**Academia International College**

Tribhuvan University

Institute of Science and Technology



**Project Report**

**On**

**“Blood Bank Management System”**

**Submitted To:**

Department of Computer Science and Information Technology

Academia International College

In partial fulfilment of the requirement for a Bachelor Degree in Computer Science and Information Technology

**Submitted By:**

Asish Khadka (15826/074)

Sagar Timalsena (15847/074)

Shyam Baral (15853/074)

Subodh Bhandari (15855/074)

March 2022

# **ACKNOWLEDGEMENT**

Numerous individuals provided us valuable assistance and guidance during the project work and for preparing this report. Our project would not have been possible without the kind support and help of many individuals and organizations. We would like to extend our gratitude to every one of them.

We are thankful to Academia International College for constant guidance and supervision, as well as for providing all the necessary information and a friendly environment for the successful completion of the project.

We would like to express our gratitude to our project supervisor Mr Ganesh Ram Suwal who took an interest in our project and guided us throughout the project by providing all the necessary ideas, information and knowledge for developing a functional web application.

We are thankful to our friends, seniors and every teaching staff of B.Sc. CSIT department for their guidance and cooperation during the development of our project. We would also like to extend our regards to all the non-teaching staff of B.Sc.CSIT department for their timely support.

# **ABSTRACT**

At present, the public can only know about the blood donation events through conventional media means such as radio, newspaper or television advertisements. Without an automated management system, there are also problems in keeping track of the actual amount of every blood type in the blood bank. Also, the donors are not notified about the events happening in their region. Blood Bank Management System (BBMS) is a web-based application system designed to store, process, retrieve and analyze administrative and inventory management within a blood bank. This project aims to maintain all the information about blood donors, various blood groups available in each blood bank and to help them handle it in a better way. The software is fully integrated with CRM (customer relationship management) as well as CMS (content management system) solutions. It is developed in a manner that is easily manageable, time-saving and relieving one from manual work.

***Keywords:*** *Blood bank, CMS, CRM*

**Table of Contents**

[ACKNOWLEDGEMENT i](#_Toc98581541)

[ABSTRACT ii](#_Toc98581542)

[List Of Figure v](#_Toc98581543)

[List Of Table vi](#_Toc98581544)

[ABBREVIATION vii](#_Toc98581545)

[CHAPTER 1 1](#_Toc98581546)

[1. Introduction 1](#_Toc98581547)

[1.1 Statement of Problem 1](#_Toc98581548)

[1.2 Objectives: 2](#_Toc98581549)

[1.3 Scope: 2](#_Toc98581550)

[1.4 Limitation: 3](#_Toc98581551)

[CHAPTER 2 4](#_Toc98581552)

[2. Methodology 4](#_Toc98581553)

[2.1 Literature Review 4](#_Toc98581554)

[2.2 Developmental methodology 6](#_Toc98581555)

[2.2.1 System Development Life Cycle (SDLC) 6](#_Toc98581556)

[2.2.2 Incremental Model 7](#_Toc98581557)

[CHAPTER 3 9](#_Toc98581558)

[3. System Analysis 9](#_Toc98581559)

[3.1 Requirement Analysis: 9](#_Toc98581560)

[3.1.1Functional Requirement: 9](#_Toc98581561)

[3.1.2 Non-Functional Requirement 11](#_Toc98581562)

[3.2 Feasibility Analysis 12](#_Toc98581563)

[3.2.1 Technical Feasibility 12](#_Toc98581564)

[3.2.2 Operational Feasibility 12](#_Toc98581565)

[3.3 Data Model 13](#_Toc98581566)

[3.4 Process Model 14](#_Toc98581567)

[CHAPTER 4 22](#_Toc98581568)

[4. System Design 22](#_Toc98581569)

[4.1. System Design 22](#_Toc98581570)

[4.1.1. Architectural Design 22](#_Toc98581571)

[4.1.2. Database Design 23](#_Toc98581572)

[4.1.3. Data Dictionary 23](#_Toc98581573)

[4.1.4 Flowchart Diagram 25](#_Toc98581574)

[CHAPTER 5 27](#_Toc98581575)

[5. System Development 27](#_Toc98581576)

[5.1 System Development Tools 27](#_Toc98581577)

[5.1.1 Front End 27](#_Toc98581578)

[5.1.2 Back End 29](#_Toc98581579)

[CHAPTER 6 31](#_Toc98581580)

[6. Implementation and Testing 31](#_Toc98581581)

[6.1 Implementation 31](#_Toc98581582)

[6.1.1 Software for system implementation 31](#_Toc98581583)

[6.1.2 Hardware for the System Implementation 32](#_Toc98581584)

[6.2 Testing 32](#_Toc98581585)

[6.1.2 Test Case Design 33](#_Toc98581586)

[CHAPTER 7 34](#_Toc98581587)

[7. Conclusion and References 34](#_Toc98581588)

[7.1 Conclusion 34](#_Toc98581589)

[7.2 Future Work 34](#_Toc98581590)

[References 35](#_Toc98581591)

[Snap Shots 36](#_Toc98581592)

# **List Of Figure**

[Figure 1 phases of SDLC 7](#_Toc98581480)

[Figure 2.2 ER Diagram 13](#_Toc98581481)

[Figure 3 context level diagram 14](#_Toc98581482)

[Figure 4 Donor DFD Level 1 15](#_Toc98581483)

[Figure 5 Admin DFD diagram Level-1 16](#_Toc98581484)

[Figure 6 Donor DFD Level 1 17](#_Toc98581485)

[Figure 7 Organization DFD diagram Level-1 18](#_Toc98581486)

[Figure 8 Admin DFD diagram Level 19](#_Toc98581487)

[Figure 9 Organization DFD diagram Level-2 20](#_Toc98581488)

[Figure 10 Hospital DFD diagram Level-2 21](#_Toc98581489)

[Figure 11 3-Tier Architecture 22](#_Toc98581490)

[Figure 12 Database Design 23](#_Toc98581491)

[Figure 13 Flowchart Diagram for Admin and organization 25](#_Toc98581492)

[Figure 14 Flowchart Diagram for Donar and organization 26](#_Toc98581493)

[Figure 15 Gantt Chart 36](#_Toc98581494)

[Figure 16 Admin Login 37](#_Toc98581495)

[Figure 17 Donor Registration 37](#_Toc98581496)

[Figure 18 Home Page 38](#_Toc98581497)

# **List Of Table**

[Table 1 Donor Registration 24](#_Toc98581418)

[Table 2 Users Table 24](#_Toc98581419)

[Table 3 Blood Group 24](#_Toc98581420)

[Table 4 Software tools used for System Development 31](#_Toc98581421)

[Table 5 Software for System Deployment (Server) 31](#_Toc98581422)

[Table 6 Hardware for System Implementation 32](#_Toc98581423)

[Table 7 Test case for Login 33](#_Toc98581424)

[Table 8 Test case for Signup 33](#_Toc98581425)

[Table 9 Test Case for Request for Blood 33](#_Toc98581426)

# **ABBREVIATION**

BBMS Blood Bank Management System

CMS Content Management System

CSS Cascading Style Sheet

DBMS Database Management System

DFD Dataflow Diagram

DOB Date of Birth

ER Entity Relationship

HTML Hyper Text Markup Language

OO Object Oriented

PHP Hypertext Preprocessor

SDLC Software Development Life Cycle

SQL Structured Query Language

# **CHAPTER 1**

# **Introduction**

Blood is universally recognized as the most precious element that sustains life. It saves innumerable lives across the world in a variety of conditions. Blood Bank Management System is a web-based platform-independent system that can assist the information of blood pouch during its handling in the blood bank. Blood Bank Management System is to be used by blood banks to advertise the nationwide blood donation events to the public and at the same time allow the public to make online reservations and request blood. The purpose of the blood bank management system is to allow requests for blood and maintain the records of blood donors, blood donation programs, blood requests and blood stocks in the blood bank.

From this system, there are many types of reports that can be generated such as bloodstock reports and the total blood donation according to fortnight, months and year. If the requested blood type is not available in the bloodstock, the system will notify donors for emergency purposes. This system helps to notify donors of the next blood donation time.

## **Statement of Problem**

At present, the public can only know about the blood donation events through conventional media means such as radio, newspaper or television advertisements. There is no information regarding the blood donation programs available on any of the portals.

The current system that is used by the blood bank is a manual. With the manual system, there are problems in managing the donors' records. The records of the donor might not be kept safely and there might be missing donor's records due to human error or disasters. Besides that, errors might occur when the staff keeps more than one record for the same donor.

There is also no centralized database used to keep the donors' records. Each bank is having its records of donors. If a donor donates to a different hospital, no previous records can be traced except if the donor brings along the donation certificate. Hence, the donor is considered to be a first-timer if they make a blood donation in a new place.

Without an automated management system, there are also problems in keeping track of the actual amount of every blood type in the blood bank. In addition, there is also no alert available when the blood quantity is below its par level or when the blood in the bank has expired.

## **Objectives:**

The project is programmed to help humans or patients who are seeking blood at a particular location. This PHP project is designed in such a way that it keeps detailed information as well as separate information of all the locations where the blood is available and what kind of blood is available and in how much quantity.

The system does not store blood but it stores the information about the blood or more precisely we can say it store the information or database of the blood available in the particular location. Because there was a time when someone needed blood urgent, then this software proved to be his best friend and help the person find the place nearby him very quickly.

## **Scope:**

The targeted audiences of our system are:

* Organization
* Donor
* Hospital

The scopes of our system are:

i. **View information online:** Donors are allowed to view their profiles.

ii. **Keeping donation records:** This system can provide efficient donor and bloodstock management functions to the blood bank through the recordings of the donors and blood details.

iii. **Manage blood inventory:** The administrator is responsible for removing the bloodstock from the inventory and updating the expired system.

iv. **Blood requests:** The user and hospital can request blood through our system.

v. **Registration:** This function allows the administrator to register the organizations and hospitals, the organization to register events and donors.

## **Limitation:**

The limitation of our system is:

1. User role permission model is not used.
2. Donors are unknown about events time they have participated in.
3. Donors can find blood donation events in their region only. They are unable to find events nearby them i.e. local-based.
4. Only expired blood is deleted in our system.
5. The status of the request for blood should be edited by the admin itself. It is not done by the system automatically.

# **CHAPTER 2**

# **2. Methodology**

## **2.1 Literature Review**

Several studies have been carried out in the definition of blood banks, some of them concerned with maintaining donor records to promote the donation process, the others worked on supporting blood banks in one system and one database, while the others used modern technology such as electronic card and barcode network. Blood Bank Management System is an information system that helps to manage the records of donors and users at a blood bank. It is mainly designed to store, process, retrieve and analyze information concerned with the administrative and inventory management within a blood bank [1]. Dr Sharad Maheshwari's article in the International Journal of Engineering Researcher and Technology (IJERA) states that the blood bank information system MIS is an integrated blood automation system in India. The web- process interconnects all the blood banks of each state into one network. The blood bank refers to all collection, testing, storage, and dissemination of specific live data and electrical knowledge about blood donation and transfusion service [2]

Hamro Life Bank is a non-profit organization that offers a smart, straightforward, and holistic service for blood management from collection to delivery. The organization allows donors to register on their website, contact for blood needs, also conducts social awareness campaigns. Hamro Life Bank, 2020 has made a helpful contribution to eliminating blood shortage in NEPAL by spending your time to make an impact, saving your blood lives or creating a holistic process of blood management with your money HLB works closely with blood banks to maintain their information and also recruit, engage and retain donors as per the demand. Folks in search of blood can get access to blood availability info. It works with blood banks to record information of blood and its supply digitally for easy access [3].

Similarly, the Blood Bank Nepal website has the facility to scan blood about blood groups and locations. On the other hand, when going through different searches, it is found that their blood database is not updated, with just a few records from Nepal's few districts [4]. Its current framework is not successful and does not provide the operating performance for the country.

Nepal's government has appointed the Nepal Red Cross Society as the sole authority in conducting blood programs in Nepal, in its 1991 policy declaration. The Nepal Red Cross Society, with the support of the British Red Cross, has established an innovative new blood bank (Inauguration of Emergency Blood Bank) as part of a huge project to help the Kathmandu Valley and its people start preparing for the earthquake. The blood bank will be operational, ensuring a supply is available the second a disaster happens. The blood bank is of unique structural nature and safe from earthquakes. This is environmentally friendly and has several choices for power supply. It can store over 1600 blood bags at a time [5]

“Blood Bank Management System” done by Kumar, R., Singh, S. and Ragavi, V.A.(2017), the researchers developed web-based blood management which assists the blood donor records management, and provides ease of control in the distribution of blood products in various parts of the country considering demands of hospitals [6]. The developed system was scalable and adaptable to meet the complex needs usually of a blood bank. Based on this study, since entering the details about the blood donors and related records were done manually, thus, tracking of blood donation activities was complicated, and even led to erroneous information. Subsequently, the researchers mentioned that manual-based system can be waste of time, lead to error-prone results, consumes a lot of manpower, lacks data security, data retrieval requires a lot of time, reports consume a long time to produce, and there is less. Precise accuracy on the results. As such, by developing and implementing a web-based blood management information system, there was quick and timely access to donor records, and the system provided management timely, confidential and secured medical reports There are many other similar organizations or systems trying to provide blood facilities to needy ones but are being unsuccessful because of the factors such as low awareness programs, negative rumours regarding blood donation, motivational campaigns, outdated systems, unsuitable blood management systems etc. [7].

## **Developmental** **methodology**

### **System Development Life Cycle (SDLC)**

System Development Life Cycle (SDLC) is a conceptual model which includes policies and procedures for developing or altering systems throughout their life cycles.

SDLC is used by analysts to develop an information system. SDLC includes the following activities −

* Requirements
* Design
* Implementation
* Testing
* Deployment
* Operations
* Maintenance

**Phases of SDLC**

Systems Development Life Cycle is a systematic approach that explicitly breaks down the work into phases that are required to implement either new or modified Information systems.

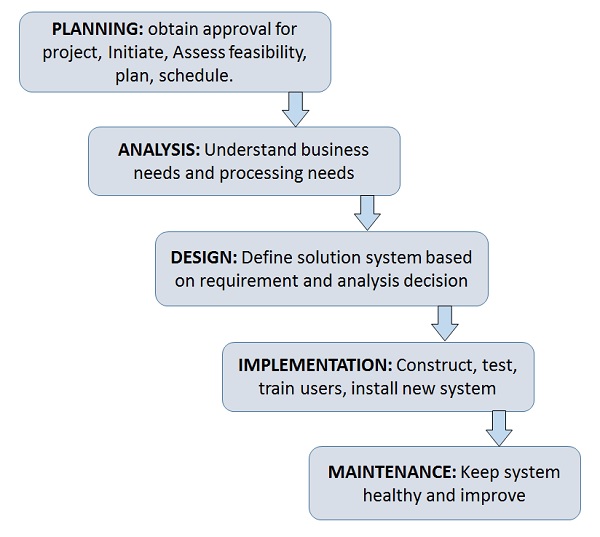


Figure 1 phases of SDLC

### **Incremental Model**

System Development Life Cycle (SDLC) is a conceptual model which includes policies and procedures for developing or altering systems throughout their life cycles.

SDLC is used by analysts to develop an information system. SDLC includes the following activities −

* Requirements
* Design
* Implementation
* Testing
* Deployment
* Operations
* Maintenance

**2.2.2.1 The various phases of the incremental model are as follows:**

* **Requirement analysis:** In the first phase of the incremental model, the product analysis expertise identifies the requirements. And the system functional requirements are understood by the requirement analysis team. To develop the software under the incremental model, this phase performs a crucial role.
* **Design & Development:** In this phase of the Incremental model of SDLC, the design of the system functionality and the development method are finished with success. When software develops new practicality, the incremental model uses the style and development phase.
* **Testing:** In the incremental model, the testing phase checks the performance of each existing function as well as additional functionality. In the testing phase, various methods are used to test the behaviour of each task.
* **Implementation:** The implementation phase enables the coding phase of the development system. It involves the final coding that is designed in the designing and development phase and tests the functionality in the testing phase. After completion of this phase, the number of the products working is enhanced and upgraded up to the final system product.

**Advantage of Incremental Model**

* Errors are easy to be recognized.
* Easier to test and debug
* More flexible.
* Simple to manage risk because it is handled during its iteration.
* The Client gets important functionality early.

# **CHAPTER 3**

# **3**. **System Analysis**

## **3.1 Requirement Analysis:**

### **3.1.1Functional Requirement:**

A functional requirement is a system-specific requirement that the system must do. It represents the main functions of the system. Since the system needs to handle a lot of donor information, a network login function is essential for ensuring security. Users who do not have the correct access rights will be prevented from connecting to the database. The blood bank Management System has different roles like Admin, organization, donor and hospital each with different permissions.

* Admin: acts as the system administrator and can perform all the functions.
* Donor: can view the event information available on the system, participate in events.
* Organization: manages events and donors.

**Modules of admin side:**

1. **User login:**

* At first the user login interface will appear once the application is opened.
* Then admin can log in by using the username and password.
* After successful login, the user will be redirected to the dashboard. A successful attempt will be informed with a login successful message.

2. **Organization Module:**

* The admin will be able to manage the organization and its related information.

1. **Blood request and bloodstock module:**

* The admin will be able to manage the blood request and bloodstock information.

**Modules of Donor Side:**

**User login:**

* At first, the donors will have to log in through the website if they want to view their profile.
* Then donor can log in by using the username and password.
* After successful login, the user will be able to view their profile.

**Profile View:**

* The donor will be able to view their profile.
* View same group blood request.

**Modules of Organization Side:**

1. **User login:**

* At first the user login interface will appear once the application is opened.
* Then admin can log in by using the username and password.
* After successful login, the user will be redirected to the dashboard. A successful attempt will be informed with a login success message.

2. **Profile View:**

* The user will be able to view their profile.

3. **Event management module:**

* The organization will be able to manage the events and their related information.

4. **Donor Registration Module:**

* The organization will get a user-friendly interface to fill up the donor registration-related information (like birth details, name, blood group, etc.).
* After successful completion of the form, the organization can submit the form. Thus, submitted information will be stored in a database. The system will provide a unique username and password to the system users.

Use Case Diagram

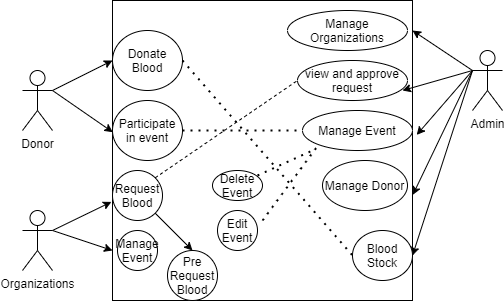


Figure 3.1 Use Case Diagram

### 3.1.2 **Non-Functional Requirement**

a) **User Friendly**

The term user-friendly is self-explanatory. When something is user-friendly, it is easy to access and work with. The blood bank management system is user friendly. Users with basic knowledge and skills of computers can easily use the web application. The blood bank management system uses a simplified design and navigation, as well as simple language on the content to improve the user-friendliness of the web application. User-friendliness helps the web application to improve and increase the number of visitors who use the web application.

b) **Easy Access**

A blood bank management system is a web application. Thus, it can be accessed anytime from anywhere with an Internet Connection.

c) **Information Accuracy**

The information included in the Blood bank management system is obtained from reliable sources. They are checked thoroughly before publishing on the web application. The web application makes sure to avoid making mistakes during data and information retrieval. Thus, the Blood bank management system provides an accuracy of information.

d) **Information Security**

from the Application level, all users are given their separate email and password. The email and password will be known by the owner only which helps in security. The admin will have their email and password too which leads to information security. At the server level, the most trusted server, latest operating system and latest security patches are used.

## 3.2 **Feasibility Analysis**

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

### 3.2.1 **Technical Feasibility**

It is a specific technical solution and availability of technical resources and expertise. Technical Feasibility ensures the technical expertise of the system. It deals with practical expertise and development schedule to be reasonable. We are using open-source technology to develop this system, which is easily available so this project is technically feasible.

### 3.2.2 **Operational Feasibility**

It is a specific technical solution and availability of technical resources and expertise. Technical Feasibility ensures the technical expertise of the system. It deals with practical expertise and development schedule to be reasonable. We are using open-source technology to develop this system, which is easily available so this project is technically feasible.

## **3.3 Data Model**

**ER Diagram**

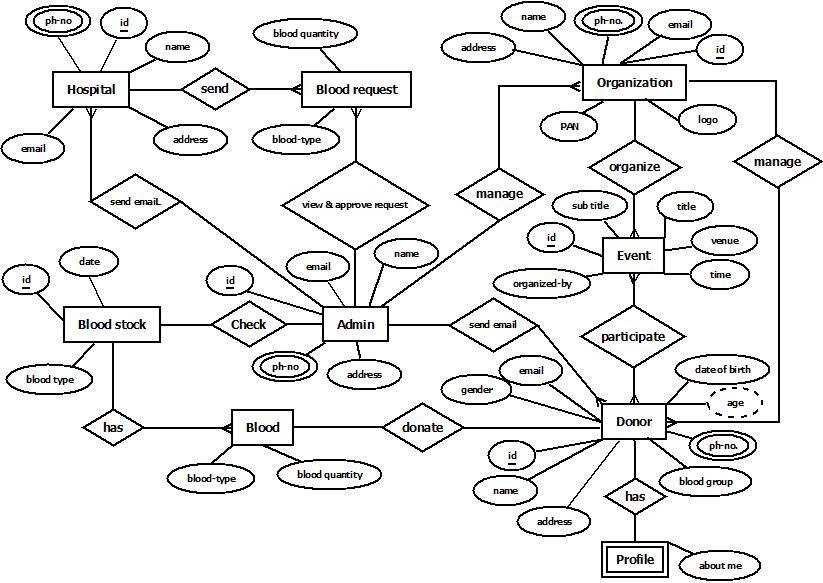


Figure 2.2 ER Diagram

## 3.4 **Process Model**

**Context level Diagram**

A Context Flow Diagram is a top-level (also known as level 0) data flow diagram. It only contains one process node (process 0) that generalizes the function of the entire system in relationship to external entities. In the context diagram, the entire system is treated as a single process and all its inputs, outputs, sinks and sources are identified and shown. The context level diagram is as given below.

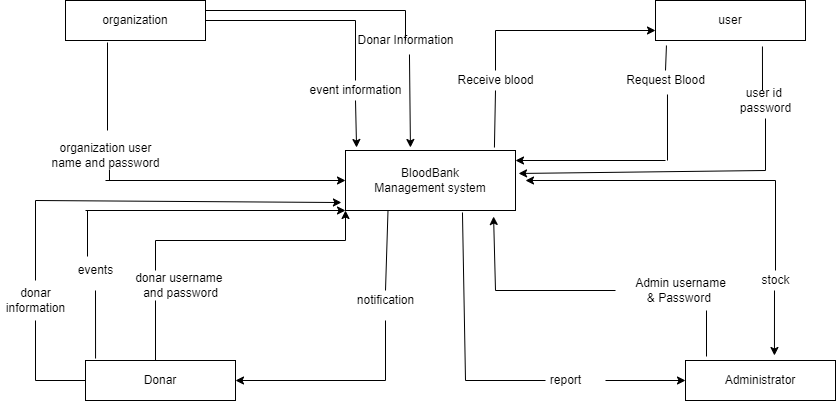


Figure 3 context level diagram

The figure above shows a context level diagram for the Blood bank management system. It contains a process (shape) that represents the system to model. It also shows the participants who will interact with the system, called the external entities. In this example, Admin and the Donor, organization, user are the entities who will interact with the system. In between the process and the external entities, there is data flow (connectors) that indicate the existence of information exchange between the entities and the system.

**DFD Diagram:**

A Data Flow Diagram (DFD) is a graphical representation of the "flow" of data through an Information System. A data flow diagram can also be used for the visualization of Data Processing. It is common practice for a designer to draw a context-level DFD first which shows the interaction between the system and outside entities. This context**-**level DFD is then "exploded" to show more detail of the system being modelled.

All of the below diagrams have been used for the visualization of data processing and structured design.

#### **DFD Level-1**

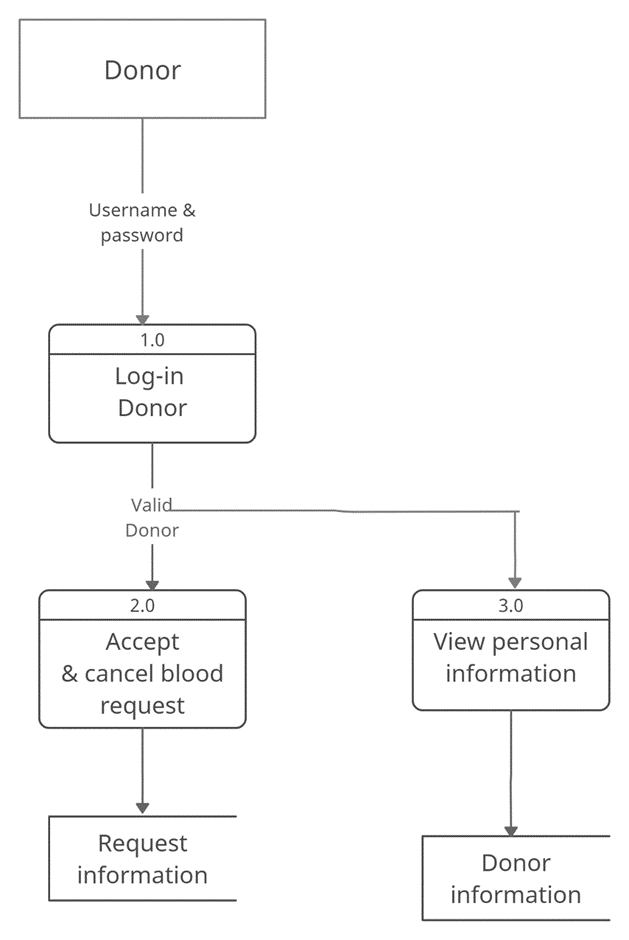


Figure 4 Donor DFD Level 1

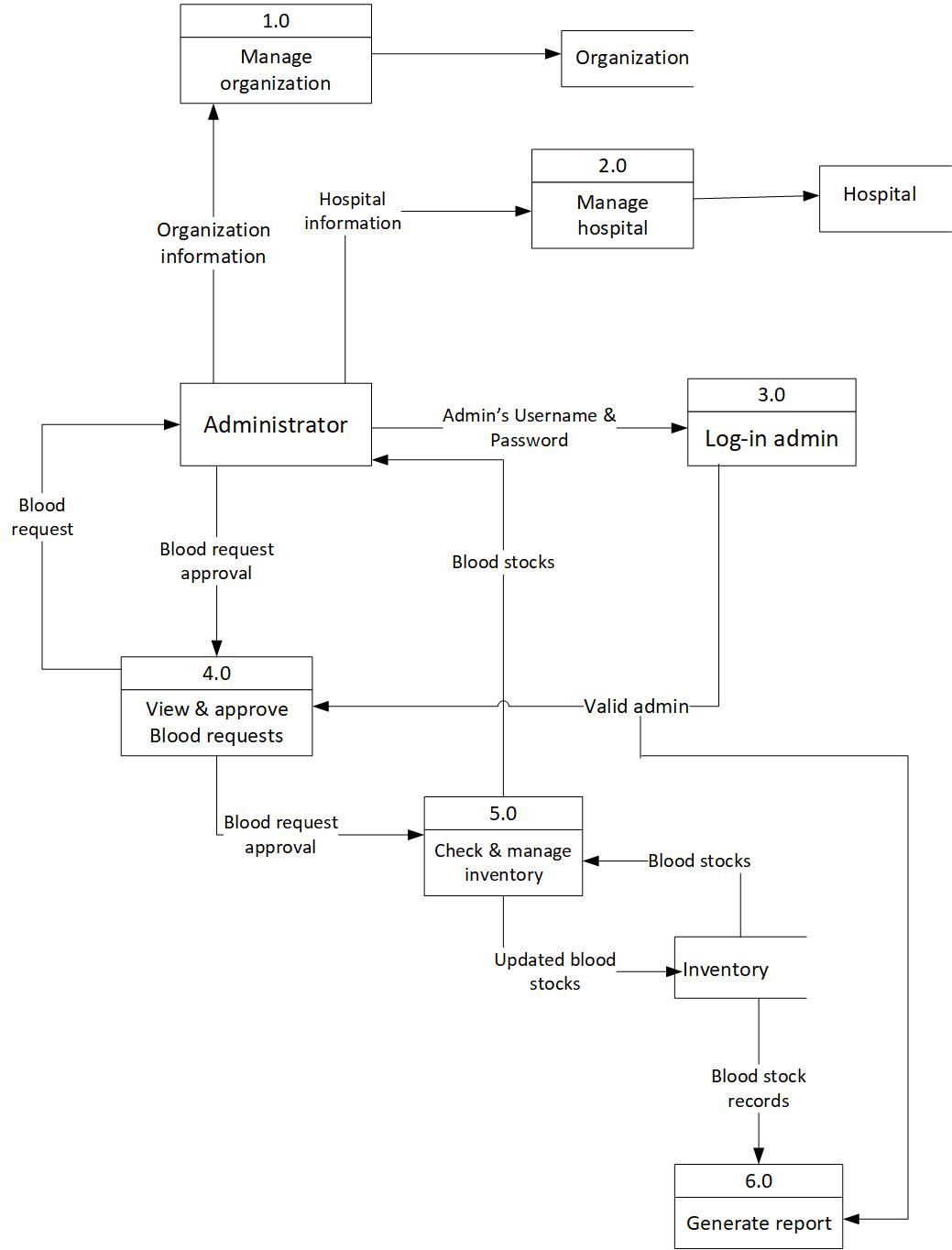


Figure 5 Admin DFD diagram Level-1

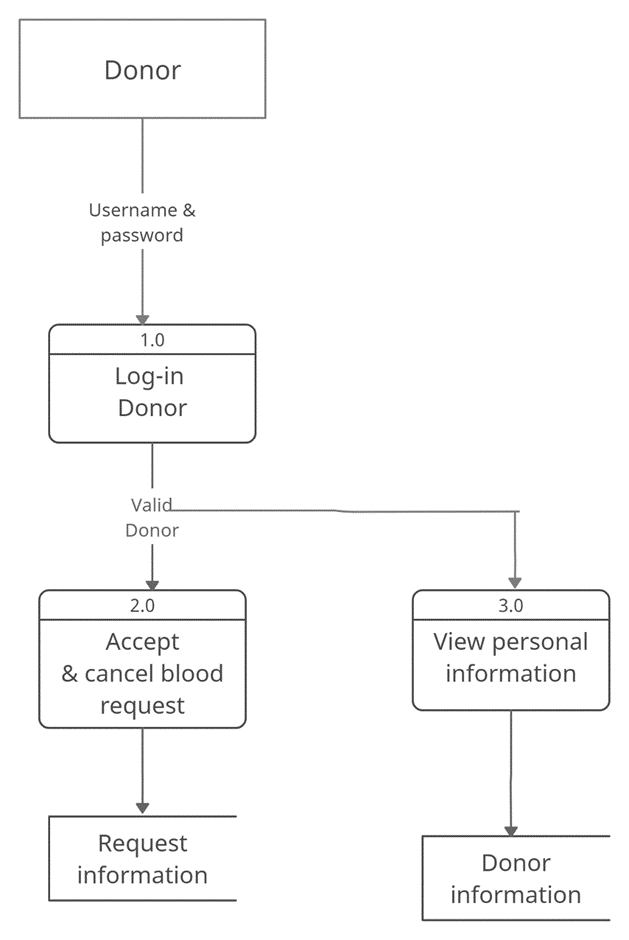


Figure 6 Donor DFD Level 1

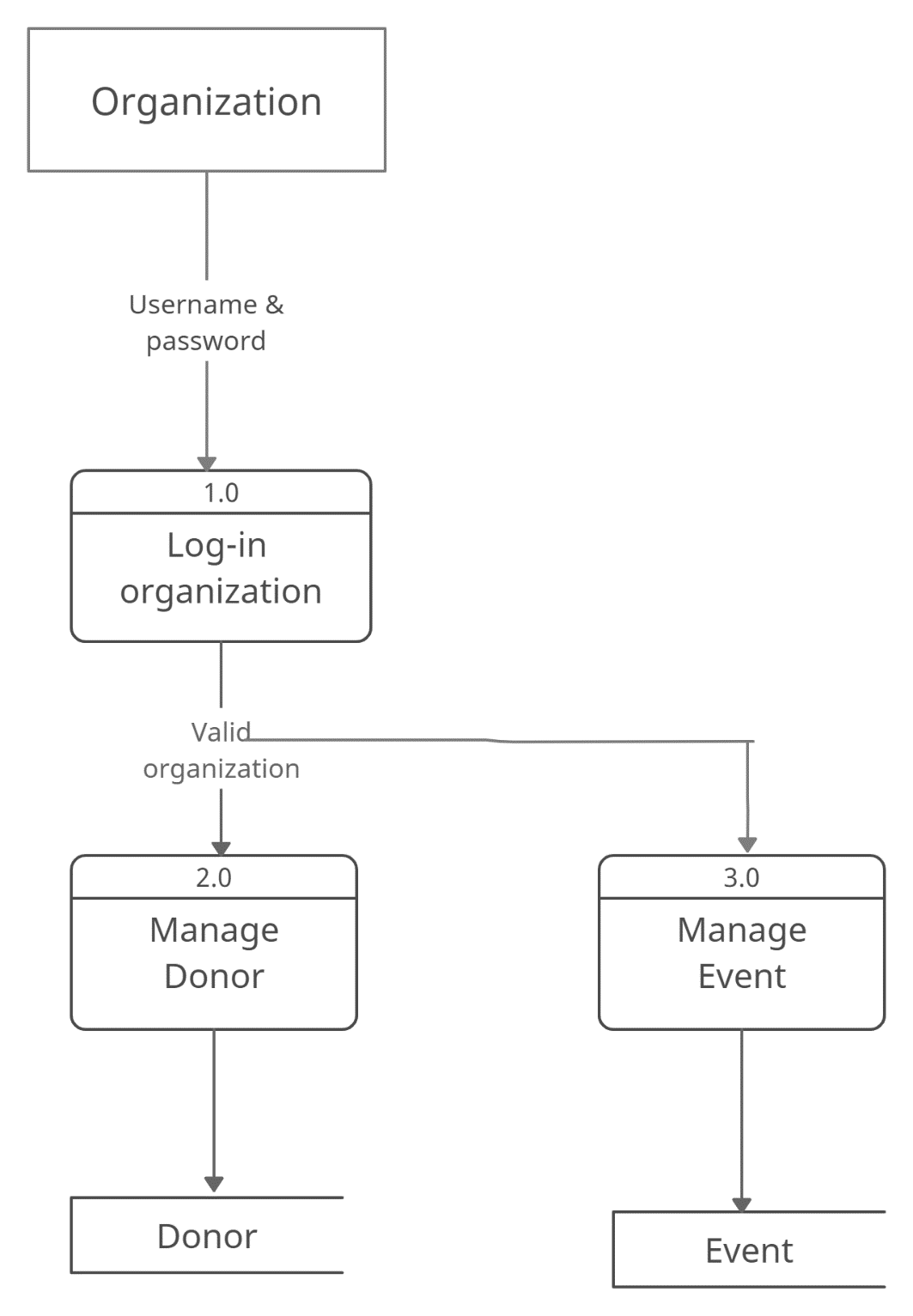


Figure 7 Organization DFD diagram Level-1

#### **DFD Level 2**

The second level DFD shows how the system is divided into sub-systems (processes), each of which deals with one or more data flows to or from external agents, and which together provide all functionality of the Blood Bank Management System.

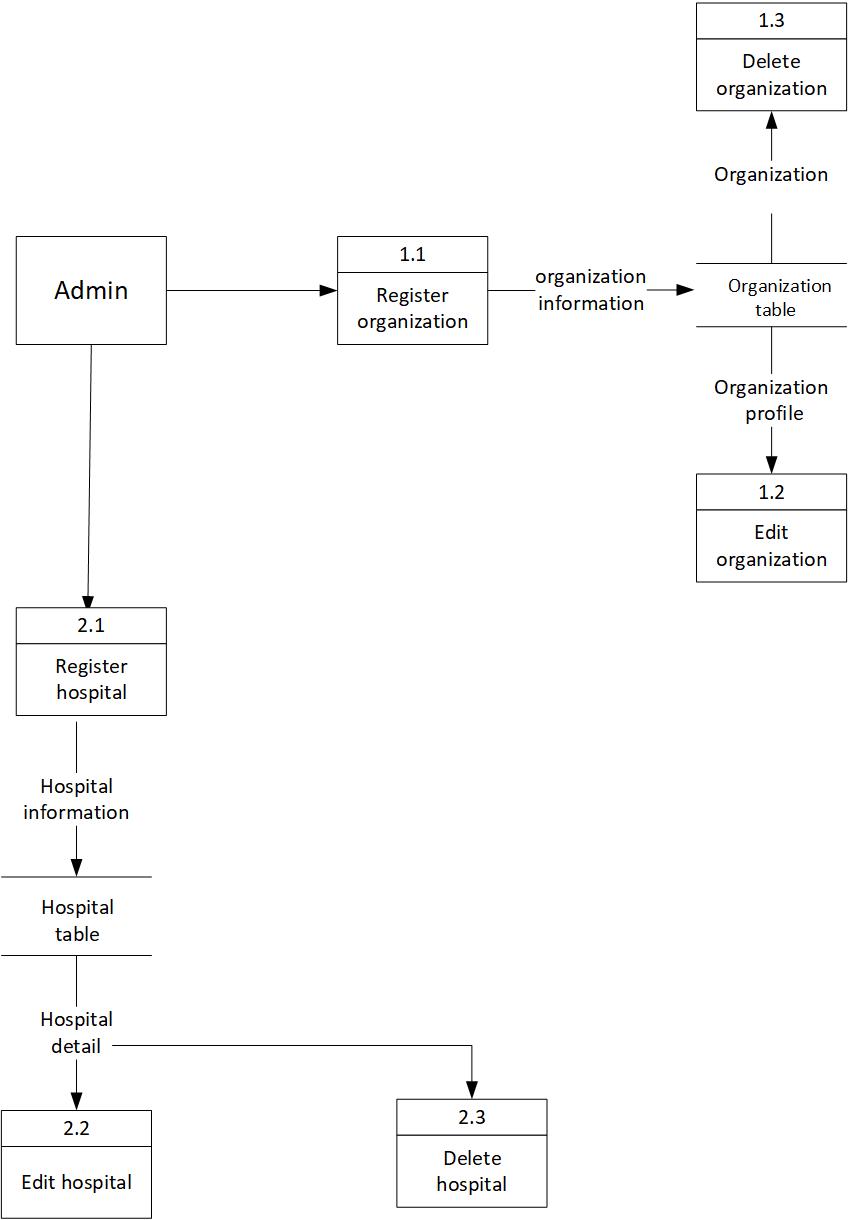


Figure 8 Admin DFD diagram Level

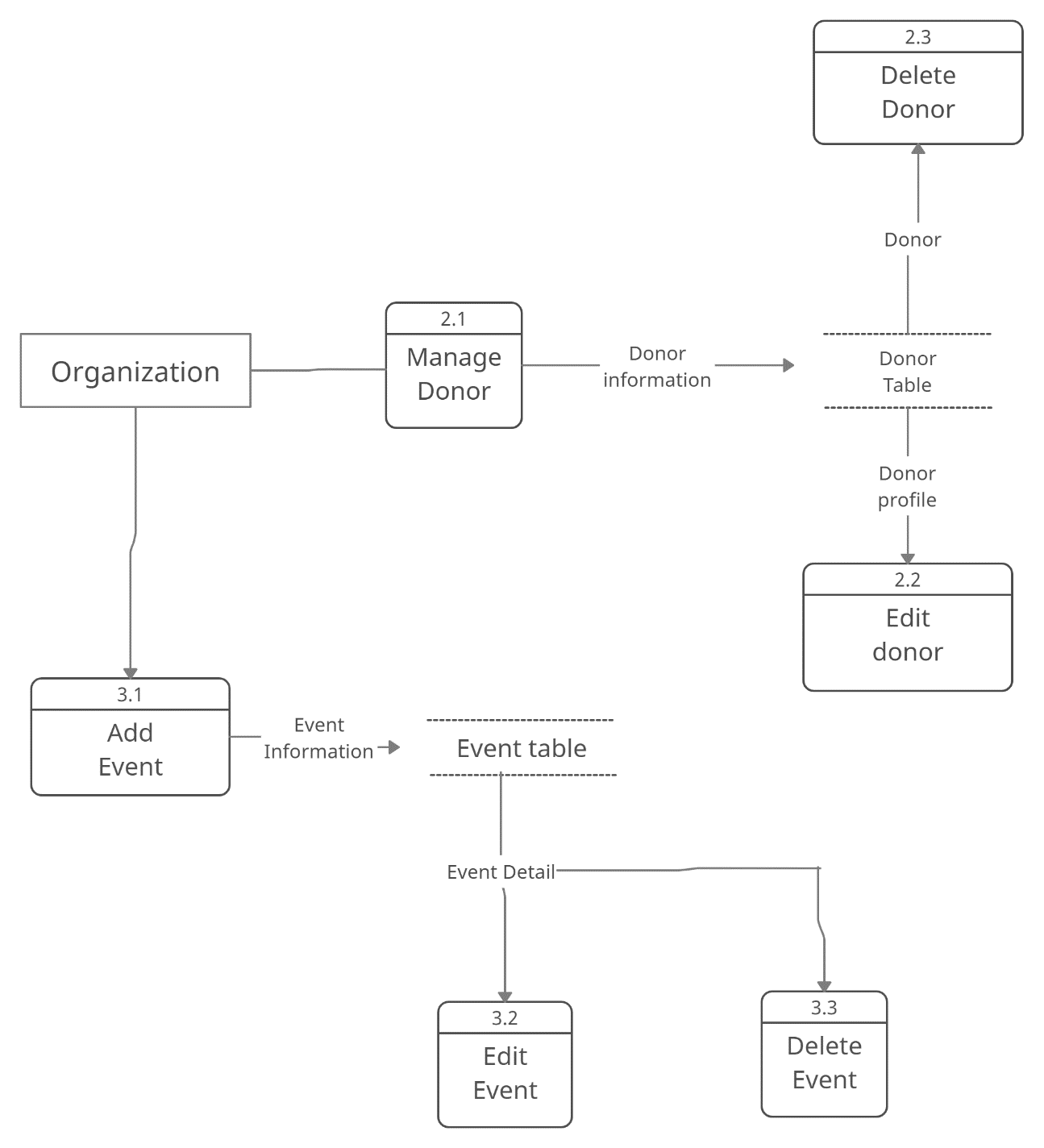


Figure 9 Organization DFD diagram Level-2

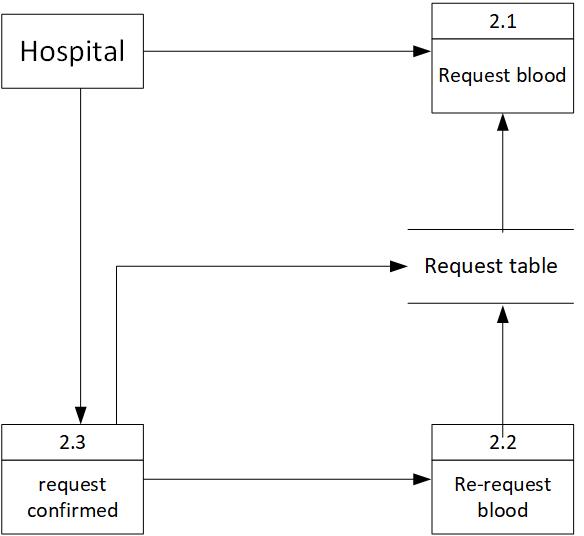


Figure 10 Hospital DFD diagram Level-2

# **CHAPTER 4**

# **4. System Design**

## **4.1. System Design**

System design is the process of defining elements of a system such as the modulesand components, database schema, interface design and process design through the flowchart.

### **4.1.1. Architectural Design**

Our system is based on 3-Tier architecture. A three-tier architecture is a client-server architecture in which the functional process logic, data access, computer data storage and user interface are developed and maintained as independent modules on separate platforms.

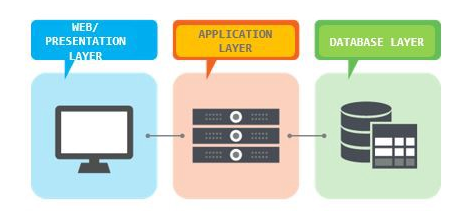


Figure 11 3-Tier Architecture

### **4.1.2. Database Design**

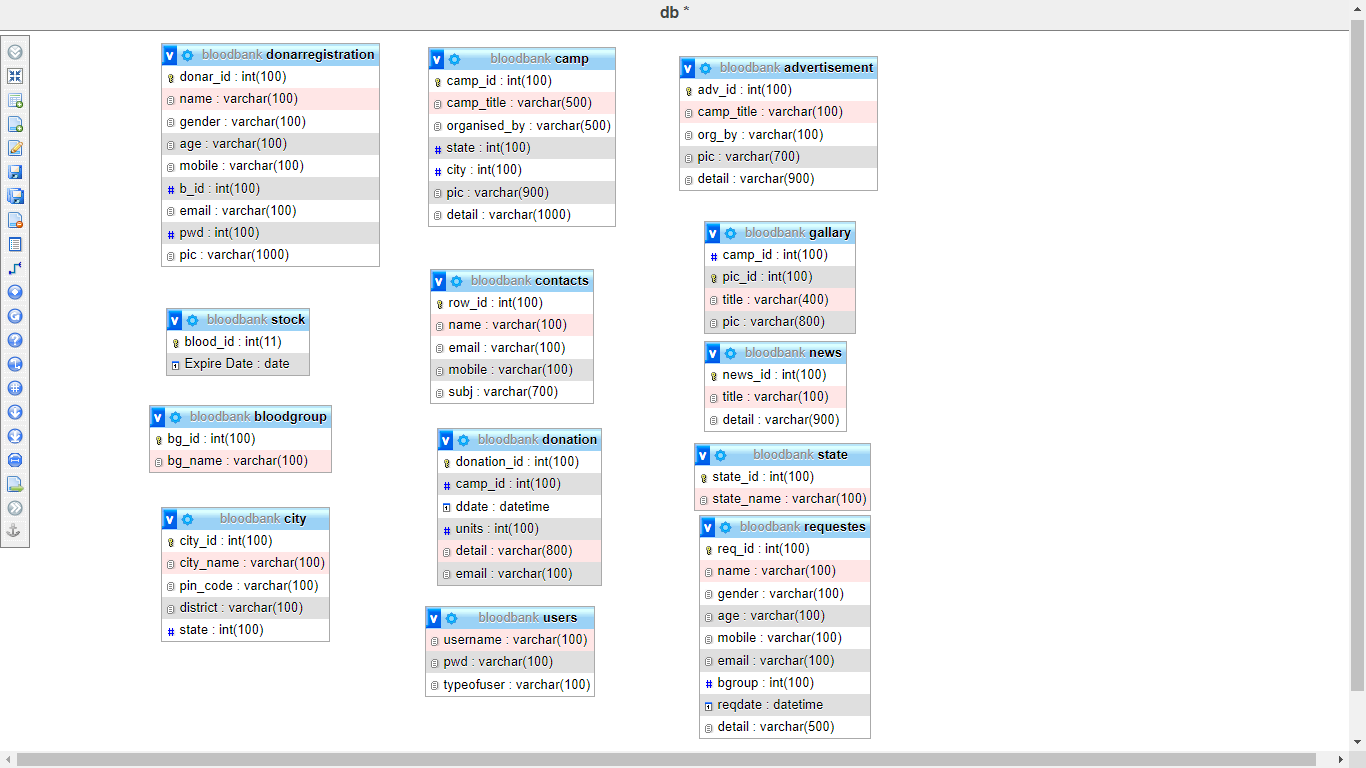


Figure 12 Database Design

### **4.1.3. Data Dictionary**

It is a set of information describing the contents, format, and structure of a database and the relationship between its elements, used to control access to and manipulation of the database. A data dictionary is a collection of descriptions of the data objects or items in a data model for the benefit of programmers and others who need to refer to them. A first step in analyzing a system of objects with which users interact is to identify each object and its relationship to other objects. After each data object or item is given a descriptive name, its relationship is described (or it becomes part of some structure that implicitly describes the relationship), the type of data (such as text or image or binary value) is described, possible predefined values are listed. This collection can be organized for reference into a book called a data dictionary.

Table 1 Donor Registration

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Data type | Constraints | Description |
| Donar\_id | Int(100) | Primary key | To store donor unique identification |
| Name | Varchar(100) | Not Null | To Store the name |
| Gender | Varchar(100) | Not Null | To store gender |
| Age | Varchar(100) | Not Null | To store age |
| Mobile | Varchar(100) | Not Null | To store contact no |
| B\_id | Int(100) | Not Null | To store blood group |
| Email | Varchar(100) | Not Null | To store email address |
| PWD | Int(100) | Not Null | To store passwords |
| pic | Varchar(100) | Not Null | To store picture |

Table 2 Users Table

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Data type | Constraints | Description |
| username | Varchar(100) | Not Null | To store username |
| password | Varchar(100) | Not Null | To store password |
| Type user | Varchar(100) | Not Null | To store type of user |

Table 3 Blood Group

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Data type | Constraints | Description |
| Bg\_id | Int (100) | Not Null | To store Blood Group |
| Bg\_Name | Varchar(100) | Not Null | To Store Blood Group Name |

### **4.1.4** **Flowchart Diagram**

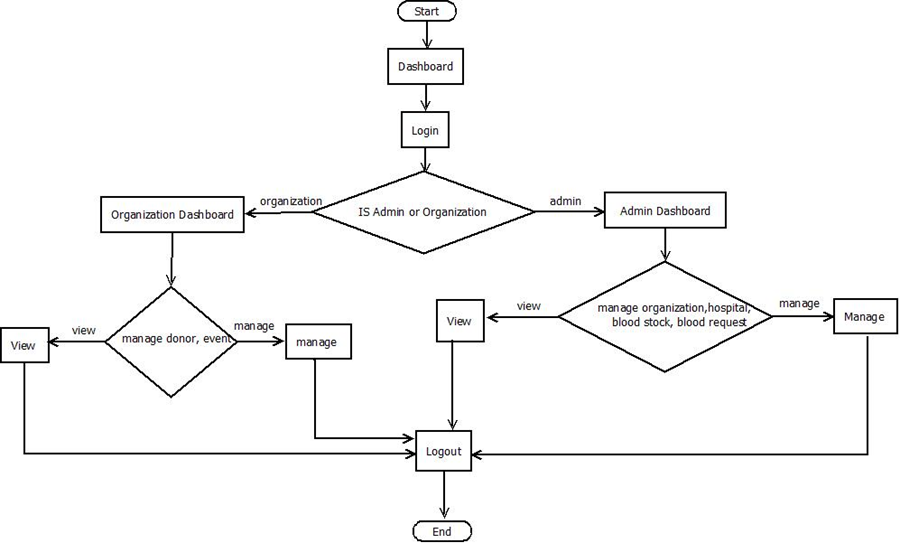


Figure 13 Flowchart Diagram for Admin and organization

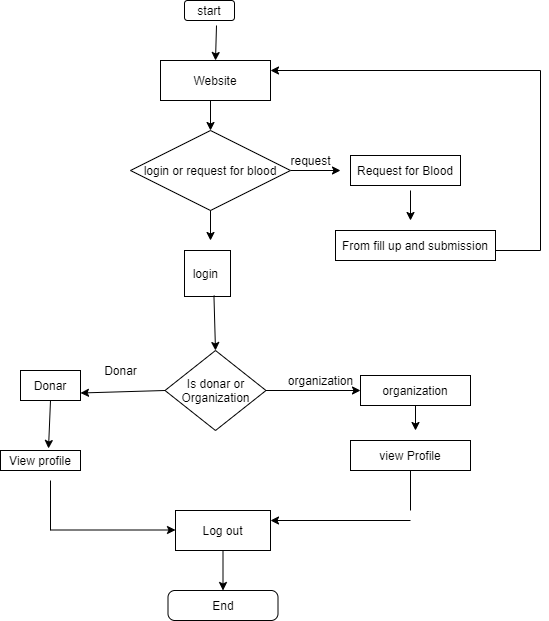


Figure 14 Flowchart Diagram for Donar and organization

**CHAPTER 5**

5. **System Development**

**5.1 System Development Tools**

Following technologies and tools were used for the developing Blood Bank Management System:

**5.1.1 Front End**

**5.1.1.1 HTML**

Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript. Web browsers receive HTML documents from a web server or local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as <img /> and <input /> directly introduce content into the page. Other tags such as <p> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags but use them to interpret the content of the page.

**5.1.1.2 CSS**

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML.CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.CSS is designed to enable the separation of presentation and content, including layout, colours, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file and reduce complexity and repetition in the structural content.

CSS information can be provided from various sources. These sources can be the web browser, the user and the author. The information from the author can be further classified into inline, media type, importance, selector specificity, rule order, inheritance and property definition. CSS style information can be in a separate document or it can be embedded into an HTML document. Multiple style sheets can be imported. Different styles can be applied depending on the output device being used; for example, the screen version can be quite different from the printed version, so authors can tailor the presentation appropriately for each medium. The style sheet with the highest priority controls the content display. Declarations not set in the highest priority source are passed on to a source of lower priority, such as the user agent style. The process is called cascading.

**5.1.1.3 JavaScript**

JavaScript s a high-level, interpreted scripting language that conforms to the ECMAScript specification. JavaScript has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions. Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web. JavaScript enables interactive web pages and is an essential part of web applications. The vast majority of websites use it, and major web browsers have a dedicated JavaScript engine to execute it. As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative (including object-oriented and prototype-based) programming styles. It has APIs for working with text, arrays, dates, regular expressions, and the DOM, but the language itself does not include any I/O, such as networking, storage, or graphics facilities. It relies upon the host environment in which it is embedded to provide these features.

Initially only implemented client-side in web browsers, JavaScript engines are now embedded in many other types of host software, including server-side in web servers and databases, and in non-web programs such as word processors and PDF software, and in runtime environments that make JavaScript available for writing mobile and desktop applications, including desktop widgets.

**5.1.2 Back End**

**5.1.2.1 PHP**

**PHP:** Hypertext Preprocessor (or simply PHP) is a general-purpose programming language originally designed for web development. It was created by Rasmus Lerdorf in 1994.PHP code may be executed with a command-line interface (CLI), embedded into HTML code, or used in combination with various web template systems, web content management systems, and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in a web server or as a Common Gateway Interface (CGI) executable. The web server outputs the results of the interpreted and executed PHP code, which may be any type of data, such as generated HTML code or binary image data. PHP can be used for many programming tasks outside of the web context, such as standalone graphical applications and robotic drone control.

**5.1.2.2 MySQL**

MySQL is open-source relational database management (RDBMS). A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The MySQL Database Software is a client/server system that consists of a multithreaded SQL server that supports different back ends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs). The MySQL Database Software is a client/server system that consists of a multithreaded SQL server that supports different back ends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs). The light version is a small package containing Apache HTTP Server, PHP, MySQL, phpMyAdmin, OpenSSL, and SQLite. Obtaining and Installing XAMPP. XAMPP is a free package available for download and use for various web development tasks. All XAMPP packages and add-ons are distributed through the Apache Friends website at the address: <http://www.apachefriends.org/hnologies> were used for our proposed system.

**5.1.2.3 THE APACHE HTTP SERVER**

The Apache HTTP Server Project is an effort to develop and maintain an open-source HTTP server for modern operating systems including UNIX and Windows. The goal of this project is to provide a secure, efficient, and extensible server that provides HTTP services in sync with the current HTTP standards.

**CHAPTER 6**

# **6. Implementation and Testing**

## **6.1 Implementation**

The “Blood Bank Management System” system needs some resources for its deployment. The different resources are hardware, software, Server, and database too. All the involved implementation sources are mentioned here in the figure below:

### **6.1.1 Software for system implementation**

Table 4 Software tools used for System Development

|  |  |
| --- | --- |
| **Software** | **Description** |
| Editor (visual studio code, Sublime text) | Used for coding and the system development |
| XAMPP | The standalone server integrates the Apache server and MySQL database system. |
| Draw.io | Purposely for diagram development |
| Microsoft Word | Purposely for Documentation |
| Notion | Project Management, Coordinate with Team Member and assign a task for group member |
| Trello | Virtual White Board for Project Management |

Table 5 Software for System Deployment (Server)

|  |  |  |
| --- | --- | --- |
| Software | Version | Description |
| Ubuntu | 18 or above | Operating System |
| Apache | 2.3 + | Application Server |
| MySQL | 5.6 + | Database Server |
| Php | >= 7.3 | Programming Language |

### **Hardware for the System Implementation**

Table 6 Hardware for System Implementation

|  |  |
| --- | --- |
| **Hardware** | **Description** |
| Personal Computer with minimum  requirements:  2 GB RAM  20 GB Hard Disk | For Development |

## **Testing**

Testing is the method of evaluating a system or its components to determine whether they meet the specifications. Testing is the process of running a system to find any loopholes, bugs, or incomplete specifications that aren't covered by the actual requirements. Testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation.

**6.1.2 Unit Testing**

Unit testing is a form of software testing in which individual software components are evaluated. During the creation of an application, unit testing of software products is performed.

The objective of Unit Testing:

* To help with code reuse.
* To isolate a section of code.
* To verify the correctness of the code.
* To test every function and procedure.
* To fix bugs early in the development cycle and to save costs and time.
* To help the developers to understand the code base and enable them to make changes quickly.

### **6.1.2 Test Case Design**

Table 7 Test case for Login

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.NO | Test Case | Expected Output | Actual Output | Remarks |
| 1 | Trying Login with an invalid user/admin email/password | Invalid Login attempt | As Expected, | Test Success |
| 2 | Trying to log in with a valid user email/password. | Valid Login attempt | As Expected, | Test Success |

Table 8 Test case for Signup

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.NO | Test Case | Expected Output | Actual Output | Remarks |
| 1 | Trying registration with null Name, Blood group, User Image, Password | Registration Fail | As Expected, | Test Success |

Table 9 Test Case for Request for Blood

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.NO | Test Case | Expected Output | Actual Output | Remarks |
| 1 | Showing same group blood request and donor list. | Showing request | As Expected, | Test Success |

# **CHAPTER 7**

1. **Conclusion and References**
   1. **Conclusion**

The blood bank management system allows a request for blood in case of emergency and maintains the records of blood donors, blood donation programs, blood requests, and blood stocks in the bank. In this system, there is information regarding the blood donation programs available. There is a centralized database of donors. The problem is keeping track of the actual amount of every blood type in the blood bank is solved. The conclusions can be summed up into the following points.

The blood bank management system provides means for donors to find blood donation events in their region. This system notifies donors about an emergency for blood in case of unavailability of blood in bloodstock. It helps to maintain records of blood donors, blood donation information, blood stocks and blood requests in a centralized database system.

## **Future Work**

In future, we are planning to add the following features:

1. Not only expired blood but the blood that has been sent will be also deleted.
2. The status of the request for blood will be edited by the system automatically.
3. The feature of forgetting passwords will be added in donor and user login.
4. Keep the blood donor’s family health history.
5. Keep the track of blood expiration dates.
6. Keep the hospital record with proper details.

# **References**

[1] A. Almamri, "Enhancing Blood Transfusion Safety Through the Use of Online Blood Bank Management System in Oman,"  
https://www.researchgate.net/publication/328305822\_Blood\_Bank\_System, Oman,2018.

[2] M. Y. Esmail and Y. S. H. Osman, "Computerized Central Blood Bank Management System (CCBBMS)," 2018 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE), 2018.

[3] D. S. M. Vikas Kulshreshtha, "Blood Bank Management Information System in  
India," vol. 1, no. 2, pp. 260-263.

[4]R. Singh, "Hamro life bank," Rumsan Group of Companies, 2020. [Online].  
Available: https://hamrolifebank.com/blood-bank. [Accessed 2020].

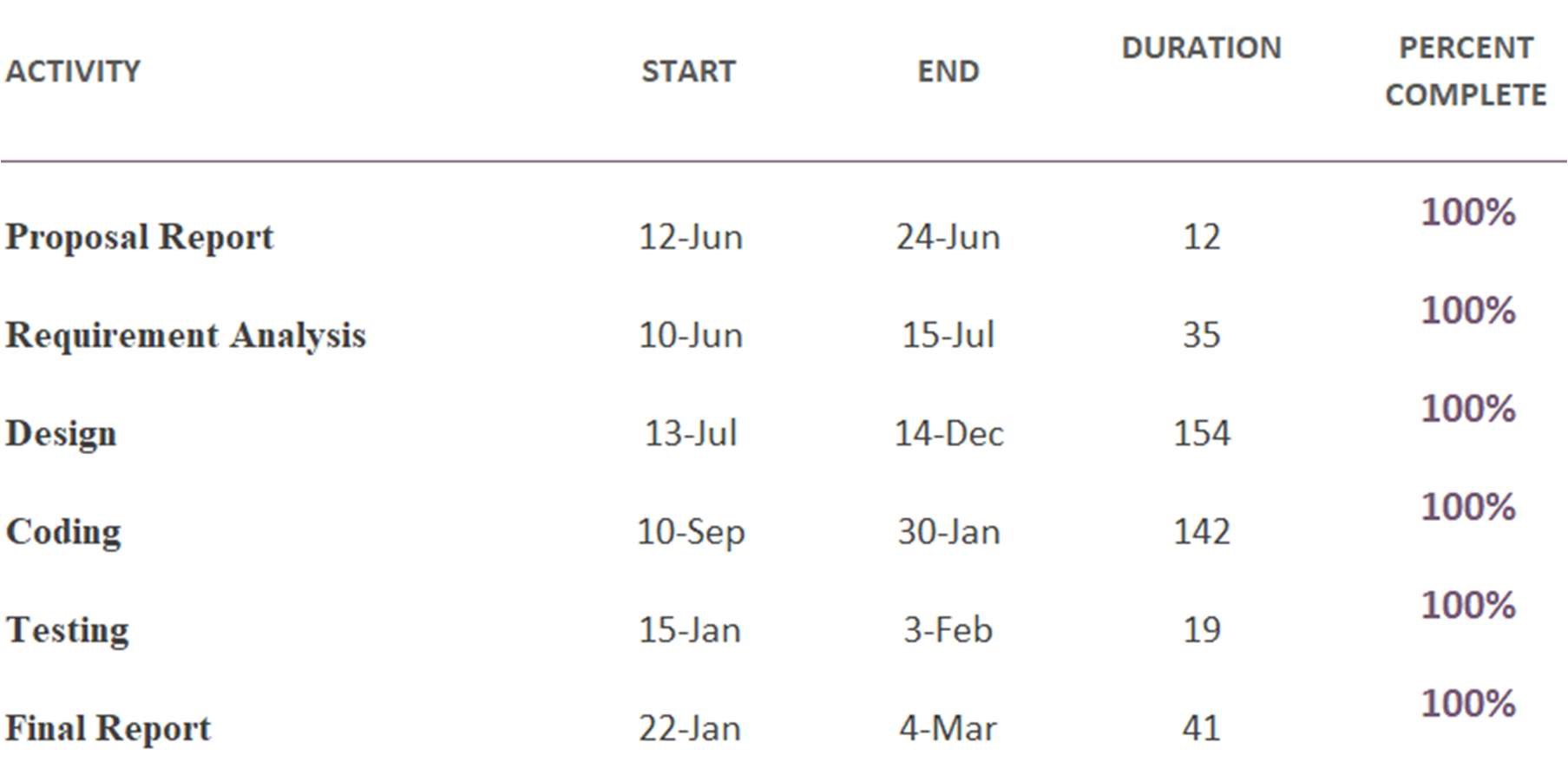
[5]"Bloodbank Nepal, "2014. [Online]. Available: https://www.bloodbanknepal.com/.

[6]N. R. c. society, "Wikipedia," Nepal Red cross society, 1963. [Online]. Available:  
https://en.wikipedia.org/wiki/Nepal\_Red\_Cross\_Society.

[7] , "Blood Bank Management System," Ravi Kumar, Shubham Singh, V Anu Razavi,  
Chennai, India, 2017.

# **Snap Shots**

**Gantt Chart**



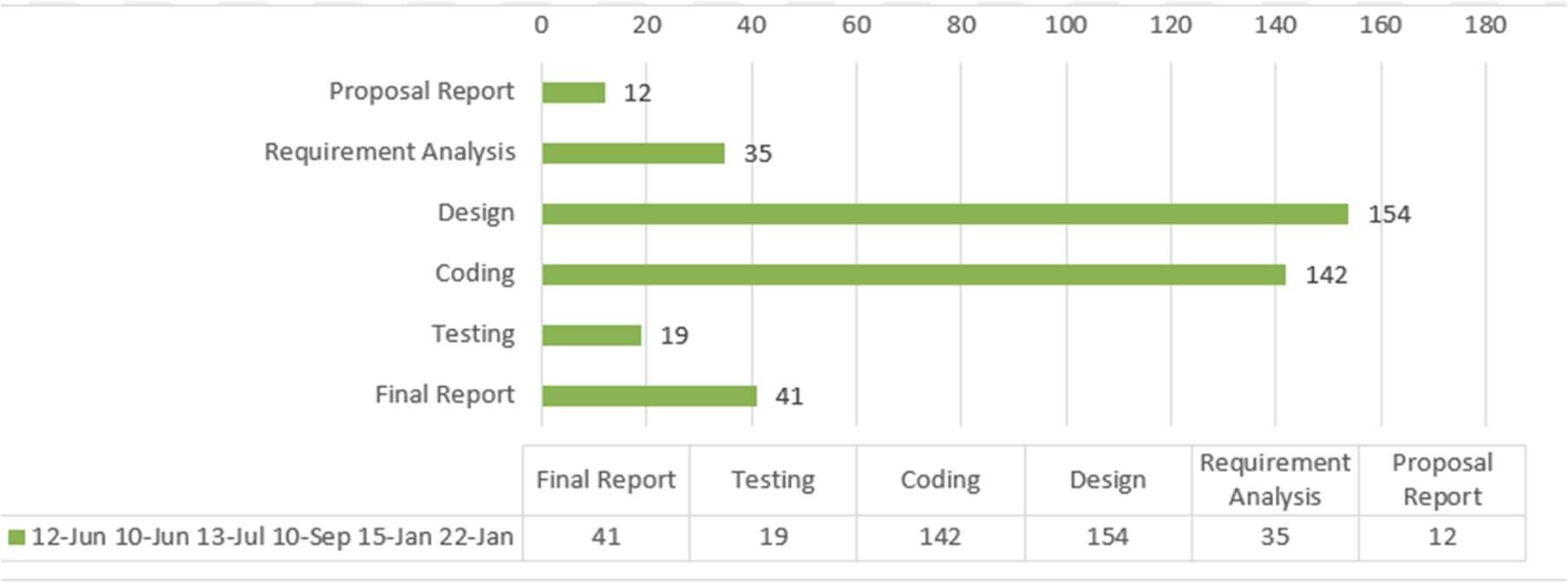


Figure 15 Gantt Chart



Figure 16 Admin Login



Figure 17 Donor Registration

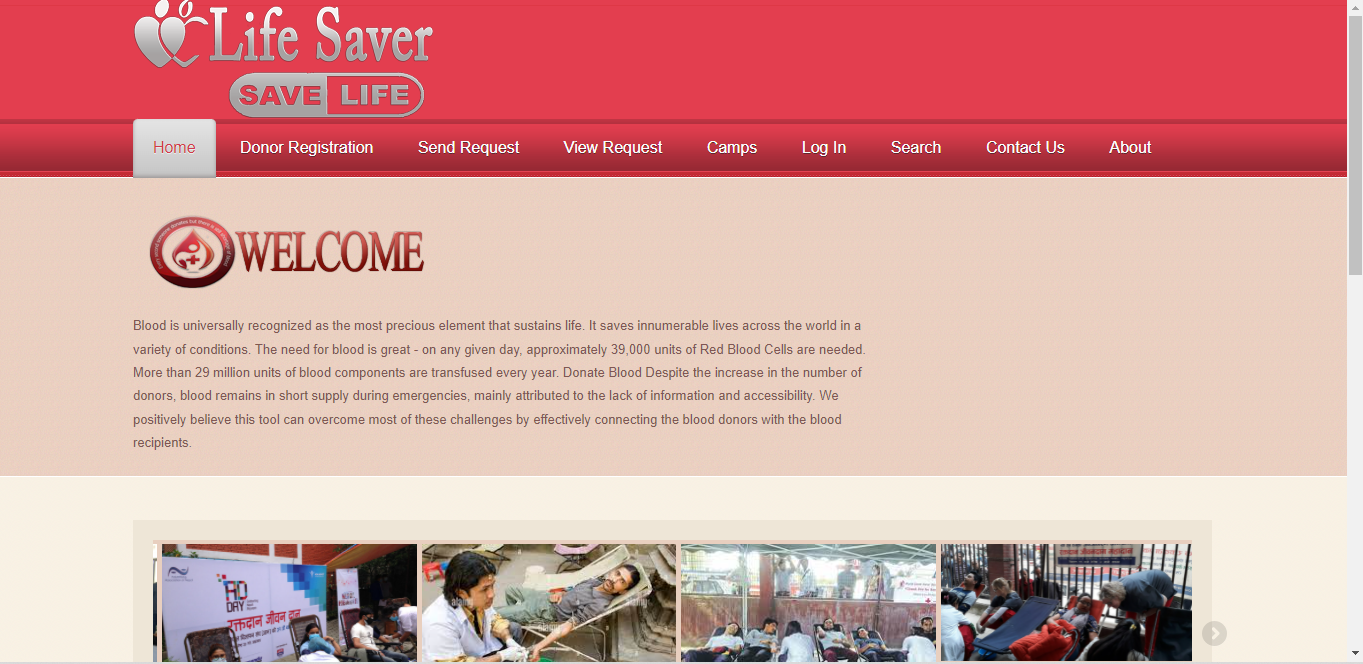


Figure 18 Home Page