
Software Requirements Specification Smart Printing System

Version 1.4 approved

Prepared by:

- 1. Trần Trung Kiên – 2252397**
- 2. Trần Lương Yên Nhi – 2252586**
- 3. Trần Hữu Phước - 2153717**
- 4. Trần Quang Thiên – 2252768**
- 5. Trần Ngọc Khánh Huy - 2252265**

Department of Software Engineering

Faculty of Computer Science and Engineering

Ho Chi Minh City University of Technology – VNU-HCM

<24/11/2024>

Table of Contents

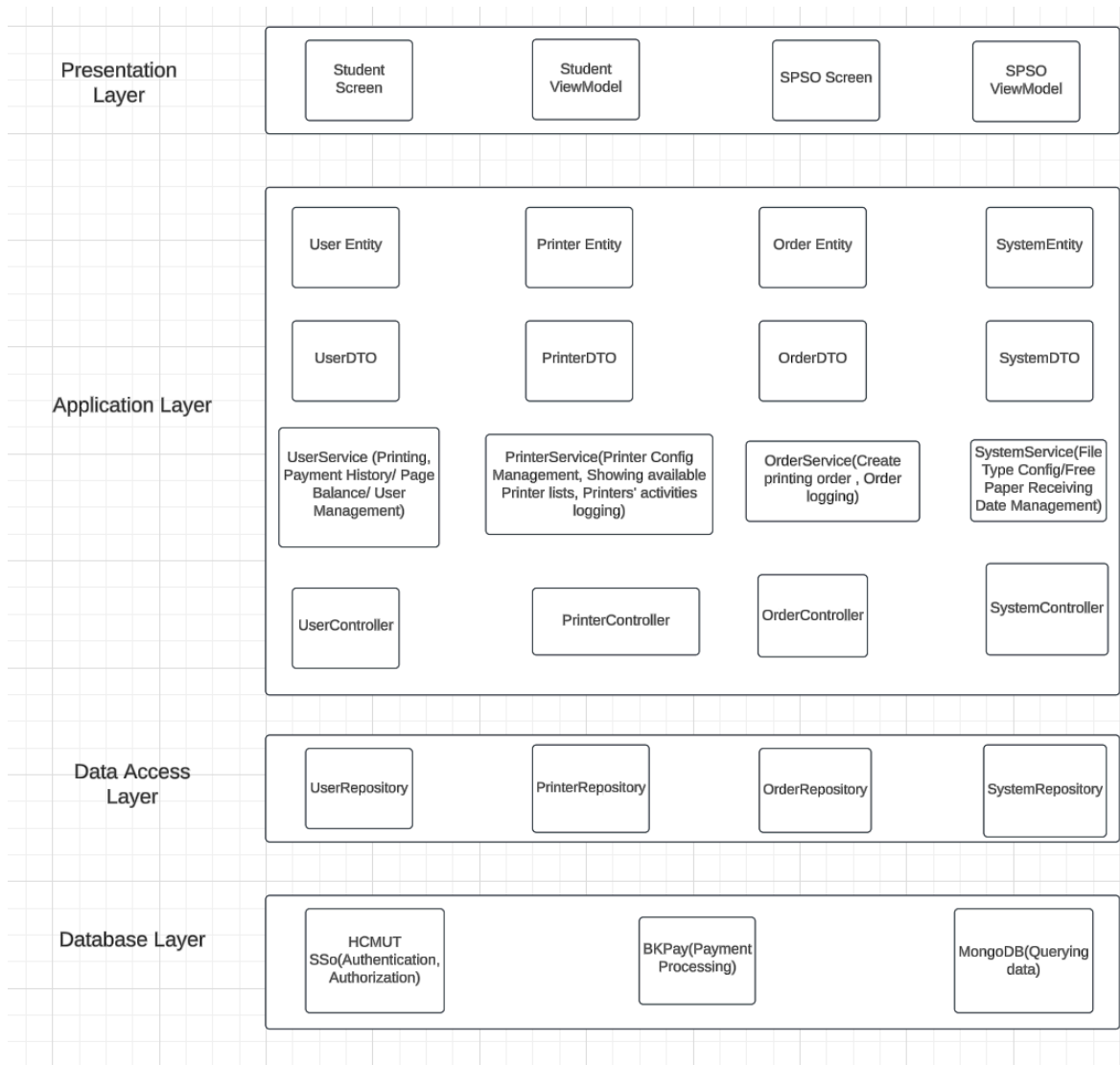
1. Task 3: Architecture Design	4
1.1. Task 3.1: Layered Architecture	4
1.1.1. Presentation Strategy	6
1.1.2. Database Approach	7
1.1.3. API Management	8
1.2. Task 3.2: Component Diagram	10

Revision History

Name	Date	Reason For Changes	Version
Project 1	24/11/2024	Số thứ tự trang xuống cuối trang, chỉnh sửa căn lề, font chữ, size chữ đều nhau, chỉnh lại format	1.0

1. Task 3: Architecture Design

1.1. Task 3.1: Layered Architecture



Giải thích:

Sơ đồ được cung cấp thể hiện kiến trúc phân lớp, chia ứng dụng thành bốn lớp riêng biệt: Giao diện, Ứng dụng, Truy cập Dữ liệu và Cơ sở Dữ liệu.

Lớp Giao diện (Presentation Layer): Lớp này bao gồm các thành phần giao diện người dùng tương tác trực tiếp với người dùng. Các thành phần bao gồm Student Screen, SPSO Screen, và các view model tương ứng của chúng (Student

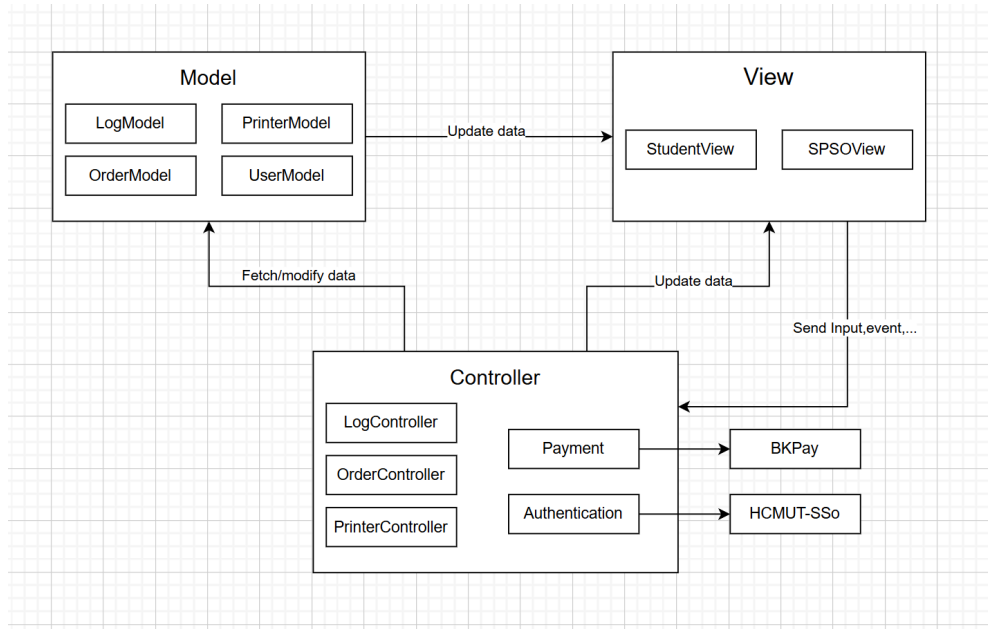
ViewModel, SPSO ViewModel). Các view model này format dữ liệu để hiển thị trên màn hình thay vì hiện toàn bộ thông tin của entity thì chỉ hiện những thông tin cần thiết để hiển thị cho người dùng, giúp tách biệt rõ ràng giữa giao diện người dùng và logic nội bộ của ứng dụng.

Lớp Ứng dụng (Application Layer): Đây là lớp xử lý nghiệp vụ cốt lõi của ứng dụng. Nó chứa các thực thể (User Entity, Printer Entity, Order Entity, Log Entity) đại diện cho các cấu trúc dữ liệu chính, cũng như các Đối tượng Truyền dữ liệu (DTO) như UserDTO, PrinterDTO, OrderDTO, và LogDTO để trao đổi dữ liệu giữa các lớp mà không cho lớp khác tương tác trực tiếp với nghiệp vụ logic. Các dịch vụ như UserService, PrinterService, OrderService, và LogService đóng gói logic nghiệp vụ và xử lý các hoạt động liên quan đến nhiều thực thể. Các bộ điều khiển như UserController, PrinterController, OrderController, và LogController xử lý các yêu cầu đến, xử lý chúng thông qua việc gọi đến Service.

Lớp Truy cập Dữ liệu (Data Access Layer): Lớp này cung cấp giao diện để tương tác với cơ sở dữ liệu, đóng gói logic cho các thao tác CRUD. Các kho lưu trữ (UserRepository, PrinterRepository, OrderRepository, và UserRepository) quản lý truy cập dữ liệu cho mỗi thực thể, đảm bảo tính nhất quán và tính bền vững của dữ liệu mà không tiết lộ chi tiết cơ sở dữ liệu cho Lớp Ứng dụng.

Lớp Cơ sở Dữ liệu (Database Layer): Đây là nơi lưu trữ và quản lý dữ liệu vật lý. Nó bao gồm các hệ thống lưu trữ dữ liệu khác nhau như HCMUT SSO để xác thực đăng nhập một lần, BKPay để xử lý thanh toán, và MongoDB là cơ sở dữ liệu chính để lưu trữ và truy xuất dữ liệu ứng dụng.

MVC Architecture:



View: tùy vào xác thực, ta sẽ chia ra giữa studentView và SPSOView. Đây là phần giao diện người dùng của ứng dụng. View hiển thị dữ liệu từ Model và gửi các yêu cầu của người dùng đến Controller.

Controller: bao gồm LogController, OrderController, PrinterController, cũng như giao tiếp với bên thứ ba (BKPay, SSo). Đây là phần trung gian giữa Model và View. Controller nhận các yêu cầu từ View, xử lý chúng (có thể tương tác với Model) và trả về kết quả cho View.

Model: gồm LogModel, OrderModel, PrinterModel, UserModel. Đây là phần quản lý dữ liệu và logic nghiệp vụ của ứng dụng. Model tương tác với cơ sở dữ liệu và xử lý dữ liệu trước khi gửi đến View hoặc nhận từ Controller.

1.1.1. Presentation Strategy

Việc presentation strategy cho hệ thống in thông minh HCMUT-SSPS sẽ xoay quanh việc làm thế nào để hệ thống này có thể đem lại cho người dùng (trong trường hợp này là sinh viên) một cảm giác thân thuộc, dễ dùng và tiện nghi phù hợp với nhu cầu đa dạng của sinh viên. Hai ngôn ngữ chính sẽ được sử dụng là Tiếng Việt và Tiếng Anh thuận tiện sử dụng cho sinh viên trong nước và các sinh viên nước ngoài học tập ở Việt Nam. Mục tiêu chính là làm sao có thể khiến trải nghiệm người dùng có thể diễn ra liền mạch, đồng thời thu hút và lôi cuốn, cả trên website hay là trên điện thoại. Thiết kế UI nhấn mạnh sự rõ ràng và giản dị

qua việc có thể đảm bảo được sinh viên họ có thể định hướng xuyên suốt quá trình sử dụng hệ thống để đăng tải liệu, chọn máy in và cả quản lý cả thuộc tính in mà họ mong muốn. Cho nên, chúng ta cần tập trung sao cho ứng dụng có thể hoạt động trơn tru và mang cả tính thẩm mỹ trên cả từ điện thoại cho đến máy tính.

Đầu tiên là sự phân bố các mục trên màn hình giao diện người dùng, tránh để cho các mục chồng lên nhau, nên sắp xếp sao cho chúng vừa khít nhau, thiết kế cấu trúc logic cho các menu và options, gộp các chức năng liên quan với nhau vào một nhóm, nhằm không quá gây khó hiểu cho người dùng. Hơn thế nữa, giao diện xuyên suốt phải thống nhất với nhau (nền, màu sắc, hình dạng các nút). Nên hỗ trợ người dùng thông qua giao diện step-by-step, có thể thêm thanh quá trình để cho biết người dùng đang ở bước nào, còn có nút "Home" để họ có thể dễ dàng quay lại màn hình chính, cung cấp người một thiết lập mặc định cho việc in nhanh và cả thiết lập nâng cao.

Hệ thống cũng phải sử dụng linh hoạt các biểu tượng minh họa cho các thao tác như in, tìm kiếm và thiết lập; Về việc chọn máy in, hệ thống phải hiển thị ra một danh sách có thể để người dùng lựa chọn máy in phù hợp theo vị trí, sự có sẵn... Ngoài ra còn có để trạng thái theo màu đỏ, vàng hoặc xanh thể hiện theo tính khả dụng của mỗi máy in; Việc tải các file in có thể thao tác bằng việc kéo và thả file. Để giúp người dùng dễ dàng di chuyển trong hệ thống, thanh Navigation cố định sẽ có các mục chính như "Tải Tài Liệu", "Chọn Máy In", "Thuộc Tính In", và "Trạng Thái In". Thanh này có thể là menu ngang hoặc menu dạng hamburger (đối với giao diện di động) để tiết kiệm không gian nhưng vẫn dễ sử dụng. Mỗi trang sẽ có nút "Tiền" và "Lùi" để người dùng di chuyển qua các bước một cách liền mạch, và nút "Home" luôn sẵn có ở góc trên trái để quay lại màn hình chính. Phần "Trạng Thái In" sẽ sử dụng màu sắc (xanh, vàng, đỏ) để hiển thị tình trạng sẵn sàng của máy in một cách nhanh chóng, thuận tiện cho sinh viên. Tiếp theo, hệ thống còn phải kèm theo thông báo lỗi thân thiện với người dùng, đồng thời cũng phải hướng dẫn người dùng một cách chi tiết và rõ ràng qua vắn đề gặp phải (error handling). Cuối cùng, về tính thẩm mỹ, hệ thống phải cung cấp các tính năng như các nút call-to-action trực quan và tự điều chỉnh cỡ chữ phù hợp với màn hình thiết bị đang sử dụng. Để quảng bá thương hiệu và tông màu sắc theo trường Đại Học Bách Khoa Hồ Chí Minh, đồng thời nhằm tạo ra cảm giác tin tưởng và thân thuộc cho sinh viên, các tông màu chính sẽ được sử dụng là màu xanh dương và màu trắng xen lẫn màu hồng.

1.1.2. Database Approach

Giải pháp lưu trữ dữ liệu với MongoDB

Trong hệ thống Student Printing Service, MongoDB được chọn làm cơ sở dữ liệu chính để lưu trữ dữ liệu liên quan đến các hoạt động của sinh viên, thông tin đơn hàng in ấn, máy in, và người dùng. Việc sử dụng MongoDB – một cơ sở dữ liệu NoSQL dạng tài liệu – giúp hệ thống có khả năng mở rộng linh hoạt, phù hợp với dữ liệu phi cấu trúc và dễ dàng xử lý các yêu cầu có tính thay đổi cao.

Hệ thống áp dụng mô hình kiến trúc Model-View-Controller (MVC), trong đó mỗi lớp Model tương ứng với một collection trong MongoDB. Cụ thể, các Model như LogModel, OrderModel, PrinterModel và UserModel lần lượt quản lý các collection để lưu trữ nhật ký hoạt động, yêu cầu in, thông tin máy in, và thông tin người dùng. Với kiến trúc này, các lớp Controller sẽ chịu trách nhiệm xử lý logic nghiệp vụ, nhận yêu cầu từ View, thao tác với các Model. Model sẽ kết nối đến database để xóa/thêm/cập nhật thông tin, sau đó database trả thông tin về lại cho Model để Model trả thông tin đã được chỉnh sửa về cho View và Controller. MongoDB đảm bảo khả năng lưu trữ và truy xuất dữ liệu nhanh chóng, đồng thời hỗ trợ triển khai các chức năng như tìm kiếm và cập nhật dữ liệu, đáp ứng nhu cầu hoạt động liên tục của hệ thống.

1.1.3. API Management

Để có thể sử dụng các dịch vụ, người dùng cần phải thông qua xác thực của HCMUT SSO. Sau khi đăng nhập, mỗi yêu cầu từ người dùng đều cần phải có token được trả về từ lần đăng nhập trước đó để xác thực và nếu hợp lệ sẽ chuyển yêu cầu này tới các endpoint thích hợp.

Yêu cầu từ Client: Người dùng (client) gửi yêu cầu đến một endpoint được quản lý bởi Controller của hệ thống.

Xử lý của Controller: Controller tiếp nhận và xử lý yêu cầu, sau đó xác định nếu cần phải lấy dữ liệu từ một dịch vụ bên ngoài hoặc từ Database nội bộ.

Phân loại Controller:

- **Payment và Authenticate:** Đối với các yêu cầu liên quan đến thanh toán và xác thực, các controller này sẽ thông qua API Gateway để giao tiếp với các dịch vụ bên ngoài như HCMUT SSO và BKPay.
- **OrderController, PrinterController, và LogController:** Các controller này xử lý các yêu cầu liên quan đến tài liệu in ấn, quản lý máy in và log hệ thống. Chúng sẽ được xác thực tại Gateway bằng token trước khi giao tiếp với

Database thông qua các model tương ứng để truy xuất hoặc cập nhật dữ liệu.

Tương tác với API Gateway:

- Payment và Authenticate cần giao tiếp với các dịch vụ bên ngoài (như HCMUT SSO hoặc BKPay), chúng sẽ gửi yêu cầu qua API Gateway và tại đây chúng sẽ được chuyển hướng sang HCMUT SSo để xác thực hay BKPay.
- API Gateway sẽ chuyển tiếp yêu cầu đến dịch vụ bên ngoài, nhận phản hồi, và áp dụng các chính sách bảo mật hoặc giới hạn tốc độ (rate-limiting) nếu cần.
- Mỗi yêu cầu sau khi đã đăng nhập đều cần có token kèm theo để được kiểm tra tại API Gateway trước khi được chuyển tới các endpoint thích hợp.

***BK Pay:** Hệ thống của chúng tôi không truy cập bất kỳ thông tin nào (hệ thống chỉ gửi yêu cầu về số tiền mà sinh viên cần thanh toán cho việc sử dụng dịch vụ của họ), và thông tin thanh toán sẽ được xử lý bởi bộ phận BKPay, đồng thời hệ thống chỉ yêu cầu khả năng điều hướng từ dịch vụ BKPay

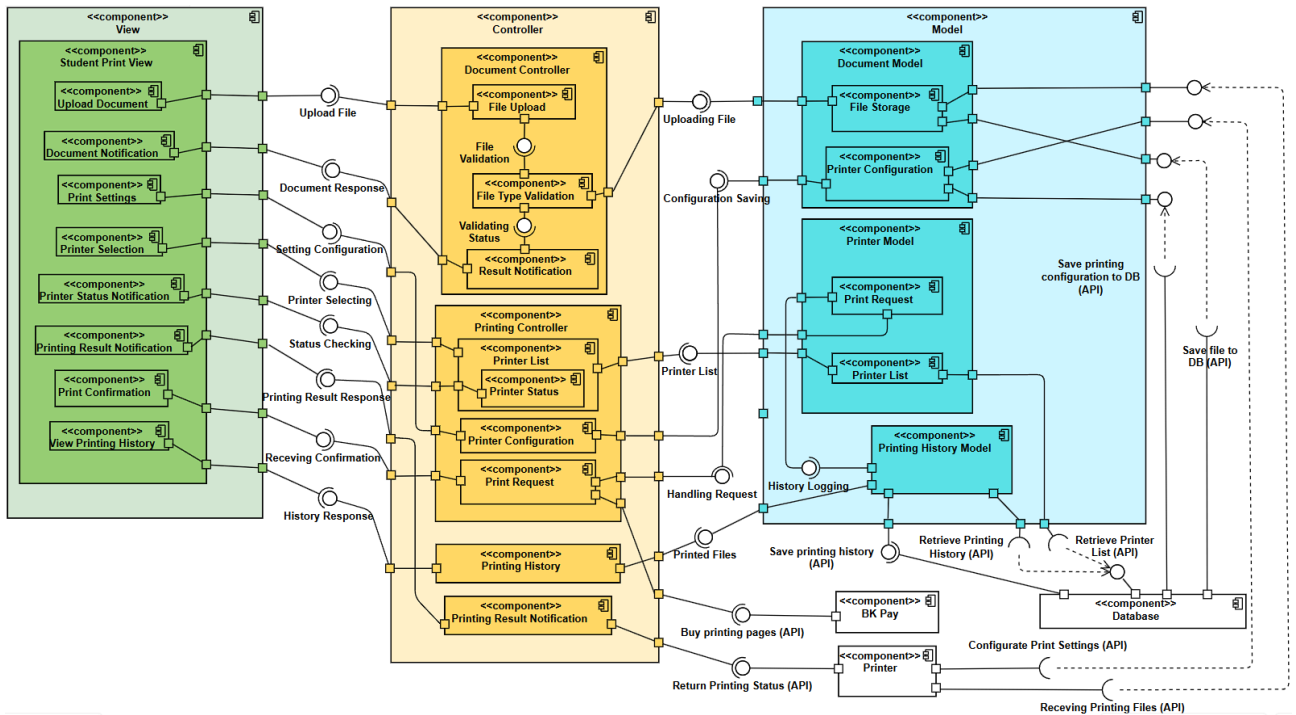
***HCMUT SSO:** Endpoint này được sử dụng cùng với SSPS application ID, được đăng ký thông qua bộ phận xác thực trung tâm của HCMUT, cùng với URL callback đến trang web HCMUT SSO. Application ID, còn được gọi là client trong mô hình client-server của dịch vụ xác thực CAS, hoạt động như một API key để truy cập vào tính năng của dịch vụ HCMUT SSO.

Xử lý phản hồi từ API Gateway:

- Nếu đăng nhập thành công, HCMUT SSO hay BKPay sẽ điều hướng về lại trang web của chúng ta kèm theo token để xác nhận.
- Sau khi đã đăng nhập thành công, API gateway sẽ kiểm tra token xác thực trong mỗi yêu cầu của người dùng, nếu hợp lệ thì sẽ chuyển yêu cầu của người dùng về internal endpoint được định nghĩa trong Controller. Và trong các endpoint này sẽ gọi các hàm trong Model qua đó Model sẽ tương tác với Database. Sau đó, Controller sẽ trực tiếp xử lý và định dạng dữ liệu lấy từ các model.

Trả Kết quả về Client: Sau khi xử lý và định dạng lại dữ liệu, Controller sẽ trả kết quả về cho View để hiển thị cho người dùng.

1.2. Task 3.2: Component Diagram



View (Giao diện người dùng)

- **Student Print View:** Giao diện cho phép sinh viên tải tệp lên và thực hiện thao tác in.
- **Document Notification:** Thông báo cho sinh viên về trạng thái tài liệu đã tải lên hoặc gửi để in.
- **Print Settings:** Giao diện cho phép sinh viên thiết lập cấu hình in như khổ giấy, số trang, màu sắc, v.v.
- **Printer Selection:** Cho phép sinh viên chọn máy in.
- **Printer Status Notification:** Hiển thị thông báo về trạng thái máy in, chẳng hạn như sẵn sàng, đang in, lỗi.
- **Printing Result Notification:** Thông báo kết quả in cho sinh viên sau khi hoàn tất.
- **Print Confirmation:** Xác nhận thông tin in trước khi in thực tế.
- **View Printing History:** Cho phép sinh viên xem lịch sử các lần in trước đây.

Controller (Điều khiển)

- **Document Controller:**
 - **File Upload:** Xử lý tải tệp lên từ sinh viên.
 - **File Type Validation:** Kiểm tra và xác minh loại tệp có hợp lệ không.
 - **Result Notification:** Gửi thông báo về trạng thái xác thực tệp.
- **Printing Controller:**
 - **Printer List:** Lấy danh sách máy in có sẵn để sinh viên lựa chọn.
 - **Printer Status:** Kiểm tra trạng thái của các máy in.
 - **Printer Configuration:** Lưu cấu hình in của sinh viên.
 - **Print Request:** Gửi yêu cầu in đến máy in.
- **Printing History:** Lưu trữ và truy xuất thông tin lịch sử in
- **Printing Result Notification:** Thông báo kết quả sau khi in xong.

Model (Mô hình)

- **Document Model:**
 - **File Storage:** Lưu trữ tệp đã tải lên của sinh viên.
- **Printer Model:**
 - **Printer Configuration:** Cấu hình máy in cho các yêu cầu in khác nhau.
 - **Print Request:** Nhận yêu cầu in từ Printing Controller.
 - **Printer List:** Quản lý danh sách các máy in.
- **Printing History Model:**
 - **History Logging:** Lưu lịch sử các lần in của sinh viên.

External Systems (Hệ thống bên ngoài)

- **BK Pay:** Hệ thống thanh toán cho sinh viên mua thêm trang in nếu cần.
- **Printer:** Máy in vật lý nhận các yêu cầu in từ hệ thống và thực hiện in.
- **Database:** Lưu trữ dữ liệu, bao gồm tệp tin, cấu hình in, và lịch sử in.

Luồng hoạt động chính

1. **Tải tệp lên:** Sinh viên truy cập **Student Print View** và chọn **Upload Document** để tải tệp cần in. Tệp được xử lý qua **File Upload** và **File Type**

- Validation** để kiểm tra định dạng, sau đó lưu vào **File Storage** nếu hợp lệ và gửi thông báo lại cho sinh viên.
2. **Thiết lập cấu hình in:** Sinh viên tùy chỉnh các thiết lập in (như số trang, khổ giấy) qua **Print Settings**. Cấu hình này được lưu lại trong **Printer Configuration** để chuẩn bị cho yêu cầu in.
 3. **Chọn máy in:** Sinh viên chọn máy in qua **Printer Selection**. **Printing Controller** truy xuất **Printer List** để lấy danh sách máy in và kiểm tra trạng thái qua **Printer Status**, sau đó hiển thị các máy in khả dụng cho sinh viên lựa chọn.
 4. **Xác nhận và gửi yêu cầu in:** Sau khi sinh viên xác nhận thông tin qua **Print Confirmation**, **Printing Controller** gửi yêu cầu in đến **Print Request**, truyền lệnh đến máy in vật lý để thực hiện in.
 5. **Xử lý thanh toán:** Nếu sinh viên hết trang in, **Printer Configuration** kết nối **BKPay** để sinh viên mua thêm trang in. Sau khi thanh toán thành công, sinh viên tiếp tục in.
 6. **Lưu trữ và thông báo kết quả:** Sau khi in xong, **Printing Result Notification** gửi thông báo cho sinh viên. **Printing History** ghi lại chi tiết lần in vào **Printing History Model** và lưu vào **Database**.
 7. **Xem lịch sử in:** Sinh viên truy cập **View Printing History** để xem lại các lần in, **Printing Controller** truy xuất thông tin từ **Printing History Model** và **Database** để hiển thị danh sách in.