Project Electrifiers -

Case Study on Electricity Board Billing Automation

The Electricity Board of the State was facing a huge work load on the billing process of their domestic consumers. Validating the consumer number, getting the respective units consumed for the users, Calculating the bill amount based on the units consumed, and more process were done manually which consumes more working time. So, the Electricity Board decides to outsource the billing process of their domestic consumers to Pioneer Software Company. Help the Pioneer to automate the above task.

Electricity Board has the following business activities that must be automated.

1. Get data and calculate the bill for ‘n’ number of customers.
2. Store the Electricity Bill of each customer to the database

1.4 Scope

The scope of the system is explained through its following modules

1. Get input data related to electricity bill and calculate the bill amount.
2. Store the Electricity Bill of each customer.
3. Retrieve the last ‘N’ number of bill details from the database.

Req. #	1
Req. Name	Get the data from the user and calculate the bill amount for the customer
Req. Description	The units consumed by the customer and the other details of the customer are entered through the Form method. Assign the values to ElectricityBill object and calculate the bill amount for each customer based on the units consumed.

Actors/ Users	Admin
Comments	The admin of the electricity board is responsible for entering the data and calculating the bill amount for all the customers

Req. #	2
Req. Name	Store the electricity bill of each customer.
Req. Description	After calculating the bill amount the electricity board will store the electricity bill of each customer into the database.
Actors/ Users	Admin
Comments	The admin is responsible for adding the EB bill details of each customer into the database.

Req. #	3
Req. Name	Retrieve the last 'N' number of bill details from the database.
Req. Description	In the Form, get the number of records to be retrieved from the user. Retrieve those last 'N' number of records from the database. Using the main, display the bill details as shown in the sample input/output (refer section 2.4).
Actors/ Users	Admin
Comments	The admin is responsible for retrieving and displaying the EB bill details from the database.

SAMPLE INPUT :

"Enter Number of Bills To Be Added :	" 2 "
Enter Consumer Number:	" EB30136 "
Enter Consumer Name:	" Bhaskar "
Enter Units Consumed:	" 250 "

Enter Consumer Number:	” EB23511 “
Enter Consumer Name:	” Raja “
Enter Units Consumed:”	700
Enter Last 'N' Number of Bills To Generate:	2

SAMPLE OUTPUT :

EB0136 Bhaskar 250 Bill Amount : 225 EB23511 Raja 700 Bill Amount : 1900	

Details of last ‘N’ bills: EB Bill for Sid is 350 EB Bill for Peter is 1750	

Data Design

Table Structure:

Table name: ElectricityBill

Column Name	Data type
consumer_number	varchar
consumer_name	varchar
units_consumed	int
bill_amount	float

Design Constraints:

 Use SQLSERVER database to store the data.

The database name is “ElectricityBillDB”.

 The table names and the column names should be the same as specified in the table structure. Consumer Number size is given as 20. Consumer Name size is given as 50.

 The database connection information is to be specified in the “Web.config” file.

Get data and calculate bill amount for each customer

In the 'Form' get the following values,

Number of bills to be added Consumer Number Consumer Name Units Consumed Enter Last 'N' Number of Bills To Generate	

Construct the Electricity Bill object and assign the values. Calculate the bill amount based on the units consumed and display the details as shown in the sample input/output.

Give option to the admin to get multiple user data.

Calculate the bill amount for each Electricity Bill based on the below condition:

Units Consumed	Rate per unit in rupees
<=100	Free(0)
>100 and <=300	1.5
>300 and <=600	3.5
>600 and <=1000	5.5
>1000	7.5

For example 1: If the units Consumed is 650,

- First 100 units are free
- Next 200 units the charges are Rs. 1.50/unit
- Next 300 units the charges are Rs. 3.50/unit
- Remaining 50 units the charges are Rs. 5.5 /unit
- The total bill amount is => 100*0 + 200*1.50 + 300*3.50 + 50*5.5 = 1625

For example 2: If the units Consumed is 1300,

- First 100 units are free
- Next 200 units the charges are Rs. 1.50/unit
- Next 300 units the charges are Rs. 3.50/unit
- Next 400 units the charges are Rs. 5.5 /unit
- Remaining 300 units the charges are Rs. 7.5 /unit
- The total bill amount is => 100*0 + 200*1.50 + 300*3.50 + 400*5.5 + 300*7.5 = 5800

After calculating the bill amount store the values in the ElectricityBill objects and store into the database. Repeat this for the number of users the admin has opted. As per our sample input, it is 2.

Validation 1:

The consumerNumber should start with the characters ”EB” and it should contain 5 numbers (eg: EB11389).
If the Consumer Number is valid then assign the data and calculate the bill amount.
If Consumer Number is NOT valid then throw a built-in Exception named ‘FormatException’ with a message "Invalid Consumer Number".
On printing the exception object “itself” this message must be displayed.

Validation 2:

The Units Consumed should NOT be less than 0. If the given units consumed is less than 0 then display a message “Given units is invalid”. In case of invalid units consumed is given, then prompt the user to enter the units again until a valid (greater or equal to 0) is entered.

```
SAMPLE INPUT / OUTPUT:
“Enter Number of Bills To Be Added : “ 1
"Enter Consumer Number:” EB12623
 “Enter Consumer Name:” Pankaj
“Enter Units Consumed:” -200

 Given units is invalid

“Enter Units Consumed:” -200

Given units is invalid
```

```
“Enter Units Consumed:” 200

Output : EB12623 Pankaj 200 Bill Amount : 150
```

Add Electricity Bill Details

Add the bill details along with the calculated bill amount to the database.

Retrieve ‘N’ number of Bill Details

In the form, get the last ‘n’ number of bill details to be retrieved from database. Retrieve the ‘n’ records. Store each record in Electricity Bill object and add the objects to a ‘Datacontrol’ . Using respective function, display the bill details added to the user

Component Specification

Class Name : BillValidator

Responsibility:

To validate the given units consumed. Units consumed must not be less than zero.

Type(Class)	Fields	Methods
BillValidator	N.A.	public String ValidateUnitsConsumed(int UnitsConsumed)

Method ‘ValidateUnitsConsumed’ must accept units consumed as parameter. Validate units consumed. If units consumed is less than zero then return a message, “Given units is invalid”.

Class Name : ElectricityBill

Responsibility:

This model object holds the state of the electricity bill at all point-in-time.

Type(Class)	Fields	Properties
ElectricityBill	String consumerNumber String consumerName int unitsConsumed double billAmount	String ConsumerNumber String ConsumerName int UnitsConsumed double BillAmount

Note: Keep all the fields ‘private’ and properties ‘public’.

Exception:

Throw a built-in exception “FormatException”, if the consumer number is invalid based on the constraint mentioned above. Implement in such a way that it returns “Invalid Consumer Number” message in the exception object.

Class Name : ElectricityBoard (utility class)

Responsibility:

This class has the supporting methods to automate the billing process given in our scope. This class includes the methods to calculate bill amount and add the bill details to database.

AddBill Method

Type(Class) Responsibilities	Method	Responsibilities
ElectricityBoard	void AddBill(ElectricityBill ebill)	This method should accept an Electricity Bill object and execute a sql query to insert the consumer details into the database

CalculateBill Method

Type(Class)	Method	Responsibilities
ElectricityBoard	void CalculateBill (ElectricityBill ebill)	This method should accept an ElectricityBill object, calculate and set the bill amount based on the units consumed.

Generate_N_BillDetails Method

Type(Class)	Method	Responsibilities
ElectricityBoard	List<ElectricityBill> Generate_N_BillDetails(int num)	This method should accept number of records to be retrieved from database. Store each record in an ElectricityBill object. Add each object to a ‘Datacontrol’ and return it. NOTE : Retrieve the last ‘N’ records.

Class Name : DBHandler(ADO class)

GetConnection Method

Responsibility:

This method should connect to the database by reading the database details from the Web.config file and it should return the connection object.

Type(Class)	Method	Resources
DBHandler	SqlConnection GetConnection()	Web.config file contains the database connection details.

-Implement Master and Themes

-Implement a DLL for Connection to SQL server

-Have a Login Form for Admin and validate the login details

Evaluation Areas

S.No	Description
1.	Declaration of attributes, properties and methods in the class ElectricityBill and ElectricityBoard
2.	Declaration of properties and methods in the class DBHandler
3.	Implementation to create a valid database connection object
4.	Implementation to validate units consumed

5.	Implementation to add bill details to database
6.	Implementation to retrieve bill details from database
7.	Implementation to throw “Exception” for invalid Consumer Number
8.	Calculation of EB bill units less than 100
9.	Calculation of EB bill units between 100 and 300
10.	Calculation of EB bill units between 300 and 600
11.	Calculation of EB bill units between 600 and 1000
12.	Calculation of EB bill units greater than 1000