**AMERICAN INTERNATIONAL UNIVERSITY-**

**BANGLADESH**

***Faculty of Science and Technology***

***Final-Semester Assignment***

**Assignment Topic: Online Blood Bank Management System**

*Submitted by-*

***Group no.-*** 19

***Group members-***

1. **Name-** Ruksat Khan Shayoni [**ID-** 20-41922-1]
2. **Name-** Oliur Rahman Oli [**ID-** 20-41838-1]

***Course-***  Object Oriented Analysis & Design

***Section-*** D

***Semester-*** Spring 20-21

*Submitted to-*

S.A.M Manzur H. Khan

Associate Professor,

Department of CSE,

FST, AIUB

**Brief Description:**

‘We for Us’ is an Online Blood Bank Management System. In this system, there are 3 types of persons- user, admin and volunteer. All of the persons are considered as the registered members of the system and can logon to the system only and only by providing correct email and password. Login is being verified by cross checking the information with the database. In case of incorrect email or password, access to the portal will be denied.

After successful login, a user will be able to see the portal dashboard where he can find creating donation request and checking pending request option. For creating request, the user has to choose the blood group among A+, A-, B+, B-, O+, O-, AB+ or AB and provide some other information and the post the donation request. And for checking pending requests, the user should go to that particular option of the dashboard and check status, if the status says that volunteer is not found then he must delete the request, so his request will be **archived**, and if the status says the request is processing then he must wait for the admin to respond and if the status says that volunteer has been found, then he should go for the option show volunteer information and contact the volunteer.

After logging in, the admin can find the donation request created by user through the portal. As per the donation requests, he will search for eligible volunteer who is available for donation. If a volunteer is found available then forward the volunteer information to the user and if a volunteer is not available then a notice will be sent that volunteer is not available.

After being registered, all volunteers are initially set up as general volunteer. They have to login to the portal and find out the user who trying to reach them for donation. When they participate in donation for the first time up to five times they are considered as active volunteer. And those who participated in donation more than five times they are considered as executive volunteers and they receive award, but if they miss participation for two times they will be set back to active volunteers. If a volunteer doesn’t participate in donation after being eligible for donation for twice, he’ll be considered as an inactive volunteer and if an inactive volunteer fails to participate in donation for once, he’ll be **expelled** and will be **never taken back in the system** but if he responds at the third time he’ll be set back to active.

**Sequence Diagram 1: ‘Create Request’**

**:Database**

**:Portal**

**:User**

1. login(email, password )

2. verify( )

[successful]

ALT

3. return verified

4. login successful

5. request donation( )

6. choose blood group(bloodgroup)

7. save choice( )

8. return choice saved

9. return blood group chosen

10. Provide other information( )

11. Save information( )

12. return information saved

13. return information provided

14. return request created

3. return not verified

[unsuccessful]

4. login denied

**Sequence Diagram 2: ‘Find Volunteer’**

**:Admin**

**:Portal**

**:Database**

1. receive donation request

2. search eligible volunteer( )

[available]

ALT

3. return found

4. return forward information

[not available]

3. return not found

4. return not available

**Activity Diagram 1: ‘Create Donation Request’**

Enter the website

Enter Email & Password

Verify Login

[not verified]

[verified]

Enter in the dashboard

Create Request

Choose Blood Group

AB-

AB+

O-

O+

B-

B+

A-

A+

Selected Blood Group

Provide Other Information

Post Request

Logout

**Activity Diagram 2: ‘Find Volunteer’**

Enter Portal Dashboard

Check Requests

Check Donation Requirements

Search Volunteer

Respond to the request

[not found]

[found]

Forward Volunteer Information

Send Notice Not Found

**Statechart Diagram 1: ‘Donation’**

On Donation Request

entry:setDR(DR\_ID)

Checking Pending/ setAC(True)

Received/ setAC(bg)

Not Found/ setDelete(True)

Not Checked/ setAC(False),setOP(True)

On Process

On Admin Check

Found/setRC(True)

Deleted

Request Deleted/ setArchive(True)

Return/ setAC(True)

Request Confirmed

Donation On Process/ setDC(True)

(Archived)

Not Donated

Not Done/ setDonated(False)

On Donation Confirmation

Done/ setDonated(True)

Donated

Do: UpdateDatabase (DRN, bg,user\_name, volunteer\_name, volunteer\_contactnumber )

Do: ReduceDR(True)

**Statechart Diagram 2: ‘Volunteer’**

General

Donated first time ≤ 5 times/ setActive(true)

Active

Donation miss 2 times/ setExecutive(false)

Donated<5 times/

setExecutive(true)

Donate 1 time/

setInactive(false)

Executive

Do: receiveaward()

Donation miss 2 times/ setActive(false)

Inactive

Donation miss 1 time/ setExpelled(true)

(Expelled)

**LCOM Value Calculation- Class 1: ‘User’**

|  |
| --- |
| Class USER |
| user\_email  user\_password  request ID  bloodgroup  name  age  sex  contact number  vol\_name  vol\_contactnumber |
| Login(user\_email, user\_password)  CreateRequestDonation(requestID, bloodgroup, name, age, sex, contact number, user\_email)  CheckPendingRequest( requestID, bloodgroup)  DeleteRequest(requestID, user\_email, user\_password)  ShowVolunteerInfo(requestID, bloodgroup, vol\_name, vol\_contactnumber )  Logout( ) |

Let,

Login(user\_email, user\_password)= O1

CreateRequestDonation(requestID, bloodgroup, name, age, sex, contact number, user\_email)= O2

CheckPendingRequest( requestID, bloodgroup)= O3

DeleteRequest(requestID, user\_email, user\_password)= O4

ShowVolunteerInfo(requestID, bloodgroup, vol\_name, vol\_contactnumber )=O5

Logout( )= O6

*Pairs:*

(O1, O2), (O1, O3), (O1, O4), (O1, O5), (O1, O6), (O2, O3), (O2, O4), (O2, O5), (O2, O6), (O3, O4), (O3, O5), (O3, O6), (O4, O5), (O4, O6), (O5, O6)

P = 7 (Non-Cohesive pairs)

Q = 8 (Cohesive pairs)

We know that, LCOM = |P|-|Q|, if |P| > |Q|, otherwise 0

Here, Q > P

So, LCOM = 0

*Comment:* The LCOM value of the class indicates that the methods of the class are cohesive, and it is a desirable design.

**LCOM Value Calculation- Class 2: ‘Admin’**

|  |
| --- |
| Class ADMIN |
| user\_email  admin\_email  admin\_password  request ID  bloodgroup  vol\_name  vol\_contactnumber |
| Login(admin\_email, admin\_password)  CheckRequests(requestID, user\_email)  CheckRequirements(requestID, bloodgroup)  SearchVolunteers(admin\_email, admin\_password, requestID, vol\_name, vol\_contactnumber, bloodgroup)  Logout( ) |

Let,

Login(admin\_email, admin\_password)= O1

CheckRequests(requestID, user\_email)= O2

CheckRequirements(requestID, bloodgroup)= O3

SearchVolunteers(admin\_email, admin\_password, requestID, vol\_name, vol\_contactnumber, bloodgroup)= O4

Logout( )= O5

*Pairs:*

(O1, O2), (O1, O3), (O1, O4), (O1, O5), (O2, O3), (O2, O4), (O2, O5), (O3, O4), (O3, O5), (O4, O5)

We know that, LCOM = |P|-|Q|, if |P| > |Q|, otherwise 0

P = 6 (Non-Cohesive pairs)

Q = 4 (Cohesive pairs)

So, LCOM = |P|-|Q| =|6|-|4|= 2

*Comment:* The LCOM value of the class indicates that the methods of the class are non-cohesive. But as the LCOM value is while quite low, it might be considered as a desirable design.

-----------*The End*----------