BACKUP AND RECOVERY SCHEDULER AUTOMATION MANAGEMENT

By

M.ARUNKUMAR
(Registration Number: 71205621006)

Of

KUMARAGURU COLLEGE OF TECHNOLOGY
COIMBATORE

A PROJECT REPORT
Submitted to the

FACULTY OF INFORMATION AND COMMUNICATION ENGINEERING

In partial fulfillment of the requirements
For the award of the degree

Of

MASTER OF COMPUTER APPLICATION

JUNE , 2008

KUMARAGURU COLLEGE OF TECHNOLOGY

Coimbatore-641006

DEPARTMENT OF COMPUTER APPLICATION

Bonafide Certificate

Certified that this project report titled **BACKUP AND RECOVERY SCHEDULER AUTOMATION MANAGEMENT** is the bonafide work of **Mr.M.ARUNKUMAR (Registration Number: 71205621006)** who carried out the research under my supervision. Certified further, that to the best of my Knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

25/6/08

Supervisor

Head of the Department

Submitted to Project and Viva Examination held on 01/07/2008

Internal Examiner

External Examiner

STS

Soft Tech Solutions

To Whomsoever It May Concern

This is to certify that Mr.ARUNKUMAR.M (71205621006), Student of Kumaraguru College of Technology, Coimbatore doing his Final Year, M.C.A (Computer Applications) has successfully completed his project entitled as "BACKUP AND RECOVERY SCHEDULER AUTOMATION MANAGEMENT" under the guidance of Mr.Tamilselvan.R, Senior Software Engineer, from December 2007 to May 2008.

During his Project duration his conduct and contribution has been excellent.

We wish all the best for his future Endeavors.

For Soft Tech Solutions

(Project Co-ordinator)

SOFT TECH SOLUTIONS
No-23,
MAHALAKSHMI St,
T.NAGAR,
Ph-044-42604210
CHENNAI - 17

ABSTRACT

The project "**BACKUP AND RECOVERY SCHEDULER AUTOMATION MANAGEMENT**" is designed using Microsoft Visual Studio.Net 2003 as front end and Microsoft SQL Server 2000, MS Access as backend which works in .Net framework version 1.1. The coding language used is VB .Net.

This System is designed for the purpose of backup and restoring the vital file source in a particular directory i.e. the databases of the company.

This project is mainly used to secure the important files of the particular Organization. In first to select the backup files and enter the backup path after complete the process to select any schedule types like Daily, Weekly, and Monthly. The time for backup process, Maximum capacity of the backup is common for all schedules.

When we need for a backup file or the folder, which is backup, select the needed file or folder and click restore button. The selected file will be restored the actual path.

Also the system provides the facility to view the backup files and folders. The log file will provide the schedule information, starting & ending time of the application. The total backup area size can be set and when it is exceeded, it will be intimated to the application. The backup area size can be customized at any time.

ACKNOWLEDGEMENT

I would like to take this opportunity to say thank you to the people who have helped to make this project.

First I would like to express my heartfelt thanks to our principal **Dr.Joseph V.Thanikal B.E., M.E., Ph.D., PDF, CEPIT,** for having given me the opportunity to do this project.

I express gratitude and thanks to **Dr.S.Thangasamy Ph.D., Dean,** Computer Science, for providing moral support towards this project work.

I express my deep sense of gratitude and profound thanks to **Dr.M.Gururajan Ph.D., HOD,** Computer Application. for providing moral support towards this project work.

I tender my special thanks to **Mr.A.Muthu Kumar M.C.A.,M.Phil.,** Assistant Professor and Project Coordinator, Department of Computer Applications for his constant encouragement.

I sincerely express my humble gratitude to **Miss.R.Veena Shree M.C.A., M.Phil,** Lecturer. Department of Computer Application. who has been my guide with valuable and holistic suggestions and extended kind of operation and encouragement to make this project.

I also express my deep sense of gratitude and profound thanks to **Mr.R.TamilSelvan Senior Software Engineer, Soft Tech Solutions,** who has been my external guide and for giving valuable suggestions and encouragement.

I acknowledge my hearty thanks to all my beloved friends on their valuable co-operation in the proceeding of my work.

I wish to credit my special accordance to my parents for their encouragement and prayers to have a successful project.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 ABOUT THE ORGANISATION

COMPANY PROFILE

Soft Tech Solutions - STS - is a technology powerhouse with proven experience and capability to deliver business results to clients. We are pioneered to generate novel concepts through our innovation to deliver unparalleled quality from the year of 2000. Our people solve complex and difficult problems that are critical to the success—and even the survival—of our clients. We use best-in-class components and apply technology at scale and in sophisticated production environments. We provide end-to-end business solutions that leverage technologies and solutions for dynamic environment to harmonize business and technology.

SERVICES

Conventional outsourcing can decrease your costs. but it can't increase your performance. So we are using contemporary outsourcing practices. which can outperform your competition. and yourselves. The time is right to explore new ways to achieve more values and our approach takes application development and maintenance to the next level.

- Application Development
- Wireless Applications
- Enterprise Resource Planning
- Portals. content and E-commerce
- Security and privacy services

Our BPO provides continually asses new opportunities in industry-specific as well as cross industry solutions that enable clients to extend the core competencies. achieve operational excellence and outperform the competition.

- Automotive
- Banking
- Insurance
- Retail
- Manufacturing
- Healthcare and life sciences
- Logistics
- Energy
- Media and Entertainment
- Governmental and Public Institutions
- Educational sectors
- Travel
- Aerospace and Defense

RESEARCH AND INSIGHTS:

Our R&D team expresses the state of an industry. the degree of competition and the lure of progress to make radical and incremental changes to our products and services. Our single minded passion always focuses on future-oriented search. We always identify the technology. tools and methodologies which are most appropriate to solve specific customer challenges. We work closely with our clients to develop proofs and models of modules to show as a model. The R&D team also regularly develops re-usable frameworks and components to solve generic industry problem.

1.2 PROJECT DESCRIPTION

The software is divided into two projects from which the first one is for altering the options. The other will be added in the windows schedule so that the application is getting automatically started when the user logins to the windows. The backup application is minimized and running as a task in the operating system.

The project titled "**Backup and Recovery Scheduler**" contains 4 modules.

1. Schedule selection
2. Path selection
3. Backup module
4. Recovery module
5. Reports

1. Schedule selection

The backup schedule may be either daily backup. weekly or monthly backup. The date or day. time. AM or PM should be selected. In addition. the maximum number of copies should be selected. At any time. the schedule can be changed.

2. Path selection

The source path i.e.. which folder is to be taken for backup is selected. In addition. backup path i.e., where the folder will be saved for backup should be selected from here. The total folder can be selected for backup.

3. Backup module:

In this module, the backup application starts running from the system logon time. First, the application checks whether today is scheduled and the backup is already being taken. If backup is already taken, the application continues running without any additional work. If no, it checks for the maximum copies. If the folders present in the backup folder are equal to the maximum, then the first created folder is deleted and the backup takes place. Otherwise, the backup takes place without any deletion.

4. Recovery module:

First, we can view all the backup folders in the backup path. Then we should select any folder and click recovery to copy the selected folder to the original source path.

5. Reports:

Schedule details and Paths can be viewed. The backup folder details also can be viewed. The log file is used to retrieve the log details of the scheduling operation and the resource path.

CHAPTER 2
SYSTEM STUDY

2.1 EXISTING SYSTEM

The backups are taken manually in the existing system. In the system. an operator is required to take backup. Moreover. the proper schedule cannot be maintained. In addition, it requires human invention and disturbing the normal office operations. Unauthorized persons may take backup and the schedule cannot be maintained well. To avoid such circumstances, a system is required to automate the backup task.

Drawbacks of existing system:

1. The record of backup time is not maintained.
2. At each and every time. checking for backup is not possible.
3. The operator may forget to take backup.
4. Path can be changed when taking backup.

2.2 PROPOSED SYSTEM

To avoid the drawbacks in the existing system, the proposed system is designed. The proposed system is a computerized one. In this system, everything is done automatically. We need to change the schedule only. In addition. the records of backup can be maintained with no effort. The recovery option is also provided and any version of the backup can be taken easily.

The advantages of proposed system are:

1. Automatic backup management.
2. Easy schedule selection.
3. Reports can be viewed any time.
4. Easy recovery.
5. More reliability than existing system.

2.3 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis. some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

2.3.1 ECONOMICAL FEASIBILITY:

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

2.3.2 TECHNICAL FEASIBILITY:

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

2.3.3 SOCIAL FEASIBILITY:

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

CHAPTER 3
DEVELOPMENT ENVIRONMENT

3.1 HARDWARE REQUIREMENTS

The hardware used for the development of the project is:

PROCESSOR	:	PENTIUM IV
RAM	:	1 GB
MONITOR	:	15" COLOR
HARD DISK	:	120 GB
FLOPPY DRIVE	:	1.44 MB
CDDRIVE	:	LG 52X
KEYBOARD	:	MULTIMEDIA
MOUSE	:	LOGITECH

3.2 SOFTWARE REQUIREMENTS

The software used for the development of the project is:

OPERATING SYSTEM	:	Windows XP Professional
ENVIRONMENT	:	Visual Studio .NET 2003
.NET FRAMEWORK	:	Version 1.0
LANGUAGE	:	Visual Basic.NET
BACKEND	:	MS SQL SERVER 2000. MS ACCESS

3.3 PROGRAMMING ENVIRONMENT

FEATURES OF VISUAL BASIC .NET

THE .NET FRAMEWORK

The .NET Framework is a new computing platform that simplifies application development in the highly distributed environment of the Internet.

OBJECTIVES OF .NET FRAMEWORK:

1. To provide a consistent object-oriented programming environment whether object codes is stored and executed locally on Internet-distributed, or executed remotely.

2. To provide a code-execution environment to minimizes software deployment and guarantees safe execution of code.

3. Eliminates the performance problems.

There are different types of application, such as Windows-based applications and Web-based applications. To make communication on distributed environment to ensure that code be accessed by the .NET Framework can integrate with any other code.

COMPONENTS OF . NET FRAMEWORK

THE COMMON LANGUAGE RUNTIME (CLR):

The common language runtime is the foundation of the .NET Framework. It manages code at execution time, providing important services such as memory management, thread management, and remoting and also ensures more security and robustness. The concept of code management is a fundamental principle of the runtime. Code that targets the runtime is known as managed code, while code that does not target the runtime is known as unmanaged code.

THE .NET FRAME WORK CLASS LIBRARY:

It is a comprehensive, object-oriented collection of reusable types used to develop applications ranging from traditional command-line or graphical user interface (GUI) applications to applications based on the latest innovations provided by ASP.NET, such as Web Forms and XML Web services.

The .NET Framework can be hosted by unmanaged components that load the common language runtime into their processes and initiate the execution of managed code, thereby creating a software environment that can exploit both managed and unmanaged features. The .NET Framework not only provides several runtime hosts, but also supports the development of third-party runtime hosts.

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 are all run on CLR.

- Security.
- Robustness.
- Productivity.
- Performance.

SECURITY:

The runtime enforces code access security. The security features of the runtime thus enable legitimate Internet-deployed software to be exceptionally feature rich. With regards to security, managed components are awarded varying degrees of trust, depending on a number of factors that include their origin to perform file-access operations, registry-access operations, or other sensitive functions.

ROBUSTNESS:

The runtime also enforces code robustness by implementing a strict type- and code-verification infrastructure called the common type system (CTS). The CTS ensures that all managed code is self-describing. The managed environment of the runtime eliminates many common software issues.

PRODUCTIVITY:

The runtime also accelerates developer productivity. For example, programmers can write applications in their development language of choice, yet take full advantage of the runtime, the class library, and components written in other languages by other developers.

PERFORMANCE:

The runtime is designed to enhance performance. Although the common language runtime provides many standard runtime services, managed code is never interpreted. A feature called just-in-time (JIT) compiling enables all managed code to run in the native machine language of the system on which it is executing. Finally, the runtime can be hosted by high-performance, server-side applications, such as Microsoft® SQL Server™ and Internet Information Services (IIS).

.NET:

When .NET was announced in late 1999, Microsoft positioned the technology as a platform for building and consuming Extensible Markup Language (XML) Web services. XML Web services allow any type of application, be it a Windows or browser-based application running on any type of computer system, to consume data from any type of server over the Internet. The reason this idea is so great is the way in which the

XML messages are transferred over established standard protocols that exist today. Using protocols such as SOAP. HTTP. and SMTP. XML Web services make it possible to expose data over the wire with little or no modifications to your existing code.

Figure 1.1 presents a high-level overview of the .NET Framework and how XML Web services are positioned.

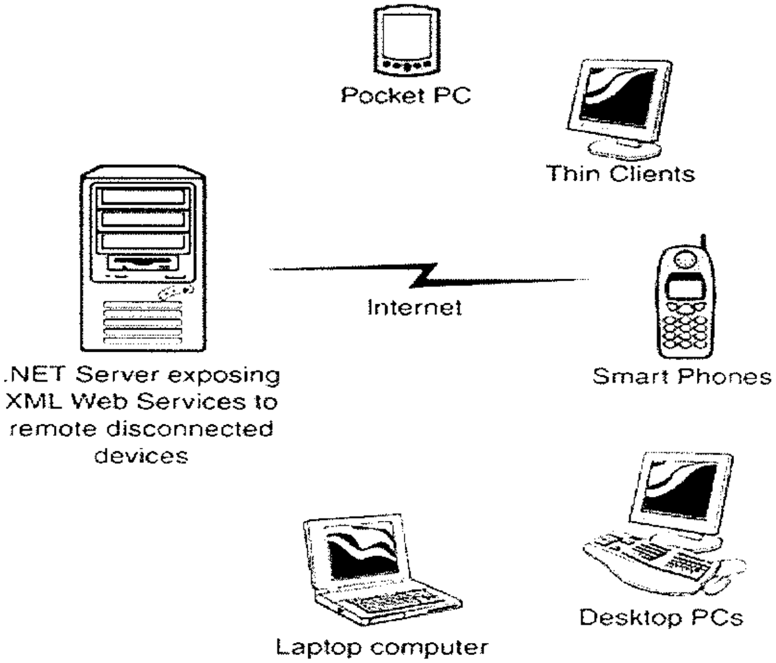
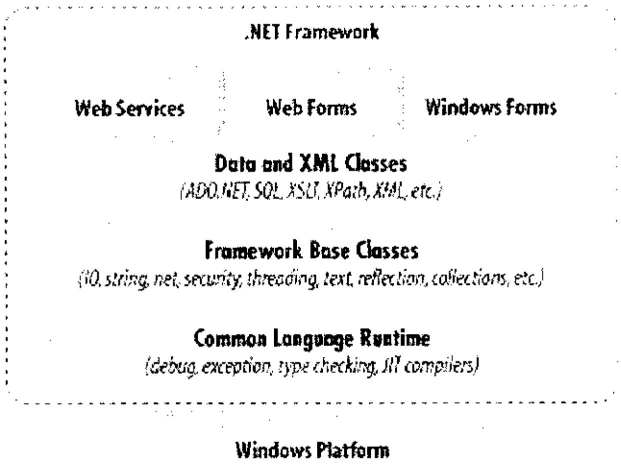

Pocket PC
Thin Clients
Internet
.NET Server exposing XML Web Services to remote disconnected devices
Smart Phones
Desktop PCs
Laptop computer

Figure 1.1 Stateless XML Web services model.

.NET Framework

Now that you are familiar with the major goals of the .NET Framework. let's briefly examine its architecture. As you can see in Figure 1-2. the .NET Framework sits on top of the operating system. which can be a few different flavors of Windows and consists of a number of components .NET is essentially a system application that runs on Windows.


.NET Framework
Web Services
Web Forms
Windows Forms
Data and XML Classes
(ADO.NET, SQL, XSLT, XPath, XML, etc.)
Framework Base Classes
(IO, string, net, security, threading, text, reflection, collections, etc.)
Common Language Runtime
(debug, exception, type checking, JIT compilers)
Windows Platform

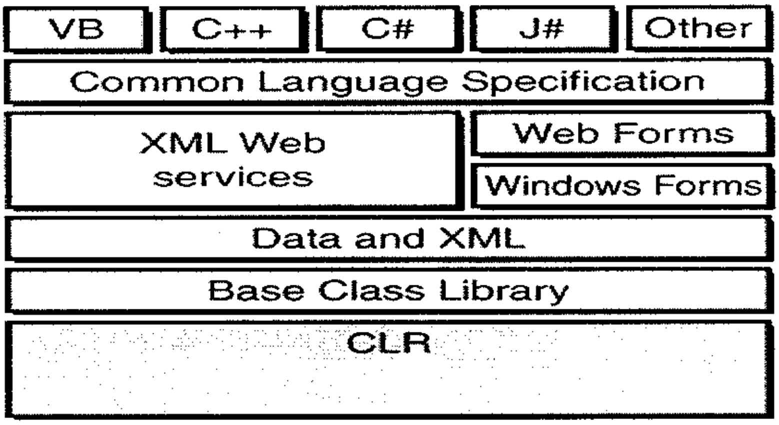
Conceptually. the CLR and the JVM are similar in that they are both runtime infrastructures that abstract the underlying platform differences. However. while the JVM officially supports only the Java language. the CLR supports any language that can be represented in its Common Intermediate Language (CIL).

The Common Language Runtime

At the heart of the .NET Framework is the common language runtime. The common language runtime is responsible for providing the execution environment that code written in a .NET language runs under. The common language runtime can be compared to the Visual Basic 6 runtime. except that the common language runtime is designed to handle all .NET languages. not just one. as the Visual Basic 6 runtime did for Visual Basic 6. The following list describes some of the benefits the common language runtime gives you:

- Cross-language debugging
- Cross-language exception handling
- Full support for component versioning
- Access to legacy COM components
- XCOPY deployment
- Robust security model
- Automatic memory management

Figure1.3 The common language runtime and the .NET Framework.



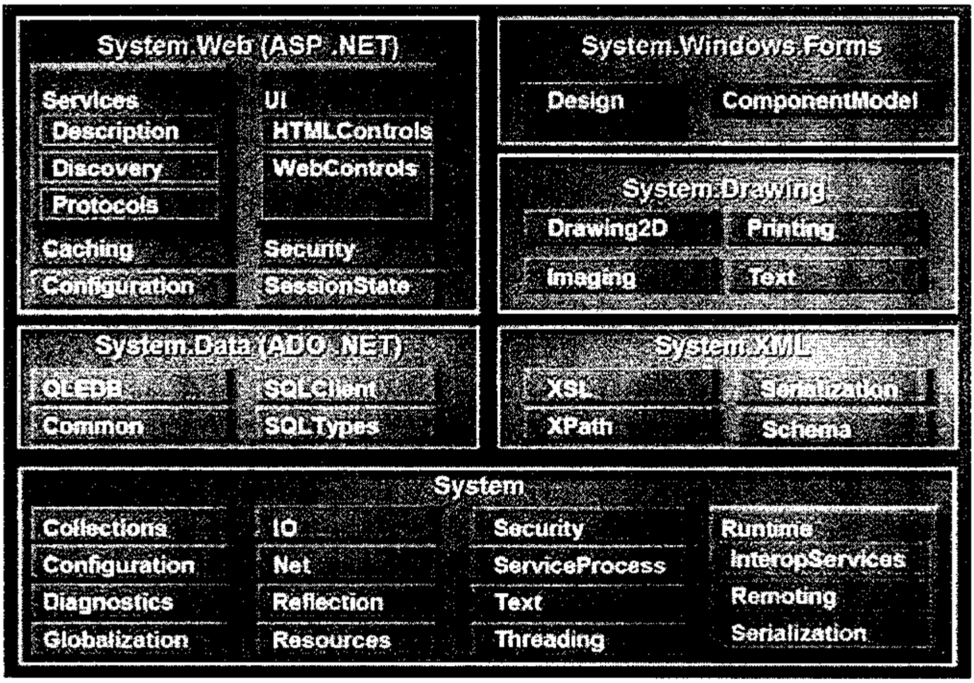
The .NET Framework Class Library

The second most important piece of the .NET Framework is the .NET Framework class library (FCL). As you've seen. the common language runtime handles the dirty work of actually running the code you write. But to write the code. you need a foundation of available classes to access the resources of the operating system. database server. or file server. classes. structures. interfaces. enumerations. and delegates that give you access to these resources.

The namespaces are logically defined by functionality. For example. the System. Data namespace contains all the functionality available to accessing databases. This namespace is further broken down into System.Data.SqlClient. which exposes functionality specific to SQL Server. and System.Data.OleDb. which exposes specific functionality for accessing OLEDB data sources. The bounds of a namespace aren't necessarily defined by specific assemblies within the FCL: rather. they're focused on functionality and logical grouping.

To use an FCL class in your application. you use the `Imports` statement in Visual Basic .NET or the `using` statement in C#. When you reference a namespace in Visual Basic .NET or C#. you also get the convenience of auto-complete and auto-list members when you access the objects types using Visual Studio .NET. This makes it very easy to determine what types are available for each class in the namespace you're using. As you'll see over the next several weeks. it's very easy to start coding in Visual Studio .NET.

Figure 1.6. The .NET Framework class library.



The Structure of a .NET Application

To understand how the common language runtime manages code execution. you must examine the structure of a .NET application. The primary unit of a .NET application is the assembly. An assembly is a self-describing collection of code. resources. and metadata. The assembly manifest contains information about what is contained within the assembly. The assembly manifest provides:

- Identity information. such as the assembly's name and version number
- A list of all types exposed by the assembly
- A list of other assemblies required by the assembly
- A list of code access security instructions, including permissions required by the assembly and permissions to be denied the assembly

An assembly contains one or more modules. A module contains the code that makes up your application or library. and it contains metadata that describes that code. When you compile a project into an assembly, your code is converted from high-level code to IL. Because all managed code is first converted to IL code. applications written in different languages can easily interact. For example. one developer might write an application in Visual C# that accesses a DLL in Visual Basic .NET. Both resources will be converted to IL modules before being executed. thus avoiding any language-incompatibility issues.

Each module also contains a number of types. Types are templates that describe a set of data encapsulation and functionality. There are two kinds of types: reference types (classes) and value types (structures). These types are discussed in greater detail in Lesson 2 of this chapter. Each type is described to the common language runtime in the assembly manifest. A type can contain fields. properties. and methods. each of which should be related to a common functionality.

It contains fields. properties, and methods related to the functions needed to implement a bank account. A field represents storage of a particular type of data. One field might store the name of an account holder. for example. Properties are similar to fields, but properties usually provide some kind of validation when data is set or retrieved. You might have a property that represents an account balance. When an attempt is made to change the value. the property can check to see if the attempted change is greater than a predetermined limit. If the value is greater than the limit. the property does not allow the change. Methods represent behavior. such as actions taken on data stored within the class or changes to the user interface.

Compilation and Execution of a .NET Application

When you compile a .NET application, it is not compiled to binary machine code: rather, it is converted to IL. This is the form that your deployed application takes—one or more assemblies consisting of executable files and DLL files in IL form. At least one of these assemblies will contain an executable file that has been designated as the entry point for the application.

When execution of your program begins, the first assembly is loaded into memory. At this point, the common language runtime examines the assembly manifest and determines the requirements to run the program. It examines security permissions requested by the assembly and compares them with the system's security policy. If the system's security policy does not allow the requested permissions, the application will not run. If the application passes the system's security policy, the common language runtime executes the code. It creates a process for the application to run in and begins application execution. When execution starts, the first bit of code that needs to be executed is loaded into memory and compiled into native binary code from IL by the common language runtime's Just-In-Time (JIT) compiler. Once compiled, the code is executed and stored in memory as native code. Thus, each portion of code is compiled only once when an application executes.

Namespaces are logical groupings of related classes. The namespaces in the .NET base class library are organized hierarchically. The root of the .NET Framework is the System namespace. Other namespaces can be accessed with the period operator. A typical namespace construction appears as follows:

System

System.Data

System.Data.SQLClient

The first example refers to the System namespace. The second refers to the System.Data namespace. The third example refers to the System.Data.SQLClient namespace. Table 1.1 introduces some of the more commonly used .NET base class namespaces.

Table 1-1. Representative .NET Namespaces

Namespace	Description
System	This namespace is the root for many of the low-level types required by the .NET Framework. It is the root for primitive data types as well, and it is the root for all the other namespaces in the .NET base class library.
System.Collections	This namespace contains classes that represent a variety of different container types, such as Array List, Sorted List, Queue, and Stack. You also can find abstract classes, such as Collection Base, which are useful for implementing your own collection functionality.
System.ComponentModel	This namespace contains classes involved in component creation and containment, such as attributes, type converters, and license providers.
System.Data	This namespace contains classes required for database access and manipulations, as well as additional namespaces used for data access.
System.Data.Common	This namespace contains a set of classes that are shared by the .NET managed data providers.
System.Data.OleDb	This namespace contains classes that make up the managed data provider for OLE DB data access.
System.Data.SQLClient	This namespace contains classes that are optimized for interacting with Microsoft SQL Server.
System.Drawing	This namespace exposes GDI+ functionality and provides

Table 1-1. Representative .NET Namespaces

Namespace	Description
	classes that facilitate graphics rendering.
System.IO	In this namespace. you will find types for handling file system I/O.
System.Math	This namespace is home to common mathematics functions such as extracting roots and trigonometry.
System.Reflection	This namespace provides support for obtaining information and dynamic creation of types at runtime.
System.Security	This namespace is home to types dealing with permissions, cryptography. and code access security.
System.Threading	This namespace contains classes that facilitate the implementation of multithreaded applications.
System.Windows.Forms	This namespace contains types involved in creating standard Windows applications. Classes that represent forms and controls reside here as well.

VB.NET

Visual Basic .NET (VB.NET) is an object oriented computer language that can be viewed as an evolution of Microsoft's visual basic (VB) implemented on the Microsoft.NET framework. Its introduction has been controversial, as significant changes were made that broke backward compatibility with older versions and caused within developer community.

With its release for the .NET platform. the Visual Basic language has undergone dramatic changes.

For example:

- The language itself is now fully object-oriented.
- Applications and components written in Visual Basic .NET have full access to the .NET Framework, an extensive class library that provides system and application services.
- All applications developed using Visual Basic .NET run within a managed runtime environment, the .NET common language runtime.

Visual Basic .NET is the next generation of Visual Basic, but it is also a significant departure from previous generations. Experienced Visual Basic 6 developers will feel comfortable with Visual Basic .NET code and will recognize most of its constructs. However, Microsoft has made some changes to make Visual Basic .NET a better language and an equal player in the .NET world. These include such additions as a Class keyword for defining classes and an Inherits keyword for object inheritance, among others. Visual Basic 6 code can't be compiled by the Visual Basic .NET compiler without significant modification. The good news is that Microsoft has provided a migration tool to handle the task.

ASP.NET

ASP.NET is the next version of Active Server Pages (ASP); it is a unified Web development platform that provides the services necessary for developers to build enterprise-class Web applications. While ASP.NET is largely syntax compatible, it also provides a new programming model and infrastructure for more secure, scalable, and stable applications.

ASP.NET is a compiled, NET-based environment, we can author applications in any .NET compatible language, including Visual Basic .NET, C#, and JScript .NET. Additionally, the entire .NET Framework is available to any ASP.NET application. Developers can easily access the benefits of these technologies, which include the managed common language runtime environment (CLR), type safety, inheritance, and so on.

ASP.NET has been designed to work seamlessly with WYSIWYG HTML editors and other programming tools, including Microsoft Visual Studio .NET. Not only does this make Web development easier, but it also provides all the benefits that these tools have to offer, including a GUI that developers can use to drop server controls onto a Web page and fully integrated debugging support.

Developers can choose from the following two features when creating an ASP.NET application. Web Forms and Web services, or combine these in any way they see fit. Each is supported by the same infrastructure that allows you to use authentication schemes, cache frequently used data, or customize your application's configuration, to name only a few possibilities.

An XML Web service provides the means to access server functionality remotely. Using Web services, businesses can expose programmatic interfaces to their data or business logic, which in turn can be obtained and manipulated by client and server applications. XML Web services enable the exchange of data in client-server or server-server scenarios, using standards like HTTP and XML messaging to move data across firewalls. XML Web services are not tied to a particular component technology or object-calling convention. As a result, programs written in any language, using any component model, and running on any operating system can access XML Web services.

ASP.NET takes advantage of performance enhancements found in the .NET Framework and common language runtime. Additionally, it has been designed to offer significant performance improvements over ASP and other Web development platforms. All ASP.NET code is compiled, rather than interpreted, which allows early binding, strong typing, and just-in-time (JIT) compilation to native code, to name only a few of its benefits. ASP.NET is also easily factorable, meaning that developers can remove modules (a session module, for instance) that are not relevant to the application they are developing.

DATA ACCESS WITH ADO.NET

As you develop applications using ADO.NET, you will have different requirements for working with data. You might never need to directly edit an XML file containing data - but it is very useful to understand the data architecture in ADO.NET.

ADO.NET offers several advantages over previous versions of ADO:

- Interoperability
- Maintainability
- Programmability
- Performance Scalability

INTEROPERABILITY:

ADO.NET applications can take advantage of the flexibility and broad acceptance of XML. Because XML is the format for transmitting datasets across the network, any component that can read the XML format can process data. The receiving component need not be an ADO.NET component.

The transmitting component can simply transmit the dataset to its destination without regard to how the receiving component is implemented. The destination component might be a Visual Studio application or any other application implemented with any tool whatsoever.

MAINTAINABILITY:

In the life of a deployed system, modest changes are possible, but substantial. Architectural changes are rarely attempted because they are so difficult. As the performance load on a deployed application server grows, system resources can become scarce and response time or throughput can suffer. Faced with this problem, software architects can choose to divide the server's business-logic processing and user-interface processing onto separate tiers on separate machines.

PERFORMANCE:

ADO.NET datasets offer performance advantages over ADO disconnected record sets. In ADO.NET data-type conversion is not necessary.

SCALABILITY:

ADO.NET accommodates scalability by encouraging programmers to conserve limited resources. Any ADO.NET application employs disconnected access to data; it does not retain database locks or active database connections for long durations.

XML WEB SERVICES

XML Web services are applications that can receive the requested data using XML over HTTP. XML Web services are not tied to a particular component technology or object-calling convention but it can be accessed by any language, component model, or operating system. In Visual Studio .NET, you can quickly create and include XML Web services using Visual Basic, Visual C#, JScript, Managed Extensions for C++, or ATL Server.

XML SUPPORT:

Extensible Markup Language (XML) provides a method for describing structured data. XML is a subset of SGML that is optimized for delivery over the Web. The World Wide Web Consortium (W3C) defines XML standards so that structured data will be uniform and independent of applications. Visual Studio .NET fully supports XML, providing the XML Designer to make it easier to edit XML and create XML schemas.

COMMON LANGUAGE SPECIFICATION (CLS):

Visual Basic.NET is also compliant with CLS (Common Language Specification) and supports structured exception handling. CLS is set of rules and constructs that are supported by the CLR (Common Language Runtime). CLR is the runtime environment provided by the .NET Framework; it manages the execution of the code and also makes the development process easier by providing services.

Visual Basic.NET is a CLS-compliant language. Any objects, classes, or components that created in Visual Basic.NET can be used in any other CLS-compliant language. In addition. we can use objects. classes. and components created in other CLS-compliant languages in Visual Basic.NET .The use of CLS ensures complete interoperability among applications. regardless of the languages used to create the application.

IMPLEMENTATION INHERITANCE:

Visual Basic.NET supports implementation inheritance. This means that. while creating applications in Visual Basic.NET. we can drive from another class. which is know as the base class that derived class inherits all the methods and properties of the base class. In the derived class. we can either use the existing code of the base class or override the existing code. Therefore. with help of the implementation inheritance. code can be reused.

CONSTRUCTORS AND DESTRUCTORS:

Constructors are used to initialize objects. whereas destructors are used to destroy them. In other words, destructors are used to release the resources allocated to the object. In Visual Basic.NET the sub finalize procedure is available. The sub finalize procedure is used to complete the tasks that must be performed when an object is destroyed. The sub finalize procedure is called automatically when an object is destroyed.

OVERLOADING:

Overloading is another feature in Visual Basic.NET. Overloading enables us to define multiple procedures with the same name, where each procedure has a different set of arguments. Besides using overloading for procedures. we can use it for constructors and properties in a class.

FEATURES OF SQL-SERVER

The OLAP Services feature available in SQL Server version 7.0 is now called SQL Server 2000 Analysis Services. The term OLAP Services has been replaced with the term Analysis Services. Analysis Services also includes a new data mining component. The Repository component available in SQL Server version 7.0 is now called Microsoft SQL Server 2000 Meta Data Services. References to the component now use the term Meta Data Services. The term repository is used only in reference to the repository engine within Meta Data Services

SQL-SERVER database consist of six type of objects.

1. TABLE
2. QUERY
3. FORM
4. REPORT
5. MACRO

Microsoft SQL server lets you quickly build powerful and reliable database applications. SQL server 7.0 highly scalable. fully relational. high performance. multi-user database server. That can be used by enterprise of any size to manage large amount of data for client\server applications.

The major new and improved features of SQL server 7.0 include the multi-user support Multi platform support, added memory support, scalability, integration with MMC, Microsoft Management console and improved multiple server management. Parallel database backup and restore. Data replication. Data warehousing distributed queries, distributed transactions. Dynamic cocking Internet Access. Integrated windows security, Mail integration Microsoft English Query. ODBC Support.

SQL Server management is accomplished through a set of component applications. SQL Server introduces a number of new and improved management tools that are SQL Server Enterprise management, profiles, and Query Analyzer service manager wizards.

TABLE:

A database is a collection of data about a specific topic.

VIEWS OF TABLE

1. Design View
2. Datasheet View

Design View

To build or modify the structure of a table we work in the table design view. We can specify what kind of data will be hold.

Datasheet View

To add, edit or analyses the data itself we work in tables datasheet view mode.

QUERY:

A query is a question that has to be asked the data. Access gathers data that answers the question from one or more table. The data that make up the answer is either dynaset (if you edit it) or a snapshot(it cannot be edited).Each time we run query, we get latest information in the dynaset.Access either displays the dynaset or snapshot for us to view or perform an action on it ,such as deleting or updating.

FORMS:

A form is used to view and edit information in the database record by record .A form displays only the information we want to see in the way we want to see it. Forms use the familiar controls such as textboxes and checkboxes. This makes viewing and entering data easy.

Views of Form:

We can work with forms in several primarily there are two views.

1. **Design View**
2. **Form View**

Design View

To build or modify the structure of a form. we work in forms design view. We can add control to the form that are bound to fields in a table or query. includes textboxes. option buttons. graphs and pictures.

Form View

The form view which display the whole design of the form.

REPORT:

A report is used to vies and print information from the database. The report can ground records into many levels and compute totals and average by checking values from many records at once. Also the report is attractive and distinctive because we have control over the size and appearance of it.

MACRO:

A macro is a set of actions. Each action in macros does something. Such as opening a form or printing a report .We write macros to automate the common tasks the work easy and save the time.

MODULE:

Modules are units of code written in access basic language. We can write and use module to automate and customize the database in very sophisticated ways.

CHAPTER 4
SYSTEM DESIGN AND DEVELOPMENT

4.1 ELEMENTS OF DESIGN

Design is a multi-step process that focuses on data structure software architecture. procedural details. (algorithms etc.) and interface between modules. The design process also translates the requirements into the presentation of software that can be accessed for quality before coding begins.

Computer software design changes continuously as new methods: better analysis and broader understanding evolved. Software Design is at relatively early stage in its revolution.

Therefore. Software Design methodology lacks the depth. flexibility and quantitative nature that are normally associated with more classical engineering disciplines. However techniques for software designs do exist. criteria for design qualities are available and design notation can be applied.

4.1.1 INPUT DESIGN

Input design is the process of converting user-originated inputs to a computer-based format. Input design is one of the most expensive phases of the operation of computerized system and is often the major problem of a system.

In the project. the schedule selection page is made with several easy to use options.

For example. to select the day. date or time. we need to select one of the values itself; we need not type the time. In addition. maximum copies can be changed anytime easily.

4.1.2 OUTPUT DESIGN

Output design generally refers to the results and information that are generated by the system for many end-users; output is the main reason for developing the system and the basis on which they evaluate the usefulness of the application. In any system. the output design determines the input to be given to the application.

Output design generally refers to the results and information that are generated by the system for many end-users; output is the main reason for developing the system and the basis on which they evaluate the usefulness of the application.

The output is designed in such a way that it is attractive. convenient and informative. Forms are designed in VB.NET with various features. which make the console output more pleasing.

As the outputs are the most important sources of information to the users. better design should improve the system's relationships with us and also will help in decision-making. Form design elaborates the way output is presented and the layout available for capturing information.

In the project. the source path. backup path and backup databases view are the output page available in the form of reports.

4.1.3 DATABASE DESIGN

The database design is a must for any application developed especially more for the data store projects. Since the chatting method involves storing the message in the table and produced to the sender and receiver. proper handling of the table is a must.

In the project. admin table is designed to be unique in accepting the username and the length of the username and password should be greater than zero.

A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general theme behind a database is to integrate all the information. The general objective of database design is to make the data access easy. inexpensive and flexible to the user.

The main objectives of designing a database are:

- Data integration
- Data integrity
- Data independence

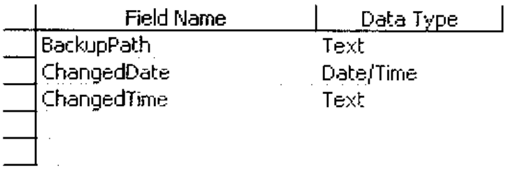
4.1.4 SYSTEM DEVELOPMENT

System development is a series of operations performed to manipulate data to produce output from computer system. This aim at translating the design of the system produced during the design phase into code in user programming language. A modular approach is used for the development of the software.

The development phase for the project was created from the specification created during the design phase. A principal activity of the development phase is coding and testing the computer program that make up the computer program component of the overall system. Other important activities include implementation. planning. equipment acquisition and system. The development phase concludes with the report and review.

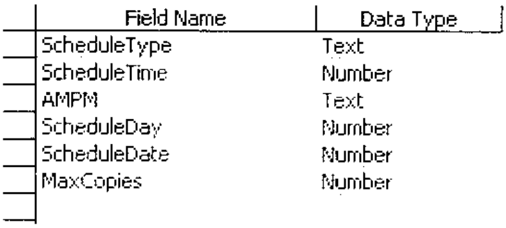
DATA TABLE STRUCTURE

Backup path:



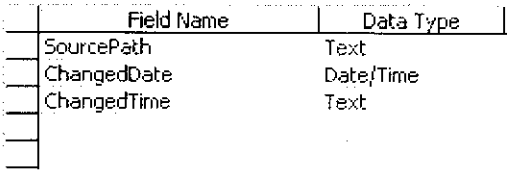
Field Name	Data Type
BackupPath	Text
ChangedDate	Date/Time
ChangedTime	Text

Schedule:



Field Name	Data Type
ScheduleType	Text
ScheduleTime	Number
AMPM	Text
ScheduleDay	Number
ScheduleDate	Number
MaxCopies	Number

Source Path:



Field Name	Data Type
SourcePath	Text
ChangedDate	Date/Time
ChangedTime	Text

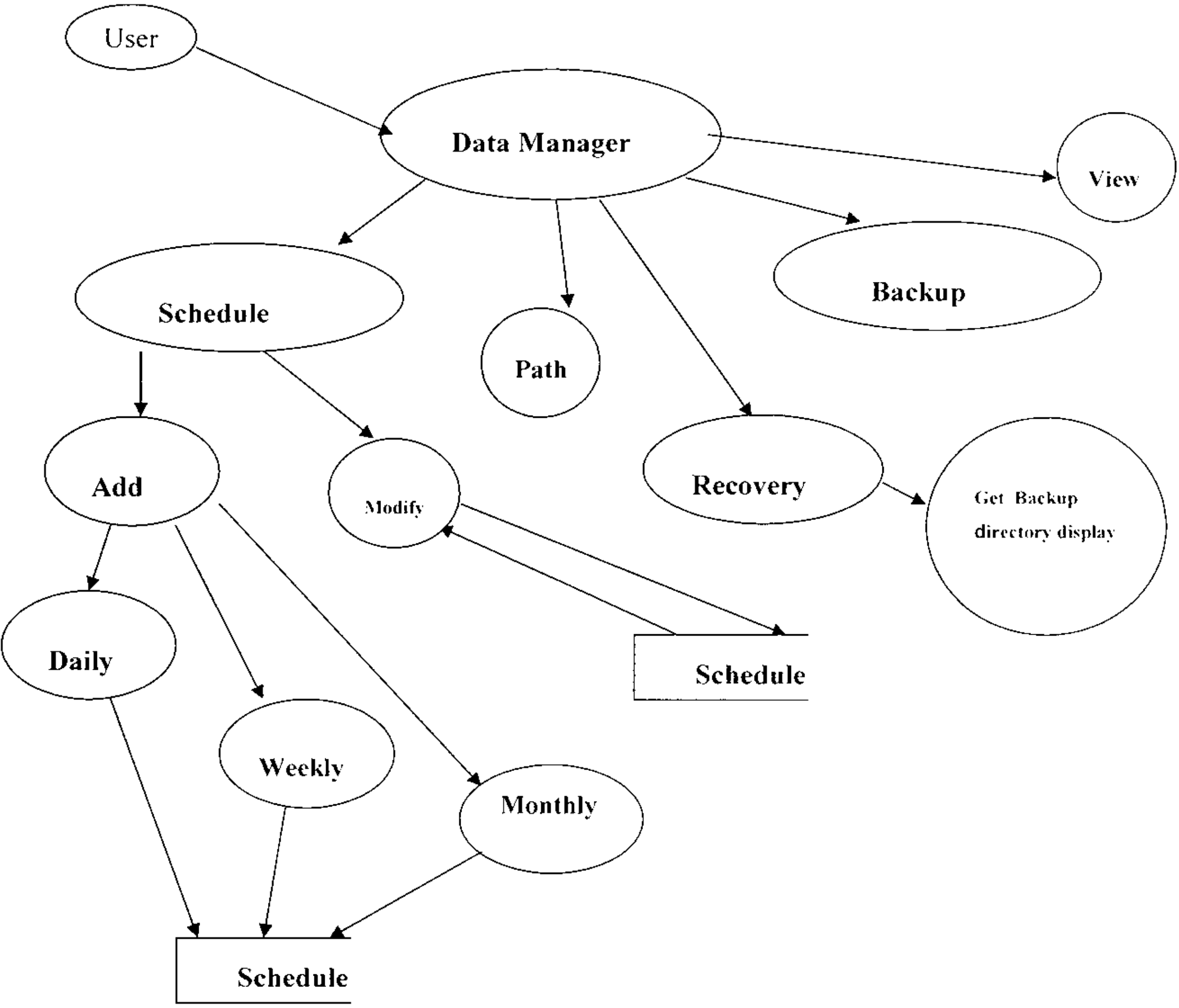
4.2 DATA FLOW DIAGRAM:

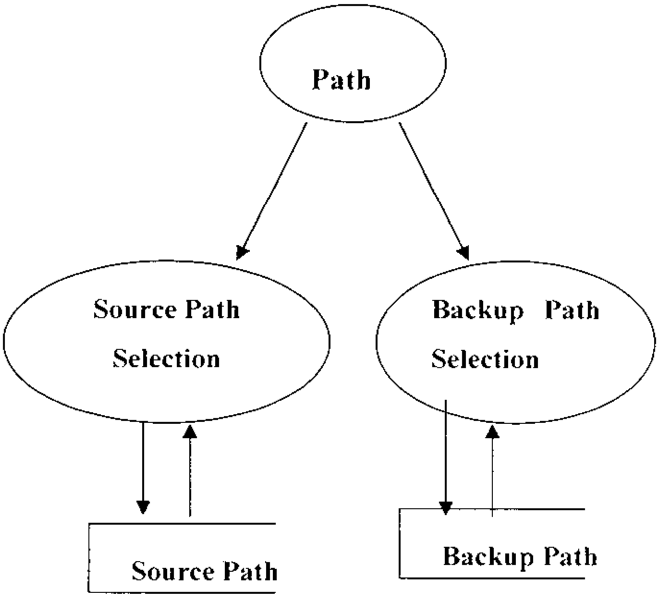
A Data Flow Diagram is used to define the flow of the system and the resources such as information. It is the way of expressing system requirements in a graphical manner. It is also known as bubble chart. It consists of a series of bubbles joined by lines. The bubbles represent data transformation and the lines represent data flow in the system.

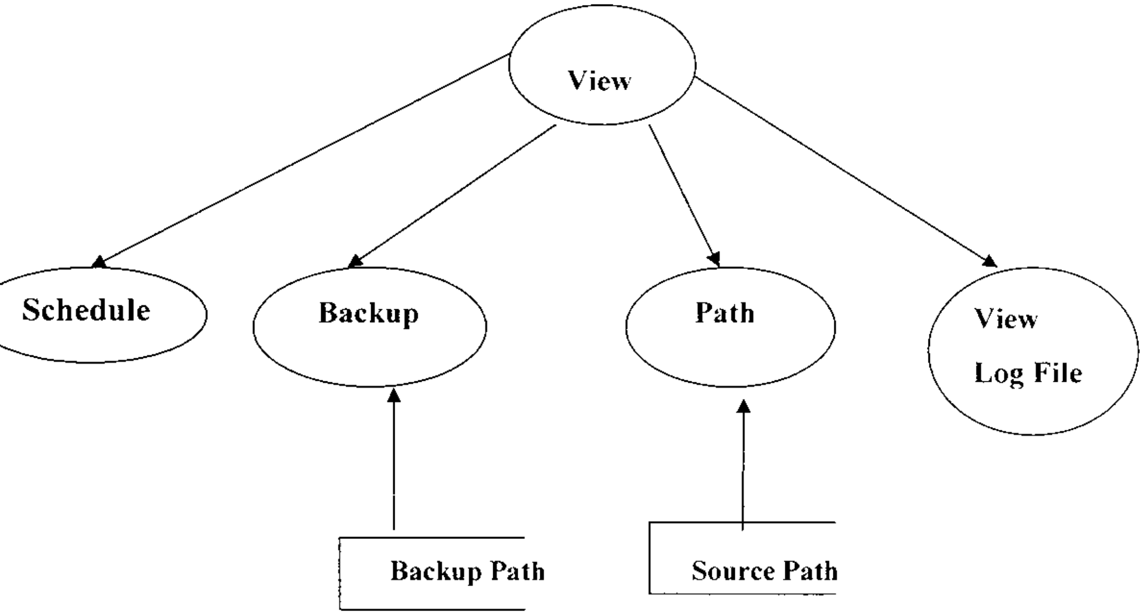
Data Flow Diagram is directed graphs in which the nodes specify processing activities and the arcs that specify data items transmitted between processing nodes. Like flow charts, data flow diagram can be used at any desired level of abstraction. A data flow diagram can be used to represent data flow between individual statements or block statements in a routine, data flow sequential routine between concurrent processes or data in a distributed computing system, where each node represents a geographically remote processing unit. Unlike flowcharts, data flow diagrams do not indicate decision logic or condition under which various processing nodes in the diagram being activated.

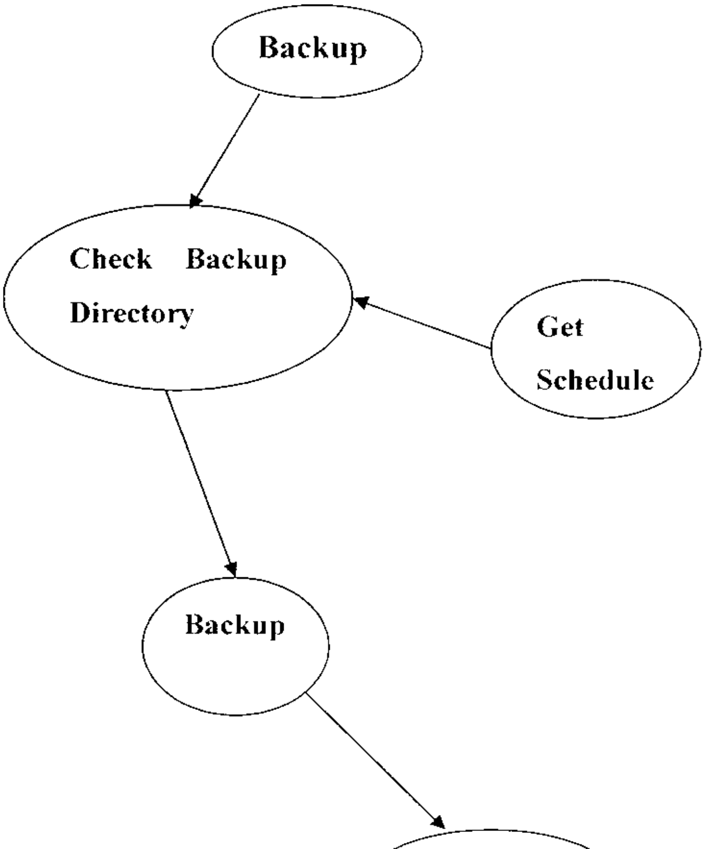
Data flow diagrams are excellent mechanisms for communicating with customers during requirement analysis; also they are widely used for representation of external and top-level internal design specifications.

The Data flow diagrams may be used to represent a system or software at any level of abstraction. DFD's may be partitioned into levels that represent increasing information flow and functional details.


User
Data Manager
View
Backup
Schedule
Path
Recovery
Get Backup directory display
Add
Modify
Schedule
Daily
Weekly
Monthly
Schedule

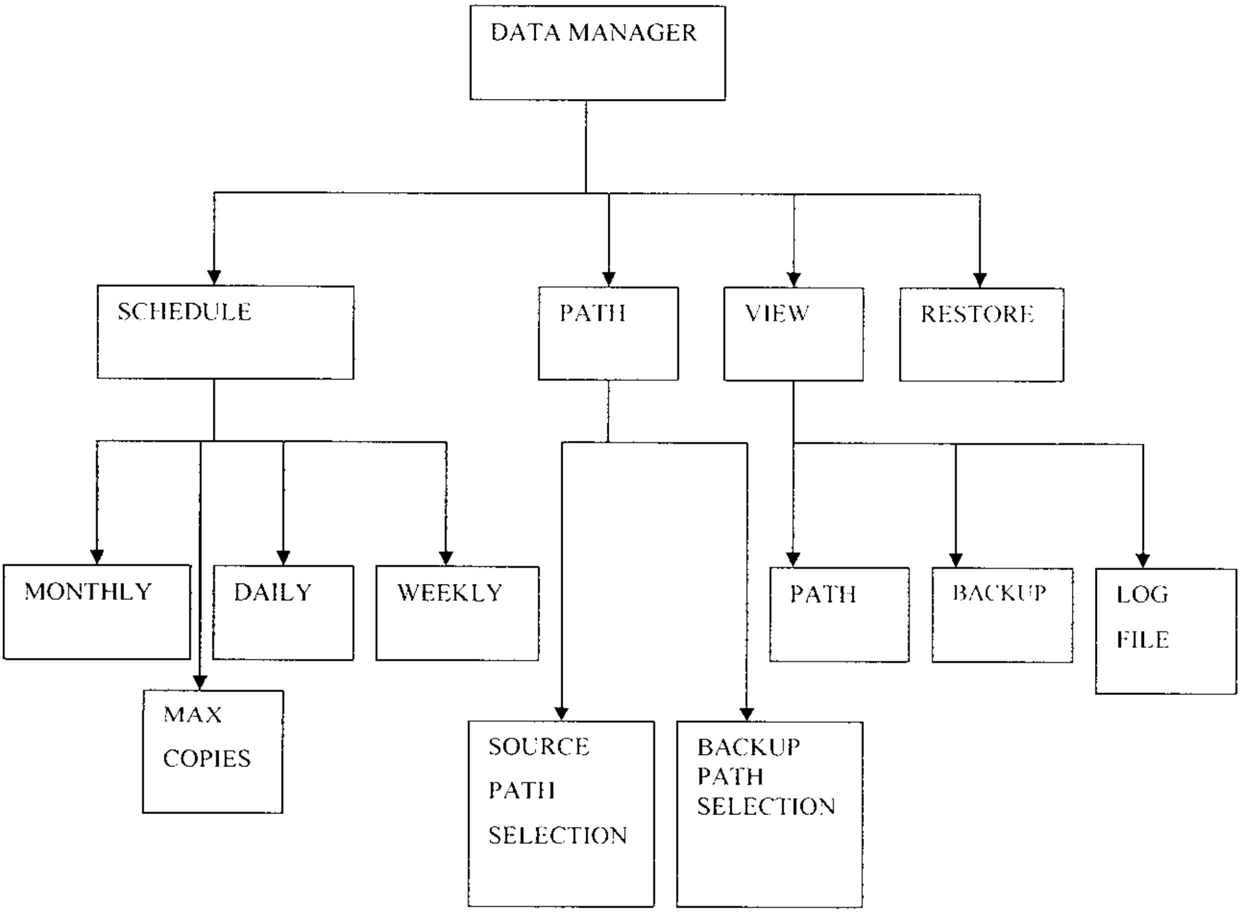

Path
Source Path Selection
Backup Path Selection
Source Path
Backup Path

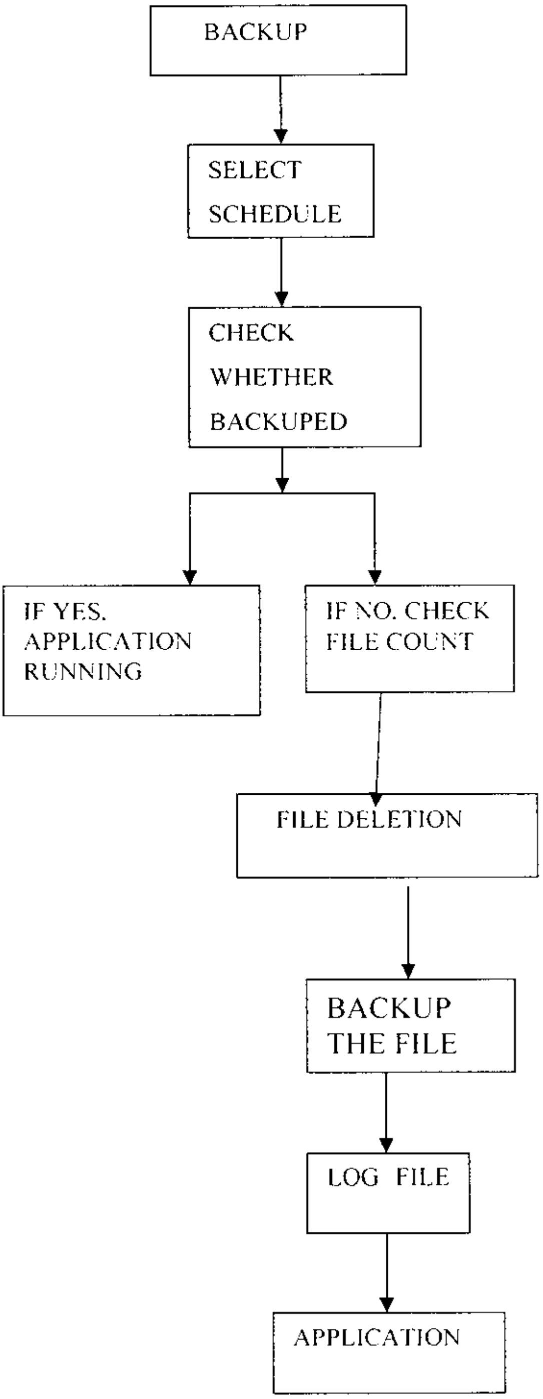

View
Schedule
Backup
Path
View Log File
Backup Path
Source Path


Backup
Check Backup Directory
Get Schedule
Backup

4.3 SYSTEM FLOW DIAGRAM

A system flow diagram (also called flow chapter) is a schematic representation of an algorithm or a process. A flowchart is one of the seven basic tools of quality control. Which also includes the histogram. Pareto chart, check sheet. control chart. cause-and-effect diagram. and scatter diagram. They are commonly used in business/economic presentation to help the audience visualize the content better. or to find flaws in the process.




BACKUP
SELECT SCHEDULE
CHECK WHETHER BACKUPED
IF YES. APPLICATION RUNNING
IF NO. CHECK FILE COUNT
FILE DELETION
BACKUP THE FILE
LOG FILE
APPLICATION

CHAPTER 5
ARCHITECTURAL DESIGN

5.1 MODULE DESIGN

The entire project is mainly divided in to 5 modules

1. Schedule selection
2. Path selection
3. Backup module
4. Recovery module
5. Reports

1. Schedule selection

The backup schedule may be either daily backup. weekly or monthly backup. The date or day. time. AM or PM should be selected. In addition. the maximum number of copies should be selected. At any time. the schedule can be changed.

2. Path selection

The source path i.e., which folder is to be taken for backup is selected. In addition, backup path i.e., where the folder will be backup should be selected from here. The total folder can be selected for backup.

3. Backup module:

In this module. the backup application starts running from the system logon time. First, the applications checks whether today is in the backup schedule and the backup is already being taken. If backup is already taken, the application continues running without any additional work. If no, it checks for the maximum copies. If the folders present in the backup folder are equal to the maximum, then the first created folder is deleted and the backup takes place. Otherwise. the backup takes place without any deletion.

4. Recovery module:

First. we can view all the backup folders in the backup path. Then we should select any folder and click recovery to copy the selected folder to the original source path.

5. Reports

Schedule details and Paths can be viewed. The backup folder details also can be viewed. The log file is used to retrieve the log details of the scheduling operation and the resource path.

CHAPTER 6
SYSTEM TESTING AND MAINTENANCE

6.1. OBJECTIVE OF TESTING

The objective of testing is to prove that there are no errors in the software. This is extremely difficult since developer cannot prove to be hundred percent accurate. Therefore the most useful and practical approach is with the understanding that testing is the process of executing a program with explicit intention of finding errors and check for the basic flow of the process.

Testing has its own cycle. The testing process begins with the product requirements phase and from there parallels the entire development process. In other words for each phase of the development process there is an important testing activity. Successful testing requires a systematic approach. It requires focusing on the basic critical factors: planning. project control. risk management. inspections. measurement. tools. organization and professionalism.

6.2 TYPES OF TESTING

6.2.1 SYSTEM TESTING

Testing is a set of activities that can be planned in advance and conducted systematically. A number of testing strategies have been proposed: in literature all provide the software developer with the template for testing and having the following generic characteristics.

- Testing begins at the component level and works outward towards the integration of the entire computer based system.
- The developer of the software conducts testing and for large products an independent test group may be used.

- Testing and debugging are different activities but debugging may be accommodated in any testing strategy.

- The Backup and Recovery is thoroughly test with this test strategy and the test report is prepared. all detected errors are corrected.

SOFTWARE TESTING TECHNIQUES:

The test case design methods applied are

- White Box Testing
- Black Box Testing

6.2.1.1 WHITE BOX TESTING:

Using this testing method it was assured that all the independent paths were exercised at least once. All the logical decisions on their true and false side were executed. All loops were executed at their boundaries.

6.2.1.2 BLACK BOX TESTING:

Using this testing technique. incorrect and missing functions were identified and corrected. incorrect information. interfacing errors; performance errors. initialization errors and termination errors were also found using this technique.

SOFTWARE TESTING STRATEGIES:

A strategy for the software testing integrates software test case design techniques into well planned series of steps that result in the successful construction of software. Any testing strategy must incorporate Test Planning. Test Case Design. Resultant Data Collection and Evaluation.

The different levels of testing are:

6.2.2 UNIT TESTING:

This kind of testing is to verify the smallest unit of the software module. This is also known as "Module Testing". This test is carried out during the programming stage. This test ensures the expected output from each of the module. The modules including bill board. media, internet and vehicle have been tested for robustness. Exceptions have been handled and appropriate Error messages have been given in each module so as to avoid abnormal termination of the program.

The unit testing considers the actions that were taken into account is as follows:

- Interfacing errors
- Integrity of local data structures.
- Boundary conditions.
- Independent paths.
- Error handling paths

TEST CASE REPORTS:

S. No	Module	Typical test strategy	Excepted Output	Actual Output	Test Status
1	Schedule	Click the daily menu option in the schedule menu	Select Hour dialog box should be displayed	Hour dialog box is display	Pass
2	Schedule	Click the ok button in the Select Hour Dialog Box	Schedule gets changed.	Schedule changed	Pass

3	Schedule	Click the weekly menu option in the schedule menu	Select Day and Hour dialog box should be displayed	Day and hour dialog box is display	Pass
4	Schedule	Click the monthly menu option in the schedule menu	Select Month. Day and Hour dialog box should be displayed	Month, day and Hour dialog box is display	Pass
5	Source Path	Select the source path in the folders tree view control	Display the selected path in the source path text box	Path displayed in the source path text	Pass
6	Source Path	Click the save button	folder will be copied	Folder is copied	Pass
7	Backup Path	Select the backup Path in the folders tree view control	Display the selected path in the Backup path text box	Path displayed in the backup path text	Pass
8	Backup path	Click the save button	folder will be added	Added message is displayed.	Pass
9	Recovery	In Recovery Menu, click the Show backup button.	Backup path will be shown with the file name.	Backup path shown.	Pass

10	Restore	Select the backup path and click the Restore button.	Folder modified message box should be displayed.	Folder modified message box displayed and the backup path is converted to source path.	Pass
11	View	Click the view menu from the menu bar.	Sub menu should be shown.	The sub menus schedule. backups. path. view log file are displayed.	Pass
12	View Schedule	From the sub menu schedule option is selected.	Backup Schedule to be shown.	Backup Schedule is displayed with Schedule type. Schedule time. Schedule date. Schedule day. Schedule month.	Pass
13	View Backup Path	From the sub menu View path option is selected.	Backup path to be displayed with schedule.	Backup path with schedule is displayed.	Pass
14	View Source Path.	From the sub menu View path option is selected.	Source path to be displayed with schedule.	Converted Backup path to Source path with schedule is displayed.	Pass.
15	Log file	Click the view log file.	Final report with all the schedule details should be shown.	Detailed final report is generated.	Pass

6.2.3 INTEGRATION TESTING:

This kind of testing is a systematic testing for constructing tests to uncover errors associated within the interface. The objective is to take unit tested modules and build a program structure. All the modules are combined and tested as a whole. The system underwent a series of Integration tests that recorded smooth transmission of data from one module to the other. The bottom up approach was applied.

In this project the developed system is tested after integrating various modules together, and the detected errors were corrected.

6.2.4 VALIDATION TESTING:

Validation testing is carried out to verify whether the software functions works in a manner that is expected by the customer. So alpha Testing was done to ensure validity.

6.3 MAINTENANCE:

The objectives of this maintenance work are to make sure that the system gets into work all time without any bug. Provision must be for environmental changes which may affect the computer or software system. This is called the maintenance of the system. Nowadays there is the rapid change in the software world. Due to this rapid change, the system should be capable of adapting these changes. In our project the process can be added without affecting other parts of the system.

Maintenance plays a vital role. The system liable to accept any modification after its implementation. This system has been designed to favor all new changes. Doing this will not affect the system's performance or its accuracy.

6.4 SYSTEM IMPLEMENTATION

Implementation is the most crucial stage in achieving a successful system and giving the user's confidence that the new system is workable and effective. Implementation of a modified application to replace an existing one. This type of conversation is relatively easy to handle, provide there are no major changes in the system.

Each program is tested individually at the time of development using the data and has verified that this program linked together in the way specified in the programs specification, the computer system and its environment is tested to the satisfaction of the user. The system that has been developed is accepted and proved to be satisfactory for the user. And so the system is going to be implemented very soon. A simple operating procedure is included so that the user can understand the different functions clearly and quickly.

Initially as a first step the executable form of the application is to be created and loaded in the common server machine which is accessible to all the user and the server is to be connected to a network. The final stage is to document the entire system which provides components and the operating procedures of the system.

CHAPTER 7
PERFORMANCE AND LIMITATIONS

7.1 MERITS

This application is useful for taking the automation backup. The user can choose the schedule for taking backup such as day. date. month and year. The request will be processed and the user will obtain the satisfaction on better performance.

7.2 LIMITATIONS OF THE SYSTEM

Each program is tested individually at the time of development using the data and has verified that this program linked together in the way specified in the programs specification. the computer system and its environment is tested to the satisfaction of the user. The system that has been developed is accepted and proved to be satisfactory for the user. And so the system is going to be implemented very soon. A simple operating procedure is included so that the user can understand the different functions clearly and quickly.

7.3 FUTURE ENHANCEMENT

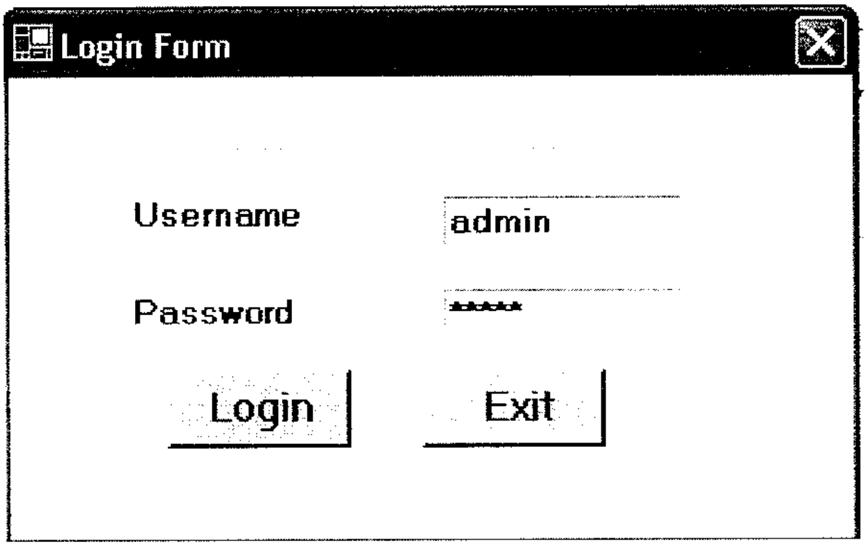
Every application has its own merits and demerits. The project has covered almost all the requirements. Further requirements and improvements can easily be done since the coding is mainly structured or modular in nature. Changing the existing modules or adding new modules can append improvements. Further enhancements can be made to the application. so that the web site functions very attractive and useful manner than the present one.

CHAPTER 8

APPENDICES

SCREEN SHOTS

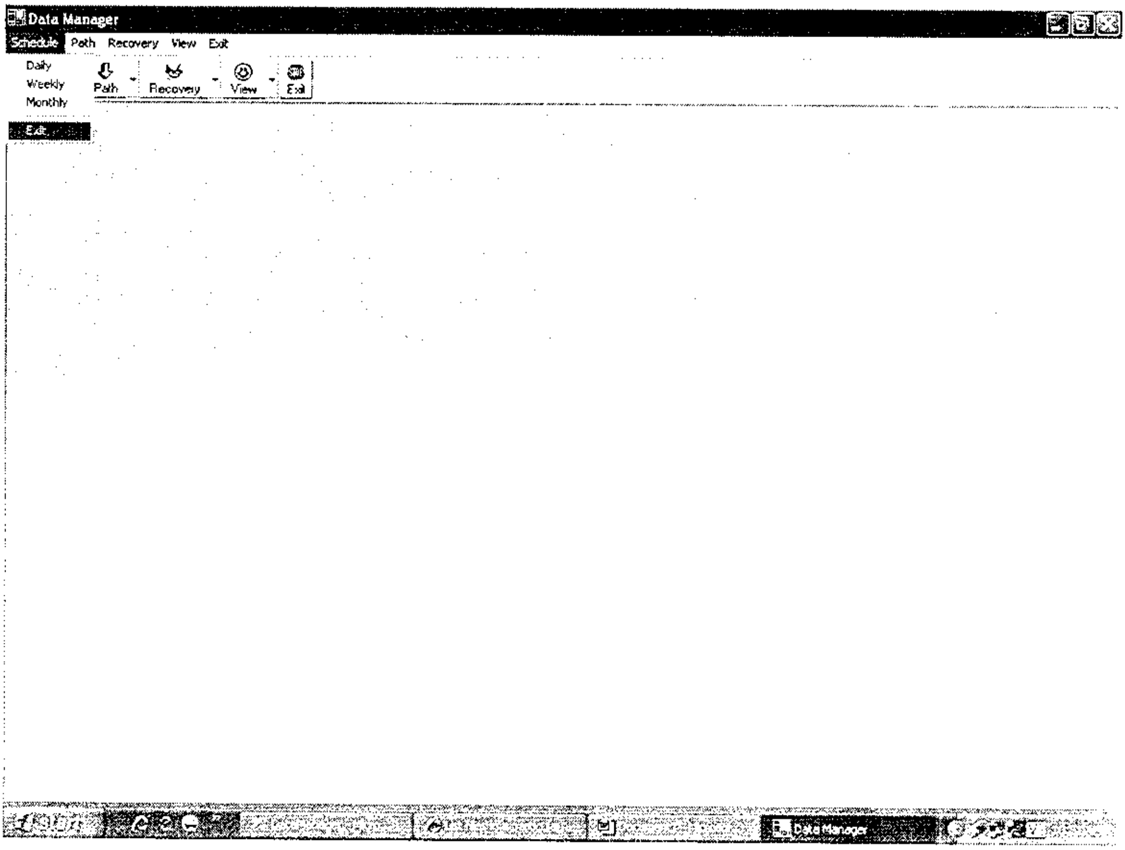
Login form



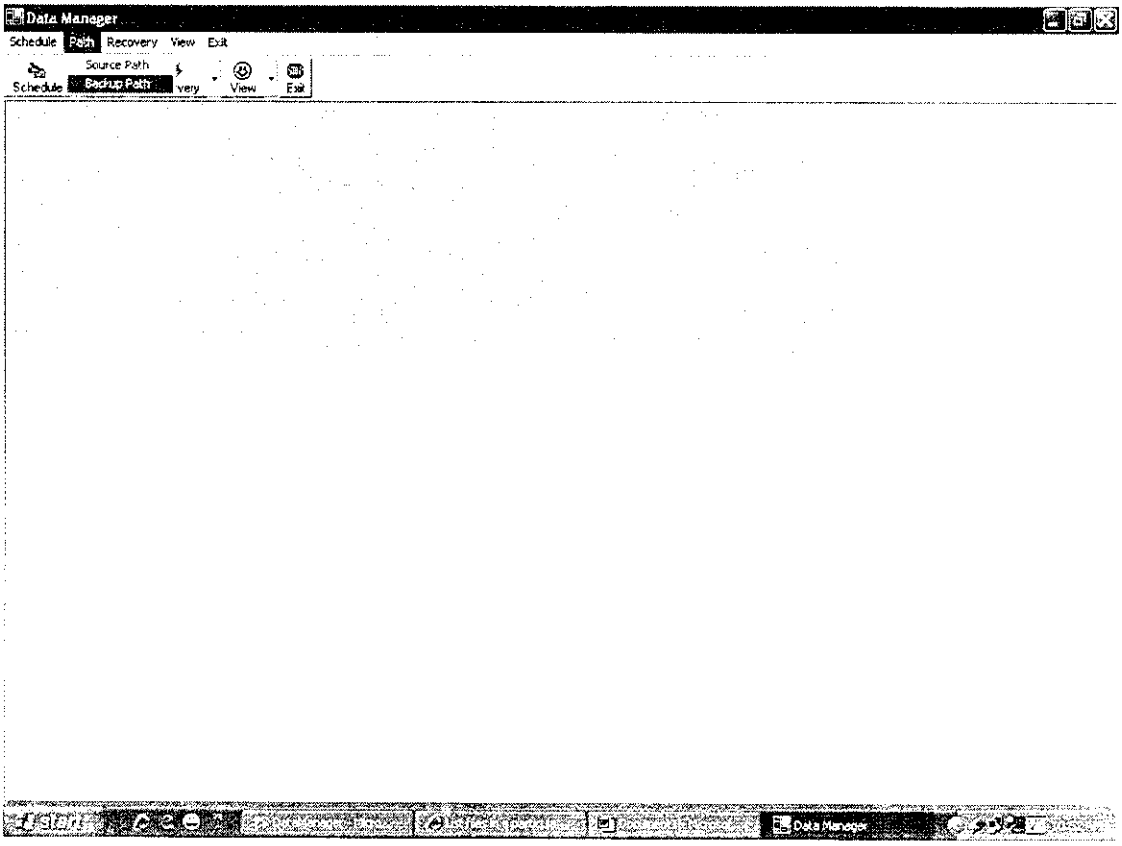
Username: admin

Password: *****

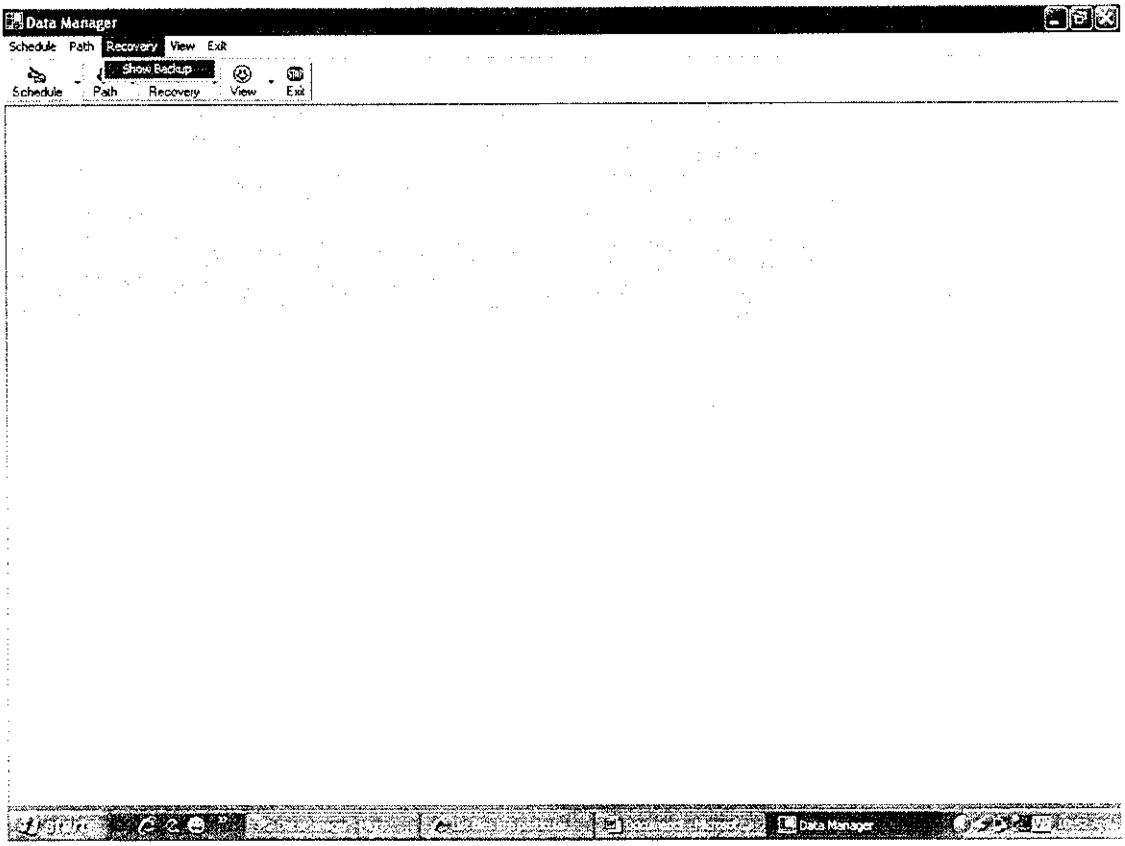
SCHEDULE:


Data Manager
Schedule
Path
Recovery
View
Exit
Daily
Weekly
Monthly
Exit
Path
Recovery
View
Exit
Data Manager

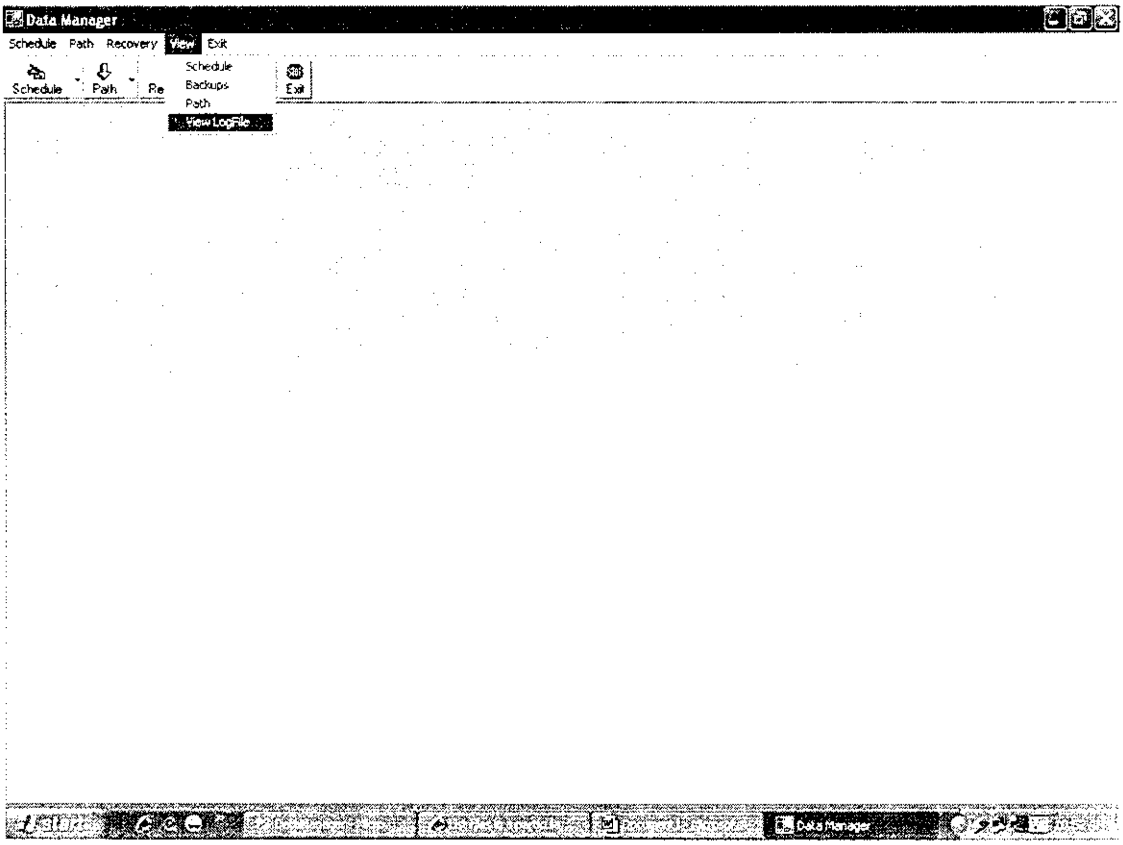
PATH:


Data Manager
Schedule
Path
Recovery
View
Exit
Source Path
Backup Path
Schedule
very
View
Exit
start
Data Manager

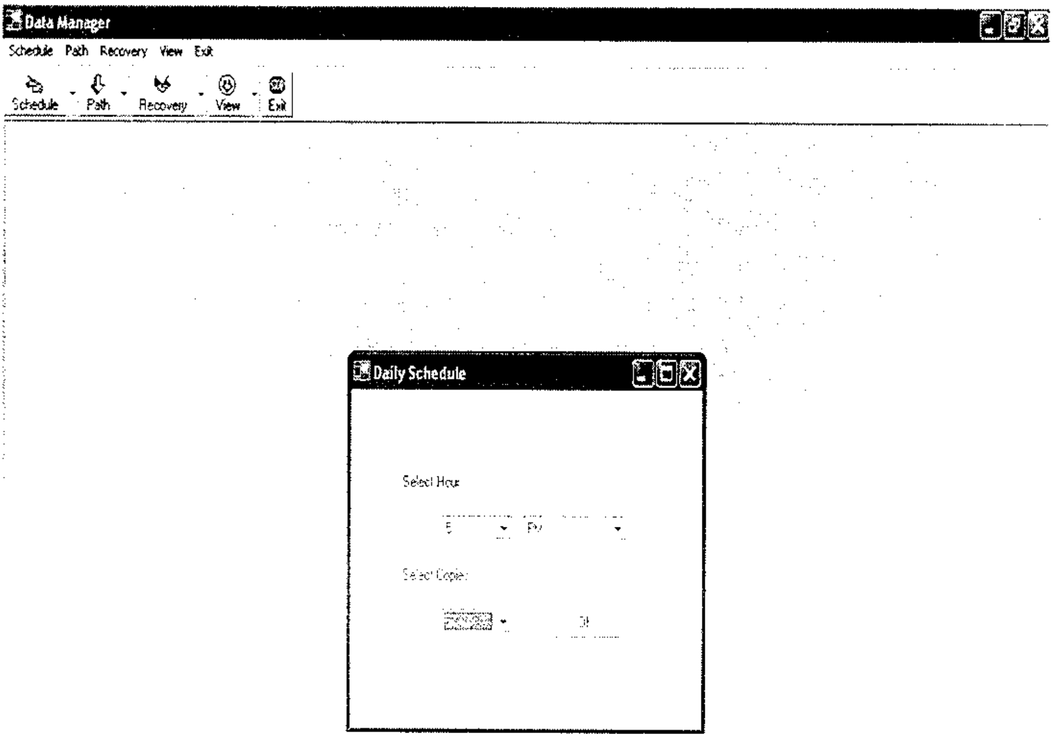
RECOVERY:


Data Manager
Schedule
Path
Recovery
View
Exit
Show Backup
Schedule
Path
Recovery
View
Exit
Data Manager

VIEW:


Data Manager
Schedule
Path
Recovery
View
Exit
Schedule
Path
Re
Exit
Schedule
Backups
Path
View LogFile
Data Manager

DAILY SCHEDULE:

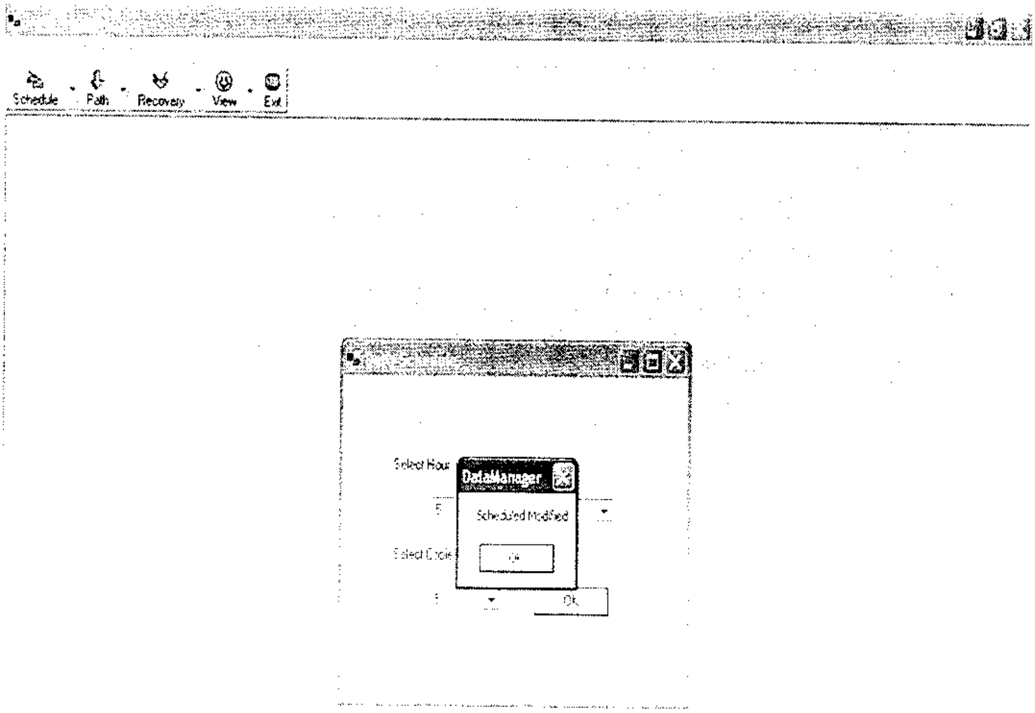


Select Hour

5 PM

Select Copies

DAILY SCHEDULE:



Select Hour

5

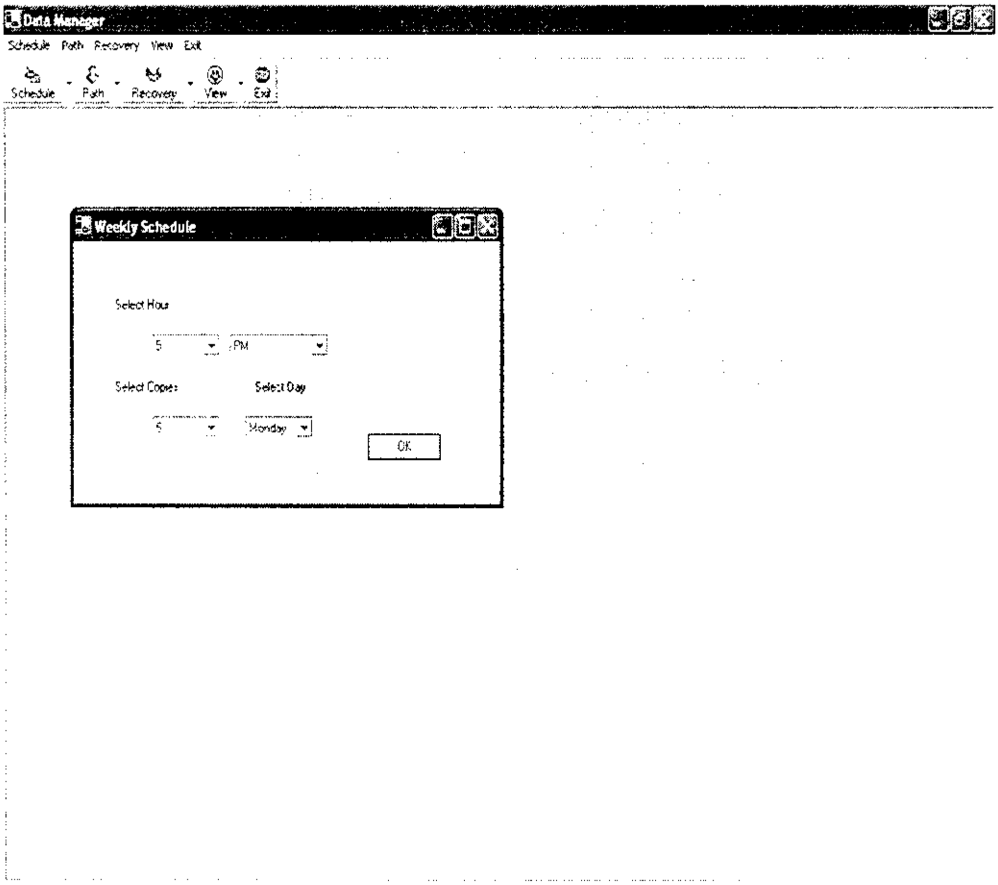
Select Choie

1

DataManager

Scheduled Modified

WEEKLYSCHEDULE:



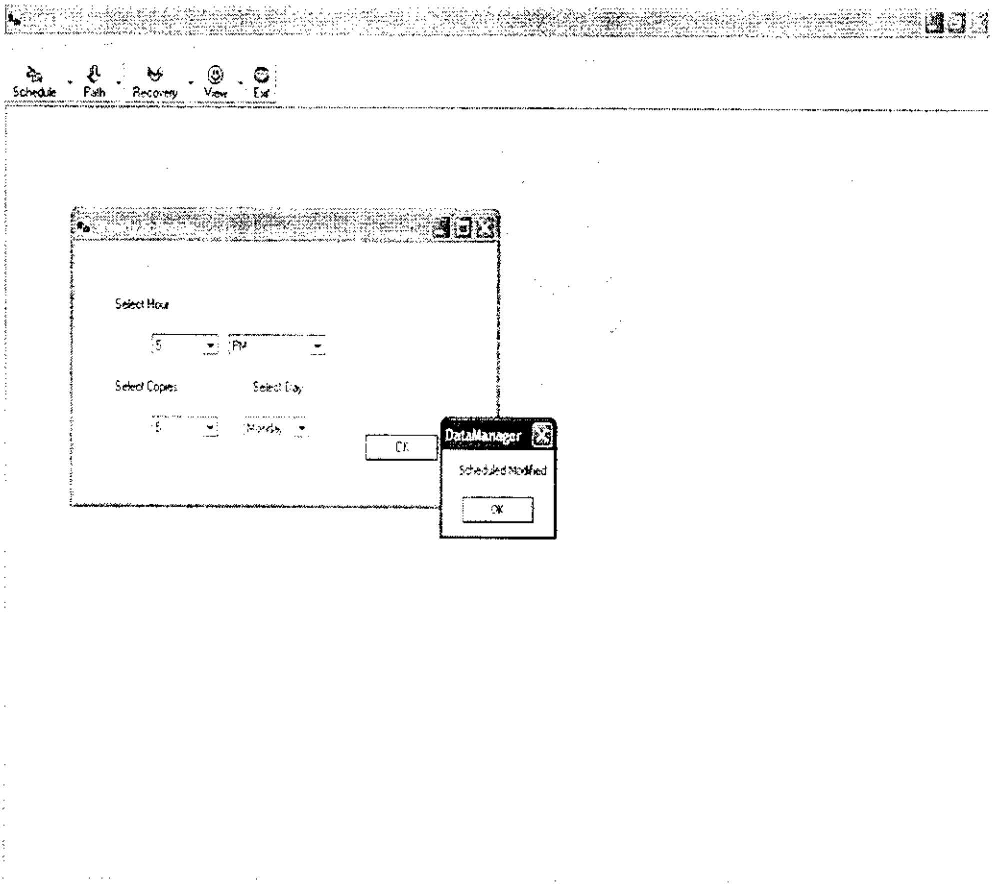
Select Hour

5 PM

Select Copies 5

Select Day Monday

WEEKLY SCHEDULE:



Select Hour

5 PM

Select Copies

5

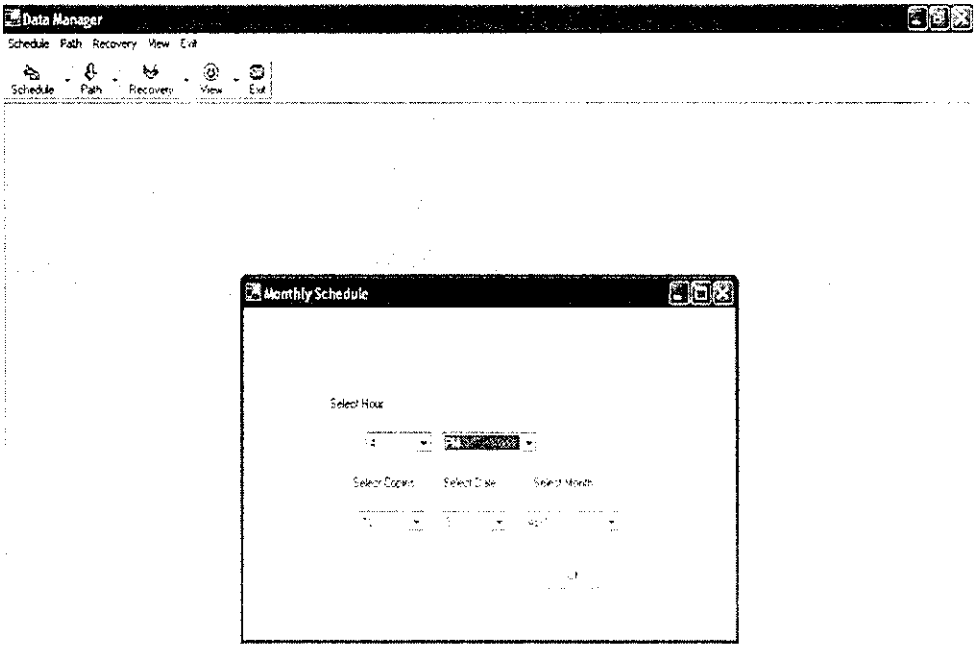
Select Day

Monday

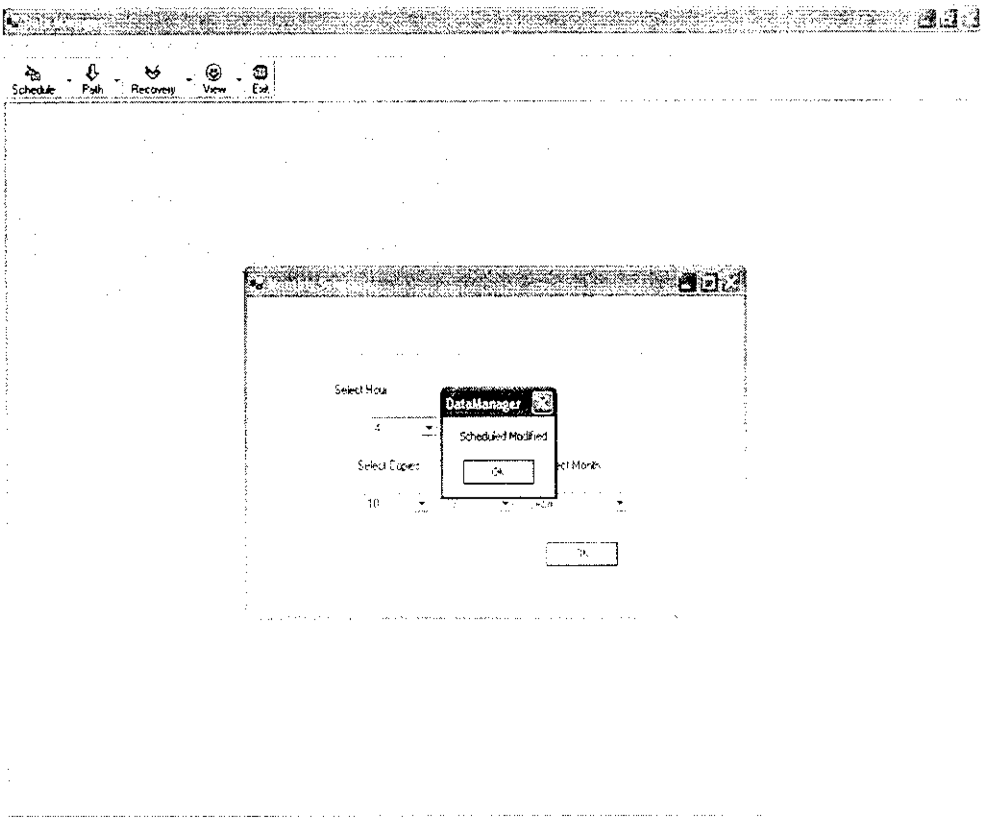
DataManager

Scheduled Modified

MONTHLY SCHEDULE:



MONTHLY SCHEDULE:



Select Hour

4

Select Copies

10

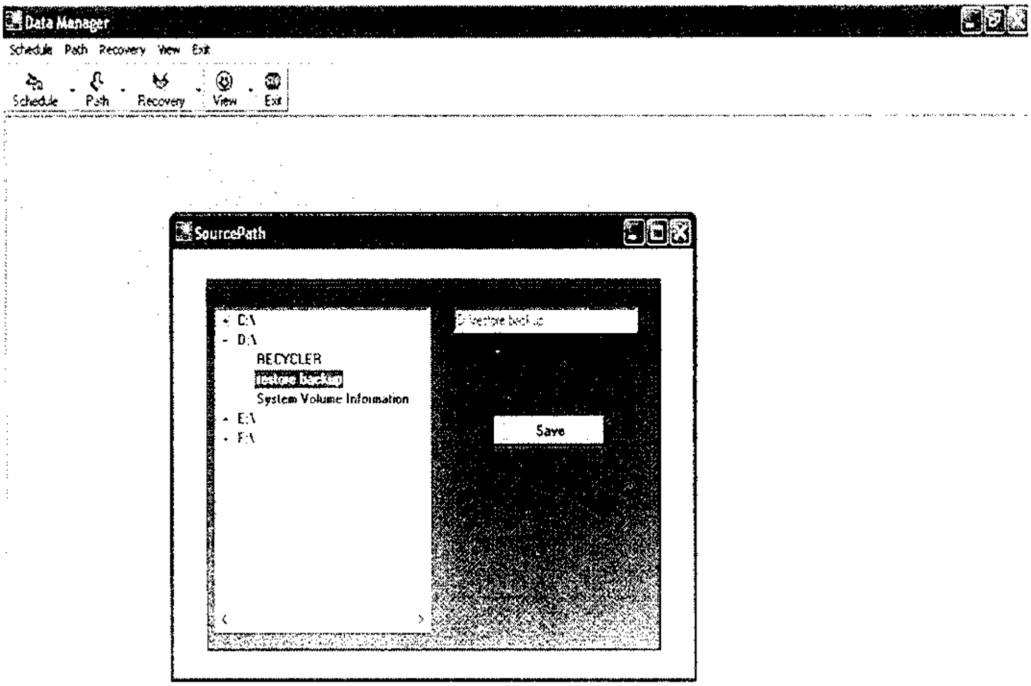
ect Month

DataManager

Scheduled Modified

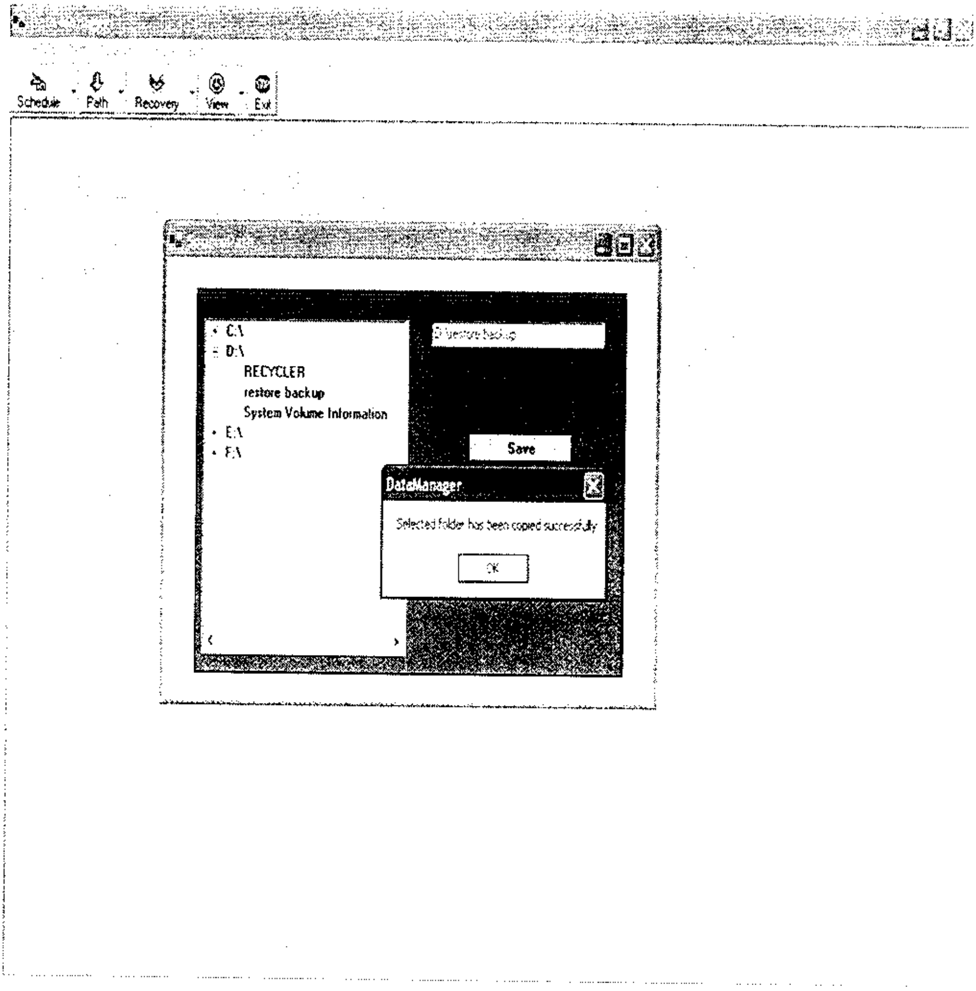
OK

SOURCEPATH:



D:\restore backup

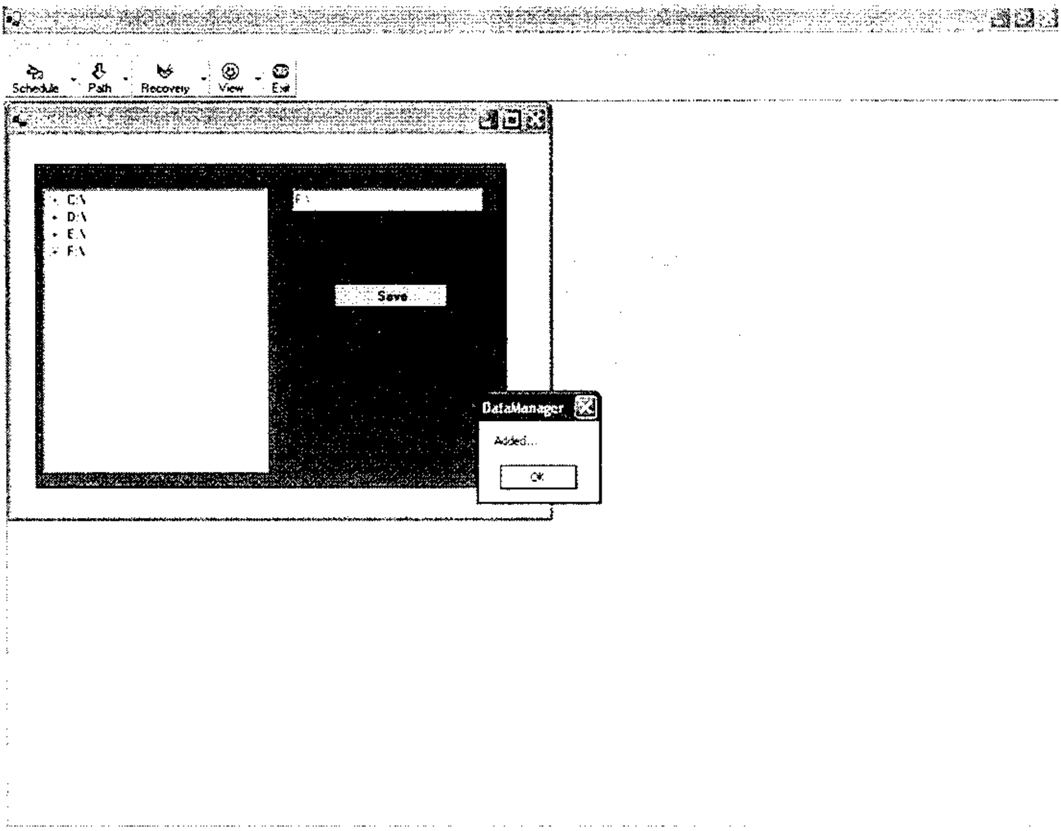
SOURCE PATH:



D:\restore backup

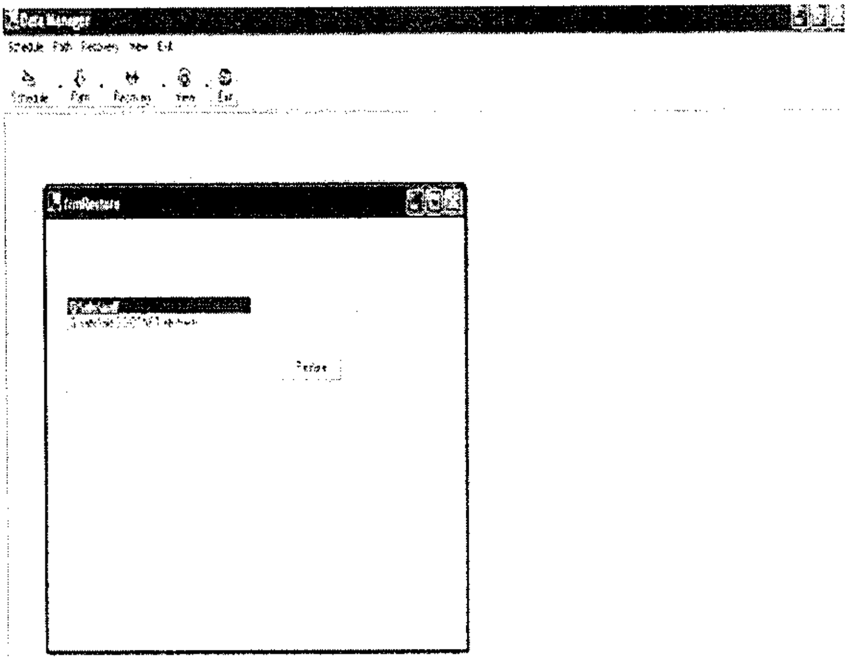
Selected folder has been copied successfully

BACKUP PATH:

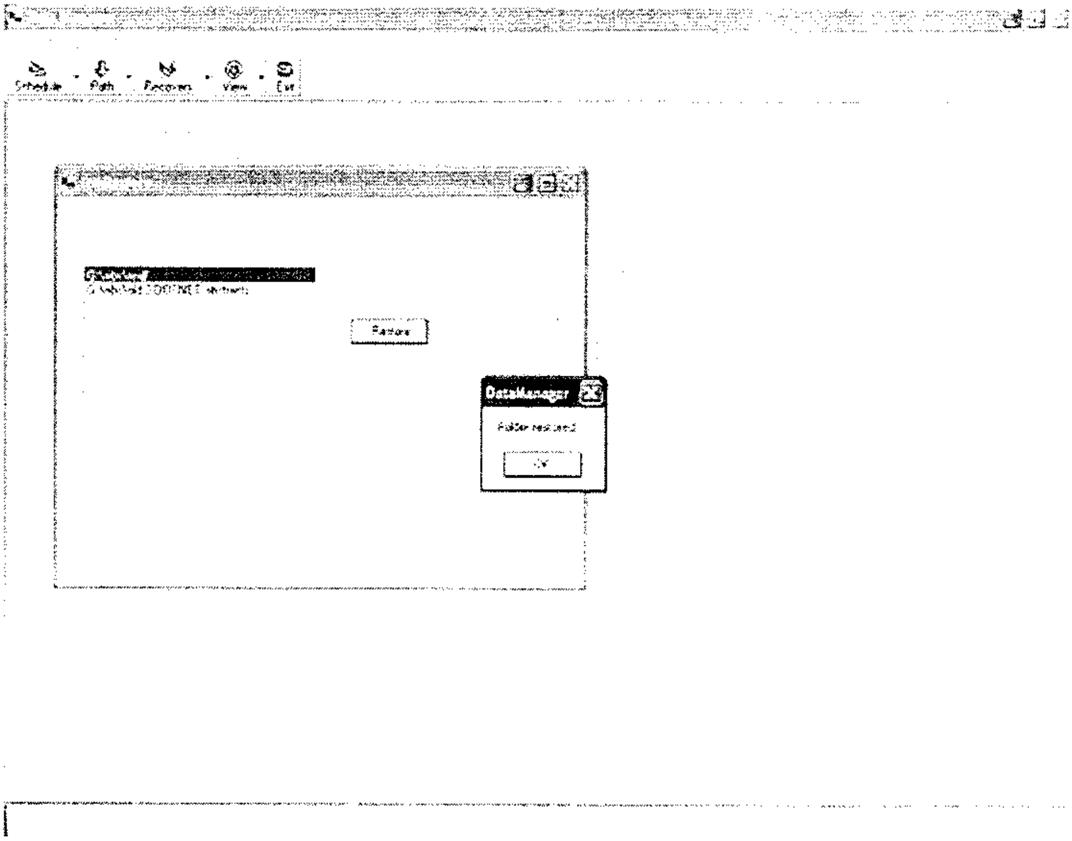


Added...

RESTORE:

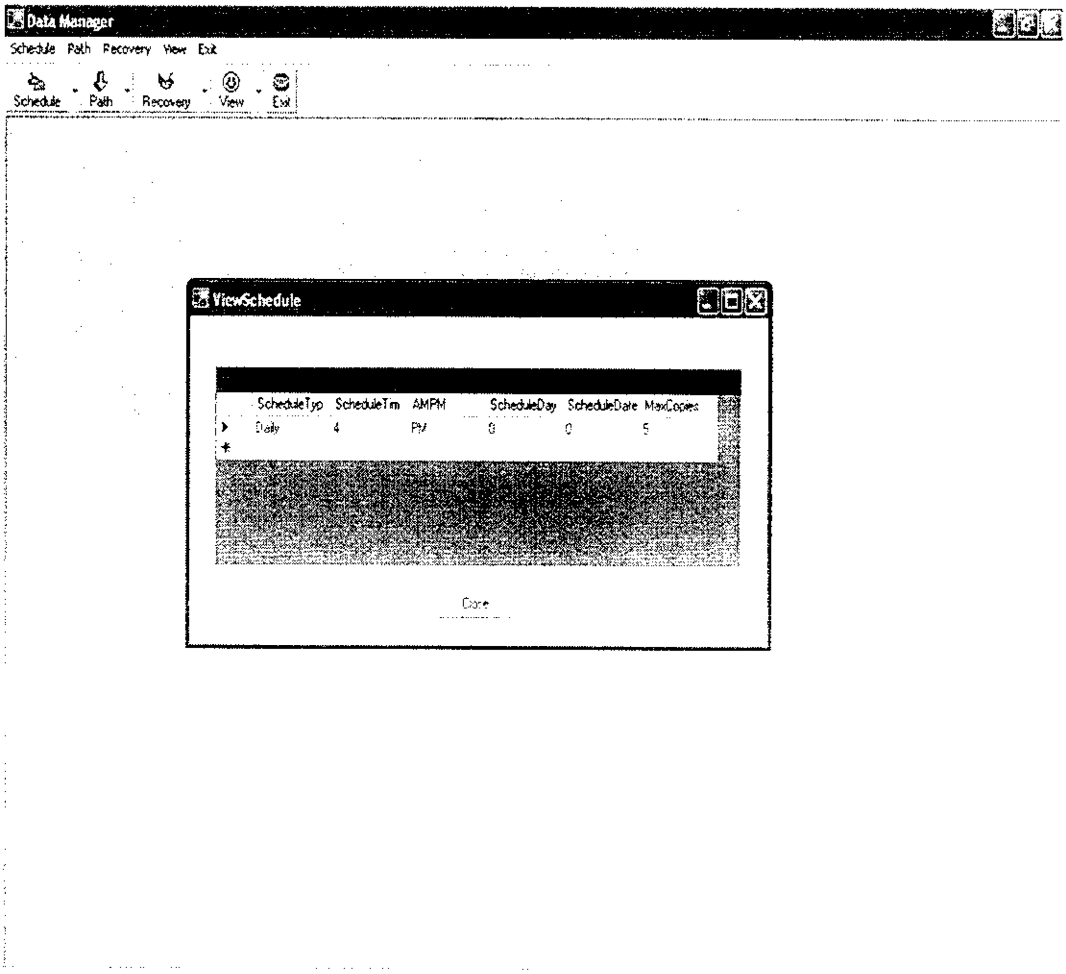

Data Manager
Schedule Path Recovery View Exit
Schedule
Path
Recovery
View
Exit
frmRestore
Restore

RESTORE:



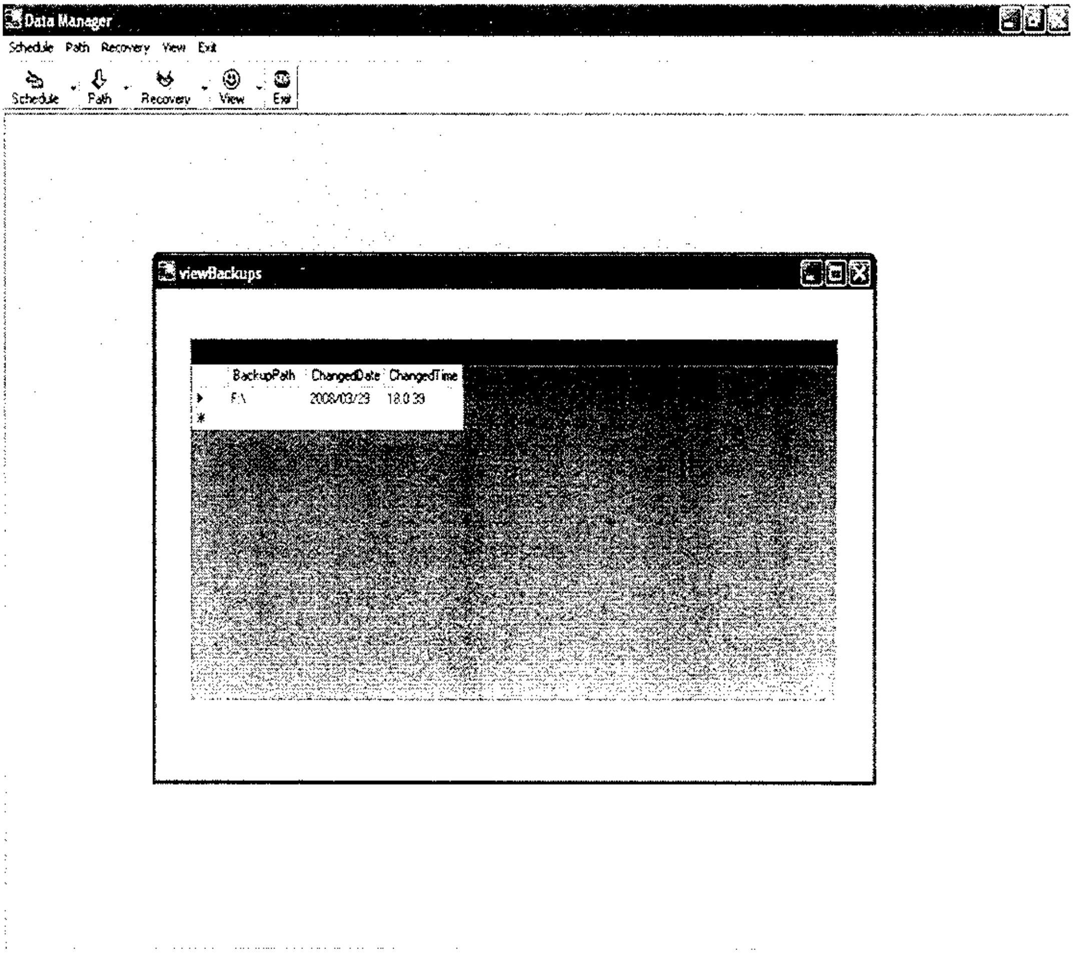
Folder restored

VIEW SCHEDULE:



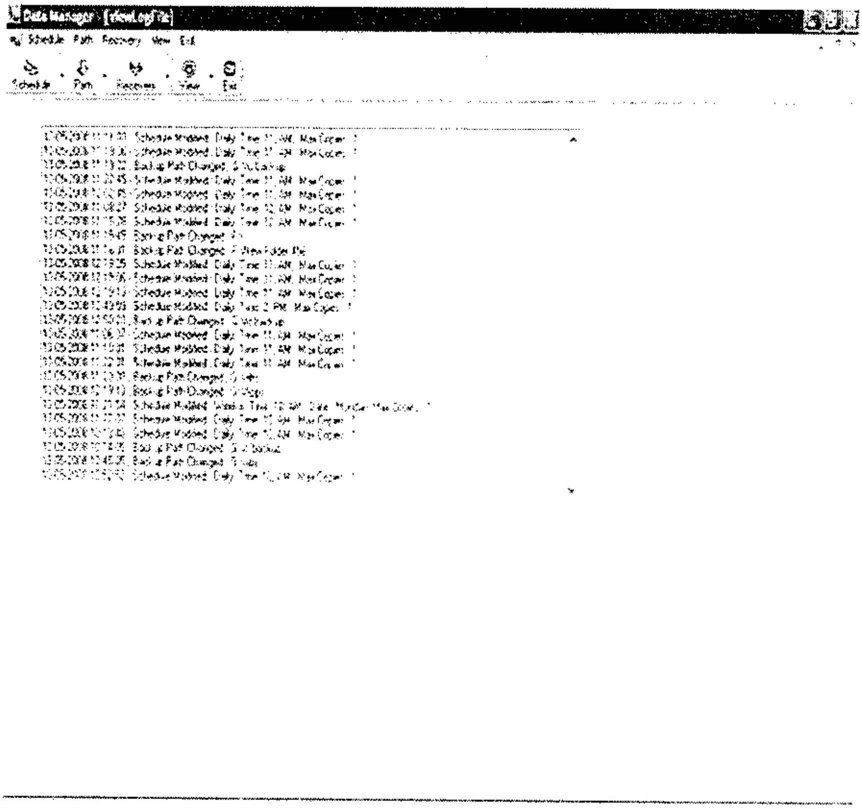
	ScheduleTyp	ScheduleTim	AMPM	ScheduleDay	ScheduleDate	MaxCopies
▶	Daily	4	PM	0	0	5
*						

VIEW PATH:



	BackupPath	ChangedDate	ChangedTime
▶	F:\	2008/03/29	18:0:39
*			

LOG FILE:


Data Manager - [viewLogFile]
Schedule Path Recovery View Exit
Schedule
Path
Recovery
View
Exit

```
[illegible] Schedule Modified: Daily Time 11 AM, Max Copies: 1
[illegible] Schedule Modified: Daily Time 11 AM, Max Copies: 1
[illegible] Backup Path Changed: [illegible] Backup
[illegible] Schedule Modified: Daily Time 11 AM, Max Copies: 1
[illegible] Schedule Modified: Daily Time 11 AM, Max Copies: 1
[illegible] Schedule Modified: Daily Time 12 AM, Max Copies: 1
[illegible] Schedule Modified: Daily Time 12 AM, Max Copies: 1
[illegible] Backup Path Changed: [illegible]
[illegible] Backup Path Changed: [illegible] New Folder [illegible]
[illegible] Schedule Modified: Daily Time 11 AM, Max Copies: 1
[illegible] Schedule Modified: Daily Time 11 AM, Max Copies: 1
[illegible] Schedule Modified: Daily Time 11 AM, Max Copies: 1
[illegible] Schedule Modified: Daily Time 2 PM, Max Copies: 1
[illegible] Backup Path Changed: [illegible] Backup
[illegible] Schedule Modified: Daily Time 11 AM, Max Copies: 1
[illegible] Schedule Modified: Daily Time 11 AM, Max Copies: 1
[illegible] Schedule Modified: Daily Time 11 AM, Max Copies: 1
[illegible] Backup Path Changed: [illegible]
[illegible] Backup Path Changed: [illegible]
[illegible] Schedule Modified: Weekly Time 12 AM, Date: Monday Max Copies: 1
[illegible] Schedule Modified: Daily Time 11 AM, Max Copies: 1
[illegible] Schedule Modified: Daily Time 11 AM, Max Copies: 1
[illegible] Backup Path Changed: [illegible] Backup
[illegible] Backup Path Changed: [illegible]
[illegible] Schedule Modified: Daily Time 11 AM, Max Copies: 1
```

REFERENCES:

1. STEVE HARRIS AND ROBMACDONALD,
"WEB DEVELOPMENT WITH VB.NET" -APRESS

2. MATTHEW MACDONALD,
"MICROSOFT VISUAL BASIC.NET PROGRAMMER'S COOKBOOK"
-TATA McGraw HILL EDITION

3. CHRIS GOODE, JOHN KAUFFMAN
"BEGINNING ASP.NET 1.0 WITH VISUAL BASIC.NET"
-WROX PROGRAMMER TO PROGRAMMER

4. DOUGLAS O.REILLY,
"DESIGNING MICROSOFT ASP.NET APPLICATIONS"
-TATA McGraw HILL EDITION

CHAPTER 9
CONCLUSION

It is concluded that the application works well and satisfy the users. The application is tested very well and errors are properly debugged. The site is simultaneously accessed from more than one system. Simultaneous login from more than one place is tested.

The site works according to the restrictions provided in their respective browsers. Further enhancements can be made to the application. so that the web site functions very attractive and useful manner than the present one. The speed of the transactions become is very good.