập	27
2.4.2 User interface cho chức năng in tài liệu	27
2.4.3 User interface cho chức năng xem lịch sử in	29
2.4.4 User interface cho chức năng báo cáo sự cố	30
2.4.5 User interface cho chức năng xem mua trang in	31
2.4.6 User interface cho chức năng quản lý máy in của ban quản lý SPSO	32
3 Architectural design - Thiết kế kiến trúc hệ thống	33
3.1 Kiến trúc phân tầng	33
3.1.1 Xác định kiến trúc cho hệ thống.	33
3.1.2 Xác định tổng quan kiến trúc hệ thống	34
3.1.3 Giao diện người dùng	36
3.1.4 Thiết kế lưu trữ	36
3.1.5 Thiết lập API	36
3.2 Deployment Diagram	37
3.3 Component diagram	39
3.3.1 Component diagram - Thực hiện in ấn và đặt lịch	39
3.3.2 Component diagram - Xem lịch sử in ấn	41
3.3.3 Component diagram - Phản hồi hệ thống	43



4 Implementation sprint 1	46
4.1 Tạo Git repository để làm việc và đưa các documents của hệ thống lên	46
4.2 Kiểm tra tính khả dụng của Giao diện người dùng	47
4.2.1 Tổng quan dự án	47
4.2.2 Quá trình kiểm tra	47
4.2.3 Kết quả kiểm thử	48
4.2.4 Kết quả và đề xuất	52
5 Implementation sprint 2	54
5.1 Hiện thực MVP2	54
5.2 Trình bày dự án	54
6 Kết luận	55



Phân công công việc

Ngày	Nội dung	Kết quả
14/9/2024	Họp bàn kế hoạch thực hiện BTL, thống nhất cách phân chia công việc, phân công thực hiện công việc tiếp theo	Nhóm thống nhất mỗi người đảm nhận một task để các thành viên nghiên cứu trước, sau đó cùng thảo luận để chỉnh sửa nếu cần.
18/9/2024	Thảo luận chốt kết quả Task 1.1 và 1.2, hướng thực hiện task 1.3	Huyền và Minh Nhiên hoàn thành task 1.1, 1.2 và viết báo cáo. Bách và Duy Khang tiếp tục thực hiện task 1.3
2/10/2024	Báo cáo kết quả task 1.3 và hướng thực hiện task 2.1, 2.2	Bách và Duy Khang hoàn thành báo cáo cho task 1.3, Giang đảm nhận task 2.1 và Hữu Thắng đảm nhận task 2.2
14/10/2024	Báo cáo kết quả task 2.1 và 2.2, hướng thực hiện task 2.3 và 2.4	Giang và Thắng viết báo cáo cho task 2.1 và 2.2, Duy Khang đảm nhận task 2.3, Huyền và Minh Nhiên làm task 2.4
26/10/2024	Báo cáo kết quả task 2.3 và 2.4, hướng thực hiện task 3	Các thành viên viết báo cáo cho task của mình. Nguyên Khang tiến hành thực hiện task 3
2/11/2024	Báo cáo kết quả task 3	Nguyên Khang tiến hành viết báo cáo cho task 3, Duy Khang tạo repo trên github và thêm các document vào project
5/11/2024	Thảo luận hướng thực hiện task 4.3 và thống nhất cách làm việc trên github	Các thành viên tạo phần của mình trên github, Duy Khang tạo form cho usability test.
13/11/2024	Báo cáo kết quả task 4.3; tiến hành hiện thực MVP 2 trên app mobile và web	Huyền viết báo cáo cho phần 4.3, các thành viên tiến hành hiện thực MVP 2
30/11/2024	Báo cáo kết quả hiện thực MVP 2	Các thành viên hoàn thiện báo cáo.

Bảng 1: Nội dung và kết quả các cuộc họp

Công việc	Thực hiện	Bắt đầu	Kết thúc
Tìm hiểu yêu cầu	Huyền & Minh Nhiên	14/9	17/09
Vẽ use case diagram	Duy Khang, Bách	18/9	1/10
Vẽ activity diagram	Giang	1/10	13/10
Vẽ sequence diagram	Thắng	1/10	13/10
Vẽ class diagram	Duy Khang	14/10	25/10
Thiết kế giao diện người dùng mobile	Huyền	13/10	26/10
Thiết kế giao diện người dùng web	Minh Nhiên	13/10	26/10
Thiết kế kiến trúc hệ thống	Nguyên Khang	25/10	4/11
Phân tích kết quả usability test	Huyền	5/11	12/11
Hiện thực giao diện đăng nhập, các thành phần header, footer (mobile)	Minh Nhiên	13/11	27/11
Hiện thực giao diện tạo đơn in và chọn các thông số in (mobile)	Nguyên Khang	13/11	27/11
Hiện thực giao diện xem lịch sử in (mobile)	Huyền	13/11	27/11
Làm slide thuyết trình	Minh Nhiên	13/11	27/11
Hiện thực giao diện đăng nhập và giao diện quản lý đơn in của nhân viên (web)	Giang	28/11	4/12



Hiện thực giao diện cài đặt máy in và giao diện xem lịch sử in của SPSO (web)	Thắng	28/11	4/12
---	-------	-------	------

Bảng 2: Chi tiết timeline công việc

STT	Họ và tên	MSSV	Nội dung thực hiện	Đánh giá
1	Lê Duy Khang	2211430	Backend, use case diagram, class diagram	15%
2	Trương Nguyễn Minh Nhiên	2212452	Frontend app, requirements, UI design 15%	
3	Trần Thị Ngọc Huyền	2211311	Frontend mobile, requirements, UI design	14%
4	Nguyễn Trường Giang	2210829	Frontend web, activity diagram	14%
5	Bùi Thanh Bách	2210178	Tích hợp external APIs, use case diagram	13%
6	Lý Nguyên Khang	2211437	Frontend mobile, architectural design, component diagram	15%
7	Nguyễn Hữu Thắng	2213200	Frontend web, sequence diagram	14%

Bảng 3: Bảng phân công công việc và kết quả thực hiện của các thành viên



1 Requirement elicitation - Tìm hiểu yêu cầu

1.1 Mô tả ngữ cảnh của Dịch vụ in ấn thông minh dành cho sinh viên tại HCMUT. Xác định các nhóm stakeholders? Nhu cầu và lợi ích của từng nhóm stakeholders?

1.1.1 Mô tả ngữ cảnh của Dịch vụ in ấn thông minh dành cho sinh viên tại HCMUT.

Kỹ thuật in ấn ngày càng trở thành một nhu cầu thiết yếu, không chỉ nhằm mục đích truyền tải thông tin mà còn đóng vai trò quan trọng trong thiết kế, quảng cáo và sản xuất. Đặc biệt, sinh viên và giảng viên tại các trường đại học có nhu cầu cao trong việc in ấn tài liệu, giáo trình, bài tập hoặc để cung cấp phục vụ cho quá trình học tập, nghiên cứu và giảng dạy. Tài liệu in không chỉ hỗ trợ việc ghi chép, lưu trữ mà còn giúp dễ dàng xem lại kiến thức, đồng thời giảm thiểu thời gian tiếp xúc với màn hình máy tính.

Tuy nhiên, việc sử dụng các dịch vụ in ấn tư nhân có thể tiềm ẩn nhiều rủi ro và tác hại. Đầu tiên là nguy cơ rò rỉ thông tin cá nhân hoặc các tài liệu nội bộ có giá trị, đặc biệt là những tài liệu có bản quyền. Các dịch vụ tư nhân thường không đảm bảo đủ tiêu chuẩn về an ninh và bảo mật, dễ dẫn đến việc phát tán thông tin mật. Thứ hai, chất lượng in ấn không đảm bảo, có thể gặp tình trạng tài liệu bị mờ, không đúng yêu cầu hoặc hư hỏng, ảnh hưởng đến kết quả học tập. Chi phí in ấn tại các cơ sở này thường cao hơn, khó phù hợp với túi tiền của sinh viên. Hơn nữa, thời gian chờ đợi để nhận tài liệu cũng có thể kéo dài, gây ảnh hưởng đến tiến độ học tập và nộp bài của sinh viên.

Dể giải quyết những khó khăn trên, phần mềm Dịch vụ in ấn thông minh tại trường Đại học Bách khoa - DHQG-HCM (Student Smart Printing Service - HCMUT_SPS) được triển khai và thiết kế. Dịch vụ đảm bảo dễ dàng truy cập, in tài liệu ngay tại khuôn viên trường một cách tự động, giảm phụ thuộc, linh hoạt và tiện lợi hơn cho người dùng.

- Hệ thống in ấn bao gồm toàn bộ máy in tại cả hai cơ sở của trường, mỗi máy in sẽ gán mã ID, thương hiệu, mẫu mã, mô tả ngắn và vị trí cụ thể (cơ sở, tòa, phòng).
- Tất cả người dùng phải được xác thực bằng cổng thông tin tương tự HCMUT_SSO trước khi sử dụng hệ thống. Các thao tác in ấn đều được ghi lại bao gồm thông tin người dùng, ID máy in, tên tệp, thời gian bắt đầu và kết thúc, cùng với số trang đã in.
- Người dùng là Sinh viên và Cán bộ của trường có thể tải tệp lên hệ thống, tùy chọn các thông số in (khổ giấy, số trang, số mặt, số bản sao), chọn máy in và tiến hành thanh toán. Họ cũng có thể xem lịch sử in và tổng số trang đã in của mình trong khoảng thời gian nhất định. Hệ thống dịch vụ cho người dùng sẽ được vận hành trên ứng dụng di động (mobile app).
- Mỗi học kỳ, sinh viên được cung cấp một số trang in khổ A4 mặc định, có thể mua thêm trang in và thanh toán online.
- Báo cáo về việc sử dụng hệ thống in ấn sẽ được tạo tự động hàng tháng (năm) và được lưu trữ để Ban quản lý (SPSO) theo dõi.
- Ban quản lý in ấn (SPSO) có quyền truy cập lịch sử in ấn của toàn bộ sinh viên và máy in, cũng như thay đổi số trang mặc định, ngày cấp. Đồng thời, họ quản lý các vấn đề liên quan đến máy in như thêm, bặt, tắt, xử lý lỗi của máy in và lỗi phát sinh trong quá trình in. SPSO sử dụng hệ thống được vận hành trên website.



- Nhân viên hỗ trợ in ấn (PS) chịu trách nhiệm quản lý trực tiếp một số máy in nhất định, lưu giữ tài liệu chưa có người nhận, hỗ trợ in trực tiếp và lấy sau tại quầy. Nhân viên in ấn sử dụng dịch vụ này trên website tương tự SPSO.

1.1.2 Xác định các nhóm stakeholders và nhu cầu của họ

Dựa trên đề tài Dịch vụ in ấn thông minh tại HCMUT, có thể xác định 4 nhóm stakeholders chính:

1. Người dùng (User): Bao gồm Sinh viên và Cán bộ trường, có nhu cầu sử dụng dịch vụ in ấn.

- Có thẻ in tài liệu theo 2 hình thức: tự động lấy liền hoặc lấy sau tại quầy.
- Linh hoạt chọn thời gian in ấn theo yêu cầu.
- Tùy chỉnh các thuộc tính in: khổ giấy, số trang, số bản sao và số mặt in.
- Xem nhật ký in ấn trong khoảng thời gian nhất định.
- Kiểm soát số trang in còn lại trong tài khoản cá nhân.
- Báo cáo sự cố liên quan quá trình in ấn đến SPSO.

2. Nhân viên hỗ trợ in ấn (Printing Staff - PS): Đảm bảo quy trình in ấn lấy sau tại quầy, quản lý một số máy in cụ thể và tài liệu.

- Quản lý và cập nhật các đơn in của người dùng.
- Lưu trữ và giao trả tài liệu chưa được nhận hoặc theo yêu cầu lấy sau tại quầy của người dùng.

3. Ban Quản lý in ấn (Student Printing Service Officer - SPSO): Quản lý toàn bộ hệ thống dịch vụ in ấn thông minh trong trường.

- Theo dõi và xem nhật ký in ấn của tất cả người dùng hoặc máy in trong khoảng thời gian nhất định.
- Đảm bảo hệ thống in ấn được bố trí phù hợp trong khuôn viên và hoạt động đúng theo lịch của trường.
- Quản lý tình trạng hoạt động của các máy in, xử lý nhanh chóng các sự cố liên quan quá trình in ấn.
- Thiết lập các thông số của hệ thống: số trang in mặc định và ngày cấp trang in.

1.1.3 Lợi ích HCMUT-SSPS mang lại cho từng nhóm stakeholders

1. Người dùng (User):

- Dễ dàng gửi tài liệu in ngay tại khuôn viên trường.
- Theo dõi lịch sử của toàn bộ giao dịch in ấn qua hệ thống.
- Tiết kiệm thời gian và linh hoạt lựa chọn thời điểm và hình thức in phù hợp với cá nhân.
- Quản lý tài khoản trang in, nắm rõ chi phí in ấn theo đơn giá quy định và thanh toán trực tuyến nhanh chóng, tiện lợi.
- Báo cáo lỗi in ấn, hệ thống và các vấn đề liên quan đến máy in qua hệ thống.

2. Nhân viên hỗ trợ in ấn (PS):

- Dễ dàng quản lý các đơn in lấy sau tại quầy nhanh chóng và tránh thất lạc.



3. Ban Quản lý in ấn (SPSO):

- Dảm bảo dịch vụ in ấn theo hình thức tự phục vụ luôn sẵn sàng và dễ tiếp cận khắp khuôn viên trường.
- Theo dõi được hệ thống và dễ dàng cập nhật tình trạng các máy in cũng như các thay đổi về dịch vụ.
- Hỗ trợ giải quyết các sự cố phát sinh trong quá trình in ấn.
- Kiểm tra toàn bộ lịch sử giao dịch in ấn của người dùng và các máy in.
- Dễ dàng quản lý và thống kê chi phí, doanh thu nhờ hệ thống báo cáo tự động của dịch vụ.

1.2 Mô tả các yêu cầu chức năng và phi chức năng từ mô tả dự án

1.2.1 Yêu cầu chức năng từ mô tả dự án

1. Người dùng (User):

- Đăng nhập và đăng xuất (login) trên hệ thống.
- Xem thông tin cá nhân.
- In tài liệu: in tài liệu tự động ở máy in hoặc đặt đơn in lấy sau tại quầy
- Theo dõi tình trạng đơn in và báo cáo sự cố lỗi.
- Mua thêm trang in
- Xem lịch sử in ấn của cá nhân.
- Xem lịch sử mua trang

2. Nhân viên hỗ trợ in ấn (PS):

- Đăng nhập và đăng xuất (login) trên hệ thống.
- Xem thông tin cá nhân.
- Quản lý các đơn in lấy sau tại quầy của người dùng.
- Báo cáo sự cố hệ thống

3. Ban Quản lý in ấn (SPSO):

- Đăng nhập và đăng xuất (login) trên hệ thống.
- Xem thông tin cá nhân.
- Quản lý máy in: thêm, xoá máy in; kích hoạt hoặc vô hiệu hóa máy in
- Quản lý người dùng: thêm, xoá người dùng
- Xem lịch sử in của hệ thống, lịch sử mua trang của người dùng
- Xem báo cáo thống kê về hoạt động in ấn, doanh thu định kỳ (tháng/ học kì/ năm).
- Chỉnh sửa số trang in mặc định của hệ thống, ngày cấp trang in mặc định của hệ thống
- Tiếp nhận báo cáo sự cố từ người dùng và xử lý.



1.2.2 Yêu cầu phi chức năng từ mô tả dự án

1. Hiệu suất (Performance)

- Thời gian phản hồi: Tất cả các thao tác trên trang giao diện sẽ phản hồi hành động của người dùng trong < 5 giây đối với 95% các yêu cầu.
- Khả năng truy cập cùng lúc: Hệ thống có thể phục vụ tối thiểu 200 người dùng đồng thời mà không giảm hiệu suất (tốc độ phản hồi không vượt quá 5 giây).

2. Khả năng mở rộng (Scalability)

- Hệ thống có khả năng phục vụ từ 300 đến 500 người dùng đồng thời mà không giảm hiệu suất (tốc độ phản hồi không vượt quá 5 giây đối với 95% các yêu cầu).
- Có thể mở rộng thêm từ 15 đến 40 máy in mà không ảnh hưởng đến hiệu suất tổng thể của hệ thống.

3. Tính di động (Portability)

- Ứng dụng di động tương thích với cả 2 hệ điều hành ios và android, và tương thích với > 90 % các thiết bị sử dụng các hệ điều hành đó
- Ứng dụng web tương thích với hầu hết các trình duyệt hiện đại (Chrome, Safari, Firefox, ...)

4. Độ tin cậy (Reliability)

- Hệ thống được thiết kế tách biệt các thành phần, do đó tránh được single point of failure
- Hệ thống xảy ra lỗi đối với không quá 5% số đơn in

5. Bảo mật (Security)

- Người dùng chỉ có thể sử dụng ứng dụng sau khi đã xác thực với hệ thống
- Sử dụng Google OAuth 2.0 để đăng ký và đăng nhập cho người dùng, không lưu trữ mật khẩu người dùng

6. Khả năng bảo trì (Maintainability)

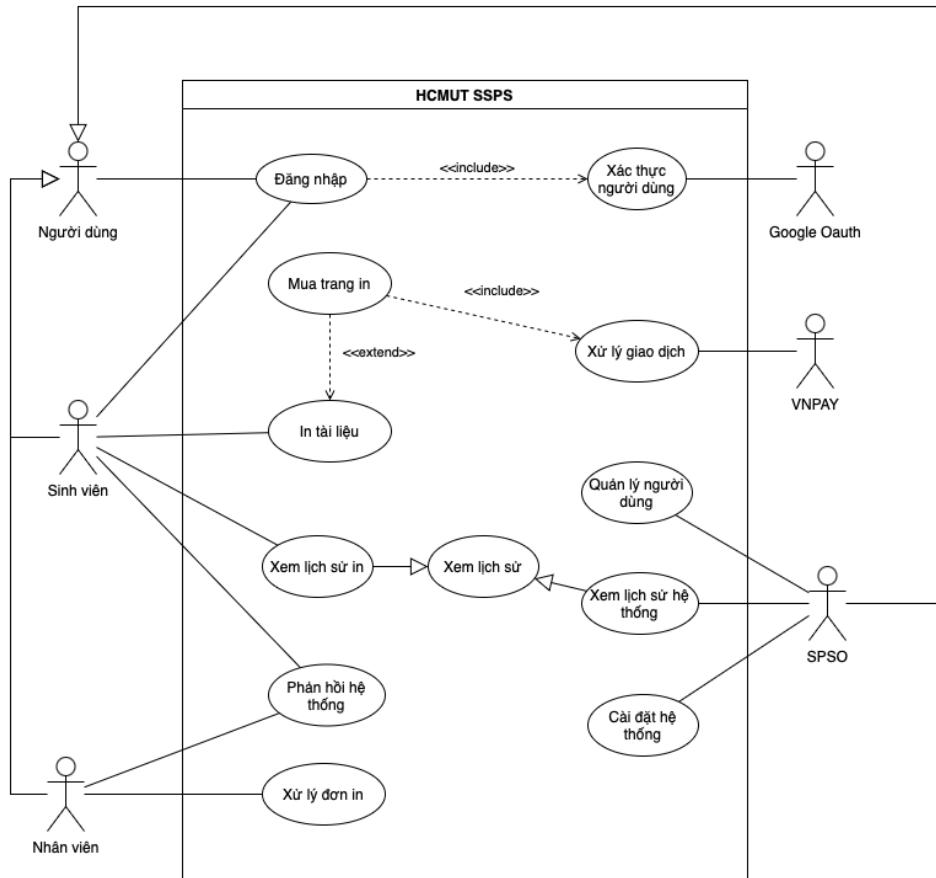
- Ứng dụng có thể khôi phục các sự cố nhỏ trong thời gian < 3 tiếng
- Đối với các sự cố lớn, thời gian tối thiểu để khắc phục là 1 ngày

7. Khả năng sử dụng (Usability)

- Giao diện người dùng phải dễ sử dụng và hiểu rõ đối với 90% người dùng mới trong lần đầu sử dụng.
- Cung cấp các tài liệu chi tiết để giúp người dùng cài đặt và sử dụng ứng dụng

1.3 Vẽ biểu đồ use-case của toàn hệ thống và phân tích module chính của hệ thống

1.3.1 Use case diagram của toàn hệ thống



Hình 1: Use-case diagram cho toàn hệ thống

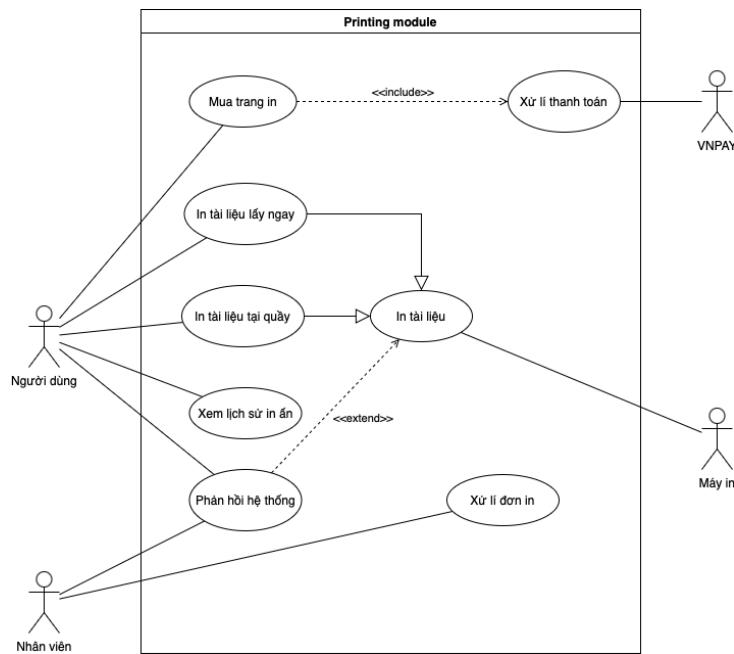
Trên đây là use case diagram của hệ thống HCMUT-SPSS. Đối với ứng dụng này, chúng tôi sẽ chia thành 3 module chính:

- Module in ấn: gồm các use cases In tài liệu, Xem lịch sử in, Phản hồi hệ thống, Xử lý đơn in, Mua trang in
- Module xác thực: gồm use case Đăng nhập và use case Xác thực người dùng.
- Module cài đặt: gồm các use case Xem lịch sử hệ thống, Cài đặt hệ thống và Quản lý người dùng

Trong đó, module in ấn là module chính nên chúng tôi sẽ đi sâu phân tích các use case của module này.

1.3.2 Phân tích use case diagram của module in ấn

Module in ấn của hệ thống gồm 4 actors là Người dùng, Nhân viên, cổng thanh toán VNPAY và các máy in trong hệ thống. Module này gồm các use case liên quan đến hoạt động in ấn của người dùng cũng như mối liên hệ của các hoạt động đó với các actors khác của hệ thống.



Hình 2: Use case diagram của module in ấn

Use case ID	UC - 1.1
Use case name	In tài liệu lấy ngay
Description	Là người dùng, tôi muốn tạo một đơn in lấy ngay
Actor(s)	Người dùng
Priority	Bắt buộc
Trigger	Người dùng ấn vào nút 'In tài liệu lấy liền' trên giao diện
Pre-conditions(s)	Người dùng đã được xác thực với hệ thống
Post-condition(s)	Người dùng in thành công tài liệu
Basic flow	<ol style="list-style-type: none"> Người dùng chọn 'In tài liệu', sau đó chọn 'In tự động lấy liền' Người dùng chọn máy in để in tài liệu. Lần lượt chọn cơ sở, toà, sau đó chọn máy in cần in Người dùng tải các tài liệu cần in lên hệ thống Người dùng chọn các tùy chọn in cho mỗi file. Người dùng xác nhận in và gửi đơn in lên hệ thống Hệ thống tiếp nhận đơn in và tiến hành in tài liệu cho người dùng
Alternative flow	Không có
Exception flow	<ol style="list-style-type: none"> Tài khoản người dùng không đủ số trang in để thực hiện in ấn Đưa ra màn hình thông báo cho người dùng Chuyển hướng người dùng sang giao diện mua trang in

Bảng 4: Đặc tả use case In tài liệu lấy liền



Use case ID	UC - 1.2
Use case name	In tài liệu tại quầy
Description	Là người dùng, tôi muốn in một lượng lớn tài liệu. Do vậy, tôi muốn nhân viên in ấn in và chuẩn bị tài liệu sẵn sàng cho tôi
Actor(s)	Người dùng
Priority	Bắt buộc
Trigger	Người dùng ấn vào nút 'In lấy sau tại quầy' trên giao diện
Pre-conditions(s)	Người dùng đã được xác thực với hệ thống
Post-condition(s)	Người dùng in thành công tài liệu
Basic flow	<ol style="list-style-type: none">Người dùng chọn 'In tài liệu', sau đó chọn 'In lấy sau tại quầy'Người dùng chọn địa điểm nhận tài liệu (cơ sở 1 hay cơ sở 2), sau đó chọn thời điểm cần lấy tài liệuNgười dùng tải các tài liệu cần in lên hệ thốngNgười dùng chọn các tùy chọn in cho mỗi file.Người dùng xác nhận in và gửi đơn in lên hệ thốngNhân viên in ấn sẽ tiếp nhận các đơn in và thông báo cho người dùng khi đơn in sẵn sàng
Alternative flow	Không có
Exception flow	<ol style="list-style-type: none">Tài khoản người dùng không đủ số trang in để thực hiện in ấnDưa ra màn hình thông báo cho người dùngChuyển hướng người dùng sang giao diện mua trang in

Bảng 5: Đặc tả use case In tài liệu lấy sau tại quầy

Use case ID	UC - 1.3
Use case name	Xem lịch sử in ấn
Description	Là người dùng, tôi muốn xem lại lịch sử các đơn in của tôi
Actor(s)	Người dùng
Priority	Bắt buộc
Trigger	Người dùng chọn 'Lịch sử in ấn' từ menu hành động
Pre-conditions(s)	Người dùng đã được xác thực với hệ thống
Post-condition(s)	Lịch sử in ấn của người dùng được hiển thị
Basic flow	<ol style="list-style-type: none">Người dùng chọn 'Lịch sử in ấn' từ menu hành độngLúc này, lịch sử in ấn của người dùng sẽ được hiển thị
Alternative flow	<ol style="list-style-type: none">Người dùng tìm kiếm đơn in theo các tiêu chí như mã đơn in, tình trạng, ...Hiển thị kết quả ứng với tìm kiếm của người dùng
Exception flow	Không có

Bảng 6: Đặc tả use case Xem lịch sử in



Use case ID	UC - 1.4
Use case name	Mua trang in
Description	Là người dùng, tôi muốn mua thêm trang in cho tài khoản
Actor(s)	Người dùng, VNPay
Priority	Bắt buộc
Trigger	Người dùng chọn 'Mua trang in' từ menu hành động
Pre-conditions(s)	Người dùng đã được xác thực với hệ thống
Post-condition(s)	Hệ thống cấp thêm số trang cho người dùng
Basic flow	<ol style="list-style-type: none">Người dùng chọn 'Mua trang in' từ menu hành độngHệ thống hiển thị lịch sử giao dịch của người dùngNgười dùng tiếp tục chọn 'Mua thêm trang in'Người dùng nhập số trang in cần mua và chọn 'Thanh toán'Hệ thống điều hướng người dùng sang giao diện thanh toán của VNPay. Người dùng nhập thông tin và tiến hành thanh toánHệ thống VNPay xử lý giao dịch và trả về kết quả cho hệ thống HCMUT SSPSGhi nhận lại giao dịch của người dùng. Nếu giao dịch thành công, cấp thêm trang in cho người dùng
Alternative flow	Không có
Exception flow	<ol style="list-style-type: none">Người dùng gặp lỗi khi thanh toán với VNPayHiển thị thông báo lỗi cho người dùng

Bảng 7: Đặc tả use case Mua trang in

Use case ID	UC - 1.5
Use case name	Báo cáo sự cố
Description	Tôi muốn phản hồi về hệ thống HCMUT SSPS để cải thiện hệ thống hơn
Actor(s)	Người dùng, Nhân viên
Priority	Bắt buộc
Trigger	Người dùng chọn 'Báo cáo sự cố' từ menu hành động
Pre-conditions(s)	Người dùng đã được xác thực với hệ thống
Post-condition(s)	Phản hồi của người dùng được ghi lại trên hệ thống
Basic flow	<ol style="list-style-type: none">Người dùng chọn 'Báo cáo sự cố' từ menu hành độngChọn loại lỗi cần báo cáo (lỗi máy in, tài liệu giao trẽ,...)Mô tả cụ thể về lỗiGửi báo cáo lên hệ thốngHệ thống ghi nhận phản hồi của người dùng.
Alternative flow	Không có
Exception flow	Không có

Bảng 8: Đặc tả use case Báo cáo sự cố



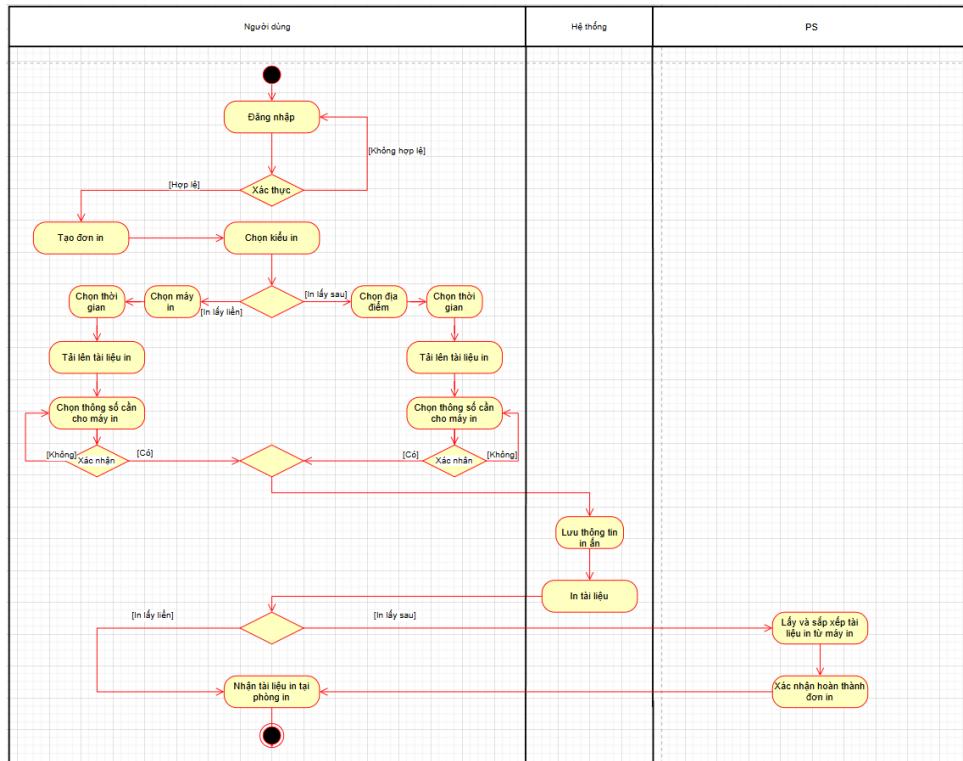
Use case ID	UC - 1.6
Use case name	Xử lý đơn in
Description	Là nhân viên in ấn, tôi muốn quản lý các đơn in của người dùng
Actor(s)	Nhân viên
Priority	Bắt buộc
Trigger	Nhân viên chọn 'Quản lý đơn in' từ giao diện chính
Pre-conditions(s)	Nhân viên đã được xác thực với hệ thống
Post-condition(s)	Trả về kết quả các đơn in của người dùng cho nhân viên
Basic flow	<ol style="list-style-type: none">1. Nhân viên chọn 'Quản lý đơn in'2. Hiển thị danh sách các đơn in tại quầy cho nhân viên3. Nhân viên có thể xem thông tin đơn in của người dùng, đồng thời cập nhật trạng thái đơn in sau khi tài liệu đã sẵn sàng
Alternative flow	<ol style="list-style-type: none">1. Nhân viên tìm kiếm đơn in theo mã đơn in, ngày tạo, ...2. Hiển thị kết quả phù hợp với tìm kiếm của nhân viên
Exception flow	Không có

Bảng 9: ĐẶC TẢ USE CASE XỬ LÝ ĐƠN IN

2 System Modeling - Thiết kế mô hình hệ thống

2.1 Activity Diagram

2.1.1 Activity diagram cho chức năng đặt in tài liệu



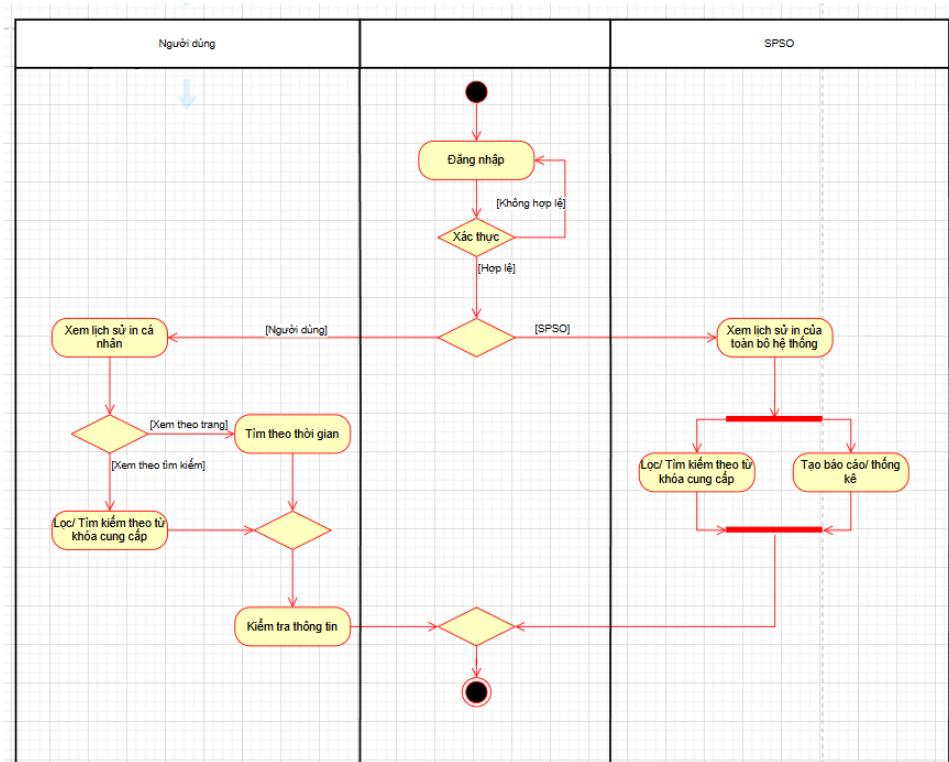
Hình 3: Activity diagram cho chức năng đặt in tài liệu

- Các đối tượng trong biểu đồ:
 - User:** Giảng viên hoặc sinh viên.
 - PS:** Nhân viên quản lý việc in lấy sau.
 - System:** Hệ thống ứng dụng.
- Tổng quan quy trình:
 - Bắt đầu quy trình, người dùng đăng nhập và được xác thực bởi hệ thống.
 - Nếu thông tin không hợp lệ, thì đăng nhập lại. Sau khi đăng nhập thành công, người dùng tiến hành tạo đơn in và sau đó sẽ chọn kiểu in phù hợp với mình.
 - Nếu chọn in lấy liền, người dùng sẽ chọn máy in muốn in tại phòng đặt máy in qua ứng dụng, sau đó chọn thời điểm lấy tài liệu in, sau khi chọn xong 2 bước này người dùng sẽ tải tài liệu in lên và tùy chỉnh thông số cho máy in.
 - Nếu chọn in lấy sau, người dùng sẽ chọn địa điểm lấy tài liệu in và thời gian lấy, sau đó tải tài liệu cần in lên và chọn thông số cho máy in.
 - Sau khi đã xác nhận toàn bộ thông tin, việc in ấn thì hệ thống sẽ tiến hành in ra tài liệu, nếu hủy xác nhận sẽ quay lại việc chọn thông số cho máy in, và tùy vào chọn kiểu in lấy liền hay

in lấy sau, nếu in lấy sau thì nhân viên quản lí tại phòng in sẽ lấy và sắp xếp tài liệu được in ra và ấn xác nhận hoàn thành đơn in trên máy in.

- Cuối cùng người dùng sẽ nhận tài liệu đã in tại phòng in và kết thúc quy trình.

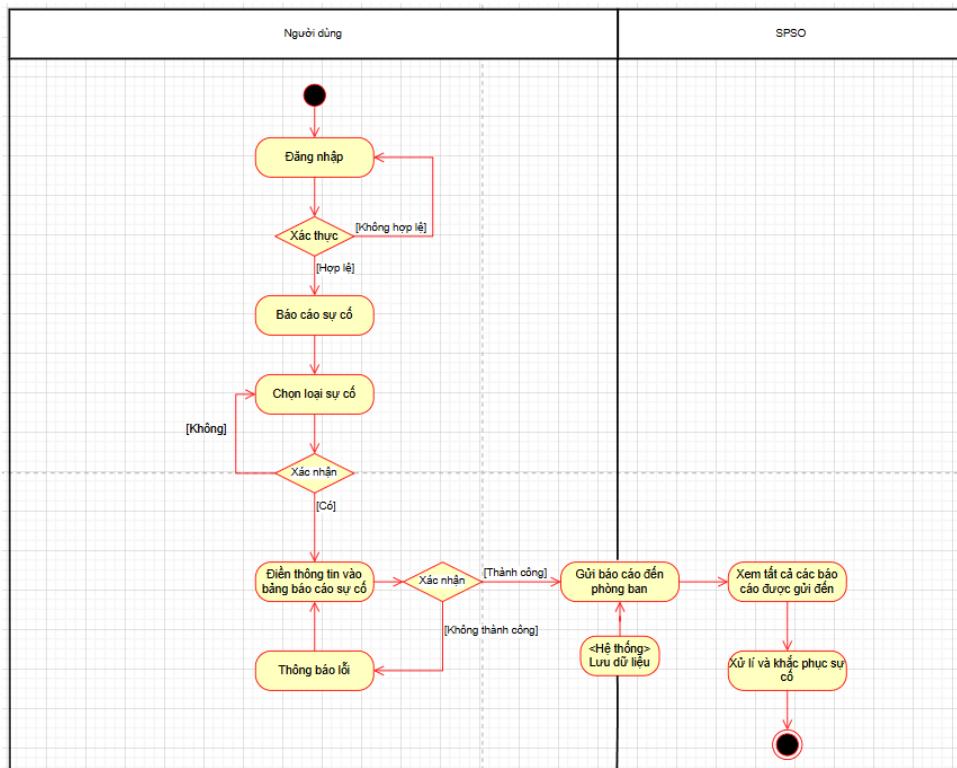
2.1.2 Activity diagram cho chức năng xem lịch sử in



Hình 4: Activity diagram cho chức năng xem lịch sử in

- Các đối tượng trong biểu đồ:
 - **User:** Giảng viên hoặc sinh viên.
 - **SPSO:** Bộ phận kỹ thuật
- Tổng quan quy trình:
 - Bắt đầu quy trình, sau khi đăng nhập và được xác thực thành công bởi hệ thống.
 - Nếu là người dùng thì sẽ chọn vào xem lịch sử cá nhân, nếu là SPSO thì chọn xem lịch sử toàn bộ hệ thống.
 - Với người dùng, có thể tìm kiếm theo từng trang được sắp xếp theo thứ tự thời gian hoặc có thể tìm kiếm trên thanh tìm kiếm bằng "từ khóa". Đối với SPSO, sẽ tìm kiếm dựa trên "từ khóa" cung cấp, đồng thời có thể chọn tạo báo cáo (thống kê) theo tháng hoặc năm.
 - Sau khi tìm kiếm và kiểm tra thông tin thành công, kết thúc quy trình.

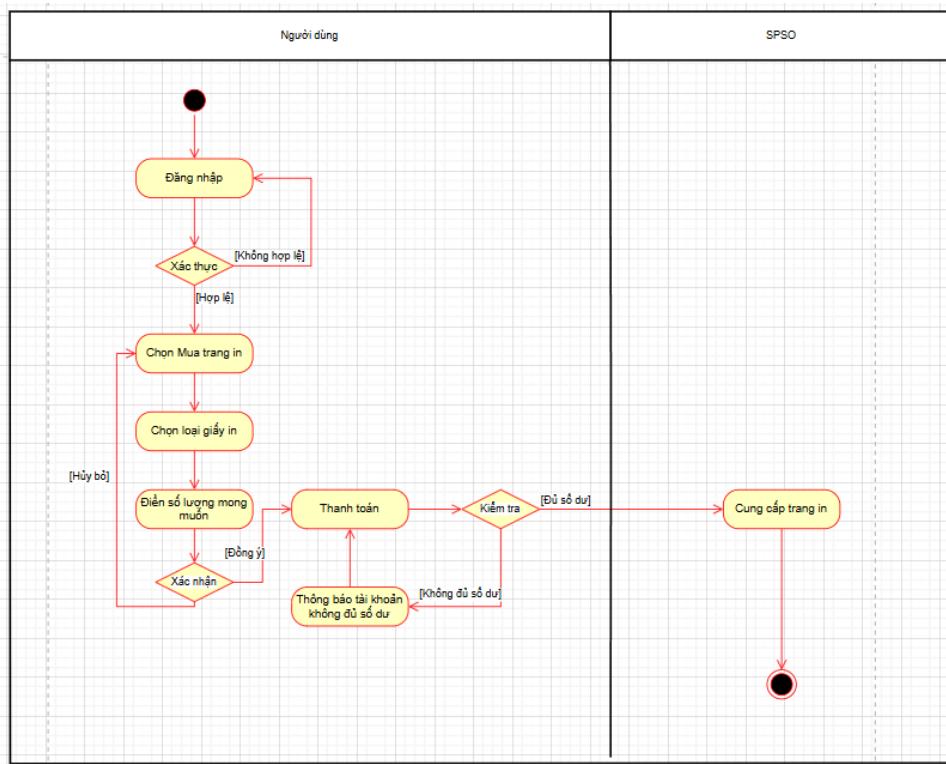
2.1.3 Activity diagram cho chức năng báo cáo sự cố



Hình 5: Activity diagram cho chức năng báo cáo sự cố

- Các đối tượng trong biểu đồ:
 - **User:** Giảng viên hoặc sinh viên.
 - **SPSO:** Bộ phận kỹ thuật
- Tổng quan quy trình:
 - Bắt đầu quy trình, người dùng đăng nhập và được xác thực thành công bởi hệ thống.
 - Chọn vào báo cáo sự cố. Sau đó chọn loại sự cố mà người dùng gặp phải và nhấn xác nhận chính xác cho loại sự cố đó.
 - Điền thông tin chi tiết cho sự cố gặp phải và xác nhận. Nếu có thay đổi thì không xác nhận và chỉnh sửa lại, còn không thì xác nhận thông tin về sự cố và gửi về phòng ban quản lý.
 - SPSO sẽ nhận được báo cáo về sự cố và tiến hành xem xét các báo cáo. Sau đó xử lý và khắc phục sự cố, kết thúc quy trình.

2.1.4 Activity diagram cho chức năng mua trang in



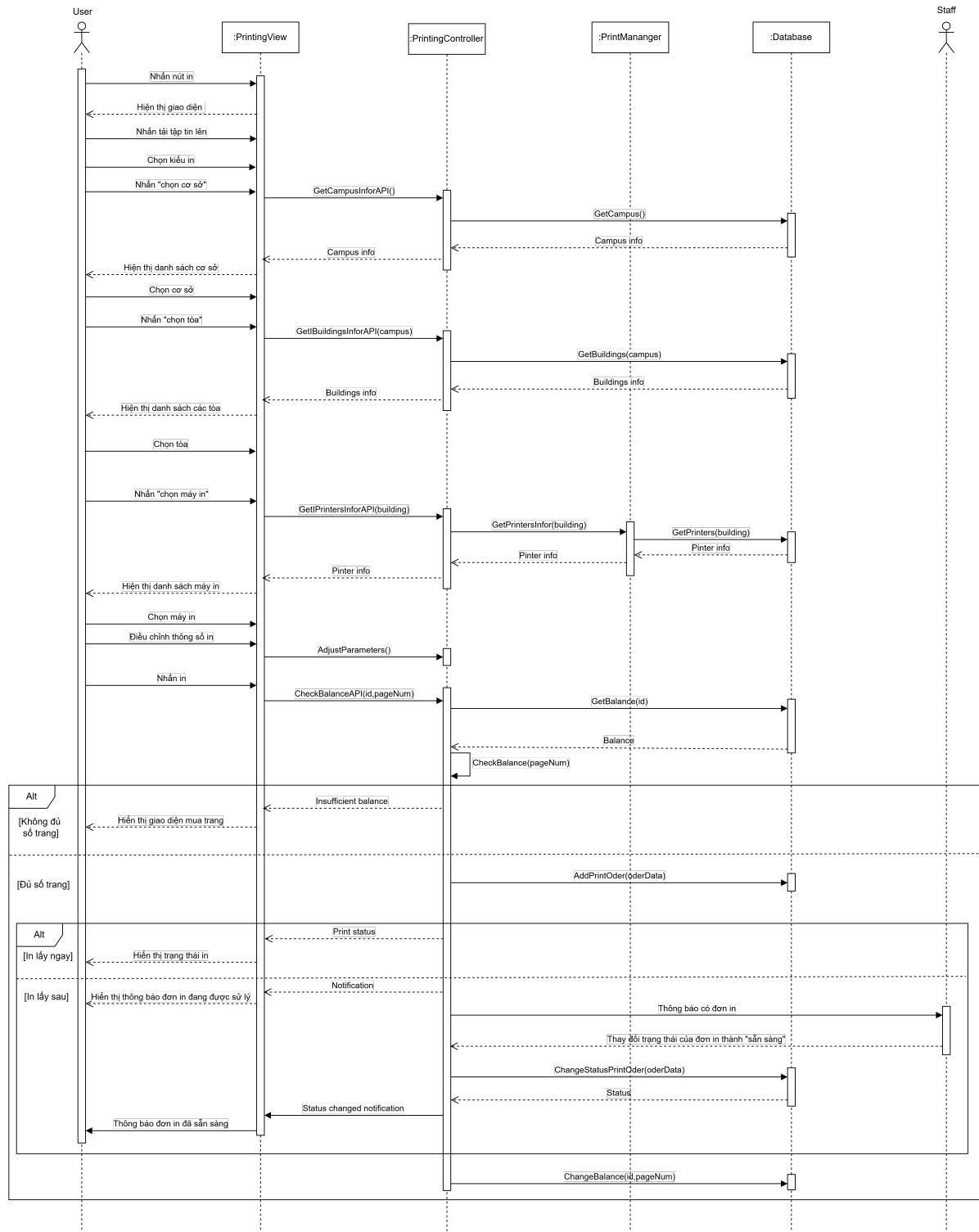
Hình 6: Activity diagram cho chức năng mua trang in

- Các đối tượng trong biểu đồ:
 - **User:** Giảng viên hoặc sinh viên.
 - **SPSO:** Bộ phận kỹ thuật
- Tổng quan quy trình:
 - Bắt đầu quy trình, người dùng đăng nhập và được xác thực thành công bởi hệ thống.
 - Chọn vào mua trang in. Sau đó chọn loại giấy mà người dùng muốn in tài liệu lên.
 - Điền số lượng mong muốn cho loại giấy và xác nhận. Nếu muốn thay đổi về loại giấy thì không xác nhận, sau khi đã xác nhận thành công thì chuyển tới thanh toán.
 - Thanh toán hoàn tất khi hệ thống kiểm tra số dư của bạn đủ để thanh toán cho việc mua trang in, nếu không sẽ thông báo không đủ số dư.
 - Khi đã đủ số dư và hoàn tất thanh toán, SPSO sẽ cung cấp trang in cho người dùng trên hệ thống để họ tiến hành in tài liệu, kết thúc quy trình.

2.2 Sequence Diagram

2.2.1 Sequence diagram cho chức năng đặt in tài liệu

- Các đối tượng trong biểu đồ:
 - **User:** Giảng viên hoặc sinh viên.



Hình 7: Sequence diagram cho chức năng đặt in tài liệu

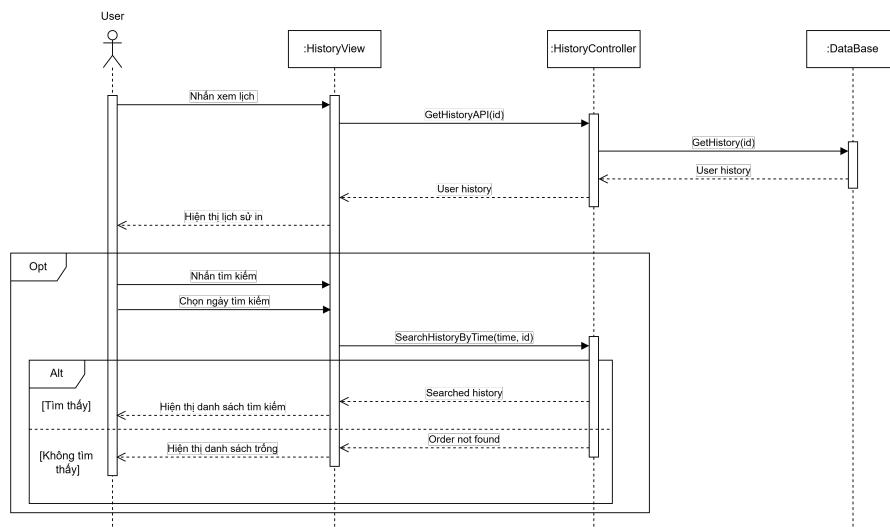


- **PrintingView:** Giao diện mà người dùng tương tác để bắt đầu và quản lý quá trình in ấn.
- **PrintingController:** Bộ điều khiển xử lý các yêu cầu từ người dùng và kết nối giữa giao diện và các thành phần xử lý phía sau.
- **PrintManager:** Quản lý logic nghiệp vụ liên quan đến việc in ấn, như truy xuất dữ liệu hoặc xử lý yêu cầu in.
- **Database:** Cơ sở dữ liệu lưu trữ thông tin về cơ sở, tòa nhà, máy in, số dư tài khoản, và các đơn đặt hàng in.
- **Staff:** Nhân viên quản lý việc in lấp sau.

- Tổng quan quy trình:

- Người dùng bắt đầu bằng cách tương tác với PrintingView, nơi giao diện để chọn cơ sở và máy in được hiển thị.
- Người dùng tải tài liệu lên hệ thống, sau đó bắt đầu chọn cơ sở in PrintingView sẽ gọi API tới PrintingController (GetCampusInfoAPI()), PrintingController sẽ gọi tới DataBase (GetCampus()) để trả về danh sách các cơ sở cho người dùng.
- Sau khi chọn cơ sở, người dùng chọn tòa nhà theo cơ sở (GetBuildings(campus)). PrintingController tiếp tục lấy danh sách tòa nhà thông qua Database và trả về danh sách các tòa.
- Sau khi chọn tòa, người dùng chọn máy in theo tòa (GetPrintersInfor(building)). PrintingController sẽ lấy danh sách máy nhà thông qua PrintManager và PrintManager sẽ trả về danh sách các máy in từ Database.
- Người dùng điều chỉnh các thông số (AdjustParameters()) và nhấn nút "In". Hệ thống kiểm tra số dư của người dùng trước khi in (CheckBalance(pageNum)).
 - * nếu số dư không đủ, giao diện mua trang in sẽ được hiển thị.
 - * Nếu số dư đủ, đơn đặt hàng in sẽ được lưu vào Database (AddPrintOder(oderData)).
- Hệ thống gửi thông báo cho người dùng về trạng thái đơn in cho đến khi hoàn tất (in lấp ngay), hoặc sẽ hiển thị thông báo đơn in đang được xử lý và người dùng sẽ nhận được thông báo sau khi nhân viên cập nhật trạng thái của đơn lên hệ thống (in lấp sau).
- Khi đơn in hoàn tất số dư của người dùng sẽ được cập nhật.

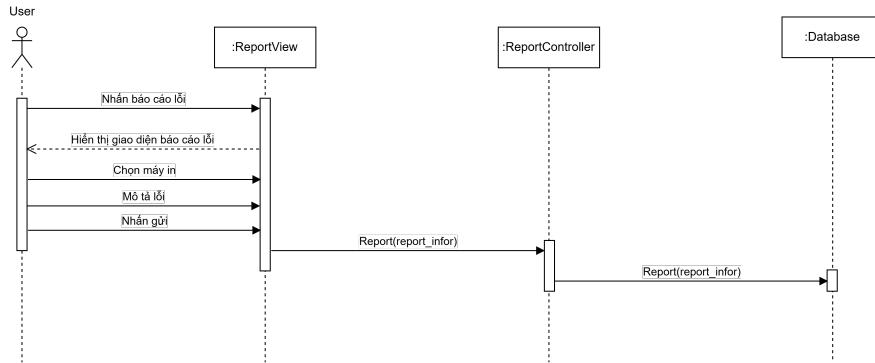
2.2.2 Sequence diagram cho chứng năng xem lịch sử in



Hình 8: Sequence diagram cho chứng năng xem lịch sử in

- Các đối tượng trong biểu đồ:
 - **User:** Giảng viên hoặc sinh viên.
 - **HistoryView:** Giao diện mà người dùng tương tác để bắt đầu và xem lịch sử in.
 - **HistoryController:** Bộ điều khiển xử lý các yêu cầu từ người dùng và kết nối giữa giao diện và các thành phần xử lý phía sau.
 - **Database:** Cơ sở dữ liệu lưu trữ thông tin về cơ sở, tòa nhà, máy in, số dư tài khoản, và các đơn đặt hàng in.
- Tổng quan quy trình:
 - Người dùng bắt đầu bằng cách tương tác với HistoryView, nơi giao diện để xem lịch sử in và các tùy chọn.
 - Sau khi nhấn vào xem lịch sử in HistoryView sẽ gọi API đến HistoryController (GetHistoryAPI(id)), sau đó HistoryController sẽ lấy lịch sử từ DataBase (GetHistory(id)) và trả về cho người dùng
 - Sau khi có được lịch sử in người dùng có thể lựa chọn việc lọc danh sách theo ngày in (SearchHistoryByTime(time, id))
 - * Nếu có đơn in trong ngày đã chọn thì hệ thống sẽ trả về danh sách in trong ngày đấy.
 - * Nếu có đơn in trong ngày đã chọn thì hệ thống sẽ trả về danh sách trống.

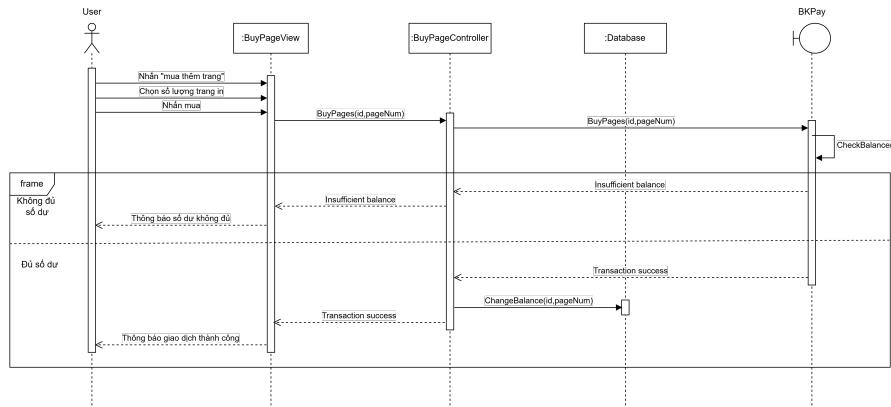
2.2.3 Sequence diagram cho chứng năng báo cáo sự cố



Hình 9: Sequence diagram cho chứng năng báo cáo sự cố

- Các đối tượng trong biểu đồ:
 - **User:** Giảng viên hoặc sinh viên.
 - **ReportView:** Giao diện mà người dùng tương tác để bắt đầu và báo cáo sự cố.
 - **ReportController:** Bộ điều khiển xử lý các yêu cầu từ người dùng và kết nối giữa giao diện và các thành phần xử lý phía sau.
 - **Database:** Cơ sở dữ liệu lưu trữ thông tin về cơ sở, tòa nhà, máy in, số dư tài khoản, và các đơn đặt hàng in.
- Tổng quan quy trình:
 - Người dùng bắt đầu bằng cách tương tác với ReportView, nơi giao diện để thực hiện báo cáo sự cố.
 - Sau đó người dùng có thể thực hiện chọn máy in và thực hiện mô tả sự cố trên máy in đó và gửi báo cáo đến ReportController (Report(report-infor)) sau đó ReportController sẽ gửi báo cáo về DataBase.

2.2.4 Sequence diagram cho chức năng mua trang in

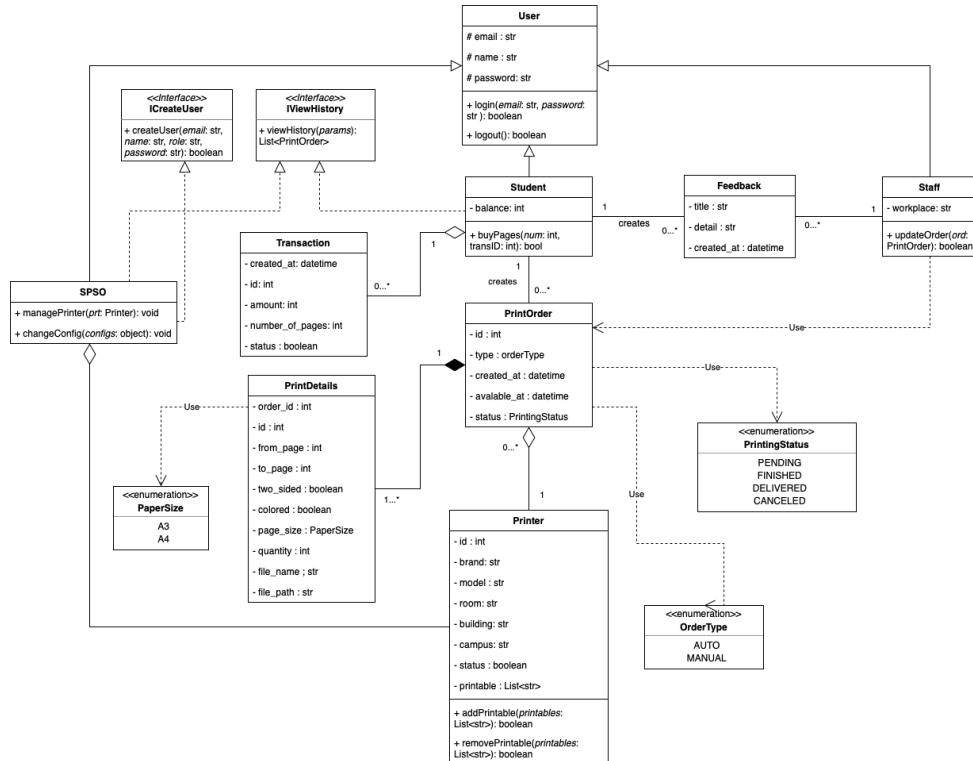


Hình 10: Sequence diagram cho chứng năng mua trang in

- Các đối tượng trong biểu đồ:

- **User:** Giảng viên hoặc sinh viên.
 - **BuyPageView:** Giao diện mà người dùng tương tác để bắt đầu và mua trang in.
 - **BuyPageController:** Bộ điều khiển xử lý các yêu cầu từ người dùng và kết nối giữa giao diện và các thành phần xử lý phía sau.
 - **Database:** Cơ sở dữ liệu lưu trữ thông tin về cơ sở, tòa nhà, máy in, số dư tài khoản, và các đơn đặt hàng in.
 - **BKPay:** hệ thống thanh toán của trường Đại Học Bách Khoa TPHCM.
- Tổng quan quy trình:
 - Người dùng bắt đầu bằng cách tương tác với ReportView, nơi giao diện để thực hiện mua trang in.
 - Sau đó người dùng có thể thực hiện chọn số trang muốn mua và bắt đầu nhán mua (BuyPages(id,pageNum)), BuyPageController sẽ thực hiện gọi tới hệ thống BKPay để thực hiện giao dịch.
 - * Nếu số dư trong BKPay không đủ hệ thống sẽ trả về thông báo không đủ số dư.
 - * Nếu số dư trong BKPay đủ thì giao dịch sẽ được diễn ra và trả về thông báo giao dịch thành công.

2.3 Class diagram



Hình 11: Class diagram cho module in ấn

1. **User:** là lớp cha, kế thừa ra các lớp con là Nhân viên và Sinh viên. SPSO sẽ là base user. Lớp này có các thuộc tính như:



- email: email của người dùng, dùng để đăng nhập vào hệ thống. Có thể sử dụng email hcmut, email của người dùng phải là duy nhất
- name: họ và tên của người dùng
- password: mật khẩu của người dùng: lưu trong cơ sở dữ liệu dưới dạng mã hash

Lớp này có các phương thức như:

- login(email, password) nhận vào email và mật khẩu của người dùng để đăng nhập người dùng vào hệ thống. Nếu người dùng đăng nhập bằng google, chỉ nhận vào email
- logout(): đăng xuất khỏi hệ thống

2. **SPSO**: Là lớp con kế thừa từ Người dùng, có các phương thức như:

- managePrinter(printer: Printer): cài đặt các thông tin của máy in
- changeConfig(configs: object): thay đổi các cài đặt của hệ thống
- viewHistory(params): hiện thực interface IViewHistory, xem toàn bộ lịch sử in của hệ thống lọc theo các thông số *params*
- createUser(email, name, password, role): hiện thực interface ICreateUser, tạo một người dùng mới cho hệ thống

3. **Staff**: là lớp con kế thừa từ Người dùng, có các thuộc tính kế thừa từ người dùng, và một thuộc tính thêm là work_at: nơi làm việc của nhân viên đó.

4. **Student**: là lớp con kế thừa từ Người dùng, có các thuộc tính của người dùng, tuy nhiên có thêm các thuộc tính như:

- balance: số trang in của người dùng hiện có

Và các phương thức:

- buyPages(num, transId): nhận vào số trang in cần mua và mã giao dịch, nếu giao dịch hợp lệ, tăng cho người dùng lượng trang in tương ứng
- viewHistory(params): hiện thực interface IViewHistory, xem toàn bộ lịch sử in của bản thân, lọc theo các điều kiện *params*

5. **Transaction**: là một giao dịch của người dùng, được ghi lại sau khi người dùng thực hiện thanh toán cho trang in

- created_at: thời điểm tạo đơn
- id: mã giao dịch
- amount: lượng tiền giao dịch
- number_of_pages: số trang
- status: tình trạng giao dịch

6. **Feedback**: là một báo cáo lỗi của người dùng lên hệ thống, một feedback cần có:

- title: chủ đề của feedback
- detail: nội dung cần feedback
- created_at: thời điểm feedback được tạo



7. **PrintOrder:** là các đơn in của khách hàng trong hệ thống, có các thuộc tính như:

- id: mã đơn in
- type: loại đơn in. Đơn in của khách hàng có thể là đơn in lấy ngay hoặc đơn in lấy sau tại quầy
- status: tình trạng của đơn in. Tình trạng của đơn in có thể là đang in, đã hoàn thành, đã giao hoặc đã bị huỷ
- created_at: thời điểm đơn in được tạo ra
- available_at: thời điểm đơn in sẵn sàng cho người dùng. (với đơn in lấy sau tại quầy, thời điểm này do người dùng chọn)

8. **Printer:** là các máy in trong hệ thống, có các thuộc tính sau:

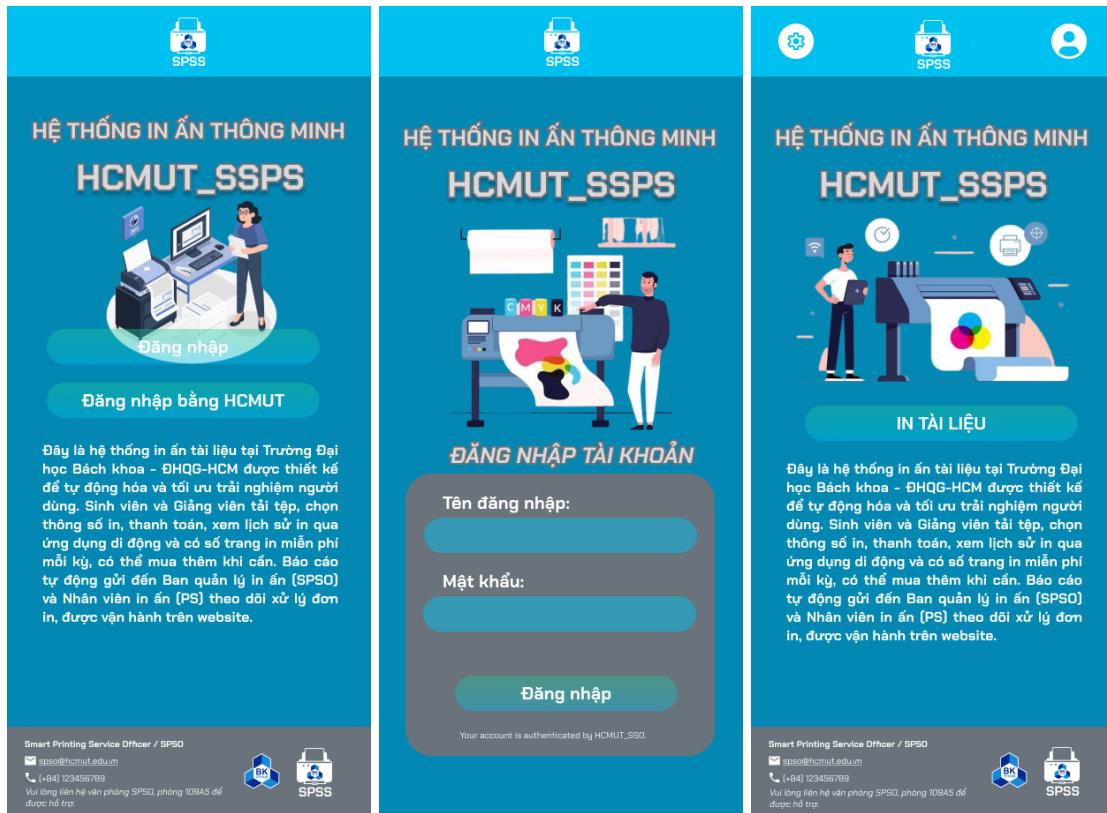
- id : mã máy in
- brand : hãng máy in
- model : mẫu mã của máy in
- room : phòng nơi máy in được đặt
- building : tòa nhà chứa máy in
- campus : cơ sở chứa máy in
- status : tình trạng hoạt động của máy in
- printables : danh sách các loại file máy in có thể in

9. **PrintDetails :** là các thông tin chi tiết để in một file, có các thuộc tính sau:

- order_id : mã đơn in
- id : mã của PrintDetails, đi kèm với order_id tạo ra khoá chính để định danh các printDetails trong database
- from_page : trang bắt đầu in
- to_page : trang kết thúc in
- two_sided : true nếu chọn in 2 mặt
- colored : true nếu chọn in màu
- page_size : kích thước giấy in
- quantity : số bản in
- file_name : tên file in
- file_path : đường dẫn tới file in trong hệ thống

2.4 MVP 1 (User interface)

Hệ thống in ấn bao gồm hai giao diện: Giao diện App mobile cho người dùng (Sinh viên/ Giảng viên) in tài liệu và Giao diện Website cho ban quản lý SPSO, nhân viên in ấn PS.



Hình 12: Giao diện đăng nhập qua App của người dùng, lần lượt từ trái qua phải:
13.a Trang chủ trước đăng nhập; 13.b Đăng nhập; 13.c Trang chủ sau đăng nhập.



Hình 13: Giao diện đăng nhập qua Website cho PS và SPSO, lần lượt từ trái qua phải:
14.a Trang chủ trước khi đăng nhập; 14.b Chọn vai trò.



Hình 14: Giao diện đăng nhập qua Website của PS và SPSO, lần lượt từ trái qua phải:
15.a Đăng nhập; 15.b Trang chủ sau đăng nhập của PS; 15.c Trang chủ sau đăng nhập của SPSO.

2.4.1 Giao diện đăng nhập

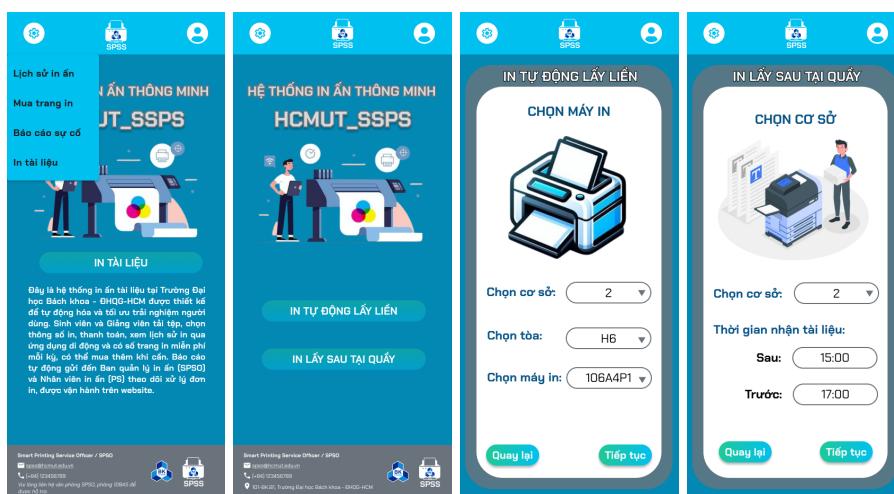
1. Giao diện đăng nhập qua App của người dùng;
2. Giao diện đăng nhập qua Website của nhân viên in ấn PSO, ban quản lý SPSO:

2.4.2 User interface cho chức năng in tài liệu

Hệ thống in ấn thông minh cung cấp hai dịch vụ in ấn cho người dùng:

- In tự động lấy liền: Người dùng tạo đơn in qua app và lấy tài liệu in tại các máy in tự động, không cần nhân viên hỗ trợ.
- In lấy sau tại quầy: Người dùng tạo đơn in qua app, chọn các thuộc tính cũng như thời gian nhận tài liệu. Nhân viên in ấn - PSO phụ trách nhận đơn in, in tài liệu và gửi thông báo in xong đến sinh viên qua hệ thống.

1. Giao diện in của người dùng;



Hình 15: Giao diện in ấn của người dùng:

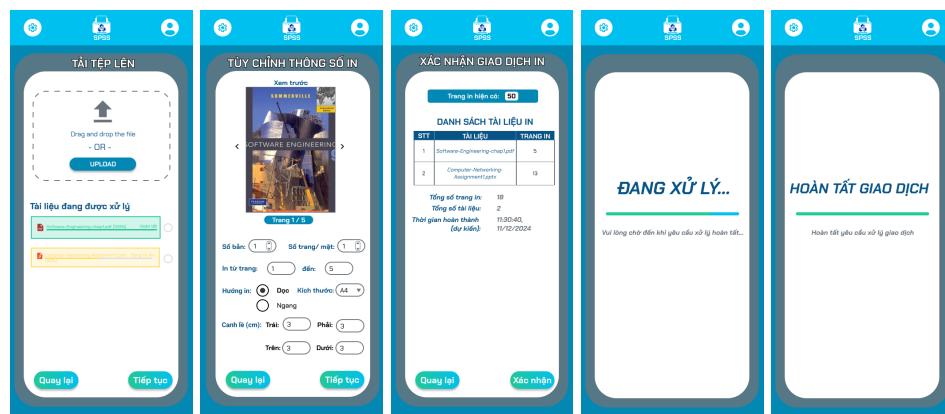
- 16.a Trang chủ sau đăng nhập với Menu;
- 16.b In tài liệu; 16.c In tự động lấy liền; 16.d In lấy sau tại quầy.

- Ở trang chủ (hình 16.a), người dùng chọn nút *In tài liệu* trong trang chủ hoặc nhấn vào *Biểu tượng Menu* góc trên trái màn hình để mở Menu, sau đó chọn *In tài liệu* để mở giao diện IN TÀI LIỆU (hình 16.b).

- Người dùng lựa chọn *IN TỰ ĐỘNG LẤY LIỀN* được điều hướng đến trang IN TỰ ĐỘNG LẤY LIỀN (hình 16.c), chọn *IN LẤY SAU TẠI QUẦY* được điều hướng đến trang IN LẤY SAU TẠI QUẦY (hình 16.d).

- Tại trang IN TỰ ĐỘNG LẤY LIỀN, sinh viên lựa chọn **Cơ sở**, **Tòa**, **Máy in** và nhấn tiếp tục để chuyển đến trang TẢI TẬP LÊN.

- Tại trang IN LẤY SAU TẠI QUẦY, sinh viên lựa chọn **Cơ sở**, **Thời gian nhận tài liệu** và nhấn tiếp tục để chuyển đến trang TẢI TẬP LÊN.



Hình 16: Giao diện in ấn của người dùng:

- 17.a Tải tệp lên; 17.b Tùy chỉnh thông số; 17.c Xác nhận giao dịch;
17.d Đơn in được xử lý; 17.e Hoàn tất xử lý.

- Tại trang TẢI TỆP LÊN, sinh viên nhấn nút *UPLOAD* để tải tài liệu cần in lên hệ thống và nhấn *Tiếp tục* để đến trang TÙY CHỈNH THÔNG SỐ hoặc nhấn *Quay lại* để về trang trước đó.
- Tại trang TÙY CHỈNH THÔNG SỐ, sinh viên lựa chọn các thông số phù hợp và nhấn *Tiếp tục* để điều hướng đến trang XÁC NHẬN GIAO DỊCH IN.
- Sau khi kiểm tra danh sách tài liệu, tổng số trang in, sinh viên nhấn *Xác nhận* để in.
- Hình 17.d, 17.e là giao diện xử lý.

2. Giao diện xác nhận đơn in lấy sau tại quầy của nhân viên in ấn PS:



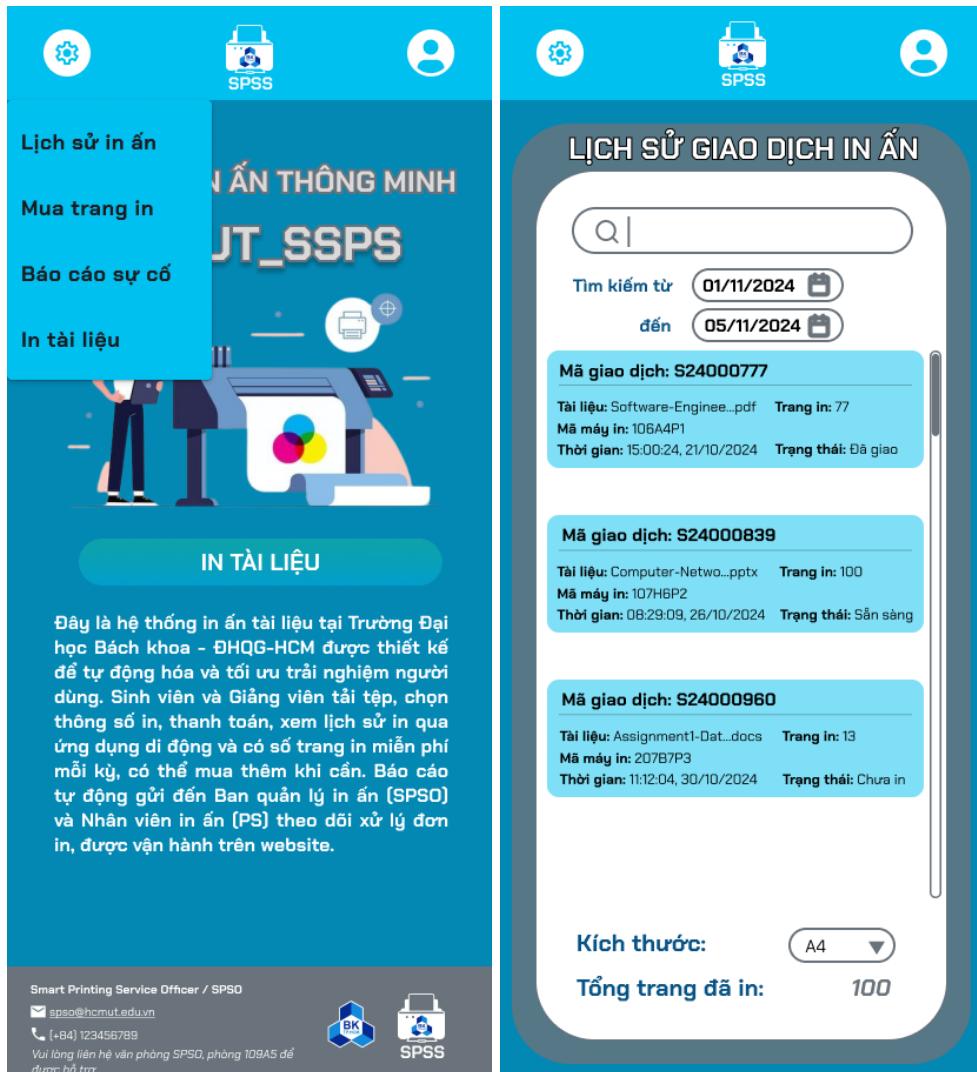
Hình 17: Giao diện đăng nhập qua Website cho nhân viên in ấn (PS), lần lượt từ trái qua phải:
18.a Trang chủ sau đăng nhập; 18.b Quản lý đơn in lấy sau tại quầy của PS.

- Sau khi đăng nhập thành công, nhân viên in PS có thể lựa chọn nhấp vào nút QUẢN LÝ ĐƠN IN để mở trang QUẢN LÝ ĐƠN IN (hình 18.b).
- Tại trang QUẢN LÝ ĐƠN IN, danh sách các yêu cầu in lấy sau của người dùng được cập nhật liên tục. Nhân viên in PS có thể coi thông tin đơn in cũng như cập nhật trạng thái đơn in tại mục *Trạng thái*:

- Chưa in: Trạng thái mặc định khi một yêu cầu in ấn được gửi.
- Sẵn sàng: Nhân viên in đã in xong và cập nhật trạng thái này, hệ thống sẽ gửi thông báo đến người dùng.
- Đã giao: Sau khi tài liệu in được giao đến người dùng, nhân viên in PS cập nhật trạng thái đã giao để thuận tiện quản lý các tài liệu in tại quầy.

2.4.3 User interface cho chức năng xem lịch sử in

1. Giao diện xem lịch sử in ấn của người dùng qua App:



Hình 18: Giao diện Xem lịch sử in ấn của người dùng:
19.a Trang chủ sau đăng nhập với Menu; 19.b Lịch sử giao dịch in ấn.

- Ở trang chủ (hình 19.a), người dùng nhấn vào *Biểu tượng Menu* góc trên bên trái màn hình để mở Menu, sau đó chọn *Lịch sử in ấn* để mở giao diện LỊCH SỬ GIAO DỊCH IN ẤN (hình 19.b).
- Tại giao diện LỊCH SỬ GIAO DỊCH IN ẤN, sinh viên coi lịch sử in ấn đã thực hiện, có chức năng tìm kiếm theo ngày và xem tổng trang đã in của từng kích thước.

2. Giao diện xem lịch sử in ấn của SPSO qua website:



The screenshot shows the 'HỆ THỐNG IN ẤN THÔNG MINH HCMUT_SSPO' interface. The 'LỊCH SỬ IN ẤN' tab is selected. It displays a table of print jobs with columns: MÃ BẢN IN, MÃ MÁY IN, TÀI KHOẢN, KIỂU IN, TRANG IN, TÀI LIỆU, THỜI GIAN, and Trạng thái. One job is highlighted with a red border.

MÃ BẢN IN	MÃ MÁY IN	TÀI KHOẢN	KIỂU IN	TRANG IN	TÀI LIỆU	THỜI GIAN	Trạng thái
S24000777	1074892	Trần Minh Hiếu Heuremn009	In tài liệu	77	Software-Engineering-chap1.pdf	15/01/2024 21/01/2024	Đã giao
T24004889	1074892	Phạm Bảo Khanh khangpham001	Inilly sau	13	Computer-Networking-chap1.pdf	17/01/2024 21/01/2024	Đã giao
T24002204	2078797	Lê Thị Huyền Vy hue.tiem99	Inilly liên	20	Data-Engineering-Assignment1.docx	09/01/2024 13/01/2024	Gửi sang
S24000022	2078793	Đinh Minh Hiếu hienuyduy	Inilly liên	100	Data-Engineering-Project.xlsx	11/01/2024 12/01/2024	Chưa in

Hình 19: Giao diện Xem lịch sử in ấn của SPSO:

20.a Trang chủ sau đăng nhập của SPSO; 20.b Lịch sử in ấn: Giao diện quản lý đơn in của SPSO.

- Sau khi đăng nhập thành công (hình 22.a), SPSO nhấp nút *LỊCH SỬ IN ẤN* để mở trang *LỊCH SỬ IN ẤN* (hình 20.b).
- Tại trang *LỊCH SỬ IN ẤN*, SPSO coi các đơn in trên hệ thống theo thứ tự thời gian, theo mã máy in, theo user và theo kiểu in.
- SPSO có thể lựa chọn nhấp vào các nút sau:
 - Thanh tìm kiếm: xem lịch sử in ấn của một tài khoản cụ thể.
 - Thời gian: xem lịch sử in ấn trong khoảng thời gian cụ thể.
 - Mã máy in: xem lịch sử in ấn của một máy in cụ thể.
 - Kiểu in: xem lịch sử in ấn theo kiểu in.

2.4.4 User interface cho chức năng báo cáo sự cố

1. Người dùng báo cáo sự cố qua App

The screenshots show the mobile application's user interface for reporting incidents. The first screen shows the main menu with options like 'Lịch sử in ấn', 'Mua trang in', 'Báo cáo sự cố', and 'In tài liệu'. The second screen shows the 'BÁO CÁO SỰ CỐ' (Report Incident) form with fields for 'Loại sự cố' (Incident Type) set to 'Lỗi máy in', 'Mô tả cụ thể' (Description), and a 'Báo cáo' (Report) button. The third screen shows the 'HOÀN TẤT BÁO CÁO' (Report Completed) confirmation message: 'Báo cáo sự cố đã được gửi đi.'

Hình 20: Giao diện Báo cáo sự cố của người dùng:

21.a Trang chủ sau đăng nhập với Menu; 21.b Báo cáo sự cố; 21.c Xử lý gửi báo cáo sự cố.

- Ở trang chủ (hình 21.a), người dùng nhấp vào *Biểu tượng Menu* góc trên bên trái màn hình để mở Menu, sau đó chọn *Báo cáo sự cố* để mở giao diện *BÁO CÁO SỰ CỐ* (hình 21.b).



- Ở trang BÁO CÁO SỰ CỐ, người dùng coi lịch sử báo cáo, phản hồi của nhân viên đối với mỗi báo cáo. Có thể thực hiện báo cáo sự cố: Chọn loại sự cố, mô tả cụ thể và gửi báo cáo.
- Hình 21.c là giao diện hệ thống gửi yêu cầu báo cáo sự cố.

2. Giao diện xử lý báo cáo sự cố của SPSO:

The screenshot shows two parts of the SPSO website. On the left is the main landing page for 'HỆ THỐNG IN ẤN THÔNG MINH HCMUT_SSPO'. It features a banner for 'LỊCH SỬ IN ẤN' and 'LỊCH SỬ MUA TRANG IN'. On the right is a detailed view of the 'XỬ LÝ BÁO CÁO SỰ CỐ' section. This view includes a table of reported incidents with columns for 'Loại sự cố', 'Mô tả cụ thể', 'Thời gian', and 'Trạng thái'. Each row shows details like 'Lỗi máy in', 'Lỗi in ấn', 'Lỗi hệ thống', and 'Lỗi máy in' with their respective status (Xong or Chờ).

Hình 21: Giao diện xử lý báo cáo sự cố qua Website cho ban quản lý (SPSO):

22.a Trang chủ sau đăng nhập (SPSO); 22.b Xử lý báo cáo sự cố (SPSO).

- Sau khi đăng nhập thành công, ban quản lý (SPSO) có thể lựa chọn nhấp vào nút *XỬ LÝ BÁO CÁO SỰ CỐ* để mở trang XỬ LÝ BÁO CÁO SỰ CỐ (hình 22.b).
- Tại trang XỬ LÝ BÁO CÁO SỰ CỐ, danh sách các báo cáo sự cố về lỗi in ấn, hệ thống, máy in từ người dùng, SPSO có theo dõi, xử lý và cập nhật các lỗi này tại mục *Trạng thái*:

- Chưa: Trạng thái lỗi chưa được xử lý.
- Xong: Đã xử lý xong lỗi.

2.4.5 User interface cho chức năng xem mua trang in

1. Giao diện mua trang in của người dùng

The four screenshots show the process of buying additional pages.
 1. The first screen shows the main menu with options like 'Lịch sử in ấn', 'Mua trang in', 'Báo cáo sự cố', and 'In tài liệu'.
 2. The second screen shows the 'LỊCH SỬ MUA TRANG IN' (Purchase History) with a table of previous purchases.
 3. The third screen shows the 'MUA THÊM TRANG IN' (Buy Additional Pages) form where users can enter the number of pages needed (20), payment amount (500 VNĐ), and password (An).
 4. The fourth screen shows the 'XỬ LÝ ĐƠN MUA TRANG IN' (Handle Purchase Order) with a note to check if payment is complete.

Hình 22: Giao diện Mua thêm trang in của người dùng:

- 23.a Trang chủ sau đăng nhập với Menu;
- 23.b Quản lý trang in của người dùng;
- 23.c Mua thêm trang in;
- 23.d Xử lý giao dịch.



- Ở trang chủ (hình 23.a), người dùng nhấn vào *Biểu tượng Menu* góc trên bên trái màn hình để mở Menu, sau đó chọn *Quản lý trang in* để mở giao diện QUẢN LÝ TRANG IN (hình 23.b).
- Ở giao diện QUẢN LÝ TRANG IN, sinh viên coi lịch sử mua trang in cũng như số trang in hiện có.
- Người dùng nhấn vào nút *Mua thêm trang in* để mua thêm trang in. - Tại giao diện MUA THÊM TRANG IN (hình 23.c), nhập số trang in cần mua, xác nhận mật khẩu và chọn *Tiếp tục* để hoàn tất mua. - Hình 23.d là màn hình xử lý giao dịch của hệ thống.

2. Giao diện xem lịch sử mua trang in của SPSO:

The screenshot shows two parts of the SPSO system interface. On the left is the main dashboard with tabs for 'LỊCH SỬ IN ẨN' (Hidden Print History), 'LỊCH SỬ MUA TRANG IN' (Print Page Purchase History), and 'XỬ LÝ BÁO CÁO SỰ CỐ' (Report Error Handling). The central part displays a table titled 'LỊCH SỬ MUA TRANG' (Print Page Purchase History) with columns for 'MÃ GIAO DỊCH' (Order ID), 'TÀI KHOẢN' (Account), 'TRANG IN' (Print Pages), 'THỜI GIAN' (Time), and 'Thanh toán' (Payment Status). Three entries are listed: PNH24957 (77 pages, payment pending), PNH24875 (13 pages, payment pending), and PNH25088 (20 pages, payment pending). A search bar at the top allows users to search by time period.

Hình 23: Giao diện xem Lịch sử mua thêm trang in của SPSO:
24.a Trang chủ sau đăng nhập (SPSO); 24.b Lịch sử mua trang in.

- Tại trang Trang chủ sau đăng nhập (SPSO) (hình 24.a), SPSO có thể nhấn vào nút *Lịch sử mua trang in* để coi LỊCH SỬ MUA TRANG (hình 24.b).

2.4.6 User interface cho chức năng quản lý máy in của ban quản lý SPSO

The screenshot shows two parts of the SPSO system interface. On the left is the main dashboard with tabs for 'LỊCH SỬ IN ẨN' (Hidden Print History), 'LỊCH SỬ MUA TRANG IN' (Print Page Purchase History), and 'THÔNG TIN MÁY IN' (Printer Information). The central part displays a table titled 'THÔNG TIN MÁY IN' (Printer Information) with columns for 'MÃ MÁY IN' (Printer ID), 'MÃ SP' (Model ID), 'TÊN MÁY IN' (Printer Name), 'VỊ TRÍ' (Location), 'HÀNG' (Brand), 'LOẠI MÁY IN' (Type), 'TỐC ĐỘ IN' (Print Speed), 'KẾT NỐI' (Connection), and 'Trạng thái' (Status). Four printers are listed: 10544P1 (HP LaserJet Pro M404dn), 107H9P2 (Brother HL-L2370DW), 20B1H97 (Canon CL40S LBP6030w), and 20797P3 (Xerox Phaser 3300DN). A dropdown menu for selecting the printer to manage is shown at the top left, and a 'Cập nhật máy in' (Update printer) button is at the top right.

Hình 24: Giao diện quản lý máy in của SPSO:
25.a Trang chủ sau đăng nhập (SPSO); 25.b Thông tin máy in.

- Tại trang Trang chủ sau đăng nhập (SPSO) (hình 25.a), SPSO có thể nhấn vào nút có biểu tượng bánh răng để xem THÔNG TIN MÁY IN (hình 25.b). - Tại trang THÔNG TIN MÁY IN, ta có thể ấn vào các nút:

- + Nút mã máy in để tìm kiếm máy in bằng mã nhanh chóng.
- + Nút "Cập nhật máy in" để chỉnh sửa thông tin hoặc cập nhật tình trạng máy in.
- + Biểu tượng "+" để thêm máy in.
- + Biểu tượng "thùng rác" để xóa máy in.

3 Architectural design - Thiết kế kiến trúc hệ thống

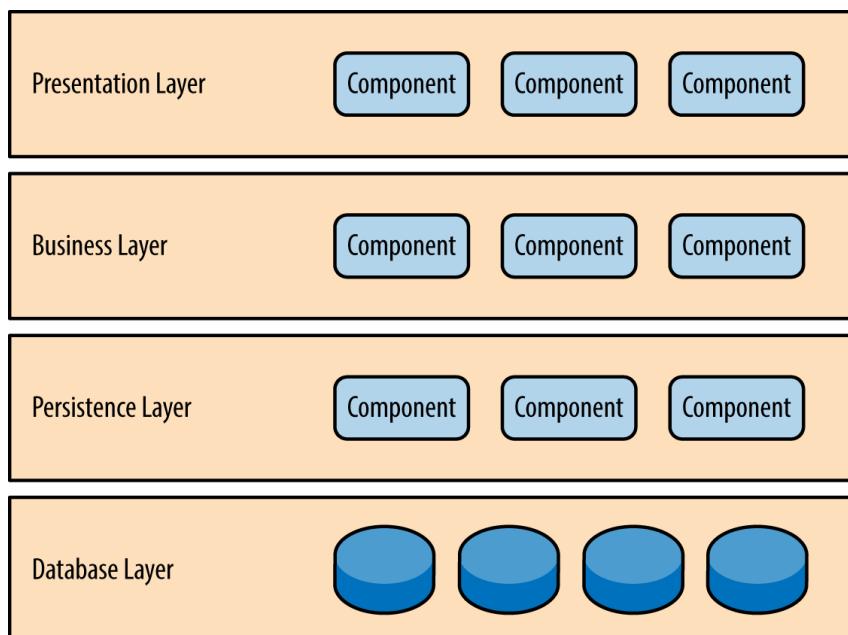
3.1 Kiến trúc phân tầng

3.1.1 Xác định kiến trúc cho hệ thống.

Trong quá trình thiết kế hệ thống HCMUT-SSPS, một nền tảng quản lý in ấn từ xa dành cho sinh viên và giảng viên, chúng tôi đã xác định một số đặc điểm quan trọng liên quan đến hệ thống như sau:

- Hệ thống in ấn từ xa cho phép người dùng đặt lịch in và gửi tài liệu từ xa, hỗ trợ quá trình in ấn tiện lợi mà không cần phải có mặt trực tiếp tại địa điểm in.
- Đây là một hệ thống đơn giản với chức năng chính là quản lý in ấn, nhưng cần tích hợp thêm các tính năng đăng nhập SSO để người dùng có thể sử dụng tài khoản trường học, cũng như các chức năng thanh toán để quản lý phí in ấn một cách linh hoạt.
- Hệ thống được thiết kế để phục vụ nhiều nhóm người dùng, bao gồm quản lý và khách hàng (sinh viên, giảng viên). Vì vậy, giao diện tương tác giữa các nhóm này sẽ khác nhau tùy thuộc vào nhu cầu sử dụng: nhóm quản lý sẽ theo dõi và xử lý các yêu cầu in ấn, trong khi nhóm khách hàng sẽ sử dụng hệ thống để đặt in.

Với những đặc điểm trên, kiến trúc phân tầng (layered architecture) là lựa chọn phù hợp nhất. Kiến trúc phân tầng (Layered Architecture) là một mô hình thiết kế hệ thống phần mềm phổ biến, trong đó hệ thống được chia thành các tầng (layers) riêng biệt, mỗi tầng có nhiệm vụ và trách nhiệm cụ thể. Các tầng này hoạt động độc lập và tương tác với nhau thông qua các giao diện đã được định nghĩa. Mục tiêu của kiến trúc này là đảm bảo sự phân tách giữa các thành phần của hệ thống, giúp quản lý dễ dàng hơn, tăng tính bảo trì và mở rộng trong tương lai.



Hình 25: Minh họa một mẫu Layered Architecture

Mô hình này cho phép hệ thống được chia thành các tầng riêng biệt, đồng thời quản lý giao diện tương tác giữa các nhóm người dùng, giúp tối ưu hóa trải nghiệm của cả quản lý và khách hàng với các chức năng riêng biệt. Với hai nhóm người dùng chính là quản lý và khách hàng, hệ thống cần có cơ chế



quản lý vai trò (role management) rõ ràng để xác định quyền hạn của từng nhóm. Nhóm quản lý có thể kiểm soát quá trình in ấn, theo dõi yêu cầu từ sinh viên và giảng viên, trong khi nhóm khách hàng chỉ có quyền đặt lệnh in và theo dõi tình trạng của tài liệu đã gửi. Việc phân quyền rõ ràng giúp giảm thiểu các sai sót trong vận hành và tăng tính minh bạch của hệ thống.

Ngoài ra, Để đảm bảo hệ thống có thể phát triển theo thời gian, chúng tôi dự định sẽ tích hợp với các dịch vụ bên thứ ba như hệ thống thanh toán trực tuyến, dịch vụ lưu trữ đám mây, và hệ thống đăng nhập SSO (Single Sign-On). Điều này sẽ giúp người dùng dễ dàng thanh toán phí in ấn, truy cập các tài liệu của họ từ nhiều nguồn khác nhau, và sử dụng tài khoản duy nhất để đăng nhập vào hệ thống.

Việc sử dụng kiến trúc phân tầng giúp hệ thống dễ bảo trì, tối ưu hóa chi phí triển khai, và có khả năng mở rộng linh hoạt về mặt quy mô người dùng và dữ liệu mà không cần phải thay đổi cấu trúc tổng thể. Điều này đặc biệt quan trọng đối với một hệ thống phục vụ cả khách hàng và quản lý, nơi nhu cầu sử dụng có thể tăng cao theo thời gian.

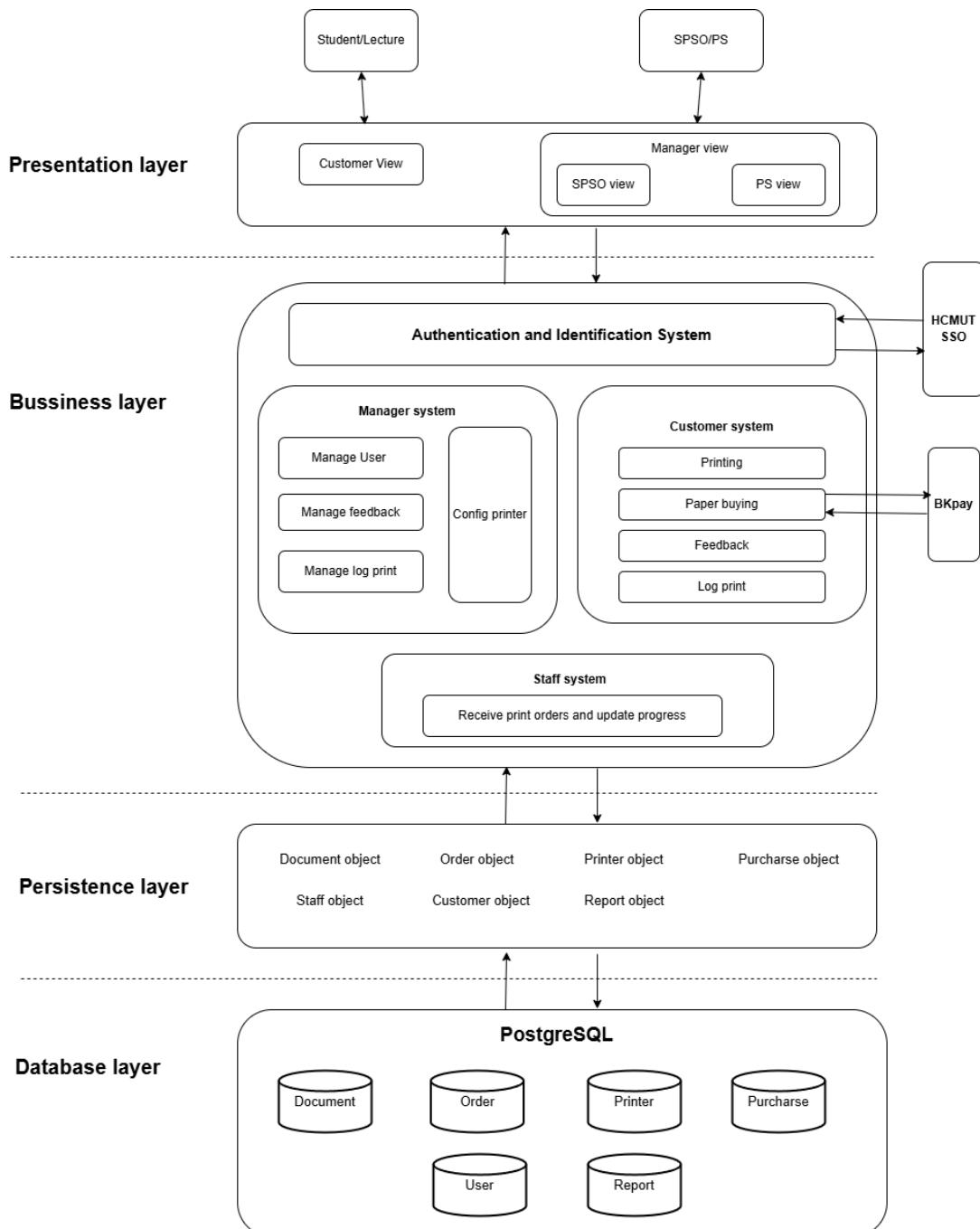
Tóm lại, việc sử dụng kiến trúc phân tầng trong thiết kế hệ thống HCMUT-SSPS không chỉ đảm bảo tính ổn định và hiệu quả của hệ thống mà còn tạo điều kiện thuận lợi cho việc mở rộng, bảo trì và tích hợp các tính năng mới trong tương lai, phục vụ tốt hơn nhu cầu của sinh viên và giảng viên và cả về mặt quản lý.

3.1.2 Xác định tổng quan kiến trúc hệ thống

Hệ thống của chúng tôi hướng đến hai đối tượng chính: khách hàng (bao gồm học sinh và giảng viên có nhu cầu in ấn) và đội ngũ quản lý - SPSO (bao gồm SPSO và các nhân viên trực thuộc).

- **Giao diện:** Để phục vụ hai đối tượng trên, hệ thống sẽ được chia thành hai phần chính về giao diện: **Giao diện Khách hàng (Customer View)** và **Giao diện Quản lý (Manager View)**. Trong phần Manager View, do có sự khác biệt về chức năng giữa nhân viên và SPSO, giao diện này sẽ tiếp tục được chia thành hai phần nhỏ: **Giao diện SPSO (SPSO View)** và **Giao diện Nhân viên (PS View)**. Vì vậy, tổng cộng hệ thống sẽ có ba giao diện chính với các cách tiếp cận người dùng khác nhau. Customer View phục vụ cho khách hàng với các dịch vụ in ấn, trong khi SPSO View và PS View cung cấp các công cụ quản lý và hỗ trợ cho việc quản lý dịch vụ in ấn.
- **Bộ điều khiển (Controller):** Sau khi xác định được các giao diện chính, hệ thống được phân thành ba tầng tương ứng với các đối tượng sử dụng khác nhau: **Manager System**, **Customer System**, và **Staff System**.
 - **Manager System:** Ở giao diện của SPSO sẽ cung cấp các chức năng như cấu hình máy in (Config printer), quản lý báo cáo sự cố (Manage report), các chức năng về quản lý máy in (Manage printer) và người dùng (Manage User) - bao gồm cả Staff và Customer .
 - **Customer System:** Đối với giao diện người dùng sẽ bao gồm các chức năng về in ấn, đặt lịch (Printing), mua bán trang in và thanh toán (Paper buying), xem lịch sử in (Log history print) và báo cáo sự cố (Report).
 - **Staff System:** Giao diện của nhân viên sẽ chuyên về các chức năng thông báo và xử lý các đơn in của khách hàng.

Ngoài ra các giao diện người dùng đều có một phần nhằm xác thực tài khoản và danh tính của người dùng, được thực hiện thông qua HCMUT_SSO(Single Sign-On). Ngoài ra hệ thống thanh toán sẽ được liên kết với BKPAY.



Hình 26: HCMUT printer Layred Architecture



3.1.3 Giao diện người dùng

Dối với giao diện của hệ thống, nhóm quyết định chia giao diện hệ thống thành ba phần chính, mỗi phần dành cho loại người dùng có quyền hạn khác nhau: khách hàng (bao gồm sinh viên và giảng viên), nhân viên (PS), và ban quản lý SPSO. Từng giao diện sẽ mang lại trải nghiệm và chức năng riêng, phù hợp với nhu cầu của mỗi loại người dùng.

Dối với khách hàng, những người thường xuyên cần sự thuận tiện trong việc in ấn, đặc biệt là sinh viên, họ sẽ có nhu cầu in ở bất cứ đâu và bất cứ lúc nào, chúng tôi sẽ phát triển ứng dụng di động nhằm thuận tiện hơn cho việc thao tác. Ngược lại, đối với nhân viên in ấn và ban quản lý SPSO, những người có nhiệm vụ quản lý và nhận thông báo từ hệ thống, giao diện web sẽ là lựa chọn phù hợp hơn. Việc này sẽ giúp họ dễ dàng xử lý các tác vụ quản lý một cách hiệu quả hơn so với khi sử dụng thiết bị di động.

Hai nền tảng web và mobile app đã được chúng tôi trình bày chi tiết với các giao diện khác nhau dành cho các тип người dùng ở phần 6 của bài báo cáo. Chúng tôi sử dụng màu chủ đạo của trường là màu xanh lam đồng thời việc thiết kế cũng đánh vào những chức năng chính để việc in ấn và quản lý sẽ đi vào trọng tâm của vấn đề, đồng thời chúng tôi cũng xem xét tới kích cỡ của các loại giao diện khác nhau giữa app và web để có được trải nghiệm tốt nhất giữa 2 nền tảng.

3.1.4 Thiết kế lưu trữ

Để lưu trữ các dữ liệu của ứng dụng, chúng tôi sẽ sử dụng hệ cơ sở dữ liệu quan hệ, cụ thể là Postgresql. Lý do là vì ứng dụng cần lưu các dữ liệu như thông tin đơn in, thông tin khách hàng, thông tin các thiết bị và thông tin các giao dịch. Các dữ liệu đó là các dữ liệu có quan hệ với nhau, do đó cần sử dụng hệ cơ sở dữ liệu quan hệ để lưu, giúp quản lý và truy xuất dữ liệu hiệu quả hơn.

Hệ cơ sở dữ liệu sẽ chạy trên cùng máy local, và ứng dụng sẽ giao tiếp với server của database thông qua giao thức TCP.

Để việc quản lý các thực thể trong ứng dụng hiệu quả hơn, tránh các sai sót và giảm công sức khi viết các câu lệnh SQL, chúng tôi sẽ sử dụng phương pháp ORM (Object Relation Mapping). ORM sẽ giúp ánh xạ các thực thể trong cơ sở dữ liệu thành các object trong ứng dụng, từ đó chúng tôi có thể giao tiếp với database thông qua các object đó, giúp việc phát triển dễ dàng hơn.

3.1.5 Thiết lập API

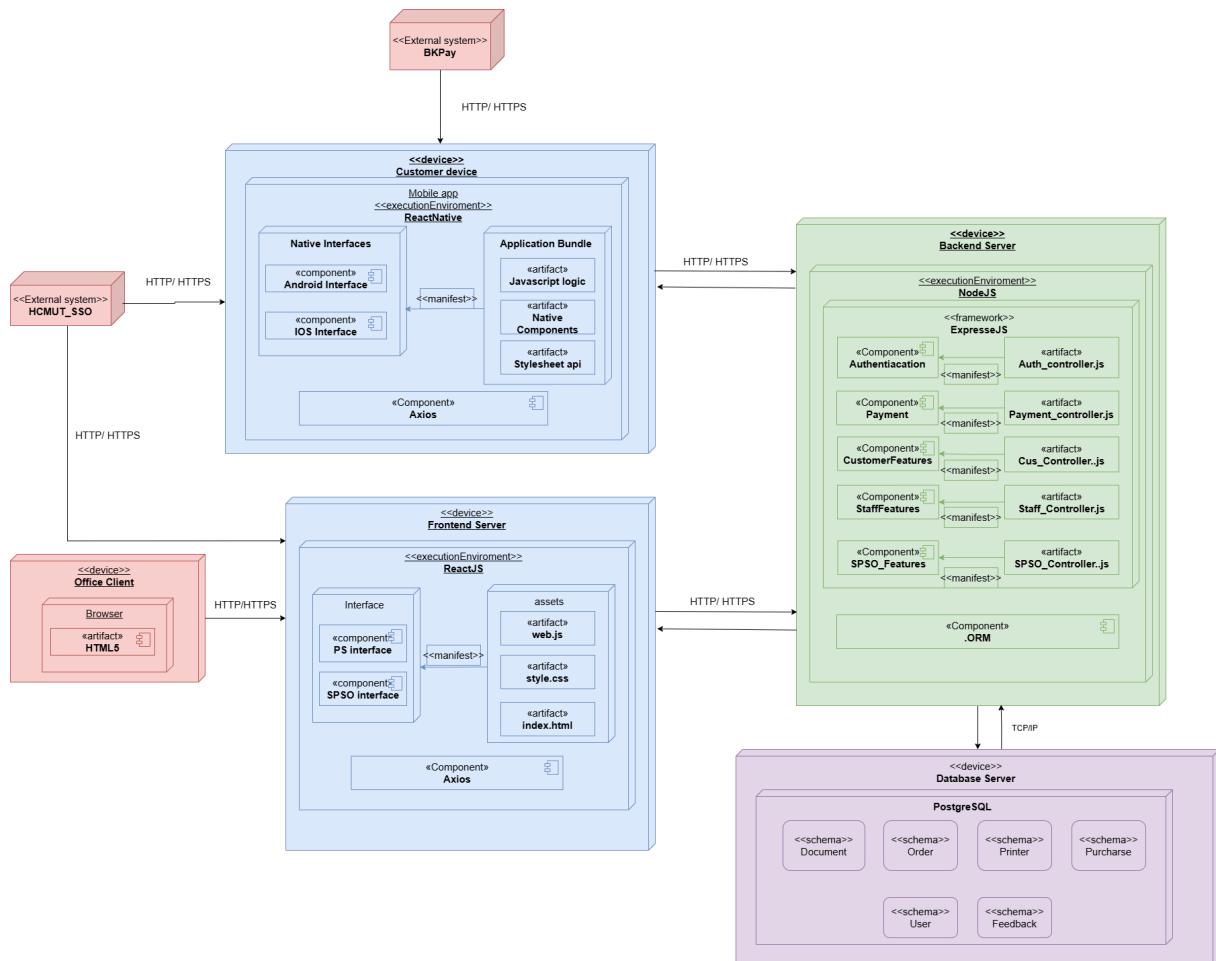
API, hay Giao diện lập trình ứng dụng, là một phương tiện giúp các phần mềm giao tiếp với nhau thông qua các định nghĩa và giao thức, hỗ trợ hệ thống web-based, hệ điều hành, hệ thống cơ sở dữ liệu, phần cứng máy tính hoặc thư viện phần mềm. API bao gồm hai thành phần chính: Request và Response. Khi client gửi Request đến server, server xử lý và trả về một Response tương ứng cho client. Trong hệ thống HCMUT Smart Printer, API hỗ trợ giao tiếp giữa hai giao diện người dùng chính, sử dụng giao thức HTTP/HTTPS để gọi API từ các controller trong tầng business logic:

- Giao diện Mobile App:** Dành cho khách hàng, cung cấp các chức năng đặt in, theo dõi trạng thái in, thanh toán, và truy cập lịch sử in ấn.
- Giao diện Web:** Dành cho nhân viên và ban quản lý SPSO, hỗ trợ quản lý yêu cầu in, điều phối công việc in, kiểm soát thanh toán, và kiểm duyệt tài liệu, quản lý máy in.

Các controller trong tầng business logic thực hiện các request HTTP/HTTPS trực tiếp từ tầng Presentation. Cách tiếp cận này giúp các thành phần trong hệ thống có thể tương tác trực tiếp đơn giản hóa quy trình gọi và xử lý API. Trong hệ thống HCMUT Smart Printer, các API từ HCMUT_SSO và BKPay là các dịch vụ bên ngoài hệ thống chính, và các controller trong tầng business logic sẽ thực hiện các request HTTP/HTTPS để gọi trực tiếp vào các API này khi cần:

- **HCMUT_SSO:** Được sử dụng cho xác thực danh tính, đảm bảo người dùng đã đăng nhập hợp lệ trước khi truy cập các chức năng in ấn và thanh toán. Từ tầng business logic, các controller gửi request đến API của HCMUT_SSO để kiểm tra thông tin xác thực của người dùng.
- **BKPay:** Được sử dụng cho các chức năng thanh toán, cho phép khách hàng thanh toán phí in ấn qua dịch vụ BKPay. Khi khách hàng tiến hành thanh toán, các controller trong tầng business logic gọi trực tiếp vào API của BKPay để thực hiện giao dịch và xác nhận trạng thái thanh toán.

3.2 Deployment Diagram



Hình 27: Deployment diagram for system

Trong quá trình triển khai, nhóm chúng tôi đã quyết định chia hệ thống thành ba phần chính: Frontend, Backend, và Database. Chúng tôi thiết lập Frontend web và Backend trên hai máy chủ độc lập, hoạt động song song để dễ dàng nâng cấp và cập nhật trong tương lai, đồng thời giảm tải cho mỗi máy chủ. Việc tách riêng Backend trên một máy chủ cũng giúp việc gọi API giữa frontend web và frontend ứng dụng di động trở nên thuận tiện hơn.

Hệ thống Frontend được chia thành hai nền tảng chính: Web và ứng dụng di động. Khách hàng sẽ sử dụng ứng dụng di động để truy cập các tính năng và sẽ giao tiếp với Backend thông qua HTTP/HTTPS. Trong khi đó, nhân viên và ban quản lý sẽ sử dụng trình duyệt web để truy cập vào web server, thực hiện các chức năng của mình trên nền tảng web và cũng giao tiếp với Backend qua HTTP/HTTPS.



Backend server đảm nhận các tác vụ chính về logic và xử lý nghiệp vụ cho toàn bộ hệ thống, đồng thời là cầu nối chính để tương tác với cơ sở dữ liệu thông qua TCP/IP. Ngoài ra Backend server còn phải thao tác và xử lý các tác vụ chính bên ngoài khác như là việc xác thực đăng nhập bởi HCMUT_SSO và phần thanh toán với BKPAY.

Sau khi tổng quát cả hệ thống, nhóm sẽ đi sâu thiết kế chi tiết từng khía cạnh. Đối với frontend, như đã nói ở trên chúng tôi tách hệ thống thành hai nền tảng là web và mobile app.

- **Mobile app:** Chúng tôi quyết định sử dụng framework React Native, một công cụ mã nguồn mở được phát triển bởi Facebook. React Native mang lại sự tiện dụng và có cộng đồng người dùng lớn, kết hợp giữa React và công nghệ Native giúp việc phát triển ứng dụng bằng JavaScript trở nên dễ dàng, đồng thời Native có thể chuyển đổi qua các nền tảng khác nhau như Android và iOS. Để kết nối và gửi các yêu cầu đến backend, chúng tôi chọn thư viện Axios hỗ trợ HTTP/HTTPS.
- **Web:** Phần web sẽ được triển khai trên server sử dụng framework ReactJS. Lý do chúng tôi chọn ReactJS là vì đây là framework được hỗ trợ rộng rãi, đồng thời nó cũng tương thích với React Native trên mobile app, giúp việc phát triển trở nên đồng bộ hơn. ReactJS sẽ bao gồm các file chính như JavaScript, HTML và CSS, cùng với thư viện Axios để gửi các yêu cầu đến backend giống như trên mobile app.

Phần backend của hệ thống sẽ được xây dựng sử dụng Node.js kết hợp với Express.js, một framework phổ biến giúp xử lý các API và quản lý các yêu cầu HTTP. Backend được chia thành các component chính, mỗi component sẽ đảm nhiệm các chức năng riêng biệt của hệ thống. Đối với ExpressJS:

1. **Authentication:** là phần quan trọng nhất trong backend, đặc biệt là khi cần đảm bảo an toàn cho hệ thống. Chúng tôi sẽ tích hợp HCMUT_SSO (Single Sign-On) để xử lý việc xác thực người dùng, bao gồm cả khách hàng, nhân viên và SPSO. Từ đó mới trả về response và giao diện phù hợp.
2. **Payment:** Component này xử lý tất cả các chức năng liên quan đến thanh toán của khách hàng. Nó được backend kết nối với hệ thống BKpay thông qua HTTP/HTTPS.
3. **Customer Features:** Component này sẽ cung cấp các chức năng phục vụ khách hàng sử dụng hệ thống in ấn từ xa. Sau khi được phân quyền bởi phần xác thực, component sẽ được xử lý các yêu cầu từ frontend liên quan đến các quyền mà khách hàng được thực hiện như đăng ký in hoặc thanh toán mua trang,... kèm với một minh chứng xác thực nào đó.
4. **Staff Features:** Component này sẽ bao gồm các chức năng để xử lý các tác vụ của Nhân viên ví dụ như xử lý đơn, nhận thông báo đơn in,... Cũng như Customer Features, sau khi đã xử lý xác thực thì component này sẽ xử lý trực tiếp các yêu cầu được frontend gửi về với quyền của nhân viên.
5. **SPSO Features:** Component này sẽ cung cấp các chức năng phục vụ ban quản lý SPSO. Sau khi được phân quyền bởi phần xác thực, component sẽ được xử lý các yêu cầu từ frontend liên quan đến các quyền mà SPSO được thực hiện như quản lý máy in, cấu hình máy in, quản lý các đơn hàng,... kèm với một minh chứng xác thực nào đó.
6. **ORM (Object-Relational Mapping):** Để dễ dàng kết nối với cơ sở dữ liệu, chúng tôi sử dụng một ORM (Object-Relational Mapping) như Sequelize (hoặc TypeORM) để quản lý các bảng dữ liệu trong hệ thống. Có thể hiểu việc tích hợp ORM để mô hình lại dữ liệu thành các đối tượng như là tầng Persistence layer trong kiến trúc đa tầng. ORM này sẽ được tích hợp bằng một thư viện của Nodejs nhằm:



- Quản lý cơ sở dữ liệu: ORM giúp backend dễ dàng kết nối và thao tác với cơ sở dữ liệu mà không cần phải viết SQL thủ công. ORM chuyển đổi các đối tượng JavaScript thành các bản ghi trong cơ sở dữ liệu, giúp truy vấn và cập nhật dữ liệu trở nên đơn giản và an toàn.
- Mô hình dữ liệu: Mỗi component (StaffFeature, CustomerFeature, SPSOFeature, v.v.) sẽ có các bảng dữ liệu riêng biệt trong cơ sở dữ liệu. ORM sẽ giúp định nghĩa các mô hình (models) này và đảm bảo rằng các thao tác với cơ sở dữ liệu được thực hiện một cách hiệu quả và nhất quán.

Cuối cùng, hệ thống sẽ sử dụng PostgreSQL làm cơ sở dữ liệu chính, Backend Server sẽ kết nối thông qua TCP/IP. Hệ thống này được tổ chức thành các schema khác nhau, mỗi schema sẽ chịu trách nhiệm lưu trữ và quản lý các loại dữ liệu liên quan đến từng phần của hệ thống. Cơ sở dữ liệu được phân chia thành các schema chuyên biệt, giúp tổ chức và quản lý dữ liệu hiệu quả:

- Schema Document:** Lưu trữ thông tin về các tài liệu được khách hàng tải lên để in, bao gồm thông tin như loại tài liệu, định dạng, dung lượng, và các thiết lập in ấn.
- Schema Order:** Quản lý các đơn đặt in từ khách hàng. Bao gồm thông tin chi tiết về đơn hàng, trạng thái đơn, thời gian đặt hàng, và các dịch vụ in được yêu cầu.
- Schema Printer:** Lưu trữ thông tin về các máy in, bao gồm ID máy, vị trí, tình trạng hoạt động, và các thông số kỹ thuật. Thông tin này giúp SPSO và nhân viên quản lý và theo dõi trạng thái của từng máy.
- Schema User:** Quản lý thông tin người dùng trong hệ thống, bao gồm như quyền hạn (khách hàng, nhân viên, ban quản lý), họ tên,...
- Schema Feedback:** Lưu trữ báo cáo về hoạt động in ấn, bao gồm số lượng đơn hàng, hiệu suất máy in, và các sự cố nếu có. Các báo cáo này giúp SPSO phân tích và tối ưu hóa hệ thống.
- Schema Purchase:** Lưu trữ các giao dịch thanh toán từ khách hàng, bao gồm phương thức thanh toán, thời gian, và trạng thái của giao dịch.

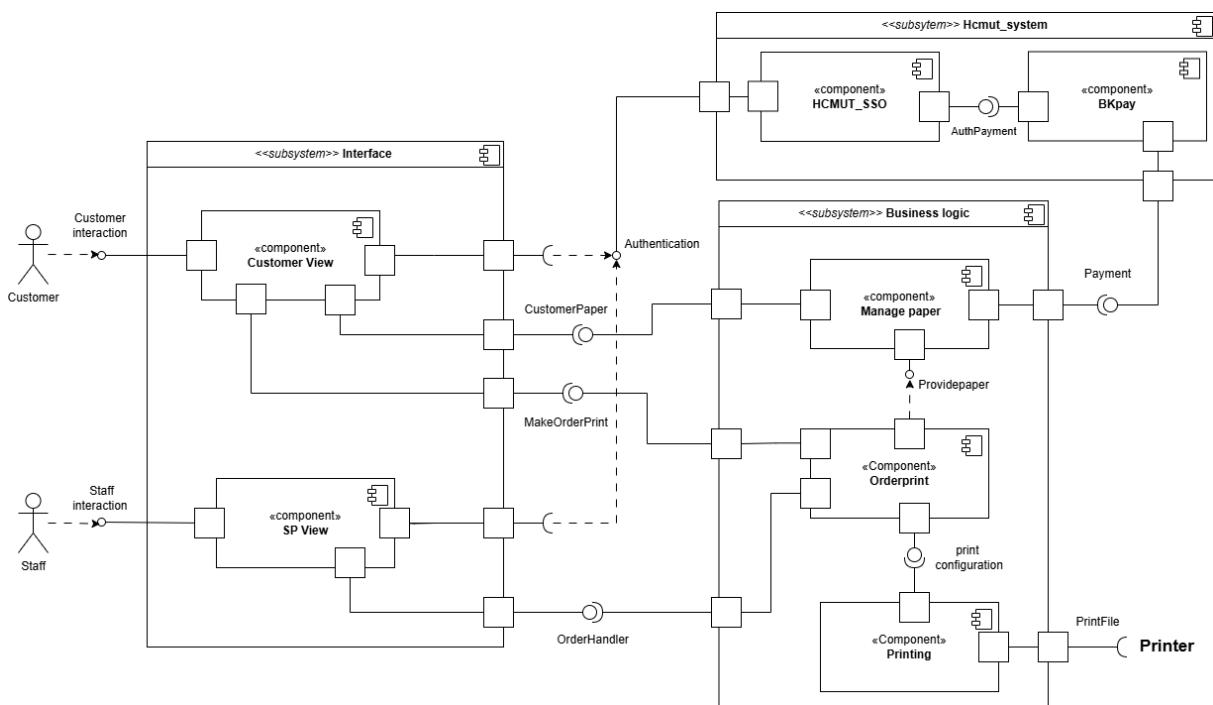
3.3 Component diagram

Trong dự án này, chúng tôi sẽ áp dụng phương pháp phát triển dựa trên thành phần (Component-Based Development). Mỗi thành phần sẽ được phát triển độc lập và giao tiếp với các thành phần khác thông qua các giao diện (interface). Phương pháp này giúp tăng cường tính linh hoạt trong quá trình phát triển phần mềm, đồng thời tạo điều kiện thuận lợi cho đội ngũ phát triển trong việc bảo trì và nâng cấp hệ thống sau này.

3.3.1 Component diagram - Thực hiện in ấn và đặt lịch

Đối với kiến trúc đa tầng, chúng tôi sẽ chia hệ thống thành các hệ thống nhỏ được chia theo từng tầng và cách liên hệ và tương tác giữa các thành phần trong module in ấn và đặt lịch. Trong diagram này, chúng tôi sẽ có ba thành phần chính là Interface, Business logic và HCMUT_system.

- Interface:** Hệ thống này sẽ quản lý các giao diện được cung cấp cho người dùng, ở đây sẽ là Customer View và PS view đây cũng tương ứng là frontend của hệ thống. Trong đó sẽ chia ra như sau:



Hình 28: Component diagram cho đặt lịch và in ấn

- **Customer View:** thành phần này chứa các lớp xử lý việc hiển thị giao diện dành cho khách hàng và giúp khách hàng tương tác với Backend của hệ thống (mua trang in, đặt lịch in). Tuy nhiên, việc được quyền tương tác với Backend còn phải thông qua hệ thống xác thực HCMUT_SSO
 - Required Interface:
 - * Authentication chức năng nhằm xác thực khách hàng.
 - * CustomerPaper yêu cầu các chức năng của Customer đến để xử lý các vấn đề liên quan đến trang in.
 - * MakeOrderPrint yêu cầu chức năng của Customer đến để tạo đơn in và đặt in.
 - Provided Interface: Customer interaction cung cấp giao diện để khách hàng có thể tương tác.
- **Staff View:** thành phần này chứa các lớp xử lý việc hiển thị giao diện dành cho nhân viên và giúp nhân viên tương tác với Backend của hệ thống (phản hồi các đơn in). Tuy nhiên, việc được quyền tương tác với Backend còn phải thông qua hệ thống xác thực HCMUT_SSO
 - Required Interface: Authentication chức năng nhằm xác thực nhân viên.
 - Provided Interface:
 - * Staff interaction cung cấp giao diện để nhân viên có thể tương tác.
 - * OrderHandler cung cấp việc phản hồi và xử lý đơn in.
- 2. **HCMUT_system:** Hệ thống nhỏ này cung cấp hai thành phần chính được tích hợp vào hệ thống chính là phần đăng nhập hay xác thực và phần thanh toán, nhằm cung cấp các chức năng bên ngoài cho hệ thống chính.
 - **HCMUT_SSO:** Đây là thành phần quan trọng của hệ thống, đóng vai trò trong việc phân quyền người dùng. Thành phần này cung cấp các token xác thực, cho phép người dùng truy cập và sử dụng các chức năng mà họ được phép.



– Provided Interface:

- * Authentication cung cấp các chức năng để xác thực người dùng như khách hàng, nhân viên để phân quyền hạn.
- * AuthPayment cung cấp xác thực tương tự nhưng để thanh toán cho BKPay.

• **BKPay:** bao gồm các tính năng và thành phần xử lý các mục liên quan đến thanh toán, và được liên kết tới thẳng hệ thống.

- Required Interface: AuthPayment yêu cầu cần xác thực để thanh toán đúng người.
- Provided Interface: Payment nó cung cấp cho hệ thống tính năng thanh toán, và cung cấp ở đây cho phần mua trang giấy để in của hệ thống in.

3. **Business Logic:** Đây là hệ thống nhỏ cung cấp các phần xử lý các yêu cầu của frontend, tương ứng với backend của cả hệ thống. Phần này sẽ xử lý các tác vụ chính liên quan đến việc đặt lịch in, in ấn và mua trang in.

• **Manage paper:** Thành phần này chịu trách nhiệm quản lý tài liệu trong hệ thống, bao gồm các chức năng như thêm, xóa, cập nhật và cung cấp giấy để in. Ngoài ra, thành phần này cũng hỗ trợ kiểm tra trạng thái và cung cấp thông tin chi tiết của số trang còn lại và mua thêm trang.

- Required Interface: Payment nó yêu cầu một lệnh trực tiếp đến với BKPay để thanh toán các trang khách hàng sẽ mua.

– Provided Interface:

- * CustomerPaper cung cấp các chức năng cho giao diện người dùng để xử lý các hàm thêm, xóa, thanh toán,... liên quan đến các trang in.
- * Providepaper cung cấp cho các đơn in các trang in để in.

• **Orderprint:** Thành phần này chịu trách nhiệm xử lý các việc liên quan đến đơn in, đặt lịch in ấn.

- Required Interface: OrderHandler giao diện này sẽ gửi yêu cầu đến giao diện cho nhân viên nhằm xử lý thông tin của các đơn in (đơn in đặt lịch lấy sau).

– Provided Interface:

- * MakeOrderPrint cung cấp hàm để cho giao diện khách hàng tạo và đặt đơn in.
- * print configuration nó sẽ gửi các thông tin cần thiết kèm như thành phần của việc in đến với thành phần Printing để xử lý việc in ấn.

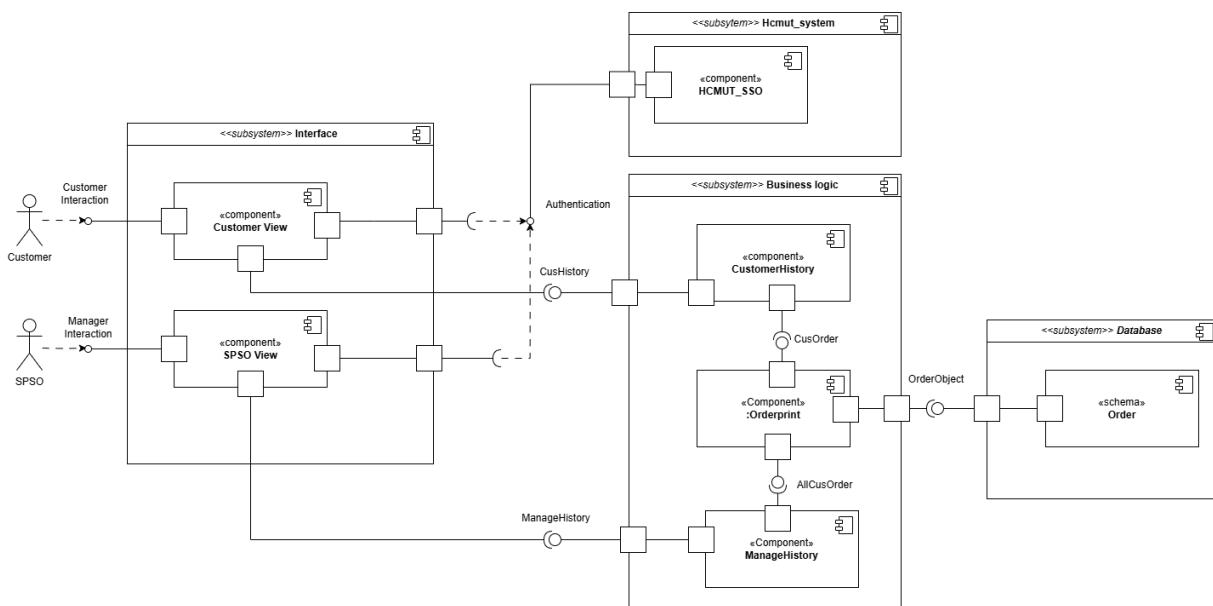
• **Printing:** Thành phần này chịu trách nhiệm xử lý các việc liên quan đến việc in ấn và kết nối trực tiếp tới máy in.

– Required Interface:

- * print configuration thành phần này cần các thông số và yêu cầu từ orderprint để thực hiện in ấn.
- * PrintFile giao diện này yêu cầu các Printer ngoài hệ thống thực hiện các tác vụ in ấn.

3.3.2 Component diagram - Xem lịch sử in ấn

Chúng tôi tiếp tục chia hệ thống thành các hệ thống nhỏ được chia theo từng tầng và cách liên hệ và tương tác giữa các thành phần trong module xem lịch sử in ấn. Trong diagram này, chúng tôi sẽ có bốn hệ thống nhỏ chính là Interface, Business logic, HCMUT_system và Database.



Hình 29: Component diagram cho lịch sử in ấn

1. **Interface:** Hệ thống này sẽ quản lý các giao diện được cung cấp cho người dùng, ở đây sẽ là Customer View và SPSO view đây cũng tương ứng là frontend của hệ thống. Trong đó sẽ chia ra như sau:

- **Customer View:** thành phần này chứa các lớp xử lí việc hiển thị giao diện dành cho khách hàng và giúp khách hàng tương tác với Backend của hệ thống (mua trang in, đặt lịch in). Tuy nhiên, việc được quyền tương tác với Backend còn phải thông qua hệ thống xác thực HCMUT_SSO
 - Required Interface:
 - * Authentication chức năng nhằm xác thực khách hàng.
 - * CusHistory chức năng nhằm thực hiện yêu cầu của khách hàng để xem lịch sử các đơn in.
 - Provided Interface: Customer interaction cung cấp giao diện để khách hàng có thể tương tác.
- **SPSO View:** thành phần này chứa các lớp xử lí việc hiển thị giao diện dành cho ban quản lý và giúp nhân viên tương tác với Backend của hệ thống (phản hồi các đơn in). Tuy nhiên, việc được quyền tương tác với Backend còn phải thông qua hệ thống xác thực HCMUT_SSO
 - Required Interface:
 - * Authentication chức năng nhằm xác thực ban quản lý.
 - * ManageHistory yêu cầu các chức năng nhằm quản lý các lịch sử tạo đơn in.
 - Provided Interface: SPSO interaction cung cấp giao diện để ban quản lý có thể tương tác.

2. **HCMUT_system:** Hệ thống nhỏ này cung cấp phần đăng nhập hay xác thực.

- **HCMUT_SSO:** Đây là thành phần quan trọng của hệ thống, đóng vai trò trong việc phân quyền người dùng. Thành phần này cung cấp các token xác thực, cho phép người dùng truy cập và sử dụng các chức năng mà họ được phép.
 - Provided Interface: Authentication cung cấp các chức năng để xác thực người dùng như khách hàng, nhân viên để phân quyền hạn.



3. **Business Logic:** Đây là hệ thống nhỏ cung cấp các phần xử lý các yêu cầu của frontend, tương ứng với backend của hệ thống. Phần này sẽ xử lý các tác vụ chính liên quan đến việc xem lịch sử in ấn.

- **CustomerHistory:** Thành phần này sẽ chịu trách nhiệm lấy dữ liệu lịch sử in ấn để thể hiện cho khách hàng.
 - Required Interface: CusOrder thành phần này sẽ yêu cầu để lấy các đối tượng OrderPrint nhằm lọc ra lịch sử của khách hàng.
 - Provided Interface: CusHistory giao diện này sẽ cung cấp các lịch sử in ấn sau khi đã qua xử lý theo yêu cầu của khách hàng.
- **:Orderprint:** Thành phần này được lấy từ Database vô mô hình lại thành Object để xử lý về sau.
 - Required Interface: OrderObject lấy thông tin Order từ Database để mô hình lại
 - Provided Interface:
 - * CusOrder cung cấp danh sách lịch sử in ấn cho khách hàng phù hợp
 - * AllCusOrder cung cấp danh sách lịch sử in ấn của tất cả khách hàng nhằm quản lý.
- **ManageHistory:** Tập hợp này sẽ cung cấp các hàm cho ban quản lý nhằm quản lý danh sách lịch sử in ấn của khách hàng.
 - Required Interface: AllCusOrder thành phần này sẽ gửi yêu cầu để lấy tất cả dữ liệu đơn đặt in của tất cả khách hàng về để xử lý.
 - Provided Interface: ManageHistory cung cấp các hàm các thuộc tính nhằm giúp giao diện SPSO có thể quản lý các đơn đặt in.

4. **Database:** Đây là phần sâu nhất của hệ thống nhằm lưu trữ những dữ liệu quan trọng.

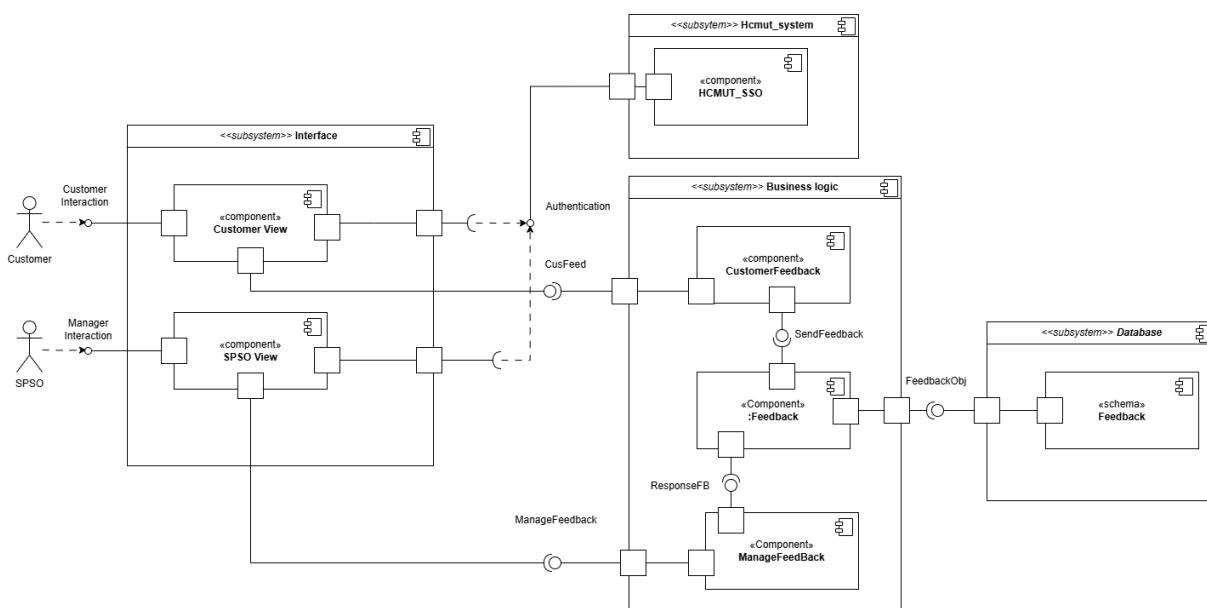
- **Order:** đây là một phần của Database lưu trữ tất cả dữ liệu của đơn in ấn của khách hàng.
 - Provided Interface: OrderObject Nó cung cấp cho backend dữ liệu về cần thiết về đơn đặt in của khách hàng

3.3.3 Component diagram - Phản hồi hệ thống

Chúng tôi chia hệ thống giống thành bốn giống như module xem lịch sử in ấn, bao gồm: Interface, Business Logic, HCMUT_system và Database.

1. **Interface:** Hệ thống này sẽ quản lý các giao diện được cung cấp cho người dùng, ở đây sẽ là Customer View và SPSO view đây cũng tương ứng là frontend của hệ thống. Trong đó sẽ chia ra như sau:

- **Customer View:** thành phần này chứa các lớp xử lý việc hiển thị giao diện dành cho khách hàng và giúp khách hàng tương tác với Backend của hệ thống (báo cáo feedback) . Tuy nhiên, việc được quyền tương tác với Backend còn phải thông qua hệ thống xác thực HCMUT_SSO
 - Required Interface: Authentication chức năng nhằm xác thực khách hàng.
 - Provided Interface:
 - * Customer interaction cung cấp giao diện để khách hàng có thể tương tác.
 - * CusFeed chức năng nhằm cung cấp feedback của khách hàng lên cho hệ thống.



Hình 30: Component diagram cho báo cáo Feedback

- **SPSO View:** thành phần này chứa các lớp xử lý việc hiển thị giao diện dành cho ban quản lý và giúp nhân viên tương tác với Backend của hệ thống (xem feedback và phản hồi). Tuy nhiên, việc được quyền tương tác với Backend còn phải thông qua hệ thống xác thực HCMUT_SSO

- Required Interface:
 - * Authentication chức năng nhằm xác thực ban quản lý.
 - * ManageFeedback yêu cầu các chức năng nhằm quản lý các feedback.
- Provided Interface: SPSO interaction cung cấp giao diện để ban quản lý có thể tương tác.

2. HCMUT_system: Hệ thống nhỏ này cung cấp phần đăng nhập hay xác thực.

- **HCMUT_SSO:** Đây là thành phần quan trọng của hệ thống, đóng vai trò trong việc phân quyền người dùng. Thành phần này cung cấp các token xác thực, cho phép người dùng truy cập và sử dụng các chức năng mà họ được phép.

- Provided Interface: Authentication cung cấp các chức năng để xác thực người dùng như khách hàng, nhân viên để phân quyền hạn.

3. Business Logic: Đây là hệ thống nhỏ cung cấp các phần xử lý các yêu cầu của frontend, tương ứng với backend của hệ thống. Phần này sẽ xử lý các tác vụ chính liên quan đến Feedback.

- **CustomerFeedback:** Thành phần này sẽ chịu trách nhiệm xử lý các feedback gửi đến cuối khách hàng.

- Required Interface: CusFeed thành phần này sẽ nhận các thông tin feedback của khách hàng.
- Provided Interface: SendFeedback giao diện này sẽ cung cấp các feedback để mô hình ra object Feedback.

- **:Feedback:** Thành phần này được lấy từ Database và mô hình lại thành Object để xử lý về sau.

- Required Interface:



- * FeedbackObj lấy thông tin feedback từ Database để mô hình lại
- * SendFeedBack nhập thông tin để mô hình ra Feedback của khách hàng.
- * ResponseFB giao diện này nhận các cung cấp quyết từ các quyền quản lý từ ban quản lý.
- **ManageFeedback:** Tập hợp này sẽ cung cấp các hàm cho ban quản lý nhằm quản lý danh sách feedback của khách hàng.
 - Provided Interface:
 - * ManageFeedback cung cấp các hàm các thuộc tính nhằm giúp giao diện SPSO có thể quản lý các feedback.
 - * ResponseFB giao diện cung cấp các hàm để xử lý và cập nhật các feedback.
- 4. **Database:** Đây là phần sâu nhất của hệ thống nhằm lưu trữ những dữ liệu quan trọng.
 - **Feedback:** đây là một phần của Database lưu trữ tất cả dữ liệu của feedback của khách hàng.
 - Provided Interface: FeedbackObj nó cung cấp cho backend dữ liệu về cần thiết về feedback của khách hàng



4 Implementation sprint 1

4.1 Tạo Git repository để làm việc và đưa các documents của hệ thống lên

The screenshot shows a GitHub repository page for 'blueonline07'. The repository has 9 branches and 0 tags. The README file contains the following content:

```
HCMUT Student Smart Printing Service

Table of Contents
• Introduction
• Features
• Installation
• Usage
• Contributing
```

The repository details section includes:

- About: This is the work for assignment of Software Engineering (CO3001) course.
- Readme
- Activity
- 0 stars
- 1 watching
- 0 forks
- Releases: No releases published. Create a new release.
- Packages: No packages published. Publish your first package.
- Contributors: 6 contributors shown with their profile icons.
- Languages: JavaScript 100.0%.

Hình 31: Link github của project

Dể phát triển ứng dụng HCMUT SSPS, nhận thấy số lượng thành viên trong dự án là tương đối nhiều (7) và các thành viên làm việc trong các chức năng khác nhau của dự án, nên chúng tôi quyết định sử dụng gitflow để cùng nhau làm việc trên git. Việc thống nhất một workflow trên git sẽ giúp quá trình phát triển của nhóm thuận tiện hơn, tránh các xung đột về code trong quá trình phát triển. Bên cạnh đó, gitflow cũng giúp tách biệt môi trường phát triển và môi trường production, bảo vệ môi trường production khỏi các lỗi phát sinh trong quá trình phát triển.

The screenshot shows the 'docs' directory of the 'student-smart-printing-service' repository. The directory contains several sub-folders: 'activity-diagram', 'architecture', 'class-diagram', and 'sequence-diagram'. The commit history for this directory shows the following changes:

Name	Last commit message	Last commit date
..		3 minutes ago
activity-diagram	docs: add diagrams to docs	3 minutes ago
architecture	docs: add diagrams to docs	3 minutes ago
class-diagram	docs: add diagrams to docs	3 minutes ago
sequence-diagram	docs: add diagrams to docs	3 minutes ago

Hình 32: Các sơ đồ được thêm vào thư mục docs

Đối với Gitflow, chúng tôi phân nhánh git ra thành các nhánh nhỏ như sau:



- main : đây là nhánh production của hệ thống, cần được bảo vệ khỏi các lỗi trong quá trình phát triển.
- dev : đây là nhánh chính trong quá trình phát triển phần mềm. Phần lớn thời gian các thành viên sẽ làm việc trên các nhánh được tách ra từ nhánh này.
- feat : các nhánh chức năng của dự án. Các thành viên sẽ đảm nhận các chức năng của riêng mình, sau khi hiện thực các chức năng đó trên các nhánh này, các thành viên sẽ merge các nhánh này vào lại nhánh dev để thêm tính năng đó vào phần mềm
- các nhánh khác: một số nhánh khác như hotfix dùng để chỉnh sửa các lỗi có thể sửa nhanh trong nhánh main. Trong dự án này, nhóm chủ yếu sử dụng các nhánh được nêu trên.

4.2 Kiểm tra tính khả dụng của Giao diện người dùng

4.2.1 Tổng quan dự án

1. Tên dự án: Kiểm tra tính khả dụng Giao diện người dùng đối với Hệ thống in ấn Sinh viên HCMUT_SPSS
2. Mục tiêu kiểm thử:
 - Khả năng người tham gia thực hiện thành công tác vụ được yêu cầu.
 - Đánh giá về mức thân thiện, dễ sử dụng của hệ thống: Về thẩm mỹ (font chữ, màu sắc, bố cục), về khả năng sử dụng (có khó hiểu, khó sử dụng không).
3. Thời gian thực hiện: 6 ngày (21/11/2024 - 26/11/2024).
4. Phương pháp kiểm tra: Thử nghiệm từ xa không có người dẫn dắt (remote unmoderated testing): Người tham gia tự thực hiện nhiệm vụ mà không có sự hướng dẫn trực tiếp.
5. Công cụ hỗ trợ: Google form, Figma.
6. Đối tượng tham gia: Sinh viên Trường đại học Bách Khoa TP. HCM.

4.2.2 Quá trình kiểm tra

Nhóm sử dụng Google form để thu thập ý kiến về tính khả dụng của giao diện người dùng MVP 1 (Mục 6). Đối tượng tham gia thử nghiệm được mô tả sơ lược về hệ thống trước khi thực hiện các tác vụ thử nghiệm trên figma. Dưới đây là danh sách kịch bản được nhóm đưa ra:

1. Kịch bản 1: Bạn là sinh viên và đang cần in một tài liệu cho tiết học sắp tới, hãy tạo một đơn in tự động lấy liền để kịp có tài liệu cho bài học.
2. Kịch bản 2: Bạn là sinh viên, bạn nhận ra mình đã hết trang in trong hệ thống, hãy mua thêm 20 trang in.
3. Kịch bản 3: Bạn là sinh viên, bạn cần xem lại lịch sử in từ ngày 1/11 tới ngày 5/11 bạn đã in những tài liệu nào.
4. Kịch bản 4: Bạn là nhân viên in ấn (PS), bạn vừa sắp xếp tài liệu cho đơn in S24000022 (đơn in cuối danh sách) xong, hãy cập nhật trạng thái đơn in thành Sẵn sàng để sinh viên đến nhận tài liệu.
5. Kịch bản 5: Bạn là SPSO, bạn cần xem lại lịch sử in ấn của một máy in có mã là 107H6P2.

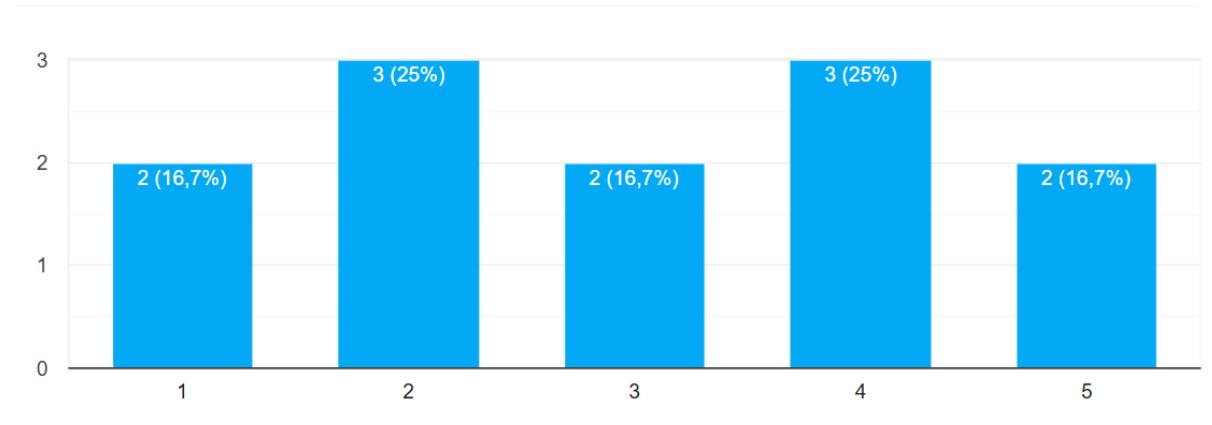
Các kịch bản trên bao quát năm chức năng chính của hệ thống: Thực hiện đơn in, mua thêm trang in, xem lịch sử in ấn, Nhân viên in cập nhật trạng thái đơn in lấy sau tại quầy, Quản trị viên theo dõi lịch sử in ấn.

4.2.3 Kết quả kiểm thử

1. **Kịch bản 1:** Bạn là sinh viên và đang cần in một tài liệu cho tiết học sắp tới, hãy tạo một đơn in tự động lấy liền để kịp có tài liệu cho bài học.

Câu hỏi 1.1: Trên thang điểm từ 1 - 5, bạn đánh giá độ khó khi thực hiện việc tạo một đơn in thế nào ?

Kết quả: Độ khó trung bình 3.0.

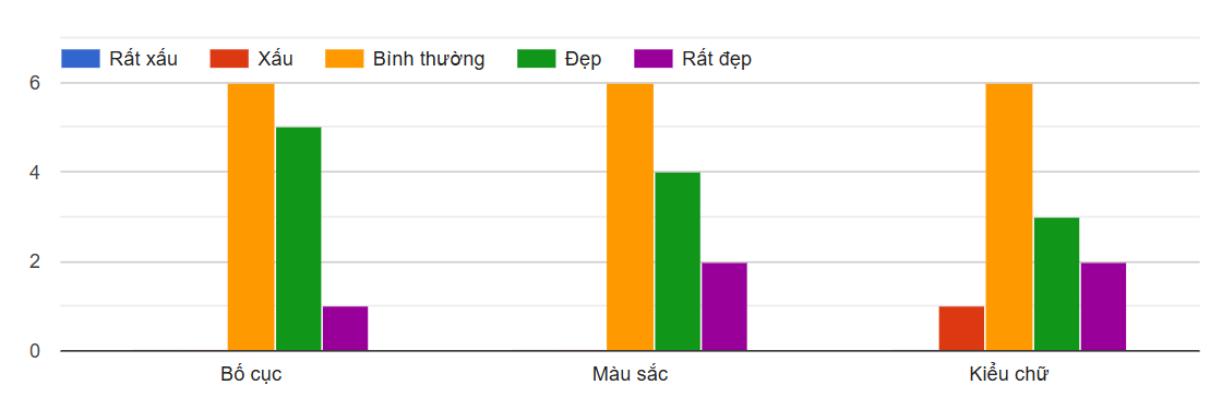


Hình 33: Biểu đồ đánh giá độ khó khi thực hiện đơn in

Câu hỏi 1.2: Trên thang điểm từ 1 - 5, bạn đánh giá giao diện của trang tạo đơn in thế nào?

Kết quả:

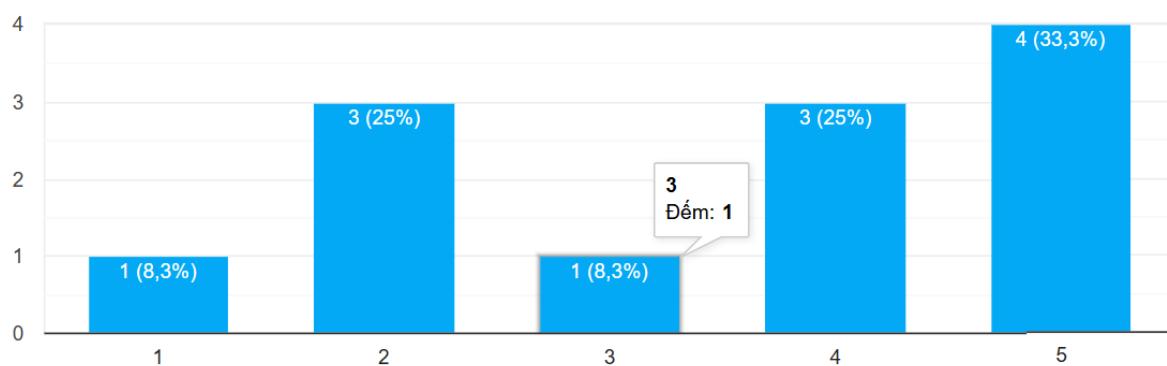
- Bố cục: 50% Bình thường, 41.6% Đẹp và 8.4% Rất đẹp.
- Màu sắc: 50% Bình thường, 33.3% Đẹp và 16.77% Rất đẹp.
- Font chữ: 50% Bình thường, 25% Đẹp và 16.6% Rất đẹp và 8.4% Xấu.



Hình 34: Biểu đồ đánh giá tính thẩm mỹ giao diện thực hiện đơn in

Góp ý kịch bản 1:

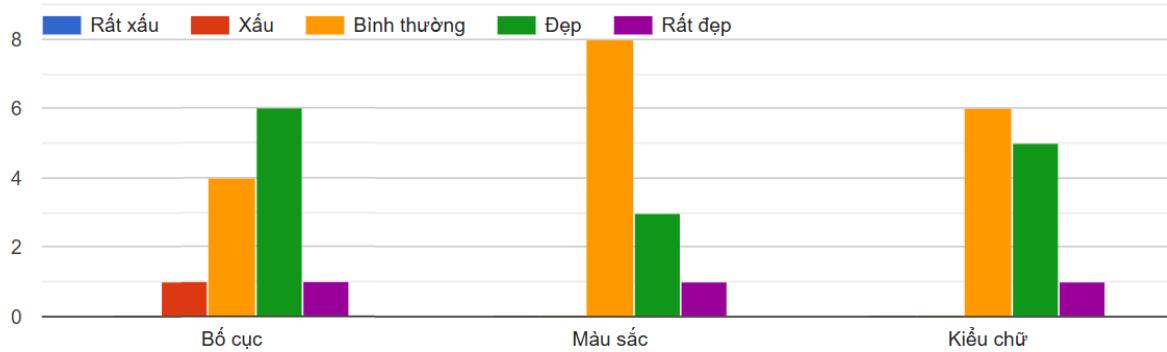
- Xác nhận giao dịch in nên show cả thông số in.
 - Các mục lựa chọn có thể show list để dễ dàng chọn hơn trên điện thoại, khi chọn một cơ sở nên ẩn các tòa ở cơ sở kia đi.
2. Kịch bản 2: Bạn là sinh viên, bạn nhận ra mình đã hết trang in trong hệ thống, hãy mua thêm 20 trang in.
- Câu hỏi 2.1:** Trên thang điểm từ 1 - 5, bạn đánh giá độ khó khi thực hiện việc mua trang in thế nào ?
- Kết quả:* Độ khó trung bình 3.5.



Hình 35: Biểu đồ đánh giá độ khó khi thực hiện mua trang in

- Câu hỏi 2.2:** Trên thang điểm từ 1 - 5, bạn đánh giá diện mua trang in thế nào
- Kết quả:*

- Bố cục: 33.3% Bình thường, 60% Đẹp và 8.35% Rất đẹp và 8.35% Xấu.
- Màu sắc: 66.7% Bình thường, 25% Đẹp và 8.3% Rất đẹp.
- Font chữ: 50% Bình thường, 41.6% Đẹp và 8.4% Rất đẹp.



Hình 36: Biểu đồ đánh giá tính thẩm mỹ giao diện mua trang in

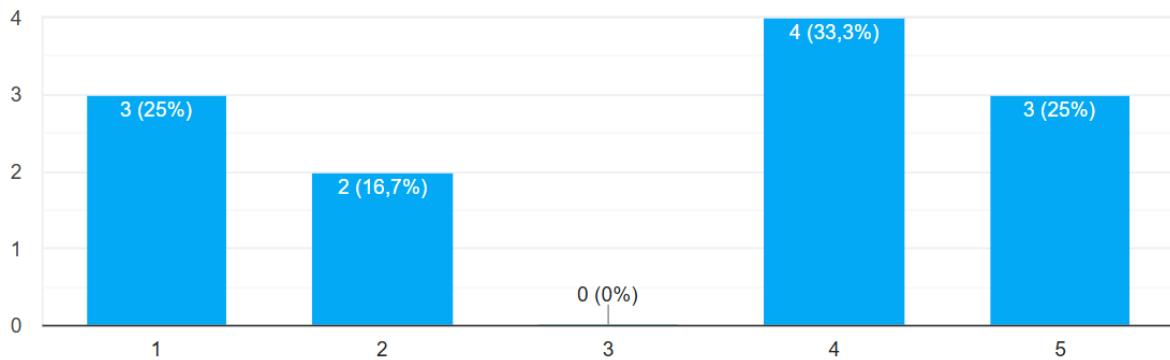
Góp ý kịch bản 2:

- Chức năng bị ẩn trong nút setting, rất khó để tìm.

3. Kịch bản 3: Bạn là sinh viên, bạn cần xem lại lịch sử in từ ngày 1/11 tới ngày 5/11 bạn đã in những tài liệu nào.

Câu hỏi 3.1: Trên thang điểm từ 1 - 5, bạn đánh giá độ khó khi xem lịch sử in thế nào ?

Kết quả: Độ khó trung bình 3.17.

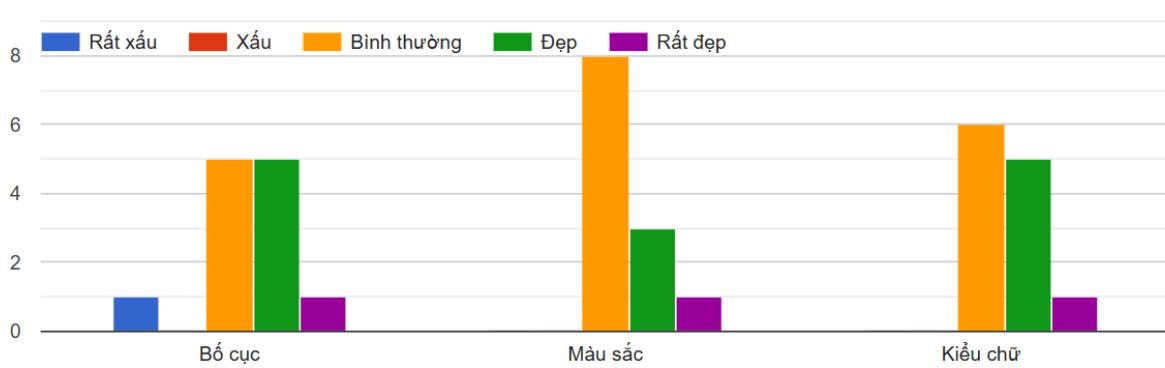


Hình 37: Biểu đồ đánh giá độ khó khi xem lịch sử đơn in

Câu hỏi 3.2: Trên thang điểm từ 1 - 5, bạn đánh giá giao diện xem lịch sử in thế nào?

Kết quả:

- Bố cục: 41.6% Bình thường, 41.6% Đẹp và 8.4% Rất đẹp và 8.4% Rất xấu.
- Màu sắc: 66.7% Bình thường, 25% Đẹp và 8.3% Rất đẹp.
- Font chữ: 50% Bình thường, 41.6% Đẹp và 8.4% Rất đẹp.



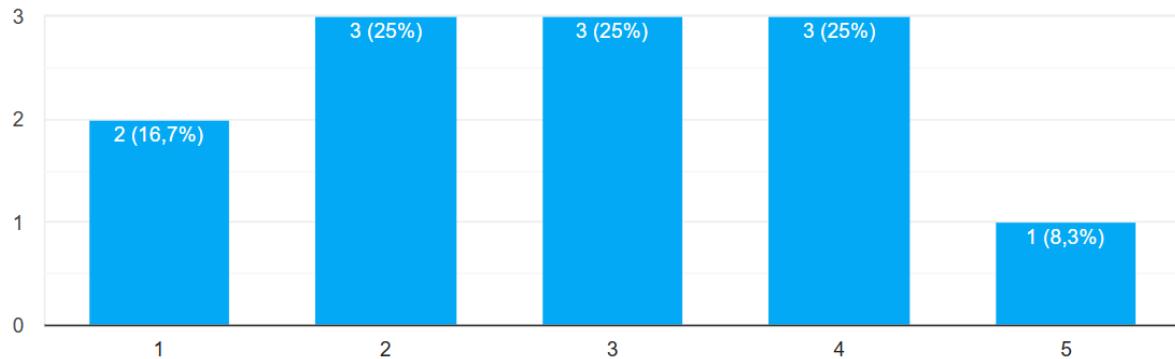
Hình 38: Biểu đồ đánh giá tính thẩm mỹ giao diện xem lịch sử in

Góp ý kịch bản 3:

- Nên xem xét bố trí ở vị trí khác, không phải nút setting.
4. Kịch bản 4: Bạn là nhân viên in ấn (PS), bạn vừa sắp xếp tài liệu cho đơn in S24000022 (đơn in cuối danh sách) xong, hãy cập nhật trạng thái đơn in thành Sẵn sàng để sinh viên đến nhận tài liệu.

Câu hỏi 4.1: Trên thang điểm từ 1 - 5, bạn đánh giá độ khó khi cập nhật đơn in thế nào ?

Kết quả: Độ khó trung bình 2.83.

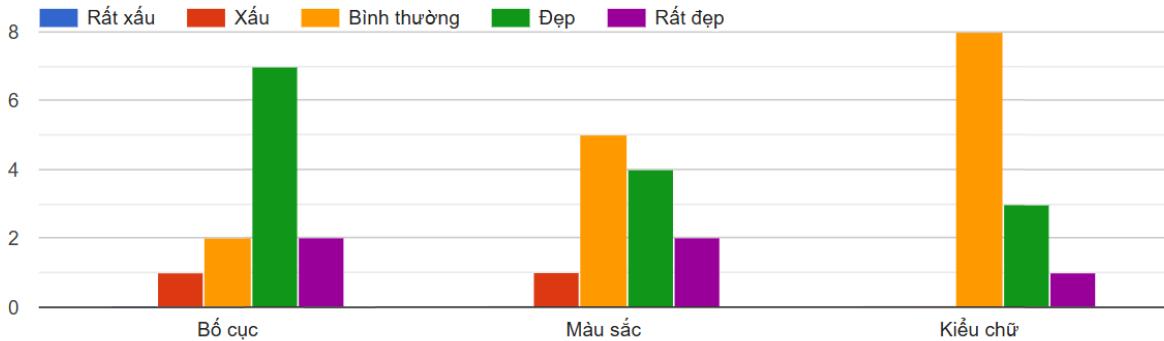


Hình 39: Biểu đồ đánh giá độ khó khi PS cập nhật trạng thái đơn in

Câu hỏi 4.2: Trên thang điểm từ 1 - 5, bạn đánh giá giao diện cập nhật đơn in thế nào?

Kết quả:

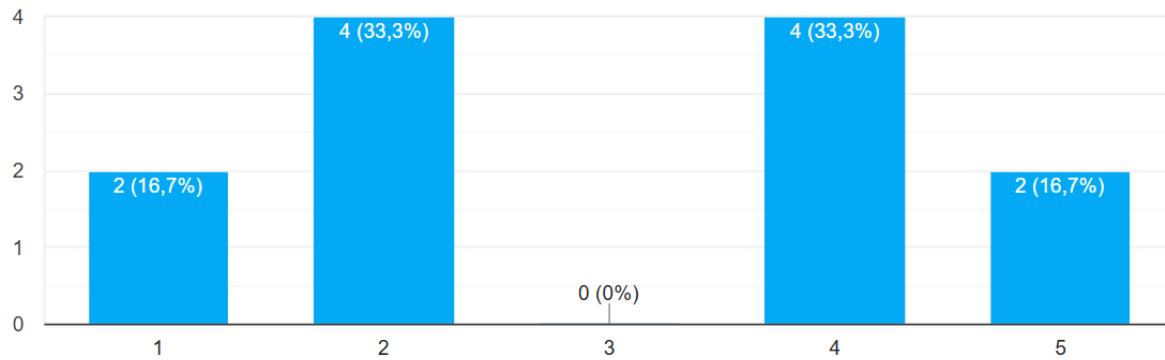
- Bố cục: 16.67% Bình thường, 58.3% Đẹp và 16.67% Rất đẹp và 8.36% Xấu.
- Màu sắc: 41.67% Bình thường, 33.3% Đẹp và 16.67% Rất đẹp và 8.36% Xấu.
- Font chữ: 66.67% Bình thường, 25% Đẹp và 8.33% Rất đẹp.



Hình 40: Biểu đồ đánh giá tính thẩm mỹ giao diện PS cập nhật trạng thái đơn in

Góp ý kịch bản 4:

- Nên cho phép click vào đơn in để xem chi tiết.
5. Kịch bản 5: Bạn là SPSO, bạn cần xem lại lịch sử in ấn của một máy in có mã là 107H6P2.
- Câu hỏi 5.1:** Trên thang điểm từ 1 - 5, bạn đánh giá độ khó khi xem lịch sử in thế nào ?
- Kết quả: Độ khó trung bình 3.0.

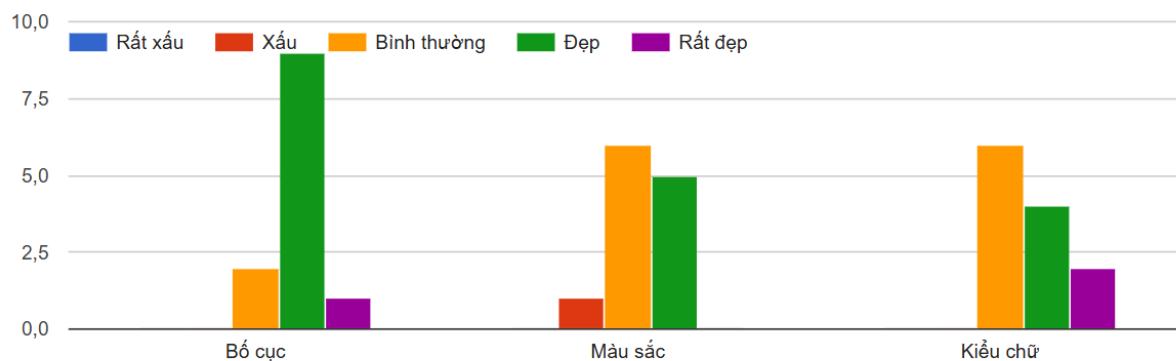


Hình 41: Biểu đồ đánh giá độ khó khi SPSS xem lịch sử in

Câu hỏi 5.2: Trên thang điểm từ 1 - 5, bạn đánh giá giao diện xem lịch sử in thế nào?

Kết quả:

- Bố cục: 16.67% Bình thường, 75% Đẹp và 8.33% Rất Đẹp.
- Màu sắc: 50% Bình thường, 41.7% Đẹp và 8.3% Xấu.
- Font chữ: 50% Bình thường, 33.3% Đẹp và 16.7% Rất đẹp.



Hình 42: Biểu đồ đánh giá tính thẩm mỹ giao diện SPSS xem lịch sử in

Góp ý kinh nghiệm :

- Nên cho phép search hoặc dropdown. Cho phép lọc theo trạng thái đơn in.

4.2.4 Kết quả và đề xuất

- Kết quả định lượng trên thang điểm 5:

Nhận xét:

- Tỷ lệ thành công khi thực hiện một tác vụ là 100 %: Tất cả đối tượng tham gia thử nghiệm có thể thực hiện thành công tác vụ yêu cầu.
- Dộ khó thực hiện các tác vụ dao động từ 2.83 đến 3.5 điểm trên thang điểm 5, trong đó tác vụ dễ thực hiện nhất là Nhân viên in (PS) cập nhật trạng thái đơn in và khó thực hiện nhất là Người dùng mua thêm trang in.



STT	Tác vụ	Độ khó thực hiện	Bố cục	Màu sắc	Kiểu chữ
1	Tạo đơn in lấy liền	3.0	3.58	3.67	3.5
2	Mua thêm trang in	3.5	3.98	3.42	3.58
3	Sinh viên xem lịch sử in	3.17	3.42	3.42	3.58
4	PS cập nhật đơn in	2.83	3.83	3.58	3.42
5	SPSS xem lịch sử in	3.0	3.92	3.58	3.67

Bảng 10: Kết quả đánh giá định lượng giao diện người dùng

- Các tiêu chí về thẩm mỹ giao diện người dùng (bố cục, màu sắc, kiểu chữ) đều có điểm trung bình từ 3.42 trở lên.

2. Kết quả định tính:

- Một số trang giao diện chưa thể hiện hết thông tin cần thiết, thuận tiện cho người dùng: Trang xác nhận giao dịch in cần có các thông số in, nhân viên in PS cần xem thông tin chi tiết đơn in.
- Cần bổ sung chức năng: Lọc trạng thái để tìm kiếm đơn in (Lịch sử đơn in).
- Một số chức năng bị ẩn trong menu (nút setting) và khó tìm kiếm (như chức năng Mua trang in).

Nhìn chung, giao diện người dùng MPV1 Hệ thống in án Sinh viên HCMUT_SPSS về cơ bản đáp ứng được nhu cầu sử dụng của người dùng, tương đối dễ sử dụng. Giao diện được chấm từ 3.42 điểm là đủ thân thiện và tính thẩm mỹ. Một số đề xuất cần xem xét sửa đổi sau khi nghiệm thử là:

- Thêm tính năng lọc theo trạng thái đơn in cho lịch sử đơn in.
- Cho phép PS nhấn mở để xem toàn bộ thông tin đơn in.

5 Implementation sprint 2

5.1 Hiện thực MVP2

```
ldkhang@ldkhangs-MacBook-Air student-smart-printing-service % git log --graph --oneline --decorate
* 4076c18 (HEAD -> main, origin/main, origin/HEAD) Merge pull request #36 from blueonline07/dev
| \
| * ba56efe (origin/dev, dev) Merge pull request #34 from blueonline07/feat/be/config
| | \
| | * fbd2a7e (origin/feat/be/config, feat/be/config) feat: add status query to getAll
| | * 2ce66b9 feat: config system
| | * c413ceb Merge pull request #33 from blueonline07/feat/fe/web/printing
| | \
| | * 326e878 fix : change printerData
| | * b41617a fix : change form layout
| | \
| * 6d75b5a Merge pull request #31 from blueonline07/feat/fe/web/role
| | \
| | * e3a8c77 feat: add role
| | \
| * ba78f26 Merge pull request #29 from blueonline07/style/fe/web
| | \
| | * 8978100 style: style pages
| | \
| * a2dbbac Merge pull request #28 from blueonline07/feat/fe/web/profile
| | \
| | * 771282f Merge branch 'dev' into feat/fe/web/profile
| | \
| | * b29b166 Merge pull request #27 from blueonline07/feat/fe/web/login
| | \
| | * f572d3b refactor: delete SelectUser Screen
| | \
| * 08a2c20 feat: add staff page
| | \
| * ae97436 Merge pull request #26 from blueonline07/feat/fe/web/print-history
```

Hình 43: Git graph của project

Nhóm đã hiện thực ứng dụng sử dụng ngôn ngữ lập trình javascript trên cả frontend và backend của ứng dụng. Quá trình hiện thực các thành viên tuân thủ các nguyên tắc và deadline. Kết quả hiện thực đã được demo trong buổi thuyết trình

5.2 Trình bày dự án



Hình 44: Link tới slide thuyết trình của nhóm

Nhóm đã thuyết trình trên lớp ngày 29/11/2024



6 Kết luận

Ứng dụng HCMUT SSPS (HCMUT Students' Smart Printing Service) được phát triển nhằm đáp ứng nhu cầu in ấn ngày càng cao và đa dạng của sinh viên trường Đại học Bách Khoa TP.HCM. Với mục tiêu mang lại một giải pháp hiện đại, tiện lợi và hiệu quả, ứng dụng kết hợp giữa Django ở phía backend và Flutter ở phía frontend, đảm bảo khả năng hoạt động ổn định và trải nghiệm người dùng mượt mà.

Quá trình phát triển ứng dụng tập trung vào việc tối ưu hóa quy trình in ấn, giúp sinh viên có thể dễ dàng đặt lệnh in từ xa, quản lý tài liệu và theo dõi trạng thái đơn in một cách nhanh chóng và minh bạch. Bên cạnh đó, tính năng quản lý hàng đợi in và thanh toán trực tuyến giúp tiết kiệm thời gian, tránh tình trạng ùn tắc tại các khu vực in ấn trong trường.

Nhóm phát triển tin rằng, ứng dụng không chỉ hỗ trợ sinh viên trong việc nâng cao hiệu quả học tập và làm việc nhóm thông qua quy trình in ấn thông minh mà còn tối ưu chi phí và thời gian nhờ các tính năng linh hoạt, hiện đại. Đồng thời, hệ thống cũng giúp nhà trường và quản lý dịch vụ in ấn theo dõi và cải thiện chất lượng dịch vụ.

Trong tương lai, chúng tôi sẽ tiếp tục mở rộng và bổ sung nhiều tính năng mới như tích hợp các công cụ chỉnh sửa tài liệu, thông báo tự động khi hoàn tất in ấn, và phân tích chi tiết lịch sử in ấn, nhằm mang đến trải nghiệm toàn diện và tiện ích nhất cho sinh viên Bách Khoa.