MS.NET
Mini Project
Supply Chain Management System (SCMS)

Document Revision History

Date	Revision No.	Author	Summary of Changes

Table of Contents

INTRODUCTION

This document outlines a mini project for the .NET Line of Technology (LOT). The project is to develop Supply Chain 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 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 Supply Chain Management System (SCMS)

Abstract of the project

Company 'eShoppy has recently planned to develop 'Supply Chain Management' application which helps to manage details about customers, dealers, products and orders. It allows admin to create, display, edit and remove products.

Company is currently looking for a developer who can develop small module called as Supply Chain Management application for entire "eShoppy solutions".

SCMS aims to manage complete life cycle of products, orders viz add (create product – where all info will be captured for the first time), edit (modify product info), Remove (delete product) data from system. Along with that Dealer Registration, Customer Registration, List of pending orders (for Admin), List of pending orders (for dealer), Track order for dealer and placing an order. Also, SCMS should allow login for Admin, Customer and Dealer login.

- **Phase 1 :** The system will first develop using C# only – where products, customers and dealers data will be store as a Collection Classes. For user interaction, system will use Console Application

- **Phase 2 :** Later on data will be store in MS SQL Server database; system will use ADO.NET or LINQ and Entity Framework for the same. User Interface will be designed using WPF

- **Phase 3** : SCMS will become web based application, following MVC design pattern. Here the application will be develop in ASP.NET MVC.

Macro level Operations/offerings:

1. The app should allow admin to login and view list of pending orders for Admin. Even Customers and Dealers will also be able to Register and then will be able to Login to place an order and Dealer will accept the order and finally, he/she will dispatch the product to customer.

2. Once admin has successfully logged in, admin should be able to add new products, edit and remove existing product details

3. During edit and remove, application should take confirmation from the admin, do you want to really edit and remove product details

4. Once admin successfully logoff, Other than login and home views, admin should not able to view Add Product, Edit Product, List Product views.

5. While login to application, adding new product, customer, dealer and editing existing product, customer, dealer details, application should validate product, customer and dealer inputs.

MODULE LIST and MODULE DETAILS

CREATE PRODUCT

Following info need to capture

- ProductId (must be unique)
- ProductName
- Quantity
- Price
- DateandTimeofProductAdded

SEARCH PRODUCT

- Customer should be able to search for an Product by Product Id and Product Name.

MODIFY PRODUCT

- Search (by product Id) select (from Product summary) about the product and modify the info of product. System should show existing data/info of product and should support modify.

REMOVE PRODUCT

- Search (by product Id) or select (from Product summary) about the product and remove the product. System should ask for confirmation and on confirmation the product will be removed.

PRODUCT SUMMARY (VIEW)

- System should show (display) product list in a tabular format.

REGISTER AS DEALER

- The following details must be captured while registering as dealer like dealer code, Organization Name, Contact Person, Contact Number, Address (Warehouse), Official Email, Address (Regd. Office), Username and Password.
- Note: Accept Username and Password for login.

REGISTER AS CUSTOMER

- The following details must be captured while registering as customer like Customer Id, Organization Name, Contact Person, Contact Number, Delivery Address, Official Email, Username and Password
- Note: Username and Password for login.
- Customer will place an order for the product by searching with product id. As soon as he/she enters product id text box or he/she may have combo box for selection of product id, remaining text boxes will be auto filled from database. Here, OrderId must be autogenerated and Order date must be either current date or future date [not greater than week from current date]
- As soon as customer will place an order, order will be received by Admin, Admin will decide and assign the dealer.
- After assigning dealer, dealer has to decide whether he wants to accept the order reject the order

- Dealer can dispatch the order to customer

Constrains

- Proper validation is required
- System must show appropriate massage on all activity (whether activity is successful or failure)
- User must have proper menu to select the activity (create, modify, search, view, remove) that user want to perform.

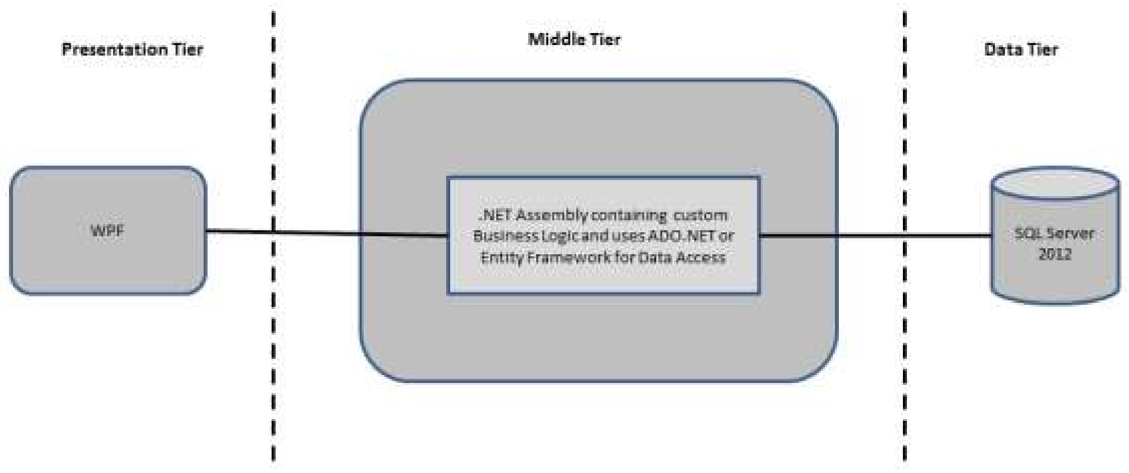
FUNCTIONAL COMPONENTS OF THE PROJECT

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.


Presentation Tier
Middle Tier
Data Tier
WPF
.NET Assembly containing custom Business Logic and uses ADO.NET or Entity Framework for Data Access
SQL Server 2012

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 Entity Framework data access using SQL stored procedures - All the database interaction would be performed using Data Access Component.

TECHNOLOGY USED:

- *Presentation Layer*
  1. *Console Application, WPF, ASP.NET MVC 5*
- *Business Layer*
  1. *Business Logic Components and Services :-*
     a. C# 5.0
- *Database Layer*
  1. *Databases:-*
     a. SQL Server 2012

IMPLEMENTATION

SUMMARY OF THE FUNCTIONALITY TO BE BUILT:

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

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# 7.0 + Introduction to WPF	10	Developing Business components (C# classes)
3	ADO.NET with LINQ and Entity Framework	4	Creating data model and data context and using LINQ to entities
4	ASP.NET MVC 5	4	Incorporating advanced UI functionality with ASP.NET MVC 5
5	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.

	Table Name - Customer	
Field Name	Primary Key(System Generated)	Data Type
CustomerId (must be unique)	Primary Key(Begin with C)	Text
OrganizationName	Not Null	Text
ContactPerson	Not Null	Text
ContactNumber	Not Null	Text
DeliveryAddress		Text
OfficialEmail		Text
UserName	Unique Key	Text
Password		Text

Table Name - Dealer		
Field Name	Constraints	Data Type
DealerCode (must be unique)	Primary Key(Begin with D) (AutoGenerated)	Text
OrganizationName	Not Null	Text
ContactPerson	Not Null	Text
ContactNumber	Not Null	Text
WarehouseAddress		Text

OfficialEmail		Text
RegdOfficeAddress		Text
UserName	Unique Key	Text
Password		Text

Table Name - Product		
Field Name	Constraints	Data Type
ProductId (must be unique)	Primary Key(Begin with P) (AutoGenerated)	Text
ProductName	Not Null	Text
Quantity		Number
Price		Number
ProductAddDateTime		DateTime

Table Name - ProductOrder		
Field Name	Constraints	Data Type
OrderId	(Begin with O) Foreign Key (Order)	Text
ProductId	Foreign Key (Product)	Text
ExpectedDeliveryDate	ProductOrderDate+7	DateTime
CustomerId	Foreign Key (Customer)	Text
DealerCode	Foreign Key (Dealer)	Text
DispatchedStatus	Check Constraint (y, n)	Text

Table Name - Order		
Field Name	Constraints	Data Type
OrderId	(Begin with O) (AutoGenerated)	Text
ProductOrderDate		DateTime