

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



Software Engineering

Assignment Report

A Smart Printing Service for Students at HCMUT

Software Requirements Specification

Advisor(s): Trương Tuấn Anh

Student(s): 1. Nguyễn Phước Ngọc Hương - 2252285

2. Nguyễn Lê Minh Huân - 2252242

3. Nguyễn Quốc Lộc - 2252458

4. Nguyễn Lê Văn Tú - 2252881

5. Nguyễn Thành Phát - 2252604

HO CHI MINH CITY, NOVEMBER 2024



Contents

1 Task 3: Architecture design	3
1.1 Layered Architecture, Presentation strategies, Data storing approach and API management	3
1.1.1 Layered Architecture	3
1.1.2 Presentation strategies	4
1.1.3 Data storing approach	5
1.1.4 API management	6
1.2 Component Diagram	7

List of Figures

1.1 Layered Architecture	3
1.2 Component diagram	7

List of Tables

Revision History

Name	Date	Reason For Changes	Version
Nguyễn Quốc Lộc	19/09/2024	Tạo bản nháp ban đầu cho task 1	1.0
Nguyễn Phước Ngọc Hương	19/09/2024	Chỉnh sửa phần 1.1, 1.2, 1.3	1.1
Nguyễn Lê Văn Tú	20/09/2024	Chỉnh sửa phần 1.4, 1.5	1.2
Nguyễn Quốc Lộc	22/09/2024	Thêm mục lục	1.3
Nguyễn Phước Ngọc Hương	22/09/2024	Chỉnh sửa phần 1.4, 1.5	1.4

1 Task 3: Architecture design

1.1 Layered Architecture, Presentation strategies, Data storing approach and API management

1.1.1 Layered Architecture

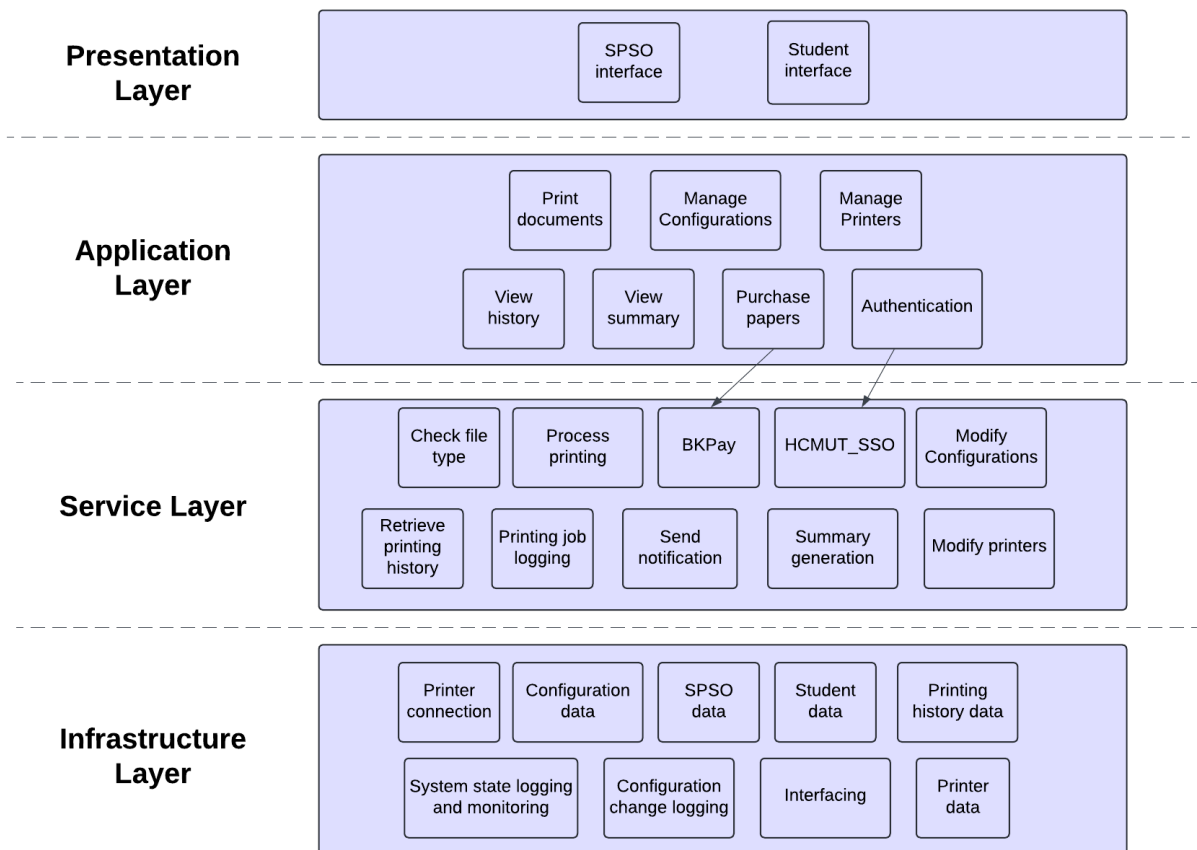


Figure 1.1: Layered Architecture

Hệ thống được sử dụng và quản lý bởi 2 đối tượng chính là Sinh viên và SPSO, vì vậy có hai giao diện chính là giao diện cho Sinh viên và giao diện cho SPSO.

- + Đối với giao diện cho Sinh viên, những chức năng có thể thực hiện được là Đăng nhập (Authentication), In tài liệu (Print Documents), Mua giấy (Purchase papers), Xem lịch sử in (View history). Trong đó, Đăng nhập sẽ truy cập tới hệ thống bên ngoài là HCMUT_SSO ở tầng Service để xác thực người dùng. Chức năng Mua giấy sẽ truy cập tới hệ thống bên ngoài là BKPay. Chức năng In tài liệu sẽ yêu cầu hệ thống kiểm tra loại file, và chạy tiến trình in, đồng thời lưu lịch sử in vào

database. Chức năng Xem lịch sử in sẽ yêu cầu hệ thống lấy thông tin về lịch sử in từ database. Đồng thời, dữ liệu của Sinh viên cũng được lưu lại ở database.

- + Đối với giao diện cho SPSO, có những chức năng chính là Đăng nhập (Authentication), Quản lý cấu hình (Manage Configurations), Quản lý máy in (Manage Printers), Xem lịch sử (View history), Xem báo cáo, tóm tắt (View summary). Chức năng Đăng nhập và Xem lịch sử sẽ tương tự với chức năng Đăng nhập của Sinh viên. Chức năng Quản lý cấu hình sẽ truy cập tới chỉnh sửa Cấu hình, và thông tin cấu hình sẽ được lưu ở database. Chức năng Quản lý máy in sẽ cho phép chỉnh sửa thông tin của máy in, và dữ liệu về máy in cũng như kết nối máy in sẽ được lưu về database. Chức năng Xem báo cáo, tóm tắt sẽ yêu cầu hệ thống tạo ra báo cáo và tóm tắt từ dữ liệu ở database.
- + Ngoài ra, hệ thống sẽ gửi thông báo khi người dùng thực hiện lệnh và tự động lưu dữ liệu để quản lý các giao diện để trao đổi dữ liệu giữa các tầng và với hệ thống bên ngoài. Đồng thời, giám sát và ghi lại hiệu suất của hệ thống, bao gồm việc sử dụng tài nguyên và các vấn đề tiềm ẩn.

1.1.2 Presentation strategies

Đây là tầng đầu tiên trong kiến trúc hệ thống. Chúng tôi sẽ tập trung vào sự đơn giản, dễ sử dụng và tối ưu hóa trải nghiệm người dùng. Để đạt được điều này, chúng tôi sẽ sử dụng một số công nghệ hiện đại và các phương pháp thiết kế cụ thể:

- Thư viện và Framework cho giao diện người dùng (Front-end library and framework): Chúng tôi lựa chọn sử dụng ngôn ngữ HTML, CSS và Javascript để phát triển giao diện người dùng (front-end). Những ngôn ngữ trên giúp chúng tôi xây dựng các giao diện linh hoạt, dễ mở rộng và tối ưu hiệu suất.
- Thiết kế đáp ứng (Responsive Design): Đảm bảo hệ thống có thể hoạt động tốt trên nhiều thiết bị khác nhau. Các thành phần giao diện và biểu mẫu sẽ được thiết kế để thích nghi với các kích thước màn hình khác nhau nhằm đảm bảo trải nghiệm người dùng nhất quán trên mọi thiết bị.
- Tính năng thân thiện với người dùng (User-Friendly Features): Chúng tôi tập trung vào việc tạo ra một giao diện trực quan, dễ sử dụng, với các nút bấm, biểu mẫu và menu đơn giản, thân thiện. Điều này giúp người dùng, dù là lần đầu tiếp cận, vẫn có thể sử dụng hệ thống một cách dễ dàng và hiệu quả.

Với việc sử dụng những ngôn ngữ trên kết hợp với thiết kế thân thiện và trải nghiệm người dùng được tối ưu, chúng tôi tin rằng giao diện của hệ thống SSPS sẽ đem lại ấn tượng tốt, đáp ứng nhu cầu của cả sinh viên và cán bộ. Giao diện này cũng sẽ tích hợp liền mạch với các lớp khác trong kiến trúc tầng của hệ thống.

1.1.3 Data storing approach

Việc lưu trữ, quản lý dữ liệu cho hệ thống Smart Printing Service (HCMUT-SSPS), một kiến trúc tầng (layered architecture) được áp dụng với sự kết hợp của các hệ thống cơ sở dữ liệu khác nhau. Cụ thể, lớp dưới cùng là lớp cơ sở dữ liệu, nơi lưu trữ toàn bộ thông tin cần thiết, bao gồm dữ liệu sinh viên, máy in, lịch sử in ấn, và các giao dịch mua trang in. Các dữ liệu của ứng dụng sẽ được lưu trữ tại đây và các thao tác như search, insert, update and delete sẽ được thực hiện thường xuyên để thao tác với dữ liệu thông qua hệ quản trị cơ sở dữ liệu. Nhóm sẽ sử dụng kiến trúc nhiều lớp (layered architecture) và lớp cơ sở dữ liệu (database layer) sẽ được lưu trữ bằng cơ sở dữ liệu NoSQL (MongoDB). Mỗi thực thể có thể được lưu dưới dạng tài liệu JSON với các thuộc tính liên quan. MongoDB giúp quản lý và lưu trữ dữ liệu phi cấu trúc, linh hoạt hơn với dữ liệu thay đổi thường xuyên và các dữ liệu phức tạp không nhất thiết có quan hệ ràng buộc.

Đối với hệ thống SSPS chúng ta cần có các kiểu collection như sau:

- Sinh viên: Lưu thông tin của sinh viên dưới các trường như: mã số sinh viên, họ và tên, mật khẩu, email, số dư trang in
- SPSO: Lưu thông tin SPSO dưới các trường như: mã SPSO, họ và tên, tên tài khoản, mật khẩu, email và số điện thoại
- Máy in: Lưu thông tin của từng máy in dưới các trường như: mã máy in, tên máy in, nhãn hiệu, mẫu máy, mô tả, vị trí (cơ sở, tòa, phòng), trạng thái.
- Tài liệu: Lưu thông tin của tài liệu dưới các trường như: tên tài liệu, định dạng tài liệu, số trang tài liệu.
- Cấu hình tài liệu: gồm các thông tin như cỡ giấy (A4, A3), số bản sao, màu sắc (đen trắng hoặc màu), in hai mặt.
- Đơn in: Lưu thông tin của đơn in dưới các trường như: mã đơn in, cấu hình (hướng giấy, cỡ giấy, tỉ lệ, số trang in, số bản sao), trạng thái.
- Đơn mua trang: Lưu thông tin của đơn mua hàng dưới các trường như: Mã giao dịch, thời điểm giao dịch, số lượng trang mua, giá tiền.

1.1.4 API management

Giao diện lập trình ứng dụng hay còn gọi tắt là API (Application Programming Interface), là tập hợp các giao thức truyền thông và chương trình con được nhiều chương trình khác nhau sử dụng để giao tiếp giữa chúng. Một lập trình viên có thể sử dụng nhiều công cụ API khác nhau để làm cho chương trình của họ dễ dàng và đơn giản hơn. Ngoài ra, API hỗ trợ lập trình viên bằng cách hiệu quả để phát triển các chương trình phần mềm của họ. Do đó, ý nghĩa của API là khi API giúp hai chương trình hoặc ứng dụng giao tiếp với nhau bằng cách cung cấp cho chúng các công cụ và chức năng cần thiết. API tiếp nhận yêu cầu từ người dùng và gửi đến nhà cung cấp dịch vụ, sau đó lại gửi kết quả được tạo ra từ nhà cung cấp dịch vụ đến người dùng mong muốn.

Các API cho hệ thống in ấn tự động SSPS bao gồm:

- Authentication API: Đảm bảo rằng chỉ những người dùng đã được xác thực (sinh viên hoặc SPSO) mới có thể truy cập vào hệ thống.
- File Management API: Sinh viên tải lên và quản lý các tài liệu cần in.
- Printing Configuration API: Sinh viên cấu hình các thuộc tính in ấn cho từng tài liệu (kích thước giấy, số bản sao, in một mặt/hai mặt).
- Printer Management API: Cung cấp các chức năng quản lý máy in cho SPSO, bao gồm kiểm tra trạng thái máy in, thêm hoặc cập nhật máy in.
- Page Management API: Quản lý số trang in còn lại của sinh viên, bao gồm kiểm tra và mua thêm trang in.
- Print Request API: Xử lý yêu cầu in từ sinh viên, bao gồm việc gửi lệnh in tới máy in được chọn và ghi lại thông tin in.
- Notification API: Gửi thông báo cho sinh viên về các trạng thái và sự kiện liên quan đến quá trình in ấn.
- Print History Management API: Lưu trữ và truy xuất lịch sử các hoạt động in ấn của sinh viên và báo cáo cho SPSO.

1.2 Component Diagram

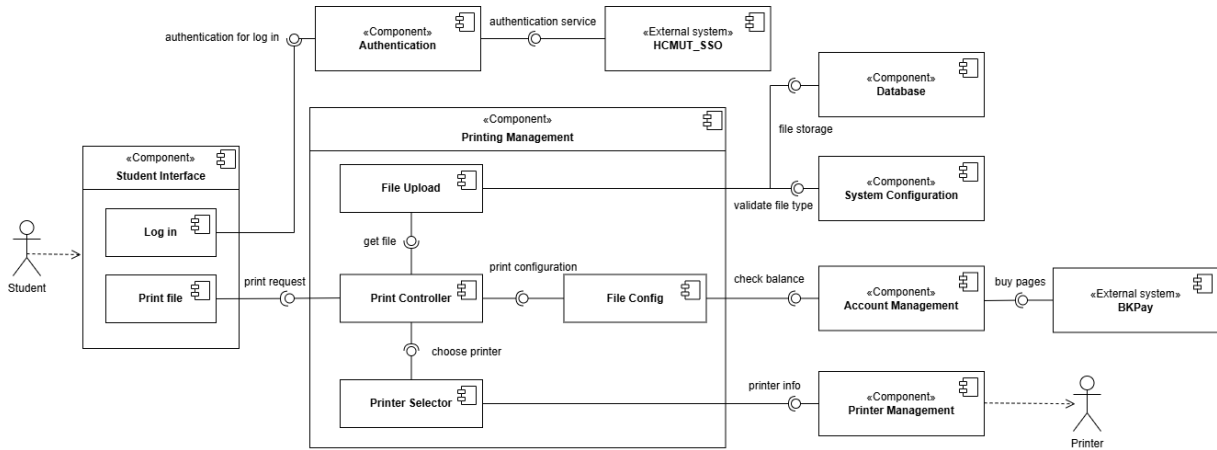


Figure 1.2: Component diagram

Mô tả Component Diagram cho module Printing Module:

- Component diagram bao gồm các component: Student Interface, Authentication, Printing Management, Database, System Configuration, Account Management, BKPay, Printer Management và các External System gồm HCMUT_SSO và BKPay.
- Student Interface: Đây là giao diện mà người dùng tương tác với hệ thống. Bao gồm các component: Log in, Print file.
- Authentication: Xác thực, kết nối với hệ thống HCMUT_SSO để sử dụng dịch vụ xác thực tập trung nhằm xác minh danh tính của sinh viên khi đăng nhập.
- Printing Management: Quản lý quy trình in, bao gồm các component bên trong:
 - File Upload: cho phép sinh viên tải các tệp cần in lên; được kết nối với component System Configuration để kiểm tra loại tệp hợp lệ và Database để lưu lại file.
 - File Config: Điều chỉnh cấu hình các trang in; kết nối với component Account Management để kiểm tra số trang in sinh viên hoặc mua thêm trang in.
 - Printer Selector: Component phụ trách việc chọn máy in của sinh viên; được kết nối với component Printer Management để lấy thông tin các máy in.
 - Printer Controller: Điều khiển quá trình in; được kết nối với các component File Upload, File Config, Printer Selector.



- Database: Cơ sở dữ liệu của hệ thống.
- System Configuration: Component quản lý toàn bộ config mà SPSO thiết lập trên hệ thống.
- Account Management: Quản lý các thông tin của tài khoản sinh viên như số dư trang in, kết nối với hệ thống thanh toán BKPay để quản lý việc mua trang in.
- Printer Management: Quản lý toàn bộ thông tin và hoạt động các máy in.
- External System (HCMUT_SSO và BKPay): Hệ thống bên ngoài; bao gồm hệ thống HCMUT_SSO của trường để xác thực tài khoản của sinh viên và BKPAY là hệ thống để xử lý thanh toán việc mua giấy in.



References