EAST WEST UNIVERSITY

Course Code: CSE-411

Course Title: Software Engineering & Information System Design

Section: 01

Project Report

"Blood Donation Database"

Submitted To:

Dr. Mohammad Salah Uddin

Assistant Professor,

Dept. of CSE, East West University

Submitted By:

Zubayar Mahatab Md. Sakif	2018-1-60-105
Nishat Tasmim Madhu	2017-2-60-097
Sk Mohammad Asem	2017-3-60-068

Date of Submission: 28 August 2021

CONTENTS:

1. INTRODUCTION

1.1. INTRODUCTION TO THE PROJECT

Blood Donation Database is an online-based blood search database management system developed in PHP. The purpose of this project is to develop an online blood management information search portal. The main aim of this project is to help users to find and manage blood at ease and in the minimum time possible and to provide blood donation service to the people in need by area wise. It also aims to collect and maintain all the data and information of the pertaining donors and customers so that it can be further developed and helpful in the future and the search of different groups of blood would be easier. This project is one kind of e-commerce project and is implemented in HTML using PHP.

1.2. DESCRIPTION OF THE PROBLEM

Blood is essential for a person to survive and every day many people need blood as a part of treatment such as surgeries, cancer, or other chronic illness, injuries, or sudden accident. A donor can save a life by donating blood. But this process of finding the donor, contact them, and then finally collecting the blood is lengthy in our country if everything is done by manually. This time taking process can risk the patient's life. So like other digitally developed countries, Bangladesh needs some well-developed online Blood Donation systems which can provide the people in need with blood at a minimum time possible by a specific category wise and in an authentic way.

1.3. DESCRIPTION OF THE SOLUTION

In our proposed project, we want to develop a Blood Donation Database to help people, who need blood quickly and easiest way. Because Blood Donation Database is designed web-based to store, retrieve, process and analyze information connected with the administrative.

We made a direct connection between donors and customers through this website. So that the customers can easily find the exact blood group and contact the donor who is in the nearest area possible. The search area would be district and 'Thana' wise. The customers and donors need to create an account using their name, email id, phone number, location nationality, blood group, password, etc information and after that, they can log in anytime they need.

The donors need to give some extra information such as weight, gender, no of the previous donation, date of birth, etc. This information of donors and customers will be store in a database. The admin also can log in separately and can add or edit data. The customers can search for necessary blood groups after login and can see the numbers and details of the donors available in the nearest area. Then they can contact the donors so that blood donation can be possible. Thus the Blood Donation Database can help people finding emergency blood in the quickest and easiest way possible.

2. PROPOSED SYSTEM

2.1 OBJECTIVES

The system mainly aims at the digitalization of the blood donating and collection process and minimizing all the complexities of manual maintenance and the lengthy process of managing donors and collecting blood manually. It can minimize the labor and period of customers also can be helpful for donors to easily find a customer. The users can easily register and log in. The data and information of the users are also stored safely and securely in a database.

2.2 FUNCTIONS

Signup

- The signup page includes the information of the donor who wants to register. Donors can register the account by clicking on registration. Users can add the account for further inquiry of the blood donation.

Login

- Admin Login: The page requires a user name and password to start the application. Login is a process by which individual access to a computer system is controlled by identifying and authenticating the user through the cardinalities presented by the user. Admin can add an update, approve or delete the users.

- Donor login: The user can register the account by fill the information about you and click on submit button. He/she can add the account for the further inquiry of the blood donation. The user has to log in to get more information about the blood bank.

- Customer login: Customer only log in to find a donor who is available to donate blood.

Pages

- Blood group searching page where the details of available donors and their information will be shown.
- The order page will show the details of processing an order.
- Review page where customers can give a review.

2.3 THE SUPERIORITY OF THIS PROPOSED SYSTEM

- The interface is user friendly
- Easy registration using fewer details
- Easy search option
- Sufficient storage
- Less time is needed
- Fats communication and transaction
- Less error

2.4 FEASIBILITY ANALYSIS

Feasibility analysis is important for a project because it considers all of the relevant factors of that project. The feasibility analysis including economical, technical, operational, legal, and scheduling considerations help to understand and analyze the development of the project successfully. A well-developed project's development highly depends on its feasibility analysis. The impact of a feasibility study can ensure the further construction of the project and if it would helpful for the users or not.

The proposed project "Blood Donation Database" is one of a kind of e-commerce service and needs to go through a thorough discussion and plan for its feasibility requirements so that we can develop it properly. The feasibility study for this project is discussed below:

- **Economical Feasibility:** Economical feasibility is the analysis of a project's cost-benefit and financial techniques. Before developing the project, the developers and stakeholders must know about the detailed analysis of the cost and bearing and further financial aspect of that project. The initial investment of this project is the cost incurred as it is a software-developing part. The necessary part of this website uses free development resources and applications. Also we used MySQL and HTML which both are free on the internet. And the software and hardware(mainly smart devices) both parts are easily available at the markets and cost-friendly, so the project is economically feasible.

- **Technical Feasibility:** Technical feasibility of a project is the detailed analysis of how the developers intend to deliver their product to users. It consists of a detailed preview of the source material, locations or environments, technology, and human resources, etc. A good analysis of the technical feasibility of the proposed project can determine whether the project will perform well or not. In our proposed project, we are using HTML as the front-end tool and PHP MySQL as the back-end tool for developing this project. It will serve as a personal web server and web page and will help both users and developers. Anyone who has smart devices and good quality internet can easily use this software.

Users can easily register or log in/sign in to the system easily and quickly and use the system smoothly without any difficulties from any place where good internet service is available. Our “Blood Donation Database” software meets the need of the users and it also provides technical reassurance.

- **Operational Feasibility:** Operational feasibility is the analysis and measurement of how well the proposed project can solve the targeted problem. In our proposed project, one’s registered, users can log in to the system any time and without much of a requirement. The software is available to the customers of the selected region and both customers and admins can use this platform without outstepping. This software will give the customers service without changing their place or location, so they won’t need to waste their time and labor and can get emergency services. They can also use this software almost freely as it only required smart devices and an internet connection which is easily available this day.

3. REQUIREMENT ANALYSIS

Requirement analysis defines whether a project meets the expectation of the users and is built and modified by that requirement. It involves the need of the users, different stakeholders, and developers requirements by analyzing, validating or modifying, or managing systems requirements. Requirement analysis should be discussed and monitored by all the time while developing the system. Requirement analysis can be divided into two parts-functional and non-functional requirements. Our project “Blood Donation Database” went through a thorough discussion and plan for its requirements so that we can develop it properly. Identifying and resolving the requirements are a necessary part of our proposed project. Some of its functional and non-functional requirements are discussed below:

3.1 Functional Requirements: Functional requirements define the behavior and include calculations and other processes of the basic system. In our proposed system, the following functional requirements are available:

- The users(customers/donors) can log in when generating a unique id after the registration successfully.
- Customers can easily find the available donors according to the necessary blood group search in the nearest area.
- Then they can choose the donor and order their product direct from the website.
- The customer can track their collectible and can contact the donor.
- Donors can edit and update their health conditions.
- Users(customers/donors) are able to see every requirement they want.
- The system can handle a large number of transactions.

- The background of the web page is a light color(white/blue).
- Admin can be able to log in the system and create a user login for the users.

3.2 Non-functional Requirements: Non-functional requirements(NFR) define the overall analysis of the whole operation of the system criteria rather than any specific behavior. In our proposed system, the following non-functional requirements are available:

- The system can be secure as it contains the personal health and contact information of the users.
- It is to access and use in the case of an emergency.
- The system can be reliable and when a large number of transactions happen it must not be crash.
- It can fill availability requirements and can be available to the customers.
- The customers can be able to find their required collectible in their selected area.
- Admin can view and modify all the data.
- The speed of the system depends on the user's internet speed and can be easily used on any smart device.
- Users can give reviews and feedbacks.
- Users can update and change their passwords.

3.3 Requirement Specifications

Any good project depends on a well thorough analysis of system requirement specifications. It is an essential part of the whole developing process of the system. Our proposed system can be developed by using a variety of devices, software, hardware, platforms. The devices, software and hardware, and other essential objects we used to create our system are listed below:

3.3.1 Software Requirements

- Operating System: Windows 10
- Frameworks/APIs: MySQL Workbench
- Database: MySQL
- IDE/Text Editor: Sublime Text, Atom, Visual studio code
- Front End: HTML 5, CSS3, JavaScript
- Browser: Preferable Google Chrome / Mozilla Firefox

3.3.2 Hardware Requirements

- Processor: Minimum Intel Core I3
- Processor speed: 2.50 GHZ
- RAM: 4 GB

4. SYSTEM DESIGN

The performance of the system requirement review process can be used as an input to the system design phase in the software development life cycle. The architectural description of a System, which includes information about its components and subcomponents, is referred to as the design of the system. The diagrammatic representation of the system, modules, and subsystems provides a visual representation of the system, modules, and subsystems a thorough understanding of the device and its design specifications. We define things with UML diagrams and imagine different aspects of a system's architecture. The System Design ensures security, reliability, and the ability to deliver desired performance to end-users based on available resources, as well as that the system is flexible and dynamically changes.

4.1 Use Case Diagram

Although only comprehending a system's static existence is inadequate, Use-Case diagrams assist in providing a dynamic view of the system. Case diagrams are a type of diagram that depicts a situation. An application's architecture and subsystems. There are some external as well as internal variables to consider. The complex essence of the Use Case diagram is exemplified by this. Actors is what we term them. During Use Case diagrams can be thought of as a high-level system requirement analysis. Provide a clear picture of the actors and their positions (use cases) and are thus important. Use case diagrams are a visual representation of the actors (internal or external factors), their positions (use cases), and the relationships between them.

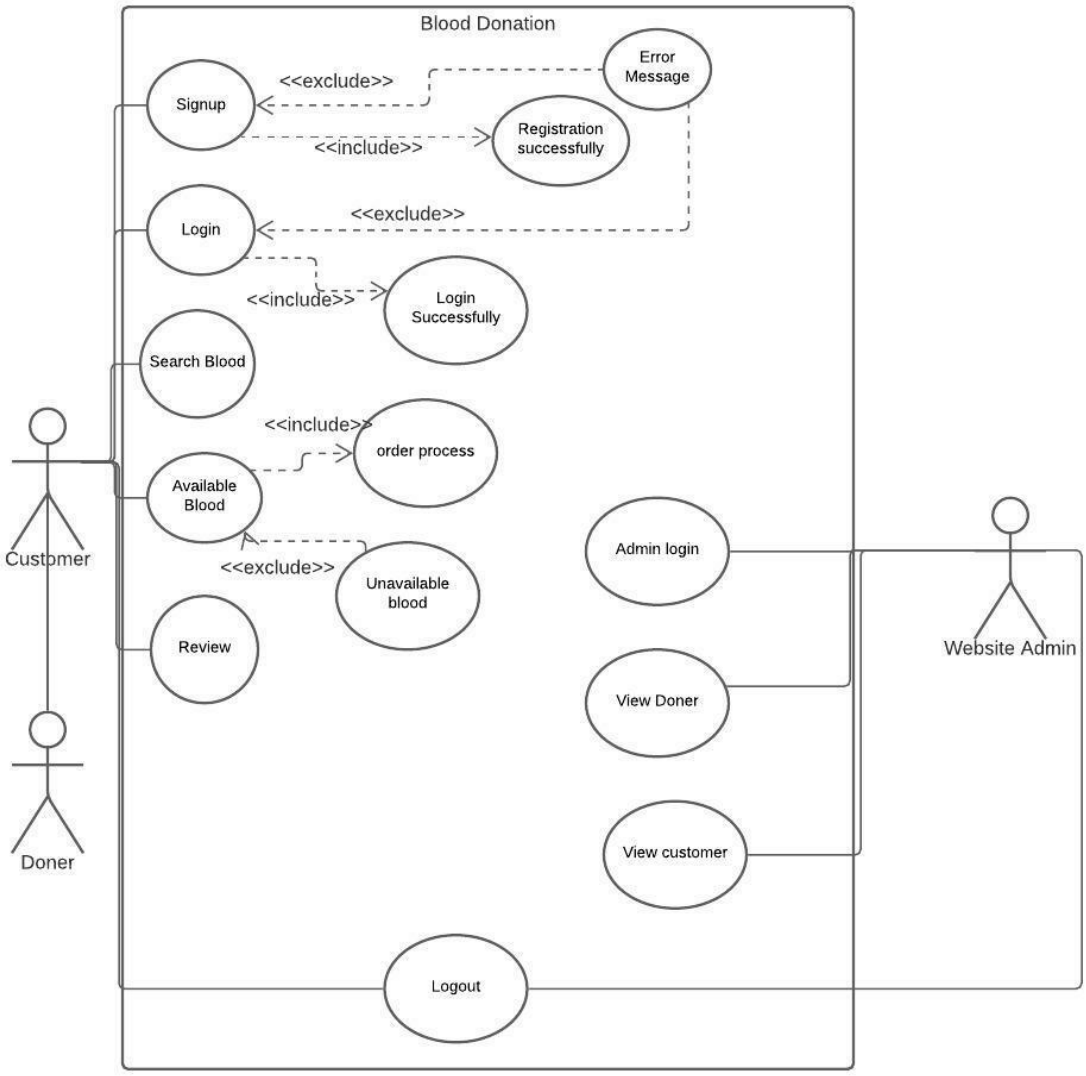


Fig: Use Case Diagram

Use Case [number # 1]	Sign Up	
Goal in Context	This use case allows donors and customers to create an account in this software to order or reserve one or multiple blood groups.	
Preconditions	The server is up and running.	
Success End Condition	Donors and Customers can successfully create an account	
Failed End Condition	Donors and Customers failed to create an account if the form is not filled up correctly.	
Primary Actor	Customer and Donor	
Secondary Actors	Admin	
Trigger	Donors and Customer goes into the homepage of the website and clicks the Registration button in the navigation bar.	
Description	Step	Action
	1.	Donors and Customer fills up the form and confirms.
Extensions or Variations	Step	Branching Action

Use Case [number #2]	Log In	
Goal in Context	This use case allows donors and customers to log in to their account in the software reserve one or multiple blood groups.	
Preconditions	The server is up and running.	
Success End Condition	Donors and Customers can successfully login	
Failed End Condition	Donors and Customers failed to log in to their accounts if they input incorrect passwords or email.	
Primary Actor	Customer & Donors	
Secondary Actors	Admin	
Trigger	Donors and Customer goes into the homepage of the website and clicks the login button in the navigation bar.	
Description	Step	Action
	1.	Donors and Customers log in successfully and see confirmation

		messages.
Extensions or Variations	Step	Branching Action
	1.	Account verification
	2.	Log in successful

Use Case [number #3]	Search Blood	
Goal in Context	This use case allows customers to search any kind of blood to their location(District, Thana).	
Preconditions	The server is up and running.	
Success End Condition	Customers can search successfully.	
Failed End Condition	…	
Primary Actor	Customer	
Secondary Actors	…	
Trigger	The customer goes into the home page of this software and clicks see menu button in the navigation bar and clicks search blood near.	
Description	Step	Action
	1.	Customer search blood with fill up their requirement.
Extensions or Variations	Step	Branching Action
	1.	Search Blood successfully

Use Case [number #4]	Available Blood
Goal in Context	This use case allows customers to search blood if blood is available, order any group of blood to their account.
Preconditions	The server is up and running.
Success End Condition	Customers can successfully order blood.
Failed End Condition	Blood unavailable.
Primary Actor	Customer

Secondary Actors	Admin	
Trigger	The customer goes into the welcome page of this software and clicks the see menu button in the navigation bar and clicks add the button to search the blood.	
Description	Step	Action
	1.	Customer search blood (required blood, district, Thana), their information in the form and confirms the order.
Extensions or Variations	Step	Branching Action
	1.	Order successfully

Use Case [number #5]	Review	
Goal in Context	In this use case, customers can give donors reviews.	
Preconditions	The server is up and running.	
Success End Condition	Customers can successfully login	
Failed End Condition	…	
Primary Actor	Customers	
Secondary Actors	…	
Trigger	End of the collecting and reviewing customers go into the homepage of the website and clicks the login button in the navigation bar.	
Description	Step	Action
	1.	Customers successfully and sees confirmation review.
Extensions or Variations	Step	Branching Action
	1.	Reviewing successfully

Use Case [number #6]	Admin Log In	
Goal in Context	Admin log in to their account in the software.	
Preconditions	The server is up and running.	
Success End Condition	Admin can successfully login	
Failed End Condition	Admin failed to log in to their accounts if they input incorrect passwords or email.	
Primary Actor	Admin	
Secondary Actors		
Trigger	Admin goes into the homepage of the website and clicks the login button in the navigation bar.	
Description	Step	Action
	1.	Admin log in successfully and see confirmation messages.
Extensions or Variations	Step	Branching Action
	1.	Account verification
	2.	Log in successful

Use Case [number #7]	View Donor	
Goal in Context	This use case allows the admin to view donors.	
Preconditions	The server is up and running.	
Success End Condition	Admin can successfully view donors	
Failed End Condition	…	
Primary Actor	Admin	
Secondary Actors	…	
Trigger	Admin goes into the homepage of the website and clicks the login button in the navigation bar.	
Description	Step	Action

	1.	Admin can successfully view donors
Extensions or Variations	Step	Branching Action
	1.	…
	2.	….

Use Case [number #8]	View Customer	
Goal in Context	This use case allows the admin to view the Customer.	
Preconditions	The server is up and running.	
Success End Condition	Admin can successfully view Customer	
Failed End Condition	…	
Primary Actor	Admin	
Secondary Actors	…	
Trigger	Admin goes into the homepage of the website and clicks the login button in the navigation bar.	
Description	Step	Action
	1.	Admin can successfully view Customer
Extensions or Variations	Step	Branching Action
	1.	…
	2.	….

Use Case [number #9]	Log Out
Goal in Context	This use case allows donors and customers, admin to log out of this software.
Preconditions	The server is up and running.

Success End Condition	Customers, donors, and Admin can successfully log out.	
Failed End Condition	…	
Primary Actor	Customer, donor	
Secondary Actors	…	
Trigger	Customer donors and Admin clicks the logout button in the navigation bar.	
Description	Step	Action
	1.	Successfully logs out from this software.
Extensions or Variations	Step	Branching Action

4.2

Activity Diagram

The Activity Diagram is another significant UML diagram that depicts the system's execution flow. Although activity diagrams are not exact flowcharts, they do have some features such as branching, swim lanes, and indicating parallel flows. It is a pictorial representation of a system's various operations that provides a holistic view. Inside the activity diagrams, the idea of forking and joining is used to display the activity of the various components of the system. An operation of the system is a process performed by the system. We begin drawing the activity Diagram once we have a mental picture of the entire flow.

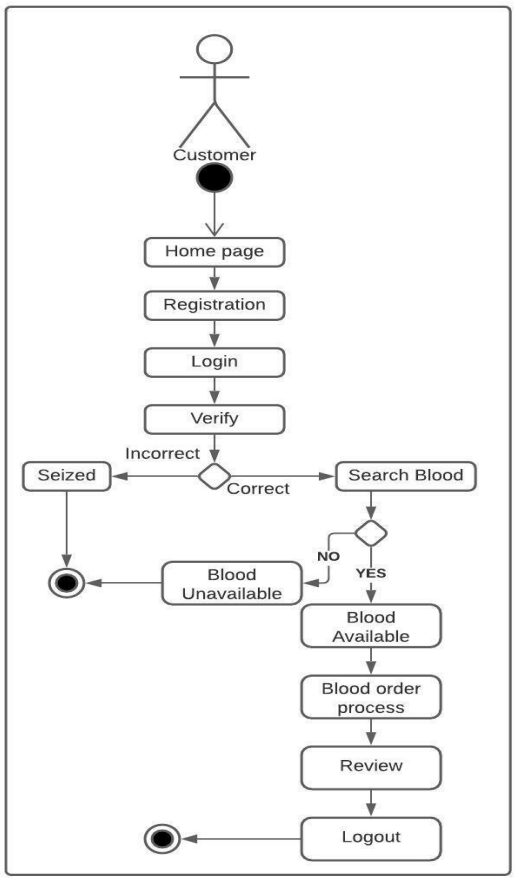


Fig: Activity Diagram (Donor)

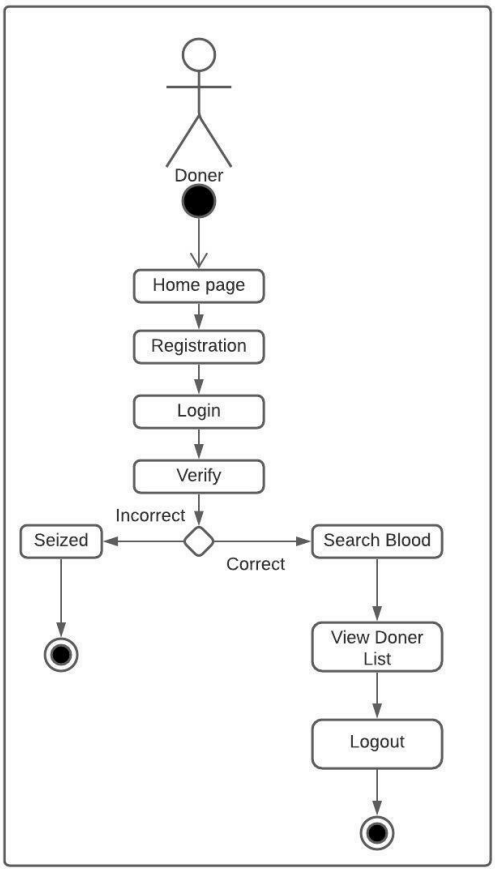


Fig: Activity Diagram (Customer)

4.3 Class Diagram

A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's: classes, their attributes, operations (or methods), and the relationships among objects.

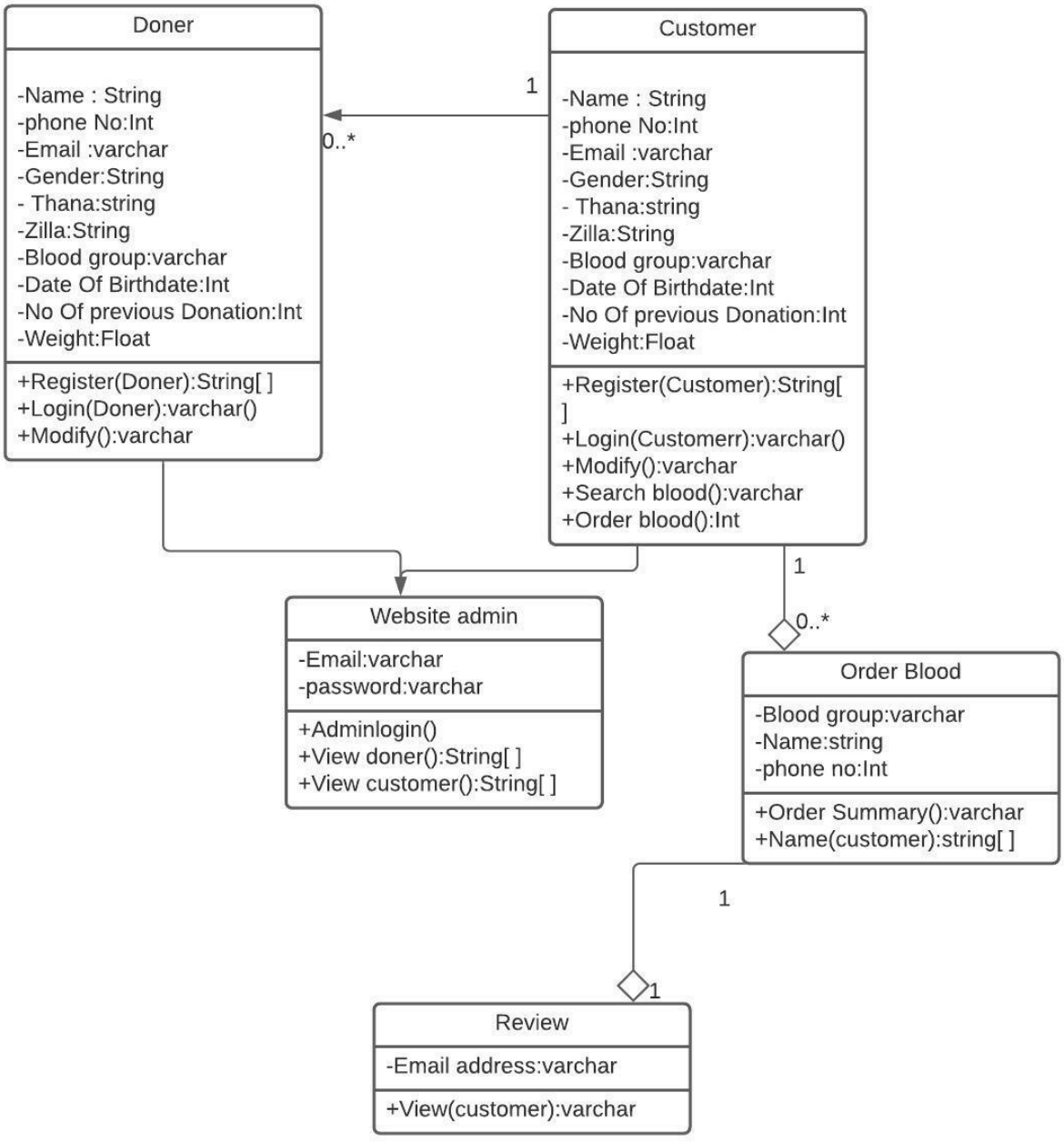


Fig: Class Diagram

4.4 ER diagram

The entity-relationship diagram depicts all of the database tables as well as the relationships that exist between them. It also displays the cardinality of the tables, i.e. the many to one, one to one, or one to many relationships. This is the first step in the database design process. All of the specifications and specification information for the various database entities are formulated at the start and then converted into a diagram. This phase is time-consuming, but once completed, creating a fine, solid, and stable database is a piece of cake.

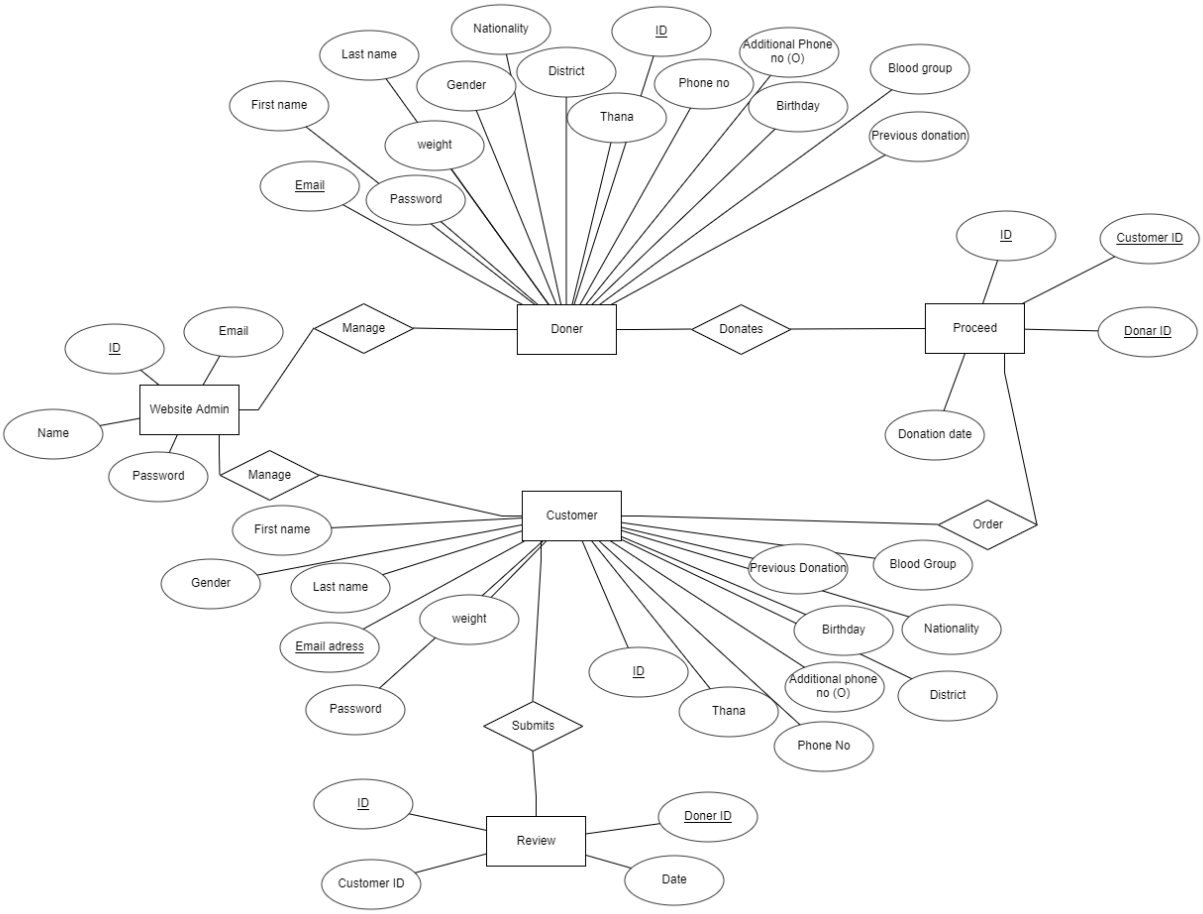
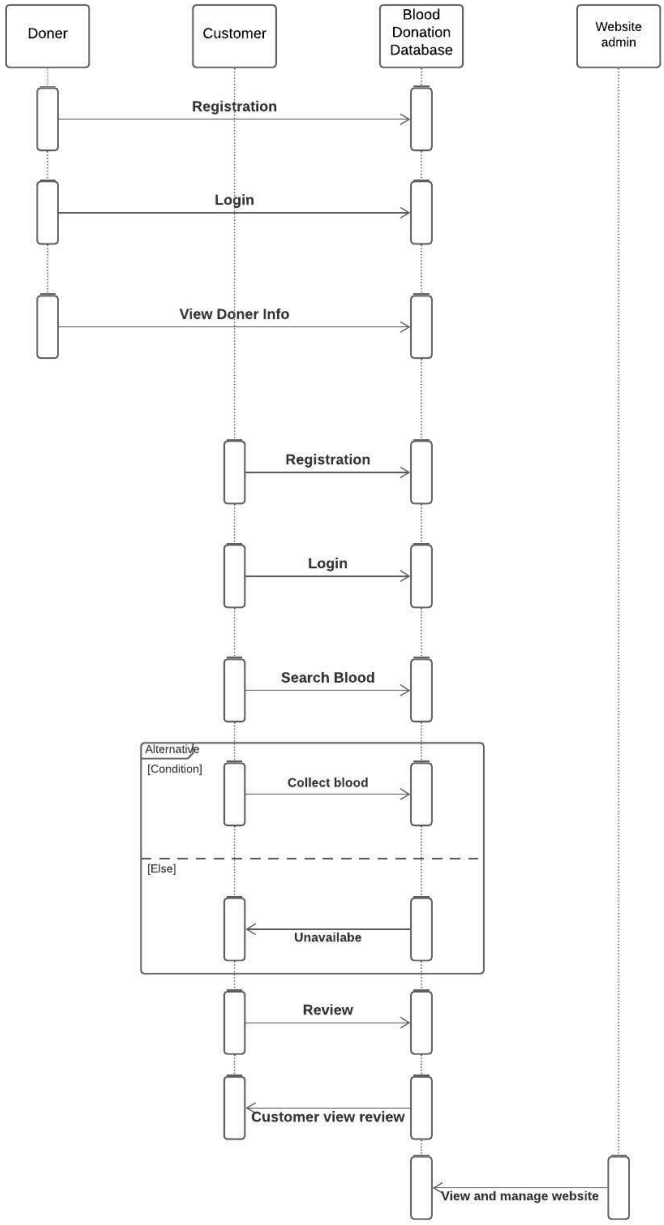


Fig: Entity Relationship Diagram

4.5 Sequence Diagram

The sequence diagram depicts the sequential flow of a system and its subsystems in pictorial form. Since the following diagram is an overall system sequence diagram, sequence diagrams for each part of the system can also be drawn at the modular level. Sequence diagrams place a greater emphasis on system specifications than on system architecture. It focuses more on the sequence of messages distributed immediately following a series of activities. In general, a sequence diagram aids in the modeling and documentation of how a system should behave, as well as the validation of the logical actions of complex operations and functions.



5. TEST CASE

5.1 TESTING PLAN:

The initial test plan addresses system test preparation, and it is revised in the elaboration, design, and implementation phases to accommodate other testing requirements. The following items must be included in the test plan:

- Scope of research
- Testing methodology Types of tests to be performed
- Resource & device criteria
- A rough Test Schedule
- Identification of different forms to be used to document test cases and test results

5.2 TESTING OF SYSTEM:

- **Unit Case Testing**:Unit testing is the process of testing an application down to its smallest component. Unit testing is all about testing each module of an application with multiple test cases and checking validations against unpredictable scenarios. When a bug is discovered, it must be dealt with immediately is entered into the bug tracker, a ticket is created, the bug is fixed, and new unit tests are written. Unit checking is performed on the debugged piece of code using cases.
- **Integration Testing**:After each specific component of the system has been checked, down to the smallest element, the system's various components are now combined and tested together. Integration testing determines whether the integration works or whether a component of the system that is functional on its own begins to fail when it is combined with another part.
- **System testing:** System Testing determines whether an integrated system meets all of its criteria and requirements. Regression testing – Once the device has been debugged, it is retested to see whether it is consistent with the improvements that have been made as well as any changes to the environment.

5.3 White Box Testing:

Case -1:

```
1. <?php
2. include_once("php/db_connect.php");
3. if (empty($_SESSION['adminID'])) {
4. if (isset($_POST['admin'])) {
5. $pass = md5($_POST['Password']);
6. $sql ="select ID from doner where Email='".$_POST['Email']."' and
   Password='".$pass."';";
```

```
7.  $result = mysqli_query($link,$sql);
8.  if ($data= mysqli_fetch_assoc($result)) {
9.  $_SESSION['adminID']=$data['ID'];
10. header("location:index.php");}}}
11. else {
12. header("location:index.php");}
13. if (empty($_SESSION['customerID'])) {
14. if (isset($_POST['customer'])) {
15.             $sql1 ="select ID from customer where Email='".$_POST['Email']."'
    and Password='".$pass."';";
16.             $result1 = mysqli_query($link,$sql1);
17. if ($data1= mysqli_fetch_assoc($result1)) {
18.     else {
19.  header("location:index.php");}
```

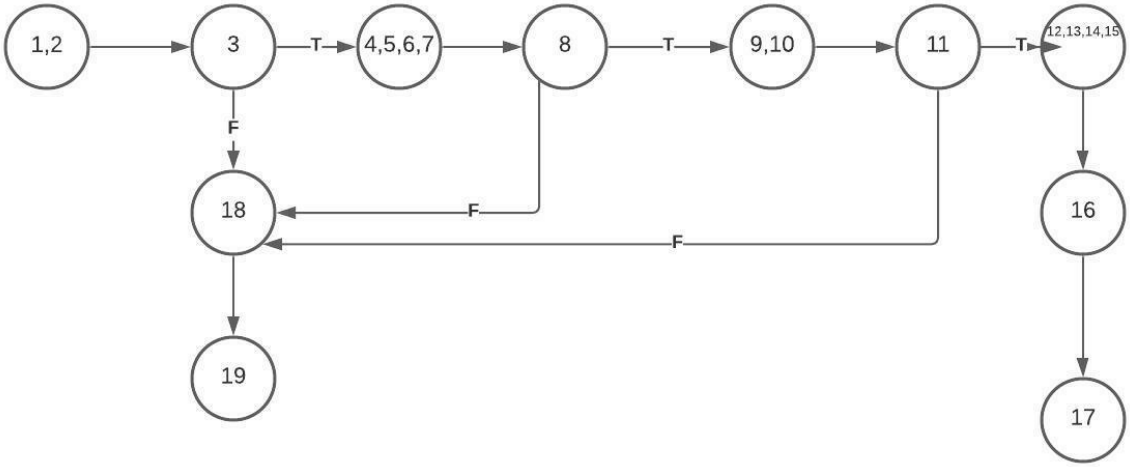


FIG: Control Flow Graph

No of edge = 11

No of node = 12

Cyclometric complexity = 11-12+2 = 1

Testing criteria:

1. Statement Coverage:

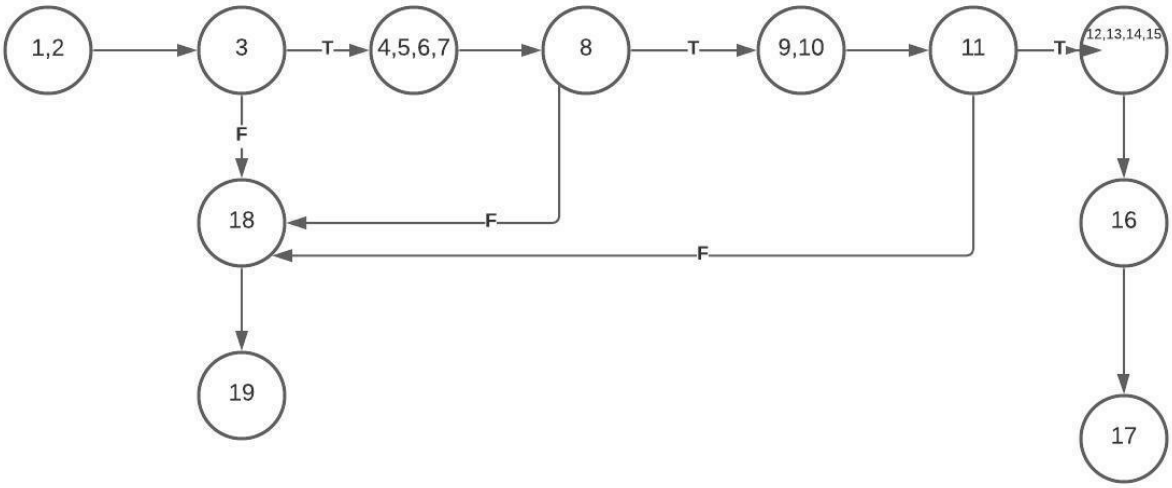


Fig: Control Flow Graph

Example all-statements-adequate test set:
($_SESSION[‘loggedin’] = true)
($_SESSION[‘loggedin’] = False) (not login)
($_GET[‘x’] = true)
P a g e ($_GET[‘x’] = False)(close unwanted enter)
($_POST[‘post’] = true, $result = true)
($_POST[‘post’] = False) data posted error
($result = false) database error
(fetch($r) != NULL , true)
($result = true) registration complete
($result = false) registration failed

2. Edge coverage**:**

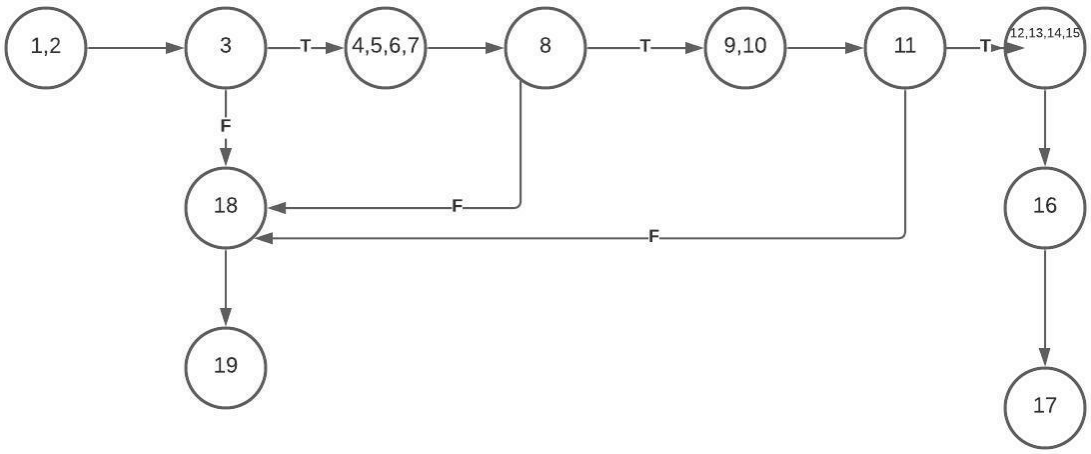


FIG: Control Flow Graph

Edge: {(1,2)→3→(4,5,6,7)→8→(9-10)→11→(12,13,14,15) →16→17→18→19 }
Edge: {(1,2) →3→18 →19,}
Edge: {(1,2)→3→(4,5,6,7) →8 →(18,19)}
Edge: {(1,2)→3→(4,5,6,7,8) →(9,10) →11→(18,19)}
Edge: {(1,2)→3→(4,5,6,7,8) →(9,10) →11→(12,13,14,15) →16→17}
Similarly All condition and path is cover

Case-2:

```
1.  <?php
2.  include_once("php/db_connect.php");
3.  if (isset($_POST['bloodSearch'])) {
4.    $blood = $_POST["sector"];
5.    $district = $_POST["donate_zila"];
6.    $upazila = $_POST["donate_upazila"];
7.    $sql ="select * from doner where BloodGroup='".$blood."' and (District
      ='".$district."' and Address='".$upazila."');}


8.    $sql ="select * from doner ;";;
9.  if (empty($_SESSION['customerID'])) {
10.  if (empty($_SESSION['adminID'])) {
11.  header("location:signIn.php");}
12.  $orderID = 0;}
13. if (isset($_POST['reviewSubmit'])) {
14.  $customerid = $_GET["cusID"];
15.  else
16.  else
17. header("checkout.php");
18. else{
19. header("proceed.php");
```

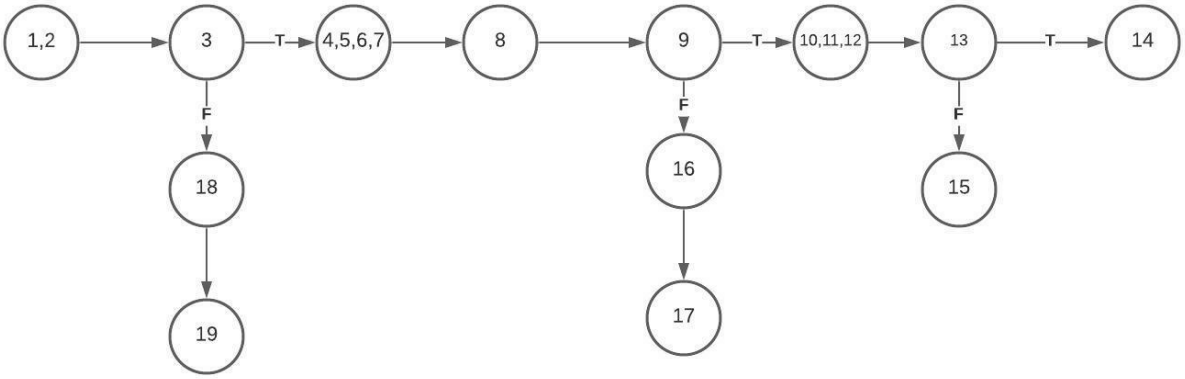


FIG: Control Flow Graph

No of edge: 13

No of node: 12

Cyclometric complexity = 13-12+2 =3

1. Statement Coverage

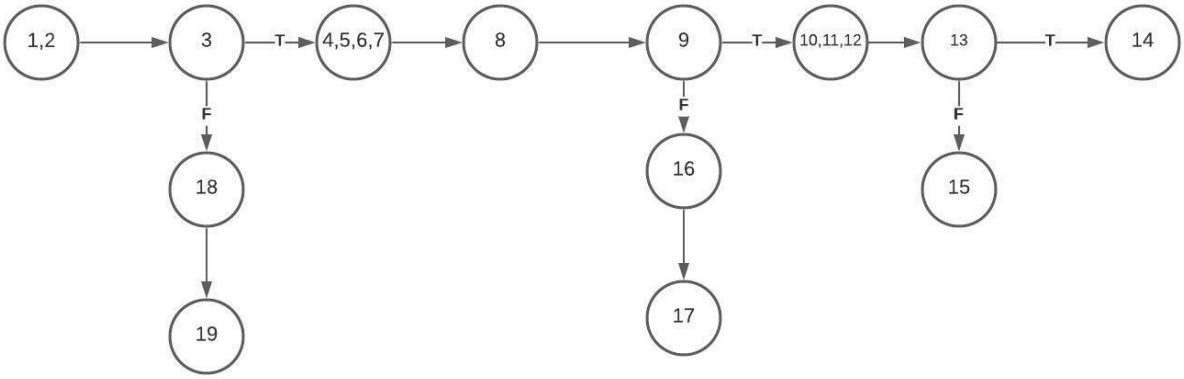


FIG:Control Flow Graph

Example all-statements-adequate test set:

(fetch($result)!= NULL , true)

(fetch($r)!= NULL , true)

(fetch($r2)!= NULL , true)

(fetch($r)== NULL ,false)exit

(fetch($r2)== NULL ,false) exit

(fetch($result)== NULL ,false) exit

2. Edge Coverage:

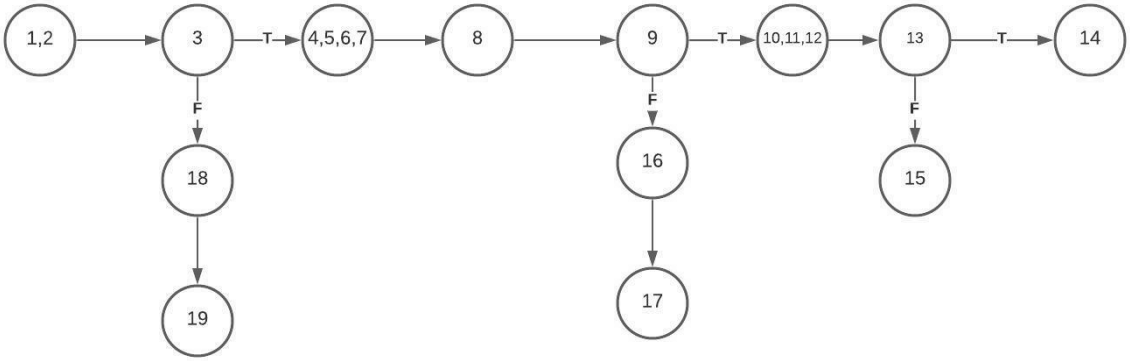
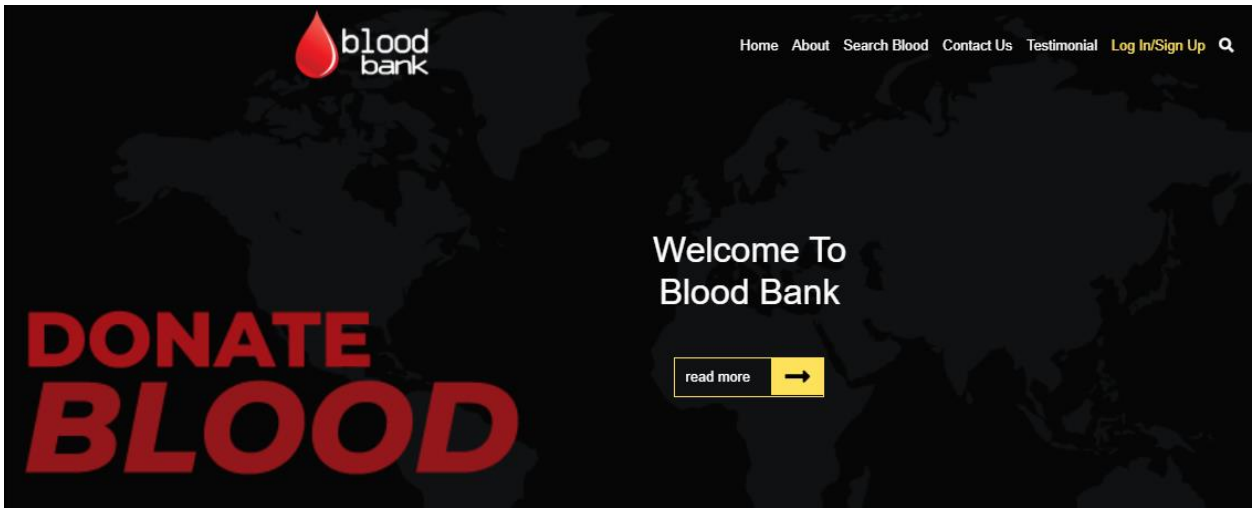


FIG: Control Flow Graph

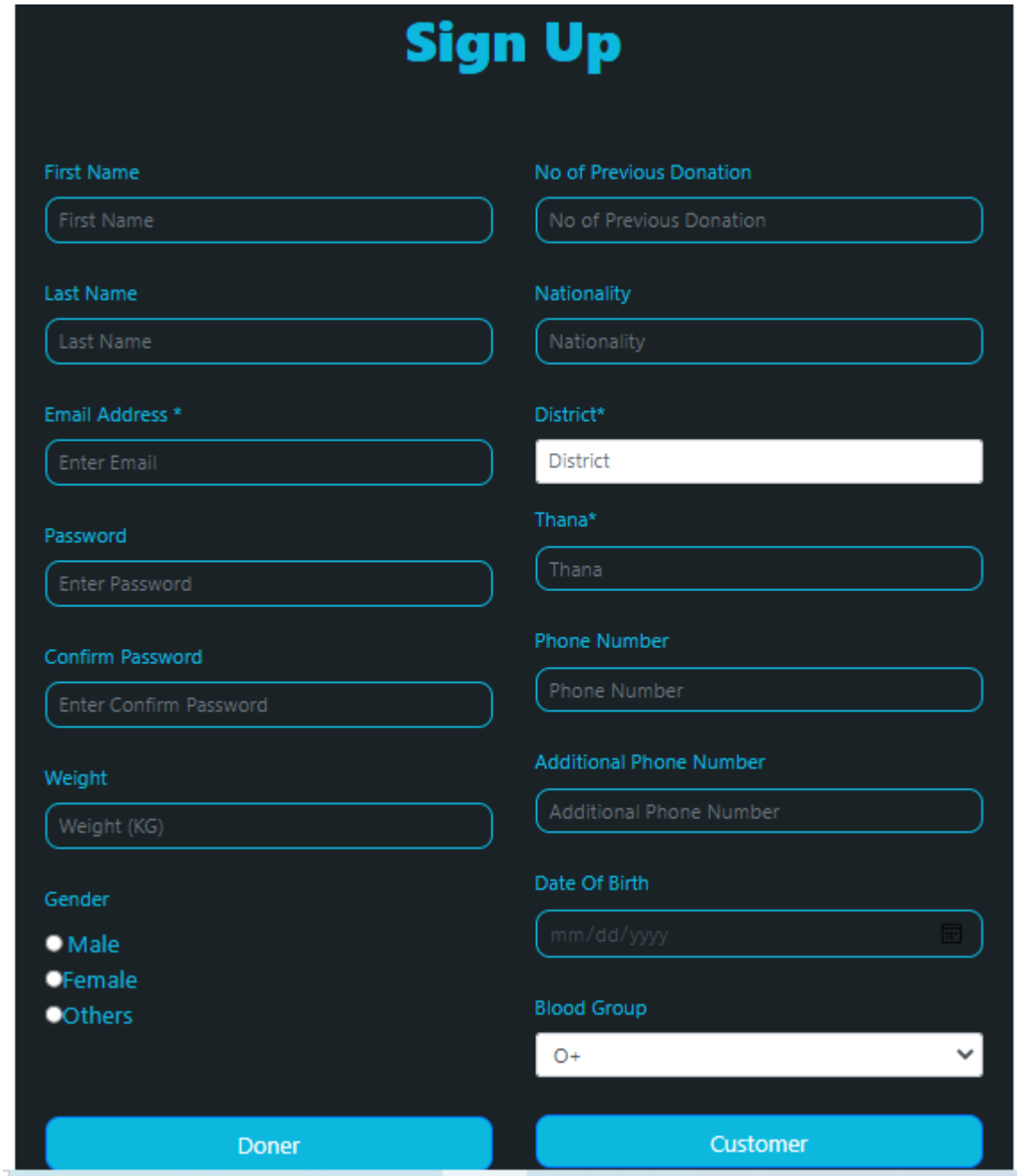
Edge: {(1,2)→3→(4,5,6,7)→8→9→(10,11,12)→13 →14→15→16→17→18→19 }
Edge: {(1,2) →3→18 →19,}
Edge: {(1,2)→3→(4,5,6,7) →8 → 9→ 16→17}
Edge: {(1,2)→3→(4,5,6,7) →8 →9 →(10,11,12) →13→14}
Edge: {(1,2)→3→(4,5,6,7) →8 →9 →(10,11,12) →13→15}
Similarly All condition and path is cover

6:Implementation:

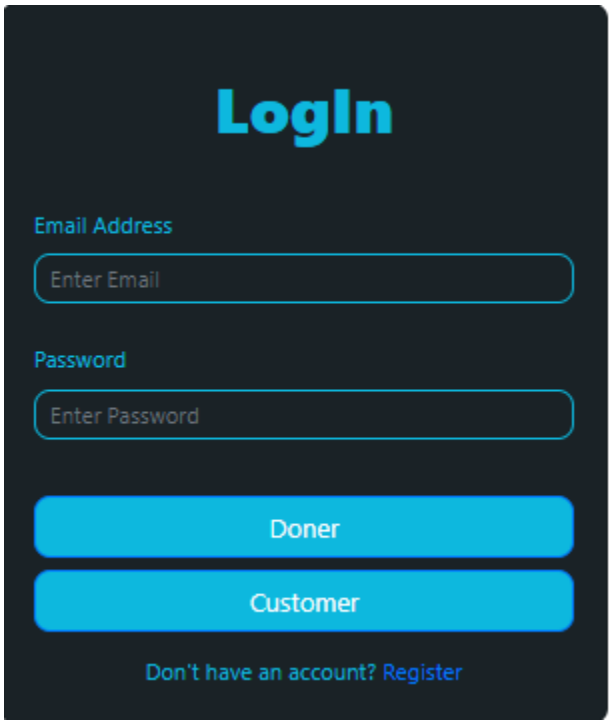
1.Home page



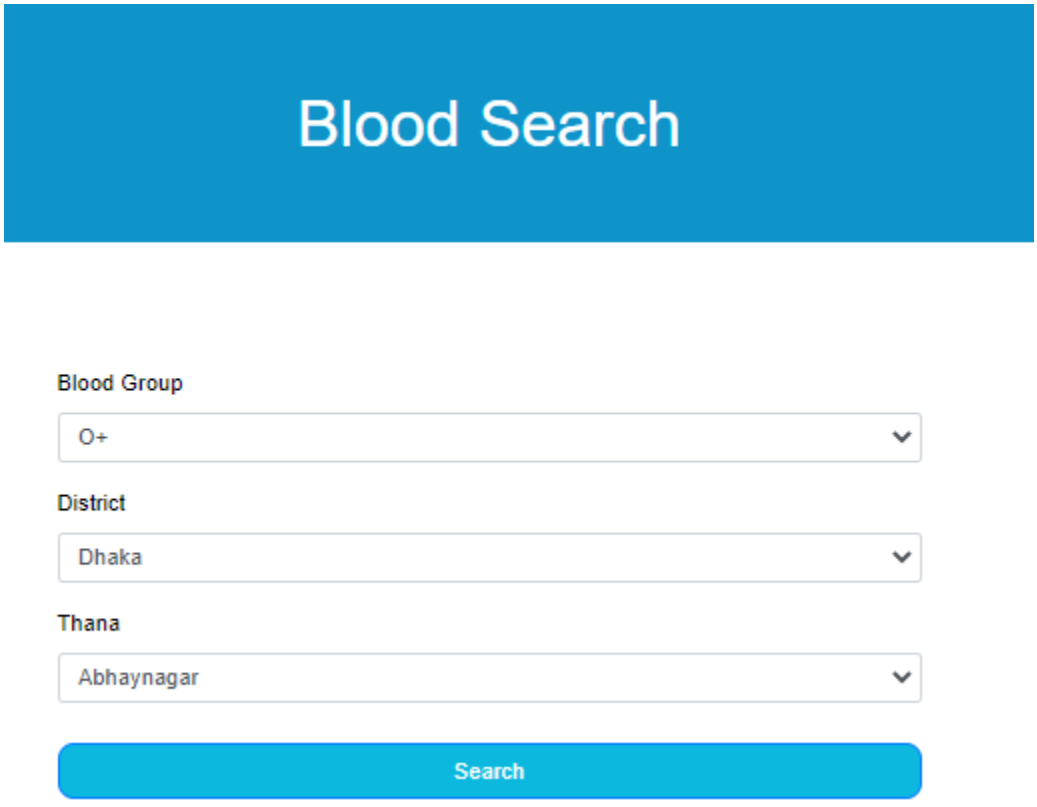
2.Signup:


Sign Up
First Name
First Name
Last Name
Last Name
Email Address *
Enter Email
Password
Enter Password
Confirm Password
Enter Confirm Password
Weight
Weight (KG)
Gender
Male
Female
Others
No of Previous Donation
No of Previous Donation
Nationality
Nationality
District*
District
Thana*
Thana
Phone Number
Phone Number
Additional Phone Number
Additional Phone Number
Date Of Birth
mm/dd/yyyy
Blood Group
O+
Doner
Customer

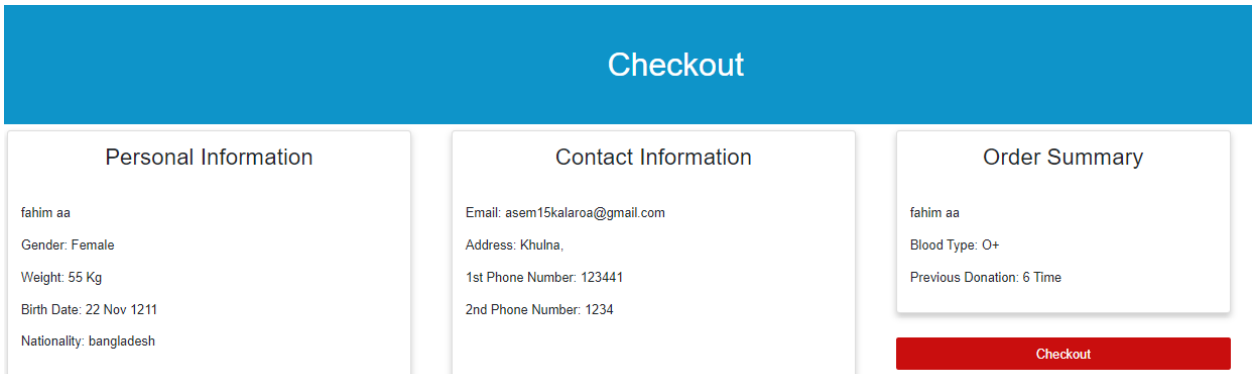
3.Login:



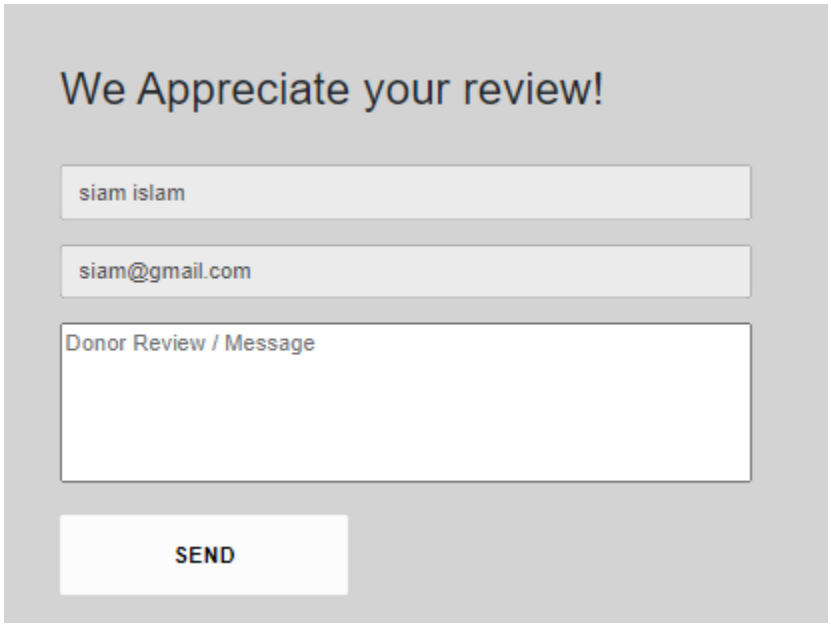
4.Search blood:



5.Checkout:



6.Review:



7.Website Admin: