EAST WEST UNIVERSITY

A Project Report on

Blood Bank Management System

Course: CSE411

Title: Software Engineering & Information System Design

Section: 02

Semester: Summer 2020

Submitted To:

Dr. Mohammad Salah Uddin

Assistant Professor

Department of

Computer Science and Engineering

Submitted by:

Name	Id
Md. Moniruzzaman Shanto	2018-1-60-075
Nazmus Sakib	2018-1-60-104
Shimanto Krishna Chakroborty	2017-2-68-001

Date of Submission: 28/08/2020

Problem statement: People often face difficulties to find authentic blood bank sources as well as donors. Currently, our country has few rightful blood banks which operate offline. People run bank to bank to find desired blood packets. Therefore, we need a system where we can check and book our desired blood packets online and collect if enough blood is available. It will save time. It is also difficult to identify the donor with his real information which is very important in blood donation. Moreover, sometimes people cannot identify whether a donor has any kind of serious disease or not. It is also challenging to find Donner from the closest location because it saves time and life. In an emergency, collecting a large unit of blood is difficult because which hospital or organization has enough supply is difficult to identify. In case of emergency operation Sometimes people collect extra units of blood for safety. The remaining units of the blood sometimes go to waste because of not finding a patient in time. It is a challenge for the donor to verify if the requested person actually needs blood or not because sometimes third parties do business by collecting blood by providing wrong information. Donors Last donations of blood, age, BMI, and much more information is also very important in the blood donations. People also face different types of problems like payment and managing blood in time because of brokers. Online Blood Bank Management System which provides different types of information and services can be a lifesaver for people.

Proposed Solution: People need an automated Management system that is organized and easy to understand. From there people can easily view availability and book blood. Moreover, people need valuable information to establish trust and authentication. Blood Bank Management System can provide authentic information to lots of people at a time in an urgent moment. People from different areas can find available blood in the bank and free donors in a short amount of time by using this system. Moreover, a large unit of blood can be found from different areas by using this system.

Our plan is to develop a Blood Bank Management System which can help common people to donate, collect and provide information about the blood. This system will provide different types of options which will help people to identify authentic donors, can find doner from the closest location, can collect a large unit of blood, collect blood from the bank in an emergency, can know the last donation time, age, BMI, dates and many important information and in the end, it will help to create links between donors and patient and it will solve the issue between the people and brokers.

Feasibility:

Technical Feasibility: The system can be developed by using HTML, CSS, JavaScript, PHP, MYSQL. It needs a Web Server to serve its clients. Any type of browser can be used for viewing the system. For developing the system any type of notepad or text is enough. All the required software and hardware are available in the market. Therefore, the system is technically feasible.

Operational feasibility: When compared to the benefits received, the proposed system's cost is almost insignificant. Consumers benefit the most because they save the most of their time. Thus, the system is operationally feasible.

Economic feasibility: Because the essential hardware and software are not so high priced, the initial investment is very low and no additional improvements are required. As a result, it is cost-effective.

Requirements:

Functional Requirements:

1. Donor and seeker sign-up option using specific information then system store the information.
2. Admin and other users can log in using specific information and the system validate it.
3. Admin and other users can change their information (personal information, contact details, availability, password) that is stored in the system.
4. Admin panel has the option to delete and insert a donor or seeker.
5. Admin and other user's passwords stores using encryption algorithms.
6. Generate reports and only the admin can view them.
7. Some information in the registration form will be mandatory.
8. Show available donor lists to the consumer.
9. Show available types of blood to the consumer.

Non-Functional Requirements:

1. The system should be portable, moving from one browser to another does not create any problem.
2. The system should maintain information privacy and security.
3. The system should available all the time.
4. The system should track every mistake and take a log of it.
5. Without affecting the performance of the system, it should handle enough users (around 1 million).

Use case:

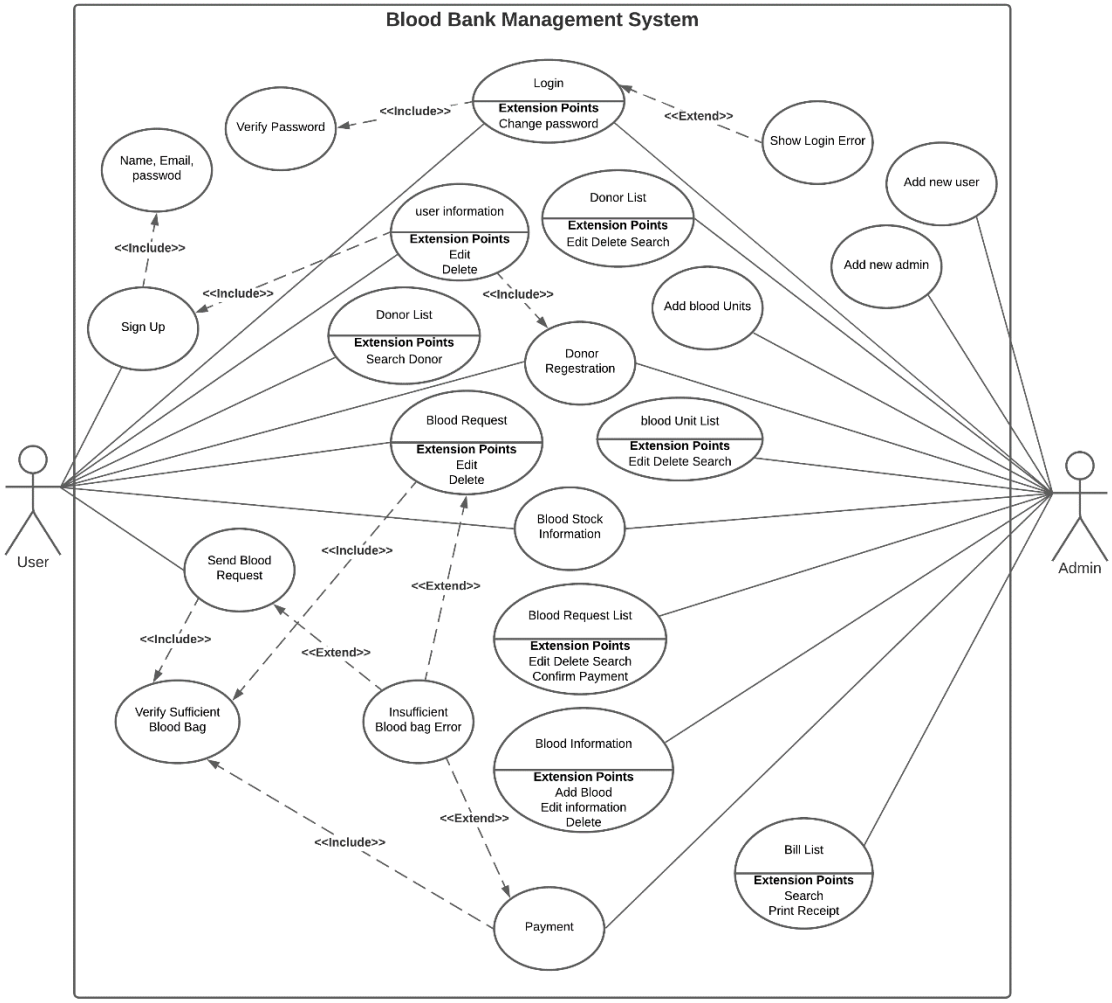


Figure 1:Use Case diagram

Use Case Description:

Use case name:	Login
Use case Description	This use case allows the user or admin to login into the web application to access all the services or functions. To login to the system users or admins have to enter a unique email and password. Admin or user can reset their password using the forgot password option. After entering the correct email and password system will display relevant profiles and options.
Related use case	1. Verify password 2. Show login error
Primary Actor	User, Admin
Secondary Actor	None
Preconditions	User or admin must have a valid account.
Postconditions	System displays User or admin profile.
Main Flow	1. Enters email 2. Enters password 3. Submits email and password 4. System validates email and password 5. System verifies email and password 6. System displays profile page
Alternative Flows	4.1. Missing email or password 1. System shows a missing alert. 2. Use case starts form step 1. 5.1. Invalid email or password 1. System shows Invalid email or password message. 2. System requests for email and password. 3. Use case resumes at step 1.

Use case name:	Sign up
Use case Description	This use case allows the user to create a new profile into the web application to access all the services or functions. To login to the system users have to sign up and create an account by entering a unique email, username and password. After entering a unique

	email, name and password, the system will display a new account created and will allow users to access relevant profiles and options.
Related use cases	1. Unique email, name and password.
Primary Actor	User
Secondary Actor	None
Preconditions	Users must not already exist.
Postconditions	A new user account is created and the system displays User profile and options.
Main Flow	1. Enters name 2. Enters email 3. Enters password 4. Submits name, email and password 5. System validates email and password 6. System verifies if an email does exist or not. 7. System displays profile page
Alternative Flows	5.1. Missing name or email or password 1. System shows a missing message. 2. Use case starts form step 1. 6.1. User already exists 1. System shows User already exists message. 2. System requests for another unique email and password. 3. Use case resumes at step 1.

Use case name:	User Information
Use case Description	This use case allows the user to see all his personal information. User can also update his personal information and also can delete personal information.
Related use case	1. Sign up use case 2. Donor registration use case.
Primary Actor	User
Secondary Actor	None
Preconditions	Users have to sign up and must have an account.

Postconditions	If a user edits or deletes his information it will be updated in the database and the system will show a message of the update.
Main Flow	1. System shows all the information the user provided. 2. System also gives option to edit and delete 3. User clicks edit 4. System requires name, email and other personal information. 5. System validates email and password 6. System verifies if an email does exist or not. 7. System displays a successful message. 8. User clicks delete 9. System asks if you really want to delete or not. 10. User clicks yes 11. System deletes information. 12. System displays delete successful messages. 13. User clicks no. 14. System does not delete information.
Alternative Flows	5.1. Missing name or email or password 1. System shows a missing message. 2. Use case starts form step 1. 6.1. User already exists 1. System shows User already exists message. 2. System requests for another unique email and password. 3. Use case resumes at step 4.

Use case name:	Send Blood Request
Use case Description	This use case allows users to send blood requests for specific blood groups. Users have to submit which blood group and how many blood bags needed. System will verify enough available blood bags in the blood stock and will compare requested bags and available bags. If there are not enough bags then the system will show not enough

	bags. Users also have to fill up the required date. After the required date, the request will be canceled.
Primary Actor	User
Secondary Actor	None
Related use case	1. Verify sufficient blood bag 2. Insufficient blood bag error.
Preconditions	Users have to be logged in and must have an account.
Postconditions	If the user sends a blood request system will save the request in the database and the system will show the request send message.
Main Flow	1. System requires blood group, quantity and date of collecting. 2. User submits blood group quantity and date of collecting data. 3. System verifies if there are enough available blood bags in the blood stock. 4. System displays a successful request message.
Alternative Flows	3.1. Not enough available blood bags in the blood stock. 1. System shows Not enough blood bag messages. 2. Use case starts form step 1.

Use case name:	Update Blood Request
Use case Description	This use case allows the user to Update blood request information. System allows the user to submit which blood group and how many blood bags needed. System will verify enough available blood bags in the blood stock and will compare requested bags and available bags. If there is not enough bag then system will show not enough bag message. Beside user can also update require date. After the require date, the request will be canceled.
Primary Actor	User
Secondary Actor	None
Related use case	1. Verify sufficient blood bag 2. Insufficient blood bag error.
Preconditions	User have to be logged in and must have an account.

Postconditions	If user Updates blood request system will save the request in the database and system will show request send message.
Main Flow	1. System requires Updated blood group, quantity and date of collecting. 2. User submits blood group, quantity and date of collecting data. 3. System verifies is there enough available blood bag in the blood stock. 4. System displays successful request message.
Alternative Flows	3.1. Not enough available blood bag in the blood stock. 3. System shows Not enough blood bag message. 4. Use case starts form step 1.

Use case name:	Blood stock Information
Use case Description	This use case allows user to see blood stock information. System allows the user see how many bloods bag is available for each blood group. It helps the user to know how many bags they can request from the blood bank.
Primary Actor	User
Secondary Actor	Admin
Preconditions	User have to be logged in and must have an account.
Main Flow	1. System will display blood stock table information.
Alternative Flows	1.1. If no information available for blood stock. 1. System shows No results message.

Use case name:	Donor List
Use case Description	This use case allows user to see Donor list. System shows donor name email, blood group and city information. User can search using any information line name, blood group or city name to find and contact with a donor.
Primary Actor	User
Secondary Actor	Admin

Preconditions	User have to be logged in and must have an account.
Postconditions	If user enters donor name, blood group or city name or any information about the donor that exist in the server system will show the donor information that match with the searched input
Main Flow	1. User enters information to search a donor. 2. System displays searched donor information.
Alternative Flows	2.1. If no information available for donor. 1. System shows No results message.

Use case name:	Donor registration
Use case Description	This use case allows user to be a donor. System asks to enter all the donor information like blood group, Birthday, gender, mobile number and city.
Primary Actor	User
Secondary Actor	Admin
Preconditions	User have to be logged in and must have an account.
Postconditions	After submitting the donor information system will display successful registration message.
Main Flow	1) Enters blood group, Birthday, gender, mobile number and city. 2) Submits the form. 3) System validates all information 4) System verifies all information 5) System displays successful message.
Alternative Flows	3.1. Missing any information 1) System shows missing message. 2) Use case starts form step 1.

Use case name:	Add new user
Use case Description	This use case allows Admin to add new user. System asks to enter all the required user information like name, email address and password.

Primary Actor	Admin
Secondary Actor	User
Preconditions	Admin have to be logged in and must have an admin account.
Postconditions	After submitting the user information system will display successful registration message.
Main Flow	1. System asks to enters name, email address and password 2. Admin submits the form. 3. System validates all information 4. System verifies all information. 5. System displays successful message.
Alternative Flows	3.1. Missing any information 1. System shows missing message. 2. System requests for all the information. 3. Use case starts form step 1. . 4.1. User already exists 1. System shows User already exists message. 2. System requests for another unique email. 3. Use case resumes at step 1.

Use case name:	Add new Admin
Use case Description	This use case allows Admin to add new Admin. System asks to enter all the required user information like name, email address and password.
Primary Actor	Admin
Secondary Actor	None
Preconditions	Admin have to be logged in and must have an admin account.
Postconditions	After submitting the admin information system will display successful registration message.
Main Flow	1. System asks to enter Admin name, email address and password. 2. Admin submits the form. 3. System validates all information 4. System verifies all information.

	5. System displays successful message.
Alternative Flows	3.1. Missing any information: 1. System shows missing message. 2. System requests for all the information. 3. Use case starts form step 1. . 4.1. Admin already exists: 1. System shows User already exists message. 2. System requests for another unique email. 3. Use case resumes at step 1.

Use case name:	Add Blood units
Use case Description	This use case allows Admin to add new blood bag information. System asks to enter all the required information like donor name, email address, mobile number, blood group, storing date and expire date of blood bag. After submitting the form system displays successful message.
Primary Actor	Admin
Secondary Actor	None
Preconditions	Admin have to be logged in and must have an admin account.
Postconditions	After submitting the blood unit information system will display successful submission message.
Main Flow	1. System asks to enter blood bag information like donor name, email address, mobile number, blood group, storing date and expire date of blood bag. 2. Admin submits the form. 3. System validates all information 4. System displays successful message.
Alternative Flows	3.1. Missing any information: 1. System shows missing message. 2. System requests for all the information. 3. Use case starts form step 1.

Use case name:	Blood unit list
Use case Description	This use case allows admin to see all his information of all the Blood unit. Admin can also update information of a Blood unit and also can delete a Blood unit from the system.
Related use case	Add blood unit.
Primary Actor	Admin
Secondary Actor	None
Preconditions	Admin have to sign up and must have an account.
Postconditions	If admin edit or delete Blood unit information it will be updated in the database and system will show message of the update.
Main Flow	1. System shows all the information of Blood unit in a table. 2. System gives option to edit and delete 3. Admin clicks edit 4. System requires blood bag information like donor name, email address, mobile number, blood group, storing date and expire date of blood bag. 5. Admin submit the form. 6. System validates information. 7. System displays successful message. 8. Admin clicks delete 9. System asks really wanted to delete or not. 10. Admin clicks yes 11. System deletes information. 12. System displays delete successful message. 13. Admin clicks no. 14. System does not delete information.
Alternative Flows	6.1. Missing any information 1. System shows missing message. 2. Use case starts form step 1.

Use case name:	Blood request list
Use case Description	This use case allows admin to see all his information of all the Blood request. Admin can also update information of a blood request and also can delete a blood request from server. Admin can search a blood request also. Admin can update payment status of a blood request.

Related use case	Add blood unit.
Primary Actor	Admin
Secondary Actor	None
Preconditions	Admin have to sign up and must have an account.
Postconditions	If admin edit or delete Blood request information it will be updated in the database and system will show message of the update.
Main Flow	1. System shows all the information of Blood request in a table. 2. System gives option to edit, delete, search and confirm payment of a blood request. 3. Admin clicks edit 4. System requires blood request information like booking id, email, blood group, quantity, request date, request expire date, payment status. 5. Admin submit the form. 6. System validates information. 7. System displays successful message. 8. Admin clicks delete 9. System asks really wanted to delete or not. 10. Admin clicks yes 11. System deletes information. 12. System displays delete successful message. 13. Admin clicks no. 14. System does not delete information. 15. Admin enters information to search Blood request. 16. System displays searched information. 17. Admin clicks confirm payment of a blood request. 18. System update payment status of a blood request. 19. System print receipts.
Alternative Flows	6.1. Missing any information 3. System shows missing message. 4. Use case starts form step 1. 16.1. If no information available for donor. 1. System shows No results message.

Use case name:	Blood Information
Use case Description	This use case allows admin to see all his information of all the Bloods. Admin can also update information of a blood and can delete also. System will display update or delete message.
Primary Actor	Admin
Secondary Actor	None
Preconditions	Admin have to sign up and must have an account.
Postconditions	If admin edit or delete Blood information it will be updated in the database and system will show message of the update.
Main Flow	1. System shows all the information of Blood a table. 2. System gives option to add, edit and delete blood information. 3. System requires blood information id, blood group, price, and details of a blood group. 4. Admin submit the form. 5. System validates information. 6. System displays successful message. 7. Admin clicks edit 8. System requires blood information id, blood group, price, and details of a blood group. 9. Admin submit the form. 10. System validates information. 11. System displays successful message. 12. Admin clicks delete 13. System asks really wanted to delete or not. 14. Admin clicks yes 15. System deletes information. 16. System displays delete successful message. 17. Admin clicks no. 18. System does not delete information.
Alternative Flows	5. or 10. Missing any information 1. System shows missing message. 2. Use case starts form step 1. 10. Fail to edit delete

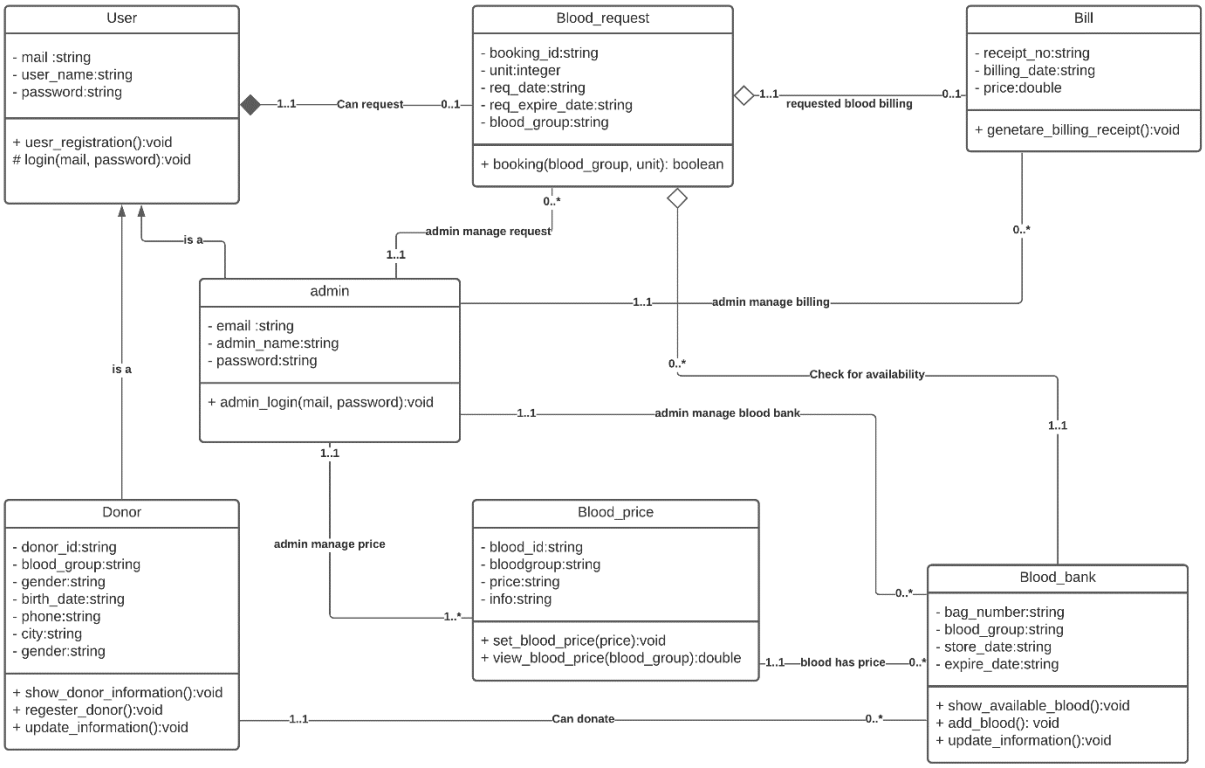
	1. System shows fail to update or delete message. 2. Use case starts form step 1.

Use case name:	Bill list
Use case Description	This use case allows admin to see all his information of all the payment and bill information in table. Admin can search any bill using any information into search bar. System will display search result of the searched input. Admin can also print receipt of any bill.
Primary Actor	Admin
Secondary Actor	None
Preconditions	Admin have to sign up and must have an account.
Postconditions	If admin click print receipt of any bill system will print the bill information and will display the information.
Main Flow	1. System shows all the information of Bill list in table. 2. System shows option print receipt. 3. Admin clicks print receipt 4. System selects the specific bill information and print it.
Alternative Flows	4.1. If System cannot select bill information and cannot able to print. 1. System shows error message.

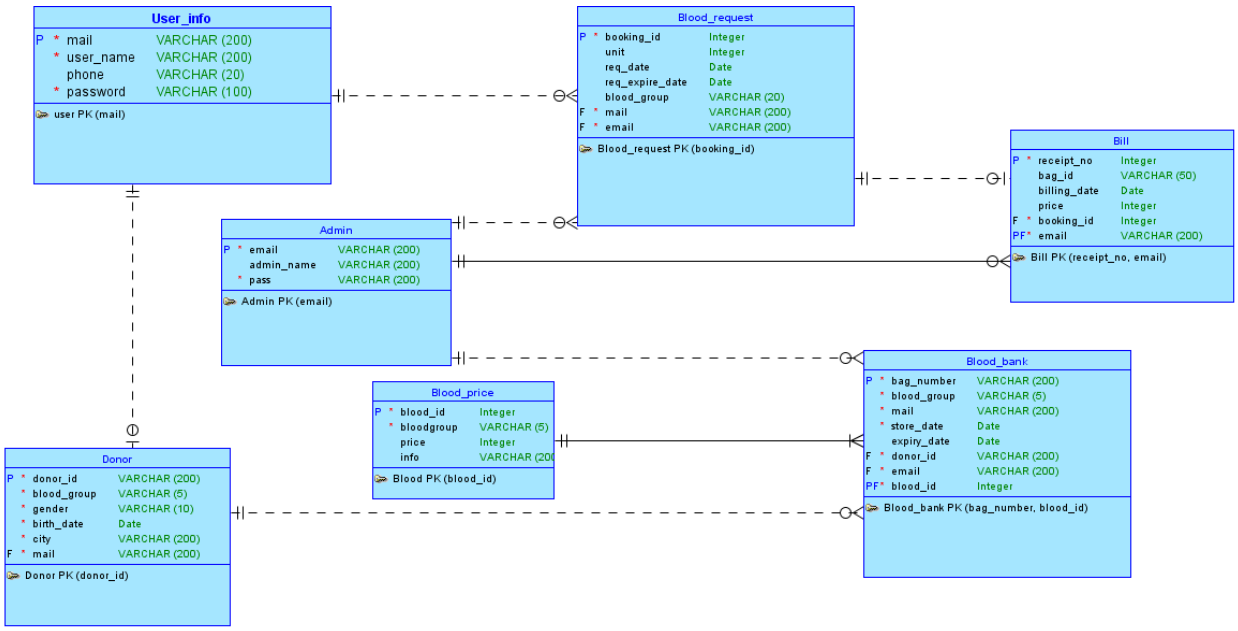
Use case name:	Payment
Use case Description	This use case allows admin to enter payment information of selling Blood units. Admin submits information of selling blood unit which are buyer name, email, phone number, blood group, quantity, and total price will be automatically calculated.
Related use case	1. Verify sufficient blood bag.

	2. Insufficient blood bag error.
Primary Actor	Admin
Secondary Actor	User
Preconditions	Admin have to sign up and must have an account.
Postconditions	If admin submits information of selling blood unit it will be updated in the database and system will show message of successful submission.
Main Flow	1. System requires buyer name, email, phone number, blood group, quantity 2. System will calculate total price automatically. 3. Admin submit the form. 4. System validates information. 5. System saves the information in database. 6. System displays successful message.
Alternative Flows	6.1. Missing any information 5. System shows missing message. 6. Use case starts form step 1. 16.1. If saves information in database fails. 1. System shows error message.

UML Class Diagram:



ER Diagram:



Sequence diagram:

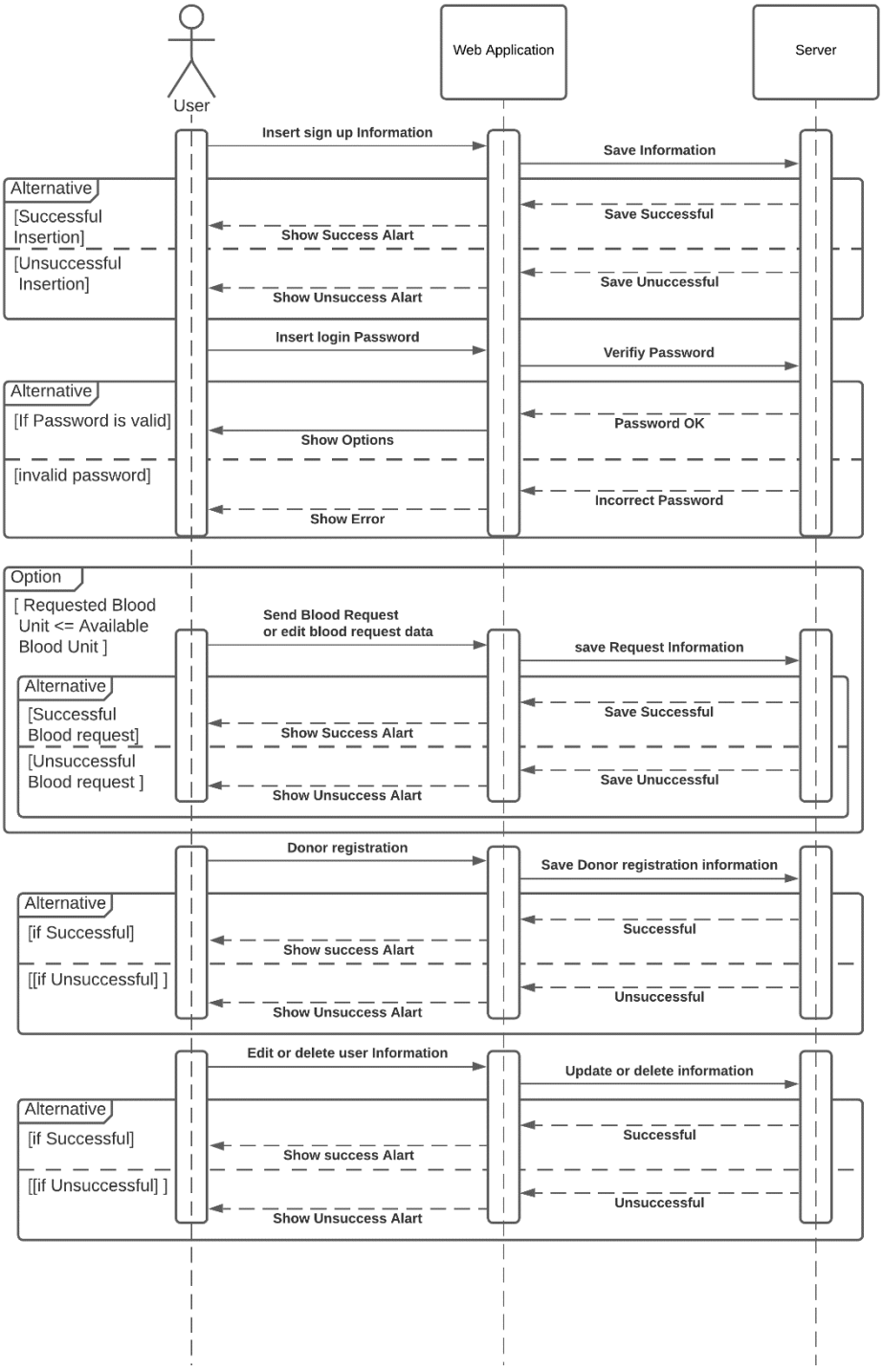


Figure 2:Sequence diagram(user)

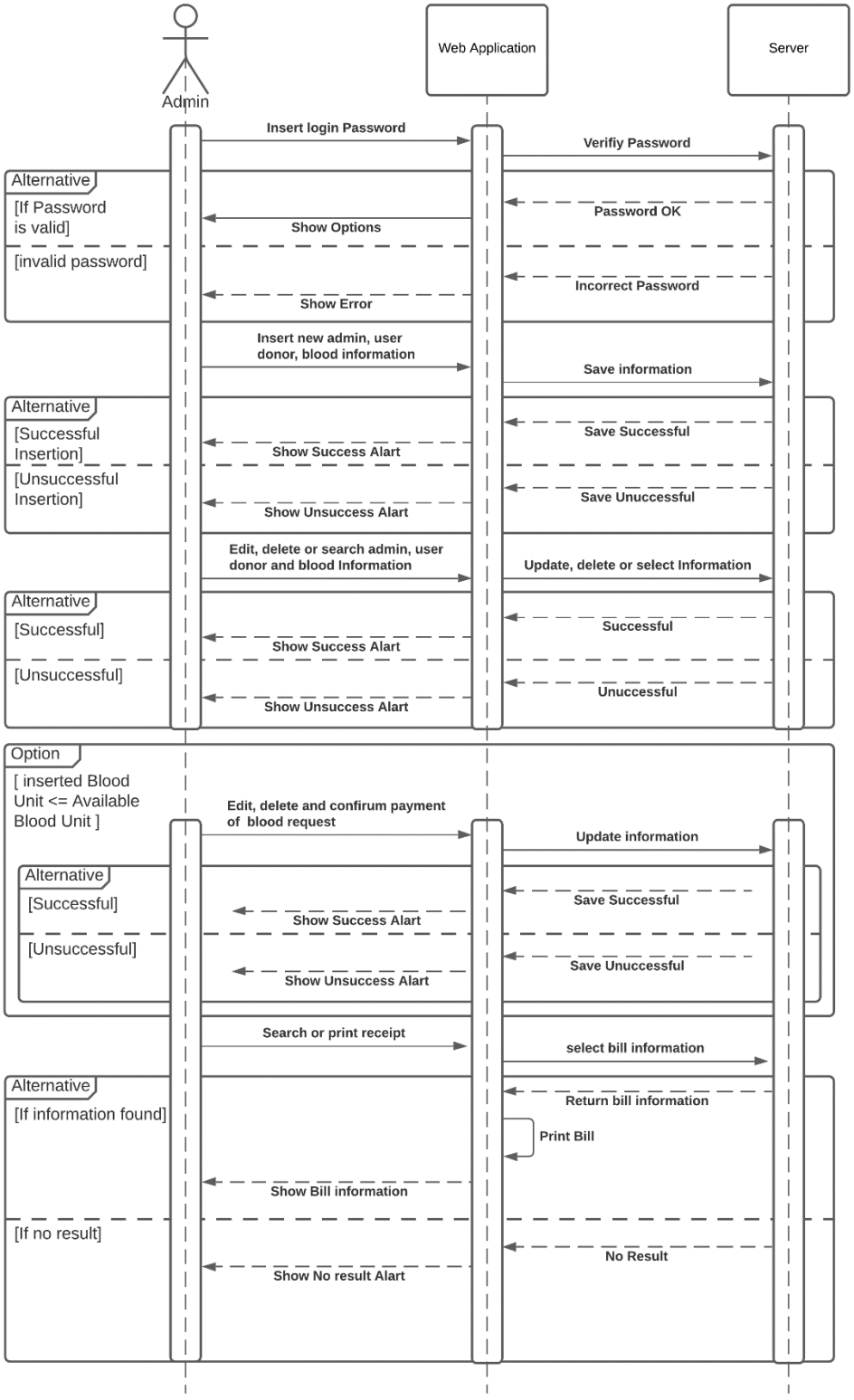
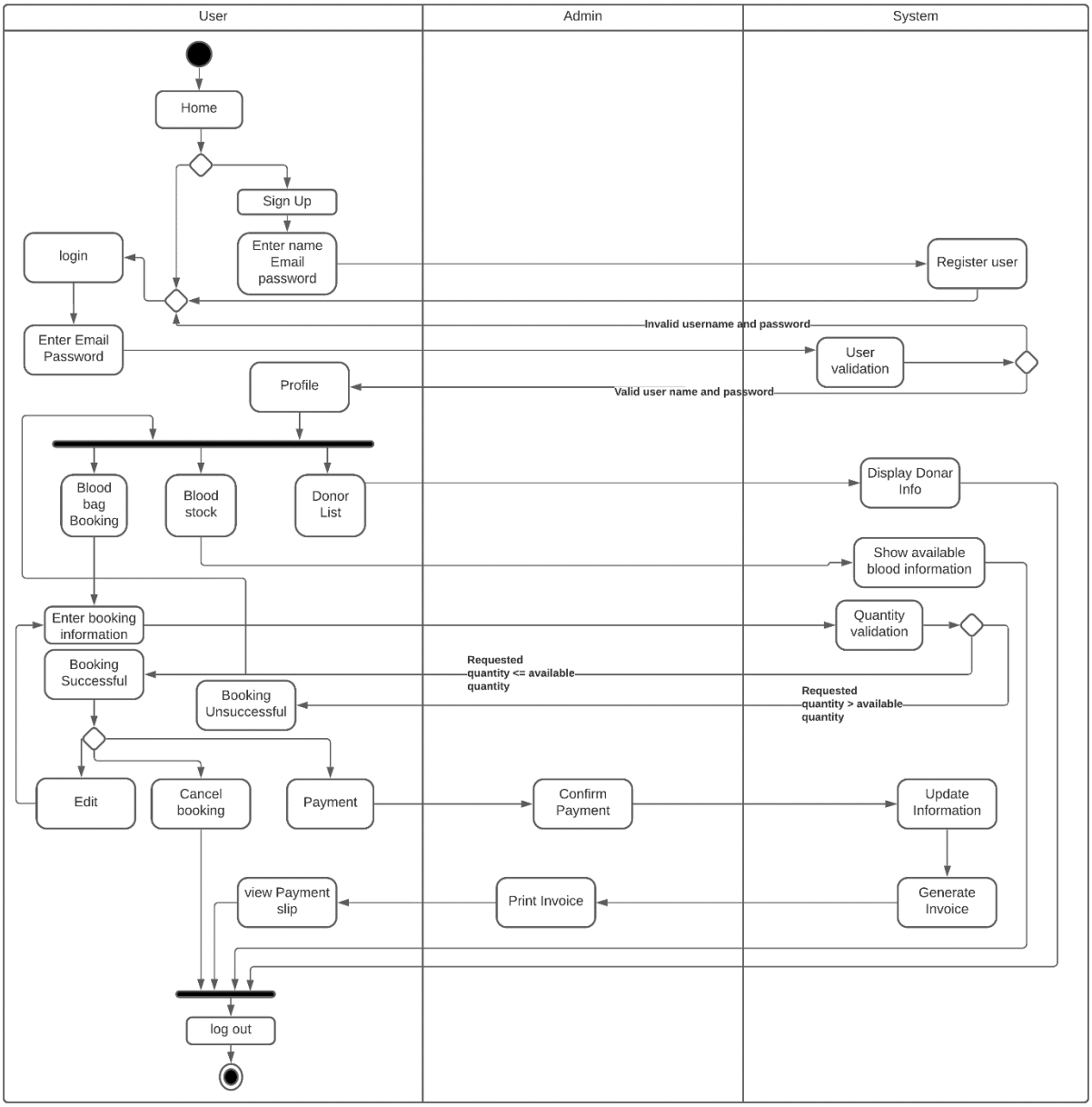


Figure 3:sequence diagram (admin)

Activity diagram:



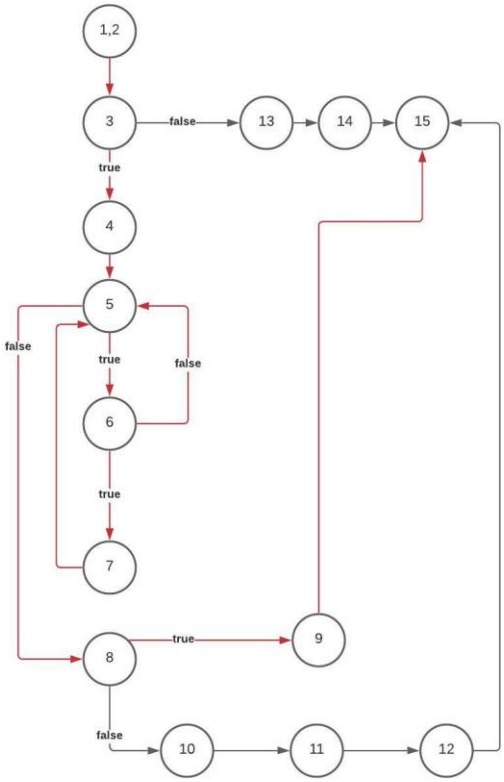
Test cases (white box method):

- Test case 1:

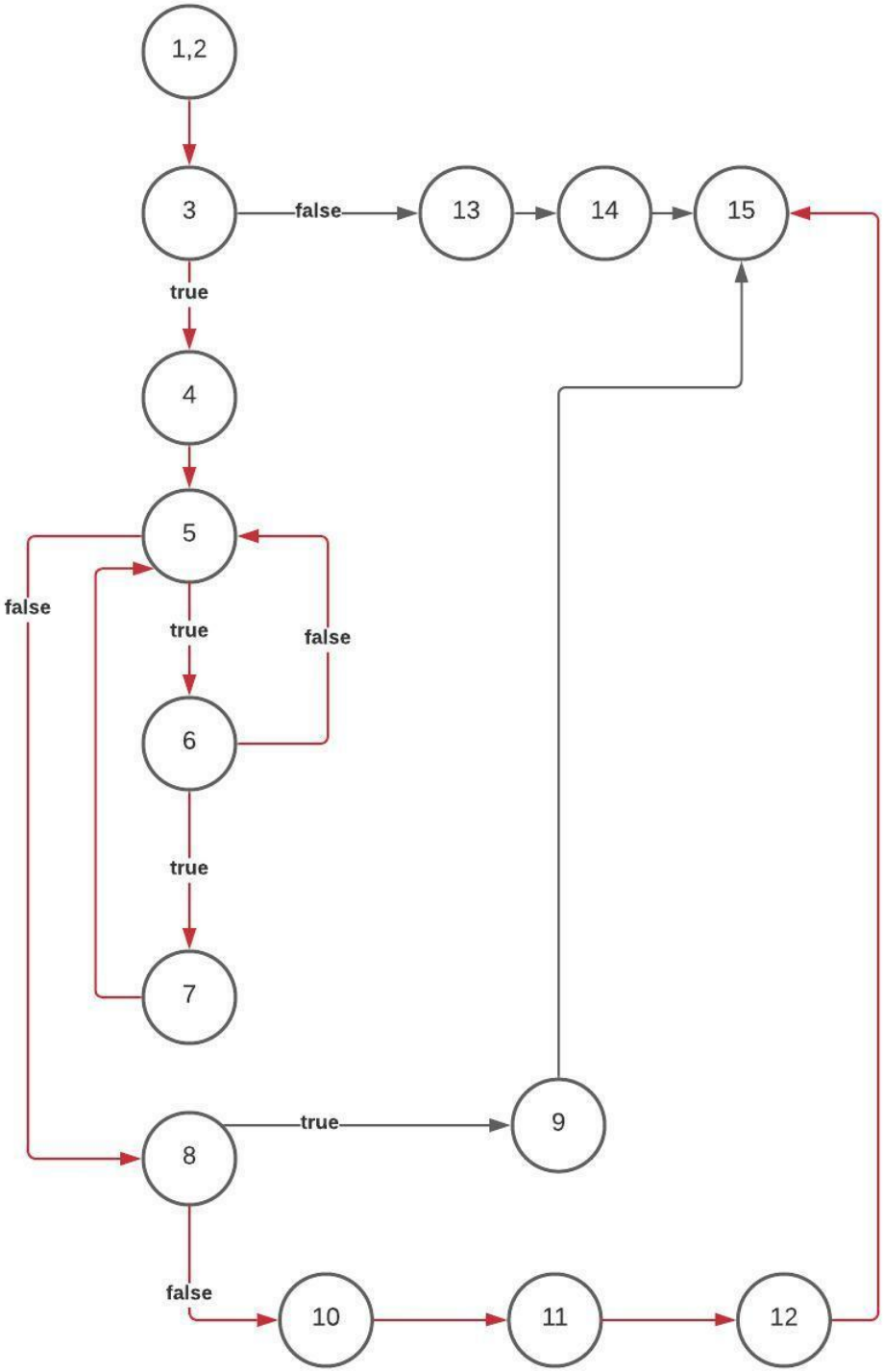
booking testcase.txt

```
1  function blood_request
2  begin
3  If($loggedin == true)
4  {   $available_bag = 0
5      for($i =0; $i<$bank.size(); $i++)
6      {   if($bank[i].group == $requested_group)
7              $available_bag++;}
8      if ($available_bag < $requested_unit)
9      {   echo "show Not Enough Blood Bag";}
10     else
11     { $available_bag == $available_bag - $requested_unit;
12         echo "Request successfuly added";}}
13 else{
14     echo "you mast login first";}
15 end
```

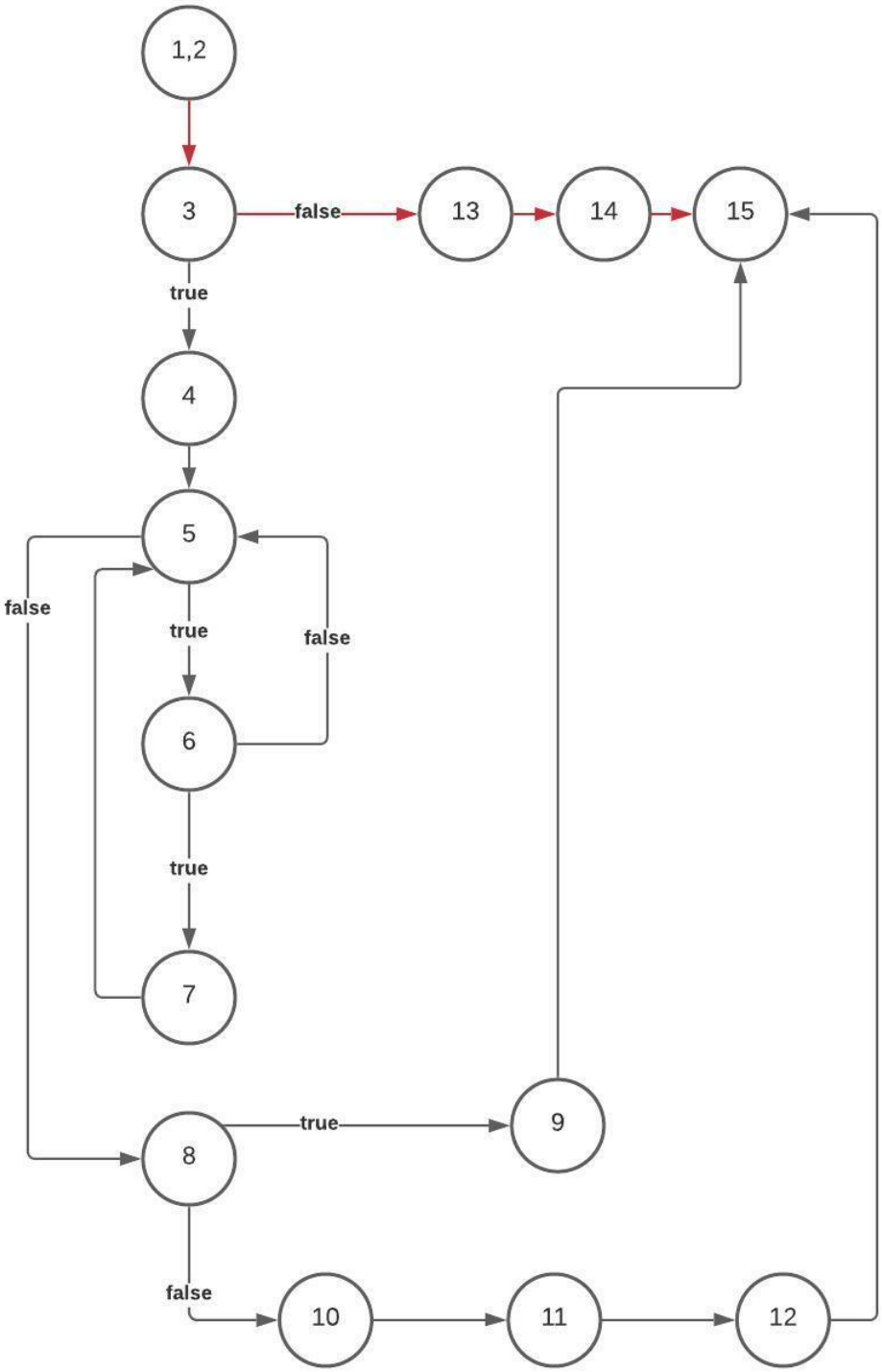
1. $loggedin == true
   i<$bank.size()
   $available_bag < $requested_unit



2. $loggedin == true
   i<$bank.size()
   $available_bag > $requested_unit



3. $loggedin == false

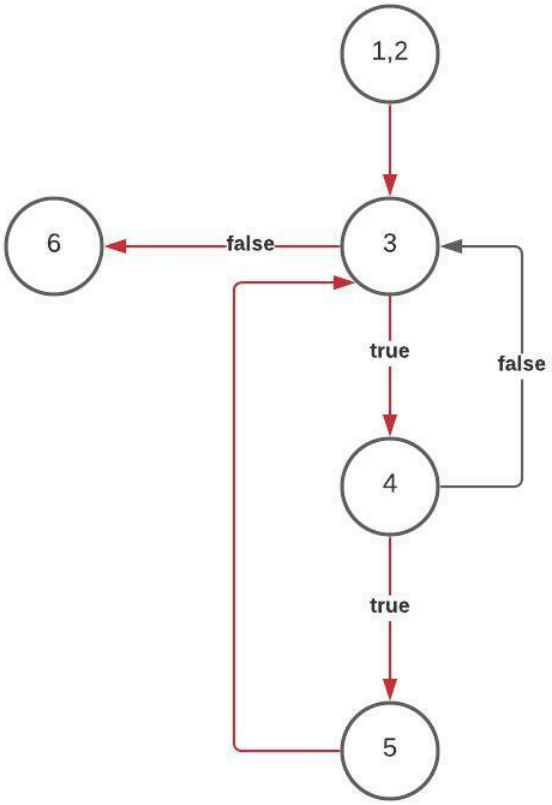


❖ Test case 2:

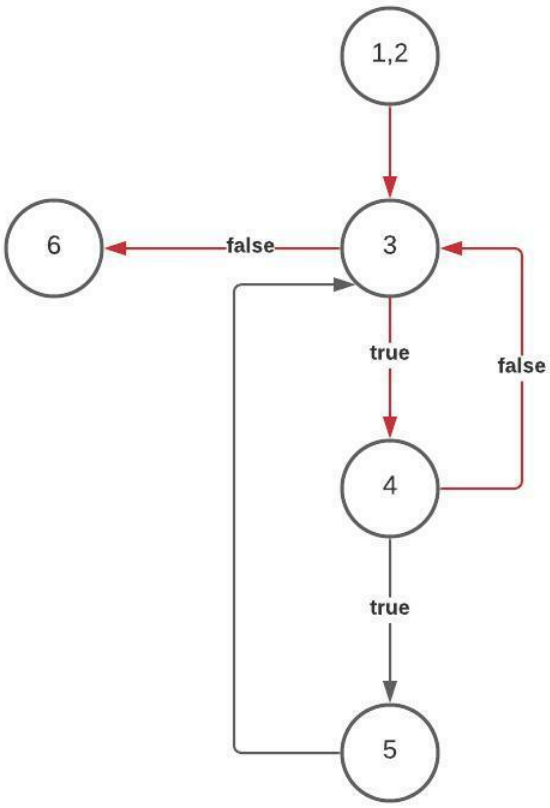
deletebloodgroup.txt

```
1  function delete_expired_blood
2  begin
3      for($i=0; $i<$bank.size(); $i++)
4      {if($bank[i].expiredate > $current_date)
5          discard_blood(i);}
6  end
```

1. $i<$bank.size()
   $bank[i].expiredate > $current_date



2. $i<$bank.size()
   $bank[i].expiredate < $current_date



Reference:

[1] Bangladesh is still to meet the demand of safe blood supply (who.int)