

AGILE

PROJECT

BLOOD MANAGEMENT SYSTEM WEB APPLICATION

SUBMITTED BY

ASHITHA P

(ROLL NO:80521006)

ATHIRA MOHANDAS

(ROLL NO:80521007)

TITLE

Development of a Blood Management System Website by Agile Methodology.

INDRODUCTION

1.1 Purpose

The purpose of this SRS document is to provide a detailed description of the Blood Bank Management Web Application. The document specifies the requirements, design, and implementation of the web application. It is designed to guide the development team through the software development process.

1.2 Scope

The Blood Bank Management Web Application is a web-based application designed to manage donor information, . The application will allow authorized users to access the database and perform various tasks related to blood donation and transfusion procedures. The application will be developed using HTML, CSS, React.js, and Node.js, with the database management system MongoDB.

1.3 Definitions, Acronyms, and Abbreviations

SRS - Software Requirements Specification

HTML - Hypertext Markup Language

CSS - Cascading Style Sheets

React.js - JavaScript library for building user interfaces

Node.js - JavaScript runtime environment

MongoDB - NoSQL database management system

PROJECT GOALS

- . Improve efficiency and accuracy of blood donation management.
- . Facilitate donor management: The website should enable easy registration of new donors, maintenance of their records, and tracking of their eligibility to donate blood. This would make it easier for blood banks to identify eligible donors and increase their donor pool.
- . Increase accessibility of blood services: The website should make it easier for people to access blood services by providing information about blood donation drives, locations of blood banks, and availability of blood types.

- . Ensure security of data: The website should ensure that donor and patient information is kept secure, with appropriate encryption and access controls in place. This would protect the privacy of individuals and ensure that sensitive data is not misused.

TECHNOLOGY STACK

The application will be built using HTML, CSS, React.js, and Node.js technologies. The front-end will be built using HTML and CSS, with React.js used for building user interfaces. The back-end will be built using Node.js, with the Express web framework used for server-side programming. The database management system will be MongoDB, a NoSQL database system.

REQUIREMENTS

2.1 User Requirements

2.1.1 Authorization

The application should allow authorized users to access the system and perform tasks based on their level of access.

2.1.2 Donor Management

The application should allow users to add, update, and delete donor information. The application should also be able to track donor eligibility, blood type, and medical history.

2.1.3 Transfusion Procedures

The application should allow users to record transfusion procedures, including tracking the patient's blood type, transfusion date, and transfusion outcome.

2.1.4 Reporting

The application should allow users to generate reports on donor information, and transfusion procedures.

2.2 System Requirements

2.2.1 User Interface

The application should have a user-friendly interface, allowing users to easily navigate and perform tasks. The interface should be responsive and work on different devices and screen sizes.

2.2.2 Data Management

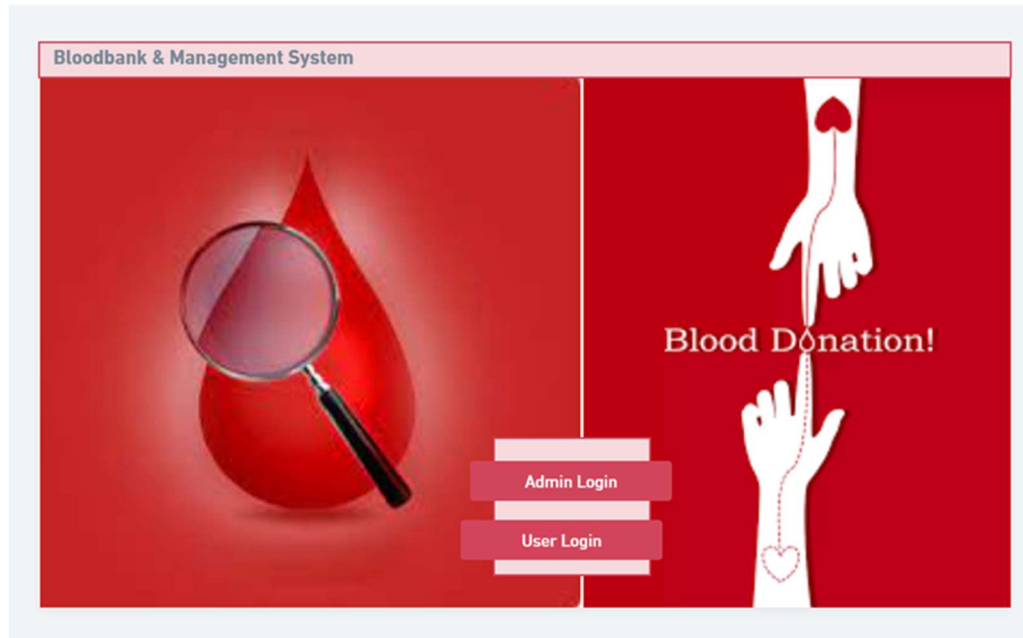
The application should be able to store and retrieve data related to donor information, blood inventory, and transfusion procedures. The database should be secure and scalable, with MongoDB used as the database management system.

2.2.3 Performance

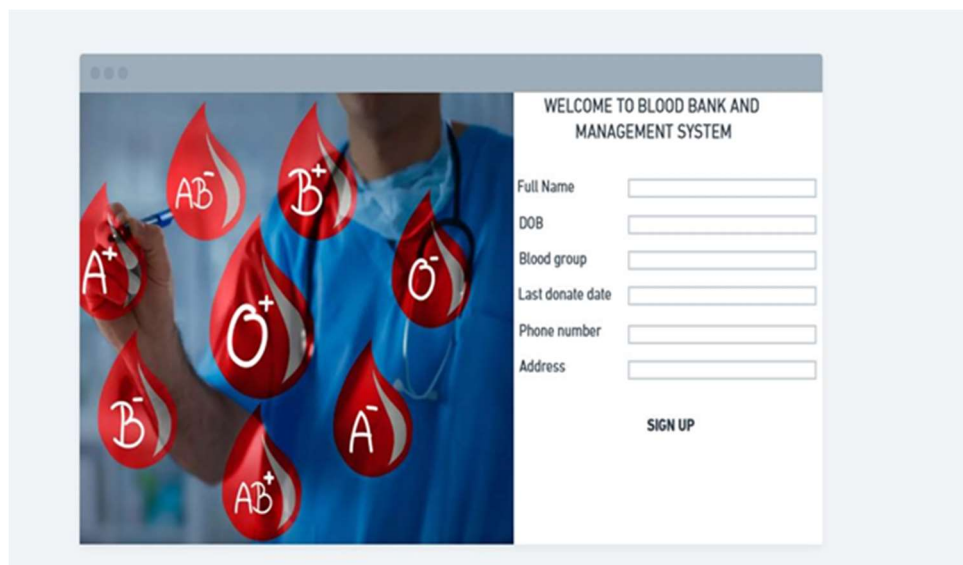
The application should perform efficiently and quickly, with minimal loading times and downtime.

WIREFRAMES

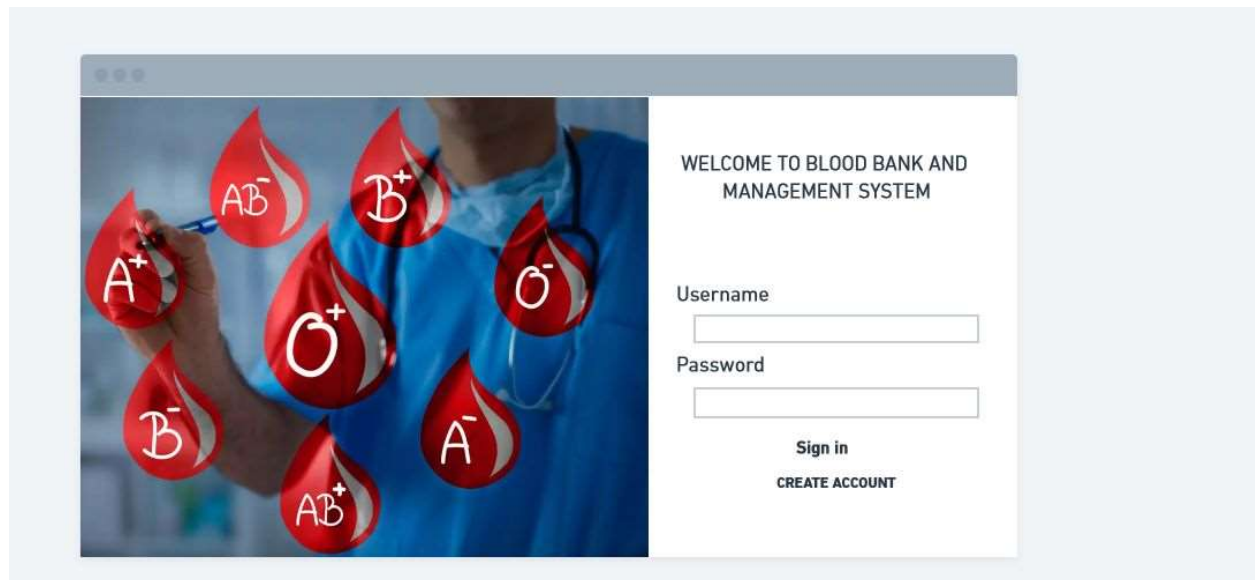
1.Home Page



2.Sign Up Page



3. Sign in Page



The sign-in page features a large image on the left showing a person in blue scrubs holding several red blood drop icons with different blood types: A⁺, A^B, B⁺, O⁺, O⁻, B⁻, A⁻, and A^B⁺. On the right, there is a white box containing the text "WELCOME TO BLOOD BANK AND MANAGEMENT SYSTEM". Below this text are two input fields: "Username" and "Password". At the bottom of the box are two buttons: "Sign in" and "CREATE ACCOUNT".

WELCOME TO BLOOD BANK AND MANAGEMENT SYSTEM

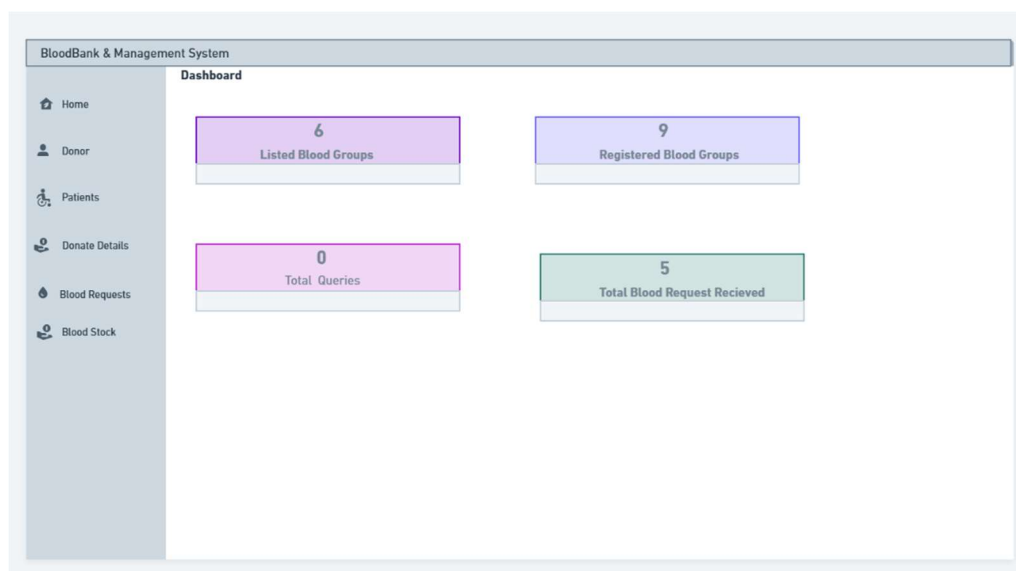
Username

Password

[Sign in](#)

[CREATE ACCOUNT](#)

4. Dashboard



BLOOD BANK MANAGEMENT DETAILS

HOME

DONAR

PATIENTS

BLOOD REQUEST

BLOOD STOCK

Patient name	Age	Reason	Blood Group	Unit	Date	Actions
Ravi	45	Accident	O+	4	12/09	<div>APPROVE</div> <div>DELETE</div>
Anju	24	Surgery	O+	8	12/09	<div>APPROVE</div> <div>DELETE</div>
Edwin	34	Fever	O+	3	03/09	<div>APPROVE</div> <div>DELETE</div>

BLOOD BANK MANAGEMENT DETAILS

HOME

DONAR

PATIENTS

BLOOD REQUEST

BLOOD STOCK

Donar name	Age	Blood Group	Unit	Date	Actions	
Raiju	26	AB-	4	12/09	<div>ACCEPT</div> <div>DELETE</div>	
Anuja	24	O+	8	12/09	<div>ACCEPT</div> <div>DELETE</div>	
Edison	34	A+	3	03/09	<div>ACCEPT</div> <div>DELETE</div>	

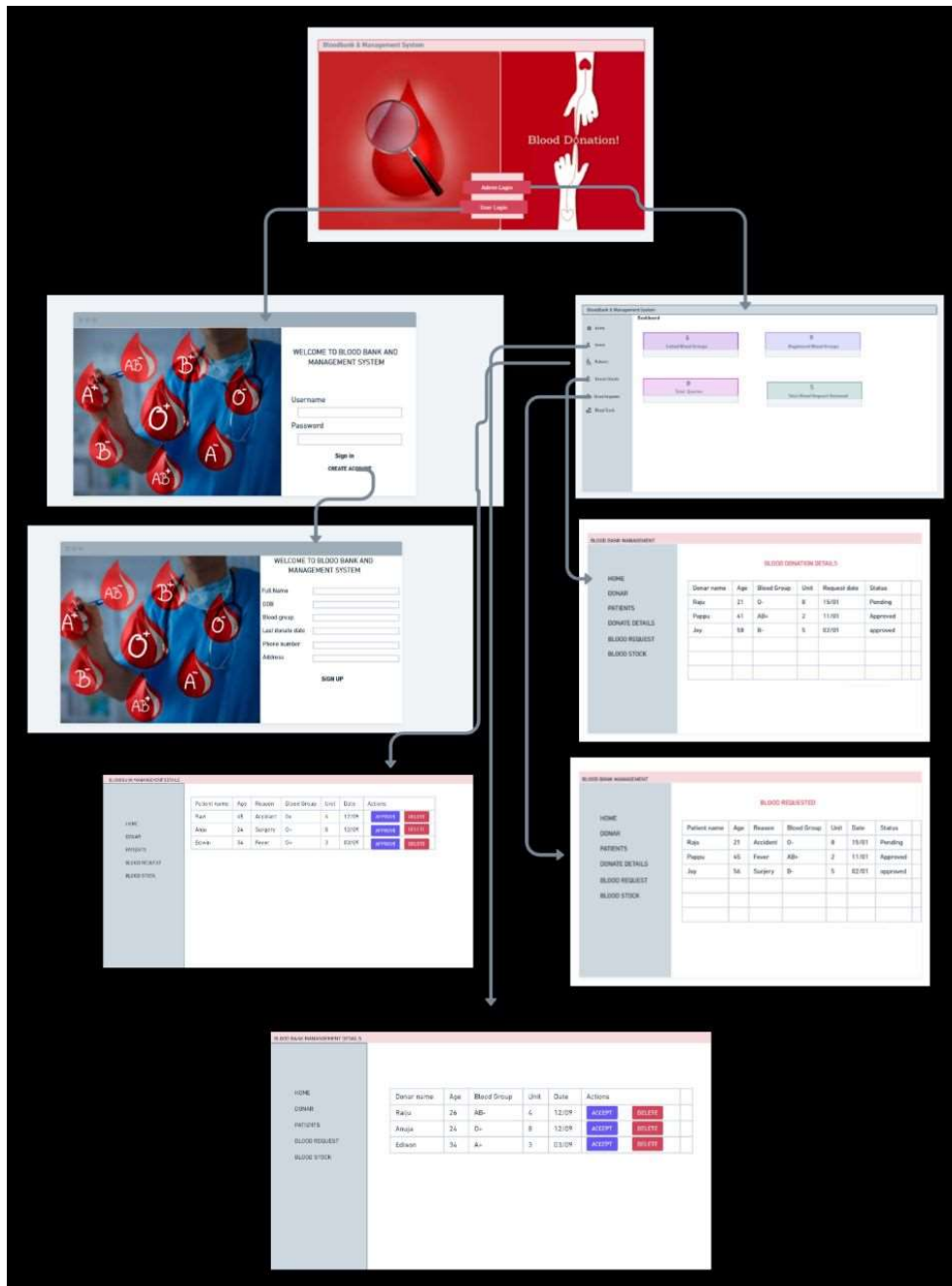
5. Blood Requested Details

BLOOD BANK MANAGEMENT							
BLOOD REQUESTED							
Patient name	Age	Reason	Blood Group	Unit	Date	Status	
Raju	21	Accident	O-	8	15/01	Pending	
Pappu	45	Fever	AB+	2	11/01	Approved	
Joy	56	Surjery	B-	5	02/01	approved	

6. Blood Donated Details

BLOOD BANK MANAGEMENT							
BLOOD DONATION DETAILS							
Donar name	Age	Blood Group	Unit	Request date	Status		
Raju	21	O-	8	15/01	Pending		
Pappu	41	AB+	2	11/01	Approved		
Joy	58	B-	5	02/01	approved		

USER FLOW DIAGRAM



CONCLUSION

The Blood Bank Management Web Application will provide a user-friendly and efficient platform to manage blood inventory, donor information, and transfusion procedures. The application will be developed using HTML, CSS, React.js, and Node.js technologies, with agile methodologies used for development. The database management system will be MongoDB, a NoSQL database system. The application will undergo rigorous testing to ensure that it meets the requirements and is delivered on.