



Cognizant Academy

Medical Billing System

**C#, ADO.Net and SQL Server Integrated
Capability Test**

Version 1.0

Table of Contents

1.0	Introduction	1
1.1	Purpose of this document	1
1.2	Definitions & Acronyms	1
1.3	Project Overview	1
1.4	Scope	2
1.5	Target Audience	2
1.6	Hardware and Software Requirement	2
1.6.1	Hardware Requirements	2
1.6.2	Software Requirements	2
2.0	Functional Requirements	3
2.1	Functional Requirements	3
2.2	Use case Diagram	4
2.3	System Architecture Diagram	5
2.4	Sample Input / Output	5
3.0	Design Specification	7
3.1	Data Design	7
3.2	Component Details for identified Use Cases	7
3.2.1	Get medical bill data for each patient and store it in database	7
3.2.2	Retrieve all patients medical bill data from database	8
3.2.3	Update bill amount of a patient	8
3.3	Project Structure	8
3.3.1	Screenshot of Project Structure	8
3.4	Component Specification	9
	Class Name: BillAmountValidator	9
3.4.1	BillValidator Method	9
	Class Name: BillInformation (model class)	9
3.4.2	Class Attributes and Properties	9
	Class Name: SQLQuery	10
3.4.3	Public static read-only variables of class	10
	Interface Name: IBillingOperations	10
3.4.4	AddRecord, DisplayRecords and EditRecord method declaration	10
	Class Name: BillingOperations (utility class)	11
3.4.5	AddRecord, DisplayRecords and EditRecord method implementation	11
	Class Name: DBHandler (DAO Class)	11
3.4.6	Connection Property	12
3.5	General Design Constraints	12
4.0	Submission	12
4.1	Code submission instructions	12
5.0	Change Log	13
6.0	Evaluation Areas	13

1.0 Introduction

1.1 Purpose of this document

The Finance Department of CureMe Hospital was facing a huge work load on the Medical Billing process of its patients. Getting the bill details of patients, editing the bill amount of patients and displaying the bills were done manually which consumes more working time. So, CureMe Hospital decides to outsource the Medical Billing process of its patients to Global Tek Software Company. Help the Global Tek to automate the above task.

CureMe Hospital wants the following business activities to be automated.

1. Store the patient's medical bill details in the database.
2. Edit the bill amount of a patient by patient id.
3. Display the medical bill details of all the patients.

1.2 Definitions & Acronyms

Definition / Acronym	Description
Req	Requirement

1.3 Project Overview

This project captures the various concepts, techniques and skills learnt and help to put them into practice using C# with ADO.NET that a software engineer must solve. Admittedly, this would be at a scaled-down level since the purpose is to let the associate experience the various concepts learned in C# as an individual. The individual associate is expected to create a console based application within the specified time.

1.4 Scope

The scope of the system is explained through its following modules:

1. Get input data related to patient's medical bill and store the data to database.
2. Retrieve and display all patients' medical bills from the database.
3. Edit a patient's medical bill amount by patient id.

1.5 Target Audience

Learner Level

1.6 Hardware and Software Requirement

1.6.1 Hardware Requirements

#	Item	Specification/Version

1.6.2 Software Requirements

#	Item	Specification/Version
1.	C#	6
2.	ADO.NET	4.5
3	SQLSERVER	2017

Note: All the required hardware and software will be provided in the Tekstac platform.

2.0 Functional Requirements

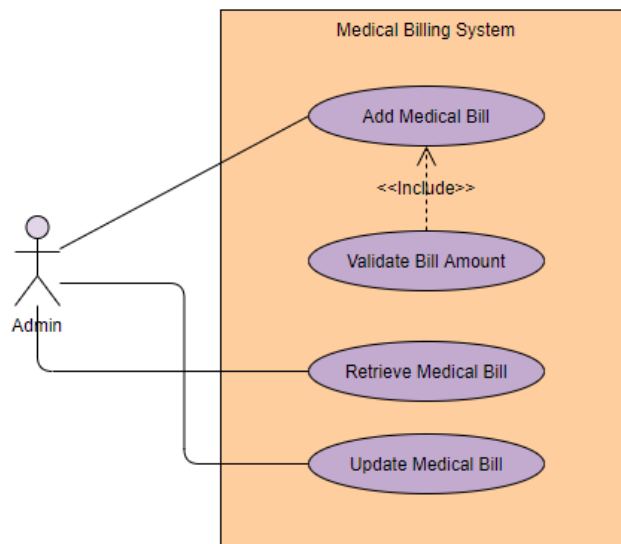
2.1 Functional Requirements

Req. #	1
Req. Name	Get the medical bill details of patients and store it in database.
Req. Description	The medical bill details of patients are entered through the Main method. Store the medical bill details in the database
Actors/ Users	Admin
Comments	The admin is responsible for adding the medical bill detail of each patient into the database.

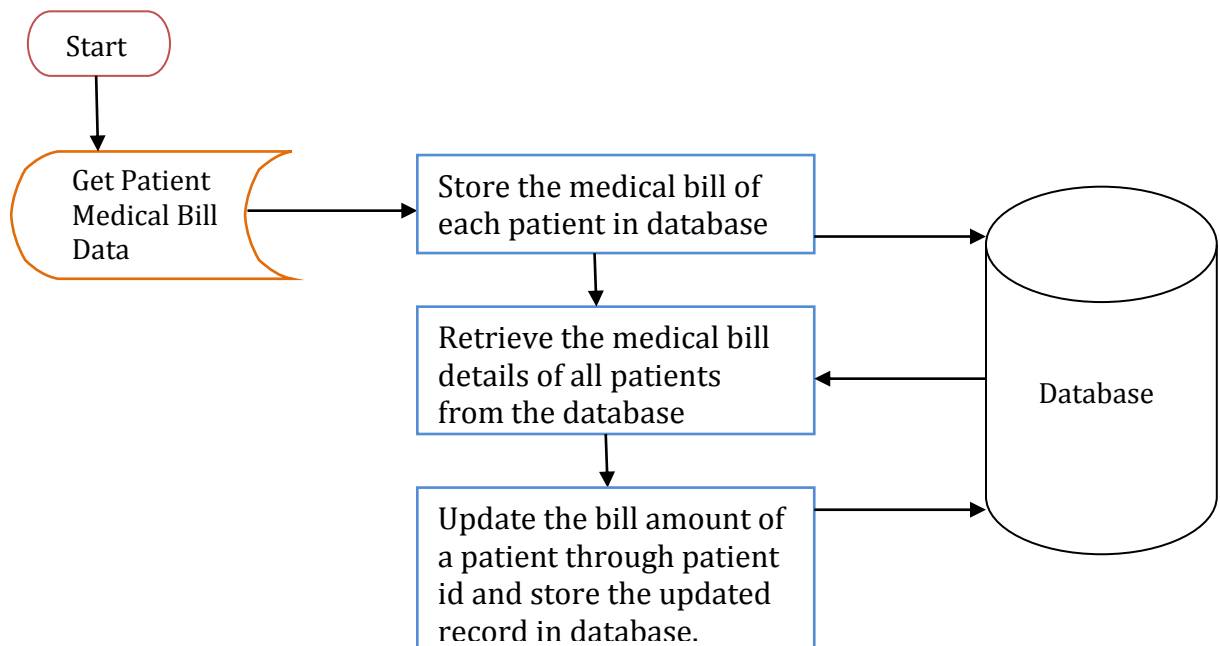
Req. #	2
Req. Name	Retrieve the medical bill details of all patients from the database.
Req. Description	Retrieve the medical bill details of all patients from the database. Using the main method, display the medical bill details as shown in the sample input/output (refer section 2.4).
Actors/ Users	Admin
Comments	The admin is responsible for retrieving and displaying the medical bill details from the database.

Req. #	3
Req. Name	Update bill amount of a patient.
Req. Description	The patient id and new bill amount is entered through main method. Update the bill amount of a patient through patient id and store the updated record in database.
Actors/ Users	Admin
Comments	The admin is responsible for updating the bill amount of a patient by patient id and storing the updated information in database.

2.2 Use case Diagram



2.3 System Architecture Diagram



2.4 Sample Input / Output

Welcome Admin to Medical Billing Application

Menu:

Press 1 to enter new record
Press 2 to display record
Press 3 to edit record
Enter your choice:
1
Enter Patient Name:
Sam
Enter Patient Mobile Number:
9600197755
Enter Bill Amount:
8000
Enter Bill Payment Date(in dd/MM/yyyy format):
20/06/2020
Medical Bill Record Successfully Inserted

Press YES to repeat Menu...Any other key to stop

yes

Menu:

Press 1 to enter new record

Press 2 to display record

Press 3 to edit record

Enter your choice:

2

Display All Medical Bills

Patient Id	Patient Name	Mobile Number	Bill Amount	Payment Date
1	Sam	9600197755	8000	20/06/2020

Press YES to repeat Menu...Any other key to stop

yes

Menu:

Press 1 to enter new record

Press 2 to display record

Press 3 to edit record

Enter your choice:

3

Enter Patient ID to search:

1

Enter New Bill Amount:

10000

Bill amount updated successfully

Press YES to repeat Menu...Any other key to stop

yes

Menu:

Press 1 to enter new record

Press 2 to display record

Press 3 to edit record

Enter your choice:

2

Display All Medical Bills

Patient Id	Patient Name	Mobile Number	Bill Amount	Payment Date
1	Sam	9600197755	10000	20/06/2020

Press YES to repeat Menu...Any other key to stop

n

Thank you for using the application. Have a nice day

3.0 Design Specification

3.1 Data Design

Table Structure:

Table name: tblMedicalBilling	
Column Name	Data type
Patient_Id	int
Patient_Name	varchar
Mobile_Number	bigint
Bill_Amount	float
Payment_Date	date

Design Constraints:

- Use SQLSERVER database to store the data. The database name is "MedicalBillingDB". This is already created for you in Tekstac.
- The table 'tblMedicalBilling' has been created already in Tekstac.
- The table name and the column names should be the same as specified in the table structure.
- ***Patient Id is the primary key for the table whose values are auto-generated in database using Identity property.***
- Patient Name size is given as 100.
- The database connection information is specified in the "App.config" file, which is also provided as part of code skeleton. THIS IS GIVEN ONLY FOR YOUR REFERENCE. You

need NOT change this.

Note: The code skeleton will be available in the Tekstac platform

3.2 Component Details for identified Use Cases

3.2.1 Get medical bill data for each patient and store it in database

In the 'Main' get the following values:

Patient Name
Mobile Number
Bill Amount
Payment Date

Note: The Patient Id value should not be supplied by user but auto-generated in database. Store each patient's medical bill information in database.

Validation:

The Bill Amount should not be less than 0. If the bill amount is less than 0 then display a message " Invalid Bill Amount ".

Note: The validation of bill amount should be done inside the BillAmountValidator class.
--

3.2.2 Retrieve all patients' bill details from the database

Retrieve all patients' bill details from the database. Store each record in BillInformation object and add the objects to a 'List'. Using main method, display the patients' bill details from the 'List'.

3.2.3 Update bill amount of a patient

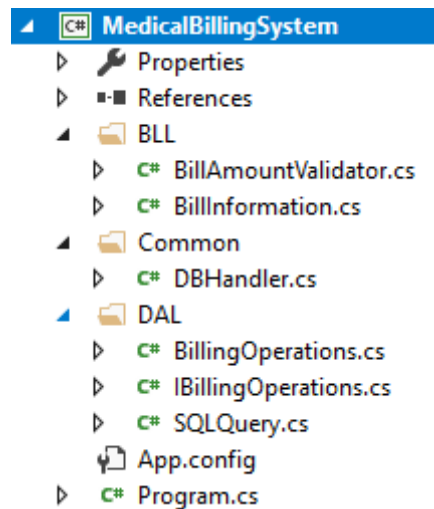
In the 'Main' get the following values:

Patient Id
Bill Amount

The patient id and new bill amount is entered through main method. Update the bill amount of a patient through patient id and store the updated record in database.

3.3 Project Structure

3.3.1 Screenshot of Project Structure



3.4 Component Specification

Folder Name: BLL

Class Name: BillAmountValidator

Responsibility:

To validate the patient's bill amount. Bill amount must not be greater than or equal to zero.

3.4.1 BillValidator Method

Type(Class)	Fields	Methods
BillAmountValidator	N/A	public string BillValidator(double billAmount)

If bill amount is less than zero then **return** a message **"Invalid Bill Amount"**.

Folder Name: BLL

Class Name: BillInformation (model class)

Responsibility:

This model object holds the state of the Patient's Medical Bill information at all point-in-time.

3.4.2 Model Class Fields and Properties

Type(Class)	Properties
BillInformation	int PatientId string PatientName long MobileNumber double BillAmount DateTime PaymentDate

Note: Keep all the properties 'public'. Also include a default constructor and a parameterized constructor.

Folder Name: DAL

Class Name: SQLQuery

Responsibility:

This class will contain public static read-only variables which will contain DML and DQL statements.

3.4.3 SQLQuery Variables

Type(Class)	Variables	Description
SQLQuery	public static readonly string InsertRecord	This variable will contain DML statement to insert record to database
	public static readonly string DisplayRecords	This variable will contain DQL statement to retrieve data from database
	public static readonly string EditRecord	This variable will contain DML statement to update bill amount in database by patient id.

Folder Name: DAL

Interface Name: IBillingOperations

Responsibility:

This Interface will contain Medical Billing Methods declaration which needs to be implemented by BillingOperations class. This interface includes the methods to add patient's medical billing information to database, retrieve all patients' billing records from database and update a patient's bill amount by patient id.

3.4.4 IBillingOperations Methods

Type(Interface)	Method
IBillingOperations	bool AddRecord(BillInformation billinformation) IList<BillInformation> DisplayRecords() bool EditRecord(double billAmount, int patientId)

Folder Name: DAL

Class Name: BillingOperations (utility class)

Responsibility:

This class will implement methods of IBillingOperations interface to automate the Medical Billing process given in our scope.

3.4.5 BillingOperations Methods

Type(Class)	Method	Responsibilities
BillingOperations	public bool AddRecord(BillInformati on billinformation)	This method should accept a BillInformation object and execute a sql query to insert the patient's medical billing details into the database. Return will be true if insertion is successful; else return false for insertion failure.
BillingOperations	public IList<BillInformation> DisplayRecords()	This method will retrieve all patients' medical billing records from the database. Store each record in BillInformation object. Add the objects to a 'List'.
	public bool EditRecord(double billAmount, int patientId)	This method will accept new bill amount and patient id as parameter and execute a sql query to update the patient's medical bill amount into the

		database. Return will true if update is successful; else return false for update failure.
--	--	---

Folder Name: Common

Class Name: DBHandler

Responsibility:

This `ConnectionString` property should connect to the database by reading the database connection information from the *App.config* file.

3.4.6 `ConnectionString` Property

Type(Class)	Property	Resources
DBHandler	public static string <code>ConnectionString</code>	App.config file contains the database connection details.

3.5 General Design Constraints

1. The fields/properties/method/class name should be correctly specified as given in the document.
2. Keep all the classes as '**public**'
3. Do not change the App.config file.
4. Do not change the namespace name.

4.0 Submission

4.1 Code submission instructions

1. Do not change the code skeleton given, as your code will be auto evaluated.

2. You can validate your solution against sample test cases during the assessment duration.
3. Your last submitted solution will be considered for detailed evaluation.
4. Make sure to submit the solution before the time limit. After the assessment duration you will not be allowed to submit the solution.

5.0 Change Log

	Changes Made			
V1.0.0	Initial baseline created on <dd-Mon-yy> by <Name of Author>			
Vx.y.z	<Please refer the configuration control tool / change item status form if the details of changes are maintained separately. If not, the template given below needs to be followed>			
	Section No.	Changed By	Effective Date	Changes Effected

6.0 Evaluation Areas

S.No	Description
1.	Declaration of properties and constructors in the class BillInformation.
2.	Declaration of static property in the class DBHandler
3.	Implementation to validate Bill Amount
4	Declaration of public static read-only variables in class SQLQuery.
5	Declaration of methods AddRecord, DisplayRecords and EditRecord in IBillingOperations interface

6.	Implementation of method AddRecord under class BillingOperations to add patient's medical billing record under database.
7.	Implementation of method DisplayRecords under class BillingOperations to retrieve all patients medical billing details from database
8.	Implementation of method EditRecord under class BillingOperations to update bill amount of a patient by patient id.