BÀI THỰC HÀNH SỐ 7

**BẢN TÔN CHỈ DỰ ÁN**

**1. HIẾN CHƯƠNG DỰ ÁN**

|  |  |
| --- | --- |
| **1.0 XÁC ĐỊNH DỰ ÁN** | |
| **Tên dự án** | Hệ thống Đặt lịch khám bệnh Nhi Đồng 2 |
| **Mô tả ngắn** | Thiết kế, phát triển và triển khai một nền tảng trực tuyến để quản lý việc đặt lịch khám bệnh tại Bệnh viện Nhi Đồng II. Hệ thống bao gồm các chức năng quản lý người dùng (bệnh nhân, bác sĩ, admin), quản lý lịch hẹn, và quản lý hồ sơ bệnh án. |
| **Nhà tài trợ** |  |
| **Quản lý dự án** | Đặng Dương Nguyên – MSSV 1150080150  Võ Anh Kiệt – MSSV 1150080143 |
| **Thành viên nhóm** | - Đặng Dương Nguyên: Phát triển ứng dụng Flutter (Frontend,Bloc).  - Võ Anh Kiệt: Xây dựng API và cơ sở dữ liệu bằng Spring Boot (Backend, PostgreSQL). |
| **Công cụ & Nguồn lực** | Flutter, Dart, Bloc,PostgreSQL, Spring Boot, Spring Security, JWT, Lombok, Thymeleaf, Swagger/OpenAPI, Cloudinary |

**2. LÝ DO KINH DOANH CỦA DỰ ÁN**

* Nâng cao trải nghiệm của bệnh nhân bằng cách cung cấp phương thức đặt lịch hẹn thuận tiện, không cần phải đến trực tiếp bệnh viện.
* Cải thiện quy trình quản lý lịch hẹn của bệnh viện, giảm tải công việc hành chính thủ công.
* Cho phép bác sĩ quản lý lịch làm việc và bệnh án điện tử một cách hiệu quả.
* Cung cấp cho admin công cụ để quản lý người dùng, chuyên khoa, và bác sĩ trong hệ thống.

**3. MỤC TIÊU DỰ ÁN**

* Xây dựng hệ thống backend hoàn chỉnh bằng Spring Boot cho phép quản lý người dùng (sinh viên, giảng viên, admin), hoạt động, và đăng ký.
* Triển khai hệ thống xác thực và phân quyền dựa trên vai trò (Role-based) sử dụng JWT (JSON Web Token).
* Cho phép sinh viên đăng ký/hủy đăng ký tham gia các hoạt động.
* Cho phép giảng viên hoặc quản trị viên tạo, cập nhật, xóa và công bố các hoạt động.
* Cung cấp chức năng điểm danh (theo từng sinh viên hoặc hàng loạt) cho các hoạt động.
* Tích hợp dịch vụ gửi email để xác thực tài khoản sau khi đăng ký.
* Cung cấp một giao diện web (dựa trên Thymeleaf) cho các vai trò quản trị (Admin) và bác sĩ để quản lý hệ thống.
* Phát triển các API công khai cho phép ứng dụng di động (Frontend) tương tác để tìm kiếm bác sĩ và đặt lịch.

**4. PHẠM VI DỰ ÁN**

1. **Bao gồm:**

* API để đăng ký, đăng nhập và xác thực tài khoản.
  + Quản lý người dùng: xem danh sách, cập nhật vai trò, khóa/mở khóa tài khoản (cho admin).
  + Quản lý chuyên khoa và bác sĩ: tạo, sửa, xóa chuyên khoa; tạo hồ sơ bác sĩ và gán chuyên khoa.
  + Quản lý lịch hẹn: tạo, sửa, hủy lịch hẹn (bệnh nhân) và xem lịch hẹn (bác sĩ).
  + Quản lý hồ sơ bệnh án: tạo bệnh án sau khi khám.
  + Hỗ trợ đa nền tảng (Frontend độc lập).

1. **Không bao gồm:**
   * Giao diện người dùng cho bệnh nhân (sẽ được phát triển riêng bởi đội ngũ Flutter).
   * Hệ thống thanh toán trực tuyến.
   * Tích hợp với hệ thống y tế hiện có của bệnh viện

**5. CÁC KẾT QUẢ CHÍNH CỦA DỰ ÁN**

|  |  |
| --- | --- |
| **API Backend** | Một ứng dụng Spring Boot hoàn chỉnh cung cấp các API RESTful để quản lý hệ thống đặt lịch khám bệnh. |
| **Ứng dụng di động (Frontend)** | Ứng dụng Flutter cho bệnh nhân, bao gồm các màn hình chính như đăng nhập, đặt lịch, và xem hồ sơ. |
| **Cơ sở dữ liệu** | Lưu trữ thông tin người dùng, chuyên khoa, bác sĩ, và lịch hẹn. |
| **Tài liệu API** | Tài liệu Swagger/OpenAPI cho các API công khai và bảo mật. |
| **Giao diện Web Admin/Doctor** | Các tệp HTML/Thymeleaf cho phép quản lý các chức năng nội bộ của hệ thống. |

**6. MỐC THỜI GIAN QUAN TRỌNG**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Hạng mục** | **Thời gian dự kiến** | **Người phụ trách** |
| 1 | Hoàn thành API Đăng ký & Đăng nhập | Tuần 3 | Nguyên |
| 2 | Hoàn thành API Quản lý Người dùng & Chuyên khoa (dành cho Admin) | Tuần 4 | Nguyên |
| 3 | Hoàn thành API Quản lý Bác sĩ & Lịch hẹn | Tuần 5 | Kiệt |
| 4 | Triển khai giao diện web cho Admin và Bác sĩ | Tuần 6 | Nguyên – Kiệt |
| 5 | Tích hợp và kiểm tra API với ứng dụng Flutter (FE) | Tuần 7 | Cả nhóm |
| 6 | Triển khai lên môi trường thử nghiệm (Staging) | Tuần 8 | Nguyên |
| 7 | Triển khai chính thức (Production) | Tuần 9 | Cả nhóm |

**7. CÁC VẤN ĐỀ CHÍNH**

* Thiếu thông tin chi tiết về các yêu cầu nghiệp vụ từ bệnh viện (ví dụ: thời gian khám, quy tắc đặt lịch).
* Khó khăn trong việc tích hợp với hệ thống y tế hiện có của bệnh viện (nếu có).
* Đảm bảo tính bảo mật của dữ liệu bệnh nhân và lịch sử khám bệnh.
* Sự khác biệt trong logic hoặc dữ liệu giữa BE và FE có thể dẫn đến việc phải điều chỉnh lại.

**8. RỦI RO**

|  |  |
| --- | --- |
| **Mức độ** | **Rủi ro** |
| Trung bình | Lỗi bảo mật trong việc xử lý JWT hoặc xác thực người dùng. |
| Cao | Sự không tương thích giữa API backend và ứng dụng di động frontend, dẫn đến việc phải điều chỉnh lại logic. |
| Trung bình | Khó khăn trong việc xử lý đồng thời nhiều yêu cầu đặt lịch, dẫn đến nguy cơ trùng lịch hẹn của bác sĩ. |
| Thấp | Sự không đồng bộ giữa các môi trường phát triển (dev, staging, production) do thiếu quy trình CI/CD. |

**9. TIÊU CHÍ THÀNH CÔNG DỰ ÁN**

* Hoàn thành 100% các chức năng cốt lõi đã được định nghĩa trong phạm vi dự án.
* Tất cả các endpoint API đều có tài liệu Swagger rõ ràng và có thể kiểm thử được.
* Hệ thống phân quyền hoạt động chính xác: người dùng với vai trò không phù hợp không thể truy cập các API được bảo vệ.
* Các quy tắc nghiệp vụ về thời gian và trạng thái (ví dụ: không cho đăng ký khi đã hết hạn, không cho điểm danh khi hoạt động chưa bắt đầu) được áp dụng thành công.
* Nộp đồ án và báo cáo đúng thời hạn quy định.
* Tỷ lệ lỗi xác thực và đặt lịch dưới 1%.
* Tất cả các chức năng được mô tả trong phạm vi dự án hoạt động ổn định và chính xác.
* Ứng dụng di động được phát hành trên ít nhất một nền tảng (iOS hoặc Android).

**10. CÁC YẾU TỐ THÀNH CÔNG QUAN TRỌNG**

* Sự hợp tác chặt chẽ giữa đội ngũ phát triển Backend (Võ Anh Kiệt) và Frontend (Đặng Dương Nguyên).
* Giao tiếp hiệu quả và thường xuyên giữa các thành viên trong dự án.
* Phản hồi và điều chỉnh kịp thời dựa trên các thay đổi yêu cầu hoặc vấn đề kỹ thuật phát sinh.
* Hiểu rõ yêu cầu: Nắm vững các yêu cầu và quy trình nghiệp vụ ngay từ đầu.
* Kiểm thử thường xuyên: Kiểm tra từng API ngay sau khi hoàn thành bằng Postman để phát hiện lỗi sớm.
* Quản lý mã nguồn tốt: Sử dụng Git để lưu trữ các phiên bản và dễ dàng quay lại khi cần.
* Thiết kế tốt: Cấu trúc code và cơ sở dữ liệu được thiết kế rõ ràng, dễ mở rộng (ví dụ: tách biệt Service, Controller, Repository).
* Bám sát phạm vi dự án: Tránh thêm các chức năng không cần thiết gây ảnh hưởng đến tiến độ chung.

**11.0 SIGNOFF**

|  |  |  |  |
| --- | --- | --- | --- |
| Project Member | Role | Signature | Date |
| Đặng Dương Nguyên | Project Manager / Flutter Developer | .......................... | ....../....../2025 |
| Võ Anh Kiệt | Backend Developer | .......................... | ....../....../2025 |
| Giảng viên hướng dẫn (Thầy Huynh) |  |  |  |