

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



## **CÔNG NGHỆ PHẦN MỀM**

---

### **BÁO CÁO BÀI TẬP LỚN**

#### **TASK 3: ARCHITECTURE DESIGN**

---

Sinh viên thực hiện:	Võ Trường Chinh	2320002
	Lê Thị Kỳ Duyên	2113050
	Thòng Thượng Hà	2113277
	Nguyễn Tuấn Khanh	2110251
	Thái Bảo Long	2113946

Tp. Hồ Chí Minh, Tháng 11/2023

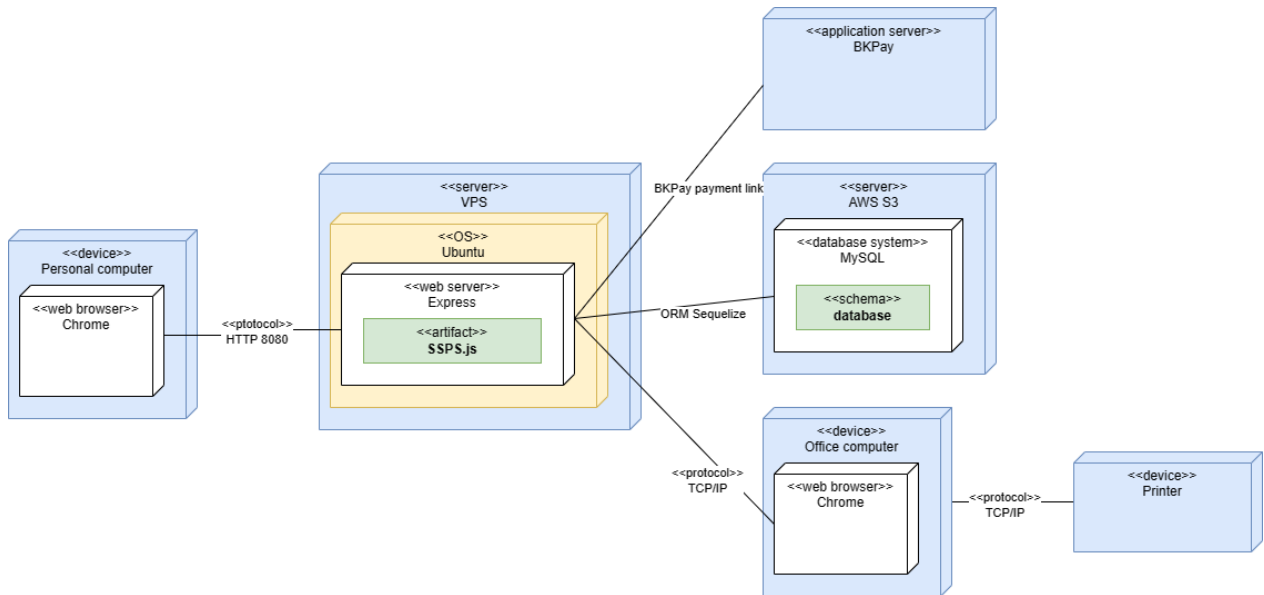
## Mục lục

<b>1</b>	<b>Thiết kế kiến trúc</b>	<b>2</b>
1.1	Lựa chọn kiến trúc . . . . .	2
1.1.1	Deployment diagram cho thiết kế tổng thể hệ thống HCMUT-SSPS .	2
1.1.2	Lựa chọn mẫu thiết kế cho hệ thống . . . . .	3
1.1.3	Box-line diagram cho thiết kế tổng thể hệ thống HCMUT-SSPS . . .	4
1.1.4	Mô tả chi tiết về Presentation strategy . . . . .	5
1.1.5	Mô tả chi tiết về Data storage . . . . .	11
1.1.6	Mô tả chi tiết về external services/APIs . . . . .	11
1.2	Task 3.2: Component diagram . . . . .	12
1.2.1	Component diagram cho module Print . . . . .	12
1.2.2	Mô tả component diagram cho module Print . . . . .	12

# 1 Thiết kế kiến trúc

## 1.1 Lựa chọn kiến trúc

### 1.1.1 Deployment diagram cho thiết kế tổng thể hệ thống HCMUT-SSPS



Hình 1: Deployment diagram cho toàn hệ thống

#### Phía người dùng:

- Người dùng (sinh viên/SPSO) sử dụng máy tính cá nhân, thông qua trình duyệt web bất kỳ (như Chrome/Microsoft Edge/FireFox...) để truy cập vào trang web hệ thống in ấn thông minh SSPS.

#### Phía hệ thống:

- Hệ thống được xây dựng trên một máy chủ riêng ảo (Virtual private server - VPS) có:
  - RAM cho toàn hệ thống: 8GB
  - RAM cho web server: 4GB
  - RAM cho database: 3GB
- Trên máy chủ VPS cài đặt hệ điều hành Ubuntu 22.04.3
- Một web server sẽ được xây dựng trên máy chủ. Tùy thuộc vào thông tin đăng nhập của người dùng, hệ thống sẽ hiển thị giao diện và các chức năng tương ứng.
- Phần Frontend được xây dựng dựa trên công nghệ HTML, CSS và JavaScript với thư viện Reactjs.
- Phần Backend sử dụng công nghệ Nodejs

- Web server truyền và nhận thông tin với trình duyệt của người dùng thông qua giao thức mạng HTTP

#### **Phía cơ sở dữ liệu:**

- Sử dụng hệ thống quản trị cơ sở dữ liệu mã nguồn mở MySQL tích hợp với AWS S3 để lưu trữ các tài liệu đã tải lên
- Cơ sở dữ liệu giao tiếp với web server thông qua ORM Sequelize

#### **Phía máy in:**

- Tại phòng có chứa máy in sẽ có một máy tính của văn phòng, máy tính tương tác với hệ thống thông qua trình duyệt
- Trên máy tính này, người dùng sẽ nhập mã code để tiến hành in tài liệu
- Trình duyệt giao tiếp với hệ thống thông qua giao thức TCP/IP

### **1.1.2 Lựa chọn mẫu thiết kế cho hệ thống**

Nhóm lựa chọn sử dụng mẫu thiết kế MVC (Model-View-Controller) để hiện thực hệ thống in ấn thông minh SSPS. Trong mẫu thiết kế này, các class liên quan đến xử lý dữ liệu được chia tách ra với phần giao diện người dùng. Hệ thống được cấu trúc thành ba thành phần:

- View: xác định và quản lý cách hiển thị dữ liệu cho người dùng. Hiển thị dữ liệu được nhận về từ Controller như là đầu ra. Đây cũng là nơi thay đổi dữ liệu của Model do người dùng nhập vào
- Controller: quản lý tương tác của người dùng (ví dụ: nhấn phím, nhấp chuột, v.v.) và chuyển các tương tác này tới View và Model
- Model: bao gồm một tập hợp các class mô tả nghiệp vụ của ứng dụng, các đối tượng làm việc trong ứng dụng và các class truy cập dữ liệu. Quản lý dữ liệu hệ thống và các hoạt động liên quan trên dữ liệu đó

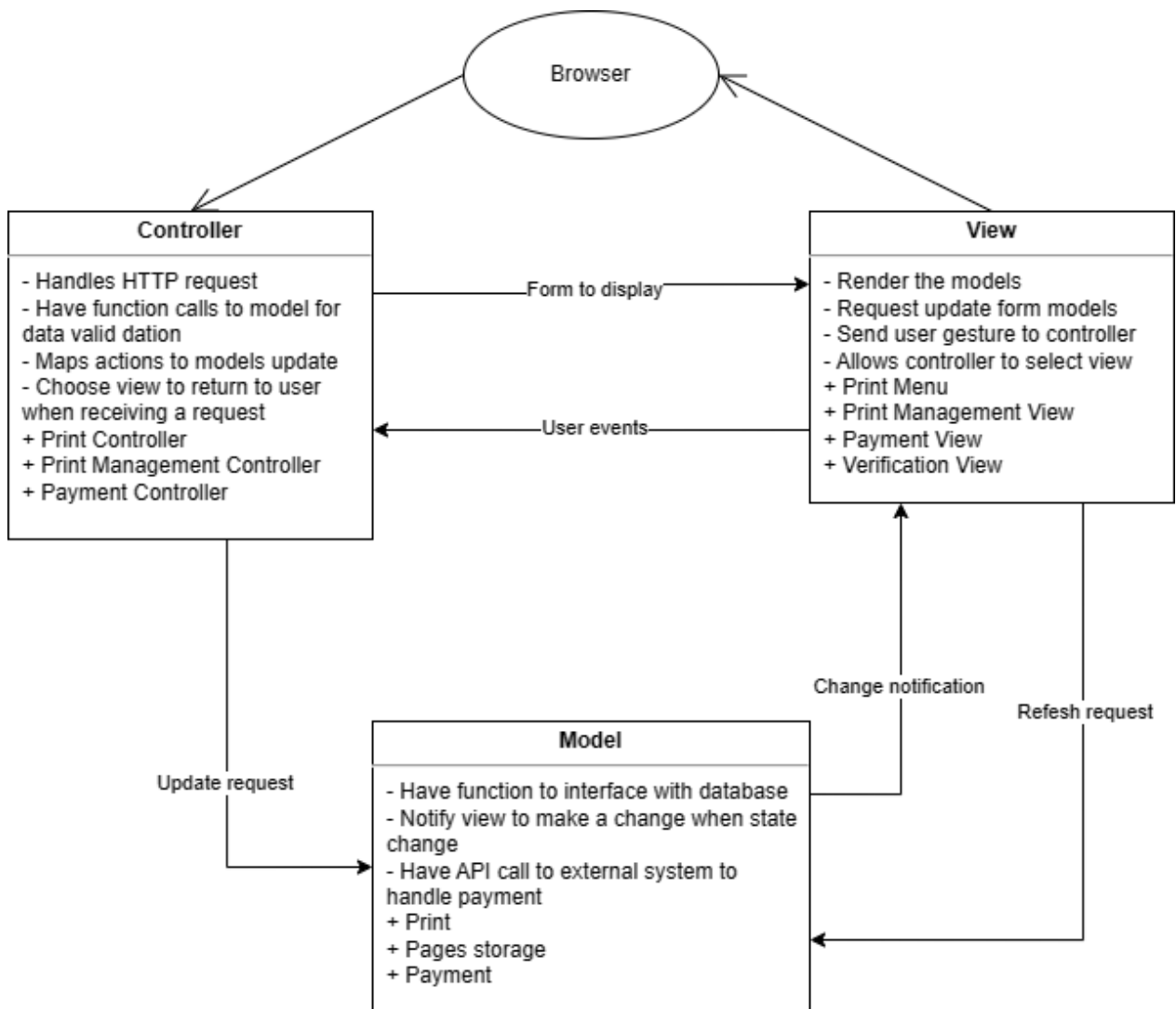
Lý do lựa chọn mẫu thiết kế MVC:

- Dễ dàng thay đổi giao diện người dùng mà không ảnh hưởng đến thiết kế logic bên dưới và luồng dữ liệu
- Hỗ trợ trình bày cùng một dữ liệu theo nhiều cách khác nhau, tạo điều kiện phát triển hệ thống trên di động (chỉ cần thiết kế thêm giao diện cho trình duyệt trên di động mà không cần tạo thêm 1 webserver riêng)
- Có thể tái sử dụng mã cho các ứng dụng sau này

Những nhược điểm của mẫu thiết kế MVC:

- Cần nhiều thời gian để tìm hiểu nếu muốn áp dụng đúng đắn
- Xây dựng hệ thống theo mô hình MVC tốn khá nhiều thời gian và công sức, bởi vậy không cần thiết áp dụng mô hình này cho các dự án nhỏ
- Thời gian trung chuyển dữ liệu cũng cần được cân nhắc kỹ khi triển khai dự án nhỏ

### 1.1.3 Box-line diagram cho thiết kế tổng thể hệ thống HCMUT-SSPS



Hình 2: Box-line diagram cho toàn hệ thống

Component **View** chứa các component phục vụ cho việc render các thuộc tính trong giao diện người dùng. Bao gồm:

- Component **Print Menu**: Render giao diện cho các trang thuộc module **Print** bao gồm: trang tải tài liệu, chọn thuộc tính, chọn máy in. Gửi các lựa chọn của người dùng cho component **Print Controller**
- Component **Print Management View**: Gửi tín hiệu kiểm tra tính hợp lệ của tệp được tải lên cho component **Print Management Controller**. Nhận request để cập nhật trạng thái tệp
- Component **Payment View**: Render giao diện thanh toán (trong trường hợp hệ thống kiểm tra thấy số trang còn lại trong tài khoản nhỏ hơn số trang cần in). Gửi request

payment cho component **Payment Controller**

- Component **Verification View**: Render giao diện xác nhận và hiện mã Code cho người dùng

Component **Controller** chứa các component phục vụ việc trả về **View** các phản hồi và thông báo tới người dùng. Bao gồm:

- Component **Print Controller**: Nhận các lựa chọn của người dùng, xác nhận tùy chọn và chuyển cho component **Print**
- Component **Print Management Controller**: Nhận yêu cầu kiểm tra tính hợp lệ của tệp, xử lý và gửi phản hồi cho component **Print Management View**
- Component **Payment Controller**: Nhận request thanh toán, gửi yêu cầu giao dịch đến component **Payment** để xử lý. Gửi thông báo kết quả tới component **Payment View**

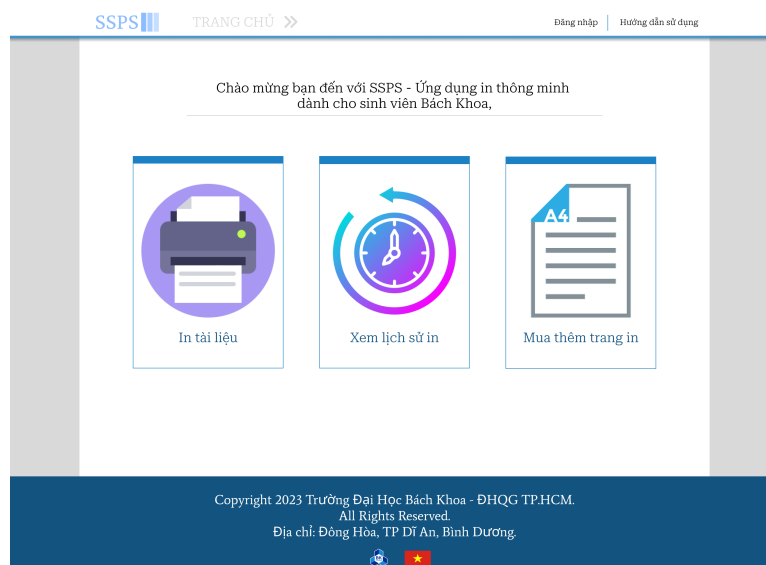
Component **Model** chứa các component thao tác với database, chứa tất cả attribute và method truy vấn trực tiếp với dữ liệu. Bao gồm:

- Component **Print**: chứa các attribute và method cần thiết cho việc tải tài liệu, chọn thông số, chọn máy in
- Component **Pages storage**: chứa các attribute và method để quản lý, kiểm tra số trang giấy còn lại của sinh viên
- Component **Payment**: chứa các attribute và method để xử lý yêu cầu thanh toán, giao tiếp với hệ thống thanh toán bên ngoài

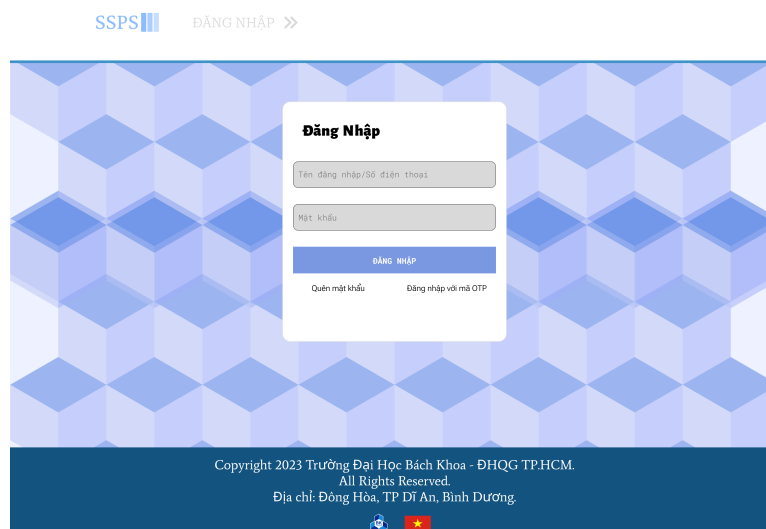
#### 1.1.4 Mô tả chi tiết về Presentation strategy

##### Describe User Interface

Trang web SPSS (Student Smart Printing Service - Dịch vụ in ấn thông minh) là một nơi cung cấp dịch vụ in ấn chất lượng cao và độc đáo, đặc biệt chuyên nghiệp trong việc in các tài liệu từ phần mềm SPSS. Với đội ngũ nhân viên kỹ thuật giàu kinh nghiệm, chúng tôi cam kết mang đến cho bạn những sản phẩm in SPSS chất lượng, sắc nét và đẹp mắt.



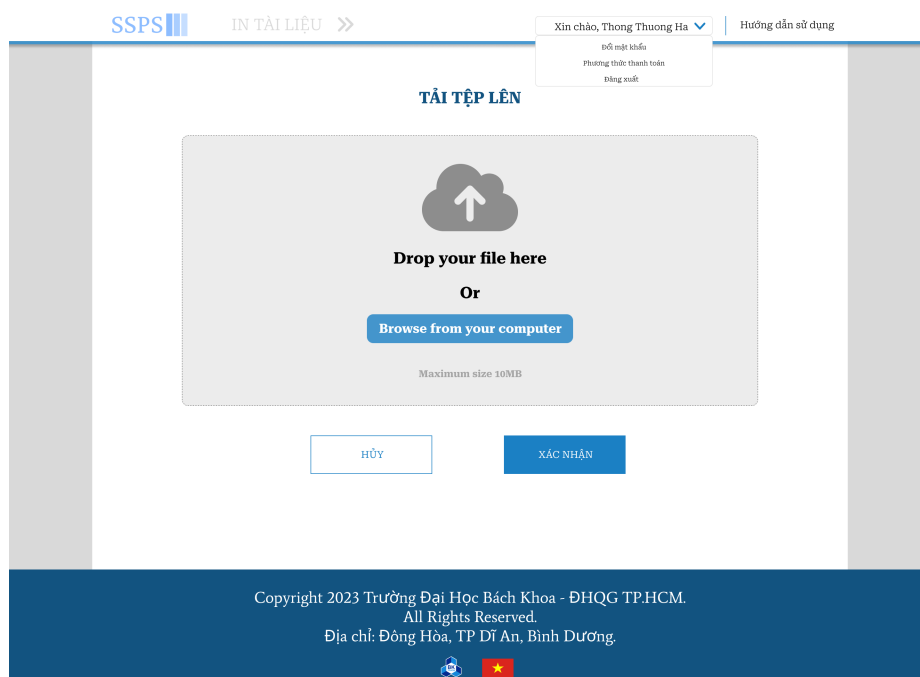
Đầu tiên, khi truy cập trang web, ta sẽ thấy "Trang chủ" của trang web. Trang chính sẽ bao gồm 3 dịch vụ chính là In tài liệu, Xem lịch sử in, Mua thêm trang in. Lúc này, nếu người dùng chọn vào một trong các dịch vụ này, hệ thống sẽ chuyển hướng người dùng đến trang đăng nhập. Người dùng cũng có thể đăng nhập bằng nút "Đăng nhập" ở góc phải màn hình, kế bên đó là "Hướng dẫn sử dụng cho người mới". Đến trang đăng nhập, người dùng sẽ nhập các thông tin đăng nhập (tên tài khoản, mật khẩu).



Nếu như người nhập sai tên tài khoản hoặc mật khẩu thì sẽ xuất hiện thông báo sai tài khoản hoặc mật khẩu tương ứng. Ngược lại, nếu nhập đúng thông tin đăng nhập thì trang web sẽ chuyển hướng đến giao diện chính.



Sau khi đăng nhập, ở giao diện chính người dùng lúc này có thể chọn các dịch vụ tương ứng.  
Nếu người dùng chọn "In tài liệu", website sẽ chuyển người dùng đến giao diện in.  
Tại đây, người dùng có thể tải tài liệu của mình lên để tiến hành in.



Người dùng còn có thể cấu hình in cho tài liệu của họ.



SSPS IN TÀI LIỆU Xin chào, Thông Thuong Ha Hướng dẫn sử dụng

**CHỌN KHỔ GIẤY**

Chọn khổ giấy in

A0 A1 A2 A3 A4 A5

Chọn số trang trên một mặt

Chọn số mặt trên một tờ

HỦY XÁC NHẬN

Copyright 2023 Trường Đại Học Bách Khoa - ĐHQG TP.HCM.  
All Rights Reserved.  
Địa chỉ: Đồng Hòa, TP Dĩ An, Bình Dương.

SSPS IN TÀI LIỆU Xin chào, Thông Thuong Ha Hướng dẫn sử dụng

**CHỌN KHỔ GIẤY**

A4

2

1

1

HỦY XÁC NHẬN

Copyright 2023 Trường Đại Học Bách Khoa - ĐHQG TP.HCM.  
All Rights Reserved.  
Địa chỉ: Đồng Hòa, TP Dĩ An, Bình Dương.

Sau khi đã cấu hình xong cho tài liệu, người dùng sẽ chọn những máy in đang khả dụng để tiến hành in.

SSPS IN TÀI LIỆU >> Xin chào, Thông Thuong Ha | Hướng dẫn sử dụng

### XÁC NHẬN

Thông tin máy in

\* H6.P603

Thông tin tài liệu

Khổ giấy in: A4  
Trang in: 2  
Số trang trên một mặt: 1  
Số mặt trên một tờ: 1

HỦY XÁC NHẬN

Copyright 2023 Trường Đại Học Bách Khoa - ĐHQG TP.HCM.  
All Rights Reserved.  
Địa chỉ: Đồng Hòa, TP Dĩ An, Bình Dương.

Nếu người dùng nhấn "Xác nhận", hệ thống sẽ cung cấp cho người dùng một "Mã xác nhận". Mã này người dùng sẽ lấy để có thể in trực tiếp.

SSPS IN TÀI LIỆU >> Xin chào, Thông Thuong Ha | Hướng dẫn sử dụng

### NHẬN YÊU CẦU THÀNH CÔNG !

MÃ XÁC NHẬN :

0523HYN

TRỞ VỀ TRANG CHỦ

Copyright 2023 Trường Đại Học Bách Khoa - ĐHQG TP.HCM.  
All Rights Reserved.  
Địa chỉ: Đồng Hòa, TP Dĩ An, Bình Dương.

Trang web SPSS (Student Smart Printing Service - Dịch vụ in ấn thông in) hướng đến



một giao diện người dùng tối giản, dễ sử dụng và trải nghiệm của người dùng được tối ưu, qua đó mong muốn tạo ra trải nghiệm dịch vụ in ấn tốt nhất tại website để đảm bảo rằng mọi tài liệu của bạn được in ấn một cách chuyên nghiệp và hoàn hảo nhất.

### 1.1.5 Mô tả chi tiết về Data storage

Cơ sở dữ liệu MySQL.

Các bảng dữ liệu chính:

- Bảng users: Lưu đầy đủ thông tin người dùng như id, tên đăng nhập, mật khẩu đã mã hóa, họ tên, email, số điện thoại, địa chỉ, giới tính, ngày sinh, ngày tạo tài khoản, số dư tài khoản, trạng thái tài khoản, vai trò/quyền hạn...
- Bảng user\_documents: Lưu danh sách các tài liệu người dùng đã tải lên với các trường như id tài liệu, id người dùng, tên tài liệu, định dạng tài liệu, kích thước tài liệu, ngày giờ tải lên, đường dẫn url tới file lưu trữ...
- Bảng print\_jobs: Lưu chi tiết tất cả các lịch sử in ấn của người dùng với các trường như id lịch sử in, id người dùng, id tài liệu, id máy in, số bản in, ngày giờ in, trạng thái in, chi phí in...
- Bảng printers: Lưu chi tiết các thông tin của máy in như id máy in, tên máy in, địa điểm đặt, trạng thái hoạt động, số lượng giấy còn lại, các tính năng hỗ trợ...
- Bảng printer\_logs: Lưu lịch sử hoạt động các máy in như thời gian bật/tắt, số lượng giấy đã bổ sung thêm, thời gian bổ sung, tổng số giấy hiện tại...
- Bảng payment\_transactions: Lưu lịch sử các giao dịch nạp tiền vào tài khoản của người dùng như id giao dịch, id người dùng, hình thức thanh toán, số tiền, thời gian giao dịch...
- Và các bảng phụ trợ khác như bảng roles/permissions để phân quyền người dùng, bảng categories/tags cho tài liệu, bảng system logs...

Các hoạt động CRUD sẽ được thực hiện thông qua các model, service, controller được xây dựng trên Nodejs/Express kết nối với MySQL thông qua ORM Sequelize.

### 1.1.6 Mô tả chi tiết về external services/APIs

External services/APIs:

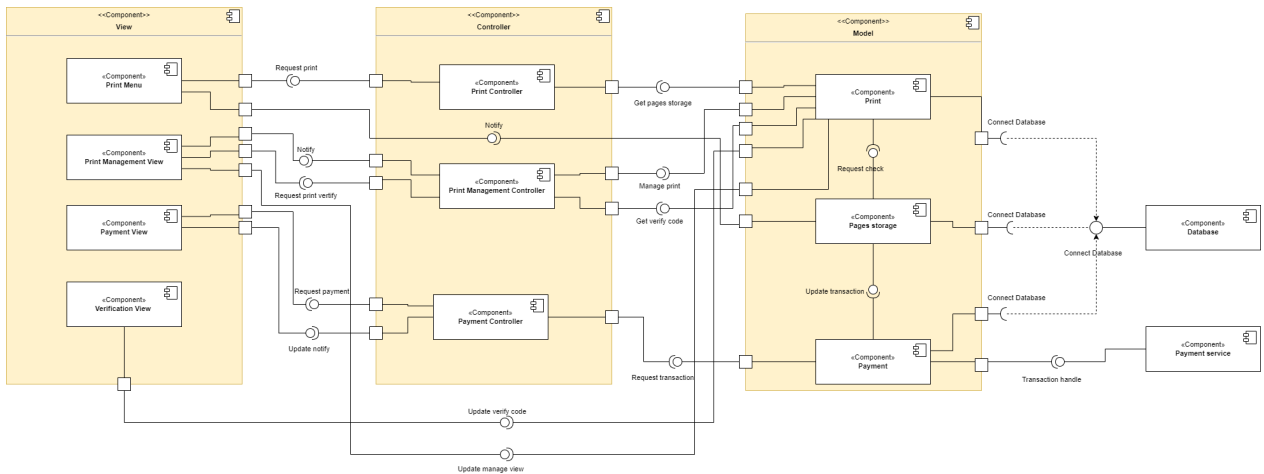
- MoMo API: Tích hợp API của MoMo để cho phép người dùng nạp tiền vào tài khoản thông qua ví MoMo. Các request sẽ được thực hiện từ các service trong Nodejs sử dụng thư viện Request.
- Sendgrid API: Tích hợp API của Sendgrid để gửi các email xác nhận, thông báo tới người dùng. Các request cũng sẽ được thực hiện từ các service trong Nodejs.
- Printer APIs: Tích hợp và gọi đến các API do nhà sản xuất máy in cung cấp để điều khiển máy in, kiểm tra trạng thái, cập nhật số lượng giấy... từ các service trong Nodejs.
- Passport.js: Sử dụng Passport.js để xác thực và phân quyền người dùng trong hệ thống.

- Cloud Storage: Tích hợp với Google Cloud Storage hoặc AWS S3 để lưu trữ các tài liệu đã tải lên.
- Logging: Sử dụng các thư viện như winston, morgan để ghi lại log cho hệ thống.

Việc gọi đến các API sẽ được thực hiện thông qua các service/controller riêng biệt, không gọi trực tiếp từ front-end để đảm bảo tính bảo mật và dễ dàng maintain, nâng cấp.

## 1.2 Task 3.2: Component diagram

### 1.2.1 Component diagram cho module Print



Hình 3: Component diagram cho module Print

### 1.2.2 Mô tả component diagram cho module Print

Component diagram cho module **Print** gồm 3 component chính, mỗi component lớn gồm các component con thực hiện những chức năng khác nhau:

#### 1. Component **View**:

- Các View component đảm nhận các nhiệm vụ như: Render các element của các UI của các trang phục vụ trong quá trình in; tương tác với người dùng trực tiếp thông qua UI và các sự kiện xảy ra trong View; gửi các sự kiện về cho Controller cũng như tiếp nhận request để update lại View mới.
- Các View component sử dụng các interface của Controller và Model để request hoặc cập nhật dữ liệu cũng như update View. Bên cạnh đó View cũng cung cấp các interface để Controller và Model đưa ra thông báo cho người dùng

#### 2. Component **Controller**:

- Các Controller component chịu trách nhiệm trả về View cũng như các phản hồi phù hợp cho từng request. Bên cạnh đó các Controller component còn sử dụng từ Model các hàm để thao tác trên dữ liệu hoặc sử dụng các API của các hệ thống bên ngoài cần thiết cho quá trình in

- Các Controller component sử dụng các interface mà View cung cấp để phản hồi các request và thông báo tới người dùng

### 3. Component **Model**:

- Các Model component cung cấp các interface để Controller thao tác với dữ liệu và View cập nhật dữ liệu
- Các Model component sử dụng các interface từ các component như Payment( để thực hiện thanh toán) thu service và Database( để thực hiện các thao tác với dữ liệu)

Ngoài ra còn các component như **Database** và **Payment service** để hỗ trợ thực hiện các tác vụ về thanh toán cũng như các thao tác với cơ sở dữ liệu