MS.NET

Mini Project

Blood Bank Management System

(BMS)

Document Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Revision No.** | **Author** | **Summary of Changes** |
| 29-May-2017 | 1.0 | Shital Patil | Initial Draft |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[Introduction 4](#_Toc304553070)

[Setup Checklist 4](#_Toc304553071)

[Instructions 4](#_Toc304553072)

[Problem Statement 5](#_Toc304553073)

[Objective 5](#_Toc304553074)

[Abstract of the project 5](#_Toc304553075)

[Functional components of the project 5](#_Toc304553076)

[Technology used: 8](#_Toc304553077)

[Implementation 9](#_Toc304553078)

[Summary of the functionality to be built: 9](#_Toc304553079)

[Guidelines on the functionality to be built: 9](#_Toc304553080)

Introduction

This document outlines a mini project for the .NET Line of Technology (LOT). The project is to develop Blood Bank Management 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 2007, 2008 and 2010
* Memory 4 GB
* Internet Explorer 8.0 or higher
* SQL Server 2012 client and access to SQL Server 2012 server
* Visual Studio 2015/2017

## 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 Blood Bank Management System

**Abstract of the project**

**Blood Bank Management System** allows to store, process, retrieve and analyze information with respective to the administrative and inventory management within a blood bank.

Blood Bank Management system will have following features:

* Blood Donation Camp Management
* Blood Donor Management – Donor Registration, Managing donor information, recording their physical and medical details
* Blood Bank Stock Management
* Transferring blood online from one blood bank to another

Users of the System will be: Blood Bank Administrator

Role of Blood Bank Administrator:

* Login to System
* Arrange Blood Donation Camp
* Register the Donor either in the blood donation camp or in any hospital
* Add the blood details in inventory
* Update the blood details in inventory, if it is given to any hospital or other blood bank
* Delete the blood details in case it is expired, unsuitable
* Transfer the blood from one blood bank to another blood bank
* View Blood Bank Inventory details
* View Donor List

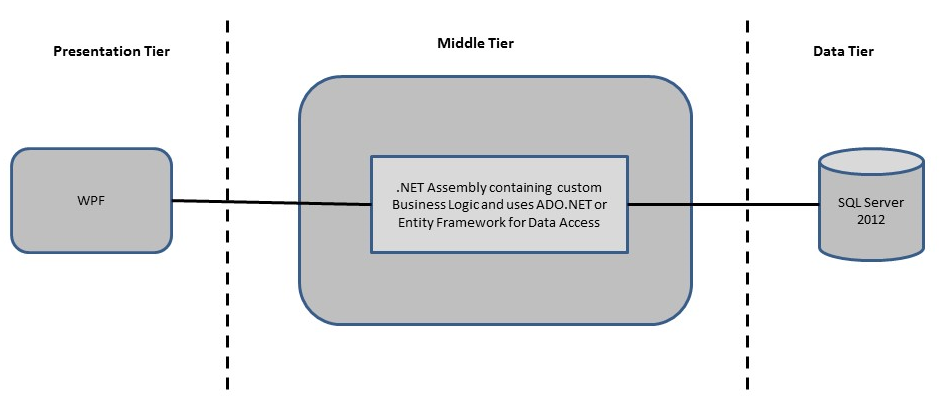
## Functional components of the project

**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**

* All the exceptions/errors to be captured and user friendly message to be displayed on the CommonError page.
* Data access layer of 3-tier use ADO.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* 
      1. *WPF 4.5*
    - *Business Layer*
      1. *Business Logic Components and Services :-* 
         1. C# 5.0 / 6.0
    - *Database Layer*
      1. *Databases:-*
         1. SQL Server 2012+

Implementation

## Summary of the functionality to be built:

The participants need to develop the Blood Bank Management System by building the functionality incrementally in each of the course modules of .NET LOT.

|  |  |
| --- | --- |
|  | |
| **Sr. No** | | **Course** | **Duration** | **Functionality to be built** |
| **(in PDs)** |
| 1 | | MS SQL Server 2012+ | 4 | Creating relevant database tables and stored procedures |
| 2 | | NET Framework 4.5 + C# 5.0/6.0 | 7.5 | Developing Business components (C# classes) |
| 3 | | ADO.NET 4.5 | 3 | Integrating the DAL ,BLL and PL |
| 4 | | LINQ to 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

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

* 1. Create the following database tables with following fields: [make your assumptions in case you require few more fields]
     + 1. BloodBank : BloodBankID, BloodBankName, Address, City, ContactNumber, UserID, Password
       2. BloodDonor : BloodDonorID, FirstName, LastName, Address, City, MobileNo, BloodGroup
       3. BloodDonationCamp : BloodDonationCampID, CampName, Address, City, BloodBank, CampStartDate, CampEndDate
       4. Hospital : HospitalID, HospitalName, Address, City, ContactNo
       5. BloodDonorDonation : BloodDonationID, BloodDonorID, BloodDonationDate, NumberofBottle, Weight, HBCount
       6. BloodInventory : BloodInventoryID, BloodGroup, NumberofBottles, BloodBankID, ExpiryDate

1. **Course: WPF 4.5**
   1. Develop the prototypes for following functionalities:
      1. **Login / Sign In**: Login screen would display asking blood bank user to enter ‘User Id’ & ‘Password’. If the supplied user credentials are valid the HomePage would be displayed, else appropriate Login error message would be displayed
      2. **Home Page**: On successful user authentication (validation of userid/password provided by the user in login screen) the homepage would be displayed. The Homepage would contain below sections/contents:

Header section: The header section would be common across all the pages and would mainly have –

* Sign In Link – On click it would take to Login page
* Search: Search for information

Main content section: It would display generic Welcome message giving overview of the site. Should have menu for Blood Donation Camp, Blood Donor, Blood Bank, Hospital, Inventory for Blood

* 1. For Blood Donation Camp :
     + - 1. Blood Bank User will get option to Arrange Blood Donation Camp, Modify the details or view the details of blood donation camp
         2. Blood Bank User will get arrange option, where they can add camp details
         3. Blood Bank User will get edit option only for those camp whose start date is greater than today’s date
         4. Blood Bank User can view the information of camps
  2. For Blood Donor:
     + - 1. User will get option to add, update or delete the donor’s information personal information
         2. User will get option to Donate blood, where they can add current donation information about any donor
         3. View the details of all donors
  3. For Blood Bank
     1. User will get option to add, update or delete the blood bank details
     2. View the details of all Blood Banks
  4. For Hospital
     1. User will get option to add, update or delete the hospital details
     2. View the details of all Hospitals
  5. For Inventory of Blood
     1. User will be able to view the blood available in all blood banks
     2. User can transfer blood from one bank to another bank or hospital as per that updation in the quantity should happen
     3. Delete the blood details if it is expired

1. **Course: C# 5.0/6.0 and ADO.NET 4.5**
   1. Develop business components (C# classes) for the following functionality:
      1. BloodBank Class :- This class will contain methods which will allow to manage all BloodBank
      2. BloodDonor class :- This class will contain methods to manage Blood Donors
      3. BloodDonationCamp Class :- This class will contain methods to manage Blood Donation Camp Details
      4. Hospital class :- This class will contain methods to manage Hospital details
      5. BloodDonorDonation Class :- This class will contain methods to manage Donation details
      6. BloodInventory Class : This class will contain methods to manage inventory of blood bank

You need to create Layered Architecture which comprises of Presentation Layer (WPF), Business Logic Layer (C# Classes) and DAL Layer (Using ADO.Net 4.5 / 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 and Search).

The connectionString should be stored in the configuration file only.

**OR**

1. **Course : LINQ and EF**
   1. Use the database first approach and create the Entity data model consist of the following entities :
      1. BloodBank Class :- This class will hold the basic details of the Blood Bank
      2. BloodDonor Class :- This class will hold the details of Blood Donor.
      3. BloodDonationCamp Class :- This class will hold the details of Blood Donation Camp
      4. Hospital Class :- This class will hold the details of Hospital
      5. BloodDonorDonation Class :- This class will hold the details of Blood Donation
      6. BloodInventoy Class :- This class will hold the details of Blood Inventory
   2. Use Data context to perform CRUD operations and Implement the required logic using LINQ (like retrieving data, sorting data, searching data etc).
2. **Course: ADO.NET 4.5 and WPF 4.5**
   1. Integrate all screens (WPF pages) with business components (C# classes) to complete the entire functionality

### 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)
   1. Proper Database Design and Stored Procedure
   2. Visual look and feel of the UI
2. Project Completion – (20 Marks)
   1. Timely Completion of the project
   2. Integration of all component of the system
3. Defect free execution – (30 Marks)
   1. Error free execution of individual modules and the whole system
   2. Validation
   3. Functionality as per the specified requirements
4. Compliance of standard and guidelines – (15 Marks)
   1. Appropriate comments entries
   2. Adherence to naming conventions for classes, functions, variables and files
   3. Simplicity of user interface and screen layouts
   4. Maintainability of codes (for example, no one function should be more than 100 lines)
5. Group Presentation and Query handling – (20 Marks)
   1. 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