

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)

---

BÀI TẬP LỚN

# HỆ THỐNG HCMUT-SSPS DỊCH VỤ IN ẤN THÔNG MINH

---

Giảng viên hướng dẫn: Trần Trương Tuấn Phát

Lớp: L02

Sinh viên: Nguyễn Phạm Quốc An - 2210020

Ngô Quang Bách - 2110770

Nguyễn Thùy Dung - 2210487

Phạm Tuấn Đạt - 2210714

Lưu Trần Minh Khang - 2211436

Hà Khởi Nguyên - 2212287

Trần Nam Sơn - 2212956

Nguyễn Anh Tài - 2212980

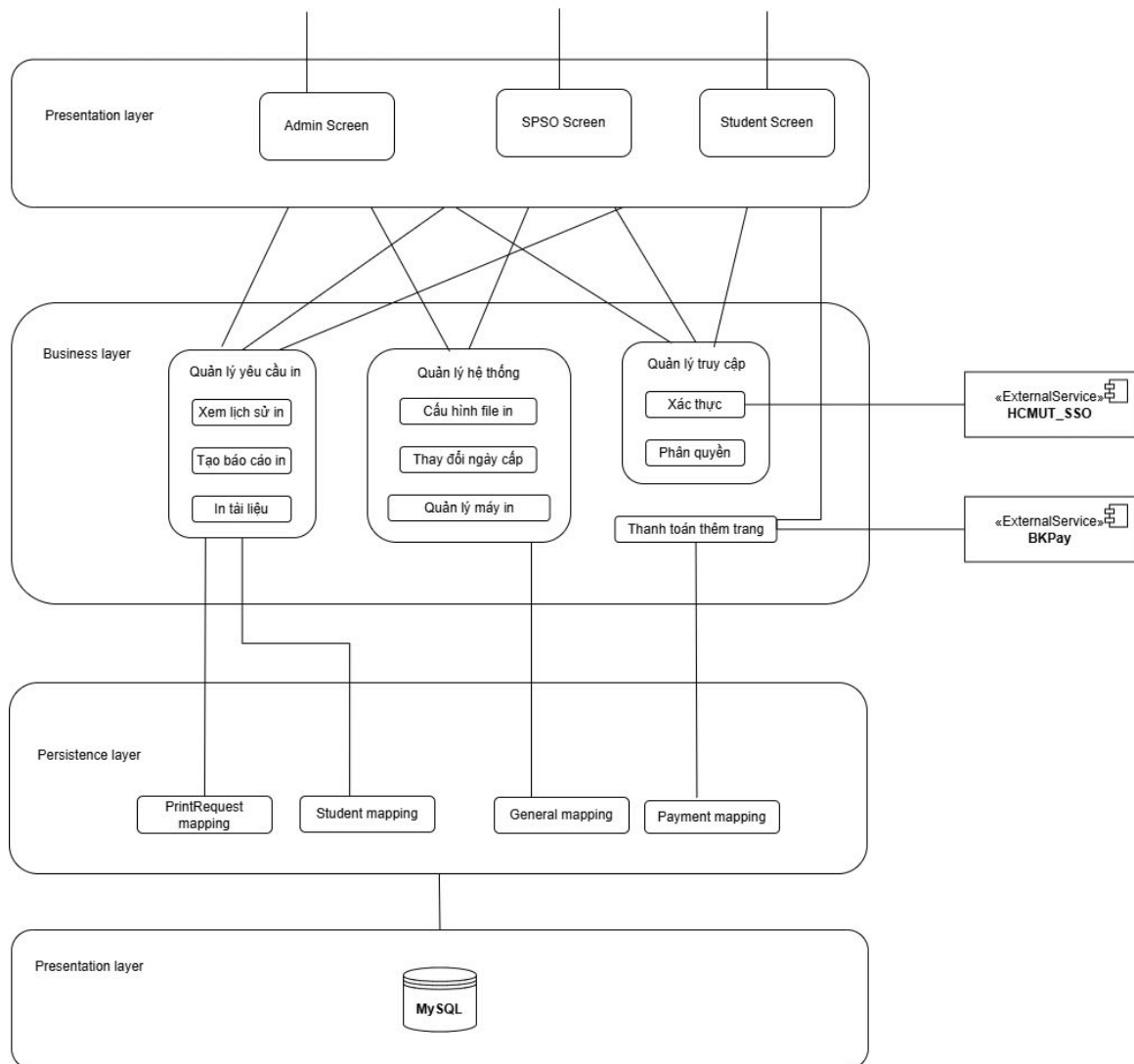
Phan Thanh Tấn - 2213076

Nguyễn Đình Tuấn - 2213781

Thành phố Hồ Chí Minh, Ngày 9 tháng 11 năm 2024

### 3 Thiết kế kiến trúc

#### 3.1 Sử dụng kiến trúc phân lớp để thiết kế hệ thống HCMUT-SSPS



Hình 1: Kiến trúc phân lớp

##### 3.1.1 Presentation Strategy

Presentation Layer là nơi cung cấp giao diện người dùng cho từng loại người dùng khác nhau, bao gồm Admin, SPSO (Student Printing Service Officer), và Student. Giao diện người dùng cho phép các đối tượng tương tác với hệ thống thông qua các màn hình như:

- Admin Screen: Giao diện cho quản trị viên để theo dõi và quản lý các hoạt động của SPSO và sinh viên.

- SPSO Screen: Giao diện dành cho SPSO gồm các trang chính như: quản lý máy in (để quản lý cấu hình máy in), trang thông tin chung (quản lý các lịch sử in ấn của các máy in, các sinh viên và tạo báo cáo).
- Student Screen: Giao diện dành cho sinh viên gồm các trang chính như: trang chủ (xem các thông báo về HCMUT\_SSPO), trang in tài liệu (thực hiện các yêu cầu in ấn), trang thông tin chung (kiểm tra lịch sử in, tạo báo cáo) và trang mua trang in (thực hiện yêu cầu thanh toán thêm trang khả dụng).

Mỗi loại người dùng sẽ có quyền truy cập vào các chức năng phù hợp thông qua giao diện riêng, giúp hệ thống rõ ràng và thân thiện với người dùng. Việc phân chia giao diện thành từng màn hình cụ thể cho từng loại người dùng giúp dễ dàng quản lý quyền truy cập và cải thiện trải nghiệm người dùng.

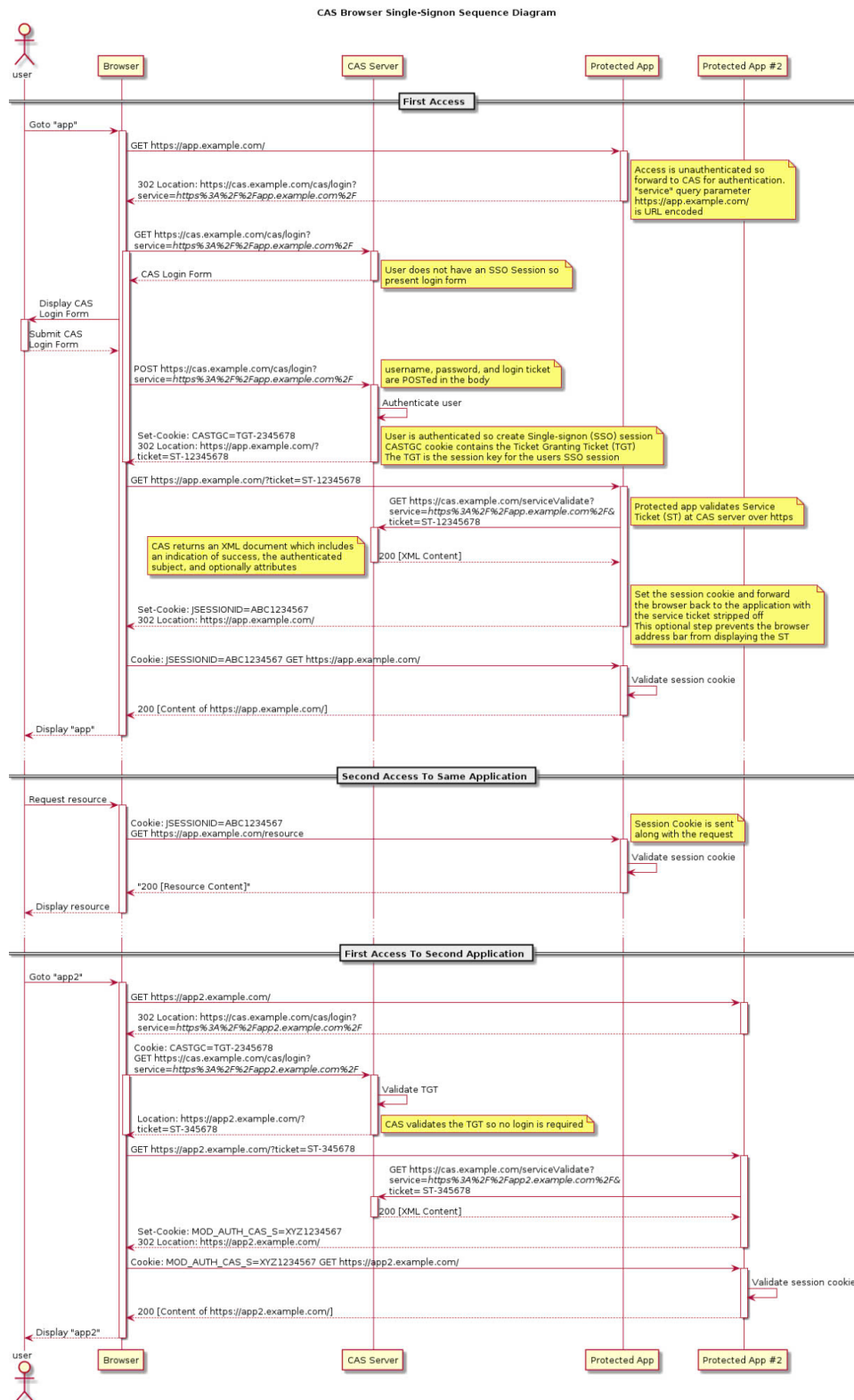
Để cung cấp trải nghiệm người dùng thuận tiện, hệ thống HCMUT-SSPO sẽ triển khai giao diện trên ứng dụng web. Ứng dụng web sẽ được thiết kế theo hướng responsive để tương thích với nhiều kích cỡ màn hình, tối ưu hóa cho sinh viên và cán bộ quản lý (SPSO) khi truy cập từ trình duyệt trên máy tính hoặc thiết bị di động. Framework React sẽ được sử dụng để đảm bảo giao diện mượt mà, dễ bảo trì và dễ mở rộng. [Xem thêm](#)

### 3.1.2 Data Storage Approach

Nhóm sẽ sử dụng MySQL cho hệ thống HCMUT\_SSPO. MySQL là cơ sở dữ liệu SQL thân thiện với người dùng, được biết đến với khả năng mở rộng, hiệu suất và tính linh hoạt cao. Nó là một lựa chọn phổ biến cho người mới bắt đầu do tính dễ sử dụng và quen thuộc. Ngoài ra, mô hình dữ liệu hướng tài liệu của MySQL cung cấp dữ liệu phi cấu trúc bằng lược đồ linh hoạt, khiến nó trở thành một lựa chọn phù hợp cho các hệ thống có yêu cầu dữ liệu ngày càng tăng. Hơn nữa, khả năng sao chép và phân chia gốc của MySQL cho phép hệ thống mở rộng quy mô theo chiều ngang khi số lượng dữ liệu tăng lên, đảm bảo tính sẵn sàng cao và khả năng chịu lỗi. [Xem thêm](#)

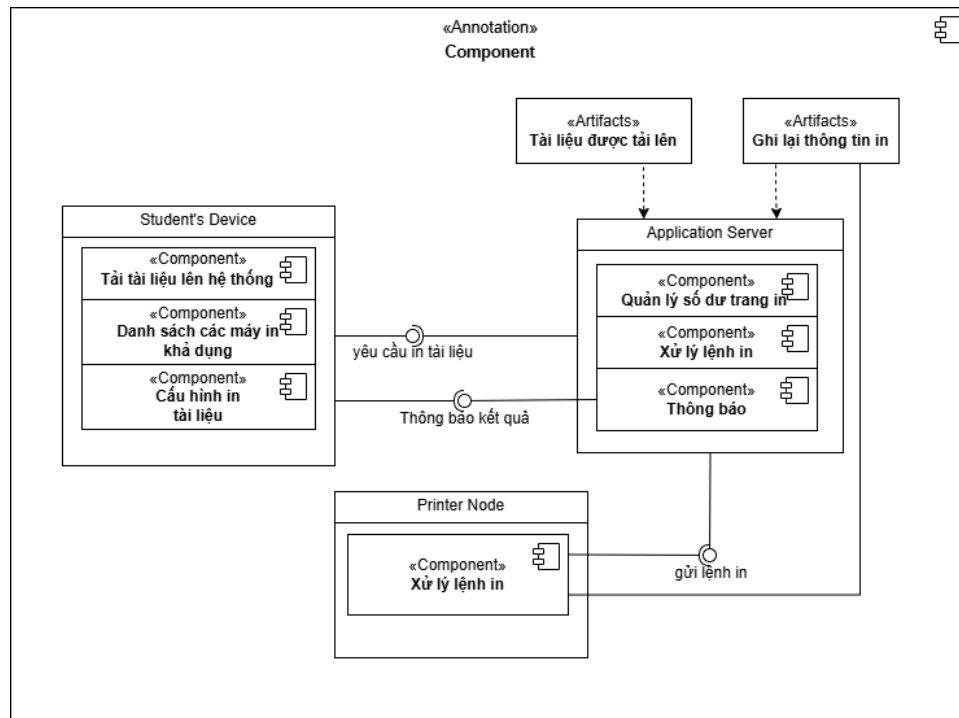
### 3.1.3 External services/API

Theo yêu cầu, hệ thống HCMUT\_SSPO phải hoạt động với hệ thống xác thực của trường đại học Bách Khoa. HCMUT\_SSO sử dụng giao thức Dịch vụ xác thực trung tâm (CAS) để cung cấp dịch vụ đăng nhập một lần (SSO). Cụ thể, nó hoạt động như sau: (vì không có tài liệu rõ ràng về HCMUT\_CAS nên nhóm giả định đó là "Máy chủ CAS" và ứng dụng của nhóm là một trong hai "Ứng dụng được bảo vệ" trong sơ đồ). [Xem thêm](#)



Hình 2: External services

### 3.2 Sơ đồ thành phần cho một module quan trọng (module được sử dụng trong task 2.1)



Hình 3: Component diagram

Component diagram mô tả chức năng in tài liệu của sinh viên, với các thành phần (components) chính bao gồm: Student's Device, Application Server, và Printer Node. Hệ thống cho phép sinh viên tải tài liệu lên, cấu hình thông số in, lựa chọn máy in khả dụng, và nhận thông báo khi in xong.

Đối với **Student's Device**: Component **Tải tài liệu lên hệ thống** cho phép sinh viên tải tài liệu cần in lên hệ thống, gửi thông tin tài liệu đến máy chủ ứng dụng. Component **Danh sách các máy in khả dụng** lấy dữ liệu từ **Application Server** và hiển thị danh sách các máy in mà sinh viên có thể sử dụng. Component **Cấu hình in tài liệu** giúp sinh viên chọn các thông số in như số bản in, kiểu in, và các tùy chọn khác. Sau khi cấu hình, nó sẽ gửi yêu cầu in đến **Application Server**.

Đối với **Application Server**: Component **Quản lý số dư trang in** kiểm tra và quản lý số lượng trang in mà sinh viên còn lại trong tài khoản. Component **Xử lý lệnh in** tiếp nhận và xử lý các yêu cầu in từ **Student's Device**, xem xét nếu tài liệu ở định dạng phù hợp và số trang in còn lại đủ để thực hiện lệnh in thì sẽ tiến hành in ở máy in đã được chọn, nếu không thì hủy lệnh in. Component **Thông báo** gửi thông báo cho sinh viên về kết quả của quá trình xử lý lệnh in.

Ngoài ra còn có các artifacts: **Tài liệu được tải lên**: Lưu trữ tài liệu mà sinh viên đã tải



lên để chuẩn bị cho việc in. **Ghi lại thông tin in:** Lưu trữ các thông tin về lệnh in, bao gồm thời gian, số lượng trang in và các thông tin chi tiết khác.

Đối với **Printer Node**: Component **Xử lý lệnh in** nhận lệnh in và các thông tin in đã được lưu lại từ **Application Server** và thực hiện việc in tài liệu.