MS.NET

Mini Project

Easy Housing Solution

(EHS)

Document Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Revision No.** | **Author** | **Summary of Changes** |
| 23rd Nov 2017 | 1 | Sangeetha C | Initial Draft |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[Introduction 4](file:///C:/Users/sangchan/AppData/Local/Microsoft/Windows/Temporary%20Internet%20Files/Content.Outlook/ZQ033TGZ/MiniProject_EasyHousingSolution_CaseStudy_16.doc#_Toc304553070)

[Setup Checklist 4](file:///C:/Users/sangchan/AppData/Local/Microsoft/Windows/Temporary%20Internet%20Files/Content.Outlook/ZQ033TGZ/MiniProject_EasyHousingSolution_CaseStudy_16.doc#_Toc304553071)

[Instructions 4](file:///C:/Users/sangchan/AppData/Local/Microsoft/Windows/Temporary%20Internet%20Files/Content.Outlook/ZQ033TGZ/MiniProject_EasyHousingSolution_CaseStudy_16.doc#_Toc304553072)

[Problem Statement 5](file:///C:/Users/sangchan/AppData/Local/Microsoft/Windows/Temporary%20Internet%20Files/Content.Outlook/ZQ033TGZ/MiniProject_EasyHousingSolution_CaseStudy_16.doc#_Toc304553073)

[Objective 5](file:///C:/Users/sangchan/AppData/Local/Microsoft/Windows/Temporary%20Internet%20Files/Content.Outlook/ZQ033TGZ/MiniProject_EasyHousingSolution_CaseStudy_16.doc#_Toc304553074)

[Abstract of the project 5](file:///C:/Users/sangchan/AppData/Local/Microsoft/Windows/Temporary%20Internet%20Files/Content.Outlook/ZQ033TGZ/MiniProject_EasyHousingSolution_CaseStudy_16.doc#Abstract)

[Functional components of the project 5](file:///C:/Users/sangchan/AppData/Local/Microsoft/Windows/Temporary%20Internet%20Files/Content.Outlook/ZQ033TGZ/MiniProject_EasyHousingSolution_CaseStudy_16.doc#FunctionalComponents)

[Technology used: 8](file:///C:/Users/sangchan/AppData/Local/Microsoft/Windows/Temporary%20Internet%20Files/Content.Outlook/ZQ033TGZ/MiniProject_EasyHousingSolution_CaseStudy_16.doc#_Toc304553077)

[Implementation 10](file:///C:/Users/sangchan/AppData/Local/Microsoft/Windows/Temporary%20Internet%20Files/Content.Outlook/ZQ033TGZ/MiniProject_EasyHousingSolution_CaseStudy_16.doc#_Toc304553078)

[Summary of the functionality to be built: 10](file:///C:/Users/sangchan/AppData/Local/Microsoft/Windows/Temporary%20Internet%20Files/Content.Outlook/ZQ033TGZ/MiniProject_EasyHousingSolution_CaseStudy_16.doc#_Toc304553079)

[Guidelines on the functionality to be built: 10](file:///C:/Users/sangchan/AppData/Local/Microsoft/Windows/Temporary%20Internet%20Files/Content.Outlook/ZQ033TGZ/MiniProject_EasyHousingSolution_CaseStudy_16.doc#_Toc304553080)

[Evaluation and assessment parameters](file:///C:/Users/sangchan/AppData/Local/Microsoft/Windows/Temporary%20Internet%20Files/Content.Outlook/ZQ033TGZ/MiniProject_EasyHousingSolution_CaseStudy_16.doc#EvaluationParameters) 13

# Introduction

This document outlines a mini project for the .NET Line of Technology (LOT). The project is to develop an Easy Housing Solution (EHS). This document contains the 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 7 and above Windows 10
* Memory 3 GB
* Internet Explorer 11.0 or higher
* SQL Server 2012 client and access to SQL Server 2012 server
* Visual Studio 2013 or Visual Studio 2015

## 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 Easy Housing Solution (EHS)

## Abstract of the project

This project is aimed at developing Easy Housing Solution (EHS) for administrators and Customers. It is proposed to develop a system to provide web based Housing Solution system which will allow users to study, compare, buy, sell & then maintain his/her property details in a systematic way. Typical operations include registering and login of the user, searching houses for buying or renting. Owners can add their property, upload images and get the property verified from the Easy Housing Company.

Easy Housing Solution Ltd launched in 2017, is an India – based startup company under the New India – Make in India initiative. The EHS project has been initiated by Housing Solution to help customers. This will be achieved

1. Reducing paper work.
2. Password secured Asset Information.
3. All property details information & all processing is Free of cost.

Role of Administrator

Administrator is the primary user of the application and will have the complete control over the property which would be added and listed on the site. Administrator details will be already available in the system. The following will be responsibilities of administrator in context of EHS:

* Verify Property: Administrator is responsible to verify the property added by the owners. Only verified properties would be listed on the site.
* View Property by Region: Administrator would be able to list out all the property by region.
* View Property by Owner: Administrator would be able to list all property listed by an owner either for sell or rent.
* Delete Property: This option will enable the administrator to deactivate a property from being displayed on the site. The administrator must provide proper reason for DE activing the listed property.

Role of Owner or Seller

Owner or Seller would have to register themselves with Easy Housing Solution site. Once registered to the system they can performs the following operations.

* Add Property: As a part of this operation owner will record the details of the property which they intent to sell or rent out.

Following information can be recorded by executive

* Property Type
* Property Name
* Address
* Region
* Property Option (Sell or Rent)
* Description
* Images
* Price Range
* Initial Deposit (In case of Rent)
* Landmark
* Edit Property: As a part of this operation owner will be able to edit the property details.
* View all Verified Property: This option will help the owner to view all the property listed by him/her and which have be verified by the Site Administrator.
* View all De activated Property: This option will allow the owner to view all the property listed by him/her and which have been de activated by the Site Administrator.
* Upload Image: This option would allow the owner to upload images of the property. The Owner can upload a maximum of 6 images per property.

Role of Consumer or Buyer

Consumer or Buyer can visit the Site and search for property. But to buy or rent a particular property he/she would have to register. Registered Buyer can compare two properties and can performs the following operations.

* Search Property: As a part of this operation buyer would be able to search all listed property based on region or Type.
* Sort Property by price: This option would help the buyer to sort the fetched data according to price.
* Add to Cart: This option would allow the buyer to add a particular property to his/her Cart for future reference.
* Get Owner Contact Details: In the list of fetched property there would be a button called Get Contact Details. This button would display the Owner Email Id and Registered Phone No.
* View Cart: This option would allow the buyer to view all the property which are there in his/her Cart.
* Delete from Cart: This option would allow the buyer to delete a property from his/her Cart.
* View Property Details: This option would provide the details of the property which the buyer has selected.

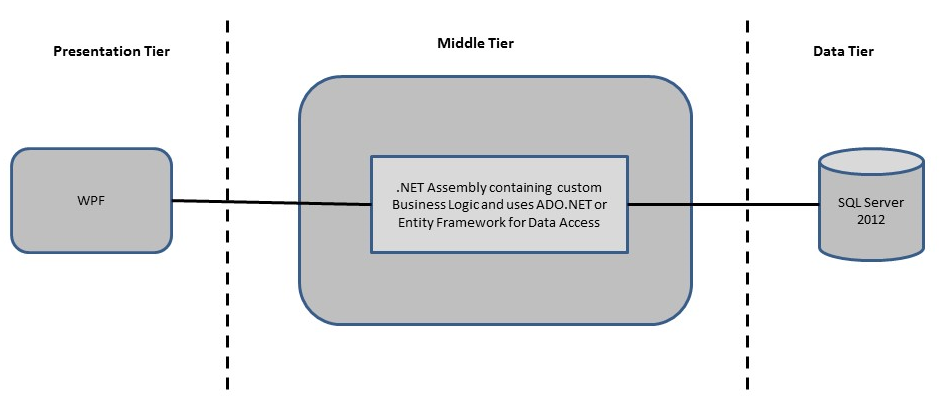
## Functional components of the project

**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 (PL)

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 Common Error page (to be done during PLP).
* Data access layer Data access layer Use ADO.NET. Make use of 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
    - *Database Layer*
      1. *Databases:-*
         1. SQL Server 2012

|  |  |  |  |
| --- | --- | --- | --- |
| **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 | 7.5 | Developing Business components (C# classes) |
| 3 | ADO.NET 4.5 | 3 | Developing Data Access Component (C# classes) |
| 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 Easy Housing Solution (EHS) 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 -** Create the following database tables (make your assumptions in case you require few more fields)

1. Users:- This will contains details of the user of EHS application
2. Seller: - This will contains details of seller or owner
3. Buyer:- This will store the details of buyers
4. Property:- This will store the details of properties
5. Images:- This will be used to store images of the property
6. Cart:- This will store the property liked by buyers
7. State:- This will store the state information
8. City:- This would store the city information

The Structure of the above mentioned tables

1. **Users**: - UserName varchar(25) primary key, Password varchar(25) not null, UserType varchar(15)

UserType – Admin, Buyer or Seller

1. **Seller**:- SellerId int identity primary key, UserName varchar(25) not null, FirstName varchar(25) not null, LastName varchar(25), DateofBirth date not null, PhoneNo varchar(10) not null, Address varchar(250) not null, StateId int foreign key not null, CityId int foreign key not null, EmailId varchar(50) not null.
2. **State: -** StateId int identity primary key, StateName varchar(30)
3. **City: -** CityId int identity primary key, CityName varchar(50) not null, StateId int foreign key not null
4. **Buyer:-** BuyerId int identity primary key, FirstName varchar(25) not null, LastName varchar(25), DateOfBirth date not null, PhoneNo varchar(10) not null, EmailId varchar(50) not null
5. **Property:-** PropertyId int identity primary key, PropertyName varchar(50) not null, PropertyType varchar(15) not null, PropertyOption varchar(10) not null, Description varchar(250), Address varchar(250) not null, PriceRange money not null, InitialDeposit money not null, Landmark varchar(25) not null, IsActive bit not null, SellerId int foreign key not null
6. **Images:-** ImageId int identity primary key, PropertyId int foreign key not null, Image not null
7. **Cart**:- CartId int identity primary key, BuyerId int foreign key not null, PropertyId int foreign key not null

**2. Course: WPF 4.5**

**a. Develop the prototypes for following functionalities:**

1. **Registration**: - This page will allow buyer/Seller to register themselves.
2. **Add Property: -** This page allows seller to add new property into the system.
3. **Upload Image: -** This page allows seller to upload images for a given property.
4. **Edit Property: -** This page allows the seller to edit the details of an existing property.
5. **View All Verified Property: -** This page will allow the seller to view all the verified property which belongs to him/her.
6. **View All De-Activated Property: -** This page allows seller to view all the de-activated property which belongs to him/her.
7. **View Property by Region: -** This page allows administrator to view all the property which belongs to a particular region.
8. **View Property by Owner: -** This page allows administrator to view all the property added by a particular seller.
9. **Verify or De-Active Property: -** This page will allow the administrator to either approve or de-activate a property.
10. **Search Property: -** This page will allow buyer to search property. This page will also have the option to sort the searched data based on price. Default sort order would be based on property name.
11. **View Property Details: -** This page will allow buyer to view the details of the property. This will also have an option which would display the seller contact details.
12. **Add To Cart: -** This page will allow buyer to add a property to his/her cart for future reference.
13. **View Cart: -** This page will allow buyer to view all the property which he/she has added to their cart.
14. **Delete from Cart: -** This page will allow buyer to delete a property from the cart.
15. **Login: -** This page will allow users to login to the system.

**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# 5.0 and ADO.NET** **4.5**

a. Develop business components (C# classes) for the above functionality:

1. Users :- This class will contain methods for managing users
2. Sellers :- This class will contain methods to manage sellers
3. Buyers :- This class will contain methods to manage buyers
4. Property :- This class will contain methods to work with property
5. Images :- This class will contain methods to manage images
6. Cart: - This class will contain methods to manage cart.

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 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, and Delete & Search).

The connectionString should be stored in the configuration file only.

**OR**

**4. Course: LINQ and EF**

a) Using the Database first approach of Entity framework create an entity data model form the database created in earlier steps. The Entity data model should consist of the following entities

1. User: - This entity/class will hold users information.
2. Property :- This entity/class will hold property information
3. Seller :- This entity/class will hold seller information
4. Buyer :- This entity/class will hold buyer information
5. Image:- This entity/class will hold image information
6. Cart :- This entity/class will hold cart information

b) Using DataContext to perform CRUD operations and use LINQ to implement the required business logic in the application (like retrieving data, sorting data, searching data etc.).

5. **Course: ADO.NET 4.5 and WPF 4.5**

* 1. Integrate all screens (WPF) with business components (C# classes) to complete the entire functionality.
  2. Proper layout and styles should be implemented. The UI should be user friendly and easy to use.
  3. All validation should be done and proper error message is displayed to the end user.

#### 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