P-3233

LIGHT WEIGHT DEVELOPMENT ENVIRONMENT FOR ALL VERSIONS OF .NET

PROJECT REPORT

Submitted By

J.JOSEPH SURESH

Register No.: 0720300016

in partial fulfilment for the award of the degree

Of

MASTER OF COMPUTER APPLICATIONS

in

COMPUTER APPLICATIONS

KUMARAGURU COLLEGE OF TECHNOLOGY

(An Autonomous Institution Affiliated to Anna University, Coimbatore)

May, 2010

KUMARAGURU COLLEGE OF TECHNOLOGY

(An Autonomous Institution Affiliated to Anna University, Coimbatore)

COIMBATORE – 641 006.

Department of Computer Applications

PROJECT WORK

MAY 2010

This is to certify that the project entitled

LIGHT WEIGHT DEVELOPMENT ENVIRONMENT FOR ALL VERSIONS OF .NET

is the bonafide record of project work done by

J.JOSEPH SURESH

Register No: 0720300016

of MCA (Computer Applications) during the year 2009-2010.

7/5/10

Project Guide

Head of the Department

Submitted for the Project Viva-Voce examination held on 17.05.2010

17/5/10

Internal Examiner

17/05/2010

External Examiner

DECLARATION

I affirm that the project work titled **LIGHT WEIGHT DEVELOPMENT ENVIRONMENT FOR ALL VERSIONS OF .NET** being submitted in partial fulfilment for the award of **MASTER OF COMPUTER APPLICATIONS** is the original work carried out by me. It has not formed the part of any other project work submitted for award of any degree or diploma, either in this or any other University.

(Signature of the Candidate)

J.JOSEPH SURESH,
0720300016.

I certify that the declaration made above by the candidate is true.

Signature of the Guide,

V.GEETHA
Assistant Professor, MCA

InQ Technologies

23th April, 2010

PROJECT COMPLETION CERTIFICATE

This is to certify that Mr. **Joseph Suresh J.** (Reg No: 0720300016), final year M.C.A student of Kumaraguru College of Technology has done his project work "**Light-Weight Development Environment for any version of .Net**" from 5th Dec 2009 to 23rd April 2010. His conduct and character during the above period were good.

We wish him all the best in his endeavors to pursue a career in IT field.

Regards,

For InQ Technologies,

Ramya K

Project Manager.

InQ Technologies Chennai-600 017

New # 201, Habibullah Road, T Nagar, Chennai - 600 017.
Tel: 44.4202 4958 Mob: 91.98416 90969 info@inqtechnologies.com www.inqtechnologies.com

ACKNOWLEDGEMENT

The Satisfaction that accompanies the successful completion of any task would be incomplete without mentioning the names of people who made it possible, whose constant guidance and encouragement crowns all efforts with success.

I wish to express my deep unfathomable feeling of gratitude and indebtedness to **Dr.S.Ramachandran, Principal** and **Dr.J.Shanmugam, Director** Kumaraguru College of Technology, Coimbatore for the successful completion of the project work.

I am very glad to express a special word of thanks to **Dr.A.Muthukumar,** Professor and Course Coordinator, Department of Computer Applications, Kumaraguru College of Technology, Coimbatore for encouraging me to do this work.

I am very much indebted to **Ms.V.Geetha,** Assistant Professor, Project Coordinator Department of Computer Applications, Kumaraguru College of Technology, Coimbatore for her complete assistance, guidance and support given to me throughout my project.

It is my pleasure to express my profound gratitude to **InQ Technologies, Chennai** for admitting into this project. I am thankful to **Ms.S.Ramya Krishnan,** InQ Technologies, Chennai, for this excellent guidance, timely suggestions and constant support in all my endeavors.

Finally, I owe a lot to my beloved parents and family members and to my friends and to my department staffs for their help and co-operation to complete this project successfully.

TABLE OF CONTENTS

ABSTRACT

LIGHT WEIGHT DEVELOPMENT ENVIRONMENT FOR ALL VERSIONS OF .NET Project supports different types of .net frameworks. It includes .net 2003, .net 2005.

This project is aimed at providing an application environment which is similar to the Visual Studio environment and reducing the usage of the memory. It provides support to develop the WINDOW, CONSOLE, CLASSLIBRARY application projects.

The .net 2003 supports framework versions of 1.0 & .net 2005 supports for frame work version 2.0. This project supports all versions of .net frame works including .net 2008

The light weight development environment for all versions of .net looks similar to visual studio environment. The user who uses this environment can develop the CONSOLE, WINDOW, DLL applications. This application doesn't contain web services option. In this project the main aim is to reduce the size of memory occupied by Visual Studio software which occupies more space. To create an application through Visual Studio the whole platform should be installed but using Light Weight Development tool the amount of memory could be reduced. Applications can be created by copying & pasting this development environment in any system.

This system has been developed using .net 2005 environment.

LIST OF FIGURES

CHAPTER-1

INTRODUCTION

1.1 SYSTEM OVERVIEW

Light weight development environment for all versions of .net supports all .net frameworks. It looks like an application environment. It has a typical Integrated Development Environment. It is as similar to Visual Studio, but this system has different versions of framework of Visual Studio software. Visual Studio software occupies lot of memory.

Light weight development environment for all versions of .net aims to reduce the memory as there is no need to install it in our system. Users can work with this environment and develop the WINDOW, CONSOLE, CLASSLIBRARY application projects.

The .NET 2003 supports only for framework 1.0& .NET 2005 supports only for framework2.0, but the light weight development environment will support any type of framework.

1.2 OBJECTIVES OF THE PROJECT

The objectives of the light weight development environment for all versions of .net are

1. This system occupies very less space in memory.
2. No need to install, just copy and paste in our system.
3. It will support any version of .net frameworks.

1.3 EXISTING SYSTEM

The existing system is occupying more space in memory, and it contains the web services. The web services are used to develop the web application which are not used in this developing project.

The drawbacks of the existing system are:

1. The existing system occupies lot of space in memory.
2. The existing systems are not supported for other versions of frameworks.

1.4 PROPOSED SYSTEM

The proposed system is very useful to users, it supports any version of .net, and it occupies less memory. There is no need to install the whole platform, instead copy and paste the light weight tool in any system. The light weight tool occupies less amount of memory but has the most of the functionalities of Visual Studio.

CHAPTER-2

SYSTEM REQUIREMENT SPECIFICATION

This chapter is a technical specification of requirement for the software product. And it explains about what are the front end and back end tools used for this system.

2.1 REQUIREMENTS

This chapter specifies hardware and software requirements needed for project creation and it also includes

HARDWARE CONFIGURATION

Processor	: Pentium IV Processor
RAM	: 512MB to 1GB
Hard Disk	: 10GB
Processor speed	: 600MHZ to 1GHZ

SOFTWARE CONFIGURATION

Language	: C#.Net
Operating System	: Windows-XP

2.1.1 User Interfaces

Application Selection and New Project Creation Module:

In this system environment creating a new application is done with use of 'New Project' dialog. The user can also open an existing project through this module.

Solution Explorer and Reference Manager Module

The solution explorer and reference manager are created for the project related reference files, forms and classes. And those files are added to the solution explorer of this system.

Code Editor Module

This module helps the user for developing the code for the project. This module also helps the user developing three types of applications. They are window, console and class library applications.

Add New Items and Class Designer Module

If the newly created project needs any class, file, form then the add new item option from file menu helps to create those new items.

Configuration manager module:

Configuration manager module manages the framework, language selection and the type of application. When ever the configuration manager can't set the properties correctly it shows some errors.

2.1.2 Software Interfaces

Microsoft.NET Framework

The .NET Framework is a new computing platform that simplifies application development in the highly distributed environment of the Internet. The .NET Framework is designed to fulfill the following objectives:

To provide a consistent object-oriented programming environment whether object code is stored and executed locally, executed locally but Internet-distributed, or executed remotely.To provide a code-execution environment that minimizes software deployment and versioning conflicts.

The .NET Framework has two main components: the common language runtime and the .NET Framework class library. The common language runtime is the foundation of the .NET Framework. You can think of the runtime as an agent that manages code at execution time, providing core services such as memory management, thread management. It enforces strict type safety and other forms of code accuracy that ensure security and robustness.

Features of the Common Language Runtime

The common language runtime manages memory, thread execution, code execution, code safety verification, compilation, and other system services. These features are intrinsic to the managed code that runs on the common language runtime.

With regards to security, managed components are awarded varying degrees of trust, depending on a number of factors that include their origin (such as the Internet, enterprise network, or local computer). This means that a managed component might or might not be able to perform file-access operations, registry-access operations, or other sensitive functions, even if it is being used in the same active application.

The common language runtime manages memory, thread execution, code execution, code safety verification, compilation, and other system services. These features are intrinsic to the managed code that runs on the common language runtime.

.NET Framework Class Library

The .NET Framework class library is a collection of reusable types that tightly integrate with the common language runtime. The class library is object oriented, providing types from which your own managed code can derive functionality. This not only makes the .NET Framework types easy to use, but also reduces the time associated with learning new features of the .NET Framework. In addition, third-party components can integrate seamlessly with classes in the .NET Framework.

For example, the .NET Framework collection classes implement a set of interfaces that you can use to develop your own collection classes. Your collection classes will blend seamlessly with the classes in the .NET Framework.

As you would expect from an object-oriented class library, the .NET Framework types enable you to accomplish a range of common programming tasks, including tasks such as string management, data collection, and database connectivity.

CHAPTER-3

SOFTWARE ANALYSIS AND DESIGN

Software design is the most creative and challenging phase in the life cycle of system development. The requirements of the proposed system are depicted by system flow diagrams.

3.1 DECOMPOSITION DESCRIPTION

The modules of this project have been explained in the following chapter.

3.1.1 MODULE DECOMPOSITION

3.1.1.1 Application Selection and New Project Creation Module

Under this module user can select any one of the application and create a new project. Whenever the user selects the particular application type, it is depended on user choice. In this module the user can create the following three types of applications:

- Windows Application
- Class library Application
- Console Application

Next the user will select any one of the languages to develop the project. The three languages available in the system are as follows:

- VB.NET
- C#.NET
- J#.NET

The applications & languages given above are selected and developed during the creation of new projects. This project creation is shown in figure 3.1.1.1.

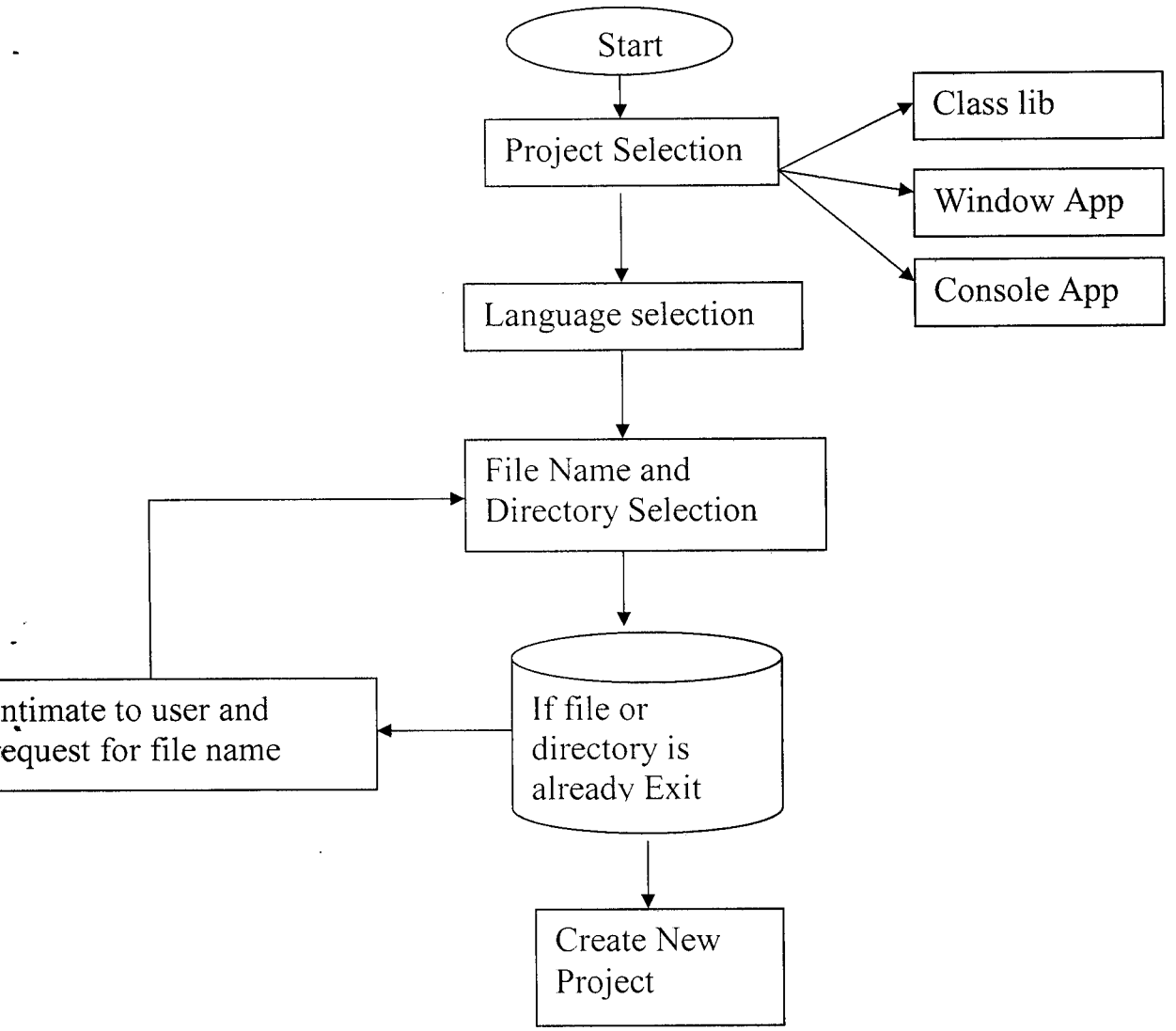

Fig: 3.1.1.1 Application Selection and New Project Creation Module

3.1.1.2 Solution Explorer and Reference Manager Module

In this module we are going to develop the solution explorer. In this solution explorer the system develops the file types, default references and add new references. These three properties have some other sub properties as shown in figure 3.1.1.2.

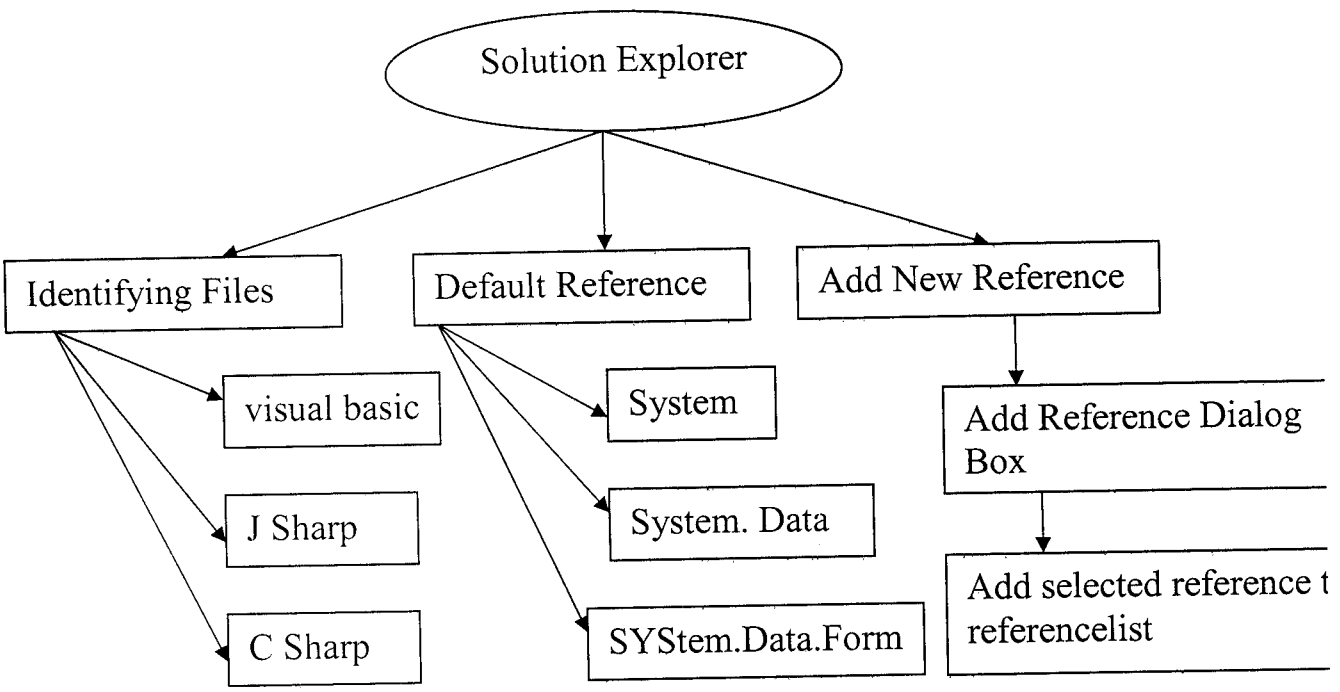

Fig: 3.1.1.2: Solution Explorer and Reference Manager Module

3.1.1.3 Add New Items and Class Designer Module

In this module the user selects the type of file that is to be created. There are text files, class files and windows forms. If the file with the mentioned name already exists then there will be an error message to the user to create another file. This add new item module has been depicted in figure 3.1.1.3.

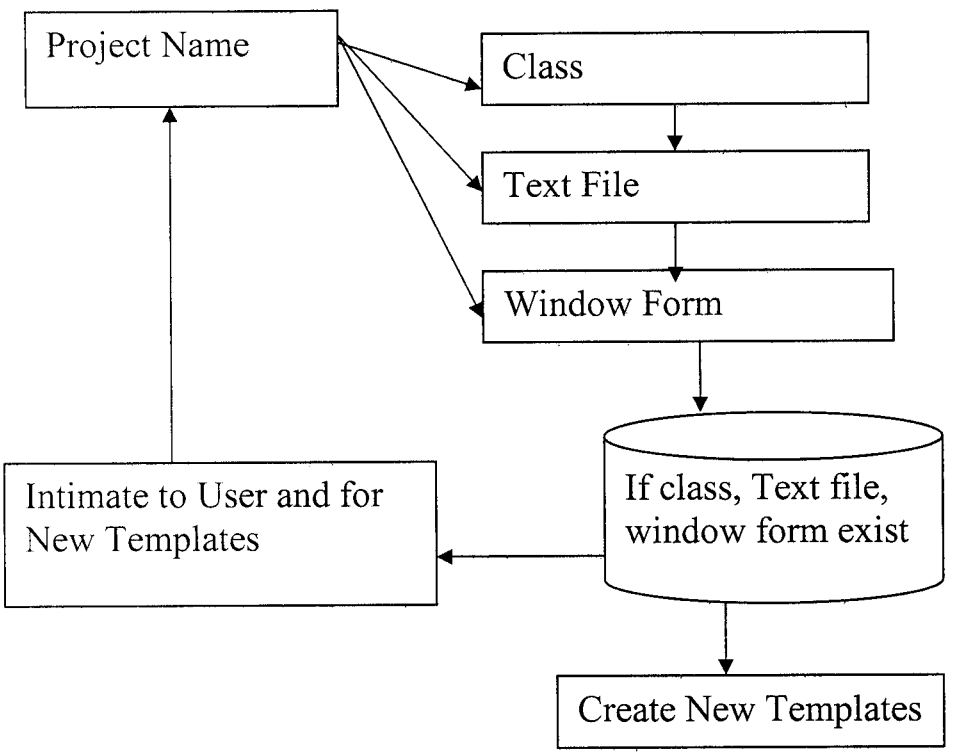

Fig: **3.1.1.3: Add New Items and Class Designer Module**

3.1.1.4 Configuration Manager Module

The total project will depend on the configuration manager, because the configuration manager can set all the properties so that the project can be executed successfully. The configuration manager will set the build specification, output specification, locate the framework path. Configuration manager module is shown in figure 3.1.1.4.

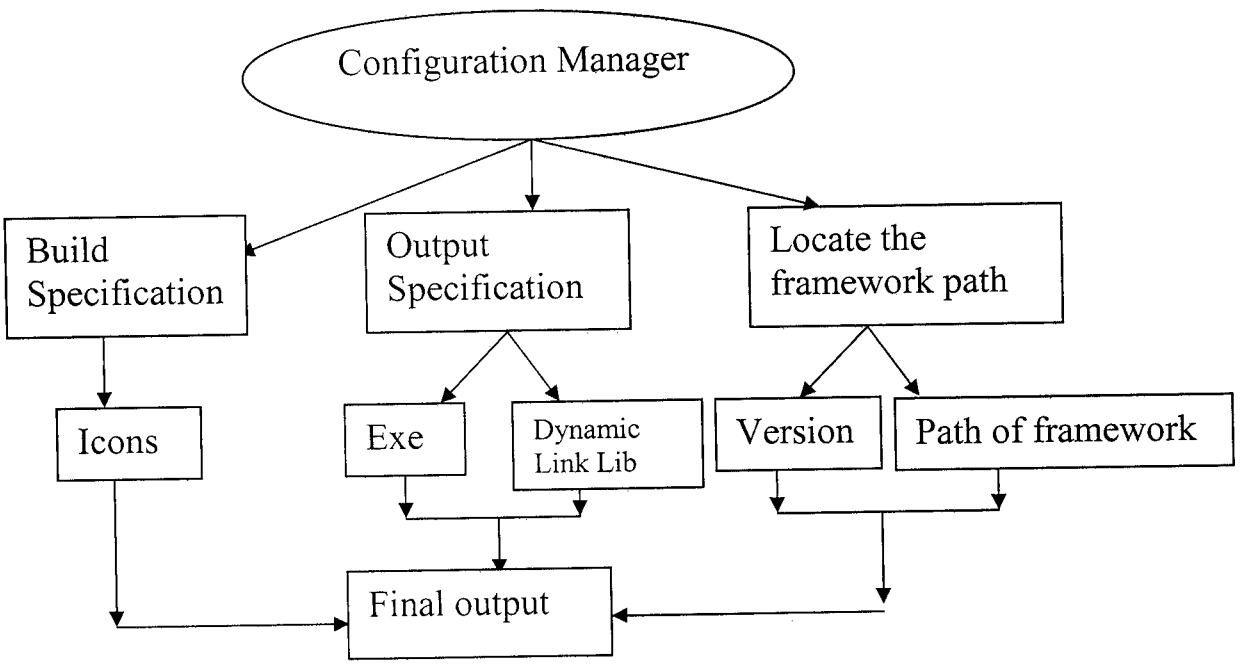

Fig: 3.1.1.4 Configuration Manager Module

3.2 CONCURRENT PROCESS DECOMPOSITION

The light weight development environment for all version of .net looks like a visual studio environment. The user using this environment can develop the CONSOLE, WINDOW, DLL applications concurrently. The concurrent process deals with different types of project.

We can compile and run more number of projects in different languages. Moreover the compiler acts in different way for different languages or different platforms.

The light weight development environment for all version of .net aims in reducing the memory and there is no need to install, just copy and paste in the system and users can work in this environment.

3.3 DETAILED DESIGN

3.3.1 Application Selection and New Project Creation Module

If the users are working on an existing project which already allocated on the machine, he can skip this step. If the user wants to create a new project, he can use **Ctrl-Shift+N** or click **New Project Icon**.

Basically **Build Project** performs the compilation and loading over all the files associated with the current project, while **Build Solution** performs the compilation and loading on not only the files associated with the current project, but also other projects even with the file using.

The Build menu gives either **Build Solution** or **Build First Project** submenus (the name of current project) give the following successful result.

Run or Execute Program

Now you can run (execute) your code using **Debug** menu. There are four ways to run the program. Enter any key to close the result window. If user selects **Start** submenu (F5), it puts up the Execution Window only while the program is actually running. It will disappear once the program completes, so user can not check the results statically displayed on the window. In order to check the execute status, you need to set a breakpoint. Now we can close the project by clicking the **Close Solution** submenu in **File Menu** shown.

Open Existing Project

If the users have an already developed project which has been already developed and he would like to load the current MVS.NET, you can select **Project...** Menu Item in **Open** submenu of **File Menu**.

It prompts **Open Project Window** which allows users to find the existing project file, named as Project Name with extension .csproj. For example, we can search the location of our First Project saved and find the project file called FirstProject.csproj, shown in Now you have the project with a C# file called Class1.cs associated with the project. It should be noted that the file name of the C# may be different from the class defined in the file. In this case, we define a class called First Hello. This is different from Java, in which the file name has to be identical to the name of class.

View Opened Project (solution) using Solution Explorer

`After you open an existing project (solution) you can view the contents of the project by click **View Menu** and **Solution Explorer** (Ctrl_Alt-L). For example, you can add new C# code by selecting **Add New Item** (Ctrl+Shift+A) which gives a new window (Figure 20). Add New Item Submenu.. Add New Item Window which has many options of item types. Let's select Text File and save the file name as FirstHello.cs, which will be saved and added into the current project, shown in Solution Explorer Window.

3.3.2: Solution Explorer and Reference Manager Module

Solution Explorer window shows which files make up the project you are working with and allows you to open any of them by double-clicking on the item. The Solution Explorer supports a variety of management commands (item dependant). For any specified item, the project type determines which management commands are available. For example, if you select a file in a Project, you can use the Delete command to delete the file permanently. To access the management commands, select an item in the Solution Explorer and right-click on it. If you right-click on a '.cs' file (not a form or a control), you get a menu containing nine items.

3.3.3 Add New Items and Class Designer Module

To add an existing item to a project

1. In **Solution Explorer**, select a target project.
2. On the **Project** menu, select **Add Existing Item**.
3. In the **Add Existing** dialog box, locate and select the project item you want to add.
4. Choose **Open** and view the item in the default editor.

3.3.5: Configuration Manager Module

Configuration management (CM) is a technical management model that focuses on establishing and maintaining consistency of a product's performance and its functional and physical attributes with its requirements, design, and operational information throughout its life. For information assurance, CM can be defined as the management of security features and assurances through control of changes made to hardware, software, firmware, documentation, test, test fixtures, and test documentation throughout the life cycle of an information system.

Configuration management is widely used by many military organizations to manage the technical aspects of any complex systems, such as weapon systems, vehicles, and information systems. The discipline combines the capability aspects that these systems provide an organization with the issues of management of change to these systems over time. Outside of the military, CM is equally appropriate to a wide range of fields and industry and commercial sector

CHAPTER -4
TESTING AND TEST PLAN

Testing is executing a program to test the logical changes needed in it and with intention of finding errors. Tests are conducted to find discrepancies between the new system and its original objective, current specifications and documents.

4.1 SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

4.1.1 TYPES OF TESTS

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program input produces valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

As mentioned above each and every unit of the project are tested using the proper testing approach. Starting with the New project dialog to the Project configuration every units were tested successfully.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

- Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.
- The overall usage of the software has been tested. It gives a successful output at the end of the test. Every unit shows its own output and the total aim of the software has been attained successfully.
- The sequence of project creation has been tested.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

The user or the programmer knows exactly what happens inside the software and what are the mechanisms that the Light Weight Development Environment uses to develop an application.

CHAPTER – 5
IMPLEMENTATI0N AND RESULT

Implementation literally means to put into effect or to carry out. The system implementation phase of the software deals with the translation of the design specification into the source code. Coding is done in this stage using a .net framework and programming language.

5.1 IMPLEMENTATION

The light weight development environment for all version of .net is developed in c#.net and implemented in a system that has any versions of .net framework. This is project looks like a visual studio environment. In this environment we didn't add the web services.

In this environment we have added some of the applications. Which are very useful to develop for the WINDOW, CONSOLE, CLASSLIBRARY projects.

This system has been implemented successfully in a machine and tested running.

5.2 RESULT:

The light weight development environment is successfully created by satisfying the needs of a programmer to make an application that needs a very simple development environment.

CHAPTER – 6
CONCLUSION AND EXTENSION OF WORK

6.1 CONCLUSION

This application "LIGHT WEIGHT DEVELOPMENT ENVIRONMENT FOR ALL VERSION OF .NET" project was successfully completed within the specified time. This application is tested and implemented to the satisfaction of the users. Working in real time development environment was an excellent experience and opened new avenues of interest and proved to be a very enriching experience.

6.2 EXTENSION OF WORK

In future we can also implement the web services to create effective web pages and some graphic managed tool boxes. This will enhance the system in a better manner.

APPENDIX

SCREEN SHOTS

Project Starting Form:

This screen shot has been taken while creating a new project.

The dialog box which is used select the type of project and language. The output shown in figure A1.

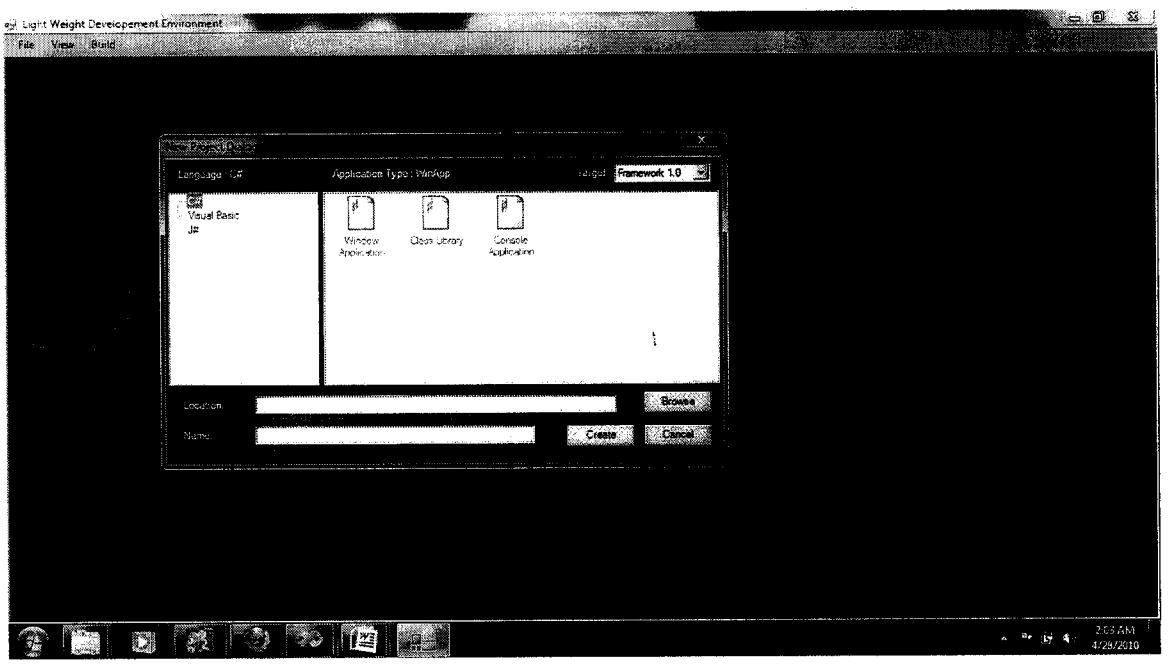

Fig A1: Application Selection and New Project Creation Module Form

LWDE Starting Window:

The starting page or the Opening page of our Light Weight development Environment. The out put shown in figure A2.

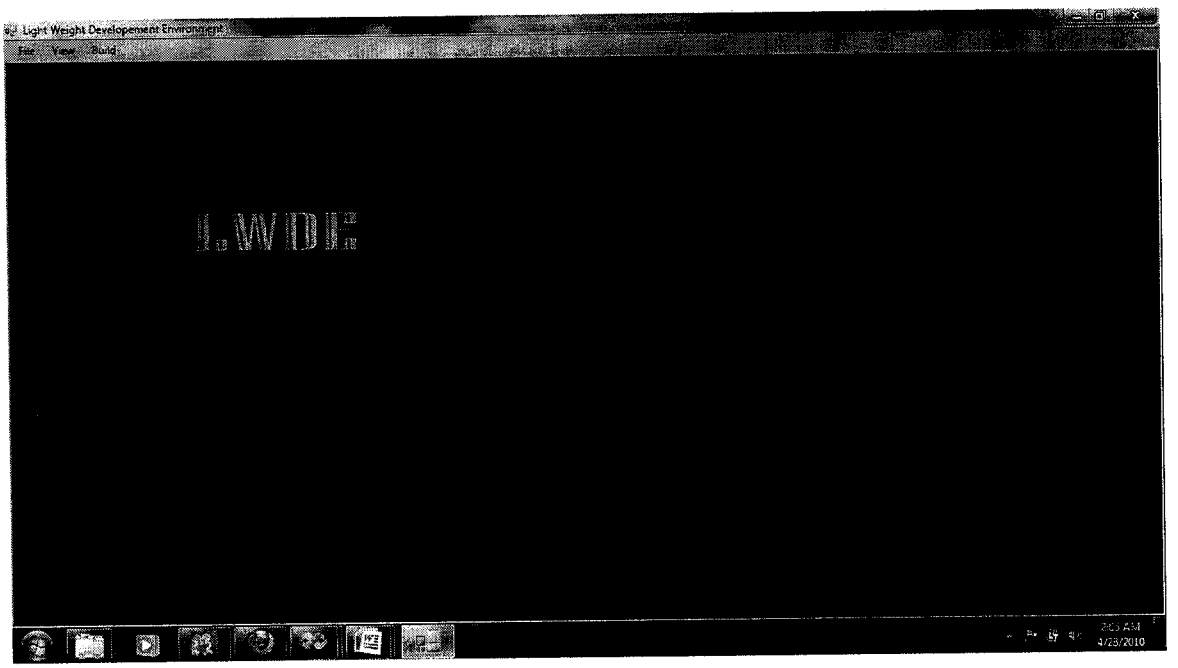

Fig A2: Application Environment

Starting page with Solution Explorer:

The solution explorer placed at the right end of the LWDE shows the project files created at the location of the project. The out put shown in figure A3.

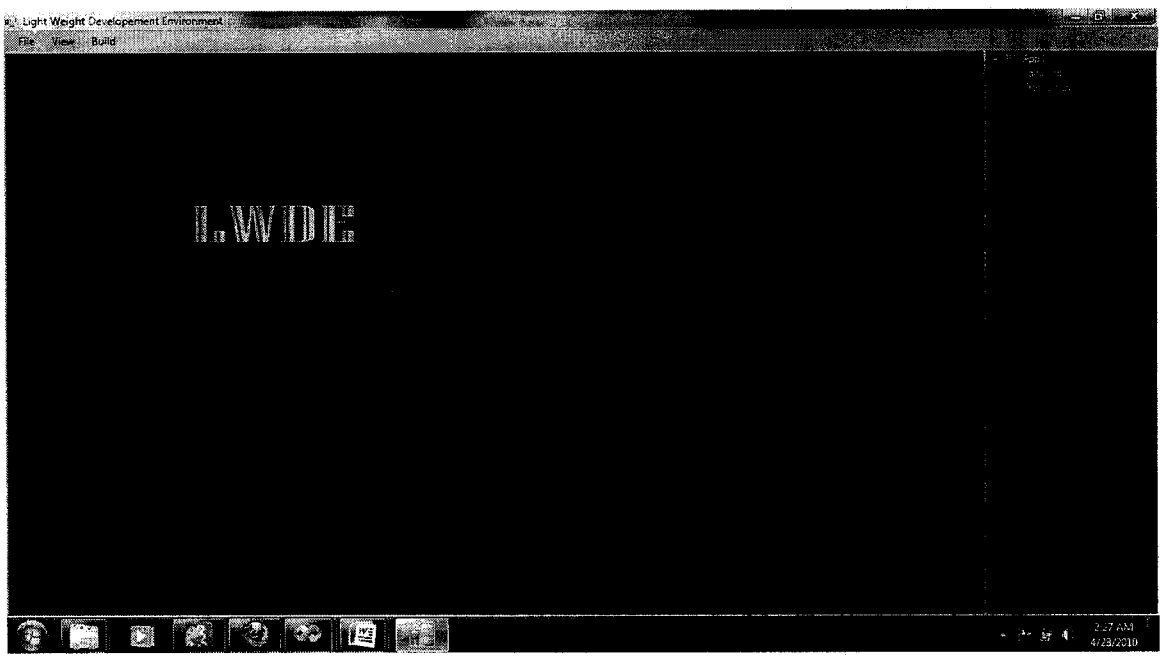

Fig A3. In main window while Solution Explorer

Selecting Add New Item Window:

When we select a new item that will show a new item dialog box. This will add the selected file to the solution explorer. The out put shown in figure A4.

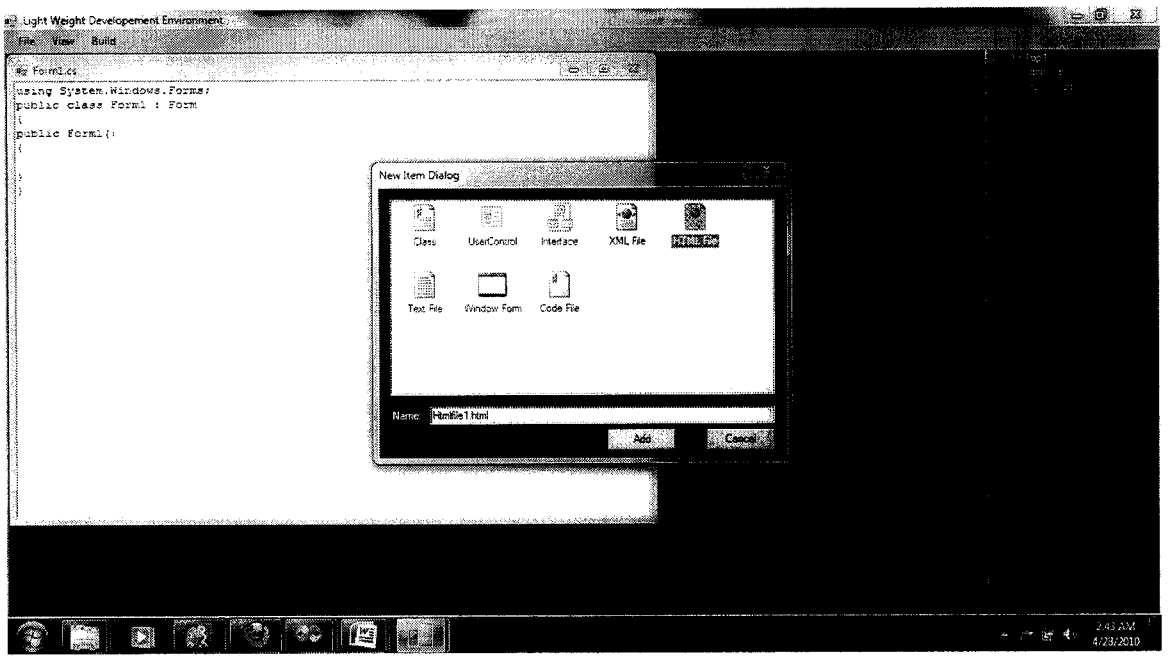

Fig A4. In main window while clicking the Add New Item application window

Adding a new HTML File to project:

The below picture shows the editor that have an HTML file.

We can crate our own code to this newly added file. The out put shown in figure A5.

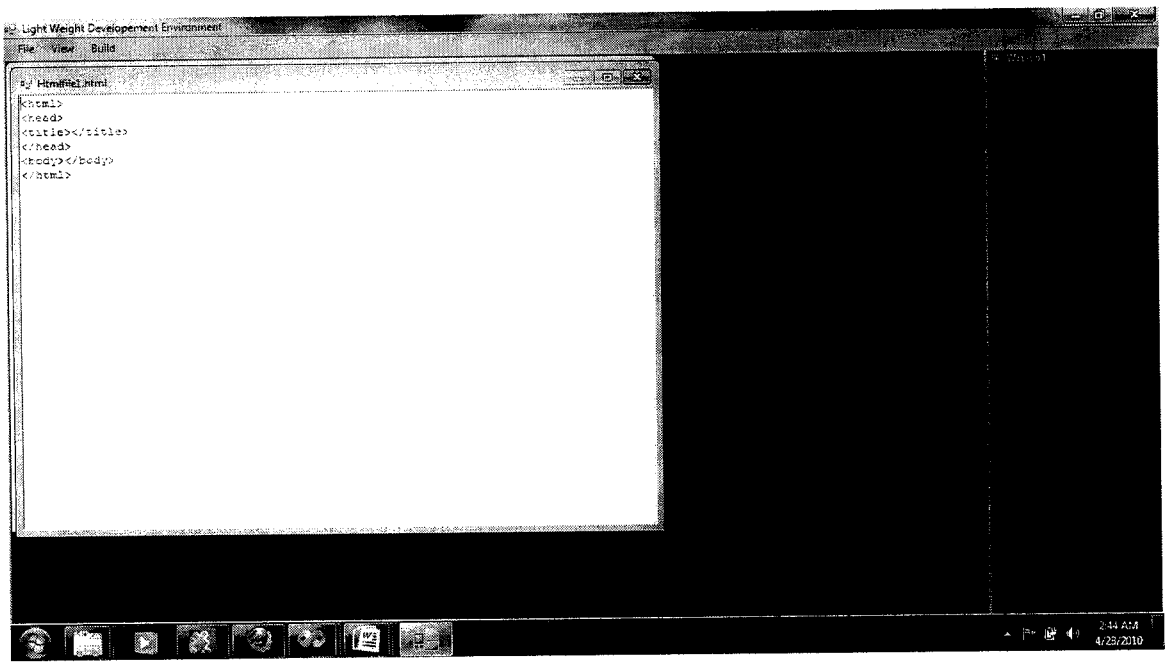

Fig A5. In main window while clicking the Add New Item HTML File

Configuration Manager:

The project type (Windows,Console,class library) ,Version of .net frame work and the language shold be selected from this configuration manager. The out put shown in figure A6.

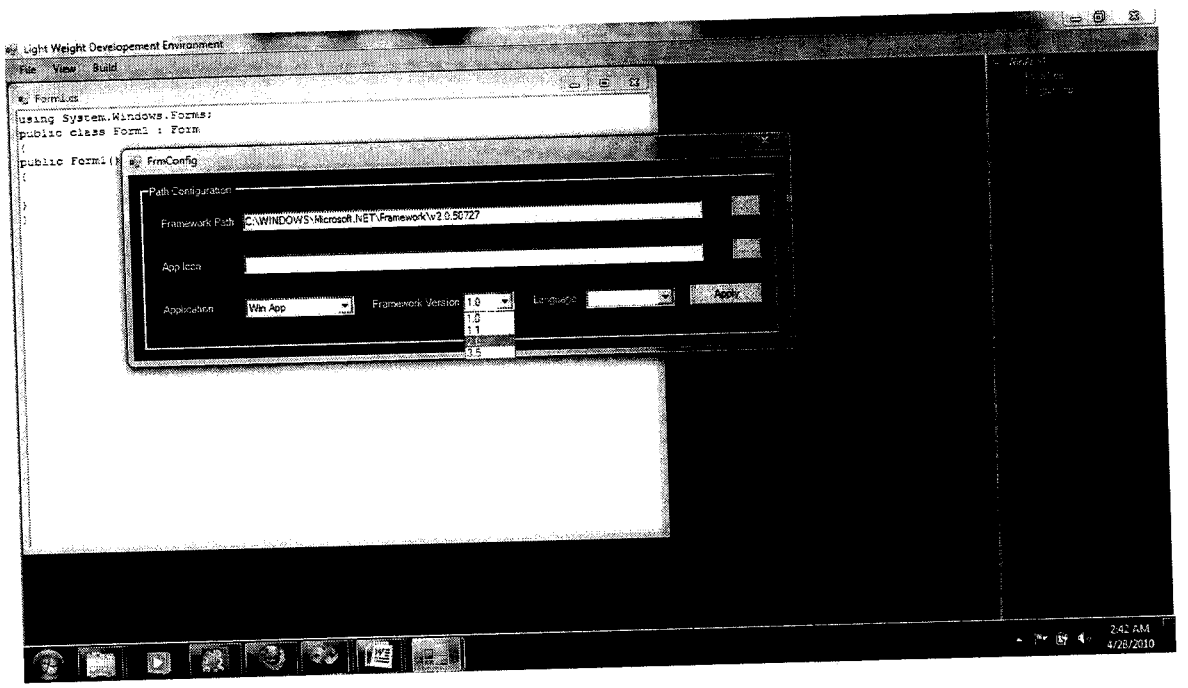

Fig A6. In main window while clicking the configuration manager dialog box

Compiling Error:

If the files in the project have any error in code they have then the error message will be shown while compiling the project. The out put shown in figure A7.

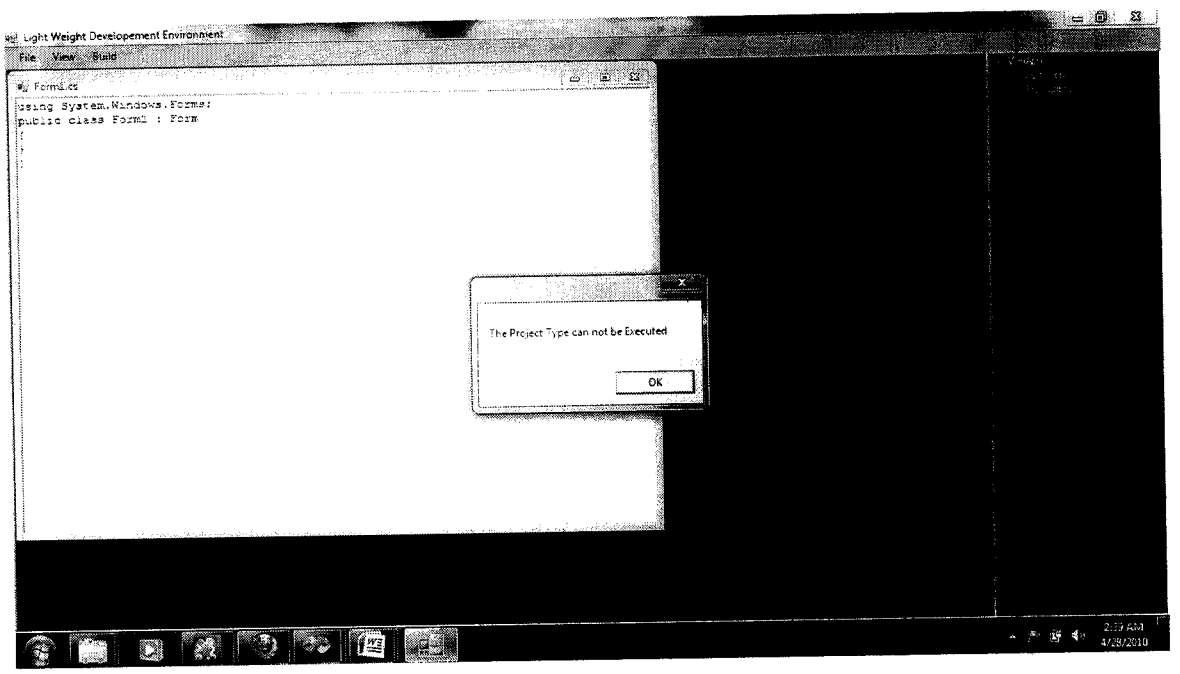

Fig A7. In main window with out project Selection Running Error dialog box

Ready For Compile:

If the files in the project don't have any errors in code they have then the message shown below confirms that they are ready to execute.

The out put shown in figure A8.

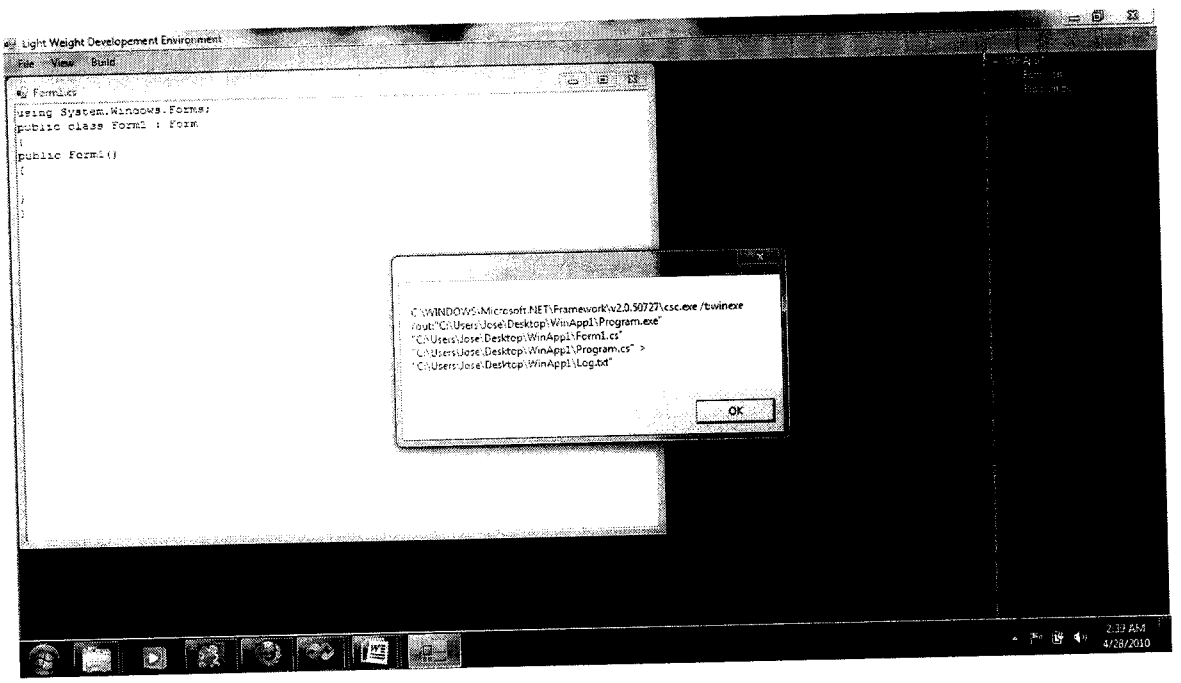

Fig A8. Set the path for compile

REFERENCES

Books:

1. Andrew Troelsen, "C# 2005 And .Net 2.0 Platform", third Edition (Hardcore)
2. Microsoft, MSDN 2005
3. Jesse Liberty, Dan Hurwitz,**"Programming ASP.NET"**
4. Laurence Moroney, John Grieb, Robin Pars," **Foundations of ASP.NET AJAX "**

Web-Reference:

www.msdn.microsoft.com

www.w3schools.com