

Scientiarum: A Multidisciplinary Journal
Vol. 2 No. S1 (2026): Supplement Issue
DOI: 10.54646/SAPARS.2026.82

SAPAR SERVICES LLP
Scholarly Articles Publication And Repository Services

A RESEARCH PAPER ON THE DESIGN AND DEVELOPMENT OF A WEB-BASED BLOOD BANK DONOR MANAGEMENT SYSTEM

K ADISESHA

Assistant Professor PRUTHVI GOWDA S

MCA 2nd year

Department of Computer Science and Applications

S.E.A College of Science, commerce and application

Adiseshal@rediffmail.com Pruthvigowdas688@gmail.com

Abstract

The Blood Bank Management System is a Python-based application developed to simplify and automate the management of blood bank activities. The system provides a centralized platform for managing donor information, blood group details, blood inventory, patient requirements, blood requests, and distribution records. It helps maintain accurate information about the availability of different blood groups and makes it easier to manage blood collection and supply. The system reduces manual paperwork, minimizes human errors, and provides quick access to important blood bank records. It can store donor and patient details securely and allows authorized users to add, update, search, and retrieve information whenever required. Database connectivity helps maintain organized and reliable records of blood donations and transactions. The proposed system improves the efficiency, accuracy, and reliability of blood bank operations. Overall, the Blood Bank Management System provides a

simple and user-friendly solution for managing blood resources and ensuring that required blood information can be accessed quickly when needed.

Keywords:

Blood Bank Management System, Python, Blood Donation, Blood Inventory, Blood Groups, Donor Management, Patient Management, Blood Requests, Blood Distribution, Database Management, Flask, SQLite, MySQL, Healthcare Management, Web Application.

INTRODUCTION

The Blood Bank Management System is a Python-based application developed to manage and automate the major activities of a blood bank. Blood banks are responsible for collecting, storing, managing, and distributing blood to patients who require transfusions. Managing donor details, blood groups, available blood units, patient requirements, and blood requests manually can be time-consuming and may lead to errors or difficulty in retrieving important information.

The proposed system provides a centralized platform for maintaining blood bank records efficiently. It allows authorized users to register donors, manage blood group information, monitor available blood units, record blood donations, manage patient details, and process blood requests. The system uses a database to store and retrieve information accurately and securely.

The application can be developed using Python and Flask**,** with HTML, CSS, and JavaScript for the user interface and SQLite or MySQL for database management. By automating routine activities, the system reduces paperwork, minimizes errors, saves time, and improves the overall efficiency of blood bank operations. The system also helps users quickly identify the availability of required blood groups and maintain

organized records for future reference.

PROBLEM STATEMENT

Managing blood bank operations manually can be time-consuming and may lead to errors in maintaining donor details, blood group records, inventory levels, patient information, and blood requests. It can also be difficult to quickly identify the availability of a particular blood group during emergencies. Manual record keeping may result in data duplication, loss of information, and delays in processing blood requests. Therefore, there is a need for a computerized system that can efficiently manage blood donations, maintain accurate blood inventory, store donor and patient information, and provide quick access to required blood group details. The Blood Bank Management System is proposed to overcome these challenges by providing a centralized, database-driven platform that improves the accuracy, speed, and efficiency of blood bank operations.

OBJECTIVES

To develop a computerized system for managing blood bank operations.

To maintain accurate records of blood donors and patients.

To manage different blood groups and available blood units.

To record and manage blood donation details.

To monitor blood inventory efficiently.

To process and manage blood requests.

To provide quick information about the availability of required blood groups.

To reduce manual paperwork and human errors.

To securely store and retrieve blood bank records.

To improve the efficiency and accuracy of blood distribution.

To provide a simple and user-friendly interface for authorized users.

LITERATURE REVIEW

The literature review of the Blood Bank Management System focuses on existing blood bank applications, donor management systems, blood inventory management, and database-based healthcare applications. Traditional blood banks often use manual registers or basic computerized systems to maintain donor details, blood group information, inventory records, and patient requests. Such methods can be time-consuming and may result in data duplication, human errors, and delays in finding the required blood group. Modern blood bank management systems provide features such as donor registration, blood collection records, inventory monitoring, patient management, and blood request processing. Database technologies such as SQLite and MySQL help store and retrieve blood bank information efficiently, while Python and Flask provide a suitable platform for developing a simple web-based management application. Based on the study of existing systems, the proposed Blood Bank Management System integrates donor management, blood inventory, patient details, blood requests, and distribution records into a centralized platform. The system aims to reduce manual effort, improve data accuracy, provide quick access to blood availability information, and increase the overall efficiency of blood bank operations.

PROPOSED SYSTEM

The proposed Blood Bank Management System is a Python-based web application designed to automate and simplify the major activities of a blood bank. The system provides a centralized platform for managing donor information, patient details, blood groups, blood donations, blood inventory, blood requests, and distribution records. Authorized users can register donors, record donations, update available blood units, manage patient requirements, and process blood requests efficiently. The system uses a centralized database such as SQLite or MySQL to store and retrieve information

securely. It helps users quickly check the availability of required blood groups and maintain accurate records of blood collection and distribution. The proposed system reduces manual paperwork, minimizes human errors, saves time, and improves the overall efficiency and reliability of blood bank operations.

Major Modules

Admin/User Management: Manages user login, authentication, and authorized access to the system.

Donor Management: Stores and manages donor details such as name, age, contact information, blood group, and donation history.

Blood Group Management: Maintains information about different blood groups and their availability.

Blood Donation Management: Records blood donation details, including donor information, donation date, blood group, and units collected.

Blood Inventory Management: Monitors available blood units and updates stock when blood is collected or issued.

Patient Management: Maintains patient information and their blood requirements.

Blood Request Management: Records and manages requests for required blood groups and quantities.

Reports and Dashboard: Provides an overview of blood stock, donations, requests, and distribution records.

SYSTEM DESIGN

The Blood Bank Management System is designed as a web-based application that provides a centralized platform for managing blood bank operations. The system follows a structured architecture consisting of the Presentation Layer, Application Layer, and Database Layer**.** The Presentation Layer provides a simple interface for

users to log in, register donors, manage patients, check blood availability, record donations, and process blood requests. The Application Layer, developed using Python and Flask, handles business logic, authentication, inventory updates, blood requests, and other system operations. The Database Layer, using SQLite or MySQL, stores donor details, patient information, blood group records, donation details, inventory data, and distribution records.

When a user performs an operation, such as recording a blood donation, the request is processed by the application and the corresponding blood inventory is updated in the database. Similarly, when blood is issued to a patient, the available stock is reduced and the transaction is recorded. The system includes validation and error-handling mechanisms to maintain accurate and consistent information.

The overall design aims to provide security, reliability, accuracy, usability, and efficient data management. The modular structure makes the system easier to maintain, test, and enhance with additional features in the future.

METHODOLOGY

The Blood Bank Management System follows a systematic software development methodology to design, develop, test, and implement the application. The process begins with identifying the problems associated with manual blood bank management and collecting the functional and technical requirements of the system. Based on these requirements, the system architecture, database structure, user interface, and individual modules are designed.

The application is developed using Python and Flask**,** with HTML, CSS, and JavaScript used for the front-end interface and SQLite/MySQL used for database management. The development process includes implementing modules such as user authentication, donor management, blood group management, blood donation, blood

inventory, patient management, blood requests, and blood distribution..

FUNCTIONAL REQUIREMENTS

User registration, login, and logout management.

Admin authentication and authorized access control.

Add, update, delete, and view donor information.

Maintain donor blood group and donation history.

Record and manage blood donation details.

Manage different blood groups and blood units.

Monitor available blood inventory.

Automatically update blood stock after donation and distribution.

Add, update, search, and view patient information.

Record and manage blood requests.

Check the availability of required blood groups.

Process blood distribution and maintain issue records.

Store blood donation and distribution history.

NON-FUNCTIONAL REQUIREMENTS

Performance: The system should process requests and retrieve blood-related information quickly.

Security: Donor, patient, and blood bank information should be protected from unauthorized access.

Reliability: The system should provide consistent and accurate results during normal operations.

Usability: The application should have a simple and user-friendly interface.

Availability: The system should be accessible whenever blood bank staff need to manage records.

Scalability: The system should support an increasing number of donors, patients, blood units, and transactions.

Maintainability: The application should be easy to maintain, modify, and update.

IMPLEMENTATION

The system begins with user authentication to provide secure access to authorized users. The donor management module is implemented to register and maintain donor information, while the blood donation module records donation details such as blood group, donation date, and quantity. The blood inventory module maintains the availability of different blood groups and updates stock whenever blood is collected or distributed.

The patient management module stores patient details and blood requirements, while the blood request module records and processes requests for required blood units. The distribution module maintains records of blood issued to patients or healthcare facilities. Search and reporting features are implemented to help users quickly find donor information, check blood availability, and view important records.

TESTING

Testing is an important phase in the development of the Blood Bank Management System**.** It is performed to ensure that the application works correctly according to the specified requirements and to identify and correct errors before deployment. Testing verifies the functionality, accuracy, reliability, security, and usability of the system. Different levels of testing are carried out on the application, including Unit Testing, Integration Testing, System Testing, and Acceptance Testing**.** Unit testing checks individual modules such as donor management, blood inventory, patient management, blood requests, and user authentication. Integration testing verifies that these modules work correctly together and communicate properly with the database. System testing

evaluates the complete application to ensure that all functions operate as expected. Acceptance testing verifies that the system meets user requirements and is suitable for practical use.

EXPECTED RESULTS

The expected to provide an efficient, accurate, and user-friendly platform for managing blood bank operations. The system should simplify donor registration, blood donation recording, blood inventory management, patient management, blood requests, and blood distribution. It is expected to provide accurate and updated information about the availability of different blood groups, allowing users to quickly identify required blood units when needed. The system should reduce manual paperwork, minimize human errors, save time, and provide secure storage of donor, patient, and transaction records. It should also enable quick retrieval of information, improve inventory monitoring, and maintain organized records through database management. Overall, the system is expected to improve the efficiency, reliability, and accuracy of blood bank operations and provide a strong foundation for future enhancements.

ADVANTAGES

Reduces manual paperwork and administrative effort.

Saves time in managing blood bank operations.

Provides accurate blood inventory management.

Allows quick checking of blood group availability.

Simplifies donor registration and management.

Maintains complete donor and donation history.

Makes patient information management easier.

Provides efficient blood request and distribution management.

Reduces human errors and duplicate records.

Provides secure and centralized data storage.

Enables quick retrieval of donor, patient, and blood records.

LIMITATIONS

The Blood Bank Management System has some limitations. The system requires a computer and proper database configuration for effective operation. The accuracy of records depends on the information entered by authorized users. Regular database backup and maintenance are necessary to prevent data loss. The system may require technical support for major modifications, errors, or security issues. Advanced features such as integration with hospitals, laboratory equipment, barcode systems, and real-time blood monitoring may require additional hardware and software. The system may also need future updates to meet changing healthcare requirements and operational needs.

CONCLUSION

This Blood Bank Management System provides an efficient and reliable solution for managing the major activities of a blood bank. By using Python, Flask, and database technology, the system helps manage donor information, blood donations, blood inventory, patient details, blood requests, and distribution records in an organized manner. It reduces manual paperwork, minimizes human errors, saves time, and provides quick access to important blood-related information. The system also improves the accuracy and reliability of blood stock management and helps users quickly identify the availability of required blood groups. Although the system has certain limitations, it provides a strong foundation for computerized blood bank management and can be further enhanced with features such as hospital integration, online blood requests, notifications, barcode support, and real-time inventory

monitoring.

REFERENCES

Elmasri, R., & Navathe, S. B. (n.d.). Fundamentals of database systems. Pearson Education.

GeeksforGeeks. (n.d.). Python programming, database management, and web development tutorials. https://www.geeksforgeeks.org/

MDN Web Docs. (n.d.). HTML, CSS, and JavaScript documentation. https://developer.mozilla.org/

Pallets Projects. (n.d.). Flask official documentation. https://flask.palletsprojects.com/

Pressman, R. S. (n.d.). Software engineering: A practitioner's approach. McGraw-Hill Education.

Python Software Foundation. (n.d.). Python official documentation. https://docs.python.org/

SQLite. (n.d.). SQLite official documentation. https://www.sqlite.org/docs.html

TutorialsPoint. (n.d.). Python and Flask web development tutorials. https://www.tutorialspoint.com/