AMERICAN INTERNATIONAL UNIVERSITY–BANGLADESH (AIUB)

Dept. of Computer Science
Faculty of Science and Technology

CSC2210: OBJECT ORIENTED PROGRAMMING 2

Spring 2024-2025

Section: M

Group No: 06

Project Report On

Project Name: Disaster Relief Management System

Supervised By

Zahiduddin Ahmed

Submitted By:

Name	ID
1. Tauhidur Rahman	24-57854-2
2. Wasisur Rahman	24-57844-2
3. Nuzhat Imam	24-58815-2
4.	

CO2: Display and verify the mean of a real-life Project using the concepts of C# Graphical User Interface based environment with database integration to depict a desktop-based application.

Assessment Criteria	Not Attended/ Incorrect (0)	Inadequate (1-2)	Average (3)	Good (4)	Excellent (5)
Evaluation Criteria	**Evaluation Definition**				**Total =**
Requirement fulfillment	Properly demonstrate a real-life scenario-based project with proper functional requirement identification for the Object-Oriented Programming project development activities.				
Validation	Ensuring the ability of students' proper demonstration on validation forms in their system in terms of dealing with the data.				
Verification	Identifying if the students can verify the system data along with proper functional requirements in terms of data flow.				

Table of Contents:

Chapter 1: Introduction

The **Disaster Relief Management System** is a role-based application designed to facilitate efficient coordination between victims affected by disasters, individuals willing to donate, volunteers handling deliveries, administrators managing the overall process, and the manager responsible for controlling administrative access. The system provides a centralized platform where victims can submit disaster requests, donors can contribute aid, and volunteers can deliver assistance while ensuring transparency and proper tracking of requests, donations, and deliveries. Administrators monitor system activities and user accounts, while the manager manages the creation and activation of admin users. The platform aims to improve disaster relief efficiency through structured workflows, centralized data management, and role-based access control.

Chapter 2: User stories

Overall System Story
The Disaster Relief Management System acts as a bridge between victims affected by disasters and individuals or organizations willing to help. Users first register and log in to the system. Based on their role, they can perform specific actions such as requesting aid, donating resources, volunteering for deliveries, or managing system data.

Victim User Story
A victim who has been affected by a disaster registers and logs into the system. After logging in, the victim creates a disaster request by providing details such as the disaster title, type, and description. The victim can view his submitted requests and track their progress (donations and deliveries). Once donations and deliveries are completed, the victim receives the assistance needed.

Donator User Story
A donator registers and logs into the system to help victims. After logging in, the donator views available disaster requests and selects one to support. The donator submits a donation by specifying the donation type, item details, and preferred delivery method. The donator can track donation and delivery status through the system and view their donation history.

Volunteer User Story
A volunteer logs into the system and views available delivery requests. Instead of being assigned by an admin, volunteers independently choose and accept delivery requests. After accepting a request, the volunteer delivers the donation to the specified location and updates the delivery status in the system.

Admin User Story
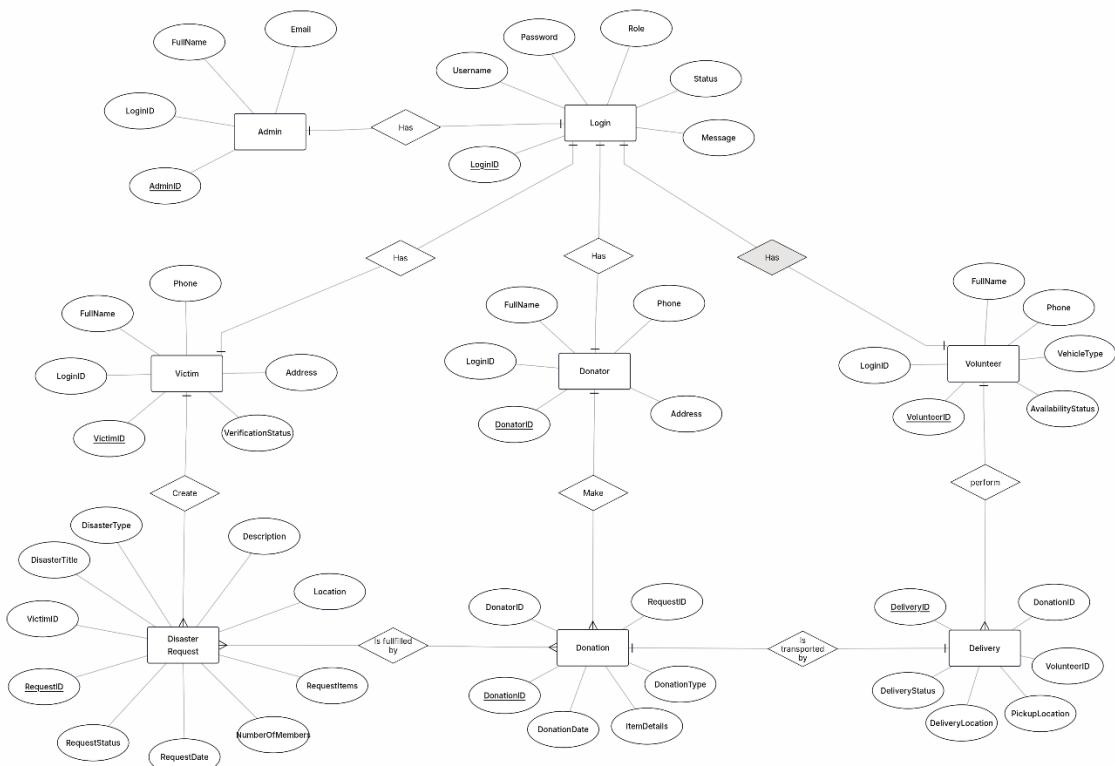
An admin logs into the system and accesses the dashboard, which displays system statistics such as total users, active users, inactive users, pending requests and the user distribution percentage. The admin reviews, approves, edits or deletes disaster requests and manages user accounts by activating or deactivating users by giving proper reason as message when necessary.

Manager User Story
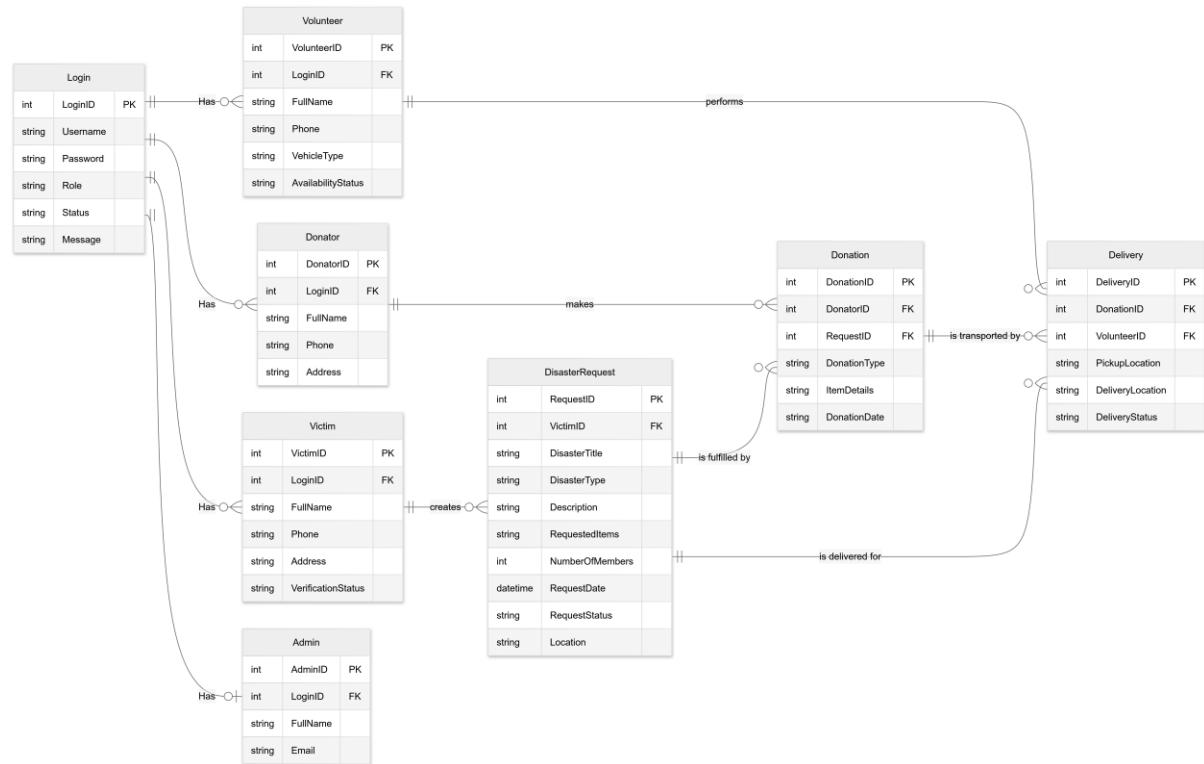
The manager logs into the system and oversees administrative accounts. The manager can view the list of admins, create new admin accounts, and activate or deactivate existing admins. This role ensures that administrative access remains controlled and secure.

Chapter 3: Database:

ER- Diagram

Data dictionary

Login

int	LoginID	PK
string	Username	
string	Password	
string	Role	
string	Status	
string	Message	

Volunteer

int	VolunteerID	PK
int	LoginID	FK
string	FullName	
string	Phone	
string	VehicleType	
string	AvailabilityStatus	

Donator

int	DonatorID	PK
int	LoginID	FK
string	FullName	
string	Phone	
string	Address	

Victim

int	VictimID	PK
int	LoginID	FK
string	FullName	
string	Phone	
string	Address	
string	VerificationStatus	

Admin

int	AdminID	PK
int	LoginID	FK
string	FullName	
string	Email	

DisasterRequest

int	RequestID	PK
int	VictimID	FK
string	DisasterTitle	
string	DisasterType	
string	Description	
string	RequestedItems	
int	NumberOfMembers	
datetime	RequestDate	
string	RequestStatus	
string	Location	

Donation

int	DonationID	PK
int	DonatorID	FK
int	RequestID	FK
string	DonationType	
string	ItemDetails	
string	DonationDate	

Delivery

int	DeliveryID	PK
int	DonationID	FK
int	VolunteerID	FK
string	PickupLocation	
string	DeliveryLocation	
string	DeliveryStatus	

SQL queries

1. Retrieve all approved disaster requests sorted by request date

   SELECT *
   FROM DisasterRequest
   WHERE RequestStatus = 'Approved'
   ORDER BY RequestDate DESC;

2. Validate user login credentials

   SELECT * FROM Login
   WHERE Username = @Username
   AND Password = @Password;

3. Retrieve admin details during login

   SELECT a.*, l.Username, l.Status, l.Message
   FROM Admin a
   INNER JOIN Login l ON a.LoginID = l.LoginID
   WHERE a.LoginID = @LoginID;

4. Retrieve donator details during login

```
SELECT d.*, l.Username, l.Status, l.Message
FROM Donator d
INNER JOIN Login l ON d.LoginID = l.LoginID
WHERE d.LoginID = @LoginID;
```

5. Retrieve victim details during login
6.

```
SELECT v.*, l.Username, l.Status, l.Message
FROM Victim v
INNER JOIN Login l ON v.LoginID = l.LoginID
WHERE v.LoginID = @LoginID;
```

7. Retrieve volunteer details during login

```
SELECT v.*, l.Username, l.Status, l.Message
FROM Volunteer v
INNER JOIN Login l ON v.LoginID = l.LoginID
WHERE v.LoginID = @LoginID;
```

8. Retrieve donation history for a specific donator

```
SELECT d.DonationID, r.DisasterTitle, v.FullName AS VictimName,
d.DonationDate, dl.DeliveryStatus
FROM Donation d
INNER JOIN DisasterRequest r ON d.RequestID = r.RequestID
INNER JOIN Victim v ON r.VictimID = v.VictimID
LEFT JOIN Delivery dl ON d.DonationID = dl.DonationID
WHERE d.DonatorID = @DonatorID;
```

9. Retrieve dashboard statistics for admin

```
SELECT
(SELECT COUNT() FROM Login) AS TotalUsers,
(SELECT COUNT() FROM Login WHERE Status = 1 AND Role NOT IN
('Admin','Manager')) AS ActiveUsers,
(SELECT COUNT() FROM Login WHERE Status = 0 AND Role NOT IN
('Admin','Manager')) AS InactiveUsers,
(SELECT COUNT() FROM DisasterRequest) AS TotalRequests,
(SELECT COUNT() FROM DisasterRequest WHERE RequestStatus =
'Pending') AS PendingRequests,
(SELECT COUNT() FROM Login WHERE Role = 'Volunteer') AS Volunteers,
(SELECT COUNT() FROM Login WHERE Role = 'Donator') AS Donors,
(SELECT COUNT() FROM Login WHERE Role = 'Victim') AS Victims,
```

(SELECT COUNT(*) *FROM Login WHERE Role = 'Admin') AS TotalAdmins,*
(SELECT COUNT() FROM Login WHERE Role = 'Admin' AND Status = 1)
AS ActiveAdmins,
(SELECT COUNT(*) FROM Login WHERE Role = 'Admin' AND Status = 0)
AS InactiveAdmins;

10. Retrieve detailed delivery information using delivery ID

```
SELECT dr.DisasterTitle, d.DonationType, d.ItemDetails, d.DonationDate,
del.DeliveryStatus,
lv.Username AS VictimUsername, v.Phone AS VictimPhone,
ld.Username AS DonatorUsername, dn.Phone AS DonatorPhone
FROM Delivery del
INNER JOIN Donation d ON del.DonationID = d.DonationID
INNER JOIN DisasterRequest dr ON d.RequestID = dr.RequestID
INNER JOIN Victim v ON dr.VictimID = v.VictimID
INNER JOIN Login lv ON v.LoginID = lv.LoginID
INNER JOIN Donator dn ON d.DonatorID = dn.DonatorID
INNER JOIN Login ld ON dn.LoginID = ld.LoginID
WHERE del.DeliveryID = @DeliveryID;
```

11. Retrieve user profile details by login ID

```
SELECT l.Username, l.Status, l.Role,
v.FullName AS VictimFullName, v.Phone AS VictimPhone, v.Address AS
VictimAddress,
d.FullName AS DonatorFullName, d.Phone AS DonatorPhone, d.Address AS
DonatorAddress,
vol.FullName AS VolunteerFullName, vol.Phone AS VolunteerPhone,
vol.VehicleType
FROM Login l
LEFT JOIN Victim v ON l.LoginID = v.LoginID
LEFT JOIN Donator d ON l.LoginID = d.LoginID
LEFT JOIN Volunteer vol ON l.LoginID = vol.LoginID
WHERE l.LoginID = @LoginID;
```

12. Retrieve a specific disaster request with victim details

```
SELECT dr.DisasterTitle, dr.DisasterType, dr.Description, dr.RequestedItems,
dr.NumberOfMembers, dr.RequestDate, dr.Location, dr.RequestStatus,
v.FullName AS VictimFullName
FROM DisasterRequest dr
INNER JOIN Victim v ON dr.VictimID = v.VictimID
WHERE dr.RequestID = @RequestID;
```

13. Insert a new login record
    INSERT INTO Login (Username, Password, Status, Role, Message)
    VALUES (@Username, @Password, @Status, @Role, @Message);

14. Insert a new admin record

    INSERT INTO Admin (LoginID, FullName, Email)
    VALUES (@LoginID, @FullName, @Email);

15. Insert a new victim record

    INSERT INTO Victim (LoginID, FullName, Phone, Address,
    VerificationStatus)
    VALUES (@LoginID, @FullName, @Phone, @Address,
    @VerificationStatus);

16. Insert a new donator record

    INSERT INTO Donator (LoginID, FullName, Phone, Address)
    VALUES (@LoginID, @FullName, @Phone, @Address);

17. Insert a new volunteer record

    INSERT INTO Volunteer (LoginID, FullName, Phone, VehicleType,
    AvailabilityStatus)
    VALUES (@LoginID, @FullName, @Phone, @VehicleType,
    @AvailabilityStatus);

18. Insert a new disaster request

    INSERT INTO DisasterRequest (VictimID, DisasterTitle, DisasterType,
    Description, RequestedItems, NumberOfMembers, Location)
    VALUES (@VictimID, @DisasterTitle, @DisasterType, @Description,
    @RequestedItems, @NumberOfMembers, @Location);

19. Insert a new donation record

    INSERT INTO Donation (DonatorID, RequestID, DonationType, ItemDetails)
    VALUES (@DonatorID, @RequestID, @DonationType, @ItemDetails);

20. Insert a new delivery record

    INSERT INTO Delivery (DonationID, PickupLocation, DeliveryLocation,
    DeliveryStatus)
    VALUES (@DonationID, @PickupLocation, @DeliveryLocation,

@DeliveryStatus);

21. Delete delivery records related to a disaster request

    DELETE FROM Delivery
    WHERE DonationID IN (
    SELECT DonationID FROM Donation WHERE RequestID = @RequestID
    );

22. Delete donation records related to a disaster request

    DELETE FROM Donation
    WHERE RequestID = @RequestID;

23. Delete a disaster request

    DELETE FROM DisasterRequest
    WHERE RequestID = @RequestID;

24. Update disaster request details

    UPDATE DisasterRequest
    SET DisasterTitle = @DisasterTitle,
    DisasterType = @DisasterType,
    Description = @Description,
    RequestedItems = @RequestedItems,
    NumberOfMembers = @NumberOfMembers,
    Location = @Location,
    RequestStatus = @RequestStatus
    WHERE RequestID = @RequestID;

25. Update login status and message

    UPDATE Login
    SET Status = @Status, Message = @Message
    WHERE LoginID = @LoginID;

26. Retrieve pending delivery requests for volunteers

    SELECT del.*, dn.DonationDate AS RequestDate
    FROM Delivery del
    INNER JOIN Donation dn ON del.DonationID = dn.DonationID
    WHERE DeliveryStatus = 'Pending';

27. Assign volunteer to a delivery
    UPDATE Delivery
    SET DeliveryStatus = 'Assigned', VolunteerID = @VolunteerID
    WHERE DeliveryID = @DeliveryID;
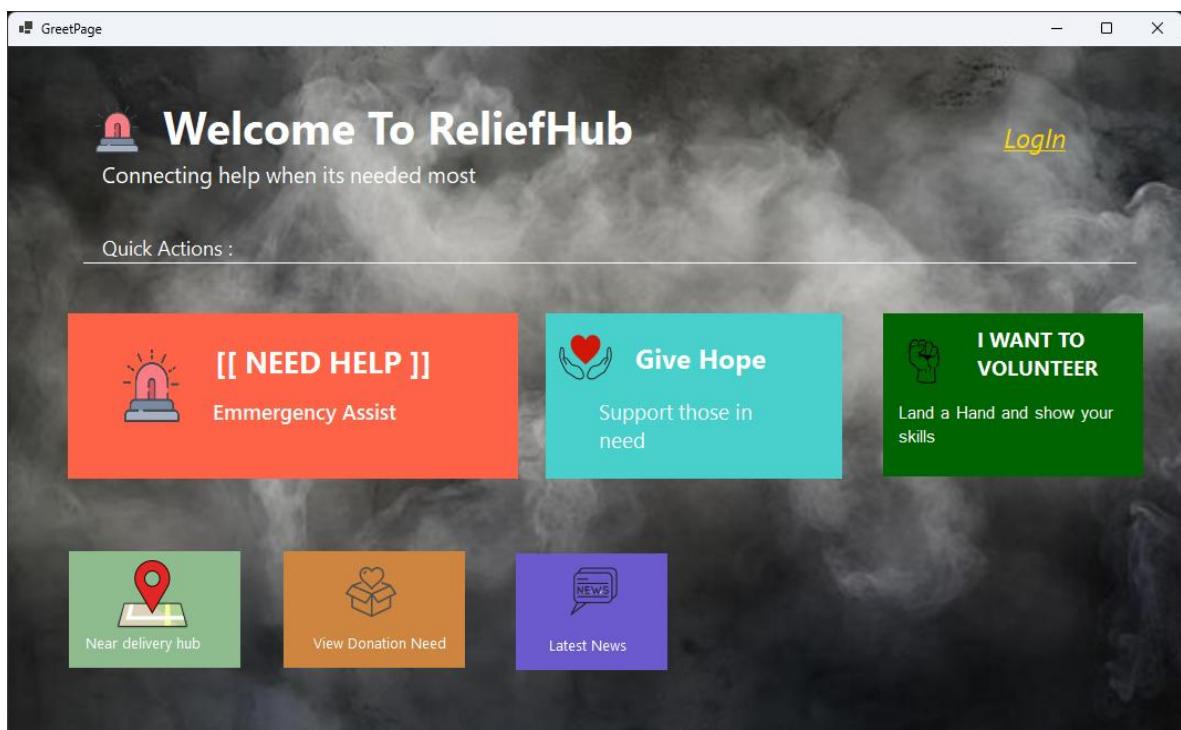
28. Update delivery status
    UPDATE Delivery
    SET DeliveryStatus = @DeliveryStatus
    WHERE DeliveryID = @DeliveryID;

Chapter 4: Screenshot of forms with figure names**:**

Form1: Start Page

Form2: Victim SignUp

Form3: Donator SignUP

Form4: Volunteer SignUP

Form5: Login Page

Form6: Manager Dashboard

Form7: Create Admin (Manager)

ReliefHub   Welcome, Manager | LogOut

Home
Create New Admin
Admin List

Search by name or username...   All

asdf   Inactive [Activate]
eee • Admin

fdsfa   Active   [Deactivate]
ad1 • Admin

Form8: Admin List (Manager)

ReliefHub   Welcome, fdsfa | LogOut

Home
Manage Users
Manage Requests
Pending Requests

Dash Board

| PENDING | ACTIVE USER | TOTAL REQ: | INACTIVE |
| 1 | 7 | 3 | 1 |

PENDING REQUESTS: 1   USER DRISTRIBUTION

boner maje ami | sundarban | Needs: afsdjkf

Donor
(37%)

Victim
(37%)

Volunteer
(25%)

Form9: Admin Dashboard

Form10: Manage Users (Admin)

Form11: Manage Requests (Admin)

Form12: Pending Requests (Admin)

Form13: Victim Home

Form14: User's Requests (Victim)

Form15: Create Request (Victim)

Form16: History of Donation Received (Victim himself only)

Form17: Donator Home

Form18: Donation Histiory (Donator)

Form19: Volunteer Home

Form20: Currently Ongoing Deliveries (Volunteer)

Form21: Delivery History (Volunteer)