Chapter 3

Requirements

3.1 Overview

This project is aimed at providing organized and scheduled usage of time. However, like any other GUI, it is also having some requirement constraints with respect to the hardware and software of the client's system. For optimum performance, it is recommended that the client must meet the following specifications for best use.

3.2 System Requirements

To be used efficiently, all computer software needs certain hardware components or other software resources to be present on a computer. These prerequisites are known as system requirements and are often used as a guideline as opposed to an absolute rule.

3.2.1 Hardware Requirements

Computer hardware specifications are technical descriptions of the computer's components and capabilities.

- Processor: Pentium IV 3.0 or above.
- Ram: At least 512 MB or above.

3.2.2 Software Requirements

A software requirement is a description of a software system to be developed. It lays out functional and non-functional requirements, and may include a set of use cases that describe user interactions that the software must provide.

- Operating System: Any version of Windows operating system above Windows 95 and above
- Microsoft Visual Studio 2010 Express

3.3 Software Packages

Software package may refer to individual files or resources which are packed together as a software collection that provides certain functionality as part of a larger system. This section gives a brief description of the software packages used in the development of our project.

3.3.1 Visual studio 2010

Visual Studio is a complete set of development tools for building ASP.NET Web applications, XML Web Services, desktop applications, and mobile applications. Visual Basic, Visual C#, and Visual C++ all use the same integrated development environment (IDE), which enables tool sharing and eases the creation of mixed-language solutions. In addition, these languages use the functionality of the .NET Framework, which provides access to key technologies that simplify the development of ASP Web applications and XML Web Services.

3.3.2 C#

C# (pronounced as see sharp) is a language encompassing strong typing, imperative, declarative, functional, generic, object-oriented (class-based), and component-oriented programming disciplines. It was developed by Microsoft within its .NET initiative and later approved as a standard by Ecma (ECMA-334) and ISO (ISO/IEC 23270:2006). C# is one of the programming languages designed for the Common Language Infrastructure.

3.4 Creating Windows-Based Applications:

We can create traditional Microsoft Windows-based applications and client/server applications by utilizing the designers in Visual Studio. This topic contains links to topics that describe the different types of Windows applications that you can create with Visual Studio.

3.5 Windows Forms:

Provides links to topics about the technologies and tools for creating Windows-based applications. As forms are the base unit of your application, it is essential that we give some thought to their function and design. A form is ultimately a blank slate that we, as a developer, enhance with controls to create a user interface and with code to manipulate data. To that end, Visual Studio provides us with an integrated development environment (IDE) to aid in writing code, as well as a rich control set written with the .NET Framework.