

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



Báo cáo
CÔNG NGHỆ PHẦN MỀM

Nhóm 4 - Lớp TN01 - HK231

Đề tài:

**Dịch vụ in thông minh cho sinh viên
HCMUT _ SSPS**

GVHD:	Trương Thị Thái Minh Bùi Công Tuấn Mai Đức Trung
SV thực hiện:	Võ Ngọc Thành Nhân 2114278 Lê Nguyễn Hải Đăng 2113176 Lê Dinh Huy 2113481 Phan Lê Nhật Minh 2114066 Trương Hoàng Nhật 2114303 Nguyễn Đình Quang 2112096 Phạm Đức Thắng 2112336

TP. HỒ CHÍ MINH, THÁNG 9/2023

Mục lục

I Thu thập nhu cầu	4
1 Tổng quan	4
1.1 Yêu cầu đề bài	4
1.2 Domain context	5
1.3 Các bên liên quan và nhu cầu của họ	5
1.4 Lợi ích của các bên liên quan	6
2 Yêu cầu chức năng và phi chức năng	7
2.1 Yêu cầu chức năng (functional)	7
2.1.1 Khách hàng (sinh viên, giảng viên, cán bộ...)	8
2.1.2 Student Printing Service Officer (SPSO)	8
2.1.3 Dịch vụ xác thực tài khoản HCMUT_SSO	9
2.1.4 Nhân viên IT	10
2.2 Yêu cầu phi chức năng (non-functional)	10
2.2.1 Hiệu suất	10
2.2.2 Tính khả dụng	11
2.2.3 Bảo mật	11
2.2.4 Dễ sử dụng	12
2.2.5 Tính pháp lý	13
2.2.6 Tính tương thích	13
2.2.7 Tính mở rộng	13
2.2.8 Da ngôn ngữ	13
3 Use-case diagram and Use-case scenario	13
3.1 Xác thực	15
3.1.1 Use case diagram	15
3.1.2 Use-case scenario	16
3.2 In	17
3.2.1 Use-case diagram	17
3.2.2 Use-case scenario	17
3.3 Quản lý máy in	20
3.3.1 Use-case diagram	20
3.3.2 Use-case scenario	20
3.4 Sửa danh mục định dạng tệp cho phép in	24
3.4.1 Use-case diagram	24
3.4.2 Use-case scenario	25
3.5 Cài đặt tặng giấy	26
3.5.1 Use-case diagram	26
3.5.2 Use-case scenario	27
3.6 Phản hồi	28
3.6.1 Use-case diagram	28
3.6.2 Use-case scenario	28
3.7 Lịch sử in	33
3.7.1 Use case diagram	33
3.7.2 Use-case scenario	33



II Mô hình hóa hệ thống	36
1 Activity diagrams	36
1.1 Xác thực	36
1.2 In	37
1.3 Mua giấy	38
1.4 Quản lý máy in	39
1.5 Sửa danh mục định dạng tệp cho phép in	40
1.6 Quản lý tặng giấy miễn phí	41
1.7 Lịch sử in	42
2 Sequence diagrams	44
2.1 Xác thực	44
2.2 In	45
2.3 Quản lý máy in	47
2.4 Sửa danh mục định dạng tệp cho phép in	49
2.5 Quản lý tặng giấy miễn phí	50
2.6 Lịch sử in	51
3 Class diagrams	54
3.1 Xác thực	55
3.2 In	56
3.3 Quản lý máy in	57
3.4 Quản lý in ấn (bao gồm định dạng tệp cho phép in và quản lý tặng giấy)	58
3.5 Lịch sử in	59
4 MVP	59
III Thiết kế kiến trúc	60
1 Kiến trúc MVC	60
1.1 Ưu và nhược điểm của MVC	60
1.2 Lí do lựa chọn MVC thay cho Layered architecture	61
2 Architectural diagram và Deployment diagram	62
2.1 Architectural diagram cho toàn hệ thống	62
2.2 Deployment diagram cho toàn hệ thống	63
2.2.1 Bản vẽ Deployment Diagram	63
2.2.2 Mô tả Deployment Diagram	64
3 Component diagram	65
3.1 Xác thực	65
3.2 In	66
3.3 Quản lý máy in	67
3.4 Sửa danh mục định dạng tệp cho phép & Quản lý tặng giấy miễn phí	68
3.5 Lịch sử in	69
IV Hiện thực - Sprint1	70
1 Cài đặt repository (github, bitbucket, etc) cho việc quản lý phiên bản.	70
2 Tài liệu và thư mục cho Requirement, System modelling và Architectural design.	70
3 Usability test bằng MVP1. Tổng hợp phản hồi và cải thiện	73
3.1 Tiến hành kiểm tra việc sử dụng với UI được phát triển trong MVP1	73
3.1.1 Trang chủ	74
3.1.2 Đăng nhập	74
3.1.3 In ngay	75
3.1.4 Xem trước và cấu hình file in	75



3.1.5	Quản lý máy in	76
3.1.6	Thêm máy in	76
3.1.7	Lịch sử in ấn	77
3.1.8	Mua giấy in	77
3.2	Tổng hợp phản hồi và cải thiện MVP1 thành MVP2 với trải nghiệm người dùng tốt hơn	77
V	Hiện thực - Sprint2	82
1	Các framework sử dụng	82
1.1	Front-end	82
1.2	Back-end	83
2	Demo sản phẩm	84
VI	Tổng kết	85
Tài liệu		86



I Thu thập nhu cầu

1 Tổng quan

1.1 Yêu cầu đề bài

Dưới đây là mô tả hệ thống Dịch vụ in thông minh dành cho sinh viên tại HCMUT:

- Mục tiêu:

- Xây dựng Dịch vụ in thông minh dành cho sinh viên (HCMUT_SSPPS) tại Trường đại học HCMUT.

- Cơ cấu hệ thống:

- Hệ thống bao gồm một số máy in trải rộng trong khuôn viên trường.
- Mỗi máy in có thông tin định danh gồm ID, tên thương hiệu/nhà sản xuất, kiểu máy in, mô tả ngắn gọn và địa điểm (bao gồm tên trường, tên tòa nhà, và số phòng).

- Chức năng in ấn:

- Sinh viên có thể tải tài liệu lên hệ thống và lựa chọn máy in.
- Sinh viên có thể chỉ định các thuộc tính in như khổ giấy, số trang, in một/hai mặt, số lượng bản sao, vv.
- Loại tệp tài liệu được giới hạn và định cấu hình bởi Nhân viên Dịch vụ In ấn Sinh viên (SPSO).

- Ghi log và lịch sử in ấn:

- Hệ thống ghi lại các thao tác in của tất cả sinh viên, bao gồm thông tin mã sinh viên, mã máy in, tên file, thời gian bắt đầu và kết thúc in, số trang cho từng khổ trang.

- Quản lý lịch sử in ấn:

- SPSO có quyền xem lịch sử in ấn của tất cả sinh viên hoặc một sinh viên cụ thể trong khoảng thời gian xác định và trên một hoặc nhiều máy in.
- Sinh viên cũng có quyền xem lịch sử in của họ cùng với tổng số trang đã in cho từng khổ trang.

- Quản lý số trang in và thanh toán:

- Mỗi học kỳ, sinh viên được cấp một số trang in khổ A4 mặc định.
- Sinh viên có thể mua thêm trang in thông qua tính năng Mua trang in và thanh toán trực tuyến qua các hệ thống như BKPay của trường.
- Hệ thống chỉ cho phép in số trang không vượt quá số dư tài khoản của sinh viên. *Lưu ý rằng trang A3 tương đương với hai trang A4.*

- Quản lý máy in:

- SPSO có quyền quản lý máy in như thêm, bật, tắt máy in.
- SPSO có quyền quản lý các cấu hình khác của hệ thống như số trang mặc định, ngày tháng cấp số trang mặc định cho sinh viên, và các loại tệp được hệ thống chấp nhận.



- **Báo cáo và lưu trữ:**

- Hệ thống tạo tự động các báo cáo về việc sử dụng hệ thống in vào cuối mỗi tháng và mỗi năm, và lưu trữ chúng trong hệ thống để SPSO có thể xem bất cứ lúc nào.

- **Xác thực người dùng:**

- Tất cả người dùng phải được xác thực bằng dịch vụ xác thực HCMUT_SSO trước khi sử dụng hệ thống.

- **Giao diện người dùng:**

- Hệ thống được cung cấp thông qua ứng dụng dựa trên web và ứng dụng di động để đáp ứng nhu cầu sử dụng trên nhiều nền tảng.

1.2 Domain context

Hiện tại, số lượng sinh viên và cán bộ nhân viên trong trường rất đông, bên cạnh đó nhu cầu in ấn tài liệu cũng tăng cao. Tuy nhiên trong khuôn viên cả trường ở 2 cơ sở chỉ có 3, 4 điểm thực hiện việc in ấn tài liệu, do đó tồn tại rất nhiều các bất cập như sau:

- Đến thời gian cao điểm như thi giữa kì, cuối kì nhu cầu in ấn tăng đột biến, dẫn đến mỗi lần thực hiện in ấn phải xếp hàng chờ lâu, không chủ động được thời gian cho mỗi cá nhân.
- Vì lý thuộc vào vấn đề nhân công khi cần người điều chỉnh bản in cho phù hợp nên số lượng máy bị giới hạn.
- Thời gian làm việc của các nơi in ấn bị giới hạn trong các khoảng thời gian nhất định trong ngày.
- Không chủ động trong việc điều chỉnh các cấu hình của trang in vì có thể người thực hiện việc in ấn hiểu sai các yêu cầu của khách hàng.

Nhận thấy những vấn đề bất cập đã nêu trên nên hệ thống in thông minh được ra đời (HCMUT_SSPS) để phục vụ cũng như giải quyết một phần các vấn đề trên.

1.3 Các bên liên quan và nhu cầu của họ

- **Sinh viên và cán bộ (người dùng cuối):**

- Hệ thống in phải đủ sự tiện lợi để có thể in một cách nhanh chóng từ nhiều địa điểm khác nhau trong khuôn viên trường
- Có thể xem lịch sử in
- Hệ thống phải có giao diện thân thiện, có thể được truy cập từ web hoặc mobile app
- Chi phí in rẻ và thời gian chờ đợi máy in nhanh.

- **Phòng đào tạo - Ban giám hiệu:**

- Hệ thống phải phục vụ được nhu cầu cho toàn bộ sinh viên nhà trường và cán bộ giáo viên
- Hệ thống phải được kết nối 2 cơ sở
- Hoạt động vào khoảng thời gian 6h30 - 20h30 từ thứ 2 đến thứ 7 hàng tuần
- Hệ thống phải đảm bảo không vi phạm các chính sách của nhà trường.



• Phòng tài chính:

- Các giao dịch thông qua hệ thống đều được liên kết với tài khoản OCB để có thể dễ dàng xem được lịch sử các lần giao dịch
- Xem được các báo cáo tài chính mỗi tháng.

• SPSO:

- Có trách nhiệm quản lý tài khoản và số lượng giấy in của mỗi sinh viên
- Kiểm tra tình trạng máy in, có thể hoàn trả lại máy in nếu xảy ra lỗi
- SPSO có thể kiểm duyệt được nội dung của những tài liệu mà sinh viên muốn in để có thể đảm bảo không in những tài liệu không được phép
- SPSO có thể tùy chỉnh số lượng giấy in mặc định phát cho mỗi sinh viên vào mỗi đầu học kỳ
- SPSO có thể tiếp nhận những phản hồi của sinh viên và cán bộ giáo viên về những sự bất tiện và những lỗi của máy in.

• IT Staff:

- Có thể tắt hệ thống để bảo trì
- Có thể tắt một số tính năng nhất định để kiểm tra lỗi
- Có thể chặn các sinh viên đã vi phạm quy định khi xài máy in hoặc in những tài liệu không phù hợp
- Có thể gửi tin nhắn thông báo về hệ thống.

• Ethic manager (người kiểm duyệt nội dung):

- Có thể xem lịch sử in.
- Có thể tự động kiểm duyệt được nội dung trước khi in, cần 1 phút.

1.4 Lợi ích của các bên liên quan

• Sinh viên và cán bộ (người dùng cuối):

- Được cung cấp dịch vụ in ấn thông minh, hoàn toàn tự động.
- Đăng nhập với một tài khoản duy nhất, không trùng lặp với người khác, cho phép cá nhân hóa việc in ấn.
- In ấn tài liệu với những chức năng tiện lợi như in nhiều khổ giấy, in màu, hẹn giờ tự động in, ...
- Được xem lịch sử in của bản thân
- Có quyền khiếu nại và được bồi thường nếu có lỗi xảy ra.
- Được đảm bảo bảo mật thông tin.
- Được hỗ trợ chăm sóc khách hàng, giải đáp thắc mắc mọi lúc.

• Phòng tài chính:

- Được truy xuất các báo cáo tài chính mỗi tháng.

• SPSO:



- Dễ dàng theo dõi, kiểm soát tình hình của hệ thống SPSO. Như xem một cách chi tiết lịch sử in, thông tin cá nhân khách hàng.
- Có quyền ngăn chặn in ấn tài liệu nhạy cảm, tài liệu bị cấm.
- Có quyền thống kê về lượng giấy đã in, số giấy được mua, bản báo cáo về hệ thống, sao kê hàng tháng.
- Được nghe đóng góp ý kiến, khiếu nại phản hồi từ khách hàng qua tính năng trên hệ thống

• **IT Staff:**

- Được quyền tắt bật hệ thống để kiểm tra, bảo trì.
- Được tắt một số tính năng nhất định để kiểm tra lỗi
- Được quyền chặn những tài khoản spam, có hành vi sai phạm bị phát hiện và tố cáo.
- Có quyền gửi tin nhắn thông báo hệ thống.

• **Ethic manager (người kiểm duyệt nội dung - quản trị đạo đức):**

- Được xem nội dung in của người sử dụng
- Hủy thao tác in trực tiếp
- Báo cáo tài khoản có hành vi xấu

2 Yêu cầu chức năng và phi chức năng

2.1 Yêu cầu chức năng (functional)

Các yêu cầu chức năng được phân loại theo từng stakeholder.



2.1.1 Khách hàng (sinh viên, giảng viên, cán bộ...)

Stakeholders	Mã số yêu cầu	Tên yêu cầu	Yêu cầu chức năng
Khách hàng	KH01	Log in, Log out	Khách hàng đăng nhập và đăng xuất bằng các thông tin hợp lệ (email, mật khẩu).
	KH02	Configure	Khách hàng có thể tải các tài liệu cũng như có thể cài đặt cấu hình in (khổ giấy, số lượng trang cần in, số mặt, số bản copy).
	KH03	Select printers	Khách hàng có thể lựa chọn máy in và hẹn giờ in.
	KH04	View history	Khách hàng có thể truy cập được lịch sử in ấn, cùng một bản thống kê số trang và kích thước mỗi trang.
	KH05	Limit page	Khách hàng chỉ được phép in khi số trang không vượt quá số trang còn lại trong tài khoản.
	KH06	Provide default pages	Mỗi học kì, khách hàng sẽ được cấp mới lại số trang mặc định.
	KH07	Buy paper	Khách hàng có quyền lựa chọn mua thêm giấy và thanh toán thông qua các nền tảng giao dịch trực tuyến.
	KH08	Report & Feedback	Khách hàng có thể cung cấp các khiếu nại và đánh giá về dịch vụ in. Cũng như có thể xóa và trả lời phản hồi của SPSO.
	KH09	Private Information	Cần phải đảm bảo sự riêng tư về nội dung in của khách hàng.

2.1.2 Student Printing Service Officer (SPSO)



Stakeholders	Mã số yêu cầu	Tên yêu cầu	Yêu cầu chức năng
SPSO	SO01	Log in, Log out	SPSO đăng nhập và đăng xuất bằng các thông tin xác thực hợp lệ.
	SO02	Control printers	SPSO có khả năng thêm, bật tắt máy in.
	SO03	View conditions	SPSO có thể xem được tình trạng các máy in.
	SO04	Regulate Type	SPSO có thể quy định các loại tệp mà khách hàng được phép tải lên để in.
	SO05	Change amount of paper	SPSO có thể thay đổi số trang mặc định và thời gian cập nhật số trang mặc định.
	SO06	View history	SPSO có thể xem được lịch sử in của khách hàng.
	SO07	View reports	SPSO có thể xem được các báo cáo, khiếu nại của khách hàng. Và có thể trả lời khiếu nại.
	SO08	Sell paper	SPSO có thể bán giấy cho khách hàng có nhu cầu.
	SO09	Ban users	SPSO có khả năng cấm truy cập của khách hàng có hành vi phá hoại.
	SO10	View revenue	SPSO có thể xem được doanh thu của dịch vụ in ấn.
	SO11	Time waiting	SPSO cần người dùng chờ một khoảng thời gian để kiểm duyệt nội dung tùy vào số lượng trang in.
	SO12	Classify	SPSO cần phân loại các ấn phẩm in để dễ dàng kiểm duyệt.
	SO13	Provide AI tools	SPSO cần tích hợp các tool hoặc agent AI để hỗ trợ trong việc kiểm duyệt.

2.1.3 Dịch vụ xác thực tài khoản HCMUT_SSO



Stakeholders	Mã số yêu cầu	Tên yêu cầu	Yêu cầu chức năng
HCMUT_SSO	SSO01	Verify users	Xác minh danh tính người dùng trước khi cấp quyền cho các tài nguyên.
	SSO02	Add users	Thêm người dùng mới vào hệ thống.
	SSO03	Delete users	Xóa tài khoản người dùng khỏi hệ thống.
	SSO04	Warning users	Cảnh báo, cấm hoặc xóa tài khoản người dùng khi cần thiết.
	SSO05	Change password	Cho phép người dùng thay đổi mật khẩu.
	SSO06	Classify users	Phân loại khách hàng theo chức vụ (sinh viên, cán bộ, giảng viên).

2.1.4 Nhân viên IT

Stakeholders	Mã số yêu cầu	Tên yêu cầu	Yêu cầu chức năng
IT Staff	IT01	Maintain system	Nhân viên IT có thể bật tắt hệ thống phục vụ việc bảo trì.
	IT02	Restrict users	Nhân viên IT có thể chặn quyền truy cập của khách hàng.
	IT03	Send messages	Nhân viên IT có thể gửi tin nhắn tới khách hàng.

2.2 Yêu cầu phi chức năng (non-functional)

Để đảm bảo đáp ứng tốt trải nghiệm của người dùng, trang web cần đáp ứng được các yêu cầu sau:

2.2.1 Hiệu suất

Cần đảm bảo được các yêu cầu về hiệu suất sau:



Tên yêu cầu	Mã số yêu cầu	Chi tiết yêu cầu
Hiệu suất	HS01	Yêu cầu phần cứng tối thiểu cho hệ thống là 128MB Ram.
	HS02	Thiết bị thông minh hỗ trợ kết nối Internet, quét mã QR để truy cập vào ứng dụng mà không cần tải app.
	HS03	Hệ thống hỗ trợ chạy realtime.
	HS04	Dảm bảo thời gian xử lý và tải thông tin cho mỗi lần người dùng thực hiện 1 thao tác tối đa 2 giây.
	HS05	Lượt truy cập đạt 100,000 lượt tại cùng 1 thời điểm, đáp ứng nhu cầu của 30,000 sinh viên và 5,000 cán bộ nhân viên.
	HS06	Tỉ lệ xảy ra lỗi khi người dùng thực hiện 1 thao tác thấp hơn 0.1%.
	HS07	Bảo trì trang web mỗi tháng 1 lần và có thông báo cho người dùng về khoảng thời gian bảo trì.

2.2.2 Tính khả dụng

Tên yêu cầu	Mã số yêu cầu	Chi tiết yêu cầu
Tính khả dụng	KD01	Số lượng record về máy in trong dữ liệu tầm 2 record cho mỗi tờ.
	KD02	Phải liên kết được giữa 2 cơ sở, để người dùng có thể in ở bất cứ cơ sở nào.
	KD03	Cho phép in từ 6h30-20h30 mỗi ngày. Ngoài khoảng thời gian trên người dùng có thể hẹn lịch in. Hệ thống vận hành từ 6h-23h59.
	KD05	Thông báo cho người dùng khi có sự cố (mất điện, máy in bị trục trặc)

2.2.3 Bảo mật

Về phía người dùng:



Tên yêu cầu	Mã số yêu cầu	Chi tiết yêu cầu
Bảo mật người dùng	US01	Mật khẩu yêu cầu ít nhất 8 ký tự, phải bao gồm chữ cái viết hoa, ký tự đặc biệt và chữ số.
	US02	Tính năng "Quên mật khẩu" bằng cách xác nhận thông qua mã OTP gửi về email hoặc số điện thoại.
	US03	Trường hợp người dùng đăng nhập sai quá 5 lần sẽ tự động khóa tài khoản, để mở khóa sẽ cần liên hệ quản trị viên.
	US04	Cam kết bảo mật thông tin cho người dùng, không bán thông tin cho bất cứ bên nào.

Về phía hệ thống:

Tên yêu cầu	Mã số yêu cầu	Chi tiết yêu cầu
Bảo mật hệ thống	SS01	Các cookies gửi kèm với header của request từ phía client hay được gửi kèm với response từ phía server đều được mã hóa bằng Json Web Token (JWT) trước khi gửi.
	SS02	Mã hóa các loại mật khẩu gồm mật khẩu của tài khoản quản trị viên,... bằng thuật toán hash "sha256" trước khi lưu vào dữ liệu.

2.2.4 Dễ sử dụng

Để đảm bảo người dùng mới có thể sử dụng trang web mà không cần tốn nhiều công sức với chỉ trong 30 phút sau training đối với SPSO và có thể dùng ngay lập tức đối với người dùng thông thường thì trang web cần đảm bảo các yêu cầu sau:

Tên yêu cầu	Mã số yêu cầu	Chi tiết yêu cầu
Dễ sử dụng	SD01	Thiết kế đơn giản, đẹp mắt, phù hợp với đối tượng sinh viên, cán bộ nhân viên của 1 trường đại học, tương tự như BKEL.
	SD02	Không có quảng cáo.
	SD03	Các chức năng cần cung cấp thông tin rõ ràng về chức năng để tránh gây hiểu lầm.
	SD04	Các tùy chọn phải có màu sắc phân biệt, khoảng cách không quá sát nhau (cách nhau tối thiểu 5mm) để tráng trượt tay khi chọn.



2.2.5 Tính pháp lý

Tên yêu cầu	Mã số yêu cầu	Chi tiết yêu cầu
Tính pháp lý	PL01	Cần phải tuân thủ các chính sách của nhà trường đưa ra trong hợp đồng.
	PL02	Không vi phạm bản quyền, thương hiệu của bên thứ ba.

2.2.6 Tính tương thích

Tên yêu cầu	Mã số yêu cầu	Chi tiết yêu cầu
Tính tương thích	TT01	Có khả năng vận hành trên nhiều trình duyệt web như Google Chrome, Microsoft Edge, FireFox, cũng như trên các hệ điều hành Window ((Window 10 22H2, Window 11 22H2), Android (Android version 13 trở lên) miễn là có kết nối Internet.

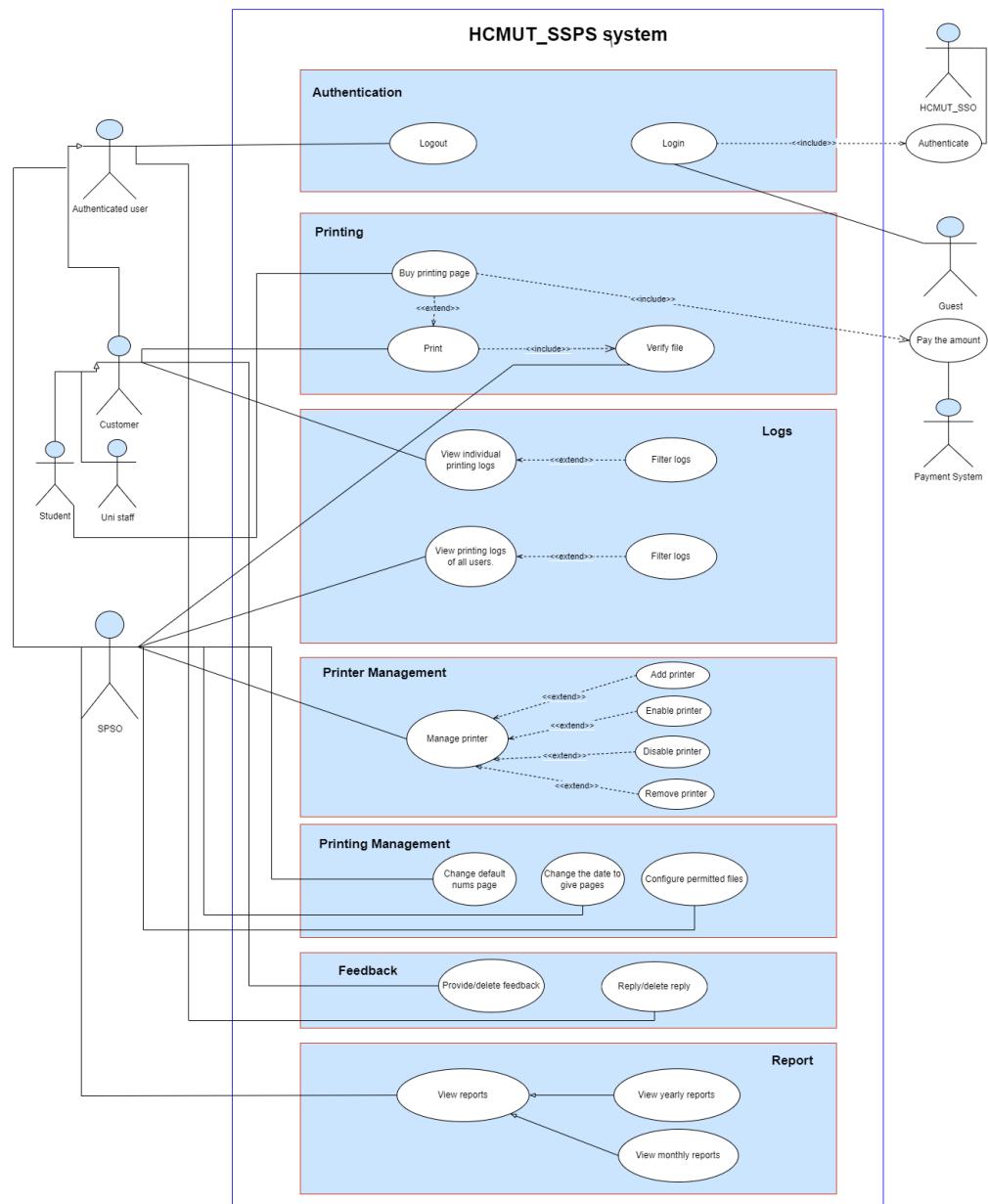
2.2.7 Tính mở rộng

Tên yêu cầu	Mã số yêu cầu	Chi tiết yêu cầu
Tính mở rộng	MR01	Mỗi năm có số lượng sinh viên mới khoảng 3000 sinh viên, do đó hệ thống cần đáp ứng được lượng người dùng mới này.
	MR02	Dễ bảo trì, dễ dàng mở rộng để có thể thêm các chức năng và tính năng mới trong tương lai.

2.2.8 Đa ngôn ngữ

Tên yêu cầu	Mã số yêu cầu	Chi tiết yêu cầu
Đa ngôn ngữ	ML01	Cần hỗ trợ chuyển đổi ngôn ngữ giữa tiếng Anh và tiếng Việt để phù hợp với nhu cầu người dùng.

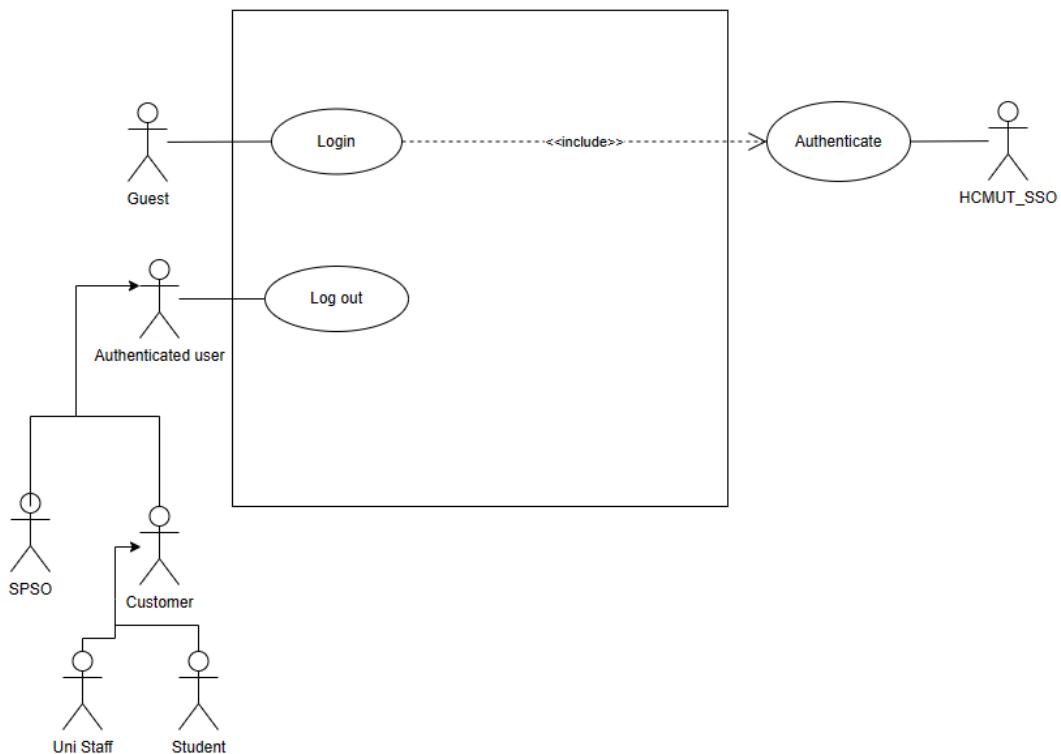
3 Use-case diagram and Use-case scenario



Hình 1: Use-case diagram cho hệ thống

3.1 Xác thực

3.1.1 Use case diagram



Hình 2: Use-case diagram cho chức năng xác thực tài khoản



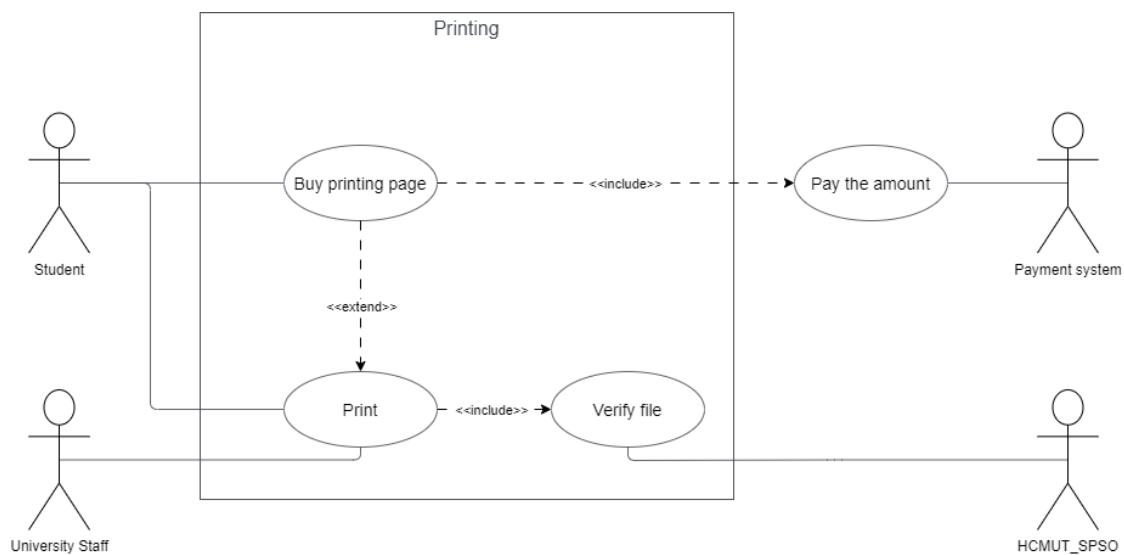
3.1.2 Use-case scenario

Use-case Login

Use case name:	Login		
Created by:	Nguyễn Đình Quang	Last updated by:	Nguyễn Đình Quang
Date created:	28/09/2023	Date last updated:	30/09/2023
Actors:	Authenticated User (SPSO và customer)		
Description:	Use case này cho phép Guest đăng nhập vào hệ thống		
Trigger:	Click vào nút "Login" trên giao diện chính của website.		
Preconditions:	Guest chưa đăng nhập vào hệ thống Guest dùng có tài khoản trên ứng dụng Thiết bị của Guest có kết nối mạng		
Postconditions:	Guest đăng nhập thành công.		
Normal Flows:	1. Guest chọn "Login". 2. Hệ thống hiển thị giao diện đăng nhập. 3. Chọn đối tượng đăng nhập. 4. Guest nhập username và password. 5. Guest nhấn nút "Log in". 6. Hệ thống xác thực thông tin đăng nhập. 7. Hệ thống cập nhật lại giao diện theo thông tin của tài khoản Guest.		
Alternative Flows:			
Exceptions:	E1: Tại bước 6 6a: Guest nhập Username/ Password sai. 6b: Hệ thống hiển thị thông báo sai thông tin đăng nhập. Quay lại bước 3 trong normal flows.		
Note and issues:	Nếu User hiện đang đăng nhập, sau đó tiếp tục đăng nhập vào hệ thống bằng thiết bị khác, thiết bị cũ sẽ không tự động bị đăng xuất		

3.2 In

3.2.1 Use-case diagram



Hình 3: Use-case diagram cho chức năng in tài liệu

3.2.2 Use-case scenario



Use-case Print

Use case name:	Print		
Created by:	Trương Hoàng Nhật	Last updated by:	Trương Hoàng Nhật
Date created:	27/09/2023	Date last updated:	27/09/2023
Actors:	Student, University staff		
Description:	Cho phép người dùng upload và in tài liệu sau khi đã được kiểm duyệt		
Trigger:	Người dùng nhấn vào nút "In ngay" ở thanh điều hướng		
Preconditions:	<ul style="list-style-type: none">- Người dùng đã có tài khoản trên hệ thống- Người dùng tải lên tài liệu đã được kiểm duyệt- Người dùng còn đủ số lượng giấy in trong tài khoản- Người dùng đã lựa chọn thời gian, vị trí máy in, cấu hình in.		
Postconditions:	Người dùng sẽ được xếp vào hàng chờ của máy in đã chọn		
Normal Flows:	<ol style="list-style-type: none">1. Người dùng chọn "In ngay" tại thanh điều hướng2. Hệ thống hiển thị giao diện để người dùng chọn máy in3. Người dùng lọc các thông tin của máy in (cơ sở, tòa, phòng)4. Hệ thống hiện danh sách các máy in khả dụng và số lượng người đang chờ trong hàng đợi của mỗi máy in5. Người dùng chọn vào máy cần in6. Người dùng upload file cần in lên hệ thống7. Hệ thống tiến hành kiểm duyệt file vừa được up8. Người dùng chọn thời gian in, định dạng cách thức in9. Người dùng xác nhận thao tác in		
Alternative Flows:	<p>A1: Tại bước 1</p> <ol style="list-style-type: none">1.1: Người dùng chưa đăng nhập vào hệ thống1.2: Hệ thống chuyển người dùng đến trang đăng nhập1.3: Người dùng đăng nhập vào hệ thống <p>Tiếp tục bước 2 trong Normal Flows.</p> <p>E1: Tại bước 4</p> <ol style="list-style-type: none">4.1: Tất cả máy in thỏa mãn điều kiện người dùng lọc đều đang được bảo trì.4.2: Hệ thống báo lỗi đến người dùng <p>E2: Tại bước 7</p> <ol style="list-style-type: none">7.1: Hệ thống kiểm duyệt file upload không hợp lệ7.2: Hệ thống báo lỗi đến người dùng <p>E3: Tại bước 8</p> <ol style="list-style-type: none">8.1: Hệ thống kiểm tra thấy số lượng giấy in còn lại trong tài khoản nhỏ hơn số lượng trang người dùng muốn in8.2: Hệ thống báo lỗi đến người dùng8.3: Chuyển sang Use-case "Buy printing page"		
Note and issues:	Không có		

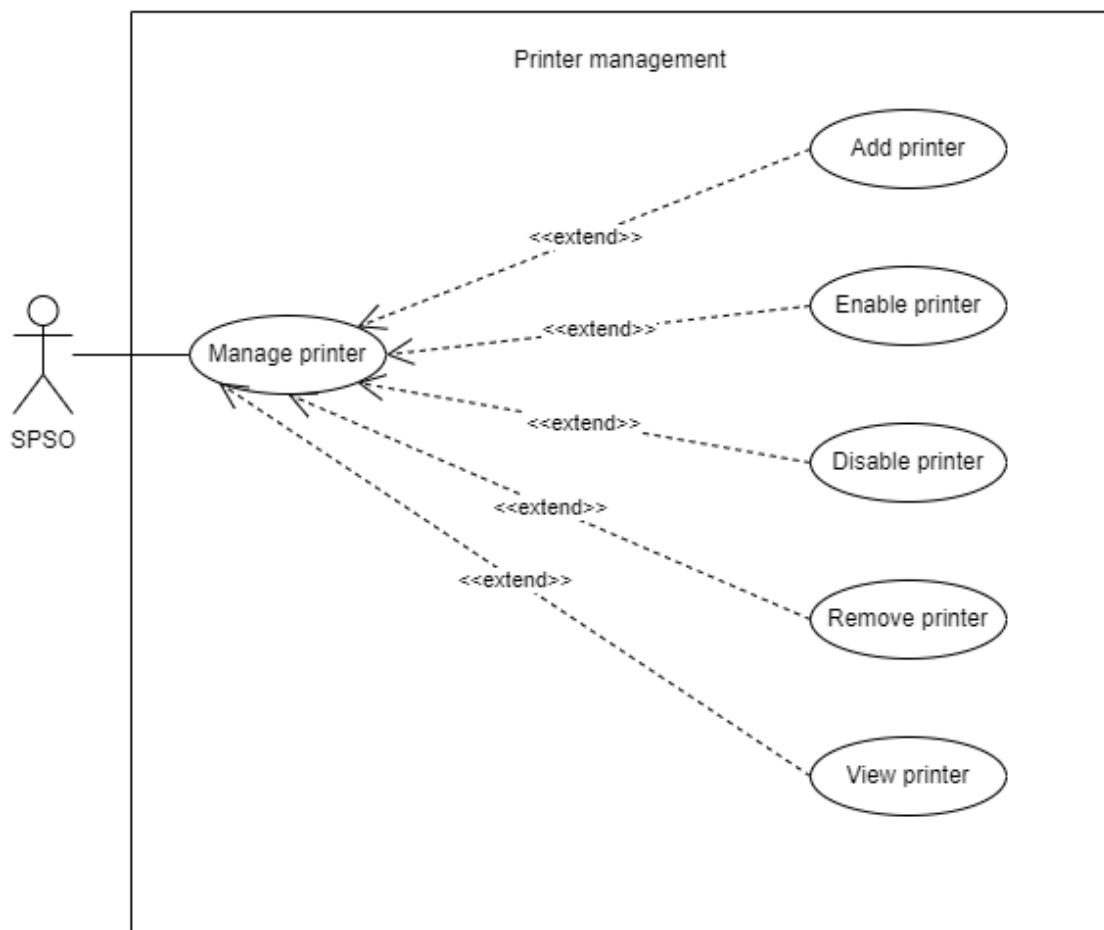


Use-case Buy printing page

Use case name:	Buy printing page		
Created by:	Trương Hoàng Nhật	Last updated by:	Trương Hoàng Nhật
Date created:	27/09/2023	Date last updated:	27/09/2023
Actors:	Student		
Description:	Cho phép người dùng mua thêm giấy in trong tài khoản		
Trigger:	Người dùng nhấn vào nút "Mua thêm giấy" trong giao diện "In ngay"		
Preconditions:	<ul style="list-style-type: none">- Người dùng đã có tài khoản và đã đăng nhập vào hệ thống- Tài khoản người dùng đã liên kết với tài khoản BK Pay.		
Postconditions:	Số lượng giấy trong tài khoản của người dùng tăng lên.		
Normal Flows:	<ol style="list-style-type: none">1. Người dùng chọn "In ngay" tại thanh điều hướng2. Hệ thống hiển thị giao diện của tính năng "In ngay"3. Người dùng nhấn vào "Mua thêm giấy"4. Hệ thống hiển thị giao diện để người dùng mua giấy in5. Người dùng nhập số lượng giấy muốn mua và chọn phương thức thanh toán, sau đó nhấn vào "Mua ngay"6. Hệ thống xác nhận thanh toán		
Alternative Flows:	Không có		
Exceptions:	E1: Tài bước 6 6.1: Xử lý thanh toán thất bại 6.2: Hệ thống báo lỗi đến người dùng.		
Note and issues:	Người dùng cũng có thể mua giấy lúc hệ thống báo lỗi thiếu giấy khi in tài liệu.		

3.3 Quản lý máy in

3.3.1 Use-case diagram



Hình 4: Use-case diagram cho chức năng quản lý máy in

3.3.2 Use-case scenario

Manage printer



Use case name:	Manage printer		
Created by:	Lê Đình Huy	Last updated by:	Lê Đình Huy
Date created:	25/09/2023	Date last updated:	29/09/2023
Actors:	SPSO		
Description:	Cho phép SPSO quản lý các máy in cũng như là xem danh sách các máy in.		
Trigger:	Chọn nút "Quản lý máy in" tại thanh điều hướng.		
Preconditions:	<ul style="list-style-type: none"> - SPSO có tài khoản trên website - SPSO đã đăng nhập thành công vào hệ thống - Thiết bị của SPSO có kết nối mạng và kết nối với hệ thống - SPSO có quyền quản lý máy in 		
Postconditions:	Giao diện quản lý máy in hiển thị đối với SPSO.		
Normal Flows:	<ol style="list-style-type: none"> 1. SPSO chọn "Quản lý máy in" trên thanh điều hướng. 2. Hệ thống hiển thị giao diện quản lý máy in gồm danh sách các máy in và các nút thêm, xóa, xem, enable và disable. 		
Alternative Flows:	Không có		
Exceptions:	Không có		
Note and issues:	Không có		

Enable printer

Use case name:	Enable printer		
Created by:	Lê Đình Huy	Last updated by:	Lê Đình Huy
Date created:	25/09/2023	Date last updated:	29/09/2023
Actors:	SPSO		
Description:	Cho phép SPSO kích hoạt một máy in.		
Trigger:	Chọn nút "Enable" cho một máy in tại giao diện quản lý máy in.		
Preconditions:	<ul style="list-style-type: none"> - SPSO có tài khoản trên website - SPSO đã đăng nhập thành công vào hệ thống - Thiết bị của SPSO có kết nối mạng và kết nối với hệ thống - SPSO có quyền quản lý máy in 		
Postconditions:	Dữ liệu về máy in đã được update trong database và trên giao diện website.		
Normal Flows:	<ol style="list-style-type: none"> 1. SPSO chọn "Enable" cho một máy in. 2. SPSO xác nhận thao tác. 3. Hệ thống cập nhật dữ liệu trong database và website. 		
Alternative Flows:	Không có		
Exceptions:	<p>E1: Tại bước 2</p> <p>2.1 SPSO không xác nhận thay đổi. 2.2 Hệ thống xóa tất cả những thay đổi SPSO vừa thực hiện, giao diện vẫn giữ nguyên ban đầu. Use-case dừng lại.</p>		
Note and issues:	Không có		

Disable printer



Use case name:	Disable printer		
Created by:	Lê Đình Huy	Last updated by:	Lê Đình Huy
Date created:	25/09/2023	Date last updated:	29/09/2023
Actors:	SPSO		
Description:	Cho phép SPSO vô hiệu hóa một máy in.		
Trigger:	Chọn nút "Disable" cho một máy in tại giao diện quản lý máy in.		
Preconditions:	<ul style="list-style-type: none">- SPSO có tài khoản trên website- SPSO đã đăng nhập thành công vào hệ thống- Thiết bị của SPSO có kết nối mạng và kết nối với hệ thống- SPSO có quyền quản lý máy in		
Postconditions:	Dữ liệu về máy in đã được update trong database và trên giao diện website.		
Normal Flows:	<ol style="list-style-type: none">1. SPSO chọn "Disable" cho một máy in.2. SPSO xác nhận thao tác.3. Hệ thống cập nhật dữ liệu trong database và website.		
Alternative Flows:	Không có		
Exceptions:	<p>E1: Tại bước 2</p> <p>2.1 SPSO không xác nhận thay đổi. 2.2 Hệ thống xóa tất cả những thay đổi SPSO vừa thực hiện, giao diện vẫn giữ nguyên ban đầu. Use-case dừng lại.</p>		
Note and issues:	Không có		

Remove printer

Use case name:	Remove printer		
Created by:	Lê Đình Huy	Last updated by:	Lê Đình Huy
Date created:	25/09/2023	Date last updated:	29/09/2023
Actors:	SPSO		
Description:	Cho phép SPSO xóa một máy in.		
Trigger:	Chọn nút "Xóa" cho một máy in tại giao diện quản lý máy in.		
Preconditions:	<ul style="list-style-type: none">- SPSO có tài khoản trên website- SPSO đã đăng nhập thành công vào hệ thống- Thiết bị của SPSO có kết nối mạng và kết nối với hệ thống- SPSO có quyền quản lý máy in		
Postconditions:	Dữ liệu về máy in đã được update trong database và trên giao diện website.		
Normal Flows:	<ol style="list-style-type: none">1. SPSO chọn "Xóa" cho một máy in.2. SPSO xác nhận thao tác.3. Hệ thống cập nhật dữ liệu trong database và website.		
Alternative Flows:	Không có		
Exceptions:	<p>E1: Tại bước 2</p> <p>2.1 SPSO không xác nhận thay đổi. 2.2 Hệ thống xóa tất cả những thay đổi SPSO vừa thực hiện, giao diện vẫn giữ nguyên ban đầu. Use-case dừng lại.</p>		
Note and issues:	Không có		



Add printer

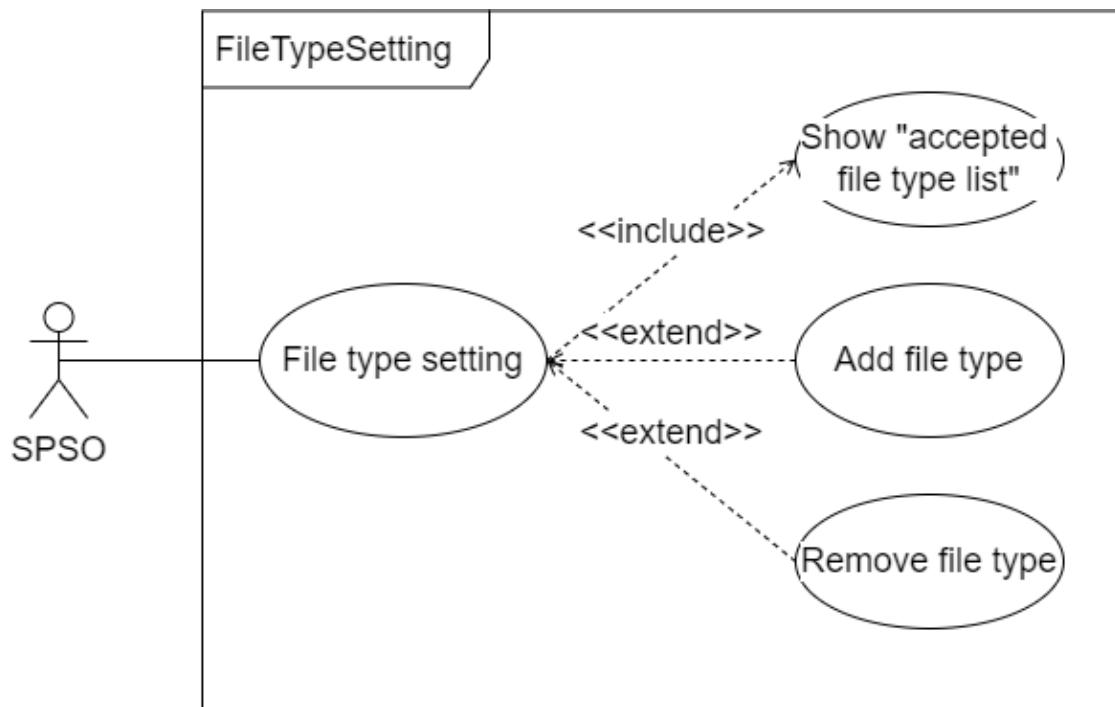
Use case name:	Add printer		
Created by:	Lê Đình Huy	Last updated by:	Lê Đình Huy
Date created:	25/09/2023	Date last updated:	29/09/2023
Actors:	SPSO		
Description:	Cho phép SPSO thêm một máy in.		
Trigger:	Chọn nút "Thêm máy in" tại giao diện quản lý máy in.		
Preconditions:	<ul style="list-style-type: none"> - SPSO có tài khoản trên website - SPSO đã đăng nhập thành công vào hệ thống - Thiết bị của SPSO có kết nối mạng và kết nối với hệ thống - SPSO có quyền quản lý máy in 		
Postconditions:	Dữ liệu về máy in đã được update trong database và trên giao diện website.		
Normal Flows:	<ol style="list-style-type: none"> 1. SPSO chọn "Thêm máy in" tại giao diện quản lý máy in. 2. Hệ thống hiển thị biểu mẫu các thông tin của máy in mới. 3. SPSO điền biểu mẫu. 4. SPSO xác nhận thao tác. 5. Hệ thống cập nhật dữ liệu trong database và website. 		
Alternative Flows:	Không có		
Exceptions:	<p>E1: Tại bước 4</p> <p>4.1 SPSO không xác nhận thay đổi.</p> <p>4.2 Hệ thống xóa tất cả những thay đổi SPSO vừa thực hiện, giao diện vẫn giữ nguyên ban đầu.</p> <p>Use-case dừng lại.</p>		
Note and issues:	Không có		

View printer

Use case name:	View printer		
Created by:	Lê Đình Huy	Last updated by:	Lê Đình Huy
Date created:	17/11/2023	Date last updated:	17/11/2023
Actors:	SPSO		
Description:	Cho phép SPSO xem thông tin của một máy in.		
Trigger:	Chọn nút "i" cho một máy in tại giao diện quản lý máy in.		
Preconditions:	<ul style="list-style-type: none"> - SPSO có tài khoản trên website - SPSO đã đăng nhập thành công vào hệ thống - Thiết bị của SPSO có kết nối mạng và kết nối với hệ thống - SPSO có quyền quản lý máy in 		
Postconditions:	Thông tin máy in được hiển thị trên giao diện của người dùng.		
Normal Flows:	<ol style="list-style-type: none"> 1. Chọn nút "i" cho một máy in tại giao diện quản lý máy in. 2. Hệ thống hiển thị các thông tin của máy in. 		
Alternative Flows:	Không có		
Exceptions:	Không có		
Note and issues:	Không có		

3.4 Sửa danh mục định dạng tệp cho phép in

3.4.1 Use-case diagram



Hình 5: Use-case diagram cho những tính năng liên quan đến in ấn

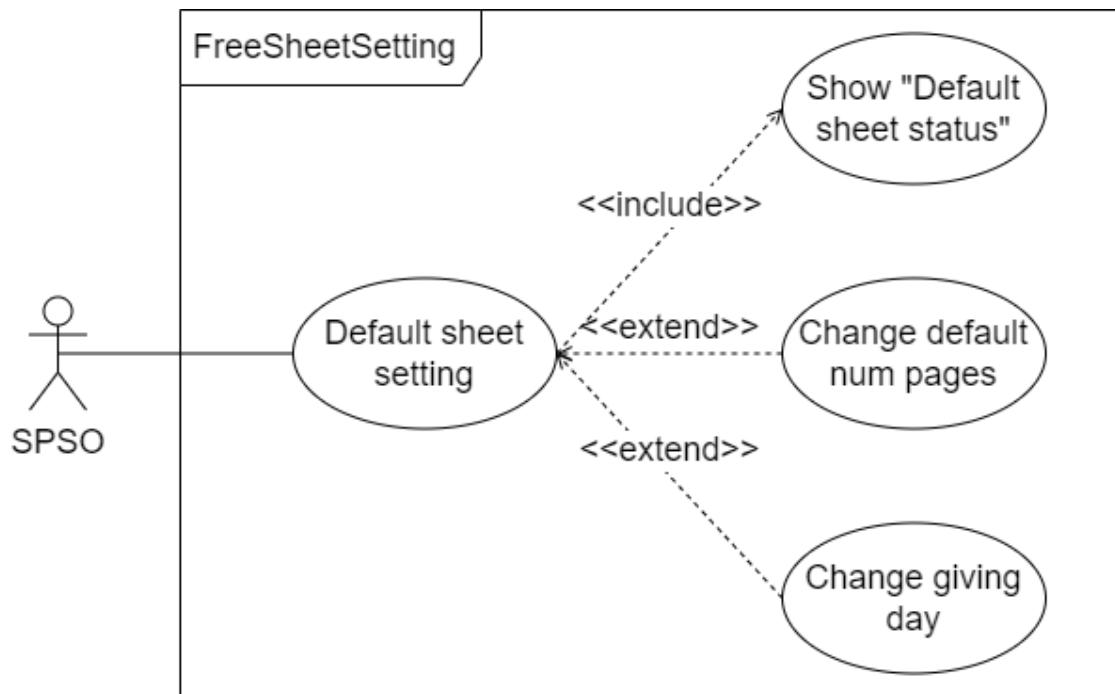


3.4.2 Use-case scenario

Use case name:	File type setting		
Created by:	Phan Lê Nhật Minh	Last updated by:	Phan Lê Nhật Minh
Date created:	28/09/2023	Date last updated:	28/09/2023
Actors:	SPSO		
Description:	SPSO dùng tính năng chỉnh sửa những định dạng tệp được phép sử dụng		
Trigger:	SPSO: Chọn nút "Cài đặt loại file in" tại giao diện quản lý in ấn.		
Preconditions:	<ul style="list-style-type: none">- SPSO có tài khoản trên website- SPSO đã đăng nhập thành công vào hệ thống với quyền hạn SPSO- Thiết bị của SPSO có kết nối mạng và kết nối với hệ thống		
Postconditions:	Dữ liệu về định dạng tệp in (file type) cho phép đã được update trong database.		
Normal Flows:	<ol style="list-style-type: none">1. SPSO chọn "Cài đặt loại file in" tại giao diện quản lý in ấn, hệ thống hiển thị những định dạng file hiện tại2. SPSO chọn "+", hệ thống hiển thị ô điền3. SPSO nhập file type mới gồm 3 chữ cái trở lên4. Hệ thống hiển thị form xác nhận và SPSO chọn Apply. Hệ thống xác nhận thay đổi, cập nhật lại database, quay lại giao diện cài đặt loại file in.		
Alternative Flows:	<p>A1: Tại bước 2 2.1 SPSO chọn vào một loại file in Tiếp tục tại bước 4 trong Normal Flows.</p>		
Exceptions:	<p>E1: Tại bước 4 4.1 SPSO chọn Apply, Hệ thống phát hiện thay đổi không hợp lệ. 4.2 Hệ thống xóa tất cả những thay đổi vừa thực hiện, giao diện vẫn giữ nguyên ban đầu. Quay lại bước 2.</p>		
Note and issues:	Không có		

3.5 Cài đặt tặng giấy

3.5.1 Use-case diagram



Hình 6: Use-case diagram cho tính năng cài đặt tặng giấy

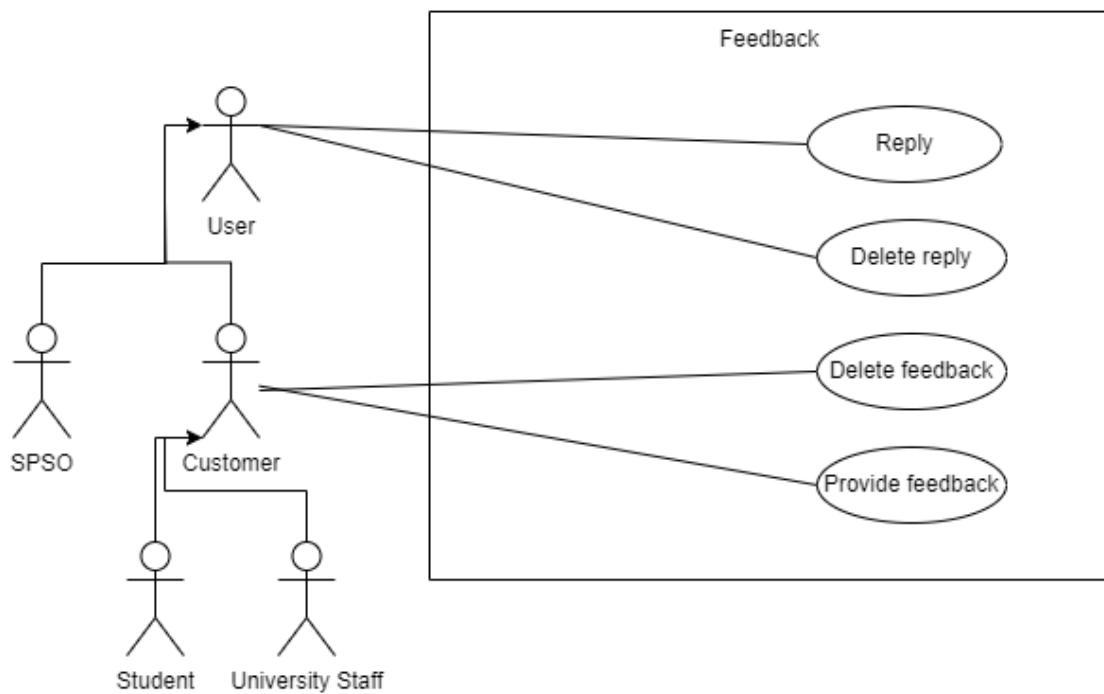


3.5.2 Use-case scenario

Use case name:	Free sheet gifting		
Created by:	Phan Lê Nhật Minh	Last updated by:	Phan Lê Nhật Minh
Date created:	28/09/2023	Date last updated:	28/09/2023
Actors:	SPSO		
Description:	SPSO có thể cài đặt các thông số cho việc tặng giấy cho customer.		
Trigger:	SPSO: Chọn nút "Cài đặt tặng giấy" tại giao diện quản lý in ấn.		
Preconditions:	<ul style="list-style-type: none">- SPSO có tài khoản trên website- SPSO đã đăng nhập thành công vào hệ thống với quyền hạn SPSO- Thiết bị của SPSO có kết nối mạng và kết nối với hệ thống		
Postconditions:	Dữ liệu về ngày tặng giấy, số giấy được tặng đã được update trong database và trên giao diện website.		
Normal Flows:	<ol style="list-style-type: none">1. SPSO chọn "Cài đặt tặng giấy" tại giao diện quản lý in ấn, hệ thống hiển thị giao diện cài đặt tặng giấy2. SPSO chọn "Chỉnh sửa", Hệ thống hiển thị form điều3. SPSO thực hiện chỉnh sửa4. SPSO chọn Xác nhận5. Hệ thống xác nhận thay đổi, cập nhật lại database, quay lại giao diện cài đặt tặng giấy		
Alternative Flows:	Không		
Exceptions:	<p>E1: Tại bước 4</p> <p>4.1 SPSO chọn Apply, Hệ thống phát hiện thay đổi không hợp lệ.</p> <p>4.2 Hệ thống xóa tất cả những thay đổi vừa thực hiện, giao diện vẫn giữ nguyên ban đầu. Quay lại bước 2.</p>		
Note and issues:	Không có		

3.6 Phản hồi

3.6.1 Use-case diagram



Hình 7: Use-case diagram cho chức năng phản hồi và khiếu nại

3.6.2 Use-case scenario



Use-case Provide feedback

Use case name:	Provide feedback		
Created by:	Lê Đình Huy	Last updated by:	Lê Đình Huy
Date created:	25/09/2023	Date last updated:	29/09/2023
Actors:	Khách hàng		
Description:	Cho phép khách hàng cung cấp các phản hồi về hệ thống.		
Trigger:	Chọn nút "Add feedback" trong giao diện feedback.		
Preconditions:	<ul style="list-style-type: none">- Khách hàng có tài khoản trên website- Khách hàng đã đăng nhập thành công vào hệ thống- Thiết bị của khách hàng có kết nối mạng và kết nối với hệ thống- Khách hàng có quyền cung cấp feedback		
Postconditions:	Feedback được gửi lên hệ thống và hiển thị trong mục "Feedback". của khách hàng.		
Normal Flows:	<ol style="list-style-type: none">1. Khách hàng chọn "Feedback" tại thanh điều hướng.2. Hệ thống hiển thị giao diện Feedback.3. Khách hàng chọn "Add feedback".4. Hệ thống hiển thị biểu mẫu các thông tin của một feedback.5. Khách hàng điền biểu mẫu.6. Khách hàng xác nhận thao tác.7. Feedback được gửi lên hệ thống và hiển thị trong giao diện Feedback của chính khách hàng đó và SPSO.		
Alternative Flows:	Không có		
Exceptions:	<p>E1: Tại bước 6</p> <p>6.1 Khách hàng không xác nhận. 6.2 Hệ thống xóa tất cả những thay đổi khách hàng vừa thực hiện, giao diện vẫn giữ nguyên ban đầu. Use-case dừng lại.</p>		
Note and issues:	Không có		



Use-case Delete feedback

Use case name:	Delete feedback		
Created by:	Lê Đình Huy	Last updated by:	Lê Đình Huy
Date created:	25/09/2023	Date last updated:	15/10/2023
Actors:	Khách hàng		
Description:	Cho phép khách hàng xóa các phản hồi của mình về hệ thống.		
Trigger:	Chọn nút "Delete" cho feedback cần xóa.		
Preconditions:	<ul style="list-style-type: none">- Khách hàng có tài khoản trên website- Khách hàng đã đăng nhập thành công vào hệ thống- Thiết bị của khách hàng có kết nối mạng và kết nối với hệ thống- Khách hàng có quyền xóa feedback đó		
Postconditions:	Feedback được chọn và tất cả Reply trong nó bị xóa khỏi hệ thống.		
Normal Flows:	<ol style="list-style-type: none">1. Khách hàng chọn "Feedback" tại thanh điều hướng.2. Hệ thống hiển thị giao diện Feedback gồm danh sách các feedback.3. Khách hàng chọn "Delete" cho một feedback.4. Khách hàng xác nhận thao tác.5. Feedback và các reply trong nó được xóa khỏi hệ thống.		
Alternative Flows:	<p>A1: Tại bước 3</p> <ol style="list-style-type: none">3.1 Khách hàng chọn "View" một feedback.3.2 Hệ thống hiển thị nội dung của feedback.3.3 Khách hàng chọn "Delete". <p>Tiếp tục tại bước 4 trong Normal Flows.</p>		
Exceptions:	<p>E1: Tại bước 4</p> <ol style="list-style-type: none">4.1 Khách hàng không xác nhận.4.2 Hệ thống xóa tất cả những thay đổi khách hàng vừa thực hiện, giao diện vẫn giữ nguyên ban đầu. <p>Use-case dừng lại.</p>		
Note and issues:	Không có		



Use-case Reply

Use case name:	Reply		
Created by:	Lê Đình Huy	Last updated by:	Lê Đình Huy
Date created:	25/09/2023	Date last updated:	29/09/2023
Actors:	User (Customer và SPSO)		
Description:	Cho phép User trả lời các Feedback của Customer.		
Trigger:	Chọn nút "Reply" trong giao diện xem của feedback đó.		
Preconditions:	<ul style="list-style-type: none">- User có tài khoản trên website- User đã đăng nhập thành công vào hệ thống- Thiết bị của User có kết nối mạng và kết nối với hệ thống- User có quyền reply		
Postconditions:	Reply được gửi lên hệ thống và hiển thị trên website.		
Normal Flows:	<ol style="list-style-type: none">1. User chọn "Feedback" tại thanh điều hướng.2. Hệ thống hiển thị giao diện Feedback gồm danh sách các feedback.3. User chọn "View" một feedback.4. Hệ thống hiển thị giao diện của một feedback.5. User chọn "Reply".6. Hệ thống hiển thị biểu mẫu của Reply.7. User điền biểu mẫu.8. User xác nhận thao tác.9. Reply được gửi lên hệ thống và hiển thị trên website.		
Alternative Flows:	Không có		
Exceptions:	<p>E1: Tại bước 8</p> <p>8.1 User không xác nhận.</p> <p>8.2 Hệ thống xóa tất cả những thay đổi User vừa thực hiện, giao diện vẫn giữ nguyên ban đầu.</p> <p>Use-case dừng lại.</p>		
Note and issues:	Không có		

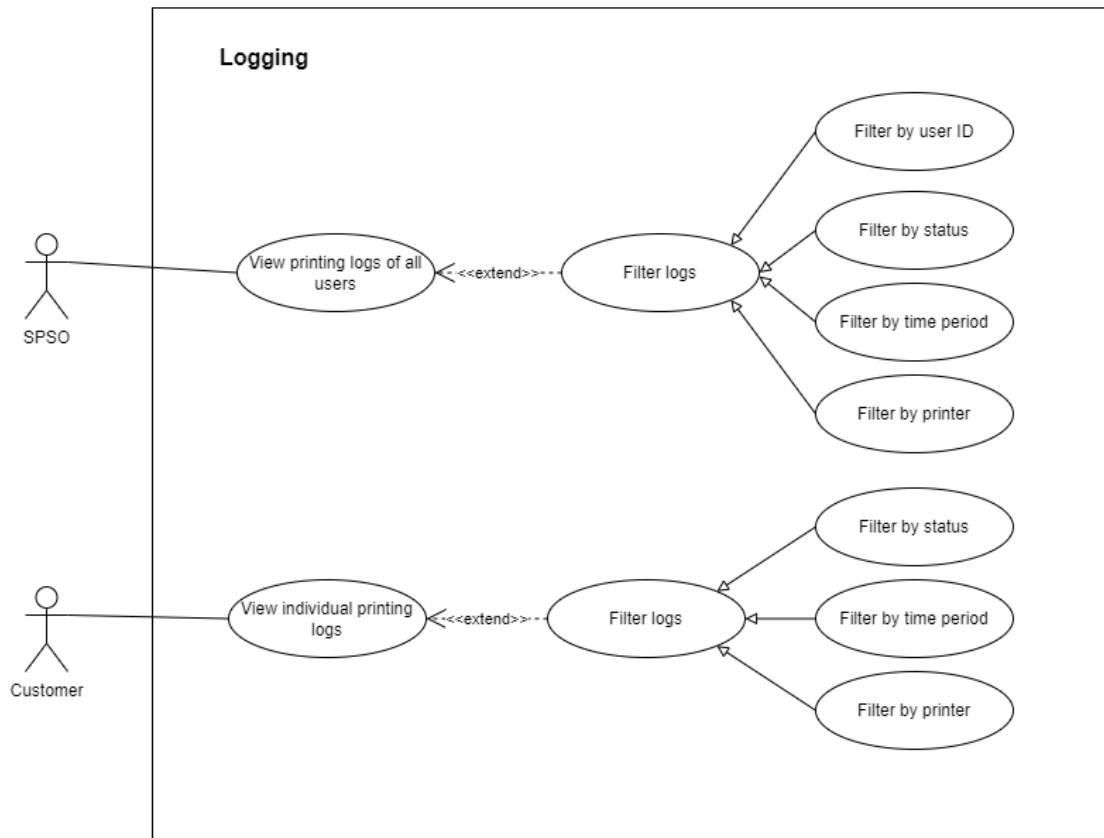


Use-case Delete reply

Use case name:	Delete reply		
Created by:	Lê Đình Huy	Last updated by:	Lê Đình Huy
Date created:	25/09/2023	Date last updated:	29/09/2023
Actors:	User (Customer và SPSO)		
Description:	Cho phép User xóa Reply của mình.		
Trigger:	Chọn nút "Delete" cho một reply của mình trong giao diện xem của feedback đó.		
Preconditions:	<ul style="list-style-type: none">- User có tài khoản trên website- User đã đăng nhập thành công vào hệ thống- Thiết bị của User có kết nối mạng và kết nối với hệ thống- User có quyền xóa reply đó		
Postconditions:	Reply được xóa khỏi hệ thống.		
Normal Flows:	<ol style="list-style-type: none">1. User chọn "Feedback" tại thanh điều hướng.2. Hệ thống hiển thị giao diện Feedback gồm danh sách các feedback.3. User chọn "View" một feedback.4. Hệ thống hiển thị giao diện của một feedback.5. User chọn "Delete" cho Reply cần xóa của mình.6. User xác nhận thao tác.7. Reply được xóa khỏi hệ thống.		
Alternative Flows:	Không có		
Exceptions:	<p>E1: Tại bước 6</p> <p>6.1 User không xác nhận.</p> <p>6.2 Hệ thống xóa tất cả những thay đổi User vừa thực hiện, giao diện vẫn giữ nguyên ban đầu.</p> <p>Use-case dừng lại.</p>		
Note and issues:	Không có		

3.7 Lịch sử in

3.7.1 Use case diagram



Hình 8: Use-case diagram cho chức năng truy cập lịch sử in

3.7.2 Use-case scenario



Use-case View printing logs of all users

Use case name:	View printing logs of all users		
Created by:	Phạm Đức Thắng	Last updated by:	Phạm Đức Thắng
Date created:	28/09/2023	Date last updated:	28/09/2023
Actors:	Student Printing Service Officer (SPSO)		
Description:	Use case này cho phép SPSO truy cập toàn bộ lịch sử in ấn của tất cả người dùng.		
Trigger:	SPSO click vào ô “View Printing History” trên thanh điều hướng.		
Preconditions:	SPSO phải đăng nhập vào hệ thống. Hệ thống được kết nối với cơ sở dữ liệu lưu trữ lịch sử người dùng.		
Postconditions:	SPSO đã truy cập thành công lịch sử in ấn của tất cả người dùng. Dữ liệu lịch sử in ấn không bị thay đổi.		
Normal Flows:	1. SPSO click vào ô “View Printing History” trên thanh điều hướng. 2. Hệ thống hiển thị mặc định thông tin lịch sử in của tất cả người dùng đã đăng ký cùng các bộ lọc (theo thời gian hoặc ID máy in), thanh tìm kiếm theo ID người dùng và sidebar chọn theo tình trạng tài liệu. Thông tin hiển thị bao gồm: ID người sử dụng, ID máy in, tên file, tình trạng in, số trang in. Các thông tin được phân trang với tối đa 4 thông tin mỗi trang.		
Alternative Flows:	A1: tại bước 2 2a: SPSO sử dụng bộ lọc, thanh tìm kiếm hoặc sidebar. 2b: Hệ thống chỉ hiển thị thông tin lịch sử in trong khoảng thời gian, của máy in được chọn, của người dùng có ID trên thanh tìm kiếm hoặc theo tình trạng của tài liệu.		
Exceptions:	E1: tại bước 2 2a: SPSO sử dụng bộ lọc thời gian không hợp lệ (ngày bắt đầu sau ngày kết thúc, ngày kết thúc sau ngày hiện tại). 2b: Hệ thống báo lỗi tương ứng và yêu cầu user nhập lại thông tin hợp lệ. E2: tại bước 2 2a: Nếu chưa có dữ liệu in hoặc không có dữ liệu in nào thỏa mãn điều kiện bộ lọc, hệ thống hiển thị thông báo không có dữ liệu in.		
Note and issues:			



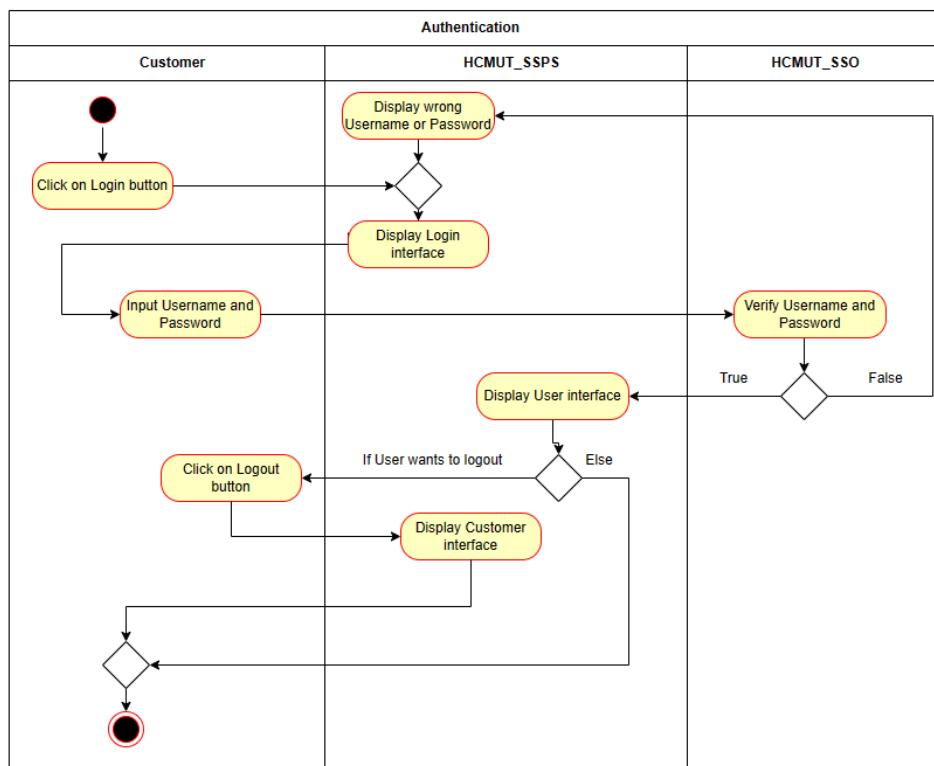
Use-case View individual printing logs

Use case name:	View individual printing logs		
Created by:	Phạm Đức Thắng	Last updated by:	Phạm Đức Thắng
Date created:	28/09/2023	Date last updated:	28/09/2023
Actors:	User		
Description:	Use case này cho phép người dùng truy cập lịch sử in ấn cá nhân.		
Trigger:	User click vào ô “View Printing History” trên thanh điều hướng.		
Preconditions:	User phải đăng nhập thành công vào hệ thống. Hệ thống được kết nối với cơ sở dữ liệu lưu trữ lịch sử người dùng.		
Postconditions:	User truy cập thành công lịch sử in ấn cá nhân. Dữ liệu lịch sử in ấn không bị thay đổi.		
Normal Flows:	1. User click vào ô “View Printing History” trên thanh điều hướng. 2. Hệ thống hiển thị trang lịch sử in cá nhân bao gồm thông tin lịch sử in ấn cá nhân và các bộ lọc (theo khoảng thời gian, ID máy in và tình trạng tài liệu). Mỗi thông tin lịch sử in được hiển thị bao gồm: ID máy in, tên file, tình trạng, số trang in. Các thông tin được phân trang với tối đa 4 thông tin mỗi trang.		
Alternative Flows:	A1: tại bước 2 2a: User sử dụng bộ lọc theo thời gian, theo ID máy in hoặc chọn tình trạng in theo sidebar. 2b: Hệ thống hiển thị thông tin lịch sử in thỏa mãn các điều kiện của bộ lọc.		
Exceptions:	E1: tại bước 2 2a: User sử dụng bộ lọc thời gian không hợp lệ (ngày bắt đầu sau ngày kết thúc, ngày kết thúc sau ngày hiện tại). 2b: Hệ thống báo lỗi tương ứng và yêu cầu user nhập lại thông tin hợp lệ. E2: tại bước 2 2a: Nếu chưa có dữ liệu lịch sử in hoặc không có dữ liệu lịch sử in nào thỏa mãn điều kiện bộ lọc, hệ thống hiển thị thông báo không có dữ liệu lịch sử in.		
Notes and Issues:			

II Mô hình hóa hệ thống

1 Activity diagrams

1.1 Xác thực

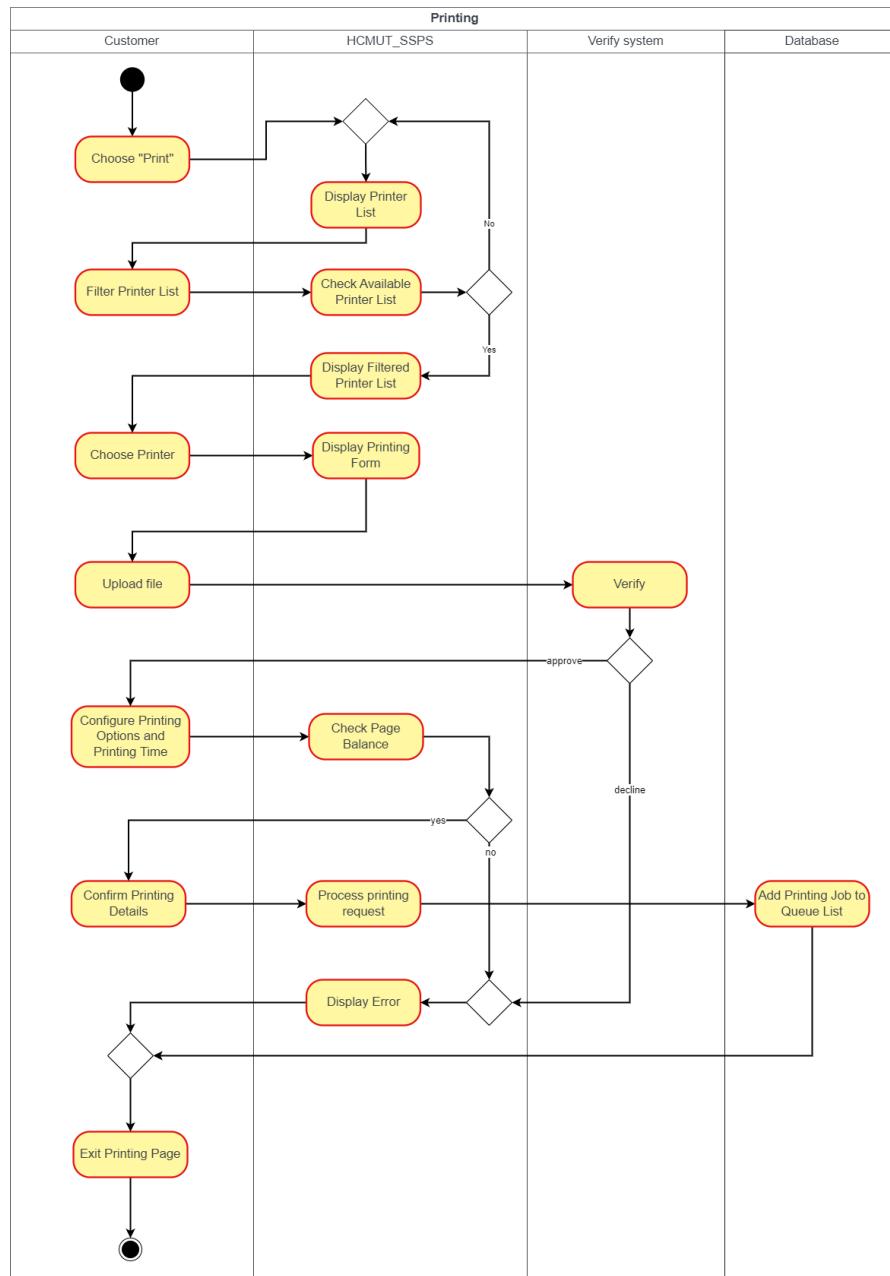


Hình 9: Activity diagram cho module xác thực tài khoản

Mô tả:

Dầu tiên, Customer chọn nút "Login", hệ thống sẽ hiển thị giao diện đăng nhập. Tại đó, Customer có thể nhập tài khoản và mật khẩu để có thể đăng nhập vào hệ thống. Sau đó hệ thống sẽ xác thực tài khoản và mật khẩu của Customer. Nếu tài khoản và mật khẩu đúng thì hệ thống sẽ hiển thị giao diện của User, nếu sai thì sẽ hiển thị tài khoản và mật khẩu sai và hiển thị lại giao diện đăng nhập. Bước tiếp theo nếu người dùng muốn đăng xuất thì chọn nút "Logout", hệ thống sẽ hiển thị giao diện của Customer.

1.2 In



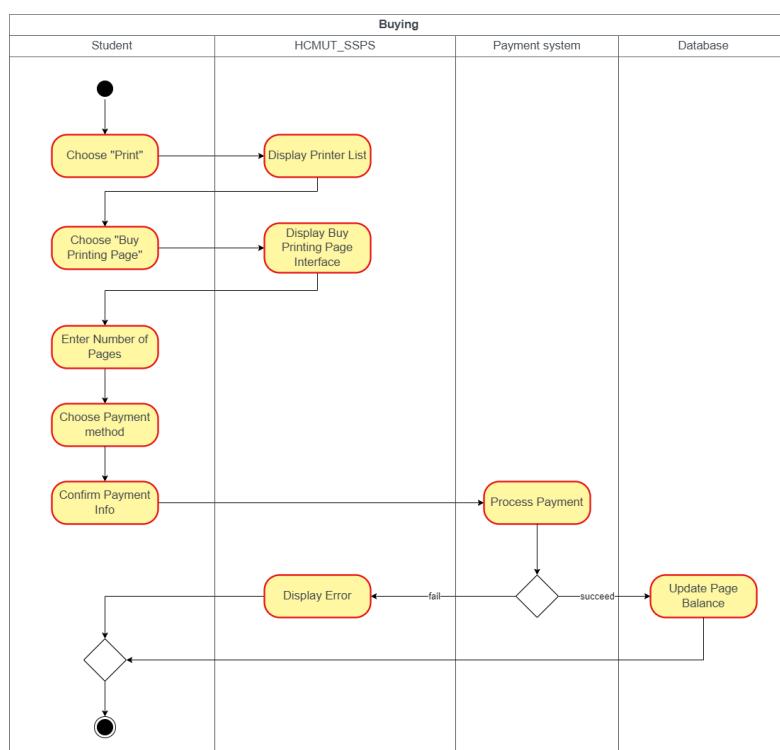
Hình 10: Activity diagram cho module in

Mô tả:

Dầu tiên, Customer chọn nút "In ngay" trên thanh điều hướng, hệ thống sẽ hiển thị danh sách các máy in. Sau đó Customer sẽ tiến hành chọn lọc máy in bằng cách lọc cơ sở, tòa, phòng.

Hệ thống sẽ kiểm tra xem những máy in thỏa mãn điều kiện lọc có đang khả thi (không bảo trì) hay không. Nếu không có máy in khả thi nào thỏa điều kiện lọc, hệ thống sẽ quay về hiển thị danh sách máy in, ngược lại hệ thống sẽ hiển thị danh sách máy in thỏa điều kiện lọc. Tại đây Customer sẽ tiến hành chọn máy in, sau đó hệ thống sẽ hiển thị form và Customer sẽ tiến hành upload file cần in lên hệ thống. Hệ thống xác thực nội dung sẽ kiểm tra file có hợp lệ hay không. Nếu file không hợp lệ, hệ thống sẽ báo lỗi về màn hình. Ngược lại, hệ thống xác thực nội dung thành công, Customer sẽ điền thông tin về số trang cần in, thời gian in, định dạng cách in. Và hệ thống sẽ kiểm tra xem số giấy Customer cần in có đủ hay không. Nếu không thì sẽ báo lỗi về Customer, và Customer cần phải giảm số trang giấy cần in hoặc mua thêm giấy. Ngược lại, Customer sẽ xác nhận thao tác in của mình và hệ thống sẽ xử lý, sau đó thêm thông tin in của Customer vào hàng chờ để xử lý.

1.3 Mua giấy

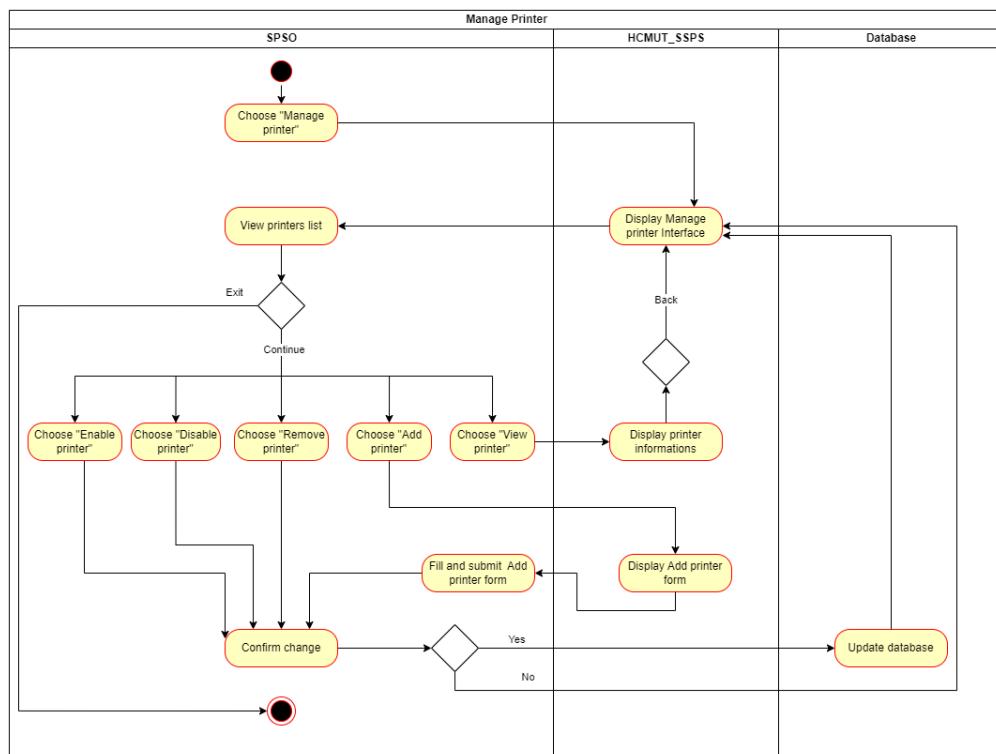


Hình 11: Activity diagram cho use-case mua giấy

Mô tả:

Tại giao diện "In ngay", sinh viên chọn vào "Mua giấy", hệ thống sẽ hiển thị giao diện mua giấy. Tiếp theo, sinh viên sẽ nhập số lượng giấy cần mua và chọn phương thức thanh toán. Sau đó hệ thống thanh toán sẽ xử lý giao dịch, nếu thất bại thì hệ thống sẽ báo lỗi về màn hình, ngược lại dữ liệu về lượng giấy của sinh viên sẽ được cập nhật.

1.4 Quản lý máy in

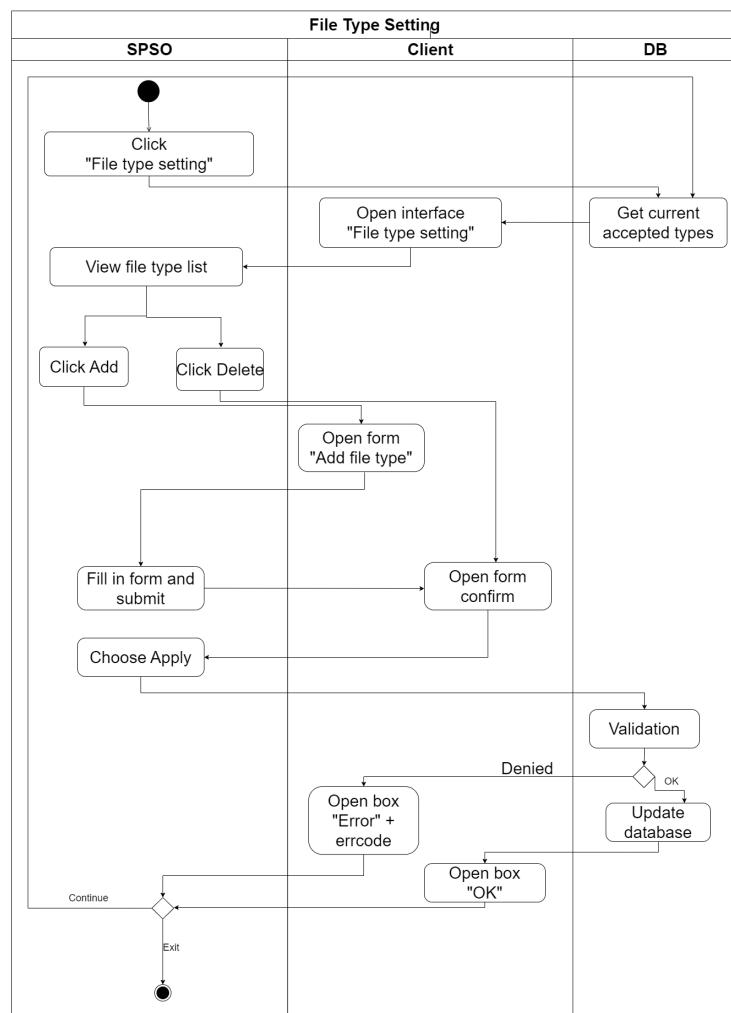


Hình 12: Activity diagram cho module quản lý máy in

Mô tả:

Dầu tiên, SPSO chọn "Quản lý máy in", hệ thống sẽ hiển thị giao diện quản lý máy in và SPSO sẽ được xem danh sách các máy in. Tại đó, SPSO có thể chọn "Enable printer" để cấp phép máy in, chọn "Disable printer" để vô hiệu hóa máy in, chọn "Remmove printer" để xóa máy in hoặc có thể thêm máy in bằng cách chọn "Add printer" rồi hệ thống sẽ hiển thị form thông tin của máy in mới và SPSO phải điền và nhấn lưu. Sau các thao tác trên, SPSO phải thực hiện xác nhận thay đổi, nếu đồng ý thì dữ liệu sẽ được cập nhật và hiển thị lại danh sách máy in hoặc không thì sẽ bỏ qua và hiển thị lại danh sách máy in. Ngoài ra, khi xem danh sách máy in, SPSO còn có thể ấn nút "View printer" để xem thông tin của một máy in. Khi đó, hệ thống sẽ hiển thị các thông tin của máy in đó, nếu SPSO nhất "BacK" thì sẽ quay lại giao diện hiển thị danh sách các máy in. Tất cả các tùy chọn đều trả lại giao diện hiển thị danh sách các máy in và người dùng có thể xem. Nếu muốn tiếp tục thực hiện các tùy chọn đó thì cứ tiếp tục, còn không sẽ kết thúc hoạt động.

1.5 Sửa danh mục định dạng tệp cho phép in



Hình 13: Activity diagram cho module Sửa định dạng tệp cho phép in

Mô tả:

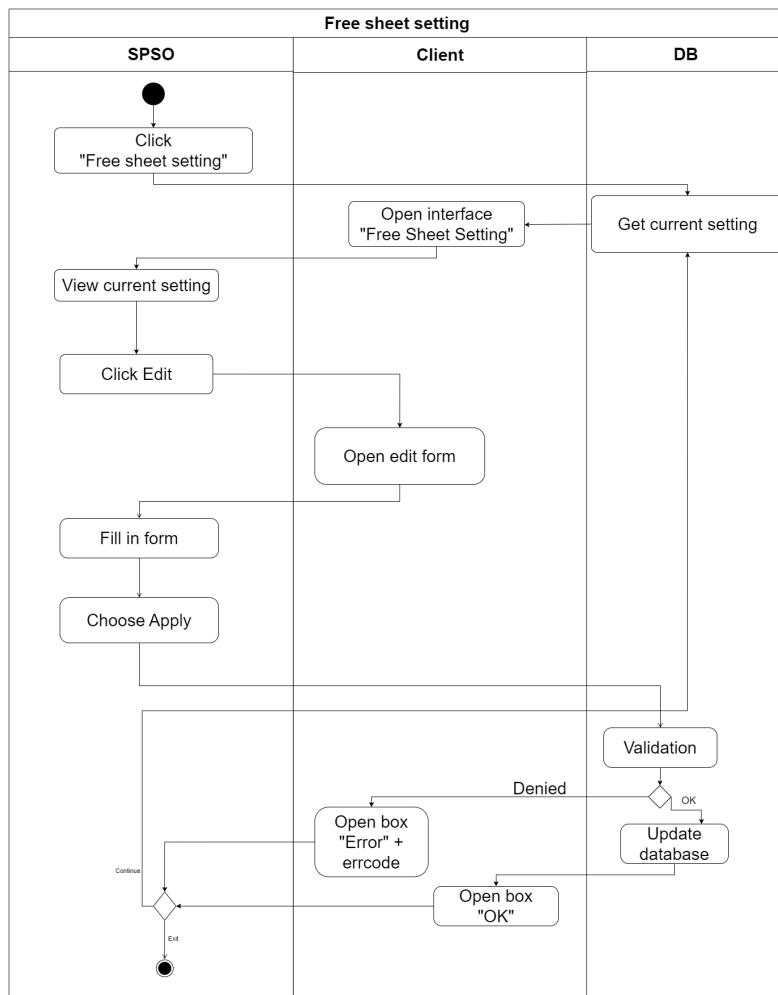
Bắt đầu, SPSO chọn "File type setting", sau đó cơ sở dữ liệu thực hiện tải những kiểu tệp được cho phép in hiện tại, và hiển thị chúng đồng thời với cửa sổ "File type setting" thông qua client.

Tiếp theo, SPSO xem danh sách kiểu tệp được in và quyết định thao tác chọn "Add" hoặc "Delete". Nếu chọn "Add", cửa sổ "Add file type" được hiển thị, người dùng tiếp tục nhập kiểu tệp muốn thay đổi và nhấn "Apply". Còn nếu chọn "Delete" thì chỉ cần nhấn "Apply".

Nếu cơ sở dữ liệu kiểm tra thay đổi hợp lệ thì cập nhật dữ liệu, sau đó client sẽ hiển thị "Ok". Nếu không hợp lệ thì không cập nhật và client báo lỗi.

Để kết thúc, người dùng thao tác chọn "Exit" hoặc có thể tiếp tục ở giao diện chỉnh sửa loại tệp cho phép in.

1.6 Quản lý tặng giấy miễn phí



Hình 14: Activity diagram cho module Quản lý tặng giấy miễn phí

Mô tả:

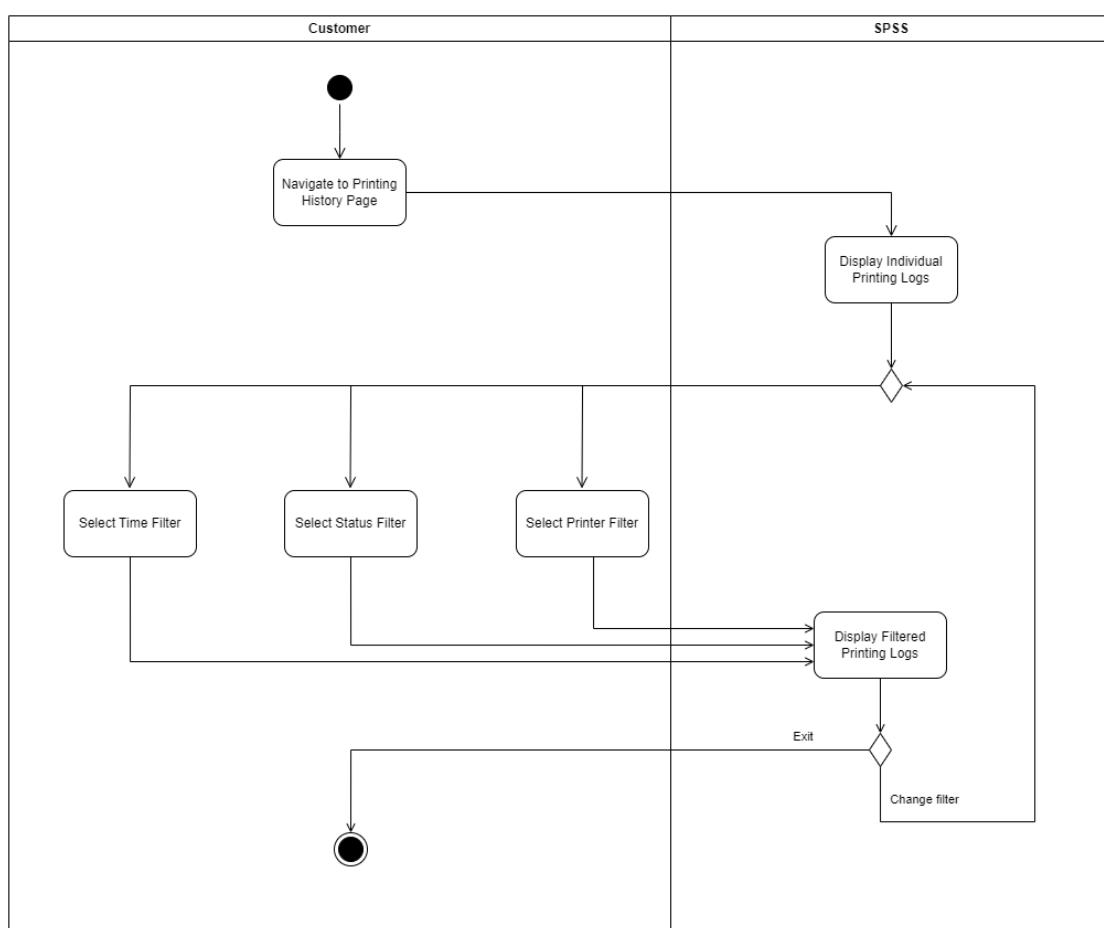
Bắt đầu, SPSO chọn "Free sheet setting", sau đó cơ sở dữ liệu thực hiện tải trạng thái ngày phát giấy miễn phí, số lượng của giấy miễn phí hiện tại, và hiển thị chúng đồng thời với cửa sổ "Free sheet setting" thông qua client.

Tiếp theo, SPSO xem và chọn "Edit". Khi đó cửa sổ "Edit" được hiển thị, người dùng thực hiện thay đổi (Ngày hoặc số tờ giấy) và nhấn "Apply".

Nếu cơ sở dữ liệu kiểm tra thay đổi hợp lệ thì cập nhật dữ liệu, sau đó client sẽ hiển thị "Ok". Nếu không hợp lệ thì không cập nhật và client báo lỗi.

Để kết thúc, người dùng thao tác chọn "Exit" hoặc có thể tiếp tục ở giao diện cài đặt tặng giấy.

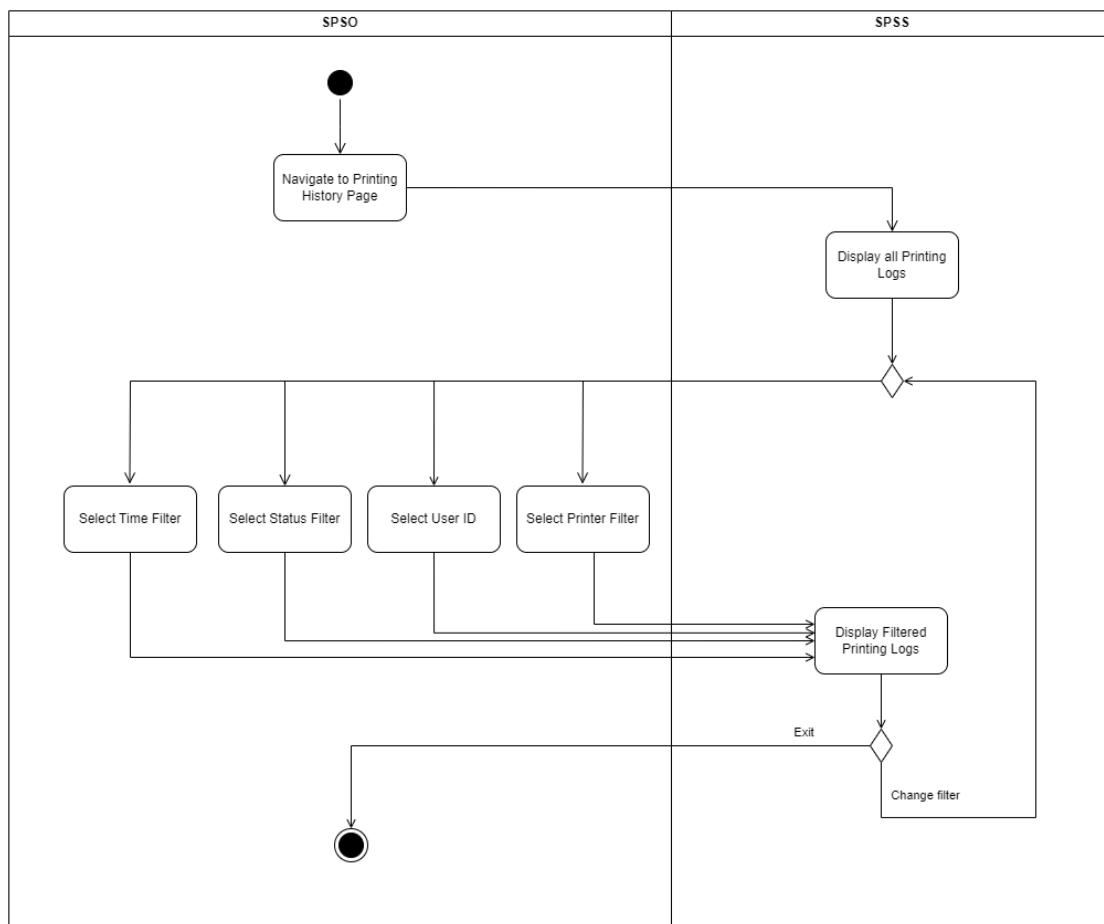
1.7 Lịch sử in



Hình 15: Activity diagram cho use case truy cập lịch sử in của Khách hàng

Mô tả:

Đầu tiên, khách hàng điều hướng đến trang Lịch sử in. Hệ thống sẽ hiển thị tất cả lịch sử in ấn của khách hàng đó. Khách hàng có thể chọn bộ lọc để lọc bớt dữ liệu. Cuối cùng, người dùng xem các thông tin cần thiết và thoát trang Lịch sử in.



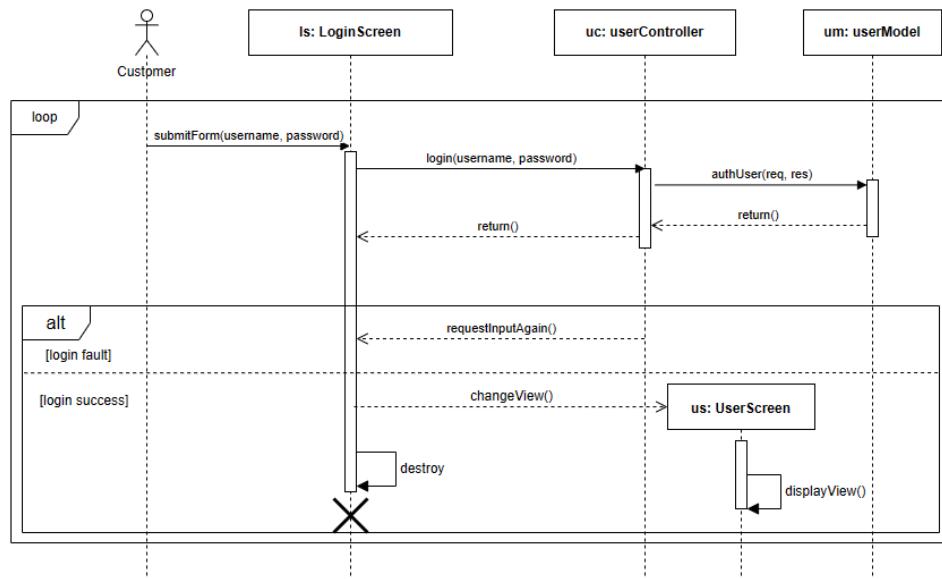
Hình 16: Activity diagram cho use case truy cập lịch sử in của SPSO

Mô tả:

Dầu tiên, SPSO điều hướng đến trang Lịch sử in. Hệ thống sẽ hiển thị tất cả lịch sử in của tất cả khách hàng. Khách hàng có thể chọn bộ lọc để lọc bớt dữ liệu. Cuối cùng, SPSO xem các thông tin cần thiết và thoát trang Lịch sử in.

2 Sequence diagrams

2.1 Xác thực

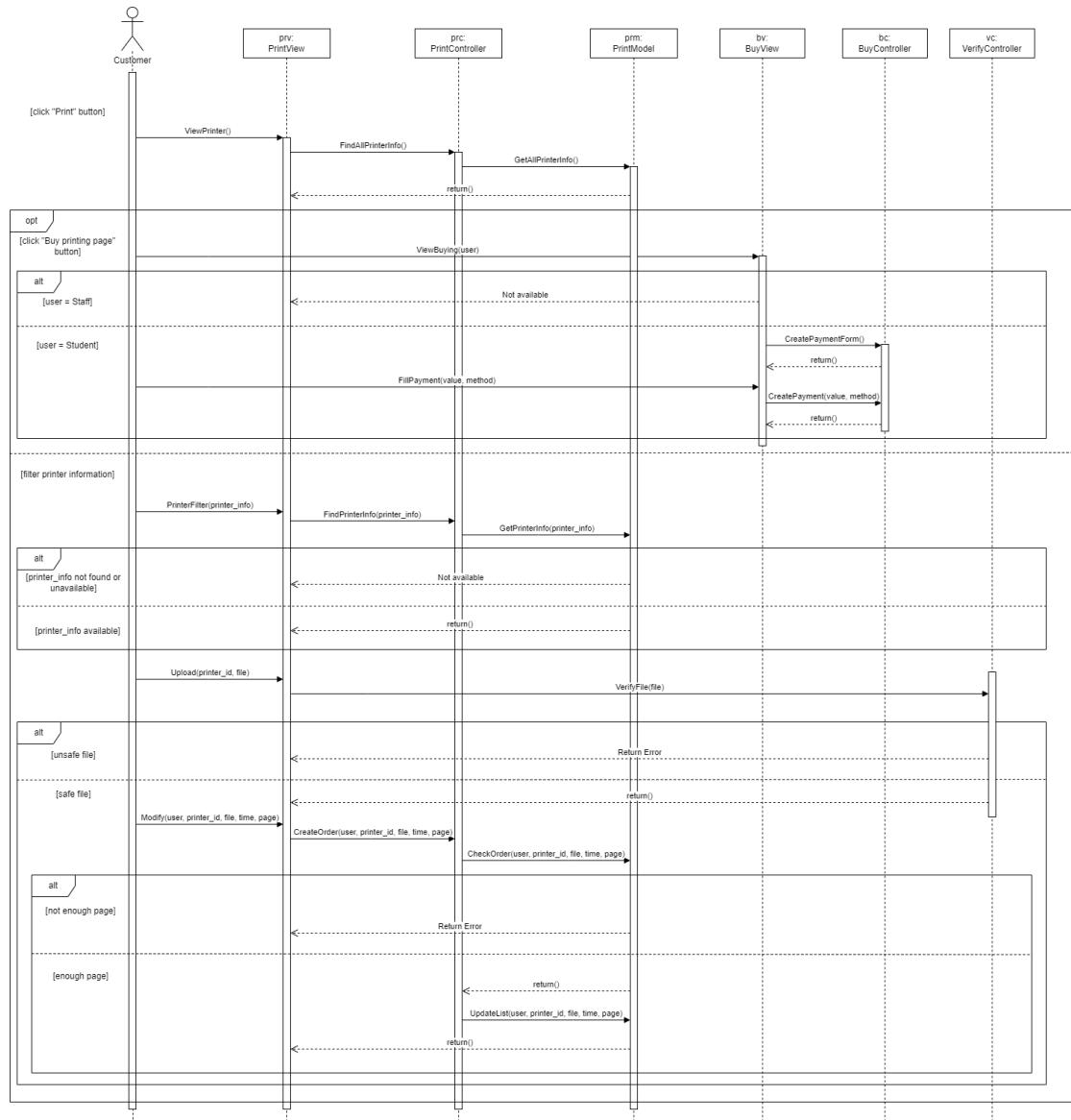


Hình 17: Sequence diagram cho module xác thực tài khoản

Mô tả:

Dầu tiên, Customer sẽ nhập username vs password thông qua hàm `submitForm(username, password)`. uc sẽ gọi hàm `login(username, password)` để tiếp nhận giá trị của username và password. uc sẽ yêu cầu um gọi hàm `authUser(req, res)` để xác thực username và password và trả giá trị về ls. Nếu giá trị sai sẽ yêu cầu nhập lại username và password, nếu giá trị đúng sẽ trả về giao diện của user.

2.2 In



Hình 18: Sequence diagram cho module in

Mô tả:

Dầu tiên, Customer sẽ mở **prv** qua hàm `ViewPrinter()`. Sau đó **prc** gọi hàm `FindAllPrinterInfo()` để lấy tất cả máy in hiển thị trên **prv**.

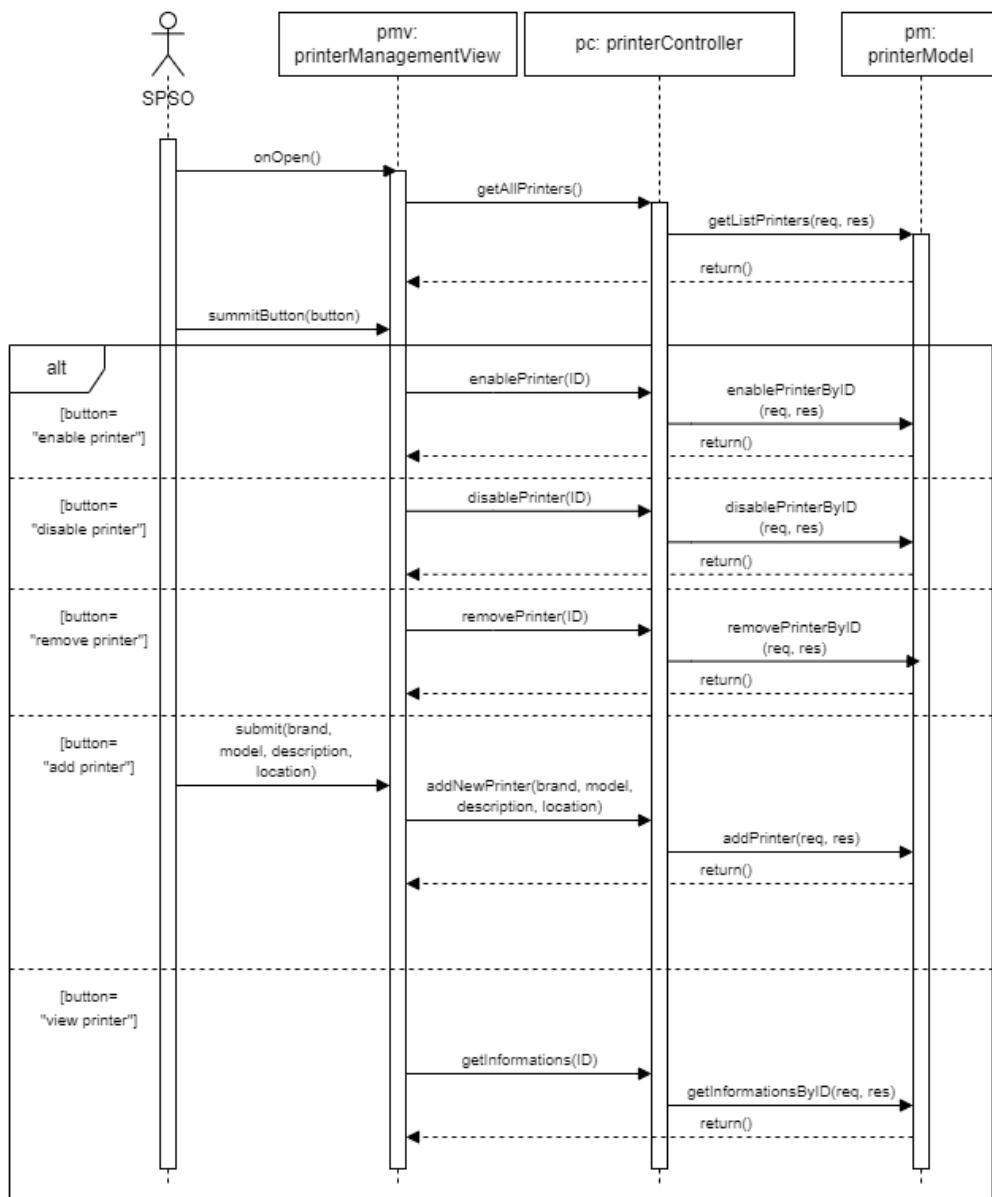
prc yêu cầu **prm** gọi hàm `GetAllPrinterInfo()` để lấy danh sách máy in từ database, sau đó danh sách máy in sẽ được trả về và hiển thị ở **prv**. Tiếp theo Customer có 2 luồng lựa chọn song song:

- Customer click vào "Mua thêm giấy", mở **bv** qua hàm `ViewBuying(user)`.



- Nếu như user truyền vào là Staff thì sẽ báo lỗi về **prv** và hiển thị lên màn hình.
 - Nếu user truyền vào là Student, **bv** yêu cầu **bc** gọi hàm *CreatePaymentForm()* để tạo form thông tin thanh toán, sau đó trả về màn hình. Tiếp theo Customer cung cấp thông tin thanh toán bao gồm số lượng giấy và phương thức thanh toán và gọi hàm *FillPayment(value, method)* từ **bv**, sau đó **bv** yêu cầu **bc** gọi hàm *CreatePayment(value, method)*, xử lý với hệ thống thanh toán và trả về màn hình kết quả giao dịch.
2. Customer tiến hành lọc thông tin của máy in, **prv** sẽ gọi hàm *PrinterFilter(printer_info)* dựa trên thông tin vừa được lọc, sau đó **prc** gọi hàm *FindPrinterInfo(printer_info)* để tìm những máy in phù hợp hiển thị lên màn hình. Tiếp theo **prc** yêu cầu **prm** gọi hàm *GetPrinterInfo(printer_info)* để lấy danh sách máy in trong database phù hợp với thông tin được cung cấp.
- Nếu không tìm được máy in phù hợp với thông tin lọc hoặc nếu có mà (các) máy in đó đang được bảo trì thì sẽ báo lỗi về **prv** và hiển thị lên màn hình.
 - Ngược lại, trả kết quả về **prv** và hiển thị danh sách các máy in phù hợp lên màn hình.
- Customer chọn vào máy muốn in và upload tài liệu. Sau đó **prv** sẽ gọi hàm *Upload(printer_id, file)* và yêu cầu **vc** gọi hàm *VerifyFile(file)* để xác minh nội dung của file vừa được upload.
- Nếu file vừa up có nội dung không an toàn, **vc** sẽ báo lỗi về **prv** và hiển thị lên màn hình.
 - Ngược lại, **vc** trả kết quả xác minh thành công và hiển thị lên màn hình. Sau đó Customer sẽ định dạng và cung cấp các thông tin in (thời gian, số lượng trang cần in), **prv** sẽ gọi hàm *Modify(user, printer_id, file, time, page)*, sau đó **prc** sẽ tạo order bằng hàm *CreateOrder(user, printer_id, file, time, page)* và yêu cầu **prm** gọi hàm *CheckOrder(user, printer_id, file, time, page)* để truy xuất vào database kiểm tra thông tin in của Customer.
 - Nếu số lượng giấy của Customer không đủ, **prm** sẽ báo lỗi về **prv** và hiển thị lên màn hình.
 - Ngược lại, **prm** trả kết quả về **prc**, sau đó **prc** sẽ tiếp tục yêu cầu **prm** gọi hàm *UpdateList(user, printer_id, file, time, page)* để cập nhật thông tin in vào database. Cuối cùng **prm** trả kết quả về **prv** và hiển thị lên màn hình.

2.3 Quản lý máy in



Hình 19: Sequence diagram cho module quản lý máy in

Mô tả:

- Đầu tiên, pmv sẽ được mở bởi SPSO qua hàm onOpen().
- pc gọi hàm getAllPrinters() để lấy tất cả máy in để hiển thị trên pmv.
- pc yêu cầu pm gọi hàm getListPrinters() để lấy danh sách các máy in từ database.
- Danh sách các máy in sẽ được return() và hiển thị ở pmv.



Tại bước 5: có 5 lựa chọn

A1:

5. SPSO chọn nút "enable printer" nên pmv gọi hàm submit("enable printer").
6. pc gọi hàm enablePrinter(ID) ứng với ID của máy in được chọn để kích hoạt máy in.
7. pm gọi hàm enablePrinterByID(req, res) từ yêu cầu của pc để set giá trị state của máy in thành 1.
8. Dữ liệu được cập nhật, return() và hiển thị ở pmv.

A2:

5. SPSO chọn nút "disable printer" nên pmv gọi hàm submit("disable printer").
6. pc gọi hàm disablePrinter(ID) ứng với ID của máy in được chọn để vô hiệu hóa máy in.
7. pm gọi hàm disablePrinterByID(req, res) từ yêu cầu của pc để set giá trị state của máy in thành 0.
8. Dữ liệu được cập nhật, return() và hiển thị ở pmv.

A3:

5. SPSO chọn nút "remove printer" nên pmv gọi hàm submit("remove printer").
6. pc gọi hàm removePrinter(ID) ứng với ID của máy in được chọn để xóa máy in.
7. pm gọi hàm removePrinterByID(req, res) từ yêu cầu của pmc để xóa máy in trong database.
8. Dữ liệu được cập nhật, return() và hiển thị ở pmv.

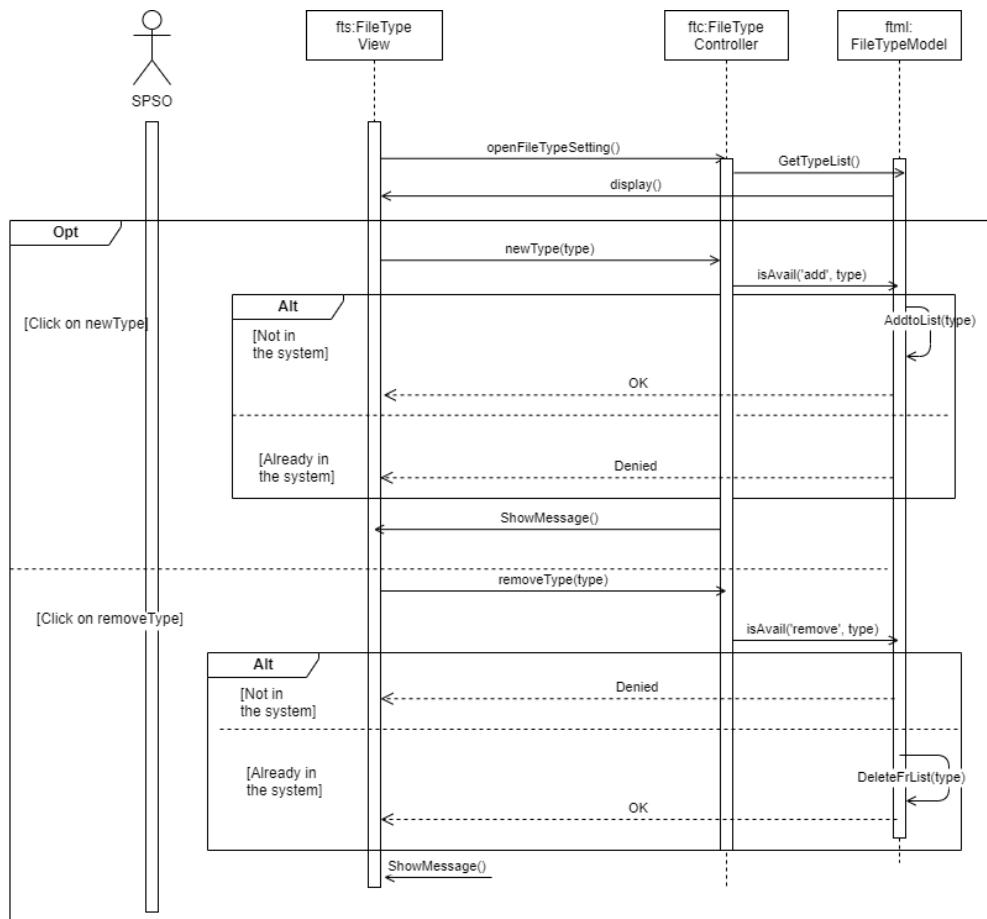
A4:

5. SPSO chọn nút "add printer" nên pmv gọi hàm submit("add printer").
6. SPSO nhập các thông tin của máy in và pmv gọi hàm submit(brand, model, description, location).
7. pc gọi hàm addNewPrinter(brand, model, description, location) để thêm máy in mới với các thuộc tính được SPSO nhập.
8. pm gọi hàm addPrinter(req, res) từ yêu cầu của pmc để thêm máy in trong database.
9. Dữ liệu được cập nhật, return() và hiển thị ở pmv.

A5:

5. SPSO chọn nút "view printer" nên pmv gọi hàm submit("remove printer").
6. pc gọi hàm getInformations(ID) ứng với ID của máy in được chọn để lấy tất cả thông tin của máy in.
7. pm gọi hàm getInformationsByID(req, res) từ yêu cầu của pmc để lấy tất cả thông tin của máy in từ database.
8. Dữ liệu được lấy, return() và hiển thị ở pmv.

2.4 Sửa danh mục định dạng tệp cho phép in



Hình 20: Sequence diagram cho module Sửa định dạng tệp cho phép in

Mô tả:

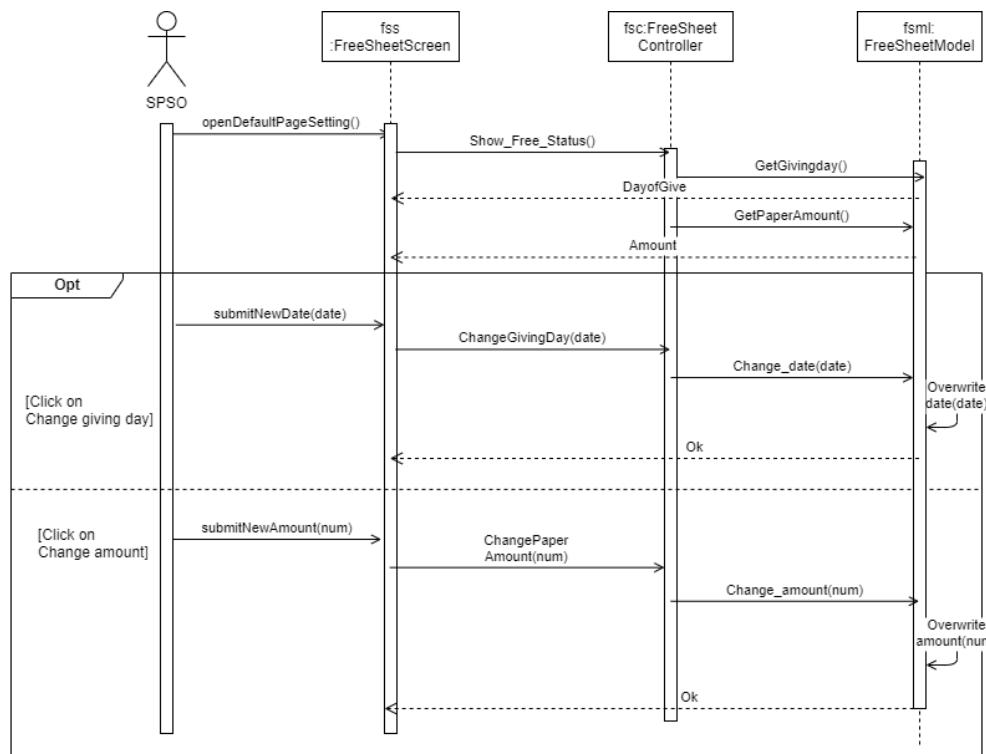
Khởi đầu flow, SPSO mở FileTypeSetting ở thanh công cụ, hiện thị trong mục Printing management. Hệ thống khởi động màn hình FileTypeView(viết tắt: fts). fts gọi tới bộ phận FileTypeController(viết tắt: fsc) được kích hoạt để gọi lệnh GetTypeList() tới FileTypeModel(ftml), từ đó nhận từ cơ sở dữ liệu danh sách các kiểu file trả về fts và hiển thị cho fts.

Tiếp theo, SPSO có 2 luồng lựa chọn song song:

- SPSO chọn thêm kiểu file mới với newType(type) trên màn hình fts. ftc sau đó chạy lệnh is_Avail('add',type) gửi tới ftm, nhằm kiểm tra hệ thống đã có sẵn hoặc chưa có kiểu file mới, nếu ftm xác nhận chưa có kiểu file đó, thực hiện thêm vào cơ sở dữ liệu FileTypeModel(ftml) với AddtoList(), trả về kết quả OK cho ftm và hiển thị thông báo lên màn hình. Ngược lại, nếu kiểu file đã tồn tại trong hệ thống thì trả về denied cho ftm, hiển thị thông báo truy vấn bị từ chối.
- SPSO chọn xóa kiểu file đang có với removeType(type) trên màn hình. ftm sau đó chạy

lệnh delete(type) gửi tới ftm, nhằm kiểm tra hệ thống đã có sẵn hoặc chưa có kiểu file mới, ftm gọi ftc kiểm tra với truy vấn isAvail(type), nếu hệ thống đã có kiểu file đó, thực hiện xóa khỏi cơ sở dữ liệu ftml với DeletefrList(type), trả về kết quả OK cho ftm và hiển thị thông báo lên màn hình. Ngược lại, nếu kiểu file chưa tồn tại trong hệ thống thì trả về denied cho ftm, hiển thị thông báo truy vấn bị từ chối.

2.5 Quản lý tặng giấy miễn phí



Hình 21: Sequence diagram cho module Quản lý tặng giấy miễn phí

Mô tả:

Để khởi đầu flow, SPSO mở DefaultPageSetting ở thanh công cụ, xuất hiện trong mục Printing management. Hệ thống khởi động màn hình FreeSheetScreen(viết tắt: fss), từ đây gửi lệnh GetFreeSheetStatus() để lấy thông tin hiện tại về quà tặng giấy miễn phí. Bộ phận FreeSheet-Manager(viết tắt: fsm) được kích hoạt để gọi 2 lệnh GetGivingDay, GetPaperAmount truy vấn từ FreeSheetController(viết tắt fsc), mục đích là lấy từ cơ sở dữ liệu DayofGive, Amount tương ứng là ngày tặng giấy và số lượng giấy đã được lưu trả về fsm. Sau đó fsm gửi lệnh hiển thị cho fss.

Tiếp theo, SPSO có 2 luồng lựa chọn song song:

- Luồng submitNewDate(date) trên fss để thay đổi ngày tháng tặng giấy miễn phí, tham số date là ngày mới được chọn, lệnh được chuyển tiếp tới fsm với ChangeGivingDay(date), chuyển tiếp tới Change_date cho fsc. Khi fsc đã nhận được lệnh, thực hiện lệnh Overwrite-



date(date) để thay đổi cơ sở dữ liệu FreeSheetModel. Để kết thúc, fsc gửi tín hiệu OK cho fsm và hiển thị thông tin xác nhận lên fss.

- Luồng submitNewAmount(num) trên fss để thay đổi số lượng giấy miễn phí, tham số num là số lượng giấy mới, lệnh được chuyển tiếp tới fsm với ChangePaperAmount(num), chuyển tiếp tới Change_amount cho fsc. Khi fsc đã nhận được lệnh, thực hiện lệnh Overwriteamount(num) để thay đổi cơ sở dữ liệu FreeSheetModel. Để kết thúc, fsc gửi tín hiệu OK cho fsm và hiển thị thông tin xác nhận lên fss.

2.6 Lịch sử in

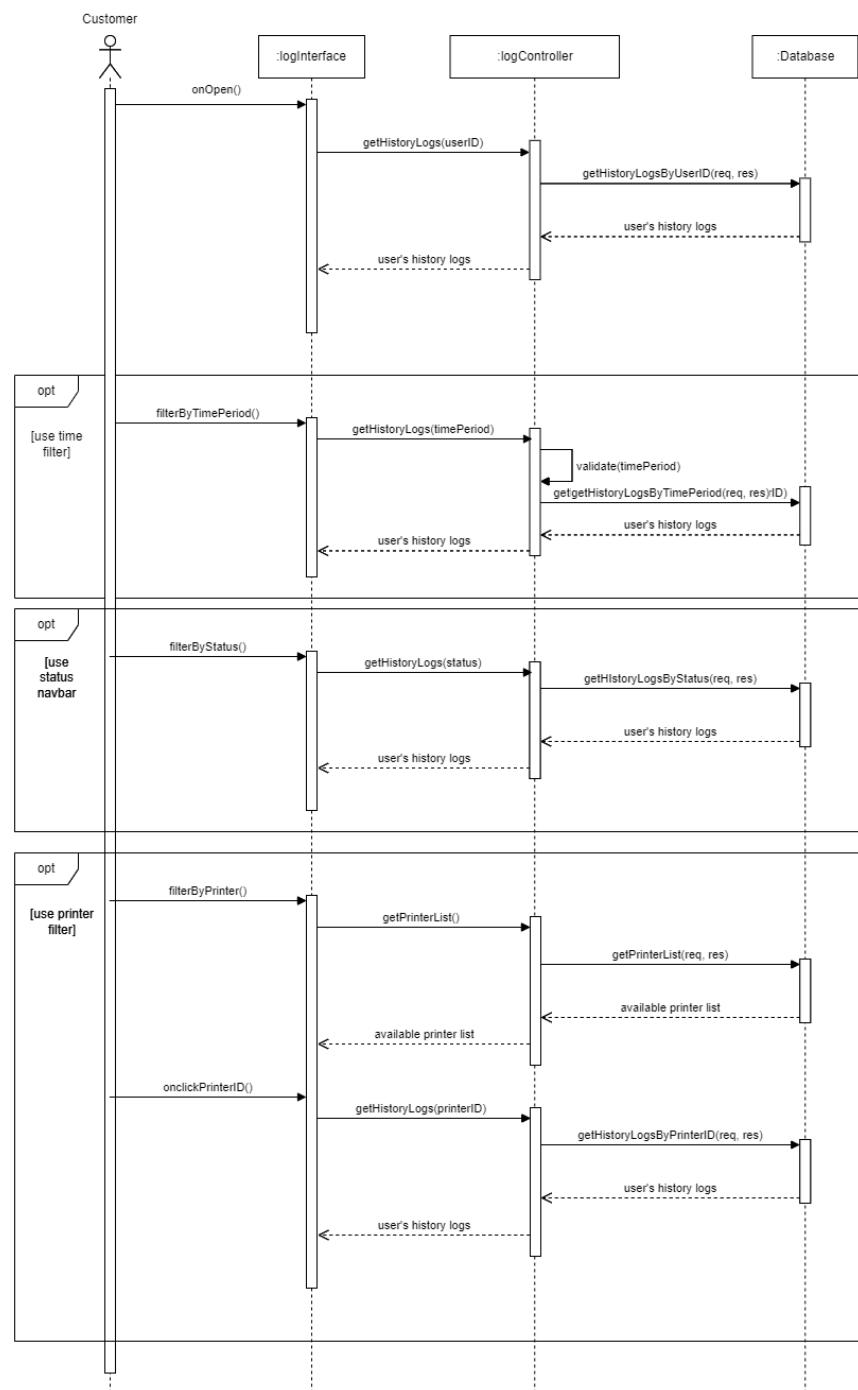
Mô tả cho sequence diagram cho use case truy cập lịch sử in của Khách hàng:

Dầu tiên, khách hàng điều hướng tới trang Lịch sử In, giao diện lịch sử in gọi hàm onOpen(). Rồi giao diện Lịch sử In sẽ gọi hàm getHistoryLogs() của logController với tham số là ID khách hàng. logController sẽ truy xuất dữ liệu lịch sử in từ cơ sở dữ liệu. Sau khi thu được dữ liệu lịch sử in, logController sẽ trả về dữ liệu cho giao diện để hiển thị cho khách hàng.

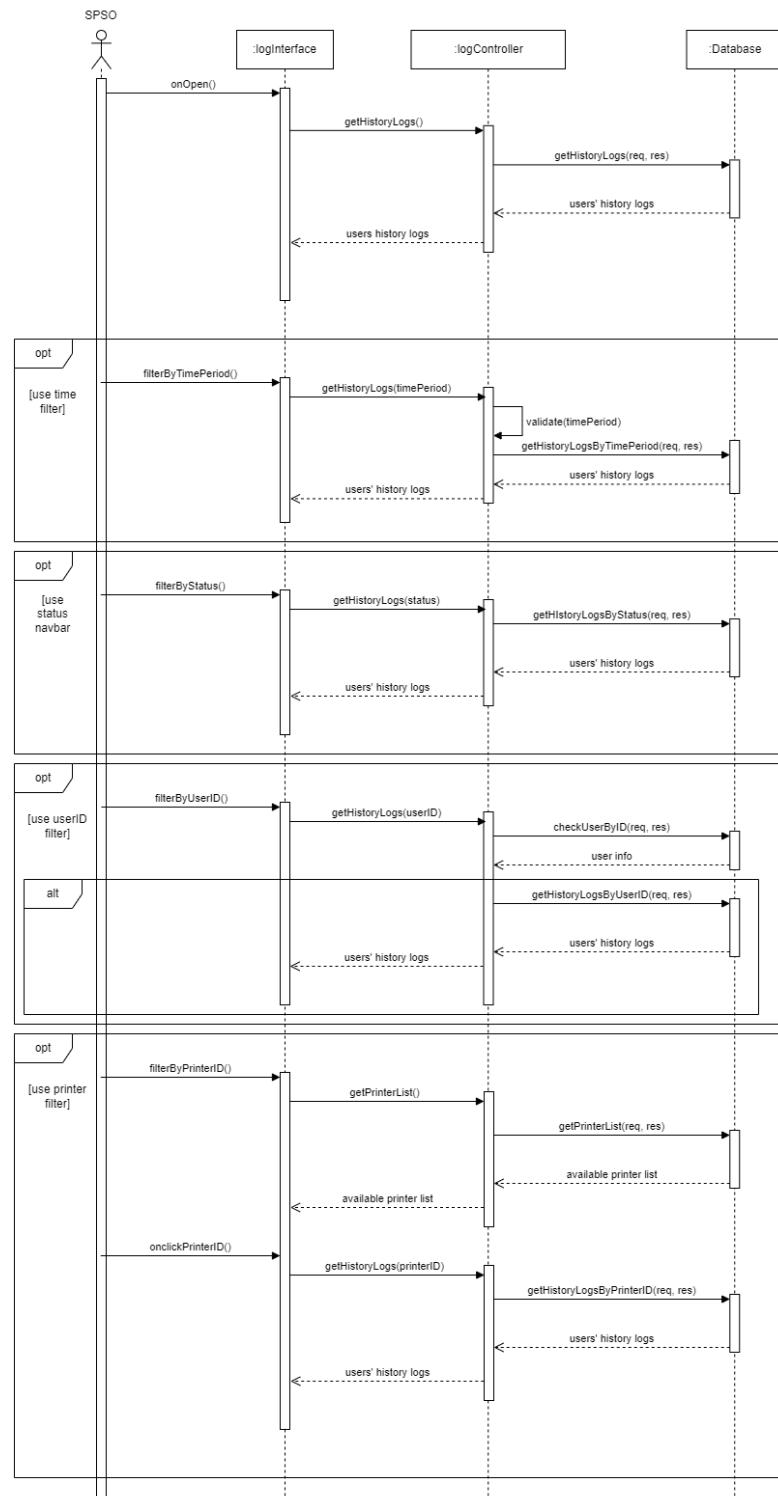
Nếu khách hàng sử dụng bộ lọc thời gian, giao diện sẽ truyền thêm tham số timePeriod vào hàm getHistoryLogs(). Trước hết, logController sẽ kiểm tra tính hợp lệ của thời gian người dùng nhập vào, nếu không hợp lệ sẽ báo lỗi. Nếu thời gian hợp lệ, logController sẽ lấy dữ liệu lịch sử in từ cơ sở dữ liệu và lọc lại dữ liệu trong khoảng thời gian khách hàng truyền vào. logController sẽ trả về kết quả cho giao diện hiển thị.

Nếu khách hàng sử dụng navbar lọc theo status, logInterface sẽ yêu cầu logController gọi hàm getHistoryLogs(status). Rồi logController sẽ truy cập vào cơ sở dữ liệu để lấy dữ liệu lịch sử in theo status và trả kết quả cho giao diện hiển thị.

Nếu khách hàng sử dụng bộ lọc theo máy in, logInterface sẽ yêu cầu danh sách các máy in từ logController. Sau đó, logController sẽ truy xuất danh sách các máy in đang hoạt động từ cơ sở dữ liệu và trả về cho giao diện. Khách hàng sẽ chọn các máy in được hiển thị và giao diện sẽ gọi hàm getHistoryLogs() với ID máy in được chọn. logController truy xuất dữ liệu lịch sử in thỏa các bộ lọc từ cơ sở dữ liệu và trả về cho giao diện hiển thị.



Hình 22: Sequence diagram cho use case truy cập lịch sử in của Khách hàng



Hình 23: Sequence diagram cho use case truy cập lịch sử in của SPSO



Mô tả:

Dầu tiên, SPSO điều hướng tới trang Lịch sử In, giao diện lịch sử in gọi hàm onOpen(). Rồi giao diện Lịch sử In sẽ gọi hàm getHistoryLogs() của logController. logController sẽ truy xuất dữ liệu lịch sử in từ cơ sở dữ liệu. Sau khi thu được dữ liệu lịch sử in, logController sẽ trả về dữ liệu cho giao diện để hiển thị cho SPSO.

Nếu SPSO sử dụng bộ lọc thời gian, giao diện sẽ truyền thêm tham số timePeriod vào hàm getHistoryLogs(). Trước hết, logController sẽ kiểm tra tính hợp lệ của thời gian người dùng nhập vào, nếu không hợp lệ sẽ báo lỗi. Nếu thời gian hợp lệ, logController sẽ lấy dữ liệu lịch sử in từ cơ sở dữ liệu và lọc lại dữ liệu trong khoảng thời gian SPSO truyền vào. logController sẽ trả về kết quả cho giao diện hiển thị.

Nếu SPSO sử dụng navbar lọc theo status, logInterface sẽ yêu cầu logController gọi hàm getHistoryLogs(status). Rồi logController sẽ truy cập vào cơ sở dữ liệu để lấy dữ liệu lịch sử in theo status và trả kết quả cho giao diện hiển thị.

Nếu SPSO sử dụng bộ lọc theo userID, logInterface sẽ gọi hàm getHistoryLogs() của logController với tham số userID. Trước hết, logController sẽ kiểm tra xem người dùng có tồn tại trong userDatabase hay không. Nếu có, logController sẽ truy xuất dữ liệu từ cơ sở dữ liệu và trả về dữ liệu cho giao diện hiển thị.

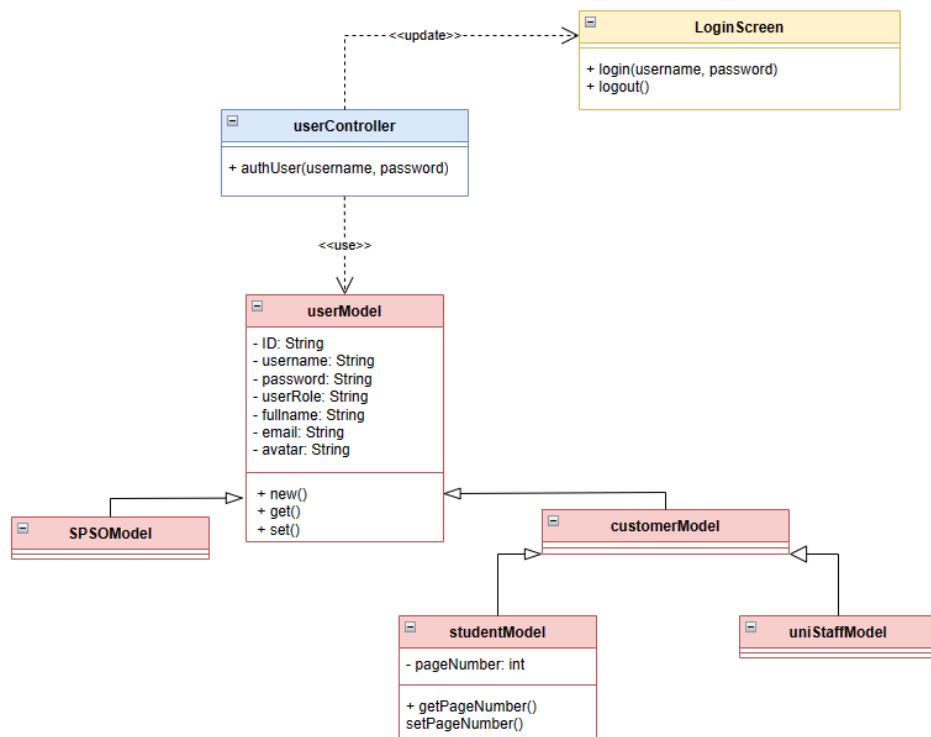
Nếu SPSO sử dụng bộ lọc theo máy in, logInterface sẽ yêu cầu danh sách các máy in từ logController. Sau đó, logController sẽ truy xuất danh sách các máy in đang hoạt động từ cơ sở dữ liệu và trả về cho giao diện. SPSO sẽ chọn các máy in được hiển thị và giao diện sẽ gọi hàm getHistoryLogs() với ID máy in được chọn. logController truy xuất dữ liệu lịch sử in thỏa các bộ lọc từ cơ sở dữ liệu và trả về cho giao diện hiển thị.

3 Class diagrams

Các class diagram được nhóm thiết kế theo mô hình kiến trúc MVC (Model-Controller-View). Mỗi tầng đảm nhiệm mỗi nhóm chức năng trong hệ thống:

- Tầng View: lắng nghe sự kiện, thu nhận dữ liệu và tương tác người dùng để chuyển đến Controller xử lý; hiển thị dữ liệu do Controller chuyển giao từ Model lên, hiển thị các giao diện, biểu mẫu xác nhận, thông báo, nhập liệu.
- Tầng Controller: xử lý sự kiện, điều khiển luồng dữ liệu, cập nhật giao diện.
- Tầng Model: lưu trữ dữ liệu hệ thống và cung cấp phương thức cơ bản xử lý logic nghiệp vụ.

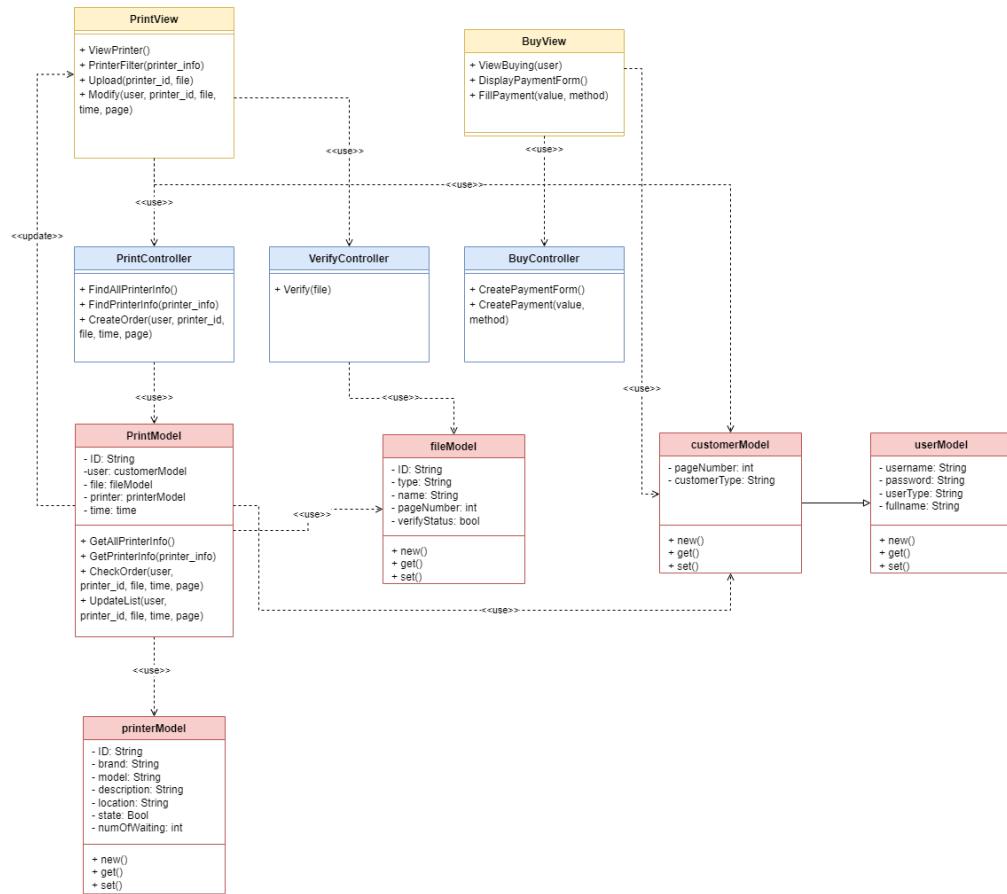
3.1 Xác thực



Hình 24: Class diagram cho module xác thực tài khoản



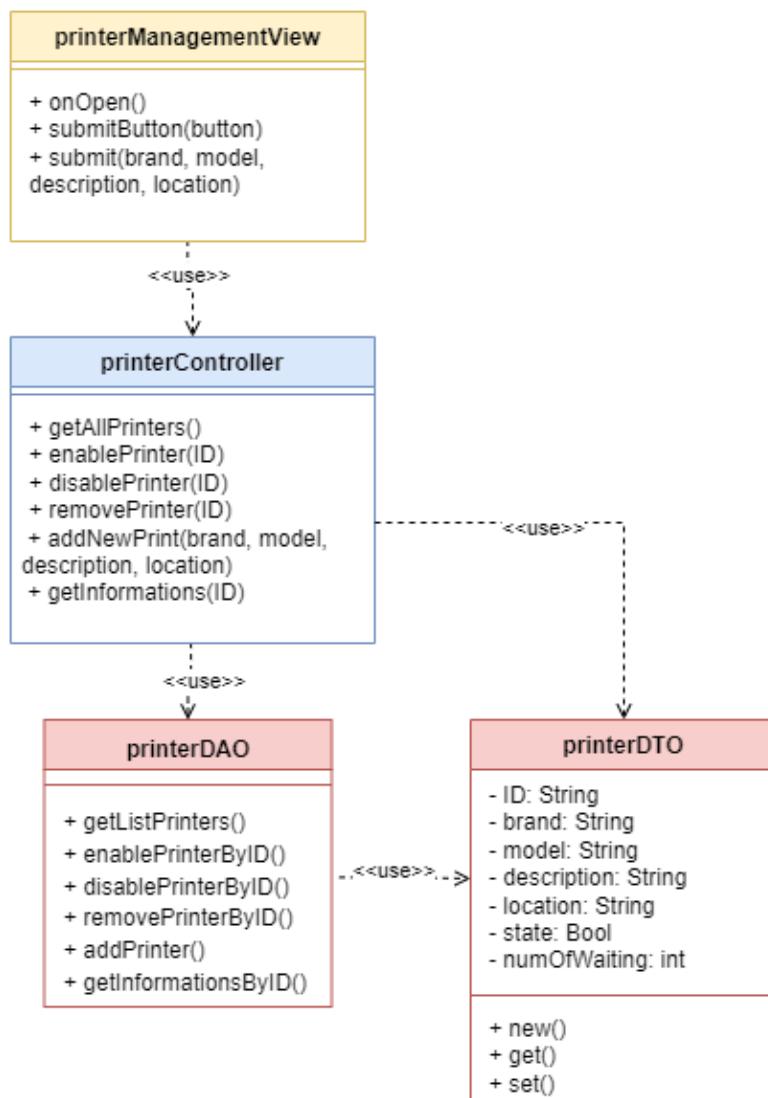
3.2 In



Hình 25: Class diagram cho module in



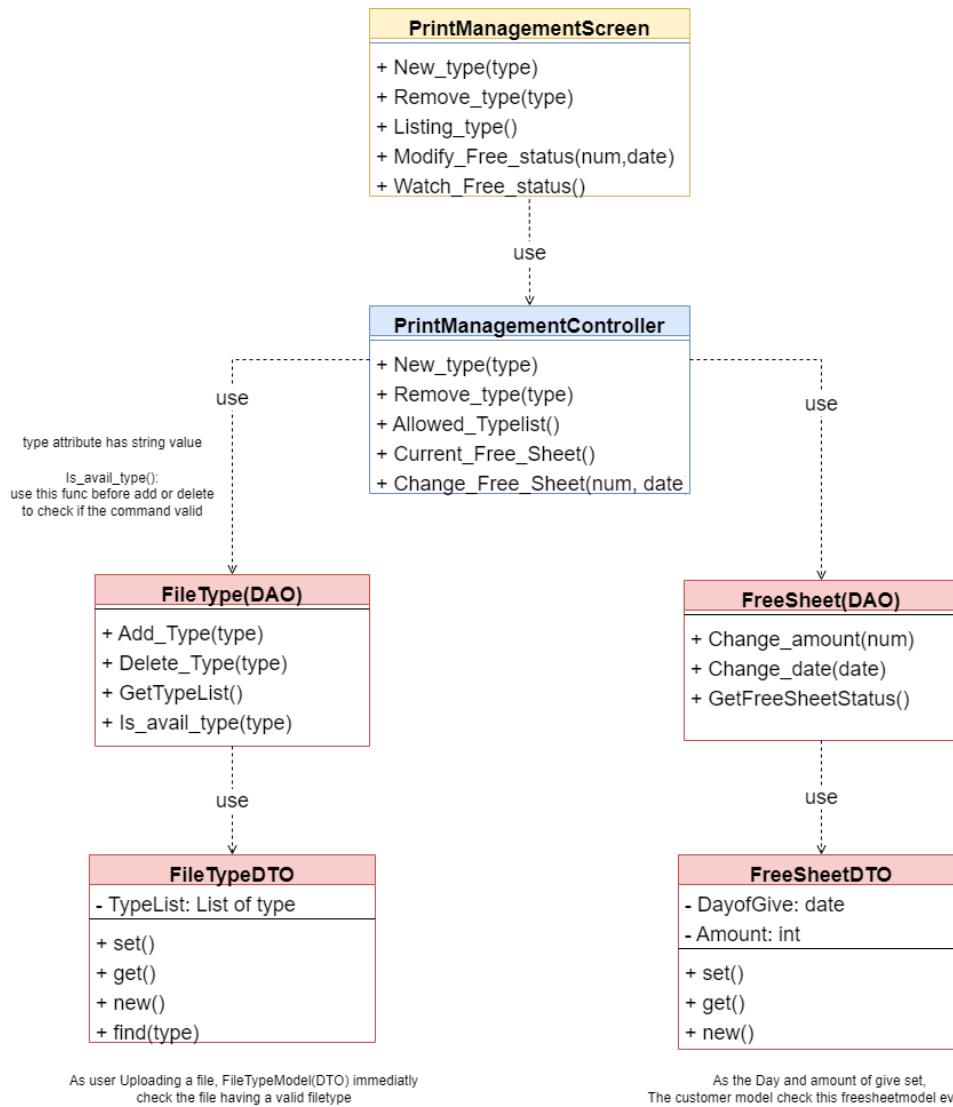
3.3 Quản lý máy in



Hình 26: Class diagram cho module quản lý máy in

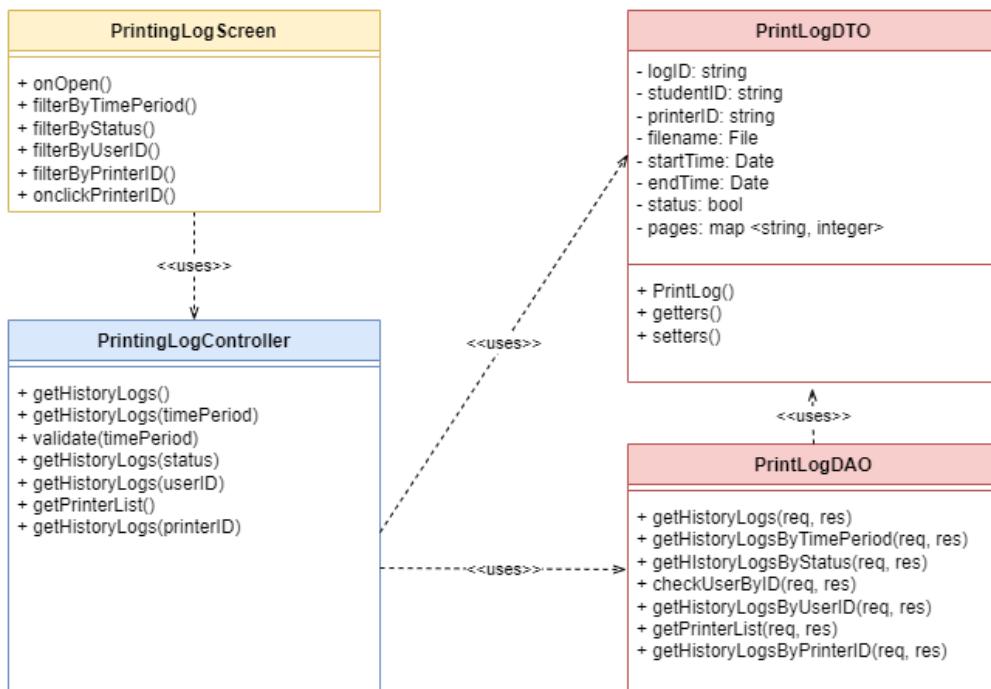


3.4 Quản lý in ấn (bao gồm định dạng tệp cho phép in và quản lý tặng giấy)



Hình 27: Class diagram cho module quản lý in ấn (bao gồm định dạng tệp cho phép in và quản lý tặng giấy)

3.5 Lịch sử in



Hình 28: Class diagram cho module lịch sử in

4 MVP

Để quan sát rõ hơn, thầy (cô) có thể truy cập:

- **Link Figma:** [Xem tại đây](#)
- **Link Prototype:** [Xem tại đây](#)

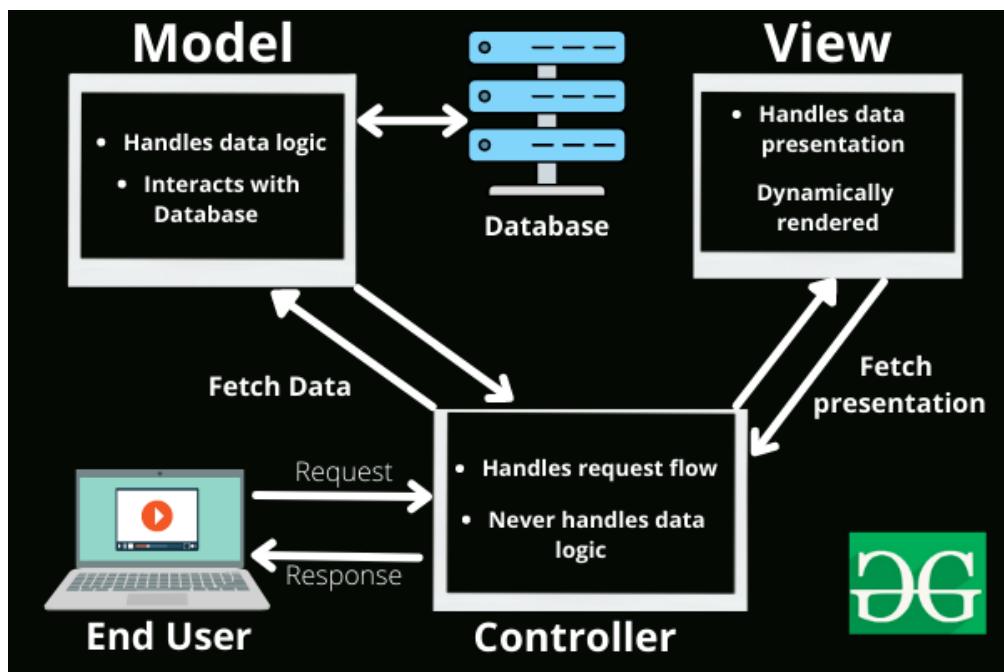
Bởi vì nhóm sử dụng bản miễn phí để soạn thảo Latex và dung lượng file quá lớn do đó nhóm chia phải chia nhỏ phần báo cáo MVP-1 và 2 sang một báo cáo khác. Mong thầy (cô) thông cảm.

Link truy cập tại: [Link xem báo cáo phần MVP](#)

III Thiết kế kiến trúc

1 Kiến trúc MVC

Nhóm tác giả lựa chọn một trong những kiến trúc rất phổ biến đó là MVC (Model - View - Controller) để thiết kế hệ thống HCMUT_SPSS.



Hình 29: Mô hình kiến trúc MVC (tham khảo từ GeeksforGeeks.org)

1.1 Ưu và nhược điểm của MVC

Ưu điểm của MVC:

- Tách biệt giữa các thành phần: Mô hình MVC tách biệt các thành phần của ứng dụng thành Model, View và Controller. Điều này giúp làm rõ trách nhiệm của từng thành phần, dễ quản lý và duy trì mã nguồn.
- Dễ bảo trì: Việc tách biệt các thành phần cho phép bạn dễ dàng thay đổi hoặc cải tiến một phần của ứng dụng mà không cần sửa đổi toàn bộ mã nguồn.
- Phát triển đồng bộ: Mô hình MVC cho phép nhiều lập trình viên làm việc cùng một lúc trên các phần khác nhau của ứng dụng mà không gây xung đột về mã nguồn.
- Tích hợp dễ dàng: Có thể thay đổi hoặc thay thế một phần của ứng dụng mà không cần ảnh hưởng đến các phần khác, giúp tích hợp các thành phần mới một cách dễ dàng.
- Dễ kiểm thử: Mô hình MVC dễ kiểm thử vì bạn có thể kiểm tra riêng lẻ các thành phần Model, View và Controller mà không cần chạy toàn bộ ứng dụng.



Nhược điểm của MVC:

- Độ trễ cao: Sử dụng mô hình MVC có thể tạo ra một chút độ trễ do việc truyền thông qua các thành phần khác nhau.
- Khó quản lý khi ứng dụng phức tạp: Đối với các ứng dụng rất phức tạp, quản lý các thành phần và tương tác giữa chúng có thể trở nên khó khăn.

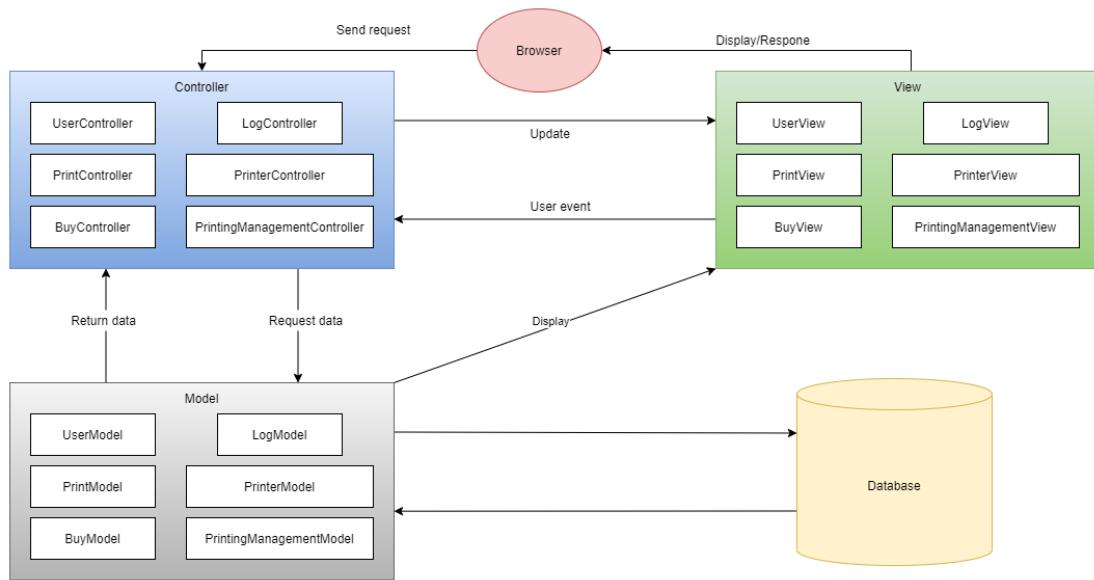
1.2 Lí do lựa chọn MVC thay cho Layered architecture

Từ việc tìm hiểu và phân tích những ưu nhược điểm của kiến trúc MVC. Nhóm tác giả quyết định lựa chọn kiến trúc này bởi vì:

- Vì MVC tách biệt giữa các thành phần nên việc phân công giữa các thành viên sẽ dễ dàng hơn. Hơn nữa việc tích hợp và phát triển đồng bộ sẽ phù hợp hơn so với kiến trúc layered cho dự án gồm nhiều thành viên bởi trong kiến trúc layered mỗi tầng là một khối chức năng cụ thể, các tầng bên trên sẽ sử dụng dịch vụ của tầng biên dưới khi đó các thành viên hiện thực các khối chức năng khác nhau sẽ khó có thể cùng nhau kiểm thử module của mình.
- MVC được hỗ trợ bởi nhiều framework và thư viện. Điều này giúp việc phát triển ứng dụng MVC trở nên dễ dàng hơn so với kiến trúc Layered
- SPSS là một hệ thống vừa và nhỏ do đó MVC sẽ giảm bớt khó khăn trong việc quản lý ứng dụng.
- Hơn hết, MVC là một mô hình phổ biến và được áp dụng rộng rãi. Vì thế việc sử dụng mô hình này như một cơ hội để nhóm tác giả tìm hiểu và vận dụng một cách hiệu quả sau này.

2 Architectural diagram và Deployment diagram

2.1 Architectural diagram cho toàn hệ thống



Hình 30: Architectural diagram

Để xem rõ hơn bản vẽ của nhóm, truy cập tại [đây](#)

- Model:

- **UserModel:** Quản lý thuộc tính và logic của User để xác thực tài khoản.
- **PrintModel:** Quản lý thuộc tính in ấn và tài liệu.
- **PrinterModel:** Quản lý các thuộc tính và xử lý logic dữ liệu liên quan tới máy in.
- **Printing Management Model:** Quản lý thông tin quản lý in ấn. Gồm 2 model đó là FreeSheetModel (quản lý thông tin về số giấy, ngày tặng miễn phí cho user) và FileTypeModel (quản lý dữ liệu về các loại file được phép in).
- **LogsModel:** Quản lý và lưu trữ lịch sử in ấn.
- **BuyModel:** Quản lý lưu trữ việc mua giấy.

- View:

- **UserView:** Phần giao diện người dùng liên quan đến xác thực.
- **PrintView:** Giao diện người dùng cho việc in tài liệu.
- **PrinterView:** Giao diện người dùng cho quản lý máy in.
- **LogsView:** Giao diện người dùng cho xem lịch sử in ấn.
- **BuyView:** Giao diện người dùng cho việc mua giấy.

- Controller:

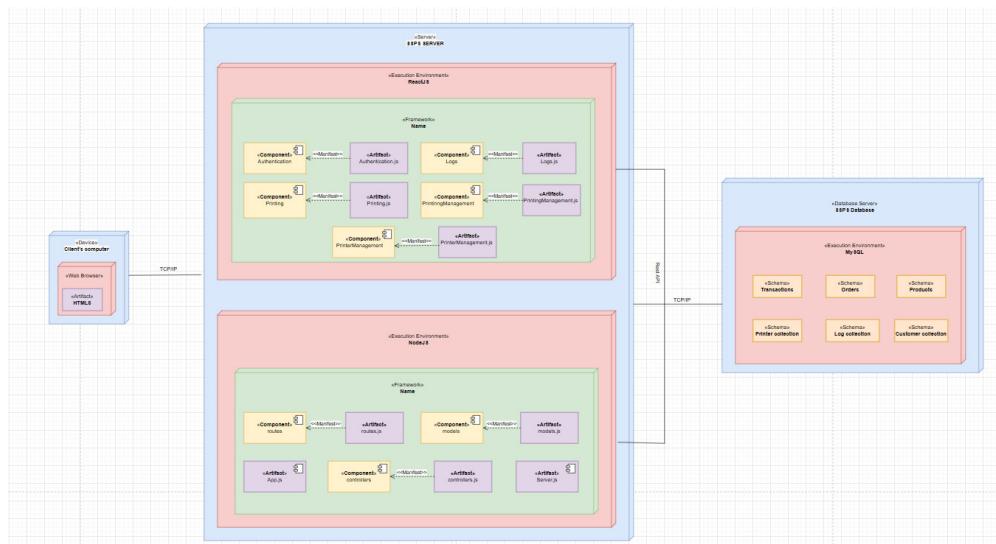
- **UserController:** Điều khiển logic và tương tác với UserModel và UserView. Kết nối với Google/HCMUT_SSO.
- **PrintController:** Điều khiển logic in ấn và sự tương tác giữa PrintModel và PrintView
- **PrinterController:** Điều khiển logic quản lý máy in, tương tác với PrinterModel và PrinterView.
- **Printing Management Controller:** Điều khiển logic quản lý in ấn, tương tác với Printing Management Model và Printing Management View.
- **LogsController:** Điều khiển logic xem lịch sử in ấn, tương tác với Logs Model và Logs View. Ngoài ra còn tương tác với UserModel và PrinterModel để yêu cầu một số thông tin như ID user, ID máy in.
- **BuyController:** Điều khiển logic và tương tác với BuyModel và BuyView

Sự tương tác giữa các thành phần:

- Những tương tác của người dùng lên phần view sẽ trở thành sự kiện để gọi các hàm tương ứng ở controller.
- Các hàm trong controller nếu cần có thể lấy dữ liệu từ hệ thống ở lớp model, xử lí và gửi phản hồi tới người dùng thông qua view
- Lớp model được dùng để truy vấn dữ liệu hoặc cập nhật dữ liệu trong database, cung cấp dữ liệu cho controller xử lí. Có thể gửi thông báo thẳng đến view (không cần thông qua controller).
- Browser được dùng để thực hiện các giao thức HTTP response/ request giữa client và server và hiển thị kết quả ra màn hình của người dùng.

2.2 Deployment diagram cho toàn hệ thống

2.2.1 Bản vẽ Deployment Diagram



Hình 31: Deployment diagram



Để xem rõ hơn bản vẽ của nhóm, truy cập tại [đây](#).

2.2.2 Mô tả Deployment Diagram

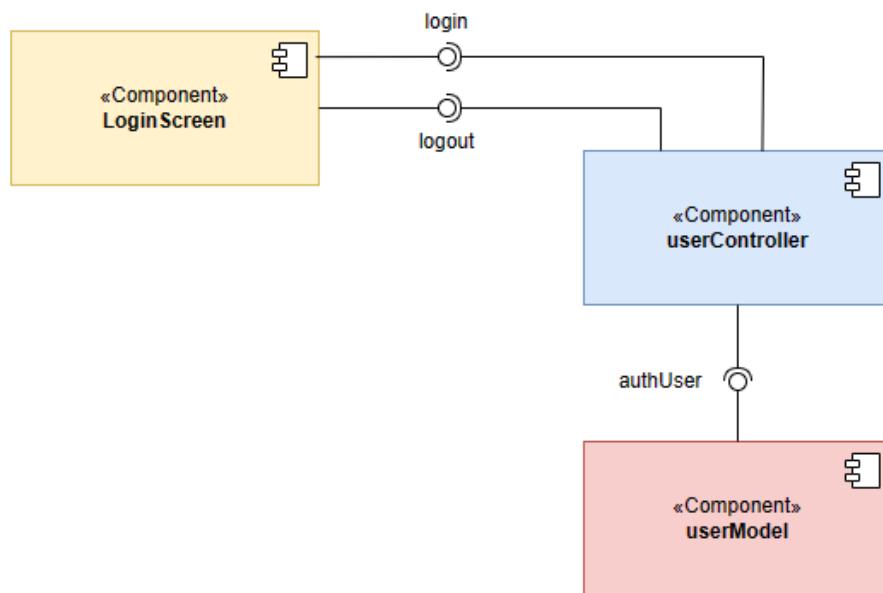
- Hệ thống là một ứng dụng Web, được phát triển theo kiến trúc MVC, bao gồm 3 máy chủ chính: Client, Server và Database Server.
- Máy chủ Client sẽ kết nối với Server thông qua giao thức TCP/IP, dữ liệu nhận được từ Server là các file hiện thực giao diện và sẽ được hiển thị trên browser website tại máy tính người dùng (Client) thông qua trình đọc HTML5 được nhúng trong trình duyệt.
- Máy chủ Server bao gồm môi trường thực thi (execution environment) các module chức năng chính của hệ thống. Mỗi trường thực thi phía **back-end** là **NodeJS**, sử dụng Framework là **Express.js**, bao gồm 5 component chính là:
 - **Authentication**: Chứa mã nguồn cho module xác thực người dùng.
 - **Printing**: Chứa mã nguồn cho module in ấn.
 - **PrinterManagement**: Chứa mã nguồn cho module quản lý máy in.
 - **PrintingManagement**: Chứa mã nguồn cho module quản lý in ấn.
 - **Logs**: Chứa mã nguồn cho module xem lịch sử in ấn.

Mỗi trường thực thi phía **front-end** là **ReactJS**, sử dụng Framework là **Material UI**, bao gồm 5 component chính là:

- **Routes**: Chứa các tệp định tuyến cho từng module.
 - **Controllers**: Chứa các tệp điều khiển cho từng module để xử lý logic kinh doanh.
 - **Models**: Chứa các tệp mô hình dữ liệu (ORM) cho cơ sở dữ liệu.
 - **App.js**: Cấu hình ứng dụng Express và gắn các tệp định tuyến.
 - **Server.js**: Khởi động máy chủ Node.js.
- Máy chủ Server sẽ kết nối với Database Server thông qua giao thức TCP/IP, để truy xuất dữ liệu hiển thị lên giao diện hoặc cập nhật lại dữ liệu mỗi khi người dùng tương tác với ứng dụng, sau đó dữ liệu sau khi cập nhật sẽ được truyền qua giao thức TCP/IP sang Server và truyền sang máy Client để cập nhật lại giao diện.
 - Máy chủ Database bao gồm môi trường thực thi là DBMS MySQL. Máy chủ Database lưu trữ các thông tin bao gồm: **Log collection**, **Customer collection**, **Printer collection**, **Transactions**, **Orders**, **Products** ở dạng bảng có cấu trúc và các mối quan hệ với nhau.

3 Component diagram

3.1 Xác thực

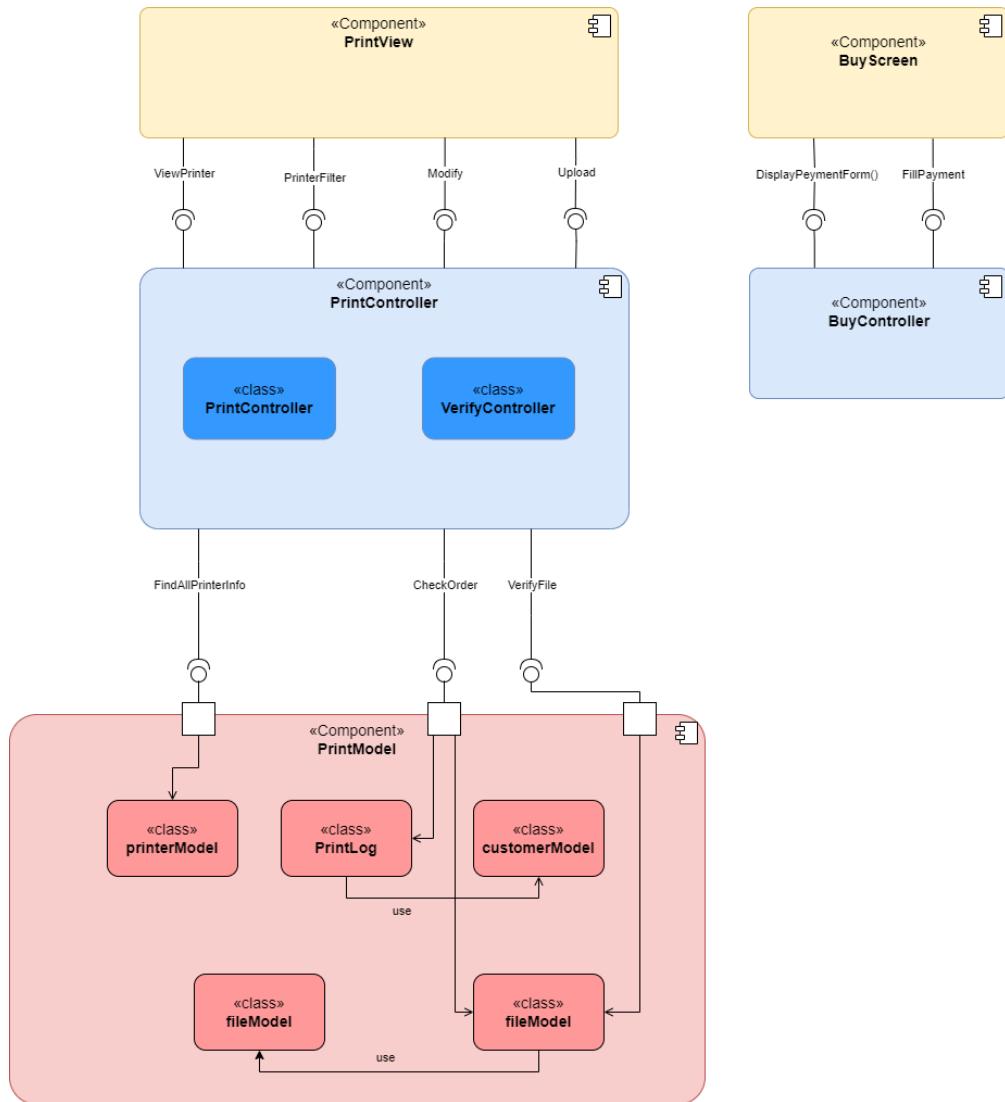


Hình 32: Component diagram cho module xác thực

Mô tả:

Trong component diagram của module xác thực, tầng view gồm có 1 component đó là LoginSrceen với chức năng là cung cấp giao diện cho người dùng đăng nhập tài khoản. Tầng controller gồm có 1 component đó là userController với chức năng là cung cấp các phương thức để xác thực tài khoản. Tầng model gồm có 1 component đó là userModel với chức năng chính là lưu trữ và thực hiện truy vấn dữ liệu liên quan tới tài khoản người dùng (xác thực tài khoản).

3.2 In

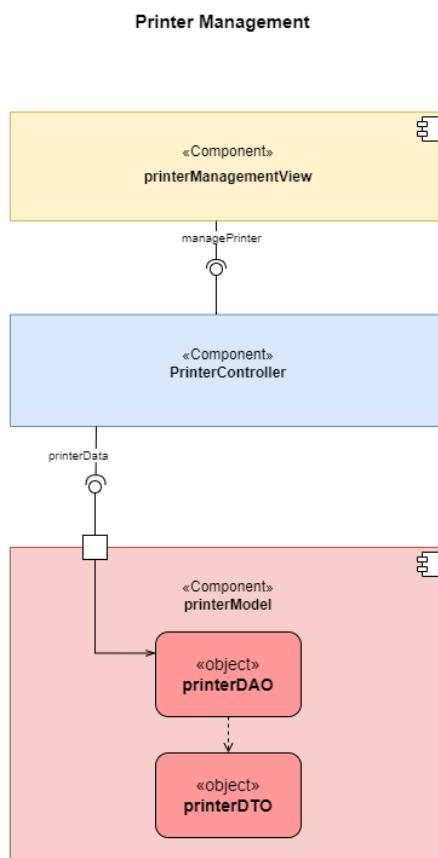


Hình 33: Component diagram cho module in

Mô tả:

Trong component diagram của module in, tầng view gồm có 2 component là **PrintView** với chức năng cung cấp giao diện cho người dùng thực hiện các thao tác (Chọn máy in, chuyển sang mua giấy) và **BuyScreen** với chức năng cung cấp giao diện cho người dùng thực hiện mua giấy. Tầng controller gồm có 2 component là **PrintController** bao gồm 2 class là **PrintController** và **VerifyController** với chức năng cung cấp các phương thức tìm kiếm thông tin và chọn máy in, component **BuyController** với chức năng cung cấp các phương thức tạo form mua giấy. Tầng model gồm có 1 component là **PrintModel** với chức năng chính là lưu trữ và thực hiện truy vấn dữ liệu liên quan đến việc in ấn (lấy thông tin máy in, kiểm tra và lưu trữ thông tin in ấn).

3.3 Quản lý máy in



Hình 34: Component diagram cho module quản lý máy in

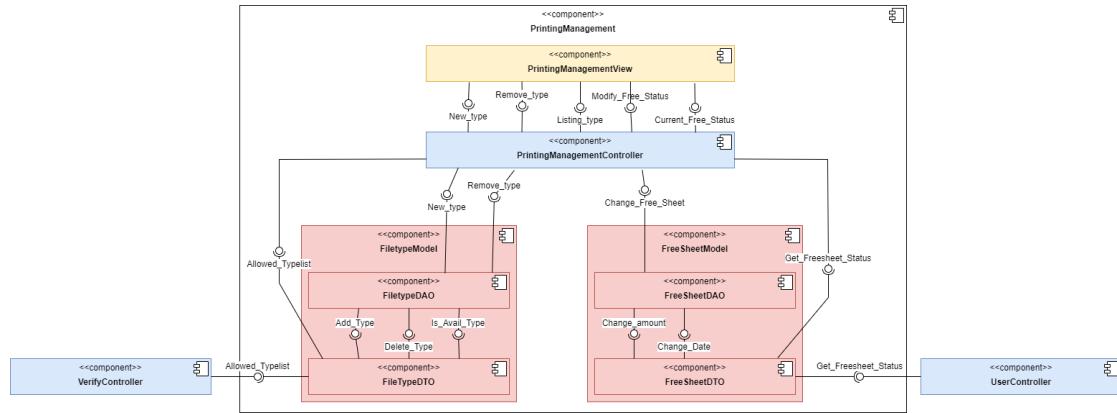
Mô tả:

Trong component diagram của module quản lý máy in, tầng view gồm 1 component đó là printerManagementView với chức năng là cung cấp giao diện cho SPSO quản lý máy in (thêm, xóa, kích hoạt và vô hiệu hóa máy in, xem). Tầng controller gồm có 1 component đó là printerController với chức năng là cung cấp các phương thức quản lý máy in đó là thêm, xóa, kích hoạt, vô hiệu hóa và xem. Tầng model gồm có 1 component đó là printerModel với chức năng chính là lưu trữ và thực hiện truy vấn dữ liệu liên quan đến máy in (thêm, xóa, kích hoạt, vô hiệu hóa, xem). Trong component printerModel thì sẽ có 2 object là printerDAO (giúp truy vấn đến cơ sở dữ liệu) và printerDTO (lưu trữ dữ liệu của máy in).

Các interface:

- **managePrinter:** cung cấp các phương thức giúp quản lý máy in (thêm, xóa, kích hoạt, vô hiệu hóa, xem).
- **printerData:** cung cấp API lấy dữ liệu máy in để quản lý (thêm, xóa, kích hoạt, vô hiệu hóa, xóa).

3.4 Sửa danh mục định dạng tệp cho phép & Quản lý tặng giấy miễn phí



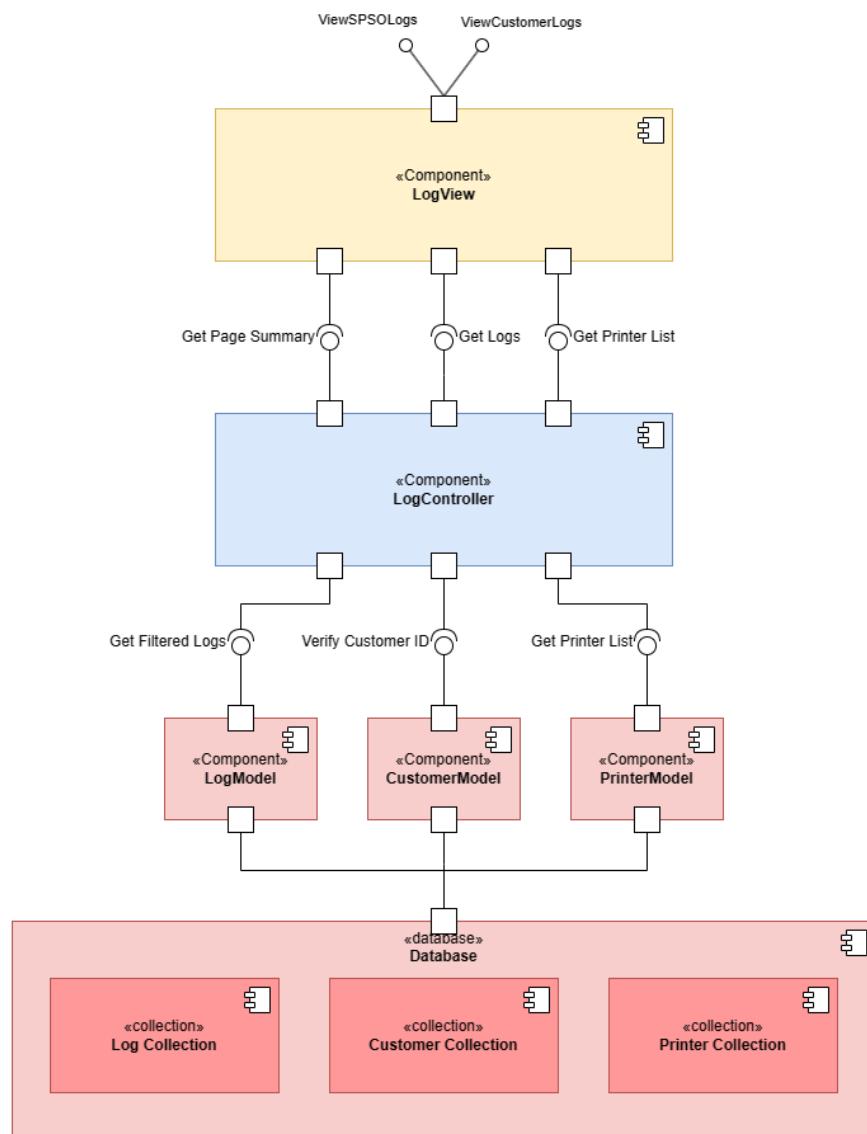
Hình 35: Component diagram cho module quản lý định dạng

Mô tả:

Thành phần PrintingManagementView chứa các giao diện cần thiết cho người dùng để thực hiện yêu cầu thêm/bớt kiểu tệp được cho phép in (New_type), (Remove_type), xem danh sách các kiểu tệp được phép in (Listing_type) và xem hoặc chỉnh sửa ngày tháng tặng giấy miễn phí tới thành phần PrintingManagementController.

Thành phần PrintingManagementController chứa các bộ xử lý những truy vấn đầu vào từ View, sau đó truyền những yêu cầu tiếp theo tương ứng tới lớp FileTypeModel hoặc FreeSheetModel. Thành phần Model nhận các yêu cầu từ Controller và thực hiện nhiệm vụ như truy xuất, thay đổi cơ sở dữ liệu về danh sách kiểu tệp cho phép in hay ngày tháng tặng giấy.

3.5 Lịch sử in



Hình 36: Component diagram cho module lịch sử in

Mô tả:

Lớp View bao gồm một thành phần LogView, cung cấp hai giao diện ViewSPSOLogs và ViewCustomerLogs để truy xuất dữ liệu lịch sử in ấn của SPSO và khách hàng. Lớp Controller có thành phần LogController, cung cấp giao diện cho LogView, bao gồm việc lấy dữ liệu lịch sử



in, tóm tắt số trang và lấy danh sách máy in đang hoạt động. Lớp Model bao gồm các thành phần LogModel, CustomerModel và PrinterModel. LogController sử dụng PrinterModel để lấy danh sách máy in, sử dụng CustomerModel để kiểm tra ID khách hàng được nhập vào bộ lọc và sử dụng LogModel để lấy dữ liệu lịch sử in án từ cơ sở dữ liệu.

IV Hiện thực - Sprint1

1 Cài đặt repository (github, bitbucket, etc) cho việc quản lý phiên bản.

Github repository của nhóm có thể được truy cập tại [đây](#).

The screenshot shows the GitHub repository page for 'SSPS_HCMUT'. At the top, there are buttons for Pin, Unwatch (1), Fork (0), Star (0), and a dropdown for 'About'. Below this, it says 'No description, website, or topics provided.' and shows a 'Readme' link. The 'Activity' section shows 9 commits from 'ThangPham2508' made 9 minutes ago. The commits are: 'modify READMEs content and replace old logo', 'code', 'documentation', and 'README.md'. To the right, there are sections for 'About', 'Releases' (no releases published), 'Packages' (no packages published), and 'Languages' (JavaScript 0.0%, Other 1.0%).

Hình 37: Trang chủ Github Repository

2 Tài liệu và thư mục cho Requirement, System modelling và Architectural design.

Repository của dự án bao gồm hai thư mục chính:

- **documentation:** thư mục này chứa các tài liệu liên quan đến dự án, cấu trúc thư mục bao gồm:



The screenshot shows a GitHub repository interface for the 'SSPS_HCMUT' project. On the left, there's a sidebar with a 'Files' tab and a tree view of the repository structure. The 'main' branch is selected. Under 'main', there are 'code', 'documentation', and 'latex' folders. The 'documentation' folder contains files like 'CNPMP_HK231.pdf', 'README.md', and another 'README.md'. The 'latex' folder contains 'CNPMP_HK231.pdf' and 'README.md'. The 'README.md' file in the 'documentation' folder is open, displaying the following content:

SSPS_HCMUT - Documentation

This folder contains the documentation for Student Smart Printing Service, a web-based application that allows students to print documents from their devices at any campus printer. The documentation includes the following sections:

- Requirement Elicitation: This section describes the context of the project, the functional and non-functional requirements of the service and use case diagrams for the whole application.
- System Modelling: This section presents the activity diagrams, sequence diagrams and class diagrams that illustrate the behavior and structure of the service.

Hình 38: Thư mục documentation

- Tập pdf chính (CNPMP_HK231.pdf) chứa phiên bản hoàn chỉnh của tài liệu.
- **latex:** thư mục này chứa các tập mã nguồn latex cho tài liệu, được tổ chức như sau:

The screenshot shows the 'SSPS_HCMUT / documentation / latex' folder on GitHub. The sidebar shows the 'main' branch and the 'documentation' folder is expanded. Inside 'documentation', the 'latex' folder is selected. The 'latex' folder contains 'Images', 'Pages', 'Assignment_report.tex', 'hcmut.png', 'CNPMP_HK231.pdf', 'README.md', and another 'README.md'. The 'Assignment_report.tex' file is open, showing its LaTeX code.

Hình 39: Thư mục latex

- * **Images:** thư mục này chứa tất cả các hình ảnh được sử dụng trong tài liệu.
- * **Pages:** thư mục này chứa các tập latex riêng lẻ cho mỗi phần của tài liệu. Nó cũng chứa một tập pdf của phần đó được trích xuất từ tập pdf chính.
- * Mỗi thư mục được mô tả ở trên được tổ chức theo các phần, bao gồm Requirement Elicitation, System Modelling và Architectural Design.



The screenshot shows a Git commit history for the 'Images' directory. The commits are as follows:

Name	Last commit message	Last commit date
..		
Architecture Design	modify READMEs content and replace old logo	39 minutes ago
Box-line + Deployment	modify READMEs content and replace old logo	39 minutes ago
Figma	modify READMEs content and replace old logo	39 minutes ago
Requirement Elicitation	modify READMEs content and replace old logo	39 minutes ago
System Modelling	modify READMEs content and replace old logo	39 minutes ago

Hình 40: Thư mục Images

The screenshot shows a Git commit history for the 'Pages' directory. The commits are as follows:

Name	Last commit message	Last commit date
..		
Architecture_design	modify READMEs content and replace old logo	40 minutes ago
Implement - Sprint1	modify READMEs content and replace old logo	40 minutes ago
Requirement_elicitation	modify READMEs content and replace old logo	40 minutes ago
System_modelling	modify READMEs content and replace old logo	40 minutes ago
main.tex	modify READMEs content and replace old logo	40 minutes ago

Hình 41: Thư mục Pages

- **code:** thư mục này chứa mã nguồn của dự án. Mã nguồn được tổ chức như sau:

- **frontend:** Thư mục này chứa mã nguồn cho frontend, được xây dựng bằng React.
Thư mục src bao gồm các thư mục con sau:
 - * **assets:** Thư mục này chứa hình ảnh, icon và các tệp khác được sử dụng trong ứng dụng.
 - * **components:** Thư mục này chứa các thành phần tái sử dụng của React.
 - * **pages:** Thư mục này chứa các thành phần React cho mỗi trang của ứng dụng.
 - * **slices:** Thư mục này chứa các slice của Redux để quản lý trạng thái của ứng dụng.



- * **main.jsx:** Tệp này là điểm nhập của ứng dụng, bao gồm hiển thị thành phần App và thiết lập Redux và routing.

The screenshot shows a GitHub repository interface for 'SSPS_HCMUT / code / frontend / src /'. On the left, there's a file tree showing the directory structure: main, code, frontend, public, src (which is expanded to show assets, components, pages, slices, App.jsx, index.css, main.jsx, store.js, .eslintrc.cjs, .gitignore, README.md, index.html, package-lock.json, and package.json). On the right, there's a list of commits from 'ThangPham2508' with the message 'modify READMEs content and replace old logo'. The commits are listed as follows:

Name	Last commit message	Last commit date
..		
assets	modify READMEs content and replace old logo	41 minutes ago
components	modify READMEs content and replace old logo	41 minutes ago
pages	modify READMEs content and replace old logo	41 minutes ago
slices	add docs	2 days ago
App.jsx	add docs	2 days ago
index.css	add docs	2 days ago
main.jsx	add docs	2 days ago
store.js	add docs	2 days ago

Hình 42: Thư mục src

- **backend:** Thư mục này sẽ chứa mã nguồn cho máy chủ back-end, đó sẽ xử lý việc giao tiếp với máy in và cơ sở dữ liệu (sẽ được thêm sau).

3 Usability test bằng MVP1. Tổng hợp phản hồi và cải thiện

3.1 Tiến hành kiểm tra việc sử dụng với UI được phát triển trong MVP1

Để tiến hành kiểm tra, nhóm tác giả quyết định tạo một nền tảng và lấy khảo sát từ mọi người về sử dụng website của nhóm để thấy được liệu họ có thích nó không. Thịt nhóm nhận thấy hầu hết người dùng cảm thấy thoải mái với giao diện người dùng của MVP1 và chỉ cần cải tiến một chút là có thể có cái nhìn tốt hơn ở MVP2. Figma thì khá là khó để chứng minh cho người dùng thấy rằng ta tính làm gì với website. Các câu hỏi trên biểu mẫu google được dùng chủ yếu để lấy ý kiến sơ bộ hoặc cơ bản về giao diện người dùng theo quan điểm của họ. Sau cuộc khảo sát, nhóm tác giả đã thực hiện những thay đổi để làm cho hệ thống thân thiện hơn với người sử dụng và phục vụ trải nghiệm người dùng tốt hơn.

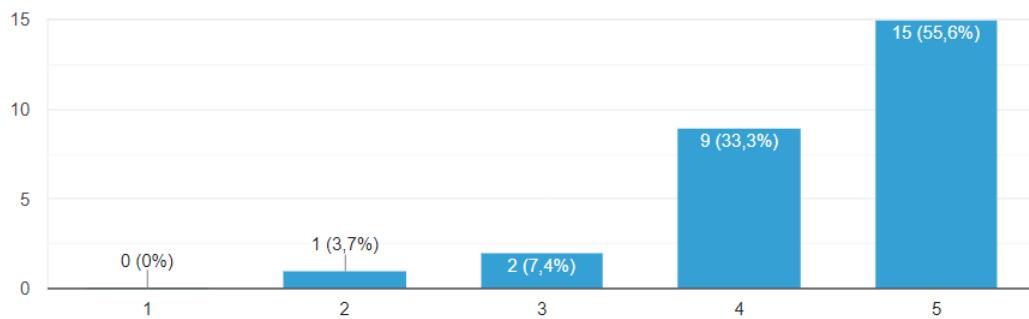
Biểu mẫu: [Link khảo sát](#).

3.1.1 Trang chủ

Bạn đánh giá giao diện Trang chủ (homepage) của website ở mức nào?

Sao chép

27 câu trả lời



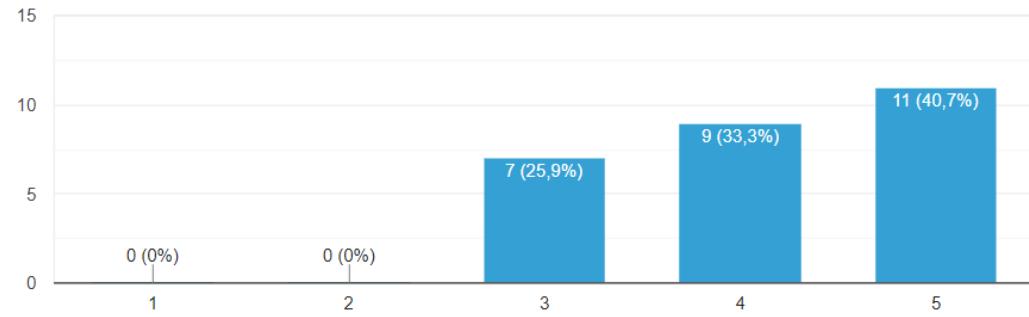
Hình 43: Khảo sát giao diện Trang chủ

3.1.2 Đăng nhập

Bạn đánh giá giao diện Đăng nhập của website ở mức nào?

Sao chép

27 câu trả lời



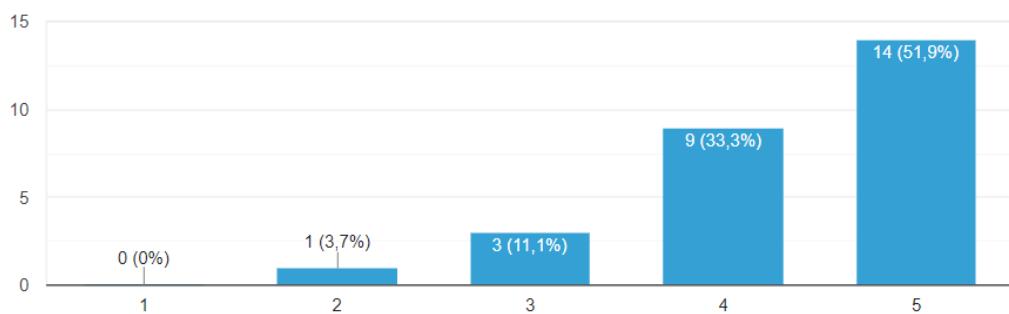
Hình 44: Khảo sát giao diện Đăng nhập

3.1.3 In ngay

Bạn đánh giá giao diện IN NGAY của website ở mức nào?

Sao chép

27 câu trả lời



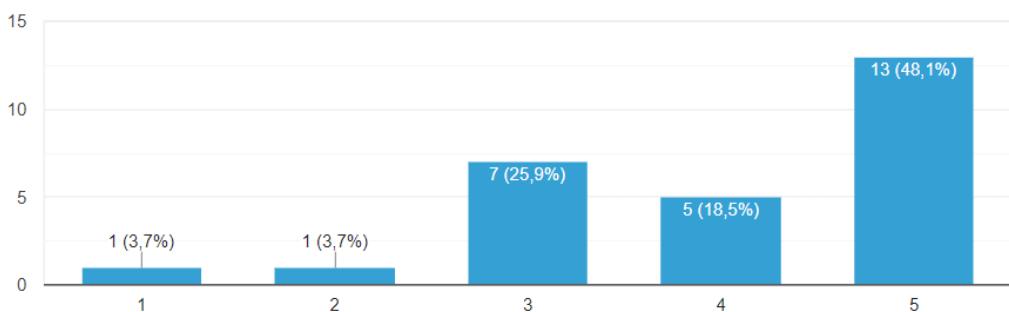
Hình 45: Khảo sát giao diện In ngay

3.1.4 Xem trước và cấu hình file in

Bạn đánh giá giao diện phần XEM TRƯỚC & CẤU HÌNH FILE IN của website ở mức nào?

Sao chép

27 câu trả lời



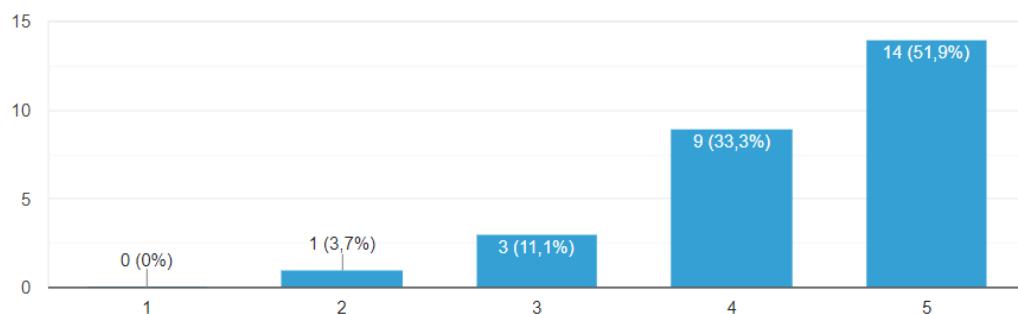
Hình 46: Khảo sát giao diện Xem trước và cấu hình file in

3.1.5 Quản lý máy in

Bạn đánh giá giao diện phần QUẢN LÝ MÁY IN của website ở mức nào?

Sao chép

27 câu trả lời



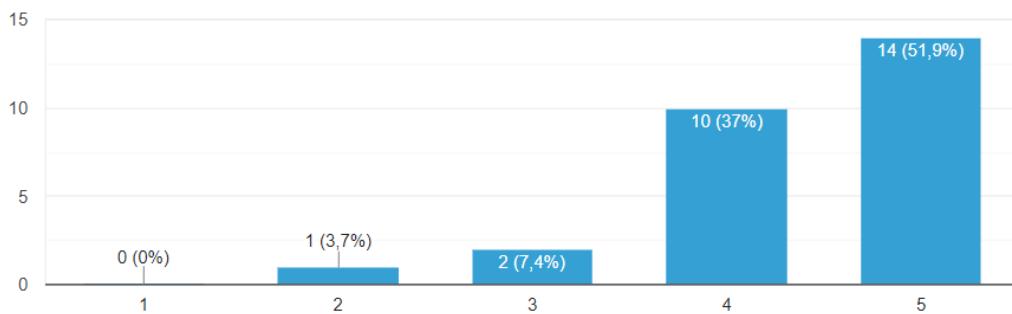
Hình 47: Khảo sát giao diện Quản lý máy in

3.1.6 Thêm máy in

Bạn đánh giá giao diện phần THÊM MÁY IN của website ở mức nào?

Sao chép

27 câu trả lời



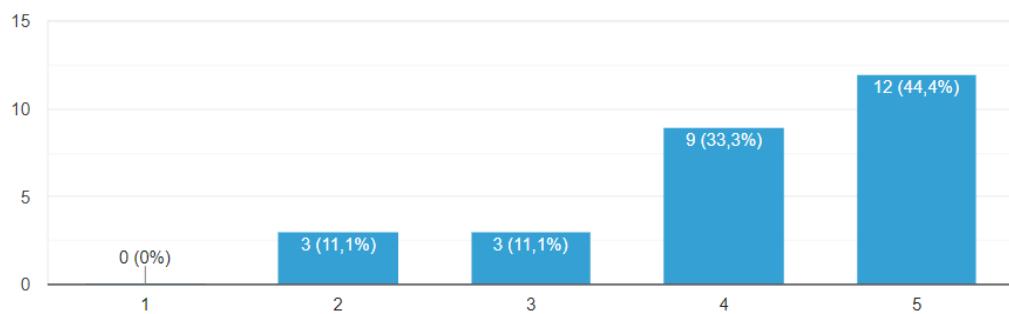
Hình 48: Khảo sát giao diện Thêm máy in

3.1.7 Lịch sử in ấn

Bạn đánh giá giao diện phần LỊCH SỬ IN ẤN của website ở mức nào?

Sao chép

27 câu trả lời



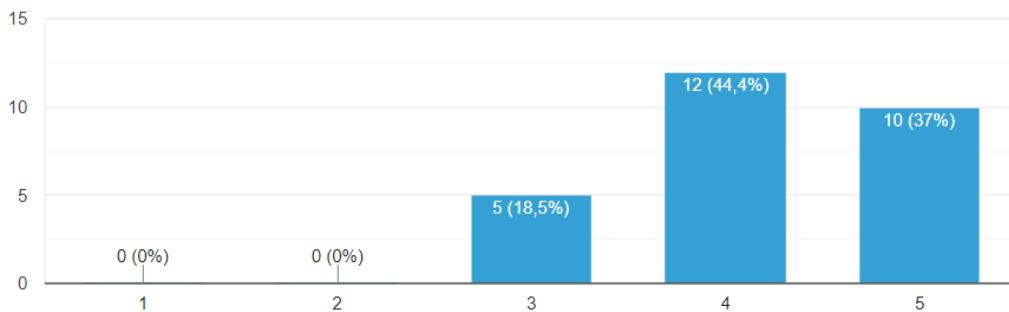
Hình 49: Khảo sát giao diện Lịch sử in ấn

3.1.8 Mua giấy in

Bạn đánh giá giao diện phần MUA GIẤY IN của website ở mức nào?

Sao chép

27 câu trả lời



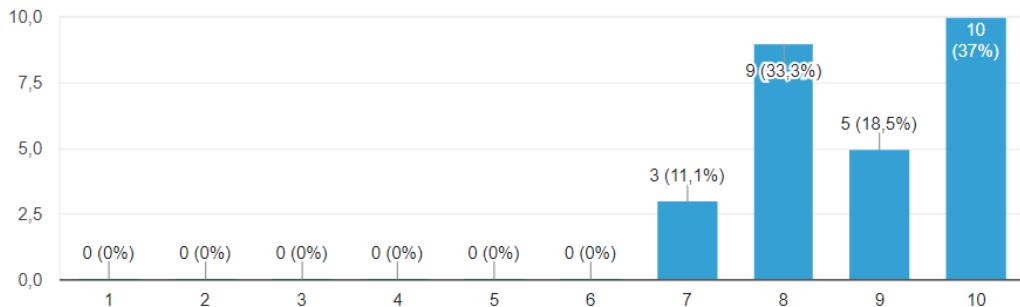
Hình 50: Khảo sát giao diện Mua giấy in

3.2 Tổng hợp phản hồi và cải thiện MVP1 thành MVP2 với trải nghiệm người dùng tốt hơn

Nhìn chung thì phần trải nghiệm người dùng cho MVP1 khá ổn với điểm số được chấm từ 7 trở lên, điểm cao nhất là 10 chiếm ưu thế với 37%, điểm 8 chiếm thứ 2 với 33.3%.

Nếu nhận xét tổng thể về trải nghiệm người dùng, bạn sẽ đánh giá ở mức nào? Sao chép

27 câu trả lời

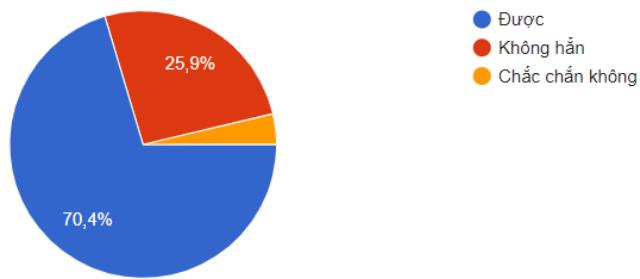


Hình 51: Khảo sát về trải nghiệm người dùng cho MVP1

Và nhóm cũng có khảo sát về việc sử dụng thành thạo website của một số người dùng thì nhận thấy phần lớn (70.4%) là có thể sử dụng thành thạo mà không cần hướng dẫn cụ thể. Tuy nhiên một số vẫn chưa thể với 25.9% là không hẳn và 3.7% chắc chắn không.

Nếu chỉ có kiến thức về chức năng của trang web và không có người hướng dẫn cụ thể, bạn có tự tin sẽ sử dụng thành thạo website không? Sao chép

27 câu trả lời



Hình 52: Khảo sát khả năng sử dụng thành thạo website không cần hướng dẫn

Sau khi tóm tắt tất cả các phản hồi, chúng tôi đã đưa ra một số cải thiện về giao diện người dùng của mình:

- Trang chủ: Sửa lại background của trang chủ cho dễ nhìn.



The screenshot shows the homepage of the University of Technology's website. At the top, there is a navigation bar with links for Home, Print History, Print Today, and Feedback. Below the navigation is a large banner featuring a colorful illustration of a modern building complex with trees and the text "CHÚNG TÔI HÂN HẠNH MANG ĐẾN GIẢI PHÁP IN ẤN THÔNG MINH". Underneath the banner, there is a section titled "SSPS CUNG CẤP DỊCH VỤ" (SSPS Provides Services) which includes three sub-sections: "In ấn dễ dàng, tiện lợi, mọi lúc, mọi nơi" (Easy printing, convenient, anytime, anywhere), "Xem lại lịch sử in ấn" (View previous print history), and "Thanh toán trực tuyến, đa nền tảng" (Online payment, multi-platform). Below this, there is a section titled "SSPS CAM KẾT MANG ĐẾN DỊCH VỤ" (SSPS Committed to Providing Services) with four icons: "THÂN THIỆN" (User-friendly), "TƯƠNG THÍCH" (Responsive), "HIỆU SUẤT" (Efficiency), and "BẢO MẬT" (Security). The footer contains the university's address, contact information, and social media links.

Hình 53: Cải thiện giao diện Trang chủ

- Button: Thống nhất lại màu sắc các nút bấm cho hài hòa, hút mắt người xem.



Trang chủ > Quản lý máy in > Thêm máy in

QUẢN LÝ MÁY IN

Thông tin máy in

Menu

Vị trí (cơ sở - tòa - phòng)

Hãng máy

Danh sách máy in

Enter printer's location

Select printer's brand

Thêm máy in

Số hiệu

Mô tả

Enter ID

Enter description

Xác nhận

Bản quyền thuộc Trường Đại học Bách Khoa - DHQG-HCM

Cơ sở Lý Thường Kiệt: 288 Lý Thường Kiệt, Phường 14, Quận 10, TP. HCM (Bản đồ)
Cơ sở Dĩ An: Khu phố Tân Lập, Phường Đông Hòa, TP. Dĩ An, Tỉnh Bình Dương (Bản đồ)

Thông tin liên hệ và hỗ trợ

Sinh viên
MyBK
Quý khách
Email
Biểu mẫu thông tin liên hệ

LƯỢT TRUY CẬP 43,508,658

LIÊN KẾT MẠNG XÃ HỘI

f i n t

Hình 54: Cải thiện màu sắc các nút

- Lịch sử in ấn: Chỉnh sửa màu sắc cho đỡ chói và hài hòa hơn, sắp xếp mọi thứ có cấu trúc, đều đặn hơn.



LỊCH SỬ IN

Trang chủ > Lịch sử in > Tất cả

Tim kiếm...

1- 4 trong số 9 < >

Tên file	Số bản in:3	Máy: CS2-1019
IMG	Số trang: ✓ Đã in: date.time	
Tên file	Số bản in:2	Máy: CS2-1014
PDF	Số trang: Chờ duyệt	
.DOCX	Số bản in:4	Máy: CS2-1019
	Số trang: ✓ Đã in: date.time	
Tên file	Số bản in:5	Máy: CS2-1019
PDF	Số trang: Đã hủy	

Tổng giấy: 134/3079

LIÊN KẾT MẠNG XÃ HỘI

Bản quyền thuộc Trường Đại học Bách Khoa - DHQG-HCM

LƯỢT TRUY CẬP 43,508,658

Hình 55: Cải thiện giao diện Lịch sử in ấn

- Mua giấy in: Chính sửa cỡ chữ cho phù hợp hơn.



The screenshot shows the university's online printing system. At the top, there is a navigation bar with links to 'TRANG CHỦ' (Home), 'LỊCH SỬ IN' (Print History), 'IN NGAY' (Print Now), 'PHẢN HỒI' (Feedback), and a user profile icon. Below the navigation bar, the main content area displays the following information:

- Trang chủ > Mua thêm giấy**
- Số giấy còn lại: 49 tờ A4**
- Ngày cập nhật: 14:08 11/10/2023**
- MUA GIẤY**
- Menu**:
 - Mua giấy** (highlighted)
 - Lịch sử thanh toán**
- Số lượng A4**: 79
- Đơn giá: 200đ/tờ A4**
- Gói khuyến mãi**:
 - Gói 100 tờ A4**: Giảm 5%
 - Gói 200 tờ A4**: Giảm 10%
 - Gói 500 tờ A4**: Giảm 20%
- Đơn hàng**:
 - 79 x Tờ A4: 15.800đ
 - KM tựu trường: -10.000đ
 - Tổng tiền:** 5.800đ
- Mua ngay**
- Chọn hình thức thanh toán**:
 - Thanh toán qua BKPay
 - Thanh toán bằng ví MoMo
 - Thanh toán bằng ví ZaloPay **Giảm 10K**
 - Thanh toán bằng VNPay
- Thông tin liên hệ và hỗ trợ**:
 - Sinh viên
 - MyBK
 - Quý khách
 - Email
 - Biểu mẫu thông tin liên hệ
- LIÊN KẾT MẠNG XÃ HỘI**:
 - f
 - i
 - y
 - in
 - tik
- Cơ sở Lý Thường Kiệt: 288 Lý Thường Kiệt, Phường 14, Quận 10, TP. HCM (Bản đồ)**
- Cơ sở Dĩ An: Khu phố Tân Lập, Phường Đồng Hòa, TP. Dĩ An, Tỉnh Bình Dương (Bản đồ)**
- Bản quyền thuộc Trường Đại học Bách Khoa - DHQG-HCM**
- LƯỢT TRUY CẬP: 43,508,888**

Hình 56: Cải thiện giao diện Mua giấy in

V Hiện thực - Sprint2

1 Các framework sử dụng

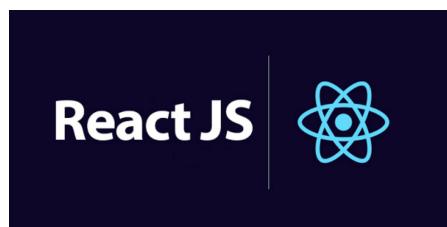
1.1 Front-end

Với front-end, nhóm tác giả sử dụng *ReactJS* để thiết kế layout và trang trí trang web một cách hiện đại, đẹp và tốn ít công sức thực hiện.

ReactJS là một thư viện của *Javascript*, giúp cho lập trình viên có thể tạo ra các giao diện phức tạp bằng việc định nghĩa các component nhỏ, đơn giản hơn với sự hỗ trợ đến từ ReactJS. Với việc tách một dự án lớn thành những phần nhỏ, ReactJS giúp cho đội dự án có thể phân



chia công việc một cách hợp lý, dễ dàng thảo luận và debug nếu có lỗi.



Hình 57: ReactJS

Bởi vì *ReactJS* đi theo hướng *SPA* (single page application), nói cách khác là ứng dụng sẽ được tải đúng một lần cho toàn bộ, lúc này người dùng sẽ mất thời gian lâu hơn cho việc tải lần đầu, song ở những lần truy cập tới, việc sử dụng trang web sẽ mượt mà hơn như đang thao tác với một ứng dụng di động, tăng trải nghiệm tương tác của người dùng.

Ngoài ra, nhóm tác giả sử dụng TailwindCSS - một Framework CSS để thiết kế front-end bởi vì khác với các framework CSS truyền thống như Bootstrap hay Foundation, Tailwind không định nghĩa sẵn các thành phần như nút, biểu mẫu, hay thanh điều hướng. Thay vào đó, Tailwind tập trung vào việc cung cấp các lớp CSS để người dùng có thể xây dựng các thành phần theo ý muốn của mình.

1.2 Back-end

Với back-end, nhóm tác giả sử dụng *NodeJS* bởi vì *NodeJS* sử dụng mô hình không đồng bộ (asynchronous) và event-driven, giúp tối ưu hóa sự xử lý đa nhiệm và tăng hiệu suất của ứng dụng. Điều này làm cho nó trở nên hiệu quả trong việc xử lý nhiều yêu cầu đồng thời.



Hình 58: NodeJS

Ngoài ra, nhóm cũng sử dụng *MongoDB* để tạo và quản lý cơ sở dữ liệu. Lí do là bởi MongoDB không yêu cầu một schema cố định, cho phép lưu trữ các bản ghi với các trường khác nhau trong cùng một bộ sưu tập (collection). Điều này giúp dễ dàng thay đổi cấu trúc của dữ liệu mà không cần phải cập nhật schema. Và hơn hết MongoDB được sử dụng rộng rãi trong cộng đồng NodeJS. Có nhiều thư viện driver MongoDB cho NodeJS giúp nhóm có thể tương tác với cơ sở dữ liệu MongoDB từ ứng dụng NodeJS một cách thuận tiện.



Hình 59: MongoDB

2 Demo sản phẩm

Bởi vì nhóm sử dụng bản miễn phí để soạn thảo Latex và dung lượng file quá lớn do đó nhóm chia phải chia nhỏ phần báo cáo MVP-1 và 2 sang một báo cáo khác. Mong thầy (cô) thông cảm.

Link truy cập tại: [Link xem báo cáo phần MVP](#)



VI Tổng kết

Link slide presentation của nhóm: [Xem tại đây](#)

Link video thuyết trình (nhóm đã thuyết trình trực tiếp trên lớp, video này được quay trước để chuẩn bị cho bài báo cáo): [Xem tại đây](#)

Link video demo sản phẩm: [Xem tại đây](#)

Trong quá trình hiện thực bài tập lớn, nhóm đã thực hiện các giai đoạn trong quá trình phát triển một phần mềm, từ mô tả và xác định các yêu cầu của dự án đến việc hiện thực giao diện và những chức năng chính của một hệ thống in ấn thông minh tại Trường Đại học Bách khoa TPHCM. Cụ thể hơn, nhóm đã xác định ngữ cảnh, quy trình nghiệp vụ, tầm vực của dự án và các yêu cầu từ người dùng, hệ thống, các yêu cầu chức năng và phi chức năng, sau đó tiếp tục đi vào mô hình hóa hệ thống và thiết kế kiến trúc.

Để làm rõ các thay đổi của hệ thống trong quá trình vận hành và mô tả cấu trúc cũng như nắm được quy trình xử lý chức năng, nhóm đã vẽ những UML Diagram như Use-case Diagram, Activity Diagram, Sequence Diagram, Class Diagram, Component Diagram, Architecture (dưới dạng Box-line) và Deployment Diagram.

Tính đến nay, hệ thống mà nhóm hiện thực cũng đã tương đối hoàn chỉnh. Phần hiện thực của nhóm đã đáp ứng cơ bản những chức năng cần có của cả khách hàng đó là: Chức năng in ấn (chọn máy in, upload & config file, mua giấy in...) và của SPSO: Chức năng quản lý máy in (thêm, sửa, xóa, bắt, tắt máy in,...), Chức năng quản lý in ấn (chỉnh sửa loại file được in và ngày tặng giấy, số giấy tặng miễn phí). Ngoài ra nhóm cũng đã đáp ứng một số yêu cầu phi chức năng như giao diện thân thiện, dễ sử dụng, xử lý dữ liệu thời gian thực,...

Bên cạnh những điểm đạt được, nhóm em cũng tự nhận thấy còn nhiều yếu điểm mà nhóm cần phải cải thiện, chủ yếu là trong khâu làm việc nhóm. Vì lượng công việc nhiều, kéo theo việc phân công công việc còn chưa đạt hiệu quả cao khiến nhóm gặp phải nhiều khó khăn, thách thức. Những vấn đề này đã phần nào ảnh hưởng đến chất lượng bài làm của nhóm tại thời điểm đó, thậm chí tạo ra mâu thuẫn giữa các thành viên. Tuy nhiên, nhóm đã không bỏ cuộc mà bình tĩnh tìm hướng giải quyết, tổ chức những lần họp mặt để cùng nhau tìm ra tiếng nói chung, giúp cho việc phối hợp cùng nhau trở nên ăn ý và gắn kết với nhau hơn.

Nhìn về mặt tích cực hơn thì các thành viên trong nhóm đã có thêm nhiều kinh nghiệm thông qua bài tập lớn lần này. Đó là kinh nghiệm làm việc nhóm, kinh nghiệm thuyết trình, kinh nghiệm trong việc giải quyết tình huống cũng như quản lý thời gian sao cho hiệu quả hơn. Và quan trọng không kém chính là kỹ năng làm việc nhóm để cùng nhau hoàn thiện một phần mềm. Có thể nói, đề tài này đã mang lại cho nhóm rất nhiều kiến thức chuyên ngành quan trọng về quá trình phát triển một phần mềm, từ việc sử dụng UML Diagram sao cho hợp lý cho đến cách dùng github và tận dụng các công cụ như Figma, ReactJS, NodeJS,...hoàn thiện dự án.

Cuối cùng, nhóm TN01_04 xin được gửi lời cảm ơn chân thành đến các thầy, cô đã nhiệt tình hướng dẫn nhóm trong suốt quá trình thực hiện bài tập lớn lần này. Nhờ vào việc được các thầy góp ý, trả lời câu hỏi cũng như tổ chức những buổi sửa bài sau mỗi phần, bài làm của nhóm đã được cải thiện và ngày càng trở nên hoàn thiện hơn.



Tài liệu

1. Learning UML 2.0, O'Reilly, Russ Miles Kim Hamilton, 2006.
2. Fundamentals of Software Architecture - An Engineering Approach, O'Reilly, Mark Richards, Neal Ford, 2020.