

LIFE SAVER

Madhuri.K Reddy¹, Pooja Kulkarni², Farzana K³, Monika C⁴, Keerthi B⁵

I. INTRODUCTION

1.1. Overview

The purpose of blood bank was to develop a blood management information system to assist in the management of blood donor records and ease/or control the distribution of blood in various parts of the country basing on the hospital demands without quick and timely access to donor records, creating market strategies for blood donation, the blood donors become very difficult. The blood management information system offers functionality to quick access to donor records collected from various parts of the country. It enables monitoring of the results and performance of the blood donation activity such that relevant and measurable objectives of the organization can be checked. This project is aimed to developing an online Blood Bank Information. The entire project has been developed keeping in view of the distributed client server computing technology, in mind. 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. Admin is the main authority who can do addition, deletion, and modification if required.

1.2. Existing system

Entering the details about the blood groups, members, address etc. And tracking the database is complicated when the details are maintained manually. This makes the maintenance of schedule erroneous.

Limitations of the Existing system:

- It is time consuming
- It leads to error prone results
- It consumes lot of manpower to better results
- It lacks of data security
- Retrieval of data takes lot of time

1.3 Proposed System

The proposed system, Online Blood Bank site overcomes the drawbacks of the Existing system. The Blood Bank helps the people who are in need of a blood by giving them overall details regarding the donors with the same blood group and within their limits.

The advantages of the proposed system are listed below.

- The people in need of blood can search for the donors by giving their blood group and city name.
- It is very flexible and user friendly.
- The person's time and work is reduced very much which prevails in the present system.
- Easy and Helpful.

- The people are not limited to receive or provide services in working hours of the branch only; he is serviced 24 hours a day, 7 days of week and 365 days of the year.
- User friendliness is provided in the application with various controls.
- There is no risk of blood manipulation.

1.4 Problem Statement

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

With the manual system, there are problems in managing the donors' records. The records of the donor might not be kept safely and there might be missing of donor's records due to human error or disasters

1.5 Scope

The Scope of Life Saver project study was to develop a blood bank information system to assist in the management of blood donor records and ease/or control the distribution of blood in various parts of the country basing on the hospital demands. Without quick and timely access to donor records.

Mission Statement

The mission Statement of this application is to serve our community by meeting the needs of patients, hospitals, and members for safe, high quality blood products and related services.

Vision Statement

Life Saver application is committed to accomplishing its mission and meeting the challenges of the future by:

- Anticipating and responding to customer needs.
- Seeking opportunities for continuous improvement.
- Building trust between the organization, its membership, and the community.
- Providing education which enhances the understanding of transfusion medicine and the need for blood.
- Enhancing the quality, efficiency, and effectiveness of the organization through teamwork.
- Providing a work environment where individuals are valued and their contributions are recognized.

Objectives

- To create life saver application which provides the information system in order to assist in the management of blood donor records, Blood Bank details and the recipient details in order to planning and share information in a more confidential, convenient and secure way using modern technology.

- To provide safe and quality blood within an hour of requirement in a health facility through a well coordinated blood transfusion service.

II. REQUIREMENTS

2.1 Functional & Non functional requirements

2.1 Functional Requirements

The main purpose of functional requirement is to define all the activities or operations that take place in the system. These are derived through interactions with the users of the system. There are three main modules, they are

- Administration
- Donor
- Blood Request:

Admin Module

Admin is provided with Username and Password to avoid unauthorized access. In this module the Administrator has the privileges to add all the Blood Groups, Blood Type, Organization, Type, Country, State, City, and Location and can also check the donor details view the recipient details.

Donor Module

Donor is the person who is interested in donating their blood so they can register themselves through this website. If any requirement comes then they will be contacted and they can donate their blood. Along with it they can search for the various organization locations wise and can also make request for blood if needed

Blood Request:

The recipient who required the blood can send their details necessary details to the admin and he can provide the service to the recipient

2.2 Non Functional Requirements

Reliability:

The system must be highly reliable as it would be handling critical data regarding the project.

Unauthorized person should not able to access the details.

This system must perform all of its operations with high accuracy.

Availability

The system must work in relatively fast and must provide the data on request as soon as possible without affecting the quality & accuracy

Security

The System must be highly secured and must authenticate users strictly.

The System would require handling confidential data and thus must provide security towards both front & back end.

Maintainability

The maintainability of the system must be high.

Proper documentation must be provided so as to perform enhancement, adaptation and to fix bugs (if any).

2.3 Hardware Requirements

It provides a structured document setting out detailed descriptions of the systems functions, services & operational constraints. It defines what should be considered such as hardware tools in this application.

Processor	-	Pentium –IV
RAM	-	256 MB to 512 MB
Hard Disk	-	40 GB to 80 GB
Key Board	-	108 Keys
Mouse	-	Optical Mouse

2.4 Software Requirements

The software requirement specification deals with the part of the system that has human users, and would describe the functionality from a user’s perspective.

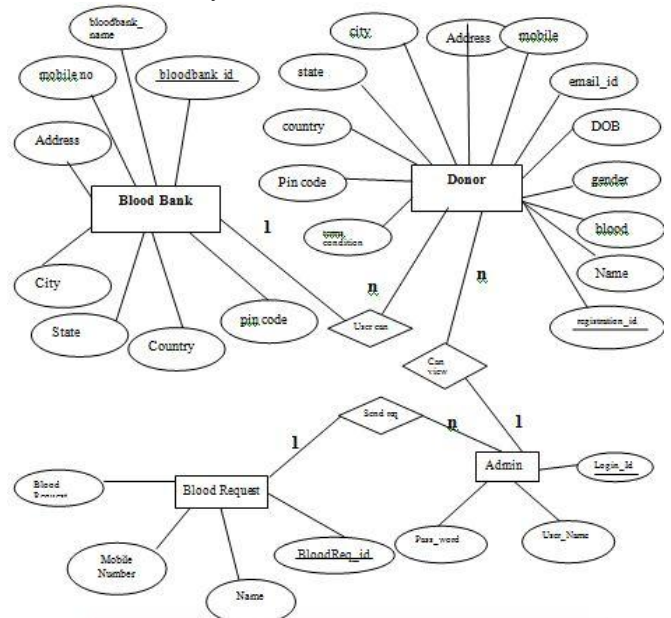
As software requirement are basically an organization understands of a client’s requirements at a particular point in the time prior to any actual design or the development work. It established the basis for agreement that assures that both the client and the organization understanding each other’s requirements.

Operating System	:	Windows 2000 and above version
Front End Tool	:	Java, JSP, HTML
Back End	:	MySQL
Web Server	:	Tomcat 6.0
Web Browser	:	IE 5.0/6.0/7.0
Developing Tool	:	Eclipse 3.2

III. DESIGN

3.1 ER-Diagram

An entity-relationship diagram (ERD) is a graphical representation of an information system that shows the relationship between people, objects, places, concepts or events within that system.

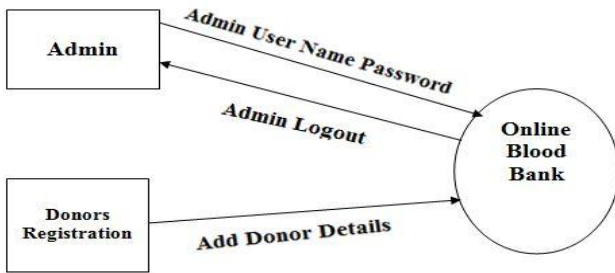


3.2 Data Flow Diagram

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system. A data flow diagram can also be used for the visualization of data processing (structured design). It is common practice for a designer to draw a context-level DFD first which shows the interaction between the system and outside entities. DFD’s show the flow of data from external entities into the system,

how the data moves from one process to another, as well as its logical storage.

Context Level Diagram



DFD LEVEL 0

Fig 3.2.1: Context level diagram

Level 1: Admin

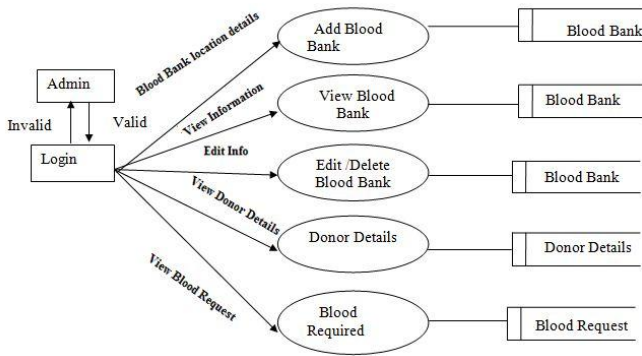


Fig 3.2. 2: Level 1 diagram for admin

3.3 Use case Diagram

A use case diagram presents a collection of use cases and actors and is typically used to specify or characterize the functionality and behavior of a whole application system interacting with one or more external actors.

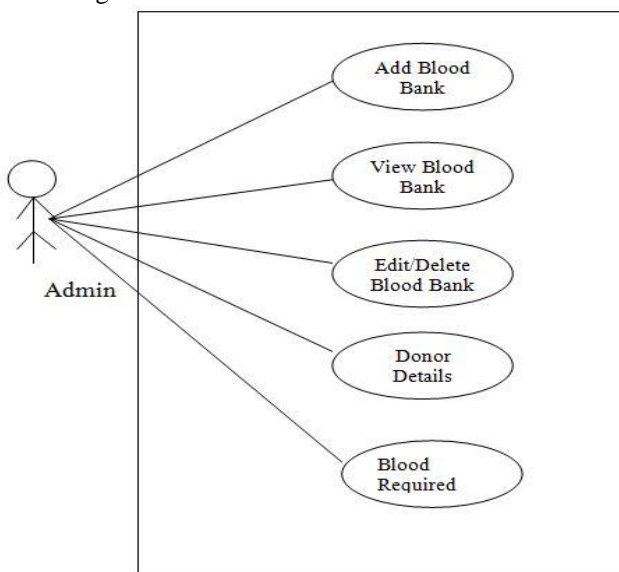


Fig 3.3.1: Use case diagram for admin

3.4 Activity Diagram

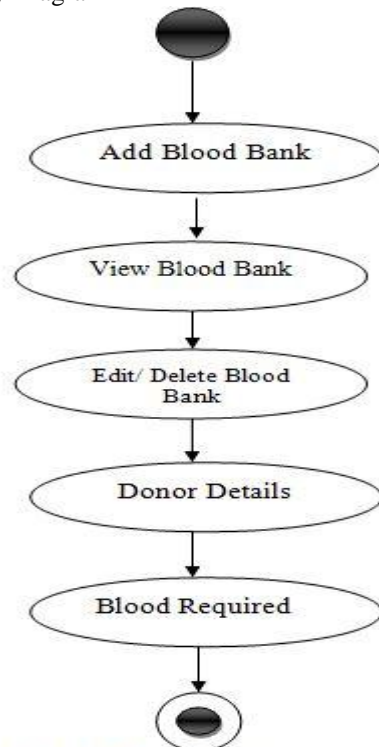


Fig 3.4.1: Activity diagram

3.5 SEQUENCE DIAGRAM

A sequence diagram is an interaction diagram that details how operations are carried out what messages are sent and when. Sequence diagrams are organized according to time. The time progresses as you go down the page. The objects involved in the operation are listed from left to right according to when they take part in the message sequence.

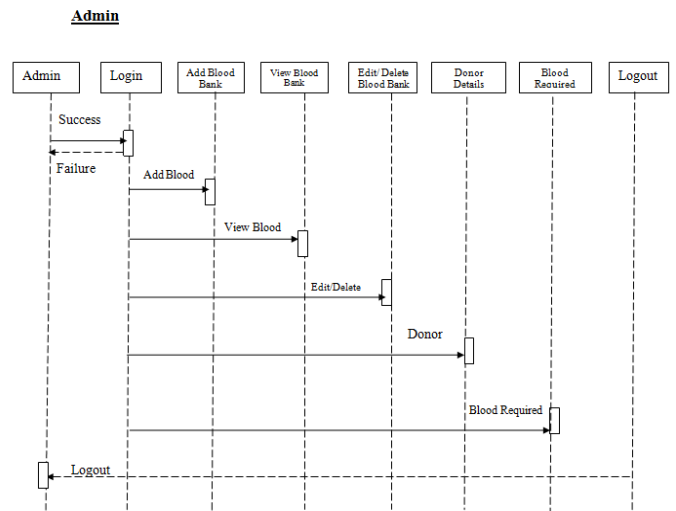


Fig 3.5.1: sequence diagram for admin

IV. IMPLEMENTATION

Implementation is the stage of the project where the theoretical design is turned into a working system. Software development involves translation of design specifications into source code by using required platform and other tools.

The entire software is implemented using JSP, JAVA, MySQL. The most crucial stage is achieving a new successful system and giving confidence on the new system for the user that it will work efficiently and effectively. It involves careful planning, investigation for the current system and its constraints on system being implemented. To implement life saver following modules are implemented:

Admin

- add blood bank details
- view blood bank details
- edit blood bank details
- delete blood bank details
- view donor details
- view blood request details

Donor registration

- add donor details

Blood request Details

- 3.1request blood

ADMIN

Admin will login with the proper login authentication, after successful login he will check the blood donor details and view the blood request user details and can the blood details.

DONOR

In this application the person who is interested to donate the blood they can register in this application by giving the necessary detail.

BLOOD REQUEST DETAILS

The person who required the blood they can send message to the admin by giving the user details like name, email id, contact number and blood group

Platform Used

Technology

HTML

HyperText Markup Language commonly referred to as HTML is the standard markup language used to create web pages. Web browsers can read HTML files and compose them into visible or audible web pages. Browsers do not display the HTML tags and scripts, but use them to interpret the content of the page. HTML describes the structure of a website semantically along with cues for presentation, making it a markup language, rather than a programming language.

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 written in languages such as JavaScript which affect the behavior of HTML web pages. Web browsers can also refer to Cascading Style Sheets (CSS) to define the look and layout of text and other material. The World Wide Web Consortium (W3C), maintainer of both the HTML and the CSS standards, encourages the use of CSS over explicit presentational HTML.

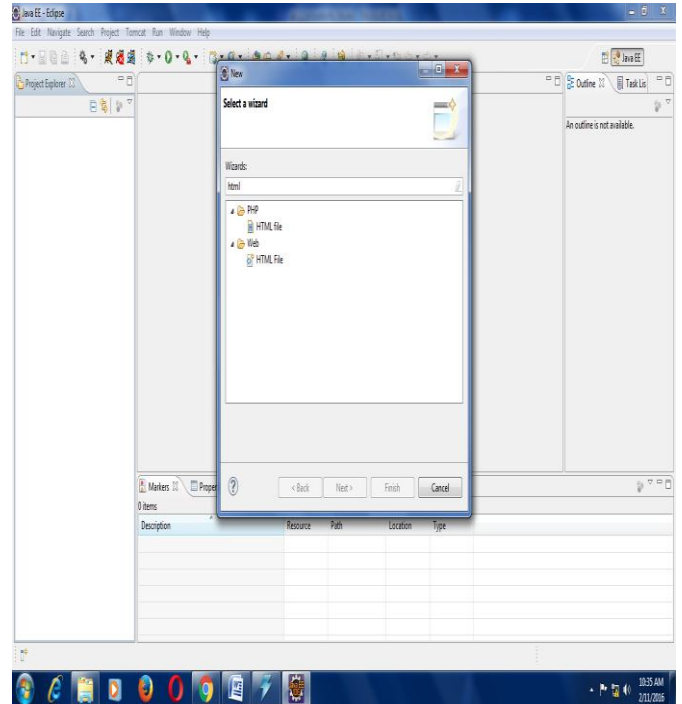


Fig 4.1 HTML web page

JavaServer Pages (JSP) is a technology that helps software developers to create dynamically generated web pages based on HTML, XML, or other document types. Released in 1999 by Sun Microsystems, [1] JSP is similar to PHP, but it uses the Java programming language.

The JSP Model 2 architecture:

Architecturally, JSP may be viewed as a high-level abstraction of Java servlets. JSPs are translated into servlets at runtime; each JSP servlet is cached and re-used until the original JSP is modified.

JSP can be used independently or as the view component of a server-side model-view-controller design, normally with JavaBeans as the model and Java servlets (or a framework such as Apache Struts) as the controller. This is a type of Model 2 architecture. JSP allows Java code and certain pre-defined actions to be interleaved with static web markup content, with the resulting page being compiled and executed on the server to deliver a document. The compiled pages, as well as any dependent Java libraries, use Java bytecode rather than a native software format. Like any other Java program, they must be executed within a Java virtual machine (JVM) that integrates with the server's host operating system to provide an abstract platform-neutral environment.

JSPs are usually used to deliver HTML and XML documents, but through the use of Output Stream, they can deliver other types of data as well. The Web container creates JSP implicit objects like page Context, servlet Context, session, request & response.

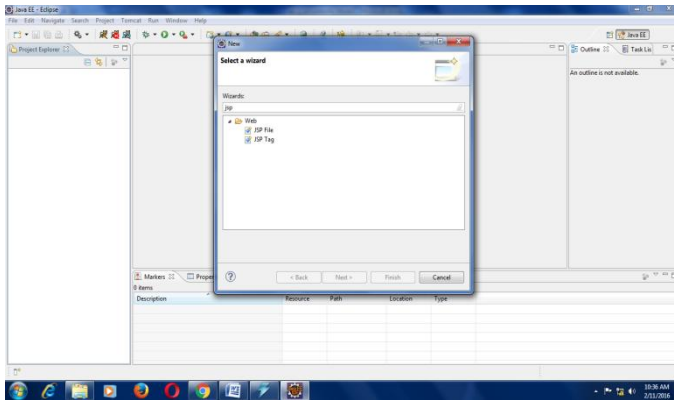


Fig 4.2: JSP home page

Software Specification

TOOLS AND TECHNOLOGY USED:

Eclipse IDE In computer programming, Eclipse is an integrated development environment (IDE). It contains a base workspace and an extensible plug-in system for customizing the environment. Eclipse is written mostly in Java and its primary use is for developing Java applications, but it may also be used to develop applications in other programming languages through the use of plugins, including: Ada, ABAP, C, C++, COBOL, Fortran, Haskell, JavaScript, Lasso, Lua, NATURAL, Perl, PHP, Prolog, Python, R, Ruby (including Ruby on Rails framework), Scala, Clojure, Groovy, Scheme, and Erlang. It can also be used to develop packages for the software Mathematical. Development environments include the Eclipse Java development tools (JDT) for Java and Scala, Eclipse CDT for C/C++ and Eclipse PDT for PHP, among others.

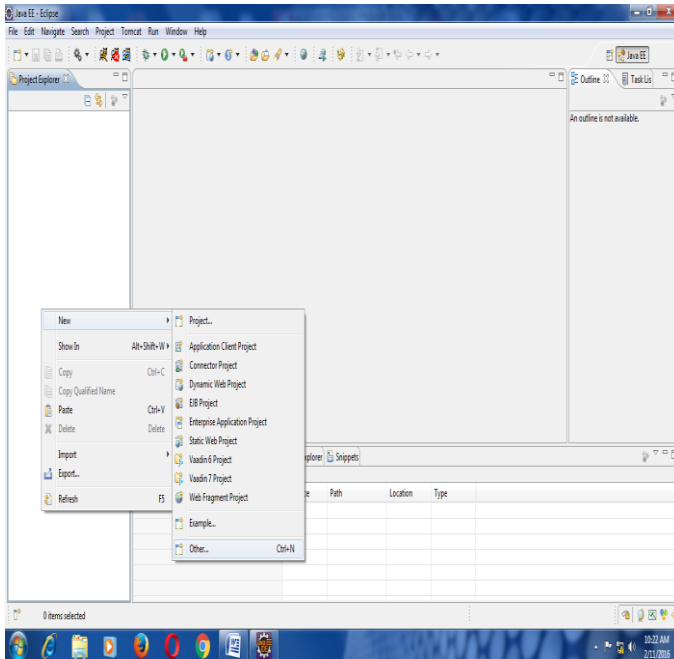


Fig 4.3: Eclipse

Apache Tomcat

Apache Tomcat, often referred to as Tomcat, is an open-source web server developed by the Apache Software Foundation (ASF). Tomcat implements several Java EE specifications including Java Servlet, JavaServer Pages (JSP), Java EL, and WebSocket, and provides a "pure Java" HTTP web server environment for Java code to run in. Tomcat is developed and maintained by an open community of developers under the auspices of the Apache Software Foundation, released under the Apache License 2.0 license, and is open-source software.



If you're seeing this page via a web browser, it means you've setup Tomcat successfully. Congratulations!

As you may have guessed by now, this is the default Tomcat home page. It can be found on the local filesystem at:

`SCATALINA_HOME/webapps/ROOT/index.html`

where "SCATALINA_HOME" is the root of the Tomcat installation directory. If you're seeing this page, and you don't think you should be, then you're either a user who has arrived at new installation of Tomcat, or you're an administrator who hasn't got his/her setup quite right. Providing the latter is the case, please refer to the [Tomcat Documentation](#) for more detailed setup and administration information than is found in the INSTALL file.

NOTE: For security reasons, using the manager webapp is restricted to users with role "manager". Users are defined in `SCATALINA_HOME/conf/tomcat-users.xml`.

Included with this release are a host of sample Servlets and JSPs (with associated source code), extensive documentation, and an introductory guide to developing web applications.

Tomcat mailing lists are available at the Tomcat project web site:

- users@tomcat.apache.org for general questions related to configuring and using Tomcat
- dev@tomcat.apache.org for developers working on Tomcat

Thanks for using Tomcat!

Administration
[Status](#)
[Tomcat Manager](#)

Documentation
[Release Notes](#)
[Change Log](#)
[Tomcat Documentation](#)

Tomcat Online
[Home Page](#)
[FAQ](#)
[Bug Database](#)
[Open Bugs](#)
[Users Mailing List](#)
[Developers Mailing List](#)
[IRC](#)

Miscellaneous
[Servlets Examples](#)
[JSP Examples](#)
[Sun's Java Server Pages Site](#)
[Sun's Servlet Site](#)



Copyright © 1999-2010 Apache Software Foundation
 All Rights Reserved

MySQL Database

- A database is a collection of data that is organized so that it may be easily searched and updated
- The most important feature of a database is its organization, which supports both, ease of use and efficient data retrieval.
- Consider an office that is organized with numbered file cabinets containing carefully labeled folders. Subject in specific folders that are kept in designated file cabinets' stores office information.
- In such a system, every folder has its place and it is easy to find a particular folder.
- In a relational database, such as those created using Access, you store information about different subjects in separate tables. To bring the information together in a meaningful way, you then tell Access how the different subjects relate to each other.
- To design a database, follow the steps given below
- Determining the purpose of the database helps to access to store.
- Determining the tables.
- Divide the information into separate subjects, such as employees or orders. Each subject will be a table in the database.

Login to Mysql Data Base

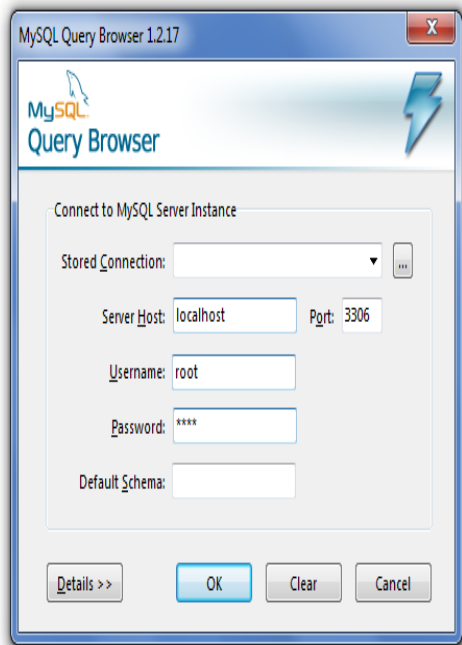


Fig 4.4 Login for MySQL

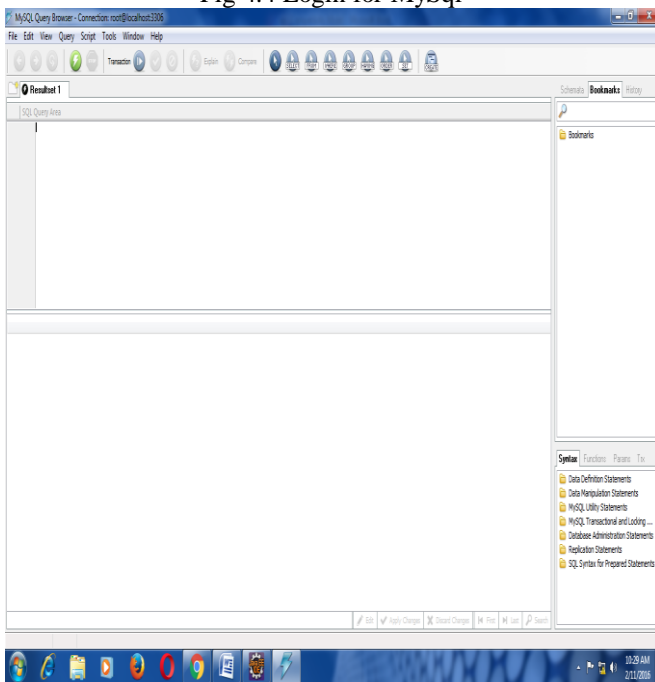


Fig 4.5 Mysql Home Page

V. TESTING

The code is tested at various levels in software testing. Unit, system and user acceptance testing's often performed. This is a grey area as many different opinions exist as to what the stages of testing are and how much if any iteration occurs. Iteration is not generally part of the waterfall model, but

usually some occur at this stage.

Software testing is the execution of program to find its faults. The testing process focuses on the logical internals of the software, ensuring that all statements have been tested and on the functional externals, that is conducting test to uncover errors and ensure that defined inputs will produce actual results agreed with required results. The following test strategies were adopted to test the system.

Testing Objective:

Testing is process of executing a program with the intent of finding an error. A good test case is one that high probability of finding an as yet undiscovered error. A successful test is one that uncovers an as yet undiscovered error. The above objectives imply a dramatic change in view point. Testing cannot show the absence of defect, it can only show that software errors are present.

5.1 Testing Strategies:

There are two general strategies for testing software. These are as follows:

Code Testing: This examines the logic of the program. To follow this test, cases are developed such that every path of program is tested.

Specification Testing: Specification Testing examines the specification starting what the program should do and how it should perform under various conditions. Then test cases are developed for each condition and combinations of conditions and to be submitted for processing.

Testing Method Used:

Here black box testing and statistical testing are used. In black box testing, all possible types of inputs and seen for corresponding outputs and if not giving, code are corrected. In statistical testing, checking for all variables whether they assigned values before using it, whether array bound correctly defined, whether looping statement terminating without going to infinite loop, whether function parameter are passed in order and about number of parameters etc., are checked successfully and found correct everything working satisfactorily.

The stages in testing process are:

Unit Testing: Individual components are tested to ensure that they operate correctly. Each component tested independently without other system components.

Ex. Checked for Username and Password with the table, after the next module is loaded session allocation.

Integration Testing: Integration testing is a systematic technique for constructing the program structure while at the same time conducting test to uncover errors associated with interfacing. This testing is done using the bottom-up approach to integrate the software components of the software system in to functioning whole.

System Testing: System testing is actually a series of different tests whose primary purpose is fully to exercise the computer-based system. The system tests that where applied are recovery testing and performance testing. Finally a review or audit is conducted which is a final evaluation that occurs only after operating the system long enough for user to have gained a familiarity with it. System testing was done

by the inspection team to verify that all the functionality identified is the software requirement specification has been implemented. Defects that crept in the system has been found defect free and is working well. System testing is concerned with interfaces, design logic, control flow recovery, procedures throughput, capacity and timing characteristics of the entire system. For blank field, alphabets, number and special character validation.

Acceptance Testing: User acceptance of the system is the key factor for the success of any system. This is done by user. The system is given to the user and they test it with live data. Acceptance testing involves the planning and execution of functional test. Performance tests, stress tests in order to demonstrate that the implemented system satisfies its requirements. Two sets of acceptance test can be run, those developed by the customer. The system has been tested for its performance at unit level by the individuals through performance testing that is designed to test the run time performance of the software. The performance of the fully integrated system is tested and was found good.

Unit test cases to test the Donor registration

S1 Test Case:-	1
Name of the Test:-	Donor Registration
Item being tested:-	Donor registration module
Sample Input:-	Select Name, Email, Gender, Username, and password.
Expected Output:-	Donor should be register to the system
Actual Output:-	Donor registration Successful
Remarks:-	Pass

Unit test cases to test the Admin login

S1 Test Case:-	2
Name of the Test:-	Admin Login
Item being tested:-	Admin
Sample Input:-	Enter Username, Password.
Expected Output:-	Admin should be logged in if enter correct user name and password
Actual Output:-	Admin logged in successfully with correct user name and password
Remarks:-	Pass

Unit test cases to test the Recipient

S1 Test Case:-	3
Name of the Test:-	Recipient details
Item being tested:-	Recipient Request module
Sample Input:-	Select Name, Email, Contact No, Blood Group
Expected Output:-	Request sent to the admin
Actual Output:-	Request registration Successful
Remarks:-	Pass

System Testing

Here the entire software system is tested. The reference document for this process is the requirements document, and the goals to see if software meets its requirements. Here entire ‘’ has been tested against requirements of project and it is checked whether all requirements of project have been satisfied or not.

S1 Test Case:-	STC
Name of the Test:-	System testing in various of OS
Item being tested:-	OS Compatibility
Sample Input:-	Execute the program in windows Xp/ Vista/Windows-7, 8.
Expected Output:-	Project will be executed in various OS
Actual Output:-	Same as expected Output
Remarks:-	Pass

VI. SNAPSHOTS

6.1 Admin login:

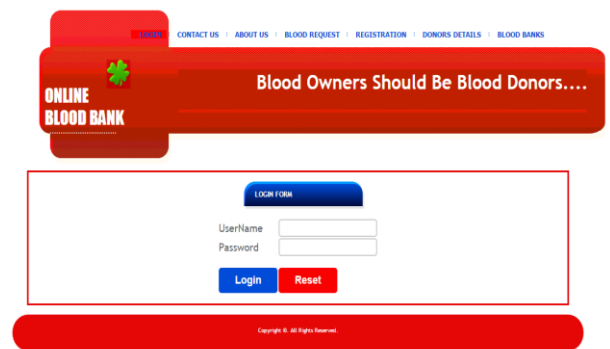


Fig 6.1: Admin Login Page

6.2 Admin home page:



Fig 6.2: Admin Home Page

6.3 Blood Request:



Fig 6.3: Request for blood

6.6 Edit Blood Bank:

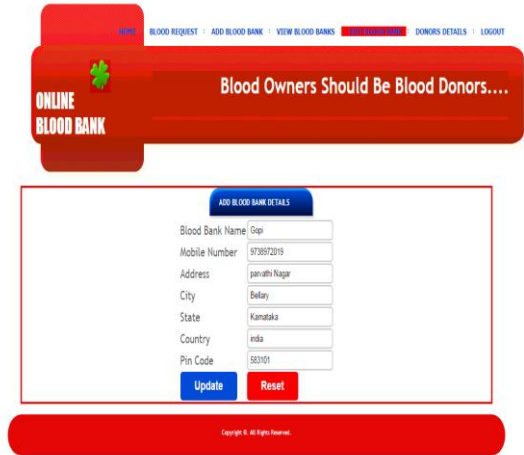


Fig 6.6 Edit blood bank details

6.4 Add Blood Bank:

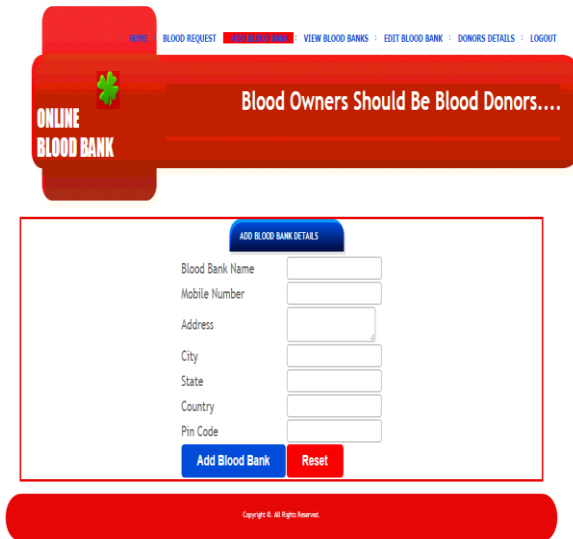


Fig 6.4: Add new blood bank

6.7 Donor Details:

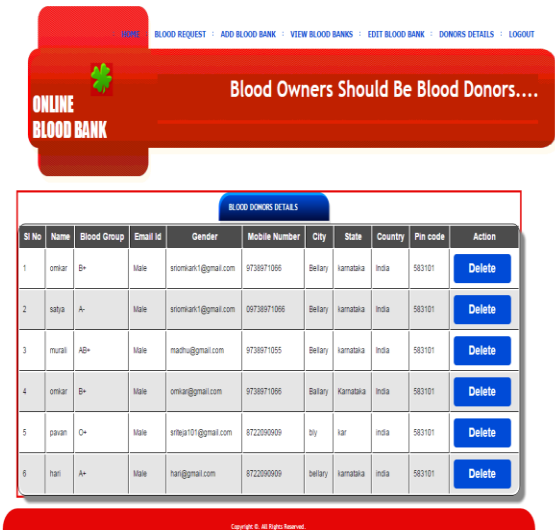


Fig 6.7 Donor details

6.5 View Blood Bank:



Fig 6.5: Blood bank details

6.8 Blood Request:



Fig 6.8: User requesting blood

6.9 Donor Registration:



Fig 6.9: Registration of donors

6.10 Blood Availability details:

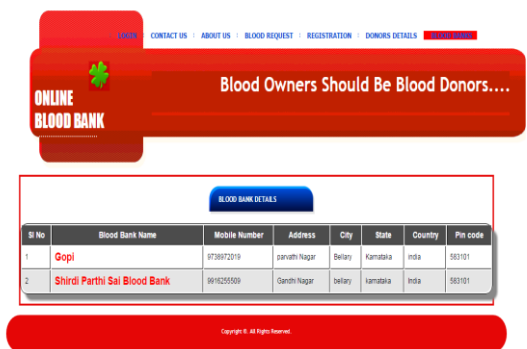


Fig 6.10 : Checking Blood availability

VII. CONCLUSION & FUTURE ENHANCEMENT

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.

FUTURE ENHANCEMENT

In future enhancement we have decided to provide more advanced look and feel to our system so as to make our system more attractive. As system we have proposed is an online system which can be connected through the internet so our system is not restricted to a single computer but also other clients can access our system through internet.

BIBLIOGRAPHY

[1] R. Buyya, C. ShinYeo, J. Bromberg, and I. Brandic, "Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility," Future Generation Computer. Syst., vol. 25, pp. 599–616, 2009.

[2] Amazon Elastic Compute Cloud (Amazon EC2) [Online]. Available: <http://aws.amazon.com/ec2/>

[3] Amazon Web Services (AWS) [Online]. Available: <https://s3.amazonaws.com>

[4] R. Martin, "IBM brings cloud computing to earth with massive new data centers," InformationWeekAug.2008 [Online]. Available: http://www.informationweek.com/news/hardware/data_centers/209901523

[5] Google App Engine [Online]. Available: <http://code.google.com/appengine>

[6] K. Barlow and J. Lane, "Like technology from an advanced alien culture: Google apps for education at ASU," in Proc. ACM SIGUCCSUser Services Conf., Orlando, FL, 2007.

[7] Core Java TM 2 volume 1-Fundamental 7th Edition Pearson Education-Sun Microsystems, Cay S. Hortsman,Gary cornell.

[8] Head First Servlets & JSP O'Reilly-SPD Eric Freeman, Elisabeth Freeman.

[9] Effective Java-Programming Language Guide Pearson education, Joshua Bloch