|  |  |
| --- | --- |
|  | BỘ GIÁO DỤC VÀ ĐÀO TẠO  **TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM** |

**BÁO CÁO ĐỒ ÁN**

**HỌC PHẦN: ĐỒ ÁN CƠ SỞ**

**ĐỀ TÀI: HỆ THỐNG CẦU NỐI HIẾN MÁU**

**Khoa:** **CÔNG NGHỆ THÔNG TIN**

**Giảng viên hướng dẫn:** Th.S Võ Hoàng Khang

**Sinh viên thực hiện:** Trần Cao Quỳnh Như (2280602255)

Trần Hoàng Huy (2280601249)

**Lớp:** 22DTHG5, 22DTHE2

TP. Hồ Chí Minh, 2025

**LỜI CAM ĐOAN**

Chúng tôi, nhóm sinh viên Trần Cao Quỳnh Như, Trần Hoàng Huy xin cam đoan rằng:

Đồ án "Hệ thống Cầu Nối Hiến Máu" là thành quả của quá trình nghiên cứu và làm việc nghiêm túc của cả nhóm. Trong suốt quá trình thực hiện, chúng tôi đã tham khảo các tài liệu học thuật, các bài nghiên cứu và nguồn thông tin liên quan để có thể hoàn thiện đồ án. Bên cạnh đó, chúng tôi luôn đảm bảo trích dẫn đầy đủ, chính xác các nguồn tài liệu đã sử dụng và tuân thủ nghiêm ngặt các quy định về đạo đức học thuật.

Chúng tôi cam kết chịu trách nhiệm hoàn toàn về tính trung thực và chính xác của những nội dung đã trình bày cũng như đảm bảo rằng đồ án này tuân thủ đầy đủ các quy định về đạo đức học thuật.

Chúng tôi hy vọng rằng đồ án "Hệ thống Cầu Nối Hiến Máu" sẽ mang lại giá trị thực tiễn trong việc ứng dụng phân tích và thiết kế hệ thống cũng như là nền tảng cho những nghiên cứu, phát triển sau này.

TP.HCM, ngày 31 tháng 3 năm 2025

**Sinh viên**

Trần Cao Quỳnh Như

Trần Hoàng Huy

**LỜI CẢM ƠN**

Nhóm chúng tôi xin gửi lời cảm ơn chân thành đến giảng viên Th.S Võ Hoàng Khang người đã tận tình hướng dẫn và hỗ trợ chúng tôi trong suốt quá trình thực hiện đồ án. Sự chỉ dẫn quý báu và sự quan tâm của cô đã giúp chúng tôi vượt qua nhiều khó khăn và hoàn thành đồ án này.

Chúng tôi cũng xin cảm ơn các thầy, cô trong khoa Công nghệ Thông tin trường Đại học Công nghệ Thành phố Hồ Chí Minh (HUTECH) đã cung cấp cho chúng tôi nền tảng kiến thức vững chắc trong suốt quá trình học tập. Những kiến thức và kỹ năng thầy cô truyền đạt là nền móng quan trọng để chúng tôi thực hiện thành công đồ án này.

Cuối cùng, chúng tôi xin cảm ơn gia đình và bạn bè đã luôn bên cạnh động viên, ủng hộ tinh thần để chúng tôi hoàn thành tốt đồ án.

**MỤC LỤC**

[DANH SÁCH HÌNH 5](#_Toc194427859)

[CHƯƠNG I: TỔNG QUAN 6](#_Toc194427860)

[1.1 Giới thiệu đề tài: 6](#_Toc194427861)

[1.2 Tổ chức và các chức năng của hệ thống 6](#_Toc194427862)

[1.3 Mô tả yêu cầu với các nghiệp vụ chính 6](#_Toc194427863)

[*1.3.1 Quy trình tiếp nhận và xử lý máu hiến từ bệnh viện* 6](#_Toc194427864)

[*1.3.2* *Quy trình hiến máu* 6](#_Toc194427865)

[*1.3.3* *Quy trình quản trị hệ thống* 7](#_Toc194427866)

[*1.3.4 Quy trình đặt máu của bệnh viện* 7](#_Toc194427867)

[*1.3.5 Quy trình hiến máu có trả tiền* 7](#_Toc194427868)

[CHƯƠNG 2: SƠ ĐỒ USE CASE 9](#_Toc194427869)

[2.1 Use Case tổng quát 9](#_Toc194427870)

[2.2 Use Case 9](#_Toc194427871)

[2.3 Use Case phục vụ khách hàng 10](#_Toc194427872)

[2.4 Use Case quy trình thanh toán 12](#_Toc194427873)

[2.5 Use Case quản lý 14](#_Toc194427874)

[CHƯƠNG 3: SƠ ĐỒ LỚP 16](#_Toc194427875)

[PHIẾU PHÂN CÔNG NHIỆM VỤ 17](#_Toc194427876)

DANH MỤC HÌNH ẢNH

*Hình 1 : Mô hình Use Case tổng quát*

*Hình 1.1 : Mô hình Use Case quản lí “ Quy trình hiến máu”*

*Hình 1.2 : Mô hình Use Case quản lí “Quy trình quản trị hệ thống”*

*Hình 1.3 : Mô hình Use Case quản lí “Quy trình đặt máu của bệnh viện”*

*Hình 1.4 : Mô hình Use Case quản lí “Quy trình hiến máu có trả tiền”*

*Hình 2 : Mô hình class hệ thống “ Cầu nối hiến máu”*

CHƯƠNG I: TỔNG QUAN

**1.1 KHẢO SÁT THỰC TRẠNG** Máu là một thành phần vô cùng quan trọng trong cơ thể con người, đóng vai trò thiết yếu trong duy trì sự sống. Tuy nhiên, trong thực tế hiện nay, tình trạng khan hiếm máu đang diễn ra tại nhiều cơ sở y tế trên cả nước, đặc biệt là trong các tình huống khẩn cấp như tai nạn giao thông, cấp cứu đột xuất, phẫu thuật lớn, sinh nở biến chứng hoặc điều trị các bệnh lý cần truyền máu định kỳ (thalassemia, ung thư...). Hầu hết việc kêu gọi hiến máu hiện nay vẫn còn phụ thuộc nhiều vào các chiến dịch truyền thông ngắn hạn hoặc hoạt động tự nguyện theo đợt, chưa có hệ thống nào giúp kết nối người hiến máu với các bệnh viện một cách chủ động và kịp thời.

Ngoài ra, một số người sẵn sàng hiến máu khi cần thiết nhưng không biết nơi nào đang cần và khi nào cần. Đồng thời, phía bệnh viện gặp khó khăn trong việc chủ động liên hệ người hiến máu theo nhóm máu và vị trí địa lý gần nhất, đặc biệt trong trường hợp cấp cứu. Do đó, việc xây dựng một hệ thống công nghệ thông tin giúp hỗ trợ tìm kiếm người hiến máu theo vị trí, nhóm máu và khả năng đáp ứng nhanh chóng là rất cần thiết và cấp bách.

**1.2 TÍNH KHẢ THI CỦA BÀI TOÁN** Hiện nay, với sự phát triển mạnh mẽ của công nghệ web, hệ thống định vị GPS, API bản đồ (Google Maps API, Leaflet, OpenStreetMap...), việc xây dựng một hệ thống web kết nối giữa người hiến máu và bệnh viện là hoàn toàn khả thi.

Việc tích hợp bản đồ và chức năng định vị cho phép bệnh viện tìm kiếm người hiến máu đang ở gần trong bán kính nhất định (ví dụ: 5km, 10km, 20km...) theo nhóm máu cụ thể. Mặt khác, hệ thống có thể mở rộng để cập nhật tình trạng sẵn sàng hiến máu, thông báo qua email/SMS và đồng bộ dữ liệu với hệ thống y tế trung ương.

Công nghệ phù hợp với bài toán bao gồm: React (xây dựng giao diện), Node.js (API backend), MongoDB hoặc MySQL (CSDL), Google Maps API (bản đồ và định vị), giúp hệ thống có thể mở rộng và triển khai thực tế trong thời gian ngắn.

**1.3 ĐỐI TƯỢNG NGHIÊN CỨU SỬ DỤNG**

* Nhân viên bệnh viện: Có quyền đăng nhập vào hệ thống, sử dụng chức năng tìm kiếm người hiến máu trong bán kính gần bệnh viện, theo nhóm máu yêu cầu. Ngoài ra còn có thể quản lý danh sách người hiến và xem hồ sơ thông tin liên quan.
* Người hiến máu: Có thể đăng ký tài khoản, cung cấp thông tin cá nhân, nhóm máu, vị trí cư trú và trạng thái sẵn sàng hiến máu. Ngoài ra, người dùng có thể nhận thông báo khi có bệnh viện cần máu khẩn cấp.

**1.4 MỤC TIÊU NGHIÊN CỨU**

* Xây dựng hệ thống thông tin hỗ trợ kết nối người hiến máu và bệnh viện một cách chủ động.
* Cho phép bệnh viện tìm kiếm người hiến máu trong bán kính mong muốn theo nhóm máu, nhanh chóng và chính xác.
* Cung cấp bản đồ trực quan định vị các người hiến máu trong khu vực.
* Tăng tốc độ xử lý cấp cứu và giảm tỷ lệ tử vong do thiếu máu truyền.
* Tạo giao diện đơn giản, trực quan và dễ sử dụng cho cả người hiến và cán bộ y tế.

**1.5 PHẠM VI GIỚI HẠN**

* Đề tài chỉ triển khai dưới dạng hệ thống web (chạy trên trình duyệt).
* Dữ liệu hiện tại là dữ liệu mô phỏng, chưa kết nối với hệ thống dữ liệu thực của bệnh viện.
* Chưa tích hợp hệ thống phân quyền đầy đủ giữa quản trị viên, bệnh viện và người dùng cá nhân.
* Không triển khai phiên bản mobile (ứng dụng di động).
* Không xử lý đồng bộ thời gian thực (push notification/sockets).

**1.6 Ý NGHĨA KHOA HỌC VÀ THỰC TIỄN CỦA ĐỀ TÀI**

* Góp phần nâng cao nhận thức về ứng dụng CNTT trong ngành y tế và hỗ trợ công tác truyền máu.
* Tạo nền tảng công nghệ ban đầu để triển khai các hệ thống lớn hơn trên phạm vi quốc gia.
* Hỗ trợ kịp thời bệnh viện trong công tác cứu chữa bệnh nhân cần máu.
* Tạo môi trường kết nối hiệu quả giữa người hiến máu và cơ sở y tế.

**1.7 CẤU TRÚC ĐỒ ÁN** Đồ án tập trung vào các quá trình nghiên cứu và phát triển chính như sau:

* Về cơ sở dữ liệu: Thiết kế mô hình dữ liệu phản ánh chính xác thực tế người dùng và bệnh viện, tối ưu hóa lưu trữ và truy vấn thông tin. Tiến hành khảo sát, phân tích các thực thể như: người hiến máu, nhóm máu, hồ sơ bệnh viện, vị trí địa lý… để xây dựng hệ cơ sở dữ liệu phù hợp.
* Về phát triển API backend: Xây dựng các API RESTful bằng Node.js nhằm cung cấp dịch vụ linh hoạt giữa frontend và cơ sở dữ liệu. API hỗ trợ xác thực, phân quyền, truy vấn danh sách người hiến máu theo vị trí và nhóm máu, tích hợp bản đồ và hỗ trợ mở rộng các tính năng trong tương lai.
* Về hệ thống:
  + BLOOD-BRIDGE ADMIN: Giao diện dành cho nhân viên bệnh viện sử dụng trên nền tảng web. Cho phép đăng nhập, tìm kiếm người hiến máu trong bán kính theo địa chỉ bệnh viện, thống kê kết quả và quản lý hồ sơ người hiến máu.
  + BLOOD-BRIDGE DONOR: Giao diện người dùng dành cho người hiến máu. Cho phép đăng ký hồ sơ, cung cấp thông tin cá nhân, nhóm máu, vị trí hiến máu. Có thể sẵn sàng cập nhật trạng thái và nhận thông báo từ hệ thống khi có nhu cầu.
  + BLOOD-BRIDGE MAP SERVICE: Mô-đun tích hợp Google Maps API, hỗ trợ định vị và tính toán khoảng cách giữa người hiến máu và bệnh viện. Cho phép lọc dữ liệu theo bán kính và hiển thị marker trực quan trên bản đồ.

Cấu trúc tổng thể gồm các chương:

* CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI
* CHƯƠNG 2: CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ SỬ DỤNG
* CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG
* CHƯƠNG 4: CÀI ĐẶT VÀ TRIỂN KHAI HỆ THỐNG
* CHƯƠNG 5: KẾT QUẢ THỰC NGHIỆM, ĐÁNH GIÁ HỆ THỐNG VÀ HƯỚNG PHÁT TRIỂN

CHƯƠNG II: CƠ SỞ LÝ THUYẾT

**2.1 CƠ SỞ LÝ THUYẾT**

**2.1.1 Hệ thống thông tin và vai trò trong lĩnh vực y tế** Hệ thống thông tin là tập hợp các thành phần như phần mềm, phần cứng, cơ sở dữ liệu và con người được tổ chức nhằm phục vụ thu thập, xử lý, lưu trữ và phân phối thông tin cho các mục tiêu cụ thể. Trong lĩnh vực y tế, hệ thống thông tin giúp hỗ trợ bác sĩ, nhân viên y tế và bệnh viện quản lý hiệu quả quy trình điều trị, chẩn đoán, lưu trữ hồ sơ bệnh án và đặc biệt trong việc kết nối người hiến máu và nơi cần máu.

**2.1.2 Tìm kiếm theo bán kính và thuật toán Haversine** Việc tìm kiếm người hiến máu trong bán kính dựa vào vị trí bệnh viện sử dụng thuật toán Haversine. Đây là công thức tính khoảng cách giữa hai điểm trên bề mặt quả địa cầu dựa trên tọa độ địa lý (latitude, longitude). Công thức này giúp xác định người ở gần bệnh viện nhất trong một khoảng cách nhất định.

**2.1.3 Mô hình MVC (Model - View - Controller)** Mô hình MVC được áp dụng để tách riêng phần giao diện người dùng (View), xử lý logic (Controller) và truy cập dữ liệu (Model). Cấu trúc này giúp dễ bảo trì, mở rộng và phát triển ứng dụng lớn một cách rõ ràng và linh hoạt.

**2.1.4 Khái niệm RESTful API RESTful API** là dạng giao tiếp giữa frontend và backend thông qua giao thức HTTP. Mỗi API đại diện cho một tài nguyên cụ thể như người hiến máu, hồ sơ bệnh viện, kết quả tìm kiếm,... Việc sử dụng REST API giúp hệ thống mở rộng và tích hợp dễ dàng.

**2.2 CÔNG NGHỆ VÀ CÔNG CỤ SỬ DỤNG**

**2.2.1 Frontend** - ReactJS ReactJS là thư viện JavaScript mã nguồn mở do Facebook phát triển, chuyên dùng để xây dựng giao diện web hiện đại. Giao diện chia thành các component giúp dễ bảo trì và tái sử dụng. Trong đồ án, React được dùng kết hợp với React Router để quản lý điều hướng và Material UI để xây dựng giao diện đẹp mắt.

**2.2.2 Backend** - Node.js và Express Node.js là môi trường chạy JavaScript phía server, còn Express là framework giúp xây dựng API nhanh chóng và dễ mở rộng. Express hỗ trợ middleware, định tuyến rõ ràng, tích hợp dễ dàng với CSDL. Backend chịu trách nhiệm xử lý dữ liệu người dùng, phân quyền, xác thực và truy vấn dữ liệu phục vụ cho bệnh viện.

**2.2.3 Cơ sở dữ liệu - MongoDB và MySQL**

* MongoDB: cơ sở dữ liệu NoSQL dạng tài liệu, phù hợp với hệ thống có dữ liệu linh hoạt, cho phép lưu dữ liệu JSON. Phù hợp với phần lưu thông tin người hiến máu, địa điểm, vị trí.
* MySQL: hệ quản trị cơ sở dữ liệu quan hệ, dễ kiểm soát dữ liệu có cấu trúc chặt chẽ. Dễ thiết kế bảng, dùng tốt cho thông tin bệnh viện, tài khoản và phân quyền.

**2.2.4 Google Maps Platform Google Maps API là bộ dịch vụ bản đồ nổi tiếng hỗ trợ định vị, tìm kiếm, đo khoảng cách. Đồ án sử dụng Geocoding API để chuyển địa chỉ bệnh viện sang tọa độ và** sử dụng Maps JavaScript API để hiển thị bản đồ, các marker và popup người hiến máu gần khu vực.

**2.2.5 Các thư viện frontend hỗ trợ**

* **Axios**: Thư viện HTTP client giúp frontend gửi yêu cầu đến backend dễ dàng, hỗ trợ các tính năng như interceptor, quản lý token.
* **React Router**: Quản lý điều hướng SPA (Single Page Application), giúp chuyển trang không cần reload lại.
* **Material UI**: Bộ component UI đẹp mắt theo chuẩn Material Design, dễ dùng, dễ tùy biến.

**2.2.6 Các thư viện backend hỗ trợ**

* **cors**: Hỗ trợ chia sẻ tài nguyên giữa frontend (port 5173) và backend (port 5000).
* **dotenv**: Quản lý biến môi trường như API Key, chuỗi kết nối CSDL.
* **express-validator**: Hỗ trợ xác thực dữ liệu đầu vào ở server.
* **nodemon**: Tự động restart server khi thay đổi mã nguồn trong quá trình phát triển.

**2.2.7 Công cụ và phần mềm hỗ trợ**

* **Git & GitHub**: Quản lý mã nguồn, theo dõi lịch sử phát triển, phân nhánh và hỗ trợ làm việc nhóm hiệu quả.
* **Visual Studio Code**: Trình soạn thảo mạnh mẽ, tích hợp Git, hỗ trợ JavaScript, React, Node.js cùng nhiều tiện ích mở rộng.
* **StarUML**: Hỗ trợ vẽ sơ đồ Use Case, Class Diagram, Sequence Diagram, dùng trong thiết kế hệ thống.
* **Case Studio**: Hỗ trợ mô hình hóa dữ liệu (ERD) và sinh script SQL cho MySQL.

**2.2.8 Đánh giá ưu nhược điểm các công nghệ**

**ReactJS**

* Ưu điểm: hiện đại, hiệu năng tốt, giao diện động mạnh.
* Nhược điểm: cần học JSX, khó SEO.

**Node.js + Express**

* Ưu điểm: cùng dùng JavaScript với frontend, non-blocking I/O.
* Nhược điểm: không phù hợp tính toán nặng, cần xử lý async tốt.

**MongoDB**

* Ưu điểm: linh hoạt, không ràng buộc cấu trúc.
* Nhược điểm: không mạnh trong join dữ liệu phức tạp.

**MySQL**

* Ưu điểm: dữ liệu có ràng buộc, mạnh với truy vấn phức.
* Nhược điểm: khó mở rộng theo chiều ngang, cứng nhắc với dữ liệu động.

**Google Maps API**

* Ưu điểm: trực quan, hỗ trợ định vị chính xác.
* Nhược điểm: cần cấu hình key đúng, giới hạn miễn phí.

Tổng kết: Các công nghệ trên đều phổ biến, có cộng đồng hỗ trợ lớn, phù hợp với mục tiêu đồ án, giúp hệ thống dễ mở rộng, triển khai thực tế và nâng cấp trong tương lai.

**2.2.8 So sánh, đánh giá ưu nhược điểm các công nghệ**

**1. Frontend - ReactJS**

* Ưu điểm:
  + Giao diện mượt mà, có khả năng cập nhật dữ liệu theo thời gian thực.
  + Cộng đồng lớn, nhiều tài liệu và thư viện hỗ trợ.
  + Cấu trúc component rõ ràng, dễ tái sử dụng.
* Nhược điểm:
  + Cần có kiến thức về JSX và lập trình hướng component.
  + Khó khăn hơn khi SEO so với các framework SSR (server-side rendering).

**2. Backend - Node.js & Express**

* Ưu điểm:
  + Sử dụng cùng ngôn ngữ JavaScript cả frontend và backend.
  + Dễ học, tốc độ xử lý nhanh, không chặn (non-blocking I/O).
  + Thư viện đa dạng, cộng đồng lớn.
* Nhược điểm:
  + Không phù hợp với các tác vụ tính toán nặng, xử lý CPU cao.
  + Cần kỹ thuật viên quản lý tốt lỗi bất đồng bộ.

**3. Cơ sở dữ liệu**

* MongoDB (NoSQL):
  + Ưu điểm: linh hoạt, không ràng buộc cấu trúc, dễ mở rộng.
  + Nhược điểm: không có hỗ trợ join mạnh như SQL, cần chuẩn hóa dữ liệu đúng cách.
* MySQL (RDBMS):
  + Ưu điểm: dữ liệu ràng buộc rõ ràng, dễ truy vấn logic.
  + Nhược điểm: khó mở rộng theo chiều ngang, khó xử lý dữ liệu không có cấu trúc.

**4. Bản đồ Google Maps Platform**

* Ưu điểm: độ chính xác cao, hỗ trợ đa dạng dịch vụ (maps, geocoding, distance, directions...)
* Nhược điểm: giới hạn sử dụng miễn phí, cần cấu hình key cẩn thận.

**5. Công cụ phát triển**

* Visual Studio Code:
  + Nhẹ, nhanh, hỗ trợ đa nền tảng, nhiều extension hữu ích.
* GitHub:
  + Quản lý phiên bản, chia nhánh phát triển, hỗ trợ cộng tác.
* StarUML & Case Studio:
  + Hỗ trợ tốt mô hình hóa UML và cơ sở dữ liệu (ERD), giúp hệ thống được thiết kế rõ ràng, dễ bảo trì.

Các công nghệ và công cụ được lựa chọn phù hợp với quy mô và tính chất của đề tài, đảm bảo việc phát triển hiệu quả, có thể mở rộng và triển khai thực tế sau này.

* **Git & GitHub**: Git là hệ thống quản lý mã nguồn phân tán phổ biến nhất hiện nay. GitHub là nền tảng lưu trữ và chia sẻ mã nguồn trực tuyến. Dự án sử dụng Git để theo dõi lịch sử thay đổi, phân nhánh phát triển và cộng tác nhóm hiệu quả thông qua GitHub.
* **Visual Studio Code (VS Code)**: Là trình soạn thảo mã nguồn hiện đại, hỗ trợ mạnh mẽ cho JavaScript, Node.js, React cùng với nhiều extension tiện lợi như Prettier, ESLint, GitLens... giúp tăng hiệu suất lập trình.
* **StarUML**: Công cụ mô hình hóa phần mềm, được sử dụng để vẽ sơ đồ Use Case, sơ đồ lớp, sơ đồ trình tự, mô hình quan hệ thực thể (ERD). Giúp biểu diễn trực quan các chức năng và cấu trúc hệ thống.
* **Case Studio**: Là công cụ chuyên vẽ mô hình cơ sở dữ liệu (ERD), hỗ trợ chuyển đổi mô hình thành mã SQL cho MySQL hoặc các hệ quản trị CSDL khác. Dễ dàng tổ chức bảng, khóa ngoại và chuẩn hóa cơ sở dữ liệu.
* **MySQL & MongoDB**:
  + MySQL là hệ quản trị cơ sở dữ liệu quan hệ, phù hợp với các hệ thống có cấu trúc dữ liệu rõ ràng. Hỗ trợ truy vấn SQL, khóa ngoại và chuẩn hóa tốt.
  + MongoDB là cơ sở dữ liệu NoSQL dạng tài liệu, phù hợp với các ứng dụng web hiện đại có dữ liệu linh hoạt, phi cấu trúc. Hệ thống có thể lưu thông tin người dùng, vị trí và dữ liệu liên quan đến tìm kiếm người hiến máu.

Các công cụ trên hỗ trợ toàn diện từ giai đoạn thiết kế, phát triển đến kiểm thử và triển khai hệ thống

CHƯƠNG III: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

* 1. **Giới thiệu đề tài:**

1. Dân số Việt Nam ngày càng tăng, nhu cầu khám chữa bệnh, phẫu thuật và cấp cứu kéo theo đó cũng tăng cao. Tuy nhiên, tình trạng khan hiếm máu trong các bệnh viện vẫn là một vấn đề nghiêm trọng, đặc biệt trong các tình huống khẩn cấp như tai nạn giao thông, sản khoa, phẫu thuật tim mạch, và các bệnh lý cần truyền máu định kỳ. Mặc dù phong trào hiến máu nhân đạo đã có nhiều tiến triển, nhưng phương pháp tổ chức và điều phối hiện vẫn còn thủ công, thiếu hiệu quả và chưa ứng dụng nhiều công nghệ thông tin.
2. Trên thực tế, nhiều người sẵn sàng hiến máu nhưng không biết địa điểm nào đang cần hoặc không được tiếp cận thông tin đúng thời điểm. Ngược lại, các bệnh viện không có công cụ tìm kiếm nhanh người hiến máu theo nhóm máu và vị trí địa lý gần nhất. Điều này dẫn đến sự lãng phí nguồn lực và chậm trễ trong cứu chữa bệnh nhân.
3. Vì vậy, việc xây dựng hệ thống "BLOOD-BRIDGE" – cầu nối hiến máu thông minh – là rất cần thiết và cấp thiết. Hệ thống hỗ trợ kết nối giữa người hiến máu và bệnh viện có nhu cầu theo thời gian thực, cung cấp tính năng tìm kiếm theo bán kính, nhóm máu và trạng thái sẵn sàng. Nhân viên bệnh viện có thể dễ dàng đăng nhập, tra cứu danh sách người hiến máu phù hợp, từ đó chủ động liên hệ và điều phối hiệu quả.
4. Hệ thống đồng thời cho phép người hiến máu đăng ký hồ sơ cá nhân, cập nhật nhóm máu, địa chỉ cư trú, trạng thái sẵn sàng và lịch sử hiến máu. Giao diện được thiết kế thân thiện, hiện đại, hỗ trợ tốt trên các thiết bị phổ biến hiện nay như PC và thiết bị di động.
5. Ngoài ra, BLOOD-BRIDGE còn tích hợp Google Maps API để định vị và hiển thị trực quan vị trí của người hiến máu gần bệnh viện, từ đó giúp tăng tốc độ phản hồi và xử lý trong tình huống khẩn cấp. Đồ án được triển khai với frontend sử dụng ReactJS, backend sử dụng Node.js kết hợp Express, và cơ sở dữ liệu MongoDB hoặc MySQL tùy theo yêu cầu triển khai thực tế.
6. Bài toán thực tế, công nghệ hiện đại, mục tiêu nhân đạo – tất cả tạo nên tính cấp thiết và ý nghĩa to lớn cho hệ thống BLOOD-BRIDGE

.

**1**.2 Tổ chức và các chức năng của hệ thống

**1. Người hiến máu (Donor)** Trước khi đăng ký tài khoản, người hiến máu có thể truy cập hệ thống để tìm hiểu các thông tin liên quan đến quy trình hiến máu, lợi ích, điều kiện và trách nhiệm. Sau đó, người dùng cần điền đầy đủ thông tin cá nhân để tạo hồ sơ hiến máu bao gồm: họ tên, giới tính, ngày sinh, địa chỉ liên lạc, nhóm máu, số điện thoại, email và vị trí hiện tại. Sau khi tài khoản được xác nhận, người hiến máu sẽ sử dụng hệ thống để:

* Cập nhật trạng thái sẵn sàng hiến máu.
* Theo dõi lịch sử hiến máu của bản thân.
* Nhận thông báo từ hệ thống khi có bệnh viện cần máu tương ứng với nhóm máu và vị trí.
* Tùy chỉnh thông tin cá nhân khi cần thiết.

**2. Nhân viên bệnh viện (Hospital Staff)** Nhân viên y tế là người có tài khoản được phân quyền bởi quản trị viên. Sau khi đăng nhập, họ có thể sử dụng hệ thống để hỗ trợ việc tìm kiếm người hiến máu nhanh chóng, cụ thể:

* Nhập địa chỉ của bệnh viện để xác định vị trí trung tâm tìm kiếm.
* Lựa chọn nhóm máu cần tìm và bán kính phù hợp.
* Xem danh sách người hiến máu có nhóm máu tương ứng, trong bán kính được chọn, kèm theo thông tin liên hệ và trạng thái sẵn sàng.
* Xem vị trí người hiến máu hiển thị trên bản đồ.
* Gửi thông báo trực tiếp đến người hiến máu thông qua email hoặc hệ thống (có thể mở rộng bằng SMS).
* Ghi nhận trạng thái liên hệ và theo dõi tình trạng đáp ứng.

**3. Quản trị viên hệ thống (Admin)** Quản trị viên đóng vai trò đảm bảo vận hành hệ thống và bảo mật thông tin. Các chức năng quản trị bao gồm:

* Phê duyệt và kích hoạt tài khoản bệnh viện.
* Quản lý, cập nhật và xoá tài khoản người hiến máu, bệnh viện.
* Thiết lập giới hạn bán kính mặc định, danh mục nhóm máu, trạng thái hệ thống.
* Giám sát số lượng người hiến máu, số lần liên hệ thành công và tỷ lệ phản hồi từ người dùng.
* Cung cấp báo cáo, thống kê định kỳ nhằm phục vụ công tác quản lý y tế.

**1.3** Mô tả yêu cầu với các nghiệp vụ chính

***1.3.1*** *Quy trình tiếp nhận và xử lý máu hiến từ bệnh viện*

Khi có nhu cầu tiếp nhận máu, bệnh viện đăng nhập vào hệ thống và tạo yêu cầu hiến máu, bao gồm thông tin nhóm máu, số lượng cần và mức độ ưu tiên. Hệ thống sẽ tự động lọc danh sách những người hiến máu phù hợp trong bán kính địa lý nhất định và gửi thông báo đến họ. Người hiến máu xem xét yêu cầu và xác nhận có tham gia hay không. Bệnh viện theo dõi danh sách người đăng ký và tổ chức tiếp nhận máu theo lịch. Sau khi hoàn tất, máu sẽ được xử lý, sàng lọc và nhập kho. Hệ thống đồng thời cập nhật trạng thái yêu cầu (hoàn thành/đang chờ/thiếu máu) và điều chỉnh tồn kho máu tương ứng.

* + 1. *Quy trình hiến máu*

Người hiến máu bắt đầu bằng việc tạo tài khoản trên hệ thống và điền thông tin cá nhân như họ tên, ngày sinh, nhóm máu, tiền sử sức khỏe. Hệ thống sẽ thực hiện kiểm tra điều kiện hiến máu dựa trên các tiêu chí y tế đầu vào. Khi có bệnh viện trong khu vực có nhu cầu nhóm máu phù hợp, người hiến máu sẽ được hiển thị thông báo và có thể lựa chọn đặt lịch hẹn hiến máu tại điểm tiếp nhận gần nhất. Bệnh viện xác nhận lịch và gửi mã QR hoặc mã xác nhận về cho người hiến. Đến ngày hẹn, người hiến máu đến địa điểm, quét mã hoặc nhập mã để xác nhận danh tính. Sau khi hoàn tất quá trình hiến, bệnh viện cập nhật trạng thái vào hệ thống. Người hiến nhận được giấy chứng nhận điện tử, điểm thưởng hoặc hỗ trợ bồi dưỡng nếu có. Hệ thống cũng sẽ tự động thiết lập lời nhắc cho lần hiến máu tiếp theo dựa trên thời gian nghỉ cần thiết theo quy định y tế.

* + 1. *Quy trình quản trị hệ thống*

Quản trị viên đăng nhập vào hệ thống để kiểm duyệt các tài khoản bệnh viện và người hiến máu mới đăng ký, đồng thời xác minh thông tin, giấy tờ liên quan. Trong quá trình vận hành, admin theo dõi dữ liệu hiến máu trên toàn hệ thống, hỗ trợ kết nối giữa bệnh viện và người hiến trong các trường hợp khẩn cấp. Khi cần, quản trị viên tiến hành cập nhật kho máu hoặc điều chỉnh dữ liệu để đảm bảo thông tin nhất quán. Ngoài ra, admin có thể tạo và gửi thông báo hệ thống về các chiến dịch hiến máu cộng đồng, cảnh báo thiếu máu hoặc các thông tin hướng dẫn.

***1.3.****4 Quy trình đặt máu của bệnh viện*

Bệnh viện thực hiện đăng ký tài khoản và nạp tiền vào hệ thống thông qua hình thức chuyển khoản hoặc ví điện tử. Sau khi số dư được xác nhận, bệnh viện tạo yêu cầu mua máu với các thông tin cụ thể như nhóm máu, số lượng và mức giá sẵn sàng chi trả cho mỗi đơn vị máu. Hệ thống sẽ kiểm tra và thống kê số lượng người hiến máu phù hợp đang sẵn sàng trong khu vực. Nếu có đủ người hiến máu tiềm năng, bệnh viện tiến hành xác nhận đơn hàng và thanh toán trước toàn bộ hoặc một phần chi phí theo chính sách. Sau khi thanh toán thành công, hệ thống sẽ gửi thông báo hiến máu đến những người hiến phù hợp, kèm theo thông tin thời gian, địa điểm, mã xác nhận và mức bồi dưỡng cụ thể

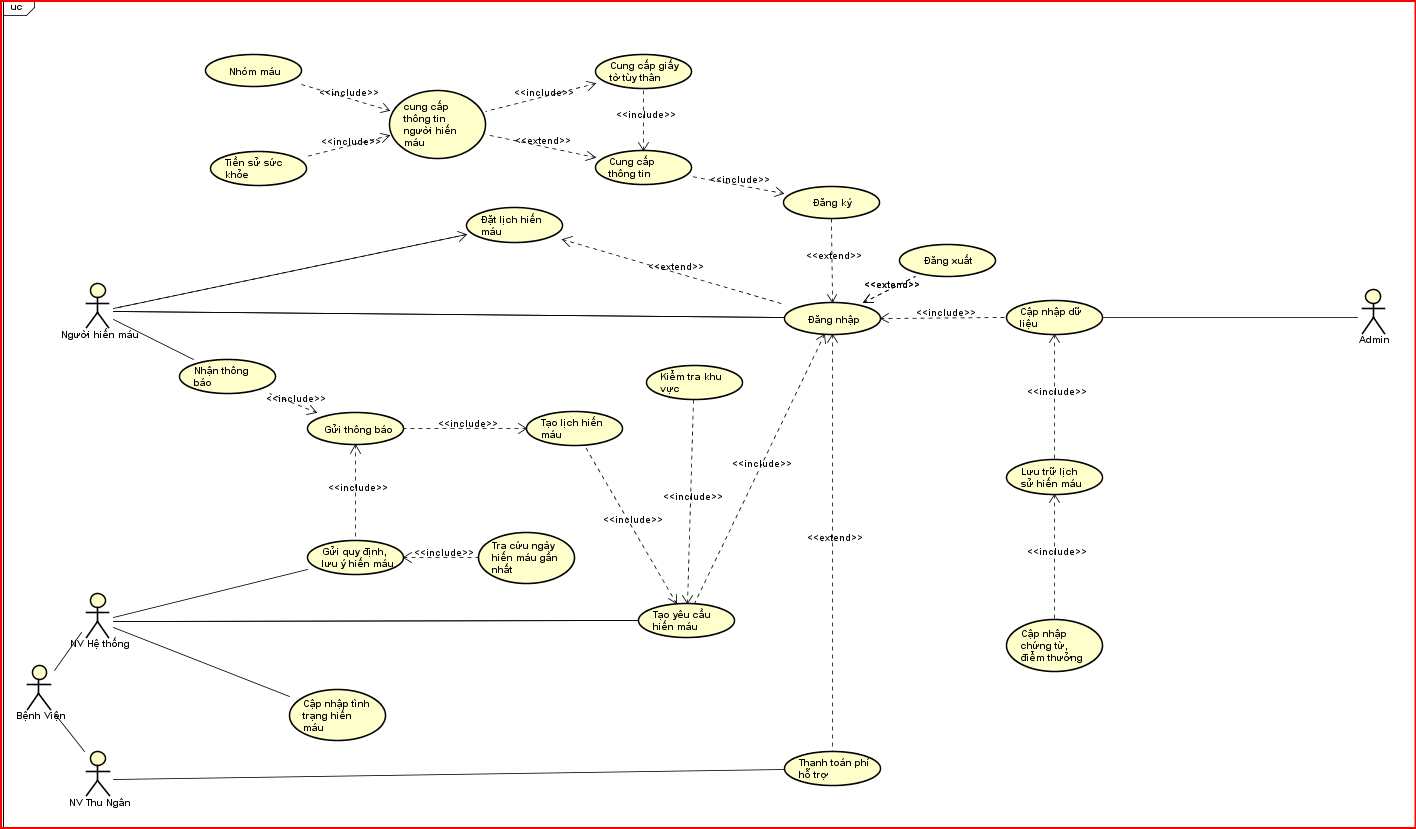
***1.3.5*** *Quy trình hiến máu có trả tiền*

Người hiến máu truy cập hệ thống để xem danh sách các yêu cầu hiến máu có bồi dưỡng được bệnh viện đăng tải. Sau khi lựa chọn yêu cầu phù hợp về nhóm máu, vị trí, thời gian và mức hỗ trợ, người hiến đăng ký tham gia. Đến thời gian đã chọn, người hiến đến đúng địa điểm hiến máu, trình mã xác nhận hoặc quét mã QR do hệ thống cung cấp. Nhân viên y tế tại bệnh viện tiếp nhận, xác minh và thực hiện quá trình lấy máu. Khi hiến máu hoàn tất và được xác nhận trên hệ thống, hệ thống sẽ tự động tính toán chi phí, trừ hoa hồng trung gian (nếu có) và chuyển khoản số tiền còn lại đến người hiến máu thông qua phương thức đã đăng ký (chuyển khoản ngân hàng, ví điện tử như Momo, ZaloPay,...).2.1 Use Case : Tổng quát

Ảnh có chứa biểu đồ, hàng, bản đồ, văn bản

Nội dung do AI tạo ra có thể không chính xác.

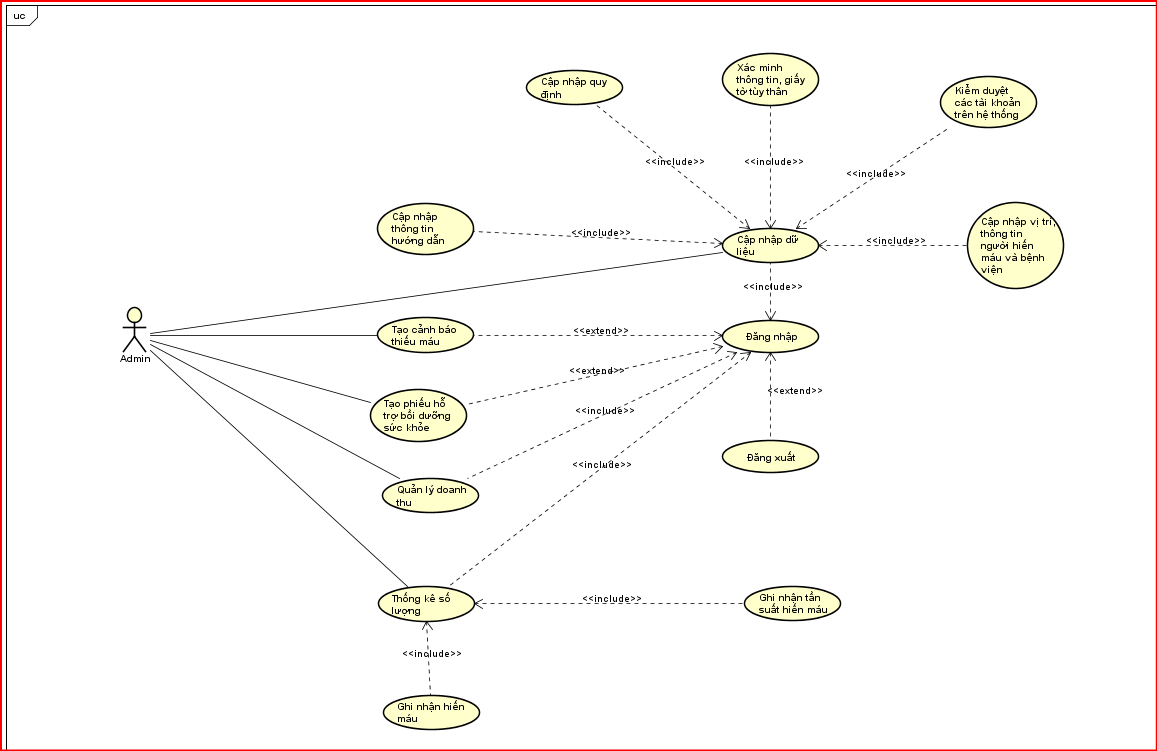
*Hình 1 : Mô hình Use Case tổng quát*

****2.2 Use Case: Quy trình hiến máu

*Hình 1.1 : Mô hình Use Case quản lí “ Quy trình hiến máu”*

* Đặc tả Use Case:

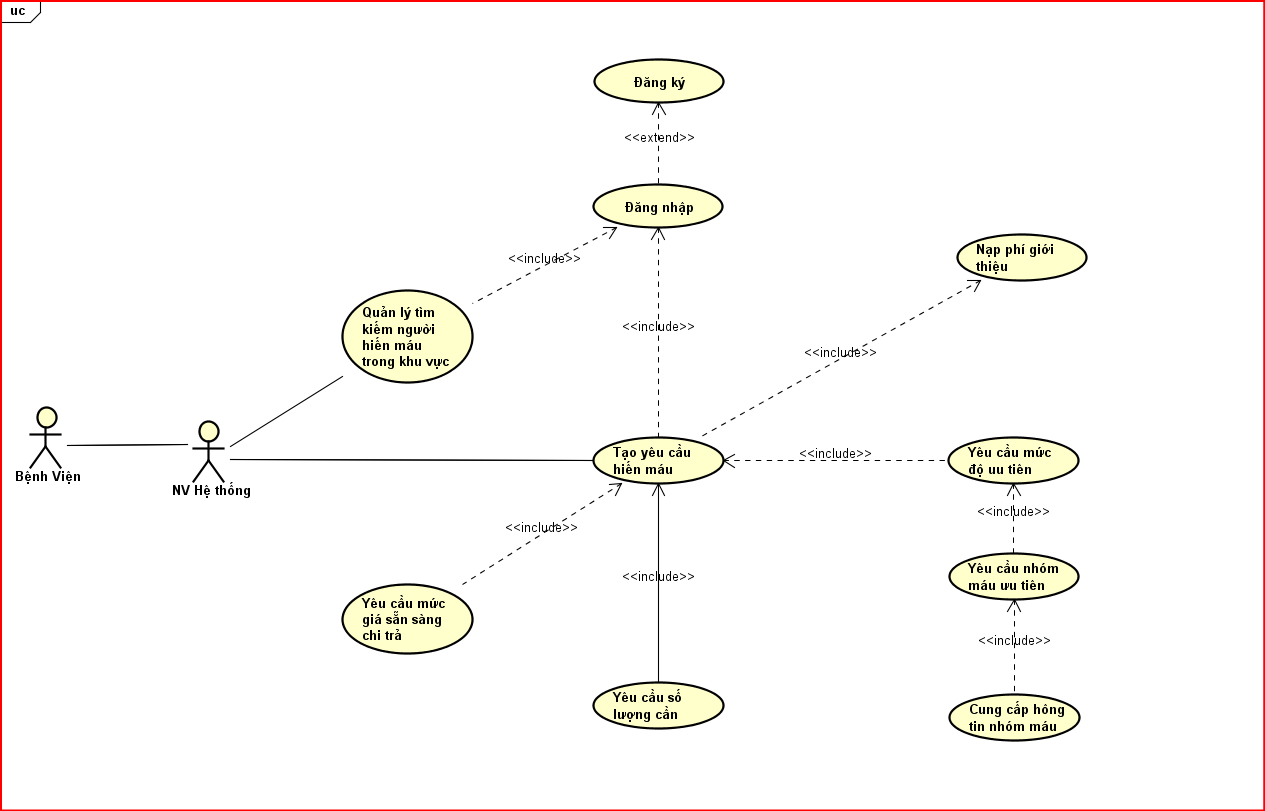
|  |  |  |
| --- | --- | --- |
| |  | | --- | | ID Use Case | | 2.2 |
| Tên Use Case | Đăng ký hiến máu |
| Tác nhân chính | Người hiến máu (Donor) |
| Mức | Cao |
| Mô tả ngắn gọn | Người dùng nhập thông tin cá nhân, nhóm máu, vị trí và trạng thái để sẵn sàng hiến máu khi cần. |
| Tiền điều kiện | Người dùng đã đăng ký tài khoản và đăng nhập hệ thống. |
| Hậu điều kiện | Thông tin hồ sơ được lưu vào hệ thống, trạng thái hiến máu là "Sẵn sàng". |
| Kích hoạt | Người dùng chọn chức năng "Đăng ký hiến máu" trên giao diện hệ thống |
| Chuỗi sự kiện chính | 1. Hệ thống hiển thị form đăng ký hiến máu.  2. Người dùng nhập thông tin cá nhân (họ tên, nhóm máu, địa chỉ, số điện thoại...).  3. Người dùng chọn trạng thái "Sẵn sàng hiến máu".  4. Người dùng nhấn nút “Lưu thông tin”.  5. Hệ thống xác nhận và lưu thông tin hồ sơ. |
| Ngoại lệ | - Nhập thiếu thông tin bắt buộc → hiển thị cảnh báo.  - Lỗi kết nối cơ sở dữ liệu → thông báo lỗi và không lưu thông tin. |

****2.3 Use Case : Quy trình quản trị hệ thống

*Hình 1.2 : Mô hình Use Case quản lí “Quy trình quản trị hệ thống”*

* Đặc tả Use Case:

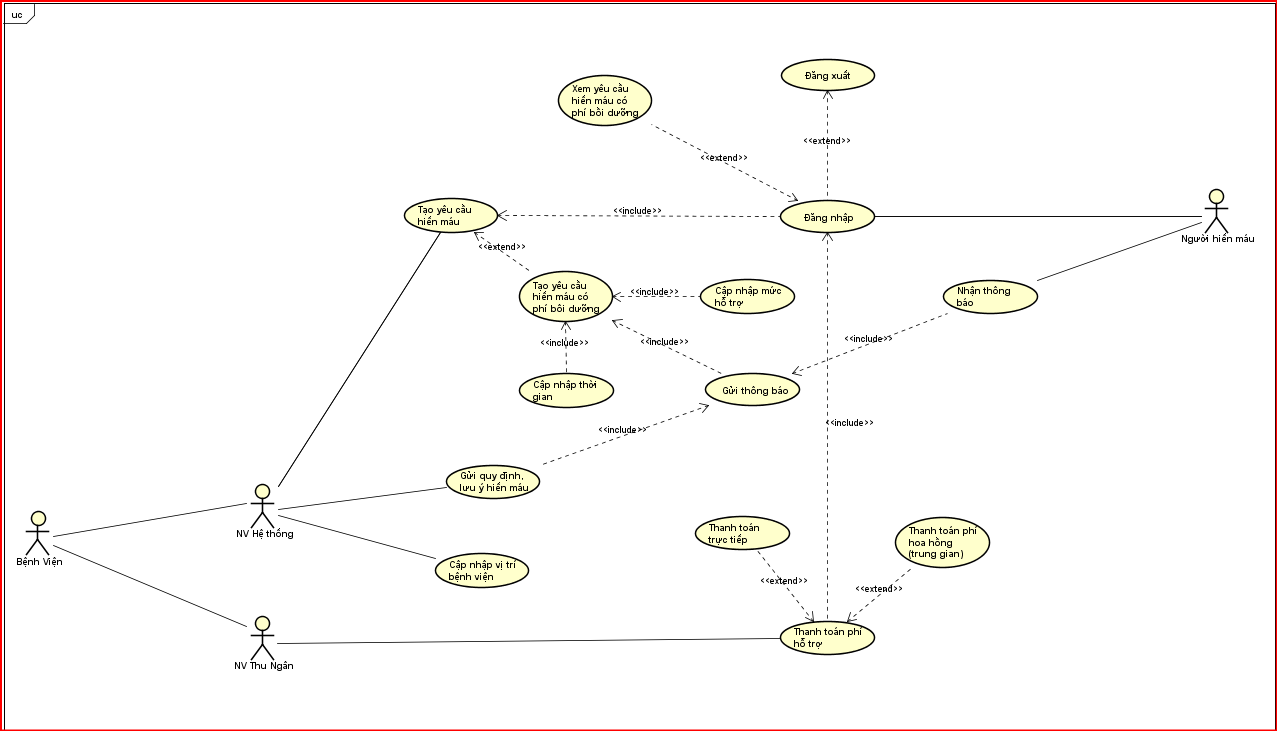
|  |  |  |  |
| --- | --- | --- | --- |
| |  | | --- | | ID Use Case |  |  | | --- | |  | | 2.3 |
| Tên Use Case | Quản trị hệ thống |
| Tác nhân chính | Quản trị viên (Admin) |
| Mức | Rất cao |
| Mô tả ngắn gọn | Admin thực hiện các thao tác cấu hình hệ thống, quản lý tài khoản và thống kê dữ liệu. |
| Tiền điều kiện | Admin đã đăng nhập hệ thống bằng tài khoản hợp lệ. |
| Hậu điều kiện | Các thay đổi được cập nhật và lưu trữ trong hệ thống. |
| Kích hoạt | Quản trị viên |
| Chuỗi sự kiện chính | 1. Admin đăng nhập vào hệ thống.  2. Truy cập trang Quản trị.  3. Tạo/sửa/xoá tài khoản người hiến máu hoặc bệnh viện.  4. Cập nhật quy định hệ thống như phạm vi tìm kiếm, danh sách nhóm máu.  5. Theo dõi thống kê số lượng người hiến, lượt gửi yêu cầu, tỷ lệ phản hồi.6. Cấu hình thông tin liên hệ, hướng dẫn sử dụng, nội dung cảnh báo mặc định. |
| Ngoại lệ | - Không thể kết nối server hoặc lỗi cơ sở dữ liệu.- Dữ liệu nhập không hợp lệ.- Không có quyền truy cập vào chức năng quản lý nâng cao. |

****2.4 Use Case : Quy trình đặt máu bệnh viện

*Hình 1.3 : Mô hình Use Case quản lí “Quy trình đặt máu của bệnh viện”*

* Đặc tả Use Case:

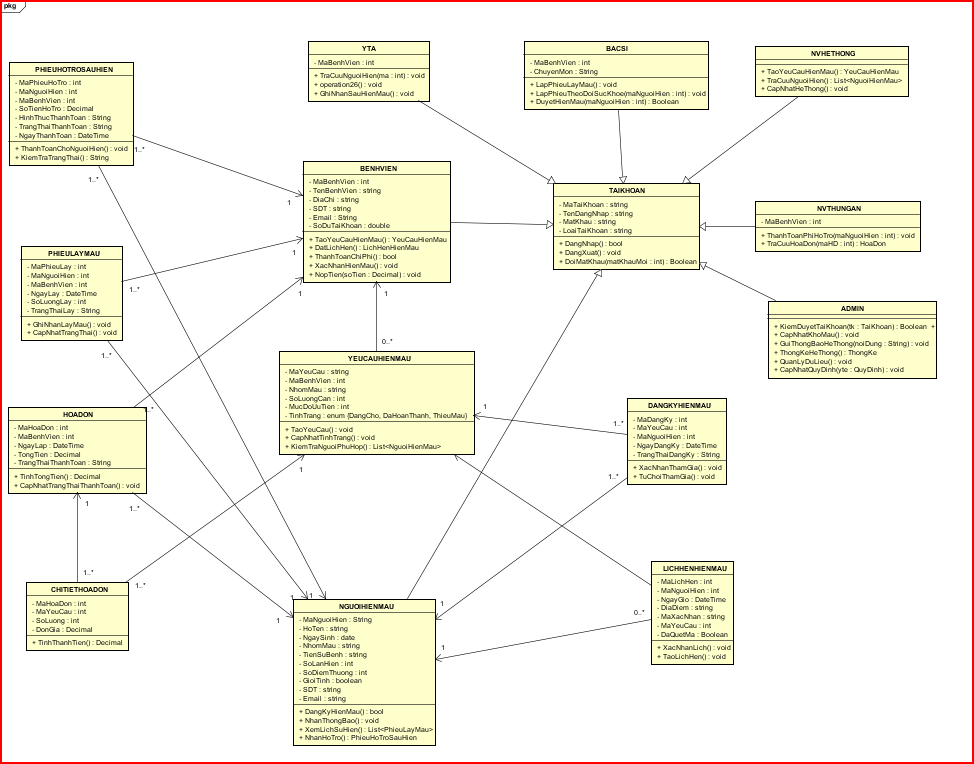
|  |  |  |  |
| --- | --- | --- | --- |
| |  | | --- | | ID Use Case |  |  | | --- | |  | | 2.4 |
| Tên Use Case | Tạo yêu cầu hiến máu |
| Tác nhân chính | Nhân viên bệnh viện / Bệnh viện |
| Mức | Trung bình |
| Mô tả ngắn gọn | Nhân viên bệnh viện gửi yêu cầu hiến máu theo nhóm máu, vị trí, ưu tiên, số lượng cần. |
| Tiền điều kiện | Nhân viên bệnh viện gửi yêu cầu hiến máu theo nhóm máu, vị trí, ưu tiên, số lượng cần. |
| Hậu điều kiện | Yêu cầu hiến máu được gửi đến hệ thống, lưu vào danh sách chờ xử lý và hiển thị tới người hiến máu phù hợp. |
| Kích hoạt | Nhân viên bệnh viện |
| Chuỗi sự kiện chính | 1. Nhân viên đăng nhập vào hệ thống.  2. Chọn chức năng "Tạo yêu cầu hiến máu".  3. Nhập các thông tin: nhóm máu cần, số lượng đơn vị máu, mức độ ưu tiên, trạng thái khẩn cấp.  4. Chọn phạm vi bán kính tìm kiếm.  5. Hệ thống hiển thị bản đồ và danh sách người hiến máu phù hợp.  6. Gửi yêu cầu đến các người hiến máu được lọc theo điều kiện.  7. Hệ thống lưu lại thông tin yêu cầu và ghi nhận trạng thái gửi. |
| Ngoại lệ | - Nếu nhập thiếu thông tin bắt buộc → cảnh báo lỗi và không gửi được.- Không tìm thấy người hiến máu trong khu vực → hiển thị cảnh báo và đề xuất mở rộng bán kính.- Lỗi hệ thống trong lúc gửi yêu cầu → hiển thị thông báo lỗi. |

****2.5 Use Case : Quy trình hiến máu có trả tiền

*Hình 1.4 : Mô hình Use Case quản lí “Quy trình hiến máu có trả tiền”*

* Đặc tả Use Case:

|  |  |  |  |
| --- | --- | --- | --- |
| |  | | --- | | ID Use Case |  |  | | --- | |  | | 2.5 |
| Tên Use Case | Hiến máu có trả phí |
| Tác nhân chính | Người hiến máu, Nhân viên hệ thống, Nhân viên thu ngân |
| Mức | Trung bình |
| Mô tả ngắn gọn | Quản lý quy trình hiến máu có hỗ trợ chi phí và xác nhận thanh toán. |
| Tiền điều kiện | Người hiến máu đã đăng ký, đăng nhập và nhận được yêu cầu hiến máu từ hệ thống hoặc bệnh viện. |
| Hậu điều kiện | Người hiến máu nhận được xác nhận hiến máu và thanh toán (nếu có). |
| Kích hoạt | Người hiến máu / Nhân viên hệ thống |
| Chuỗi sự kiện chính | 1. Người hiến máu đăng nhập hệ thống và nhận yêu cầu.  2. Chọn hình thức hiến máu có trả phí.  3. Xác nhận mức hỗ trợ, thông tin chuyển khoản hoặc trực tiếp.  4. Đến địa điểm được chỉ định và hiến máu.  5. Nhân viên hệ thống xác nhận đã hiến máu.  6. Nhân viên thu ngân xử lý thanh toán theo chính sách (tiền mặt hoặc chuyển khoản).  7. Hệ thống ghi nhận hoàn tất quy trình và gửi biên nhận. |
| Ngoại lệ | - Người hiến máu không đến điểm hiến đúng giờ → hủy yêu cầu.- Không xác minh được người hiến máu tại điểm hiến → từ chối thanh toán.- Lỗi hệ thống thanh toán hoặc thiếu thông tin → chuyển xử lý thủ công |

4: Mô hình lớp

*Hình 2 : Mô hình class hệ thống “ Cầu nối hiến máu”*

TÀI LIỆU THAM KHẢO

[1] [www.w3schools.com](http://www.w3schools.com)

[2] <https://mui.com/material-ui/>

[3] <https://code.visualstudio.com/>

[4] <https://codelearn.io/>

[5] <https://monhoc.vn/tai-lieu/phan-tich-thiet-ke-he-thong-thong-tin-huong-dan-su-dung-case-studio-221-de-tao-mo-hinh-dfd-va-mo-hinh-er-1404/>

[6] <https://fullstack.edu.vn/courses/html-css>

[7] <https://apidog.com/vi/blog/know-about-apis-vi/?utm_source=google_dsa&utm_medium=g&utm_campaign=22417619241&utm_content=186231380868&utm_term=&gad_source=1&gad_campaignid=22417619241&gbraid=0AAAAA-gKXrADOZVpO4eMAIVXoWQcz2Y1P&gclid=CjwKCAjw56DBBhAkEiwAaFsG-oY3BTWF-SfC6lrIFvUZaP0yGkKZXVxJ3fuN0SVE7c4q_Yx0WOUVNBoCKCgQAvD_BwE>

[8] <https://viblo.asia/p/di-tim-chiec-structure-hoan-hao-cho-ung-dung-react-Az45b41NZxY>

PHIẾU PHÂN CÔNG NHIỆM VỤ

**THỰC HIỆN ĐỒ ÁN HỌC PHẦN: ĐỒ ÁN CƠ SỞ**

**1. Tên đề tài:** Hệ thống Cầu Nối Hiến Máu

**2. Giảng viên hướng dẫn:** Th.S Võ Hoàng Khang

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **MSSV** | **Họ tên** | **Công việc phân công** | **Mức độ đóng góp (%)** |
| 1 | 2280602255 | Trần Cao Quỳnh Như |  | 100 |
| 2 | 2280601249 | Trần Hoàng Huy |  | 100 |

**PHỤ LỤC**

Phụ lục 1: Phiếu phân công nhiệm vụ