

# BLOOD DONATION DATABASE

DONATE BLOOD SAVE LIFE

*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

# Problem Statement

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.

# Proposed System

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

# FEASIBILITY ANALYSIS

## **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.

## **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.

## **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.

# Requirement Analysis

## Functional Requirements

- 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.

# Requirement Analysis

## Non-Functional Requirements

- 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.

# Requirement Specifications

## 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

## Hardware Requirements

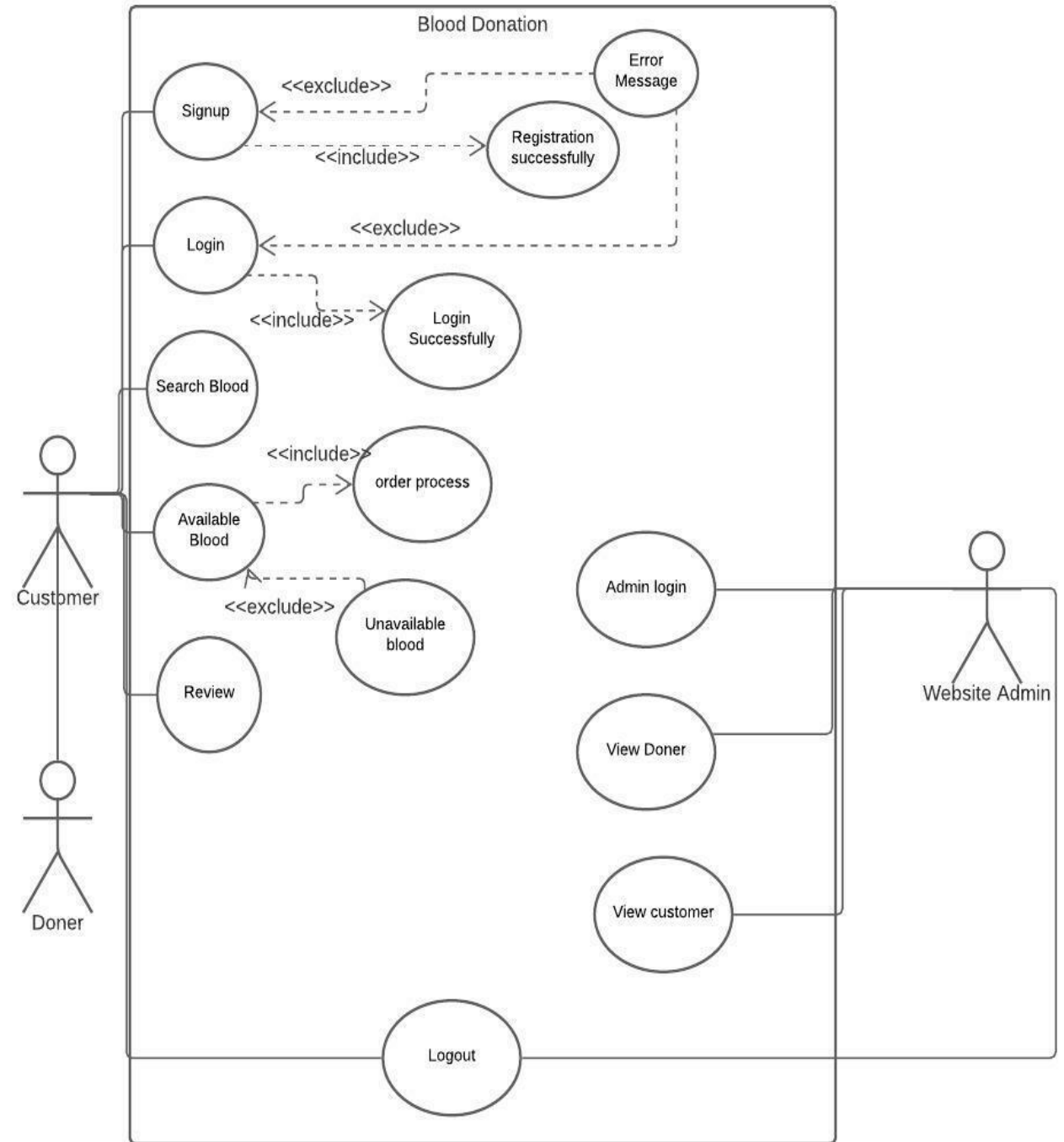
- Processor: Minimum Intel Core I3
- Processor speed: 2.50 GHZ
- RAM: 4 GB



# SYSTEM DESIGN

## Use Case Diagram

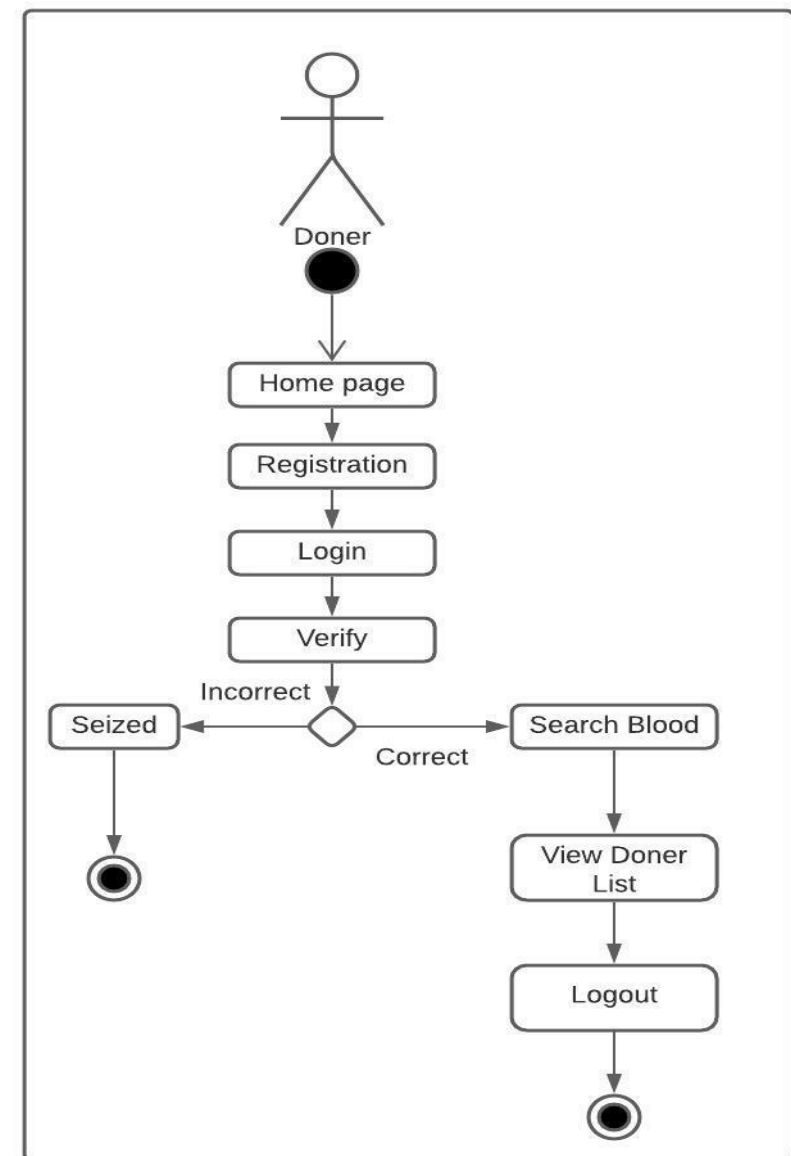
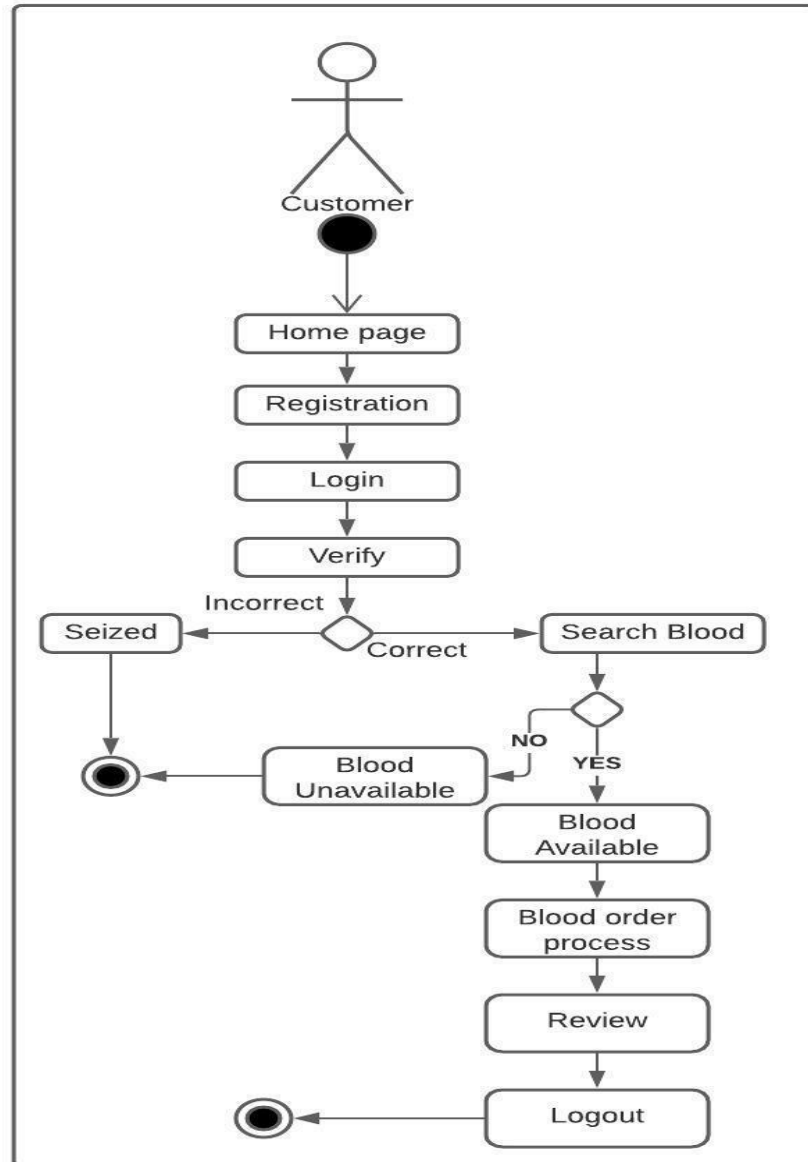
Although only comprehending a system's static existence is inadequate, Use-Case diagrams assist in providing a dynamic view of the system. Case diagrams are a type of diagram that depicts a situation. An application's architecture and subsystems. There are some external as well as internal variables to consider. The complex essence of the Use Case diagram is exemplified by this. Actors is what we term them.



# SYSTEM DESIGN

## Activity Diagram

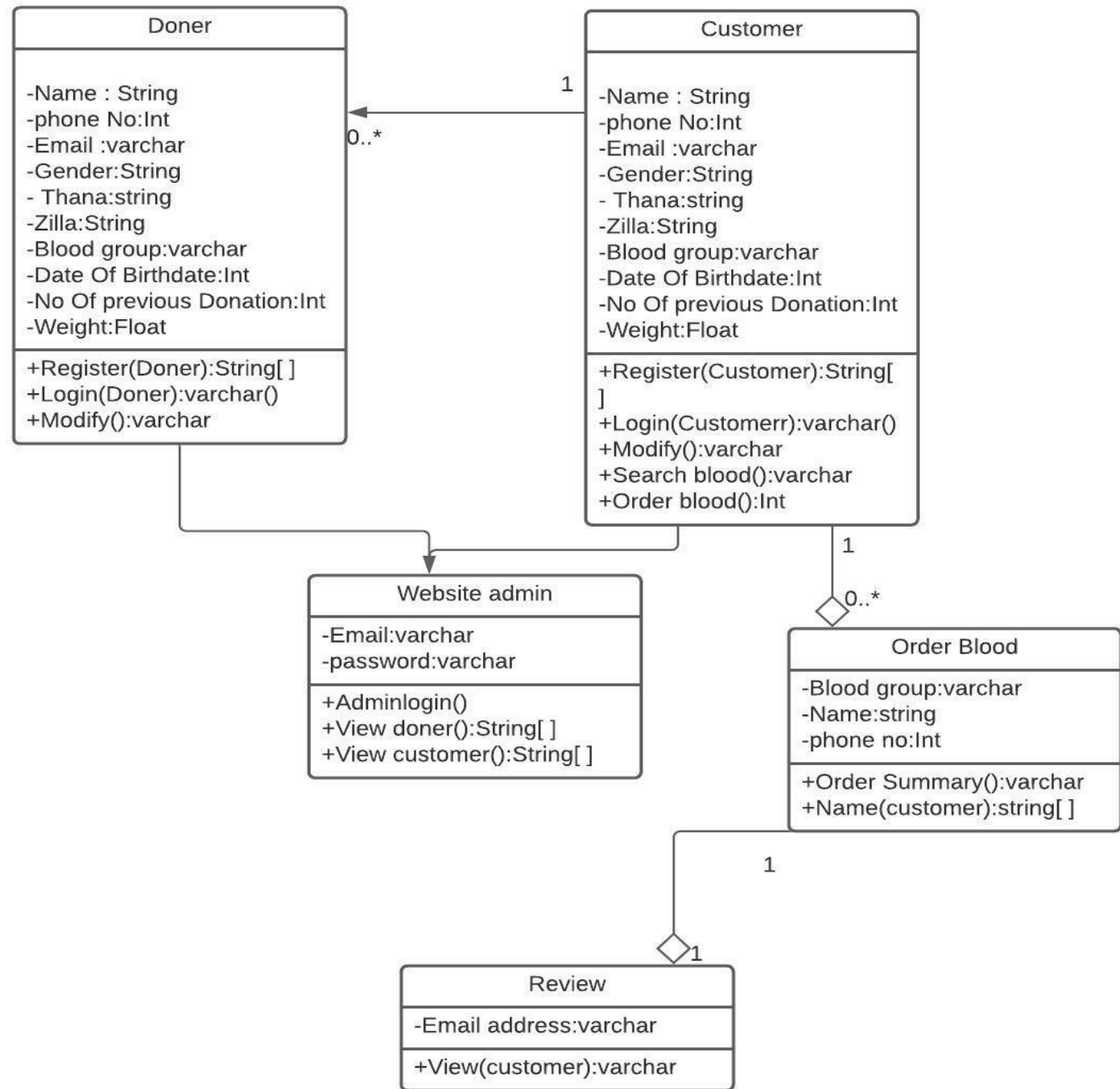
The Activity Diagram is another significant UML diagram that depicts the system's execution flow. Although activity diagrams are not exact flowcharts, they do have some features such as branching, swim lanes, and indicating parallel flows. It is a pictorial representation of a system's various operations that provides a holistic view



# SYSTEM DESIGN

## Class Diagram

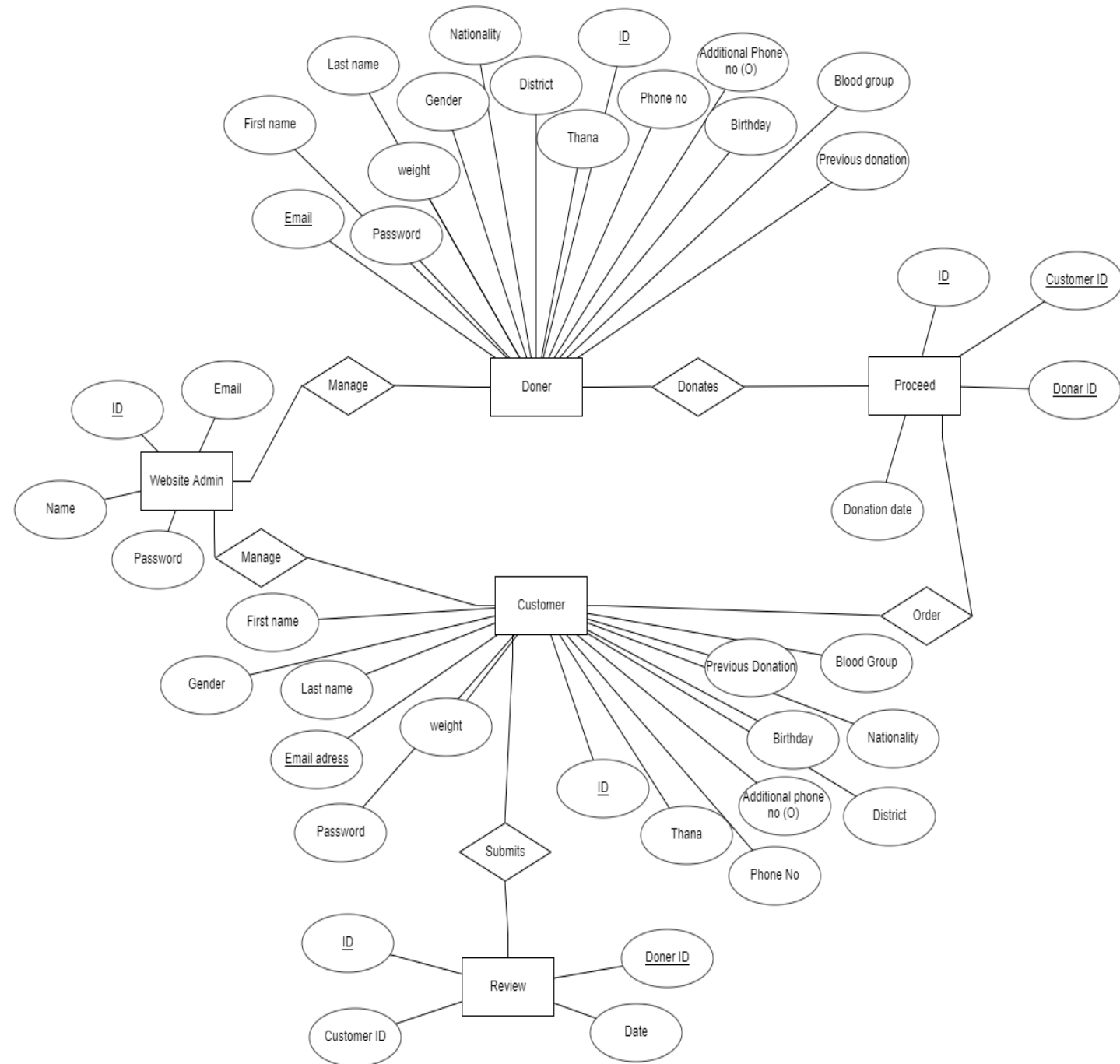
A class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's: classes, their attributes, operations (or methods), and the relationships among objects.



# SYSTEM DESIGN

## ER diagram

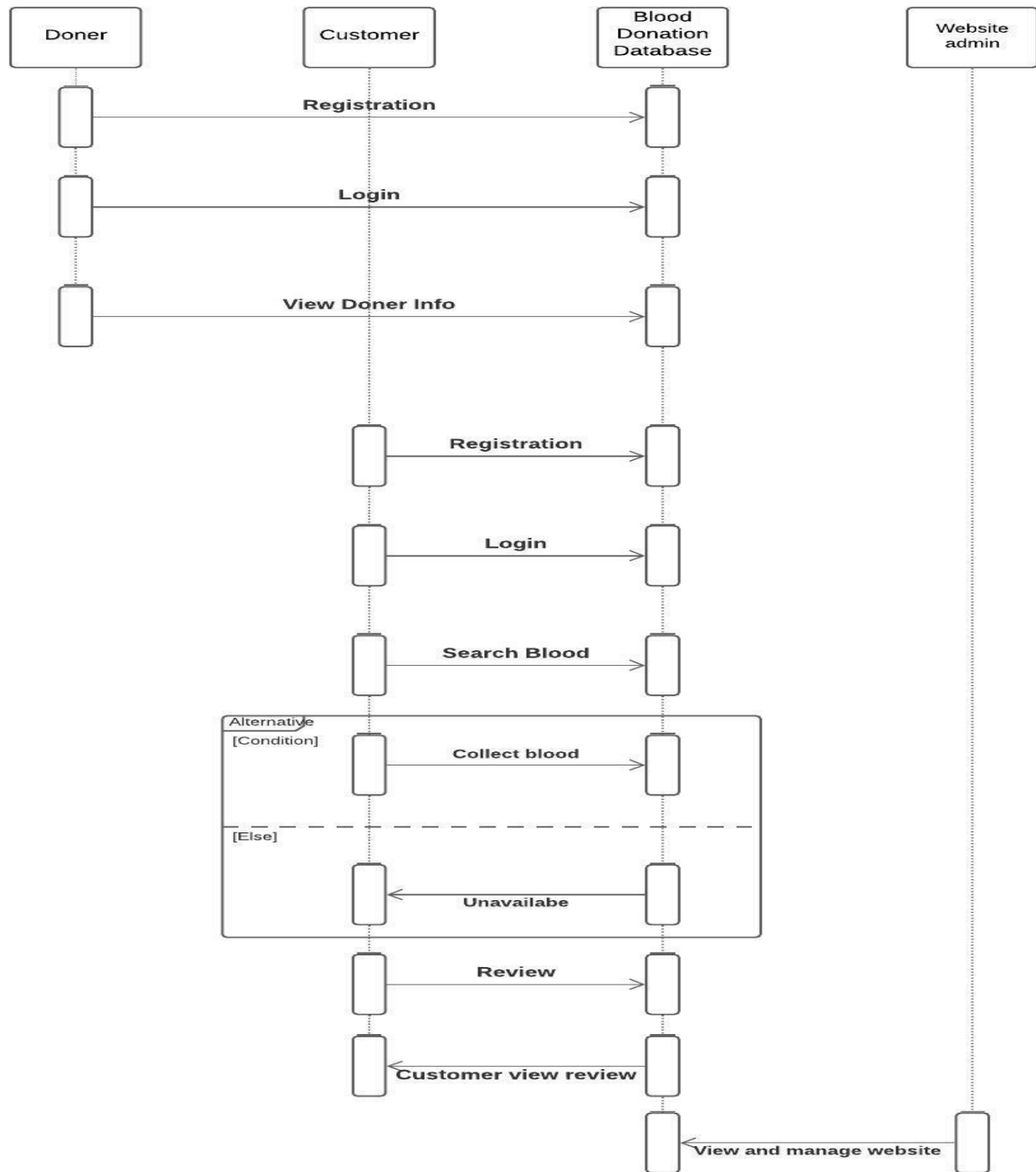
The entity-relationship diagram depicts all of the database tables as well as the relationships that exist between them. It also displays the cardinality of the tables, i.e. the many to one, one to one, or one to many relationships. This is the first step in the database design process. All of the specifications 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.



# SYSTEM DESIGN

## 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.

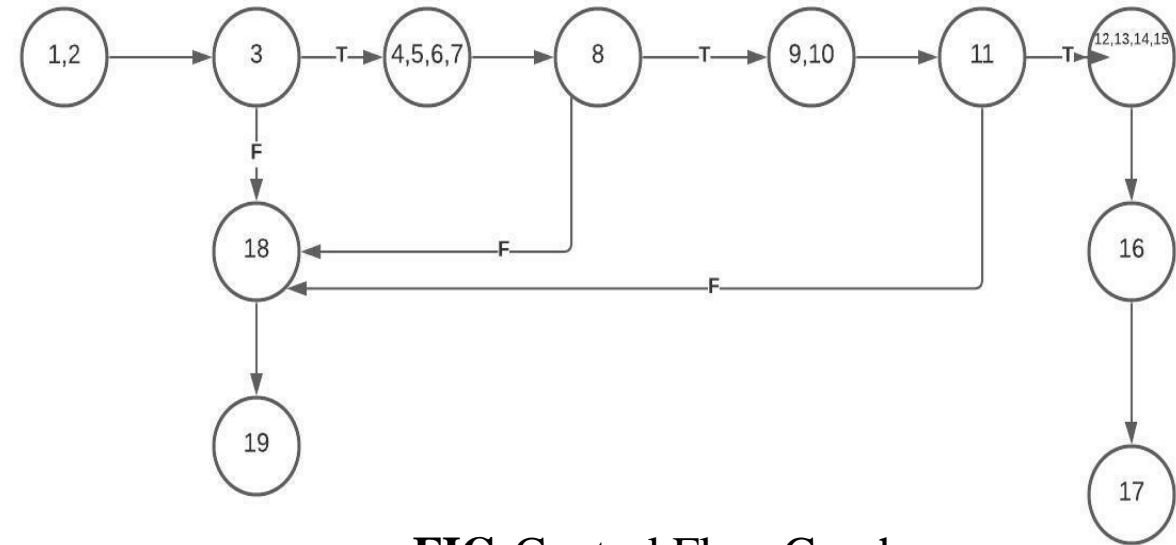


# TESTING OF SYSTEM

## 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
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

### Statement Coverage

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 )

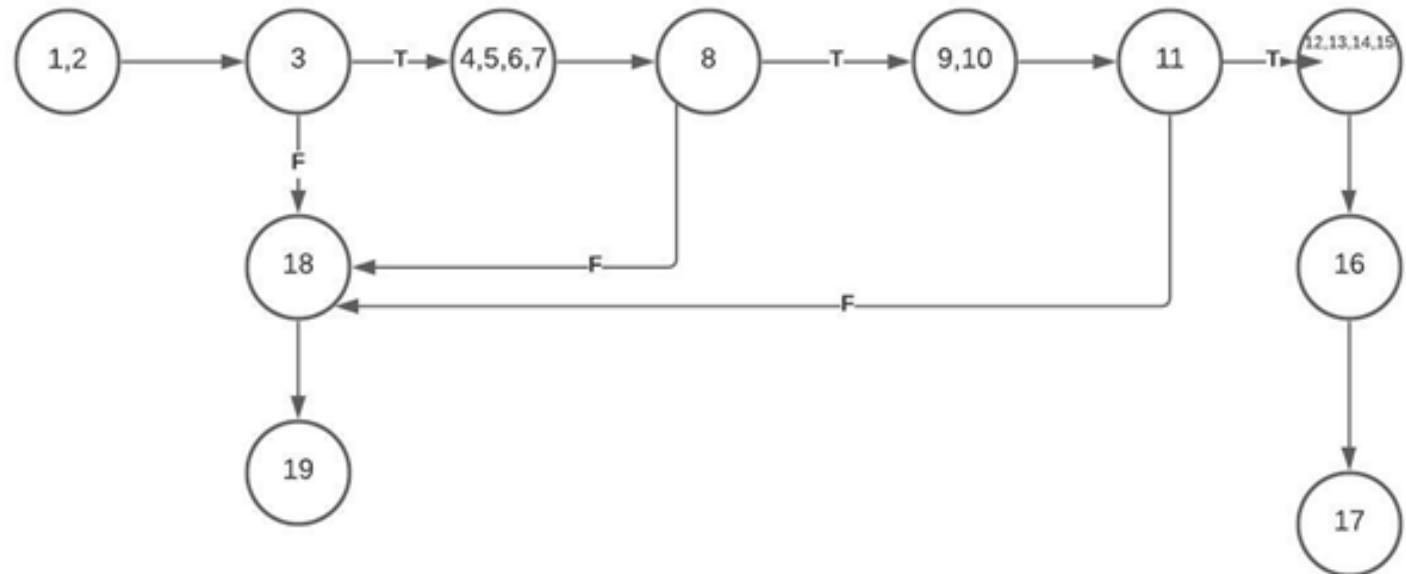
(\$POST['post'] = False) data posted error

(\$result = false) database error

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

(\$result = true) registration complete

(\$result = false) registration failed



## Testing Criteria

### Edge coverage



Edge:  $\{(1,2) \rightarrow 3 \rightarrow (4,5,6,7) \rightarrow 8 \rightarrow (9,10) \rightarrow 11 \rightarrow (12,13,14,15) \rightarrow 16 \rightarrow 17 \rightarrow 18 \rightarrow 19\}$

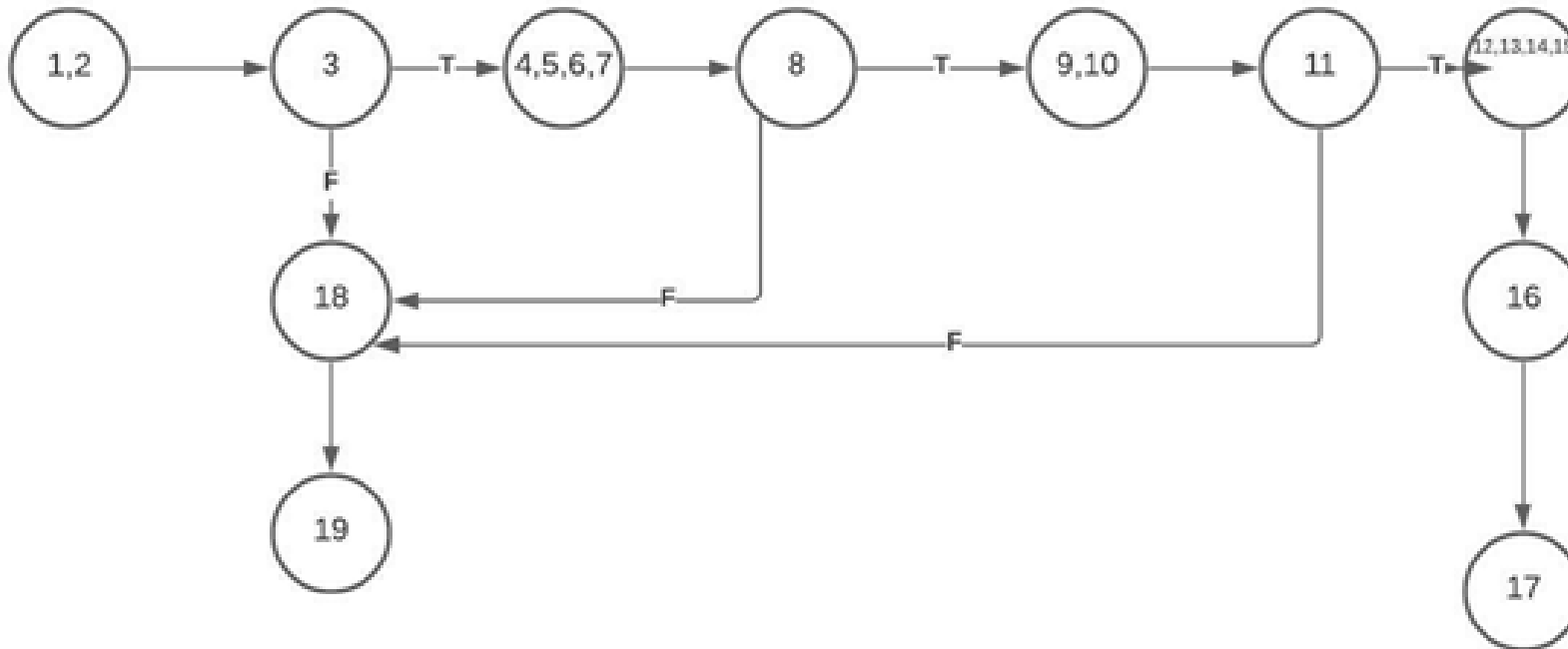
Edge:  $\{(1,2) \rightarrow 3 \rightarrow 18 \rightarrow 19\}$

Edge:  $\{(1,2) \rightarrow 3 \rightarrow (4,5,6,7) \rightarrow 8 \rightarrow (18,19)\}$

Edge:  $\{(1,2) \rightarrow 3 \rightarrow (4,5,6,7,8) \rightarrow (9,10) \rightarrow 11 \rightarrow (18,19)\}$

Edge:  $\{(1,2) \rightarrow 3 \rightarrow (4,5,6,7,8) \rightarrow (9,10) \rightarrow 11 \rightarrow (12,13,14,15) \rightarrow 16 \rightarrow 17\}$

Similarly All condition and path is cover



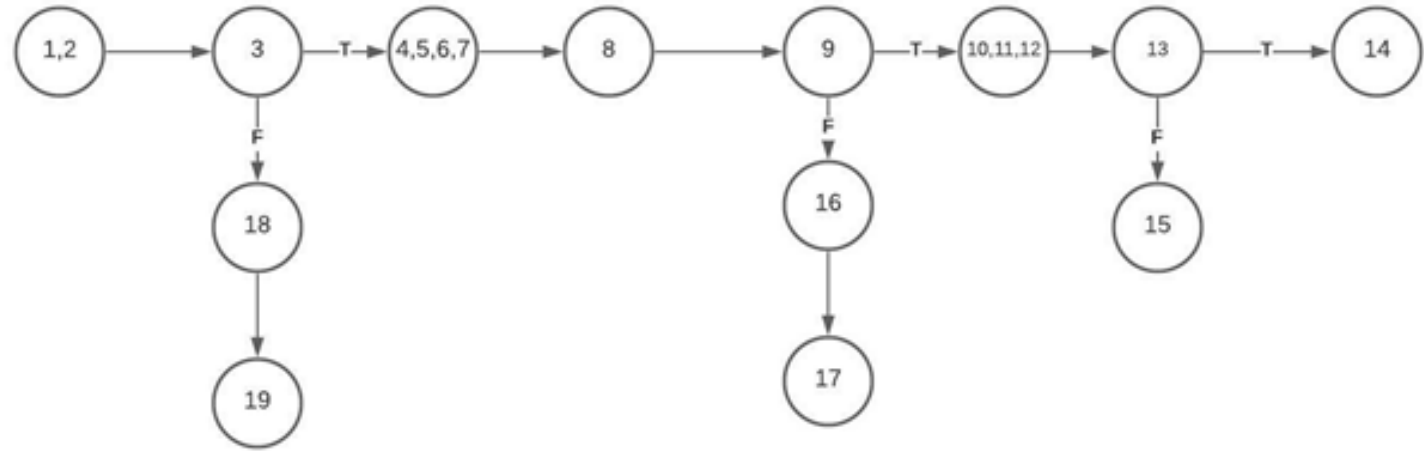


## Testing criteria

### White Box Testing

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.     $sql = "select * from doner where BloodGroup='".$_$blood.'" and
8.     (District = '".$_$district.'" and Address='".$_$upazila.'");
9.     if (empty($_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");
```



No of edge: 13

No of node: 12

Cyclometric complexity =  $13 - 12 + 2 = 3$

## Testing Criteria

### Statement Coverage

Example all-statements-adequate test set:

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

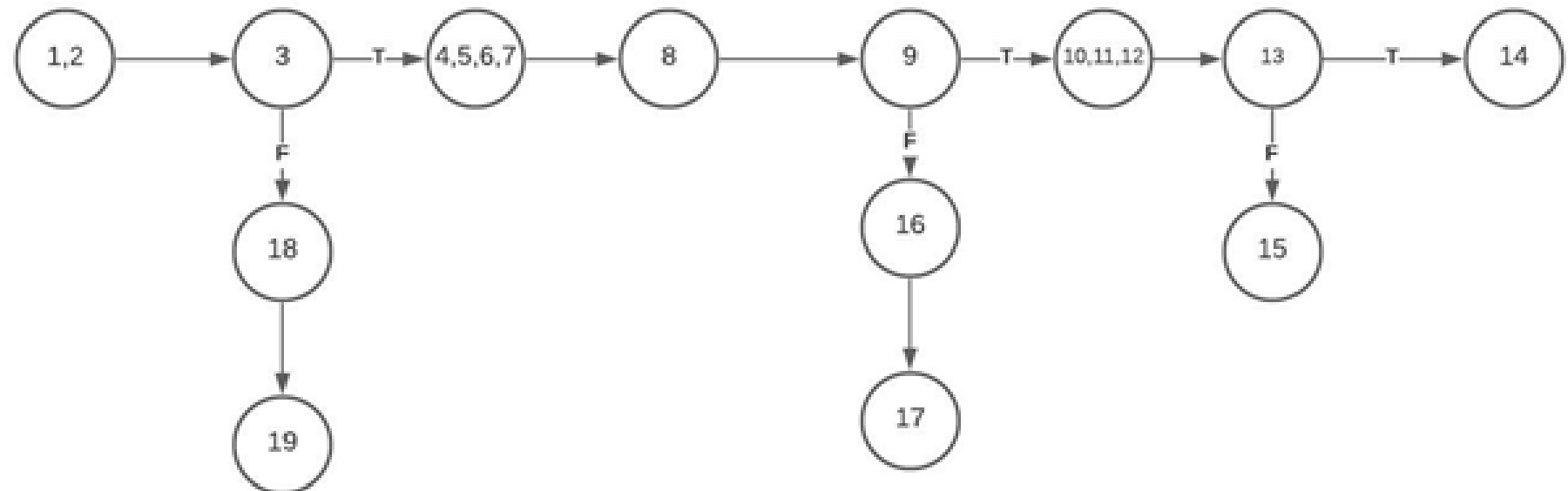
(fetch(\$r) != NULL , true)

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

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

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

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



## Testing Criteria

### Edge Coverage

Edge:  $\{(1,2) \rightarrow 3 \rightarrow (4,5,6,7) \rightarrow 8 \rightarrow 9 \rightarrow (10,11,12) \rightarrow 13 \rightarrow 14 \rightarrow 15 \rightarrow 16 \rightarrow 17 \rightarrow 18 \rightarrow 19\}$

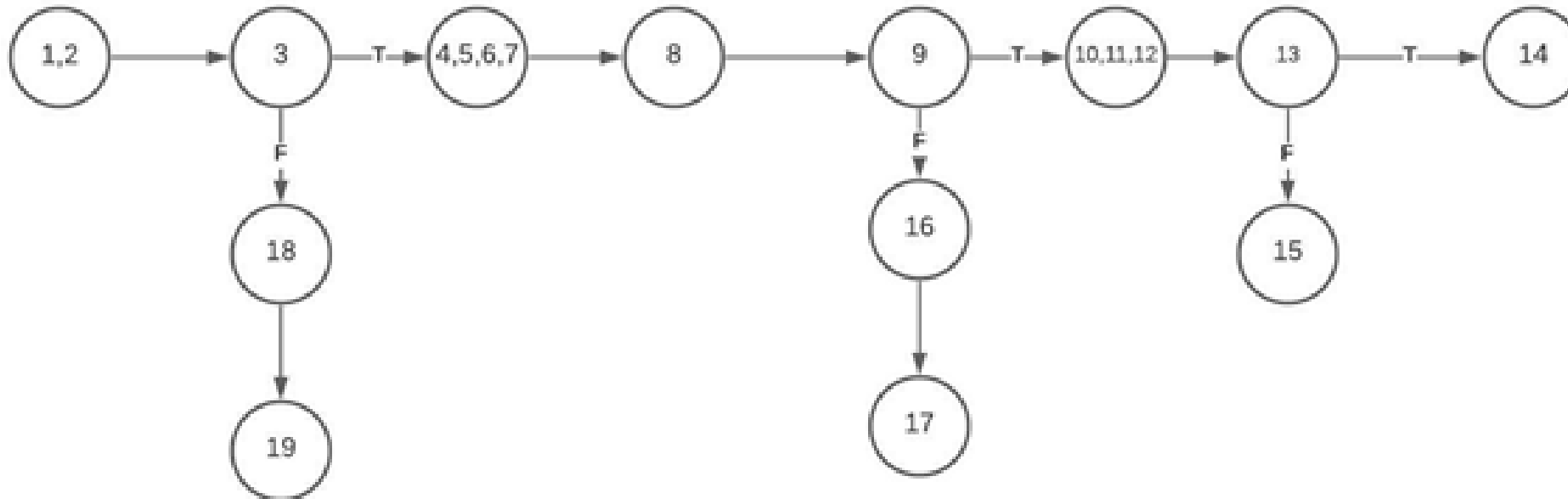
Edge:  $\{(1,2) \rightarrow 3 \rightarrow 18 \rightarrow 19\}$

Edge:  $\{(1,2) \rightarrow 3 \rightarrow (4,5,6,7) \rightarrow 8 \rightarrow 9 \rightarrow 16 \rightarrow 17\}$

Edge:  $\{(1,2) \rightarrow 3 \rightarrow (4,5,6,7) \rightarrow 8 \rightarrow 9 \rightarrow (10,11,12) \rightarrow 13 \rightarrow 14\}$

Edge:  $\{(1,2) \rightarrow 3 \rightarrow (4,5,6,7) \rightarrow 8 \rightarrow 9 \rightarrow (10,11,12) \rightarrow 13 \rightarrow 15\}$

Similarly All condition and path is cover



## Future Work

For future development we intend to add reservation option, password changing option, Map location, Donor and Customers profile and Mobile based, etc. This program has a lot of room for improvement and introducing new features and technology. Software using process and access should be easier, more relatable features should be added so that users can find this website more reliable. We will also try to increase performance and service quality.



**Thank You**