**ABSTRACT**

**Digital data in the form of text documents is rapidly growing. Analysing such data manually is a tedious task. Data mining techniques have been around to analyse such data and bring about interesting patterns. Many existing methods are based on term-based approaches that can’t deal with synonymy and polysemy. Moreover they lack the ability in using and updating the discovered patterns. Zhong et al. proposed an effective pattern discovery technique. It discovers patterns and then computes specificities of patterns for evaluating term weights as per their distribution in the discovered patterns. It also takes care of updating patterns that exhibit ambiguity which is a feature known as pattern evolution. In this paper we implemented that technique and also built a prototype application to test the efficiency of the technique. The empirical results revealed that the solution is very useful in text mining domain.**

**Text mining, also referred to as text data mining, roughly equivalent to text analytics, refers to the process of deriving high-quality information from text. High-quality information is typically derived through the devising of patterns and trends through means such as statistical pattern learning. Many data mining techniques have been proposed for mining useful patterns in text documents. However, how to effectively use and update discovered patterns is still an open research issue, especially in the domain of text mining. Since most existing text mining methods adopted term-based approaches, they all suffer from the problems of polysemy and synonymy.**

**Over the years, people have often held the hypothesis that pattern (or phrase)-based approaches should perform better than the term-based ones, but many experiments do not support this hypothesis. This paper presents an innovative and effective pattern discovery technique which includes the processes of pattern deploying and pattern evolving, to improve the effectiveness of using and updating discovered patterns for finding relevant and interesting information. Substantial experiments on RCV1 data collection and TREC topics demonstrate that the proposed solution achieves encouraging performance.**

**INDEX**

**CHAPTER 1**

**INTRODUCTION**

DUE to the rapid growth of digital data made available in recent years, knowledge discovery and data mining have attracted a great deal of attention with an imminent need for turning such data into useful information and knowledge. Many applications, such as market analysis and business management, can benefit by the use of the information and knowledge extracted from a large amount of data. Knowledge discovery can be viewed as the process of nontrivial extraction of information from large databases, information that is implicitly presented in the data, previously unknown and potentially useful for users. Data mining is therefore an essential step in the process of knowledge discovery in databases.

In the past decade, a significant number of data mining techniques have been presented in order to perform different knowledge tasks. These techniques include association Rule mining, frequent item set mining, sequential pattern mining, maximum pattern mining, and closed pattern mining. Most of them are proposed for the purpose of developing efficient mining algorithms to find particular patterns within a reasonable and acceptable time frame. With a large number of patterns generated by using data mining approaches, how to effectively use and update these patterns is still an open research issue. In this paper, we focus on the development of a knowledge discovery model to effectively use and update the discovered patterns and apply it to the field of text mining.

Text mining is the discovery of interesting knowledge in text Documents. It is a challenging issue to find accurate knowledge (or features) in text documents to help users to find what they want.

**CHAPTER 2**

**SOFTWARE REQUIREMENT SPECIFICATION**

**2.1 INTRODUCTION**

The introduction provides an overview of the entire SRS with purpose, scope, definitions, and references. The SRS aims to provide guidelines for how the project will be deemed successful and how these concepts and ideas will be created effectively. It provides a detailed overview of our software product, its parameters and goals.

**2.1.1 Purpose**

This document is the Software Requirement Specification for Effective pattern discovery for text mining.

This SRS describes the functions and performance requirements of this application. We provide an effective pattern discovery technique, which first calculates discovered specificities of patterns and then evaluates term weights according to the distribution of terms in the discovered patterns rather than the distribution in documents for solving the misinterpretation problem. The proposed approach can improve the accuracy of evaluating term weights because discovered patterns are more specific than whole documents.

**2.1.2 Document Convention**

The following document conventions have been used to ensure the easy of readability

* + - * Font: Times New Roman
      * Main Headings- Font size 16, Bold
      * Sub-Headings- Font size 14, Bold
      * Sub-sub-headings- Font size 12, Initial Capital letters

**2.1.2.1 Definitions**

* User- The person who operate the software product.
* GUI- Graphical User Interface
* Open Source Software- Software for which the code is freely available for use and research.
* API- Application Programming Interface
* JDK- Java Development Kit
* XML: Extensible Markup Language

**2.1.3 Scope**

This document plays a vital role in the software development life cycle (SDLC) and it describes the complete requirement of the system. It is meant for use by the developers and will be the basic during testing phase. Any changes made to the requirements in the future will have to go through formal change approval process.

**2.1.4 References**

[1] [Beginning ASP.NET 4: in C# and VB](http://www.amazon.com/gp/product/0470502215?ie=UTF8&tag=aspnettelligent-20&linkCode=as2&camp=1789&creative=9325&creativeASIN=0470502215)by Imar Spaanjaars.

[2] [Programming ASP.NET 3.5](http://www.amazon.com/gp/product/0596529562?ie=UTF8&tag=aspnettelligent-20&linkCode=as2&camp=1789&creative=9325&creativeASIN=0596529562)by Jesse Liberty, Dan Maharry, Dan Hurwitz*.*

[3] C. Wang, N. Cao, J. Li, K. Ren, and W. Lou, “Secure ranked keyword search over encrypted cloud data,” in Proc. of ICDCS’10,2010.

[4] Software Engineering: A Practitioner’s Approach, Roger S Prressman, 6th edition

[5] Grady Booch, James Rumbaugh, Ivar Jacobson: The Unified Modeling Language User Guide

[6] S. Kamara and K. Lauter, “Cryptographic cloud storage,” in RLCPS, January 2010, LNCS. Springer, Heidelberg.

[7] <http://www.asp.net.com>

[8] <http://www.dotnetspider.com/>

[9] <http://www.dotnetspark.com>

**2.2 OVERALL DESCRIPTION**

The description of the product includes a discussion about its features, types of users to whom the application may be useful, the environment in which the application can be operated etc.

**2.2.1 Product Perspective**

Since most existing text mining methods adopted term-based approaches, they all suffer from the problems of polysemy and synonymy. Over the years, people have often held the hypothesis that pattern (or phrase)-based approaches should perform better than the term-based ones, but many experiments do not support this hypothesis. This product presents an innovative and effective pattern discovery technique which includes the processes of pattern deploying and pattern evolving, to improve the effectiveness of using and updating discovered patterns for finding relevant and interesting information.

**2.2.2 Product Features**

• The website application features a powerful SQL server which enables its users for efficient searching of patterns over a wide range of files which are uploaded into the database.  
• Every search query is stored by the admin into the database for any further verification.  
•A variety of efficient algorithms are used for searching and text retrieval such as the D-pattern mining algorithm and IPEvolving algorithm which use the Apriori property for reducing the searching space.

• The result is based on precision and recall to evaluating the document. First the document loaded in the database and each document are split into paragraph to remove the unwanted word and noise in the document.

• The output obtained after searching for the pattern given by user can be downloaded onto the client-side system for any further use.

• It also provides the option of adding files to database by admin end which makes it easier for time-to-time updating by the admin.

**2.2.3 User Classes and Characteristics**

* Open Source Community

The open source community is expected to be the main user class of this application. Nowadays, with boom of internet where everything is available online, the open source community needs to be very sure of what data is made available online and also the information retrieval patterns are a crucial part of all search engines.

* Business Organisations

Companies have large databases of information and thus searching specific information from that and increasing accuracy of information retrieval is very important. Therefore our implementation would surely benefit business organisations to manage their huge repository of data.

* General Users

The general uses have to deal with a lot unnecessary data which is put up on the web. In such a situation, an effective text discovery scheme gets very helpful in order to extract the data which is needed.

**2.2.4 Operating Environment**

**Client Side Requirement**

OS : Windows 95/98/2000/XP/8/8.1

Software Packages : .NET FrameWork 3.5

**Server Side Requirements**

OS : Windows 95/98/2000/XP/8/8.1

Software Packages : SQL Server

**2.2.5 Design and Implementation Constraints**

In the time which was available to us, we have tried to implement all possible security features in the product. As an added security, the whole database can be encrypted including User ID fields.

**2.3 SYSTEM FEATURES**

This section of the SRS describes all the functionalities that the product provides with a detailed instructions guide to the users of how to use those features.

**2.3.1 Logging**

There will three forms of logging provided by the system:

* New User Login- First logging will be a new user to register, from which user will provide certain details about himself to facilitate efficient configuration of the system.
* Existing User Login-Second form of logging will be for existing user to log on to the system.
* Admin Login-Third form of logging will be for admin who can view all the users and their activities. He can even upload new files in both whole and split format.

**2.3.2 Searching**

A user can search files by entering different patterns of text. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

**2.3.3 Downloading**

Once the required files are searched it can be downloaded into the system and viewed. System takes over and the downloading operation is performed independent of user intervention.

**2.4 OTHER NON-FUNCTIONAL REQUIREMENTS**

Non-Functional Requirements are often called qualities of a system. Other terms for non-functional Requirements are constraints, quality attributes, quality goals, quality of service requirements and non-behavioural requirements. Qualities that are non-functional requirements can be divided into two main categories:

* Execution qualities, such as security and usability, which are observable at run time.
* Evaluation qualities, such as testability, extensibility and scalability, which are embodied in the static structure of the software system.

**2.4.1 Performance Requirements**

The system performs searching and retrieval of files at a relatively high speed. The download is also performed automatically on clicking the download button and system takes over. The performance depends on the input design and output design of the product.

**Input Design**- The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. Input Design considered the following things-

* + What data should be given as input?
  + How the data should be arranged or coded?
  + The dialog to guide the operating personnel in providing input.
  + Methods for preparing input validations and steps to follow when error occur.

**Output Design-**A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system’s relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2. Select methods for presenting information.

3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

* Convey information about past activities, current status or projections of the future.
* Signal important events, opportunities, problems, or warnings.
* Trigger an action.
* Confirm an action.

**2.4.2 Security Requirements**

The file content is stored in an encrypted format in the database therefore anybody who tries accessing the database cannot view the file content.

**2.4.3 Safety Requirements-** The security measures focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy.

**2.4.4 Software Quality Attributes**

The system will provide the user with easy to use and understand GUI interface. User can easily interact with the system with menus and text

**CHAPTER 3**

**ANALYSIS**

**3.1 EXISTING SYSTEM**

Since most existing text mining methods adopted term-based approaches, they all suffer from the problems of polysemy and synonymy. Over the years, people have often held the hypothesis that pattern (or phrase)-based approaches should perform better than the term-based ones, but many experiments do not support this hypothesis.

**3.2 DRAWBACKS OF THE EXISTING SYSTEM**

* Phrases have inferior statistical properties to terms.
* They have low frequency of occurrence.
* There are large numbers of redundant and noisy phrases among them.

**3.3 PROPOSED SYSTEM**

* Here we propose an effective pattern discovery technique.
* It evaluates specificities of patterns and then evaluates term weights according to the distribution of terms in the discovered patterns.
* It solves misinterpretation problem.
* It considers the influence of patterns from the negative training examples to find ambiguous (noisy) patterns and tries to reduce their influence for the low-frequency problem.
* The proposed approach can improve the accuracy of evaluating term weights because discovered patterns are more specific than whole documents.
* In General there are two phases-training and testing.
* In training phase the d-patterns in positive documents (Dþ) based on a min sup are found, and evaluates term supports by deploying d patterns to terms.
* In Testing Phase to revise term supports using noise negative documents in D based on an experimental coefficient.
* The incoming documents then can be sorted based on these weights.



Figure 3.1

**3.4 MERITS OF THE PROPOSED SYSTEM**

* The proposed approach is used to improve the accuracy of evaluating term weights because the discovered patterns are more specific than whole documents.
* To avoid the issues of phrase-based approach we use the pattern-based approach.
* Pattern mining techniques can be used to find various text patterns.

**3.5 FEASIBILITY STUDY**

Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation-

* Technical feasibility
* Economic feasibility
* Social feasibility

**3.5.1 Technical Feasibility**

The technical issue usually raised during the feasibility stage of the investigation includes the following:

* Does the necessary technology exist to do what is suggested?
* Is the underlying technology used secure and up to date with the current trend?
* Do the proposed equipment’s have the technical capacity to hold the data required to use the new system?
* Will the proposed system provide adequate responses to inquiries?
* Are there technical guarantees of accuracy, reliability, ease of access and data security?

**3.5.2 Economic Feasibility**

A system can be developed technically and that if installed must be a good investment for the organization. In the economic feasibility, the development cost in creating the system is evaluated against the ultimate benefits must equal or exceed the costs. This application has no maintenance costs and its development cost is very less compared to the service which it provides to the users.

**3.5.3 Social Feasibility**

The aspect of this 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.

**3.6 SCOPE OF THE PROJECT**

Data mining techniques have been around for long time. The techniques used to discover knowledge include sequential pattern mining, frequent item set mining, closed pattern mining and maximum pattern mining. These data mining techniques are not useful for text mining. This is due to lack of high specificity of discovered patterns. Not all frequent patterns discovered by mining algorithm are useful. Moreover then can be misinterpreted to make the problem worse. To overcome the problems of misinterpretation and low frequency, we proposed an effective pattern discovery. Pattern deploying and evolving are the two parts in the proposed technique. The empirical results revealed that the proposed technique is effective.

**CHAPTER 4**

**REQUIREMENTS**

**4.1 SOFTWARE REQUIREMENTS**

A set of programs associated with the operation of a computer is called software. Software is the part of the computer system, which enables the user to interact with several physical hardware devices.

The software requirement specifications for developing this project are as follows-

* Operating system : Windows XP and IIS
* Coding Language : ASP.Net with C# SP1
* Server : SQL Server 2008.

**4.1.1 Features Of .Net**

Microsoft .NET is a set of Microsoft software technologies for rapidly building and integrating XML Web services, Microsoft Windows-based applications, and Web solutions. The .NET Framework is a language-neutral platform for writing programs that can easily and securely interoperate. There’s no language barrier with .NET: there are numerous languages available to the developer including Managed C++, C#, Visual Basic and Java Script. The .NET framework provides the foundation for components to interact seamlessly, whether locally or remotely on different platforms. It standardizes common data types and communications protocols so that components created in different languages can easily interoperate.

“.NET” is also the collective name given to various software components built upon the .NET platform. These will be both products (Visual Studio.NET and Windows.NET Server, for instance) and services (like Passport, .NET My Services, and so on).

**4.1.2 The .Net Framework**

The .NET Framework has two main parts:

* The Common Language Runtime (CLR).
* A hierarchical set of class libraries.

The CLR is described as the “execution engine” of .NET. It provides the environment within which programs run. The most important features are

* Conversion from a low-level assembler-style language, called Intermediate Language (IL), into code native to the platform being executed on.
* Memory management, notably including garbage collection.
* Checking and enforcing security restrictions on the running code.
* Loading and executing programs, with version control and other such features.

The following features of the .NET framework are also worth description:

**Managed Code**

The code that targets .NET and which contains certain extra Information - “metadata” to describe itself. Whilst both managed and unmanaged code can run in the runtime, only managed code contains the information that allows the CLR to guarantee, for instance, safe execution and interoperability.

**Managed Data**

With Managed Code comes Managed Data. CLR provides memory allocation and Deal location facilities, and garbage collection. Some .NET languages use Managed Data by default, such as C#, Visual Basic.NET and JScript.NET, whereas others, namely C++, do not. Targeting CLR can, depending on the language you’re using, impose certain constraints on the features available. As with managed and unmanaged code, one can have both managed and unmanaged data in .NET applications - data that doesn’t get garbage collected but instead is looked after by unmanaged code.

**Common Type System**

The CLR uses something called the Common Type System (CTS) to strictly enforce type-safety. This ensures that all classes are compatible with each other, by describing types in a common way. CTS define how types work within the runtime, which enables types in one language to interoperate with types in another language, including cross-language exception handling. As well as ensuring that types are only used in appropriate ways, the runtime also ensures that code doesn’t attempt to access memory that hasn’t been allocated to it.

**Common Language Specification**

The CLR provides built-in support for language interoperability. To ensure that you can develop managed code that can be fully used by developers using any programming language, a set of language features and rules for using them called the Common Language Specification (CLS) has been defined. Components that follow these rules and expose only CLS features are considered CLS-compliant.

**4.1.3 The .Net Class Library**

.NET provides a single-rooted hierarchy of classes, containing over 7000 types. The root of the namespace is called System; this contains basic types like Byte, Double, Boolean, and String, as well as Object. All objects derive from System. Object. As well as objects, there are value types. Value types can be allocated on the stack, which can provide useful flexibility. There are also efficient means of converting value types to object types if and when necessary.

The set of classes is pretty comprehensive, providing collections, file, screen, and network I/O, threading, and so on, as well as XML and database connectivity.

The class library is subdivided into a number of sets (or namespaces), each providing distinct areas of functionality, with dependencies between the namespaces kept to a minimum.

**4.2 HARDWARE REQUIREMENTS**

The collection of internal electronic circuits and external physical devices used in building a computer is called the Hardware.

The minimum hardware requirement specifications for developing this project are as follows-

* System : Pentium IV 2.4 GHz.
* Hard Disk : 40 GB.
* Floppy Drive : 1.44 Mb.
* Ram : 512 Mb.

**CHAPTER 5**

**DESIGN**

**5.1 UML DIAGRAMS**

UML is a standard language for specifying, visualizing, constructing, and documenting the artefacts of software systems.

UML was created by Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997.

OMG is continuously putting effort to make a truly industry standard.

* UML stands for Unified Modeling Language.
* UML is different from the other common programming languages like C++, Java, COBOL etc.
* UML is a pictorial language used to make software blue prints.

So UML can be described as a general purpose visual Modeling language to visualize, specify, construct and document software system. Although UML is generally used to model software systems but it is not limited within this boundary. It is also used to model non software systems as well like process flow in a manufacturing unit etc.

UML is not a programming language but tools can be used to generate code in various languages using UML diagrams. UML has a direct relation with object oriented analysis and design. After some standardization UML is become an OMG (Object Management Group) standard.

UML can be described as the successor of object oriented analysis and design.

**5.1.1 Class Diagram**

The class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing and documenting different aspects of a system but also for constructing executable code of the software application.

So the purpose of the class diagram can be summarized as:

* Analysis and design of the static view of an application.
* Describe responsibilities of a system.
* Base for component and deployment diagrams.
* Forward and reverse engineering.

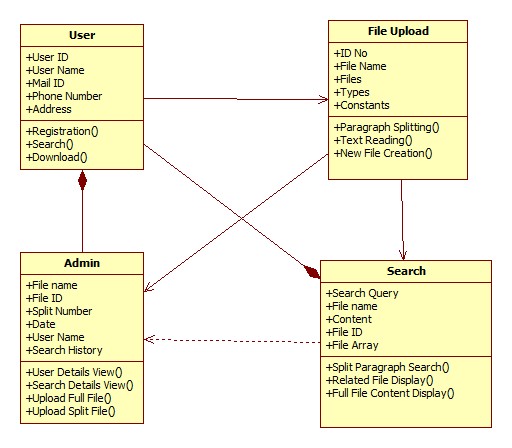


Fig 5.1 Class Diagram

**5.1.2 Usecase Diagram**

To model a system the most important aspect is to capture the dynamic behaviour. To clarify a bit in details, dynamic behaviour means the behaviour of the system when it is running /operating.

The purpose of use case diagram is to capture the dynamic aspect of a system. But this definition is too generic to describe the purpose. Because other four diagrams (activity, sequence, collaboration and Statechart) are also having the same purpose. So we will look into some specific purpose which will distinguish it from other four diagrams. So in brief, the purposes of use case diagrams can be as follows:

* Used to gather requirements of a system.
* Used to get an outside view of a system.
* Identify external and internal factors influencing the system.
* Show the interacting among the requirements are actors.

## How to draw Use Case Diagram?

Use case diagrams are considered for high level requirement analysis of a system. So when the requirements of a system are analysed the functionalities are captured in use cases.

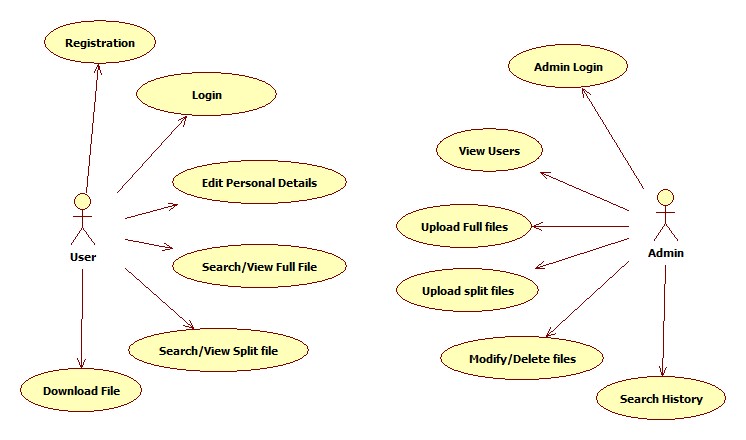
So we can say that uses cases are nothing but the system functionalities written in an organized manner. Now the second things which are relevant to the use cases are the actors. Actors can be defined as something that interacts with the system.

The actors can be human user, some internal applications or may be some external applications. So in a brief when we are planning to draw an use case diagram we should have the following items identified.

Use case diagrams are formally included in two modelling languages defined by the OMG: the [Unified Modelling Language (UML)](http://en.wikipedia.org/wiki/Unified_Modeling_Language) and the [Systems Modelling Language (SML)](http://en.wikipedia.org/wiki/Systems_Modeling_Language).

Use case is a set of scenarios that describing an interaction between a user and a system.  A use case diagram displays the relationship among actors and use cases.  The two main components of a use case diagram are use cases and actors.

* Functionalities to be represented as an use case
* Actors
* Relationships among the use cases and actors.

****

**Fig 5.2 Usecase Diagram**

**5.1.3 Activity Diagram**

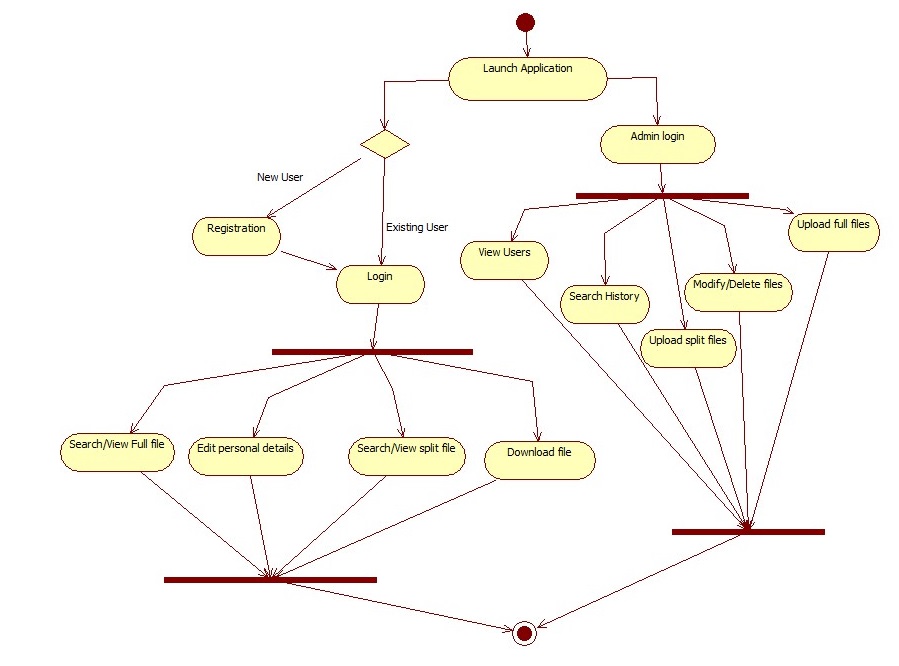
Activity diagram is another important diagram in UML to describe dynamic aspects of the system.

Activity diagram is basically a flow chart to represent the flow form one activity to another activity. The activity can be described as an operation of the system.

The basic purposes of activity diagrams are similar to other four diagrams. It captures the dynamic behavior of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another.

So the purposes can be described as-

* Draw the activity flow of a system.
* Describe the sequence from one activity to another.



**Fig 5.3 Activity Diagram**

**5.1.4 Sequence Diagram**

The diagram is used to describe some type of interactions among the different elements in the model. So this interaction is a part of dynamic behaviour of the system.

The purposes of interaction diagrams are to visualize the interactive behavior of the system. Now visualizing interaction is a difficult task. So the solution is to use different types of models to capture the different aspects of the interaction.

That is why sequence and collaboration diagrams are used to capture dynamic nature but from a different angle.

Sequence Diagrams are used primarily to design, document and validate the architecture, interfaces and logic of the system by describing the sequence of actions that need to be performed to complete a task or scenario.

UML sequence diagrams are useful design tools because they provide a dynamic view of the system behavior which can be difficult to extract from static diagrams or specifications.

Although UML sequence diagrams are typically used to describe object-oriented software systems, they are also extremely useful as system engineering tools to design system architectures, in business process engineering as process flow diagrams, as message sequence charts and call flows for telecom/wireless system design, and for protocol stack design and analysis.

So the purposes of interaction diagram can be describes as:

* To capture dynamic behaviour of a system.
* To describe the message flow in the system.
* To describe structural organization of the objects.
* To describe interaction among objects.



**Fig 5.4 Sequence Diagram**

**5.1.5 Collaboration Diagram**

A sophisticated modeling tool can easily convert a collaboration diagram into a sequence diagram and the vice versa. Hence, the elements of a Collaboration diagram are essentially the same as that of a Sequence diagram.

UML Collaboration diagrams (interaction diagrams) illustrate the relationship and interaction between software objects. They require use cases, system operation contracts, and domain model to already exist. The collaboration diagram illustrates messages being sent between classes and objects (instances). A diagram is created for each system operation that relates to the current development cycle (iteration).



**Fig 5.5 Collaboration Diagram**

**5.2 DATABASE TABLES**

**CHAPTER 6**

**IMPLEMENTATION**

**6.1 MODULES AND DESCRIPTION**

Implementation is the stage in the project where the theoretical design is turned into a working system. The implementation phase constructs, installs and operates the new system. The most crucial stage in achieving a new successful system is that it will work efficiently and effectively.

There are several activities involved while implementing a new project. They are

* **End user Training:**

The successful implementation of the new system will purely upon the involvement of the officers working in that department. The officers will be imparted the necessary training on the new technology

* **End User Education:**

The education of the end user start after the implementation and testing is over. When the system is found to be more difficult to understand and complex, more effort is put to educate the end used to make them aware of the system, giving them lectures about the new system and providing them necessary documents and materials about how the system can do this.

* **Training of application software:**

After providing the necessary basic training on the computer awareness, the users will have to be trained upon the new system such as the screen flows and screen design type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the way to correct the data entered. It should then cover information needed by the specific user or group to use the system.

* **Post Implementation View:**

The department is planning a method to know the states of the past implementation process. For that regular meeting will be arranged by the concerned officers about the implementation problem and success.

The modules of this project are-

**1.     Loading document**

* In this module we load the list of all documents.
* The user is allowed to retrieve one of the documents.
* This document is given to next process which is pre-processing.

**2.     Text pre-processing**

* The retrieved document pre-processing is done in this module.
* There are two types of process is done.
* 1)stop words removal 2)text stemming
* Stop words are words which are filtered out prior to, or after, processing of natural language data.
* Stemming is the process for reducing inflected (or sometimes derived) words to their stem base or root form. It generally a written word forms.

**3.     Pattern taxonomy process**

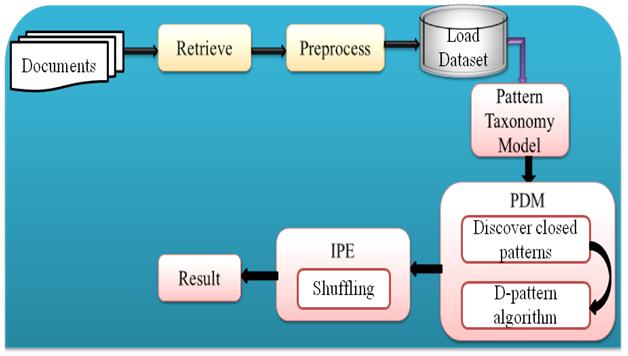
* In this module, the documents are split into paragraphs.
* Each paragraph is considered to be a document.
* In each document, the set of terms are extracted.
* The terms, which can be extracted, form a set of positive documents which are displayed.

**4.     Pattern deploying**

* The discovered patterns are summarized.
* The d-pattern mining algorithm is used to discover all patterns in positive documents are composed.
* The term supports are calculated by all terms in d-pattern.
* Term support means weight of the term is evaluated.

**5.     Pattern evolving**

* This module is used to identify the noisy patterns in documents.
* Sometimes, system falsely identifies negative document as a positive.
* So, noise is occurred in positive document and the noised pattern named as offender.
* If partial conflict offender contains in positive documents, the reshuffle process is applied.

[](http://2.bp.blogspot.com/-KSDbAacJOms/UIPM_levdBI/AAAAAAAABJM/trnaSx7Q2Zc/s1600/TEXTMINING.JPG)

**Fig 6.1 System architecture**

**6.2 TECHNOLOGIES USED**

**6.2.1 ASP.NET**

A Microsoft server-side Web technology- ASP.NET takes an object-oriented programming approach to Web page execution. Every element in an ASP.NET page is treated as an object and run on the server. An ASP.NET page gets compiled into an intermediate language by a .NET Common Language Runtime-compliant compiler. Then a JIT compiler turns the intermediate code to native machine code, and that machine code is eventually run on the processor because the code is run straight from the processor, pages load much faster than classic ASP pages, where embedded VBsricpt or Jscript had to be continuously interpreted and cached. ASP.NET is used to create Web pages and Web services and is an integral part of Microsoft's .NET vision. ASP.NET is a development framework for building web pages and web sites with HTML, CSS, JavaScript and server scripting.ASP.NET supports three different development models:

Web Pages, MVC (Model View Controller), and Web Forms:

The traditional ASP.NET event driven development model: Web pages with added server controls, server events, and server code. Web Pages is the easiest development model for developing ASP.NET web sites.

Web Forms is the traditional ASP.NET model, based on event driven Web Forms and post backs. Over the years, developers have used ASP.NET Web Forms to create many of the largest websites in the world.

ASP.NET is a unified Web development model that includes the services necessary for you to build enterprise-class Web applications with a minimum of coding. ASP.NET is part of the .NET Framework, and when coding ASP.NET applications you have access to classes in the .NET Framework. You can code your applications in any language compatible with the common language runtime (CLR), including Microsoft Visual Basic and C#. These languages enable you to develop ASP.NET applications that benefit from the common language runtime, type safety, inheritance, and so on.

|  |
| --- |
| If you want to try ASP.NET, you can install Visual Web Developer Express using the [Microsoft Web Platform Installer](http://go.microsoft.com/fwlink/?LinkId=153787), which is a free tool that makes it simple to download, install, and service components of the Microsoft Web Platform. These components include Visual Web Developer Express, Internet Information Services (IIS), SQL Server Express, and the .NET Framework. All of these are tools that you use to create ASP.NET Web applications. You can also use the Microsoft Web Platform Installer to install open-source ASP.NET and PHP Web applications. |

This topic describes the following features of ASP.NET and of Visual Web Developer, the development environment for creating ASP.NET applications.

* [Visual Web Developer](http://msdn.microsoft.com/en-us/library/4w3ex9c2.aspx#VisualWebDeveloper)
* [ASP.NET Web Sites and ASP.NET Web Application Projects](http://msdn.microsoft.com/en-us/library/4w3ex9c2.aspx#ASPNETWebSitesAndASPNETWebApplicationProjects)
* [ASP.NET API Reference](http://msdn.microsoft.com/en-us/library/4w3ex9c2.aspx#aspnet_api_reference)
* [Page and Controls Framework](http://msdn.microsoft.com/en-us/library/4w3ex9c2.aspx#PageAndControlsFramework)
* [ASP.NET Compiler](http://msdn.microsoft.com/en-us/library/4w3ex9c2.aspx#ASPNETCompiler)
* [Security Infrastructure](http://msdn.microsoft.com/en-us/library/4w3ex9c2.aspx#SecurityInfrastructure)
* [State-Management Facilities](http://msdn.microsoft.com/en-us/library/4w3ex9c2.aspx#StateManagementFacilities)
* [ASP.NET Configuration](http://msdn.microsoft.com/en-us/library/4w3ex9c2.aspx#ASPNETConfiguration)
* [Health Monitoring and Performance Features](http://msdn.microsoft.com/en-us/library/4w3ex9c2.aspx#HealthMonitoringAndPerformanceFeatures)
* [Debugging Support](http://msdn.microsoft.com/en-us/library/4w3ex9c2.aspx#DebuggingSupport)
* [Web Services Framework](http://msdn.microsoft.com/en-us/library/4w3ex9c2.aspx#XMLWebServicesFramework)
* [Extensible Hosting Environment and Application Life-Cycle Management](http://msdn.microsoft.com/en-us/library/4w3ex9c2.aspx#ExtensibleHostingEnvironment)
* [Extensible Designer Environment](http://msdn.microsoft.com/en-us/library/4w3ex9c2.aspx#ExtensibleDesignerEnvironment)
* [Web Applications Based on the MVC Pattern](http://msdn.microsoft.com/en-us/library/4w3ex9c2.aspx#MVC)
* ASP.NET Dynamic Data

[**Visual Web Developer**](javascript:void(0))

Visual Web Developer is a full-featured development environment for creating ASP.NET Web applications. Visual Web Developer offers you the following features:

* Web page design   A powerful Web page editor that includes WYSIWYG editing and an HTML editing mode with IntelliSense and validation.
* Page design features   Consistent site layout with master pages and consistent page appearance with themes and skins.
* Code editing   A code editor that enables you to write code for your dynamic Web pages in Visual Basic or C#. The code editor includes syntax coloration and IntelliSense.
* Testing and Debugging   A local Web server for testing and a debugger that helps you find errors in your programs.
* Deployment   Tools to automate typical tasks for deploying a Web application to a hosting server or a hosting provider.

### Testing and Debugging

Visual Web Developer provides an ideal environment in which to build Web sites and then publish them to a hosting site. Using the development tools in Visual Web Developer, you can develop ASP.NET Web pages on your own computer. Visual Web Developer includes a local Web server that provides all the features you need to test and debug ASP.NET Web pages, without requiring Internet Information Services (IIS) to be installed.

When your site is ready, you can publish it to the host computer using the built-in Copy Web tool, which transfers your files when you are ready to share them with others. Alternatively, you can precompile and deploy a Web site by using the Build Web Site command. The Build Web Site command runs the compiler over the entire Web site (not just the code files) and produces a Web site layout that you can deploy to a production server.

**6.2.2** [**ASP.NET Web Sites and ASP.NET Web Application Projects**](javascript:void(0))

Using Visual Studio 2010, you can create different types of ASP.NET projects, which includes Web sites, Web applications, Web services, and AJAX server controls.

There is a difference between Web site projects and Web application projects. Some features work only with Web application projects, such as MVC and certain tools for automating Web deployment. Other features, such as Dynamic Data, work with both Web sites and Web application projects. For more information about the differences between Web application projects and Web site projects, see [Web Application Projects versus Web Site Projects](http://msdn.microsoft.com/en-us/library/dd547590.aspx).

**6.2.3** [**ASP.NET API Reference**](javascript:void(0))

Some of the most important namespaces in the .NET Framework class library that pertain to ASP.NET are the following:

* [**System.Web**](http://msdn.microsoft.com/en-us/library/system.web.aspx)

Provides classes and interfaces that enable browser-server communication. This namespace includes the [HttpRequest](http://msdn.microsoft.com/en-us/library/system.web.httprequest.aspx) class, which provides extensive information about the current HTTP request, the [HttpResponse](http://msdn.microsoft.com/en-us/library/system.web.httpresponse.aspx) class, which manages HTTP output to the client, and the [HttpServerUtility](http://msdn.microsoft.com/en-us/library/system.web.httpserverutility.aspx) class, which provides access to server-side utilities and processes. [System.Web](http://msdn.microsoft.com/en-us/library/system.web.aspx) also includes classes for cookie manipulation, file transfer, exception information, and output cache control.

* [**System.Web.ApplicationServices**](http://msdn.microsoft.com/en-us/library/system.web.applicationservices.aspx)

Provides classes that provide access to ASP.NET forms authentication, roles, and profiles application services as Windows Communication Foundation (WCF) services.

* [**System.Runtime.Caching**](http://msdn.microsoft.com/en-us/library/system.runtime.caching.aspx)

Contains types that let you implement caching in .NET Framework applications.

* [**System.Web.ClientServices**](http://msdn.microsoft.com/en-us/library/system.web.clientservices.aspx)

Contains classes that support access to the ASP.NET login, roles, and profiles services from Windows-based applications.

* [**System.Web.Configuration**](http://msdn.microsoft.com/en-us/library/system.web.configuration.aspx)

Contains classes that are used to programmatically manage ASP.NET configuration. (Most configuration settings can be made in XML files.)

* [**System.Web.DynamicData**](http://msdn.microsoft.com/en-us/library/system.web.dynamicdata.aspx)

Contains classes that provide the core functionality for ASP.NET dynamic data and extensibility features that let you customize dynamic data behaviour.

* [**System.Web.Handlers**](http://msdn.microsoft.com/en-us/library/system.web.handlers.aspx)

Contains HTTP handler classes that process HTTP requests to a Web server. (An ASP.NET Web Forms page -- .aspx file -- is a special form of an HTTP handler.)

* [**System.Web.Management**](http://msdn.microsoft.com/en-us/library/system.web.management.aspx)

Contains classes and interfaces for managing and monitoring the health of Web applications.

* [**System.Web.Profile**](http://msdn.microsoft.com/en-us/library/system.web.profile.aspx)

Contains classes that are used to implement the ASP.NET user profile in Web server applications.

* [**System.Web.Query.Dynamic**](http://msdn.microsoft.com/en-us/library/system.web.query.dynamic.aspx)

Contains classes that are used to parse expressions from a [LinqDataSource](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.linqdatasource.aspx) control into a Language-Integrated Query (LINQ).

* [**System.Web.RegularExpressions**](http://msdn.microsoft.com/en-us/library/system.web.regularexpressions.aspx)

Provides regular expressions that are used to parse ASP.NET files. All members of the [System.Web.RegularExpressions](http://msdn.microsoft.com/en-us/library/system.web.regularexpressions.aspx) namespace are descendants of the [Regex](http://msdn.microsoft.com/en-us/library/system.text.regularexpressions.regex.aspx) class. (You typically do not have to parse ASP.NET pages yourself.)

* [**System.Web.Routing**](http://msdn.microsoft.com/en-us/library/system.web.routing.aspx)

Provides classes that are used with URL routing, which enables you to use URLs that do not map to a physical file.

* [**System.Web.Script**](http://msdn.microsoft.com/en-us/library/system.web.script.aspx)

Contains classes that provide client-script resource information.

* [**System.Web.Script.Services**](http://msdn.microsoft.com/en-us/library/system.web.script.services.aspx)

Provides attributes to customize Web service support for using Ajax functionality in ASP.NET.

* [**System.Web.Security**](http://msdn.microsoft.com/en-us/library/system.web.security.aspx)

Contains classes that are used to implement ASP.NET security in Web server applications.

* [**System.Web.Services**](http://msdn.microsoft.com/en-us/library/system.web.services.aspx)

Consists of the classes that enable you to create XML Web services using ASP.NET and XML Web service clients. XML Web services are applications that provide the ability to exchange messages in a loosely coupled environment using standard protocols such as HTTP, XML, XSD, SOAP, and WSDL. XML Web services let you build modular applications that are interoperable across a broad variety of implementations, platforms, and devices.

* [**System.Web.SessionState**](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.aspx)

Contains classes and interfaces that enable storage of data specific to a single client during a single browser session on the server. Session state data is used to give the client the appearance of a persistent connection with the application.

**6.2.4 C#.NET**

**C#** (pronounced as *see sharp*) is a [multi-paradigm programming language](http://en.wikipedia.org/wiki/Multi-paradigm_programming_language) encompassing [strong typing](http://en.wikipedia.org/wiki/Strong_typing), [imperative](http://en.wikipedia.org/wiki/Imperative_programming), [declarative](http://en.wikipedia.org/wiki/Declarative_programming), [functional](http://en.wikipedia.org/wiki/Functional_programming), [procedural](http://en.wikipedia.org/wiki/Procedural_programming), [generic](http://en.wikipedia.org/wiki/Generic_programming), [object-oriented](http://en.wikipedia.org/wiki/Object-oriented_programming) ([class](http://en.wikipedia.org/wiki/Class_%28computer_science%29)-based), and [component-oriented](http://en.wikipedia.org/wiki/Component-based_software_engineering) programming disciplines. It was developed by [Microsoft](http://en.wikipedia.org/wiki/Microsoft) within its [.NET](http://en.wikipedia.org/wiki/.NET_Framework) initiative and later approved as a standard by [Ecma](http://en.wikipedia.org/wiki/Ecma_International) (ECMA-334) and [ISO](http://en.wikipedia.org/wiki/International_Organization_for_Standardization) (ISO/IEC 23270:2006). C♯ is one of the programming languages designed for the [Common Language Infrastructure](http://en.wikipedia.org/wiki/Common_Language_Infrastructure). C♯ is built on the syntax and semantics of C++, allowing C programmers to take advantage of .NET and the common language runtime.

C# syntax is highly expressive, yet it is also simple and easy to learn. The curly-brace syntax of C# will be instantly recognizable to anyone familiar with C, C++ or Java. Developers who know any of these languages are typically able to begin to work productively in C# within a very short time. C# syntax simplifies many of the complexities of C++ and provides powerful features such as nullable value types, enumerations, delegates, lambda expressions and direct memory access, which are not found in Java. C# supports generic methods and types, which provide increased type safety and performance, and iterators, which enable implementers of collection classes to define custom iteration behaviours that are simple to use by client code. Language-Integrated Query (LINQ) expressions make the strongly-typed query a first-class language construct.

As an object-oriented language, C# supports the concepts of encapsulation, inheritance, and polymorphism. All variables and methods, including the Main method, the application's entry point, are encapsulated within class definitions. A class may inherit directly from one parent class, but it may implement any number of interfaces. Methods that override virtual methods in a parent class require the override keyword as a way to avoid accidental redefinition. In C#, a struct is like a lightweight class; it is a stack-allocated type that can implement interfaces but does not support inheritance.

In addition to these basic object-oriented principles, C# makes it easy to develop software components through several innovative language constructs, including the following:

* Encapsulated method signatures called delegates, which enable type-safe event notifications.
* Properties, which serve as accessors for private member variables.
* Attributes, which provide declarative metadata about types at run time.
* Inline XML documentation comments.
* Language-Integrated Query (LINQ) which provides built-in query capabilities across a variety of data sources.

If you have to interact with other Windows software such as COM objects or native Win32 DLLs, you can do this in C# through a process called "Interop." Interop enables C# programs to do almost anything that a native C++ application can do. C# even supports pointers and the concept of "unsafe" code for those cases in which direct memory access is absolutely critical.

The C# build process is simple compared to C and C++ and more flexible than in Java. There are no separate header files, and no requirement that methods and types be declared in a particular order. A C# source file may define any number of classes, structs, interfaces, and events.

The following are additional C# resources:

* For a good general introduction to the language, see Chapter 1 of the [C# Language Specification](http://msdn.microsoft.com/en-us/library/ms228593.aspx).
* For detailed information about specific aspects of the C# language, see the [C# Reference](http://msdn.microsoft.com/en-us/library/618ayhy6.aspx).
* For more information about LINQ, see [LINQ (Language-Integrated Query)](http://msdn.microsoft.com/en-us/library/bb397926.aspx).
* To find the latest articles and resources from the Visual C# team, see the [Visual C# Developer Center](http://go.microsoft.com/fwlink/?LinkId=47811).

C# is Microsoft’s new language. It’s a C-style language that is essentially “C++ for Rapid Application Development”. Unlike other languages, its specification is just the grammar of the language. It has no standard library of its own, and instead has been designed with the intention of using the .NET libraries as its own.

Microsoft Visual C# .NET provides the easiest transition for Java-language developers into the world of XML Web Services and dramatically improves the interoperability of Java-language programs with existing software written in a variety of other programming languages.

Active State has created Visual Perl and Visual Python, which enable .NET-aware applications to be built in either Perl or Python. Both products can be integrated into the Visual Studio .NET environment. Visual Perl includes support for Active State’s Perl Dev Kit.

Other languages for which .NET compilers are available include

* FORTRAN
* COBOL
* Eiffel

|  |  |
| --- | --- |
| ASP.NET  XML WEB SERVICES | Windows Forms |
| Base Class Libraries | |
| Common Language Runtime | |
| Operating System | |

**Fig 6.2 .Net Framework**

C#.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.

C#.NET is a CLS-compliant language. Any objects, classes, or components that created in C#.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 C#.NET .The use of CLS ensures complete interoperability among applications, regardless of the languages used to create the application.

* **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 C#.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. In addition, the sub finalize procedure can be called only from the class it belongs to or from derived classes.

* **GARBAGE COLLECTION**

Garbage Collection is another new feature in C#.NET. The .NET Framework monitors allocated resources, such as objects and variables. In addition, the .NET Framework automatically releases memory for reuse by destroying objects that are no longer in use.

In C#.NET, the garbage collector checks for the objects that are not currently in use by applications. When the garbage collector comes across an object that is marked for garbage collection, it releases the memory occupied by the object.

* **OVERLOADING**

Overloading is another feature in C#. 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.

* **MULTITHREADING**

C#.NET also supports multithreading. An application that supports multithreading can handle multiple tasks simultaneously, we can use multithreading to decrease the time taken by an application to respond to user interaction.

* **STRUCTURED EXCEPTION HANDLING**

C#.NET supports structured handling, which enables us to detect and remove errors at runtime. In C#.NET, we need to use Try…Catch…Finally statements to create exception handlers. Using Try…Catch…Finally statements, we can create robust and effective exception handlers to improve the performance of our application.

* **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.

**6.2.4 SQL SEVER**

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,

They are,

1. TABLE

2. QUERY

3. FORM

4. REPORT

5. MACRO

**TABLE:**

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

**VIEWS OF TABLE:**

We can work with a table in two types,

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.

**6.3 SCREEN SHOTS**

**Fig 6.3 Login screen**

**Fig 6.2 Pop up box on not entering username**

**Fig 6.3**

**Fig 6.4 Pop up box if invalid username is entered**

**Fig 6.5 Home Screen**

**Fig 6.6 Create Account Frame**

**Fig 6.7 Adding Password to Password Wallet**

**Fig 6.8 if passwords don’t match while adding password to Password Wallet**

**Fig 6.9 Passwords successfully added**

**Fig 6.10 Modifying passwords**

**Fig 6.11 Deleting Passwords from Password Wallet**

**Fig 6.12 Confirmation for deleting password**

**Fig 6.13 Retrieving Passwords by entering the appropriate title**

**Fig 6.14 Changing Login Password**

**Fig 6.15 Forgot Password Frame**

**Fig 6.16 Logging out of Password Wallet**

**Fig 6.15 MAIN\_TB table in the database**

**CHAPTER 7**

**TESTING**

**7.1 PURPOSE**

The purpose of testing is to assess product quality. It helps to strengthen and stabilize the architecture early in the development cycle. We can verify through testing, the various interactions, integration of components and the requirements which were implemented. It provides timely feedback to resolve the quality issues, in a timely and cost effective manner. The test workflow involves the following:

* Verifying the interactions of components.
* Verifying the proper integration of components.
* Verifying that all requirements have been implemented correctly.
* Identifying and ensuring that all discovered defects are addressed before the software is deployed.

**7.2 DIMENSIONS OF TESTING**

To assess product quality, different kinds of tests, each one with a different focus, are needed. These tests can be categorized by several dimensions:

* **Quality dimension**: The major quality characteristic or attribute that is the focus of test.
* **Stage of testing**: The point in the lifecycle at which the test, usually limited to a single quality dimension.
* **Type of testing**: The specific test objective for an individual test, usually limited to a single quality dimension.

**7.3 TYPES OF TESTING**

**7.3.1 Unit Testing**

Unit testing verification efforts on the smallest unit of software design, module. This is known as “Module Testing”. The modules are tested separately. This testing is carried out during programming stage itself. In these testing steps, each module is found to be working satisfactorily as regard to the expected output from the module.

**7.3.2 Integration Testing**

Integration testing is a systematic technique for constructing tests to uncover error associated within the interface. In the project, all the modules are combined and then the entire programmer is tested as a whole. In the integration-testing step, all the error uncovered is corrected for the next testing steps.

**7.3.3 White Box Testing**

The purpose of any security testing method is to ensure the robustness of a system in the face of malicious attacks or regular software failures. White box testing is performed based on the knowledge of how the system is implemented. White box testing includes analyzing data flow, control flow, information flow, coding practices, and exception and error handling within the system, to test the intended and unintended software behavior. White box testing can be performed to validate whether code implementation follows intended design, to validate implemented security functionality, and to uncover exploitable vulnerabilities. White box testing requires access to the source code.

It is a good practice to perform white box testing during the unit testing phase. White box testing requires knowing what makes software secure or insecure, how to think like an attacker, and how to use different testing tools and techniques.

The first step in white box testing is to comprehend and analyze source code, so knowing what makes software secure is a fundamental requirement. Second, to create tests that exploit software, a tester must think like an attacker. Third, to perform testing effectively, testers need to know the different tools and techniques available for white box testing. The three requirements do not work in isolation, but together.

**7.3.4 Black Box Testing**

It is also known as functional testing- a software testing technique whereby the internal workings of the item being tested are not known by the tester. For example, in a black box test on software design the tester only knows the inputs and what the expected outcomes should be and not how the program arrives at those outputs. The tester does not ever examine the programming code and does not need any further knowledge of the program other than its specifications.

The advantages of this type of testing include:

* The test is unbiased because the designer and the tester are independent of each other.
* The tester does not need knowledge of any specific programming languages.
* The test is done from the point of view of the user, not the designer.
* Test cases can be designed as soon as the specifications are complete.

**7.3.5 System Testing**

System testing validates software once it has been incorporated into a larger system. Software is incorporated with other system elements and a series of system integration and validation tests are conducted. System testing is actually a series of different test whose primary purpose is to fully exercise the computer- based system. Once the system has been developed it has to be tested.

In the present system we have to take care of valid property and assessment numbers i.e. there should not exist any duplicate number in each case. Care should be taken that the appropriate data is retrieved in response to the queries.

**7.3.6 Validation Testing**

The terms verification and validations are used interchangeably we will describe both these methods. Verification is the process of determining whether or not the products of given phase of software development fulfil the specifications established in the previous phase. These activities include proving and reviews. Validation is the process of evaluating the software at the end of software development process; we find how well the software satisfies the requirement specifications. The requirement of the software starts with requirement document and requirement specifications without errors and specifying client’s requirements correctly. The validation process of evaluating the developed system at the end is to ensure that it must satisfy all the necessary requirement specification. Requirement verification also checks the factors as completeness, consistency and testability of the requirements. As we all know that testing plays a crucial role in evaluation of the system. That is in order to know whether the system working properly or not. In other words we can say that in order to know whether the system which we have developed will give the expected output or not can be known by doing the testing. Testing phase comes after coding phase. Usually organizations or the software developing companies use different types of testing strategies in order to evaluate the performance of a system. Also it gives the output which provides clear information regarding the project or system, whether the project which we have developed will going to give the expected output or not, that is whether the system fails or succeed in the market. We have many types of testing such as unit testing, integration testing, system testing, and black box testing, white box testing and regression analysis testing and so on. In our project secure cryptographic messaging we are using unit testing, integration testing, and system testing. Unit testing is the one in which each entity or objects in the module will be tested.

That’s once unit testing is done with all modules, than integration testing will be done, on the every module or on group of two or three modules. Finally system testing will be done , in which all the modules of a system will be tested at once , there by getting the overall performance of a system that means we can conclude the result on the entire system whether our system is working as per our requirements or as per our expectations or not. The advantage of developing or testing modules wise is that, we can reduce the effort, cost and time. Because if we are testing module wise than we can know clearly which module is working fine and which module is not working, thereby the module which is not working perfectly can be evaluated once again by going necessary modifications unlike the system being tested on a whole, where if any errors comes in than the entire system need to be tested or evaluated which consumes more effort, time and cost.

**7.4 TEST CASES**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Function** | **Expected Result** | **Observed Result** | **Status** | **Reference** |
| 1. | Checking if username field is entered or not. | Error-message requesting the user to enter the user name should pop up. | If username field is empty, error message pops up. | Pass | Refer to Fig 6.2 |
| 2. | Checking if password field is entered or not. | Error-message requesting the user to enter the password should pop up. | If password field is empty, error message pops up. | Pass | Refer to Fig 6.3 |
| 3. | Authentication | If given credentials are valid, then redirect to the home-page. | Redirected to the respective page. | Pass | Refer to Fig 6.4 |
| 4. | Retrieving passwords on giving the accurate titles. | If the title given by user matches the title in database then retrieve the password. | The password is retrieved from the database. | Pass | Refer to Fig 6.12 |
| 5. | Checking if username is valid or not. | If entered username does not match the username in database then error message pops up. | If username is not valid, an error message pops up. | Pass | Refer to Fig 6.5 |
| 6. | Checking if entered password and confirmation passwords match. | If entered password does not match confirmation password then an error message pops up. | If passwords don’t match then an error message pops up. | Pass | Refer to Fig 6.8 |

**CHAPTER 8**

**CONCLUSION AND FUTURE ENHANCEMENT**

Many data mining techniques have been proposed in the last decade. These techniques include association rule mining, frequent item set mining, sequential pattern mining, maximum pattern mining, and closed pattern mining. However, using these discovered knowledge (or patterns) in the field of text mining is difficult and ineffective. The reason is that some useful long patterns with high specificity lack in support (i.e., the low-frequency problem). We argue that not all frequent short patterns are useful. Hence, misinterpretations of patterns derived from data mining techniques lead to the ineffective performance. In this research work, an effective pattern discovery technique has been proposed to overcome the low-frequency and misinterpretation problems for text mining. The proposed technique uses two processes, pattern deploying and pattern evolving, to refine the discovered patterns in text documents. The experimental results show that the proposed model outperforms not only other pure data mining-based methods and the concept based model, but also term-based state-of-the-art models, such as BM25 and SVM-based models.

**REFERENCES**

[1] [Beginning ASP.NET 4: in C# and VB](http://www.amazon.com/gp/product/0470502215?ie=UTF8&tag=aspnettelligent-20&linkCode=as2&camp=1789&creative=9325&creativeASIN=0470502215)by *Imar Spaanjaars*.

[2] [Programming ASP.NET 3.5](http://www.amazon.com/gp/product/0596529562?ie=UTF8&tag=aspnettelligent-20&linkCode=as2&camp=1789&creative=9325&creativeASIN=0596529562)*by Jesse Liberty, Dan Maharry, Dan Hurwitz.*

[3] C. Wang, N. Cao, J. Li, K. Ren, and W. Lou, “Secure ranked keyword search over encrypted cloud data,” in Proc. of ICDCS’10,2010.

[4] Software Engineering: A Practitioner’s Approach, Roger S Prressman, 6th edition

[5] Grady Booch, James Rumbaugh, Ivar Jacobson: The Unified Modeling Language User Guide

[6] S. Kamara and K. Lauter, “Cryptographic cloud storage,” in RLCPS, January 2010, LNCS. Springer, Heidelberg.

[7] <http://www.asp.net.com>

[8] <http://www.dotnetspider.com/>

[9] <http://www.dotnetspark.com>