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

CÔNG NGHỆ PHẦN MỀM (CO3001)

BÁO CÁO BÀI TẬP LỚN

LỚP L01 - NHÓM 8

ĐỀ TÀI:

A SMART PRINTING SERVICE FOR STUDENTS AT HCMUT

GV hướng dẫn: Lê Đình Thuận

Nhóm sinh viên thực hiện:

STT	Họ và tên	MSSV
1	Nguyễn Đức Bình	2112899
2	Trần Nguyễn Thái Bình	2110051
3	Mai Hoàng Danh	2110896
4	Hồ Huy Hoàng	2110181
5	Nguyễn Duy Tùng	2115232
6	Trương Hoàng Nguyên Vũ	2112673

TP. Hồ Chí Minh, Tháng 10/2023

Contents

1 Requirement elicitation **3**

1.1 Bối cảnh và các bên liên quan trong dự án . . . 3

1.2 Các yêu cầu về functional và non-functional trong dự án . . . 5

1.2.1 Functional Requirements . . . 5

1.2.2 Non-Functional Requirements . . . 6

1.3 Use-Case Scenario . . . 8

1.3.1 Biểu đồ Use-case của toàn hệ thống SSPS . . . 8

1.3.2 Đặc tả Use-case "Print" . . . 9

1.3.3 Đặc tả Use-case "Printing Management" . . . 12

1.3.4 Đặc tả Use-case "Printing History" . . . 15

2 System modelling **17**

2.1 Activity diagrams . . . 17

2.1.1 Cung cấp thông tin in . . . 17

2.1.2 Xác nhận in . . . 19

2.1.3 Quản lý máy in . . . 20

2.1.4 Quản lý cài đặt in . . . 22

2.1.5 Xem lịch sử in . . . 24

2.1.6 Xem báo cáo định kỳ . . . 26

2.1.7 Tạo báo cáo định kỳ tự động . . . 28

2.2 Sequence Diagrams . . . 29

2.2.1 Cung cấp thông tin in . . . 29

2.2.2 Xác nhận in . . . 31

2.2.3 Quản lý máy in . . . 32

2.2.4 Quản lý cài đặt in . . . 34

2.2.5 Xem lịch sử in . . . 36

2.2.6 Xem báo cáo in . . . 38

2.3 Class Diagrams . . . 39

2.4 Thiết kế giao diện phần mềm cho những Module đã chọn . . . 44

2.4.1 Cung cấp thông tin in . . . 44

2.4.2 Xác nhận in . . . 47

2.4.3 Quản lý máy in . . . 48

2.4.4 Quản lý cài đặt in . . . 50

2.4.5 Xem lịch sử in 51
2.4.6 Xem báo cáo định kỳ 52

3 Architecture design **54**
3.1 Sử dụng kiến trúc phân lớp để thiết kế hệ thống HCMUT-SSPS 54
3.2 Mô tả cách trình bày giao diện người dùng 56
3.3 Mô tả cách lưu trữ dữ liệu 57
3.4 Mô tả cách truy cập vào các dịch vụ/API bên ngoài 59
3.5 Component Diagram cho các Module quan trọng 61
3.6 Deployment Diagram cho các module quan trọng 64
3.7 Mô tả Deployment Diagram 65

4 Implementation – Sprint 1 **66**
4.1 Github for version control 66
4.2 Chi tiết commit 66
4.3 Kiểm tra khả năng sử dụng 67
4.3.1 Tuyển dụng người tham gia kiểm thử 67
4.3.2 Xác định công việc 67
4.3.3 Xác định phương pháp kiểm tra 68
4.3.4 Tiến hành kiểm tra 68

1 Requirement elicitation

Task 1.1

Describe the domain context of a smart printing service for students at HCMUT. Who are relevant stakeholders? What are their current needs? In your opinion, what benefits HCMUT-SSPS will be for each stakeholder?

1.1 Bối cảnh và các bên liên quan trong dự án

In ấn là một phần quan trọng của cuộc sống học tập và sinh viên cần có quyền truy cập dễ dàng đến các thiết bị in ấn. Sinh viên có nhu cầu in ấn đa dạng, bao gồm bài tập, luận văn nghiên cứu, ghi chú lớp học, bài thuyết trình và nhiều công việc khác. Bối cảnh lĩnh vực của dịch vụ in thông minh cho sinh viên trên một khuôn viên trường học là một hệ thống chuyên biệt Student Smart Printing Service (HCMUT_SSPS) được thiết kế để đáp ứng nhu cầu in ấn của sinh viên trong trường Đại học Bách Khoa – Đại học Quốc gia Thành phố Hồ Chí Minh. Hệ thống được thiết kế phục vụ cho sinh viên trên khắp các tòa nhà ở các cơ sở của trường, cung cấp cho sinh viên một phương tiện để in tài liệu một cách tiện dụng và hiệu quả. Hệ thống tương tác với rất nhiều thực thể như học sinh, các máy in và Student Printing Service Officer. Từ mô tả hệ thống, ta có thể xác định các stakeholders và nhu cầu của họ đối với hệ thống như sau:

- Ban quản lý của trường đại học (University Administration)
  - Đảm nhiệm quản lý, theo dõi quá trình xây dựng phát triển hệ thống.
  - Hệ thống printing service hoạt động hiệu quả đáp ứng nhu cầu sinh viên.
- Sinh viên và nhân viên của trường (Student and Staff)
  - Một công cụ tiện lợi giúp in văn bản trong khuôn viên trường.
  - Có thể điều chỉnh các thuộc tính in để có thể in ấn theo mong muốn.
  - Thanh toán dễ dàng và có thể theo dõi lịch sử in ấn.
- Quản lý của SPSO (Administration of SPSO)
  - Xem được lịch sử in ấn của tất cả sinh viên và các máy in trong một khoảng thời gian xác định.
  - Có thể quản lý, thêm/bật/tắt máy in.
  - Quản lý được cấu hình hệ thống như thay đổi số lượng trang cho toàn thể sinh viên, loại file được upload lên hệ thống.
  - Xem lại báo cáo về sử dụng dịch vụ in vào cuối mỗi tháng, mỗi năm.
- Nhóm phát triển hệ thống (Development Team)
  - Phát triển thành công hệ thống in thông minh cho trường Đại học Bách Khoa.
  - Hệ thống in thông minh hoạt động ổn định đáp ứng nhu cầu của trường.
  - Có thể phát triển thêm các tính năng hệ thống trong tương lai.

- Quản trị viên của hệ thống thanh toán BKPay (Administration of payment system BKPay)
  - Kết nối từng tài khoản sinh viên với tài khoản thanh toán tương ứng của sinh viên trong BKPay.
  - Có thể thực hiện giao dịch ở hệ thống và thanh toán hóa đơn trong BKPay.
  - Quản lý được hóa đơn giao dịch của sinh viên trên hệ thống.
- Đội ngũ hỗ trợ kỹ thuật (Techinician Support)
  - Có quyền chỉnh sửa hệ thống khi hệ thống gặp lỗi nhằm khắc phục sự cố.
  - Đảm bảo hệ thống phát triển ổn đinh.
- Quản trị viên hệ thống xác thực tập trung (SSO Administration)
  - Sinh viên phải đăng nhập xác thực mới có thể sử dụng hệ thống.
  - Có thể truy cập vào hệ thống không cần đăng nhập khi đã đăng nhập trên các website khác của trường (Các trang MyBK).
  - Tài khoản trên MyBK cũng là định danh của sinh viên trên hệ thống in thông minh.
- Legal Department
  - Hệ thống thiết kế đảm bảo tuân thủ các quy định của pháp luật.

Hệ thống HCMUT-SSPS sẽ mang lại nhiều lợi ích cho mỗi stakeholder. Đối với sinh viên, hệ thống cung cấp các dịch vụ thuận tiện và thích hợp để in tài liệu, giảm thiểu thời gian của sinh viên không phải tìm kiếm máy in ở bên ngoài, cũng như cung cấp cách thức thanh toán thuận tiện thông qua BKPay. Hệ thống cũng đảm bảo quyền riêng tư của sinh viên thông qua xác thực người dùng đăng nhập vào tài khoản.

Đối với SPSO, hệ thống giúp các nhân viên quản lý máy in và cung cấp các công cụ theo dõi được lịch sử in ấn. Điều này giúp cho nhân viên biết được xu hướng và tần suất sử dụng máy in, giám sát để có thể bảo trì. Thông qua việc điều chỉnh cấu hình, SPSO có thể giúp cho hệ thống thân thiện hơn để sinh viên có thể sử dung, đáp ứng nhu cầu sinh viên.

Task 1.2

Describe all functional and non-functional requirements that can be inferred from the project description.

1.2 Các yêu cầu về functional và non-functional trong dự án

1.2.1 Functional Requirements

Đối với sinh viên:

1. Phải có chế độ sáng (Light) và tối (Dark) cho sinh viên lựa chọn.
2. Có chức năng nhấp chuột vào từng máy in hiện ra form để thực hiện yêu cầu in ấn tại máy in đó.
3. Sinh viên có thể upload tập tin cần được in ấn lên hệ thống.
4. Sinh viên được chọn máy in để in tài liệu.
5. Sinh viên có thể điều chỉnh các thuộc tính in như kích cỡ giấy, trang nào trong tập tin cần in, in hai mặt hay một mặt, số bản sao, ...
6. Sinh viên có thể xem lịch sử in trong một khoảng thời gian với một tổng hợp về số trang đã in cho mỗi kích cỡ trang.
7. Sinh viên có thể mua thêm giấy in sử dụng mục Buy Printing Pages của hệ thống và thanh toán online thông qua BKPay system.
8. Hệ thống chỉ cho phép sinh viên in nếu số trang in không vượt quá số trang được phép in hiện có trong tài khoản sinh viên.
9. Khi in ấn, nếu máy in không đủ giấy để in một tài liệu nào đấy của sinh viên, tuỳ chọn in trên máy in đó sẽ ẩn đi, sinh viên chỉ có thể thấy những lựa chọn máy in có đủ giấy để in cho sinh viên.
10. Hệ thống cần có chức năng tự động cộng số giấy mặc định vào số tờ giấy còn lại cho mỗi sinh viên trong một học kì mỗi khi đến ngày cập nhật được văn phòng quy định.
11. Hệ thống cần lưu lại lịch sử in của tất cả sinh viên bao gồm student ID, printer ID, file name, printing start and end time, number of pages for each page size.
12. Hệ thống có chức năng đăng nhập xác thực định danh người dùng bằng dịch vụ xác thực HCMUT_SSO trước khi cho phép người dùng sử dụng hệ thống.
13. Có tài khoản cá nhân trên hệ thống, có thể đăng nhập, đăng xuất.
14. Các thao tác quan trọng như in ấn, đăng nhập, đăng xuất, sửa đổi các thông số in phải có thông báo hoặc hiệu ứng thao tác thành công.
15. Hệ thống có chức năng hiển thị giới hạn truy cập thông tin cho sinh viên, sinh viên chỉ được xem file do mình upload lên, không được xem file của sinh viên khác.

Đối với Student Printing Service Officer (SPSO):

1. SPSO có khả năng quản lý các máy in thông qua các tác vụ như thêm, khởi động (enable), tắt (disable) một máy in nào đó.
2. SPSO có thêm xem lịch sử in (log) của tất cả sinh viên trong một khoảng thời gian nào đó và lịch sử in của tất cả máy in.
3. SPSO có khả năng quản lý cấu hình khác như thay đổi số trang cung cấp mỗi kỳ cho sinh viên, ngày hệ thống cung cấp số trang mặt định cho tất cả sinh viên, loại file được phép upload lên hệ thống.
4. SPSO có thể xem các báo cáo được tạo tự động vào cuối mỗi tháng hoặc cuối năm.
5. Hệ thống cần có chức năng kiểm soát tình trạng máy in: Khi máy in gặp vấn đề như kẹt giấy, hết giấy, ... máy in sẽ tự động gửi thông báo dữ liệu cho SPSO.
6. Hệ thống phải có chức năng sao lưu định kỳ hàng ngày và chức năng phục hồi dữ liệu trong trường hợp sự cố xảy ra.
7. Hệ thống có chức năng phát hiện truy cập bất thường và thông báo tới SPSO.
8. Hệ thống có chức năng hiển thị giới hạn truy cập thông tin đối với SPSO, SPSO chỉ được xem tên file và thông tin file như kích thước file, số trang in của sinh viên chứ không xem được nội dung in.

1.2.2 Non-Functional Requirements

- **Yêu cầu về giao diện**
  1. Giao diện đẹp, hiện đại và thân thiện với sinh viên.
  2. Phong chữ to, rõ ràng, dễ đọc và có độ tương phản cao.
  3. Các ô chức năng hay các hộp thoại nội dung được bố trí hợp lý.
  4. Màu sắc chủ đạo là màu lạnh, chọn màu tương phản làm nổi bật thông báo và nhiệm vụ.
  5. Có thể hỗ trợ đa ngôn ngữ; ngôn ngữ hiện tại là Tiếng Việt, trong tương lai có thể nâng cấp thêm nhiều ngôn ngữ khác.
  6. Có khả năng thích ứng với nhiều kích cỡ màn hình khác nhau (desktop, mobile, tablet).
  7. Những thông tin về các máy in đang hoạt động ở các tòa được hệ thống hiển thị chỉ trên một khung hình (không cần phải cuộn để xem).
- **Yêu cầu về hiệu năng, hiệu suất**
  1. Hệ thống phải có khả năng kiểm soát dữ liệu thời gian thực từ ít nhất 1000 sinh viên cùng lúc.
  2. Có khả năng cho phép hoạt động tốt khi có 1000 sinh viên truy cập đồng thời.
  3. Máy in phải hoạt động tốt và năng suất cao, có khả năng in tối thiểu 5000 tờ một ngày.
  4. Thời gian tải các trang web dưới 3 giây.
  5. Tốc độ xử lý các thao tác dưới 2 giây.
  6. Phản hồi hệ thống trong thời gian thực, độ trễ dưới 1 giây.

7. Do số lượng file sinh viên tải lên để in quá lớn, đòi hỏi hệ thống phải tối ưu dung lượng cho các file của sinh viên tải lên, mỗi file sinh viên tải lên không được quá 30MB.
8. Phản hồi từ các hệ thống ngoài như hệ thống xác thực HCMUT_SSO hay hệ thống thanh toán trực tuyến BKPay khi thực hiện chức năng dưới 5 giây.

- **Yêu cầu về độ tin cậy**

1. Hoạt động ổn định trong thời gian từ 7g00 – 17g00 tất cả các ngày (trừ chủ nhật và các ngày lễ).
2. Các máy in phải phục vụ được tối thiểu trong 12h liên tục mỗi ngày.
3. Xác suất hệ thống bị sập dưới 3%. Thời gian phục hồi nếu bị sập dưới 90 giây.
4. Khi máy in gặp vấn đề như kẹt giấy, hết giấy, ... việc máy in tự động gửi thông báo dữ liệu cho SPSO phải trong vòng tối đa 10 giây; đội ngũ hỗ trợ vấn đề chậm trễ không quá 5 phút.
5. Thông tin dữ liệu về trạng thái máy in, tài khoản sinh viên mỗi 15 phút với độ khả dụng ít nhất 95% thời gian vận hành.
6. Thông báo mới được cập nhật không quá 10 giây kể từ khi có sự thay đổi.
7. Chức năng sao lưu và phục hồi dữ liệu đảm bảo hoạt động ổn định và chính xác trong 99% thời gian.
8. Chức năng tổng hợp và đưa ra báo cáo phải chính xác theo lịch và chậm trễ tối đa 2 ngày cho hệ thống.
9. Đường truyền liên kết server với máy in phải đáng tin cậy; các hệ thống bên ngoài có liên kết phải đáng tin cậy.

- **Yêu cầu về tính bảo mật và riêng tư**

1. Chỉ có các tài khoản được cung cấp sẵn mới có khả năng đăng nhập (không có chức năng đăng ký).
2. Khi phát hiện được truy cập bất thường, hệ thống cần thông báo tới SPSO trong vòng dưới 3 giây; hệ thống xử lý vấn đề chậm trễ không quá 5 phút.
3. Đảm bảo bảo mật cao giữa người dùng và server, trong nội bộ server và giữa server với máy in.
4. Đảm bảo bảo mật cao trong quá trình liên kết phương thức thanh toán trực tuyến và xác thực người dùng.
5. Đảm bảo giới hạn thông tin có thể truy cập cho mỗi đối tượng sử dụng; đảm bảo bảo mật thông tin cho các đối tượng sử dụng.

- **Yêu cầu về tính hợp pháp**

1. Phù hợp với pháp luật hiện hành.

- **Yêu cầu về tổ chức hệ thống**

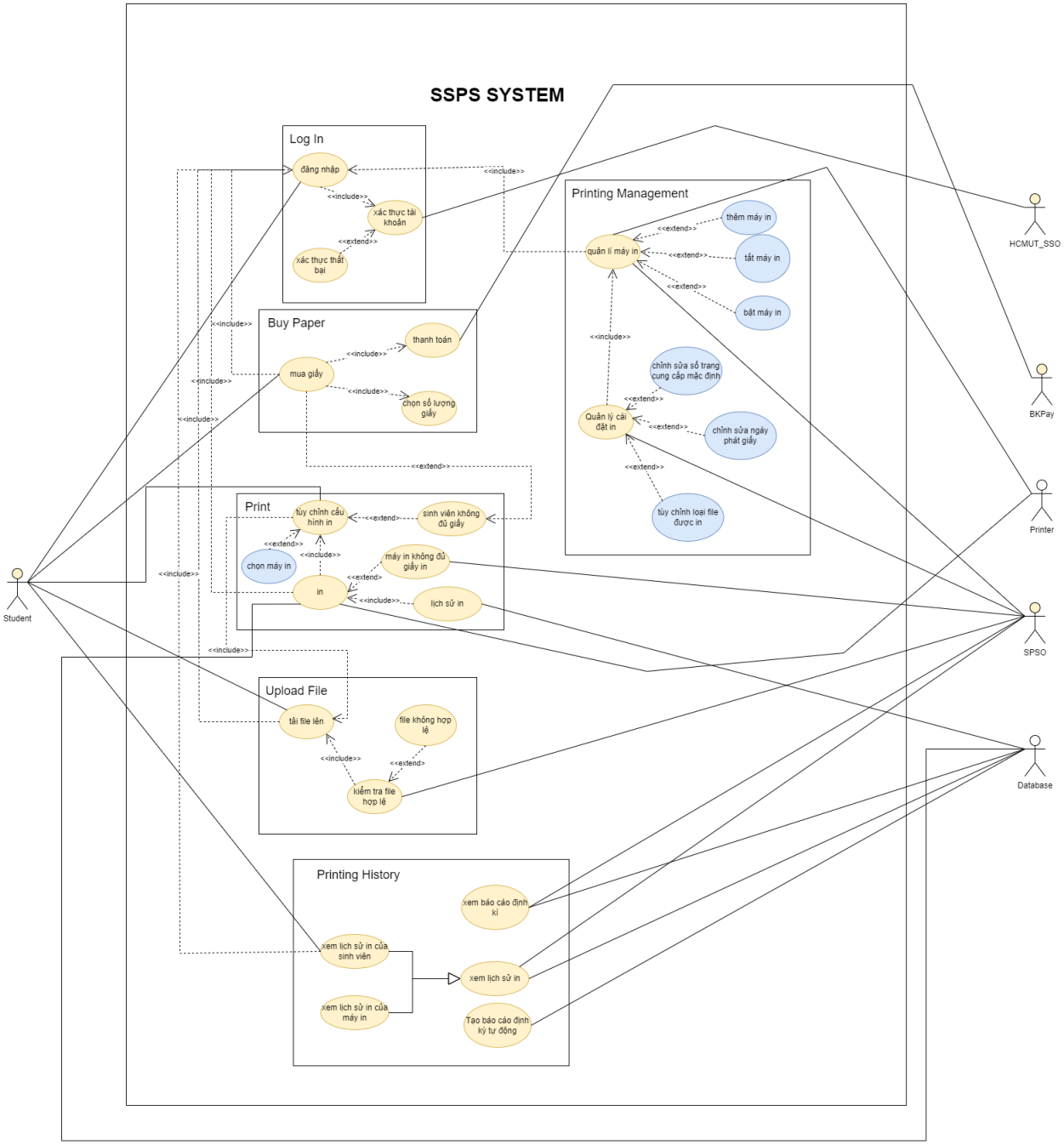
1. Các máy in, servers được bố trí ở các vị trí hợp lý, các máy in không cách nhau xa quá 100m và không gần quá 50m; tối thiểu ở mỗi toà học lý thuyết phải có một máy in.
2. Sử dụng không quá 2GB RAM trong quá trình hệ thống chạy.
3. Có khả năng mở rộng và phát triển trong tương lai.

Task 1.3

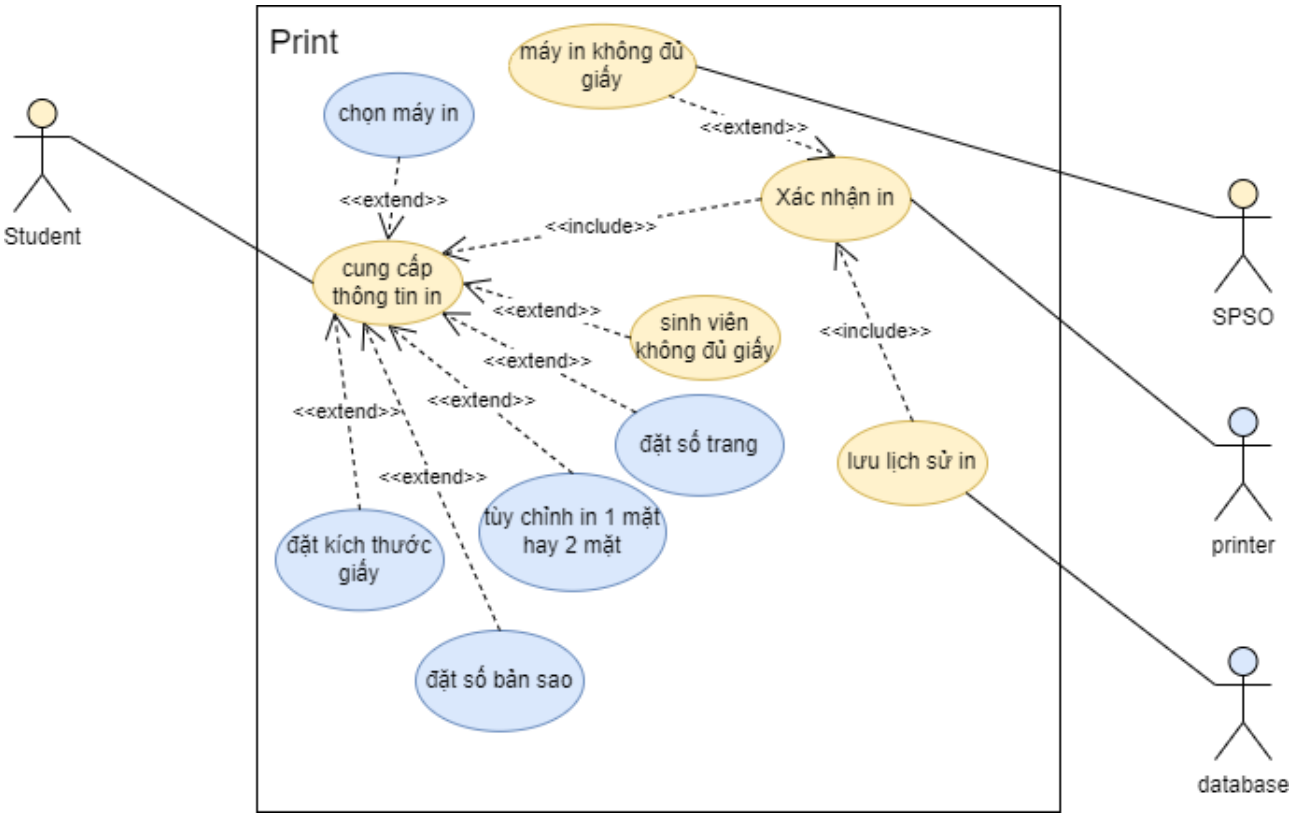
Draw a use-case diagram for the whole system.Choose at least one important module and draw its use-case diagram, as well as describe the use-cases using a table format.

1.3 Use-Case Scenario

1.3.1 Biểu đồ Use-case của toàn hệ thống SSPS



1.3.2 Đặc tả Use-case "Print"



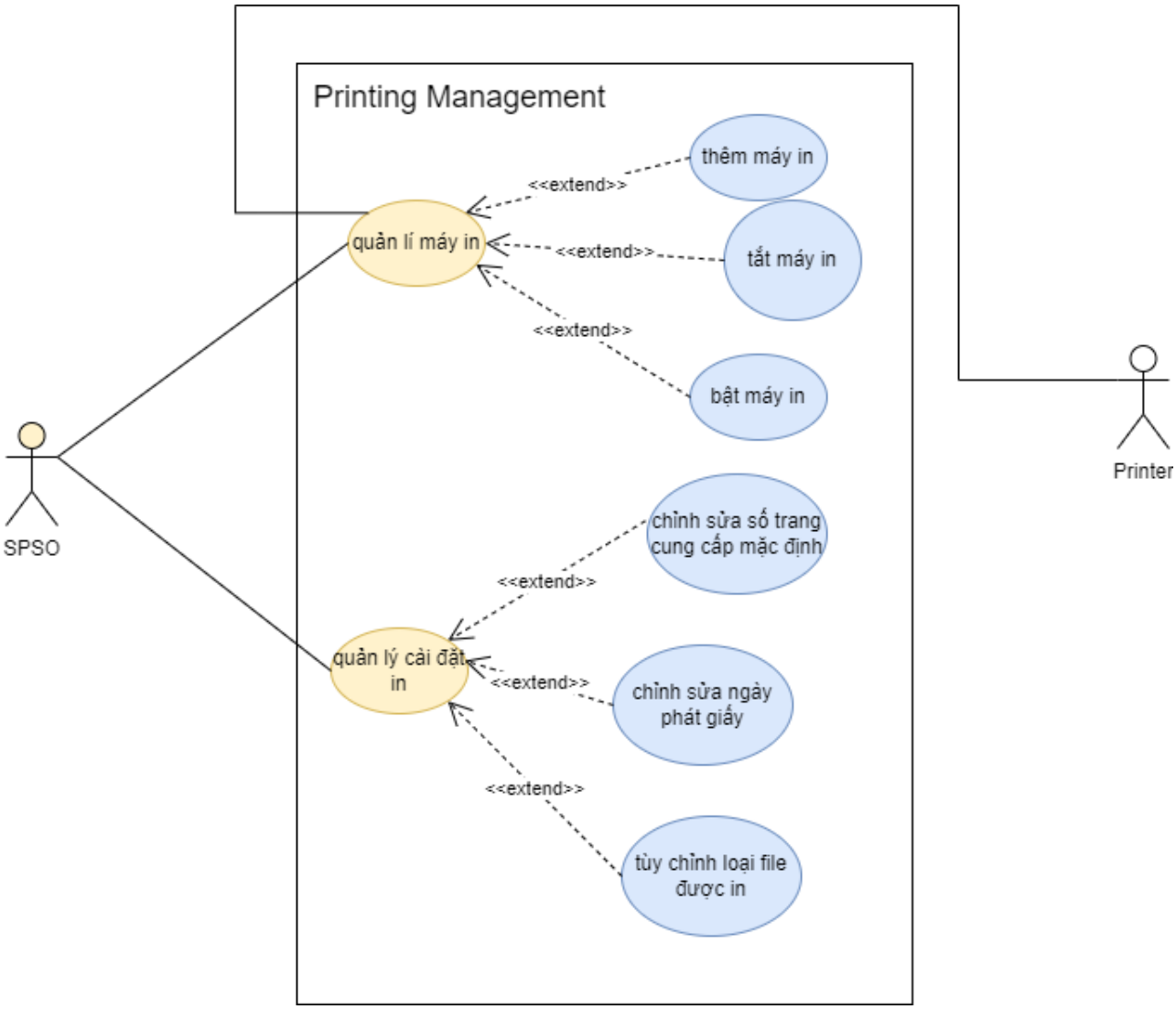
Use-case name	Cung cấp thông tin in
Description	Dùng để xác định chính xác nơi in và các thông tin mà tài liệu sắp được in ra cần có bao gồm cỡ giấy, bao nhiêu tờ, ...
Trigger	Người sử dụng nhấn vào nút “Chỉnh sửa thông tin in”
Actor	Student
Pre-conditions	- Hệ thống HCMUT-SSPS phải đang hoạt động ổn định (không trong trạng thái bảo trì) - Tình trạng Internet ổn định - Hệ thống đã xác thực người dùng - Tài liệu đã được upload lên hệ thống thành công
Post-conditions	Những thông tin cần thiết cho quá trình in đã được nhập đầy đủ và hoàn chỉnh, sẵn sàng cho việc in ấn, hệ thống chuyển sang use-case "Xác nhận in" hoặc use-case "Mua thêm giấy in" nếu người sử dụng chọn lựa chọn mua thêm giấy, hoặc quay về trang chủ nếu có điều kiện không thỏa.
Normal Flow	1. Hệ thống hiển thị một bảng dạng form trên màn hình với các tùy chọn sau: - Kích thước giấy và khổ giấy (ngang hoặc dọc) - Lựa chọn in hai mặt hoặc một mặt - Ô nhập số trang cần in - Ô nhập số bản sao cần in 2. Người sử dụng hoàn thiện bảng form 3. Hệ thống kiểm tra các giá trị được nhập so với các giá trị cho phép từ SPSO. 4. Hệ thống hiện danh sách máy in để sinh viên chọn máy in. Hệ thống chỉ hiển thị các máy in không bị lỗi và có số lượng giấy còn lại đủ đáp ứng yêu cầu in 5. Hệ thống xác nhận đã nhập đầy đủ thông tin và chuyển sang use-case "Xác nhận in"
Alternative Flow	None
Exception Flow	*Exception 1: Tại bước 3* 3a. Nếu các giá trị được nhập vào không phù hợp với các giá trị được cho phép từ SPSO thì: *3a.1.* Hệ thống hiển thị thông báo lỗi *3a.2.* Người dùng nhấn vào nút OK và chuyển về Normal Flow 1 *Exception 2: Tại bước 3* 3b. Nếu tổng số trang cần in vượt quá số trang còn lại mà người dùng hiện có thì: *3b.1.* Hệ thống hiển thị lên màn hình số trang còn lại và thông báo cho người sử dụng chọn nút "Thoát" để mua thêm giấy, hoặc "Quay lại" để quay về Normal Flow 1 sửa số trang in. *3b.2.* Có 2 trường hợp xảy ra: - Nếu người sử dụng chọn chỉnh sửa số trang cần in để nằm trong số trang còn lại thì chuyển về Normal Flow 1 - Nếu người sử dụng chọn thoát để mua thêm giấy thì quay về trang chủ giấy in" *Exception 3: Tại bước 4* *4a.* Nếu các máy in đều bị lỗi *4a.1.* Hệ thống hiển thị lên màn hình thông báo các máy in đều đang bị lỗi *4a.2.* Người dùng nhấn vào nút "OK" thì quay về trang chủ *Exception 4: Tại bước 4* *4b.* Nếu không còn lựa chọn máy in nào đủ giấy để đáp ứng yêu cầu in *4b.1.* Hệ thống hiển thị lên màn hình thông báo không còn máy in nào còn đủ giấy để in, nhấn nút Chỉnh sửa số trang in để quay về Normal Flow Bước 1. *4b.2.* Nếu người sử dụng không chọn chỉnh sửa lại và chọn nhấn nút "Thoát" để quay về trang chủ

Table 1: ***Use-case "Cung cấp thông tin in"***

Use-case name	Xác nhận in
Description	Xác nhận in và gửi thông tin in về lại hệ thống để thực hiện việc in ấn
Trigger	Người sử dụng nhấn vào nút “In”
Actor	Student, Printer, database, SPSO
Pre-conditions	- Hệ thống HCMUT-SSPS phải đang hoạt động ổn định (không trong trạng thái bảo trì) - Tình trạng Internet ổn định - Hệ thống đã xác thực người dùng - Tài liệu đã được upload thành công và các thông tin in đã được xác nhận thành công - Đường truyền từ hệ thống đến máy in ổn định - Các máy in phải đang hoạt động - Database phải được kết nối với hệ thống và hoạt động ổn định
Post-conditions	Xác nhận in và tài liệu được in thành công khi có thông báo thành công, lịch sử in được lưu vào hệ thống và quay về trang chủ hoặc không có công việc nào được thực hiện khi có thông báo thất bại (lỗi), hệ thống chuyển về bước 4 trong Normal Flow của use-case "Cung cấp thông tin in"
Normal Flow	1. Thông tin in được gửi về hàng đợi trên máy in 2. Kiểm tra các điều kiện in trên máy in 3. Hiển thị thông báo "Thành công" khi xác nhận thành công. Sinh viên nhấn nút OK để có thể thoát khỏi thông báo và trở về trang chủ 4. Máy in thực hiện in tài liệu và cập nhật tài khoản giấy của sinh viên và số giấy còn lại của máy in 5. Thông tin về sự kiện in được lưu vào database
Alternative Flow	None
Exception Flow	*Exception 1: Tại bước 2* 2a. Nếu gặp lỗi liên quan đến máy in thì hiển thị lên màn hình thông báo lỗi với yêu cầu người dùng chọn lại máy in khác, chuyển trạng thái máy in đã chọn sang bị lỗi và gửi thông báo tự động tới SPSO 2b. Sinh viên nhấn nút OK trên thông báo. Chuyển về bước 4 trong Normal Flow của use-case "Cung cấp thông tin in"

Table 2: ***Use-case "Xác nhận in"***

1.3.3 Đặc tả Use-case "Printing Management"



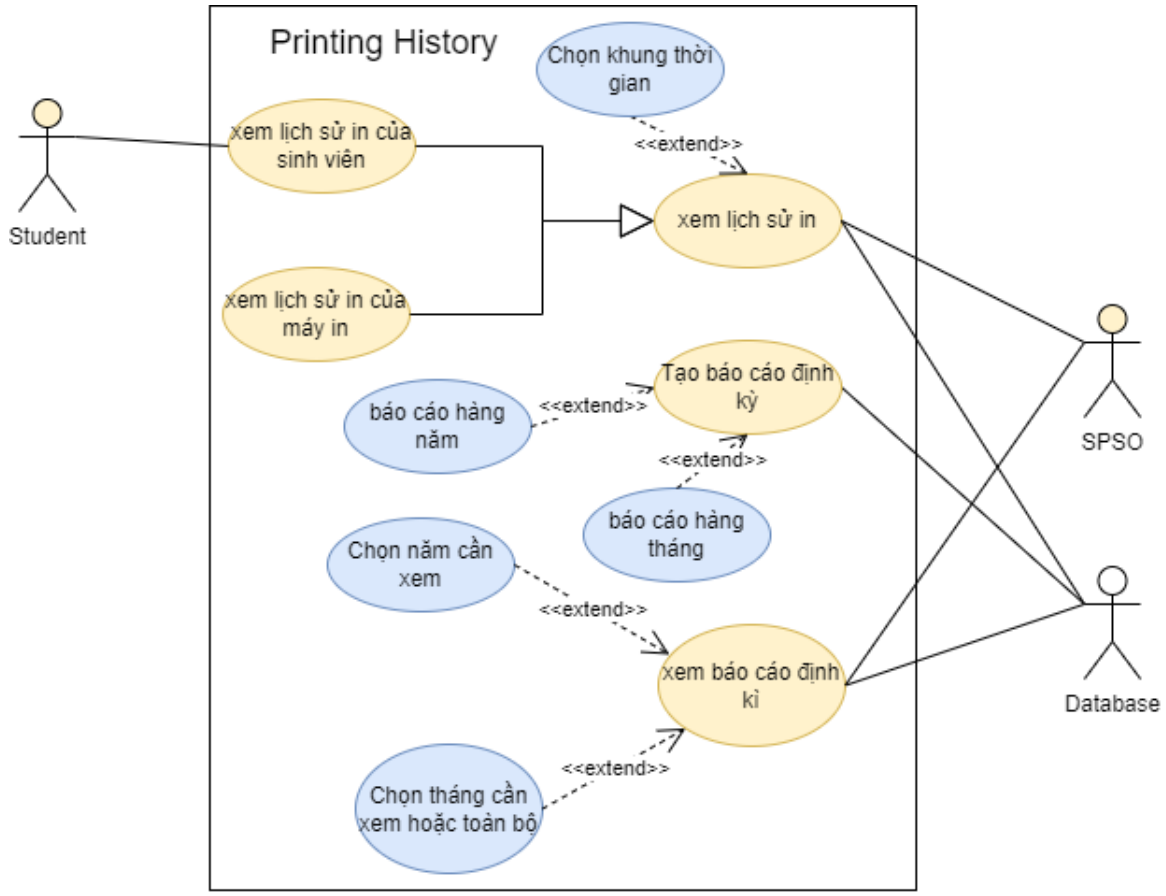
Use-case name	Quản lý máy in
Description	SPSO quản lý các máy in thông qua các tác vụ như thêm (add), xóa (remove), khởi động (enable), tắt (disable) một máy in nào đó
Trigger	Người sử dụng nhấn vào nút "Quản lý máy in"
Actor	SPSO, Printer
Pre-conditions	- Người dùng có quyền điều khiển của SPSO - Hệ thống đang hoạt động ổn định - Các thiết bị có kết nối mạng - Tình trạng Internet ổn định - Các máy in đều được kết nối với hệ thống - Thực hiện xác thực người dùng là SPSO
Post-conditions	Máy in được điều khiển khi có thông báo thành công hoặc không bị điều khiển khi có thông báo thất bại (lỗi). Sau đó có thể quay về trang chủ.
Normal Flow	1. Hiển thị lên màn hình bảng chứa tất cả máy in đang có trong hệ thống; người dùng chọn máy in cần thực hiện điều khiển 2. Hiển thị bảng chứa các lựa chọn điều khiển có thể có với máy in; người dùng thực hiện điều khiển máy in được chọn 3. Người dùng xác nhận các thay đổi 4. Hệ thống kiểm tra máy in có phù hợp để thực hiện các thay đổi hay không 5. Hệ thống thực hiện các thay đổi 6. Hiển thị thông báo "Thành công" và lưu các thay đổi nếu thực hiện thành công; thông báo "Lỗi" kèm nội dung lỗi nếu thực hiện không thành công 7. Người dùng chọn nút OK trên thông báo để quay về trang chủ
Alternative Flow	*Alternative 1: Tại bước 1* 1a. Người dùng chọn chức năng "Thêm máy in" 1b. Hiển thị bảng chứa các máy in đã được kết nối nhưng chưa được thêm vào hệ thống 1c. Người dùng chọn máy in muốn được thêm vào hệ thống *Tiếp tục bước 3 ở trong Normal Flow*
Exception	*Exception 1: Tại bước 1* 1d. Nếu không có máy in nào trong hệ thống thì chuyển về Alternative Flow 1a *Exception 1: Tại bước 1b* 1e. Nếu không có máy in nào được kết nối, hệ thống hiển thị dòng chữ "Không có máy in nào được kết nối" ở vùng hiển thị các máy in. Hiển thị nút Quay lại. Người dùng click vào thì quay lại Normal Flow 1. *Exception 2: Tại bước 4* 4a. Nếu thực hiện điều khiển lựa chọn không thể được áp dụng thì hệ thống hiển thị thông báo lỗi và lý do không áp dụng được. 4b. Người dùng nhấn nút OK trên bảng thông báo và chuyển về Normal Flow 1.

Table 3: ***Use-case "Quản lý máy in"***

Use-case name	Quản lý cài đặt in
Description	SPSO quản lý cấu hình khác như thay đổi số trang cung cấp mỗi kỳ cho sinh viên, ngày hệ thống cung cấp số trang mặc định cho tất cả sinh viên, loại file được phép upload lên hệ thống
Trigger	Người sử dụng nhấn vào nút "Quản lý cài đặt in" trên trang chủ
Actor	SPSO
Pre-conditions	- Người dùng có quyền điều khiển của SPSO - Hệ thống đang hoạt động ổn định - Các thiết bị có kết nối mạng - Tình trạng Internet ổn định - Hệ thống đã xác thực người dùng - Các máy in đều được kết nối với hệ thống
Post-conditions	Các thay đổi được áp dụng khi có thông báo thành công hoặc không được áp dụng khi có thông báo thất bại (lỗi). Sau đó có thể quay về trang chủ
Normal Flow	1. Hiển thị lên màn hình bảng tất cả thông tin có thể tùy chỉnh được của hệ thống 2. Người sử dụng thực hiện tùy chỉnh các thông tin và xác nhận các thay đổi tùy chỉnh 3. Người dùng xác nhận các thay đổi 4. Hệ thống kiểm tra các tùy chỉnh có thể được áp dụng hay không 5. Hệ thống thực hiện các tùy chỉnh 6. Hiển thị thông báo "Thành công" và lưu thông tin tùy chỉnh vào hệ thống nếu thực hiện thành công; thông báo "Lỗi" kèm nội dung lỗi nếu thực hiện không thành công 7. Người dùng nhấn nút OK trên thông báo và quay về trang chủ.
Alternative Flow	None
Exception	*Exception 1: Tại bước 4* 4a. Nếu các giá trị tùy chỉnh không thể được áp dụng thì hệ thống hiển thị thông báo lỗi cụ thể vì sao không áp dụng được. 4b. Người dùng nhấn nút OK trên bảng thông báo và chuyển về Normal Flow 1

Table 4: ***Use-case "Quản lý cài đặt in"***

1.3.4 Đặc tả Use-case "Printing History"



Use-case name	Xem báo cáo định kỳ
Description	SPSO xem các báo cáo được tạo tự động vào cuối mỗi tháng hoặc cuối năm
Trigger	Người sử dụng nhấn vào nút “Hiển thị báo cáo định kỳ”
Actor	SPSO, Database
Pre-conditions	- Xác thực người dùng có quyền điều khiển của SPSO - Hệ thống đang hoạt động ổn định - Tình trạng Internet ổn định - Database phải được kết nối với hệ thống và hoạt động ổn định
Post-conditions	Hiển thị thành công báo cáo định kỳ của hệ thống theo yêu cầu của người dùng
Normal Flow	1. Hiển thị bảng thông tin gồm mục "Năm cần xem" và "Tháng cần xem" 2. Người dùng chọn "Năm cần xem" 3. Ở mục "Tháng cần xem", nếu muốn xem báo cáo một tháng cụ thể, người dùng chọn một tháng cần xem. Nếu muốn xem báo cáo cả năm đó, người dùng chọn lựa chọn "Toàn bộ". 4. Hệ thống truy xuất dữ liệu từ database lấy tương ứng 5. Hiển thị ra báo cáo theo yêu cầu của người dùng
Alternative Flow	None
Exception	*Exception 1: Tại bước 3* 3a. Nếu database không đủ dữ liệu để tạo báo cáo hoặc không tồn tại dữ liệu, hệ thống hiển thị thông báo "Dữ liệu không tồn tại để tạo báo cáo" 3b. Người dùng nhấn nút OK trên bảng thông báo và chuyển về Normal Flow 1

Table 5: ***Use-case "Xem báo cáo định kỳ"***

Use-case name	Xem lịch sử in
Description	SPSO xem lịch sử in của máy in hoặc sinh viên và sinh viên xem lịch sử in
Trigger	Người sử dụng nhấn vào nút “Xem lịch sử in”
Actor	SPSO, Student, Database
Pre-conditions	- Xác thực người dùng có quyền điều khiển của SPSO hoặc sinh viên - Hệ thống đang hoạt động ổn định - Tình trạng Internet ổn định - Database phải được kết nối với hệ thống và hoạt động ổn định
Post-conditions	Hiển thị thành công lịch sử in theo yêu cầu của người dùng
Normal Flow	1. Nếu người dùng là SPSO, hiển thị bảng form gồm các mục "Sinh viên", "Máy in", "Nhập ID cho đối tượng" và "Khung thời gian" 2. Chọn "Sinh viên" nếu xem lịch sử in sinh viên. Chọn "Máy in" nếu xem lịch sử in máy in 3. Ở mục "Nhập ID đối tượng", Nhập ID của sinh viên hoặc máy in cần xem 4. Ở mục "Khung thời gian", chọn khoảng thời gian cần xem. 5. Hệ thống truy xuất dữ liệu từ database lấy lịch sử in tương ứng. 6. Hiển thị ra lịch sử theo yêu cầu của người dùng
Alternative Flow	Alternative Flow 1: Tại bước 1 1a. Nếu người dùng là sinh viên, không hiển thị mục "Máy in", "Sinh viên" "Nhập ID đối tượng" 1b. Ở mục "Khung thời gian", chọn khoảng thời gian cần xem *Tiếp tục bước 5 trong Normal Flow*
Exception	*Exception 1: Tại bước 4* 4a. Nếu database không đủ dữ liệu để tạo báo cáo hoặc không tồn tại dữ liệu, hệ thống hiển thị dòng chữ "Dữ liệu không tồn tại để tạo báo cáo" 4b. Người dùng nhấn nút OK trên bảng thông báo và chuyển về Normal Flow 1 đối với SPSO hoặc Alternative Flow 1a đối với sinh viên

Table 6: ***Use-case "Xem lịch sử in"***

Use-case name	Tạo báo cáo tự động định kỳ
Description	Các báo cáo được tạo tự động bởi hệ thống một cách định kỳ vào cuối mỗi tháng hoặc cuối các năm
Trigger	Thời điểm cuối mỗi tháng và cuối các năm
Actor	Database
Pre-condition	- Hệ thống đang hoạt động ổn định - Tình trạng Internet ổn định - Database phải được kết nối với hệ thống và hoạt động ổn định
Post-condition	(Các) báo cáo định kỳ của tháng (năm) được tạo và lưu vào Database
Normal Flow	1. Hệ thống trích xuất dữ liệu của tháng vào thời điểm cuối tháng 2. Hệ thống thực hiện tổng hợp các dữ liệu được trích xuất thành báo cáo định kỳ mỗi tháng 3. Hệ thống lưu báo cáo định kỳ vừa tạo vào Database
Alternative Flow	*Alternative 1: Tại bước 1* 1a. Ở thời điểm cuối năm, hệ thống trích xuất cả hai dữ liệu cho cả năm và cả tháng đó 1b. Hệ thống thực hiện tổng hợp cả hai báo cáo định kỳ tháng và năm từ dữ liệu đã được trích xuất *Tiếp tục bước 3 ở trong Normal Flow*
Exception	None

Table 7: ***Use-case "Tạo báo cáo tự động định kỳ"***

2 System modelling

Task 2.1

Draw activity diagrams to capture the business processes between systems and the stakeholders for the important module(s) chosen in Task 1.3.

2.1 Activity diagrams

2.1.1 Cung cấp thông tin in

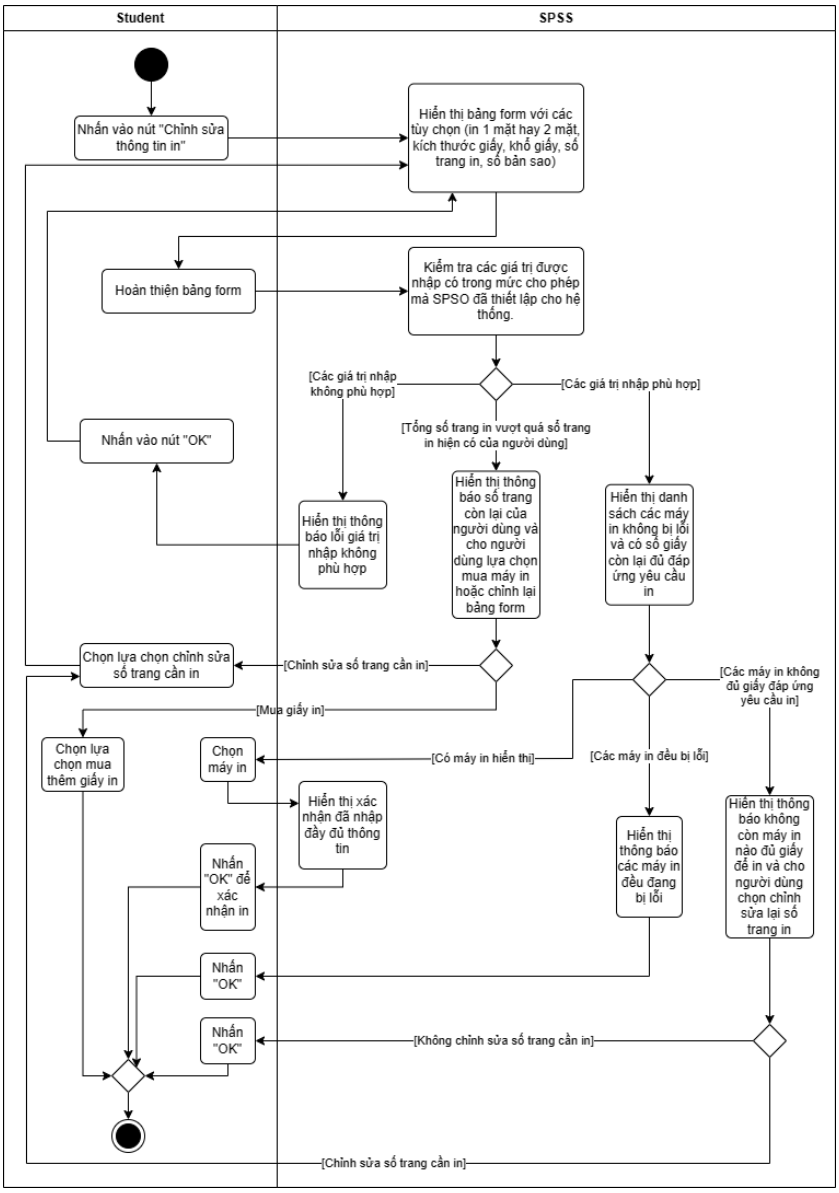


Figure 1: *Activity Diagram cung cấp thông tin in*

Xem chi tiết diagram tại đây

Mô tả các bước để thực hiện nhiệm vụ cung cấp thông tin in từ Activity Diagram:

1. Người dùng sau khi đăng nhập và tải lên tài liệu thành công nhấn vào nút "Chỉnh sửa thông tin in" để cung cấp các thông tin in như kích thước giấy, khổ giấy, số trang in, số bản sao và máy in muốn lựa chọn để in.

2. Sau khi nhấn vào nút "Chỉnh sửa thông tin in", hệ thống SSPS sẽ hiển thị bảng form gồm các thông tin in một mặt hay hai mặt, kích thước giấy, khổ giấy, số trang in, số bản sao, người dùng sẽ hoàn thiện bảng form này và xác nhận.

3. Hệ thống SSPS sẽ kiểm tra các giá trị được nhập trong bảng form có trong mức cho phép mà SPSO đã thiết lập cho hệ thống hay không. Tại đây có 3 trường hợp xảy ra:

   (a) Nếu các giá trị nhập không phù hợp thì hệ thống sẽ thông báo lỗi, sau đó người dùng nhấn nút "OK" thì hệ thống sẽ hiển thị lại bảng form gồm các thông tin in một mặt hay hai mặt, kích thước giấy, khổ giấy, số trang in, số bản sao để người dùng chọn lại.

   (b) Nếu tổng số trang in vượt quá số trang in hiện có của người dùng thì hệ thống sẽ hiển thị lên màn hình số trang in còn lại của người dùng và thông báo cho người dùng chọn mua thêm giấy hoặc chỉnh sửa số trang cần in, tại đây:

   - Nếu người dùng chọn mua thêm giấy in thì hệ thống sẽ chuyển sang trang mua thêm giấy in.
   - Nếu người dùng chọn lựa chọn chỉnh sửa số trang cần in thì hệ thống sẽ hiển thị lại bảng form gồm các thông tin in một mặt hay hai mặt, kích thước giấy, khổ giấy, số trang in, số bản sao để người dùng chọn lại.

   (c) Nếu các giá trị nhập phù hợp thì hệ thống sẽ hiển thị danh sách các máy in không bị lỗi và có số giấy còn lại đủ đáp ứng yêu cầu in, tại đây:

   - Nếu có máy in nào hiển thị thì người dùng sẽ chọn máy in, sau đó hệ thống hiển thị xác nhận đã nhập đầy đủ thông tin, người dùng nhấn nút "OK" trên thông báo và hệ thống sẽ chuyển sang trang xác nhận in.
   - Nếu các máy in đều bị lỗi thì hệ thống hiển thị thông báo các máy in đang bị lỗi, người dùng nhấn nút "OK" trên thông báo và hệ thống sẽ trở về trang chủ.
   - Nếu các máy in đều không đủ giấy để đáp ứng yêu cầu in thì hệ thống hiển thị thông báo không còn máy in nào đủ giấy và cho người dùng chọn chỉnh sửa số trang cần in hoặc thoát. Tại đây:
     - Nếu người dùng chọn lựa chọn chỉnh sửa số trang cần in thì hệ thống sẽ hiển thị lại bảng form gồm các thông tin in một mặt hay hai mặt, kích thước giấy, khổ giấy, số trang in, số bản sao để người dùng chọn lại.
     - Nếu người dùng không chọn chỉnh sửa số trang cần in thì người dùng chọn nút "OK" trên thông báo thì hệ thống sẽ trở về trang chủ.

2.1.2 Xác nhận in

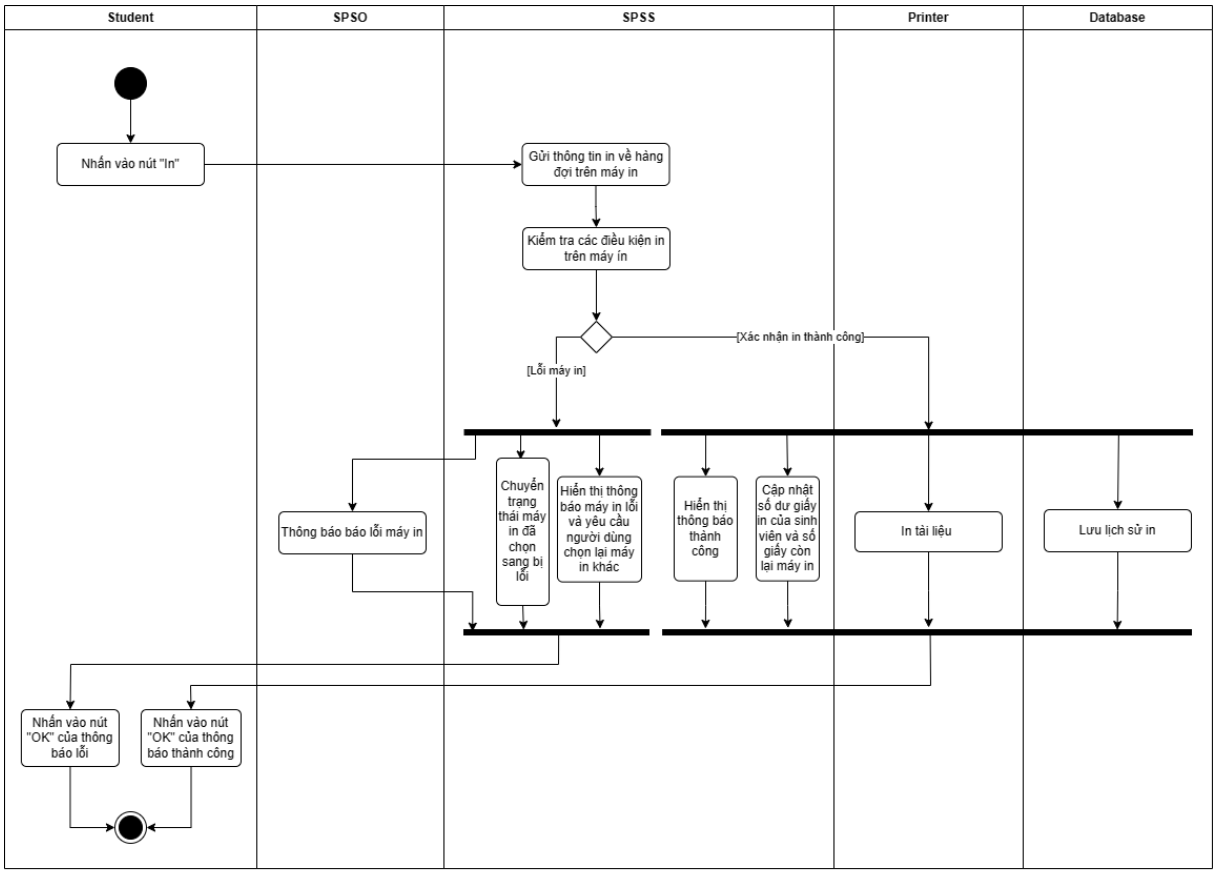


Figure 2: *Acivity Diagram xác nhận in*

Xem chi tiết diagram tại đây

Mô tả các bước để thực hiện nhiệm vụ xác nhận in từ Activity Diagram:

1. Người dùng sau khi đăng nhập và tải lên tài liệu thành công nhấn vào nút "In" để xác nhận in, in tài liệu và lưu lịch sử in.

2. Sau khi nhấn vào nút "In", hệ thống SSPS sẽ gửi thông tin về hàng đợi trên máy in, sau đó hệ thống sẽ kiểm tra các điều kiện in trên máy in. Tại đây:

   - Nếu máy in người dùng chọn bị lỗi thì đồng thời thực hiện 3 công việc: gửi thông báo máy in đó bị lỗi cho SPSO, chuyển trạng thái của máy in đã chọn sang bị lỗi và hiển thị thông báo máy in lỗi và yêu cầu người dùng chọn lại máy in khác. Sau đó người dùng nhấn vào nút "OK" trên thông báo lỗi để hệ thống hiện lại danh sách máy in để sinh viên chọn máy in.
   - Nếu xác nhận in thành công thì sẽ đồng thời thực hiện 4 công việc: hệ thống hiển thị thông báo thành công và cập nhập lại số dư giấy in của sinh viên và số giấy còn lại máy in, máy in đã chọn sẽ thực hiện in tài liệu, lịch sử in sẽ được lưu vào database. Sau đó người dùng nhấn vào nút "OK" trên thông báo thành công để thoát khỏi thông báo và trở về trang chủ.

2.1.3 Quản lý máy in

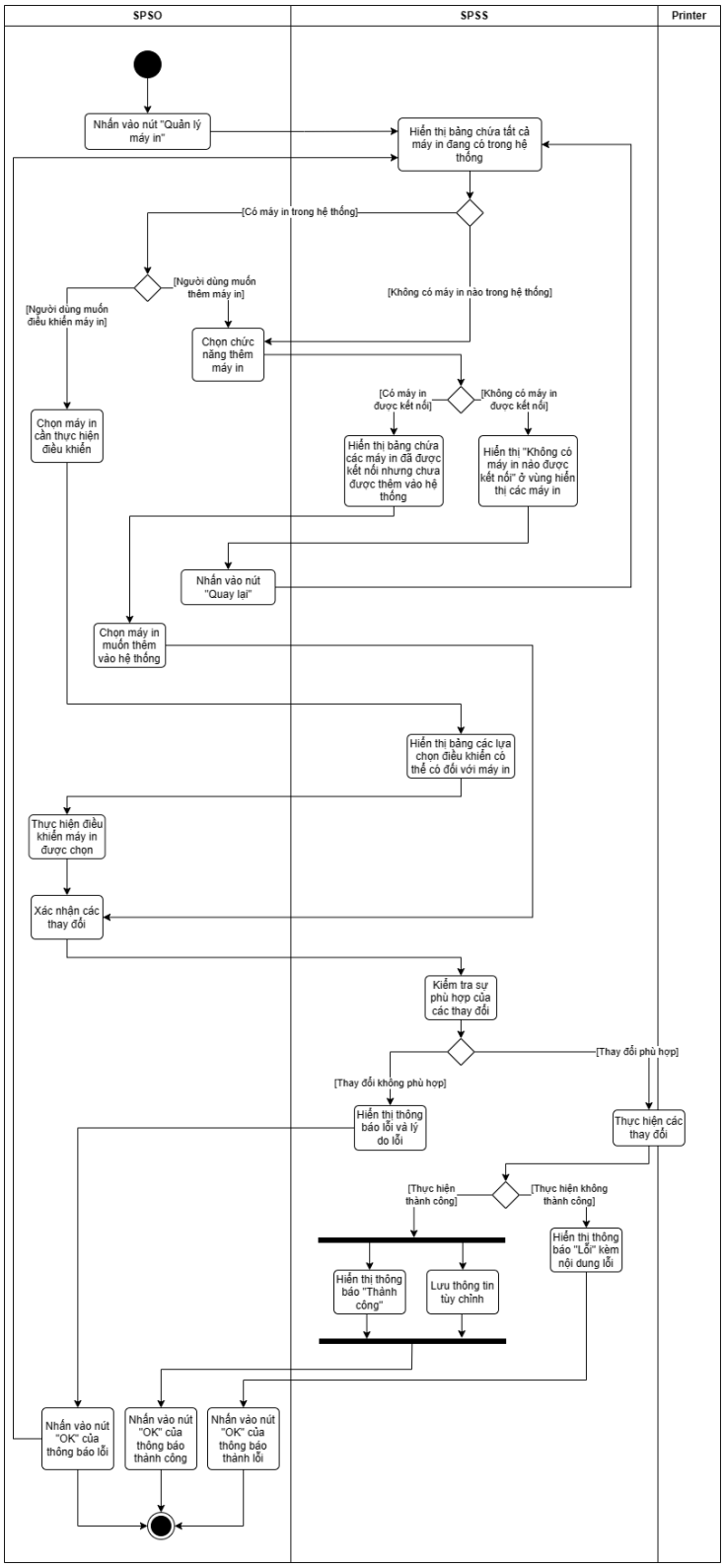


Figure 3: *Activity Diagram quản lý máy in*

Xem chi tiết diagram tại đây

Mô tả các bước để thực hiện nhiệm vụ quản lý máy in từ Activity Diagram:

1. SPSO sau khi đăng nhập thành công nhấn vào nút "Quản lý máy in" để quản lý các máy in thông qua các tác vụ như thêm (add), xóa (remove), khởi động (enable), tắt (disable) một máy in nào đó.

2. Sau khi nhấn vào nút "Quản lý máy in", hệ thống SSPS sẽ hiển thị bảng chứa tất cả các máy in đang có trong hệ thống, tại đây:

   - Nếu đang có máy in trong hệ thống và người dùng muốn điều khiển máy in thì người dùng chọn máy in cần thực hiện điều khiển. Hệ thống sẽ hiển thị bảng các lựa chọn điều khiển có thể có đối với máy in, người dùng thực hiện điều khiển máy in được chọn (thêm, xóa, khởi động, tắt máy in) và xác nhận các thay đổi.
   - Nếu đang có máy in trong hệ thống và người dùng muốn thêm máy in vào hệ thống hoặc đang không có máy in trong hệ thống thì người dùng chọn chức năng thêm máy in.
     - Nếu có máy in được kết nối thì hệ thống hiển thị bảng chứa các máy in đã được kết nối nhưng chưa được thêm vào hệ thống, người dùng chọn máy in muốn thêm vào hệ thống và xác nhận các thay đổi.
     - Nếu không có máy in nào được kết nối thì hệ thống hiển thị "Không có máy in nào được kết nối" ở vùng hiển thị các máy in, người dùng bấm vào nút "Quay lại", hệ thống sẽ hiển thị lại bảng chứa tất cả các máy in đang có trong hệ thống để người dùng thực hiện lại.
   - Sau đó, hệ thống sẽ kiểm tra sự phù hợp của các thay đổi, tại đây có 2 trường hợp xảy ra:
     - Nếu thay đổi không phù hợp thì hệ thống sẽ hiển thị thông báo lỗi và nội dung lỗi, sau đó người dùng ấn vào nút "OK" trên thông báo thì hệ thống sẽ hiển thị lại bảng chứa tất cả các máy in đang có trong hệ thống để người dùng thực hiện lại.
     - Nếu thay đổi phù hợp thì hệ thống sẽ thực hiện các thay đổi, tại đây:
       * Nếu thay đổi thực hiện thành công thì hệ thống đồng thời hiển thị thông báo thành công và lưu các thông tin tùy chỉnh.
       * Nếu thay đổi thực hiện không thành công thì hệ thống hiển thị thông báo lỗi kèm nội dung lỗi.

3. Người dùng nhấn vào nút "OK" trên thông báo để quay về trang chủ.

2.1.4 Quản lý cài đặt in

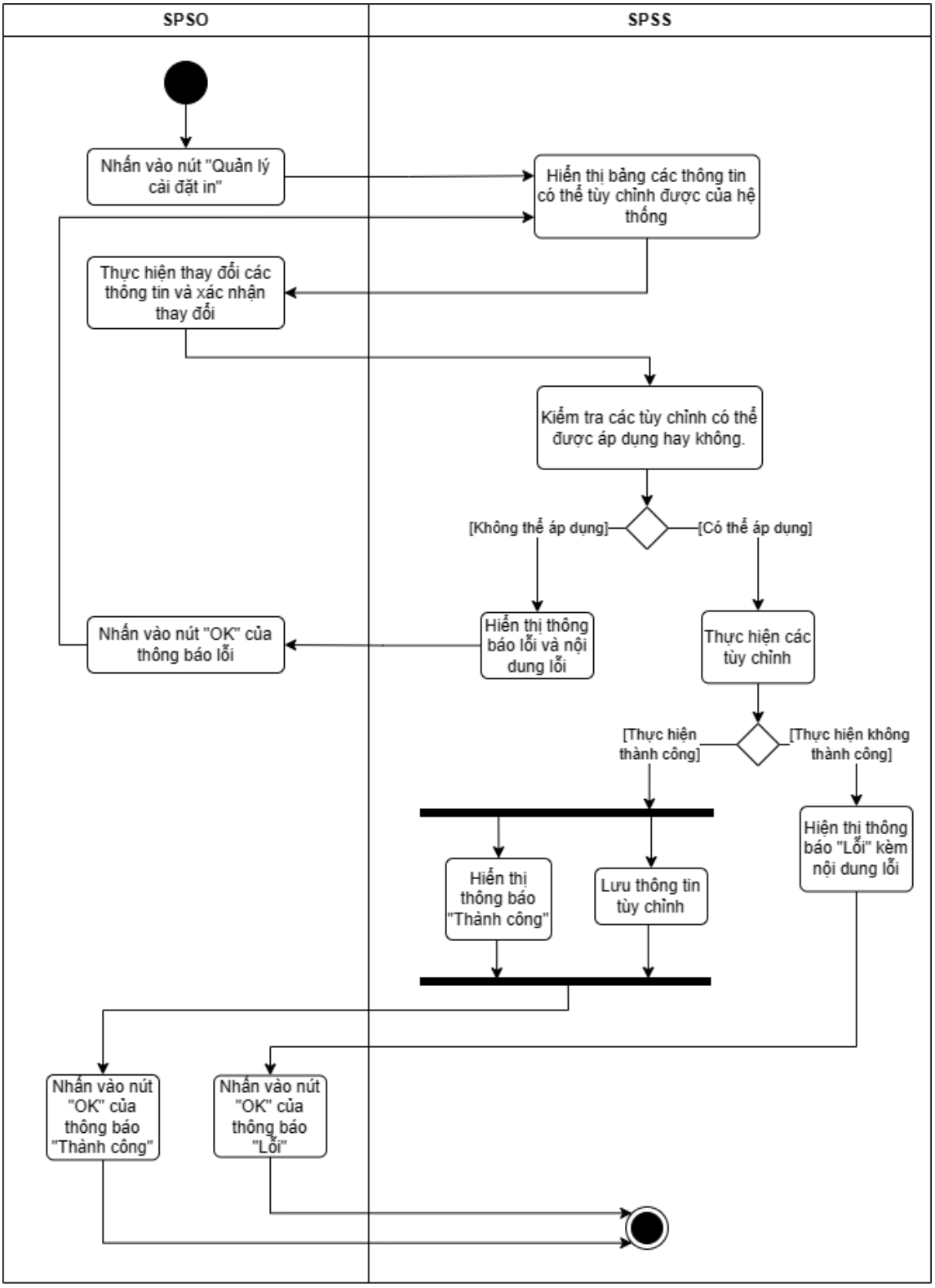


Figure 4: *Activity Diagram quản lý cài đặt in*

Xem chi tiết diagram tại đây

Mô tả các bước để thực hiện nhiệm vụ quản lý cài đặt in từ Activity Diagram:

1. SPSO sau khi đăng nhập thành công thì nhấn vào nút "Quản lý cài đặt in" để quản lý cấu hình như thay đổi số trang cung cấp mỗi kỳ cho sinh viên, ngày hệ thống cung cấp số trang mặc định cho tất cả sinh viên, loại file được phép upload lên hệ thống.

2. Sau khi nhấn vào nút "Quản lý cài đặt in", hệ thống SSPS sẽ hiển thị bảng các thông tin tùy chỉnh được của hệ thống, SPSO thực hiện thay đổi các thông tin và xác nhận các thay đổi tùy chỉnh.

3. Sau đó, hệ thống sẽ kiểm tra các tùy chỉnh có được áp dụng hay không, tại đây có 2 trường hợp xảy ra:

   - Nếu thay đổi không thể áp dụng được thì hệ thống sẽ hiển thị thông báo lỗi và nội dung lỗi, sau đó người dùng ấn vào nút "OK" trên thông báo thì hệ thống sẽ hiển thị lại bảng các thông tin tùy chỉnh được của hệ thống để người dùng tùy chọn lại.
   - Nếu thay đổi có thể áp dụng được thì hệ thống sẽ thực hiện thay đổi, tại đây:
     - Nếu thay đổi thực hiện thành công thì hệ thống đồng thời hiển thị thông báo thành công và lưu các thông tin tùy chỉnh.
     - Nếu thay đổi thực hiện không thành công thì hệ thống hiển thị thông báo lỗi kèm nội dung lỗi.

4. Người dùng nhấn vào nút "OK" trên thông báo để quay về trang chủ.

2.1.5 Xem lịch sử in

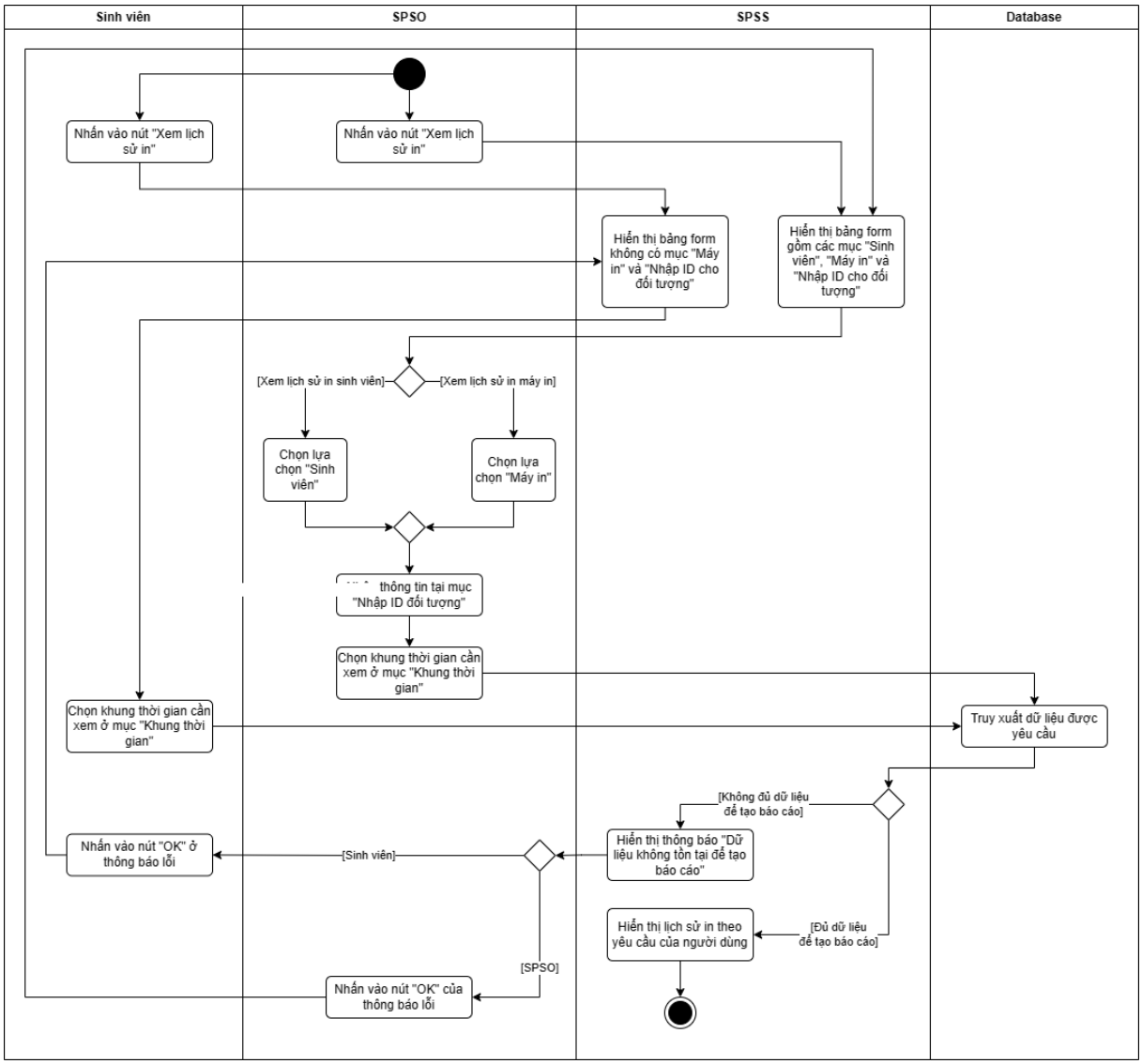


Figure 5: *Activity Diagram xem lịch sử in*

Xem chi tiết diagram tại đây

Mô tả các bước để thực hiện nhiệm vụ xem lịch sử in từ Activity Diagram:

1. Để sử dụng chức năng xem lịch sử in, người dùng đầu tiên sẽ phải nhấn vào nút có nội dung "Xem lịch sử in" trên giao diện. Lúc này hệ thống sẽ xác định xem người đang sử dụng chức năng là sinh viên hay là SPSO, khi này sẽ có hai trường hợp:
   Nếu người dùng là SPSO:
   (a) Một bảng form sẽ hiện lên giao diện với các mục "Sinh viên", "Máy in", "Nhập ID cho đối tượng" và "Khung thời gian"
   (b) Nếu người dùng muốn xem lịch sử in của đối tượng là sinh viên, chọn vào mục "Sinh viên", còn nếu người dùng muốn xem lịch sử in của đối tượng là máy in thì chọn vào mục "Máy in"
   (c) Sau khi đã chọn xong lựa chọn của mình, người dùng nhập thông tin ID của đối tượng cần truy cập tại mục "Nhập ID cho đối tượng", sau đó chọn vào khung thời gian cần xem lịch sử in ở mục "Khung thời gian"
   (d) Sau khi các thông tin đã được cung cấp đầy đủ, database thực hiện truy xuất dữ liệu tương ứng gửi về cho hệ thống để tạo báo cáo. Khi này sẽ có hai trường hợp xảy ra:
      - Nếu dữ liệu gửi về không đủ hay không tồn tại dữ liệu đó, một thông báo lỗi sẽ hiển thị lên màn hình, người dùng nhấn nút "OK" để trở về bảng form cung cấp thông tin ban đầu
      - Nếu dữ liệu gửi về thành công, hệ thống thực hiện tạo báo cáo từ dữ liệu đó và hiển thị lên màn hình báo cáo về lịch sử in vừa được tạo

   Nếu người dùng là Sinh viên:
   (a) Một bảng form hiện lên giao diện nhưng khác với SPSO ở chỗ sẽ chỉ còn mục "Khung thời gian"
   (b) Người dùng chọn khung thời gian cần xem lịch sử in ở mục "Khung thời gian"
   (c) Sau khi các thông tin đã được cung cấp đầy đủ, database thực hiện truy xuất dữ liệu tương ứng của người dùng gửi về cho hệ thống để tạo báo cáo. Khi này sẽ có hai trường hợp xảy ra:
      - *Nếu dữ liệu gửi về không đủ hay không tồn tại dữ liệu đó:* Một thông báo lỗi sẽ hiển thị lên màn hình, người dùng nhấn nút "OK" để trở về bảng form cung cấp thông tin ban đầu
      - *Nếu dữ liệu gửi về thành công:* Hệ thống thực hiện tạo báo cáo từ dữ liệu đó và hiển thị lên màn hình báo cáo về lịch sử in vừa được tạo

2.1.6 Xem báo cáo định kỳ

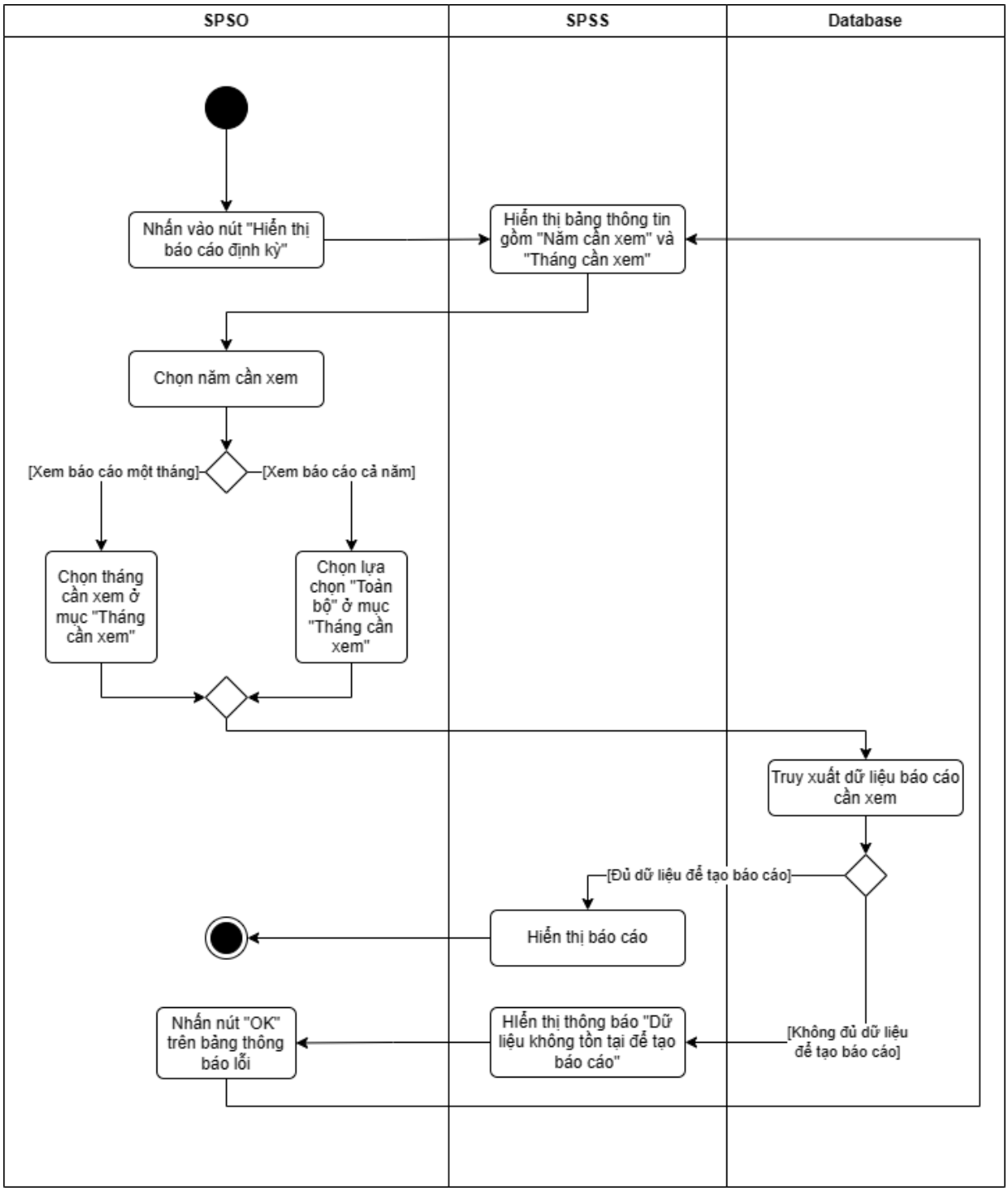


Figure 6: *Activity Diagram xem báo cáo định kỳ*

Xem chi tiết diagram tại đây

Mô tả các bước để thực hiện nhiệm vụ xem báo cáo định kỳ từ Activity Diagram:

1. SPSO muốn xem báo cáo định kỳ phải nhấn vào nút "Hiển thị báo cáo định kỳ"
2. Sau khi nhấn vào nút "Hiển thị báo cáo định kỳ", hệ thống hiển thị lên màn hình bảng thông tin gồm "Năm cần xem" và "Tháng cần xem"
3. SPSO chọn năm cần xem báo cáo ở mục "Năm cần xem"
4. Sau khi đã chọn năm của báo cáo cần xem, SPSO sẽ có hai lựa chọn:
   (a) *Xem báo cáo định kỳ của một tháng trong năm đó:* SPSO sẽ chọn tháng cần xem ở mục "Chọn tháng cần xem"
   (b) *Xem báo cáo cả năm đó:* SPSO sẽ chọn lựa chọn "Toàn bộ" ở mục "Chọn tháng cần xem"
5. Sau khi đã xác định được báo cáo định kỳ cần xem, thực hiện truy xuất dữ liệu báo cáo cần xem đã được lưu ở Database và gửi về hệ thống. Lúc này sẽ có hai trường hợp xảy ra:
   (a) *Nếu dữ liệu được truy xuất và gửi về thành công:* Hiển thị báo cáo cần xem lên màn hình
   (b) *Nếu dữ liệu được truy xuất và gửi về không thành công:* Hiển thị thông báo "Dữ liệu không tồn tại để tạo báo cáo", SPSO nhấn vào nút "OK" để có thể trở về bảng thông tin ban đầu

2.1.7 Tạo báo cáo định kỳ tự động

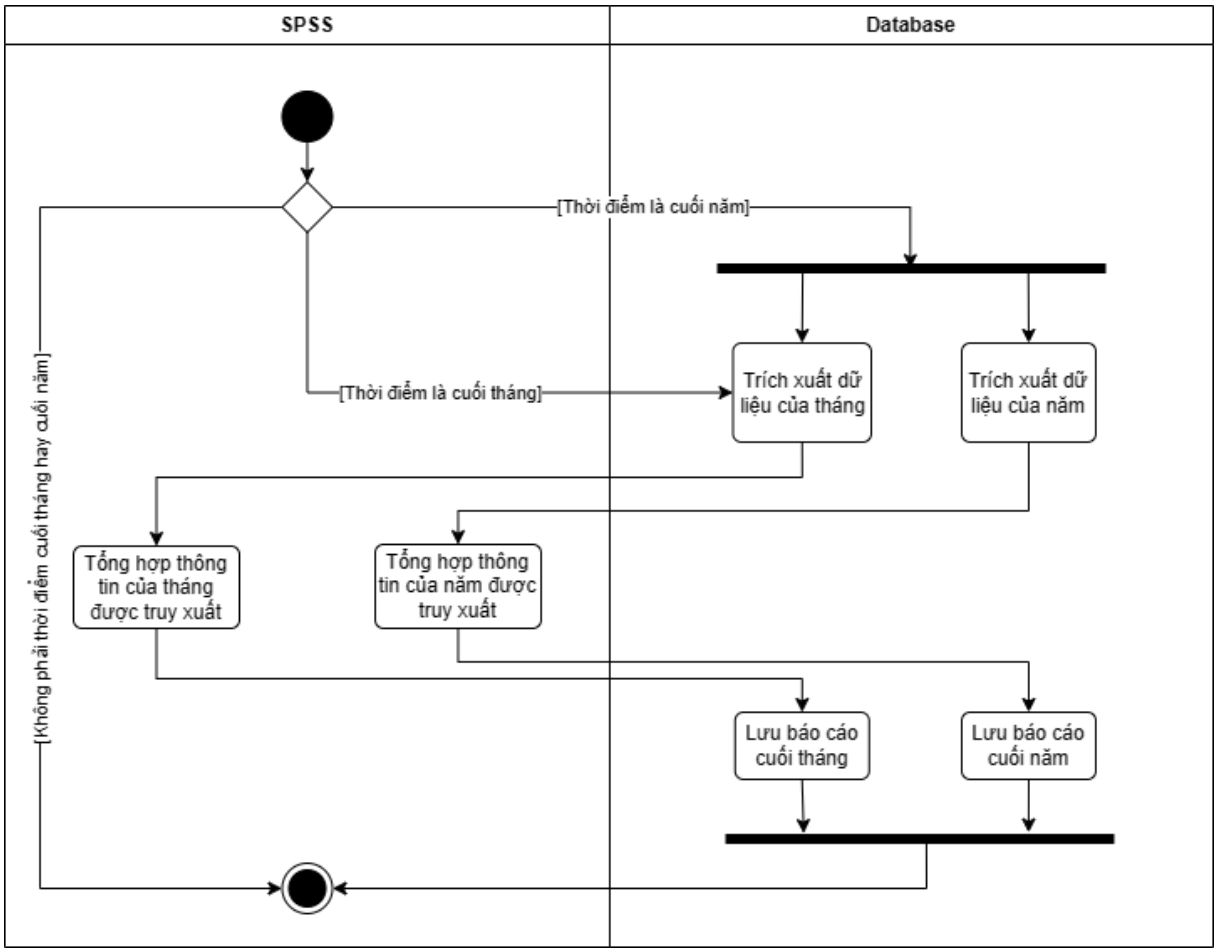


Figure 7: *Activity Diagram tạo báo cáo định kỳ tự động*

Xem chi tiết diagram tại đây

Mô tả các bước để thực hiện nhiệm vụ tạo báo cáo định kỳ tự động từ Activity Diagram:

1. Vào mỗi thời điểm cuối tháng, hệ thống sẽ tự động truy vấn vào Database để trích xuất dữ liệu của tháng đó. Đặc biệt là vào thời điểm cuối năm và cũng là thời điểm cuối tháng, dữ liệu của cả năm và của cả tháng tương ứng đó sẽ được trích xuất một lượt bởi hệ thống

2. Dựa vào dữ liệu được trích xuất được (của cả tháng hoặc của cả năm và cả tháng), hệ thống thực hiện tổng hợp thông tin của cả tháng (và cả năm) tương ứng thành (các) báo cáo hằng tháng (và hằng năm)

3. (Các) báo cáo được tổng hợp sẽ được lưu trở lại vào Database

Task 2.2

Draw sequence diagrams for the important module(s) chosen in Task 1.3.

2.2 Sequence Diagrams

2.2.1 Cung cấp thông tin in

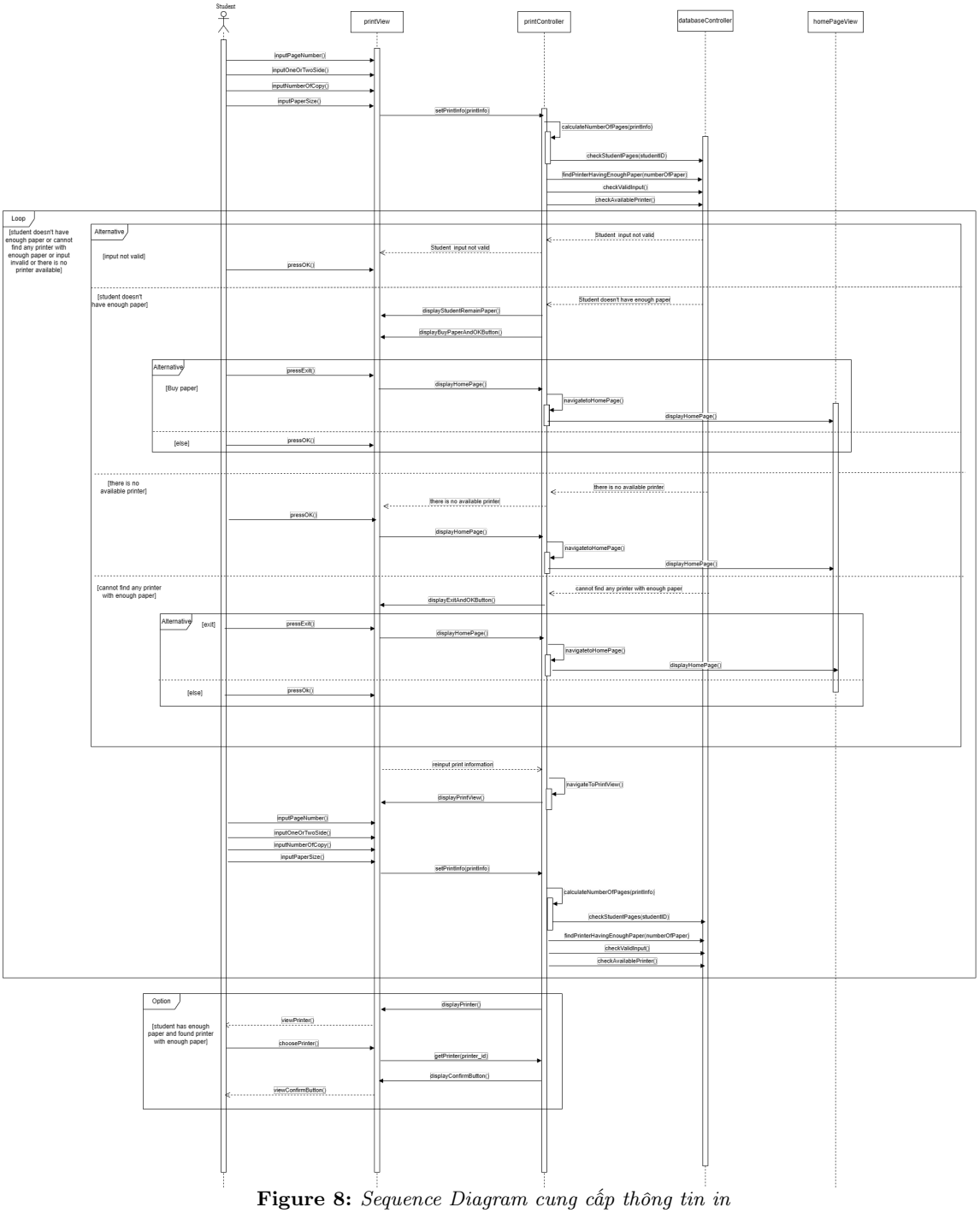


Figure 8: *Sequence Diagram cung cấp thông tin in*

Xem chi tiết Diagram: Here!

Mô tả các bước để thực hiện nhiệm vụ cung cấp thông tin in từ Sequence Diagram:

1. Student đang ở printView, tại đó student nhập các thông tin in cần thiết như số trang giấy, in một mặt hay hai mặt, số bản copy, kích thước giấy, sau đó ấn nút submit: gọi hàm setPrintInfo(printInfo): void để hệ thống cài đặt những thông tin in cần thiết. Thông tin của printInfo bao gồm số trang giấy, in một mặt hay hai mặt, số bản copy, kích thước giấy. Hệ thống sẽ kiểm tra input có hợp lệ hay không, có máy in nào đang hoạt động không, tính toán số trang cần in dựa vào thông tin của Student nhập vào và đối chiếu với số trang in của Student hiện có xem Student có đủ giấy hay không, cùng lúc đó, hệ thống cũng tìm những máy in có đủ giấy để in trong database.
   - Nếu input không hợp lệ thì hệ thống sẽ hiển thị thông báo lỗi “invalid input" lên màn hình, và yêu cầu sinh viên nhập lại những thông tin in cho phù hợp. Sau đó hệ thống lại tính toán và kiểm tra input có hợp lệ hay không. Quá trình này lặp lại cho đến khi sinh viên nhập vào input hợp lệ.
   - Nếu Student không đủ giấy thì hệ thống sẽ hiển thị thông báo lỗi “sinh viên không đủ giấy" lên màn hình. Sinh viên có hai lựa chọn: nhập lại thông in cho phù hợp hoặc mua thêm giấy. Nếu sinh viên chọn nhập lại thông tin thì hệ thống sẽ điều hướng lại về printView để sinh viên nhập lại thông tin. Sau đó hệ thống lại tính toán và kiểm tra sinh viên có đủ giấy hay không. Quá trình này lặp lại cho đến khi sinh viên có đủ số giấy cần thiết để in. Nếu sinh viên chọn mua thêm giấy thì hệ thống sẽ điều hướng lại về trang chủ (homePageView) để sinh viên có thể thực hiện mua thêm giấy.
   - Nếu không có máy in nào đang hoạt động thì hệ thống sẽ hiển thị thông báo lỗi “no printer available" lên màn hình. Sau khi sinh viên ấn nút OK thì hệ thống sẽ điều hướng về trang chủ.
   - Nếu không tìm thấy máy in nào có đủ giấy để in, hệ thống sẽ hiển thị thống báo lỗi “không tìm thấy máy in có đủ giấy” và yêu cầu sinh viên nhập lại những thông tin cho phù hợp hoặc thoát khỏi trang in và quay về trang chủ. Khi sinh viên nhập lại thông tin thì hệ thống lại tính toán lại số giấy cần thiết và tìm kiếm những máy in có đủ giấy. Quá trình này lặp lại cho đến khi hệ thống tìm được máy in phù hợp.
2. Student sẽ chọn một máy in phù hợp: gọi hàm getPrinter(printerID: Integer) để hệ thống lấy được thông tin máy in mà Student đã chọn.
3. Hệ thống sẽ gọi hàm displayConfirmButton(): void hiển thị nút xác nhận in ra màn hình.

2.2.2 Xác nhận in

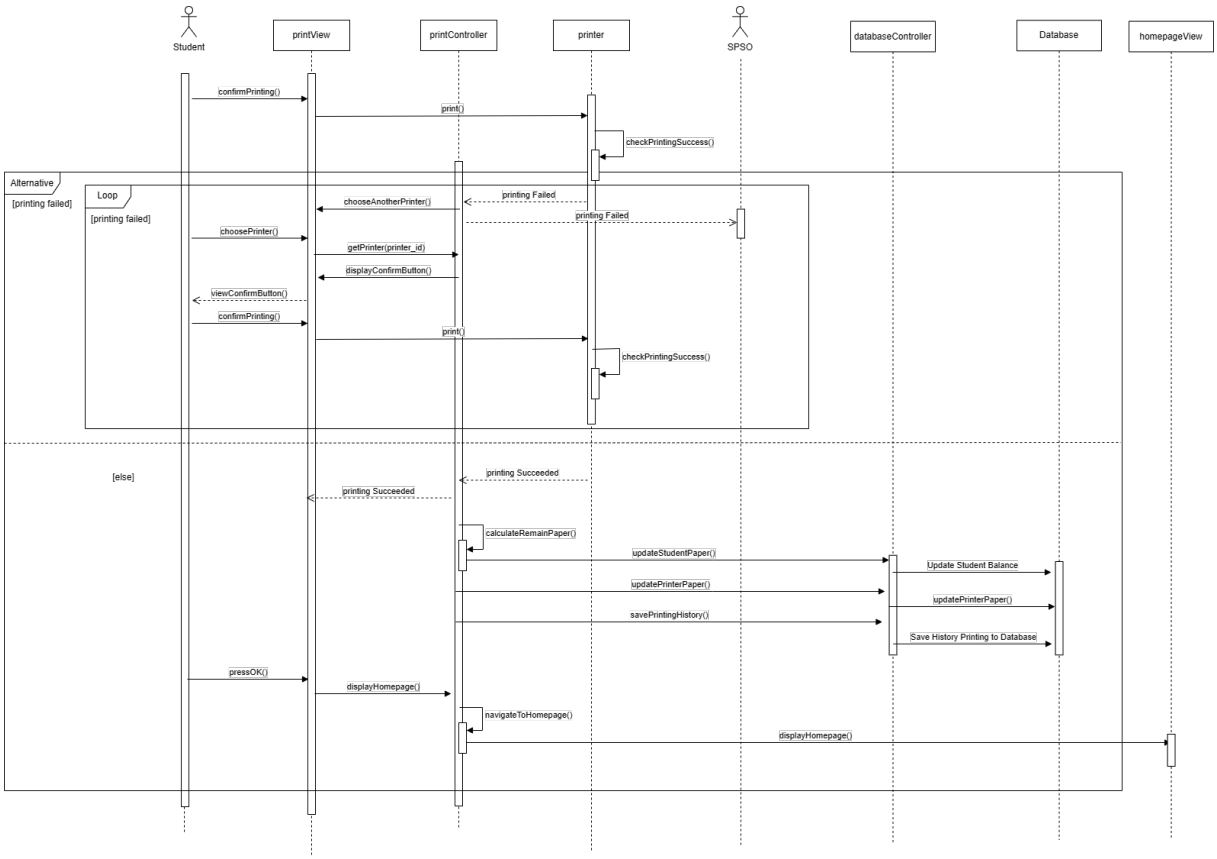


Figure 9: *Sequence Diagram xác nhận in*

Xem chi tiết Diagram: Here!

Mô tả các bước để thực hiện nhiệm vụ xác nhận in từ Sequence Diagram:

1. Student đang ở printView sau khi nhập xong những thông tin in cần thiết và muốn in tài liệu. Student click vào nút xác nhận thì hệ thống sẽ gọi hàm print(): void để gửi tín hiệu đến máy in được chọn và in.

2. Sau khi in xong, máy in kiểm tra tại liệu có được in thành công hay không:

   - Nếu in không thành công, hệ thống sẽ hiện thông báo lỗi in không thành công ra màn hình và yêu cầu sinh viên chọn máy in khác để in. Đồng thời hệ thống cũng gửi thông báo máy in in không thành công đến SPSO. Quá trính này lặp lại cho đến khi in thành công hoặc không còn máy in nào in thành công.
   - Nếu in thành công, hệ thống sẽ hiện thông báo in thành công ra màn hình, đồng thời tính toán lại lượng giấy in của Student và printer để cập nhật vào database thông qua hàm updateStudentPaper() và updatePrinterPage() của nó.

3. Hệ thống lưu lại lịch sử in vào database để tạo báo cáo hàng năm và để Student, cũng như SPSO xem lại lịch sử in sau này.

4. Cuối cùng, sinh viên ấn nút Ok để trở về trang chủ.

2.2.3 Quản lý máy in

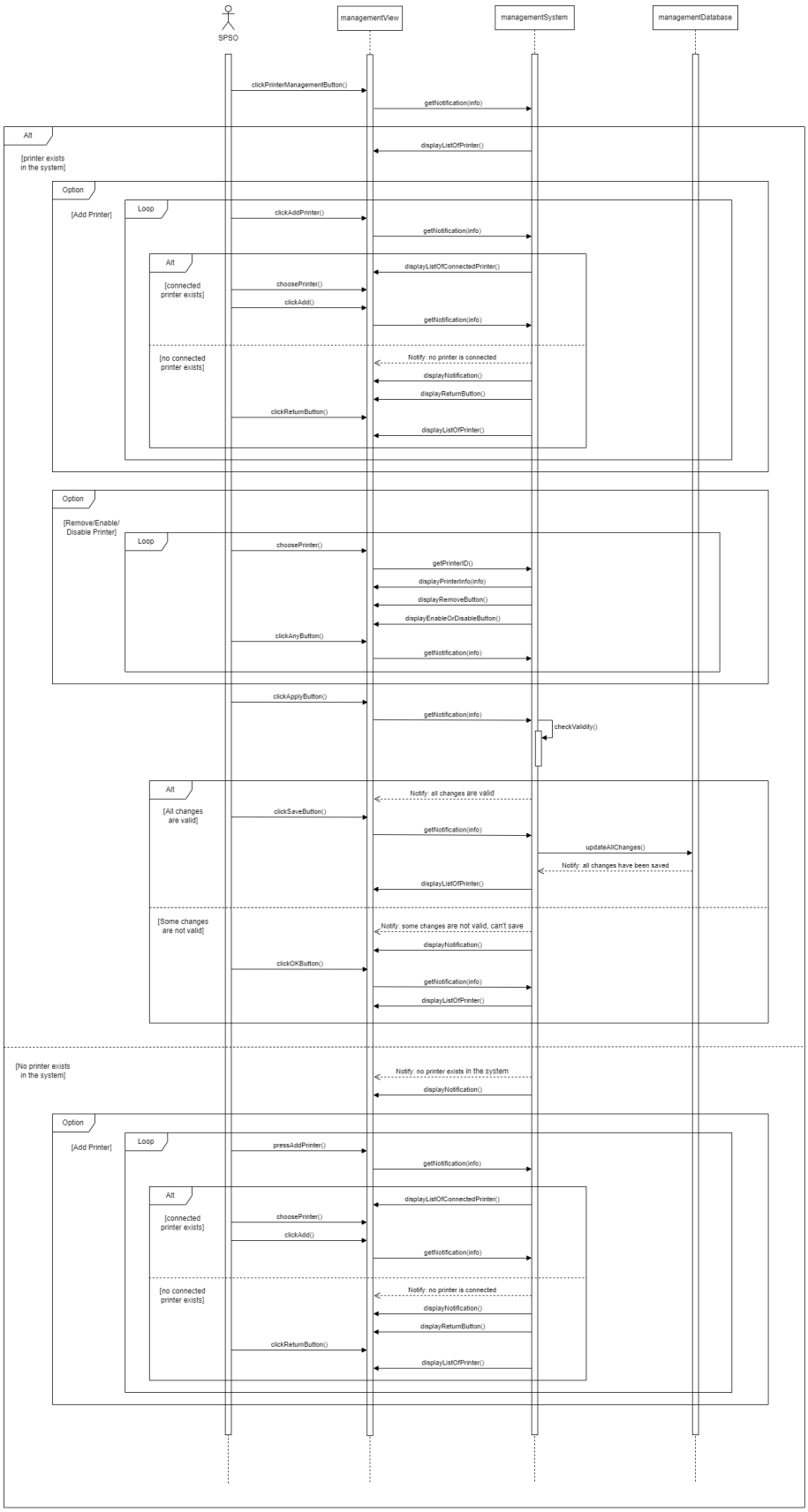


Figure 10: *Sequence Diagram quản lý máy in*

Xem chi tiết Diagram: Here!

Mô tả các bước để thực hiện nhiệm vụ quản lý máy in từ Sequence Diagram:

1. SPSO đang ở giao diện chính dành cho SPSO - Homepage/SPSO. Tại đây, SPSO muốn quản lý máy in của hệ thống sẽ nhấn vào nút "Quản lý máy in" qua đó hàm clickPrinterManagementButton(): void gửi đi một thông báo tới hệ thống quản lý qua hàm getNotification(info: String): void, hệ thống sẽ gọi hàm displayListOfPrinter(): void.
   - Nếu có tồn tại máy in trong hệ thống thì sẽ hiển thị qua danh sách máy in lên màn hình.
   - Ngược lại, nếu không tồn tại máy in trong hệ thống màn hình sẽ hiển thị ra dòng chữ "Hiện không có máy in trong hệ thống!" ở vùng hiển thị danh sách máy in.
2. Tại danh sách máy in này, SPSO có các lựa chọn như:
   (a) Thêm máy in đã được kết nối nhưng chưa được thêm vào hệ thống.
      - Đầu tiên, SPSO nhấn vào nút "Thêm máy in", lúc này hệ thống thực hiện lệnh clickAddPrinter(): void gửi đi một thông báo tới hệ thống quản lý qua hàm getNotification(info: String): void, hệ thống sẽ gọi hàm displayListOfConnectedPrinter(): void.
        – Nếu có máy in được kết nối hệ thống sẽ hiển thị ra màn hình danh sách các máy in đã được kết nối nhưng chưa được thêm vào hệ thống. Lúc này, SPSO có thể chọn máy in bất kì và nhấn nút "Thêm", hệ thống sẽ gọi hàm clickAdd() và lưu trữ một thông báo thay đổi.
        – Ngược lại, nếu không có máy in được kết nối với hệ thống màn hình sẽ hiển thị ra dòng chữ "Hiện không có máy in nào được kết nối với hệ thống!" ở vùng hiển thị danh sách máy in kết nối. Màn hình cũng hiển thị nút nhấn "Quay lại", khi SPSO nhấn vào sẽ đưa SPSO quay trở lại trang danh sách máy in (mục 2).

   (b) Xem thông tin và thao tác các lệnh trên một máy in cụ thể.
      - SPSO sẽ nhấn vào một máy in cụ thể để xem thông tin của máy in đó. Đồng thời hiển thị các nút nhấn như "Xóa", "Enable"/"Disable", khi SPSO thực hiện bất cứ thay đổi nào qua các nút nhấn, hệ thống sẽ lưu trữ một thông báo thay đổi.
3. Sau mọi thay đổi, SPSO thực hiện nhấn nút "Áp dụng" những thay đổi đã thực hiện với hệ thống, lúc này hệ thống gọi hàm clickApplyButton(): void.
4. Một thông báo được gửi tới hệ thống thông qua hàm getNotification(info: String): void, hệ thống bắt đầu kiểm tra tính hợp lệ của những thay đổi bằng cách tự gọi hàm checkValidity(): void.
   - Nếu hệ thống nhận thấy các thay đổi của SPSO là hợp lệ sẽ gửi thông báo ra màn hình "Những thay đổi của bạn được áp dụng thành công!". Khi đó, SPSO có thể lưu khi click vào nút "Lưu", những thay đổi sẽ được lưu trữ vào database và hệ thống sẽ hiển thị thông báo "Những thay đổi của bạn đã được lưu!". Sau đó, SPSO được đưa về trang danh sách máy in (mục 2).
   - Nếu hệ thống nhận thấy các thay đổi của SPSO là không hợp lệ sẽ gửi thông báo ra màn hình "Những thay đổi của bạn không hợp lệ!" kèm với lỗi cụ thể, đồng thời ngay phía dưới là nút OK để SPSO có thể nhấn vào khi nhận được thông báo. Sau khi nhấn nút OK, SPSO được đưa về trang danh sách máy in (mục 2).

2.2.4 Quản lý cài đặt in

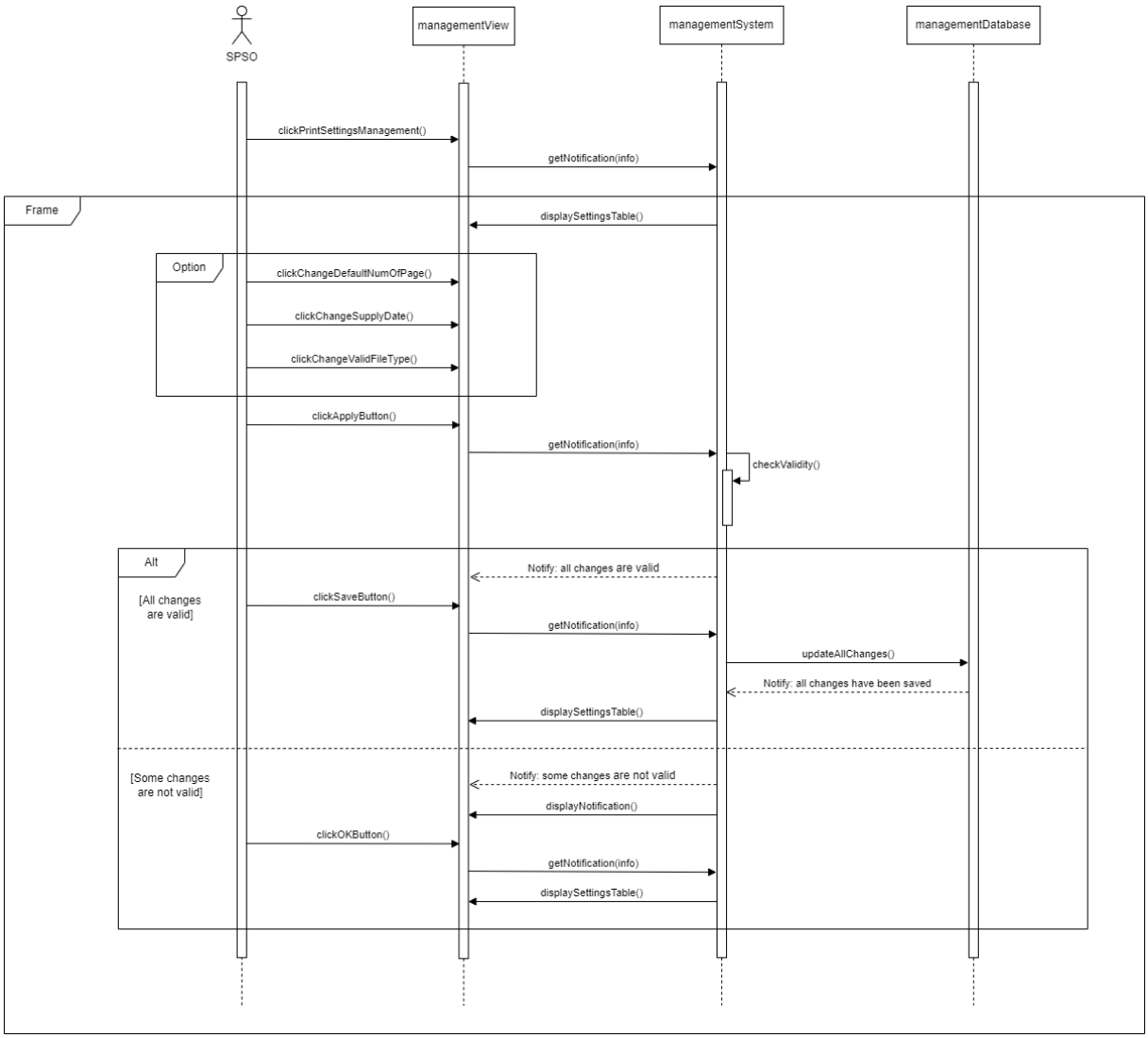


Figure 11: *Sequence Diagram quản lý cài đặt in*

Xem chi tiết Diagram: Here!

Mô tả các bước để thực hiện nhiệm vụ quản lý cài đặt in từ Sequence Diagram:

1. SPSO đang ở giao diện chính dành cho SPSO - Homepage/SPSO. Tại đây, SPSO muốn quản lý cài đặt in của hệ thống sẽ nhấn vào nút "Quản lý cài đặt in" qua đó hàm clickPrintSettingsManagement(): void gửi đi một thông báo tới hệ thống quản lý qua hàm getNotification(info: String): void, hệ thống sẽ hiển thị lên màn hình bảng cài đặt in qua hàm displaySettingsTable(): void.

2. Tại bảng cài đặt này, SPSO có các lựa chọn như:

   - Thay đổi số trang mặc định được cung cấp cho sinh viên vào mỗi học kỳ.
   - Thay đổi ngày hệ thống cung cấp số trang mặc định cho sinh viên vào mỗi học kỳ.
   - Thay đổi những loại file mà sinh viên được phép upload lên hệ thống để thực hiện in ấn.

   Sau mọi thay đổi, SPSO thực hiện nhấn nút "Áp dụng" những thay đổi đã thực hiện với hệ thống, lúc này hệ thống gọi hàm clickApplyButton(): void.

3. Một thông báo được gửi tới hệ thống thông qua hàm getNotification(info: String): void, hệ thống bắt đầu kiểm tra tính hợp lệ của những thay đổi bằng cách tự gọi hàm checkValidity(): void.

   - Nếu hệ thống nhận thấy các thay đổi của SPSO là hợp lệ sẽ gửi thông báo ra màn hình "Những thay đổi của bạn được áp dụng thành công!". Khi đó, SPSO có thể lưu khi click vào nút "Lưu", những thay đổi sẽ được lưu trữ vào database và hệ thống sẽ hiển thị thông báo "Những thay đổi của bạn đã được lưu!". Sau đó, SPSO được đưa về giao diện bảng cài đặt in của hệ thống.
   - Nếu hệ thống nhận thấy các thay đổi của SPSO là không hợp lệ sẽ gửi thông báo ra màn hình "Những thay đổi của bạn không hợp lệ!" kèm với lỗi cụ thể, đồng thời ngay phía dưới là nút OK để SPSO có thể nhấn vào khi nhận được thông báo. Sau khi nhấn nút OK, SPSO được đưa về giao diện bảng cài đặt in của hệ thống.

2.2.5 Xem lịch sử in

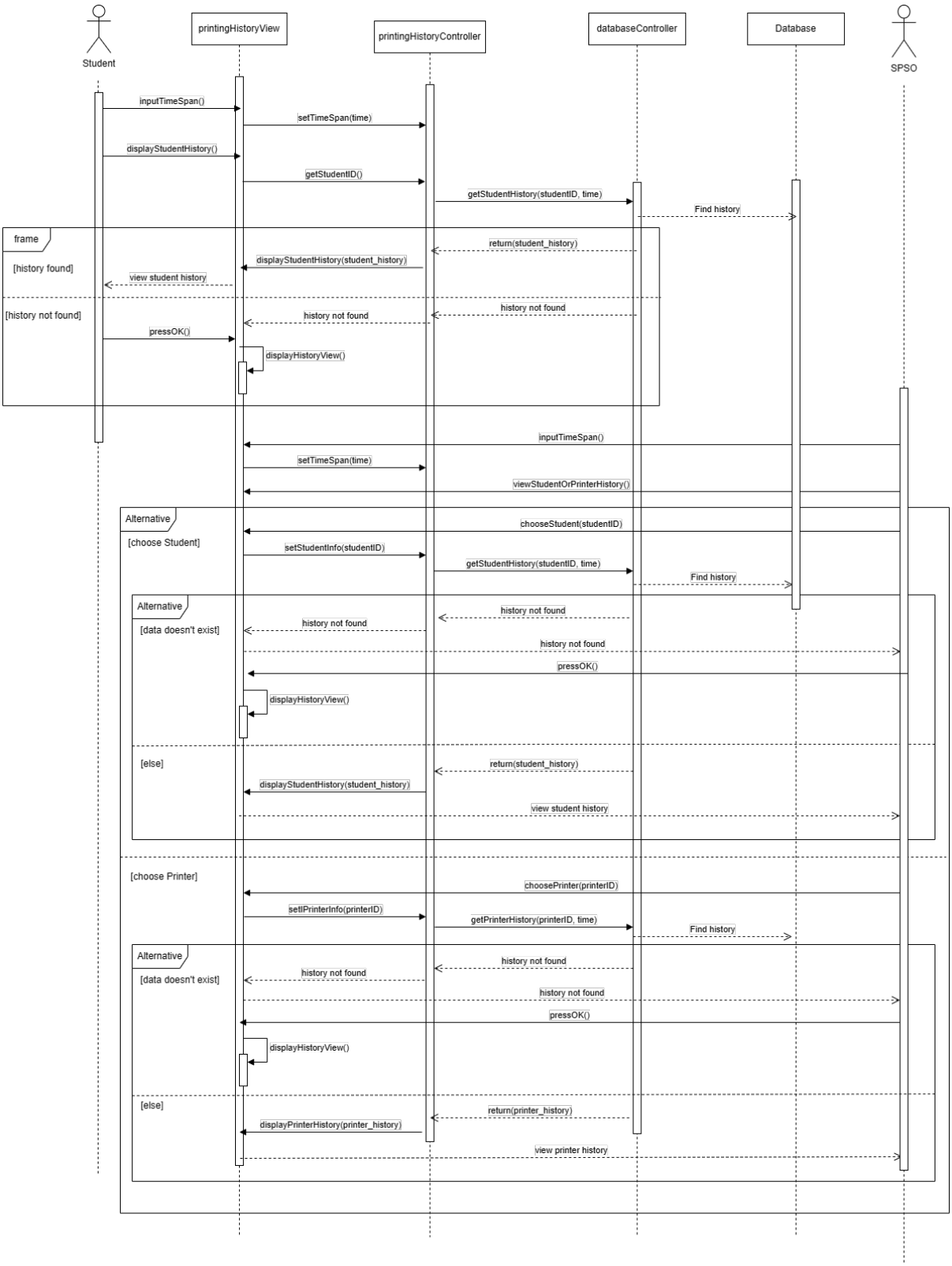


Figure 12: *Sequence Diagram xem lịch sử in*

Xem chi tiết Diagram: Here!

Mô tả các bước để thực hiện nhiệm vụ xem lịch sử in từ Sequence Diagram:

1. Student đang ở printingHistoryView, tại đó muốn xem lịch sử in của chính Student đó, gọi hàm inputTimeSpan(): void, nhập vào khoảng thời gian muốn xem lịch sử của Student hoặc Printer. Sau đó gọi hàm displayStudentHistory(): void, hệ thống sẽ tìm kiếm dữ liệu in của sinh viên thông qua hàm getStudentHistory(StudentID: Integer, time), trong đó StudentID là ID của Student gọi hàm displayStudentHistory(), time là khoảng thời gian mà student đã nhập vào.

   - Nếu hệ thống tìm thấy lịch sử của student, Kết quả của student_history sẽ được trả về, printController thực hiện hàm displayStudentHistory(student_history) để hiện thị thông tin về lịch sử in chi tiết cho Student có thể quan sát.
     Student_history sẽ bao gồm những thông tin về lịch sử in của sinh viên: ID của sinh viên, ID của máy in, tên file in, thời gian bắt đầu và kết thúc in, số trang A3 và A4.
   - Nếu hệ thống không tìm thấy lịch sử của student, hệ thống sẽ hiển thị thông báo lỗi "không tìm thấy lịch sử in" ra màn hình, student sẽ ấn nút OK để quay về trang chủ.

2. SPSO đang ở printingHistoryView, tại đó muốn xem lịch sử in của Student bất kì hoặc Printer bất kì: gọi hàm inputTimeSpan(): void, nhập vào khoảng thời gian muốn xem lịch sử của Student hoặc Printer. Sau đó hệ thống gọi hàm viewStudentOrPrinter(): bool để hỏi SPSO xem lịch sử của Student hay Printer, nếu SPSO chọn Student, kết quả của hàm sẽ trả về true, ngược lại nếu SPSO chọn Printer, kết quả của hàm sẽ trả về false.

   (a) Nếu SPSO chọn Student, hệ thống gọi hàm chooseStudent(studentID: Integer) yêu cầu SPSO nhập ID của Student muốn xem lịch sử. Sau đó hệ thống sẽ gọi hàm getStudentHistory(studentID: Integer, time) để lấy lịch sử in của Student từ database.

   - Nếu không tìm được lịch sử in của sinh viên, hệ thống sẽ báo lỗi “history not found” và hiển thị ra màn hình. Khi Student ấn nút OK thì use-case sẽ kết thúc do tìm lịch sử không thành công.
   - Ngược lại, nếu hệ thống tìm được lịch sử in của sinh viên thì sẽ gọi hàm displayStudentHistory(student_history) để hiện thị thông tin về lịch sử in chi tiết của Student đó.

   (b) Nếu SPSO chọn Printer, hệ thống gọi hàm choosePrinter(studentID: Integer) yêu cầu SPSO nhập ID của Printer muốn xem lịch sử. Sau đó hệ thống sẽ gọi hàm getPrinterHistory(printerID: Integer, time) để lấy lịch sử in của Printer từ database.

   - Nếu không tìm được lịch sử in của máy in, hệ thống sẽ báo lỗi “history not found” và hiển thị ra màn hình cùng với nút OK. Khi SPSO ấn vào nút OK thì hệ thống sẽ chuyển về giao diện printHistoryView() và kết thúc.
   - Ngược lại, nếu hệ thống tìm được lịch sử in của máy in thì sẽ gọi hàm displayPrinterHistory(printer_history) để hiện thị thông tin về lịch sử in chi tiết của Printer đó.

2.2.6 Xem báo cáo in

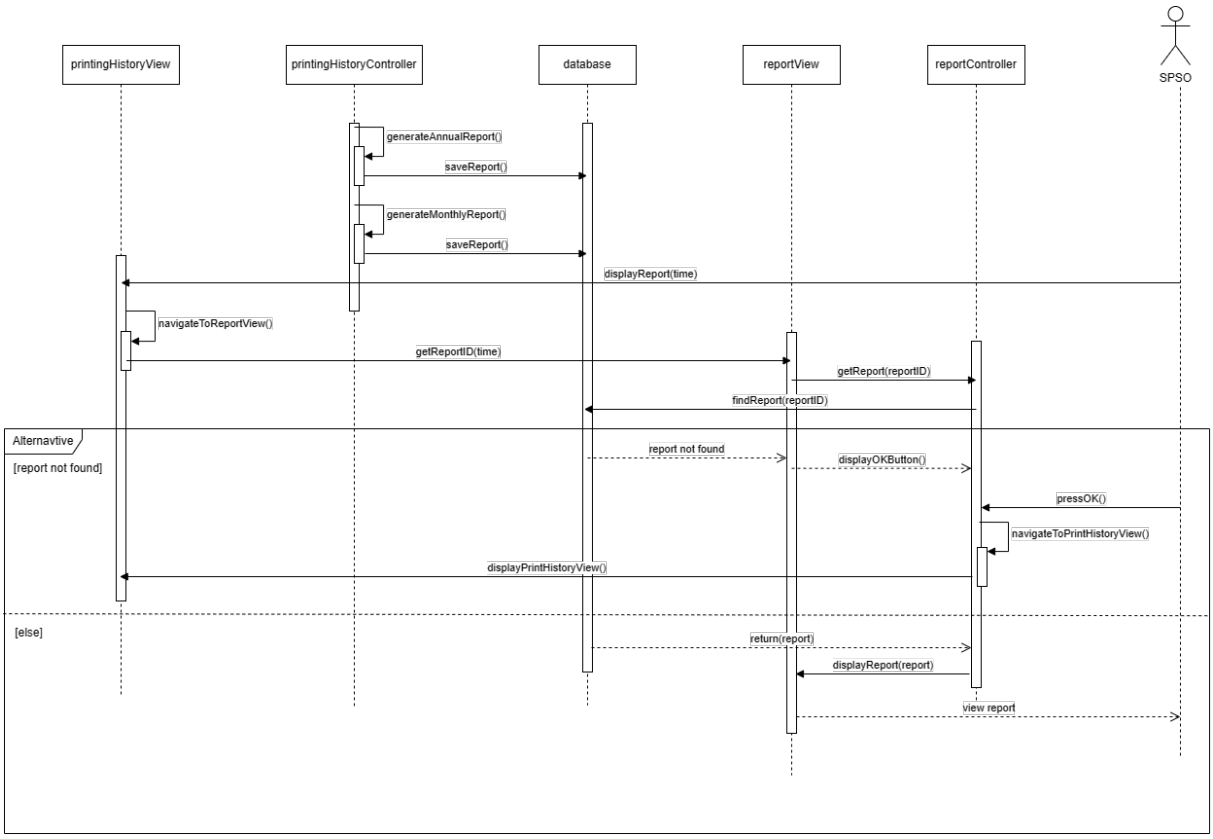


Figure 13: *Sequence Diagram xem báo cáo in*

Xem chi tiết Diagram: Here!

Mô tả các bước để thực hiện nhiệm vụ xem báo cáo in từ Sequence Diagram:

1. Vào cuối mỗi tháng và mỗi năm, hệ thống sẽ tự động tạo báo cáo định kì bằng các hàm generateMonthlyReport(): void và generateAnnualReport(): void và lưu vào database.

2. SPSO đang ở printingHistoryView, tại đó muốn xem báo cáo định kì của hệ thống, gọi hàm displayReport(time) nhập vào thời gian muốn xem báo cáo. Hệ thống tự động gọi hàm navigateToReportView(): void để điều hướng tới reportView.

3. Hệ thống gọi hàm getReportID(time) để lấy ID của report cần xem dựa vào thời gian mà SPSO đã nhập trước đó. Sau khi lấy được ID của report, hệ thống sẽ tìm kiếm thông tin của report trong database thông qua hàm findReport(reportID).

   - Nếu không tìm được thông tin của report thì hệ thống sẽ báo lỗi "report not found" và nút OK ra màn hình. Khi SPSO ấn nút OK thì hệ thống sẽ chuyển về giao diện printHistoryView và kết thúc.

   - Ngược lại, nếu tìm được thông tin của report thì hệ thống sẽ gọi hàm displayReport(report) để hiển thị thông tin chi tiết của báo cáo đó.

Task 2.3

Draw a class diagram of the important module(s) chosen in Task 1.3 as comprehensive as possible.

2.3 Class Diagrams

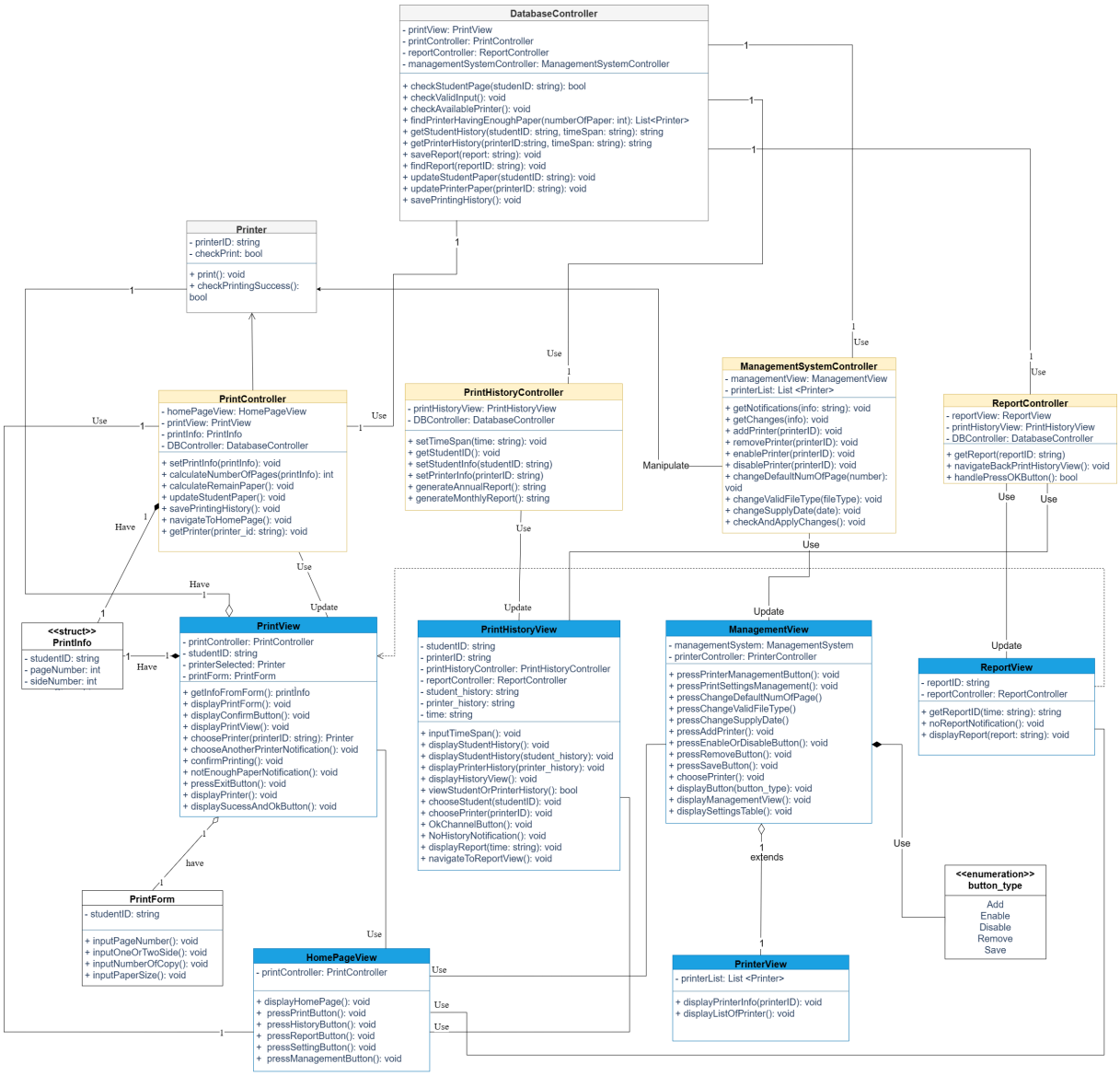


Figure 14: *Class Diagram For Important Module*

Xem chi tiết Diagram: Here!

Mô tả Class và các quan hệ của chúng

- PrintView

  1. **Chức năng chính:** hiển thị giao diện thực hiện việc in ấn, hỗ trợ việc nhập các thông số in, chọn máy in và xác nhận in.

2. **Các thuộc tính:**
   - printController: PrintController
   - studentID: string
   - printSelected: Printer
   - printForm: printForm
3. Các phương thức
   - getInfoFromForm(): Đọc các thông số in người dùng nhập vào
   - displayPrintForm(): hiển thị form nhập thông số in cho người dùng
   - displayConfirmButton(): hiển thị xác nhận in và trigger việc in ấn
   - displayPrintView(): hiển thị giao diện trang web thực hiện in.
   - choosePrinter(printerID: string): chọn máy in người dùng yêu cầu thực hiện in ấn
   - chooseAnotherPrinterNotification(): Hiển thị thông báo chọn lại máy in khi máy in không đủ giấy hoặc máy in bị lỗi.
   - confirmPrinting(): xác nhận sau khi chọn máy in
   - notEnoughPaperNotification(): hiển thị thông báo không đủ giấy của máy in lựa chọn
   - pressExitButton(): nút thoát khỏi thông báo không đủ giấy in để quay về trang chủ.
   - displayPrinter(): hiển thị ra danh sách máy in đủ giấy và không bị lỗi để người dùng lựa chọn
   - displaySuccessAndOkButton(): hiển thị thông báo in thành công và nút OK để thoát
4. **Các mối quan hệ**:
   - **Với PrintController:** <association> chức năng nhập thông số in vào hệ thống, đếm số giấy, và chọn các máy in phù hợp với nhu cầu in từ PrintController.
   - **Với PrinInfo:** <association> Mỗi PrintView có một thuộc tính PrintInfo lưu thông số in sinh viên nhập vào.
   - **Với Printer:** <association> Chức năng in và kiểm tra in thành công từ Printer.
   - **Với PrintForm:** <association> Chức năng nhập vào các thông số từ PrintForm.
   - HomePageView: <association> nút chọn in ấn từ HomePageView

- PrintHistoryView

1. **Chức năng chính:** Hiển thị giao diện xem lịch sử in, hỗ trợ hiển thị lịch sử in của sinh viên hoặc máy in, báo cáo định kỳ.
2. **Các thuộc tính:**
   - studentID: string
   - printerID: string
   - printHistoryController: PrintHistoryController
   - reportController: ReportController
   - student_history: string
   - printer_history: string
   - time: string

3. **Các phương thức:**
   - inputTimeSpan(): đọc vào khoảng thời gian cần xem lịch sử của đối tượng.
   - displayStudentHistory(): Hiển thị ra lịch sử in của sinh viên khi đăng nhập dạng sinh viên.
   - displayStudentHistory(student_history): Hiển thị ra lịch sử in của sinh viên khi đăng nhập dưới dạng SPSO.
   - displayPrinterHistory(printer_history): Hiển thị ra lịch sử của máy in.
   - viewStudentOrPrinterHistory(): Trigger kiểm tra SPSO chọn xem lịch sử sinh viên hay máy in.
   - chooseStudent(studentID): Đọc vào ID của sinh viên để truy xuất dữ liệu.
   - choosePrinter(printerID): Đọc vào ID của máy in để truy xuất dữ liệu lịch sử.
   - NoHistoryNotification(): Hiển thị thông báo khi không tìm thấy lịch sử in.
   - displayReport(time): Nút bấm hiển thị báo cáo định kỳ với thời gian yêu cầu.
   - navigateToReportView(): Điều hướng tới trang ReportView.
4. **Các mối quan hệ**
   - printHistoryController: <association> chức năng nhập các khoảng thời gian, ID của sinh viên hoặc máy in để in ra lịch sử từ printHistoryController.
   - reportController: <association> chức năng chuyển hướng ngược về trang printHistoryView từ ReportController.
   - HomePageView: <association> nút chọn xem lịch sử từ HomePageView

- ManagementView

1. **Chức năng chính:** Hiển thị giao diện quản lý máy in và quản lý cài đặt in
2. **Các thuộc tính:**
   - managementSystemController: ManagementSystemController
   - printerController: PrinterController
   - printerView: PrinterView
3. **Các phương thức:**
   - pressPrinterManagementButton(): Nút chọn điều khiển điều hướng sang quản lý máy in.
   - pressPrintSettingManagement(): Nút chọn điều hướng sang quản lý cài đặt thông số in.
   - pressChangeDefaultNumOfPage(): Hiển thị phần nhập thay đổi số trang mặc định.
   - pressChangeValidFileType(): Hiển thị phần nhập thay đổi dạng file được chấp nhận.
   - pressChangeSupplyDate(): Hiển thị phần nhập thay đổi ngày cấp số trang mặc định cho sinh viên.
   - pressSaveButton(): Nút lưu những thay đổi của thông số mặc định
   - pressAddPrinter(): Nút thêm máy in
   - pressEnableOrDisableButton(): Nút khởi động hoặc vô hiệu một máy in.
   - pressRemoveButton(): Nút xóa máy in.
   - choosePrinter(): Chọn máy in cần hiện thực thay đổi.
   - displayButton(button_type): Hiển thị các nút điều khiển máy in.
   - displayManagementView(): Hiển thị giao diện quản lý.
   - displaySettingsTable(): Hiển thị bảng cài đặt thông số in.

4. **Các mối quan hệ**
   - ManagementSystemController: <association> các chức năng điều khiển máy in và thay đổi cài đặt thông số in từ ManagementSystemController.
   - HomePageView: <association> nút chọn quản lý hệ thống từ HomePageView.
   - PrinterView: <association> chức năng hiển thị danh sách máy in và thông tin từng máy in từ PrintView.
   - button_type: <association> Các nút điều khiển máy in.

- ReportView
  1. **Chức năng chính:** Hiển thị giao diện và báo cáo định kỳ.
  2. **Các thuộc tính:**
     - reportID: string
     - reportController: ReportController
  3. **Các phương thức:**
     - getReportID(time: string): Lấy ID của báo cáo từ thời gian SPSO đưa vào
     - NoReportNotification(): Hiển thị thông báo không tìm thấy báo cáo định kỳ.
     - displayReport(report:string): Hiển thị báo cáo định kỳ SPSO yêu cầu.
  4. **Các mối quan hệ:**
     - ReportController: <association> chức năng lấy báo cáo dựa trên ID.
     - HomePageView: <association> nút báo cáo (Report) từ HomePageView.

- HomPageView
  1. **Chức năng chính**: Hiển thị giao diện trang chủ hệ thống và các chức năng.
  2. **Các thuộc tính:**
     - printController: PrintController
  3. **Các phương thức:**
     - displayHomePage(): hiển thị giao diện trang chủ
     - pressPrintButton(): nút chức năng in và điều hướng sang PrintView.
     - pressHistoryButton(): nút chức năng lịch sử in và điều hướng sang PrintHistoryView.
     - pressReportButton(): nút chức năng xem báo cáo định kỳ và điều hướng sang ReportView.
     - pressSettingButton(): nút chức năng cài đặt thông số in và điều hướng sang ManagementView.
     - pressManagementButton(): nút chức năng quản lý máy in và điều hướng sang ManagementView.
  4. **Các mối quan hệ**
     - PrintController: <association> chức năng điều hướng về homepage từ PrintController.
     - PrintView: <association> cung cấp bổ trợ chức năng thực hiện in.
     - PrintHistoryView: <association> cung cấp bổ trợ chức năng xem lịch sử in.
     - ManagementView: <association> cung cấp bổ trợ chức năng quản lý hệ thống in.
     - ReportView: <association> cung cấp bổ trợ chức năng xem báo cáo.

- PrinterView
  1. **Chức năng chính**: Cung cấp hiển thị giao diện danh sách máy in và thông tin in
  2. **Các thuộc tính**
     - printerList: List <Printer>
  3. **Các phương thức:**
     - displayPrinterInfo(printerID): Hiển thị thông tin của máy in.
     - displayListOfPrinter(): Hiển thị danh sách các máy in.
  4. **Các mối quan hệ:**
     - ManagementView: <association> hỗ trợ thêm cho các tính năng trong ManagementView
     - Printer: <aggregation> Một PrintView chứa nhiều thông tin về các máy in.
- PrintForm
  1. **Chức năng chính:** Form để điền các thông số in
  2. **Các thuộc tính:**
     - studentID: string
  3. **Các phương thức:**
     - inputPageNumber(): Nhập vào trang cần in của tài liệu.
     - inputOneOrTwoSide(): Xác định in một mặt hay hai mặt.
     - inputNumberOfCopy(): Nhập vào só bản copy muốn in.
     - inputPaperSize(): Xác định kích cỡ giấy in
  4. **Các mối quan hệ:**
     - PrintView: <association> hỗ trợ cho việc nhập thông tin in ấn trong PrintView.
- PrintController
  1. **Chức năng chính:** Điều khiển luồng dữ liệu thông tin và thực hiện in
  2. **Các thuộc tính:**
     (a) homePageView: HomePageView
     (b) printView: PrintView
     (c) printInfo: PrintInfo
     (d) DBController: DatabaseController
  3. **Các phương thức:**
     (a) setPrintInfo(printInfo): setter cho thông tin in vào controller.
     (b) calculateNumberOfPages(printInfo): trả về tổng số trang thực hiện in
     (c) calculateRemainPaper(): kiểm tra số trang còn lại trong tài khoản sinh viên và máy in thực hiện in.
     (d) navigateToHomePage(): Điều hướng về trang chủ
     (e) getPrinter(printer_id: string): Thiết lập máy in thực hiện in.
  4. **Các mối quan hệ:**

Task 2.4

Develop MVP 1 as user interfaces of either a Desktop-view central dashboard for a particular module (the same with the module used in task 2.1). Decide yourself what to include in the view. Use a wireframe tool like Figma or Adobe XD, or Illustrator

2.4 Thiết kế giao diện phần mềm cho những Module đã chọn

Link Figma: Here!

2.4.1 Cung cấp thông tin in

Sinh viên và SPSO sau khi đăng nhập được vào hệ thống thì có thể nhấn vào nút "Print Now" ở Homepage để thực hiện chức năng "Cung cấp thông tin in". Tại đây hệ thống sẽ chuyển hướng về trang "Print Now" và hiện ra một bảng để người dùng tải file cần in lên.

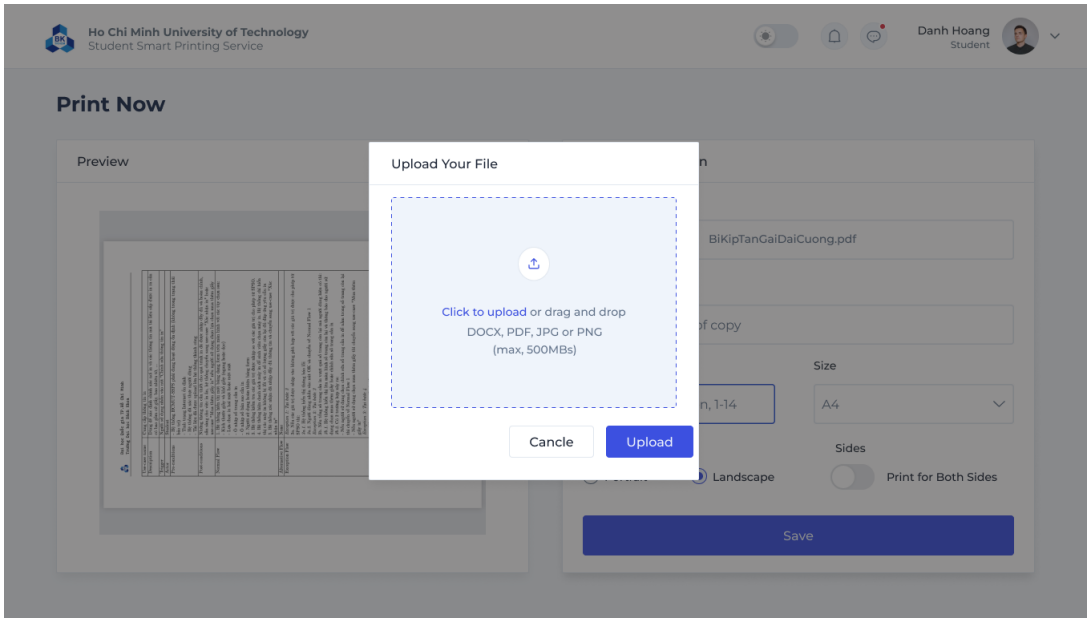


Figure 15: *UI Cung cấp thông tin in - Upload File*

Sau khi đã tải file thành công và bấm "Upload" (hoặc bấm "Cancel" để không tải file lên) thì sẽ xuất hiện form điền thông tin in. Tại đây người dùng sẽ nhập các thông tin như kích thước giấy, số trang in, số bản sao và chế độ in. Sau đó người dùng bấm "Save" và hệ thống sẽ kiểm tra xem các thông tin này có hợp lệ để tiến hành in hay không.

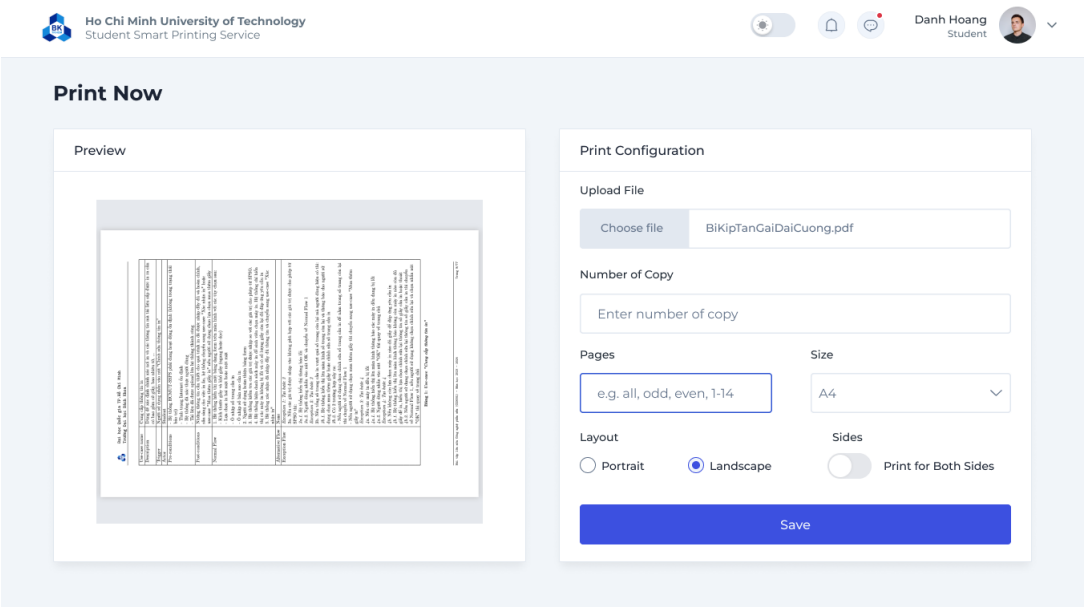


Figure 16: *UI Cung cấp thông tin in - Form điền thông tin và Preview*

Tại đây có 3 trường hợp xảy ra:

Thông tin in không hợp lệ: Lúc này sẽ hiện ra thông báo lỗi và người dùng có thể nhấn nút quay lại trang điền thông tin để chỉnh sửa lại.

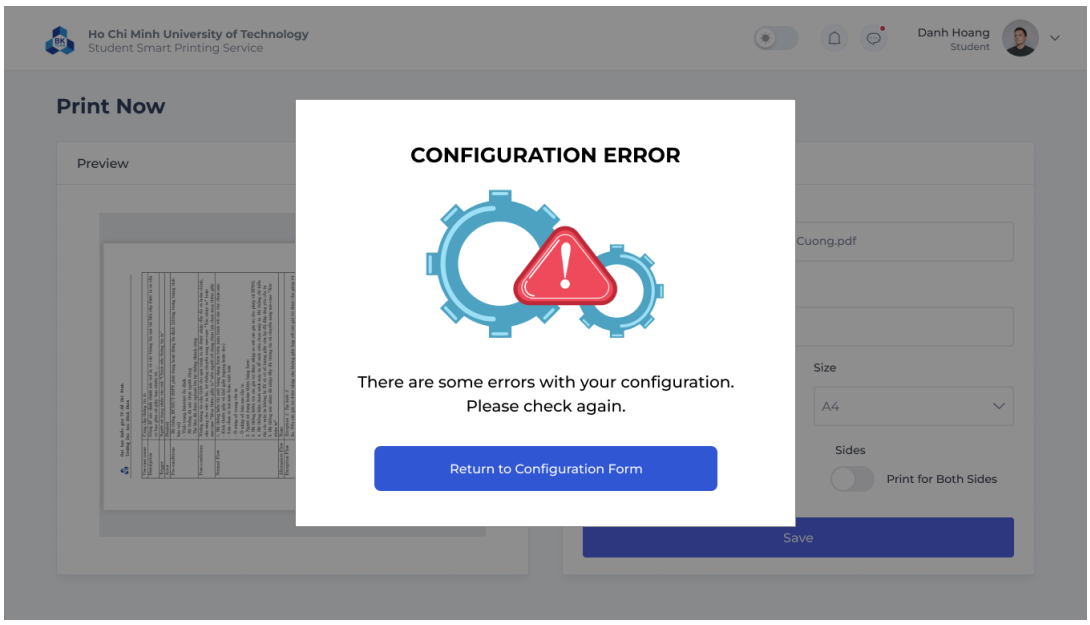


Figure 17: *UI Cung cấp thông tin in - Thông tin in không hợp lệ*

Người dùng không đủ giấy in: Lúc này sẽ có thông báo "Không đủ giấy in" hiện ra và người dùng có hai lựa chọn: mua thêm giấy in hoặc là chỉnh sửa lại số trang in.

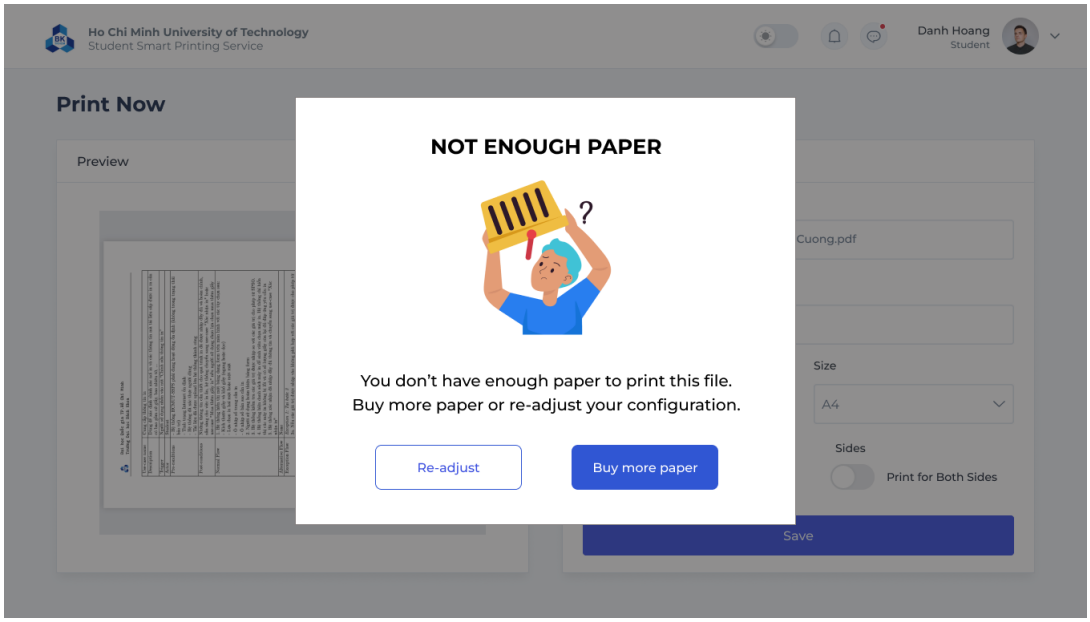


Figure 18: *UI Cung cấp thông tin in - Không đủ giấy in*

Thông tin in hợp lệ và lượng giấy in còn đủ: Lúc này hệ thống sẽ chuyển hướng đến trang chọn máy in để in. Người dùng có thể chọn máy in bằng cách nhấn vào nút "Choose" ở bên phải mỗi máy in.

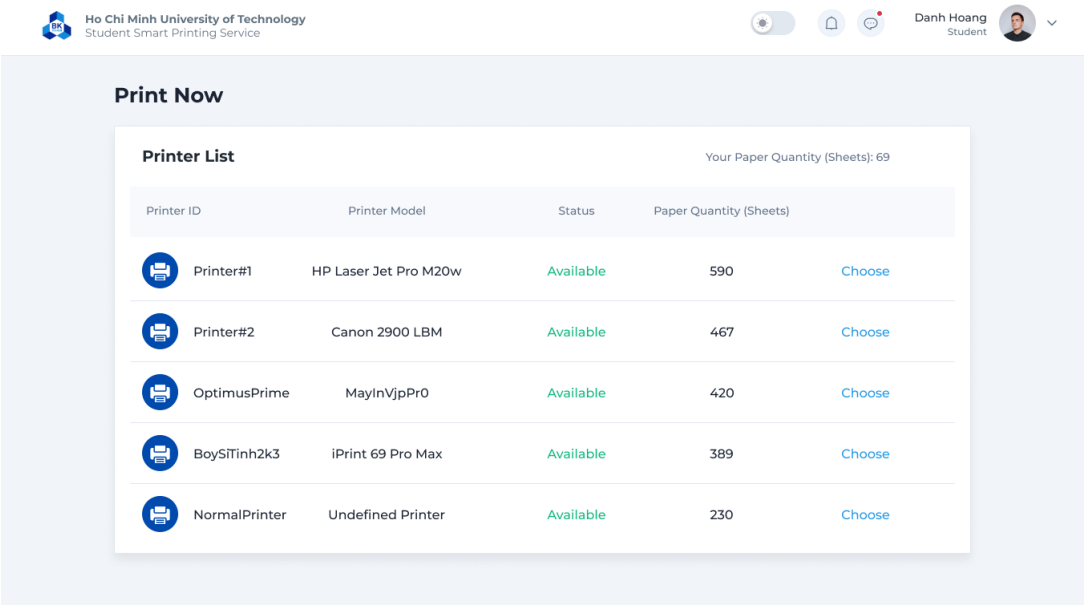


Figure 19: *UI Cung cấp thông tin in - Chọn máy in*

Chức năng của các nút/đường dẫn trong giao diện Cung cấp chức năng in được mô tả trong bảng sau:

Tên	Loại	Mô tả
Logo Bách Khoa	Link	Chuyển về trang chủ
Choose File	Button	Hiển thị giao diện upload file cần in
Number of Copy, Pages, Size	Input	Điền các thông tin in cần thiết để in
Layout	Checkbox Button	Chọn chế độ in Landscape hoặc Portrait
Print for Both Sides	Toggle	Chọn in một mặt hay hai mặt
Save	Button	Lưu các thông tin in
Choose	Button	Chọn máy in muốn in

2.4.2 Xác nhận in

Sau khi chọn máy in thành công và máy in đang trong tình trạng có thể thực hiện việc in thì hệ thống chuyển sang chức năng "Xác nhận in". Lúc này File sẽ được gửi về hàng chờ trên máy in và chờ được in.

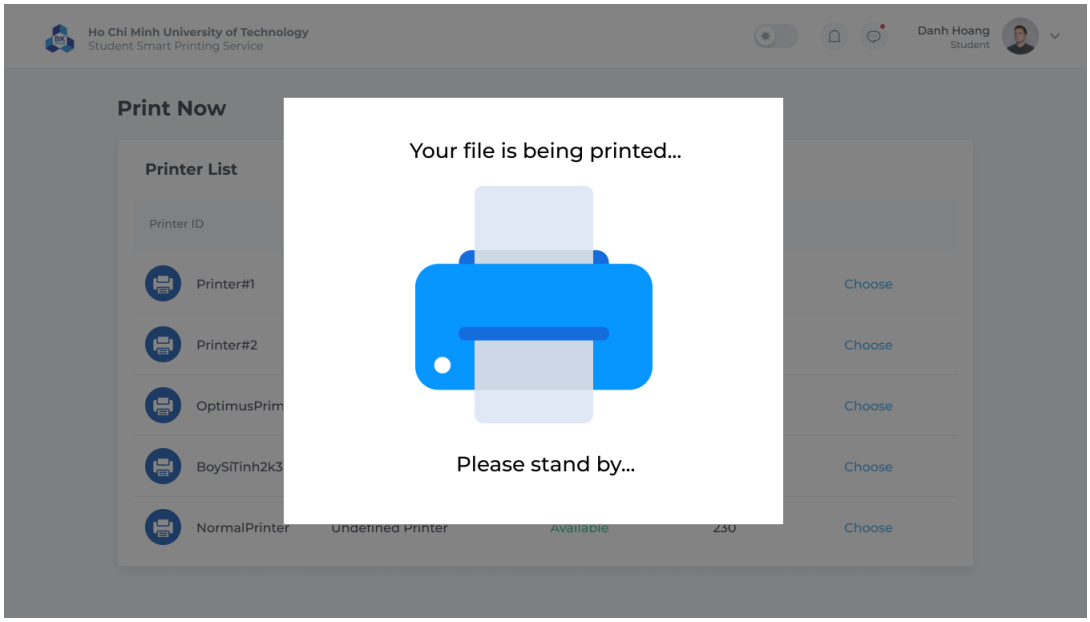


Figure 20: *UI Xác nhận in - Đang in*

Khi file được in xong thì hệ thống sẽ thông báo là đã in file thành công. Người dùng có thể nhấn nút để quay về trang chủ của SSPS.

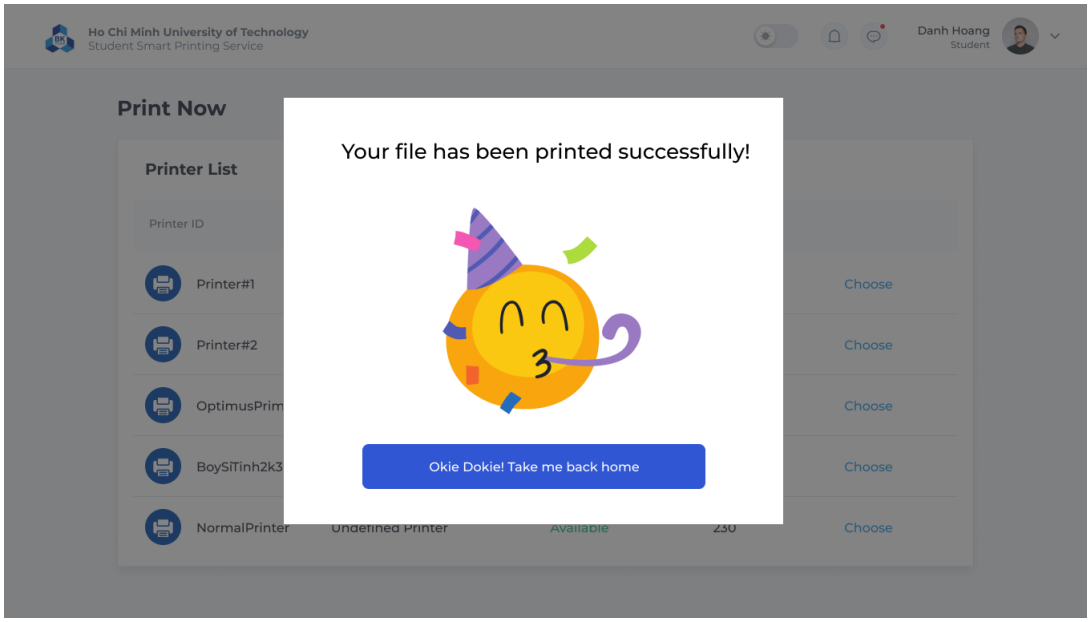


Figure 21: *UI Xác nhận in - In thành công*

2.4.3 Quản lý máy in

SPSO có thể nhấn vào nút "Management" để thực hiện chức năng quản lý máy in. Lúc này sẽ hiện ra danh sách các máy in đang có trong hệ thống cùng với các thông tin như ID, mẫu máy in, số lượng giấy còn lại, tình trạng kết nối và nút bấm chức năng "Settings".

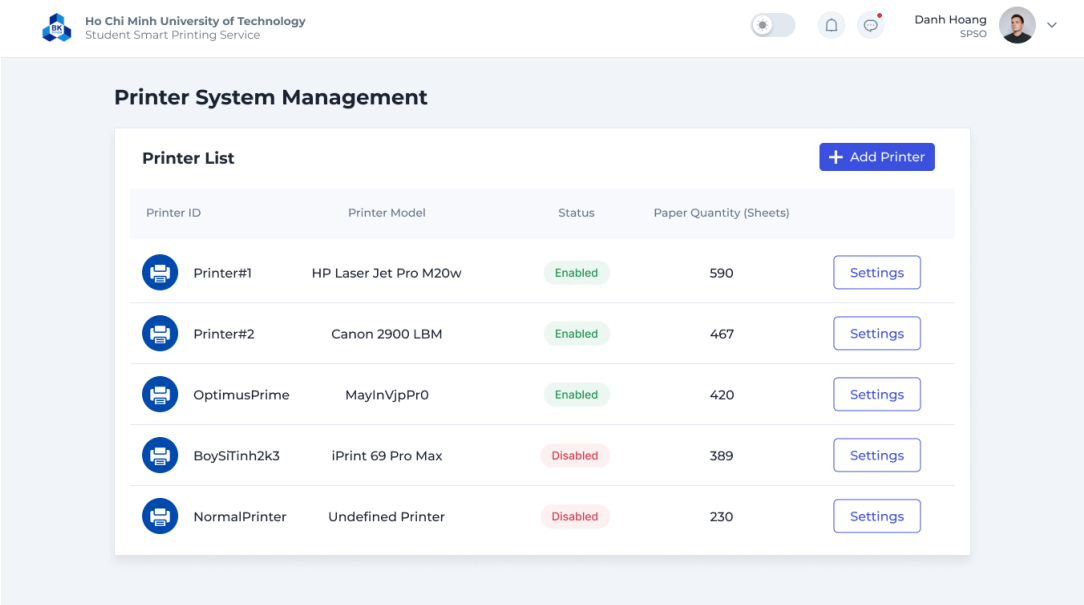


Figure 22: *UI Quản lý máy in - Danh sách máy in*

Khi SPSO nhấn vào nút "Settings" của một máy in thì sẽ hiện ra bảng thông tin của máy in đó cùng với các điều khiển quản lý máy in.

Lưu ý: Nút bấm màu xanh là cho phép bấm, nút bấm màu xám là không cho phép bấm.

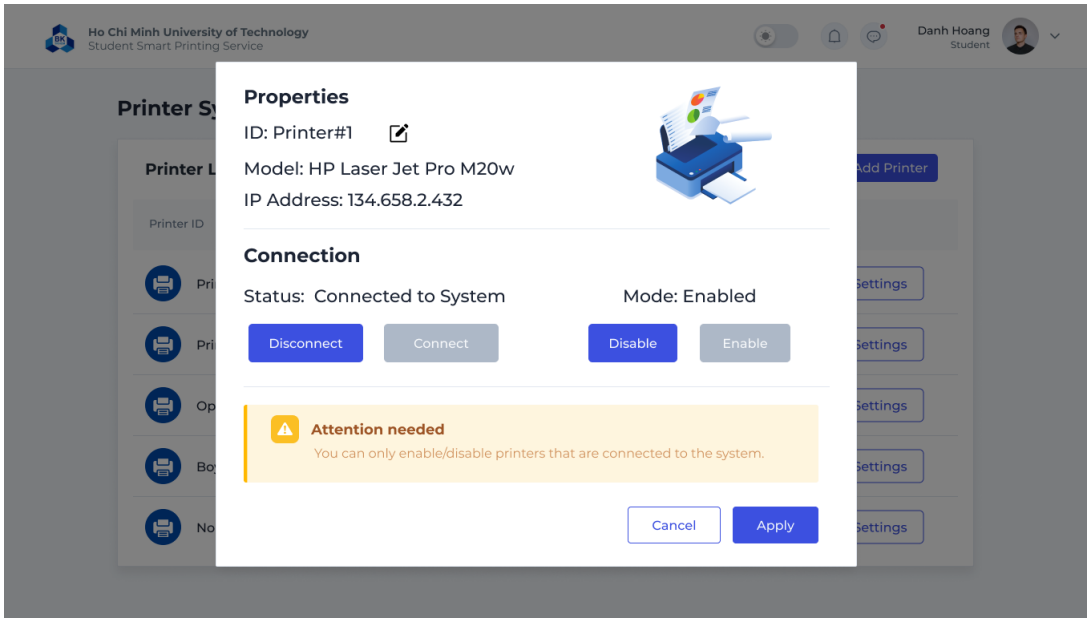


Figure 23: *UI Quản lý máy in - Cài đặt của máy in*

Ngoài ra, trong trường hợp không có máy in nào trong hệ thống thì sẽ hiện ra thông báo không có máy in. Lúc này thì SPSO cần phải sử dụng chức năng "Add Printer" để thêm máy in vào hệ thống.

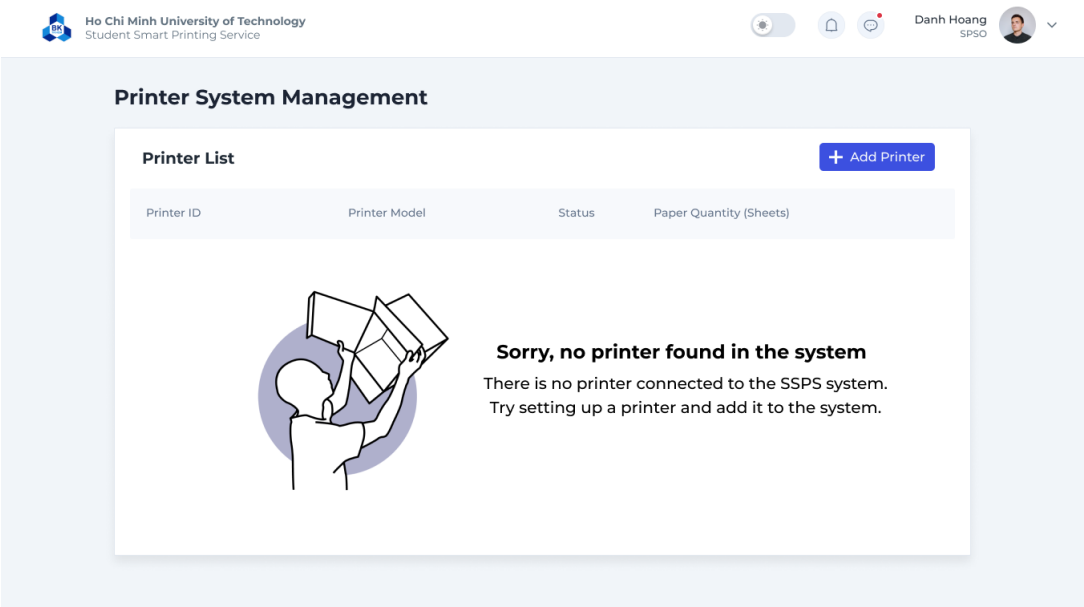


Figure 24: *UI Quản lý máy in - Không có máy in*

Sau khi SPSO nhấn vào nút "Add Printer" thì sẽ hiển thị danh sách các máy in đã được kết nối nhưng chưa được thêm vào hệ thống SSPS (được kết nối thông qua Wifi). Danh sách này chứa các thông tin như tên máy in, mẫu máy, địa chỉ IP của máy in. SPSO có thể chọn máy in bất kỳ để thêm vào hệ thống.

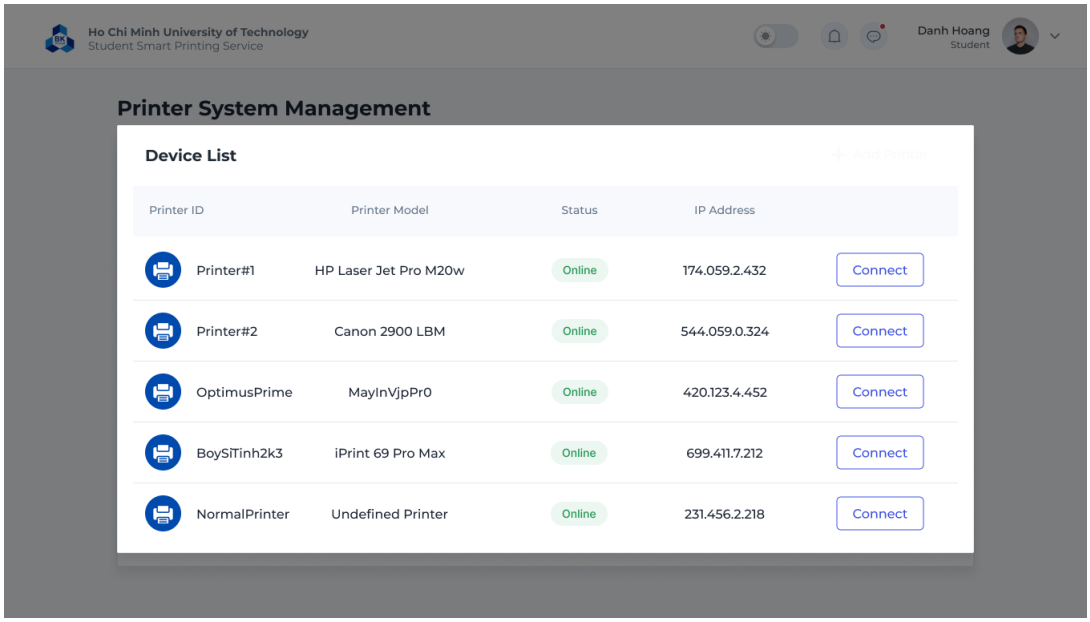


Figure 25: *UI Quản lý máy in - Thêm máy in*

2.4.4 Quản lý cài đặt in

SPSO có thể thay đổi các cài đặt in như số lượng giấy cấp định kỳ cho sinh viên, ngày cấp giấy định kỳ, các loại file được phép in. SPSO sử dụng chức năng này bằng cách nhấn vào nút "Settings" ở Homepage, sau đó hệ thống sẽ hiện ra một form với các thông tin để cho SPSO điều chỉnh.

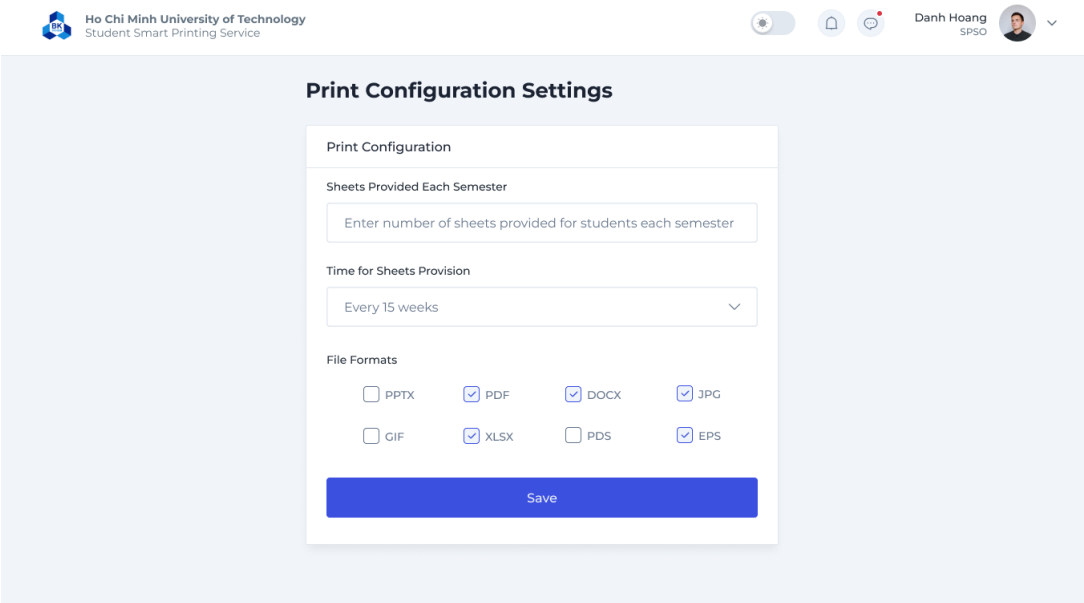


Figure 26: *UI Quản lý cài đặt in*

Chức năng của các nút/đường dẫn trong giao diện Quản lý cài đặt in được mô tả trong bảng sau:

Tên	Loại	Mô tả
Logo Bách Khoa	Link	Chuyển về trang chủ
Sheets Provided Each Semester	Input	Điền số lượng giấy được cấp cho sinh viên định kỳ
Time for Sheets Provision	Selection	Chọn thời gian để cấp giấy cho sinh viên
File Formats	Checkbox Button	Chọn các loại file sinh viên được phép tải lên
Save	Button	Lưu các thông tin cài đặt

2.4.5 Xem lịch sử in

Sinh viên và SPSO có thể xem lịch sử in của bản thân (đối với sinh viên) hoặc lích sử in của một sinh viên hoặc máy in bất kỳ (đối với SPSO) qua chức năng "History" ở Homepage.

Sinh viên chỉ có thể xem lịch sử in của bản thân trong một khoảng thời gian nào đó. Còn SPSO có thể nhập mã số sinh viên hoặc mã của máy in để xem lịch sử in của đối tượng đó.

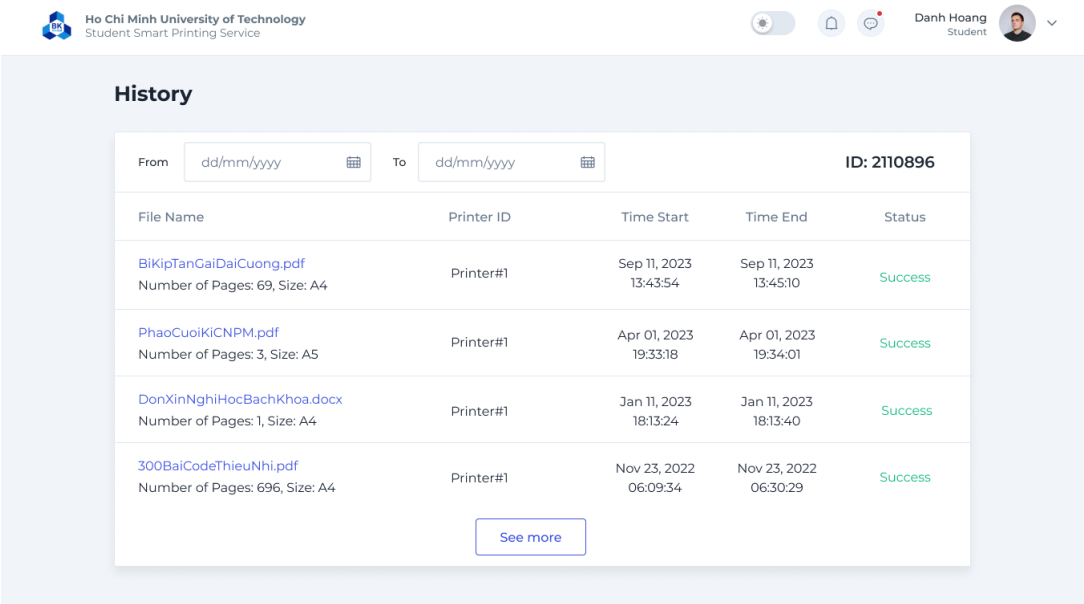


Figure 27: *UI Xem lịch sử in - Sinh viên*

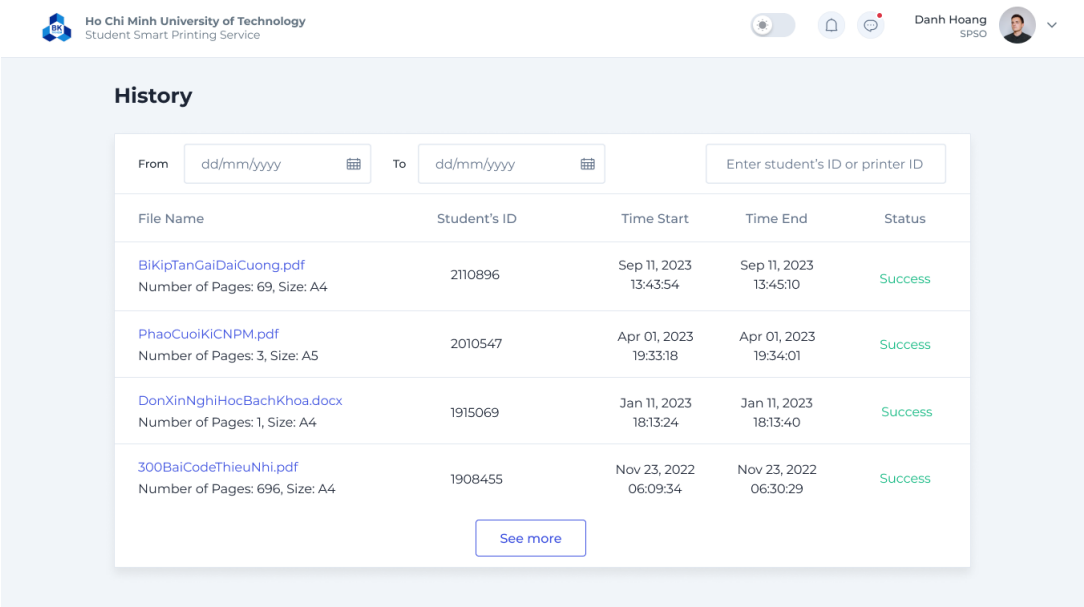


Figure 28: *UI Xem lịch sử in - SPSO*

2.4.6 Xem báo cáo định kỳ

SPSO có thể xem báo cáo định kỳ về lượng đăng nhập, số lượng giấy đã sử dụng, tổng số lần in qua chức năng "Reports" ở Homepage. Hệ thống sẽ trích xuất dữ liệu từ database để tạo ra các phân tích báo cáo ở trang Reports. SPSO có thể thay đổi báo cáo định kỳ là hàng tháng (monthly) hoặc là hàng năm (annual). Ngoài ra SPSO có thể lựa chọn tháng hoặc năm cụ thể để xem báo cáo.

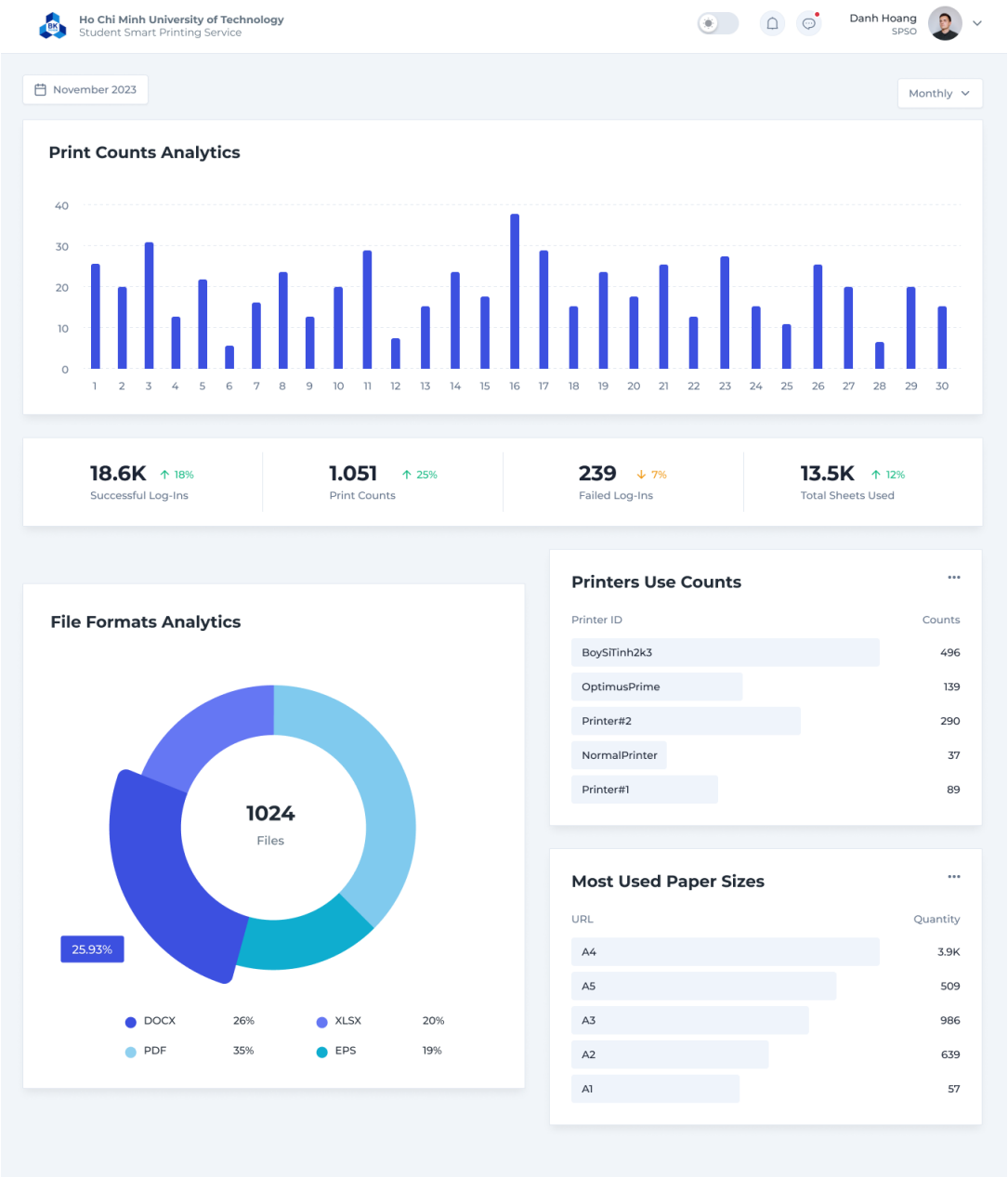


Figure 29: *UI Xem báo cáo định kỳ*

3 Architecture design

Task 3.1

Use a layered architecture to design the HCMUT-SSPS system. Describe how will you present your User Interface. Describe how will you store your data. Describe how you will access to external services/APIs.

3.1 Sử dụng kiến trúc phân lớp để thiết kế hệ thống HCMUT-SSPS

Để thực hiện hệ thống HCMUT-SSPS, nhóm quyết định xây dựng hệ thống dựa theo hướng tiếp cận mô hình kiến trúc MVC (Model - View - Controller). Đây là mô hình kiến trúc phân lớp rất phổ biến, nhóm lựa chọn kiến trúc này với một số lý do sau:

- Mỗi tính năng trong hệ thống SSPS thường đòi hỏi nhiều giao diện (View) và liên quan đến nhiều thao tác với dữ liệu. Ví dụ, tính năng cung cấp thông tin in đòi hỏi hiển thị thông tin in, quản lý máy in, và quản lý cài đặt in trên nhiều giao diện khác nhau. Mỗi giao diện này thường phải xử lý các thao tác như xem, chỉnh sửa, cập nhật dữ liệu tương ứng. Sử dụng kiến trúc MVC giúp tách biệt rõ ràng giữa các thành phần chính: Model, View, và Controller. Model chịu trách nhiệm về dữ liệu, View đảm nhiệm hiển thị thông tin cho người dùng, và Controller quản lý logic xử lý dữ liệu. Điều này giúp làm cho mã nguồn trở nên rõ ràng và dễ bảo trì.

- Sự tách biệt giữa Model, View và Controller cũng giúp cho việc mở rộng hệ thống trở nên dễ dàng hơn. Bất kỳ yêu cầu nào liên quan đến tương tác với dữ liệu có thể được thêm vào hệ thống mà không ảnh hưởng đến giao diện người dùng và ngược lại. Điều này cho phép tích hợp và mở rộng các tính năng mới một cách thuận tiện và độc lập.

Tuy nhiên, kiến trúc MVC cũng đi kèm với nhược điểm của sự phức tạp. Để đảm bảo sự tương tác hiệu quả giữa Model, View và Controller, phải đảm bảo rằng các phần tử này hoạt động cùng nhau một cách hợp nhất. Điều này đòi hỏi kiến thức kỹ thuật và quản lý dự án chặt chẽ để đảm bảo tính ổn định và bảo trì của hệ thống.

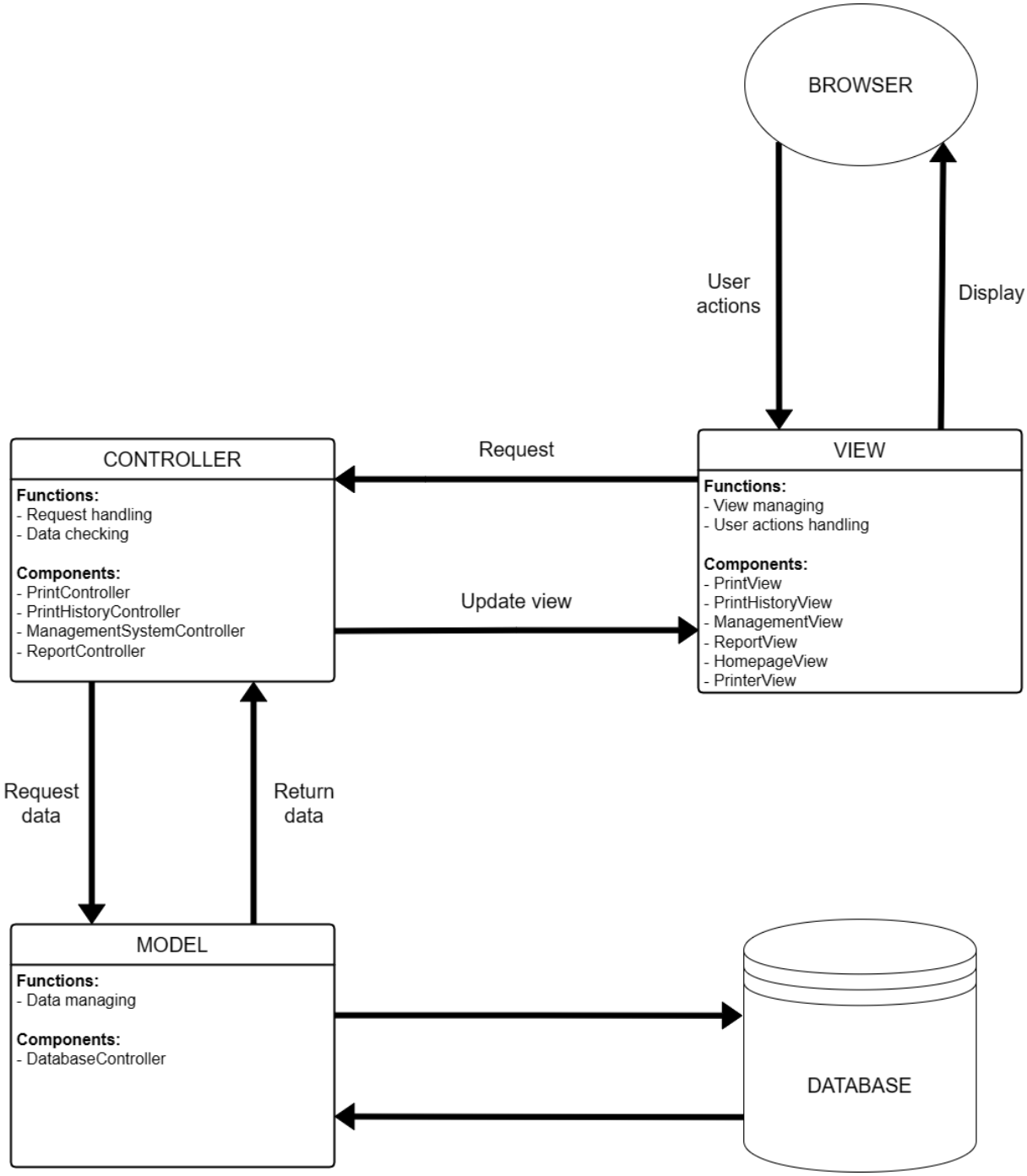


Figure 30: *Mô hình về kiến trúc MVC mà nhóm sẽ hiện thực trong hệ thống HCMUT-SSPS*

Xem ảnh chi tiết: Here!

Dưới đây là mô tả về kiến trúc MVC trong hệ thống mà nhóm sẽ hiện thực:

- **Browser:** Là nơi mà người dùng sẽ giao tiếp và sử dụng hệ thống thông qua giao diện. Các thông tin nhập xuất cũng sẽ được thu thập từ đây
- **View:** Là nơi quản lý phần nhìn (View managing) bao gồm các việc thiết lập giao diện cho toàn bộ hệ thống, thiết lập các thông báo, form hiện lên màn hình,... Bên cạnh đó, đây cũng là nơi tiếp

nhận các hành vi của người dùng (User actions) thực hiện trên browser khi họ tương tác với hệ thống; các hành động có thể là nhập dữ liệu, thay đổi dữ liệu, yêu cầu trích xuất để xem dữ liệu,... Khi này các hành động đó sẽ được quản lý và giao cho lớp **Controller** giải quyết thông qua các dạng yêu cầu từ lớp **View**; các kết quả trả về cho từng yêu cầu sẽ lại được **View** tiếp nhận và cập nhật phần nhìn tùy theo hành động mà người dùng đã thực hiện

- **Controller:** Đây là lớp xử lý chính của hệ thống với nhiệm vụ làm kênh trung gian liên lạc giữa lớp **View** và **Model**. Các chức năng quan trọng của lớp này có thể kể đến là:
  - *Xử lý yêu cầu (Request handling):* Khi này lớp **Controller** sẽ tiếp nhận các yêu cầu từ lớp **View**, thực hiện trích xuất dữ liệu cần cho yêu cầu đó từ lớp **Model** và xử lý yêu cầu đó trên dữ liệu vừa được trích xuất. Kết quả sau khi xử lý được trả về lại cho lớp **View** để thực hiện cập nhật phần nhìn. Với chức năng này, lớp **Controller** chứa các phương thức xử lý chính của toàn hệ thống
  - *Kiểm tra dữ liệu (Data checking):* Khi này lớp **Controller** sẽ tiến hành kiểm tra tính hợp lệ của dữ liệu cho các thao tác nhận được từ yêu cầu của lớp **View**; các dữ liệu này có thể phục vụ cho nhiều yêu cầu bao gồm các yêu cầu cần trích xuất dữ liệu cũng như lưu dữ liệu được cập nhật bởi người dùng vào database
- **Model:** Là nơi quản lý dữ liệu (Data managing) bao gồm các chức năng cơ bản liên quan đến xử lý dữ liệu. Ở hệ thống HCMUT-SSPS này, lớp **Model** sẽ là phần giao tiếp chính với database - nơi lưu trữ dữ liệu của hệ thống. Lớp này tiếp nhận các cập nhật dữ liệu từ lớp **Controller** đến database và thực hiện nó cũng như trích xuất dữ liệu được yêu cầu

3.2 Mô tả cách trình bày giao diện người dùng

Trang web HCMUT-SSPS là một nơi cung cấp giao diện cho người sử dụng để có thể thao tác và sử dụng các dịch vụ của hệ thống HCMUT-SSPS từ xa cách tiện lợi và nhanh chóng nhất. Do đó, việc thiết kế giao diện hệ thống phải đảm bảo được các tiêu chí về thẩm mĩ, thân thiện với người dùng, trực quan và dễ thao tác để người dùng có thể sử dụng dịch vụ một cách hiệu quả:

- Đầu tiên khi truy cập vào trang web của hệ thống, người sử dụng sẽ được điều hướng về trang chủ. Trang chủ ở đây chỉ thể hiện hình ảnh và thông điệp chào đón, để có thể sử dụng các dịch vụ của hệ thống trên vai trò của sinh viên hay của SPSO, người sử dụng phải đăng nhập bằng cách bấm vào nút đăng nhập được đặt trên trang chủ.
- Sau khi bấm vào nút đăng nhập, người sử dụng sẽ được chuyển hướng sang một trang mới phục vụ quá trình đăng nhập. Khi này người sử dụng sẽ nhập thông tin tài khoản bao gồm tên đăng nhập và mật khẩu vào khu vực mà giao diện cung cấp; thông báo lỗi sẽ hiển thị khi người dùng đăng nhập không thành công, ngược lại, người dùng sẽ có khả năng truy cập đến với các dịch vụ của hệ thống.
- Khi đã đăng nhập thành công, người dùng sẽ được chuyển hướng đến trang hoạt động chính của hệ thống. Ở trang này, phần truy cập thông tin người dùng sẽ được đặt trên trang hoạt động chính và nổi bật ở chính giữa trang chính là các chức năng dịch vụ mà người sử dụng có thể chọn để có thể sử dụng bao gồm **In, Xem lịch sử in** nếu vai trò đăng nhập là sinh viên và **In, Xem lịch sử in, Xem báo cáo định kỳ, Cài đặt in** và **Quản lý hệ thống** nếu vai trò đăng nhập là SPSO. Bằng cách thể hiện các icon lớn đi đôi với tên các chức năng cố định, người dùng có thể dễ dàng

nhấp vào bất cứ icon nào mà không lẫn lộn để thực hiện chức năng mà mình cần. Khi nhấp chọn chức năng nào đó, người dùng sẽ được chuyển hướng đến trang riêng phục vụ cho chức năng đó.

- Ở mỗi trang riêng phục vụ cho mỗi chức năng, các thông tin cần thiết đều được cung cấp đầy đủ, ở vị trí dễ nhìn và đảm bảo tính bao quát thông tin cho người dùng. Đối với các chức năng bắt phải nhập dữ liệu, các vị trí cần nhập các dữ liệu cần thiết đều được cung cấp đầy đủ ở các vị trí dễ nhìn cho người dùng nhập vào, bên canh đó giao diện còn cung cấp chức năng kiểm tra dữ liệu nhập vào của người dùng để đưa ra các thông báo lỗi kịp thời. Với mỗi hoạt động khi thực hiện thao tác xác nhận, các thông báo lỗi và thành công đều được cung cấp để người dùng có thể biết được tiến độ và tình trạng công việc mà mình thực hiện.
  - Đối với trang Xem báo cáo định kỳ, giao diện người dùng cung cấp các bảng điều khiển thống kê và biểu đồ trực quan, giúp SPSO có cái nhìn tổng quan về hoạt động của hệ thống.
  - Các chức năng quản lý, như cài đặt in, quản lý máy in được thiết kế một cách chi tiết và dễ sử dụng.
    * Đối với chức năng Quản lý cài đặt in, SPSO có thể thay đổi các cài đặt in như số lượng giấy cấp định kỳ cho sinh viên, ngày cấp giấy định kỳ, các loại file được phép in.
    * Đối với chức năng Quản lý máy in, SPSO sẽ được thấy danh sách các máy in đang có trong hệ thống cùng với các thông tin như ID, mẫu máy in, số lượng giấy còn lại, tình trạng kết nối. SPSO có thể quản lý máy in bằng cách nhấn vào nút Settings bên cạnh máy in cần quản lý. SPSO cũng có thể thêm máy in mới vào hệ thống bằng cách chọn chức năng "Add Printer".
  - Đối với chức năng xem lịch sử in, để nâng cao sự thuận tiện và tương tác, giao diện người dùng của HCMUT-SSPS cũng tích hợp các tính năng tiên tiến như tìm kiếm thông minh và bộ lọc.
    * Đối với sinh viên, chức năng bộ lọc giúp họ có thể giúp họ tìm kiếm lịch sử in của mình trong khoảng thời gian xác định từ ngày nào đến ngày nào.
    * Đối với SPSO, ngoài chức năng bộ lọc như dành cho sinh viên thì còn có chức năng xem lịch sử in của sinh viên bằng cách tìm kiếm theo mã số sinh viên của sinh viên đó hoặc xem lịch sử in của máy in bằng cách tìm kiếm theo mã máy in.
- Các danh sách máy in cũng như danh sách lịch sử in được tối ưu để chỉ hiển thị 4 đến 5 dòng trên màn hình được sắp xếp thứ tự ưu tiên các máy in đang hoạt động đối với danh sách máy in và ưu tiên các lịch sử in gần đây nhất đối với danh sách lịch sử in.
- Để tối ưu hóa trải nghiệm người dùng, giao diện người dùng được thiết kế có thể tương tác trên nhiều thiết bị khác nhau, bao gồm cả máy tính, máy tính bảng, và điện thoại di động. Giao diện linh hoạt và tự động điều chỉnh kích thước để phản ánh đúng trên mọi loại màn hình, đảm bảo rằng người dùng có thể trải nghiệm mượt mà và thuận lợi mọi lúc, mọi nơi.

3.3 Mô tả cách lưu trữ dữ liệu

Để lưu trữ dữ liệu cho hệ thống HCMUT-SSPS, cần sử dụng một hệ thống quản lý cơ sở dữ liệu mạnh mẽ và linh hoạt. Trong số các phần mềm quản lý cơ sở dữ liệu hiện nay, MySQL là một lựa chọn hợp lý để thực hiện vấn đề này. Các ưu điểm của MySQL có thể kể đến như:

- **Hiệu suất cao:** MySQL được tối ưu hóa để cung cấp hiệu suất cao cho các truy vấn và ghi dữ liệu. Điều này là quan trọng khi có nhu cầu xử lý lớn lượng dữ liệu từ các máy in và người dùng.

- **Tính đồng nhất và Tính nhất quán:** MySQL tuân thủ ACID (Atomicity, Consistency, Isolation, Durability), đảm bảo tính nhất quán và đồng nhất của dữ liệu trong mọi trạng thái.
- **Hỗ trợ truy vấn phức tạp:** MySQL hỗ trợ ngôn ngữ truy vấn SQL mạnh mẽ, cho phép thực hiện các truy vấn phức tạp để truy xuất, cập nhật và xử lý dữ liệu.
- **Tính dễ mở rộng:** MySQL có thể được mở rộng dễ dàng để đáp ứng nhu cầu tăng cường về dữ liệu.
- **Tích hợp tốt với nhiều ngôn ngữ Lập trình:** MySQL tương thích với nhiều ngôn ngữ lập trình phổ biến như PHP, Python, Java, giúp việc phát triển ứng dụng web trở nên thuận tiện.
- **An toàn và bảo mật:** MySQL cung cấp các tính năng bảo mật mạnh mẽ như kiểm soát quyền truy cập, mã hóa dữ liệu và theo dõi hoạt động.
- **Hỗ trợ nhiều hệ điều hành:** MySQL có sẵn cho nhiều hệ điều hành, từ Linux đến Windows, tạo điều kiện thuận lợi cho triển khai đa nền tảng.
- **Tính linh hoạt và tùy chỉnh:** MySQL cho phép tùy chỉnh cấu hình theo nhu cầu cụ thể của dự án, giúp tối ưu hóa hiệu suất hệ thống.
- **Hỗ trợ đa luồng:** MySQL cho phép các tác vụ đa luồng theo nhu cầu của người sử dụng. Ví dụ như cho phép một file in trên nhiều máy in hoặc một máy in có thể in nhiều file tuần tự.

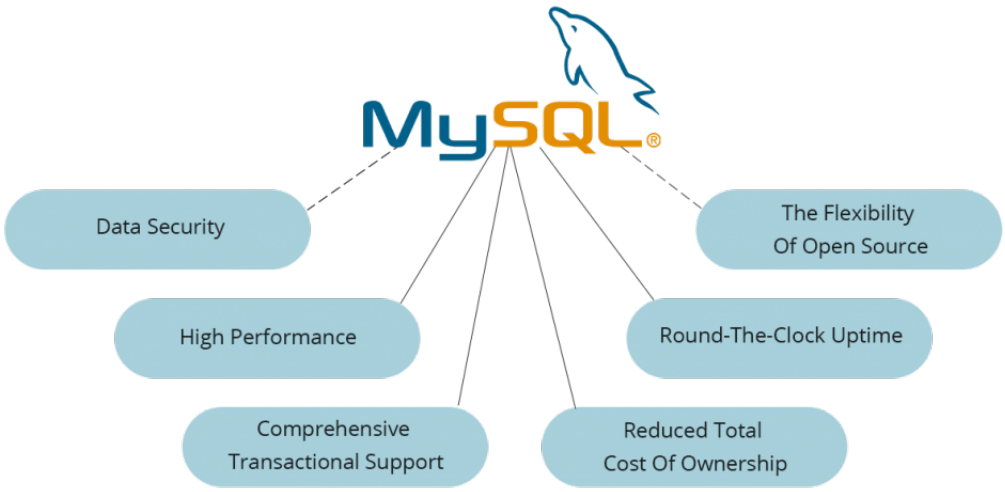


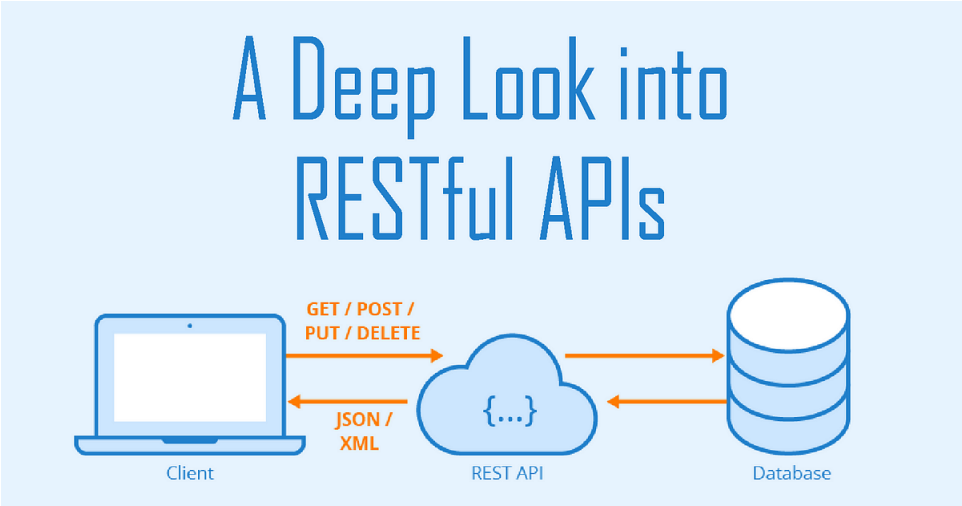
Figure 31: *Các ưu điểm của MySQL*

Hệ thống cơ sở dữ liệu cho trang web in ấn cần bao gồm các bảng chính để lưu trữ thông tin về máy in, người dùng, và các tài liệu in. Dưới đây là một số bảng cơ sở dữ liệu quan trọng:

- **Bảng Máy In (Printers):** Mã máy in (Printer ID), tên máy in, thông tin kết nối (địa chỉ IP, cổng,...), số lượng giấy còn lại, số lần sử dụng in.
- **Bảng Người Dùng (Users):** Tên đăng nhập, mật khẩu, quyền truy cập, số lượng giấy còn lại của người dùng, các thông tin khác liên quan đến tài khoản người dùng.
- **Bảng Tài Liệu In (Printed Documents):** Mã tài liệu, tên tài liệu, người dùng đã in, máy in sử dụng, ngày in, số lần in, số lượng giấy đã in.

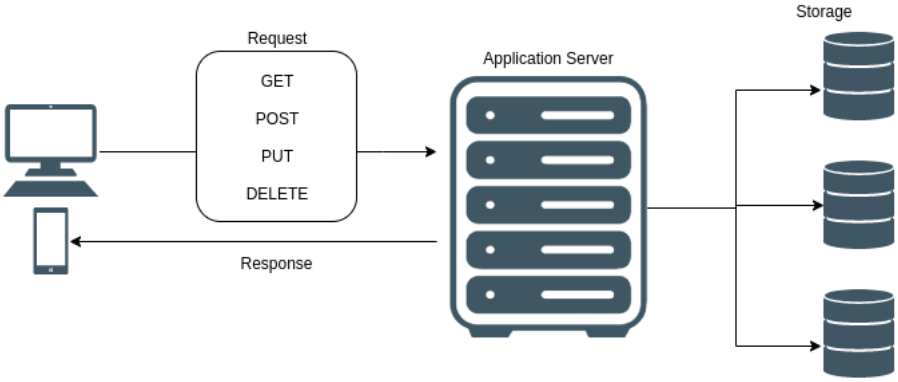
3.4 Mô tả cách truy cập vào các dịch vụ/API bên ngoài

Về mặt kỹ thuật, API (Application Programming Interface) là viết tắt của Giao diện Lập trình Ứng dụng, đó là một cơ chế trung gian phần mềm cho phép hai ứng dụng khác nhau tương tác và trao đổi dữ liệu với nhau. API có thể được sử dụng trong nhiều ngữ cảnh, bao gồm các hệ thống web-based, hệ điều hành, hệ thống cơ sở dữ liệu, phần cứng máy tính, hoặc thư viện phần mềm. Ở dạng đơn giản nhất, API cung cấp một giao diện cho phép một ứng dụng tương tác với ứng dụng khác thông qua các yêu cầu và phản hồi, và cách mà dữ liệu được truy cập thông qua API có thể khác nhau tùy thuộc vào loại API, chẳng hạn như **API SOAP** hoặc **REST**.



REST (Representational State Transfer) là một kiểu kiến trúc được sử dụng để thiết kế các API. Nó sử dụng các phương thức HTTP đơn giản như GET, POST, DELETE, và PUT để tạo giao tiếp giữa các máy. Thay vì sử dụng một URL cho việc xử lý một số thông tin người dùng, REST gửi yêu cầu HTTP đến một URL cụ thể để thực hiện các thao tác trên dữ liệu.

Cuộc sống của HTTP dựa vào quy trình luân phiên giữa Yêu cầu (Request) và Phản hồi (Response). Khách hàng gửi một yêu cầu, máy chủ phản hồi để xác định liệu máy chủ có thể thực hiện yêu cầu đó hay không. API được xây dựng dựa trên hai thành phần chính này: Yêu cầu và Phản hồi.



Yêu cầu (Request) đúng chuẩn cần bao gồm bốn phần chính:

- **URL:** Đây là địa chỉ duy nhất cho tài nguyên hoặc dịch vụ mà yêu cầu được gửi đến. Ví dụ, nó có thể là địa chỉ của một trang web, hình ảnh hoặc video. API mở rộng ý tưởng URL ban đầu để áp dụng cho các tài nguyên khác như danh sách khách hàng, sản phẩm, và những thứ tương tự. Các tài nguyên này thường được gọi là "resources" - nguồn lực.
- **Method:** Đây là hành động mà khách hàng muốn thực hiện trên tài nguyên hoặc dịch vụ. Thường, method thường là các động từ và bao gồm bốn loại phổ biến: GET (yêu cầu máy chủ trả về tài nguyên), POST (yêu cầu máy chủ tạo ra một tài nguyên mới), PUT (yêu cầu máy chủ sửa đổi hoặc bổ sung thông tin vào một tài nguyên đã tồn tại), DELETE (yêu cầu máy chủ xóa một tài nguyên).
- **Headers:** Đây là nơi chứa các thông tin cần thiết cho yêu cầu, mà end-users thường không cần biết. Ví dụ, nó có thể chứa độ dài của dữ liệu yêu cầu, thời gian gửi yêu cầu, loại thiết bị đang sử dụng, và định dạng phản hồi mà khách hàng có thể đọc.
- **Body:** Đây là nơi chứa dữ liệu mà khách hàng muốn gửi đến máy chủ. Ví dụ, nếu bạn đặt một chiếc bánh pizza, thì phần nội dung trong yêu cầu sẽ chứa thông tin như loại bánh pizza, kích cỡ, số lượng đặt.

Phản hồi (Response) là phản hồi từ máy chủ sau khi xử lý yêu cầu từ khách hàng. Cấu trúc của một phản hồi tương tự như yêu cầu, bao gồm:

- **Status code:** Đây là mã trạng thái mô tả kết quả của yêu cầu. Mã này thể hiện liệu yêu cầu đã thành công, bị từ chối hoặc có lỗi gì đó.
- **Headers:** Chứa thông tin quan trọng liên quan đến phản hồi, giống như phần headers của yêu cầu.
- **Body:** Chứa dữ liệu hoặc thông tin mà máy chủ gửi lại cho khách hàng sau khi xử lý yêu cầu.

Việc **quản lý APIs** liên quan đến việc tạo, xuất bản và quản lý các kết nối API trong một doanh nghiệp hoặc môi trường đám mây. Nó không chỉ là cách để kết nối các API lại với nhau, mà còn là một nền tảng cho phép doanh nghiệp chia sẻ và quản lý cấu hình API trong khi vẫn kiểm soát quyền truy cập, thu thập và phân tích thống kê sử dụng, và thực hiện các chính sách bảo mật liên quan.

Có nhiều **công cụ quản lý API** khác nhau, trong đó một số ví dụ như JMeter, một công cụ mã nguồn mở ban đầu thiết kế để kiểm tra hiệu suất phần mềm, và các nền tảng khác nhau được sử dụng để quản lý và kiểm soát quyền truy cập vào các dịch vụ và tài nguyên qua API.

Ngoài ra, có bốn loại chính của **web APIs** là Open APIs, Partner APIs, Internal APIs và Composite APIs, trong đó Open APIs thường được sử dụng rộng rãi. Open APIs thường là các API

mã nguồn mở có thể được truy cập thông qua giao thức HTTP. Chúng xác định API endpoints, định dạng yêu cầu và phản hồi, và được sử dụng để chia sẻ dữ liệu và chức năng từ các ứng dụng và dịch vụ khác.

Task 3.2

Draw a component diagram for the important module(s) chosen in Task 1.3.

3.5 Component Diagram cho các Module quan trọng

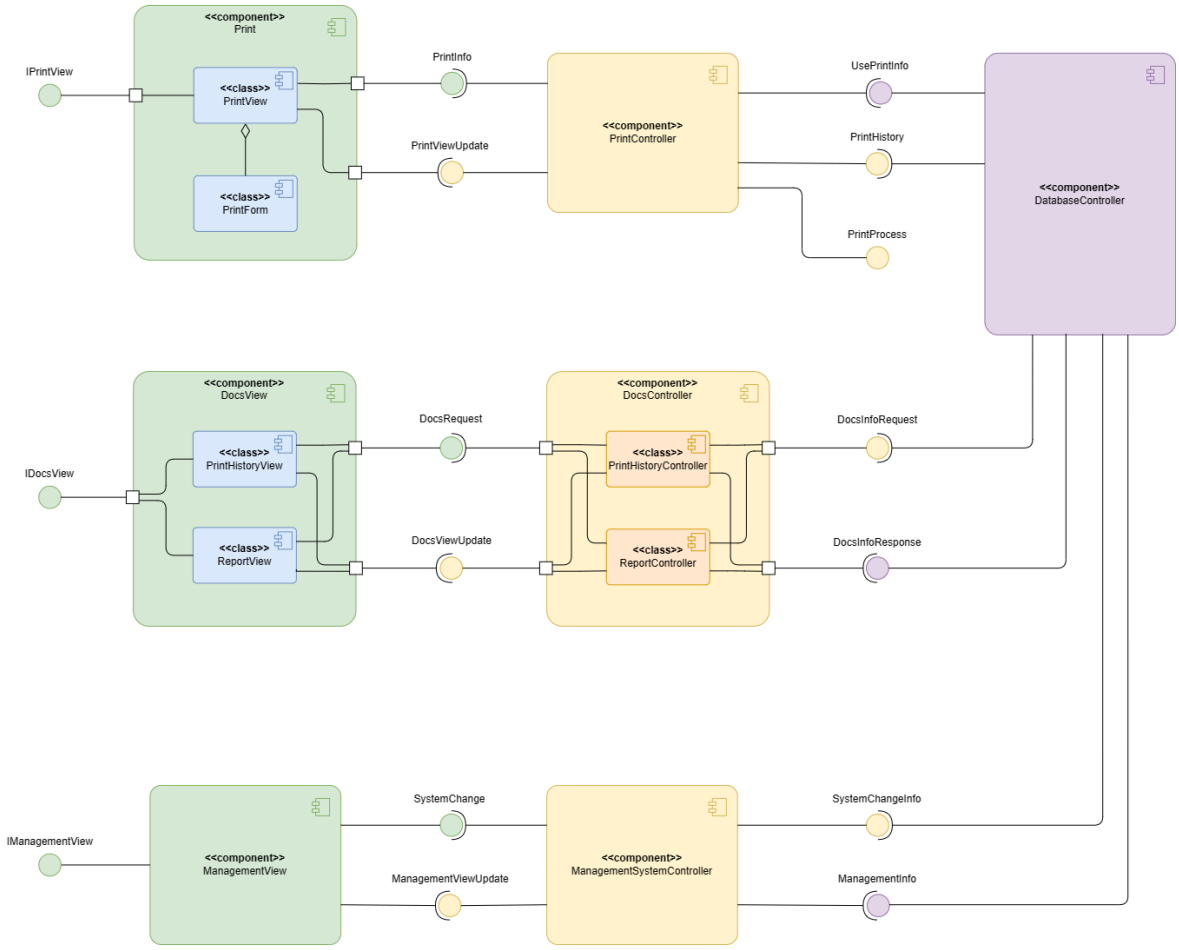


Figure 32: *Component Diagram cho các Module quan trọng*

Xem chi tiết Diagram: Here!

Mô tả Component Diagram:

1. **Tầng View**

   (a) **Print**

   - **Chức năng chính:** hiện thực các class quản lý giao diện cho việc hỗ trợ in ấn.
   - **Gồm các class: PrintView, PrintForm**. Class **PrintView** cung cấp giao diện trang thực hiện các thao tác upload file, chọn máy in và in ấn. Class **PrintForm** cung cấp giao diện form để điền thông tin về thông số in vào.

- **Mối quan hệ giữa các thành phần:** Class **PrintView** dựa vào các thông tin nhập vào từ **PrintForm** để nhận các giá trị về thông số in sau đó gửi cho controller xử lý.
- **PrintView** hiện thực interface **iPrintView**, **PrintInfo**.
- **Toàn bộ component** yêu cầu interface **PrintViewUpdate**.

(b) **DocsView**

- **Chức năng chính**: Hiện thực các class cung cấp giao diện xem lịch sử in và báo cáo định kỳ.
- **Gồm các class: PrintHistoryView, ReportView**. Class **PrintHistoryView** cung cấp giao diện xem lịch sử in của sinh viên (và máy in nếu đăng nhập SPSO). Class **ReportView** cung cấp giao diện xem báo cáo định kỳ.
- Cả component hiện thực interface **IDocsView**, **DocsRequest**.
- Cả component yêu cầu interface **DocsViewUpdate**.

(c) **MangamentView**

- **Chức năng chính:** Hiện thực giao diện quản lý máy in cho SPSPO.
- Cả component hiện thực interface **IManagementView**, **SystemChange**.
- Cả component yêu cầu interface **ManagementViewUpdate**.

2. **Tầng Controller**

(a) **PrintController**

- **Chức năng chính:** hiện thực class cung cấp phương thức xử lý sự kiện và thông tin về thông số in người dùng nhập vào, tài khoản số dư in, quá trình in.
- Cả **Component** hiện thực interface **PrintViewUpdate**, **PrintHistory**, **PrintProcess**.
- Cả component yêu cầu interface **UsePrintInfo**.

(b) **DocsController**

- **Chức năng chính:** hiện thực các class cung cấp phương thức xử lý các mốc thời gian (hoặc thêm mã máy in nếu là SPSO) người dùng nhập vào.
- **Gồm các class: PrintHistoryController, ReportController**. Class **PrintHistoryController** cung cấp phương thức xử lý thông tin về mốc thời gian để xem lịch sử in sinh viên (và của máy in nếu là SPSO). Class **ReportController** cung cấp phương thức xử lý thông tin mốc thời gian để xem báo cáo định kỳ.
- Cả component hiện thực interface **DocsViewUpdate** và **DocsInfoRequest**.
- Cả component yêu cầu interface **DocsRequest**, **DocsInfoResponse**.

(c) **ManagementSystemController**

- **Chức năng chính:** Hiện thực class cung cấp phương thức xử lý các luồng thông tin yêu cầu cập nhật thông tin về quản lý máy in.
- **Component** hiện thực các interface **ManagementViewUpdate**, **SystemChangeInfo**.
- **Component** yêu cầu các interface **SystemChange**, **ManagementInfo**.

3. **Tầng Model**

(a) **DatabaseController**

- **Chức năng chính:** Quản lý thông tin yêu cầu từ các controller và thực hiện truy vấn lên database để lấy thông tin, hiệu chỉnh thông tin.

- **Component** hiện thực các interface **UserPrintInfo**, **DocsInfoResponse**, **ManagementInfo**.
- **Component** yêu cầu các interface **PrintHistory**, **DocsInfoRequest**, **ManagementInfo**.

Bảng mô tả các interface:

Interface	Chức năng	Phương thức

iPrintView	Cung cấp giao diện hiển thị
	những thông số in được nhập vào
	và gửi cho controller để xử lý
		renderPrintView(printInfo);
		getInfoFromForm(): printInfo;
		displayConfirmButton() void

PrintInfo	Cung cấp giao diện form
	để điền thông tin thông số in vào.
		displayPrintForm();
		inputPageNumber();
		inputOneOrTwoSide();
		inputNumebrOfCopy();
		inputPaperSize();

PrintViewUpdate	Cung cấp giao diện trang
	thực hiện các thao tác upload file,
	chọn máy in và in ấn.
		renderPrintViewUpdate(printInfo);
		requestUploadFile();
		choosePrinter();
		displayPrinter();

IDocsView	Cung cấp giao diện hiển thị
	những thông số được nhập vào
		setTimeSpan(time: string);
		setStudentInfo(studentID: string)
		setPrinterInfo(pritnerID: string);
		generateAnnualReport(): string;;
		generateMonthlyReport(): string

DocsRequest	Cung cấp các phương thức
	nhập các thông số xem lịch sử in
	và báo cáo định kì.
		inputTimeSpan();
		viewStudentOrPrinterHistory(): bool;
		chooseStudent(studentID);
		choosePrinter(printerID);

DocsViewUpdate	Cung cấp giao diện xem lịch sử in của sinh viên (và máy in nếu đăng nhập SPSO).	displayReport(time: string); displayHistoryView();

IManagementView	Cung cấp giao diện hiển thị
	những thay đổi và gửi cho controller
		getChanges(info);
		displaySettingsTable();
		displayButton(button_type)

SystemChange	Cung cấp các phương thức
	hiện thực tác vụ thay đổi hệ thống.
		printSettingManagement();
		changeDefaultNumOfPage();
		changeValidFileType();
		changeSupplyDate();

Interface	Chức năng	Phương thức
ManagementViewUpdate	Cung cấp giao diện quản lí máy in cho SPSO.	renderManagementView();

PrintHistory	Cung cấp giao diện hiển thị thông tin về lịch sử in.	displayHistoryView();
		displayStudentHistory(student_history);
		displayStudentHistory(pritner_history);
		displayStudentHistory();

PrintProcess	Cung cấp phương thức xử lý sự kiện về quá trình in.	print();
		checkPrintingSuccess();
		calculateNumberOfPages(printInfo): int
		calculateRemainPaper();
		getPrinter(printer_id: string);

UsePrintInfo	Cung cấp phương thức xử lý sự kiện và thông tin về thông số in người dùng nhập vào.	renderUserPrintInfo();

DocsInfoRequest	Cung cấp phương thức nhập vào các mốc thời gian do người dùng nhập vào.	setTimeSpan(time:string);
		setStudentInfo(studentID: string);
		setPrinterInfo(studentID: string);

DocsInforResponse	Cung cấp giao diện hiển thị thông tin về mốc thời gian xem lịch sử in sinh viên.	getReportID(time: string); displayReport(report: string);
SystemChangeInfo	Cung cấp giao diện hiển thị những thông tin về cập nhật những thay đổi của hệ thống.	displaySystemChangeInfo();
ManagementInfo	Cung cấp giao diện hiển thị những thông tin về quản lý hệ thống.	getManagementInfo(); displayManagementInfo();

3.6 Deployment Diagram cho các module quan trọng

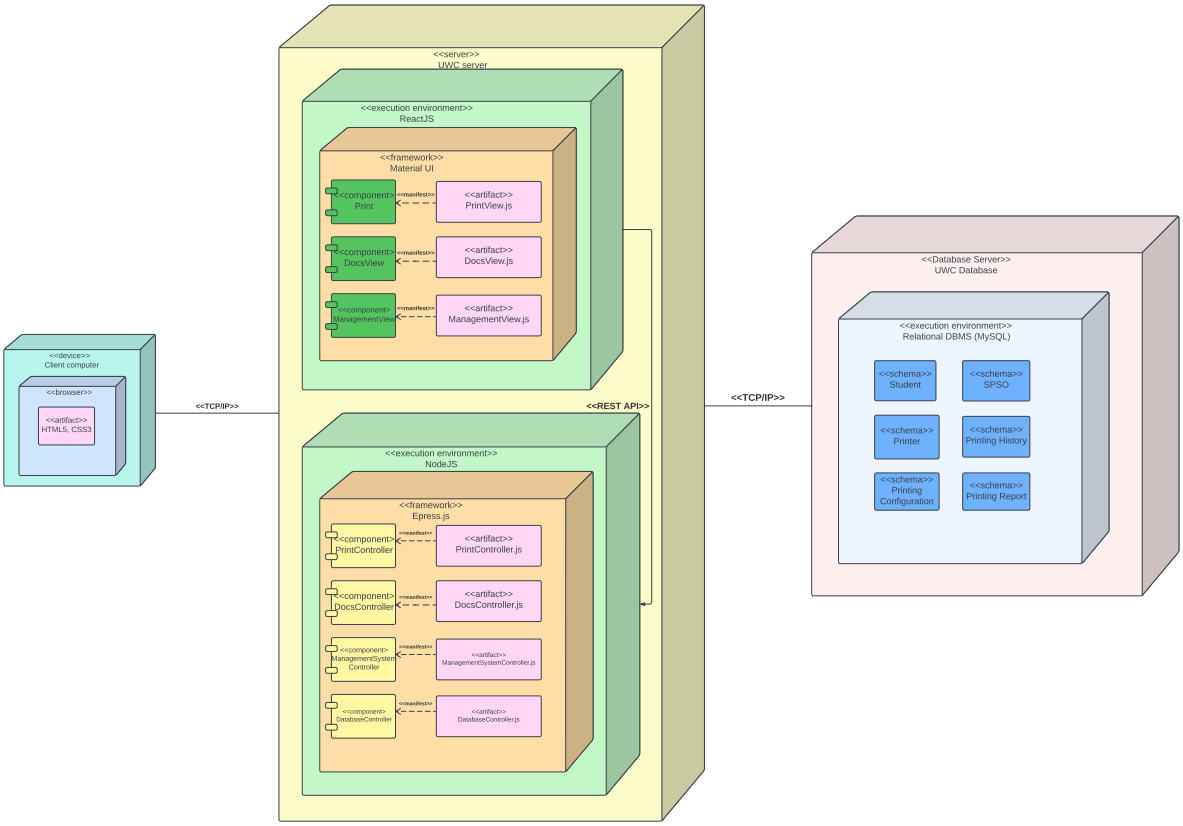


Figure 33: *Deployment Diagram cho các Module quan trọng*

Xem bản vẽ của Deployment Diagram của nhóm (PDF) tại đây: Google Drive

3.7 Mô tả Deployment Diagram

- Hệ thống là một ứng dụng Web, được phát triển theo kiến trúc MVC, bảo gồm 3 máy chủ chính: Client, Server và Database Controller (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 lên browser website tại máy tính người dùng (Client) thông qua trình đọc HTML3 và CSS3 được nhúng trong trình duyệt.

- Máy chủ Server bao gồm môi trường thực thi (execution environment) các module chức năng chính của hệ thống. Môi trường thực thi phía **back-end** là *NodeJS*, sử dụng Framework là *Express.js*, bao gồm 4 component chính là **PrintController**, **DocsController**, **ManangementSystem Controller**, **DatabaseController** đảm nhận các tác vụ xử lý chính và điều khiển luồng dữ liệu liên quan đến xử lý in, xem lịch sử in và báo cáo. Môi trường thực thi phái **front-end** là *ReactJS*, sử dụng Framework là *Material UI*, bao gồm 3 component chính là **Print**, **DocsView**, **ManagementView** đảm nhận việc hiện thực các giao diện người dùng liên quan thực hiện in, xem báo cáo, và quản lý máy in. Các component sẽ được hiện thực từ các file mã nguồn JavaScript tưtuwonuwngs. Thông qua các REST API, front-end và back-end có thể giao tiếp được với nhau để cập nhật, hiện thị giao diện và nhận phản hồi các tác vụ người dùng.

- Máy chủ Server sẽ kết nối với Database Server thông qua giao thức TCP/IP, để truy xuất dữ liệu hiển thị giao diện hoặc cập nhật lại dữ liêu mỗi khi người dùng tương tác với ứng dụng. Sau đó, dữ liệu sau khi được cập nhật sẽ được truyền qua giao thức TCP/IP sang Servẻ và truyền sang máy Client để cập nhật lại giao diện

- Máy chủ Database bao gồm môi trường thực thi là DBMS MySQL. Trong phạm vi module thực hiện in, xem lịch sử và quản lý máy in, máy chủ Database lưu trữ các thông tin bao gồm: Student, SPSO, Printer, Printing History, Printing Configuration, Printing Report ở dạng bảng có cấu trúc và có mối quan hệ với nhau.

4 Implementation – Sprint 1

Task 4.1

Setting up an online repository (github, bitbucket, etc) for version control.

4.1 Github for version control

Link Github: https://github.com/LekiizRaii/HCMUT-SSPS_L01_Group8

Task 4.2

Adding documents, materials and folders for Requirement, System modelling and Architectural design. Use the selected version control system to report the changes to these files.

4.2 Chi tiết commit

Chạy lệnh **git log > log.txt**

Ta được thông tin chi tiết git log như sau:

```
Delete docs/System Modelling/MVP 1 UI Figma/~$sk 2.4 MVP 1 User Interface Figma.docx

commit 69c2a3cb8aaef01e07e4a689996c12e12707287f
Author: huyhoang379 <hohuyhoang01@gmail.com>
Date:   Wed Nov 22 19:39:42 2023 +0700

    Update docs System Modelling

commit 7f626cf0f4b6d04092a472c5e5b8d1a112c63d23
Author: huyhoang379 <hohuyhoang01@gmail.com>
Date:   Wed Nov 22 19:37:47 2023 +0700

    Update docs System Modelling

commit df9f43dec9df3a3bc12cae5bd9e4553639b114e5
Author: tndgoat <tungnd.goat@gmail.com>
Date:   Wed Nov 22 19:33:48 2023 +0700

    update task3: architecture design

commit 20f566aa5ed09f82a76899d07fa21d10f94a8c70
Author: binhnguyen3816 <binhnguyen3816@gmail.com>
Date:   Wed Nov 22 19:26:48 2023 +0700

    update task1: requirement

commit 876f3612724b9a244fa8b4b43f1103205a5dde94
Author: binhnguyen3816 <binhnguyen3816@gmail.com>
Date:   Wed Nov 22 19:22:10 2023 +0700

    update task 1 requirement

commit ba678213fa9849b3690582ce85514fbe23c35872
Author: vuthn <vu.truongcompsci@hcmut.edu.vn>
Date:   Wed Nov 22 18:24:15 2023 +0700

    feat: add folders models and views

commit 1c335b5f0b5225fd04db8129e789948eb01f9649
Author: vuthn <vu.truongcompsci@hcmut.edu.vn>
Date:   Wed Nov 22 18:23:08 2023 +0700

    feat: add folders models and views

commit 1c335b5f0b5225fd04db8129e789948eb01f9649
Author: vuthn <vu.truongcompsci@hcmut.edu.vn>
Date:   Wed Nov 22 18:23:08 2023 +0700

    feat: add spso and student to folder controllers

commit 33e338ecc08af2eb5fa1090f98d598b5455b7bbc
Author: vuthn <vu.truongcompsci@hcmut.edu.vn>
Date:   Wed Nov 22 18:20:41 2023 +0700

    feat: move .sql files to folder db

commit 30dcaff55fd80b6914839c43f6d3279c36a7be8e
Author: vuthn <vu.truongcompsci@hcmut.edu.vn>
Date:   Wed Nov 22 18:14:12 2023 +0700

    feat: add docs folder

commit 6961416c7c0331b41e4986bf1f45c6922e22e213
Author: tndgoat <tungnd.goat@gmail.com>
Date:   Mon Nov 20 17:34:56 2023 +0700

    init the app structure

commit 18dbeb616bb162ef31a90822b56237260d6893f2
Author: Trương Hoàng Nguyên Vũ <vu.truongcompsci@hcmut.edu.vn>
Date:   Tue Nov 14 21:48:59 2023 +0700

    Initial commit
```

Figure 34: *Tạo đường link đến từng nhiệm vụ trong Google Docs*

Task 4.3

Conducted a usability test with the user interface you developed in MVP 1.

4.3 Kiểm tra khả năng sử dụng

Kiểm tra khả năng sử dụng (Usability Testing) là một phương pháp để kiểm tra mức độ dễ sử dụng của một thiết kế sản phẩm với nhóm người dùng mà sản phẩm hướng tới. Những người thực hiện kiểm tra nghiên cứu, gọi là researcher (hay "facilitator" hoặc "moderator") yêu cầu người tham gia thực hiện các nhiệm vụ nào đó được đưa ra.

Thông qua việc kiểm tra, các nhà phát triển phần mềm có thể tìm thấy các lỗi mà phần mềm đang mắc phải cũng như các cơ hội phát triển thêm cho phần mềm.

4.3.1 Tuyển dụng người tham gia kiểm thử

Ở bước này, nhóm sẽ chọn ra 5 người bao gồm 3 sinh viên và 2 người ở phòng đào tạo trong trường thử kiểm tra trải nghiệm của họ đối với interface front-end trong MVP1 mà nhóm đã xây dựng. Sinh viên trong trường là đối tượng sử dụng chính của sản phẩm này. Lý do ta chỉ chọn 5 người vì nếu ta chọn quá ít người dùng thử sẽ dẫn đến tình trạng các vấn đề khi sử dụng phần mềm sẽ chưa phát hiện hết, hoặc sẽ có những task không gây khó khăn cho sinh viên này nhưng lại gây khó khăn cho sinh viên khác. Ngược lại, nhóm không thể triển khai cho quá nhiều người dùng vì ràng buộc về thời gian và chi phí thực hiện

Nhóm sẽ tiến hành cho 5 người này sử dụng hoàn thành các tasks với phần mềm trong 2 lần. Sau khi nhóm người dùng thử và đưa ra phản hồi sau lần kiểm tra đầu tiên, nhóm sẽ chỉnh sửa lại MVP1 để thực hiện kiểm tra lần 2, sau đó ghi nhận và tiến hành hoàn chỉnh phần mềm web từ MVP1 với các chỉnh sửa.

4.3.2 Xác định công việc

Các công việc (tasks) cần thể hiện được chính xác và đầy đủ mục tiêu của người kiểm tra hoặc thỏa mãn các use-cases của phần mềm. Vì các task sẽ ảnh hưởng trực tiếp đến chất lượng sản phẩm, nhóm xác định một số quy tắc để xác định tasks và mô tả tốt hơn giúp người thực hiện kiểm tra có thể hiểu rõ các việc cần làm:

- Tránh để những từ có trong giao diện vào mô tả công việc. Ví dụ, khi người dùng đang đăng nhập là SPSO, ta không nên mô tả công việc là: "Nhập vào ô tìm kiếm lịch sử mã số sinh viên của sinh viên X để xem lịch sử của họ" mà thay vào đó là "Hãy tìm lịch sử in của sinh viên X thực hiện trong khoảng thời gian..."
- Đừng mô tả công việc như một hướng dẫn sử dụng, tức là mô tả công việc không chứa các bước để thực hiện công việc đó.
- Xây dựng công việc có tính thực tế. Ví dụ không yêu cầu SPSO xem lịch sử in ngày 5/12/2023 nếu hôm nay chỉ mới là ngày 4/12/2023.
- Đừng tạo các công việc quá đơn giản. Chẳng hạn thay vì chỉ yêu cầu sinh viên đăng nhập vào hệ thống thì có thể yêu cầu một công việc như: "Thực hiện việc in hoàn chỉnh một file với các yêu cầu

nhập vào như mong muốn". Ngược lại, ta không cần tạo các công việc quá phức tạp mà chủ yếu cần người dùng phải có khả năng xử lý thông tin thay vì chỉ tìm kiếm nó.

- Xây dựng tình hướng giả đinh: Một mô tả ngắn về tình huống giả định sẽ giúp người dùng thử hiểu rõ hơn về công việc cần làm.

Từ các quy tắc trên, nhóm đã tiến hành mô tả một số công việc tasks dựa vào các use-case quan trọng của hệ thống như sau:

Tên các use-case	**Mô tả công việc (task)**
Cung cấp thông tin in, Xác nhận in	Hãy thực hiện in một file bất kỳ. Trong quá trình thực hiện, có thể thao tác sai để xem các thông báo lỗi.
Xem lịch sử in	Trong 1 tuần qua, bản thân sinh viên đã in những file nào
Quản lý máy in	Máy in ở tòa X được phải gỡ đi vì hỏng
Quản lý máy in	Ngày mai cần ngưng hoạt động máy in X để thực hiện bảo trì
Quản lý máy in	Trường vừa mua thêm máy in và cần thêm vào hệ thống để phục vụ học sinh
Quản lý cài đặt in	Sinh viên muốn trường giảm khoảng cách giữa các lần cung cấp giấy định kỳ
Quản lý cài đặt in	Sinh viên muốn được cho phép in các file dạng ảnh
Xem báo cáo định kỳ	Trong tháng 8 vừa qua đã có bao nhiêu lượt in

Table 8: *Mô tả công việc ứng với các use-case*

4.3.3 Xác định phương pháp kiểm tra

Dựa vào tính chất của dự án, cùng với giới hạn về kinh phí và thời gian nhóm quyết định chọn kiểm tra theo phương hướng sau:

- **Phương pháp kiểm tra từ xa có giám sát (remote unmoderated)**: Người giám sát tương tác và người dùng thử và yêu cầu thực hiện các công việc thông qua các phần mềm có tính năng chia sẻ màn hình mà nhóm sử dụng là Google Meet.

Quyết định này được đưa ra dựa trên các lý do sau:

- Nhanh và ít tốn chi phí hơn: Phương pháp này không đòi hỏi người thực hiện kiểm tra phải chi trả chi phí di chuyển đến nơi gặp trực tiếp cho bản thân cũng như người tham gia. Bên cạnh đó, việc kiểm tra từ xa cũng giúp cả hai bên có thể linh hoạt thời gian hơn.
- Dễ dàng lựa chọn người tham gia: Nếu sử dụng phương pháp trực tiếp thì ta phải tiến hành tất cả các kiểm tra tại một nơi nào đó. Điều này đòi hỏi những người tham gia phải sinh sống ở gần nơi tập trung đó để tiện di chuyển. Ngược lại, phương pháp kiểm tra từ xa cho phép nhóm có thể mở rộng phạm vi tìm kiếm người tham gia trên toàn thành phố.

Bên cạnh đó nhóm cũng sẽ sử dụng **phương pháp định tính**. Ở phương pháp này, ta tập trung vào việc thu thập những thông tin chi tiết, những phát hiện khi thực hiện quá trình kiểm tra cũng như cách mà từng người dùng sử dụng sản phẩm hoặc dịch vụ của mình. Lý do nhóm chọn phương pháp này vì nhóm chỉ cần quan tâm đến hành vi và trải nghiệm của người dùng hơn nên phương pháp này sẽ phù hợp với sản phẩm của nhóm.

4.3.4 Tiến hành kiểm tra

Để tiến hành kiểm tra từ xa có giám sát nhóm đã tiến hành thực hiện các bước sau:

Bước 1: Lựa chọn công cụ để giao tiếp với người tham gia

- Nhóm lựa chọn công cụ Google Meet vì đơn giản và dễ sử dụng, đồng thời công cụ này được hỗ trợ trên nhiều loại thiết bị khác nhau (laptop, điện thoại. . .)
- Khuyến khích người tham gia mở webcam để tăng tính tương tác trong quá trình kiểm tra.
- Công cụ chat của Google Meet có thể gây xao nhãng người tham gia khi có đông người. Để giải quyết vấn đề này mỗi lần kiểm tra sẽ chỉ có 2 người là người tham gia và người giám sát.

Bước 2: Giao nhiệm vụ cho người tham gia

- Nhóm sẽ sử dụng công cụ chat trong Google Meet để gửi file Google Docs chứa các nhiệm vụ của người tham gia. File Google Doc này sẽ chỉ chứa đường link đến từng nhiệm vụ mà không hiển thị nội dung các nhiệm vụ trực tiếp để đảm bảo người dùng không vô tình xem trước nội dung của các nhiệm vụ khác.

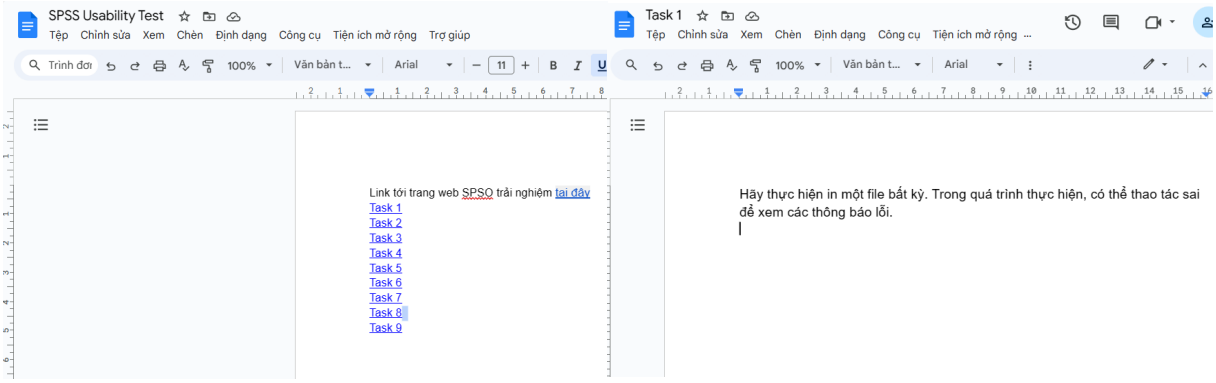


Figure 35: *Tạo đường link đến từng nhiệm vụ trong Google Docs*

- Yêu cầu người tham gia đọc to mô tả của từng nhiệm vụ. Điều này giúp đảm bảo người tham gia đã đọc toàn bộ nội dung của nhiệm vụ và không xem trước các nhiệm vụ khác.

Ngoài ra trước khi thực hiện việc kiểm tra thì nhóm cũng có gửi tin nhắn đến những người tham gia để nhắc nhở họ về thời gian thực hiện cuộc kiểm tra, đồng thời yêu cầu người tham gia phải có laptop được kết nối với nguồn điện, có tai nghe để cải thiện chất lượng âm thanh, có kết nối WiFi mạnh và ở nơi yên tĩnh để họ không bị xao nhãng.

Bước 3: Thực hiện việc kiểm tra: Trong bước này, người tham gia sẽ bắt đầu thực hiện các công việc được đề ra, người kiểm tra sẽ chú ý quan sát hành vi và biểu hiện của họ trong quá trình này. Sau khi người tham gia hoàn thành mỗi công việc, người kiểm tra có thể hỏi 1-2 câu về công việc đó để hiểu hơn về trải nghiệm của người dùng trong quá trình thực hiện tác vụ đó.

Bước 4: Kết thúc bài kiểm tra

Người hướng dẫn dừng là lưu video record, sau đó gặp các thành viên trong nhóm dự án để thảo luận về các vấn đề phát sinh (nếu có) trong bài kiểm tra đó. Từ đó đưa ra những thay đổi cần thiết cho phần mềm của nhóm

Bước 5: Ghi nhận phản hồi của người tham gia.

Sau khi kết thúc kiểm tra khả năng sử dụng của phần mềm, nhóm đã ghi nhận lại ý kiến và thực hiện các sửa đổi như sau:

- Sinh viên thứ nhất cho rằng lúc upload tập tin lên hệ thống và thực hiện nhập các thông số in, để có thêm một vùng tạo xem trước file được in để người dùng có thể kiểm tra các thông số chọn đã phù hợp hay chưa (ví dụ: các trang chọn để in đúng hay không). Để khắc phục nhược điểm này, nhóm đã chỉnh sửa lại giao diện thêm phần để xem file trước khi in.

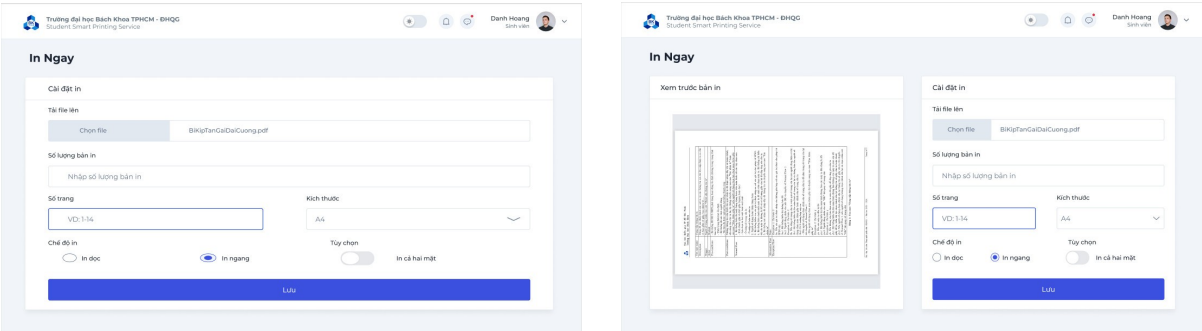


Figure 36: *Vùng upload tập tin trước và sau khi sửa đổi*

- Sinh viên thứ hai cho rằng ở phần xem lịch sử in, khung nhập thời gian nên có thêm biểu tượng lịch mà khi click vào người dùng có thể chọn ngày từ lịch để có thể thuận tiện hơn.

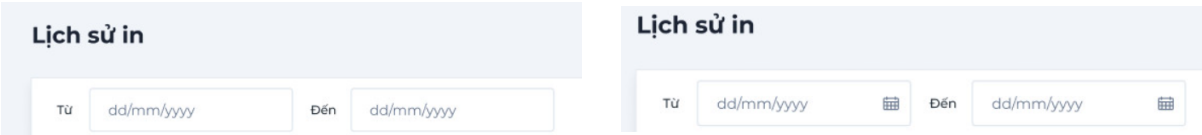


Figure 37: *Chọn khung thời gian xem lịch sử trước và sau*

- Sinh viên thứ ba có thể thực hiện các task dành cho sinh viên ổn thỏa. Tuy nhiên, sinh viên nhận ra người dùng có thể có nhu cầu muốn quay về trang chủ nhanh chóng bằng cách click vào logo của trang web thì họ không thể làm được vì logo chưa được liên kết để quay trở về trang chủ, điều này làm họ cảm thấy khó chịu vì thường người dùng sẽ mặc định rằng logo của phần mềm là công cụ để chuyển đến trang chủ của hệ thống.

- Đối với giảng viên ở phòng Đào tạo thứ nhất, cho rằng ở phần quản lý máy in, có trường hợp sẽ chưa có máy in nào được thêm vào trong hàng chờ dẫn tới trống trang, nhưng người dùng có thể hiểu lầm là trang vẫn đang load danh sách máy in đang kết nối trong hệ thống. Vì vậy nên có thêm thông báo khi không có máy in đang kết nối với hệ thống. Nhóm đã ghi nhận và thực hiện sửa đổi:

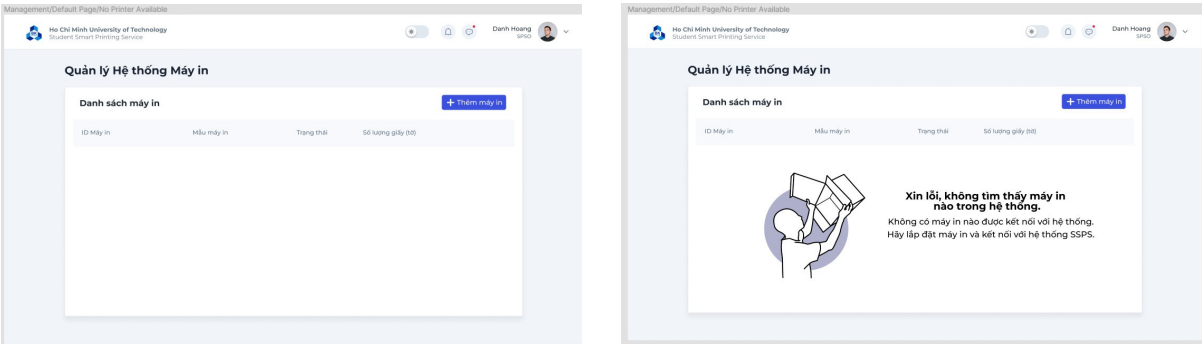


Figure 38: *Thêm thông báo khi không có máy in kết nối với hệ thống*

- Giảng viên ở phòng Đào tạo thứ hai cho rằng ở quản lý máy in, khi click vào chi tiết một máy in, nếu làm mờ đi (hover) những nút cài đặt nào không thể được chọn (ví dụ: máy in đã khởi động (enable) thì làm mờ đi nút enable). Hơn nữa, việc khởi động (enable) hay tắt (disable) chỉ có thể làm được khi máy in đã kết nối vào hệ thống, vì vậy nên tạo thêm một lưu ý trong bảng chi tiết để người dùng lưu ý. Nhóm đã ghi nhận và chỉnh sửa:

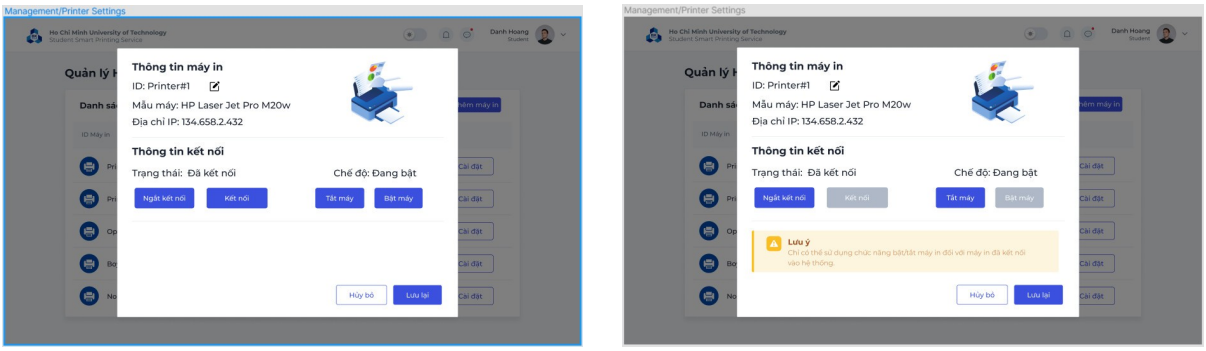
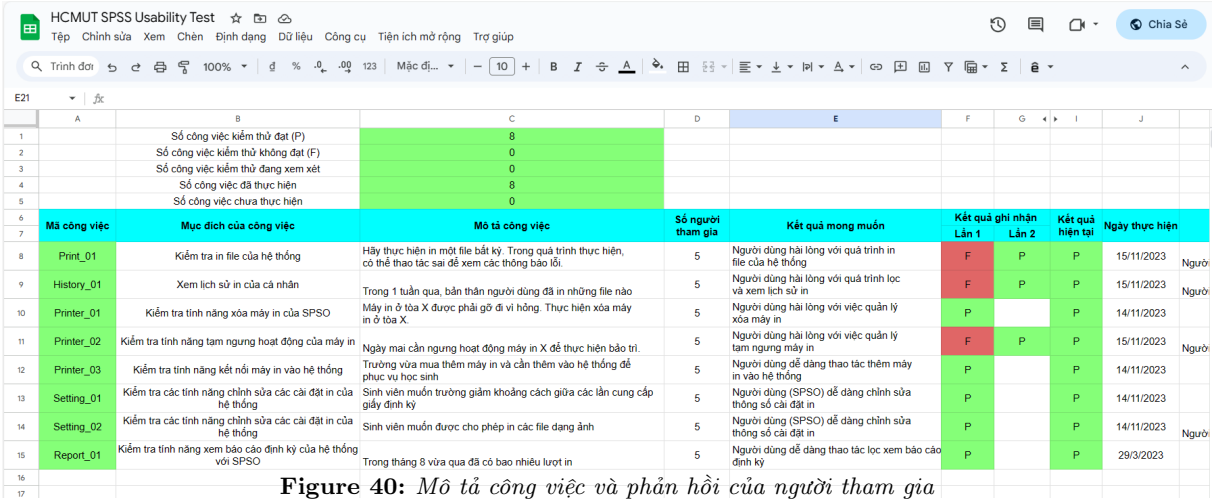


Figure 39: *Chỉnh sửa bảng chi tiết của quản lý từng máy in*

Demo về trang interface của nhóm có thể được tham khảo tại đây: https://lekiizraii.github.io/HCMUT-SSPS_L01_Group8.

Các thông tin về mô tả công việc, số lượng người tham gia mỗi công việc, và phản hồi không hài lòng của họ (nếu có) đều được nhóm ghi nhận lại trong file Google Sheets (xem tại đây) để tiện việc theo dõi và nắm bắt được các chức năng đã đạt và chưa đạt yêu cầu của trang web.



HCMUT SPSS Usability Test

1	Số công việc kiểm thử đạt (P)	8
2	Số công việc kiểm thử không đạt (F)	0
3	Số công việc kiểm thử đang xem xét	0
4	Số công việc đã thực hiện	8
5	Số công việc chưa thực hiện	0

Mã công việc	Mục đích của công việc	Mô tả công việc	Số người tham gia	Kết quả mong muốn	Kết quả ghi nhận Lần 1	Kết quả ghi nhận Lần 2	Kết quả hiện tại	Ngày thực hiện	
Print_01	Kiểm tra in file của hệ thống	Hãy thực hiện in một file bất kỳ. Trong quá trình thực hiện, có thể thao tác sai để xem các thông báo lỗi.	5	Người dùng hài lòng với quá trình in file của hệ thống	F	P	P	15/11/2023	Người
History_01	Xem lịch sử in của cá nhân	Trong 1 tuần qua, bản thân người dùng đã in những file nào	5	Người dùng hài lòng với quá trình lọc và xem lịch sử in	F	P	P	15/11/2023	Người
Printer_01	Kiểm tra tính năng xóa máy in của SPSO	Máy in ở tòa X được phải gỡ đi vì hỏng. Thực hiện xóa máy in ở tòa X.	5	Người dùng hài lòng với việc quản lý xóa máy in	P		P	14/11/2023	
Printer_02	Kiểm tra tính năng tạm ngưng hoạt động của máy in	Ngày mai cần ngưng hoạt động máy in X để thực hiện bảo trì.	5	Người dùng hài lòng với việc quản lý tạm ngưng máy in	F	P	P	15/11/2023	Người
Printer_03	Kiểm tra tính năng kết nối máy in vào hệ thống	Trường vừa mua thêm máy in và cần thêm vào hệ thống để phục vụ học sinh	5	Người dùng dễ dàng thao tác thêm máy in vào hệ thống	P		P	14/11/2023	
Setting_01	Kiểm tra các tính năng chỉnh sửa các cài đặt in của hệ thống	Sinh viên muốn trường giảm khoảng cách giữa các lần cung cấp giấy định kỳ	5	Người dùng (SPSO) dễ dàng chỉnh sửa thông số cài đặt in	P		P	14/11/2023	
Setting_02	Kiểm tra các tính năng chỉnh sửa các cài đặt in của hệ thống	Sinh viên muốn được cho phép in các file dạng ảnh	5	Người dùng (SPSO) dễ dàng chỉnh sửa thông số cài đặt in	P		P	14/11/2023	Người
Report_01	Kiểm tra tính năng xem báo cáo định kỳ của hệ thống với SPSO	Trong tháng 8 vừa qua đã có bao nhiêu lượt in	5	Người dùng dễ dàng thao tác lọc xem báo cáo định kỳ	P		P	29/3/2023	

Figure 40: *Mô tả công việc và phản hồi của người tham gia*