

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



Công nghệ phần mềm - CO3001

Báo cáo

STUDENT SMART PRINTING SERVICE HCMUT

Giảng viên hướng dẫn: Phan Trung Hiếu

Sinh viên thực hiện: 2114417 - Nguyễn Ngọc Phú
2112122 - Nguyễn Hồng Quân
2110527 - Nguyễn Hoàng Duy Tân
2112594 - Trần Nguyễn Minh Tuệ
2110342 - Nguyễn Minh Lộc
2111762 - Phạm Võ Quang Minh

TP. Hồ Chí Minh, 11/2023

Mục lục

1 Requirement elicitation	4
1.1 Giới thiệu về bối cảnh miền của hệ thống phát triển dịch vụ in ấn thông minh. Những bên liên quan, nhu cầu hiện tại và lợi ích SPSS mang lại cho từng bên liên quan	4
1.1.1 Giới thiệu về bối cảnh miền của dịch vụ in ấn thông minh cho sinh viên đại học Bách Khoa thành phố Hồ Chí Minh	4
1.1.2 Những bên liên quan và nhu cầu hiện tại	4
1.1.3 Lợi ích SPSS mang lại cho những bên liên quan	5
1.2 Mô tả tất cả yêu cầu chức năng và phi chức năng có thể suy ra từ tổng thể bài toán	5
1.2.1 Yêu cầu chức năng (functional requirements) (Ít nhất 5 yêu cầu chức năng cho mỗi bên liên quan)	5
1.2.2 Yêu cầu phi chức năng (non-functional requirements)	6
1.3 Use Case Diagram	7
1.3.1 Whole System	7
1.3.2 Use case Order Printing Job	8
1.3.3 Use case Login Service	9
2 System modelling	10
2.1 Activity diagram	10
2.2 Sequence diagram	15
2.3 Class diagram	18
2.4 User interfaces	20
2.4.1 Trang đăng nhập (Login Page)	20
2.4.2 Trang chủ (Home Page)	22
2.4.3 Trang tải file lên hệ thống	23
2.4.4 Trang danh sách đang trong hàng đợi	25
2.4.5 Trang xác nhận danh sách đơn hàng	26
2.4.6 Trang thanh toán	28
2.4.7 Trang danh sách đơn hàng có trên hệ thống	29
3 Architecture design	31
3.1 Kiến trúc phân lớp (Layered architecture)	31
3.1.1 Kế hoạch trình bày giao diện (Interface strategy)	31
3.1.2 Data storage approach (Phương pháp lưu trữ dữ liệu)	31
3.1.3 Quản lý API (API management)	31
3.1.4 Các biểu đồ phân lớp	32
3.2 Component diagram	33

Danh mục hình ảnh

Hình 1: Use Case Diagram for the Whole System	7
Hình 2: Use Case Diagram for Order Printing Job	8
Hình 3: Use Case Diagram for Login Service	9
Hình 4: Activity diagram for Order Printing	10
Hình 5: Activity diagram for Upload File	11
Hình 6: Activity diagram for Page Setting	12
Hình 7: Activity diagram for Confirm Order	13
Hình 8: Sequence diagram of Upload file	15
Hình 9: Sequence diagram of Preview and Remove file	16
Hình 10: Sequence diagram of Checkout	17

Hình 11: Frontend Class diagram	18
Hình 12: Backend Class diagram	19
Hình 13: FE_BE Connect diagram	19
Hình 14: Giao diện trang đăng nhập	20
Hình 15: Lỗi khi người dùng nhập thông tin không thỏa mãn các trường	21
Hình 16: Lỗi khi người dùng nhập sai thông tin được lưu trữ	21
Hình 17: UI Home of Desktop View	22
Hình 18: Giao diện Box file	23
Hình 19: Giao diện trang tải file	23
Hình 20: Giao diện trang tải file khi người dùng xóa file mới được upload	24
Hình 21: Các thông số người dùng điều chỉnh	24
Hình 22: Giao diện trang danh sách đang trong hàng đợi	25
Hình 23: Chế độ preview file	26
Hình 24: Giao diện trang xác nhận danh sách đơn hàng	26
Hình 25: Chế độ preview file	27
Hình 26: Chế độ preview file	27
Hình 27: Giao diện trang thanh toán	28
Hình 28: Giao diện trang đơn hàng có trên hệ thống	29
Hình 29: Lọc theo đơn hàng theo ngày	30
Hình 30: Box-line diagram	32
Hình 31: Deployment Diagram	33
Hình 32: Component diagram	34

Danh mục bảng biểu

Bảng 1: Order Printing Job scenario	9
---	---

1 Requirement elicitation

1.1 Giới thiệu về bối cảnh miền của hệ thống phát triển dịch vụ in ấn thông minh. Những bên liên quan, nhu cầu hiện tại và lợi ích SPSS mang lại cho từng bên liên quan

1.1.1 Giới thiệu về bối cảnh miền của dịch vụ in ấn thông minh cho sinh viên đại học Bách Khoa thành phố Hồ Chí Minh

Bối cảnh miền của một hệ thống bao gồm miền vấn đề (problem domain) và miền ứng dụng (application domain). Dịch vụ in ấn thông minh (SPSS) là một hệ thống được phát triển bởi trường Đại học Bách Khoa Thành phố Hồ Chí Minh nhằm cung cấp cho sinh viên một dịch vụ in ấn tiện lợi, an toàn và nhanh chóng nhất với số lượng nhỏ. Dịch vụ có thể sẽ được cung cấp dưới hai dạng: tự phục vụ và truyền thống. Mọi sinh viên đều được khuyến khích sử dụng dịch vụ này nếu bản thân sinh viên gặp phải những vấn đề khó khăn khi sử dụng dịch vụ in truyền thống tại trường như sau (problem domain):

- Không chủ động thời gian vì phụ thuộc vào các tiệm in.
- Không tận dụng được thời gian di chuyển đến tiệm in.
- Vấn đề bảo mật thông tin khi sử dụng dịch vụ truyền thống như virus.
- Không bảo đảm về phương thức thanh toán phù hợp.
- Giá thành cao hơn so với dịch vụ in ấn truyền thống ngoài trường.

Miền ứng dụng (application domain) dùng để chỉ phạm vi chức năng hoặc lĩnh vực cụ thể mà hệ thống được thiết kế để yêu cầu phục vụ. Đối với SPSS, chúng ta có phạm vi chức năng và lĩnh vực như sau:

- Phạm vi lĩnh vực
 - Giáo dục: Hỗ trợ in ấn tài liệu cho sinh viên, giảng viên.
 - Kinh doanh: Những người phát triển hệ thống kiếm lợi từ việc cung cấp dịch vụ.
- Phạm vi chức năng
 - In ấn tài liệu của sinh viên giảng viên đa dạng từ tài liệu học tập, giáo trình, đề thi, nghiêm cứu khoa học, bảng điểm, chứng chỉ,...
 - Đóng gói và bao bì an toàn, bảo mật tài liệu.
 - Giao hàng và chuyển phát với tốc độ nhanh (không bắt buộc).

1.1.2 Những bên liên quan và nhu cầu hiện tại

Stakeholder - một thuật ngữ dùng để chỉ bất kỳ người hay tổ chức mà bị ảnh hưởng bởi hệ thống và có lợi ích chính đáng. Hệ thống dịch vụ in ấn thông minh của trường (SSPS) gồm các bên liên quan sau:

- Sinh viên, giảng viên trường đại học Bách Khoa: sử dụng SSPS để phục vụ việc in ấn tài liệu theo nhu cầu một cách thuận tiện và nhanh chóng.
- SPSO - Quản lý dịch vụ in ấn sinh viên: có nhu cầu nhằm nắm rõ được hoạt động in ấn của sinh viên như thói quen, sở thích, tần suất và quản lý cơ sở vật chất, trang thiết bị.
- Trường đại học bách khoa TP HCM (HCMUT): tích hợp SSPS vào hệ thống của trường để theo dõi và nắm bắt hoạt động in ấn của sinh viên và kiếm thêm lợi nhuận.
- Người giám sát máy in và thực hiện in ấn (trong trường hợp triển khai theo kiểu truyền thống): nhu cầu kiếm thêm thu nhập (nhân viên hỗ trợ của trường) hoặc ngày CTXH/điểm rèn luyện (sinh viên của trường).
- Nhân viên IT: nguyện vọng phát triển một phần mềm phù hợp để giải quyết vấn đề in ấn của sinh viên.

- Đại lý cung cấp giấy, mực in và kỹ thuật viên bảo trì: nhu cầu tìm kiếm một nơi tiêu thụ sản phẩm/dịch vụ, một đối tác ổn định và lâu dài.
- Các hệ thống thanh toán online: có nhu cầu thu hút thêm người sử dụng và thu được hoa hồng thông qua các kênh thanh toán.

1.1.3 Lợi ích SPSS mang lại cho những bên liên quan

Các bên liên quan (stakeholder) khi tham gia vào một hệ thống bên cạnh việc đáp ứng nhu cầu cấp thiết của khách hàng thì họ đều cần một lợi ích chính đáng. Lợi ích của các bên liên quan trong SPSS:

- Sinh viên sử dụng dịch vụ: in ấn tài liệu thuận tiện, chủ động về thời gian và tiết kiệm chi phí in với các chương trình giảm giá và trợ giá từ trường.
- Trường đại học Bách Khoa: cải thiện chất lượng cơ sở vật chất cho sinh viên, thu lợi nhuận thông qua việc quản lý và cung cấp dịch vụ in ấn.
- Người quản lý máy in và thực hiện in ấn: Kiểm thêm thu nhập hoặc ngày CTXH/ĐRL ngoài giờ.
- Nhân viên IT: phát triển và duy trì hệ thống giúp họ có cơ hội phát triển lớn hơn về nghề nghiệp trong tương lai.
- Đại lý cung cấp giấy, mực in và kỹ thuật viên bảo trì: Kiểm thêm lợi nhuận và có được khách hàng lâu dài thông qua sự hợp tác của đại lý và nhà trường.
- Các hệ thống thanh toán online: cung cấp thêm dịch vụ để thu hút thêm người sử dụng, tạo ra hoa hồng thông qua các giao dịch.

1.2 Mô tả tất cả yêu cầu chức năng và phi chức năng có thể suy ra từ tổng thể bài toán

1.2.1 Yêu cầu chức năng (functional requirements) (Ít nhất 5 yêu cầu chức năng cho mỗi bên liên quan)

Sinh viên (user):

- Đăng ký/đăng nhập tài khoản.
- Thay đổi mật khẩu.
- Đặt yêu cầu in ấn từ xa.
- Chọn máy in và theo dõi trạng thái in ấn.
- Kiểm tra thông báo về tình trạng của tài liệu đã gửi.
- Theo dõi lịch sử in ấn của mình.

Người quản lý hệ thống (SPSO):

- Theo dõi lịch sử in ấn của sinh viên.
- Quản lý hoạt động của máy in trong hệ thống như thêm/xóa/điều chỉnh trạng thái hoạt động, thông tin định danh của máy in.
- Quản lý các thông số khác của hệ thống như số trang mặc định, ngày gửi tài liệu, các kiểu tài liệu có thể phục vụ.
- Hủy tài liệu trong trường hợp học sinh không đến lấy đúng thời hạn.
- Theo dõi báo cáo về việc sử dụng hệ thống in ấn sau mỗi tháng và mỗi năm.

Người quản lý máy in (trường hợp flow truyền thông):

- Tiếp nhận đơn đặt in
- In ấn tài liệu theo yêu cầu
- Sắp xếp tài liệu và gửi sinh viên

Đơn vị cung cấp giấy, mực in và kỹ thuật viên bảo trì:

- Cung cấp giấy in, mực in cho hệ thống trong trường hợp hết giấy, hết mực.
- Điều động kỹ thuật viên đến bảo trì trang thiết bị định kỳ và kịp thời khi xuất hiện lỗi kỹ thuật.

1.2.2 Yêu cầu phi chức năng (non-functional requirements)

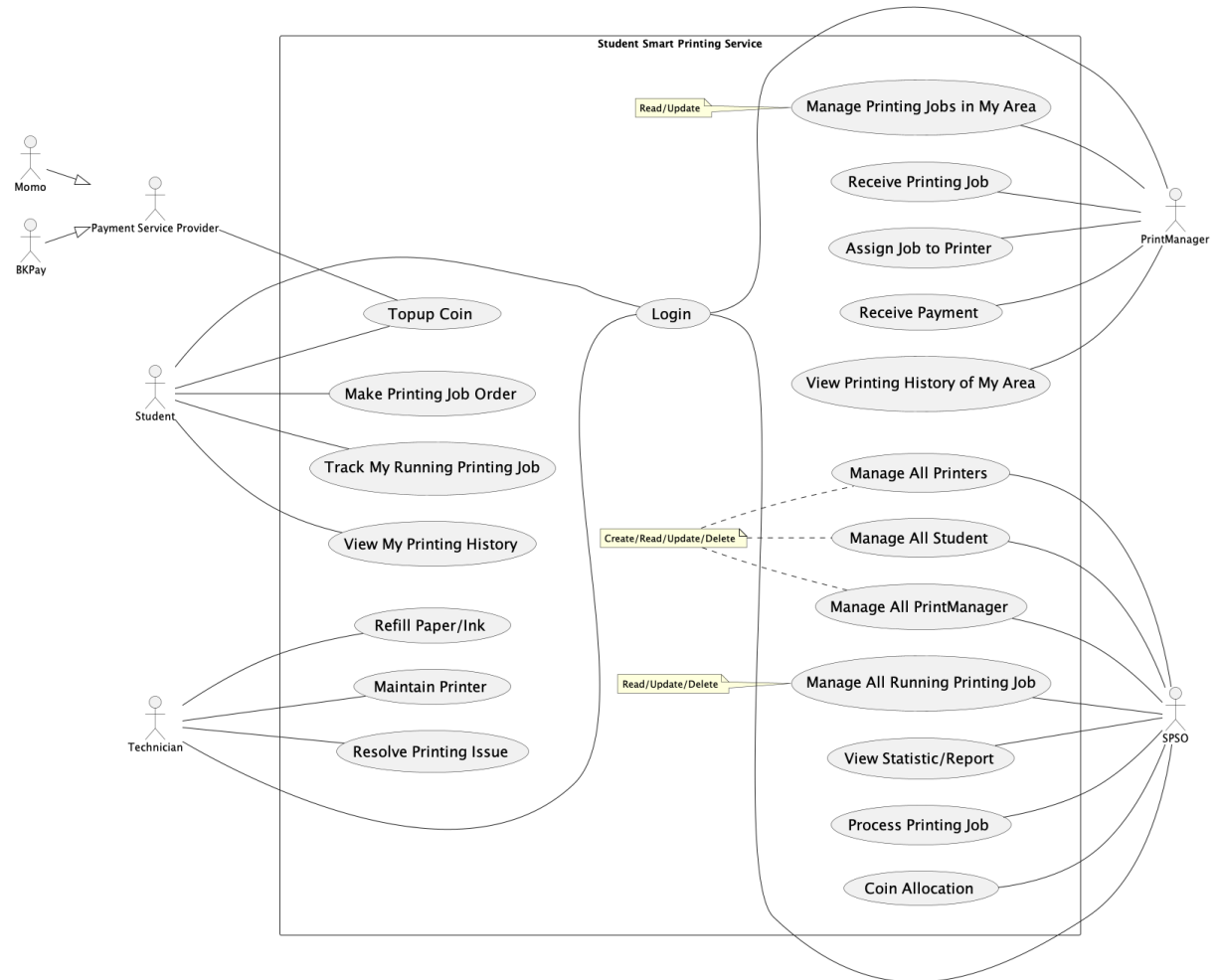
Yêu cầu phi chức năng liên quan đến việc xác định những ràng buộc trong dịch vụ hoặc chức năng có ảnh hưởng đến hệ thống như ràng buộc về thời gian, quá trình phát triển, hiệu suất và thường áp dụng tới toàn bộ hệ thống hơn là những đặc điểm hay dịch vụ riêng lẻ. Đối với SSPS, chúng ta có những yêu cầu phi chức năng như sau:

- Sự hiệu quả (Efficiency): Trang web/app đảm bảo đáp ứng yêu cầu in ấn của sinh viên với nhiều loại tài liệu khác nhau như giáo trình, đề thi, nghiên cứu khoa học,... (hỗ trợ in màu). Cung cấp thông tin về tình trạng sử dụng trang thiết bị, chi phí sử dụng và tài khoản cá nhân để người dùng chủ động sắp xếp, đưa ra quyết định phù hợp nhất.
- Tính khả dụng (Availability): Trang web/app hoạt động tốt và an toàn 24/7 để phục vụ yêu cầu đặt in và kiểm soát quá trình in ấn của các máy in.
- Độ tin cậy (Reliability): Hệ thống phải có cơ chế sao lưu dữ liệu định kỳ hoặc khi xuất hiện lỗi kỹ thuật, có khả năng thống kê, ghi nhận lại lỗi phát sinh trong quá trình hoạt động. Tất cả tác vụ phải được hoàn thành theo yêu cầu đặt in của người dùng, hoặc có cơ chế thông báo và hướng dẫn xử lý rõ ràng nếu không thành công.
- Tính dễ sử dụng (Ease Of Use): Cung cấp hướng dẫn sử dụng bằng một bản hướng dẫn, xử lý các tình huống phát sinh và có thể thuận tiện thao tác chỉ với sự hướng dẫn/điều hướng và tài liệu của ứng dụng. Người dùng không cần thiết lập thông số kỹ thuật để kết nối với máy in, chỉ cần khoảng từ 5-10 phút đọc hướng dẫn để có thể sử dụng dịch vụ một cách mượt mà nhất
- Thời gian phản hồi nhanh chóng (Fast response): Đảm bảo thời gian từ lúc người dùng tải tài liệu lên cho đến khi hệ thống xác nhận và đưa vào hàng đợi từ 1-2(s) cho trường hợp chậm nhất.
- Giới hạn kiểu tài liệu (Limited files types): Hệ thống chỉ chấp nhận những file có độ lớn tối đa là 100MB và chỉ chấp nhận tài liệu dưới dạng PDF, JPEG, PNG, DOCX.

1.3 Use Case Diagram

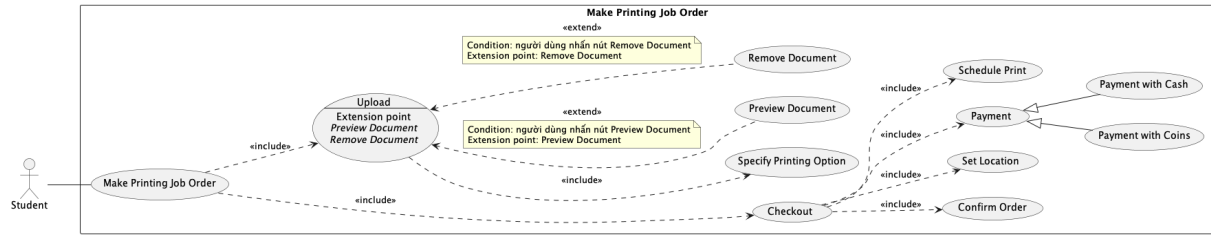
1.3.1 Whole System

Draw a use case diagram for whole System



Hình 1: Use Case Diagram for the Whole System

1.3.2 Use case Order Printing Job



Hình 2: Use Case Diagram for Order Printing Job

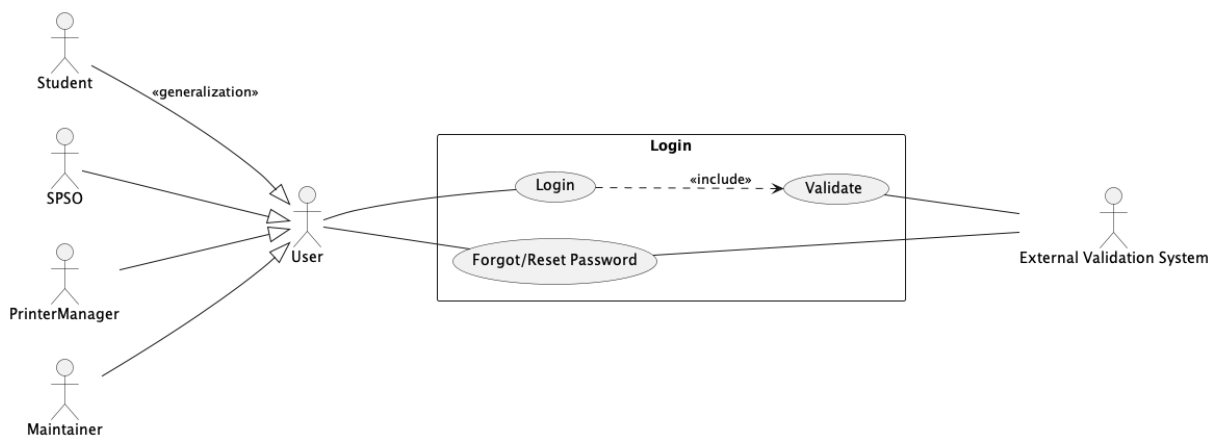
Order Printing Job scenario

Use Case Name	Order Printing Job
Actors	Student, Payment Service Provider.
Brief Description	Sinh viên thực hiện chức năng Order print để đặt in tài liệu trong hệ thống.
Preconditions	Sinh viên đã đăng nhập vào hệ thống và có tài liệu cần in.
Postconditions	Sinh viên đã đặt in tài liệu thành công và được lên lịch in trong hệ thống.
Trigger	Sinh viên nhấn nút Order print và chọn tài liệu cần in.
Normal Flow	<ol style="list-style-type: none"> Sau khi sinh viên hoàn tất việc upload file, hệ thống sẽ lưu file trên cloud storage là MinIo. Hệ thống sẽ lưu thông tin về các cài đặt in của file cùng với một vài thông tin khác của file(tên, kích cỡ,...) thành metadata. Thông tin metadata của file sẽ được dùng để truy xuất ngược lại file ở MinIo khi sinh viên tiến hành preview file hoặc remove file. Hệ thống sẽ tiến hành tạo model cho metadata bằng Prisma ORM. Model của metadata sẽ được lưu vào database là Postgres. Sau khi lưu thành công, hệ thống sẽ trả về cho sinh viên một đường dẫn để sinh viên có thể preview file. Sinh viên có thể xóa file nếu muốn. Sau khi sinh viên hoàn tất việc chọn file, hệ thống sẽ tiến hành tính toán số lượng coins mà sinh viên phải trả cho đơn hàng. Khi sinh viên xác nhận hoàn tất đơn hàng, hệ thống sẽ thực hiện hai việc: <ol style="list-style-type: none"> Update thông tin về số lượng coins thông qua model người dùng. Tạo model cho đơn hàng bằng Prisma ORM. Model này sẽ bao gồm các thông tin mà sinh viên đã cài đặt cho đơn hàng (số lượng bản in, màu sắc, giấy,...). Hệ thống sẽ lưu model của đơn hàng vào database là Postgres. Hệ thống sẽ trả về cho sinh viên một đường dẫn để sinh viên có thể xem lại thông tin đơn hàng.
Exceptional flow	Ở bước 10a, Nếu số lượng coins mới của sinh viên không đủ, đơn hàng sẽ không được đặt và thông báo hiển thị cho sinh viên, sinh viên có thể chọn đến trang nạp tiền để nạp thêm tiền vào tài khoản và quay lại bước 10.

Alternative flow	Ở bước 8 , Nếu sinh viên xác nhận xóa tài liệu sau khi nó tải lên, họ có thể xóa tài liệu đó và tải lên một tài liệu khác.
Includes	Upload Document, Preview Document, Remove Document, Specify Printing Option, Checkout, Confirm Order, Set Location, Payment, Schedule Print.

Bảng 1: Order Printing Job scenario

1.3.3 Use case *Login Service*

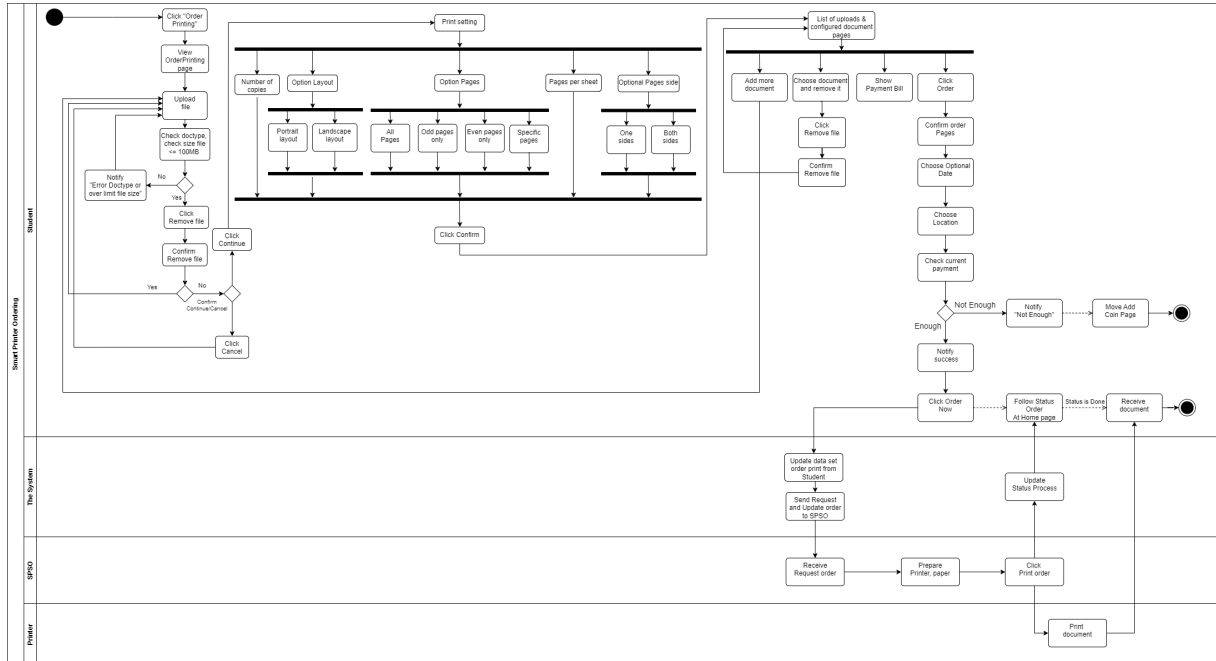


Hình 3: Use Case Diagram for Login Service

2 System modelling

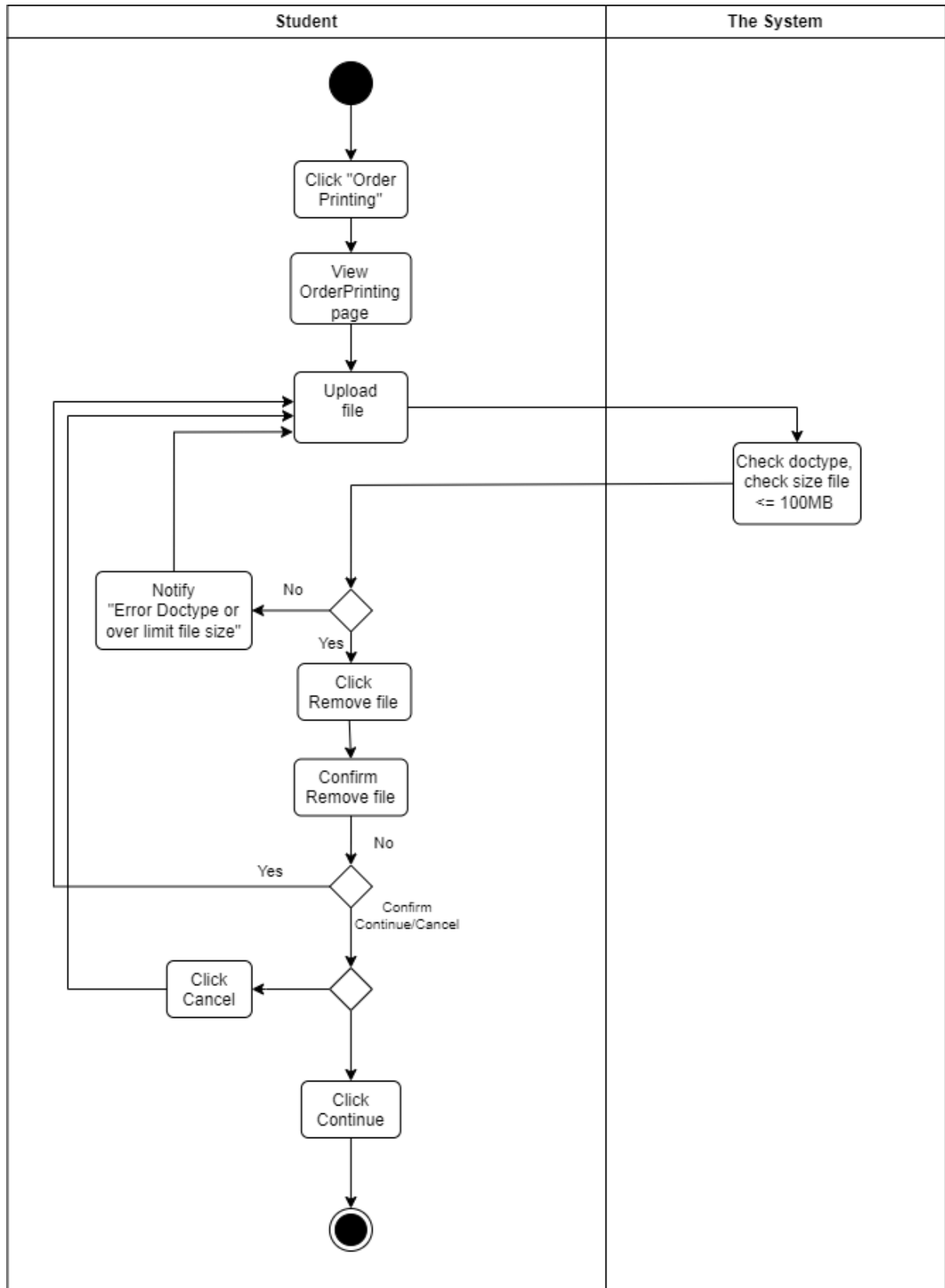
2.1 Activity diagram

Draw an activity diagram to capture the business process between systems and the stakeholders in Task Assignment module



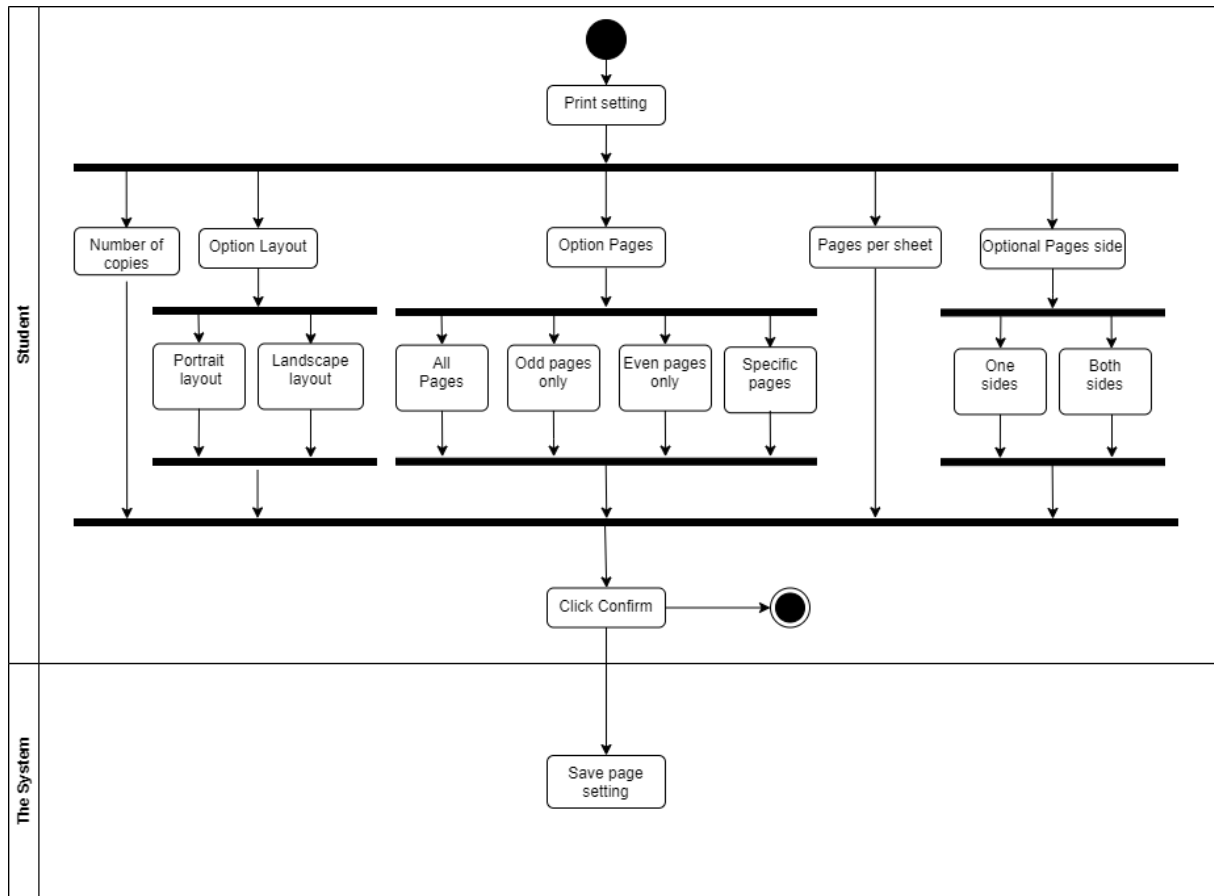
Hình 4: Activity diagram for Order Printing

Activity Diagram for Upload Document



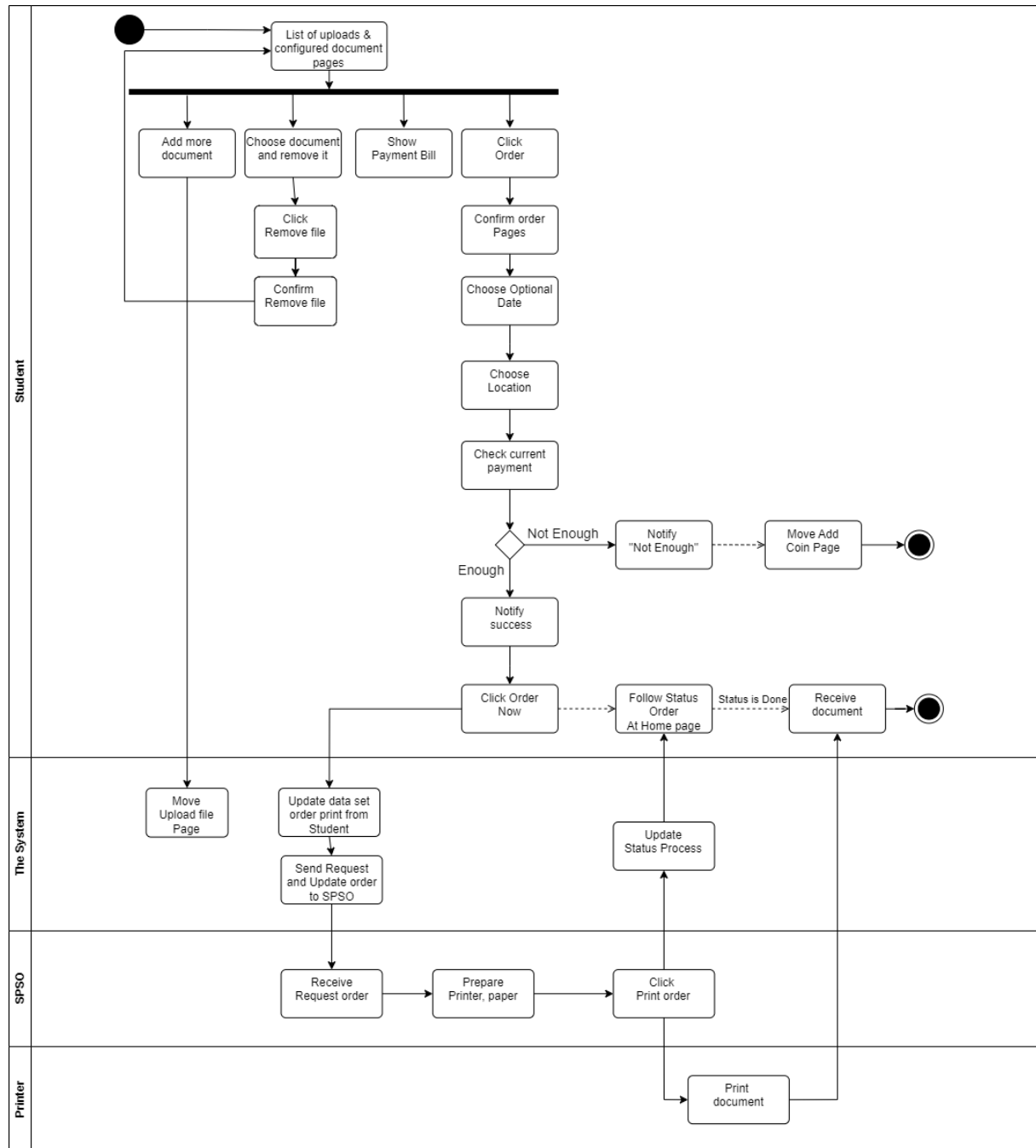
Hình 5: Activity diagram for Upload File

Activity Diagram for Page Setting



Hình 6: Activity diagram for Page Setting

Activity Diagram for Confirm Order



Hình 7: Activity diagram for Confirm Order

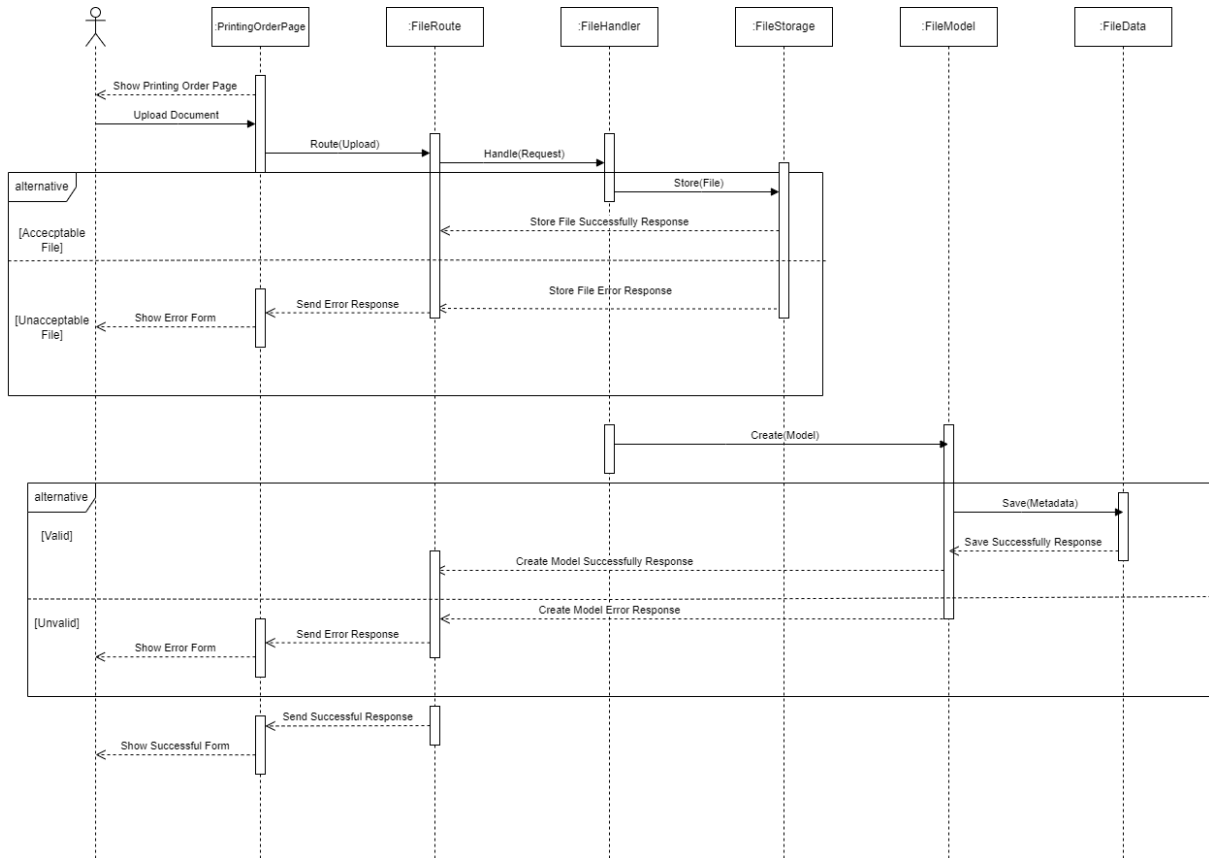
Mô tả:

- Sau khi đăng nhập, sinh viên sẽ ở trang Home page.
- Lúc này sinh viên có thể chọn vào button “Order Printing” để thực hiện đặt in tài liệu.
- Sinh viên sẽ được di chuyển đến trang OrderPrinting, sinh viên có thể Upload file cần in lên. Sinh viên có thể upfile lên hệ thống, hệ thống sẽ kiểm tra loại doctype của tài liệu và size file $\leq 100\text{MB}$. Nếu thỏa điều kiện thì sinh viên Upload file thành công, nếu không thành công hệ thống sẽ thông báo để sinh viên Upload lại.
- Sau khi Upload file thành công, sinh viên có thể chọn file đó và remove file, hệ thống sẽ cho phép sinh viên Upload file khác sau khi remove file.

- Tiếp theo sinh viên sẽ thiết lập những thông số in bao gồm: Number of copies, Option Layout (Portrait layout Landscape layout), OptionPages (All Pages, Odd pages only, Even pages Only, Specific pages), Pages per sheet, Optinal Pages side (One sides, Both sides), etc.
- Sau khi đã thiết lập những thông số in, sinh viên click vào button “Confirm” thì hệ thống đưa sinh viên đến trang List of uploads & configured document.
- Tại trang này, sinh viên có thể tiếp tục Thêm tài liệu in, Chọn tài liệu và xóa, Thông tin giá tiền đơn hàng, Button “Order” để tiến hành quá trình thanh toán và sinh viên được di chuyển đến trang Confirm order.
- Ở trang Confirm Order, sinh viên tiếp tục chọn Date để giao in, Location để giao và Thanh toán đơn hàng bằng coin.
- Nếu số dư coin trong tài khoản sinh viên còn dư, sinh viên sẽ được phép thanh toán và sinh viên chọn Order Now hệ thống sẽ thông báo đơn hàng đã được đặt thành công và sinh viên được di chuyển đến trang Home page.
- Nếu số dư coin trong tài khoản sinh viên không đủ, sinh viên sẽ không được phép thanh toán và hệ thống sẽ thông báo cho sinh viên, sinh viên có thể di chuyển đến trang nạp thêm coin để thanh toán.
- Sau khi click vào Order Now, sinh viên được chuyển về Home page, để theo dõi trạng thái (status) khi in. Và lúc này hệ thống sẽ gửi một Trigger Update data set order print đến hệ thống, và gửi Request and Update order đến SPSO.
- Sau khi nhận được request order, SPSO sẽ chuẩn bị máy in và giấy, Click vào Print order để máy in (Printer) in tài liệu và gửi Trigger Update status process lên hệ thống.
- Sau khi có tài liệu, sinh viên theo dõi trạng thái (Done) in tài liệu có thể đến địa điểm đã confirm để nhận tài liệu.

2.2 Sequence diagram

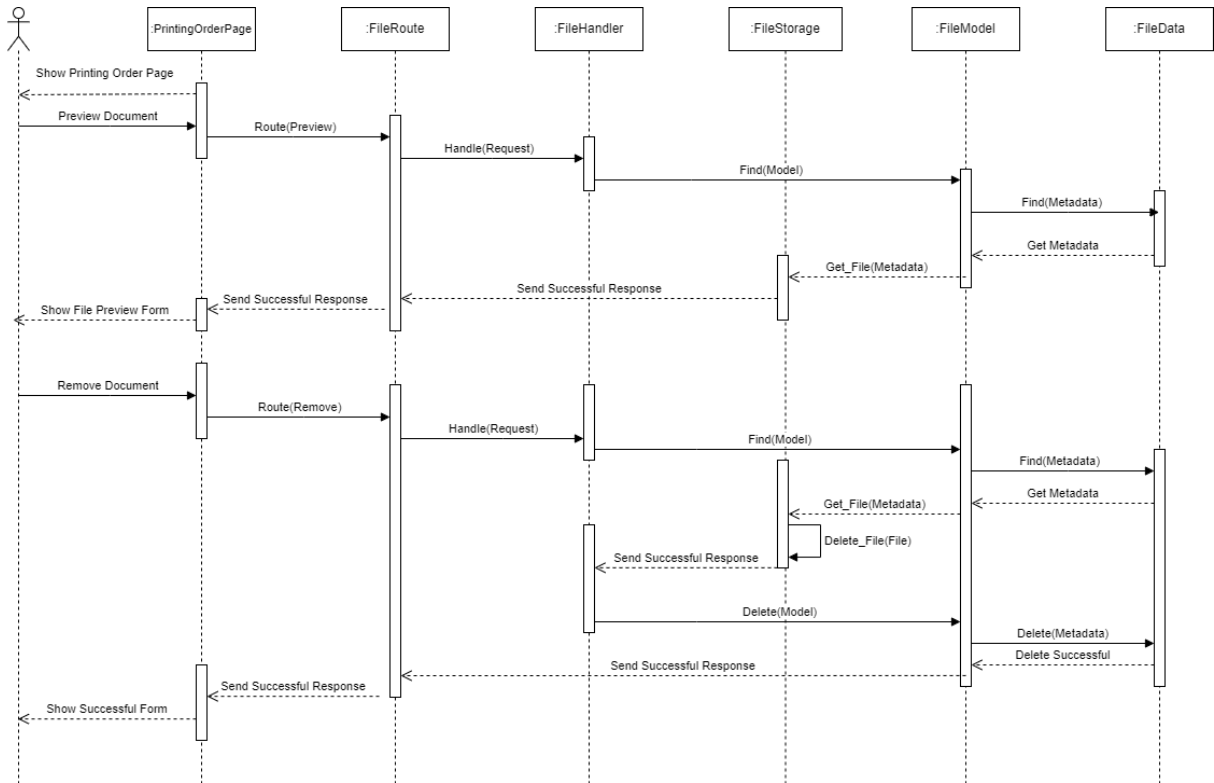
Draw an Sequence diagram to capture the business process between systems and the stakeholders in Task Assignment module



Hình 8: Sequence diagram of Upload file

Mô tả về quá trình upload file của sinh viên lên hệ thống.

- Sau khi sinh viên lựa chọn file cần in và chỉnh sửa các thông số in, sinh viên sẽ click vào button “Confirm” để tiến hành đặt in.
- Object PrintingOrderPage sẽ gọi method Route(Upload) của Object FileRouter để xử lý.
- Object FileRouter sẽ gọi method Handler(Request) của Object FileHandler để xử lý.
- Object FileHandler sẽ gọi method Store(File) của Object FileStorage để xử lý.
 - Nếu file được lưu thành công, Object FileHandler sẽ gọi method Store(Metadata) của Object FileModel để xử lý.
 - Nếu file được lưu không thành công, Object FileStorage sẽ trả về kết quả thất bại cho Object FileRouter và các Object còn lại lần lượt trả về kết quả thất bại.
- Nếu model được tạo thành công, Object FileModel sẽ gọi method Save(File) của Object FileDatabase để xử lý.
- Nếu model được tạo không thành công, Object FileModel sẽ trả về kết quả thất bại cho Object FileHandler và các Object còn lại lần lượt trả về kết quả thất bại.



Hình 9: Sequence diagram of Preview and Remove file

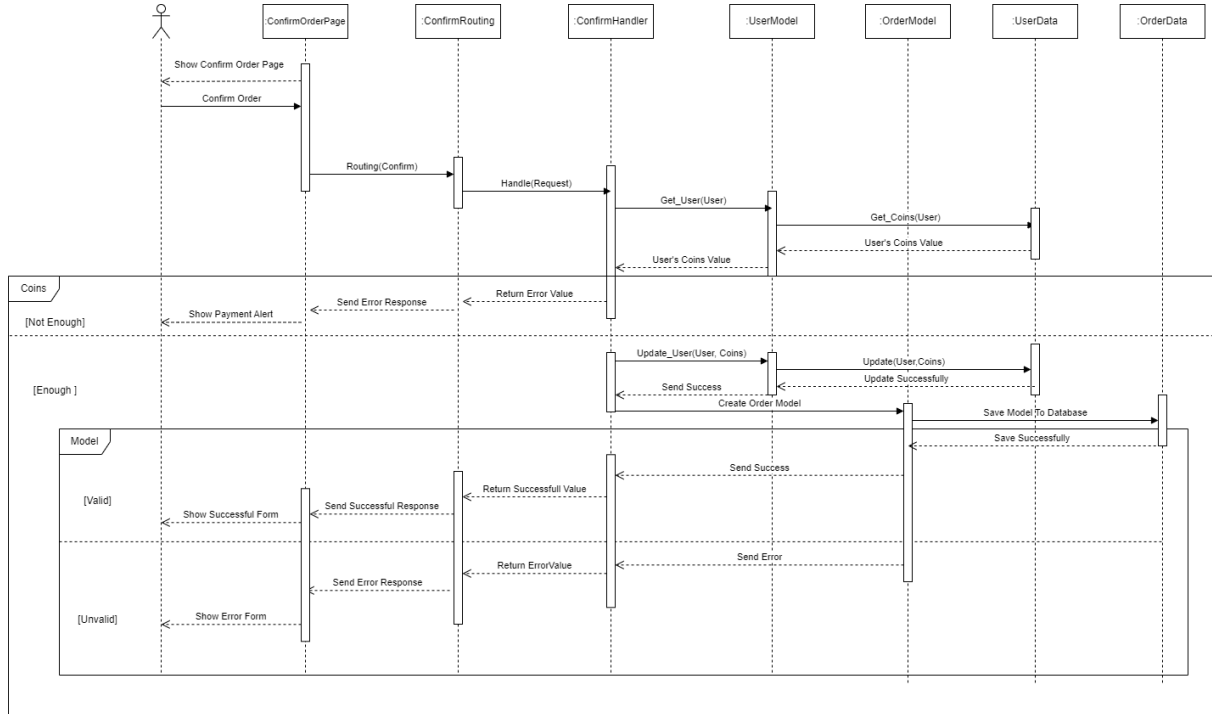
Mô tả về quá trình preview file của sinh viên lên hệ thống.

- Sau khi sinh viên upload file lên hệ thống, sinh viên sẽ được chuyển đến trang List of uploads & configured document.
- Object PrintingOrderPage sẽ gọi method Route(Preview) của Object FileRouter để xử lý.
- Object FileRouter sẽ gọi method Handler(Request) của Object FileHandler để xử lý.
- Object FileHandler sẽ gọi method Find(Model) của Object FileModel để xử lý.
- Object FileModel sẽ gọi method Find(Metadata) của Object FileDatabase để xử lý.
- Object FileDatabase sẽ trả về kết quả thành công cho Object FileModel.
- Object FileModel sẽ trả về kết quả thành công cho Object FileHandler.
- Object FileHandler sẽ trả về kết quả thành công cho Object FileRouter.
- Object FileRouter sẽ trả về kết quả thành công cho Object PrintingOrderPage.
- Object PrintingOrderPage sẽ hiển thị thông tin của file cho người dùng.

Mô tả về quá trình remove file của sinh viên lên hệ thống.

- Sau khi sinh viên upload file lên hệ thống, sinh viên sẽ được chuyển đến trang List of uploads & configured document.
- Object PrintingOrderPage sẽ gọi method Route(Remove) của Object FileRouter để xử lý.
- Object FileRouter sẽ gọi method Handler(Request) của Object FileHandler để xử lý.
- Object FileHandler sẽ gọi method Find(Model) của Object FileModel để xử lý.
- Object FileModel sẽ gọi method Find(Metadata) của Object FileDatabase để xử lý.
- Object FileDatabase sẽ trả về Metadata của file cho Object FileModel.
- Object FileStorage sẽ tự gọi method Delete(File) của mình để xóa file ra khỏi cloud storage.
- Object FileModel sẽ trả về kết quả thành công cho Object FileHandler.
- Object FileHandler sẽ gọi method Delete(Model) của Object FileModel để xử lý.
- Object FileModel sẽ gọi method Delete(Metadata) của Object FileDatabase để xử lý.
- Object FileDatabase sẽ trả về kết quả thành công cho Object FileModel.

- Object FileModel sẽ trả về kết quả thành công cho Object FileHandler.
- Object FileHandler sẽ trả về kết quả thành công cho Object FileRouter.
- Object FileRouter sẽ trả về kết quả thành công cho Object PrintingOrderPage.
- Object PrintingOrderPage sẽ hiển thị thông báo thành công cho người dùng.



Hình 10: Sequence diagram of Checkout

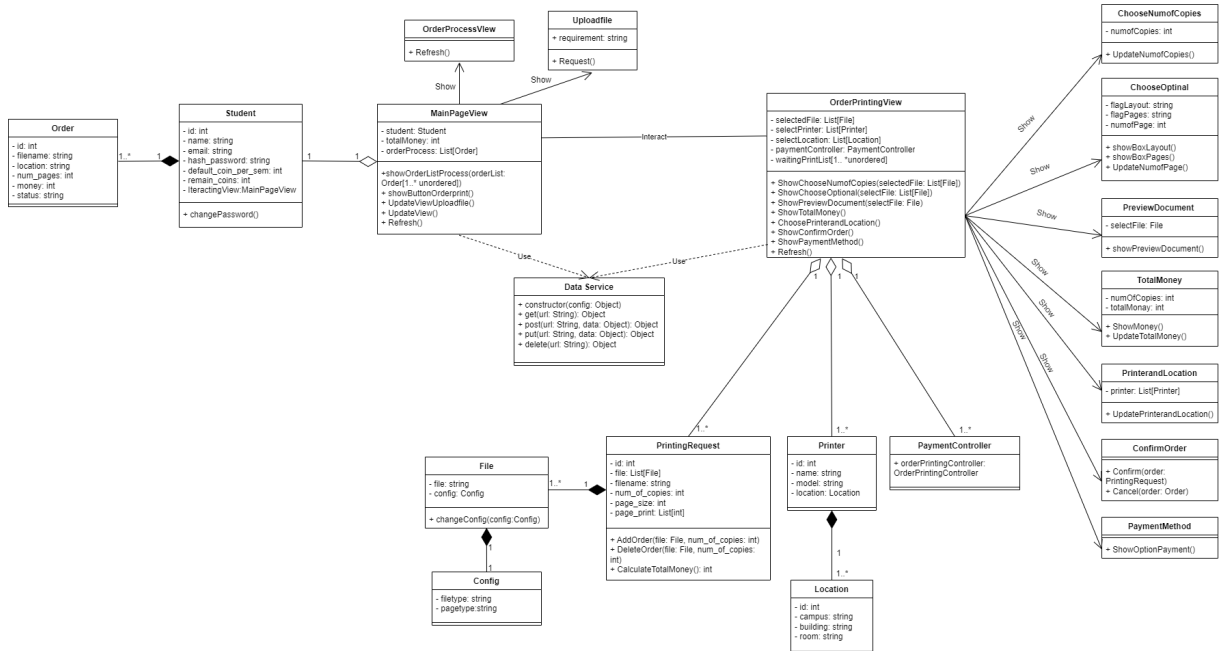
Mô tả về quá trình Confirm Order của sinh viên lên hệ thống.

- Sau khi sinh viên lựa chọn file cần in và chỉnh sửa các thông số in (lich in, vị trí phòng in, phương thức thanh toán), sinh viên sẽ click vào button “Confirm” để tiến hành đặt in.
- Object PrintingOrderPage sẽ gọi method Route(Checkout) của Object OrderRouter để xử lý.
- Object OrderRouter sẽ gọi method Handler(Request) của Object OrderHandler để xử lý.
- Object OrderHandler sẽ gọi method Get_User(User) của Object UserModel để lấy ra model của sinh viên.
- Object UserModel sẽ gọi method Get_Coins(User) của Object UserDatabase để lấy ra số coins của sinh viên.
- Object UserDatabase sẽ trả về số coins của sinh viên cho Object UserModel.
- Object UserModel sẽ trả về kết quả thành công cho Object OrderHandler.
- 1. Nếu số coins của sinh viên không đủ để thanh toán, Object OrderHandler sẽ trả về kết quả thất bại cho Object OrderRouter và các Object còn lại lần lượt trả về kết quả thất bại.
- 2. Nếu số coins của sinh viên đủ để thanh toán, Object OrderHandler sẽ tiến hành cập nhật lại số coins của sinh viên thông qua method Update(User) của Object UserModel.
- Object OrderHandler sẽ gọi method Create(Model) của Object OrderModel để tạo model của order.
- Object OrderModel sẽ gọi method Create(Order) của Object OrderDatabase để tạo order.
- 1. Nếu order được tạo thành công, Object OrderModel sẽ gọi method Save(Order) của Object OrderDatabase để lưu order.
- 2. Nếu order được tạo không thành công, Object OrderDatabase sẽ trả về kết quả thất bại cho Object OrderModel và các Object còn lại lần lượt trả về kết quả thất bại.

2.3 Class diagram

Draw a class diagram of the printer order system.

Frontend Class diagram: Các class trong sơ đồ UML class diagram đều sử dụng chung 1 class là DataService. Class DataService có nhiệm vụ gọi các API cung cấp bởi phía server.



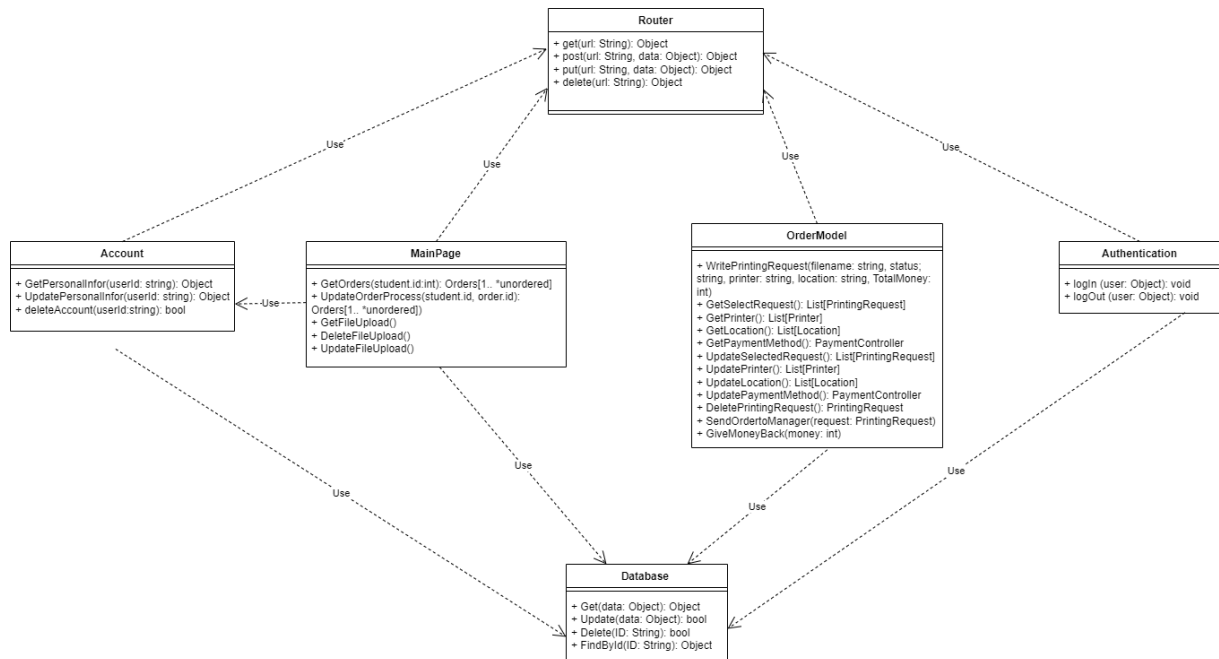
Hình 11: Frontend Class diagram

Mô tả

- Đối tượng **Student** bao gồm các thuộc tính: id, name, email, hash_password, default_coin_per_sem, remain_coins, và InteractingView với Main Page View và **Student** có phương thức changePassword().
- Đối tượng **Order** bao gồm các thuộc tính: id, filename, location, num_pages, money, status.
- Đối tượng **File** bao gồm các thuộc tính: file, config: Config (filetype, pagetype), và phương thức changeConfig().
- Đối tượng **Location** bao gồm các thuộc tính: id, campus, building, room.
- Đối tượng **Printer** bao gồm các thuộc tính: id, name, model, location(Location).
- Đối tượng **PrintingRequest** bao gồm các thuộc tính: id, List, filename, num_of_copies, page_size, page_print(List) và phương thức AddOrder(file, num_of_copies), DeleteOrder(file, num_of_copies), CalculateMoney().
- **Student** có thể tương tác với **MainPageView**. **MainPageView** bao gồm các thuộc tính student: Student, totalMoney, orderProcess: List. Các phương thức của **MainPageView** bao gồm: ShowOrderListProcess(order:List), showButtonOrderPrint(), Uploadfile() và Refresh().
- Sau khi **Student** upload file, **MainPageView** sẽ chuyển hướng đến **OrderPrintingView** để hiển thị thông tin về file vừa upload và tùy chọn các thông số in ấn. **OrderPrintingView** bao gồm các thuộc tính: selectedFile: List, selectPrinter: List, selectLocation: List, paymentController: PaymentController, waitingPrintList< 1..unordered> và các phương thức: ShowFileUpload(selectedFile: List), ShowChooseNumOfCopies(selectedFile: List), ShowPreviewDocument(selectFile: File),

ShowChooseOptimal(selectedFile: List), ShowTotalMoney(), ChoosePrinterandLocation(), ShowConfirmOrder(), ShowPatmentMethod().

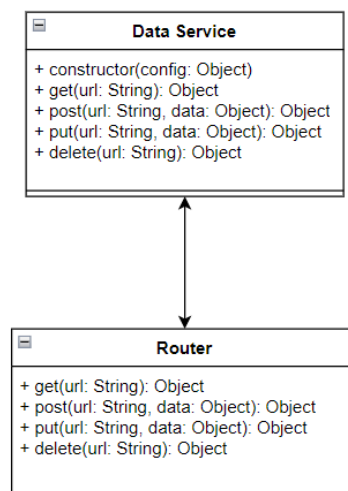
Class Diagram cho Business Logic Layer Tương tự vai trò của class DataService bên phía presentation, class Router phía business logic đóng vai trò tiếp nhận và điều hướng các lời gọi API từ phía presentation layer tới các logic xử lý trong hệ thống. Ngoài Router, các lớp đều sử dụng class Database để thực hiện thao tác dữ liệu trực tiếp từ cloud database.



Hình 12: Backend Class diagram

Kết nối giữa frontend và backend

Class DataService là đại diện bên frontend, chuyên gửi các request và nhận dữ liệu trả về từ backend. Tương tự, class Router là đại diện bên backend, chuyên tiếp nhận các request từ frontend, điều hướng tới các logic xử lý nghiệp vụ, và trả về dữ liệu tương ứng với request nhận được.



Hình 13: FE_BE Connect diagram

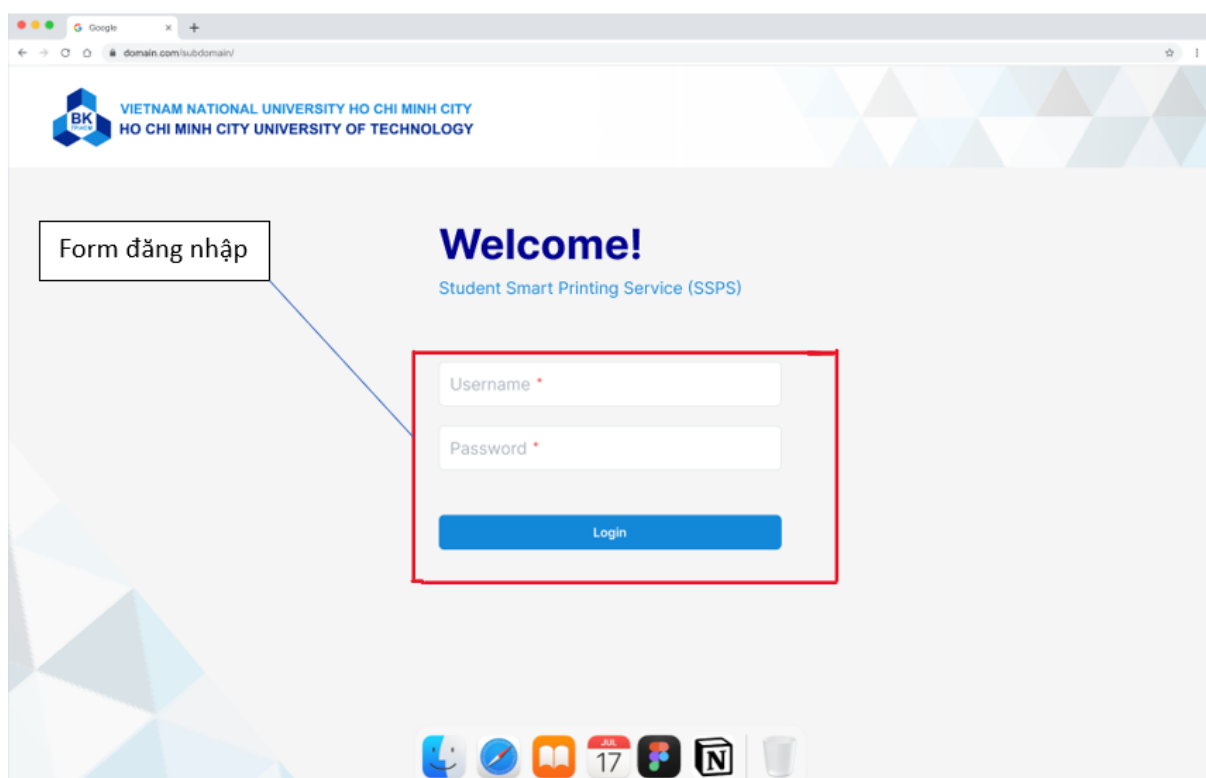
2.4 User interfaces

MVP 1 - UI for desktop-view central dashboard for Student to Order Print

Ở Task này, nhóm đã sử dụng Figma để xây dựng giao diện cho các thao tác cơ bản của Student để Order Print.

Cụ thể phần hiển thị thực Figma ở link sau: <https://www.figma.com/file/7eKhLKrVARC1YBJPBpM6dO/%5BDraft%5D-Hi-fi-design?type=design&node-id=0-1&mode=design&t=bq6clbMcHuCwIvk0-0>

2.4.1 Trang đăng nhập (Login Page)



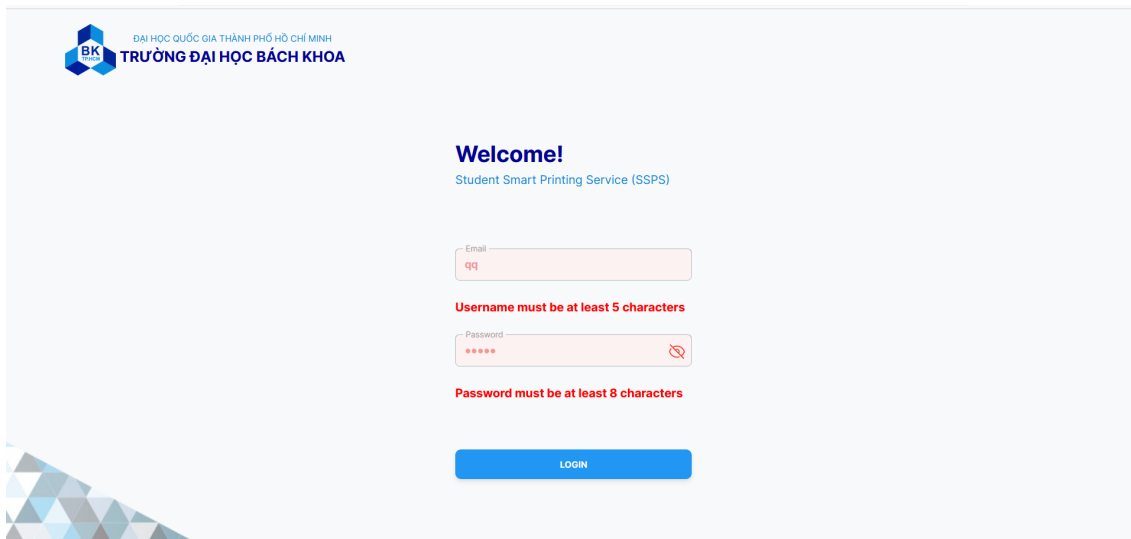
Hình 14: Giao diện trang đăng nhập

Trang đăng nhập của hệ thống có form đăng nhập với các trường sau đây:

- Trường đăng nhập: Người dùng nhập tên tài khoản
- Trường mật khẩu: Người dùng nhập mật khẩu đã được lưu trữ
- Nút Login (Đăng nhập): Người dùng nhấn nút **Login** sau khi điền thông tin (tài khoản, mật khẩu), thông tin này sẽ được gửi sang bên thứ ba để xác thực.

Nếu đăng nhập thành công, người dùng sẽ được chuyển hướng sang trang chủ (Homepage) (chi tiết ở mục 2.4.2) Nếu đăng nhập không thành công, sẽ hiển thị ra các lỗi tương ứng như sau

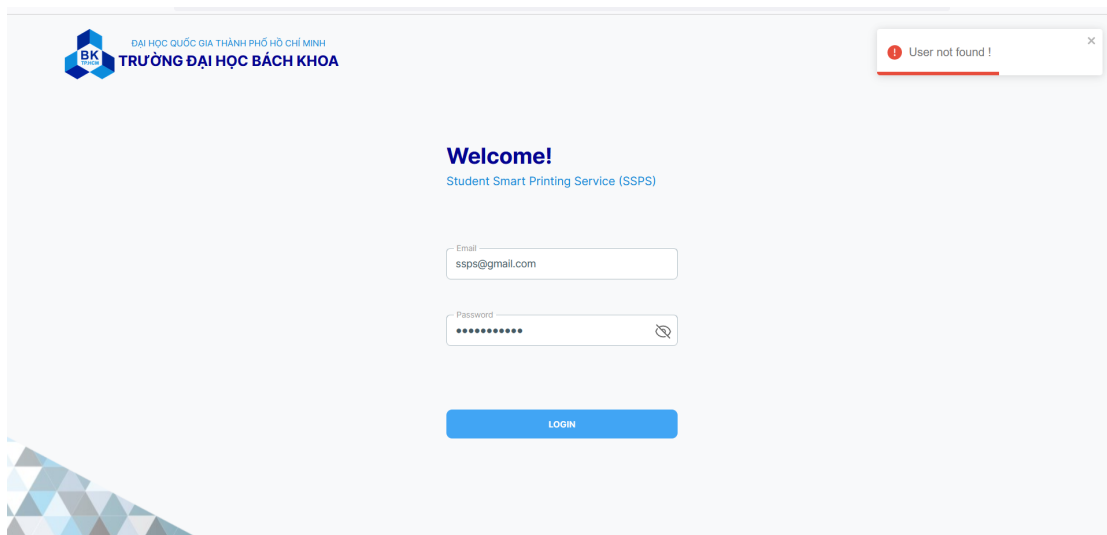
- Người dùng nhập các thông tin không thỏa mãn điều kiện của các trường (Tên người dùng ít nhất là 5 ký tự, mật khẩu ít nhất 8 ký tự)



The image shows a login page for the Student Smart Printing Service (SSPS) at the University of Science, Ho Chi Minh City. The page has a header with the university's logo and name. Below the header, there is a 'Welcome!' message and the service name. The login form consists of two input fields: 'Email' and 'Password'. The 'Email' field contains the text 'qq' and has a red error message below it: 'Username must be at least 5 characters'. The 'Password' field contains eight dots and has a red error message below it: 'Password must be at least 8 characters'. A blue 'LOGIN' button is located below the password field.

Hình 15: Lỗi khi người dùng nhập thông tin không thỏa mãn các trường

- Người dùng đăng nhập sai thông tin về tài khoản hoặc mật khẩu: hiển thị lỗi **User not found**

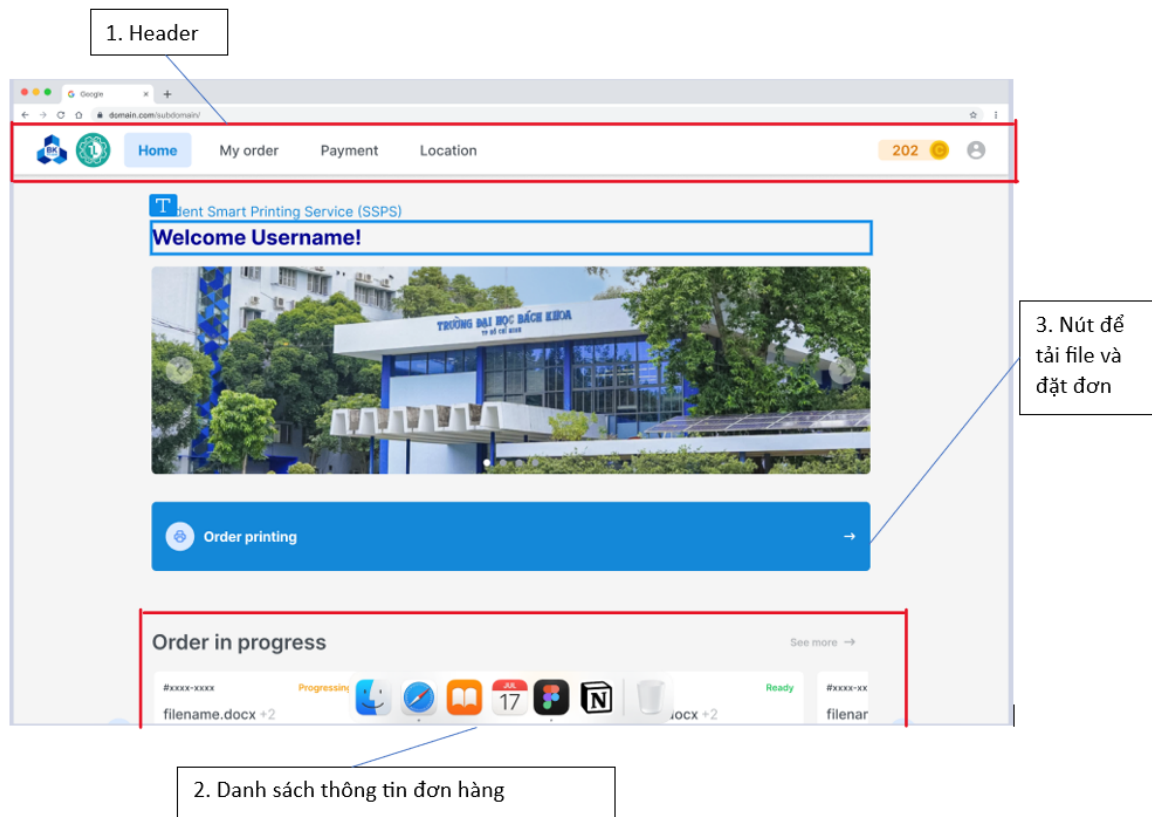


The image shows the same login page as in Figure 15, but with a different error message. The 'Email' field now contains 'ssps@gmail.com' and the 'Password' field contains eight dots. A red error message 'User not found !' is displayed in a box at the top right of the page. The 'LOGIN' button is still present below the password field.

Hình 16: Lỗi khi người dùng nhập sai thông tin được lưu trữ

2.4.2 Trang chủ (Home Page)

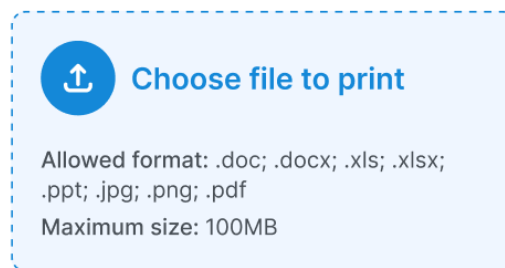
Để có thể truy cập được vào trang chủ của hệ thống, người dùng đã đăng nhập thành công.



Hình 17: UI Home of Desktop View

Các thành phần chính của trang chủ hệ thống (Homepage):

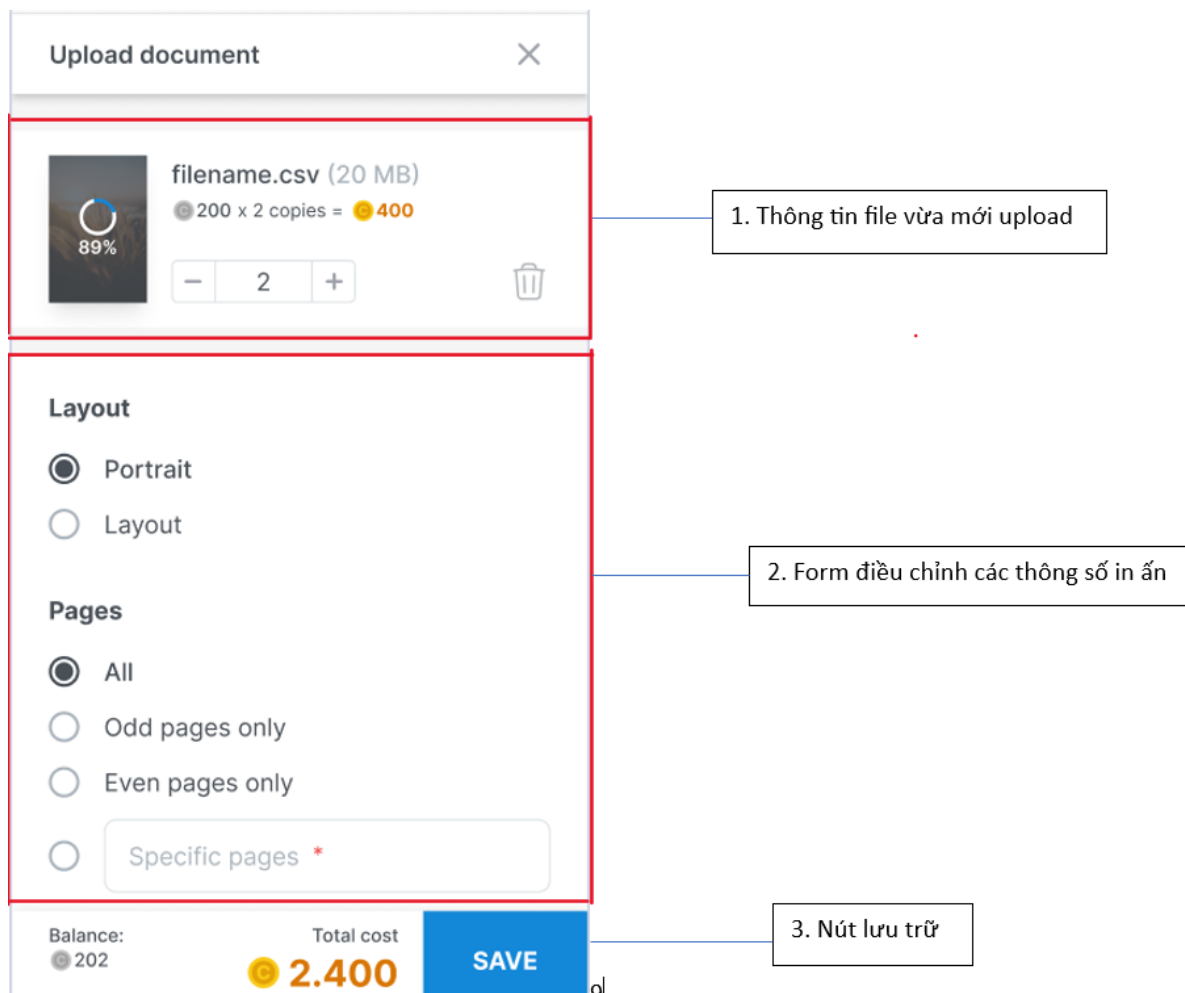
- 1. Header:** biểu diễn những thông tin chính của trang chủ (thành phần cố định trong các trang)
 - Home: Điều hướng người dùng về trang chủ.
 - My order: Điều hướng người dùng đến trang theo dõi danh sách đơn hàng đã được lưu trên hệ thống (chi tiết ở mục 2.4.7).
 - Payment: Điều hướng đến trang thanh toán đơn hàng (chi tiết ở mục 2.4.6).
 - Location: Điều hướng đến trang vị trí máy in trong khuôn viên của trường (chi tiết ở mục 2.4.).
 - Coin: Hiển thị số coin hiện tại của người dùng.
- 2. My Order:** Nơi người dùng có thể xem lại thông tin của các đơn đặt hàng của mình về tình trạng, tên file, số lượng,... Khi nhấn vào **See more**, người dùng sẽ được điều hướng đến trang danh sách đơn hàng (chi tiết ở mục 2.4.7)
- 3. Order Printing:** Khi sinh viên nhấn vào nút này, người dùng được trả về một box file (nơi giúp người dùng có thể tải file lên hệ thống và sau đó được định hướng sang trang tải tài liệu, chi tiết ở mục 2.4.3).



Hình 18: Giao diện Box file

2.4.3 Trang tải file lên hệ thống

Để có thể truy cập được vào trang tải file lên hệ thống, người dùng đã nhấn vào nút **Order printing** ở trang chủ hệ thống.



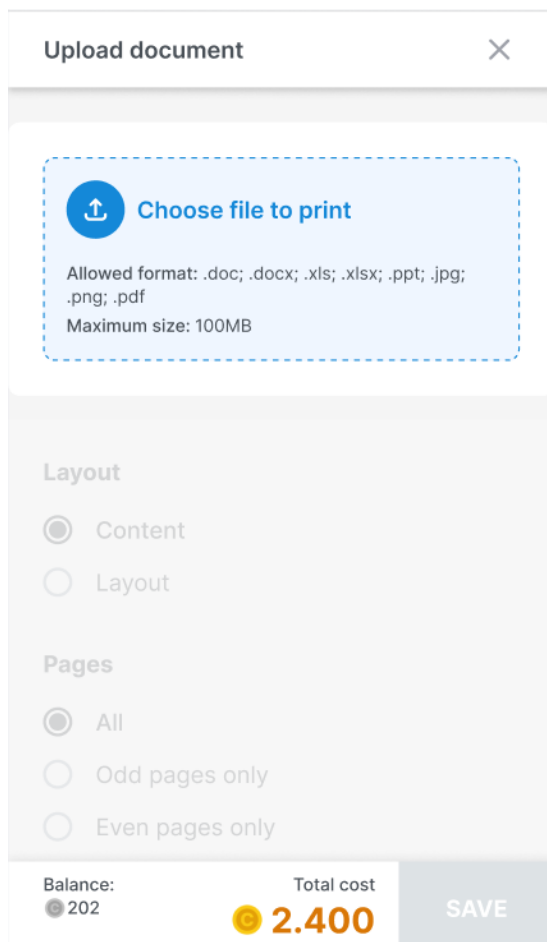
Hình 19: Giao diện trang tải file

Các thành phần chính trong trang tải file lên hệ thống:

1. Thông tin về file upload: biểu diễn những thông tin ban đầu của file upload

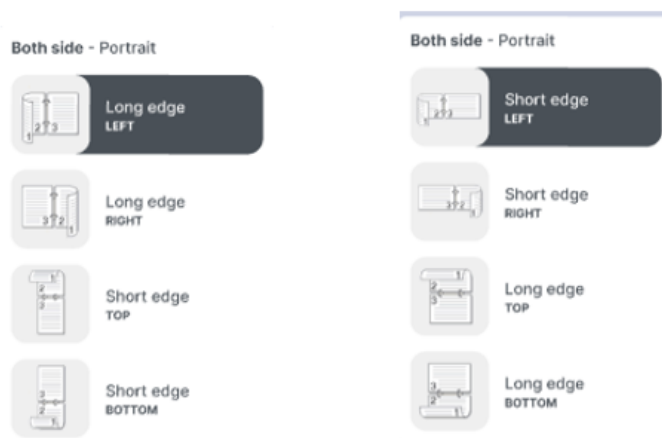
- Trạng thái upload: thể hiện thông tin tiến trình - % thông tin của file đã được tải lên hệ thống.
- Tên và kích thước file upload.
- Số tiền dự kiến phải thanh toán của file upload.

- Biểu tượng xóa file upload: Khi người dùng nhấn vào biểu tượng này trang tải file của người dùng sẽ được chuyển đổi như sau:



Hình 20: Giao diện trang tải file khi người dùng xóa file mới được upload

2. **Form điều chỉnh các thông số in ấn:** Người dùng điều chỉnh các thông số trước khi lưu trữ file lên đơn hàng bao gồm layout, những trang mà người dùng muốn in của file được upload lên hệ thống

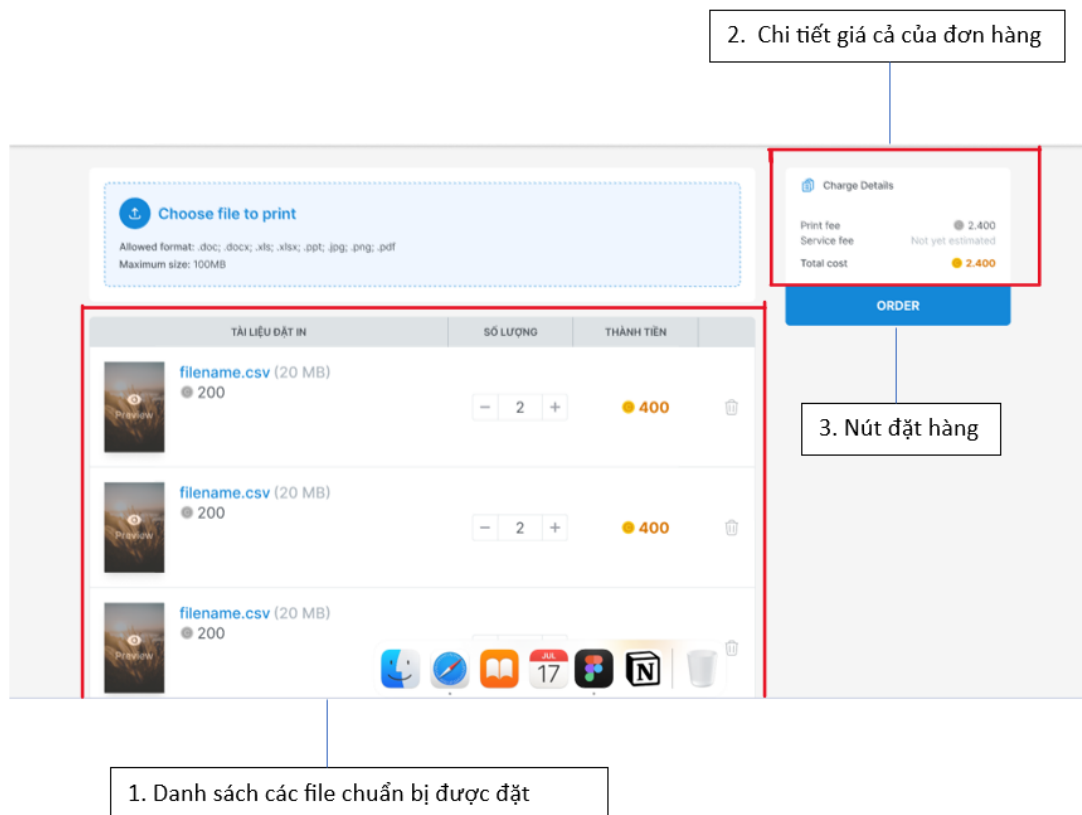


Hình 21: Các thông số người dùng điều chỉnh

3. **Nút lưu trữ:** Người dùng muốn lưu trữ file upload vào danh sách order sau khi điều chỉnh và xác nhận các thông số như layout, pages của file đó. Sau đó, người dùng sẽ được điều hướng sang trang danh sách trong hàng đợi. (chi tiết ở mục 2.4.4).

2.4.4 Trang danh sách đang trong hàng đợi

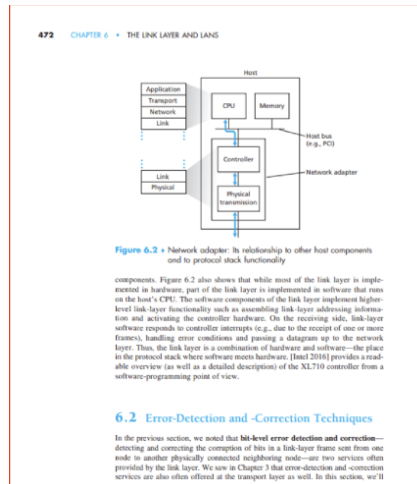
- Để có thể truy cập vào trang này, người dùng đã upload file lên và lưu trữ vào danh sách.



Hình 22: Giao diện trang danh sách đang trong hàng đợi

Các thành phần chính trong trang danh sách đang trong hàng đợi:

- 1. Danh sách các file chuẩn bị được đặt:** Những file đã được người dùng upload và được đưa vào danh sách trước khi order. Ở các file trong danh sách sẽ hiển thị các thông tin sau để người dùng có thể thay đổi và xác nhận
 - Tên và kích thước file.
 - Số lượng file upload (có thể được điều chỉnh trực tiếp trên trang này).
 - Tổng số tiền dự tính của mỗi file.
 - Chế độ preview của thông tin file upload

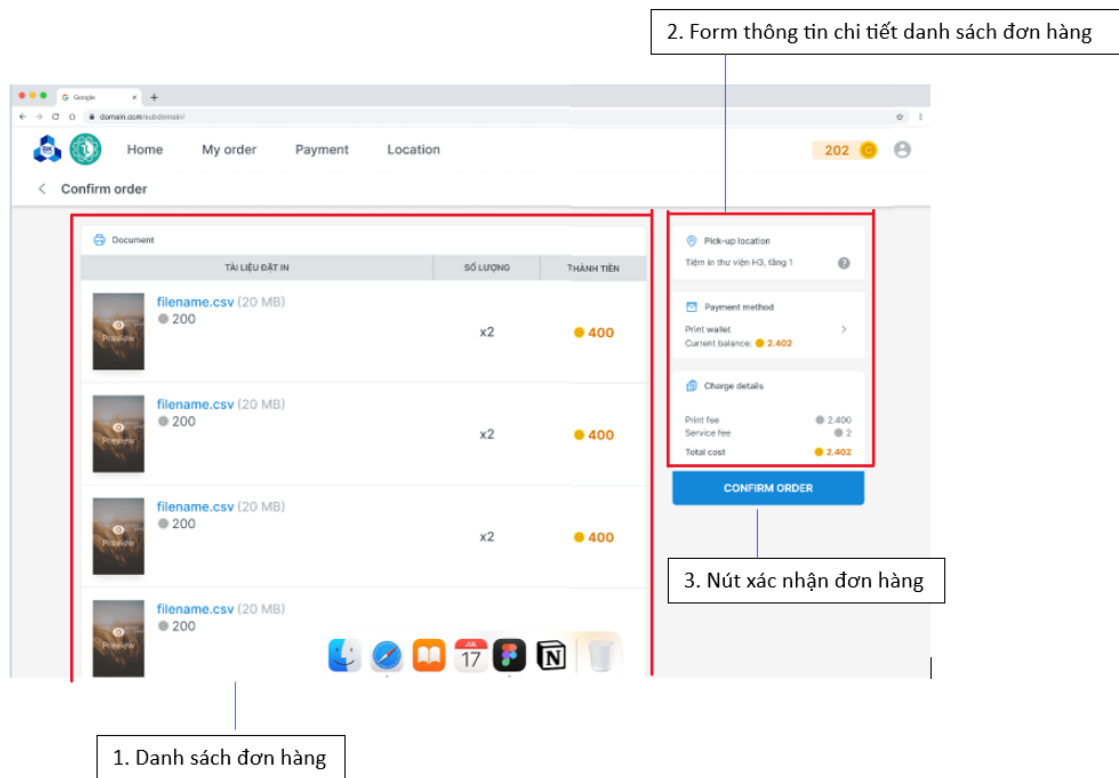


Hình 23: Chế độ preview file

- Chi tiết giá cả của danh sách đơn hàng:** Thông tin về tổng giá tạm thời của danh sách đơn hàng bao gồm giá cứng và giá sử dụng dịch vụ.
- Nút đặt hàng:** Người dùng xác nhận danh sách đơn hàng mình đã upload và chuẩn bị được điều hướng sang trang xác nhận đơn hàng (chi tiết ở mục 2.4.5).

2.4.5 Trang xác nhận danh sách đơn hàng

Để có thể truy cập được trang này, người dùng đã nhấn nút **Order** ở trang danh sách trong hàng đợi sau khi điều chỉnh về số lượng.

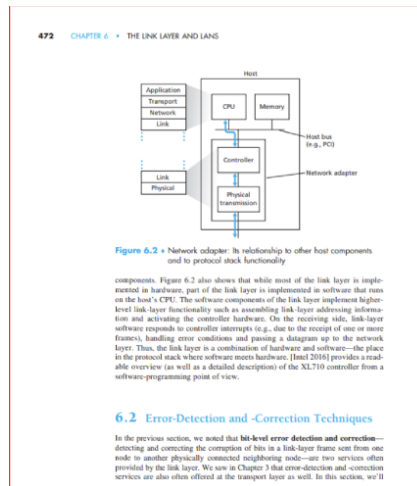


Hình 24: Giao diện trang xác nhận danh sách đơn hàng

Các thành phần chính trong trang xác nhận danh sách đơn hàng:

1. Danh sách các đơn hàng: Những file đã được người dùng upload và được order. Ở các file trong danh sách sẽ hiển thị các thông tin sau để người dùng có thể thay đổi và xác nhận

- Tên và kích thước file.
- Số lượng file upload (có thể được điều chỉnh trực tiếp trên trang này).
- Tổng số tiền dự tính của mỗi file.
- Chế độ preview của thông tin file upload

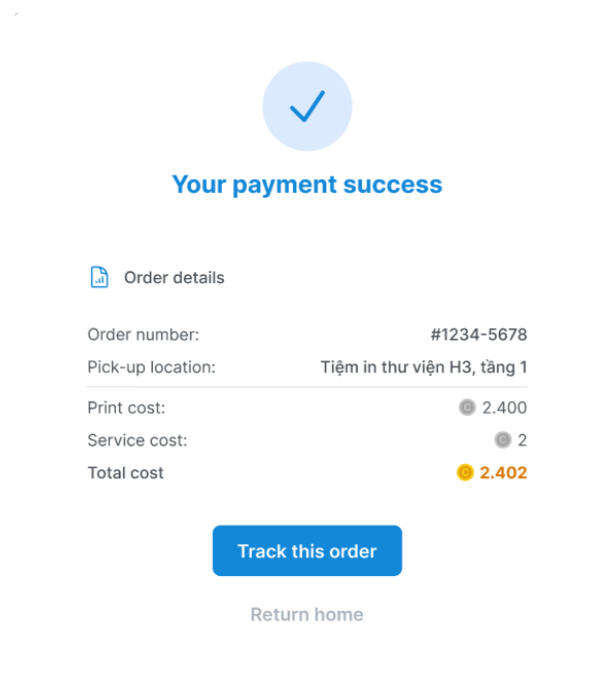


Hình 25: Chế độ preview file

2. Chi tiết giá cả của danh sách đơn hàng:

- Thông tin vị trí người dùng sẽ tiến hành lấy đơn hàng tại khuôn viên của trường.
- Xác nhận kiểu thanh toán cho danh sách đơn hàng và hiển thị tình trạng có thanh toán được hay không (chi tiết ở mục **2.4.6**).
- Chi tiết giá cả của danh sách đơn hàng: Tổng giá bao gồm giá cứng và giá sử dụng dịch vụ.

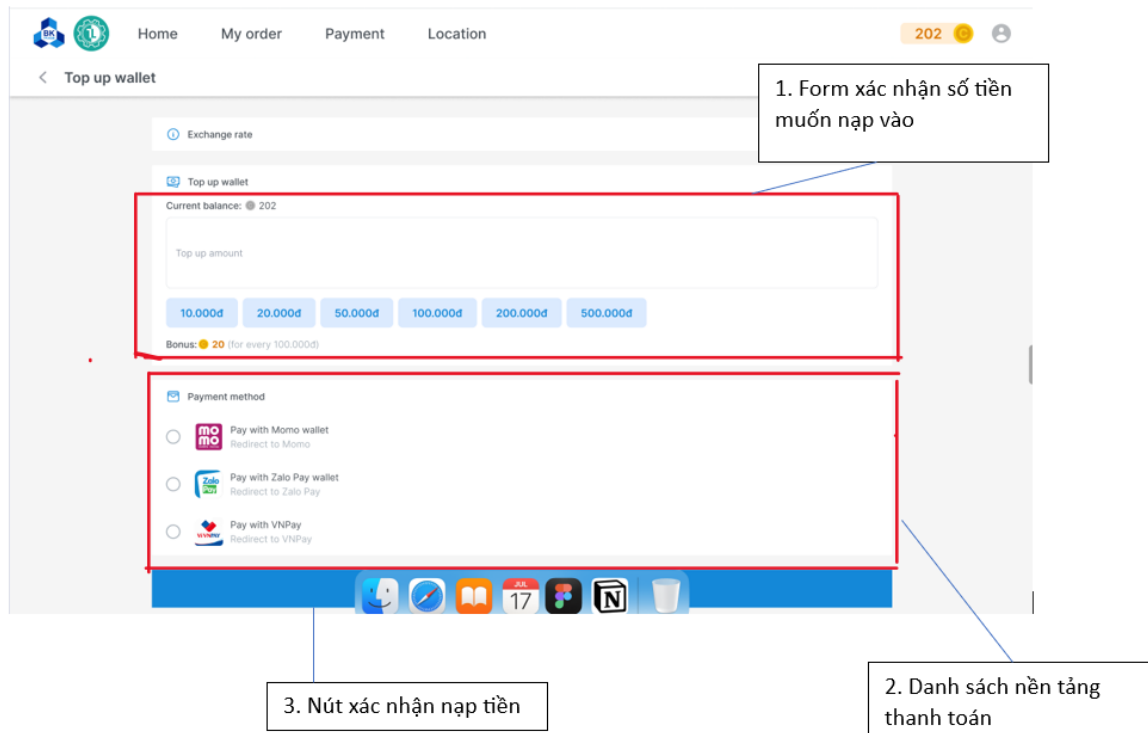
3. Nút xác nhận đặt hàng: Người dùng xác nhận thanh toán danh sách đơn hàng và được xác nhận thành công.



Hình 26: Chế độ preview file

2.4.6 Trang thanh toán

- Để truy cập trang này, người dùng đã nhấn vào vùng xác nhận kiểu thanh toán ở trang xác nhận danh sách đơn hàng khi tài khoản của họ không đủ để xác nhận danh sách đơn hàng đó. Vì hệ thống sẽ sử dụng đơn vị coin để thanh toán, chức năng chính của trang thanh toán này sẽ giúp người dùng nạp coin vào tài khoản thông qua tiền mặt dựa vào các phương thức thanh toán khác nhau.



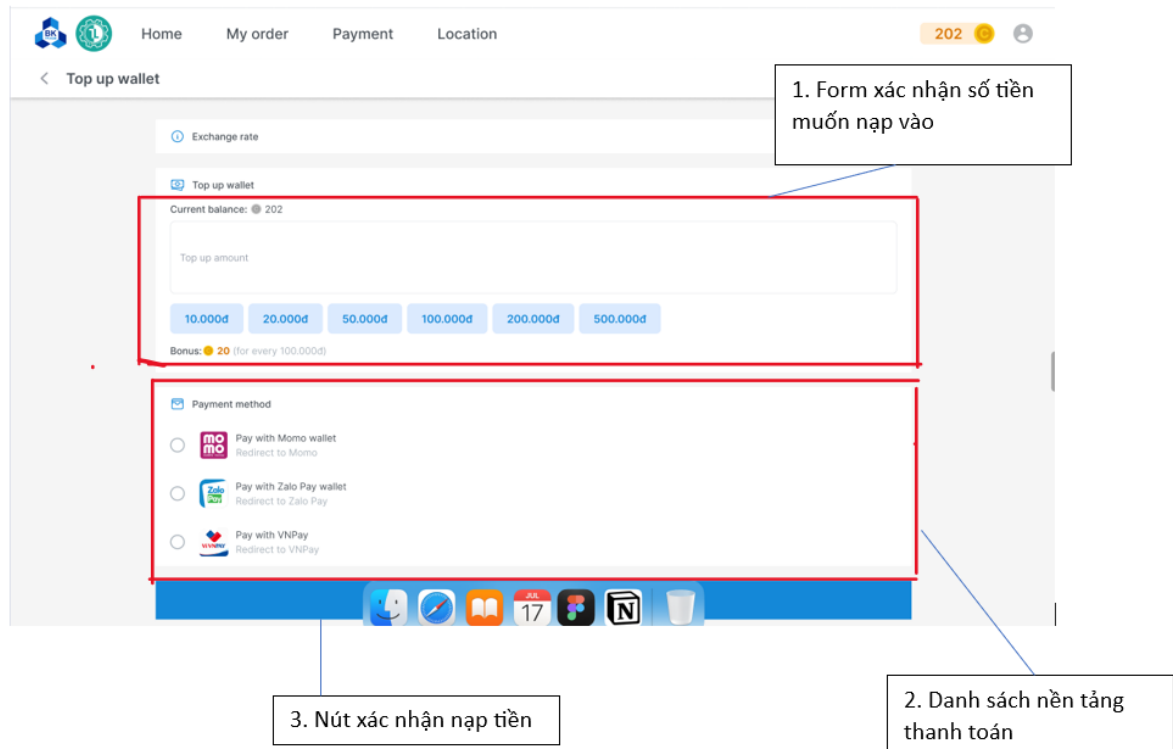
Hình 27: Giao diện trang thanh toán

Các thành phần chính của trang thanh toán:

- Form xác nhận số tiền muốn nạp:** Lựa chọn mức tiền muốn nạp vào và sau đó được quy đổi ra thành coin của hệ thống.
- Danh sách nền tảng thanh toán:** Người dùng lựa chọn bên thanh toán thứ ba để có thể nạp tiền và quy đổi sang coin trong hệ thống.
- Nút xác nhận nạp tiền:** Người dùng xác nhận nạp tiền và được điều hướng lại trang xác nhận danh sách đơn hàng để thanh toán.

2.4.7 Trang danh sách đơn hàng có trên hệ thống

Để có thể vào trang này, người dùng đã có thể nhấn vào nút **See more** ở trang chủ hoặc **My order** trên thanh header của hệ thống.



Hình 28: Giao diện trang đơn hàng có trên hệ thống

Các thành phần chính của trang danh sách đơn hàng có trên hệ thống:

1. Lọc danh sách đơn hàng theo trạng thái: Lọc các đơn hàng theo tiêu chí là trạng thái bao gồm:

- All: Tất cả
- Progressing: Đang trong tiến trình in
- Ready: Chuẩn bị được in
- Done: Đã được in xong
- Cancel: Đã hủy

2. Danh sách nền tảng thanh toán: Lọc các đơn hàng theo ngày: người dùng sẽ tiến hành chọn ngày bắt đầu và kết thúc và lọc ra những đơn hàng đã được upload trên hệ thống nằm trong khoảng thời gian đó.

Filter

Date range

Start date

End date

dd/mm/yyyy

dd/mm/yyyy

←

June 2021

→

Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	30	26	27	28
29	30	1	2	3	4	5

Reset

Apply

Hình 29: Lọc theo đơn hàng theo ngày

3. **Danh sách đơn hàng trên hệ thống:** Hiển thị danh sách đơn hàng trên hệ thống sau khi được lọc bởi 2 bộ lọc trạng thái đơn hàng và ngày upload.

3 Architecture design

3.1 Kiến trúc phân lớp (Layered architecture)

Hệ thống HCMUT-SSPS sẽ sử dụng kiến trúc phân lớp (Layered architecture)

- Ưu điểm: giảm bớt phụ thuộc vì các lớp sẽ độc lập với các lớp khác, giúp việc mở rộng hay debug hệ thống đơn giản và dễ dàng hơn.
- Nhược điểm: việc tách biệt hoàn toàn các lớp trong thực tế tương đối khó hiện thực. Các luồng dữ liệu đi qua nhiều lớp khác nhau có thể làm giảm tốc độ xử lý của hệ thống.

3.1.1 Kế hoạch trình bày giao diện (Interface strategy)

Giao diện người dùng sẽ được chia thành nhiều thành phần, mỗi thành phần đảm nhiệm các chức năng cụ thể:

- User Interaction: bao gồm Get Order List, Upload Files, Configuring Orders, Executing Orders, Log In.
- Manager Interaction: Managing Printers, Get Logs, Log In.

Giao diện người dùng được hiện thực trên công nghệ React với Typescript bởi tính gọn nhẹ thân thiện với người dùng và tính dễ tiếp cận, type-safe dành cho developer.

3.1.2 Data storage approach (Phương pháp lưu trữ dữ liệu)

Dữ liệu sẽ được lưu trữ qua hai server:

- PostgreSQL server: Lưu trữ dữ liệu về đơn đặt in (Order), máy in (Printer), lịch sử (history log).
- MinIO server: Lưu trữ các file được người dùng tải lên.

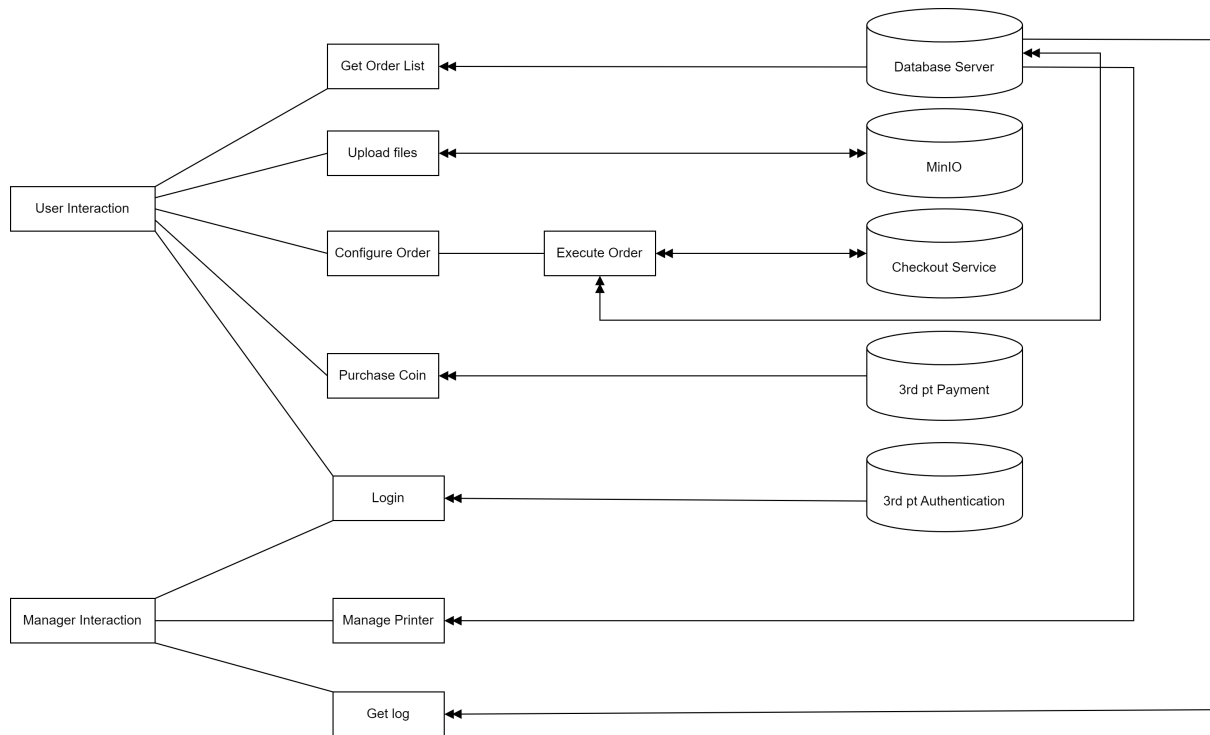
3.1.3 Quản lý API (API management)

Các API được quản lý bằng công cụ quản lý Swagger. Các API của hệ thống được chia ra làm 2 loại:

- 3rd Party: người dùng gọi API khi muốn sử dụng các dịch vụ thanh toán/ nạp tiền (Momo, Paypal,...) và dịch vụ lưu trữ file (MinIO).
- PostgreSQL server: sinh viên gọi API để theo dõi đơn hàng của mình và location, người quản lý chỉnh sửa thông tin, lưu lại lịch sử đơn hàng,...

3.1.4 Các biểu đồ phân lớp

Box-line diagram

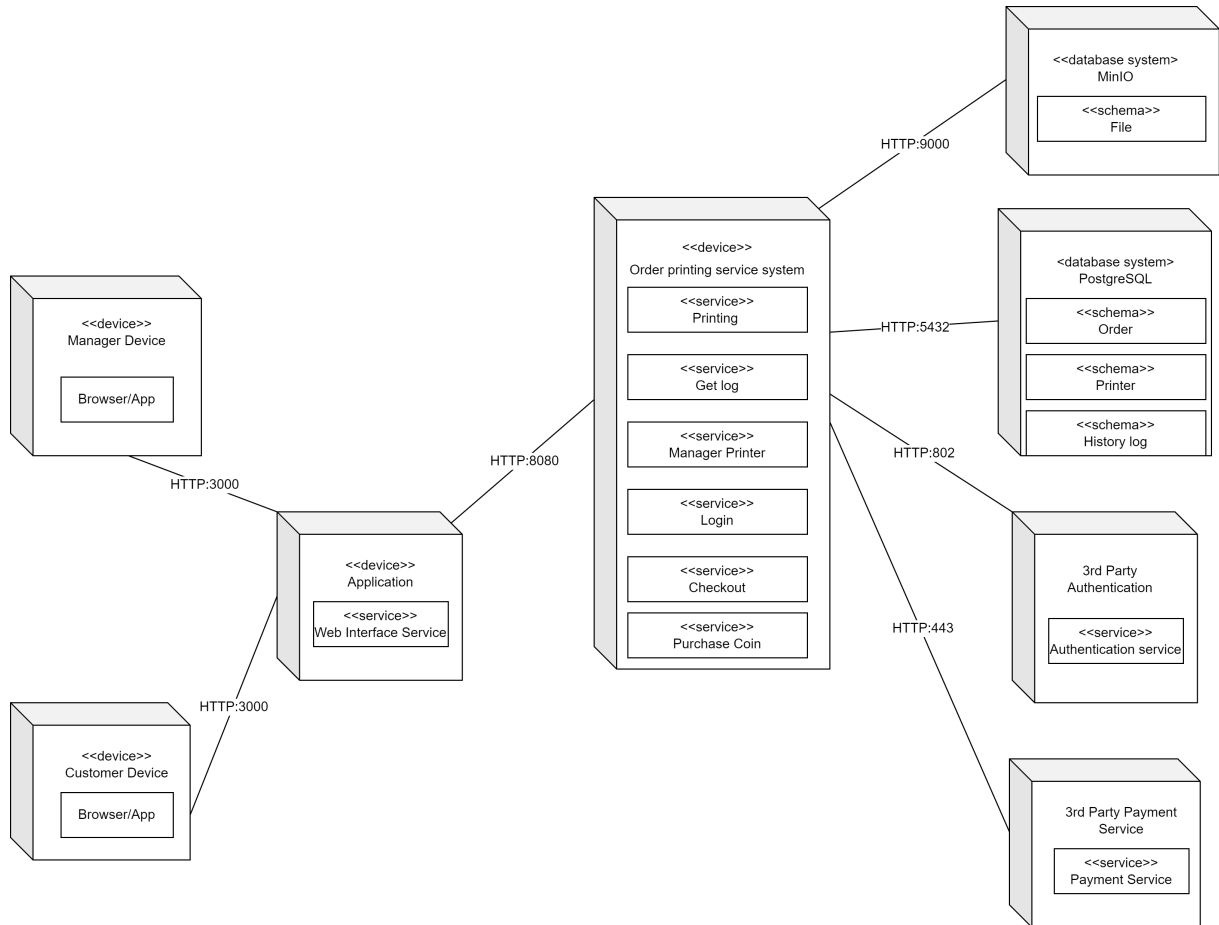


Hình 30: Box-line diagram

Mô tả box-line diagram:

- Được xây dựng theo mô hình microservice - biểu hiện interaction của khách hàng và người quản lý với từng dịch vụ, tác động đến kho lưu trữ dữ liệu.
- Tương tác của khách hàng:
 - Lấy ra danh sách đơn hàng bằng cách truy cập vào hệ cơ sở dữ liệu.
 - Đăng nhập dựa vào xác thực ở bên thứ ba.
 - Tải file từ máy lên minIO.
 - Điều chỉnh cấu hình của danh sách đơn hàng trên giao diện người dùng và thực hiện thanh toán đơn hàng nhờ vào dịch vụ bên thứ ba. Sau khi thanh toán đơn hàng, thông tin thanh toán sẽ được lưu vào hệ cơ sở dữ liệu.
 - Mua coin (phương tiện thanh toán trên hệ thống) bằng dịch vụ bên thứ 3 (Momo, paypal,...).
- Tương tác của quản lý:
 - Quản lý máy in - cập nhật, xóa, thêm và lưu trữ thông tin máy in sau khi điều chỉnh lên hệ cơ sở dữ liệu.
 - Lưu lại lịch sử đơn hàng trên hệ cơ sở dữ liệu.

Deployment diagram:

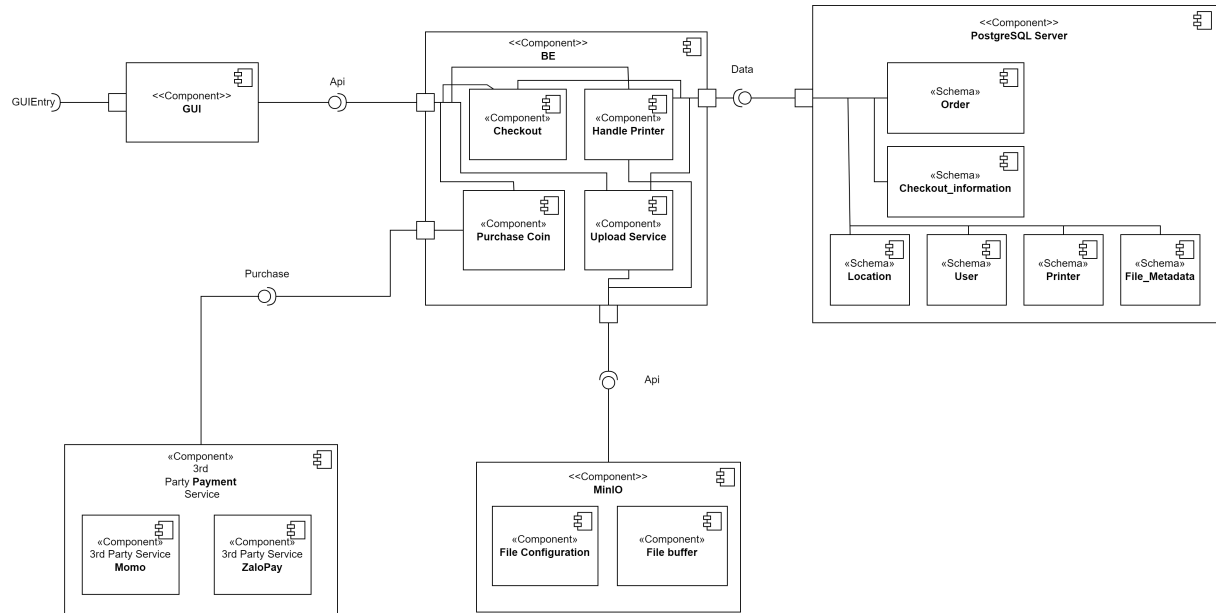


Hình 31: Deployment Diagram

Mô tả deployment diagram:

- Được xây dựng để mô tả quá trình ứng dụng hệ thống và các dịch vụ được deploy lên Internet và tiếp cận đến khách hàng thông qua các giao thức đã được định nghĩa.
- Khách hàng và quản lý truy cập giao diện ứng dụng thông qua giao thức HTTP với port 3000.
- Sau khi truy cập vào giao diện ứng dụng, người dùng sử dụng cách dịch vụ của hệ thống như in ấn, theo dõi lịch sử, đăng nhập, quản lý máy in,... bằng giao thức HTTP với port 8080.
- Các dịch vụ đó sẽ gọi API đến các nơi lưu trữ dữ liệu thông qua giao thức HTTP với các port khác nhau được quy ước như sau:
 - MinIO: 9000
 - PostgreSQL: 5432
 - 3rd Authentication: 802
 - 3rd Party Payment Service: 443

3.2 Component diagram



Hình 32: Component diagram

Mô tả Component diagram của module Request Printing Order.

- Component diagram bao gồm các component: GUI, BE, PostgreSQL, MinIO, 3rd Party Payment Service.
- GUI: Đây là giao diện mà người dùng tương tác với hệ thống. Component này giao tiếp qua API với các component con của BE để người dùng thực hiện tải tài liệu (Upload Service), mua coin (Purchase Coin), xác nhận đơn in (Checkout) và thực hiện in (Handle Printer)
- BE: Bao gồm các component Upload Service, Checkout, Purchase Coin, Handle Printer. Các component trên sẽ nhận lệnh qua API từ GUI, cụ thể: -Component Upload Service yêu cầu dịch vụ tải file lên qua API từ component MinIO và truyền thông tin của file vào PostgreSQL.
 - Component Purchase yêu cầu dịch vụ qua API từ component 3rd Party Payment Service.
 - Component Checkout truyền dữ liệu sang component PostgreSQL.
 - Component Handle Printer yêu cầu thông tin in tài liệu qua API từ component PostgreSQL và yêu cầu file tài liệu qua API từ component MinIO để thực hiện in tài liệu.
- MinIO: lưu trữ file tài liệu tải lên từ Upload Service và truyền tài liệu cho Handle Printer
- PostgreSQL: lưu trữ dữ liệu thông tin tài liệu từ component Upload Service và Checkout, truyền dữ liệu tài liệu cho Handle Printer.
- 3rd Party Payment Service: cung cấp API cho Purchase Coin.

TÀI LIỆU THAM KHẢO

1. Projects Inventory. (1/6/2022). *Functional and Non-functional Requirements of Printing Press Management System System*. Truy cập từ <https://projectsinventory.com/functional-and-non-functional-requirements-of-printing-press-management-system-system/>
2. ResearchGate. *The system and its context*. Truy cập từ https://www.researchgate.net/figure/The-system-and-its-context-The-context-is-split-into-problem-domain-and-application_fig1_228417190