A Project Report on

**BLOOD DONATION WEBSITE**

*Submitted in partial fulfillment for the*

*Requirement of the award of the Degree of*

*Bachelor of Technology*

***In***

***Computer Science and Engineering***

*By*

VENKATESH KADALI 12241A0559

RAMKUMAR BHUKYA 12241A0567

SWARUP 13245A0519

SAIRAM 11211A0579

Under the Esteemed guidance of

**Mrs.G.N.BeenaBethel**

*Associate Professor*



**Department Of Computer Science and Engineering**

**Gokaraju Rangaraju Institute of Engineering and**

**Technology**

**(**Affiliated to Jawaharlal Nehru Technological University)

Hyderabad-500090

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING

AND TECHNOLOGY

(Approved by AICTE, Affiliated to JNTU, Kukatpally)

Nizampet X Roads, Bachupally, Hyderabad

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

****

CERTIFICATE

This is to certify that the Industry Oriented Mini project entitled

**“Blood Donation Website”**  Is submitted by

VENKATESH KADALI 12241A0559

RAMKUMAR BHUKYA 12241A0567

SWARUP 13245A0519

SAIRAM 11211A0579

In partial fulfillment of the requirement for the award of the degree Bachelor of Technology in Computer Science and Engineering from Gokaraju Rangaraju Institute of Engineering and Technology, Kukatpally, Hyderabad for the academic year 2014-2015.

Mrs.G.N.BeenaBethel Dr. K. Anuradha Internal Guide Head of the Department

External Examiner

**ACKNOWLEDGMENT**

We are very indebted to our principal **Dr.Jandhyala N.Murthy** and our

Director **Prof P.S. Raju** for inspiring us to learn grow and excel.

We wish to express our sincere thanks to **Dr. K. Anuradha** our Head of Dept,.COMPUTER SCIENCE AND ENGINEERING , for her valuable guidance and support in the course of the project.

We specially thank our internal guide **Mrs.G.N.BeenaBethel** Asst. professor without her moral support guidance and encouragement this task has been unachievable her words provide motivation and courage for us.

We also thank the panel review committee members Dr. P. Vijay pal Reddy and Prof. P.V.S. Sinivas for their time to time suggestions to

shape up the project work.

Finally , we would also like to express our gratitude to all the staff members and lab faculty , department of Computer Science and Engineering ,

**GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND**

**TECNOLOGY** for the constant help and support.

VENKATESH KADALI

RAMKUMAR BHUKYA

SWARUP

SAIRAM

**CONTENTS**

Title Page. No

CHAPTER 1

**1. Introduction**

1.1. Introduction 7

1.2. Description

1.3. Background to Study

1.4. General Objectives

1.4.1. Specific Objectives

1.5. Scope

1.6. Significance of the Study

1.7. Blood Donor System

1.8 Applications 8

1.9. Existing System: 8

1.10. Proposed System 8

CHAPTER 2

**2. Review of State of Art**

2.3. Software Requirements 10

2.4. Hardware Requirements 10

CHAPTER 3

**3. System Design**

3.1. UML Design 13

3.2. Aim of Modeling 14

3.3.1. Use Case Diagram 15

3.3.2. Class Diagram 17

3.3.3. Activity Diagram 18

3.3.4. State Chart Diagram 19

3.3.5. Sequence Diagram 19

CHAPTER 4

**4. Feasibility study**

4.1. Technical feasibility

4.2. Economical feasibility

4.3. Operational Feasibility

4.4. Schedule feasibility

CHAPTER 5

**5. Implementation**

5.1. Features of html 20

5.2. Feature of Php 21

5.3. Java Script 21

5.4. My SQL 22

5.5 CSS 22 CHAPTER 6

**6. Coding**

6.1. Donor Registration Form 25

6.2. Hospital Registration Form 26

6.3. Hospital Login Form 28

6.4. Donor Login Form 29

6.5. Donor Inbox page 30

CHAPTER 7

**7. Testing**

7.1. Introduction

7.2. Strategic Approach to Software Testing

7.3. Unit Testing

CHAPTER 8

**8. Screen Shots**

8.1. Home Page

8.2. Hospital Registration Form

8.3. Blood Donor Registration Form

8.4. Blood Donor Login Page

8.5. Blood Recipient Login Page

8.6. Blood Donor Inbox

8.7. Blood Donor Mailbox

8.8. List of Blood Donors

8.9. Hospital Registration Table

8.10. Blood Donor Registration Table

8.11. Patient Entry Table

8.12. HTML Code

8.13. PHP Code

8.14. CSS Code

**9. Conclusion**

**10. Bibliography**

1. **INTRODUCTION**

**1.1. Introduction**

Quality of Web-applications plays major role in its success. The use of strong Web-application architecture with strong development platform Not only make Web-applications robust and of high quality but also give Web-Application an ability to meet changing and demanding customer requirements in Efficient manner. The Blood Donation Agent is to create an e-Information about the donor and organization that are related to donating the blood. Through this application any person who is interested in donating the blood can register himself in the same way if any organization wants to register itself with this site that can also register. Moreover if any general consumer wants to make request blood online he can also take the help of this site. This consists of two set of users, Administrators and Blood Donors. Administrator login is created for chief doctors in major hospitals which run this web application. Blood Donors who are willing to donate blood have to register in the system

**1.2. Description**

Blood Donation Website is a web application developed for blood donations. This consists of two set of users, Administrators and Blood Donors. Administrator login is created for chief doctors in major hospitals which run this web application. Blood Donors who are willing to donate blood have to register in the system. When blood is needed for a person, then the corresponding administrator should add the details of the patient such as, Patient name, Blood group, Location of the hospital, etc. The system will automatically analyze the list of donors who can donate blood for a particular patient. Then the system compares the location of the person and the hospital where the patient is admitted. Then the system compares the day of blood needed and the day’s availability of the donor. Finally for the qualified donors E-Mail will be sent to their personal Mail Address and a mail will be dispatched to the donor login page of the system. If the donor is interested in donating blood then he can login to the system to see the full details.

**1.3 Background to the Study**

Blood Donor Recruitment (BDR) is the process of drawing blood from a voluntary Blood Donor (BD) for future blood transfusion, Wikipedia (2006). In Uganda, blood collection, safety and management is an activity that is carried out by Uganda Red Cross Society (URCS) in partnership with Uganda Blood Transfusion (UBTS). Founded in 1939, URCS is part of the world wide Red Cross Humanitarian Movement whose mission is to mobilize the power of humanity for improving the lives of the vulnerable in Uganda, Muller (2001).

URCS fulfills this mission while adhering to the principles of impartiality, neutrality, independence, unity, universality and voluntary service for the Red Cross/Red Crescent Movement. It operates throughout Uganda with 45 branch offices. Besides providing adequate supply of blood for transfusion, URCS is involved in the first aid services, road safety, tracing, disaster mitigation/preparedness, mobilization for routine immunization, HIV homecare, youth empowerment and Community based HealthCare (CBHC). URCS had a manual system using paper cards to recruit BDs, collect/keep blood donor records and disseminate results to BDs who are scattered throughout the country. The paper card system (PCS) used to specifically capture personal data and medical history of the BDs.

This information would be used in identifying/locating existing BDs, carrying out pre-donation counseling and taking blood results. Unauthorized persons however, easily accessed the paper system and hence making it impossible to keep secrecy and confidentiality expected of medical records. The security of the medical records was also not inadequate as any person could easily access them. Lukande (2003), states that such a system is time consuming, prone to errors of entry and analysis resulting from the fatigue of the users. The PCS at URCS had lead to accumulation of physical paper cards due to increasing number of blood donors, a situation that frustrated the system users because of the delays and at times failure to access historical records.

The safe blood policy was lacking at URCS because the PCS could not cater for the keyattributes of the policy. Gerard (2002), states that the main principles upon which the safe blood policy is based on are the informed consent, confidentiality and secrecy of the BDs. The Ethiopian Red Cross Society publication, Development in the 1990 states that information from blood donors should be completely confidential and if this is not assured, names of the blood donors should not be recorded at all and/or an alternative record identification should be used.

Full implementation of the safe blood policy has called the use of information technology (IT) in providing working solution to the identified challenges. The associated problems with the PCS included delays in accessing historical records, inconsistencies and errors in data entry that stem right from acquisition of data from the blood donors because the exercise is of routine nature and very tedious to the system users. The automation of the system using modern IT has improved the quality of service. Secondly, with the use of IT, now relevant and timely blood donor reports can easily be generated and hence facilitating planning and decision-making. Scolamiero (2000), recommends blood donor services automated information system as a solution to routinely collected, accurate and readily available information in blood transfusion services. It is also important to note that the impact of information technology on organizations is increasing as new technologies evolve and existing ones expand.

According to Clifton (1995), nearly all business executives say that information technology is vital to their business and that they use IT extensively. Certainly business executives main concern is planning, coordination and decision-making, therefore, the role of IT in enhancing management of blood donor records is of major importance. In all, the computerization of blood donor PCS at URCS came at the ripe time given the background to the situation. This is more so because the demand for safe blood in Uganda has increased due to soaring increase in total population. Therefore, modern means to manage the PCS using IT had to take route.

**1.4 General Objective**

The main objective of the study was to create electronic blood donor management information system in order to assist in the management of blood donor records, planning and share information in a more confidential, convenient and secure way using modern technology.

**1.4.1 Specific Objectives**

To conduct a study on blood donor management

To design an electronic blood donor management system

To validate the design using a prototype

**1.5 Scope**

The study geographically limited itself at the URCS blood donation/collection centers. It focused more on the acquisition, distribution and management of blood units for BDR activities. The study specially emphasized the creation and implementation of an electronic management information system that automated blood donor data acquisition and dissemination of results. This in turn will ease and speeds up the planning, decision-making process because of the timely, secure, confidential and reliable reports.

**1.6 Significance of the Study**

This study is important to URCS and the blood donors because it aimed at addressing problems of security, secrecy and confidentiality of blood donor records. It also strived to check the delays, errors, inconsistencies in medical records and timely access to historical records all of which had far fetched impact on planning and decision-making.

The study resulted into the following benefits:

* It has eased the control and distribution of blood in various parts of the country basing on
* the hospital demands.
* URCS can now create market strategies for blood donation, lobbying and sensitization of the blood donors.
* Automated data acquisition and quick access to medical records by the legal users of the
* system will be assured.
* It has eased the monitoring of the results and performance of the blood donation activity
* and hence relevant and measurable objectives of URCS are checked.
* It will continue to improve on the planning and decision-making process by providing to
* management timely, secure and confidential medical reports related to blood donation.
* It will also improve medical service delivery due to timely and easy generation of
* management reports by the relevant entities.

The study will benefit the URCS management, who will find it easy to strategically plan, coordinate and take decisions concerning BDR activities. URCS counsellors on the other hand will be able to keep confidentiality of the donor’s results and disseminate blood results to donors with ease. Meanwhile that is the case, the automation of the data collection process will simplify the work of the data clerks. Equally important, the blood donor mobilizes will be have strong grounds for laying sensitization strategies between regions that yield more blood units and those with less. The study also has formed further environment of knowledge for students who may wish to take research in blood donor management.

**1.7 Blood Donor Systems: Challenges and Successes**

The blood donation service involve a series of interdependent operations such as donor registration, donor screening/evaluation, blood collection, blood screening, inventory management and blood dissemination. Most of the popular existing blood information systems in the western world today are mainly online systems. The systems interfaces do not meet fully the blood safe policy described in this study and as such not suitable for illiterate population. Most blood donors in Uganda are rural based where online systems may not be the best. The level of computer literate among the blood donors in Uganda is growing because the majority of them are school students. The main challenge remains customizing interfaces that are suitable for capturing basic donor information. Some of the attributes on the interfaces used in the western world such as state and province are not applicable in Uganda. Tripura blood donor information system is a good example of the blood donorsystem that is not suitable for Uganda. Also some key attributes such as age and sessions in Uganda are lacking on most the interfaces viewed. The interfaces also are not user-friendly as there are many links within the system that can easily confuse the system users and hence leading to data entry errors and boredom.

At the Macau blood Transfusion Centre, system Integrado de Bancos de Sangue (SIBAS) works as its solution of computerized blood bank information system. SIBAS complies with the client/server infrastructure, as does its client, and provides an integrated environment for those isolated but interdependent operation in the blood center. With the introduction of the SIBAS the blood service at Macau has been enhance in the following aspect. Operational efficiency- the processing time has been shortened in that blood donors need not fill in many regular items. On the other hand, the steps for donor cards are under full control and hence leading to donor satisfaction and confidence. There is also improved information consistency and validity.

The Indian case study of Prathma Blood Center, Gupta (2004), promises insights into the integration of IS/IT in management of blood records. The Prathma Blood Center is a quest for modernizing blood banking. The entire function from blood donation to its testing and separation, storage, issue and usage have been integrated through a custom designed enterprise resource planning (ERP) software that minimizes human intervention and making it less error prone. The implementation of ERP in blood bank in India has registered many successes in medical data such as security, confidentiality, secrecy and quick retrieval of historical records all of which were challenges at URCS blood center. However, full automation of all blood donation activities like the case cannot be done in Uganda due to limited resources. It requires transition, as it is resource constraining in terms of IT, other equipments and human resources.

**1.8. Applications:**

* User friendliness I provided in the application with various controls.
* The system makes the overall project management much easier and flexible.
* It provides high level of security with different level of authentication.

**1.9. Existing System**

Recent figures provided by the World Health Organisation (WHO) states that India faces a shortage of 3million units of blood per year. Due to the increased demand of the blood that hospitals are dependent on a constant supply of blood. There are some websites related to Blood Donation. But in those websites the recipient should call all the donors who are listed in that website.

**1.10. Proposed System**

In Life saver system ,the recipient need not call to all the blood donors. The system will automatically send the messages to all the donors. So the donors who are willing to donate the blood can go to the corresponding hospital. Once a blood donor reaches to the hospital to donatethe blood. Then the system will automatically send E-Mail to all the remaining donors that their need is fulfilled.

**2. Review of State of Art**

In the previous chapter we have discussed about the literature survey of the project.

It included the details about the existing system and the proposed system. This

Chapter deals with the Hardware and software Requirements of the project.The description of the software requirements are also provided in this chapter. The domain required for the project is explained along with its details.

**2.1 Requirements and Analysis**

The requirement analysis stage of a software engineering project involves collecting and analyzing information about the part of the organization that is supported by the application. This information is then used to identify the users' requirement of the new system . Identifying the required functionality of the system is very important as a system with incomplete functionality may lead to it being rejected. A description of the aim of the project is given here along with details of the functional and non-functional requirements for the system. The test sheets for evaluating the completed system are also presented.

**2.2 Requirements**

The requirements of the Web-based management information system are to develop:

* a web based front end for entering donated blood details including the donor, his/her
* blood group, sex, age, and status of the donated blood
* a web based front end for searching the information relating to a given donor or a given
* blood group;
* a facility to still enter donor and donated blood information via Endnote and also
* maintain the Endnote database using those details entered via the web front end and
* a facility to produce summary information of donor and donated blood particulars and

any other related activities.

**2.3 Functional Requirements**

In this research project we aim at developing a system which should improve on the current

one with a lot of functionalities and therefore the Major target or goal here is to:

• to develop a blood donor database that can support the five above mention sub-

databases that is to say; DonorDB, Donation DB, DiseaseDB, Transfusion DB and

Statistical DB

• to develop a client interface that allows privileged users to carry out tasks such as

inserting or modifying and deleting data in the database;

• to develop a searching functionality in order to allow normal and privileged users to

search the details of a given donor, blood group, stakeholder and if necessary a type

of disease common which causes one to need the donated blood

• to fully integrate the Web-based management information system to the World-

Wide-Web and hence allow access from any Internet networked terminal and Web

browser around the world;

• to develop a facility that can export details entered via the web front end to Endnote

as well as import and confidential detail from the Endnote Database;

• to develop a functionality that produces summary information of required data to

enhance decision making;

• to embed high security features in the Web DBMS to provide privacy, integrity;

• to allow privileged users to maintain the Web-based management information system

by adding/deleting particulars, backing-up or resetting the database and extract online summary in the form of histograms for each donor and lists of free-format comments.

• Thus a graphical reporting tool should be provided for analyzing the data and finally the system should be flexible enough to store data for several years and also be able provide sufficient User and Administration Guides.

**2.4 Non-functional Requirements**

The system must be developed to suit the particular needs of a user-friendly environment. This means that the system must accommodate a clearly understandable user interface as well as clear online help documentation at any stage of the user interaction with the system. A fast response time in obtaining and providing information to the system may also prove to be a significant advantage. In addition to these requirements, the system should also embrace the following requirements:-

**Security:** Each user is required to log in. The system should log staff that has been assigned user names and passwords. The system should be designed to make it impossible for anybody to logon without a valid username and password. Data encryption should be employed to keep the user login name and password secret.

**Reliability:** The system would be used by about 50 staff working at the Red Cross head

quarters and also some other many staff in the collaborating clinics and hospitals. The

system should have little or no downtime and be able to handle multiple concurrent users.

**Ease of Use:** The general and administrative views should be easy to use and intuitive.

Online help and documentation should be provided.

**Performance:** The system should have a quick response time. For the purpose of this

research project, this would be defined as less than 5 seconds.

**System and Browser compatibility Testing:** The system should be accessible on the

following browsers - Microsoft Internet Explorer 5.5+, NetScape Navigator 6.0+ and

Mozilla 1. 3+.

2.5. **Hardware requirements**

* Processor : Pentium 1V.
* Hard Disk : 40 GB.
* Speed : 2.0GHz.
* RAM : 256 Mb.

**2.6. Software requirements**

FRONTEND

* Language : CSS, javascript, HTML, PHP.
* OS : Windows 2003, XP, 2007.
* Server : Xampp server.

BACKEND

* Database : Mysql.

**3. SYSTEM DESIGN**

This chapter explains the design of our system using UML diagrams.

**3.1 UML Design**

The unified modeling language is a standard language for specifying, Visualizing, Constructing and documenting the software system and its components. It is a graphical language which provides a vocabulary and set of semantics and rules. The UML focuses on the conceptual and physical representation of the system. It captures the decisions and n understandings about systems that must be Constructed. It is used to understand, design, configure, maintain and control Information about the systems.

**3.2 Aims of Modeling**

The following are aims of modeling:

* Models help us to visualize a system as it is or as we want to be
* Models permit us to specify the structure of system.
* Model gives us template that guides us in constructing system.
* Models can document the decisions we have made.

The vocabulary of the UML encompasses 3 kinds of building blocks.

They are

• Things

• Relationships

• Diagrams

There are 4 kinds of things in the UML.

• Structural things

• Behavioral things

• Grouping things

• An notational things

There are 4 kinds of relationships

• Dependency

• Association

• Generalization

• Realization

**Dependency:** A dependency is a semantic relationship between 2 things in which an Change to one thing affects the semantics of the other things.

**Symbol:**

**------------------------------>**

Fig 3.2.1: **Dependency symbol**

**Association:** An association is a structural relationship that describes a set of links a link being a connection among objects.

**Symbol:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Fig 3.2.2: **Association symbol**

**Aggregation:** An association is a special kind of Association, representing a structural

relationship between a whole and its parts.

**Composition:** Composition is a strong relationship between whole and its parts.

**Generalization:** A Generalization is a relationship in which objects of the Specialized elements are suitable substitutable for objects of the generalized element. Relationship exist between a general thing and a kind of relationship.

**Realization:** A realization is a semantic relation between classifiers, where in one classifier guarantees to carry out and represented as dotted line with a hollow Arrow head pointing to the parent. It is a combination of dependency & Generalization.Object oriented concepts were introduced much earlier than UML. So at that time there were no standard methodologies to organize and consolidate The object oriented development. At that point of time UML came into picture.There are a number of goals for developing UML but the most important is to Define some general purpose modeling language which all modelers can use and Also it needs to be made simple to understand and use. UML diagrams are not only made for developers but also for business users, common people and anybody Interested to understand the system. The system can be a software or non-software. So it must be clear that UML is not a development method rather it accompanies with processes to make a successful system. At the conclusion the goal of UML can be defined as a simple modeling mechanism to model all possible practical Systems in today’s complex environment. Provide users with a ready-to-use, Expressive visual modeling language so they can develop and exchange meaningful models. Provide extensibility and specialization mechanisms to extend the core concepts. Be independent of particular programming languages and development processes. Provide a formal basis for understanding the modeling language. Encourage the growth of the OO tools market. Support

**3.3. Diagrams in the UML:**

A diagram is the graphical presentation of a set of elements, most often rendered A connected graph of vectors and arcs.

There are 9 diagrams in the UML. These are

• Class Diagram

• Object Diagram

• Use-Case Diagram

• Sequence Diagram

• Collaboration Diagram

• State-Chart Diagram

• Activity Diagram

• Component Diagram

• Deployment Diagram

In our project we designed our system by using the following Diagrams.

**3.3.1. Use case diagrams**

It shows the set of use cases and actors and their relationships. It shows the static view of the system and used in organizing and modeling the behaviors of the system.

Actors involved in above use-case diagram

• User

• Tool

These internal and external agents are known as actors. So use case diagrams are consists of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. A single Use case diagram captures a particular functionality of a system. The purpose of use case diagram is to capture the dynamic aspect of a system. But this definition is too generic to describe the purpose. Because other four diagrams (activity, Sequence, collaboration and State chart) are also having the same purpose. So we will look into some specific purpose which will distinguish it from other four diagrams.

Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. So when a system is analyzed to gather its functionalities use cases are prepared and actors are identified. Use case diagrams specify the events of a system and their flows.Then this high level design is refined again and again to get a Complete and practical picture of the system. A well-structured use case also describes the pre-condition, post condition, exceptions. And these extra elements are used to make test cases when performing the testing.



Fig 3.3.1: **Use case diagram**

Use case diagram can be imagined as a black box where only the input, Output and the function of the black box is known. These diagrams are used at a very high level of design. Then this high level design is refined again and again to get a complete and practical picture of the system. A well-structured use case also describes the pre-condition, post condition, Exceptions.

**3.3.2. Class Diagram**

The class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing and documenting different aspects of a system but also for constructing executable code of the software application. The class diagram describes the attributes and operations of a class and also the constraints imposed on the system.There are a number of goals for developing UML but the most important is to define some general purpose modeling language which all modelers can use And also it needs to be made simple to understand and use.The class diagrams are widely used in the modelling of object oriented Systems because they are the only UML diagrams which can be mapped directly

With Object oriented languages. The class diagram shows a collection of classes, interfaces, associations, collaborations and constraints. It is also known as a Structural diagram. The purpose of the class diagram is to model the static view of an application.

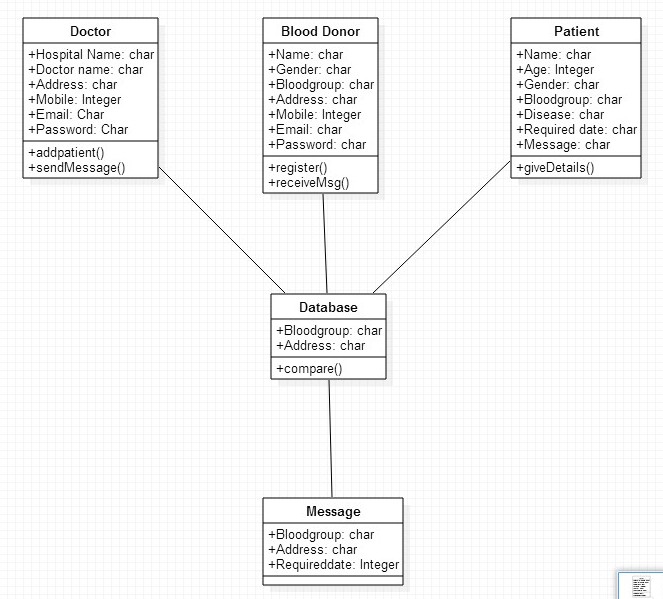


Fig 3.3.2: **Class diagram**

The class diagrams are the only diagrams which can be directly mapped with object oriented languages and thus widely used at the time of construction. Class diagrams are the most popular UML diagrams used for construction of Software applications. So it is very important to learn the drawing procedure of class diagram. Class diagrams have lot of properties to consider while drawing but here the diagram will be considered from a top level view. Class diagram is basically a graphical representation of the static view of the system and represents different aspects of the application. So a collection of class diagrams represent the whole system. Class diagram is also considered as the foundation for component and deployment diagrams. Class diagrams are not only used to visualize the static view of the system but they are also used to construct the executable code for forward and reverse engineering of any system.

**3.3.3. Activity Diagram**

Activity diagrams are used to visualize the flow of controls in a system. This is prepared to have an idea of how the system will work when executed. Activity diagram is basically a flow chart to represent the flow form one activity to another activity. The activity can be described as an operation of the system. So the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent. Activity diagrams deals with all type of flow control by using different elements like fork, join etc. Activity diagrams are not only used for visualizing dynamic nature of a system but they are also used to construct the executable system by using forward and reverse engineering techniques

Diagrams are not only used for visualizing dynamic nature of a system but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in activity diagram is the message part. Activity diagrams are mainly used as a flow chart consists of activities performed by the system. But activity diagram are not exactly a flow chart as they have some additional capabilities. These additional capabilities include branching, parallel flow, swim lane etc. Before drawing an activity diagram we must have a

Clear understanding about the elements used in activity diagram. The main element of an activity diagram is the activity itself. An activity is a function performed by the system.

After identifying the activities we need to understand how they are associated with constraints and conditions. The activity diagram is suitable for modeling the activity flow of the system. An application can have multiple systems. Activity diagram also captures these systems and describes flow from one system to another. This specific usage is not available in other diagrams. These systems can be database, external queues or any other system.

**3.3.4. State Chart Diagram**

Any real time system is expected to be reacted by some kind of internal/external events. These events are responsible for state change of the system. State chart diagram is used to visualize the reaction of a system by internal/external factors. State chart diagram is one of the five UML diagrams used to model dynamic nature of a system.

A State chart diagram describes a state machine. Now to clarify it state machine can be defined as a machine which defines different states of an object and these States are controlled by external or internal events. State chart diagram is one of the five UML diagrams used to model dynamic Nature of a system. They define Different states of an object during its lifetime and these states are changed by Events.

So State chart diagrams are useful to model reactive systems. Reactive Systems can be defined as a system that responds to external or internal events. State chart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. So the most important purpose of State chart diagram is to model life time of an object from creation to termination. State chart diagrams are also used for forward and reverse engineering of a system objects in its life cycle.

**3.3.5. Sequence Diagram**

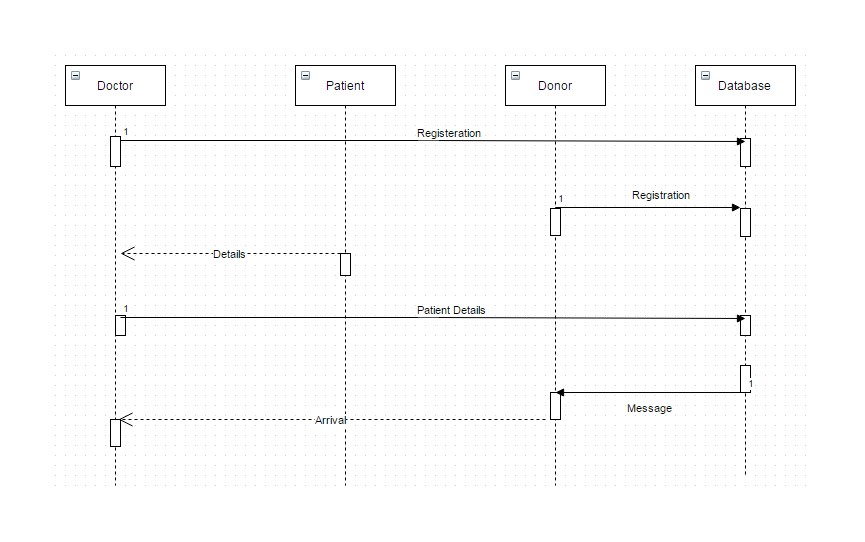
****

Fig 3.3.5 : **Sequence diagram for user**

UML sequence diagrams are used to show how objects interact in a given situation. An important characteristic of a sequence diagram is that time passes from top to bottom: the interaction starts near the top of the diagram and ends at the bottom (i.e. **L**ower equals**later**). A popular use for them is to document the dynamics in an object-oriented system. For each key collaboration, diagrams are created that show how objects interact in various representative scenarios for that collaboration. Sequence diagram emphasizes on time sequence of messages and collaboration diagram emphasizes on the structural organization of the objects that send and receive messages. The sequence diagram is having four objects (Customer, Order, Special-order and Normal Order).

**4. FEASIBILITY STUDY**

Depending on the results of the initial investigation the survey is now expanded to a

more detailed feasibility study. “***FEASIBILITY STUDY***” is a test of system proposal

according to its workability, impact of the organization, ability to meet needs and

effective use of the resources. It focuses on these major questions:

1. What are the user’s demonstrable needs and how does a candidate system

meet them?

2. What resources are available for given candidate system?

3. What are the likely impacts of the candidate system on the organization?

4. Whether it is worth to solve the problem?

During feasibility analysis for this project, following primary areas of interest are to be

considered. Investigation and generating ideas about a new system does this.

**Steps in feasibility analysis**

Eight steps involved in the feasibility analysis are:

\_ Form a project team and appoint a project leader.

\_ Prepare system flowcharts.

\_ Enumerate potential proposed system.

\_ Define and identify characteristics of proposed system.

\_ Determine and evaluate performance and cost effective of each proposed system.

\_ Weight system performance and cost data.

\_ Select the best-proposed system.

\_ Prepare and report final project directive to management

**4.1. Technical feasibility**

A study of resource availability that may affect the ability to achieve an acceptable

system. This evaluation determines whether the technology needed for the proposed

system is available or not.

• Can the work for the project be done with current equipment existing software

technology & available personal?

• Can the system be upgraded if developed?

• If new technology is needed then what can be developed?

This is concerned with specifying equipment and software that will successfully satisfy

the user requirement. The technical needs of the system may include:

**Front-end and back-end selection**

An important issue for the development of a project is the selection of suitable front-end

and back-end. When we decided to develop the project we went through an extensive

study to determine the most suitable platform that suits the needs of the organization as

well as helps in development of the project.

The aspects of our study included the following factors.

**Front-end selection:**

1. It must have a graphical user interface that assists employees that are not from IT

background.

2. Scalability and extensibility.

3. Flexibility.

4. Robustness.

5. According to the organization requirement and the culture.

6. Must provide excellent reporting features with good printing support.

7. Platform independent.

8. Easy to debug and maintain.

9. Event driven programming facility.

10. Front end must support some popular back end like Ms Access.

According to the above stated features we selected VB6.0 as the front-end for

developing our project.

**Back-end Selection:**

1. Multiple user support.

2. Efficient data handling.

3. Provide inherent features for security.

4. Efficient data retrieval and maintenance.

5. Stored procedures.

6. Popularity.

7. Operating System compatible.

8. Easy to install.

9. Various drivers must be available.

10. Easy to implant with the Front-end.

According to above stated features we selected Ms-Access as the backend.

The technical feasibility is frequently the most difficult area encountered at this stage. It

is essential that the process of analysis and definition be conducted in parallel with an

assessment to technical feasibility. It centers on the existing computer system

(hardware, software etc.) and to what extent it can support the proposed system.

**4.2. Economical feasibility**

Economic justification is generally the “Bottom Line” consideration for most systems.

Economic justification includes a broad range of concerns that includes cost benefit

analysis. In this we weight the cost and the benefits associated with the candidate

system and if it suits the basic purpose of the organization i.e. profit making, the project

is making to the analysis and design phase.

The financial and the economic questions during the preliminary investigation are

verified to estimate the following:

• The cost to conduct a full system investigation.

• The cost of hardware and software for the class of application being considered.

• The benefits in the form of reduced cost.

• The proposed system will give the minute information, as a result the

performance is improved which in turn may be expected to provide increased

profits.

• This feasibility checks whether the system can be developed with the available

funds. The **Hospital Management System** does not require enormous amount of

money to be developed. This can be done economically if planned judicially, so it

is economically feasible. The cost of project depends upon the number of manhours

required.

**4.3. Operational Feasibility**

It is mainly related to human organizations and political aspects. The points to be

considered are:

• What changes will be brought with the system?

• What organization structures are disturbed?

• What new skills will be required? Do the existing staff members have these

skills? If not, can they be trained in due course of time?

The system is operationally feasible as it very easy for the End users to operate it. It

only needs basic information about Windows platform.

**4.4. Schedule feasibility**

Time evaluation is the most important consideration in the development of project. The

time schedule required for the developed of this project is very important since more

development time effect machine time, cost and cause delay in the development of

other systems.

A reliable **Hospital Management System** can be developed in the considerable amount

Of time.

**5. IMPLEMENTATION**

**5.1. Features of HTML**

Hypertext Markup Language, commonly referred to as HTML, is the standard [markup language](http://en.wikipedia.org/wiki/Markup_language) used to create [web pages](http://en.wikipedia.org/wiki/Web_page). It is written in the form of [HTML elements](http://en.wikipedia.org/wiki/HTML_element) consisting of tags enclosed in [angle brackets](http://en.wikipedia.org/wiki/Bracket#Angle_brackets) (like <html>). HTML tags most commonly come in pairs like <h1> and </h1>, although some tags represent empty elements and so are unpaired, for example <imp>. The first tag in a pair is the start tag, and the second tag is the end tag (they are also called opening tags and closing tags). HTML elements form the building blocks of all websites.

HTML allows [images and objects](http://en.wikipedia.org/wiki/Img_(HTML_element)) to be embedded and can be used to create [interactive forms](http://en.wikipedia.org/wiki/Fieldset). It

provides a means to create [structured documents](http://en.wikipedia.org/wiki/Structured_document) by denoting structural semantics for text such as

headings, paragraphs, lists, [links](http://en.wikipedia.org/wiki/Hyperlink), quotes and other items. It can embed [scripts](http://en.wikipedia.org/wiki/Scripting_language) written in languages such as [JavaScript](http://en.wikipedia.org/wiki/JavaScript) which affect the behavior of HTML web pages.HTML is a set of codes that a website author inserts into a plain text file to format the content. The author inserts HTML tags, or commands, before and after words or phrases to indicate their format and location on the page. HTML tags are also used to add tables, lists, images, music, and other elements to a webpage. Web documents contain three main sections: the head, title, and body. The head includes the webpage’s identifying information, including the Website’s title and important keywords. The viewer sees the site’s title, but any other information is hidden. The site’s title appears in the browser’s tab and is what appears when a user tries to Bookmark a site.

The body section is the main portion of the webpage visible to the viewer, Including the text and graphics. HTML tags are also used in two additional ways that are not Visible to the viewer as Meta tags and comments. Meta tags indicate keywords associated with the webpage to search engines. Comments are intended as explanation or additional information for other writers or readers of HTML code. Hypertext is text that references other web pages or text and that, when clicked on, allows the user to access the referenced text or webpage. HTML is used to embed hyperlinks within web pages. Hyperlinks allow the user to move easily within web pages and between websites stored on different servers.

**5.2. Features of PHP**

PHP is a server scripting language, and a powerful tool for making dynamic and interactive Web pages. PHP is a widely-used, free, and efficient alternative to competitors such as Microsoft's ASP.PHP is a [server-side scripting](http://en.wikipedia.org/wiki/Server-side_scripting) language designed for [web development](http://en.wikipedia.org/wiki/Web_development) but also used as a [general-purpose programming language](http://en.wikipedia.org/wiki/General-purpose_programming_language). Basically, PHP allows a static webpage to become dynamic. "PHP" is an acronym that stands for "PHP: Hypertext Preprocessor". PHP code can be simply mixed with [HTML](http://en.wikipedia.org/wiki/HTML) code, or it can be used in combination with various [tinplating engines](http://en.wikipedia.org/wiki/Web_template_system) and [web frameworks](http://en.wikipedia.org/wiki/Web_framework). PHP code is usually processed by a PHP [interpreter](http://en.wikipedia.org/wiki/Interpreter_(computing)), which is usually implemented as a web server's native [module](http://en.wikipedia.org/wiki/Plugin_(computing)) or a [Common Gateway Interface](http://en.wikipedia.org/wiki/Common_Gateway_Interface) (CGI) executable. After the PHP code is interpreted and executed, the web server sends resulting output to its client, usually in form of a part of the generated web page; for example, PHP code can generate a web page's HTML code, an image, or some other data.

PHP has also evolved to include a [command-line interface](http://en.wikipedia.org/wiki/Command-line_interface) (CLI) capability and can be used in standalone [graphical applications](http://en.wikipedia.org/wiki/Graphical_user_interface). PHP web development means developing websites and dynamic web pages using the versatile and capable server-side scripting language, PHP. PHP code can be simply mixed with [HTML](http://en.wikipedia.org/wiki/HTML) code, or it can be used in combination with various [templating engines](http://en.wikipedia.org/wiki/Web_template_system" \o "Web template system) and [web frameworks](http://en.wikipedia.org/wiki/Web_framework). PHP code is usually processed by a PHP [interpreter](http://en.wikipedia.org/wiki/Interpreter_(computing)), which is usually implemented as a web server's native [module](http://en.wikipedia.org/wiki/Plugin_(computing)) or a [Common Gateway Interface](http://en.wikipedia.org/wiki/Common_Gateway_Interface) (CGI) executable. After the PHP code is interpreted and executed, the web server sends resulting output to its client, usually in form of a part of the generated web page; for example, PHP code can generate a web page's HTML code, an image, or some other data. PHP has also evolved to include a [command-line interface](http://en.wikipedia.org/wiki/Command-line_interface) (CLI) capability and can be used in[standalone](http://en.wikipedia.org/wiki/Computer_software) [graphical applications](http://en.wikipedia.org/wiki/Graphical_user_interface).

The canonical PHP interpreter, powered by the Zend Engine, is [free software](http://en.wikipedia.org/wiki/Free_software) released under the [PHP License](http://en.wikipedia.org/wiki/PHP_License). PHP has been widely ported and can be deployed on most web servers on almost every [operating system](http://en.wikipedia.org/wiki/Operating_system) and [platform](http://en.wikipedia.org/wiki/Computing_platform), free of charge.

Despite its popularity, no written [specification](http://en.wikipedia.org/wiki/Formal_specification) or standard existed for the PHP language until 2014, leaving the canonical PHP interpreter as a [*de facto*](http://en.wikipedia.org/wiki/De_facto) standard. Since 2014, there is ongoing work on creating a formal PHP specification.

**5.3. Features of java script**

Java Script  also known as ECMA Script  is a [dynamic programming language](http://en.wikipedia.org/wiki/Dynamic_programming_language). It is most commonly used as part of [web browsers](http://en.wikipedia.org/wiki/Web_browser), whose implementations allow [client-side scripts](http://en.wikipedia.org/wiki/Client-side_scripting) to [interact with the user](http://en.wikipedia.org/wiki/User_interface), control the browser, communicate [asynchronously](http://en.wikipedia.org/wiki/Ajax_(programming)), and alter the [document content](http://en.wikipedia.org/wiki/Document_Object_Model) that is displayed.[[5]](http://en.wikipedia.org/wiki/JavaScript#cite_note-FOOTNOTEFlanaganFerguson20061-5) It is also used in server-side network programming with runtime environments such as [Node.js](http://en.wikipedia.org/wiki/Node.js), game development and the creation of desktop and mobile applications. With the rise of the single-page web app and JavaScript-heavy sites, it is increasingly being used as a compile target for [source-to-source compilers](http://en.wikipedia.org/wiki/Source-to-source_compiler) from both [dynamic languages](http://en.wikipedia.org/wiki/Dynamic_typing) and [static languages](http://en.wikipedia.org/wiki/Static_typing). JavaScript is classified as a [prototype-based](http://en.wikipedia.org/wiki/Prototype-based_programming) [scripting language](http://en.wikipedia.org/wiki/Scripting_language) with [dynamic](http://en.wikipedia.org/wiki/Dynamic_language) typing and [first-class functions](http://en.wikipedia.org/wiki/First-class_functions). This mix of features makes it a [multi-paradigm](http://en.wikipedia.org/wiki/Multi-paradigm) language, supporting [object-oriented](http://en.wikipedia.org/wiki/Object-oriented_programming),[[7]](http://en.wikipedia.org/wiki/JavaScript#cite_note-ECMA-262-7) [imperative](http://en.wikipedia.org/wiki/Imperative_programming), and [functional](http://en.wikipedia.org/wiki/Functional_programming) programming styles. Java script is highly popular as client side scripting language for web browsers. In any web application, java script can be used to implement simple features like rollover of images as well as to make asynchronous requests to server using Ajax. Few years back when flash was not so popular, java script was widely used to add beautiful effects to webpages and is still being used for the same purpose. The advantages of java are **Java script is executed on the client side Java script is a relatively easy language Java script is relatively fast to the end user Extended functionality to web pages.**

**5.4. Features of MySQL**

MySQL is the world's second most widely used [relational database management system](http://en.wikipedia.org/wiki/Relational_database_management_system) (RDBMS) and most widely used open-source RDBMS. MySQL is a popular choice of database for use in web applications, and is a central component of the widely used [LAMP](http://en.wikipedia.org/wiki/LAMP_(software_bundle)) open source web application software stack (and other ['AMP'](http://en.wikipedia.org/wiki/List_of_AMP_packages) stacks). LAMP is an acronym for "[Linux](http://en.wikipedia.org/wiki/Linux), [Apache](http://en.wikipedia.org/wiki/Apache_HTTP_Server), My SQL, [Perl](http://en.wikipedia.org/wiki/Perl)/[PHP](http://en.wikipedia.org/wiki/PHP)/[Python](http://en.wikipedia.org/wiki/Python_(programming_language))." [Free-software](http://en.wikipedia.org/wiki/Free_software)-open source projects that require a full-featured database management system often use MySQL. MySQL is offered under two different editions: the open source MySQL Community Server and the proprietary [Enterprise Server](http://en.wikipedia.org/wiki/MySQL_Enterprise). MySQL Enterprise Server is differentiated by a series of proprietary extensions which install as server plugins, but otherwise shares the version numbering system and is built from the same code base. Major features as available in MySQL, A broad subset of [ANSI SQL 99](http://en.wikipedia.org/wiki/SQL:1999), as well as extensions Cross-platform support, [Stored procedures](http://en.wikipedia.org/wiki/Stored_procedure), using a procedural language that closely adheres to [SQL/PSM](http://en.wikipedia.org/wiki/SQL/PSM), [Triggers](http://en.wikipedia.org/wiki/Database_trigger), [Cursors](http://en.wikipedia.org/wiki/Cursor_(databases)), Updatable [views](http://en.wikipedia.org/wiki/View_(SQL)), [Online DDL](http://en.wikipedia.org/wiki/Data_Definition_Language) when using the Inner DB Storage Engine, [Information schema](http://en.wikipedia.org/wiki/Information_schema), Performance Schema. A set of SQL Mode options to control runtime behavior, including a strict mode to better adhere to SQL standards.

MySQL can be built and installed manually from source code, but this can be tedious so it is more commonly installed from a binary package unless special customizations are required. On most Linux distributions the [package management system](http://en.wikipedia.org/wiki/Package_management_system) can download and install MySQL with minimal effort, though further configuration is often required to adjust security and optimization settings.

**5.5. Features of CSS**

**Cascading Style Sheets** (**CSS**) is a [style sheet language](http://en.wikipedia.org/wiki/Style_sheet_language) used for describing the [look and formatting](http://en.wikipedia.org/wiki/Presentation_semantics) of a document written in a [markup language](http://en.wikipedia.org/wiki/Markup_language). While most often used to change the style of [web pages](http://en.wikipedia.org/wiki/Web_page) and user interfaces written in [HTML](http://en.wikipedia.org/wiki/HTML) and [XHTML](http://en.wikipedia.org/wiki/XHTML), the language can be applied to any kind of [XML](http://en.wikipedia.org/wiki/XML) document, including [plain XML](http://en.wikipedia.org/wiki/Plain_Old_XML), [SVG](http://en.wikipedia.org/wiki/Scalable_Vector_Graphics) and [XUL](http://en.wikipedia.org/wiki/XUL). Along with HTML and [JavaScript](http://en.wikipedia.org/wiki/JavaScript), CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for [web applications](http://en.wikipedia.org/wiki/Web_applications), and user interfaces for many mobile applications.[[1]](http://en.wikipedia.org/wiki/Cascading_Style_Sheets#cite_note-1)

CSS is designed primarily to enable [the separation of document content from document presentation](http://en.wikipedia.org/wiki/Separation_of_presentation_and_content), including elements such as the [layout](http://en.wikipedia.org/wiki/Page_layout), [colors](http://en.wikipedia.org/wiki/Color), and [fonts](http://en.wikipedia.org/wiki/Typeface).[[2]](http://en.wikipedia.org/wiki/Cascading_Style_Sheets#cite_note-2) This separation can improve content [accessibility](http://en.wikipedia.org/wiki/Accessibility), provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content, such as semantically insignificant tables that were widely used to format pages before consistent CSS rendering was available in all major browsers. CSS makes it possible to separate presentation instructions from the HTML content in a separate file or style section of the HTML file. For each matching[HTML element](http://en.wikipedia.org/wiki/HTML_element), it provides a list of formatting instructions. For example, a CSS rule might specify that "all heading 1 elements should be [bold](http://en.wikipedia.org/wiki/Bold)," leaving pure semantic HTML markup that asserts "this text is a level 1 heading" without formatting code such as a <bold> tag indicating how such text should be displayed.

This separation of formatting and content makes it possible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (when read out by a speech-based browser or [screen reader](http://en.wikipedia.org/wiki/Screen_reader)) and on[Braille-based](http://en.wikipedia.org/wiki/Braille_display), tactile devices. It can also be used to display the web page differently depending on the screen size or device on which it is being viewed. While the author of a web page typically links to a CSS file within the markup file, readers can specify a different style sheet, such as a CSS file stored on their own computer, to override the one the author has specified. If the author or the reader did not link the document to a style sheet, the default style of the browser will be applied. Another advantage of CSS is that aesthetic changes to the [graphic design](http://en.wikipedia.org/wiki/Graphic_design) of a document (or hundreds of documents) can be applied quickly and easily, by editing a few lines in one file, rather than by a laborious (and thus expensive) process of crawling over every document line by line, changing markup.

The CSS specification describes a priority scheme to determine which style rules apply if more than one rule matches against a particular element. In this so-called *cascade*, priorities or *weights* are calculated and assigned to rules, so that the results are predictable.

**6. CODING**

**6.1 DONOR Registration form:**

<?php

$conn=mysql\_connect("localhost","root","venkatesh");

$db=mysql\_select\_db("venky",$conn);

if(isset($\_POST['submit']))

{

$name1=$\_POST['name'];

$gender1=$\_POST['gender'];

$day1=$\_POST['day'];

$month1=$\_POST['month'];

$year1=$\_POST['year'];

$bloodgroup1=$\_POST['bloodgroup'];

$address1=$\_POST['address'];

$state1=$\_POST['state'];

$city1=$\_POST['city'];

$mobile1=$\_POST['mobile'];

$email1=$\_POST['email'];

$password1=$\_POST['password'];

$query = mysql\_query("SELECT \* FROM bdreg WHERE email = '$email1'");

if(!(empty($\_POST['name']) || empty($\_POST['gender']) || empty($\_POST['day']) || empty($\_POST['month']) || empty($\_POST['year']) || empty($\_POST['bloodgroup']) || empty($\_POST['address']) || empty($\_POST['state']) || empty($\_POST['state']) || empty($\_POST['city']) || empty($\_POST['mobile']) || empty($\_POST['email']) || empty($\_POST['password'])))

{

if(!$row = mysql\_fetch\_array($query))

{

$sql="INSERT INTO bdreg VALUES('$name1','$gender1','$day1','$month1','$year1','$bloodgroup1','$address1','$state1','$city1','$mobile1','$email1','$password1')";

$x=mysql\_query($sql,$conn);

if($x)

header('location: bdlog.php');

}

else{echo '<script language="javascript">';

echo 'alert("Sorry this Email is already registered.")';

echo '</script>';}

}

}

?>

**6.2. Hospital Registration form:**

<?php

$conn=mysql\_connect("localhost","root","venkatesh");

$db=mysql\_select\_db("venky",$conn);

if(isset($\_POST['sub']))

{

$hospital=$\_POST['hospital1'];

$hospaddr=$\_POST['address1'];

$contactperson=$\_POST['name1'];

$designation=$\_POST['designation1'];

$state=$\_POST['state1'];

$city=$\_POST['city1'];

$contactno=$\_POST['mobile1'];

$email=$\_POST['email1'];

$password=$\_POST['password1'];

$query = mysql\_query("SELECT \* FROM brreg WHERE email = '$email'");

if(!(empty($hospital) || empty($hospaddr) || empty($contactperson) || empty($designation) || empty($state) || empty($city) || empty($contactno) || empty($email) || empty($password)))

{

if(!$row = mysql\_fetch\_array($query))

{

$sql="INSERT into brreg values('$hospital','$hospaddr','$contactperson','$designation','$state','$city','$contactno','$email','$password')";

$query=mysql\_query($sql,$conn);

if($query)

header('location: brlog.php');

}

else

{

echo '<script language="javascript">';

echo 'alert("Sorry this Email is already registered.")';

echo '</script>';

}

}

}

?>

**6.3. Hospital Login Form:**

<?php

session\_start();

if(isset($\_POST['submit']))

{

if(empty($\_POST['name']) || empty($\_POST['password']))

{

}

else

{

$username=$\_POST['name'];

$password=$\_POST['password'];

$conn=mysql\_connect("localhost","root","venkatesh");

$username=stripslashes($username);

$password=stripslashes($password);

$username=mysql\_real\_escape\_string($username);

$password=mysql\_real\_escape\_string($password);

$db=mysql\_select\_db("venky",$conn);

$query="SELECT \* FROM brreg WHERE email='$username' AND password='$password'";

$res=mysql\_query($query,$conn);

$row = mysql\_num\_rows($res);

if($row==1)

{

$\_SESSION['login\_user'] = $username;

header("location: brlog1.php");

}

else

{echo '<script language="javascript">';

echo 'alert("Sorry You entered a wrong ID or password.")';

echo '</script>';}

mysql\_close($conn);

}

}

?>

**6.4. Donor Login Form:**

<?php

session\_start();

$conn=mysql\_connect("localhost","root","venkatesh");

$db=mysql\_select\_db("venky",$conn);

if(isset($\_POST['sub']))

{

if(empty($\_POST['name']) || empty($\_POST['password']))

{ }

else

{

$username=$\_POST['name'];

$password=$\_POST['password'];

$conn=mysql\_connect("localhost","root","venkatesh");

$db=mysql\_select\_db("venky",$conn);

$rec=mysql\_query("SELECT \* FROM bdlog");

$numrows=mysql\_num\_rows($rec);

$mailID=$numrows+1;

mysql\_query("INSERT INTO bdlog VALUES('$mailID','$username','$password')");

$query="SELECT \* FROM bdreg WHERE email='$username' AND password='$password'";

$res=mysql\_query($query,$conn);

$row = mysql\_num\_rows($res);

if($row==1)

{

$\_SESSION['name'] = $username;

header("location: inbox.php");

}

else

{echo '<script language="javascript">';

echo 'alert("Sorry You entered a wrong ID or password.")';

echo '</script>';} }

}

?>

**6.5. Donor Inbox Page:**

<?php

$conn=mysql\_connect("localhost","root","venkatesh");

$db=mysql\_select\_db("venky",$conn);

$hel=mysql\_query("SELECT name FROM bdlog ORDER BY mailID DESC LIMIT 1");

$hel2=mysql\_fetch\_array($hel);

$username=$hel2['name'];

$res=mysql\_query("SELECT \* FROM mail WHERE dmail='$username'");

echo "<table id='table'>";

echo "<tr><td><h2 id='head'>From</h2></td>

<td><h2 id='head'>Email</h2></td>

<td><h2 id='head'>Date</h2></td></tr>";

while($mail=mysql\_fetch\_array($res))

{

echo "<tr>";

echo "<td>".$mail['from']."</td>";

echo "<td>".$mail['email']."</td>";

echo "<td><a href='inbox1.php' name='sub'><form action='inbox1.php' method='post'>".$mail['date']."</form></a></td>";

echo "</tr>";

}

echo "</table>";

?>

**7.Testing**

**7.1. Introduction**

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 well-planned 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.

**7.2. Strategic Approach To Software Testing**

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.

UNIT TESTING

MODULE TESTING

SUB-SYSTEM TESING

SYSTEM TESTING

ACCEPTANCE TESTING

Component Testing

Integration Testing

User Testing

Fig 7.2.

**7.3. Unit Testing**

Unit testing focuses verification effort on the smallest unit of software design, the module. The unit testing we have is white box oriented and some modules the steps are conducted in parallel.

**1. WHITE BOX TESTING**

This type of testing ensures that

* All independent paths have been exercised at least once
* All logical decisions have been exercised on their true and false sides
* All loops are executed at their boundaries and within their operational bounds
* All internal data structures have been exercised to assure their validity.

To follow the concept of white box testing we have tested each form .we have created independently to verify that Data flow is correct, All conditions are exercised to check their validity, All loops are executed on their boundaries.

**2. BASIC PATH TESTING**

Established technique of flow graph with Cyclomatic complexity was used to derive test cases for all the functions. The main steps in deriving test cases were:

Use the design of the code and draw correspondent flow graph.

Determine the Cyclomatic complexity of resultant flow graph, using formula:

V(G)=E-N+2 or

V(G)=P+1 or

V(G)=Number Of Regions

Where V(G) is Cyclomatic complexity,

E is the number of edges,

N is the number of flow graph nodes,

P is the number of predicate nodes.

Determine the basis of set of linearly independent paths.

**3. CONDITIONAL TESTING**

In this part of the testing each of the conditions were tested to both true and false aspects. And all the resulting paths were tested. So that each path that may be generate on particular condition is traced to uncover any possible errors.

**4. DATA FLOW TESTING**

This type of testing selects the path of the program according to the location of definition and use of variables. This kind of testing was used only when some local variable were declared. The *definition-use chain* method was used in this type of testing. These were particularly useful in nested statements.

**5. LOOP TESTING**

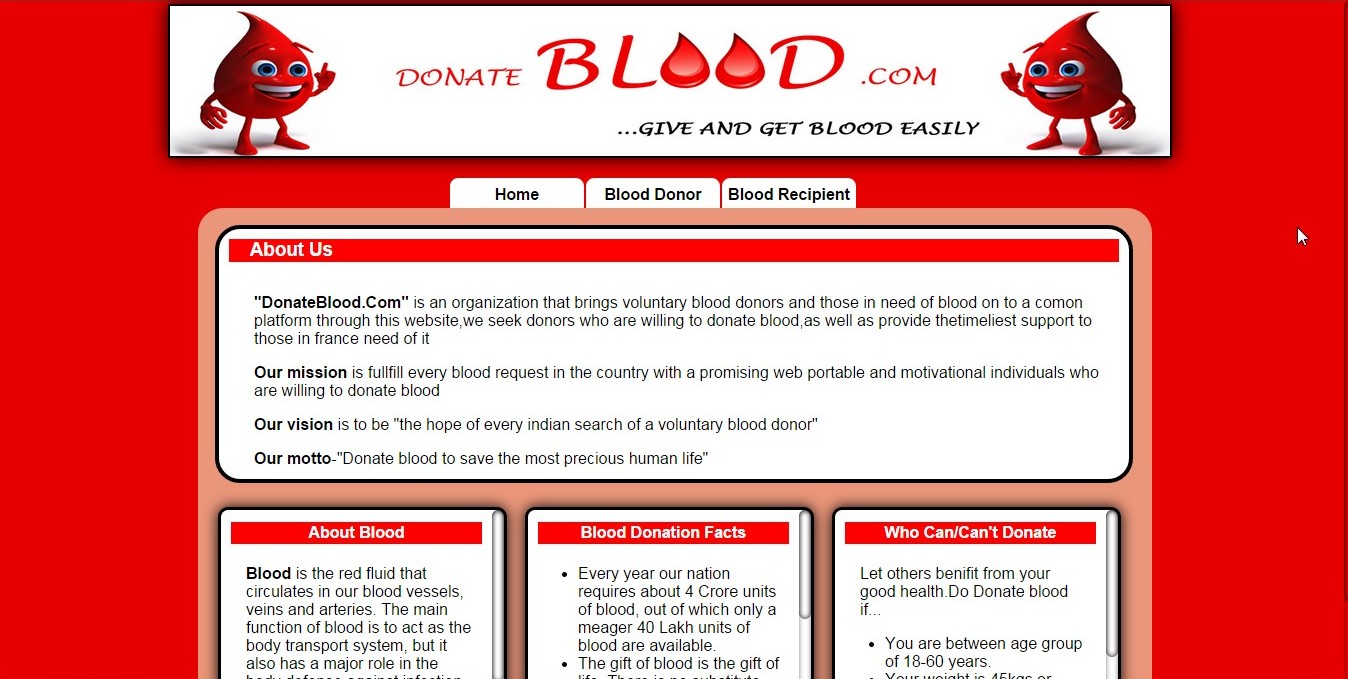
In this type of testing all the loops are tested to all the limits possible. The following exercise was adopted for all loops:

* All the loops were tested at their limits, just above them and just below them.
* All the loops were skipped at least once.
* For nested loops test the inner most loop first and then work outwards.
* For concatenated loops the values of dependent loops were set with the help of connected loop.
* Unstructured loops were resolved into nested loops or concatenated loops and tested as above.

Each unit has been separately tested by the development team itself and all the input have been validated.

**8. SCREEN SHOTS**

**8.1. HomePage**

****

**8.2. Hospital Registration Form**

****

**8.3. Blood Donor Registration Page:**

****

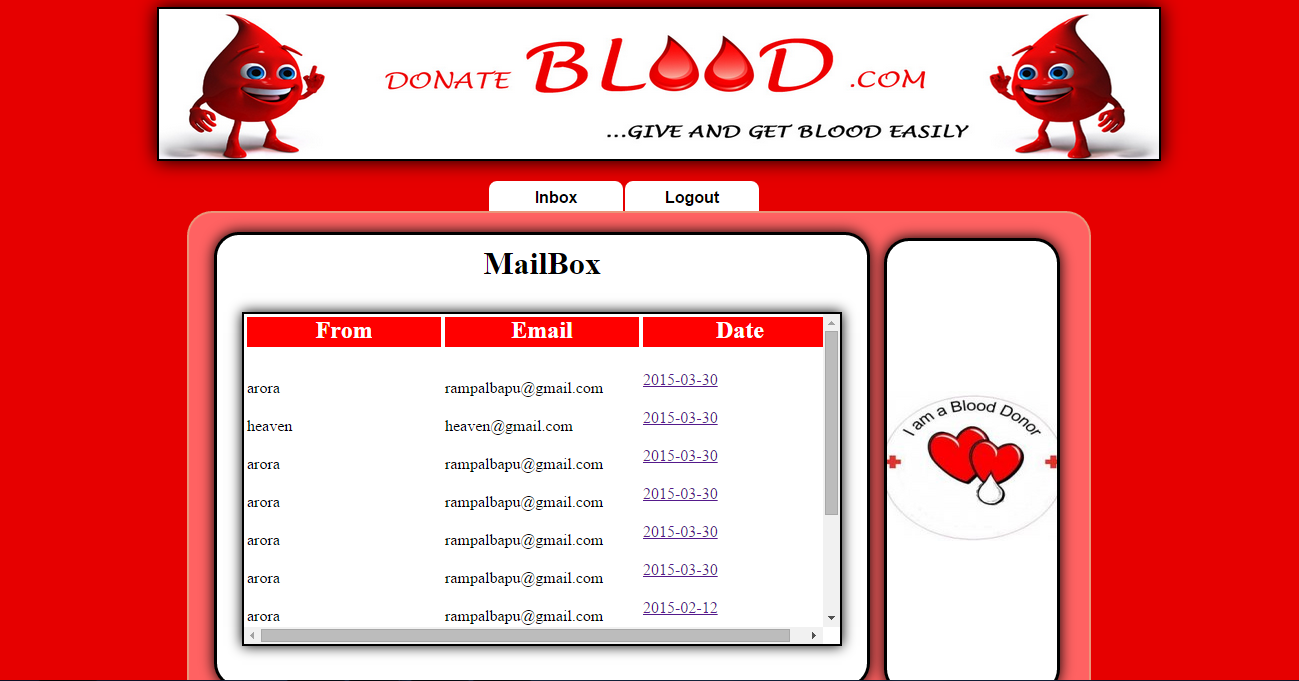
**8.4. Blood Donor Login Page:**

****

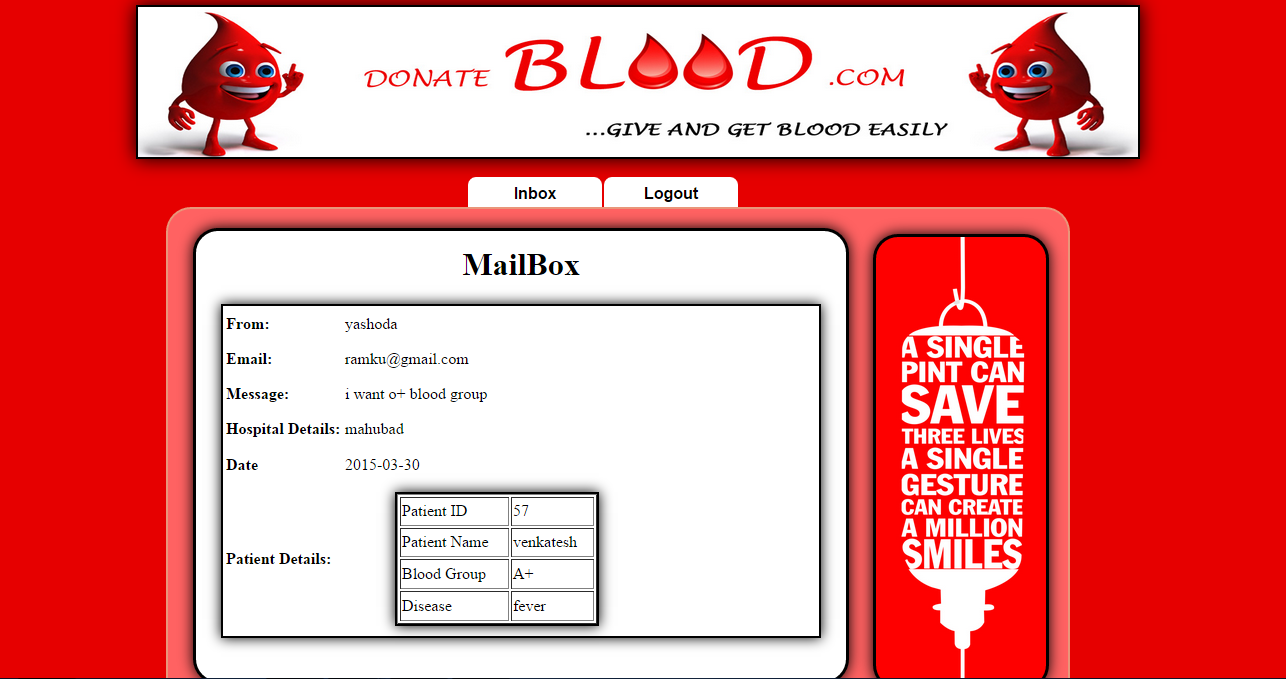
**8.5. Blood Recipient Login Page:**

****

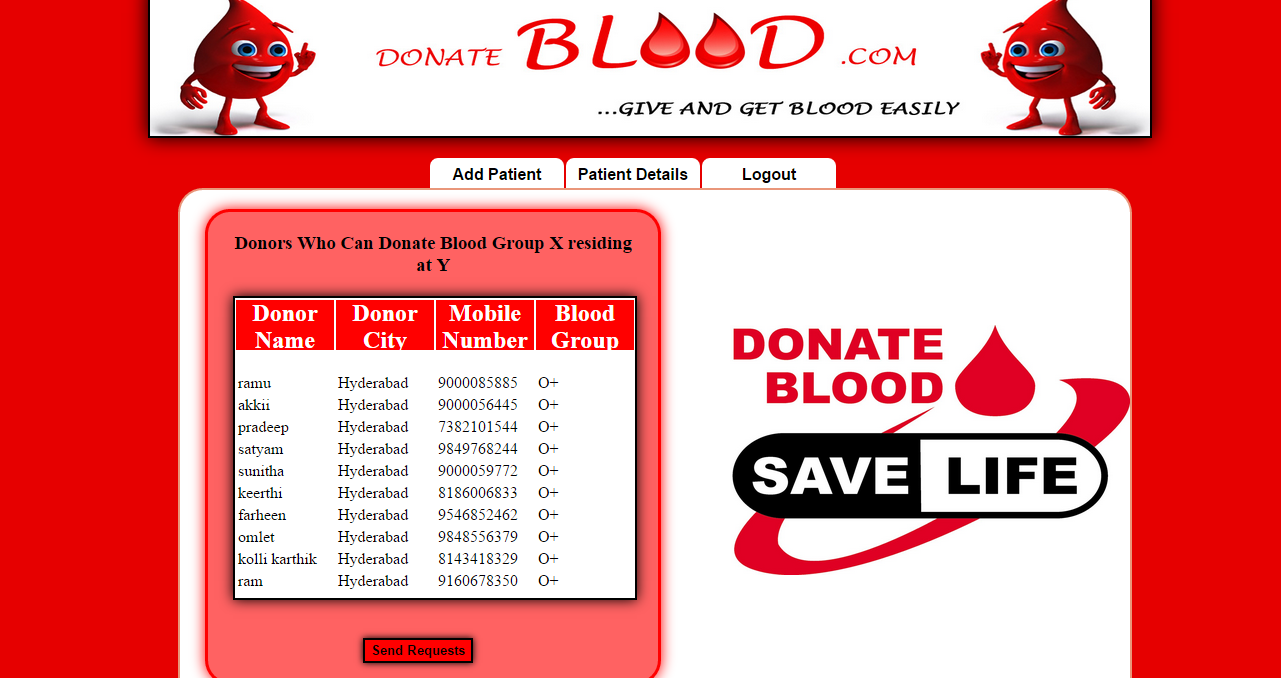
**8.6. Blood Donor Inbox:**

****

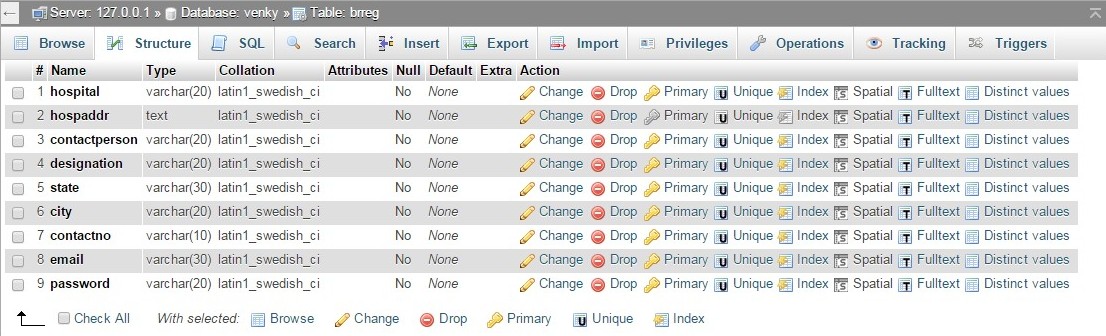
**8.7. Blood Donor MailBox:**

****

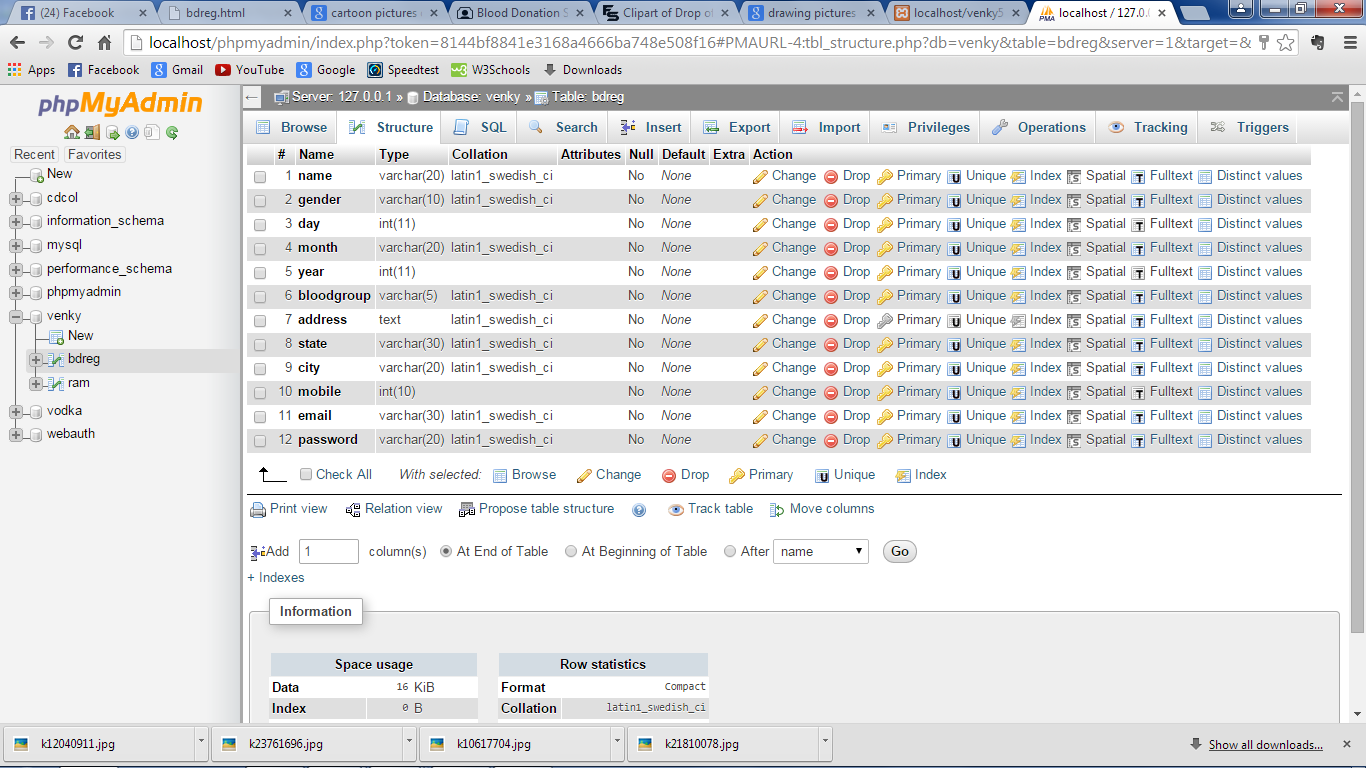
**8.8. List of Blood Donors:**

****

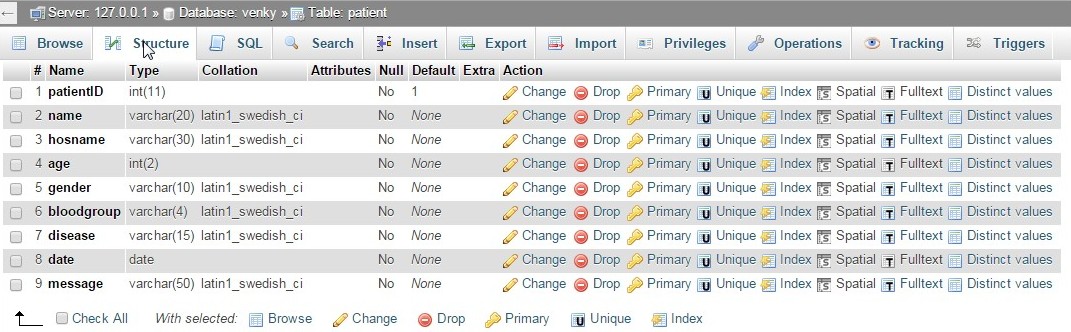
**8.9. Hospital Registration Table:**

****

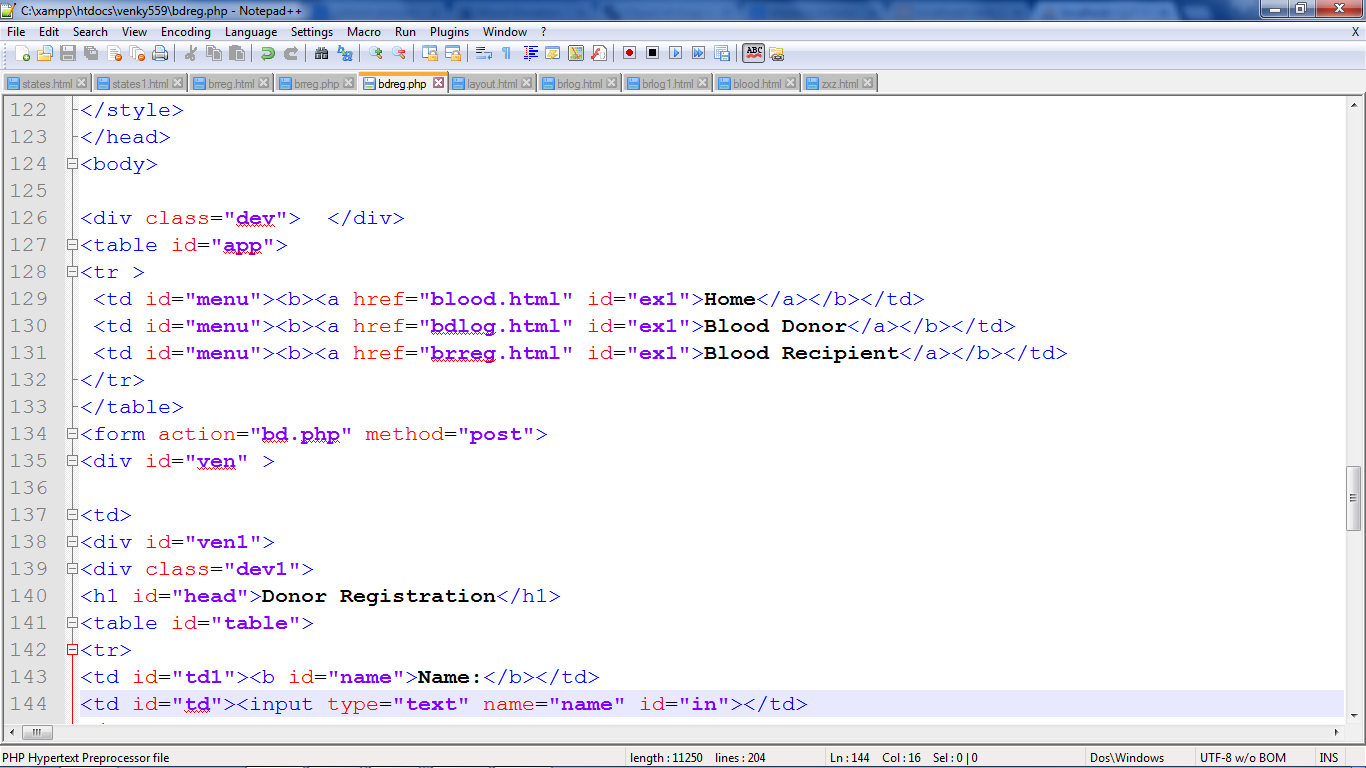
**8.10. Blood Donor Registration Table:**

****

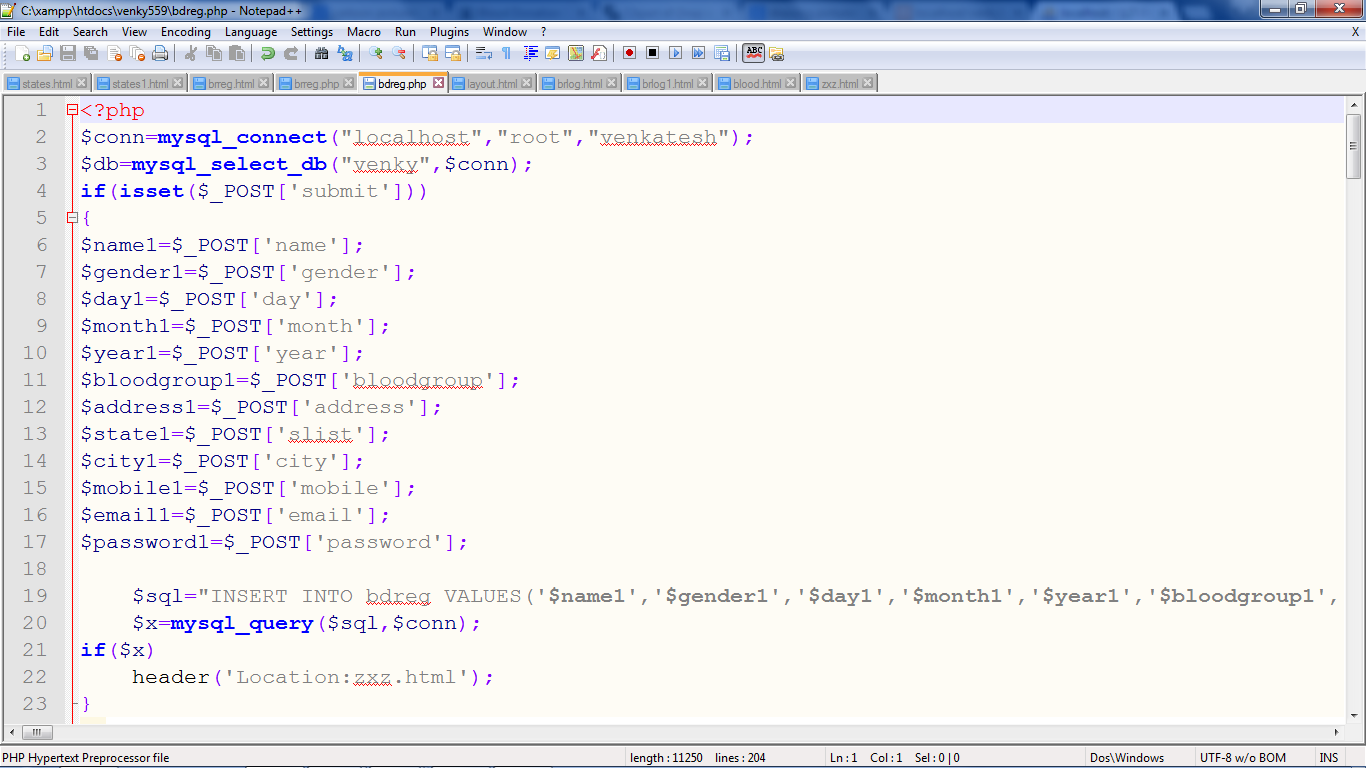
**8.11. Patient Entry Table:**

****

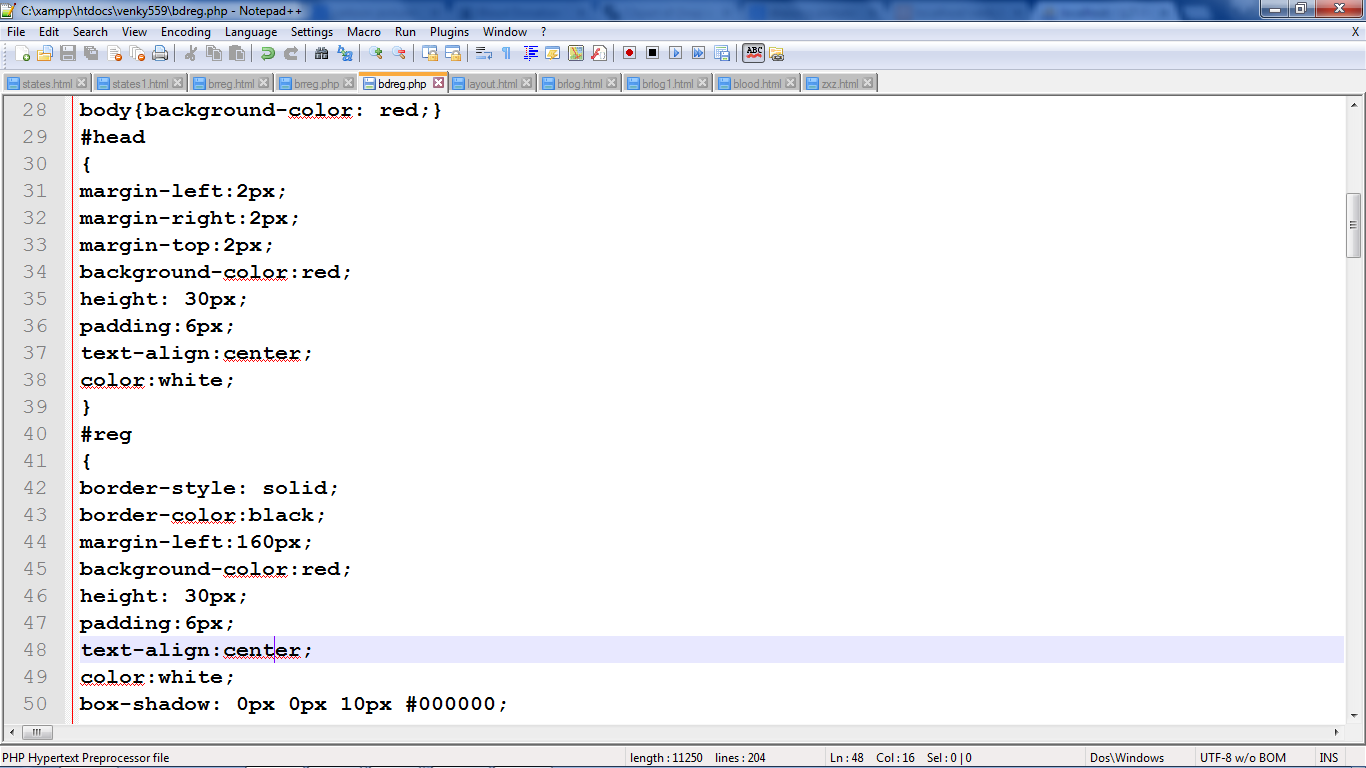
**8.12. HTML code:**

****

**8.13. PHP Code:**

****

**8.14. CSS Code:**

****

**9. CONCLUSION**

It has been a great pleasure for me to work on this exciting and challenging project. This project proved good for me as it provided practical knowledge of not only programming in HTML, CSS, PHP, Javascript, MYSQL web based application and no some extent Windows Application and SQL Server.It also provides knowledge about the latest technology used in developing web enabled application and client server technology that will be great demand in future. This will provide better opportunities and guidance in future in developing projects independently.

**BIBLIOGRAPHY**

**For Xampp Installation:**

**www.apachefriends.org/download.html**

**FOR SQL:**

**www.msdn.microsoft.com**

**For Scripting Languages:**

**www.W3Schools.com**

**www.stackoverflow.com**

**www.wikipedia.com**

**www.php.net**