Computing Project

On

Blood Bank and Donor Management System



**Sabin Gautam**

00175013

Computing Project

Level 5 in Computing

Softwarica College of IT and E-Commerce

Kathmandu, Nepal

July 12th 2019

**Abstract**

Blood donor and donation management system is web based application which is used to search the blood donor information in case of required blood. This application is based on the PHP laravel framework. This system record all information of donor and user can search the donor info at any time. This is also helpful to those person who required blood at emergency case. Blood donor and donation hires some of the multi skills person based on their skills and capability to solve problems.

In this application there is two interface that is one for user and another other for admin. For user this application easy to use. This application does not contain any complex link that makes users feel too used. In user interface at first there will homepage in where all details of this page will occur. In user interface there is no any login system. User can register their information to donate blood. And if user have any question to ask then there is query form that can be used to ask questions.

In Blood donor and donation management system I have used waterfall methodology which helps us to manage this project easily and before end of date as mentioned in Gantt chart. To build Blood donor and donation management system I have used PHP programming language and in the framework I have used Laravel Framework. Laravel Frameworks is PHP framework which used MVC design pattern. So to develop this application I have used MVC pattern and in future other developer also can easily maintain this application. As we cannot complete whole project in given time so left work has been deferred for future work. In future admin can hide donor info and there will be function of read or pending function send queries.

Acknowledgement

In the preparation of this project, I had to take some help from the tutorials and guidance from different sites.  As the completion of this project gave me much aspiration, I would like to show my gratitude to Softwarica college of It and e-commerce for giving me a good advices for project throughout several sessions. I would also like to expand my gratitude to all those who have directly and indirectly guided me to complete this project.

To complete this project many people has supported me, especially my classmates have made suggested me valuable comments on my project and inspired me to improve the quality of the project.

Contents

[Chapter 1: INTRODUCTION 8](#_Toc13856317)

[1.1 Project Introduction 8](#_Toc13856318)

[1.2 Background of the project 8](#_Toc13856319)

[1.3 Overview of project 8](#_Toc13856320)

[1.4 Aims 8](#_Toc13856321)

[1.5 Objectives 9](#_Toc13856322)

[Chapter 2: Analysis 9](#_Toc13856323)

[2.1 Introduction 9](#_Toc13856324)

[2.2 Feasibility Study 10](#_Toc13856325)

[2.3 Requirement Analysis 11](#_Toc13856326)

[2.3.1 Functional Analysis 11](#_Toc13856327)

[2.3.2 Non Functional Analysis 13](#_Toc13856328)

[2.3.3 Moscow Prioritization 14](#_Toc13856329)

[2.3.4 SRS 15](#_Toc13856330)

[2.4 Use Case 16](#_Toc13856331)

[2.5 Class Diagram 17](#_Toc13856332)

[Conclusion 18](#_Toc13856333)

[Chapter 3: Design 19](#_Toc13856334)

[3.1 Structural design 19](#_Toc13856335)

[3.1.1 Class Diagram: 19](#_Toc13856336)

[3.1.2 Dataflow Diagram (DFD): 20](#_Toc13856337)

[3.2 Behavioral design 24](#_Toc13856338)

[3.2.1 Activity Diagram 24](#_Toc13856339)

[3.2.2 Sequence Diagram: 27](#_Toc13856340)

[3.3 Database design 29](#_Toc13856341)

[3.3.1 Data Dictionary: 29](#_Toc13856342)

[3.3.2 ER Diagram: 31](#_Toc13856343)

[3.4 System Architecture 34](#_Toc13856344)

[3.5 User Interface 35](#_Toc13856345)

[3.5.1 Prototype: 35](#_Toc13856346)

[Conclusion 39](#_Toc13856347)

[Chapter 4: Testing 39](#_Toc13856348)

[4.1 Introduction 39](#_Toc13856349)

[4.2 Importance of testing 39](#_Toc13856350)

[4.3 Types of testing 39](#_Toc13856351)

[4.3.1 Black box testing: 40](#_Toc13856352)

[4.3.2 Unit Testing: 51](#_Toc13856353)

[Chapter 6: Other Project issues 61](#_Toc13856354)

[6.1 Risk Management 61](#_Toc13856355)

[6.2 Configuration Management 63](#_Toc13856356)

[6.3 Project Issues 63](#_Toc13856357)

[6.4 Limitation 64](#_Toc13856358)

[6.5 Future Works 64](#_Toc13856359)

[Conclusion 64](#_Toc13856360)

Table of Figure

[Figure 1: Use Case 16](#_Toc13856361)

[Figure 2: class diagram 18](#_Toc13856362)

[Figure 1: Inserting donor Information 40](#_Toc13856363)

[Figure 2: database after inserting donor information 41](#_Toc13856364)

[Figure 3: Searching for result 41](#_Toc13856365)

[Figure 4: Result found 42](#_Toc13856366)

[Figure 5: Entering input searching for result 43](#_Toc13856367)

[Figure 6: Result not found 43](#_Toc13856368)

[Figure 7: Entering Input for sending Message 44](#_Toc13856369)

[Figure 8: database of send message 44](#_Toc13856370)

[Figure 9: Donor information before delete 45](#_Toc13856371)

[Figure 10: Donor information after delete 45](#_Toc13856372)

[Figure 11: database record after delete 46](#_Toc13856373)

[Figure 12: Deleting blood group Record 46](#_Toc13856374)

[Figure 13: database of record blood group after record 46](#_Toc13856375)

[Figure 14: deleting send message 47](#_Toc13856376)

[Figure 15: database of deleted message 47](#_Toc13856377)

[Figure 16: entering input to update contact info 48](#_Toc13856378)

[Figure 17: displaying update information 48](#_Toc13856379)

[Figure 18: database contact information for user 48](#_Toc13856380)

[Figure 19: Updating about us page information 49](#_Toc13856381)

[Figure 20: Displaying information after update 49](#_Toc13856382)

[Figure 21: database of Updated information in page 50](#_Toc13856383)

[Figure 22: Layout for changing password 50](#_Toc13856384)

[Figure 23: Redirect to login after password change 51](#_Toc13856385)

[Figure 24: database after password change 51](#_Toc13856386)

[Figure 25: Testing login for admin 52](#_Toc13856387)

[Figure 26: Result of admin login test case 52](#_Toc13856388)

[Figure 27: Registering donor info 53](#_Toc13856389)

[Figure 28: result of register donor after testing 54](#_Toc13856390)

[Figure 29: unit testing for deleting blood group record 54](#_Toc13856391)

[Figure 30: result after deleting blood group record in unit testing 55](#_Toc13856392)

[Figure 31: unit testing for deleting donor info 56](#_Toc13856393)

[Figure 32: result after deleting donor info record in unit testing 56](#_Toc13856394)

[Figure 33: displaying recorded donor information 57](#_Toc13856395)

[Figure 34: unit testing result to show recorded data of donor 57](#_Toc13856396)

[Figure 35: Unit testing to show recorded blood group 57](#_Toc13856397)

[Figure 36: Result of unit testing for blood group 58](#_Toc13856398)

[Figure 37: Unit test for sending message 58](#_Toc13856399)

[Figure 38: results of unit testing for sending message 59](#_Toc13856400)

[Figure 39: Unit testing for become a donor page link 59](#_Toc13856401)

[Figure 40: Result of unit testing after clicking on become a donor 59](#_Toc13856402)

[Figure 41: Unit testing for Search page link 60](#_Toc13856403)

[Figure 42: Result of unit testing after clicking on Search page 60](#_Toc13856404)

[Figure 43: Unit testing for about us page link 61](#_Toc13856405)

[Figure 44: Result of unit testing after clicking on About us page 61](#_Toc13856406)

# Chapter 1: INTRODUCTION

## 1.1 Project Introduction

Today world, people wants very quick services. People does not wants to wait or run here and there for services. On the developing the new technology, people wants quicks and fast services for application so that they do not wants to be panic for the services.. In this project people will not have to be panic for the blood. They can’t get easily their number of donor and also they can fill up the form to donate their blood so that at time of emergency people can easily get blood from another people.

### 1.2 Background of the project

Blood bank and donor management system is web application in where people can easily donate blood through form and also they can get services of searching required blood.

In this application user can get all the information of the blood donor and different blood groups available in each blood bank. This project aim is to provide transparency in this areas. This project also helps user to get blood from blood banks without any cost and hassle free.

This application is based on the user perspective which is user friendly. Using this application user can donate blood through this application and can get their needed blood at any time they want. This application is based on PHP, MySQL, HTML, CSS, and JavaScript and along with Laravel framework.

## 1.3 Overview of project

This project gives to record the detail of donor. This system provide recording system of blood donor management system. This is the automated system to record data of blood donor. This system provide the searching facilities of donor details so that user may contact to them to request for blood in case of emergency.

## 1.4 Aims

* The aims of this blood bank management system is to provide quick services to the user and to provide automated system of searching of blood in case of emergency.
* Also to provide automated system of recording the information of blood donor, blood donation program.

## 1.5 Objectives

The objectives of developing this system are mentioned below:

* The main reasons of developing this software is to provide urgent blood needed to public in case of emergency and to advertise the blood donation program.
* Also to provide the searching facilities of blood which are needed for the people.
* To provide the facilities of blood stock management functions in the blood banks by recording the details of the donor and available blood donor information.

# Chapter 2: Analysis

## 2.1 Introduction

Analysis is the process of gathering requirement for a software. Analysis is conducted for the purpose if studying a system or its parts in order to identify its objectives. This process also helped to solve the problem occurred in the system and again help the system to works efficiently. Analysis clarify what the system should do.

In the analysis phase developer communicate frequently with user, it will be easily for developer to determine the specific feature that user want and resolve of conflict in the requirement as demanded by the user and documentation of all features of the project development process from start to end. Also by performing this process, software can be testable easily.

We need to perform analysis due to following reasons.

* To gather all requirement for the software development.
* By performing brainstorming and walkthrough to understand the requirement.
* To know that, it will suitable for feasibility test to ensure that requirement is testable or not.

To complete this project I have used CATWOE Analysis. This analysis is one of many techniques that analyst uses to find what is the goal of their application and that goal was achieve or not and to find out problem area in that application.

CATWOE Stands for

* **Customer:** the receiver of the production from the system
* **Actor:** user who perform the role of system processing.
* **Transformation:** the main development that provide the outputs to the user.
* **World-Wide View:** the basic world-view for the change.
* **Owner:** the stakeholder with the complete power for the system.
* **Environment:** the rules and controls nearby the system.

## 2.2 Feasibility Study

A feasibility study is an analysis used in measuring the ability and likelihood to complete a project successfully including all relevant factors. To determine the goal of the software developer use feasibility study so that developer can achieve their goal. There are different types of feasibility study which are explained below:

* **Economic feasibility**

This feasibility study will helps to find out the cost and all the benefits of the purposed software.

This study will include following reasons:

* + - * Hardware cost.
      * Application’s software costs.
      * Documentation preparation cost.

Due this reasons my project is economically feasible

* **Technical feasibility**

This process will analysis the skill and capabilities of the team member while developing the software. Ascertains that the technology chosen for software development has a large number of users so that they can be consulted when problems arise or improvements are required. Also this project maintain all the above technical feasibility so my project is technically feasible.

* **Schedule feasibility**

In this study all time and date will be calculated to until the project complete. In the feasibility study not only requirement and cost will be gather but also time will calculate so that project can go through times and completed at given time. In this analysis we calculate all time required to complete this whole project. That’s why my project is schedule feasible.

* **Operational feasibility**

This study will helps to determine that new software can adapt by the user. Determine whether the organization is satisfied by the alternative solutions proposed by the software development team. I have provides user manual system so that user feel comfortable to use this system which means my system is operationally feasible.

* **Legal feasibility**

The process of studying to know that project conform the legal and ethical requirement is known as legal feasibility. It is important that the project or business is following the requirements needed to start a business or a project including business licenses, certificates, copyrights, business insurance, tax number, health and safety measures and many mores. This things needed to consider while performing legal feasibility study. To complete this project I have manage all the things as mentioned so that my project is legally feasible.

## 2.3 Requirement Analysis

### 2.3.1 Functional Analysis

This process will identify the function or behavior of the software. Also this process specify what system will do.

**Function for User (Front end):**

* Responsive Template, Mobile Friendly
* Registration System for blood Donor
* search engine for blood group and location
* Donor Details Information
* Inquiry form

**Function for Administration (Admin Section):**

* Manage Blood Group(add, delete)
* Add and Delete Donor Information
* Enable and Disable Donors
* Manage to Contact us Queries
* Update the contact us Info
* Admin Dashboard

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Title** | **Description** | **Rational** | **Dependency** |
| FR01 | Responsive Template, Mobile Friendly | To make application attractive and to make supportive on mobile version | To make attractive interface | N/A |
| FR02 | Registration System | This will record donor detail on the system. | To know the detail of donor | N/A |
| FR03 | search engine | This will helps user to search blood donor through blood group and location | Donor information can find easily | FR02 |
| FR04 | Donor Details Information | Donor record can be found easily and can be contact through information | User can donor information | FR02 |
| FR05 | Inquiry form | User can contact us through inquiry form and can give feedback to our system | This maintain relation between user and developer | N/A |
| FR06 | Manage Blood Group(add, delete) | Admin can remove those blood group which are not required or already donated and can be added blood group which are new to the system | Blood group can be add and delete by admin | FR02 |
| FR07 | Add and Delete Donor Information | Those detail which are not necessary for the system can be removed by the admin and admin can add new donor info in the system | Admin can have authority to manage the donor detail. | FR04 |
| FR08 | Manage to Contact us Queries | Those who have bad comment can be remove by the admin. | Admin have right to manage all the inquiries form. | N/A |
| FR09 | Update the contact us Info | Those provide info in the contact us section cab be manages by admin | Contact regularly updated by admin | FR08 |
| FR10 | Admin Dashboard | Admin can watch all list detail of donor, list of inquiry forma and register user from this function. | Total number of detail of all information can be look this function. | FR02,FR05 |

### 2.3.2 Non Functional Analysis

The process specifying how the system will performed is known as non-functional analysis. This analysis will include all activities which is not performed by functional analysis. The non-functional analysis are mentioned below:

* **Performance**

In this process the speed of the software will be measure so that it can measure how much software is efficient in given task. This can be measure by the MIPS (millions of instruction per second).

* **Supportability**

This is the process of install, configure and monitor the software in the system. This process also used to find the error and debug in the software. This process is also done within the maintainability cost.

* **Security**

This is the most important part of the analysis in where developer must see the security by seeing which framework we used. In this process also we have to see that network is secured we were using from the firewall. Also we have to application infrastructure security so that no one can hack our software.

* **Maintainability**

It’s about code quality how easy for someone else to read the code and maintain application by someone else and the code quality is not good it cannot maintain by someone other so during the process must keep in mind the application must be read or maintained by someone else instead of application maker.

* **Usability**

In this function, the frequently used function should be tested for usability. Give the more prioritizes to most important function in the system so that can tested to maintain system easily.

### 2.3.3 Moscow Prioritization

MoSCoW is a prioritization technique for helping to understand and manage priorities. The letters stand for:

* **M**ust Have

In this priority software will be unsafe without it. Also without this function software cannot provide solution.

* **S**hould Have

In this priority solution will not be much important but it can play vital role in this software.

* **C**ould Have

In this priority solution will be less impact if that function is left.

* **W**on’t Have this time

This also helps to manage expectations that some requirements will simply not make it

Into the Deployed Solution, at least not this time around.

Below mentioned table Moscow Prioritization which give priority in my functional and non-functional requirement analysis.

|  |  |  |
| --- | --- | --- |
| **ID** | **Title** | **Priority** |
| **Functional Requirement** | | |
| FR01 | Responsive Template, Mobile Friendly | Should have |
| FR02 | Registration System | Must Have |
| FR03 | search engine | Must Have |
| FR04 | Donor Details Information | Should Have |
| FR05 | Inquiry form | Could have |
| FR06 | Manage Blood Group(add, delete) | Must Have |
| FR07 | Add and Delete Donor Information | Must Have |
| FR08 | Manage to Contact us Queries | Won’t Have |
| FR09 | Update the contact us Info | Won’t Have |
| FR10 | Admin Dashboard | Should Have |
| **Non Functional requirement** | | |
| FR11 | Performance | Should Have |
| FR12 | Supportability | Could Have |
| FR13 | Security | Must Have |
| FR14 | Maintainability | Should have |
| FR15 | Usability | Could Have |

### 2.3.4 SRS

SRS is a software requirement specification of the software system to be developed with its function and non-function requirement. This is done between client and developer to make agreement on the requirement provided by the client.

**Software Requirement**

**Programming Language:** PHP version 5.6

**Database:** My SQL 2014

**UI Design:** HTML 5, JQUERY, JAVASCRIPT

**Web Browser:** Mozilla, Google Chrome, and other browser

**Software Used:** Apache Server

**Hardware Requirement**

**Memory:** 4GB RAM

**OS:** Windows 10 64 bits

## 2.4 Use Case

Use Case is the process of interaction between user and system in a particular environment. The characteristics of use case are:

* Manage functional requirement.
* Record the interaction between system and user.
* Record developments from initiate result to objective.
* Explain the flow of the system through diagram.

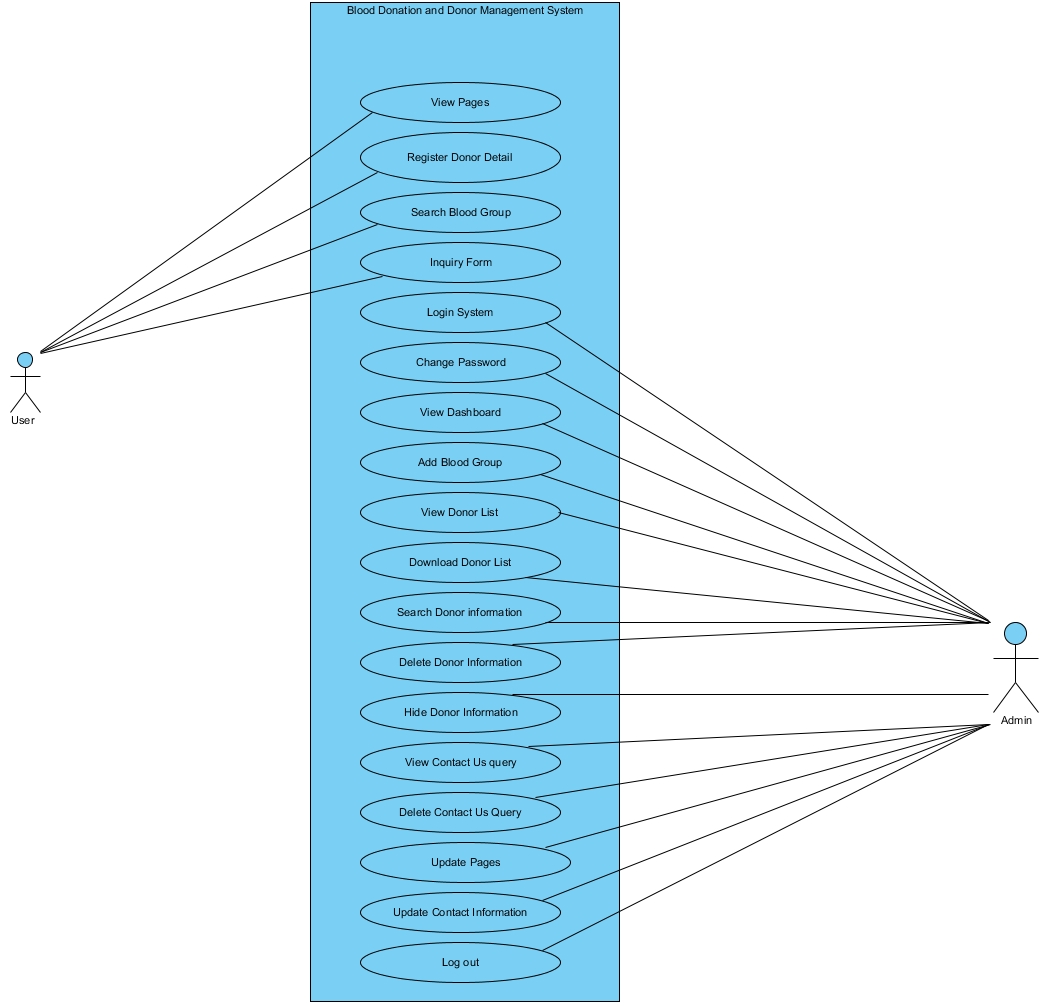


Figure 1: Use Case

|  |  |  |
| --- | --- | --- |
| **Id** | **Title** | **Description** |
| 1 | View page | User can view the pages directly without any login to the system. |
| 2 | Register Donor Detail | Donor have to register their detail to the system so that donor can identified |
| 3 | Search blood group | User who needed blood can search through the blood group and location |
| 4 | Inquiry form | User can inquiry about the required blood in emergency |
| 5 | Login system | This system is valid for only admin |
| 6 | Change password | This right is given only to the admin. |
| 7 | View dashboard | Admin can look all information through dashboard |
| 8 | Add blood group | Admin have to add blood group into system so that user can find blood group. |
| 9 | View donor list | Admin can view the donor list through admin section. |
| 10 | Download donor list | Admin have authority to download the list of donor. |
| 11 | Search donor information | There is a search engine for admin to search the donor information |
| 12 | Delete donor information | Admin can delete the donor information if they already given blood. |
| 13 | Hide donor information | Admin hide the detail of the donor from the system. |
| 14 | View contact us query | Admin can view contact us query and manage it |
| 15 | Delete contact us query | Admin can the contact us query that provide by the user. |
| 16 | Update contact information | Admin can update the information provided in the contact us form |
| 17 | Log out | Finally admin have to logout from the system so that other user cannot use admin section. |

## 2.5 Class Diagram

**Initial Class diagram**

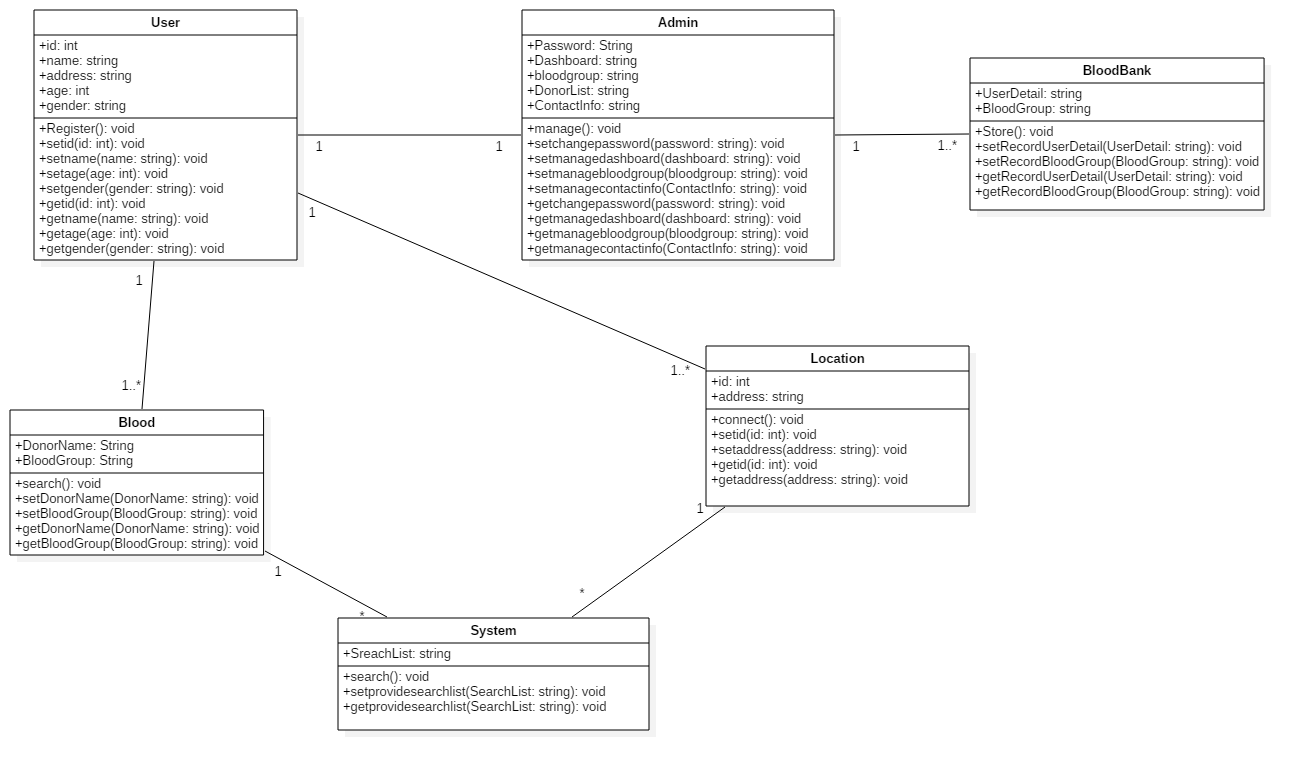


Figure 2: class diagram

# Conclusion

Finally, the analysis of blood donor and donation management system is completed. Also to analysis the system CATWOE analysis is used. In this analysis, functional and non-functional requirement for system is list out. And the listed requirement are prioritize through MoSCoW prioritization. With the help of requirement use case diagram also created and from this diagram the flow of the system is find out. In this use case the actor, system and use case are define through the different role found in the system.

# Chapter 3: Design

## 3.1 Structural design

### 3.1.1 Class Diagram:

**Introduction**

Class diagram is structural diagram which is static structure that shows the system process by defining the classes, attributes and operation use in it. Class diagram will define the relationship between object that were used in system.

**Purpose of using class diagram:**

* It helps to provide the static configuration of classifier in the system.
* It is helpful for developer and other member too.
* It can be used to model the system from business viewpoint.

**Notation Used in class diagram:**

1. **Class Name:**

* The name of class perform in the first part.
* It can be extract for the scenario’s noun.

1. **Class Attributes:**

* Attributes are shown in the second part.
* Attributes record into associate variables in program.

1. **Class operations:**

* It helps to provide services to class.
* Operation record into class method in program.
* The return type of a process is displayed after the colon at the end of the method signature.



**Explanation**

This is final class diagram for my project which is mentioned above. There are five classes generated with the help of noun. I have used different operations for this classes. Also I have used setter and getter for methods for this classes. They all are interconnected with each other. I have used one to many relation for the classes.

### 3.1.2 Dataflow Diagram (DFD):

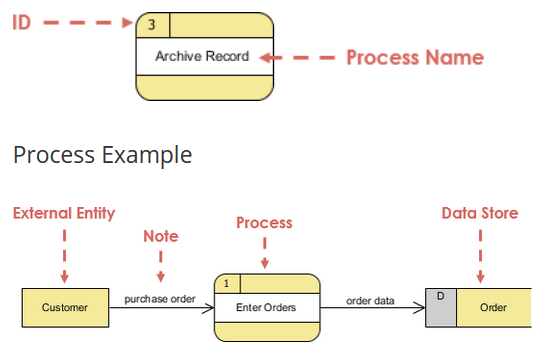
**Introduction**

Dataflow diagram are the process which involves in the system to transfer data from the input to file storage. It can be dived into two parts that is logical dataflow diagram and physical dataflow diagram.

**Notation used in DFD:**

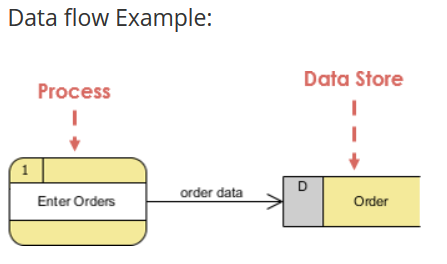
1. **Process:**

* A process receives input data and produces output with a different content or form.
* Processes are given ID for easy references.



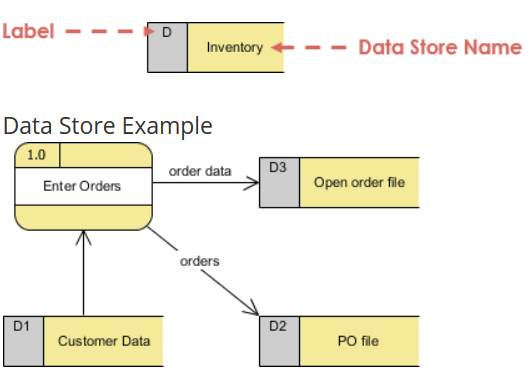
1. **Data flow:**

* A data-flow is a route for data to transfer from one part of the information system to another.
* Straight lines with received arrows are input data flow.
* Straight lines with leaving arrows are output data flows.

****

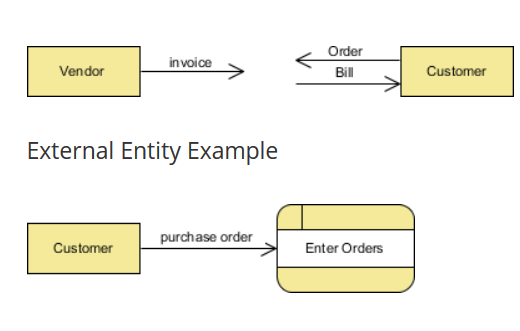
1. **Data store:**

* A data store is used in a data-flow diagram to represent a location when the system must keep information because one or more methods need to use the kept information in a future.

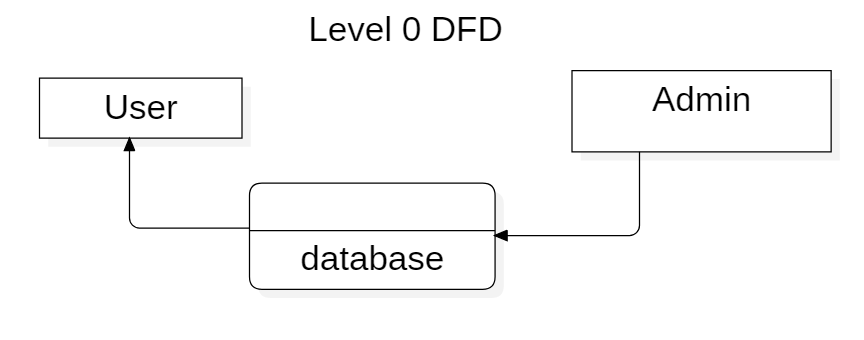


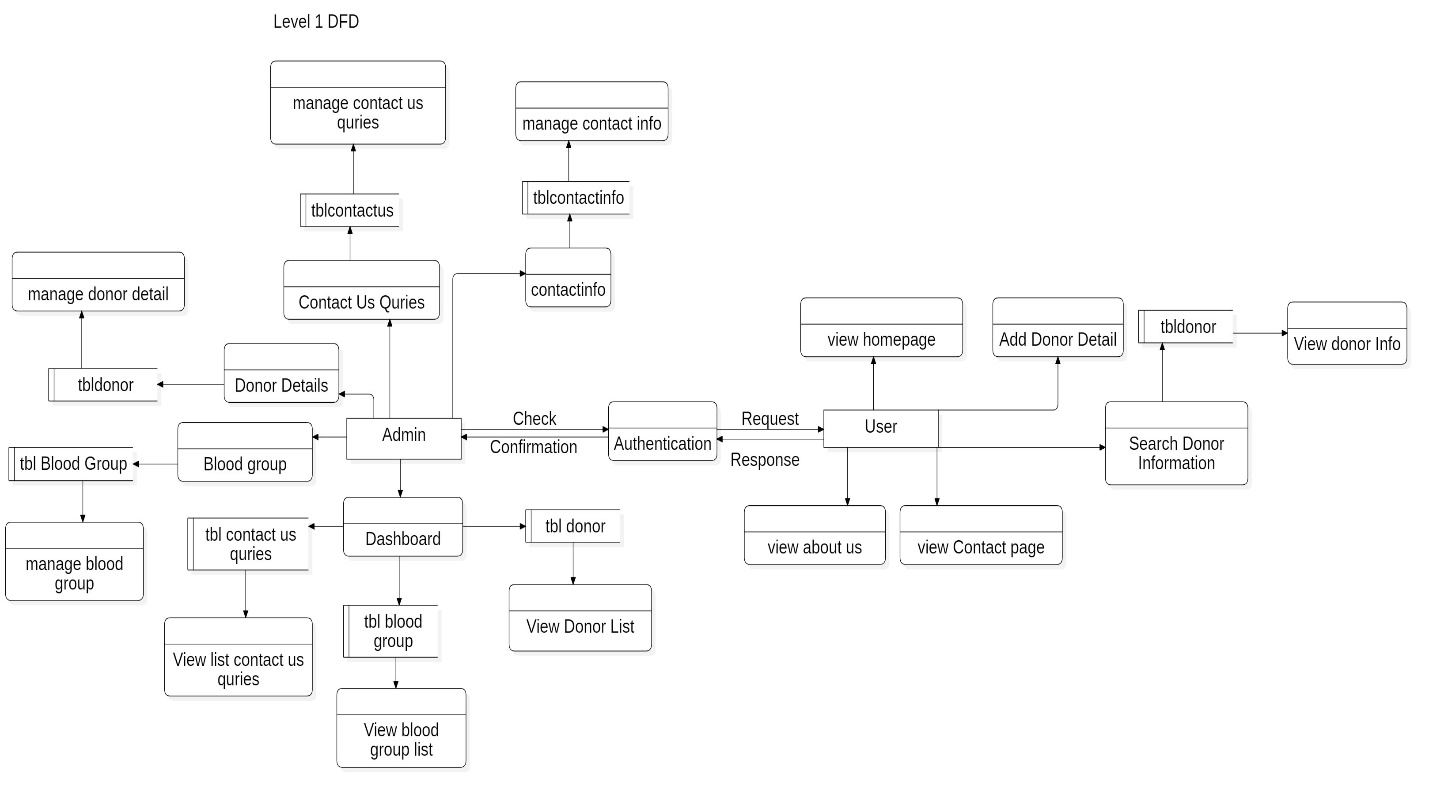
1. **External Entity:**

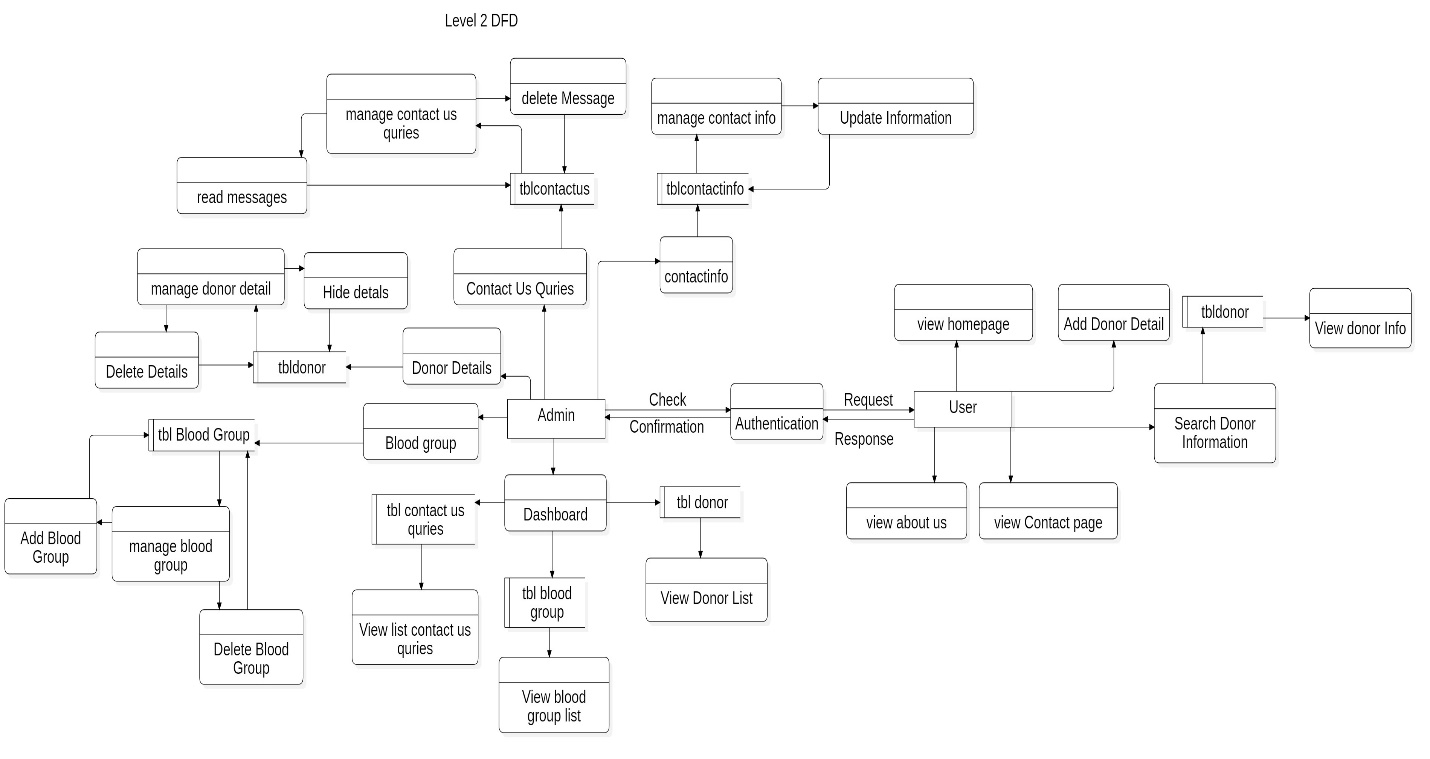
* They either send data or receive data.
* They do not process data.



**Diagram:**

****





**Explanation:**

I have create DFD In three level that is level 0, level 1 and level 2. In level 0 I have just shown the user database and admin. After then I have come to level 1 DFD in where user and admin are two entity for DFD. In user entity, user has to send request for authentication after response user can use application to look homepage, add donor details, about us page, contact us page and search the required blood through location and blood group and these all data will be store their respected database table. After that in admin entity can see all the donor information, message and store blood group in the dashboard. Admin can manages donor information, pages, contact info, and send messages and their data will be store in the respected database. After completing level 1 DFD, I have created level 2 DFD in where user can only use application to look homepage, add donor details which will be store in tbldonor in database table, about us page, contact us page and search the required blood through location and blood group. After that in admin entity can see all the donor information, message and store blood group in the dashboard. Admin can manage donor details that is admin can delete or hide the donor details from the tbldonor table in the database. Admin also manage blood group that is admin can update or delete blood group from tblblood from the database table. Admin can update contact us information from tblcontact table in the database. Also admin can manage send message from user that is read or delete the message from tblcontactus table in the database. In this way I have completed the Dataflow diagram for my application.

## 3.2 Behavioral design

### 3.2.1 Activity Diagram

**Introduction:**

Activity diagram is another main behavioral diagram in UML Design to define dynamic features of the system which is basically an advanced form of flow chart that forming the flow from one activity to another activity.

**Notation Used in Activity diagram:**

**Activity –** It used to represent set of action.

****

**Action –** A task to be performed.

****

**Control Flow –** It shows the sequence of execution.



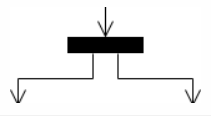
**Initial Node – it will start set of action or activities.**



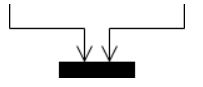
**Activity Final Node – it will stop all control of flow and object flow in the action.**



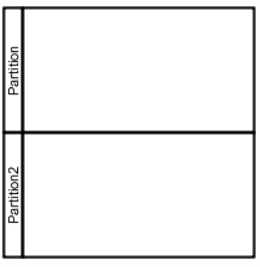
**Fork Node – It will split behavior into set of parallel flow of action.**



**Join Node – it will merge all split behavior from set of parallel flows of action.**

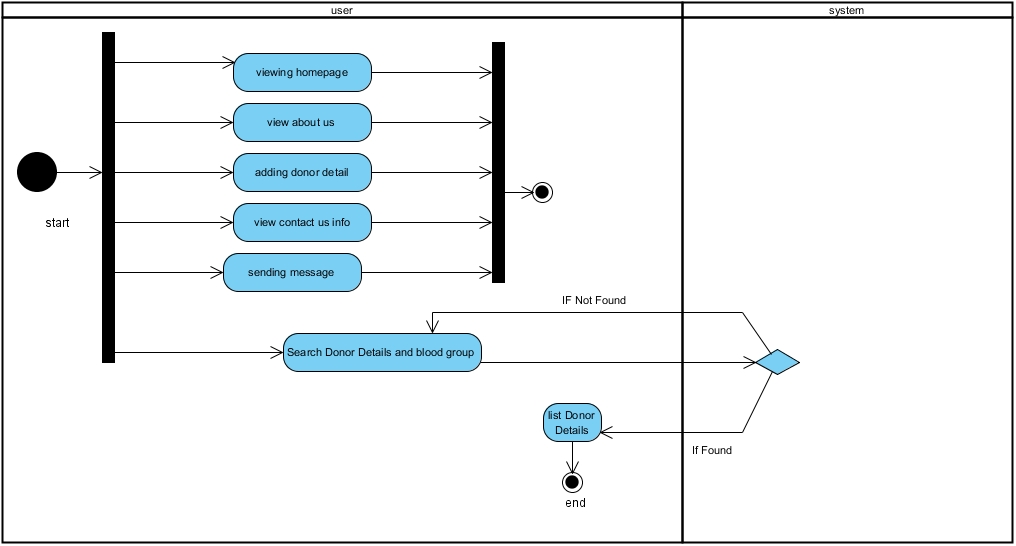


**Swimlane and Partition -** A way to collection actions executed by the same actor on an activity diagram or to collection actions in a single line.

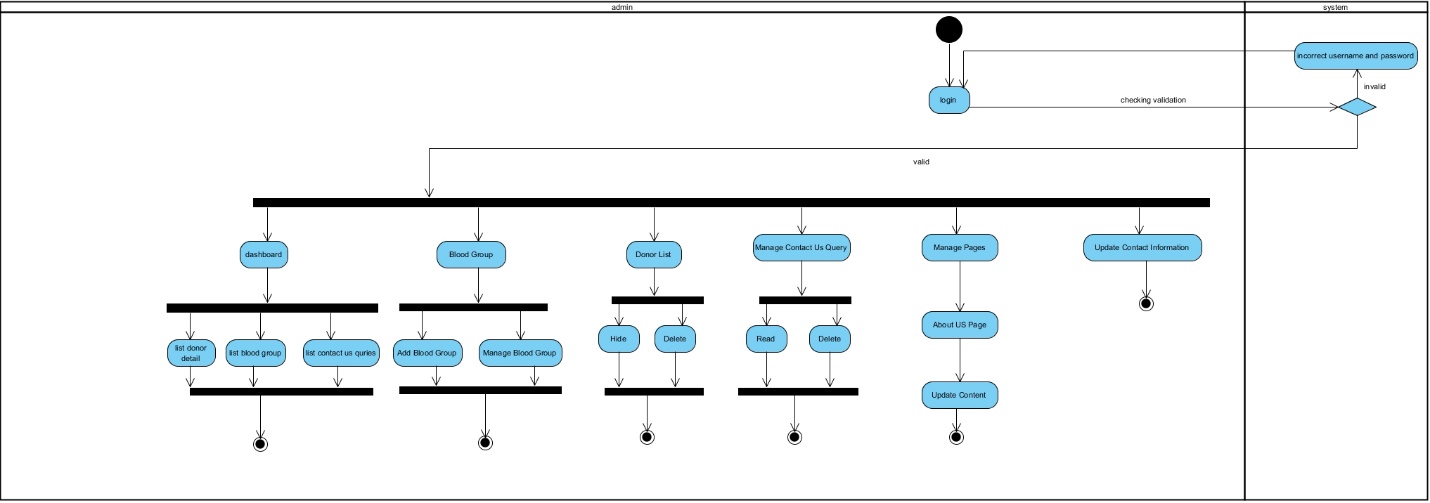


**Diagram:**

**For user:**

****

**For admin:**

****

**Explanation:**

For activity diagram I have separated two section for application that is for user and admin.

In user section user can view home page, about us page, contact us page, and search. In the search page, user has to enter blood group and location and then system will verify the data and If found it will listed the blood donor details and if not found it will send message not found.

After that admin part will come in where admin can do all the system manages. In admin section admin has to login into system which will be verify by the system. If system found the valid details it will open dashboard for the admin and if username and [password is incorrect then it will send message that invalid username and password. After login into system admin will manage the donor details that it delete and hide. Also admin will delete and add blood group. After that about us page will updated. Then at last contact info will be updated. Finally my activity for this application will complete.

### 3.2.2 Sequence Diagram:

**Introduction:**

The diagram which interact with system to carry out the detail how operations performed is known Sequence diagram. They capture the relations between object in the perspective of a cooperation.

**Notation Used in Sequence diagram:**

**Lifeline -** A lifeline represents an individual participant in the Relations.



**Call Message -** A message defines a specific communication between Lifelines of an Interaction.

Call message is a kind of message that represents an invocation of operation of target lifeline.



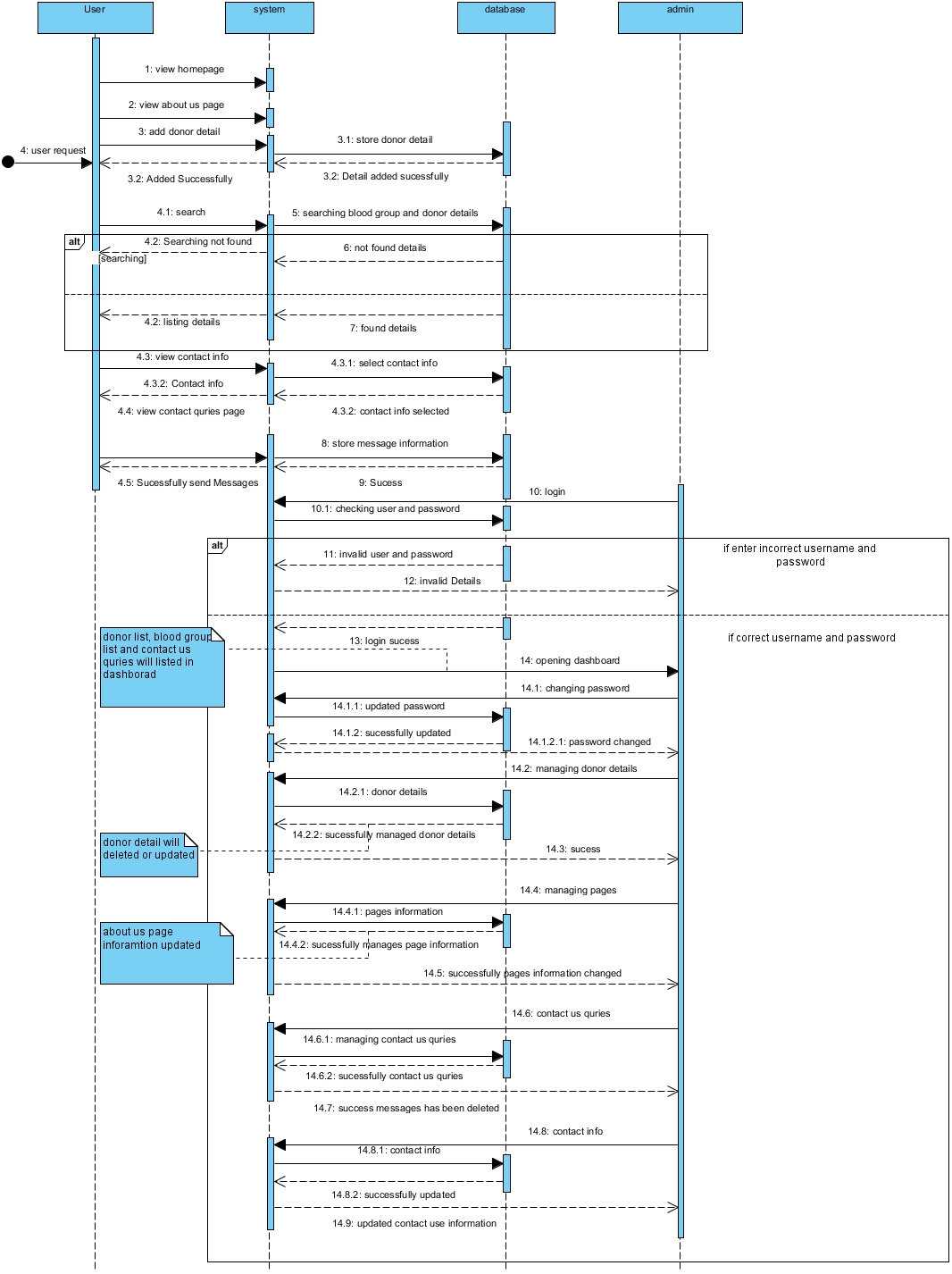
**Return Message -** Call message is a kind of message that characterizes call of process of mark lifeline.



**Note -** A note (comment) gives the capacity to attach different comments to elements. A comment carries no semantic force, but may contain information that is useful to a user.



**Diagram:**

****

**Explanations:**

In this sequence diagram I have created four lifeline and they are user, system database and admin. At first user request to view home page, about us page, contact us page then system will response to their request and open that request pages. After that user will add donor details then it will be send to system and system send that data into database. Then user search will required blood group a then system will search it and if found it will display the list and if not found then it will display not found message to the user. After then admin lifeline will come in where first admin have to login into system if the user and password is correct then it will open dashboard in where list of donor detail, list of contact query and list of blood group will be shown. If not correct username and password then it will show invalid detail. After login admin can manage the blood groups, manage pages, updated contact info and contact query. This process was shown in above diagram. This how I have create sequences diagram for my project

## 3.3 Database design

### 3.3.1 Data Dictionary:

**Introduction**

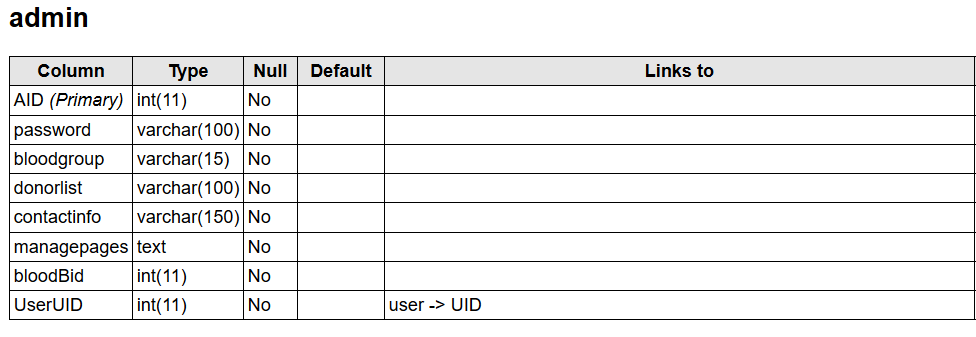
It is a record of data elements in a database or data model with detailed description of its format, relationships, meaning, source and usage.

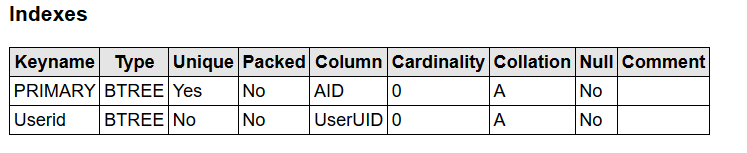
**DBMS data dictionary**

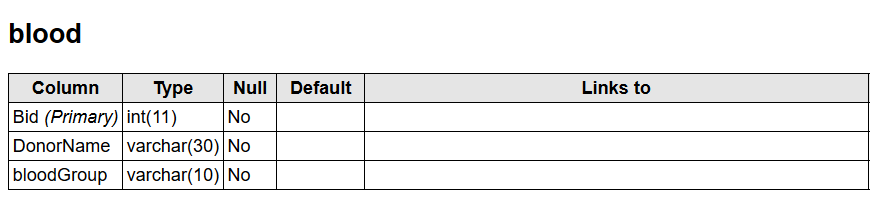
Relational database engines support much more explanation of data models and provide this info through their data dictionaries. This information is:

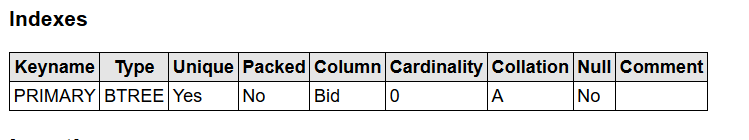
* Data type of presented in the columns.
* Nullable column.
* Relationship of tables (Like foreign key).
* Unique values that presented in the column (like primary key etc.).

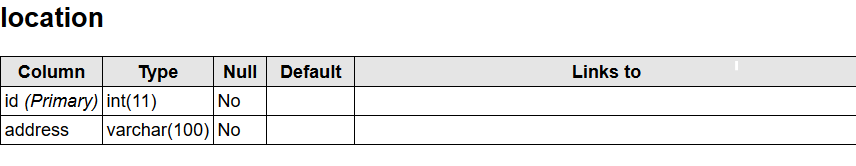
**Diagram:**

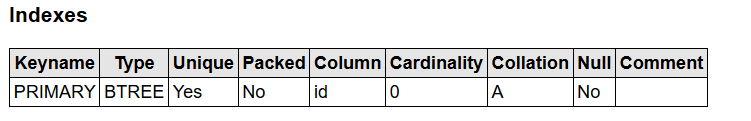
****

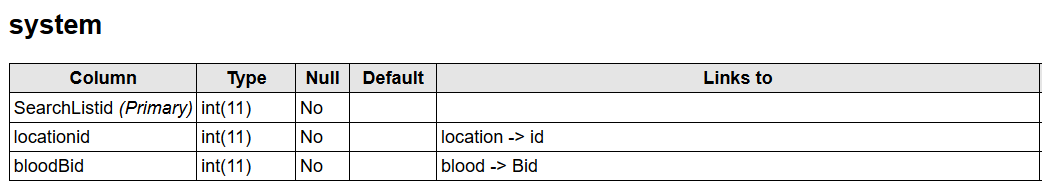
****

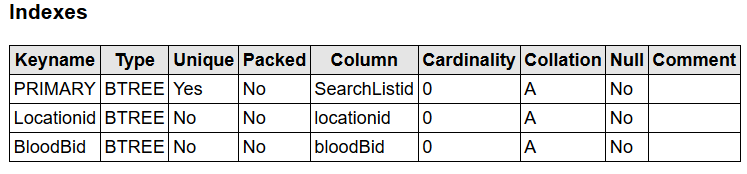
****

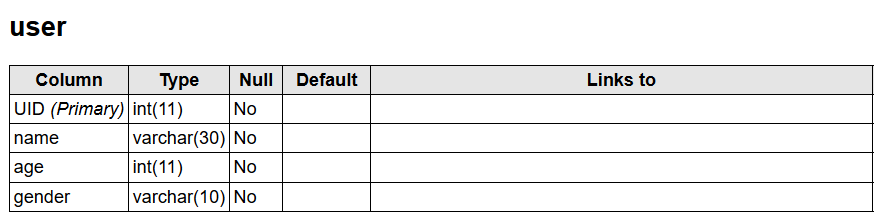
****

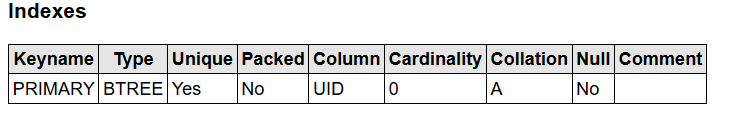
****

****

****

****

****

****

### 3.3.2 ER Diagram:

**Introduction:**

Entity Relationship Diagram is a type of structural diagram for use in database design. In this diagram there different symbols and connectors were used which visualize two vital information they are:

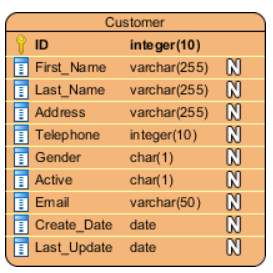
* The major entities within system scope and
* The relationship between these entities

**Notation used in ER Diagram:**

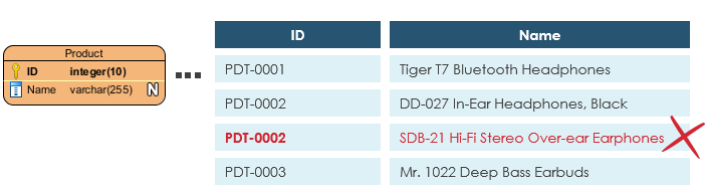
**Entity -** An entity is a **definable thing or concept within a system.**

****

**Entity Attributes -** an attribute is a **property of the entity that holds it.**



**Primary Key –** The special kind of entity attributes that are store a record in database table which define uniqueness in the table.

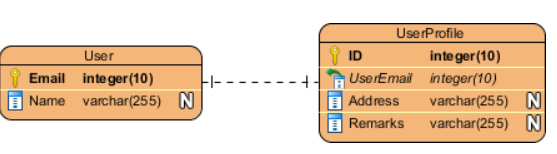
****

**Cardinality -** Cardinality defines the **possible number of existence in one entity which are linked to the number of existences in another.**

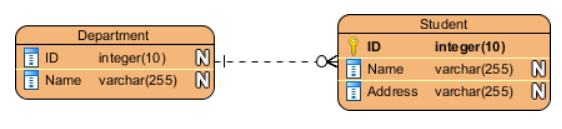
There are three main cardinality relations used which are mentioned below:

* One-to-one,
* One-to-many, and
* Many-to-many.

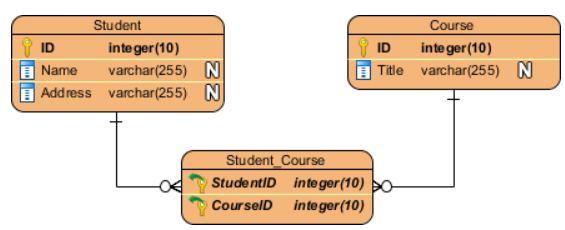
**One-to-One cardinality example:**

****

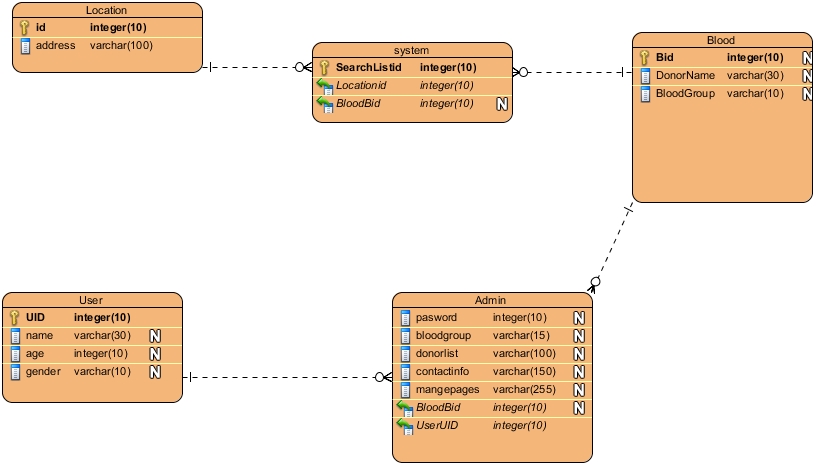
**One-to-Many cardinality example**

****

**Many-to-Many cardinality example**

****

**Diagram:**

****

**Explanation:**

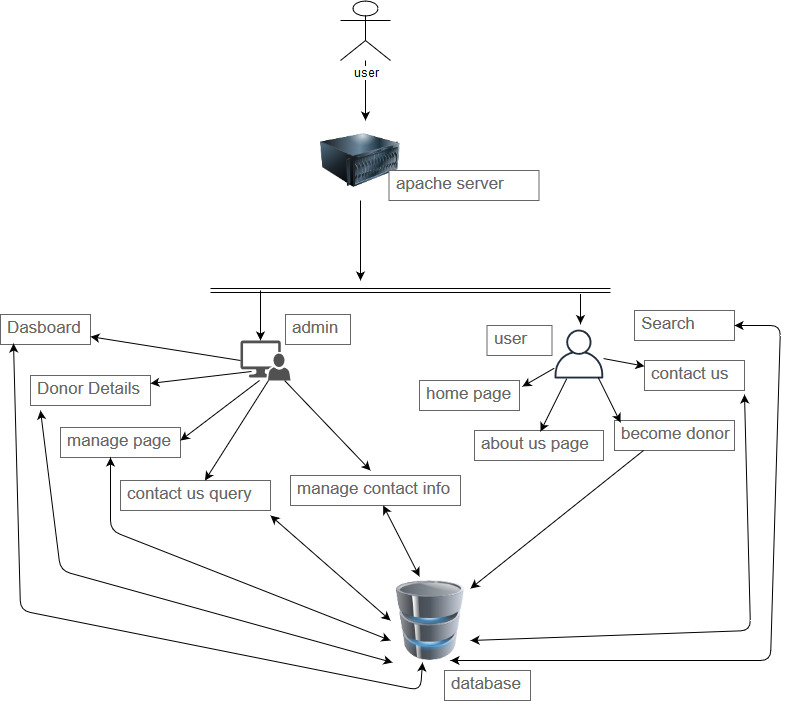
This is ER diagram for my project. In this diagram I have created 5 entity which are location, system, blood, user, and admin. In this entity also there are different attributes uses inside the entity. I have given primary key for each entity. Along with there are foreign key also given for the required entity. I have use one – many relation for this diagram.

## 3.4 System Architecture

**Introduction**

The structural diagram which define the function of automated system is system architecture. Architecture can be estimated built on business purposes in ranges such as cost, functionality, reliability, maintainability and operability.

**Diagram**

****

In this system architecture design I have used 3 tier design. In this diagram user send request to sever for connection and after getting connection user will have two option that is admin and user. Also after getting connection user if user is client then client can use only user interface in where home page, about pages, become donor page, contact us and search option will be located. Or if the client is admin then he authority to use dashboard, managing donor details, manage page, manage contact us query and manage contact info. All the required data will be store in the database.

**Why I have used 3 Tier in system architecture:**

There are various benefits to using a 3-layer architecture including speed of development, scalability, performance, and availability.  As mentioned, modularizing various tiers of an application gives progress teams the capacity to progress and improve a software with more speed than developing a alone code base because a specific layer can be improved with least impact on the other layers. By splitting out the different layers we can scale each freely conditional on the want at any given time.

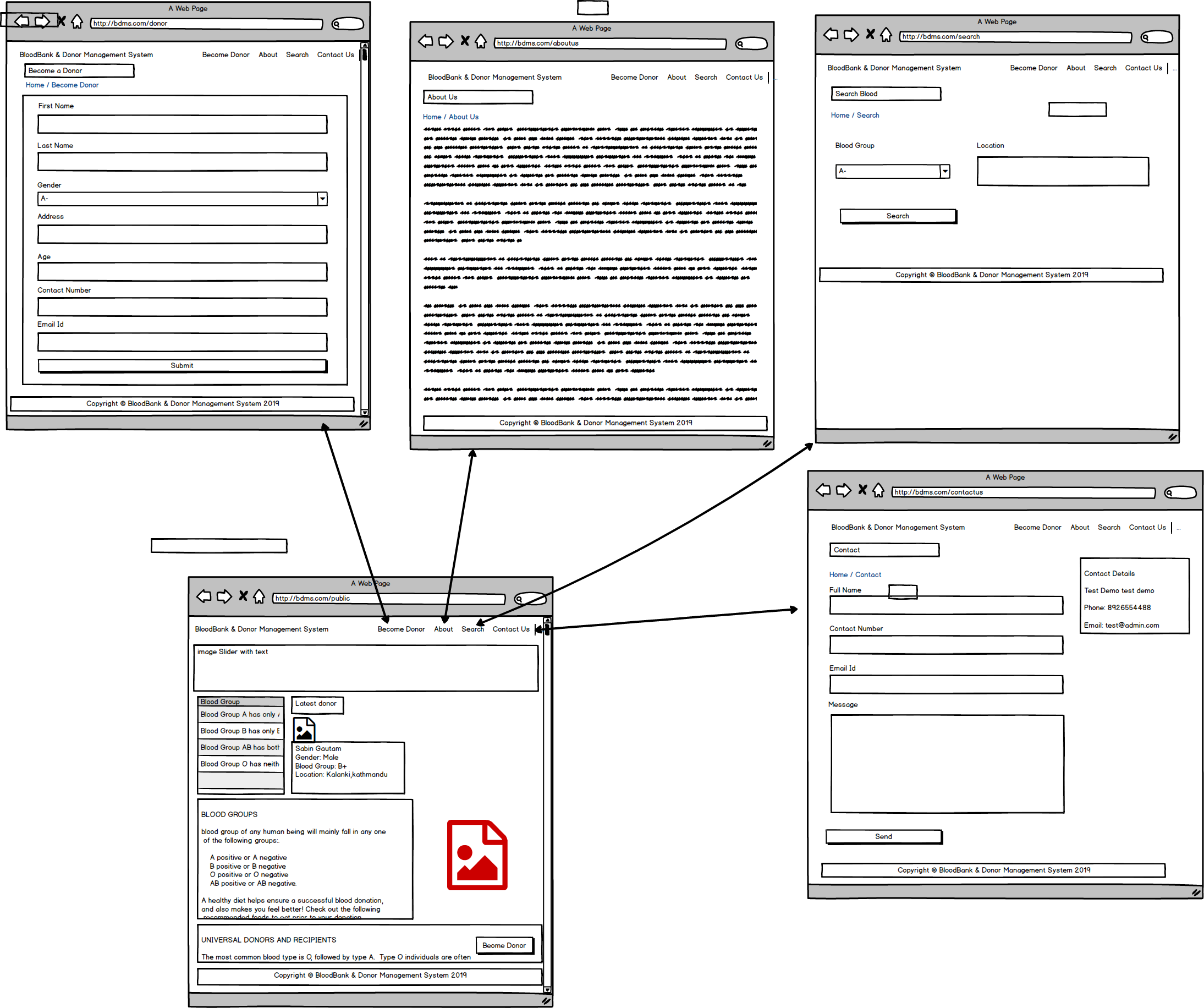
## 3.5 User Interface

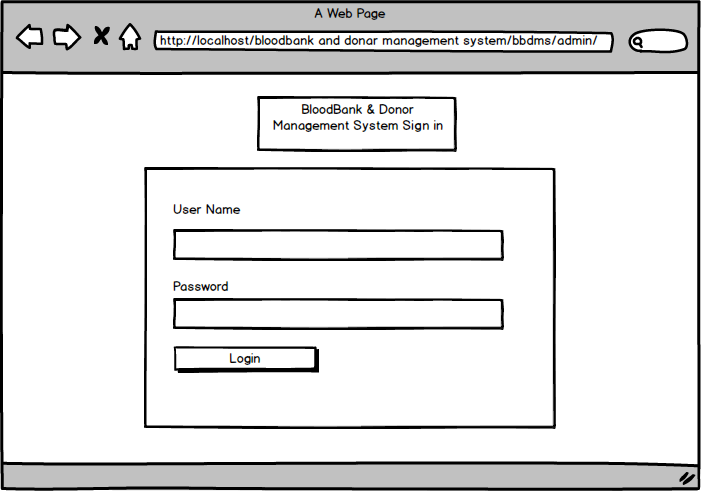
### 3.5.1 Prototype:

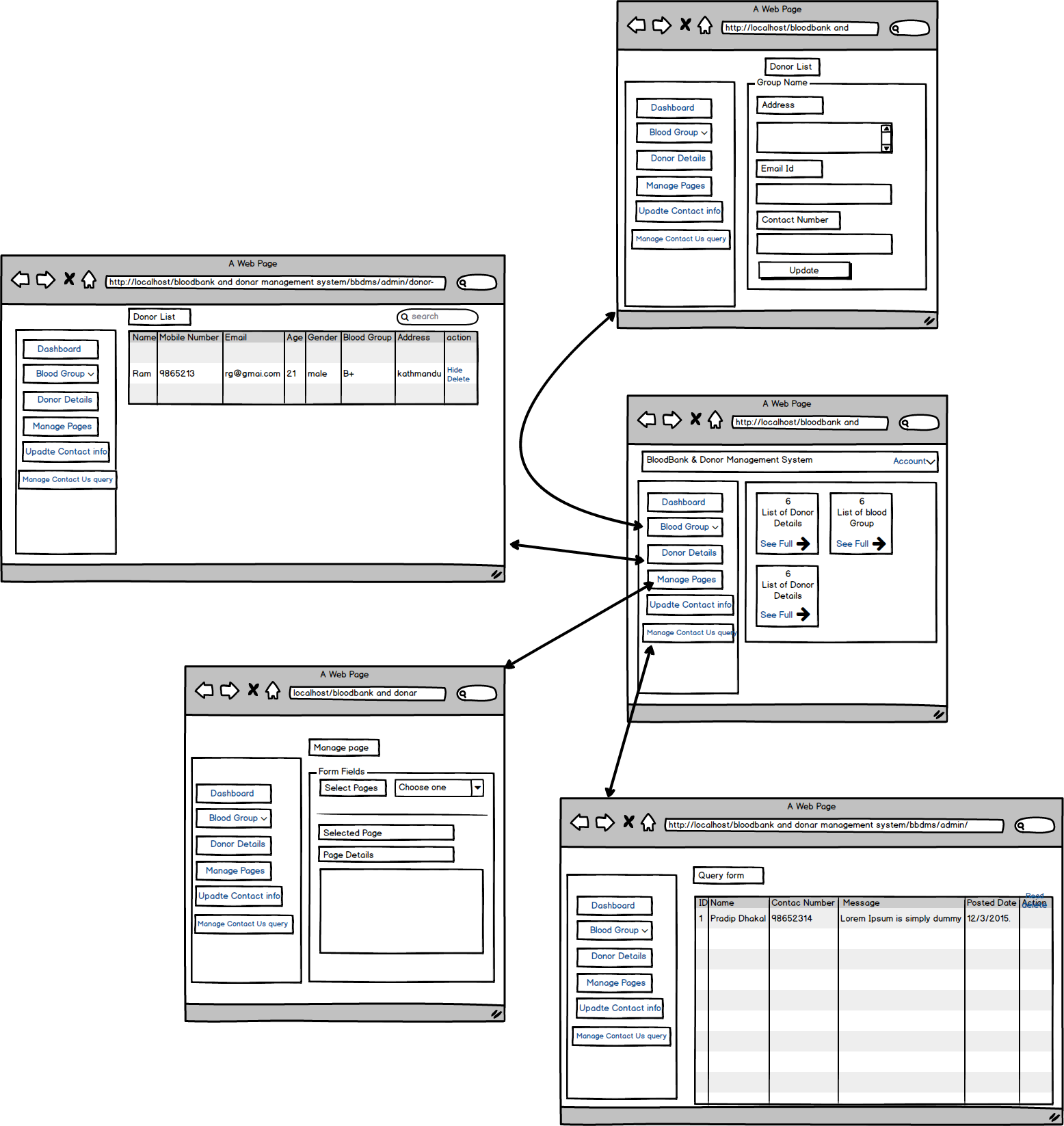
**Introduction:**

A prototype is a rudimentary working sample, model, mock-up or just a simulation of the actual product based on which the other forms (final product, and variations) are developed).

**Diagram:**

****

****

****

**Explanation**

Above mentioned diagram is the prototype of my project. There are two interfaces that is one for user and another is for admin. In user there is home page will open at first then user can go to different pages. And for admin there is login system that means admin has to login into system to use the interfaces. After login admin will open dashboard from that dashboard admin can find list of donor details, list of blood group, list of messages send from user. Also he can connect to different pages from that dashboard as shown in above picture.

# Conclusion

This is design section for my project. By performing this design I have come to how to create class diagram, activity diagram, sequences diagram and so…on. Also I have create prototype and data dictionary for my project.

# Chapter 4: Testing

## 4.1 Introduction

The process finding out the system is performing well or not is known as testing. In computer software, testing is used to find the key points in overall process which determines whether the function are working or not.

## 4.2 Importance of testing

* Software testing is used find the customer satisfaction and reliability in the application.
* Testing also ensure the quality of the product. Quality product supplied to client supports in achievement their self-confidence.
* Testing is essential for an actual performance of software application.
* Its main importance is that software should not produce any failures result because it can be very costly future.

## 4.3 Types of testing



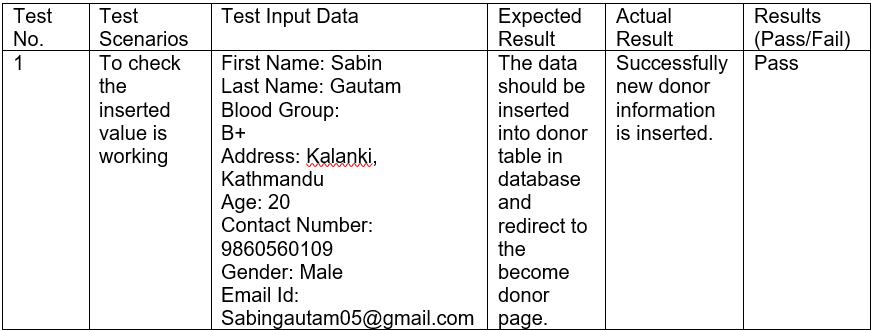
Among these testing I have chosen only following testing

### 4.3.1 Black box testing:

In this design, the internal system is not consider to test but the tests are based on the requirements and functionality of the system.

**For User Black Box Texting:**

**Test Name: Registering Donor Form**

****

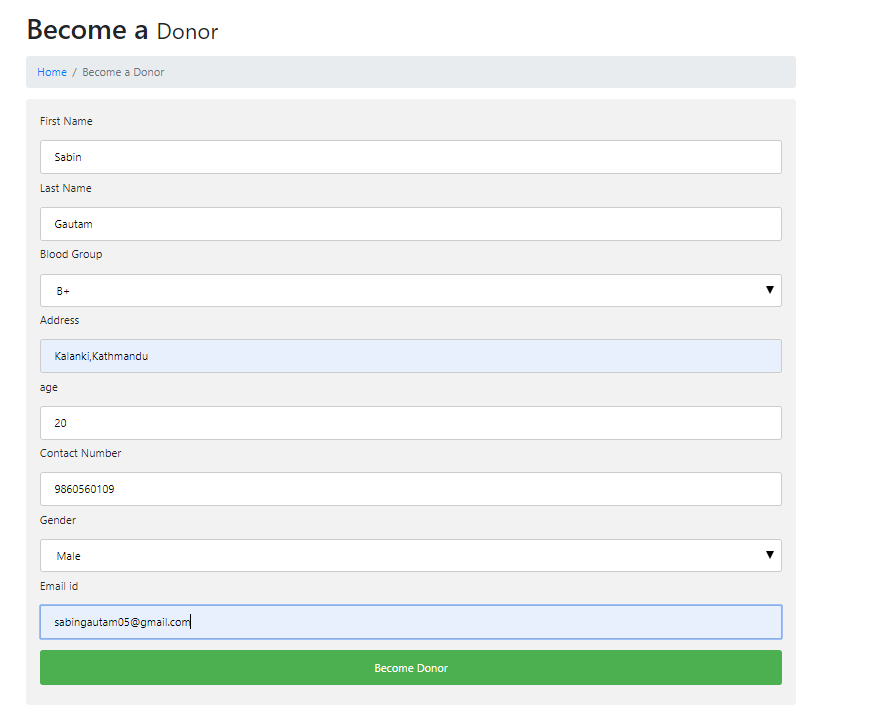
****

Figure 1: Inserting donor Information

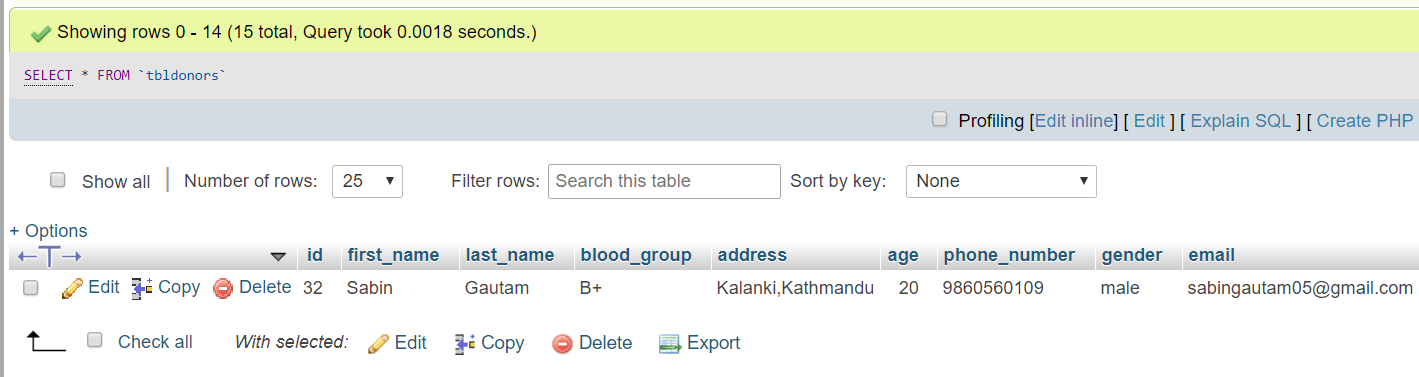
****

Figure 2: database after inserting donor information

**Test Name: Searching Found**

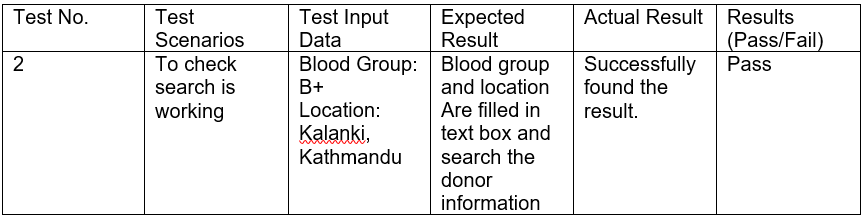
****



Figure 3: Searching for result

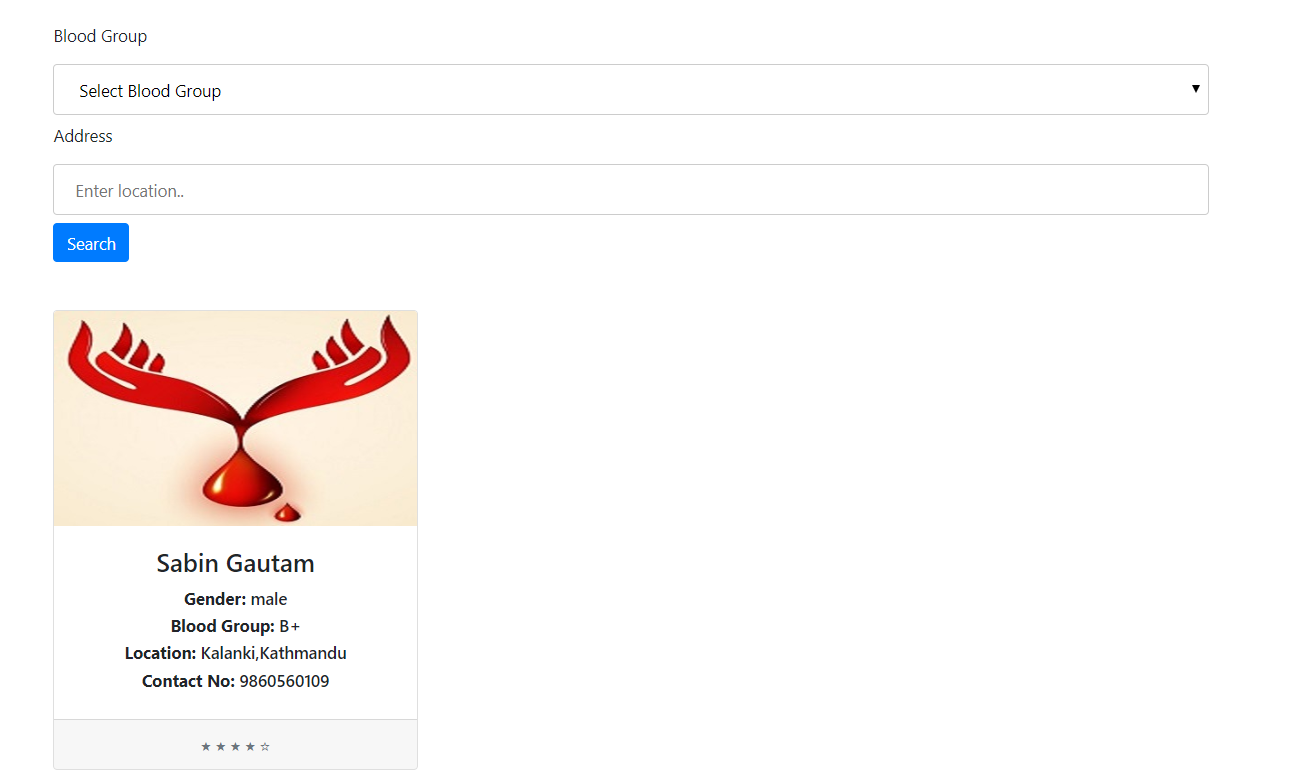
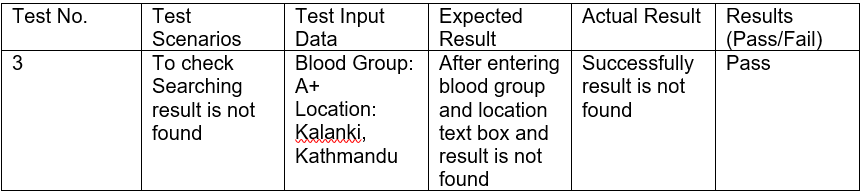


Figure 4: Result found

**Test Name: Searching not found**



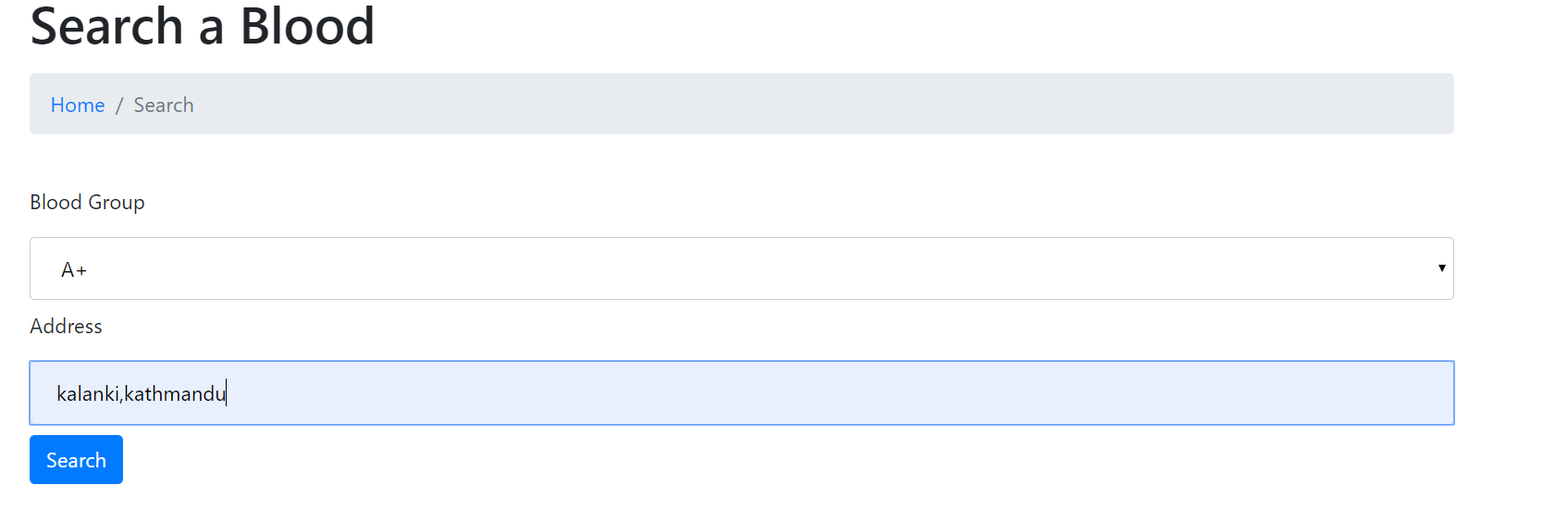


Figure 5: Entering input searching for result

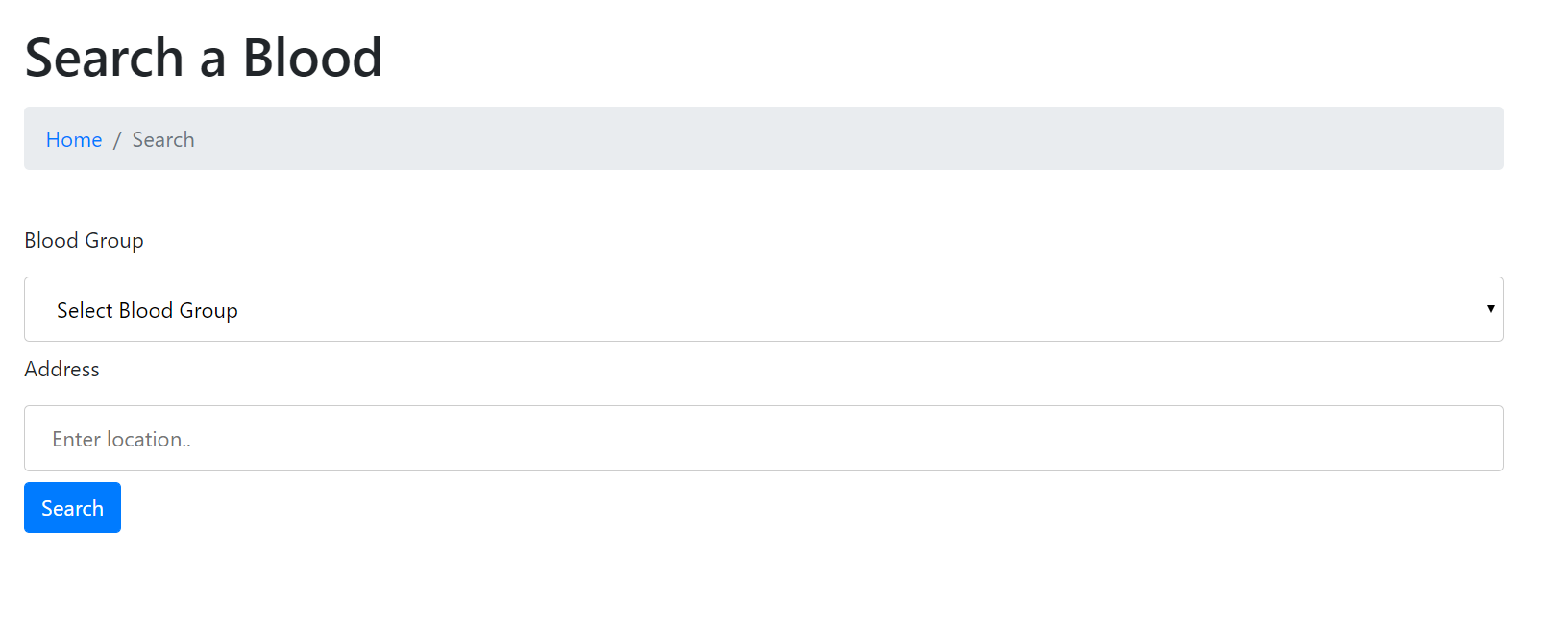
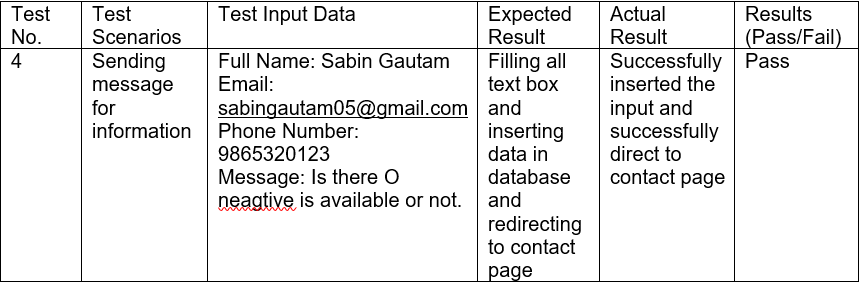


Figure 6: Result not found

**Test Name: Sending Message**

****

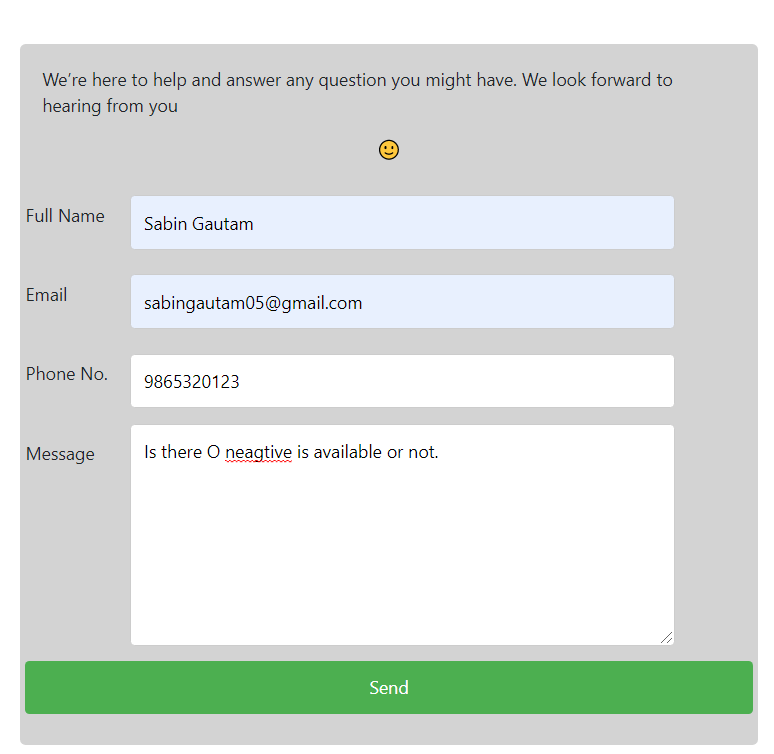
****

Figure 7: Entering Input for sending Message

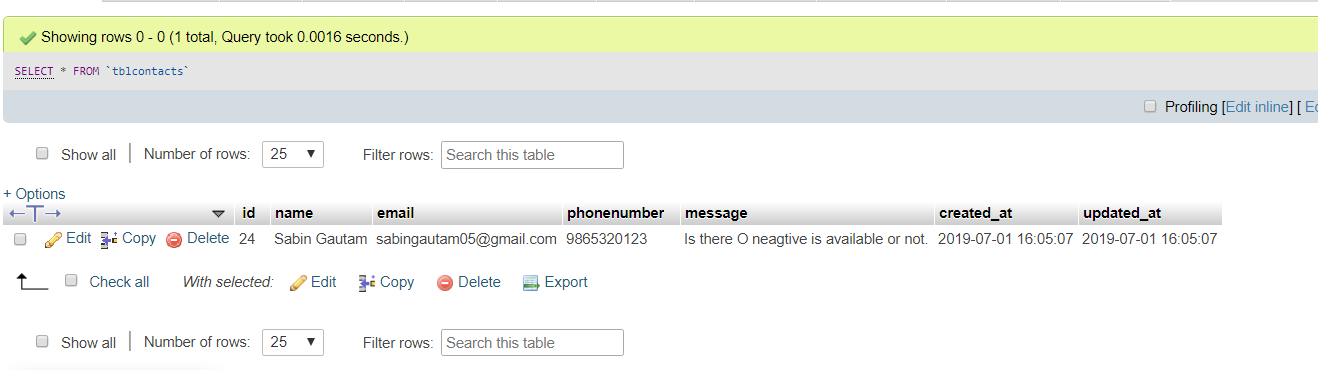
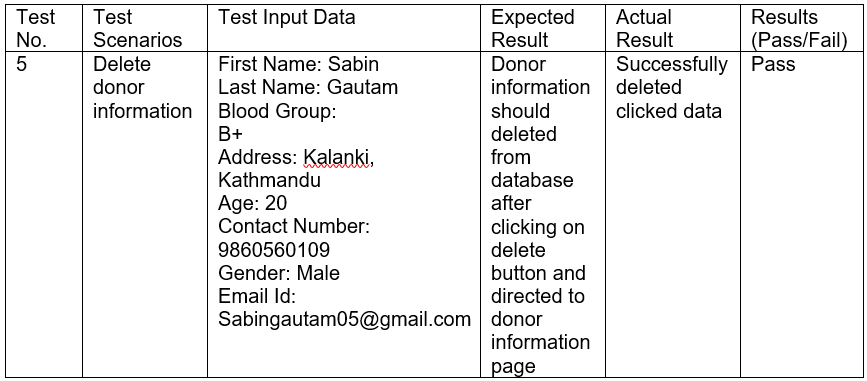
****

Figure 8: database of send message

**Black Box testing For Admin**

**Test Name: Deleting Register Donor Information**



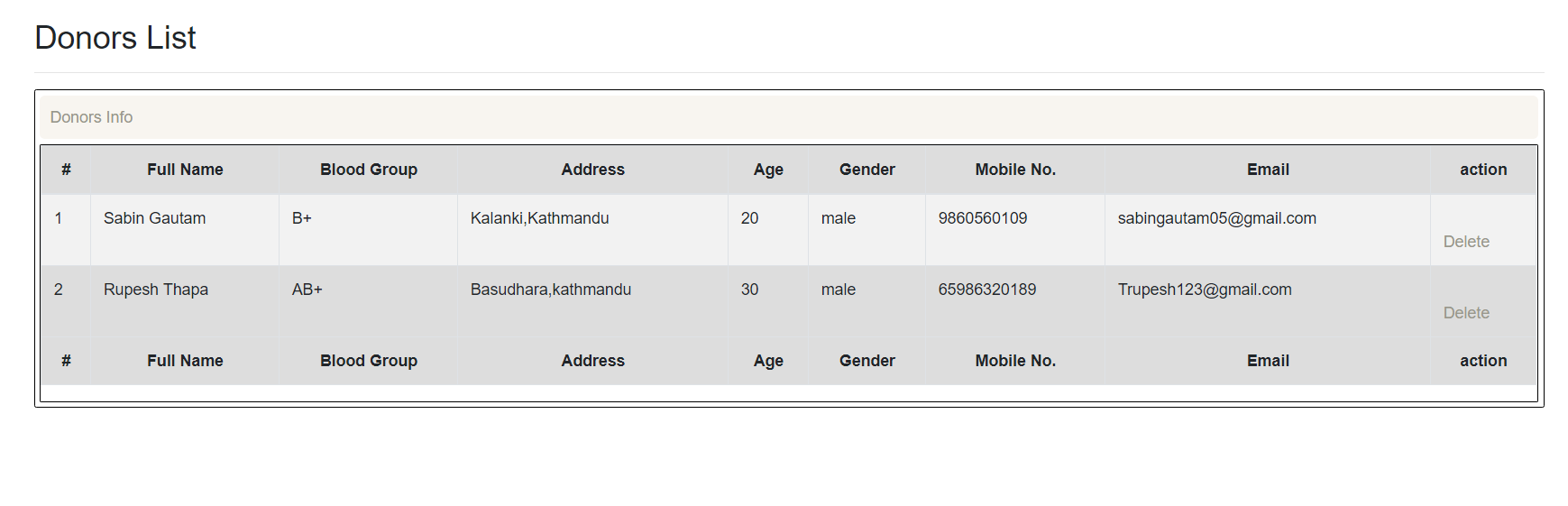


Figure 9: Donor information before delete

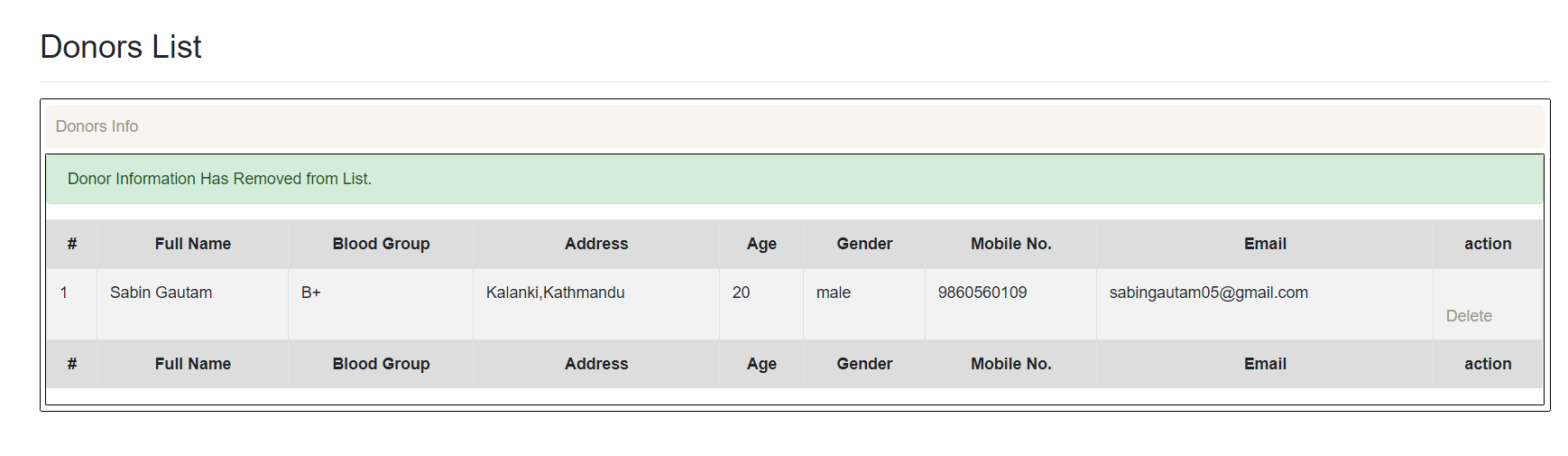


Figure 10: Donor information after delete

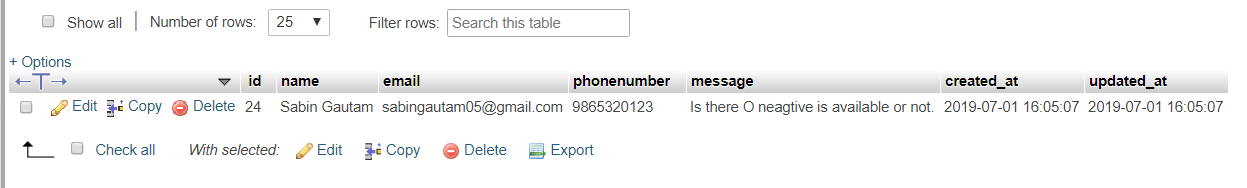


Figure 11: database record after delete

**Test Name: Deleting Record Blood Group**



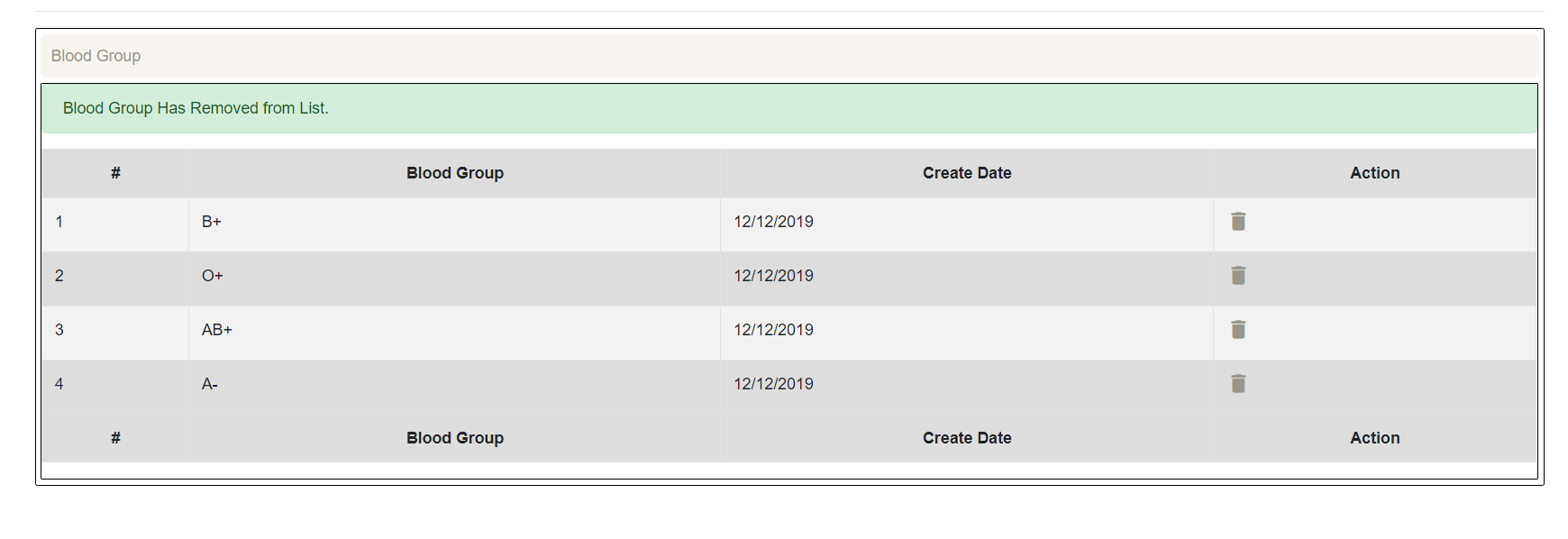


Figure 12: Deleting blood group Record

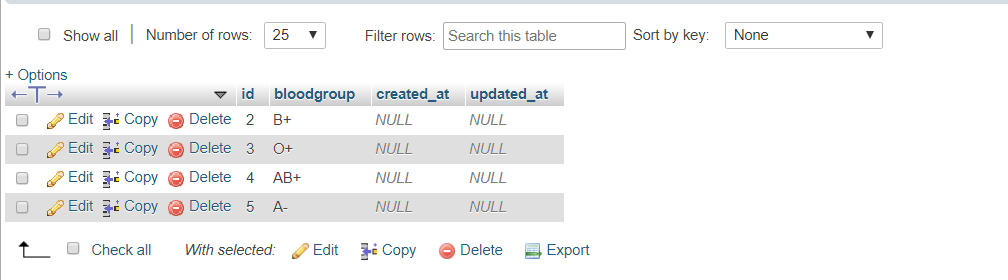
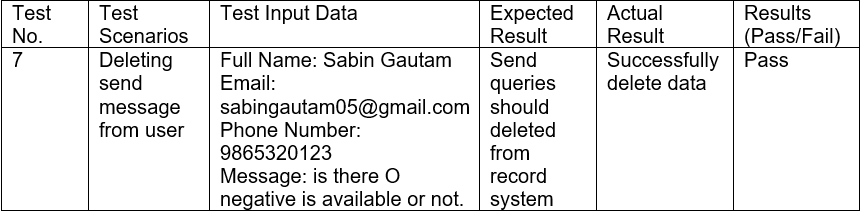


Figure 13: database of record blood group after record

**Test Name: Deleting Contact us Queries**

****

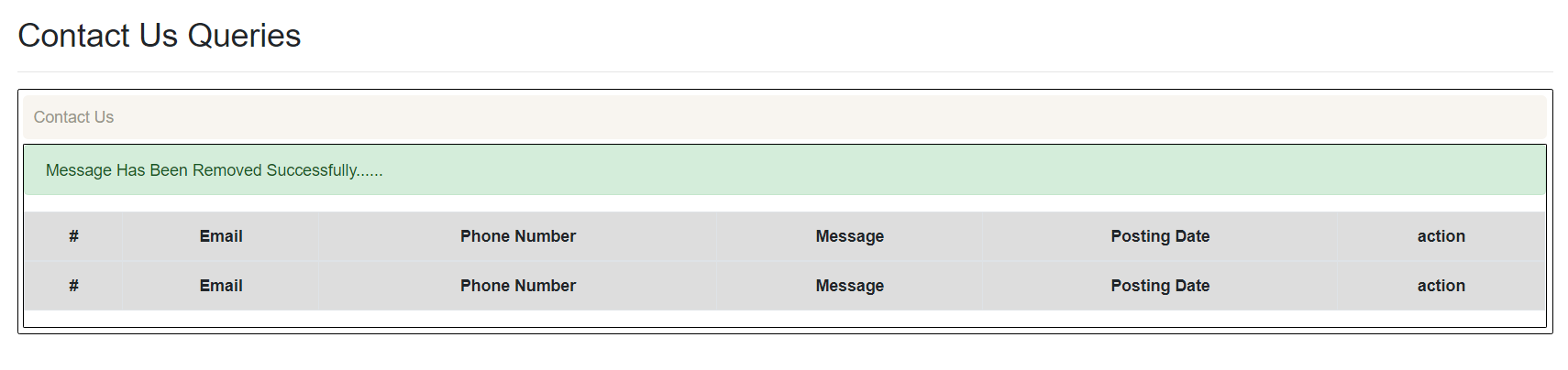
****

Figure 14: deleting send message

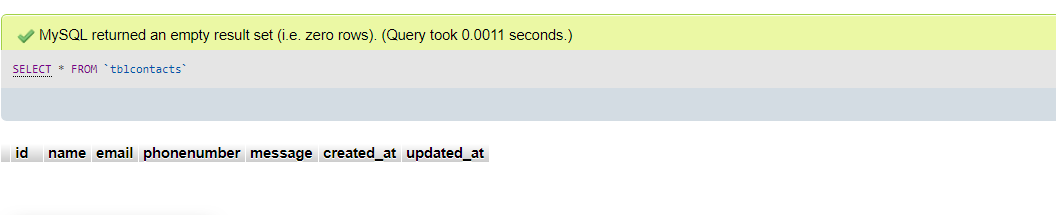
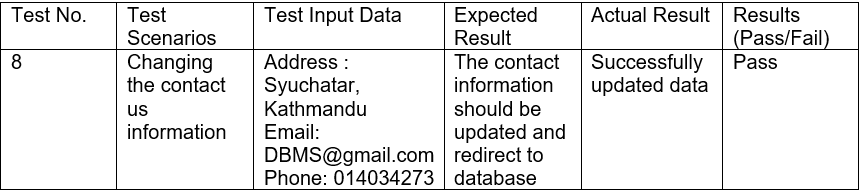
****

Figure 15: database of deleted message

**Test Name: Updating Contact Information**

****

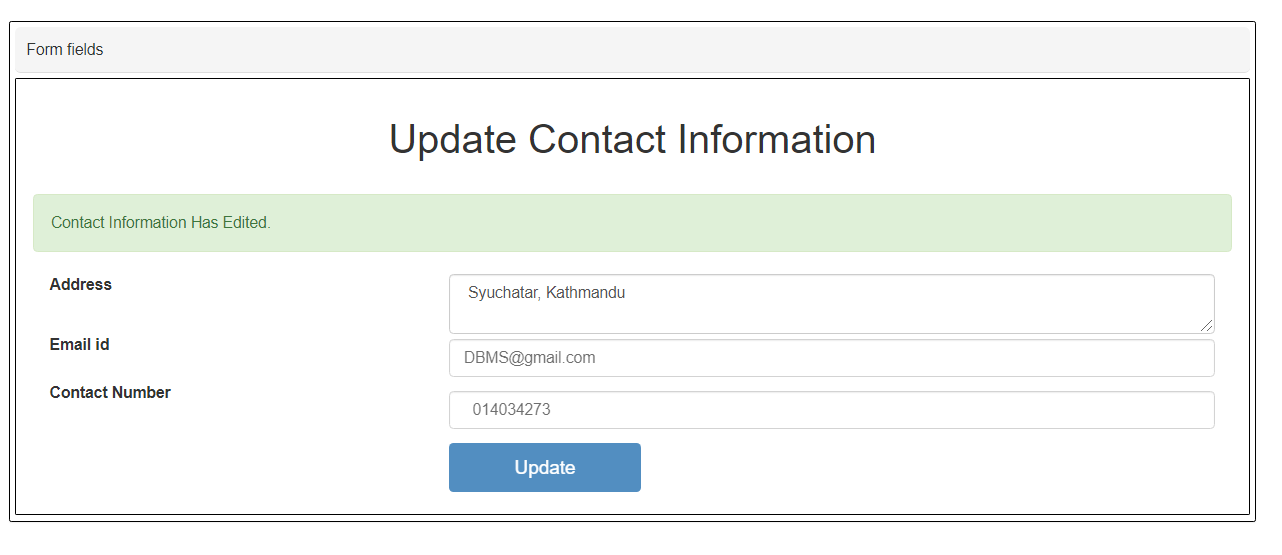
****

Figure 16: entering input to update contact info

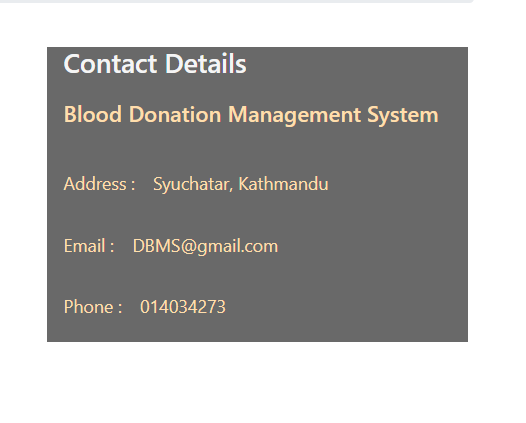
****

Figure 17: displaying update information

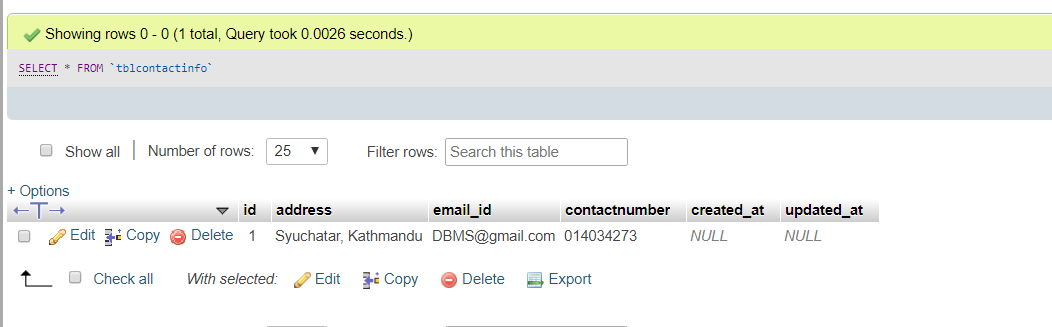
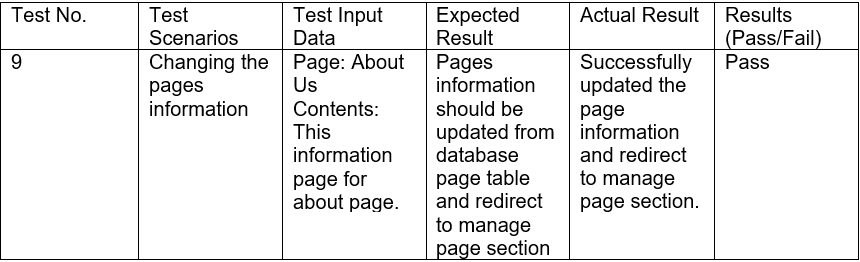
****

Figure 18: database contact information for user

**Test Name: Updating About Us Page Information**



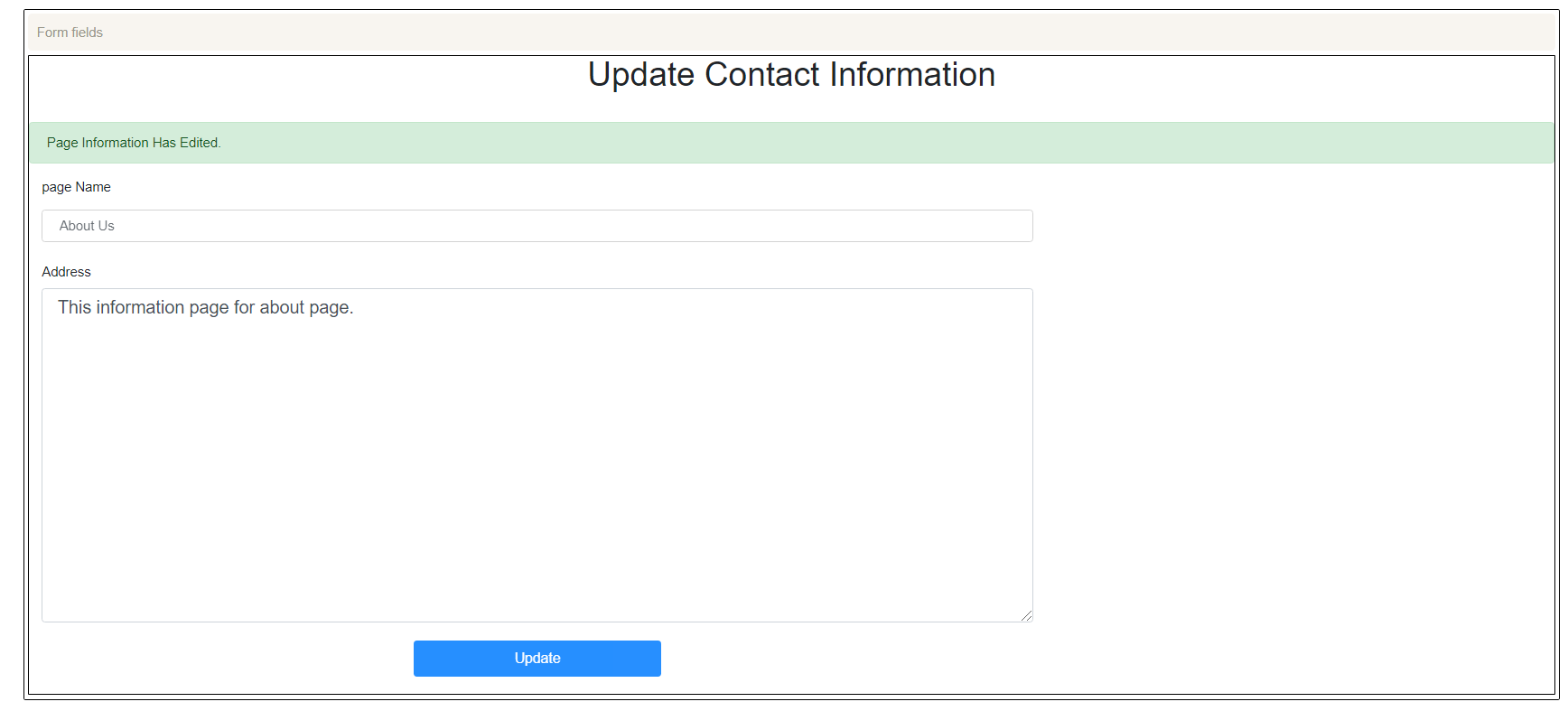


Figure 19: Updating about us page information



Figure 20: Displaying information after update

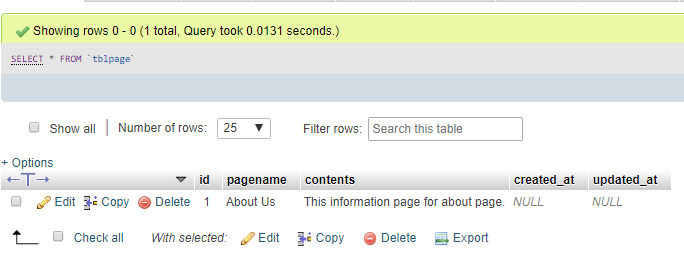
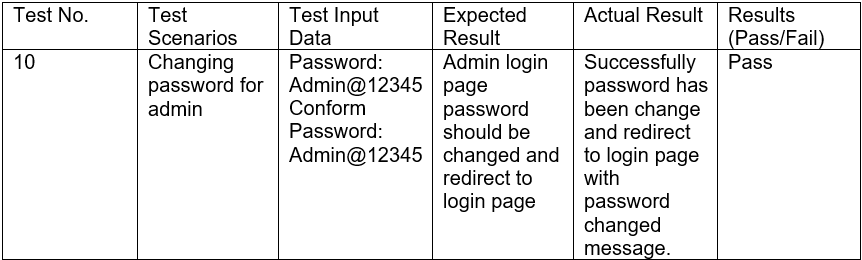


Figure 21: database of Updated information in page

**Test Name: Changing Password for Admin**

****

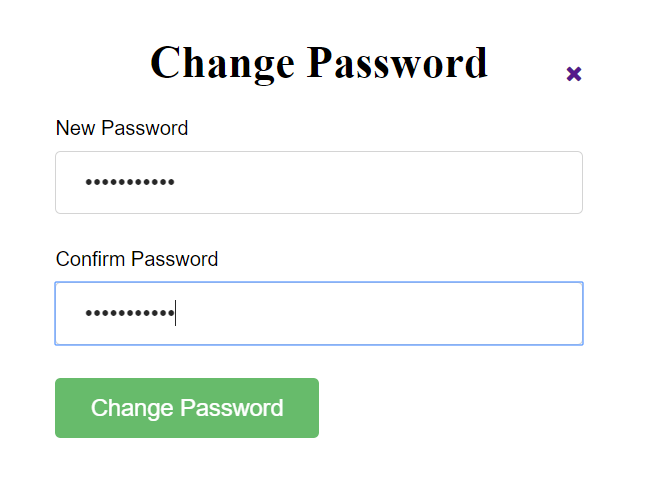


Figure 22: Layout for changing password

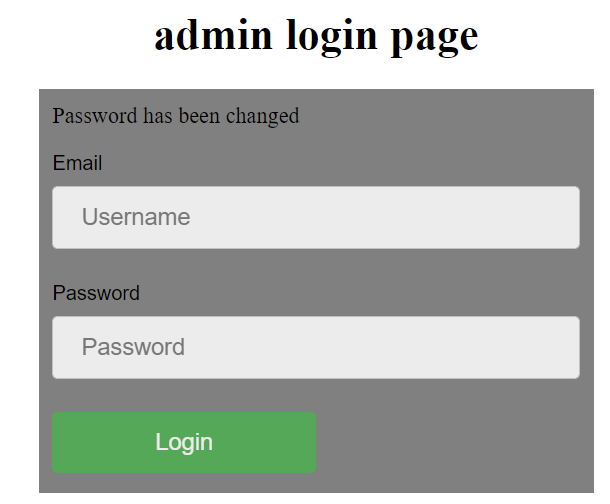


Figure 23: Redirect to login after password change

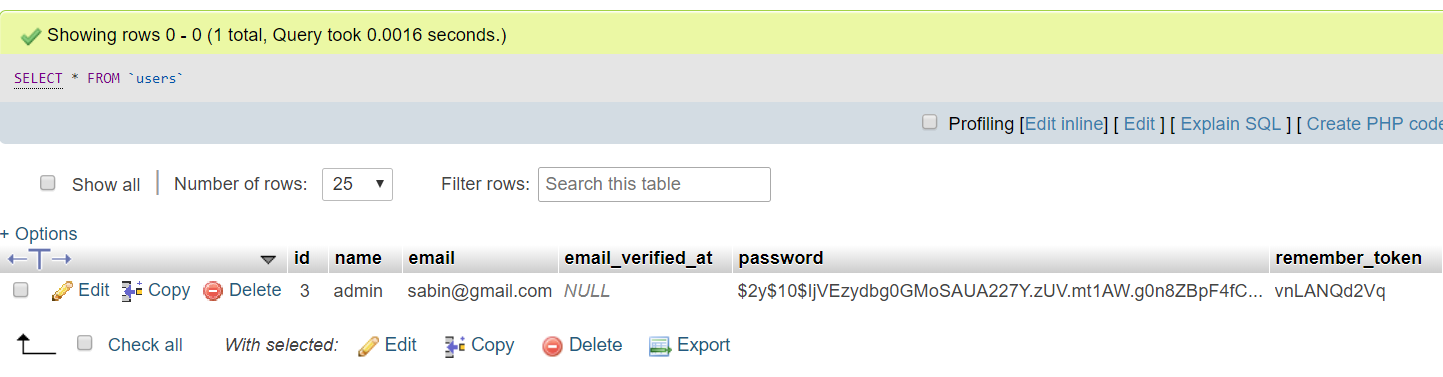


Figure 24: database after password change

### 4.3.2 Unit Testing:

UNIT TESTING is a process of software testing where single units/ components of a software are tested. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output.

**Test Name: Admin Login**



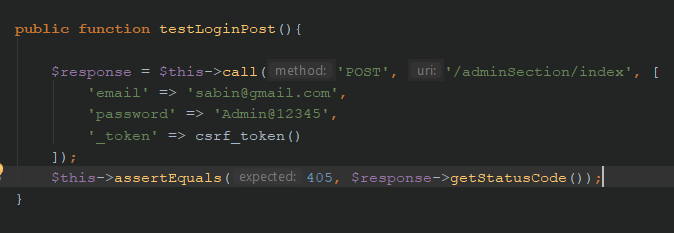


Figure 25: Testing login for admin

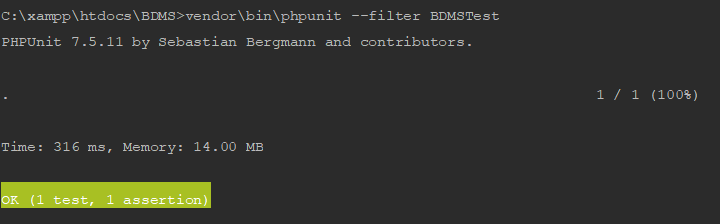


Figure 26: Result of admin login test case

**Test Name: Inserting Donor information**

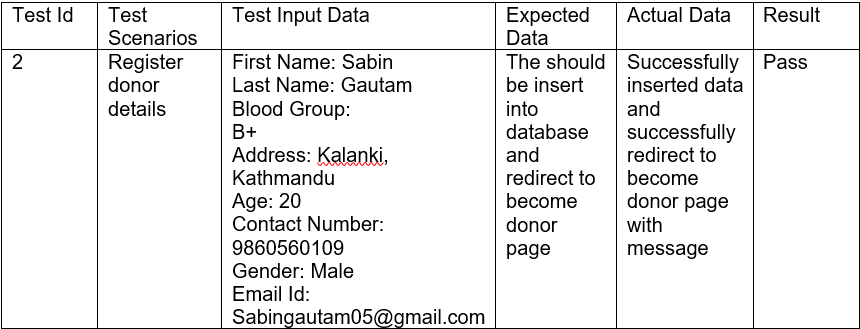


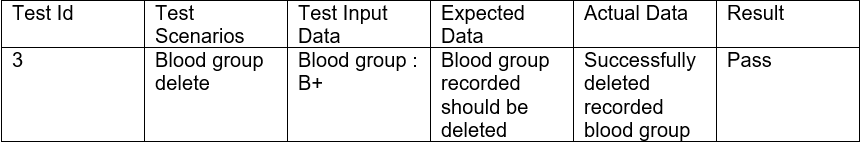


Figure 27: Registering donor info



Figure 28: result of register donor after testing

**Test Name: deleting blood group**



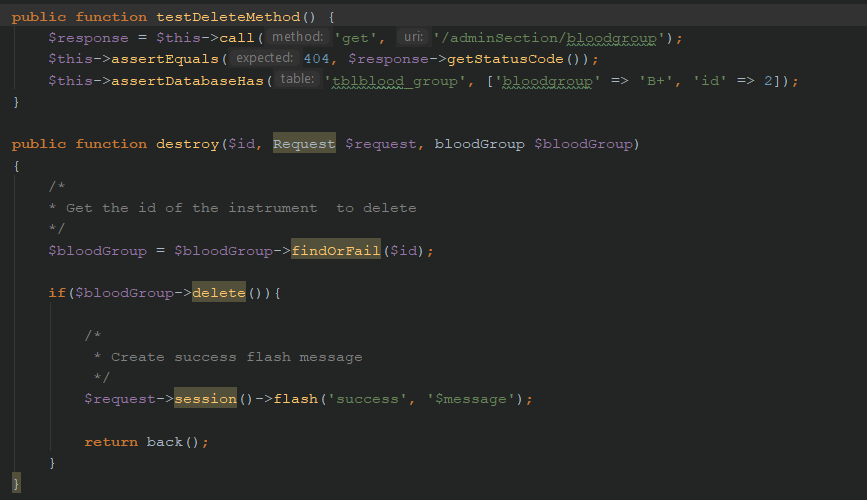


Figure 29: unit testing for deleting blood group record

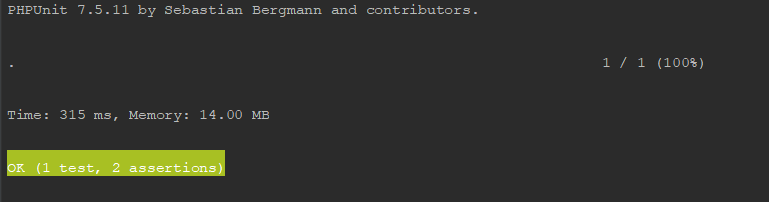
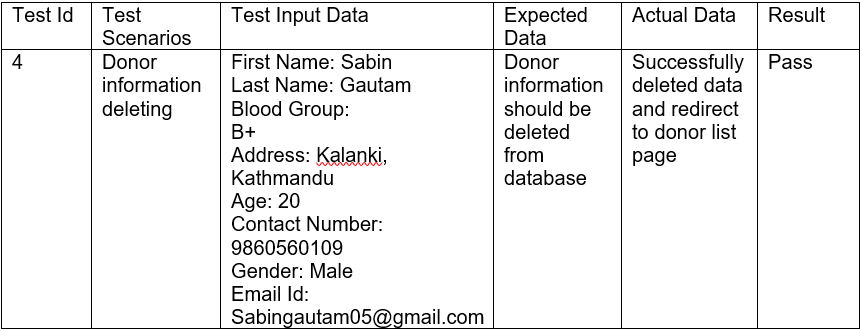


Figure 30: result after deleting blood group record in unit testing

**Test Name: deleting donor info**



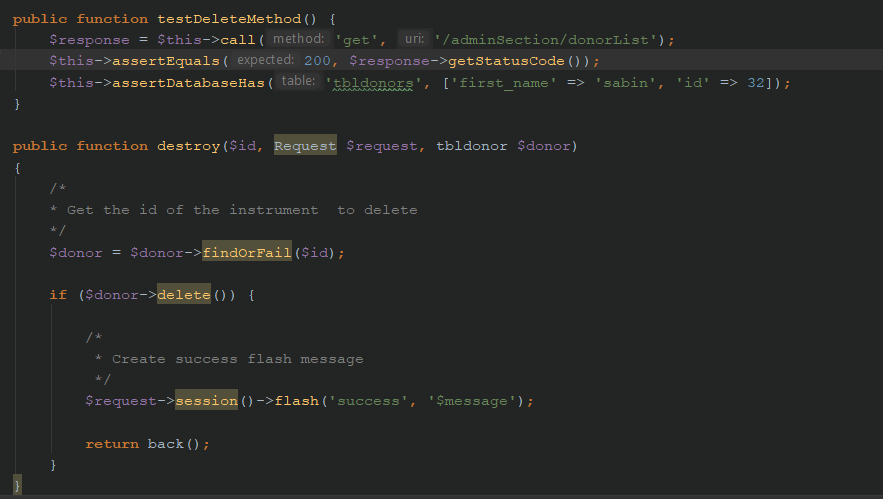


Figure 31: unit testing for deleting donor info

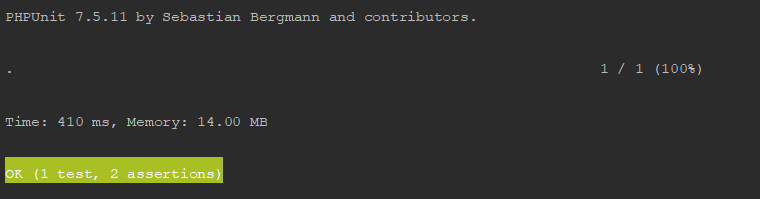
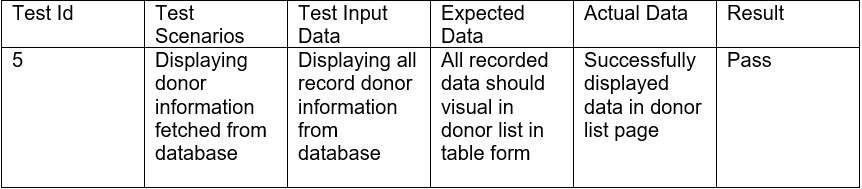


Figure 32: result after deleting donor info record in unit testing

**Test Name: Displaying Donor Information**



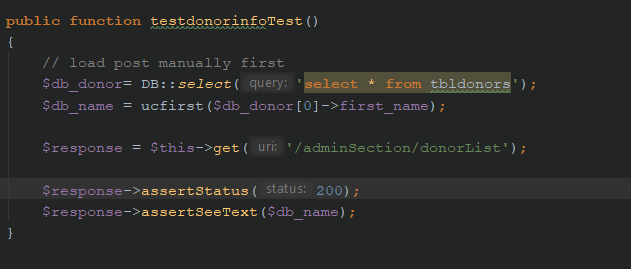


Figure 33: displaying recorded donor information

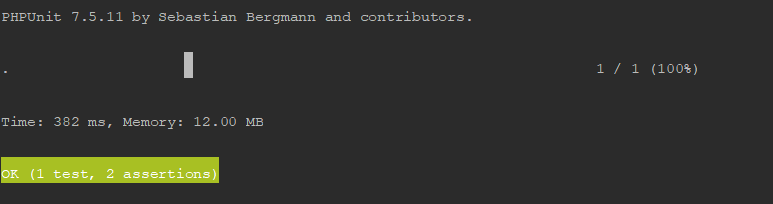
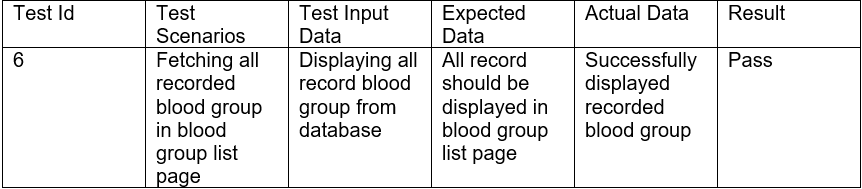


Figure 34: unit testing result to show recorded data of donor

**Test Name: displaying recorded blood group**



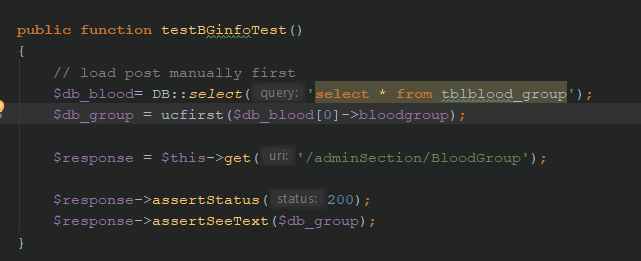


Figure 35: Unit testing to show recorded blood group

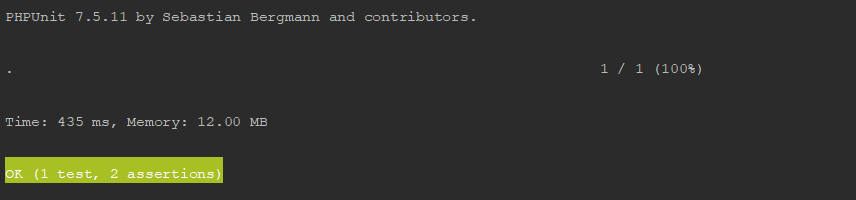
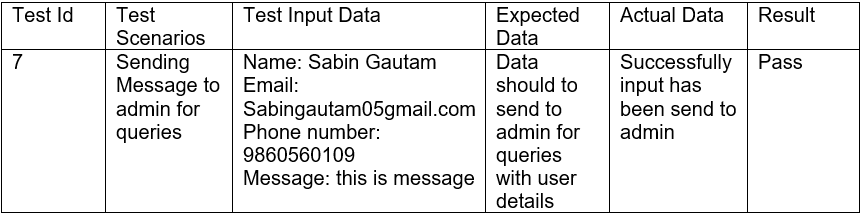


Figure 36: Result of unit testing for blood group

**Test Name: Sending Message**



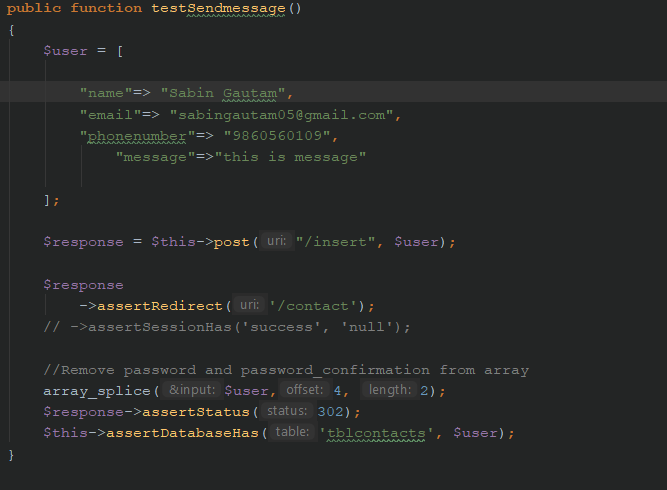


Figure 37: Unit test for sending message

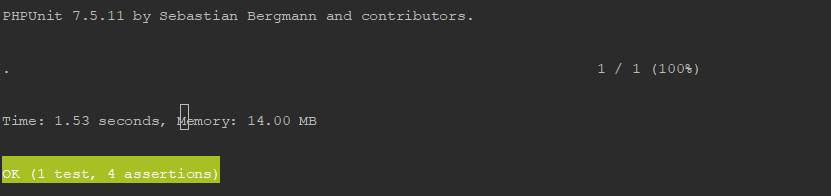
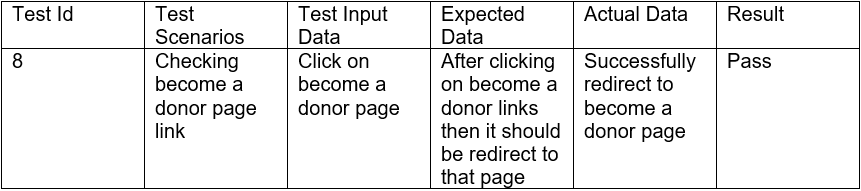


Figure 38: results of unit testing for sending message

**Test Name: Checking Link to become a donor Page**



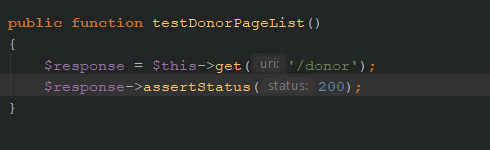


Figure 39: Unit testing for become a donor page link

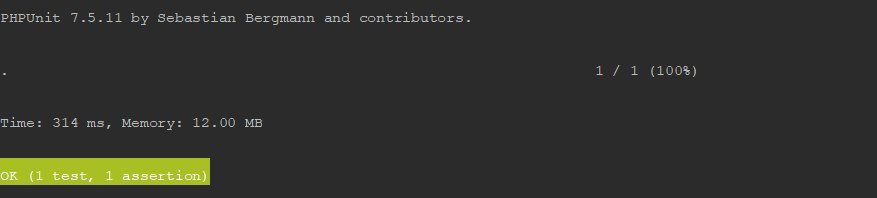
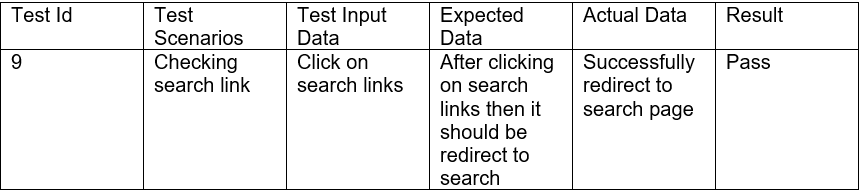


Figure 40: Result of unit testing after clicking on become a donor

**Test Name: Checking search links**



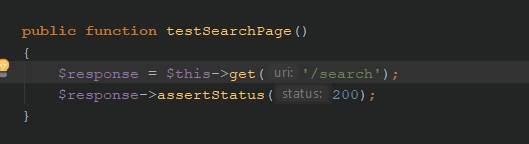


Figure 41: Unit testing for Search page link

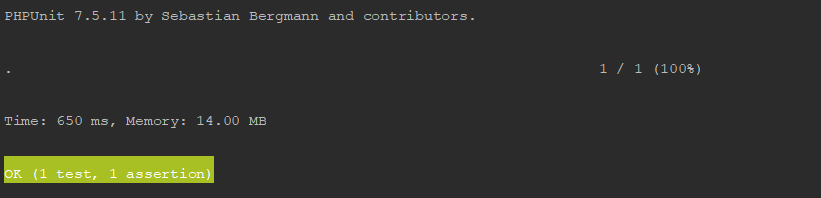


Figure 42: Result of unit testing after clicking on Search page

**Test Name: Checking About Us page**

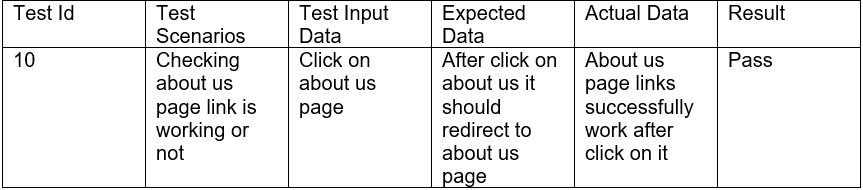




Figure 43: Unit testing for about us page link

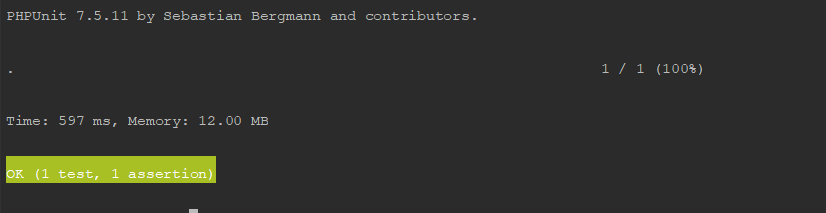


Figure 44: Result of unit testing after clicking on About us page

# Chapter 6: Other Project issues

## 6.1 Risk Management

While doing this project I haves faces many problem and by tackling them I have move ahead. Managing this risk is tough challenge for programmer. The risk I have faced while completing this project are mentioned below:

**Impact = Likelihood \* Consequence**

Risk Likelihood values are shown as follows

|  |  |
| --- | --- |
| **Likelihood** | **Value** |
| Low | 1 |
| Medium | 2 |
| High | 3 |

Risk Consequence values are shown below

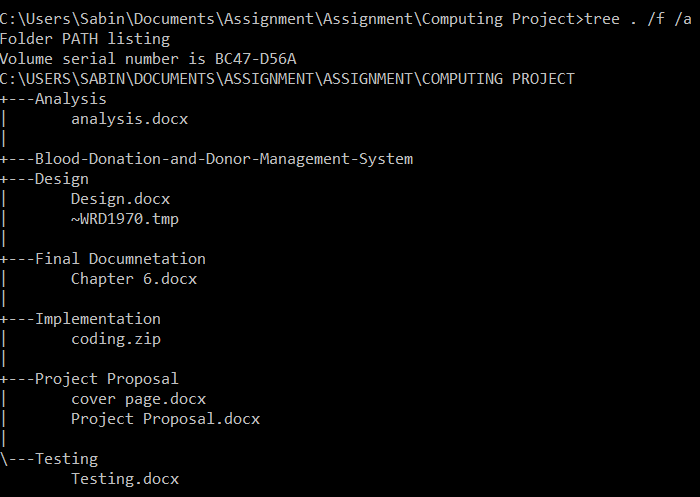
|  |  |
| --- | --- |
| **Consequence** | **Value** |
| Very low | 1 |
| Low | 2 |
| Medium | 3 |
| High | 4 |
| Very High | 5 |

Risk Consequence values are shown below

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S. NO.** | **Risks** | Likelihood | Consequences | Impact | **Solution** |
| 1 | Network system down | 2 | 3 | 6 | While doing this project I have face the problem of internet so that I have change the ISP Services and could not face the problem of Network System Down. |
| 2 | Not backing up the coding files | 2 | 4 | 8 | I have forgotten to take the back up of coding files time to time so to overcome this problem I have GIT hub version controller to recover my files. |
| 3 | Hard drive crashes | 2 | 3 | 6 | While Doing this project my hard drive has been crashed. Before crashing I have taken backup of project files and uploaded it to google drive so that I can recover it. |
| 4 | Not planned resources | 1 | 1 | 1 | Before doing this project there is not managed any resources to complete this project. I have managed project whatever available resources I have found for this project. |
| 5 | No advanced technology available or the existing technology is in initial stages. | 2 | 2 | 4 | In this project there no any advanced tool has been used. I have done this project what tools I have available. |

## 6.2 Configuration Management

The process of maintaining computer system and software in the automated method in a consistent state is known configuration management. The main advantages of configuration management is maintaining consistency system. The configuration management for my project is mentioned below:



## 6.3 Project Issues

Project issues are the problem which give error while perform the project. This issues makes so many errors in the project. In my project also I have faced so many problem which makes so many error in my project. The issues I have faced in my project is syntax error which makes my application in not workable. There is problem from the interface. Interface might not at correct place. Sometimes div error will occur. This error in caused due to the width and height and position in CSS I have create in my application. This all problem have been solved by correcting the div size with the body and all position is corrected by adjusting the length required.

Other project issues I have faced while doing this project is electricity problem from Nepal Electricity Authority. Due to heavy rainfall, there is the problem of electricity in my locality. To overcome this problem I have used alternative power supply for my project. There is problem in my system due to excessive use of the system. Also my IDE license also expired while doing this project. Also there in problem while I am testing my application. While testing the application due to slow performance of my system I have to wait for a bit of time. While testing, I have problem in login testing. PHP laravel tested the auto generate login form but in my application I have used custom login form. Which means I have forgotten to use authentication. It means system will not secured from user also. So at least have inserted guard for my system.

## 6.4 Limitation

A limitation is a restriction executed by the application of (mostly traditional) Project Management. There might be also some limitation on my project which are mentioned below:

* This is web based application so it cannot used without internet.
* This application cannot be used until user get internet connection.
* This is web based application so it will required sever also.
* In this application data are less protected because there no SSL enforcement to secure HTTP.
* Security concerns about sensitive private data being transmitted over the Internet.
* This application will run at a slightly slower speed than one hosted on a server locally.
* Unfortunately, we don’t all use the same browser. This means during development you’ll need to ensure your app is supported across a variety of browsers.

## 6.5 Future Works

Future works means adding feature in the future as an update version so that user also might also feel too easy to use. After completing the Blood Donor and Donation Management system there are some work left to do. The work that are left to complete are mentioned below:

* User detail can be hidden functions.
* Message Queries send to admin has read or pending function.
* User details can be edit or removed by user function.
* In future development, mobile app for Blood Donor and Donation management system will be developed.

# Conclusion

After all the step I have performed above, at last I conclude my project. After completing this project I have mentioned risk that arise while doing this project.. Also I have some project issues occurred while doing this project. There are some limitation also in my project. There are some works left to complete in the time so I have mention those work in future works. Finally, I have completed my project by performing all the steps as mentioned above.

Finally, project is complete by performing all the steps. This is the last concluded part of this project. I have complete all the phase required for this project. Like Requirement analysis, design, implementation, testing, documentation. I have water fall model to complete this project. At first, all requirement has been fulfilled in the analysis phase of this methodology. After then, in design phase all required diagram has been designed which includes class diagram, activity diagram as well as sequence diagram. After completing the design phase, implementation phase will come and in this phase all coding part will be done and in this phase define the MVC pattern of this project. After completing this phase, we proceeds for testing phase and in this phase all the code will be tested. Finally the project ready to release and at last the final documentation will be made. This how my whole project is completed.