MS.NET

PLP

Music-JukeBox-System (MJS)

Document Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Revision No.** | **Author** | **Summary of Changes** |
| 13-07-2017 | 1.0 | Pankaj 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 Music-JukeBox-System (MJS). 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 7, 8 and 10
* 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 **PLP**. 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 **Music Jukebox System** (MJS)

**Abstract of the project**

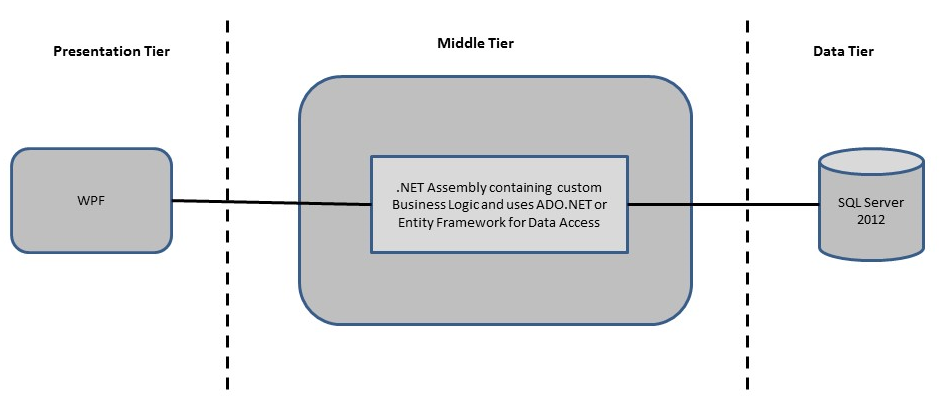
The main aim of developing this **Music Jukebox System** it’s a collection of MP3 songs of different languages in one place where users can search, based on the year also play and listen the songs in our website at free of cost only is to provide user friendly tool for music web sites. This is one type of online music Player. Most of the websites now a day’s Music products through online but download music files with free of cost makes problem with piracy so this is the best place to stop music piracy over the internet. Here the customer should login through Music store website and play & Lesion selected music files like songs, and private albums by using papal. Then selected music files can be downloaded directly to the local system of the customer. The main advantage is it is user friendly, provide us 24 hours customer service, and decrease the manual efforts and Time.

* + **Music Jukebox System** allows to store, process, retrieve and analyze information with respective to the administrative and inventory management within a Music Store.
  + Music Jukebox System will have following features:
* Collection of MP3 songs of different languages in one place.
* Music Jukebox User System - User Registration, Managing User information
* Music Search – based on Singer, Year, Movie, Actor, Actress, Language
  + Users of the System will be: Music Jukebox System Administrator
  + Role of Music Jukebox System Administrator:
* Login to System
* Upload, Remove the Music
* Add and Remove Singer
* Add and Remove Customer
* Update Album details
* Delete the Album details
* Search Customer, Singer
* View Album  
  + Role of Music Jukebox System User:
* Login to System
* Upload, Remove the Music in Playlist
* Update Playlist details
* Delete the Playlist
* Search Music, Singer, Album
* View Album
* Upload songs
* Download songs

## 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**

* Clean code/html content separation using server controls
* Usage of Html Controls and Server side controls
* All the exceptions/errors to be captured and user friendly message to be displayed on the CommonError page.
* You can write Data Access Logic in WCF / Web API by using ADO.NET Entity Framework including 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

**OR**

LINQ and Entity Framework

## Technology used:

* + - *Presentation Layer* 
      1. *Web Components:–* 
         1. ASP.NET MVC 5.0
         2. jQuery
    - *Business Layer*
      1. *Business Logic Components and Services :-* 
         1. C# 5.0
         2. WCF Services
         3. Web API
    - *Database Layer*
      1. *Databases:-*
         1. SQL Server 2012
         2. ADO.Net
         3. LINQ and Entity Framework

Implementation

## Summary of the functionality to be built:

The participants need to develop the Music JukeBox 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 | Web Basics (HTML, CSS, JS, Bootstrap, etc.) | | 3 | Web Designing |
| 2 | ASP.NET 4.5 + MVC + SignalR | | 8 | Designing User interface & Building application using MVC architecture. |
| 3 | Web API + WCF | | 4 | Creating services for any business logic & DB operation using ADO.NET EF. |
| 4 | Testing (TDD+BDD) + DevOps with TFS | | 6 | Implementing TDD & BDD for Web Application Development. DevOps using TFS for Versioning |
| 5 | WPF 4.5 | | 2 | Incorporating advanced UI functionality with WPF 4.5 |
| 6 | PLP & PLP Presentation | | 8 | PLP Presentation will be given to BU Members |

## 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:

* **Course: SQL Server 2012**
  + Use can use the tables created at the time of Mini Project
* **Course: Asp.Net MVC 5.0**
  + Use the existing WPF screens and migrate to ASP.Net MVC Views
* **Course: C# 5.0 & WCF / Web API**
  + Convert your existing BLL classes prepared at the time of Mini Project into WCF/Web API Services whenever applicable.
* **Course: ADO.NET 4.5, LINQ and Entity Framework**
  + Write Data Access Components to perform any CRUD operations using ADO.NET & Entity Framework.
  + Use LINQ to query record sets returned by Data Access Components
  + Build the web application and deploy on the IIS server and test the application.