|  |  |
| --- | --- |
|  | **MINISTRY OF INDUSTRY AND TRADE**  **HANOI UNIVERSITY OF INDUSTRY**  **---------------------------------------** |
| **PHAM QUANG HUY** | **UNIVERSITY GRADUATION PROJECT** |
| **INFORMATION SYSTEMS INDUSTRY** |
| **LEARN AND BUILD ELECTRONIC MEDICAL RECORDS HONG NGOC HOSPITAL** |
| **INFORMATION SYSTEMS** | **GVHD: TS. Le Truong Giang** |
| **Student: Pham Quang Huy** |
| **Student ID: 2019600247** |
|  |
|  | **Hanoi – In 2024** |

**ACKNOWLEDGMENTS**

First of all, I would like to express my deep thanks to Mr. Dr. Le Truong Giang - Lecturer of Hanoi University of Industry, who facilitated the research and guided me very enthusiastically throughout the process of making the graduation project. Thanks for the enthusiastic help and comments of brothers and sisters and colleagues at DigiHC company. Finally, I would like to thank my family and friends for always caring, encouraging and motivating me to successfully complete my graduation project.

**SUMMARY OF THE PROJECT**

According to Clause 1, Article 59 of the Law on Medical Examination and Treatment 2009, each patient has only one medical record during each medical examination and treatment at a medical examination and treatment establishment. This medical record can be made in paper or electronic copies. However, in practice, traditional paper-based medical records are still widely used in most hospitals. This causes many difficulties in accessing, sharing information and searching patient data.

Therefore, in this project, I choose to use normal web technology to build an electronic medical record system as prescribed in Circular 46 of the Ministry of Health issued in 2018. This choice was made based on the convenience and ubiquity of web technology, which allows access and use across multiple platforms and devices. Her solution will create an easy-to-use web interface for medical facilities to record, store and access medical records.

The main contribution of this project is to build an electronic medical record system using web technology, helping to improve medical information management and enhance the quality of health care. The intended result is to create an efficient tool for storing and retrieving medical records, while optimizing workflows and enhancing linkages between medical facilities.

**TABLE OF CONTENTS**

[**CHAPTER 1. TOPIC INTRODUCTION** 1](#_Toc167612150)

[**1.1** **Set a problem** 1](#_Toc167612151)

[**1.2** **About HIS and EMR** 1](#_Toc167612152)

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

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

[**1.3** **Objectives and scope of the topic** 3](#_Toc167612155)

[**1.4** **Solution-oriented** 4](#_Toc167612156)

[**1.5** **Project layout** 5](#_Toc167612157)

[**CHAPTER 2. SURVEY AND REQUIREMENTS ANALYSIS** 7](#_Toc167612158)

[**2.1 Current survey** 7](#_Toc167612159)

[**2.2 Functional overview** 7](#_Toc167612160)

[**2.2.1 General use case chart** 7](#_Toc167612161)

[2.2.2 Decay case chart: Patient search 8](#_Toc167612162)

[2.2.3 Decay case chart: Medical record management 9](#_Toc167612163)

[**2.2.4 Decay case chart: Related document management** 9](#_Toc167612164)

[**2.2.5 Business process** 10](#_Toc167612165)

[**2.3 Functional specifications** 11](#_Toc167612166)

[**2.3.1 Use case specification: Find patients** 11](#_Toc167612167)

[**2.3.2 Use case specification: Create medical records** 12](#_Toc167612168)

[**2.3.3 Use case specification: Create relevant documents** 12](#_Toc167612169)

[**2.4 Non-functional requirements** 13](#_Toc167612170)

[**2.4.1 Technical requirements** 13](#_Toc167612171)

[**2.4.2 UI/UX requirements** 14](#_Toc167612172)

[**2.4.3 Security and privacy requirements** 14](#_Toc167612173)

[**2.4.4** **Requires additional scalability** 14](#_Toc167612174)

[**2.5** **Functional requirements** 15](#_Toc167612175)

[**2.5.1 Patient Information Management Requirements** 15](#_Toc167612176)

[**2.5.2 Requirements for managing the medical examination process** 15](#_Toc167612177)

[**CHAPTER 3: TECHNOLOGY USE** 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)

[**CHAPTER 4: EXPERIMENTATION AND EVALUATION** 24](#_Toc167612185)

[**4.1 Architectural design** 24](#_Toc167612186)

[**4.1.1 Software architecture selection** 24](#_Toc167612187)

[**4.1.2 Design overview** 26](#_Toc167612188)

[**4.2 Detailed design** 27](#_Toc167612189)

[**4.2.1 Interface design** 27](#_Toc167612190)

[**4.2.2 Database design** 29](#_Toc167612191)

[**4.3 Build the app** 29](#_Toc167612192)

[**4.3.1 Libraries and tools used** 29](#_Toc167612193)

[**4.3.2 Achievements** 30](#_Toc167612194)

[**4.3.3 Illustration of main functions** 30](#_Toc167612195)

[**4.4 Testing** 38](#_Toc167612196)

[**4.5 Deployment** 42](#_Toc167612197)

[**4.5.1 End of deployment** 43](#_Toc167612198)

[**4.5.2 Summary of IHS specifications** 45](#_Toc167612199)

[**CHAPTER 5. CONCLUSION AND DIRECTION OF DEVELOPMENT** 46](#_Toc167612200)

[**5.1 Conclusion** 47](#_Toc167612201)

[**5.2 Development direction** 49](#_Toc167612202)

**LIST OF DRAWINGS**

[Figure 2.1: General use case chart 7](#_Toc167611870)

[Figure 2.2: Patient search case chart 8](#_Toc167611871)

[Figure 2.3: Medical case management chart 9](#_Toc167611873)

[Figure 2.4: Related document management case chart 9](#_Toc167611874)

[Figure 2.5: Business process of electronic medical records 10](#_Toc167611875)

[Figure 2.6: System function chart 16](#_Toc167611876)

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

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

[Figure 3.3: Golang logo 21](#_Toc167611879)

[Figure 3.4: MongoDB logo 22](#_Toc167611880)

[Figure 4.1: General package chart 26](#_Toc167611881)

[Figure 4.2: Package Detail Design 27](#_Toc167611882)

[Figure 4.3 depicts the interface layout of the page managing medical records and related documents. 28](#_Toc167611883)

[Figure 4.4: Medical record page interface model 28](#_Toc167611884)

[Figure 4.5: System databases 29](#_Toc167611885)

[Figure 4.6: Search for patients 30](#_Toc167611886)

[Figure 4.7: Medical record creation function 31](#_Toc167611887)

[Figure 4.8: Medical record creation function 31](#_Toc167611888)

[Figure 4.9: Medical record creation function 32](#_Toc167611889)

[Figure 4.10: Viewing medical records 33](#_Toc167611890)

[Figure 4.11: Viewing medical records 33](#_Toc167611891)

[Figure 4.12: View related documents 34](#_Toc167611892)

[Figure 4.13: Printing slips, exporting PDF files 35](#_Toc167611893)

[Figure 4.14: Deployment architecture 44](#_Toc167611894)

**LIST OF TABLES**

[Table 2.1: Patient search case specification 10](#_Toc167112646)

[Table 2.2: Specification of medical record creation cases 10](#_Toc167112647)

[Table 2.3: Specification of medical record creation cases 11](#_Toc167112648)

[Table 4.1: List of libraries and tools used 23](#_Toc167112649)

[Table 4.2: Functional testing 28](#_Toc167112650)

**LIST OF TERMS AND ACRONYMS**

|  |  |
| --- | --- |
| **Terminology** | **Meaning** |
| API  BHXH/BHYT BMI  BN  BYT Database  CT  DICOM  DNS  EMR  IHS  HL7  HTML  HTTT  ID  IT, IT  KCB  LIS  MRI  PACS  RIS  TT | Application Programming Interface  Social Insurance/Health Insurance  Body mass index Patients  Ministry of Health  Imaging Diagnosis Subclinical  Databases  Computed tomography  Data Processing Standards  Domain Name System  Electronic medical records  International Standard Hospital Management Information System provides a standard protocol for managing, exchanging and integrating health data between health information systems to support medical operations  (Health Level Seven).  HyperText Markup Language  Information systems  Identifier  Information Technology  Medical examination and treatment  Laboratory Information Management Software  Magnetic resonance imaging  Medical image storage and communication system  Radiological Information Systems  Circular |

# **CHAPTER 1. TOPIC INTRODUCTION**

* 1. **Set a problem**

Today, information technology is developing rapidly, leading to the network of software also increasing both in quantity on an expanded scale and in software quality in depth.

In the medical field, IT applications are developing strongly, health information systems HIS, RIS, LIS, PACS are widely and effectively deployed, built based on DICOM and HL7 standards towards unification of data information exchange and processing between medical facilities for management, diagnosis, treatment and health care.

However, currently the implementation of EMR (electronic medical records) in hospitals is very limited. Traditional medical records based on paper and physical form are still widely used in healthcare settings. It causes many difficulties for managing, storing and retrieving information in medical records, consuming a lot of space to store medical records. Problems such as loss, loss of information, difficulty finding and sharing information carry the risk of errors in diagnosis and treatment.

The successful implementation of electronic medical records brings many important benefits such as: allowing the storage of patients' medical information in electronic form, helping to save time and storage space. Patient medical information can be easily retrieved from any locality by those who have access. This enhances availability and efficiency in health information management.

* 1. **About HIS and EMR**
     1. **HIS (Hospital Information System)**

A hospital information system commonly known as "Hospital Management System"; serving the management and administration at the Hospital with the main functions: managing patient information and medical history; management of patients for inpatient and outpatient examination and treatment, management of medical records, pharmacy, finance, hospital fees, equipment, medical supplies, personnel... Today, HIS is an optimization tool in executive management; for therapeutic purposes; for research and training; statistics, forecasts, redundancy... at Hospitals.

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

According to Article 59 of the Law on Medical Examination and Treatment 2009, medical records are medical, medical and legal documents; Each patient has only one medical record during each medical examination and treatment at a medical examination and treatment facility. Inpatient and outpatient patients in medical examination and treatment facilities must have their medical records made. Medical records shall be made in paper or electronic copies and must be clearly and fully stated in the medical records. Medical records include documents and information related to patients and the process of medical examination and treatment.

Article 2 of Circular 46/2018/TT-BYT also stipulates that electronic medical records made, updated, displayed, digitally signed and stored by electronic means meeting the provisions of Circular 46/2018/TT-BYT are as valid as paper medical records specified in Article 59 of the Law on Medical Examination, Healing 2009.

Talk about an electronic medical record

* Firstly, based on the text of the law: the highest level document is the circular (Circular 54/2017/TT-BYT – part on electronic medical record criteria and Circular 46/2018/TT-BYT: regulation on electronic medical records).
* 2nd, to a lesser extent, there are decisions: Currently, there are 3 decisions of the Ministry of Health to issue forms on medical records, on medical record keeping. These are the decisions: (i)4069/2001/QD-BYT dated 28/09/2001, (ii)QĐ 3443/QD- BYT dated 21/09/2011, (iii)QĐ 3730 BYT dated 5/08/2021. These 3 decisions specify and describe in detail all medical records that should now be stored in what form they are based on.

Regarding legal bases, it must meet the legal bases, to design software instead of storing paper medical records in accordance with circulars and decisions, now transferred to the machine.

So if you move up on the machine, what does it mean to do? That is, doctors do not have to use their hands to write, but everything must be manipulated on the computer: print on the machine, digitally sign on the machine and store and share, then it will finally meet the standards of electronic medical records according to Circular 54.

Therefore, EMR's biggest goal is to digitize all medical records and papers. That is, doing everything on the machine, not having to fight back the entered information and administrative information of the patient. Digitally signed, summarized, and shared with related units including patients, insurance, doctors.

* 1. **Objectives and scope of the topic**

As I mentioned in section 1.1, although there has been a circular issued on regulations on the development of electronic medical records, the number of hospitals that have deployed electronic medical records is still very small. Due to difficulties such as the database of hospitals is not synchronized and not unified between hospitals, the data interconnection still faces many difficulties. According to Circular No. 46/2018/TT-BYT, by 2030, all medical examination and treatment facilities nationwide must deploy electronic medical records.

To solve the above problem, participating in promoting the deployment of electronic medical records, the technical director of my company decided to produce an EMR module to manage medical records and papers related to patients' medical records.

Currently, this module will be tightly integrated into the company's deployed HIS software. This decision was made due to the robustness of the HIS architecture (the architecture of HIS will be presented in the ...) section, allowing it to serve a large number of clinics.

Therefore, with a team consisting of a software developer who is you, a support person and an IT BA (who is also a designer). Our team had to develop a module that retained the most important features of the old system, while improving it. The goal is to build a user-friendly website that can be used on desktop, laptop, and ipad screens. With the main functions are:

* Allow users to search for inpatient/outpatient information
* Click function to view patient details'
* Function to create a new set of medical records
* Allows you to select the type of medical record
* View unsigned medical records
* View signed medical records
* Function to create new slips in medical records
* Ticketing function
* Medical record printing function
* View function to view details of medical records/papers
* Function to delete slips in medical records
  1. **Solution-oriented**

From the problem requirements in Section [1.3,](#_bookmark6)  I propose solution orientations for some of the following problems as shown in Table 1.1:

|  |  |  |
| --- | --- | --- |
| **Request** | **Problem to solve** | **Workaround** |
| Interface | The interface must ensure intuitiveness and ease of use for users. | Reference and interface design with the same layout as HIS. |
| Medical record management | There must be sufficient information fields such as paper medical records. | Design medical records according to the form of the ministry of health. |
| Manage relevant documents | All relevant documents must be compiled during treatment. | Closely link HIS to link and retrieve data during medical examination and treatment. |

* 1. **Project layout**

This project report will be arranged into 6 main chapters and the content of the chapters is presented in turn as follows: In chapter 1: is an introduction to the topic, here I present the problems to be solved that the project can be implemented in accordance with reality in life. Based on that, I learn about the issues and give a reasonable development direction that I myself carry out to complete the graduation project 1 completely and best.

In chapter 2, I will present the practical situation of myself having researched and surveyed the needs of businesses. Therefore, I have identified all the remaining disadvantages and limitations to build my website in the most effective way. Next, we will present the overview of the function of the website and describe the main functions of the system with use case charts to be the most comprehensive to give the non-functional requirements of the system.

In chapter 3, I will talk about the technologies I have applied to build my system software. In order to be able to complete the project in the most convenient way, I also tried to choose some of my own technology solutions through the suggestions of the brothers and sisters in the company to meet the necessary needs to develop the product in the most complete way.

In chapter 4, I will implement the website and present the results of the development process. In which the processes are presented in the following stages: (i) System design, (ii) System detailed design, (iii) Application construction, (iv) Product testing and deployment.

Finally, in chapter 5, I would like to conclude the content of the graduation project, the problems and things that I have tried to do and not done in the process of making the graduation project and the direction of developing my own product in the future in the largest way.

# **CHAPTER 2. SURVEY AND REQUIREMENTS ANALYSIS**

## **2.1 Current survey**

As mentioned in section [1.1,](#_bookmark2) the application of electronic medical records at medical facilities is still quite limited. Because this implementation is relatively new, there is very little information about the technology used, as well as the business process of electronic medical records in these hospitals. According to my research, the hospitals that have implemented most of them separate the management of papers and medical records. Therefore, in this project, our team will design a module that can manage both medical records and all related documents of medical records, so that it becomes an electronic medical record in accordance with the requirements of the Ministry of Health. Below I will present an overview of this module.

## **2.2 Functional overview**

### **2.2.1 General use case chart**

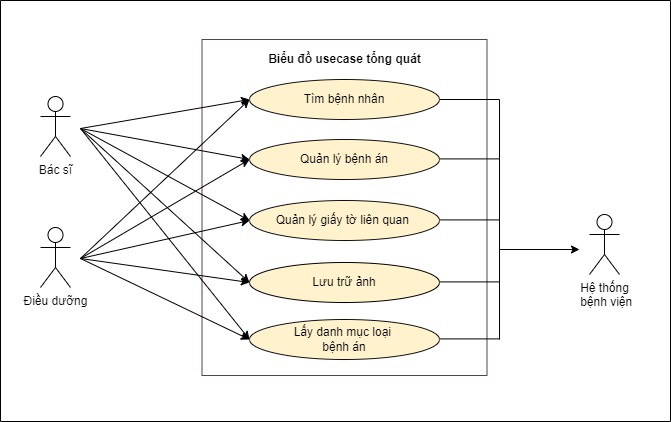
Through the above analysis, the functions of the system can be represented by a use case graph as shown in Figure [2.1:](#_bookmark13)

Figure 2.1: General use case chart

### 2.2.2 Decay case chart: Patient search

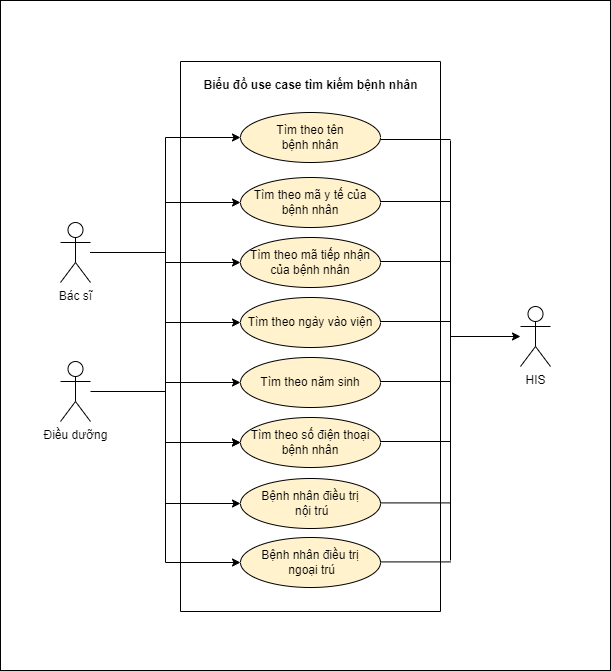


Figure 2.2: Patient search case chart

Figure 2.2 depicts patient search cases. Doctors and nurses will choose to search for inpatient or outpatient patients. The patient can then be searched according to the patient's administrative information such as: name, year of birth, phone number, date of admission, medical code or admission code of the patient. This information of the patient will be retrieved HIS.

### 2.2.3 Decay case chart: Medical record management

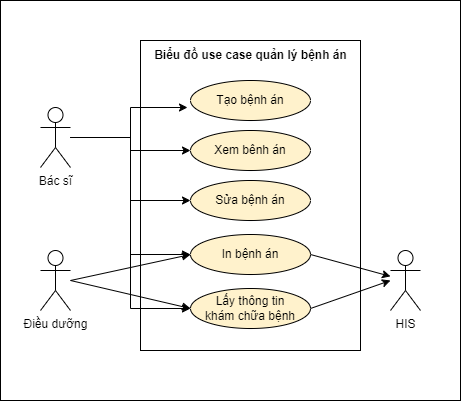


Figure 2.3: Medical case management chart

Figure [2.3](#_bookmark17) depicts the function of medical record management. Doctors have the right to create new medical records based on medical examination and treatment information from HIS. The nurse can then view the medical record and print the saved medical record

### **2.2.4 Decay case chart: Related document management**

Figure 2.4: Related document management case chart

Figure [2.4](#_bookmark19) depicts the function of managing medical records. Doctors and nurses obtain medical examination and treatment results from HIS or create new relevant documents, then can view and print relevant medical documents.

### **2.2.5 Business process**

Figure 2.5: Business process of electronic medical records

Figure [2.5](#_bookmark21) depicts the business process of making electronic medical records. Initially, patients arriving at the hospital will be received by medical staff and assigned services. After that, the patient will be examined as indicated. If the patient does not have a disease or has a disease that does not want treatment, he will be discharged without medical records. If the patient has a disease and wants treatment, they will be able to create medical records and carry out procedures for hospitalization and treatment. During the treatment of the disease, you will simultaneously do relevant documents such as ultrasound papers, scans, surgeries ... This process is performed and saved on HIS. When the patient finishes treatment or is discharged from the hospital in the form of outpatient treatment or transfer, the doctor will enter the patient's medical examination and treatment information into the medical record. Synthesize relevant documents during medical examination and treatment into the patient's medical records.

**2.3 Functional specifications**

**2.3.1 Use case specification: Find patients**

|  |  |  |
| --- | --- | --- |
| Name Use case | Search for patients | |
| Agent | Doctor, Nursing | |
| Pre-conditions | 1. The agent is logged on to the system 2. The patient already exists in IHS 3. The patient has been created a medical record from IHS | |
| Post-conditional | Display on-demand patient search results | |
| Main event stream | Agent | System |
| 1. Access agent to   Medical record page | 2. The system displays the results page with search filters |
| 3. Search agent by  condition | 4. Search result display system |
| Alternate event streams | Not | Not |

Table 2.1: Patient search case specification

**2.3.2 Use case specification: Create medical records**

|  |  |  |
| --- | --- | --- |
| Name Use case | Create a new medical record | |
| Agent | Physician | |
| Pre-conditions | The agent selected the patient after the search | |
| Post-conditional | A new medical record is created | |
| Main event stream | Agent | System |
| 1. Agent to select disease  multiplication after search | 2. Chart display system  Medical record form |
| 3. Data entry agent  for data fields in a form | 4. The system saves data to  form |
| 5. Save click agent | 6.Disease record system  Go to the database and select the newly created medical record |
| Alternate event streams | Not | Not |

Table 2.2: Specification of medical record creation cases

**2.3.3 Use case specification: Create relevant documents**

|  |  |  |
| --- | --- | --- |
| Name Use case | Create relevant documents | |
| Agent | Doctors, nurses | |
| Pre-conditions | 1. The agent selected the patient after the search 2. The patient has been created a medical record and saved | |
| Post-conditional | Some related documents created | |
| Main event stream | Agent | System |
| 1. Agent click select disease   Multiply after searching and selecting a view of unsigned medical records | 2. List display system  Book how relevant paperwork needs to be wrong |
| 3. Paper type selection agent  Related sheets to do | 4. Chart display system  Sample of document |
| 5. Data Entry Agent  for the required data fields in the form | 6.The system saves data to  form |
| 7. Save click agent | 4. Information saving system  relevant documents to the database |
| Alternate event streams | Not | Not |

Table 2.3: Specification of medical record creation cases

**2.4 Non-functional requirements**

**2.4.1 Technical requirements**

The system must be usable, maintainable and scalable, easy to convert databases

**2.4.2 UI/UX requirements**

The interface must be designed to be simple, accessible to medical staff, and must be designed similarly to the HIS system that the hospital is using. The theme color should be able to change depending on each clinic. Can be compatible on many browsers and many different screens.

**2.4.3 Security and privacy requirements**

Patient medical data must be encrypted both at rest and in transmission to ensure it is not subject to unauthorized access. The system needs to manage and assign access permissions to each user based on their role (doctor, nurse, administrator...). All data access and correction activities must be recorded for monitoring and auditing when necessary. The system must use strong authentication methods (such as two-factor authentication) to ensure only authorized users can access the system. Ensure that all data transmitted over the network is secure, such as using the HTTPS protocol to encrypt the transmitted data. The system must have a mechanism for periodic data backup and the ability to quickly recover data in case of loss or failure.

* + 1. **Requires additional scalability**

The system should be designed in a modular architecture, allowing adding, removing or upgrading components without affecting the entire system. Each module can take on a specific function such as patient information management, examination history management, medication management, etc. Deploy the system across multiple servers and data centers to ensure high availability and load distribution. Use horizontal scalable database management systems such as Apache Cassandra, MongoDB, or Google Bigtable. Perform load tests to evaluate system performance under different load conditions. Perform stress tests to assess the maximum limit of the system and detect potential bottlenecks.Ensure the system can recover from peak load situations without serious problems.

* 1. **Functional requirements**

### **2.5.1 Patient Information Management Requirements**

* The system must allow the creation, storage and management of patient records including personal information (name, age, gender, address, phone number, email).
* Support to store health insurance information and other relevant information such as insurance record number, insurance expiration date.
* Archives the patient's medical history including acquired diseases, surgeries performed, family history, and social history.
* Manage information about patients' allergies to drugs, foods, and environmental factors.
* Store the patient's emergency contact information, including the name, relationship, phone number, and address of the contact.

### **2.5.2 Requirements for managing the medical examination process**

* Record the patient's treatment history, including medical visits, diagnosis, treatment, and test results.
* Assist in storing and accessing medical documents such as laboratory reports, diagnostic images, and doctor's notes.
* The system must allow the doctor to enter and update diagnoses, treatment plans, and notes during the examination.
* Support ICD-10 disease code lookup tool to standardize diagnosis and treatment.
* Support the storage and access of test results, including results from laboratories and diagnostic imaging equipment.
* Send notifications to your doctor when new test results are available and allow test results to be compared over different points in time.

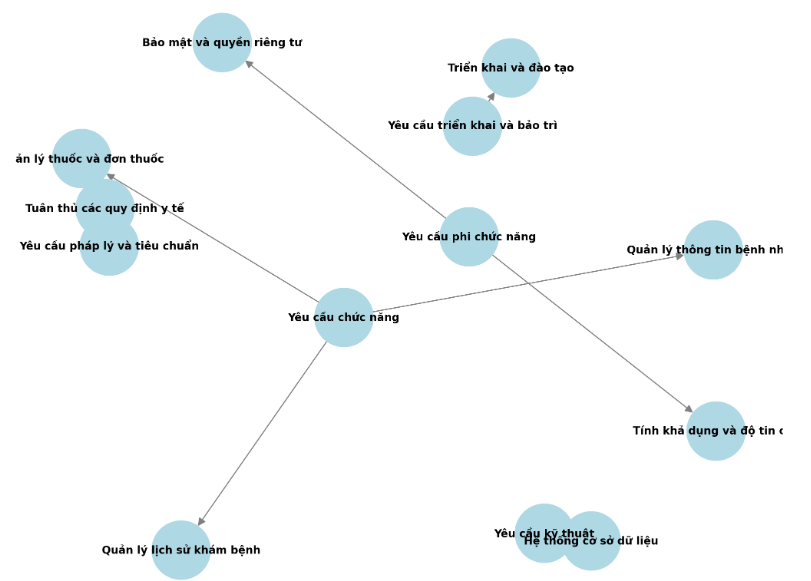


Figure 2.6: System function chart

Above I have introduced an overview of our system analysis. In the next chapter, I will cover the technologies that we chose to use to build this module

**CHAPTER 3: TECHNOLOGY USE**

Chapter 2 concludes with a survey and an overview analysis of the electronic medical record management module. Below, I will briefly describe the technologies that I used for this project.

This system is built to operate on the Web in a private network of a medical facility. The requirements are only related to data management, and such a typical Web application, including the front-end, server, and database, should be sufficient to solve these problems.

**3.1 Front-end**

A front-end is the part of the user interface (UI) of a Web application where users can see and interact with the application's data. A mandatory tool for Web development is the Hyper-Text Markup Language (HTML) markup language for structuring and displaying Web pages, often supplemented by Cascading Style Sheets (CSS) to visually customize the look. To make pages interactive, JavaScript (JS) is used to process programming logic. The main tool for front-end developers is HTML-CSS-JS.

There are many high-level tools, called frameworks, that are used to speed up front-end development. For this project, I used Angular (specifically Angular 14), a popular framework developed and maintained by Google and its community of collaborators.

**3.1.1 Angular**

Angular is built on a number of strict rules, which almost does not tolerate any flaws. Its components are structured into subcomponents according to a hierarchical structure, similar to how HTML arranges elements. This helps keep the app organized and easy to manage. Components in Angular can be easily created and embedded into each other, making it easy to reuse them. Another reusable feature that Angular offers is "dependency injection" (DI): It allows you to reuse services from other components without much difficulty. This makes it easy to share global data, communicate with servers, or perform complex project-wide computational functions.



Hình 3.1: Logo Angular

Components and services in Angular, along with other parts, can be bundled into one module. This helps to make the project modular - all system functions can be separated and reused in a single package, and they can be reused in other Angular projects. This makes the project easy to divide into separate sections and easy to reuse in different projects. Using TypeScript contributes to the modularity of this project - TypeScript is an extension of JavaScript that supports strict data types. TypeScript helps the application maintain consistency and ease of maintenance, while applying Object-Oriented Programming (OOP) rules well.

This framework was chosen because my company is familiar with it. Its strict rules reduce the number of compilation errors during development and make it easier to detect. Although it is quite difficult to learn, Angular has very complete documentation of all the functions available to developers.

Some other popular alternatives to Angular include libraries like React and other frameworks like Vue. React and Vue are simpler to code and provide higher performance, however, they both follow the same Model-View-Controller (MVC) generalized architecture as Angular (more on MVC in the **??**). ). React also allows building mobile apps using the same principles of React as React Native, which can be deployed on multiple devices, including mobile devices and televisions.

Angular gives our team a powerful tool for building apps. However, the user interface will not only be built with HTML but will include other technologies.

**3.1.2 Angular Material**

Angular Material is a user interface (UI) library that provides fully Angular-enabled components, with layouts and styling that follow Material Design's design specifications. This library includes diverse UI elements such as forms, navigation cards, buttons, dialogs, and pagination, optimized for performance, reliability, and accessibility. These components can be easily embedded into Web pages at will. In addition, Angular Material also supports "theming", which allows changing color themes, text typography, and more in the project more easily.

Hình 3.2: Logo Angular material

Angular Material is a great and convenient choice for our team, since we already work with Angular. Although most form fields and controls have had to be redesigned to better fit the HIS system, the existing navigation tabs, buttons, and pagination have proven useful in speeding up UI implementation. Dialogs are easily customizable, allowing entire components to be embedded within them.

This interface library was chosen for its ease of use and popularity. It is partly maintained by Google, who also maintains part of the Angular framework. At that time, our team had a lot of experience with DevExtreme, a library that provided more components, but also heavier when building early versions of the program. I was then proposed to research and use Angular Material in the system.

The front-end is the user interface department, which allows users to interact with the system visually. The user interface will need to communicate with the back-end through requests, which are necessary to process and store the data that the user has interacted with.

**3.2 Back-end**

The backend is an important part of computer applications, especially Web applications. It refers to business components and logic that are handled on the server and are not directly visible to the user. In the context of Web applications, the backend handles requests from the frontend (user interface) and performs complex tasks such as data processing, database management, computation, security, and handling business logic.

The backend typically includes servers, databases, and other data processing systems such as caching, external data retrieval, and user authentication. It is where data is stored and processed, ensuring the system operates in an efficient, secure and reliable manner. In this project, I was proposed to use Golang for servers and MongoDB as databases.

**3.2.1 Golang**

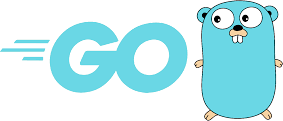
Golang (Go Language) is a programming language created and designed by Google employees. However, Golang exists as an open source project and is not owned by Google. Golang is built to make programming easy to read, simple, efficient, and achieve high performance. Golang is also known as a static typed language. This means that everything that belongs to Golang must have a data type, and it is opposed to dynamic typed languages like Python or Javascript. Golang is similar to Java or C++ because it is also a language used for programming purposes. However, simplicity is what sets Golang apart from other programming languages. It possesses quite streamlined syntax and has a great degree of similarity to C++.

Figure 3.3: Golang logo

The choice of Golang as the server was because it was used for the company's HIS system and it was very familiar to my company. Using Golang, we built a simple application programming interface (API) server, which allows the front-end to interact with it using Hypertext Transport Protocol (HTTP).

Alternatives to building back-end servers include JavaScript in the NodeJS environment, however, the NodeJS server is not easy to scale due to the use of only one thread of execution. Another option is Microsoft's .NET framework, which can improve system maintenance by design for Object-Oriented Programming.

Overall, Golang is compact, sharp and possesses very high compatibility. As a result, software programs operated by Golang are incredibly fast from a variety of conditions.

**3.2.2 MongoDB**

MongoDB is an open source and non-relational database management system (DBMS), developed by MongoDB Inc. Mon-goDB uses a flexible and flexible JSON (JavaScript Object Notation) storage type to store data, instead of the traditional table structure like SQL database management systems. Some highlights about MongoDB:

* Document orientation: MongoDB stores data as JSON (BSON) documents, which makes it flexible structured data storage and easy to scale data without changing the database structure.
* Scalability: MongoDB supports easy scalability by dividing data into replica sets or clusters, which increases the processing and load capacity of the system.
* Good integration with programming languages: MongoDB provides drivers for many popular programming languages, making it easy to integrate with applications that use these languages.
* Support for rich queries: MongoDB provides powerful queries for data retrieval and processing, including complex queries such as range queries, conditional queries, and geospatial queries.
* Good integration with the cloud: MongoDB supports integration with cloud services such as Amazon Web Services (AWS) and Microsoft Azure, making it easy to deploy and manage databases in the cloud environment.

Figure 3.4: MongoDB logo

With the above advantages, MongoDB has become a popular database management system and is widely used in Web and mobile applications.

The use of MongoDB as a database is because my company's HIS system has previously used it, so using MongoDB makes it more convenient to link data to the old system.

In this chapter, I have presented and the main technologies I used in this project. Next, in chapter 4, I will present a detailed description of the architecture and design of this module.

**CHAPTER 4: EXPERIMENTATION AND EVALUATION**

**4.1 Architectural design**

**4.1.1 Software architecture selection**

**a) Model – View – Controller(MVC)**

MVC is an extremely popular and growing model in web programming. The MVC model consists of 3 parts: View, Model and Controller.

* View: This section has the main task of communicating with the user, this is the component responsible for describing the presentation of data. This is the section that will be visible to users, where users can interact, giving users a broad overview of the application.
* Controller: is the component responsible for processing business, it receives requests from users from which to give data that matches the requirements of that user.
* Model: is the component responsible for data management, it stores and retrieves information from databases (such as mysql, sql server, postgreSQL,. . . ), from file systems, from other websites (apis, web services) and contains logic executed by the application.

Each Angular component primarily consists of a template file and a controller file (e.g. app.component.html and app.component.ts). In a TypeScript (TS) file, the main component object can contain many properties and methods. The values of the attributes can be displayed on the screen by calling them in the HTML file. The HTML file is then a "View" by displaying attributes or "Models". Users can interact with the View using buttons, forms, etc., and can call component methods. These methods can change the value of the component's properties, turning the object into a "Controller" that can update models.

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

A Restful API (Representational State Transfer API) is a type of software architecture used in the design of web and mobile applications. It is based on the principles of REST, an HTTP-based approach to managing and interacting with resources.

Highlights of the RESTful API include:

* Resources: APIs manage and provide access to resources such as objects, information, or in-app data. Each resource is represented by a URL (Uniform Resource Identifier).
* HTTP manipulation: RESTful APIs use HTTP methods such as GET (get data), POST (create new resources), PUT (update resources), DELETE (delete resources) to interact with resources.
* Data representation: Data is transmitted between client and server in standard formats such as JSON (Javascript Object Notation) or XML (eXtensible Markup Language), making it easy to read and understand.
* Stateless: Requests from client to server do not store the client's state. Each request is independent and independent of previous requirements, making it easy to expand and change the system.
* Single interface: RESTful APIs provide a single interface for accessing resources. Each resource is managed by a unique URI and manipulated using unique HTTP methods.
* Easy integration: RESTful APIs use popular programming languages and support integration into many different types of applications, including web, mobile, and desktop.

Using RESTful APIs helps develop applications that are flexible, scalable, and interoperable with other systems and services. It is a common standard in building distributed applications, which aids well in integrating and providing powerful services to users.

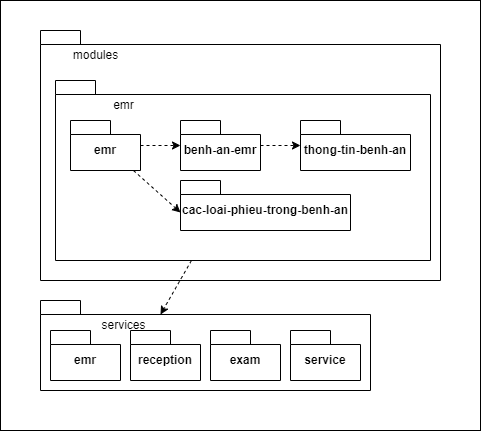
**4.1.2 Design overview**

Figure 4.1: General package chart

Figure [4.1](#_bookmark47) depicts the general package diagram of the module. In the hospital system, including many modules, my module in this project is the emr module (electronic medical record). The input will be the emr package, which contains the main screen for manipulation. The medical records used in hospitals have many types of medical records. The interface of these types of medical records will be in the patient-an-emr package. The interface of the medical record is made up of the information items of the medical record located in the thong-tin-patient-an package. The services package is the main package that interacts with the backend through APIs to create and edit information in the medical record. Users can enter the information from scratch or can obtain the patient's information from the hospital system (HIS). Includes reception information and medical examination and treatment information.

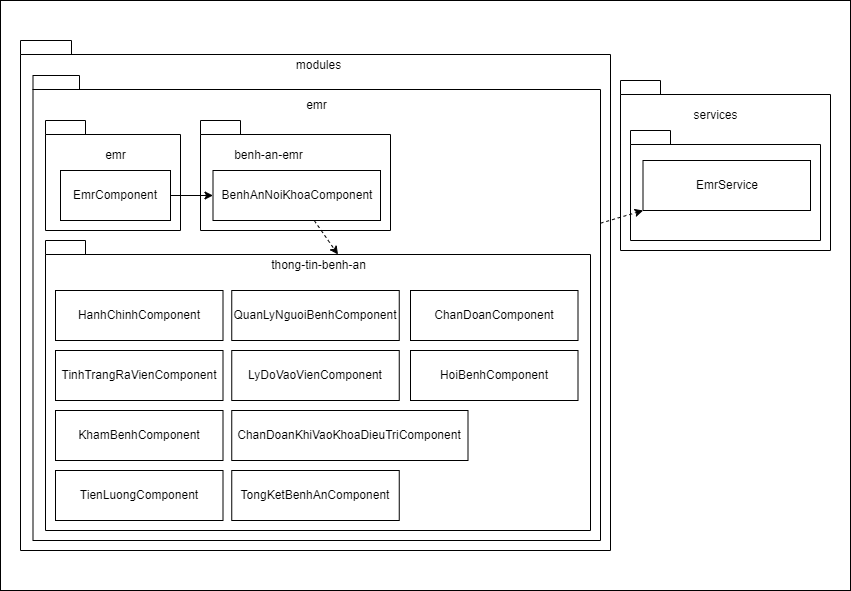
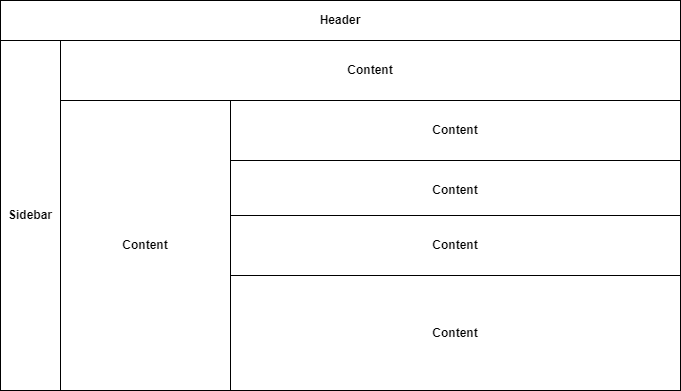
**4.1.3 Detailed package design**

Figure 4.2: Package Detail Design

## **4.2 Detailed design**

### **4.2.1 Interface design**

The system interface is designed to be displayed on various monitors. The system has been tested to display well at several resolutions such as 1920x1080, 1366x768, 1280x720, 960 x 540, usually on PC, Laptop and iPad screens.



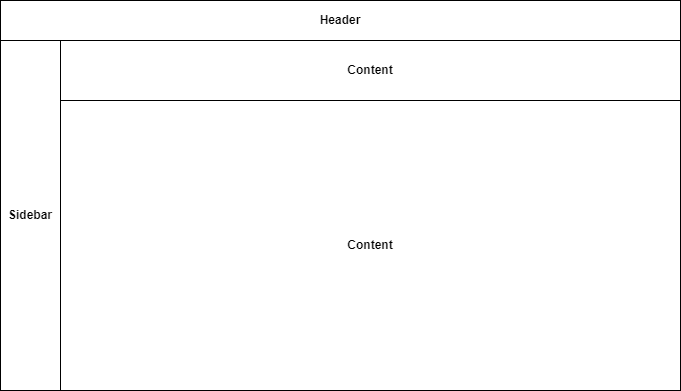
Figure [4.3](#_bookmark52) depicts the interface layout of the page managing medical records and related documents.

Figure 4.4: Medical record page interface model

All pages in the system have a layout similar to HIS, consisting of three main components: navigational sidebar, header, and content elements that display detailed content of the various functions of the sidebar option. The sidebar will provide a series of navigation menus and submenus between functions, grouped according to customer requirements. The header will contain a menu for key page actions, such as create, edit, delete, save,.. Each function has different purposes and user interactions, so the layout of the main content will be different for most pages.

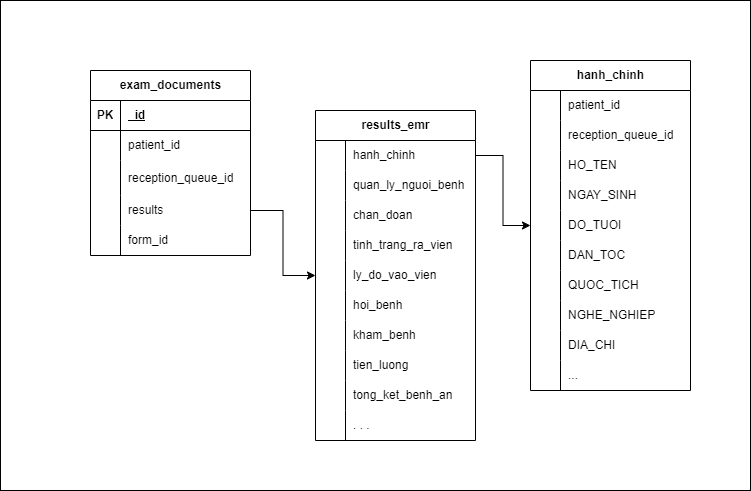
**4.2.2 Database design**

Figure 4.5: System databases

Figure 4.5 depicts the system's database. Depending on the form-id, the results will return different data fields. For example, form-id = 5045, the results will return data fields that store medical record information such as: administration-chinh, diagnosis, management-ly-patient, medical examination, . If the value of form-id is different from 5045, the results will return the relevant document circulation data fields of the medical record.

**4.3 Build the app**

**4.3.1 Libraries and tools used**

During the implementation of the project, I used the tools and supporting libraries presented in table [4.1](#_bookmark58)

|  |  |  |
| --- | --- | --- |
| **Purpose** | **Tool** | **URL Address** |
| Programming IDE | WebStorm | https://www.jetbrains.com/webstorm/ |
| Frontend programming library | Angular | https://angular.io/ |
| Backend programming library | Golang | https://go.dev/ |
| Charting | Draw.io | https://app.diagrams.net/ |

Table 4.1: List of libraries and tools used

**4.3.2 Achievements**

After completing the preliminary, our module has basically completed the basic functions that an electronic medical record needs to have as follows:

* Allows users to search for inpatient/outpatient information.
* Click function to view more detailed patient information
* Function to create a new set of medical records
* Allows you to select the type of medical record
* View unsigned medical records
* Function to create new slips in medical records
* Ticketing function
* Medical record printing function
* View function to view details of medical records/papers

**4.3.3 Illustration of main functions**

In this section will show some screenshots of the user interface of the system, with a brief explanation of those screens.

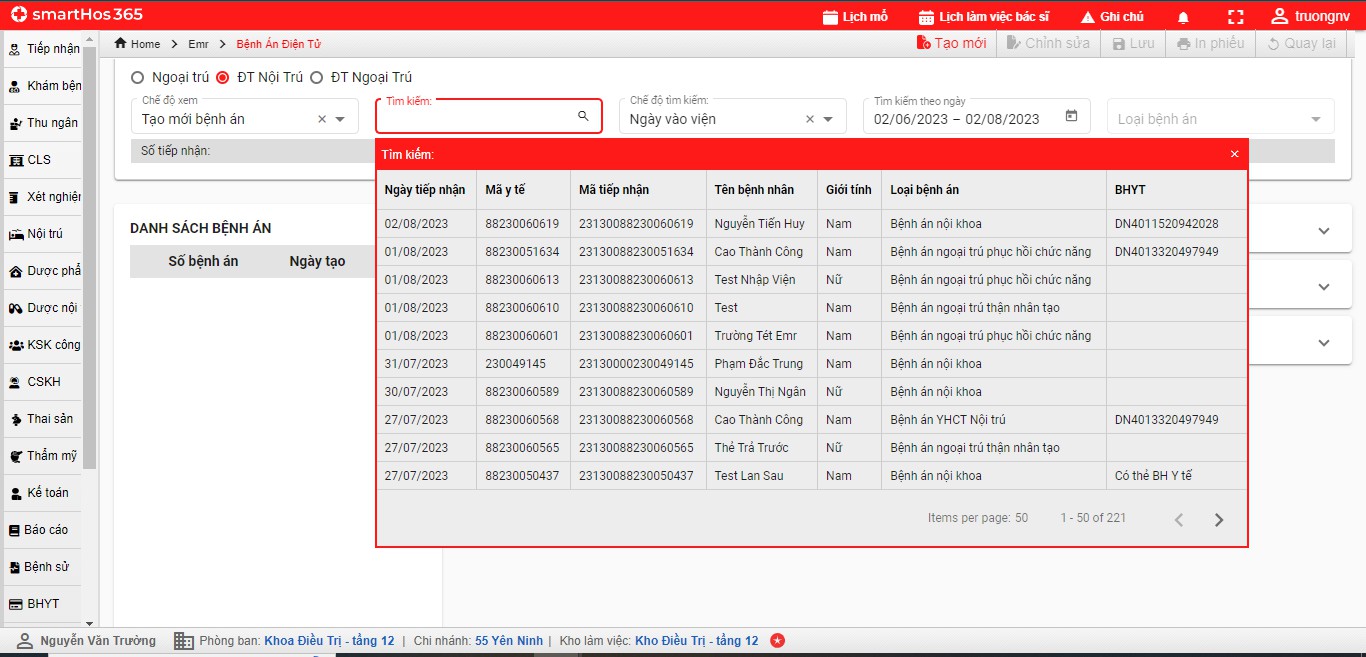


Figure 4.6: Search for patients

Figure [4.6](#_bookmark61) shows the patient search screen. Staff can search by one of the patient's information.

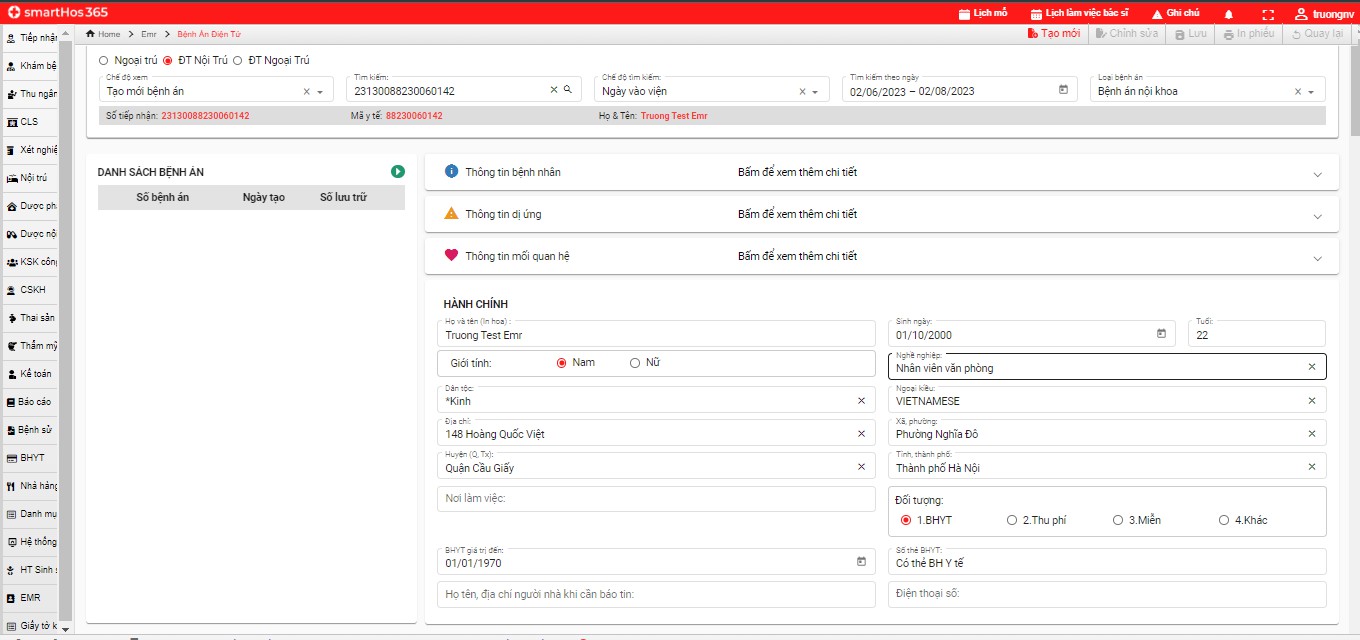


Figure 4.7: Medical record creation function

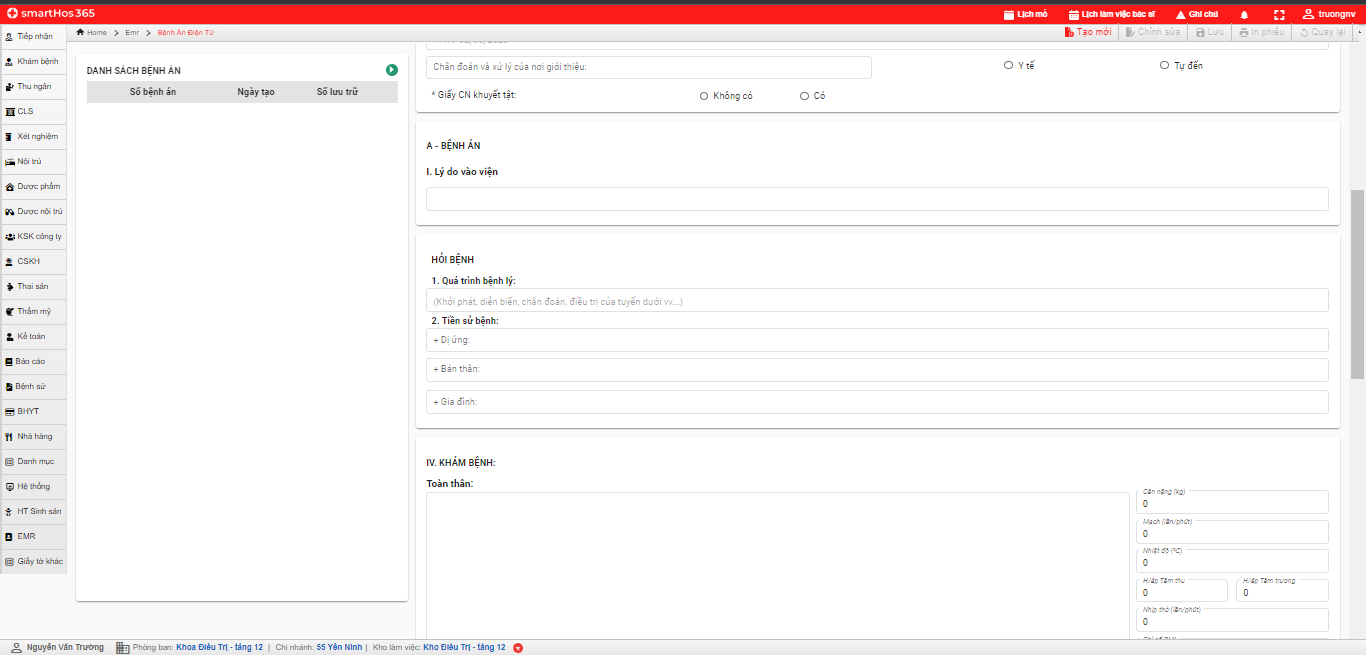


Figure 4.8: Medical record creation function

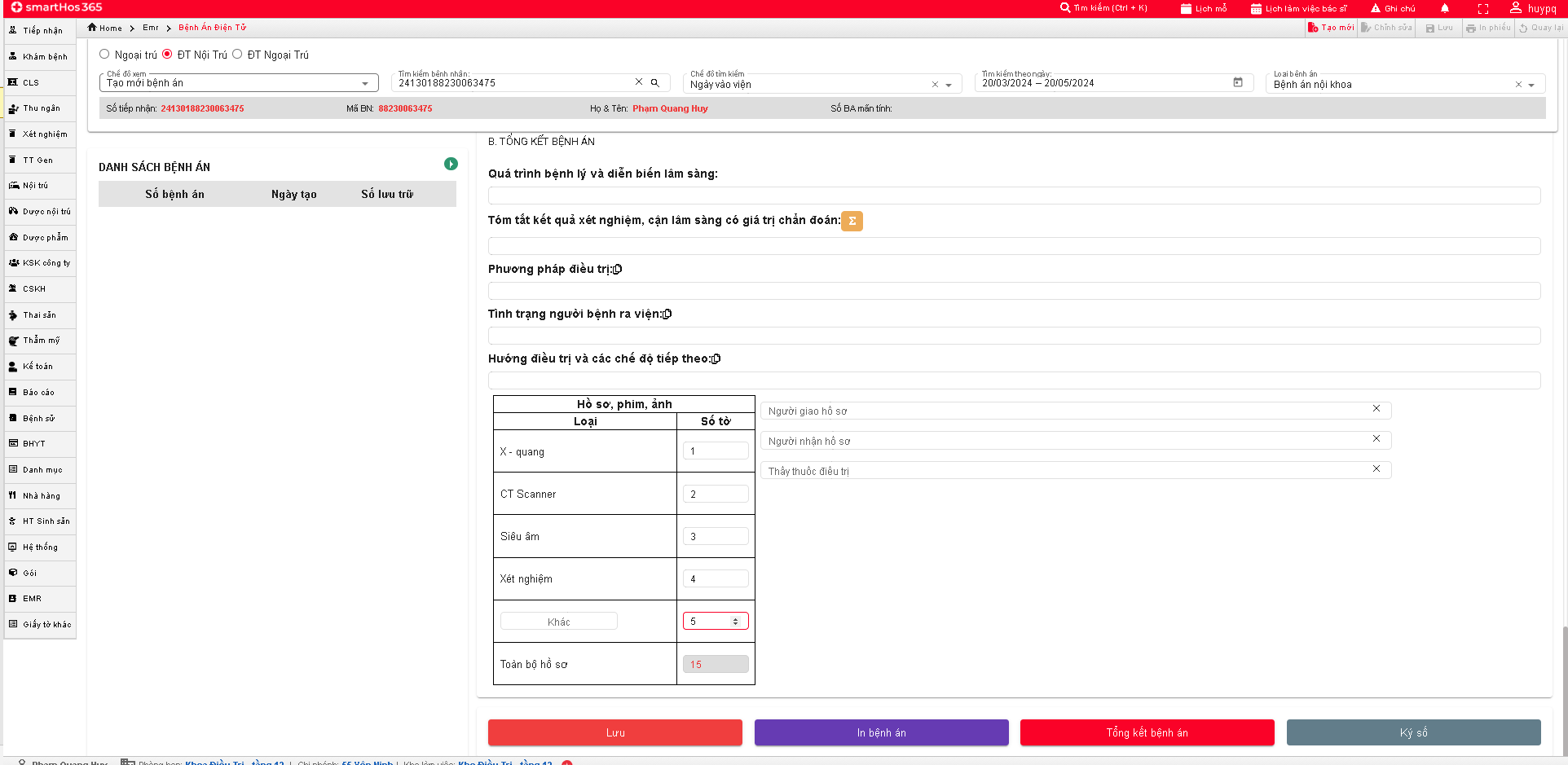
****

Figure 4.9: Medical record creation function

Figure [4.7,4.8,4.9](#_bookmark62) shows the screen creating a medical record for a patient with a medical code of 88230051634. To create a medical record, the user will initially have to select the view: Create medical record after selecting the patient in the search section. The user will then select the type of medical record. Inpatient medical records will be taken according to the type of medical record on HIS available upon admission. Next, the doctor will examine and supplement the type of medical record.

Here, the doctor will enter the patient's information such as administrative information, information during medical examination and treatment into the medical record section. At the bottom of the page is a summary of the patient's medical history. Common will include: (i) Pathological process and clinical course, (ii) Summary of subclinical laboratory results with diagnostic value, (iii) Treatment method, (iv) Discharge status, (v) Treatment direction and subsequent regimens. These information fields include all required information fields of the medical record according to the form of each medical record issued by the Ministry of Health.

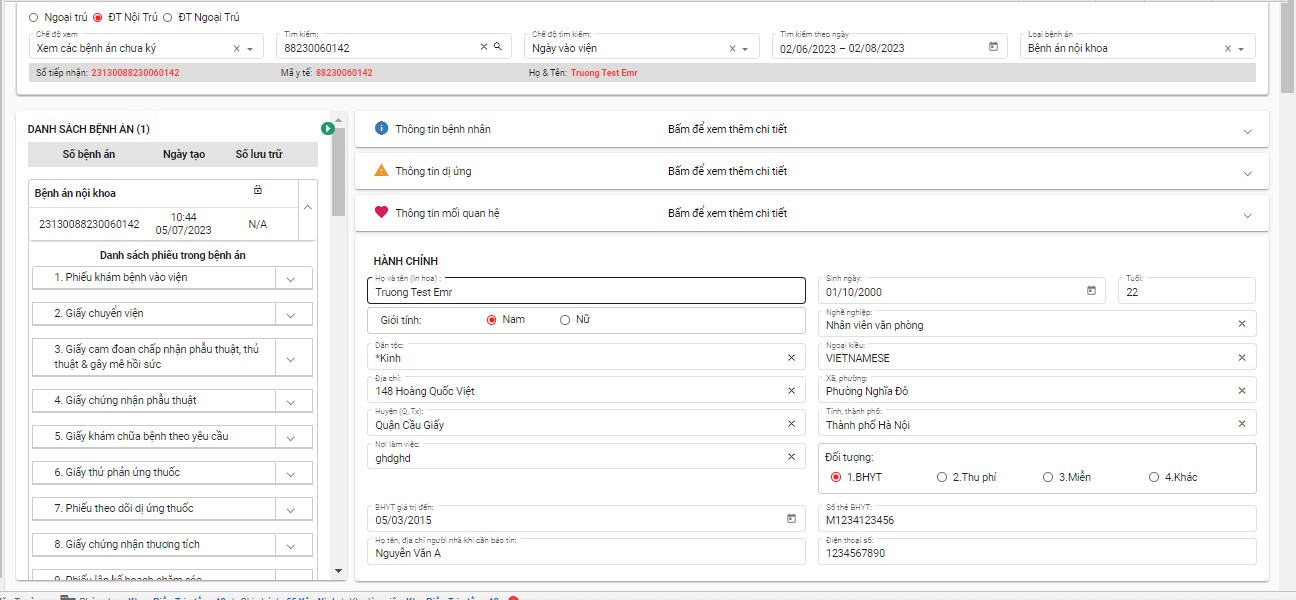


Figure 4.10: Viewing medical records

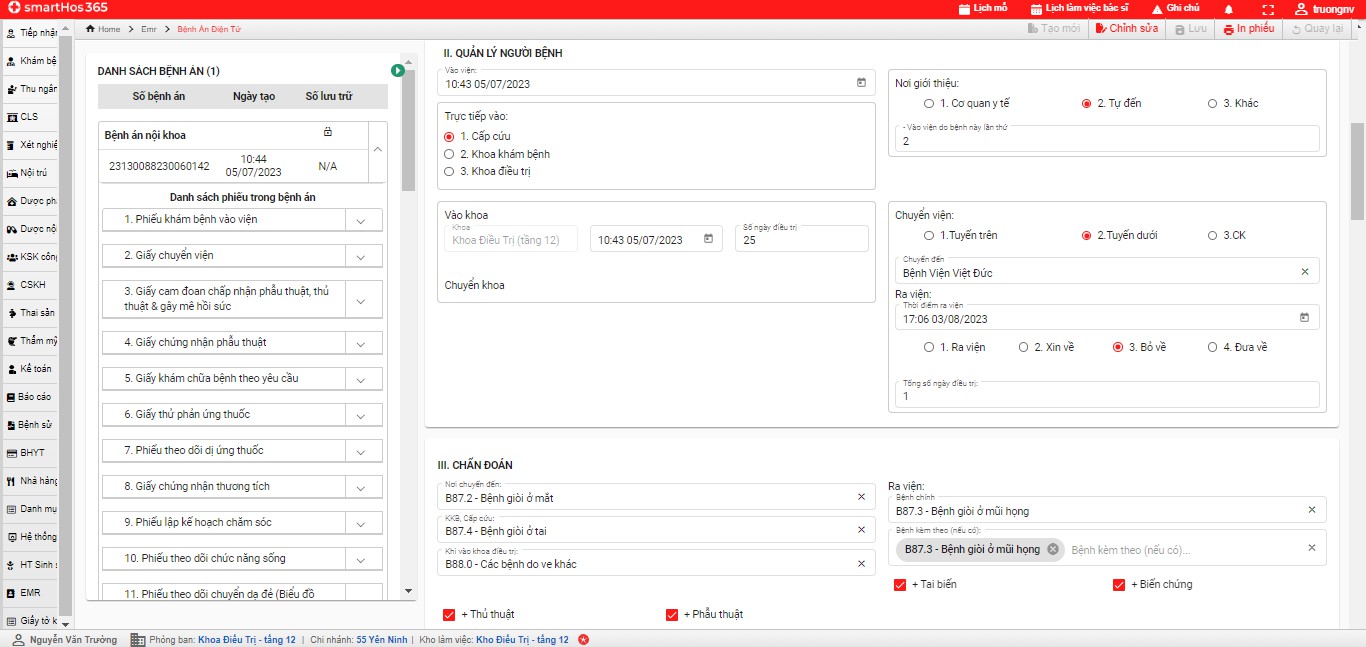


Figure 4.11: Viewing medical records

Figure [4.10,4.11](#_bookmark65) shows a screen viewing saved medical records and a list of documents related to medical records. This includes all newly created papers in EMR and papers compiled from IHS such as: ultrasound paper, X-rays,...

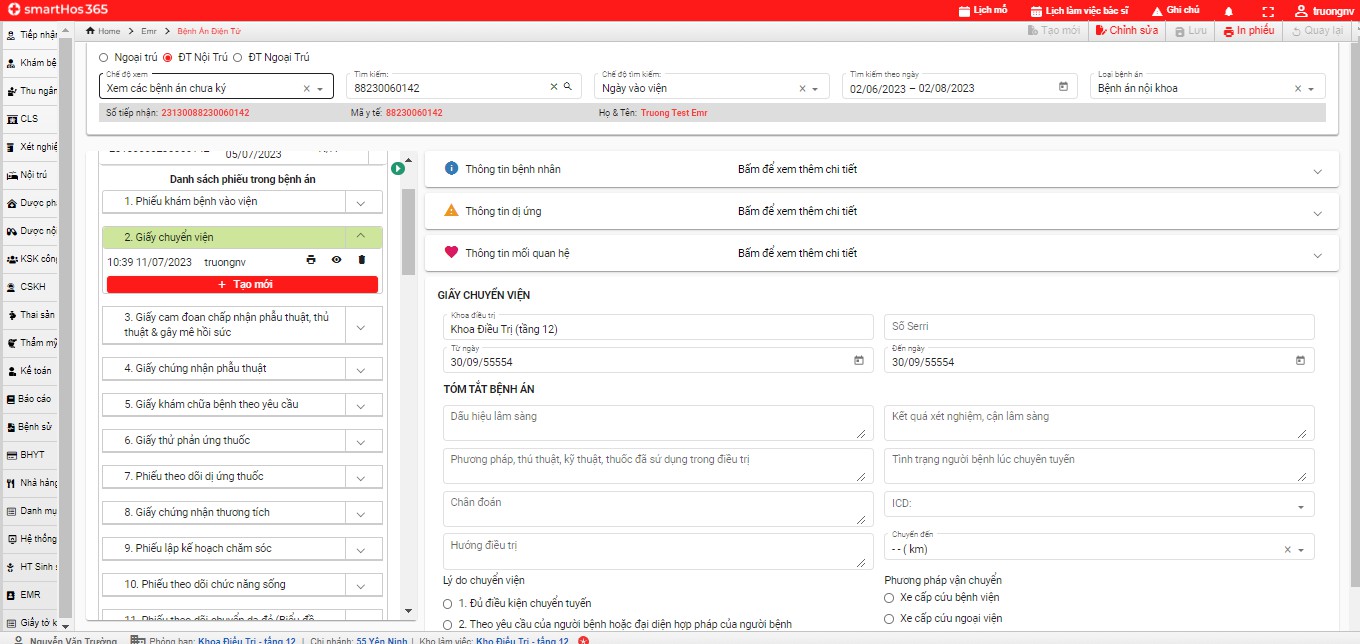


Figure 4.12: View related documents

Figure [4.12](#_bookmark67) shows the information entry screen of 1 of the relevant medical documents (which can be a new document created in the EMR alone or a document taken from HIS and newly edited in the EMR). The medical record list will include the medical record created and related documents during that medical visit. This will help users search and synthesize patient papers quickly.

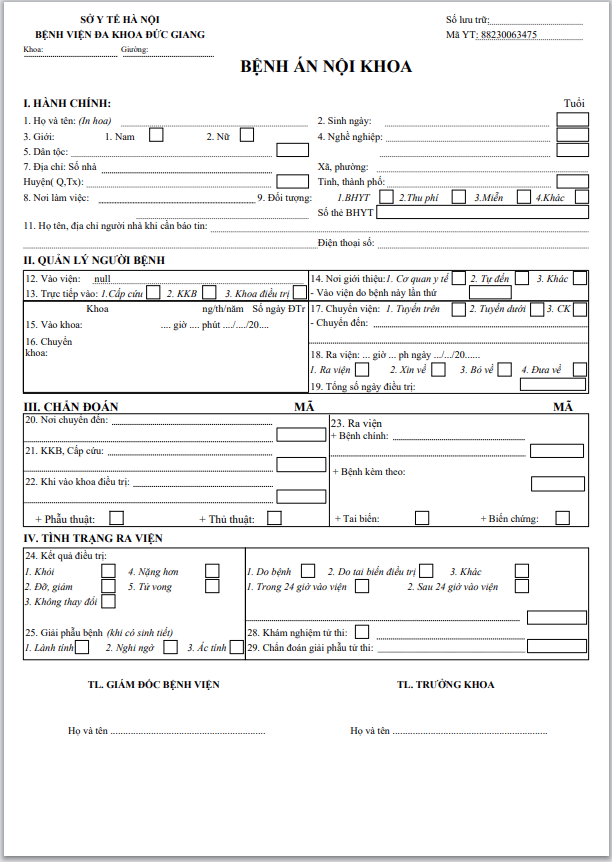
****

Figure 4.13: Printing slips, exporting PDF files

Figure [4.13](#_bookmark68) shows the printed medical report screen. Here medical staff can choose to print paper or can export to PDF files for storage.

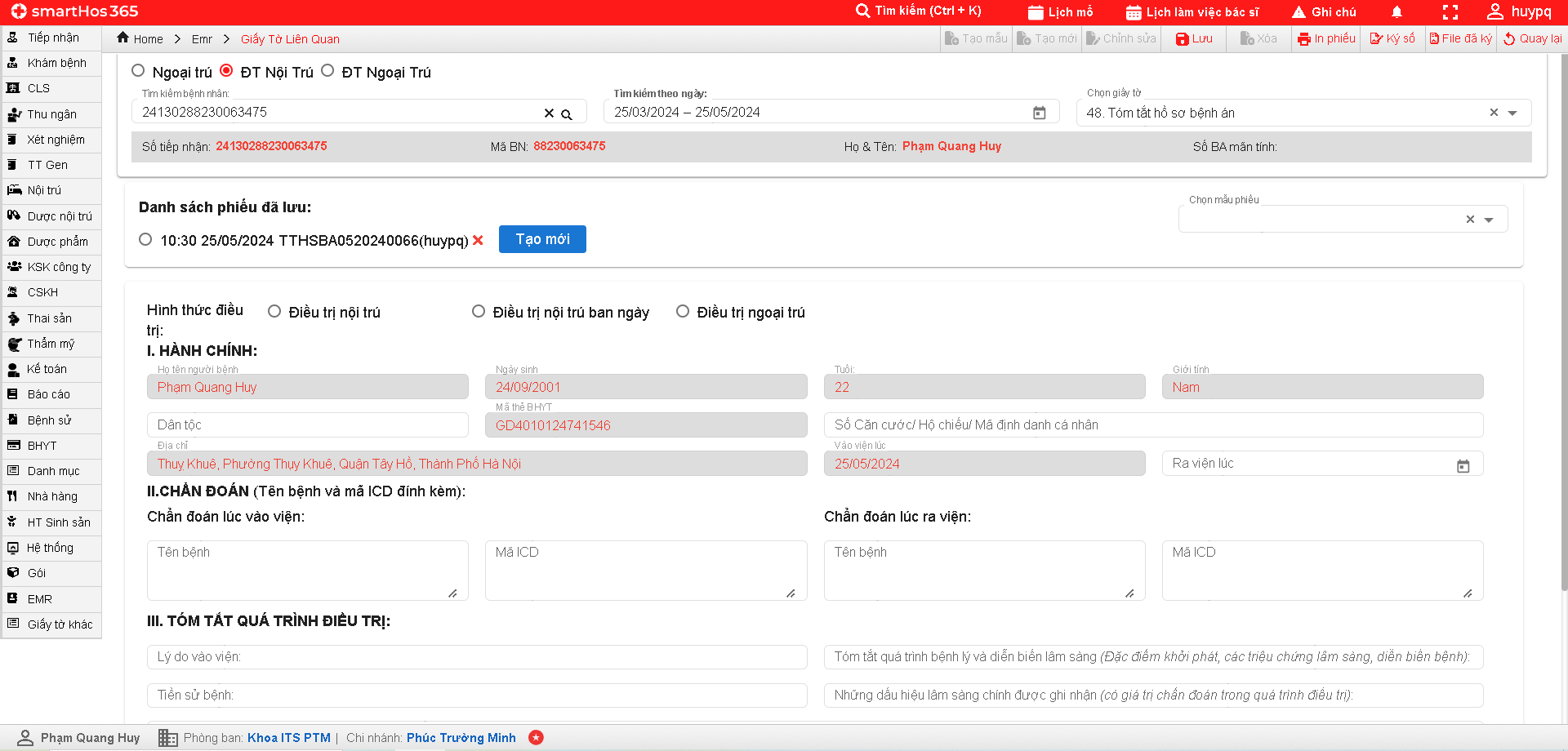


Figure 4.14: Related document screen

Figure 4.14 shows the relevant document screen (summary of medical records). The medical staff will select the patient and the type of relevant documentation he wants to do. Here the patient's administrative information will automatically pour from the reception screen. The name of the disease and the icd code when diagnosing hospitalization will be linked from the examination screen. The patient's name and diagnosis icd code at discharge will be linked from the discharge screen. After that, medical staff will enter information during the medical examination and save relevant papers

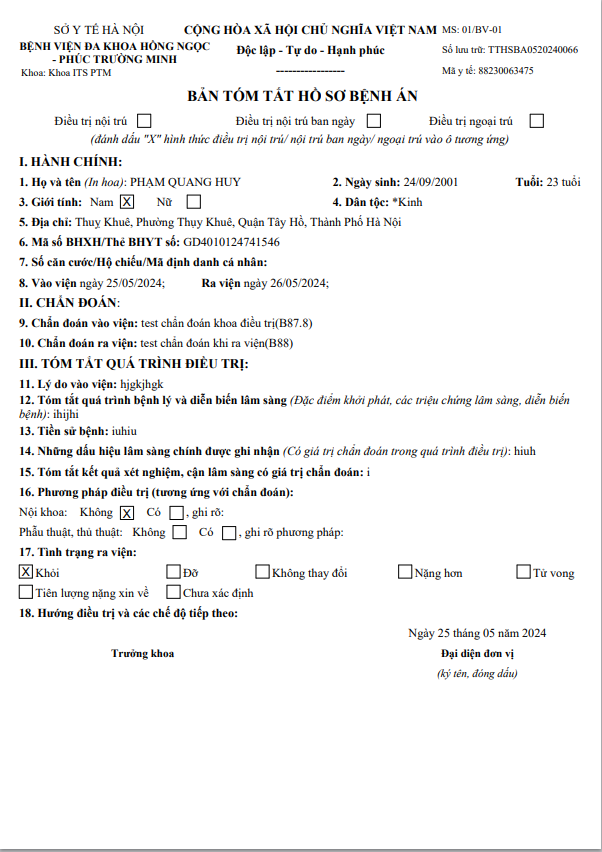


Figure 4.15: Medical summary printed slip

## **4.4 Testing**

Purpose: To ensure that the system can run in accordance with the requirements set out earlier.

Scope of testing: Testing the functions in the developed design (functions designed but not yet implemented will not be tested), the information displayed exactly as desired. The performance of the system is temporarily untested.

The system is used for the main target of doctors and nurses, so the testing will be divided into testing each function of these two objects.

Type of testing used: Unit testing - software testing in which individual units or components of software are tested.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Function | Test Process | Case | Desire | Result |
| Search for patients | 1. Click on the search bar  2.Select search mode  3.Choose a search time | Admission dates are May 1, 2024 to May 20, 2024 | Shows patients admitted between May 1, 2024 and May 20, 2024 | Pass |
| Discharge date from May 1, 2024 to May 20, 2024 | Shows patients discharged between May 1, 2024 and May 20, 2024 | Pass |
| Year of birth is 2001 | Showing patients born in 2001 | Pass |
| Phone number: 123456789 | Show patients with phone numbers 123456789 | Pass |
|  |  | Address: Hanoi | Show patients with Hanoi addresses | Pass |
| Gender: Male | Show out patients of male gender | Pass |
| Create a medical record | B1:Patient selection  S2: An "Create New"  S3:Fill in the information  S4: Click "Save" | Enter all required information | Add medical records to the database and display a successful save message | Pass |
| Enter missing required information | Display a message that required fields have not been filled in | Pass |
| Create relevant documents | B1:Patient selection  Step 2: Select the relevant document you want to create for the patient  S3: An "Create New"  Step 4: Fill in the information  Step 5: Click "Save" | Enter all required information | Add medical records to the database and display a successful save message | Pass |
| Enter missing required information | Display a message that required fields have not been filled in | Pass |
| Print electronic medical records | B1:Patient selection  S2:Select the generated electrical medical record  Step 3: Press the slip |  | Display printed slips and design according to the health ministry form | Pass |
| Print relevant documents | B1:Patient selection  S2:Select the generated electrical medical record  Step 3: Press the slip |  | Display printed slips and design according to the health ministry form | Pass |
| Patient information link | S1: Patient selection | Administrative information of the patient | Link to the patient's administrative information from the reception screen | Pass |
| Patient examination information | Link to the patient's medical examination from the examination screen | Pass |
| Create a template of information fields | S1: Select the disease  S2: Create new | Create a new template | Can save and use templates | Pass |
| Edit an existing template | Successful correction and template update after correction | Pass |
| Delete an existing template | Remove the created template from the list | Pass |

Table 4.2: Functional testing

**4.5 Deployment**

In this project, I was tasked with developing an interface and API integration for EMR (Electronic Medical Record) displays. The goal of the project is to improve the user experience and increase the interoperability of the EMR system, making it easier for doctors and medical staff to access and manage medical information.

-Interface development

* Build a user-friendly interface that is compatible with many different devices and screen sizes.
* Use modern technologies like HTML5, CSS3, and JavaScript to create a beautiful and easy-to-use interface.
* Optimize user experience by considering aesthetics and performance.

-Mount api

* Analyze API requests from the EMR system and identify endpoints to integrate.
* Use software development libraries and tools to embed APIs into the user interface.
* Check and handle error cases when calling API to ensure system stability.

-Make printed slips

* Conduct requirements analysis from relevant departments to understand the information that needs to be displayed on the printed form.
* Determine the format, size, and model requirements for printed slips.
* Use jasper studio software to create an aesthetically pleasing and easy-to-read printed slip template.
* Optimize the layout and consider colors and fonts to make printed slips professional and easy to use.

**4.5.1 End of deployment**

The module is deployed in the same network as HIS, illustrated in Figure [4.14.](#_bookmark73) The following only briefly explains HIS, as their details are beyond the scope of this project

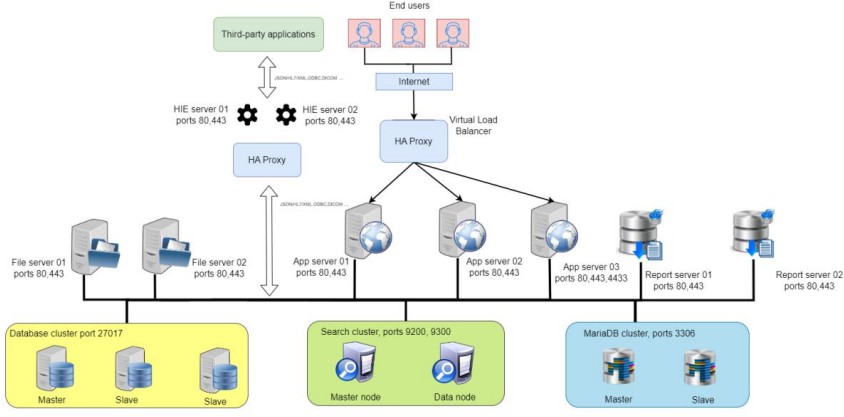


Figure 4.14: Deployment architecture

Requests from users are fed into a load balancer using a High Availability Proxy (HAProxy), which is then routed to three servers:

* *Application server 01*: This server is responsible for hosting the aggregated user interface of applications in HIS, written in PHP, AngularJS, or Angular to make requests to the two back-end servers at the bottom.
* *Application server 02*: This server handles requests made by server 01 using PHP or Golang. The backend of HIS applications is stored here.
* *Application server 03*: This server is similar to server 02, and also handles requests routed from server 01. It has additional subdomains to cater to the company's experimental applications, including our Medical Records management module.

Back-end servers 02 and 03 typically need to store, manage, and query data continuously. This is done in IHS by routing database requests through a common bus to two database groups and query requests to a search group. Two databases and a search service each have a primary server and at least one secondary server, which replicates actions performed on the main server.

Commonly used data fields from a MariaDB cluster group and a MongoDB cluster group are indexed on a search server group to speed up queries, with the help of an Elasticsearch service. Search requests made by the application are routed to these servers to quickly retrieve the necessary data records, rather than querying directly into the database.

This database architecture also allows the company to manage an efficient reporting system. IHS has an integrated reporting system powered by JasperReports, an open-source reporting tool that operates on the company's two servers. These servers can be accessed to build, store, and print report templates visually with clear indications of where data is needed and from which data tables or database collections. When the application requests a report for a database record, this report template is populated with the required data of the record and finally sent back to the application as a printable document.

**4.5.2 Summary of IHS specifications**

By using this architecture, HIS can simultaneously process about 1000 online users and is capable of handling 3000 to 5000 requests per day by healthcare facilities using HIS. Data is available in about 100 to 300 milliseconds. Server-hosted applications can deliver user interfaces at fast speeds, within no more than 100 milliseconds.

Our modules will be deployed into this system, ensuring availability and speed when participating in actual customer use. Since the scale of a medical record is much smaller than HIS, HIS will be a sufficient solution for the implementation of this module.

Above, I presented architectural design, detailed design, system construction and implementation. Next, in chapter 5, I present my contributions as well as conclusions and further development directions in my project.

**CHAPTER 5. CONCLUSION AND DIRECTION OF DEVELOPMENT**

Above, I have presented everything related to my project. This is the last chapter where I will summarize the results achieved during the process of making the graduation project, and also review the knowledge I have learned and the valuable experiences I have learned and future development directions, which are limited in time I have not been able to implement.

**5.1 Conclusion**

In summary, after finishing the graduation project, I have completed the work of building an electronic medical record, meeting the basic functions required by the Ministry of Health. A module can manage medical records and related documents, instead of separating them like the software most hospitals have used before. Through the process of implementing the project, I have drawn many important conclusions and learned many useful things. From analyzing each operational process in detail to researching new technologies, I had the opportunity to deepen my understanding of basic Angular, MongoDB, Golang and successfully apply it in the project. This will be an important and extremely valuable baggage and knowledge for me to be confident and steady to help me step out of school and develop more for my career in the future.

The remarkable result that I achieved was to build a module that can manage patients' medical records and related documents during medical examination and treatment at the hospital. She has established basic functions related to the management of medical records and related documents, helping doctors easily find patients' medical records, reducing examination time, supporting timely treatment, helping to improve the quality of diagnosis and treatment. Contributing to the digitization of patient documents, aiming towards the directive to develop electronic medical records in all hospitals of the Ministry of Health: Hospitals nationwide must start developing a roadmap to convert from paper medical records to electronic medical records, By 2030 at the latest, hospitals across the country must complete this transition.

After this module of mine is implemented, it will partially solve the problem and initial requirements, greatly helping the medical examination and treatment process at medical examination and treatment facilities. Previously, when patients were discharged from the hospital, nurses had to take paper medical records to the General Planning Department, hospital leaders ... To sign and close, now it is only necessary to transfer the patient's electronic medical record on a computer or tablet through the electronic medical record software system to related departments, directly leading the patient, the entire patient's medical record is reviewed and modified on the system. Therefore, nurses do not have to go back and forth and wait. Or the day before, if there was an abnormal case to seek medical evidence, the staff of the General Planning Department had to climb to the paper medical record archive on the 7th floor, had to search through thousands of records to find the right medical records they needed, very tired, time-consuming, Now, with electronic medical records, just enter the name and the patient's medical record will appear. As for doctors, in the past, each visit to the nurse had to push the whole car with about 50 paper medical records of the patient to follow, but now the doctor only holds the iPad, to the patient who enters that patient code and manipulates on the machine, so it is always very convenient, All information about disease progress, treatments, subclinical results such as ultrasound, tests, filming ... are saved on electronic medical records, doctors can review them at any time needed, without having to call a nurse to search like saving in paper medical records. Another convenient thing that our electronic medical record module brings is that: save all treatment information of the patient during that hospital stay, if the next time the patient is hospitalized, the doctor just needs to enter the medical code and the information appears, which helps a lot for the doctor in diagnosis and treatment. As for patients, when going to the hospital for medical examination and treatment, they do not have to bring medical examination books, test papers, films, or prescriptions ... previously treated. But through electronic medical records, the doctor will update the patient's medical history, from which there will be an appropriate treatment plan. Our electronic medical record module is connected to software such as Hospital Information System (HIS), Picture Archiving and Communication System (PACS), Laboratory Information System (LIS), etc Therefore, all medical examination and treatment results, test results, films, ultrasound are pushed to electronic medical record software. Doctors will view the results on electronic medical records, without having to print them, while saving money on printing, while nurses do not have to run to the Department of Diagnostic Imaging - Imaging tests or ultrasound results, tests as before. Due to limited time, up to now, we have not been able to complete due to inadequacies related to electronic digital signing. In the future, we will develop modules on tablets to facilitate the work of doctors, nurses ...

**5.2 Development direction**

As mentioned above, currently this electronic medical record module of mine still has a major limitation: it cannot be self-summarized, cannot be digitally signed, deployed on tablets to facilitate the work of doctors and nurses. So, in the near future I want to fix this problem.

* Create a medical summary button, when clicking on the system will summarize the medical record and will display the medical examination record and all documents related to the medical record without having to click on each item in the list of documents to check as before
* Complete the digital signature function to meet all requirements related to electronic medical records set out by the Ministry of Health in Circular No. 46/2018/TT-BYT, promulgating regulations on electronic medical records.
* Complete modules on tablets, to facilitate the work of doctors and nurses, can work in many places, no longer depend on computers.

I hope to continue to follow this project of the company to solve the problem of shortcomings and weaknesses, as well as improve my skills and methodology.

**BIBLIOGRAPHY**

1. Circular No. 46/2018/TT-BYT,Circular *No. 46/2018/TT-BYT*, 28/12/2018,http [s://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-and-management-ho-so-sickness-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. Article 59 of Law No. 40/2009/QH12,Law *on Medical examination and treatment, Law No. 40/2009/QH12*, 31/11/2009,https://thuvienphapluat.vn/van-ban/The-thao-Y

[-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. Circular No. 54/2017/TT-BYT,Circular *No. 54/2017/TT-BYT*, 29/12/2017,ht [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-mini-chi-application-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 Phuong, *"THE JOURNEY TO DEPLOY ELECTRONIC MEDICAL RECORDS IN VIETNAM"*, 03/2021,https://giaiphapviettel.vn/tin-tuc/446/han [h-trinh-deploy-disease-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)