|  |
| --- |
| BỘ GIÁO DỤC VÀ ĐÀO TẠO  **ĐẠI HỌC PHENIKAA**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_    **BÁO CÁO TỔNG KẾT**  **MÔN:PHÂN TÍCH VÀ THIẾT KẾ PHẦN MỀM**  **NHÓM:05**    **TÊN ĐỀ TÀI:**  **BookHaven: Hệ thống quản lý thư viện với chữ ký số**  **Lớp học phần: Phân tích và thiết kế phần mềm-1-1-25(N05)**      **Giảng viên hướng dẫn: ThS.Vũ Quang Dũng**  **ThS.Nguyễn Xuân Quế**  **Sinh viên thực hiện: Nguyễn Thọ Nhân -23010786**  **Nguyễn Xuân Chức -23010452**  **Phạm Anh Thái -23010784**  **Lê Tuấn Anh -21011577**  **Hà Nội, tháng 09 năm 2025** |

***Mục Lục***

**LỜI NÓI ĐẦU..............................................................................................................................5**  
**DANH MỤC HÌNH ẢNH...............................................................................................................8**  
**DANH MỤC BẢNG BIỂU..............................................................................................................9**  
**DANH MỤC CÁC TỪ VIẾT TẮT.....................................................................................................9**

### **CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI**

* **1.1. Đặt Vấn Đề và Lý Do Chọn Đề Tài............................................................................ 10**
  + 1.1.1. Bối cảnh của thư viện trong kỷ nguyên số.................................................... 10
  + 1.1.2. Các vấn đề còn tồn tại của hệ thống truyền thống ...................................... 10
  + 1.1.3. Sự cần thiết của việc áp dụng công nghệ mới .............................................. 10
* **1.2. Mục Tiêu của Dự Án................................................................................................ 11**
  + 1.2.1. Mục tiêu về chức năng ................................................................................. 11
  + 1.2.2. Mục tiêu về kỹ thuật .....................................................................................11
  + 1.2.3. Mục tiêu về trải nghiệm người dùng ............................................................11
* **1.3. Đối Tượng và Phạm Vi Nghiên Cứu..........................................................................11**
  + 1.3.1. Đối tượng sử dụng.........................................................................................11
  + 1.3.2. Phạm vi chức năng.........................................................................................11
  + 1.3.3. Phạm vi công nghệ.........................................................................................12

### **CHƯƠNG 2: CƠ SỞ LÝ THUYẾT VÀ PHÂN TÍCH YÊU CẦU**

* **2.1. Cơ Sở Lý Thuyết.......................................................................................................13**
  + 2.1.1. Tổng quan về Framework Laravel 10 và kiến trúc MVC................................ 13
  + 2.1.2. Lý thuyết về Chữ ký số (Digital Signature)..................................................... 13
    - a. Mật mã hóa bất đối xứng (Asymmetric Cryptography)......................... 13
    - b. Hàm băm (Hash Function)......................................................................13
    - c. Quy trình Ký và Xác thực với thuật toán RSA-SHA256............................14
  + 2.1.3. Lý thuyết về Gamification và ứng dụng..........................................................14
  + 2.1.4. Lý thuyết về Progressive Web App (PWA) và Service Workers......................14
* **2.2. Phân Tích Yêu Cầu Hệ Thống.....................................................................................15**
  + 2.2.1. Yêu cầu chức năng (Functional Requirements)..............................................15
    - a. Chức năng dành cho Quản trị viên (Admin)............................................15
    - b. Chức năng dành cho Người dùng (User).................................................16
  + 2.2.2. Yêu cầu phi chức năng (Non-functional Requirements).................................17
    - a. Yêu cầu về Bảo mật (Security).................................................................17
    - b. Yêu cầu về Hiệu năng (Performance)......................................................17
    - c. Yêu cầu về Giao diện và Trải nghiệm (UI/UX)..........................................17
    - d. Yêu cầu về Tính Tương thích (Compatibility)..........................................17

### **CHƯƠNG 3: THIẾT KẾ HỆ THỐNG**

* **3.1. Thiết Kế Kiến Trúc Tổng Thể.....................................................................................17**
  + 3.1.1. Mô hình kiến trúc 3 lớp
  + 3.1.2. Sơ đồ kiến trúc hệ thống BookHaven
* **3.2. Thiết Kế Chức Năng (Biểu đồ Usecase).....................................................................18**
  + 3.2.1. Xác định các Tác nhân (Actors)......................................................................18
  + 3.2.2. Biểu đồ Usecase tổng quát............................................................................18
  + 3.2.3. Đặc tả chi tiết một vài Usecase quan trọng ..................................................18
* **3.3. Đặc Tả Chi Tiết Usecase và Quy Tắc Nghiệp Vụ (Business Rules)...............................19**
* **3.4. Thiết Kế Cơ Sở Dữ Liệu............................................................................................21**
  + 3.3.1. Biểu đồ Lớp (Class Diagram)..........................................................................22
  + 3.3.2. Sơ đồ quan hệ thực thể
  + 3.3.3. Mô tả chi tiết các bảng dữ liệu
* **3.5. Thiết Kế Luồng Hoạt Động.......................................................................................23**
  + 3.4.1. Biểu đồ Hoạt động (Activity Diagram) cho chức năng Mượn và Xác thực sách online......................................................................................................................23
  + 3.4.2. Biểu đồ Trạng thái (State Diagram) cho vòng đời của Lượt mượn (Loan) và Sách (Book)......................................................................................................................27

### **CHƯƠNG 4: TRIỂN KHAI XÂY DỰNG HỆ THỐNG**

* **4.1. Môi Trường và Công Nghệ Sử Dụng..........................................................................29**
  + 4.1.1. Phần cứng và Phần mềm................................................................................29
  + 4.1.2. Bảng kê chi tiết các Công nghệ, Framework, và Thư viện..............................29
* **4.2. Cấu Trúc Thư Mục Dự Án..........................................................................................30**
  + (Trình bày cấu trúc thư mục của dự án Laravel)
* **4.3. Triển Khai Các Chức Năng Nổi Bật.............................................................................31**
  + 4.3.1. Xây dựng module Chữ ký số RSA-SHA256......................................................31
    - a. Logic tạo cặp khóa Public/Private Key....................................................31
    - b. Code snippet cho chức năng Ký số khi mượn.........................................31
    - c. Code snippet cho chức năng Xác thực khi trả.........................................31
  + 4.3.2. Xây dựng module Gamification......................................................................31
    - a. Logic tính điểm kinh nghiệm (XP) và lên cấp
    - b. Sử dụng Events & Listeners để trao huy hiệu
  + 4.3.3. Xây dựng module PWA và Đọc Offline.........................................................32
    - a. Cấu hình Service Worker và Manifest file.............................................32
    - b. Logic tải và lưu trữ sách trên trình duyệt..............................................32
  + 4.3.4. Xây dựng module Quản lý và Tương tác Cộng đồng
* **4.5. Áp dụng các Mẫu thiết kế (Design Patter).................................................................38**
* **4.5. Giao diện người dùng (UI).......................................................................................40**

### **CHƯƠNG 5: KIỂM THỬ VÀ ĐÁNH GIÁ**

* **5.1. Kế Hoạch và Kịch Bản Kiểm Thử (Test Plan & Test Cases)**
  + 5.1.1. Kiểm thử chức năng Chữ ký số
  + 5.1.2. Kiểm thử chức năng Gamification
  + 5.1.3. Kiểm thử các chức năng CRUD
  + 5.1.4. Kiểm thử tính năng PWA & Offline
* **5.2. Kết Quả Kiểm Thử**
* **5.3. Đánh Giá Dự Án**
  + 5.3.1. Các kết quả đã đạt được (so với mục tiêu đề ra)
  + 5.3.2. Ưu điểm của hệ thống
  + 5.3.3. Hạn chế và nhược điểm

### **CHƯƠG 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

* **6.1. Kết Luận**
* **6.2. Hướng Phát Triển Trong Tương Lai**
  + 6.2.1. Tích hợp AI gợi ý sách
  + 6.2.2. Xây dựng API cho ứng dụng di động
  + 6.2.3. Hỗ trợ đa ngôn ngữ
  + 6.2.4. Mở rộng tính năng cho doanh nghiệp (quản lý đa chi nhánh)

**TÀI LIỆU THAM KHẢO**  
**PHỤ LỤC**

* Phụ lục A: Hướng dẫn cài đặt và sử dụng
* Phụ lục B: Mã nguồn một số chức năng quan trọng

### **LỜI NÓI ĐẦU**

Trong bối cảnh của cuộc Cách mạng Công nghiệp 4.0, quá trình chuyển đổi số đã và đang tác động sâu sắc đến mọi lĩnh vực của đời sống xã hội, và ngành thư viện cũng không nằm ngoài xu hướng đó. Các thư viện số ra đời không chỉ giải quyết bài toán về không gian lưu trữ và khả năng tiếp cận, mà còn mở ra những cơ hội mới để nâng cao trải nghiệm của người đọc. Tuy nhiên, đi cùng với những cơ hội là các thách thức không nhỏ, đặc biệt là vấn đề về bảo mật, xác thực tính toàn vẹn của tài liệu điện tử và làm thế nào để duy trì sự tương tác, tạo động lực cho người dùng trong một môi trường số.

Nhận thức được những vấn đề cấp thiết đó, đề tài **“Xây dựng Hệ thống Quản lý Thư viện số Hiện đại BookHaven tích hợp Chữ ký số và Gamification”** đã được hình thành. Dự án này không chỉ hướng tới việc xây dựng một nền tảng quản lý thư viện thông thường, mà còn tập trung vào việc giải quyết hai bài toán cốt lõi: đảm bảo an toàn tuyệt đối cho tài liệu số thông qua chữ ký số và cách mạng hóa trải nghiệm đọc sách bằng các cơ chế trò chơi hóa (Gamification) và tương tác cộng đồng.

Báo cáo này sẽ trình bày một cách chi tiết và hệ thống toàn bộ quá trình nghiên cứu, phân tích, thiết kế, triển khai và đánh giá hệ thống BookHaven. Nội dung báo cáo là kết quả của quá trình học tập, tìm tòi và nỗ lực vận dụng các kiến thức đã học vào một sản phẩm thực tế.

Để hoàn thành được đề tài này, em xin chân thành cảm ơn sự hướng dẫn tận tình của giảng viên [Tên Giảng viên Hướng dẫn]. Thầy/Cô đã luôn chỉ bảo, định hướng và tạo điều kiện tốt nhất để em có thể hoàn thành dự án một cách trọn vẹn.

Mặc dù đã rất cố gắng, song do kiến thức và kinh nghiệm còn hạn chế, báo cáo chắc chắn không thể tránh khỏi những thiếu sót. Em rất mong nhận được những ý kiến đóng góp quý báu từ quý Thầy/Cô để đề tài được hoàn thiện hơn.

Nhóm chúng em xin chân thành cảm ơn!

*Hà Nội, ngày 24 tháng 09 năm 2025*

### **Bảng phân chia công việc chi tiết của các thành viên trong nhóm theo từng tuần**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tuần** | **Nguyễn Thọ Nhân (Leader, Backend)** | **Nguyễn Xuân Chúc (Frontend, UI/UX)** | **Phạm Anh Thái (BA, Tester)** | **Lê Tuấn Anh (DB, DevOps)** |
| **1** | - Chủ trì và thống nhất phạm vi, mục tiêu dự án.  - Phân công nhiệm vụ, thiết lập kế hoạch tổng thể.  - Thiết lập cấu trúc dự án Laravel 10. | - Phác thảo wireframe cho các màn hình chính (Trang chủ, chi tiết sách, đăng nhập). | - Phân tích và viết tài liệu đặc tả yêu cầu hệ thống (SRS).  - Xác định Actor và danh sách Use Case ban đầu. | - Thiết kế lược đồ CSDL version 1.0 (ERD).  - Thiết lập và quản lý repository trên Git. |
| **2** | - Lập trình module Xác thực người dùng (Đăng ký, đăng nhập).  - Giám sát và review tiến độ chung của nhóm. | - Thiết kế mockup (UI design) chi tiết dựa trên wireframe.  - Xây dựng layout chính của website bằng Blade và Tailwind CSS. | - Vẽ biểu đồ Use Case tổng quát.  - Viết kịch bản chi tiết (scenario) cho 2 Use Case quan trọng: "Mượn sách" và "Đăng ký tài khoản". | - Viết script migrations để tạo cấu trúc bảng CSDL.  - Viết script seeders để tạo dữ liệu mẫu ban đầu. |
| **3** | - Lập trình API CRUD cho các module cốt lõi: Sách (Book), Danh mục (Category).  - Nghiên cứu thư viện và giải pháp kỹ thuật cho Chữ ký số. | - Lập trình giao diện tĩnh cho các trang: Danh sách sách, Chi tiết sách, Đăng nhập, Đăng ký. | - Thiết kế Biểu đồ Lớp (Class Diagram) dựa trên kịch bản Use Case và CSDL.  - Viết các kịch bản kiểm thử (Test Cases) cho chức năng Xác thực. | - Hoàn thiện và tối ưu hóa CSDL (khóa ngoại, chỉ mục).  - Hỗ trợ Backend trong việc tích hợp Model với CSDL. |
| **4.** | - Tích hợp API để hiển thị danh sách sách và chi tiết sách.  - Xây dựng giao diện trang Lịch sử mượn sách. | - Vẽ Biểu đồ Trình tự (Sequence Diagram) cho chức năng "Mượn sách và Ký số".  - Bắt đầu kiểm thử các chức năng CRUD Sách. | - Cấu hình môi trường và kết nối CSDL lên Aiven Cloud.  - Triển khai migrations và seeders lên môi trường dev cloud. | Lập trình module Chữ ký số (phần Ký):  - Xây dựng Service xử lý ký tài liệu bằng Private Key khi người dùng mượn sách. |
| **5** | - Lập trình module Chữ ký số (phần Xác thực): <br> - Xây dựng Service xử lý xác thực chữ ký bằng Public Key khi trả sách. | - Tích hợp API Xác thực vào form đăng nhập, đăng ký. <br> - Hiển thị trạng thái mượn/trả trên giao diện người dùng. | - Vẽ Biểu đồ Trạng thái (State Diagram) cho vòng đời của đối tượng Lượt mượn (Loan).  - Viết test cases chi tiết cho luồng Chữ ký số. | - Giám sát và đảm bảo CSDL trên cloud hoạt động ổn định.  - Bắt đầu xây dựng kịch bản (script) cho việc triển khai tự động. |
| **6** | - Thiết kế và lập trình kiến trúc hướng sự kiện cho Gamification (Events & Listeners).  - Lập trình logic tính điểm XP và trao huy hiệu. | - Lập trình giao diện trang Profile cá nhân để hiển thị XP, Level, và các Huy hiệu đạt được. | - Kiểm thử chuyên sâu chức năng Chữ ký số.  - Viết test cases cho toàn bộ hệ thống Gamification. | - Tối ưu hóa các câu truy vấn CSDL phức tạp (query optimization).  - Thiết lập cơ chế sao lưu (backup) cho CSDL. |
| **7** | - Lập trình API cho các tính năng Cộng đồng (Review sách, Thảo luận).  - Tối ưu hóa và review code backend. | - Lập trình giao diện cho chức năng Review, Đánh giá sách và hệ thống Thảo luận. | - Thực hiện kiểm thử các chức năng Gamification.  - Bắt đầu thực hiện kiểm thử tích hợp giữa Frontend và Backend. | - Hoàn thiện các kịch bản triển khai (deployment scripts).  - Nghiên cứu cấu hình server cho PWA. |
| **8** | - Lập trình các module phụ trợ: Hệ thống Thông báo, API tạo mã QR.  - Hoàn thiện API cho trang quản trị của Admin. | - Triển khai các tính năng của PWA (Service Worker, Manifest).  - Xây dựng giao diện cho chức năng quét mã QR. | - Kiểm thử các tính năng Cộng đồng và Thông báo.  - Báo cáo tổng hợp các lỗi tìm được trong quá trình tích hợp. | - Triển khai phiên bản thử nghiệm đầu tiên lên môi trường Staging.  - Cấu hình web server (Nginx/Apache). |
| **9** | - Hỗ trợ fix các bug nghiêm trọng. <br> - Xây dựng API cho Dashboard thống kê của Admin. <br> - Bắt đầu viết báo cáo tổng kết phần kỹ thuật. | - Hoàn thiện toàn bộ giao diện trang Admin (Dashboard, các bảng quản lý). <br> - Rà soát và tối ưu hóa UI/UX toàn bộ hệ thống. | - Thực hiện kiểm thử toàn diện hệ thống (end-to-end testing) trên môi trường Staging. <br> - Viết tài liệu hướng dẫn sử dụng. | - Theo dõi, giám sát hệ thống trên môi trường Staging. <br> - Tối ưu hóa hiệu năng server. |
| **10** | - Chủ trì tổng kết dự án, đánh giá kết quả.  - Hoàn thiện báo cáo cuối kỳ. <br> - Ra quyết định và chỉ huy triển khai phiên bản chính thức. | - Hỗ trợ chỉnh sửa giao diện theo phản hồi cuối cùng.  - Đóng góp nội dung và hình ảnh cho báo cáo cuối kỳ. | - Thực hiện Kiểm thử Chấp nhận Người dùng (UAT).  - Hoàn thiện tài liệu kiểm thử và kết quả. | - Thực hiện triển khai (deploy) phiên bản chính thức lên Production.  - Bàn giao tài liệu kỹ thuật và hạ tầng. |

### **DANH MỤC HÌNH ẢNH**

Phần này liệt kê tất cả các sơ đồ, biểu đồ và hình ảnh chụp màn hình được sử dụng trong báo cáo, giúp người đọc dễ dàng tra cứu và tham chiếu.

* **Hình 3.1:** Sơ đồ Usecase tổng quát của hệ thống BookHaven
* **Hình 3.2:** Sơ đồ Lớp (Class Diagram) mô tả cấu trúc dữ liệu
* **Hình 3.3:** Sơ đồ Hoạt động quy trình Chữ Ký Số và Xác thực Tính Toàn Vẹn
* **Hình 3.4:** Sơ đồ Hoạt động quy trình Tự động trao Huy hiệu
* **Hình 3.5:** Sơ đồ Trạng thái vòng đời của đối tượng Lượt mượn (Loan)
* **Hình 4.1:** Giao diện Trang Chủ & Slider Sách Phổ Biến
* **Hình 4.2:** Giao diện Trang Chi Tiết Sách & Nút Tương Tác
* **Hình 4.3:** Giao diện Trang Cộng Đồng & Thảo Luận
* **Hình 4.4:** Giao diện Thành Tích & Bảng Xếp Hạng
* **Hình 4.5:** Giao diện Quản lý Lịch sử Mượn/Trả
* **Hình 4.6:** Giao diện Dashboard Quản Trị của Admin

### **DANH MỤC BẢNG BIỂU**

Phần này liệt kê tất cả các bảng biểu được sử dụng để hệ thống hóa thông tin, dữ liệu trong báo cáo.

* **Bảng 1.1:** Bảng mô tả phạm vi chức năng chính của dự án BookHaven
* **Bảng 2.1:** Bảng đặc tả các yêu cầu chức năng chính
* **Bảng 3.1:** Đặc tả quy tắc nghiệp vụ cho Usecase "Mượn Sách Online và Ký Số"
* **Bảng 3.2:** Đặc tả quy tắc nghiệp vụ cho Usecase "Trả Sách và Xác thực Chữ ký số"
* **Bảng 4.1:** Bảng kê chi tiết các công nghệ được sử dụng trong dự án
* **Bảng 5.1:** Bảng mô tả các kịch bản kiểm thử (Test Cases) cho chức năng Chữ ký số
* **Bảng 5.2:** Bảng mô tả các kịch bản kiểm thử (Test Cases) cho chức năng Gamification

### **Danh Mục Các Từ Viết Tắt:**

|  |  |  |
| --- | --- | --- |
| **Từ viết tắt** | **Tiếng Anh** | **Tiếng Việt** |
| **UML** | Unified Modeling Language | Ngôn ngữ Mô hình hóa Hợp nhất |
| **MVC** | Model-View-Controller | Mô hình - View - Controller |
| **ORM** | Object-Relational Mapping | Ánh xạ Đối tượng - Quan hệ |
| **CRUD** | Create, Read, Update, Delete | Tạo, Đọc, Cập nhật, Xóa |
| **PWA** | Progressive Web App | Ứng dụng Web Tiến bộ |
| **API** | Application Programming Interface | Giao diện Lập trình Ứng dụng |
| **RSA** | Rivest–Shamir–Adleman | (Tên riêng của thuật toán) |
| **SHA** | Secure Hash Algorithm | Thuật toán Băm An toàn |
| **BR** | Business Rule | Quy tắc Nghiệp vụ |
| **UC** | Use Case | Ca sử dụng / Tình huống sử dụng |

### **CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI**

#### **1.1. Đặt Vấn Đề và Lý Do Chọn Đề Tài**

Thư viện, từ lâu đã là kho tàng tri thức của nhân loại, đang trải qua một cuộc chuyển mình mạnh mẽ trong kỷ nguyên số. Việc chuyển đổi từ mô hình thư viện vật lý truyền thống sang thư viện số mang lại nhiều lợi ích vượt trội về khả năng truy cập, lưu trữ và chia sẻ tài nguyên. Tuy nhiên, mô hình này cũng làm nảy sinh những vấn đề phức tạp cần được giải quyết:

1. **Thách thức về tính toàn vẹn và xác thực tài liệu:** Trong môi trường số, một tài liệu điện tử có thể dễ dàng bị sao chép và chỉnh sửa mà không để lại dấu vết. Điều này đặt ra câu hỏi lớn: Làm thế nào để đảm bảo rằng nội dung cuốn sách mà người dùng mượn là nguyên bản, không bị thay đổi? Làm thế nào để xác thực được danh tính người mượn một cách không thể chối cãi? Đây là bài toán cốt lõi về **tính toàn vẹn (integrity)**, **tính xác thực (authenticity)**, và **tính chống chối bỏ (non-repudiation)**.
2. **Thách thức về trải nghiệm và động lực người dùng:** Không giống như không gian vật lý, môi trường số có thể khiến người dùng cảm thấy thụ động và thiếu sự kết nối. Việc chỉ đơn thuần cung cấp một kho tài liệu online là chưa đủ để thu hút và giữ chân người đọc. Cần có những cơ chế mới để biến việc đọc sách thành một hành trình thú vị, có tính tương tác và tạo động lực lâu dài.
3. **Thách thức về khả năng truy cập hiện đại:** Người dùng hiện đại mong muốn một trải nghiệm liền mạch, nhanh chóng và tiện lợi trên mọi thiết bị. Các hệ thống cũ thường thiếu đi các tính năng truy cập nhanh như quét mã QR hay khả năng hoạt động khi không có kết nối internet, làm giảm sự tiện dụng.

Xuất phát từ những yêu cầu thực tiễn đó, dự án **BookHaven** được đề xuất như một giải pháp toàn diện, áp dụng các công nghệ tiên tiến để giải quyết triệt để các vấn đề trên. Bằng việc tích hợp **Chữ ký số RSA-SHA256**, **hệ thống Gamification**, và công nghệ **Progressive Web App (PWA)**, đề tài hướng tới việc xây dựng một chuẩn mực mới cho thư viện số: Bảo mật - Tương tác - Hiện đại.

#### **1.2. Mục Tiêu của Dự Án**

Dự án được xây dựng với các mục tiêu cụ thể, rõ ràng như sau:

* **Mục tiêu về Kỹ thuật & Bảo mật:**
  + Thiết kế và triển khai thành công thuật toán chữ ký số RSA-SHA256 để ký và xác thực tài liệu điện tử, đảm bảo tính toàn vẹn và chống chối bỏ.
  + Xây dựng hệ thống trên nền tảng Laravel 10 vững chắc, tuân thủ các tiêu chuẩn bảo mật web hiện đại (chống CSRF, XSS, SQL Injection).
  + Đảm bảo hiệu năng hệ thống, tối ưu hóa truy vấn cơ sở dữ liệu và thời gian phản hồi.
* **Mục tiêu về Chức năng & Quản lý:**
  + Xây dựng đầy đủ các chức năng quản lý CRUD (Create, Read, Update, Delete) cho các tài nguyên cốt lõi của thư viện (Sách, Người dùng, Lượt mượn, Sự kiện...).
  + Phát triển các tính năng truy cập hiện đại như quét mã QR để mượn sách nhanh và PWA cho phép đọc sách offline.
  + Triển khai một hệ thống thông báo tự động, đa kênh (email, in-app) để cập nhật thông tin kịp thời đến người dùng.
* **Mục tiêu về Trải nghiệm Người dùng:**
  + Xây dựng một hệ thống Gamification toàn diện (XP, level, huy hiệu, bảng xếp hạng) nhằm tạo động lực và gia tăng sự gắn kết của người dùng với việc đọc sách.
  + Phát triển các tính năng cộng đồng (nhóm đọc, thảo luận, review sách) để tạo ra một không gian tương tác sôi nổi.
  + Thiết kế giao diện người dùng (UI/UX) hiện đại, thân thiện, và tối ưu trên mọi thiết bị (responsive design).

#### **1.3. Đối Tượng và Phạm Vi Nghiên Cứu**

* **Đối tượng sử dụng:**
  + **Người dùng (User):** Độc giả có nhu cầu tìm kiếm, mượn, đọc và tương tác với các đầu sách và cộng đồng.
  + **Quản trị viên (Admin):** Người có toàn quyền quản lý hệ thống, bao gồm quản lý tài nguyên, người dùng, và các hoạt động của thư viện.
* **Phạm vi chức năng:** Dự án tập trung vào việc xây dựng và hoàn thiện các nhóm chức năng chính được trình bày trong bảng dưới đây.

**Bảng 1.1: Bảng mô tả phạm vi chức năng chính của dự án BookHaven**

|  |  |  |
| --- | --- | --- |
| **Hạng mục** | **Chức năng chính** | **Mô tả chi tiết** |
| **Bảo mật Cốt lõi** | Chữ ký số RSA-SHA256 | Tự động ký số tài liệu online khi mượn và xác thực khi trả để đảm bảo tính toàn vẹn. |
| **Trải nghiệm Người dùng** | Hệ thống Gamification | Tích hợp điểm kinh nghiệm (XP), cấp độ, huy hiệu, thử thách và bảng xếp hạng. |
| **Tương tác Cộng đồng** | Social Features | Bao gồm hệ thống review sách, nhóm đọc, diễn đàn thảo luận, và hồ sơ người dùng. |
| **Truy cập Hiện đại** | PWA & Mã QR | Khả năng cài đặt ứng dụng (PWA), đọc sách offline và quét mã QR để mượn sách vật lý. |
| **Quản lý Thư viện** | Hệ thống Quản trị | Cung cấp đầy đủ các thao tác CRUD cho Sách, Danh mục, Người dùng, Sự kiện, và Lượt mượn. |
| **Hệ thống Phụ trợ** | Thông báo Tự động | Gửi thông báo đa kênh (email, in-app) cho các sự kiện quan trọng (quá hạn, trả thành công...). |

* **Phạm vi công nghệ:**
  + **Backend:** Ngôn ngữ PHP 8.1+, Framework Laravel 10.
  + **Frontend:** Blade Template Engine, Tailwind CSS, Alpine.js.
  + **Cơ sở dữ liệu:** Hệ quản trị CSDL MySQL 8.0+.
  + **Môi trường:** Triển khai trên môi trường server cục bộ (local) và kết nối CSDL tới dịch vụ Cloud (Aiven)

-------------------------------------------------------------------------------------------------------------------------------------------

### **CHƯƠNG 2: CƠ SỞ LÝ THUYẾT VÀ PHÂN TÍCH YÊU CẦU**

Chương này tập trung vào việc xây dựng nền tảng kiến thức vững chắc cho dự án, đồng thời xác định một cách rõ ràng các yêu cầu mà hệ thống BookHaven cần phải đáp ứng. Phần đầu sẽ đi sâu vào các cơ sở lý thuyết về công nghệ được áp dụng, và phần sau sẽ trình bày kết quả của quá trình phân tích yêu cầu hệ thống.

#### **2.1. Cơ Sở Lý Thuyết**

Laravel là một PHP framework mã nguồn mở và miễn phí, được phát triển bởi Taylor Otwell, tuân thủ theo kiến trúc **Model-View-Controller (MVC)**. Laravel 10 là phiên bản mới nhất, mang lại nhiều cải tiến về hiệu năng và trải nghiệm cho nhà phát triển.

* **Model:** Đại diện cho cấu trúc dữ liệu và logic nghiệp vụ. Trong Laravel, các Model thường tương tác với cơ sở dữ liệu thông qua một ORM (Object-Relational Mapping) mạnh mẽ có tên là Eloquent.
* **View:** Chịu trách nhiệm hiển thị dữ liệu cho người dùng. Laravel sử dụng Blade Template Engine, một công cụ tạo mẫu đơn giản nhưng mạnh mẽ, cho phép tách biệt hoàn toàn logic ứng dụng khỏi mã HTML.
* **Controller:** Đóng vai trò trung gian, tiếp nhận các yêu cầu (request) từ người dùng, xử lý chúng bằng cách tương tác với Model, và sau đó trả về kết quả cho View để hiển thị.

Việc áp dụng kiến trúc MVC giúp mã nguồn được tổ chức một cách rõ ràng, dễ dàng bảo trì, mở rộng và phát triển theo nhóm.

Chữ ký số là một cơ chế mật mã học được sử dụng để xác minh tính xác thực, tính toàn vẹn và tính chống chối bỏ của một dữ liệu số. Trong dự án BookHaven, thuật toán **RSA-SHA256** được sử dụng, kết hợp hai thành phần chính:

* **a. Mật mã hóa bất đối xứng (Thuật toán RSA):**  
  Đây là hệ thống mật mã sử dụng một cặp khóa có quan hệ toán học với nhau: một **Khóa Công khai (Public Key)** và một **Khóa Bí mật (Private Key)**.
  + **Private Key:** Được giữ bí mật tuyệt đối bởi chủ sở hữu, dùng để **ký** dữ liệu.
  + **Public Key:** Được chia sẻ công khai cho bất kỳ ai, dùng để **xác thực** chữ ký được tạo bởi Private Key tương ứng.  
    Đặc điểm cốt lõi là thông điệp được mã hóa bằng một khóa chỉ có thể được giải mã bằng khóa còn lại trong cặp.
* **b. Hàm băm an toàn (Thuật toán SHA-256):**  
  Hàm băm là một thuật toán toán học nhận đầu vào là dữ liệu có kích thước bất kỳ và tạo ra một chuỗi đầu ra có kích thước cố định, được gọi là **giá trị băm (hash value)**. SHA-256 (Secure Hash Algorithm 256-bit) có các đặc tính quan trọng:
  + **Tính một chiều:** Không thể suy ngược lại dữ liệu gốc từ giá trị băm.
  + **Tính duy nhất:** Một thay đổi nhỏ nhất trong dữ liệu đầu vào cũng sẽ tạo ra một giá trị băm hoàn toàn khác.
  + **Chống xung đột (Collision Resistance):** Cực kỳ khó để tìm ra hai dữ liệu khác nhau có cùng một giá trị băm.
* **c. Quy trình Ký và Xác thực:**
  1. **Quy trình Ký (Signing):**
     + **Bước 1:** Hệ thống lấy nội dung của tài liệu điện tử cần ký.
     + **Bước 2:** Áp dụng hàm băm SHA-256 lên nội dung tài liệu để tạo ra một giá trị băm duy nhất.
     + **Bước 3:** Sử dụng **Private Key** của hệ thống (hoặc người mượn) để mã hóa giá trị băm này. Kết quả thu được chính là **Chữ ký số**. Chữ ký này sau đó được đính kèm với tài liệu.
  2. **Quy trình Xác thực (Verification):**
     + **Bước 1:** Bên nhận (hệ thống khi người dùng trả sách) tách chữ ký số và tài liệu gốc.
     + **Bước 2:** Sử dụng **Public Key** tương ứng để giải mã chữ ký số, thu được giá trị băm ban đầu (Hash A).
     + **Bước 3:** Áp dụng lại hàm băm SHA-256 lên nội dung tài liệu nhận được để tạo ra một giá trị băm mới (Hash B).
     + **Bước 4:** So sánh Hash A và Hash B. Nếu chúng **trùng khớp**, chữ ký là hợp lệ, đồng nghĩa với việc tài liệu là toàn vẹn và được ký bởi đúng chủ thể. Nếu không, tài liệu đã bị thay đổi hoặc chữ ký là giả mạo.

Gamification là việc ứng dụng các cơ chế, thành phần thiết kế của trò chơi vào các bối cảnh không phải trò chơi nhằm thúc đẩy sự tham gia, tạo động lực và thay đổi hành vi của người dùng. Trong BookHaven, các thành phần sau được áp dụng:

* **Hệ thống Điểm (Points System):** Thưởng điểm kinh nghiệm (XP) cho các hành động tích cực như mượn sách, viết review, giúp người dùng cảm thấy có sự tiến bộ.
* **Huy hiệu (Badges) và Thành tích (Achievements):** Công nhận và vinh danh các cột mốc quan trọng của người dùng, tạo ra cảm giác hoàn thành và sưu tầm.
* **Bảng xếp hạng (Leaderboards):** Khơi dậy tính cạnh tranh lành mạnh, thúc đẩy người dùng hoạt động tích cực hơn để cải thiện thứ hạng.
* **Thử thách (Challenges):** Đưa ra các nhiệm vụ có mục tiêu rõ ràng (ví dụ: "đọc 3 cuốn sách trong tuần") để giữ cho trải nghiệm luôn mới mẻ.

PWA là một loại ứng dụng được phân phối qua web, được xây dựng bằng các công nghệ web phổ biến bao gồm HTML, CSS và JavaScript. Chúng được thiết kế để hoạt động trên mọi nền tảng có trình duyệt web tiêu chuẩn. Các công nghệ cốt lõi bao gồm:

* **Service Workers:** Là một đoạn mã JavaScript chạy ngầm trên trình duyệt, tách biệt với trang web, cho phép triển khai các tính năng như xử lý yêu cầu mạng, lưu trữ cache và push notification. Đây là công nghệ nền tảng cho khả năng hoạt động offline.
* **Web App Manifest:** Là một tệp JSON đơn giản, cung cấp thông tin (như tên, icon, màu sắc chủ đạo) để trình duyệt biết rằng trang web này có thể được "cài đặt" lên màn hình chính của thiết bị.

#### **2.2. Phân Tích Yêu Cầu Hệ Thống**

Quá trình này nhằm xác định các yêu cầu cụ thể mà hệ thống BookHaven phải đáp ứng, được chia thành hai nhóm chính: yêu cầu chức năng và yêu cầu phi chức năng.

Đây là các yêu cầu mô tả những gì hệ thống phải làm, các chức năng cụ thể mà hệ thống cung cấp cho người dùng.

**Bảng 2.1: Bảng đặc tả các yêu cầu chức năng chính**

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã YC** | **Tên chức năng** | **Mô tả chi tiết** | **Đối tượng** |
| **F-ADM-01** | Quản lý Sách (CRUD) | Hệ thống phải cho phép Admin thêm sách mới, xem danh sách, cập nhật thông tin và xóa sách khỏi hệ thống. | Admin |
| **F-ADM-02** | Quản lý Người dùng | Hệ thống phải cho phép Admin xem danh sách người dùng, xem chi tiết và thay đổi trạng thái tài khoản (kích hoạt/vô hiệu hóa). | Admin |
| **F-ADM-03** | Quản lý Mượn/Trả | Hệ thống phải cho phép Admin theo dõi tất cả các lượt mượn, xem trạng thái (quá hạn) và xác nhận các lượt trả sách. | Admin |
| **F-ADM-04** | Quản lý Gamification | Hệ thống phải cho phép Admin tạo và quản lý các loại huy hiệu, cấu hình các thử thách cho người dùng. | Admin |
| **F-USR-01** | Xác thực Người dùng | Hệ thống phải cho phép người dùng đăng ký tài khoản, đăng nhập, đăng xuất và đặt lại mật khẩu. | User |
| **F-USR-02** | Tìm kiếm Sách | Hệ thống phải cung cấp công cụ tìm kiếm và bộ lọc để người dùng có thể tìm sách theo tiêu đề, tác giả, hoặc danh mục. | User |
| **F-USR-03** | Mượn và Ký số Sách | Hệ thống phải cho phép người dùng mượn sách online. Khi mượn, hệ thống tự động ký số lên tài liệu. | User |
| **F-USR-04** | Trả và Xác thực Sách | Hệ thống phải cho phép người dùng trả sách online. Khi trả, hệ thống tự động xác thực chữ ký số để đảm bảo tính toàn vẹn. | User |
| **F-USR-05** | Tương tác Cộng đồng | Hệ thống phải cho phép người dùng viết và xem review sách, tham gia các nhóm đọc và diễn đàn thảo luận. | User |
| **F-USR-06** | Theo dõi Tiến độ | Người dùng phải có thể xem được điểm kinh nghiệm (XP), cấp độ, các huy hiệu đã đạt được và vị trí trên bảng xếp hạng. | User |
| **F-USR-07** | Sử dụng PWA & QR | Người dùng phải có thể cài đặt ứng dụng lên thiết bị, tải sách đọc offline và sử dụng camera để quét mã QR mượn sách. | User |

Đây là các yêu cầu về các đặc tính, ràng buộc của hệ thống, mô tả hệ thống nên *như thế nào* thay vì nó nên *làm gì*.

* **a. Yêu cầu về Bảo mật (Security):**
  + **SEC-01:** Mật khẩu của người dùng phải được băm (hashing) bằng thuật toán an toàn (ví dụ: Bcrypt) trước khi lưu vào cơ sở dữ liệu.
  + **SEC-02:** Hệ thống phải có cơ chế phân quyền chặt chẽ, đảm bảo người dùng thông thường không thể truy cập các chức năng của Admin.
  + **SEC-03:** Hệ thống phải được bảo vệ khỏi các lỗ hổng web phổ biến như Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF), và SQL Injection.
  + **SEC-04:** Cặp khóa Public/Private Key dùng cho chữ ký số phải được lưu trữ và bảo vệ một cách an toàn.
* **b. Yêu cầu về Hiệu năng (Performance):**
  + **PERF-01:** Thời gian tải trang chính (trang chủ, trang danh sách sách) không được vượt quá 3 giây trong điều kiện mạng thông thường.
  + **PERF-02:** Thời gian phản hồi cho các thao tác mượn/trả sách (bao gồm cả ký và xác thực) không được vượt quá 5 giây.
  + **PERF-03:** Hệ thống phải có khả năng xử lý ít nhất 100 người dùng truy cập đồng thời mà không bị suy giảm hiệu năng đáng kể.
* **c. Yêu cầu về Giao diện và Trải nghiệm (UI/UX):**
  + **UI-01:** Giao diện phải được thiết kế theo phong cách hiện đại, sạch sẽ và nhất quán trên toàn bộ ứng dụng.
  + **UI-02:** Hệ thống phải có thiết kế đáp ứng (responsive), hiển thị tốt trên các thiết bị khác nhau như máy tính để bàn, máy tính bảng và điện thoại di động.
  + **UI-03:** Luồng thao tác của người dùng phải trực quan, dễ hiểu và dễ sử dụng, giảm thiểu số lần nhấp chuột không cần thiết.
* **d. Yêu cầu về Tính Tương thích (Compatibility):**
  + **COMP-01:** Hệ thống phải hoạt động ổn định trên các phiên bản mới nhất của các trình duyệt web phổ biến: Google Chrome, Mozilla Firefox, và Microsoft Edge.

-------------------------------------------------------------------------------------------------------------------------------------------

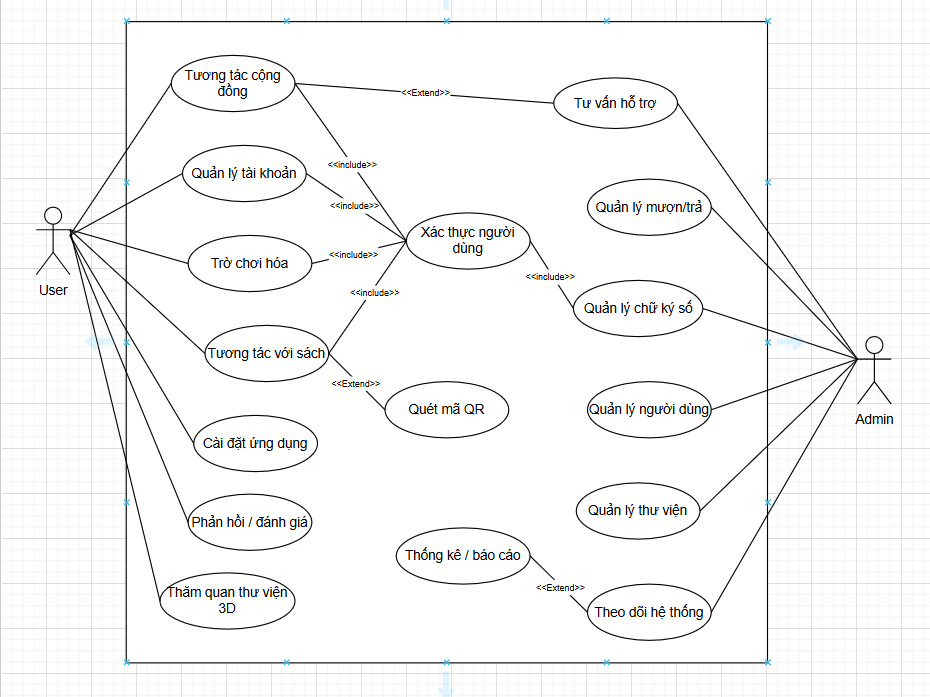
### **CHƯƠNG 3: THIẾT KẾ HỆ THỐNG**

Sau khi đã xác định rõ các yêu cầu ở chương trước, chương này tập trung vào việc chuyển hóa các yêu cầu đó thành một bản thiết kế hệ thống chi tiết. Quá trình thiết kế được thực hiện thông qua việc sử dụng Ngôn ngữ Mô hình hóa Hợp nhất (UML - Unified Modeling Language) để trực quan hóa kiến trúc, cấu trúc dữ liệu và các luồng nghiệp vụ cốt lõi của hệ thống BookHaven.

#### **3.1. Thiết Kế Chức Năng (Biểu đồ Usecase)**

Biểu đồ Usecase được sử dụng để xác định các chức năng chính của hệ thống từ góc nhìn của người dùng. Nó mô tả sự tương tác giữa các tác nhân (Actors) và hệ thống, làm rõ ai có thể làm gì.

* **Xác định các Tác nhân:**
  + **User:** Đại diện cho người đọc, độc giả của thư viện. Họ là những người tương tác chính với các chức năng đọc, mượn, trả và các tính năng cộng đồng.
  + **Admin:** Đại diện cho thủ thư hoặc người quản trị hệ thống. Admin có đầy đủ quyền của một User, đồng thời có thêm các quyền quản lý và giám sát toàn bộ hệ thống.
* **Biểu đồ Usecase tổng quát:**



* **Mô tả Biểu đồ:**
  + Sơ đồ thể hiện rõ hai nhóm chức năng chính. **User** có thể thực hiện các hành động như *Quản lý tài khoản*, *Tương tác với sách*, *Tương tác cộng đồng*, *Trò chơi hóa*, *Quét mã QR*, và *Cài đặt ứng dụng*.
  + **Admin** kế thừa toàn bộ chức năng của User (biểu thị bằng mũi tên kế thừa) và có thêm các chức năng quản trị cấp cao như *Quản lý Thư viện (CRUD)*, *Quản lý Người dùng*, *Quản lý Mượn/Trả*, và *Theo dõi hệ thống*.
  + Các mối quan hệ <<include>> và <<extend>> được sử dụng để làm rõ sự phụ thuộc và các luồng chức năng mở rộng, ví dụ như hầu hết các chức năng chính đều yêu cầu <<include>> chức năng *Xác thực người dùng*.

**3.3. Đặc Tả Chi Tiết Usecase và Quy Tắc Nghiệp Vụ (Business Rules)**

#### **UC-01: Mượn Sách Online và Ký Số**

* **Mục đích:** Đặc tả quy trình người dùng mượn một tài liệu điện tử và hệ thống tự động áp dụng chữ ký số để đảm bảo tính toàn vẹn.
* **Tác nhân:** User
* **Điều kiện tiên quyết:**
  + User đã đăng nhập thành công vào hệ thống.
  + Cuốn sách online phải ở trạng thái "Sẵn có".

**Quy tắc nghiệp vụ (Business Rules):**

|  |  |
| --- | --- |
| **Step** | **Description** |
| **1** | B1. Tại trang Chi tiết Sách, người dùng bấm nút **[Mượn Ngay]**. |
| **2** | B2. Hệ thống hiển thị một hộp thoại (modal/popup) xác nhận.  - **Nội dung:** "Bạn có chắc chắn muốn mượn cuốn sách này? Tài liệu sẽ được ký số để đảm bảo tính toàn vẹn."  - **Các nút:** [Xác nhận], [Hủy bỏ]. |
| **3** | B3. Người dùng bấm nút **[Xác nhận]**. |
| **4** | B4. Hệ thống thực hiện các hành động sau ở backend:  1. **Kiểm tra điều kiện:** Xác minh lại người dùng có đủ điều kiện mượn không (ví dụ: không có sách quá hạn).  2. **Tạo bản ghi Loan:** Tạo một dòng mới trong bảng loans với các thông tin: user\_id, book\_id, loan\_date, due\_date, status = 'đang mượn'.  3. **Thực hiện Ký số:**  - Đọc nội dung gốc của tài liệu.  - Sử dụng **Private Key** của hệ thống và thuật toán RSA-SHA256 để tạo chữ ký số từ nội dung.  - Lưu chữ ký này vào cột digital\_signature của bản ghi Loan vừa tạo.  4. **Cập nhật trạng thái sách:** Cập nhật số lượng sách sẵn có trong bảng books.  5. **Phát ra sự kiện (Event):** BookBorrowed::dispatch($loan). |
| **5** | B5. Hệ thống hiển thị thông báo thành công cho người dùng trên giao diện: "Mượn sách thành công!" và điều hướng người dùng đến trang "Sách của tôi". |

#### **UC-02: Trả Sách và Xác thực Chữ ký số**

* **Mục đích:** Đặc tả quy trình người dùng trả một tài liệu điện tử và hệ thống tự động xác thực chữ ký số để kiểm tra tính toàn vẹn.
* **Tác nhân:** User
* **Điều kiện tiên quyết:**
  + User đã đăng nhập.
  + Người dùng đang ở trong trang "Sách của tôi" và có một lượt mượn đang hoạt động.

**Quy tắc nghiệp vụ (Business Rules):**

|  |  |
| --- | --- |
| **Step** | **Description** |
| **1** | B1. Tại trang "Sách của tôi", bên cạnh cuốn sách đang mượn, người dùng bấm nút **[Trả Sách]**. |
| **2** | B2. Hệ thống thực hiện các hành động sau ở backend:  1. **Lấy dữ liệu:**  - Lấy nội dung **hiện tại** của file tài liệu.  - Truy vấn CSDL để lấy chữ ký số **gốc** đã lưu trong bản ghi Loan tương ứng.  2. **Thực hiện Xác thực:**  - Sử dụng **Public Key** của hệ thống để xác thực nội dung hiện tại với chữ ký gốc.  3. **Xử lý kết quả:**  - **Nếu xác thực thành công (hợp lệ):** Cập nhật trạng thái integrity\_status trong bản ghi Loan thành 'Vẹn Toàn'.  - **Nếu xác thực thất bại (không hợp lệ):** Cập nhật trạng thái integrity\_status thành 'Không Vẹn Toàn'.  4. **Hoàn tất lượt trả:** Cập nhật status của Loan thành 'đã trả', ghi nhận return\_date.  5. **Cập nhật trạng thái sách:** Tăng lại số lượng sách sẵn có trong bảng books.  6. **Phát ra sự kiện (Event):** BookReturned::dispatch($loan). |
| **3** | B3. Hệ thống hiển thị thông báo kết quả cho người dùng: "Trả sách thành công. Kết quả xác thực: **[Vẹn Toàn / Không Vẹn Toàn]**." |
| **4** | B4. **(Luồng phụ - Gamification)** Hệ thống sự kiện của Laravel kích hoạt GamificationSubscriber, tự động cộng điểm kinh nghiệm và kiểm tra điều kiện trao huy hiệu cho người dùng dựa trên sự kiện BookReturned. |

#### **3.4. Thiết Kế Cơ Sở Dữ Liệu (Biểu đồ Lớp)**

Biểu đồ Lớp (Class Diagram) mô tả cấu trúc tĩnh của hệ thống, bao gồm các lớp đối tượng, các thuộc tính (attributes) và các mối quan hệ logic giữa chúng. Đây là bản thiết kế cho cơ sở dữ liệu của ứng dụng.

**a. Biểu đồ lớp ( Class Diagram )**



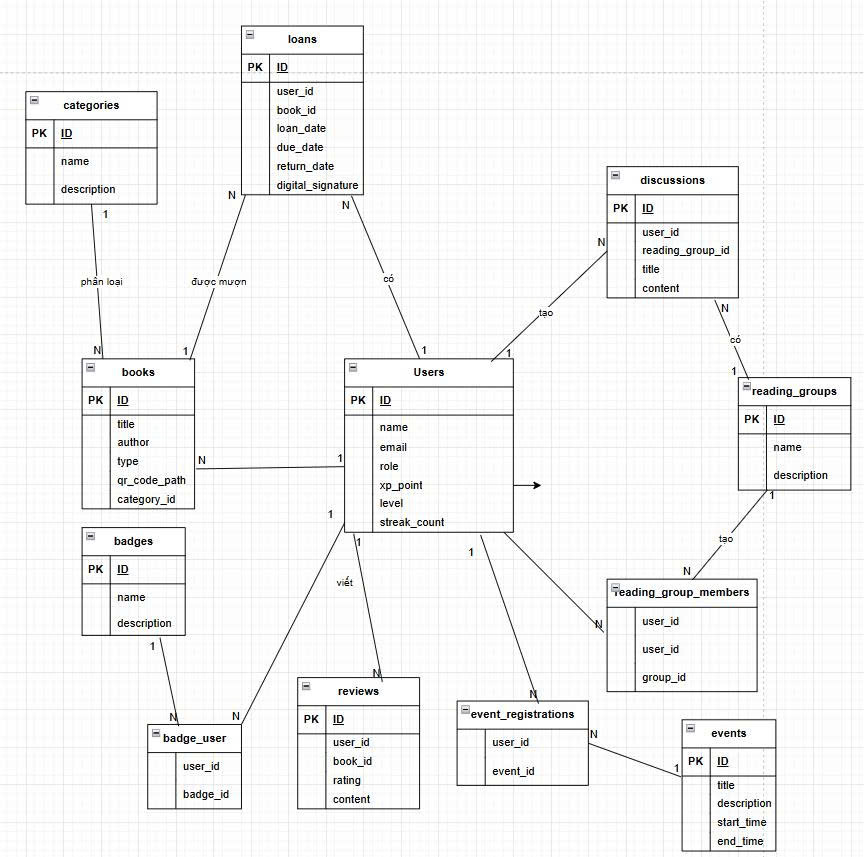
* **Mô tả Biểu đồ:**

Hệ thống được thiết kế nhằm quản lý toàn diện quá trình mượn, trả và đánh giá sách, đồng thời tạo ra một môi trường cộng đồng đọc sách trực tuyến. Cấu trúc cơ sở dữ liệu bao gồm các thành phần chính sau:

* **Book & Category**: Quản lý thông tin sách (tựa đề, tác giả, mô tả, hình ảnh, mã QR) và phân loại sách theo danh mục.
* **Loan**: Ghi nhận quá trình mượn – trả sách của người dùng, kèm theo ngày mượn, hạn trả, tình trạng chữ ký số và tính hợp lệ.
* **User**: Lưu trữ thông tin tài khoản, vai trò, điểm kinh nghiệm, cấp độ và lịch sử hoạt động. Người dùng có thể viết đánh giá, tham gia nhóm đọc, tạo thảo luận và nhận huy hiệu.
* **Review**: Cho phép người dùng đánh giá và phản hồi về sách thông qua điểm số và nội dung bình luận.
* **Badge & badge\_user**: Quản lý hệ thống huy hiệu và theo dõi quá trình trao thưởng cho người dùng, góp phần khuyến khích sự tham gia.
* **ReadingGroup & reading\_group\_member**: Hỗ trợ việc hình thành các nhóm đọc, quản lý thành viên và vai trò trong nhóm.
* **Discussion**: Tạo không gian thảo luận, trao đổi giữa các thành viên trong cùng một nhóm đọc.

Với mô hình này, hệ thống không chỉ đáp ứng nhu cầu quản lý thư viện truyền thống mà còn mở rộng chức năng theo hướng cộng đồng, giúp người dùng tương tác, trao đổi kiến thức và gia tăng trải nghiệm đọc sách thông qua cơ chế nhóm đọc, đánh giá, thảo luận và hệ thống huy hiệu khuyến khích.

**b. Sơ đồ cơ sở dữ liệu**

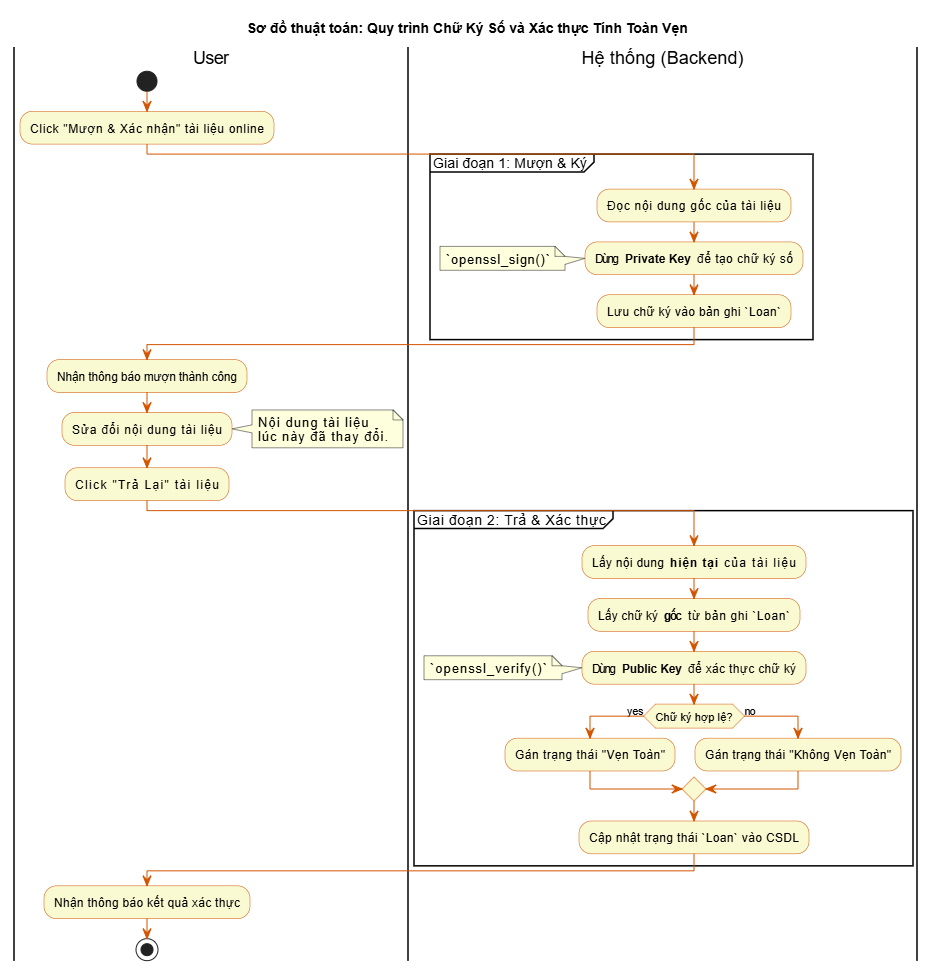


* **Mô tả Biểu đồ:**
  + Sơ đồ trực quan hóa **9 lớp đối tượng chính** của hệ thống, tương ứng với 9 bảng trong cơ sở dữ liệu: users, books, loans, categories, events, badges, reviews, discussions, và reading\_groups.
  + **Các thuộc tính quan trọng:** Mỗi lớp được định nghĩa với các thuộc tính cần thiết. Ví dụ, lớp loans chứa thuộc tính digital\_signature để lưu chữ ký số, và lớp users có các thuộc tính xp\_points, level để phục vụ cho hệ thống Gamification.
  + **Mối quan hệ:**
    - **Quan hệ Một-Nhiều (1-n):** Được thể hiện rõ ràng, ví dụ một User có thể có nhiều Loan (lượt mượn), một Book có thể có nhiều Review.
    - **Quan hệ Nhiều-Nhiều (n-n):** Các mối quan hệ phức tạp như giữa users và badges (một người dùng có thể nhận nhiều huy hiệu, một huy hiệu có thể được trao cho nhiều người) được mô hình hóa thông qua các bảng trung gian (pivot tables) như badge\_user.

#### **3.5. Thiết Kế Luồng Hoạt Động (Biểu đồ Hoạt động)**

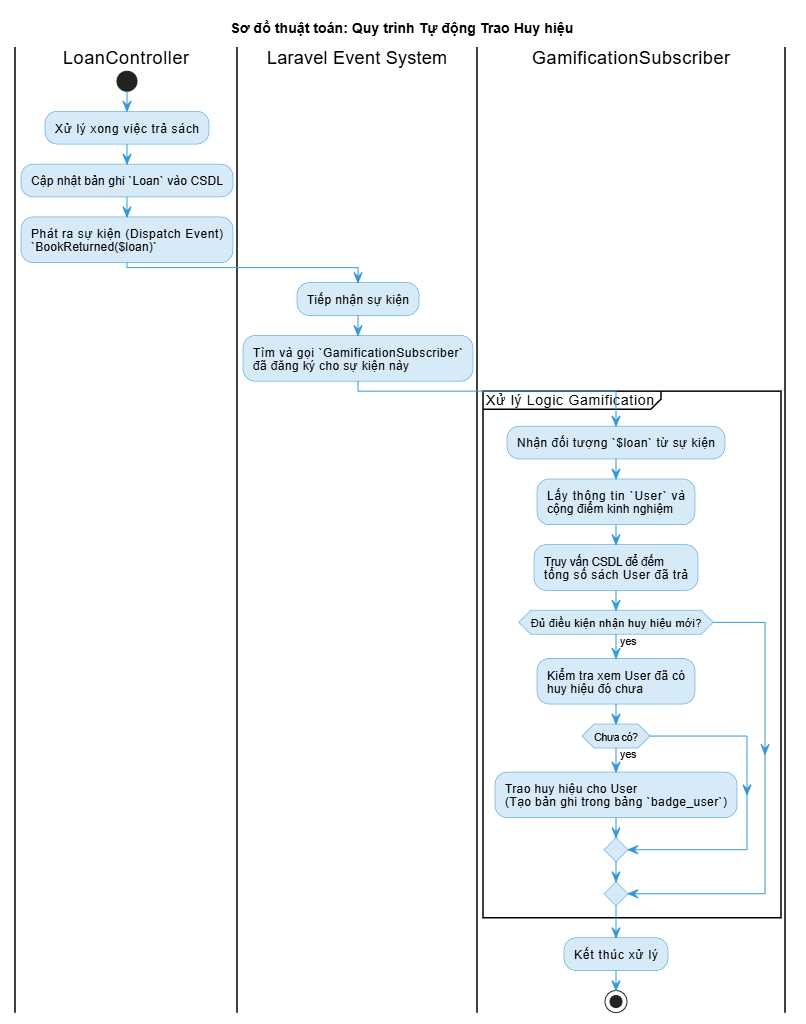
Biểu đồ Hoạt động (Activity Diagram) được sử dụng để mô tả các luồng công việc (workflows) phức tạp của hệ thống. Trong phạm vi dự án, hai luồng hoạt động quan trọng nhất được thiết kế chi tiết là quy trình ký số và quy trình tự động trao huy hiệu.

Đây là luồng nghiệp vụ "trái tim" của dự án, thể hiện cách hệ thống đảm bảo an toàn và toàn vẹn cho tài liệu điện tử.



* **Mô tả Luồng hoạt động:** Sơ đồ được chia thành hai làn bơi (swimlanes) là **User** và **Hệ thống (Backend)**, và bao gồm hai giai đoạn chính:
  + **Giai đoạn 1: Mượn & Ký (Signing)**
    1. **User** thực hiện hành động "Mượn & Xác nhận" trên giao diện.
    2. **Hệ thống** tiếp nhận yêu cầu, đọc nội dung gốc của tài liệu.
    3. Sử dụng hàm openssl\_sign() với **Private Key** của hệ thống để tạo ra chữ ký số từ nội dung gốc.
    4. Lưu chữ ký số này vào bản ghi Loan tương ứng trong cơ sở dữ liệu.
    5. Hệ thống gửi thông báo mượn thành công cho User.
  + **Giai đoạn 2: Trả & Xác thực (Verification)**
    1. Sau một thời gian, **User** thực hiện hành động "Trả Lại" tài liệu (giả định rằng tài liệu có thể đã bị sửa đổi).
    2. **Hệ thống** tiếp nhận yêu cầu, lấy nội dung **hiện tại** của tài liệu mà người dùng trả.
    3. Truy vấn cơ sở dữ liệu để lấy chữ ký số **gốc** đã được lưu trong bản ghi Loan.
    4. Sử dụng hàm openssl\_verify() với **Public Key** để xác thực nội dung hiện tại dựa trên chữ ký gốc.
    5. Dựa vào kết quả của openssl\_verify(), hệ thống quyết định chữ ký có hợp lệ hay không. Nếu hợp lệ, trạng thái được gán là "Vẹn Toàn"; nếu không, là "Không Vẹn Toàn".
    6. Cập nhật trạng thái toàn vẹn này vào bản ghi Loan và gửi thông báo kết quả xác thực cho User.

Sơ đồ này minh họa kiến trúc hướng sự kiện (Event-Driven Architecture) của Laravel được áp dụng để triển khai tính năng Gamification một cách linh hoạt và tách biệt.



* **Mô tả Luồng hoạt động:** Sơ đồ được chia thành ba làn bơi, thể hiện sự tương tác giữa các thành phần khác nhau của hệ thống.
  1. **LoanController:**
     + Khi người dùng thực hiện thành công việc trả sách, Controller này sẽ xử lý logic nghiệp vụ chính (cập nhật bản ghi Loan vào CSDL).
     + Sau khi hoàn tất, thay vì xử lý logic Gamification trực tiếp, nó sẽ **phát ra một sự kiện (Dispatch Event)** có tên BookReturned, gói kèm đối tượng $loan chứa thông tin về lượt trả.
  2. **Laravel Event System:**
     + Hệ thống sự kiện của Laravel đóng vai trò trung gian, tiếp nhận sự kiện BookReturned.
     + Nó tự động tìm kiếm và thông báo cho tất cả các **Listeners** hoặc **Subscribers** đã đăng ký lắng nghe sự kiện này. Trong trường hợp này, nó sẽ tìm thấy GamificationSubscriber.
  3. **GamificationSubscriber:**
     + Đây là nơi chứa toàn bộ logic xử lý của tính năng Gamification, tách biệt hoàn toàn khỏi LoanController.
     + Khi được kích hoạt, nó nhận đối tượng $loan từ sự kiện.
     + Từ $loan, nó lấy ra thông tin User và cộng điểm kinh nghiệm.
     + Nó truy vấn CSDL để kiểm tra các điều kiện nhận huy hiệu mới (ví dụ: đếm tổng số sách User đã trả).
     + Nếu người dùng đủ điều kiện và chưa sở hữu huy hiệu đó, hệ thống sẽ **trao huy hiệu** bằng cách tạo một bản ghi mới trong bảng trung gian badge\_user.
     + Kết thúc xử lý.

Thiết kế này giúp hệ thống dễ dàng bảo trì và mở rộng; ví dụ, nếu sau này muốn thêm chức năng gửi email khi người dùng nhận huy hiệu, ta chỉ cần tạo thêm một Listener mới mà không cần sửa đổi code trong LoanController.

#### **3.6. Thiết Kế Trạng Thái Đối Tượng (Biểu đồ Trạng thái)**

Biểu đồ Trạng thái (State Diagram) mô tả các trạng thái khác nhau mà một đối tượng có thể trải qua trong vòng đời của nó, cũng như các sự kiện gây ra sự chuyển đổi giữa các trạng thái. Đối tượng được chọn để mô hình hóa là Loan (Lượt mượn), vì nó có vòng đời phức tạp và quan trọng nhất trong hệ thống.



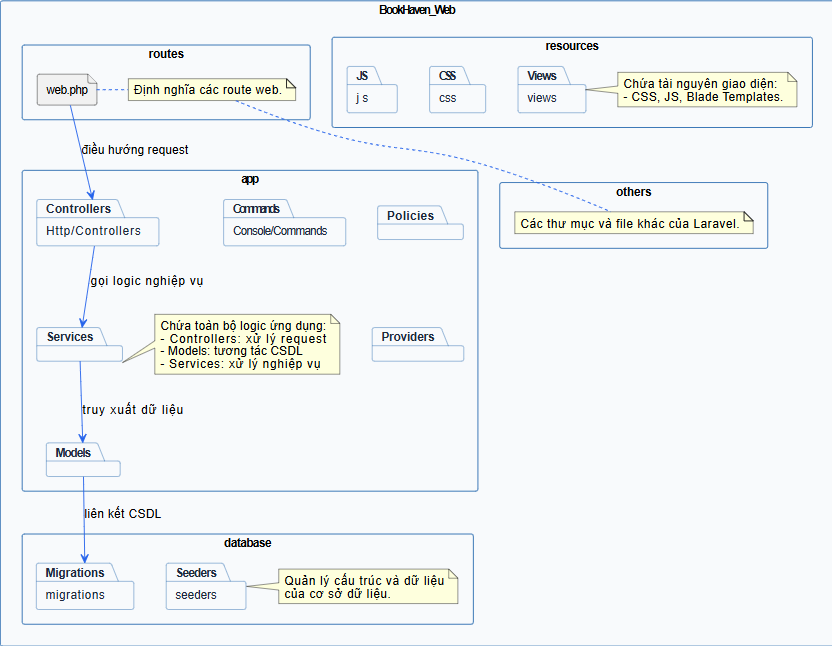
* **Mô tả Biểu đồ:**
  + **Bắt đầu:** Vòng đời của một lượt mượn bắt đầu khi người dùng tạo yêu cầu, chuyển vào trạng thái **"Đang chờ duyệt"**.
  + **Các trạng thái chính:**
    - **Đang chờ duyệt:** Chờ Admin xác nhận yêu cầu.
    - **Đang mượn:** Trạng thái chính sau khi được duyệt. Đây là một trạng thái phức hợp (composite state) chứa các trạng thái con là **"Trong hạn"** và **"Quá hạn"**.
    - **Đang xử lý trả:** Trạng thái trung gian khi người dùng trả sách, chờ hệ thống xác thực chữ ký số.
    - **Hoàn thành:** Trạng thái kết thúc thành công sau khi trả sách và xác thực thành công.
    - **Đã hủy:** Trạng thái kết thúc khi người dùng hủy yêu cầu trước khi được duyệt.
  + **Các chuyển đổi:** Các mũi tên biểu thị sự kiện gây ra sự thay đổi trạng thái. Ví dụ, sự kiện "Admin duyệt" chuyển Loan từ "Đang chờ duyệt" sang "Đang mượn". Sự kiện "Xác thực thất bại" sẽ chuyển Loan từ "Đang xử lý trả" quay trở lại "Đang mượn" để người dùng xử lý lại.

#### **3.6. Thiết Kế kiến trúc phân tầng (Biểu đồ Gói)**

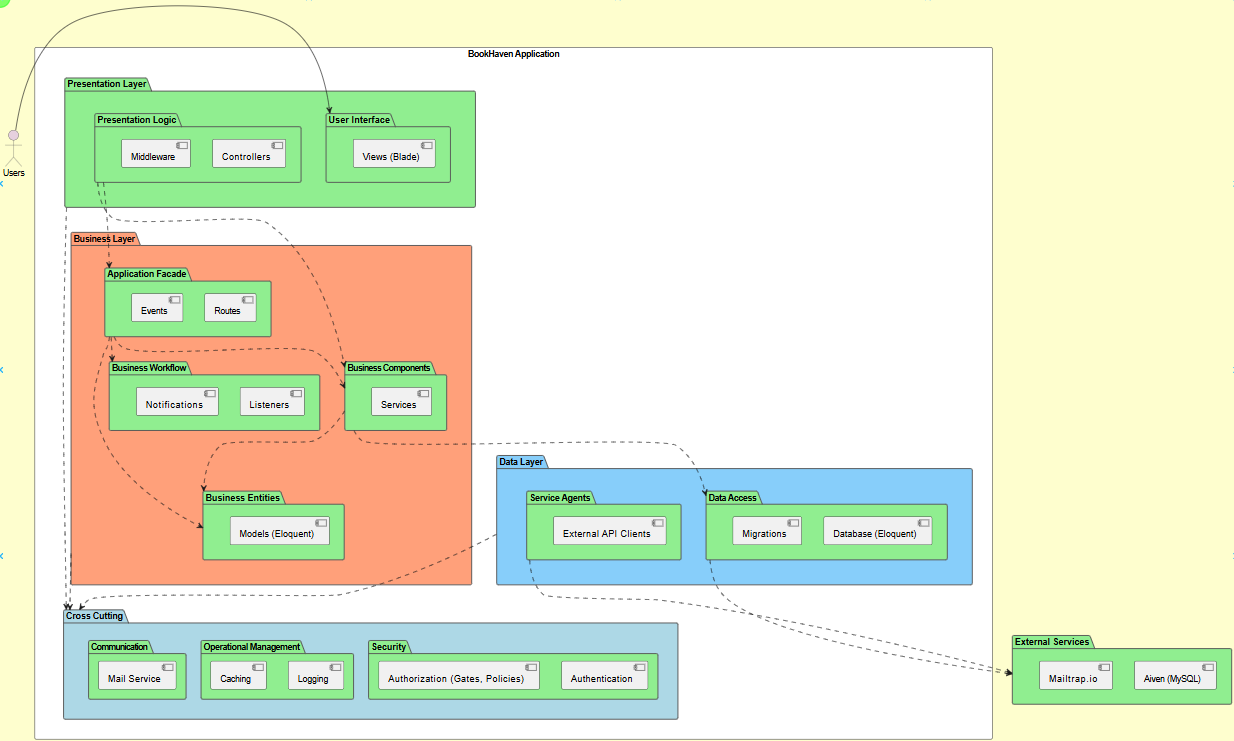
Biểu đồ Gói là một công cụ tuyệt vời để mô tả kiến trúc cấp cao của hệ thống. Nó cho thấy cách mã nguồn được tổ chức thành các gói (packages) hoặc các tầng (layers) logic và sự phụ thuộc giữa chúng.

### **Giải thích về Kiến trúc và các Gói**

Biểu đồ này chia hệ thống BookHaven thành 4 tầng logic chính, đây là một kiến trúc sạch (Clean Architecture) giúp mã nguồn dễ dàng bảo trì, kiểm thử và mở rộng.



1. **Tầng Giao diện:**
   * **Trách nhiệm:** Là lớp ngoài cùng, chịu trách nhiệm tiếp nhận các yêu cầu HTTP từ người dùng và hiển thị dữ liệu ra giao diện.
   * **Thành phần chính:**
     + Controllers: Nhận request, gọi các service ở tầng dưới để xử lý và trả về response (thường là một View).
     + Views: Các file Blade template chứa mã HTML để hiển thị giao diện.
     + Middleware: Các lớp xử lý trung gian để lọc request (ví dụ: kiểm tra người dùng đã đăng nhập chưa).
2. **Tầng Ứng dụng:**
   * **Trách nhiệm:** Điều phối các hoạt động và chứa logic nghiệp vụ chính của ứng dụng. Tầng này không quan tâm đến cách dữ liệu được hiển thị hay lưu trữ.
   * **Thành phần chính:**
     + Services: Nơi chứa các logic nghiệp vụ phức tạp (ví dụ: DigitalSignatureService, GamificationService).
     + Events & Listeners: Triển khai kiến trúc hướng sự kiện, giúp tách biệt các mối quan tâm (ví dụ: khi người dùng trả sách, phát ra sự kiện để listener cộng điểm).
     + Notifications: Quản lý việc gửi thông báo qua các kênh như email.
3. **Tầng Lõi:**
   * **Trách nhiệm:** Là trái tim của hệ thống, chứa các đối tượng nghiệp vụ (entities) và các quy tắc cốt lõi không thay đổi.
   * **Thành phần chính:**
     + Models: Các lớp Eloquent đại diện cho các thực thể trong CSDL như User, Book, Loan.
     + Policies & Gates: Nơi định nghĩa các quy tắc phân quyền, gắn liền với các đối tượng trong Domain.
4. **Tầng Hạ tầng:**
   * **Trách nhiệm:** Bao gồm tất cả các chi tiết kỹ thuật về cách dữ liệu được lưu trữ và cách giao tiếp với các hệ thống bên ngoài.
   * **Thành phần chính:**
     + Database: Bao gồm các file migrations, seeders, và cấu hình kết nối CSDL.
     + External Services: Các lớp để giao tiếp với dịch vụ của bên thứ ba như Aiven (CSDL cloud) hay Mailtrap (gửi email).
     + File Storage: Quản lý việc lưu trữ các file vật lý như ảnh bìa, mã QR.



### **Giải thích về việc ánh xạ kiến trúc BookHaven vào biểu đồ:**

1. **Users:** Chính là các người dùng cuối tương tác với hệ thống qua trình duyệt.
2. **Presentation Layer (Tầng Giao diện):**
   * User Interface: Chính là các file **Views (Blade)**, chịu trách nhiệm hiển thị HTML cho người dùng.
   * Presentation Logic: Là các **Controllers** (tiếp nhận request) và **Middleware** (lọc request).
3. **Business Layer (Tầng Nghiệp vụ):** Đây là nơi chứa logic cốt lõi của ứng dụng.
   * Business Entities: Là các **Models (Eloquent)** như User, Book, Loan. Chúng đại diện cho các thực thể nghiệp vụ.
   * Business Components: Là các lớp **Services** (DigitalSignatureService, GamificationService) nơi xử lý các nghiệp vụ phức tạp.
   * Business Workflow: Được thể hiện qua cặp **Listeners** và **Notifications**. Ví dụ, một Event xảy ra (người dùng trả sách) sẽ kích hoạt một Listener để thực hiện một chuỗi công việc (cộng điểm, trao huy hiệu).
   * Application Facade: Là "cổng vào" của tầng nghiệp vụ, được đại diện bởi file **Routes** (định tuyến request đến đúng controller) và các **Events** (kích hoạt các workflow).
4. **Data Layer (Tầng Dữ liệu):** Chịu trách nhiệm về việc lưu trữ và truy xuất dữ liệu.
   * Data Access: Chính là bản thân **Eloquent ORM** và các file **Migrations**. Eloquent cung cấp một giao diện lập trình để tương tác với CSDL.
   * Service Agents: Là các lớp client dùng để giao tiếp với các dịch vụ bên ngoài.
5. **External Services:** Là các dịch vụ của bên thứ ba mà BookHaven sử dụng, bao gồm **Aiven** cho CSDL và **Mailtrap.io** để gửi email.
6. **Cross Cutting (Các thành phần Xuyên suốt):** Đây là các chức năng ảnh hưởng đến toàn bộ các tầng của ứng dụng. Laravel đã cung cấp sẵn các cơ chế này.
   * Security: Bao gồm hệ thống **Authentication** (xác thực) và **Authorization** (phân quyền qua Gates, Policies).
   * Operational Management: Các cơ chế **Logging** (ghi log lỗi) và **Caching** (lưu cache) của Laravel.
   * Communication: Dịch vụ **Mail Service** để gửi email thông báo.

-------------------------------------------------------------------------------------------------------------------------------------------

### **CHƯƠG 4: TRIỂN KHAI XÂY DỰNG HỆ THỐNG**

Sau khi hoàn tất giai đoạn thiết kế chi tiết ở Chương 3, chương này sẽ tập trung vào quá trình hiện thực hóa các bản thiết kế thành một sản phẩm phần mềm hoàn chỉnh. Nội dung chương sẽ trình bày về môi trường phát triển, các công nghệ được sử dụng, cấu trúc mã nguồn của dự án và đi sâu vào việc triển khai một số module chức năng nổi bật, kèm theo đó là các giao diện chính của hệ thống BookHaven.

#### **4.1. Môi Trường và Công Nghệ Sử Dụng**

Để đảm bảo dự án được phát triển một cách hiệu quả và ổn định, một môi trường làm việc tiêu chuẩn cùng với các công nghệ hiện đại đã được lựa chọn và thiết lập.

* **4.1.1. Môi trường phát triển:**
  + **Hệ điều hành:** Windows 11
  + **Web Server:** XAMPP / LARAGON (Apache)
  + **Hệ quản trị CSDL:** MySQL 8.0+
  + **Trình thông dịch:** PHP 8.1+
  + **Công cụ quản lý dependencies:** Composer, NPM
  + **IDE (Môi trường phát triển tích hợp):** Visual Studio Code
* **4.1.2. Công nghệ và Thư viện sử dụng:**

**Bảng 4.1: Bảng kê chi tiết các công nghệ được sử dụng trong dự án**

|  |  |  |
| --- | --- | --- |
| **Hạng mục** | **Công nghệ/Thư viện** | **Mô tả và Vai trò** |
| **Backend Framework** | Laravel 10 | Nền tảng chính để xây dựng toàn bộ logic ứng dụng theo kiến trúc MVC. |
| **Frontend Framework** | Tailwind CSS | Utility-first CSS framework giúp xây dựng giao diện responsive một cách nhanh chóng và nhất quán. |
| **JavaScript Framework** | Alpine.js | Một framework JavaScript gọn nhẹ để xây dựng các thành phần tương tác trên giao diện người dùng. |
| **Database** | MySQL 8.0+ | Hệ quản trị cơ sở dữ liệu quan hệ để lưu trữ toàn bộ dữ liệu của hệ thống. |
| **Authentication** | Laravel Breeze | Cung cấp hệ thống xác thực người dùng an toàn và đầy đủ (đăng ký, đăng nhập, đặt lại mật khẩu). |
| **Asset Bundling** | Vite | Công cụ build frontend thế hệ mới, giúp biên dịch assets (CSS, JS) với tốc độ cực nhanh. |
| **Email Testing** | Mailtrap.io | Dịch vụ SMTP giả lập để kiểm thử chức năng gửi email thông báo trong môi trường phát triển. |

#### **4.2. Cấu Trúc Thư Mục Dự Án**

Dự án tuân thủ cấu trúc thư mục tiêu chuẩn của Laravel, giúp mã nguồn được tổ chức một cách logic, rõ ràng và dễ dàng bảo trì.

BookHaven\_Web/  
├── app/  
│ ├── Console/Commands/ # Chứa các lệnh Artisan tùy chỉnh (VD: xử lý thông báo)  
│ ├── Http/Controllers/ # Chứa các lớp Controller xử lý request  
│ ├── Models/ # Chứa các lớp Eloquent Model tương tác với CSDL  
│ ├── Policies/ # Chứa các lớp Policy định nghĩa quyền truy cập  
│ ├── Providers/ # Chứa các Service Provider của ứng dụng  
│ └── Services/ # Chứa các lớp Service xử lý logic nghiệp vụ phức tạp  
├── database/  
│ ├── migrations/ # Chứa các file migration định nghĩa cấu trúc bảng  
│ └── seeders/ # Chứa các file seeder để tạo dữ liệu mẫu  
├── public/ # Thư mục gốc của web server, chứa file index.php  
│ ├── sw.js # File Service Worker cho PWA  
│ └── manifest.json # File Manifest cho PWA  
├── resources/  
│ ├── css/ # Chứa file CSS nguồn  
│ ├── js/ # Chứa file JavaScript nguồn  
│ └── views/ # Chứa các file Blade template cho giao diện  
├── routes/  
│ └── web.php # Nơi định nghĩa các route của ứng dụng web  
└── ... # Các thư mục và file khác của Laravel

#### **4.3. Triển Khai Các Module Chức Năng Nổi Bật**

Phần này sẽ đi sâu vào việc hiện thực hóa một số chức năng cốt lõi và phức tạp nhất của hệ thống.

Đây là module quan trọng nhất, đảm bảo tính bảo mật cho tài liệu điện tử.

* **Tạo cặp khóa:** Một cặp khóa RSA (Private/Public Key) được tạo ra và lưu trữ an toàn trên server. Public Key có thể được phân phối công khai, trong khi Private Key được bảo vệ nghiêm ngặt.
* **Logic Ký số khi mượn:** Khi người dùng yêu cầu mượn một tài liệu online, luồng xử lý sau được thực thi:
  1. Đọc nội dung gốc của file tài liệu.
  2. Sử dụng hàm openssl\_sign() của PHP, kết hợp nội dung tài liệu, **Private Key** và thuật toán OPENSSL\_ALGO\_SHA256 để tạo ra chữ ký.
  3. Lưu chữ ký số (dưới dạng binary hoặc base64 encoded) vào cột digital\_signature trong bảng loans.
* **Logic Xác thực khi trả:**
  1. Nhận nội dung tài liệu mà người dùng trả về.
  2. Lấy chữ ký số gốc từ bảng loans tương ứng với lượt mượn.
  3. Sử dụng hàm openssl\_verify(), kết hợp nội dung tài liệu trả về, chữ ký gốc, **Public Key** và thuật toán OPENSSL\_ALGO\_SHA256.
  4. Hàm này sẽ trả về 1 nếu hợp lệ, 0 nếu không hợp lệ. Dựa vào kết quả này, hệ thống cập nhật trạng thái toàn vẹn của tài liệu.

Để logic Gamification không làm ảnh hưởng đến các luồng nghiệp vụ chính, dự án đã áp dụng kiến trúc hướng sự kiện (Event-Driven Architecture) của Laravel.

* **Định nghĩa Events:** Các lớp sự kiện như BookReturned, ReviewPosted được tạo ra trong thư mục app/Events.
* **Phát ra Events:** Trong các Controller tương ứng, sau khi xử lý xong logic chính, sự kiện sẽ được phát ra. Ví dụ, trong LoanController:

// Sau khi cập nhật trạng thái trả sách thành công  
BookReturned::dispatch($loan);

* **Đăng ký Subscribers (Listeners):** Một lớp GamificationSubscriber được tạo để lắng nghe tất cả các sự kiện liên quan. Trong EventServiceProvider, ta đăng ký subscriber này:

protected $subscribe = [  
 GamificationSubscriber::class,  
];

* **Xử lý Logic:** Bên trong GamificationSubscriber, các phương thức sẽ xử lý logic tương ứng cho từng sự kiện, ví dụ:

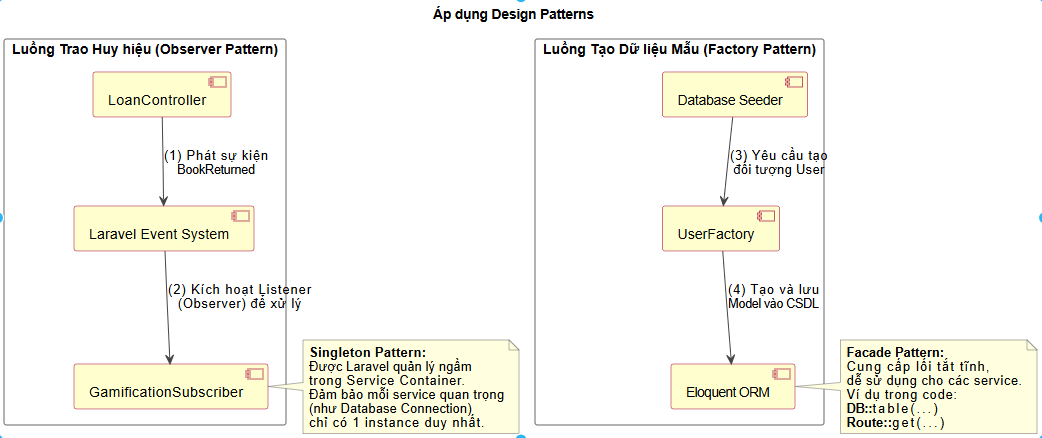
public function handleBookReturned(BookReturned $event) {  
 // Lấy user từ $event->loan  
 // Cộng điểm kinh nghiệm cho user  
 // Kiểm tra và trao huy hiệu "Trả sách lần đầu"  
}

Để biến BookHaven thành một PWA, hai thành phần chính đã được triển khai:

* **Web App Manifest (manifest.json):** Một file JSON được đặt trong thư mục public/, định nghĩa các thông tin cần thiết để trình duyệt có thể "cài đặt" ứng dụng, bao gồm tên ứng dụng, icon, màu nền, trang bắt đầu...
* **Service Worker (sw.js):** Một file JavaScript được đăng ký với trình duyệt. File này chứa logic để chặn các yêu cầu mạng (fetch event), lưu trữ các tài nguyên tĩnh (CSS, JS, images) và dữ liệu sách vào Cache Storage. Khi người dùng truy cập lại hoặc bị mất mạng, Service Worker sẽ trả về tài nguyên từ cache thay vì gửi yêu cầu lên server, giúp ứng dụng có thể hoạt động offline.

#### **4.4. Áp dụng các Mẫu thiết kế (Design Patterns)**

Trong quá trình phát triển dự án BookHaven, nhóm đã tận dụng các mẫu thiết kế phần mềm tiêu chuẩn để giải quyết các bài toán phổ biến, nâng cao chất lượng mã nguồn, và đảm bảo hệ thống dễ dàng bảo trì, mở rộng. Dưới đây là các mẫu thiết kế chính đã được áp dụng.



**Mô tả sơ đồ:**

* Sơ đồ thể hiện LoanController sử dụng Event Dispatcher để thông báo cho GamificationSubscriber, minh họa cho **Observer Pattern**.
* Database Seeder sử dụng UserFactory để tạo Model, minh họa cho **Factory Pattern**.
* Các ghi chú bên cạnh làm rõ vai trò của **Singleton Pattern** (thông qua Service Container) và **Facade Pattern**.
* **4.4.1. Singleton Pattern:**
  + (Mô tả ngắn gọn mục đích của pattern).
  + **Triển khai:** Hệ thống BookHaven tận dụng cơ chế Service Container của Laravel, nơi các dịch vụ quan trọng như quản lý kết nối cơ sở dữ liệu (Database Connection), quản lý bộ đệm (Cache) được đăng ký dưới dạng singleton. Điều này đảm bảo mỗi dịch vụ chỉ có một thể hiện duy nhất trong mỗi vòng đời của request, giúp tối ưu hóa tài nguyên và đảm bảo tính nhất quán của dữ liệu.
* **4.5.2. Factory Pattern:**
  + (Mô tả ngắn gọn mục đích của pattern).
  + **Triển khai:** Mẫu này được áp dụng thông qua tính năng Model Factories của Eloquent. Nhóm đã tạo các lớp Factory như UserFactory, BookFactory, LoanFactory để định nghĩa cách tạo ra các đối tượng Model với dữ liệu giả. Các factory này được sử dụng trong quá trình seed dữ liệu, giúp nhanh chóng tạo ra một môi trường dữ liệu mẫu phong phú để phát triển và kiểm thử.
* **4.4.3. Observer Pattern:**
  + (Mô tả ngắn gọn mục đích của pattern).
  + **Triển khai:** Đây là mẫu thiết kế nền tảng cho kiến trúc hướng sự kiện của module Gamification. Khi một hành động quan trọng xảy ra (ví dụ: người dùng trả sách), LoanController sẽ phát ra một sự kiện BookReturned. Một lớp GamificationSubscriber đã đăng ký "lắng nghe" sự kiện này sẽ tự động được kích hoạt để thực hiện logic cộng điểm và trao huy hiệu. Việc này giúp tách biệt hoàn toàn logic nghiệp vụ chính khỏi các logic phụ trợ, làm cho mã nguồn trở nên sạch sẽ và linh hoạt hơn.
* **4.4.4. Facade Pattern:**
  + (Mô tả ngắn gọn mục đích của pattern).
  + **Triển khai:** Nhóm đã sử dụng rộng rãi các Facades mà Laravel cung cấp như Route, DB, Log, Auth. Các Facades này cung cấp một cú pháp tĩnh, ngắn gọn và biểu cảm để truy cập các dịch vụ phức tạp của framework, giúp quá trình lập trình trở nên nhanh chóng và dễ đọc hơn mà không cần phải inject các dependency một cách thủ công.

#### **4.5. Giao Diện Người Dùng (UI)**

Giao diện của BookHaven được thiết kế theo phong cách hiện đại, tối giản và tập trung vào trải nghiệm người dùng, sử dụng Tailwind CSS để đảm bảo tính nhất quán và responsive.

|  |  |
| --- | --- |
| **Trang Chủ & Slider Sách Phổ Biến** | **Trang Chi Tiết Sách & Nút Tương Tác** |
|  |  |
| **Mô tả:** Giao diện trang chủ được thiết kế thoáng đãng, nổi bật với slider giới thiệu các đầu sách phổ biến. Người dùng có thể dễ dàng tìm kiếm hoặc khám phá các danh mục sách. | **Mô tả:** Hiển thị đầy đủ thông tin về một cuốn sách, bao gồm bìa sách, tóm tắt, tác giả và các đánh giá. Các nút hành động chính như "Mượn Ngay" hay "Thêm vào Yêu thích" được đặt ở vị trí nổi bật. |
| **Thư Viện Ảo 3D** | **Dashboard Quản Trị của Admin** |
|  |  |
| **Mô tả:** Một không gian 3D tương tác độc đáo, cho phép người dùng sử dụng chuột và bàn phím để "dạo bước" giữa các kệ sách, mang lại một trải nghiệm khám phá mới lạ và thú vị. | **Mô tả:** Giao diện quản trị cung cấp cái nhìn tổng quan về hoạt động của thư viện thông qua các widget và biểu đồ (biểu đồ đường về lượt mượn, biểu đồ tròn về phân bố sách), giúp Admin dễ dàng nắm bắt tình hình. |
| **Quản lý Lịch sử Mượn/Trả** | **Trang Profile với Huy hiệu** |
|  |  |
| **Mô tả:** Người dùng có thể theo dõi toàn bộ lịch sử mượn sách của mình. Giao diện hiển thị rõ ràng các thông tin như ngày mượn, hạn trả, và đặc biệt là trạng thái toàn vẹn của tài liệu sau khi được hệ thống xác thực. | **Mô tả:** Trang hồ sơ cá nhân của người dùng, nơi hiển thị các thành tích đã đạt được trong quá trình đọc sách. Các huy hiệu được thiết kế trực quan, thể hiện các cột mốc mà người dùng đã chinh phục. |

|  |  |
| --- | --- |
| **Trang Cộng Đồng & Thảo Luận** | **Thành Tích & Bảng Xếp Hạng** |
|  |  |
| **Mô tả:** Một không gian social feed nơi hiển thị các hoạt động mới nhất từ cộng đồng như review, bình luận, và các cuộc thảo luận. Người dùng có thể tham gia tương tác, tạo nhóm đọc và kết nối với những người cùng sở thích. | **Mô tả:** Giao diện vinh danh nỗ lực của người đọc, hiển thị danh sách các huy hiệu đã đạt được, tiến độ các thử thách, và bảng xếp hạng những người dùng tích cực nhất, tạo động lực cạnh tranh lành mạnh. |

-------------------------------------------------------------------------------------------------------------------------------------

### **CHƯƠNG 5: KIỂM THỬ VÀ ĐÁNH GIÁ**