

Blood Donation Management System

1. Introduction

Introduction

The project blood bank management system is known to be a pilot project that is designed for the blood bank to gather blood from various sources and distribute it to the needy people who have high requirements for it.

The software is designed to handle the daily transactions of the blood bank and search the details when required.

It also helps to register the details of donors, blood collection details as well as blood issued reports.

Problem Statement

Scarcity of rare blood group.

Unavailability of blood during emergency.

Less awareness among people about blood donation and blood transfusion.

Deaths due to lack of blood during operations.

The Blood Bank Management System project aims to make all the procedures automated and therefore with computer system it can be faster and more accurate.

This project is a high-quality software to manage all these cumbersome jobs.

Existing System

The operation of the blood bank still now is maintained in the manual system.

The operation is tedious, time consuming and space consuming.

It creates room for errors as the data is entered manually by the persons.

It includes the risk of the documents being lost over years and maintenance of the records is difficult.

The data recorded during testing or while acquiring the details of different aspects of blood bank management system is not so accurate and precise.

Maintaining the stock of blood and the daily transactions without computerization also poses a challenge

Objective:

This applied research aims to design, develop and implement online blood bank management system. This web-based application provides:

- To ensure hospital to have good supply or inventories of blood bags.
- To check the availability of blood bags anytime.
- To manage the information of its blood donor.
- Function to check if the person donate blood for the last 3 months.
- To allow good documentation about the donor and its blood donation activities.
- Support fast searching to find match blood bags for the right person.

Features:

1. Provides the searching facilities of blood based on blood group.
2. It keeps all the information of a donor including blood group and his/ her health.
3. Person can donate or sell his/ her blood to acceptor, hospitals and plasma donation centre associated with the application.
4. Provides the e-certificate for blood donation, by which a donor can accept blood (if required) for free or less cost.

Advantages:

Online Blood Bank project aims at maintaining all the information pertaining to blood donors, different blood groups available in each blood bank and help them manage in a better way. The blood donors can register to this system by entering their profile information.

2. SRS DOCUMENT

A software requirements specification (SRS) is a document that captures complete description about how the system is expected to perform. It is usually signed off at the end of requirements engineering phase.

FUNCTIONAL REQUIREMENTS

A Functional Requirement (FR) is a description of the service that the software must offer. It describes a software system or its component. A function is nothing but inputs to the software system, its behaviour, and outputs. It can be a calculation, data manipulation, business process, user interaction, or any other specific functionality which defines what function a system is likely to perform.

Requirements in Software Engineering are also called Functional Specification.

- For the product we search it should show the relevant information about the product.
- Store the images of the products to the cloud rather than storing in file storage.
- Take feedback from the user about the product.
- The final amount generated during the checkout should be same as the product the user has chosen.

NON-FUNCTIONAL REQUIREMENTS

Non-Functional Requirement (NFR) specifies the quality attribute of a software system. They judge the software system based on Responsiveness, Usability, Security, Portability and other non-functional standards that are critical to the success of the software system.

- Availability Requirement: The system is accessible to the user at all times and is used 24 hours a day, 365 days a year. The system must be available 24 hours a day, seven days a week.
- Performance Requirement: Depending on whether or not some updates have occurred, the information is refreshed at regular intervals. From the time the request is submitted, the system must react to the member in fewer than two seconds.

- Accuracy Requirement: The system must guarantee access dependability of 100 percent.
- Efficiency - Once a user has gained a thorough understanding of the system through engagement, he can effortlessly complete the task.
- Operability - The system's user interface will be consistent.
- User-Friendly Interface-It is more understandable to users because of the user-friendly interface.

Non-Functional requirements in Software Engineering allow you to impose constraints or restrictions on the design of the system across the various agile backlogs.

MINIMUM HARDWARE REQUIREMENTS:

Processor-IntelCorei5

- HARDDISK-2 GB
- RAM- 4GB

MINIMUM SOFTWARE REQUIREMENTS:

- Scripting Languages-HTML5, Tailwind CSS, JAVASCRIPT
- Operating System -Windows 10(64bit)
- IDEs - Visual studio code
- Libraries –Reactjs, ExpressJS, MySQL, React-redux
- Tools--

3. Review of the Literature

Introduction

This section discusses findings and observations done by some research works on web-based blood bank management system. The gathered information on these related papers strengthens and supports the research study.

Literature Studies

According to Teena, C.A, Sankar, K. and Kannan, S. (2014) in their study entitled “A Study on Blood Bank Management”, they defined Blood Bank Information System as an information management system that contributes to the management of donor records and blood bank. Their system allowed an authorized blood bank administrator to sign in with a password to manage easily the records of donors and patients who need blood. The system provided many features including the central database, quick access to the system content through the login, includes the search code to find donors on a given basis, and the ease of adding and updating donor data. The main aim of the system was to complete the process of the blood bank. This system was designed to suit all types of blood banks. Once successful in the implementation of the application, it can be applied and rolled out in several blood banks. This application contains User Login Screen, Blood Management, Menu Form, Blood Stock, Donor Management, Donor Registration, Blood Reservation, Donor Blood Test, Recipient Management and Blood Reservation. In similar manner, the researchers planned in their application to have hospital administrator, doctors, and blood bank receptionists as users. The authors did not mention the research method they used, and failed to provide screenshots of the system prototypes, making difficult for the researchers to visualize their application. No discussion also for their respondents, samples and sampling techniques used. Subsequently, the researchers planned to provide figures to explain the system, screenshots of system prototypes, and other diagrams that can help other researchers to visualize the development of web-based blood bank management system. Also, the researchers will explicitly discuss its research methods, sampling procedures, and statistical treatment to be used for analysing the gathered data.

On the other hand, study entitled “Blood Bank Management System” done by Kumar, R., Singh, S. and Ragavi, V.A. (2017), the researchers developed a web-based blood management which assists the blood donor records management, and provides ease of control

in the distribution of blood products in various parts of the country considering demands of hospitals. The developed system was scalable and adaptable to meet the complex needs usually of a blood bank. Based on this study, since entering the details about the blood donors and related records were done manually, thus, tracking of blood donation activities was difficult and complicated, and even led to erroneous information. Subsequently, the researchers mentioned that manual-based system can be waste of time, lead to the error-prone results, consumes a lot of manpower, lacks data security, data retrieval requires a lot of time, reports consume a long time to produce, and there is less precise accuracy on the results. As such, by developing and implementing a web-based blood management information system, there was a quick and timely access to donor records, and the system provided management timely, confidential and secured medical reports. There were three (3) users in the system, namely: Administrator, Donor, and Acceptor. Each user has been given user ID and password to identify their identity. The said application was developed using ASP.NET, C#.NET, and using SQL Server 2000/2005 for the database. The research paper failed to mention the methods of research used.

In this study, the researchers learnt the importance of implementing a web-based blood donation management system in handling records for blood donors and blood donation activities to ensure accurate and readily available information for blood transfusion services. Indeed, the impact of using Information Technology on hospitals provides better healthcare services for the public. Likewise, the researchers learnt that there are programming languages suitable for web-based applications such as ASP.NET, PHP, to name a few.

In the study entitled “Blood Bank Management System Using Rule-Based Method” undertaken by 4Liyana, F. (2017), it found out that it is important for every hospital to use an information system to manage data in blood bank. Also, it observed that the manual system has disadvantages for the user and the hospital. One of the disadvantages identified was the blood bank staff should enter the donor details in each time he/she donate blood in which led to duplicate data of the donor and also the data may be lost or missing after period of time. Thus, the author developed a web-based system to help the blood bank to record the donor details fast and easy. The system used rule-based decisions to ensure to have a right decision on right time. Also, system can send messages to donors if any particular blood type is needed. She developed blood bank system based on incremental model. She had chosen this model

because the system can be developed through cycle of phase and also because of the advantages of this model such as:

- I. Easy to understand to flow of the phases.
- II. Changes possible in the middle of any phases.
- III. The system can be developed even if there is an error in the middle and it can be corrected in testing phase.

In this study, the researchers observed that the developer failed to include in the system the function to check the availability of blood bags, and to check the shelf life or expiration of blood bags or products. As such, the researchers will include these in their developed system to enhance safety for blood transfusion.

Conclusion

The purpose of these literature reviews was to collect information on how an information system helped the management of blood banks. Based on the reviews, it was found out that web-based blood bank systems provide convenience, efficiency and security to the system users and hospitals compared to the manual systems. It was found out that manual systems have many disadvantages that disappoint and dissatisfy users. Indeed, online blood bank applications make work easy, and ensures fast retrieval of data when needed.

BEE DATA

4. Methods and Procedures

Introduction

This section presents the research methodology used in the study, the research design, and the data collection process. This section also presents the theoretical or conceptual framework of the study, the sampling plan, and tools to be used for data analysis.

Requirement Analysis: The first step is to identify the requirements for the system. This involves conducting a thorough analysis of the blood donation process and understanding the needs of the blood banks, hospitals, and other stakeholders. This information is used to develop a detailed list of functional and non-functional requirements for the system.

Design: Based on the requirements analysis, the system is designed. This involves creating a system architecture, designing the database schema, and developing user interface wireframes. The design phase also includes identifying the technologies and tools that will be used to develop the system.

Development: The system is developed using html, CSS, JavaScript and php in Vs Code. The development process involves creating the necessary database tables, implementing the business logic, and developing the user interface. The system is developed in iterations, with each iteration adding new functionality and features.

Testing: Once the system is developed, it is tested to ensure that it meets the requirements and is free of bugs and errors. The testing process includes unit testing, integration testing, and system testing. The system is also tested for performance, scalability, and security.

Deployment: After testing, the system is deployed to a production environment. This involves setting up the necessary hardware and software, configuring the system, and migrating data from the old system, if applicable. The system is then made available to the end-users.

Requirement Analysis

Requirements	Purpose
Donar	To donate the blood
User/Accepter	To receive the blood
Admin	To manage the operation between donor and accepter



5. SOFTWARE DESCRIPTION

NODE JS

Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on a JavaScript Engine (i.e., V8 engine) and executes JavaScript code outside a web browser, which was designed to build scalable network applications. Node.js lets developers use JavaScript to write command line tools and for server-side scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm, unifying web-application development around a single programming language, rather than different languages for server-side and client-side scripts.

Node.js has an event-driven architecture capable of asynchronous I/O. These design choices aim to optimize throughput and scalability in web applications with many input/output operations, as well as for real-time Web applications (e.g., real-time communication programs and browser games).

EXPRESS JS

Express.js, or simply Express, is a back-end web application framework for building RESTful APIs with Node.js, released as free and open-source software under the MIT License. It is designed for building web applications and APIs.[3] It has been called the de facto standard server framework for Node.js.

React Redux

Redux is an open-source JavaScript library used to manage application state. React uses Redux for building the user interface. It was first introduced by Dan Abramov and Andrew Clark in 2015.

React Redux is the official React binding for Redux. It allows React components to read data from a Redux Store, and dispatch Actions to the Store to update data. Redux helps apps to scale by providing a sensible way to manage state through a unidirectional data flow model.

React Redux is conceptually simple. It subscribes to the Redux store, checks to see if the data which your component wants have changed, and re-renders your component.

HTML5

The Hypertext Markup Language or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document. HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page.

HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Other tags such as surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags but use them to interpret the content of the page. HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

CSS3

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. CSS is designed to enable the separation of presentation and content, including layout, colours, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate.

CSS file which reduces complexity and repetition in the structural content as well as enabling the .CSS file to be cached to improve the page load speed between the pages that share the file and its formatting. Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as onscreen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device. The name cascading comes from the specified priority scheme to determine

which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.

JAVASCRIPT

JavaScript (commonly abbreviated as JS) is a lightweight, interpreted, object-oriented programming language with first-class functions. It's best known as a scripting language for Web pages, although it's also used in a variety of non-browser applications. It's a multiparadigm, prototype-based scripting language that supports object-oriented, imperative, and functional programming techniques. JavaScript is a client-side scripting language that may be used to create and programmed how web pages react when an event occurs. JavaScript is a scripting language that is both simple to learn and powerful, and it is commonly used to control the behavior of online pages.

Tailwind CSS

unlike other CSS frameworks like Bootstrap, it does not provide a series of predefined classes



for elements such as buttons or tables. Instead, it creates a list of "utility" CSS classes that can be used to style each element by mixing and matching. Tailwind CSS is an open-source CSS framework. The main feature of this library is that,

6. SYSTEM DESIGN

Introduction to UML

Unified Modelling Language (UML) is a general-purpose modelling language. The main aim of UML is to define a standard way to visualize the way a system has been designed. It is quite similar to blueprints used in other fields of engineering. UML is not a programming language; it is rather a visual language. We use UML diagrams to portray the behaviour and structure of a system. UML helps software engineers, businessmen and system architects with modelling, design and analysis. The Object Management Group (OMG) adopted Unified Modelling Language as a standard in 1997. It's been managed by OMG ever since. International Organization for Standardization (ISO) published UML as an approved standard in 2005. UML has been revised over the years and is reviewed periodically.

Why we need UML

Complex applications need collaboration and planning from multiple teams and hence require a clear and concise way to communicate amongst them.

Businessmen do not understand code. So, UML becomes essential to communicate with non-programmers' essential requirements, functionalities and processes of the system.

A lot of time is saved down the line when teams are able to visualize processes, user interactions and static structure of the system.

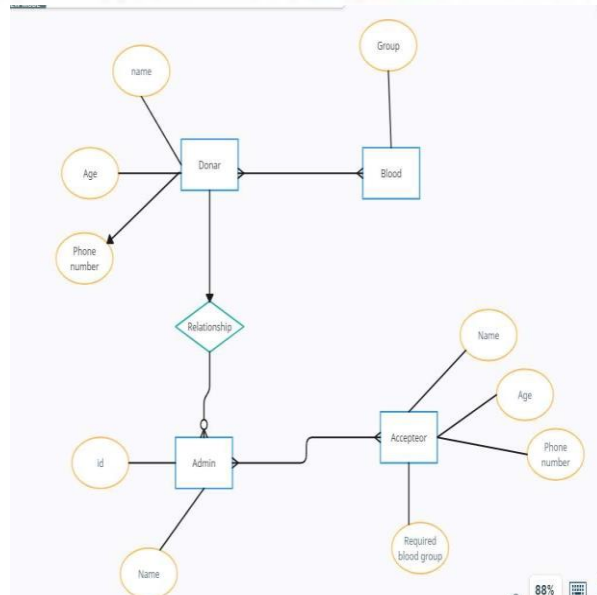
ER- diagram

Creating an Entity-Relationship Diagram (ERD) for a blood donation system involves identifying the main entities, their attributes, and the relationships between them. Below is a simplified ERD for a blood donation system:

Entities:

- Donor
 - o Id (primary key)
 - o Name
 - o Age
 - o Phone number
 - o Gender
 - o Address
 - o Blood group
- Admin Id (Primary Key)

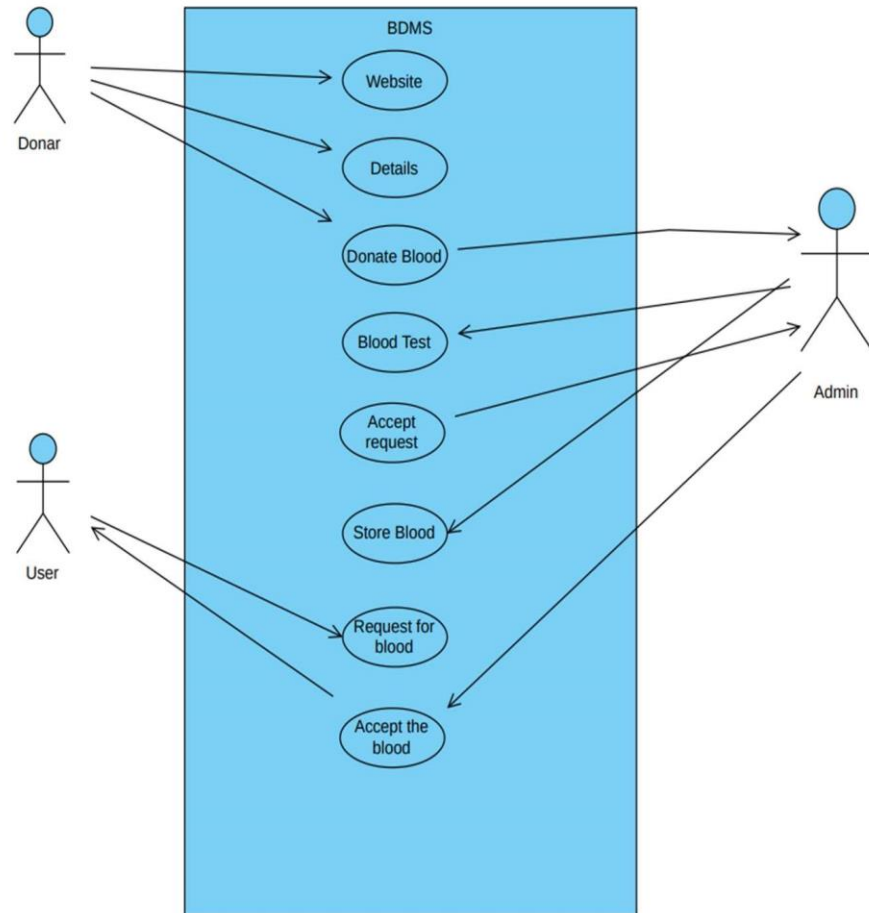
- Name
- Acceptor
 - Name
 - Age
 - Phone number
 - Requested blood group



Use Case diagram

A use case diagram at its simplest is a representation of a user's interaction with the system. Cases included in following use-case diagram:

- website
- Details
- Donate blood
- Blood test
- Accept request
- Store blood
- Request for blood



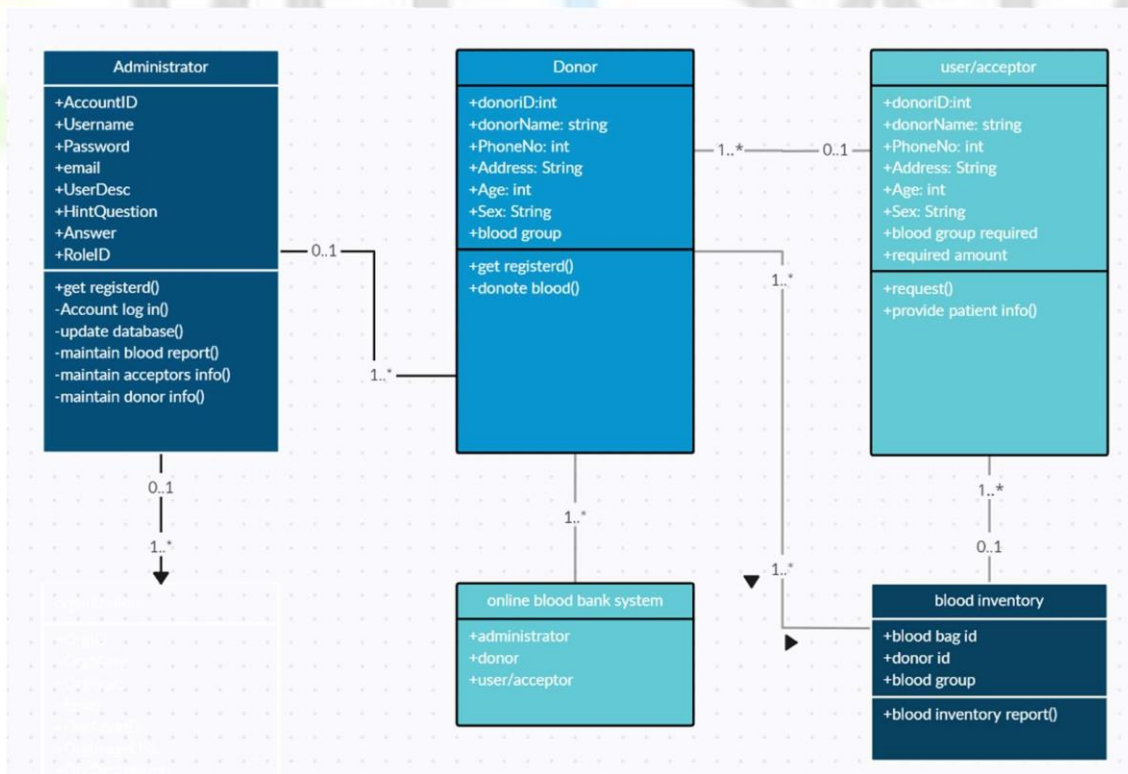
□ Accept the blood

Class Diagram

A class diagram in the UML is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

Classes included in following diagram:

- Administrator
- Donor
- Online blood bank system
- User/ acceptor
- Blood inventory



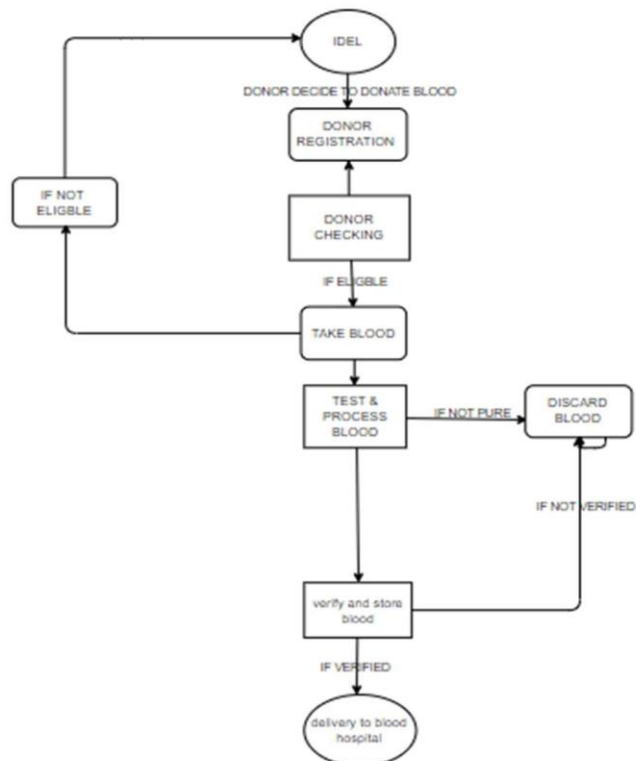
State diagram

A sequence diagram is an interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart.

A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur.

This allows the specification of simple runtime scenarios in a graphical manner.

- Donar registration
- Admin check all details of donor
- Donar donates blood
- Admin acknowledgement donor
- User requested with admin for blood
- Admin checks if requested blood is there or not
- If there admin accepts acceptor's request.



Sequence diagram

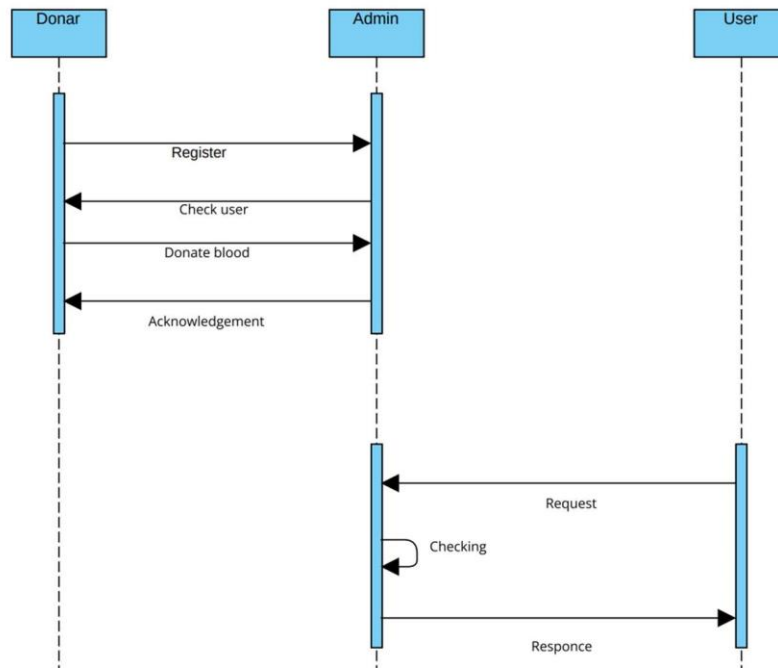
A sequence diagram is an interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart.

A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur.

This allows the specification of simple runtime scenarios in a graphical manner.

Donar registration

- Admin check all details of donor
- Donar donates blood
- Admin acknowledgement donor
- User requested with admin for blood
- Admin checks if requested blood is there or not
- If there admin accepts acceptor's request.



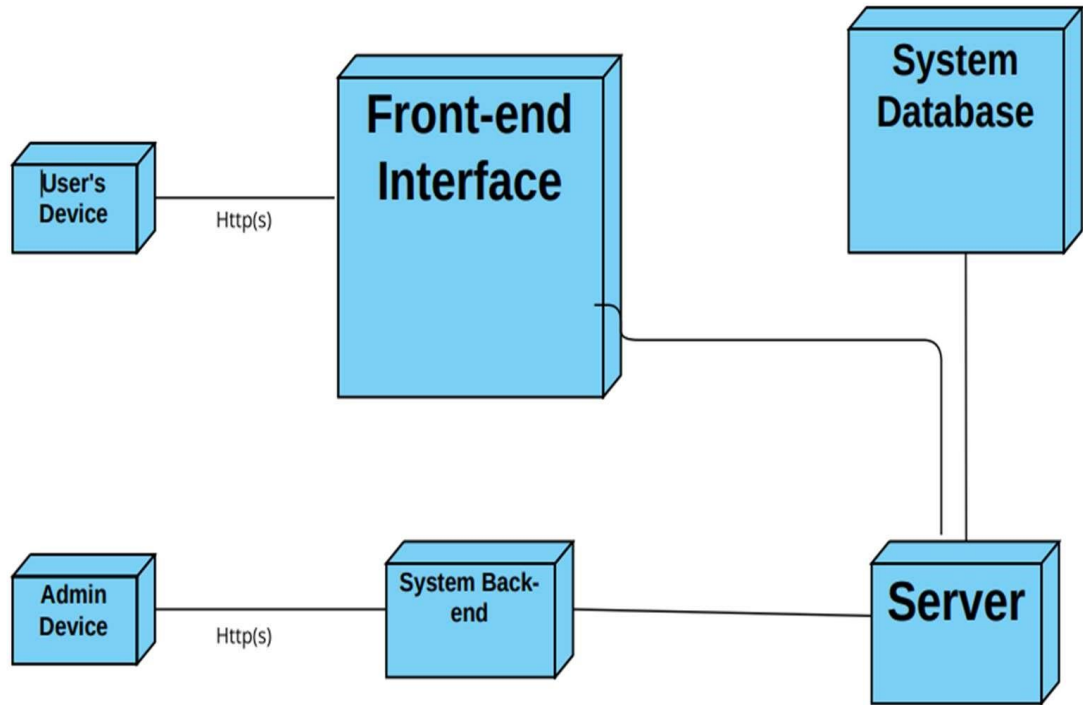
Deployment Diagram

A deployment diagram in the context of a blood donation management system illustrates how the various software components, hardware devices, and communication channels are deployed and interconnected in a real-world environment. It provides a high-level view of the system's physical architecture. Below is a simplified deployment diagram for a blood donation management system:

Key Components and Deployment Nodes:

- User Device connects with Front-end Interface of website
- Front-end Interface connects with server
- Server do the process in Back-end

User data will store in Database



- Admin device will verify the user details

□

Conclusion

The Blood Bank Management System is a web-based application that manages the process of blood donation from registration to distribution. The system uses a Database Management System (DBMS) technology to store and manage donor information, blood types, and inventory records. The system provides an easy-to-use interface for managing blood donations, scheduling appointments, and monitoring inventory levels.

The Blood Bank Management System is designed to ensure that the blood donation process is efficient, safe, and meets the needs of hospitals and patients. The system enables blood banks to manage donor information and blood inventory records efficiently, and it enables hospitals to request and receive blood donations quickly. The system provides several reports that can be generated based on donor information, blood inventory levels, providing valuable insights into the blood donation process and enabling continuous improvement.

Overall, The Blood Bank Management System is an essential tool for managing the blood donation process and improving the efficiency and effectiveness of blood banks and hospitals.

The system can be improved by adding new features like integration with other systems, mobile application, analytics and reporting, online blood bank, and integration with IoT devices. These improvements can help in the efficient utilization of blood samples, reduce wastage, and ultimately save more lives.