# CHAPTER FOUR

## SYSTEM DESIGN

# 4.1 overview

Design is process of describing, organizing, and structuring system components at architectural design level and detailed design level. Design converts functional models from analysis into models that represent the solution. This project is designed in a manner that solves the problems of the organization by minimizing the work load of the existing system and employee. It provides more efficient, reliable and time saving system.

System design is the first part to get into the solution domain in a software development. This chapter focuses on transforming the analysis model into the design model that takes into account the non-functional requirements and constraints described in the problem statement and requirement analysis sections discussed earlier. This document describes the design issues of the overall system, such as design goal, subsystem decomposition, hardware/software mapping, and persistent data management. It provides the complete architectural overview of the proposed system. It is intended to capture and express the significant architectural decisions, which have been made, on the system.

In this chapter we are going to describe System design concepts with the respect to our system. System design is the process of translating alternative solution generated by analysis phase into detailed logical and physical system specifications.

#### 4.1.1. Purpose of the system design

The purpose of designing is to show the direction how the system is built and to obtain clear and enough information needed to drive the actual implementation of the system. It is based on understanding of the model the software built on. The objectives of design are to model the system with high quality. Implementing of high quality system depend on the nature of design created by the designer. If one wants to change to the system after it has been put in to operation depends on the quality of the system design. So if the system is design effetely, it will be easy to make changes to it.

Its main purpose is:-

Provides the link between the requirement elicitation, analysis and the implementation

Shows how the various components interact in the design.

#### 4.1.2. Design goal

The design goals are derived from non-functional requirements that mean non-functional requirement is the description of the feature characteristics and attribute of the system as well as any constraints that may limit the boundary of the proposed solution, and it describes the quality of the system that should be satisfied for the final product. The objectives of design are to model the system with high quality.

Design goals describe the qualities of the system that the developers should consider.

* + Performance
  + Dependability
  + Maintenance
  + Compatibility
  + Fault Tolerance
  + Throughput
  + End user
* **Performance**

In terms of performance, the existing/manual system is as satisfactory because it is slow/time consuming, energy consuming and does not support online information system about the awareness of blood donation, when to donate and how to donate. For this they use human power, and some group, this takes more time. The system should respond fast with high throughput, i.e. it should perform the task quickly possible as possible such as generating report and receiving, viewing project status and also employee and material information etc. The system performs its task within a user acceptable time and space. This includes the following:-

* **Response time**: - depending on the network connection that the user machine has the system is going to interact and respond to user’s request in a maximum of a second, if the user is just viewing the pages, but if the user’s request requires the processing of the database, like searching for schedule. Depending on the strength of available network the system should be response in short period of time.
* **Storage space**:-to do work efficiently the processor to be more than 2GB RAM and HD storage to be more than20MB.
* **Dependability**

The Company needs the system to be highly dependable as it is expected to be used by non-IT professions. The system should be robust and fault tolerant. Furthermore as the system is handling sensitive data of the blood bank, high emphasis should be given with regards to security, as there are subsystems to be accessed through web.

The citizenship identification system should achieve the following dependability characteristics in order to resist crash and be available and reliable.

* **Robustness:-**since the system is a web based system, that mainly use a menu driver entry there wouldn’t be an input problem by the user side. But for the server side there might be an error during the process of entering a data. At this time the system will provide an error pages and the system will continue without failure or crush.
* **Availability:-**as long as there is an internet connection the system will be available 7days a week and 24 hours a day.
* **Security:- -** by the user side there is low security issue to be considered, the only thingthat must be controlled is in the action of rating a movie that, they can rate or click the like or dislike button only one at a time. But by the official user side the system will provide a user name and password that will manage their own page according to their level of access.
* **Maintenance**

In time of failure or need modification the system need to maintain. To be maintainable the system should meet the following maintenance criteria.

* **Extensibility: -** if it is needed to add new functionality to the system, this must beachieved by only making a separate page and integrate this page with the existing system.
* **Modifiability: -** system should be modifiable for further modification and enhancement of the system.
* **Portability:-** the system is developed to be viewed and retrieved from any web browserregardless of their version and platform it resides in it.
* **Readability:-** the system code can be viewed by clicking on the current web page andchoose “view the source code**”** option.
* **Compatibility:-**The system should have to be compatible. Since this system is web based it is compatible with any operating system environment, if web browser installed and internet connection in the computer.
* **Efficiency**: - The system should perform its job efficiently according to its functional and nonfunctional requirements; i.e. it should be able to survive invalid user inputs, fault tolerant, reliable and available. Since the system stores private data, high emphasis is given to security to withstand malicious attacks.
* **Fault Tolerance-**The system should be able to give response (error message) when the user enter incorrect input. This recommends the user to enter correct input.
* **Throughput:-**Since BBMS has both desktop application and web application it is able to perform many tasks in fixed period of time. Different service center do different tasks in their working time without worrying the other service center are using the same system.
* **Backup and Recovery:**-We have used backup mechanisms such as removable flash disks, CDs and hard disks. Because the data might lose due to computer viruses or power fluctuation.
* **End user**

All the interfaces, forms and buttons are written or designed in a simple language or common language so that the user can access it without difficult. From the user point of view the system should provide the following end user criteria’s so that the system can achieve At least a90% usability by the user.

* **Utility: - in** order to help the user, to easily understand and interact with the system.
* **Usability:-** to enhance the usability of the system.

### 4.2 proposed System Architecture

In this project, the team uses a three-tier architecture, which has three layers. These three layers are the Application or Presentation layer, the business layer and the data access layer. Application or presentation layer is the form, which provides the user interface to either programmer or end user. The business layer is the class, which the team uses to write the function, which works as a mediator to transfer data from application layer or presentation layer to data layer. This layer also has a property layer which is a class where variables are declared corresponding to the fields of the database which can be required for the application and make the properties so that the team can get or set the data using these properties into the variables. The third tire is the data access layer which is also a class to get or set data to the database queries back and forth. This layer only interacts with the database. The database queries or stored procedures will be written here to access the data from the database or to perform any operation to the database.

#### Overview

The proposed system is mainly based on the SRS document that’s already developed and submitted. It mainly deals with subsystem decomposition – dividing the system in to manageable components. Another major task in system design deals with hardware/software mapping which deals with which components would be part in which hardware. Yet another aspect of system design deals with persistent data management, which illustrate as to how persistent data (file, database, etc.) are stored and managed and at last Access control will be presented. Our system is three tires architectural it has client side, server and database.

* **Client side**: here in the client side Blood seeker, Donor, worker, Doctor, Administrator, and user interface will be existing.
* **Server side:** here the web servers to connect the data base application are found; mean that theapplication logic to perform the application by the client is found.
* **Data base**: here the data bases that store the information are found.

The proposed system is expected to replace the existing system. The architecture provides the top level design view of a system and provides a basis for more detailed design work provide or reference a detailed description and diagrams of the architecture. The overall architecture

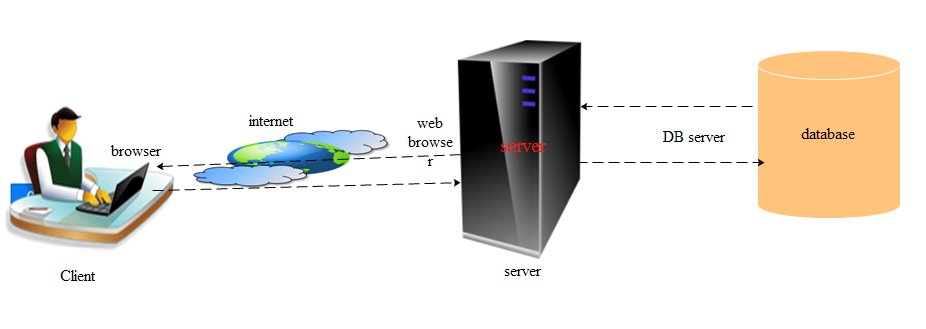


Figure 4.1 System architecture

Figure 4. 1 System architecture

* + 1. **System process**

System process can be perceived as series of well-defined steps. Though it varies according to design approach (object oriented), yet It may have the following steps involved:

* A solution design is created from requirement analysis
* Objects are identified and grouped into classes on behalf of similarity in attribute characteristics.
* Class hierarchy and relation among them are defined.
* Application framework is defined.

Other

PHP

Internet

Custom

HTML

Data access gateway

Xampp Application server

Xampp Application server

Clients

Services

Data management

DBMS

Enterprise

Local

# 4.2.2 Subsystem decomposition

Subsystem decompositions will help reduce the complexity of the system. The subsystems can be considered as packages holding related classes/objects. These subsystems are further decomposed into other subsystems. The major subsystems identified are “Registration”, “Login”, “Screening”, “Donate Blood”, “Blood Distribution”, “Blood Collection” “Refer Friend” and “Report” subsystems. Users are classified in to roles. The “Login” subsystem authenticates a user to grant access based on the role of the user.



# Hardware/ Software mapping

The system will have two processes, deployed in single or separate machine, that run in parallel, namely, web server process and the database process. The database process, which runs on MYSQL database engine, is responsible for maintaining data manipulation operations.

Where us the web server process is responsible to host the web pages of the system and process clients’ request. In case of the client side, only a browser is required to access the objects.



# 4.2.4 Component diagram

A component diagram depicts how [components](http://en.wikipedia.org/wiki/Component_(UML)) are wired together to form larger components and they are used to illustrate the structure of arbitrarily complex systems.



# 4.2.5 Deployment diagram

Deployment diagram depicts a static view of the run-time configuration of processing nodes and the components that run on those nodes. In other words, deployment diagrams show the hardware for your system, the software that is installed on that hardware, and the middleware used to connect the disparate machines to one another. You want to create a deployment diagram for applications that are deployed to several machines.



**4.2.9 Boundary condition**

Most of the system design effort is concerned with steady-state behavior. However, the system design phase must also address the initiation and finalization of the system. This is addressed by a set of new use cases called authentication use cases.

**Initialization**

Describes how the system is brought from a non-initialized state to state ("startup use cases”).Having internet connection, a platform and web browser are enough to initialize the system.

**Dynamic Model of the System Startup**

BBMS is a system that always must be on to run properly. Therefore, administrator must start the system when they wish to get information about BBMS After the system is started up, it is also necessary to steady connect the Internet, which must be installed on the machine to start view the information. For users to be able to access to the system from web and web server that will run the web App subsystem and web server that will run the web App must be started.

**Termination**

Describes what resources are cleaned up and which systems are notified upon termination If web server or web browser is not started to run then web interface will not be visible by the users. After finishing the system interaction user can be logging out and closes the web.

**Failure**

Many possible causes: internet connection fails, Bugs, errors, external problems (power supply) and also web application faller may happen.

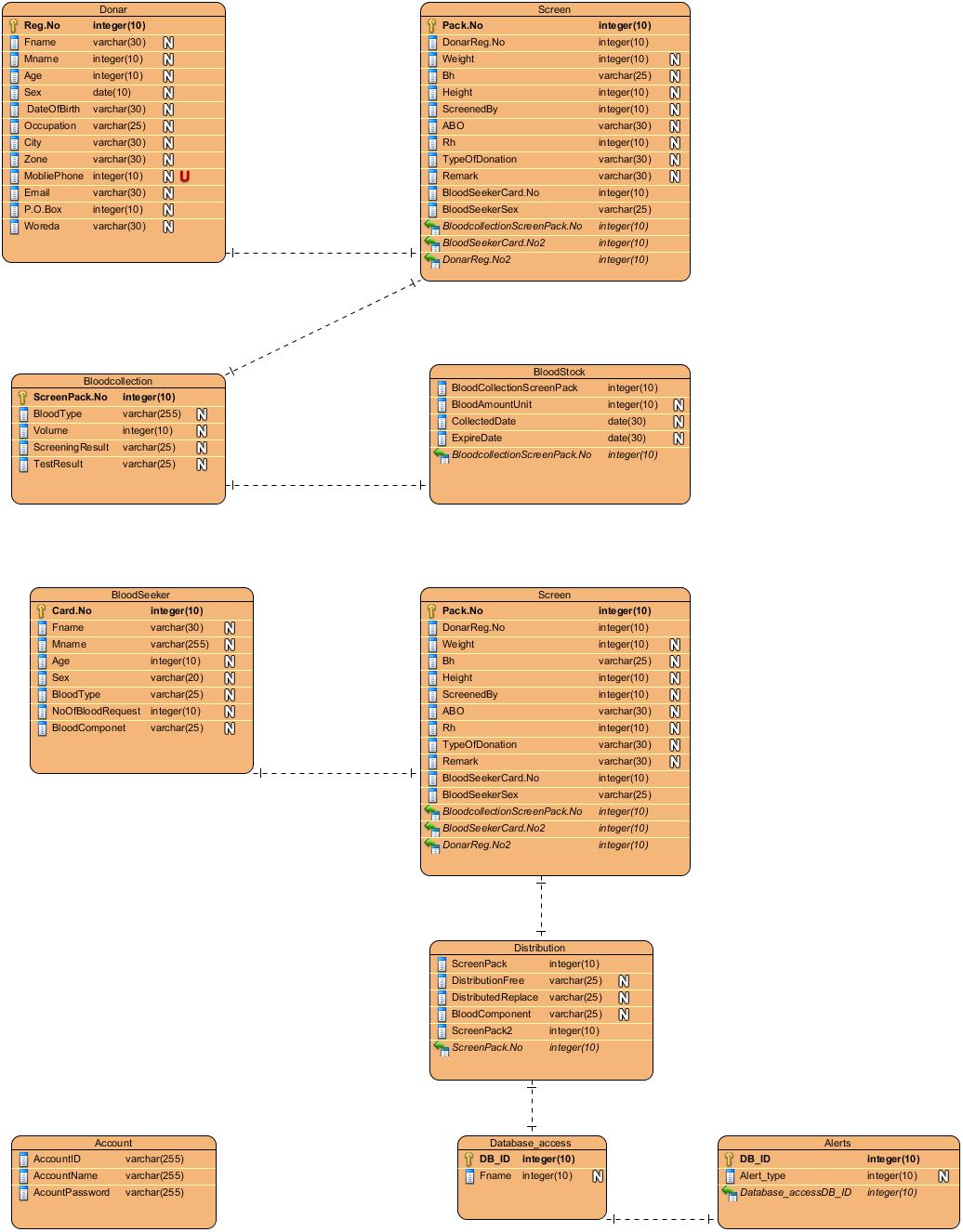
*Boundary condition failures*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Error | Cause |  | Forecast solution | | | |  |  |
|  | Fatal failure | eBay spyware | | Cleared | | the | cache | | and |
|  |  |  |  | cookies as suggested. | | | | |  |
|  |  |  |  | Browser repair (upgrade) or | | | | | |
|  |  |  |  | reinstalling. | | |  |  |  |
|  |  |  |  | cleared | | the | cache | | and |
|  |  |  |  | cookies as suggested; | | | | |  |
|  | Server Not found | Internet connection fail | | Network | | connection | | | or |
|  |  |  |  | firewall (proxy) protection. | | | | | |
|  | communication link fails | IP address or hostname in | | Verify and test them with | | | | | |
|  |  | JDBC URL is wrong. | | ping. |  |  |  |  |  |
|  |  | Hostname in JDBC URL | | Refresh DNS or use IP | | | | | |
|  |  | is not recognized by local | | address | | in | JDBC | | URL |
|  |  | DNS server. | | instead. | |  |  |  |  |
|  |  | Port number is missing or | | Verify it based on my.cnf of | | | | | |
|  |  | wrong in JDBC URL. | | SQL DB. | |  |  |  |  |
|  |  | DB server is down. | |  |  |  |  |  |  |
|  |  | DB server | doesn't accept |  |  |  |  |  |  |
|  |  | TCP/IP connections. | |  |  |  |  |  |  |
|  |  | Something | in between |  |  |  |  |  |  |
|  |  | ASP.Net | blocking |  |  |  |  |  |  |
|  |  | connections, e.g. a firewall | |  |  |  |  |  |  |
|  |  | or proxy |  |  |  |  |  |  |  |
|  | Web server fail | Computational/logic | | Using Server failover. | | | | |  |
|  |  | Errors |  | To | send | | a | backup | |
|  |  | Power outages | | automatically | | | on | Backup | |
|  |  | Overheating | | server. |  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | High humidity |  |  | That means | An | automatic |
|  | Natural disasters, | | e.g., | server failover solution can | | |
|  | hurricanes, Floods | |  | prevent your website from | | |
|  |  |  |  | going down in the event of a | | |
|  |  |  |  | server failure. | |  |
| Web Application fail | web application attacks | | | Firewalls | and | similar |
|  | (treat) | cross-site | | intrusion |  | detection |
|  | scripting(XSS) | and | SQL | mechanisms | provide little | |
|  | injection |  |  | defense against full-scale | | |
|  | Device driver failures | |  | web attacks. |  |  |
|  | I/O errors, e.g., hard disk | | | Using Server failover. | | |
|  | failures(see | database | |  |  |  |
|  | media failures) |  |  |  |  |  |
|  | Memory parity errors | |  |  |  |  |
|  | Network hardware failures | | |  |  |  |

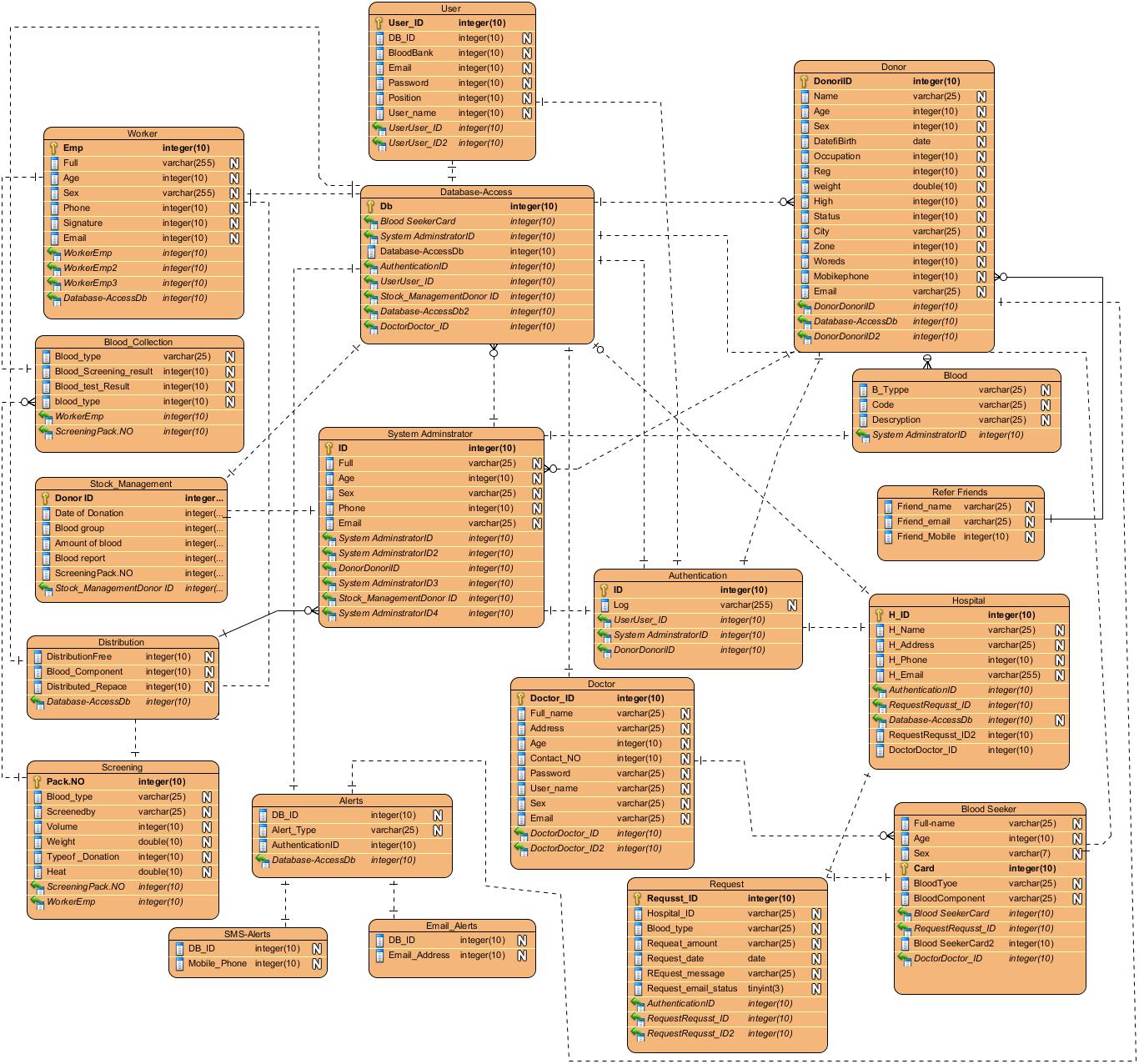
# 4.2.6 Persistent Data management

Persistence modeling is used to communicate the design of the database, usually the data base to both the users and the developers. It is also used to describe the persistence data aspect of the system. The following diagram indicates the persistence diagram of the system.



# 4.2.10 Database design

Database design is one of the basic part of system design. Database design can briefly show as what kind of information should have to be register in the system and in what table and it have the power to show relationship among deferent tables



# 4.2.8 Access controlling

In multiuser systems, different actors have access to different functionality and data. We  
modeled these distinctions by associating different use casesto different actors.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NO** | **Class** | **Attributes** | **Operations** | **Description** |
| 1 | **Admin** | EID,  Full Name, username, password address, Age,  Sex, phone,  Email,  qualification, | Login()  Logout()  Generate report()  Change account info()  Register user() | Gives a brief description about system administrator, and shows its main functions. |
| 2 | **Doctor** | dob,  Name,  Address,  Contact no  Sex,  username,  password,  Email  position, | Login()  Logout()  Check donor() | Stores record of Patient found in the blood bank.  Advertize by sending  Message to friends |
| 3 | **Donor** | Full Name, Date of birth, Age, , sex, Occupation, Reg.No,  City,  Zone,  Woreda, Mobile phone, email, | Login()  Logout()  Register for donation()  Donate() | Register for donation,  View blood bank info detail() |
| 4 | **Blood Seeker** | Name  Card NO, Ward, Recipient blood type, Amount of blood requested,  Name of physician requested,  Signature,  Date, | Search for()  Request for()  Blood seeker register to get blood() | He /she, who search for blood donor, find donors at emergency zone, request or report to admin panel. |
| 5 | **Worker** | username  password | Worker screen the blood()  Register-donor()  send notification()  Worker test the blood()  Worker could Discard() | He/she use the system from registration to medication. |
| **6** | **Lab Technician** | Full-name,  ID  email,  mobile number,  Address,  Age,  Sex | Record Donor medication()  View result() | Show His/her medical result. |