

VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY
UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING



SOFTWARE ENGINEERING

Assignment

HCMUT Student Smart Printing Service

Advisor: Trần Trương Tuấn Phát
Students: Võ Nguyễn Đức Phát - 2212540 .
Lê Vĩnh Nghiệp - 2212213 .
Nguyễn Việt Anh - 2210116 .
Nguyễn Bá Việt Quang - 2212741 .
Trần Chính Bách - 2210187
Đặng Quốc Phong - 2212548 .
Nguyễn Đăng Cường - 2210432
Nguyễn Lê Anh Đức - 2210796

HO CHI MINH CITY, NOVEMBER 2024

Mục lục

1 Phần 1	2
1.1 Mô tả bối cảnh miền của dịch vụ in thông minh cho sinh viên tại HCMUT	2
1.2 Yêu cầu chức năng và phi chức năng có thể suy ra từ mô tả dự án	3
1.3 Use-case diagram	4
1.3.1 Use-case diagram: User	4
1.3.2 Use-case diagram: SPSO	7
2 Phần 2	10
2.1 Vẽ Activity Diagram cho mô hình	10
2.2 Vẽ Sequence Diagram cho mô hình	16
2.3 Vẽ Class Diagram cho mô hình	24
3 Architecture design	26
3.1 Architectural Diagram: User	26
3.2 Architectural Diagram: SPSO	28
3.2.1 Presentation Strategy	30
3.2.2 Presentation approach	30
3.2.3 Data storage approach	32
3.2.4 API management	33
3.3 Component Diagram	35
3.3.1 Component Diagram: Printing	35
3.3.2 Component Diagram: View History	36
3.3.3 Component Diagram: Page Config	36
4 Implementation – Sprint 1	38
4.1 Cài đặt Github	38
4.2 Thêm tài liệu vào Github	38
4.3 Thực hiện usability testing	38
4.3.1 Usability Testing là gì?	38
4.3.2 Chọn người kiểm tra và người tham gia	40
4.3.3 Xác định nhiệm vụ	40
4.3.4 Phương thức kiểm tra	41
4.3.5 Tiến hành kiểm tra	42
4.3.6 Thu thập phản hồi và báo cáo kết quả	48
5 Giao diện trang web	49

1 Phần 1

Task 1: Requirement Elicitation

1.1 Mô tả bối cảnh miền của dịch vụ in thông minh cho sinh viên tại HCMUT

Bối cảnh miền của dịch vụ in thông minh cho sinh viên tại HCMUT:

Hiện nay, sinh viên, giảng viên cũng như tập thể cán bộ học tập và làm việc tại trường Đại học Bách Khoa có nhu cầu cao về việc in ấn tài liệu. Mặc dù với sự phát triển của khoa học công nghệ, tài liệu điện tử đã xuất hiện nhiều hơn nhưng vẫn chưa thể thay thế được tính tiện lợi cũng như dễ sử dụng của tài liệu bản giấy. Sinh viên thường in các giáo trình, đề thi, bảng công thức, ... phục vụ cho học tập và thi cử, giảng viên cũng có nhu cầu in ấn bài giảng, báo cáo, ...

Chính vì lẽ đó, những phòng photocopy truyền thống tại cả 2 cơ sở đã trở nên quá tải và tốn đọng nhiều bất cập. Dễ thấy nhất chính là việc quy trình in ấn hiện tại đang quá rườm rà, trải qua nhiều công đoạn, do đó không thể nào đáp ứng được tốt khi số lượng sinh viên cần photo tài liệu trở nên đông đúc. Ngoài ra, các vấn đề về tùy chỉnh định dạng, giá cả của việc in ấn vẫn còn đang gây ra nhiều khúc mắc những sinh viên, giảng viên và những người làm dịch vụ in ấn.

Chính vì lẽ đó, cần có một hệ thống cung cấp dịch vụ in ấn thông minh cho người dùng. Hệ thống này có tên Student Smart Printing Service (HCMUT-SSPS). Hệ thống này sẽ khắc phục được những nhược điểm của việc in truyền thống cũng như mang lại lợi ích cho sinh viên và giảng viên trong trường.

Các bên liên quan và nhu cầu của họ:

- **Sinh viên (Users):**

- Sinh viên là người dùng chính của hệ thống. Họ cần một dịch vụ in ấn tiện lợi, dễ sử dụng và có thể truy cập từ mọi nơi trong khuôn viên trường.
- Nhu cầu của sinh viên bao gồm việc in ấn tài liệu học tập nhanh chóng, theo dõi lịch sử in ấn của họ, và mua thêm trang in khi vượt quá số trang mặc định được cung cấp mỗi học kỳ.
- Sinh viên cũng cần một phương thức thanh toán dễ dàng và minh bạch khi mua thêm trang in, ví dụ qua BKPay – hệ thống thanh toán của trường.

- **Nhân viên quản lý dịch vụ in ấn (SPSO):**

- SPSO chịu trách nhiệm quản lý các máy in và cấu hình hệ thống. Họ cần một công cụ để dễ dàng thêm, vô hiệu hóa hoặc kích hoạt máy in, đồng thời theo dõi lịch sử in của sinh viên để xử lý các sự cố phát sinh.
- SPSO còn cần quản lý các tệp tin được phép in, thay đổi số trang mặc định cấp cho sinh viên



mỗi học kỳ, và tạo báo cáo hàng tháng hoặc hàng năm về việc sử dụng dịch vụ in.

• **Đội ngũ công nghệ thông tin (IT Team):**

- IT Team cần đảm bảo hệ thống vận hành ổn định và dễ dàng bảo trì.
- Họ cũng cần có khả năng mở rộng hệ thống để hỗ trợ nhiều sinh viên hơn trong tương lai và tích hợp với các hệ thống khác như HCMUT_SSO (dịch vụ xác thực sinh viên) và BKPay.
- Điều này đòi hỏi triển khai các giải pháp hạ tầng phù hợp và đảm bảo tính bảo mật, ổn định của hệ thống.

Lợi ích của HCMUT-SSPS đối với các bên liên quan:

- **Sinh viên:** HCMUT-SSPS mang lại lợi ích lớn nhờ khả năng in ấn tài liệu dễ dàng, nhanh chóng mà không cần phải tìm kiếm nơi in bên ngoài trường.
- **SPSO:** Hệ thống giúp quản lý máy in hiệu quả, theo dõi hoạt động, xử lý sự cố và tạo báo cáo.
- **Đội ngũ IT:** IT có thể giám sát, bảo trì và đảm bảo tính bảo mật, mở rộng hệ thống khi cần thiết.

1.2 Yêu cầu chức năng và phi chức năng có thể suy ra từ mô tả dự án

Yêu cầu chức năng (Functional Requirements):

1. **Đối với sinh viên:**

- Xác thực danh tính sinh viên qua tài khoản MyBK.
- Tải lên tài liệu ở định dạng cho phép và chọn thuộc tính in ấn.
- Chọn máy in dựa trên vị trí trong khuôn viên trường.
- Kiểm tra số lượng trang in còn lại.
- Mua thêm trang in qua BKPay.
- Xem lịch sử in ấn bao gồm tên tệp, số trang, thời gian in.

2. **Đối với SPSO:**

- Thêm, kích hoạt hoặc vô hiệu hóa máy in.
- Quản lý định dạng tệp được phép in.
- Truy cập lịch sử in của sinh viên hoặc toàn bộ sinh viên.

- Cung cấp giấy in cho người dùng.
- Tạo báo cáo hàng tháng/năm về việc sử dụng hệ thống.
- Thay đổi số lượng trang in mặc định mỗi học kỳ.

3. Đối với đội ngũ IT:

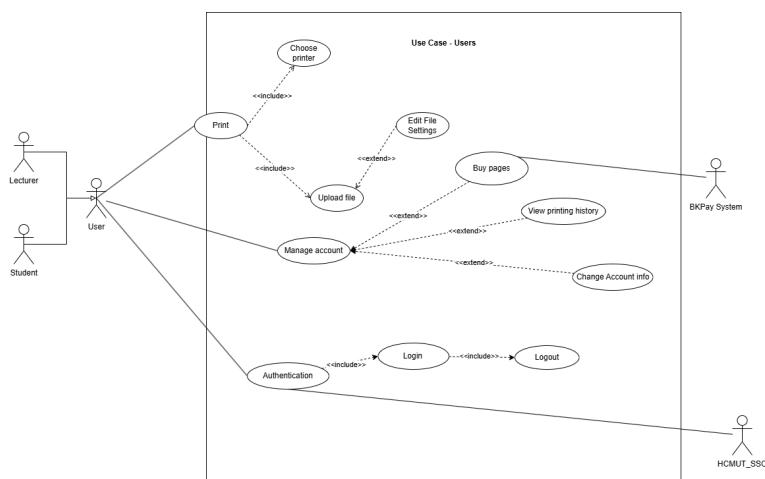
- Tích hợp với HCMUT_SSO và BKPay.
- Giám sát hiệu suất hệ thống và nhận thông báo khi có sự cố.
- Sao lưu và khôi phục dữ liệu khi cần.
- Hỗ trợ nâng cấp và bảo trì hệ thống.

Yêu cầu phi chức năng (Non-functional Requirements):

- **Tính bảo mật:** Chỉ sinh viên xác thực mới có thể sử dụng dịch vụ, bảo vệ dữ liệu tài liệu và thanh toán.
- **Tính dễ sử dụng:** Giao diện thân thiện, dễ hiểu với hướng dẫn chi tiết.
- **Tính mở rộng:** Hỗ trợ thêm máy in và sinh viên mà không ảnh hưởng hiệu suất.
- **Hiệu suất:** Xử lý hàng trăm yêu cầu đồng thời mà không chậm.
- **Tính ổn định:** Hoạt động ổn định, không gián đoạn.

1.3 Use-case diagram

1.3.1 Use-case diagram: User





Use Case: Print

- **Actors:** Sinh viên, nhân viên sử dụng dịch vụ in ấn
- **Description:** Người dùng chọn máy in, upload file cần in và định dạng file cần in.
- **Trigger:** Người dùng bấm vào nút “IN NGAY”
- **Pre-conditions:**
 - Người dùng đã đăng nhập vào hệ thống
 - Thiết bị của người dùng đã được kết nối internet
- **Post-conditions:** Người dùng chọn được máy in và định dạng file cần in theo ý mình mong muốn
- **Normal Flow:**
 1. Sau khi bấm nút “IN NGAY” ở trang chủ, một cửa sổ in sẽ được hiển thị phía người dùng gồm nơi tải file và các tùy chọn in.
 2. Người dùng bấm vào nút upload và chọn file cần in.
 3. Người dùng chọn các tùy chọn in theo ý muốn như: máy in, số bản copy, trang cần in, kích thước giấy, ...
 4. Sinh viên bấm nút in để gửi yêu cầu về hệ thống.
 5. Sinh viên được đăng nhập vào hệ thống in ấn thành công.
- **Alternative Flow:** Không có
- **Exceptions Flow:**
 1. Người dùng chưa upload file hoặc không điền đủ các tùy chọn được hiển thị, hệ thống trả về thông báo lỗi thiếu thông tin cho người dùng.
 2. Người dùng không đủ số dư in, hệ thống trả về thông báo hết tiền.

Use Case: Manage Account

- **Actors:** Sinh viên, giảng viên sử dụng dịch vụ in ấn, hệ thống BKPAY.
- **Description:** Người dùng chỉnh sửa thông tin cá nhân, nạp tiền vào tài khoản, và xem lịch sử in ấn.
- **Trigger:** Người dùng bấm vào biểu tượng ở góc trên bên phải của trang chính.

- **Pre-conditions:**

- Người dùng đã đăng nhập vào hệ thống.
- Thiết bị của người dùng đã được kết nối internet.

- **Post-conditions:**

- Người dùng chỉnh sửa thông tin cá nhân thành công.
- Người dùng xem được lịch sử in.
- Người dùng nạp thêm tiền vào tài khoản thành công.

- **Normal Flow:**

1. Sau khi người dùng bấm vào biểu tượng ở góc trên bên phải của trang chính.
2. Hệ thống trả về giao diện tài khoản cho người dùng.
3. Người dùng chọn các tùy chọn:

- **Info:** Để xem và chỉnh sửa thông tin cá nhân.
- **History:** Để xem lịch sử in.
- **BuyPage:** Để nạp thêm trang in thông qua dịch vụ BKPAY.

- **Alternative Flow:** Không có.

- **Exceptions Flow:** Người dùng không đủ tiền để thanh toán, hệ thống trả về thông báo hết tiền.

Use Case: Authentication

- **Actors:** Người dùng, HCMUT_SPSO, HCMUT_SSO.

- **Description:** Người sử dụng dịch vụ và nhân viên của hệ thống thực hiện đăng nhập vào hệ thống.

- **Trigger:** Người dùng bấm vào biểu tượng ở góc trên cùng bên phải của trang chính.

- **Pre-conditions:**

- Người dùng chưa đăng nhập vào hệ thống.
- Thiết bị của người dùng được kết nối mạng internet đầy đủ.

- **Post-conditions:**

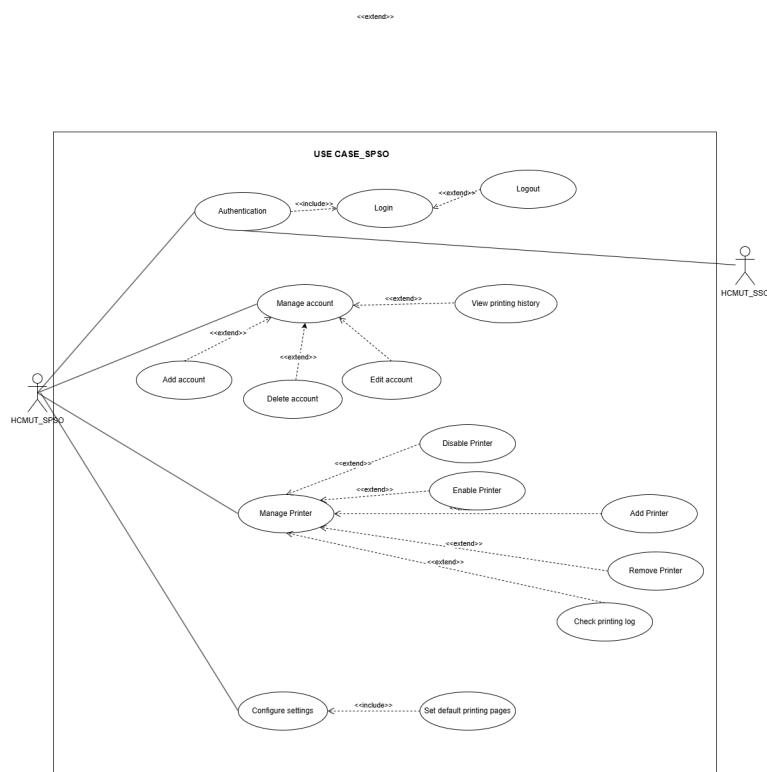
- Người dùng đăng nhập thành công vào hệ thống.

- **Normal Flow:**

1. Người dùng bấm vào biểu tượng ở góc trên cùng bên phải của trang chính.
2. Một trang đăng nhập được hiển thị cho người dùng.
3. Người dùng nhập tài khoản và mật khẩu.
4. Người dùng ấn nút đăng nhập.
5. Hệ thống HCMUT_SSO xác thực tài khoản của người dùng và trả về thông báo đăng nhập thành công.

- **Alternative Flow:** Sau khi đăng nhập thành công, người dùng có thể đăng xuất bằng cách ấn nút *Log Out*.
- **Exceptions Flow:** Nếu người dùng nhập sai tài khoản hoặc mật khẩu, hệ thống sẽ trả về thông báo sai thông tin.

1.3.2 Use-case diagram: SPSO



Use Case: Manage Account

- **Actors:** HCMUT_SPSO.
- **Description:** Nhân viên của hệ thống thực hiện việc quản lý tài khoản của người dùng.
- **Trigger:** Nhân viên bấm vào nút “Quản lý tài khoản” trên thanh điều hướng.

- **Pre-conditions:**

- Người dùng phải có chức vụ là nhân viên và đã đăng nhập vào hệ thống.
 - Thiết bị của người dùng phải được kết nối internet đầy đủ.

- **Post-conditions:**

- Nhân viên có thể thực hiện thành công việc thêm, xóa, sửa tài khoản cũng như xem lịch sử in của tất cả người dùng.

- **Normal Flow:**

1. Sau khi nhân viên bấm vào nút “Quản lý tài khoản” trên thanh điều hướng, một giao diện gồm các tài khoản người dùng sẽ được hiển thị.
2. Tại giao diện này, nhân viên có thể:
 - Bấm vào nút *Xóa* để xóa tài khoản được chọn.
 - Bấm vào nút *Sửa* để chỉnh sửa thông tin của tài khoản.
 - Bấm vào nút *Lịch sử in* để xem lịch sử in của tài khoản.

- **Alternative Flow:** Không có.

- **Exceptions Flow:** Không có.

Use Case: Manage Printer

- **Actors:** HCMUT_SPSO.
- **Description:** Nhân viên của hệ thống thực hiện việc quản lý các máy in.
- **Trigger:** Nhân viên bấm vào nút “Quản lý máy in” trên thanh điều hướng.

- **Pre-conditions:**

- Người dùng phải có chức vụ là nhân viên và đã đăng nhập vào hệ thống.

- Thiết bị của người dùng phải được kết nối internet đầy đủ.

- **Post-conditions:**

- Nhân viên có thể thực hiện thành công việc thêm, xóa, chỉnh sửa trạng thái của máy in, xem lịch sử in của các máy in.

- **Normal Flow:**

1. Sau khi nhân viên bấm vào nút “Quản lý máy in” trên thanh điều hướng, một giao diện danh sách máy in sẽ được hiển thị.
2. Khi nhân viên bấm vào nút *Thêm máy in*, một giao diện sẽ được hiển thị để nhập thông tin máy in mới.
3. Khi nhân viên bấm vào nút *Xóa máy in*, hệ thống sẽ xóa máy in tương ứng.
4. Khi nhân viên chọn vào nút *Enable* hoặc *Disable*, hệ thống sẽ thực hiện việc bật hoặc tắt máy in.
5. Khi nhân viên chọn vào nút *Xem lịch sử*, hệ thống sẽ hiển thị lịch sử in của máy in.

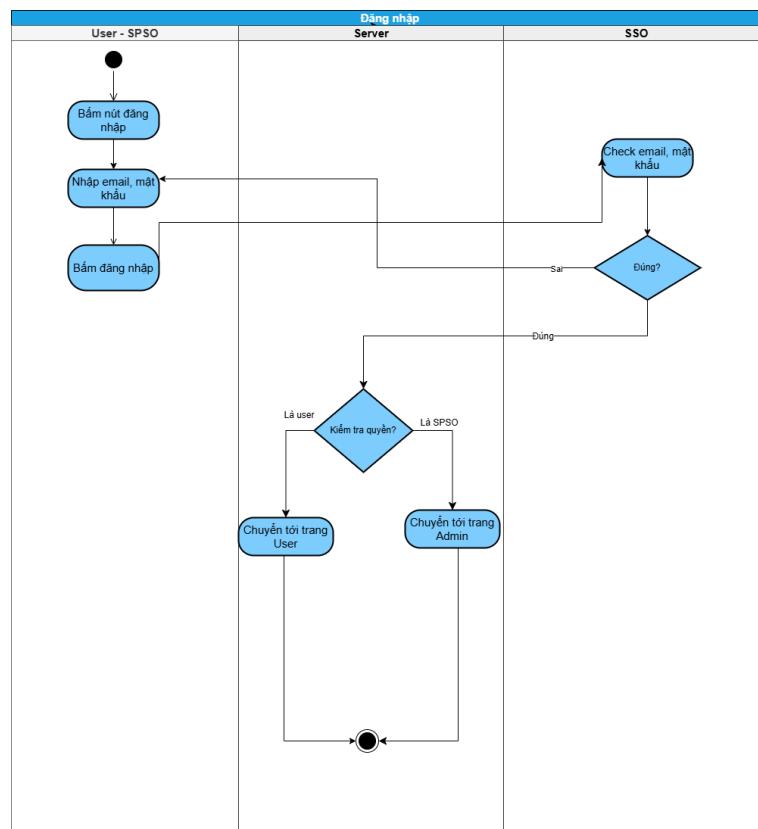
- **Alternative Flow:** Không có.

- **Exceptions Flow:** Không có.

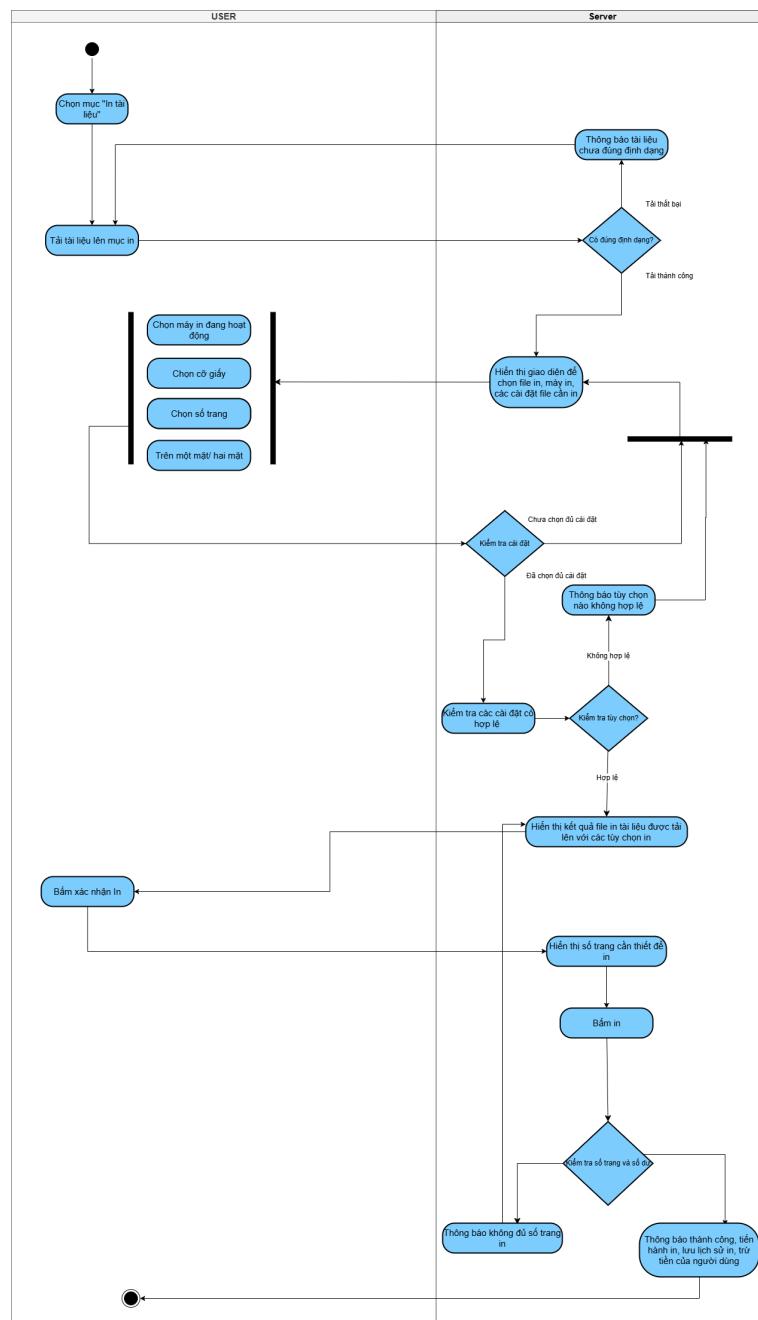
2 Phần 2

Task 2: System Modelling

2.1 Vẽ Activity Diagram cho mô hình

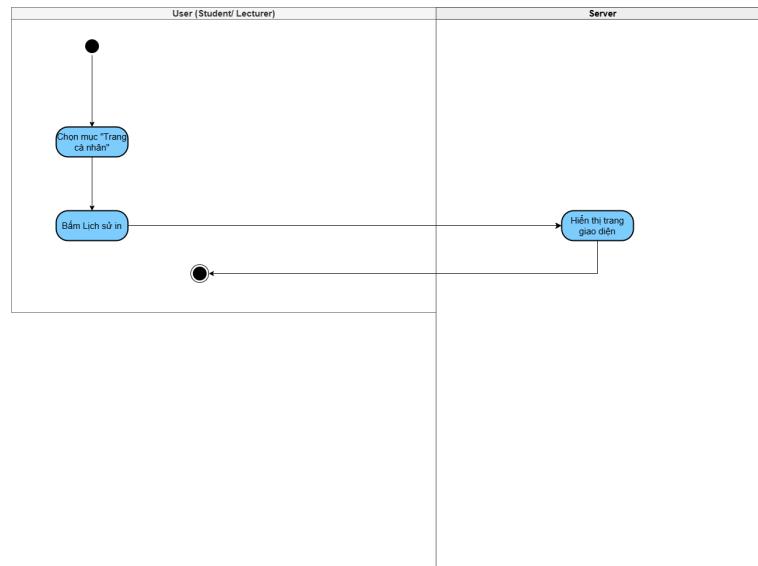


- Activity diagram biểu diễn chức năng đăng nhập. Quá trình bắt đầu khi người dùng hoặc nhân viên bấm nút đăng nhập. Người dùng nhập email, mật khẩu và bấm nút đăng nhập. Khi này hệ thống SSO sẽ kiểm tra tài khoản mật khẩu có hợp lệ hay không. Nếu thông tin đăng nhập là đúng thì hệ thống sẽ tiến hành kiểm tra quyền của người dùng là Admin hay User và chuyển tới trang tương ứng, quá trình kết thúc. Nếu thông tin sai thì sẽ trở lại trang đăng nhập.

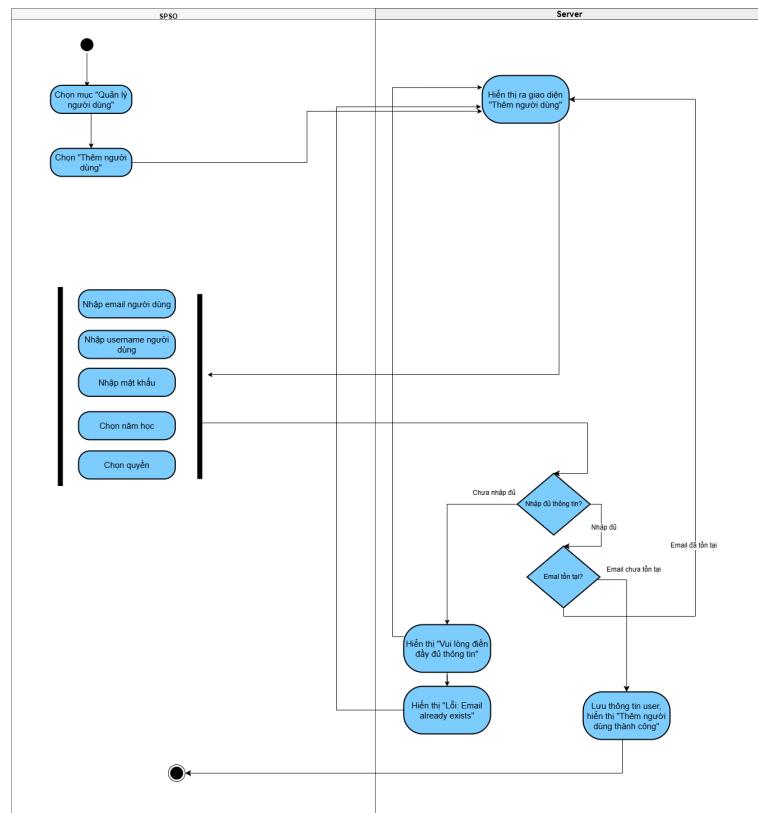


- Activity diagram biểu diễn chức năng in ấn của người sử dụng dịch vụ in. Quá trình bắt đầu khi người dùng ấn vào mục “IN NGAY, sau đó dùng sẽ tải tài liệu lên hệ thống in ấn. Nếu tải lên thất bại thì hệ thống sẽ trả về thông báo tải tệp thất bại cho người dùng. Nếu việc tải tệp diễn ra thành công thì một giao diện để tùy chọn việc in ấn sẽ xuất hiện. Lúc này người dùng tiến hành tùy chọn cho việc in ấn. Sau đó hệ thống sẽ kiểm tra xem người dùng có điền đủ các tùy chọn cần thiết cho việc in chưa. Nếu chưa thì hiện thông báo chưa chọn đủ cho người dùng và trở lại bước nhập tùy chọn. Nếu đã đủ thì tiếp tục kiểm tra tính đúng của dữ liệu người dùng nhập vào (vd: trang in phải là số, trang in không được quá số trang tối đa của file, . . .). Nếu có dữ liệu không hợp lệ thì trả về thông báo dữ liệu không hợp lệ cho người dùng và quay lại bước nhập tùy chọn in. Nếu dữ liệu hợp lệ thì hệ thống thực hiện tính toán số

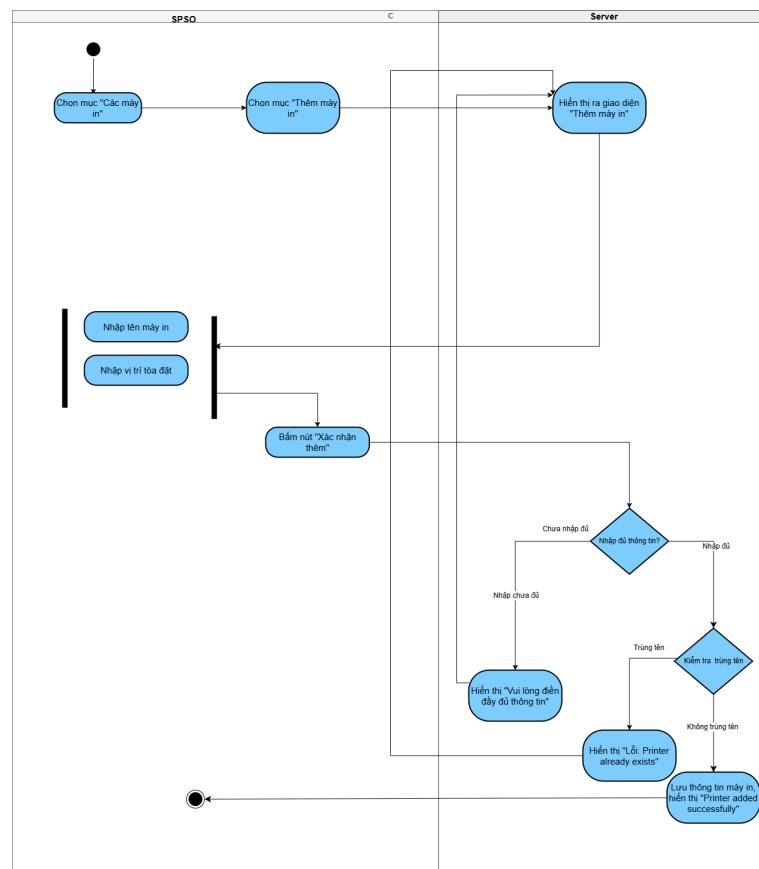
trang in cần thiết và hiển thị cho người dùng. Sau đó người dùng bấm in, lúc này hệ thống sẽ kiểm tra số dư của người dùng có đủ hay không. Nếu không đủ thì trả về thông báo “Số dư không đủ”. Nếu tài khoản của người dùng còn đủ thì thực hiện gửi yêu cầu in, kết thúc quá trình.



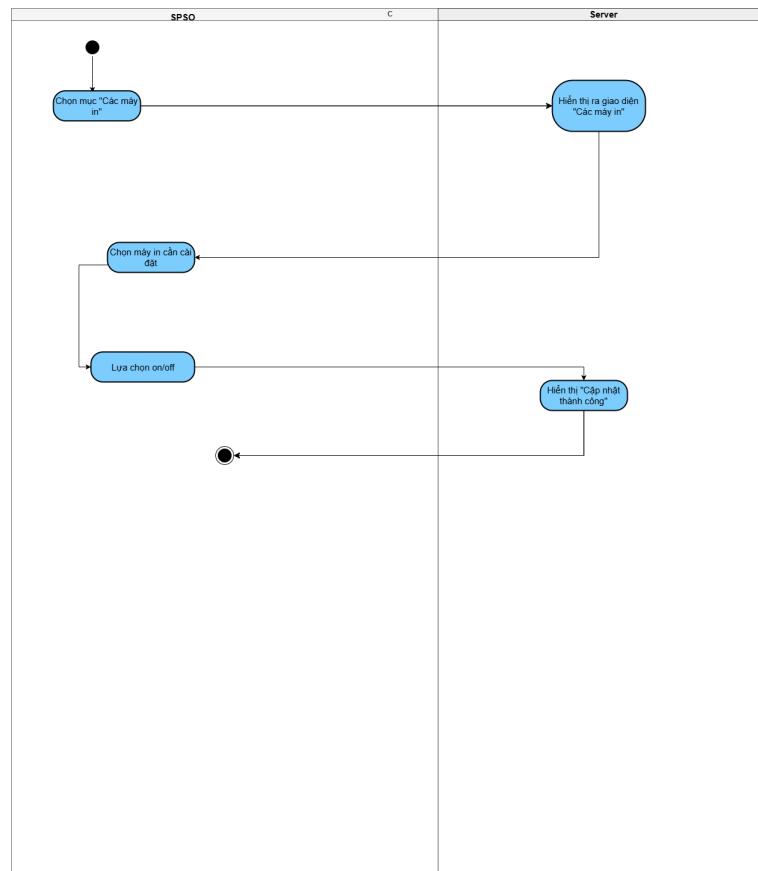
- Activity diagram biểu diễn chức năng xem lịch sử in của người sử dụng dịch vụ in ấn. Đầu tiên, người dùng chọn mục “Trang cá nhân”. Sau đó, bấm vào lịch sử in. Lúc này, server sẽ trả về một giao diện chứa lịch sử in cho người dùng. Kết thúc quá trình.



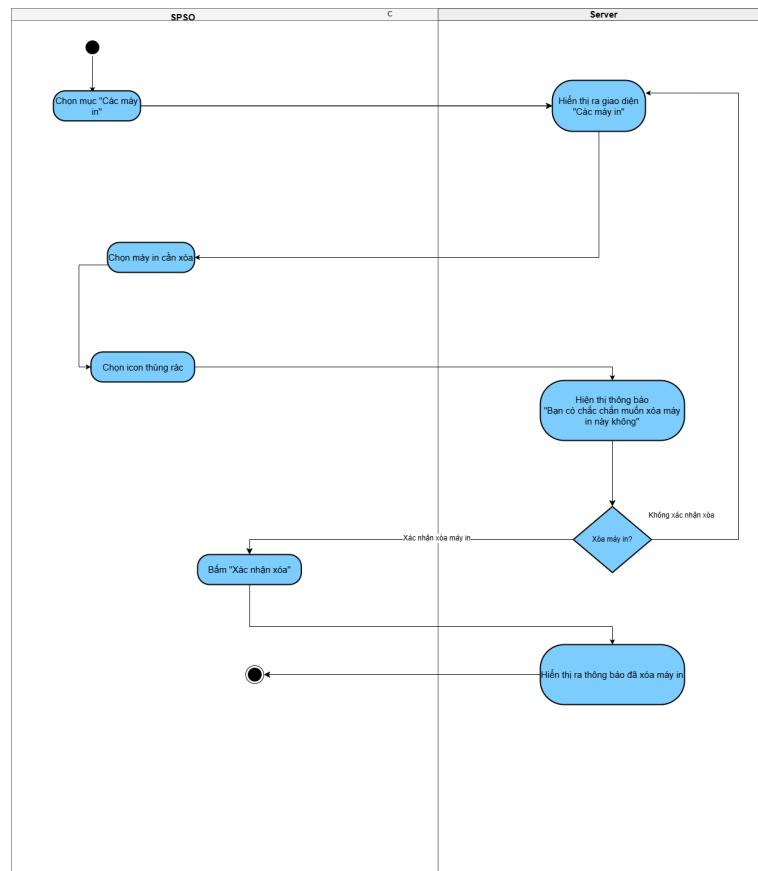
- Activity diagram biểu diễn chức năng thêm tài khoản người dùng của nhân viên. Đầu tiên, nhân viên chọn mục quản lý người dùng. Nhân viên bấm nút thêm người dùng. Lúc này, server sẽ trả về một giao diện “Thêm người dùng”. Nhân viên lần lượt nhập các thông tin của người dùng mới. Sau khi nhập xong, server sẽ kiểm tra xem người dùng đã nhập đủ thông tin hay chưa. Nếu chưa thì trả về thông báo “Vui lòng nhập đủ thông tin” và server hiển thị lại trang “Thêm người dùng”. Nếu đã đủ thì kiểm tra email nhập vào có tồn tại hay chưa. Nếu đã tồn tại thì hiển thị lỗi: “Email already exist”, server hiển thị lại trang “Thêm người dùng”. Nếu chưa tồn tại thì lưu thông tin người dùng mới vào hệ thống và hiển thị thông báo “Thêm người dùng thành công”, kết thúc quá trình.



- Activity diagram thực hiện chức năng thêm máy in của nhân viên. Đầu tiên, chọn mục các máy in. Sau đó chọn mục thêm máy in. Lúc này server sẽ hiển thị một giao diện “Thêm máy in” cho nhân viên. Nhân viên tiến hành nhập các thông tin của máy in mới. Sau đó server sẽ kiểm tra xem nhân viên đã điền đầy đủ các thông tin của máy in chưa, nếu chưa đủ thì hiển thị thông báo “Vui lòng điền đầy đủ thông tin”, server hiển thị lại trang “Thêm máy in”. Nếu đã đủ thì kiểm tra xem thông tin của máy in mới có trùng lặp với các máy in cũ hay không. Nếu có thì hiển thị lỗi “Printer already exist”, server hiển thị lại trang “Thêm máy in”. Nếu không thì lưu thông tin máy in mới, hiển thị thông báo lưu máy in thành công, quá trình kết thúc.



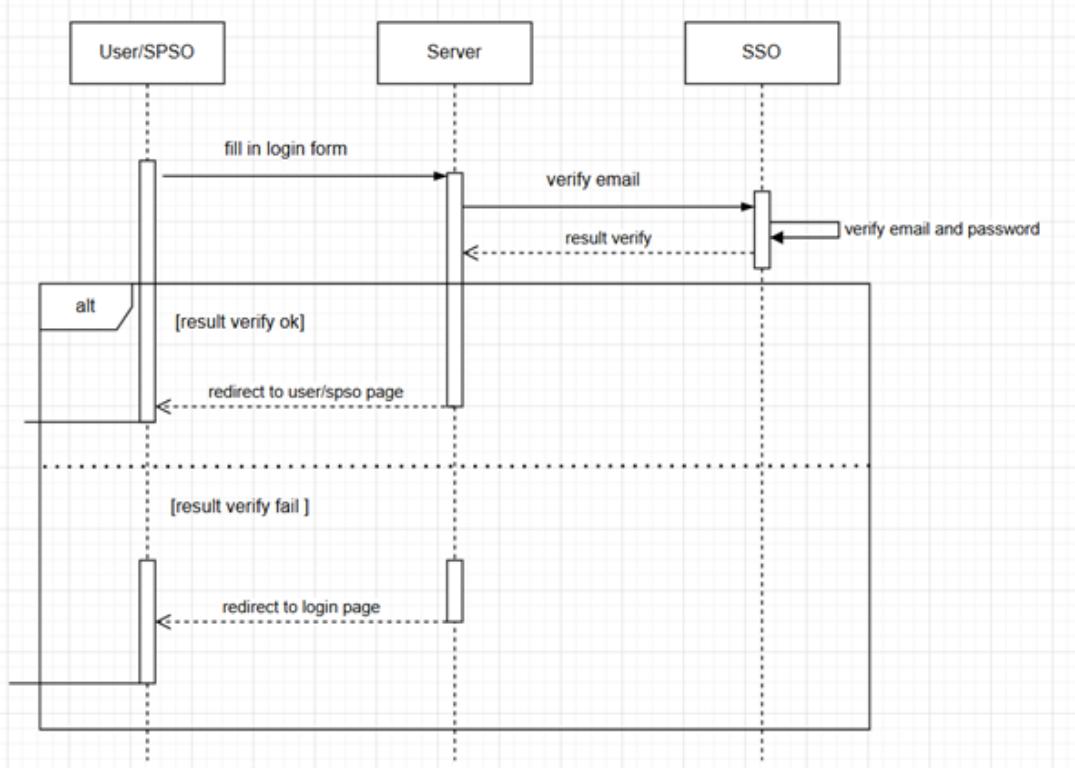
- Activity Diagram cho chức năng kích hoạt hoặc vô hiệu hóa máy in của nhân viên. Đầu tiên nhân viên chọn mục “Các máy in”. Lúc này server sẽ hiển thi giao diện chứa danh sách các máy in. Nhân viên chọn máy in cần cài đặt, lựa chọn on/off, nhân viên xác nhận và server trả về thông báo “Cập nhật thành công”. Kết thúc quá trình



- Activity diagram cho chức năng xóa máy in của nhân viên. Đầu tiên nhân viên chọn mục “Các máy in”. Lúc này server sẽ hiển thi giao diện chứa danh sách các máy in. Nhân viên chọn icon thùng rác trùng hàng với máy in cần xóa. Lúc này, server sẽ hiển thi thông báo “Bạn có chắc chắn hay không” nhằm xác nhận. Nếu không xác nhận thì quay lại trang giao diện các máy in. Nếu xác nhận thì xóa máy in thì hiển thi thông báo “Đã xóa máy in” và kết thúc quá trình.

2.2 Vẽ Sequence Diagram cho mô hình

Sequence Diagram: User Login through SSO



- **Đối tượng tham gia:**

- **User/SPSO:** Đại diện cho người dùng đang thực hiện đăng nhập.
- **Server:** Là máy chủ của hệ thống, xử lý yêu cầu đăng nhập.
- **SSO:** Dịch vụ xác thực, chịu trách nhiệm xác minh thông tin đăng nhập của người dùng.

- **Phân tích trình tự:**

1. Người dùng điền thông tin đăng nhập (email và mật khẩu), sau đó gửi yêu cầu đăng nhập đến Server.
2. Server nhận yêu cầu và gửi thông tin email đến dịch vụ SSO để xác minh.
3. SSO kiểm tra thông tin email:
 - Nếu email tồn tại, tiếp tục xác minh mật khẩu.
 - Nếu email không tồn tại, xác minh thất bại.
4. SSO gửi kết quả xác minh (thành công hoặc thất bại) về cho Server.

5. Server xử lý kết quả xác minh:

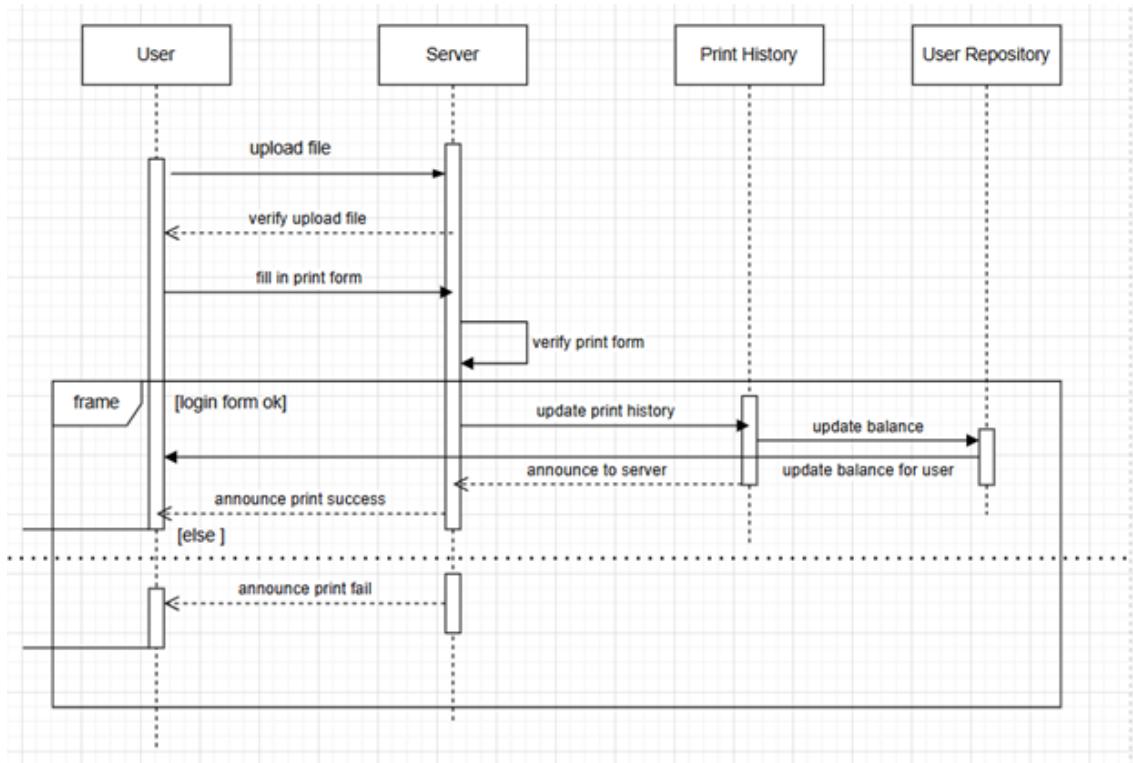
– Nếu xác minh thành công:

- * Gửi thông báo cho người dùng rằng đăng nhập thành công.
 - * Chuyển hướng người dùng đến trang chủ thích hợp dựa trên vai trò (*User* hoặc *SPSO*).

– Nếu xác minh thất bại:

- * Gửi thông báo cho người dùng rằng thông tin đăng nhập không đúng.
 - * Yêu cầu người dùng nhập lại thông tin đăng nhập.

Sequence Diagram: User Printing Process



- Đối tượng tham gia:

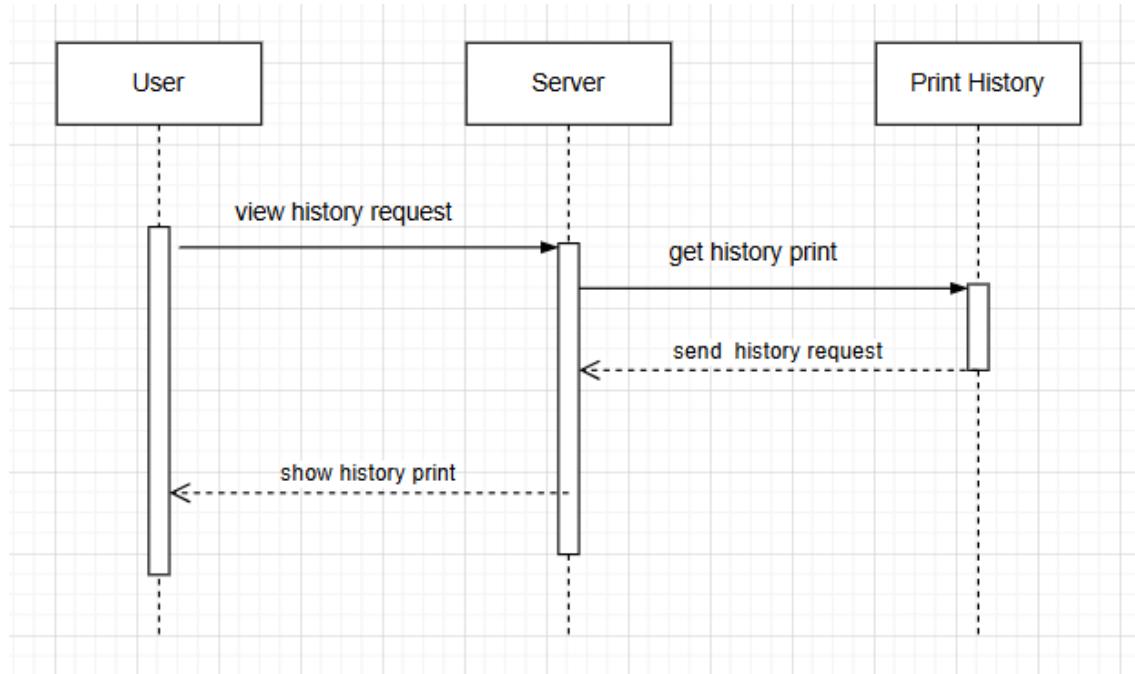
- **User:** Đại diện cho người dùng thực hiện yêu cầu in.
 - **Server:** Là máy chủ, xử lý các yêu cầu của người dùng và thực hiện các tác vụ in ấn.
 - **PrintHistory:** Lưu trữ lịch sử các bản in của người dùng.
 - **UserRepository:** Lưu trữ thông tin cá nhân và các thông tin liên quan đến người dùng.



• **Phân tích trình tự:**

1. Người dùng chọn tệp cần in và gửi yêu cầu tải tệp lên **Server**.
2. **Server** nhận được tệp và kiểm tra tính hợp lệ của tệp.
3. Sau khi tệp được xác nhận, **Server** trả thông báo cho người dùng, yêu cầu điền các thông tin liên quan đến việc in (số lượng bản in, loại giấy,...).
4. Người dùng nhập thông tin in và gửi lại yêu cầu đến **Server**.
5. **Server** kiểm tra tính hợp lệ của thông tin in:
 - (a) **Trường hợp thông tin hợp lệ:**
 - **Server** cập nhật **PrintHistory**, bao gồm các thông tin: tên tệp, thời gian in, số lượng bản in,...
 - **Server** tính toán số tiền cần thanh toán và cập nhật số dư tài khoản của người dùng trong **UserRepository**.
 - **Server** thông báo cho người dùng rằng việc in đã hoàn tất và cập nhật lại giao diện người dùng.
 - (b) **Trường hợp thông tin không hợp lệ:**
 - **Server** thông báo lỗi cho người dùng, chỉ rõ rằng thông tin điền vào form in không hợp lệ.

Sequence Diagram: View Print History



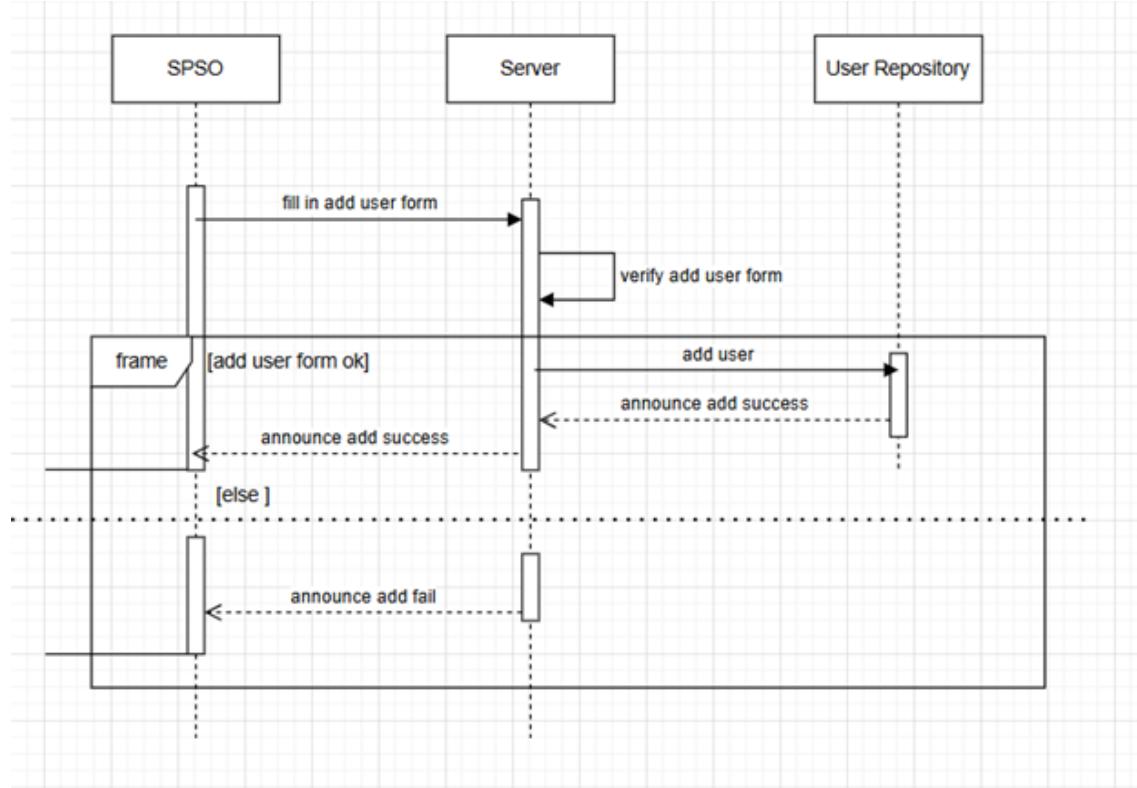
- **Đối tượng tham gia:**

- **User:** Đại diện cho người dùng cuối cùng, người thực hiện yêu cầu xem lịch sử in.
- **Server:** Máy chủ ứng dụng, xử lý yêu cầu của người dùng và truy xuất dữ liệu từ hệ thống lưu trữ.
- **Print History:** Đại diện cho hệ thống lưu trữ lịch sử in của tất cả người dùng.

- **Phân tích trình tự:**

1. Người dùng gửi yêu cầu đến **Server**, biểu thị mong muốn xem lịch sử các lần in đã thực hiện.
2. **Server** nhận yêu cầu và truy cập vào **Print History** để lấy thông tin liên quan đến các lần in của người dùng.
3. **Print History** trả về thông tin lịch sử in cho **Server**.
4. **Server** xử lý dữ liệu và hiển thị thông tin về lịch sử in cho người dùng.

Sequence Diagram: Add New User



- **Đối tượng tham gia:**

- **SPSO:** Đại diện cho nhân viên, người thực hiện việc thêm người dùng.
- **Server:** Máy chủ ứng dụng, xử lý yêu cầu của nhân viên và thực hiện các tác vụ thêm người dùng.
- **User Repository:** Kho lưu trữ thông tin của tất cả người dùng trong hệ thống.

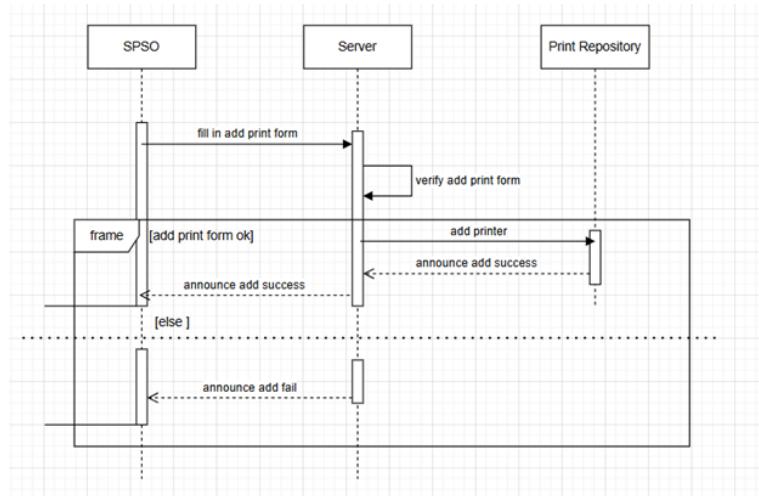
- **Phân tích trình tự:**

1. Nhân viên nhập thông tin của người dùng mới vào form thêm người dùng.
2. **Server** nhận thông tin và tiến hành kiểm tra tính hợp lệ của các thông tin đã nhập (ví dụ: định dạng email, trùng lặp thông tin).
3. Nếu thông tin hợp lệ:
 - **Server** gửi yêu cầu thêm người dùng mới vào **User Repository**.
 - **User Repository** xác nhận việc thêm người dùng và trả kết quả về cho **Server**.
 - **Server** thông báo cho nhân viên rằng quá trình thêm người dùng đã thành công.

4. Nếu thông tin không hợp lệ:

- **Server** phát hiện lỗi (ví dụ: định dạng sai, thông tin trùng lặp) và thông báo lỗi về cho nhân viên.

Sequence Diagram: Add New Printer



- **Đối tượng tham gia:**

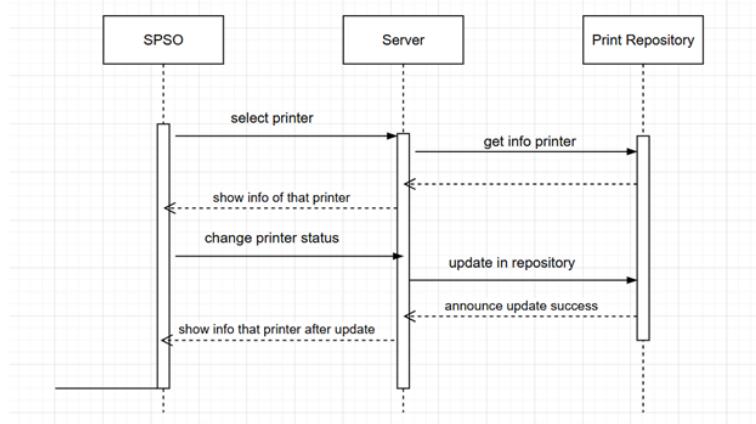
- **SPSO:** Đại diện cho nhân viên, người thực hiện việc thêm máy in.
- **Server:** Máy chủ ứng dụng, xử lý yêu cầu của nhân viên và thực hiện các tác vụ thêm máy in.
- **Print Repository:** Kho lưu trữ thông tin của tất cả máy in trong hệ thống.

- **Phân tích trình tự:**

1. Nhân viên nhập thông tin của máy in mới vào *form* thêm máy in.
2. **Server** nhận thông tin và tiến hành kiểm tra tính hợp lệ của các thông tin đã nhập (ví dụ: tên máy in, IP máy in, thông tin kết nối mạng).
3. Nếu thông tin hợp lệ:
 - **Server** gửi yêu cầu thêm máy in mới vào **Print Repository**.
 - **Print Repository** xác nhận việc thêm máy in và trả kết quả về cho **Server**.
 - **Server** thông báo cho nhân viên rằng quá trình thêm máy in đã thành công.
4. Nếu thông tin không hợp lệ:

- **Server** phát hiện lỗi (ví dụ: thông tin trùng lặp, thiếu thông tin) và thông báo lỗi về cho nhân viên.

Sequence Diagram: Update Printer's State



- **Đối tượng tham gia:**

- **SPSO:** Đại diện cho nhân viên, người tương tác trực tiếp với hệ thống để chọn và cập nhật trạng thái máy in.
- **Server:** Máy chủ ứng dụng, xử lý các yêu cầu của nhân viên, truy xuất và cập nhật dữ liệu trong cơ sở dữ liệu.
- **Printer Repository:** Là nơi lưu trữ thông tin của tất cả máy in trong hệ thống.

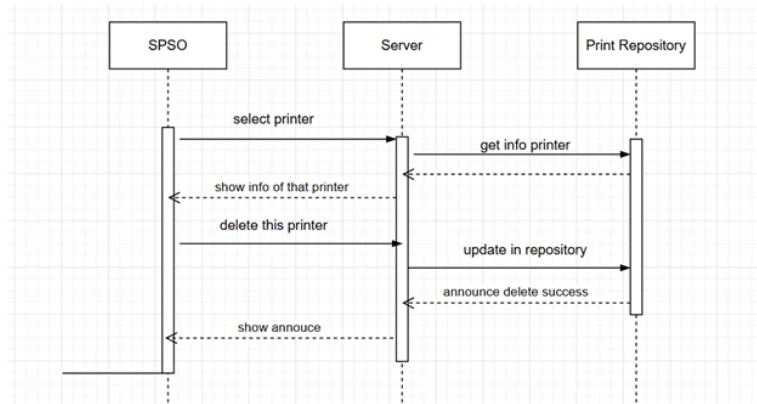
- **Phân tích trình tự:**

1. Nhân viên chọn một máy in từ danh sách các máy in có sẵn trong hệ thống. Yêu cầu này được gửi đến máy chủ.
2. **Server** nhận được yêu cầu và truy xuất cơ sở dữ liệu để lấy thông tin chi tiết về máy in mà nhân viên đã chọn.
3. **Server** gửi thông tin chi tiết của máy in (ví dụ: tên máy in, trạng thái hiện tại,...) về cho nhân viên để nhân viên xác nhận.
4. Sau khi xem thông tin, nhân viên quyết định thay đổi trạng thái của máy in (ví dụ: từ trạng thái sẵn sàng sang trạng thái bận).
5. **Server** nhận được yêu cầu thay đổi trạng thái và cập nhật thông tin mới vào **Printer Repository**.
6. **Printer Repository** xác nhận rằng việc cập nhật trạng thái đã thành công và gửi thông báo

lại cho **Server**.

7. **Server** thông báo cho nhân viên rằng việc cập nhật trạng thái đã được thực hiện thành công và hiện thông tin sau khi cập nhật của máy in.

Sequence Diagram: Delete Printer



• Đối tượng tham gia:

- **SPSO:** Đại diện cho nhân viên, người tương tác trực tiếp với hệ thống để chọn và xóa máy in.
- **Server:** Máy chủ ứng dụng, xử lý các yêu cầu của nhân viên, truy xuất và cập nhật dữ liệu trong cơ sở dữ liệu.
- **Printer Repository:** Là nơi lưu trữ thông tin của tất cả máy in trong hệ thống.

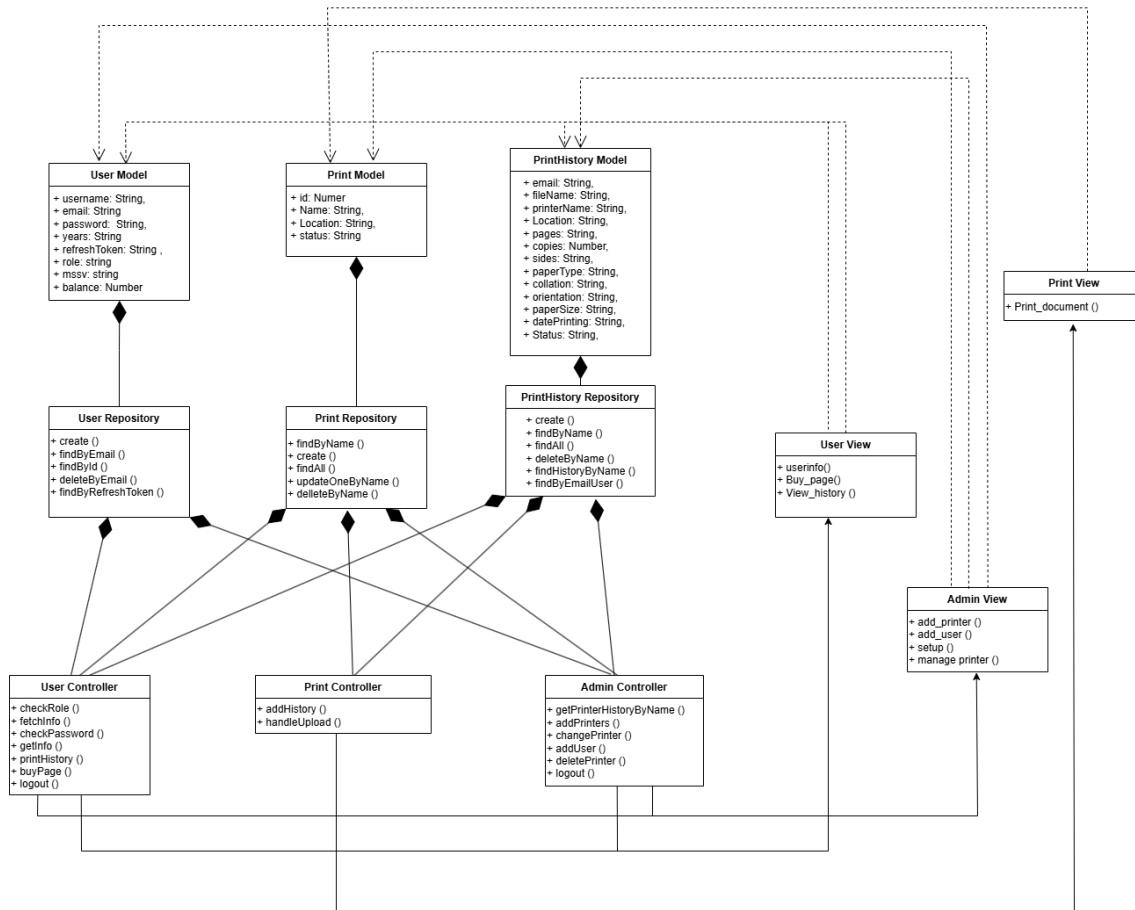
• Phân tích trình tự:

1. Nhân viên chọn một máy in từ danh sách các máy in có sẵn trong hệ thống. Yêu cầu này được gửi đến máy chủ.
2. **Server** nhận được yêu cầu và truy xuất cơ sở dữ liệu để lấy thông tin chi tiết về máy in mà nhân viên đã chọn.
3. **Server** gửi thông tin chi tiết của máy in (ví dụ: tên máy in, trạng thái hiện tại,...) về cho nhân viên để nhân viên xác nhận lại trước khi xóa.
4. Sau khi xem thông tin, nhân viên xác nhận rằng họ muốn xóa máy in đã chọn.
5. **Server** nhận được xác nhận từ nhân viên và tiến hành xóa thông tin của máy in đó khỏi **Printer Repository**.
6. **Printer Repository** xác nhận rằng việc xóa đã được thực hiện thành công và gửi thông báo

lại cho Server.

7. Server thông báo cho nhân viên rằng việc xóa máy in đã được thực hiện thành công.

2.3 Vẽ Class Diagram cho mô hình



Class Diagram: MVC Design Architecture

- Mô hình MVC gồm 4 thành phần chính: Model, Repository, Controller, và View.

- Model:**

- Các lớp như User, Print, PrintHistory định nghĩa cấu trúc dữ liệu và thông tin cần lưu trữ.

- Repository:**

- Chịu trách nhiệm giao tiếp với cơ sở dữ liệu.
- Cung cấp các chức năng truy vấn, thêm, sửa, và xóa dữ liệu.

- Controller:**

- Đóng vai trò xử lý logic nghiệp vụ.
- Tiếp nhận yêu cầu từ người dùng thông qua các giao diện (View).
- Tương tác với Repository để lấy hoặc cập nhật dữ liệu.

- **View:**

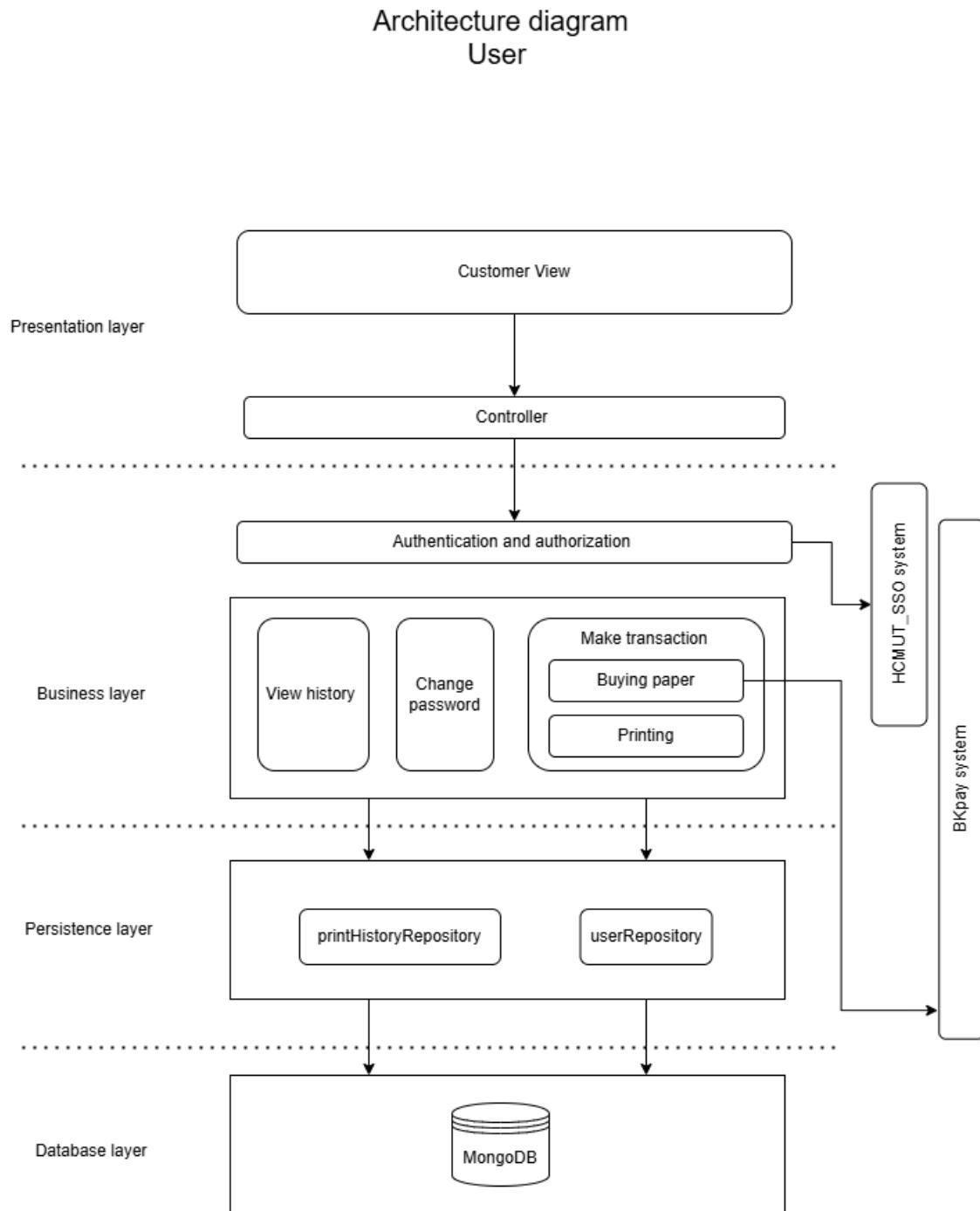
- Lớp giao diện, cho phép người dùng:
 - * Thao tác với hệ thống.
 - * Xem thông tin.
 - * Gửi yêu cầu và nhận phản hồi từ hệ thống.

- **Ưu điểm của kiến trúc:**

- Tách bạch rõ ràng giữa các tầng dữ liệu, logic, và hiển thị.
- Dễ bảo vệ hệ thống, dễ dàng bảo trì, mở rộng, và nâng cấp trong tương lai.

3 Architecture design

3.1 Architectural Diagram: User



Presentation layer

Đây là giao diện người dùng (Customer View) mà người dùng tương tác trực tiếp. Lớp trình bày chịu trách nhiệm hiển thị thông tin và nhận các yêu cầu từ người dùng.

Controller

Lớp này điều khiển luồng thực thi của ứng dụng. Nó xử lý các yêu cầu từ giao diện người dùng và truyền dữ liệu đến lớp nghiệp vụ, đồng thời nhận phản hồi từ lớp nghiệp vụ để cập nhật giao diện.

Business layer

Chứa các logic nghiệp vụ của hệ thống, bao gồm các chức năng:

- Xem lịch sử in.
- Đổi mật khẩu.
- Thực hiện giao dịch.
- In ấn.

Lớp nghiệp vụ cũng tương tác với các hệ thống bên ngoài như HCMUT SSO và BKpay để xác thực và thực hiện thanh toán.

Persistence layer

Lớp lưu trữ chịu trách nhiệm lưu trữ dữ liệu của ứng dụng vào cơ sở dữ liệu MongoDB. Dữ liệu bao gồm thông tin người dùng và lịch sử in ấn.

Các chức năng chính của hệ thống

Xác thực và ủy quyền

Hệ thống kiểm tra thông tin đăng nhập của người dùng và xác định quyền truy cập dựa trên thông tin xác thực.

Quản lý người dùng

Cho phép người dùng thực hiện các chức năng như:

- Xem thông tin cá nhân.
- Đổi mật khẩu.

Quản lý lịch sử in

Hệ thống lưu trữ và hiển thị lịch sử các bản in của người dùng.

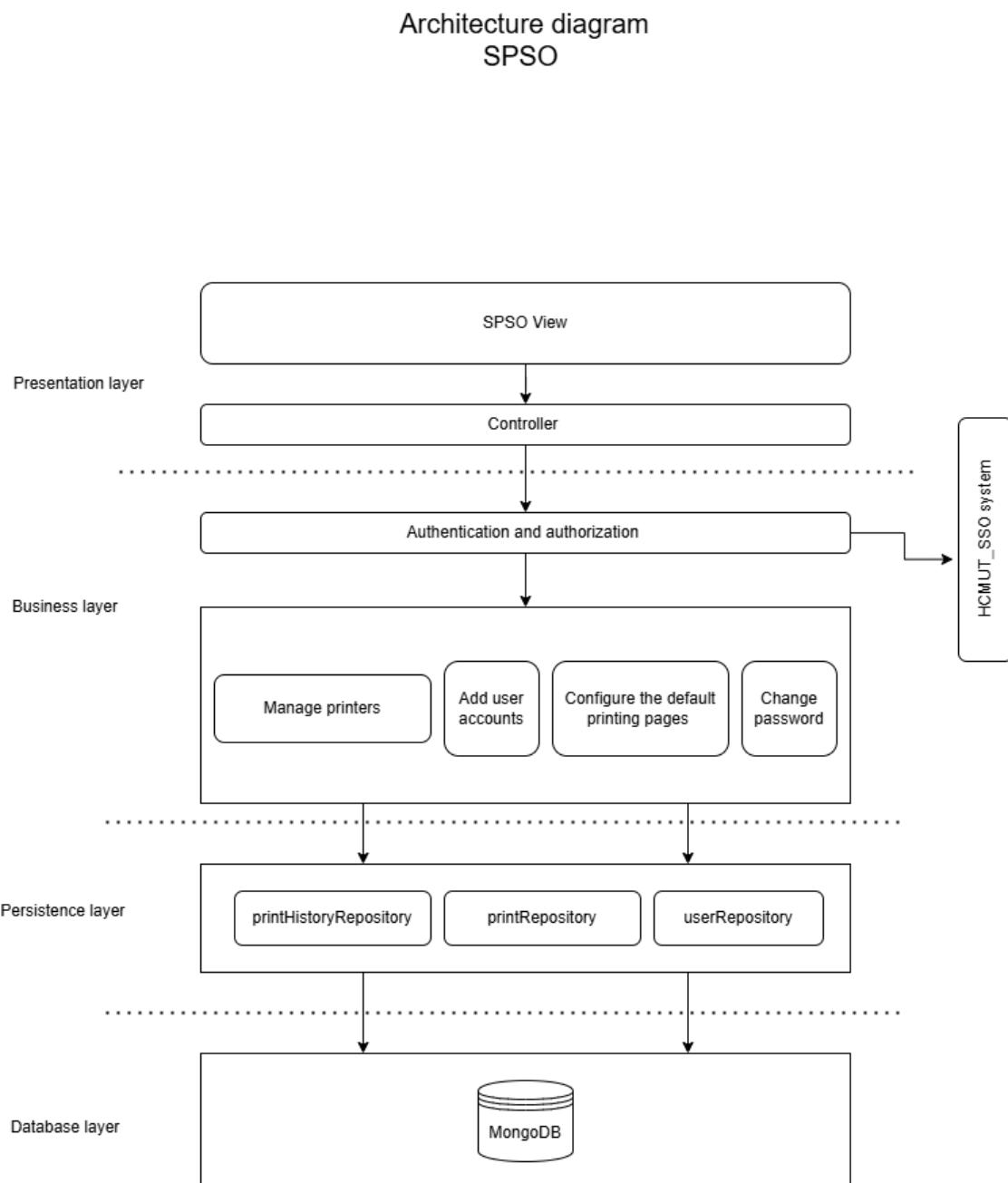
Thanh toán

Hệ thống kết nối với BKpay để thực hiện các giao dịch mua. Người dùng có thể thanh toán trực tuyến thông qua tích hợp này.

In ấn

Hệ thống gửi yêu cầu in đến máy in được chọn và quản lý quá trình in để đảm bảo hoàn thành yêu cầu của người dùng.

3.2 Architectural Diagram: SPSO



Presentation layer

- **SPSO View:** Giao diện người dùng trực tiếp mà SPSO tương tác. Đây là nơi người dùng thực hiện các thao tác như xem lịch sử in, đổi mật khẩu, quản lý máy in...

Controller

Lớp này điều khiển luồng thực thi của ứng dụng. Nó xử lý các yêu cầu từ giao diện người dùng và truyền dữ liệu đến lớp nghiệp vụ, đồng thời nhận phản hồi từ lớp nghiệp vụ để cập nhật giao diện.

- **Controller:** Tiếp nhận các yêu cầu từ giao diện người dùng, xử lý các yêu cầu đó và điều phối các hoạt động của hệ thống. Ví dụ: khi người dùng muốn đổi mật khẩu, Controller sẽ nhận yêu cầu này, kiểm tra tính hợp lệ của mật khẩu mới và gọi đến lớp nghiệp vụ để cập nhật thông tin người dùng.

Business layer

Chứa các logic nghiệp vụ của hệ thống, bao gồm các chức năng:

- **Authentication and authorization:** Xác thực danh tính người dùng và kiểm tra quyền truy cập của họ. Lớp này thường tương tác với hệ thống xác thực bên ngoài như HCMUT SSO.
- **Manage printers, Add users, Configure the default printing pages, Change password:** Các module thực hiện các chức năng nghiệp vụ chính của hệ thống như quản lý máy in, thêm người dùng, cấu hình mặc định và đổi mật khẩu.

Persistence layer

Lớp lưu trữ chịu trách nhiệm lưu trữ dữ liệu của ứng dụng vào cơ sở dữ liệu MongoDB. Dữ liệu bao gồm thông tin người dùng và lịch sử in ấn.

- **printHistoryRepository, printRepository, userRepository:** Các lớp này chịu trách nhiệm tương tác với cơ sở dữ liệu để lưu trữ và truy xuất dữ liệu. Chúng cung cấp các phương thức để thêm, sửa, xóa và tìm kiếm dữ liệu về lịch sử in, thông tin máy in và thông tin người dùng.

Lớp cơ sở dữ liệu (Database layer)

- **MongoDB:** Cơ sở dữ liệu được sử dụng để lưu trữ dữ liệu của hệ thống. MongoDB là một cơ sở dữ liệu NoSQL, linh hoạt và phù hợp với việc lưu trữ dữ liệu có cấu trúc không cố định.

Các chức năng chính của hệ thống

Xác thực và ủy quyền

Hệ thống kiểm tra thông tin đăng nhập của người dùng và xác định quyền truy cập dựa trên thông tin xác thực.

Quản lý người dùng

Cho phép người dùng thực hiện các chức năng như:

- Xem thông tin cá nhân.

- Đổi mật khẩu.

Quản lý lịch sử in

Hệ thống lưu trữ và hiển thị lịch sử các bản in của người dùng.

3.2.1 Presentation Strategy

Chiến lược mà chúng tôi hướng tới cho trang web dịch vụ in thông minh dành cho sinh viên HCMUT sẽ tập trung vào việc tạo ra một giao diện thân thiện và dễ sử dụng nhằm tối ưu hóa trải nghiệm người dùng và đáp ứng nhu cầu in ấn của sinh viên một cách hiệu quả nhất có thể. Giao diện trang web sẽ được thiết kế với bố cục rõ ràng, sử dụng các biểu tượng và hướng dẫn trực quan để giúp sinh viên dễ dàng thực hiện các thao tác từ đăng nhập, tải lên tài liệu đến lựa chọn máy in và cấu hình các thông số in ấn như kích thước giấy, số lượng bản in, in một mặt hay hai mặt. Trang web giúp hiển thị thông tin về các máy in sẵn có trên khuôn viên, bao gồm thông tin về vị trí cụ thể (tòa nhà, phòng), mã máy in và trạng thái hoạt động của từng máy đó, giúp người dùng chọn máy in gần và phù hợp nhất một cách dễ dàng.

Khi bắt đầu sử dụng dịch vụ, sinh viên sẽ được hướng dẫn chi tiết cách đăng nhập qua hệ thống xác thực HCMUT_SSO và cách thao tác trên các chức năng chính của hệ thống. Đặc biệt, tính năng theo dõi lịch sử in ấn của sinh viên sẽ được thiết kế để có thể dễ dàng xem lại các tài liệu đã in cùng với số trang in theo từng kích thước giấy, giúp sinh viên quản lý hiệu quả số trang in còn lại trong tài khoản của mình. Mỗi học kỳ, trường sẽ cung cấp số trang in A4 mặc định miễn phí cho mỗi sinh viên. Trong trường hợp cần thêm số lượng trang in, sinh viên có thể sử dụng tính năng “Mua thêm trang in” và thanh toán qua các phương thức tích hợp như BKPay, tạo sự thuận tiện và linh hoạt cho nhu cầu in ấn của từng cá nhân.

Trang web cũng cung cấp một số tính năng quản lý hữu ích dành riêng cho nhân viên quản lý dịch vụ in ấn (SPSO). Nhân viên quản lý có thể dễ dàng thêm mới, kích hoạt, vô hiệu hóa hoặc cập nhật thông tin các máy in trên hệ thống, giúp duy trì tính ổn định và sẵn sàng phục vụ của dịch vụ.Thêm vào đó, SPSO có thể thiết lập các tùy chọn cấu hình như số trang in mặc định mỗi học kỳ, ngày cấp số trang miễn phí, loại file được chấp nhận khi tải lên, đáp ứng nhu cầu vận hành linh hoạt của hệ thống. Tất cả các hoạt động in ấn sẽ được ghi lại chi tiết trong hệ thống, giúp nhân viên quản lý có thể xem xét, đánh giá tình hình sử dụng dịch vụ in ấn của sinh viên thông qua các báo cáo hàng tháng và hàng năm, từ đó có những điều chỉnh hợp lý.

Tất cả người dùng hệ thống đều được yêu cầu xác thực qua HCMUT_SSO trước khi có thể sử dụng dịch vụ, đảm bảo tính bảo mật và quyền riêng tư cho từng tài khoản cá nhân.

Chúng tôi hướng đến sự trải nghiệm và quan tâm đến mọi góp ý từ phía người dùng, cố gắng để cải thiện thêm.

3.2.2 Presentation approach

Trong việc thiết kế giao diện cho trang web dịch vụ in thông tin HCMUT_SSPS, chúng tôi đã lựa chọn **NodeJS** làm ngôn ngữ lập trình cho backend của dự án. NodeJS là một nền tảng được xây dựng trên

“V8 Javascript engine” được viết bằng c++ và Javascript.

Lý do chọn NodeJS:

- **Mã nguồn mở:** Mã nguồn của Node.js được công bố công khai, điều này có nghĩa là bất kỳ ai cũng có thể truy cập, sử dụng, và đóng góp vào mã nguồn. NodeJS được duy trì bởi cộng đồng lập trình viên trên toàn thế giới, và hướng dẫn đóng góp của Node.js hướng dẫn bạn cách để bạn có thể đóng góp phần phát triển nó.
- **Đa nền tảng:** NodeJS không phụ thuộc vào bất kỳ hệ điều hành nào cụ thể nào, nghĩa là nó có thể chạy trên Linux, macOS hoặc Windows. Điều này làm cho NodeJS trở thành một lựa chọn linh hoạt cho các nhà phát triển muốn xây dựng các ứng dụng có thể hoạt động trên nhiều nền tảng khác nhau mà không cần thay đổi mã nguồn.
- **Môi trường thực thi JavaScript:** Để mã JavaScript có thể được thực thi, nó cần một môi trường chạy phù hợp. Trong khi trình duyệt như Chrome và Firefox cung cấp một môi trường thực thi cho JavaScript, Node.js mở rộng khả năng này ra ngoài trình duyệt. Node.js cho phép chạy JavaScript trên máy chủ, hoặc trong bất kỳ môi trường máy tính nào khác, không chỉ trong trình duyệt.
- **Dựa trên V8 JavaScript Engine:** NodeJS được xây dựng dựa trên V8, động cơ JavaScript được phát triển bởi Google cho trình duyệt Chrome. Điều này giúp NodeJS có khả năng thực thi JavaScript nhanh và hiệu quả, đồng thời hỗ trợ các tính năng mới nhất của ngôn ngữ JavaScript.

Cách thức hoạt động của NodeJS:

- **Kiến trúc Non-blocking I/O và Event-Driven:** Node.js sử dụng một mô hình non-blocking I/O (input/output) và event-driven, nghĩa là các hoạt động như đọc file, truy vấn cơ sở dữ liệu, hoặc giao tiếp mạng được thực hiện mà không chặn tiến trình chính. Điều này cho phép xử lý nhiều yêu cầu cùng lúc mà không cần tạo nhiều luồng (thread), giúp giảm bớt chi phí liên quan đến quản lý luồng và tối ưu hóa hiệu suất.
- **V8 JavaScript Engine:** Node.js được xây dựng trên động cơ JavaScript V8 của Google Chrome, đây là một động cơ rất nhanh cho phép biên dịch mã JavaScript thành mã máy để thực thi trực tiếp trên phần cứng, làm tăng hiệu suất thực thi.
- **Single-Threaded:** Mặc dù Node.js hoạt động trên một luồng duy nhất cho logic ứng dụng của người dùng, nó vẫn sử dụng nhiều luồng ở tầng thấp hơn thông qua thư viện **libuv** để xử lý các hoạt động I/O. Tuy nhiên, những chi tiết này được ẩn giấu khỏi người dùng, giúp việc lập trình đơn giản hơn mà vẫn đảm bảo hiệu suất.
- **Event Loop:** Trái tim của Node.js là “event loop”. Đây là vòng lặp sự kiện mà ở đó Node.js tiếp tục lắng nghe sự kiện và thực hiện các hàm gọi lại khi một sự kiện được kích hoạt. Vòng lặp sự kiện cho phép Node.js xử lý hàng nghìn kết nối đồng thời mà không cần phải tạo ra chi phí quản lý luồng.

- **Trigger Callback:** Khi thao tác I/O hoàn tất, hệ điều hành thông báo cho Node.js, và Node.js sau đó thực thi hàm callback tương ứng để xử lý kết quả hoặc tiếp tục xử lý logic.
- **NPM (Node Package Manager):** NPM là hệ thống quản lý gói cho Node.js, cho phép các nhà phát triển dễ dàng chia sẻ và sử dụng mã nguồn từ nhau. NPM là một trong những kho lưu trữ mã nguồn mở lớn nhất thế giới và chứa hàng ngàn module có thể được tích hợp vào ứng dụng của bạn.

Kết luận:

Việc lựa chọn NodeJS làm ngôn ngữ lập trình chính cho hệ thống backend giúp chúng tôi xây dựng hệ thống backend linh hoạt, hiệu quả, nhanh chóng và dễ dàng mở rộng và bảo trì trong tương lai. Điều này đảm bảo dịch vụ in ấn thông minh HCMUT_SSPS có thể đáp ứng tốt công cụ cho lập trình viên để dễ sử dụng và quản lý. Cũng như tạo nền tảng vững chắc vững cho việc mở rộng hệ thống trong tương lai.

3.2.3 Data storage approach

Trong việc thiết kế kiến trúc cho trang web dịch vụ in thông minh HCMUT_SSPS, chúng tôi đã lựa chọn **MongoDB** làm cơ sở dữ liệu chính cho dự án. MongoDB là một hệ quản trị cơ sở dữ liệu NoSQL, sử dụng mô hình dữ liệu dạng tài liệu (document-oriented), cho phép lưu trữ dữ liệu dưới dạng JSON linh hoạt và dễ dàng mở rộng.

Lý do chọn MongoDB:

- **Tính linh hoạt của mô hình dữ liệu:** Do đặc tính schema-less, MongoDB cho phép chúng tôi dễ dàng điều chỉnh cấu trúc dữ liệu mà không cần phải thay đổi cấu trúc bảng như trong các cơ sở dữ liệu quan hệ. Điều này rất hữu ích trong quá trình phát triển và mở rộng các tính năng mới cho hệ thống.
- **Hiệu năng cao:** MongoDB được thiết kế để xử lý lượng lớn dữ liệu và có khả năng mở rộng tốt, phù hợp với nhu cầu lưu trữ và truy xuất dữ liệu nhanh chóng của hệ thống dịch vụ in ấn dành cho số lượng lớn sinh viên.
- **Khả năng mở rộng:** Hỗ trợ sharding và replication, MongoDB giúp hệ thống có thể mở rộng theo chiều ngang, đảm bảo tính sẵn sàng và độ tin cậy cao.

Cách thức lưu trữ dữ liệu trong hệ thống:

- **Thông tin người dùng:** Lưu trữ các thông tin xác thực từ HCMUT_SSO, lịch sử in ấn, số trang in còn lại, và các giao dịch mua thêm trang in của sinh viên.
- **Thông tin in:** Bao gồm vị trí, trạng thái hoạt động, cấu hình và lịch sử bảo trì của từng máy in trong khuôn viên trường.
- **Cấu hình hệ thống:** Lưu trữ các thiết lập về số trang in mặc định mỗi học kỳ, ngày cấp số trang miễn phí, loại file được chấp nhận khi tải lên, và các tùy chọn cấu hình khác do SPSO quản lý.



- **Báo cáo và log hoạt động:** Ghi lại chi tiết các hoạt động in ấn, báo cáo hàng tháng và hàng năm, giúp SPSO theo dõi và phân tích tình hình sử dụng dịch vụ.

Quản lý và bảo mật dữ liệu:

- **Xác thực và phân quyền:** Tất cả người dùng phải xác thực qua HCMUT_SSO trước khi truy cập hệ thống, đảm bảo chỉ có những người dùng hợp lệ mới có thể sử dụng dịch vụ.
- **Sao lưu và phục hồi dữ liệu:** Thiết lập cơ chế sao lưu định kỳ và kế hoạch phục hồi dữ liệu để đảm bảo an toàn thông tin và duy trì hoạt động liên tục của hệ thống.
- **Bảo mật dữ liệu:** Áp dụng các biện pháp mã hóa và bảo mật cần thiết để bảo vệ dữ liệu người dùng và thông tin quan trọng của hệ thống khỏi các truy cập trái phép.

Kết luận:

Việc lựa chọn MongoDB làm cơ sở dữ liệu chính giúp chúng tôi xây dựng một hệ thống linh hoạt, hiệu quả và dễ dàng mở rộng trong tương lai. Điều này đảm bảo dịch vụ in thông minh HCMUT_SSPS có thể đáp ứng tốt nhu cầu ngày càng tăng của sinh viên và nhân viên quản lý, đồng thời tạo nền tảng vững chắc cho việc phát triển các tính năng mới trong tương lai.

3.2.4 API management

Giới thiệu

Để triển khai phần quản lý API trong hệ thống HCMUT-SSPS, có thể thực hiện theo hướng dẫn sau.

Mô tả cách quản lý API cho hệ thống HCMUT-SSPS

Hệ thống HCMUT-SSPS sẽ tích hợp và quản lý các dịch vụ hoặc API bên ngoài như hệ thống thanh toán BKPay và dịch vụ xác thực HCMUT_SSO để đảm bảo tính ổn định và bảo mật khi xử lý thông tin của sinh viên. Khi tích hợp các API này, hệ thống sẽ cần một API Gateway để làm điểm trung gian cho mọi giao tiếp giữa hệ thống và các dịch vụ bên ngoài. API Gateway sẽ chịu trách nhiệm điều phối luồng dữ liệu, xác thực yêu cầu từ phía client, và xử lý lỗi để ngăn chặn các vấn đề bảo mật khi kết nối với các API của bên thứ ba.

Kết nối với hệ thống BKPay

- API Gateway sẽ gửi yêu cầu thanh toán đến hệ thống BKPay thông qua các endpoint bảo mật của BKPay. Các giao dịch thanh toán sẽ được mã hóa, và kết quả thanh toán sẽ được trả về hệ thống sau khi xử lý thành công.
- Trong trường hợp thanh toán thất bại (do lỗi kết nối hoặc thông tin không hợp lệ), API Gateway sẽ quản lý thông báo lỗi và gửi lại phản hồi cho sinh viên, yêu cầu họ kiểm tra lại thông tin thanh toán.



Xác thực người dùng với HCMUT_SSO

- HCMUT-SSPS sử dụng HCMUT_SSO để xác thực danh tính của sinh viên khi đăng nhập. API Gateway sẽ gửi yêu cầu xác thực với thông tin đăng nhập từ phía người dùng đến HCMUT_SSO và đợi phản hồi xác thực.
- Nếu xác thực thành công, hệ thống sẽ cấp quyền truy cập vào hệ thống in ấn cho sinh viên, và lưu trạng thái đã đăng nhập để tránh yêu cầu xác thực lại trong cùng một phiên làm việc.

Quản lý các yêu cầu và lỗi

- API Gateway sẽ xử lý mọi lỗi phát sinh trong quá trình kết nối với API (như lỗi server hoặc yêu cầu không hợp lệ) và gửi thông báo đến sinh viên khi cần thiết.
- API Gateway cũng sẽ thực hiện logging để lưu trữ các nhật ký yêu cầu đến API, giúp đội ngũ vận hành dễ dàng theo dõi và xử lý các vấn đề phát sinh.

Lợi ích khi sử dụng API Gateway

- **Bảo mật:** Đảm bảo dữ liệu sinh viên và giao dịch thanh toán được mã hóa và kiểm soát chặt chẽ.
- **Dễ dàng bảo trì:** API Gateway tập trung tất cả các kết nối với các dịch vụ bên ngoài, giúp đội ngũ vận hành dễ dàng quản lý các thay đổi về API hoặc sửa lỗi.
- **Khả năng mở rộng:** API Gateway giúp dễ dàng bổ sung các dịch vụ API mới nếu cần tích hợp thêm các dịch vụ từ bên thứ ba khác.

Hệ thống sẽ sử dụng cơ chế xác thực OAuth hoặc JWT để đảm bảo rằng chỉ các người dùng có thẩm quyền mới có thể truy cập được các dịch vụ này. Ví dụ, khi một sinh viên thực hiện giao dịch thanh toán qua BKPay, API Gateway sẽ gửi yêu cầu đến BKPay kèm theo mã xác thực duy nhất của người dùng hiện tại. Mọi phản hồi từ BKPay sẽ được API Gateway xử lý trước khi gửi lại cho người dùng, nhằm đảm bảo rằng dữ liệu nhạy cảm luôn được bảo vệ. Các dịch vụ API như HCMUT_SSO cũng sẽ được tích hợp theo cách tương tự để xác thực danh tính của người dùng trước khi cho phép truy cập vào hệ thống.

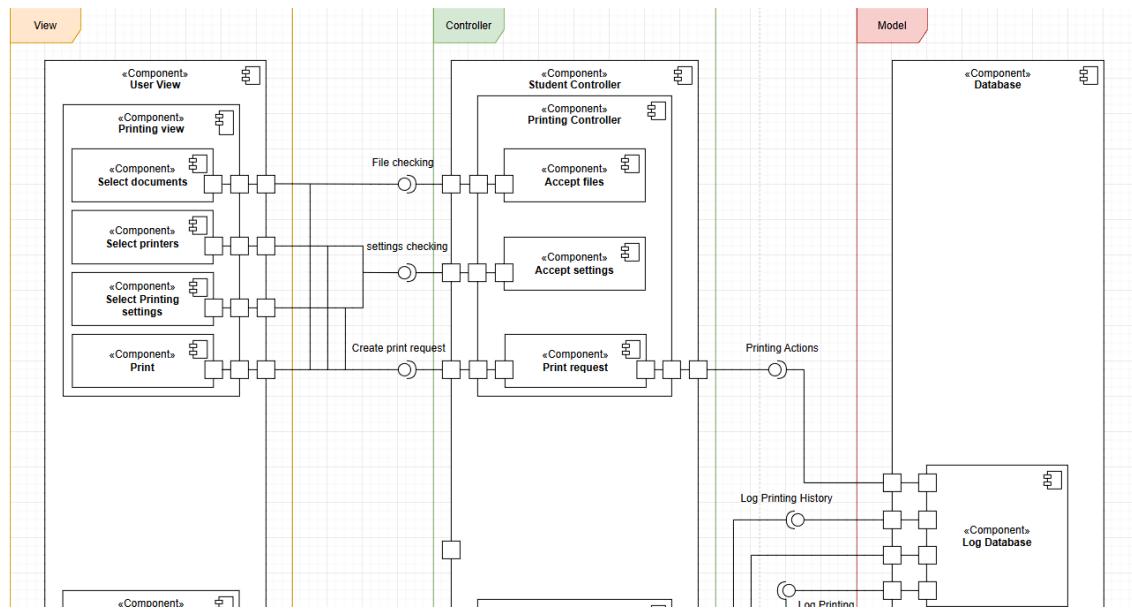
Ngoài ra, hệ thống có thể sử dụng một middleware để xác thực sinh viên trước khi thực hiện giao dịch với các API trước khi yêu cầu được chuyển đến controller.

API BKPay

API được tích hợp BKPay có thể truy cập tại: <https://bkpay.hcmut.edu.vn/bkpay/home.action>

3.3 Component Diagram

3.3.1 Component Diagram: Printing



Mô tả Component Diagram

View

- **User View:** Giao diện người dùng chính, nơi sinh viên tương tác trực tiếp với hệ thống. Giao diện này cung cấp các chức năng cho phép sinh viên chọn tài liệu cần in, chọn máy in, và tùy chỉnh các cài đặt in.

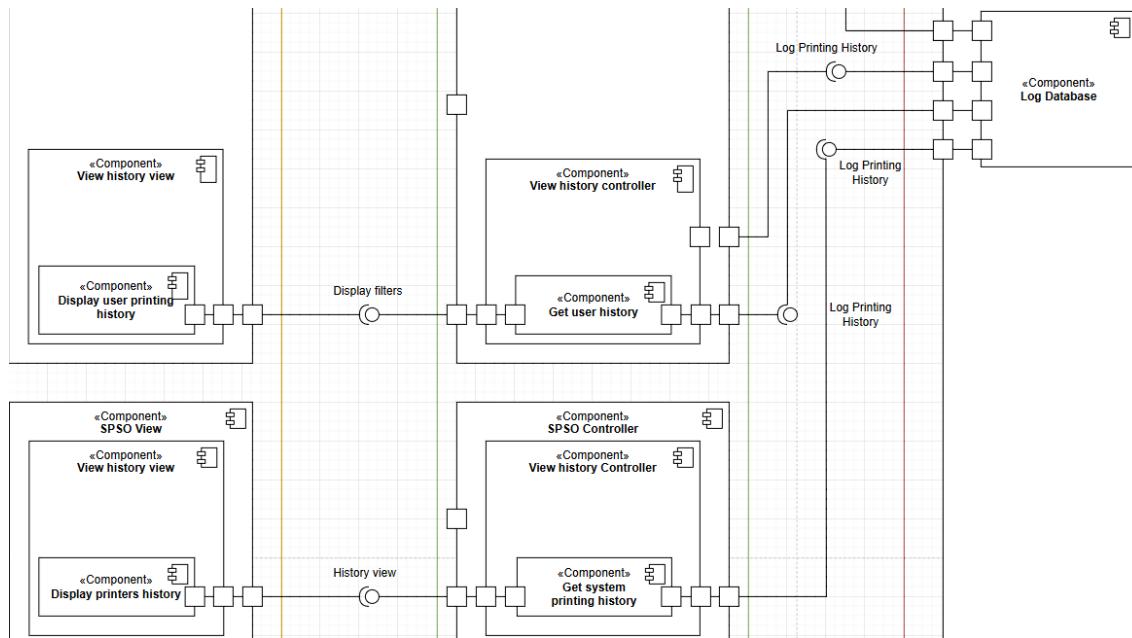
Controller

- **Student Controller:** Xử lý các yêu cầu từ giao diện người dùng, kiểm tra tính hợp lệ của dữ liệu nhập vào và điều phối các hoạt động của hệ thống.
- **Printing Controller:** Chuyên biệt xử lý các yêu cầu liên quan đến in ấn, như tạo yêu cầu in, theo dõi trạng thái in và lưu trữ lịch sử in.

Model

- **Database:** Cơ sở dữ liệu lưu trữ thông tin về người dùng, máy in, tài liệu, lịch sử in và các cài đặt hệ thống.

3.3.2 Component Diagram: View History



View

- **User View:** Giao diện người dùng chính, nơi sinh viên xem lịch sử in, SPSO xem lịch sử của máy in cụ thể.

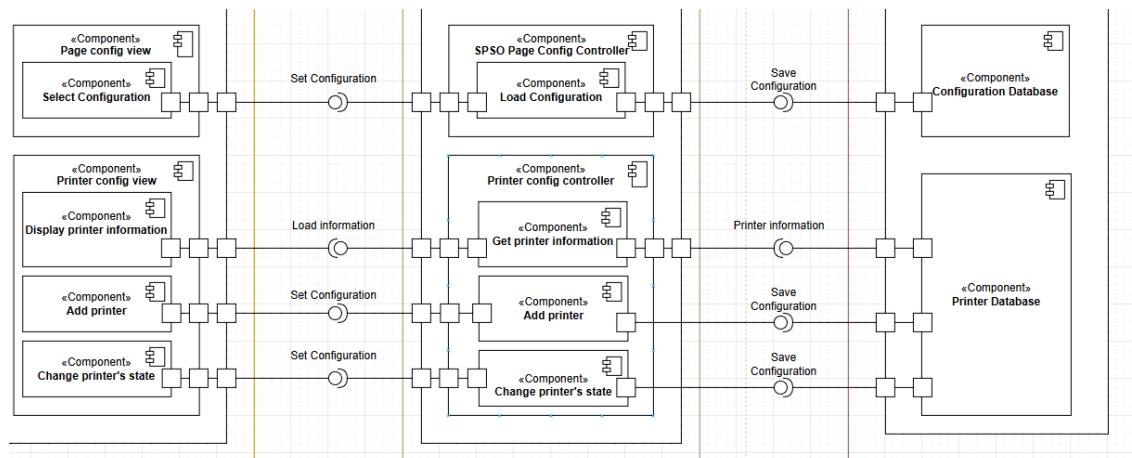
Controller

- **Printing Controller:** Xử lý lịch sử in của người dùng vào máy in đó, hiển thị ra cho SPSO xem

Model

- **Database:** Cơ sở dữ liệu lưu thông tin lịch sử in của người dùng.

3.3.3 Component Diagram: Page Config





View

- **User View:** Giao diện người dùng, nơi SPSO có thể cài đặt số trang in mặc định cho sinh viên

Controller

- **Printing Controller:** Xử lý các cài đặt in mặc định cho người dùng, thay đổi trạng thái cụ thể của máy in.

Model

- **Database:** Cơ sở dữ liệu lưu thông tin của máy in.

4 Implementation – Sprint 1

4.1 Cài đặt Github

Đây là github của nhóm: [GITHUB](#)

4.2 Thêm tài liệu vào Github

Nhóm bọn em cũng đã hoàn thiện việc thêm các tài liệu lên Github, bao gồm các diagram:

The screenshot shows a GitHub repository page. At the top, there is a commit from 'PhatVND' titled 'Update diagram' with a timestamp of '36b633e · now' and '24 Commits'. Below this, a list of commits is shown:

Commit	Message	Time
Backend	backend added	4 hours ago
Diagram	Update diagram	now
Frontend	Thêm sắp xếp cho lịch sử in - có truy vấn url , chuyển thành ...	4 days ago
README.md	Initial commit	last month
readmeasdasd.txt	Create readmeasdasd.txt	4 hours ago

Below the commit list, there is a 'README' file with the following content:

```
SE241_NiemVui
Website for Student Smart Printing Service
```

4.3 Thực hiện usability testing

4.3.1 Usability Testing là gì?

Usability Testing Là Gì?

Usability Testing (Kiểm tra khả năng sử dụng) là một kỹ thuật nghiên cứu được sử dụng để đánh giá mức độ dễ sử dụng của sản phẩm hoặc dịch vụ. Quá trình này thường liên quan đến việc quan sát người dùng thực tế khi họ thực hiện các nhiệm vụ cụ thể với sản phẩm trong một môi trường kiểm soát. Mục đích chính là xác định các vấn đề mà người dùng gặp phải, đánh giá mức độ hài lòng của họ và đưa ra các cải tiến.

- Nó không chỉ áp dụng cho thiết kế giao diện người dùng (UI/UX), mà còn được sử dụng cho các sản phẩm vật lý, hệ thống phần mềm, hoặc quy trình bất kỳ nơi người dùng tương tác với sản phẩm hoặc dịch vụ.

Quy Trình Usability Testing

1. **Xác định mục tiêu thử nghiệm:** Tìm hiểu xem bạn muốn đo lường hoặc cải tiến điều gì (ví dụ: tìm lỗi, cải thiện giao diện, đánh giá trải nghiệm).



2. **Tuyển chọn người dùng:** Người tham gia nên đại diện cho đối tượng mục tiêu của sản phẩm.
3. **Thiết kế kịch bản thử nghiệm:** Tạo ra các nhiệm vụ mà người dùng cần hoàn thành trong sản phẩm (ví dụ: "Tìm kiếm sản phẩm và đặt hàng").
4. **Tiến hành thử nghiệm:** Quan sát người dùng thực hiện các nhiệm vụ. Có thể sử dụng các công cụ ghi hình, ghi âm hoặc theo dõi hành động.
5. **Phân tích kết quả:** Ghi nhận các vấn đề, khó khăn hoặc hành vi thú vị từ người dùng.
6. **Dưa ra cải tiến:** Dựa trên kết quả, điều chỉnh thiết kế hoặc hệ thống.

Lợi Ích Của Usability Testing

- **Cải thiện trải nghiệm người dùng (UX):** Phát hiện sớm các vấn đề khiến người dùng bối rối hoặc khó chịu.
- **Tăng tính thân thiện của sản phẩm:** Giúp sản phẩm dễ hiểu và dễ sử dụng hơn, từ đó tăng tỷ lệ sử dụng và sự hài lòng của khách hàng.
- **Giảm chi phí sửa đổi:** Phát hiện lỗi sớm giúp giảm chi phí sửa chữa trong tương lai, đặc biệt là sau khi sản phẩm đã ra mắt.
- **Tối ưu hóa mục tiêu kinh doanh:** Giao diện thân thiện và hiệu quả có thể giúp tăng doanh thu, giữ chân khách hàng lâu hơn.

Ví Dụ Minh Họa Usability Testing

- **Website Thương Mại Điện Tử:** Người dùng được yêu cầu tìm kiếm và mua một sản phẩm cụ thể. Quan sát xem họ có gặp khó khăn trong quá trình tìm kiếm hoặc thanh toán không.
- **Ứng Dụng Di Động:** Người dùng thử đăng ký tài khoản hoặc sử dụng một tính năng cụ thể trên app. Quan sát xem giao diện có dễ hiểu hay không.
- **Thiết Bị Thông Minh:** Một nhóm người dùng thử sử dụng máy pha cà phê tự động để xem hướng dẫn có rõ ràng và dễ làm theo hay không.

Ý Tưởng Minh Họa Cho Usability Testing

- **Cảnh thử nghiệm người dùng:** Một nhóm nhỏ người dùng ngồi trước máy tính, điện thoại hoặc thiết bị đang được thử nghiệm.
- **Sơ đồ quy trình:** Minh họa quy trình Usability Testing:

Bước 1: Xác định mục tiêu → **Bước 2:** Tuyển chọn người dùng → **Bước 3:** Thiết kế kịch bản → **Bước 4:** Tiến hành → **Bước 5:** Phân tích → **Bước 6:** Cải tiến.

- **Tương tác người dùng:** Người dùng bấm vào nút trong giao diện hoặc cỗ găng hoàn thành một nhiệm vụ cụ thể (ví dụ: nhập thông tin thanh toán).
- **Phòng thử nghiệm thực tế:** Một phòng thử nghiệm với thiết bị được thiết lập: người dùng cầm điện thoại hoặc ngồi trước màn hình máy tính, camera quay lại hành động của họ.

4.3.2 Chọn người kiểm tra và người tham gia

4.3.3 Xác định nhiệm vụ

1. Xác định các tác vụ chính mà người dùng cần thực hiện (User Tasks):

a. Đối với sinh viên:

- Đăng nhập hệ thống thông qua HCMUT_SSO.
- Tải lên một file tài liệu để in.
- Chọn máy in phù hợp.
- Thiết lập các thuộc tính in:
 - Loại giấy (A4, A3...).
 - Số trang (chọn trang in cụ thể từ tài liệu).
 - Chế độ in (một mặt/hai mặt).
 - Số bản sao cần in.
- Thanh toán số trang in qua hệ thống BKPay nếu hết trang in.
- Xem lại lịch sử in và kiểm tra số trang còn lại trong tài khoản.

b. Đối với nhân viên SPSO

- Đăng nhập vào hệ thống.
- Quản lý danh sách máy in: thêm, vô hiệu hóa hoặc kích hoạt lại máy in.
- Cập nhật cấu hình hệ thống (định dạng file được phép, số trang mặc định mỗi kỳ).
- Xem và tạo báo cáo sử dụng hệ thống in (theo tháng/năm).

2. Định nghĩa chi tiết các kịch bản kiểm tra usability:

In tài liệu (sinh viên):

- Đăng nhập vào hệ thống qua giao diện.
- Tải lên tài liệu cần in.
- Chọn máy in thích hợp nhất.
- Thiết lập các thuộc tính in (A4, 2 mặt, trang 1-10, 2 bản sao).
- Kiểm tra thông báo về số trang còn lại.
- Nhấn nút in và nhận xác nhận in thành công.

Quản lý máy in (SPSO):

- Đăng nhập vào hệ thống quản trị.
- Kiểm tra danh sách máy in (ID, trạng thái, vị trí).
- Vô hiệu hóa một máy in cụ thể.
- Kiểm tra trạng thái cập nhật (hiển thị là "Disabled").
- Kích hoạt lại máy in và xác nhận.

Mua thêm số trang in (sinh viên):

- Đăng nhập vào hệ thống..
- Chọn tính năng Mua thêm số trang in.
- Nhập số trang muốn mua và thực hiện thanh toán qua BKPay.
- Kiểm tra thông báo về số trang được cập nhật thành công.

4.3.4 Phương thức kiểm tra

Phương thức mà nhóm quyết định sẽ nhờ người dùng kiểm tra là thông qua 2 phương thức sau:

Phương thức 1: Có giám sát trực tiếp

Phương thức này được thực hiện dưới sự giám sát trực tiếp từ các nhà phát triển. Quá trình kiểm tra diễn ra thông qua:

- Các buổi họp trực tuyến qua Google Meet.
- Thảo luận, phản hồi trực tiếp giữa người kiểm tra và nhà phát triển.

Mục tiêu của phương pháp này là đảm bảo việc phân tích lỗi và điều chỉnh kịp thời, giúp cải thiện trải nghiệm và chất lượng của tính năng.

Phương thức 2: Kiểm tra gián tiếp

Phương thức này được tiến hành thông qua các phiếu khảo sát trực tuyến trên Google Forms, được gửi đến các bên liên quan. Người dùng được yêu cầu:

- Cung cấp ý kiến và đánh giá sau khi sử dụng tính năng.
- Báo cáo các lỗi hoặc vấn đề mà họ gặp phải.

Phương pháp này sẽ thu thập thông tin phong phú từ nhiều nguồn, đồng thời giảm thiểu sự phụ thuộc vào giám sát trực tiếp.

Cả hai phương thức đều sẽ hỗ trợ lẫn nhau, đảm bảo rằng mọi khía cạnh của tính năng được kiểm tra kỹ lưỡng và toàn diện.

Hình ảnh ghi nhận được

Thông kê câu trả lời	Tính năng upload file hoạt động tốt không:	Tính năng chọn vị trí máy in có hoạt động tốt không	Các cài đặt in có hoạt động tốt không	Trang hiển thị file được upload lên có vấn đề gì không?	Sau khi
Người thực hiện 1	Có	Có	Có	Khi upload file lên, chỗ hiển thị file gần như quá to, không thấy được rõ các nội dung File, ngoài ra thì các kích thước hiển thị giao diện giữa các file có vấn đề, không đồng đều	Hoạt động
Người thực hiện 2	Có	Có	Có	Vấn đề màu phông không hợp mắt	Hoạt động

Ta có thể thấy được rằng đã có tổng số thực hiện đến thời điểm hiện tại là 2, cả 2 đều cho rằng những tính năng đều hoạt động tốt. Tuy nhiên, phần giao diện hiển thị vẫn chưa được đúng ý của người dùng lắm. Đội ngũ phát triển sẽ ghi nhận và sẽ khắc phục đó vào MVC2.

4.3.5 Tiến hành kiểm tra

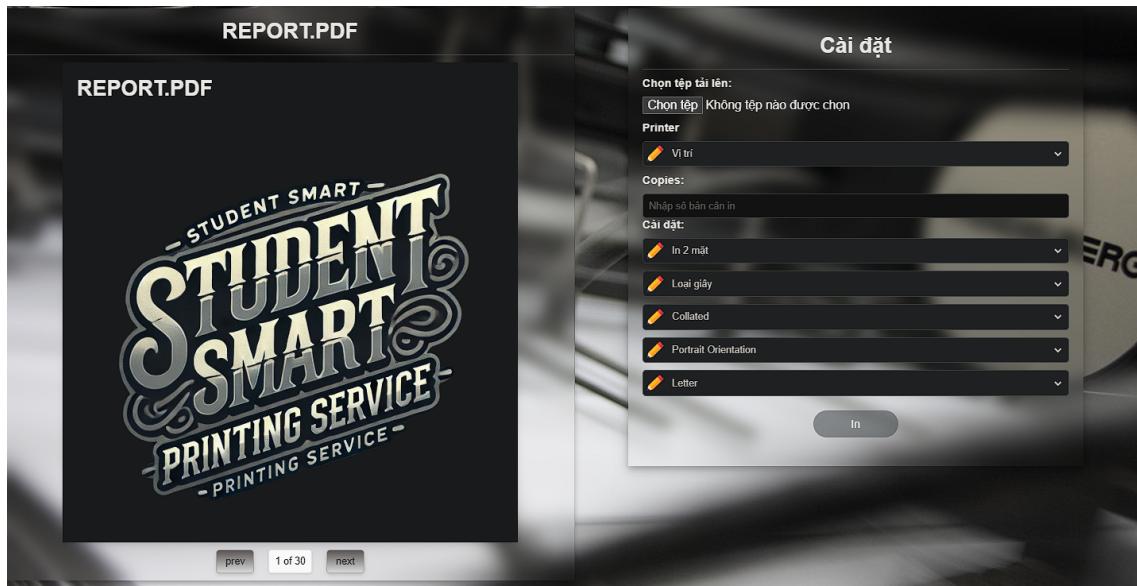
Người dùng mua thêm số trang in





- Nhận xét: Tính năng hoạt động ổn, có thông báo hiển thị khi thiết lập thành công.
- Nhược điểm:
 - Nhấn Enter sau khi nhập số (không nhấn lưu) sẽ reload lại trang mà không lưu thiết lập.
 - Cần hiển thị số trang in hiện có đang là bao nhiêu

Người dùng tùy chọn in



- Nhận xét:
 - Chọn file thành công
 - Có thể chọn máy in theo vị trí hoặc theo trạng thái
 - Có thể chọn số bản cần in
 - Về phần thông số: có thể chọn in 1 hoặc 2 mặt, loại giấy, hướng trang in,...
- Nhược điểm: Khi bấm Submit không có thông báo thiết lập thành công.

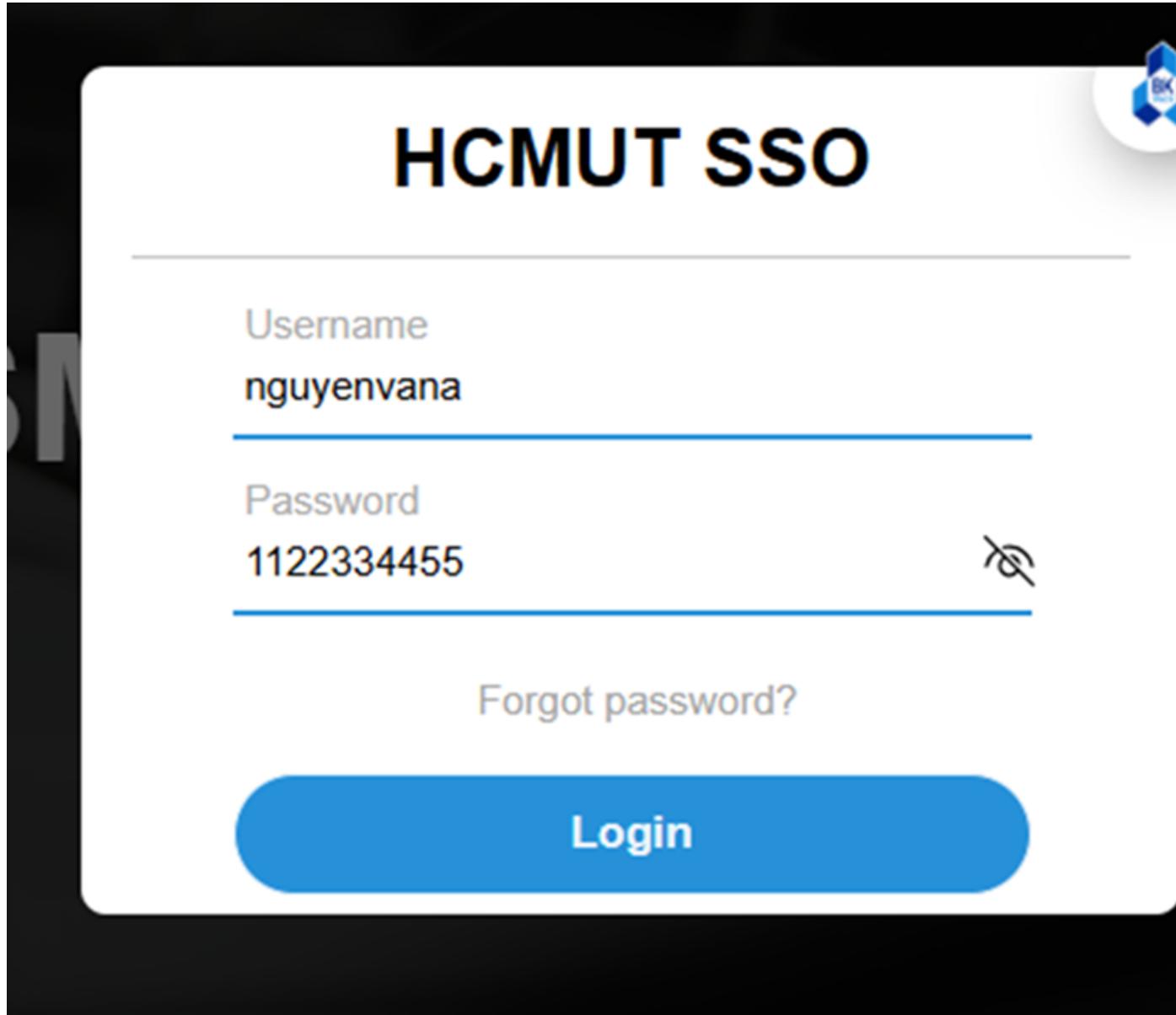


Admin xem lịch sử in của hệ thống

Lịch sử in ấn			
Mã in	Ngày in	Nơi thực hiện	Option
1111111	25/02/2025	H1-104	<button>View</button>
1111111	25/02/2025	H1-104	<button>View</button>
1111111	25/02/2025	H1-104	<button>View</button>
1111111	25/02/2025	H1-104	<button>View</button>
1111111	25/02/2025	H1-104	<button>View</button>
1111111	25/02/2025	H1-104	<button>View</button>

- Nhận xét: Hiển thị đầy đủ thông tin của các lần in, gồm: mã in, ngày in, nơi thực hiện.
- Nhược điểm: Không có.

User login vào hệ thống



- Nhận xét: Có thẻ nhập thông tin username và password.

Admin xem thông tin của người dùng

Thông tin người dùng			View All
Profile	Email	Cài đặt	
	kujodino@hcmut.edu.vn	 	
	mafia@hcmut.edu.vn	 	
	phat.vonguyen@hcmut.edu.vn	 	
	yessir@hcmut.edu.vn	 	

- Nhận xét: Có thẻ xem thông tin của người dùng và có thẻ chỉnh sửa ở mục cài đặt.
- Nhược điểm: Nút chỉnh sửa trong phần cài đặt không hoạt động.

Admin thao tác hoạt động của các máy in

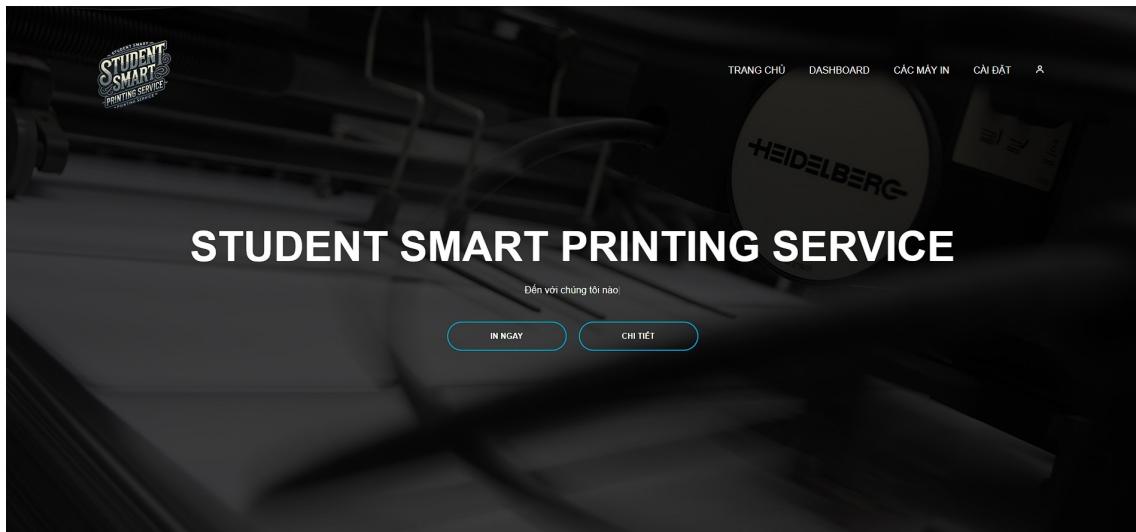


- Nhận xét: Admin có thể chỉnh sửa hoạt động của các máy in.
- Nhược điểm: Không có.

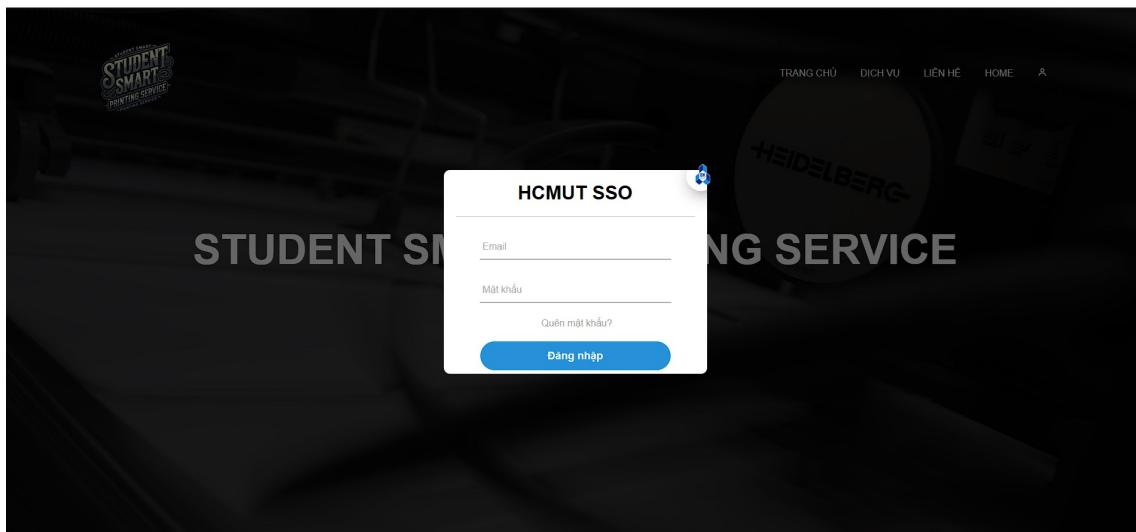
4.3.6 Thu thập phản hồi và báo cáo kết quả

Nhìn chung, giao diện hiện tại vẫn còn thay đổi sao cho phù hợp với các ý kiến của người dùng. Giao diện mới sẽ được cập nhật ở lần cập nhật cuối.

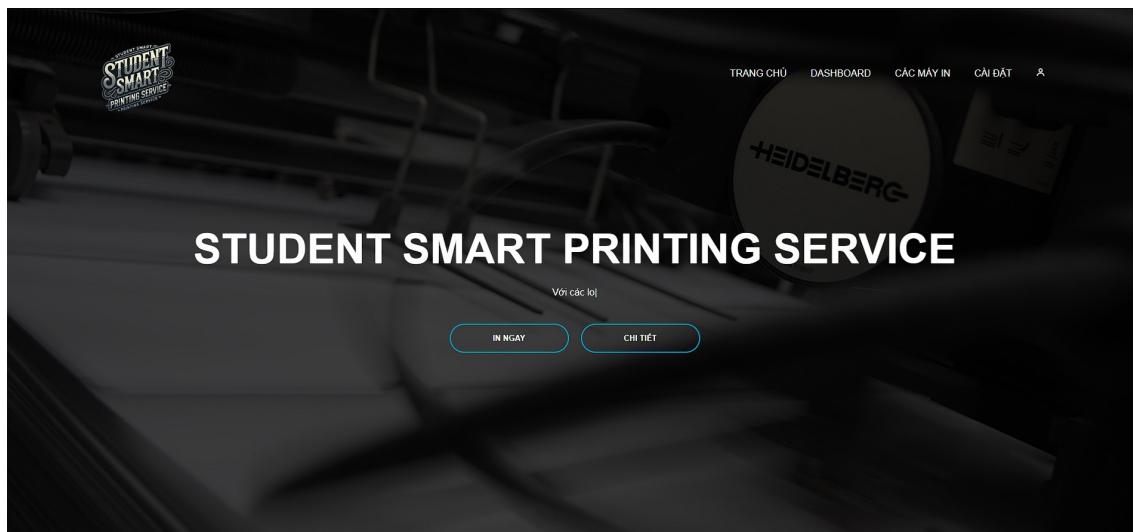
5 Giao diện trang web



Đây là giao diện chính khi vào trang web của nhóm chúng em. Chúng ta để có thể In được thì ta cần phải Đăng nhập vào



Khi đăng nhập thành công, sẽ được lưu thông tin lên cookie. Và hệ thống sẽ xét xem cái cookie đó để phân quyền hợp lý: Admin hay User bình thường.



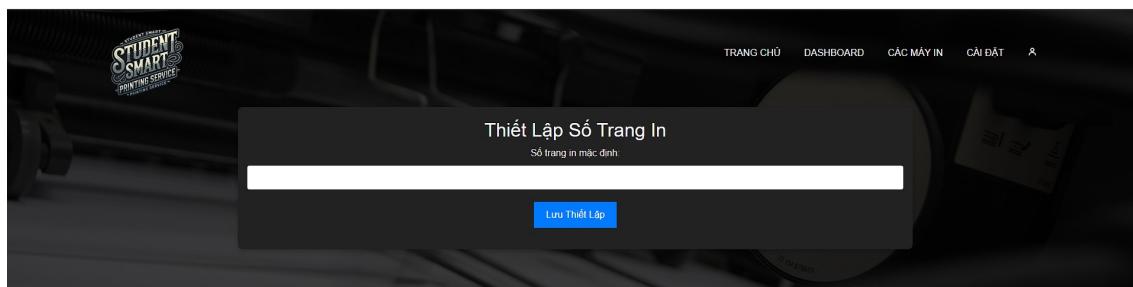
Đây là trang Admin, có thể kiểm soát được các máy in, người dùng và Cài đặt được trang mặc in mặc định.

Mã in	Ngày in	Nơi thực hiện	Option
1111111	25/02/2025	H1-104	View
1111111	25/02/2025	H1-104	View
1111111	25/02/2025	H1-104	View
1111111	25/02/2025	H1-104	View
1111111	25/02/2025	H1-104	View
1111111	25/02/2025	H1-104	View

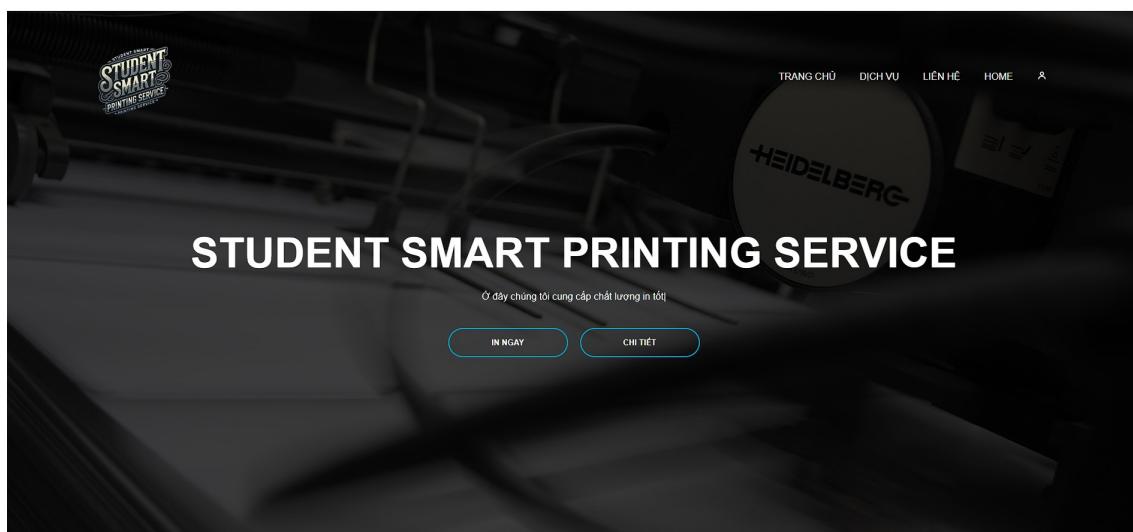
Profile	Email	Cài đặt
	kujodino@hcmut.edu.vn	
	meilia@hcmut.edu.vn	
	phat.vonguyen@hcmut.edu.vn	
	yessir@hcmut.edu.vn	

Mã in	Ngày in	Nơi thực hiện	Option
1111111	25/02/2025	H1-104	View
1111111	25/02/2025	H1-104	View
1111111	25/02/2025	H1-104	View
1111111	25/02/2025	H1-104	View
1111111	25/02/2025	H1-104	View
1111111	25/02/2025	H1-104	View

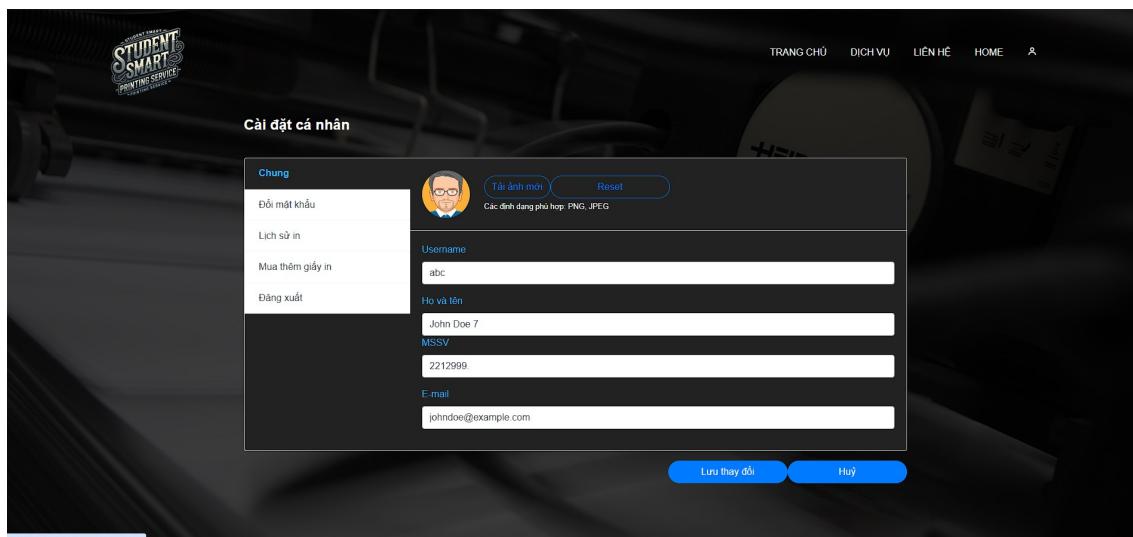
Printer	Vị trí	Lựa chọn
	H1-104	Enable

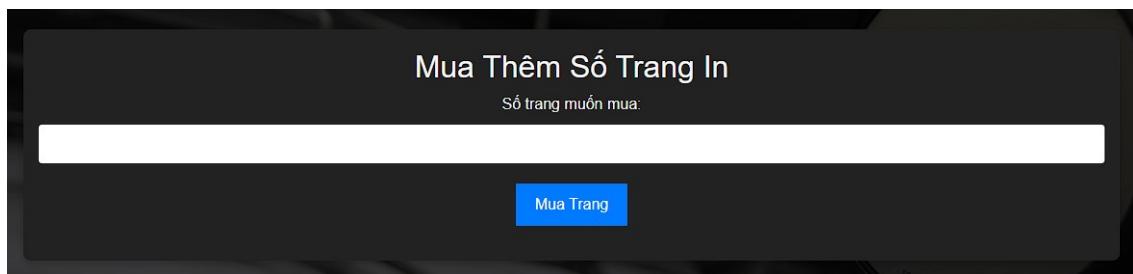
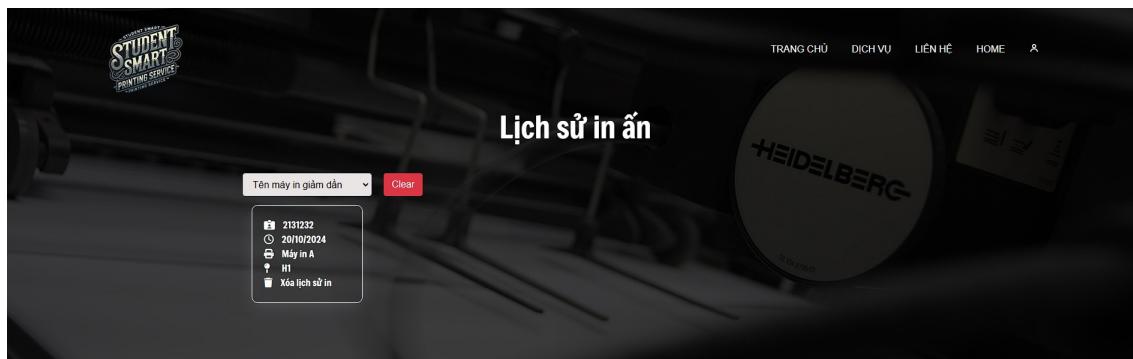


Còn đối với người dùng, khi đăng nhập vào thì sẽ có thể tiến hành vào In, cũng như thay đổi các thông tin cá nhân.

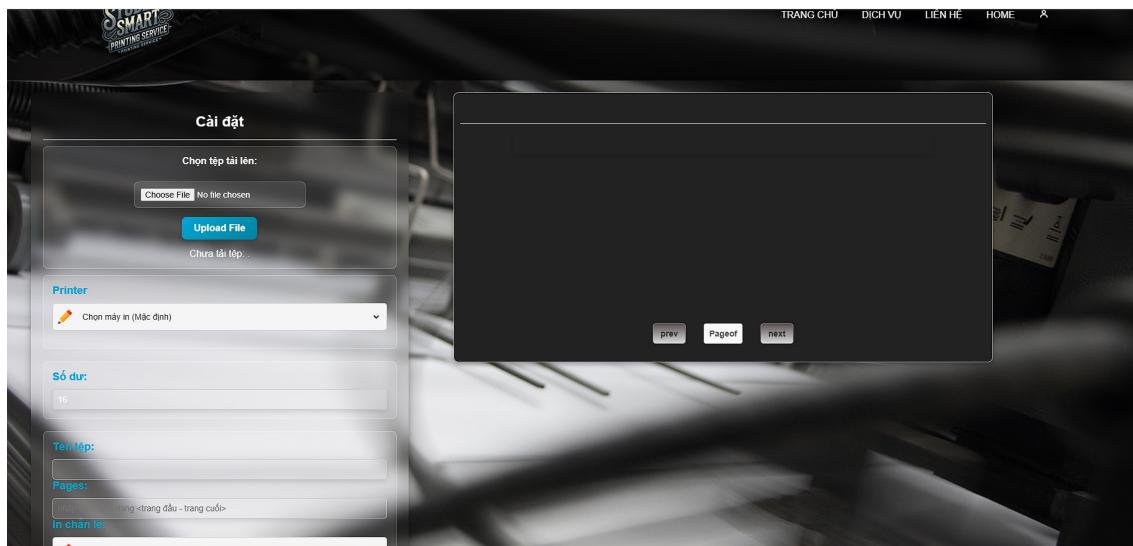


Người dùng có thể bấm vô phần Chính sửa thông tin để đổi được mật khẩu, xem lịch sử in cũng như Mua thêm giấy in.





Và cuối cùng là trang để người dùng in



Ban đầu, người dùng phải lựa chọn tệp tải lên, chọn toàn bộ cài đặt thì nó sẽ hiển thị ra như bên phải:



Sau khi đã cài đặt xong, bấm lệnh In thì hệ thống sẽ gửi yêu cầu in lên database.



Tài liệu