MS.NET

Parallel Project

Housing Society Management System

(HSMS)

Document Revision History

Date	Revision No.	Author	Summary of Changes

Table of Contents

INTRODUCTION

This document outlines a parallel project for the .NET Line of Technology (LOT). The project is to develop Housing Society 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 parallel project should follow all the coding standards.
- Create a directory by your name in drive **<drive>**. In this directory, create a subdirectory **ParallelProject**. 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 Housing Society Management System (HSMS)

Abstract of the project

Housing Society Management System (HSMS) helps society members to log complaint related to society issues and view their complaint status. There are two actors in this application.

(Note: Currently we have to create application for Society Members only)

1) Society Member: Can perform the following tasks:

- Add Complaint
- Update Complaint
- Delete Complaint
- Can Search for a Complaint and view its details, on the basis of its ComplaintNo (Society members should be able to get their own complaints only).
- Get List of their Complaints and view the details.

2) Society Committee Member: Can perform the following tasks:

- Can Search for a Complaint and view its details, on the basis of its Complaint.

- Get List of Complaints and view the details.
- Edit a Complaint and change its status.

- **Phase 1 :** The system will first develop using C# only – where society 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** : HSMS will become web based application, following MVC design pattern. Here the application will be develop in ASP.NET MVC.

Macro level Operations/offerings:

1. Society Member:
    a. Add Complaint Details: ComplaintId, Category (Accounting, CCTV, Electrical, WaterLeakage etc.), Block, FlatNo, Description etc.
    b. Modify Complaint Details
    c. Remove Complaint
    d. Get List of Complaint and view the details like ComplaintId, Category, Block, FlatNo, Description, etc. (Society Members should be able to view their complaints only)

e. Search for a Complaint and view its details, on the basis of its ComplaintId. (Society Members should be able to Search their complaints only)

2. Society Committee Member:

   a. Modify Complaint Details. Should be able to add a note, change status and close the complaint.

   b. Search for a Complaint and view its details, on the basis of its ComplaintId.

   c. Get List of Complaints logged by society members and view the details like ComplaintId, Category, Block, FlatNo, Description, etc.

MODULE LIST and MODULE DETAILS

1) Society Member

CREATE COMPLAINT

Following info need to capture

- ComplaintId
- Category (Accounting, CCTV, Electrical, WaterLeakage etc.)
- Block
- FlatNo

- Description
- Date

SEARCH COMPLAINT

User should be able to search a Complaint by ComplaintId.

MODIFY COMPLAINT

Search (By complaintId) a Complaint and modify its details. System should show existing data/info of Complaint and support modify one, more or all info.

REMOVE COMPLAINT

Search (by complaintId) a Complaint and remove the Complaint. System should ask for confirmation and on confirmation the data will be removed.

COMPLAINT SUMMARY (VIEW)

System should show (display) Complaint list in a tabular format (one row for each Complaint, and columns for Complaint details). It is not required to show all the details of a Complaint in a table; only important info like – ComplaintId, Category, Block, FlatNo, Description, Date should be displayed.

2) **Society Committee Member**

SEARCH COMPLAINT

User should be able to search a Complaint by ComplaintId.

MODIFY COMPLAINT

Search (By complaintId) a Complaint and modify its details. Should be able to add a note, change status and close the complaint.

REMOVE COMPLAINT

Search (by complaintId) a Complaint and remove the Complaint. System should ask for confirmation and on confirmation the data will be removed.

COMPLAINT SUMMARY (VIEW)

System should show (display) Complaint list in a tabular format (one row for each Complaint, and columns for Complaint details). It is not required to show all the details of a Complaint in a table; only important info like – ComplaintId, Category, Block, FlatNo, Description, Date should be displayed.

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.

- For “Complaint” Entity:
    - ComplaintId : String. Should be 7 characters long. With first 2 characters upper case letters followed by hyphen and then 4 digits
    - Category: String (Accounting, Parking, Electrical, WaterLeakage)
    - Block: String.
    - FlatNo: Numeric
    - Date: Should be datetime
    - Status: String (Automatically set for Society Members) Default: “Pending”
    - Status Options: Pending, UnderInvestigation, Closed
    - Note : String (Will be added by society committee member only)
- For “ComplaintCategory” Entity:
    - Id: Numeric
    - Description: String. (Accounting, Parking, Electrical, WaterLeakage)
- For “ComplaintStatus” Entity:
    - Id: Numeric
    - Description: String (Pending, UnderInvestigation, Closed)

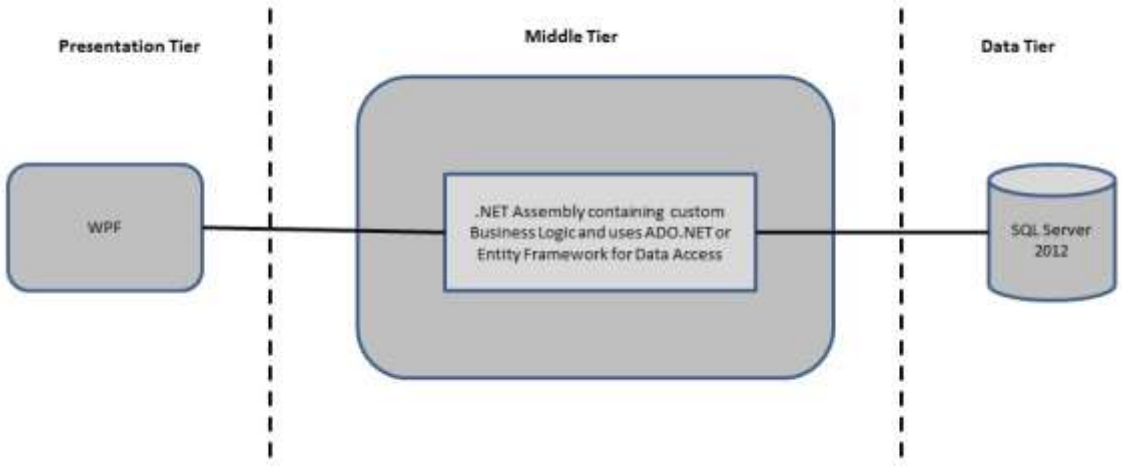
FUNCTIONAL COMPONENTS OF THE PROJECT

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 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 Customer 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	Parallel Project Presentation	1	The Parallel Project Presentation day

Note: Saturday half day will be devoted for Parallel 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.

a. Create the following database tables with following fields: [make your assumptions in case you require few more fields]

Table Name: Complaint		
Field Name	**Constraint**	**Data Type**
Id	Primary Key (System Generated)	Int
CategoryId	Foreign Key (References Id column from Table - ComplaintCategory)	int
Block	Not Null	varchar
FlatNo	Not Null	Int
Date	Not Null	datetime
StatusId	Foreign Key (References Id column from Table - ComplaintStatus)	Int
Note	Null	varchar

Table Name: ComplaintCategory		
Field Name	**Constraint**	**Data Type**
Id	Primary Key (System Generated)	Int
Description	Check (Accounting, Parking, Electrical, WaterLeakage)	varchar

Table Name: ComplaintStatus		
Field Name	**Constraint**	**Data Type**
Id	Primary Key (System Generated)	Int
Description	Check(Pending, Under Investigation, Closed)	varchar

