

LIFELINK

PROJECT THESIS

SUBMITTED

TO

AWH ENGINEERING COLLEGE

KUTTIKATTOOR, KOZHIKODE

IN PARTIAL FULFILMENT

OF THE REQUIREMENTS FOR THE AWARD OF THE

DEGREE

OF

Master Of Computer Applications

BY

SANIGA M K

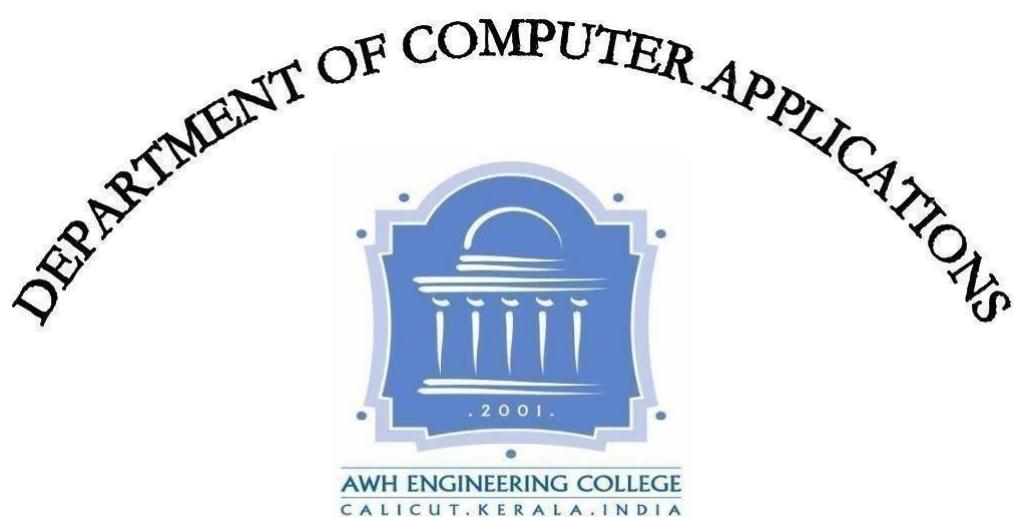


DEPARTMENT OF COMPUTER APPLICATIONS

AWH ENGINEERING COLLEGE KUTTIKKATTOOR,

KOZIKODE

MAY2024



AWH ENGINEERING COLLEGE KOZHIKODE

CERTIFICATE

This is to certify that this thesis entitled “LifeLink” submitted herewith is an authentic record of the thesis work done by SANIGA MK (AWH22MCA-2034) under our guidance in partial fulfillment of the requirements for the award of Master of Computer Applications from APJ Abdul Kalam Technological University during the academic year 2024.

Mrs.Sruti Sudevan

Assistant Professor

Dept. of Computer Applications

Head of the department

Mrs.Sruti Sudevan

Assistant Professor

Dept. of Computer Applications

Project guide

External Examiner

Internal Examiner

ACKNOWLEDGEMENT

I express my sincere gratitude to our beloved principal **Dr.Sabeena M V** for providing me an opportunity with the required facilities for doing this project. I express my hearty thanks to **Mrs.Sruti Sudevan**, Head of the department of Computer Applications for her guidance. I am thankful to all other staff of the MCA department for their encouragement, timely guidance, valuable suggestions and inspiring ideas given throughout this project. I am grateful to my friends for the way they have cooperated, expected me to achieve success and have always stirred my ambition to do the best. Above all, I am grateful to the almighty, who has showered His blessings on me throughout my life and throughout the project.

SANIGA M K

ABSTRACT

This “LifeLink” is a robust platform designed to enhance the efficiency of blood and organ donation processes by establishing seamless collaboration among hospitals, colleges, and blood and organ donors, users. Lifelink is a complete solution that helps make donation processes smoother and improves healthcare results. By facilitating transparent and efficient communication channels, Lifelink empowers hospitals to manage blood and organ donation requests effortlessly, thereby optimizing the supply chain and ensuring timely access to critical resources. Blood donors use Lifelink's easy-to-understand website to handle donation requests they receive. This makes it easier to communicate and speed up the donation process. Likewise, organ donors can quickly sign up, check, and agree to donation requests from hospitals, making organ transplants happen faster. Lifelink helps colleges and hospitals talk easily, so colleges can quickly ask for blood donations. The web application aims to overcome challenges, encourage community involvement, and promote a culture of giving within the healthcare ecosystem. This collaborative network create a more accessible and responsive donation framework, ultimately contributing to improved healthcare services and community well-being.

CONTENTS

	Page No
1. INTRODUCTION	1
2. SYSTEM ANALYSIS	3
2.1 Existing System	4
2.2 Proposed System	4
2.3 Module Description	5
2.4 Sprint	7
2.5 User Stories	11
3. FEASIBILITY STUDY	12
3.1 Economical Feasibility	13
3.2 Technical Feasibility	13
3.3 Operational Feasibility	13
3.4 Behavioural Feasibility	14
3.5 Software Feasibility	14
3.6 Hardware Feasibility	14
4. SOFTWARE ENGINEERING PARADIGM	15
4.1 Agile Model	16
4.2 Scrum	17
5. SYSTEM REQUIREMENT SPECIFICATIONS	20
5.1 Software Requirements	21
5.2 Hardware Requirements	21
6. SYSTEM DESIGN	22
6.1 Mongo DB	23
6.2 Collection	23
6.3 UML Design	25

6.4 Use Case Diagram	26
6.5 Scenario	28
6.6 Sequential Diagram	30
7. SYSTEM DEVELOPMENT	37
7.1 Coding	38
7.2 Node JS	38
7.3 Express JS	39
7.4 Mongo DB	39
8. SYSTEM TESTING AND IMPLEMENTATION	40
8.1 Types of Testing	41
9. SYSTEM MAINTENANCE	44
10. FUTURE ENHANCEMENT	46
11. CONCLUSION	48
12. APPENDIX	50
13. BIBLIYOGRAPHY	79

INTRODUCTION

1.INTRODUCTION

Introducing "LifeLink" a smart system that makes donating blood and organs easier and faster. It helps hospitals, colleges, and blood and organ donors work together smoothly and ensuring seamless collaboration in meeting the critical need for blood and organ donations.. The web system is all about solving donation challenges, getting communities involved, and spreading a culture of giving in healthcare. With LifeLink, blood donors and organ donors can quickly register and donate blood and organ, ensuring no delay in life-saving contributions. Users experience a user-friendly interface, allowing them to request blood promptly without any hassle or time lag, ensuring efficient and timely responses to critical medical needs.

SYSTEM ANALYSIS

2.SYSTEM ANALYSIS

2.1 Existing system

The existing system may lack the seamless collaboration and bidirectional communication proposed by LifeLink. Current processes for blood and organ donation may be less efficient, with potential challenges in information management and accessibility. The user interfaces for hospitals, colleges, and blood and organ donors might not be as user-friendly, and there may be gaps in ensuring the transparent protection of sensitive information. The collaborative network introduced by LifeLink aims to address these limitations, creating a more accessible and responsive donation framework to contribute to improved healthcare services and community well-being.

The existing system has several disadvantages:

- Lack of seamless collaboration and bidirectional communication.
- Non-user-friendly interfaces for hospitals, colleges, and donors.

2.2 Proposed system

The "LifeLink" platform proposes a robust system to enhance blood and organ donation processes by facilitating seamless collaboration among hospitals, colleges, and donors. Key features include bidirectional communication between hospitals and colleges for blood donation, user-friendly interfaces for all stakeholders, and a transparent system to protect sensitive information. The project aims to overcome challenges, encourage community involvement, and foster a culture of giving within the healthcare ecosystem.

The proposed system has several advantages:

- Enables bidirectional communication between hospitals to hospital and colleges, blood donors for blood donation.
- Provides user-friendly interfaces for all stakeholders involved
- Overcomes challenges in the organ and blood donation process.

2.3 Module Description

This project has 6 modules:

Admin:

- Login
- View & manage hospital
- View & manage users
- View & manage blood donors
- View & manage organ donors
- Add categories
- View donations
- View & manage college

Hospital:

- Register
- Login
- View & manage blood donors request
- View organ donors registered under hospital and send request
- View organ request from hospital & send request to organ donors
- Send organ request to hospital
- Manage blood request from hospital
- Send blood request
- Manage blood request from user
- Manage college request
- Send blood request to college
- View blood donation history
- View organ donation history
- Profile edit

College:

- Register
 - Login
 - Profile edit
 - View and manage hospital request
-

- Send request to hospital
- View history

Blood donors:

- Register
- Login
- Profile edit
- View & manage user request
- Send request to hospital for blood donation
- View history

Organ donors:

- Register
- Login using nominee credentials and organ donor credentials
- View & accept hospital organ donation request
- Update donor details

User:

- Register
- Login
- Send blood request
- View history
- Profile edit

2.4 Sprint

Sprint 1

Module	Task	Hours for completion	Expected date of completion	Actual date of completion	Reason for Deviation
Admin/Hospital/Organ donor/User/college/blood donor	Login	6 hours	29/01/2024	29/01/2024	-
Admin	Manage hospital	6 hours	30/01/2024	30/01/2024	-
	Manage blood donor	6 hours	31/01/2024	31/01/2024	-
	Manage organ donor	7 hours	01/02/2024	01/02/2024	-
	Manage user	6 hours	02/02/2024	02/02/2024	-
	Manage college	6 hours	03/02/2024	03/02/2024	-
	View hospital blood donation	5 hours	05/02/2024	05/02/2024	-
	View hospital organ donation	6 hours	06/02/2024	06/02/2024	
	View blood donor donation	6 hours	07/02/2024	07/02/2024	-
	View college donation	6 hours	08/02/2024	08/02/2024	-

	Add categories	6 hours	09/02/2024	09/02/2024	-
--	----------------	---------	------------	------------	---

Sprint 2

Module	Task	Hours for completion	Expected date of completion	Actual date of completion	Reason for Deviation
Blood donor	Send request for donation	5 hours	12/02/2024	14/02/2024	
	View hospitals	5 hours	13/02/2024	15/02/2024	
	View send request	5 hours	14/02/2024	16/02/2024	
	View user request	6 hours	15/02/2024	17/02/2024	
	View accepted request	6 hours	16/02/2024	19/02/2024	
	Registration	5 hours	17/02/2024	20/02/2024	
	Manage profile	6 hours	19/02/2024	21/02/2024	
	Validation	5 hours	20/02/2024	22/02/2024	
	Session	6 hours	21/02/2024	21/02/2034	
Organ donor	View hospital request	5 hours	22/02/2024	23/02/2024	
	Manage profile	5 hours	23/02/2024	26/02/2024	
	Nominee can update profile	5 hours	26/02/2024	27/02/2024	
	Registration	5 hours	27/02/2024	28/02/2024	
	Nominie login	5 hours	28/02/2024	28/02/2024	
	Validation	5 hours	1/03/2024	1/03/2024	

Sprint 3

Module	Task	Hours for completion	Expected date of completion	Actual date of completion	Reason for Deviation
Hospital	Registration	5 hours	02/03/2024	26/02/2024	-
	View and accept blood donors request	5 hours	06/03/2024	27/02/2024	-
	View organ donors and send request	5 hours	11/03/2024	28/02/2024	-
	View hospital blood request	6 hours	13/03/2024	1/03/2024	-
	View hospital organ request	6 hours	15/03/2024	2/03/2024	-
	Send blood request	5 hours	16/03/2024	6/03/2024	-
	Send organ request	5 hours	18/03/2024	11/03/2024	-
	Send blood request to college	3 hours	20/03/2024	13/03/2024	-
	View user request	3 hours	22/03/2024	15/03/2024	-
	View college request	5 hours	23/03/2024	16/03/2024	-
	View donor history	5 hours	25/03/2024	18/03/2024	-
	View Hospital blood history	6 hours	26/03/2024	20/03/2024	-
	View college donation history	6 hours	27/3/2024	22/3/2024	
	View send hospital organ request	5 hours	28/3/2024	23/3/2024	

	View send request to organ donors	6 hours	29/3/2024	25/3/2024	
	Update profile	6 hours	30/3/2024	26/3/2024	
	Validation	5 house	01/4/2024	27/3/2024	

Sprint 4

Module	Task	Hours for completion	Expected date of completion	Actual date of completion	Reason for Deviation
College	View hospital request	5 hours	02/04/2024	14/02/2024	
	Send request to hospital	5 hours	03/04/2024	15/02/2024	
	View hospitals	5 hours	05/04/2024	16/02/2024	
	Manage profile	6 hours	06/04/2024	17/02/2024	
	View history	6 hours	08/04/2024	19/02/2024	
	Registration	5 hours	09/04/2024	20/02/2024	
	Validation	6 hours	11/04/2024	21/02/2024	
	Session	5 hours	12/04/2024	22/02/2024	
User	Send request for blood	5 hours	15/04/2024	23/02/2024	
	View history	5 hours	19/04/2024	26/02/2024	
	Update the profile	5 hours	20/04/2024	27/02/2024	
	Registration	5 hours	28/02/2024	28/02/2024	
	Validation	5 hours	01/03/2024	01/03/2024	

2.5 User Stories

LIFELINK is a web application which consist of 6 modules as Admin, Blood donor, Organ donor, Hospital, College and User. Admin will be responsible for managing the Blood donor , Organ donor, Hospital, College and User. Admin will be able to view the all blood and organ donations of college , blood donors and hospital, Add the category of organ after death and before death

Blood donors should be able to register and login to send blood donation request to hospital, Blood donors will be able to view and manage the users request for blood, view the blood donation history. Blood donors will also able to view the profile and update it.

Organ donors should be able to register and login to view and accept the hospital request and organ donor can update the profile also the nominie can login and update the profile.

Hospital should be able to register and login, can view and manage the blood donors request for blood donation with blood donor preference. Hospital will be able to add organ donors and update the organ donors, manage request for blood and organ donation of other hospitals, assign organ donor to the organ request, They will be able to send request for organ and blood to hospital, request for blood to college. They manage request from college for blood donation and also manage request from the user for blood. They can view the blood donation history of blood donors, view the send and received blood donation history of hospitals and colleges and view the send and received organ donation history. Hospital will also able to view the profile and update it.

Colleges should be able to register and login to send a request to hospital for blood donation, they can manage the request from the hospitals for blood, view the history of requests. College will also able to view the profile and update it.

User should be able to register and login to send a request for blood ,they view the donation history. User will be able to view the profile and update it.

FEASIBILITY STUDY

3.FEASIBILITY STUDY

An analysis of the ability to complete a project successfully, taking into account legal, economic, technological, scheduling, and other factors is considered a feasibility study. Rather than just diving into a project and hoping for the best, feasibility study allows project managers to investigate the possible negative and positive outcomes of a project before investing too much money and time.

3.1 Economical Feasibility

The economic analysis is done to determine the benefits and savings that are expected from the candidate system and compare them with costs. Thus, coming to a conclusion on whether the system is economically feasible or not. This system is cost effective as well as time effective, thereby making it economically feasible. This study presents tangible and intangible benefits from the project by comparing the developments and operational costs. The technique of cost benefit analysis is often used as a basis for assessing economic feasibility.

3.2 Technical Feasibility

The technical requirements for the system is economic and it does not use any other additional hardware. This application is developed using MERN stack, whose development kit are easily available and free of cost, thus making our system technically feasible.

3.3 Operational Feasibility

This analysis involves how it will work when it is installed and the assessment of political and managerial environment in which it is implemented. The new proposed system is very much useful to the users and there for it will accept broad audience.

3.4 Behavioural Feasibility

This analysis involves how it will work when it is installed and the assessment of the political and managerial environment in which it is implemented. People are inherently resistant to change and computers have been known to facilitate change. The new proposed system is very much useful to the users and therefore it will accept a broad audience.

3.5 Software Feasibility

Even though this application is developed in a very high software environment, it is also supported by many other environments with minimal changes. The system is fully feasible to be executed on any kind of operating systems and browsers.

3.6 Hardware Feasibility

Software can be developed with the existing resources. But the existing resources may or may not be used to produce hardware. If no hardware is newly bought for a project, then software is said to achieve hardware feasibility. The system is hardware-wise feasible because it needs absolutely no new hardware.

SOFTWARE ENGINEERING PARADIGM

4.SOFTWARE ENGINEERING PARADIGM

The software engineering paradigm which is also referred to as a software process model or Software Development Life Cycle (SDLC) model is the development strategy that encompasses the process, methods and tools. SDLC describes the period of time that starts with the software system being conceptualized.

4.1 Agile model

Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product. Agile Methods break the product into small incremental builds. These builds are provided in iterations. Each iteration typically lasts from about one to three weeks. At the end of the iteration, a working product is displayed to the customer and important stakeholders. Agile Methods break the product into small incremental builds. These builds are provided in iterations. Each iteration typically lasts from about one to three weeks.

At the end of the iteration, a working product is displayed to the customer and important stakeholders. Agile model believes that every project needs to be handled differently and the existing methods need to be tailored to best suit the project requirements. In Agile, the tasks are divided to time boxes (small time frames) to deliver specific features for a release.

Agile software development is an umbrella term for a set of frameworks and practices based on the values and principles expressed in the Manifesto for Agile Software Development and the 12 Principles behind it. When user approach software development in a particular manner, it's generally good to live by these values and principles and use them to help figure out the right things to do given users particular context. One thing that separates Agile from other approaches to software development is the focus on the people doing the work and how they work together. Solutions evolve through collaboration between self-organizing cross-functional teams utilizing the appropriate practices for their context.

In Agile software development, fostering a culture of continuous improvement is paramount. Teams embrace change and adaptability, valuing feedback loops and frequent iterations to deliver high-quality software that meets evolving customer needs. Transparency and open communication are encouraged, allowing for early identification and resolution of issues. By prioritizing individuals and interactions over processes and tools, Agile teams harness the collective expertise and creativity of team members to drive innovation and deliver value effectively. Embracing Agile principles empowers teams to navigate complexity with agility, delivering software that not only meets expectations but also exceeds them, driving success in today's dynamic and competitive landscape.

4.2 Scrum

Scrum is an agile framework for managing knowledge work, with an emphasis on software development. It is designed for teams of three to nine members, who break their work into actions that can be completed within time boxed iterations, called "sprints", no longer than one month and most commonly two weeks, then track progress and re-plan in 15-minute stand-up meetings, called daily scrums.

Scrum is an iterative and incremental framework for managing product development. It defines "a flexible, holistic product development strategy where a development team works as a unit to reach a common goal", challenges assumptions of the "traditional, sequential approach to enables teams to product development, and enables teams to selforganize by encouraging physical co-location or close online collaboration of all team members, as well as daily face-to-face communication among all team members and disciplines involved.

Scrum is a framework that helps teams work together. Much like a rugby team (where it gets its name) training for the big game, scrum encourages teams to learn through experiences, self-organize while working on a problem, and reflect on their wins and losses to continuously improve.

While the scrum is talking about is most frequently used by software development teams, its principles and lessons can be applied to all kinds of teamwork. This is one of the reasons scrum is so popular. Often thought of as an agile project management framework, scrum describes a set of meetings, tools, and roles that work in concert to help teams structure and manage their work.

Scrum is the most common agile framework, and the one most people start with. Agile practices on the other hand, are techniques applied during phases of the software development lifecycle. Planning poker for example, is a collaborative estimation practice designed to encourage team members to share their understanding of what done means. The process is quite fun, and has proven to help foster teamwork and better estimates. Continuous integration (also known as ci) is a common agile engineering practice where code changes are integrated into the main branch frequently. An automated build verifies changes, leading to a reduction in integration debt and a continually shippable main branch. These practices, like all agile practices, carry the agile label, because they are consistent with the principles in the agile manifesto.

In the project management, scrum, sometimes written scrum or scrum, is a framework for developing, delivering, and sustaining products in a complex environment, with an initial emphasis on software development, although it has been used in other fields including research, sales, marketing and advanced technologies. It is designed for teams of ten or fewer members, who break their work into goals that can be completed within time-boxed iterations, called sprints, no longer than one month and most commonly two weeks. The scrum team assess progress in time-boxed daily meetings of 15 minutes or less, called daily scrums (a form of stand-up meeting). At the end of the sprint, the team holds two further meetings: the sprint review which demonstrates the work done to stakeholders to elicit feedback, and sprint retrospective which enables the team to reflect and improve.

A key principle of scrum is the dual recognition that customers will change their minds about what they want or need and that there will be unpredictable challenges-for which a predictive or planned approach is not suited. As such, scrum adopts an evidence based empirical approach accepting that the problem cannot be fully understood or defined up front, and instead focusing on how to maximize the team's ability to deliver quickly, to respond to emerging requirements, and to adapt to evolving technologies and changes in market conditions. Many of the terms used in scrum (e.g., scrum master) are typically written with leading capitals (e.g., scrum master) or as conjoint words written in camel case (e.g., scrum master). To maintain an encyclopaedic tone, however, this article uses normal sentence case for these terms-unless they are recognized marks.

This is occasionally seen written in all -capitals, as scrum. The word is not an acronym, so this is not correct; however, it likely arose due to an early paper by ken schwaber which capitalized scrum in its title. While the trademark on the term scrum itself has been allowed to lapse, so that it is deemed as owned by the wider community rather than an individual, the leading capital is retained-except when used with other words.

SYSTEM REQUIREMENT SPECIFICATION

5.SYSTEM REQUIREMENTS SPECIFICATION

5.1 Software Requirements

One of the most difficult tasks is selecting software, once the system requirement is find out then we have to determine whether a particular software package fits for those system requirements. This section summarizes the application requirement.

- Operating system :Windows 8 or above
- Frontend :HTML,CSS,React JS
- Backend :Node JS,Express JS
- IDE :Visual Studio
- Database :Mongo DB

5.2 Hardware Requirements

The selection of hardware is very important in the existence and proper working of any of the software. When selecting hardware, the size and capacity requirements are also important. The hardware must suit all application developments.

- A device with an internet connection
- Processor :Intel core i3 or above
- RAM :4GB or Above
- HDD :500GB or Above

SYSTEM DESIGN

6.SYSTEM DESIGN

System design is the first in the development phase for many engineered product or system. It may define the process of applying various techniques and principles for the purpose of defining a device, a process or system in sufficient detail to permit its physical realization.

6.1. MongoDB

Database design is the process of producing a detailed data model of a database. This logical data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a data definition language, which can then be used to create a database. The term database design can be used to describe many different parts of the design of an overall database system.

Non-relational model databases, also known as NoSQL databases, are a type of database management system that diverge from the traditional relational model. Instead of relying on tables with predefined schemas and fixed relationships, NoSQL databases use flexible and dynamic data models, such as document-based, key-value, graph, or column-family.

6.2. Collection

In MongoDB, a collection is a grouping of MongoDB documents. It is the equivalent of a table in relational databases. Collections exist within databases and can store multiple documents in a structured format. Each document within a collection can have a unique structure, meaning they don't have to follow a rigid schema like in traditional relational databases. This flexibility allows for dynamic and scalable data storage, ideal for applications with evolving data requirements. Collections in MongoDB can be queried using the powerful MongoDB query language, making it a versatile choice for various data models and application needs.

Project Collection

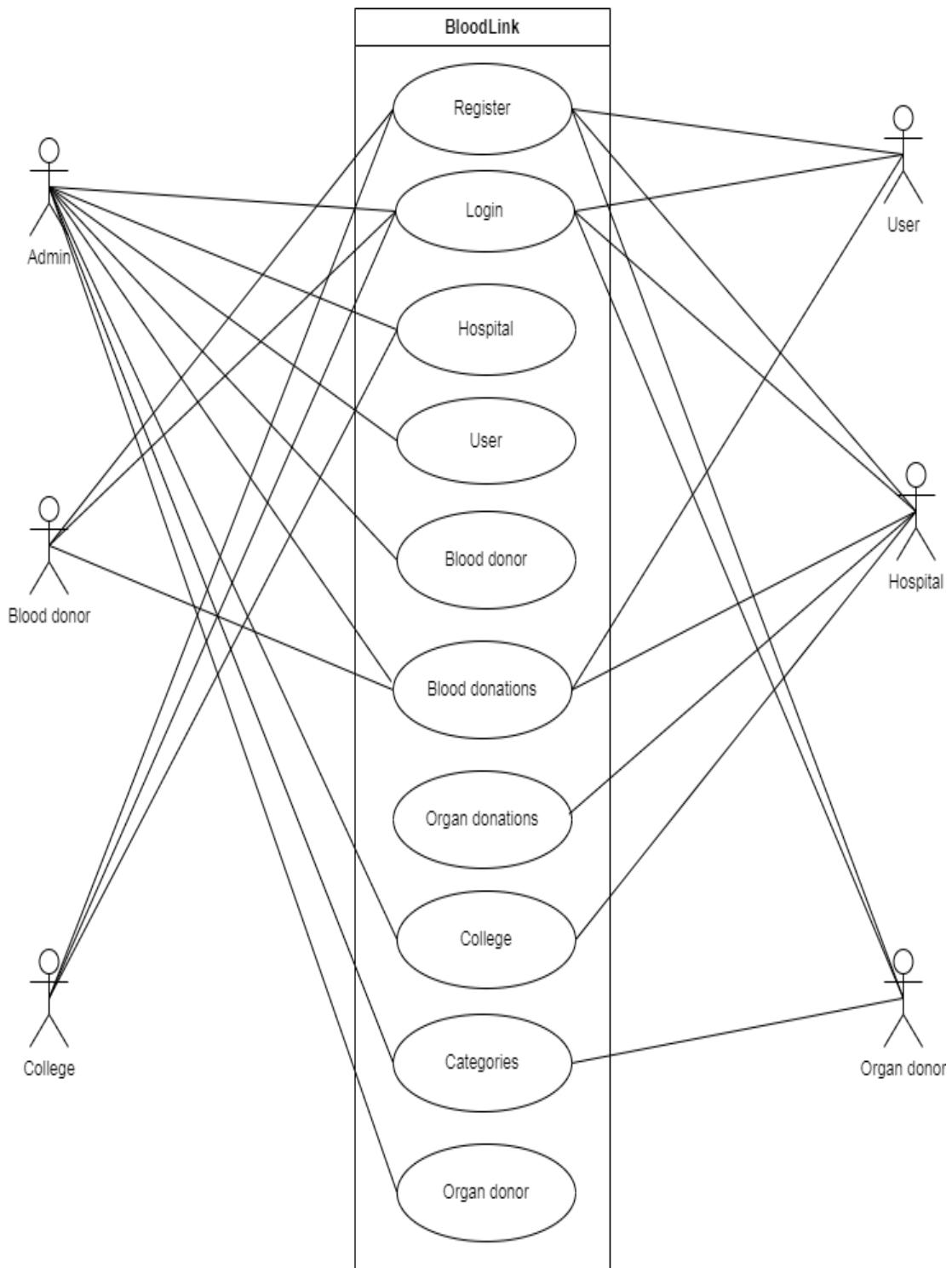
- Add Categories
- College send request
- Blood donor send request
- Hospital send request to organ donors
- User send request
- Hospital Organ request
- Hospital blood request to hospital
- Hospital blood request to college
- Organ donors
- Users

6.3UML Designs

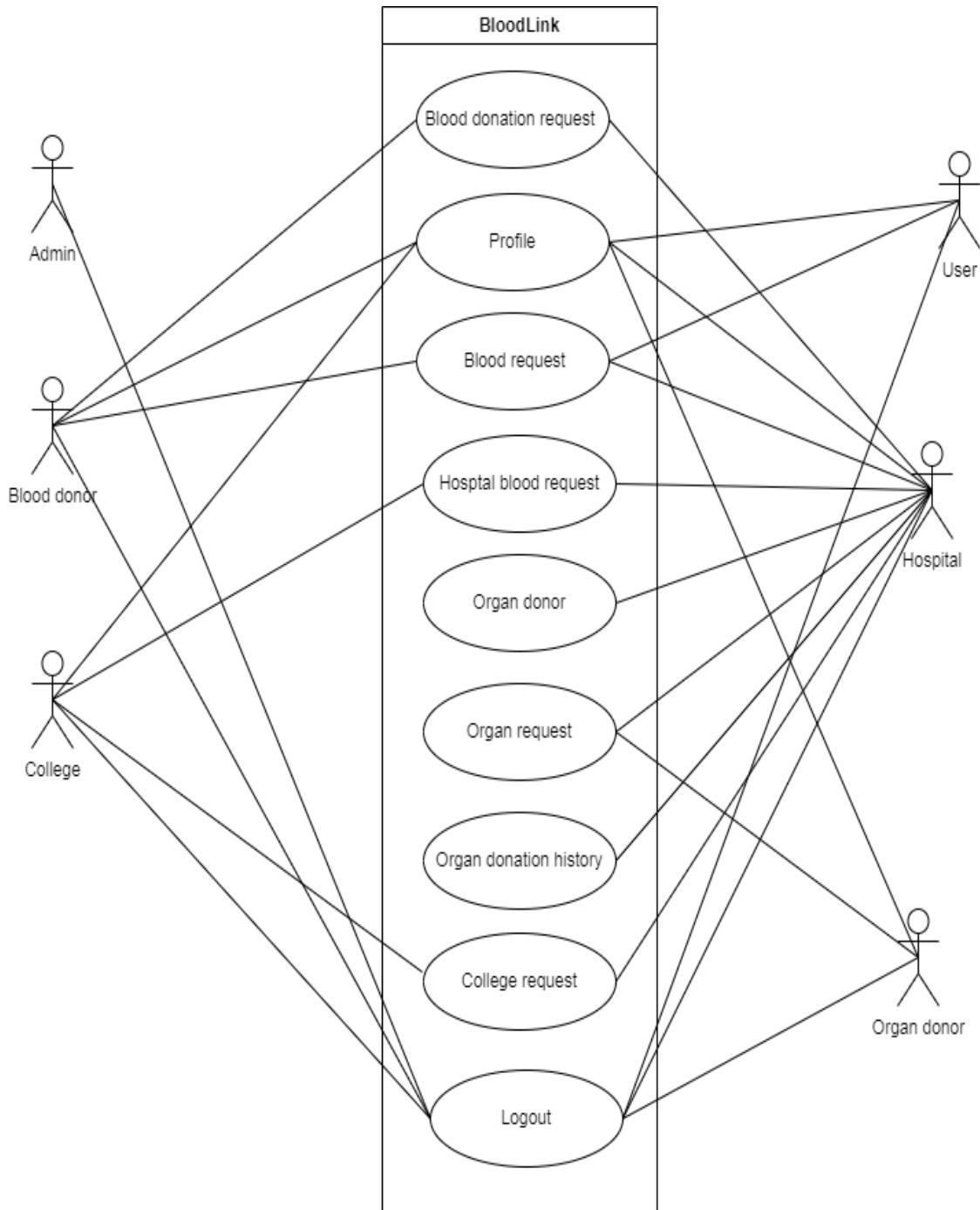
The Unified Modelling Language (UML) is indeed a standardized language used for specifying, visualizing, constructing, and documenting software systems, as well as for business modelling and other non-software systems. It encompasses a collection of best engineering practices that have been proven successful in modelling large and complex systems. UML provides a set of graphical notations that allow software developers and other stakeholders to express and communicate the design of software projects effectively. By using UML, project teams can visualize and explore potential designs, communicate design decisions, and validate the architectural design of the software system. UML diagrams serve as a means to represent various aspects of the system being developed. These diagrams can be used to depict the structure of the system, its behaviour , interactions between components, and the overall flow of activities. UML diagrams serve as a means to represent various aspects of the system being developed. These diagrams can be used to depict the structure of the system, its behaviour , interactions between components, and the overall flow of activities. The graphical nature of UML diagrams makes them intuitive and easier to understand for both technical and non-technical stakeholders involved in the software development process. UML provides a standardized and widely accepted notation, which promotes consistency and clarity in design documentation. This allows for better collaboration among team members and facilitates the understanding and maintenance of software systems over time. The use of UML in software development can enhance communication, facilitate design exploration, and provide a solid foundation for developing and documenting complex software systems.

6.2 Use case diagram

1)



2)



6.3 Scenario

Admin:

- Can Login
- Can manage hospital
- Can manage user
- Can manage blood donor
- Can view all donation
- Can manage college
- Can manage organ donor
- Can add categories

Manager:

- Can register
- Can login
- Can manage blood donors request
- Can view organ donors and send request for organs
- Can manage organ request from hospital
- Can assign organ donor
- Can send organ request to hospital
- Can manage blood request from hospital
- Can send blood request to hospital
- Can manage blood request from user
- Can manage college request
- Can send blood request to college
- Can view blood donation history
- Can view organ donation history
- Can edit profile

College:

- Can register

- Can login
- Can send request to hospital
- Can view & manage hospital request
- Can edit profile
- Can view history

Blood donor:

- Can register
- Can login
- Can send request to hospital for blood donation
- Can view & manage user request
- Can view history
- Can edit profile

User:

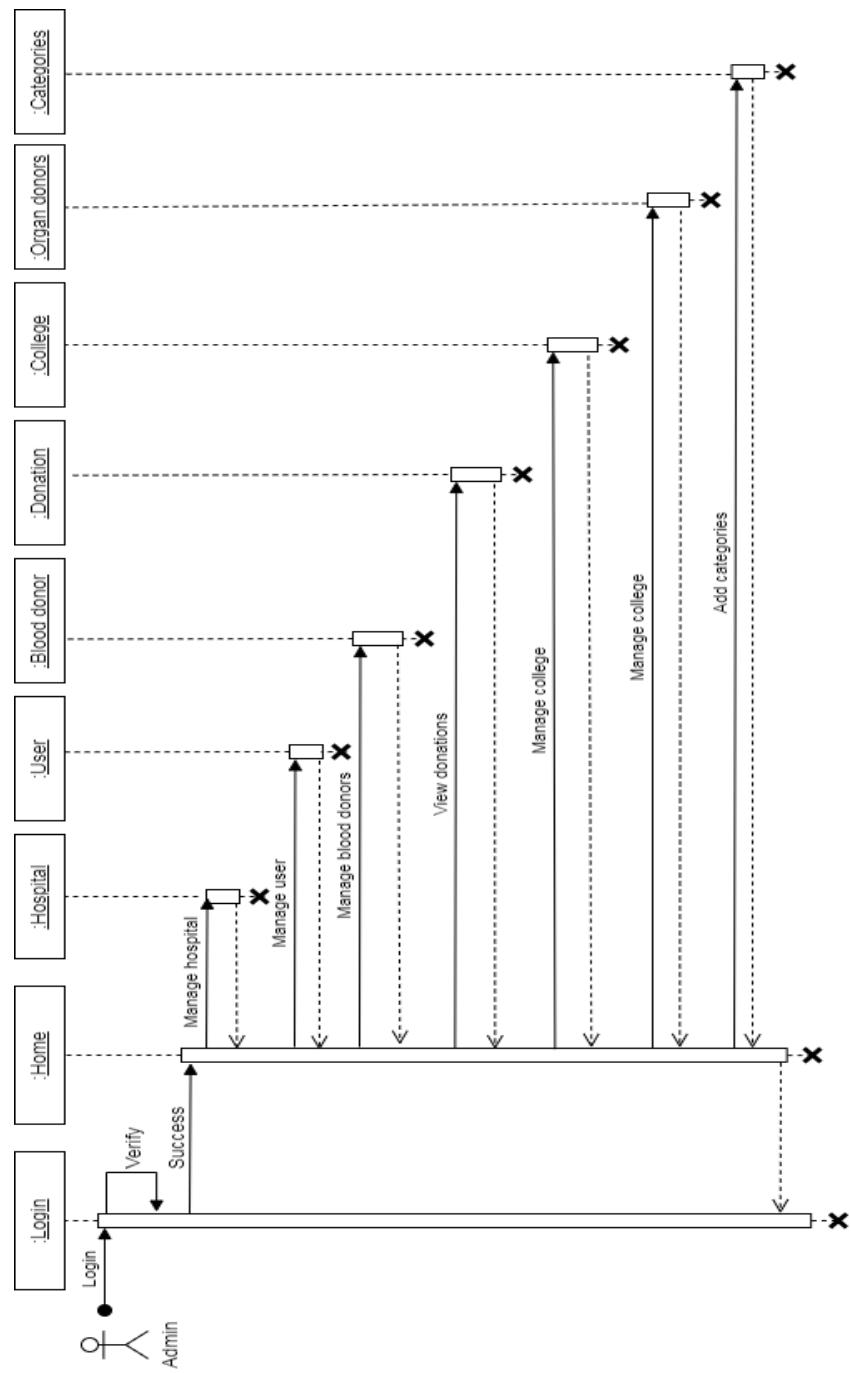
- Can register
- Can login
- Can send request for blood
- View history
- Can edit profile

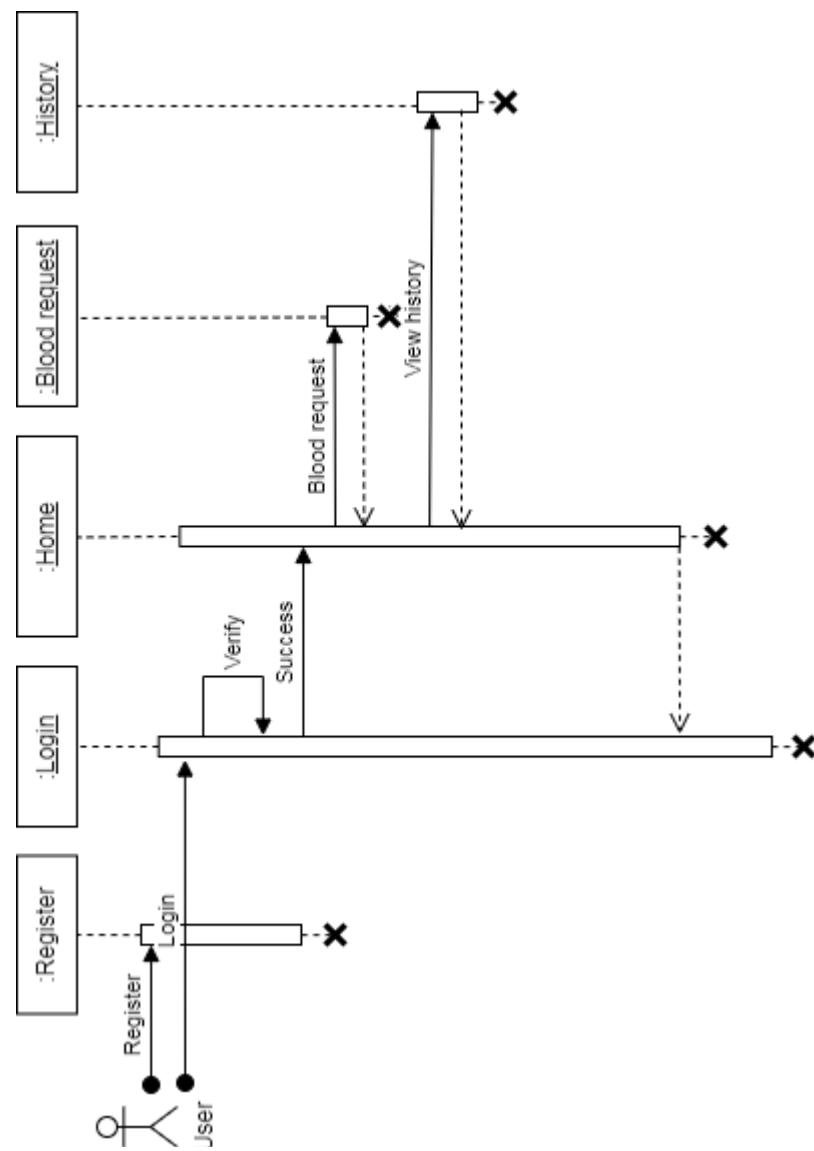
Organ donor:

- Can register
- Can login
- Can view & manage hospital request

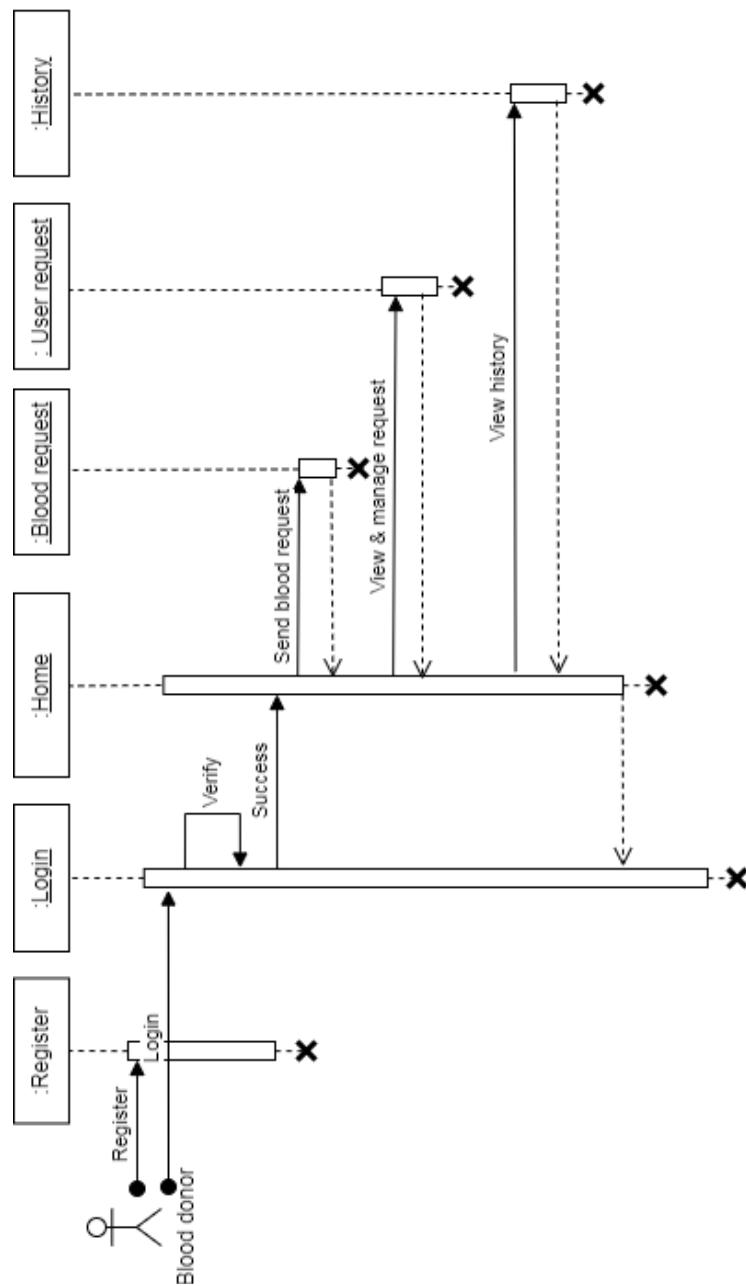
6.4 Sequence Diagram

Admin

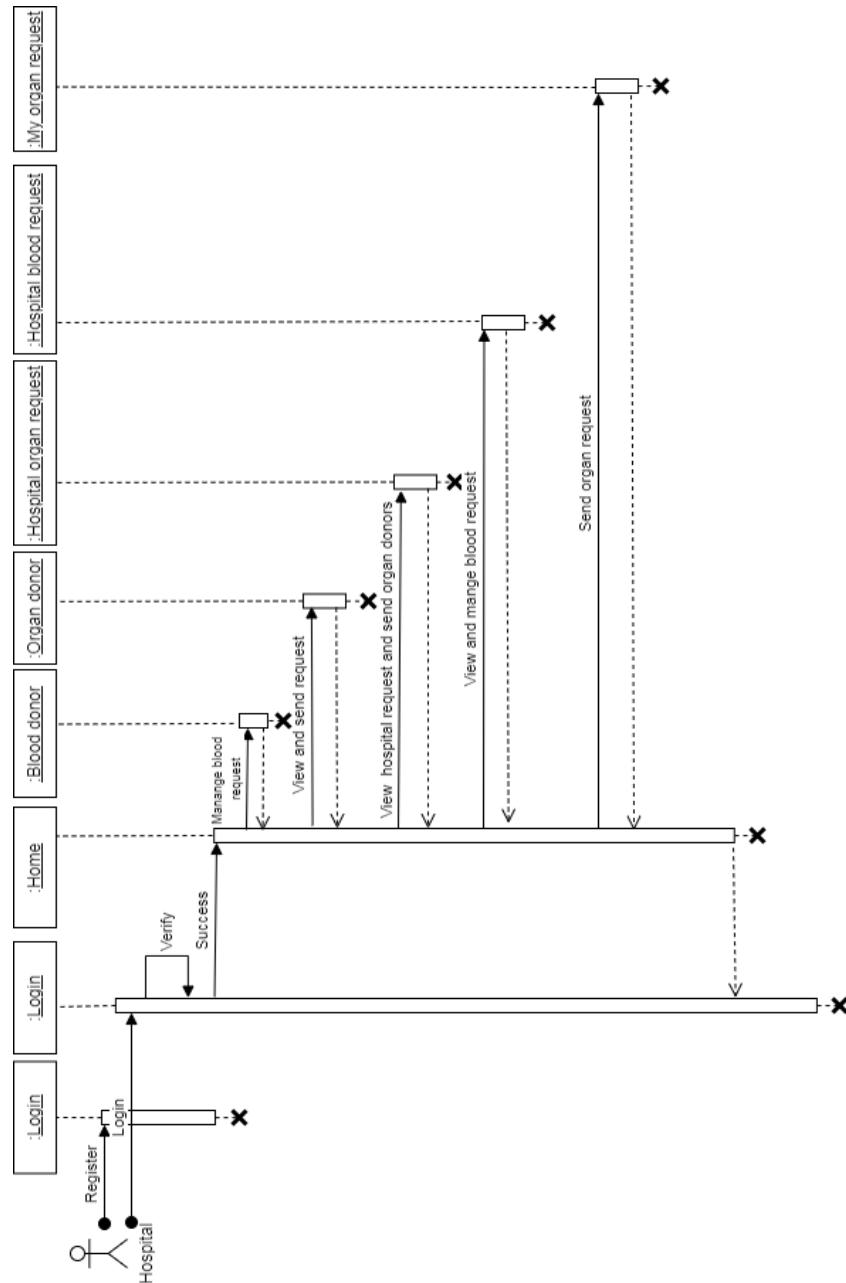


User

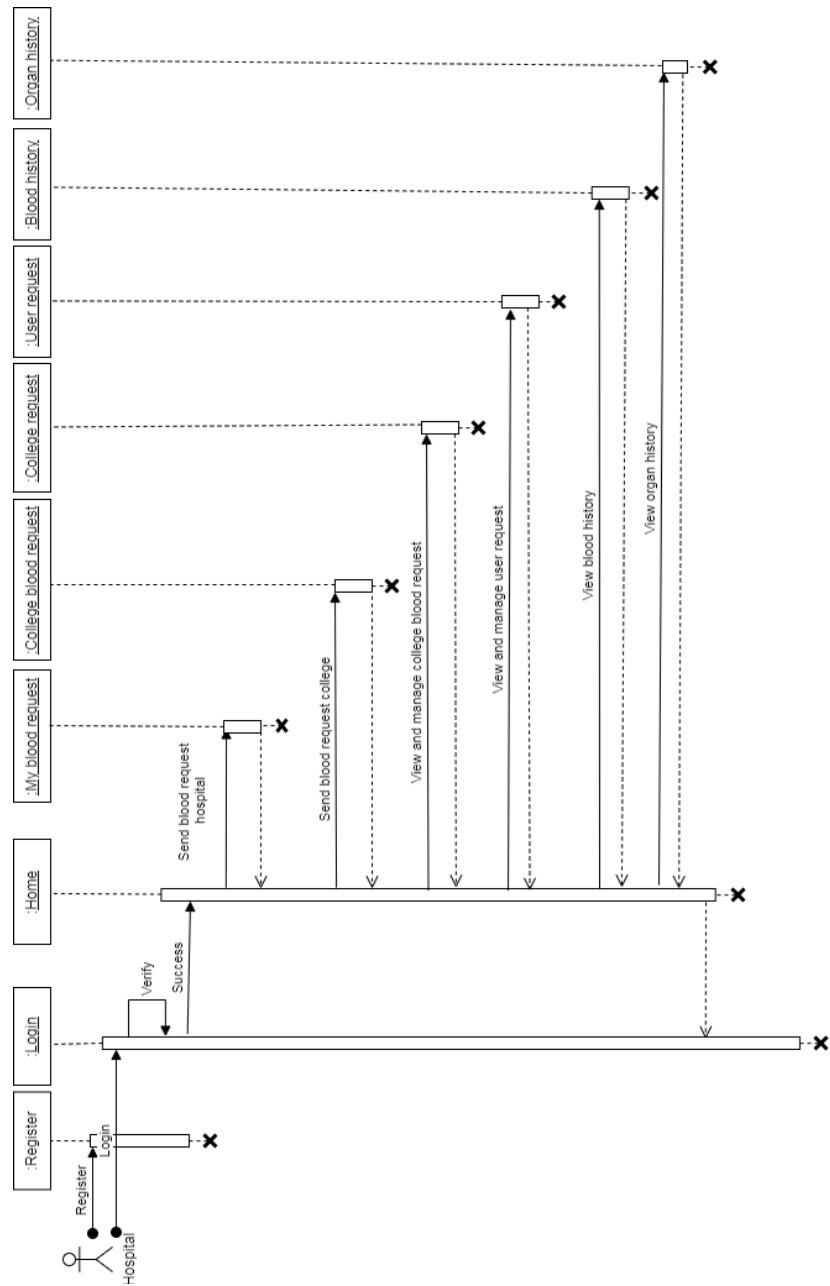
Blood donor



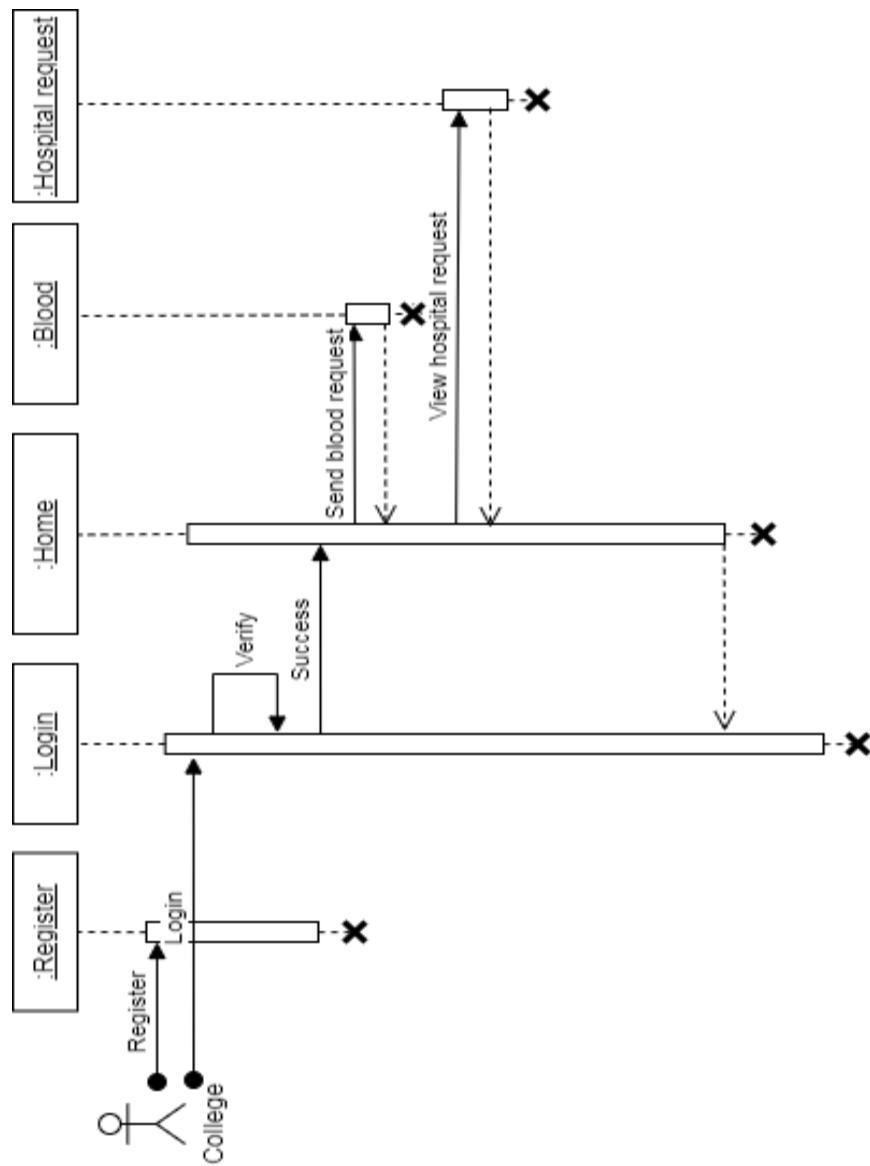
Hospital



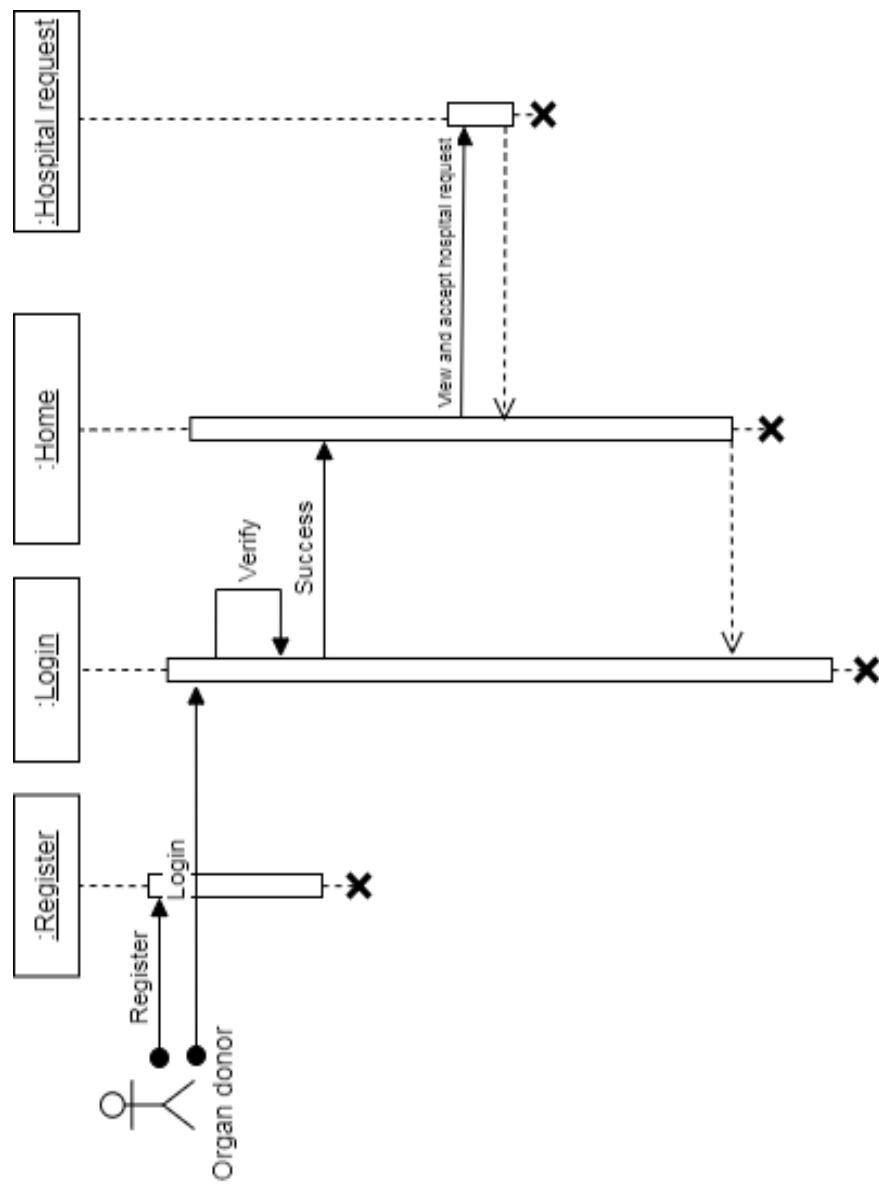
Hospital



College



Organ donor



SYSTEM DEVELOPMENT

7.SYSTEM DEVELOPMENT

System development is series of operations to manipulate data to produce output from computer system. The principal activities performed during the development phase can be divided into two major related sequences.

- External system development
- Internal system development

The major external system activities are:

- Implementation
- Planning
- Equipment acquisition
- Installation

7.1 Coding

The purpose of code is to facilitate the identification and retrieval of items of information. A code is an ordered collection of symbols designed to provide unique identification of an entity or an attribute. Code also shows interrelationship among different items. Codes are used to identify, access, sort, matching records. The code ensures that only one value of code with a single meaning is applied to give an entity or attribute as described in various ways.

Node JS

Node.js is an open-source, cross-platform JavaScript runtime environment that enables developers to build scalable and high-performance applications. It is built on top of the V8 JavaScript engine used by Google Chrome and provides an event-driven, non-blocking I/O model that makes it well-suited for real-time web applications. Node.js enables developers to write server-side applications using JavaScript, which is a popular and widely-used programming language on the web. It has a vast ecosystem of third-party packages and libraries that can be easily installed using the Node Package Manager (NPM). Node.js applications can be run on various platforms such as Windows, Mac, and Linux.

Express JS

Express.js is a minimal and flexible Node.js web application framework that provides a set of robust features for building web and mobile applications. It is one of the most popular and widely-used frameworks for Node.js, and is known for its simplicity and ease of use. Express.js provides a set of features for developing serverside web applications, including routing, middleware support, template engines, and much more. It also provides an easy-to-use API for interacting with databases such as MongoDB and MySQL, and supports a variety of templating engines, such as Pug, Handlebars, and EJS..

Mongo DB

MongoDB is a popular document-oriented NoSQL database system that allows developers to store and manage large amounts of data in a flexible and scalable way. It is an open-source database that uses JSON-like documents with optional schemas, which makes it easy to work with and suitable for a variety of use cases. One of the key benefits of MongoDB is its ability to scale horizontally. This means that developers can add new servers to their database cluster as the amount of data or traffic increases, which allows the database to handle more requests and ensures that it can continue to perform well even as the application grows.

SYSTEM TESTING AND IMPLEMENTATION

8.SYSTEM TESTING AND IMPLEMENTATION

Testing is vital to the success of the system. It makes a logical assumption that if all the parts of the system are correct, the goal will be successfully achieved in this project. It is the stage of implementation, which ensures that the system works accurately and effectively before the live operation commences. It is a confirmation that all are correct and an opportunity to show users that the system must be tested and show that the system will operate successfully and produce expected results under expected conditions. Software testing is a crucial element of software quality assurance and represents the unlimited review of specification, design and coding. Testing represents an interesting anomaly for the software. During the earlier definition and development phase, it was attempted to build the software from an abstract concept to implement.

Testing is a set of activities that can be planned in advance and conducted. Systematically, this is aimed at ensuring that the system works accurately and efficiently before live operations commences

8.1 Types of Testing

Different types of testing are:

- Unit testing
- Black box testing
- Integration testing
- System testing
- White box testing

Unit testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases. All modules were tested individually as soon as they were completed and were checked for their correct functionality. Unit testing deals with testing a unit as a whole. This would test the interaction of many functions but confine the test within one unit. This testing is carried out during the programming stage itself. In this testing step each Module is found to be working

satisfactorily as regard to the expected output from the module.

Black box testing

In black-box testing the structure of the program is not considered. Test cases are decided solely on the basis of the requirements or specifications of the program or module, and the internals of the module or the program are not considered for selection of test cases. In black-box testing, the tester only knows the inputs that can be given to the system and what output the system should give. This form of testing is also called functional or behavioural testing. The most obvious functional testing procedure is exhaustive testing. One criterion for generating test cases is to generate them randomly. There are no formal rules for designing test cases for functional testing.

Integration testing

Integration testing plays a critical role in software development by verifying the interactions and interfaces between different modules or components. This testing phase ensures that the integrated components work together seamlessly as intended, detecting any inconsistencies or communication issues early in the development process. By executing integration tests, software teams can validate the flow of data, control, and interfaces between modules, contributing to the overall stability and functionality of the system.

System testing

System testing is a comprehensive evaluation of the entire software system to confirm that it meets the specified requirements and functions correctly in its intended environment. This testing phase involves testing the system as a whole, including all integrated components, subsystems, and external interfaces. By simulating real-world scenarios and user interactions, system testing verifies the system's behavior, performance, and reliability, ensuring that it delivers the expected outcomes to end-users and stakeholders.

White box testing

White-box testing is a software testing approach that delves deep into the internal structure, logic, and code of a software application. Unlike black-box testing, which focuses solely on evaluating the functionality from an external perspective, white-box testing involves examining the inner workings of the software, including the source

code and system architecture. Testers or developers performing white-box testing have access to the codebase and utilize various techniques such as statement coverage, branch coverage, and path coverage to ensure thorough test coverage. By analyzing the code's control flow, data flow, and execution paths, white-box testing aims to uncover defects, vulnerabilities, and inconsistencies within the software early in the development process. This type of testing not only validates the correctness and integrity of the code but also promotes code quality, maintainability, and efficiency. White-box testing is essential for ensuring robustness, reliability, and security in software applications, as it provides insights into the inner workings of the software and facilitates effective debugging and optimization efforts. Overall, white-box testing plays a vital role in the software development lifecycle by complementing other testing approaches and ensuring the overall quality and reliability of the software product.

SYSTEM MAINTENANCE

9.SYSTEM MAINTENANCE

Maintenance is making adaptation of the software for external changes (requirements changes or enhancements) and internal changes (fixing bugs). When changes are made during the maintenance phase all preceding steps of the model must be revisited.

There are 3 types of maintenance:

- Corrective (Fixing bugs/errors)
- Adaptive (Updates due to environment changes)
- Perfective (Enhancements, requirements changes)

Maintenance is enigma of the system development. The definition of the software maintenance can be given describing four activities that are undertaken after the program is released for use. The maintenance activity occurs since it is unreasonable to assume that software testing will uncover all in a large system. The second activity that contributes the definition of maintenance occurs since rapid changes are encountered in every aspects of computing. The third activity involves recommendation for new capabilities, modification to the existing functions and general enhancements when the software is used. The fourth maintenance activity occurs when software is changed to improve future maintainability or reliability.

FUTURE ENHANCEMENT

10.FUTURE ENHANCEMENT

In the future, LifeLink plans to introduce features such as mobile apps for easier access and notification systems to alert donors about urgent needs. Additionally, there will be efforts to expand the network to include more hospitals, colleges, and potential donors, thus increasing the availability of blood and organs. Furthermore, the platform aims to incorporate advanced data analytics to predict and manage donation trends more effectively, ensuring a steady supply of blood and organs when needed. These enhancements will further streamline the donation process and ultimately improve healthcare services and community well-being.

CONCLUSION

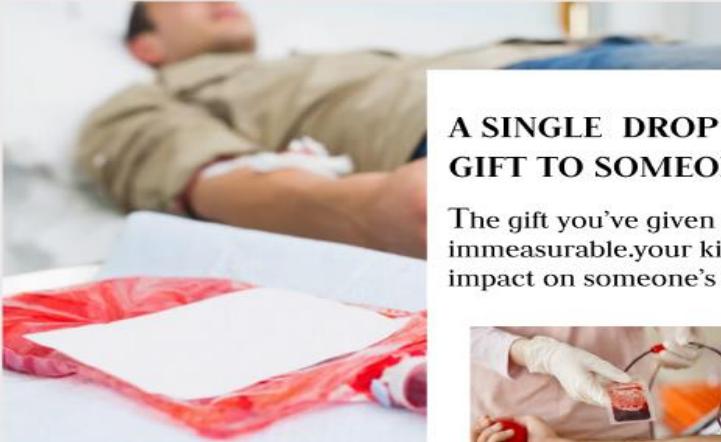
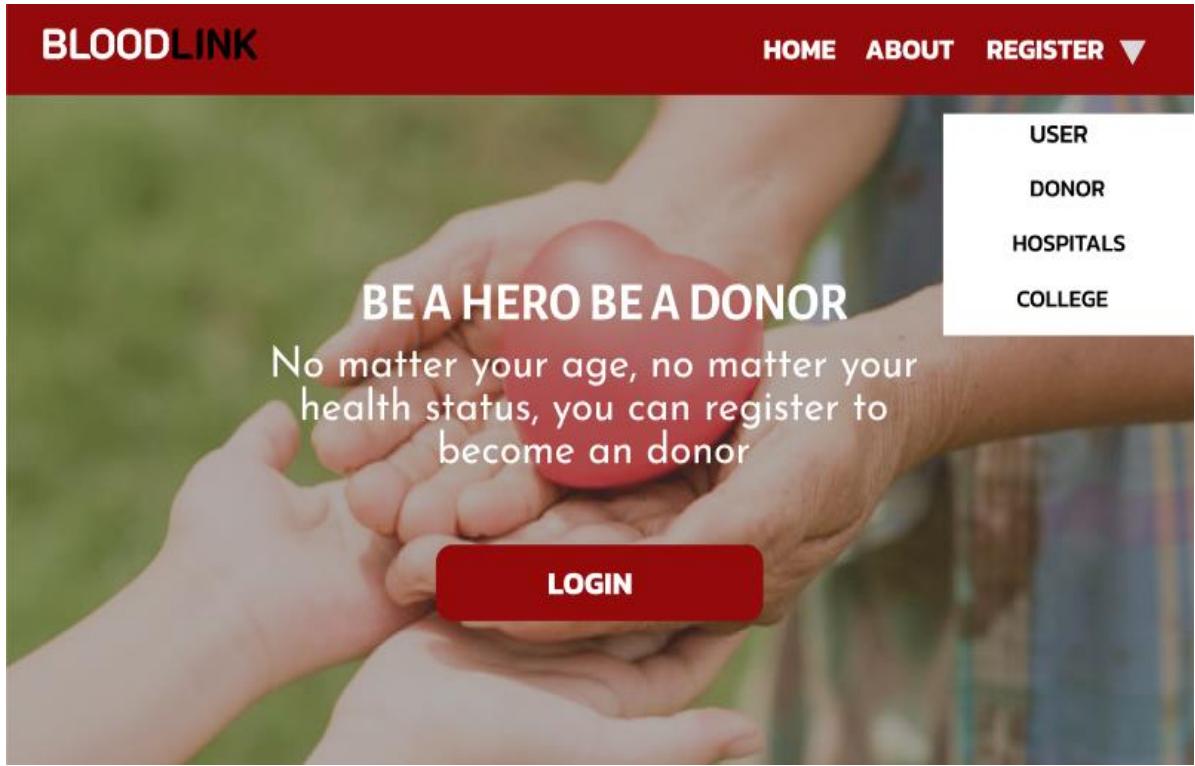
11.CONCLUSION

The proposed LifeLink web application is designed to streamline the blood and organ donation processes by fostering seamless collaboration between hospitals, colleges, and donors. Through its user-friendly interfaces and transparent systems, LifeLink ensures efficient communication and protects sensitive information. By encouraging active participation from communities, the project aims to instill a culture of giving within the healthcare ecosystem. This collaborative network not only enhances accessibility to donations but also ensures a timely response to medical needs, ultimately leading to improved healthcare services and overall community well-being.

APPENDIX

12.APPENDIX

Landing page



A SINGLE DROP OF BLOOD IS A GIFT TO SOMEONE'S LIFE

The gift you've given through your blood donation is immeasurable. Your kindness has made a profound impact on someone's life.



Join with us

Empowering humanity through the gift of life.
Join "BLOODLINK", where every drop counts



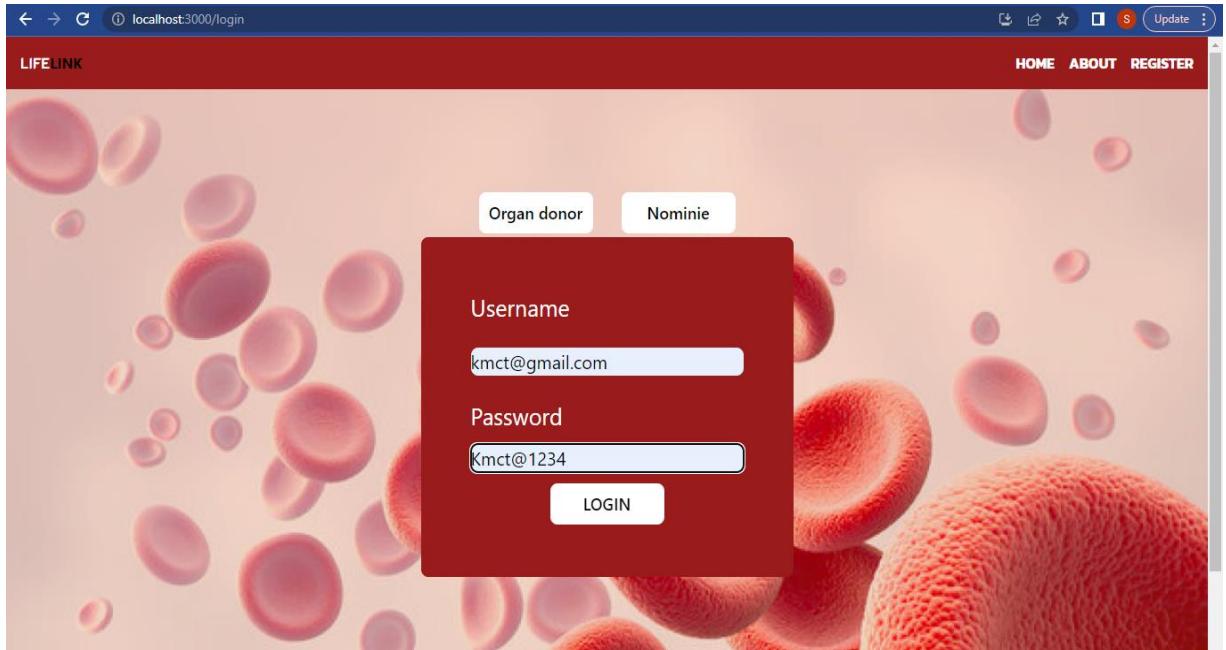
BLOODLINK

✉ bloodlink@gmail.com

📞 +91 6487478378

◎ Kerala, India

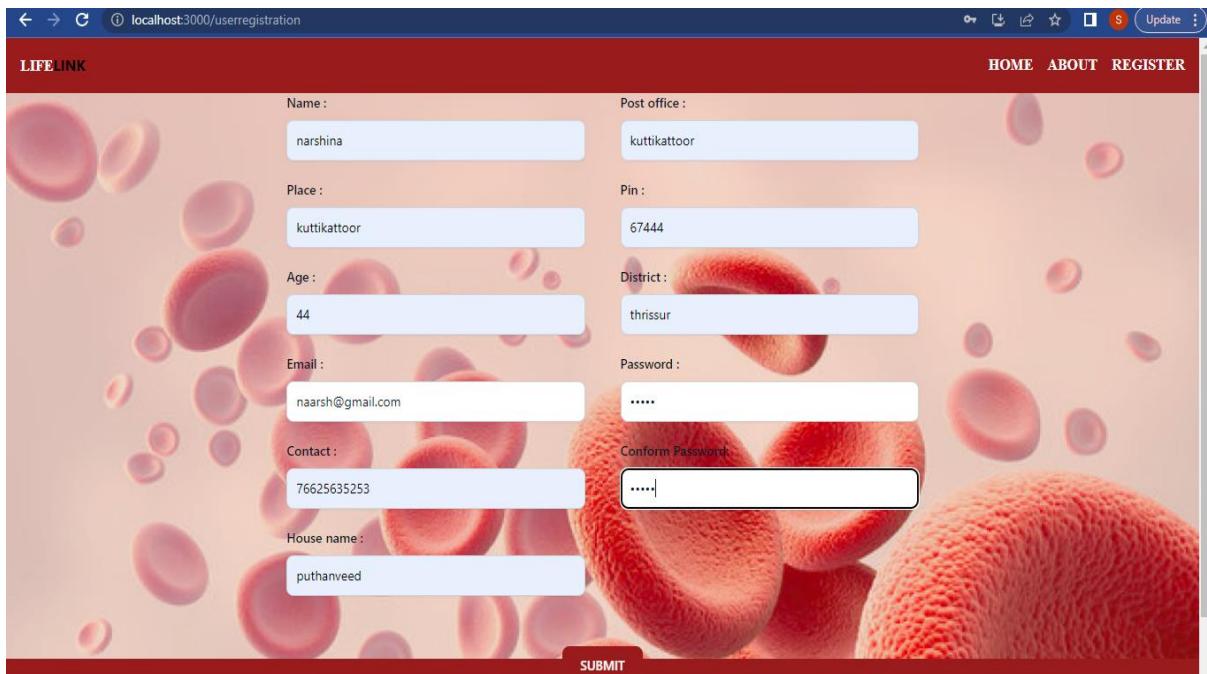
Login page



hospital registration

The screenshot shows a web browser window for 'localhost:3000/hospitalregistration'. The title bar says 'localhost:3000/hospitalregistration'. The header has a red background with the 'LIFELINK' logo on the left and 'HOME ABOUT REGISTER' on the right. Below the header is a background image of red blood cells. The main form consists of several input fields arranged in pairs. From top-left to bottom-right, the fields are: 'Hospital name : medical college', 'Post office : kuttikattoor', 'Place : vadakara', 'Pin : 67444', 'Email : mei@gmail.com', 'District : kannur', 'Contact : 8888833322', 'Password :', 'Liscence number : 7665555', 'Conform Password :', 'Proof : Choose File {D5E9E873-21FF-...84D}.pnglog.png'. Each field is enclosed in a light blue rounded rectangle.

User registration



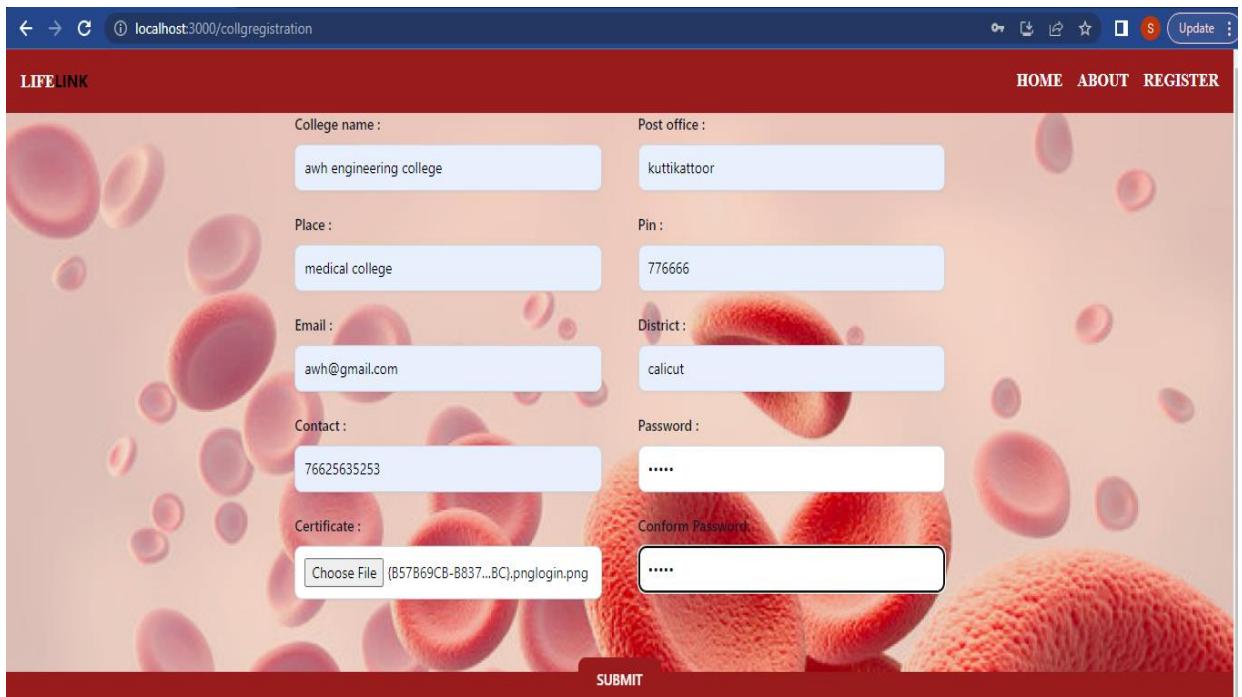
localhost:3000/userregistration

LIFELINK

HOME ABOUT REGISTER

Name :	narshina	Post office :	kuttikattoor
Place :	kuttikattoor	Pin :	67444
Age :	44	District :	thrissur
Email :	naarsh@gmail.com	Password :
Contact :	76625635253	Conform Password:
House name :	puthanveed		
SUBMIT			

College registration



localhost:3000/collgregistration

LIFELINK

HOME ABOUT REGISTER

College name :	awh engineering college	Post office :	kuttikattoor
Place :	medical college	Pin :	776666
Email :	awh@gmail.com	District :	calicut
Contact :	76625635253	Password :
Certificate :	<input type="file" value="Choose File (B57869CB-B837...BC).pnglogin.png"/>	Conform Password:
SUBMIT			

Blood donor registration

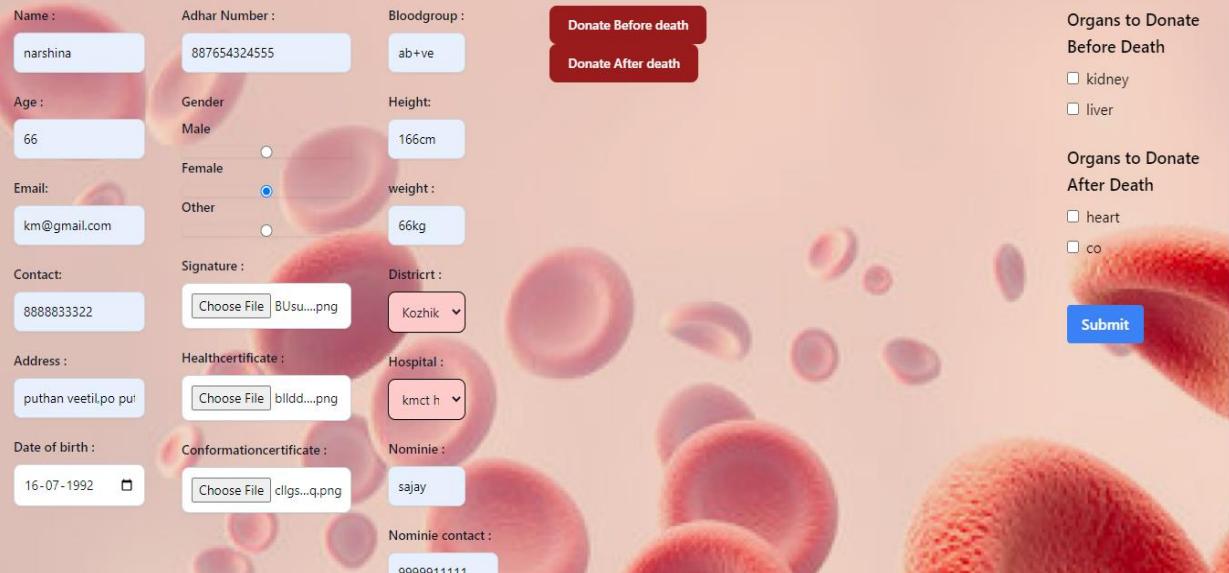


LIFE LINK

HOME ABOUT REGISTER

Name :	narshina	Blood group:	ab+ve
Age :	33	Height :	166cm
Place :	mukkam	Weight :	80
Email :	nashi@gmail.com	Health certificate :	<input type="file" value="Choose File"/>
Contact :	8888833322	Password :
House name :	veetil	Conform Password:
Post office :	mukkam		

Organ donor registration



LIFE LINK

HOME ABOUT REGISTER

Name :	Adhar Number :	Bloodgroup :	<input type="button" value="Donate Before death"/> <input type="button" value="Donate After death"/>
narshina	887654324555	ab+ve	
Age :	Gender	Height:	
66	Male	166cm	
Email:	Female		
km@gmail.com	Other	weight :	
Contact:	Signature :	District :	
8888833322	<input type="file" value="Choose File"/>	Kozhik	
Address :	Healthcertificate :	Hospital :	
puthan veetil.po pu	<input type="file" value="Choose File"/>	kmct h	
Date of birth :	Conformationcertificate :	Nominie :	
16-07-1992	<input type="file" value="Choose File"/>	sajay	
		Nominie contact :	
		9999911111	

Organs to Donate
Before Death

kidney
 liver

Organs to Donate
After Death

heart
 co

Admin manage hospitals

ADMIN **LIFELINK** **HOME** **LOGOUT**

HOSPITAL

HOSPITAL NAME	PLACE	EMAIL	CONTACT	View
mims	mankaav	mims@gmail.com	75443322232	View
fsgsgrs	mankaav	med@gmail.com	8888833322	View
meitra	kunduparamb	mei@gmail.com	76625635253	View
ashoka	vadakara	ash@gmail.com	76625635253	View

localhost:3000/admin/managehospt/6607e8026e7c2a8627679673

HOSPITAL

ashoka	thrissur vadakara
Liscence	View
Liscence no	7665555
Email	ash@gmail.com
Contact	76625635253
Post office	mukkam
Pin	776666

ACCEPT **REJECT**

Admin manage college

The screenshot shows a web browser window for the 'LifeLink' application. The URL is `localhost:3000/admin/managecollege/660e945f646e29bec0e76da4`. The top navigation bar includes links for 'ADMIN', 'LIFELINK', 'HOME', and 'LOGOUT'. On the left, a sidebar lists 'HOSPITAL', 'USER', 'BLOOD DONOR', 'DONATION', and 'COLLEGE'. The main content area displays a user profile for 'fisat' with the following details:

Name	fisat
Address	eranakulam angamali
Email	fis@gmail.com
Contact	8888833322
Certificate	view
Post office	kochi
Pin	776666

At the bottom of the profile card are two buttons: 'ACCEPT' and 'REJECT'.

Admin manage blood donors

The screenshot shows a web browser window for the 'LifeLink' application. The URL is `localhost:3000/admin/blooddonor`. The top navigation bar includes links for 'ADMIN', 'LIFELINK', 'HOME', and 'LOGOUT'. On the left, a sidebar lists 'HOSPITAL', 'USER', 'BLOOD DONOR', 'DONATION', and 'COLLEGE'. The main content area displays a table titled 'BLOOD DONOR' listing the following information:

NAME	PLACE	AGE	EMAIL	CONTACT	ACTION
hg	kuttikattoor	2	aw@gmail.com	76625635253	View
gjg	kjjhg	77	jhgjhgj@kh.com	797879798	View
gjg	kjjhg	77	jhgjhgj@kh.com	797879798	View
gjg	kjjhg	77	jhgjhgj@kh.com	797879798	View
kranjan	kuttikattoor	34	kora@gmail.com	76625635253	View
kalaam	mankaav	66	kala@gmail.com	76625635253	View
rili	mankaav	66	rili@gmail.com	76625635253	View
rili	mankaav	66	rili@gmail.com	76625635253	View
pranav	mankaav	22	pra@gmail.com	76625635253	View

Admin manage user

HOSPITAL
USER
BLOOD DONOR
DONATION
COLLEGE

varna	kannur kuttikattoor
Age	34
Email	vae@gamil.com
Contact	76625635253
House name	veed
Post office	kuttikattoor
Pin	776666

ACCEPT**REJECT**

Admin all view donations

ADMIN**LIFELINK****HOME** **LOGOUT**

HOSPITAL
USER
BLOOD DONOR
DONATION
COLLEGE

Donor**Hospital****College**

NAME	AGE	PLACE	HOSPITAL NAME	DATE
Manu	22	new stand	aster mims	19/7/22
Baby memorial hospital	New stand	Baby@gmail.com	9782676222	31/09/24
Medical college	Black	Accessories	\$99	6/9/20
Google Pixel Phone	Gray	Phone	\$799	22/3/10
Apple Watch 5	Red	Wearables	\$999	1/6/22

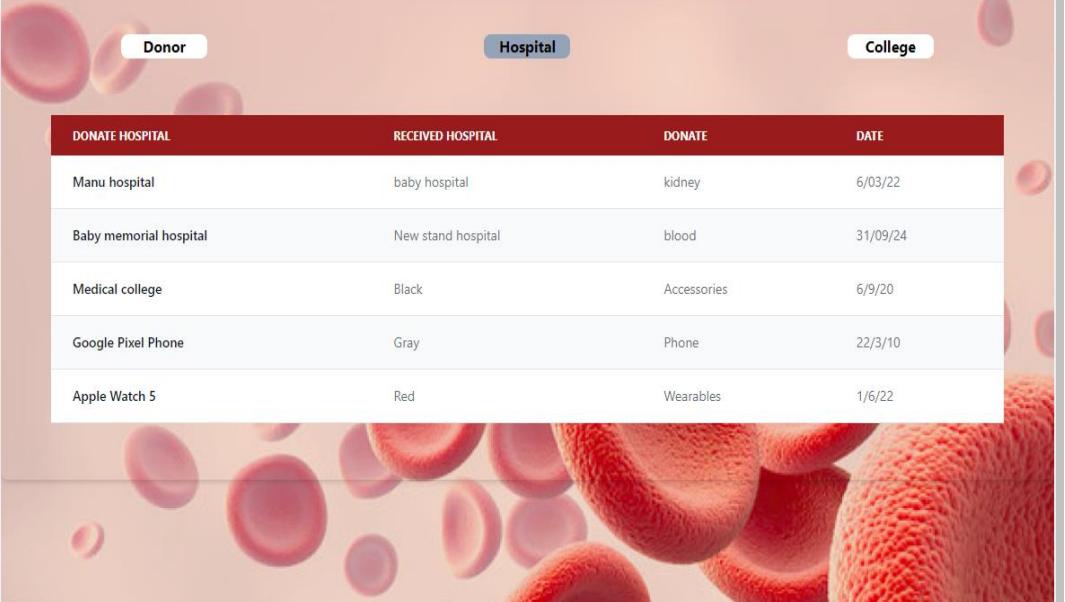
localhost:3000/admin/hospitaldonation

ADMIN **LIFELINK** **HOME** **LOGOUT**

HOSPITAL **USER** **BLOOD DONOR** **DONATION** **COLLEGE**

Donor **Hospital** **College**

DONATE HOSPITAL	RECEIVED HOSPITAL	DONATE	DATE
Manu hospital	baby hospital	kidney	6/03/22
Baby memorial hospital	New stand hospital	blood	31/09/24
Medical college	Black	Accessories	6/9/20
Google Pixel Phone	Gray	Phone	22/3/10
Apple Watch 5	Red	Wearables	1/6/22



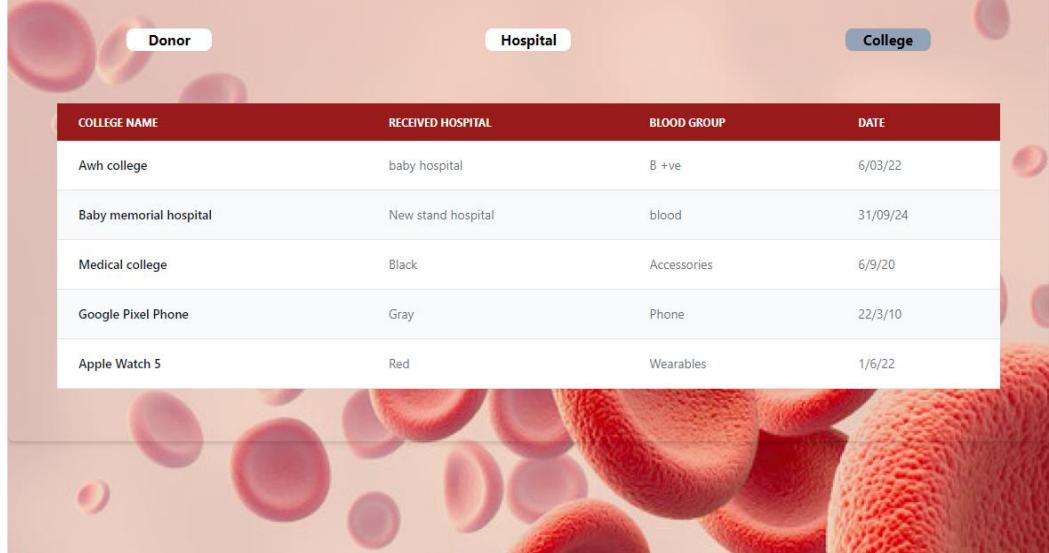
localhost:3000/admin/collegedonation

ADMIN **LIFELINK** **HOME** **LOGOUT**

HOSPITAL **USER** **BLOOD DONOR** **DONATION** **COLLEGE**

Donor **Hospital** **College**

COLLEGE NAME	RECEIVED HOSPITAL	BLOOD GROUP	DATE
Awh college	baby hospital	B +ve	6/03/22
Baby memorial hospital	New stand hospital	blood	31/09/24
Medical college	Black	Accessories	6/9/20
Google Pixel Phone	Gray	Phone	22/3/10
Apple Watch 5	Red	Wearables	1/6/22



Admin manage organ donor

The screenshot shows a web application interface titled "MANAGE ORGANDONOR". On the left, there is a sidebar with a red header "ADD CATEGORY" and several menu items: HOSPITAL, USER, BLOOD DONOR, DONATION, COLLEGE, and ORGAN DONORS. The main content area has a background image of red blood cells. It displays a table with columns: NAME, AGE, EMAIL, ADDRESS, BLOODGROUP, CONTACT, and STATUS. The table contains four rows of data:

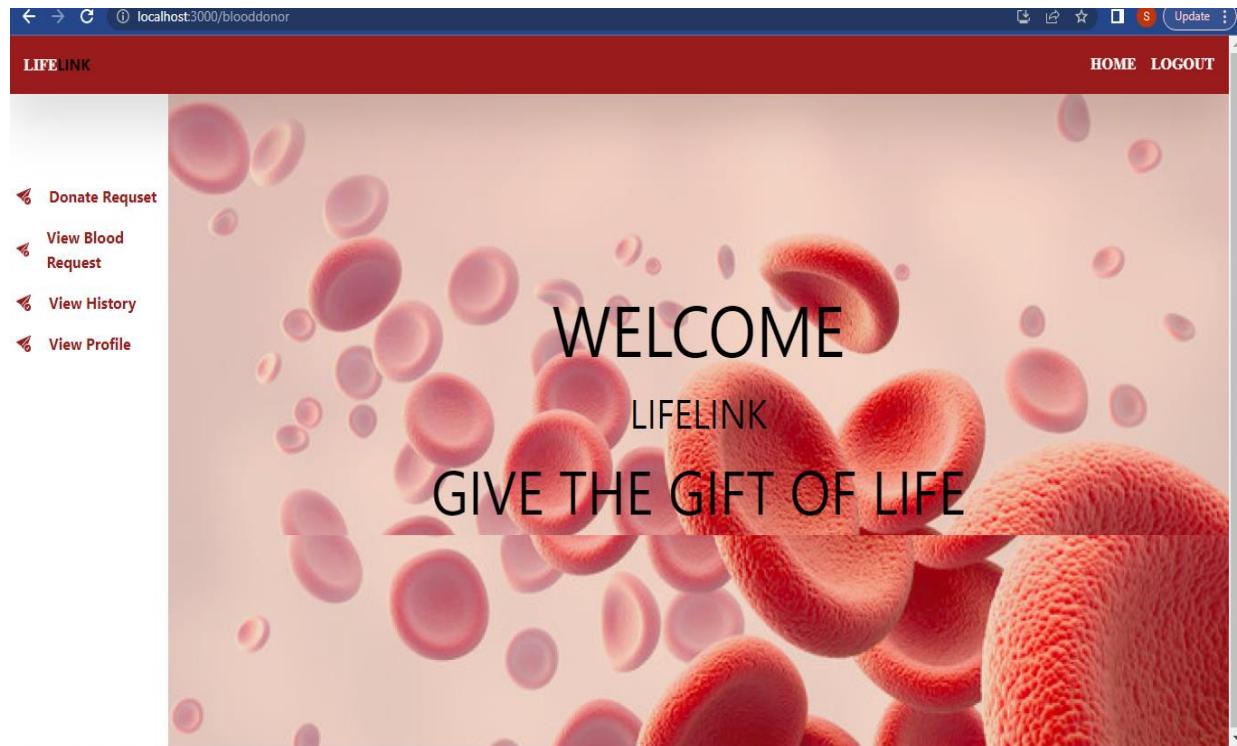
NAME	AGE	EMAIL	ADDRESS	BLOODGROUP	CONTACT	STATUS	
dhevan	39	dhevan@gmail.com	puthiyottil house	b-ve	95885745654565	Accepted	View
kiran	34	kir@gmail.com	puthanweetil	o-ve	67523263553	Accepted	View
Vandhana	54	vandhu@gmail.com	thekkethodi	o+ve	77655543433	Accepted	View
Varna	45	var@gmail.com	paalakott	ab-ve	78855543433	Accepted	View

Admin add categories

The screenshot shows a web application interface titled "ADD CATEGORY". On the left, there is a sidebar with a red header "ADD CATEGORY" and several menu items: HOSPITAL, USER, BLOOD DONOR, DONATION, COLLEGE, and ORGAN DONORS. The main content area has a background image of red blood cells. It features two input fields: "bone marrow" and "cornea", and a red button "Add Category". Below these fields, there are two sections: "AFTER DEATH ORGANS" and "BEFORE DEATH ORGANS". The "AFTER DEATH ORGANS" section contains "heart" and "liver". The "BEFORE DEATH ORGANS" section contains "kidney" and "liver".

AFTER DEATH ORGANS	BEFORE DEATH ORGANS
heart	kidney
co	liver

Blood donors home page



Blood donor send request

The screenshot shows a web browser window for 'localhost:3000/blooddonor/donorreq'. The title bar says 'localhost:3000/blooddonor/donorreq'. The header has a red bar with 'LIFELINK' on the left and 'HOME LOGOUT' on the right. On the left, there's a sidebar with icons for 'Donate Request', 'View Blood Request', 'View History', and 'View Profile'. The main content area has a background image of red blood cells. It contains form fields for 'District' (set to 'thrissur'), 'Hospital' (set to 'ashoka'), 'Blood group' (set to 'o-ve'), and 'Date' (set to '12-04-2024'). A 'SUBMIT' button is at the bottom.

Blood donor view user request

The screenshot shows a web browser window for 'localhost:3000/blooddonor/viewbloodrqst'. The title bar says 'LIFE LINK' and has 'HOME LOGOUT' buttons. On the left, there's a sidebar with links: 'Donate Request', 'View Blood Request', 'View History', and 'View Profile'. The main content area is titled 'BLOOD REQUEST' and contains a table with one row. The table columns are NAME, HOSPITAL NAME, PLACE, DISTRICT, BLOOD GROUP, CONTACT, STATUS, and ACTION. The data is: NAME - gggggg, HOSPITAL NAME - Abcd org, PLACE - pavanghad, DISTRICT - calicut, BLOOD GROUP - b-, CONTACT - 879786765, STATUS - Pending, ACTION - two buttons: 'Accept' and 'Reject'. The background features a 3D rendering of red blood cells.

NAME	HOSPITAL NAME	PLACE	DISTRICT	BLOOD GROUP	CONTACT	STATUS	ACTION
gggggg	Abcd org	pavanghad	calicut	b-	879786765	Pending	<button>Accept</button> <button>Reject</button>

Blood donor view history

The screenshot shows a web browser window for 'localhost:3000/blooddonor/donatehist'. The title bar says 'LIFE LINK' and has 'HOME LOGOUT' buttons. On the left, there's a sidebar with links: 'Donate Request', 'View Blood Request', 'View History', and 'View Profile'. The main content area contains a table with five rows of historical data. The table columns are DISTRICT, HOSPITAL NAME, BLOOD GROUP, DATE, and STATUS. The data is:

DISTRICT	HOSPITAL NAME	BLOOD GROUP	DATE	STATUS
Kozhikode	mims	A +ve	7/5/24	Pending
Baby memorial hospital	New stand	Baby@gmail.com	9782676222	Accepted
Medical college	Black	Accessories	\$99	Pending
Google Pixel Phone	Gray	Phone	\$799	Rejected
Apple Watch 5	Red	Wearables	\$999	

Blood donor update profile

Name : ashoka

Post office: mukkam

Age : 776666

Pin : 776666

Email : ash@gmail.com

District : thrissur

Contact : 76625635253

Blood group:

Place : vadakara

Weight :

House name :

Height :

Hospital home page

Blood donor request

Organ donors

View hospital request

My Request

College Request

View User Request

Blood donor history

Organ donor history

View Profile

WELCOME

LIFELINK

GIVE THE GIFT OF LIFE

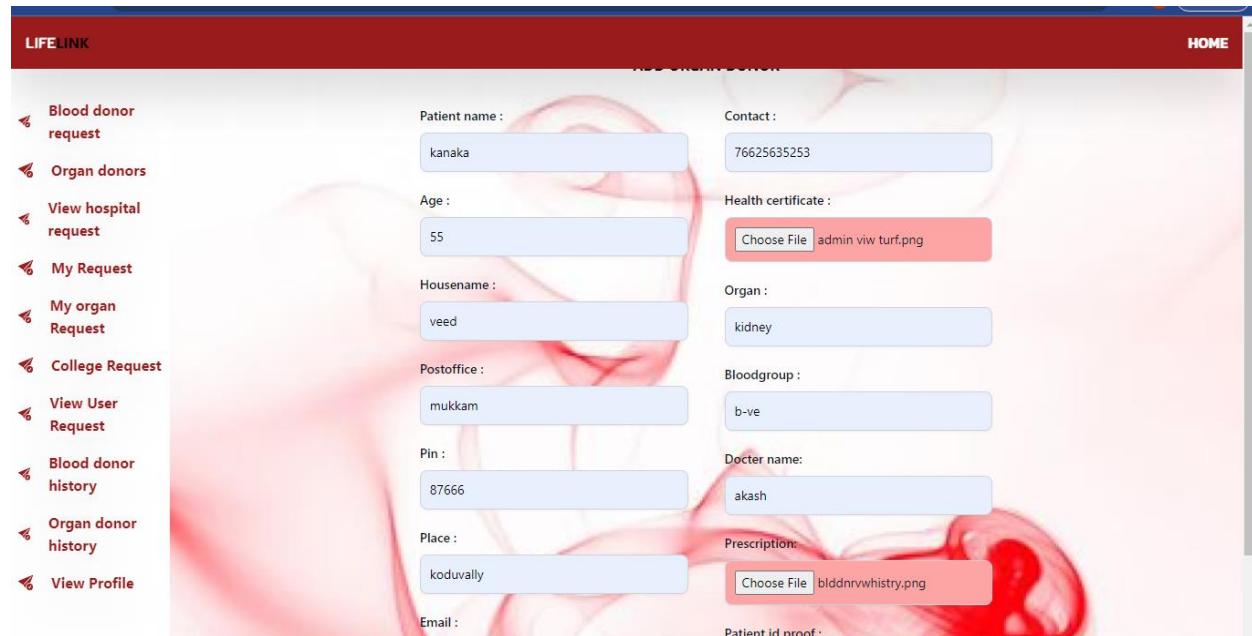
Hospital Manage blood donor request

pranav
xsx mankaav...
Email pra@gmail.com
Contact 76625635253
Certificate view
Blood group a-ve
Weight 79
height 188cm
Date 2024-04-04

Hospital view organ donors

NAME	AGE	ADDRESS	EMAIL	CONTACT	BLOODGROUP	DATE OF DEATH
dhevan	39	puthiyottil house	dhevan@gmail.com	95885745654565	b-ve	View
kiran	34	puthanveetil	kir@gmail.com	67523263553	o-ve	View
Vandhana	54	thekkethodi	vandhu@gmail.com	77655543433	o+ve	View
Varna	45	paalakkott	var@gmail.com	78855543433	ab-ve	View

Send request to organ donor



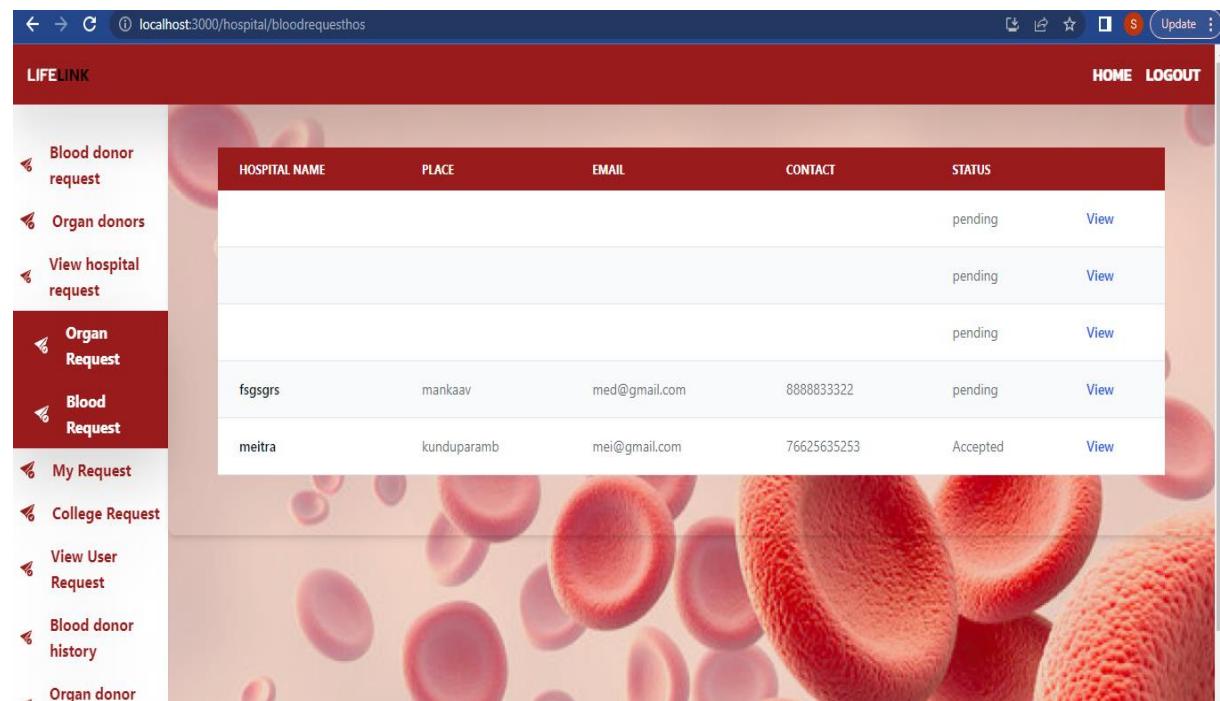
The screenshot shows a web-based application for sending a request to an organ donor. The interface has a red header bar with the text "LIFE LINK" on the left and "HOME" on the right. On the left side, there is a sidebar with a list of navigation items:

- Blood donor request
- Organ donors
- View hospital request
- My Request
- My organ Request
- College Request
- View User Request
- Blood donor history
- Organ donor history
- View Profile

The main form area contains the following fields:

Patient name :	kanaka	Contact :	76625635253
Age :	55	Health certificate :	<input type="button" value="Choose File"/> admin viw turf.png
Housename :	veed	Organ :	kidney
Postoffice :	mukkam	Bloodgroup :	b-ve
Pin :	87666	Docter name:	akash
Place :	koduvally	Prescription:	<input type="button" value="Choose File"/> bldnnrvwhistry.png
Email :			
Patient id proof :			

Hospital view hospital blood request



The screenshot shows a table of hospital blood requests. The interface has a red header bar with the text "LIFE LINK" on the left and "HOME LOGOUT" on the right. On the left side, there is a sidebar with a list of navigation items:

- Blood donor request
- Organ donors
- View hospital request
- Organ Request
- Blood Request
- My Request
- College Request
- View User Request
- Blood donor history
- Organ donor

The main table area displays the following data:

HOSPITAL NAME	PLACE	EMAIL	CONTACT	STATUS	
				pending	View
				pending	View
fsqsgsrs	mankaav	med@gmail.com	8888833322	pending	View
meitra	kunduparamb	mei@gmail.com	76625635253	Accepted	View

The screenshot shows a web application interface for a hospital. On the left, a sidebar menu lists various request types. The main content area displays the following details:

- Hospital name: meitra
- Pin: 765554
- Place: kunduparamb
- District: thrissur
- Email: mei@gmail.com
- Blood group: o-ve
- Contact: 76625635253
- Blood unit: 1000ml

A large red button labeled "ACCEPT" is centered at the bottom.

Hospital view organ request

The screenshot shows a table listing organ requests from the hospital "Baby memorial hospital". The columns are:

HOSPITAL NAME	PLACE	DISTRICT	EMAIL	CONTACT	POSTOFFICE	PIN	STATUS	VIEW
Baby memorial hospital	puthiya stand	kannur	baby@gmail.com	7662563525	puthiya stand	776666	Accepted	View
Baby memorial hospital	puthiya stand	kannur	baby@gmail.com	7662563525	puthiya stand	776666	Accepted	View
Baby memorial hospital	puthiya stand	kannur	baby@gmail.com	7662563525	puthiya stand	776666	Accepted	View
Baby memorial hospital	puthiya stand	kannur	baby@gmail.com	7662563525	puthiya stand	776666	Accepted	View

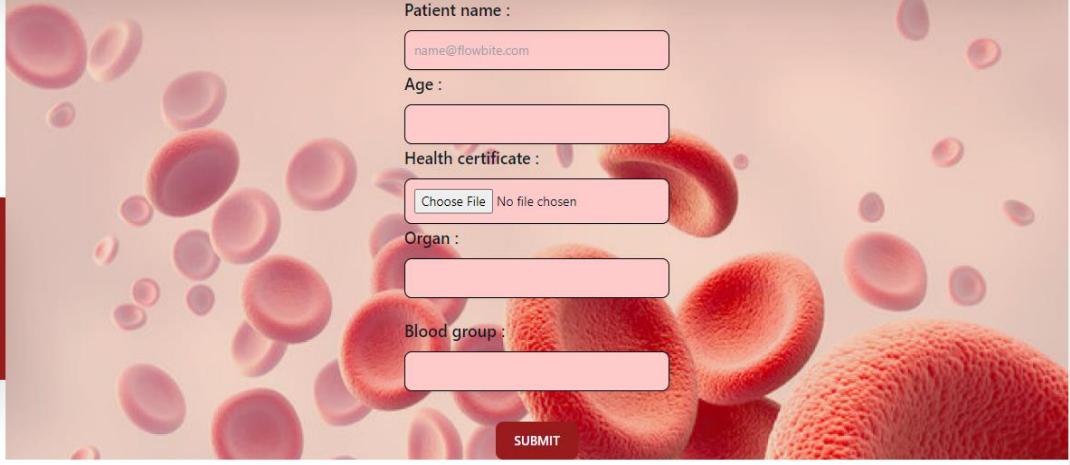
The screenshot shows a web browser window for the 'LIFELINK' application at localhost:3000/hospital/vieworganpage/660d34fc913c9a58d30c2fc7. The page displays a patient profile for 'meitra' from 'thrissur kunduparamb'. The profile includes fields for Email (mei@gmail.com), Contact (76625635253), Patientname (varna), Age (33), Blood group (ab+ve), organ (kidn), and Healthcertificate (view). A red 'ASSIGN' button is visible in the profile card. The left sidebar contains a navigation menu with items like 'Blood donor request', 'Organ donors', 'View hospital request', 'Organ Request', 'Blood Request', 'My Request', 'College Request', 'View User Request', 'Blood donor history', 'Organ donor history', and 'View Profile'. The top right corner has 'HOME' and 'LOGOUT' links.

Hospital send request to organ donor

The screenshot shows a web browser window for the 'LIFELINK' application at localhost:3000/hospital/Searchorgan/662a3fcfb635d4b7936c3140. The page displays a list of organ donors with columns for NAME, AGE, ADDRESS, EMAIL, CONTACT, STATUS, and ACTION. The donors listed are dhevan, kiran, Vandhana, and Varna, all with Accepted status. Each row has an 'ASSIGN' button and a 'View' link. The left sidebar contains a navigation menu with items like 'Blood donor request', 'Organ donors', 'View hospital request', 'Organ Request', 'Blood Request', 'My Request', 'My organ Request', 'College Request', 'View User Request', 'Blood donor history', and 'Organ donor history'. The top right corner has 'HOME' and 'LOGOUT' links.

NAME	AGE	ADDRESS	EMAIL	CONTACT	STATUS	ACTION
dhevan	39	puthiyottil house	dhevan@gmail.com	95885745654565	Accepted	<button>ASSIGN</button> View
kiran	34	puthanveetil	kir@gmail.com	67523263553	Accepted	<button>ASSIGN</button> View
Vandhana	54	thekkethodi	vandhu@gmail.com	77655543433	Accepted	<button>ASSIGN</button> View
Varna	45	paalakkott	var@gmail.com	78855543433	Accepted	<button>ASSIGN</button> View

Hospital add organ request



LIFE LINK

HOME LOGOUT

- [Blood donor request](#)
- [Organ donors](#)
- [View hospital request](#)
- [My Request](#)
 - [Organ request](#)
 - [Blood Request](#)
 - [Blood Request college](#)
- [College Request](#)
- [View User Request](#)
- [Blood donor history](#)
- [Organ donor history](#)

Patient name :

Age :

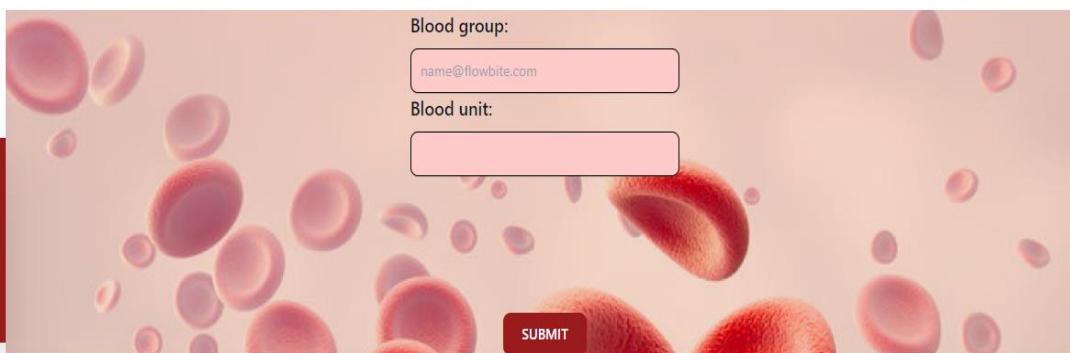
Health certificate : No file chosen

Organ :

Blood group :

SUBMIT

Hospital add blood request to hospital



LIFE LINK

HOME LOGOUT

- [Blood donor request](#)
- [Organ donors](#)
- [View hospital request](#)
- [My Request](#)
 - [Organ request](#)
 - [Blood Request](#)
 - [Blood Request college](#)
- [College Request](#)
- [View User Request](#)
- [Blood donor history](#)
- [Organ donor history](#)

Blood group:

Blood unit:

SUBMIT

Hospital send blood request to college

LIFELINK

HOME LOGOUT

District : eranakulam

college : fisat

Blood group : name@flowbite.com

Number of students :

SUBMIT

- [View hospital request](#)
- [My Request](#)
- [Organ request](#)
- [Blood Request](#)
- [Blood Request college](#)
- [College Request](#)
- [View User Request](#)
- [Blood donor history](#)
- [Organ donor history](#)
- [View Profile](#)

Hospital manage college request

LIFELINK

HOME

BLOOD REQUEST

COLLEGE NAME	PLACE	DISTRICT	CAMPNAME	DESCRIPTION	DATE	STATUS	ACTION
Awh engineering college	kuttikattoor	kozhikode	bloodlink	our college have blood donation camp	2024-05-01	Accepted	Accept Reject

- [Blood donor request](#)
- [Organ donors](#)
- [View hospital request](#)
- [My Request](#)
- [My organ Request](#)
- [College Request](#)
- [View User Request](#)
- [Blood donor history](#)
- [Organ donor history](#)
- [View Profile](#)

Hospital manage user request

NAME	HOSPITAL NAME	PLACE	DISTRICT	BLOOD GROUP	BLOOD UNIT	CONTACT	STATUS	ACTION
gggggg	Abcd org	pavanghad	calicut	b-	45ml	879786765	Pending	<button>Accept</button> <button>Reject</button>

Hospital view blood donor history

NAME	AGE	PLACE	EMAIL	CONTACT
Manu	22	new stand	aster mims	19/7/22
Baby memorial hospital	New stand	Baby@gmail.com	9782676222	31/09/24
Medical college	Black	Accessories	\$99	6/9/20
Google Pixel Phone	Gray	Phone	\$799	22/3/10
Apple Watch 5	Red	Wearables	\$999	1/6/22

Hospital view hospital send and received blood request

localhost:3000/hospital/hospitalbloodhist

LIFE LINK

HOME LOGOUT

- Blood donor request
- Organ donors
- View hospital request
- My Request
- College Request
- View User Request
- Blood donor history
- Organ donor history
- View Profile

HOSPITAL SEND REQUEST

HOSPITAL NAME	PLACE	DISTRICT	BLOOD GROUP	BLOOD UNIT	DATE	STATUS
---------------	-------	----------	-------------	------------	------	--------

LIFE LINK

HOME LOGOUT

- Blood donor request
- Organ donors
- View hospital request
- My Request
- College Request
- View User Request
- Blood donor history
- Organ donor history
- View Profile

HOSPITAL RECEIVED REQUEST

HOSPITAL NAME	PLACE	DISTRICT	BLOOD GROUP	BLOOD UNIT	DATE
Manu hospital	baby hospital	kidney	Blood group	Blood group	6/03/22
Baby memorial hospital	New stand hospital	blood	Blood group	Blood group	31/09/24
Medical college	Black	Accessories	Blood group	Blood group	6/9/20
Google Pixel Phone	Gray	Phone	Blood group	Blood group	22/3/10
Apple Watch 5	Red	Wearables	Blood group	Blood group	1/6/22

Hospital view send request to organ donor

The screenshot shows a web browser window for the 'LIFE LINK' platform. The URL is localhost:3000/hospital/sendorganrqst. The page has a red header bar with 'LIFE LINK' and 'HOME LOGOUT'. On the left, there's a sidebar with a list of links: 'Blood donor request', 'Organ donors', 'View hospital request', 'My Request', 'College Request', 'View User Request', 'Blood donor history', 'Organ donor history', 'Send organ request', and 'Received organ request'. The main content area displays a table of organ donor requests:

PATIENT NAME	AGE	HEALTH CERTIFICATE	HOSPITAL NAME	STATUS
Varun	33	View	New stand	Accepted View
Baby memorial hospital	New stand	Baby@gmail.com	9782676222	Accepted View
Medical college	Black	Accessories	\$99	Accepted View
Google Pixel Phone	Gray	Phone	\$799	Accepted View
Apple Watch 5	Red	Wearables	\$999	Accepted View

The screenshot shows a web browser window for the 'LIFE LINK' platform. The URL is localhost:3000/hospital/sendorganhistpage. The page has a red header bar with 'LIFE LINK' and 'HOME LOGOUT'. On the left, there's a sidebar with a list of links: 'Blood donor request', 'Organ donors', 'View hospital request', 'My Request', 'College Request', 'View User Request', 'Blood donor history', 'Organ donor history', 'Send organ request', and 'Received organ request'. The main content area displays a 'HOSPITAL REQUEST' section with patient details and an 'ORGAN DONOR DETAILS' section with donor details:

HOSPITAL REQUEST

- Patient name :abi
- Age :22
- health certificate :
- Organ : [View](#)
- Blood group : B-ve

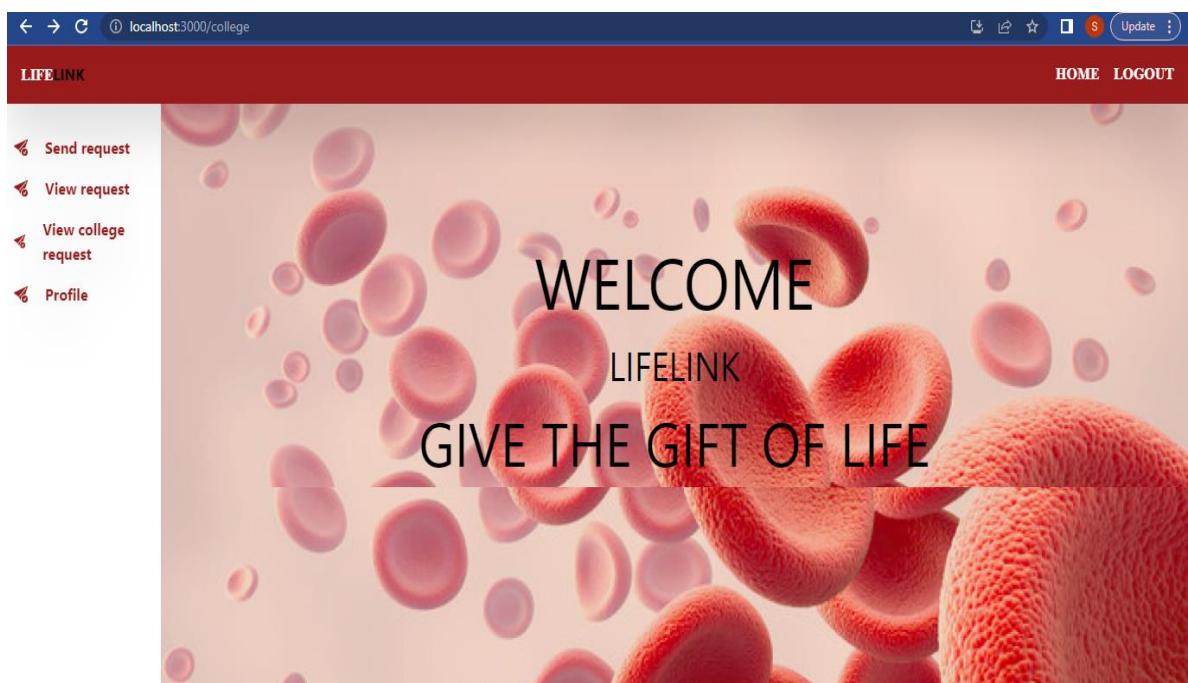
ORGAN DONOR DETAILS

Name : abi	District : kannur
Age : 45	Organ : liver
Place : kuti	Blood group : B-
Email : kuti@gmail.com	Height : 98
Contact : 0994787855	Weight : 98

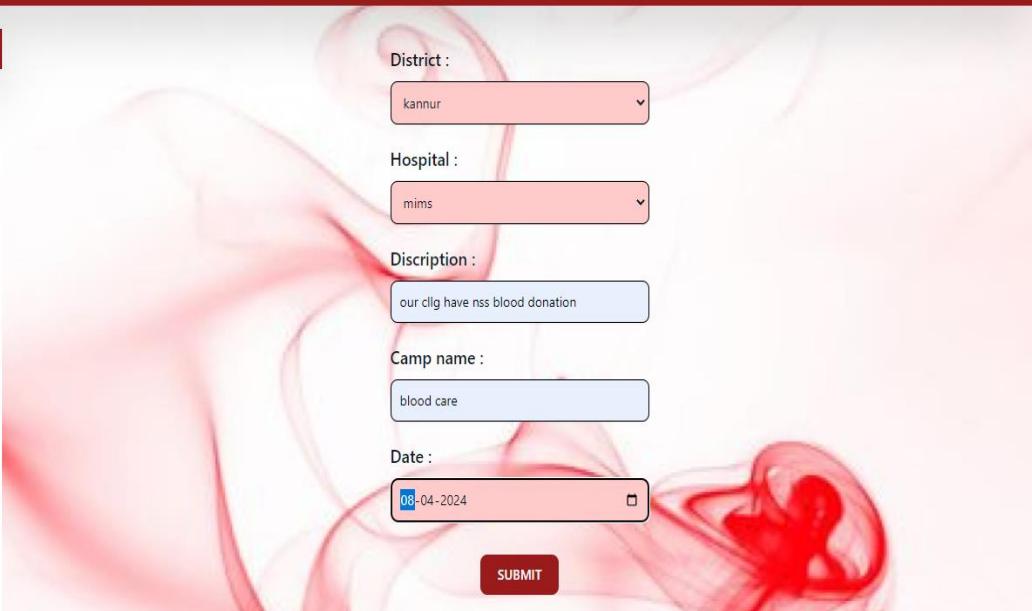
Hospital update profile

The screenshot shows a web browser window for the URL `localhost:3000/hospital/editprofile`. The page has a dark red header bar with the text "LIFE LINK" on the left and "HOME LOGOUT" on the right. A toolbar at the top includes icons for back, forward, search, and update. On the left, a sidebar lists various options: "Blood donor request", "Organ donors", "View hospital request", "My Request", "College Request", "View User Request", "Blood donor history", "Organ donor history", and "View Profile". The main content area contains form fields for updating a hospital profile. The fields include: "Hospital name" (ashoka), "Place" (vadakara), "Email" (ash@gmail.com), "Contact" (7662563525), "Liscence no" (7665555), "proof" (represented by a small green icon and a "Choose File" button with "No file chosen"), "Post office" (mukkam), "Pin" (776666), and "District" (thirissur). At the bottom right are "Update" and "Delete" buttons.

College home page



College send request



localhost:3000/college/collegerqst

LIFELINK HOME LOGOUT

Send request

- [View request](#)
- [View college request](#)
- [Profile](#)

District :

Hospital :

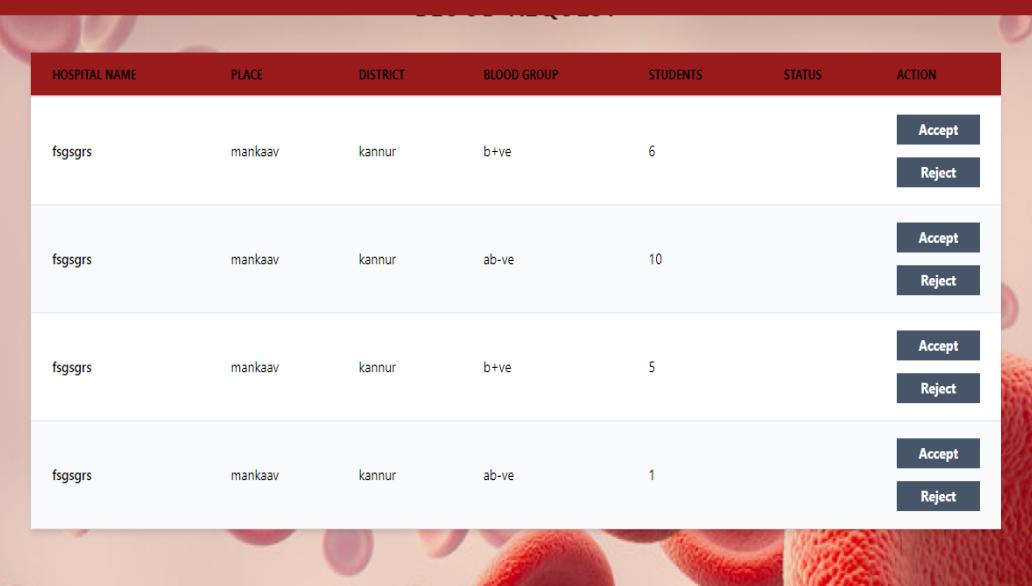
Description :

Camp name :

Date :

SUBMIT

College view hospital request



localhost:3000/college/viewhospitalrqst

LIFELINK HOME LOGOUT

Send request

- [View request](#)
- [View college request](#)
- [Profile](#)

HOSPITAL NAME	PLACE	DISTRICT	BLOOD GROUP	STUDENTS	STATUS	ACTION
fsgsgrs	mankaav	kannur	b+ve	6	Accepted	Accept
fsgsgrs	mankaav	kannur	ab-ve	10	Accepted	Accept
fsgsgrs	mankaav	kannur	b+ve	5	Accepted	Accept
fsgsgrs	mankaav	kannur	ab-ve	1	Accepted	Accept

College update profile

localhost:3000/college/editprofile

LIFELINK

[HOME](#) [LOGOUT](#)

Name : ashoka

Certificate : Choose File No file chosen

Place : vadakara

Post office: mukkam

Email : ash@gmail.com

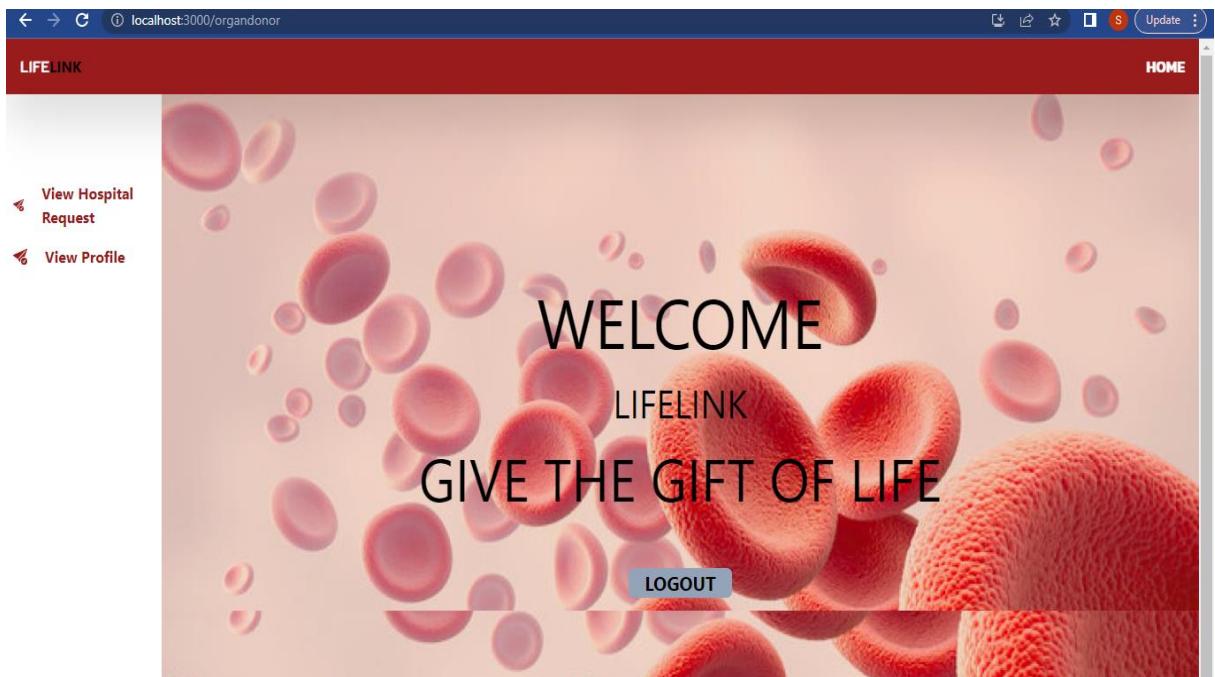
Pin : 776666

Contact : 76625635253

District : thrissur

Update **Delete**

Organ donor home page



View and accept hospital request

localhost:3000/organdonor/viewhospitalrequest

LIFELINK

MANAGE HOSPITAL REQUEST

HOSPITAL NAME	PLACE	DISTRICT	EMAIL	CONTACT	PROOF	STATUS	
kmct hospital	mukkam	Kozhikkode	kmct@gmail.com	9745465656	view	pending	View
kmct hospital	mukkam	Kozhikkode	kmct@gmail.com	9745465656	view	pending	View
kmct hospital	mukkam	Kozhikkode	kmct@gmail.com	9745465656	view	Accepted	View

localhost:3000/organdonor/viewpagehospitalrequestorgan/662a3cb7f635d4b7936c2fc0

LIFELINK

HOME

View Hospital Request

View Profile

kmct hospital	mukkammukkam
Email	kmct@gmail.com
Certificate	view
Patient name	baskar
Age	44
Address	puthanveetil
Blood group & Organ	ab+vekidney
Date	2024-04-25T11:17:36.005Z 2024-04-25T11:21:27.468Z
Doctor	maanav
Prescription	C:\fakepath\clgsendreq.png

Update profile

Name : Varna Height: 156

Age : 45 Weight: 56kg

Email : var@gmail.com

Contact : 78855543433

Address: paalakott

Date of birth : 1982-07-22T00:00:00.000Z

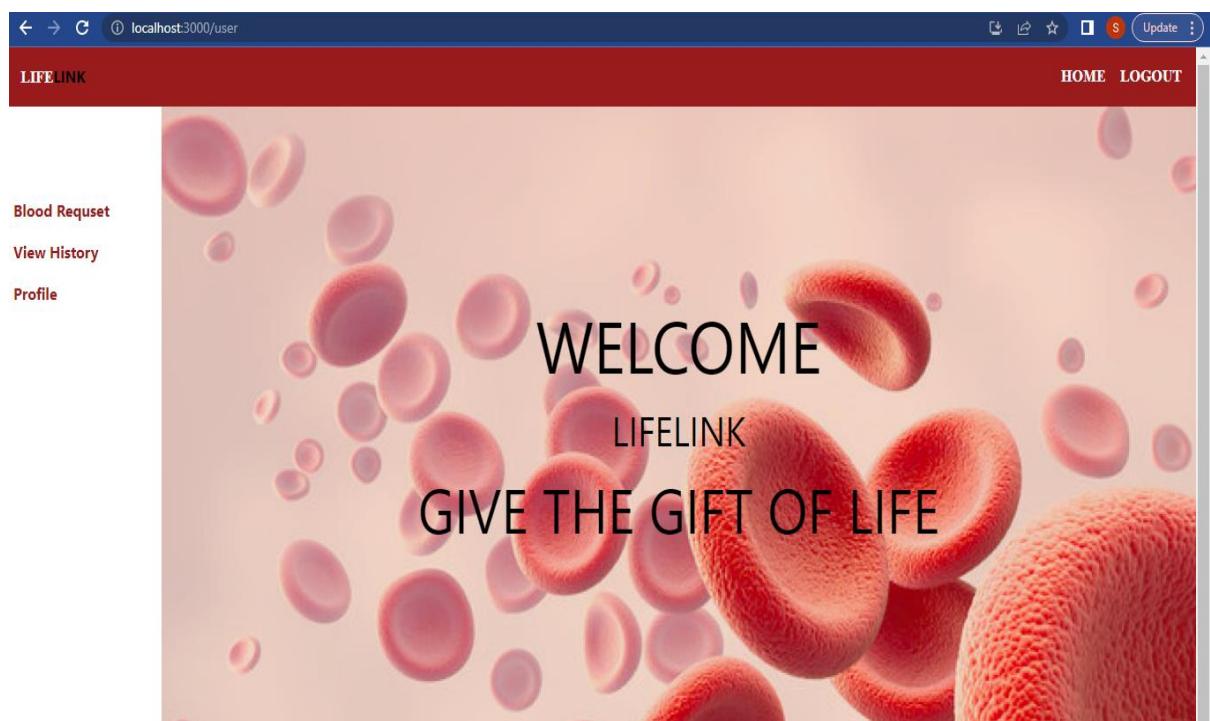
Conformation certificate :

Nominee : abhinav

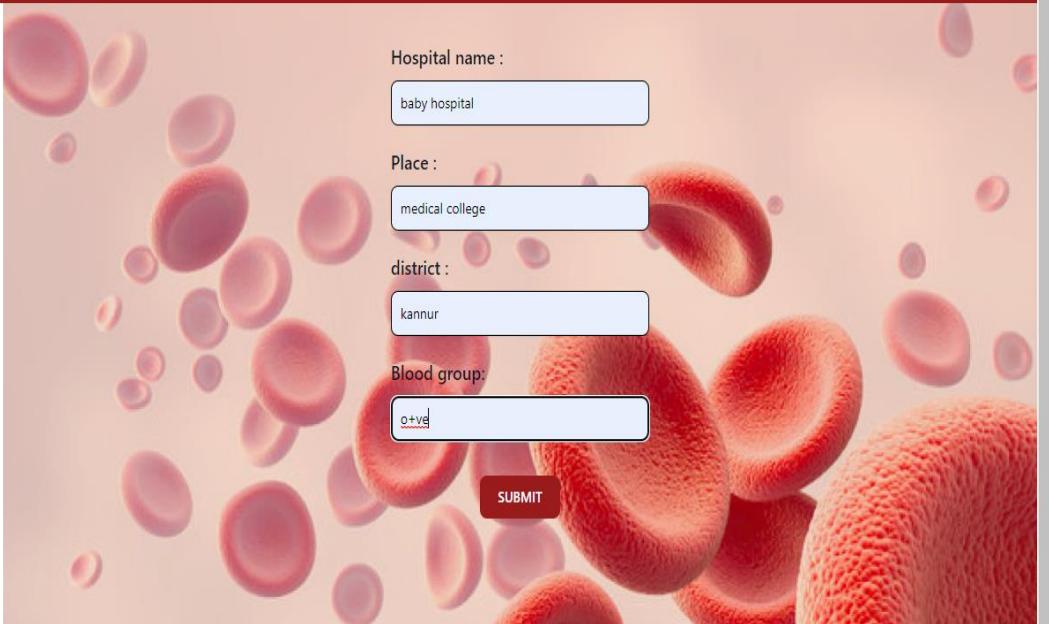
Nominee contact : 9844885698

Nominee relation : husband

User home page



User send request



LIFE LINK

localhost:3000/user/bloodreq

HOME LOGOUT

Blood Request

View History

Profile

Hospital name :

Place :

district :

Blood group:

SUBMIT

User view history



LIFE LINK

localhost:3000/user/viewhistory

HOME LOGOUT

Blood Request

View History

Profile

HOSPITAL NAME	PLACE	DISTRICT	BLOOD	STATUS
mims hospital	kuttikattoor		b+ve	pending
baby hospital	stand		o+ve	pending
kalam hospital	koduvally	trivandrum	ab+ve	pending
kmct hospital	mankaav	xsns	ab+ve	pending

User update profile

localhost:3000/user/profile

LIFE LINK

HOME LOGOUT

Name : varna

House name : Varnas

Age : 34

Post office : medical college

Place : medical college

Pin : 6744221

Email : vae@gmail.com

District: kollam

Contact : 88776566544

Update Delete

BIBLIOGRAPHY

7.BIBLIOGRAPHY

Websites

- [1] https://www.w3schools.com/js/js_functions.asp
- [2] <https://developer.mozilla.org/en-US/docs/Web/HTML>
- [3] <https://developer.mozilla.org/en-US/docs/Web/CSS>
- [4] <https://nodejs.org/en/docs>
- [5] <https://www.mongodb.com/docs>

Books

- [1] *Thomas A Powell, Fritz Schneider, "JavaScript: The Complete Reference", 3rd Edition, Tata McGraw Hill*
- [2] *Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node" by Vasan Subramanian*
- [3] *Code complete: a practical handbook of software construction by Steve McConnell, Microsoft Press,2nd Edition(2004).*
- [4] *Full Stack Development with JHipster: Build scalable and maintainable web applications using the MERN stack" by Deepu K Sasidharan and Sendil Kumar N*
- [5] *Full Stack JavaScript: Learn Backbone.js, Node.js, and MongoDB" by Azat Mardan*

