

# **Web Usage Mining**

**A Dissertation submitted to The University of Manchester  
for the degree of Msc. Software Engineering**

**In the faculty of Humanities**

**2007**

**Waqqas Naqi**

**School of Informatics**

# Table of Contents

<b>Chapter 1</b>	<b>11</b>
1.1 Introduction	11
1.2 Problem Domain Description	12
1.3 Investigation of the project	13
1.4 Dissertation Objectives	13
1.5 Deliverables of the project	14
1.6 Dissertation Structure	14
1.7 Summary	16
<b>Chapter 2</b>	<b>17</b>
2.1 Introduction	Σφάλμα! Δεν έχει οριστεί σελιδοδείκτης.
2.2 Web Site Maintenance	18
2.3 Knowledge Discovery in Databases	20
2.3.1 KDD and Data Mining	20
2.3.2 Steps of KDD Process	21
2.4 Data mining	22
2.5 Data mining and World Wide Web	24
2.5.1 Data Sources	25
2.6 Enhancing Information	26
2.6.1 Pre-processing	27
2.6.1.1 Usage Pre-processing	27
2.6.1.2 Content Pre-Processing	27
2.6.1.3 Structure Pre-processing	27
2.6.2 Pattern Discovery	28

2.6.2.1	Association Rules .....	28
2.6.2.2	Classification .....	29
2.6.2.3	Clustering .....	30
2.7	Data Transformation .....	33
2.8	Applications of Data Mining .....	34
2.9	Summary .....	34
<b>Chapter 3</b>	<b>.....</b>	<b>35</b>
3.1	Introduction .....	35
3.2	Requirements .....	36
3.2.1	Functional Requirements .....	37
3.2.2	Non-Functional Requirements .....	39
3.3	Success Criteria .....	39
3.3.1	Objectives .....	40
3.4	Summary .....	41
<b>Chapter 4</b>	<b>.....</b>	<b>42</b>
4.1	Introduction .....	42
4.2	Steps Involved .....	43
4.2.1	Programmer's Activity Diagram .....	44
4.3	Basic System Design .....	45
4.3.1	Actor Diagram .....	45
4.3.2	System Flow Diagram .....	46
4.3.2.1	Initiator & Source files .....	46
4.3.2.2	Clean Collect & Summaries .....	46
4.3.2.3	Extract Useful Information .....	47
4.3.2.4	Storage of data on persistent storage .....	47

4.3.2.5 Apply DM Algorithm.....	47
4.3.2.6 Output File .....	48
4.3.3 System Use Case .....	48
4.3.3.1 The Initialization Operation.....	48
4.3.3.2 The Data mining Algorithm.....	49
4.3.3.3 State Format .....	50
4.3.3.4 Summary of the Client Application functions .....	51
4.3.4 Design of the user-program interaction .....	52
4.3.5 Design of the Data Extraction Algorithm.....	54
4.3.6 Design of the Data Mining Algorithm (K-Means).....	55
4.3.7 Functionality of the System.....	57
4.4 System's sequence Diagram .....	61
4.5 Summary.....	62
<b>Chapter 5.....</b>	<b>63</b>
5.1 Introduction .....	63
5.2 High Level Programming Specifications .....	64
5.3 Implementation Phase and Tools .....	65
5.4 Implementation of the Model of Input Data .....	66
5.5 Main Interface .....	69
5.6 The Pre-Processing Procedure.....	71
5.6.1 Open/Define Format .....	72
5.6.2 Choose File .....	72
5.6.3 Extraction of Data.....	74
5.6.4 Apply Similarity Matrix .....	76
5.6.5 Calculate Average Similarity .....	78

5.6.6 K-means Algorithm .....	80
5.7 Summary .....	80
<b>Chapter 6 .....</b>	<b>81</b>
6.1 Introduction .....	81
6.2 Testing the Web Usage Mining Tool .....	82
6.3 Evaluation.....	87
6.3.1 Evaluation in terms of Architecture .....	87
6.3.2 Evaluation in terms of Application's functionality .....	87
6.3.4 Evaluation in terms of Traversal Speed .....	87
6.3.5 Evaluation in terms of Database .....	88
6.3.6 Experimental Results.....	89
6.4 System's Drawbacks .....	92
6.5 Summary.....	92
<b>Chapter 7.....</b>	<b>93</b>
7.1 Introduction .....	93
7.2 Dissertation Structure .....	94
7.3 Development Methodology .....	94
7.4 Lessons Learned.....	95
7.5 Summary of the Project.....	96
7.6 Future Work .....	97
7.7 Summary .....	98
<b>A.) REFERENCES .....</b>	<b>99</b>
<b>Appendix A.....</b>	<b>102</b>
<b>Appendix B.....</b>	<b>105</b>
<b>Appendix C.....</b>	<b>109</b>

<b><i>List of Figures</i></b>	<b><i>Page No.</i></b>
Fig 2.1: Web site Life Cycle	19
Fig 2.2: Data mining as a step in KDD	21
Fig 2.3 Architecture of Typical Data mining System	23
Fig 2.4 High Level Web usage Mining Process	26
Fig 2.5: A classification model given test data	29
Fig 2.6: Traditional Hierarchal clustering	30
Fig 2.7: Euclidean Distance	31
Fig 2.8: Working of K-means Algorithm	32
Fig 2.9: Similarity Matrix	33
Fig 3.1: Sample Server Log File	38
Fig 4.1: Programmer's Activity Diagram	44
Fig 4.2: Initial operation Actor Diagram	45
Fig 4.3: System flow diagram	46
Fig 4.4: Use case Diagram of Initialization Operation	48
Fig 4.5: Use case diagram of Applying Algorithm	49
Fig 4.6: Use case diagram of Applying Algorithm on Data	50
Fig 4.7: Summary of the Client Application Functions	51
Fig 4.8: Defining User's Own Format	52
Fig 4.9: Defining User's Own Format	53
Fig 4.10: Pre-Processing Algorithm	54
Fig 4.11: Data Mining Algorithm	56
Fig 4.12 System Class Diagram	57
Fig 4.13 User- System Sequence Diagram	61
Fig: 5.1 Structure of the System	65
Fig: 5.2 LogAnalyzerDb Schema	66
Fig: 5.3 Database view of table Format	67
Fig: 5.4 Database view of table Clustering	68
Fig: 5.5 Database view of table LogData	69
Fig: 5.6 The Define Format interface	69
Fig: 5.7: Structure of implementation phase	71
Fig 5.8: Database view of entries in table LogData	78
Fig 5.9: Database view of entries in table Clustering	79

Fig 5.10: Similarity averages of one field with others	79
Fig 6.1: Saved Format window	82
Fig 6.2: Define Format window	83
Fig 6.3: User defined format in DB	83
Fig 6.4: Extracted data in LogData table	84
Fig 6.5: Similarity Values in front of Id's	85
Fig 6.6: Clustering information in Clustering Table	85
Fig 6.7: User input for number of clusters	86
Fig 6.8: Result generation having four clusters	86
Fig 6.9: SQL Server Scalability	88
Fig 6.10: point Graph for LogData1	89
Fig 6.11: Column Graph for LogData1	90
Fig 6.12: Point Graph for LogData2	91
Fig 6.13: Column Graph for LogData2	91
Fig 7.1 Wrong User input	97

## **ABSTRACT**

With billions of documents online and millions of them being published each day, the Web has become a massive source of information in the present era. In response, the job for web site publishers to certain customers is becoming more and more difficult. This is where web usage mining comes into action. This work is a Dissertation report of a project concerning the development of a Web usage mining application. The primary objective of the system is to develop a web usage mining system which generates facts and figures on the bases of information given to it. A web log file is given as an input to the system on the basis of which the system generates performance statistics. These statistics can be useful to check user access patterns. The application will be developed using C# on a .Net platform having SQL server 2003 database management system at its backend. This report documents the Requirements, Design and Implementation phase. System is tested by applying different test cases and results are then evaluated.

## **Declaration**

*No portion of the work referred to in this dissertation has been submitted in support of an application for another degree or qualification of this or any other university or other institutions of learning.*

*Waqqas Naqi*

*2007*

## **Copyright Statement**

Copyright in text of this dissertation rests with the author. Copies (by any process) either in full, or of extracts, may be made only in accordance with instructions given by the author. Details may be obtained from the appropriate Graduate Office. This page must form part of any such copies made. Further copies (by any process) of copies made in accordance with such instructions may not be made without the permission (in writing) of the author.

The ownership of any intellectual property rights which may be described in this dissertation is vested in the University of Manchester, subject to any prior agreement to the contrary, and may not be made available for use by third parties without the written permission of the University, which will prescribe the terms and conditions of any such agreement.

Further information on the conditions under which disclosures and exploitation may take place is available from the Head of the School of Informatics.

## **Acknowledgements**

I would like to express my deepest gratitude to my supervisor, Dr Christos Tjortjis, for his continuous and kind support to me throughout the whole project phase. His guidance and advice made the difference especially when I came through different problems.

I would also like to give my sincere regards to some of my friends for the nice cooperation we had throughout this year.

I am grateful to my family and especially my father, for his continuous support they have provided me throughout this tough phase.

Waqqas Naqi

2007

# ***Chapter 1***

## ***Introduction***

### **1.1 Introduction**

The chapter is concerned with in depth description of the problem domain of the project. The chapter provides an overview of the report and the fundamental concepts concerning it. Finally, the structure and the chapters included in the report are discussed.

## **1.2 Problem Domain Description**

With billions of documents online and millions of them being published each day, the Web has become a massive source of information in the present era. In response, the job for web site publishers to certain customers is becoming more and more difficult. In order to be useful and have added value, these websites should meet certain criteria such as to operate correctly, be flexible and always available. To meet the requirements defined above, websites subject to several changes during their whole life time depending on user requirements. Most vital part is the management of these changes as a lot of effort, time and money is required to apply them. Web usage information can be very useful for a developer to analyze user access patterns on a website. This useful information can give direction to a business web model to analyze the traffic on it and then determine effective marketing strategies across products; secondly it helps to describe the logical structure of a website to a designer. The objective of developing Web usage mining system is to capture the behavioural patterns and profiles of users interacting with a Web site [12]. The main sources of information which can be used to analyze the performance of a web site are the web log files kept by web servers. Web log files are always in raw form, the raw data is first cleaned then usage patterns are extracted from them identifying many attributes like IP address, User Id, protocol used and time spent etc. The most widely used log file formats are, implied by [13], the Common Log File format (CLF) [14] and the Extended Log File format (ExLF) [15]. A web log file contains the information like IP address, Http request, authentication name, response status, time stamps, request size, referral's URL and web browsers. The work also discusses the notion of KDD (knowledge Discovery in Databases) which can be beneficial in maintenance of a web site. KDD elaborates how a mining technique can be applied and help a maintainer to maintain a system i.e. a website in this case.

### **1.3 Investigation of the project**

The web usage mining tool receives a web log file as its input; the log file contains data in raw form. A web usage mining tool is needed to clean data, formalize it and apply mining techniques on it to generate results. Web usage mining is a process of picking up information from user how to use web sites [11]. Reason for choosing Web log data to apply data mining is, since it is a new and an emerging field. Despite the reason of this newly emerging field, the project's purpose is to explore and generate those facts and figures which could help a publisher to improve a website.

### **1.4 Dissertation Objectives**

The main aim of the project is to show that how web site maintenance can be assisted with the help of data mining techniques such as clustering.

Main Objectives are:

- ✓ Design and Implementation of the Pre-processing Application. Pre-processing means extraction of data from the web log files. This information is then stored in the database created. The application consists of the following modules.
  - Front End, with which the user interacts directly and initiates different functionalities.
  - Back End in which extraction of data from the given web log file and inserting into the database. Generation of results is also a part of the backend.
- ✓ Design of the underlined database schema. The data stored in the database used by the data mining algorithm such as clustering.

- ✓ Using the Application in order to generate results useful statistics.
- ✓ Evaluating the Pre-Processing application to check whether the generated results are useful for the maintainer or not.
- ✓ Choice of the data mining algorithm which will play the most important role in evaluating the extracted data and outcome of the developed product.
- ✓ Developing a web usage mining tool, having user friendly interfaces for the user.

## **1.5 Deliverables of the project**

The dissertation will produce a web usage mining tool written in c#, for parsing, analyzing and mining usage data contained in a web log file. The reason of choosing the programming language and certain issues are explained in further chapters of the report.

The whole procedure is structured as:

- i) Define format of a log file.
- ii) Introduce log file to the system
- iii) Parse the file, clean it and insert data into database
- iv) Apply data mining algorithm
- v) Display facts and figures which are used by developers and stake holders.

## **1.6 Dissertation Structure**

- ✓ The first chapter, *Introduction*, states the problem domain and the issues relating this dissertation. Main objectives of the dissertation are also discussed in the introductory chapter

- ✓ The second chapter, *Background*, includes a detailed description of the problem domain. Detailed notions of web mining are also discussed in the background chapter. The chapter is also concerned with the use of Knowledge discovery in data bases, and more specifically Data mining techniques in order to explain the understanding of web log data.
- ✓ The chapter, *Success Criteria*, is concerned with the circumstances under which the work is regarded as successful. This chapter gives a detailed description for the requirements of the system both functional and non-functional. The chapter sets the foundation of the development process for further use.
- ✓ The, *Design*, chapter is concerned with presenting the design of the system and justifies the success criteria. The client application and its relation with the underlined database are discussed in detail with the help of UML (unified modeling diagrams). These figures depict the system and the database on abstract as well as in highly detailed level.
- ✓ The *Implementation* chapter focuses on what exactly is done. Main features of model of input data i.e. the front end and the back end algorithm is explained. Some interesting approaches to extract data and applying algorithm are also discussed.
- ✓ The sixth chapter which is *Testing/Evaluation* states the testing of the preprocessing application and the developed tool. Testing is done by introducing different kinds of web log files to the system and then the behavior of the software and algorithm is evaluated. *Evaluation* is done by checking the accuracy of the output of the web usage mining tool. Different samples of web log files having different formats will be introduced to the system. Their output is compared with each other after applying clustering algorithm.

- ✓ The last chapter, Conclusion - Future Work, describes the challenges faced and the improvements which can be made in the current system. System constraints are also discussed in this chapter.

## **1.7 Summary**

The introduction chapter gave description of the domain of the problem investigation of the project and stated the main objectives of the project. A full structure of this report and the list of the contents and their brief detail are also given in this chapter.

## ***Chapter 2***

### ***Background***

#### **2.1 Introduction**

This chapter relates to development methodologies which will be used creating the report and the tool. The chapter also describes the full working of the system the techniques used to make the system successful. The chapter provides the full understanding of the project and fundamental concepts related to the system. Formal description of components of the system and software methodology adopted is explained in detail.

## 2.2 Web Site Maintenance

The companies, in order to keep most of their users intact, prefer to modify their websites rather than creating a new one or pay other companies to develop one for them according to the requirements of a user. This can be done by analyzing the performance of a website or by looking at the user's interactions and transactions or time spent on it. [1]

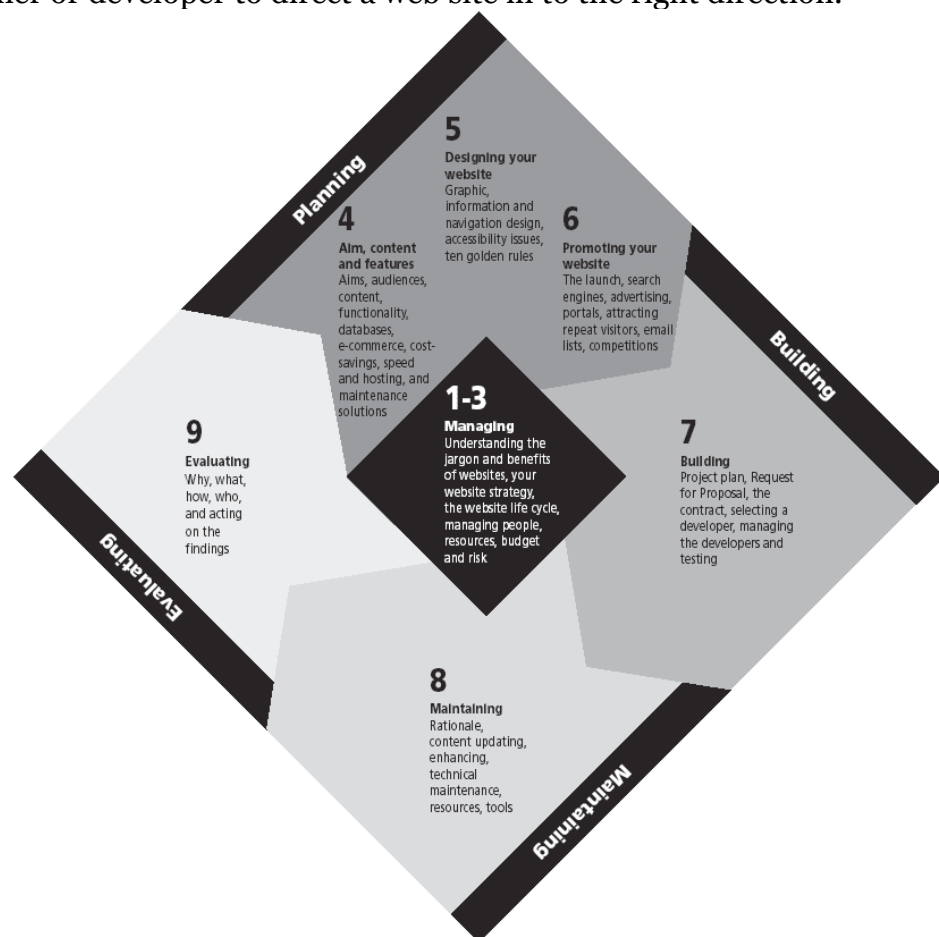
Web site maintenance can be defined as, modification after publishing a web site, in order to improve the performance and progress. The basic reasons for maintaining a web site is to provide continuity of service, which includes fixing of errors and bugs. Mandatory upgrades, for example if a new product is launched by a company it has to advertise on that web site. This will support user requests for improvements in functionality and performance. Web usage mining can be used and results generated can be analyzed to seek user requirements and their behavior on several web pages or specific areas on a web site. Website design is an important criterion for the success of a website. In order to improve website design, it is essential to understanding how the website is used through analyzing users' browsing behavior [18].

Marc Millon in his book "*Creative Content for the Web*" states that web is not and never will be a medium where content is carved in stone, left for years, months, even weeks unchanged [19].

Maintenance of a web site can be explained by help of the following issues:

- ✓ To keep visitors intact, it is necessary for the web site publisher to keep the web site up to date by adding new contents and highlighting those new changes. Adding contents just for the sake of adding them is not a good practice. Web sites with information like news articles and magazines should be updated nearly daily. Similarly monthly or weekly updates would be beneficial for a normal website.

- ✓ External links added in a website should be checked on daily basis. The basic reason for checking external links is that they might be of no relevance as those links may be updated or there might be a case that the external link no more exists.
- ✓ Finally progress of a website can be examined as well. Unlike most traditional print media, the web has in-built powerful mechanisms that enable the content provider to learn a great deal about the audience that visits your site or sites [19]. All the users visiting a web site leave some information about their visit each time they visit a web page. The information left by the user is IP of that user, the browser used for accessing that web page, time and date of access, areas visited and protocol used for communication etc. This information is enough for a publisher or developer to direct a web site in to the right direction.



**Fig 2.1 Web-site Life Cycle [20]**

## **2.3 Knowledge Discovery in Databases**

In terms of data analysis Knowledge discovery in data bases is the most appropriate concept. An abstract definition of KDD would be *“It is a process of identifying useful, novel, valid and understandable patterns of data. [13]”*.

In the context of knowledge discovery, the term process refers to some steps such as data preparation, data cleaning, refinement and evaluation of data etc. It is the job of a KDD system to extract patterns in such a way that they are easily understandable to humans.

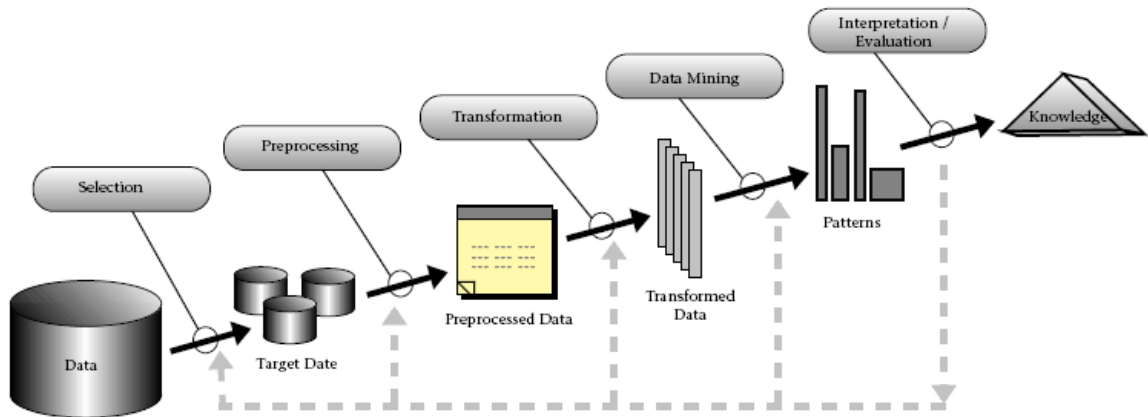
### **2.3.1 KDD and Data Mining**

Many people place data mining and Knowledge discovery in databases in the same category but according to Fayyad [13] KDD is a process which consists of particular data mining algorithms used for extracting of data from a set of data in raw form. According to these facts it can be stated that KDD uses data mining methods to extract knowledge according to some specifications and given thresholds on a database in which data is already inserted after pre-processing it. Data mining is a mandatory step in KDD while discovering knowledge. After defining the distinction between KDD and Data mining it can be stated that

*“Data mining is a step in the KDD process that consists of applying data analysis and discovery algorithms that, under acceptable computational efficiency limitations, produce a particular enumeration of patterns (or models) over the data [13].”*

### 2.3.2 Steps of KDD Process

Knowledge discovery in data bases comprises of an iterative sequence. This sequence involves many steps which are shown in the figure below:



**Fig 2.2 Data mining as a step in KDD [Fayyad 1996]**

KDD consists of an iterative sequence of the following steps [21]:

#### 1. Data Cleaning

The step is very important in terms of removal of noise and inconsistent data. Data to be filtered is always in a raw form. Databases have got very minimum support for data mining applications in the present era.

#### 2. Data Transformation

This is one of the most significant goals of data mining. Data transformation is applied in terms of dimensions to reduce or compress the size of data. This makes manipulation of data easier at later stages.

### **3. Data Selection**

Data selection is basically the process of defining a target data set. The procedure involves retrieval of interesting data from the data base. For correct data selection it is very vital that the information stored in the data base is in the correct format.

### **4. Data mining**

For the data mining process several intelligent methods are applied in order to extract patterns from data set. Different techniques can be used for data mining. Techniques like classification, regression, clustering and association are some examples of data mining.

### **5. Knowledge Presentation**

Presentation of knowledge using visuals and representation techniques are the methods to present mined knowledge to the user. This representation includes removal of redundant patterns and transformations and so on.

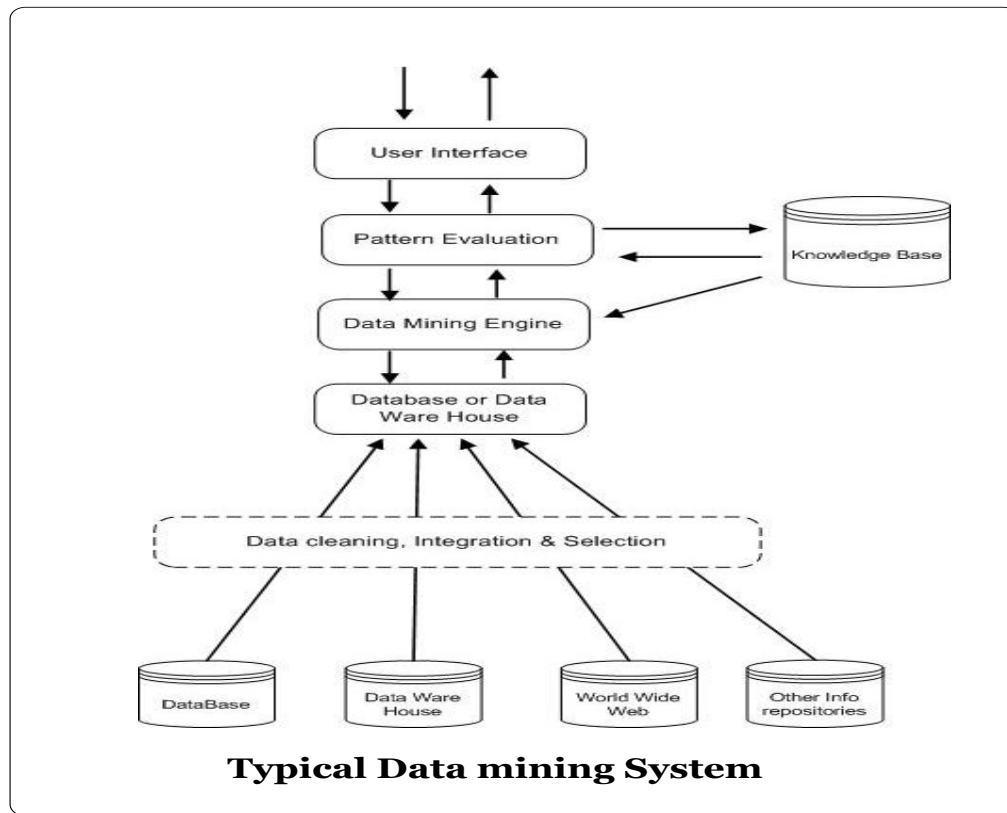
## **2.4 Data mining**

The dissertation is based upon a very important step during the process of Knowledge discovery in data bases which is Data Mining.

“Data mining involves the use of sophisticated data analysis tools to discover previously unknown, valid patterns and relationships in large data sets. [22]”

Due to huge amounts of transactions of data among organizations, firms and on the World Wide Web there is a need to turn that data useful. This useful information can then be used for applications like fraud detection, banking applications, e- commerce and software maintenance etc.

The figure below shows the structure of a typical data mining system.



**Fig 2.3 Architecture of Typical Data mining System [22]**

There are two kinds of data Mining Techniques [23]. One is directed data mining using top down approach and the other is undirected data mining which is a bottom up procedure. In this project UN directed data mining technique will be implemented. Prediction and description are two primary goals of data mining [13]. Prediction uses variables and attributes and fields of tables in the data base, these values are used to predict unknown and other variables of interest. Description emphasizes on patterns which identify data items. The concepts defined above can be achieved by using these data mining techniques defined below.

- 1. Classification:** A means to develop profiles of items with similar characteristics. This ability enhances the discovery of relationships that are otherwise not obvious.

- 2. Association Rules:** These are used to generate interesting relationships among large data sets.
- 3. Regression:** In this technique data item is mapped to a real valued prediction variable.
- 4. Sequence Analysis:** The method is basically used to model sequential patterns.
- 5. Clustering:** Clustering joins sets of data objects having similar properties in to one cluster and remaining data objects in another. There are many sets of clusters but one important distinction is between hierarchical and partitioned sets of clusters [24].

## 2.5 Data mining and World Wide Web

Application of data mining techniques to the World Wide Web, referred to as Web mining [26]. Web mining is basically extracting useful information across the World Wide Web and mining data for extracting usage patterns and browsing information.

There are many kinds of data that can be used in Web Mining such as Content, Structure, Usage and User Profile. [25]

- **Content:** It is real time data on the Web pages which are designed to convey this data to the potential users. The usual material is text, graphics and so on.
- **Structure:** The organization of the content. Arrangement of various HTML and XML tags is included within a given page. This can be represented as a tree structure, where the (html) tag becomes the root of the tree. Hyperlinks are a main part of structure as they connect one page to another.

- **User Profile:** Data that provides information about users of the Web site. This includes customer registration data and profile information. In the context of a log file the most crucial part is to identify a unique user profile, as the only information a log file can provide is the IP address; and a single user can have multiple IP's and many users may be using a single IP address.
  
- **Usage:** The pattern of usage of Web pages, such as IP addresses, page references, and the date and time of accesses [25]. This is the most important information included in a log file, by using this information time spent on a website and number of hits can easily be identified. For example Customers for e-commerce can be identified by analyzing regularities in a Web Log File. To develop applications for mining web log data it is very important to know that what kind of information can be extracted from these log files. Most of the times data in these files is not formatted. Data cleaning, data cleansing and data transformation may be applied on to the data to make it suitable for web mining applications. Web log data provides information about the class of users accessing which kind of data. Getting this type of information helps to categorize web pages or help in Web page Ranking.

### 2.5.1 Data Sources

- **Server Level Collection:** A Web server log is an important source for performing Web Usage Mining as the browsing behaviour of site visitors and other critical information is explicitly recorded in it. The unique feature of a web server is that it maintains information about user accesses on a web page.
  
- **Client Level Collection:** Client Level collection has an advantage over server level collection because it improves both the caching and session identification problem as well. Client Level data collection can be

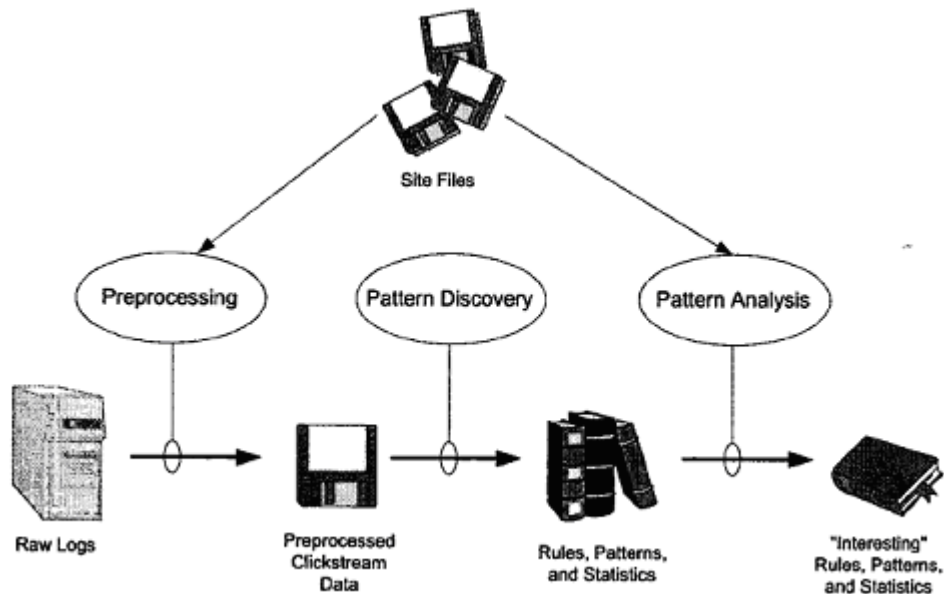
implemented by using remote agents or by modifying the source code of an existing browser to improve its data collection capabilities.

- **Proxy Level Collection:** Between client browsers and Web servers a Web Proxy acts as an intermediate level of caching. It can be used to reduce the load time of a Web page as well as the network traffic load.

The dissertation is based on applying a data mining technique after extracting useful information and storing that information in to database. The data source used will be web log files.

## 2.6 Enhancing Information

As shown in Figure 2.4, there are three main tasks for performing Web Usage Mining or Web Usage Analysis. This section presents an overview of the tasks for each step and discusses the challenges involved. [12]



**Fig 2.4 High Level Web usage Mining Process [12]**

### **2.6.1 Pre-processing**

In this step conversion of usage, content, and structure information is done for pattern discovery. Pre-processing comprises of steps like Usage Pre-processing, Content Pre-processing and Structure Pre-processing.

#### **2.6.1.1 Usage Pre-processing**

Usage pre-processing is by far the most difficult task in the Web Usage Mining process due to the incompleteness of the available data in the log files and other sources. Some of the experienced problems are as follows

- Single IP Address/ Multiple Server sessions
- Multiple IP Addresses/ Single Server sessions
- Multiple IP Addresses/ Single User
- Multiple Agent/ Single User

#### **2.6.1.2 Content Pre-Processing**

Steps like conversion of text, image, scripts, and other files such as multimedia into forms that are useful for the Web Usage Mining process are performed during content pre-processing. Often, these steps perform content mining such as classification or clustering. The content of each page view to be pre-processed must be assembled in the desired form first, either by an HTTP request from a crawler, or a combination of template, script, and database accesses [12].

#### **2.6.1.3 Structure Pre-processing**

The structure can be obtained and pre-processed in the same manner as the content of a site. The structure of a site is created by the hypertext links between page views Dynamic content cause more problems than static page views.

## 2.6.2 Pattern Discovery

Pattern discovery is based upon methods and algorithms related to fields such as statistics, data mining, machine learning and pattern recognition. Due to the difficulty in identifying unique sessions, additional knowledge is required. Some techniques which can be used for pattern discovery are Association Rules, Classification and Clustering.

### 2.6.2.1 Association Rules

In the context of Web Usage Mining, association rules (generate interesting relationships among large data sets) refer to sets of pages that are accessed together with a support value (the number of transactions that include all items in rule) exceeding some specified threshold.

It is basically a two step approach [29]

1. Generate all frequent item sets (sets of items whose support  $\geq$  <any condition>)
2. Generate high confidence association rules from each frequent item set.

The drawback of using association rules is that their generation requires a lot of cost. Apriori algorithm is the most famous algorithm for computing frequent item sets.

A rule must have some minimum user- specified confidence and support.

**Association rule:**  $X^{(s, c)} \Rightarrow Y$

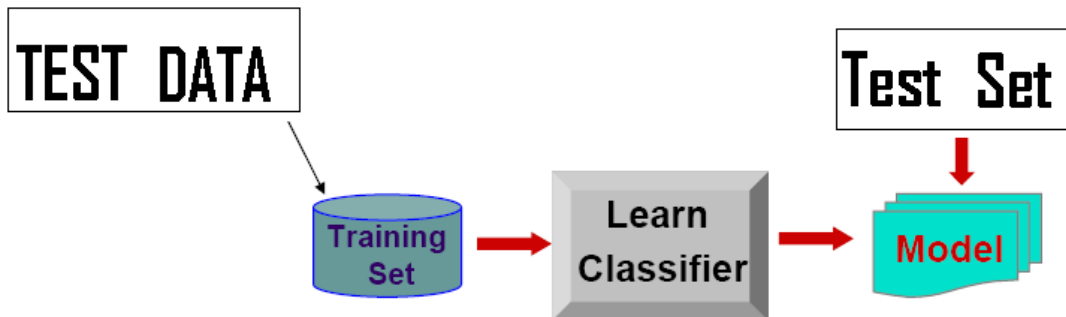
(Where  $s$  is the Support &  $c$  is the confidence)

For example in a medical store database there are 100,000 transactions of sales. Among these transactions 2000 include sales of both items A and B and 800 include item c. the rule for this kind of sale can be when A&B, C should be purchased on the same trip. Confidence:  $800/2000 = 40\%$  and confidence:  $800/100,000 = 0.8\%$

### 2.6.2.2 Classification

Develops profiles of items with similar characteristics. This ability enhances the discovery of relationships that are otherwise not obvious .In the Web domain, one is interested in developing a profile of users belonging to a particular class or category. This requires extraction and selection of features that best describe the properties of a given class or category. For example,

Classification on server logs may lead to the discovery of interesting rules such as: 30% of users who placed an online order in/Product/Music are in the 18-25 age groups [27].



**Fig 2.5: A classification model given test data [27]**

In the figure above a test data categorically defined is given to the training set which is then passed on to the learn classifier module and the model on the bases of test set describing the classified data which should be generated.

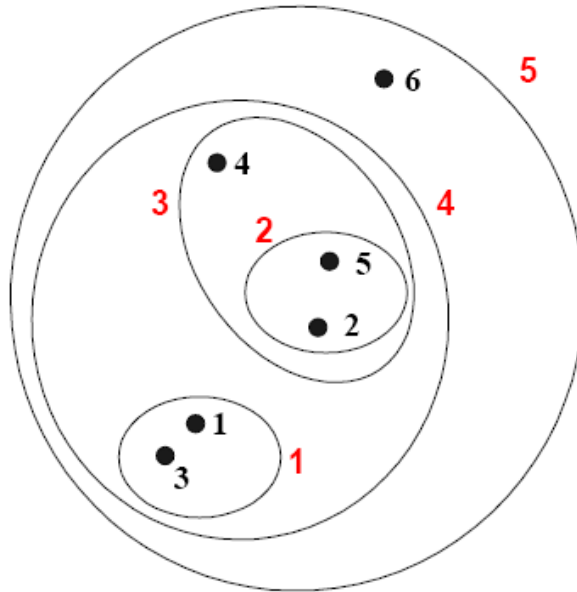
For Example: the test set is state the number of people which are married and have an income of above 50k. On the basis of test data given as an input to the system result will be generated on the basis of test set. There are several classification methods such as Bayesian Classification which uses probability model, second is decision trees which performs categorical classification another method is Neural networks, a biologically inspired model.

### 2.6.2.3 Clustering

Clustering is a technique to combine pages having similar characteristics. Usage clusters and page clusters are two important aspects of Web Usage Mining. Clustering of pages will discover groups of pages having related content. This information is useful for Internet search engines and Web assistance providers. There are many sets of clusters but one important distinction is between hierarchical and partitioned sets of clusters [25]. Partitioned and Hierarchical clustering is stated in detail in this chapter.

#### ➤ Hierarchical clustering

Hierarchical clustering builds a cluster hierarchy or, in other words, a tree of clusters, also known as a dendrogram [28]. Child cluster nodes are a part of each node. The advantage of using this technique is that data is available on different granularity levels. In Hierarchical clustering bottom up approach is used. Introducing linkage clusters into hierarchical clustering outputs proper shapes.



**Fig 2.6: Traditional Hierarchical clustering [29]**

Instead, hierarchical clustering frequently deals with the  $N \times N$  matrix of distances (dissimilarities) or similarities between training points [12]. This technique can be sub-optimal while processing huge sets of data, whereas data

compression can improve its performance. Hierarchical clustering involves techniques like Linkage Matrix, Hierarchical Clusters of Arbitrary Shapes and Binary Divisive Partitioning and so on. Hierarchical Clustering can be categorized in to different methods like Agglomerative, Divisive and so on.

#### ➤ **Non- Hierarchical or Partitioned Clustering**

Non- Hierarchical clustering when applied to a set of data generates classification rules by partitioning, in result groups are generated having no hierarchy between them. Non- Hierarchical clustering methods generally do not require more computational resources as compared to Hierarchical Clustering. Non- Hierarchical clustering is classified in to three main categories single pass, relocation and nearest neighbour. The most famous and common one among these classifications is the relocation clustering technique. Disjoint clusters are produced while using non-hierarchical clustering. These algorithms work best on distinct classes.

#### ➤ **K-means Algorithm**

This is a non-hierarchical clustering algorithm and preferably used for large data sets. This is an iterative algorithm based on Euclidian distance. The formula for finding the Euclidian distance is given by

$$d(\vec{X}, \vec{Y}) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_n - y_n)^2}$$

**Fig 2.7: Euclidean Distance**

Where x and y are two samples of data. Centres of all the data objects are returned to this distance function.

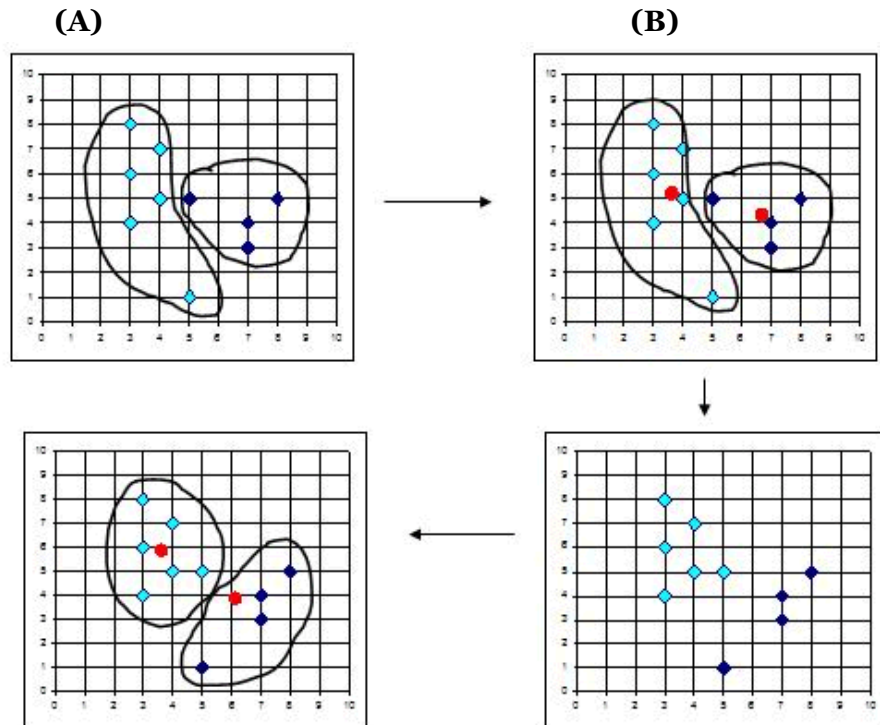
#### ➤ **Working**

The formulation of the algorithm is

- State the number of clusters to be generated, which is **k**.
- Choose k number of cluster centres. This should be done randomly.
- Using Euclidian distance, assign each instance to its cluster.
- Calculate mean of each cluster with new distances.
- Reassign all instances to the closet centres
- Repeat till the centre doesn't changes any more.

$$\sum_{i=1}^K \sum_{n=1}^N (\|x_n - \mu_i\|)^2 \quad \text{_____} (2.1)$$

In the above equation N is the number of data objects and K is the number of clusters which are given as an input to the system.  $x_n$  is the sample data and  $\mu_i$  is the centre. The process of applying k-means algorithm can be graphically shown as



**Fig 2.8: Working of K-means Algorithm [21]**

## 2.7 Data Transformation

Most of the times knowledge stored in to the data base are not suitable for the data mining algorithm to use, Therefore they have to be transformed in to Clustering suitable data. K-means algorithm requires data to be in numerical format so the first step should be to convert data in to numerical form. To achieve the above described format of data technique used will be Similarity Matrix. All the records in this matrix will be the distances between two samples or fields of data. All the distances in the figure below are in a range between 0 and 1. This distance can be generated by using the formula.

$$d(i,j) = \frac{\sum_{f=1}^n X_{i,j} Y_{i,j}}{\sum_{f=1}^n X_{i,j}} \quad (2.2)$$

$d(i,j)$  is the distance between two records  $i$  and  $j$ .  $n$  is the number of attributes of each record.  $X_{i,j}$  is a function obtaining just two values. Whereas  $Y_{i,j}$  is a function that depends on the type of each record. Complete description of these functions is given in the design chapter.

The figure below shows the distance values generated using formula 2.2.

$$S = \begin{vmatrix} S_{21} & & & & \\ S_{31} & S_{32} & & & \\ S_{41} & S_{42} & S_{43} & & \\ \vdots & \vdots & \vdots & \ddots & \\ S_{N1} & S_{N2} & S_{N3} & \dots & S_{N(N-1)} \end{vmatrix}$$

**Fig 2.9: Similarity Matrix**

$S_{i,j}$  is the similarity between two records of data in the data base. Similarity and dissimilarity depends on the values of distances. If a distance value is near 0 i.e.

0.5 or less it is said that two records are very similar where as values closer to 1 show dissimilarity between records.

## **2.8 Applications of Data Mining**

Applications of data mining would be

- ✓ Business Intelligence systems
- ✓ Financial institutions
- ✓ Fraud detection systems
- ✓ Software Maintenance
- ✓ E-commerce
- ✓ Telecommunications
- ✓ Banking and Financial applications

## **2.9 Summary**

The chapter analyzed the concepts which will be used throughout the dissertation. Data mining concepts along with KDD and Website maintenance were discussed in detail. Data mining algorithm chosen for the dissertation is K-means. The next chapter explains the requirements and success criteria on which the dissertation will be validated.

# ***Chapter 3***

## ***Requirements Specification***

### **3.1 Introduction**

The chapter is concerned with establishing the circumstance under which the work will be regarded as successful. The nature of this criterion depends on a web usage mining system which is being investigated. The overall requirement of this dissertation is to develop a web usage mining tool which should be able to generate results by applying data mining techniques on web log data. The stakeholders of the system include Users, analysts and system developers. The chapter also states the objectives of the project and the factors on which the project will later be evaluated.

## 3.2 Requirements

A software requirements definition is an abstract description of the services which the system should provide and the constraints under which the system must operate. System requirements may be either functional or non-functional requirements [8].

The system requirements are gathered mainly from the study of electronic web logs and examination of web applications and finally with the discussion with my project supervisor. The requirements illustrated in this section basically constitute of two categories Functional requirements and Non-Functional requirements. “Functional requirements capture the intended behavior of the system. This behavior may be expressed as services, tasks or functions the system is required to perform” [16] Where as Non-Functional requirements are “The qualities we desire of a problem solution other than those concerning its functionality, e.g. its robustness, its efficiency, its security, its extensibility, its maintainability, its portability, etc” [17] capture functional and non-functional requirements, it is essential to decompose the whole process in to different modules and design a specific sequence to implement them. These different modules help to analyze each and every task which will be performed for e.g. data cleaning, profile identification, session identification and transaction identification etc.

It is very important to state the stake holders and identification of actors acting upon the system. Each of the actors has different characteristics, which shall be modeled in to the system later on. Use cases, which formulate various functionalities of the system together and describe the workflow of each and every process in detail. Use cases play an important role in defining actors and the stake holders of the system. The basic stake holders of the system would be Users, Maintainers and Developers of the system.

### **3.2.1 Functional Requirements**

❖ **The most essential Functional requirements are:**

- **Data Cleaning**

The system will be capable of removing garbage data from a set of data, saving some amount of memory and avoiding unexpected results. This will automatically increase reliability and efficiency of the system as the work load will be reduced. Data cleaning will also help to reduce unwanted errors resulting in easy exception handling and less effort.

- **Methodology**

To apply data mining on a particular set of data the software should help a user, providing a step by step methodology to avoid mistakes and false results.

- **Reporting**

Detailed results will be reported after applying data mining algorithm on the set of data. On the basis of these results facts & graphs will be generated proving fruitful for a publisher or an administrator.

- **Log Record**

Each field in the table should contain information: IP Address, User ID, Time, Protocol, Status code, Size, and Agent etc. The information will be obtained by applying data mining technique on the given set of data. These values will then be used to assess a web domain.

#	IP Address	Userid	Time	Method/ URL/ Protocol	Status	Size	Referrer	Agent
1	123.456.78.9	-	[25/Apr/1998:03:04:41 -0500]	"GET A.html HTTP/1.0"	200	3290	-	Mozilla/3.04 (Win95, I)
2	123.456.78.9	-	[25/Apr/1998:03:05:34 -0500]	"GET B.html HTTP/1.0"	200	2050	A.html	Mozilla/3.04 (Win95, I)
3	123.456.78.9	-	[25/Apr/1998:03:05:39 -0500]	"GET L.html HTTP/1.0"	200	4130	-	Mozilla/3.04 (Win95, I)
4	123.456.78.9	-	[25/Apr/1998:03:06:02 -0500]	"GET F.html HTTP/1.0"	200	5096	B.html	Mozilla/3.04 (Win95, I)
5	123.456.78.9	-	[25/Apr/1998:03:06:58 -0500]	"GET A.html HTTP/1.0"	200	3290	-	Mozilla/3.01 (X11, I, IRIX6.2, IP22)
6	123.456.78.9	-	[25/Apr/1998:03:07:42 -0500]	"GET B.html HTTP/1.0"	200	2050	A.html	Mozilla/3.01 (X11, I, IRIX6.2, IP22)
7	123.456.78.9	-	[25/Apr/1998:03:07:55 -0500]	"GET R.html HTTP/1.0"	200	8140	L.html	Mozilla/3.04 (Win95, I)
8	123.456.78.9	-	[25/Apr/1998:03:09:50 -0500]	"GET C.html HTTP/1.0"	200	1820	A.html	Mozilla/3.01 (X11, I, IRIX6.2, IP22)
9	123.456.78.9	-	[25/Apr/1998:03:10:02 -0500]	"GET O.html HTTP/1.0"	200	2270	F.html	Mozilla/3.04 (Win95, I)
10	123.456.78.9	-	[25/Apr/1998:03:10:45 -0500]	"GET J.html HTTP/1.0"	200	9430	C.html	Mozilla/3.01 (X11, I, IRIX6.2, IP22)
11	123.456.78.9	-	[25/Apr/1998:03:12:23 -0500]	"GET G.html HTTP/1.0"	200	7220	B.html	Mozilla/3.04 (Win95, I)
12	209.456.78.2	-	[25/Apr/1998:05:05:22 -0500]	"GET A.html HTTP/1.0"	200	3290	-	Mozilla/3.04 (Win95, I)
13	209.456.78.3	-	[25/Apr/1998:05:06:03 -0500]	"GET D.html HTTP/1.0"	200	1680	A.html	Mozilla/3.04 (Win95, I)

**Fig 3.1: Sample table containing information gathered from a  
Server Log File [2]**

❖ **The desirable requirements are:**

- **Specific Data Set**

System will be capable of providing only required information by a user, based on the request of that user. For example, the system should only display the total number of visits by a user on a specific area or the dates on which a user visited that web site etc. depending on the request. Similarly, graph generation will be performed on the basis of specific needs of the user using the system.

- **Log History**

If time permits, history of results and some log information will be saved. This information will help to compare new results with the older ones, depending on the performance of that domain or a particular domain area.

### **3.2.2 Non-Functional Requirements**

- **Usability:**

The system should allow users to install it or use it with little or no training. User friendly interfaces will provide the user a very easy to learn and friendly environment inviting more and more users towards itself.

- **Scalability:**

The system should be designed in such a way that at later stages, changes in the components of the system would be easy requiring less effort and producing more output. The modularity of the system should be on such level that it can accommodate future changes or enhancements needed without redesigning its components.

- **Maintenance:**

Maintenance in the context of a software system is basically the preventive measures to prevent failure of software functions. Object oriented paradigm is being used, which helps in adding or removing modules from program without affecting other program areas. Attributes like configuration management, reverse engineering and re-engineering etc. are to be looked upon during the development phase, as they help a lot in maintenance of such systems.

### **3.3 Success Criteria**

In the rapid growth of web and electronic commerce, Web has been a key driving force. Improvement of Web communication is essential for a better satisfaction for the objectives of both, the Web site and its users. In this context Web usage mining, a relatively new research area has gained more attention. The objective of this project is the prediction of the user's behavior within the site, comparison between expected and actual Web site usage, and adjustment and alteration of a Web site with respect to the needs of the users. Another objective of this project is the development of a web usage mining system, capable of extracting navigation behavior patterns that allow analysis of a user's

interactions with the web based environments. Another aim of the project is to analyze the principles/facts gathered by the system and apply them to the required areas. This will not only increase the efficiency but positively affect the satisfaction level of the target audience. The final deliverables will be a web usage mining tool and a dissertation report broadly analyzing the above targets.

The system to be developed is targeting to the users general practices and actions on a website i.e. group or a specific set of users. The data collection and data cleaning will be based on classification scheme. The data collected and cleaned by using this scheme will be further used for analysis and performance of a particular area on the website and will help to improve or introduce any changes if needed. By applying the data mining algorithm or technique the fields or data which will be looked upon are:

- Client's IP Address
- User ID
- Path of resource on the web server
- Status Code
- Number of Bytes Transmitted
- Access Time

### **3.3.1 Objectives**

The objectives building towards the aims of the project are stated below:

1. Technologies that will be used for the implementation of the system are C# on a .Net Platform and the back end of the system i.e. the database will be based on SQL server.
2. Another objective of the system would be testing and evaluation of the system, this will revolve around the specifications and validation against conformance to the requirements. Moreover, the evaluation from feed

backs from test should greatly help to detect missed targets from the early stage.

3. The system will deal with crucial log file data, thus avoiding mistakes when accessing data is very important for making predictions and then applying them. Hence, risk analysis and scheduling should be followed during the project lifetime. For this purpose a full project plan is attached in the appendix which will be consulted and updated as any new changes are introduced to the project.

### **3.4 Summary**

The chapter analyzed the main software requirements. The requirements were categorized in to essential and desirable requirements. Categorization was done after analyzing the deliverables of the project. The requirements defined will work as a frame work for the later phases of the report like design and implementation phases. Stake holders of the system were also identified in this chapter. Finally, the success criteria and the primary and secondary aims were discussed in detail.

# ***Chapter 4***

## ***Design***

### **4.1 Introduction**

This chapter relates to various design issues concerning the development of the data mining tool. The system design presented in this chapter follows the specifications stated while collecting the requirements in the requirement phase. These requirements are then used to produce the UML (Unified Modeling Language) diagrams. The design phase lays the foundation for the next part which is the implementation phase. Each part of the design phase is discussed in detail and graphically represented to show the full and detailed working of the whole system.

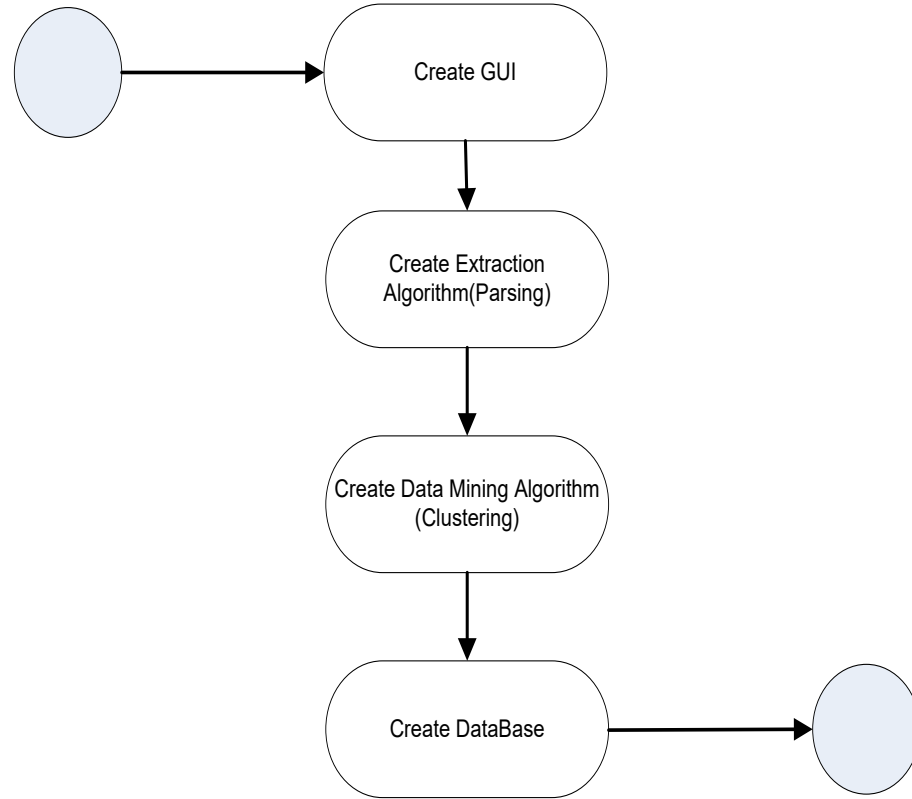
## 4.2 Steps Involved

The design chapter of the project consists of the following steps:

- *The Graphical User Interface:*  
GUI allows the user to interact with the tool and help a user to use the functionality of the system.
- *The Definition and Flow of Input data through the design of the model:*  
This is the most important part because the design schema of the database is defined and implemented according to the specifications of Actor, System, Use Case class and some UML diagrams. The correctness of these UML diagrams is very important as the system's implementation basis on them.
- *The design of the underlined Database schema:*  
To design the database schema, which stores the retrieved information from the given log files is the most vital part because a flexible and good design of the database schema depends on the specifications of use case and system diagram. For analyzing the underlined relational database, a number of tables are used which provide detailed specification of the tables used and their respective fields.
- *The design of the Pre-Processing application:*  
The preprocessing application comprises of two main parts: The graphical user interface with which the user interacts with and the algorithm that extracts data from log files provided and stores extracted information in to the database. The probability of extracting and storing the data in to the database successfully depends upon the design model of input data (UML diagrams) and the database schema.

### 4.2.1 Programmer's Activity Diagram

The figure below demonstrates the internal working of the system.



**Fig 4.1: Programmer's Activity Diagram**

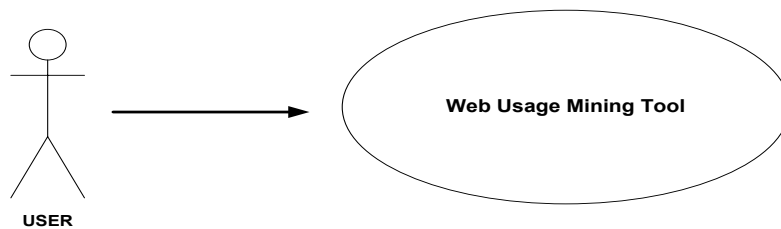
According to the above knowledge the system consists of mainly two parts: The pre-processing application (GUI, Parsing, and DM Algorithm) and the underlined relational database, the integrity and efficiency of the developed system heavily rely on their smooth interaction and cooperation. The front end of the application is simple and easy for the user to understand. Important phases of developing this tool are the extraction algorithm and the data mining algorithm. Creating the underlined database is also an important issue because information is saved in the data base and then returned back to the front end which is the GUI.

## 4.3 Basic System Design

The system is designed by keeping four aspects of the system in mind: The initialization of the system, the database operations, data extraction operations and Data mining Algorithm.

### 4.3.1 Actor Diagram

The only initiator of the system is the user itself which initiates and uses the system. In context of UML user is the primary actor of the system.

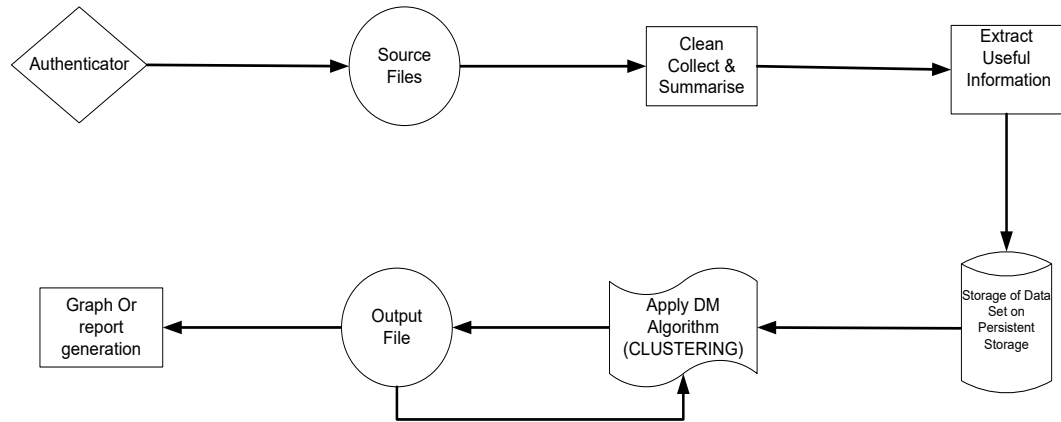


**Figure 4.2: Initial operation Actor Diagram**

The client application component of the system is relatively simple. User defines a format of a web log file and introduces that file to the web usage mining system which uses the built in filtering rules and stores the information in the database. The user also initiates the report generation component of the system producing results in the form of tables, graphs and flow charts etc.

### 4.3.2 System Flow Diagram

The system flow diagram shows all the components of the system which join and cooperate with each other making a full working system.



**Fig 4.3: System flow diagram**

#### 4.3.2.1 Initiator & Source files

These two components are initiated by the primary actor (User) of the system. The user initiates the system by defining a format for his log file and then source file (web log file) is provided to the system on which different operations are performed. Defining format is a very important task as extraction is carried out on the basis of this format. Apart from these two modules graph or report generation component is also initiated by the user.

#### 4.3.2.2 Clean Collect & Summaries

This module is a built in module which validates whether the log file is in the required format to apply filtering operations on or not. The formation of data in a log file is a very important aspect as data is extracted

according to this format. The format followed by the system is “IP Address, User Id, Time/Date, Method/URL/Protocol, Status, Size, refer, Agent”. This format should be strictly followed to ensure correct extraction of data from the desired log file.

#### **4.3.2.3 Extract Useful Information**

This component can also be termed as the filtering component because it reads the file, analyze it and remove garbage values like NULL values and extra spaces etc. Unwanted data from the user is also termed GARBAGE in this report. The module complies of some filtering rules and methods to extract data, the log file is rejected and error is displayed to the user on input of a log file having incorrect format.

#### **4.3.2.4 Storage of data on persistent storage**

This component relates to the underlined data base or the tables in which the information extracted is to be stored. It is very important for the “Extract useful Information” component to store correct information to the right tables. Values from the tables in the database are then used to apply mining techniques. Other tables included in the database are Format and Clustering. User specified tables are stored in Format table and clustering information is saved in Clustering table.

#### **4.3.2.5 Apply DM Algorithm**

This is the main module of the system as the data mining technique (i.e. K-means Clustering) is applied on the dataset stored in database. The database system will contain and manipulate all the data the client application identified and extracted while parsing a web log file. By using the stored operations clusters are generated and results are shown.

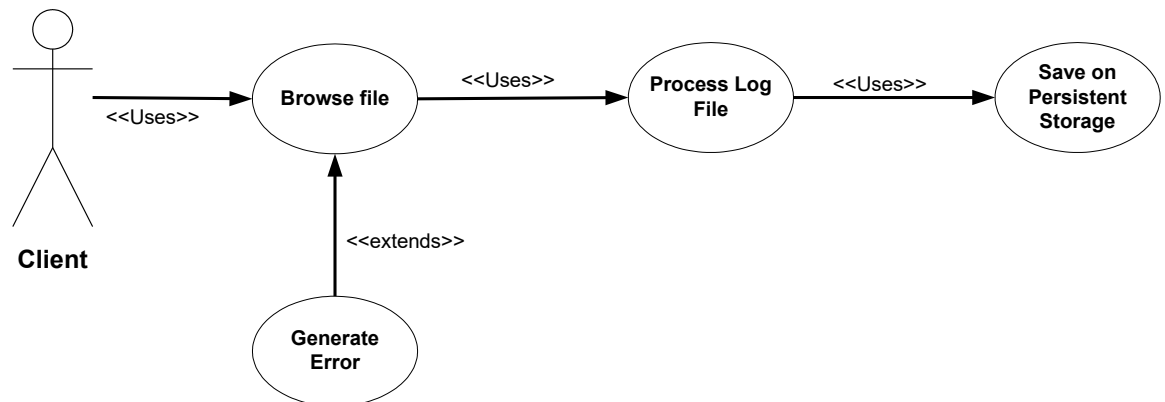
#### 4.3.2.6 Output File

The output files module basically corresponds to the resultant values generated after applying the mining technique. The component is connected back to the DM algorithm module as the user may check the variation of values on a specific dataset. The desired values are then used to generate graphs and tables.

### 4.3.3 System Use Cases

#### 4.3.3.1 The Initialization Operation

The initialization operation of the system is initiated by the user and comprises of three use cases in it as shown in the diagram below.



**Fig 4.4: Use case Diagram of Initialization operation of the system**

✓ **Browse File Use Case:** The user introduces the file to the system by using the browse file use case. The use case extends a “Generate Error” use case which identifies the occurrence of an error. Error is generated in case when format of the file does not match the required format by the system. The browse file use case ensures that user must enter a right file path and extension (i.e. ends with .log extension) to the system.

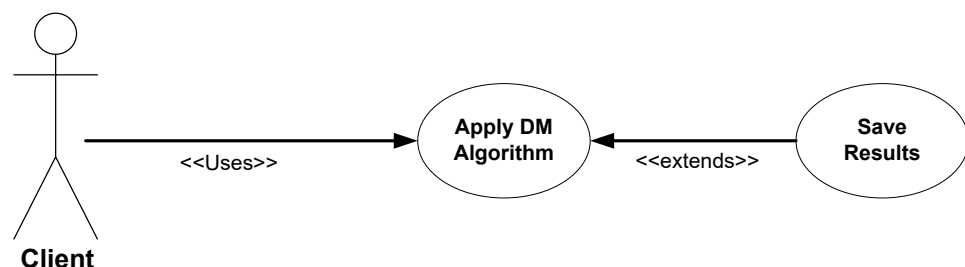
✓ **Process Log File Use Case:** The use case is related to the operations used in order to filter a log file. The filtering of data relates to operations like “deleting extra spaces”, “deleting garbage values” and “selecting appropriate data from file”. The appropriate data will be saved in lists or variables. The main reason of saving data on variables first and then onto database is to avoid frequent read/write operations.

✓ **Save on persistent Storage Use Case:** The use case gets data from the variables, data sets and stores it in the database.

#### 4.3.3.2 The Data mining Algorithm

In the background chapter various data mining techniques were explored. After a thorough and in depth investigation the most appropriate technique chosen was Clustering. There are various clustering Algorithms, and the most suitable one is K-means Clustering Algorithm. The main reason for choosing this algorithm is that it generates optimal results when appropriate; secondly, it assumes that all the attributes are independent and normally distributed. Results generated are accurate most of the times.

After opening the log file, the source data has to be processed. The model that is chosen for it is the top down one. The most important part i.e. the user id and the IP address is processed. As the processing continues more detailed information (i.e. time, method, URL, status, size, refer & agent etc.) is accumulated

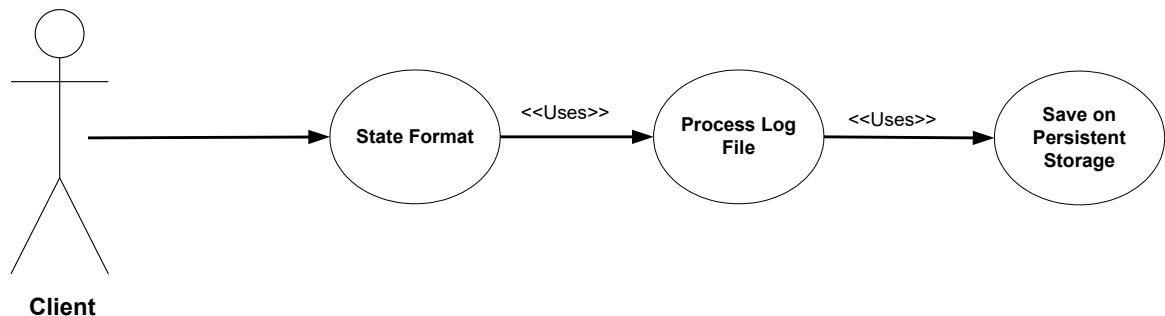


**Fig 4.5: Use case diagram of Applying Algorithm on Data**

The client uses the “Apply DM Algorithm” use case. “Save Results” use case is an extends to “Apply DM Algorithm”. The “Save Results” use case is an extend to the first use case because it may be used or not. This can be satisfied by the fact that the user might save the results or apply the DM algorithm again and then save it. Other use cases involved are “View Extracted Information”, in which the values saved on the persistent storage are displayed to the client. ” Generate Graphs and Reports” use case provides the graphical aid to analyze the results.

#### 4.3.3.3 State Format

This is a very vital part of the extraction process. By using this use case, user defines his/her own data format for the file to process. The state format module provides user with a set of attributes and the required set of delimiters to input and devise a format for the whole log file. The format defined is applied on each line of the web log file.



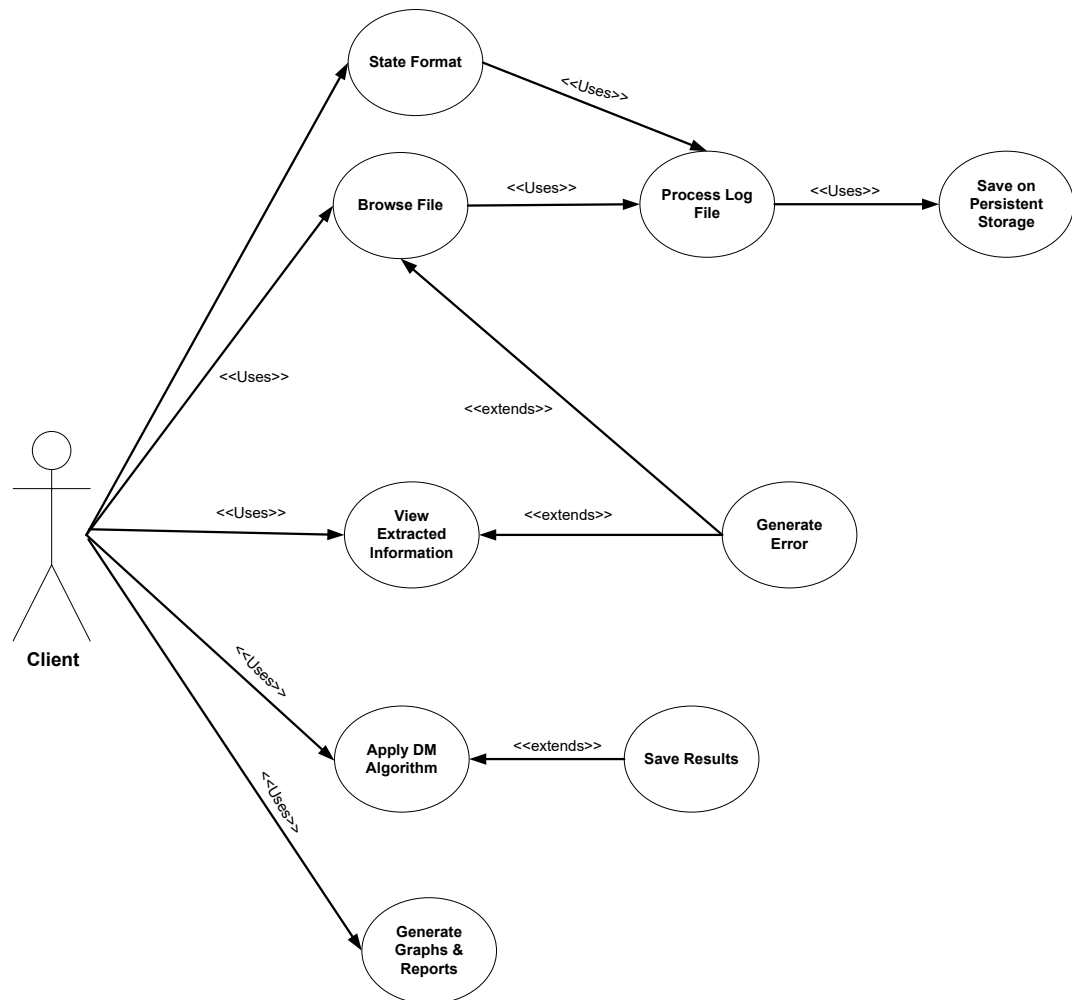
**Fig 4.6: Use case diagram of Applying Algorithm on Data**

An important feature of a web log file is that it has separate information for each transaction at each of its line. For example the information about an IP say “192.168.0.1” is at the first line where as the information or the access pattern of another IP say “192.168.0.2” is at another line. This feature of a web log file helps a programmer to devise an algorithm. Another advantage for a programmer while parsing a web log file is that the format is the same for all the line in a log file. One definition of a correct format for the whole file is enough. All the use cases discussed above are integrated in a way that

they can interact with the user and each other easily. A UML use case diagram is one of the most important modules in designing phase.

#### 4.3.3.4 Summary of the Client Application functions

A complete summary of the client related operations can be seen in the use case diagram shown below:

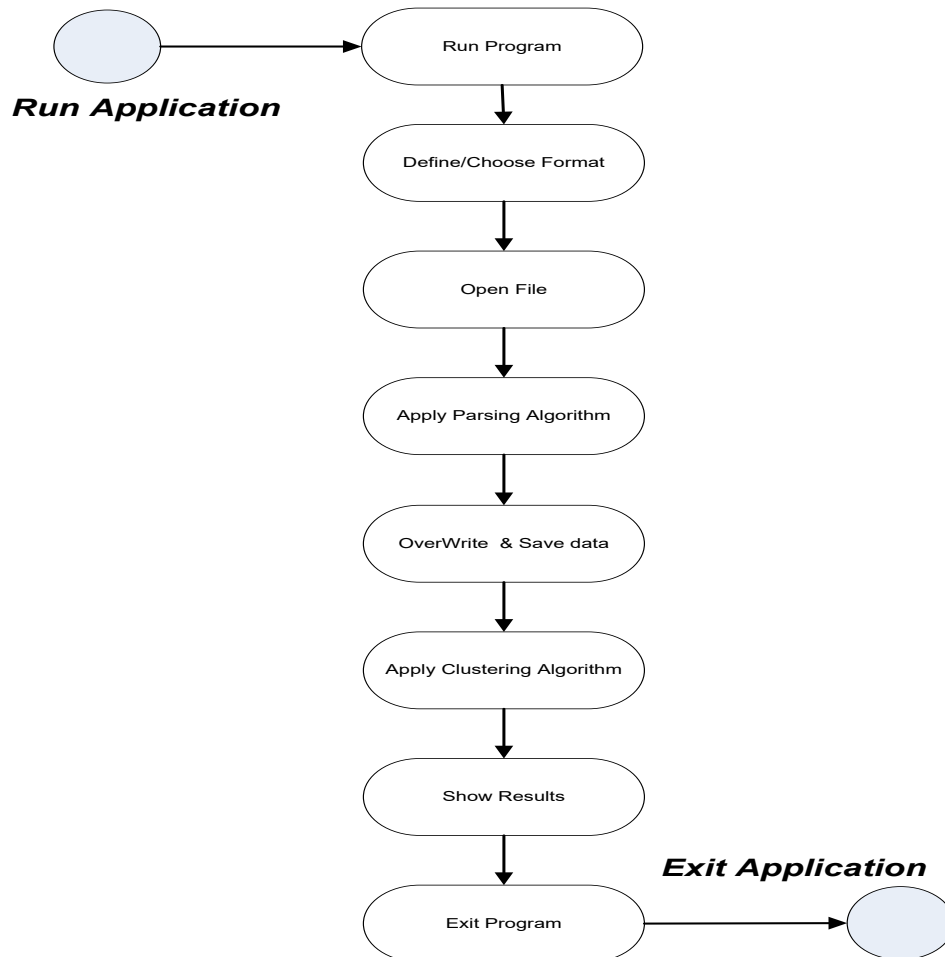


**Fig 4.7: Summary of the Client Application Functions**

#### 4.3.4 Design of the user-program interaction

The interaction between the user and the mining tool or the software is very vital. The user should be guided from applications messages and the feed backs should be received. Suitable error messages should also be displayed to the user in case of misuse. All of the above discussed messages help to state the status of the program at that very moment.

The figure below shows the user - system interaction. The steps to follow for the user to extract data and apply mining algorithm is shown in the figure below.



**Fig 4.8: Defining User's Own Format**

After running the program the user chooses to define a new format or a previously saved format. In case of defining a new format the user should know the exact pattern of the web log file being introduced to the system.

The figure below shows some *Web Log Data* on the left hand side this data is the actual data present in that file. Whereas on the right hand side format defined by the user is shown. The important thing here to keep in mind is that all the separators/Delimiters among the attributes should be clearly defined because extraction process requires absolute precision.

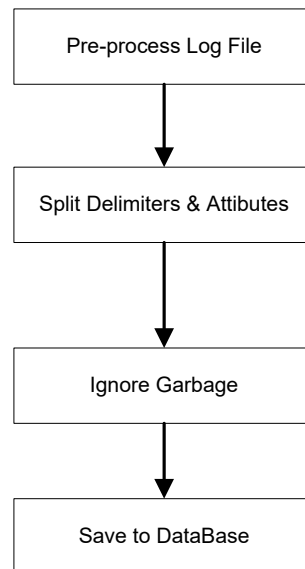
Log Data	Format
165.139.87.3	IP
--	--
[28/Feb/2007:22:29:39 +0500]	[DATE:TIME GARBAGE]
"GET	"METHOD
/~jba/images/SKB989.jpg	FILEPATH
HTTP/1.0"	PROTOCOL"
200	STATUSCODE
8614	PORT
"http://www.brain.net.pk/~jba/vollyballs1.html"	"URL"
"Mozilla/4.0	"BROWSER
(compatible; MSIE 6.0;	(GRABAGE;
Windows NT 5.1;	OS;
.NET CLR 1.1.4322)"	GARBAGE"

**Fig 4.9: Defining User's Own Format**

The new format is saved in to the database in the appropriate table “Format” if the user wants to save it. The user chooses the file of type “.log” to be processed and according to the defined format data extraction is done and values are saved in the table( LogData). Clustering is then applied on the extracted data and a set of clusters is generated. On the basis of that cluster information results are generated and shown to the user.

#### 4.3.5 Design of the Data Extraction Algorithm

The extraction Algorithm of the pre-processing application is a critical component of the system as data is extracted on the basis of this algorithm. The steps of the Data Extraction Algorithm are illustrated in the following figure.



**Fig 4.10: Pre-Processing Algorithm**

The diagram above shows that the algorithm is based on a top down model, which processes high level details then works progresses towards low- level details. The algorithm will process the data in a log file depending upon the format specified by the user. The second step of the algorithm separates

attributes and delimiters. This step is performed to get the data in the form suitable to save the underlined database. All the steps defined in the algorithm are performed line by line in a web log file. The next step is the removal of garbage from each file of the log file. Saving information in to the database is the last step of the extraction algorithm. Information is inserted in to the required fields by comparing the attributes names and the field names in the database. Inserting correct values in to correct fields is a very vital part as all clustering operations depend on the correct information. After extraction of data clustering algorithm is performed and data is retrieved from the related fields.

#### **4.3.6 Design of the Data Mining Algorithm (K-Means)**

Various data mining techniques were discussed in the background chapter. The mining algorithm chosen is K-means algorithm which is a type of Hierarchal Clustering Algorithm. The algorithm returns a set of clusters on the basis of which various results are generated. Information in the database should be prepared before applying clustering. The important thing to notice here is that K-means works on numerical data but for web log files it is different. To get numerical values the concept of Similarity matrix is applied on extracted data.

##### **✓ Generating Average Similarity**

Before applying K-means algorithm on to the newly generated distances there is a small step which helps to generate interesting and more accurate results. Distances generated by Similarity matrix are obtained by checking the similarity of each sample or row of data with every other row in that data set. After obtaining this information, Average similarity of each record with the whole data set can be obtained. For example: If we consider an IP say 192.168.0.1 as a row of database having different similarities with each field of the record. Average similarity of this sample data can be obtained by

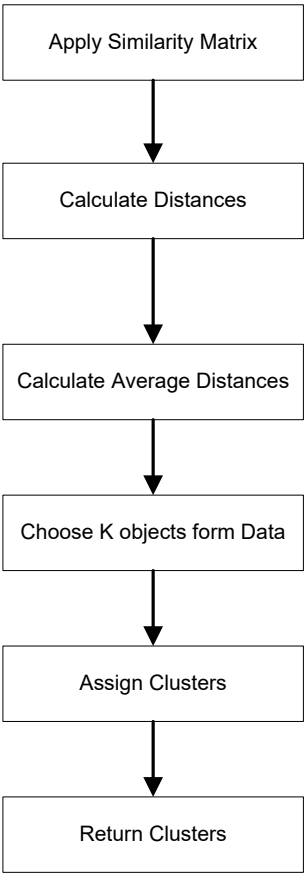
adding all the similarity distances of the field with the total number of fields related to it.

The formula for it could be

$$Avg_i = \frac{(\text{Sum of All the Distances})_i}{(\text{Number of Related Fields})_i} \dots\dots\dots(4.1)$$

Where i is a row having a set of attributes. K-means algorithm will be applied on to these average distances which will give results of most accessed pages by a user or an IP.

Working of the data mining algorithm is given below.



**Fig 4.11: Data Mining Algorithm**

After applying similarity matrix different distances are returned on the basis of which Average distances are measured .K are the number of clusters are

chosen and clustering is applied on data. Data objects are assigned until the mean clusters do not change.

#### 4.3.7 Functionality of the System

The structure of working of the User Application and relation of user components to the underlined database is presented in figure 4.12, which is also termed as the system class diagram. The front and back end of the system consists of different modules or classes which communicate with each other to perform the appropriate function. There are seven main classes which guide to a correct design by the programmer.

The figure below illustrates complete working of the system.

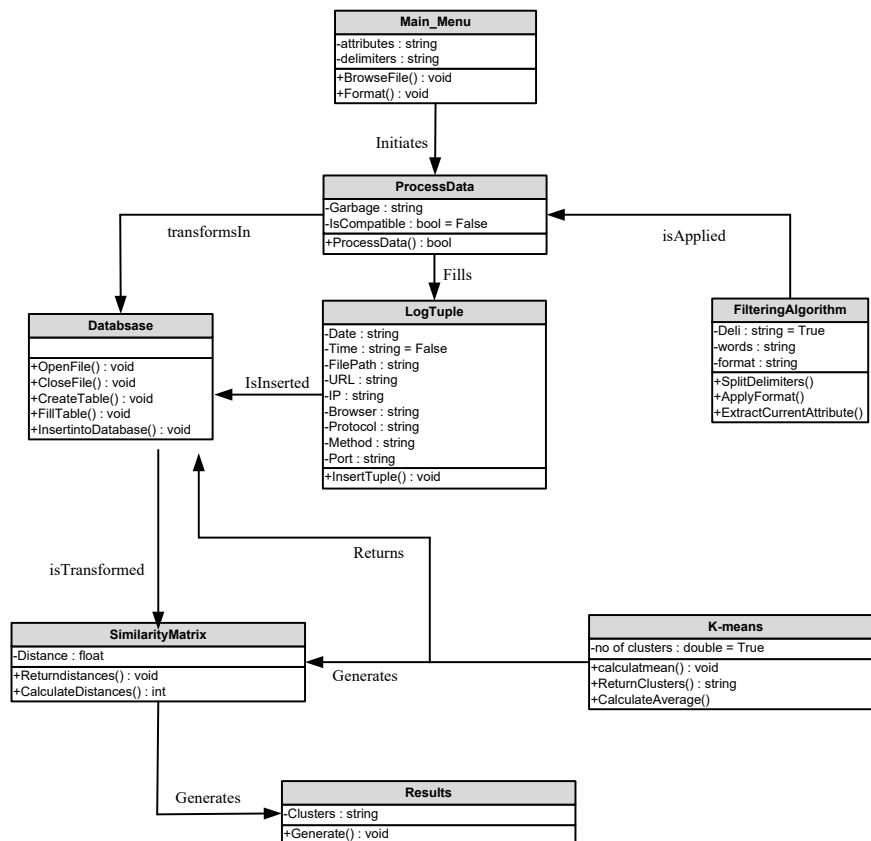


Figure 4.12 System Class Diagram

- **Main\_Menu:** This class helps the user to define a new format or choose a saved format; the new format is then saved in to the database. The class also creates the basic interface components which make it easier for a user to interact with the application.
  - i) **BrowseFile()**, the function enables the user to choose a file to be processed. The function allows the user to only select log files (.log). The chosen log file is then passed on to the class which filters data (Filtering Algorithm).
  - ii) **Format()**, In the format function a list of attributes and delimiters are defined by using these lists a user can define format. The function also provides the user with the facility of using previously saved formats.
- **Process Data:** The class basically checks the format of the file whether the file is according to the format defined by the user or not. The processdata() function reads the first line of the file and compares it with the format specified and returns a Boolean variable. Web log file is then passed on to the filtering class.
- **Extraction Algorithm:** It is one of the most significant class; it extracts the data from a file by separating delimiters and attributes. The class consists of the following functions.
  - i) **SplitDelimiters()**, this class maintains a list of the defined attributes and corresponding delimiters. Basically the operation of this function is to read a line from the file and traverse the whole line. While traversing a log files, attributes are extracted by keeping track of the information about the delimiter before and after each attribute.

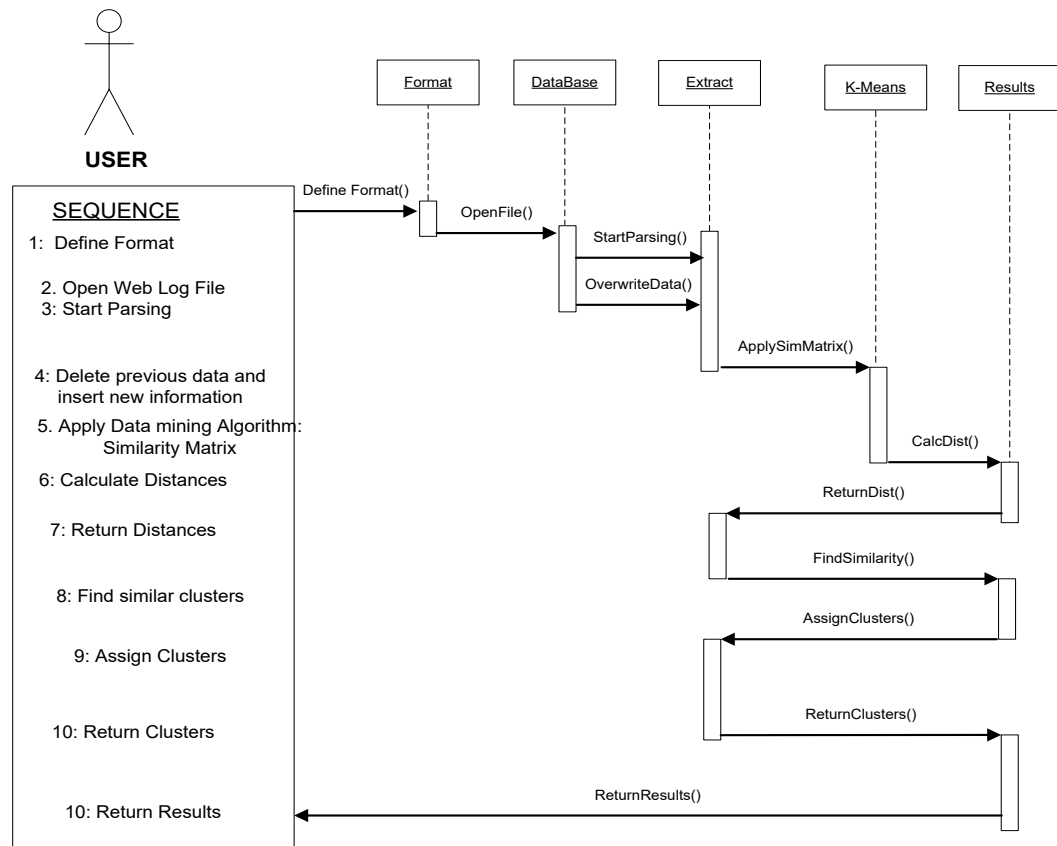
- ii) `Applyformat()`, this function basically sends the list of attributes to the `LogTuple` function in the class `LogTuple`. The `Logtuple` class then saves the information in to the underlined database i.e. in to `LogData Table`.
  - iii) `ExtractCurrentAttribute()`, the function is developed in support to the `ApplyFormat()` function. The function returns the current attribute on the pointer is currently pointing on. This function helps to insert correct information to its corresponding field in a table.
- **LogTuple:** The function is used to arrange the set of attributes to be inserted in to the database.
- i) `InsertTuple()`, the function uses all the attributes of the class, in each attribute, values of the log file attributes returned by the `ApplyFormat()` function are saved and data is transferred to the Database class in forms of tuples.
- **SimilarityMatrix:** The class is used to form a similarity matrix by using the information stored in the database. The class uses the following functions
- i) `CalculateDistances()`, the function calculates distances among two fields respectively and traverses whole of the database. Different ranges of clusters are generated by using this function. The distances generated are then passed on to the Cluster class.
  - ii) `ReturnDistances()`, the class collects all the distances generated by the `CalculateDistances()` function and returns it to the `MergeClusters()` function which resides in another class (Clustering).

- **K-Means:** The class aims to apply the data mining algorithm (Clustering) on the information extracted. Clusters are generated and returned from this class.  
The class has the following function
  - i) CalculateAvg(), the function calculates average distances among each record by using formula (4.1). These distances are then passed on to the Clustermean() Function.
  - ii) Clustermean(), The function receives a list of distances from the CalculateAvg() Function and groups data with same distances and forms a cluster for data with similar properties. This process goes on till the mean value of the cluster is changed no more.
  - iii) ReturnClusters(), the functions returns all the cluster information the DataBase class. The cluster information is then displayed using DUNDAS charts.
- **Results:** The class is used to display results to the user in visual form Dundas graphs are used to generate graphs. The class uses only one function to display results.
  - i) Generate (), the function corresponds to the stored data in the database. Dundas charts are used for this purpose. Dundas Chart for .NET provide SQL Server Reporting Services with advanced, fully managed, and highly extensible charting to help compliment the advanced features of Microsoft's new reporting tool [6].
- **DataBase:** The class receives loads of information to store from different classes. The class implements different data base operations like create connection, opening a connection, closing a function filling a table. The most important function in this class is InsertintoDatabase() function.
  - i) InsertintoDatabase(), The function receives information from LogTuple class. Main functionality of the function is to insert

appropriate attributes in to their corresponding fields. The function also inserts data received from similarityMatrix class and Clustering class.

## 4.4 System's sequence Diagram

The sequence diagram of the system is shown below. The class shows how the user controls the data mining tool. The figure also shows how Similarity Matrix and Clustering is applied to the stored information.



**Figure 4.13 User- System Sequence Diagram**

## **4.5 Summary**

In this chapter, two major procedures of the tool; parsing a web log file and clustering the extracted data were discussed. Different UML diagrams (System, Actor, Class, Use Case, Activity and sequence) are provided. The main components of the software were broken down and were analyzed in detail. The next part of the report is the implementation phase where the implementation issues of the software and system constraints are given.

# ***Chapter 5***

## ***Realisation***

### **5.1 Introduction**

The chapter details a full description of its implementation. The chapter describes the methodology adopted by the developer for the transition from design to the implementation phase. The chapter also explains the choice of tools and methods used within the system. Pseudo-code along with C# code is provided for some important functionalities and programs. Screen shots of the Data mining tool are also provided.

## 5.2 High Level Programming Specifications

In order to make the implementation of the code easy and efficient some specifications should be met. The platform or the programming language should be new, because a lot of languages tend to turn obsolete. Another issue before choosing a programming language is its documentation in literature and on internet. Concurrency is also an important factor because while parsing a web log file, large amounts of data is traversed. Finally, the language should have the ability to connect to the underlined database used to perform SQL queries.

The language selected for coding the data mining tool is C# on a .Net platform. The reason behind choosing C# is its text processing capability. Another important factor is that C# is completely platform (windows, UNIX, Linux) compatible. Having a language platform independent reduces the overhead for porting any other application which acts as a communicator between the source code and the Operating System.

Other advantages of using C# are [7],

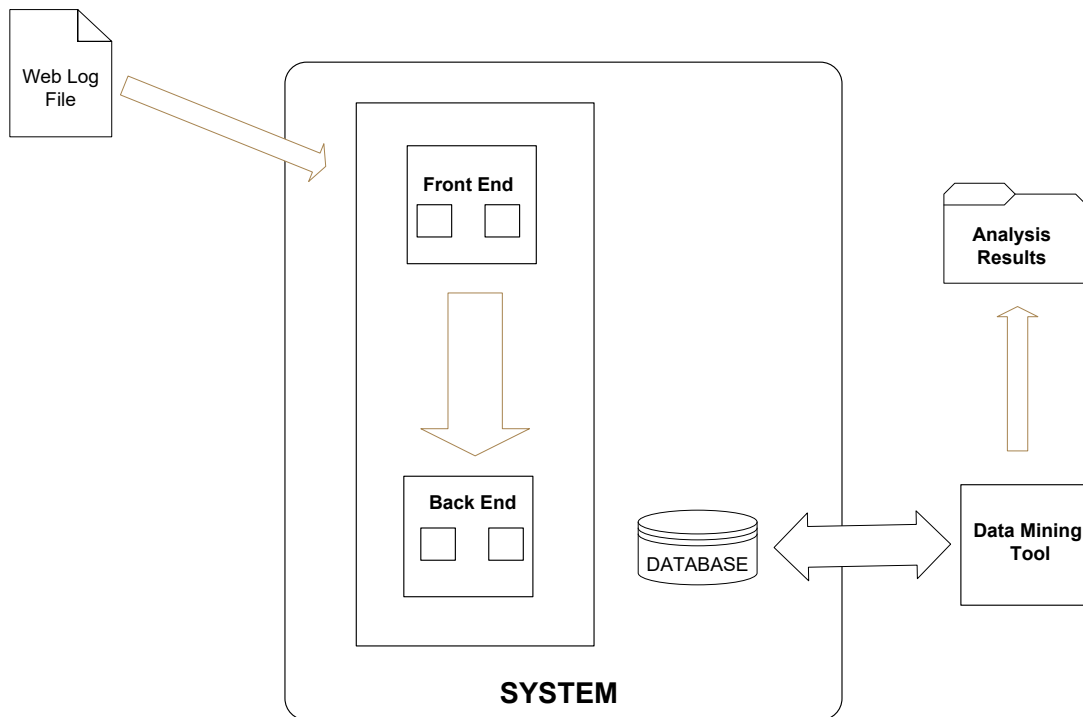
- ✓ .NET Framework (access to thousands of classes that you won't have re-create)
- ✓ Very friendly
- ✓ No buffer overflows
- ✓ Memory management and garbage collection.

Database connectivity is also an important factor. C# supports a number of relational and an object database system. The advantage of using such a language is that it provides the software with the necessary flexibility and maintainability if the underlined database needs some changes. Last but not the least C# receives a large amount of technical support throughout the world which helps a developer while writing the source code.

### 5.3 Implementation Phase and Tools

The previous chapter suggested that the web usage mining tool will include pre-processing Application which parses the Web log data form a web log file and a database management system where information is stored in a form capable of being clustered. The front end uses C# (Pre-Processing Application) and the DBMS (data base management system) I used in SQL server 2005. Advantage of using SQL server on the ask end is that it can scale up easily and can handle large amounts of data.

The figure below displays the whole structure of the system.



**Fig: 5.1 Structure of the System**

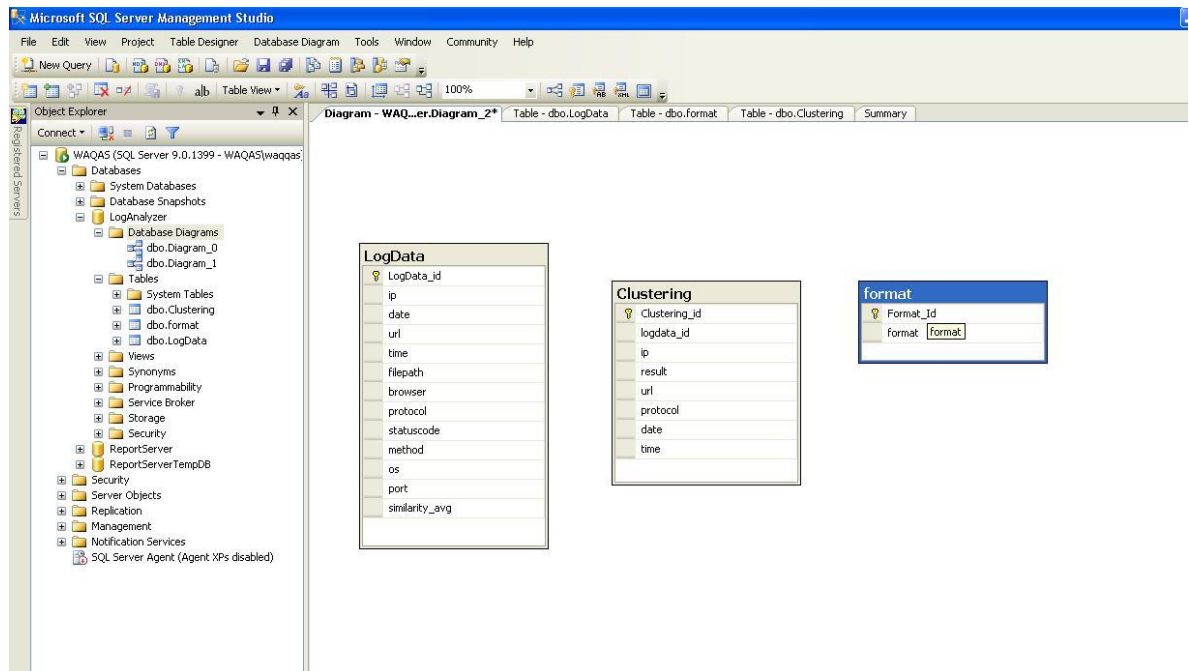
In order to accomplish the criteria defined in the designed chapter, Programming language should be chose which is C# on a Dot. Net platform. The parsing algorithm for data extraction should be defined and finally the data

mining algorithm should be devised, HAC Agglomerative algorithm which is a type of Hierarchical clustering is chosen for the tool. Similarly, the extraction and the data mining algorithm should be implemented in such a way that database operation like reading and writing should be easy and reliable.

## 5.4 Implementation of the Model of Input Data

The following picture shows the model of input data used by the preprocessing application. The database contains three tables “Format”, “LogData” and “Clustering”.

There is no relationship between the tables as a single table can handle all the inputs from the data mining tool.



**Fig: 5.2 LogAnalyzerDb Schema**

The most important table among all the tables is “LogData” which contains extraction information about all the attributes. Format defined by the user in the “Format” table. Clustering information is saved in to the table named “Clustering”.

- **Format:** The table saves and retrieves the format defined by the user.
  - Format\_Id: This is the Id of formats in that table, it is auto increment and self generated by the data base.
  - format: Strings of data are stored in this field. User can save the format stings in this field and the tool retrieves data from this field as well.

Table - dbo.format	Table - dbo.LogData	Diagram - WAQ...er.Diagram_1*	Table - dbo.format	Table - dbo.Clustering*	Table - dbo.LogData*	Summary	▼ X
	Format_Id	format					
▶	18	IP - - [DATE:TIME GARBAGE] "METHOD FILEPATH PROTOCOL" STATUSCODE PORT "URL" "BROWSER (OS)GARBAGE"					
	20						
*	NULL	NULL					

**Fig: 5.3 Database view of table Format**

- **Clustering:** The class contains the information to be clustered. Results and graphs are generated using the information in this table. Technique of similarity matrix is applied on the field’s ip, url, protocol, date and time.
  - Clustering\_id: This is the Id of Clusters in that table; it is auto increment and self generated by the data base.
  - Ip It contains two strings i.e. ip’s in it, similarity matrix calculates and returns two similar ip’s in this field.
  - Result: Result is the distance calculated by applying similarity matrix. The range of distances is between 0 -1.
  - url: It contains two strings i.e. url’s in it, similarity matrix calculates and returns two similar url’s in this field.

- Protocol: It contains two strings i.e. protocols in it, similarity matrix calculates and returns two similar protocols in this field.
- Date: It contains two strings i.e. Date in it, similarity matrix calculates and returns two similar Dates in this field.
- Time: It contains two strings i.e. Time in it, similarity matrix calculates and returns two similar Times in this field.

Table - dbo.Clustering		Summary					
	Clustering_id	ip	result	url	protocol	date	time
▶	378221	165.139.87.3 , ...	0.6	http://www.brain.net.pk/~jba/vollyballs1.html, http://...	HTTP/1.0, HTTP...	28/Feb/2007	22:29:39
	378222	165.139.87.3 , ...	0.6	http://www.brain.net.pk/~jba/vollyballs1.html, http://...	HTTP/1.0	28/Feb/2007	22:29:39, 22:29
	378223	165.139.87.3 , ...	0.8	http://www.brain.net.pk/~jba/vollyballs1.html, http://...	HTTP/1.0, HTTP...	28/Feb/2007	22:29:39, 22:29
	378224	165.139.87.3 , ...	0.8	http://www.brain.net.pk/~jba/vollyballs1.html, http://...	HTTP/1.0, HTTP...	28/Feb/2007	22:29:39, 22:29
	378225	165.139.87.3 , ...	0.8	http://www.brain.net.pk/~jba/vollyballs1.html, http://...	HTTP/1.0, HTTP...	28/Feb/2007	22:29:39, 22:29
	378226	165.139.87.3 , ...	0.8	http://www.brain.net.pk/~jba/vollyballs1.html, http://...	HTTP/1.0, HTTP...	28/Feb/2007	22:29:39, 22:29
	378227	165.139.87.3 , ...	0.8	http://www.brain.net.pk/~jba/vollyballs1.html, http://...	HTTP/1.0, HTTP...	28/Feb/2007	22:29:39, 22:29
	378228	165.139.87.3 , ...	0.8	http://www.brain.net.pk/~jba/vollyballs1.html, http://...	HTTP/1.0, HTTP...	28/Feb/2007	22:29:39, 22:29
	378229	165.139.87.3 , ...	0	http://www.brain.net.pk/~jba/vollyballs1.html	HTTP/1.0	28/Feb/2007	22:29:39

**Fig: 5.4 Database view of table Clustering**

- **LogData:** The values after applying the extraction algorithm are returned to this table. Data is then used to calculate distances and similarity matrix is also applied on the data in this field.
- LogData\_id: This is the Id of each field in that class; it is auto increment and self generated by the data base.
  - The table also contains average similarities calculated after extraction.
  - The field's ip, date, url, time, filepath, browser, protocol, port and status code contain all the extracted information. All fields in this table are of type string.

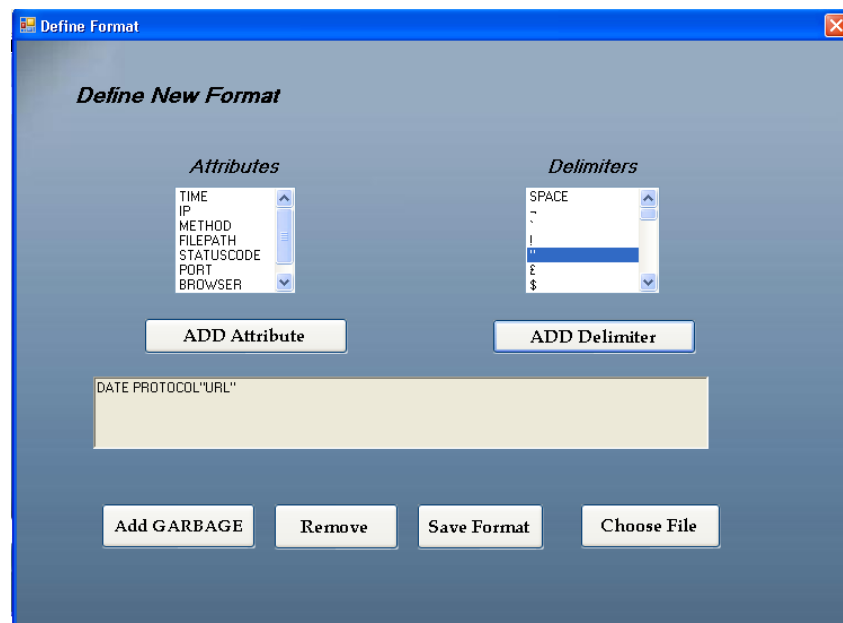
The figure below shows the LogData table.

Table - dbo.LogData	Diagram - WAQ...er.Diagram_1*	Table - dbo.format	Table - dbo.Clustering*	Table - dbo.LogData*	Summary			
id	ip	date	url	time	filepath	browser	protocol	statusCode
▶ 26304	165.139.87.3	28/Feb/2007	http://www.brai...	22:29:39	/~jba/images/SK...	Mozilla/4.0	HTTP/1.0	200
26305	202.125.143.68	28/Feb/2007	http://www.brai...	22:29:39	/newpage/links/...	Mozilla/4.0	HTTP/1.1	200
26306	193.251.135.123	28/Feb/2007	http://www.brai...	22:29:40	/~aup/IBMS.h1....	Mozilla/5.0	HTTP/1.0	200
26307	203.128.11.15	28/Feb/2007	http://brain.net....	22:29:40	/newpage/8.gif	Mozilla/4.0	HTTP/1.1	200
26308	202.125.143.68	28/Feb/2007	http://www.brai...	22:29:40	/newpage/googli...	Mozilla/4.0	HTTP/1.1	200
26309	202.125.143.68	28/Feb/2007	http://www.brai...	22:29:40	/newpage/links/...	Mozilla/4.0	HTTP/1.1	200
26310	202.125.143.68	28/Feb/2007	http://www.brai...	22:29:40	/newpage/mp3.gif	Mozilla/4.0	HTTP/1.1	200
26311	202.125.143.68	28/Feb/2007	http://www.brai...	22:29:40	/newpage/ll.gif	Mozilla/4.0	HTTP/1.1	200
26312	202.125.143.68	28/Feb/2007	http://www.brai...	22:29:40	/newpage/web.gif	Mozilla/4.0	HTTP/1.1	200
26313	165.139.87.3	28/Feb/2007	http://www.brai...	22:29:39	/~jba/images/ne...	Mozilla/4.0	HTTP/1.0	200
26314	202.125.143.67	28/Feb/2007	-	22:28:36	/%7Eaasif/2001...	DA 7.0"	HTTP/1.1	206
26315	165.139.87.3	28/Feb/2007	http://www.brai...	22:29:40	/~jba/images/SK...	Mozilla/4.0 (com...	HTTP/1.0	200
26316	203.128.29.182	28/Feb/2007	http://www.brai...	22:29:40	/webmail/src/rea...	Mozilla/4.0 (com...	HTTP/1.1	200
26317	203.128.11.15	28/Feb/2007	http://brain.net....	22:29:41	/newpage/9.gif	Mozilla/4.0 (com...	HTTP/1.1	200
26318	193.251.135.123	28/Feb/2007	http://www.brai...	22:29:41	/~aup/normal1.jpg	Mozilla/5.0 (Win...	HTTP/1.0	200
26319	203.128.11.15	28/Feb/2007	http://brain.net....	22:29:42	/newpage/sdfg.gif	Mozilla/4.0 (com...	HTTP/1.1	200
26320	203.128.11.15	28/Feb/2007	http://brain.net....	22:29:43	/newpage/12.gif	Mozilla/4.0 (com...	HTTP/1.1	200
26321	74.6.70.237	28/Feb/2007	-	22:29:33	/~penigma/_the...	Mozilla/5.0 (com...	HTTP/1.0	200

**Fig: 5.5 Database view of table LogData**

## 5.5 Main Interface

The “Define Format” interface can be categorized as the main interface because the user’s maximum interaction with the system is available. This windows form helps the user to define a new format on the basis of which data will be extracted by using the extraction algorithm.



**Fig: 5.6 The Define Format interface**

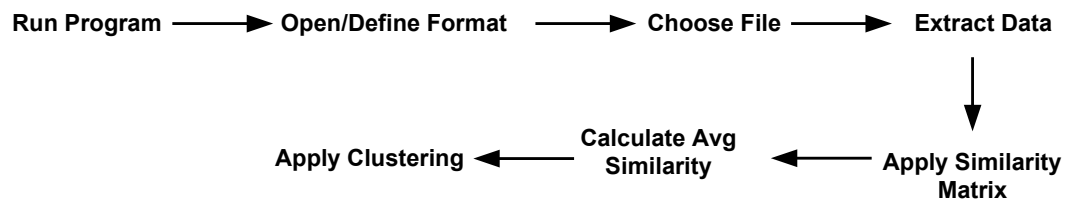
The form contains the following components:

- ✓ On the left top corner a list of attributes is provided. The attributes relate to the attributes used in a web log file. Attributes are added on to the text box below and deleted from the list itself as there are no duplicate attributes in format definition; once the “ADD Attribute” button is pressed.
- ✓ Similarly, on the right top corner a list of delimiters is provided. The delimiters relate to the separators used in a web log file. Attributes are separated by the delimiters so there can be many occurrences of a single delimiter in format definition.
- ✓ The “ADD Attribute” button adds the attribute text in the rich text box and at the same time removes that attribute from the attribute list. This operation restricts the user for duplicating an attribute.
- ✓ The “ADD Delimiter” button adds the delimiter text in the rich text box. There can be multiple occurrences of a delimiter in a file unlike attributes.
- ✓ The rich text box is used to display the format defined by the user. New text is appended and removed according to user commands.
- ✓ The “ADD GARBAGE” button allows the user to identify useless text in a format. Garbage is always ignored by the system and never extracted for processing.

- ✓ The “Remove” button removes the recent attribute or a delimiter inserted. If an attribute is removed it is added back to the attribute list.
- ✓ The “Save Format” button allows the user to save the defined format as it can be used again and again.
- ✓ The “choose file” button opens a new open file dialogue box. The open file dialogue box only allows the user to choose web log files.

## 5.6 The Pre-Processing Procedure

Within a web log file there are a lot of fields of interest such as Ip address of any user. Ip is the identification of every user. Apart from Ip address fields like URL and filepath are also important as URL is the navigation path of that IP and file path suggests the area in a website which is viewed by that particular Ip.



**Fig: 5.7: Structure of implementation phase**

When the user runs the program main menu appears providing the options for opening a saved format or defining a new format. Help and exit buttons are also displayed in that interface.

### 5.6.1 Open/Define Format

The functionality is useful for the users who analyze files of the same format all the time. List of saved formats previously defined by the user is displayed from which appropriate format is chosen by the user. Format is retrieved from the “Format” which resides in the database. The query to retrieve all the formats saved is written as:

**“query = "select format from format";”**

In the above query format is a field from where all the data is collected and the query ends with the name of the table which is format in this case.

To save a new format defined by the user the query used is given as follows:

**“query = "INSERT INTO format VALUES ('"+format+"");”**

The query uses basic SQL command to save a format defined by a user. The newly defined format is saved in to a variable named format and inserted in to the table.

### 5.6.2 Choose File

The choose file module lets the user to choose files with extension (.log) as all the log files will have this extension. Other than this there might be a case where all the data is saved in to a text file. Source code for choosing only log files is given as follows:

**“openFileDialog1.Filter="All files (\*.\*)|\*.\*|Log files (\*.log)|\*.log";”**

- **File Reading:** the pattern of file reading was designed a way such that a file should be read till end of file and for each line it should be checked that a line should not start with a special character.

Code for file reading is given below:

```
if (openFileDialog1.ShowDialog() == DialogResult.OK)
{
    if ((myStream = openFileDialog1.OpenFile()) != null)
    {
        Console.WriteLine("File opened");

        using (StreamReader sr = new StreamReader(myStream))
        {
            String line;

            while ((line = sr.ReadLine()) != null)
            {
                if (line.StartsWith("#") || line.StartsWith("%") ||
                    line.StartsWith("?") || line.StartsWith("*") ||
                    line.StartsWith("@") || line.StartsWith("!"))
                    continue;

                line = line + "EOF";
                words.Add(line);
                tuples.Add(new LogTuple());
            }
            SplitFormat(words, format);
            InsertIntoDatabase();
        }
        myStream.Close();
    }
}
```

StreamReader is a built in class used to read the lines in a file. The code starts reading a file line by line till EOF (end of file). Inside the while loop the lines starting with a special character are ignored. While reading a file each line is passed to a tuple class which organizes the data in a form to be inserted into the database. “Words” is a string array passed to the SplitFormat function in which attributes and delimiters are separated. InsertIntoDatabase is a function used to inserts attributes it to the underlined database.

### 5.6.3 Extraction of Data

The extraction operation is initiated by the system itself after a file is chosen by the user. The algorithm uses three functions named as ApplyFormat(), extract(), SplitFormat(), ExtractCurrentAttr(), InsertTuple(), Insertintodatabase().

Flow of extraction algorithm is as follows:

*Extract()* → *SplitFormat()* → *ExtractCurrentAttr()* → *Insertintodatabse()*  
→ *InsertTuple()*

The extract() function opens a file and reads line by line data and passes each line to the split format function. The splitformat() function receives a list words having a line and format which is defined by the user.

The pseudo code of the function is given as follows:

#### **Variables**

*startdelimiter* = “ “;

*enddelimiter*=” ”;

*format*+=”EOF”;                      // Inserts EOF (end of file) at the end of  
*format*

*currentattribute*= “ “;

#### **Functionality**

While length of format is not equal to zero

{

  If (format starts with “GARBAGE”)

  {      *currentattribute*=”GARBAGE”;

```

        remove( currentattribute from format);
    }

    For each (string in attribute)

    {

        If (format starts with attribute)

        {
            currentattribute=attribute;

            Remove(attribute from format) }

    }

    For each (string in delimiters)

    {

        If(format starts with a delimiter)

        {

            If (currentattribute == " ")      // " " means space character

            {Startdelimiter += delimiter; // append delimiter in to start delimiter string

            }

            Else{

                Enddelimiter = delimiter;}

            Remove( delimiter from the format)}}

        If ( format starts with "")

        { enddelimiter = EOF;

        }

        Format= "",""}

```

*If(currentattribute != “ “){*

*ExtractcurrentAttribute(startdelimiter, enddelimiter, format, words(by  
reference))*

*Startdelimiter=””;*

*Enddelimiter=””;*

*Currentattribute=”” ;}*

The pseudo code basically explains that while extracting the attribute delimiter before the attribute and delimiter after the attribute are checked and applied line by line on the whole file to extract it.

#### **5.6.4 Apply Similarity Matrix**

After extraction of data from the database the next step is to transform that data in a way which makes it suitable for apply data mining techniques. Similarity matrix techniques can be applied on the stored data. To apply similarity matrix the formula used is:

$$d(i,j) = \frac{\sum_{f=1}^n X_{i,j} Y_{i,j}}{\sum_{f=1}^n X_{i,j}}$$

$d(i,j)$  is the distance between two fields in the database. The range of all the distances generated will be between 0 and 1.

#### **Algorithm**

- ✓  $X_{i,j}$  is a binary attribute which can have values 0 or 1 depending on the availability of data.  $X_{i,j}$  is 0 (zero) if one of the two records under consideration is empty. Otherwise the value of  $X_{i,j}$  is 1.

- ✓ Unlike  $X_{i,j}$  the value of  $Y_{i,j}$  depends upon the type of attribute under consideration.

$$Y_{i,j} = 0 \text{ (zero) if } q_{i,f} = q_{i,f} \dots\dots\dots (2.4)$$

The value of Y in this scenario depends upon the similarity of two attributes used for calculation.

Whereas, If the attributes or the data is numerical  $Y_{i,j}$  is calculated by using the formula given below.

$$Y_{i,j} = \frac{|q_{i,f} - q_{i,j}|}{(\max(q_{m,f}) \min(Y_{m,f}))} \dots\dots\dots (2.5)$$

In formula (2.4) max and min functions calculate minimum and maximum values of the columns of a record. The source code for calculating X and Y is given below

```
int calculateX(string val1, string val2){
    if (val1.Trim().Equals("") || val2.Trim().Equals("") || val1.Equals(null) || val2.Equals(null))
        return 0;
    else
        return 1; }

int calculateY(string val1, string val2) {
    if (val1.Contains(val2))
        return 0;
    else
        return 1; }
```

The functions stated above are used to calculate the values of X and Y. The formula (2.5) was never used as no information extracted from the web log file was numerical.

For Calculating similarity matrix, pseudo code is given below:

```
result = ((x[ip] * y[ip]) + (x[date] * y[date]) +  
          (x[protocol] * y[protocol]) +  
          (x[time] * y[time]) +  
          (x[url] * y[url])) /  
          (x[ip] + x[date] + x[protocol] + x[time] + x[url]);
```

Result receives a value within the range 0 and 1. result is then mapped as a single entry of the whole matrix.

### 5.6.5 Calculate Average Similarity

The code below shows the query used to calculate average of the distances calculated by applying similarity matrix.

```
string query = "SELECT AVG(result) FROM Clustering WHERE  
logdata_id LIKE '%" + logdata_id + "%'";
```

The result column resides in the clustering table. The example given explains the concept in detail. Let us consider the field

LogData_id	ip	date	url	time
57024	165.139.87.3	28/Feb/2007	http://www.brai...	22:29:39
57025	202.125.143.68	28/Feb/2007	http://www.brai...	22:29:39

**Fig 5.8: Database view of entries in table LogData**

We want to check the average similarity of “LogData\_id” (57024) with all the other entries in the table.

Entry for “LogData\_id” (57024) in LogData table would be

logdata_id	ip	result	url	protocol	date	time
57024,57025	165.139.87.3 , 202.125.143.68	0.4	http://www.brai...	HTTP/1.0, HTTP...	28/Feb/2007	22:29:39
57024,57026	165.139.87.3 , 193.251.135.123	0.6	http://www.brai...	HTTP/1.0	28/Feb/2007	22:29:39, 22:29...
57024,57027	165.139.87.3 , 203.128.11.15	0.8	http://www.brai...	HTTP/1.0, HTTP...	28/Feb/2007	22:29:39, 22:29...
57024,57028	165.139.87.3 , 202.125.143.68	0.6	http://www.brai...	HTTP/1.0, HTTP...	28/Feb/2007	22:29:39, 22:29...
57024,57029	165.139.87.3 , 202.125.143.68	0.6	http://www.brai...	HTTP/1.0, HTTP...	28/Feb/2007	22:29:39, 22:29...
57024,57030	165.139.87.3 , 202.125.143.68	0.6	http://www.brai...	HTTP/1.0, HTTP...	28/Feb/2007	22:29:39, 22:29...
57024,57031	165.139.87.3 , 202.125.143.68	0.6	http://www.brai...	HTTP/1.0, HTTP...	28/Feb/2007	22:29:39, 22:29...

**Fig 5.9: Database view of entries in table Clustering**

In the figure above distances between two fields is stored in the result field. By taking the average of results field taking in account id (57024) of the first field states the average similarity which saved in the LogData table. The data base of the similarities is stored in LogData table as showed below.

ip	date	url	time	filepath	browser	protocol	statuscode	method	os	port	similarity_i
165.139.87.3	28/Feb/2007	http://www.brai...	22:29:39	/~jba/images/SK...	Mozilla/4.0	HTTP/1.0	200	GET	compatible; MSI...	8614	0.2655737
202.125.143.68	28/Feb/2007	http://www.brai...	22:29:39	/newpage/links/...	Mozilla/4.0	HTTP/1.1	200	GET	compatible; MSI...	1228	0.3852459
193.251.135.123	28/Feb/2007	http://www.brai...	22:29:40	/~aup/IBMS.h1...	Mozilla/5.0	HTTP/1.0	200	GET	Windows; U; Wi...	1561	0.2868852
203.128.11.15	28/Feb/2007	http://brain.net...	22:29:40	/newpage/8.gif	Mozilla/4.0	HTTP/1.1	200	GET	compatible; MSI...	5126	0.4049180
202.125.143.68	28/Feb/2007	http://www.brai...	22:29:40	/newpage/googl...	Mozilla/4.0	HTTP/1.1	200	GET	compatible; MSI...	137	0.3967213
202.125.143.68	28/Feb/2007	http://www.brai...	22:29:40	/newpage/links/...	Mozilla/4.0	HTTP/1.1	200	GET	compatible; MSI...	1993	0.3967213
202.125.143.68	28/Feb/2007	http://www.brai...	22:29:40	/newpage/mp3.gif	Mozilla/4.0	HTTP/1.1	200	GET	compatible; MSI...	115	0.3967213
202.125.143.68	28/Feb/2007	http://www.brai...	22:29:40	/newpage/ll.gif	Mozilla/4.0	HTTP/1.1	200	GET	compatible; MSI...	2760	0.3967213
202.125.143.68	28/Feb/2007	http://www.brai...	22:29:40	/newpage/web.gif	Mozilla/4.0	HTTP/1.1	200	GET	compatible; MSI...	747	0.3967213
165.139.87.3	28/Feb/2007	http://www.brai...	22:29:39	/~jba/images/ne...	Mozilla/4.0	HTTP/1.0	200	GET	compatible; MSI...	4997	0.2557377
202.125.143.67	28/Feb/2007	-	22:28:36	/%7Eaasf/2001...	DA 7.0"	HTTP/1.1	206	GET		57466	0.3836065
165.139.87.3	28/Feb/2007	http://www.brai...	22:29:40	/~jba/images/SK...	Mozilla/4.0 (com...	HTTP/1.0	200	GET		8720	0.2655737
203.128.29.182	28/Feb/2007	http://www.brai...	22:29:40	/webmail/src/rea...	Mozilla/4.0 (com...	HTTP/1.1	200	GET		7554	0.3852459

**Fig 5.10: Similarity averages of one field with others**

### 5.6.6 K-means Algorithm

Clustering based in this report is based on Euclidean k-means. The working of the algorithm is explained in the Pseudo code stated below

**Input Parameters**

Number of Clusters:  $c$

Euclidean Distance:  $Eu$

Data Items:  $t_1, \dots, t_n$

Mean:  $m$

**Initialization**

1. Construct clusters according to the number of clusters given
2. Select frequent Item set according to the similarity values
3. Categorize clusters according to the average similarity provided

**Main Phase**

1. Assign each data item to the cluster having minimum score
2. Relocate items till the mean is constant
3. Search all clusters and assign them in to a new cluster

The approach is based on the comparison of frequent item sets. In each step mean of each cluster is recalculated and the process goes on until there is no change in mean value of each cluster.

## 5.7 Summary

The implementation chapter explained in detail the methods used to develop the front end and the back end (algorithms) of the system. The chapter discussed the data mining algorithm used for clustering the extracted information. Data base operations like read write operations were also explained in detail. Pseudo code of algorithms and extraction methodologies were shown in detail.

# ***Chapter 6***

## ***Testing and Evaluation***

### **6.1 Introduction**

The chapter provides the description of testing strategies applied on the developed system. Different test cases are tested on to the preprocessing application, the underlined database and user defined format module. Finally the requirements defined in third chapter are examined.

## 6.2 Testing the Web Usage Mining Tool

Testing is done on the basis of system flow diagram discussed in the design chapter. According to that flow diagram the pre-processing application should allow the user to define format, that format should be saved in to the LogAnalyzer1 database on request. A file has to be pre-processed and data extraction should be performed. All the extracted data should be saved in to the LogData table. Data mining algorithm should be applied on the data after acquiring some clustering information from the user and results should be displayed in the forms of graphs. All the process is tested further in the chapter.

### Step 1: Format Definition

The step will test whether the “defining format” requirement in the section 3.2.1 is satisfied or not.

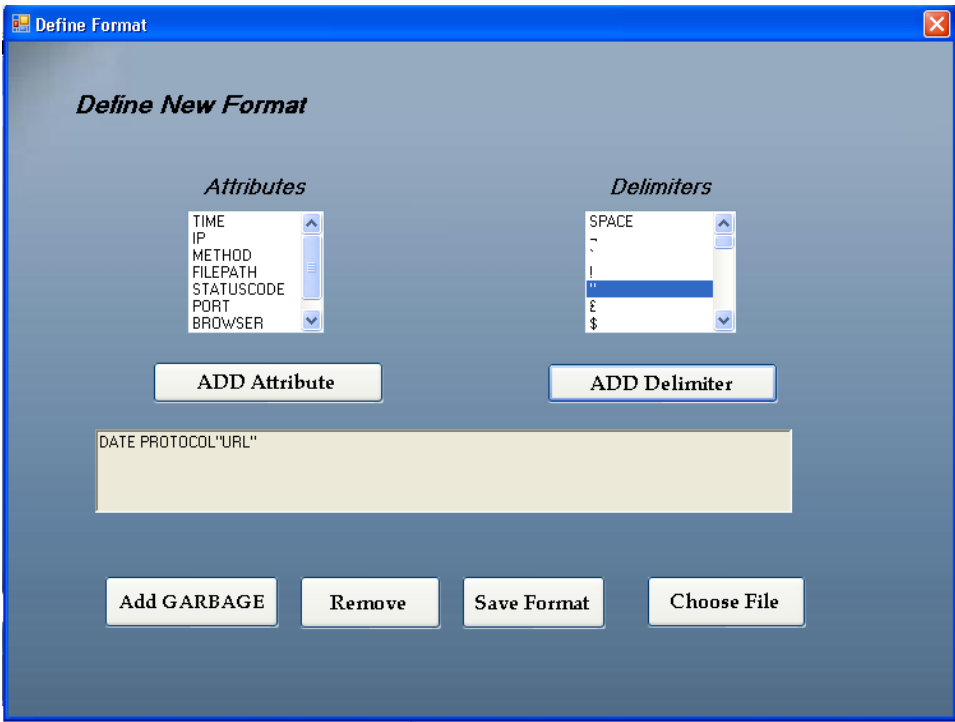
1. ) Use already saved format .Already saved formats by the users should be displayed. The format is retrieved from the data base.



**Fig 6.1: Saved Format window**

- 2.) Defining user's own format and saving that format in the database if desired. User should be able to insert attributes, delimiters and garbage in the rich text box. Attributes should be removed from the attribute list and added in the rich

text box and on removing an attribute it should be added back in the attribute list.



**Fig 6.2: Define Format window**

After defining a format user can either choose a file to apply format on or save format first and apply format. If user chooses to save the format it would appear in the database like

Table - dbo.format		Summary
	Format_Id	format
▶	18	IP - - [DATE:TIME GARBAGE] "METHOD FILEPATH PROTOCOL" STATUSCODE PORT "URL" "BROWSER (OS)GARBAGE"
*	NULL	NULL

**Fig 6.3: User defined format in DB**

The interface displays the data base table having the user defined format. All the formats will be retrieved from this table, next time the system is initiated.

## Step 2: Data Extraction

The testing step refers to the requirement “data cleaning” in the section 3.2.1. The step will confirm that data is extracted to the right tables and save in the LogData table and format is applied on the log files exactly.

- 1.) Data should be under correct attributes in the database as defined in the format.
- 2.) Old values in the data base should be removed first before the new insertion operation. This will help in avoiding overflows.

The database below shows extracted data followed by the format used.

Table - dbo.LogData	Table - dbo.format	Summary									
LogData_id	ip	date	url	time	filepath	browser	protocol	statuscode	method	os	port
57147	165.139.87.3	28/Feb/2007	http://www.brai...	22:29:39	/~jba/images/SK...	Mozilla/4.0	HTTP/1.0	200	GET	compatible; MSI...	8614
57148	202.125.143.68	28/Feb/2007	http://www.brai...	22:29:39	/newpage/links/...	Mozilla/4.0	HTTP/1.1	200	GET	compatible; MSI...	1228
57149	193.251.135.123	28/Feb/2007	http://www.brai...	22:29:40	/~aup/BMS.h1...	Mozilla/5.0	HTTP/1.0	200	GET	Windows; U; Wi...	1561
57150	203.128.11.15	28/Feb/2007	http://brain.net...	22:29:40	/newpage/8.gif	Mozilla/4.0	HTTP/1.1	200	GET	compatible; MSI...	5126
57151	202.125.143.68	28/Feb/2007	http://www.brai...	22:29:40	/newpage/googl...	Mozilla/4.0	HTTP/1.1	200	GET	compatible; MSI...	137
57152	202.125.143.68	28/Feb/2007	http://www.brai...	22:29:40	/newpage/links/...	Mozilla/4.0	HTTP/1.1	200	GET	compatible; MSI...	1993
57153	202.125.143.68	28/Feb/2007	http://www.brai...	22:29:40	/newpage/mp3.gif	Mozilla/4.0	HTTP/1.1	200	GET	compatible; MSI...	115
57154	202.125.143.68	28/Feb/2007	http://www.brai...	22:29:40	/newpage/ll.gif	Mozilla/4.0	HTTP/1.1	200	GET	compatible; MSI...	2760
57155	202.125.143.68	28/Feb/2007	http://www.brai...	22:29:40	/newpage/web.gif	Mozilla/4.0	HTTP/1.1	200	GET	compatible; MSI...	747
57156	165.139.87.3	28/Feb/2007	http://www.brai...	22:29:39	/~jba/images/ne...	Mozilla/4.0	HTTP/1.0	200	GET	compatible; MSI...	4997
57157	202.125.143.67	28/Feb/2007	-	22:28:36	/%7Easif/2001...	DA 7.0"	HTTP/1.1	206	GET		57466
57158	165.139.87.3	28/Feb/2007	http://www.brai...	22:29:40	/~jba/images/SK...	Mozilla/4.0 (com...	HTTP/1.0	200	GET		8720
57159	203.128.29.182	28/Feb/2007	http://www.brai...	22:29:40	/webmail/src/rea...	Mozilla/4.0 (com...	HTTP/1.1	200	GET		7554

**Fig 6.4: Extracted data in LogData table**

Format defined for this file is:

**IP - - [DATE:TIME GARBAGE] "METHOD FILEPATH PROTOCOL"  
STATUSCODE PORT "URL" "BROWSER (OS)GARBAGE"**

### Step 3: Data Mining Algorithm

The step will confirm that the data mining algorithm acquires the correct values of data from the data base saved by the Similarity Matrix technique. The step will also confirm the requirement “User Interaction” defined in section 3.2.1. System should get the desired number of clusters and clustering process starts when the user presses the process button.

- 1.) The clustering table should receive clustering information.
- 2.) Distances should be inserted in the result column.
- 3.) Average similarities are calculated and inserted in front of correct id's.

The following figures show the successful insertion of data in to the database. Average similarities are inserted in to the LogData table and distances among records are inserted in to the Clustering table.

ip	date	url	time	filepath	browser	protocol	statuscode	method	os	port	similarity_
165.139.87.3	28/Feb/2007	http://www.brai...	22:29:39	/~jba/images/SK...	Mozilla/4.0	HTTP/1.0	200	GET	compatible; MSI...	8614	0.2655737
202.125.143.68	28/Feb/2007	http://www.brai...	22:29:39	/newpage/links/...	Mozilla/4.0	HTTP/1.1	200	GET	compatible; MSI...	1228	0.3852459
193.251.135.123	28/Feb/2007	http://www.brai...	22:29:40	/~aup/IBMS.h1...	Mozilla/5.0	HTTP/1.0	200	GET	Windows; U; Wi...	1561	0.2868852
203.128.11.15	28/Feb/2007	http://brain.net...	22:29:40	/newpage/8.gif	Mozilla/4.0	HTTP/1.1	200	GET	compatible; MSI...	5126	0.4049180
202.125.143.68	28/Feb/2007	http://www.brai...	22:29:40	/newpage/googl...	Mozilla/4.0	HTTP/1.1	200	GET	compatible; MSI...	137	0.3967213
202.125.143.68	28/Feb/2007	http://www.brai...	22:29:40	/newpage/links/...	Mozilla/4.0	HTTP/1.1	200	GET	compatible; MSI...	1993	0.3967213
202.125.143.68	28/Feb/2007	http://www.brai...	22:29:40	/newpage/mp3.gif	Mozilla/4.0	HTTP/1.1	200	GET	compatible; MSI...	115	0.3967213
202.125.143.68	28/Feb/2007	http://www.brai...	22:29:40	/newpage/ll.gif	Mozilla/4.0	HTTP/1.1	200	GET	compatible; MSI...	2760	0.3967213
202.125.143.68	28/Feb/2007	http://www.brai...	22:29:40	/newpage/web.gif	Mozilla/4.0	HTTP/1.1	200	GET	compatible; MSI...	747	0.3967213
165.139.87.3	28/Feb/2007	http://www.brai...	22:29:39	/~jba/images/ne...	Mozilla/4.0	HTTP/1.0	200	GET	compatible; MSI...	4997	0.2557377
202.125.143.67	28/Feb/2007	-	22:28:36	/%7Easf/2001... DA 7.0"	DA 7.0"	HTTP/1.1	206	GET		57466	0.3836065
165.139.87.3	28/Feb/2007	http://www.brai...	22:29:40	/~jba/images/SK...	Mozilla/4.0 (com...	HTTP/1.0	200	GET		8720	0.2655737
203.128.29.182	28/Feb/2007	http://www.brai...	22:29:40	/webmail/src/rea...	Mozilla/4.0 (com...	HTTP/1.1	200	GET		7554	0.3852459

**Fig 6.5: Similarity Values in front of Id's**

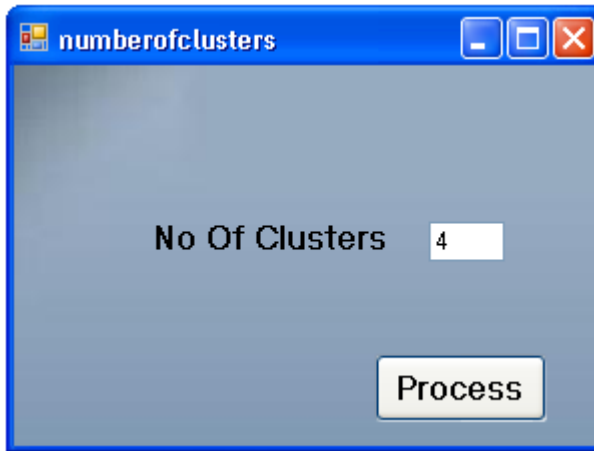
logdata_id	ip	result	url	protocol	date	time
57024,57025	165.139.87.3 , 202.125.143.68	0.4	http://www.brai...	HTTP/1.0, HTTP...	28/Feb/2007	22:29:39
57024,57026	165.139.87.3 , 193.251.135.123	0.6	http://www.brai...	HTTP/1.0	28/Feb/2007	22:29:39, 22:29...
57024,57027	165.139.87.3 , 203.128.11.15	0.8	http://www.brai...	HTTP/1.0, HTTP...	28/Feb/2007	22:29:39, 22:29...
57024,57028	165.139.87.3 , 202.125.143.68	0.6	http://www.brai...	HTTP/1.0, HTTP...	28/Feb/2007	22:29:39, 22:29...
57024,57029	165.139.87.3 , 202.125.143.68	0.6	http://www.brai...	HTTP/1.0, HTTP...	28/Feb/2007	22:29:39, 22:29...
57024,57030	165.139.87.3 , 202.125.143.68	0.6	http://www.brai...	HTTP/1.0, HTTP...	28/Feb/2007	22:29:39, 22:29...
57024,57031	165.139.87.3 , 202.125.143.68	0.6	http://www.brai...	HTTP/1.0, HTTP...	28/Feb/2007	22:29:39, 22:29...

**Fig 6.6: Clustering information in Clustering Table**

## Step 4: Generation of Results

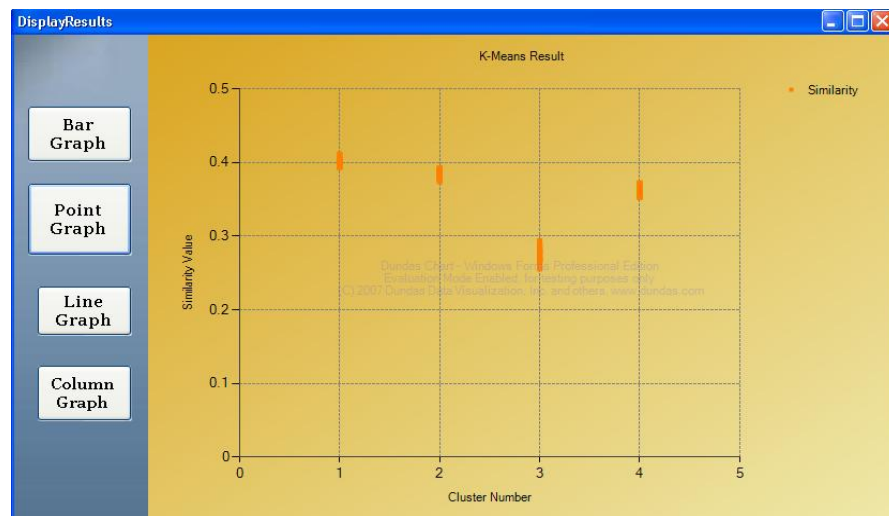
This step will confirm that the data mining algorithm generates results according to the number of clusters given by the user and result generated according to the average distances stored in the data base.

- 1.) Get input from the user



**Fig 6.7: User input for number of clusters**

User has asked the system to generate four clusters. This input is passed on to the data mining algorithm and the graphs should be generated. The figure below displays the results.



**Fig 6.8: Result generation having four clusters**

The above figure confirms that four clusters are generated as defined by the user. Cluster numbers are displayed on the X-axis and corresponding similarity values on Y-axis.

## **6.3 Evaluation**

The following section describes the evaluation of the system in terms of its output. The evaluation procedure is very vital as it uncovers a lot of deficiencies to the developer. To evaluate the system two web log files having different formats were tested.

### **6.3.1 Evaluation in terms of Architecture**

The eventual development of the system enabled the user to process Web log files provided to it. A database schema was also created used to store Extracted data as well as the clustering information and cluster results. The developed system provides basis for or a kind of a new frame work on which more complex and advanced systems can built upon.

### **6.3.2 Evaluation in terms of Application's functionality**

The user introduced two kinds of web log data during the testing phase. Each file had around 26000 records. In addition to this the records generated by applying similarity matrix on the extracted data were around 250,000.

The system has got some limitations as well; these limitations can be justified by the lack of time. For an incorrect input of format by the user the system still extracted data and insertion of attributes was performed correctly but in wrong database fields. This however is a big drawback of the system which is a compromise with the underlined database.

In terms of the user interface, it is simple and user friendly. The process of defining an interface is relatively easy and a new innovation of this project. The system also generates basic errors for example choosing a file without a format is not allowed.

### **6.3.4 Evaluation in terms of Traversal Speed**

Processing of Web log files originate a large amount of data. Parsing a web log file was efficient but preparing that extracted data for clustering introduced some time delays. This was checked by a time calculating functionality if underlined database. The

database processed roughly 1050 lines/ sec. This processing of data included reading a record and storing the new prepared data in Clustering table.

The amount of time taken by the data preparation step can be reduced by performing the operation on multiple threads.

### 6.3.5 Evaluation in terms of Database

Database was evaluated in terms of data integrity. Missing and inconsistent values in the database are a big threat to the system. In order to ensure data integrity the missing fields were replaced with the default values (space character) in this case.

Scalability of the system can be explained by the figure below:

	Update Frequency	Ability to Change Application	Data Partitionability	Data Coupling
Scalable Shared Databases	Read Only.	Little or no change required.	No requirement.	No requirement.
Peer-to-Peer Replication	Read mostly, no conflicts.	Little or no change required.	No requirement.	No requirement.
Linked Servers	Minimize cross-database updates.	Minor changes.	Not generally required.	Very important to have low coupling.
Distributed Partitioned Views	Frequent updates OK.	Some changes may be required.	Very important.	Little impact.
Data-Dependent Routing	Frequent updates OK.	Significant changes possible.	Very important.	Low coupling may help some applications.
Service-Oriented Data Architecture	Frequent updates OK.	Extensive changes required.	Not generally required, unless combined with DDR.	Low coupling between services required.

**Fig 6.9: SQL Server Scalability [30]**

### 6.3.6 Experimental Results

- **Scenario 1:**

File name: LogData1

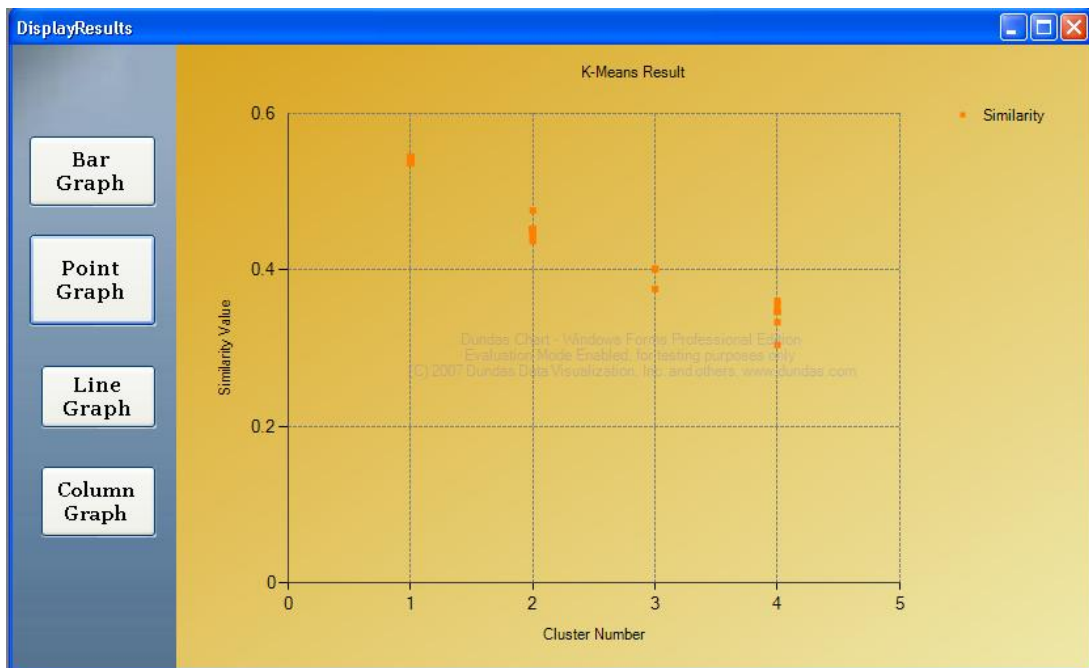
Size: 10Kb

No of Lines: 25

No of Clusters Input: 4

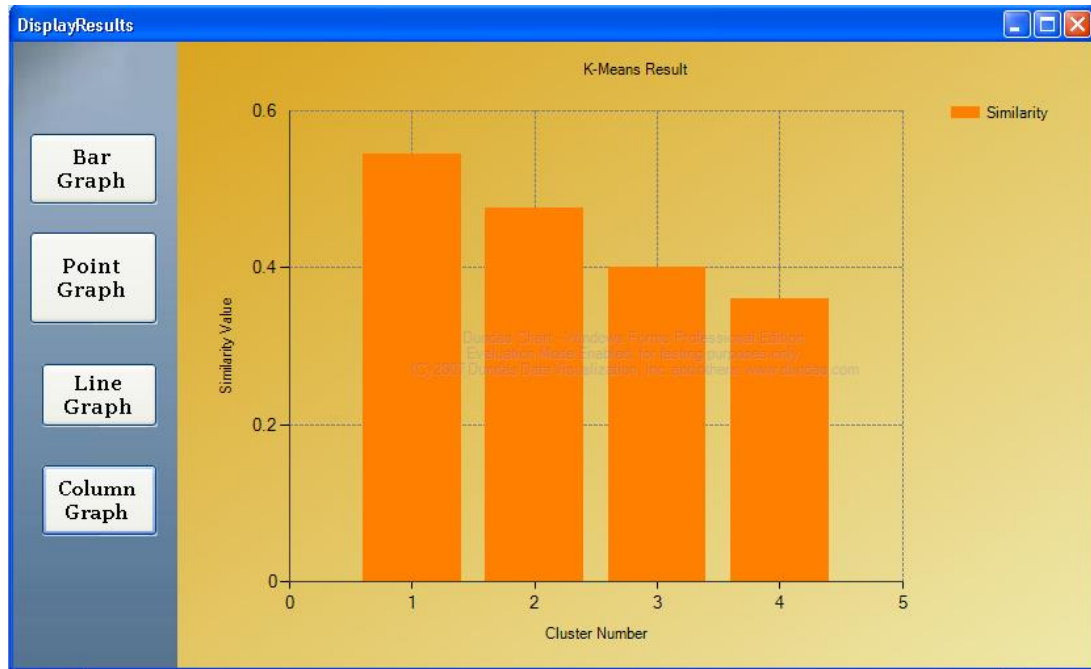
✓ **Results:**

**Graph 1:**



**Fig 6.10: Point Graph for LogData1**

## Graph2



**Fig 6.11: Column Graph for LogData**

- **Scenario 2**

File name: LogData2

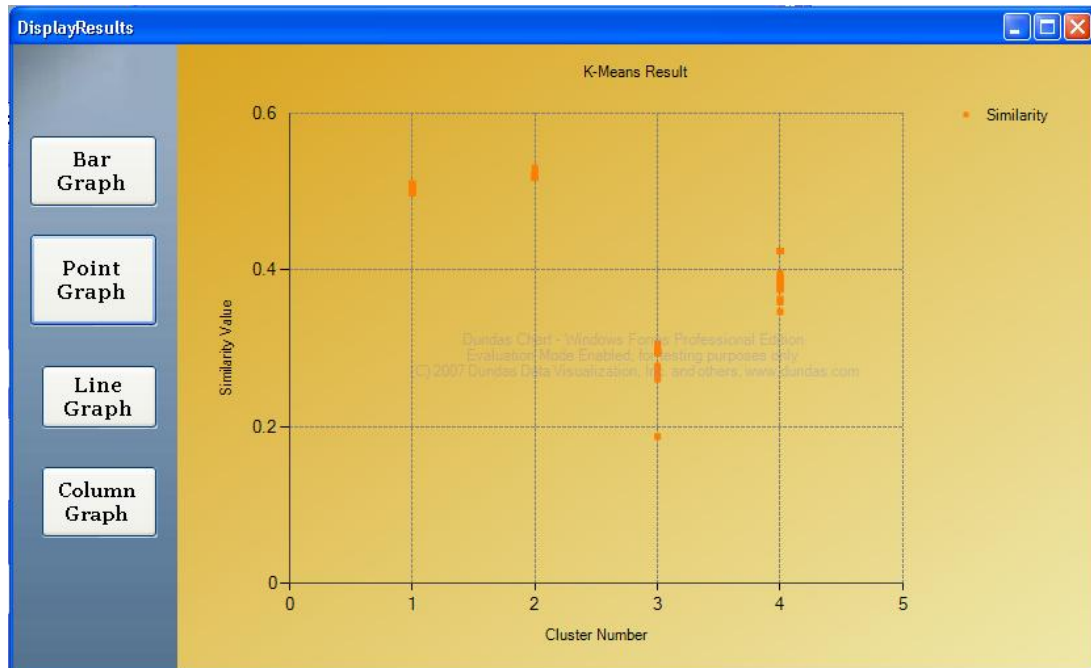
Size: 11Kb

No of Lines: 25

No of Clusters Input: 4

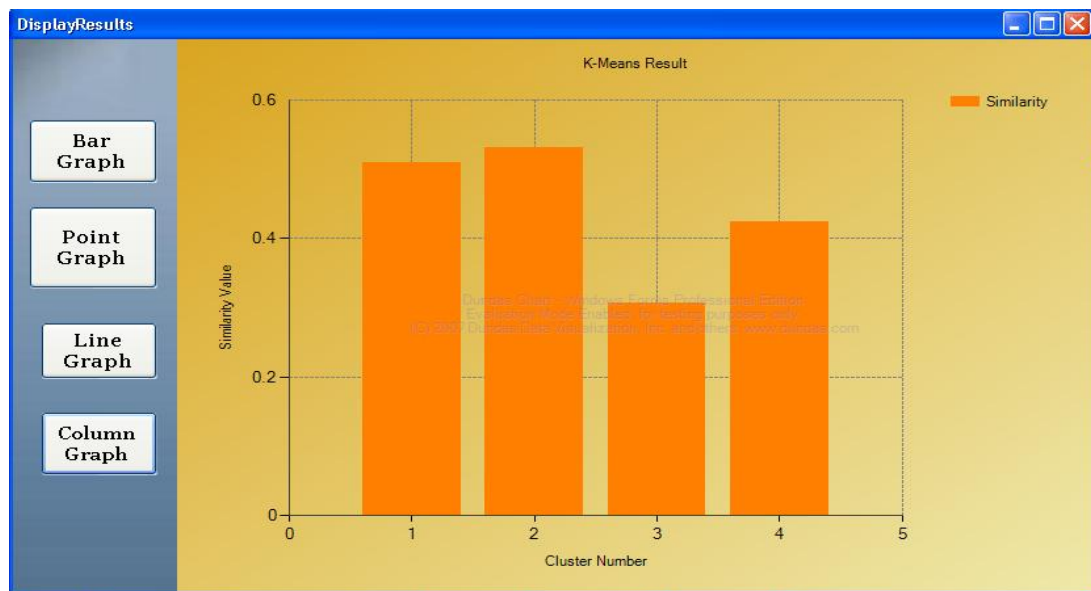
✓ **Results:**

**Graph 1:**



**Fig 6.12: Point Graph for LogData**

**Graph 2:**



**Fig 6.13: Column Graph for LogData2**

### Comparison of two Log Files:

	LogData1	LogData2
<b>SIZE</b>	<b>11Kb</b>	<b>10Kb</b>
<b>Time Taken</b>	<b>5 sec</b>	<b>4 sec</b>
<b>No Of Cluster</b>	<b>4</b>	<b>4</b>
<b>No. of Lines</b>	<b>25</b>	<b>25</b>
<b>Maximum Similarity Value</b>	<b>Cluster NO: 1 similarity value: 0.56 approx</b>	<b>Cluster NO: 2 similarity value: 0.53 approx</b>

## 6.4 System's Drawbacks

Regarding the implementation there were some flaws detected in the system during the evaluation phase.

- The biggest flaw of the system is that it does not validate the format entered by the user with the format of the web log file. It can be a future improvement. The flaw introduces wrong information in the database.
- Due to the structure of the programming language(c#), a lot of functionalities have same structure. This resulted in large pieces of code existing twice or more in the whole source code.

## 6.5 Summary

The chapter presented the structural testing of the whole system. Two real time scenarios were applied on the system using two web log files with different formats. The results obtained were analyzed and compared with the similarities values. Finally systems draw backs were identified.

# ***Chapter 7***

## ***Conclusion and Future Work***

### **7.1 Introduction**

The chapter presents the conclusion and lessons learnt by the whole process of development and research. The chapter starts with the overview of the whole dissertation structure. Aims achieved and the knowledge obtained by studying references regarding the dissertation is discussed. Finally, a number of suggestions are made for future which can help a developer to improve or upgrade the system.

## **7.2 Dissertation Structure**

The report was structured in such a way that every chapter functioned as basis for the next chapter. Each chapter of the report reflected the domain or work done. All the material written in the dissertation outlines the methodology used to apply data mining operations on data.

## **7.3 Development Methodology**

The development in order to carry out research and writing consisted of the following steps.

1. Background research was the first step taken in order to start development and writing up. The phase included examination of Web Log Files, format of these web log files, techniques of data extraction and finally some data mining techniques were also taken in to account.
2. After making a solid ground and gathering sufficient information about the problem domain. Requirements were gathered regarding the model of input data (Database), the front end (GUI) and the back end Algorithm (Clustering). Requirements gathered were then categorized. The requirement phase served as the basis for the implementation phase.
3. Designing the tool or the system based on the requirements gathered. To make the design more solid, powerful UML diagrams was developed which provided great help while developing the system.
4. On the basis of visual diagrams and design concepts previously discussed, the system was implemented. A pre-processing application which was used to read or parse the Web Log file, the under lined data base, the data extraction algorithm and the data mining algorithm used to perform cluster analysis. The programming used for implementation of the design is “Dot. Net” and the data base is implemented in “SQL Server 2005”.

5. The system is tested and evaluated in order to validate system requirements. Different test cases are applied on to the developed tool and results are checked. The results are analyzed and conclusions are drawn.

## **7.4 Lessons Learned**

- ✓ Useful lessons were derived from the research carried out in the domain of Web Usage Mining. The research helped me to learn a lot of concepts regarding web content and user profile mining.
- ✓ Literature review of research papers, scientific articles, and online journals helped a lot to improve the scientific background of the report and provided useful knowledge regarding present trends and technologies.
- ✓ To design the web usage mining tool, UML diagrams were developed. These UML diagrams can prove to be a great help in future developments.
- ✓ Log files were analyzed in detail while implementing the tool. The deep study of these log files helped me to understand user access patterns and different formats used by these log files.
- ✓ To implement the system, language used was c# 2005 on a dot .Net platform. In addition a lot of C# programming principles were learnt and programming sense was improved. To access the data base SQL server 2005 was used. Principles relating to SQL server also learnt.
- ✓ Finally, the whole software life cycle was learnt. Methodology to create a project from scratch in a limited period of time was learnt. An efficient project plan was made and acted upon.

## **7.5 Summary of the Project**

As previously discussed in the dissertation, World Wide Web is massive source of information available online. Web site publishers in order to compete in this massive industry try to keep more and more users intact.

User activities are stored in web log files placed at servers or local systems. These activities can be used to analyze user access patterns to show progress of different web pages or areas. The dissertation investigated the domain of extraction of data and process extracted data by applying different data mining techniques. Data resides in log files having different formats. Results generated by these mining techniques are very useful for developers, analysts and publishers.

A number of conclusions were derived after implementing the data mining tool. The main objectives of the project (Extraction, Data mining) were achieved and work fine having some minor mistakes regarding extraction of data. The application does not require a user to be a data mining expert as far as using the application is concerned, but to benefit from the results generated some data mining knowledge is needed. The system uses mechanisms of exception handling and error messages are generated depending upon the scenario. The tool was able to understand the log file pattern defined by the user and apply it to the log file storing the information to correct fields.

The extraction algorithm is a simple one. The algorithm works on the basis of the format defined by the user. The correctness of format is a very vital part. The algorithm splits delimiters and attributes, the extraction process acts line by line on the whole line returning tuple of data. The main drawback of the system is that user defined format is not checked with the format of the web log file. The solution for it is discussed in Future work section (7.5). The k-means algorithm was chosen for clustering. The algorithm worked accurately on large data sets. Processing time depend on the size of file given as an input to the system. The algorithm received no of clusters as its input and on the basis of similarities formed clusters till the mean of each cluster does not change any more. The results generated and the patterns discovered proved to be exact.

Overall the whole experience of research in areas like knowledge discovery in data bases and data mining was a pleasant and a challenging task. Current technologies and trends were gone through helping to understand the basis of data mining concepts. The design and implementation was built up on the scientific knowledge gained by the research and study of similar systems. The scope of the project is not limited as the system includes a wide range of topics like data mining, relational data bases, text processing and result generation.

## 7.6 Future Work

The future work of the project depends upon the requirements. The main reason for degrading the requirements was lack of time. Some future work suggestions are as follows:

- ✓ The main improvement which can be introduced to the system might be validation of format. This will not allow the user to input incorrect format to the system. The example below shows the suggestion more clearly.

IP	TIME	DATE	GARBAGE	METHOD	URL	BROWSER
193.251.135.123	[28/Feb/2007	22:29:40	+0500]	"GET	http://www.brain.net.pk/~aup/	"Mozilla/5.0

**Fig 7.1 Wrong User input**

The figure shows that format input is wrong in case of time and date as time is defined for date and date for time. The present system will treat the format as it is defined without checking whether it is time or date. The newly improved system may have the functionality to read all the characters and validate the format of each attribute defined. This will not only increase reliability but results will also be generated on correct data input.

- ✓ There can be administrative functionalities provided to the user which help him to use his/her own database instead of using SQL server only.
- ✓ Another improvement which can be introduced in to the system is to save the newly generated results. By using this functionality a user can compare old results with the newly generated facts to analyze the changes or progress of a system or websites in case of web log data.

## **7.7 Summary**

The chapter concluded the dissertation. Knowledge gained from the development process was highlighted along with the brief description of project phases. Useful conclusions drawbacks of the system were stated. Future enhancements and solutions to drawbacks of the system were also provided.

## A.) REFERENCES

- [1] Doru Tanasa and Brigitte Trousse (2004), “**Advanced Data Pre-Processing for Intersites Web Usage Mining**” AxIS Project Team, INRIA Sophia Antipolis, (IEEE), pp 59-65.
- [2] B. Mobasher and M. P. Singh, ed. (2004) “**Web Usage Mining and Personalization**”, CRC Press LLC pp 2 of 31
- [3] Richard Miller (Aug 9, 2007), Net craft Web Server Survey. © Net Craft 2002  
<http://www.netcraft.com/survey/archive.html>
- [4] W3C (1995), “**Logging Control in W3C httpd**”  
<http://www.w3.org/Daemon/User/Config/Logging.html>
- [5] W3C (1995),” **Extended Log File Format**”  
<http://www.w3.org/TR/WDlogfile.html>
- [6] Dundas data Visualization Inc,(2007)“**Dundas Chart for .NET** “  
<http://www.dundas.com/Products/Chart/NET/Why/SuccessStories/index.aspx>
- [7] Deitel and Deitel(2007),“**Book: Visual C# How to Program**”  
ISBN: 0-13-152523-9
- [8] Kyungpook National University (2006), “**Requirements Definitions and Dictionary**”<http://woorisol.kyungpook.ac.kr/lab/prof/SoftEng/ch7.htm>
- [9] Ruth Malan and Dana Bredemeyer (Mar 08, 2001), “**Architecture Resources for Enterprise Advantage**”, BREDEMEYER CONSULTING.
- [10] Shehzad Rafique, (2003), “**Non-Functional requirements Template**” (Strategic Planning & Architecture Wing) Electronic Government Directorate Pakistan.
- [11] Shangcai Ma, Lizhen Liu, Hantao song, (2003) “**Web Usage Mining Supporting E- Business**” 8<sup>th</sup> international conference(IEEE).

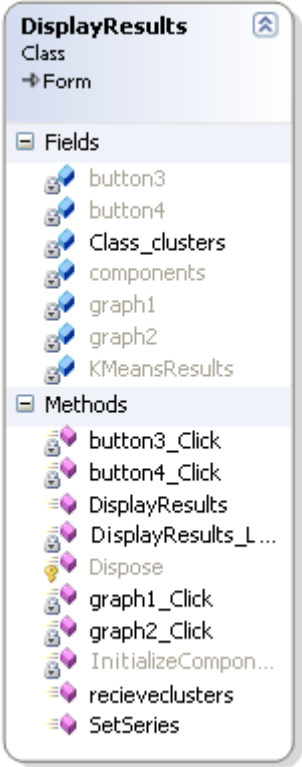
- [12] Jaideep Srivastava, Robert Cooley, Mukund Deshpande, Pang-Ning Tan (2000), ***Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data***, University of Minnesota, Minneapolis.
- [13] Fayyad, U., Piatetsky-Shapiro, G. and Uthurusamy R., (1996), ***Advances In Knowledge Discover and Data Mining***, AAAI Press/ The MIT Press, ISBN 0-262-56097-6 Principles of Knowledge Discover in Databases Module tutorial, University of Alberta, Canada
- [14] B. Mobasher and M. P. Singh, ed. (2004 CRC Press LLC) ***Web Usage Mining and Personalization***, pp 2 of 31
- [15] Tanya Berezin, (Jun 14, 1999), ***Writing Software Requirements Document***, pp 4 of 23.
- [16] Brusilovsky, P., Kobsa, A., Nejdl, W. (2006) ***Data Mining for Personalization. In The Adaptive Web: Methods and Strategies of Web Personalization***, Lecture Notes in Comp Sc, Vol. 4321. Springer-Verlag, Berlin Heidelberg.
- [17] Feng Zhang, Hui-You Chang (Nov. 2002), ***Research and development In Web Usage Mining System—Key Issues and Proposed Solutions a Survey***. pp 986 of 990.
- [18] Hsien Ting, Chris Kimble and Daniel Kudenko (2007) ***Applying Web Usage Mining Techniques to Discover Potential Browsing Problems of Users*** Seventh IEEE International Conference on Advanced Learning Technologies.
- [19] Marc Millon, (1999), ***Creative Content for the Web***, published by : Intellect Books School of Art and Design. Exeter UK.
- [20] Steven Smith, ***LIVING WEBSITE – Understanding the life Cycle***   
<http://www.unitedfocus.com.au/livingwebsites/inside.html>

- [21] Jiawei Han, Micheline Kamber,(2006) “**Data Mining: Concepts and Techniques**” second Edition
- [22] Two Crows Corporation, “**Introduction to Data Mining and Knowledge Discovery**”, Third Edition (Potomac, MD: Two Crows Corporation, 1999); Pieter Adriaans and Dolf Zantinge, Data Mining (New York: Addison Wesley, 1996).
- [23] Groth. G., (2000) “ **Data mining (Building Competitive Advantage)**” , Prentice Hall PTR, ISBN 0-13-086271-1
- [24] **Barnette** ND, McQuain WD, (1998), “**Software Proces Models**”,Computer Science Dept Va Tech.
- [25] Jaideep Srivastava, Prasanna Desikan, Vipin Kumar University of Minnesota, “**Web Mining accomplishments & Future Directions**”, pp 9-22 of 148
- [26] R. Cooley, B. Mobasher, and J. Srivastava,(1997) “**Web Mining: Information and Pattern Discovery on the World Wide Web**”, ISBN: 1082-3409197,( IEEE).
- [27] Bamshad Mobasher, 1997. “**Clustering and Classification**” at <http://maya.cs.depaul.edu/~mobasher/webminer/survey/node17.html#SECTION00032400000000000000>
- [28] Pavel Berkhin, 2002, “**Survey of Clustering Data Mining Techniques**” Accrue Software, Inc.
- [29] Yiannis Kanellopoulos , Thimious Dimopulos, Christos Tjortjis, Christos Makris, (2006) “**Mining Source Code Elements for Comprehending Object Oriented Systems and Evaluating their Maintainability**”.Volume 8, Issue 1
- [30] Microsoft Corporation (2006), “**Scaling Out SQL Server 2005**”  
<http://msdn2.microsoft.com/en-us/library/aa479364.aspx>  
[Accessed 20 August 2007]

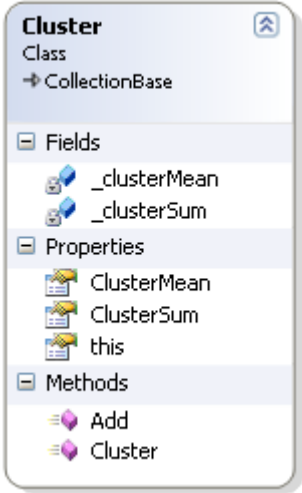
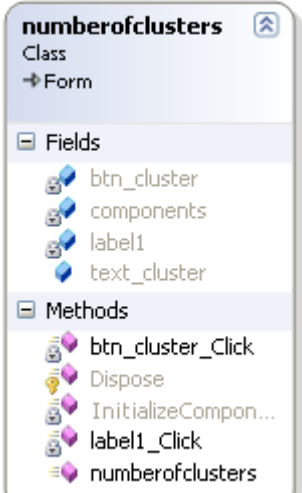
**Appendix A**  
**Class View 1**

<b>Main_Menu</b>	<b>Clustering</b>
<div></div>	<div></div>

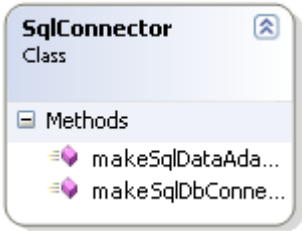
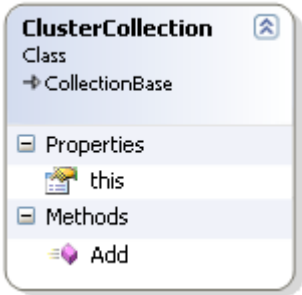
## Class View 2:

Format	Display Results
	 <p>The screenshot shows the 'DisplayResults' class in the Class View. It is a 'Class' and a 'Form'. The 'Fields' section lists: button3, button4, Class_clusters, components, graph1, graph2, and KMeansResults. The 'Methods' section lists: button3_Click, button4_Click, DisplayResults, DisplayResults_L..., Dispose, graph1_Click, graph2_Click, InitializeCompon..., recieveclusters, and SetSeries.</p>

### Class View 3

Cluster	numberofclusters
 <p><b>Cluster</b> Class ↳ CollectionBase</p> <p><b>Fields</b></p> <ul style="list-style-type: none"> <li>_clusterMean</li> <li>_clusterSum</li> </ul> <p><b>Properties</b></p> <ul style="list-style-type: none"> <li>ClusterMean</li> <li>ClusterSum</li> <li>this</li> </ul> <p><b>Methods</b></p> <ul style="list-style-type: none"> <li>Add</li> <li>Cluster</li> </ul>	 <p><b>numberofclusters</b> Class ↳ Form</p> <p><b>Fields</b></p> <ul style="list-style-type: none"> <li>btn_cluster</li> <li>components</li> <li>label1</li> <li>text_cluster</li> </ul> <p><b>Methods</b></p> <ul style="list-style-type: none"> <li>btn_cluster_Click</li> <li>Dispose</li> <li>InitializeCompon...</li> <li>label1_Click</li> <li>numberofclusters</li> </ul>

### Class View 4

SqlConnection	ClusterCollection
 <p><b>SqlConnection</b> Class</p> <p><b>Methods</b></p> <ul style="list-style-type: none"> <li>makeSqlDataAda...</li> <li>makeSqlDbConne...</li> </ul>	 <p><b>ClusterCollection</b> Class ↳ CollectionBase</p> <p><b>Properties</b></p> <ul style="list-style-type: none"> <li>this</li> </ul> <p><b>Methods</b></p> <ul style="list-style-type: none"> <li>Add</li> </ul>

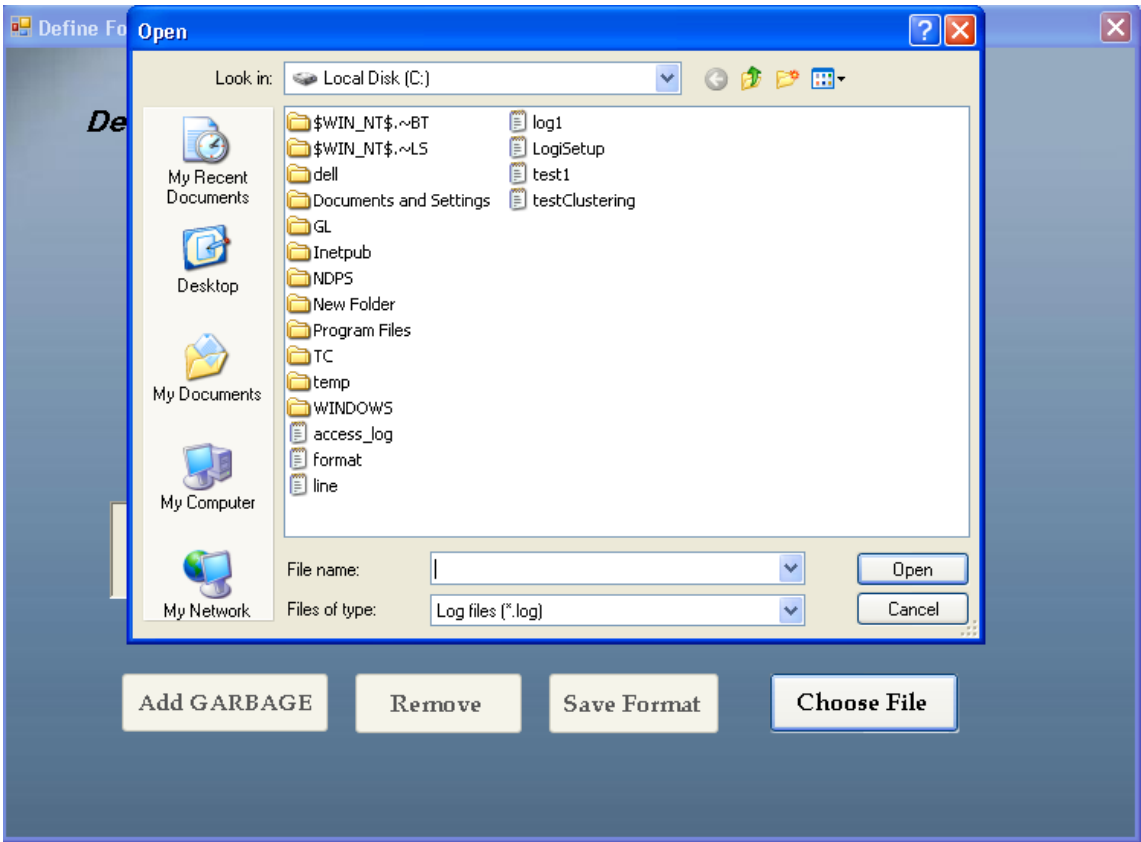
## Appendix B

### Interface Diagram 1



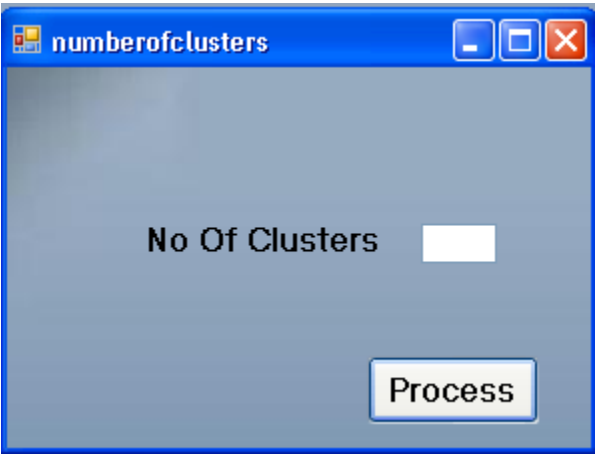
**Fig : Main Interface**

# Interface Diagram 2



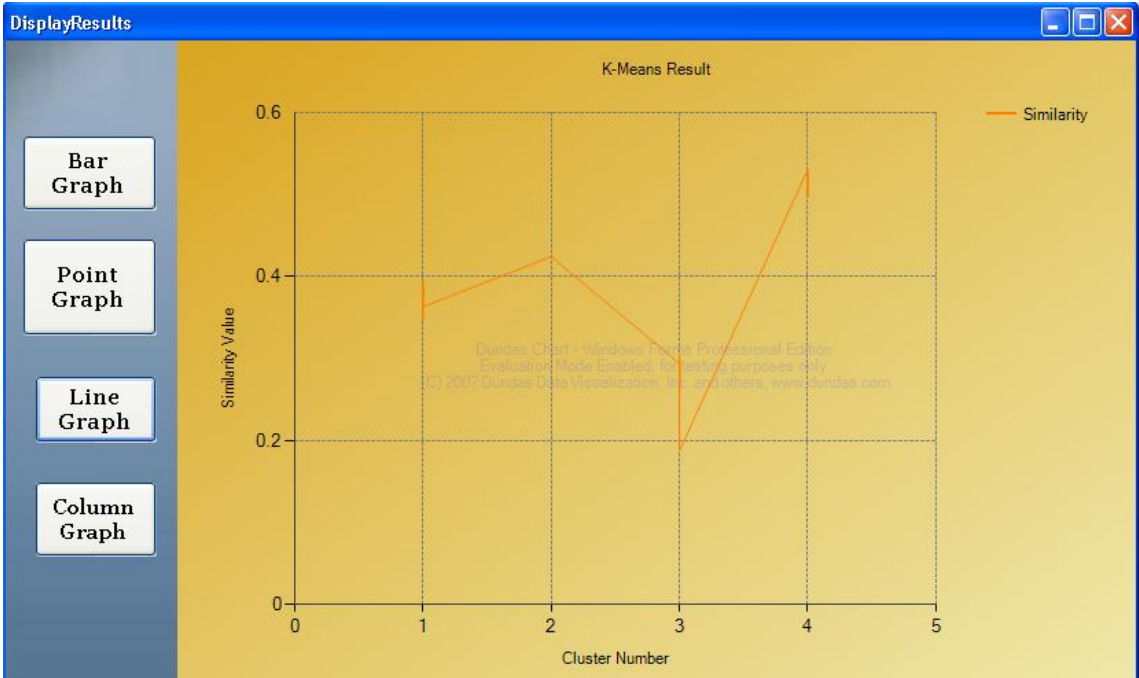
**Fig : Open File Dialogue Box**

### Interface Diagram 3:



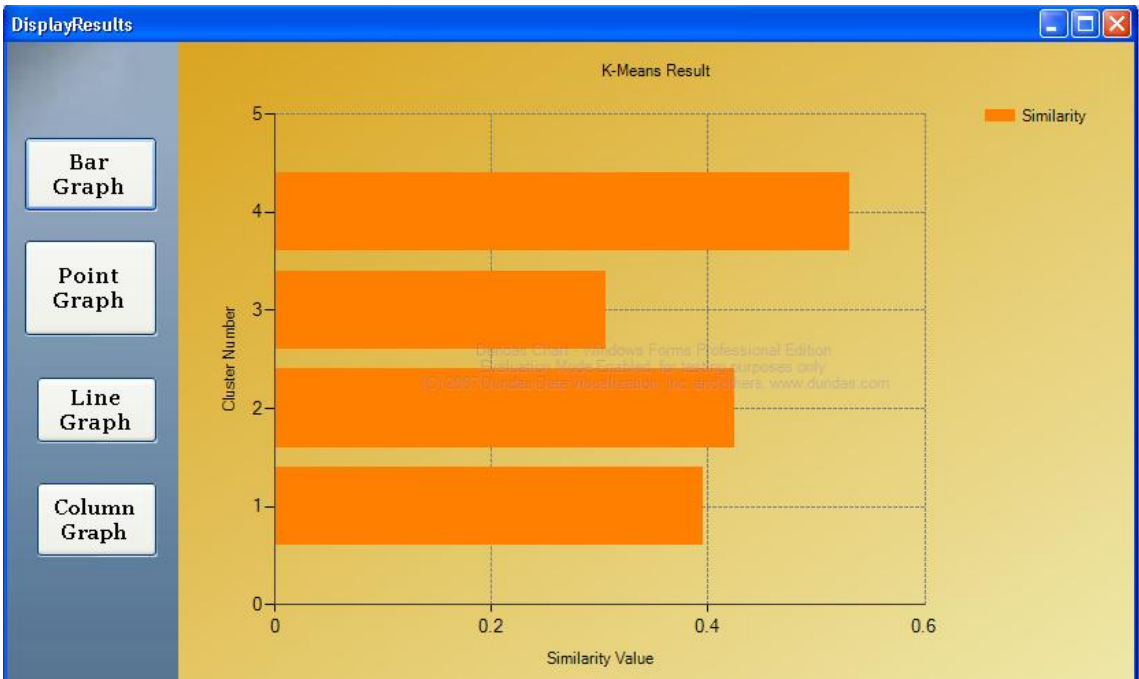
**Fig : Cluster Input Interface**

### Interface Diagram 4:



**Fig : Line Graph for Results**

**Interface Diagram5:**



**Fig : Bar Graph for Results**

## Appendix C

### Sample Web Log Files 1

2005-03-02 23:04:37 /Default.asp 85.164.147.42 Mozilla/5.0+(Macintosh;+U;+PPC+Mac+OS+X;+no  
no)+AppleWebKit/125.5.6+(KHTML,+like+Gecko)+Safari/125.12  
ASPSESSIONIDSSBDBRAT=BENDCNLCGDEPKPGPPFODJBOP;+Dato=02%2E03%2E2005+10%3A18%3A18 -

2005-03-02 23:04:37 /gfx/nyheter.jpg 85.164.147.42 Mozilla/5.0+(Macintosh;+U;+PPC+Mac+OS+X;+no-  
no)+AppleWebKit/125.5.6+(KHTML,+like+Gecko)+Safari/125.12  
ASPSESSIONIDSSBDBRAT=BENDCNLCGDEPKPGPPFODJBOP;+Dato=02%2E03%2E2005+10%3A18%3A18 <http://www.brsen.com/>

2005-03-02 23:04:37 /gfx/newstop.gif 85.164.147.42 Mozilla/5.0+(Macintosh;+U;+PPC+Mac+OS+X;+no-  
no)+AppleWebKit/125.5.6+(KHTML,+like+Gecko)+Safari/125.12  
ASPSESSIONIDSSBDBRAT=BENDCNLCGDEPKPGPPFODJBOP;+Dato=02%2E03%2E2005+10%3A18%3A18 <http://www.brsen.com/>

2005-03-02 23:04:37 /gfx/lesmer.gif 85.164.147.42 Mozilla/5.0+(Macintosh;+U;+PPC+Mac+OS+X;+no-  
no)+AppleWebKit/125.5.6+(KHTML,+like+Gecko)+Safari/125.12  
ASPSESSIONIDSSBDBRAT=BENDCNLCGDEPKPGPPFODJBOP;+Dato=02%2E03%2E2005+10%3A18%3A18 <http://www.brsen.com/>

2005-03-02 23:04:38 /gfx/newsbottom.gif 85.164.147.42 Mozilla/5.0+(Macintosh;+U;+PPC+Mac+OS+X;+no-  
no)+AppleWebKit/125.5.6+(KHTML,+like+Gecko)+Safari/125.12  
ASPSESSIONIDSSBDBRAT=BENDCNLCGDEPKPGPPFODJBOP;+Dato=02%2E03%2E2005+10%3A18%3A18 <http://www.brsen.com/>

2005-03-02 23:04:38 /gfx/pil2.gif 85.164.147.42 Mozilla/5.0+(Macintosh;+U;+PPC+Mac+OS+X;+no-  
no)+AppleWebKit/125.5.6+(KHTML,+like+Gecko)+Safari/125.12  
ASPSESSIONIDSSBDBRAT=BENDCNLCGDEPKPGPPFODJBOP;+Dato=02%2E03%2E2005+10%3A18%3A18 <http://www.brsen.com/>

2005-03-02 23:04:38 /gfx/tabell.jpg 85.164.147.42 Mozilla/5.0+(Macintosh;+U;+PPC+Mac+OS+X;+no-  
no)+AppleWebKit/125.5.6+(KHTML,+like+Gecko)+Safari/125.12  
ASPSESSIONIDSSBDBRAT=BENDCNLCGDEPKPGPPFODJBOP;+Dato=02%2E03%2E2005+10%3A18%3A18 <http://www.brsen.com/>

2005-03-02 23:04:38 /images/nyhetsbilder/mark\_Hughes\_ALVORLIG!.jpg 85.164.147.42 Mozilla/5.0+(Macintosh;+U;+PPC+Mac+OS+X;+no-  
no)+AppleWebKit/125.5.6+(KHTML,+like+Gecko)+Safari/125.12

## Sample Web Log Files 2

165.139.87.3 - - [28/Feb/2007:22:29:39 +0500] "GET /~jba/images/SKB989.jpg HTTP/1.0" 200 8614 "http://www.brain.net.pk/~jba/vollyballs1.html" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; .NET CLR 1.1.4322)"

202.125.143.68 - - [28/Feb/2007:22:29:39 +0500] "GET /newpage/links/2.gif HTTP/1.1" 200 1228 "http://www.brain.net.pk/" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; FDM)"

193.251.135.123 - - [28/Feb/2007:22:29:40 +0500] "GET /~aup/IBMS.h1.jpg HTTP/1.0" 200 1561 "http://www.brain.net.pk/~aup/" "Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.2) Gecko/20070219 Firefox/2.0.0.2"

203.128.11.15 - - [28/Feb/2007:22:29:40 +0500] "GET /newpage/8.gif HTTP/1.1" 200 5126 "http://brain.net.pk/" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; SIMBAR={E4FFD467-FD39-4249-A3BF-AE9EA96514AC})"

202.125.143.68 - - [28/Feb/2007:22:29:40 +0500] "GET /newpage/google.gif HTTP/1.1" 200 137 "http://www.brain.net.pk/" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; FDM)"

202.125.143.68 - - [28/Feb/2007:22:29:40 +0500] "GET /newpage/links/1.gif HTTP/1.1" 200 1993 "http://www.brain.net.pk/" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; FDM)"

202.125.143.68 - - [28/Feb/2007:22:29:40 +0500] "GET /newpage/mp3.gif HTTP/1.1" 200 115 "http://www.brain.net.pk/" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; FDM)"

202.125.143.68 - - [28/Feb/2007:22:29:40 +0500] "GET /newpage/ll.gif HTTP/1.1" 200 2760 "http://www.brain.net.pk/" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; FDM)"

202.125.143.68 - - [28/Feb/2007:22:29:40 +0500] "GET /newpage/web.gif HTTP/1.1" 200 747 "http://www.brain.net.pk/" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; FDM)"

165.139.87.3 - - [28/Feb/2007:22:29:39 +0500] "GET /~jba/images/next.gif HTTP/1.0" 200 4997 "http://www.brain.net.pk/~jba/vollyballs1.html" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; .NET CLR 1.1.4322)"