|  |
| --- |
|  |
| Blood Donor Management System |
| Synopsis |
|  |

**DIPANWITA DEY(105136520)**

|  |
| --- |
|  |

Table of Contents

[1. Title of the Project 1](#_Toc320841466)

2. Introduction and Objective

[2.1 Introduction 2](#_Toc320841468)

[2.2 Objective 3](#_Toc320841469)

3. Project Category……………………………………………………………………………………………………………………………………………

4. [Hardware and Software Requirement Specification 12](#_Toc320841487)

[4.1 Hardware Requirement 12](#_Toc320841488)

[4.2 Software Requirements 12](#_Toc320841489)

[5. REQUIREMENTS AND ANALYSIS 7](#_Toc320841478)

[5.1 Problem Definition 7](#_Toc320841479)

[5.2 Requirements Specification 7](#_Toc320841480)

[5.2.1 Functional Requirement 7](#_Toc320841481)

[5.2.2 Technical Specification 10](#_Toc320841482)

[5.3 Planning and Scheduling 11](#_Toc320841483)

[5.3.1 Gantt chart 11](#_Toc320841484)

[5.3.2 Tracking Gantt 11](#_Toc320841485)

[5.3.3 Pert chart (Network Diagram) 12](#_Toc320841486)

6. Scope of the Solution………………………………………………………………………………………………………………………………………

[7 CONCEPTUAL MODELS 14](#_Toc320841491)

[7.1 E-R Diagram 14](#_Toc320841492)

[7.2 Context Diagram 16](#_Toc320841493)

[7.3 Data Flow Diagram 17](#_Toc320841494)

8. Database and tables Detail…………………………………………………………………………………………………………………………….

9. Complete Structure…………………………………………………………………………………………………………………..

9.1 Module Description…………………………………………………………………………………….

9.2 Data Structure……………………………………………………………………………………………………………………………

9.3 Process Logic………………………………………………………………………………………………………………………

9.4 Implementation Methodology……………………………………………………………………………………….

9.5 List of Reports…………………………………………………………………

10. Overall Network Architecture…………………………………………………………………………………………………………..

11. Implementation Of Security Mechanisms at Various Levels……………………………………………………………………

12. Future Scope & Further Enhancement Of the Project………………………………………………………………………..

13. Bibliography……………………………………………………………………………………………………………………………

# 2. Introduction and Objective

## 2.1 Introduction

Now a days “Blood donation camp”, “Blood crisis” are very well-known term. We all know that the supply of blood is lower than the demand. As a result naturally a crisis arises. In the same time it is also true that blood bank dispose large amount of blood as because it is impossible to store or preserve blood after a certain period properly. To overcome this crisis we came up with a suggestion of having a parallel eco-system of blood bank - “Blood Donor Management System (**BDMS**)”. In this eco-system the willing blood donor will enroll with **BDMS** and whenever a patient will need blood, member of this community will go and donate blood. We will develop a software application to maintain the whole process.

## Objective



Blood Donor Management System (will be referred as **BDMS** in this document) will create a bridge between blood donor and patient which will cater critical blood requirements, hence save life. It is a software solution for managing Blood Donor in any organization. It is very inefficient to use paper books, ledgers or excel sheets, word documents to track and manage donor, patient’s requirements. Blood Donor Management System will enable users to maintain computerized records and manage patients’ needs more efficiently with help of sophisticated data management techniques and technologies. **BDMS** will have a web based client and mobile client also to enable update & query from anywhere.

# 3. Project Category

# 4. Hardware and Software Specification

## 4.1 Hardware Requirement

* **Disc capacity :** 10 MB of available hard disk space
* **RAM :** 1 GB (32 Bit) or 2 GB (64 Bit)
* **Processor :** 1.6GHz or faster
* DVD-ROM Drive / USB **Port**

## 4.2 Software Requirement

* Windows XP (x86) with Service Pack 3 / Windows Vista (x86 & x64) with

Service Pack 2 / Windows 7 (x86 & x64)

* Microsoft .NET 4.0

# 5. REQUIREMENTS AND ANALYSIS

## 5.1 Problem Definition

In our country blood crisis is a problem. In this situation we are trying to find out some alternative ways to solve this crisis. Blood Donor Management System is aneffort to alleviate this issue.

Generally it takes long time to give service to the needy patients in traditional system, to avoid this time consuming process we are coming up with **BDMS**. Computerized **BDMS** solution will be more efficient and it will save time by enabling search, querying the information faster. We can gather information through JAVA enabled mobile devices also and store into the main Server system. We can communicate with large number of people through the web client any time anywhere. Register and Update process will take less effort.

Storing blood in blood bank is expensive and sometimes blood is wasted due to lack of maintenance. **BDMS** will manage donors and patient such a way that we can save and utilize blood for saving life.

Blood gets expired after certain duration being properly stored also. Our **BDMS** will overcome this issue by managing the need and supply in an efficient manner.

## 5.2 Requirements Specification

### 5.2.1 Functional Requirement

#### 5.2.1.1 Add Donor

**Introduction:**

Register a new Donor.

**Input:**

Relevant Donor data like name, address, contact number, blood group, date of birth.

**Processing:**

Admin will enter the data in the **BDMS** and create a new Donor entry.

**Output:**

The **BDMS** will generate a donor I for future reference.

#### 5.2.1.2 Patient Requirement Registration for Blood

**Introduction:**

Patient can register for blood.

**Input:**

Patient will enter data like name, address, blood group, admitted hospital address, date of need.

**Processing:**

The **BDMS** will check for availability and create the registration confirmation.

**Output:**

The **BDMS** will generate a Case ID for future reference.

#### 5.2.1.3 Search donor

**Introduction:**

Anyone can search for blood.

**Input:**

He will enter data like Blood group, area, date of need.

**Processing:**

The **BDMS** will search for the requirement.

**Output:**

The **BDMS** will display the search result.

#### 5.2.1.4 Make a donation

**Introduction:**

People can donate money to encourage the volunteers and bear the expenses.

**Input:**

Admin will enter the name, amount, and address etc. data in the **BDMS**.

**Processing:**

The **BDMS** will create a new financial donor entry.

**Output:**

The **BDMS** will generate financial donation details.

#### 5.2.1.5 Create an Event

**Introduction:**

In**BDMS** blood donation events can be created.

**Input:**

Admin will enter the Event name, venue, requirementsand date etc. data in the **BDMS**.

**Processing:**

The **BDMS** will create a new event entry.

**Output:**

The **BDMS** will generate anupcoming event ID and details.

#### 5.2.1.6 Mobile data entry & query

**Introduction:**

**BDMS** data can be entered and queriedusing a mobile device.

**Input:**

User will enter the registration details, donor information in the mobile device.

**Processing:**

The device will stored the entered data and sync with Server while manual sync operation. While querying device will search its internal storage for the query and display the result.

**Output:**

The mobile device will display the search result.

#### 5.2.1.7 Web data entry & query

**Introduction:**

**BDMS** data can be entered and queriedusing a web interface.

**Input:**

Admin will new donor details, event details as well as search query.

**Processing:**

Web interface will store new entry in the Google doc storage and while searching it will search its internal storage. Web interface will sync with main server while manual sync.

**Output:**

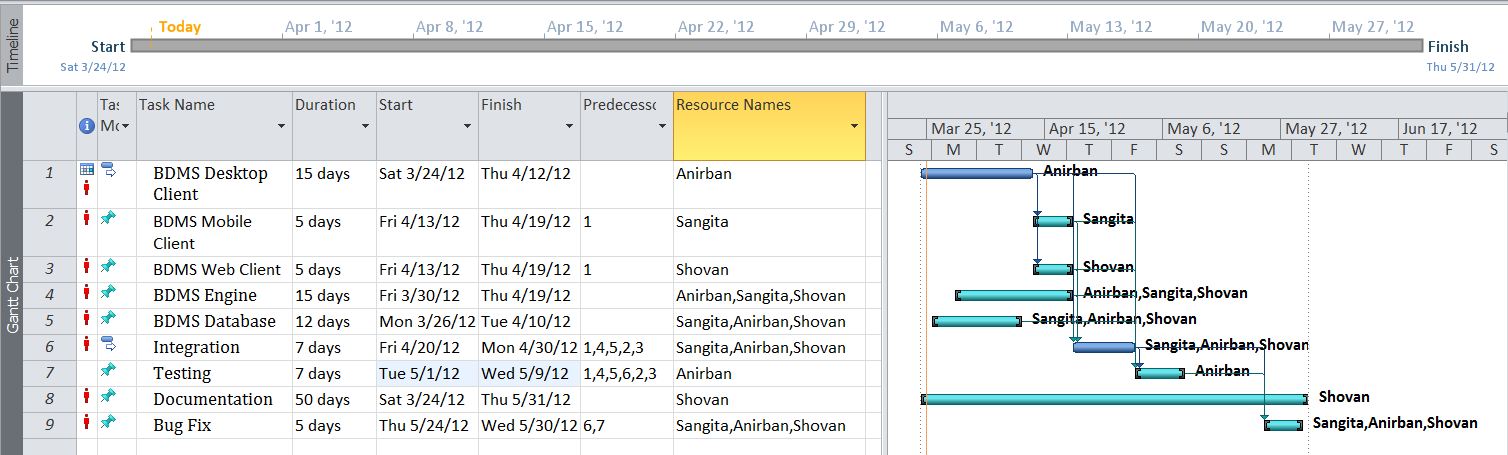
Website will show all the related information.

### 5.2.2 Technical Specification

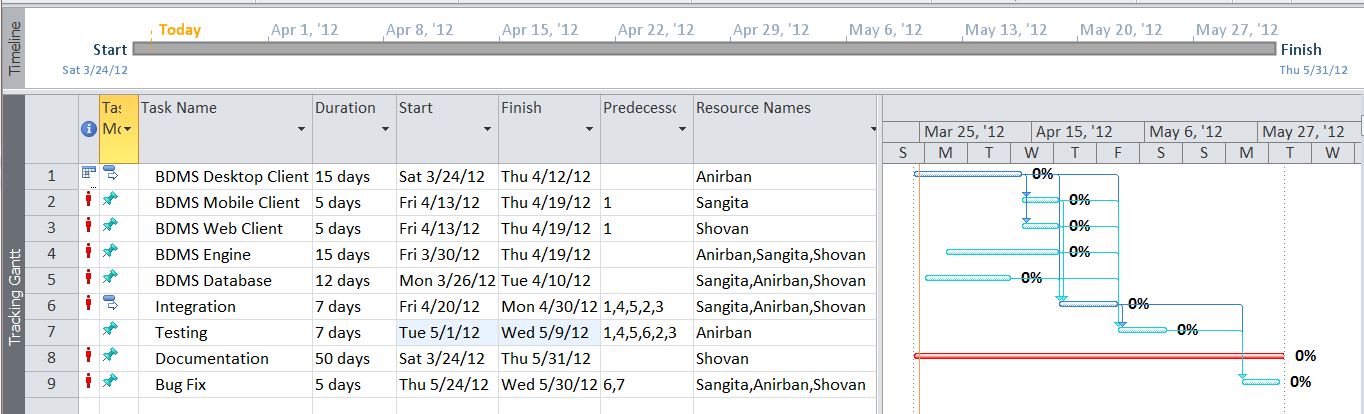
* The application will be **self-dependent** and no dependency on other parties required.
* There will be a digital **backup** and restore system.
* There will be more **opportunity** to extend the application in future.
* As the system contains a mobile connectivity then we can say the application is **portable**.
* The response time will be low and the system will **response** fast.
* It will be very **user friendly** and **usable** by any person with minimal computer knowledge.
* In terms of **security** unauthorized access will be denied and register user will be able to change as necessary.
* It will be **efficient** as it reduces manual labor and searching.
* **BDMS** will have user manual and help **documents**.
* It is designed such a way that it can be **maintained** with minimal effort.

## 5.3 Planning and Scheduling

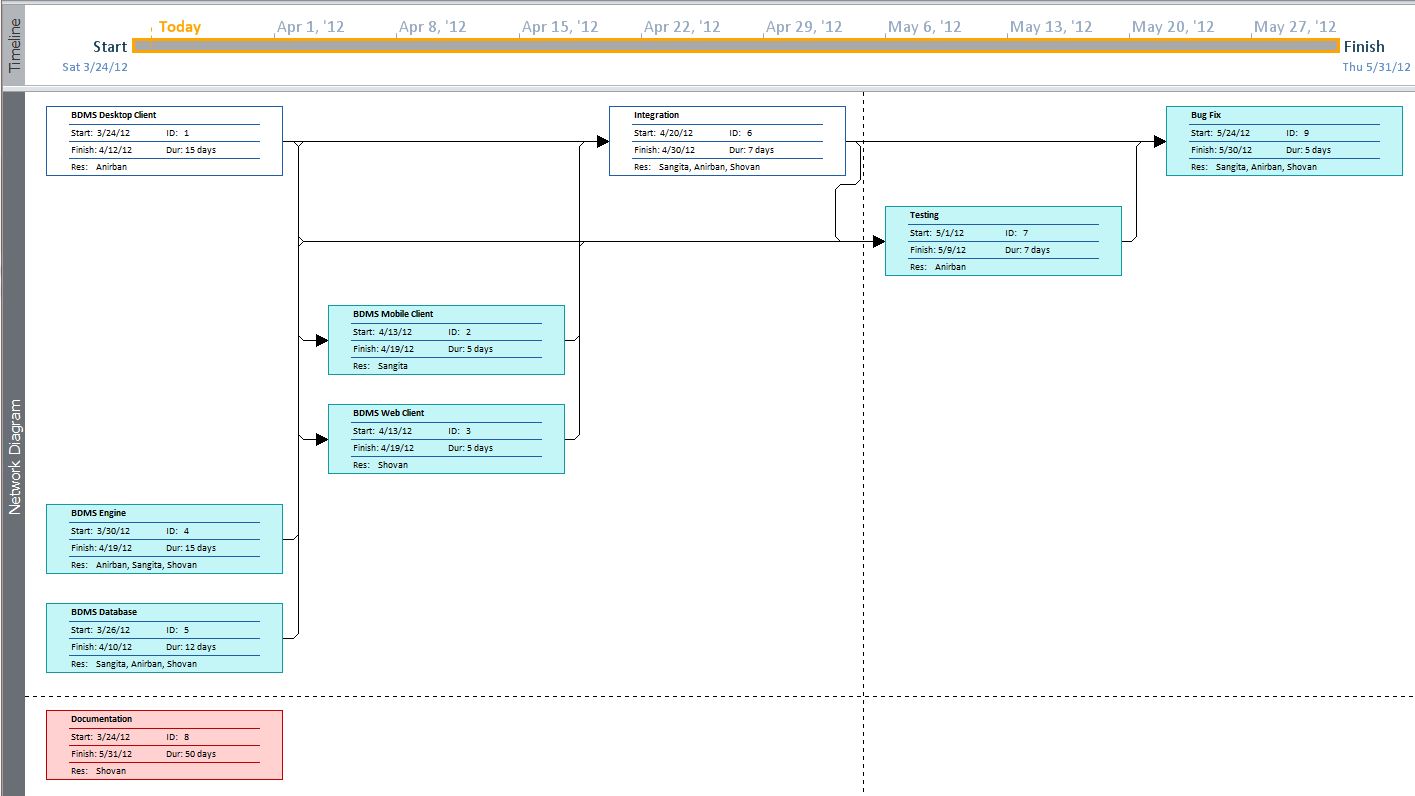
### 5.3.1 Gantt chart

****

### 5.3.2 Tracking Gantt

****

### 5.3.3 Pert Chart



# 6. Scope Of the Solution

Blood Donor Management Systemis not a substitute of blood bank or blood donation camps. This is a parallel eco-system along with other existing system to improve & utilize the blood donation process.

This application will be developed for Windows operating system (Win7, Win XP) only; in future we are planning to make it runnable under LINUX, MAC operating system also.

In case of mobile client we would develop it for java supported mobiles only; in future we would extend it to make it runnable under other mobile operating systems like Android, iOS or Windows Mobile OS.

Our web client will be developed using Google App Framework& Google Doc interface. Web client & Mobile client will not be synced automatically; it will require a manual sync with the server.

It is under a continuous process of development and we are working hard to make it perfect and error free project.

# 7. Conceptual Models

## 7.1 E-R Diagram

We will design a RDBMS for Blood Donor Management System. The entities and their attributes are listed below. Attributes in Bold letter is the unique key.

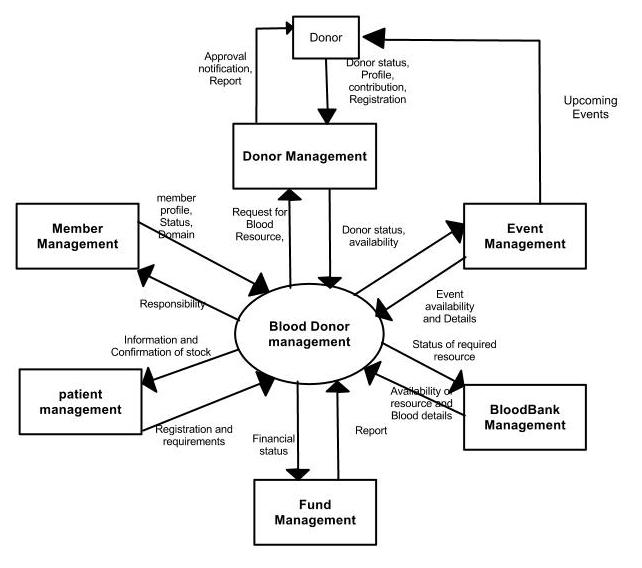
|  |  |
| --- | --- |
| **Entities** | **Attributes** |
| Donor | **donor Id**, Name, Address , Availability,Details |
| Blood Donor Management System | **ORG Id** , Name, Address, Registered no |
| Machine | **Machine Id**, Name, Software |
| Event | **event Id, place**, Time,patient\_id |
| Admin | **Admin Id**, Name, address, contact number |
| User Preference | **Preference Id,** Type, Description |
| Fund | Amount,dateOfdonation,Serial\_no |
| Patient | Patient\_id,name,contact no,photo id no,address, |

**Relationship between Entities:**

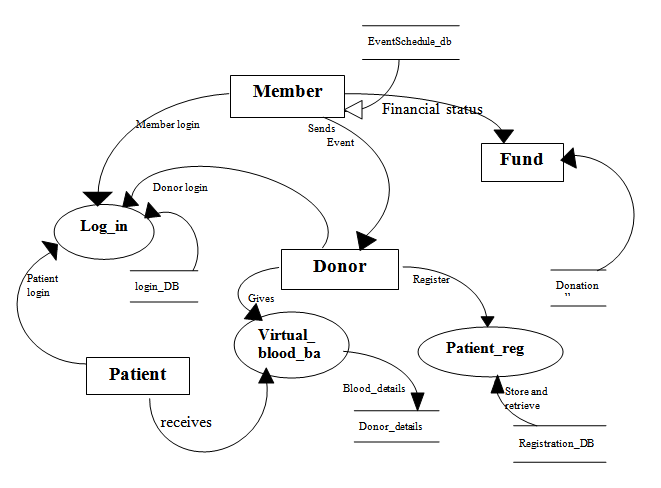
* Blood Donor Management System has donors  1 : N
* Blood Donor Management System has Machine 1 : N
* Users System uses Session 1 : 1
* Blood Donor Management System avails donation 1 : N
* Users provide Preferences  M : N
* Donor donates blood to patient→1:1
* Admin organizes events→1:1



## 7.2 Context Diagram



## 7.3 Data-Flow Diagram



# 8. Database & Table Details

# 9. Complete Structure

## 9.1 Module Description

## 9.2 Data Structure

## 9.3 Process Logic

## 9.4 Implementation Methodology

## 9.5 List of Reports

# 10. Overall Network Architecture

# 11. Implementation Of Security Mechanism At Various Levels

# 12. Future Scope & Further Enhancement Of The Project

# 13. Bibliography

* http://en.wikipedia.org
* http://msdn.microsoft.com/en-us/
* http://www.microsoft.com/en-us/default.aspx
* http://www.codeplex.com/
* http://stackoverflow.com/
* http://www.codeguru.com/
* http://www.w3schools.com
* www.mysql.org