Design and Implementation of Improving Transparency between Doner and Seeker in Blood Bank Management System Implementing on Cloud Computing

Miss. Prachi Umesh Ingole[1], Prof. Krutika. K. Chhajed[2]
[1,2]*P. R. Pote Patil College of Engineering & Management, Amravati*

Abstract: Blood bank is a central repository where blood as a result of blood collection and donation is stored and managed for future use in blood transfusions [1]. There are several online web-based blood bank management systems for data storage for blood center and hospitals to manage donor information, available blood, and transaction information. Recent research on this topic shows that manual systems consume time, tedious and expensive compared to the computerized information system [1]. This is also evident in praising computerization as a mechanism to achieve efficiency and effectiveness in this area and highlighting some critical issues that are left out such as the correct accountability of the system administration. In this paper, we are study about previous work on blood bank management system. No one offers the possibility of direct contact between the taxpayer and the beneficiary. This is tremendous damage, especially in conditions where blood is really needed. This document formed the development of the cloud architecture by giving a cloud database of the blood donation centre that contains entire sights and from various roots as the center of the blood donors, the systems of national service, non-governmental organizations, healing centers and through Webu.

Keywords: Doner, seeker, blood bank, cloud architecture, cloud database.

INTRODUCTION

Every hospital operates within its own framework, governed by unique rules, standards, and constraints. Unfortunately, effective coordination between hospitals and blood donation centers is severely lacking, primarily due to foundational issues. Moreover, the viability and functionality of blood donation centers suffer due to limited taxpayer support and other factors. One significant consequence of this is the mismanagement and wastage of blood and its components. The key problems in the current scenario include:

1. Urban areas are saturated with blood banks, yet face a shortage of donors, whereas provincial (rural) areas lack sufficient blood banks altogether.
2. Rural areas lack adequate infrastructure for blood collection, processing, and storage.
3. Some private healthcare facilities have exclusive blood donation centers.
4. Certain public institutions lack blood banks altogether.
5. Donors lack records of their contributions or relevant information regarding their blood-related conditions.

Furthermore, while some blood donation center databases are accessible, facilitating communication between donation centers and medical facilities, there's a significant gap in enabling direct communication between donors and recipients.

LITERATURE REVIEW

The current blood bank storage system is focused on files. This ensures that data and knowledge about blood, donors, and recipients are stored in documents and archives. Data and information processing becomes difficult and time-consuming as a result of this. All tests of blood donation and transfusion are recorded on physical papers as well. This leaves information vulnerable to gross and human error, which in turn endangers human life. Another underlying problem with this framework is the lack of productivity. Because recovery is such a time-consuming process, it is very difficult for hospitals to save lives at crucial moments. Information security and data backup is another point to consider as documents and records are easily lost or stolen. This makes it an unreliable framework. The goal of our project was to provide a cloud platform containing all information about blood donations and registered donors, which in turn can contribute to fast blood delivery. We have endeavoured to research everything about blood management systems and practices and have used the knowledge to make our project as good as possible**.** Every blood donation management system has to fulfil some basic tasks. You must have a

mechanism in place to make the information sharing available to donors, recipients, and other interested parties. It should also ensure that information on the status of blood inventory is available from various stakeholders such as blood banks and hospitals. It was important for us to find the faults of the existing system so that we could find the solutions to the faults and implement them in our project.

Propose System: This project consists of two sections: an admin section and a blood bank management section. Upon visiting our site, only administrators can log in to the admin section. Administrators have complete access to tasks such as adding blood banks, registering new users, and approving events organized by blood banks, such as blood donation camps and awareness campaigns. Additionally, administrators can approve users to view donor contact details. In the blood bank management section, users have access to functions including adding blood banks, updating blood stock, and organizing events like blood donation camps and awareness campaigns. In blood bank management, transparency between donors and seekers is lacking, hindering efficient communication and resource allocation. Existing systems suffer from limited accessibility, opaque processes, inefficient communication, and data security concerns, exacerbated by the transition to cloud computing. Addressing these challenges requires a comprehensive system leveraging cloud capabilities to enhance transparency, communication, and data security in blood donation and distribution.

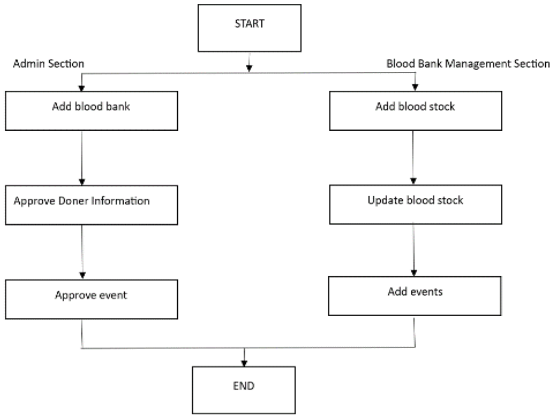

Figure 3.2.1 Workflow diagram of system

System Design

This project has two parts:

- Admin Section: Here, the admin can add new blood banks, update blood stock, add new donors and seekers, and manage blood donation events and awareness campaigns.
- Blood Bank Management Section: This part is for blood banks. They can manage their blood stock, receive updates, and organize donation events and awareness campaigns.

Working Methodology: This algorithm provides a basic outline of how a login system could be implemented. However, in a real-world scenario, additional features such as password hashing, account lockout mechanisms, and multi-factor authentication might be necessary for security purposes.

Algorithm 1: Login to admin section and blood bank management system.

Step 1: Start

Step 2: Prompt the user to enter their username and password. Display a message asking the user to enter their username and password.

Step 3: Read the entered username and password. Capture the input provided by the user.

Step 4. Validate the entered username and password:

a. Check if the username exists in the system.

- Query the system's database or data store to check if the entered username exists.

b. If the username exists, check if the entered password matches the password associated with that username.

- Retrieve the password associated with the entered username from the database.

- Compare the entered password with the stored password.

c. If the entered password matches the password associated with the username, allow the user to login. Grant access to the system. Proceed to step 7.

d. If the entered password does not match the password associated with the username, prompt the user to re-enter the password.

- Display a message indicating that the entered password is incorrect.

- Prompt the user to re-enter the password.

- Repeat step 3.

Step 5: If the entered username does not exist in the system, prompt the user to re-enter the username.

- Display a message indicating that the entered username is not found.

- Prompt the user to re-enter the username.

- Repeat step 3.

Step 6: Repeat steps 3-5 until the user successfully logs in or chooses to exit the login process. Continue

looping until the user successfully logs in or decides to quit the login process.
Step 7: If the user successfully logs in, grant access to the system and display a welcome message.
- Display a welcome message acknowledging the successful login.
- Grant the user access to the system's features and functionalities.
Step 8. Terminate the login process.

This algorithm provides a basic outline of how a login system could be implemented. However, in a real-world scenario, additional features such as password hashing, account lockout mechanisms, and multi-factor authentication might be necessary for security purposes.

Algorithm 2: find nearest location from current location
Step1: Input

- Current location (latitude, longitude)
- List of locations with their latitude and longitude coordinates

Step 2: Initialization:

- Set a variable `nearest_location` to `null`.
- Set a variable `min_distance` to a very large number (initially positive infinity).

Step 3: Iterate Through Locations:
For each location in the list:

- Calculate the distance between the current location and the location in the list using a distance formula (e.g., Haversine formula).
- If the calculated distance is less than `min_distance`, update `min_distance` to this distance and update `nearest_location` to the current location.

Step 4: Output is Return `nearest_location`.
This pseudocode represents the algorithm in a more generalized way. You would need to implement the calculateDistance function using a suitable formula for calculating distances between two points given their latitude and longitude coordinates, such as the Haversine formula.

Algorithm 3: Handling Form Submission, Saving to Database, Sending Emails.
Step 1: Validate Form Data

```
if not validateFormData(form_data):
    return "Error: Invalid form data"
```

Step 2: Save Data to Database

```
db_connection = database_library.connect(db_config)
save Data To Database (db_connection, form_data)
```

Step 3: Send Email to Admin

```
admin_email_content = create AdminEmailContent (form_data)
sendEmail(email_config, "admin@example.com", admin_email_content)
```

Step 4: Reply to User

```
user_email_content = create User Reply Email Content (form_data)
send Email (email_config, form_data['user_email'], user_email_content)
```

Step 5: Return Success Response

```
return "Success: Form submitted successfully"
except Exception as e:
# Handle any errors that occur
logError(e)
return "Error: " + str(e)
```

This algorithm outlines the process for handling a form submission, encompassing data validation, database storage, and email notifications. Initially, necessary libraries for database operations, email sending, and form handling are imported, and configurations for the database connection and email server are set up. When a form is submitted, the main function retrieves and validates the form data to ensure all required fields are correctly filled out. If the data is valid, a connection to the database is established, and the data is inserted into a specific table, with error handling in place to manage any issues during this process.

Implementation Details
This project is divided into 3 sections are as follows:

- User Section: In this section, users have access to pertinent information regarding the closest blood bank, forthcoming blood donation drives, and awareness campaigns. Valid users are also provided with the capability to contact blood donors as needed.
- Admin Section: In this section, the administrator is empowered to include new blood banks, update blood stock, add new donors and seekers. Additionally, there is oversight over registering blood donation and awareness camps, along with

validation checks for user authenticity. Upon entering the Admin Section, administrators are presented with a comprehensive dashboard featuring key functionalities:

1. Add Blood Bank: This option allows administrators to add new blood banks to the system. They can input details such as the blood bank's name, location, contact information, and operational hours, ensuring the database remains current and comprehensive.
2. Update Blood Stock: Administrators can update the inventory of blood products. This includes adding new donations, recording usage, and monitoring stock levels to ensure a balanced supply of all blood types.
3. Approve Donor Information: This section enables administrators to review and approve donor registration requests. They can verify the submitted information for accuracy and completeness, ensuring only eligible donors are registered in the system.
4.Events Information Requests: Administrators can manage and respond to requests for information regarding blood donation events. This includes providing details on upcoming drives, special campaigns, and community outreach programs to promote blood donation.

These functionalities streamline the management of blood bank operations, ensuring efficient and effective service delivery.

- Blood Bank Section:

In this section, the blood bank possesses access to activities pertinent to its operations, such as adding new blood stock, receiving updates on existing stock, and organizing both blood donation events and awareness campaigns. Within the Blood Bank Management system, administrators have access to essential features for effective blood bank operations:

1.Adding Blood Bank: Administrators can seamlessly integrate new blood banks into the system, inputting detailed information such as location, contact details, and operational hours to maintain a comprehensive database.
2.Adding Blood Stock: This functionality empowers administrators to update and manage blood stock levels, ensuring a balanced supply of blood products. Through this feature, they can record donations, monitor inventory, and facilitate efficient distribution.
3.Adding Events and Camps: Administrators can organize and schedule blood donation events and camps through this feature. They can input event details, such as date, time, and location, and coordinate outreach efforts to encourage community participation.
4. Requesting Donor Information: This feature enables administrators to request donor information based on specific criteria such as blood type or location. By efficiently accessing and managing donor data, administrators can facilitate prompt responses to donation needs and ensure effective blood bank operations.

These functionalities collectively streamline blood bank management, ensuring timely and efficient handling of donor information, blood stock, and outreach efforts.

RESULT ANALYSIS

The existing system for a cloud-based blood bank management system allows donors to donate blood based on criteria such as age, weight, haemoglobin levels, the gap between donation dates, and the presence of diseases. The new approach expands this by also considering vaccination times. In terms of design architecture, the existing system includes detailed plans for data storage and user interfaces, whereas the new approach adds communication protocols like email to this architecture.

Both systems ensure real-time data access, allowing parties to view information such as the availability of specific blood types and the status of blood donations. However, the new approach also provides real-time location data for blood donations. Transparency and trust are core to both systems; the existing system builds trust by making the donation process transparent, while the new approach further enhances trust through improved communication between donors and seekers.

Data security and privacy are available in the existing system, but the new approach emphasizes a higher level of availability. System implementation in the existing setup includes an admin panel and a user panel. The new approach broadens this to feature an admin panel, a blood bank management panel, and a user panel. Finally, while the existing system lacks communication between donors and seekers, the new approach facilitates this interaction via email, enhancing connectivity and trust in the donation process.

Advantages:

- Trust and Confidence:
- Improved Resource Allocation
- Enhanced Donor Engagement
- Quality Assurance
- Data-Driven Decision Making
- Community Empowerment
- Health Impact

Limitations

- Data Privacy Concerns
- Cost and Resource Intensiveness:
- Resistance to Change
- Accessibility Issues
- Incomplete Data or Reporting
- Misinterpretation of Information
- Legal and Regulatory Compliance
- Cultural and Social Factors

CONCLUSION

In conclusion, this project represents a significant step towards enhancing transparency within the blood donation process, with the aim of saving lives and improving healthcare outcomes. By implementing innovative solutions such as blockchain integration, mobile health applications, and data analytics, we have laid the foundation for a more accessible, efficient, and accountable blood donation ecosystem. Through stakeholder collaboration, community engagement, and technological innovation, we have addressed key challenges related to trust, resource allocation, and data management, paving the way for sustainable improvements in blood donation practices. While acknowledging the project's accomplishments, we also recognize the ongoing need for continuous evaluation, adaptation, and expansion to realize its full potential and address evolving healthcare needs. Looking ahead, we remain committed to advancing transparency initiatives, advocating for policy reforms, and fostering global partnerships to ensure equitable access to safe blood transfusions for all individuals in need. Together, we can build a brighter future where every donation counts, every life is valued, and every community thrives.

FUTURE SCOPE

- Blockchain Integration
- Mobile Health Applications
- Data Analytics for Demand Forecasting
- Remote Blood Donation Services
- Global Blood Donation Network

REFERENCE

[1] Anju Khandelwar, “Cloud Computing Architecture for Improving Transparency between Donor/Seeker in Blood Bank Management System” published in the Proceedings of the 12th INDIACom-2018; IEEE Conference ID: 42835; ISSN 0973-7529; ISBN 978-93-80544-28-1.

[2] Selvamani K., Ashok Kumar Rai, “A Novel Technique for online Blood Bank Management”, Procedia Computer Science, Vol. 48, Pg. 568-573, 2015, Doi: 10.1016/j.procs.2015.04.137.

[3] Ibrahim Fawze Akar, Tukur Anas Mohammad, Mohamed Ismail Z, “CBBR Centralized Blood Bank Repository”, international Journal of Information System and Engineering Vol. 3 (No.1), April, 2015 ISSN: 2289-7615 DOI: 10.24924/ijise/2015.11/v3.iss1/85.97.

[4] Muhammad Nur Sahid Ramadhan1*, Arinda Amyus 1 , Ahmad Nurul Fajar1 , Sfenrianto Sfenrianto1 , Abiyyu Fawwaz Kanz1 , Moh Sukron Mufaqih1, “Blood Bank Information System Based on Cloud Computing In Indonesia”, IOP Conf. Series: Journal of Physics: Conf. Series 1179 (2019) 012028.

[5] Kudipudi Geetha Prasanna, Padala Srinivasa Reddy, “Emergency Blood Bank Management System”, JETIR May 2020, Volume 7, Issue 5, www.jetir.org (ISSN-2349-5162).

[6] Subrata Talapatra, Raihnul Kabir,Akash Buppy, “ Development of an Online Blood Management System”, Proceedings of the International Conference on Industrial Engineering and Operations Management Bangkok, Thailand, March 5-7, 2019.

[7] Shyam Sundaram and Santhanam T., “Real Time Blood Donar Management using Dashboards based on Data Mining Model”, International Journal of Computer Science Issues, Vol. 8, Issue 5,No. 2, Pg. 159-163, September 2011, ISSN:1694-0814.

[8] Vikas Kulshreshtha & Sharad Maheshwari, “Blood Bank Management Information System in India”, International Journal of Engineering Research and Applications (IJERA), Vol. 1, Issue 2, Pg.260-263, ISSN: 2248-9622.

[9] A. Clemen Teena, K. Sankar and S. Kannan, “A Study on Blood Bank Management”, Department of MCA, Bharath University, Selaiyur, Chennai-73,

Tamil Nadu, India, Middle-East Journal of Scientific Research 19 (8): 1123- 1126, 2014 ,ISSN 1990-9233,DOI: 10.5829/idosi.mejsr.2014.19.8.11202