

ĐẠI HỌC QUỐC GIA TP.HCM
TRƯỜNG ĐẠI HỌC BÁCH KHOA



BÁO CÁO BÀI TẬP LỚN
CÔNG NGHỆ PHẦN MỀM
(CO3001)

Đề tài: Smart Printing Service

GVHD: Lê Đình Thuân
Bùi Công Tuấn

Thành viên:	Nguyễn Ngọc Khánh My	-	2114094
	Nguyễn Trung Vương	-	2115337
	Đoàn Minh Hiếu	-	2110162
	Kim Nhật Thành	-	2112292
	Nguyễn Thái Thời	-	2114926
	Phạm Châu Thanh Tùng	-	2115235

Ho Chi Minh city, September 2023

Contents

I. REQUIREMENT ELICITATION	3
1 Domain context	3
2 Stakeholders and Their Current Needs & Benefits	3
3 Functional requirements	4
4 Non-functional requirements	5
5 Use-case diagram	7
5.1 Use-case diagram of whole system	7
5.2 Use-case diagram và use-case scenario cho In	8
5.3 Use-case diagram và use-case scenario cho Đăng nhập	14
5.4 Use-case diagram và use-case scenario cho Thông tin cá nhân	15
5.5 Use-case diagram và use-case scenario cho Quản lý tài liệu	19
5.6 Use-case diagram và use-case scenario cho Quản lý Sinh viên	21
5.7 Use-case diagram và use-case scenario cho Quản lý máy in	24
5.8 Use-case diagram và use-case scenario cho Gửi thông tin	27
II. SYSTEM MODELLING	29
1 Activity Diagram	29
1.1 Activity Diagram cho Quản lý Máy In	29
1.2 Activity Diagram cho Quản lý Sinh Viên	30
1.3 Activity Diagram cho In	31
2 Sequence Diagram	32
2.1 Sequence Diagram cho In	32
2.2 Sequence Diagram cho Upload Document	33
2.3 Sequence Diagram cho Tìm kiếm tài liệu (Search Document)	34
2.4 Sequence Diagram cho Mua trang in (Buy Page)	35
3 Class Diagram	36
4 MVP - Wireframe Details	43
4.1 Trang chủ khi chưa đăng nhập	43
4.2 Giao diện khi nhấn nút Đăng nhập	43
4.3 Giao diện khi đã đăng nhập	44
4.4 Giao diện khi đã nhấn nút In từ Kho cộng đồng hoặc cá nhân	44
4.5 Giao diện khi cấu hình cài đặt In	45
4.6 Giao diện khi mua trang in	46
4.7 Giao diện Chọn máy in và thời gian in	46
4.8 Giao diện khi Upload	46
4.9 Giao diện Theo dõi Lịch sử và tình trạng in	47
4.10 Giao diện quản lý sinh viên	47
4.11 Giao diện quản lý máy in	49
III ARCHITECTURE DESIGN	50
1 Architectural approach and Modules for the whole HCMUT-SSPS system	50
1.1 Architectural approach	50
1.2 List of Modules	51
2 Component Diagram	52
2.1 Mô tả Component Diagram	52
3 Deployment Diagram	56
IV IMPLEMENTATION SPRINT 1	57
1 Github for version control	57
2 Usability test	57
2.1 Participants	57
2.2 Environment	57
2.3 Tasks	57
2.4 Success Criteria	57
2.5 Feedback	58



2.5.1	Task 1: Đăng nhập vào hệ thống bằng email của trường Đại học Bách Khoa TP.HCM	58
2.5.2	Task 2: Upload tài liệu lên hệ thống	59
2.5.3	Task 3: Tìm kiếm tài liệu	59
2.5.4	Task 4: In tài liệu	60
2.5.5	Task 5: Mua trang in	61
2.5.6	Task 6: Các chức năng còn lại	62
2.5.7	Tổng quan	64



I REQUIREMENT ELICITATION

1 Domain context

Ứng dụng **Student Smart Printing Service** do nhóm **Baosranger** thực hiện là một ứng dụng mobile hỗ trợ cho việc in ấn, photo và lưu trữ tài liệu một cách tự động, nhanh chóng, thông minh, tiện lợi dành cho các sinh viên của trường Đại học Bách Khoa - TP.HCM (HCMUT).

Hiện tại, cộng đồng sinh viên HCMUT luôn có một lượng lớn nhu cầu in ấn, photo các loại tài liệu cũng như là muốn lưu trữ chúng một cách an toàn, nhanh chóng trên các nền tảng điện tử online. Thực tế vẫn có một số bất cập còn tồn tại như sau:

- Tình trạng các sinh viên phải xếp hàng dài, chờ đợi để đến lượt mình được quyền sử dụng máy tính và máy in thường gây mất một lượng lớn thời gian, có những lúc sinh viên phải tốn thời gian và công sức di chuyển để tìm đến các quầy photo in ấn ở bên ngoài trường.
- Để in ấn hay photo một file tài liệu, sinh viên cần phải upload nó lên một nền tảng lưu trữ trực tuyến nào đó từ trước (ví dụ: Google Drive, OneDrive) rồi sau đó đăng nhập vào các nền tảng lưu trữ này ở các máy tính tại quầy in ấn, photo. Điều này sẽ khiến rủi ro trong việc bảo mật các tài khoản cá nhân này tăng, có một số trường hợp người dùng quên việc phải đăng xuất sau khi dùng và bị những người xấu lợi dụng để chiếm đoạt tài liệu hoặc nhiều hơn.
- Việc luôn luôn cần vài người để trực tiếp quản lý đồng thời nhiều quầy in ấn, tài liệu gây ra sự bất đồng bộ và tốn hao chi phí, nhân lực nhiều hơn.
- Doanh thu định kì từ việc in ấn, photo không được hệ thống hóa và số hóa dẫn có thể dẫn đến sự thiếu chính xác và không trực quan.
- Việc định dạng và điều chỉnh bố cục in của tài liệu đa số phụ thuộc vào người quản lý máy in và thường không đạt 100% ý muốn của người sử dụng dịch vụ.

Chính vì các khó khăn trên, nhóm phác họa ý tưởng tạo nên một ứng dụng giúp sinh viên dễ dàng trong việc quản lý chi phí in ấn, photo của chính họ, thực hiện việc in ấn nhanh gọn, đơn giản hơn ngay trong khuôn viên trường Đại học Bách Khoa đồng thời lưu trữ, chia sẻ các tài liệu giữa các sinh viên trong trường Đại học Bách Khoa. Việc này không những tạo nên một dịch vụ tiện lợi và nhanh chóng chưa từng có trong việc in ấn, photo tài liệu mà còn tạo điều kiện cho các sinh viên tìm đến những nguồn tài liệu phong phú, giúp ích rất nhiều trong việc học cũng như có thể tạo ra một nền tảng điện tử để Nhà trường có thể dễ dàng quản lý cả về doanh thu lẫn tình trạng của dịch vụ.

2 Stakeholders and Their Current Needs & Benefits

• Students

- **Current needs:** Sinh viên cần in tài liệu một cách thuận tiện và hiệu quả. Họ yêu cầu sự linh hoạt trong việc lựa chọn máy in, khổ giấy và các tùy chọn in khác. Họ cũng cần theo dõi lịch sử in ấn của mình và mua thêm trang khi cần.

– Benefits

- * *Thuận tiện:* Sinh viên có thể dễ dàng in tài liệu của mình trong khuôn viên trường mà không gặp khó khăn khi tìm máy in có sẵn.
- * *Tùy chỉnh:* Sinh viên có thể chọn các tùy chọn in phù hợp với yêu cầu cụ thể của mình.
- * *Theo dõi:* Sinh viên có thể theo dõi lịch sử in ấn và quản lý số dư trang của mình.
- * *Tiết kiệm chi phí:* Bằng cách theo dõi số dư trang và mua thêm trang nếu cần, sinh viên có thể kiểm soát chi phí in ấn.

• Student Printing Service Officer (SPSO)

- **Current needs:** SPSO cần quản lý máy in một cách hiệu quả, cấu hình cài đặt hệ thống và giám sát các hoạt động in ấn của các máy in ở 2 cơ sở. Ngoài ra cũng cần các công cụ theo dõi và báo cáo về việc sử dụng hệ thống.

– Benefits



- * *Quản lý tập trung:* SPSO có thể dễ dàng quản lý và giám sát máy in ở các địa điểm khác nhau trong khuôn viên trường từ một giao diện duy nhất.
- * *Kiểm soát cấu hình:* Họ có thể linh hoạt cấu hình cài đặt hệ thống, chẳng hạn như giới hạn trang mặc định và loại tệp được phép.
- * *Thông tin chi tiết về hoạt động sử dụng:* SPSO thu thập được các thông tin chi tiết về các hành vi in ấn thông qua báo cáo, cho phép họ đưa ra quyết định một cách hiệu quả.

• Printers Maintenance Team

- **Current needs:** Đội ngũ chịu trách nhiệm bảo trì máy in cần một công cụ hiển thị lỗi của máy in để có thể giải quyết một cách nhanh chóng.
- **Benefits**
 - * *Cảnh báo theo thời gian thực:* Hệ thống có thể đưa ra cảnh báo theo thời gian thực về sự cố của máy in hoặc nguồn cung cấp sắp hết, đảm bảo việc bổ sung giấy/mực hoặc bảo trì một cách kịp thời.
 - * *Hiệu quả:* Nhóm có thể ưu tiên các nhiệm vụ bảo trì dựa trên dữ liệu nhận được, để tránh việc in của sinh viên bị gián đoạn.

• Payment System (BKPay)

- **Current needs:** BKPay cần bên trung gian để có thể liên kết tới hệ thống thanh toán của họ.
- **Benefits**
 - * *Hiệu quả:* Tích hợp với HCMUT-SSPS giúp đơn giản hóa quy trình thanh toán cho sinh viên.

3 Functional requirements

• Students:

- **User Authentication :** Sinh viên phải xác thực danh tính của mình thông qua HCMUT_SSO authentication service.
- **Printer Selection and Pickup Time :** Sinh viên có thể lựa chọn máy in để in ấn và chọn thời gian lấy tài liệu (dựa thời gian dự kiến của hệ thống).
- **Document Upload and Selection :** Sinh viên có thể tải lên các tài liệu cho mục đích in ấn hoặc sử dụng những tài liệu có sẵn của hệ thống được public bởi những người dùng khác và sàng lọc bởi SPSO.
- **Print Configuration :** Sinh viên có thể cài đặt cấu hình in, ví dụ như khổ giấy, số lượng trang cần in, số mặt, số bản copy.
- **Printing History and Statistics :** Sinh viên có thể truy cập lịch sử in cá nhân trong mỗi kì, cùng một bản thống kê số trang đã in với mỗi kích thước trang.
- **Personal Document Storage:** Sinh viên có thể in tài liệu trong kho lưu trữ cá nhân (giới hạn dung lượng lưu trữ cụ thể).
- **Printing Limitation:** Sinh viên chỉ được phép in một số trang không vượt quá số trang cho phép mỗi kì. (số trang dư sẽ reset mỗi kì).
- **Purchase More Printing Pages:** Sinh viên có quyền lựa chọn mua thêm trang in khi dùng hết số trang miễn phí và thanh toán thông qua BKPay (bill sẽ tự động chuyển sang BKPay để vào đó thanh toán).
- **Feedback and Ratings :** Sinh viên có thể cung cấp phản hồi hoặc đánh giá về dịch vụ in ấn.
- **Printing Status Check:** Sinh viên của hệ kiểm tra tình trạng in (chưa in, in thành công, in thất bại).

• Student Printing Service Officer (SPSO):

- **Printer Management :** SPSO có khả năng thêm, bật hoặc tắt máy in trong hệ thống.



- **File Type Customization** : SPSO có thể tùy chỉnh các loại tệp sinh viên được phép tải lên để in.
- **Default Page Count and Page Distribution Date** : SPSO có thể thay đổi số trang mặc định và ngày hệ thống cung cấp số trang mặc định đó cho toàn bộ sinh viên.
- **View Printing History**: SPSO có thể xem lịch sử in của tất cả sinh viên hoặc của một sinh viên cụ thể trong một khoảng thời gian bất kỳ đối với tất cả hoặc một số máy in nhất định.
- **Access Usage Reports**: SPSO có khả năng truy cập các báo cáo về việc sử dụng hệ thống in được tạo tự động cuối mỗi tháng, mỗi năm và lưu trữ trong hệ thống.
- **Sell Printing Pages** : SPSO có thể bán thêm số lượng trang in cho sinh viên có nhu cầu.
- **View Service Reviews** : SPSO có thể xem đánh giá về dịch vụ in ấn.
- **User Suspension**: SPSO có khả năng cấm truy cập các sinh viên có hành vi phá hoại.
- **View Revenue** : SPSO có thể xem doanh thu được tạo ra từ các dịch vụ in ấn.
- **Document Storage Management** : SPSO có thể quản lý kho lưu trữ tài liệu của hệ thống và cá nhân.
- **Confirmation of Printing Status** : SPSO có thể xác nhận với người dùng đã in thành công/thất bại.

4 Non-functional requirements

- **Tính khả dụng (Usability)**

- Người dùng có thể sử dụng hệ thống sau 15 phút hướng dẫn.
- Responsive app có kích thước thay đổi sao cho phù hợp với các kích thước khác nhau ở nhiều mẫu điện thoại và máy tính bảng.

- **Tính tin cậy (Reliability)**

- Thời gian ngừng hoạt động của hệ thống mỗi tháng không quá 1 giờ.
- Tất cả các lỗi mà hệ thống gặp phải, phải được ghi lại và người dùng phải nhận được thông báo lỗi rõ ràng trong vòng 10 giây.

- **Tốc độ, hiệu năng (Performance)**

- Thời gian phản hồi của 80% transaction (các hoạt động in ấn) dài không quá 1500 mili giây kể từ lúc user gửi yêu cầu.
- Hệ thống sẽ tải các thành phần trang, chẳng hạn như hình ảnh và tập lệnh, trong vòng 1-2 giây.
- Có khả năng xử lý 2000 yêu cầu trong cùng một thời điểm.
- 1000 người có thể truy cập hệ thống cùng lúc.

- **Tính bền vững của hệ thống (Robustness)**

- Mất hoặc hỏng dữ liệu sẽ xảy ra ở tỷ lệ thấp hơn 0,01% trong tất cả các hoạt động in.
- Thời gian khởi động lại mà hệ thống cần không quá 60 giây.

- **Tính mở rộng (Scalability)**

- Có thể thêm các module mới, chỉnh sửa nâng cấp tính năng các module cũ mà vẫn không thay đổi cấu trúc ban đầu của app.
- Có khả năng nâng cấp, cải tiến app để mở rộng quy mô và phạm vi người dùng (phạm vi toàn quốc và toàn cầu).

- **Tính bảo trì & sao lưu dữ liệu (Maintainability & Backup)**

- Hệ thống sẽ được bảo trì, sao lưu dữ liệu thường xuyên, 1 lần mỗi 3 tháng, mỗi lần không quá 3 tiếng và không trùng vào giờ hành chính.



- Hệ thống luôn ghi lại các quy trình khôi phục đề phòng cần khôi phục hệ thống về trạng thái chức năng trong trường hợp mất dữ liệu hoặc gặp lỗi hệ thống.
- Hệ thống luôn ghi nhận các feedback từ người dùng về các vấn đề trên hệ thống, sẵn sàng cho các tình huống bảo trì khẩn cấp khi có những lỗi nghiêm trọng phát sinh.

- **Tính đa nền tảng (Portability)**

- Có thể chạy ổn định ở cả hệ điều hành Android lẫn IOS, các tính năng đảm bảo sự đồng bộ, nhất quán.

- **Tính pháp lý (Compliance)**

- Đảm bảo quyền sở hữu trí tuệ: Không vi phạm bản quyền, thương hiệu của bên thứ ba.
- Đảm bảo quyền riêng tư: Dữ liệu cá nhân của người dùng được bảo mật.

- **Bảo mật an ninh (Security)**

- Mật khẩu của mỗi tài khoản yêu cầu ít nhất 8 ký tự, phải bao gồm chữ cái viết hoa, ký tự đặc biệt và chữ số.
- Trong trường hợp người dùng đăng nhập sai quá 6 lần sẽ tự động khóa tài khoản, để mở khóa sẽ cần liên hệ với quản trị viên.
- Có lựa chọn quên mật khẩu bằng cách xác nhận thông qua mã OTP gửi về email.

5 Use-case diagram

Xem các bản vẽ Use-case diagram FULL của nhóm tại [đây](#)

5.1 Use-case diagram of whole system

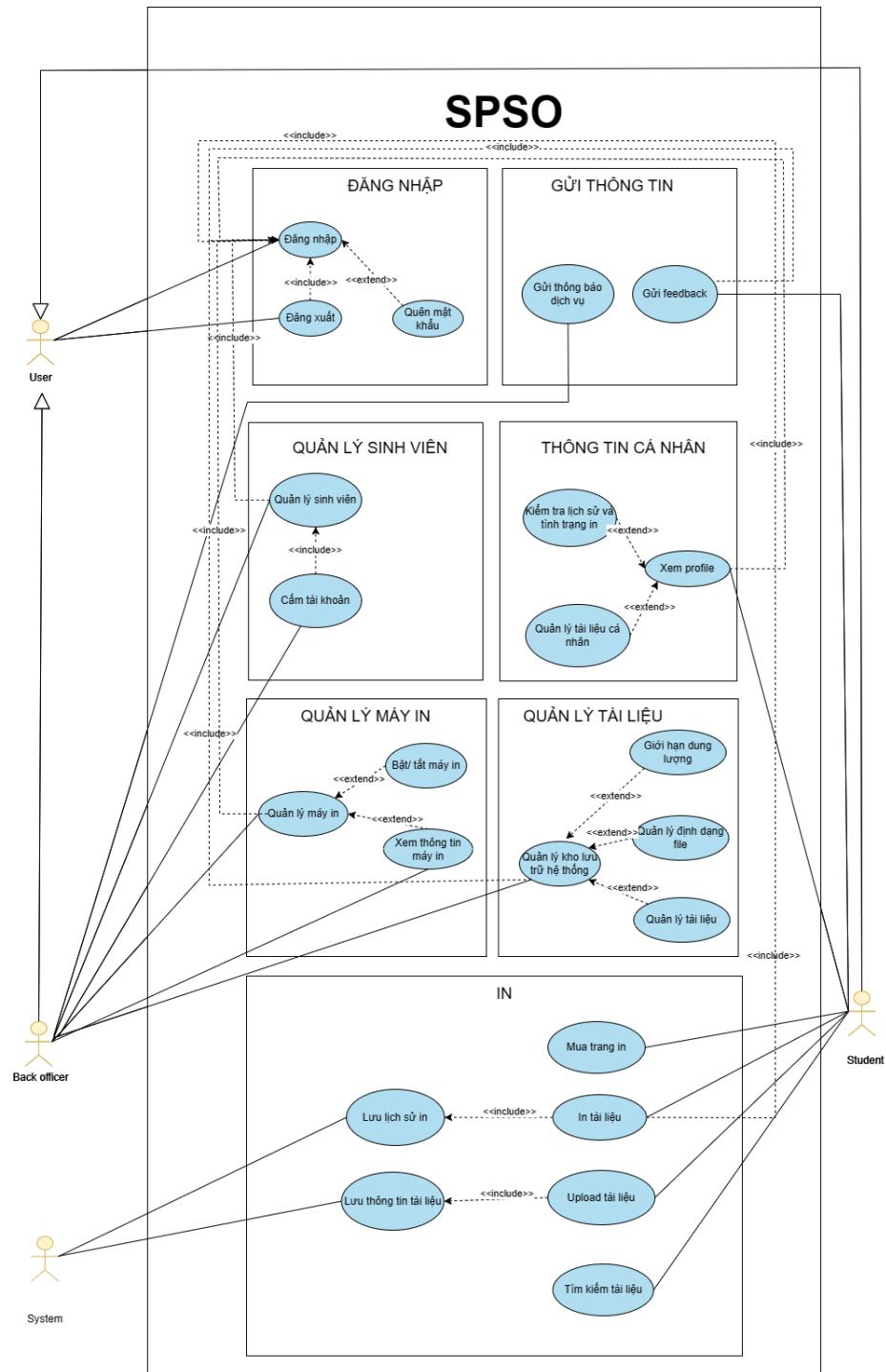


Figure 1: Use-case diagram Hệ thống

5.2 Use-case diagram và use-case scenario cho In

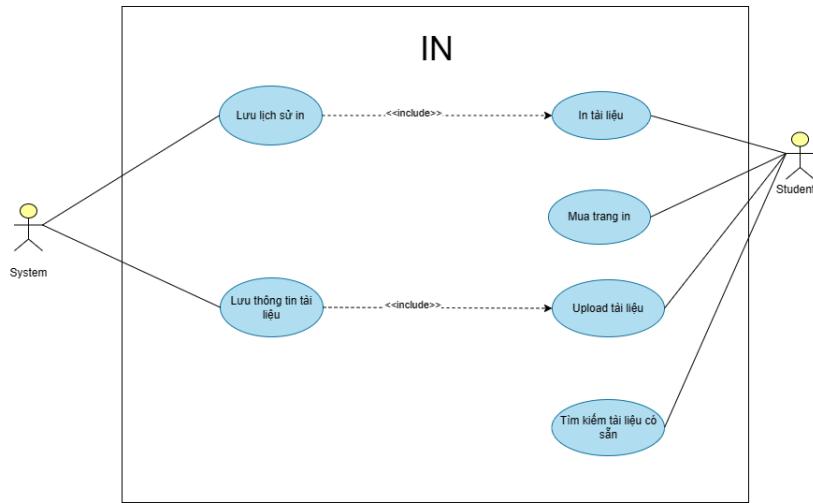


Figure 2: Use-case diagram cho In

Đặc tả use-case In tài liệu

Use-case name	In tài liệu
Actors	Student
Description	Là một người dùng, tôi muốn in tài liệu.
Trigger	Người dùng xác nhận file để in(từ kho cá nhân/ kho cộng đồng).
Preconditions	<ul style="list-style-type: none"> Người dùng đã đăng nhập vào hệ thống và đang ở trang chủ chính của hệ thống. File mà người dùng chọn để in tồn tại trong kho lưu trữ. Hệ thống đang hoạt động bình thường.
Postconditions	Hệ thống thông báo in thành công, đồng thời thông tin in và tình trạng in sẽ được lưu lại trong lịch sử in.
Normal Flow	<ol style="list-style-type: none"> Hệ thống mở giao diện cài đặt cấu hình in. Người dùng cài đặt cấu hình in. Người dùng chọn máy in. Người dùng chọn thời gian lấy tài liệu. Người dùng Xác nhận in. Hệ thống duyệt các thông tin và xử lý.
Alternative Flow	<ul style="list-style-type: none"> Tại bước 3,4,5 <ul style="list-style-type: none"> A.1 Người dùng chọn quay lại. A.2 Hệ thống quay lại bước 1.



Exception Flow	<ul style="list-style-type: none">• Tại bước 2,3,4,5<ul style="list-style-type: none">A.1 Người dùng chọn Hủy.A.2 Hệ thống hủy tác vụ Print, thoát khỏi giao diện Print và quay về giao diện màn hình chính.A.3 Usecase dừng lại.• Tại bước 2<ul style="list-style-type: none">2.1 Hệ thống xác nhận người dùng không còn đủ số trang cho phép để in.2.2 Hệ thống yêu cầu người dùng mua thêm trang in.2.3 Người dùng chọn mua thêm trang in.2.4 Chuyển sang usecase Mua trang in.• Tại bước 6<ul style="list-style-type: none">6.1 Hệ thống xử lý dữ liệu lỗi.6.2 Hệ thống thông báo in thất bại.6.3 Hệ thống hủy tác vụ In, thoát khỏi giao diện In và quay về giao diện màn hình chính.6.4 Usecase dừng lại.
Constraints	Không có.

Đặc tả use-case Mua trang in

Use-case name	Mua trang in
Actors	Student
Description	Là một người dùng, tôi muốn mua thêm trang in để phục vụ cho nhu cầu in ấn.
Trigger	Người dùng bấm vào mục Mua trang in tại menu chính hoặc tại bước chọn cấu hình in của tác vụ In (khi hệ thống thông báo người dùng không còn đủ số trang in có sẵn).
Preconditions	<ul style="list-style-type: none">• Người dùng đã đăng nhập vào hệ thống.• Hệ thống đang hoạt động bình thường.
Postconditions	Hệ thống thông báo thanh toán thành công và cấp trang in thêm cho người dùng.
Normal Flow	<ol style="list-style-type: none">1. Hệ thống mở giao diện mua hàng.2. Người dùng chọn số trang cần mua(không vượt quá số trang hệ thống có thể cấp).3. Người dùng xác nhận đơn hàng, cam kết chịu trách nhiệm.4. Hệ thống xử lý thông tin và tạo hóa đơn chuyển tới BKPay.5. Hệ thống cập nhật số trang của người dùng, cập nhật lịch sử giao dịch.



Alternative Flow	<ul style="list-style-type: none">• Tại bước 33.1 Người dùng chọn Hủy.3.2 Hệ thống quay lại bước 2.
Exception Flow	<ul style="list-style-type: none">• Tại bước 44.1 Hệ thống xử lý thông tin thất bại.4.2 Hệ thống thông báo ra màn hình, hủy tác vụ và quay về màn hình chính.4.3 Usecase dừng lại.
Constraints	Không có.

Dặc tả use-case Tìm kiếm tài liệu có sẵn

Use-case name	Tìm kiếm tài liệu có sẵn
Actors	Student
Description	Là một student, tôi muốn chọn tài liệu để in từ trong kho tài liệu.
Trigger	Student chọn tìm kiếm tài liệu Từ kho hệ thống hoặc Từ kho cá nhân
Preconditions	<ul style="list-style-type: none">• Người dùng đăng nhập thành công vào hệ thống.
Postconditions	Màn hình sẽ hiển thị các tài liệu từ kho lưu trữ có liên quan đến thông tin tìm kiếm của sinh viên.
Normal Flow	<ol style="list-style-type: none">1. Người dùng nhập tên tài liệu cần in2. Hệ thống sẽ tìm kiếm trong và lọc ra những tài liệu có liên quan đã được in trước đó.3. Hệ thống hiện thị lên màn hình thông tin tài liệu và các thao tác cho người dùng.
Alternative Flow	Không có
Exception Flow	<ul style="list-style-type: none">• Tại bước 22.1 Hệ thống không tìm thấy tài liệu cần tìm2.2 Hệ thống in ra màn hình không tìm thấy tài liệu.2.3 Use-case dừng lại• Tại bước 33.1 Người dùng chọn các thao tác in3.2 Hệ thống chuyển sang giao diện của use-case in tài liệu.3.3 Use-case dừng lại.
Constraints	Kho lưu trữ cá nhân chỉ lưu lại tài liệu trong thời gian một học kỳ.

**Đặc tả use-case Tải tài liệu**

Use-case name	Tải tài liệu
Actors	Student
Description	Là một người dùng, tôi muốn tải tài liệu lên kho lưu trữ cá nhân hoặc kho lưu trữ chung.
Trigger	Ấn vào biểu tượng Tải lên trên màn hình giao diện chính.
Preconditions	<ul style="list-style-type: none">• Người dùng đã đăng nhập vào hệ thống.• Hệ thống đang hoạt động bình thường.
Postconditions	Hệ thống thông báo tải lên thành công và cập nhật lại kho dữ liệu theo yêu cầu người dùng.
Normal Flow	<ol style="list-style-type: none">1. Hệ thống mở cửa sổ Tải tài liệu2. Người dùng chọn tài liệu muốn tải từ thiết bị.3. Hệ thống tải tài liệu lên.4. Người dùng nhập thông tin tài liệu(tên, môn học, vị trí lưu tài liệu).5. Người dùng xác nhận tài liệu tải lên.6. Hệ thống cập nhật lại kho lưu trữ.
Alternative Flow	Không có
Exception Flow	<ul style="list-style-type: none">• Tại bước 2, 4<ul style="list-style-type: none">A.1 Người dùng chọn HủyA.2 Hệ thống thoát khỏi giao diện Tải tài liệu và quay lại giao diện chínhA.3 Use-case dừng lại• Tại bước 6<ul style="list-style-type: none">6.1 Hệ thống thấy người dùng đã đạt mức giới hạn dung lượng cần tải6.2 Hệ thống thông báo cho người dùng, hủy tác vụ và quay về menu6.3 Use-case dừng lại
Constraints	<ul style="list-style-type: none">• File upload phải $\leq 50MB$• Định dạng bị giới hạn trong 3 loại doc, xsl và pdf

**Đặc tả use-case Lưu lịch sử in**

Use-case name	Lưu lịch sử in
Actors	System
Description	Hệ thống lưu trữ thông tin của tài liệu được đặt in và các thông tin liên quan và cơ sở dữ liệu.
Trigger	Khi Student ấn vào nút Xác nhận in ở trang In tài liệu.
Preconditions	<ul style="list-style-type: none">• Người dùng đã đăng nhập vào hệ thống và thực hiện chức năng In tài liệu.• Cả người dùng và hệ thống đang hoạt động.
Postconditions	Người dùng có thể kiểm tra thông tin in ấn ở mục Lịch sử và tình trạng in
Normal Flow	<ol style="list-style-type: none">1. Hệ thống tiếp nhận request in tài liệu từ Student2. Hệ thống lưu các thông tin của file tài liệu (tên file, link dẫn đến file, số trang, dung lượng), ngày đặt in, thời gian lấy.
Alternative Flow	Không có
Exception Flow	<ul style="list-style-type: none">• Tại bước 1<ol style="list-style-type: none">1.1 Hệ thống tiếp nhận thông tin thất bại.1.2 Hệ thống gửi phản hồi Tiếp nhận thông tin không thành công đến giao diện người dùng1.3 Use-case dừng lại• Tại bước 2<ol style="list-style-type: none">2.1 Hệ thống cập nhật dữ liệu thất bại2.2 Hệ thống gửi phản hồi Cập nhật dữ liệu không thành công đến giao diện người dùng2.3 Use-case dừng lại
Constraints	Không có

**Đặc tả use-case Lưu thông tin tài liệu**

Use-case name	Lưu thông tin tài liệu
Actors	System
Description	Hệ thống lưu trữ thông tin của tài liệu được tải lên vào kho lưu trữ, dựa vào lựa chọn của người dùng
Trigger	Khi người dùng xác nhận tài liệu để tải lên từ thiết bị ở trang Tải tài liệu .
Preconditions	<ul style="list-style-type: none">• Người dùng đã đăng nhập vào hệ thống và thực hiện chức năng Tải tài liệu• Cả người dùng và hệ thống đang hoạt động.
Postconditions	Xuất hiện thông tin dữ liệu ở các kho lưu trữ theo yêu cầu người dùng.
Normal Flow	<ol style="list-style-type: none">1. Hệ thống tiếp nhận thông tin được gửi đi từ người dùng.2. Hệ thống sẽ xem xét vị trí lưu trữ của tài liệu theo yêu cầu, sau đó sẽ lọc ra thông tin môn học để lưu lại.
Alternative Flow	Không có
Exception Flow	<ul style="list-style-type: none">• Tại bước 1<ol style="list-style-type: none">1.1 Hệ thống tiếp nhận thông tin thất bại.1.2 Hệ thống gửi phản hồi Tiếp nhận thông tin không thành công đến giao diện người dùng1.3 Use-case dừng lại• Tại bước 2<ol style="list-style-type: none">2.1 Hệ thống lưu trữ dữ liệu thất bại2.2 Hệ thống gửi phản hồi Lưu trữ dữ liệu không thành công đến giao diện người dùng2.3 Use-case dừng lại
Constraints	Kho lưu trữ cá nhân chỉ lưu lại tài liệu trong thời gian một học kỳ.

5.3 Use-case diagram và use-case scenario cho Đăng nhập

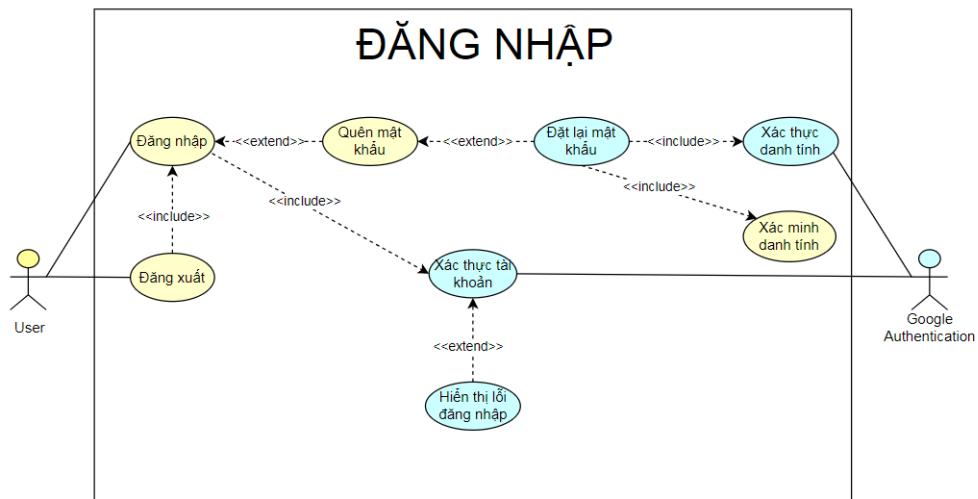


Figure 3: Use-case diagram cho Đăng nhập

Đặc tả use-case Đăng nhập

Use-case name	Đăng nhập
Actors	Student, Admin
Description	Người dùng muốn truy cập vào web cần sử dụng tài khoản, mật khẩu để đăng nhập.
Trigger	Ấn vào mục Đăng nhập tại trang chủ web.
Preconditions	<ul style="list-style-type: none"> Thiết bị người dùng phải kết nối với Internet. Người dùng phải có tài khoản được cấp sẵn.
Postconditions	Hệ thống duyệt thông tin đăng nhập, cho phép người dùng truy cập vào web và sử dụng các tính năng.
Normal Flow	<ol style="list-style-type: none"> Hệ thống mở giao diện đăng nhập. Người dùng nhập tài khoản và mật khẩu. Hệ thống kiểm tra thông tin đầu vào, và cho phép người dùng truy cập vào tài khoản.

Alternative Flow	<ul style="list-style-type: none"> • Tại bước 2 <ul style="list-style-type: none"> 2.1 Người dùng chọn quên mật khẩu. 2.2 Hệ thống chuyển tiếp tới Central Authentication Service. 2.3 Người dùng xác minh danh tính và đặt lại mật khẩu. 2.4 Hệ thống quay lại bước 1. • Tại bước 3 <ul style="list-style-type: none"> 3.1 Hệ thống kiểm tra tài khoản không hợp lệ. 3.2 Hệ thống quay lại bước 1.
Exception Flow	<ul style="list-style-type: none"> • Tại bước 3 <ul style="list-style-type: none"> 3.1 Hệ thống kiểm tra tài khoản nằm trong danh sách cấm. 3.2 Hệ thống thông báo "Tài khoản này đã bị cấm, vui lòng sử dụng tài khoản khác". 3.3 Use-case dừng lại. • Tại bước 3 <ul style="list-style-type: none"> 3.1 Người dùng nhập sai mật khẩu quá nhiều lần. 3.2 Hệ thống khóa tài khoản người dùng. 3.3 Use-case dừng lại.
Constraints	Không có.

5.4 Use-case diagram và use-case scenario cho Thông tin cá nhân

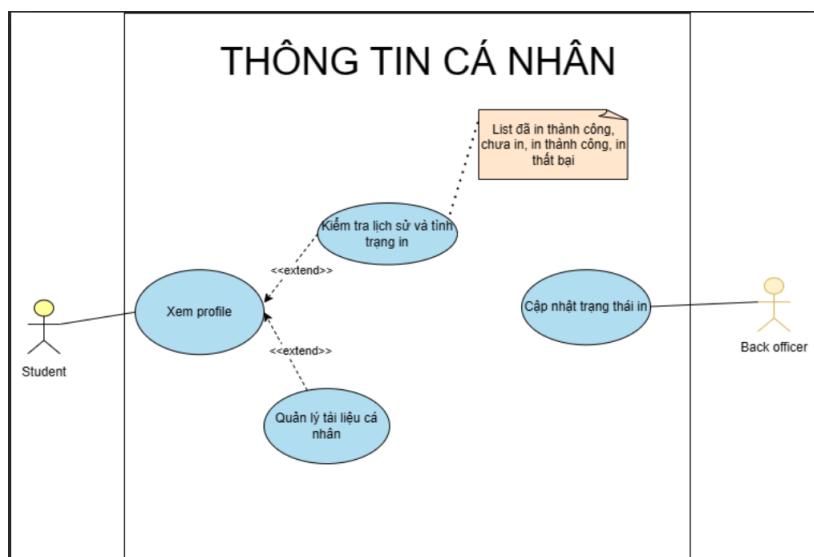


Figure 4: Use-case diagram cho Thông tin cá nhân người dùng



Dặc tả use-case Xem Profile

Use-case name	Xem profile
Actors	Student
Description	Là người dùng, tôi muốn theo dõi những thông tin của tài khoản và chỉnh sửa, thay đổi theo ý muốn.
Trigger	Người dùng nhấn vào mục Profile sau khi trỏ chuột vào thanh Tên tài khoản ở góc trên cùng bên phải
Preconditions	<ul style="list-style-type: none">• Thiết bị của người dùng phải kết nối với Internet.• Người dùng phải đăng nhập vào hệ thống.
Postconditions	Hệ thống sẽ hiển thị dữ liệu về thông tin cá nhân của tài khoản người dùng
Normal Flow	<ol style="list-style-type: none">1. Hệ thống hiển thị giao diện chứa những thông tin cá nhân của người dùng như Họ tên, Email, Số điện thoại, v.v...2. Người dùng bấm vào nút Edit ở bên cạnh các dòng thông tin để chỉnh sửa dữ liệu, 2 nút Save và Cancel ở góc dưới sẽ được phép sử dụng (sáng lên).3. Người dùng chọn nút Done.4. Người dùng chọn nút Save.5. Hệ thống hiện thông báo pop up đã lưu thành công
Alternative Flow	<ul style="list-style-type: none">• Tại bước 3<ul style="list-style-type: none">3.1 Người dùng chọn nút Cancel bên cạnh nút Done, thông tin mới được nhập trên thanh thông tin đó vào sẽ reset về thông tin cũ.3.2 Người dùng nhập thông tin khác định dạng so với định dạng thông tin của chú thích, nút Done và Save sẽ không khả dụng (bị mờ).• Tại bước 4<ul style="list-style-type: none">4.1 Người dùng chọn nút Cancel bên cạnh nút Save, toàn bộ thông tin mới ở tất cả các thanh được nhập vào sẽ reset về thông tin cũ.
Exception Flow	<ul style="list-style-type: none">• Tại bước 4<ul style="list-style-type: none">4.1 Hệ thống cập nhật dữ liệu thất bại.4.2 Hệ thống thông báo lỗi ra màn hình.4.3 Hệ thống quay lại giao diện menu.
Constraints	Thông tin nhập vào không được sai định dạng so với chú thích

**Đặc tả use-case Kiểm tra lịch sử và tình trạng in**

Use-case name	Kiểm tra lịch sử và tình trạng in
Actors	Student
Description	Là người dùng, tôi muốn truy cập và theo dõi thông tin về lịch sử cũng như tình trạng in ấn tài liệu mà tôi đã đặt in.
Trigger	Người dùng nhấn vào mục Lịch sử in và trạng thái in trong mục Profile .
Preconditions	<ul style="list-style-type: none">• Thiết bị của người dùng phải kết nối với Internet.• Người dùng phải đăng nhập vào hệ thống.
Postconditions	Hệ thống sẽ hiển thị danh sách lịch sử in ấn trong 3 tháng gần nhất
Normal Flow	1. Hệ thống hiển thị giao diện chứa danh sách lịch sử in ấn và tình trạng in của các tài liệu đã đặt in (Chưa in, Thành công, Thất bại).
Alternative Flow	Không có.
Exception Flow	<ul style="list-style-type: none">• Tại bước 1<ol style="list-style-type: none">1.1 Hệ thống lấy dữ liệu thất bại.1.2 Hệ thống thông báo lỗi ra màn hình.1.3 Hệ thống quay lại giao diện menu.1.4 Use-case dừng lại.
Constraints	Không có.

**Đặc tả use-case Quản lý tài liệu cá nhân**

Use-case name	Quản lý tài liệu cá nhân
Actors	Student
Description	Là người dùng, tôi muốn tải, xem, chỉnh sửa và xóa những tài liệu trong kho tài liệu cá nhân của mình.
Trigger	Người dùng nhấn vào mục Quản lí kho cá nhân trong mục Profile .
Preconditions	<ul style="list-style-type: none">• Thiết bị của người dùng phải kết nối với Internet.• Người dùng phải đăng nhập vào hệ thống.
Postconditions	Dữ liệu thay đổi do Student thực hiện các thao tác quản lý (nếu có) sẽ được cập nhật trong database
Normal Flow	<ol style="list-style-type: none">1. Hệ thống hiển thị giao diện quản lý kho lưu trữ cá nhân2. Hệ thống lấy danh sách các tài liệu cá nhân và hiển thị ra giao diện3. Hệ thống thực hiện các thao tác quản lý (nếu có) tùy thuộc vào người dùng.
Alternative Flow	Không có.
Exception Flow	<ul style="list-style-type: none">• Tại bước 2:<ol style="list-style-type: none">2.1 Hệ thống cập nhật dữ liệu thất bại.2.2 Hệ thống thông báo lỗi ra màn hình.2.3 Hệ thống quay lại giao diện menu.
Constraints	Không có.

5.5 Use-case diagram và use-case scenario cho Quản lý tài liệu

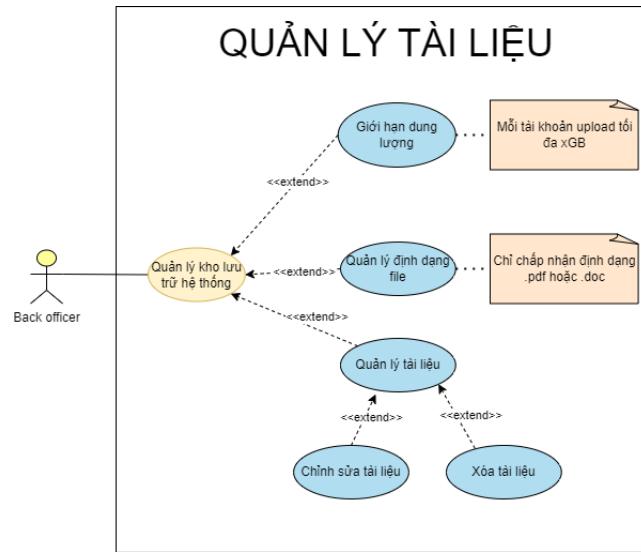


Figure 5: Use-case diagram cho Xem

Đặc tả use-case Quản lý tài liệu

Use-case name	Quản lý tài liệu
Actors	Back Officer
Description	Là một Back Officer, tôi muốn thực hiện các thao tác quản lý (chỉnh sửa, xóa) các tài liệu đang được lưu trữ trong hệ thống.
Trigger	Ấn vào mục Quản lý tài liệu tại trang chủ web(chỉ dành cho tài khoản admin).
Preconditions	<ul style="list-style-type: none"> Thiết bị Back Officer phải kết nối với Internet. Back Officer phải đăng nhập vào hệ thống.
Postconditions	Dữ liệu thay đổi do Back Officer thực hiện các thao tác quản lý(nếu có) sẽ được cập nhật trong database.
Normal Flow	<ol style="list-style-type: none"> Hệ thống mở giao diện Quản lý tài liệu. Hệ thống lấy danh sách các tài liệu và hiển thị ra giao diện. Hệ thống thực hiện các thao tác quản lý(nếu có) tùy thuộc vào Back Officer.
Alternative Flow	Không có.
Exception Flow	<ul style="list-style-type: none"> Tại bước 3 <ol style="list-style-type: none"> Hệ thống lấy dữ liệu thất bại. Hệ thống thông báo lỗi ra màn hình. Hệ thống quay lại giao diện menu.
Constraints	Không có.

**Đặc tả use-case Chính sửa tài liệu**

Use-case name	Chỉnh sửa tài liệu
Actors	Back Officer
Description	Là một Back Officer, tôi muốn chỉnh sửa các tài liệu đang được lưu trữ trong hệ thống.
Trigger	Trong lúc màn hình ở cửa sổ Quản lý tài liệu , bấm vào nút Edit tại phía bên phải của hàng của tài liệu muốn chỉnh sửa.
Preconditions	<ul style="list-style-type: none">• Thiết bị Back Officer phải kết nối với Internet.• Back Officer phải đăng nhập vào hệ thống.• Thiết bị của Back Officer đang ở cửa sổ Quản lý tài liệu.• Tài liệu có trong danh sách hiển thị.
Postconditions	Thông tin của tài liệu sẽ được cập nhật lại trong database và render ra giao diện.
Normal Flow	<ol style="list-style-type: none">1. Hệ thống lấy dữ liệu về tài liệu trong database.2. Hệ thống hiển thị 1 dialog chứa mẫu nhập thông tin chứa các trường muốn chỉnh sửa.3. Back Officer chỉnh sửa tài liệu.4. Back Officer nhấn nút Save.5. Hệ thống thông báo chỉnh sửa thành công và cập nhật lại tài liệu trong database và giao diện.
Alternative Flow	Không có.
Exception Flow	<ul style="list-style-type: none">• Tại bước 1<ul style="list-style-type: none">1.1 Hệ thống lấy dữ liệu thất bại.1.2 Người dùng bấm nút Cancel, hệ thống thoát khỏi giao diện Edit và quay lại màn hình chính.
Constraints	Không có.

Đặc tả use-case Xóa tài liệu

Use-case name	Xóa tài liệu
Actors	Back Officer
Description	Là một Back Officer, tôi muốn xóa các tài liệu đang được lưu trữ trong hệ thống.
Trigger	Trong lúc màn hình ở cửa sổ Quản lý tài liệu , bấm vào nút Delete tại phía bên phải của hàng của tài liệu muốn chỉnh sửa.
Preconditions	<ul style="list-style-type: none"> • Thiết bị Back Officer phải kết nối với Internet. • Back Officer phải đăng nhập vào hệ thống. • Thiết bị của Back Officer đang ở cửa sổ Quản lý tài liệu. • Tài liệu có trong danh sách hiển thị.
Postconditions	Thông tin của tài liệu sẽ được xóa khỏi database và giao diện.
Normal Flow	<ol style="list-style-type: none"> 1. Hệ thống hiển thị 1 dialog để xác nhận xóa tài liệu. 2. Bach Officer xác nhận xóa. 3. Hệ thống thực hiện xóa và thông báo xóa thành công.
Alternative Flow	Không có.
Exception Flow	<ul style="list-style-type: none"> • Tại bước 2 <ol style="list-style-type: none"> 2.1 Hệ thống lấy dữ liệu thất bại. 2.2 Người dùng bấm nút Cancel, hệ thống thoát khỏi giao diện Delete và quay lại màn hình chính.
Constraints	Không có.

5.6 Use-case diagram và use-case scenario cho Quản lý Sinh viên

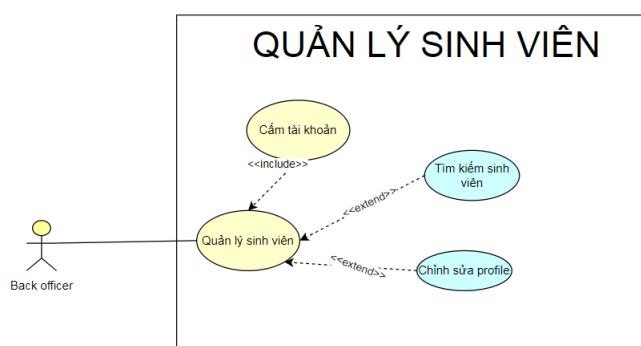


Figure 6: Use-case diagram cho Quản lý Sinh Viên

**Đặc tả use-case Quản lí Sinh Viên**

Use-case name	Quản lí Sinh Viên
Actors	Back Officer
Description	Là một Back Officer, tôi muốn thực hiện các thao tác quản lí (xem, chỉnh sửa, xóa) thông tin của tất cả sinh viên.
Trigger	Ấn vào mục Quản lí Sinh Viên tại trang chủ web(chỉ dành cho tài khoản admin).
Preconditions	<ul style="list-style-type: none">• Thiết bị Back Officer phải kết nối với Internet.• Back Officer phải đăng nhập vào hệ thống.
Postconditions	Dữ liệu thay đổi do Back Officer thực hiện các thao tác quản lí(nếu có) sẽ được cập nhật trong database.
Normal Flow	<ol style="list-style-type: none">1. Hệ thống mở giao diện Quản lí Sinh viên.2. Hệ thống lấy danh sách tất cả sinh viên sử dụng SPSO.3. Hệ thống thực hiện các thao tác quản lí(nếu có) tùy thuộc vào Back Officer.
Alternative Flow	Không có.
Exception Flow	<ul style="list-style-type: none">• Tại bước 2<ol style="list-style-type: none">2.1 Hệ thống lấy dữ liệu thất bại.2.2 Hệ thống thông báo lỗi ra màn hình.2.3 Hệ thống quay lại giao diện menu.
Constraints	Không có.



Detailed description of the use-case: Lock account

Use-case name	Lock account
Actors	Back Officer
Description	Là một back officer, tôi muốn cấm các tài khoản upload các nội dung nhạy cảm.
Trigger	Ấn vào mục Quản lý Sinh Viên tại trang chủ web(chỉ dành cho tài khoản admin). Ấn vào mục tài khoản vi phạm.
Preconditions	<ul style="list-style-type: none">• Thiết bị Back Officer phải kết nối với Internet.• Back Officer phải đăng nhập vào hệ thống.
Postconditions	Các tài khoản bị cấm sẽ được đưa vào danh sách tài khoản bị cấm.
Normal Flow	<ol style="list-style-type: none">1. Hệ thống hiển thị một danh sách các tài khoản đã được bén phẩn hồi đánh giá là " Cấm tài khoản ".2. Back officer sẽ chọn hình thức xử lý và xác nhận cấm.3. Hệ thống xử lý việc truy cập của tài khoản theo yêu cầu, và thông báo cấm thành công.
Alternative Flow	Không có.
Exception Flow	<ul style="list-style-type: none">• Tại bước 1<ol style="list-style-type: none">1.1 Hệ thống lấy dữ liệu thất bại.1.2 Hệ thống thông báo lỗi ra màn hình.
Constraints	Không có.

Dặc tả use-case Tìm kiếm sinh viên

Use-case name	Tìm kiếm sinh viên
Actors	Back Officer
Description	Là một back officer, tôi muốn tìm kiếm sinh viên để có thể xem, chỉnh sửa thông tin.
Trigger	Ấn vào mục Quản lý Sinh Viên tại trang chủ web(chỉ dành cho tài khoản admin). Search tên sinh viên hoặc mssv của sinh viên.
Preconditions	<ul style="list-style-type: none"> Thiết bị Back Officer phải kết nối với Internet. Back Officer phải đăng nhập vào hệ thống.
Postconditions	Hiển thị sinh viên được tìm kiếm.
Normal Flow	<ol style="list-style-type: none"> Back officer search thông tin sinh viên trên thanh tìm kiếm. Hệ thống sẽ tìm kiếm sinh viên theo thông tin và hiện ra sinh viên.
Alternative Flow	Không có.
Exception Flow	<ul style="list-style-type: none"> Tại bước 1 <ol style="list-style-type: none"> Nếu không tìm thấy sinh viên, hệ thống sẽ hiển thị không tìm thấy kết quả.
Constraints	Không có.

5.7 Use-case diagram và use-case scenario cho Quản lý máy in

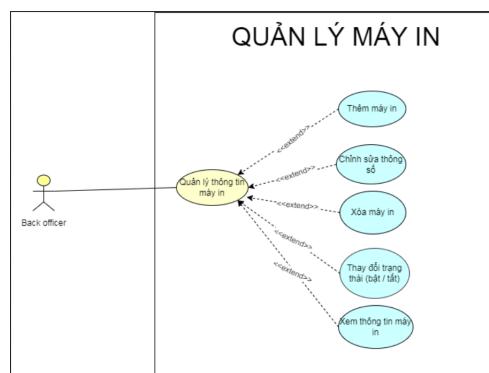


Figure 7: Use-case diagram cho Quản lý máy in



Detailed description of the use-case: Printer Management

Use-case name	Quản lý máy in
Actors	Back Officer
Description	Là một Back Officer, tôi muốn quản lý các máy in để phục vụ cho công việc in ấn.
Trigger	Người dùng ấn vào mục Printers Management ở thanh Menu (chỉ dành cho tài khoản admin).
Preconditions	<ul style="list-style-type: none">• Thiết bị của người dùng phải kết nối với Internet.• Người dùng phải đăng nhập vào hệ thống.
Postconditions	Dữ liệu thay đổi do Officer thực hiện các thao tác quản lý (nếu có) sẽ được cập nhật trong database.
Normal Flow	<ol style="list-style-type: none">1. Hệ thống hiển thị giao diện quản lý máy in.2. Hệ thống lấy dữ liệu danh sách các máy in và hiển thị ra giao diện3. Hệ thống thực hiện các thao tác quản lý (nếu có) tùy thuộc vào người dùng.
Alternative Flow	Không có.
Exception Flow	<ul style="list-style-type: none">• Tại bước 2:<ol style="list-style-type: none">2.1 Hệ thống cập nhật dữ liệu thất bại.2.2 Hệ thống thông báo lỗi ra màn hình.2.3 Hệ thống quay lại giao diện menu.
Constraints	Không có.



Dặc tả use-case Thêm máy in

Use-case name	Thêm máy in
Actors	Back Officer
Description	Là một back officer, tôi muốn tìm kiếm sinh viên để có thể xem, chỉnh sửa thông tin.
Trigger	Người dùng ấn vào mục Add Printers trong phần Printers Management ở thanh Menu (chỉ dành cho tài khoản admin)
Preconditions	<ul style="list-style-type: none">• Thiết bị Back Officer phải kết nối với Internet.• Back Officer phải đăng nhập vào hệ thống.
Postconditions	Hệ thống sẽ cập nhật thêm các máy in mới vào cơ sở dữ liệu và hiển thị thêm máy in đó vào danh sách các máy in.
Normal Flow	<ol style="list-style-type: none">1. Hệ thống hiển thị một dialog chứa các trường thông tin cần nhập cho máy in mới.2. Người dùng nhập các thông tin và chọn Add.3. Hệ thống thông báo thêm máy in mới thành công.
Alternative Flow	Người dùng nhập sai định dạng so với định dạng thông tin của chú thích, nút Add sẽ không khả dụng.
Exception Flow	<ul style="list-style-type: none">• Tại bước 2<ol style="list-style-type: none">2.1 Người dùng chọn Cancel, hệ thống quay lại giao diện của Printers Management.
Constraints	Không có.

Dặc tả use-case Chính sửa thông số máy in

Use-case name	Chỉnh sửa thông số máy in
Actors	Back Officer
Description	Là một Back Officer, tôi muốn chỉnh sửa các thông số của các máy in ở trong danh sách.
Trigger	Người dùng ấn vào nút Adjust bên cạnh các máy in trong danh sách.
Preconditions	<ul style="list-style-type: none"> Thiết bị Back Officer phải kết nối với Internet. Back Officer phải đăng nhập vào hệ thống. Phải có ít nhất 1 máy in ở trong danh sách.
Postconditions	Hệ thống sẽ cập nhật dữ liệu về thông số của các máy in trong cơ sở dữ liệu và hiển thị ra giao diện Thông tin máy in
Normal Flow	<ol style="list-style-type: none"> Hệ thống hiển thị 1 dialog chứa các trường thông tin để người dùng thay đổi. Người dùng chọn Done sau khi hoàn tất thay đổi. Hệ thống thông báo cập nhật thông số thành công.
Alternative Flow	Người dùng nhập sai định dạng so với định dạng thông tin của chú thích, nút Done sẽ không khả dụng.
Exception Flow	<ul style="list-style-type: none"> Tại bước 2 <ol style="list-style-type: none"> Người dùng chọn Cancel, hệ thống quay lại giao diện của Printers Management.
Constraints	Không có.

5.8 Use-case diagram và use-case scenario cho Gửi thông tin

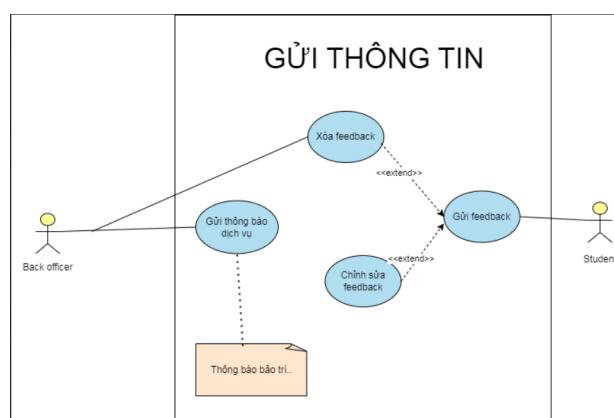


Figure 8: Use-case diagram cho Gửi phản hồi



Dặc tả use-case Gửi phản hồi

Use-case name	Gửi phản hồi
Actors	Student
Description	Là người sử dụng dịch vụ, tôi muốn gửi phản hồi, đánh giá về chất lượng của dịch vụ cho người quản trị hệ thống.
Trigger	Ấn vào mục Feedback .
Preconditions	Người dùng đã đăng nhập vào hệ thống và thiết bị của người dùng có kết nối Internet
Postconditions	Thông báo cho người dùng biết ý kiến, phản hồi của người dùng đã được gửi đến người quản trị hệ thống.
Normal Flow	<ol style="list-style-type: none">1. Hệ thống mở cửa sổ Phản hồi2. Người dùng điền tiêu đề.3. Người dùng điền ý kiến phản hồi.4. Người dùng chọn nút Send.
Alternative Flow	Không có
Exception Flow	Ở bước 4, người dùng chọn nút Cancel , quay trở lại giao diện trước đó.
Constraints	Không có

II SYSTEM MODELLING

1 Activity Diagram

Xem các bản vẽ Activity Diagram FULL của nhóm tại [đây](#)

1.1 Activity Diagram cho Quản lý Máy In

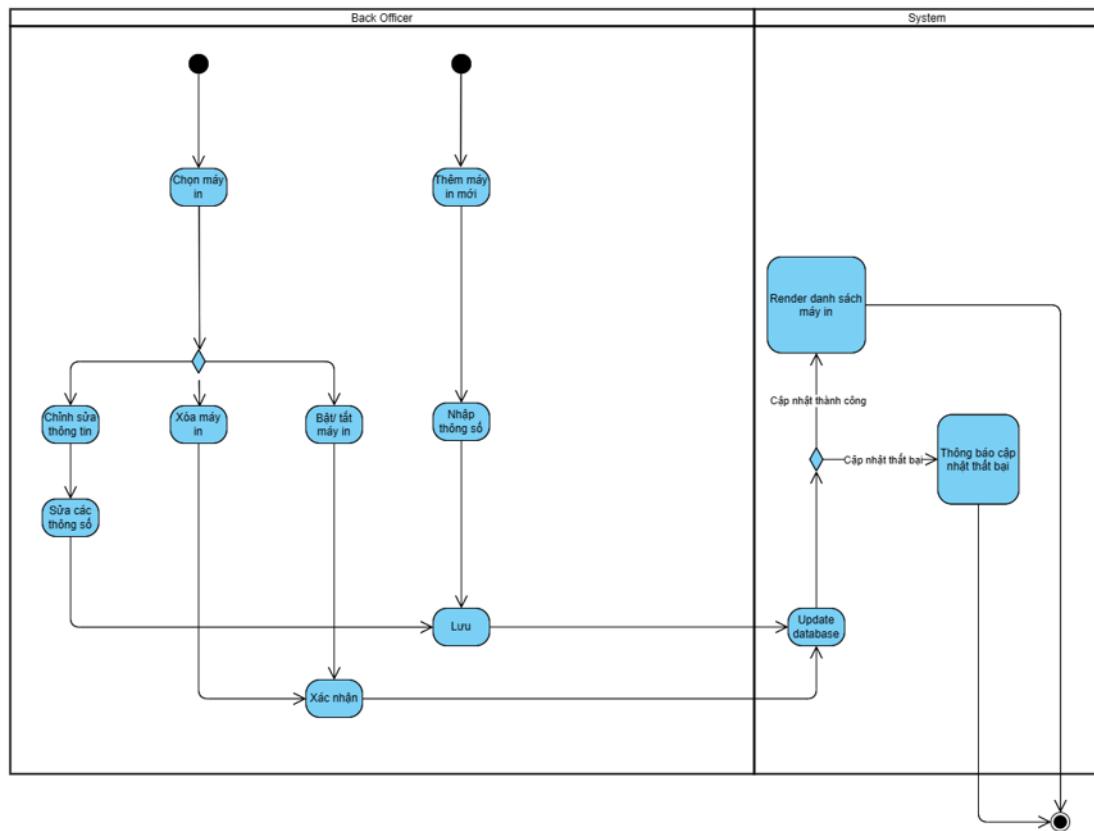


Figure 9: Activity Diagram cho Quản lý Máy In

Mô tả Activity Diagram cho Quản lý Máy In

Back Officer có thể chỉnh sửa thông tin máy in, xóa máy in, bật tắt máy in, thêm máy in mới.

- Nếu Back Officer muốn chỉnh sửa thông tin máy in thì cần chọn máy in đó, sau đó sửa các thông số và nhấn nút Lưu, System sẽ cập nhật lại thông tin của máy in đó ở Database.
- Nếu Back Officer muốn xóa máy in thì cần chọn máy in đó và xác nhận lại là có muốn xóa hay không. Nếu có System sẽ truy cập vào Database và sẽ xóa dữ liệu của máy in đó.
- Nếu Back Officer muốn thay đổi trạng thái của máy in thì cần chọn máy in và xác nhận lại, System sẽ cập nhật trạng thái của máy in.
- Nếu Back Officer muốn thêm máy in mới thì cần nhập các thông số liên quan đến máy in và nhấn Lưu. System sẽ lưu thông tin máy in vào Database.

1.2 Activity Diagram cho Quản lý Sinh Viên

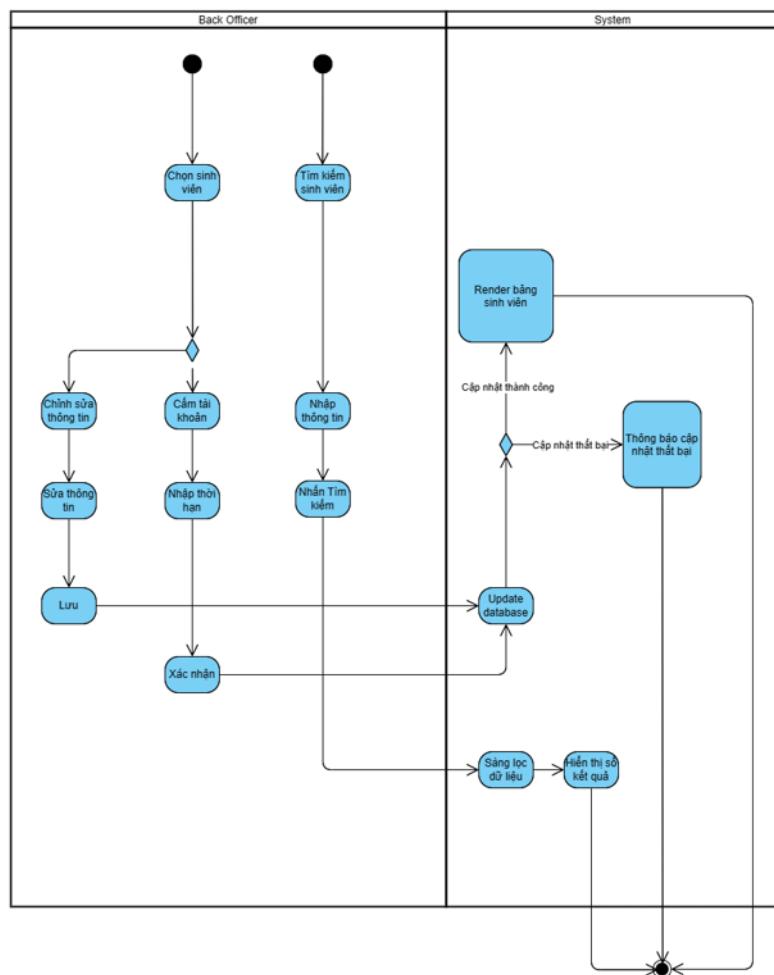


Figure 10: Activity Diagram cho Quản lý Sinh Viên

Mô tả Activity Diagram cho Quản lý Sinh Viên

Back Officer có thể chỉnh sửa thông tin sinh viên, cấm tài khoản, tìm kiếm sinh viên.

- Nếu Back Officer muốn chỉnh sửa thông tin sinh viên thì cần phải chọn sinh viên. Lưu ý chỉ có thể chỉnh sửa những thông tin liên quan đến hệ thống không sửa thông tin cá nhân của sinh viên.
- Nếu Back Officer muốn cấm tài khoản của sinh viên thì cần nhập thời hạn cấm và xác nhận lại với System. System sẽ lưu trữ lên Database và tiến hành cấm tài khoản.
- Nếu Back Officer muốn tìm kiếm sinh viên thì cần nhập thông tin liên quan đến sinh viên (như là tên, mssv, email, ..). System sẽ tìm kiếm trên Database và sẽ trả về một danh sách sinh viên có liên quan đến thông tin mà Back Officer đã nhập.

1.3 Activity Diagram cho In

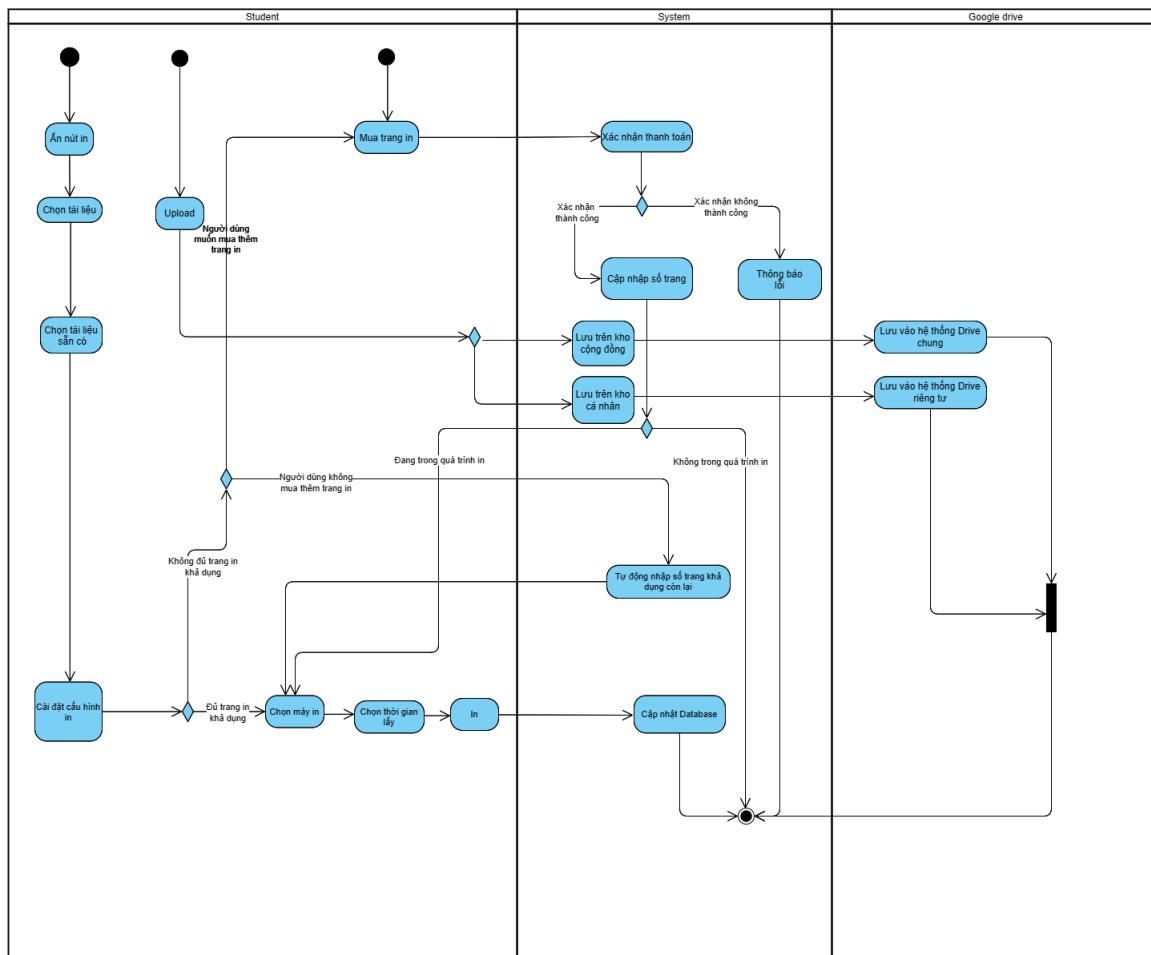


Figure 11: Activity Diagram cho In

Mô tả Activity Diagram cho In

Student có thể in tài liệu hoặc chọn mua thêm trang.

- Nếu Student chọn In tài liệu thì cần phải chọn tài liệu. Để chọn tài liệu, Student có thể chọn tài liệu khi upload hoặc tìm kiếm tài liệu đã có sẵn (public hoặc private). Lưu ý khi upload, Student có thể chọn public hoặc private sau đó System sẽ lưu lên Google Drive. Nếu lưu thành công thì System sẽ lưu vào Database. Sau khi đã có tài liệu thì sẽ cài đặt cấu hình in.
 - Nếu số trang cần in vượt quá số trang còn lại của Student thì Student có thể chọn mua trang in. Khi mua trang in, Student sẽ ghi nợ trên BkPay. System sẽ lưu lịch sử mua lên Database.
 - Nếu số trang in không vượt quá, thì Student tiếp tục chọn máy in, chọn thời gian lấy và nhấn nút "In" để thực hiện việc In.
- Student cũng có thể mua thêm trang ở trang chủ.

2 Sequence Diagram

Xem các bản vẽ Sequence Diagram FULL của nhóm tại [đây](#)

2.1 Sequence Diagram cho In

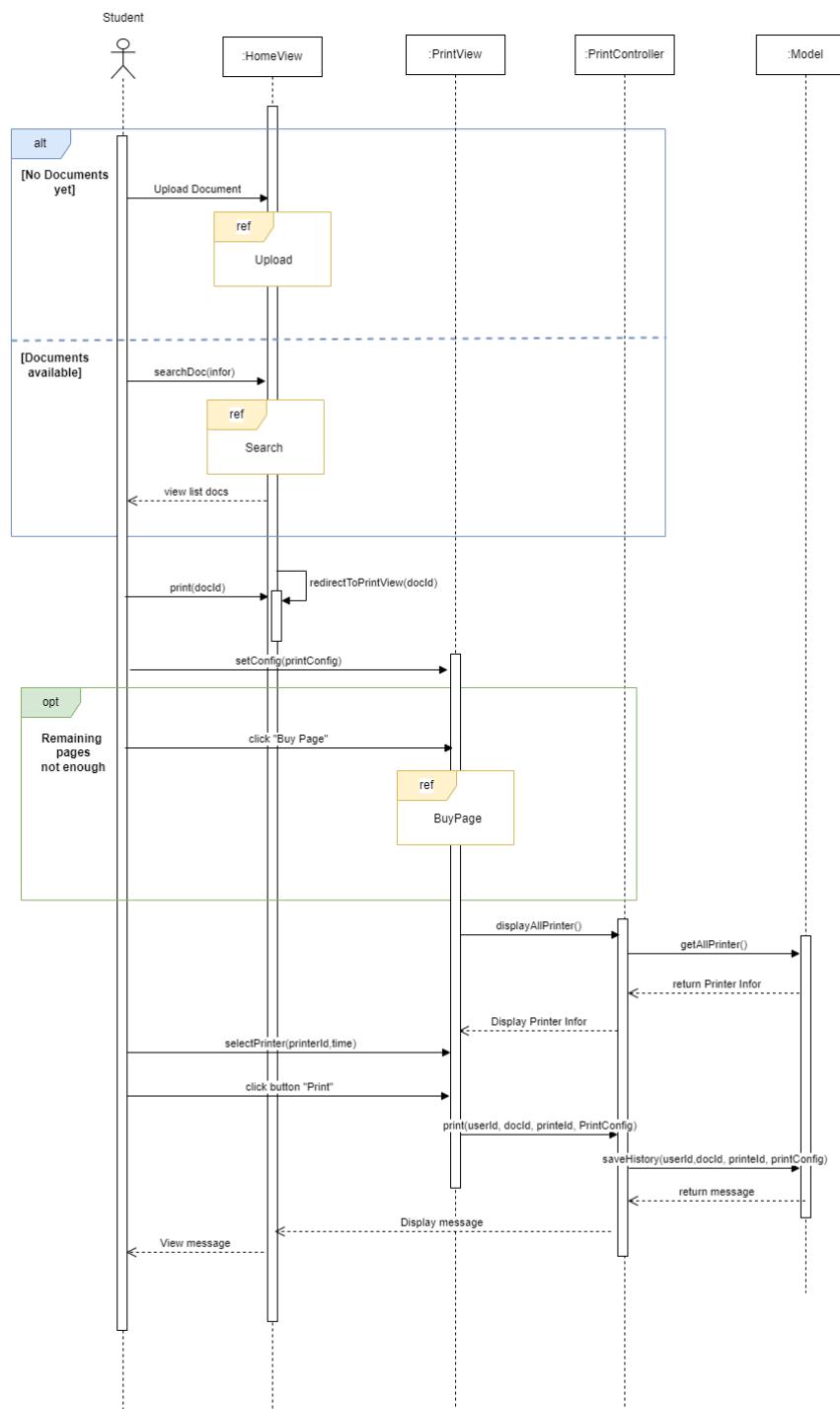


Figure 12: Sequence Diagram cho In tài liệu

Mô tả các bước để thực hiện việc in tài liệu từ Sequence Diagram

Khi Student muốn in tài liệu :

- Nếu Student chưa chọn tài liệu để in thì Student cần tải tài liệu (Upload Document) lên hoặc Tìm kiếm tài liệu (Search Document).
- Nếu Student chọn được tài liệu từ kho cá nhân hoặc kho hệ thống tại **HomeView** (trang chủ) thì Student sẽ tiếp tục cài đặt các thông số in ở **PrintView** như là số trang in, in một mặt hay hai mặt, kích thước trang in (A4, A3,...). Nếu như số trang in vượt quá số trang còn lại của Student có thể lựa chọn mua thêm trang in (BuyPage) hoặc giảm số trang cần in.
- Sau đó **PrintView** sẽ hiển thị các máy in hiện có để Student có thể chọn. Nhưng muốn có dữ liệu tất cả máy in thì cần gọi hàm `DisplayAllPrinter()` ở **PrintController**. Khi đó **PrintController** sẽ tiếp tục gọi hàm `getAllPrinter()` tại Model **Printer**. Model **Printer** sẽ truy cập đến Database để có thể lấy được tất cả dữ liệu liên quan đến máy in (printer). Sau khi Model **Printer** lấy được dữ liệu từ **Database** thì sẽ trả về cho **PrintController** và **PrintController** sẽ trả dữ liệu cho **PrintView** để có thể hiển thị máy in cho Student.
- Nếu tất cả các bước trên thực hiện xong, ngay khi Student nhấn nút in (printButton) tại **PrintView** thì sẽ gọi hàm `print(userId, docId, printerId, printConfig)` ở **PrintController** và **PrintController** sẽ gọi hàm `saveHistory(userId, docId, printerId, printConfig)` của Model **PrintHistory** để có thể lưu lịch sử in của Student vào **Database**. Sau khi lưu, Model **PrintHistory** sẽ trả về message cho **PrintController** và **PrintController** sẽ trả về cho **HomeView** (trang chủ) để có thể hiển thị cho Student là thao tác thành công hay thất bại.

2.2 Sequence Diagram cho Upload Document

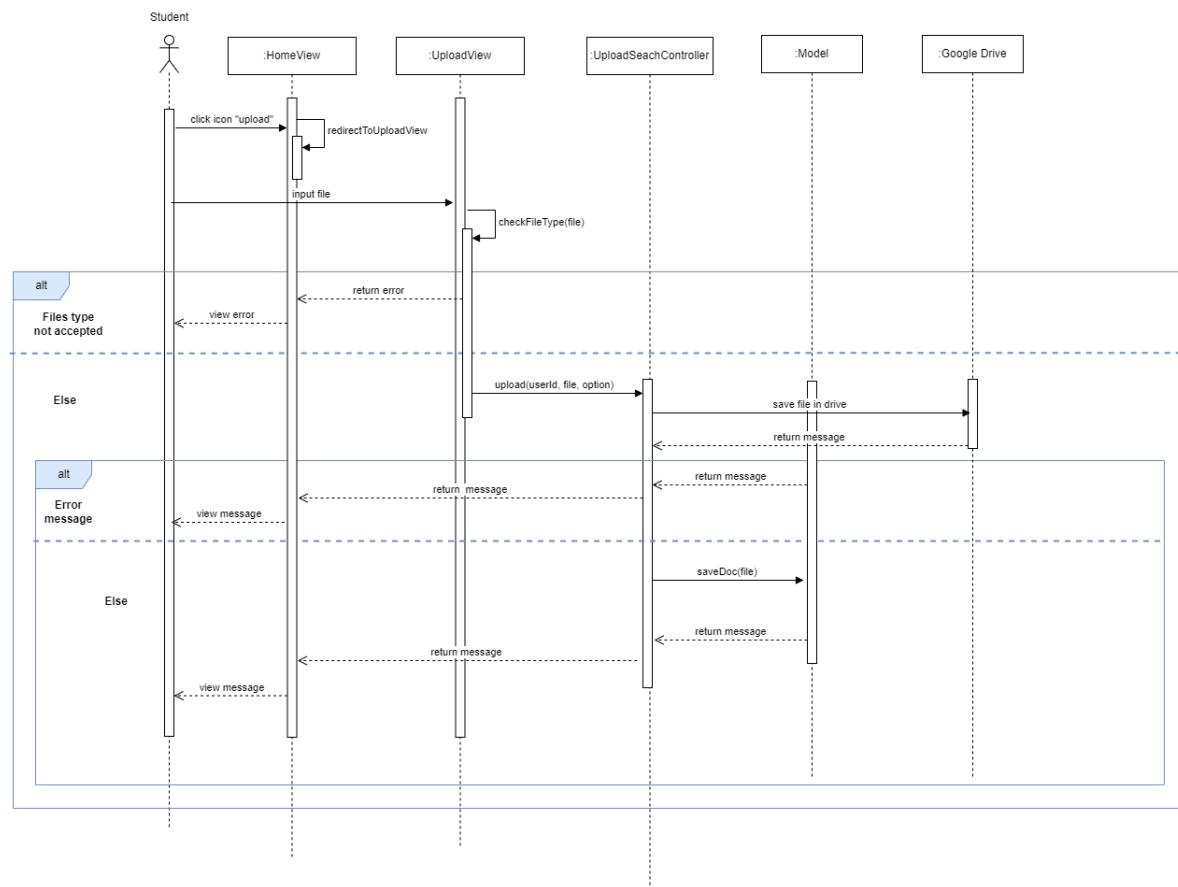


Figure 13: Sequence Diagram cho Upload

Mô tả các bước để thực hiện việc upload tài liệu từ Sequence Diagram

- **Student** nhấn nút "Upload" tại **HomeView** (trang chủ), chọn tài liệu muốn upload và nhấn "Upload", **UploadView** gọi hàm checkFileType(file) truyền vào file là tài liệu được upload để kiểm tra định dạng có hợp lệ hay không của tài liệu.
 - Nếu tài liệu không hợp lệ, **UploadView** thông báo lỗi ra màn hình và kết thúc việc upload tài liệu.
 - Nếu tài liệu hợp lệ, thì tiếp tục các bước tiếp theo.
- **UploadSearchController** gọi hàm upload(userId, file, option) trong đó userId là ID của người upload, file là tài liệu được upload để cập nhật tài liệu vào kho lưu trữ trên Google Drive, option là lựa chọn có chia sẻ công khai không.
- Sau đó, **UploadSearchController** gọi hàm saveDoc(userId, file) của Model **Document** để lưu dữ liệu vào Database.
 - Nếu thất bại trong việc lưu dữ liệu vào Database, **HomeView** thông báo lỗi ra màn hình và kết thúc việc upload tài liệu.
 - Nếu thành công, Model **Document** trả về message và file_id là ID của tài liệu vừa được upload về cho **UploadController** và **UploadController** sẽ trả về cho **HomeView** để hiển thị thông báo thành công .

2.3 Sequence Diagram cho Tìm kiếm tài liệu (Search Document)

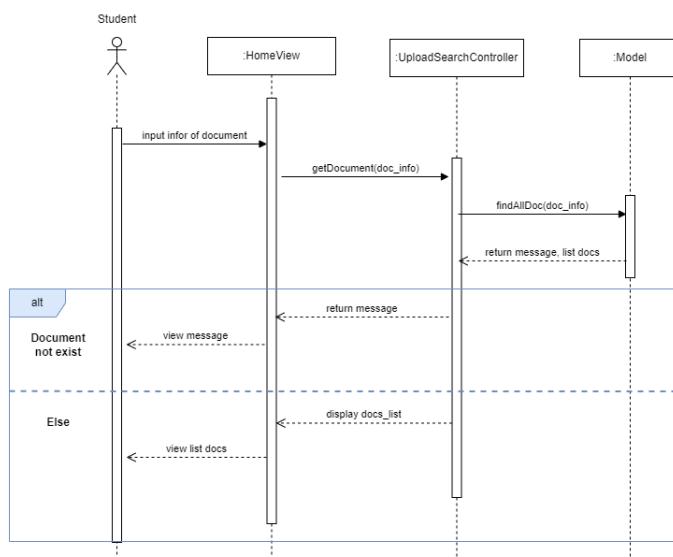


Figure 14: Sequence Diagram cho Tìm kiếm tài liệu

Mô tả các bước để thực hiện việc tìm kiếm tài liệu

- **Student** đang ở **HomeView**, ở đó sinh viên nhập thông tin về tài liệu và nhấn nút "Search".
- **SearchView** gọi hàm getDocument(doc_info) tới **UploadSearchController**, trong đó doc_info là thông tin tài liệu cần tìm.
- **UploadSearchController** gửi yêu cầu lấy tất cả tài liệu có liên quan đến thông tin tìm kiếm đến Model **Document** : Gọi hàm findAllDoc(doc_info), trong đó doc_info là thông tin tài liệu cần tìm.
- Model **Document** kết nối đến **Database** và thực hiện việc tìm kiếm tài liệu. **Database** sau khi hoàn thành việc tìm kiếm sẽ trả về dữ liệu Model **Document**.

- Nếu không tìm thấy tài liệu, Model **Document** sẽ gửi thông báo về cho **HomeView** thông qua **UploadSearchController**, sau đó **HomeView** hiển thị lên màn hình thông báo tài liệu không tồn tại.
- Ngược lại nếu tìm thấy tài liệu, Model **Model** sẽ trả về các tài liệu có liên quan, **UploadSearchController** sẽ nhận các tài liệu ấy và gửi yêu cầu đến **HomeView**. **HomeView** sẽ làm nhiệm vụ hiển thị lên màn hình danh sách các tài liệu.

2.4 Sequence Diagram cho Mua trang in (Buy Page)

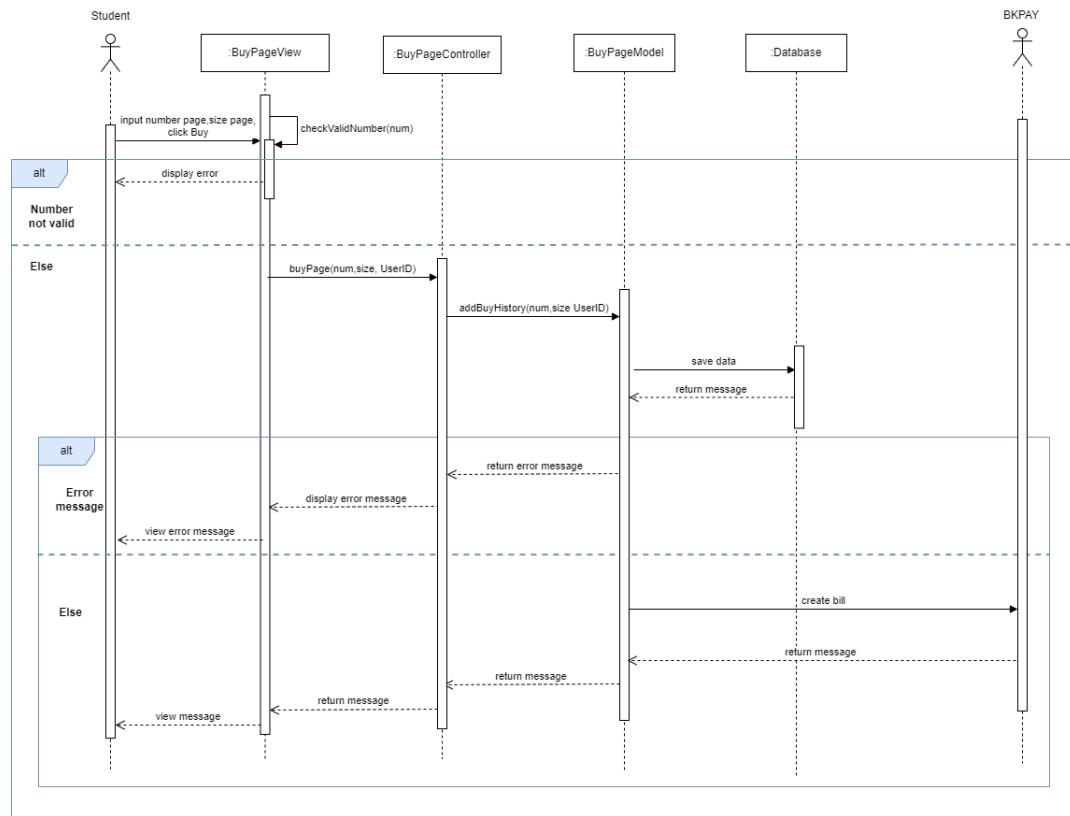


Figure 15: Sequence Diagram cho Mua trang in

Mô tả các bước để thực hiện việc mua trang in (Buy Page)

- Student** nhập số trang in cần mua, chọn loại trang cần mua và nhấn nút "Buy", hệ thống sẽ gọi hàm `checkValidNumber(num)` với num là số trang cần mua
 - Nếu số trang cần mua không hợp lệ, **BuyPageView** sẽ hiển thị lỗi ra màn hình và kết thúc việc mua trang in.
 - Nếu số trang cần mua là hợp lệ, thì tiếp tục.
- BuyPageView** gọi hàm `buyPage(num, size, userId)` của **BuyPageController** trong đó num là số trang cần mua, size là kích thước trang, userId là ID của người mua.
- BuyPageController** gọi hàm `addBuyHistory(num, size, userId)` của Model **Bill** trong đó num là số trang cần mua, size là kích thước trang, userId là ID của người mua.
- Model **Bill** kết nối tới **Database** và thực hiện việc lưu lịch sử mua trang in của người dùng.
 - Nếu lưu không thành công, **BuyPageView** thông báo lỗi ra màn hình và kết thúc việc mua trang in.
 - Nếu lưu thành công, **BuyPageController** kết nối tới **BKPAY** và thực hiện việc tạo hóa đơn thanh toán và trả về thông báo cho **Student** tại **BuyPageView**.

3 Class Diagram

Xem bản vẽ Class Diagram FULL của nhóm tại [đây](#)

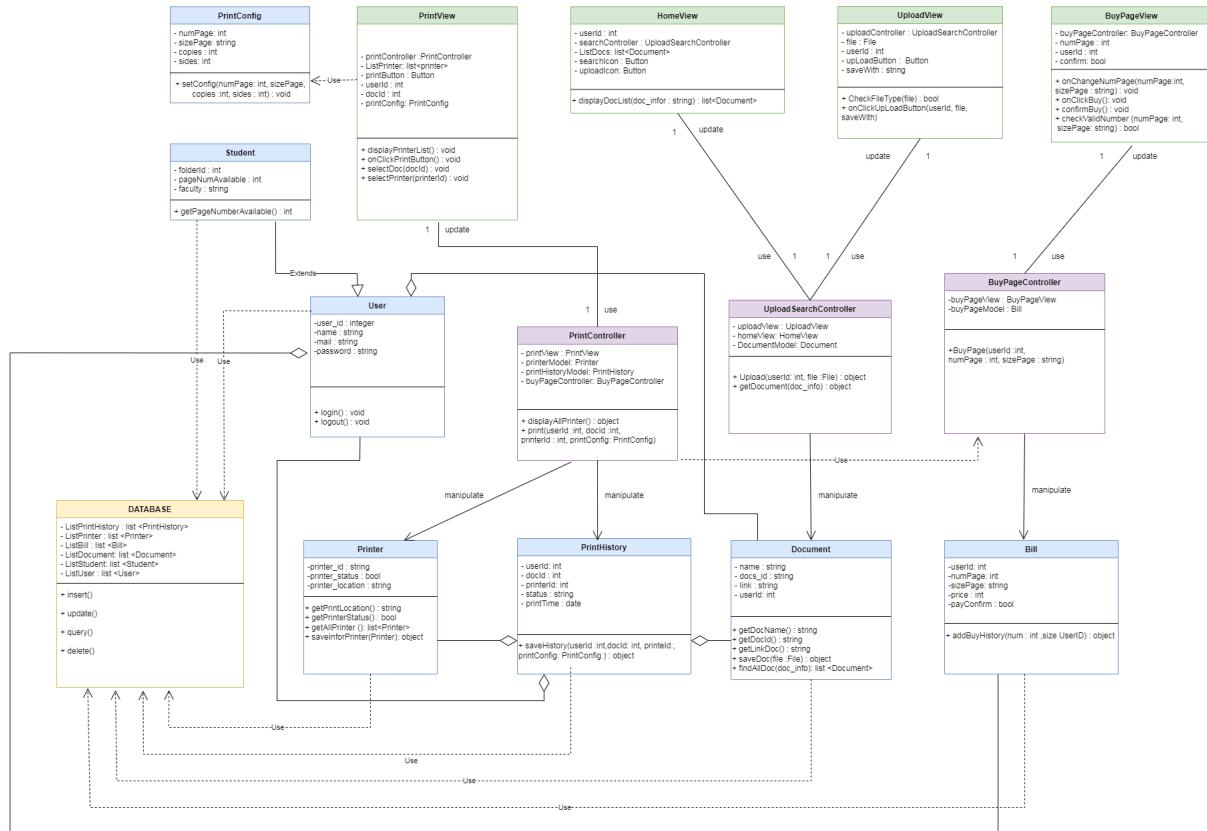


Figure 16: Class Diagram cho In tài liệu

Class Diagram của module **In tài liệu** được nhóm thiết kế theo mô hình kiến trúc **MVC** (Model-Controller-View). Mỗi lớp (layer) đảm nhiệm một nhóm chức năng trong hệ thống:

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



Class	Mô tả	Mối quan hệ với
UploadView	<ul style="list-style-type: none">• Chức năng chính : Hiển thị giao diện Upload tài liệu.• Các thuộc tính<ul style="list-style-type: none">- uploadController:UploadSearchController- file : Document- userId : int- upLoadButton : Button- saveWith : string• Các phương thức<ul style="list-style-type: none">+ CheckFileType(file): bool (kiểm tra định dạng của file có hợp lệ không)+ onClickUpLoadButton(userId, file) : void (xử lý sự kiện onClick của Student, truyền dữ liệu cho UploadController)	UploadController (Association): Gọi function Upload từ UpLoadController.
PrintView	<ul style="list-style-type: none">• Chức năng chính : Hiển thị chức năng In tài liệu.• Các thuộc tính<ul style="list-style-type: none">- printController : PrintController- ListPrinter : list<Printer>- userId : int- printButton : Button- docId : int- printConfig : PrintConfig• Các phương thức<ul style="list-style-type: none">+ displayPrinterList() : void (Hiển thị tất cả máy in đang hoạt động trong hệ thống)+ onClickUpPrintButton() : void (xử lý sự kiện onClick của Student, truyền dữ liệu cho PrintController)+ selectDoc(docId) : void(Hàm chọn tài liệu để in theo docId)+ selectPrinter(printId) : void(Hàm sử dụng biến truyền vào là printerId để chọn máy in thực hiện việc in tài liệu)	PrintController: (Association) Gọi function Print từ PrintController.



HomeView	<ul style="list-style-type: none">• Chức năng chính : Hiển thị chức năng Tìm kiếm.• Các thuộc tính<ul style="list-style-type: none">– searchController: UploadSearchController– ListDocs : list<Document>– searchButton : Button• Các phương thức<ul style="list-style-type: none">+ <code>displayDocList(doc_infor:string): list<Document></code>(Hiển thị tất cả tài liệu được tìm thấy dựa vào doc_infor)+ <code>onClickSearchButton():void</code>(gọi function <code>displayDocList(doc_infor:string)</code>)+ <code>reDirectToUploadView():void</code>(Chuyển hướng tới giao diện upload))+ <code>reDirectToPrintView():void</code>(Chuyển hướng tới giao diện in))+ <code>reDirectToBuyPageView():void</code>(Chuyển hướng tới giao diện mua trang))	<p>UploadSearchController: (Association) Gọi function <code>getDocument(doc_infor : string)</code> từ SearchController.</p>
BuyPageView	<ul style="list-style-type: none">• Chức năng chính : Hiện thị chức năng Mua trang in• Các thuộc tính<ul style="list-style-type: none">– buyPageController: BuyPageController– numPage : int– userId : int– confirm: bool• Các phương thức<ul style="list-style-type: none">+ <code>onChangeNumPage(numPage:int, sizePage : string) : void</code> (xử lý sự kiện onChange của Student khi nhập số trang cần mua)+ <code>onClickBuy(): void</code> : void (truyền dữ liệu cho BuyPageController)+ <code>confirmBuy() : void</code>(Xác nhận Student thực hiện việc mua trang in, Student sẽ phải thanh toán cho BKPAY)+ <code>checkValidNumber (numPage: int, sizePage: string) : bool</code> (kiểm tra số trang Student mua có hợp lệ không(không được <=0))	<p>BuyPageController: (Association) Gọi function từ BuyPageController.</p>



UploadSearchController	<ul style="list-style-type: none">• Chức năng chính : Xử lý, điều khiển các tài liệu được tải lên từ người dùng, cập nhật giao diện.• Các thuộc tính<ul style="list-style-type: none">– uploadView: UpLoadView– homeView: HomeView– DocumentModel: Document• Các phương thức<ul style="list-style-type: none">+ Upload(userId: int, file :File) : object (Hàm dùng để tải tài liệu lên hệ thống, trong đó file là tài liệu được chọn để tải lên, userId dùng để định danh người dùng đã tải lên tài liệu)	<p>UploadView:(Association) Cung cấp chức năng điều khiển luồng dữ liệu và cập nhật giao diện.</p> <p>HomeView:(Association) Cung cấp chức năng điều khiển luồng dữ liệu và cập nhật giao diện.</p> <p>Document:(Association) Gửi yêu cầu thực hiện việc tải lên hoặc tìm kiếm tài liệu .</p>
PrintController	<ul style="list-style-type: none">• Chức năng chính : Xử lý, điều khiển các luồng dữ liệu in ấn, cập nhật tài liệu cho chức năng in, cập nhật giao diện• Các thuộc tính<ul style="list-style-type: none">– printView: PrintView– printModel: PrintModel– uploadController: UploadController– searchController: SearchController– buyPageController: BuyPageController• Các phương thức<ul style="list-style-type: none">+ displayAllPrinter() : object (Hàm dùng để hiển thị tất cả các máy in đang hoạt động)+ Print(userId :int, docId :int, printerId : int, printConfig: PrintConfig) : object (Hàm gọi đến hàm xử lý in của printModel dựa vào thông tin là các biến được truyền vào hàm.)	<p>PrintView (Association): Cung cấp chức năng điều khiển luồng dữ liệu và cập nhật giao diện.</p> <p>PrintHistory (Association): Gửi yêu cầu thực hiện việc in tài liệu.</p> <p>Printer (Association): Gửi yêu cầu thực hiện việc in tài liệu.</p>



BuyPageController	<ul style="list-style-type: none">• Chức năng chính : Xử lý, điều khiển các luồng dữ liệu mua trang in, cập nhật giao diện.• Các thuộc tính<ul style="list-style-type: none">– buyPageView: BuyPageView– buyPageModel: BuyPageModel• Các phương thức<ul style="list-style-type: none">+ <code>BuyPage(userId :int, numPage : int, sizePage : string) :object</code> (Hàm gửi yêu cầu tạo hóa đơn và cập nhật lịch sử của việc mua trang in đến buyPageModel dựa vào thông tin là các biến được truyền vào hàm)	<p>BuyPageView (Association): Cung cấp chức năng điều khiển luồng dữ liệu và cập nhật giao diện. Bill (Association): Gửi yêu cầu thực hiện việc tìm kiếm tài liệu.</p>
PrintHistory	<ul style="list-style-type: none">• Chức năng chính: Lưu lại lịch sử in của Student• Các thuộc tính:<ul style="list-style-type: none">– userId: int– docId : int– printerId: int– status : string– printTime : date	<p>PrintController: (Association) Cung cấp CTDL để hỗ trợ việc lưu lịch sử in của Student. Database: (Dependency) Lưu lịch sử in của Student. Printer: (Aggregation) cung cấp printer_id để tạo PrintHistory.</p>
Printer	<ul style="list-style-type: none">• Chức năng chính: Lưu lại thông tin các máy in• Các thuộc tính<ul style="list-style-type: none">– printer_id : string– printer_status : bool– printer_location : string• Các phương thức:<ul style="list-style-type: none">+ <code>getPrintLocation(): string</code> (cung cấp vị trí của máy in)+ <code>getPrinterStatus(): bool</code> (cung cấp trạng thái (hoạt động/ không hoạt động) của máy in)	<p>PrintController (Association): trả dữ liệu đã xử lý cho PrintController. Database (Dependency): Lưu tất cả máy in. PrintHistory (Aggregation): Cung cấp vị trí máy in mà Student đã sử dụng, mỗi Printer có thể thuộc 0..n PrintHistory</p>



Student	<ul style="list-style-type: none">• Chức năng chính: Lưu lại thông tin sinh viên dùng máy in• Các thuộc tính:<ul style="list-style-type: none">- folderId : int- pageNumAvailable : int- faculty : string• Các phương thức:<ul style="list-style-type: none">+ <code>getPageNumberAvailable(): int</code> (số trang còn lại mà sinh viên có thể in)	<p>User: Kế thừa từ User Database(Dependency): Lưu lại tất cả user.</p>
User	<ul style="list-style-type: none">• Chức năng chính: Lưu lại thông tin những người dùng SSPS• Các thuộc tính:<ul style="list-style-type: none">- user_id : integer- name : string- mail : string- password : string• Các phương thức:<ul style="list-style-type: none">+ <code>login(): void</code> (Đăng nhập)+ <code>logout(): void</code> (Đăng xuất)	<p>PrintHistory (Aggregation): Cung cấp ID người dùng, mỗi User thuộc 0..n PrintHistory Student: (Generalization) Là class cha của Student. Document: (Aggregation) Cung cấp ID người dùng, mỗi User có 0..n Document Bill: (Aggregation) Cung cấp ID người dùng, mỗi User có 0..n Bill</p>
Document	<ul style="list-style-type: none">• Chức năng chính: Lưu thông tin tài liệu• Các thuộc tính:<ul style="list-style-type: none">- name : string- docs_id : string- link : string- userId: int• Các phương thức:<ul style="list-style-type: none">+ <code>getDocName() : string</code> (Lấy tên tài liệu)+ <code>getDocId() : string</code> (Lấy ID tài liệu)+ <code>getLinkDoc() : string</code> (Lấy link liên kết Google Drive của tài liệu)	<p>PrintHistory: (Aggregation) Cung cấp ID tài liệu được in, mỗi Document thuộc 0..n PrintHistory Database: (Dependency) Lưu lại tất cả Document. UploadSearch-Controller: (Association) cung cấp thông tin để xử lý. User: (Aggregation) lấy ID của người dùng sở hữu tài liệu, mỗi Document thuộc 1 User</p>



Bill	<ul style="list-style-type: none">• Chức năng chính: Lưu thông tin hóa đơn thanh toán khi mua trang in• Các thuộc tính:<ul style="list-style-type: none">– userId: int– numPage: int– sizePage: string– price : int– payConfirm : bool	Với BuyPageController: (Association) cung cấp truy vấn lấy, thêm, sửa, xóa dữ liệu. User: Lấy ID của người dùng và cập nhật thông tin của hóa đơn, mỗi Bill chỉ chứa 1 UsePrint
PrintConfig	<ul style="list-style-type: none">• Chức năng chính: Cài đặt các cấu hình trang in• Các thuộc tính:<ul style="list-style-type: none">– numPage: int– sizePage: string– copies : int– sides: int• Các phương thức:<ul style="list-style-type: none">+ <code>setConfig(numPage: int, sizePage: int, copies :int, sides : int) : void</code> (Cài đặt các cấu hình cho trang in như numPage (số trang in), sizePage(kích thước trang in), copies (số bản in), sides (số mặt mỗi trang))	Với PrintView: (Dependency) PrintView sử dụng fucntion của PrintConfig

4 MVP - Wireframe Details

Xem MVP - Wireframe FULL của nhóm tại [đây](#)

4.1 Trang chủ khi chưa đăng nhập

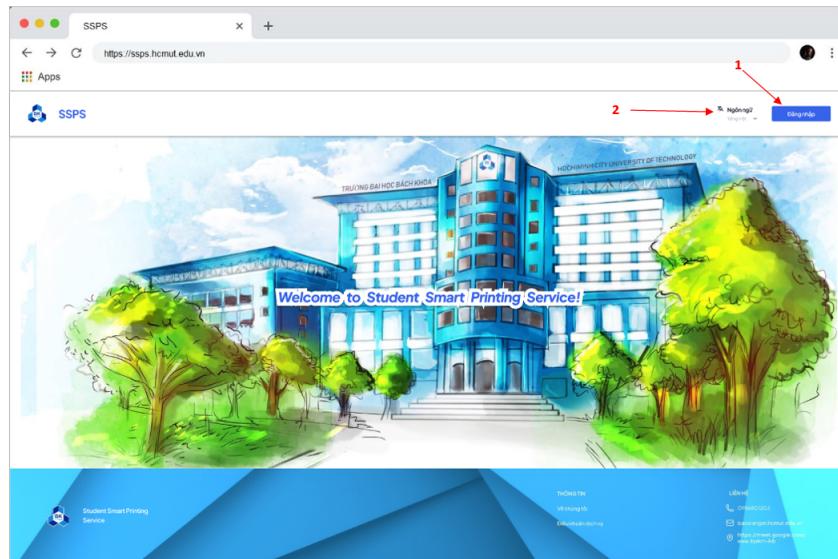


Figure 17: Giao diện trang chủ khi chưa đăng nhập

STT	Mô tả	Hành động
1	Dùng để đăng nhập	Chuyển đến trang đăng nhập
2	Dùng để chuyển đổi ngôn ngữ	Không

4.2 Giao diện khi nhấn nút Đăng nhập

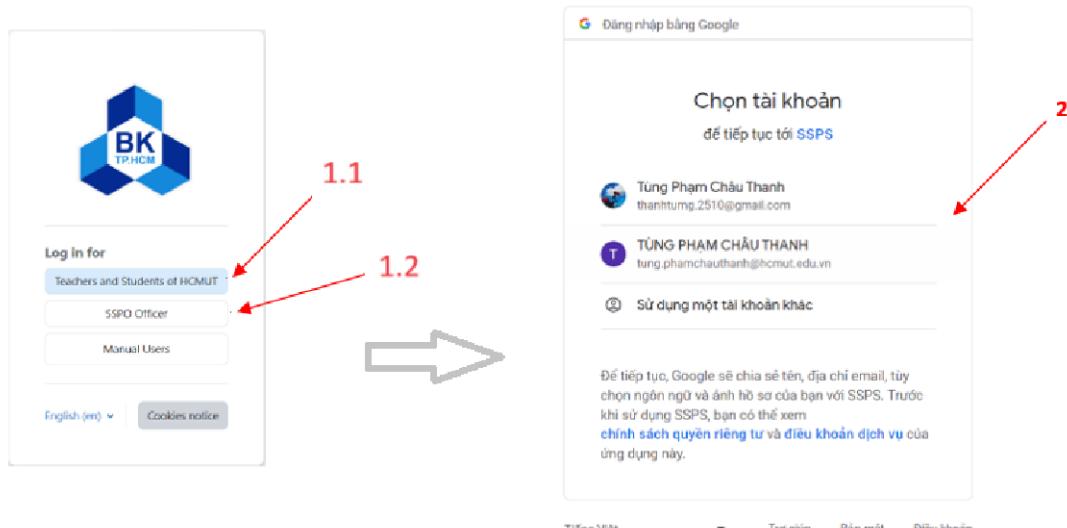


Figure 18: Giao diện khi nhấn nút Đăng nhập

STT	Mô tả	Hành động
1	1.1 Đăng nhập với tư cách là Student 1.2 Đăng nhập với tư cách là Officer	Chuyển đến trang login của Google
2	Đăng nhập với tài khoản Google	Chuyển tới trang chủ khi đăng nhập thành công

4.3 Giao diện khi đã đăng nhập

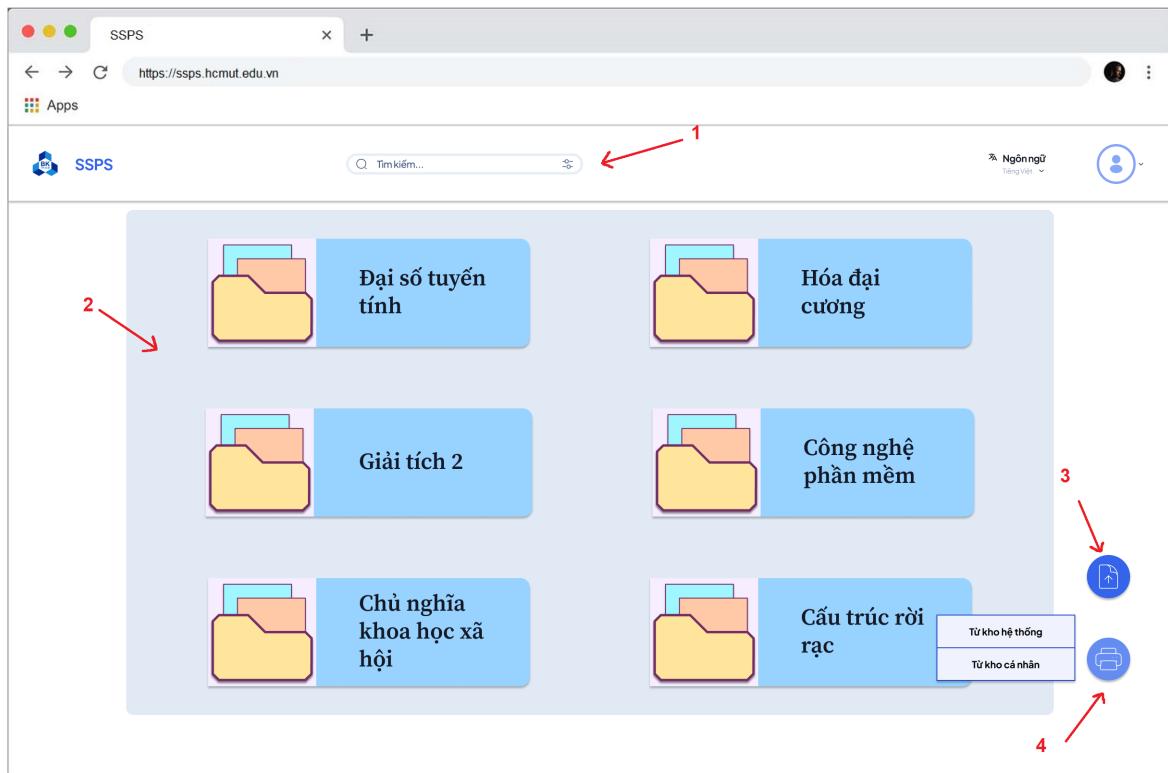


Figure 19: Giao diện khi đã đăng nhập

STT	Mô tả	Hành động
1	Tìm kiếm tài liệu	Hiển thị tài liệu
2	Hiển thị các Folder theo từng môn học	Không
3	Upload tài liệu	Mở modal upload tài liệu
4	In tài liệu	Mở modal in tài liệu

4.4 Giao diện khi đã nhấn nút In từ Kho cộng đồng hoặc cá nhân

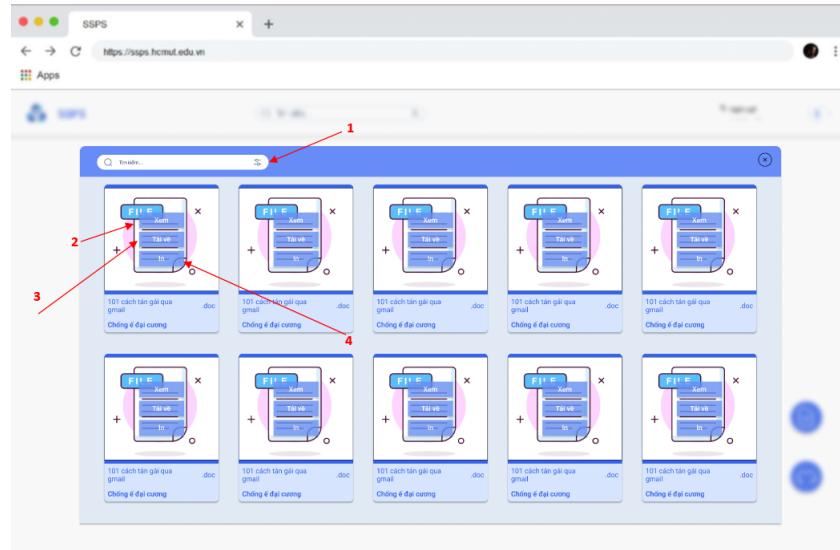


Figure 20: Giao diện khi đã nhấn nút In từ Kho công động cá nhân

STT	Mô tả	Hành động
1	Tìm kiếm tài liệu	Hiển thị tài liệu
2	Xem tài liệu	Mở tab mới theo link Drive của tài liệu đó
3	Tải xuống tài liệu	Không
4	In tài liệu	Mở modal in tài liệu

4.5 Giao diện khi cấu hình cài đặt In

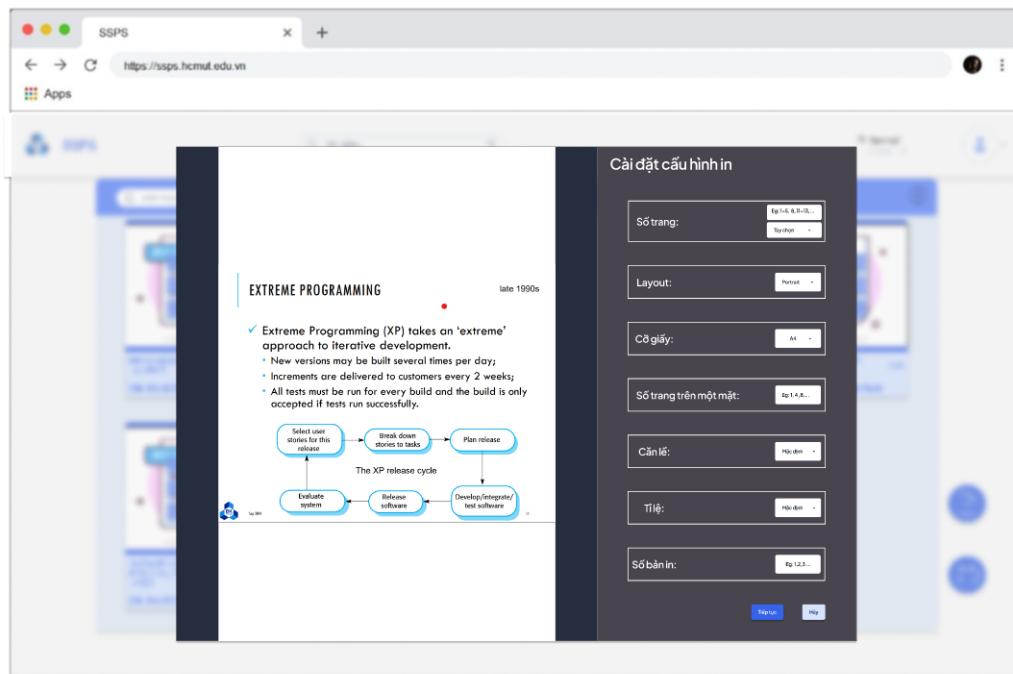


Figure 21: Giao diện khi cấu hình cài đặt In

Student cần cài đặt các thông số in: Số trang, Layout, Cỡ giấy (A3,A4,...), Số trang trên 1 mặt, Căn lề, Tỉ lệ, Số bản in.

4.6 Giao diện khi mua trang in

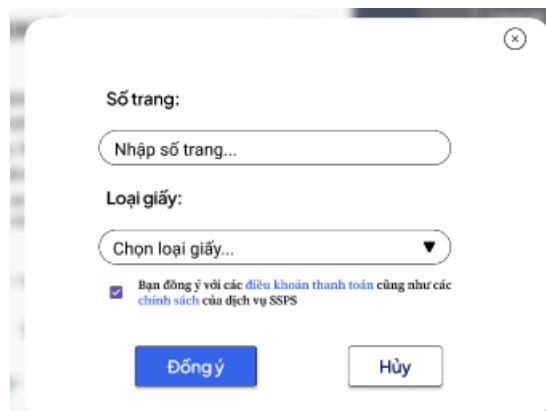


Figure 22: Giao diện khi mua trang in

Student cần nhập số trang và chọn loại giấy cần Mua.

4.7 Giao diện Chọn máy in và thời gian in

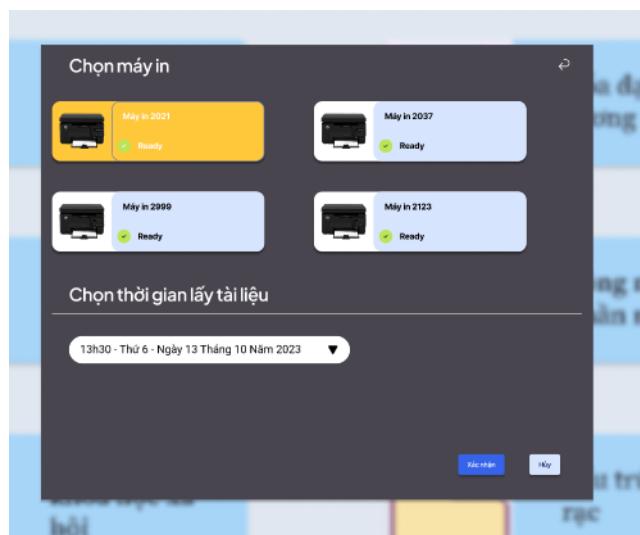


Figure 23: Giao diện Chọn máy in và thời gian in

Student cần chọn máy để in và thời gian lấy tài liệu.

4.8 Giao diện khi Upload

Khi Student upload tài liệu lên cần chọn vị trí lưu (Kho cộng đồng hoặc Kho cá nhân), chọn môn học .

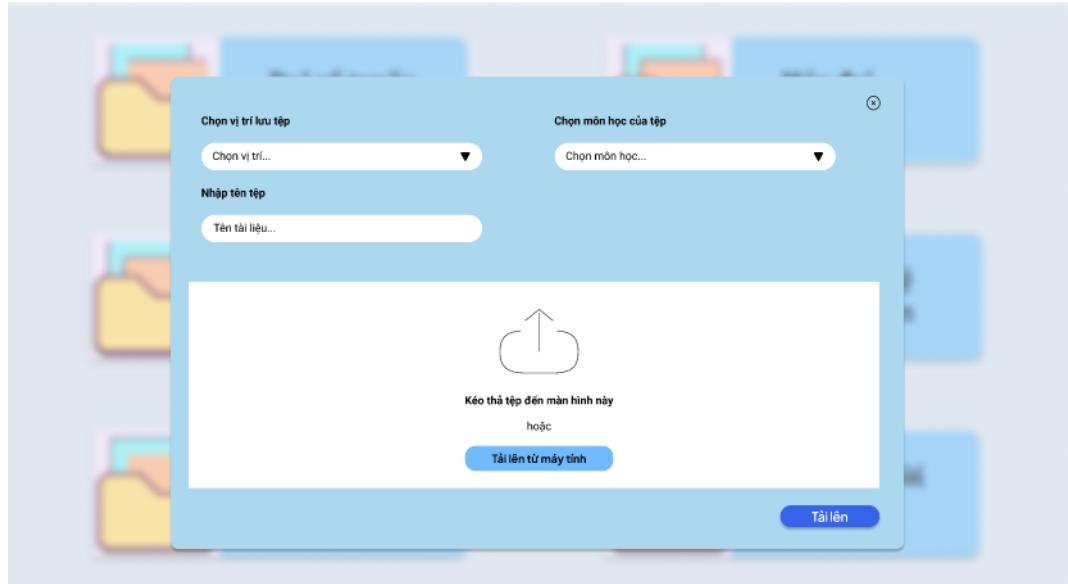


Figure 24: Giao diện khi Upload

4.9 Giao diện Theo dõi Lịch sử và tình trạng in

Lịch sử và tình trạng in

#	Tên tài liệu	Số trang	Ngày đặt in	Dung lượng	Ngày lấy	Trạng thái
1	Slide Công nghệ phần mềm HK231	15	16/10/2023	5.2 MB	9h00 - 19/10/2023	Đang trong hàng đợi
2	Đề thi Cuối kỳ DSA HK221	5	15/10/2023	2 MB	13h00 - 18/10/2023	Đang trong hàng đợi
3	Đề thi Giữa kỳ Mô hình hóa toán học HK...	5	11/10/2023	1.7 MB	7h00 - 13/10/2023	In thành công
4	Tổng hợp câu hỏi ôn tập Kiến trúc máy t...	20	01/10/2023	4.5 MB	14h30 - 5/10/2023	In thành công
5	Bài tập lớn LSD HK231	25	27/09/2023	6.0 MB	15h00 - 29/09/2023	In thành công
6	Mẫu đơn đánh giá ĐRL	2	10/09/2023	0.4 MB	8h00 - 12/09/2023	In thất bại
7	Giáo trình Triết học Mác Lê-nin	150	29/08/2023	37.2 MB	13h00 - 03/09/2023	In thành công

[«<Trang trước](#) 1 2 ... 9 10 [Trang sau>](#)

Figure 25: Giao diện Theo dõi Lịch sử và tình trạng in

Sinh viên có thể theo dõi tình trạng các file tài liệu mà sinh viên đã đặt in trong tối đa 3 tháng.

4.10 Giao diện quản lý sinh viên

Họ và tên	Email	Số trang khả dụng	Tùy chỉnh
Phạm Châu Thành Tùng	tung.phamchauthanh@hcmut.edu.vn	100	CẨM
Nguyễn Thái Thới	thoi.nguyenwibu@hcmut.edu.vn	120	CẨM
Nguyễn Ngọc Khánh My	my.nnk@hcmut.edu.vn	150	CẨM
Đoàn Minh Hiếu	hieu.doanminh@hcmut.edu.vn	46	CẨM
Nguyễn Trung Vương	vuong.nguyentrung@hcmut.edu.vn	90	CẨM
Kim Nhật Thành	thanh.kimnhat@hcmut.edu.vn	56	CẨM

Figure 26: Giao diện quản lý sinh viên (Danh sách sinh viên)

STT	Mô tả	Hành động
1	Tìm kiếm Sinh viên	Hiển thị danh sách sinh viên
2	Cấm sinh viên upload	Mở model xác nhận cấm
3	Sửa số trang khả dụng	Không

Họ và tên	Email	Số trang khả dụng	Tùy chỉnh
Võ Ngọc Thành Nhân	nhan.vo1113@hcmut.edu.vn	99	Hủy cấm
Nguyễn Ngọc Khánh My	my.nnk@hcmut.edu.vn	150	Hủy cấm

Figure 27: Giao diện quản lý sinh viên (Danh sách cấm)

Officer có thể hủy lệnh cấm cho Student.

4.11 Giao diện quản lý máy in

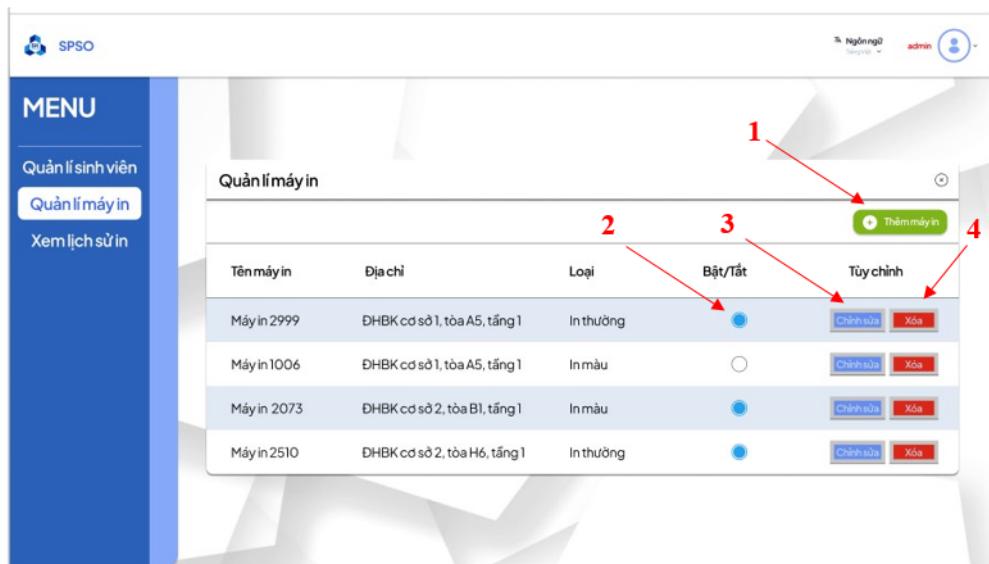


Figure 28: Giao diện quản lý máy in

STT	Mô tả	Hành động
1	Thêm máy in	Mở modal thêm máy in
2	Bật/tắt máy in	Không.
3	Chỉnh sửa	Mở modal chỉnh sửa thông tin
4	Xóa	Mở modal xác nhận

III ARCHITECTURE DESIGN

1 Architectural approach and Modules for the whole HCMUT-SSPS system

1.1 Architectural approach

Để hiện thực HCMUT-SSPS system, nhóm chọn hướng tiếp cận kiến trúc theo mô hình MVC (Model - View- Controller), trong đó:

- Người dùng truy cập và sử dụng hệ thống trên (**browser**)
- **View** : đây là phần giao diện dành cho người dùng, hiển thị các đối tượng trong hệ thống (các biểu mẫu, thông báo, ...) để người dùng có thể quan sát và tương tác được. Ngoài ra, **View** sẽ lắng nghe các sự kiện và thu nhận dữ liệu đầu vào từ người dùng (input) khi họ tương tác với hệ thống, sau đó chuyển giao cho Controller xử lý, rồi thực hiện hiển thị ra màn hình những kết quả/ dữ liệu đã xử lý từ Model hoặc hiển thị các trang theo input.
- **Controller**: là nơi điều khiển sự tương tác giữa hai thành phần **Model** và **View**.
 - Request processing: **Controller** thu nhận và xử lý các yêu cầu từ người dùng được đưa đến thông qua **View**, nhận vào input rồi thực hiện các cập nhật tương ứng.
 - Application-specific logic: **Controller** là nơi chứa các phương thức xử lý chính của hệ thống, thực hiện các chức năng cụ thể để phục vụ cho hệ thống.
 - Data validation: kiểm tra tính hợp lệ của dữ liệu nhập vào (tên tài khoản, mật khẩu, thông tin tài khoản, ..).
- **Model**: là nơi quản lý dữ liệu của hệ thống, làm việc với Database (nơi lưu trữ dữ liệu hệ thống) và xử lý các thao tác logic.
- **Database** : là nơi lưu trữ thông tin người dùng, sinh viên, máy in, lịch sử in, lịch sử mua trang, tài liệu,...

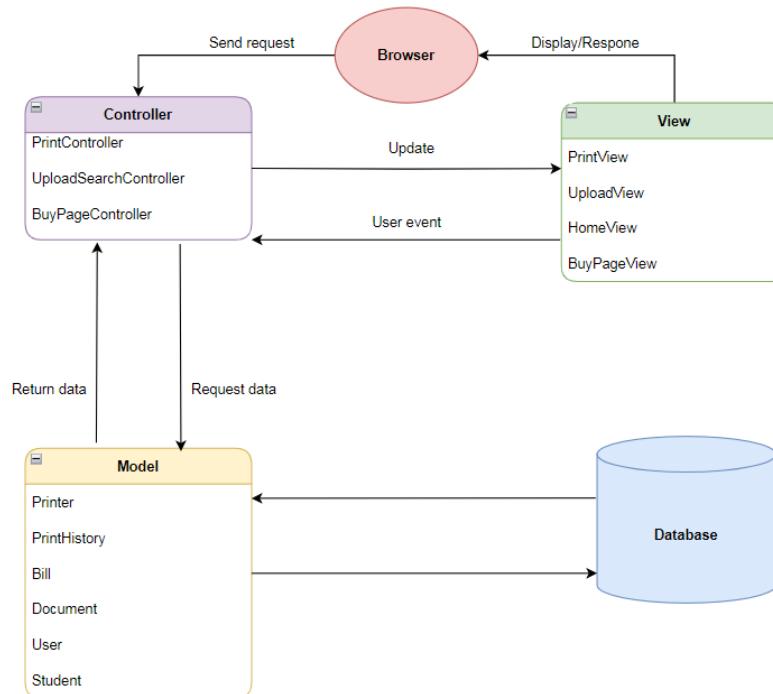


Figure 29: Mô hình kiến trúc MVC



1.2 List of Modules

1. Module In tài liệu

Input:

- Student X.
- Thông tin tài liệu (id tài liệu, tên tài liệu, link drive,...).
- Thông số cấu hình (Số trang, số mặt, kích thước,...).

Output:

- Danh sách lịch sử in đã được cập nhật.
- Số trang của Student X được cập nhật (trừ đi số trang đã sử dụng để in).

2. Module Upload tài liệu

Input:

- Student X.
- Tệp tài liệu Y.
- Tên và vị trí lưu trữ tài liệu Y.
- Môn học mà tài liệu Y thuộc về.

Output:

- Thông báo tệp tài liệu được tải lên có phù hợp yêu cầu không.
- Vị trí của tệp trong kho lưu trữ theo yêu cầu người dùng (kho hệ thống hoặc kho cá nhân).
- Đường dẫn đến vị trí tệp tài liệu.
- Danh sách tài liệu được cập nhật.

3. Module Tìm kiếm tài liệu

Input:

- Student X.
- Vị trí tìm kiếm (Kho cộng đồng hoặc Kho cá nhân).
- Thông tin tài liệu Y.

Output:

- Hiển thị danh sách tài liệu theo thông tin tài liệu Y đã tìm kiếm (nếu có). Nếu không có thì hiển thị là không tìm thấy tài liệu.

4. Module Mua trang in

Input:

- Student X.
- Số lượng trang in muốn mua.
- Kích thước trang muốn mua.

Output:

- Cập nhật danh sách lịch sử mua trang in.
- Hiển thị thông báo (mua thành công hay thất bại).

2 Component Diagram

Xem Component Diagram FULL của nhóm tại [đây](#)

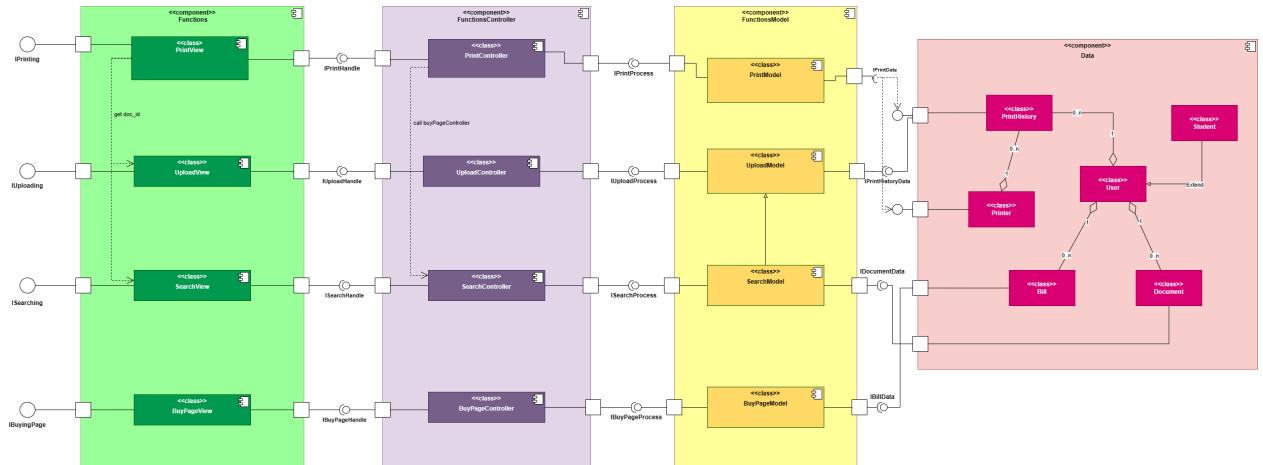


Figure 30: Component Diagram

2.1 Mô tả Component Diagram

Tầng View gồm có 1 component chính:

1. Functions:

- Chức năng chính: hiện thực các class cung cấp giao diện In, Tải lên, Tìm kiếm, Mua trang in.
- Gồm các class: **PrintView**, **UploadView**, **SearchView**, **BuyPageView**. Class **PrintView** cung cấp giao diện cài đặt cấu hình in (số trang, cỡ giấy, căn lề, layout, số trang trên một mặt, tỉ lệ, số bản in). Class **UploadView** cung cấp giao diện tải lên tài liệu (cung cấp lựa chọn vị trí lưu tài liệu, nhập tên tài liệu, chọn môn học). Class **SearchView** cung cấp giao diện tìm kiếm tài liệu (tìm kiếm trên kho cá nhân và tìm kiếm trên kho cộng đồng). Class **BuyPageView** cung cấp giao diện mua trang in (cung cấp lựa chọn nhập số trang muốn mua, loại giấy).
- Mỗi quan hệ giữa các thành phần: Class **PrintView** dựa vào doc_id do **SearchView** cung cấp để hiển thị tài liệu mà người dùng muốn tìm kiếm; sử dụng doc_id mà **UploadView** cung cấp khi Upload tài liệu.
- Class **PrintView** hiện thực (realization) interface **IPrinting**, **UploadView** hiện thực interface **IUploading**, class **SearchView** hiện thực interface **ISearching**, **BuyPageView** hiện thực interface **IBuyingPage**.
- Component yêu cầu (require) các interface **IPrintHandle** cho **PrintView**, **IUploadHandle** cho **UploadView**, **ISearchHandle** cho **SearchView**, **IBuyPageHandle** cho **BuyPageView**.

Tầng Controller gồm có 1 component chính:

1. FunctionsController:

- Chức năng chính: hiện thực các class cung cấp các phương thức xử lý sự kiện và điều khiển luồng dữ liệu liên quan đến việc in, upload, search và buy page.
- Gồm các class: **PrintController**, **UploadController**, **SearchController**, **BuyPageController**. Class **PrintController** cung cấp các phương thức điều khiển luồng dữ liệu in ấn, kiểm soát tài liệu được in, điều hướng đến các controller khác khi thỏa điều kiện, cập nhật giao diện. Class **UploadController** xử lý, điều khiển các tài liệu được tải lên từ người dùng,



cập nhật giao diện. Class **SearchController** xử lý, điều khiển các luồng dữ liệu tìm kiếm, cập nhật giao diện. Class **BuyPageController** xử lý, điều khiển các luồng dữ liệu mua trang in, cập nhật giao diện.

- Mỗi quan hệ giữa các thành phần: Class **PrintController** sẽ sử dụng object **buyPageController** của chính nó để gọi đến class **BuyPageView** thông qua class **BuyPageController** khi người dùng muốn tiếp tục sử dụng dịch vụ in nhưng không còn đủ số trang khả dụng.
- Class **PrintController** hiện thực interface *IPrintHandle*, class **UploadController** hiện thực interface *IUploadHandle*, class **SearchController** hiện thực interface *ISearchHandle*, class **BuyPageController** hiện thực interface *IBuyPageHandle*.
- Class **PrintController** yêu cầu interface *IPrintProcess*, class **UploadController** yêu cầu interface *IUploadProcess*, class **SearchController** yêu cầu interface *ISearchProcess*, class **BuyPageController** yêu cầu interface *IBuyPageProcess*.

Tầng Model gồm có 2 component chính:

1. FunctionsModel:

- Chức năng chính: lưu trữ và thực hiện truy vấn dữ liệu, xử lý logic liên quan đến việc in, upload, search và buy page.
- Gồm các class **PrintModel**: lưu trữ danh sách lịch sử in và danh sách các máy in, đồng thời có các phương thức điều chỉnh các danh sách trên. **UploadModel**: lưu trữ danh sách tài liệu và các phương thức để điều chỉnh danh sách đó, **searchModel**: có các phương thức để tìm kiếm tài liệu, class **BuyPageModel**: lưu trữ danh sách các hóa đơn và các hàm để điều chỉnh.
- Mỗi quan hệ giữa các thành phần: class **searchModel** kế thừa **UploadModel** để có thể truy cập *ListDoc*.
- Class **PrintModel** hiện thực interface *IPrintProcess*, class **UploadModel** hiện thực interface *IUploadProcess*, class **searchModel** hiện thực interface *ISearchProcess*, class **BuyPageModel** hiện thực interface *IBuyPageProcess*.
- Class **PrintModel** yêu cầu interface *IPrintData*, class **UploadModel** yêu cầu interface *IPrintHistoryData*, class **searchModel** yêu cầu interface *IDocumentData*, class **BuyPageModel** yêu cầu interface *IBillData*.

2. Data

- Chức năng chính: lưu trữ và thực hiện truy vấn dữ liệu liên quan đến các công việc in, upload, search và buy page.
- Gồm các class: **PrintHistory** (Lịch sử in), **Student** (Sinh viên), **User** (Người dùng), **Printer** (Máy in), **Bill** (Hóa đơn), **Document** (Tài liệu).
- Class **PrintHistory** hiện thực interface *IPrintData* (cùng với class **Printer**) và **IPrintHistory**, class **Bill** hiện thực interface *IBillData*, class **Document** hiện thực interface *IDocumentData*.

Bảng mô tả các Interface:

Interface	Chức năng	Phương thức
IPrinting	Cung cấp giao diện giúp người dùng sử dụng dịch vụ in ấn.	renderPrintView(); renderPrintConfig(); displayPrinterList():void; onClickUpPrintButton(): void; selectDoc(docId): void; selectPrinter(printId): void;
IUploading	Cung cấp giao diện giúp người dùng upload tài liệu lên kho lưu trữ cá nhân hoặc hệ thống.	renderUploadView(); CheckFileType(file): bool; onClickUploadButton(userId, file): void;
ISearching	Cung cấp giao diện giúp người dùng tìm kiếm tài liệu trên kho hệ thống và kho cá nhân.	renderSearchView(); displayDocList(doc_infor: string): list<Document>; onClickSearchButton(): void;
IBuyingPage	Cung cấp giao diện giúp người dùng thực hiện việc mua thêm trang in.	renderBuyPageView(); onChangeNumPage(numPage: int, sizePage: string): void; onClickBuy(): void; confirmBuy(): void; checkValidNumber(numPage: int, sizePage: string): bool;
IPrintHandle	Cung cấp các phương thức giúp điều khiển luồng dữ liệu liên quan đến các tác vụ in.	processPrint(userId:int, docId:int, printerId:int, printConfig: PrintConfig); displayAllPrinter(): object
IUploadHandle	Cung cấp phương thức giúp điều khiển luồng dữ liệu liên quan đến các tác vụ upload.	Upload(userId: int, file: File): object;
ISearchHandle	Cung cấp các phương thức giúp điều khiển luồng dữ liệu liên quan đến dữ liệu tìm kiếm, thao tác tìm kiếm.	getDocument(doc_infor: string): object;
IBuyPageHandle	Cung cấp các phương thức giúp điều khiển luồng dữ liệu liên quan đến thao tác và dữ liệu khi mua trang in.	BuyPage(userId: int, numPage: int, sizePage: string): obj;
IPrintProcess	Cung cấp các phương thức giúp xử lý logic liên quan đến thao tác in.	print(userId: int, file: Document, printConfig: PrintConfig): object; addPrintHistory(printHistory: PrintHistory): object; getAllPrinter(): object;
IUploadProcess	Cung cấp các phương thức giúp xử lý logic liên quan đến thao tác upload.	addDoc(userId: int, doc: Document): object; getAllDoc(): list<Document>; upLoadToGgDrive(folderId, file: File): object;
ISearchProcess	Cung cấp các phương thức để xử lý các logic liên quan tới việc tìm kiếm tài liệu.	findAllDoc(doc_infor): object;
IBuyPageProcess	Cung cấp các phương thức giúp xử lý logic liên quan đến việc mua trang in.	addBuyHistory(userId: int, numPage: int, sizePage: string): object; getAllBill(): object;



IPrintData	Cung cấp API lấy dữ liệu của máy in và lịch sử in để phục vụ cho thao tác in.	getPrintLocation(): string; getPrinterStatus(): bool;
IPrintHistoryData	Cung cấp API lấy dữ liệu của lịch sử in để phục vụ cho việc theo dõi lịch sử tình trạng in.	getPrintHistory(): string;
IDocumentData	Cung cấp API lấy dữ liệu của các tài liệu để phục vụ cho việc tìm kiếm và in.	getDocName(): string; getDocId(): string; getLinkDoc(): string;
IBillData	Cung cấp API để lấy dữ liệu về các hóa đơn và lịch sử thanh toán khi mua trang in của người dùng.	

3 Deployment Diagram

Xem Deployment Diagram FULL của nhóm tại [đây](#)

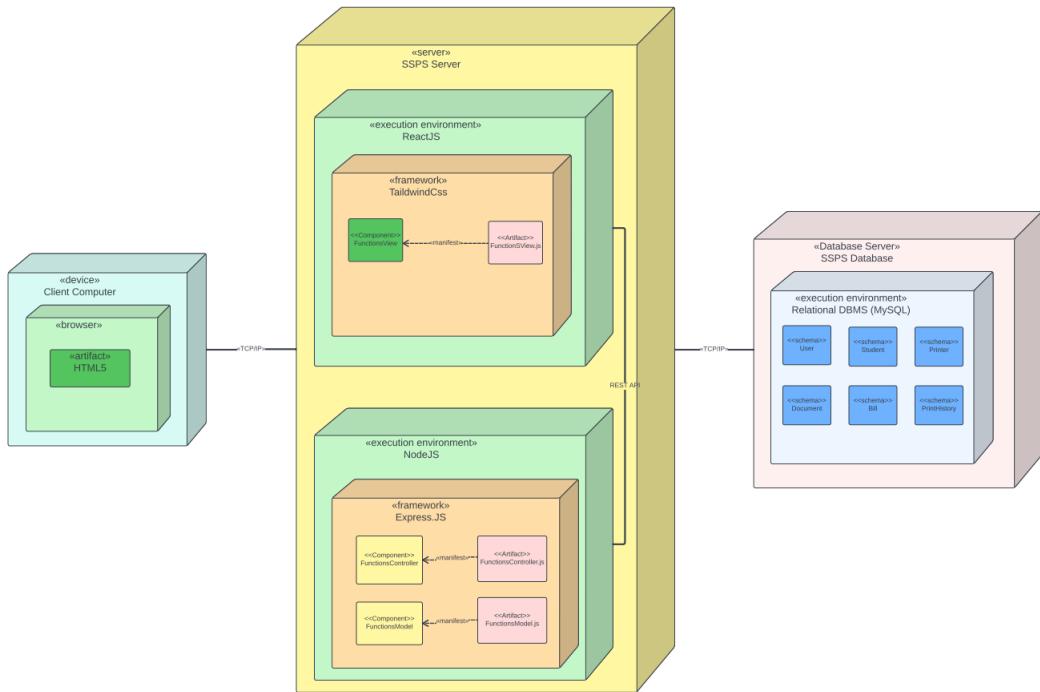


Figure 31: Deployment Diagram

Mô tả Deployment Diagram

- Hệ thống là một ứng dụng Web, được phát triển theo kiến trúc MVC, bao gồm 3 máy chủ chính: Client, Server và Database Server.
- Máy chủ Client sẽ kết nối với Server thông qua giao thức TCP/IP, dữ liệu nhận được từ Server là các file hiện thực giao diện và sẽ được hiển thị trên browser website tại máy tính người dùng (Client) thông qua trình đọc HTML5 được nhúng trong trình duyệt.
- Máy chủ Server bao gồm môi trường thực thi (execution environment) các module chức năng chính của hệ thống. Mọi trường thực thi phía back-end là NodeJS, sử dụng Framework là Express.js, bao gồm 2 component chính là **FunctionsController** và **FunctionsModel** đảm nhận các tác vụ xử lý chính và điều khiển luồng dữ liệu liên quan đến các chức năng như in tài liệu, tải tài liệu, tìm kiếm tài liệu, mua thêm trang,... Mọi trường thực thi phía front-end là ReactJS, sử dụng framework TailwindCss, component chính là **FunctionsView** đảm nhận việc hiện thực các giao diện người dùng liên quan đến in tài liệu, tải tài liệu, tìm kiếm tài liệu, mua thêm trang,... Các component sẽ được hiện thực từ các file mã nguồn JavaScript tương ứng. Thông qua các REST API, front-end và back-end có thể giao tiếp được với nhau để cập nhật, hiển 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 lên giao diện hoặc cập nhật lại dữ liệu mỗi khi người dùng tương tác với ứng dụng, sau đó dữ liệu sau khi cập nhật sẽ được truyền qua giao thức TCP/IP sang Server và truyền sang máy Client để cập nhật lại giao diện.
- Máy chủ Database bao gồm môi trường thực thi là DBMS MySQL. Trong phạm vi các module trên, máy chủ Database lưu trữ các thông tin bao gồm: User, Student, Printer, Document, Bill, PrintHistory,... ở dạng bảng có cấu trúc và các mối quan hệ với nhau.



IV IMPLEMENTATION SPRINT 1

1 Github for version control

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

2 Usability test

2.1 Participants

Form khảo sát của nhóm: Xem tại [đây](#)

- Thành viên tham gia khảo sát: Tất cả sinh viên đã điền form khảo sát.
- Chi tiết danh sách của những người đã tham gia điền form: Tại [đây](#).

2.2 Environment

- Tất cả thành viên tiến hành khảo sát sản phẩm thông qua đường link [Figma](#).

2.3 Tasks

Tất cả thành viên tham gia sẽ được yêu cầu làm các task sau :

- Task 1 : Đăng nhập vào hệ thống bằng email của trường Đại học Bách Khoa TP.HCM.
- Task 2 : Upload tài liệu lên hệ thống.
- Task 3 : Tìm kiếm tài liệu.
- Task 4 : In tài liệu.
- Task 5 : Mua Trang in.
- Task 6 : Các chức năng còn lại.

2.4 Success Criteria

- Task 1: Đăng nhập vào hệ thống bằng email của trường Đại học Bách Khoa TP.HCM.
 - *Dễ sử dụng*: Người dùng có thể dễ dàng tìm thấy và nhập thông tin đăng nhập không?
 - *Nhìn nhận tổng thể*: Người dùng có thấy quy trình đăng nhập là một trải nghiệm thuận lợi không?
- Task 2: Upload tài liệu lên hệ thống.
 - *Dễ sử dụng*: Quy trình upload tài liệu có dễ hiểu không?
 - *Phản hồi người dùng*: Người dùng có gặp khó khăn hoặc vướng mắc nào trong quá trình tải lên không?
- Task 3: Tìm kiếm tài liệu.
 - *Dễ sử dụng*: Giao diện tìm kiếm có dễ sử dụng không?
 - *Phản hồi người dùng*: Người dùng có gặp khó khăn hoặc vướng mắc nào trong quá trình tìm kiếm không?
- Task 4: In tài liệu.
 - *DỄ SỬ DỤNG*: Quy trình in tài liệu có dễ hiểu không?

- *Phản hồi người dùng:* Người dùng có gặp khó khăn hoặc vướng mắc nào trong quá trình in không?
- Task 5: Mua Trang In.
 - *Dễ sử dụng:* Quy trình mua thêm trang có dễ hiểu không?
 - *Phản hồi người dùng:* Người dùng có gặp khó khăn hoặc vướng mắc nào trong quá trình mua trang in không?

2.5 Feedback

2.5.1 Task 1: Đăng nhập vào hệ thống bằng email của trường Đại học Bách Khoa TP.HCM

Dưới đây là biểu đồ thống kê kết quả trả lời các câu hỏi liên quan tới Task 1 của các thành viên tham gia testing:

- Câu hỏi số 1.

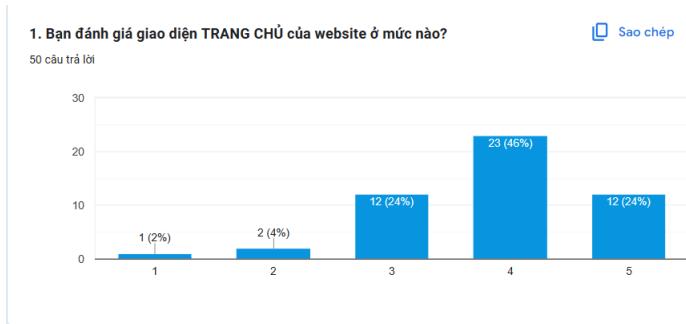


Figure 32: Task 1 - Question 1

- Câu hỏi số 2

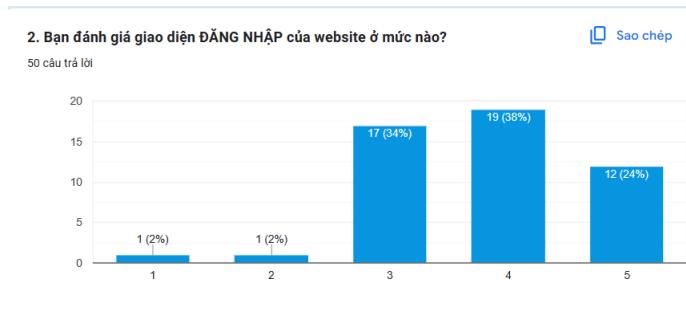


Figure 33: Task 1 - Question 2

Các góp ý của người dùng về giao diện ĐĂNG NHẬP

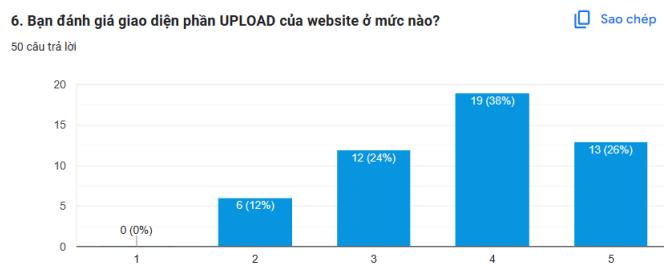
Cần trang trí thêm
Hơi chói, cần tinh chỉnh màu sắc
Cần 1 đoạn văn hay gì đó để giới thiệu dịch vụ
nên đổi footer khác
cần đổi footer
chữ ở footer nên bị chói nên đổi màu và tăng kích thước chữ
Sửa lại Footer
Trang chủ này ngoài chức năng đăng nhập ra nếu như không còn tính năng khác thì có thể nên loại bỏ, khi truy cập sẽ vào thẳng trang login để người dùng thuận tiện hơn
Nên đổi footer

Figure 34: Feedback task 1

2.5.2 Task 2: Upload tài liệu lên hệ thống

Dưới đây là biểu đồ thống kê kết quả trả lời các câu hỏi liên quan tới Task 2 của các thành viên tham gia testing:

- Câu hỏi số 6

**Figure 35:** Task 2 - Question 6

Các góp ý của người dùng về giao diện UPLOAD

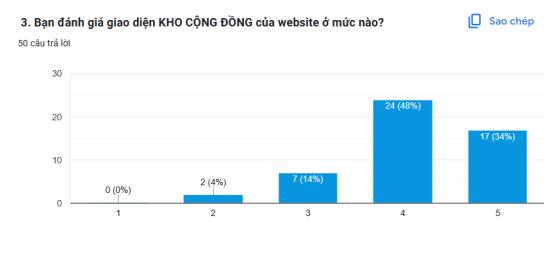
Hãy thêm chú thích về dung lượng giới hạn
Nên cho cỡ chữ to hơn 1 chút
Nếu có thêm option "tải lên và in" thì sẽ tiện cho người dùng muốn in file ngay và không cần lưu trữ trong bộ nhớ.

Figure 36: Feedback task 2

2.5.3 Task 3: Tìm kiếm tài liệu

Dưới đây là biểu đồ thống kê kết quả trả lời các câu hỏi liên quan tới Task 3 của các thành viên tham gia testing:

- Câu hỏi số 3

**Figure 37:** Task 3 - Question 3

- Câu hỏi số 4

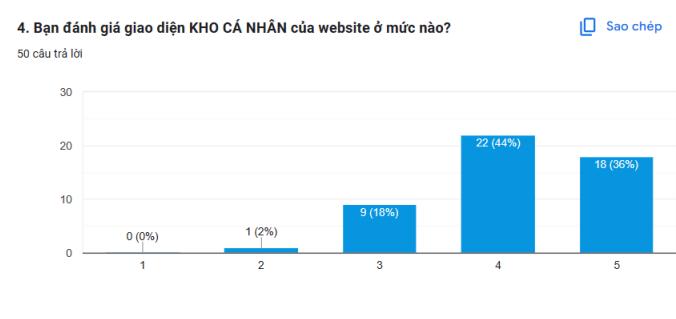


Figure 38: Task 3 - Question 4

- Câu hỏi số 5

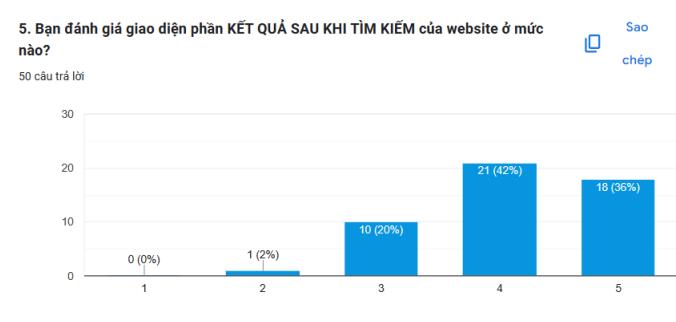


Figure 39: Task 3 - Question 5

2.5.4 Task 4: In tài liệu

Dưới đây là biểu đồ thống kê kết quả trả lời các câu hỏi liên quan tới Task 4 của các thành viên tham gia testing:

- Câu hỏi số 7

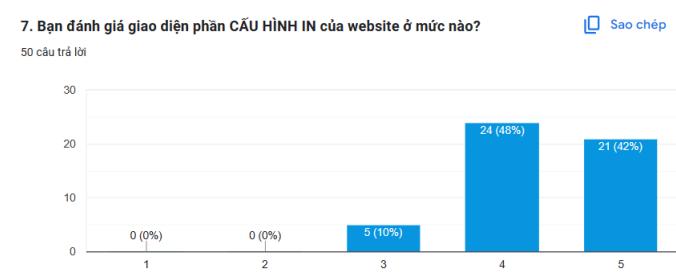


Figure 40: Task 4 - Question 7

- Câu hỏi số 8

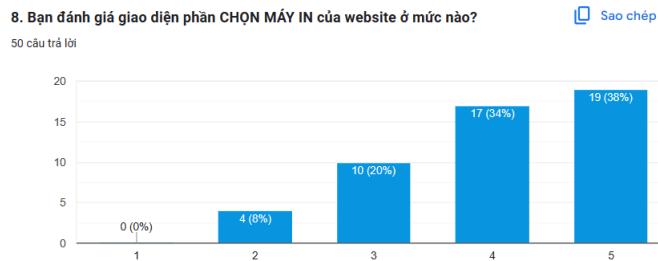


Figure 41: Task 4 - Question 8

Các góp ý của người dùng về giao diện

Nếu được hãy hiển thị số đơn in ở mỗi máy
Nên bổ sung thông tin máy in (vị trí, loại...)

Figure 42: Feedback task 4

2.5.5 Task 5: Mua trang in

Dưới đây là biểu đồ thống kê kết quả trả lời các câu hỏi liên quan tới **Task 6** của các thành viên tham gia testing:

- Câu hỏi số 10

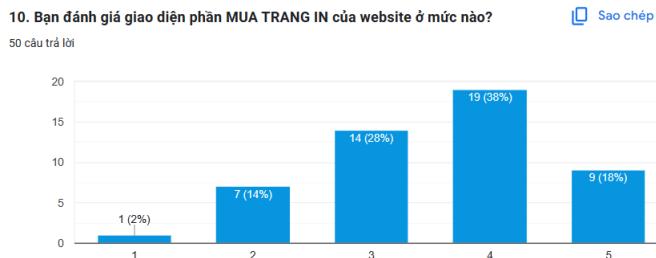


Figure 43: Task 5 - Question 10

Các góp ý của người dùng về giao diện

Hơi đơn giản, cần thêm một số thuộc tính để đảm bảo quyền lợi thanh toán
Chữ hơi nhỏ nên tăng kích thước chữ
Chữ hơi nhỏ nên tăng kích cỡ
Yêu cầu đề bài có nói là 1 trang a3 sẽ bằng 1 trang a4. Vì vậy việc chọn loại giấy mua có vẻ không cần thiết.

Figure 44: Feedback task 5

2.5.6 Task 6: Các chức năng còn lại

Dưới đây là biểu đồ thống kê kết quả trả lời các câu hỏi liên quan tới **Task 6** của các thành viên tham gia testing:

- Câu hỏi số 9

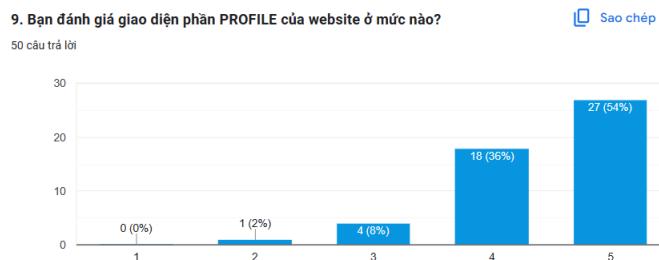


Figure 45: Task 6 - Question 9

- Câu hỏi số 11

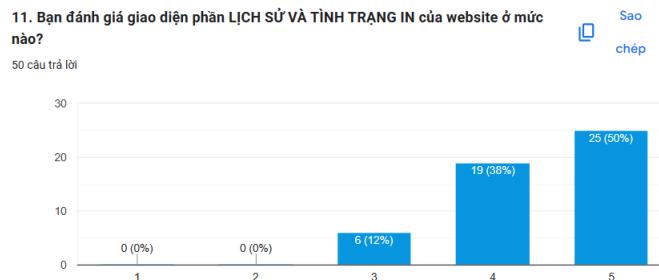


Figure 46: Task 6 - Question 11

- Câu hỏi số 12

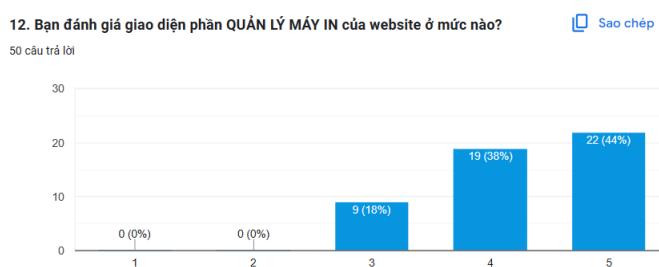


Figure 47: Task 6 - Question 12

- Câu hỏi số 13

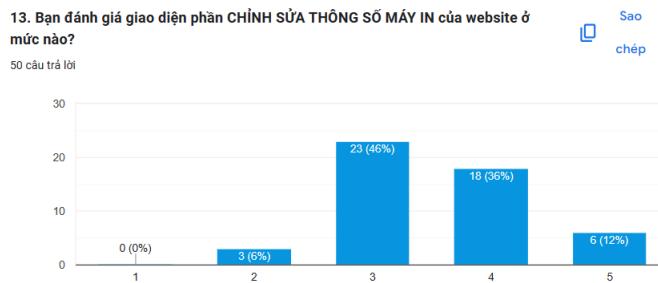


Figure 48: Task 6 - Question 13

- Câu hỏi số 14

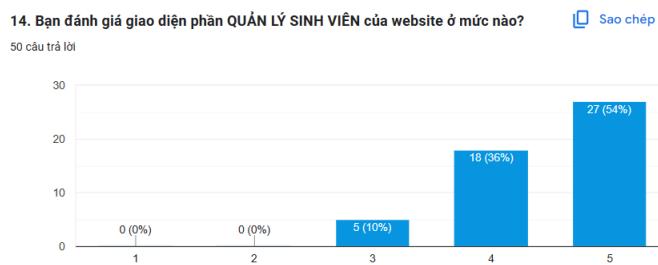


Figure 49: Task 6 - Question 14

- Câu hỏi số 15

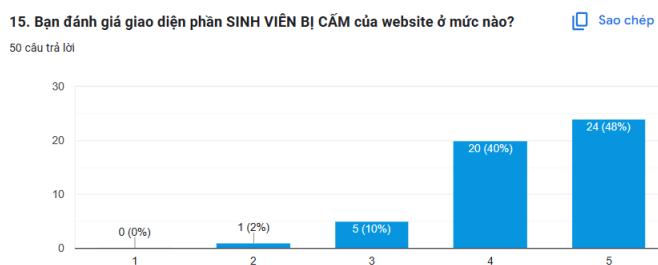


Figure 50: Task 6 - Question 15

2.5.7 Tổng quan

- Câu hỏi số 16

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

Sao
chèp

49 câu trả lời

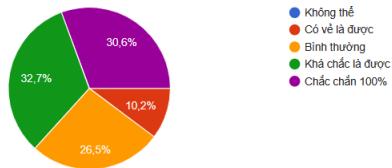


Figure 51: Tổng quan - Question 16

- Câu hỏi số 17

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

Sao
chèp

50 câu trả lời

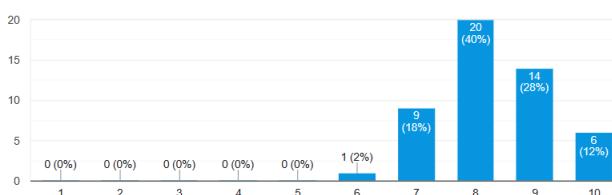


Figure 52: Tổng quan - Question 17

Xem bảng chi tiết tất cả các kết quả trả lời của thành viên tham gia: Xem tại [đây](#)