****

**University of Medical Science & Technology**

**Faculty of Computer Science & Information Technology**

**Batch 13 – CS - (2016/2017)**

**BBMS**

**(BLOOD BANK MANAGEMENT SYSTEM)**

**Prepared by:**

* **Ahmed Mohammed Baha Eldin.**
* **Mohammed Mahmoud Mohammed.**

**Under Supervision:**

* **Mr. Mohammed Izzeldin.**

**Saturday, 1st June, 2017**

**Acknowledgement**

First and before most, we greet you in the name of his majesty, Allah almighty the most beneficent and the most merciful, our creator, for giving us the strength, knowledge, and vision and complete this project. We dedicate this work to everyone who helped us through getting it done.

Special thanks to Mr. Mohammed Izzelddin for being a source of guidance, correcting our faults and for giving us the needed confidence. And to all the teachers of the university who helped us by any means.

Finally, we would like to thank our families and friends for the support we got.

Ahmed Mohammed Baha Eldin (CS-2013-021)

Mohammed Mahmoud Mohammed (CS-2013-008)

Saturday, 1st June, 2017

**Abstract**

Blood bank management system is a web-based bank of blood, which helps to assist the organization of donor information, reducing the risk of inaccurate results, and switch paper-works of blood banks to computerized systems. It gives all the answers to the problems in the current working systems, hence improved service quality, flexibility, security, and faster access time.

The blood bank management system includes having new donor registration through simple and easy forms to maintain the information for future uses.

The web based blood bank management system reflects more efficient utilization of blood details along with data accuracy and tracking of blood transactions.

**Table of Contents**

|  |  |
| --- | --- |
| Acknowledgement | 2 |
| Abstract | 3 |

**Chapter 1 (Introduction)**

|  |  |
| --- | --- |
| 10 | 1.1 Introduction |
| 10 | 1.2 Problem Statement |
| 10 | 1.3 Project Objectives |
| 11 | 1.4 Project Methodology |
| 11 | 1.5 Scope of Project |

**Chapter 2 (Literature Review)**

|  |  |
| --- | --- |
| 13 | 2.1 Introduction |
| 15 | 2.2 History of Blood banking |
| 16 | 2.3 Current System |
| 16 | 2.4 Proposed System |
| 16 | 2.5 Summary |

**Chapter 3 (Analysis & Design)**

|  |  |
| --- | --- |
| 18 | 3.1 Introduction |
| 19 | 3.2 Requirements Gathering |
| 20 | 3.2.1 Functional Requirements |
| 20 | 3.2.2 Non-Functional Requirements |
| 21 | 3.3 UML Diagrams |
| 21 | 3.3.1 Use Case Diagrams |
| 32 | 3.3.2 Sequence Diagrams |
| 40 | 3.3.3 Activity Diagram |
| 41 | 3.3.4 Entity Diagram |
| 42 | 3.3.5 Class Diagram |
| 43 | 3.3.6 Database Schema |

**Chapter 4 (Implementation)**

|  |  |
| --- | --- |
| 45 | 4.1 Introduction |
| 45 | 4.2 Screenshots of the System |
| 56 | 4.3 Database tables |

**Chapter 5 (Conclusion, Recommendation& Future Works)**

|  |  |
| --- | --- |
| 66 | 5.1 Conclusion |
| 66 | 5.2 Recommendation |
| 67 | 5.3 Future Works |

|  |  |
| --- | --- |
| **References** | 68 |

**List of Figures**

|  |  |
| --- | --- |
| Figure 3.1- Problems in the National Public Health Laboratory | 19 |
| Figure 3.2 - Admin Use Case Diagram | 21 |
| Figure 3.3 - User Use Case Diagram | 24 |
| Figure 3.4 - Doctor Use Case Diagram | 27 |
| Figure 3.5 - System Use Case Diagram | 29 |
| Figure 3.6 - Login Sequence Diagram | 32 |
| Figure 3.7 - Manage Accounts (Add Account) Sequence Diagram | 33 |
| Figure 3.8 - Manage Accounts (Delete Account) Sequence Diagram | 34 |
| Figure 3.9 - Register and Enter Blood Information Sequence Diagram | 35 |
| Figure 3.10 - Check Login, Check Existing Donor, and Store Blood Sequence | 36 |
| Figure 3.11 - Fill Medical Information and Enter Virus Information Sequence | 37 |
| Figure 3.12 - Enter outgoing Blood Information Sequence Diagram | 38 |
| Figure 3.13 - Receive Reports Sequence Diagram | 39 |
| Figure 3.14 - Activity Diagram | 40 |
| Figure 3.15 - Entity Relationship Diagram | 41 |
| Figure 3.16 - Class Diagram | 42 |
| Figure 3.17 - Database Schema | 43 |
| Figure 4.1 - Create User Screenshot | 45 |
| Figure 4.2 - Login Screenshot | 46 |
| Figure 4.3 - Add Donor Screenshot | 47 |
| Figure 4.4 - Donor Details Screenshot | 48 |
| Figure 4.5 - Edit Donor Screenshot | 49 |
| Figure 4.6 - Applied Donor Screenshot | 50 |
| Figure 4.7 - Questions Form Screenshot | 51 |
| Figure 4.8 - Today’s Donor Screenshot | 52 |
| Figure 4.9 - Collect Blood Screenshot | 53 |
| Figure 4.10 - Collected Blood Screenshot | 54 |
| Figure 4.11 - Edit Donor Screenshot | 55 |
| Figure 4.12 - Donor table | 56 |
| Figure 4.13 - Donor Info table | 57 |
| Figure 4.14 - Collected Blood table | 58 |
| Figure 4.15 - Blood Type table | 59 |
| Figure 4.16 - Virus table | 60 |
| Figure 4.17 - Blood Stock table | 61 |
| Figure 4.18 - Incoming Blood table | 62 |
| Figure 4.19 - Outgoing Blood table | 62 |
| Figure 4.20 - Patient table | 63 |
| Figure 4.21 - Patient Relation table | 63 |
| Figure 4.22 - User table | 64 |
| Figure 4.23 - User Type table | 64 |

**List of Tables**

|  |  |
| --- | --- |
| 14 | Table 2.1 - Compatibility Between Blood Types |
| 22 | Table 3.1 - Description of Admin to Manage Accounts Use Case |
| 22 | Table 3.2 - Description of Admin Enter Medical Information Use Case |
| 22 | Table 3.3 - Description of Admin to Enter Collected Blood Use Case |
| 23 | Table 3.4 - Description of Admin to Check Virus Use Case |
| 23 | Table 3.5 - Description of Admin to Receive Reports Use Case |
| 23 | Table 3.6 - Description of Admin to Register Use Case |
| 25 | Table 3.7 - Description of User to Enter Outgoing Blood Use Case |
| 25 | Table 3.8 - Description of User to Register Use Case |
| 25 | Table 3.9 - Description of User to Receive Reports Use Case |
| 26 | Table 3.10 - Description of User to Enter Collected Blood Use Case |
| 28 | Table 3.11 - Description of Doctor to Receive Reports Use Case |
| 28 | Table 3.12 - Description of Doctor to Enter Medical Information Use Case |
| 28 | Table 3.13 - Description of Doctor to Check Virus Use Case |
| 30 | Table 3.14 - Description of System to Store Blood Use Case |
| 30 | Table 3.15 - Description of System to Check Login Use Case |
| 30 | Table 3.16 - Description of System to Check Existing Donor Use Case |
| 31 | Table 3.17 - Description of System to Generate Reports Use Case |

**List of Abbreviations**

|  |  |
| --- | --- |
| **UMST** | University of Medical Science & Technology |
| **BBMS** | Blood Bank Management System |
| **UML** | Unified Modelling Language |
| **HIV** | Human Immunodeficiency Virus |
| **VDRL** | Venereal Disease Research Laboratory |
| **HCV** | Hepatitis C Virus |
| **HBV** | Hepatitis B Virus |
|  |  |
|  |  |

**CHAPTER 1**

**(Introduction)**

**1.1 Introduction**

The term “Blood Bank” means the segments of a clinic labs where the testing and storage of blood occurs. Blood bank management systemis a web-based application designed to store, analyze, process and retrieve information about inventory within a blood bank. it helps to save all the donators information. Also through it we can store and maintain the transactions made in blood bank.

**1.2 Problem Statement:**

The main facing struggle in this field is the lack of computerized management systems to facilitate the processes and categories of available blood types.

The paper-based system leads to faults in produced reports due to papers losses. Also paper works can be time consuming. Computerized systems can be more effective in both time and results.

**1.3 Project Objectives:**

* To Reduce the paper works concerned with blood banks.
* To develop a well-numbered record of blood categories.
* To develop a web based application that manipulates the need of supply or demand of blood.
* To maintain the accuracy and consistency of the data used.

**1.4 Project Methodology:**

Blood bank management system will be designed and implemented using Microsoft’s Visual Studio and ASP.Net forms and the .NET Framework that is developed and maintained using the C# programming language.

SQL Server Express 2014 database server as a backend is used to create and maintain the logical infrastructure of the database and also evaluate requests on the server side and then returns data to the client. Users need only a browser, the systems front-end will be developed using visual studio 2013 and C Sharp programming language of .NET Framework, which is a high level language that is characterized to be flexible and objected oriented.

**1.5 Scope of Project:**

The blood bank management system gives maximum care of donors and all their information, and enabling new registrations to save information for future use. The system supports and improves the tracking of (in & out) blood transaction. And as it is web-based, it can be used in any computer device.

This application will be created within the University of Medical Science and Technology in the bachelor’s degree.

**CHAPTER 2**

**(Literature Review)**

**Chapter 2**

**Literature Review**

**2.1 Introduction**

Blood is the liquid pumped through the body by the heart through blood vessels. It helps clear the waste of the body and it delivers all the important materials to the right organs. Blood asset for surviving different situations. Therefore, donation of blood is important for human beings to save their species.

Blood consists of 40% red cells, and 60% plasma, the plasma is approximately about 90% water but it also has protein, and hormones.

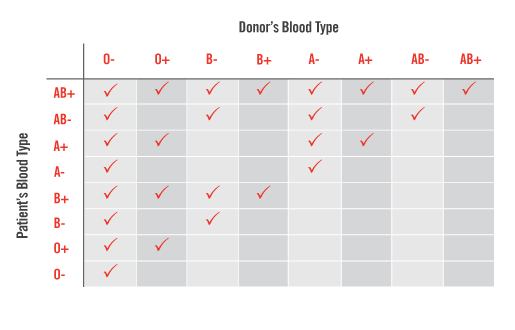
The blood group in every human is identified by the antigens and the antibodies in the blood, the antibodies are basically the protection in the human body, and the antigens are protein molecules found on the blood cells. Scientists have made a distinct different between these molecules, they labelled one molecule “A” and the other molecule “B”, there are four blood groups identified by the antigens (A & B) on the red cells:

1. “A” group – has only A antigen on the red cells.
2. “B” group – has only B Antigens on the red cells.
3. “AB” group – has both A and B antigens on the red cells.
4. “O” group – has neither A nor B antigens on the red cells.

Another protein is found on the blood cells called “Rh” which differentiate the blood type to negative (+) and positive (-). This means we get eight types of blood:

* A+
* A-
* B+
* B-
* AB+
* AB-
* O+
* O-

The blood type in the human system determines who you can receive blood from, and who you can donate to. The table below explains the compatibility of blood types between the donors and the patient(Receiver):

**Table 2.1 - Compatibility Between Blood Types**

Online blood bank management system for hospitals and clinic labs, describes the development of blood bank management system to assist managing all blood records and control the distribution of various blood types based on patient’s demand. The organization of blood bank services should receive the most care and attention for smooth, better and easier living.

In the blood bank management system module, donors get to sign up, then the donor will be asked be complete a form which includes various type of information needed by system. After the forms are filled the doctor or the employee performs some tests and examinations on the donor, such as reading the blood pressure, temperature, and finally testing a blood sample from the donor. After the tests are done, all information is entered to the system, then the system decides whether the donor is accepted or not. if the blood is rejected by the system, the user will be rejected as a donor, and if the blood is accepted, the blood will be taken and put a plastic bag designed for storage of blood. This plastic bag will be given a unique index or a unique ID to identify the blood and the blood donor.

Blood bank management system takes care of the security, data protection mechanism, different modules and their associated reports, which are generated upon specific strategies that are put by the project team.

**2.2 History of Blood Banking:**

Blood giving has always been common act since the early 1800’s. the passion of taking care of sick, however, blood meant to be directed (vein to vein). James Blundell applied the first successful blood transfusion in 1818[1]. but this practice was often disastrous because there were no information and knowledge of blood, blood types and compatibility between blood type owners.

And the first document of the blood types was done in 1901 by an Austrian physician, Karl Landsteiner, without this research there would be no blood banking [2].

Landsteiner discovered that when two individual incompatible types are mixed, an immune response is triggered and the red blood cells clump. The reaction occurs when the receiver of a blood transfusion has antibodies against the donor blood cells. Landsteiner's researches made it possible to identify blood type and allowed a way for blood transfusions to be carried out much more safely. For this discovery he was awarded the Nobel Prize in Physiology and Medicine in 1930.

Following the year 1930, Vladimir Shamov &Sergei Yudin took an important step to that blood can be shortly stored (canned blood) in refrigerated blood centers. After that, the Hungarian physician, Bernard Fantus, knew the importance of blood storing, learnt that storing blood isn’t just developing a sort mechanism in the laboratory, he spent many years working on the blood transfusion for longer period until the blood bank officially opened in March 15, 1937 [3].

**2.3 Current System (The National Public Health Labs)**

The National Public Health Laboratory located in Khartoum-Sudan, the system of the health labs is a paper-based system which means more time, and more efforts on completing the transaction forms.

This system faces a lot of problems, but the highest risk is that there is 60% probability of losing a form associated with a blood bag. Therefore, retrieving the report comes with an error.

**2.4 Proposed System:**

The blood bank management system is designed to be flexible, error free, secure and it will operate in a manner that will avoid problems in the current system, such as:

* Loss of data.
* Time consuming.
* Difficulties in generating the reports.
* Lack of consistency.
* Lack of organized databases.

The blood bank management system involves donor registration, evaluation, blood collecting and storing. Keeping track of all blood forms.

The desired system will be used through a friendly user interface.

**2.5 Summary**

Blood bank management system aims to save time, keep track of blood, and improve the processes and the transactions of blood. The system will be able to store, process, retrieve reports of exchange in blood.

**CHAPTER 3**

**(Analysis & Design)**

**Chapter 3**

**Analysis & Design**

**3.1 Methodology Introduction:**

The technology which will be used in the front-end of the application will be the visual studio 2013, using C# programming language, which is a high level programming language used for creating web sites, applications for Microsoft Windows, and web applications.

Microsoft SQL server Express 2014 is the back-end in the application, an RDBMS designed by the Microsoft corporation, used to create and maintain all the information and procedures related to the database, and respond to requests sent by user and then return data to the client.

The model used to finish the project is the Waterfall model**,** which means work and tasks should be going in a sequence(pre-planned), task A is finished before starting task B and so on.

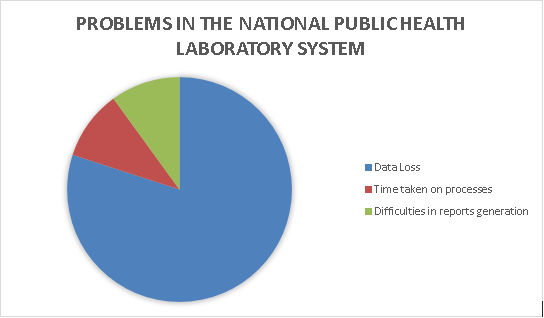
The Waterfall model consists of several phases:

1. Definition. (understanding what are the requirements of the system).
2. Analysis. (analyzing the requirements).
3. Design. (this phase is concerned with detailed definition of the inputs, outputs of the system).
4. Coding. (system is coded in the programming language).
5. Testing. (after the coding is done, this process is done is fetch any system errors).
6. Operation and Maintenance. (the operation begins if the application is on the client site. Maintenance is required to keep the application running).

**3.2 Requirement Gathering:**

Interviews and questionnaire were done in National Public Health laboratory in Khartoum-Sudan.

The requirements were gathered by interviewing the manager of the National Public Health Laboratory, two nurses, and one receptionist.

****

**Figure 3.1 - Problems in the National Public Health Laboratory System**

**3.2.2 Functional Requirements:**

They are the requirements that describes the functionality of the system, **WHAT** the system should do, they are statements of services the system should provide how the system should react to particular inputs and how the system should behave in particular situations [4].

* The system will give the right authorization according to the user (login site).
* The system will check if the user is new or existing.
* The system will provide a registration site for donors.
* The system will enable the user to enter blood details.
* The system will provide a model for the donating information (donor, age, gender, and blood type).
* The system will allow the user to check the availability of blood.
* The system will choose if the donor’s blood is eligible or not.
* The system will provide reports generation.
* The system will provide functions such as printing, previewing, and searching.
  + 1. **Non-Functional Requirements:**

They specify the attributes of the system, **How** the system should do a certain function, they are constraints on the service or functions offered by the system, they include timing constrains, constrains of the development process and standards [5].

* The system will deny any given wrong details on the login site.
* The system will be easily portable.
* The system will be available at any time.
* The system will have an understandable friendly user interface.
* The system will take no much time in response; providing and previewing information will be technically fast.
* The general administrative functions of the system will be easy to use.
* The system will handle concurrent users.
* The system will be well protected and secured.
  1. **UML Diagrams:**

UML is the Unified Modeling Language, used to represent systems in a graphical way. The below Components of the UML diagrams explains the flow of data and entities of the blood bank management system.

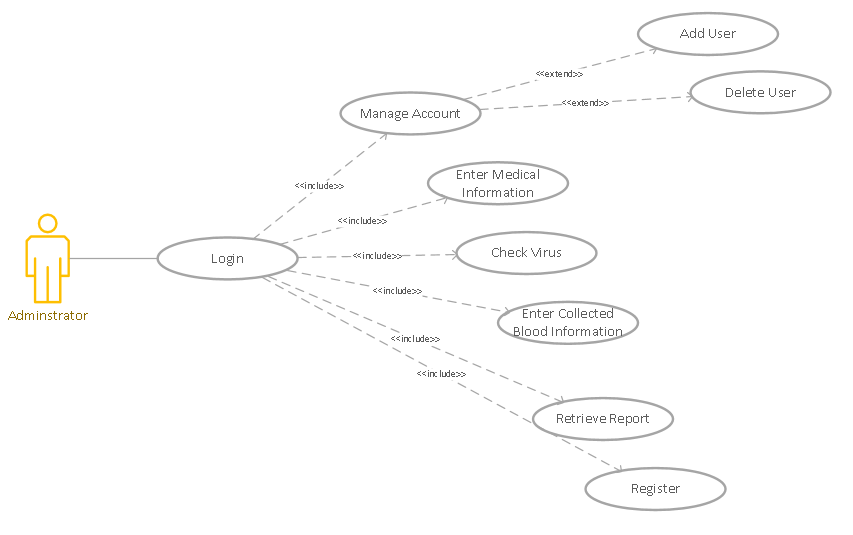
* + 1. **Use Case Diagrams:**

Use case diagrams consist of actors, use cases and their relationships.

The diagram is used to show the system/subsystem of an application. A single use case diagram captures a particular functionality of a system [6].

* + - 1. **Administrator Use Case:**

The administrator is the user who has full access to the system and can perform almost every functional requirement of the system.

**Figure 3.2 - Administrator Use case**

|  |  |
| --- | --- |
| **Use case** | **Manage Accounts (Add, Delete)** |
| **Actors** | Admin |
| **Type** | Primary and essential |
| **Description** | * Select, Add or delete User * Update Information * Save transaction |

**Table 3.1 -**Description of Admin to Manage Accounts

|  |  |
| --- | --- |
| **Use case** | **Enter Medical Information** |
| **Actors** | Admin |
| **Type** | Primary and essential |
| **Description** | * Select donor ID * Update Information * Save transaction |

**Table 3.2 -** Description of Admin to Enter Medical Information

|  |  |
| --- | --- |
| **Use case** | **Enter collected blood information** |
| **Actors** | Admin |
| **Type** | Primary and essential |
| **Description** | * Select blood ID * Update Information * Save transaction |

**Table 3.3 -** Description of Admin to Enter Collected Blood

|  |  |
| --- | --- |
| **Use case** | **Check Virus** |
| **Actors** | Admin |
| **Type** | Primary and essential |
| **Description** | * Select blood ID * Update Information * Save transaction |

**Table 3.4 -** Description of Admin to Check Virus

|  |  |
| --- | --- |
| **Use case** | **Retrieve reports** |
| **Actors** | Admin |
| **Type** | Primary and essential |
| **Description** | * Select report * Print report * Save transaction |

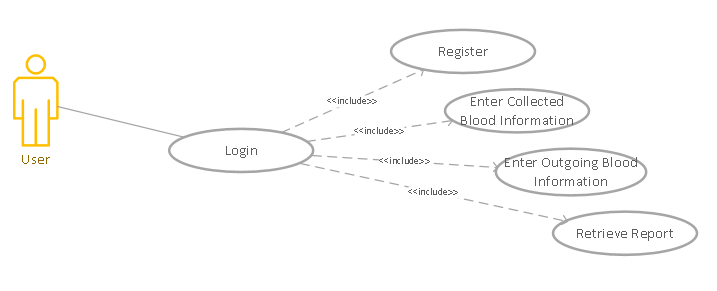
**Table 3.5 -** Description of Admin to Receive Reports

|  |  |
| --- | --- |
| **Use case** | **Register** |
| **Actors** | Admin |
| **Type** | Primary and essential |
| **Description** | * Select New Donor * Update Information * Save transaction |

**Table 3.6 -** Description of Admin to Register

* + - 1. **User Use Case Diagram:**

The user is responsible for registering and collecting donor’s blood information, also managing outgoing blood information.



**Figure 3.3 - User Use Case**

|  |  |
| --- | --- |
| **Use case** | **Enter outgoing blood information** |
| **Actors** | User |
| **Type** | Primary and essential |
| **Description** | * Select Blood ID * Update Information * Save transaction |

**Table 3.7 -** Description of User to Enter Outgoing Blood

|  |  |
| --- | --- |
| **Use case** | **Register** |
| **Actors** | User |
| **Type** | Primary and essential |
| **Description** | * Select New Donor * Update Information * Save transaction |

**Table 3.8 -** Description of User to Register New Donor

|  |  |
| --- | --- |
| **Use case** | **Retrieve reports** |
| **Actors** | User |
| **Type** | Primary and essential |
| **Description** | * Select report * Print report * Save transaction |

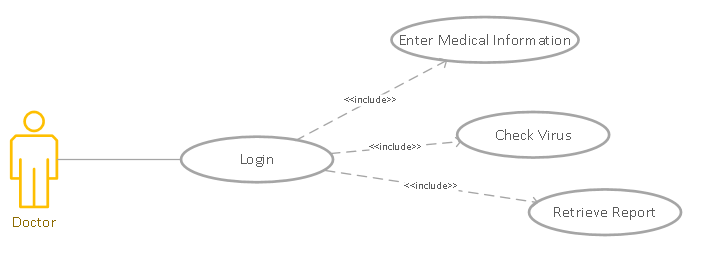
**Table 3.9 -** Description of User to Receive Reports

|  |  |
| --- | --- |
| **Use case** | **Enter collected blood information** |
| **Actors** | User |
| **Type** | Primary and essential |
| **Description** | * Select blood ID * Update Information * Save transaction |

**Table 3.10 -** Description of User to Enter Collected blood

* + - 1. **Doctor Use Case Diagram:**

The doctor is responsible of performing medical inquiries of the donor’s and taking a blood sample to check if the blood is infected with any virus.

**Figure 3.4 - Doctor Use Case**

|  |  |
| --- | --- |
| **Use case** | **Retrieve reports** |
| **Actors** | Doctor |
| **Type** | Primary and essential |
| **Description** | * Select report * Print report * Save transaction |

**Table 3.11 -** Description of Doctor to Receive Reports

|  |  |
| --- | --- |
| **Use case** | **Enter Medical Information** |
| **Actors** | Doctor |
| **Type** | Primary and essential |
| **Description** | * Select Donor ID * Update Information * Save transaction |

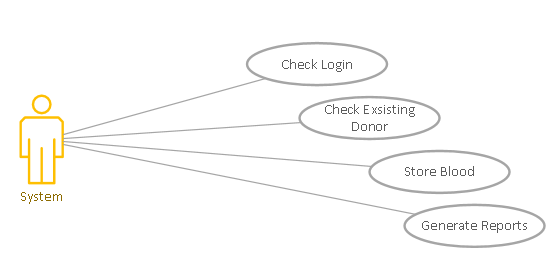
**Table 3.12 -** Description of Doctor to Enter Medical Information

|  |  |
| --- | --- |
| **Use case** | **Check Virus** |
| **Actors** | Doctor |
| **Type** | Primary and essential |
| **Description** | * Select blood ID * Update Information * Save transaction |

**Table 3.13 -** Description of Doctor to Check Virus

* + - 1. **System Use Case Diagram:**

The system stores all collected blood information and applies functions that helps the other users to communicate and generate reports.

**Figure 3.5 - System Use Case**

|  |  |
| --- | --- |
| **Use case** | **Store Blood** |
| **Actors** | System |
| **Type** | Primary and essential |
| **Description** | * Select Blood ID * Update Information * Save transaction |

**Table 3.14 -** Description of System to Store Blood

|  |  |
| --- | --- |
| **Use case** | **Check Login** |
| **Actors** | System |
| **Type** | Primary and essential |
| **Description** | * Attempt of login by user * Check data on Database * Return (success or failure) * Save |

**Table 3.15 -** Description of System to Check Login

|  |  |
| --- | --- |
| **Use case** | **Check Existing Donor’s** |
| **Actors** | System |
| **Type** | Primary and essential |
| **Description** | * Check existence of data on Database * Return (success or failure) * Save |

**Table 3.16 -** Description of System to Check Existing Donor

|  |  |
| --- | --- |
| **Use case** | **Generate Reports** |
| **Actors** | System |
| **Type** | Primary and essential |
| **Description** | * Select report * Print report * Save transaction |

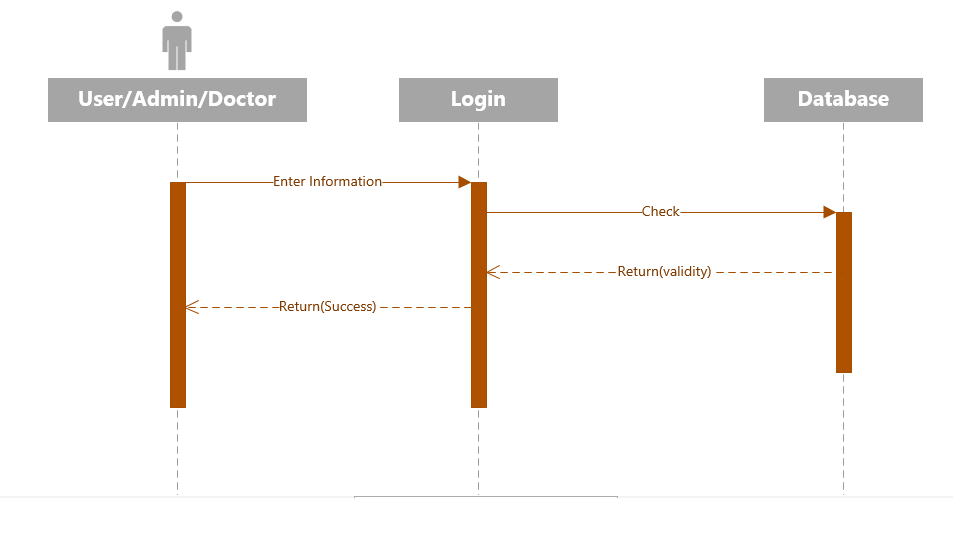
**Table 3.17 -** Description of System to Generate Reports

* + 1. **Sequence diagrams:**

Sequence diagrams are used to model how objects interact in a given scenario. An important characteristic of a sequence diagram is that time passes from top to bottom: the interaction starts near the top of the diagram and ends at the bottom [7].

* + - 1. **Login Sequence Diagram:**

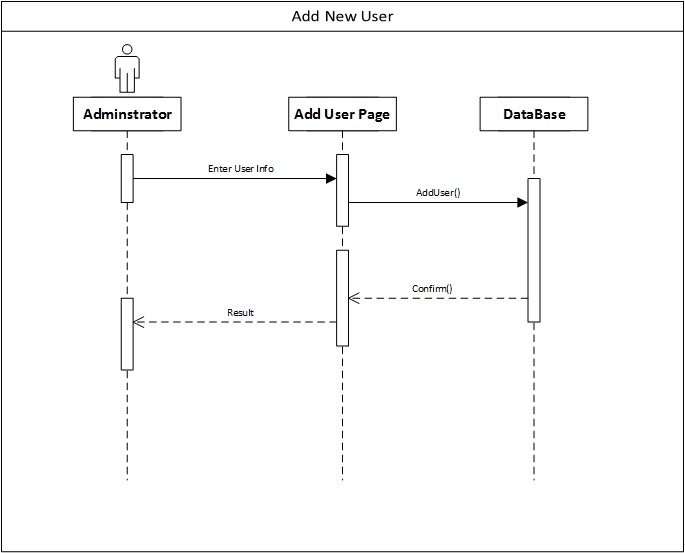
The user provides his/her login information, the system checks and validates the given information, then returns the results.



**Figure 3.6 - Login Sequence Diagram**

* + - 1. **Manage Account (Add Account) Sequence Diagram:**

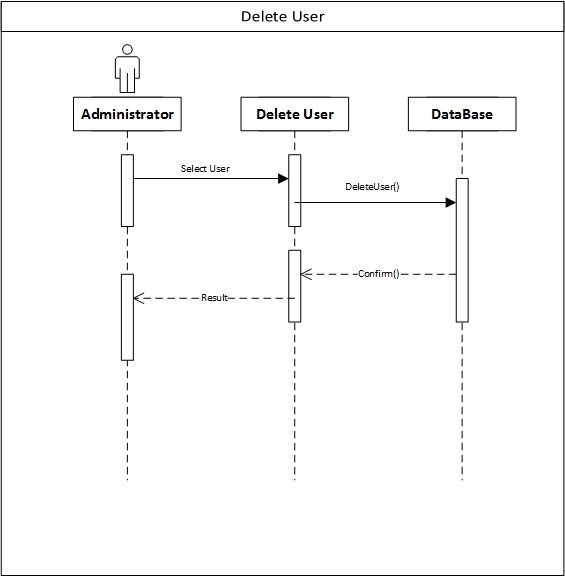
The administrator enters the information of the account and the account will be created.

****

**Figure 3.7 - Manage(Add) Account sequence**

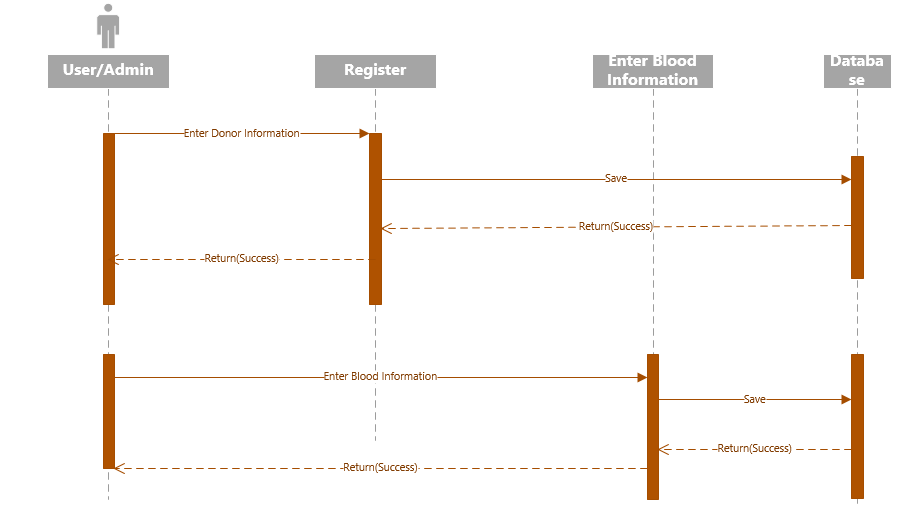
* + - 1. **Manage Account (Delete Account) Sequence Diagram:**

The administrator enters the information of the account and the selected account will be deleted(deactivated).

****

**Figure 3.8 - Manage(Delete) Account sequence**

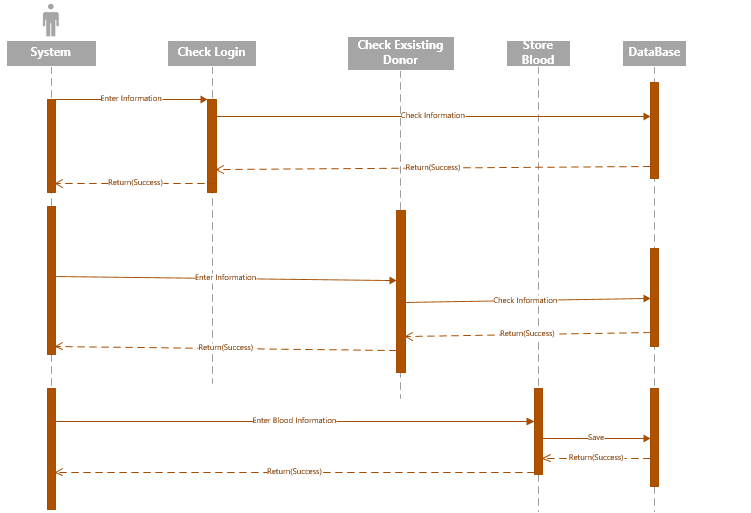
* + - 1. **Register and Enter blood information Sequence Diagram:**

****

**Figure 3.9 - Register and Enter blood information Sequence**

* + - 1. **Check Login, Check Existing Donor, and Store Blood Sequence Diagram:**

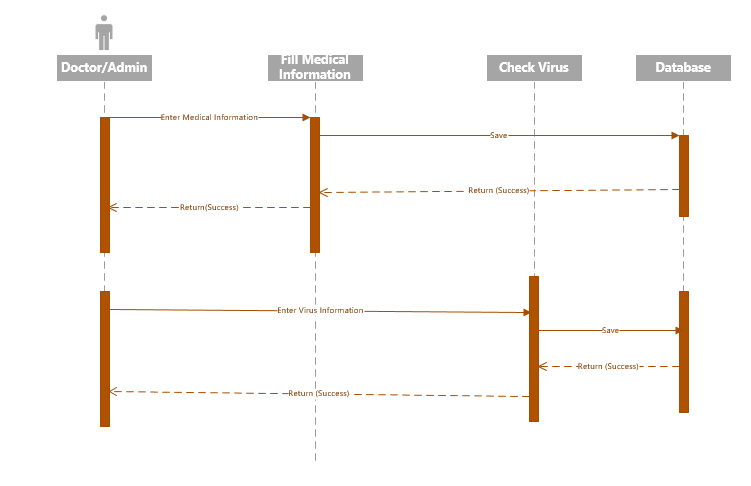
All these are functions done by the system; Checking if the login info is correct, checking for existing donors, and storing blood.

****

**Figure 3.10 - Check Login, Check Existing Donor, and Store Blood Sequence**

* + - 1. **Fill Medical Information and Enter Virus Information Sequence diagram:**

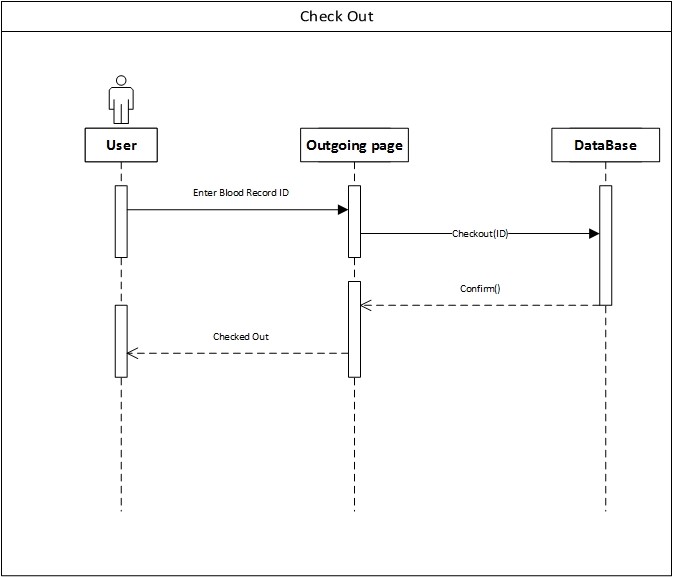
The user enters the information about the medical state of the potential donor.

****

**Figure 3.11 - Fill Medical Information and Enter Virus Information Sequence**

* + - 1. **Enter Outgoing Blood Information Sequence Diagram:**

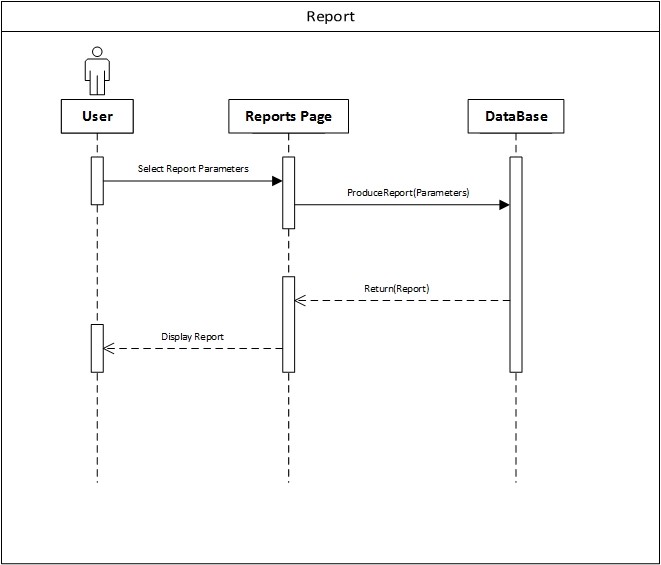
The user enters the desired information, checking the (out) transaction of the blood.

****

**Figure 3.12 - Enter Outgoing Blood Information Sequence**

* + - 1. **Retrieve Reports:**

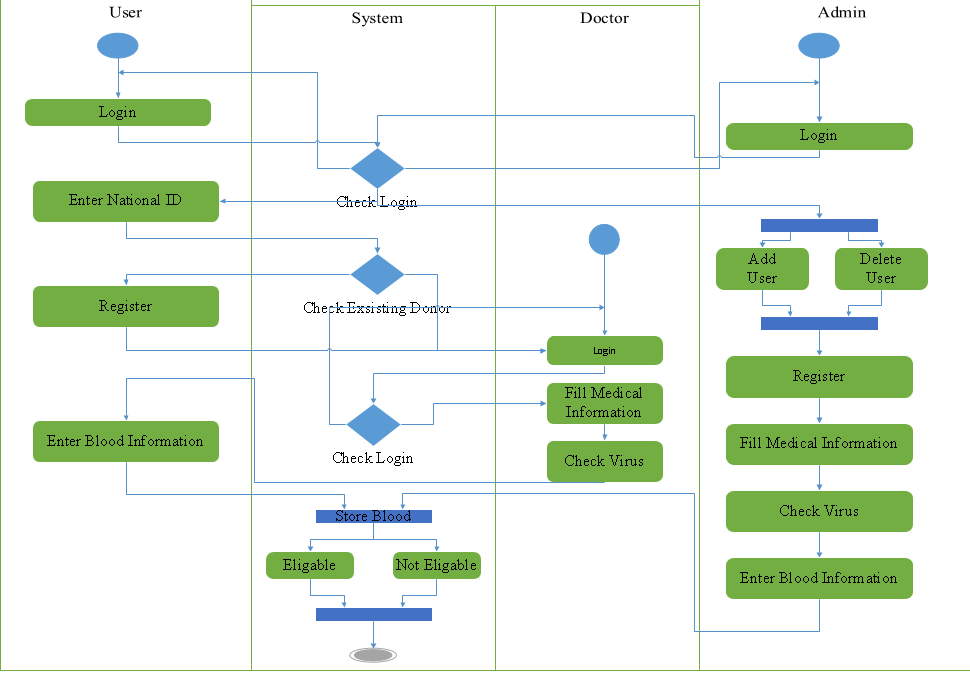
The user selects the report needed, the system will provide and display the report.

****

**Figure 3.13 - Receive Reports Sequence**

* + 1. **Activity Diagram:**

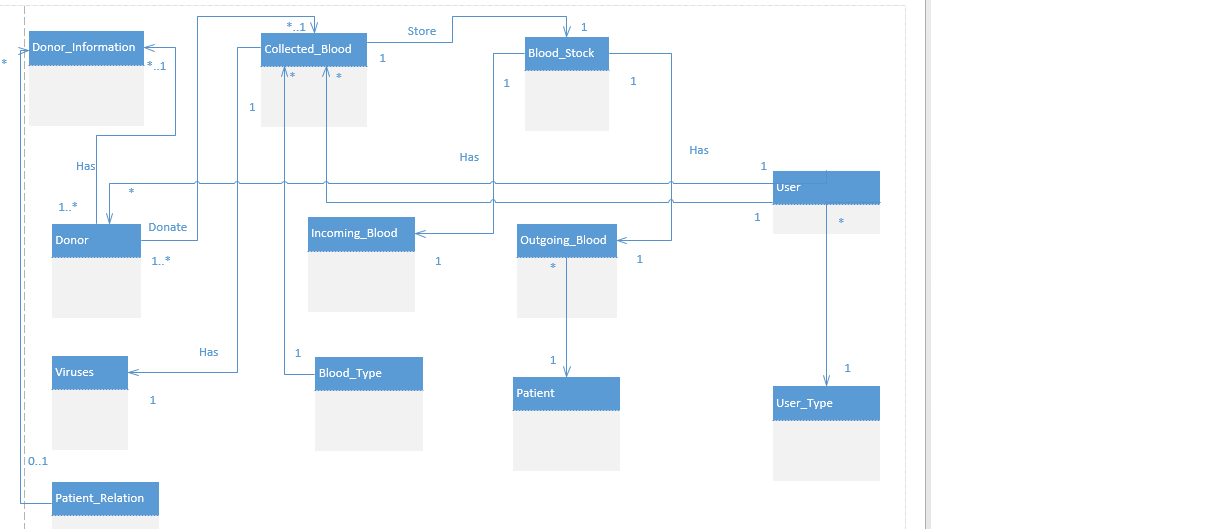
Activity diagram is a flowchart to represent the workflow from one activity to another activity. The activity can be described as an operation of the system. [8]

****

**Figure 3.14 - Activity Diagram**

* + 1. **Entity Relationship Diagram:**

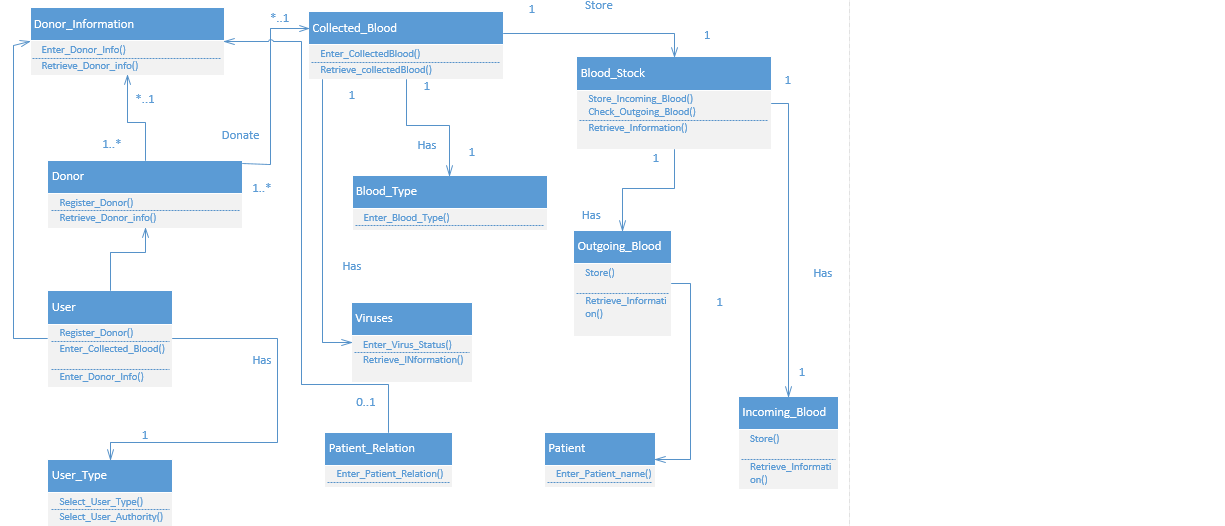
A diagram that helps represent entities, relationship sets and the attributes of relationship sets [9].

****

**Figure 3.15 - Entity Diagram**

* + 1. **Class Diagram:**

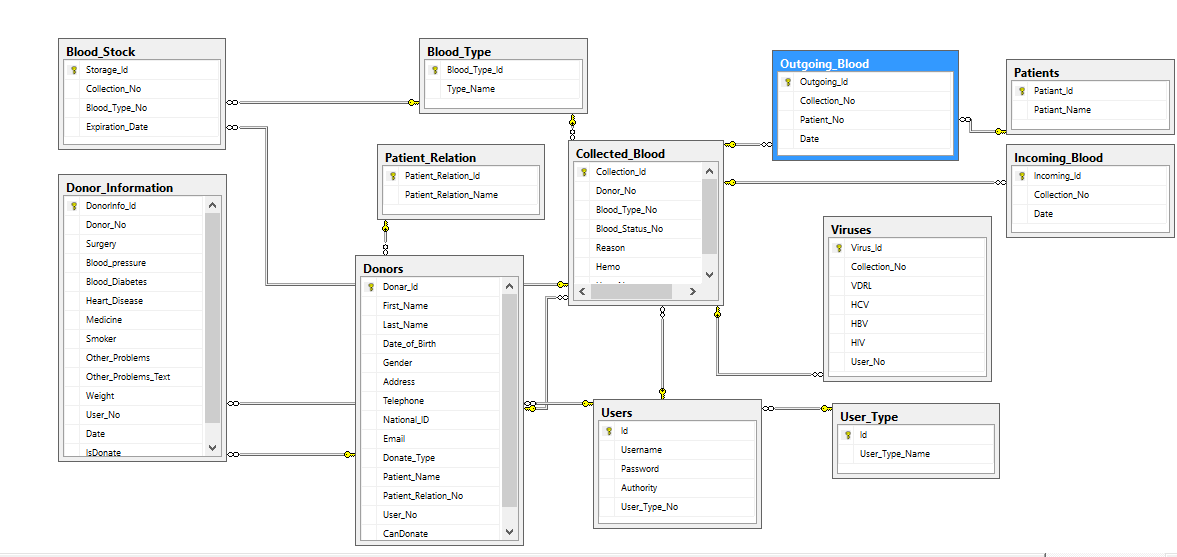
Class diagram identifies the attributes and operations of a class and also the constraints introduced on the system. It represents the static view of an application [9].

****

**Figure 3.16 - Class Diagram**

* + 1. **Database Schema:**

The database schema is a representation of the logical structure of the database, it shows how data are put in tables and it also shows the relationships between the tables [10].

****

**Figure 3.17 - Database Schema**

**CHAPTER 4**

**(Implementation)**

**Chapter 4**

**Implementation**

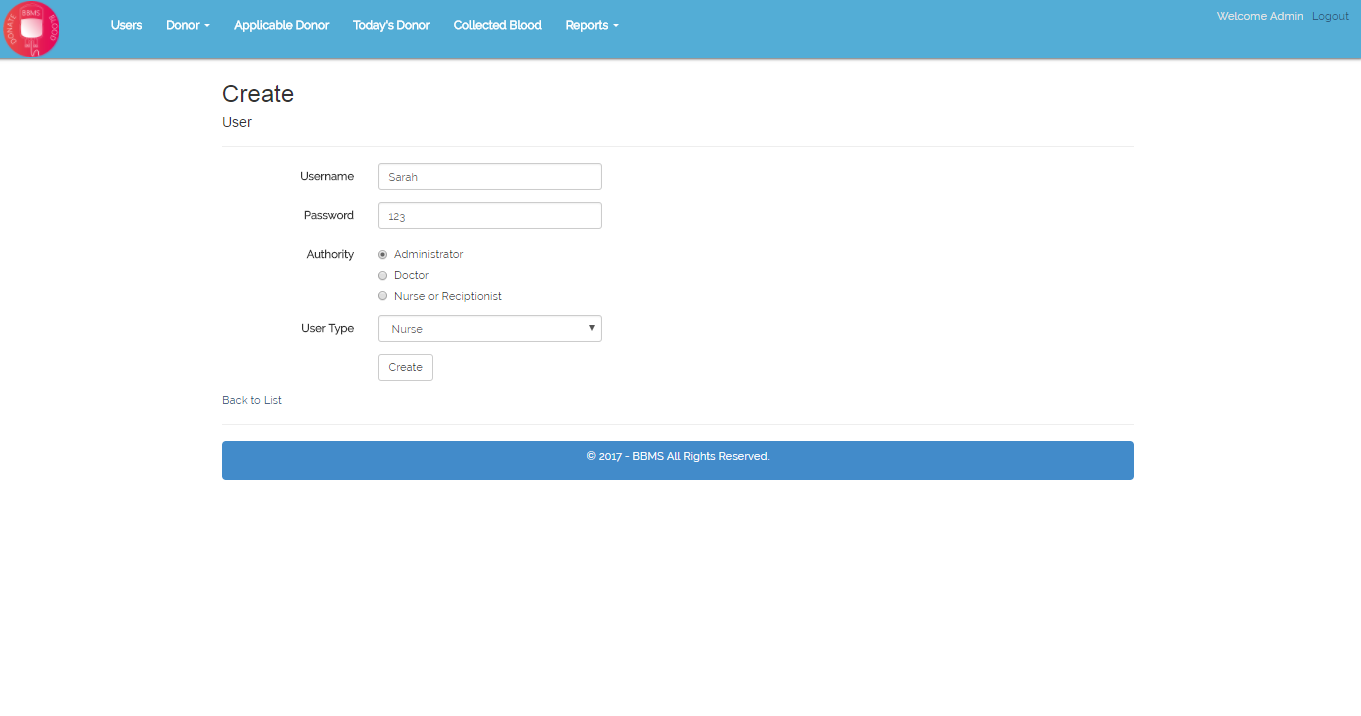
**4.1 Introduction:**

Is to identify how the data should be designed, ensuring the information system is operational and used, and ensuring that information meets quality standards [11].

**4.2Screenshots of the system:**

**4.2.1 Create User:**

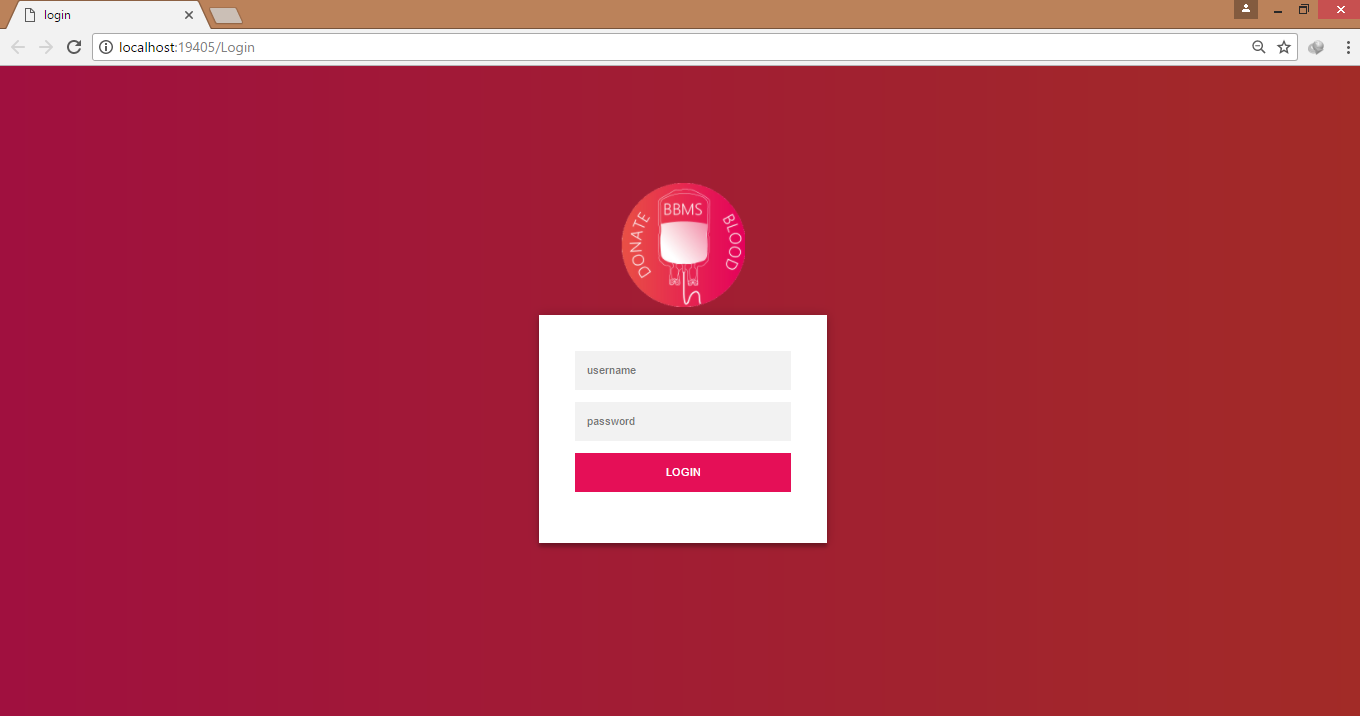
The page that responsible of the creation of a new user.



**Figure 4.1 - Create User Screenshot**

**4.2.2 Login:**

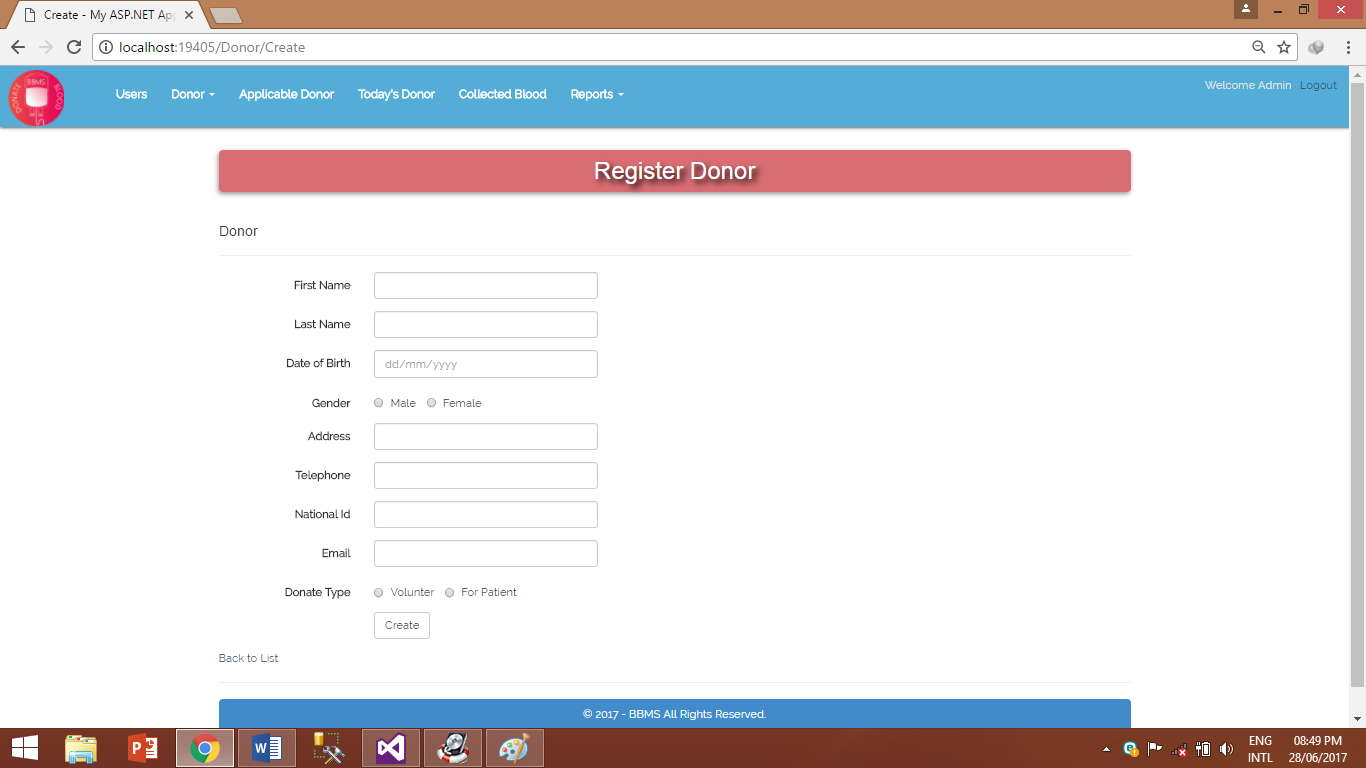
After providing the username and password, the system differentiates the levels of authority of each user.

****

**Figure 4.2 - Login Screenshot**

**4.2.3 Add Donor:**

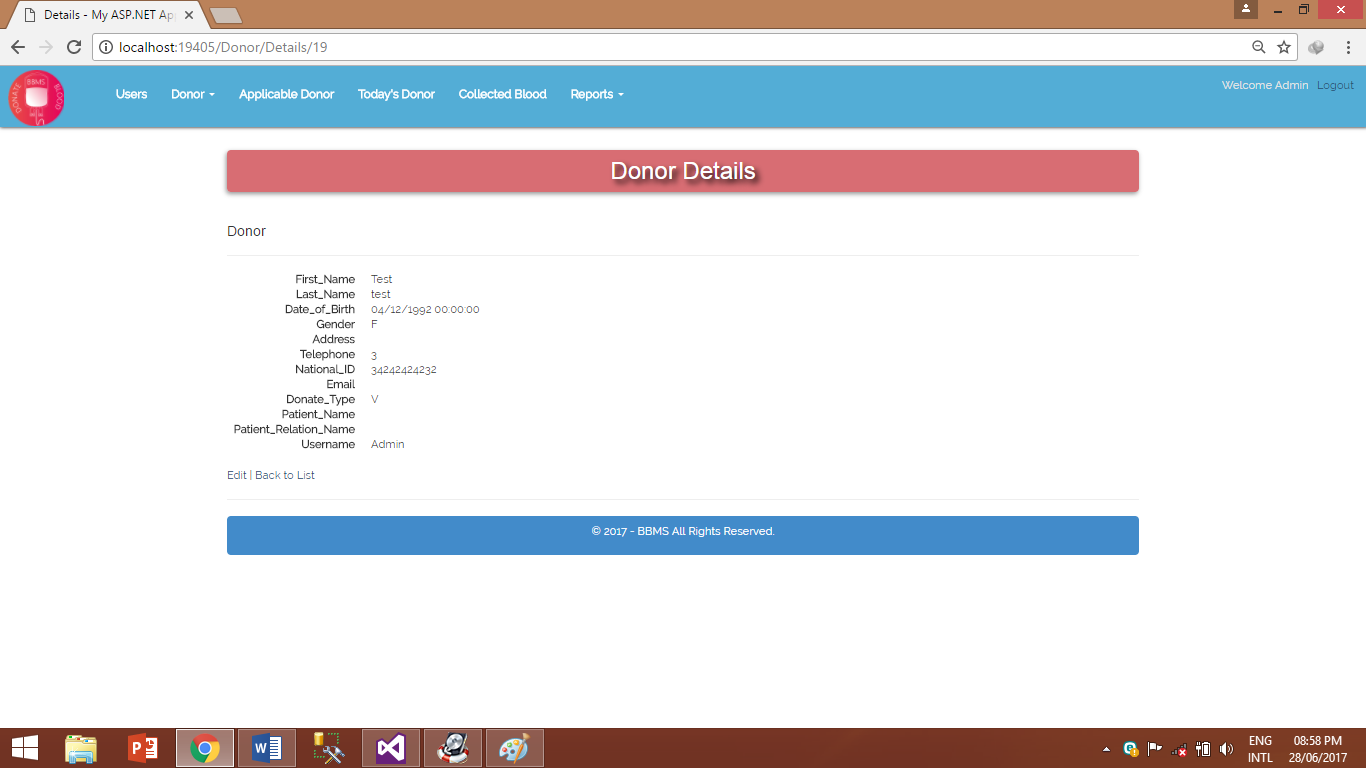
The page that responsible of the creation of new donor.

****

**Figure 4.3 - Add Donor Screenshot**

**4.2.4 Donor Details:**

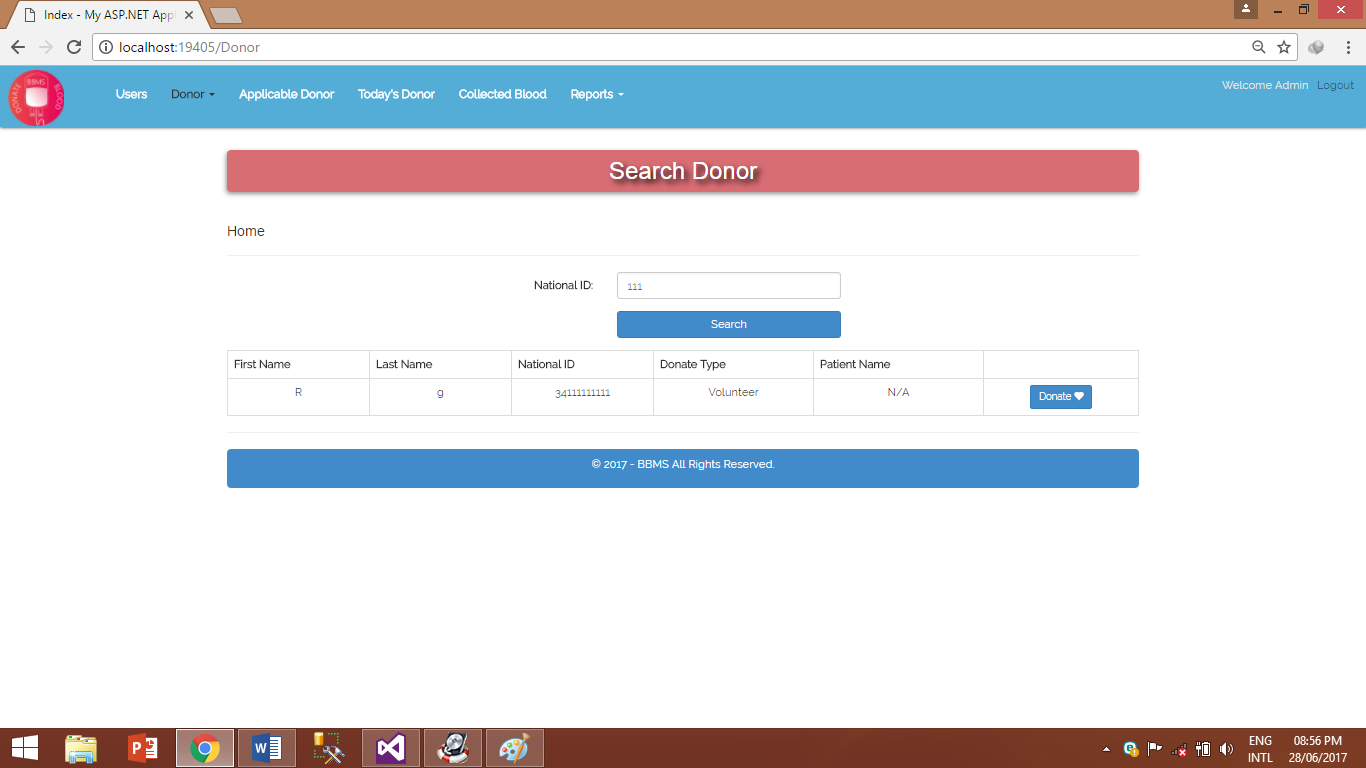
A page designed to display the details of donors.

****

**Figure 4.4 - Donor Details Screenshot**

**4.2.5 Search Donor:**

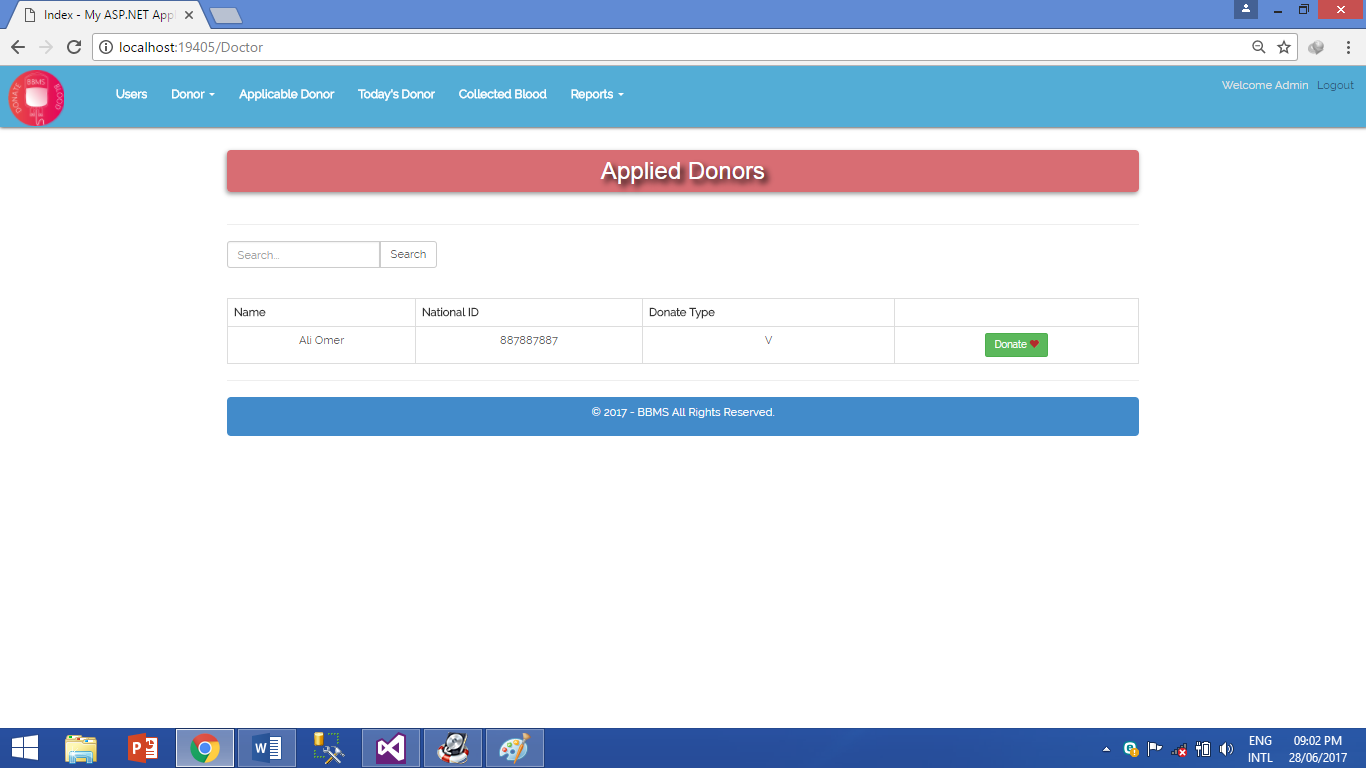
This page enables the user to search donors using national ID.

****

**Figure 4.5 - Donor’s List Screenshot**

**4.2.6 Applied Donor:**

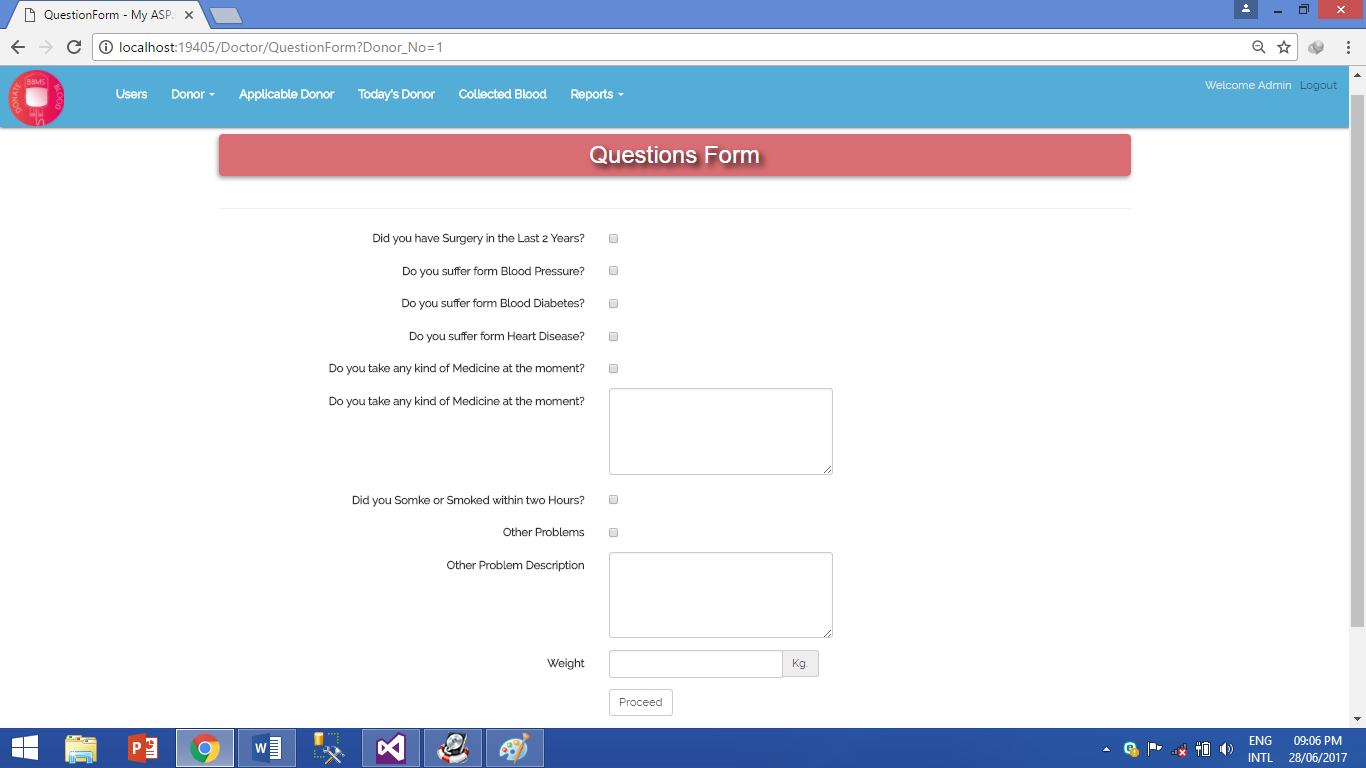
A page created to show whom is ready to proceed to begin tests. If the user had already applied within 2 weeks’ period of time, the system will return a message showing that the donor has already applied within 2 weeks.

****

**Figure 4.6 - Applied Donor Screenshot**

**4.2.7 Questions Form:**

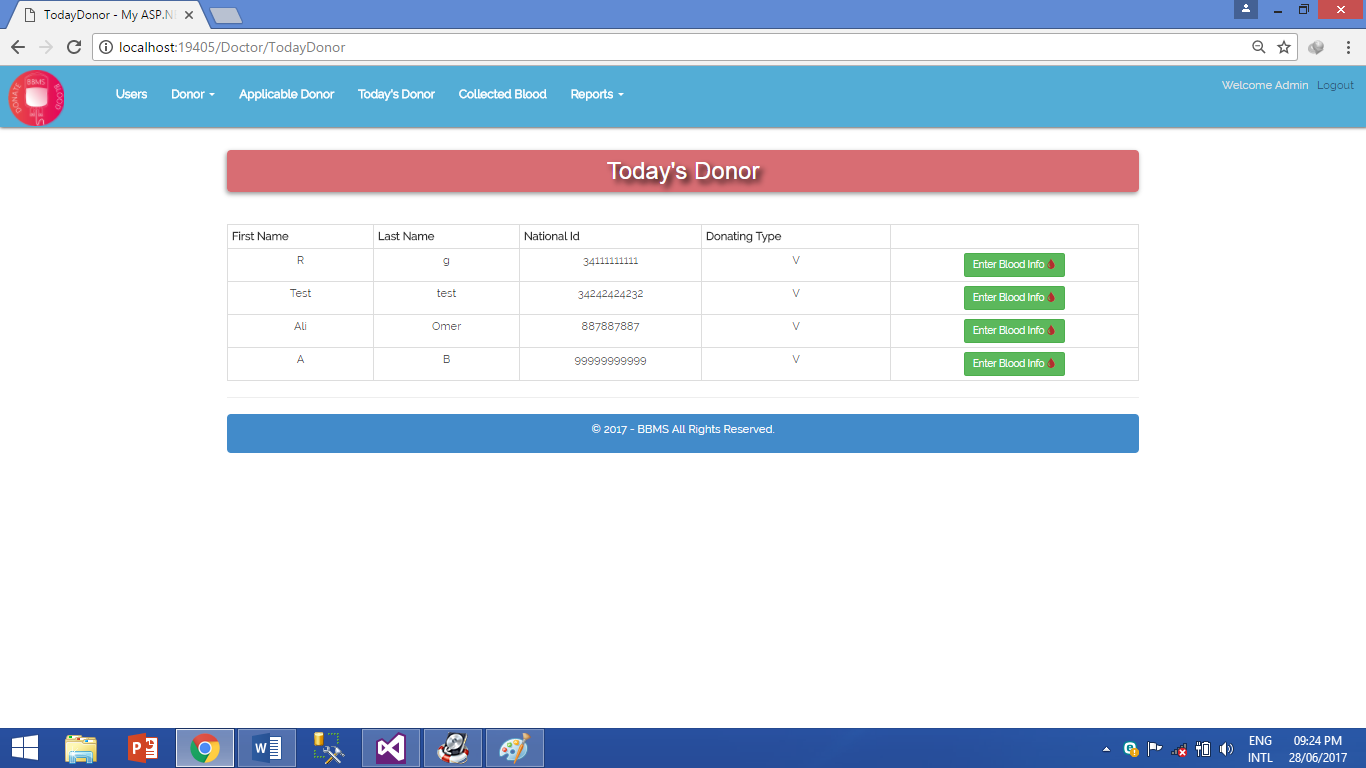
The questions form is the first test to the donor, filled by the user or doctor (general health status). The donor might be rejected due to some conditions such as suffering from heart disease.



**Figure 4.7 - Questions Form Screenshot**

**4.2.8 Today’s Donor:**

A page that shows the list of people ready to begin blood tests.

****

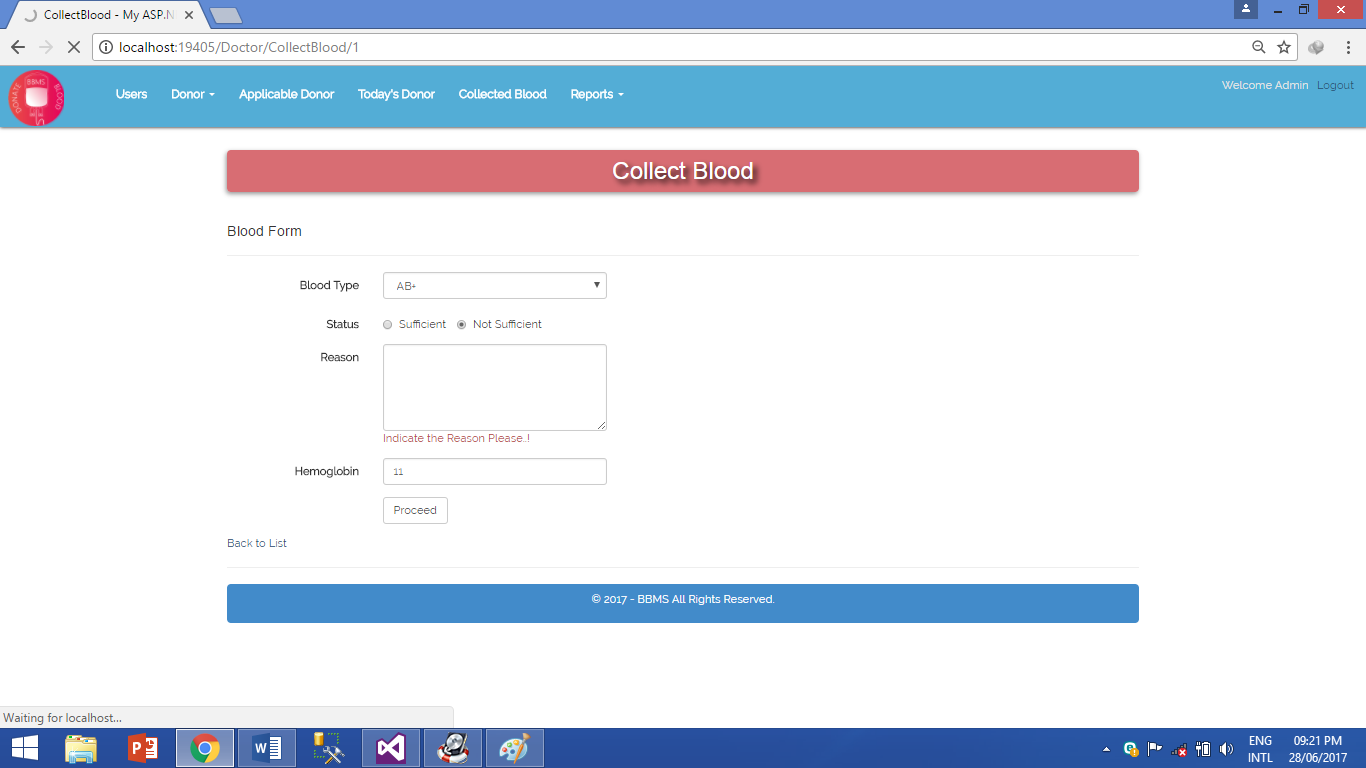
**Figure 4.8 - Today’s Donor**

**4.2.9 Collect Blood:**

This comes after taking the general health status test. A blood sample is collected from the donor then the form is filled by the user or doctor.

If the blood has clumps and needs more tests, the system will store the information of the blood as “not sufficient”.

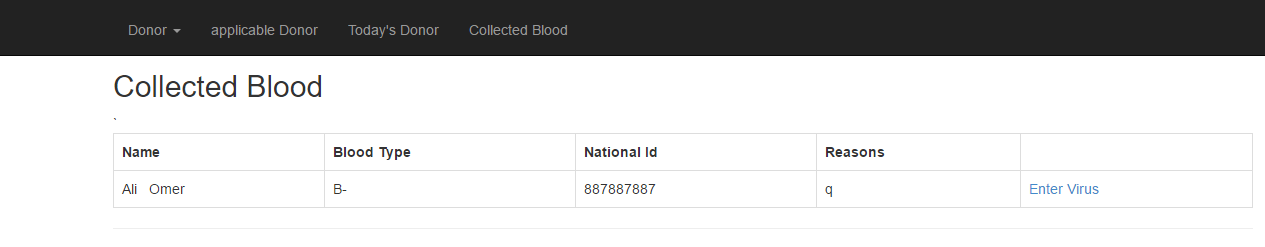
Haemoglobin: the percentage of iron in the blood.

****

**Figure 4.9 - Collect Blood Screenshot**

**4.2.10 Collected Blood:**

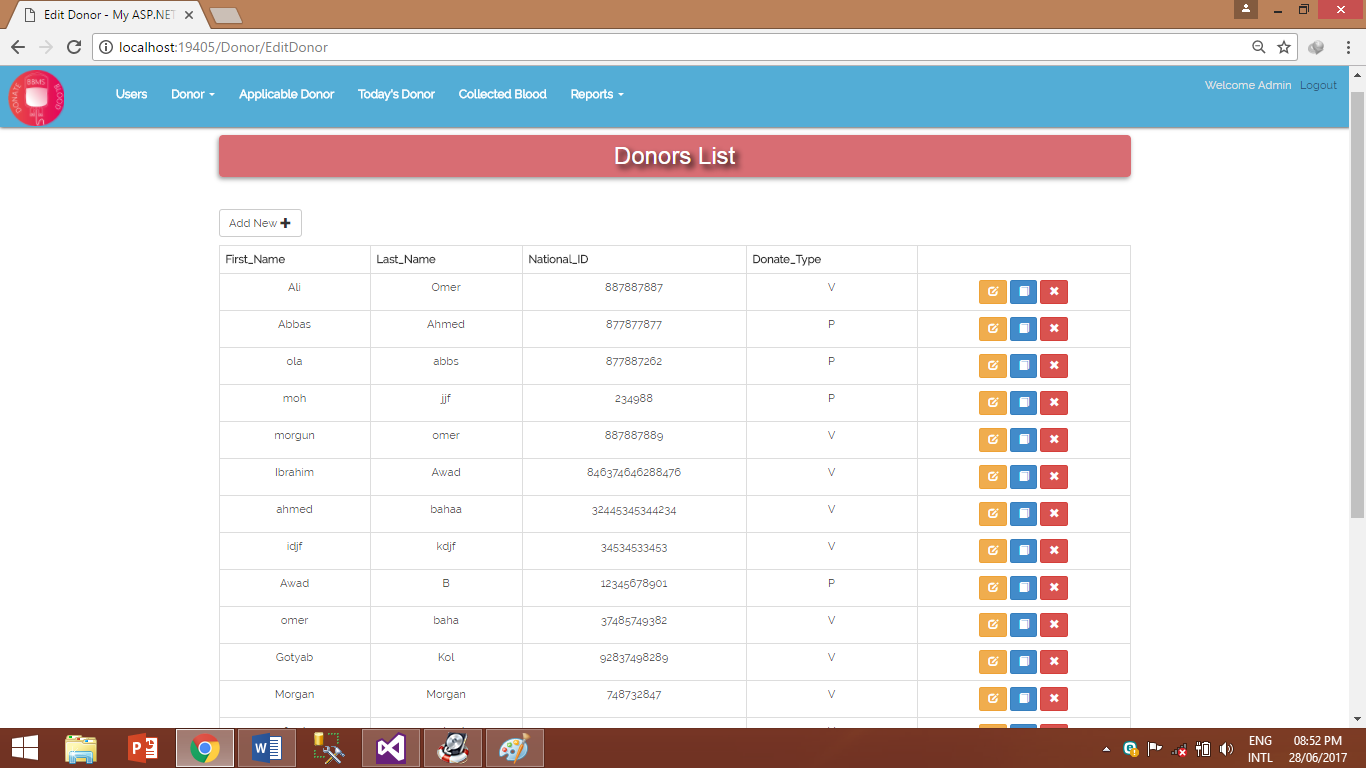
A page that displays the information when a donor has completed the blood test.

****

**Figure 4.10 - Collected Blood Screenshot**

**4.2.11 Edit Donor:**

A page designed to allow editing existing donors.

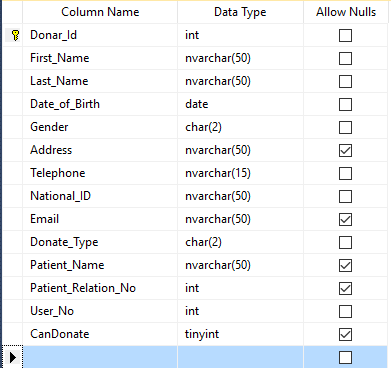
****

**Figure 4.11 - Edit Donor Screenshot**

**4.3 Database Tables:**

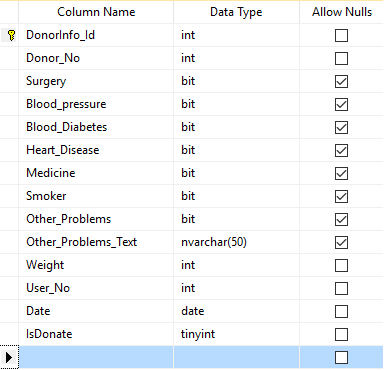
A database is a collection of data that is organized so that it can be easily accessed, managed and updated. Data is organized into rows, columns and tables, and it is indexed to make it easier to find relevant information.

**4.3.1 Donor table:**

****

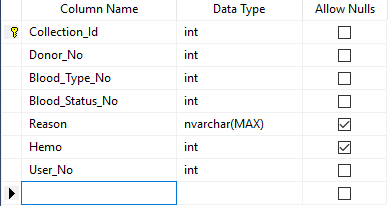
**Figure 4.12 - Donor Database Table**

**4.3.2 Donor Info Table:**



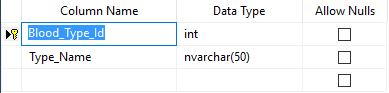
**Figure 4.13 - Donor Info Table**

**4.3.3 Collected Blood table**:



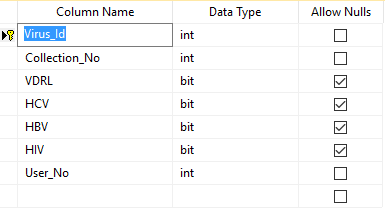
**Figure 4.14 - Collected Blood Database Table**

**4.3.4 Blood Type table:**



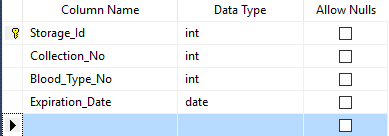
**Figure 4.15 - Blood Type Database Table**

**4.3.5 Virus table:**



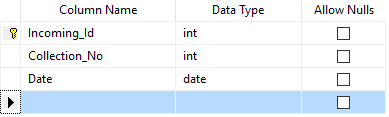
**Figure 4.16 - Virus Database Table**

**4.3.6 Blood Stock Table:**



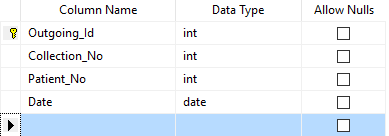
**Figure 4.17 - Blood Stock Database Table**

**4.3.7 Incoming Blood table:**



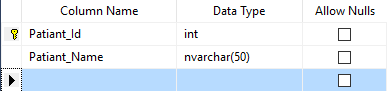
**Figure 4.18 - Incoming Blood Database Table**

**4.3.8 Outgoing Blood table:**



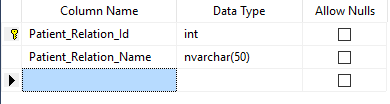
**Figure 4.19 - Outgoing Blood Database Table**

**4.3.9 Patient table:**



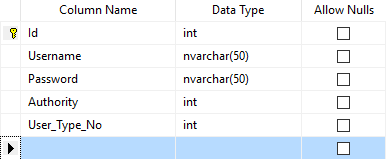
**Figure 4.20 - Patient Database Table**

**4.3.10 Patient Relation table:**



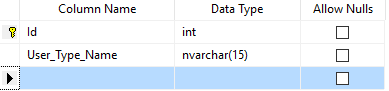
**Figure 4.21 - Patient Relation Database Table**

**4.3.11 User table:**



**Figure 4.22 - User Database Table**

**4.3.12 User Type table:**



**Figure 4.23 - User Type Database Table**

**CHAPTER 5**

**(Conclusion, Recommendation& Future Work)**

**Chapter 5**

**5.1 Conclusion:**

The designed effective solution of the blood bank management system can overcome all problems in the National Public Health Laboratory system, the system’s workflow will facilitate all processes within a blood bank.

**5.2 Recommendation:**

The system is recommended to work on the National Public Health Laboratory. Information integrity is to be done to allow all hospitals to use the software.

**5.3 Future Works:**

Blood Bank Management system future development will be adding a feature and the functionality in a big scale by installing the system in each hospital across the country so every patients and doctors can view the stock of each blood type. Moreover, blood management system can be integrated with the ministry of health so they detect the ratio of donating and diseases as the system stores the blood information, however if the blood is infected with some disease or not. This information can be very helpful for the ministry of health so they can extract from it many useful information, not to forget to mention that the system can have the feature of managing the delivering of the blood to any registered hospital.

**References:**

[1] http://www.redcrossblood.org/learn-about-blood/blood-transfusions/history-blood-transfusions

[2] http://www.bloodbook.com/banking.html

[3]https://en.wikipedia.org/wiki/Bernard\_Fantus#The\_Blood\_Bank.5B7.5D

[4]https://www.tutorialspoint.com/software\_testing\_dictionary/functional\_requirements.htm

[5] https://www.tutorialspoint.com/software\_testing\_dictionary/requirements.htm

[6] https://www.tutorialspoint.com/uml/uml\_use\_case\_diagram.htm

[7]http://www.tracemodeler.com/articles/a\_quick\_introduction\_to\_uml\_sequence\_diagrams/

[8] https://www.tutorialspoint.com/uml/uml\_activity\_diagram.htm

[9] https://www.tutorialspoint.com/uml/uml\_class\_diagram.htm

diagram\_representation.htm

[10] https://docs.oracle.com/cd/E15276\_01/doc.20/e13677/data\_modeling.htm

[11] http://www.uky.edu/~dsianita/695A&D/lecture5.html