DOCUMENT COLLATION TOOL

By

M. Uma Maheswari

Reg. No. 71204621057

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 APPLICATIONS

JULY, 2007

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE – 641006.

DEPARTMENT OF COMPUTER APPLICATIONS

BONAFIDE CERTIFICATE

Certified that this project report titled **Document Collation Tool** is the bonafide work of **Ms. M. Uma Maheswari (Reg.no7120462157)** 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.

Project Guide **Head of Department**

Submitted for the University Examination held on 03-07-2007

Internal Examiner **External Examiner** 3/7/07

CARITOR

Vision > Valu

May 15th, 2007

To whomsoever it May Concern

This is to inform you that **M.UMA MAHESWARI** has successfully completed her project assignment titled **DOCUMENT COLLATION TOOL** as a part of **MCA curriculum.**

As a Project Trainee, she started this project on **December 18, 2006** and completed it on **May 15, 2007**

Please note, as per the company's policies and practices, the company retains ownership of the intellectual property rights concerning work undertaken during projects and disclosure of the source code and any other relevant information or data out of the organization is strictly prohibited.

M.UMA MAHESWARI designated, as project trainee will not be delivering the respective source code pertaining to her project.

For Caritor (India) Pvt Ltd,

BHAVANI DEVAIAH
MANAGER – HUMAN RESOURCES

ABSTRACT

Creating documents are an integral part of any organizations day-to-day activities. Documents will be created whenever there is a need to share information between two people or departments or with people external to the organization. For example, when a design for a new application needs to be sent to the customer, it will be sent as a document. When a response to a proposal is sent to a prospective customer, it will be sent as a document. However, none of these documents will be created by a single person. Different people will work on different section of the document. In such cases, synchronizing the changes of all users is a never-ending exercise. The proposed Document Collation Tool will play a major role in solving this problem.

The Document Collation tool takes a two step approach to solve the above mentioned problem. Firstly, it creates separate documents for the individual sections so that editors can work independently of each other. Secondly, after the editors have changed their respective documents, it will also do the consolidation of the different sections to come up with the consolidated document.

The input to the Document Collation tool will be the Table of Contents of the expected document. Given this as input, the Document Collation Tool will create as many separate documents as there are entries in the Table of Contents (TOC). Now, different sections can be changed independently by the respective editors. Whenever a new version of a section is uploaded, the Document Collation Tool will pick up the latest versions of all the sections, and create the consolidated document.

The tool would be implemented on a Sharepoint Site. Sharepoint is a Microsoft product running on windows that allows creating highly dynamic websites. There are out-of-the box lists & libraries provided by Sharepoint which are very helpful in maintaining documents. The Document Collation tool attempts to leverage these features and extend them to create a synchronized multi-user environment.

ACKNOWLEDGEMENT

I wish to express my sincere thanks to **Dr. JOSEPH V. THANIKAL Ph.D.**, Principal, Kumaraguru College of Technology, Coimbatore, for permitting me to undertake this project.

My deepest acknowledgement to **Dr. M. GURURAJAN Ph.D.,** Head of the Department, Computer Applications, Kumaraguru College of Technology, Coimbatore, for his timely help and guidance throughout this project.

I also express my thanks to **Mr. A. MUTHUKUMAR M.Sc., M.C.A, M.Phil.,** Project Coordinator, Assistant professor, Department of Computer Applications, Kumaraguru College of Technology, Coimbatore, for encouraging me to do this project.

I am greatly indebted to my guide **Mrs. P. PARAMESWARI M.C.A, M.Phil.,** Senior Lecturer, Department of Computer Applications, Kumaraguru College of Technology, Coimbatore, for her valuable guidance and encouragement at every stage of this project work

I express my sincere thanks to **Mr. PAWAN ARYA**, Project Leader, CARITOR (INDIA) PRIVATE LIMITED, for his support and assistance at various level of my project work.

Finally, I owe a lot to my beloved parents and family members and to my department staffs without their help and co-operation the project would not have taken a final shape.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

Creating documents are an integral part of any organizations day-to-day activities. Documents will be created whenever there is a need to share information between two people or departments or with people external to the organization. For example, when a design for a new application needs to be sent to the customer, it will be sent as a document. When a response to a proposal is sent to a prospective customer, it will be sent as a document. However, none of these documents will be created by a single person. Different people will work on different section of the document. In such cases, synchronizing the changes of all users is a never-ending exercise. The proposed Document Collation Tool will play a major role in solving this problem.

The Document Collation tool takes a two step approach to solve the above mentioned problem. Firstly, it creates separate documents for the individual sections so that editors can work independently of each other. Secondly, after the editors have changed their respective documents, it will also do the consolidation of the different sections to come up with the consolidated document.

The input to the Document Collation tool will be the Table of Contents of the expected document. Given this as input, the Document Collation Tool will create as many separate documents as there are entries in the Table of Contents (TOC). Now, different sections can be changed independently by the respective editors. Whenever a new version of a section is uploaded, the Document Collation Tool will pick up the latest versions of all the sections, and create the consolidated document.

The tool would be implemented on a Sharepoint Site. Sharepoint is a Microsoft product running on windows that allows creating highly dynamic websites. There are out-of-the box lists & libraries provided by Sharepoint which are very helpful in maintaining documents. The Document Collation tool attempts to leverage these features and extend them to create a synchronized multi-user environment.

1.2 ORGANIZATION PROFILE

Caritor, incorporated and headquartered in USA, is a global IT Consulting & Systems Integration firm that delivers high-quality IT services to leading clients around the world. It has been playing the role of a trusted IT partner to the clients since 1993 by helping them translate their IT vision into solid, measurable value.

The word Caritor has its origin in two languages. "Cari" is derived from the Latin word "Caritas", signifying the concept of caring. Cari demonstrates relationship building through customer care and value creation. It stands for our values-driven culture. The word "tor" has Spanish origins and is derived from the word "Toreador", signifying the boldness and strength of a bull fighter. It symbolizes the can-do spirit and entrepreneurial drive of our organization. It brings out the ability and courage to take on heavy odds and win. The complete name "Caritor" signifies the aggressiveness in being a caring partner for their customers.

Caritor offers cost-effective and intelligent IT solutions to clients in the areas of Financial Services, Communications, Retail, Manufacturing, High-Technology, Travel and Transportation and Public Sector industries. They offer IT services in the areas of Application Development, Application Management, Enterprise Business Solutions, Software Testing and Systems Integration through a global delivery model that ensures security, cost-effectiveness and quality for clients.

Caritor have firmly believed in ensuring the highest quality and security for the IT solutions that it delivers to clients. Caritor is one of the very few companies in the world to be certified at SEI-CMM Level 5, PCMM Level 5, CMMI Level 5, ISO:9001 and the BS7799 standards. As part of their continuing quality initiatives they are also rolling out Six-Sigma processes internally.

In pursuance of its quality objectives, Caritor has also embarked upon the Six Sigma Initiatives to bring about a significant magnitude of improvements to its quality processes and deliverables. This initiative is expected to result in substantial value

addition as well as measurable and quantifiable benefits to our global clientele. Caritor has institutionalized these quality practices in its comprehensive Business Management System (BMS).

The BMS is a comprehensive process framework that leverages the proven SEI-CMMi practices to create superior value for the customer. Caritor has independent teams for Quality Assurance and Process Management called SQA (Software Quality Assurance) and PMG (Process Management Group), which caters to verification, validation and process compliance to Caritor's BMS.

1.3 SYSTEM SPECIFICATION

1.3.2 Hardware Requirements:

Processor	:	Pentium IV 2.80 GHz
Monitor	:	SVGA Color
RAM	:	1 GB
Floppy Drive	:	1.44 MB
Printer	:	HP-Laser Jet

1.3.2 Software Requirements

Operating System	:	Windows 2003 server
Language	:	C#.Net,ASP.Net
Web Server / App-Server	:	Internet Information Server 6.0
Scripting Language	:	JavaScript
Markup Language	:	XML
Database	:	SQL Server 2000

1.4 PROGRAMMING ENVIRONMENT

VISUAL STUDIO.NET

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

This section contains information about some of the latest tools and technologies available in this release of Visual Studio.

Visual C#

Microsoft C# (pronounced C sharp) is a new programming language designed for building a wide range of enterprise applications that run on the .NET Framework. An evolution of Microsoft C and Microsoft C++, C# is simple, modern, type safe, and object oriented. C# code is compiled as managed code, which means it benefits from the services of the common language runtime. These services include language interoperability, garbage collection, enhanced security, and improved versioning support.

C# is introduced as Visual C# in the Visual Studio .NET suite. Support for Visual C# includes project templates, designers, property pages, code wizards, an object model, and other features of the development environment. The library for Visual C# programming is the .NET Framework.

The .NET Framework

The .NET Framework is a multi-language environment for building, deploying, and running XML Web services and applications. It consists of three main parts:

Common Language Runtime

Despite its name, the runtime actually has a role in both a component's runtime and development time experiences. While the component is running, the runtime is responsible for managing memory allocation, starting up and stopping threads and processes, and enforcing security policy, as well as satisfying any dependencies that the component might have on other components. At development time, the runtime's role changes slightly; because it automates so much (for example, memory management); the runtime makes the developer's experience very simple, especially when compared to COM as it is today. In particular, features such as reflection dramatically reduce the amount of code a developer must write in order to turn business logic into a reusable component.

Unified programming classes

The framework provides developers with a unified, object-oriented, hierarchical, and extensible set of class libraries (APIs). Currently, C++ developers use the Microsoft Foundation Classes and Java developers use the Windows Foundation Classes. The framework unifies these disparate models and gives Visual Basic and JScript programmer's access to class libraries as well. By creating a common set of APIs across all programming languages, the common language runtime enables cross-language inheritance, error handling, and debugging. All programming languages, from JScript to C++, have similar access to the framework and developers are free to choose the language that they want to use.

SharePoint provides various features

- Manage documents Version Control & Tracking
- Create/Use Collaborative Team Rooms
- Perform Full text index portal-wide & team room searches
- View announcements & participate in discussion groups
- Integrate Outlook Calendar and Inbox

- Receive alerts about recently published content
- Link to other knowledge bases and applications

Sharepoint Portal Server Architecture

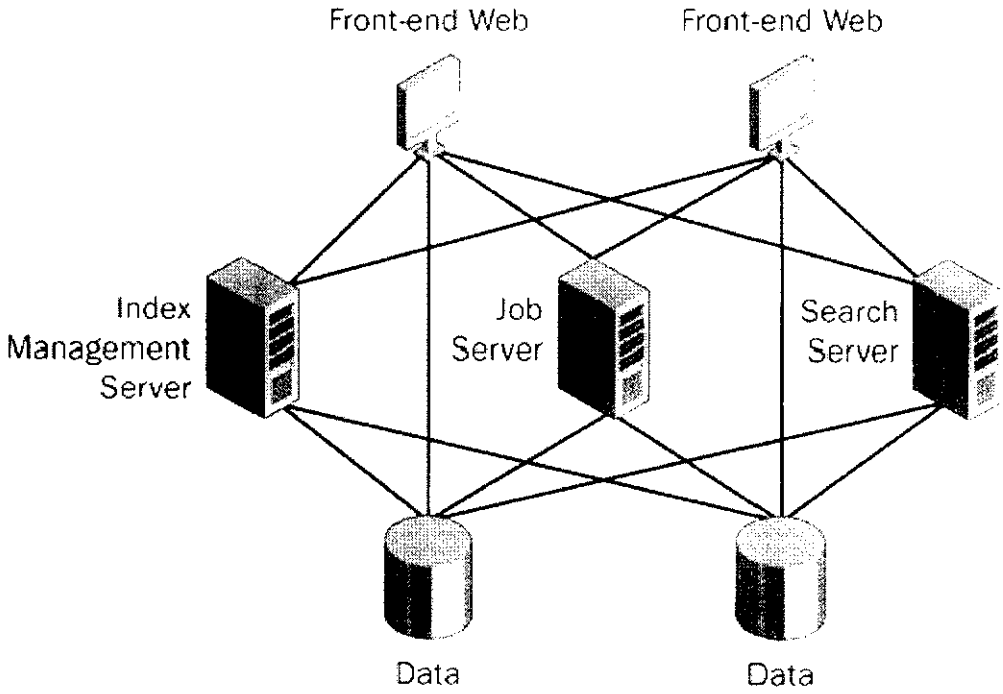

Fig.1.4.1 Sharepoint Portal Server Architecture

- **Front End Servers:**
  - Host sites and process search requests.
  - Sharepoint Portal Server includes the ability to separate front-end roles for Web servers and also includes separate roles for index management servers, search servers, and a job server.
  - One or more servers can fill each role, except for the job server, depending on the number of users that are supported.

- **Job Server:**
  - There is only one job server per portal site.
  - The purpose of a job server is to manage the additional services that are supported in Sharepoint Portal Server that are not available in Windows SharePoint Services.
  - Additional services are:
    - Hosting the Single Sign-On pages.
    - Importing user profiles.
    - Performing audience calculations.
    - Crawling and indexing portal site content.
    - Hosting the Alerts service.

- **Index Server**:
  - An index management server is dedicated to building and updating indexes, which includes crawling content and breaking down the crawled information through the indexing process.
  - Windows Sharepoint Services combines the services of hosting site pages, crawling content, and processing search queries from clients into one front-end Web server.

Portal

- Unified place that connects people to contextually relevant information, services and applications.
- Delivered by a framework, not a single product.
- Collection of services and applications.
  - Distinct needs
  - Shared services

Generic Architecture of Sharepoint Portal Server

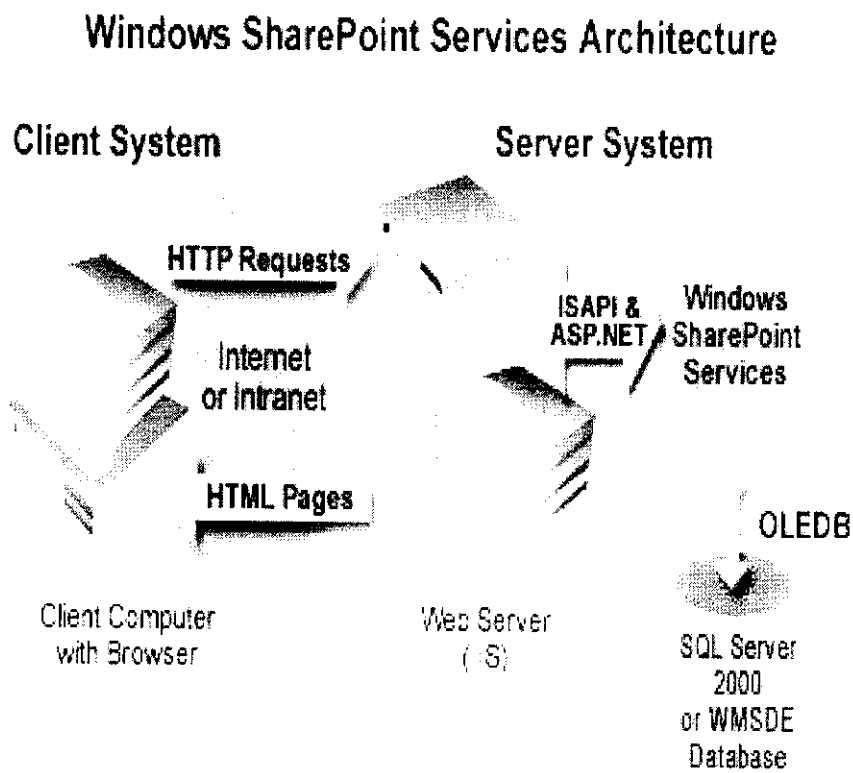

Fig.1.4.2 Architecture of Sharepoint Portal Server

CHAPTER 2

THEME OF PROJECT

2.1. SYSTEM ANALYSIS

2.1.1. LITERATURE REVIEW

I got an opportunity to do my academic project in Caritor. They allotted a project, which I thought as a challenging one and useful for my career. I was trained in .NET platform so it is easy for me to do this project when I am allotted in this project. This motivated me to do this project

2.1.2. METHODOLOGY

A process model for software engineering is chosen based on the nature of the project and application, the methods and tools to be used, and the controls and deliverables that are required.

The Linear Sequential (Waterfall Model) Model was used to develop the document management system which suggests a systematic, sequential approach to software development that begins at the system level and progresses through analysis, design, coding, testing and maintenance. This model encompasses the following activities:

System/information engineering and modeling:

System engineering and analysis encompasses requirements gathering at the system level with a small amount of top-level analysis and design. Information engineering encompasses requirements gathering at the strategic business level and at the business area level.

Design:

It is a multi-step process that focuses on four distinct attributes of a program: data structure software architecture, interface representations, and procedural (algorithmic) detail. This process translates requirements into a representation of the software that can be assessed for quality before code generation begins. The design is documented and becomes part of the software configuration.

Code generation:

The design must be translated into a machine-readable form. If this task is performed in a detailed manner, code generation can be accomplished mechanistically.

Testing:

Testing begins after code generation. It focuses on the logical internals of the software, assuring that all statements have been tested, and on the functional external-that is conducting tests to uncover errors and ensure that defined input will produce actual results that agree with required results.

Maintenance:

Software will undergo change after it is delivered to the customer. Change must occur because errors have been encountered, because the software must be adapted to accommodate changes in its external environment or because the customer requires functional or performance enhancements

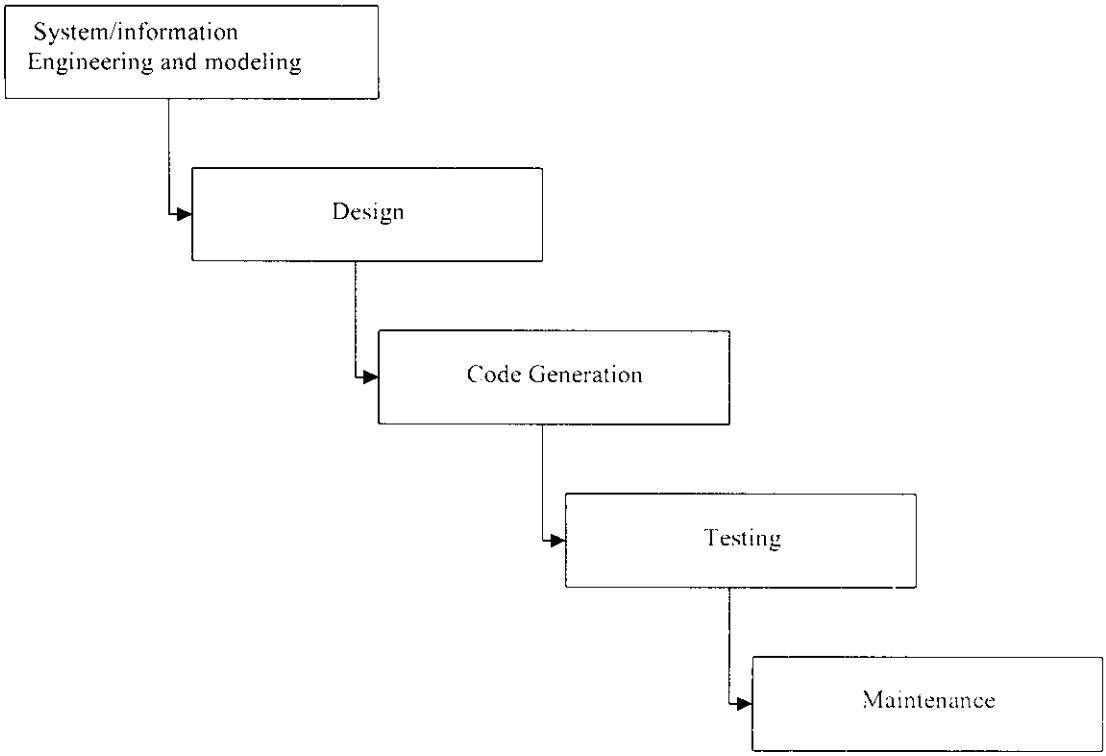

Fig.2.1.2.1

Advantages

- Testing is inherent to every phase of the waterfall model.
- It is an enforced disciplined approach.
- It is documentation driven, that is, documentation is produced at every stag.

2.1.3 SYSTEM REVIEW

2.1.3.1 Existing System

Creating documents are an integral part of any organizations day-to-day activities. Documents will be created whenever there is a need to share information between two people or departments or with people external to the organization. For example, when a design for a new application needs to be sent to the customer, it will be sent as a document. When a response to a proposal is sent to a prospective customer, it will be sent as a document. However, none of these documents will be created by a single person. Different people will work on different section of the document. In such cases, synchronizing the changes of all users is a never-ending exercise. The proposed Document Collation Tool will play a major role in solving this problem. The problems, which are perceived by the customers/users in the existing system, are as follows

- Less efficiency and accuracy due to lot of manual effort.
- Increased in effort due to synchronize issue.
- Constant upload is impossible.
- Many documents have to be maintained to record daily transaction process.

2.1.3.2 Proposed System

The proposed system would automate all of the manual processes described which would help reduce the overload incurred by the user and make the whole process simple and efficient. The proposed system will have computerized entry screens and process can be carried out based on the input of the user. The proposed system is designed to provide a solution for the draw backs of present system. It aims to

- Replace manual processing system with an automated one.
- Reduce synchronization issues.
- Update information system and provide easy access to documents.
- Total security is available.
- Easy maintenance of documents.

2.1.4 FEASIBILITY ANALYSIS

In the System analysis phase, the facts gathered during the system study were interpreted.

The following steps were followed in the analysis phase:

- Feasibility analysis was done to see if the system was feasible.
- The input to be given was analyzed.
- The output required by the system was analyzed.
- The processes which convert the input and available data to the necessary output were analyzed.

Thus the client's needs were analyzed and the system was evaluated for feasibility during system analysis. The different elements of the system were analyzed and the allocation of hardware, software and other elements were done. Before the system is developed, it is important to conduct a feasibility analysis and only if the results are favorable, the system is developed. Feasibility Study is the first stage of System Development Life Cycle of a project. The solutions are evaluated in the light of their economic, technical and operational implications in an attempt to establish whether or not it is worthwhile for the organization commit further resources to the project. All these studies have been taken into considerations for developing "DOCUMENT COLLATION TOOL".

Technical Feasibility

It is a study of function, performance, and constraints that may affect the ability to achieve an acceptable system. Types of hardware and software are assessed to determine whether they can support the tasks required.

- Basic configurations of "Document Collation Tool" use Microsoft products. The system is developed on these products are very common and very easy to use.
- "Document Collation Tool" is made capable of handling all the transactions using .NET components which are developed using C# language.

- As all the users of "Document Collation Tool" are new to the components, they will be provided with adequate training, which is taken care by the customer support of the organization.

Operational Feasibility

Organizational, political and human aspects are considered in order to ensure that the proposed system will be workable when implemented. Impact of "Document Collation Tool" workflow is assessed. Progressive developers can enrich this tool so that the resistance is minimized. It is ensured that the users have been involved in the development of this "Document Collation Tool". The naive users are highly supported with the help provided by this "Document Collation Tool" since it provides them with the necessary information in the form of informative and formatted screens. "Document Collation Tool" provides efficiency, accuracy and processing speed.

Economical Feasibility

It is commonly known as cost/benefit analysis. It is an evaluation of development cost weighed against the ultimate income or benefit derived from the system or product. Only if the benefits outweigh costs, then only the system will be developed. The "Document Collation Tool" satisfies this constraint successfully.

2.2 SYSTEM DESIGN

2.2.1 INPUT DESIGN

Input design is a part of the overall system design, which requires very careful attention. Inaccurate input data are the most common cause of errors in data processing. Collection of input data is the most expensive part of the system in terms of both the equipment and the number of people involved. So to make the study, the inputs given by the user are strictly validated before making a manipulation with it.

The input screens are:

- **Login Form:**

  The "Departmental Admin" makes entry into the system through the login form only. The login form consists of user name and password fields.

- **Lists:**

  A Web site stores and displays information that users can add to by using their browsers. You can add SharePoint lists to portal pages. You can create a custom list or select one of the predefined list types, such as a form library, contacts, tasks, discussion boards, surveys, and link lists. You can use the lists as defined or you can modify them. You can also delete lists that you don't use. After you create the list, you can display it on the page by adding its Web Part. List managers can approve or reject items that are submitted to the list and add comments. List managers can also apply permissions to a list, allowing only specific users to make changes to the list.

- **Document Libraries**

  A folder where a collection of files is stored and the files often use the same template. Each file in a library is associated with user-defined information that is displayed in the content listing for that library.

- **Discussion Boards**

A series of messages or comments in which replies to a message or comment are nested directly under it, instead of the messages or comments being arranged in chronological or alphabetical order.

- **Surveys**

A Web site enables users to respond to a set of questions specified by the creator of the survey. Results are tallied in a graphical summary. It requires a Web server that is running Windows SharePoint Services or SharePoint Portal Server.

- **Integration with Client programs**

It allows a multiple transaction interaction between client and server.

- **Team Website Customization**

The default, top-level site provided by a Web server or virtual server. To gain access to the top-level Web site, you supply the URL of the server without specifying a page name or sub site.

- **Security**

A proposed open standard was for establishing a secure communications channel to prevent the interception of critical information.

2.2.2 Output Design:

The output design is another very important phase. The output is mainly used to communicate with the user, processing the input data input data given by the user, processing the input data given by the user. Without the quality output, the entire system appear to be so unnecessary that users will use it, possible causing it to fail. The output screens are designed in a very simple and easy to understand format. The right output must be developed while ensuring that each output element is designed so that people will find the system easy to use effectively and improve the system's relationships with the user and help in decision making.

2.2.3 Database Design

A Database is a collection of information. It is a logical development process used by computer to access and manipulate data stored in various parts of computer systems. The overall objective in the development of database technology has been to trend data as an organizational resource. It is the backbone of any system. Hence success of any system depends on database design.

SYSTEM FLOW

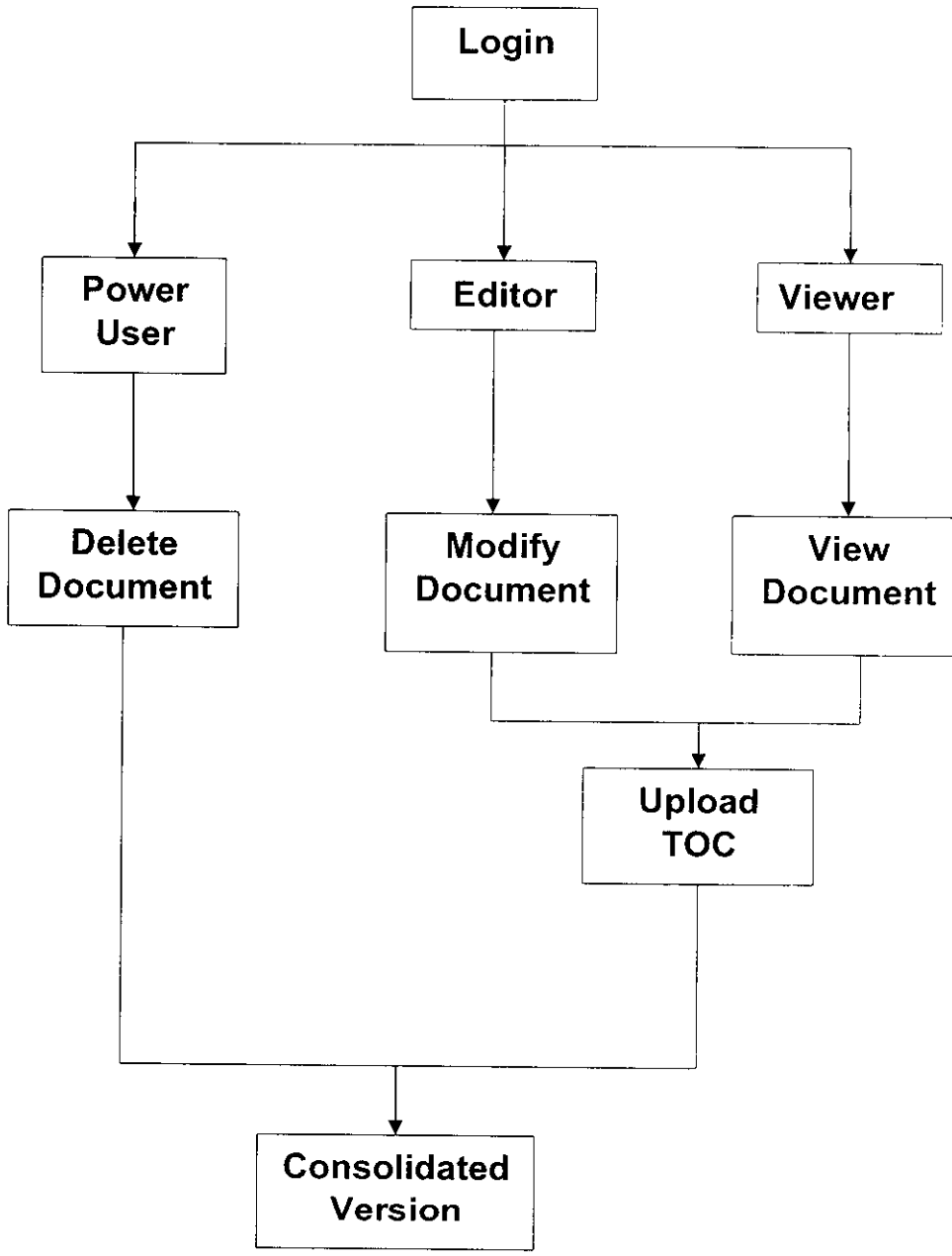

Fig.2.2.3.1 System Flow

UML

The Unified Modeling Language (UML) is a standard language for specifying, visualizing, constructing and documenting the artifacts of software system, as well as for business modeling. The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notation to express the design of software projects. Using the UML helps projects teams communicate, explore potential designs, and validate the architecture design of the software.

Goals of UML

The primary goals in the design of the UML were:

1. Provide users with a ready-to-use, expression visual modeling language so they can develop and exchange meaningful models.
2. Provide extensibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development processes.
4. Provide a formal basis for understanding the modeling language.
5. Integrate best practices.
6. Support higher-level development concept such as collaborations, framework patterns and components.

USECASE DIAGRAM

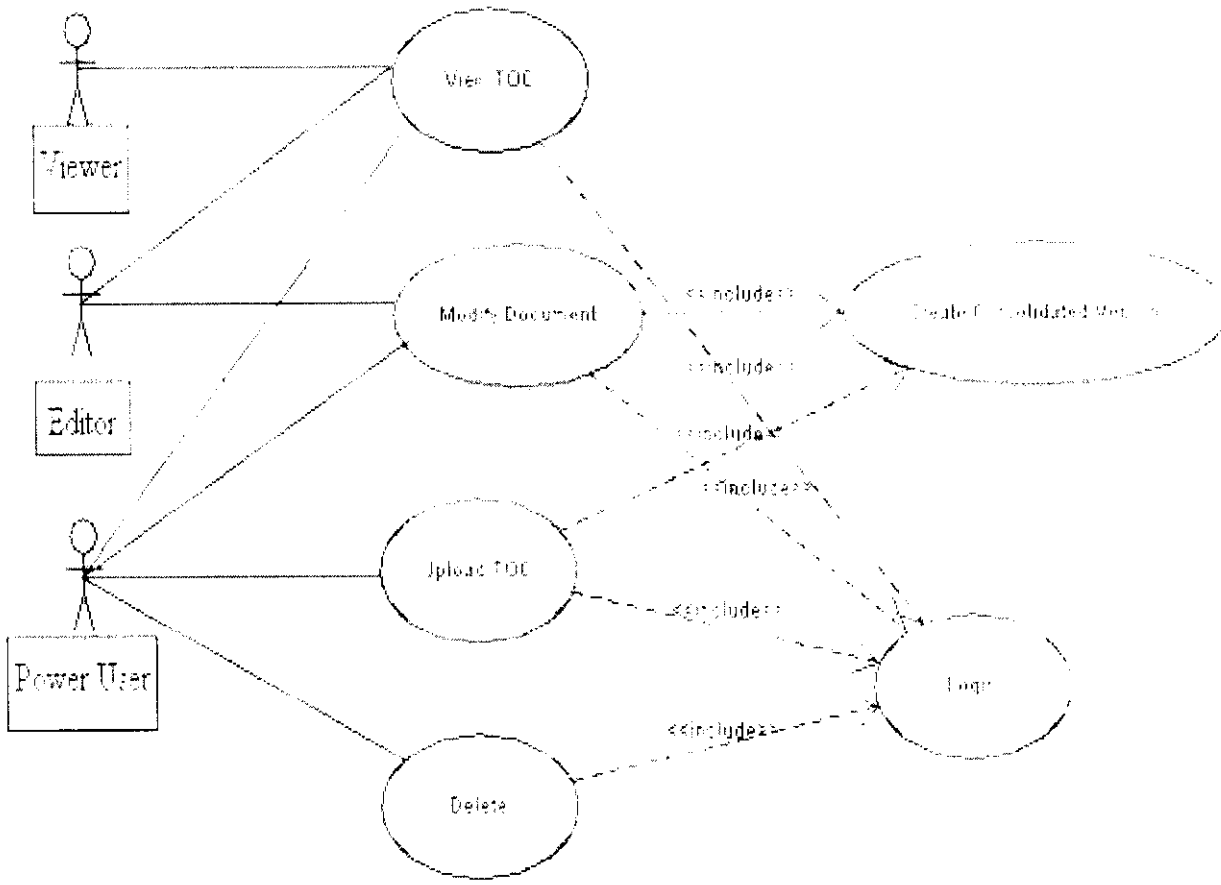

Fig.2.2.3.2 Use Case Diagram

STATE CHART DIAGRAM

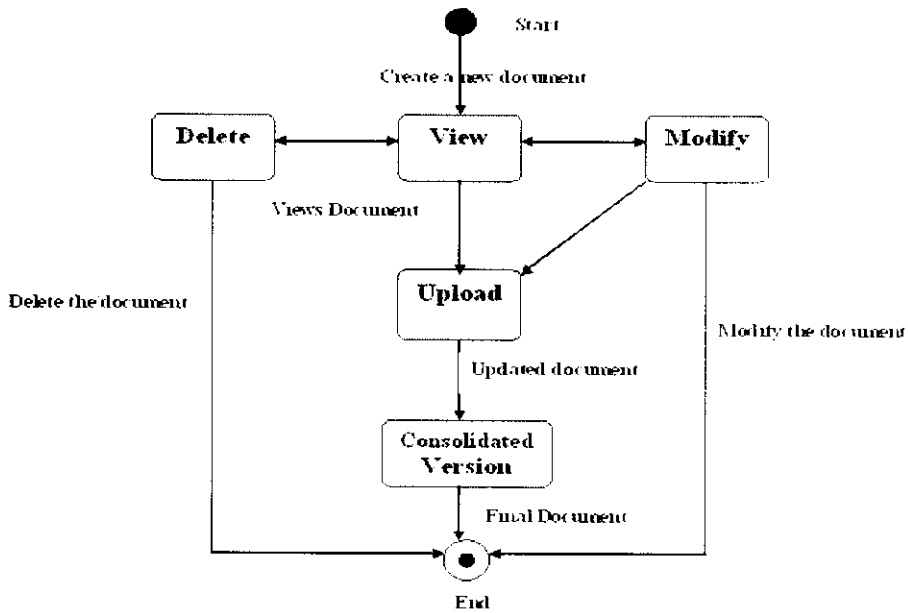

Fig.2.2.3.3 State Chart Diagram

2.3 MODULE DEVELOPMENT

Events

On Click:

The event frequently used in this system for achieving several actions. Generally, it is used to navigate through the links, to the next pages in the site.

Actions

- Add Listing
- Create Subarea
- Upload Document
- Change Settings
- Manage Users
- Manage Content
- Manage Portal Site
- Edit Page

Add Listing:

This page helps to add a listing to an area. Listings in this area may require approval before they are displayed

Create Subarea:

This page helps to create an area.

Upload Document:

This page helps to upload a document.

Change Settings:

This page helps to change the general settings associated with this area. Click the tabs to change other settings.

Manage Users:

This page allows adding new users, removing users from all site groups, or assigning users to site groups. The following users have been added to this Web site. To edit which site groups a user belongs to, click the user's name in the list.

Manage Content:

This page shows all the libraries, lists, discussion boards, and surveys in this Web site. Click the name of a library or list to view its contents. To create a new library or list, click Create.

Manage Portal Site:

This is a view of the structure of the portal. From this view, you can drag areas and listings to new locations. You can also limit the view to only the current area by clicking Filter on the area's menu. To go back to the whole list, click Reset.

Edit Page:

This page helps to edit a page.

2.4 TESTING & IMPLEMENTATION

Testing Guidelines

Testing indicates the utility of attempting to detect all errors. It is the process of executing test cases with the indent of exposing errors. Software testing is the critical element of the software quality assurance and represents the ultimate view of specialization, design and coding. The increasing availability of the software as a system element and the attendant costs associated with software failure are motivating forces for well planned through testing.

The various levels at which testing are conducted are,

- Unit Testing
- Integration Testing
- System Testing
- Validation Testing

Testing Objectives

The following are the objectives in testing,

- Testing is the process of executing the program with the intent of finding an error.
- A good testing is one that has a high probability of finding an as yet undiscovered error.
- A successful test is one that uncovers an as yet undiscovered error.

These objectives imply a dramatic change in viewpoint. They move counter to the commonly held view that a successful test is one in which no errors are found. Our objective is to design a test that systematically uncovers different classes of errors and to do so with a minimum amount of time and effort.

Unit Testing

In unit testing each program unit is tested individually. So any errors in a unit debugged. Sample data is given for unit testing. The unit test results are recorded for further reference. During unit testing, the functionality of the programs is tested. The validations for the input files are also tested.

Integration Testing

Integration testing is a systematic technique for constructing the program structure while at the same conducting test to uncover errors associated with interfacing. The objective is to take unit tested modules and build a program structure that has been dictated by design.

Validation Testing

Software validation is achieved through a series of black box test that demonstrated conformity with requirements. Both plan and procedure are designed to ensure that all functional requirements are achieved using validation testing. The forms need to be validated in which the concerned person should not leave any input area as blank and it cannot be allowed to enter improper data.

White Box Testing

Guarantees that all independent paths within a module have been exercised at least once exercise all logical conditions on their true and false. Exercise all loops at their boundaries and with in their operational bounds. Exercise internal data structures to ensure their validity functions, interfaces errors, errors in data structures or external data sources, performance errors, initialization and termination errors are traced through this testing and they are rectified.

IMPLEMENTATION

Implementation means the process of converting a new or a revised system design into an operational one. The "Document Collation Tool" consists of C# files, XML files, .frm files and route files of .NET. It is the most crucial stage in achieving a new successful system and in giving confidence on the new system for the users that it will work efficiently and effectively. So the system is tested successfully on various clients. In this phase, we can build the components either from scratch or by composition. Given the architecture document from the design phase and requirement document from the analysis phase, we can build exactly what has been requested.

CHAPTER 3

CONCLUSION

3.1 CONCLUSION

In designing the software "SHAREPOINT PORTAL" utmost care was taken to meet the user requirements as much as possible. The analysis and design phase was reviewed. Care was taken strictly to follow the Software Engineering concepts and principles so as to maintain quality in the developed system as per the user requirements.

This system was developed under modular approach since the implementation and maintenance is easy. Necessary software was provided so that the user would find it easy in maintaining the software in future. The test cases were generated on some sample data to test the working of the system.

3.2 FURTHER ENHANCEMENT

The project has covered almost all the requirements by the customers. Further requirements and improvements can be easily done. Since the tool is mainly user-friendly, the process of migration can be conducted by changing the existing modules or adding new modules. The migration can be carried out efficiently in future with reference to the older version through the tool which enables the process to be done.

APPENDIX

SCREENS

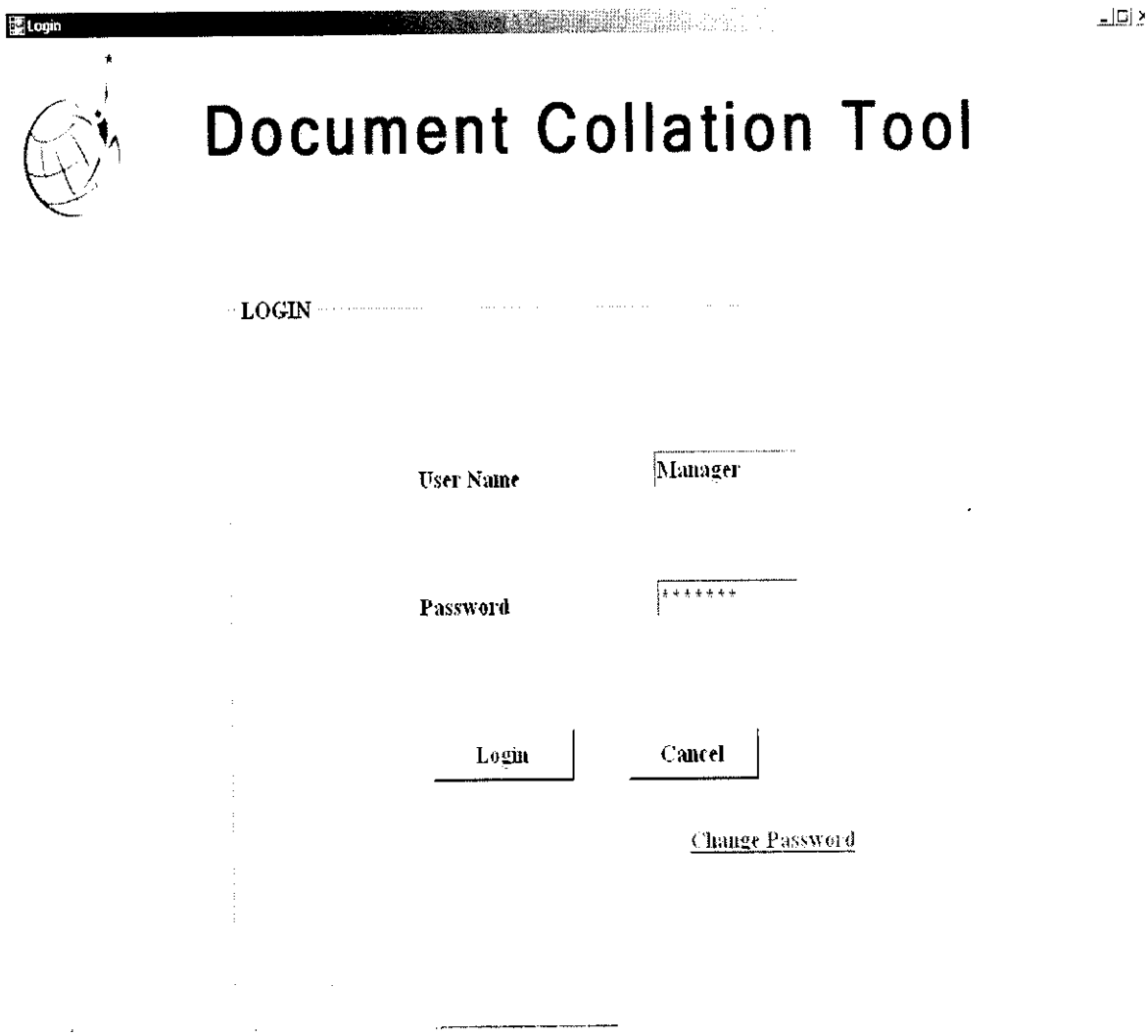

Fig.A.1 Login Screen

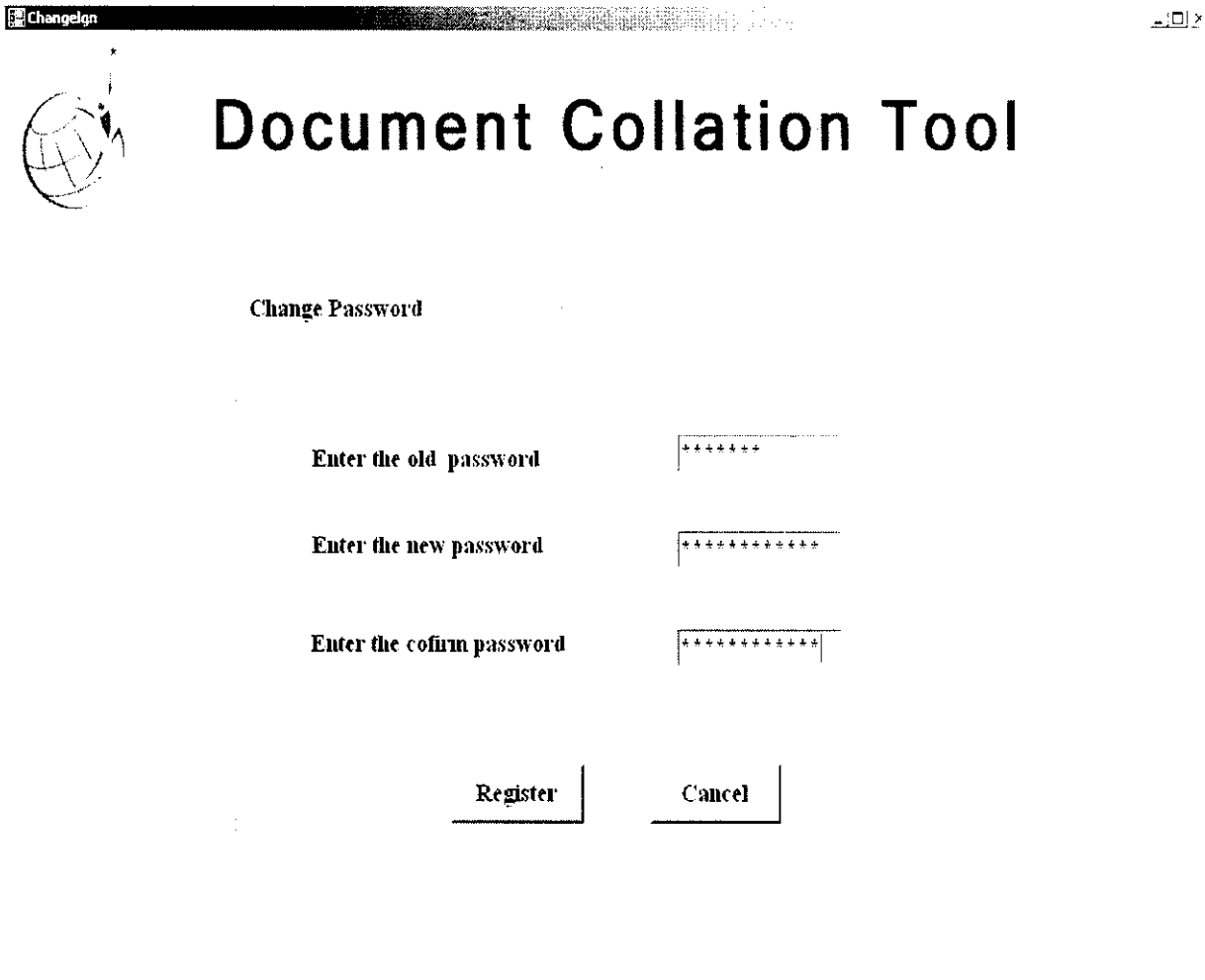

Fig.A.2 Change Password Screen

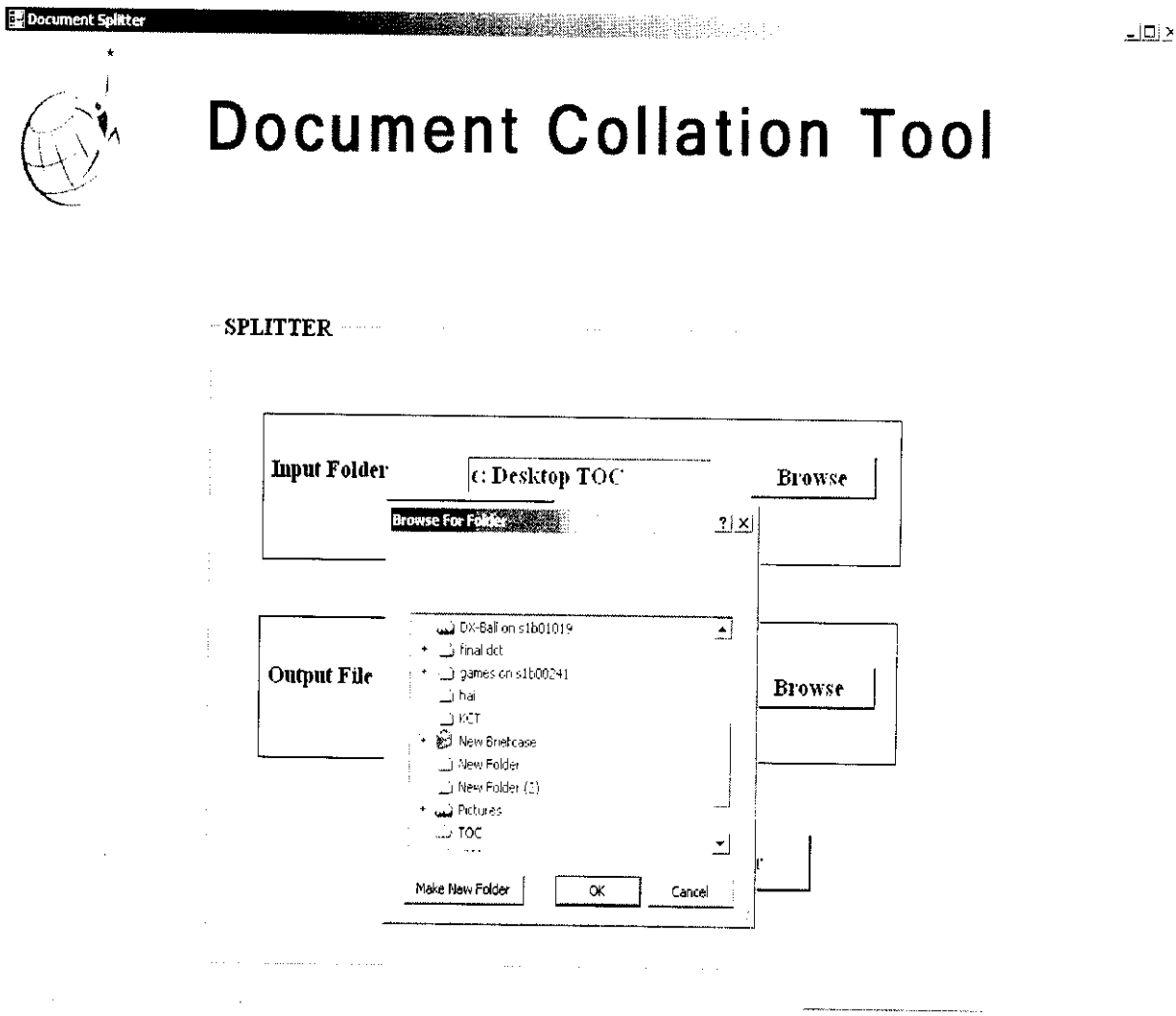

Fig.A.3 Splitter Screen

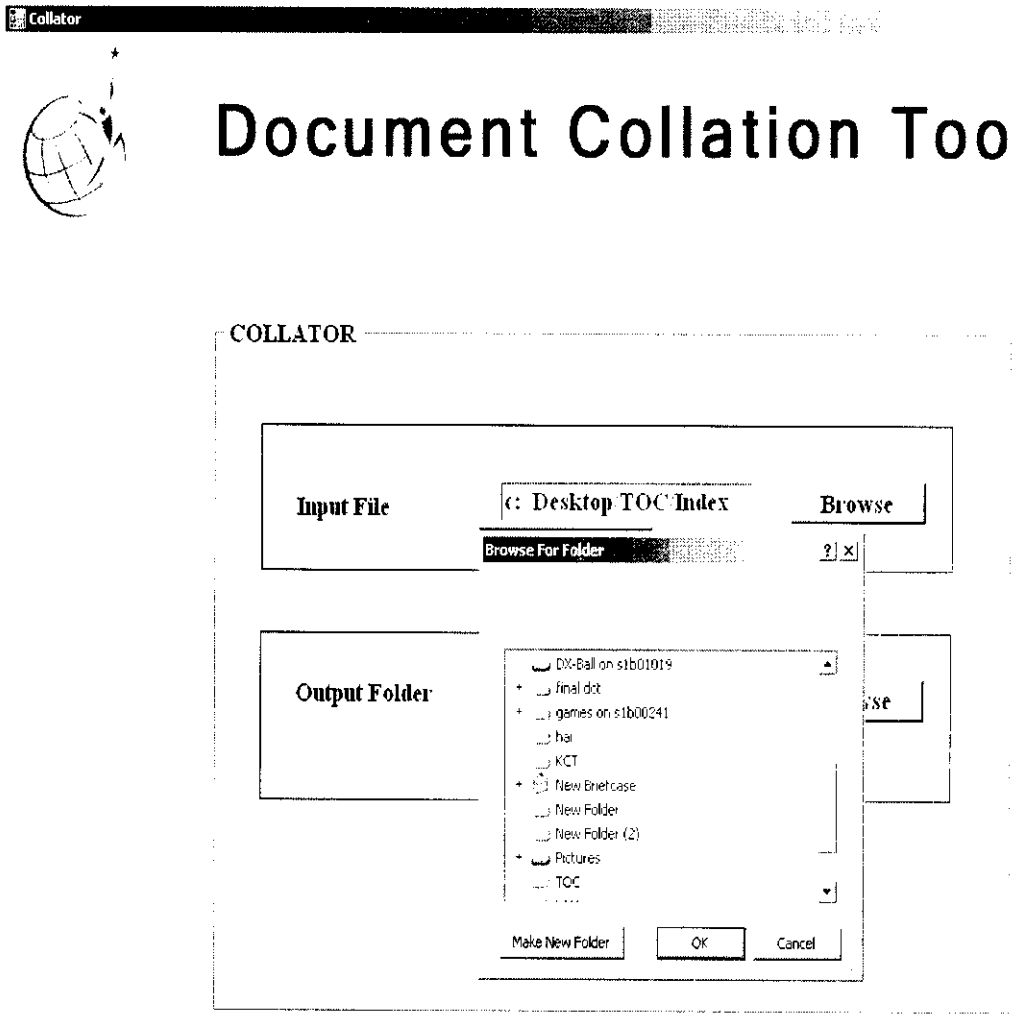

Fig.A.4 Collator Screen

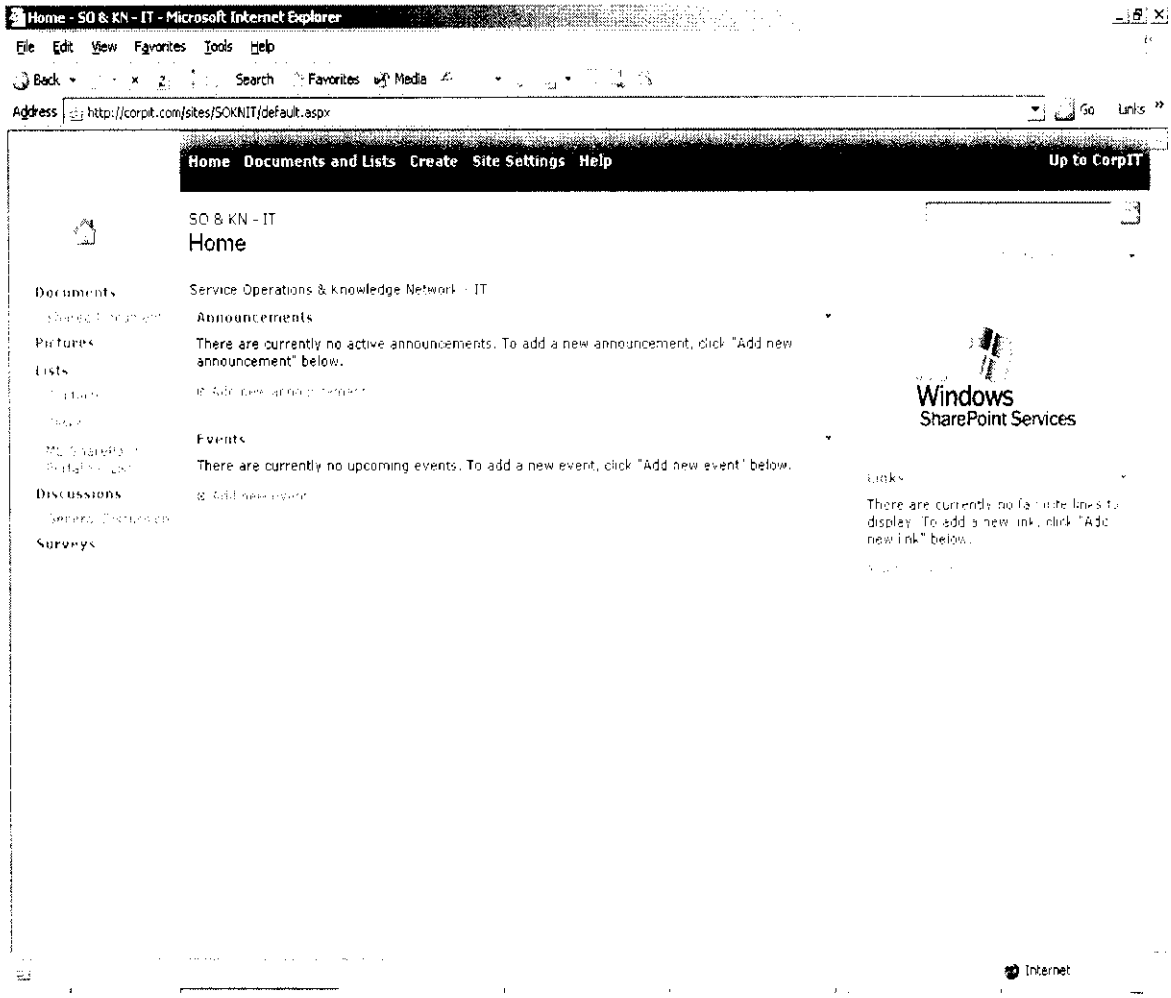

Fig.A.5 Home site

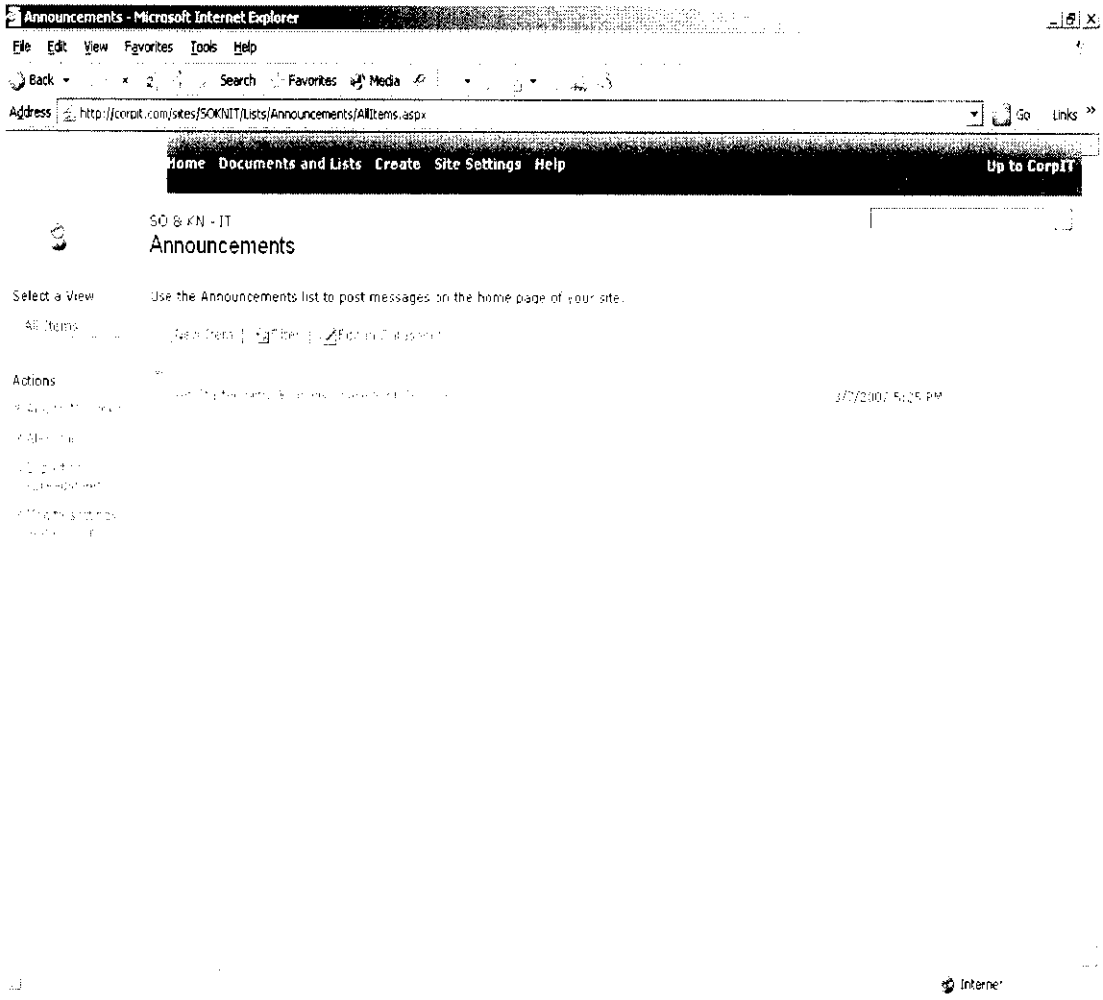

Fig.A.6 Announcements Site

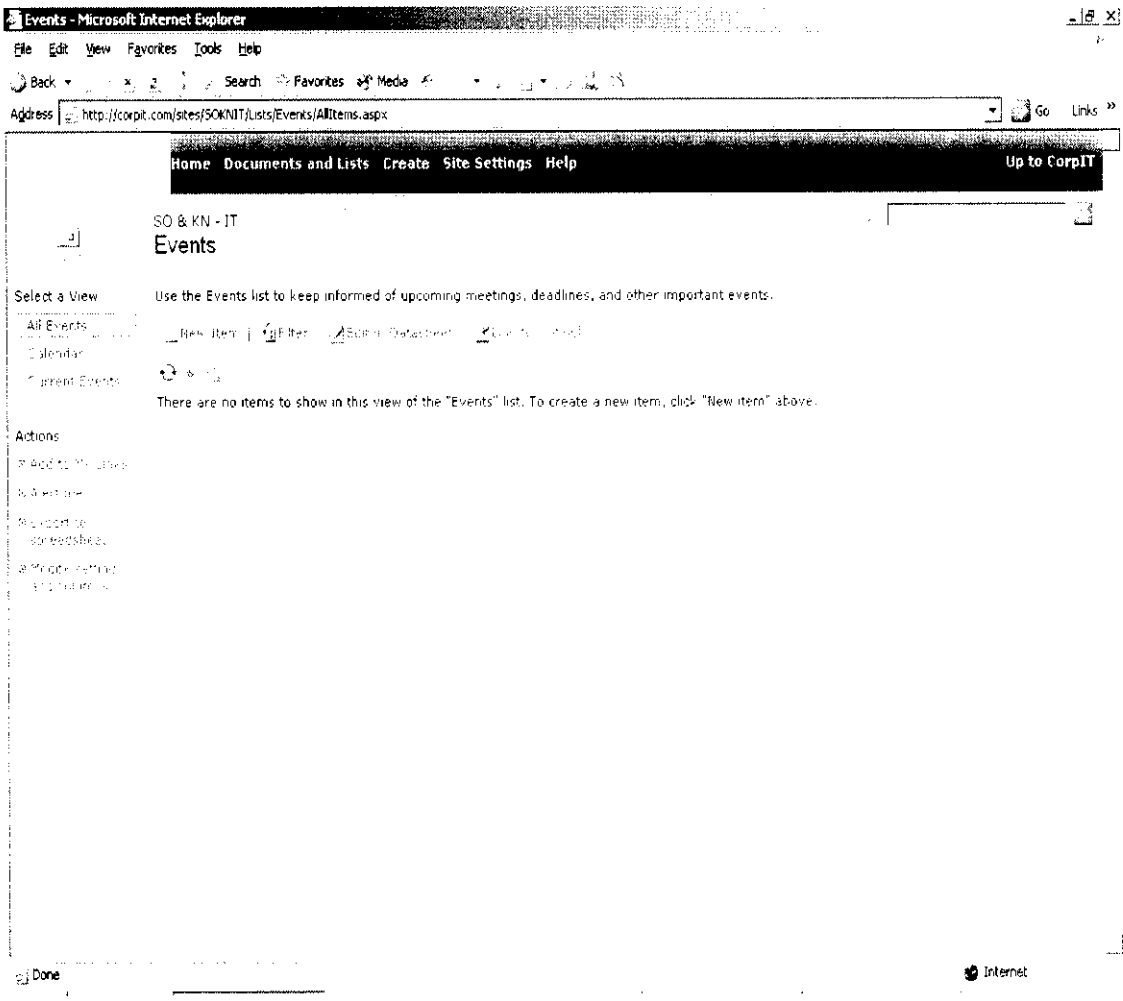

Fig.A.7 Events Site

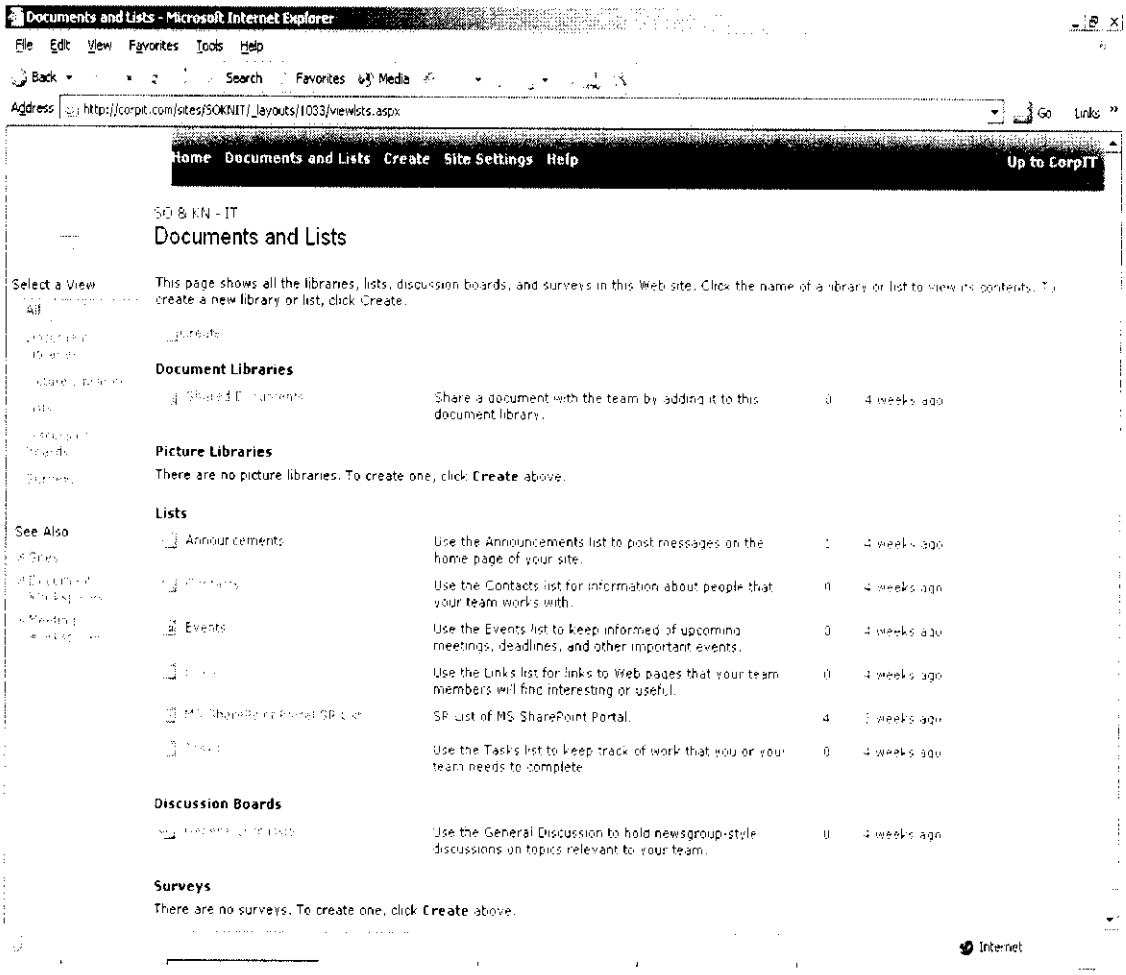

Fig.A.8 Documents & Lists Site

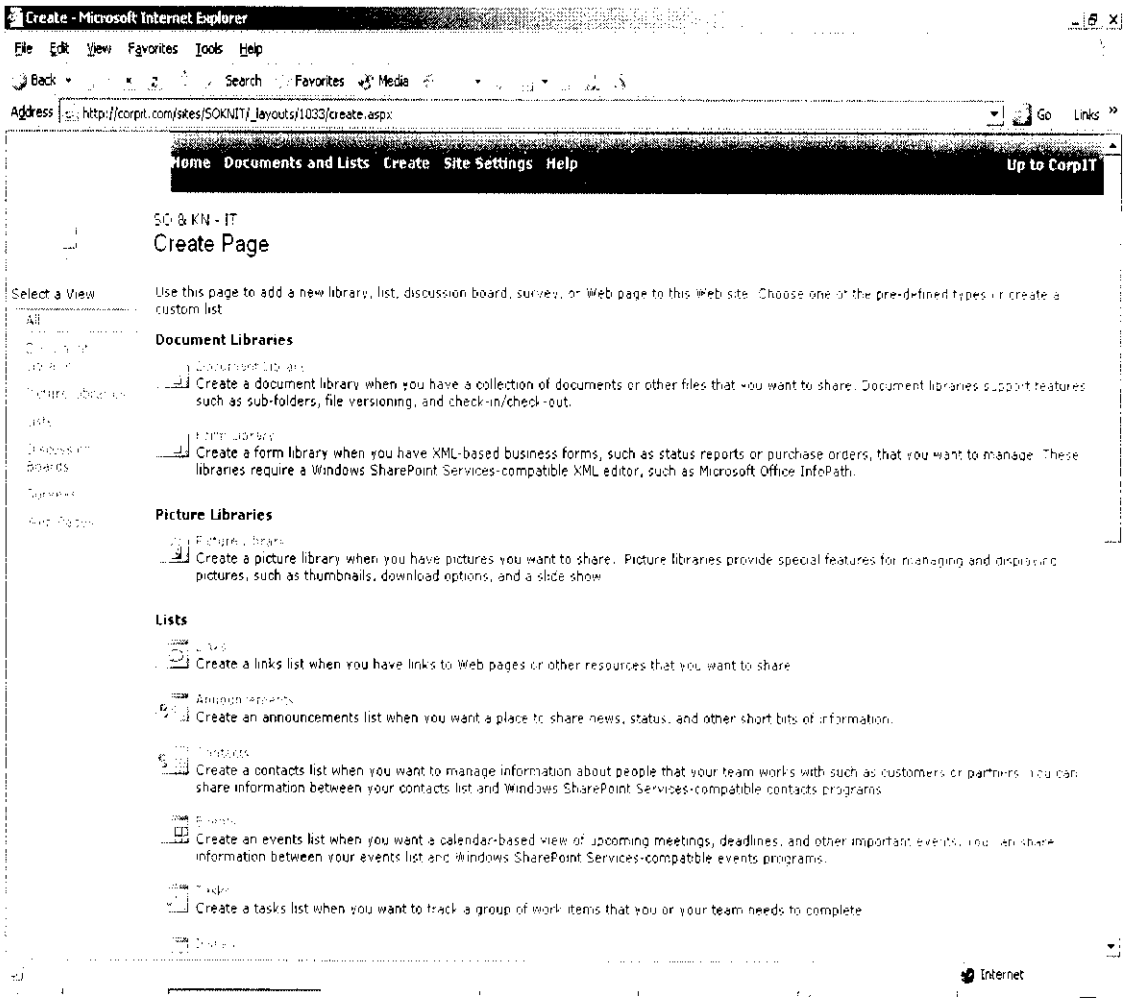

Fig.A.9 Page Creation Site

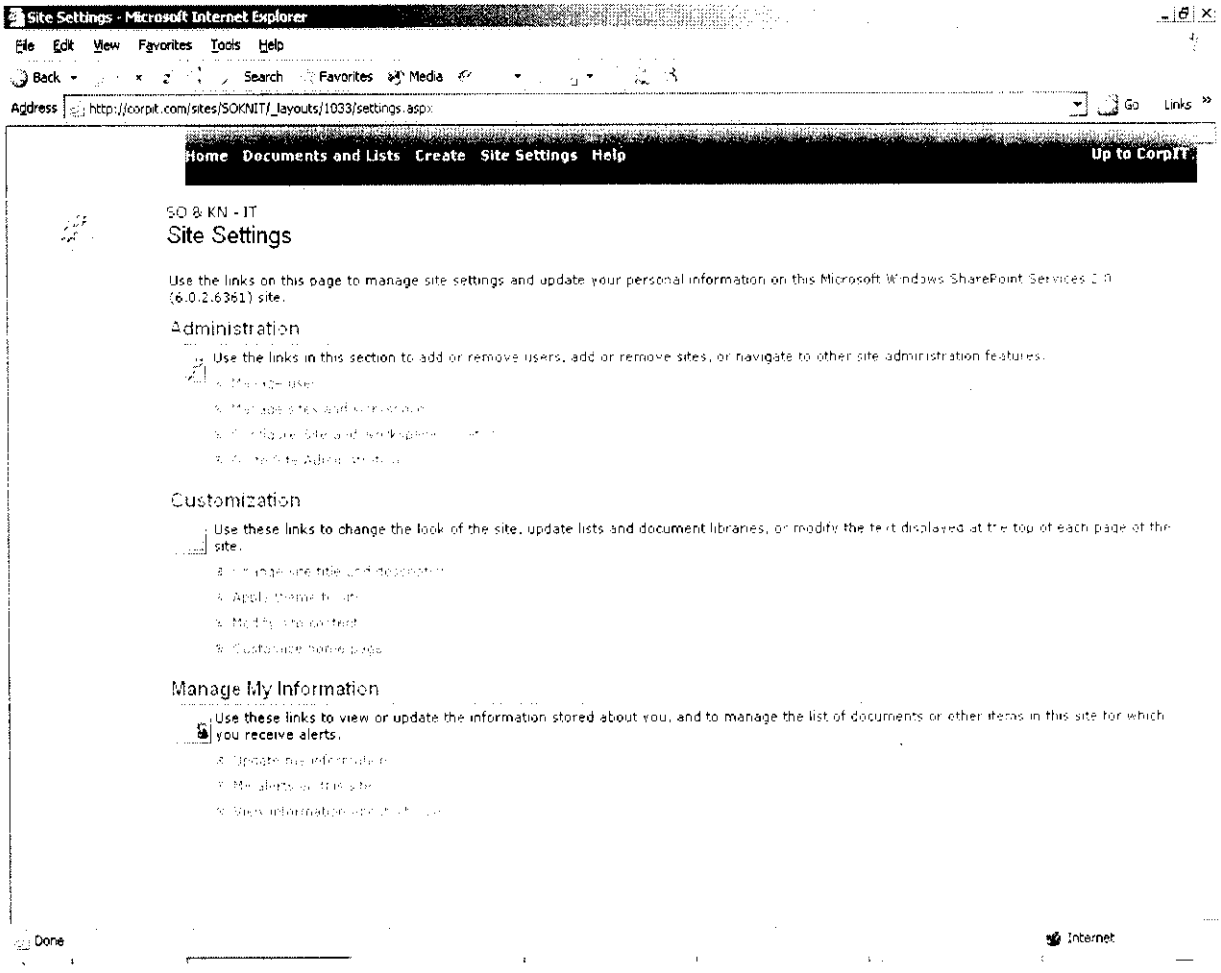

Fig.A.10 Site Settings

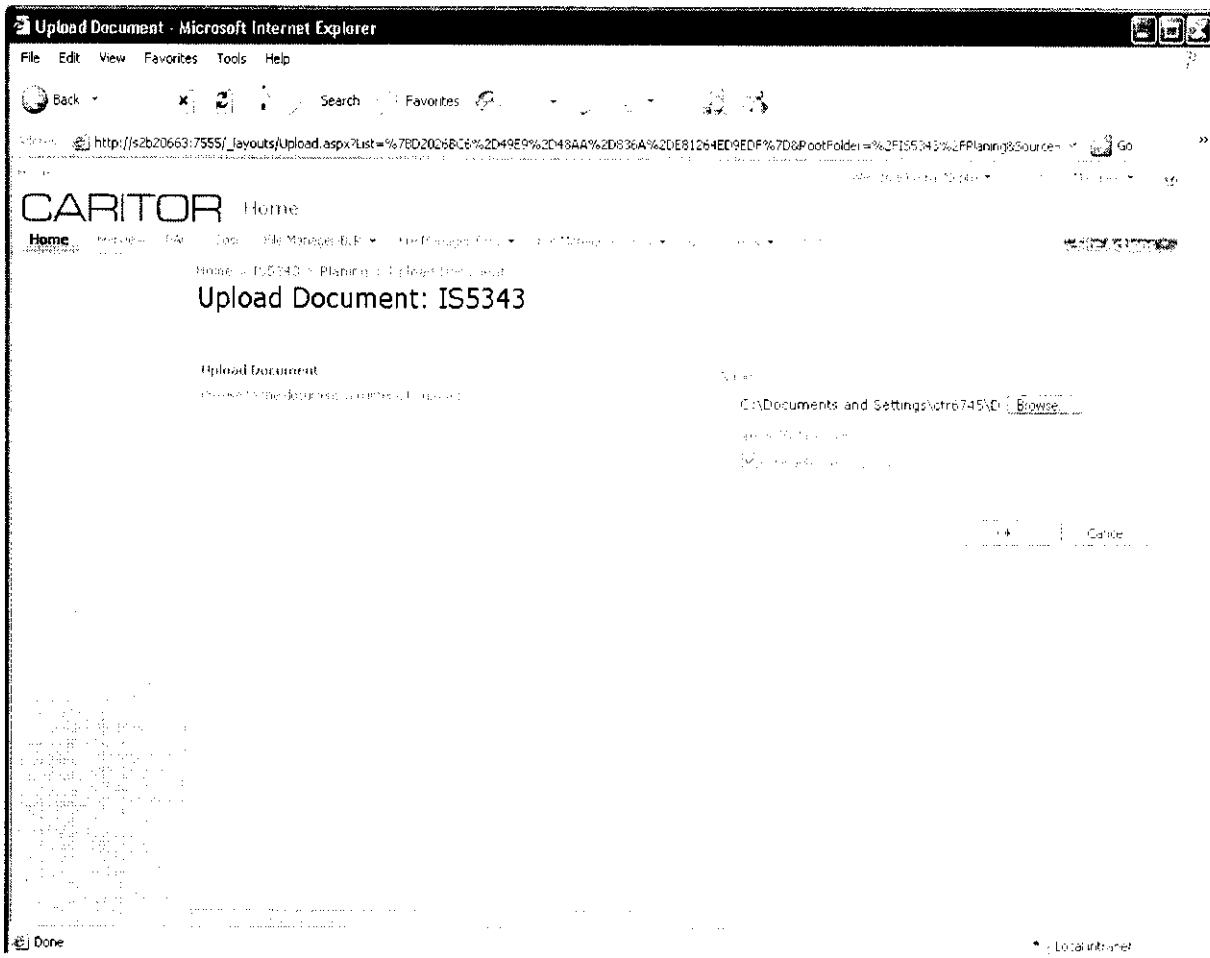

Fig.A.10 Uploading Site

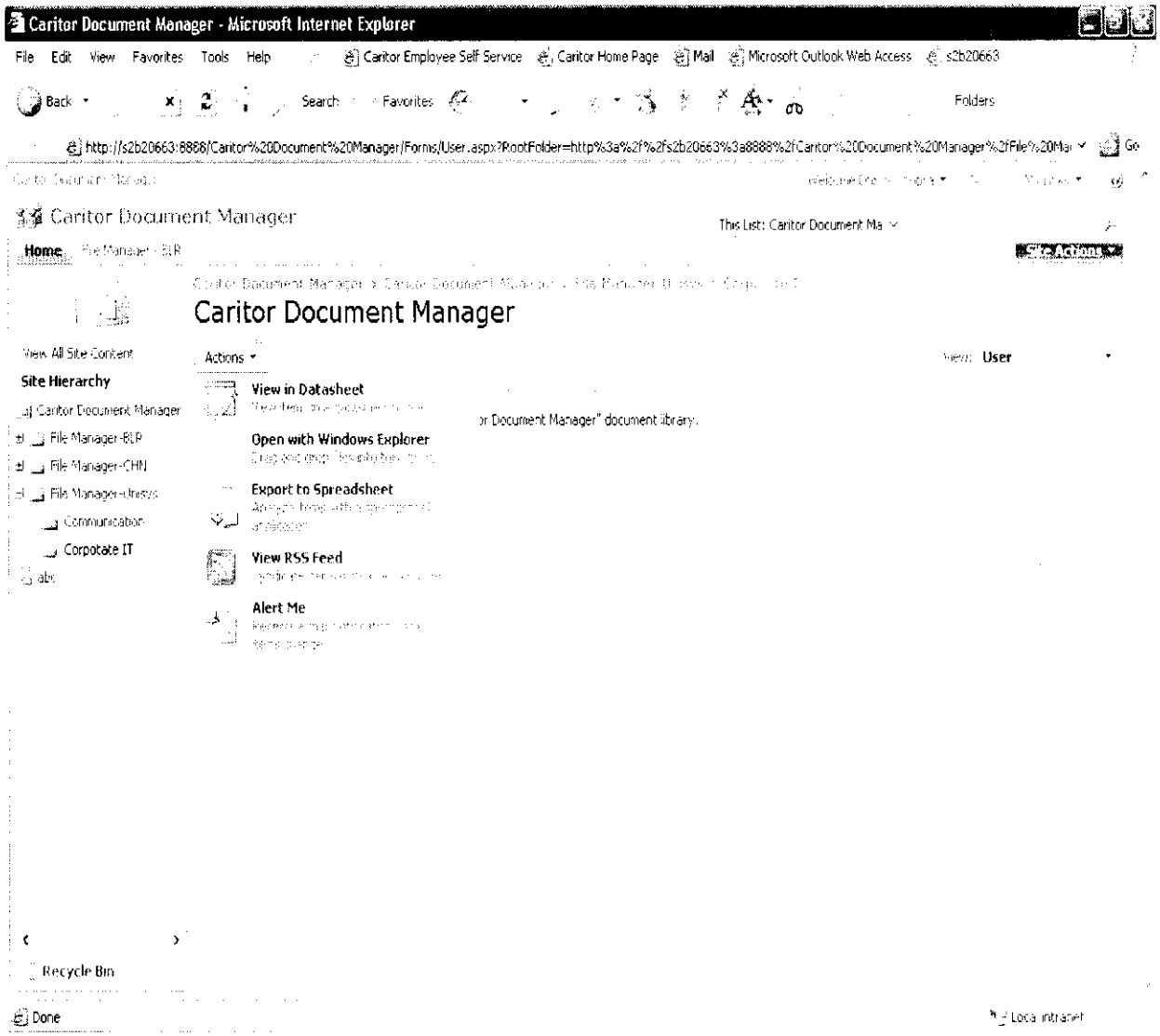

Fig.A.11 Viewing Documents

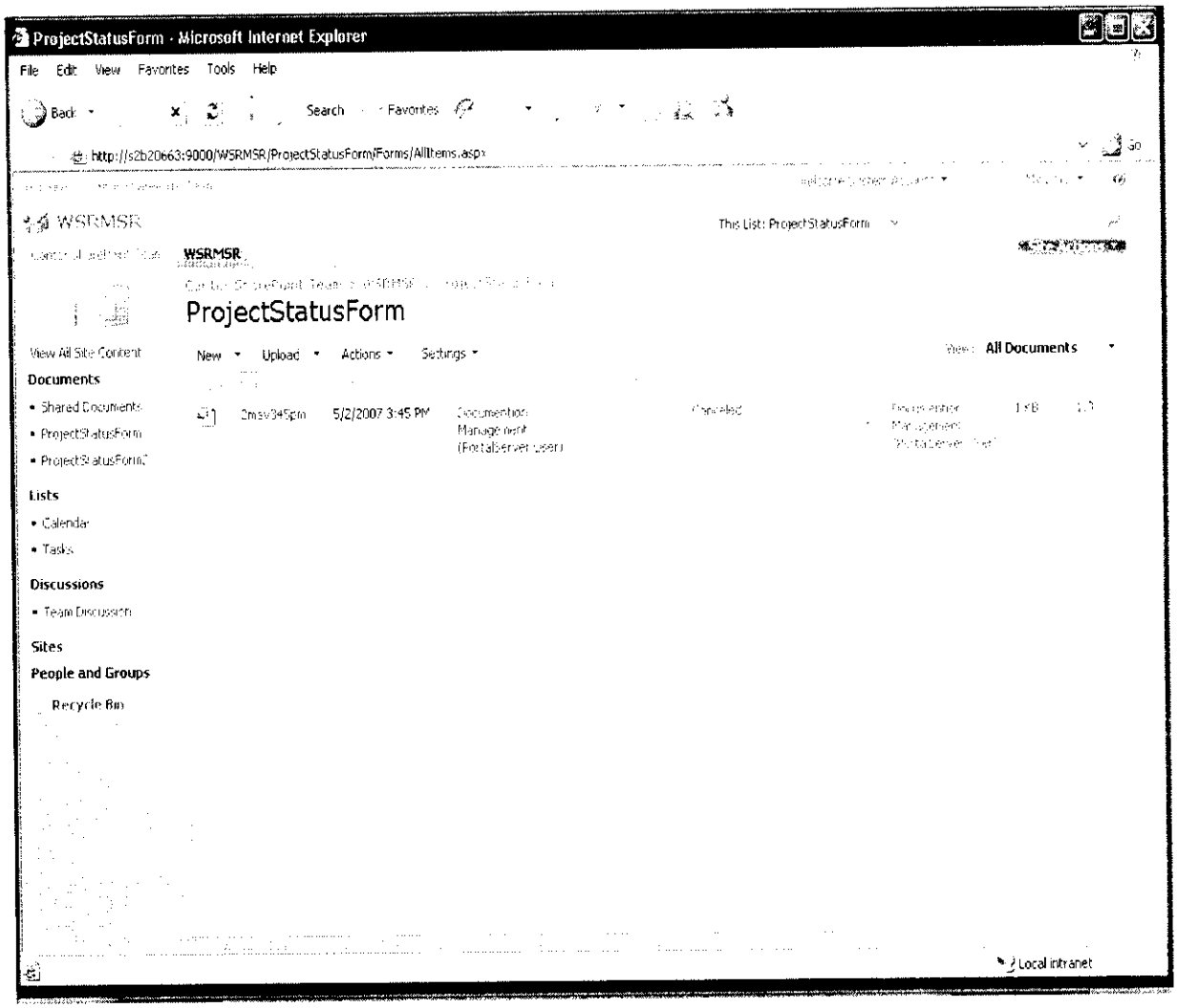

Fig.A.11 Report Generation Site

REFERENCES

Reference Books:

1. Developer's Guide to Windows SharePoint v3 platform – Todd C.Bleeker
2. Getting to know SharePoint 2007 - Ted Crace.
3. Mathew Macdonald, Mario Szpuszta 'ASP.NET 2.0 IN C#2005 APRESS-2005' .
4. Roger. S. Pressman (1991) 'Software Engineering - A Practitioners Approach' McGraw – Hill International Editions.
5. Simon Robinson, Christian Nagel, and Karli Watson "Professional C#", publications June 2004

Reference Sites:

1. http://en.wikipedia.org/wiki/.NET Framework
2. http://msdn.csharpfriends.com
3. http://msdn2.microsoft.com
4. http://www.microsoft.com/office/preview/servers/sharepointserver
5. http://www.skelta.com/products/sps/sharepoint.aspx