MS.NET

Mini Project

Insurance Policy Endoresement

(ePES)

Document Revision History

Date	Revision No.	Author	Summary of Changes
10th Oct. 2017	1	Kishan Chaubey, Nitin Rai	Initial Draft

Table of Contents

INTRODUCTION

This document outlines a mini project for the .NET Line of Technology (LOT). The project is to develop online Policy Endorsement System. This document contains the requirements, work flow of the system and gives guidelines on how to build the functionality gradually in each of the course modules of the .NET LOT.

SETUP CHECKLIST

Minimum System Requirements

- Intel Pentium 4 and above Windows XP, 2000 and 2003
- Memory 1 GB
- Internet Explorer 6.0 or higher
- SQL Server 2012 client and access to SQL Server 2012 server
- Visual Studio 2013

INSTRUCTIONS

- The code modules in the mini project should follow all the coding standards.
- Create a directory by your name in drive **<drive>**. In this directory, create a subdirectory **MiniProject**. Store your Project here.
- You can refer to your course material.
- You may also look up the help provided in the MSDN
- Since this project work will span over couple of months, you will need to take care of maintaining the code

PROBLEM STATEMENT

OBJECTIVE

Development of online Policy Endorsement System (ePES)

A life insurance endorsement is a document that is used in order to make some type of change to the original life insurance policy. This is typically a piece of paper that is attached to the original document. You will be able to read the document and see exactly what has been changed from the original policy that was used by the insurance company.

ABSTRACT OF THE PROJECT

An Insurance company wants to set up On Line policy endorsement facility for their Customers. They sell Life and Non Life products insurance to their customers.

This Online Policy Endorsement System should provide the following features to the customers:

1. Search Policy:
    - The customers should be allowed to search a policy by using Policy number,Customer Id or Name and DOB combination.
    - On clicking search button the policy details should be displayed in a gridview on the same screen.
    - The gridview records should be clickable , and on click the policy details should open in an editable format.
2. View Policy details
    - The policy details should be displayed in editable format using controls like TextBox,Radio Button, DropDown etc.
    - The details screen should include Policy Number, ProductLine (Life/NonLife), Product Name, Insured name, Insured Age , DOB, Gender, Nominee and Relation, Smoker /Non Smoker ,Address, Telephone No, Premium Payment frequency (Monthly,Quarterly,Half Yearly,Annually).

- The fields Product Name, Policy Number, ProductLine, Product Name should be **non-editable**.

3. Update Policy details (Endoresement)
    - The customers should be able update other fields on the page.
4. View Endorsement status
    - This screen should provide details of the endorsement updates performed.
    - This details (fields on which updates were performed, and the status) should be displayed in an readonly gridview. For every update a new row should be visible in the gridview.
    - The status details should be maintained in a different table in the database (e.g. tblEndorsementStatus)

Once the update is submitted, Admin (underwriter) will review the changes and can accordingly either approve or reject the update. This review screen should be displayed only to Admin(Underwriter).
Thus if changes are approved customer can see the changes in details screen as well as on the status screen as approved.
Whereas if the submission is rejected the details screen for customer remains unchanged and status screen will show status as rejected for the last updates.

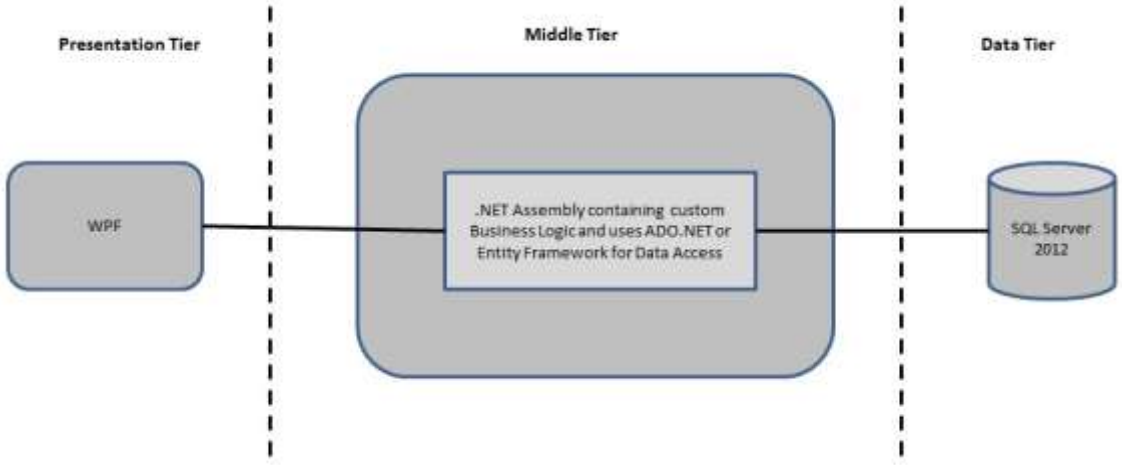
Develop the Application for the above requirement using MS.NET Technology.

FUNCTIONAL COMPONENTS OF THE PROJECT

Application Architecture:
Distributed web applications traditionally to be designed and built across three logical tiers:

- Database Access Layer (DAL)
- Business Logic Layer (BLL)
- Presentation Layer

The DAL refers to the database itself, the stored procedures, and the component that provides an interface to the database. The BLL refers to the component that encapsulates all the business logic of the application. And, the Presentation layer refers to the web application pages.



- **Design guidelines**
- Presentation layer comprises of Winforms and WPF
- Data access layer of 3-tier useADO.NET data access using SQL stored procedures - All the database interaction would be performed using Data Access Component. Most common methods in Data Access Component would be –
  1. Create Connection to the Database
  2. Create Command Object
  3. Set Command Type to Stored Procedure
  4. Create and Populate Parameters
  5. Execute the Command
  6. Close the Connection

TECHNOLOGY USED:

- *Presentation Layer*
  a. *WPF 4.5*

- *Business Layer*
  1. *Business Logic Components and Services :-*
     a. C# 6.0

- *Database Layer*
  1. *Databases:-*
     a. SQL Server 2012

Sr. No	Course	Duration (in PDs)	Functionality to be built
1	MS SQL Server 2012	4	Creating relevant database tables and stored procedures
2	NET Framework 4.6 + C# 6.0	7.5	Developing Business components (C# classes)
3	ADO.NET 4.5	3	Integrating the DAL and BLL And PL
4	LINQ and Entity Framework	2	Creating data model and data context and using LINQ to Entities
5	WPF 4.5	2	Incorporating advanced UI functionality with WPF 4.5
6	Mini Project Presentation	1	The Mini Project Presentation day

NOTE: SATURDAY HALF DAY WILL BE DEVOTED FOR MINI PROJECT

IMPLEMENTATION

SUMMARY OF THE FUNCTIONALITY TO BE BUILT:

The participants need to develop the Online Policy Endorsement System by building the functionality incrementally in each of the course modules of .NET LOT.

GUIDELINES ON THE FUNCTIONALITY TO BE BUILT:

The functionality and components to be built in each of the course modules of .NET LOT is as follows:

1. **Course: SQL Server 2012**

This section describes some of the basic steps involved in designing and creation of the database for the application.

Create Data Model - identify the different tables and fields that we will need, which would later be used for building the rest of the application.

Database Schema - In order to design the database schema, we needed to look at the requirements of the application. In our application, we are going to have a policy and insured information that the insured (customer) can browse and select one for endoresement. So, taking these high level requirements, we can quickly and easily identify the tables and stored procedures in our database by simply identifying many of the verbs and nouns in some of the below user cases:

- Customer will **login** to the application.
- **Customers** can **search** for policy details based on one or more search criteria mentioned above.
- **Customers** can **view** policy details he /she owns.
- **Customers** can **create** endorsement record for the selected policy.

Out of these user cases, we can identify a number of objects, which are nouns in our sentences.

- Customers
- Insurance Products (Life/Non-life), name
- Policy
- Endorsement

Taking these objects, we can easily identify our main tables in the database.

a. Create the following database tables:

i. Customers: This will contain the list of Customers with their personal information like name, address, telephone, gender, DOB, smoker/non smoker, hobbies
ii. Insurance Products: This contains the list of products available..
iii. Policy: This will contain policy information like policy number, customer number the policy belongs to and product id
iv. Endorsement: This will contain the details of changes requested for. This is a transaction table.
v. Login: This will contain Login ID and password by customer ID as the key.

b. All the tables will contain audit columns:
   1. Create ID
   2. Create Date
   3. Update ID
   4. Update Date

2. Course: WPF 4.5

a) Develop the protoypes for the following functionalities:

i. Login Page : This page will allow user to provide their login credential for authentication
ii. Home Page : This page will provide details based on the user
iii. Search Policy Page : This page will allow the user to search policy based on different criteria
iv. View Policy Page : This page will provide the user with all the required policy details
v. Update Policy Page : This page will allow user to update policy details

vi. View Endorsment Status : This pages will allow the user to view Endorsement status

b. In this course you need to develop the user interface using WPF. The screens should include the fields as per the functionality mentioned above. Include Validations where ever required.

3. **Course: C# 6.0 and ADO.NET** 4.5
   a. Develop business components (C# classes) for the following functionality:
   i. Customer class : This class will contain methods for managing Customer information
   ii. Insurance Product Class : This class will contain methods for managing all the insurance product details
   iii. Policy Class : This class will contain methods for managing Policy details of customers
   iv. Endorsment class : This class will contain methods for managing endorsement details
   v. Login Class : This class will contain methods for performing authentication activities.

You need to create Layered Architecture which comprises of PresentationLayer (winforms,WPF),Business Logic Layer (C# Classes) and DAL Layer(Using ADO.Net 4.5 or Linq and Entity Framework)

DAL Layer of ADO.NET 4.5 will include all the required code snippets for CRUD Operations.

All the CRUD operations should use SQL Server Stored Procedures(For Insert,Update,Delete & Search).

The connectionString should be stored in the configuration file only.

OR

4. **Course: LINQ and EF**
    a. Using Database First Approach create the entity data model form the database created in the earlier steps. The model will consist of the following entities
        i. Customer class : This class/entity will hold Customer Information
        ii. Insurance Product Class : This class/entity will hold Insurance product information
        iii. Policy Class : This class/entity will hold policy information
        iv. Endorsment class : This class/entity will hold endorsement information
        v. Login Class : This class/entity will hold login information

    b. Use data context to perform the CRUD operations and implement required logic using LINQ(like retrieving data, sorting data, searching data etc)

5. **Course: ADO.NET 4.5 and WPF 4.5**
    a. Integrate all screens (ssWPF) with business components (C# classes) to complete the entire functionality.

1.1.1 Project Evaluation Guidelines

The project it is to be evaluated based on the following five parameters:

1) Proper Database Structure and UI designing as per the specifications –(15 Marks)
    a. Proper Database Design and Stored Procedure
    b. Visual look and feel of the UI

2) Project Completion – (20 Marks)
    a. Timely Completion of the project
    b. Integration of all component of the system
3) Defect free execution – (30 Marks)
    a. Error free execution of individual modules and the whole system
    b. Validation
    c. Functionality as per the specified requirements
4) Compliance of standard and guidelines – (15 Marks)
    a. Appropriate comments entries
    b. Adherence to naming conventions for classes, functions, variables and files
    c. Simplicity of user interface and screen layouts
    d. Maintainability of codes (for example, no one function should be more than 100 lines)
5) Group Presentation and Query handling – (20 Marks)
    a. Participants (Group of 3 to 4) to present the project with UML Diagrams(use case diagram and one of the sequence or activity diagram) and PPT

