**CHAPTER 3**

**METHODOLOGY**

In this chapter, the methodologies used throughout the study are described together with the diagram presentation. Here are the following sections that will be discussed in this chapter: Requirements Analysis, Requirement Documentation, Design of Software, System Product and/or Process, Development and Testing and Implementation Plan.

**REQUIREMENTS ANALYSIS**

The Red Cross Muntinlupa Chapter faces significant challenges in managing blood donations due to manual appointment scheduling and communication gaps between donors and recipients.

Red cross Muntinlupa Chapter

User / Beneficiaries

User / Beneficiaries

Donor Inquiry

Blood request

Appointment Booking

Donate Blood

Request Blood

Check inventory for matching blood type

Receive inquiries via phone or in-person

Receive inquiries via phone or in-person

Receive blood requests via phone or email

Donation center

Donation Inquiry

Ensure compatibility

Provide waver to fill out

Provide information about donation process and requirements

Verify requester’s credentials

Fill out registration forms

Inquire through phone call or email

Schedule appointment in paper log or basic digital tool

Confirm appointment details with donor via phone or email

Prepare blood units for dispatch

Inform requester through email or phone call

Wait for available blood type

Submit to staff

Donate blood depends if there's an available nurse

Go to center to pick up

Conduct preliminary health check

Arrange for pickup

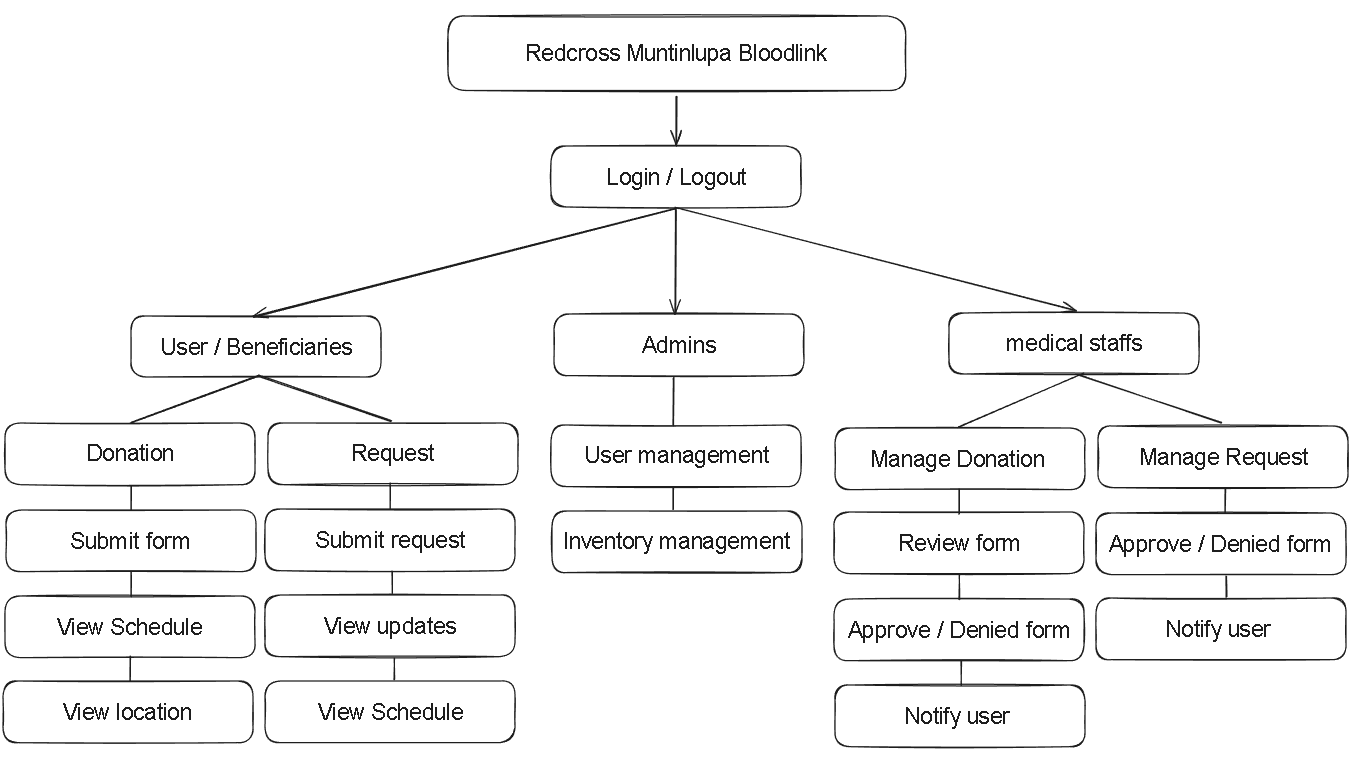
Verify donor eligibility

*Figure 4.* Functional Decomposition Diagram of Redcross Muntinlupa Blood Request Existing Process and Communication Between Donors and Recipients

The Red Cross Muntinlupa Center's appointment and blood request processes are inefficient and prone to errors due to their reliance on phone calls, in-person visits, and record-keeping. Donors contact the center to schedule appointments, and staff check available slots and record information, leading to delays and higher no-show rates due to the lack of automated reminders. Similarly, recipients request blood by phone or in-person, with staff verifying inventory and coordinating pickups or deliveries. If the required blood type is unavailable, recipients are placed on a waiting list, and staff reach out to potential donors. This outdated communication method causes delays and hinders quick donor mobilization. An automated system is needed to enhance efficiency, reduce errors, and improve donor and recipient engagement.

**REQUIREMENTDOCUMENTATION**

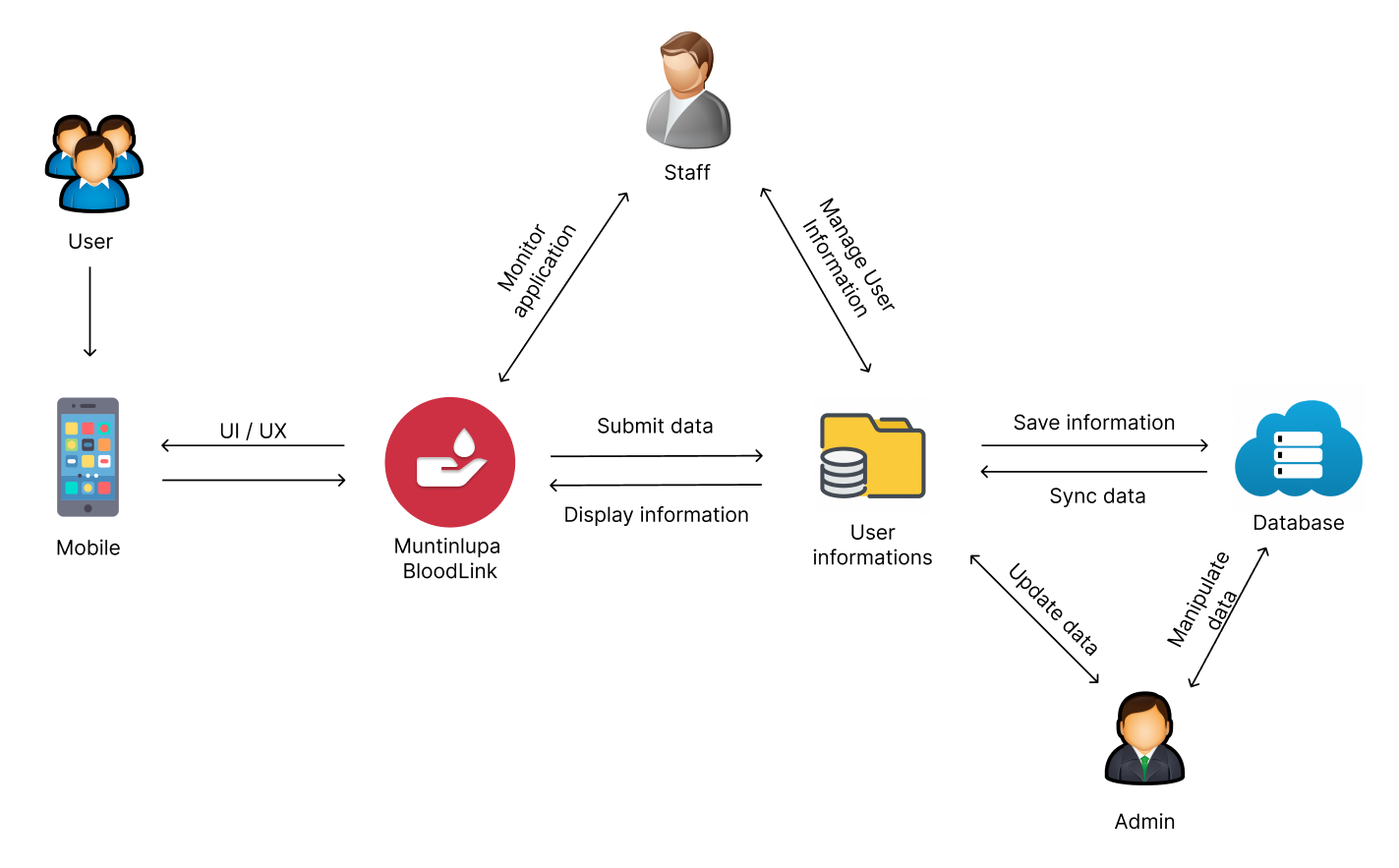
The Functional Decomposition Diagram (FDD) for the proposed system at the Red Cross Muntinlupa Center outlines key functions to streamline and improve blood donation management. The system includes automated appointment scheduling through a user-friendly mobile app, efficient blood collection procedures with integrated health screenings, and real-time blood inventory management. Enhanced communication features ensure timely notifications and coordination between donors and recipients. This technology-driven approach aims to increase efficiency, reduce errors, and improve the overall blood donation experience.



*Figure 5.* Functional Decomposition Diagram of the Proposed System

Figure 5 illustrates the Redcross Muntinlupa Bloodlink system’s architecture, mapping out the essential functions for various user roles such as ‘User/Beneficiaries,’ ‘Admins,’ and ‘Medical Staffs.’ It details the system’s access points through ‘Login/Logout,’ delineates the donation and request procedures for users, outlines administrative control over user management and inventory, and describes the medical staff’s responsibilities in approving forms and facilitating communication. This hierarchical diagram accentuates the system’s structured operational workflow and the specific duties assigned to each user group.

**System Architecture**



*Figure 6.* System Architecture of the proposed Application

figure 6 illustrates the comprehensive system architecture of the proposed application, designed to facilitate seamless interaction between users, staff, and system administrators. The architecture is depicted as a flowchart, emphasizing the integral components and their inter connectivity.

**User Interface (UI)/User Experience (UX)** component is represented, indicating the point of contact for users through mobile devices. This interface is directly linked to the Muntinlupa BloodLink module, which serves as the core platform for users to submit and retrieve information.

**Staff** module is shown to have bidirectional communication with both the data submission and information display functions, signifying their role in managing and overseeing these processes.

**Admin** , which maintains oversight over user information and system operations. It is connected to a Database, which is responsible for the storage, synchronization, and updating of user profiles and system data.

**Save Information**, **Sync Data**, and **Update User Profile** , showcasing the dynamic capabilities of the system to manage and process data efficiently.

**DESIGN OF SOFTWARE, SYSTEMS, PRODUCT AND/OR PROCESS**

On the basis of foregoing concepts, theories and findings of related literature, Studies and insights taken from them a concept model was develop as shown below. It shows in the input the requirements used in the application by the developers such as knowledge algorithm, software requirements and the hardware requirements and the user will used it also includes the methodology and the researcher.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | **Knowledge Requirements**  - Database Management  - Mobile Development  - Backend Devepment and APIs  - Software tools  - Frontend Development and frameworks  - Mobile App Development platform | | **Software Requirements**  - Visual Studio  - Git/Github  - JavaScript  - Node.js  - Expo Go  - MySql  - Nativewind  - ChatGPT | | **Hardware Requirements**  - Computer  - Internet Connection  - Mobile Devices | |  |   **INPUT** | **PROCESS**   |  | | --- | | **RAD Methodology**  - Requirements Analysis and Planning  - Software Architecture and Design  - System Construction  - Cutover | | **OUTPUT**   |  | | --- | | **Blood Link: A Mobile-Based Hybrid Blood Bank For Red Cross Muntinlupa Chapter** | |
| |  | | --- | | **Evaluation**  (ISO/IEC 2501:2011) | | | | | |

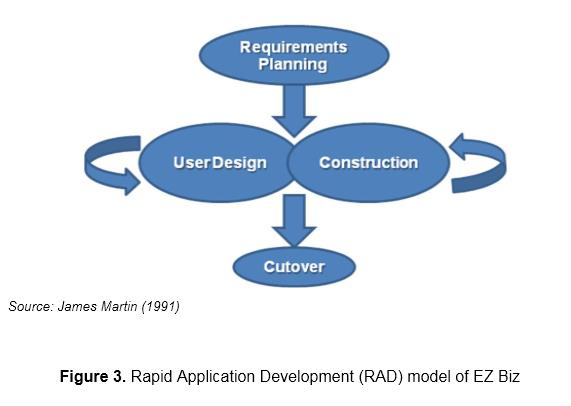
*Figure 7.* Conceptual Model of the Study

The model outlines a study’s workflow, starting with INPUT requirements such as knowledge and software tools, moving through a PROCESS guided by RAD methodology, and culminating in an OUTPUT that includes a mobile-based blood bank application and its evaluation.

**DEVELOPMENT AND TESTING**

**Project Development**

The researchers use the Rapid Application Development (RAD) model is based on prototyping and iterative development with no specific planning involved. The process of writing the software itself involves the planning required for developing the product. Rapid Application development focuses on gathering customer requirements through workshop or focus groups, early testing of prototypes by the customer using interactive concepts reuse of existing prototypes(components) continuous integration and rapid delivery.



*Figure 8*.Rapid Application Development (RAD) Methodology

**Phase 1: Requirements Planning**

The researcher first gathers data and plans to develop the system using React Native for the application’s development. In any case, probably in proposing a system the developers should have talked about the mechanics of the plans first before doing the system. The developers should know the problem of the users and even more in the selected beneficiary. The researchers identify that problem and plans its best solution. To develop the application, the researcher gathered/prepared all the requirements to start the application development.

**Phase 2: Design phase**

The design phase focuses on translating requirements into detailed design specifications. User interface designs, system architecture, and database schema are established during this phase, ensuring a clear roadmap for implementation.

React Native is chosen as the development framework for the mobile application interface due to its ability to provide cross-platform capabilities. This allows the team to develop a single codebase that can run on both iOS and Android devices, streamlining development efforts and reducing time-to-market. React Native's component-based architecture facilitates the creation of reusable UI elements, ensuring consistency across different screens and functionalities within the blood bank application.

**Phase 3: Construction**

During the construction phase, the actual development of the mobile-based hybrid blood bank system takes place. Using the React Native framework, the researchers implement features such as donor registration, appointment scheduling, real-time inventory updates, and communication tools. Continuous integration allows for rapid prototyping and iterative development.

**Phase 4: Cutover Phase**

The cutover phase involves transitioning from the existing blood bank system to the new mobile-based hybrid system. Data migration, user training, and system testing are conducted to ensure a smooth transition. Stakeholder engagement and feedback play a crucial role in validating system functionality and usability.

**Testing Procedure**

When testing the application to find out where the system fault occurs, a testing technique is required. Even while it cannot ensure that all flaws in a program will be eliminated, if the software is tested effectively and precisely, it can minimize the amount of defects to the absolute minimum. To ensure that a program and the system it controls can operate as intended, program tests were carried out in a predetermined order.

**Table 1**

*Alpha Testing*

|  |  |
| --- | --- |
| **Aspect** | **Alpha Testing** |
| Purpose | Identify and fix bugs, enhance features, and ensure the system meets requirements |
| Participants | Developers, testers, and selected internal users |
| Scope | All major features including Database Connection |
| Focus Areas | Functionality, usability, performance, and security |
| Duration | Typically shorter, lasting a few weeks |
| Feedback | Collected from development team |
| Issue Resolution | Immediate and iterative, as issues are reported and fixed quickly |
| Output | Improved and more stable version of the system |

Alpha testing phase in software development, highlighting its purpose, participants, focus areas, scope, duration, issue resolution, and the expected output of an improved system.

**Table 2**

*Beta Testing*

|  |  |
| --- | --- |
| **Aspect** | **Alpha Testing** |
| Purpose | Validate the system's performance, usability, and reliability in real-world scenarios |
| Participants | A broader group of actual end-users |
| Scope | Real-world usage of the system by actual users |
| Focus Areas | User experience, system performance, and overall reliability |
| Duration | Typically longer, lasting several weeks to months |
| Feedback | Collected from actual end-users |
| Issue Resolution | Issues are documented for post-beta fixes and improvements |
| Output | Final adjustments and fixes leading to the official release |

Beta testing phase in software development, which includes validating system performance, usability, and reliability with a broader group of end-users, focusing on user experience and system reliability over several weeks to months, leading to the final product release.

**Project Evaluation**

The evaluation instrument was based on the characteristics and sub-characteristics provided by ISO/IEC 25010:2011. Respondents of the study were the individual users of the system such as the 10 Information Technology (I.T.) experts and 20 actual users. Purposive sampling was used to select the number of respondents. Purposive sampling is a non-probability sampling technique; it is a form of sampling in which the selection of the sample is based on the judgment of the researchers as to which subjects.

**Table 3**

*System Evaluation Characteristics ISO/IEC 25010:2011 Software Evaluation*

*for both Users and IT Experts*

|  |  |
| --- | --- |
| **Software Characteristics** | **Description** |
| Functionality Suitability | Degree to which a product or system provides functions that meet stated and implied needs when used under specified conditions. |
| Performance Efficiency | Performance relative to the amount of resources used under stated conditions |
| Compatibility | Degree to which a product, system or component can exchange information with other products, systems or components and/or perform its required functions, while sharing the same hardware or software environment |
| Usability | Degree to which a product or system can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use |
| Reliability | Degree to which a system, product or component performs specified functions under specified conditions for a specified period of time |
| Security | Degree to which a product or system protects information and data so that persons or other products or systems have the degree of data access appropriate to their types and levels of authorization |
| Maintainability | Degree of effectiveness and efficiency with which a product or system can be modified by the intended maintainers |
| Portability | Degree of effectiveness and efficiency with which a system, product or component can be transferred from one hardware, software or other operational or usage environment to another |

The statistical tool were used in the interpretation of data is weighted arithmetic mean as shown at table 3. Arithmetic mean is used to determine the average responses of the five option in each item, namely, 5(excellent), 4(very good), 3(good), 2(fair) and 1(poor). The arithmetic means for each software characteristics were computed. These were used to derive the overall evaluation mean.

## Table 4

*Likert Scale*

|  |  |  |
| --- | --- | --- |
| **Scale** | **Range of Mean Value** | **Interpretation** |
| 5 | 4.51 – 5.00 | Excellent |
| 4 | 3.51 – 4.50 | Very Good |
| 3 | 2.51 – 3.50 | Good |
| 2 | 1.51 – 2.50 | Fair |
| 1 | 1.00 – 1.51 | Poor |

**IMPLEMENTATION PLAN**

**Table 5**

*Implementation Table*

|  |  |  |  |
| --- | --- | --- | --- |
| **Strategy** | **Activities** | **Persons Involved** | **Duration** |
| Approval from the company | Send letters for the approval of Administrators | Researchers, Administrator | 1 – 2 Days |
| System  Installation | Installation of the system and checking of the facility that needs an upgrade (software and hardware). | Researchers, Administrator | 2 - 3 Days |
| Information Distribution | Send Flyers, Brochures,  Posters, and User Manual | Researchers, Administrator | 1 Day |
| 3- Day  Training | Hands-on Training and System Demo/ Lectures | Researchers, Adviser,  Officer and  Administrator | 3 Days |

After finalizing the Blood Link system, it will be presented to the Red Cross Muntinlupa Chapter for evaluation. If approved for implementation, the complete system and its documentation will be submitted to the Red Cross Muntinlupa Center for deployment.