

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)

---

BÀI TẬP LỚN – LỚP L01 – NHÓM Localhost – HK241

# A SMART PRINTING SERVICE FOR STUDENTS AT HCMUT

---

GVHD:	ThS. Trần Trương Tuấn Phát	
SV thực hiện:	Trần Thế Nhân	2212383
	Trần Ngọc Lâm Vỹ	2214061
	Lê Võ Đăng Khoa	2211606
	Nguyễn Duy Bách	2210185
	Trần Phước Nhật	2212412
	Nguyễn Thành Nhân	2212366
	Hồ Tấn Phát	2212507
	Nguyễn Khánh Bình	2210318
	Nguyễn Thanh Nhân	2212364
	Đặng Trần Minh Nhật	2212388

TP Hồ Chí Minh, Tháng 10 Năm 2024

## Mục lục

<b>1</b>	<b>Giới thiệu (Task 1.1)</b>	<b>3</b>
1.1	Đề tài . . . . .	3
1.2	Bối cảnh . . . . .	4
1.3	Các bên liên quan và nhu cầu . . . . .	4
1.4	Lợi ích . . . . .	5
<b>2</b>	<b>Thu thập yêu cầu (Task 1.2)</b>	<b>5</b>
2.1	Yêu cầu chức năng . . . . .	5
2.2	Yêu cầu phi chức năng . . . . .	6
<b>3</b>	<b>Use-case (Task 1.3)</b>	<b>8</b>
3.1	Sơ đồ use-case cho hệ thống . . . . .	8
3.2	Sơ đồ use-case cho việc in tài liệu . . . . .	10
3.3	Bảng mô tả use-case cho việc in tài liệu . . . . .	11
<b>4</b>	<b>Mô hình hóa hệ thống (Task 2.1, 2.2, 2.3)</b>	<b>15</b>
4.1	Activity diagram . . . . .	15
4.2	Sequence diagram . . . . .	17
4.3	Class diagram . . . . .	20
<b>5</b>	<b>Thiết kế giao diện (Task 2.4)</b>	<b>20</b>
5.1	Giao diện của sinh viên . . . . .	20
5.1.1	Giao diện trang chủ - trước Đăng nhập . . . . .	20
5.1.2	Giao diện đăng nhập . . . . .	21
5.1.3	Giao diện trang chủ - sau Đăng nhập . . . . .	21
5.1.4	Giao diện dịch vụ in tài liệu . . . . .	22
5.1.5	Dịch vụ in tài liệu - Tải file . . . . .	22
5.1.6	Dịch vụ in tài liệu - Chọn thuộc tính và máy in . . . . .	23
5.1.7	Dịch vụ in tài liệu - In thành công . . . . .	23
5.1.8	Giao diện dịch vụ mua trang in . . . . .	24
5.1.9	Giao diện lịch sử in . . . . .	24
5.1.10	Giao diện lịch sử thanh toán . . . . .	25
5.2	Giao diện của SPSO . . . . .	25
5.2.1	Giao diện đăng nhập . . . . .	25
5.2.2	Giao diện quản lý máy in . . . . .	26
5.2.3	Giao diện quản lý chi tiết máy in . . . . .	26
5.2.4	Giao diện quản lý người dùng . . . . .	27
5.2.5	Giao diện quản lý chi tiết người dùng . . . . .	27
5.2.6	Giao diện cấu hình hệ thống . . . . .	28



<b>6</b>	<b>Thiết kế kiến trúc (Task 3)</b>	<b>29</b>
6.1	Layered architecture . . . . .	29
6.2	Chiến lược . . . . .	29
6.2.1	Chiến lược hiển thị giao diện người dùng . . . . .	29
6.2.2	Chiến lược quản lý API và lưu trữ dữ liệu . . . . .	32
6.3	Component diagram . . . . .	35

## 1 Giới thiệu (Task 1.1)

### 1.1 Đề tài

Trường đại học có ý định xây dựng Dịch vụ in thông minh cho sinh viên (HCMUT\_SSPTS) để phục vụ sinh viên trong khuôn viên trường in tài liệu của họ.

Hệ thống bao gồm một số máy in xung quanh khuôn viên trường. Mỗi máy in có ID, tên thương hiệu/nhà sản xuất, kiểu máy in, mô tả ngắn và vị trí (tên khuôn viên trường, tên tòa nhà và số phòng).

Hệ thống cho phép sinh viên in tài liệu bằng cách tải tệp tài liệu lên hệ thống, chọn máy in và chỉ định các thuộc tính in như kích thước giấy, số trang (của tệp) cần in, một mặt/hai mặt, số lượng bản sao, v.v. Các loại tệp được phép bị giới hạn và được cấu hình bởi Cán bộ dịch vụ in ẩn cho sinh viên (SPSO).

Hệ thống phải ghi lại các hành động in cho tất cả sinh viên, bao gồm ID sinh viên, ID máy in, tên tệp, thời gian bắt đầu và kết thúc in, số trang cho mỗi kích thước trang.

Hệ thống cho phép SPSO xem lịch sử in (nhật ký) của tất cả sinh viên hoặc một sinh viên trong một khoảng thời gian (ngày đến ngày) và cho tất cả hoặc một số máy in. Tất nhiên, sinh viên cũng có thể xem nhật ký in ẩn của mình trong một khoảng thời gian cùng với bản tóm tắt số trang đã in cho từng kích thước trang.

Đối với mỗi học kỳ, trường đại học cung cấp cho mỗi sinh viên một số trang khổ A4 mặc định để in. Sinh viên được phép mua thêm một số trang bằng tính năng Mua trang in của hệ thống và thanh toán số tiền thông qua một số hệ thống thanh toán trực tuyến như hệ thống BKPay của trường đại học. Hệ thống chỉ cho phép sinh viên in một số lượng trang nhất định khi số trang đó không vượt quá số dư tài khoản (trang) của sinh viên. Lưu ý rằng, một trang A3 tương đương với hai trang A4.

SPSO có một tính năng để quản lý máy in như thêm/bật/tắt máy in.

SPSO cũng có một tính năng để quản lý các cấu hình khác của hệ thống như thay đổi số trang mặc định, ngày mà hệ thống sẽ cung cấp số trang mặc định cho tất cả sinh viên, các loại tệp được phép mà hệ thống chấp nhận.

Các báo cáo về việc sử dụng hệ thống in ấn được tạo tự động vào cuối mỗi tháng và mỗi năm và được lưu trữ trong hệ thống và SPSO có thể xem bất cứ lúc nào.

Tất cả người dùng phải được xác thực bởi dịch vụ xác thực HCMUT\_SSO trước khi sử dụng hệ thống.

## 1.2 Bối cảnh

Xuất phát từ nhu cầu thực tế về việc in ấn tài liệu nhằm nâng cao chất lượng dịch vụ tại trường Đại học Bách Khoa TP.HCM, nhà trường dự định làm một hệ thống dịch vụ in ấn thông minh (HCMUT\_SPSS). Sinh viên, học viên thường xuyên cần in tài liệu học tập, báo cáo và các tài liệu khác, do đó, một hệ thống in thông minh rất cần thiết để đáp ứng nhu cầu ấy. Hệ thống này mang lại tính tiện lợi khi cho phép sinh viên, học viên in tài liệu từ bất kỳ đâu trong khuôn viên trường, chỉ cần truy cập vào hệ thống và chọn máy in gần nhất. Ngoài ra, hệ thống còn ghi lại lịch sử in của sinh viên, sinh viên có thể xem lịch sử và số trang còn lại của tài khoản mình. Thông qua đó, cán bộ quản trị có thể theo dõi được lịch sử in của sinh viên toàn trường bằng các báo cáo định kỳ. Hơn nữa, hệ thống có khả năng mở rộng và tích hợp với các dịch vụ khác của trường, như hệ thống thanh toán trực tuyến BKPay.

## 1.3 Các bên liên quan và nhu cầu

Về phía sinh viên, học viên luôn có nhu cầu in ấn những tài liệu cần thiết trong thời gian đợi ngắn nhất và quãng đường di chuyển ít nhất để dành thời gian cho việc học tại trường và những hoạt động cá nhân khác. Về cán bộ quản lý có nhu cầu giám sát, quản lý việc sử dụng hệ thống máy in.

## 1.4 Lợi ích

Hệ thống này không chỉ giúp tiết kiệm thời gian và công sức cho sinh viên mà còn tăng cường hiệu quả học tập và nghiên cứu. Việc quản lý và theo dõi việc sử dụng máy in giúp tối ưu hóa việc bảo trì và sử dụng tài nguyên in ấn, giảm thiểu lãng phí giấy và mực in. Hơn nữa, khả năng mở rộng và tích hợp với các dịch vụ khác của hệ thống tạo ra một hệ dịch vụ tiện ích cho sinh viên. Việc xây dựng và triển khai hệ thống này không chỉ đáp ứng nhu cầu thực tế của sinh viên mà còn góp phần phát triển kỹ năng và kiến thức về công nghệ thông tin, quản lý hệ thống và dịch vụ khách hàng cho sinh viên và cán bộ kỹ thuật.

## 2 Thu thập yêu cầu (Task 1.2)

### 2.1 Yêu cầu chức năng

#### Yêu cầu của người dùng

- Tải file văn bản lên.
- Chọn máy in.
- Chọn thuộc tính in: Kích thước giấy; Số trang in; Chế độ in (một/hai mặt); Số bản in.
- Xem lịch sử in, bao gồm số trang và kích thước trang giấy đã in. Lịch sử in sẽ được lưu trữ trong một khoảng thời gian nhất định.
- Mua thêm trang in từ hệ thống.
- Thanh toán bằng dịch vụ thanh toán trực tuyến.
- Sử dụng dịch vụ xác thực HCMUT\_SSO để đăng ký, đăng nhập.

#### Yêu cầu của cán bộ quản lý (SPSO)

- Sử dụng dịch vụ xác thực HCMUT\_SSO để đăng ký, đăng nhập.

- Xem lịch sử in của một hoặc toàn bộ học sinh; một hoặc toàn bộ máy in.  
Lịch sử được lưu trữ trong một khoảng thời gian nhất định.
- Thêm, kích hoạt, vô hiệu hóa, loại bỏ một máy in.
- Quản lý các điều chỉnh khác của hệ thống:
  - Thay đổi số trang mặc định.
  - Thiết lập ngày cung cấp số trang mặc định miễn phí cho toàn bộ người dùng.
  - Thiết lập loại file cho phép để đăng tải lên hệ thống.
- Xem báo cáo.
- Ghi lại lịch sử in của toàn bộ người dùng.
- Tự động tạo báo cáo sử dụng hệ thống in vào cuối mỗi tháng, năm và lưu trên hệ thống.

## 2.2 Yêu cầu phi chức năng

- Hiệu suất:
  - Thời gian phản hồi: Hệ thống cần phản hồi các yêu cầu in, thay đổi cấu hình và xem lịch sử với thời gian dưới 3 giây trong đa số tình huống.
  - Khả năng mở rộng: Hệ thống cần có khả năng xử lý số lượng lớn tương tác đồng thời lên đến hàng trăm người dùng. Đặc biệt tại các cao điểm như trong mùa thi hoặc giai đoạn nộp báo cáo, luận án.
  - Thông lượng: Hệ thống cần có khả năng xử lý hàng trăm lượt tải lên tài liệu và hoạt động in ấn mỗi phút.
- Độ tin cậy và tính khả dụng:
  - Uptime: Hệ thống cần có uptime đạt 99.9%, đảm bảo luôn có máy in có thể được dùng trong khuôn viên với downtime ở mức tối thiểu.

- Xử lý lỗi: Hệ thống cần xử lý linh hoạt các tình huống hỏng máy in hoặc lỗi mạng bằng cách như giao lại việc cho các máy in còn hoạt động hoặc báo lỗi cho người dùng.
- Khả năng khôi phục: Thường xuyên backup lịch sử in và cấu hình hệ thống, đảm bảo hệ thống có thể được khôi phục trong thời gian ngắn khi gặp sự cố.
- Bảo mật:
  - Mã hóa dữ liệu: Dữ liệu nhạy cảm như ID sinh viên, tên file và thông tin thanh toán cần được mã hóa trong quá trình truyền và lưu trữ.
  - Kiểm soát truy cập dựa trên vai trò: Các vai trò khác nhau (sinh viên, SPSO) phải có quyền truy cập vào các tính năng khác nhau của hệ thống. SPSO có quyền truy cập quản trị, trong khi sinh viên có quyền truy cập hạn chế vào nhật ký và số dư của riêng họ.
  - Có khả năng ngăn chặn tấn công DOS.
- Khả năng bảo trì:
  - Tài liệu hệ thống: Cần cung cấp tài liệu phù hợp cho nhà phát triển và người dùng để đảm bảo dễ hiểu và bảo trì hệ thống.
- Khả năng sử dụng:
  - Hỗ trợ nhiều ngôn ngữ: Hệ thống nên hỗ trợ nhiều ngôn ngữ, chủ yếu là tiếng Việt và tiếng Anh, để phù hợp với nhiều đối tượng người dùng khác nhau.
  - Giao diện thân thiện với người dùng: Ứng dụng trên web phải có giao diện trực quan, cho phép sinh viên dễ dàng tải lên tài liệu, chọn máy in và xem lịch sử.



### 3 Use-case (Task 1.3)

#### 3.1 Sơ đồ use-case cho hệ thống

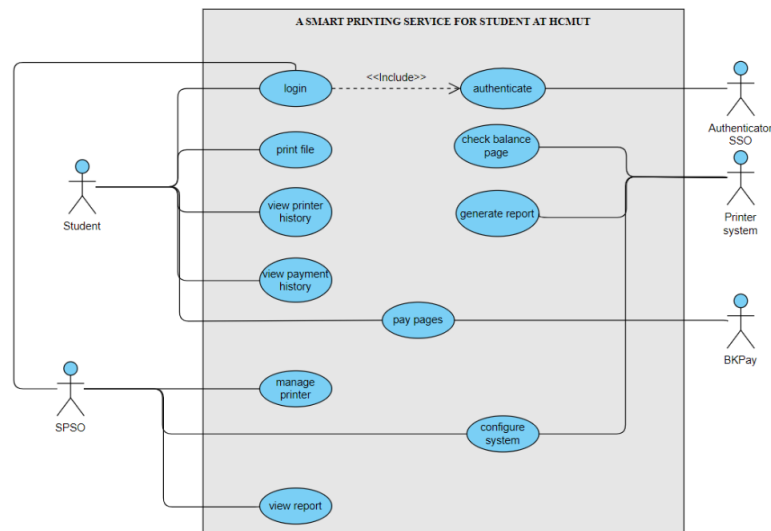


Figure 1: System Use-case diagram

#### Bảng mô tả các use-case

ID	Use-case	Mô tả	Actor
PF	Print file	Sinh viên tương tác với hệ thống in để tải file lên hệ thống, chọn cấu hình để in, tìm máy in. Hệ thống kiểm tra kiểu file cho phép và số trang có khả dụng với yêu cầu sinh viên.	Student, Printer system
VPR	View printer history	Sinh viên xem lại lịch sử in	Student
VPA	View payment history	Sinh viên xem lại lịch sử giao dịch	Student

PP	Pay pages	Sinh viên mua trang in thông qua hệ thống BKpay.	Student, BKpay
CS	Configure system	Cấu hình hệ thống in bao gồm loại file cho phép được upload cho hệ thống	
MP	Manage printer	Quản lý hệ thống máy in bao gồm thêm, xóa, kích hoạt, vô hiệu hóa máy in, kiểm tra lịch sử giao dịch, lịch sử in của từng máy in	Printer system
VR	View report	Xem báo cáo hệ thống in định kỳ theo tháng, năm	SPSO
GR	Generate report	Hệ thống in sinh báo cáo định kỳ	Print system
CB	Check available page	Kiểm tra số trang khả dụng của máy in	Printer system
LG	Login	Người dùng đăng nhập và được xác thực thông qua SSO Authenticator	User, SPSO
AU	Authenticate	Xác thực người dùng đăng nhập	SSO Authenticator

### 3.2 Sơ đồ use-case cho việc in tài liệu

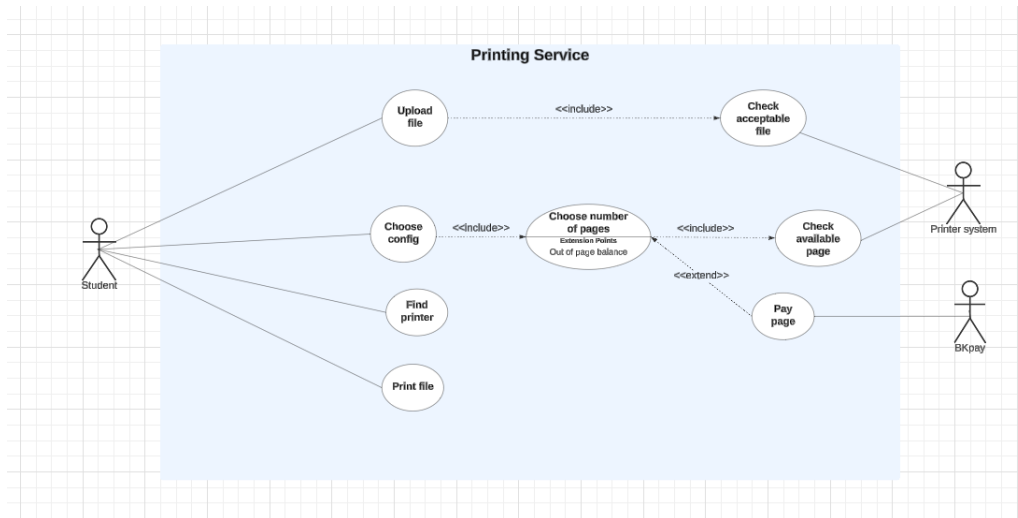


Figure 2: Printing file Use-case diagram

### 3.3 Bảng mô tả use-case cho việc in tài liệu

Use-case ID	PF1
Use-case Name	Upload file
Actors	Sinh viên, printer system
Description	Sinh viên đăng tải file cần in.
Trigger	Người dùng nhấn vào nút "Tải file" trên ứng dụng web.
Precondition	<ol style="list-style-type: none"><li>1. Hệ thống đang hoạt động.</li><li>2. Sinh viên đã đăng nhập.</li><li>3. Sinh viên đăng tải một file có định dạng hợp lệ và dung lượng không vượt mức cho phép.</li></ol>
Postcondition	<ol style="list-style-type: none"><li>1. Sinh viên chọn file vào dropbox và nhấn nút "Tải file".</li><li>2. Hệ thống hiện popup xác nhận lại.</li><li>3. Sau khi chọn "Đồng ý", hệ thống sẽ kiểm tra định dạng file.</li></ol>
Normal Flow	<ol style="list-style-type: none"><li>1. Người tìm việc nhấn nút "Tải file".</li><li>2. Hệ thống hiện popup xác nhận lại.</li><li>3. Sau khi chọn "Đồng ý", hệ thống sẽ kiểm tra định dạng file.</li><li>4. Hệ thống lưu thông tin file vào cơ sở dữ liệu.</li></ol>
Alternative Flows	Không có.
Exceptions	<p>Exception 1: tại bước 3</p> <ul style="list-style-type: none"><li>- Nếu người tìm việc chọn "Hủy", popup xác nhận sẽ tắt và người dùng sẽ trở lại giao diện trước popup.</li><li>- Nếu kiểm tra file có định dạng không phù hợp, popup hủy tải và yêu cầu tải lại.</li></ul>

Use-case ID	PF2
Use-case Name	Choose config
Actors	Sinh viên, printer system
Description	Người dùng cài đặt các thuộc tính in như loại giấy, số bản in, chế độ in 1 mặt, 2 mặt, v.v.
Trigger	Người dùng nhấn nút “Tiếp theo” sau khi hoàn thiện việc đăng tải file.
Precondition	<ol style="list-style-type: none"><li>1. Hệ thống đang hoạt động.</li><li>2. Người dùng đã đăng nhập.</li><li>3. Người dùng đã đăng tải file cần in.</li></ol>
Postcondition	<ol style="list-style-type: none"><li>1. Hệ thống ghi nhận cài đặt của người dùng để chuẩn bị cho việc in.</li><li>2. Người dùng có đủ số dư trang in để thực hiện yêu cầu in ấn.</li></ol>
Normal Flow	<ol style="list-style-type: none"><li>1. Người dùng nhấn nút “Tiếp theo” trên giao diện đăng tải file sau khi hoàn thành hành động này.</li><li>2. Người dùng cài đặt các thuộc tính theo nhu cầu như số bản in, loại giấy in, chế độ in.</li><li>3. Người dùng bấm “Xác nhận” để lưu thông tin cài đặt.</li></ol>
Alternative Flows	Không có
Exceptions	Exception 1: Tại bước 3 - Khi số dư trang in người dùng không đủ, hệ thống sẽ popup thông báo thất bại và yêu cầu nhập lại cài đặt hoặc mua thêm trang.

Use-case ID	PF3
Use-case Name	Find printer
Actors	Sinh viên (end-user)
Description	Người dùng tra cứu và chọn máy in mà mình muốn sử dụng.
Trigger	Người dùng nhấn nút “Tiếp theo” sau khi hoàn thiện việc điều chỉnh thuộc tính bản in.
Precondition	<ol style="list-style-type: none"><li>1. Hệ thống đang hoạt động.</li><li>2. Người dùng đã đăng nhập.</li><li>3. Người dùng đã đăng tải file cần in và hoàn thiện việc điều chỉnh thuộc tính bản in.</li></ol>
Postcondition	<ol style="list-style-type: none"><li>1. Hệ thống ghi nhận lựa chọn của người dùng và chuẩn bị máy in khả dụng.</li><li>2. Máy in có đủ giấy in vật lý cho nhu cầu sinh viên.</li></ol>
Normal Flow	<ol style="list-style-type: none"><li>1. Người dùng nhấn nút “Tiếp theo” trên giao diện điều chỉnh thuộc tính bản in.</li><li>2. Hệ thống căn cứ vào số trang cần in và hiển thị các máy in còn giấy cũng như vị trí của chúng.</li><li>3. Hệ thống ghi nhận lại thiết lập của người dùng.</li></ol>
Alternative Flows	Không có.
Exceptions	Exception 1: Tại bước 2 - Khi không còn máy in nào đủ giấy để đáp ứng nhu cầu in của sinh viên, hệ thống popup thông báo hết giấy và thoát ra.

Use-case ID	PF4
Use-case Name	Print file
Actors	Sinh viên và máy in
Description	Sinh viên in tài liệu đã tải lên.
Trigger	Sinh viên nhấn nút "Xác nhận in".
Precondition	<ol style="list-style-type: none"><li>1. Hệ thống đang hoạt động.</li><li>2. Sinh viên đã đăng nhập.</li><li>3. Sinh viên có số dư trang trong tài khoản lớn hơn hoặc bằng so với lượng giấy sử dụng để in.</li><li>4. Số lượng giấy trong máy in lớn hơn hoặc bằng nhu cầu in của sinh viên.</li></ol>
Postcondition	<ol style="list-style-type: none"><li>1. Hệ thống xử lý yêu cầu in ấn của sinh viên.</li><li>2. Lưu lại lịch sử in ấn của sinh viên.</li></ol>
Normal Flow	<ol style="list-style-type: none"><li>1. Sau khi đã chọn máy in khả dụng bấm “Xác nhận in”.</li><li>2. Hệ thống hiện popup xác nhận lại.</li><li>3. Sau khi chọn “Đồng ý”, hệ thống sẽ tiến hành in ấn.</li><li>4. Máy in gửi thông báo in thành công đến người dùng và tài liệu được in.</li><li>5. Hệ thống trừ số dư trang trong tài khoản sinh viên và trừ trong số dư giấy khả dụng tại máy in.</li><li>6. Hệ thống lưu lại lịch sử in.</li></ol>
Alternative Flows	Không có.
Exceptions	<p>Exception 1: tại bước 3</p> <ul style="list-style-type: none"><li>- Nếu người dùng chọn “Hủy”, popup xác nhận sẽ tắt và người dùng sẽ trở lại giao diện trước popup.</li></ul> <p>Exception 2: tại bước 3</p> <ul style="list-style-type: none"><li>- Nếu in không thành công do lỗi kỹ thuật, hệ thống hiện thông tin liên hệ của admin và report lỗi.</li></ul>

## 4 Mô hình hóa hệ thống (Task 2.1, 2.2, 2.3)

### 4.1 Activity diagram

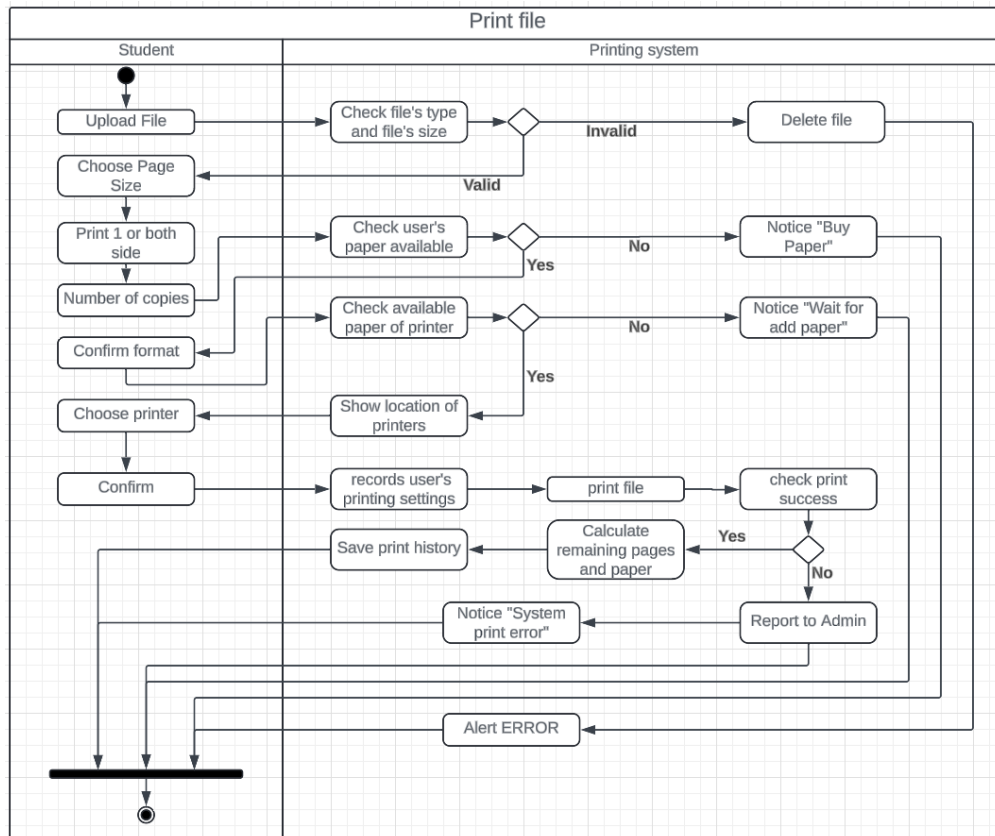


Figure 3: Activity diagram for print file module

#### Mô tả sơ đồ hoạt động của việc in file

1. Người dùng sau khi chọn và mục “In tài liệu” thì tiến hành upload file cần in.
2. Sau khi đã tải được file in lên, hệ thống tiến hành kiểm tra kiểu dữ liệu cũng như kích thước của file in được admin quy định:
  - Nếu sai quy định sẽ hiện cảnh báo định dạng/ kích thước file không hợp lệ.
  - Nếu đúng thì sẽ tiến hành bước tiếp theo.



3. Sau khi upload file thành công, người dùng thiết lập định dạng in và số lượng bản in cần thiết
4. Hệ thống sẽ tính toán số trang in và so sánh với số lượng trang in hiện có của người dùng:
  - Nếu số lượng trang in của người dùng nhỏ hơn số trang in cần thiết, hệ thống sẽ yêu cầu người dùng mua thêm giấy.
  - Nếu số lượng trang in hiện có hợp lệ thì tiếp tục.
5. Xác nhận format in
6. Sau khi xác nhận format in, hệ thống sẽ tiến hành tìm các máy in còn đủ số lượng giấy để in và sau đó hiện ra các máy in để người dùng lựa chọn phù hợp với vị trí của bản thân. Trường hợp máy in hết giấy thì hiện thông báo đợi bổ sung giấy.
7. Người dùng chọn máy in và xác nhận để tiến hành in.
8. Hệ thống sau đó sẽ lưu lại các cài đặt in của người dùng và tiến hành in.
9. Sau khi in hệ thống sẽ kiểm tra xem việc in có gặp lỗi không:
  - Nếu gặp lỗi thì sẽ báo cáo lại cho admin để admin xử lý các trường hợp gặp lỗi. Hệ thống sẽ thông báo “Lỗi khi in” đến người dùng.
  - Nếu không gặp lỗi thì coi như việc in thành công. Hệ thống tiến hành tính toán lại số trang in của người dùng cũng như số giấy của máy in. Sau đó lưu lại lịch sử in của người dùng.

## 4.2 Sequence diagram

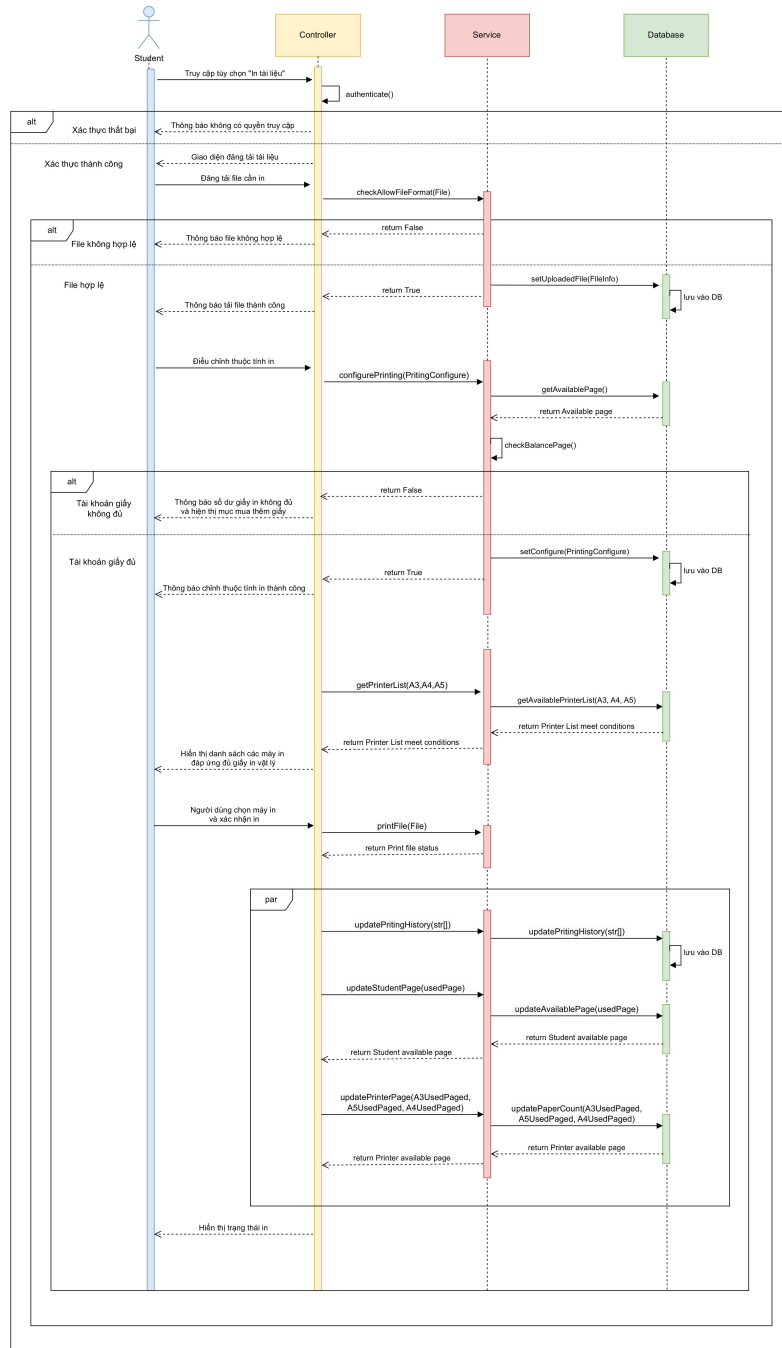


Figure 4: Sequence diagram for print file module

### *Mô tả sơ đồ tuần tự cho việc in file*

1. Khi sinh viên (Student) truy cập tùy chọn "In tài liệu" trên giao diện hệ thống, **Controller** gọi phương thức `authenticate()` để xác thực người dùng.
  - Nếu xác thực thất bại, hệ thống thông báo người dùng không có quyền truy cập dịch vụ này.
  - Nếu xác thực thành công, thực hiện tiếp Bước 2.
2. Hệ thống hiện giao diện đăng tải file cho sinh viên. Sau đó sinh viên gửi file cần in lên hệ thống. **Controller** gọi phương thức `checkAllowFileFormat(File)` đến **Service** để kiểm tra định dạng file có hợp lệ với cấu hình của hệ thống hay không.
  - Nếu phương thức trên trả về **False**, tức là file không hợp lệ. Hệ thống sẽ gửi thông báo file không hợp lệ cho sinh viên.
  - Nếu phương thức trên trả về **True**, tức là file hợp lệ. **Service** gọi phương thức `setUploadedFile(FileInfo)` để lưu thông tin file đã đăng tải vào **Database**. Đồng thời, **Controller** gửi thông báo tải file thành công đến sinh viên và thực hiện tiếp bước 3.
3. Sinh viên điều chỉnh thuộc tính in (số trang, loại giấy, số bản sao, ...). **Controller** gọi phương thức `configurePrinting(PrintingConfigure)` đến **Service** để cấu hình cho thuộc tính in. Sau đó, **Service** gọi phương thức `getAvailablePage()` đến **Database** để lấy thông tin số trang trong tài khoản sinh viên (là trang miễn phí và trang mua thêm). **Database** trả về dữ liệu là **Available page**. Sau đó **Service** kiểm tra số trang trong tài khoản sinh viên (**Available Page**) với số trang sinh viên cần in (đã nhập trong thuộc tính in) bằng phương thức `checkBalancePage()`.
  - Nếu **Available Page** của sinh viên nhỏ hơn số trang cần in, **Service** sẽ trả về **False** tức là số trang in không đủ. **Controller** sẽ gửi thông báo

số dư giấy in không đủ và hiện thị mục mua thêm giấy.

- Nếu **Available Page** của sinh viên không nhỏ hơn số trang cần in, **Service** sẽ trả về **True** tức là số trang in đủ. **Controller** sẽ gửi thông báo chỉnh thuộc tính in thành công. Thực hiện tiếp bước 4.

4. **Controller** gọi phương thức `getPrinterList(A3,A4,A5)` đến **Service** để tìm kiếm danh sách máy in đủ giấy cho nhu cầu in của sinh viên. **Service** gọi phương thức `getAvailablePrinterList(A3, A4, A5)` để truy vấn danh sách máy in thỏa điều kiện trong **Database**. Tiếp đó, **Database** trả về **Printer List meet conditions** cho **Service**, **Service** trả về **Printer List meet conditions** cho **Controller**. Sau đó giao diện hiển thị danh sách các máy in đáp ứng đủ giấy in vật lý cho Sinh viên chọn.

5. Sinh viên chọn máy in và xác nhận in. Lúc này, **Controller** gọi phương thức `printFile(File)` đến **Service** để in file. Sau khi máy in in xong, **Service** trả về trạng thái in file **Print file status**. Tiếp đó, **Controller** thực hiện gọi song song các phương thức sau đến **Service**:

- Phương thức `updatePrintingHistory(str[])` để lưu thông tin vừa in vào lịch sử in.
- Phương thức `updateStudentPage(usedPage)` để cập nhật số trang trong tài khoản của sinh viên (= *số trang ban đầu* – *số trang đã sử dụng*). **Service** trả về số trang còn lại trong tài khoản sinh viên **Student available page**.
- Phương thức `updatePrinterPage(A3UsedPaged, A5UsedPaged, A4UsedPaged)` để cập nhật số giấy còn lại trong máy in vừa sử dụng (= *số giấy ban đầu* – *số giấy đã sử dụng*). **Service** trả về số trang còn lại trong máy in **Printer available page**.

6. Cuối cùng, **Controller** hiển thị trạng thái in cho Sinh viên.

### 4.3 Class diagram

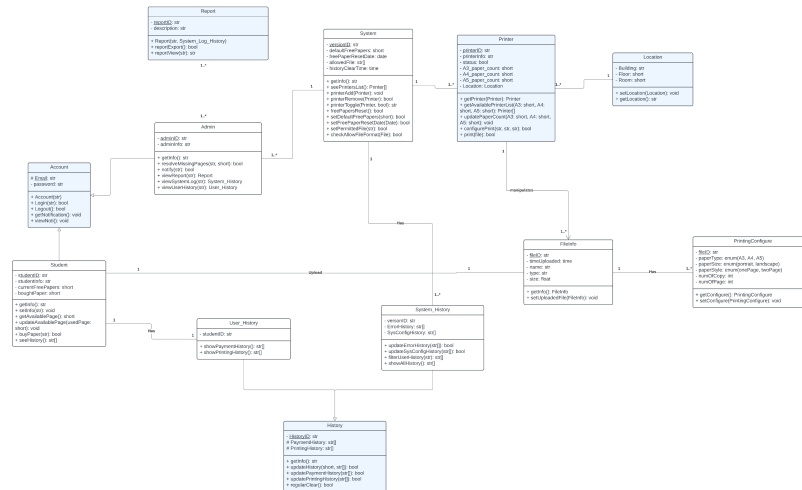


Figure 5: Class diagram for SPSS system

## 5 Thiết kế giao diện (Task 2.4)

### 5.1 Giao diện của sinh viên

#### 5.1.1 Giao diện trang chủ - trước Đăng nhập

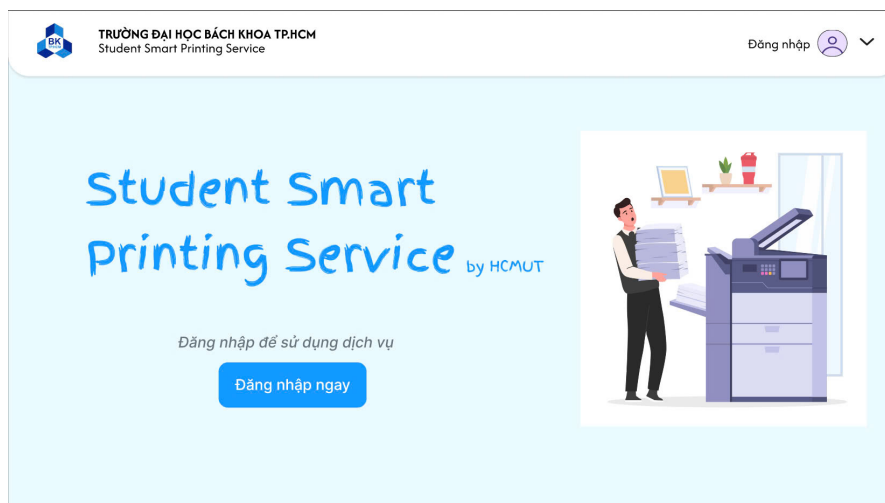


Figure 6: Trang chủ - trước Đăng nhập



### 5.1.2 Giao diện đăng nhập

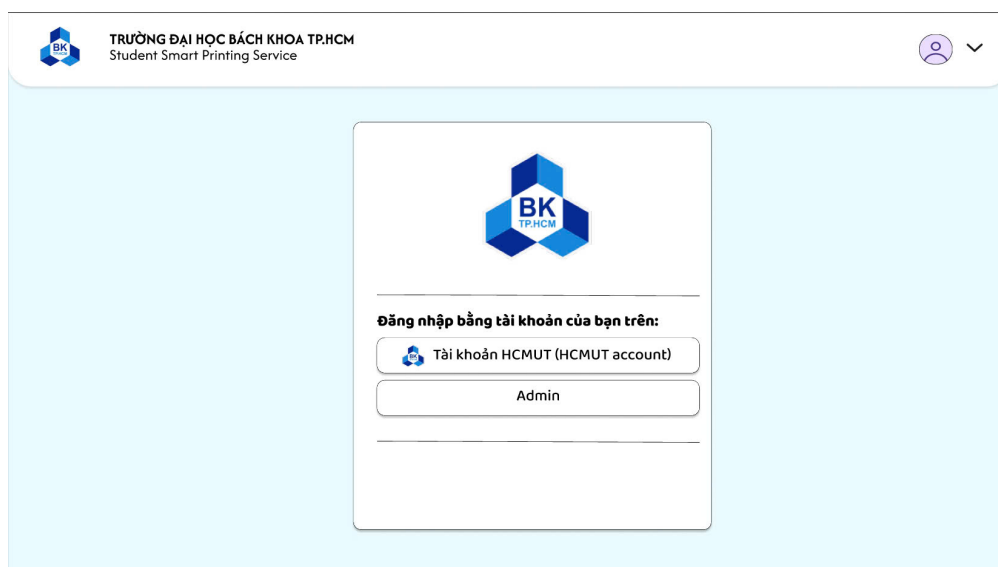


Figure 7: Sinh viên đăng nhập

### 5.1.3 Giao diện trang chủ - sau Đăng nhập



Figure 8: Trang chủ - sau Đăng nhập



#### 5.1.4 Giao diện dịch vụ in tài liệu

Figure 9: Dịch vụ in tài liệu

#### 5.1.5 Dịch vụ in tài liệu - Tải file

Figure 10: Tải file



### 5.1.6 Dịch vụ in tài liệu - Chọn thuộc tính và máy in

Figure 11: Chọn thuộc tính và máy in

### 5.1.7 Dịch vụ in tài liệu - In thành công

Figure 12: In thành công





### 5.1.8 Giao diện dịch vụ mua trang in

TRƯỜNG ĐẠI HỌC BÁCH KHOA TP.HCM  
Student Smart Printing Service

Lê Võ Đăng Khoa  
Student

### Mua trang in

Số trang in hiện có: 50

Số trang in cần mua thêm

Đơn giá: 200 đồng/ trang

Tổng tiền

Phương thức thanh toán

☐ Momo

☐ Ngân hàng

☒ BK Pay

☐ Tại nơi in

Thanh toán

<< QUAY VỀ TRANG CHỦ

Figure 13: Mua trang in

### 5.1.9 Giao diện lịch sử in

TRƯỜNG ĐẠI HỌC BÁCH KHOA TP.HCM  
Student Smart Printing Service

Lê Võ Đăng Khoa  
Student

### Lịch sử in

File in	Ngày in	Nơi nhận	Trạng thái
Giaitich3.pdf	22/09/2024	H1-321	Chưa in
Giaitich4.pdf	21/09/2024	H6-213	Đã in
Giaitich4.pdf	21/09/2024	H6-213	Đã in
Giaitich4.pdf	21/09/2024	H6-213	Đã in
Giaitich4.pdf	21/09/2024	H6-213	Đã in

Trang trước

1 / 5

Trang sau

<< QUAY VỀ TRANG CHỦ

Figure 14: Lịch sử in



### 5.1.10 Giao diện lịch sử thanh toán

TRƯỜNG ĐẠI HỌC BÁCH KHOA TP.HCM  
Student Smart Printing Service

Lê Võ Đăng Khoa  
Student

### Lịch sử thanh toán

Lọc kết quả

Q Tìm kiếm

Số lượng trang A4	Ngày mua	Số tiền	Trạng thái
10 trang	22/09/2024	2000 đồng	Đã thanh toán
20 trang	21/09/2024	4000 đồng	Đã thanh toán
50 trang	21/09/2024	10000 đồng	Đã thanh toán
30 trang	21/09/2024	6000 đồng	Thanh toán thất bại
100 trang	21/09/2024	20000 đồng	Đã thanh toán

Trang trước1 / 5Trang sau

<< QUAY VỀ TRANG CHỦ

Figure 15: Lịch sử thanh toán

## 5.2 Giao diện của SPSO

### 5.2.1 Giao diện đăng nhập

Central Authentication Service

### Enter your Username and Password

Username

abc.def

Password

\*\*\*\*\*

☐ Warn me before logging me into other sites

LoginClear

[Change password?](#)

Languages

[Vietnamese](#) [English](#)

Please note

The Login page enables single sign-on to multiple websites at HCMUT. This means that you only have to enter your user name and password once for websites that subscribe to the Login page.  
You will need to use your HCMUT Username and password to login to this site. The "HCMUT" account provides access to many resources including the HCMUT Information System, e-mail, ...  
For security reasons, please Exit your web browser when you are done accessing services that require authentication!

Technical support

E-mail: [support@hcmut.edu.vn](mailto:support@hcmut.edu.vn) Tel: (84-8) 38647256 - 5200

Copyright © 2011 - 2012 Ho Chi Minh University of Technology. All rights reserved.

Figure 16: SPSO đăng nhập



## 5.2.2 Giao diện quản lý máy in

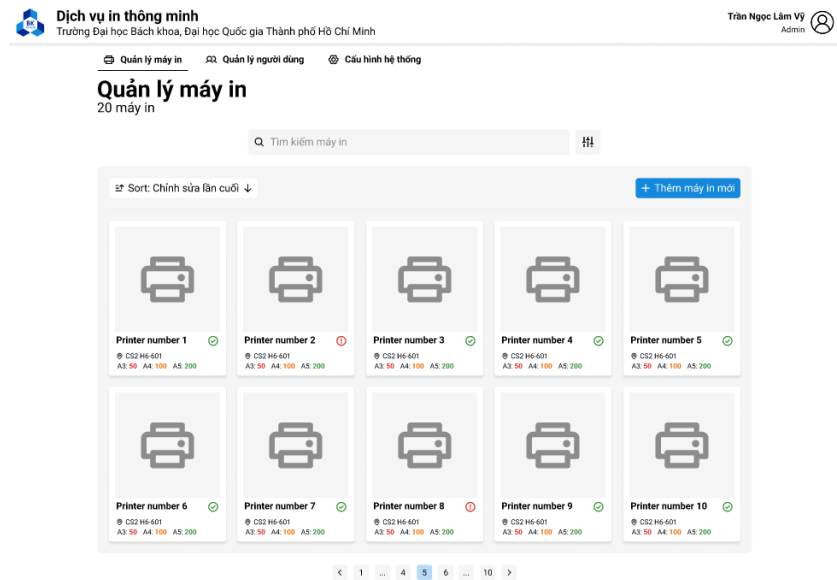


Figure 17: Quản lý máy in

## 5.2.3 Giao diện quản lý chi tiết máy in

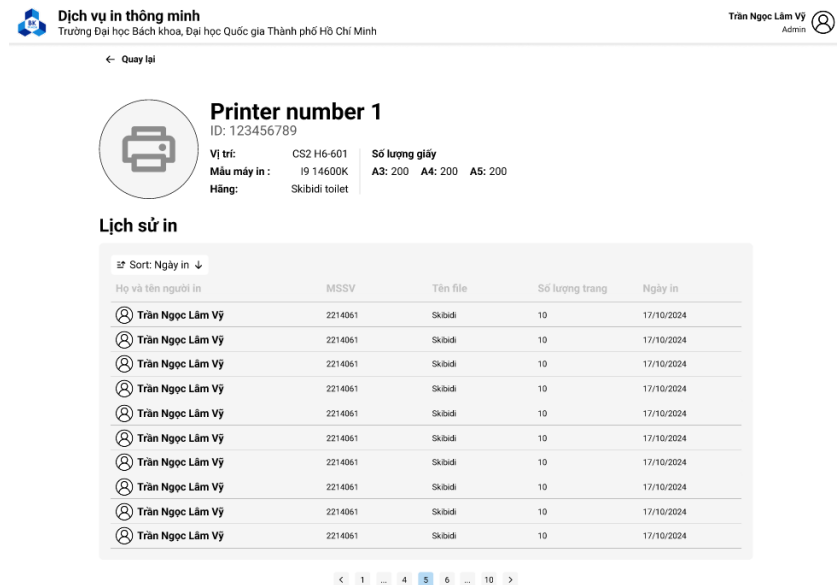


Figure 18: Quản lý chi tiết máy in



#### 5.2.4 Giao diện quản lý người dùng

**Dịch vụ in thông minh**  
Trường Đại học Bách khoa, Đại học Quốc gia Thành phố Hồ Chí Minh

Trần Ngọc Lâm Vỹ  
Admin

Quản lý máy in Quản lý người dùng Cấu hình hệ thống

### Quản lý người dùng

20 người dùng

Q Tìm kiếm người dùng

Sort: Lăn in gần đây ↓				
Họ và tên	MSSV	SL giấy miễn phí	SL giấy trả phí	Lăn in gần nhất
Trần Ngọc Lâm Vỹ	2214061	20	10	17/10/2024
Trần Ngọc Lâm Vỹ	2214061	20	10	17/10/2024
Trần Ngọc Lâm Vỹ	2214061	20	10	17/10/2024
Trần Ngọc Lâm Vỹ	2214061	20	10	17/10/2024
Trần Ngọc Lâm Vỹ	2214061	20	10	17/10/2024
Trần Ngọc Lâm Vỹ	2214061	20	10	17/10/2024
Trần Ngọc Lâm Vỹ	2214061	20	10	17/10/2024
Trần Ngọc Lâm Vỹ	2214061	20	10	17/10/2024
Trần Ngọc Lâm Vỹ	2214061	20	10	17/10/2024
Trần Ngọc Lâm Vỹ	2214061	20	10	17/10/2024

< 1 ... 4 5 6 ... 10 >


Figure 19: Quản lý người dùng

#### 5.2.5 Giao diện quản lý chi tiết người dùng

**Dịch vụ in thông minh**  
Trường Đại học Bách khoa, Đại học Quốc gia Thành phố Hồ Chí Minh

Trần Ngọc Lâm Vỹ  
Admin

← Quay lại



**Trần Ngọc Lâm Vỹ**  
@Nolvy  
MSSV: 2214061  
Email: vy.tranngoclamy@hcmut.edu.vn  
SDT: 1234567890  
[Chỉnh sửa thông tin người dùng](#)

**Số trang miễn phí**  
**20**

**Số trang trả phí**  
**0**

### Lịch sử in

Sort: Ngày in ↓				
Tên máy in	Vị trí máy in	Tên file	Số lượng trang	Ngày in
Printer number 1	CS2 H6-601	Skibidi	10	17/10/2024
Printer number 1	CS2 H6-601	Skibidi	10	17/10/2024
Printer number 1	CS2 H6-601	Skibidi	10	17/10/2024
Printer number 1	CS2 H6-601	Skibidi	10	17/10/2024
Printer number 1	CS2 H6-601	Skibidi	10	17/10/2024
Printer number 1	CS2 H6-601	Skibidi	10	17/10/2024
Printer number 1	CS2 H6-601	Skibidi	10	17/10/2024
Printer number 1	CS2 H6-601	Skibidi	10	17/10/2024
Printer number 1	CS2 H6-601	Skibidi	10	17/10/2024
Printer number 1	CS2 H6-601	Skibidi	10	17/10/2024

< 1 ... 4 5 6 ... 10 >

Figure 20: Quản lý chi tiết người dùng



### 5.2.6 Giao diện cấu hình hệ thống

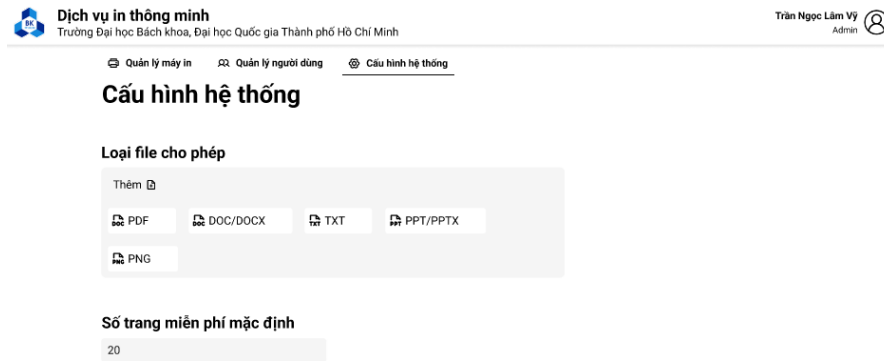


Figure 21: Cấu hình hệ thống

## 6 Thiết kế kiến trúc (Task 3)

### 6.1 Layered architecture

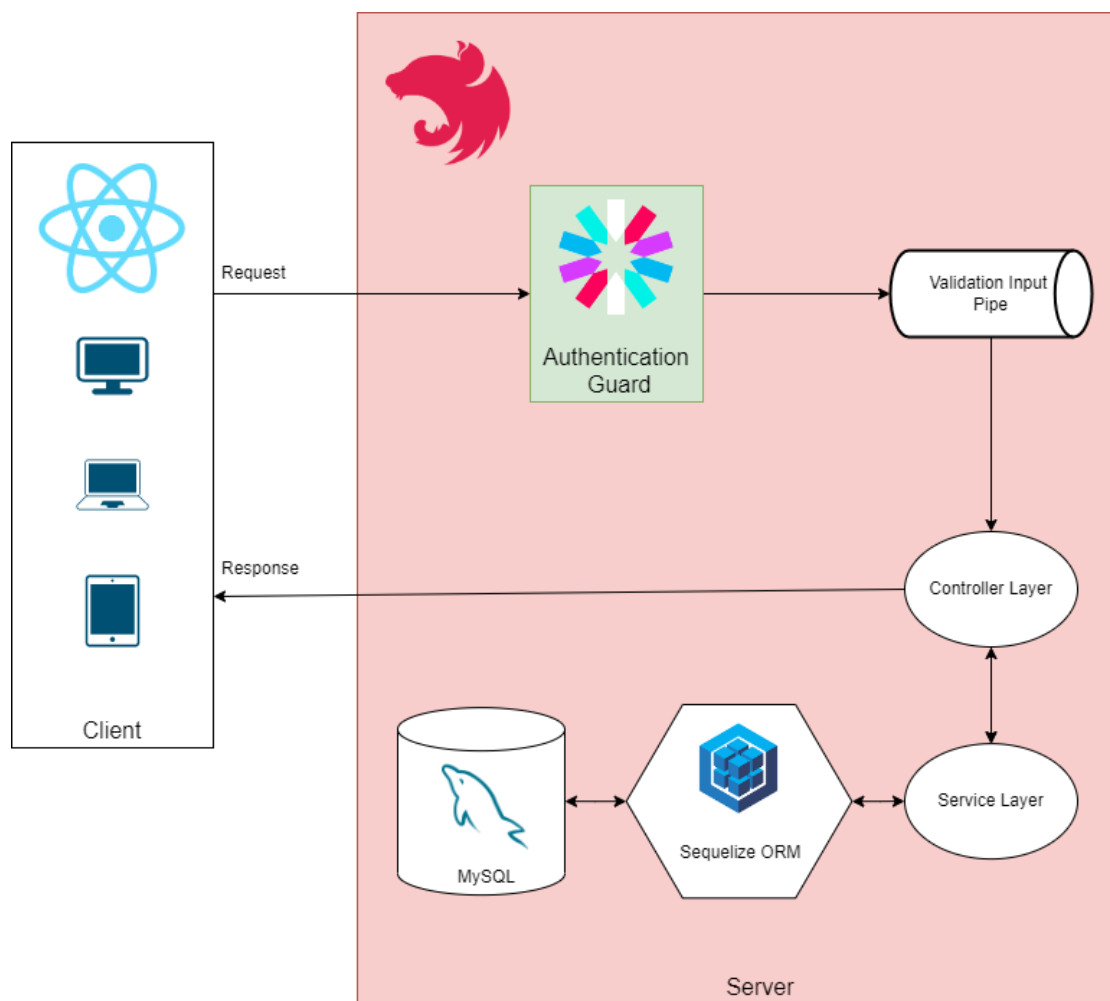


Figure 22: Architectural diagram

### 6.2 Chiến lược

#### 6.2.1 Chiến lược hiển thị giao diện người dùng

Hệ thống dịch vụ in ấn thông minh HCMUT\_SPSS được thiết kế nhằm đáp ứng nhu cầu ngày càng tăng về việc in ấn tài liệu của sinh viên và học viên tại Trường Đại học Bách Khoa TP.HCM. Để đảm bảo tính tiện lợi và hiệu quả trong việc sử

dụng, chiến lược trình bày của hệ thống sẽ tập trung vào việc tạo ra một giao diện người dùng thân thiện, dễ sử dụng, đồng thời cung cấp đầy đủ các chức năng cần thiết cho cả người dùng sinh viên và cán bộ quản lý.

## 1. Giao diện Người Dùng (UI)

- **Giao diện cho Sinh viên:** Giao diện dành cho sinh viên sẽ được thiết kế đơn giản và trực quan, giúp họ dễ dàng thực hiện các thao tác in ấn từ bất kỳ đâu trong khuôn viên trường. Các thành phần chính của giao diện bao gồm:
  - (a) *Tải File Lên:* Sinh viên sẽ có một nút "Tải lên" để chọn và tải tài liệu cần in. Hệ thống sẽ hỗ trợ nhiều định dạng file phổ biến như PDF, DOCX và hình ảnh.
  - (b) *Chọn Máy In:* Sau khi tải file, sinh viên có thể chọn máy in gần nhất thông qua một danh sách hoặc bản đồ hiển thị vị trí máy in trong trường. Hệ thống sẽ tự động liệt kê các máy in khả dụng dựa trên vị trí của người dùng.
  - (c) *Cấu Hình In:* Giao diện sẽ cung cấp các tùy chọn để sinh viên điều chỉnh các thuộc tính in như kích thước giấy, số trang in, chế độ in (một mặt hoặc hai mặt), và số bản in cần thiết. Các tùy chọn này sẽ được trình bày một cách dễ hiểu, với các hướng dẫn ngắn gọn đi kèm.
  - (d) *Xem Lịch Sử In:* Sinh viên có thể dễ dàng truy cập vào phần lịch sử in của mình, cho phép họ xem lại số trang và kích thước trang đã in trong một khoảng thời gian nhất định. Thông tin này sẽ giúp sinh viên quản lý tài khoản in của mình hiệu quả hơn.
  - (e) *Mua Thêm Trang In:* Nếu tài khoản của họ không đủ trang in, sinh viên có thể truy cập vào phần mua thêm trang in. Hệ thống sẽ hướng dẫn họ qua quy trình thanh toán trực tuyến một cách mượt mà.

- **Giao diện cho Cán bộ Quản lý:** Giao diện dành cho cán bộ quản lý sẽ phức tạp hơn, cung cấp các công cụ cần thiết để họ quản lý hệ thống in ấn một cách hiệu quả. Các thành phần chính bao gồm:

- (a) *Quản Lý Máy In:* Cán bộ quản lý có thể xem danh sách tất cả các máy in trong hệ thống, với các tùy chọn để thêm, kích hoạt, vô hiệu hóa hoặc loại bỏ máy in. Giao diện sẽ cho phép họ dễ dàng cập nhật thông tin về máy in, bao gồm tình trạng hoạt động và vị trí.
- (b) *Quản Lý Sinh Viên:* Cán bộ quản lý có thể truy cập vào danh sách sinh viên, theo dõi lịch sử in của họ và xem các báo cáo sử dụng. Họ có thể tìm kiếm sinh viên theo tên hoặc mã sinh viên, giúp việc quản lý trở nên thuận tiện hơn.
- (c) *Quản Lý Thông Số Cấu Hình In:* Giao diện sẽ cho phép cán bộ quản lý điều chỉnh các thông số cấu hình in như số trang mặc định cho mỗi tài khoản, ngày cung cấp số trang miễn phí và các loại file được phép tải lên hệ thống. Điều này giúp đảm bảo hệ thống hoạt động hiệu quả và đáp ứng được nhu cầu của người dùng.
- (d) *Báo Cáo và Phân Tích:* Cán bộ quản lý có khả năng xem báo cáo chi tiết về lịch sử in, sử dụng thống kê để phân tích xu hướng in ấn của người dùng. Hệ thống sẽ tự động tạo báo cáo tháng và năm, giúp họ dễ dàng nắm bắt được tình hình sử dụng hệ thống.

2. **Trải nghiệm Người Dùng (UX):** Nhóm sẽ tập trung vào việc tối ưu hóa trải nghiệm người dùng thông qua việc giảm thiểu số bước cần thiết để hoàn thành một quy trình in. Hệ thống sẽ cung cấp hướng dẫn trực quan và thông báo phản hồi ngay lập tức, giúp người dùng hiểu rõ hơn về các bước tiếp theo. Ví dụ, khi người dùng tải file lên, hệ thống sẽ xác nhận ngay lập tức liệu file có đủ điều kiện để in hay không, và nếu không, sẽ cung cấp thông tin chi tiết về lý do.



3. **Tích hợp Dịch vụ Bên Ngoài:** Hệ thống sẽ tích hợp với dịch vụ xác thực HCMUT\_SSO để đảm bảo rằng người dùng có thể dễ dàng đăng ký và đăng nhập. Điều này không chỉ tăng cường tính bảo mật mà còn tạo điều kiện thuận lợi cho việc quản lý tài khoản người dùng. Ngoài ra, tính năng thanh toán sẽ được tích hợp với BKPay, cho phép người dùng mua thêm trang in một cách nhanh chóng và an toàn.

### 6.2.2 Chiến lược quản lý API và lưu trữ dữ liệu

1. **Về nguyên tắc:** Server được xây dựng theo nguyên tắc RESTful API giúp các hệ thống có thể trao đổi dữ liệu qua HTTP một cách dễ dàng và thống nhất. Các phương thức HTTP thường được sử dụng trong RESTful API bao gồm:

- GET: Lấy dữ liệu từ server.
- POST: Gửi dữ liệu đến server để tạo tài nguyên mới.
- PUT: Cập nhật dữ liệu hiện có.
- DELETE: Xóa tài nguyên trên server

2. **Về kiến trúc:**

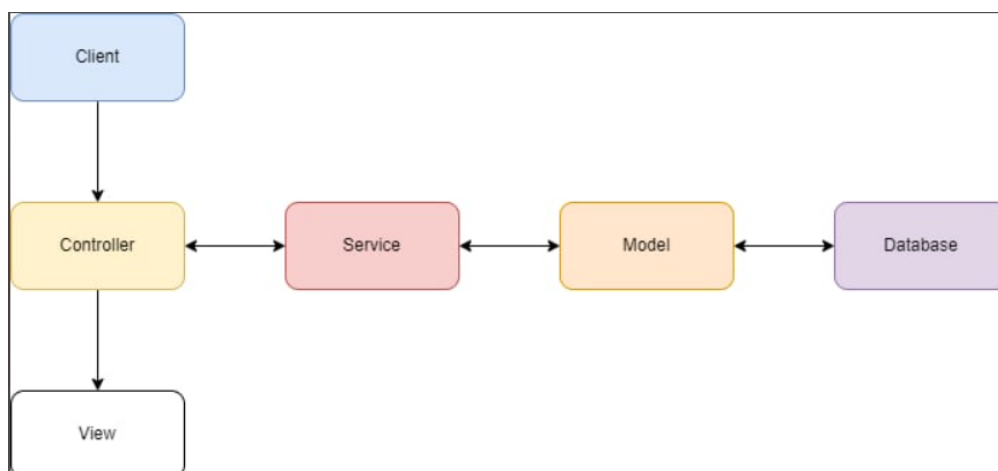


Figure 23: MVC architecture

Server được xây dựng theo kiến trúc MVC trên nền Monolithic. Trong đó, Monolithic là một kiến trúc phần mềm trong đó toàn bộ ứng dụng được tích hợp vào một khối mã duy nhất, bao gồm tất cả các thành phần từ giao diện người dùng, logic nghiệp vụ, đến các thao tác cơ sở dữ liệu. Tất cả các chức năng của ứng dụng được đóng gói trong một đơn vị triển khai duy nhất. MVC (Model-View-Controller) là một mẫu kiến trúc phần mềm được sử dụng rộng rãi trong việc phát triển ứng dụng để tạo ra các giao diện người dùng trực quan và có khả năng tương tác cao. MVC chia một ứng dụng thành 3 phần chính và mỗi phần có một vai trò riêng biệt:

- **Model** là lớp đại diện cho dữ liệu của ứng dụng. Nó có thể là một cơ sở dữ liệu, một file cấu hình hoặc một đối tượng phức tạp. Model chịu trách nhiệm lưu trữ, truy xuất và cập nhật dữ liệu.
- **View** là lớp giao diện người dùng. Nó hiển thị dữ liệu từ Model cho người dùng và cho phép người dùng tương tác với ứng dụng. View thường được xây dựng bằng các ngôn ngữ template như HTML, JSP hoặc React.
- **Controller** là lớp điều khiển luồng của ứng dụng. Nó nhận các yêu cầu từ người dùng, cập nhật Model và chọn View phù hợp để hiển thị.

Tuy nhiên kiến trúc MVC có một số chỉnh sửa khi thêm 1 lớp Service đệm giữa Controller và Model tách biệt phần xử lý logic trong Controller sang Service.

3. **Về tổ chức hệ thống:** Server được xây dựng bằng framework NestJS. Trong đó, NestJS là một framework để xây dựng các ứng dụng backend với Node.js, tập trung vào các nguyên tắc kiến trúc như modular hóa, Dependency Injection (DI), và SOLID. Nó được xây dựng dựa trên TypeScript và cung cấp nhiều tính năng hữu ích cho việc phát triển các ứng dụng web và API mạnh mẽ, có khả năng mở rộng. Module là một thành phần cốt lõi trong NestJS, được sử dụng để tổ chức cấu trúc ứng dụng thành các khối logic độc

lập. Một module trong NestJS được định nghĩa bằng một class với decorator `@Module()`. Class này được dùng để nhóm các controller, provider và có thể import các module khác. Mỗi ứng dụng NestJS sẽ có ít nhất một module gốc (thường là `AppModule`), nơi các module con được import vào để tạo thành ứng dụng hoàn chỉnh. Module giúp cô lập các chức năng, từ đó làm tăng tính bảo trì và mở rộng.

4. **Về xác thực người dùng:** Server sử dụng Token-based Authorization với JWT để xác thực yêu cầu người dùng. Token-based authorization là một phương pháp xác thực người dùng dựa trên việc phát hành các mã thông báo (token) mà người dùng sẽ sử dụng để truy cập tài nguyên. Thay vì sử dụng các thông tin nhạy cảm như tên người dùng và mật khẩu cho mỗi lần yêu cầu, hệ thống sẽ phát hành một token sau khi người dùng đăng nhập thành công. Token này sẽ đi kèm trong các yêu cầu tiếp theo và server sẽ kiểm tra token để xác thực quyền truy cập. JWT (JSON Web Token): Là loại token phổ biến nhất với cấu trúc dễ đọc, bao gồm ba phần: header, payload, và signature. JWT có thể chứa các thông tin xác thực và thời hạn của token. Quy trình sử dụng token thường bao gồm các bước sau:

- Đăng nhập: Người dùng gửi yêu cầu đăng nhập với thông tin tài khoản.
- Phát hành token: Nếu đăng nhập thành công, server sẽ phát hành một token cho người dùng.
- Gửi token trong yêu cầu: Người dùng gửi token trong phần header của các yêu cầu tiếp theo.
- Xác thực và kiểm tra token: Server xác thực token và kiểm tra thời hạn hoặc tính hợp lệ trước khi thực hiện yêu cầu.

5. **Về lưu trữ dữ liệu:** Server lưu trữ dữ liệu trong cơ sở dữ liệu MySQL và sử dụng công cụ Sequelize như một ORM để giao tiếp với cơ sở dữ liệu.

MySQL là một hệ quản trị cơ sở dữ liệu mã nguồn mở dựa trên mô hình quan hệ. Được sử dụng rộng rãi trong các ứng dụng web, MySQL nổi bật với hiệu suất cao, tối ưu hóa tốt cho các ứng dụng web và xử lý nhanh chóng. Tính tương thích của MySQL với nhiều nền tảng và tích hợp tốt với các ngôn ngữ lập trình phổ biến.

Sequelize là một Object-Relational Mapping (ORM) cho Node.js, cho phép tương tác với cơ sở dữ liệu SQL bằng cách sử dụng các đối tượng JavaScript. Một trong những điểm mạnh của Sequelize là dễ sử dụng, giảm thiểu việc viết các truy vấn SQL thủ công và giúp mã nguồn rõ ràng, dễ duy trì. Sequelize hỗ trợ việc tạo các mô hình dữ liệu rõ ràng và cấu trúc, giúp quản lý dữ liệu dễ dàng hơn. Nó cũng tương thích với nhiều hệ quản trị cơ sở dữ liệu khác nhau như MySQL.

### 6.3 Component diagram

Component diagram cho module in file:

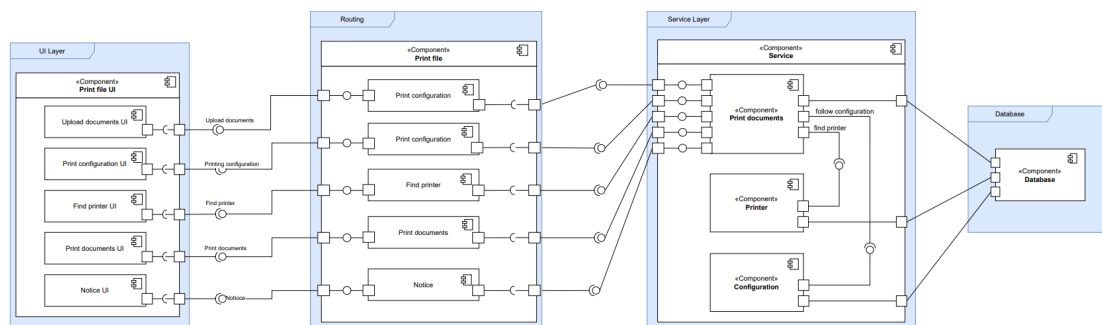


Figure 24: Component diagram for print file module

Hệ thống được dùng khi thực hiện việc in tài liệu sẽ gồm 4 component lớn:

1. **Print file UI:** Bao gồm các component nhỏ dùng cho việc hiển thị cho người dùng trong quá trình in file như đăng tải file in, cấu hình file in, tìm kiếm máy in, thực hiện in và hiển thị các thông báo cần thiết. Các component sẽ

thông qua các API tương ứng để giao tiếp và nhận dữ liệu từ database nhằm hiển thị lên giao diện thông qua lớp Print file.

2. **Print file:** Chứa các component cung cấp các hàm để thực hiện các tính toán logic của hệ thống và giao tiếp với các Service để xử lý dữ liệu.

- Component “Upload file” chịu trách nhiệm nhận file cũng như kiểm tra file in phù hợp với cấu hình của hệ thống.
- Component “Print configuration” chịu trách nhiệm nhận các cài đặt in của người dùng
- Component “Find Printer” chịu trách nhiệm tìm kiếm máy in phù hợp.
- Component “Page available” chịu trách nhiệm kiểm tra số lượng trang in của người dùng
- Component “Notice” chịu trách nhiệm hiển thị các thông báo cần thiết.

3. **Service:** Chứa các model component nhỏ gồm: print document, printer, configuration. Các service component sẽ được kết nối với database để xử lý dữ liệu.

4. **Database:** cơ sở dữ liệu của hệ thống.