

A Project Report On

BLOOD BANK MANAGEMENT SYSTEM

Submitted in partial fulfilment of the

BACHELOR OF COMPUTER APPLICATION

By

AYUSH DHANRAJ

Enrolment No. AJU/190542

Under the esteemed guidance of

Dr. ARUN MARANDI

(Internal Guide)



DEPARTMENT OF COMPUTER SCIENCE & IT

ARKA JAIN UNIVERSITY, JHARKHAND

Jamshedpur

2019-2022



ARKA JAIN UNIVERSITY

A

PROJECT REPORT

ON

BLOOD BANK MANAGEMENT SYSTEM

IN PARTIAL FULFILLMENT OF REQUIREMENT OF

DEPARTMENT OF COMPUTER SCIENCE AND

INFORMATION TECHNOLOGY

BATCH 2019-22

GUIDED BY:

Dr.ARUN MARANDI

PREPARED BY:

AYUSH DHANRAJ

SUBMITTED TO

DEPARTMENT OF COMPUTER SCIENCE & IT

ARKA JAIN UNIVERSITY

A Project Report On
BLOOD BANK MANAGEMENT SYSTEM

by

AYUSH DHANRAJ

Enrolment Number: AJU/190542

Under the esteemed guidance of

DR. ARUN MARANDI

(Internal Guide)

DEPARTMENT OF COMPUTER SCIENCE & IT

ARKA JAIN UNIVERSITY, JHARKHAND

JAMSHEDPUR

2019-2022

ARKA JAIN UNIVERSITY, JHARKHAND

DEPARTMENT OF COMPUTER SCIENCE & IT



DECLARATION

I hereby declare that the project entitled, **BLOOD BANK MANAGEMENT SYSTEM** done at **ARKA JAIN UNIVERSITY** not been in any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

The project is done in partial fulfilment of the requirements for the award of degree of **BACHELOR OF COMPUTER APPLICATION (BCA)** to be submitted as final semester project as part of our curriculum.

Date: - 26/05/22

AYUSH DHANRAJ

(AJU/190542)

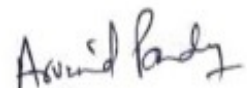
DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY



CERTIFICATE

This is to certify that the project entitled, "**Blood Bank Management System**", is bonafied work of **Ayush Dhanraj** bearing **Enrollment No.:** AJU/190542 submitted in partial fulfilment of the requirements for the award of degree of BACHELOR OF COMPUTER APPLICATION from ARKA JAIN UNIVERSITY, JHARKHAND.


Internal Guide


HOD

Date: 26/05/22



IT DEVELOPER

IT ENGINEERS

Ref. No: ITD-IP-0322-028

Date: 12-03-2022

PROJECT INTERNSHIP LETTER

TO WHOM IT MAY CONCERN

This is to certify that Mr. **Ayush Dhanraj**, a student of ARKA JAIN University, Jharkhand, Jamshedpur has successfully completed his Project Internship/Training (Virtual) with our organization as a **Windows Application Developer** during the period of November 2021 to February 2022.

The topic of the project was:

“Blood Bank Management System”

The Frontend and backend used for project development was Java, MS-Access.

During this period of internship/training, he was found to be honest, creative & able to perform all his duties perfectly on the project.

We wish him all the best in all his future accomplishments.

For IT DEVELOPER



(Authorized Signatory)

Address: Rourkela, Odisha, India

Website: itdeveloper.in

E-mail: info@itdeveloper.in

ACKNOWLEDGEMENT

It is a genuine pleasure to express my profound gratitude and deep regards to my Internal Guide “**Mr. ARUN MARANDI**” and our HOD “**Prof. ARVIND KUMAR PANDEY**” for their exemplary guidance, monitoring and constant encouragement. I would like to express my special thanks to **ARKA JAIN UNIVERSITY** who gave me the golden opportunity to do this wonderful project on the topic “**BLOOD BANK MANAGEMENT SYSTEM**”, which helped me in doing a lot of Research and I came to know about so many new things.

ABSTRACT

BLOOD BANK MANAGEMENT SYSTEM is a project which aims in developing a Desktop Application to maintain all the daily work of Blood Bank Centre. This project has many features which are generally not available in normal Blood Bank Management Systems like Blood/Donors Records, Issue Patient Bill etc. It also has a facility of admin login through which the admin can monitor the whole system. This System can be used to search for Assign Work, Add/Remove Staff, Add / Remove Blood etc. The Admin after logging into his account can generate various reports such as Blood Stock Report and Service/Work Report.

Overall, this project of ours is being developed to help the Blood Bank Centre to maintain the Service Centre in the best way possible and also reduce the human efforts.

TABLE OF CONTENTS

BLOOD BANK MANAGEMENT SYSTEM

DECLARATION:	5
Chapter 1: INTRODUCTION	
1.1 Introduction	12
1.2 Objective	13
1.3 Advantage of Purpose System	14
Chapter 2: Survey of Technologies	
2.1 About Software	15
2.2 Java	16
2.3 Microsoft Access	17
Chapter 3: Requirements and Analysis Problem	
3.1 Requirement and Analysis Problem	18
3.2 Planning and Scheduling	19
3.3 Hardware and Software Requirements	20
3.4 Data Flow Diagram (DFD)	22
3.4.1 DFD 0 Level	23
3.4.2 DFD 1st Level	24
3.4.3 DFD 2nd Level	25
3.4 Entity Relational Diagram (ER-Diagram)	26
Chapter 4: System Design	
4.1 Form Description	27
4.2 Reports involved in the System	30
4.3 Databases to be expected in the proposed system	31

Chapter 5: System Testing

5.1 White-box Testing	33
5.2 Black-box Testing	34
5.3 Unit Testing	36
5.4 Integration Testing	37
5.5 System Testing	38
5.6 Acceptance Testing	39
5.7 Validation Testing	40

Chapter 6: Interface Design and Source Code

6.1 Interface Design	52
6.2 Source Code	58

Chapter 7: Results and Discussion

7.1 Future Scope of System	59
7.2 Drawback of Existing System	60
7.3 Conclusion	61

Chapter 1: Introduction

“BLOOD BANK” is a cache or bank of blood or blood components, gathered as a result of blood...or Blood bank is a place where blood is collected from Donors so it can be used to treat someone else and typed, separated into components, stored, and prepared. This project AIMS at maintaining all the information pertaining to blood donors. BLOOD BANK MANAGEMENT SYSTEM in java is great project. The basic building aim is to provide blood donation service to the city. The major task of Blood Bank system is to provide blood to help people who want blood. This blood bank system project manages all kind of information related to Blood.

Blood bank donation system in java is planned to collect blood from many donators in short from various sources and distribute that blood to needy people who require blood. To

Do all this we require high quality software to manage those jobs? The government spending lot of money to develop high quality “BLOODBANK MANAGEMENT SYSTEM PROJECT” for do all those kinds of need blood bank management system project in java contain modules which are include the detail of following areas. There is also a help line it is a voluntary and non-governmental organization. It maintains online library of blood donors in India sometimes Doctors and blood bank project have to face the difficulty in finding the blood group Donors at right time. Help line has attempted to provide the answer by taking upon itself the task of collecting Blood bank project nationwide for the cause and care of people in need. Every blood bank has a big plan to collect the blood from many different sources and distribute the same for the needy. To manage all these they require full fledged software which will take care.

OBJECTIVES

- The main OBJECTIVE of this application is to automate the complete operation of the blood bank.
- They need maintain hundreds of thousands of records. Also searching should be very faster so they can find required details instantly.
- Time consumption will be less.
- Customers won't have to wait for long time everything will be very fast instead of hand written.
- No calculation problem as it will do by the computers.
- A proper database will be generated, avoiding the use of register, books.
- A proper and managed detail of each and every customer will be maintained.
- Customer satisfaction will be good which will in turn lead to a faster growth in profit of the bank.

ADVANTAGE OF THE PROPOSE SYSTEM

There are many advantages of Blood Bank Management system

- Blood Bank is used to save resources while flawlessly meeting.
- Automation provides the advantages of improving quality of testing.
- Automated laboratory testing has several advantages including increased quality of pre-analytical steps, reduced error rates.
- By donating blood we can save a life of a person which will give you a mental peace forever.
- The blood bank system consists of independent blood centers which collect, store, and distribute human body and hospital blood bank charging of transfusion-related services.
- Blood bank is a humanitarian organization for meeting the demand for blood in various emergency conditions.

CHAPTER 2: SURVEY OF TECHNOLOGY

ABOUT SOFTWARES

To make the website of Book management System, I have used the platform of visual Studio Code.

Visual Studio Code is a source-code editor developed by Microsoft for Windows, Linux and MacOS. It includes support for debugging, embedded Git control and syntax highlighting, intelligent code completion, snippets, and code refactoring.

It is highly customizable, allowing users to change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality. The source code is free and open-source, released under the permission of MIT license.

The compiled binaries are freeware for any use. In the Stack Overflow 2019 Developer Survey, Visual Studio Code was ranked the most popular developer environment tool, with 50.7% of 87,317 respondents claiming to use it.

Some other softwares and programming languages used in making this project are:

FRONT-END

- JAVA Programming

BACK-END

- MICROSOFT ACCESS

TOOLS

- VISUAL STUDIO

JAVA

Java is a high-level programming language developed by Sun Microsystems. It was originally designed for developing programs for set-top boxes and handheld devices, but later became a popular choice for creating web applications.

The Java syntax is similar to C++, but is strictly an object-oriented programming language. For example, most Java programs contain classes, which are used to define objects, and methods, which are assigned to individual classes. Java is also known for being more strict than C++, meaning variables and functions must be explicitly defined. This means Java source code may produce errors or "exceptions" more easily than other languages, but it also limits other types of errors that may be caused by undefined variables or unassigned types.

Unlike Windows executables (.EXE files) or Macintosh applications (.APP files), Java programs are not run directly by the operating system. Instead, Java programs are interpreted by the Java Virtual Machine, or JVM, which runs on multiple platforms. This means all Java programs are multiplatform and can run on different platforms, including Macintosh, Windows, and Unix computers. However, the JVM must be installed for Java applications or applets to run at all. Fortunately, the JVM is included as part of the Java Runtime Environment (JRE), which is available as a free download.

MICROSOFT ACCESS

Microsoft Access is a database management system (DBMS) from Microsoft that combines the relational Microsoft Jet Database Engine with a graphical user interface and software-development tools. It is a member of the Microsoft 365 suite of applications, included in the Professional and higher editions or sold separately.

Microsoft Access stores data in its own format based on the Access Jet Database Engine. It can also import or link directly to data stored in other applications and databases.[3]

Software developers, data architects and power users can use Microsoft Access to develop application software. Like other Microsoft Office applications, Access is supported by Visual Basic for Applications (VBA), an object-based programming language that can reference a variety of objects including the legacy DAO (Data Access Objects), ActiveX Data Objects, and many other ActiveX components. Visual objects used in forms and reports expose their methods and properties in the VBA programming environment, and VBA code modules may declare and call Windows operating system operations. It doesn't have a web version.

CHAPTER: - 3

REQUIREMENTS AND ANALYISE PROBLEM

Analysis specifies what the software should do. Analysis is the process of gathering and interpreting facts, diagnosing problems and using the information to recommend improvements to the system. Analysis is a detailed study of various operations performed by a system and their relationships within and outside the system. During analysis, data are collected on the available files, decision points and transaction handled by the present system.

During this phase we determined the following system elements:

1. System Objective: We determine the centralized and single objective of the system.
2. Required Resources: Resources may be hardware, people, software etc. We use resources such as MS – ACCESS in the initial phase.
3. Assessment of Feasibility: Our system satisfy the technical, economical, operational feasibilities.

Working of Existing system: -

- ❖ Maintenance and updating of blood bank records, currently, are done manually which, at times, results in loss of critically important records or delay information updates.
- ❖ Difficult to calculate amount of blood in stock.
- ❖ Difficulties in retrieving data (for any specific reason).
- ❖ Difficulty in identifying blood expiry date.
- ❖ Security of records is minimum.
- ❖ Records are very liable to be misfiled or soiled.

PLANNING AND SCHEDULING

Planning involves identification of a task or tasks that need to occur.

Scheduling involves assigning the future action needed to accomplish a given task to occur on a certain date and time. The planning and scheduling of large projects requires the integration of all processes of project management.

In making this project I had made a plan that from where to collect information, what were the softwares that I could use in making this project, how many languages to use and surveyed everything .Than put it in sequencial order to implement it into my project, so if I could get anerror or mistake I could check my planning process.

And at last I have schedule the time and date my project will take .This project took around five months to get completed.

So this was the hole planning and scheduling of my project.

Hardware and software Requirements

Hardware:

- **RAM:** 2 GB or above
- **Hard disk:** 500 GB or above
- **Printer:** Laser printer for good output printing.
- **Monitor:** 18' INCH
- **Processor:** PENTIUM OR ABOVE

Software:

- **Operating system:** Windows 7 or above

DATA FLOW DIAGRAM (DFD)

DFD is an important tool used by system analysis. A data flow diagram model, a system using external entities from which data flows to a process which transforms the data and create output data transforms which go to other processes or external entities such as files. The main merit of DFD is that it can provide an overview of what data a system would process.

SYMBOLS

A Circle represents a process that transforms incoming data flow into outgoing data flows

A Square defines a source or destination of system data

An Arrow identifies data flow direction. It is the pipeline through which the information flows.

An Open Rectangle is a data store, data at rest or a temporary repository of data.

Data flow diagram symbol



Data Flow – Data flow are pipelines through the packets of information flow.



Process : A Process or task performed by the system.

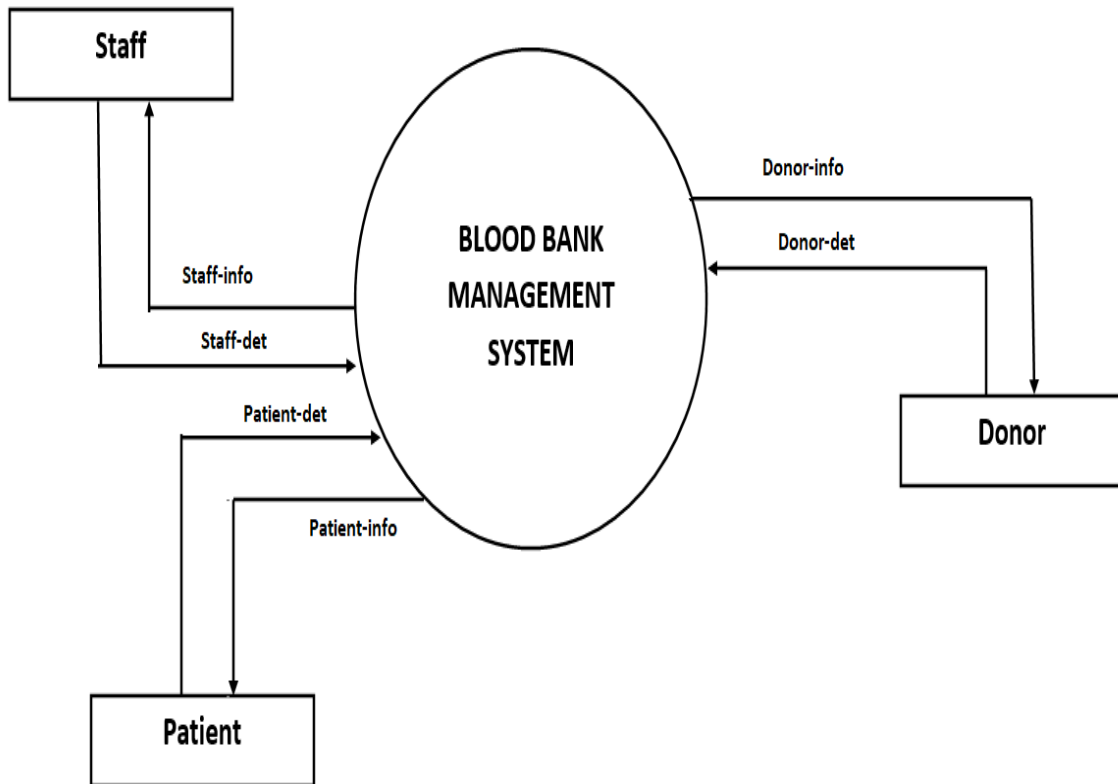


Entity : Entity are object of the system. A source or destination data of a system.

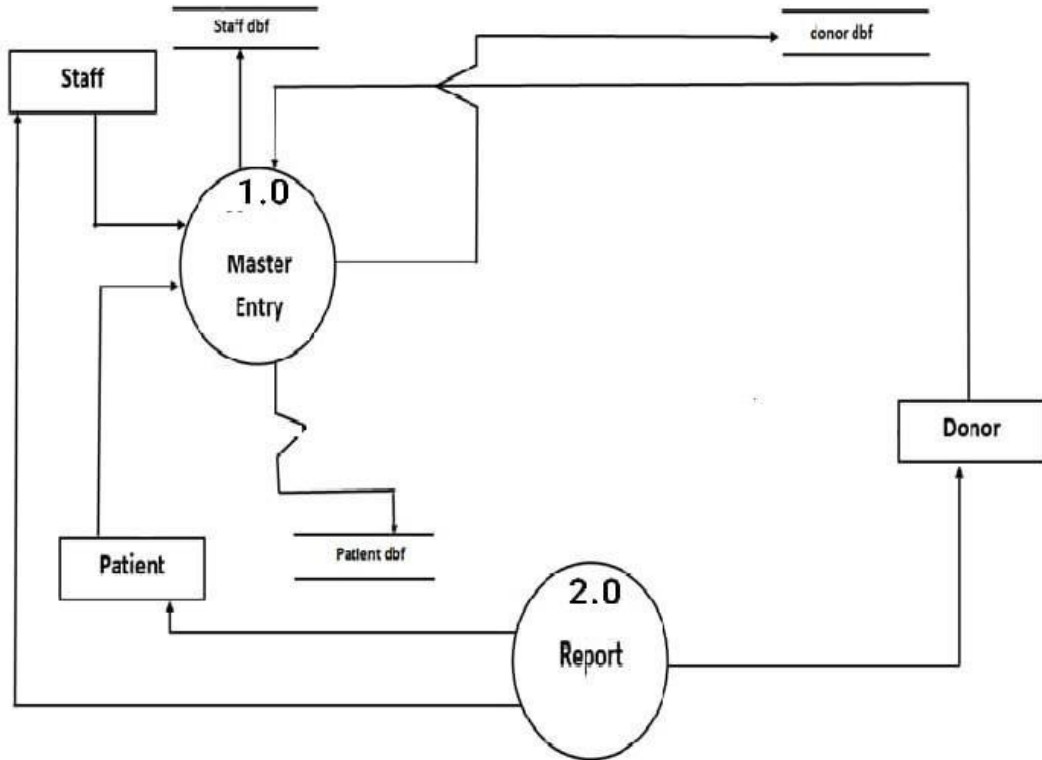


Data Store : A place where data to be stored.

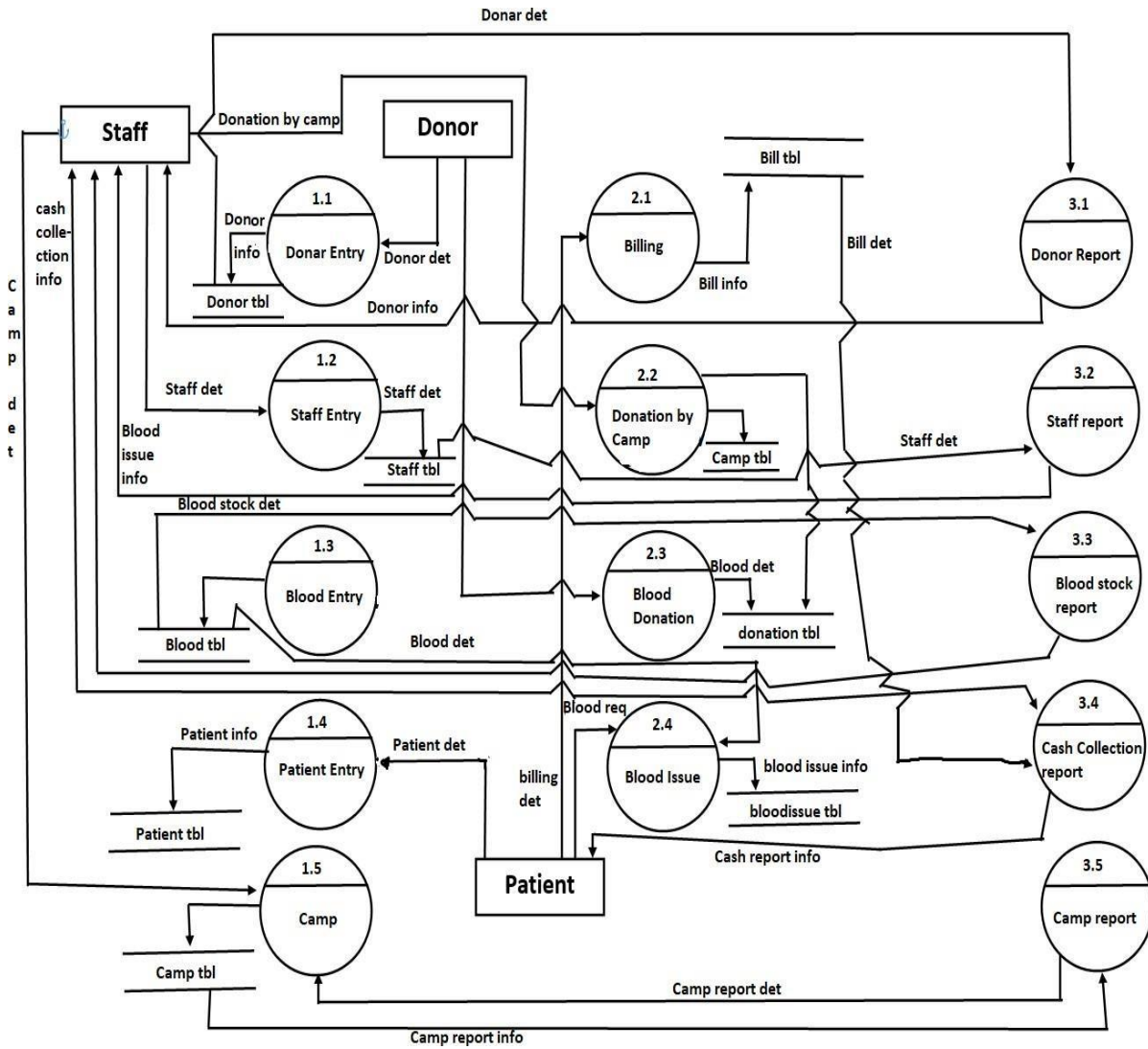
Context Level DFD



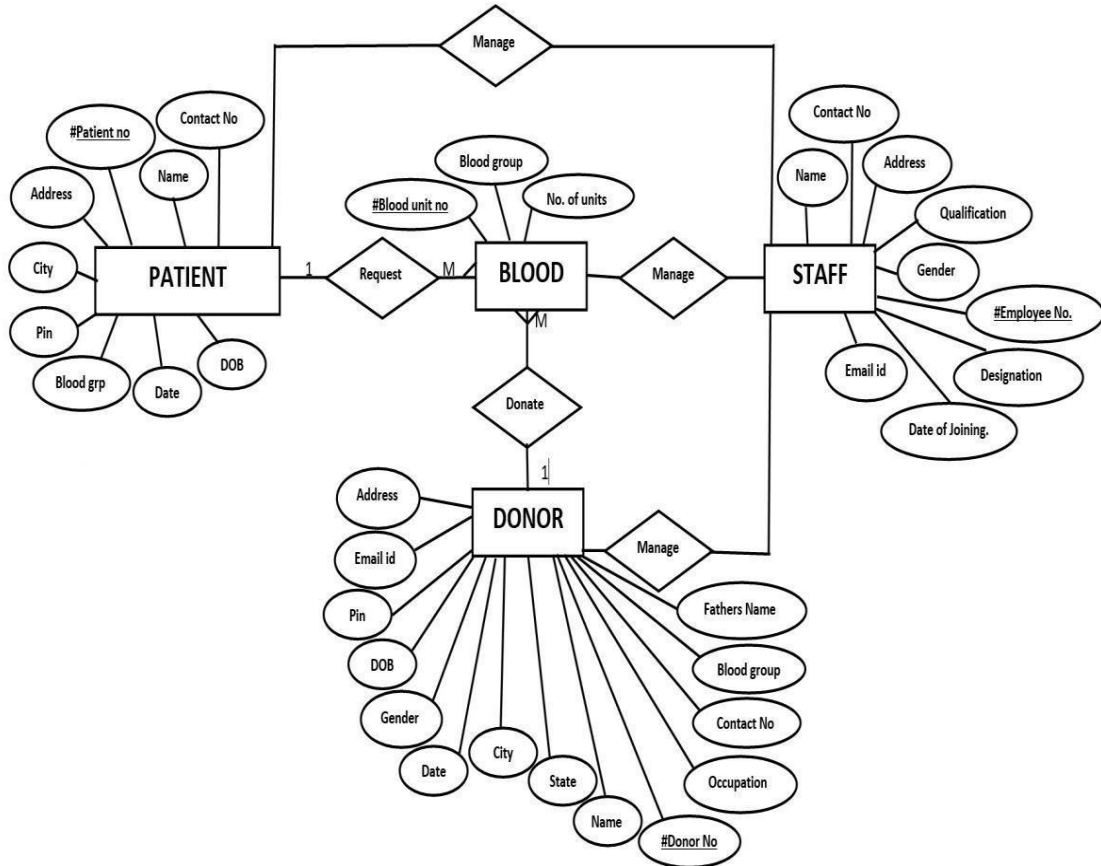
1st LEVEL DED



2nd LEVEL DFD



E R - DIAGRAM



CHAPTER 4: SYSTEM DESIGN

FORM DESCRIPTION

MASTER FORM

- ❖ **DONOR FORM:** This is a master form and whenever a donor comes for donating blood then his/her record are kept here. This may be beneficial for organization in future. In this section has following facilities. Insert to new donor, update to donor detail, search the donor details by given donor No or by date etc.
- ❖ **DONOR DETAILS:** Donor registration should include the following details every donor has a donor number which is unique.
 - Date.
 - Donor number:
 - Donor name:
 - D.O.B:
 - Gender:
 - Address:
 - City:
 - Pin:
 - State:
 - Contact No:
 - E-mail ID:
 - Blood group:
 - Last donated:
- ❖ **STAFF FORM:** This is also a master form this is used for a staff or employee .we store all the detail about the entire staff or employee here. The details of the employees are kept so that whenever reference is required, it may be referred. All the Details are kept so that they can be managed properly.

❖ **STAFF DETAILS:** Staff registration should include the following details.

- Employee number :
- Date of joining:
- Name:
- Address:
- Gender:
- Qualification:
- Contact no:
- Designation:
- E-mail id:

❖ **BLOOD FORM:** This is a master form; in this form we store all the information about the blood group and the blood unit which is store in a stock. Blood is collected in different type of packs. They are double, triple etc. We can Stored and preserved for later use in blood transfusion.

❖ **BLOOD DETAILS:** Blood registration should include the following details.

- Blood number:
- Blood group:
- No of units:

❖ **PATIENT FORM:** This is a master form, in this form we store detail about patients requests for the blood components with the request they bring the blood sample of the patient. Which patient comes on which date and for which type of blood they need like plasma, platelets or RBC and how much unit they needed. This from collects all the necessary personal information for new patient.

❖ **PATIENT DETAILS:** Patient registration should include the following details.

- Patient no:
- Date :
- Name:
- Date of birth:
- Address:
- City:
- Contact no:
- Pin:
- Blood group:

❖ **CAMP FORM:** This is the last master form. In this form we kept details of every camp organization. Which has been organized a blood donation camp or which is ready to Organized. We store all the information about the organization like organization name the date when they organized or location. The camp organization have to book for an event in advance.

❖ **CAMP DETAILS:** Camp registration should include the following details.

- Date :
- Camp ID:
- Parent organization:
- Address :
- Group name:
- Contact:
- Person1:
- Phone no 1:
- Person 2:
- Phone no 2:
- Person 3:
- Phone no 3:

➤ **TRANSACTION FORM**

❖ **BILLING FORM:** This is the transaction form. In this form we make a bill for patient by this we can collect daily cash amount or by this we can find how much cash is daily collected. This form is used for keeping and monitoring the payment being made. The payments are made in installments and proper monitoring should be done.

❖ **BILLING DETAILS:** Billing registration should include the following details.

- Bill no :
- Date :
- Referred by:
- Issued to(patient no):
- Blood type:
- Amount:
- Pay status:

❖ **DONATION BY CAMP FORM:** This is the transaction form. In this form we write about camp which has been organized in referred location and how much unit they collect from there and in what date. If any existing organization comes again to Organized a camp then we search camp detail by giving camp ID.

❖ **DBC DETAILS:** DBC registration should include the following details.

- Group ID:
- Group name:
- Address:
- Parent organization:
- Date:
- Units:
- Technical support:

❖ **BLOOD DONATION FORM:** This is a transaction form. This form have all the detail about every existing donor if a donor comes and he/she said they are regular donor then we ask about the given Donor No which we give to every donor after donating the blood by this we find out the time Elapse if it is above 90 days then donor is capable for donating blood.

❖ **BLOOD DONATION DETAILS:** Blood donation registration should include the following details.

- Donor no:
- Date:
- Blood unit no:
- Last donated:
- Time elapse:
- Age :
- No. of units:

❖ **BLOOD ISSUE FORM:** This is the last transaction form. We kept here all the records of issued blood or issued to which patient. They issue the components to the patient according to the compatibility report.

❖ **BLOOD ISSUE DETAILS:** Blood issue registration should include the following details.

- Date :
- Issue type:
- Blood type:
- Issued to(patient no):
- No. of units:
- Blood group:
- Blood units no:

REPORTS INVOLVED IN THE SYSTEMS

- ❖ **DONOR REPORT:** This report contains details about donor who comes to donate blood in a blood bank. Through this report we can search the donors who have donated blood earlier by name, location and date.
- ❖ **STAFF REPORT:** This report contains details about staff who are working in a blood bank. Through this report we can search staffs name, location or date of joining.
- ❖ **ISSUE REPORT:** This report contains detail about blood issue in a blood bank that how much blood is issued to which patient.
- ❖ **DAILY BLOOD COLLECTION REPORT:** This report is used to show the daily blood collected by a blood bank or by a blood camp. This report is helpful for the management.
- ❖ **DAILY CASH COLLECTED REPORT:** This report is used to show the daily cash collection in a blood bank. By this report a blood bank management is aware about how much cash should be collected daily. All the payments are acknowledged and the records of the payment kept here.

CHAPTER 5 SYSTEM TESTING

5. SYSTEM TESTING

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. In fact, testing is the one step in the software engineering process that could be viewed as destructive rather than constructive. A strategy for software testing integrates software test case design methods into a wellplanned series of steps that result in the successful construction of software. Testing is the set of activities that can be planned in advance and conducted systematically. The underlying motivation of program testing is to affirm software quality with methods that can economically and effectively apply to both strategic to both large and small-scale systems.

The software engineering process can be viewed as a spiral. Initially system engineering defines the role of software and leads to software requirement analysis where the information domain, functions, behavior, performance, constraints and validation criteria for software are established. Moving inward along the spiral, we come to design and finally to coding. To develop computer software we spiral in along streamlines that decrease the level of abstraction on each turn. A strategy for software testing may also be viewed in the context of the spiral. Unit testing begins at the vertex of the spiral and concentrates on each unit of the software as implemented in source code. Testing progress by moving outward along the spiral to integration testing, where the focus is on the design and the construction of the software architecture. Talking another turn on outward on the spiral we encounter validation testing where requirements established as part of software requirements analysis are validated against the software that has been constructed. Finally we arrive at system testing, where the software and other system elements are tested as a whole.

- White-box testing,
- Black-box testing,
- Unit testing,
- Integration testing,
- System testing,
- Acceptance testing,
- Validation testing

5.1 WHITE-BOX TESTING

White-box testing can be applied at the unit, integration and system levels of the **software testing** process, it is usually done at the unit level. It can test paths within a unit, paths between units during integration, and between subsystems during a system-level test. Though this method of test design can uncover many errors or problems, it might not detect unimplemented parts of the specification or missing requirements.

Techniques used in white-box testing include:

1. API testing (application programming interface) – testing of the application using public private APIs.
2. Code coverage – creating tests to satisfy some criteria of code coverage (e.g., the test can create tests to cause all statements in the program to be executed atleast once).
3. Fault injection methods – intentionally introducing faults to gauge the efficacy of testing strategies.
4. Mutation testing methods – testers changes specific components of source code to ensure a software test suite will be able to detect the changes.
5. Static testing methods – examination of a program, along with any associated documents, but does not require the program to be executed.

Code coverage tools can evaluate the completeness of a test suite that was created with any method, including black-box testing. This allows the software team to examine parts of a system that are rarely tested and ensures that the most important function points have been tested. Code coverage as a software metric can be reported as a percentage for:

1. Function coverage, which reports on functions executed
2. Statement coverage, which reports on the number of lines executed to complete the test

100% statement coverage ensures that all code paths, or branches (in terms of control flow) are executed at least once. This is helpful in ensuring correct functionality, but not sufficient since the same code may process different inputs correctly or incorrectly.

5.2 BLACK-BOX TESTING

Black-box testing treats the software as a "black box", examining functionality without any knowledge of internal implementation. The tester is only aware of what the software is supposed to do, not how it does it. Black-box testing methods include: equivalence partitioning, boundary value analysis, all-pairs testing, state transition tables, decision table testing, fuzz testing, model-based testing, use case testing, exploratory testing and specification-based testing.

Specification-based testing aims to test the functionality of software according to the applicable requirements. This level of testing usually requires thorough test cases to be provided to the tester, who then can simply verify that for a given input, the output value (or behavior), either "is" or "is not" the same as the expected value specified in the test case. Test cases are built around specifications and requirements, i.e., what the application is supposed to do. It uses external descriptions of the software, including specifications, requirements, and designs to derive test cases. These tests can be functional or non-functional, though usually functional.

Specification-based testing may be necessary to assure correct functionality, but it is insufficient to guard against complex or high-risk situations.

One advantage of the black box technique is that no programming knowledge is required. Whatever biases the programmers may have had, the tester likely has a different set and may emphasize different areas of functionality. On the other hand, black-box testing has been said to be "like a walk in a dark labyrinth without a flashlight. Because they do not examine the source code, there are situations when a tester writes many test cases to check something that could have been tested by only one test case, or leaves some parts of the program untested.

5.3 UNIT TESTING

Unit testing, also known as component testing, refers to tests that verify the functionality of a specific section of code, usually at the function level. In an object-oriented environment, this is usually at the class level, and the minimal unit tests include the constructors and destructors.

These types of tests are usually written by developers as they work on code (whitebox style), to ensure that the specific function is working as expected. One function might have multiple tests, to catch corner cases or other branches in the code. Unit testing alone cannot verify the functionality of a piece of software, but rather is used to assure that the building blocks the software uses work independently of each other.

Unit testing is a software development process that involves synchronized application of a broad spectrum of defect prevention and detection strategies in order to reduce software development risks, time, and costs. It is performed by the software developer or engineer during the construction phase of the software development lifecycle. Rather than replace traditional QA focuses, it augments it. Unit testing aims to eliminate construction errors before code is promoted to QA; this strategy is intended to increase the quality of the resulting software as well as the efficiency of the overall development and QA process.

Depending on the organization's expectations for software development, unit testing might include static code analysis, data flow analysis metrics analysis, peer code reviews, code coverage analysis and other software verification practices.

5.4 INTEGRATION TESTING:

Integration testing is any type of software testing that seeks to verify the interfaces between components against a software design. Software components may be integrated in an iterative way or all together ("big bang"). Normally the former is considered a better practice since it allows interface issues to be located more quickly and fixed.

Integration testing works to expose defects in the interfaces and interaction between integrated components (modules). Progressively larger groups of tested software components corresponding to elements of the architectural design are integrated and tested until the software works as a system.

5.5 SYSTEM TESTING:

System testing tests a completely integrated system to verify that it meets its requirements.

In addition, the software testing should ensure that the program, as well as working as expected, does not also destroy or partially corrupt its operating environment or cause other processes within that environment to become inoperative (this includes not corrupting shared memory, not consuming or locking up excessive resources and leaving any parallel processes unharmed by its presence).

5.6 ACCEPTANCE TESTING:

The acceptance test suite is run against the supplied input data or using an acceptance test script to direct the testers. Then the results obtained are compared with the expected results. If there is a correct match for every case, the test suite is said to pass. If not, the system may either be rejected or accepted on conditions previously agreed between the sponsor and the manufacturer.

The objective is to provide confidence that the delivered system meets the business requirements of both sponsors and users. The acceptance phase may also act as the final quality gateway, where any quality defects not previously detected may be uncovered.

A principal purpose of acceptance testing is that, once completed successfully, and provided certain additional (contractually agreed) acceptance criteria are met, the sponsors will then sign off on the system as satisfying the contract (previously agreed between sponsor and manufacturer), and deliver final payment.

Acceptance testing can mean one of two things:

5.6.1 A smoke test is used as an acceptance test prior to introducing a new build to the main testing process, i.e. before integration or regression.

5.6.2 Acceptance testing performed by the customer, often in their lab environment on their own hardware, is known as user acceptance testing (UAT). Acceptance testing may be performed as part of the hand-off process between anytwo phases of development.

5.7 VALIDATION TESTING:

Validation is intended to check that development and verification procedures for a product, service, or system (or portion thereof, or set thereof) result in a product, service, or system (or portion thereof, or set thereof) that meets initial requirements. For a new development flow or verification flow, validation procedures may involve modeling either flow and using simulations to predict faults or gaps that might lead to invalid or incomplete verification or development of a product, service, or system (or portion thereof, or set thereof). A set of validation requirements, specifications, and regulations may then be used as a basis for qualifying a development flow or verification flow for a product, service, or system (or portion thereof, or set thereof). Additional validation procedures also include those that are designed specifically to ensure that modifications made to an existing qualified development flow or verification flow will have the effect of producing a product, service, or system (or portion thereof, or set thereof) that meets the initial design requirements, specifications, and regulations; these validations help to keep the flow qualified. It is a process of establishing evidence that provides a high degree of assurance that a product, service, or system accomplishes its intended requirements. This often involves acceptance of fitness for purpose with end users and other product stakeholders. This is often an external process.

Databases to be expected in the proposed system:

1. Login table

ATTRIBUTE NAME	DATA TYPE	FEATURE
User ID	Text	Primary Key
Password	Text	

2. Donor table

ATTRIBUTE NAME	DATA TYPE	FEATURE
Donor no	Text	Primary Key
Donor no	Text	
Occupation	Text	
Father's name	Number	
Date	Text	
D.O.B	Text	
Gender	Text	
Address	Text	
City	Text	

State	Text	
Pin	Text	
Contact	Text	
e-mail id	Text	
Blood group	Text	

3. Staff table

ATTRIBUTE NAME	DATA TYPE	FEATURE
Employee no	Number	Primary Key
Date of joining	Text	
Name	Text	
Address	Text	
Gender	Text	
Qualification	Text	
Contact	Text	
Designation	Text	
E mail	Text	

4. Blood table

ATTRIBUTE NAME	DATA TYPE	FEATURES
Blood group	Text	Primary Key
Blood unit	Text	
Blood number	Text	

5. Patient table

ATTRIBUTE NAME	DATA TYPE	FEATURES
Patient no	Number	Primary key
Date	Text	
Name	Text	
Date of birth	Text	
Address	Text	
City	Text	
Contact	Text	
Pin	Text	
Blood group	Text	

6. Camp table

ATTRIBUTE NAME	DATA TYPE	FEATURES
DATE	Text	Primary key
CAMP ID	Text	
Organization	Text	
Address	Text	
Group	Text	
Person 1	Text	
Phone 1	Text	
Person 2	Text	

Person 3	Text	
Phone 3	Text	

7. Bill table

ATTRIBUTE NAME	DATA TYPE	FEATURES
Bill no	Text	Primary key
Date	Text	
Reffered by	Text	
Patient no	Number	
Bill type	Text	
Amount	Text	
Status	Text	

8. Donation by camp table

ATTRIBUTE NAME	DATA TYPE	FEATURES
Camp id	Text	Primary key
Group name	Text	
Address	Text	
Parent origination	Text	
Date	Text	
Unit	Text	
Technical support	Text	

9. Blood Donation table

ATTRIBUTE NAME	DATA TYPE	FEATURES
Donor no	Number	PRIMARY KEY
Date	Text	
Last donated	Text	
BLOOD units NO	Text	
AGE	Text	
TIME ELAPSED	Text	
NO. OF UNITS	Text	

10. Blood issue table

ATTRIBUTE NAME	DATA TYPE	FEATURES
Date	Text	PRIMARY KEY
Issue type	Text	
Blood type	Text	
Issued to	Text	
No of units	Text	
Blood group	Text	
Blood unit no	Text	

CHAPTER 6: INTERFACE DESIGN

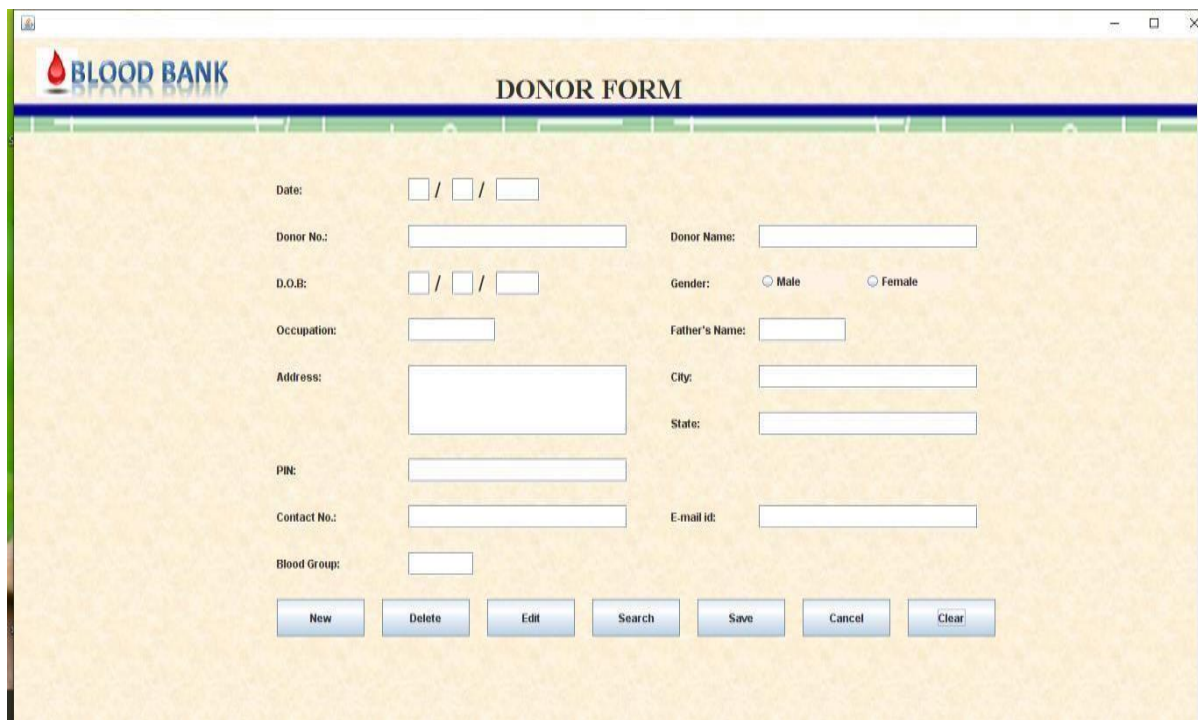
INTERFACE DESIGN

LOGIN FORM



The screenshot shows a web browser window titled "BLOOD BANK" with a "Login Form" centered on the page. The form includes two input fields: "User Name:" and "Password:". The password field contains seven asterisks. Below the fields are two buttons: "Login" and "Cancel".

DONOR FORM



The screenshot shows a web browser window titled "BLOOD BANK" with a "DONOR FORM" centered on the page. The form contains the following fields and controls:

- Date: / /
- Donor No.:
- D.O.B: / /
- Occupation:
- Address:
- PIN:
- Contact No.:
- Blood Group:
- Donor Name:
- Gender: Male Female
- Father's Name:
- City:
- State:
- E-mail id:

At the bottom of the form are seven buttons: "New", "Delete", "Edit", "Search", "Save", "Cancel", and "Clear".

STAFF FORM

The screenshot shows a web application window titled "BLOOD BANK" with a sub-header "Staff Form". The form contains the following fields and controls:

- Employee No:
- Date of Joining: / /
- Name:
- Address:
- Gender: Male Female
- Qualification:
- Contact No:
- Designation:
- E-mail id:

At the bottom of the form, there are seven buttons: New, Delete, Edit, Search, Save, Cancel, and Clear.

PATIENT INFORMATION

The screenshot shows a web application window titled "BLOOD BANK" with a sub-header "Patient Information". The form contains the following fields and controls:

- Patient No:
- Date: / /
- Name:
- D.O.B: / /
- Address:
- City:
- Pin:
- Contact No:
- Blood Group:

At the bottom of the form, there are seven buttons: New, Delete, Edit, Search, Save, Cancel, and Clear.

COLLECTION CAMP

The screenshot shows a web application window titled "BLOOD BANK" with a sub-header "Collection Camp". The form contains the following fields:

- Date: / /
- Camp ID:
- Parent Org.:
- Address:
- Group Name:
- Contact:
- Person 1:
- Phone No. 1:
- Person 2:
- Phone No. 2:
- Person 3:
- Phone No. 3:

At the bottom, there are seven buttons: New, Edit, Search, Save, Cancel, Delete, and Clear.

BILL

The screenshot shows a web application window titled "BLOOD BANK" with a sub-header "BILL". The form contains the following fields:

- BILL No.:
- Date: / /
- Referred By:
- Issued To:
- Type:
- Amount:
- Pay Status:

At the bottom, there are seven buttons: New, Delete, Edit, Search, Save, Cancel, and Clear.

DONATION BY CAMP

BLOOD BANK Donation By Camp

Camp ID: Refresh

Group Name: Address:

Parent Org.:

Date: / /

Units:

Technical Supp...:

BLOOD DONATION

BLOOD BANK BLOOD DONATION

Donor No.: Date: / /

Blood Unit No.:

Last Donated: / / Time Elapse:
(Since Last Donation)

Age: No. Of Units:

BLOOD ISSUE

BLOOD BANK

BLOOD ISSUE

Date: / /

Type:

Issued To:
(Patient No.)

Blood Group:

Issue type: Replacement
 Voluntary

No of units:

Blood Unit No:

DONOR DETAILS

BLOOD BANK

Donor Details

Search By: Name
 Location
 Date

Donor No.:

Date:

City:

Donor Name:

State:

Occupation:

PIN:

Father's Name:

Contact No.:

D.O.B.:

E-mail id:

Gender:

Blood Group:

Address:

Last Donated:

Date	Name	Donor No.	Gender	Blood Group	Location	Contact
13/03/2021	Sabyasachi	17	Male	A+	Dimna	34567899999

STAFF DETAILS

Report

BLOOD BANK

Staff Details

Employee no.: E-mail:

Date of joining:

Name:

Address:

Gender:

Qualification:

Contact:

Designator:

Search By: Name
 Location
 Date of joining

Date of joining	Name	Employee No.	Gender	Location	Contact
-----------------	------	--------------	--------	----------	---------

BLOOD STOCK DETAILS

Report

BLOOD BANK

Blood Stock Details

Blood Unit No.:

Blood Group:

Status:


Search By: Group
 Status

Blood Unit No.	Blood Group	Status
AA0001	B+	INP
AA0002	AB-	INP
AA0003	O-	INP
AA0004	O-	INP
AA0005	B-	INP
AA0006	O+	INP
AA0007	A+	INP
AA0008	B+	INP
AA0009	B-	INP
AA0010	AB-	INP
AA0011	A+	INP

Total Units:

CASH COLLECTION DETAILS

Cash Report

 **Cash Collection Details**

Search By: Date Patient No. Blood Type

Bill No.:
Date:
Referred By:
Patient No.:
Blood Type:
Amount:
Status:

Bill No.	Date	Referred By	Patient No.	Blood Type	Amount	Status
152	23/12/2015	er	13	RBC	fer	df

CHAPTER 7: RESULTS AND DISCUSSIONS

FUTURE SCOPE OF SYSTEM:

This application is built such a way that it should suits for all type of blood bank in future. So every effort is taken to implement this project in this blood bank, on successful implementation`in this blood bank, we can target other blood banks in a city. The system will provide the user the option to look at the details of the existing donor list, blood group and to add new donor. The administrator can alter all the systemdata.

DRAWBACK OF THE EXISTING SYSTEM.

- Maintenance and updating of blood bank records, currently, are done manually which, at times, results in loss of critically important records or delay information updates.
- Difficult to calculate amount of blood in stock.
- Difficulties in retrieving data (for any specific reason).
- Difficulty in identifying blood expiry date.
- Security of records is minimum.
- Records are very liable to be misfiled or soiled.

CONCLUSION:

Blood bank is a software application to maintain day to day transactions in a blood bank. "BLOOD BANK MANAGEMENT SYSTEM" is a software application to maintain day to day transaction in a blood bank. This software help to register all the Donors, Blood collection details, Blood issued details etc. BLOOD BANK MANAGEMENT SYSTEM in java is great project. The basic building aim is to provide blood donation service to the city. The major task Blood Bank system to provide blood to help people who want to blood. This blood bank system project manage the all kind of information related to Blood.