|  |  |
| --- | --- |
|  | **BỘ CÔNG THƯƠNG**  **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**  **---------------------------------------** |
| **PHẠM QUANG HUY** | **ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC** |
| **NGÀNH HỆ THỐNG THÔNG TIN** |
| **TÌM HIỂU VÀ XÂY DỰNG HỒ SƠ BỆNH ÁN ĐIỆN TỬ BỆNH VIỆN HỒNG NGỌC** |
| **HỆ THỐNG THÔNG TIN** | **GVHD: TS. Lê Trường Giang** |
| **Sinh viên: Phạm Quang Huy** |
| **Mã số sinh viên: 2019600247** |
|  |
|  | **Hà Nội – Năm 2024** |

**LỜI CẢM ƠN**

Đầu tiên, em xin gửi lời cảm ơn sâu sắc tới thầy TS. Lê Trường Giang - Giảng viên Trường Đại Học Công Nghiệp Hà Nội, người đã tạo điều kiện thực hiện nghiên cứu và hướng dẫn em rất tận tình trong suốt quá trình làm đồ án tốt nghiệp. Cảm ơn sự giúp đỡ nhiệt tình, đóng góp ý kiến của những anh chị, đồng nghiệp tại công ty DigiHC. Cuối cùng, em xin gửi lời cảm ơn đến gia đình, bạn bè đã luôn bên cạnh quan tâm, động viên, tạo động lực cho em hoàn thành tốt đồ án tốt nghiệp.

**TÓM TẮT NỘI DUNG ĐỒ ÁN**

Theo khoản 1 Điều 59 Luật Khám bệnh, chữa bệnh 2009, mỗi người bệnh chỉ có một hồ sơ bệnh án trong mỗi lần khám bệnh, chữa bệnh tại cơ sở khám bệnh, chữa bệnh. Hồ sơ bệnh án này có thể được lập bằng giấy hoặc bản điện tử. Tuy nhiên, trong thực tế hiện nay, hồ sơ bệnh án truyền thống dựa trên giấy tờ vẫn được sử dụng rộng rãi ở đa số các bệnh viện. Điều này gây ra nhiều khó khăn trong việc truy cập, chia sẻ thông tin và tìm kiếm dữ liệu của bệnh nhân.

Vì vậy trong đồ án này, em lựa chọn sử dụng công nghệ web bình thường để xây dựng một hệ thống hồ sơ bệnh án điện tử theo quy định trong thông tư 46 của Bộ y tế ban hành năm 2018. Sự lựa chọn này được đưa ra dựa trên tính tiện lợi và phổ biến của công nghệ web, cho phép truy cập và sử dụng trên nhiều nền tảng và thiết bị. Giải pháp của em sẽ tạo ra một giao diện web dễ sử dụng cho các cơ sở y tế để ghi lại, lưu trữ và truy cập thông tin bệnh án.

Đóng góp chính của đồ án này là xây dựng một hệ thống hồ sơ bệnh án điện tử sử dụng công nghệ web, giúp cải thiện quản lý thông tin bệnh án và tăng cường chất lượng chăm sóc sức khỏe. Kết quả dự kiến là tạo ra một công cụ hiệu quả để lưu trữ và truy xuất thông tin bệnh án, đồng thời tối ưu hóa quy trình làm việc và tăng cường sự liên kết giữa các cơ sở y tế.

**MỤC LỤC**

[**CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI** 1](#_Toc167612150)

[**1.1** **Đặt vấn đề** 1](#_Toc167612151)

[**1.2** **Vài nét về HIS và EMR** 1](#_Toc167612152)

[**1.2.1** **HIS (Hospital Information System)** 1](#_Toc167612153)

[**1.2.2** **EMR (Electronic Medical Record)** 2](#_Toc167612154)

[**1.3** **Mục tiêu và phạm vi đề tài** 3](#_Toc167612155)

[**1.4** **Định hướng giải pháp** 4](#_Toc167612156)

[**1.5** **Bố cục đồ án** 5](#_Toc167612157)

[**CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU** 7](#_Toc167612158)

[**2.1 Khảo sát hiện trạng** 7](#_Toc167612159)

[**2.2 Tổng quan chức năng** 7](#_Toc167612160)

[**2.2.1 Biểu đồ use case tổng quát** 7](#_Toc167612161)

[2.2.2 Biểu đồ use case phân rã: Tìm kiếm bệnh nhân 8](#_Toc167612162)

[2.2.3 Biểu đồ use case phân rã: Quản lý bệnh án 9](#_Toc167612163)

[**2.2.4 Biểu đồ use case phân rã: Quản lý giấy tờ liên quan** 9](#_Toc167612164)

[**2.2.5 Quy trình nghiệp vụ** 10](#_Toc167612165)

[**2.3 Đặc tả chức năng** 11](#_Toc167612166)

[**2.3.1 Đặc tả use case: Tìm bệnh nhân** 11](#_Toc167612167)

[**2.3.2 Đặc tả use case: Tạo bệnh án** 12](#_Toc167612168)

[**2.3.3 Đặc tả use case: Tạo giấy tờ liên quan** 12](#_Toc167612169)

[**2.4 Yêu cầu phi chức năng** 13](#_Toc167612170)

[**2.4.1 Yêu cầu về kĩ thuật** 13](#_Toc167612171)

[**2.4.2 Yêu cầu về UI/UX** 14](#_Toc167612172)

[**2.4.3 Yêu cầu về bảo mật và quyền riêng tư** 14](#_Toc167612173)

[**2.4.4** **Yêu cầu thêm khả năng mở rộng** 14](#_Toc167612174)

[**2.5** **Yêu cầu chức năng** 15](#_Toc167612175)

[**2.5.1 Yêu cầu quản lý thông tin bệnh nhân** 15](#_Toc167612176)

[**2.5.2 Yêu cầu quản lý quá trình khám bệnh** 15](#_Toc167612177)

[**CHƯƠNG 3: CÔNG NGHỆ SỬ DỤNG** 17](#_Toc167612178)

[**3.1 Front-end** 17](#_Toc167612179)

[**3.1.1 Angular** 17](#_Toc167612180)

[**3.1.2 Angular Material** 19](#_Toc167612181)

[**3.2 Back-end** 20](#_Toc167612182)

[**3.2.1 Golang** 20](#_Toc167612183)

[**3.2.2 MongoDB** 21](#_Toc167612184)

[**CHƯƠNG 4: THỰC NGHIỆM VÀ ĐÁNH GIÁ** 24](#_Toc167612185)

[**4.1 Thiết kế kiến trúc** 24](#_Toc167612186)

[**4.1.1 Lựa chọn kiến trúc phần mềm** 24](#_Toc167612187)

[**4.1.2 Thiết kế tổng quan** 26](#_Toc167612188)

[**4.2 Thiết kế chi tiết** 27](#_Toc167612189)

[**4.2.1 Thiết kế giao diện** 27](#_Toc167612190)

[**4.2.2 Thiết kế cơ sở dữ liệu** 29](#_Toc167612191)

[**4.3 Xây dựng ứng dụng** 29](#_Toc167612192)

[**4.3.1 Thư viện và công cụ sử dụng** 29](#_Toc167612193)

[**4.3.2 Kết quả đạt được** 30](#_Toc167612194)

[**4.3.3 Minh họa các chức năng chính** 30](#_Toc167612195)

[**4.4 Kiểm thử** 38](#_Toc167612196)

[**4.5 Triển khai** 42](#_Toc167612197)

[**4.5.1 Kết thúc triển khai** 43](#_Toc167612198)

[**4.5.2 Tóm tắt về thông số kỹ thuật của HIS** 45](#_Toc167612199)

[**CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN** 46](#_Toc167612200)

[**5.1 Kết luận** 47](#_Toc167612201)

[**5.2 Hướng phát triển** 49](#_Toc167612202)

**DANH MỤC HÌNH VẼ**

[Hình 2.1: Biểu đồ use case tổng quát 7](#_Toc167611870)

[Hình 2.2: Biểu đồ use case tìm kiếm bệnh nhân 8](#_Toc167611871)

[Hình 2.3: Biểu đồ use case quản lý bệnh án 9](#_Toc167611873)

[Hình 2.4: Biểu đồ use case quản lý giấy tờ liên quan 9](#_Toc167611874)

[Hình 2.5: Quy trình nghiệp vụ hồ sơ bệnh án điện tử 10](#_Toc167611875)

[Hình 2.6: Biểu đồ chức năng hệ thống 16](#_Toc167611876)

[Hình 3.1: Logo Angular 18](#_Toc167611877)

[Hình 3.2: Logo Angular material 19](#_Toc167611878)

[Hình 3.3: Logo Golang 21](#_Toc167611879)

[Hình 3.4: Logo MongoDB 22](#_Toc167611880)

[Hình 4.1: Biểu đồ gói tổng quát 26](#_Toc167611881)

[Hình 4.2: Thiết kế chi tiết gói 27](#_Toc167611882)

[Hình 4.3 mô tả bố cục giao diện trang quản lý hồ sơ bệnh án và giấy tờ liên quan. 28](#_Toc167611883)

[Hình 4.4: Mô hình giao diện trang hồ sơ bệnh án 28](#_Toc167611884)

[Hình 4.5: Cơ sở dữ liệu hệ thống 29](#_Toc167611885)

[Hình 4.6: Tìm kiếm bệnh nhân 30](#_Toc167611886)

[Hình 4.7: Chức năng tạo bệnh án 31](#_Toc167611887)

[Hình 4.8: Chức năng tạo bệnh án 31](#_Toc167611888)

[Hình 4.9: Chức năng tạo bệnh án 32](#_Toc167611889)

[Hình 4.10: Xem bệnh án 33](#_Toc167611890)

[Hình 4.11: Xem bệnh án 33](#_Toc167611891)

[Hình 4.12: Xem giấy tờ liên quan 34](#_Toc167611892)

[Hình 4.13: In phiếu, xuất file PDF 35](#_Toc167611893)

[Hình 4.14: Kiến trúc triển khai 44](#_Toc167611894)

**DANH MỤC BẢNG BIỂU**

[Bảng 2.1: Đặc tả use case tìm kiếm bệnh nhân 10](#_Toc167112646)

[Bảng 2.2: Đặc tả use case tạo bệnh án 10](#_Toc167112647)

[Bảng 2.3: Đặc tả use case tạo bệnh án 11](#_Toc167112648)

[Bảng 4.1: Danh sách thư viện và công cụ đã sử dụng 23](#_Toc167112649)

[Bảng 4.2: Kiểm thử chức năng 28](#_Toc167112650)

**DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT**

|  |  |
| --- | --- |
| **Thuật ngữ** | **Ý nghĩa** |
| API  BHXH/BHYT BMI  BN  BYT CĐHA CSDL  CT  DICOM  DNS  EMR  HIS  HL7  HTML  HTTT  ID  IT, CNTT  KCB  LIS  MRI  PACS  RIS  TT | Giao diện lập trình ứng dụng (Application Programming Interface)  Bảo hiểm xã hội/Bảo hiểm y tế  Chỉ số thể trọng (Body mass index) Bệnh nhân  Bộ Y tế  Chẩn đoán hình ảnh Cận lâm sàng  Cơ sở dữ liệu  Chụp cắt lớp vi tính  Tiêu chuẩn xử lý dữ liệu  Hệ thống phân giải tên miền (Domain Name System)  Hồ sơ bệnh án điện tử  Hệ thống thông tin quản lý bệnh viện Tiêu chuẩn quốc tế cung cấp giao thức chuẩn về quản lý, trao đổi và tích hợp dữ liệu y tế giữa các hệ thống thông tin y tế nhằm hỗ trợ các hoạt động y tế  (Health Level Seven).  Ngôn ngữ đánh dấu siêu văn bản (HyperText Markup Language)  Hệ thống thông tin  Mã định danh  Công nghệ thông tin  Khám chữa bệnh  Phần mềm quản lý thông tin khoa Xét nghiệm (XN)  Chụp cộng hưởng từ  Hệ thống thống lưu trữ và truyền thông hình ảnh y tế  Hệ thống thông tin x quang  Thông tư |

# **CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI**

* 1. **Đặt vấn đề**

Ngày nay công nghệ thông tin đang ngày càng phát triển nhanh chóng, kéo theo đó là hệ thống mạng các phần mềm cũng gia tăng cả về số lượng theo quy mô mở rộng và cả về chất lượng phần mềm theo chiều sâu.

Trong lĩnh vực y tế, ứng dụng CNTT ngày càng phát triển mạnh mẽ, các hệ thống thông tin y tế HIS, RIS, LIS, PACS được triển khai ứng dụng rộng rãi và hiệu quả, xây dựng dựa trên tiêu chuẩn DICOM và HL7 nhằm hướng tới thống nhất về trao đổi và xử lý thông tin dữ liệu giữa các cơ sở y tế phục vụ công tác quản lý, chẩn đoán, điều trị và chăm sóc sức khỏe.

Tuy nhiên, hiện nay việc triển khai EMR(hồ sơ bệnh án điện tử) tại các bệnh viện còn rất hạn chế. Hồ sơ bệnh án truyền thống dựa vào giấy tờ và hình thức vật lý vẫn đang được sử dụng rộng rãi trong các cơ sở chăm sóc sức khỏe. Nó gây ra nhiều khó khăn cho việc quản lý, lưu trữ, truy xuất thông tin trong hồ sơ bệnh án, tiêu tốn khá nhiều không gian lưu trữ hồ sơ bệnh án. Những vấn đề như mất mát, đánh mất thông tin, khó khăn trong tìm kiếm và chia sẻ thông tiềm ẩn nguy cơ sai sót trong chẩn đoán và điều trị.

Việc triển khai thành công hồ sơ bệnh án điện tử mang lại nhiều lợi ích quan trọng như: cho phép lưu trữ thông tin y tế của bệnh nhân dưới dạng điện tử, giúp tiết kiệm thời gian, không gian lưu trữ. Thông tin ý tế của bệnh nhân có thể dễ dàng truy xuất từ bất kỳ địa phương nào bởi những người có quyền truy cập. Điều này giúp tăng cường tính khả dụng và hiệu quả trong quản lý thông tin y tế.

* 1. **Vài nét về HIS và EMR**
     1. **HIS (Hospital Information System)**

Là hệ thống thông tin bệnh viện thường được biết đến với tên gọi khác là "Hệ thống quản lý bệnh viện"; phục vụ công tác quản lý, điều hành tại Bệnh viện với các chức năng chính: quản lý thông tin bệnh nhân và bệnh sử; quản lý bệnh nhân đến khám và điều trị nội và ngoại trú, quản lý bệnh án, dược, tài chính, viện phí, trang thiết bị vật tư y tế, nhân sự... Ngày nay, HIS là công cụ tối ưu hóa trong quản lý điều hành; phục vụ điều trị; phục vụ nghiên cứu và đào tạo; thống kê, dự báo, dự phòng... tại các Bệnh viện.

* + 1. **EMR (Electronic Medical Record)**

Theo Điều 59 Luật Khám bệnh, chữa bệnh 2009, hồ sơ bệnh án là tài liệu y học, y tế và pháp lý; mỗi người bệnh chỉ có một hồ sơ bệnh án trong mỗi lần khám bệnh, chữa bệnh tại cơ sở khám bệnh, chữa bệnh. Người bệnh điều trị nội trú và ngoại trú trong các cơ sở khám bệnh, chữa bệnh đều phải được lập hồ sơ bệnh án. Hồ sơ bệnh án được lập bằng giấy hoặc bản điện tử và phải được ghi rõ, đầy đủ các mục có trong hồ sơ bệnh án. Hồ sơ bệnh án bao gồm các tài liệu, thông tin liên quan đến người bệnh và quá trình khám bệnh, chữa bệnh.

Điều 2 Thông tư 46/2018/TT-BYT cũng quy định hồ sơ bệnh án điện tử được lập, cập nhật, hiển thị, ký số, lưu trữ bằng phương tiện điện tử đáp ứng các quy định của Thông tư 46/2018/TT-BYT thì có giá trị pháp lý như hồ sơ bệnh án giấy quy định tại Điều 59 Luật Khám bệnh, chữa bệnh 2009.

Nói về một bệnh án điện tử

* Thứ nhất, căn cứ dựa trên văn bản của pháp luật: thì văn bản mức cao nhất là thông tư (thông tư 54/2017/TT-BYT– phần quy định về tiêu chí bệnh án điện tử và thông tư 46/2018/TT-BYT: quy định về Hồ sơ bệnh án điện tử).
* Thứ 2, mức độ thấp hơn thì có các quyết định: Hiện tại có 3 quyết định của bộ y tế là đưa ra các mẫu biểu về bệnh án, về lưu trữ hồ sơ bệnh án. Đó là các quyết định:(i)4069/2001/QĐ-BYT ngày 28/09/2001, (ii)QĐ 3443/QĐ- BYT ngày 21/09/2011, (iii)QĐ 3730 BYT ngày 5/8/2021. 3 quyết định này quy định rõ và mô tả rõ chi tiết toàn bộ các hồ sơ bệnh án bây giờ cần lưu trữ theo mẫu nào dựa trên các quyết định đấy.

Về căn cứ pháp luật thì phải đáp ứng theo các căn cứ pháp luật, để thiết kế ra phần mềm thay vì việc lưu trữ hồ sơ bệnh án bằng giấy theo đúng các thông tư và quyết định thì bây giờ chuyển lên trên máy.

Vậy nếu chuyển lên trên máy thì có nghĩa là phải làm được gì? Tức là bác sĩ không phải dùng tay để viết mà mọi thứ là phải thao tác trên máy tính: in ra trên máy, ký số được trên máy và lưu trữ được, chia sẻ được thì cuối cùng nó sẽ đáp ứng được tiêu chuẩn của bệnh án điện tử theo thông tư 54.

Vậy nên, mục tiêu lớn nhất của EMR là số hóa được toàn bộ hồ sơ bệnh án giấy tờ. Tức là làm mọi thứ trên máy, không phải đánh lại thông tin đã nhập và thông tin hành chính của bệnh nhân. Ký số được, tổng kết được, chia sẻ được cho những đơn vị liên quan trong đó có người bệnh, bảo hiểm, bác sĩ.

* 1. **Mục tiêu và phạm vi đề tài**

Như em đã trình bày ở phần 1.1, tuy đã có thông tư ban hành về quy định xây dựng hồ sơ bệnh án điện tử, nhưng hiện nay số bệnh viện đã triển khai hồ sơ bệnh án điện tử vẫn còn rất ít. Do gặp phải những khó khăn như cơ sở dữ liệu của các bệnh viện chưa được đồng bộ, chưa thống nhất giữa các bệnh viện nên việc kết nối liên thông dữ liệu còn gặp nhiều khó khăn. Theo thông tư số 46/2018/TT-BYT, đến năm 2030 tất cả các cở sở khám bệnh, chữa bệnh trên toàn quốc phải triển khai được hồ sơ bệnh án điện tử.

Để giải quyết vấn đề trên, tham gia vào thúc đẩy triển khai hồ sơ bệnh án điện tử, Giám đốc kĩ thuật của công ty em đã quyết định sản xuất một module EMR để quán lý các hồ sơ bệnh án, giấy tờ liên quan đến bệnh án của bệnh nhân.

Hiện tại, module này sẽ được tích hợp chặt chẽ vào phần mềm HIS đã triển khai của công ty. Quyết định này được đưa ra do tính mạnh mẽ của kiến trúc HIS ( kiến trúc của HIS sẽ được trình bày trong phần...), cho phép nó phục vụ được số lượng lớn các cơ sở khám chữa bệnh.

Do đó, với một nhóm gồm một người phát triển phần mềm là em, một người hỗ trợ và một IT BA (đồng thời là người thiết kế). Nhóm chúng em phải phát triển một module giữ lại được những tính năng quan trọng nhất của hệ thống cũ, đồng thời thực hiện cải tiến nó. Mục tiêu hướng đến đó là xây dựng một trang web thân thiện với người dùng, có thể sử dụng trên cả màn hình máy tính bàn, laptop, ipad. Với các chức năng chính là:

* Cho phép người dùng tìm kiếm thông tin bệnh nhân nội/ngoại trú
* Chức năng bấm để xem thông tin chi tiết bệnh nhân’
* Chức năng tạo mới một bộ hồ sơ bệnh án
* Cho phép chọn loại hồ sơ bệnh án
* Xem các bệnh án chưa ký
* Xem các bệnh án đã ký
* Chức năng tạo mới các phiếu trong bệnh án
* Chức năng lưu phiếu
* Chức năng in bệnh án
* Chức năng View để xem chi tiết các loại bệnh án/ giấy tờ
* Chức năng xóa phiếu trong bệnh án
  1. **Định hướng giải pháp**

Từ yêu cầu vấn đề trong mục [1.3](#_bookmark6) em đề xuất định hướng giải pháp cho một số vấn đề như sau có trong bảng 1.1

|  |  |  |
| --- | --- | --- |
| **Yêu cầu** | **Vấn đề cần giải quyết** | **Hướng giải quyết** |
| Giao diện | Giao diện phải đảm bảo tính trực quan, dễ sử dụng cho người dùng. | Tham khảo và thiết kế giao diện có bố cục giống như HIS. |
| Quản ý hồ sơ bệnh án | Phải có đủ các trường thông tin như bệnh án giấy. | Thiết kế biểu mẫu bệnh án theo biểu mẫu của bộ y tế. |
| Quản lý giấy tờ liên quan | Phải tổng hợp được đủ các giấy tờ liên quan trong quá trình điều trị. | Liên kết chặt chẽ HIS nhằm liên kết và lấy các dữ liệu trong quá trình khám chữa bệnh. |

* 1. **Bố cục đồ án**

Báo cáo đồ án này sẽ được sắp xếp thành 6 chương chính và nội dung của các chương được trình bày lần lượt như sau: Trong chương 1: là giới thiệu đề tài, tại đây em trình bày các vấn đề cần giải quyết mà đồ án có thể thực hiện phù hợp với thực tiễn trong cuộc sống. Dựa cơ sở đó em tìm hiểu các vấn đề và đưa ra hướng phát triển hợp lý mà bản thân em thực hiện để hoàn thành đồ án tốt nghiệp 1 cách trọn vẹn và tốt nhất.

Trong chương 2, em sẽ trình bày về tình hình thực tiễn bản thân em đã có nghiên cứu khảo sát nhu cầu của doanh nghiệp. Do đó em đã xác định tất cả nhược điểm, hạn chế còn tồn tại để xây dựng nên website của mình một cách hiệu quả nhất. Tiếp đó sẽ trình bày tổng quan chức năng của website và đặc tả các chức năng chính của hệ thống bằng các biểu đồ use case để toàn diện nhất em đưa ra các yêu cầu phi chức năng của hệ thống.

Trong chương 3, em sẽ trình bày về những công nghệ bản thân em đã áp dụng để xây dựng phần mềm hệ thống của mình. Để có thể hoàn thành đồ án một cách thuận lợi nhất, em cũng đã cố gắng lựa chọn được một số giải pháp công nghệ riêng bản thân thông qua những gợi ý của các anh chị trong công ty để đáp ứng nhu cầu cần thiết để phát triển sản phẩm một cách hoàn chỉnh nhất.

Trong chương 4 em sẽ triển khai website và trình bày kết quả thực nghiệm quá trình phát triển. Trong đó các quá trình được trình bày theo từng giai đoạn sau:(i)Thiết kế hệ thống, (ii)Thiết kế chi tiết hệ thống, (iii) Xây dựng ứng dụng, (iv)Kiểm thử và triển khai sản phẩm.

Cuối cùng trong chương 5 em xin được kết luận nội dung của đồ án tốt nghiệp, những vấn đề và những điều mà bản thân đã cố gắng làm được cũng như chưa làm được trong quá trình làm đồ án tốt nghiệp và hướng phát triển sản phẩm của bản thân trong tương lai 1 cách quy mô nhất.

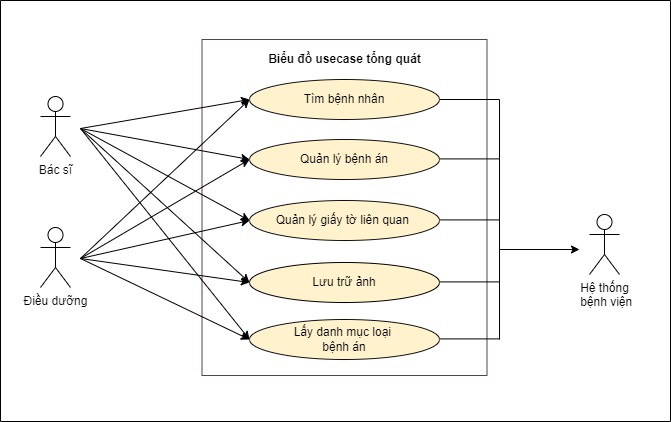
# **CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU**

## **2.1 Khảo sát hiện trạng**

Như đã đề cập ở phần [1.1,](#_bookmark2) thì hiện nay việc ứng dụng bệnh án điện tử tại các cơ sở khám chữa bệnh vẫn còn khá hạn chế. Vì việc triển khai này còn khá mới mẻ, nên có rất ít thông tin về công nghệ sử dụng, cũng như quy trình nghiệp vụ của bệnh án điện tử trong các bệnh viện này. Theo tìm hiểu của em, các bệnh viện đã triển khai thì đa số họ tách riêng việc quản lý giấy tờ và hồ sơ bệnh án. Vì vậy, trong đồ án này, nhóm chúng em sẽ thiết kế ra một module có thể quản lý cả hồ sơ bệnh án và tất cả các giấy tờ liên quan của bệnh án, để nó trở thành một hồ sơ bệnh án điện tử theo đúng yêu cầu của bộ y tế đề ra. Dưới đây em sẽ trình bày về tổng quan chức của module này.

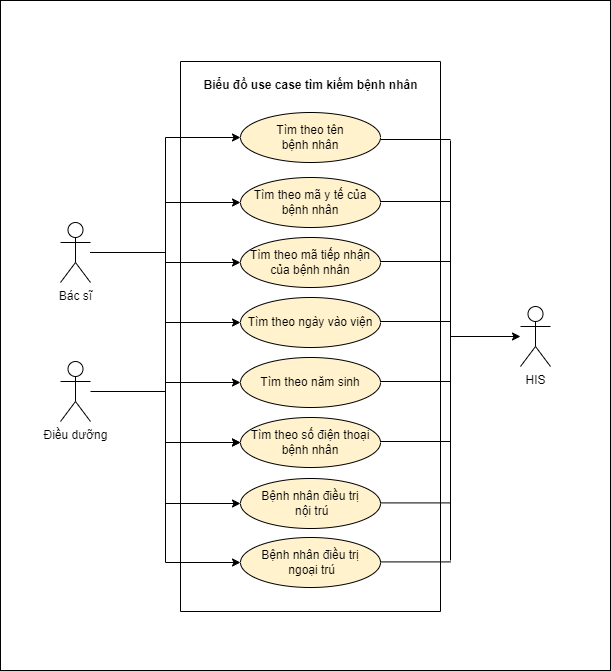
## **2.2 Tổng quan chức năng**

### **2.2.1 Biểu đồ use case tổng quát**

Thông qua các phân tích ở trên, các chức năng của hệ thống có thể được biểu diễn bằng một biểu đồ use case như Hình [2.1:](#_bookmark13)

Hình 2.1: Biểu đồ use case tổng quát

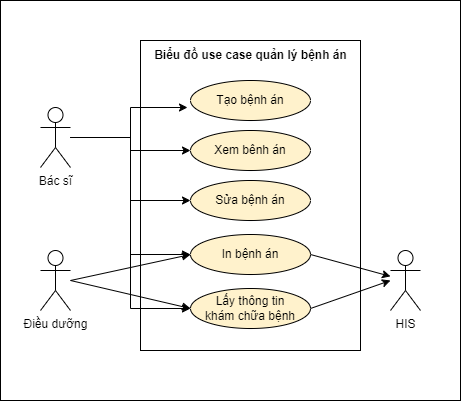
### 2.2.2 Biểu đồ use case phân rã: Tìm kiếm bệnh nhân



Hình 2.2: Biểu đồ use case tìm kiếm bệnh nhân

Hình 2.2 mô tả các trường hợp tìm kiếm bệnh nhân. Bác sĩ và điều dưỡng sẽ chọn tìm kiếm bệnh nhân điều trị nội trú hoặc ngoại trú. Sau đó có thể tìm kiếm bệnh nhân theo thông tin hành chính của bệnh nhân như: tên, năm sinh, số điện thoại, ngày vào viện , mã y tế hoặc mã tiếp nhận của bệnh nhân. Những thông tin này của bệnh nhân sẽ được lấy HIS.

### 2.2.3 Biểu đồ use case phân rã: Quản lý bệnh án



Hình 2.3: Biểu đồ use case quản lý bệnh án

Hình [2.3](#_bookmark17) mô tả chức năng quản lý bệnh án. Bác sĩ có quyền tạo mới bệnh án dựa trên nhưng thông tin khám chữa bệnh từ HIS. Sau đó điều đưỡng có thể xem bệnh án và in bệnh án lưu

### **2.2.4 Biểu đồ use case phân rã: Quản lý giấy tờ liên quan**

Hình 2.4: Biểu đồ use case quản lý giấy tờ liên quan

Hình [2.4](#_bookmark19) mô tả chức năng quản lý các giấy tờ liên quan đến bệnh án. Bác sĩ và điều dưỡng lấy kết quả khám chữa bệnh từ HIS hoặc tạo mới giấy tờ liên quan, sau đó có thể xem và in giấy các giấy tờ liên quan của bệnh án.

### **2.2.5 Quy trình nghiệp vụ**

Hình 2.5: Quy trình nghiệp vụ hồ sơ bệnh án điện tử

Hình [2.5](#_bookmark21) mô tả quy trình nghiệp vụ làm hồ sơ bệnh án điện tử. Ban đầu, bệnh nhân đến viện sẽ được nhân viên y tế tiếp nhận và chỉ định các dịch vụ. Sau đó bệnh nhân sẽ được tiến hành khám bệnh theo chỉ định. Nếu bệnh nhân không có bệnh hoặc có bệnh mà không muốn điều trị thì sẽ ra viện mà không cần làm hồ sơ bệnh án. Nếu bệnh nhân có bệnh và muốn điều trị thì sẽ được tạo hồ sơ bệnh án và tiến hành các thủ tục nhập viện, điều trị bệnh. Trong quá trình điều trị bệnh sẽ đồng thời làm các giấy tờ liên quan như giấy tờ siêu âm, chụp chiếu, phẫu thuật... Quá trình này được thực hiện và lưu lại trên HIS. Đến khi bệnh nhân kết thúc quá trình điều trị hoặc ra viện dưới hình thức điều trị ngoại trú hoặc chuyển viện, bác sĩ sẽ nhập các thông tin khám chữa bệnh của bệnh nhân vào hồ sơ bệnh án. Tổng hợp các giấy tờ liên quan trong quá trình khám chữa bệnh vào hồ sơ bệnh án của bệnh nhân.

**2.3 Đặc tả chức năng**

**2.3.1 Đặc tả use case: Tìm bệnh nhân**

|  |  |  |
| --- | --- | --- |
| Tên Use case | Tìm kiếm bệnh nhân | |
| Tác nhân | Bác sĩ, Điều dưỡng | |
| Tiền điều kiện | 1. Tác nhân đã đăng nhập hệ thống 2. Bệnh nhân đã tồn tại trong HIS 3. Bệnh nhân đã được tạo bệnh án từ HIS | |
| Hậu điều kiện | Hiển thị kết quả tìm kiếm bệnh nhân theo yêu cầu | |
| Luồng sự kiện chính | Tác nhân | Hệ thống |
| 1. Tác nhân truy cập vào   trang làm hồ sơ bệnh án | 2. Hệ thống hiển thị trang kết quả với bọ lọc tìm kiếm |
| 3. Tác nhân tìm kiếm theo  điều kiện | 4. Hệ thống hiển thị kết quả tìm kiếm |
| Luồng sự kiện thay thế | Không | Không |

Bảng 2.1: Đặc tả use case tìm kiếm bệnh nhân

**2.3.2 Đặc tả use case: Tạo bệnh án**

|  |  |  |
| --- | --- | --- |
| Tên Use case | Tạo mới bệnh án | |
| Tác nhân | Bác sĩ | |
| Tiền điều kiện | Tác nhân đã chọn bệnh nhân sau khi tìm kiếm | |
| Hậu điều kiện | Một hồ sơ bệnh án mới được tạo ra | |
| Luồng sự kiện chính | Tác nhân | Hệ thống |
| 1. Tác nhân bấm chọn bệnh  nhân sau khi tìm kiếm | 2. Hệ thống hiển thị biểu  mẫu hồ sơ bệnh án |
| 3. Tác nhân nhập dữ liệu  cho các trường dữ liệu trong biểu mẫu | 4. Hệ thống lưu dữ liệu vào  biểu mẫu |
| 5. Tác nhân bấm lưu | 6.Hệ thống lưu hồ sơ bệnh  án vào cơ sở dữ liệu và chọn hồ sơ bệnh án mới tạo |
| Luồng sự kiện thay thế | Không | Không |

Bảng 2.2: Đặc tả use case tạo bệnh án

**2.3.3 Đặc tả use case: Tạo giấy tờ liên quan**

|  |  |  |
| --- | --- | --- |
| Tên Use case | Tạo giấy tờ liên quan | |
| Tác nhân | Bác sĩ, điều dưỡng | |
| Tiền điều kiện | 1. Tác nhân đã chọn bệnh nhân sau khi tìm kiếm 2. Bệnh nhân đã được tạo hồ sơ bệnh án và lưu lại | |
| Hậu điều kiện | Một số giấy tờ liên quan được tạo | |
| Luồng sự kiện chính | Tác nhân | Hệ thống |
| 1. Tác nhân bấm chọn bệnh   nhân sau khi tìm kiếm và chọn chế độ xem các bệnh án chưa ký | 2. Hệ thống hiển thị danh  sách cách giấy tờ liên quan cần lầm |
| 3. Tác nhân chọn loại giấy  tờ liên quan cần làm | 4. Hệ thống hiển thị ra biểu  mẫu của giấy tờ |
| 5. Tác nhân nhập dữ liệu  cho các trường dữ liệu cần thiết trong biểu mẫu | 6.Hệ thống lưu dữ liệu vào  biểu mẫu |
| 7. Tác nhân bấm lưu | 4. Hệ thống lưu thông tin  giấy tờ liên quan vào cơ sở dữ liệu |
| Luồng sự kiện thay thế | Không | Không |

Bảng 2.3: Đặc tả use case tạo bệnh án

**2.4 Yêu cầu phi chức năng**

**2.4.1 Yêu cầu về kĩ thuật**

Hệ thống phải có thể sử dụng được, có thể bảo trì và có thể mở rộng, dễ chuyển đổi cơ sở dữ liệu

**2.4.2 Yêu cầu về UI/UX**

Giao diện phải thiết kế đơn giản, dễ tiếp cận với nhân viên y tế, đồng thời phải thiết kế tương tự hệ thống HIS mà bệnh viện đang sử dụng.Màu chủ đề nên có thể thay đổi tùy thuộc vào từng cơ sở khám chữa bệnh. Có thể tương thích trên nhiều trình duyệt và nhiều màn hình khác nhau.

**2.4.3 Yêu cầu về bảo mật và quyền riêng tư**

Dữ liệu y tế của bệnh nhân phải được mã hóa cả khi lưu trữ và khi truyền di để đảm bảo không bị truy cập trái phép. Hệ thống cần phải quản lý và phân quyền truy cập cho từng người dùng dựa trên vai trờ của họ(bác sĩ, y tá, quản trị viên…). Tất cả các hoạt động truy cập và chỉnh sửa dữ liệu phải được ghi lại để theo dõi và kiểm tra khi cần thiết. Hệ thống phải sử dụng các phương pháp xác thực mạnh mẽ (như xác thực hai yếu tố) để đảm bảo chỉ những người dùng được ủy quyền mới có thể truy cập vào hệ thống. Đảm bảo rằng tất cả dữ liệu được truyền qua mạng phải được bảo mật, chẳng hạn như sử dụng giao thức HTTPS để mã hóa dữ liệu truyền tải. Hệ thống phải có cơ chế sao lưu dữ liệu định kỳ và khả năng khôi phục dữ liệu nhanh chóng trong trường hợp mất mát hoặc sự cố.

* + 1. **Yêu cầu thêm khả năng mở rộng**

Hệ thống nên được thiết kế theo kiến trúc mô-đun, cho phép thêm, bớt hoặc nâng cấp các thành phần mà không ảnh hưởng đến toàn bộ hệ thống. Mỗi mô-đun có thể đảm nhận một chức năng cụ thể như quản lý thông tin bệnh nhân, quản lý lịch sử khám bệnh, quản lý thuốc, v.v. Triển khai hệ thống trên nhiều máy chủ và trung tâm dữ liệu để đảm bảo tính sẵn sàng cao và phân phối tải. Sử dụng các hệ quản trị cơ sở dữ liệu có khả năng mở rộng ngang như Apache Cassandra, MongoDB, hoặc Google Bigtable. Thực hiện các bài kiểm thử tải để đánh giá hiệu suất hệ thống dưới các điều kiện tải khác nhau. Thực hiện các bài kiểm thử căng thẳng để đánh giá giới hạn tối đa của hệ thống và phát hiện các điểm nghẽn tiềm ẩn.Đảm bảo hệ thống có thể phục hồi sau các tình huống tải cực đại mà không gặp sự cố nghiêm trọng.

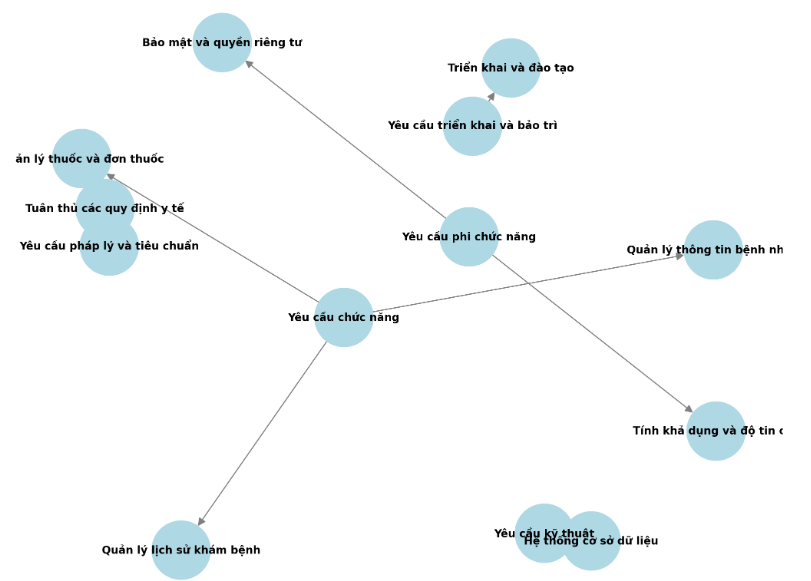
* 1. **Yêu cầu chức năng**

### **2.5.1 Yêu cầu quản lý thông tin bệnh nhân**

* Hệ thống phải cho phép tạo, lưu trữ và quản lý hồ sơ bệnh nhân bao gồm thông tin cá nhân (tên, tuổi, giới tính, địa chỉ, số điện thoại, email).
* Hỗ trợ lưu trữ thông tin bảo hiểm y tế và các thông tin liên quan khác như số hồ sơ bảo hiểm, ngày hết hạn bảo hiểm.
* Lưu trữ tiền sử bệnh lý của bệnh nhân bao gồm các bệnh đã mắc, phẫu thuật đã thực hiện, tiền sử gia đình, và tiền sử xã hội.
* Quản lý thông tin về các dị ứng của bệnh nhân đối với thuốc, thực phẩm, và các yếu tố môi trường.
* Lưu trữ thông tin liên lạc khẩn cấp của bệnh nhân, bao gồm tên, quan hệ, số điện thoại, và địa chỉ của người liên hệ.

### **2.5.2 Yêu cầu quản lý quá trình khám bệnh**

* Ghi lại lịch sử điều trị của bệnh nhân, bao gồm các lần khám bệnh, chẩn đoán, điều trị, và kết quả xét nghiệm.
* Hỗ trợ lưu trữ và truy cập các tài liệu y khoa như báo cáo xét nghiệm, hình ảnh chẩn đoán, và ghi chú của bác sĩ.
* Hệ thống phải cho phép bác sĩ nhập và cập nhật chẩn đoán, kế hoạch điều trị và ghi chú trong quá trình khám bệnh.
* Hỗ trợ công cụ tra cứu mã bệnh ICD-10 để chuẩn hóa chẩn đoán và điều trị.
* Hỗ trợ lưu trữ và truy cập kết quả xét nghiệm, bao gồm cả các kết quả từ phòng xét nghiệm và các thiết bị chẩn đoán hình ảnh.
* Gửi thông báo cho bác sĩ khi có kết quả xét nghiệm mới và cho phép so sánh kết quả xét nghiệm qua các thời điểm khác nhau.



Hình 2.6: Biểu đồ chức năng hệ thống

Ở trên em đã giới thiệu tổng quan về phân tích hệ thống của chúng em. Trong chương tiếp theo, em sẽ đề cập đến các công nghệ mà chúng em chọn để sử dụng xây dựng module này

**CHƯƠNG 3: CÔNG NGHỆ SỬ DỤNG**

Chương 2 kết thúc với phần khảo sát và phần tích tổng quan về module quản lý hồ sơ bệnh án điện tử. Dưới đây, em sẽ trình bày ngắn gọn về các công nghệ mà em đã sử dụng cho dự án này.

Hệ thống này được xây dựng để hoạt động trên Web trong một mạng riêng của một cơ sở khám chữa bệnh. Các yêu cầu chỉ liên quan đến việc quản lý dữ liệu và một ứng dụng Web điển hình như vậy, bao gồm giao diện người dùng(front-end), máy chủ(back-end) và cơ sở dữ liệu (database), sẽ đủ giải quyết những vấn đề này.

**3.1 Front-end**

Front-end là phần giao diện người dùng (UI) của một ứng dụng Web, nơi người dùng có thể nhìn thấy và tương tác với dữ liệu của ứng dụng. Công cụ bắt buộc cho phát triển Web là ngôn ngữ đánh dấu Hyper-Text Markup Language (HTML) để cấu trúc và hiển thị các trang Web, thường được bổ sung bởi Cascading Style Sheets (CSS) để tùy chỉnh giao diện một cách trực quan. Để làm cho các trang trở nên tương tác, JavaScript (JS) được sử dụng để xử lý logic lập trình. Các công cụ chính cho các nhà phát triển front-end là HTML-CSS-JS.

Có nhiều công cụ cấp cao, gọi là các framework, được sử dụng để tăng tốc độ phát triển front-end. Trong dự án này, em đã sử dụng Angular (cụ thể là Angular 14), một framework phổ biến được phát triển và duy trì bởi Google và cộng đồng các nhà cộng tác.

**3.1.1 Angular**

Angular được xây dựng dựa trên một số quy tắc nghiêm ngặt, gần như không chấp nhận bất kỳ sai sót nào. Các thành phần của nó được cấu trúc thành các thành phần con theo một cấu trúc phân cấp, tương tự như cách HTML sắp xếp các thành phần. Điều này giúp tạo sự tổ chức và dễ quản lý cho ứng dụng. Các thành phần trong Angular có thể dễ dàng tạo và nhúng vào nhau, giúp chúng ta dễ dàng tái sử dụng chúng. Một tính năng tái sử dụng khác mà Angular cung cấp là "dependency injection" (DI): Nó cho phép ta tái sử dụng các dịch vụ từ các thành phần khác mà không gặp nhiều khó khăn. Điều này giúp chia sẻ dữ liệu toàn cục, giao tiếp với máy chủ hoặc thực hiện các chức năng tính toán phức tạp trên toàn dự án dễ dàng.



Hình 3.1: Logo Angular

Các thành phần và dịch vụ trong Angular, cùng với các phần khác, có thể được bao gói vào một module. Điều này giúp tạo tính mô-đun cho dự án - tất cả các chức năng của hệ thống có thể được tách ra và tái sử dụng trong một gói duy nhất, và chúng có thể được sử dụng lại trong các dự án Angular khác. Điều này giúp dự án dễ dàng phân chia thành các phần riêng biệt và dễ dàng tái sử dụng trong các dự án khác nhau. Sử dụng TypeScript đóng góp vào tính mô-đun của dự án này - TypeScript là một phần mở rộng của JavaScript hỗ trợ kiểu dữ liệu nghiêm ngặt. TypeScript giúp ứng dụng duy trì tính nhất quán và dễ bảo trì, đồng thời áp dụng tốt các quy tắc Lập trình Hướng đối tượng (OOP).

Framework này được lựa chọn do công ty của em quen thuộc với nó. Những quy tắc nghiêm ngặt của nó giúp giảm số lỗi biên dịch trong quá trình phát triển và dễ dàng phát hiện lỗi. Mặc dù khá khó để học, Angular có tài liệu rất đầy đủ về tất cả các chức năng có sẵn cho nhà phát triển.

Một số lựa chọn phổ biến khác thay thế Angular bao gồm các thư viện như React và các framework khác như Vue. React và Vue đơn giản hơn trong việc viết mã và cung cấp hiệu suất cao hơn, tuy nhiên, cả hai đều tuân theo kiến trúc tổng quát Model-View-Controller (MVC) giống như Angular (xem thêm về MVC trong phần**??**). React cũng cho phép xây dựng ứng dụng di động bằng cùng nguyên tắc của React dưới dạng React Native, có thể triển khai trên nhiều thiết bị, bao gồm cả thiết bị di động và truyền hình.

Angular cung cấp cho đội của chúng em một công cụ mạnh mẽ để xây dựng ứng dụng. Tuy nhiên, giao diện người dùng sẽ không chỉ được xây dựng bằng HTML mà sẽ bao gồm các công nghệ khác.

**3.1.2 Angular Material**

Angular Material là một thư viện giao diện người dùng (UI) cung cấp các thành phần hoàn toàn hỗ trợ Angular, với bố cục và kiểu dáng tuân theo các đặc tả thiết kế của Material Design. Thư viện này bao gồm các thành phần giao diện người dùng đa dạng như các biểu mẫu, thẻ điều hướng, nút, hộp thoại và trình phân trang, được tối ưu hóa về hiệu suất, đáng tin cậy và khả năng tiếp cận. Những thành phần này có thể dễ dàng nhúng vào trang Web theo ý muốn. Ngoài ra, Angular Material cũng hỗ trợ tính năng "theming", cho phép thay đổi chủ đề màu sắc, kiểu chữ văn bản và nhiều thứ khác trong dự án một cách dễ dàng hơn.

Hình 3.2: Logo Angular material

Angular Material là một lựa chọn tuyệt vời và tiện lợi cho đội của chúng em, vì chúng em đã làm việc với Angular. Mặc dù hầu hết các trường biểu mẫu và điều khiển đã phải được thiết kế lại để phù hợp tốt hơn với hệ thống HIS, các tab điều hướng, nút và trình phân trang đã sẵn có đã chứng minh được tính hữu dụng trong việc gia tăng tốc độ triển khai giao diện người dùng. Hộp thoại có thể dễ dàng tùy chỉnh, cho phép nhúng toàn bộ các thành phần bên trong chúng.

Thư viện giao diện này đã được chọn vì sự dễ sử dụng và tính phổ biến. Nó được duy trì một phần bởi Google, người cũng duy trì một phần của framework Angular. Lúc đó, đội của chúng em đã có nhiều kinh nghiệm với DevExtreme, một thư viện cung cấp nhiều thành phần hơn, nhưng cũng nặng hơn khi xây dựng các phiên bản đầu tiên của chương trình. Em sau đó được đề xuất nghiên cứu và sử dụng Angular Material trong hệ thống.

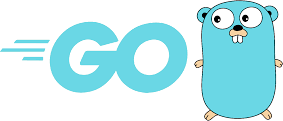
Front-end là bộ phận giao diện người dùng, cho phép người dùng tương tác với hệ thống một cách trực quan. Giao diện người dùng sẽ cần liên lạc với back-end thông qua các yêu cầu, các yêu cầu này cần thiết để xử lý và lưu trữ dữ liệu mà người dùng đã tương tác với.

**3.2 Back-end**

Backend là một phần quan trọng của ứng dụng máy tính, đặc biệt là các ứng dụng Web. Nó đề cập đến các thành phần và logic kinh doanh được xử lý trên máy chủ và không hiển thị trực tiếp cho người dùng. Trong bối cảnh ứng dụng Web, backend đảm nhiệm xử lý các yêu cầu từ phía frontend (giao diện người dùng) và thực hiện các tác vụ phức tạp như xử lý dữ liệu, quản lý cơ sở dữ liệu, tính toán, bảo mật và xử lý các logic nghiệp vụ.

Backend thường bao gồm các máy chủ, cơ sở dữ liệu và hệ thống xử lý dữ liệu khác như lưu trữ tạm thời (caching), truy xuất dữ liệu từ bên ngoài, và xác thực người dùng. Nó là nơi dữ liệu được lưu trữ và xử lý, đảm bảo hệ thống hoạt động một cách hiệu quả, bảo mật và đáng tin cậy. Trong dự án này, em đã được đề xuất sử dụng Golang cho máy chủ và MongoDB làm cơ sở dữ liệu.

**3.2.1 Golang**

Golang (Go Language) là một ngôn ngữ lập trình được sáng tạo và thiết kế bởi các nhân viên của Google. Tuy nhiên, Golang tồn tại như một dự án mã nguồn mở chứ không thuộc về quyền sở hữu của Google. Golang được xây dựng để việc lập trình trở nên dễ đọc, đơn giản, hiệu quả và đạt được hiệu suất cao. Golang còn được biết đến như là một ngôn ngữ static typed. Điều này có nghĩa là mọi thứ thuộc về Golang đều phải có kiểu dữ liệu và nó trái ngược với các ngôn ngữ dynamic typed như Python hoặc Javascript. Golang tương tự như Java hay C++ bởi nó cũng là một ngôn ngữ được sử dụng cho mục đích lập trình. Tuy nhiên, tính đơn giản chính là điểm khiến cho Golang trở nên khác biệt so với các ngôn ngữ lập trình khác. Nó sở hữu những cú pháp khá tinh gọn và có mức độ tương đồng rất lớn với C++.

Hình 3.3: Logo Golang

Việc lựa chọn Golang làm máy chủ là do nó đã được sử dụng cho hệ thống HIS của công ty và nó rất quen thuộc với công ty của em. Sử dụng Golang, chúng em đã xây dựng một máy chủ giao diện lập trình ứng dụng (API) đơn giản, cho phép phần front-end tương tác với nó bằng giao thức truyền tải siêu văn bản (HTTP).

Những phương án thay thế cho việc xây dựng máy chủ back-end bao gồm JavaScript trong môi trường NodeJS, tuy nhiên, máy chủ NodeJS không dễ mở rộng do chỉ sử dụng một luồng thực thi. Phương án khác là framework .NET của Microsoft, có thể cải thiện tính bảo trì của hệ thống do thiết kế cho Lập trình Hướng đối tượng.

Nhìn chung, Golang mang tính nhỏ gọn, sắc bén và sở hữu độ tương thích rất cao. Do đó, các chương trình phần mềm được vận hành bởi Golang vô cùng nhanh chóng từ nhiều điều kiện khác nhau.

**3.2.2 MongoDB**

MongoDB là một hệ quản trị cơ sở dữ liệu (Database Management System - DBMS) mã nguồn mở và phi quan hệ, được phát triển bởi MongoDB Inc. Mon- goDB sử dụng kiểu lưu trữ JSON (JavaScript Object Notation) linh hoạt và linh động để lưu trữ dữ liệu, thay vì cấu trúc bảng truyền thống như các hệ quản trị cơ sở dữ liệu SQL. Một số điểm nổi bật về MongoDB:

* Hướng tài liệu: MongoDB lưu trữ dữ liệu dưới dạng các tài liệu JSON (BSON- Binary JSON), giúp lưu trữ dữ liệu có cấu trúc linh hoạt và dễ dàng mở rộng dữ liệu mà không cần thay đổi cấu trúc cơ sở dữ liệu.
* Khả năng mở rộng: MongoDB hỗ trợ khả năng mở rộng dễ dàng bằng cách chia dữ liệu thành các replica set hoặc cluster, giúp tăng khả năng xử lý và tải của hệ thống.
* Tích hợp tốt với ngôn ngữ lập trình: MongoDB cung cấp các trình điều khiển (drivers) cho nhiều ngôn ngữ lập trình phổ biến, giúp dễ dàng tích hợp với ứng dụng sử dụng các ngôn ngữ này.
* Hỗ trợ các truy vấn phong phú: MongoDB cung cấp các truy vấn mạnh mẽ cho việc truy xuất và xử lý dữ liệu, bao gồm các truy vấn phức tạp như truy vấn phạm vi (range query), truy vấn theo điều kiện (conditional query), và truy vấn theo vị trí (geospatial query).
* Tích hợp tốt với cloud: MongoDB hỗ trợ tích hợp với các dịch vụ đám mây (cloud services) như Amazon Web Services (AWS) và Microsoft Azure, giúp dễ dàng triển khai và quản lý cơ sở dữ liệu trên môi trường đám mây.

Hình 3.4: Logo MongoDB

Với những ưu điểm trên, MongoDB đã trở thành một hệ quản trị cơ sở dữ liệu phổ biến và được sử dụng rộng rãi trong các ứng dụng Web và di động.

Việc sử dụng MongoDB làm cơ sở dữ liệu là do hệ thống HIS của công ty em trước đó đã sử dụng nó, nên việc sử dụng MongoDB giúp cho việc liên kết dữ liệu với hệ thống cũ trở nên thuận tiện hơn.

Trong chương này, em đã trình bày và các công nghệ chính mà em sử dụng trong đồ án này. Tiếp theo, trong chương 4, em sẽ trình bày mô tả chi tiết về kiến trúc và thiết kế của module này.

**CHƯƠNG 4: THỰC NGHIỆM VÀ ĐÁNH GIÁ**

**4.1 Thiết kế kiến trúc**

**4.1.1 Lựa chọn kiến trúc phần mềm**

**a) Model – View – Controller(MVC)**

MVC là một mô hình cực kì phổ biến và ngày càng phát triển mạnh mẽ trong lập trình web. Mô hình MVC bao gồm 3 phần là: View, Model và Controller.

* View: Phần này có nhiệm vụ chính là giao tiếp với người dùng, đây chính là thành phần có nhiệm vụ mô tả trình bày dữ liệu. Đây là phần sẽ hiển thị đối với người dùng, là nơi người dùng có thể tương tác, giúp người dùng có cái nhìn tổng quan bao quát về ứng dụng.
* Controller: là thành phần làm nhiệm vụ xử lý nghiệp vụ, nó nhận request từ người dùng từ đó đưa ra dữ liệu phù hợp với yêu cầu của người dùng đó.
* Model: là thành phần chịu trách nhiệm quản lý dữ liệu, nó lưu trữ và truy xuất thông tin từ cơ sở dữ liệu (như mysql, sql server, postgreSQL,. . . ), từ hệ thống file, từ các website khác (api, web service) đồng thời chứa các logic được thực thi bởi ứng dụng.

Mỗi thành phần Angular chủ yếu bao gồm một tập tin mẫu (template file) và một tập tin điều khiển (controller file) (ví dụ: app.component.html và app.component.ts). Trong tập tin TypeScript (TS), đối tượng thành phần chính có thể chứa nhiều thuộc tính và phương thức. Giá trị của các thuộc tính có thể được hiển thị trên màn hình bằng cách gọi chúng trong tập tin HTML. Tập tin HTML sau đó là một "View" bằng cách hiển thị các thuộc tính hoặc "Models". Người dùng có thể tương tác với View bằng cách sử dụng các nút, biểu mẫu, v.v., và có thể gọi các phương thức của thành phần. Các phương thức này có thể thay đổi giá trị của các thuộc tính của thành phần, biến đối tượng thành một "Controller" có thể cập nhật các Models.

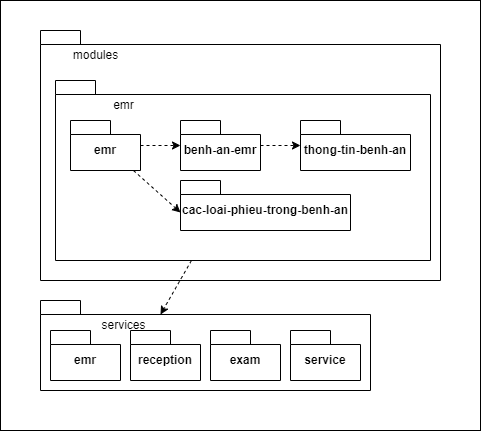
**b) Restful API (Representational State Transfer API)**

Restful API (Representational State Transfer API) là một kiểu kiến trúc phần mềm được sử dụng trong việc thiết kế các ứng dụng web và di động. Nó dựa trên các nguyên tắc của REST, một phương thức tiếp cận dựa trên HTTP để quản lý và tương tác với các tài nguyên.

Các điểm nổi bật của RESTful API bao gồm:

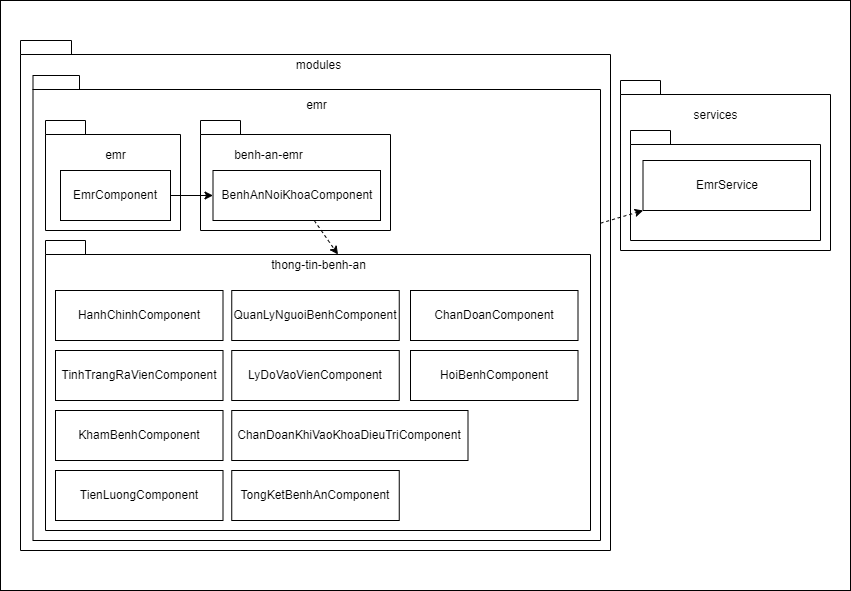
* Tài nguyên: API quản lý và cung cấp truy cập cho các tài nguyên như đối tượng, thông tin hoặc dữ liệu trong ứng dụng. Mỗi tài nguyên đại diện bởi một URL(Uniform Resource Identifier).
* Thao tác HTTP: RESTful API sử dụng các phương thức HTTP như GET (lấy dữ liệu), POST (tạo mới tài nguyên), PUT (cập nhật tài nguyên), DELETE (xóa tài nguyên) để tương tác với các tài nguyên.
* Biểu diễn dữ liệu: Dữ liệu được truyền giữa client và server dưới dạng các định dạng chuẩn như JSON (Javascript Object Notation) hoặc XML (eXtensible Markup Language), giúp dễ dàng đọc và hiểu.
* Stateless (Không lưu trạng thái): Các yêu cầu từ client đến server không lưu trạng thái của client. Mỗi yêu cầu là độc lập và không phụ thuộc vào các yêu cầu trước đó, giúp dễ dàng mở rộng và thay đổi hệ thống.
* Giao diện duy nhất: RESTful API cung cấp một giao diện duy nhất để truy cập vào các tài nguyên. Mỗi tài nguyên được quản lý bởi một URI riêng biệt và được thao tác bằng các phương thức HTTP duy nhất.
* Tích hợp dễ dàng: RESTful API sử dụng ngôn ngữ lập trình phổ biến và hỗ trợ tích hợp vào nhiều loại ứng dụng khác nhau, bao gồm web, di động và desktop.

Sử dụng RESTful API giúp phát triển các ứng dụng linh hoạt, dễ dàng mở rộng và tương tác với các hệ thống và dịch vụ khác. Nó là một tiêu chuẩn phổ biến trong việc xây dựng các ứng dụng phân tán, hỗ trợ tốt trong việc tích hợp và cung cấp các dịch vụ mạnh mẽ cho người dùng.

**4.1.2 Thiết kế tổng quan**

Hình 4.1: Biểu đồ gói tổng quát

Hình [4.1](#_bookmark47) mô tả biểu đồ gói tổng quát của module. Trong hệ thống bệnh viện, gồm nhiều module, module của em trong đồ án này là module emr(hồ sơ bệnh án điện tử). Đầu vào sẽ là gói emr, trong này chứa màn hình chính để thao tác. Các bệnh án dùng trong bệnh viện có rất nhiều loại bệnh án. Giao diện của các loại bệnh án này sẽ nằm trong gói benh-an-emr. Giao diện của bệnh án được tạo thành bởi các mục thông tin của bệnh án nằm trong gói thong-tin-benh-an. Gói services là gói chính tương tác với backend thông qua các API nhằm tạo và sửa các thông tin trong bệnh án. Người dùng có thể nhập các thông tin từ đầu hoặc có thể lấy các thông tin của bệnh nhân từ hệ thống bệnh viện (HIS). Bao gồm thông tin tiếp nhận và thông tin khám chữa bệnh.

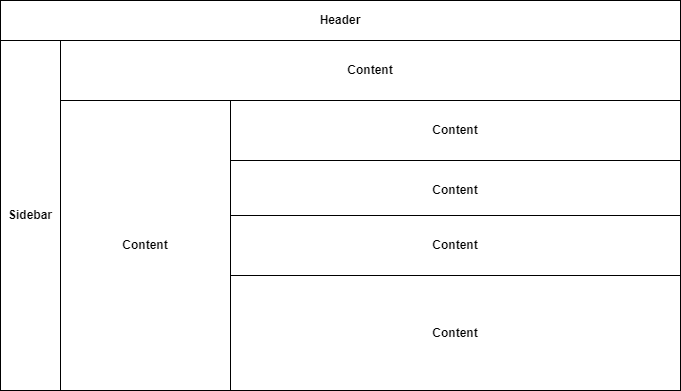
**4.1.3 Thiết kế chi tiết gói**

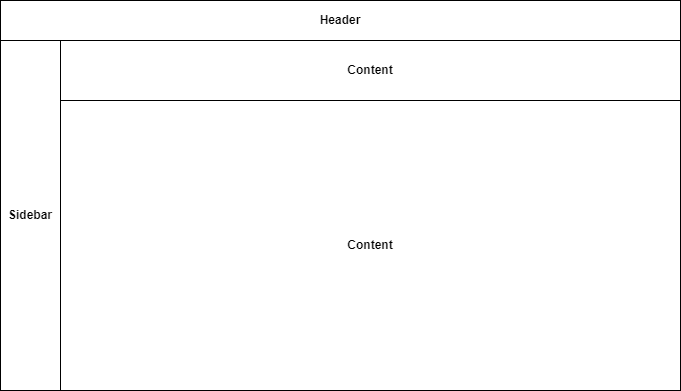
Hình 4.2: Thiết kế chi tiết gói

## **4.2 Thiết kế chi tiết**

### **4.2.1 Thiết kế giao diện**

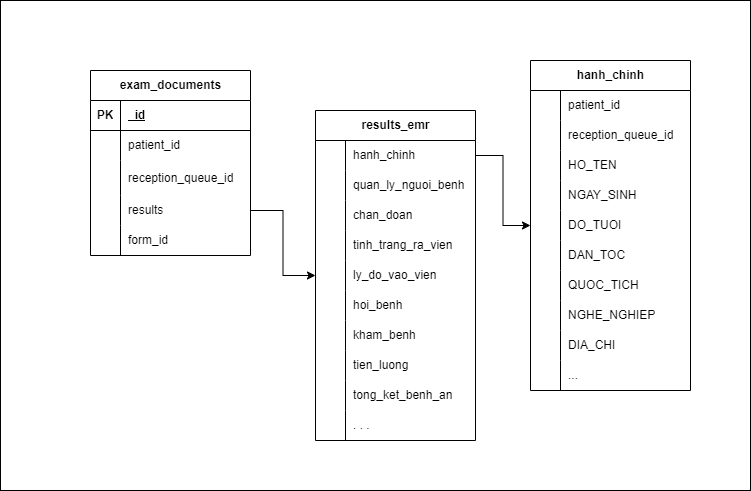
Giao diện hệ thống được thiết kế để hiển thị trên nhiều màn hình khác nhau. Hệ thống đã được thử nghiệm hiển thị tốt ở một số độ phân giải như 1920x1080, 1366x768, 1280x720, 960 x 540, thường có trên màn hình PC,Laptop và Ipad.



Hình [4.3](#_bookmark52) mô tả bố cục giao diện trang quản lý hồ sơ bệnh án và giấy tờ liên quan.

Hình 4.4: Mô hình giao diện trang hồ sơ bệnh án

Tất cả các trang trong hệ thống đều có bố cục tương tự như HIS, bao gồm ba thành phần chính: navigational sidebar, header và và các thành phần content hiển thị nội dung chi tiết của các chức năng khác nhau của tùy chọn sidebar. Sidebar sẽ cung cấp một loạt menu điều hướng và menu con giữa các chức năng, được nhóm theo các yêu cầu của khách hàng. Header sẽ chứa một menu cho các hành động chính của trang, chẳng hạn như tạo, chỉnh sửa, xóa, lưu,.. Mỗi chức năng có các mục đích và tương tác người dùng khác nhau, do đó, bố cục của nội dung chính sẽ khác nhau đối với hầu hết các trang.

**4.2.2 Thiết kế cơ sở dữ liệu**

Hình 4.5: Cơ sở dữ liệu hệ thống

Hình 4.5 mô tả cơ sở dữ liệu của hệ thống. Tùy thuộc vào form-id mà results sẽ trả về các trường dữ liệu khác nhau. Ví dụ form-id=5045 thì results sẽ trả về các trường dữ liệu lưu thông tin hồ sơ bệnh án như: hanh-chinh, chan-doan, quan-ly- nguoi-benh, kham-benh,.Còn giá trị của form-id mà khác 5045 thì results sẽ trả về các trường dữ liệu lưu thông giấy tờ liên quan của bệnh án.

**4.3 Xây dựng ứng dụng**

**4.3.1 Thư viện và công cụ sử dụng**

Trong quá trình thực hiện đồ án , em đã sử dụng những công cụ , thư viện hỗ trợ được trình bày trong bảng [4.1](#_bookmark58)

|  |  |  |
| --- | --- | --- |
| **Mục đích** | **Công cụ** | **Địa chỉ URL** |
| IDE lập trình | WebStorm | https://www.jetbrains.com/webstorm/ |
| Thư viện lập trình Frontend | Angular | https://angular.io/ |
| Thư viện lập trình Backend | Golang | https://go.dev/ |
| Vẽ biểu đồ | Draw.io | https://app.diagrams.net/ |

Bảng 4.1: Danh sách thư viện và công cụ đã sử dụng

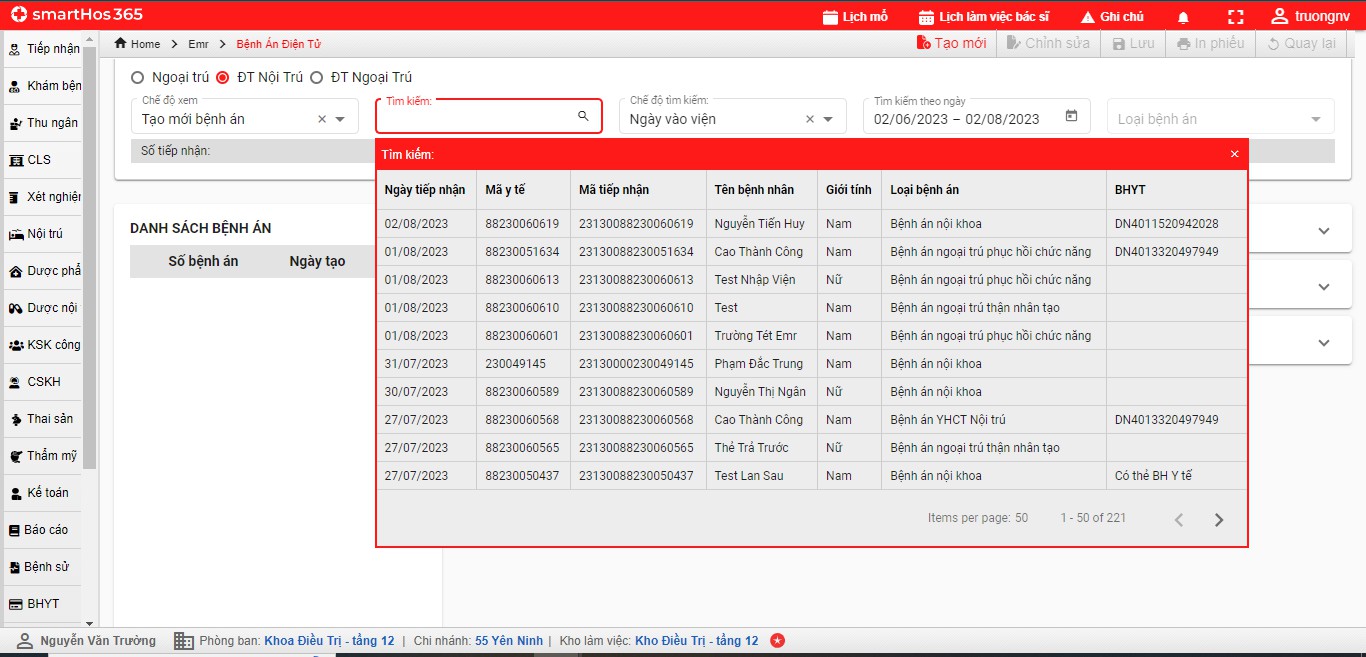
**4.3.2 Kết quả đạt được**

Sau khi hoàn thành sơ bộ, module của chúng em đã hoàn thành cơ bản những chức năng cơ bản mà một hồ sơ bệnh án điện tử cần phải có như sau:

* Cho phép người dùng tìm kiếm thông tin bệnh nhân nội/ngoại trú.
* Chức năng Bấm để xem thêm chi tiết thông tin bệnh nhân
* Chức năng tạo mới một bộ hồ sơ bệnh án
* Cho phép chọn loại hồ sơ bệnh án
* Xem các bệnh án chưa ký
* Chức năng tạo mới các phiếu trong bệnh án
* Chức năng lưu phiếu
* Chức năng in bệnh án
* Chức năng View để xem chi tiết các loại bệnh án/ giấy tờ

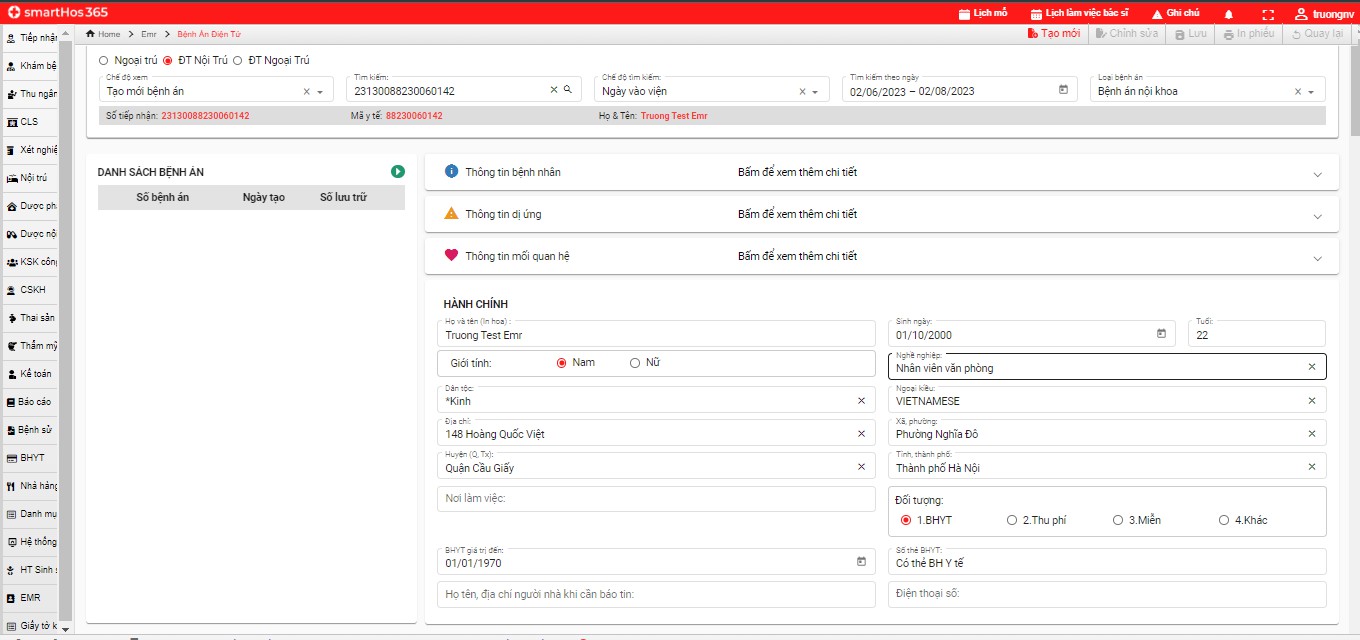
**4.3.3 Minh họa các chức năng chính**

Trong mục này sẽ hiển thị một số ảnh chụp màn hình giao diện người dùng của hệ thống, với lời giải thích ngắn gọn về những màn hình đó.

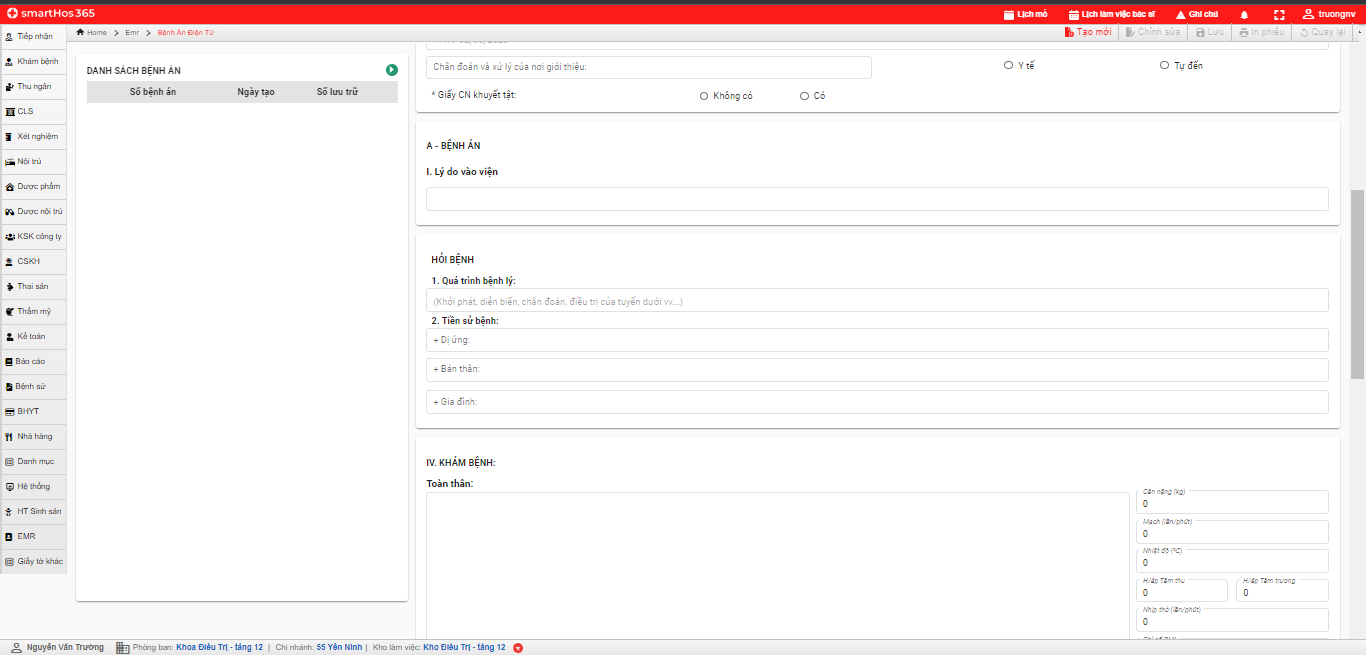


Hình 4.6: Tìm kiếm bệnh nhân

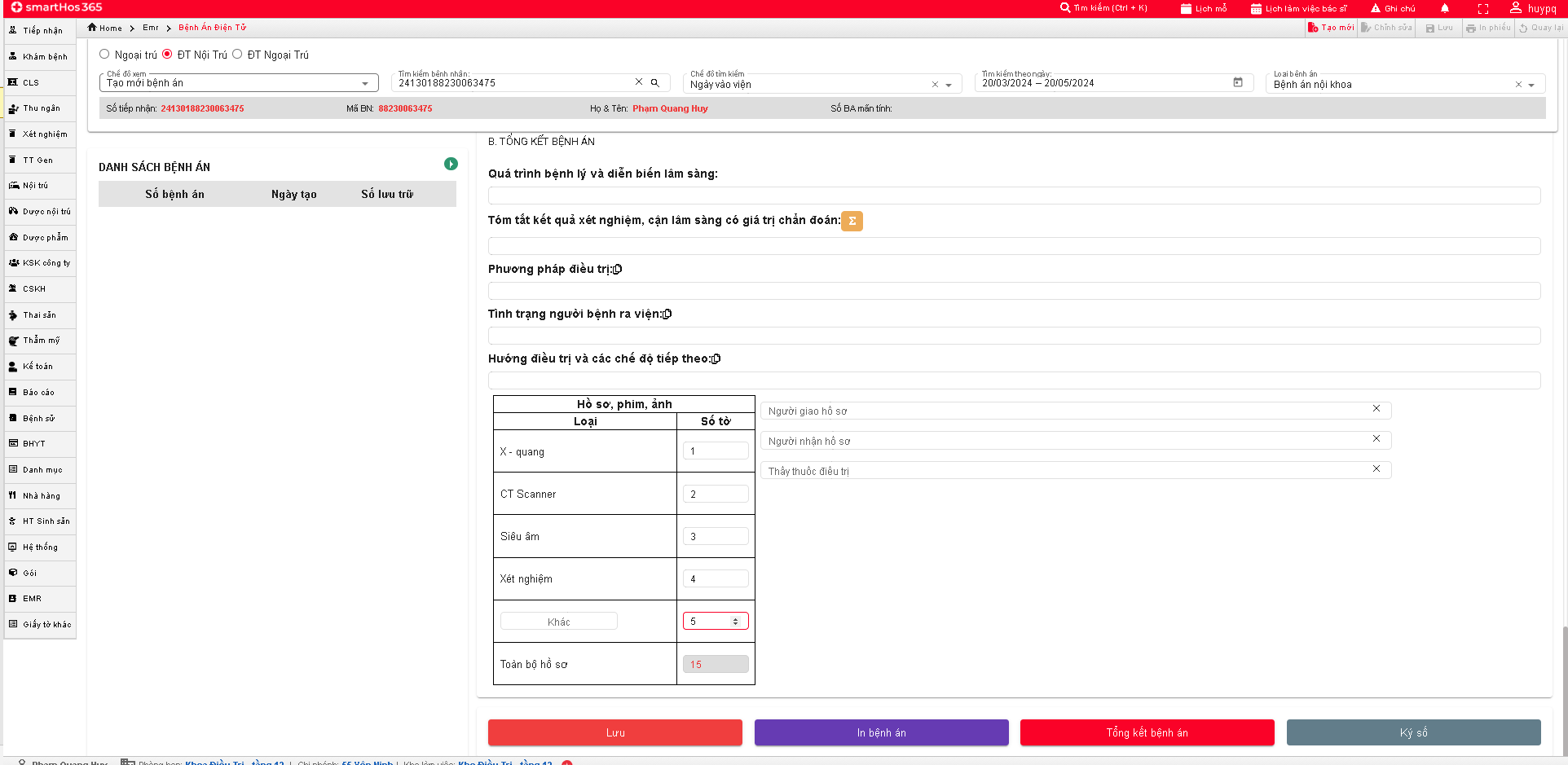
Hình [4.6](#_bookmark61) hiển thị màn hình tìm kiếm bệnh nhân. Nhân viên có thể tìm kiếm theo một trong những thông tin của bệnh nhân.



Hình 4.7: Chức năng tạo bệnh án



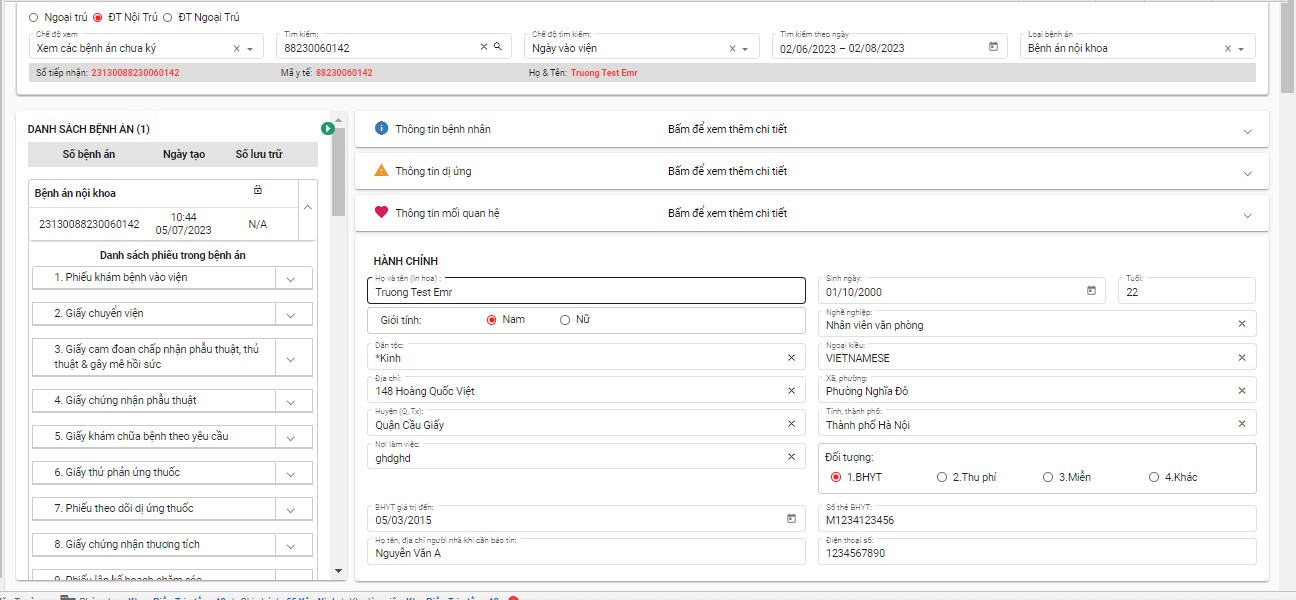
Hình 4.8: Chức năng tạo bệnh án

****

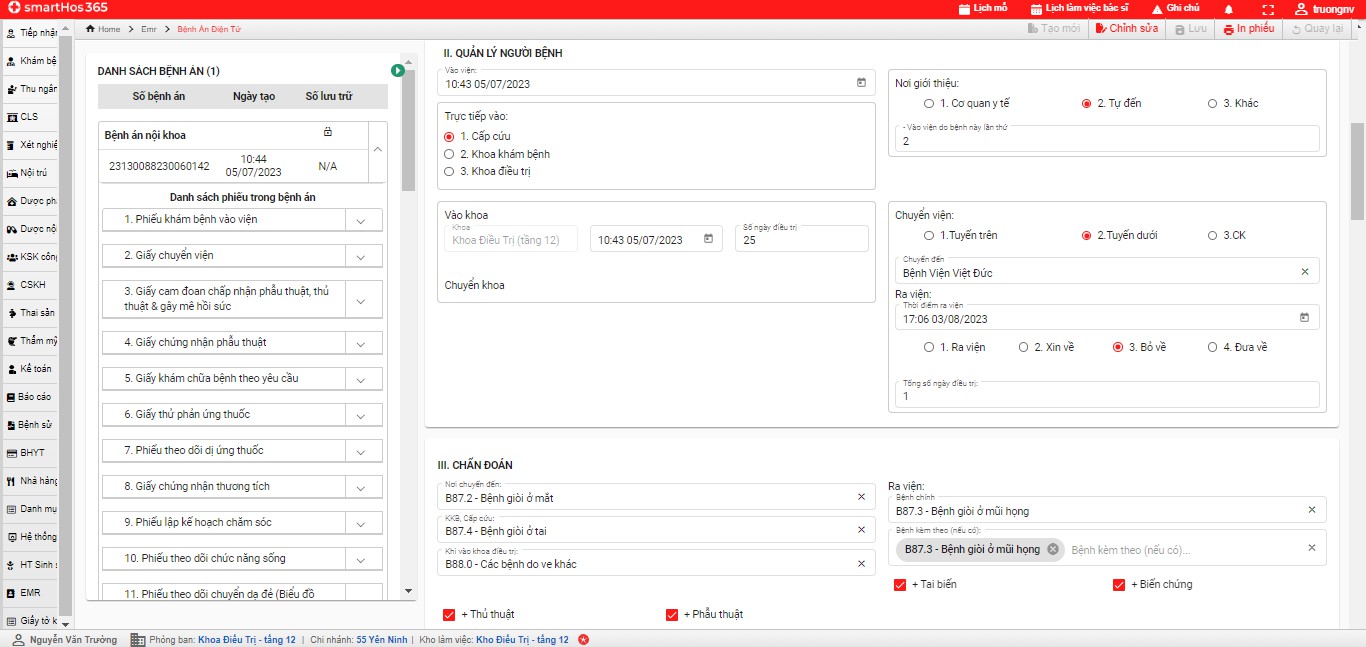
Hình 4.9: Chức năng tạo bệnh án

Các hình [4.7,](#_bookmark62) [4.8,](#_bookmark63) [4.9](#_bookmark64) hiển thị màn hình tạo một hồ sơ bệnh án cho bệnh nhân có mã y tế là 88230051634. Để tạo một bệnh án, ban đầu người dùng sẽ phải chọn chế độ xem: tạo bệnh án sau khi đã chọn bệnh nhân ở mục tìm kiếm. Sau đó người dùng sẽ chọn loại bệnh án. Những bệnh án nội trú sẽ lấy theo loại bệnh án trên HIS có sẵn khi nhập viện. Tiếp theo bác sĩ sẽ kiểm tra và bổ sung thông loại bệnh án.

Tại đây bác sĩ sẽ nhập các thông tin của bệnh nhân như thông tin hành chính, thông tin trong quá trình khám chữa bệnh vào mục bệnh án. Cuối trang là thông mục thông tin tổng kết bệnh án của bệnh nhân. Thông thường sẽ bao gồm: (i) Quá trình bệnh lý và diễn dải lâm sàng, (ii) Tóm tắt kết quả xét nghiệm cận lâm sàng có giá trị chẩn đoán,(iii)Phương pháp điều trị, (iv)Tình trạng người bệnh ra viện, (v) Hướng điều trị và các chế độ tiếp theo. Các trường thông tin này bao gồm đẩy đủ tất cả các trường thông tin cần có của bệnh án theo biểu mẫu của từng bệnh án do bộ y tế ban hành.

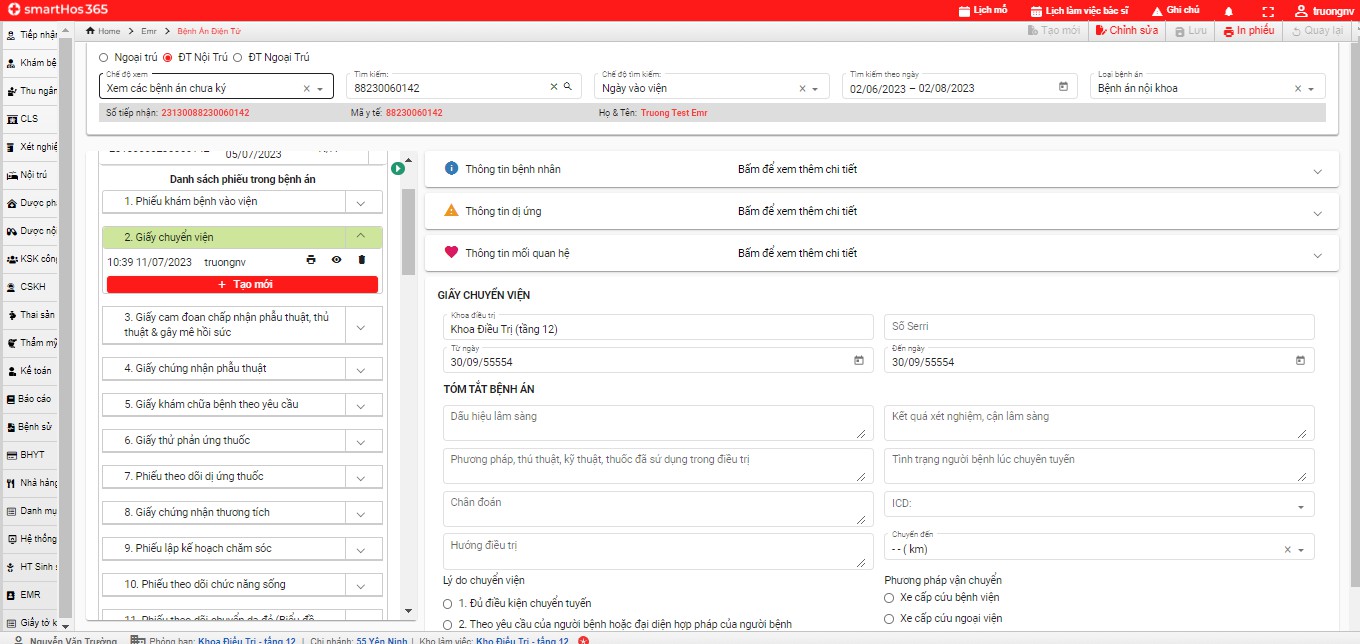


Hình 4.10: Xem bệnh án



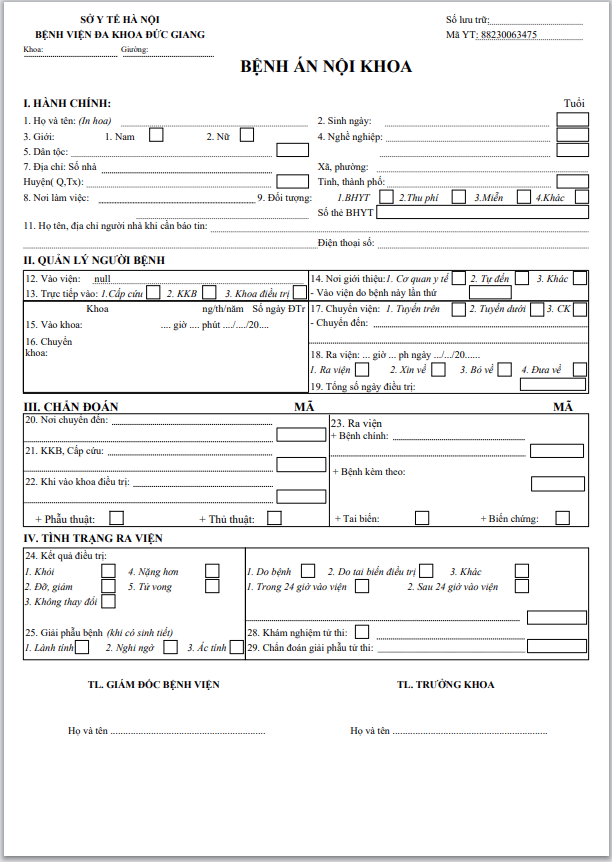
Hình 4.11: Xem bệnh án

Các hinh [4.10,](#_bookmark65) [4.11](#_bookmark66) hiển thị màn hình xem bệnh án đã lưu và danh sách các giấy tờ liên quan đến bệnh án. Ở đây bao gồm tất cả các giấy tờ được tạo mới ở EMR và các giấy tờ được tổng hợp từ HIS như: giấy siêu âm, chụp X-quang,...



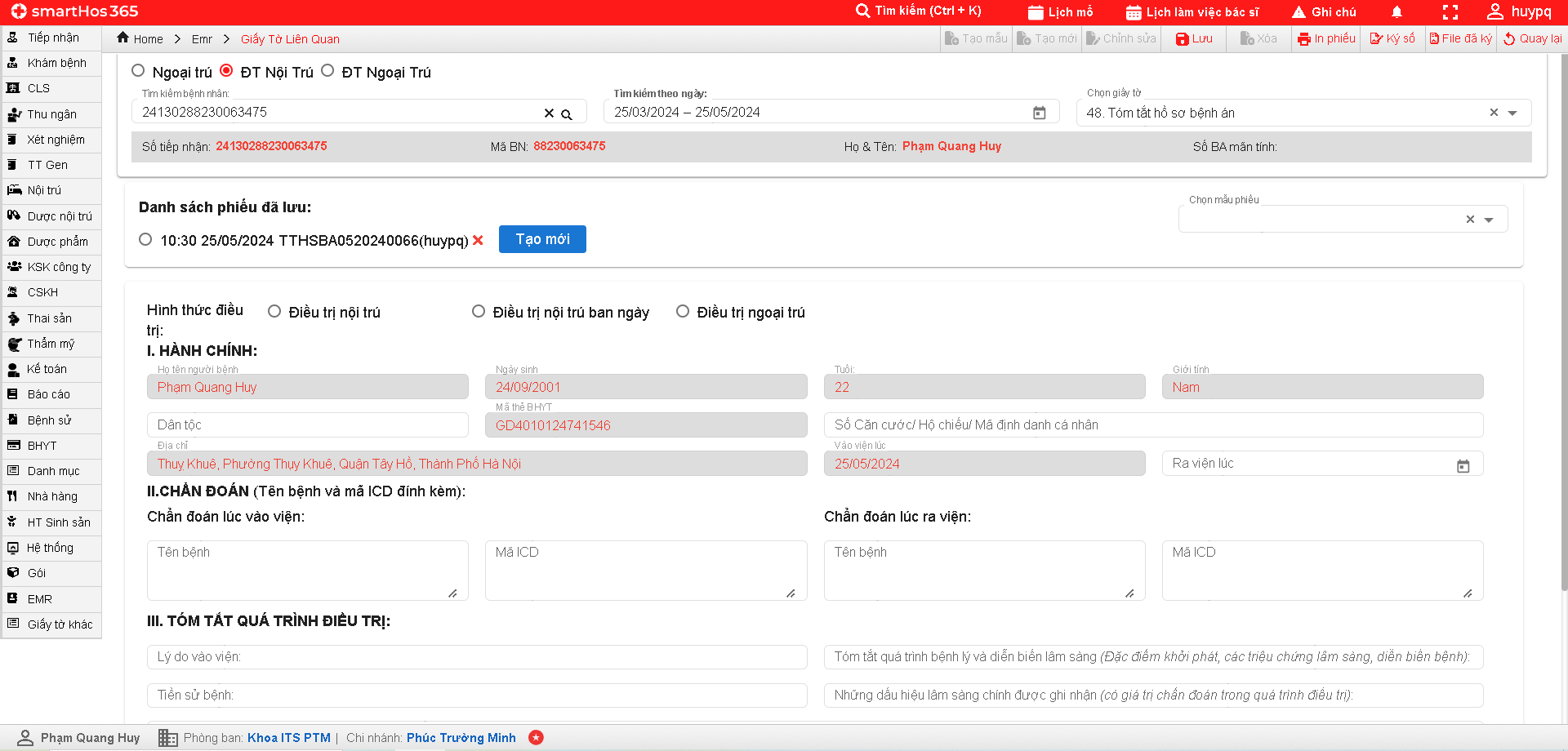
Hình 4.12: Xem giấy tờ liên quan

Hình [4.12](#_bookmark67) hiển thị màn hình nhập thông tin của 1 trong các giấy tờ liên quan của bệnh án (trong đó có thể là giấy tờ tạo mới trong riêng EMR hoặc giấy tờ được lấy từ HIS và được sửa mới ở EMR). Trong danh sách bệnh án sẽ bao gồm bệnh án đã tạo và các giấy tờ liên quan trong lần khám chữa bệnh đó. Điều này sẽ giúp người dùng tìm kiếm và tổng hợp các giấy tờ của bệnh nhân nhanh chóng.

****

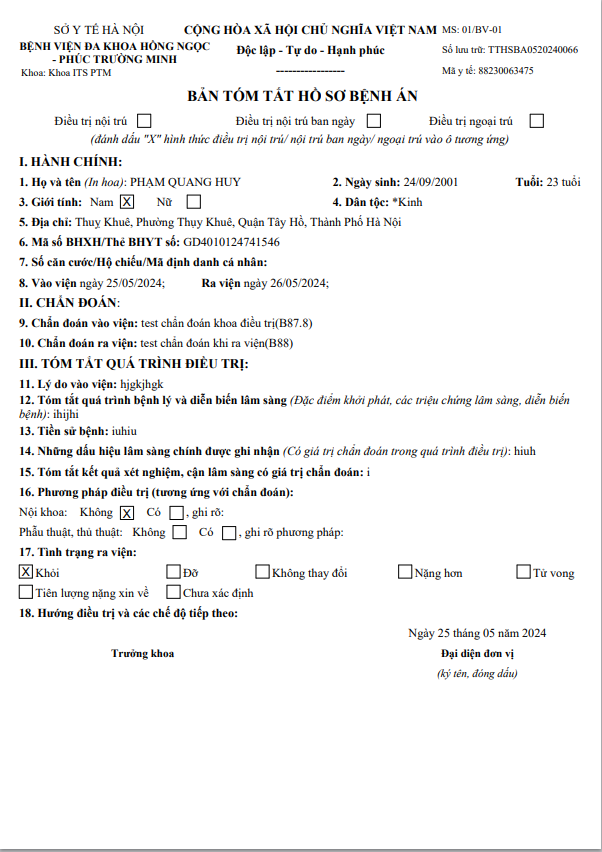
Hình 4.13: In phiếu, xuất file PDF

Hình [4.13](#_bookmark68) hiển thị màn hình in phiếu bệnh án. Tại đây nhân viên y tế có thể lựa chọn in giấy hoặc có thể xuất ra file PDF để lưu trữ.



Hình 4.14: Màn hình giấy tờ liên quan

Hình 4.14 hiển thị màn hình giấy tờ liên quan(tóm tắt hồ sơ bệnh án). Nhân viên y tế sẽ chọn bệnh nhân và loại giấy tờ liên quan muốn làm.Tại đây các thông tin hành chính của bệnh nhân sẽ tự đổ từ màn hình tiếp nhận. Tên bệnh và mã icd khi chẩn đoán nhập viện sẽ được link từ màn hình khám bệnh. Tên bệnh và mã icd chẩn đoán lúc ra viện sẽ được link từ màn hình ra viện.Sau đó nhân viên y tế sẽ nhập thông tin trong quá trình khám bệnh vào và thực hiện lưu phiếu giấy tờ liên quan



Hình 4.15: Phiếu in tóm tắt bệnh án

## **4.4 Kiểm thử**

Mục đích: Đảm bảo hệ thống có thể chạy đúng theo các yêu cầu đặt ra trước đó.

Phạm vi kiểm thử: Kiểm thử các chức năng trong thiết kế được phát triển (các chức năng đã thiết kế nhưng chưa thực hiện sẽ không kiểm thử), các thông tin hiện lên chính xác như mong muốn. Hiệu năng của hệ thống tạm thời chưa kiểm thử.

Hệ thống sử dụng cho cho đối tượng chính là bác sĩ và điều dưỡng vậy nên việc kiểm thử sẽ chia ra kiểm thử từng chức năng của hai đối tượng này.

Loại kiểm thử sử dụng: Unit Test - kiểm thử phần mềm trong đó các đơn vị hay thành phần riêng lẻ của phần mềm được kiểm thử.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Chức năng | Quy trình kiểm thử | Trường hợp | Mong muốn | Kết quả |
| Tìm kiếm bệnh nhân | 1. Ấn vào thanh tìm kiếm  2.Chọn chế độ tìm kiếm  3.Chọn thời gian tìm kiếm | Ngày vào viện từ 1/5 /2024 đến 20/5/2024 | Hiển thị những bệnh nhân vào viện trong khoảng 1/5/2024 đến 20/5/2024 | Pass |
| Ngày ra viện từ 1/5/2024 đến 20/5/2024 | Hiển thị những bệnh nhân ra viện trong khoảng 1/5/2024 đến 20/5/2024 | Pass |
| Năm sinh là năm 2001 | Hiển thị ra những bệnh nhân sinh năm 2001 | Pass |
| Số điện thoại: 123456789 | Hiển thị ra bệnh nhân có sđt 123456789 | Pass |
|  |  | Địa chỉ: Hà nội | Hiển thị ra những bệnh nhân có địa chỉ Hà Nội | Pass |
| Giới tính: Nam | Hiển thị ra những bệnh nhân có giới tính nam | Pass |
| Tạo bệnh án | B1:Chọn bệnh nhân  B2: Ân “Tạo mới”  B3:Điền thông tin  B4:Ấn “Lưu” | Nhập đủ thông tin bắt buộc | Thêm bệnh án vào CSDL và hiển thị thông báo lưu thành công | Pass |
| Nhập thiếu thông tin bắt buộc | Hiển thị thông báo trường bắt buộc chưa được điền | Pass |
| Tạo giấy tờ liên quan | B1:Chọn bệnh nhân  B2: Chọn loại giấy tờ liên quan muốn tạo cho bệnh nhân  B3: Ân “Tạo mới”  B4: Điền thông tin  B5: Ấn “Lưu” | Nhập đủ thông tin bắt buộc | Thêm bệnh án vào CSDL và hiển thị thông báo lưu thành công | Pass |
| Nhập thiếu thông tin bắt buộc | Hiển thị thông báo trường bắt buộc chưa được điền | Pass |
| In phiếu bệnh án điện tử | B1:Chọn bệnh nhân  B2:Chọn bệnh án điện đã tạo  B3: Ấn in phiếu |  | Hiển thị ra phiếu in và thiết kết theo đúng mẫu bộ y tế | Pass |
| In phiếu giấy tờ liên quan | B1:Chọn bệnh nhân  B2:Chọn bệnh án điện đã tạo  B3: Ấn in phiếu |  | Hiển thị ra phiếu in và thiết kết theo đúng mẫu bộ y tế | Pass |
| Link thông tin của bệnh nhân | B1: Chọn bệnh nhân | Thông tin hành chính của bệnh nhan | Link thông tin hành chính của bệnh nhân từ màn hình tiếp nhận | Pass |
| Thông tin khám bệnh của bệnh nhân | Link thông khám bệnh của bệnh nhân từ màn hình khám bệnh | Pass |
| Tạo mẫu các trường thông tin | B1: Chọn bệnh  B2: Tạo mới | Tạo mới mẫu | Có thể lưu và sử dụng mẫu | Pass |
| Sửa mẫu đã có | Sửa thành công và cập nhật mẫu sau khi sửa | Pass |
| Xóa mẫu đã có | Xóa mẫu đã tạo khỏi danh sách | Pass |

Bảng 4.2: Kiểm thử chức năng

**4.5 Triển khai**

Trong dự án này, em đã được giao nhiệm vụ phát triển giao diện và tích hợp API cho màn hình EMR (Electronic Medical Record - Hồ sơ bệnh án điện tử). Mục tiêu của dự án là cải thiện trải nghiệm người dùng và tăng tính tương tác của hệ thống EMR, giúp bác sĩ và nhân viên y tế dễ dàng truy cập và quản lý thông tin bệnh án.

-Phát triển giao diện

* Xây dựng giao diện người dùng thân thiện, tương thích với nhiều thiết bị và kích thước màn hình khác nhau.
* Sử dụng các công nghệ hiện đại như HTML5, CSS3 và JavaScript để tạo ra giao diện đẹp mắt và dễ sử dụng.
* Tối ưu hóa trải nghiệm người dùng bằng cách cân nhắc về mặt thẩm mỹ và hiệu suất.

-Gắn kết api

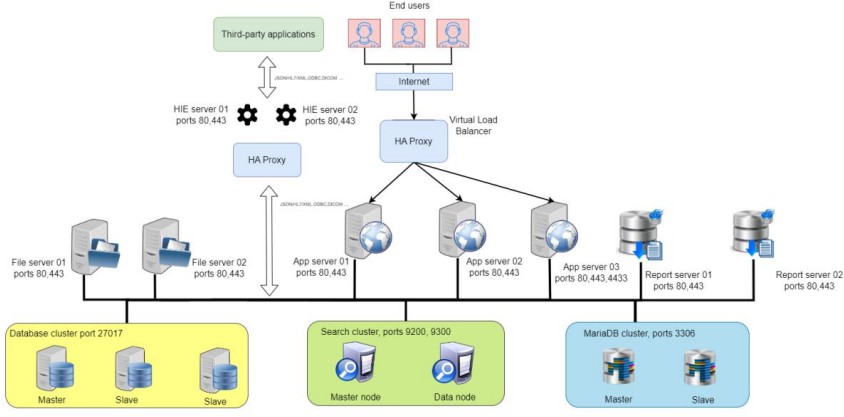
* Tiến hành phân tích yêu cầu API từ hệ thống EMR và xác định các endpoint cần tích hợp.
* Sử dụng các thư viện và công cụ phát triển phần mềm để gắn kết API vào giao diện người dùng.
* Kiểm tra và xử lý các trường hợp lỗi khi gọi API để đảm bảo tính ổn định của hệ thống.

-Làm phiếu in

* Tiến hành phân tích yêu cầu từ các bộ phận liên quan để hiểu rõ các thông tin cần được hiển thị trên phiếu in.
* Xác định các yêu cầu về định dạng, kích thước và mẫu mã của phiếu in.
* Sử dụng phần mềm jasper studio để tạo ra một mẫu phiếu in thẩm mỹ và dễ đọc.
* Tối ưu hóa bố cục và cân nhắc về màu sắc và phông chữ để làm cho phiếu in trở nên chuyên nghiệp và dễ sử dụng.

**4.5.1 Kết thúc triển khai**

Mô-đun được triển khai trong cùng một mạng với HIS, được minh họa trong Hình [4.14.](#_bookmark73) Dưới đây chỉ giải thích ngắn gọn về HIS, vì chi tiết của chúng nằm ngoài phạm vi của đồ án này



Hình 4.14: Kiến trúc triển khai

Các yêu cầu từ người dùng được đưa vào một bộ cân bằng tải bằng cách sử dụng một Proxy có tính sẵn có cao (HAProxy), sau đó được định tuyến vào ba máy chủ:

* *Application server 01*: Máy chủ này chịu trách nhiệm lưu trữ giao diện người dùng tổng hợp của các ứng dụng trong HIS, được viết bằng PHP, AngularJS hoặc Angular để thực hiện yêu cầu tới hai máy chủ back-end ở phía dưới.
* *Application server 02*: Máy chủ này xử lý các yêu cầu được thực hiện bởi máy chủ 01 bằng PHP hoặc Golang. Phần backend của các ứng dụng HIS được lưu trữ ở đây.
* *Application server 03*: Máy chủ này tương tự như máy chủ 02, đồng thời xử lý các yêu cầu được định tuyến từ máy chủ 01. Nó có thêm các tên miền phụ để phục vụ cho các ứng dụng thử nghiệm của công ty, bao gồm mô-đun quản lý Hồ sơ bệnh án của chúng em.

Các máy chủ back-end 02 và 03 thường cần lưu trữ, quản lý và truy vấn dữ liệu liên tục. Điều này được thực hiện trong HIS bằng cách định tuyến các yêu cầu cơ sở dữ liệu thông qua một bus chung đến hai nhóm cơ sở dữ liệu và các yêu cầu truy vấn đến một nhóm tìm kiếm. Hai cơ sở dữ liệu và một dịch vụ tìm kiếm mỗi cái đều có một máy chủ chính và ít nhất một máy chủ phụ, sao chép các hành động được thực hiện trên máy chủ chính.

Các trường dữ liệu thường được sử dụng từ nhóm cụm MariaDB và nhóm cụm MongoDB được tạo chỉ mục trên một nhóm máy chủ tìm kiếm để tăng tốc độ truy vấn, với sự hỗ trợ của một dịch vụ Elasticsearch. Các yêu cầu tìm kiếm được thực hiện bởi ứng dụng được định tuyến đến các máy chủ này để nhanh chóng lấy các bản ghi dữ liệu cần thiết, thay vì truy vấn trực tiếp vào cơ sở dữ liệu.

Kiến trúc cơ sở dữ liệu này cũng cho phép công ty quản lý một hệ thống báo cáo hiệu quả. HIS có tích hợp hệ thống báo cáo được hỗ trợ bởi JasperReports, một công cụ báo cáo mã nguồn mở hoạt động trên hai máy chủ của công ty. Các máy chủ này có thể được truy cập để xây dựng, lưu trữ và in các mẫu báo cáo một cách trực quan với các chỉ dẫn rõ ràng về vị trí dữ liệu cần thiết và từ bảng dữ liệu hoặc bộ sưu tập cơ sở dữ liệu nào. Khi ứng dụng yêu cầu một báo cáo cho một bản ghi cơ sở dữ liệu, mẫu báo cáo này sẽ được điền với dữ liệu cần thiết của bản ghi và cuối cùng được gửi lại cho ứng dụng dưới dạng tài liệu có thể in được.

**4.5.2 Tóm tắt về thông số kỹ thuật của HIS**

Bằng cách sử dụng kiến trúc này, HIS có thể xử lý đồng thời khoảng 1000 người dùng trực tuyến và có khả năng xử lý 3000 đến 5000 yêu cầu mỗi ngày bởi các cơ sở khám chữa bệnh sử dụng HIS. Dữ liệu sẵn có trong khoảng 100 đến 300 mili giây. các ứng dụng do máy chủ lưu trữ có thể cung cấp giao diện người dùng với tốc độ nhanh, trong vòng không quá 100 mili giây.

Module của chúng em sẽ được triển khai vào hệ thống này, đảm bảo tính sẵn sàng và tốc độ khi tham gia vào việc sử dụng thực tế của khách hàng. Vì quy mô của một hồ sơ bệnh án nhỏ hơn rất nhiều so với HIS, nên HIS sẽ là một giải pháp đủ cho việc triển khai của module này.

Trên đây, em đã trình bày về thiết kế kiến trúc, thiết kế chi tiết, xây dựng hệ thống và triển khai. Tiếp theo, trong chương 5, em xẽ trình bày về những đóng góp cũng như kết luận và hướng phát triển tiếp theo trong đồ án của em.

**CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

Ở trên, em đã trinh bày toàn bộ những gì liên quan đến đồ án của em. Đây là chương cuối cùng em sẽ tổng kết các kết quả đã đạt được trong suốt quá trình làm đồ án tốt nghiệp, đồng thời cũng đánh giá lại kiến thức đã học và những kinh nghiệm quý báu đã học được và các hướng phát triển cải thiện trong tương lai, những phần hạn chế về thời gian em chưa thể thực hiện.

**5.1 Kết luận**

Tổng kết lại, sau khi kết thúc quá trình làm đồ án tốt nghiệp, em đã hoàn thành công việc xây dựng một hồ sơ bệnh án điện tử, đáp ứng được những chức năng cơ bản theo yêu cầu của bộ y tế. Một module có thể quản lý chúng hồ sơ bệnh án và các giấy tờ liên quan, thay vì tách riêng ra như những phần mềm mà đa số các bệnh viện đã sử dụng trước đó. Qua quá trình thực hiện đồ án, em đã rút ra được nhiều kết luận quan trọng và học được nhiều điều bổ ích. Từ việc phân tích chi tiết từng quy trình hoạt động đến nghiên cứu các công nghệ mới, em đã có cơ hội hiểu sâu hơn về Angular, MongoDB, Golang cơ bản và áp dụng thành công trong dự án. Đây sẽ là hành trang, kiến thức quan trọng và vô cùng quý báu để cho em có thể tự tin, vững bước giúp em có thể bước ra từ ghế nhà trường, phát triển hơn nữa cho sự nghiệp bản thân trong tương lai.

Kết quả đáng chú ý mà em đạt được đó là xây dựng được một module có thể quản lý hồ sơ bệnh án và giấy tờ liên quan của bệnh nhân trong quá trinh khám chữa bệnh tại bệnh viện. Em đã thiết lập được cơ bản các chứ năng liên quan đến quản lý hồ sơ bệnh án và giấy tờ liên quan, giúp bác sĩ dễ dàng tìm lại hồ sơ bệnh án của bệnh nhân, giảm thời gian thăm khám, hỗ trợ điều trị kịp thời, giúp nâng cao chất lượng chẩn đoán và điều trị. Góp phần vào công cuộc số hóa các giấy tờ của bệnh nhân, nhằm hướng tới chỉ thị phát triển hồ sơ bệnh án điện tử trong tất cả các bệnh viện của bộ y tế đề ra: Các bệnh viện trên toàn quốc phải bắt đầu triển khai xây dựng lộ trình thực hiện chuyển đổi từ bệnh án giấy sang bệnh án điện tử, đảm bảo mục tiêu chậm nhất đến năm 2030 các bệnh viện trên toàn quốc phải hoàn thành việc chuyển đổi này.

Sau khi module này của em được triển khai, nó sẽ giải quyết được phần nào bài toán và yêu cầu đưa ra ban đầu, giúp ích lớn cho quá trình khám chữa bệnh tại các cơ sở khám chữa bệnh. Trước đây, khi bệnh nhân xuất viện, điều dưỡng phải ôm hồ sơ bệnh án giấy đến Phòng Kế hoạch tổng hợp, lãnh đạo bệnh viện... để ký, đóng mộc thì nay chỉ cần chuyển bệnh án điện tử của bệnh nhân đó trên máy vi tính hoặc máy tính bảng thông qua hệ thống phần mềm bệnh án điện tử cho các khoa phòng liên quan, trực lãnh đạo thì toàn bộ bệnh án của bệnh nhân được xem xét và sửa đổi trên hệ thống. Do vậy, điều dưỡng không phải đi lại trình ký, chờ đợi nữa. Hay ngày trước nếu có ca bệnh nào có bất thường muốn kiếm y chứng, nhân viên Phòng Kế hoạch Tổng hợp phải leo lên kho lưu trữ hồ sơ bệnh án giấy ở lầu 7, phải lục trong cả hàng ngàn hồ sơ để kiếm đúng bệnh án mình cần, rất mệt mỏi, tốn thời gian, công sức, thì nay với bệnh án điện tử chỉ cần nhập tên vào là bệnh án của bệnh nhân đó sẽ hiện ra. Còn đối với bác sĩ thì trước đây mỗi lần đi khám bệnh điều dưỡng phải đẩy nguyên cái xe có khoảng 50 hồ sơ bệnh án giấy của bệnh nhân đi theo, còn giờ bác sĩ chỉ cầm cái iPad, đến bệnh nhân nào nhập mã số bệnh nhân đó và thao tác trên máy luôn nên rất thuận tiện, tất cả thông tin diễn biến bệnh, phương pháp điều trị, các kết quả cận lâm sàng như siêu âm, xét nghiệm, chụp phim... đều được lưu trên bệnh án điện tử, bác sĩ có thể xem lại bất kỳ lúc nào cần, mà không phải gọi điều dưỡng tìm kiếm như lưu ở bệnh án giấy. Một điều thuận tiện nữa mà module bệnh án điện tử của chúng em mang lại đó là: lưu lại tất cả thông tin điều trị của bệnh nhân trong đợt nằm viện đó, nếu lần sau bệnh nhân nhập viện, bác sĩ chỉ cần nhập mã bệnh án là các thông tin hiện ra, giúp ích rất nhiều cho bác sĩ trong chẩn đoán điều trị. Còn với người bệnh, khi đến bệnh viện khám chữa bệnh không phải mang sổ khám bệnh, các giấy tờ xét nghiệm, chụp phim, hay toa thuốc... đã điều trị trước đây. Mà thông qua bệnh án điện tử bác sĩ sẽ cập nhật được tiền sử bệnh tật của bệnh nhân, từ đó sẽ có phương án điều trị thích hợp. Module bệnh án điện tử này của chúng em được kết nối với các phần mềm như hệ thống quản lý bệnh viện (Hospital Information System - HIS), hệ thống thông tin lưu trữ và thu nhận hình ảnh (Picture Archiving and Communication System - PACS), hệ thống quản lý thông tin xét nghiệm (Laboratory Information System - LIS), do đó, tất cả các kết quả khám chữa bệnh, kết quả xét nghiệm, chụp phim, siêu âm đều được đẩy lên phần mềm bệnh án điện tử. Bác sĩ sẽ xem các kết quả trên bệnh án điện tử, mà không phải in ra, vừa tiết kiệm được tiền in ấn, còn điều dưỡng thì không phải chạy đến Khoa Chẩn đoán hình ảnh - Xét nghiệm lấy phim hay kết quả siêu âm, xét nghiệm như trước nữa. Do thời gian có hạn, nên đến hiện tại, chúng em vẫn chưa hoàn thiện được do những bất cập liên quan đến ký số điện tử. Trong tương lai chúng em sẽ phát triển module lên những chiếc máy tính bảng để tiện lợi cho công việc của bác sĩ, điều dưỡng…

**5.2 Hướng phát triển**

Như đã đề cập ở phần trên, thì hiện nay module bệnh án điện tử này của em vẫn còn hạn chế lớn đó là: chưa tự tổng kết được,chưa thể ký số được, triển khai lên máy tính bảng để tiện lợi cho công việc của bác sĩ, điều dưỡng. Vì vậy, trong tương lai gần em muốn khắc phục được vấn đề này.

* Tạo một nút tổng kết bệnh án, khi bấm vào hệ thống sẽ tổng kết bệnh án và sẽ hiển thị ra hồ sơ khám chưa bệnh và tất cả những giấy tờ liên đến bệnh án mà không phải bấm vào từng mục trong danh sách giấy tờ để kiểm tra như trước
* Hoàn thiện được chức năng kí số để đáp ứng được tất cả các yêu cầu liên quan đến hồ sơ bệnh án điện tử mà bộ y tế đề ra trong thông tư Số: 46/2018/TT-BYT, ban hành quy định về hồ sơ bệnh án điện tử.
* Hoàn thiện được module trên những chiếc máy tính bảng, để tiện lợi cho công việc của bác sĩ, điều dưỡng, có thể làm việc ở nhiều nơi, không còn phụ thuộc vào máy tính.

Em hy vọng sẽ tiếp tục theo sát dự án này của công ty để giải quyết vấn đề những thiếu sót và nhược điểm, cũng như nâng cao kỹ năng và phương pháp luận của bản thân.

**TÀI LIỆU THAM KHẢO**

1. Thông tư 46/2018/TT-BYT,*Thông tư số 46/2018/TT-BYT*, 28/12/2018,[https://thuvienphapluat.vn/van-ban/Cong-nghe-thong-tin/T](https://thuvienphapluat.vn/van-ban/Cong-nghe-thong-tin/Thong-tu-46-2018-TT-BYT-su-dung-va-quan-ly-ho-so-benh-an-dien-tu-391438.aspx) [hong-tu-46-2018-TT-BYT-su-dung-va-quan-ly-ho-so-benh-a](https://thuvienphapluat.vn/van-ban/Cong-nghe-thong-tin/Thong-tu-46-2018-TT-BYT-su-dung-va-quan-ly-ho-so-benh-an-dien-tu-391438.aspx) [n-dien-tu-391438.aspx](https://thuvienphapluat.vn/van-ban/Cong-nghe-thong-tin/Thong-tu-46-2018-TT-BYT-su-dung-va-quan-ly-ho-so-benh-an-dien-tu-391438.aspx)
2. Điều 59 Luật số: 40/2009/QH12,*Luật khám bệnh, chữa bệnh, Luật số: 40/2009/QH12*, 31/11/2009,[https://thuvienphapluat.vn/van-ban/The-thao-Y](https://thuvienphapluat.vn/van-ban/The-thao-Y-te/Luat-kham-benh-chua-benh-nam-2009-98714.aspx)

[-te/Luat-kham-benh-chua-benh-nam-2009-98714.aspx](https://thuvienphapluat.vn/van-ban/The-thao-Y-te/Luat-kham-benh-chua-benh-nam-2009-98714.aspx)

1. Thông tư số 54/2017/TT-BYT,*Thông tư số 54/2017/TT-BYT*, 29/12/2017,[ht](https://thuvienphapluat.vn/van-ban/Cong-nghe-thong-tin/Thong-tu-54-2017-TT-BYT-Bo-tieu-chi-ung-dung-cong-nghe-thong-tin-tai-cac-co-so-kham-chua-benh-373292.aspx) [tps://thuvienphapluat.vn/van-ban/Cong-nghe-thong-tin/T](https://thuvienphapluat.vn/van-ban/Cong-nghe-thong-tin/Thong-tu-54-2017-TT-BYT-Bo-tieu-chi-ung-dung-cong-nghe-thong-tin-tai-cac-co-so-kham-chua-benh-373292.aspx) [hong-tu-54-2017-TT-BYT-Bo-tieu-chi-ung-dung-cong-ngh](https://thuvienphapluat.vn/van-ban/Cong-nghe-thong-tin/Thong-tu-54-2017-TT-BYT-Bo-tieu-chi-ung-dung-cong-nghe-thong-tin-tai-cac-co-so-kham-chua-benh-373292.aspx) [e-thong-tin-tai-cac-co-so-kham-chua-benh-373292.aspx](https://thuvienphapluat.vn/van-ban/Cong-nghe-thong-tin/Thong-tu-54-2017-TT-BYT-Bo-tieu-chi-ung-dung-cong-nghe-thong-tin-tai-cac-co-so-kham-chua-benh-373292.aspx)
2. Lan Phương, *“HÀNH TRÌNH TRIỂN KHAI BỆNH ÁN ĐIỆN TỬ TẠI VIỆT NAM"*, 03/2021,[https://giaiphapviettel.vn/tin-tuc/446/han](https://giaiphapviettel.vn/tin-tuc/446/hanh-trinh-trien-khai-benh-an-dien-tu-tai-viet-nam--mo-duong-cho-y-te-thong-minh.html) [h-trinh-trien-khai-benh-an-dien-tu-tai-viet-nam--mo-d](https://giaiphapviettel.vn/tin-tuc/446/hanh-trinh-trien-khai-benh-an-dien-tu-tai-viet-nam--mo-duong-cho-y-te-thong-minh.html) [uong-cho-y-te-thong-minh.html](https://giaiphapviettel.vn/tin-tuc/446/hanh-trinh-trien-khai-benh-an-dien-tu-tai-viet-nam--mo-duong-cho-y-te-thong-minh.html)