ASSIGNMENT
OF
ADVANCED SOFTWARE ENGINEERING

SUBMITTED TO

ANIT JAMES

SUBMITTED BY

TOM JOSEPH

MCA-1 YEAR

S.R.S

BLOOD BANK MANAGEMENT SYSTEM

ABSTRACT

I had developed “BLOOD BANK MANAGEMENT SYSTEM” is the convenient way to designed for successful completion of project on blood bank management system. The basic building aim is to provide blood donation service to the city recently. This project aims at maintaining all the information pertaining to blood donors, different blood groups available in each blood bank and help them manage in a better way.

TABLE OF CONTENT

INTRODUCTION

1. INTRODUCTION

1.2 PROBLEM OVERVIEW

The main objective of the proposed system is to eliminate the limitations of the existing manual system .Computers are fast tireless machines that are provided large amount data quickly and deficiently and give the output in the required format. Cost reduction is one of the objectives of the existing system. This achieved by reduced labor and data maintains speed of the transaction can be increased by computerized system since the system computerised. There is no need of so many employees. Thus the cost can be reduced. Since all details are stored in the computer searching became easy. We have developed "BLOOD BANK MANAGEMENT SYSTEM" is the convenient way to Designed for successful completion of project on blood bank management system. The basic building aim is to provide blood donation service to the city recently. This project aims at maintaining all the information pertaining to blood donors, different blood groups available in each blood bank and help them manage in a better way.

SYSTEM ANALYSIS

2. SYSTEM ANALYSIS

2.1 PRINCIPLES OF SYSTEM ANALYSIS

System analysis is the detailed study of various operations and their relationships within or outside the system. It is a first phase of a project development. System analysis is concerned with becoming aware of the problem, identifying the relevant and most decisional variables, analyzing and synthesizing the various factors and determining an optional or at least a satisfactory or program of action.

System analysis phases identifies and analysis and then evaluates the information system under consideration in an organisation.One aspects of analysis is the defining boundaries of the system and determining whether or not a candidate system should consider other related system during analysis, data are collected on the available files. Decision points and transaction handled by the presented system. Data flow diagrams, interviews, on-site observation and questionnaires are the examples for the tool analysis.

On the analysis completed the analysis has a form of understanding of what is to be done. The next step is to analysis how the problem might be solved. Thus in a system design we move from the logical to the physical aspects of the life cycles.

2.2 EXISTING SYSTEM

The current system has a lot of paper work, requires large quantities of file cabinets, which are huge and require quite a bit of space in the office, which can be used for storing records of previous details.
The drawback of the existing system is that it is very difficult to retrieve data from case files. It is difficult to handle the whole system manually and it is less accurate and difficult to keep the data.

2.2.1 PROBLEM WITH EXISTING SYSTEM

- Manual work.
- Consumes large volume of paper work.
- Limited source only.
- Time consuming.
- It need experts.
- To avoid all these limitations and make the system working more.
- Accurately it needs to be computerized.
- Limited Record.

2.2.2 OBJECTIVES OF PROPOSED SYSTEM

- Registration and preparation of policy are making easy

- Agents and his customer records searching is possible

2.3 PROPOSED SYSTEM

The LIC Office Management System is user-friendly software. The main objectives is to create agents specialized in selling policies. It reduce the work of LIC office. The system is very simple in design and to implement. The system requires very low system resources and the system will work in almost all configurations

2.3.1 MODULES

1. ADMIN

Admin has full control over this website, he can add or remove donors and receptors from the website. Also admin can check user is genuine. Admin can check user feedbacks and take necessary action.

2. DONORS

Donors are type of users how donate blood. They can easily donate blood for requesting Receptors.

3. RECEPTORS

Receptors are type of users how want need blood.They can easily receive blood by requesting and entering some details.

2.3.2 ADVANTAGES OF PROPOSED SYSTEM

- Security of data.
- Ensure data accuracy.
- It is easy to use.
- Minimize manual data entry.
- Greater efficiency.
- User friendly and interactive.
- Minimum time required.

2.4 FEASIBILITY STUDY

In this project interviews and discussion are conducted to the staffs and employees to analyses the problem.

The feasibility study is the high level capsule version of staff and the entire system analysis and design process. The objective of the feasibility study is to determine Whether the proposed system is feasible. The feasibility study can be classified into the following categories.

- Cost/benefit feasibility
- Technical feasibility
- Operational feasibility

2.4.1 COST/BENEFIT FEASIBILITY

The main criterion that is involved in the creation of this system is to maintain records. The main problem with the existing manual system is record loss. The TMS solves the problem of record management. The chance of loss of any data is optimized to the minimal.

2.4.2 TECHNICAL FEASIBILITY

The assessment of technical feasibility must be based on an outline design of system requirement in terms of input output files programs and procedures. This can be qualified in terms of volumes of data, trends, frequency of updating, cycles of activity etc. in order to give an introduction of technical system. The technologies that we are using are PHP. This technology is easy to study and understand. The complexities in these technologies are very less. So our system is technically feasible

- Data keeping capacity of the proposed equipment to be used for the system are enough.
- Data retrieval for the various enquires are fast enough technically, according to the proposed hardware. The entire terminal user connected to the proposed system will get the adequate response.
- The proposed system is very easy in use, database security is very high, easy in access, and reliability and accuracy are enough.

  Considering the above facts the proposed system is fully technically feasible

2.4.3 OPERATIONAL FEASIBILITY

The developed system is completely driven and user friendly. Also the system is developed in PHP. There is little need skill for new user to operate the software. Reports will be exactly as

per the requirement. At the beginning of preliminary investigation work all the personnel approached responded positively this reduces the chance of resistance to the proposed system. Considering all the issue stated above makes the proposed system operationally feasible. In our organization the admin user (owner), staffs agency and customer use the software. So they need to be aware of the software initially. Then they can use it easily. So it is feasible.

PROGRAMMING ENVIRONMENT

3. PROGRAMMING ENVIORNMENT

3.1 HARDWARE CONFIGURATION

Hardware requirements needs for the system:

Processor	:	Intel Core i5
Manufacturer	:	intel
System type	:	64 bit
RAM	:	8GB DDR4
Monitor	:	LED

3.2 SOFTWARE CONFIGURATION

Operating System	:	Windows 7 or Higher Version
Operating System	:	Microsoft Windows 7
Front End	:	PHP
Back End	:	MYSQL

3.3 BACK END

MYSQL

MySQL is a relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases. A relational database stores data in separate tables rather than putting all data in one big storeroom. The tables are linked in by defined relations, making it possible to combine data from several tables on request. The SQL phrase stands for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. It is developed, distributed and supported by MySQL AB.

Free-software projects that require a full-featured database management system often use MySQL. Where the project may lead to something in commercial use, the license terms need careful study. The MySQL database has become the world's most popular open source database because of its high performance, high reliability and ease of use. It is also the database

of choice for a new generation of applications built on the LAMP stack (Linux, Apache, MySQL, PHP / Perl / Python). MySQL offers a comprehensive range of database tools, support, training and consulting services to make the project successful.

FEATURES: MySQL implements the following features, which some other RDBMS systems may not:

• Multiple storage engines, allowing one to choose the one that is most effective for each table in the application. In MySQL, storage engines can be dynamically loaded at run time.

• Commit grouping, gathering multiple transactions from multiple connections together to increase the number of commits per second.

The MySQL Extension (MySQL improved) is a relational database driver used in PHP programming language to provide an interface with MySQL databases. MySQL is an improved version of the older PHP MySQL driver, offering various benefits. The developers of the PHP programming language recommend using MySQL when dealing with MySQL server versions 4.1.3 or later. It has the following benefits:

- An object oriented interface.
- Support for prepared statements.
- Support for multiple statements.
- Support for transactions.
- Embedded server support.

3.4 FRONT END

PHP

PHP was originally created by Ramus Leadoff in 1995 and has been in continuous development ever since. PHP stands for "PHP: Hypertext Pre-processor". The main implementation of PHP is now produced by the PHP Group and serves as the de facto standard for PHP as there is no formal specification. PHP is free software released under the PHP License. PHP: Hypertext Pre-processor is a widely used, general-purpose scripting language that was originally designed for web development to produce dynamic web pages. For this purpose, PHP code is embedded into the HTML source document and interpreted by a web server with a PHP processor module, which generates the web page document. As a general-

purpose programming language, PHP code is processed by an interpreter application in command-line mode performing desired operating system operations and producing program output on its standard output channel. It may also function as a graphical application. PHP is available as a processor for most modern web servers and as a standalone interpreter on most operating systems and computing platforms.

PHP originally stood for personal home page. PHP is a server-side scripting language whose primary purpose is to generate HTML content. Three things make PHP popular. The first is that it is easy: easy to implement, easy to learn, and easy to use. The second is that it is free. The third is that it runs on almost any Web server on almost any platform currently available.

One of set modules control database access. Using PHP with MySQL has become common enough that the MySQL interface is now part of core PHP instead of a plug-in module. Most other databases have modules that can be included in a PHP build to allow access. PHP can access most any SQL or ODBC database. It can both read and write information in the database. This opens up the door for a whole variety of online business applications that require data storage on the server. Because of this, PHP is becoming an increasingly popular tool for e-commerce.

FEATURES: The core features of PHP are built around the ability to process strings and arrays, as well as to work as an object-oriented programming language. Beyond this most of PHP is a collection of modules that can be added in on the server as needed to perform a large variety of specific tasks? In other words, it is a highly customizable application, and you can keep it small by only installing as much as you need to perform required tasks.

- Access Control - A built-in web-based configuration screen handles access control configuration. It is possible to create rules for all or some web pages owned by a certain person which place various restrictions on who can view these pages and how they will be viewed. Pages can be password protected, completely restricted, logging disabled and more based on the client's domain, browser, e-mail address or even the referring document.
- PHP is one of the most popular open-source scripting languages. Open-source means free, so you can use PHP as often as you like without having to pay for licenses or support. It is a full-featured programming language, capable of managing very large database-driven online environments.

- PHP works in conjunction with HTML. If you are already familiar with HTML, making the jump to PHP is simple, and the two languages are interchangeable on the page. PHP can add new features to your site, but the appearance will be determined by HTML.

- PHP is a server-side language. This means that PHP has access to all the information the server has, and very little access to information the client has. In fact, PHP only has access to client information that the client himself has given to the server. The fact that PHP is a server-side language makes it very useful for:
    - Embedding dynamic text into static text
    - Integrating databases with websites
- C programs are 'pre-compiled', which in short means that they need to be re-compiled every time you switch machines. Most Internet Service Providers, or ISPs, do not provide you with a C compiler. Using PHP avoids this problem. If it works on one server, it will generally work on any other server that has it. And, most ISPs that provide server-side scripting provide PHP. So the final benefit is portability.

HTML

HTML, which stands for Hyper Text Mark-up Language, is the predominant mark-up language for web pages. HTML is the basic building-blocks of web pages. A mark-up language is a set of mark-up tags, and HTML uses mark-up tags to describe web pages. The purpose of a web browser is to read HTML documents and compose them into visual or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page. HTML elements form the building blocks of all websites.

HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts in languages such as JavaScript which affect the behaviour of HTML web pages.

Web browsers can also refer to Cascading Style Sheets (CSS) to define the appearance and layout of text and other material. The W3C, maintainer of both the HTML and the CSS standards, encourages the use of CSS over explicitly presentational HTML mark-up.

CSS

Cascading Style Sheet (CSS) is a style sheet language used to describe the presentation semantics (the look and formatting) of a document written in a Mark-up language. It's most common application is to style web pages written in HTML and XHTML, but the language can also be applied to any kind of XML document, including plain XML, SBG and XUL.

CSS is designed primarily to enable the separation of document content (written in HTML or a similar mark-up language) from document presentation, including elements such as the layout, colour and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple pages to share formatting, and reduce complexity and repetition in the structural content(such as by allowing for table less web design). CSS can also allow the same mark-up page to be presented in different styles for different rendering methods, such as on-screen, in print, by voice (when read out by a speed-based browser or screen reader) and on Braille-based, tactile devices.

XAMPP

XAMPPs are packages of independently-created programs installed on computers that use any modern operating systems (Microsoft Windows, MacOS, or Linux). XAMPP is an acronym formed from the initials of the operating system X (any OS) and the principal components of the package: Apache, MySQL and one of PHP, Perl or Python. Apache is a web server. MySQL is an open-source database. Other programs may also be included in a package, such as phpMyAdmin which provides a graphical user interface for the MySQL database manager, or the alternative scripting languages Python or Perl.

JAVASCRIPT

JavaScript is a prototype-based scripting language that is dynamic, weekly, typed and has first-class function. It is a multi-paradigm language, supporting object-oriented, imperative, and functional programming styles.

JavaScript was formalized in the ECMA Script language standard and is primarily used in the form of client-side JavaScript, implemented as part of a Web browser in order to provide enhanced user interfaces and dynamic websites. This enables programmatic access to computational objects within a host environment.

JavaScript's use in applications outside Webpages--for example in PDF documents, site-specific browses and desktop widgets-is also significant. Newer and faster JavaScript VMs and

frameworks built up on them (notably Nobe.js) have also increased the popularity of JavaScript for server-side Web applications.

JavaScript user's syntax influenced by that of JavaScript copies many names and naming conventions from Java, but the two languages are otherwise unrelated and have very different semantics. The key design principles within JavaScript are taken from the self and scheme programming languages.

3.5 FEATURES OF OPERATING SYSTEM

Windows 10 delivers a refined, vastly improved vision for the future of computing with an operating system that's equally at home on tables and traditional PCs--and it's a free upgrade for most users. Windows 10 is the Goldilocks version of Microsoft's venerable PC operating system-- a "just right" compromise between the familiar dependability of Windows 7, and the forward -looking touch screen vision of Windows 8.

This new Windows is built from the ground up to pursue Microsoft's vision of a unified OS that spans all devices without alienating any one platform. It's an attempt to safeguard Microsoft's crumbling software hegemony, assailed on all sides by Google and Apple.

This new OS is chock-full of fresh features. To name just a few : a lean, fast Internet Explorer replacement called Edge; Microsoft's Sire-like voice-controlled virtual assistant, Cortina; and the ability to stream real-time games to your desktop from an Xbox One in another room. And in case you're wondering: there is no "Windows 9"--Microsoft skipped it, going straight from 8 to 10.

Window 10 is a welcome return to form. The Start menu, inexplicably yanked from 8, is back and working the way you expect it to. Those live tiles from the Windows 8 home screen still exist, but they've been attached to the Start menu, where they make a lot more sense. And the fiendishly hidden Charms bar has been morphed into the more straightforward (and easier to find) Action Centre.

Windows 10 is the first step to an era of more personal computing, one in which Microsoft is moving Windows from its heritage of enabling a single device-the PC -to a world that is more mobile, natural, and grounded in trust. With Windows 10, applications, services, and content move across devices seamlessly and easily. Windows 10 features a universal app platform and universal store, providing a consistent experience across devices.

SYSTEM DESIGN

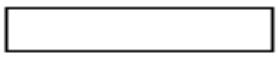
4. SYSTEM DESIGN

The system design is the most creative and challenging phase of system development life cycle. It is an approach for the creation of proposed system, in which the logic and details structure of the proposed system is designed, which will help the system coding. The most creative and challenging phase of the system development process is design phase it is a solution, how to approach to the creation of the proposed system. Design is the first step in the development of the engineered product is initiated only after a clear exposition of expected product is available. System Design is vital for efficient database management. It provides the understanding of procedural details necessary for implementing the system .A number of sub- systems is to be identified which constitute the whole system

4.1 DATABASE DESIGN

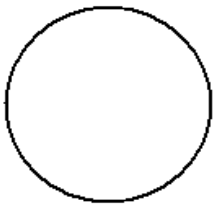
4.1.1 DATA FLOW DIAGRAM

The symbols used in DFD are shown below



Source and destination of data

Data flow

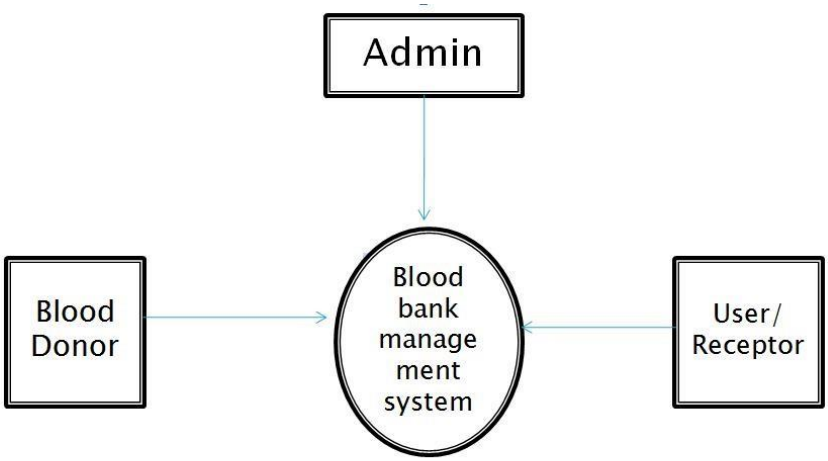


Process that transforms the data

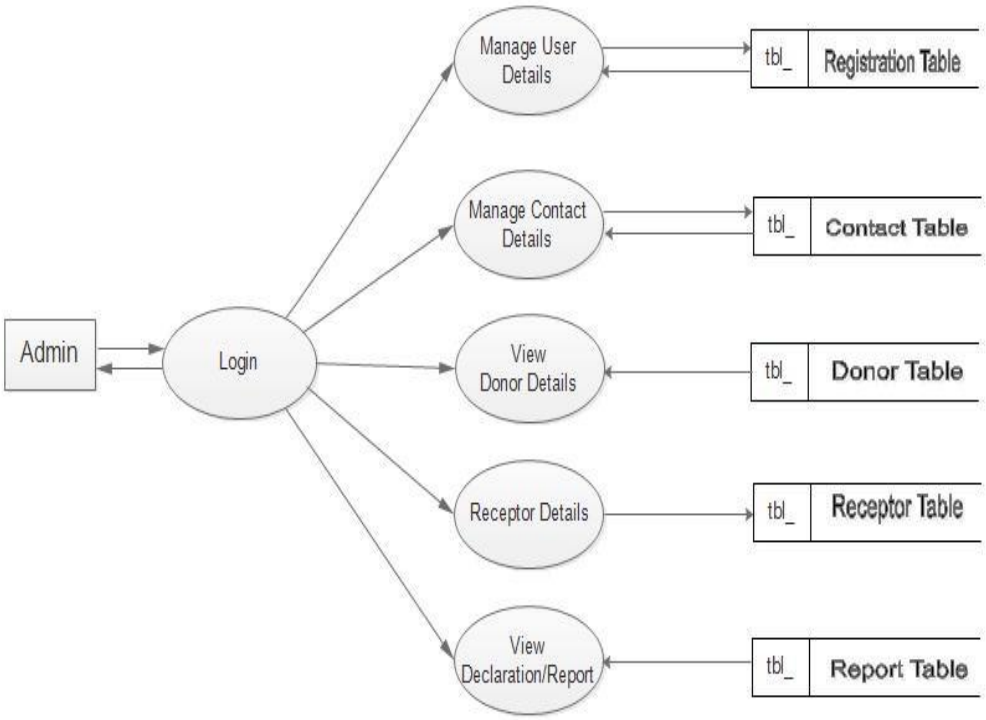
Data store

A circle is used to represent a process. A rectangle is used to represent sources and destination of data. These are called external entities, entities that supply data are known as sources and those that consume data are called destinations. An open rectangle is used to represent a data store and arrows represent data flows. In addition, the arrows show the direction of data flows.

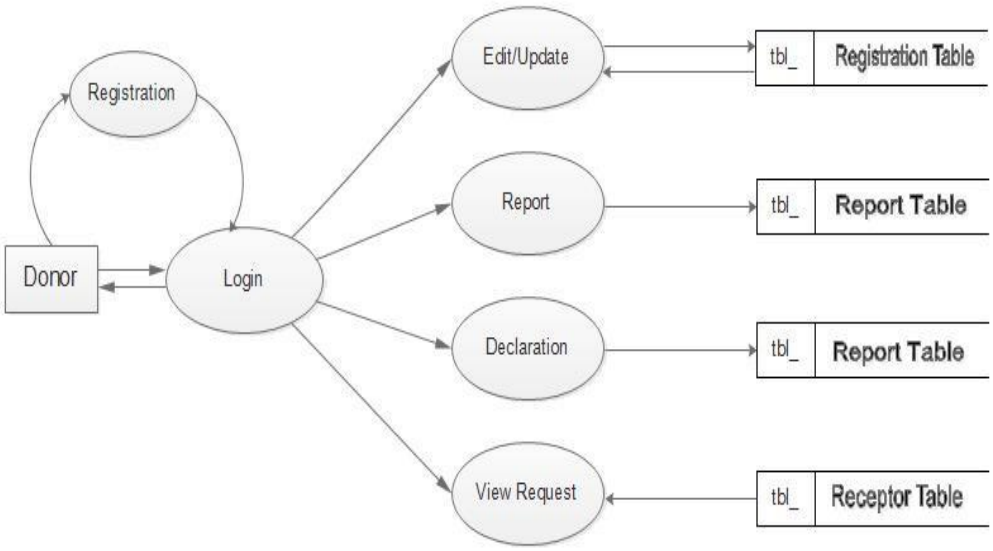
Context Diagram/Zeroth level



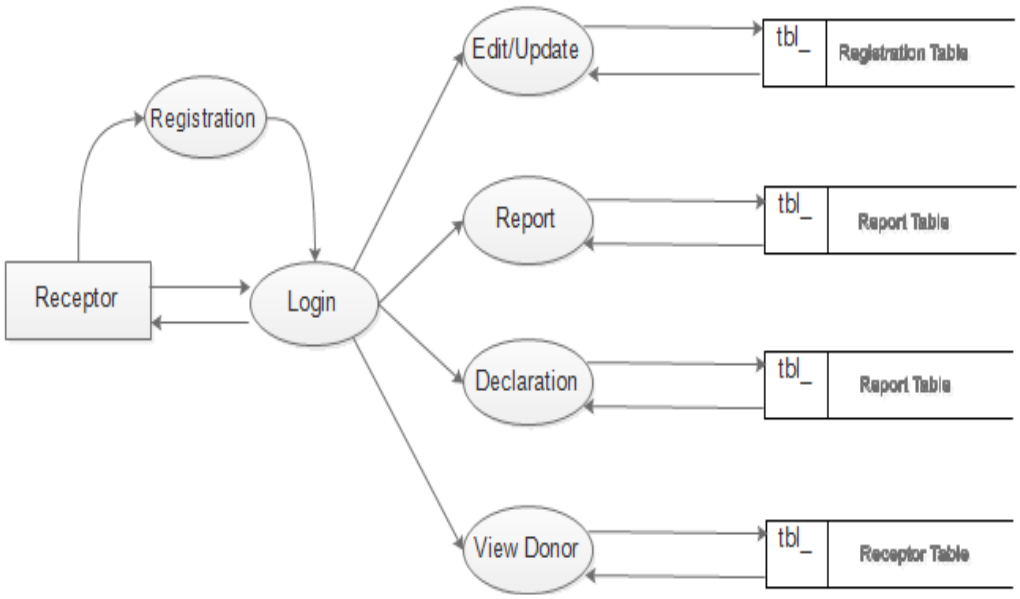
LEVEL ONE DFD



LEVEL TWO DFD



LEVEL THREE DFD



4.1.2 TABLES

Table 1: Admin Table

Primary key: Login_id

	Field	Type	Collation	Attributes	Null	Default	Extra
☐	login_id	int(11)			No	*None*	AUTO_INCREMENT
☐	admin_name	varchar(10)	latin1_swedish_ci		No	*None*	
☐	admin_password	varchar(10)	latin1_swedish_ci		No	*None*	
☐	login_type	varchar(10)	latin1_swedish_ci		No	*None*	

Table 2: Registration Table

Primary key: registration_id

	Field	Type	Collation	Attributes	Null	Default	Extra
☐	registration_id	int(11)			No	*None*	AUTO_INCREMENT
☐	registration_name	varchar(15)	latin1_swedish_ci		No	*None*	
☐	registration_age	int(2)			No	*None*	
☐	registration_email	varchar(15)	latin1_swedish_ci		No	*None*	
☐	registration_mob	int(10)			No	*None*	
☐	registration_city	varchar(15)	latin1_swedish_ci		No	*None*	
☐	registration_state	varchar(10)	latin1_swedish_ci		No	*None*	
☐	registration_pincode	int(6)			No	*None*	
☐	registration_bloodgroup	varchar(3)	latin1_swedish_ci		No	*None*	
☐	registration_username	varchar(10)	latin1_swedish_ci		No	*None*	
☐	registration_password	varchar(15)	latin1_swedish_ci		No	*None*	
☐	registration_address	varchar(100)	latin1_swedish_ci		No	*None*	
☐	registration_gender	varchar(10)	latin1_swedish_ci		No	*None*	
☐	registration_status	int(11)			No	*None*	
☐	login_type	varchar(10)	latin1_swedish_ci		No	user	

TABLE 3: Contact Table

Primary key: contact_id

	Field	Type	Collation	Attributes	Null	Default	Extra
☐	contact_id	int(11)			No	*None*	AUTO_INCREMENT
☐	contact_message	varchar(100)	latin1_swedish_ci		No	*None*	
☐	contact_name	varchar(10)	latin1_swedish_ci		No	*None*	
☐	contact_email	varchar(30)	latin1_swedish_ci		No	*None*	
☐	contact_topic	varchar(10)	latin1_swedish_ci		No	*None*	

TABLE 4: Donor Table

Primary key: donor_id

	Field	Type	Collation	Attributes	Null	Default	Extra
☐	donor_id	int(11)			No	*None*	AUTO_INCREMENT
☐	donor_name	varchar(15)	latin1_swedish_ci		No	*None*	
☐	donor_phoneno	int(10)			No	*None*	
☐	donor_city	varchar(15)	latin1_swedish_ci		No	*None*	
☐	donor_state	varchar(10)	latin1_swedish_ci		No	*None*	
☐	donor_last_blood_donated	date			No	*None*	
☐	donor_blood	varchar(6)	latin1_swedish_ci		No	*None*	
☐	donor_email	varchar(20)	latin1_swedish_ci		No	*None*	
☐	donor_status	int(11)			No	*None*	

TABLE 5: Receptor Table

Primary key: recep_id

Field	Type	Collation	Attributes	Null	Default	Extra
recep_id	int(11)			No	*None*	AUTO_INCREMENT
receptor_name	varchar(15)	latin1_swedish_ci		No	*None*	
receptor_phone	text	latin1_swedish_ci		No	*None*	
receptor_city	varchar(15)	latin1_swedish_ci		No	*None*	
receptor_state	varchar(10)	latin1_swedish_ci		No	*None*	
receptor_required	int(11)			No	*None*	
receptor_amount	int(11)			No	*None*	
receptor_blood	varchar(6)	latin1_swedish_ci		No	*None*	
receptor_email	varchar(15)	latin1_swedish_ci		No	*None*	
receptor_status	int(11)			No	*None*	

TABLE 6: Report Table

Primary key: report_id

	Field	Type	Collation	Attributes	Null	Default	Extra
☐	report_id	int(11)			No	*None*	AUTO_INCREMENT
☐	report_name	varchar(15)	latin1_swedish_ci		No	*None*	
☐	report_phone	int(10)			No	*None*	
☐	report_usertype	varchar(8)	latin1_swedish_ci		No	*None*	
☐	report_email	varchar(15)	latin1_swedish_ci		No	*None*	
☐	report_report_type	varchar(9)	latin1_swedish_ci		No	*None*	
☐	report_user_report	varchar(100)	latin1_swedish_ci		No	*None*	

4.2 INPUT DESIGN

The user interface design is very important for any application. The interface design describes how the software communicates within itself, to system that interpreted with it and with humans who use it. The input design is the process of converting the user-oriented inputs into the computer based format. The data is fed into the system using simple inactive forms. The forms have been supplied with messages so that the user can enter data without facing any difficulty. They data is validated wherever it requires in the project. This ensures that only the correct data have been incorporated into system. The goal of designing input data is to make the automation as easy and free from errors as possible. For providing a good input design for the application easy data input and selection features are adopted. The input design requirements such as user friendliness, consistent format and interactive dialogue for giving the right messages and help for the user at right are also considered for development for this project.

Input Design is a part of the overall design. The input methods can be broadly classified into batch and online. Internal controls must be established for monitoring the number of inputs and for ensuring that the data are valid. The basic steps involved in input design are:

- Review input requirements.
- Decide how the input data flow will be implemented.
- Decide the source document.
- Prototype on line input screens.
- Design the input screens.

The quality of the system input determines the quality of the system output. Input specifications describe the manner in which data enter the system for processing. Input design features can ensure the reliability of the system and produce results from accurate data. The input design also determines whether the user can interact efficiently with the system.

4.3 OUTPUT DESIGN

A quality output is one, which meets the requirements of end user and presents the information clearly. In any system result of processing are communicated to the user and to the other system through outputs. In the output design it is determined how the information is to be displayed for immediate need.

It is the most important and direct source information is to the user. Efficient and intelligent output design improves the system's relationships with the user and helps in decision -making. The objective of the output design is to convey the information of all the past activities, current status and to emphasis important events. The output generally refers to the results and information that is generated from the system. Outputs from computers are required primarily to communicate the results of processing to the users.

Output also provides a means of storage by copying the results for later reference in consultation. There is a chance that some of the end users will not actually operate the input data or information through workstations, but will see the output from the system. Two phases of the output design are:

1. Output Definition
2. Output Specification

Output Definition takes into account the type of output contents, its frequency and its volume, the appropriate output media is determined for output. Once the media is chosen, the detail specification of output documents are carried out. The nature of output required from the proposed system is determined during logical design stage. It takes the outline of the output from the logical design and produces output as specified during the logical design phase.

In a project, when designing the output, the system analyst must accomplish the following:

- Determine the information to present.
- Decide whether to display, print, speak the information and select the output medium.
- Arrange the information in acceptable format.
- Decide how to distribute the output to the intended receipt.

Thus by following the above specifications, a high quality output can be generated.

SYSTEM IMPLMENTATION

5. SYSTEM IMPLEMENTATION

5.1 TESTING, TRAINING&IMPLEMENTATION

System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operation commences. Testing is a vital to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct, the goal will be successfully achieved. The candidate system is subject to a variety of tests.

Testing is a process of technical investigation, performed on behalf of stakeholders, that is intended to reveal quality related information about the product with respect to the context in which it is intended to operate. This includes, but is not limited to, the process of executing a program or application with the intent of finding errors. Quality is not an absolute; it is value to some person. With that in mind, testing can never completely establish the correctness of arbitrary computer software; testing furnishes a criticism or comparison that compares the state and behaviour of the product against a specification. An important point is that software testing should be distinguished from the separate discipline of software quality assurance (SQA), which encompasses all business process areas, not just testing.

Today, software has grown in complexity and size. The software product developed by a developer is according to the System Requirement Specification. Every software product has a target audience. Therefore, when an organization invests large sums in making a software product, it must ensure that the software product must be acceptable to the end users or its target audience. This is where Software Testing comes into play. Software testing is not merely finding defects or bugs in the software; it is the completely dedicated discipline of evaluating the quality of the software.

Testing is a process of executing a program with the intent of finding an error. Software testing is a critical element of software quality assurance and represents the ultimate review or specification, design and coding. Testing includes verification of the basic logic of each program and verification that the entire system works properly. Testing demonstrates that software functions appear to be working according to specification. In addition, data collected as testing is conducted provided a good indication of software quality as a while. The debugging process is the most unpredictable part of testing process.

Testing begins at the module level and works towards the integration of the entire computer based system testing and debugging are different activities, but any testing includes debugging strategy for software testing must accommodate low level tests that are necessary to verify that a small source code segment has been currently implemented as well as high-level tests that validate major system function, against customer requirements. No testing is complete without verification and validation part.

The goals of verification and validation activities are to access and improve the quality of work products generated during the development and modification of the software. There are two types of verification: **life cycle verification** and **formal verification**. Life cycle verification is the process of determining the degree to which the products of the given phase of the development cycle fulfil the specification established during the prior process. Formal verification is the rigorous mathematical demonstration that source code confirms to its specification. Validation is a process of evaluating software at the end of its specification. Validation is a process of evaluating software at the end of the software development process to determine compilation with the requirements.

5.2 TESTING TECHNIQUES

The various testing techniques are given below.

5.2.1 WHITE-BOX TESTING

White-box testing, also called as glass-box testing, is a test case design method that goes to the control structure of the procedural design to derive test cases. Using white-box testing methods, the software engineer can derive test cases that

- Guarantee that all independent paths within a module have been exercised at least once.
- Exercise all logical decision on their true and false sides.
- Execute all loops at their boundaries and within their operational bounds.
- Exercise internal data structure to ensure their validity.

White-box testing was successfully conducted on our system. All independent paths within a module have been executed at least once and all logical decisions have been exercised on their true and false sides.

5.2.2 BLACK-BOX TESTING

Black-box testing also called behavioural testing, focuses on the functional requirements of the software. It is a complementary approach that is likely to uncover a different class of errors than white-box methods. Black-box testing attempts to find errors in the following categories.

- Incorrect or missing functions
- Interface errors
- Errors on data structures or external database access
- Behavior or performance errors
- Initialization and termination errors

Black-box testing was successfully conducted on our system. The system was divided into a number of modules and testing was conducted on each module. We have tested the system for incorrect or missing functions and interface errors. Performance errors and the flow of information between modules ensuring interface.

5.3 LEVELS OF TESTING

Systems are not designed as entire systems nor are they as single systems. The analyst must perform both unit and system testing.

5.3.1 UNIT TESTING

Unit testing comprises the set of tests performed by an individual programmer prior to the integration of the system. Testing removes residual bugs and improves the reliability of the system. Testing allows the developer to find out the design faults if any, and enable correction if needed. Exhaustive unit testing has to be carried out to ensure the validity of the data. In order to successfully test the entire package, unit testing is carried out. Each module was tested as and when it was developed. Thus, it proved easier to conduct minute testing operation and correct them then and there.

5.3.2 INTEGRATION TESTING

Bottom-up integration is the traditional strategy used to integrate the component of a software system into a functional whole. Bottom-up integration consists of unit testing,

followed by subsystem testing and followed by testing of the entire system. Unit testing has the goal of discovering errors in the individual parts of the system. Parts are tested in isolation from one another in an artificial environment known as "Test Harness", which consists of driver programs and data necessary to exercise the modules.

Unit testing should be as exhaustive as possible to entire that each representative case handled by each module has been tested. Unit testing is eased by a system structure that is composed of small loosely coupled modules.

A subsystem consists of several modules that communicate with each other through well-defined interfaces. Normally, a subsystem implements a major segment of the total system. The primary purpose of the subsystem testing is to verify operation of the interfaces between modules in the subsystem. Both control and data interfaces must be tested. Large software system may require several levels of subsystem testing. Lower level subsystems are successively combined to form higher level subsystems. In most software systems, exhaustive testing of subsystem capabilities is not feasible due to the combination complexity of the module interfaces. Therefore, test cases must be carefully chosen to exercise the interfaces in the desired manner.

5.3.3 ACCEPTANCE TESTING

Acceptance testing involves planning and execution of functional tests, performance and stress tests in order to demonstrate that the implemented system satisfies its requirements. It is not unusual for two sets of acceptance tests to be run, those developed by the quality group and those developed by the customer.

In addition to the functional and performance tests, stress tests are performed to determine the limitation of the system. For example, a compiler might be tested to determine the effect of the symbol table overflow, or real-time system might be tested to determine the effect of simultaneous arrival of numerous high priorities interrupts.

5.3.4 VALIDATION TESTING

Validation testing is done to ensure complete assembly of the error-free software. Validation can be termed successful only if it functions in manner, reasonably expected by the customer under validation is alpha and beta testing. Alpha testing is where the end user tests the system rather than the developer, but in a controlled environment. Beta testing comes after alpha testing. Versions of the software, known as beta versions, are released to a limited audience

outside of the programming team. The software is released to groups of people so that further testing can ensure the product has few faults or bugs.

Here we have checked whether the data passed to each service is valid or not. For that we have entered incorrect values and did the validation testing and checked whether the errors are being considered and incorrect values are discarded. The errors were rectified. In this System, verifications are done correctly. So there is no chance for users to enter incorrect values. It will give error messages by using different validations. The validation testing is done very clearly and found it is error free.

5.3.5 OUTPUT TESTING

After performing the validation testing the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in a specific format. Asking the users about the format required by them, testing the output generated are considered into two ways. One is onscreen and another is printed format.

5.4 SOFTWARE TRAINING

Software Training and Support is important and a lot of developers fail to realize that.it would not matter how much time and planning a development team puts into creating software if nobody in an organization ends up using it. People are often resistant to change and avoid venturing into an unfamiliar area, so as a part of the deployment phase, it is very important to have training classes for new clients of your software. Another training element is training demonstration. The third element is the resident export. For example, one clerk read manual carefully, spent time on his practice and ended up being resident export-a natural teacher.

5.4.1 DOCUMENTATION

Software documentation or source code documentation is written text that accompanies computer software. It both explains how it operates or how to use it, and may mean different things to people in different roles. Documentation is an important part of software engineering. Types of documentation include:

- Requirements-Statements that identify attributes capabilities, characteristics, or qualities of a system. This is the foundation for what shall be or has been implemented.

- Architecture/Design-Overview of software. Includes relations to an environment and construction principles to be used in design of software components.
- Technical-Documentation of code, algorithms, interfaces, and API's.

5.5 IMPLEMENTATION

As the system is tested it starts to move into the implementation phase. Ideally the system should be completed and fully tested implementation gets under way but unless a package is being installed this seldom happens. Normally what happens is that parts of the system which are required for file set-up are completed first and this process gets under way. Conversion programs may also have to be available which allow data from another system to be used in setting up the files. Once this data is set up it must kept up-to-date and thus the first use is made of the new system. This may be followed by a period of parallel running and then a decision is made to drop the old system.

Implementation involved placing the completed and tested system of hardware and software into the actual work environment of the users. When systems personnel check out and put new equipment into use, train user personnel, install the new application, and construct any files of data needs to use it, we say it is implemented. There are both technical-and people-oriented activities during this stage. Examples of technical activities include converting data files, replacing old programs with new ones, and scheduling computer operations. Examples of people-oriented activities include orientation, training and support.

Implementation includes all those activities that take place to convert from the old system to the new one. The new system may be totally new, replacing an external system manual or automated system or it may be a modification to an external system. The process of putting the developed system in actual use is called system implementation. This includes all those activities that take place to convert to old system to new one. The most crucial stage is achieving a new successful system and giving confidence in new system that it will work efficiently and effectively.

5.5.1 Training

Training is a vital part of the computer based information system. The main aim of training is to make aware of the system concepts to users in detail.

The training focuses on two issues:

- User Capabilities.
- Name of the System being installed.

The training sections that are given to the operators and the end users are different. The operators need to know the concept of system in deep. However, the end user training is only concerned with how to perform the transaction and how to access the transaction details.

There are mainly three types of training:

- Brain storming section.
- Seminar.
- Computer awareness training.

5.6 SYSTEM MAINTENANCE

The maintenance phase of the software cycle is the time in which a software product performs the useful work. After a system is successfully implemented, it should be maintained in a proper manner; according to the technological advancements. It ensures the data integrity, data control and security. The system must be protected from fire and other natural calamities. The backup copies of data must be maintained daily so as to prevent the loss of data due to various reasons. Software maintenance activities can be classified into:

CONCLUSION AND FUTURE SCOPE

6. CONCLUSION AND FUTURE SCOPE

6.1 CONCLUSION

To conclude the description about the project :The project developed using PHP and MySQL is based on the requirement specification of the user and the analysis of the existing system ,with flexibility for future enhancement .The expanded functionality of today's software requires an appropriate approach towards software development. A software application of internal mark assessment system where in it deals with the tests, assignment, seminar, and attendance. The objective and scope of the proposed system is to record the details of the various activities of the user. Proposed system avoid the drawback of the existing system. Existing system is partially computerised. Proposed system can be updated efficiently and accurately .This system can calculated internal mark easily. The forms are designed in a user-friendly manner by providing messages and captions whenever necessary, so that the user has no problem to overcome difficulties in data entry, validation, searching etc.

Once again, we would like to thank everyone who has somehow or other related with the successful completion of the project.

.

ADVANTAGES

- It is fast, efficient and reliable
- Reduce work load
- Very user friendly
- Provide more security and integrity of data
- High Speed and accuracy
- Easy to use(edit,delete,display)
- Easy accessibility of data

6.2 FUTURE ENHANCEMENT

The proposed system is Blood bank management system. We can enhance this system by including more facilities for entering and viewing the user details, sharing of blood details. Providing such features enable the users to include more comments into the system.