

FILE SECURITY SYSTEM USING CRYPTOGRAPHY

Submitted in partial fulfillment of the requirements
for the award of the degree of

BACHELOR OF SCIENCE IN COMPUTER SCIENCE

Submitted by

Name : SUJITHA.R

Reg.No : 2122K1434

Under the Guidance of

Mrs.K.RAMA , M.C.A.,M.Phil.

Department of Computer Science



DEPARTMENT OF COMPUTER SCIENCE

PIONEER COLLEGE OF ARTS AND SCIENCE

(Affiliated to Bharathiar University)

Coimbatore-47

2023 - 2024



DECLARATION

DECLARATION

I hereby declare that this Project entitled

“FILE SECURITY SYSTEM USING CRYPTOGRAPHY

Submitted to Pioneer college of Arts and Science , Coimbatore – 47

is partial fulfilment of the requirement for the degree

BACHELOR OF COMPUTER SCIENCE

is a record of the original project work done by me during the period of

my study 2021 – 2024 at Pioneer college of Arts and Science ,

Coimbatore – 47

Under the Guidance of

Mrs.K.RAMA , M.C.A.,M.Phil.,

(Department of Computer Science)

Pioneer college of Arts and Science ,

Coimbatore – 47

SUJITHA.R

(2122K1434)



JOENS '16

CERTIFICATE

CERTIFICATE

This is to certify that this project work entitled

“FILE SECURITY SYSTEM USING CRYPTOGRAPHY”

is a bonafide record of Project work done by

NAME : SUJITHA.R

REG.NO : 2122K1434

is submitted for the award of the degree of

BACHELOR OF COMPUTER SCIENCE

Viva Voce examination held on at

PIONEER COLLEGE OF ARTS AND SCIENCE

Coimbatore – 47

During the academic year 2023 - 2024

INTERNAL GUIDE

Mrs.K.RAMA,M.C.A.,M.Phil.,

HEAD OF THE DEPARTMENT

Mrs.K.VASANTHI,M.C.A.,M.Phil.,(Ph.D).,

**COUNTER SIGNED
(PRINCIPAL)**

INTERNAL EXAMINER

EXTERNAL EXAMINER



ACKNOWLEDGEMENT

ACKNOWLEDGEMENT

At the outset I record my profound thanks to **Smt.APARNARAJKUMAR** Secretary , Pioneer college of Arts and Science for permitting me to carry my Project work.

I place it on record my gratefulness and convey my thanks to our respected Principal, **Mr.P.RAJAPPAN, BE.,M.B.A.,M.Phil.,PGDCA** for the valuable encouragement and direction extended towards the completion of my project.

I warmly extend my thanks to **Mrs.K.VASANTHI,M.C.A.,M.Phil.,(Ph.D),** Head , Department of Computer Science for her timely support.

I specially thank my beloved Tutor **Mr.K.MANICKARAJ,M.C.A.,M.Phil.,** Department of Computer Science, for his valuable guidance and encouragement throughout the project.

It is special privilege to convey my everlasting thanks to my internal guide **Mrs.K.RAMA,M.C.A.,M.Phil.,** Department of Computer Science, for her immense help, insightful suggestions, valuable advice, and constructive comments.

I would like to express my sincere thanks to my family members , friends and all other for their help and support.



SYNOPSIS

SYNOPSIS

The project “FILE SECURITY SYSTEM USING CRYPTOGRAPHY” is designed using Microsoft Visual Studio.Net 2008 as front end, which works in .Net framework version 3.5. The coding language used is Visual Basic .Net and Back End Sql Server 2005.

The project aims at securing the important documents by collecting them into one project file with or without encrypting all files. During the project file creation. Files are added, password entry made to individual files and any/all files can be encrypted using Triple DES encryption mechanism and stored as single project file. The project may also be password protected. Anytime the archive can be viewed, edited i.e., file addition, removal is done.

If all files are not password protected, then all files may get extracted at one time; otherwise we can extract one by one. If the file is password protected, then it is extracted after password submission. Thus the project aims in protecting the files and avoid unauthorized access.



CONTENTS

TABLE OF CONTENTS

S.NO	CONTENT	PAGE.NO
	ACKNOWLEDGEMENT	
	SYNOPSIS	
1	INTRODUCTION	
	1.1 OVERVIEW	1
2	SYSTEM REQUIREMENTS	
	2.1 HARDWARE REQUIREMENTS	2
	2.2 SOFTWARE REQUIREMENTS	2
	2.2.1 SOFTWARE DESCRIPTION	3
3	SYSTEM STUDY	
	3.1 EXISTING SYSTEM	10
	3.1.1 LIMITATIONS	10
	3.2 PROPOSED SYSTEM	10
	3.2.1 ADVANTAGES	11
4	SYSTEM DESIGN AND DEVELOPMENT	
	4.1 INPUT DESIGN	12
	4.2 OUTPUT DESIGN	13
	4.3 DATA BASE DESIGN	13
	4.4 MODULES	13

4.4.1	DESCRIPTIONS OF MODULE	14
5	SYSTEM TESTING AND IMPLEMENTATION	
5.1	OBJECTIVE OF TESTING	16
5.2	TESTING METHODS	16
5.2.1	UNIT TESTING	17
5.2.2	INTEGRATION TESTING	17
5.2.3	SYSTEM TESTING	17
5.2.4	ACCEPTANCE TESTING	18
5.3	SYSTEM IMPLEMENTATIONS	18
6	CONCLUSION	19
	BIBILIOGRAPHY	
	APPENDICES	
	A . DATA FLOW DIAGRAM	
	B . ENTITY RELATIONSHIP DIAGRAM	
	C .TABLE DESIGN	
	D. FORM DESIGN	
	E .SAMPLE CODING	



INTRODUCTION

1. INTRODUCTION

1.1 OVERVIEW

The project aims at protecting a single files or group of files in a system. All type of files can be protected using this project. First the user should select the files and give the password for the files to be protected. Files can also be encrypted using Triple DES Algorithm to prevent unauthorized access. After selecting the password, a project file is created. To extract the project, the project file should be opened and the password should be given for files to be extracted.

If we want to add any new files to the project, then add project option can be used to add new files to the existing project. This will erase the old project and creates a new project with updated information. Password protection can be applied to single files or multiple of files. If there is no password for any files, then extract all option can be performed to extract all the files to the specified folder.



SYSTEM REQUIREMENTS

2.SYSTEM REQUIREMENTS

2.1 Hardware Requirements

- SYSTEM : I3
- HARD DISK : 50 GB
- MONITOR : LCD screen
- MOUSE : 2 button optical mouse
- RAM : 8GB
- KEYBOARD : Standard Keyboard

2.2 Software Requirements

- Operating system : Windows 10
- Front End : VB.net
- Back End : Sql server

2.2.1 SOFTWARE DESCRIPTION

What is .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.

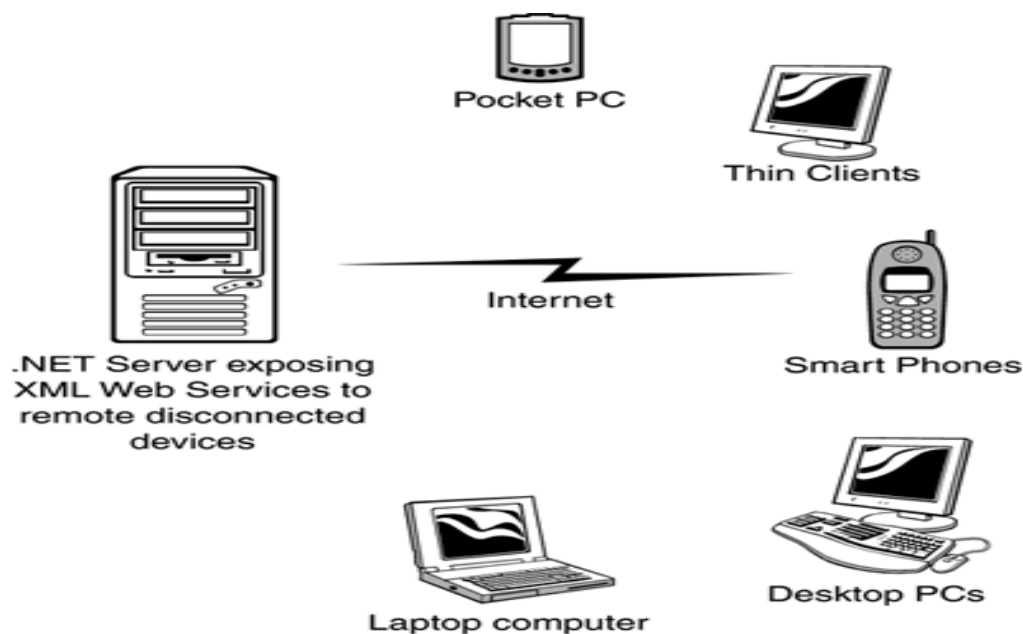
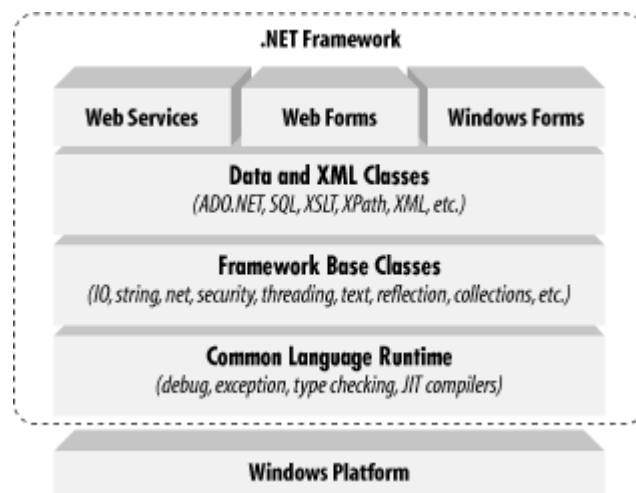


Figure 1.1

.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.



Conceptually, the CLR and the JVM are similar in that they are both runtime infrastructures 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 JVM executes bytecode, so it can, in principle, support many languages, too. Unlike Java's bytecode, though, CIL is never interpreted. Another conceptual difference between the two infrastructures is that Java code runs on any platform with a JVM, whereas .NET code runs only on platforms that support the CLR. In April, 2003, the International Organization for Standardization and the International Electrotechnical Committee (ISO/IEC) recognized a functional subset of the CLR, known as the Common Language Interface (CLI), as an

international standard. European standards organization, opens the way for third parties to implement their own versions of the CLR on other platforms, such as Linux or Mac OS X. For information on third-party and open source projects working to implement the ISO/IEC CLI and C# specifications

The layer on top of the CLR is a set of framework base classes. This set of classes is similar to the set of classes found in STL, MFC, ATL, or Java. These classes support rudimentary input and output functionality, string manipulation, security management, network communications, thread management, text management, reflection functionality, collections functionality, as well as other functions. On top of the framework base classes is a set of classes that extend the base classes to support data management and XML manipulation. These classes, called ADO.NET, support persistent data management—data that is stored on backend databases. Alongside the data classes, the .NET Framework supports a number of classes to let you manipulate XML data and perform XML searching and XML translations.

Classes in three different technologies (including web services, Web Forms, and Windows Forms) extend the framework base classes and the data and XML classes. Web services include a number of classes that support the development of lightweight distributed components, which work even in the face of firewalls and NAT software. These components support plug-and-play across the Internet, because web services employ standard HTTP and SOAP. Windows Forms support a set of classes that allow you to develop native Windows GUI applications. You can think of these classes collectively as a much better version of the MFC in C++ because they support easier and more powerful GUI development and provide a common, consistent interface that can be used in all languages.

The Common Language Runtime

At the heart of the .NET Framework is the common language runtime. The following list describes some of the benefits the common language runtime gives you:

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

You might expect all those features, but this has never been possible using Microsoft development tools. Figure 1.3 shows where the common language runtime fits into the .NET Framework.

Figure 1.3. The common language runtime and the .NET Framework.

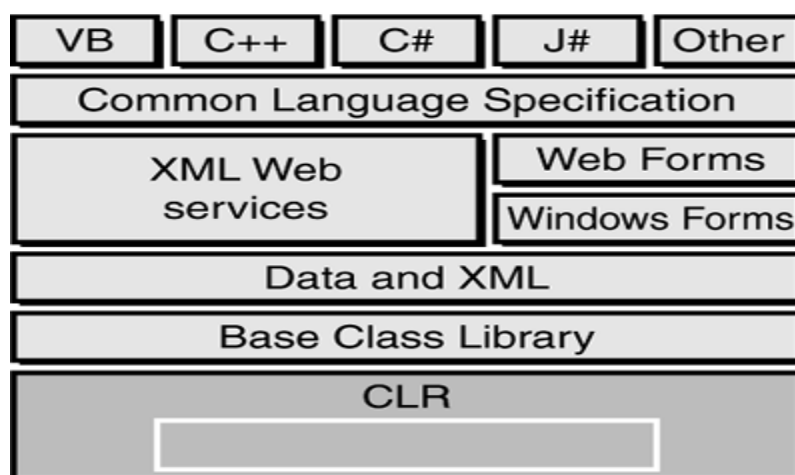
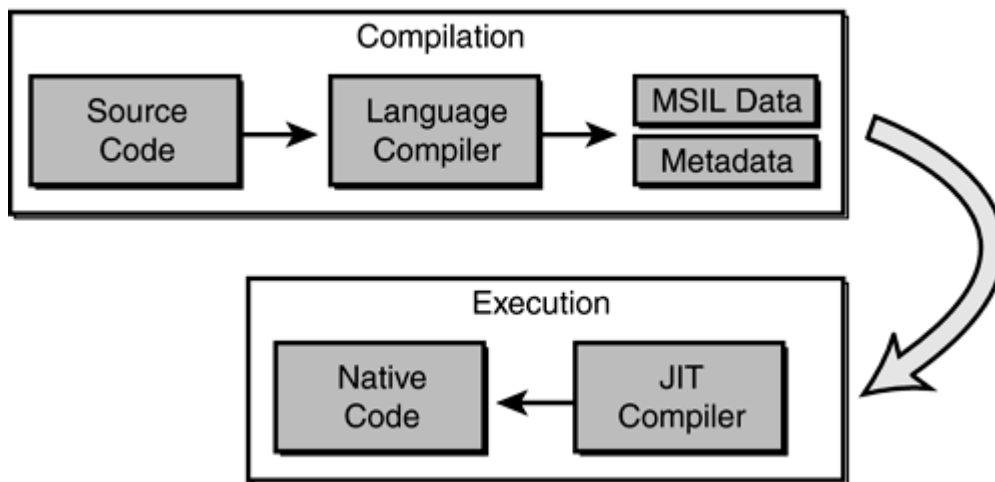


Figure 1.5. The JIT process and verification.



When code is JIT compiled, the common language runtime checks to make sure that the IL is correct. The rules that the common language runtime uses for verification are set forth in the Common Language Specification (CLS) and the Common Type System (CTS).

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

Each assembly has one and only one assembly manifest, and it contains all the description information for the assembly. However, the assembly manifest can be contained in its own file or within one of the assembly's modules.

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. For example, you might have a class that represents a bank account. 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. stored within the class or changes to the user interface. Continuing with the bank account example, you might have a Transfer method that transfers a balance from a checking account to a savings account, or an Alert method that warns users when their balances fall below a predetermined level.

Overview of ADO.NET

Most applications require some kind of data access. Desktop applications need to integrate with central databases, Extensible Markup Language (XML) data stores, or local desktop databases. ADO.NET data-access technology allows simple, powerful data access while maximizing system resource usage. Different applications have different requirements for data access. Whether your application simply displays the contents of a table, or processes and updates data to a central SQL server, ADO.NET provides the tools to implement data access easily and efficiently.

Microsoft SQL server

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.



SYSTEM STUDY

3.SYSTEM STUDY

3.1 Existing system

In companies, the documents or other resources need to be protected and unauthorized persons must not access i.e., view or edit the documents. The documents if misused by others may cause the theft of the valuable data between the companies. Moreover, the document if split-ted in various partitions needs to be search so that they can be found out and collectively used. In addition, in normal desktop applications the files in partitions other than (New Technology File System)NTFS file system may not be protected. Any user if the path is known can access the resources. Hence if a system such that combining the industry secure documents, password protecting and encrypting them is available, then they can be secured from theft. Hence the new system is being designed to achieve the result

3.1.1 Limitations of the existing system

- It is not possible to set password all kind of files.
- Encryption and decryption is not possible
- Requires huge manpower.
- Preventing data loss is too difficult.
- Required output is not possible.
- all files are not password protected

3.2 Proposed system

The new system i.e., the software developed is capable of adding the document or other files, may or may not password protected individually, either encrypted or not and collectively written as a project file. Even the project file is password protected if necessary so that other users can not open the project in this application. After archiving them to single file, it can be viewed and extracted either individual or all files.

If none of the file is password protected, they can be extracted whole otherwise it is must to provide password to extract a particular file. The file is encrypted through the Triple DES (Data Encryption Standard Algorithm for safety. The Dotnet System.Security.Cryptography namespace is used for encryption and structure, file handling concepts are used for archiving the project. Since the project is created and edited as binary, the content may not be viewed in text editors easily.

3.2.1 Advantages:

- In the proposed system all the drawbacks of the existing system are carefully analyzed and all the pitfalls are resolved in an effective and efficient manner.
- The Proposed System has been made with better user interactive screens. The system is much user friendly and the end user will find it easy to do operations.
- Encryption and decryption is possible
Protect all file in the folder



SYSTEM DESIGN AND DEVELOPMENT

4. SYSTEM DESIGN AND DEVELOPMENT

4.1 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.**

Add Files

This method is used to add new files to project list. Single or multiple files can be added to this project list

Remove Files

This method is used to delete the files from the project list. Single or multiple files can also be deleted from the project list

Sort Files

This method is used sort the files in a specified order. Files can be sorted in both ways i.e. in ascending or descending order. Files can also be placed according to the user specified order.

Input Password

This method is used to obtain the password for both the file and project. Each password is stored in two different variables.

Encrypt File

This method is used to check whether the project should be encrypted by using triple des algorithm. Encryption password must be supplied to encrypt the project.

4.2 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. 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.

4.3 DATABASE DESIGN

The database design is a must for any application developed especially more for the data store projects. Since the application proper handling of the table is a must. In the project, login table is designed to be unique in accepting the username and the length of the username and password should be greater than zero. The different users view the data in different format according to the privileges given.

4.4 MODULES

File Security System is divided into six modules

- Authentication
- Accumulation of Files
- Security Module
- Encrypt Data
- File Format Conversion
- Extracting Files

4.4.1 DESCRIPTION OF MODULES

Authentication

In this module the authorized person only can enter in this security process. The data are stored in a database for the user verification.

Accumulation of Files

In this module, the files are accumulated for the privacy of data. We can add single files or multiple files to the project. Files can also be included later after creating the project. In this case, the old projects file will be deleted and the new project file will be created when including some files to your project list.

When adding single or multiple files, password can be given to each file to protect the access of unknown users from viewing the file. Any type of files can be added to the project by selecting the files.

Security Module

Security Module is implemented using two ways in this project. First way is to provide security to single files or multiple files by specifying the password for each file. The second way is to provide security to the whole project by encrypting all the project files using Triple DES Algorithm. Encryption can be enabled or disabled for the particular project by enabling or disabling the encryption option for the project. When encrypting the project, the filename extension is changed into a new extension and saved.

The contents of the file are also encrypted so the unknown users can't access the file. If anybody needs to open the project file, then he needs to know the authenticated information such as password for each file to be accessed. And the project application cannot be opened in any other application after encrypting the project.

Encrypt Data

This module is used to display the encrypted files to the user. All the information about the projects is stored in the header information and all the details about the files are stored in the file header. The Encrypted project file can be opened only if the password is known and can be viewed only in the File protector application. After supplying the correct password, the protected files are listed in the project list window.

Here, the user can add new files to the project or delete existing files from the project. The files are displayed by reading project header information and file header information from the project file.

File Format conversion

The file format is converted to some other format. So the attacker cannot find the original format of a source file. For example the text file is ends with extension as .txt. So the hacker can easily find the format of a source. In our system the extension is to be changes as some other formats like .ccs etc.,

Extracting File

Extracting data module is used to extract files from the project window to the specified location. If the file is protected using password, then the exact password should be supplied to the password input window. After providing the correct password, the files are extracted into the specified folder. If the files are not password protected, then single files or multiple files can be extracted at the same time using the extract all option available under the project menu.



**SYSTEM TESTING AND
IMPLEMENTATION**

5. SYSTEM TESTING AND MAINTANANCE

5.1 OBJECTIVE OF TESTING

The most important phase in system development life cycle is system testing. The number and nature of errors in a newly designed system depends on the system specifications and the time frame given for the design.

A newly designed system should have all the subsystems working together, but in reality each subsystems work independently. During this phase, all the subsystems are gathered into one pool and tested to determine whether it meets the user requirements.

Testing is done at two level -Testing of individual modules and testing the entire system. During the system testing, the system is used experimentally to ensure that the software will run according to the specifications and in the way the user expects. Each test case is designed with the intent of finding errors in the way the system will process it.

5.2 TESTING METHODS

Testing plays a very critical role in determining the reliability and efficiency of software and hence is a very important stage in software development. Software testing is done at different levels. They are the unit testing and system testing which comprises of integration testing and acceptance testing.

- ✓ Unit Testing
- ✓ System Testing
- ✓ Integration Testing
- ✓ Acceptance Testing

5.2.1 UNIT TESTING

This is the first level of testing. The different modules are tested against the specifications produced during the integration. This is done to test the internal logic of each module. Those resulting from the interaction between modules are initially avoided. The input received and output generated is also tested to see whether it falls in the expected range of values. Unit testing is performed from the bottom up, starting with the smallest and lowest modules and proceeding one at a time.

The units in a system are the modules and routines that are assembled and integrated to perform a specific function. The programs are tested for correctness of logic applied and detection of errors in coding. Each of the modules was tested and errors are rectified. They were then found to function properly.

5.2.2 INTEGRATION TESTING

In integration testing, the tested modules are combined into sub-systems, which are then tested. The goal of integration testing to check whether the modules can be integrated properly emphasizing on the interfaces between modules. The different modules were linked together and integration testing done on them.

5.2.3 SYSTEM TESTING

The integration of each module in the system is checked during this level of testing. The objective of system testing is to check if the software meets its requirements. System testing is done to uncover errors that were not found in earlier tests. This includes forced system failures and validation of total system as the user in the operational environment implements it. Under this testing, low volumes of transactions are generally based on live data. The total system is also tested for recovery after various major failures to ensure that no data are lost during the breakdown.

5.2.4 ACCEPTANCE TESTING

The objective of the acceptance test is to tell the user about the validity and reliability of the system. It verifies whether the system operates as specified and the integrity of important data is maintained. User motivation is very important for the successful performance of the system.

All the modules were tested individually using both test data and live data. After each module was ascertained that it was working correctly and it had been "integrated" with the system. Again the system was tested as a whole. We hold the system tested with different types of users. The System Design, Data Flow Diagrams, procedures etc. were well documented so that the system can be easily maintained and upgraded by any computer professional at a later

Acceptance testing is done with live data provided by the client to ensure that the software works satisfactorily. This test focuses on the external behavior of the system. Data was entered and acceptance testing was performed.

5.3 SYSTEM IMPLEMENTATION

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective. The implementation stage involves careful planning, investigation of the existing system and its constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods. Implementation is the process of converting a new system design into operation. It is the phase that focuses on user training, site preparation and file conversion for installing a candidate system. The important factor that should be considered here is that the conversion should not disrupt the functioning of the organization.



CONCLUSION

6. CONCLUSION

It is concluded that the application works well and satisfy the needs. The application is tested very well and errors are properly debugged. It also acts as the sharing of network files to the valuable resources. This system is user friendly so everyone can use easily. Proper documentation is provided. The end user can easily understand how the whole system is implemented by going through the documentation. The system is tested, implemented and the performance is found to be satisfactory. All necessary output is generated. Thus, the project is completed successfully.

FUTURE AND ENHANCEMENT

There is scope for future development of this project. The world of computer fields is not static; it is always subject to be dynamic. The technology which is famous today becomes outdated the very next day. To keep abstract of technical improvements, the system may be further refined. So, it is not concluded. Yet it will improve with further enhancements.

Enhancements can be done in an efficient manner. We can even update the same with further modification establishment and can be integrated with minimal modification. Thus the project is flexible and can be enhanced at anytime with more advanced features.



BIBILIOGRAPHY

BIBLIOGRAPHY

BOOKS:

1. Alistair McMonnies, “**Object-oriented programming in Visual Basic.NET**”, Pearson Education, and ISBN: 81-297-0649-0, First Indian Reprint 2004.
2. Jittery R.Shapiro, “**The Complete Reference Visual Basic .NET**” Edition 2002, Tata McGraw-Hill, Publishing Company Limited, New Delhi.
3. Robert D.Schneider, Jetty R.Garbus, “**Optimizing SQL Server**”, Second Edition, Pearson Education Asia, ISBN: 981-4035-20-3

WEBSITES:

- <http://www.microsoft.com/dotnet/visual basic>
- <http://www.dotnetheaven.com>
- <http://www.apache.org>
- <http://www.w3schools.com>

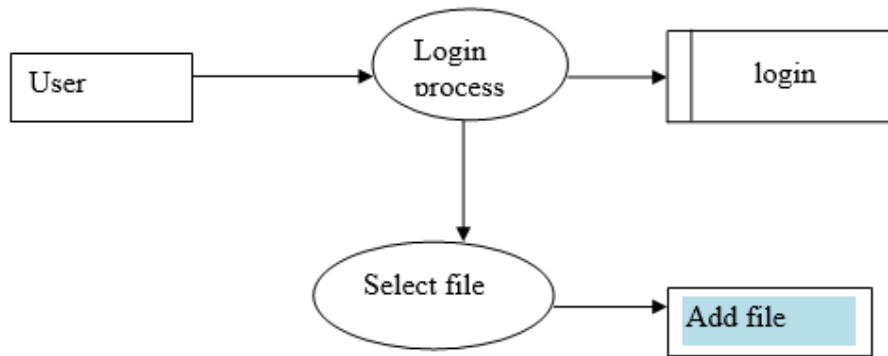


JOENS '16

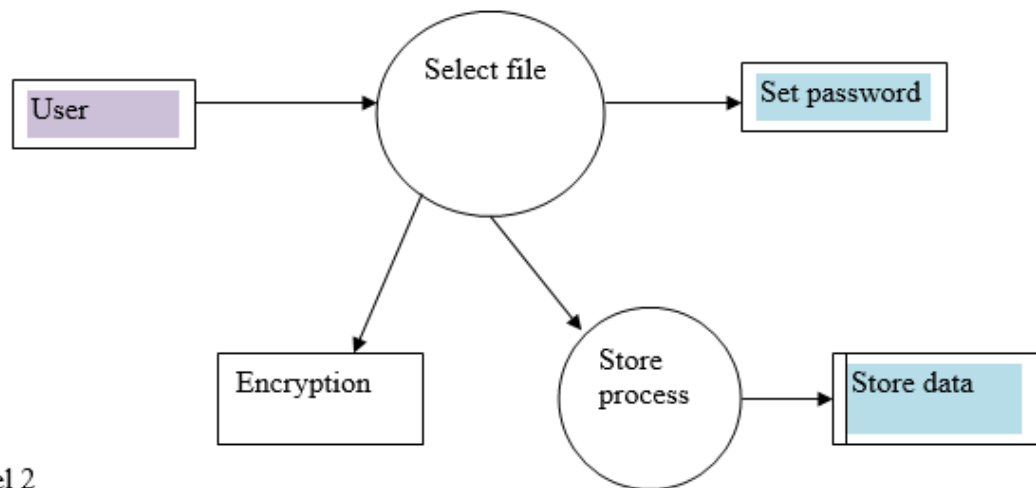
APPENDICES

A.DATA FLOW DIAGRAM

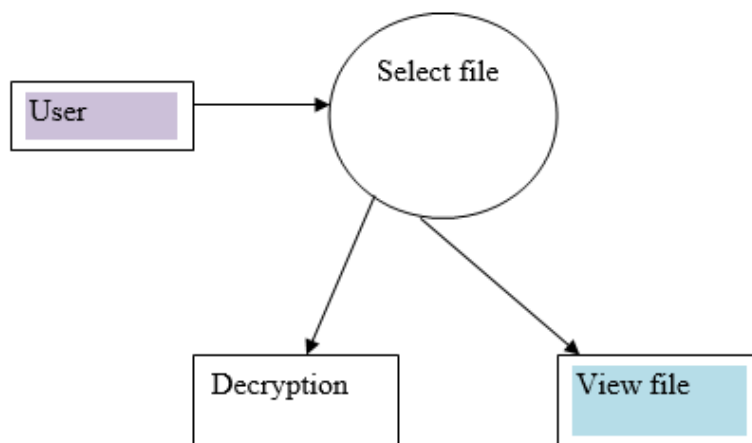
Level 0



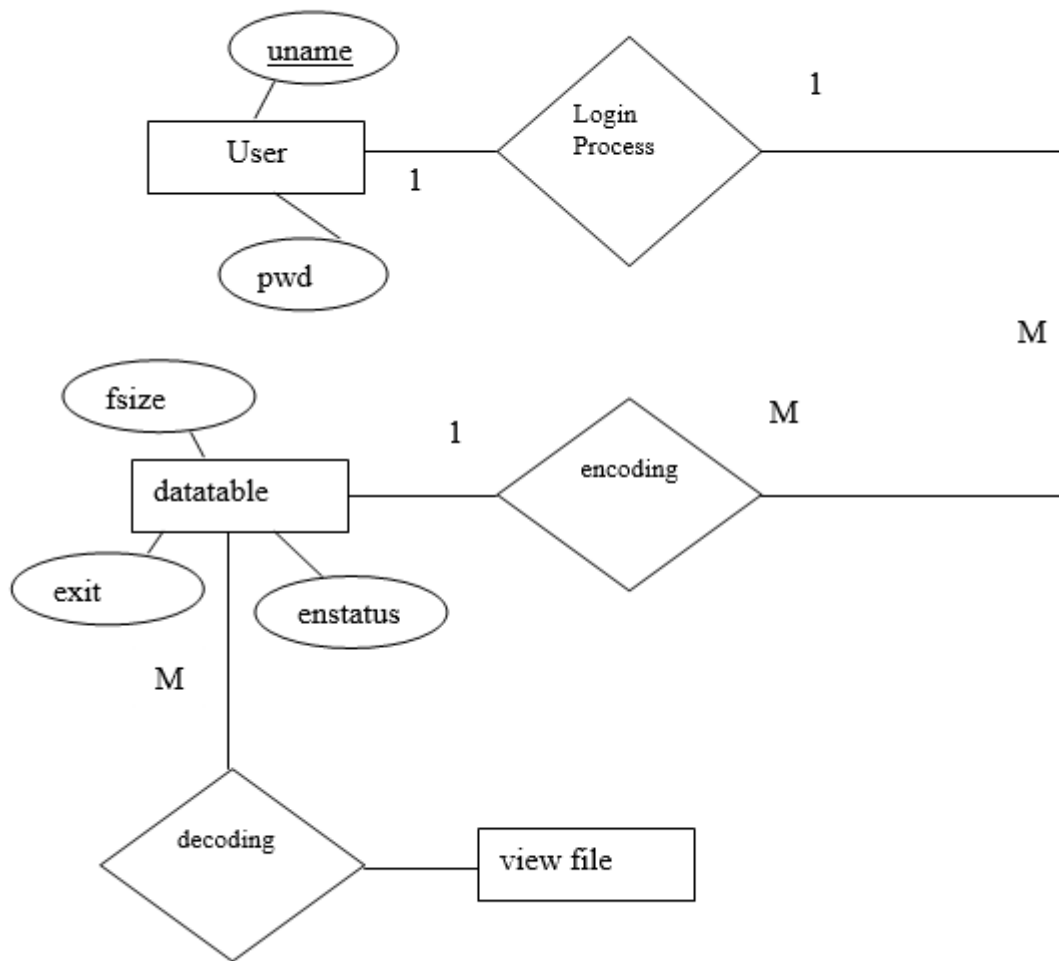
Level 1



Level 2



B.ENTITY RELATIONSHIP DIAGRAM



C.TABLE DESIGN

1.Datatable

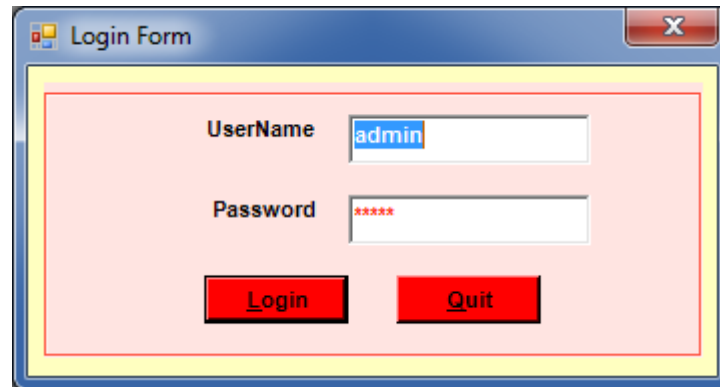
S.NO	COLUMN NAME	DATA TYPE	LENGTH	ALLOW NULLS
1	filesize	int	4	yes
2	extension	varchar	15	yes
3	filepath	varchar	200	yes
4	offset	int	4	yes
5	encstatus	int	4	yes
6	passstate	int	4	yes

2.Admin

S.NO	COLUMN NAME	DATA TYPE	LENGTH	ALLOW NULLS
1	User Name	nvarchar	50	yes
2	Password	nvarchar	50	yes

D.FORM DESIGN

1.LOGIN FORM DESIGN



A screenshot of a Windows-style application window titled "Login Form". The window has a blue title bar with a close button (X) in the top right corner. The main content area has a light pink background and is enclosed in a yellow border. It contains two text input fields: "UserName" with the text "admin" and "Password" with masked characters "*****". Below the fields are two red buttons with black text: "Login" and "Quit".



New Project

Files

C:\Users\Ravi\Desktop\123.txt

↑

↓

+

×

PASSWORD STATUS 1

C:\Users\Ravi\Desktop\123.txt

Enter Password ***

Encrypt

C:\Users\Ravi\Desktop\123.txt

☒ Encrypt

ENCRYPT STATUS 1

Enter Project Password (Optional)

**

Save Project

C:\Users\Ravi\Desktop\Project11.FP

Create Project File

Create Project

Save

View

VIEW DETAILS

NewDataSet:						
	filesize	extention	filepath	offset	encstatus	passstatus
▶	16	.csd	C:\Users\Ravi	326	1	1
*						

Close

E.SAMPLE CODING

Add Project

```
Dim PH As ProjectHeader
Dim FH() As FileHeader
'Dim Enc() As Boolean
'Dim Pass() As Boolean
'Dim PassText() As String
Dim si As Short

Private Sub Check1_CheckStateChanged(ByVal eventSender As
System.Object, ByVal eventArgs As System.EventArgs) Handles
Check1.CheckStateChanged
    If Check1.CheckState =
System.Windows.Forms.CheckState.Checked Then
        'Combo2.ItemData(Combo2.ListIndex) = 1
        VB6.SetItemString(Combo4, Combo2.SelectedIndex,
"1")
    Else
        'Combo2.ItemData(Combo2.ListIndex) = 0
        VB6.SetItemString(Combo4, Combo2.SelectedIndex,
"0")
    End If
End Sub

Private Sub Combo1_SelectedIndexChanged(ByVal eventSender As
System.Object, ByVal eventArgs As System.EventArgs) Handles
Combo1.SelectedIndexChanged
    Text1.Text = VB6.GetItemString(Combo3,
Combo1.SelectedIndex)
End Sub
```



```

Private Sub Combo2_SelectedIndexChanged(ByVal eventSender As
System.Object, ByVal eventArgs As System.EventArgs) Handles
Combo2.SelectedIndexChanged
    If VB6.GetItemString(Combo4, Combo2.SelectedIndex) =
"1" Then
        Check1.CheckState =
System.Windows.Forms.CheckState.Checked
    Else
        Check1.CheckState =
System.Windows.Forms.CheckState.Unchecked
    End If
End Sub

Private Sub Command1_Click(ByVal eventSender As System.Object,
ByVal eventArgs As System.EventArgs) Handles Command1.Click
    ChangeOrder(List1, -1)
    ChangeOrder(Combo1, -1)
    ChangeOrder(Combo2, -1)
    ChangeOrder(Combo3, -1)
    ChangeOrder(Combo4, -1)
    ChangeOrder(Combo5, -1)
End Sub

Private Sub Command2_Click(ByVal eventSender As
System.Object, ByVal eventArgs As System.EventArgs) Handles
Command2.Click
    ChangeOrder(List1, 1)
    ChangeOrder(Combo1, 1)
    ChangeOrder(Combo2, 1)
    ChangeOrder(Combo3, 1)
    ChangeOrder(Combo4, 1)
    ChangeOrder(Combo5, 1)
End Sub

```

```
Private Sub Command3_Click(ByVal eventSender As System.Object,  
ByVal eventArgs As System.EventArgs) Handles Command3.Click
```

```
Dim i As Object
```

```
    CommonDialog1.Filter = "All Files (*.*)|*.*"
```

```
    CommonDialog1.ShowOpen()
```

```
    If CommonDialog1.FileName <> "" Then
```

```
        For i = 0 To List1.Items.Count - 1
```

```
            If VB6.GetItemString(List1, i) =
```

```
CommonDialog1.FileName Then
```

```
                MsgBox("Already Added",
```

```
MsgBoxStyle.Information)
```

```
                Exit Sub
```

```
            End If
```

```
Next
```

```
    List1.Items.Add(CommonDialog1.FileName)
```

```
    Combo3.Items.Add("")
```

```
    Combo4.Items.Add("")
```

```
    Combo5.Items.Add("")
```

```
    RefillComboBoxes()
```

```
    'ReDim Preserve Enc(List1.ListCount - 1)
```

```
    'ReDim Preserve Pass(List1.ListCount - 1)
```

```
    'ReDim Preserve PassText(List1.ListCount - 1)
```

```
End If
```

```
End Sub
```

```
Private Sub Command4_Click(ByVal eventSender As System.Object,  
ByVal eventArgs As System.EventArgs) Handles Command4.Click
```

```
    Dim i As Short
```

```
    For i = 0 To List1.Items.Count - 1
```

```
        If List1.GetSelected(i) Then
```

```
            List1.Items.RemoveAt(i)
```

```
            Combo3.Items.RemoveAt(i)
```

```
            Combo4.Items.RemoveAt(i)
```

```
            Combo5.Items.RemoveAt(i)
```

```
        Exit For          29
```

```

        End If
    Next
    RefillComboBoxes()
End Sub

Public Sub ChangeOrder(ByRef List1 As ListBox, ByVal mode As
Short)
    On Error GoTo down
    Dim arr() As String
    Dim i As Short
    Dim x As String
    Dim tmp As String
    If si >= 0 Then
ReDim arr(List1.Items.Count - 1)
For i = 0 To List1.Items.Count - 1
    arr(i) = VB6.GetItemString(List1, i)
Next
    tmp = arr(si)
    arr(si) = arr(si + mode)
    arr(si + mode) = tmp
    List1.Items(si) = arr(si)
    List1.Items(si + mode) = arr(si + mode)
    List1.SelectedIndex = si
    'List1.Selected(si) = True
    List1.Focus()
End If

```