ỦY BAN NHÂN DÂN THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC SÀI GÒN

KHOA CÔNG NGHỆ THÔNG TIN



******

**Học phần: Kiểm thử phần mềm**

**Lớp: DCT122C4**

|  |  |
| --- | --- |
| **Thành viên nhóm:** | **3122411049 - Lê Gia Hân** |
|  | **3122411079 - Nguyễn Thị Thu Hường** |
|  | **3122411256 - Nguyễn Ngọc Thúy Vy** |
|  | **3122411141 - Phan Thị Hồng Nhiên** |

**Giảng viên: Đỗ Như Tài**

**MỤC LỤC**

[THIẾT KẾ KIẾN TRÚC CHUẨN C4 MODEL 4](#_Toc209919037)

[Bài 1. Vẽ lại sơ đồ kiến trúc SPA (Single Page Application): 4](#_Toc209919038)

[Bài 2: Vẽ sơ đồ triển khai CI/CD 4](#_Toc209919039)

[Bài 3: Vẽ sơ đồ API của hệ thống: 6](#_Toc209919040)

[Bài 4: Vẽ sơ đồ C1 - System Context 7](#_Toc209919041)

[Bài 5: Vẽ sơ đồ C2 – Container 9](#_Toc209919042)

[Bài 6: Vẽ sơ đồ C3 – Component (High-Level) 11](#_Toc209919043)

[Bài 7. Vẽ sơ đồ C3 – Component (Module-Level) 13](#_Toc209919044)

[Bài 8: Vẽ sơ đồ xử lý 1 request: 14](#_Toc209919045)

[BÀI TẬP ỨNG DỤNG: 15](#_Toc209919046)

[Thiết kế kiến trúc phần mềm theo chuẩn C4 model: 15](#_Toc209919047)

[Level 1: (Context Diagram): 15](#_Toc209919048)

[Level 2 (Container Diagram): 16](#_Toc209919049)

[Level 3 (Component Diagram): 16](#_Toc209919050)

[Level 4 (Code/Implementation Diagram): 17](#_Toc209919051)

**PHÂN CÔNG CÔNG VIỆC**

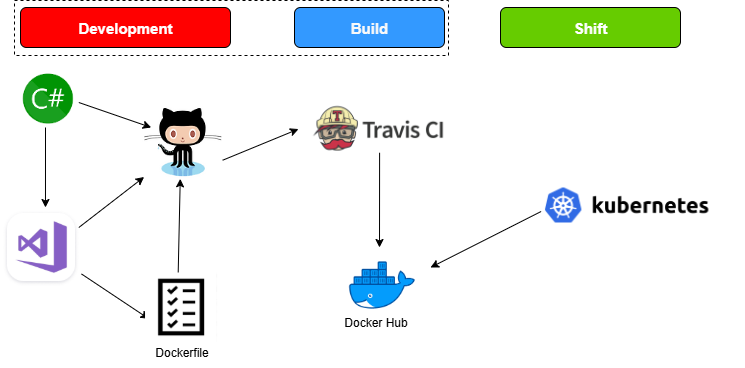
|  |  |  |
| --- | --- | --- |
| **Mã số sinh viên** | **Tên thành viên** | **Công việc** |
| 3122411049 | Lê Gia Hân | Câu 6, 7, Level 1 (BTUD) |
| 3122411079 | Nguyễn Thị Thu Hường | Câu 1, 8, Level 2 (BTUD) |
| 3122411256 | Nguyễn Ngọc Thúy Vy | Câu 4, 5, Level 3 (BTUD) |
| 3122411141 | Phan Thị Hồng Nhiên | Câu 2, 3, Level 4 (BUTD) |

# **THIẾT KẾ KIẾN TRÚC CHUẨN C4 MODEL**

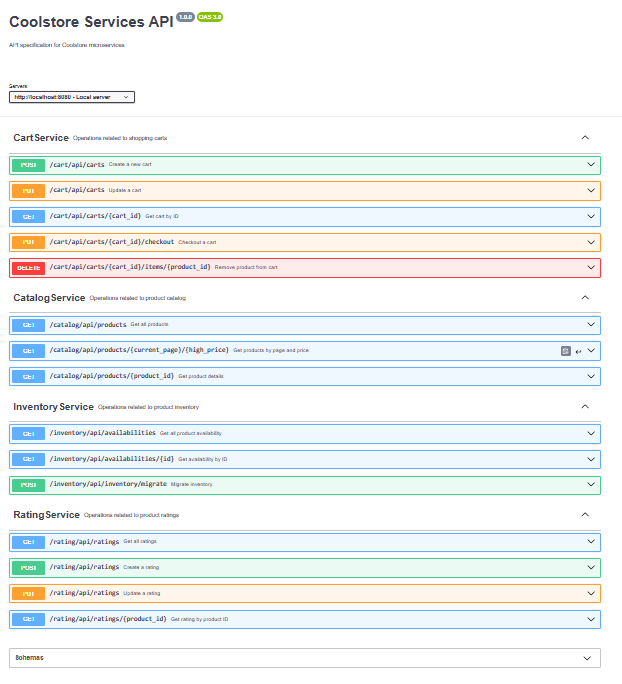
## **Bài 1. Vẽ lại sơ đồ kiến trúc SPA (Single Page Application):**

****

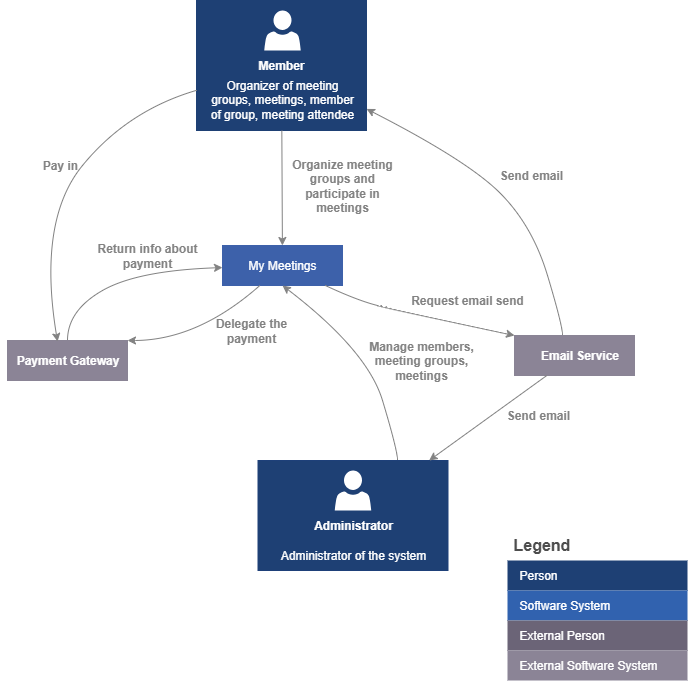
## **Bài 2: Vẽ sơ đồ triển khai CI/CD**

****

## **Bài 3: Vẽ sơ đồ API của hệ thống:**

****

## **Bài 4: Vẽ sơ đồ C1 - System Context**

****

**Các thành phần chính trong hệ thống:**

* Người dùng (Person):  
  + Member (Thành viên)  
   Vai trò: người tổ chức nhóm họp, tham gia cuộc họp, thành viên của nhóm. Có thể tạo nhóm họp, tham gia họp, và thanh toán phí qua Payment Gateway.  
  + Administrator (Quản trị viên hệ thống)  
   Vai trò: quản trị toàn bộ hệ thống (quản lý thành viên, nhóm họp, và các cuộc họp).
* Hệ thống phần mềm (Software System):  
  + My Meetings: đây là hệ thống chính được xây dựng, nơi thành viên và quản trị viên tương tác.
* Hệ thống bên ngoài (External Software System):  
  + Payment Gateway (Cổng thanh toán): xử lý thanh toán cho hệ thống.  
  + Email Service (Dịch vụ email): hỗ trợ gửi email thông báo.

**Các luồng tương tác:**

* Member ↔ My Meetings  
  + Tổ chức nhóm họp, tham gia cuộc họp.

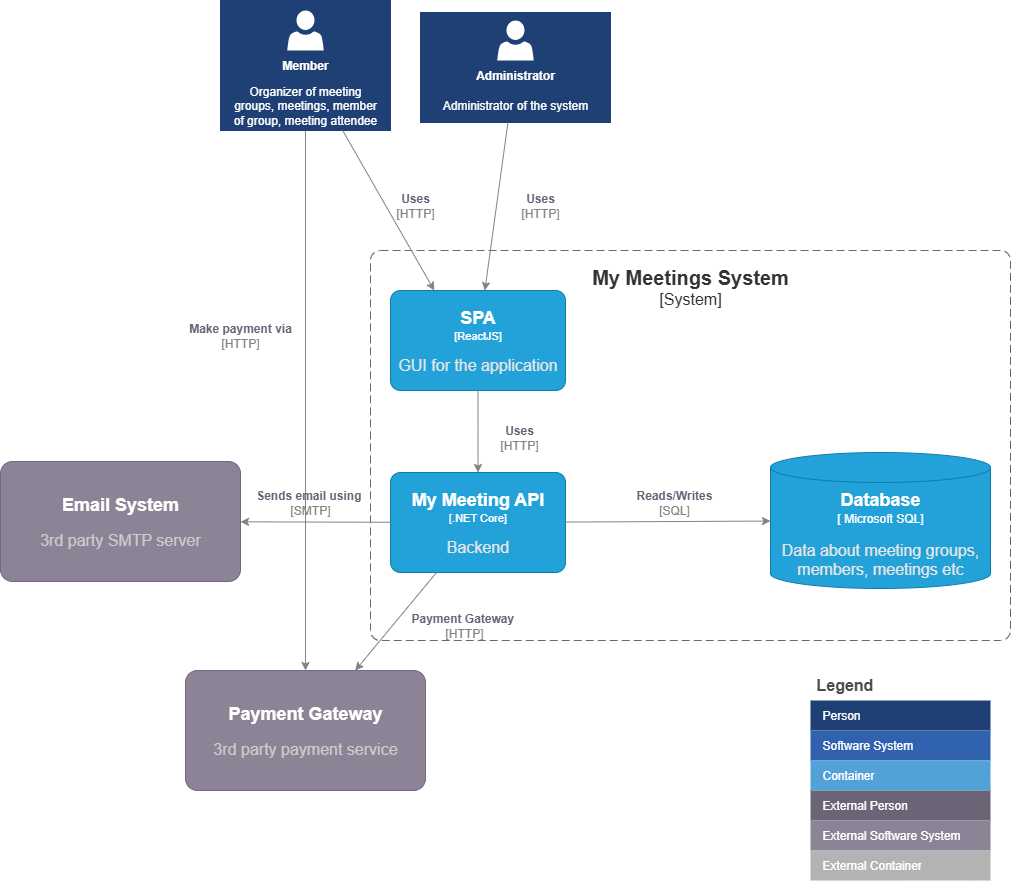
+ Hệ thống gửi email thông báo cho Member qua Email Service.

* Member ↔ Payment Gateway  
  + Member thực hiện thanh toán (Pay in).  
  + Payment Gateway trả kết quả về My Meetings (Return info about payment).
* My Meetings ↔ Payment Gateway  
  + Uỷ quyền việc xử lý thanh toán (Delegate the payment).  
  + Payment Gateway xác nhận thông tin thanh toán về lại hệ thống.
* My Meetings ↔ Email Service  
  + Gửi yêu cầu dịch vụ email (Request email send).  
  + Email Service gửi mail đến Member và Administrator.
* Administrator ↔ My Meetings  
  + Quản lý thành viên, nhóm họp, và cuộc họp.

**Ý nghĩa:**

* Sơ đồ cho thấy My Meetings là trung tâm của hệ thống.
* Các Member tương tác chủ yếu để tham gia và tổ chức cuộc họp, đồng thời thanh toán phí.
* Administrator chịu trách nhiệm quản trị toàn bộ.
* Payment Gateway và Email Service là dịch vụ ngoài tích hợp để hỗ trợ (thanh toán và gửi thông báo).

## **Bài 5: Vẽ sơ đồ C2 – Container**

****

**Các thành phần chính trong hệ thống**

* Người dùng (Person):  
  + Member: Người tổ chức nhóm họp, tham gia cuộc họp, là thành viên nhóm.  
  + Administrator: Quản trị viên của hệ thống.
* Containers trong My Meetings System:  
  + SPA (Single Page Application) – [ReactJS]  
  Đây là giao diện người dùng (GUI) của ứng dụng.  
  Người dùng (Member, Admin) truy cập vào SPA qua trình duyệt, giao tiếp với hệ thống qua HTTP.  
  + My Meeting API – [ASP.NET Core]  
  Backend, xử lý logic nghiệp vụ chính của hệ thống.

SPA gọi đến API qua HTTP.  
API giao tiếp với cơ sở dữ liệu, dịch vụ email và cổng thanh toán.  
+ Database – [Microsoft SQL]  
Lưu trữ dữ liệu về nhóm họp, thành viên, và các cuộc họp.  
API đọc/ghi dữ liệu bằng SQL.

**Các hệ thống bên ngoài:**

- Email System (SMTP Server bên thứ 3): dùng để gửi email thông báo (API gửi yêu cầu gửi mail qua SMTP).  
- Payment Gateway (Dịch vụ thanh toán bên thứ 3): xử lý thanh toán của thành viên (API gửi yêu cầu qua HTTP, Payment Gateway phản hồi kết quả).

**Các luồng tương tác:**

* Member / Admin → SPA (HTTP):  
   Người dùng thao tác trên giao diện web.
* SPA → My Meeting API (HTTP):  
   Gửi request đến backend để xử lý (ví dụ: tạo cuộc họp, tham gia nhóm).
* My Meeting API → Database (SQL):  
   Đọc/ghi dữ liệu (thông tin thành viên, cuộc họp).
* My Meeting API → Email System (SMTP):  
   Gửi email thông báo đến người dùng.
* My Meeting API → Payment Gateway (HTTP):  
   Thực hiện giao dịch thanh toán.

**Ý nghĩa:**

* Thể hiện cấu trúc 3 lớp chính của hệ thống:  
  + Frontend (SPA – ReactJS)  
  + Backend (API – .NET Core)  
  + Database (SQL Server)
* Các dịch vụ bên ngoài (Email, Payment Gateway) được tích hợp qua API.
* Người dùng chỉ tiếp xúc với SPA, còn tất cả nghiệp vụ và lưu trữ được xử lý ở API + Database.

## **Bài 6: Vẽ sơ đồ C3 – Component (High-Level)**

****

**Các thành phần chính**

* My Meetings System:
* SPA (ReactJS): Giao diện người dùng (GUI).
* Người dùng sẽ truy cập SPA → gửi/nhận dữ liệu qua HTTP tới API.
* My Meeting API:
* Đây là API viết bằng .NET Core, là container trung tâm.
* API gọi tới nhiều component (module) khác nhau để xử lý logic nghiệp vụ:
* Registrations [.Net Libraries]: Quản lý đăng ký.
* User Access [.Net Libraries]: Quản lý quyền truy cập.
* Meetings [.Net Libraries]: Quản lý thông tin cuộc họp.
* Administration [.Net Libraries]: Quản lý hệ thống.
* Payments [.Net Libraries]: Xử lý thanh toán.
* Database:
* Mỗi module đều có schema riêng để lưu dữ liệu:
* Registrations data
* User Access data
* Meetings data
* Administration data
* Payments data
* Các component store/retrieve dữ liệu từ schema tương ứng.
* Events Bus (In memory):
* Đây là cơ chế Event-driven trong hệ thống.
* Các module (Registrations, Meetings, Administration, Payments) có thể publish sự kiện hoặc subscribe sự kiện qua Event Bus → Giúp các module giao tiếp với nhau mà không phụ thuộc trực tiếp.

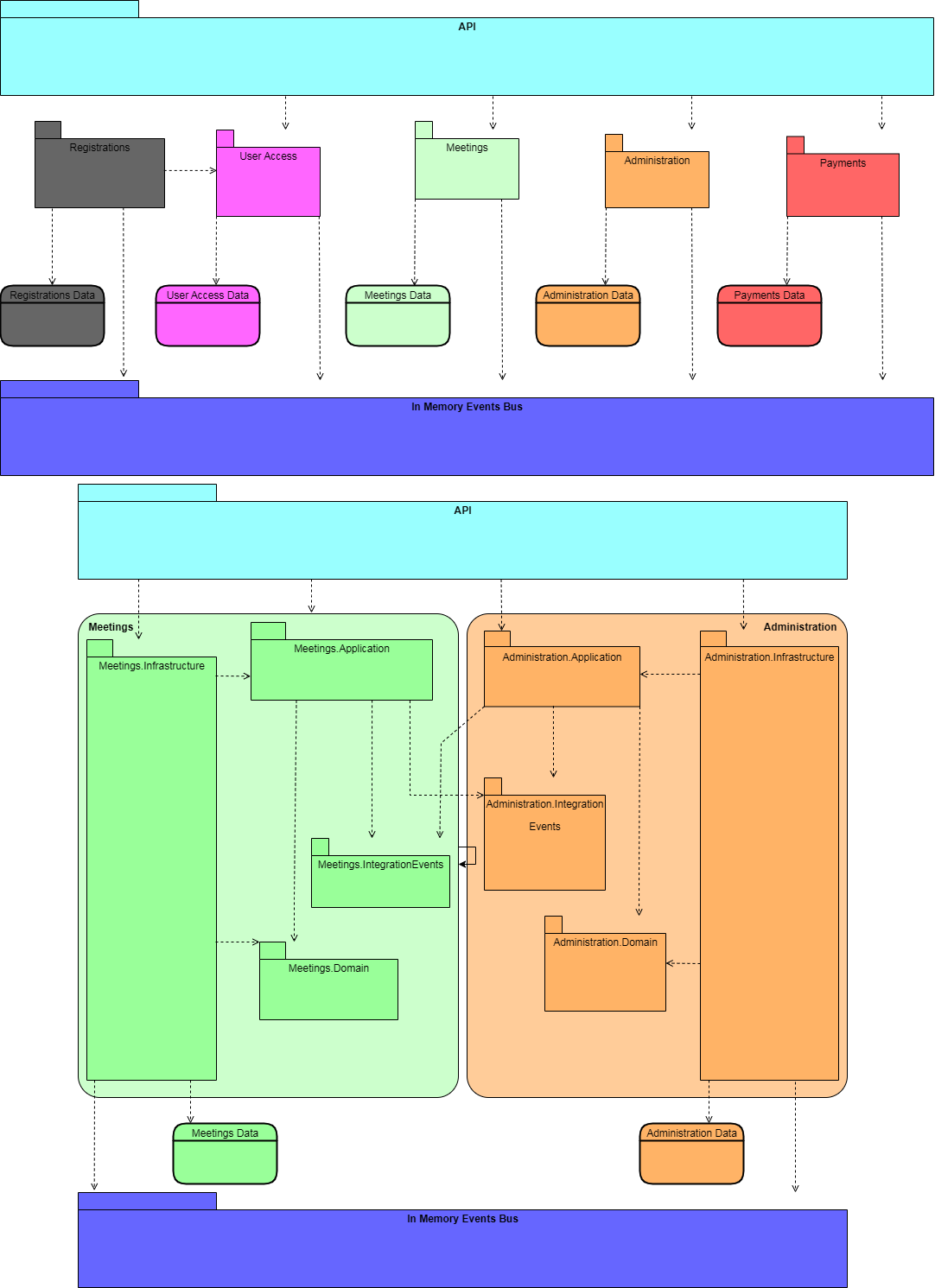
**Cách các thành phần liên kết**

* Người dùng → truy cập SPA (ReactJS).
* SPA gọi API qua HTTP.
* API phân luồng request đến từng component tương ứng.
* Component xử lý:
* Lấy hoặc lưu dữ liệu trong Database.
* Phát sinh sự kiện → gửi vào Event Bus.
* Nhận sự kiện từ Event Bus để thực hiện hành động khác.

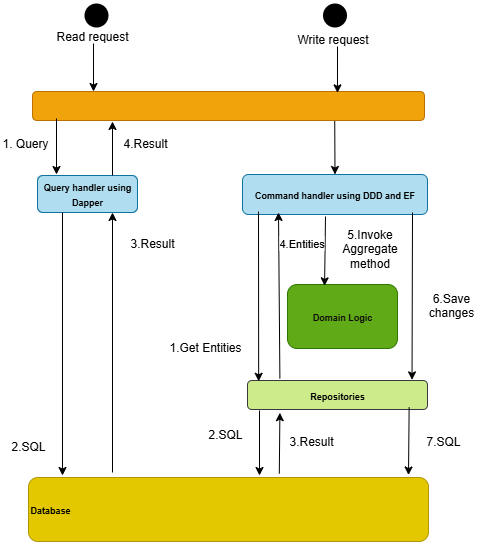
**Ý nghĩa**

* Phân tầng rõ ràng: UI - API - Business logic - Database.
* Tách biệt module: mỗi phần (Registration, Payments,..) độc lập, dễ mở rộng.
* Hỗ trợ event-driven: hệ thống linh hoạt, giảm phụ thuộc trực tiếp giữa các module.
* Dễ bảo trì, mở rộng khi phát triển thêm tính năng.

## **Bài 7. Vẽ sơ đồ C3 – Component (Module-Level)**

****

## **Bài 8: Vẽ sơ đồ xử lý 1 request:**

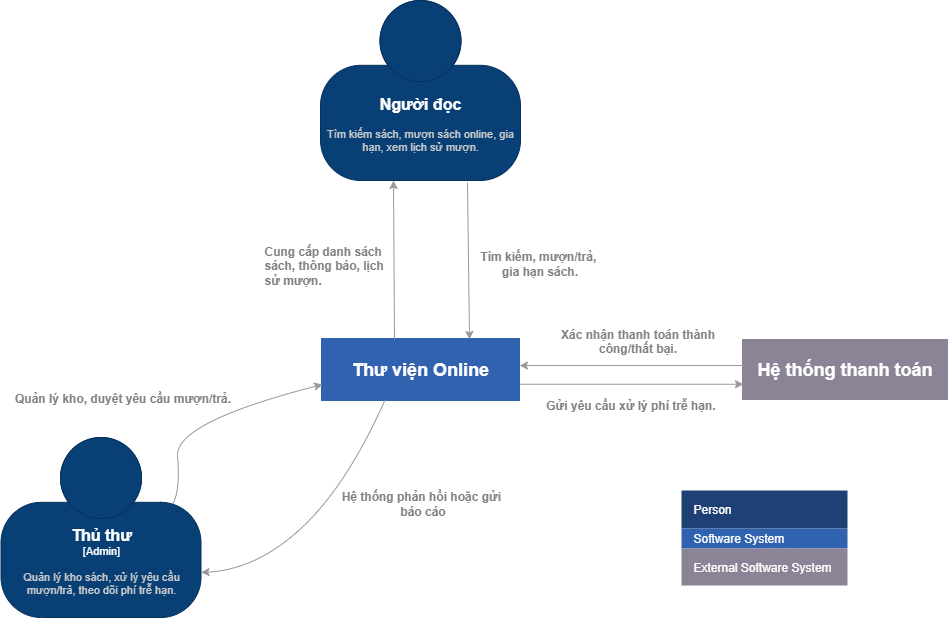
****

# **BÀI TẬP ỨNG DỤNG:**

## **Thiết kế kiến trúc phần mềm theo chuẩn C4 model:**

### **Level 1: (Context Diagram):**

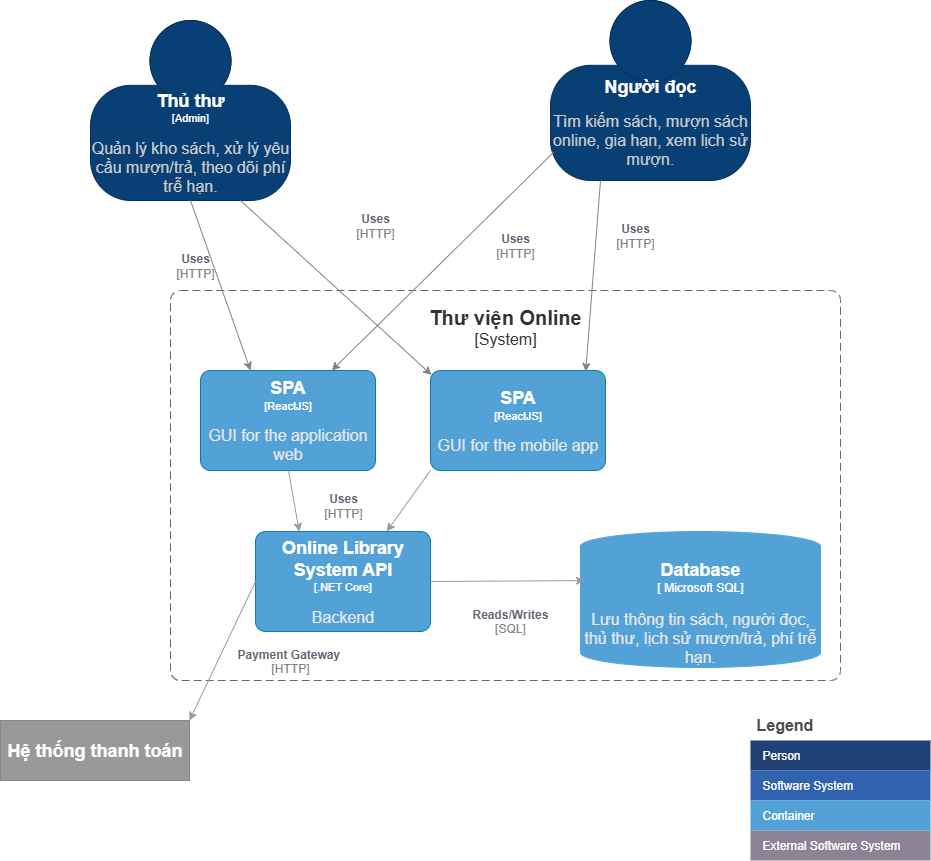
Hệ thống Thư viện Online cho phép người đọc tìm kiếm, mượn/trả sách và theo dõi lịch sử mượn. Thủ thư/Admin sử dụng hệ thống để quản lý kho sách, xử lý yêu cầu mượn/trả, và theo dõi phí trễ hạn. Ngoài ra, hệ thống còn tích hợp với dịch vụ thanh toán để tự động xử lý phí thành viên và phí trễ.  
Quyết định thiết kế chính: kết nối với hệ thống thanh toán bên ngoài thay vì xử lý thủ công nhằm đảm bảo tự động hóa, chính xác và giảm phụ thuộc vào thủ thư.

****

### **Level 2 (Container Diagram):**

Hệ thống gồm các container chính: Web App (SPA) cho người dùng máy tính, Mobile App cho người dùng di động, Database để lưu trữ dữ liệu người dùng và sách, và Payment Service để xử lý giao dịch.

Quyết định thiết kế chính: tách riêng Web App và Mobile App nhằm tối ưu trải nghiệm đa nền tảng; chọn SPA cho Web App để tăng tốc độ và giảm tải server; tách Payment Service riêng giúp dễ mở rộng hoặc thay đổi cổng thanh toán.

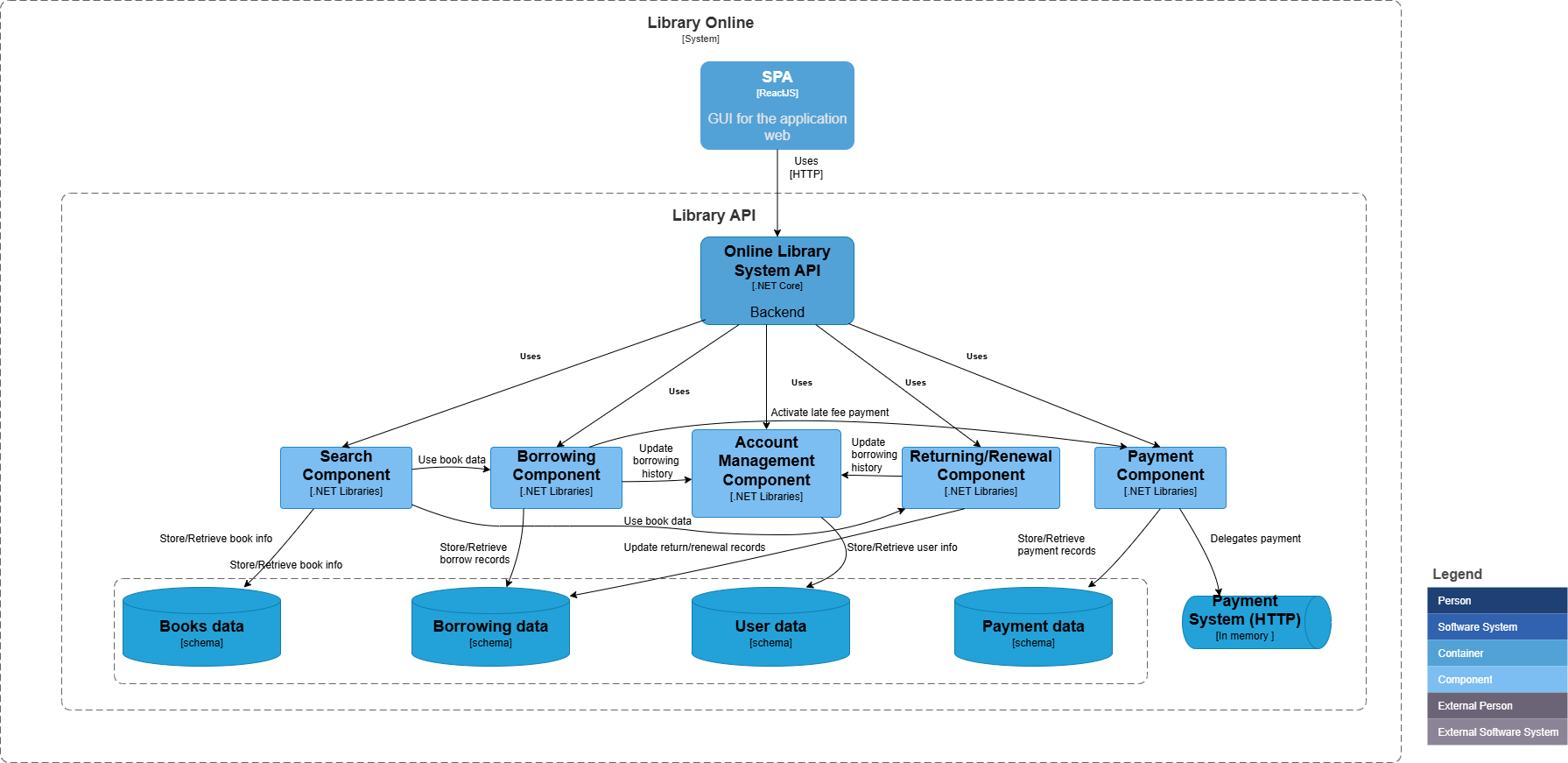
****

### **Level 3 (Component Diagram):**

Web App được chia thành các thành phần chính:

* Search Component: tìm kiếm và gợi ý sách nhanh.
* Borrowing & Returning/Renewal Component: xử lý mượn, trả, gia hạn và phí trễ hạn.
* Account Management Component: quản lý tài khoản, phân quyền.
* Payment Component: xử lý và lưu trữ thanh toán, tích hợp hệ thống ngoài.

Quyết định thiết kế chính: tách thành component độc lập để dễ bảo trì, mở rộng, tái sử dụng (Web/Mobile). Giao diện dùng SPA (ReactJS) giúp tải nhanh, trải nghiệm mượt mà và hỗ trợ mở rộng đa nền tảng.

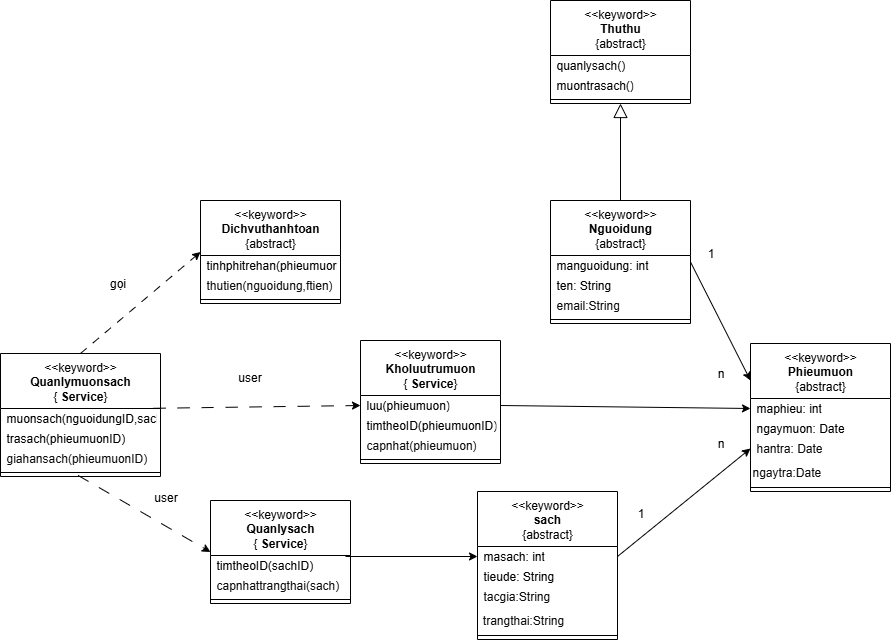


### **Level 4 (Code/Implementation Diagram):**

Module Quản lý mượn sách có các lớp chính:

Người dùng và sách: biểu diễn người đọc và sách.

Kho lưu trữ mượn: lưu thông tin mượn/trả.  
Quản lý mượn trả : xử lý nghiệp vụ (mượn, trả, gia hạn, tính phí trễ).  
Phiếu mượn: thao tác với dữ liệu.  
Dịch vụ thanh toán : hỗ trợ thanh toán phí trễ.  
Quyết định thiết kế chính: sử dụng kiến trúc phân lớp (Entity – Service – Repository) để tách biệt logic nghiệp vụ và dữ liệu; tách dịch vụ thanh toán riêng để linh hoạt khi thay đổi cổng thanh toán.

****