**CHAPTER 1**

**INTRODUCTION**

Emergency situations, such as accidents, create an immediate, critical need for specific blood type. In addition to emergency requirements, advances in medicine have increased the need for blood in many on-going treatments and elective surgeries. Despite increasing requirements for blood, only about 5% of the Indian population donates blood. We have to create a new idea, just touch the button. Donor will be prompted to enter an individual’s details, like name, phone number and blood type. After that the contact details will appear on the screen; the urgent time of a blood requirement one can quickly check for contacts matching a particular or related blood group and reach out to them via Phone call/SMS through this website. This website provides list of donors in the city/area. A large number of blood donors are attracted using this website. Since almost everyone carries a mobile phone with them, it ensures instant location tracking and communication. This website will help users easily to find donors of matching blood groups in their location and access their mobile numbers for instant help.

**1.1 Organization of report**

The information in this report is organized into the following chapters as shown in Table 1.1. Each chapter gives the information about the project development.

|  |  |  |
| --- | --- | --- |
| **SL no.** | **Chapters** | **Description** |
| 01 | Introduction | Describes the problem statement, solution of the problem, objectives of the project, advantages and limitations of study. |
| 02 | Literature survey | It is the brief history of our project. |
| 03 | System requirements and specification | This chapter describes the functional overview and software and hardware requirements. |
| 04 | System design | This chapter includes project description, and dataflow diagrams. |
| 05 | Software environment | This chapter describes the software used to built the project. |
| 06 | Programming techniques | This chapter describes the languages used to write the code. |
| 07 | Implementation details | This chapter consists of implementation details, and dataflow diagrams. |
| 08 | Testing | This chapter encloses few test cases and results. |
| 09 | Resulta & Snapshots | Describes execution of the project and results obtained. |
| 10 | Conclusion | This chapter covers conclusion and future enhancements. |

**Table 1.1 Organisation of report**

**1.2 Existing system**

As in statement we have seen the word like “Donate Blood Save Life “, which is very good job by the people for the people. Donating a blood is done in many ways like donate in some important events or else in blood donation camp etc...

But many times situation occurs where many people need blood on immediate situation like accident, some major operations where even blood banks also not having that blood group for instance.

Blood bank organization is one where blood is reserved for operations and emergencies, taking example of different blood bank organizations which is residing in different places having different blood group information.

When someone need blood in emergency for patients who in some other hospital which is far away from the blood bank organization at that time it is difficult to go to ask availability of blood which is required for the patients is very tedious task, in this situation we need immediate solution regarding of getting information about availability of blood in all the blood banks with quantity of blood available.

In the present system we can get only those blood for the patients which is present in the blood bank, if you need the blood which is not available in the blood bank then we have to search the donors for the blood and take them to the blood bank and collect the blood from them, but now it is difficult to get those people on emergency situation for donating blood.

**1.3 Limitations**

* The whole website depends on the availability of internet.
* The website will not work if the server is down and the user might have to wait for the fixation of the server or may have to depend on the backup server.
* All the blood banks should use the same software to update the blood availability.
* Erroneous updates by the hospital management may misguide the recipients.

**1.4 Problem statement**

The main aim of the project is to effectively manage the blood banking system. This project enables the users to access blood bank and donors, it also checks the availability of the required blood group, provides necessary information about the volunteer blood donors.

* 1. **Advantages**
* The website educates the know how about the blood banking system.
* Enables the users to easily locate the nearest blood banks.
* The users to request for the necessary blood units to a large number of donors if they fail to find blood in blood banks.
* Hospitals can also effectively find the volunteer blood donors.
* Nowadays most of the people are equipped with the smart phones which creates a greater exposure for the blood banking system which saves considerably greater number of lives.
  1. **Applications**
* This website is used in blood banks.
* It is also used in hospitals to get required blood.
* This website can be used by the people while there is an ememrgency due to the unexpected accidents.
  1. **Scope of project**
* The system will provide the user the option to look at the details of the existing donor list, blood group and to add a new donor.
* Blood recipient can search for the required blood group in the blood bank through this website.
* Recipient can also search for the donors who have registered using this website.
* It also allows the donor to modify the record.

.

.

**CHAPTER 2**

**LITERATURE SURVEY**

**2.1 Blood Bank and Blood Bank Information Management System**

Blood banking refers to the process of collecting, testing, preparing, and storing whole blood and blood components intended primarily for transfusion. Blood bank center entails the rigorous controlling, monitoring and the complete documentation of the whole procedure from blood collection to blood infusion.

Blood Bank Information Management System is an information system which helps to manage the records of donors and patient at a blood bank. It is mainly designed to store, process, retrieve and analyze information concerned with the administrative and inventory management within a blood bank. Such kind of system will allow the authorized blood bank officer to login using a secret password and easily manage the records of the blood donors and the patients in need of blood. In addition, the blood bank information management system is not obsolete to the professionals; rather it plays a great role in attracting the donors and other stakeholders due to its simplicity in the reservation, and notification of donation time to the needy. On top of this any stakeholder and/or regional blood banks in need of blood can easily access to the type and quantity of blood available in any blood bank center.

**2.2 Information System Development**

An Information system consists of input messages, message processing and output messages. It has also processing rules which control the execution of the Information system. If the processing rules are formalized we can have computer based Information systems. But if the processing rules need a lot of personal knowledge, judgment and Intuition, the information systems a manual. A purposeful Information system shall help users to make good decisions and support their actions.

Information systems, like other products, are going through a life cycle. Such as Information Systems Development, Information Systems in Use (Operation), Information Systems Maintenance Management and Information Systems Withdrawal. An information system development is the process of defining, designing, testing, and implementing a new software application or program. The system developer uses different tools, techniques, procedure, method and philosophy to implement the information system development.

Blood is a saver of all existing lives in case of emergency needs. The task of blood bank is to receive blood from various donors, to monitor the blood groups database and to send the required blood during the need to the hospital in case of emergencies. The problem is not insufficient number of donors, but finding a willing donor at the right time. We want to build a network of people who can help each other during an emergency. This application timely

updates the information regarding the donors where the administrator accesses the whole information about blood bank management system. Donor will be prompted to enter an individual's details, like name, phone number, and blood group. In the urgent time of a blood requirement, you can quickly check for blood banks or hospitals matching a particular or related blood group and reach out to them through the App. Blood bank App provides list of blood banks in your area. A large number of blood donors are attracted using an Android application. Since almost everyone carries a mobile phone with him, it ensures instant location tracking and communication. Only a registered person, with willingness to donate blood, will be able to access the service. In this application we are using the GPS technology that will be used to trace the way to the blood bank. The user will get the route to reach the desired location and he won't have to ask manually, therefore time can be saved.

**CHAPTER 3**

**SYSTEM REQUIREMENTS AND SPECIFICATION**

**3.1 Introduction to SRS**

A Software Requirements Specification (SRS) is a complete description of the behaviour of the system that is to be developed. Use Case technique can be used in order to find functional requirements of the software product. SRS also includes non-functional (or supplementary) requirements. Non-functional requirements are requirements which impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints).

**3.2 Functional requirements**

* User should be able to register to website using name and his details.
* User should be able to search blood donors, blood banks and hospitals without registering to the application.
* Donor should be able to update his profile.
* User should be able search donors, hospitals and blood banks.
* Admin with his user name and password should be able to modify the details of available blood in blood bank.
* Admin should be able to modify the details of donor.

**3.3 Non Functional requirements**

These are requirements that are not functional in nature, that is, these are constraints within which the system must work.

* **Usability:** The system interface shall be implemented without much graphical design.
* **Security:** The PC should be highly secured and accessible only by the administrator to avoid the misuse of the application. During login process the interface should not leak images & the click points. The appropriate image click points are captured & encrypted & stored on database.
* **Maintainability:** The installation of appropriate application and operation manual will be provided to the user with video.
* **Availability:** The images are always available and displayed on time during password creation & login process without delay and system shall be available around the clock.
* **Portability:** The GUIs of this application shall be user-friendly so it will be very easy for the user to understand and to respond to the same. The system shall use appropriate tools & languages to implement this application so it could be easily deployed with different platform.
* **Speed:** The system should not consume more time during password creation, login process & password recovery operations.
* **Ease of use:** The system usage manual & training videos, manual help & online help facility are provided.
* **Reliability:** During password creation, login & password recovery process the system shall not fail or work incorrectly.
* **Size:** The images should not consume more space.
* **Robustness:** The percentage of login event should be less during login process.
* **Response Time:** The time taken by the system to complete a task given by the user is found to be very less.
* **Scalability:** The system can be extended to integrate the modification done in the present application to improve the quality of the product. This is meant for the future works that is to be done on the application.

**3.4 Hardware requirements**

Processor : Pentium based systems with a minimum of P4

RAM : 512MB (minimum)

Hard disk : 20GB (minimum)

**3.5 Software requirements**

Operating System : Windows XP & Android OS for mobile

Languages : Java/J2EE (JDBC, JSP), DBMS

Software tools : Java JDK kit

Web Technologies : Html, JavaScript, CSS

Web Server : Glassfish

Database : My-SQL

**CHAPTER 4**

**4.1 SYSTEM DESIGN**

System design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements. System design could be seen as the application of systems theory to product development.

**4.1.1 Overview**

System design is a transition phase from a user oriented documented system to a purely programmatic oriented system for programmers database personnel. The system design makes the high level decisions about the overall architecture of the system. The system design phase provides the understanding and procedure details necessary for implementing the system recommended study. The target system is arranged into subsystems based on the analysis structure and the proposed architecture.

**4.1.2 System architecture**

**Add newly collected blood group details, Add maintain all necessary details about organization.**

**Data Base**

j0285750

**Donor,Hospital,Blood bank can register the details regarding donating the blood & also view available donor details.**

**D:\mob.jpgj0285750**

**j0285750**

**D:\mob.jpg**

**They can get necessary details at any time about available blood in the bank and also available donor details**

**Fig 4.1.2 System achitechture**

**4.1.3 Modular description**

**Blood Bank**

It is the main entity for every organization who is responsible to maintain all the details of blood bank organization such as add new arrived blood details to organization, maintain all donated blood details, view all available blood and blood donor details etc.

**Donor**

Donor for blood bank organization is main entity without donor we are unable to provide services to public. Here donor job is to register their details in case of interest and he can also check available donor and blood details as and necessary.

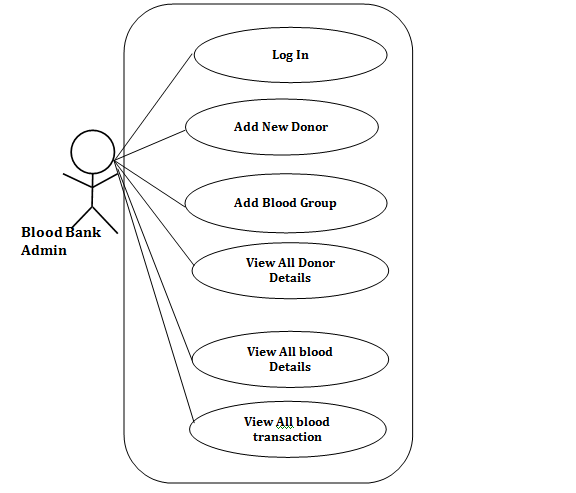
**Public**

Public are the end user to whom we are developing this application, public are those who need blood in emergencies, they can get details of available blood details in organization and they are allowed to search blood donor details in emergencies.

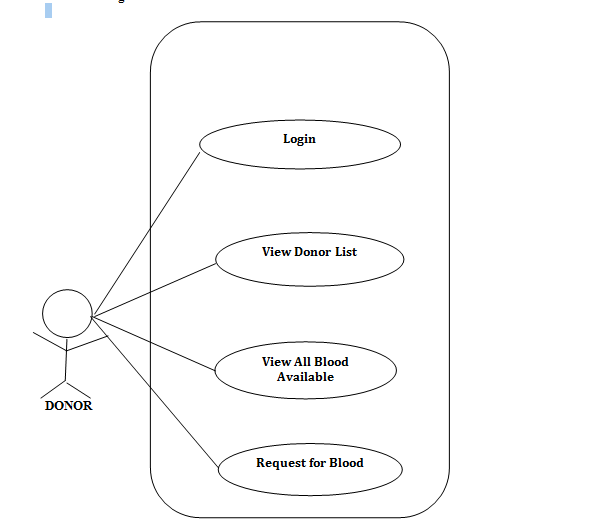
**4.2 Modular design**

**4.2.1 Use case diagram**

**Use case diagram for blood bank**

****

**Use Case Diagram for the Donor**

****

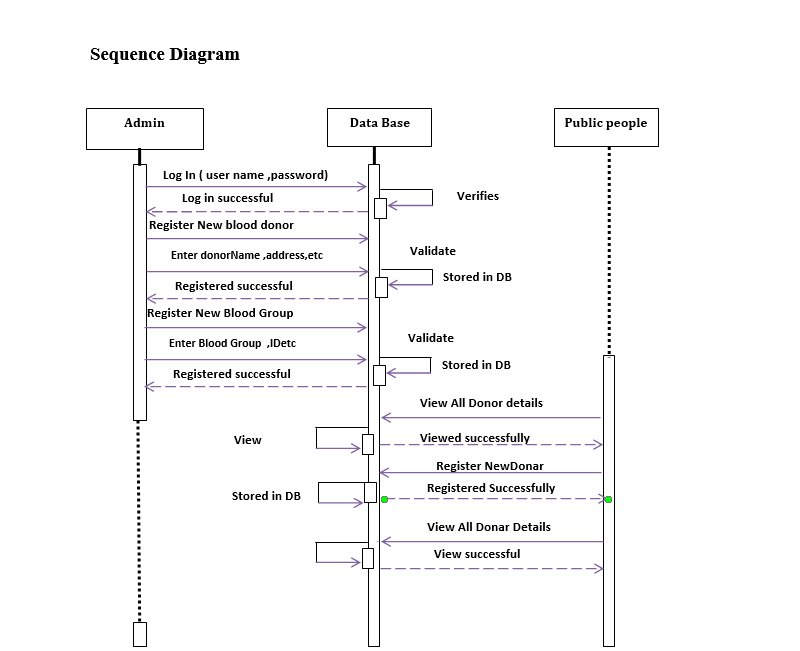
**4.2.2 Data flow diagram**

****

**Fig 4.2.2 Dataflow diagram**

Fig 4.2.2 shows the dataflow diagram of the application. When the user enters the application for the first time with proper internet connection he will be given options either to search the blood group without any registrations or he may be given options to register as a volunteer blood donor. After the successful registration the user will be directed to the activity where he will be provided with options like search donors, search blood in the nearest blood banks. A series of activities will be done when the user clicks on any of the displayed options. The database is used to search blood groups and donors. When the user double taps on back button the application will be stopped.

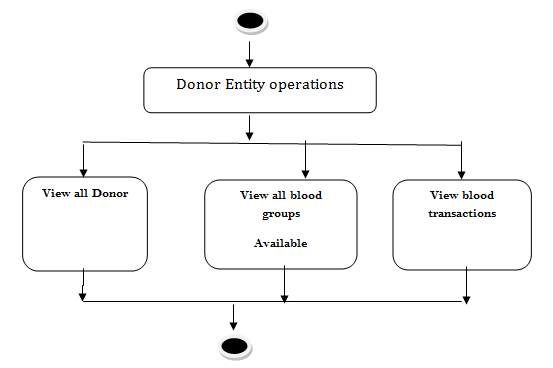
**4.2.3 Sequence diagram**

****

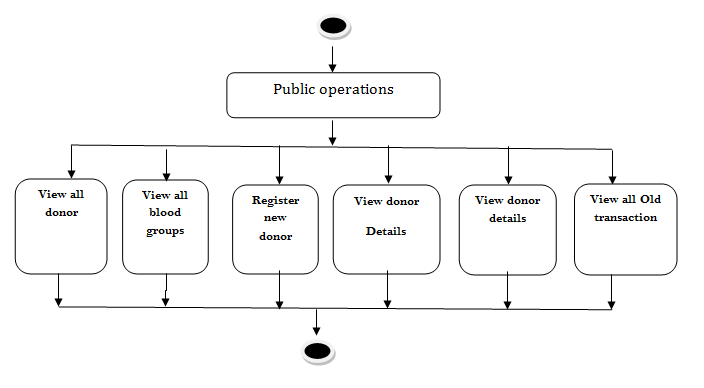
**Fig 4.2.3 sequence diagram for online blood bank**

**4.2.4 Activity diagram**

**Donor Activity**

****

**Public Activity**

****

**CHAPTER 5**

**SOFTWARE ENVIRONMENT**

**5. SOFTWARE TOOLS USED**

**5.1 MySQL**

MySQL is the most popular Open Source Relational SQL database management system. MySQL is one of the best RDBMS being used for developing web-based software applications.

* MySQL is becoming so popular because of many good reasons:
* MySQL is released under an open-source license. So you have nothing to pay to use it.
* MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
* MySQL uses a standard form of the well-known SQL data language.
* MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
* MySQL works very quickly and works well even with large data sets.
* MySQL is very friendly to PHP, the most appreciated language for web development.
* MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).

**5.2 NET BEANS:**

The Net Beans Platform allows applications to be developed from a set of modular software components called modules. A module is a Java archive file that contains Java classes written to interact with the Net Beans open APIs and a manifest file that identifies it as a module. Reason for Using Java is to explore systems running different operating system.In order to do so, there should be some way to connect to bridge those operating systems so that all the differences between them are solved and the functionalities are achieved***.***

**CHAPTER 6**

**PROGRAMMING TECHNIQUES**

The programming language chosen should reflect the necessities of the project to be completely expressed in terms of the analysis and design documents. As the project is an android application the java is the base for the project.

Java, an Object-Oriented language from Sun Microsystems with syntax similar to that of C, draws its roots from the C++ language. The Java programming language and some of its features that set it apart from other programming languages are:

* It is simple, object oriented and familiar.
* It is robust and secure.
* It has an architecture-neutral and portable environment.
* It executes with high performance.
* It is interpreted, threaded and is dynamic.

To enable a Java application to execute, the Java Compiler generates an architecture neutral object file format called the Java byte codes and this compiled code is executable on many processors, given the presence of the Java Runtime Environment.

**6.1 Cascading style sheets:**

CSS is an abbreviation for Cascading Style Sheets. CSS works with HTML and other Markup Languages (such as XHTML and XML) to control the way the content is presented. Cascading Style Sheets is a means to separate the appearance of a webpage from the content of a webpage. CSS is a recommendation of the World Wide Web Consortium (the W3C). The W3C is a consortium of web stakeholders: universities, companies such as Microsoft, Netscape and Macromedia, and experts in many web related fields. The presentation is specified by styles, which are presented in a style sheet. If you're familiar with word processing programs like Microsoft Word, you have probably played around at least a little bit with styles. For example, when you want to make the headline text of your document big and bold, the hard way to do it would be to select the text, select a font face and weight, and select the color. The easier way to do it (presuming your document has more than one headline) is to create a "rule", or style, for all the headlines in your document. Then all you have to do is to make one rule, and keep on applying that to all your headers. CSS in its most basic form works exactly like this. Instead of using <FONT> tags over and over again to control little sections of your page, you can establish some rules to apply globally, to a single page or all the pages on your site. CSS is a great time saver.

**6.2 Java Script:**

Javascript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

JavaScript was first known as **LiveScript**, but Netscape changed its name to JavaScript, possibly because of the excitement being generated by Java. JavaScript made its first appearance in Netscape 2.0 in 1995 with the name **LiveScript**. The general-purpose core of the language has been embedded in Netscape, Internet Explorer, and other web browsers.

**CHAPTER 7**

**IMPLEMENTATION DETAILS**

Implementation is the stage in the project where the theoretical design is turned into a working system and is giving confidence on the new system for the users, which it will work efficiently and effectively. It involves careful planning, investigation of the current System and its constraints on implementation, design of methods to achieve the change over, an evaluation, of change over methods. Apart from planning major task of preparing the implementation are education and training of users. The more complex system being implemented, the more involved will be the system analysis and the design effort required just for implementation.

An implementation co-ordination committee based on policies of individual organization has been appointed. The implementation process begins with preparing a plan for the implementation of the system. According to this plan, the activities are to be carried out, discussions made regarding the equipment and resources and the additional equipment has to be acquired to implement the new system.

Implementation is the final and important phase, the most critical stage in achieving a successful new system and in giving the users confidence. That the new system will work be effective .The system can be implemented only after through testing is done and if it found to working according to the specification. This method also offers the greatest security since the old system can take over if the errors are found or inability to handle certain type of transactions while using the new system.

**7.1 Pseudo code**

**Login form**

**<**Form name=”form1” method=”post” action=”login”>

<pre>

<input type=”text” name=”bname” placeholder=”username”/>

<select name=”bhu”>

<option value=”user”>User</option>

<option value=”h”>Hospital</option>

<option value=”b”>blood bank</option>

<option value=”a”>Admin</option>

</select>

<input type=”password” name=”pass” placeholder=”password”/>

<input type=”submit” value=”Login” onclick=”return check();”>

<a href=”forpass.jsp”>Forgot password</a>

</pre>

</FORM>

**7.2 Coding**

**Database CONNECTION**

public class MyDb {

Connection con;

public Connection getCon(){

try {

Class.forName("com.mysql.jdbc.Driver"); con=DriverManager.getConnection("jdbc:mysql://localhost:3306/bldgroup","root","nitin");

} catch (ClassNotFoundException ex) {

Logger.getLogger(MyDb.class.getName()).log(Level.SEVERE, null, ex);

System.out.println(ex);

} catch (SQLException ex) {

Logger.getLogger(MyDb.class.getName()).log(Level.SEVERE, null, ex);

System.out.println(ex);

}

return con; }}

**REGISTRATION FORM DESIGN**

<form name="form1" action="Ureg.jsp" method="post">

<label class="main">Donor Name: </label> <input type="text" placeholder="uername" name="dname" required>

<label class="main">Mobile : </label> <input type="text" placeholder="mobile" name="mobile" required>

<label class="main">Email: </label> <input type="email" placeholder="Email" name="email" required>

<label class="main">Gender:</label> <select style="width: 145px;" name="gender">

<option value="male">male</option>

<option value="female">female</option>

</select>

<label class="main">Age: </label> <input type="text" placeholder="age" name="age" required>

<label class="main">Blood Group: </label> <select style="width: 145px;" name="bld"> <option value="a+">a+</option> <option value="a-">a-</option>

<option value="b+">b+</option> <option value="b-">b-</option>

<option value="ab+">ab+</option> <option value="ab-">ab-</option>

<option value="o+">o+</option> <option value="o-">o-</option></select>

<label class="main">Location: </label> <input type="text" placeholder="location" name="location" required>

<label class="main">Password: </label> <input type="password" placeholder="password" name="password" required>

<label class="main">Hint question: </label> <select style="width: 145px;" name="qn"> <option value="1">fav car</option> <option value="2">nickname</option>

<option value="3">best friend</option> <option value="4">vechical number</option> <optionvalue="5">enemy</option> </select>

<label class="main">answer: </label> <input type="text" placeholder="answer" name="ans" required>

<input type="submit" onclick="return check();" value="Register"></form>

**SEARCH DONER FORM**

**<**form action=”searchudoner.jsp”>

<pre>

Blood Group <select style=”width: 145px;” name=”bld”>

<option value=”a+”>A+</option>

<option value=”a-“>A-</option>

<option value=”a-“>B-</option>

<option value=”b+”>B+</option>

<option value=”ab+”>AB+</option>

<option value=”ab-“>AB-</option>

<option value=”o+”>O+</option>

<option value=”o-“>O-</option>

</select>

Location <input type=”text” name=”location” placeholder=”location”/>

</form>

**VALIDATION**

<script>

Function check()

{

if(document.form1.bname.value==” ”)

{

alert(“Please Enter User Name “);

return false;

}

else if(document . forml . pass . value==”)

{

alert(“ please enter password ”);

return false;

}

else if(document.form1.bname.value==”blood” &&document.form1.pass . value==”project”)

{

alert (“valid”);

window.location=”adminpage.html”;

return true;

}}

</script>

**CHAPTER 8**

**TESTING**

**8.1 Overview**

Testing is a main part of any project development cycle. A project is incomplete without successful testing and implementation. A program or system design is perfect only when communication between the user and the designer is complete and clear. A successful system design helps in efficient testing and implementation. Testing is vital to the success of a system. This chapter concentrates on testing conducted on the project. Testing has helped in developing and improving the project.

During testing, the program to be tested is executed with a set of test cases and the output of the program for the test cases is evaluated to determine whether the program is performing as expected. Errors were found and corrected by using the following testing steps and correction will be recorded for future references.

Testing is a process of exercising software with an intent of finding errors.

**8.2 Types of testing**

**8.2.1 Unit testing:**

The application is also unit tested. Since this system represents a client server model while sending data if any of the mandatory fields are not filled then the application should alert.

If the Blood bank or Donor enters invalid username and password, the application must alert them.

Testing these things to uncover the errors and also connectivity between client and server test since the system is a client-server based system the server should be ready to serve the clients before client request any service if the connectivity is failed the application should alert the user.

**8.2.2 Integration testing:**

After the application is unit tested for each forms, then they are integrated and tested for their interactions Since in this project the sender sends data to the receiver both the modules have to be integrated and tested for interaction in between the data

passes through detector so the detector module is also integrated with both sender module and receiver module and vice versa and tested for integrity.

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

**8.2.3 System testing:**

The assembled application is tested for overall functionality and content delivery. Like conventional validations the application focuses on the user-visible action and user-recognizable output from the system. To assist in the derivation of validation tests, the tester should draw upon the test case. The use-case provides a scenario that has a high like-hood of uncovering errors in user interaction requirements.

**8.2.4 Functional testing**

Functional testing is a quality assurance process and a type of black box testing that bases its test cases on the specifications of the software component under test. Functions are tested by feeding them input and examining the output, and internal program structure is rarely considered (not like in white-box testing). Functional Testing usually describes what the system does.

Functional testing differs from system testing in that functional testing "verifies a program by checking it against design document(s) or specification(s)", while system testing "validate[s] a program by checking it against the published user or system requirements".

Functional testing typically involves five steps.

* The identification of functions that the software is expected to perform
* The creation of input data based on the function's specifications
* The determination of output based on the function's specifications
* The execution of the test case
* The comparison of actual and expected outputs

**8.2.5 Validation testing**

At the culmination of integration testing software is completely assembled as a package; interfacing errors have been covered and corrected, and a final series of software tests-validation testing may begin. Validation can be defined in many ways, but a simple definition is that validation succeeds when software functions in a manner that can be reasonably expected by the customer.

Reasonable expectation is defined in the SRS a document that describes all user-visible attributes of the software. Information contained in that section forms the basis for a validation testing approach.

**8.2.6 Software testing life cycle**

Although testing varies between organizations, there is a cycle to testing as shown in figure 8.2.6

* Requirements Analysis: Testing should begin in the requirements phase of the software development life cycle. The ambiguous, improper information should be avoided during the requirement analysis
* Test Planning: Test Strategy, Test Plan(s), Test Bed creation. All these elements are gathered to test our activities in the next step.
* During the test case development phase, testers work with developers in determining what aspects of a design are testable and under what parameter those tests work. In our project in this phase of testing we test all the activities involved in our application with the relevant test cases as shown in the Table
* Test Execution: We execute the software based on the plans and test cases and report any errors found. Once testing is completed, we generate metrics and make final reports on the test effort and whether or not the software tested is ready for release.
* Test cycle closure: Here we end up with the testing cycle and thus the software is ready to use.



**Fig 8.2.6 Software Testing Life Cycle.**

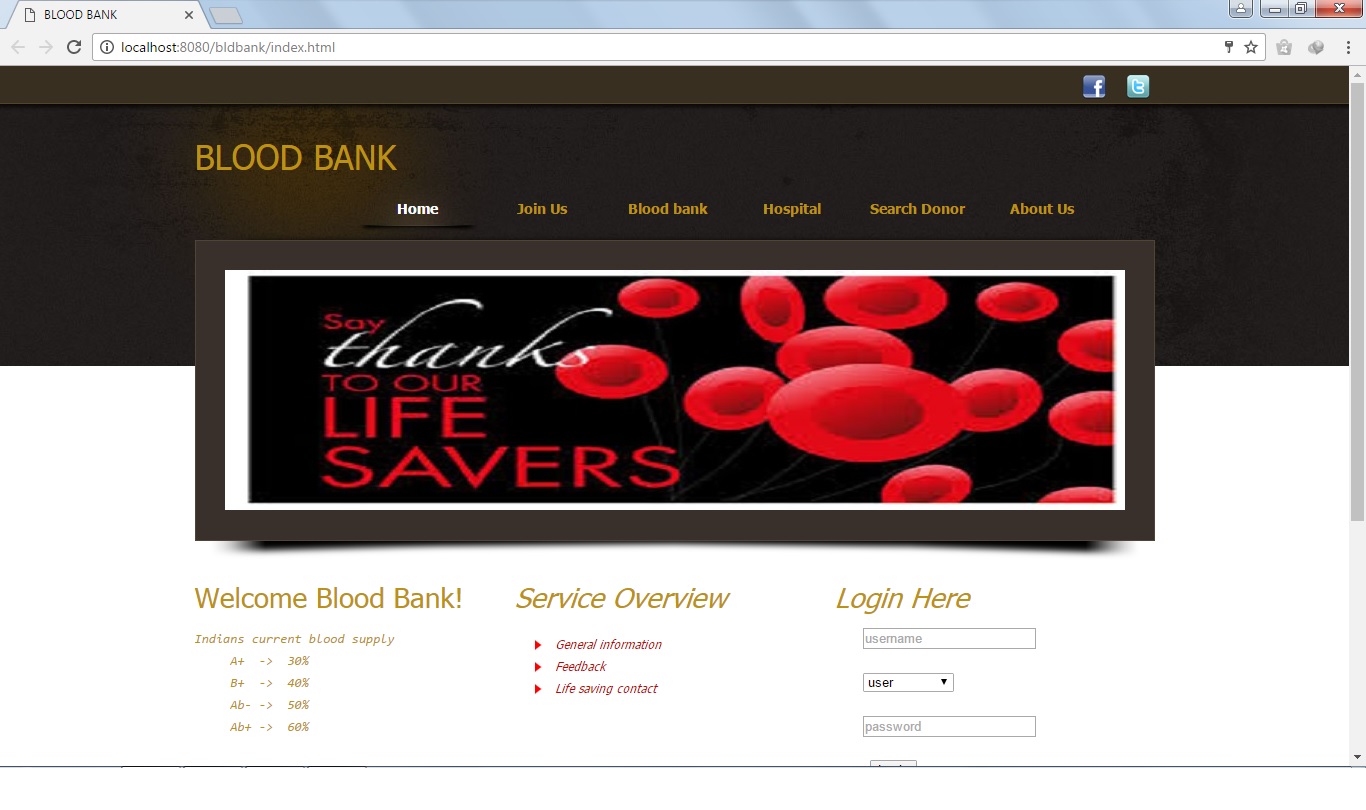
**8.3 TEST CASES**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Description** | **Expected Output** | **Actual output** | **Status** |
| 1 | Whenever Admin has to access the given options in the application. | He has to enter the pre-defined username and password to login and perform any action. | He has to enter the pre-defined username and password to login and perform any action. | Pass |
| 2 | Whenever Admin has entered invalid username and password | An alert message has to be displayed saying that invalid username and password | An alert message has to be displayed saying that invalid username and password | Pass |
| 3 | Whenever public wants to become volunteer blood donor to the application. | He must be provided with registration form where he has to enter all the necessary details | He must be provided with registration form where he has to enter all the necessary details | Pass |
| 4 | Whenever donor wants access to the options to which he has been provided. | Soon after the successful registration, donor will be provided with the unique ID. His name as the username and unique ID as the password he can login and access the options he has been provided with. | Soon after the successful registration, donor will be provided with the unique ID. His name as the username and unique ID as the password he can login and access the options he has been provided with. | Pass |
| 5 | Whenever Donor has entered invalid username and password | A web page with the message has to be displayed saying that invalid username and password | A web page with the message has to be displayed saying that invalid username and password | Pass |
| 6 | Whenever user enters the application without internet connection. | Application should not work. | Application should not work. | Pass |
| 7 | User clicks the search donor button on the main page. | The list of donors must be displayed with the contact info. | The list of donors must be displayed with the contact info. | Pass |
| 8 | Registered user clicks on the update profile button if he intends to do so. | A web page must appear where the user can enter his name, phone no., address, blood group etc., | A web page must appear where the user can enter his name, phone no., address, blood group etc., | Pass |
| 9 | User clicks on the search blood bank button in the main menu. | The nearest blood bank from the current location with their contact information will be provided. | The nearest blood bank from the current location with their contact information will be provided. | Pass |
| 10 | Whenever public/Donor/Admin clicks on blood count details. | They must be provided with the table which contains the blood groups and the number of units available for each blood group in the particular blood bank. | They must be provided with the table which contains the blood groups and the number of units available for each blood group in the particular blood bank | Pass |

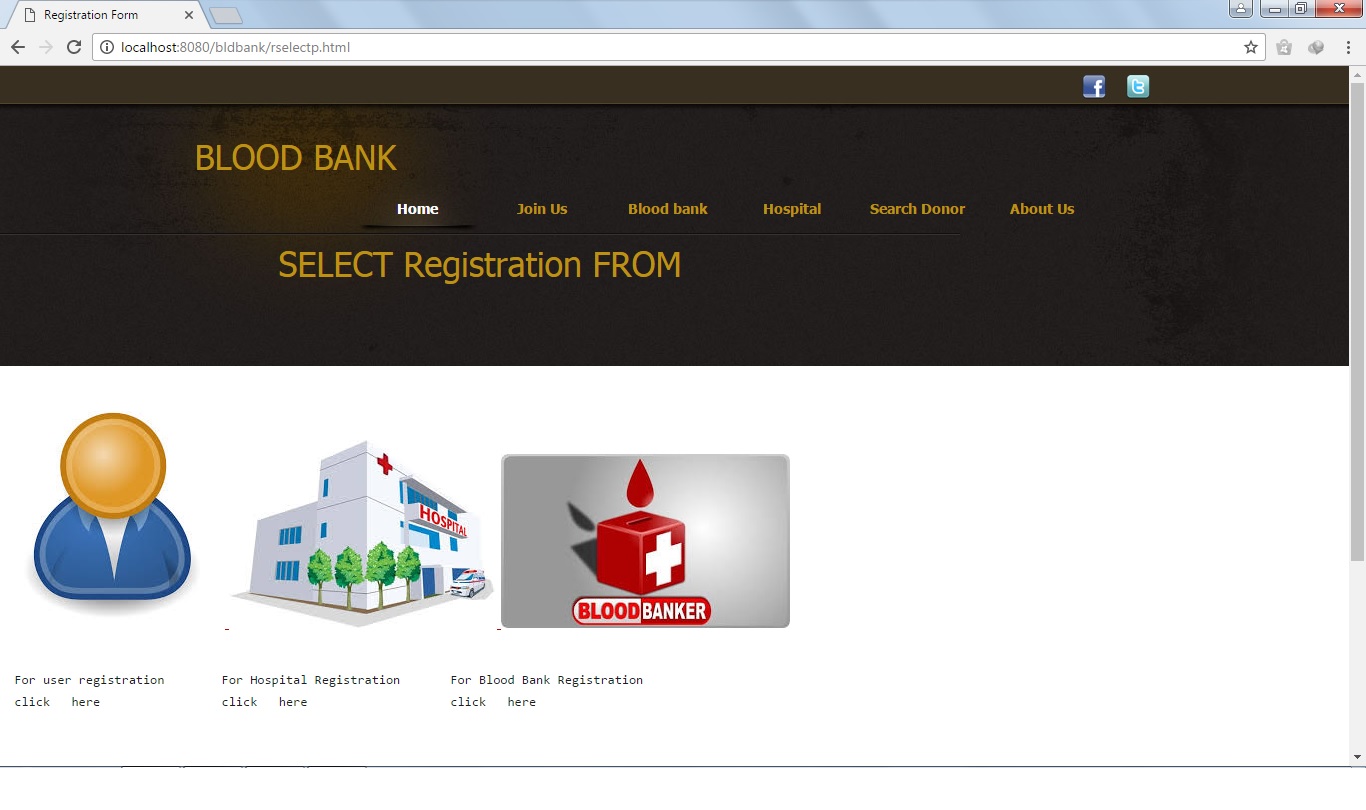
**CHAPTER 9**

**RESULTS & SNAPSHOTS**

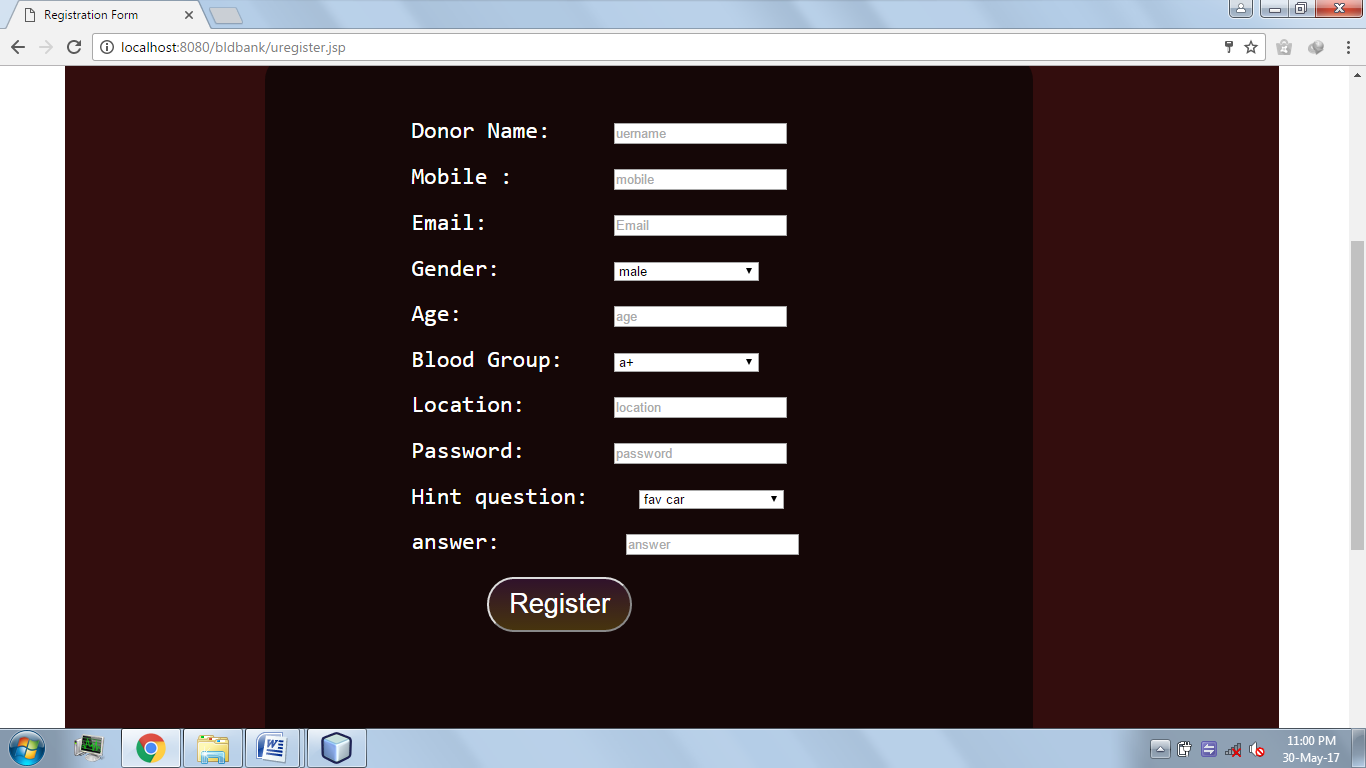
**9.1 Home page**

****

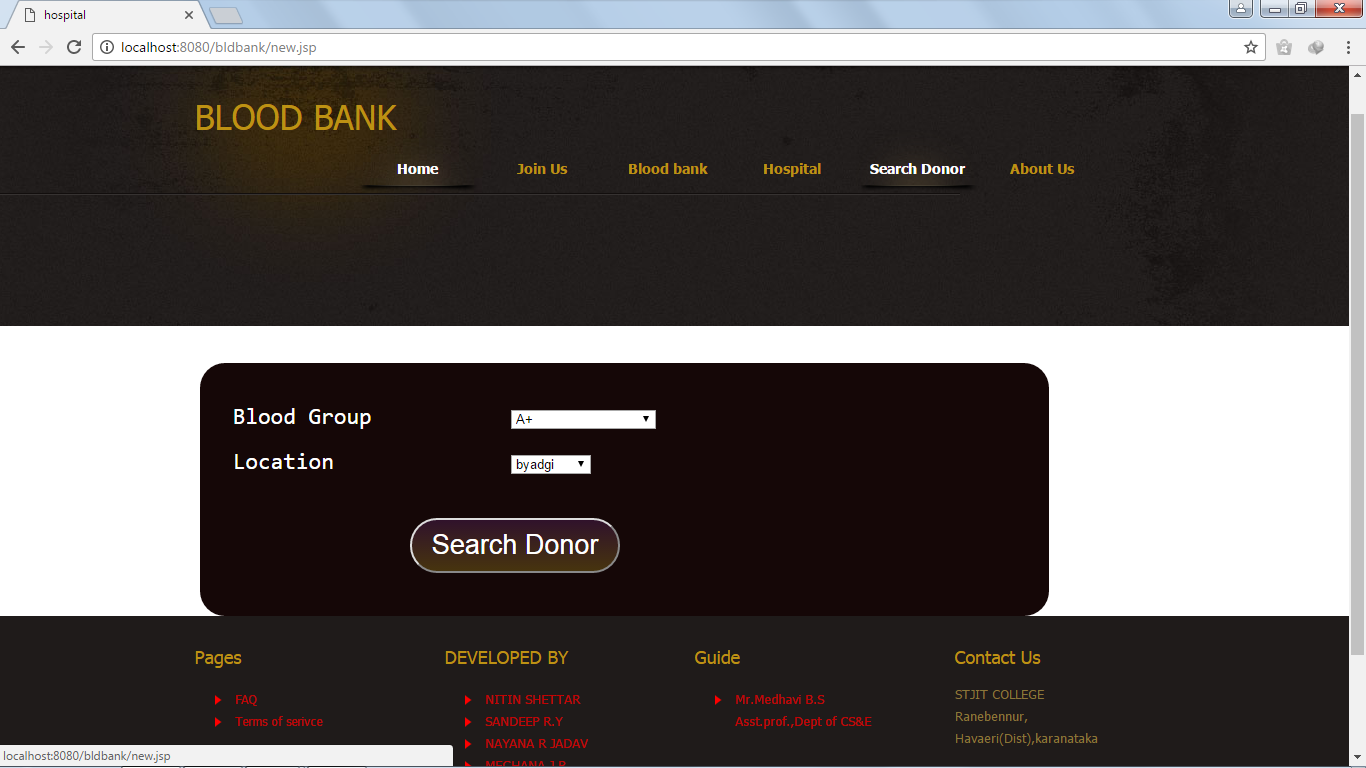
**Fig 9.1 Home page**

****

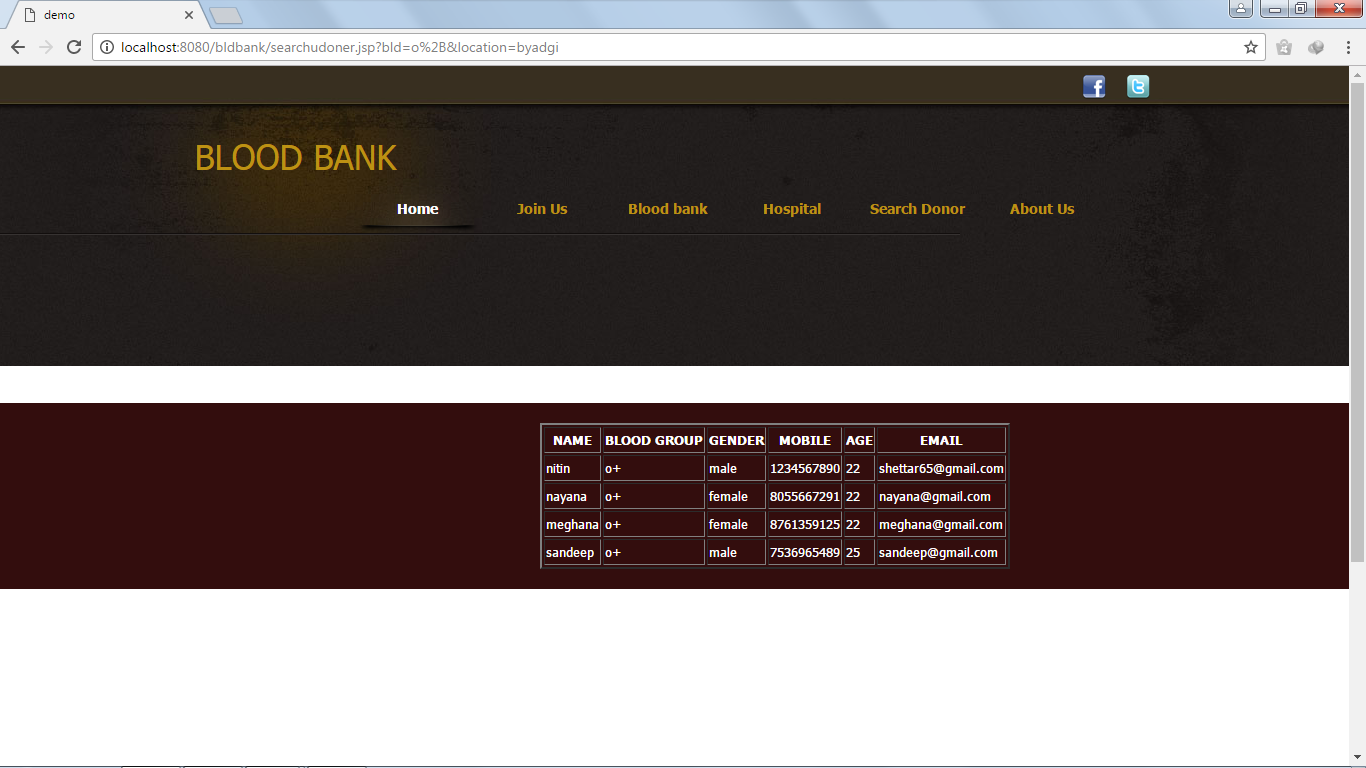
**Fig 9.2 Select Registration form**

****

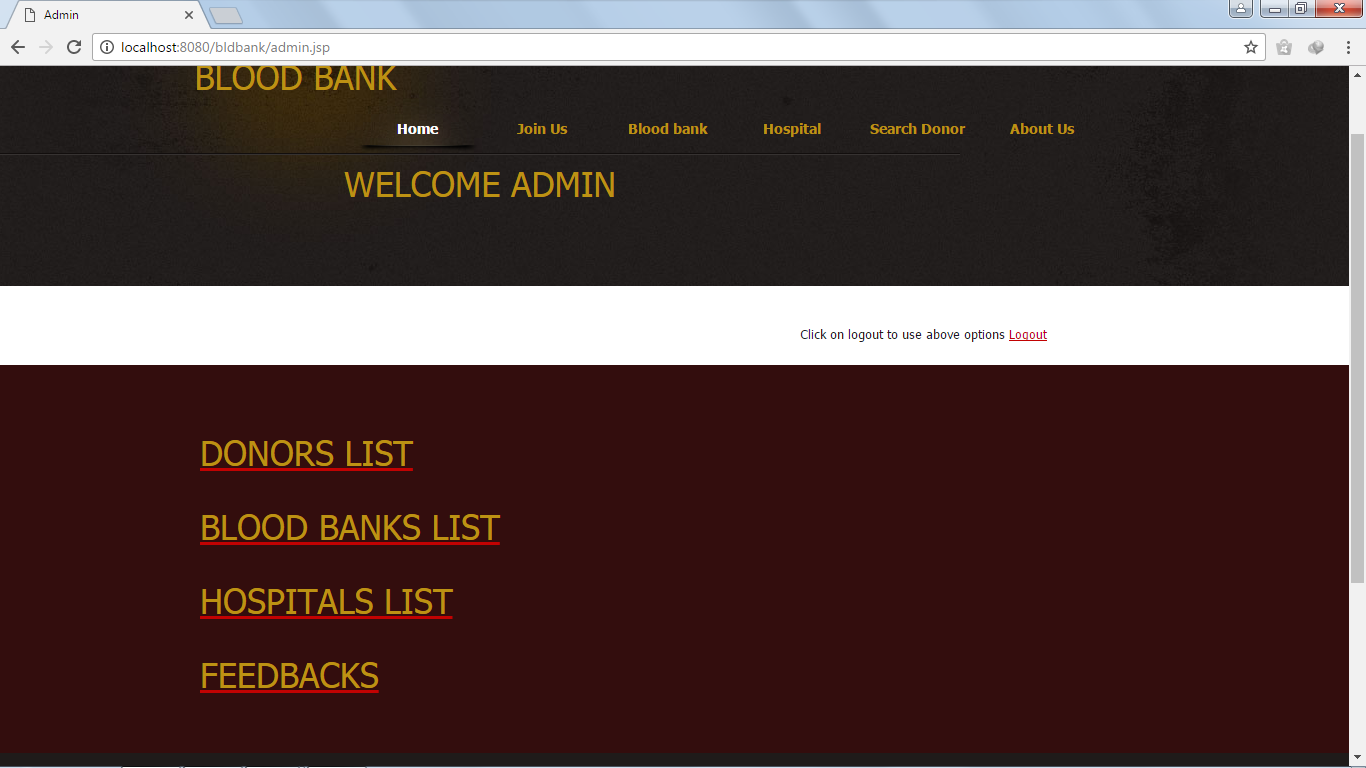
**Fig 9.3 Registration form for donor**

****

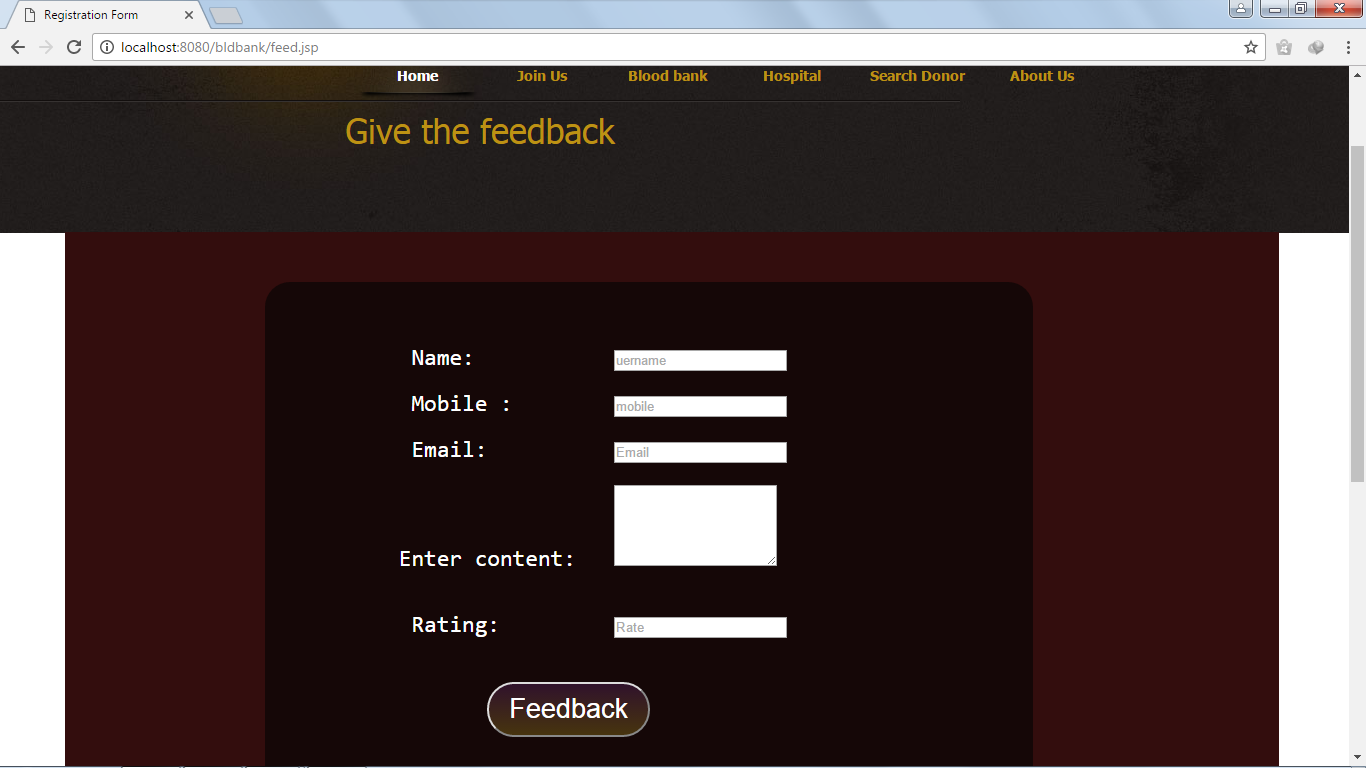
**Fig 9.4 Search donor**

****

**Fig 9.5 Search results**

****

**Fig 9.6 Admin page**

****

**Fig 9.7 Feedback form**

**CHAPTER 10**

**CONCLUSION**

This application helps many people who are in need of blood, can locate nearest blood banks by making use of this application provided the contact information and can get them by calling.

It saves many lives during emergency situations as it takes less time to locate blood units. So that one who is in need of blood group need not contact doctors or need not go to hospitals which is time consuming process.

Users can also send notification providing information like blood group, contact number, address and date specifying on or before the specified date. This helps many donors to know about donation of blood and they can donate blood before the specified date.