



MS.NET Mini Project **Job Portal System (JPS)**



Document Revision History

Date	Revision No.	Author	Summary of Changes



Table of Contents

Introduction	4
Setup Checklist	4
Instructions	4
Problem Statement	5
Objective	5
Development of Job Portal System (JPS)	5
Functional components of the project	9
Technology used:	10
Implementation	11
Summary of the functionality to be built:	11
Guidelines on the functionality to be built:	12



INTRODUCTION

This document outlines a mini project for the .NET Line of Technology (LOT). The project is to develop Job Portal 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 Job Portal System (JPS)

Abstract of the project

Job Portal System (JPS) provides job seeking facilities to job seekers across the globe. It is an online system through which job seekers can post their details and search for jobs.

The Online Job Portal System should support the following users.

1. Administrator (A)
 2. User(Job Seeker) (U)
- **Phase 1** : The system will first develop using C# only – where job seekers 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** : JPS will become web based application, following MVC design pattern. Here the application will be develop in ASP.NET MVC.

Macro level Operations/offerings:

1. Add Job Details
-



2. Edit Job Details
3. Delete Job Details
4. Search and View Details
5. Manage the Loan Processing by Loan Officials
6. Manage the Users

MODULE LIST and MODULE DETAILS

1. Administrator

Add Job Details

- As part of this operation, the admin should have the ability to add job details into the system and get confirmation on successful entry and JobID is auto-generated.

Edit Job Details

- As part of this operation, the admin should have the ability to edit Job details in the system by picking JobID from the list, and get confirmation on successful entry.

Delete Job Details

- As part of this operation, the admin should have the ability to delete Job details in the system by picking JobID from the list whenever particular job request from employer is closed.



Search and View details

- Search criteria are:
 - Name starts with
 - Year of experience

2. User (Job Seeker)

- User(Job Seeker) :User Registration An unregistered user should be able to register into the system with the following details.
 - Registration details include (all mandatory):
 - FirstName
 - LastName
 - Age
 - Gender
 - Password
 - Address
 - PhoneNo

When the registration is complete, a UserID (numeric) will be auto-generated and displayed to the user



- **User Login**

- The User should be able to do the following operations once he has logged in with his unique UserID and password.

- **Search/View jobs**

Constrains

- Proper validation is required
- System must show appropriate message 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.

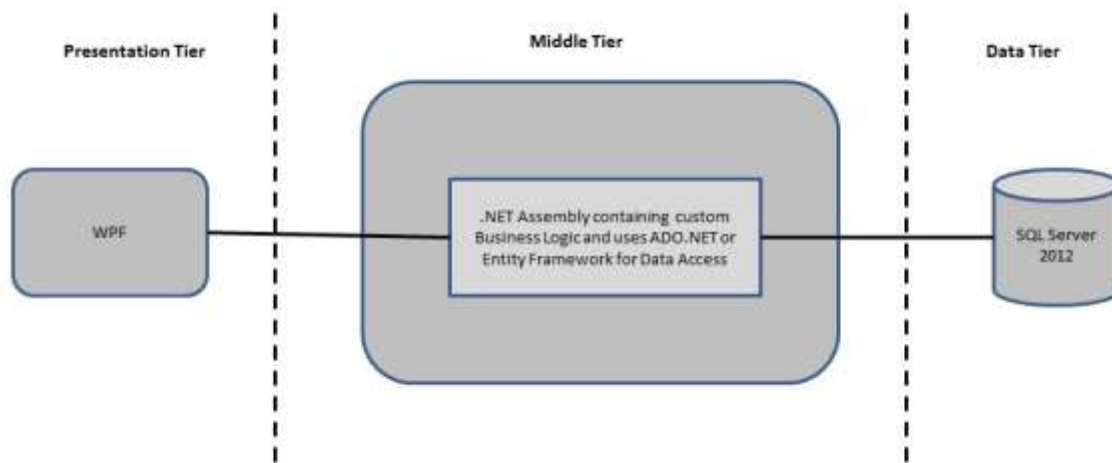
FUNCTIONAL COMPONENTS OF THE PROJECT

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.





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 Loan 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 4.6 + 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.

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

Table Name: User Information		
Field Name	Data Type	Description
UserID	VARCHAR	It will be a numeric value sequentially generated by the system and need not be passed from the client. It will be added by the system whenever a new user gets registered.
Password	VARCHAR(9)	
First Name	VARCHAR(15)	First Name of the user
Last Name	VARCHAR(15)	Last Name of the user
Age	NUMBER	Age of the user
Gender	VARCHAR(7)	Gender of the user
Address	VARCHAR(30)	Address of the user
PhoneNo	VARCHAR(10)	Phone No of the user
UserType	VARCHAR(10)	A-Administrator



MS.NET MINI PROJECT
JOB PORTAL SYSTEM (JPS)

Table Name: LoginCredentials		
Field Name	Data Type	Description
UserType	VARCHAR(10)	A-Administrator
UserID	VARCHAR(9)	User Identification
Password	VARCHAR(9)	Password

Table Name: Job Details		
Field Name	Data Type	Description
JobID	VARCHAR	It will be a numeric value sequentially generated by the system and need not be passed from the client. It will be added by the system whenever a new user gets registered.
Employer	VARCHAR(9)	
Address	VARCHAR(15)	
ContactNumber	VARCHAR(15)	
ContactEmailID	VARCHAR	
SkillsRequired	VARCHAR(7)	
Qualification	VARCHAR(30)	
Location	VARCHAR(10)	
Salary	NUMBER (10)	
NoOfVacancies	VARCHAR(10)	