

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



Công nghệ phần mềm

Bài tập lớn

Student Smart Printing Service HCMUT SSPS

GVHD:	Trần Trương Tuấn Phát	
SV thực hiện:	Chế Minh Đức	– 2210783
	Nguyễn Đình Đức	– 2210794
	Phạm Lê Hoài Hải	– 2210901
	Hoa Toàn Hạc	– 2210917
	Lê Quốc Huy	– 2211194
	Hà Tiến Thắng	– 2213184
	Hà Lê Tín	– 2213484

TP. HỒ CHÍ MINH, THÁNG 9, 2024



Danh sách thành viên & Nhiệm vụ

Task 1:

Thời gian	Họ và tên	MSSV	NHIỆM VỤ
Ngày 18/09/2024	Chế Minh Đức	2210783	Tuần 1: Chỉnh sửa và bổ sung Tổng hợp và viết báo cáo
	Nguyễn Đình Đức	2210794	Tuần 1: Task 1.1 Task 1.2
	Phạm Lê Hoài Hải	2210901	Tuần 1: Task 1.1 Task 1.2
	Hoa Toàn Hạc	2210917	Tuần 1: Chỉnh sửa và bổ sung Tổng hợp và viết báo cáo
	Lê Quốc Huy	2211194	Tuần 1: Task 1.1 Task 1.2
	Hà Tiến Thắng	2213184	Tuần 1: Task 1.1 Task 1.2
	Hà Lê Tín	2213484	Tuần 1: Task 1.1 Task 1.2
Ngày 26/09/2024	Lê Quốc Huy	2211194	Vẽ use case cho chức năng chính Mô tả use case upload file, set print options dạng bảng
	Hà Tiến Thắng	2213184	Vẽ use case cho toàn hệ thống Mô tả use case choose printer, book print schedule dạng bảng
	Hà Lê Tín	2213484	Vẽ use case cho toàn hệ thống Mô tả use case print document dạng bảng
Ngày 07/10/2024	Chế Minh Đức	2210783	Chỉnh sửa góp ý use case Review các diagram đã vẽ
	Nguyễn Đình Đức	2210794	Chỉnh sửa góp ý use case Vẽ activity diagram cho chức năng in
	Phạm Lê Hoài Hải	2210901	Chỉnh sửa góp ý use case Vẽ sơ bộ ER diagram
	Hoa Toàn Hạc	2210917	Tổng hợp và viết báo cáo Lên lịch cho các hoạt động tiếp theo



Task 2:

Thời gian	Họ và tên	MSSV	NHIỆM VỤ
Ngày 20/10/2024	Chế Minh Đức	2210783	Chỉnh sửa góp ý hệ thống Review các diagram đã vẽ trong task 2
	Nguyễn Đình Đức	2210794	Hoàn thành activity diagram cho chức năng in
	Phạm Lê Hoài Hải	2210901	Hoàn thành ERD Hoàn thành class diagram
	Hoa Toàn Hạc	2210917	Tổng hợp và viết báo cáo Hoàn thành MVP
	Hà Lê Tín	2213484	hoàn thành sequence diagram cho mục đặt lịch và thiết lập tùy chọn in
	Hà Tiến Thắng	2213184	Hoàn thành mô tả sequence diagram
	Lê Quốc Huy	2211194	Hoàn thành phần còn lại của sequence diagram



Task 3:

Thời gian	Họ và tên	MSSV	NHIỆM VỤ
Ngày 5/11/2024	Chế Minh Đức	2210783	Mô tả Component Diagram
	Nguyễn Đình Đức	2210794	Vẽ Component Diagram
	Phạm Lê Hoài Hải	2210901	Thiết kế Data Layer
	Hoa Toàn Hạc	2210917	Thiết kế Presentation Layer
	Hà Lê Tín	2213484	Vẽ Architecture
	Hà Tiến Thắng	2213184	Vẽ Architecture Diagram
	Lê Quốc Huy	2211194	Thiết kế API Layer

Task 4 & 5:

Thời gian	Họ và tên	MSSV	NHIỆM VỤ
Ngày 17/11/2024	Chế Minh Đức	2210783	Review và chỉnh sửa diagram
	Nguyễn Đình Đức	2210794	Backend và Architecture
	Phạm Lê Hoài Hải	2210901	Databse
	Hoa Toàn Hạc	2210917	Khảo sát MVP, thiết kế full giao diện
	Hà Lê Tín	2213484	Front-end
	Hà Tiến Thắng	2213184	Front-end
	Lê Quốc Huy	2211194	Review và chỉnh sửa diagram



Mục lục

1	Bối cảnh và các bên liên quan	6
1.1	Bối cảnh	6
1.2	Bất cập	6
1.3	Giải pháp: Hệ thống in ấn thông minh SSPS	6
1.4	Các bên liên quan và nhu cầu	6
2	Yêu cầu với hệ thống	8
2.1	Yêu cầu chức năng	8
2.2	Yêu cầu phi chức năng (Non-Functional Requirements - NFR)	8
3	Use-case diagram	10
3.1	Use-case cho toàn bộ hệ thống	10
3.2	Use-case cho chức năng in	11
3.3	Use-case dạng bảng	12
4	Activity diagram	17
4.1	Mô tả	17
4.2	Diagram	18
5	Sequence diagram	19
5.1	Mô tả	19
5.2	Diagram	20
6	Class Diagram	24
6.1	Các thành phần chính của Class Diagram	24
6.1.1	Entity Class	24
6.1.2	UI Class	26
6.2	Diagram	26
7	Minimum Viable Product	27
7.1	Mô tả	27
7.2	MVP	28
8	Thiết kế kiến trúc phân lớp cho hệ thống SSPS	30
8.1	Lý do chọn kiến trúc	30
8.2	Mô tả kiến trúc	30
8.3	Diagram	31
8.4	View Layer	32
8.5	API Layer	34
8.5.1	Khái niệm API	34
8.5.2	Rest API	34
8.5.3	Request và Response	34
8.5.4	Sử dụng và quản lý API	35
8.6	Data Layer	35
8.6.1	Mô tả về Data Layer	35
8.6.2	ERD	37



9	Component Diagram	38
9.1	Mô tả	38

1 Bối cảnh và các bên liên quan

1.1 Bối cảnh

In ấn tài liệu vẫn là một nhu cầu thiết yếu trong môi trường giáo dục, đặc biệt tại Trường Đại học Bách khoa - ĐHQG TP.HCM. Sinh viên thường xuyên cần in tài liệu học tập, slide bài giảng hoặc tài liệu phục vụ thi cử. Hiện nay, việc in ấn chủ yếu thực hiện thông qua:

- Các cơ sở in ấn trong trường (như thư viện BK.B1 ở cơ sở 2).
- Các tiệm in bên ngoài.

1.2 Bất cập

- **Tại cơ sở in trong trường:**
 - Quá tải trong giờ cao điểm: Đặc biệt vào mùa thi hoặc đầu học kỳ, tình trạng chờ đợi lâu khiến việc in ấn bị trì hoãn.
 - Thiếu tính linh hoạt: Sinh viên không thể đặt lịch in hoặc tùy chỉnh theo nhu cầu đặc biệt.
- **Tại các tiệm in bên ngoài:**
 - Nguy cơ rò rỉ thông tin: Các tài liệu như bài tập lớn, đồ án hoặc các tài liệu cá nhân có thể bị lộ.
 - Vấn đề bản quyền: Nhiều tài liệu học tập thuộc sở hữu trí tuệ của giảng viên, không đảm bảo quyền tác giả.

1.3 Giải pháp: Hệ thống in ấn thông minh SSPS

SSPS (Smart Student Printing System) được thiết kế để:

- **Tích hợp tại trường:** Đáp ứng kịp thời nhu cầu in ấn của sinh viên.
- **Bảo mật:** Đảm bảo an toàn thông tin cá nhân và tuân thủ bản quyền.
- **Tối ưu hóa trải nghiệm:** Giảm thiểu thời gian chờ đợi và tăng tính tiện lợi.

1.4 Các bên liên quan và nhu cầu

Các bên liên quan:

- Sinh viên: sinh viên đang học tập tại trường.
- Nhà trường: Trường Đại học Bách khoa - Đại học Quốc gia TP.HCM.
- Nhân viên quản lý hệ thống máy in: người quản lý, hỗ trợ hệ thống vận hành.

Nhu cầu

- Sinh viên cần một hệ thống in ấn đáp ứng các nhu cầu sau:
 - + **In ấn nhanh chóng:** Yêu cầu hệ thống giúp sinh viên in tài liệu một cách nhanh gọn, không mất thời gian chờ đợi, đặc biệt là khi cần tài liệu gấp cho các buổi học, thuyết trình, hay nộp bài.
 - + **Bảo mật tài liệu:** Đảm bảo tính riêng tư và an toàn cho các tài liệu mà sinh viên tải lên để in. Chỉ người dùng có thẩm quyền mới có thể truy cập và in tài liệu, giúp tránh rò rỉ thông tin quan trọng.
 - + **Quản lý thông tin in ấn:** Hệ thống cần cung cấp chức năng theo dõi lịch sử in ấn, quản lý số trang giấy đã sử dụng, và có thể giám sát quá trình in để giúp sinh viên kiểm soát chi phí cũng như hiệu suất sử dụng tài liệu của mình.
- Nhà trường:
 - + **Tối ưu hóa việc sử dụng cơ sở hạ tầng in ấn:** Đảm bảo hệ thống in ấn hoạt động hiệu quả, tối ưu hóa việc sử dụng thiết bị và tài nguyên như giấy, mực in, từ đó giảm thiểu lãng phí và chi phí vận hành.
 - + **Theo dõi và quản lý việc in ấn:** Cần có hệ thống theo dõi chi tiết lượng in ấn, quản lý người dùng, và phân bổ hợp lý tài nguyên in ấn dựa trên nhu cầu sử dụng. Điều này giúp nhà trường dễ dàng quản lý chi phí và cải thiện hiệu suất in ấn tổng thể.
 - + **Đảm bảo vấn đề bản quyền của các tài liệu nội bộ:** Hệ thống cần kiểm soát chặt chẽ việc sao chép, phân phối các tài liệu nội bộ, nhằm đảm bảo việc tuân thủ bản quyền và giữ gìn sự bảo mật cho các tài liệu thuộc sở hữu của nhà trường.

Lợi ích

- Sinh viên:
 - + **In ấn tài liệu thuận tiện và nhanh chóng:** Sinh viên có thể dễ dàng tải tài liệu và in ấn mà không cần phải chờ đợi lâu, hỗ trợ tốt cho việc học tập và nộp bài đúng hạn.
 - + **Bảo mật các tài liệu và thông tin cá nhân:** Hệ thống đảm bảo quyền riêng tư và bảo mật đối với các tài liệu quan trọng, giúp sinh viên an tâm khi sử dụng.
 - + **Giá thành hợp lý, thanh toán thuận tiện:** Hệ thống cung cấp mức giá hợp lý và hỗ trợ thanh toán dễ dàng, phù hợp với nhu cầu của sinh viên.
- Nhân viên quản lý hệ thống máy in:
 - + **Tự động hoá quy trình in ấn, giảm bớt công việc cần làm:** Nhân viên quản lý có thể giảm thiểu công sức và thời gian dành cho việc điều hành hệ thống in ấn nhờ quy trình tự động hóa.
 - + **Quản lý máy in hiệu quả:** Hệ thống cung cấp công cụ để nhân viên dễ dàng theo dõi trạng thái máy in, tắt mở thiết bị từ xa, và xử lý các sự cố nhanh chóng.
- Nhà trường:

- + **Đảm bảo tính bảo mật của các tài liệu nội bộ:** Nhà trường có thể kiểm soát chặt chẽ việc in ấn và sao chép tài liệu, đảm bảo các tài liệu nội bộ được bảo mật tối đa.
- + **Quản lý và tối ưu hoá việc sử dụng máy in một cách hiệu quả:** Hệ thống giúp nhà trường quản lý chi phí, tài nguyên và sử dụng cơ sở hạ tầng in ấn một cách thông minh và hiệu quả, giảm lãng phí và tăng cường hiệu suất.

2 Yêu cầu với hệ thống

2.1 Yêu cầu chức năng

Đối với Sinh viên

- Đăng nhập thông qua hệ thống SSO của trường.
- Có thể tải lên (upload) tài liệu trong những định dạng cho phép để in.
- Có thể chọn máy in và thiết lập tùy chọn in (khổ giấy, 2 mặt hay 1 mặt, số lượng bản sao, ...).
- Có thể xem lịch sử in ấn của bản thân cũng như báo cáo cơ bản về việc sử dụng máy in của bản thân.
- Có thể đặt trước lịch in tài đối với một số máy in nhất định.
- Ngoài số lượng giấy được cấp sẵn vào đầu mỗi học kì, sinh viên có thể mua thêm nếu có nhu cầu và thực hiện thanh toán qua BKPay.

Đối với Nhân viên quản lý hệ thống máy in

- Đăng nhập thông qua hệ thống SSO của trường.
- Có thể quản lý máy in (thêm, kích hoạt, tạm dừng, ...).
- Có thể xem được lịch sử in ấn của sinh viên.
- Có thể quản lý sinh viên sử dụng dịch vụ (cấm dùng, tạm thời không cho phép dùng, ...).
- Cấu hình các thông số như số trang giấy mặc định, loại file cho phép, ngày hệ thống cấp phát giấy cho sinh viên.
- Hệ thống sẽ tự động tạo báo cáo hằng tháng, hằng năm; người quản lý có thể xem lại bất cứ lúc nào.

2.2 Yêu cầu phi chức năng (Non-Functional Requirements - NFR)

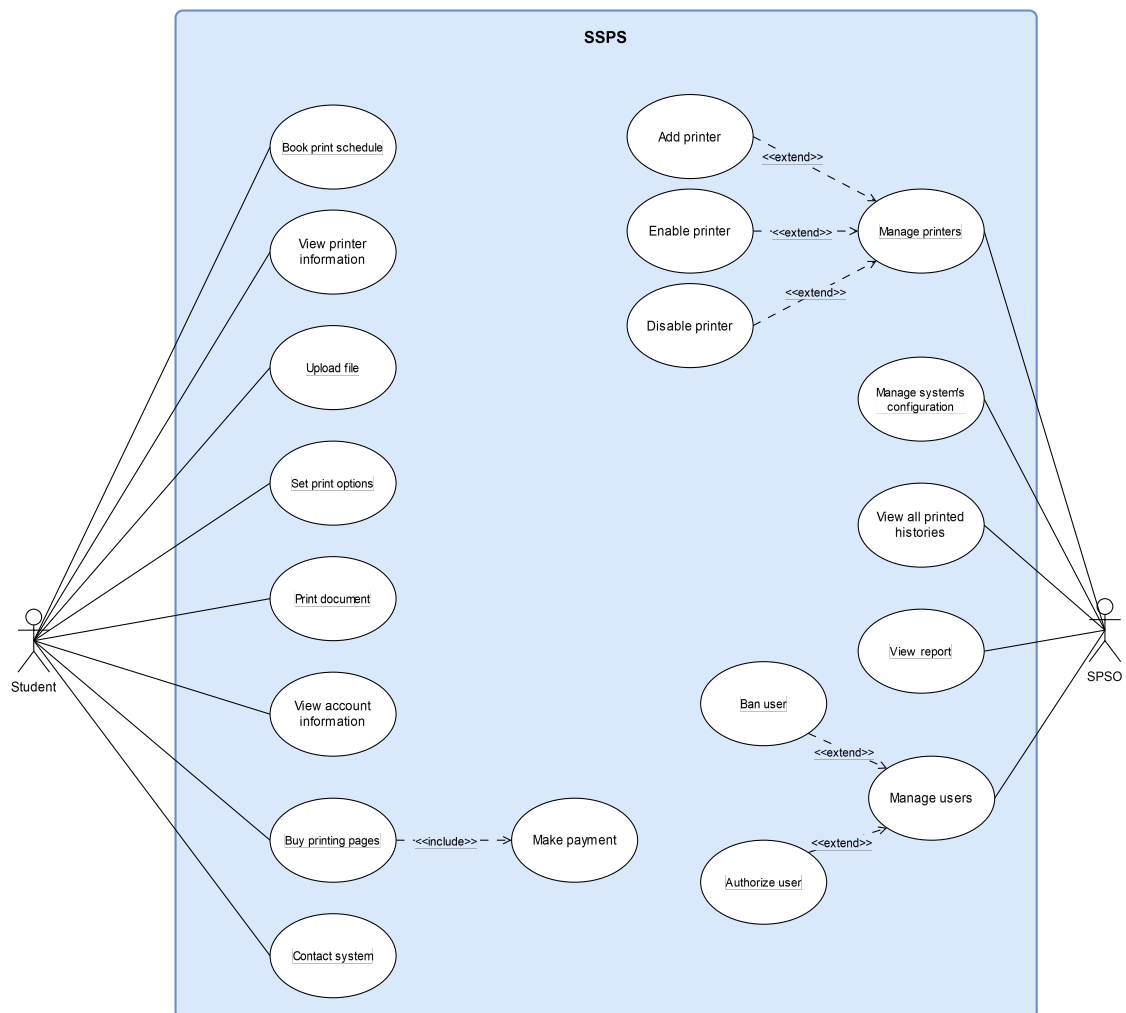
- **Thời gian phản hồi:**
 - Upload tài liệu: dưới 5 giây với tệp 20 MB.
 - Ra lệnh in: dưới 2 giây sau khi xác nhận.
- **Tính khả dụng:**



- Hệ thống duy trì hoạt động ít nhất **99.9% thời gian** mỗi tháng.
- Thời gian gián đoạn bảo trì không quá 1 giờ/lần.
- **Bảo mật:**
 - Tài liệu được in ẩn sẽ được tự động xoá khi đã in xong.
- **Khả năng mở rộng:**
 - Dễ dàng thêm mới máy in và quản lý tối đa 200 máy in trong toàn hệ thống.
- **Tiêu chuẩn kích thước nút:**
 - Trên thiết bị di động: tối thiểu 48x48 pixel.
- **Hỗ trợ đa nền tảng:**
 - Chạy mượt mà trên Chrome, Firefox, Safari.

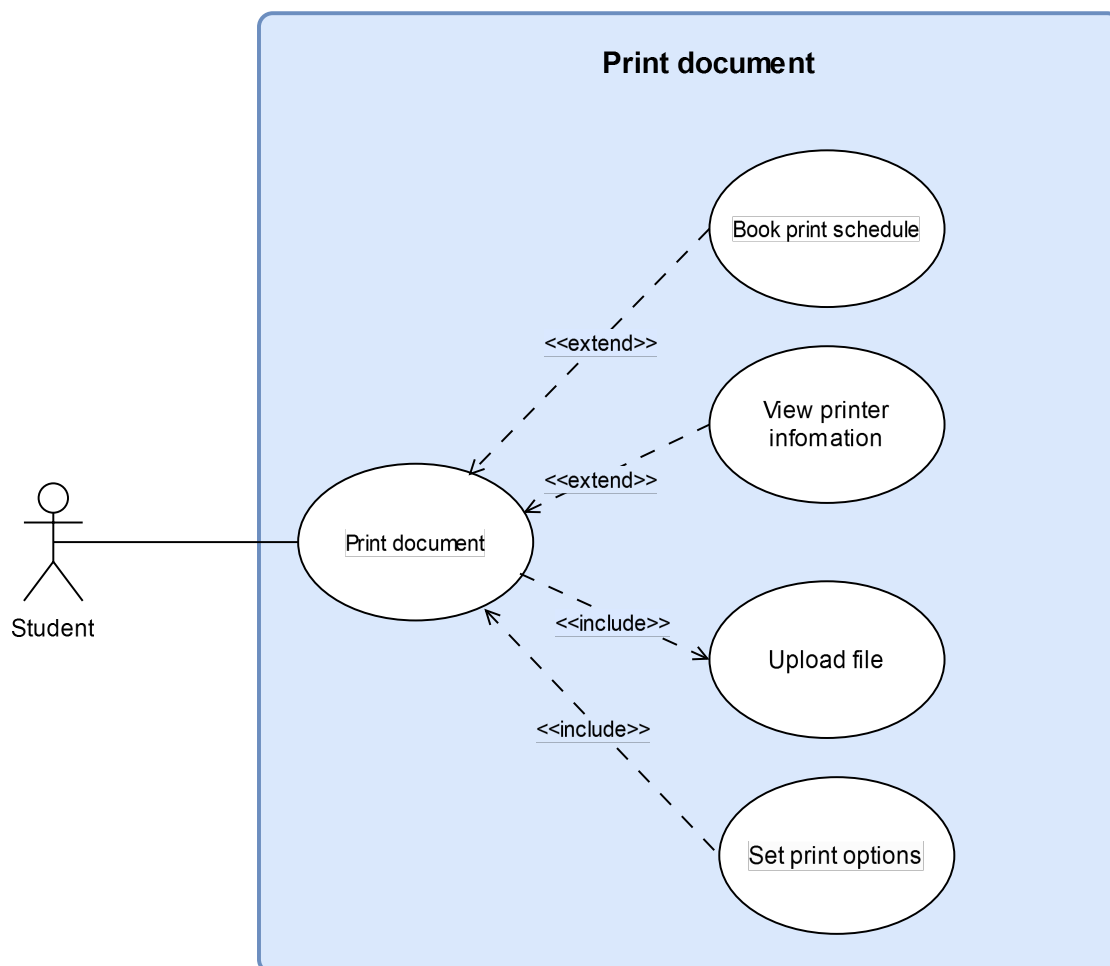
3 Use-case diagram

3.1 Use-case cho toàn bộ hệ thống



Hình 1: Use-case cho toàn bộ hệ thống

3.2 Use-case cho chức năng in



Hình 2: Use-case cho toàn bộ hệ thống

3.3 Use-case dạng bảng

Đặt lịch in

Name	Book print schedule
Actor	Sinh viên
Description	Khi sinh viên muốn không cần phải đợi máy in in tài liệu có thể chọn chức năng đặt lịch in thông qua SSPS.
Trigger	Sinh viên nhấn vào nút "Có" tại giao diện dạng pop-up với nội dung "Bạn có muốn đặt lịch in không".
Precondition	Sinh viên đã nhấn vào nút "In" và hiển thị pop-up với nội dung "Bạn có muốn đặt lịch in không".
Main Flow	<ol style="list-style-type: none">1. Hệ thống sẽ nhận yêu cầu từ sinh viên và hiển thị giao diện đặt lịch in.2. Sinh viên chọn ngày in và giờ in mong muốn.3. Sinh viên xác nhận lịch in bằng cách nhấn nút "Tiếp theo".4. Hệ thống sẽ nhận lịch in và kiểm tra lịch in có hợp lệ hay không.5. Hệ thống cập nhật giữ chỗ lịch in vào giờ được chọn.
Alternative flow	<ol style="list-style-type: none">2.1 Sinh viên có thể thêm ghi chú nếu mong muốn bằng cách điền nội dung vào ô "Ghi chú".
Postcondition	Đã đặt được lịch in cho sinh viên.
Exceptions	<ol style="list-style-type: none">3.1 Nếu sinh viên không muốn đặt lịch in thì nhấn vào nút "Hủy bỏ".4.1 Nếu hệ thống kiểm tra lịch in không hợp lệ sẽ hiển thị thông báo "Đặt lịch in thất bại, vui lòng thử lại".1.1 , 5.1 Nếu hệ thống gặp lỗi thì hệ thống sẽ trả về trang Error và thông báo "Đã có lỗi xảy ra, vui lòng thử lại sau."



Xem thông tin máy in

Name	View printer infomation
Actor	Sinh viên
Description	Khi sinh viên muốn chọn một máy in có phù hợp với nhu cầu của mình hay không thì có thể xem thông tin của máy in để biết rõ hơn.
Trigger	Sinh viên nhấn vào nút icon "i" (infomation) nằm ở góc phải trên cùng của icon máy in.
Precondition	Giao diện đang hiển thị danh sách các máy in.
Main Flow	<ol style="list-style-type: none">1. Hệ thống sẽ nhận yêu cầu từ sinh viên và truy vấn các thông tin về máy in được chọn.2. Hệ thống sẽ hiển thị thông tin máy in thông qua giao diện pop-up box.3. Sinh viên xem các thông tin máy in được chọn.
Alternative flow	<ol style="list-style-type: none">3.1 Sinh viên không muốn tiếp tục xem thông tin máy in thì nhấn vào nút "Đóng".
Postcondition	Sinh viên đã xem được các thông tin máy in mong muốn.
Exceptions	<ol style="list-style-type: none">2.1 Nếu hệ thống gặp lỗi thì hệ thống sẽ trả về trang Error và thông báo "Đã có lỗi xảy ra, vui lòng thử lại sau."



Đăng tải tài liệu

Name	Upload file
Actor	Sinh viên
Description	Sinh viên tải file tài liệu muốn in lên hệ thống.
Trigger	Sinh viên nhấn vào nút "Chọn" của máy in tại trang danh sách các máy in.
Precondition	Giao diện đang hiển thị thành công trang danh sách máy in.
Main Flow	<ol style="list-style-type: none">1. Hệ thống sẽ nhận yêu cầu và truy xuất máy in được chọn.2. Hệ thống hiển thị giao diện upload tài liệu.3. Sinh viên chọn một hoặc nhiều file cần in bằng cách nhấn vào nút "Tải lên".4. Hệ thống sẽ kiểm tra file được chọn có hợp lệ hay không.<ol style="list-style-type: none">4.1 Hệ thống kiểm tra định dạng file.4.2 Hệ thống kiểm tra dung lượng file.4.3 Hệ thống kiểm tra file có chứa virus hay không.5. Hệ thống sẽ hiển thị các file được chọn và tải lên hệ thống thành công.6. Sinh viên xác nhận các file đúng như mong muốn bằng cách nhấn vào nút "Tiếp theo" ở góc phải dưới cùng màn hình.
Alternative flow	<ol style="list-style-type: none">6.1 Nếu sinh viên muốn xóa file không muốn in thì nhấn vào icon "trash".
Postcondition	File được lưu thành công vào hệ thống và sinh viên có thể tiếp tục quá trình thiết lập in.
Exceptions	<ol style="list-style-type: none">3.1 Nếu sinh viên không muốn upload tài liệu thì nhấn vào nút "Hủy bỏ" ở góc trái dưới cùng màn hình.2.1 , 5.1 Nếu hệ thống gặp lỗi thì hệ thống sẽ trả về trang Error và thông báo "Đã có lỗi xảy ra, vui lòng thử lại sau."4.1.1 Nếu định dạng file không được hỗ trợ, hệ thống thông báo lỗi "Định dạng file không được hỗ trợ. Vui lòng tải file đúng định dạng."4.2.1 Nếu dung lượng file vượt quá giới hạn cho phép, hiển thị thông báo "Dung lượng file vượt quá giới hạn. Vui lòng chọn file nhỏ hơn."4.3.1 Nếu file chứa virus thì hiển thị thông báo "File tiềm ẩn chứa virus. Vui lòng chọn file khác."



Tùy chọn in

Name	Set print options
Actor	Sinh viên
Description	Sinh viên thiết lập tùy chọn in bao gồm các tùy chọn in như khổ giấy, số lượng in, trang in, một mặt hay hai mặt, in màu.
Trigger	Sinh viên bấm vào nút "Tiếp theo" tại trang tải tài liệu.
Precondition	Hệ thống đã kiểm tra các file tài liệu hợp lệ.
Main Flow	<ol style="list-style-type: none">1. Hệ thống sẽ nhận yêu cầu và hiển thị giao diện dạng pop-up box với các tùy chọn in mặc định.2. Sinh viên xác nhận các tùy chọn in bằng cách bấm vào nút "In"3. Hệ thống sẽ nhận các tùy chọn in và kiểm tra các tùy chọn in có hợp lệ hay không.4. Hệ thống cập nhật thành công và lưu lại áp dụng cho các lần sau.
Alternative flow	<ol style="list-style-type: none">2.1 Nếu sinh viên muốn thay đổi tùy chọn in thì nhấn vào tùy chọn in mong muốn.
Postcondition	Sinh viên đã thiết lập tùy chọn in thành công.
Exceptions	<ol style="list-style-type: none">1.1 , 4.1 Nếu hệ thống gặp lỗi thì hệ thống sẽ trả về trang Error và thông báo "Đã có lỗi xảy ra, vui lòng thử lại sau."1.2 , 2.1 Nếu sinh viên không muốn thiết lập tùy chọn in thì nhấn nút "Hủy bỏ".3.1 Nếu hệ thống kiểm tra tùy chọn in không hợp lệ thì hiển thị thông báo "Tùy chọn in thất bại, vui lòng thử lại".



In tài liệu

Name	Print document
Actor	Sinh viên
Description	Sinh viên thực hiện in tài liệu thông qua hệ thống
Trigger	Sinh viên nhấn vào nút "In"
Precondition	Sinh viên đã đăng nhập thành công vào hệ thống.
Main Flow	<ol style="list-style-type: none">1. Hệ thống sẽ nhận yêu cầu và hiển thị giao diện pop-up với nội dung "Bạn có muốn đặt lịch in không".2. Sinh viên đặt lịch in bằng cách nhấn nút "Có".3. Hệ thống nhận yêu cầu và hiển thị giao diện đặt lịch in.4. Sinh viên chọn lịch in và xác nhận lịch in bằng cách nhấn nút "Tiếp theo".5. Hệ thống sẽ nhận yêu cầu và kiểm tra lịch in có hợp lệ hay không.6. Hệ thống truy vấn và hiển thị các danh sách các máy in theo yêu cầu.7. Sinh viên chọn máy in bằng cách nhấn nút "Chọn" bên dưới icon máy in.8. Hệ thống sẽ nhận yêu cầu và hiển thị giao diện upload tài liệu.9. Sinh viên chọn một hoặc nhiều tài liệu bằng cách nhấn nút "Tải lên".10. Hệ thống sẽ kiểm tra các file và hiển thị các file được chọn.11. Sinh viên xác nhận file được chọn bằng cách nhấn nút "Tiếp theo".12. Hệ thống sẽ hiển thị giao diện pop-up box với các tùy chọn in mặc định.13. Sinh viên xác nhận tùy chọn in bằng cách xác nhận "In".14. Hệ thống kiểm tra tài khoản có số dư đủ để in tài liệu hay không.15. Hệ thống cập nhật các thông tin và thông báo thành công.
Alternative flow	<ol style="list-style-type: none">2.1 Nếu sinh viên không muốn đặt lịch in thì nhấn nút "Không".7.1 Nếu sinh viên muốn xem thông tin máy in thì nhấn vào nút icon "i" (infomation) nằm ở góc phải trên cùng của icon máy in.9.1 Nếu sinh viên muốn xóa file không muốn in thì nhấn vào icon "trash".13.1 Nếu sinh viên muốn thay đổi tùy chọn in thì nhấn vào tùy chọn in mong muốn.
Postcondition	Tài liệu đã được gửi in và trở về trang chủ nhận thông báo "In thành công".

In tài liệu

Exceptions	<p>5.1 Nếu hệ thống kiểm tra lịch in không hợp lệ thì hiển thị thông báo "Đặt lịch in thất bại, vui lòng chọn lại".</p> <p>7.1 , 9.1 Sinh viên không muốn in thì nhấn vào nút "Hủy bỏ" ở góc trái dưới cùng màn hình.</p> <p>3.1 , 6.1, 8.1, 12.1 Nếu hệ thống gặp lỗi thì hệ thống sẽ trả về trang Error và thông báo "Đã có lỗi xảy ra, vui lòng thử lại sau.".</p> <p>14.1 Nếu tài khoản có số dư không đủ thì hiển thị trang mua giấy in và thông báo "Tài khoản có số dư không đủ, vui lòng mua giấy in.".</p>
------------	---

4 Activity diagram

4.1 Mô tả

Đây là activity diagram cho tính năng chính của hệ thống: tính năng in tài liệu của sinh viên. Các hành động sẽ chỉ có thể được thực hiện khi người dùng đăng nhập vào hệ thống để xác minh danh tính.

Đầu tiên, sinh viên có hoạt động đăng nhập vào hệ thống, sau đó HCMUT_SSO có hoạt động kiểm tra tính đúng đắn của tài khoản, mật khẩu của sinh viên, nếu không hợp lệ (nghĩa là tài khoản hoặc mật khẩu không đúng) thì HCMUT_SSO sẽ thực hiện hoạt động đưa ra thông báo đăng nhập không hợp lệ và kết thúc vòng đời hoạt động, sinh viên có thể thoát ứng dụng hoặc thực hiện đăng nhập lại để có thể sử dụng các tiện ích của hệ thống. Nếu HCMUT_SSO kiểm tra thông tin tài khoản đăng nhập hợp lệ.

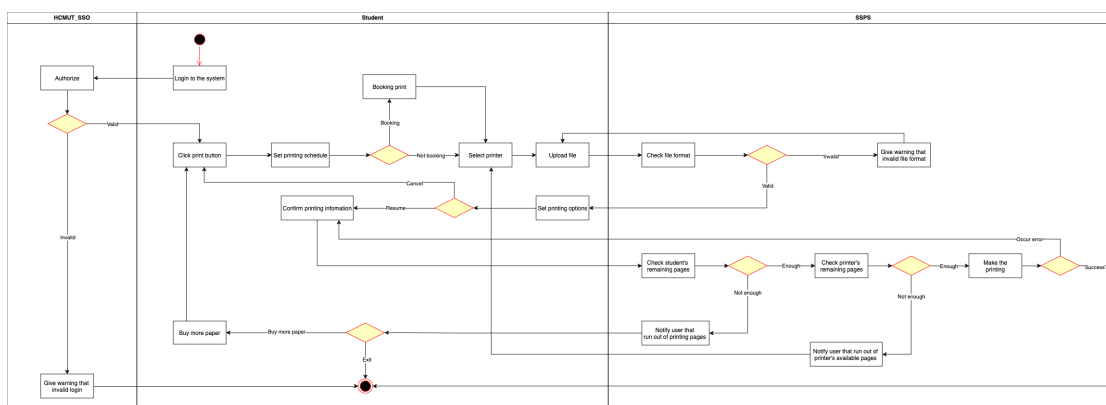
Để thực hiện tính năng in, sinh viên có hành động bấm vào nút "In" trên giao diện hệ thống, tiếp đó sinh viên sẽ chọn có muốn thực hiện đặt lịch in lấy sau hay không. Nếu có thì sinh viên sẽ chọn chính xác lịch in lấy sau phù hợp với nhu cầu của mình, còn nếu không thì sẽ đi thẳng đến hoạt động tiếp theo là chọn máy in. Các máy in được hiện ra để có thể chọn sẽ căn cứ theo lựa chọn trước đó của sinh viên về việc đặt lịch in (ngày giờ sinh viên chọn là còn trống nếu là máy hỗ trợ đặt lịch, còn nếu chọn không đặt lịch thì sẽ chỉ hiển thị máy in không hỗ trợ đặt lịch) các máy in khác sẽ bị mờ đi). Tiếp đến là hành động tải lên tệp tin mà sinh viên cần in. Sau khi tệp tin đã được tải lên bởi sinh viên, hệ thống sẽ có hành động kiểm tra định dạng có hợp lệ hay không (hệ thống có thể hỗ trợ một số loại định dạng file nhất định để tránh gây lỗi trong quá trình in (pdf, doc,...)). Nếu định dạng file không phù hợp với các định dạng mà hệ thống hỗ trợ thì hệ thống sẽ đưa ra thông báo và quay về để sinh viên thực hiện lại hành động tải lên tệp tin khác. Còn nếu định dạng hợp lệ thì sẽ đi đến trang để sinh viên thực hiện chỉnh sửa các tùy chọn, cấu hình in (khổ giấy, số bản in, có in màu không...).

Sau khi các thao tác cấu hình in hoàn tất, ở bước này nếu không còn muốn in nữa sinh viên có thể bấm Hủy để quay lại trang chính, còn nếu tiếp tục, sinh viên có hành động tiếp theo là bấm nút "Xác nhận" để lưu lại cài đặt in để có thể backup nếu xảy ra lỗi trong quá trình in. Sau đó, hệ thống sẽ thực hiện hành động kiểm tra số dư trang in của sinh viên, nếu không đủ hệ thống sẽ hiện thông báo, lúc này sinh viên có thể thực hiện hành động mua thêm trang in và quay lại in nếu còn có nhu cầu hoặc thoát nếu không đủ kinh phí mua thêm trang in hoặc không còn nhu cầu in nữa. Nếu số dư trang của sinh viên đủ thì hệ thống tiếp tục kiểm tra đến số dư giấy còn lại của loại khổ giấy sinh viên chọn có còn đủ trong máy in hiện tại hay không. Nếu không thì hệ thống sẽ thông báo sinh viên chọn máy in khác và đưa sinh viên đến trang chọn máy in. Nếu số dư giấy trong máy in hiện tại còn đủ để đáp ứng số lượng trang in đối với khổ giấy mà sinh viên yêu cầu thì hệ thống sẽ thực hiện hành động gọi đến máy in để thực hiện thao tác in tài liệu. Nếu xảy ra lỗi trong quá trình in thì sẽ quay lại trang Thiết lập cấu hình in để sinh viên thực hiện xác nhận lại (ở bước này, các dữ liệu cấu hình in mà sinh viên đã thiết

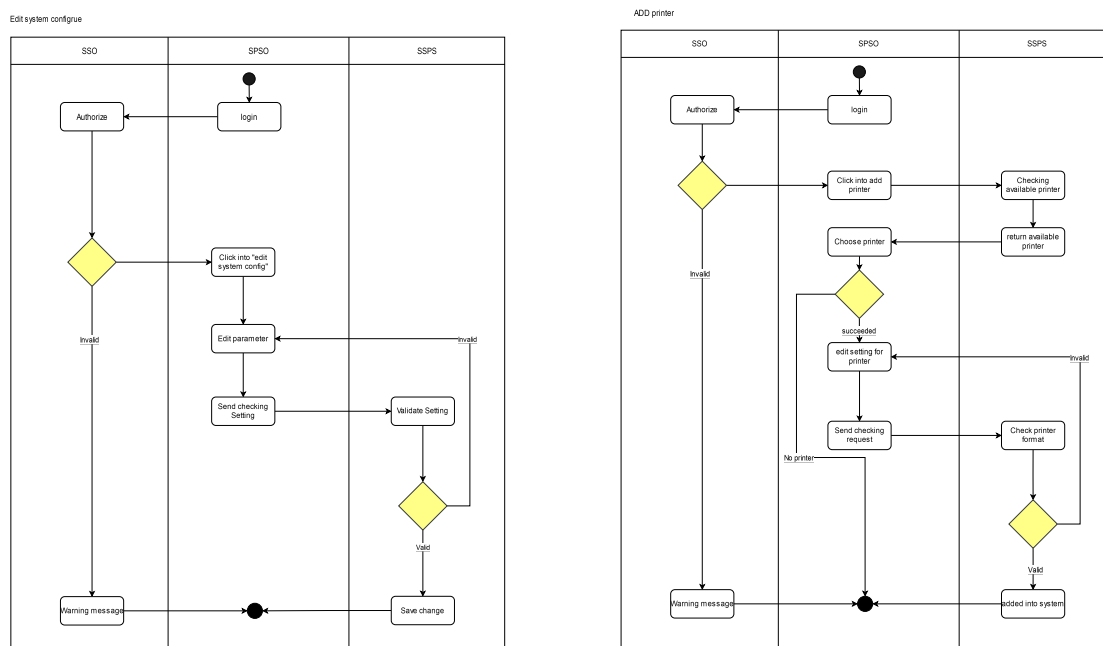
lập trước đó sẽ được backup), còn nếu không có lỗi xảy ra hệ thống sẽ kết thúc.

4.2 Diagram

Có thể xem tại link: [GitHub](#)



Hình 3: Activity Diagram cho chức năng in



Hình 4: Activity Diagram cho chức năng thêm máy in và tùy chọn hệ thống

5 Sequence diagram

5.1 Mô tả

Trình tự thực hiện các chức năng

- Người dùng nhấn vào nút “Bắt đầu in” trên trang chủ. Trước khi chuyển sang trang khác, hệ thống sẽ gọi đến hàm ở Controller để kiểm tra xem người dùng đã đăng nhập hay chưa. Controller tiếp tục gọi hàm ở tầng Model truy cập vào database để tìm “login token”. Nếu tìm thấy “login token” khớp với “login token” lưu ở session của người dùng thì chuyển đến trang danh sách máy in, ngược lại chuyển đến trang đăng nhập và yêu cầu người dùng đăng nhập.
- Người dùng chọn máy in mong muốn bằng cách nhấn vào nút “Chọn” của máy in đó. Trước khi chuyển sang trang khác, hệ thống sẽ gọi đến hàm ở Controller để kiểm tra xem máy in có đang hoạt động hay không. Nếu Controller kiểm tra máy in được chọn đang hoạt động thì chuyển đến trang thông tin của máy in đó, ngược lại yêu cầu người dùng chọn lại máy in khác.
- Người dùng nhấn vào nút “Tải lên” và chọn file cần in để tải lên hệ thống. Trước khi chuyển sang trang khác, hệ thống sẽ gọi đến hàm ở Controller để kiểm tra định dạng file có phù hợp hay không. Nếu Controller kiểm tra định dạng file phù hợp thì hiển thị file mà người dùng đã tải lên, ngược lại không tải file lên hệ thống và báo lỗi, yêu cầu người dùng chọn file đúng định dạng.
- Người dùng nhấn vào nút “Xác nhận” để xác nhận yêu cầu in file. Trước khi chuyển sang trang khác, hệ thống sẽ gọi đến hàm ở Controller để kiểm tra số trang hiện có trong tài khoản có đủ thực hiện thao tác in hay không. Nếu số trang giấy có trong tài khoản đủ thực hiện thì Controller tiếp tục đến tầng Model gọi hàm lưu lịch sử in vào database, trả về thông báo cho người dùng “Tài liệu được gửi in thành công”. Nếu số trang giấy có trong tài khoản không đủ thực hiện thì hiển thị thông báo “Số giấy còn lại trong tài khoản của bạn không đủ”.

Trình tự thực hiện chức năng thiết lập tùy chọn in

- Tại trang thiết lập tùy chọn in, hệ thống sẽ gọi đến hàm ở Controller để gửi yêu cầu lấy dữ liệu cài đặt mặc định của tài liệu in. Controller gọi hàm ở tầng Model truy cập vào database để lấy dữ liệu trả về cho người dùng.
- Nếu người dùng muốn thay đổi kiểu in thì chọn vào những thuộc tính muốn thay đổi rồi bấm nút “Xác nhận”. Trước khi chuyển sang trang khác, hệ thống sẽ gọi đến hàm ở Controller để yêu cầu thay đổi tùy chọn in. Controller tiếp tục gọi hàm ở tầng Model thông qua hàm để lưu lại tùy chọn in mới. Hệ thống hiển thị thông báo “Thiết lập tùy chọn in thành công”

Trình tự thực hiện chức năng đặt lịch in

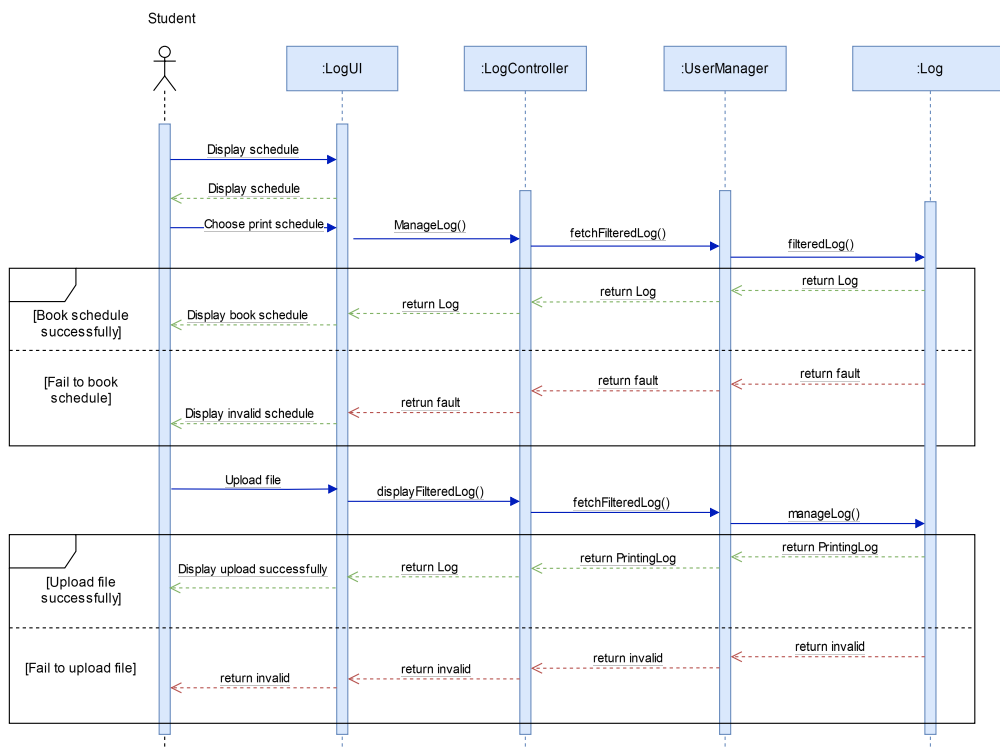
- Khi nhấn vào “Bắt đầu in” sẽ có hộp thoại popup hỏi người dùng có muốn sử dụng tính năng này hay không. Nếu có hệ thống sẽ gọi đến hàm ở Controller để gửi yêu cầu lấy dữ liệu lịch in của hệ thống. Controller tiếp tục gọi hàm ở tầng Model truy cập vào database để tìm dữ liệu lịch in. Database sẽ trả dữ liệu về và hiển thị các lịch in của ngày hôm nay.



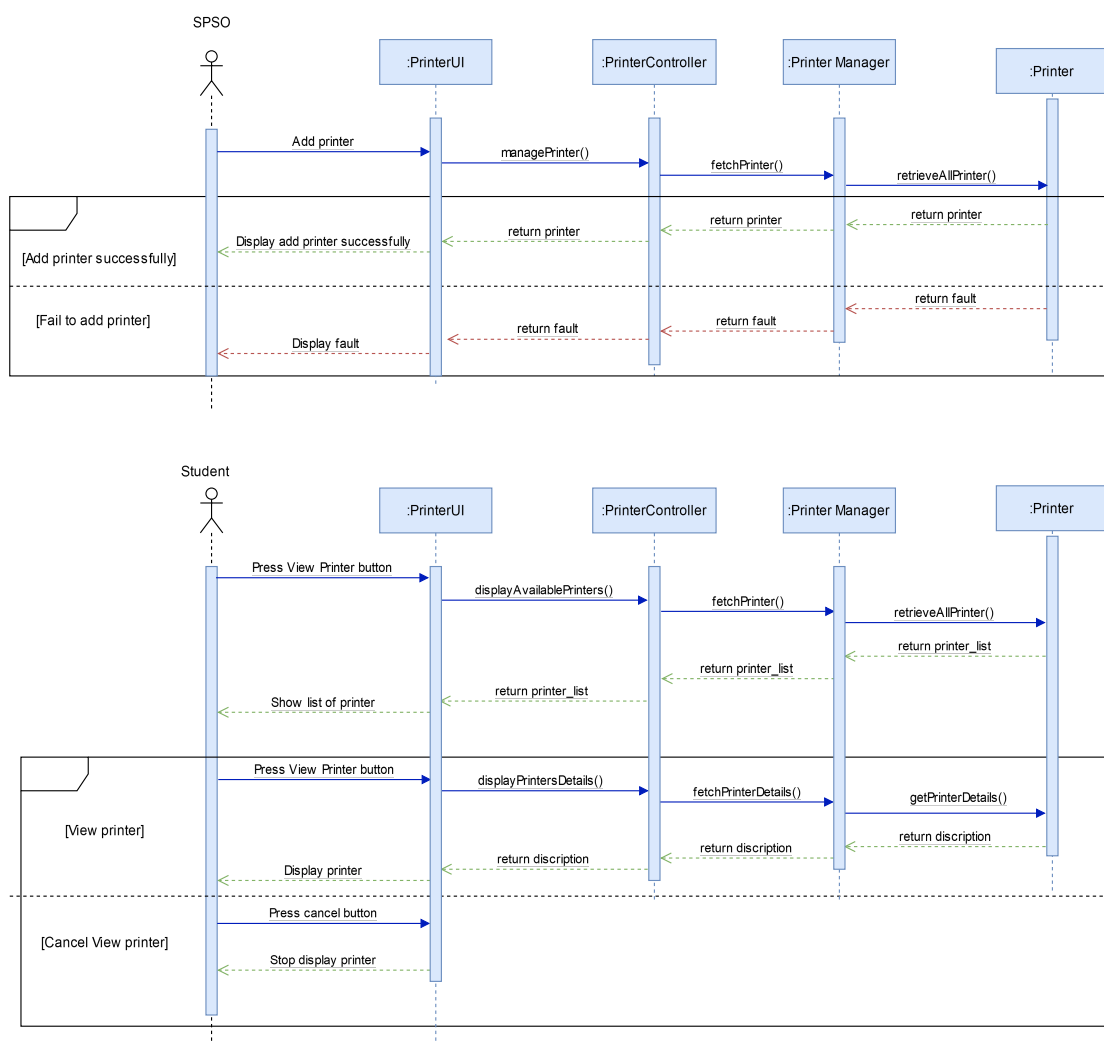
- Người dùng tiếp tục chọn lịch in mong muốn bằng cách chọn khung giờ còn trống. Trước khi chuyển sang trang khác, hệ thống sẽ gọi đến hàm ở Controller để gửi yêu cầu kiểm tra lịch in đó vẫn còn đang trống hay không. Nếu còn thì Controller tiếp tục đến tầng Model gọi hàm yêu cầu cập nhật lại slot của lịch in đó và thông báo đến người dùng “Đặt lịch in thành công”, ngược lại thì hiển thị thông báo “Đặt lịch in thất bại, vui lòng chọn lịch in khác”.

5.2 Diagram

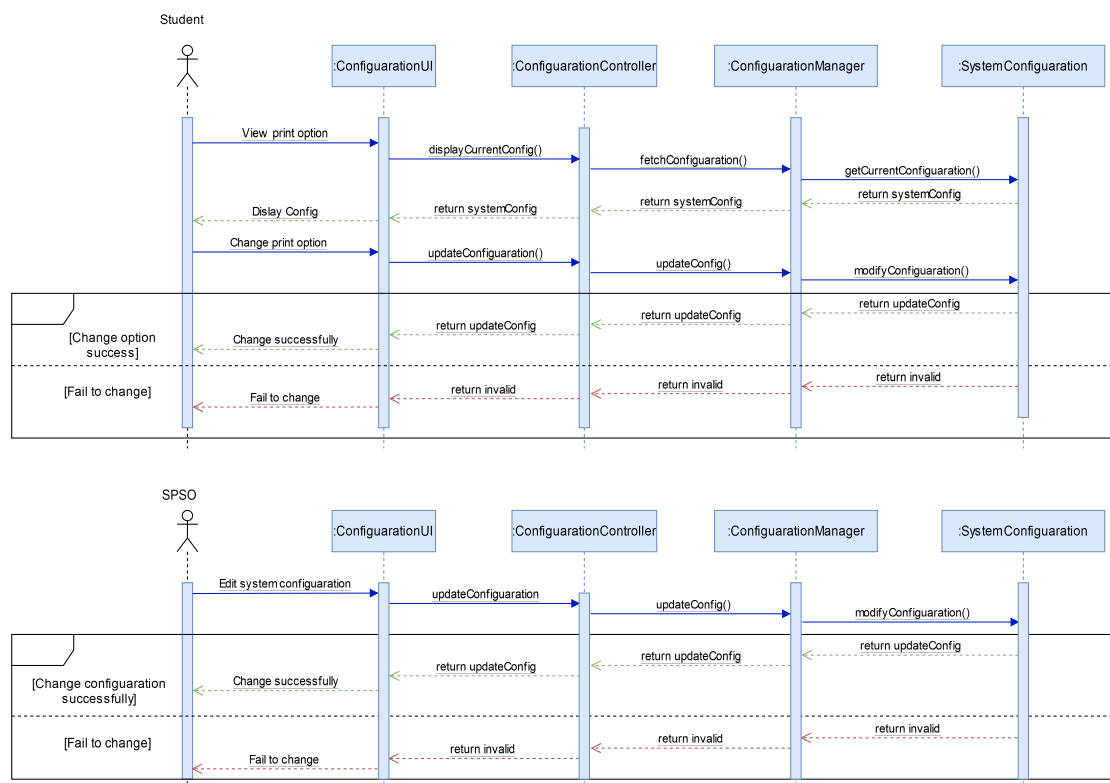
Có thể xem tại link: [GitHub](#)



Hình 5: Sequence Diagram - 1



Hình 6: Sequence Diagram - 2



Hình 7: Sequence Diagram - 2

6 Class Diagram

6.1 Các thành phần chính của Class Diagram

Class Diagram bao gồm bốn loại chính: **Entity**, **UI**, **Controller**, và **Business**.

6.1.1 Entity Class

Các lớp thực thể chính trong hệ thống, chứa dữ liệu và các phương thức liên quan.

- **Person:**

- **Thuộc tính:**

- * ID: String
 - * email: String
 - * phoneNumber: String
 - * password: String
 - * username: String
 - * name: String
 - * is_active: Boolean

- **Phương thức:**

- * login(username: String, password: String): Boolean

- **Student** (kế thừa từ Person):

- **Thuộc tính:**

- * paperBalance: Int
 - * faculty: String
 - * defaultSide: Boolean
 - * defaultSize: String
 - * defaultColor: Boolean

- **Phương thức:**

- * displayDetail(): void
 - * uploadDoc(file: File): Boolean
 - * selectPrinter(printerID: String): Printer
 - * sendPrintRequest(printerID: String): Printer
 - * buyPaper(amount: Int): PaperLog

- **SPSO** (kế thừa từ Person):

- **Phương thức:**

- * managePrinter(printerID: Printer): Boolean
 - * updateConfiguration(config: List<String>): Boolean

- **Log:**

- **Thuộc tính:**

- * logID: String

- * time: Datetime

- * status: String

- * message: String

- **Phương thức:**

- * showLog(): void

- * showStudentLog(studentID: String): void

- * updateLog(logID: String): void

- **PrintingLog** (kế thừa từ Log):

- **Thuộc tính:**

- * color: Boolean

- * is2Side: Boolean

- * size: String

- * isScheduled: Boolean

- * scheduleTime: Datetime

- * documentPages: Int

- * numberOfCopies: Int

- * printerNumber: String

- * documentName: String

- **PaperLog** (kế thừa từ Log):

- **Thuộc tính:**

- * amount: Int

- * totalPrice: Int

- * paymentMethod: String

- **Printer:**

- **Thuộc tính:**

- * printerID: String

- * model: String

- * is_scheduled: Boolean

- * description: String

- * brand: String

- * paperLeft: List<Int>

- * building: String

- * room: String

- * campus: String

- * is_enabled: Boolean

- **Phương thức:**

- * enable(): Boolean

- * disable(): Boolean

- * print(): Boolean



* `verifyPrintingRequirements(): Boolean`

- **SystemConfiguration:**

- **Thuộc tính:**

- * `ID: String`
 - * `defaultAmount: Int`
 - * `pageResetDate: Datetime`
 - * `permittedFile: List<String>`
 - * `permittedSize: List<String>`
 - * `permittedPaymentMethod: List<String>`
 - * `unitPrice: Int`

- **Phương thức:**

- * `displaySystemConfig(): void`
 - * `updateSystemConfig(config: List<String>): void`

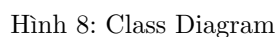
6.1.2 UI Class

UI Class đại diện cho các lớp xử lý giao diện người dùng, chịu trách nhiệm thu thập dữ liệu từ người dùng và hiển thị kết quả. Các lớp chính bao gồm:

- **PrinterUI**
- **ConfigurationUI**
- **StudentUI**
- **LogUI**
- **SPSOUI**
- **UserUI**

6.2 Diagram

Có thể xem tại link: [GitHub](#)



7.1 Mô tả

- Sau khi đăng nhập sinh viên có thể chọn "BẮT ĐẦU IN" để tiến hành in.
- Khi này sẽ hiện ra 2 lựa chọn về việc có muốn đặt lịch in không:
 - Có: dẫn tới trang chọn ngày và giờ (có thể kèm theo ghi chú). Sau đó sẽ dẫn tới trang chọn máy in.
 - Không: dẫn thẳng tới trang chọn máy in.
- Tiến hành ấn "CHỌN" ở máy in có vị trí phù hợp với nhu cầu.
- Sau khi chọn xong sẽ được dẫn tới trang Upload tài liệu, sau khi upload hệ thống sẽ kiểm tra tính hợp lệ của file.
 - Nếu hợp lệ: màn hình sẽ chuyển thành danh sách các file đã upload, tại đây sinh viên có thể xem lại các file muốn in nếu upload nhầm có thể xóa đi.
 - Nếu không hợp lệ: màn hình sẽ hiện thông báo (chữ đỏ góc phải dưới khung upload) để thông báo cho sinh viên.
- Sau khi đã upload ấn "TIẾP THEO" sẽ hiện ra hộp thoại "CÀI ĐẶT IN" tại đây sinh viên có thể tùy chọn cho bản in của mình. Có 5 tùy chọn: Khổ giấy, số lượng bản in, trang, một mặt hay hai, in màu hay không. Sau khi hoàn thành sinh viên nhấn "IN" và tài liệu sẽ được in.

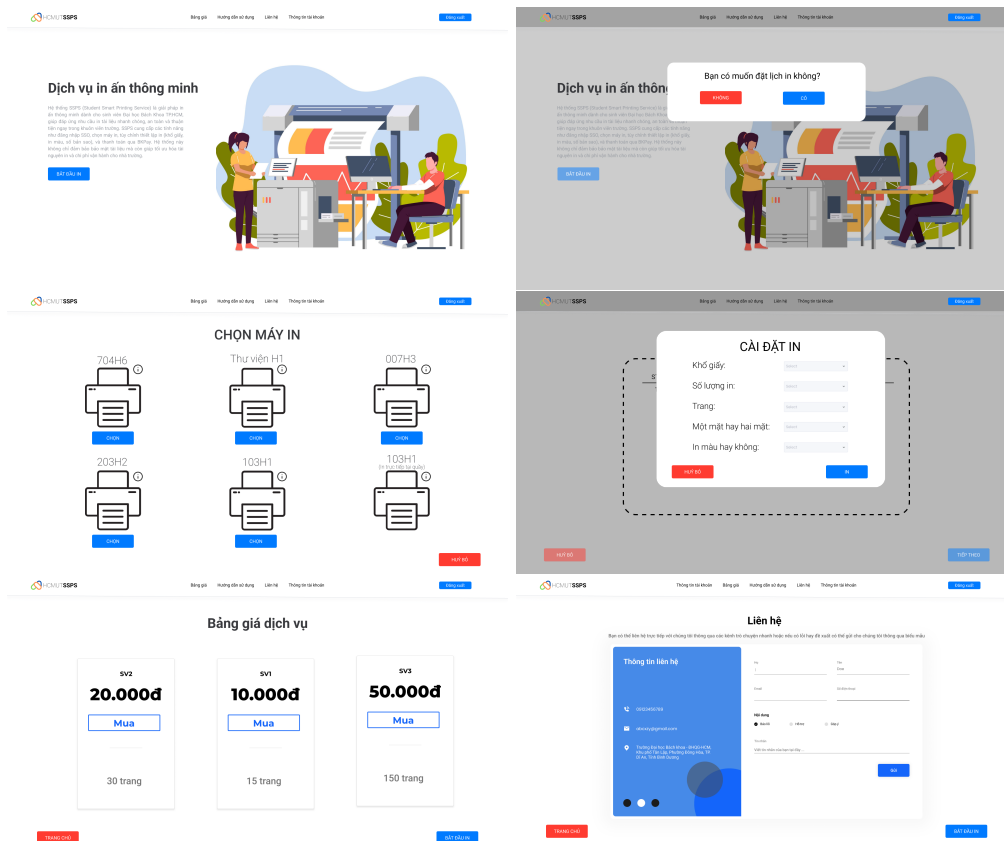


7.2 MVP

MVP này được vẽ bằng Figma, bản demo là chức năng "Present" của Figma không phải web đã được code. MVP có thể sẽ khác đôi chút với phiên bản web hoàn chỉnh.

- Có thể xem phiên bản hình ảnh tại: Có thể xem tại link: [GitHub](#)
- Có thể xem video demo tại: [GitHub - README.md](#)

Một số hình ảnh của MVP.



Hình 9: MVP

8 Thiết kế kiến trúc phân lớp cho hệ thống SSPS

8.1 Lý do chọn kiến trúc

Để tìm mô hình kiến trúc phù hợp với hệ thống thì nhóm sẽ xem xét các đặc điểm của hệ thống:

- Hệ thống cần được tổ chức một cách rõ ràng giữa giao diện người dùng, các thành phần dữ liệu và logic xử lý nghiệp vụ. Điều này giúp mã nguồn dễ quản lý và bảo trì hơn.
- Hệ thống cần khả năng tái sử dụng các mã nguồn nhằm phát triển nhanh hơn và tăng hiệu quả.
- Hệ thống cần hỗ trợ cho việc phát triển đồng thời bởi nhiều thành viên, mỗi thành viên có thể làm việc trên các phần khác nhau mà không gây ảnh hưởng lẫn nhau. Từ đó giúp dễ dàng kiểm thử từng phần độc lập và khắc phục nhanh chóng.
- Hệ thống yêu cầu tính bảo mật cao. Do đó, việc tổ chức từng phần của hệ thống giúp cô lập quyền dữ liệu và chức năng.

Với các yêu cầu cơ bản như trên, nhóm đề xuất sử dụng kiến trúc thiết kế theo kiến trúc phân lớp (Layered Architecture). Kiến trúc phân lớp là kiến trúc phổ biến trong đó hệ thống tổ chức thành các lớp, mỗi lớp trong kiến trúc này có chức năng riêng và tương tác với lớp ngay bên trên nó.

8.2 Mô tả kiến trúc

Mô hình kiến trúc phân lớp (Layered Architecture) mà nhóm thiết kế là sự kết hợp giữa kiến trúc 3 tầng (Three-tier architecture) và mẫu kiến trúc phần mềm MVC (MVC design pattern), kết hợp với 1 tầng bảo mật để cung cấp xác thực và phân quyền rõ ràng cho người dùng. Kiến trúc này bao gồm có 7 lớp chính: View Layer, Authentication Layer, Authorization Layer, Presentation Layer, Business Logic Layer, Data Access Layer và Database Layer. Trong đó, kiến trúc 3 tầng (Three-tier architecture) được thể hiện ở 3 lớp: Presentation Layer, Business Logic Layer và Data Access Layer. Phần bảo mật (Security) của hệ thống được chia thành 2 lớp: Authentication Layer và Authorization Layer. Ngoài ra, MVC design pattern (Model - View - Controller) được sử dụng thông qua sự liên kết giữa các lớp: View Layer (View), Presentation Layer (Controller) và Model để tương tác (request - response) dữ liệu giữa các tầng của hệ thống.

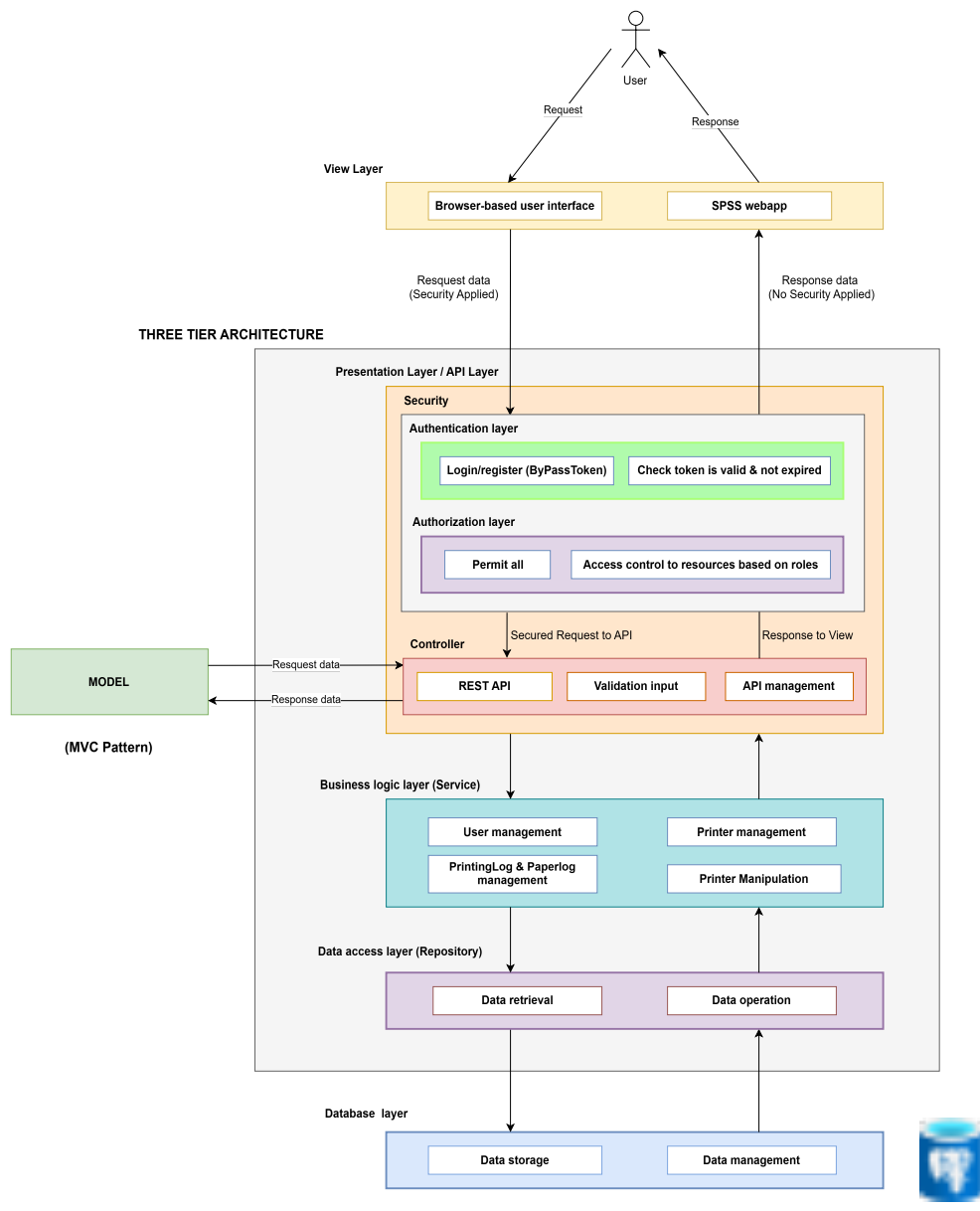
- Khi người dùng sử dụng hệ thống thì lớp thứ nhất là View Layer sẽ hiển thị giao diện để người dùng tương tác với hệ thống và hiển thị thông tin cho người dùng tại giao diện này. Thông qua giao diện trực quan, người dùng có thể dễ dàng thực hiện yêu cầu (request) đến hệ thống và có thể nhận dữ liệu trả ra (response) để đáp ứng yêu cầu người dùng.
- Request của người dùng sẽ được nhận bởi lớp Presentation layer (hay API layer), lớp này có tích hợp Security và tầng Controller để bắt và xử lý RESTful API
 - Khi thực hiện một thao tác request đến hệ thống, người dùng phải đi qua phần bảo mật (Security) của hệ thống để đảm bảo xác thực và có thể truy cập được đúng loại tài nguyên mà mình được phân quyền, tránh người dùng có thể xem được những thông tin nhạy cảm ảnh hưởng đến tính bảo mật của hệ thống. Tầng security này chia thành 2 lớp:

- * Authentication Layer: Lớp này thực hiện xác thực người dùng trước khi cho phép truy cập vào các tài nguyên của hệ thống. Việc xác thực này bao gồm đăng kí tài khoản và đăng nhập để sử dụng hệ thống. Khi đăng kí tài khoản thành công, hệ thống sẽ thực hiện thêm một người dùng vào cơ sở dữ liệu và lưu lại tài khoản, mật khẩu kèm với vai trò và một số thông tin khác của người dùng. Khi đăng nhập thành công (đúng tài khoản và mật khẩu), người dùng sẽ được cấp một token (JSON Web Token) tương ứng với người dùng đó, token sẽ được gửi kèm với các yêu cầu của người dùng ở trong phiên đăng nhập để xác thực yêu cầu một cách liên tục. Token cũng sẽ có thời hạn, khi quá hạn, người dùng cần đăng nhập lại để có thể tiếp tục thực hiện các yêu cầu.
 - * Authorization Layer: Lớp này sẽ thực hiện việc phân quyền tài nguyên hệ thống cho người dùng. Một người dùng với vai trò cụ thể (sinh viên hoặc SPSO) sẽ có quyền truy cập một số tài nguyên cụ thể mà không phải toàn bộ hệ thống để tránh có thể xem được thông tin nhạy cảm của hệ thống hoặc của người dùng khác, đó là lí do lớp này được sử dụng.
- Sau khi qua được lớp bảo mật của hệ thống, Controller sẽ bắt các RESTful API và thực hiện thao tác với Model (trong MVC design pattern) để thực hiện các thao tác bên trong hệ thống. Lớp này sau khi thực hiện kiểm tra tính đúng đắn của dữ liệu đầu vào thông qua (validation) sẽ gọi đến các hàm ở tầng bên dưới để xử lí logic. Ở chiều ngược lại, khi nhận dữ liệu trả ra từ các tầng bên dưới thì sẽ thông qua Model (trong MVC design pattern) để trả trực tiếp ra View (không thông qua Security)
- Business Logic Layer (Service) là lớp chịu trách nhiệm triển khai các chức năng cốt lõi và logic nghiệp vụ của hệ thống. Lớp này nhận yêu cầu từ lớp Presentation, xử lý nó và gửi yêu cầu tương ứng đến lớp Data Access. Ở chiều ngược lại, chiều trả dữ liệu về, lớp này cũng sẽ thực hiện việc xử lí và chuẩn hóa dữ liệu server trả ra theo định dạng client mong muốn và gửi lên cho lớp Presentation.
 - Data access layer (Repository) là lớp đảm nhiệm việc nhận yêu cầu từ lớp Business logic và gửi yêu cầu đến lớp Database để thực hiện các thao tác liên quan đến dữ liệu.
 - Database layer là lớp thấp nhất trong kiến trúc phân lớp. Đây là lớp thực hiện việc lưu trữ, quản lý và thao tác dữ liệu theo cách được định nghĩa từ lớp Data Access.

8.3 Diagram

Có thể xem tại link: [GitHub](#)

**The Architecture of the SPSS system:
Layered architecture with Three tier architecture & MVC design pattern**



Hình 10: Layer Architecture

8.4 View Layer

Trang chủ

- Giới thiệu ngắn gọn về hệ thống: Cung cấp một đoạn giới thiệu ngắn về hệ thống SSPS.



- Nút In: Nút này sẽ hiện lên bảng chọn đặt lịch in hay in ngay lập tức, nếu chưa đăng nhập sẽ đưa đến trang đăng nhập.

Đăng nhập Thông qua hệ thống SSO của trường

Trang đặt in

- Chọn lịch: Nếu chọn đặt lịch giao diện cho phép người dùng chọn ngày, giờ và ghi chú in.
- Chọn máy in: Chọn máy in ở vị trí phù hợp với nhu cầu.
- Upload tài liệu: Hiện thị giao diện upload tài liệu bao gồm danh sách các loại file được chấp nhận
- Upload tài liệu (sau khi upload): Hiện thông tin về file vừa upload nếu file hợp lệ, nếu không hợp lệ hiển thị lỗi.
- Cài đặt tùy chọn in:
 - Chế độ in: Lựa chọn in đen trắng hoặc in màu.
 - Kích thước giấy: Cho phép người dùng chọn các kích thước phổ biến như A4, A5,
 - Kiểu in: Lựa chọn in một mặt hoặc hai mặt.
 - Số bản in: Nhập số lượng bản in mong muốn.
 - Chọn trang in: Chẵn, lẻ, toàn bộ, cụ thể trang nào đó hoặc từ trang x tới trang y

Trang thanh toán

- Nếu lượng giấy còn đủ: Tự động trừ lượng giấy còn lại trong tài khoản.
- Nếu lượng giấy không đủ: Chuyển sang trang mua giấy.

Trang mua giấy

- Mua thêm giấy: Có thể chủ động mua thêm giấy bằng chọn các gói phù hợp sau đó thanh toán.
- Cấp giấy tự động: Sinh viên sẽ được cấp một lượng giấy nhất định sau một khoản thời gian và không cần thanh toán.

Thanh toán Tiền sẽ được cộng vào nợ ở BKPay và người dùng sẽ tiến hành ở BKPay.

Trang hỗ trợ khách hàng

- Hỗ trợ trực tiếp: Có thể lại trực tiếp các điểm in để được hỗ trợ.
- Live Chat: Hỗ trợ live chat (thông qua fanpage facebook) để người dùng có thể hỏi đáp trực tiếp với đội ngũ hỗ trợ.
- Thông tin liên hệ: Bao gồm số điện thoại, email, và địa chỉ của các điểm in.

Dashboard cho người quản trị Đây là một phần riêng cho quản trị viên hệ thống, giúp quản lý các đơn hàng, người dùng và các thiết lập hệ thống:

- Quản lý khách hàng: Theo dõi tài khoản và các hoạt động của khách hàng.
- Báo cáo: Thống kê về số lượng giấy in, giấy mua, doanh thu theo từng ngày, tháng.

8.5 API Layer

8.5.1 Khái niệm API

API (Application Programming Interface) là một tập hợp các câu lệnh, hàm, giao thức, đối tượng, ... cho phép các ứng dụng có chức năng riêng biệt giao tiếp, trao đổi dữ liệu với nhau. Có thể hình dung API như một hợp đồng dịch vụ giữa 2 ứng dụng. Hợp đồng này xác định cách thức hai ứng dụng giao tiếp với nhau thông qua các yêu cầu (request) và phản hồi (response). Kiến trúc API thường được giải thích dưới dạng client và server. Ứng dụng gửi yêu cầu được gọi là client, còn ứng dụng gửi phản hồi được gọi là server. Có nhiều loại API như SOAP API, RPC API, Websocket API, REST API.

8.5.2 Rest API

Nhóm sẽ chọn sử dụng REST API là loại API phổ biến và linh hoạt nhất trên web hiện nay. Client gửi yêu cầu đến server dưới dạng dữ liệu. Server dùng dữ liệu đầu vào từ client này để bắt đầu các hàm nội bộ và trả lại dữ liệu đầu ra cho client. REST bao gồm các phương thức như GET, POST, PUT, PATCH, DELETE, ... để client có thể gửi yêu cầu đến server tùy theo mục đích tương ứng. Client và server trao đổi dữ liệu qua giao thức HTTP. Tính năng chính của REST API là tính không trạng thái. Tính không trạng thái nghĩa là server không lưu dữ liệu của client giữa các yêu cầu. Các yêu cầu mà client gửi cho server tương tự như URL nhập vào trình duyệt để truy cập vào trang web. Phản hồi từ server là dữ liệu thuần chứ không được kết xuất thành đồ họa như thường thấy trên trang web.

8.5.3 Request và Response

Request Một request đúng chuẩn cần có 4 thành phần:

- URL.
- Method.
- Headers.
- Body.

Chi tiết như sau :

- URL là một địa chỉ duy nhất của một thứ nào đó có thể là trang web, hình ảnh hoặc video. Dựa trên ý tưởng đó, API mở rộng hơn cho những thứ khác chẳng hạn printers, users. Nhờ như thế client dễ dàng cho server biết cái nó muốn là cái gì, những cái này còn được gọi chung là “resources”.
- Method là hành động client muốn tác động lên “resources”. Một số phương thức phổ biến:
 - GET: Trả về một resource hoặc một danh sách resources.

- POST: Tạo mới một resource.
 - PUT: Cập nhật thông tin cho resource (toàn bộ resource).
 - PATCH: Cập nhật thông tin cho resource (một phần resource).
 - DELETE: Xóa một resource.
- Headers: nơi chứa các thông tin cần thiết của một request nhưng end-users không biết có sự tồn tại của nó.
 - Body: nơi chứa thông tin mà client sẽ điền vào (đối với phương thức POST hoặc PUT, PATCH). Ví dụ gửi yêu cầu đăng ký tài khoản thì người dùng cần điền tài khoản, mật khẩu, email, họ tên,...

Response Sau khi nhận được yêu cầu từ phía client, server sẽ xử lý yêu cầu đó và gửi ngược lại cho client một phản hồi. Cấu trúc của 1 response tương đối giống phần request nhưng Status code (cho biết trạng thái của yêu cầu được gửi đi như 200 OK hoặc 404 NOT FOUND) sẽ thay thế cho URL và Method. Tóm lại, response gồm 3 thành phần:

- Status code .
- Headers.
- Body.

8.5.4 Sử dụng và quản lý API

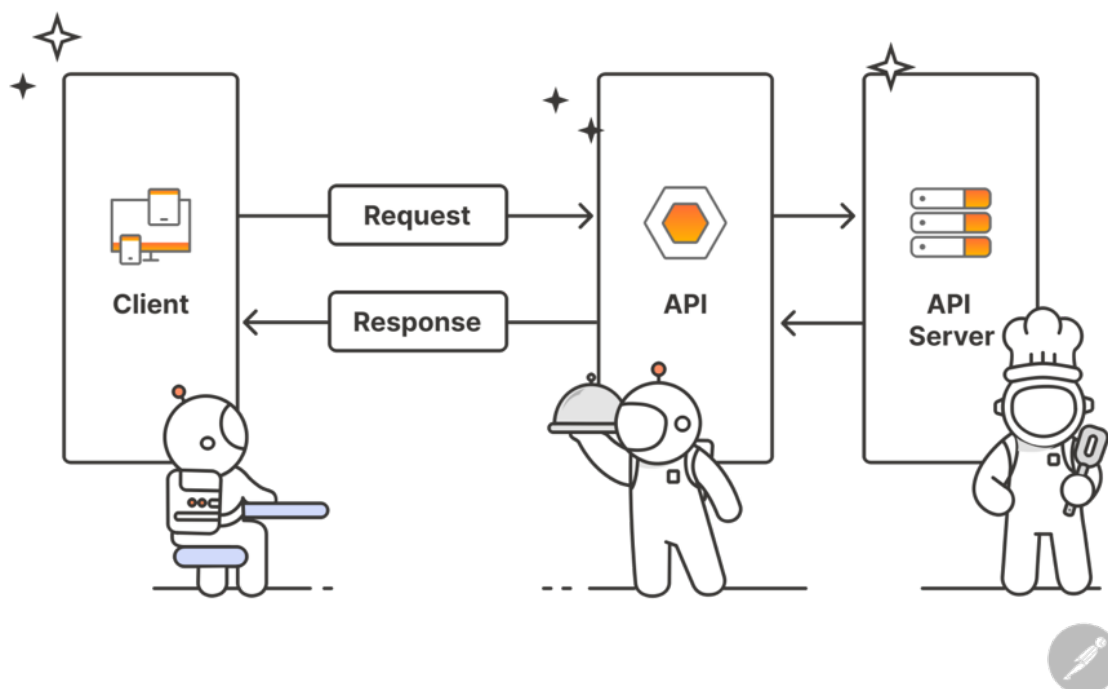
Từ những khái niệm cơ bản trên, để phía frontend có thể truy cập API từ backend hoặc API từ các dịch vụ thứ ba như cổng thanh toán BKPay sẽ gửi yêu cầu thông qua các URL để gọi hàm cần thiết, server sẽ tiếp nhận, gọi hàm tương ứng xử lý các logic phức tạp và trả dữ liệu về. Khi người dùng thực hiện thao tác ví dụ xem danh sách máy in, ở phía frontend tiếp nhận và gửi request với phương thức là GET /api/v1/printers đến máy chủ, máy chủ sẽ trả lời bằng response cho biết thành công hay thất bại qua status code, nếu thành công sẽ bao gồm dữ liệu gồm danh sách các máy in (dữ liệu thô thường sẽ là JSON) ở body, nếu thất bại sẽ kèm theo lỗi tùy theo API đó quy định.

Khi số lượng API trở nên quá lớn, cần có phương pháp để quản lý API tránh việc các ứng dụng gọi API qua lại với nhau tạo sẽ khiến hệ thống khó bảo trì, khó mở rộng. Phương pháp để quản lý đó là API Gateway. API Gateway có thể coi là một cổng trung gian, nó là cổng vào duy nhất tới hệ thống, API Gateway sẽ nhận các requests từ phía client, chỉnh sửa, xác thực và điều hướng chúng đến các API cụ thể trên các services phía sau. Ngoài nhiệm vụ chính là proxy request thì một hệ thống API Gateway thường sẽ đảm nhận luôn vài vai trò khác như bảo mật API, giám sát, phân tích số lượng requests cũng như tình trạng hệ thống phía sau. Tất cả chỉ việc gọi đến API Gateway nên phần code frontend sẽ gọn gàng hơn so với việc phải tracking hàng tá endpoints trên từng services một, nhất là khi mà hệ thống ngày một phình to ra.

8.6 Data Layer

8.6.1 Mô tả về Data Layer

Phương pháp lưu trữ dữ liệu cho hệ thống này được xây dựng xoay quanh thiết kế dữ liệu có quan hệ (relational database), sử dụng PostgreSQL để lưu trữ tất cả dữ liệu. Các thực thể chính như Máy in (Printer), Báo cáo (Report), Người dùng (Person), Sinh viên (Student), Lịch sử mua giấy (PaperLog), Lịch sử in (PrintingLog), SPSO (Student Printing Service Officer) và



Hình 11: Mô tả cách hoạt động API

Cấu hình hệ thống (SystemConfiguration) được tổ chức để quản lý thông tin nhằm tạo ra một hệ thống quản lý dịch vụ in ấn hiệu quả, dễ dùng và tối ưu.

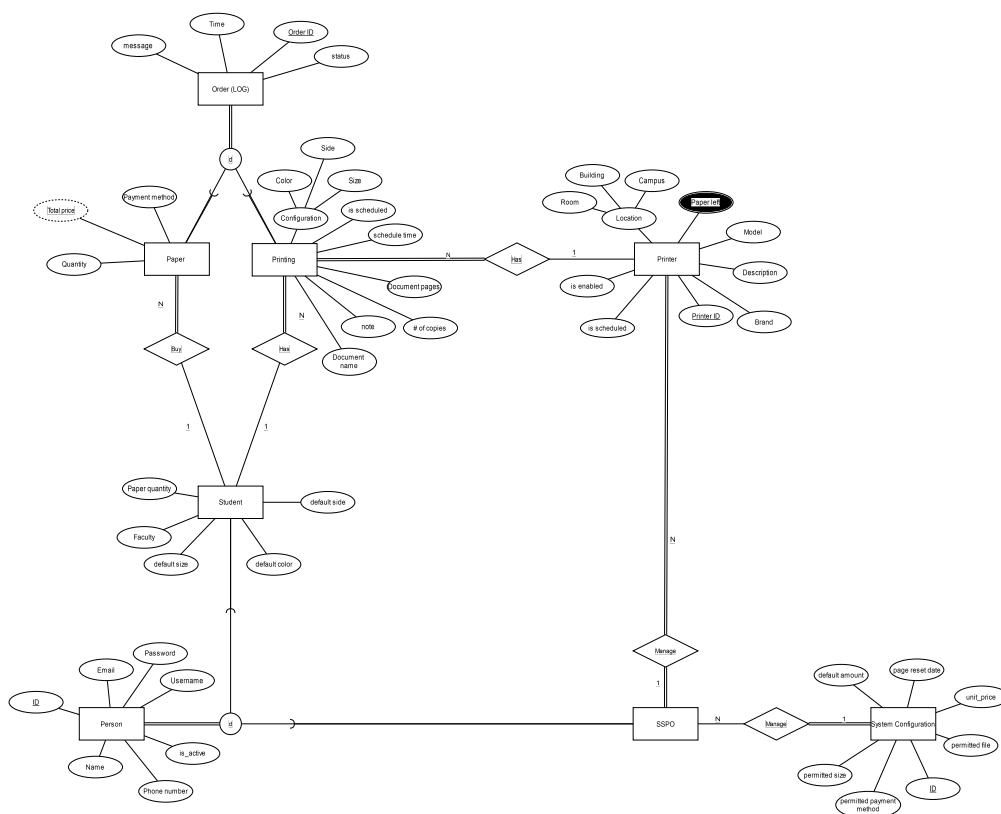
- Máy in: Thực thể Máy in trong PostgreSQL sẽ lưu trữ các thuộc tính như mã máy in (printerID), mẫu máy (model), loại máy (type), nhãn hiệu (brand), và các chi tiết vị trí vật lý như cơ sở (campus), tòa (building), và phòng (room). Các trường liên quan đến cơ chế vận hành của máy in (status, paperLeft, v.v.) sẽ được cập nhật và thiết lập bởi các phương thức enable(), disable(), và print() để theo dõi khả năng hoạt động và tính sẵn sàng của máy in.
- Cấu hình Hệ thống: Thực thể Cấu hình Hệ thống hoạt động như một kho lưu trữ tập trung cho các cài đặt mặc định trên toàn hệ thống, chẳng hạn như số lượng giấy mặc định (defaultAmount - số lượng giấy mà mỗi sinh viên sẽ nhận được sau một khoảng thời gian), ngày đặt lại số trang (pageResetDate - ngày mà mỗi sinh viên sẽ được nhà trường cấp giấy), kích thước mặc định (defaultSize), in màu mặc định (defaultColor), chế độ in hai mặt mặc định (default2Side) - thiết lập mặc định đối với cấu hình khi người dùng lần đầu sử dụng dịch vụ, các loại tệp, kích thước và phương thức thanh toán được hỗ trợ (permittedFiles, permittedSizes và permittedPaymentMethods). Các thông số và cài đặt trên sẽ được quản lý, theo dõi bởi SSPO (Nhân viên Dịch vụ In ấn Sinh viên), các cài đặt này đảm bảo tính nhất quán trong hoạt động của tất cả các máy in. Việc lưu trữ dữ liệu này trong PostgreSQL cho phép truy xuất và cập nhật cấu hình một cách nhanh chóng cũng như hạn chế các lỗi không mong muốn trong quá trình sử dụng của người dùng và vận hành của hệ thống.
- Người dùng: Thực thể Người dùng, được mở rộng bởi các vai trò chuyên biệt như SSPO và Sinh viên, sẽ lưu trữ dữ liệu người dùng trong PostgreSQL với các trường như tên (name),

email, tên đăng nhập (username), mật khẩu (password), và số điện thoại (phoneNumber). PostgreSQL hỗ trợ các ràng buộc quan hệ sẽ giúp áp dụng quyền truy cập dựa trên vai trò: SSPO có thể quản lý máy in và cấu hình, trong khi Sinh viên có thể tải lên tài liệu, chọn máy in, và theo dõi số dư giấy in của mình (paperBalance).

- Ghi lại nhật ký: Cơ sở dữ liệu sẽ lưu trữ chi tiết các giao dịch và hoạt động thông qua các thực thể PaperLog và PrintingLog. PaperLog ghi lại thông tin về việc mua giấy của học sinh (số lượng, phương thức thanh toán, tổng giá), trong khi PrintingLog theo dõi những lần học sinh sử dụng dịch vụ in (tên tài liệu, kích thước, màu sắc, in 1 hay 2 mặt, số lượng bản in). Thực thể Báo cáo (Report) hỗ trợ việc tạo báo cáo để SPSO có thể phân tích, đánh giá về nhu cầu sử dụng và hoạt động của nội bộ HCMUT.

Bằng cách tận dụng PostgreSQL, mô hình dữ liệu quan hệ này cho phép truy vấn hiệu quả, lưu trữ và đảm bảo tính toàn vẹn dữ liệu bằng cách sử dụng các tính năng mạnh mẽ của hệ cơ sở dữ liệu quan hệ, chẳng hạn như khóa ngoại (foreign key), giao dịch (transaction) và các ràng buộc, giúp cho việc quản lý dữ liệu có cấu trúc của hệ thống, hỗ trợ các tương tác gắn kết giữa tất cả các thực thể trở nên dễ dàng trong khi vẫn có thể đảm bảo tính bảo mật và tính nhất quán của dữ liệu.

8.6.2 ERD



Hình 12: Layer Architecture

Có thể xem tại link: [GitHub](#)

9 Component Diagram

9.1 Mô tả

User Authentication Thành phần này chịu trách nhiệm xác thực người dùng khi họ đăng nhập vào hệ thống. Khi người dùng nhập thông tin đăng nhập (tên đăng nhập và mật khẩu), thành phần này kiểm tra xem thông tin có trùng khớp với dữ liệu lưu trữ hay không. Nếu thông tin chính xác, thành phần này sẽ gửi thông tin đã xác thực đến thành phần User Authorization. Đây là bước đầu tiên trong quy trình và bắt buộc để đảm bảo tính bảo mật của hệ thống.

User Authorization Thành phần này được kích hoạt sau khi người dùng được xác thực thành công. Nó kiểm tra quyền hạn của người dùng dựa trên thông tin và vai trò của họ trong hệ thống, đảm bảo rằng người dùng chỉ có thể truy cập và sử dụng các chức năng mà họ được phép, ví dụ Admin hay SPSO thì sẽ không có các tính năng in hay thanh toán. Thông tin từ thành phần này sẽ được gửi đến các thành phần khác trong hệ thống như File Uploading & Saving và Printer Selection để đảm bảo rằng người dùng có đủ quyền thực hiện các hành động tiếp theo.

File Uploading & Saving Thành phần này đảm nhận việc tải lên và lưu trữ tệp do người dùng cung cấp. Nó yêu cầu thông tin đã được xác thực và ủy quyền từ các thành phần trước đó. Tệp tải lên phải tuân theo một số tiêu chí nhất định về định dạng và kích thước. Sau khi tệp được lưu trữ thành công, thông tin về tệp sẽ được chuyển đến thành phần File Validation để kiểm tra thêm.

File Validation Đây là thành phần kiểm tra tính hợp lệ của tệp vừa được tải lên. Ngoài việc xác định tệp có đúng định dạng yêu cầu (ví dụ: PDF, DOCX, v.v.), thành phần này còn cần thông tin từ thành phần Printer Selection để biết máy in đã chọn có hỗ trợ định dạng đó hay không. Điều này giúp đảm bảo rằng tệp được chọn có thể in được trên máy in cụ thể. Nếu tệp không đạt yêu cầu, hệ thống sẽ trả về thông báo lỗi để người dùng biết và tải lên lại tệp đúng.

Printer Selection Thành phần này cho phép người dùng chọn máy in từ danh sách các máy in khả dụng. Thông tin từ User Authorization được sử dụng để quyết định người dùng có quyền truy cập vào máy in nào. Thành phần này cũng gửi thông tin máy in đến File Validation để kiểm tra khả năng hỗ trợ định dạng tệp và đến Printing Configuration để xác định các tùy chọn cấu hình có thể chỉnh.

Printing Booking Thành phần này chịu trách nhiệm tạo lịch hẹn in ấn, bao gồm thông tin như thời gian, số lượng bản in và các chi tiết khác. Nó yêu cầu thông tin từ User Authorization và Printer Selection để đảm bảo rằng người dùng có quyền đặt lịch in và máy in được chọn đang khả dụng. Thông tin đặt chỗ này sẽ được chuyển đến Printing Configuration để tiếp tục xử lý.

Printing Configuration Đây là thành phần cho phép người dùng cấu hình các tùy chọn in chi tiết như độ phân giải, kích thước giấy, số lượng bản in, in hai mặt hay đơn mặt, v.v. Thành phần này cần kết quả từ File Validation để xác định xem tệp có hợp lệ hay không, và thông tin từ Printer Selection để xác định máy in nào đang được sử dụng, từ đó chỉ hiển thị các tùy chọn cấu hình phù hợp với máy in đó. Nếu kết quả kiểm tra tệp không đạt yêu cầu, các tùy chọn cấu hình sẽ không hiển thị cho đến khi tệp hợp lệ được cung cấp.

Remaining Page (Subsystem) Đây là một hệ con phụ trách việc kiểm tra số lượng trang còn lại trong hạn mức in của người dùng và máy in.

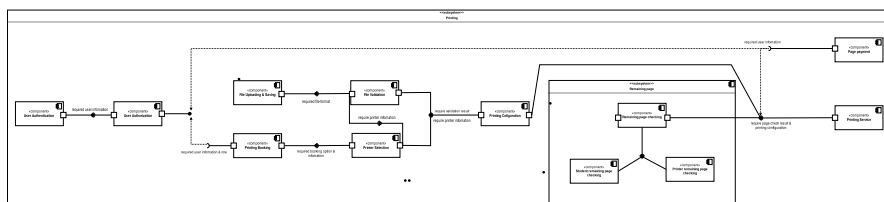
- Student Remaining Page Checking: Thành phần này kiểm tra số lượng trang mà sinh viên còn có thể in dựa trên hạn mức in ấn được phân bổ. Nó yêu cầu thông tin từ User Authorization để xác định người dùng cụ thể.
- Printer Remaining Page Checking: Thành phần này kiểm tra số lượng trang còn lại của từng máy in. Điều này giúp xác nhận rằng máy in có đủ tài nguyên để hoàn thành lệnh in. Kết quả từ hai thành phần này sẽ được gửi đến Printing Service.

Page Payment Thành phần này xử lý thanh toán cho số trang mà người dùng dự định in. Nó cần thông tin người dùng từ User Authorization và kết quả từ Remaining Page để xác định số lượng trang còn lại và tính toán chi phí. Người dùng cần thực hiện thanh toán trước khi lệnh in có thể được gửi đến Printing Service.

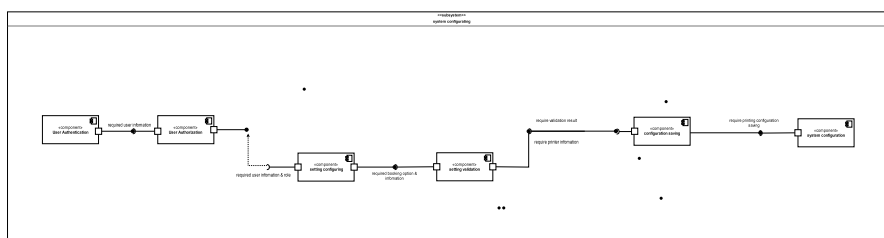
Printing Service Đây là thành phần xử lý yêu cầu in sau khi tất cả các bước chuẩn bị đã được hoàn tất. Nó nhận kết quả kiểm tra trang từ Remaining Page, thông tin thanh toán từ Page Payment, và cấu hình in từ Printing Configuration để xử lý lệnh in. Thành phần này đảm bảo rằng lệnh in chỉ được thực hiện khi mọi thông tin đã được xác minh và thông số in đã được cấu hình chính xác.

9.2 Diagram

Có thể xem tại link: [GitHub](#) (hiện tại chưa p ublic github) vì thế có thể xem tạm tại: [Google Drive](#)



Hình 1: Component Diagram cho chức năng in



Hình 2: Component Diagram cho chức năng cài đặt hệ thống

10 Usability Testing cho SSPS

10.1 Thiết kế bài kiểm tra

Mục tiêu Kiểm tra tính dễ sử dụng của website SSPS và các lỗi tìm ẩn

Nhiệm vụ

1. Tải file cần in lên website để tiến hành in.
2. Kiểm tra chức năng setting trang giấy khi in.
3. Thực hiện chức năng in.
4. Sử dụng website để tìm ra các lỗi và các chỗ chưa hợp lý.

Đối tượng tham gia

- Số lượng: 5-7 người.
- Đặc điểm:
 - Có trình độ công nghệ tốt
 - Có tính cẩn thận và gu thẩm mỹ nhất định.

Kịch bản kiểm tra Bạn có một file tài liệu muốn in dưới một định dạng phức tạp và cần đặt lịch in. Khi gặp lỗi cần tìm thông tin liên hệ để có thể sửa lỗi. Không có hướng dẫn sử dụng từ trước.

Chuẩn bị cho bài kiểm tra

- Phiếu khảo sát về thông tin của người kiểm tra.
- Phiếu ghi chú để ghi lại các góp ý, phản hồi.

Chiến lược kiểm tra

- Định tính: Quan sát hành vi người dùng và thu thập phản hồi trực tiếp.
- Định lượng: Ghi nhận thời gian hoàn thành nhiệm vụ và số lỗi mắc phải.

Danh sách người tham gia kiểm tra

- Chế Minh Đức – 2210783
- Nguyễn Đình Đức – 2210794
- Phạm Lê Hoài Hải – 2210901
- Lê Quốc Huy – 2211194
- Hà Tiến Thắng – 2213184
- Hà Lê Tín – 2213484

10.2 Kết quả bài kiểm tra

10.2.1 Chế Minh Đức

Điểm mạnh

- Giao diện đơn giản dễ sử dụng.
- Các nút bấm to rõ dễ tìm thấy và sử dụng.
- Các chức năng được thể hiện một cách rõ ràng.

Điểm yếu

- Thông tin liên hệ chưa được thể hiện rõ, khó tìm.
- Thiếu hướng dẫn sử dụng.

10.2.2 Nguyễn Đình Đức

Điểm mạnh

- Tốc độ tải file nhanh, không gặp vấn đề về định dạng file.
- Các tùy chọn cài đặt trang giấy rất chi tiết.
- Giao diện trực quan, dễ nhận diện các chức năng chính.

Điểm yếu

- Khi gặp lỗi tải file, không có thông báo lỗi cụ thể.
- Menu cài đặt trang giấy chưa thân thiện với người dùng lần đầu.

10.2.3 Phạm Lê Hoài Hải

Điểm mạnh

- Quy trình in rõ ràng, dễ theo dõi.
- Tích hợp tốt giữa các bước tải file, cài đặt và in.
- Thiết kế hiện đại, phù hợp với xu hướng.

Điểm yếu

- Chức năng tìm kiếm thông tin liên hệ chưa hiệu quả.
- Mục hướng dẫn nên đặt ở nơi dễ thấy hơn (hiện tại bị ẩn trong menu phụ).

10.2.4 Lê Quốc Huy

Điểm mạnh

- Dễ dàng đặt lịch in và theo dõi trạng thái.
- Tích hợp nút quay lại khi chọn sai bước.

Điểm yếu

- Chưa hỗ trợ thông báo tình trạng máy in (đang hoạt động hay lỗi).
- Một số nút bấm (như nút tải lại file) không rõ ràng về chức năng.

10.2.5 Hà Tiến Thắng

Điểm mạnh

- Hệ thống hoạt động ổn định, không gặp lỗi trong quá trình sử dụng.
- Thích hợp với người dùng có trình độ công nghệ trung bình trở lên.

Điểm yếu

- Thiếu tính năng gợi ý khắc phục khi file bị lỗi không tương thích.
- Phần cài đặt nâng cao khá khó tiếp cận với người dùng không chuyên.

10.2.6 Hà Lê Tín

Điểm mạnh

- Hệ thống phản hồi nhanh khi thực hiện các thao tác in.
- Các tùy chọn cài đặt được phân nhóm hợp lý, dễ tìm kiếm.
- Màu sắc và phông chữ của giao diện thân thiện, dễ đọc.

Điểm yếu

- Không có thông báo rõ ràng khi hoàn tất in.
- Nút hỗ trợ (help) khó tìm, không nổi bật trên giao diện.

10.3 Kết quả tổng thể

10.3.1 Task 1: Tải file cần in lên website

- Thời gian trung bình hoàn thành: 25 giây.
- Lỗi phổ biến:
 - 4/6 người không nhận ra nút "Tải file" ngay lập tức (do kích thước và vị trí của nút không đủ nổi bật).
- Phản hồi:
 - "Tôi đã phải tìm khá lâu để thấy nút tải file, nó cần lớn hơn hoặc dễ nhận diện hơn."
 - "Chức năng tải file hoạt động tốt, nhưng biểu tượng hơi mờ nhạt."

10.3.2 Task 2: Kiểm tra chức năng cài đặt trang giấy khi in

- **Thời gian trung bình hoàn thành:** 40 giây.
- **Lỗi phổ biến:**
 - 3/6 người không tìm thấy cài đặt định dạng trang giấy vì chúng nằm sâu trong menu phụ.
 - 2/6 người gặp khó khăn khi chọn kích thước khổ giấy phù hợp.
- **Phản hồi:**
 - "Menu cài đặt giấy hơi phức tạp và không dễ hiểu với người mới."
 - "Tôi thích cách phân loại các tùy chọn, nhưng chúng cần rõ ràng hơn."

10.3.3 Task 3: Thực hiện chức năng in

- **Thời gian trung bình hoàn thành:** 15 giây.
- **Lỗi phổ biến:**
 - 2/6 người không thấy nút in vì nó không được làm nổi bật trên giao diện.
 - Không có thông báo trạng thái rõ ràng khi lệnh in được thực hiện.
- **Phản hồi:**
 - "Nút in nên được làm nổi bật hơn, đặc biệt với người dùng mới."
 - "Tôi không biết chắc máy in đã bắt đầu in hay chưa, cần có thông báo rõ hơn."

10.3.4 Task 4: Tìm lỗi và các điểm chưa hợp lý trên website

- **Lỗi phổ biến:**
 - Thông tin liên hệ bị ẩn trong mục "Trợ giúp", khiến người dùng khó tìm.
 - Thiếu hướng dẫn chi tiết về các chức năng nâng cao.
- **Phản hồi:**
 - "Thông tin liên hệ cần được đặt ở một vị trí dễ tìm hơn, ví dụ trên trang chính."
 - "Không có hướng dẫn cụ thể khi gặp lỗi, tôi phải tự mò mẫm."

10.3.5 Tóm tắt kết quả

- **Điểm mạnh:**
 - Giao diện đơn giản, thân thiện với người dùng.
 - Các chức năng in cơ bản hoạt động tốt, không có lỗi kỹ thuật lớn.
 - Hỗ trợ đa dạng định dạng file và khổ giấy.
- **Điểm yếu:**
 - Một số nút chức năng (như tải file, in) không nổi bật, gây khó khăn cho người dùng.
 - Thiếu thông báo trạng thái khi thực hiện các thao tác in.
 - Thông tin liên hệ và hướng dẫn sử dụng chưa được bố trí hợp lý.

11 Hiện thực dự án

11.1 Spring Boot cho Backend

Spring Boot là một framework mã nguồn mở dựa trên Spring Framework, được thiết kế để giúp việc phát triển các ứng dụng Java trở nên đơn giản và nhanh chóng hơn. Với các tính năng mạnh mẽ và khả năng mở rộng, Spring Boot là lựa chọn hàng đầu cho nhiều nhà phát triển trong việc xây dựng backend hiện đại.

Đặc điểm nổi bật của Spring Boot

- **Cấu hình tự động:** Spring Boot tự động cấu hình các thành phần cần thiết, giúp giảm thiểu thời gian và công sức khi thiết lập môi trường ứng dụng.
- **Khởi tạo nhanh:** Hỗ trợ các công cụ như Spring Initializr để tạo dự án nhanh chóng với các phụ thuộc phù hợp.
- **RESTful API dễ dàng:** Tích hợp tốt với Spring MVC, giúp lập trình viên dễ dàng xây dựng các dịch vụ RESTful để giao tiếp với frontend.
- **Tích hợp hệ sinh thái Spring:** Spring Boot hoạt động tốt với các module trong hệ sinh thái Spring như Spring Data, Spring Security, và Spring Cloud.
- **Hỗ trợ đa dạng cơ sở dữ liệu:** Tích hợp với các cơ sở dữ liệu quan hệ (MySQL, PostgreSQL) và NoSQL (MongoDB, Cassandra).
- **Khả năng mở rộng:** Kiến trúc modular và khả năng tích hợp nhiều thư viện bên thứ ba giúp ứng dụng dễ dàng mở rộng khi nhu cầu tăng lên.

Cách Spring Boot hoạt động Spring Boot sử dụng các nguyên tắc thiết kế và mô hình của Spring Framework, nhưng đơn giản hóa đáng kể nhờ các tính năng sau:

- **Starter POMs:** Cung cấp các bộ phụ thuộc (dependencies) được cấu hình sẵn như `spring-boot-starter-web`, `spring-boot-starter-data-jpa`.
- **Embedded Server:** Tích hợp sẵn các máy chủ như Tomcat hoặc Jetty, cho phép chạy ứng dụng như một tệp JAR độc lập mà không cần cấu hình máy chủ bên ngoài.
- **Spring Actuator:** Cung cấp các endpoint theo dõi sức khỏe và hiệu năng của ứng dụng.
- **Externalized Configuration:** Hỗ trợ cấu hình ứng dụng thông qua file `application.properties`, `application.yml`, hoặc biến môi trường.

Các ứng dụng phổ biến của Spring Boot

- **Dịch vụ RESTful:** Spring Boot được sử dụng rộng rãi để xây dựng các API REST phục vụ cho giao tiếp giữa các hệ thống hoặc ứng dụng.
- **Ứng dụng doanh nghiệp:** Với sự hỗ trợ mạnh mẽ từ Spring Data và Spring Security, Spring Boot phù hợp để phát triển các hệ thống quản lý, CRM hoặc ERP.
- **Ứng dụng phân tán:** Khi kết hợp với Spring Cloud, Spring Boot trở thành giải pháp lý tưởng cho kiến trúc microservices.

Ưu điểm của Spring Boot

- **Dễ học và sử dụng:** Cung cấp các công cụ và thư viện giúp lập trình viên nhanh chóng làm quen và phát triển ứng dụng.
- **Hiệu suất cao:** Được tối ưu hóa cho các ứng dụng có quy mô lớn.
- **Hỗ trợ cộng đồng mạnh mẽ:** Có một cộng đồng phát triển lớn, tài liệu phong phú và được cập nhật thường xuyên.

Nhược điểm cần cân nhắc

- **Đòi hỏi tài nguyên:** Ứng dụng Spring Boot có thể yêu cầu nhiều tài nguyên hơn so với các framework nhỏ gọn khác.
- **Đường cong học tập:** Đối với người mới, việc hiểu sâu các khái niệm của Spring Boot và Spring Framework có thể mất thời gian.

Spring Boot là một công cụ mạnh mẽ và linh hoạt, cung cấp nền tảng ổn định và hiệu quả để xây dựng các ứng dụng backend, đặc biệt trong các dự án có yêu cầu cao về khả năng mở rộng và bảo trì.

11.2 React JS cho Frontend

React JS, được phát triển bởi Facebook, là một thư viện JavaScript phổ biến dùng để xây dựng giao diện người dùng (UI). Nhờ kiến trúc component-based và hiệu suất cao, React JS trở thành lựa chọn hàng đầu cho việc phát triển frontend trong các ứng dụng hiện đại.

11.2.0.1 Đặc điểm nổi bật của React JS

- **Kiến trúc Component-based:** Giao diện người dùng được chia thành các thành phần nhỏ gọn (components), dễ quản lý và tái sử dụng.
- **Virtual DOM:** Tăng tốc độ cập nhật giao diện bằng cách chỉ thay đổi những phần thực sự cần thiết thay vì cập nhật toàn bộ DOM.
- **Hỗ trợ JSX:** JSX (JavaScript XML) cho phép viết mã React dưới dạng cú pháp giống HTML, giúp mã dễ đọc và bảo trì hơn.
- **Unidirectional Data Flow:** Sử dụng luồng dữ liệu một chiều, giúp tăng tính dự đoán và kiểm soát trạng thái ứng dụng.
- **Tích hợp tốt với các thư viện khác:** Dễ dàng tích hợp với các công cụ như Redux, React Router hoặc các thư viện CSS như TailwindCSS.

11.2.0.2 Ứng dụng của React JS React JS được sử dụng trong nhiều loại dự án frontend, bao gồm:

- **Ứng dụng đơn trang (SPA):** Xây dựng các SPA hiệu quả, nơi hầu hết các yêu cầu giao tiếp với máy chủ đều được thực hiện qua API.
- **Giao diện động:** Phát triển các ứng dụng có giao diện tương tác cao, như bảng điều khiển (dashboards) hoặc ứng dụng thương mại điện tử.
- **Ứng dụng đa nền tảng:** Kết hợp với React Native để xây dựng ứng dụng di động từ cùng một codebase.

11.2.0.3 Cách hoạt động của React JS React tập trung vào xây dựng giao diện người dùng qua các thành phần (components). Một số điểm nổi bật trong cách hoạt động của React:

- **Components:** Giao diện được chia thành các thành phần nhỏ, mỗi thành phần quản lý logic và trạng thái của riêng nó.
- **Props và State:**
 - **Props:** Truyền dữ liệu từ component cha sang component con.
 - **State:** Quản lý dữ liệu thay đổi trong mỗi component, ảnh hưởng trực tiếp đến giao diện.
- **Lifecycle Methods:** Cung cấp các phương thức vòng đời để kiểm soát hành vi của component trong các giai đoạn khác nhau, như khi được gắn kết (mounted) hoặc hủy bỏ (unmounted).

11.2.0.4 Ưu điểm của React JS

- **Hiệu suất cao:** Virtual DOM và cách tiếp cận hiệu quả giúp React xử lý các ứng dụng lớn một cách nhanh chóng.
- **Dễ bảo trì:** Kiến trúc component-based giúp việc quản lý và bảo trì mã nguồn trở nên dễ dàng hơn.
- **Hệ sinh thái phong phú:** Có nhiều thư viện và công cụ hỗ trợ, giúp mở rộng chức năng một cách linh hoạt.
- **Cộng đồng lớn:** Được hỗ trợ bởi một cộng đồng phát triển lớn và tài liệu phong phú.

11.2.0.5 Nhược điểm cần cân nhắc

- **Đòi hỏi cấu hình ban đầu:** Cần cấu hình nhiều hơn so với một số framework khác khi bắt đầu dự án.
- **Học tập liên tục:** Đòi hỏi lập trình viên phải hiểu sâu các khái niệm như JSX, props, state, hooks và lifecycle để khai thác tối đa khả năng của React.
- **Phụ thuộc vào thư viện bên ngoài:** Tập trung vào UI nên cần thêm thư viện như Redux hoặc React Query để quản lý trạng thái phức tạp.

React JS là công cụ mạnh mẽ, phù hợp với các ứng dụng frontend từ đơn giản đến phức tạp. Với khả năng tái sử dụng component, hiệu suất cao và hệ sinh thái phát triển mạnh mẽ.

Tài liệu tham khảo

- [1] R. C. Martin, *Clean Code: A Handbook of Agile Software Craftsmanship*. Upper Saddle River, NJ: Prentice Hall, 2008.
- [2] A. Hunt and D. Thomas, *The Pragmatic Programmer: Your Journey to Mastery*, 20th Anniversary Edition. Addison-Wesley Professional, 2019.
- [3] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional, 1994.
- [4] I. Sommerville, *Software Engineering*, 10th ed. Harlow, England: Pearson Education, 2015.