



Database Management

METLIFE 2.0

Group 24 – Team Blade Warriors

Submitted By

Student ID	Name
Mohaiminul Islam	2030815
Zaber Al Mamun	2022676
Raima Haque Prema	2031362
Ridwan Mahmud Jihan	2020536
Mehenaz Faria Rahman	2022540
Rabeya Noor Tarin	2030170

Introduction

Requirement Analysis

Conclusion

CHAPTER-1 INTRODUCTION

A. BACKGROUND OF THE ORGANIZATION - IUB:

Established in 1993, Independent University, Bangladesh (IUB) is the leading private university in Bangladesh with an explicit focus on research and global partnerships. With a current enrollment of around 10,000 at undergraduate and graduate levels, more than 450 highly qualified faculty members, 47% of whom have PhD degrees, and an alumni strength of close to 14,000, IUB is trying to ensure education all to promote the national agenda of sustainable inclusive development.[1]

The academic curriculum is based on the North American Liberal Arts Model and the medium of instruction is English. There are currently five academic schools: [1]

- 1) School of Business & Entrepreneurship
- 2) School of Engineering, Technology and Sciences
- 3) School of Environment & Life Sciences
- 4) School of Liberal Arts & Social Sciences
- 5) School of Pharmacy and Public Health

B. BACKGROUND OF THE PROJECT – METLIFE 2.0:

MetLife is a multinational insurance and financial services corporation providing a diverse range of products and solutions. Specializing in life, dental, and disability insurance, MetLife also offers retirement planning and investment services. The company serves individuals and businesses, operating globally to meet the diverse financial needs of its extensive customer base.

C. OBJECTIVE OF THE PROJECT - SPMS 4.0:

The METLIFE 2.0 database contains all the information needed to evaluate the performance of the users. Users can therefore statistically evaluate their own success.

D. SCOPE OF THE PROJECT:

We thoroughly analyzed the current system (METLIFE) and found some problems in the processes that could make the process ineffective, and lead to communication breakdowns. The suggested remedy for those problems is to develop a better version of the system called METLIFE 2.0. We want to create user interfaces so that every user can access the data they need and use the system to create, examine, and download the reports and documents they want. Additionally, we want to create a platform that will enable faculty members to work together on creating course outlines.

CHAPTER-2 REQUIREMENT ANALYSIS

Requirement Analysis is the process of determining what the database is used for. It involves interviews with users to identify the functionality and system requirements they expect and require from the database, what operations need to be performed and what data they need to process. By doing so, we can get a proper understanding of the stakeholders and how they interact with each other.

A. RICH PICTURE – EXISTING SYSTEM (SPMS 3.0):

A rich picture is a way to explore, acknowledge and define a situation and express it through diagrams to create a preliminary mental model and can help to open discussion and come to a broad, shared understanding of a situation.[3] A rich picture enables us to recognize connections and interactions that we might otherwise overlook. It aids in determining one or more themes that participants might wish to investigate and handle further. Rich pictures are thus always employed during the pre-analysis stage.

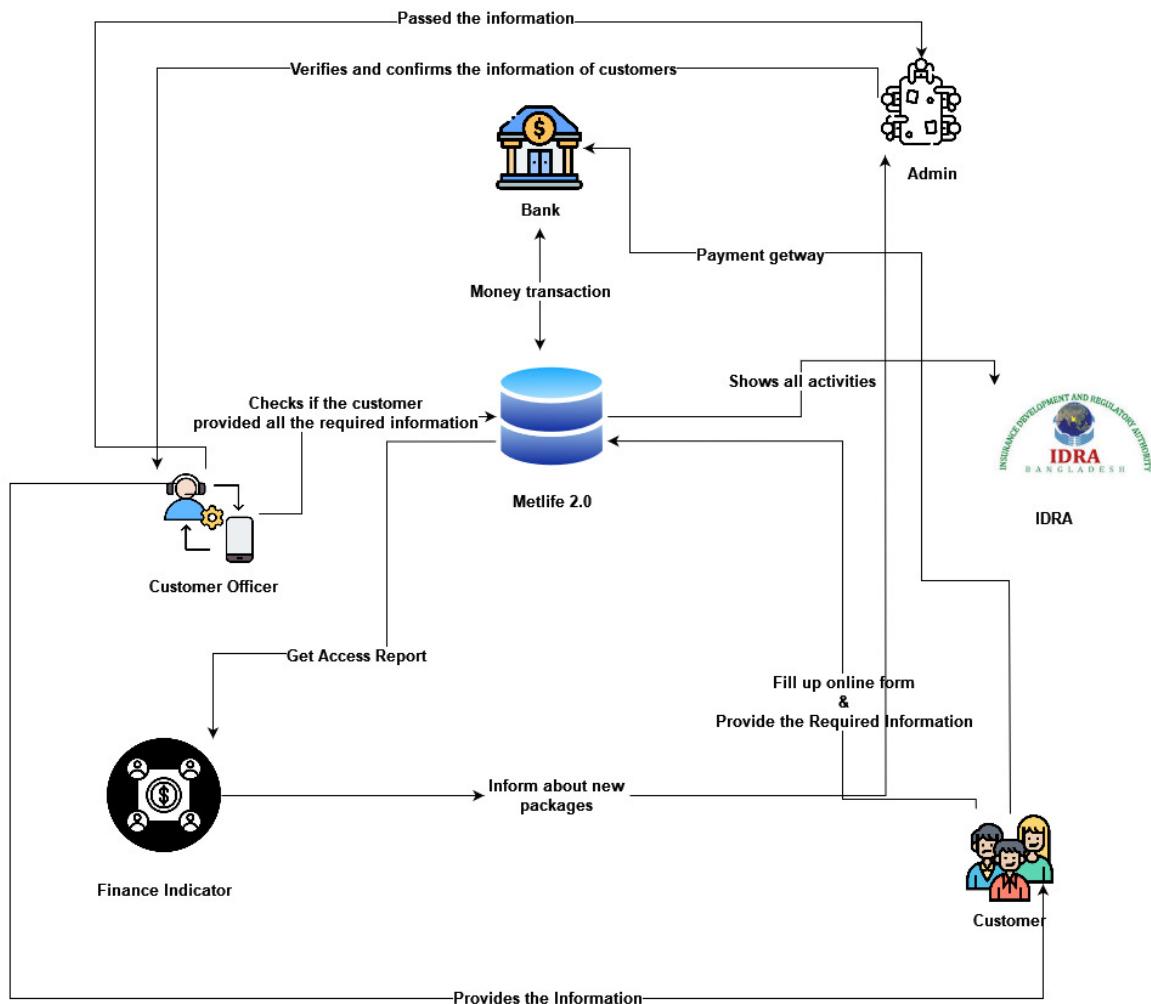


FIGURE 2.1: RICH PICTURE (AS IS)

D. RICH PICTURE – PROPOSED SYSTEM(METLIFE 2.0):

MetLife 2.0

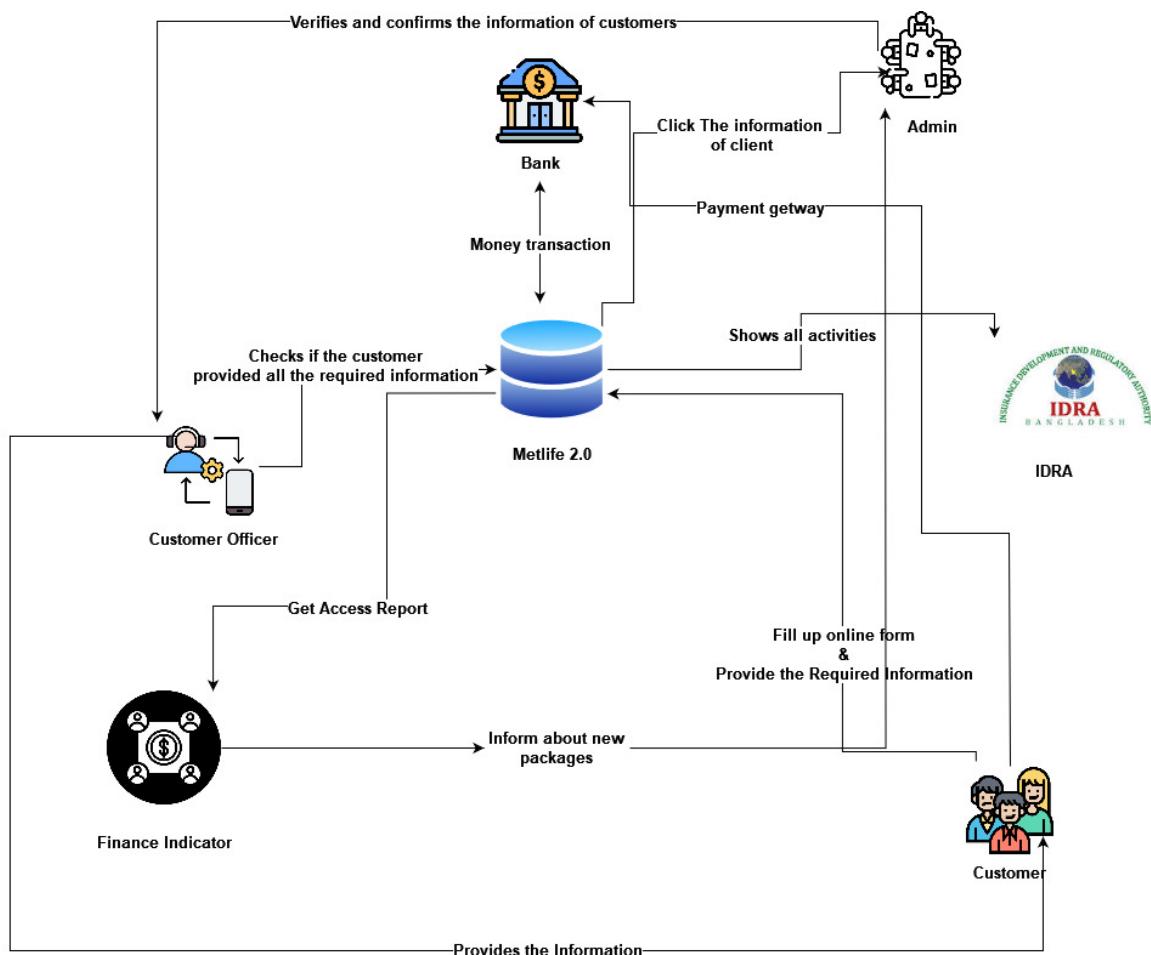


FIGURE 2.2: RICH PICTURE (TO BE)

A. ENTITY RELATIONSHIP DIAGRAM:

MetLife 2.0

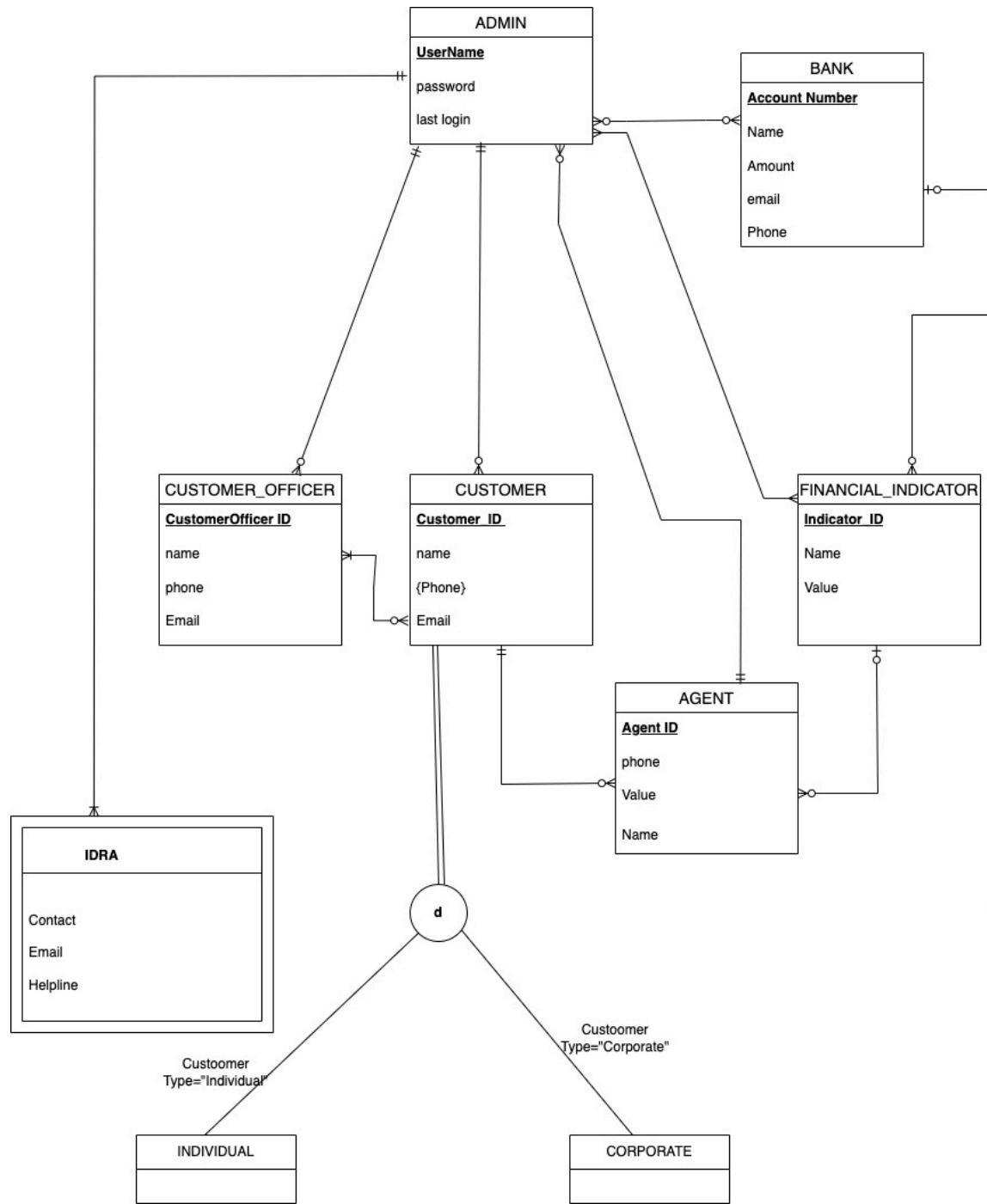


FIGURE 3.1: ENTITY RELATIONSHIP DIAGRAM

B. ENTITY RELATIONSHIP DIAGRAM TO RELATIONAL

SCHEMA:

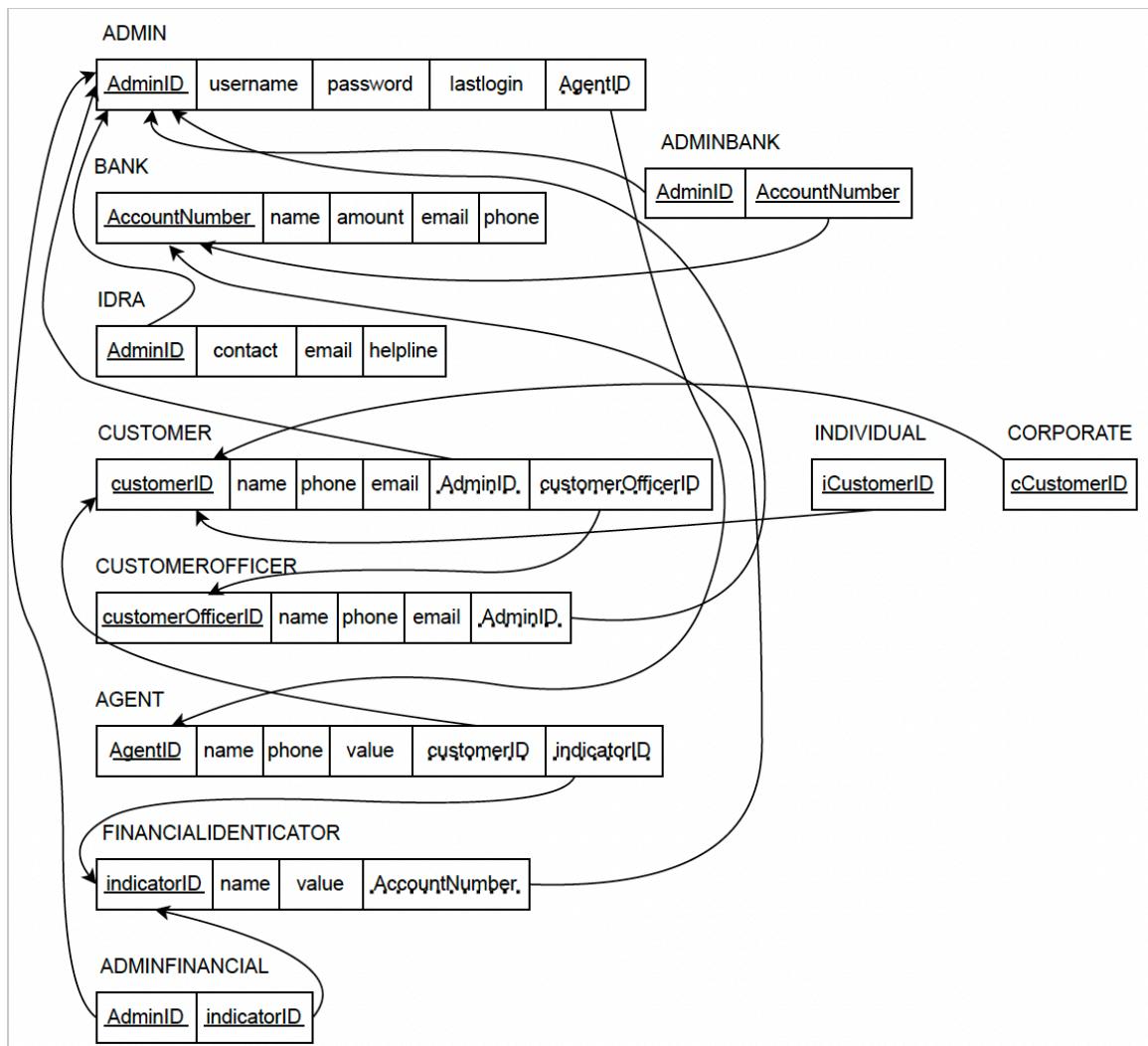


FIGURE 3.2: RELATIONAL SCHEMA

C. NORMALIZATION:

ADMIN	adminid	a1	Customer officer	Customer officerid	o1
	username	a2		name	o2
	password	a3		phone	o3
	lastlogin	a4		email	o4
	agentid	g1		adminid	a1
BANK	account number	b1	AGENT	agentid	g1
	name	b2		name	g2

MetLife 2.0

	amount	b3		phone	g3
	email	b4		value	g4
	phone	b5		customerid	c1
	customerid	c1		Indicator id	f1
IDRA	adminID	a1	Financial indicator	agentid	g1
	contact	i1		Indicator id	f1
	email	i2		name	f2
	helpline	i3		value	f3
CUSTOMER	customerid	c1		Account number	f4
	name	c2	ADMIN FINANCIAL	adminid	a1
	phone	c3		indicatorid	f1
	email	c4	INDIVIDUAL	icustomerid	x1
	Admin id	a1		customerid	c1
	Customer officer id	o1	CORPORATE	ccustomerid	y1
	Account number	b1		customerid	c1

a1	a2, a3,a4,g1
b1	b2, b3,b4.b5,c1
a1	i1,i2,i3
c1	c2,c3,c4,a1,o1,b1
o1	o2,o3,o4,a1
g1	g2, g3,g4,c1,f1
f1	f2,f3,f4,g1
a1	f1
x1	c1
y1	c1

1NF:

R:

a1	c1	a2	a3	a4	b1	b2	b3	b4	b5	i1	i2	i3	c2	c3	c4	o1	o2	o3	o4
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

g1	g2	g3	g4	f1	f2	f3	f4	X1	Y1
----	----	----	----	----	----	----	----	----	----

2NF:

R1:

a1	a2	a3	a4	g1
----	----	----	----	----

R2:

c1	c2	c3	c4	a1	o1	b1
----	----	----	----	----	----	----

R3:

c1	x1	y1
----	----	----

R4:

o1	o2	o3	o4	a1
----	----	----	----	----

R5:

b1	b2	b3	b4	b5	c1
----	----	----	----	----	----

R6:

g1	g2	g3	g4	c1	f1
----	----	----	----	----	----

R7:

f1	f2	f3	f4	g1
----	----	----	----	----

3NF:

Has no transitive dependencies.

2NF:

1)Partial dependency has been removed

3NF:

1)Has no transitive dependencies

E.DATA DICTIONARY

ADMIN_PANEL T

Name	Data Type	Size	Remarks
AdminID	INTEGER	11	This is the primary key for the admin Table. E.g. : 0123456
UserNameFirst	VARCHAR	50	This is the name of the admin. E.g : Zaber
UserNameLast	VARCHAR	50	This is the name of the admin. E.g : Mamun
Password	VARCHAR	30	This is the mainly use of security. E.g: Z@ber00007
LastLogin	VARCHAR	40	Basically, username and password are used here.

CUSTOMER_OFFICER T

MetLife 2.0

Name	Data Type	Size	Remarks
CustomerOfficerID	INTEGER	11	This is the primary key for the Customer officer Table. E.g. : 0123456
CustomerOfficerName	VARCHAR	50	This is the name of the Customer officer. E.g : Zaber
COPhone	INTEGER	11	This is the mobile number of the nominee.E.g.:12345678910
COmail	VARCHAR	25	This is the email address of the Customer Officer.E.G-Zaber00007@gmail.com

CUSTOMER T

Name	Data Type	Size	Remarks
CustomerID	INTEGER	11	This is the primary key for the Customer Table. E.g. : 0123456
CustomerName	VARCHAR	50	This is the name of the Customer. E.g : Zaber
CPhone	INTEGER	11	This is the mobile number of the customer.E.g.:12345678910
Cmail	VARCHAR	25	This is the email address of the Customer. E.G-Zaber00007@gmail.com

BANK T

MetLife 2.0

Name	Data Type	Size	Remarks
AccountNumber	INTEGER	11	This is the primary key for the Bank Table. E.g. : 0123456
BName	VARCHAR	50	This is the name of the Customer officer. E.g : brac, zaber bank
amount	INTEGER	11	This is the amount.E.g.:12345678910
BPhone	INTEGER	11	This is the mobile number of the BANK.E.g.:12345678910

IDRA T

Name	Data Type	Size	Remarks
ContactNumber	INTEGER	11	This is the Contact Number of the IDRA.E.g.:12345678910
Cmail	VARCHAR	50	This is the email address of the IDRA. E.G-Zaber00007@gmail.com
HelpLINE	INTEGER	50	This is the Contact Number of the IDRA. E.g : Mamun

FINANCIAL IDICATOR T

Name	Data Type	Size	Remarks
IndicatorID	INTEGER	20	This is the primary key for the FINANCIAL IDICATOR Table. E.g. : 0123456
FName	VARCHAR	50	This is the name of the FINANCIAL IDICATOR Table. E.g : Zaber
FIValue	INTEGER	50	This is the value amount of FINANCIAL IDICATOR Table. E.g : Mamun

AGENT T

MetLife 2.0

Name	Data Type	Size	Remarks
AgentID	INTEGER	11	This is the primary key for the agent Table. E.g. : 0123456
AName	VARCHAR	50	This is the name of the agent Table. E.g : Zaber
APhone	INTEGER	11	This is the mobile number of the agent Table.E.g.:12345678910
Value	INTEGER	25	This is the value amount of agent Table. E.g : Mamun

```

MEDICAL_INSURANCE
> assets
# admin.css
bronze.jpg
config.php
cover.jpg
cover2.jpg
gold.jpg
index.html
insurance.sql
login.php
showOrder.php
# showorderstyle.css
silver.jpg
tailwind.config.js

login.php
1 <?php
2 include 'config.php';
3 ?>
4
5 <!doctype html>
6 <html lang="en">
7   <head>
8     <!-- Required meta tags -->
9     <meta charset="utf-8">
10    <meta name="viewport" content="width=device-width, initial-scale=1">
11    <!-- Bootstrap CSS -->
12    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">
13    <!-- tailwind -->
14    <script src="https://cdn.tailwindcss.com"></script>
15    <link rel="shortcut icon" href=".//assets/health-insurance.png" type="image/x-icon">
16    <link rel="stylesheet" href="admin.css">
17    <title>MetLife 2.0 - Login</title>
18  </head>
19  <body>
20    <div class="container-contact100">
21      <div class="wrap-contact100">
22        <form class="contact100-form validate-form" method="post">
23          <p class="text-center text-4xl bg-sky-500 rounded-xl p-3 m-5 text-white font-fatface">
24            MetLife 2.0
25          </p>
26          <p class="text-2xl text-center pb-4 font-bold font-fatface">
27            Your perfect destination for right medical insurance

```

In 76 Col 5 Spaces: 4 JTF-R

MetLife 2.0

MetLife 2.0

The image shows two side-by-side code editors, likely from a development environment like VS Code or similar, displaying PHP code for the MetLife 2.0 application.

Top Editor (login.php):

```
1 <?php
2 include 'config.php';
3 ?>
4
5 <!doctype html>
6 <html lang="en">
7   <head>
8     <!-- Required meta tags -->
9     <meta charset="utf-8">
10    <meta name="viewport" content="width=device-width, initial-scale=1">
11    <!-- Bootstrap CSS -->
12    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDla" la
13  <!-- tailwind -->
14  <script src="https://cdn.jsdelivr.net/npm/tailwindcss@2.0.2/dist/tailwind.min.js" rel="stylesheet" integrity="sha384-+PjzXfJZ" la
15  <link rel="shortcut icon" href="/assets/health-insurance.png" type="image/x-icon">
16  <link rel="stylesheet" href="admin.css">
17  <title>Metlife 2.0 - Login</title>
18 </head>
19 <body>
20   <div class="container-contact100">
21     <div class="wrap-contact100">
22       <form class="contact100-form validate-form" method="post">
23         <p class="text-center text-4xl bg-sky-500 rounded-xl p-3 m-5 text-white font-serif contact100-form-title">
24           Metlife 2.0
25         </p>
26         <p class="text-2xl text-center pb-4 font-bold font-fatface ">
27           Your perfect destination for right medical insurance
28         </p>
29         <span class="contact100-form-title">
30           User Login
31         </span>
32
33         <div class="wrap-input100 validate-input" data-validate="Please enter email">
34           <input class="input100" type="text" name="email" placeholder="Email" >
35           <span class="focus-input100"></span>
36         </div>
37
38         <div class="wrap-input100 validate-input" data-validate="Please enter password">
39           <input class="input100" type="password" name="password" placeholder="Password" >
40           <span class="focus-input100"></span>
41         </div>
```

Bottom Editor (showOrder.php):

```
74   <thead>
75     <tr>
76       <th>Name</th>
77       <!-- <th>Number</th> -->
78       <th>Email</th>
79       <th>Package</th>
80       <th>Address</th>
81       <th>Total Price</th>
82     </tr>
83   </thead>
84   <tbody>
85     <php
86       while ($row = mysqli_fetch_assoc($result)) {
87         echo "<tr>";
88         echo "<td>" . htmlspecialchars($row['name']) . "</td>";
89         // echo "<td>" . htmlspecialchars($row['number']) . "</td>";
90         echo "<td>" . htmlspecialchars($row['email']) . "</td>";
91         echo "<td>" . htmlspecialchars($row['method']) . "</td>";
92         echo "<td>" . htmlspecialchars($row['address']) . "</td>";
93         echo "<td>" . htmlspecialchars($row['total_price']) . "</td>";
94         echo "</tr>";
95       }
96     </tbody>
97   </table>
98 </div>
99 </div>
100 <?php
101 mysqli_close($conn);
102 ?>
103
104 </body>
105 </html>
```

MetLife 2.0

The screenshot shows a code editor interface with two tabs open: `login.php` and `addProduct.php`.

login.php:

```
...  
login.php  
...  
64 if(isset($_POST['login'])){  
65 {  
66     $query = "SELECT * FROM `users` WHERE `email`='".$_POST[email]' AND `password`='".$_POST[password]';  
67     $result = mysqli_query($conn,$query);  
68     if(mysqli_num_rows($result)==1){  
69         session_start();  
70         $_SESSION['AdminName'] = $_POST['email'];  
71         header("location: ./index.html");  
72     }  
73     else{  
74         echo "<script> alert('Invalid Credentials'); </script>";  
75     }  
76 }  
77  
78 if(isset($_POST['adminlogin'])){  
79 {  
80     $query = "SELECT * FROM `admin` WHERE `username`='".$_POST[email]' AND `pass`='".$_POST[pass]';  
81     $result = mysqli_query($conn,$query);  
82     if(mysqli_num_rows($result)==1){  
83         session_start();  
84         $_SESSION['AdminName'] = $_POST['name'];  
85         header("location: ./showOrder.php");  
86     }  
87     else{  
88         echo "<script> alert('Invalid Credentials'); </script>";  
89     }  
90 }  
91 }
```

addProduct.php:

```
...  
addProduct.php  
...  
1 <?php  
2     include 'config.php';  
3 ?>  
4  
5 <!doctype html>  
6 <html lang="en">  
7     <head>  
8         <!-- Required meta tags -->  
9         <meta charset="utf-8">  
10        <meta name="viewport" content="width=device-width, initial-scale=1">  
11  
12        <!-- Bootstrap CSS -->  
13        <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QD+iT23nUHlJa7uL2O1FqL" la  
14  
15        <link rel="shortcut icon" href=".//assets/health-insurance.png" type="image/x-icon">  
16        <title>Add A Product</title>  
17  
18        <link href="https://cdn.jsdelivr.net/npm/daisyui@4.4.19/dist/full.min.css" rel="stylesheet" type="text/css" />  
19  
20    <script src="https://cdn.tailwindcss.com"></script>  
21  
22  
23  
24    </head>  
25    <body>  
26  
27    <?php  
28        include 'config.php';  
29  
30  
31    if ($_SERVER['REQUEST_METHOD'] === 'POST') {  
32        if (isset($_POST['product_name']) && isset($_POST['product_price']) && isset($_POST['product_description']) && isset($_FILES['img']) && $_FILE  
33            $productName = $_POST['product_name'];  
34            $price = $_POST['product_price'];  
35            $des = $_POST['product_description'];  
36  
37            // Handle image upload  
38            $imgData = file_get_contents($_FILES['img']['tmp_name']);  
39  
40            // Use prepared statement to insert image binary data into the database  
41            $statement = $conn->prepare("INSERT INTO products(name, image, price, product_details) VALUES(?, ?, ?, ?)");  
42            $statement->bind_param("s", $productName);  
43            $statement->bind_param("s", $imgData);  
44            $statement->bind_param("d", $price);  
45            $statement->bind_param("s", $des);  
46            $statement->execute();  
47  
48    }  
49  
50    header("location: ./index.html");  
51 }  
52  
53 }  
54  
55 }  
56  
57 }  
58  
59 }  
60  
61 }  
62  
63 }  
64  
65 }  
66  
67 }  
68  
69 }  
70  
71 }  
72  
73 }  
74  
75 }  
76  
77 }  
78  
79 }  
80  
81 }  
82  
83 }  
84  
85 }  
86  
87 }  
88  
89 }  
90  
91 }
```

MetLife 2.0

The image shows two side-by-side code editors, likely from a developer's IDE, displaying the source code for the MetLife 2.0 medical insurance database management system.

Top Editor (PHP):

```
$query = "SELECT * FROM `users` WHERE `email`='$_POST[email]' AND `password`='$_POST[password]'";
$result = mysqli_query($conn,$query);
if(mysqli_num_rows($result)==1){
    session_start();
    $_SESSION['AdminName'] = $_POST['email'];
    header("location: ./index.php");
}
else{
    echo "<script> alert('Invalid Credentials'); </script>";
}

if(isset($_POST['adminlogin'])){
    $query = "SELECT * FROM `admin` WHERE `username`='$email' AND `pass`='$password'";
    $result = mysqli_query($conn,$query);
    if(mysqli_num_rows($result)==1){
        session_start();
        $_SESSION['AdminName'] = $_POST['name'];
        header("location: ./showOrder.php");
    }
    else{
        echo "<script> alert('Invalid Credentials'); </script>";
    }
}
if(isset($_POST['registration'])){
    header("location: ./regform.php");
    exit();
}
?>
</body>
</html>
```

Bottom Editor (CSS/HTML):

```
<span class="focus-input100"></span>
</div>
<div class="flex gap-10">
    <div class="container-contact100-form-btn">
        <input class="bg-sky-500 contact100-form-btn rounded-xl" type="submit" name="login" value="login">
    </div>
    <div class="container-contact100-form-btn">
        <input class="bg-sky-500 contact100-form-btn rounded-xl" type="submit" name="adminlogin" value="Admin login">
    </div>
    <div class="container-contact100-form-btn">
        <input class="bg-sky-500 contact100-form-btn rounded-xl" type="submit" name="registration" value="Registration">
    </div>
</div>
</div>

<!-- Optional JavaScript; choose one of the two! -->
<!-- Option 1: Bootstrap Bundle with Popper -->
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-MrcW6ZMFYlzcLA8Nl+A7MsX" >
</script>
<?php
if(isset($_POST['login'])){
    $query = "SELECT * FROM `users` WHERE `email`='$_POST[email]' AND `password`='$_POST[password]'";
    $result = mysqli_query($conn,$query);
    if(mysqli_num_rows($result)==1){
        session_start();
        $_SESSION['AdminName'] = $_POST['email'];
        header("location: ./index.php");
    }
    else{
        echo "<script> alert('Invalid Credentials'); </script>";
    }
}
if(isset($_POST['adminlogin'])){
    $query = "SELECT * FROM `admin` WHERE `username`='$email' AND `pass`='$password'";
    $result = mysqli_query($conn,$query);
    if(mysqli_num_rows($result)==1){
        session_start();
        $_SESSION['AdminName'] = $_POST['name'];
        header("location: ./showOrder.php");
    }
    else{
        echo "<script> alert('Invalid Credentials'); </script>";
    }
}
```

MetLife 2.0

The image shows two side-by-side screenshots of the Microsoft Edge browser. Both windows are displaying the same PHP file, "regform.php", from a local file path: "metlife_2.0-medical-insurance-dbs-main". The file content is a PHP script for a registration form.

```
<?php  
if ($_SERVER["REQUEST_METHOD"] == "POST") {  
    if (isset($_POST["register"])) {  
        $email = $_POST["email"];  
        $password = $_POST["password"];  
        $name = $_POST["name"];  
  
        $sql = "INSERT INTO users(name,email, password) VALUES ('$name','$email','$password');";  
        $result = mysqli_query($conn, $sql);  
        header("location: ./login.php");  
        mysqli_close($conn);  
    } elseif (isset($_POST["back"])) {  
        header("location: ./login.php");  
        exit();  
    }  
}  
?  
81  
82  
83 </body>  
84 </html>
```

The code includes a registration logic where it inserts data into a "users" table and then redirects to "login.php". It also includes a "back" button logic that returns to "login.php". The code uses standard PHP syntax with MySQLi database interactions.

MetLife 2.0

The image shows a code editor interface with two tabs open, displaying PHP code for a medical insurance application named 'MetLife 2.0'.

Top Tab: showOrder.php

```
<div class="hidden md:hidden flex-row-reverse">
    <button class="btn btn-ghost normal-case text-black">
        <span class="icon">
            | <i class="fas fa-bars"></i>
        </span>
    </button>
</div>

<div class="hidden md:flex flex-col md:flex-row space-y-4 md:space-y-0 md:space-x-4 ">
    <a class="btn btn-ghost text-lg text-white font-medium href=".//chart.php">Order chart </a>
    <a class="btn btn-ghost text-lg text-white font-medium href=".//showOrder.php">Recent Orders</a>
    <a class="btn btn-ghost text-lg text-white font-medium href=".//addProduct.php">Add Product</a>
    <a class="btn btn-ghost text-lg text-white font-medium href=".//manageStuffs.php">Manage Stuffs</a>
    <a class="btn btn-ghost text-lg text-white font-medium href=".//login.php" > Logout </a>
</div>

<div class="flex justify-between flex-row-reverse">
    <a class="btn btn-ghost normal-case text-4xl text-black font-fatface font-bold">
        Metlife 2.0 - admin panel
    </a>
</div>
```

Bottom Tab: addProduct.php

```
// Handle image upload
$imgData = file_get_contents($_FILES['img']['tmp_name']);

// Use prepared statement to insert image binary data into the database
$stmt = $conn->prepare("INSERT INTO products(name, image, price, product_details) VALUES(?, ?, ?, ?)");
$stmt->bind_param("ssds", $productName, $imgData, $price, $details);

if ($stmt->execute()) {
    session_start();
    echo "<script> alert('Product Added'); </script>";
} else {
    // Handle insertion error
    echo "<script>alert('Product addition failed.');//</script>";
}

$stmt->close();
} else {
    // Handle missing or incomplete form data
    echo "<script>alert('Please fill in all required fields.');//</script>";
}
?>
```

MetLife 2.0

The image shows a developer's environment with a code editor at the top and two browser windows below it.

Code Editor: The code editor displays the file `showOrder.php` from the `METLIFE 2.0-MEDICAL-INSURANCE DBMS` project. The code is a PHP script that queries a MySQL database to retrieve order information. It includes meta tags for character encoding and viewport, a title, and a CSS style block for a table. It also links to a Tailwind CSS file.

```
<?php
include 'config.php';
session_start();
$statement = "SELECT * FROM orders";
$result = mysqli_query($conn, $statement);
?>
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Orders</title>
<style>
table {
border-collapse: collapse;
margin: 0 auto;
text-align: center;
}
h2{
font-size: 25px;
text-align: center;
}
td, th {
border: 1px solid white;
padding: 8px;
}
</style>
<link rel="shortcut icon" href=".//assets/health-insurance.png" type="image/x-icon">
<link href="https://cdn.jsdelivr.net/npm/daisysui@4.4.19/dist/full.min.css" rel="stylesheet" type="text/css" />
<script src="https://cdn.tailwindcss.com"></script>
</head>
<body>
<nav>
```

User Login: The first browser window shows the Metlife 2.0 login page. It features a blue header with the text "Metlife 2.0". Below it is a sub-header: "Your perfect destination for right medical insurance". The main form is titled "User Login" and contains fields for email ("raima@gmail.com") and password (""). There are three buttons at the bottom: "LOGIN", "ADMIN LOGIN", and "REGISTRATION".

User Registration: The second browser window shows the Metlife 2.0 registration page. It has a similar header and sub-header. The main form is titled "User Registration" and contains fields for first name ("Doggy"), email ("doggy@gmail.com"), and password (""). There are two buttons at the bottom: "REGISTER" and "BACK".

Metlife 2.0 - admin panel
 Order chart Recent Orders Add Product Manage Stuffs

Add A Product

Package Name	<input type="text" value="Gold"/>
Package Price	<input type="text" value="1000"/>
Package Description	
Gold	
Product Image	<input type="button" value="Choose File"/> No file chosen <input type="button" value="Add Product"/>

Metlife 2.0 - admin panel
 Order chart Recent Orders Add Product Manage Stuffs

Recent Orders

Name	Email	Package	Address	Total Price
zaber	zaber@gmail.com	COD	bashundhara r/a	367.00
mihuhit	tasin@gmail.com	Bronze	khilgaon	298.00

MetLife 2.0



MetLife 2.0



MetLife 2.0



CHAPTER-5 CONCLUSION

A. PROBLEM AND SOLUTION

Analysis Phase:

During the Analysis Phase, one of the major problems faced was the confusion around the Rich Picture of the organizational operations since there was no data available regarding those operations. However the information was collected in order to get a better understanding of the system that was being developed.

Designing Phase:

Some problems were faced while creating the EERD and Relational Schema during the Design Phase, However, constant feedbacks from the faculty were enough to overcome those issues.

Implementation Phase:

All the System Requirements were completed successfully.

Front-End Developing tools: HTML, CSS, JavaScript, Google Charts,

Chart JS Back-End Developing tools: PHP, JSON

Database-integration: MySQL

References

[1] <http://www.iub.edu.bd/AboutIUB/ataglance>. [Online].

[2] <http://www.cse.iub.edu.bd/degrees/1>. [Online].

[3] <https://www.betterevaluation.org/methods-approaches/methods/rich-pictures>

[4] <https://www.ibm.com/cloud/blog/bpmn>