LOG SETTER

Summer Training Report submitted in partial fulfilment of the requirement for the degree of

B.Tech

In

Computer Science &Engineering

Training Coordinator	By
Mrs. Minakshi Dhamija	**Dev Gaur**

Bhagwan Parshuram Institute of Technology

PSP-4, Sector-17, Rohini, Delhi - 89

August – December 2019

DECLARATION

This is to certify that Report entitled “Log Setter” which is submitted in partial fulfilment of the requirement for the award of degree B.Tech. in Computer Engineering/Information Technology to BPIT, GGSIP University, Dwarka, Delhi comprises only my original work and due acknowledgement has been made in the text to all other material used.

Date: **Dev Gaur**

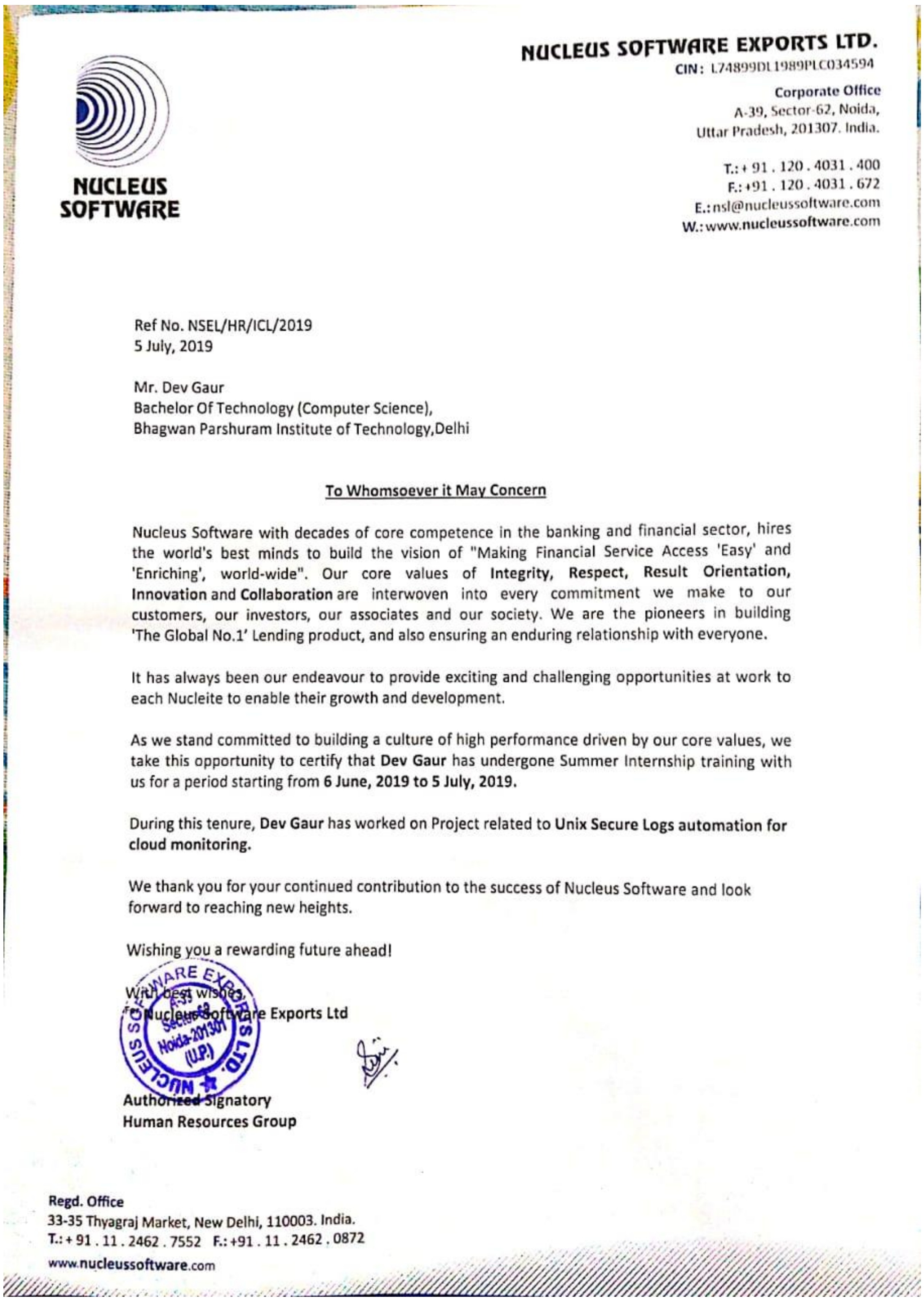
ACKNOWLEDGEMENT

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

I am highly indebted to Mr. Surender Kumar for his guidance and constant supervision as well as for providing necessary information regarding the project & also for his support in completing the project.
I would also like to mention the name of very special person Mr Amit Srivastava for his great knowledge in java and providing me the same.
I would like to express my gratitude towards my parents and my companion for their kind co-operation and encouragement which helped me in completion of this project.
I would like to express my special gratitude and thanks to industry persons for giving me such attention and time.
My thanks and appreciations also go to our coordinator Mrs Minakshi Dhamija for her constant supervision and guidance throughout.

DEV GAUR

COMPANY CERTIFICATE



NUCLEUS SOFTWARE

NUCLEUS SOFTWARE EXPORTS LTD.
CIN: L74899DL1989PLC034594

Corporate Office
A-39, Sector-62, Noida,
Uttar Pradesh, 201307. India.

T.: + 91 . 120 . 4031 . 400
F.: +91 . 120 . 4031 . 672
E.: nsl@nucleussoftware.com
W.: www.nucleussoftware.com

Ref No. NSEL/HR/ICL/2019
5 July, 2019

Mr. Dev Gaur
Bachelor Of Technology (Computer Science),
Bhagwan Parshuram Institute of Technology,Delhi

To Whomsoever it May Concern

Nucleus Software with decades of core competence in the banking and financial sector, hires the world's best minds to build the vision of "Making Financial Service Access 'Easy' and 'Enriching', world-wide". Our core values of **Integrity, Respect, Result Orientation, Innovation** and **Collaboration** are interwoven into every commitment we make to our customers, our investors, our associates and our society. We are the pioneers in building 'The Global No.1' Lending product, and also ensuring an enduring relationship with everyone.

It has always been our endeavour to provide exciting and challenging opportunities at work to each Nucleite to enable their growth and development.

As we stand committed to building a culture of high performance driven by our core values, we take this opportunity to certify that **Dev Gaur** has undergone Summer Internship training with us for a period starting from **6 June, 2019 to 5 July, 2019.**

During this tenure, **Dev Gaur** has worked on Project related to **Unix Secure Logs automation for cloud monitoring.**

We thank you for your continued contribution to the success of Nucleus Software and look forward to reaching new heights.

Wishing you a rewarding future ahead!

With best wishes,
For Nucleus Software Exports Ltd

NUCLEUS SOFTWARE EXPORTS LTD. A-39 Sector-62 Noida-201301 (U.P.)

[illegible]

Authorized Signatory
Human Resources Group

Regd. Office
33-35 Thyagraj Market, New Delhi, 110003. India.
T.: + 91 . 11 . 2462 . 7552 F.: +91 . 11 . 2462 . 0872
www.nucleussoftware.com

TRAINING COORDINATOR CERTIFICATE

This is to certify that Report entitled "……………………." which is submitted by ………………………… in partial fulfilment of the requirement for the award of degree B. Tech in Computer Engineering/ Information Technology to BPIT, GGSIP University, Dwarka, Delhi is a record of the candidate own work and the matter embodied in this report is adhered to the given format.

Date: **Coordinator**

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

As the Internet users have been increased lately. The more data is being transferred daily. When a user logins and performs several activities, all of them are collected somewhere. Just as in Java or any other languages programmers makes a setting file which keeps the track of the software so that in future if the code gets any bugs or errors they just need to study the log file to know what causes the termination of the software. Every company needs a database centre or storage centre. Nowadays companies look forward to buy cloud computing servers to store their data. They need to track their client activities for the security purpose. That is why they need logs. The logs are not automated, they were written manually. The present system is a small scale software that will automatically select the important information and convert it into an excel sheet. This software will save the users time and hard work of writing the logs manually.

Chapter 1
INTRODUCTION

1.1 Objective:

To make a software that would collect required information from bunch of users log coming and store them into excel sheet.

1.2 Description:

To look after every single activity done by client is a very difficult job. Just like in any other programming language software requires logs so that if that software crashes at some point we can track by the help of logs. Similarly, in cloud computing the companies have to keep tracks on the logs. They usually write it in an excel sheet manually. I have made this software to solve this problem and reduce the working time. To make this software I have used only java language which would make it work on any platform. It is an offline application. Regex played very important role in this project and also apache poi for making an excel sheet.

Current software application often produce (or can be configured to produce) some auxiliary text files known as log files. Such files are used during various stages of software development, mainly for debugging and profiling purposes. Use of log files helps testing by making debugging easier. It allows following the logic of the program, at high level, without having to run it in debug mode.

Nowadays, log files are commonly used also at customer's installations for the purpose of permanent software monitoring and/or fine-tuning. Log files became a standard part of large application and are essential in operating systems, computer networks and distributed systems. Log files are often the only way how to identify and locate an error in software, because log file analysis is not affected by any time-based issues known as probe effect. This is an opposite to an analysis of a running program, when the analytical process can interfere with time–critical or resource–critical conditions within the analysed program. Log files are often very large and can have complex structure. Although the process of generating log files is quite simple and straightforward, log file analysis could be a tremendous task that requires enormous computational resources, long time and sophisticated procedures. This often leads to a common situation, when log files are continuously generated and occupy valuable space on storage devices, but nobody uses them and utilizes enclosed information.

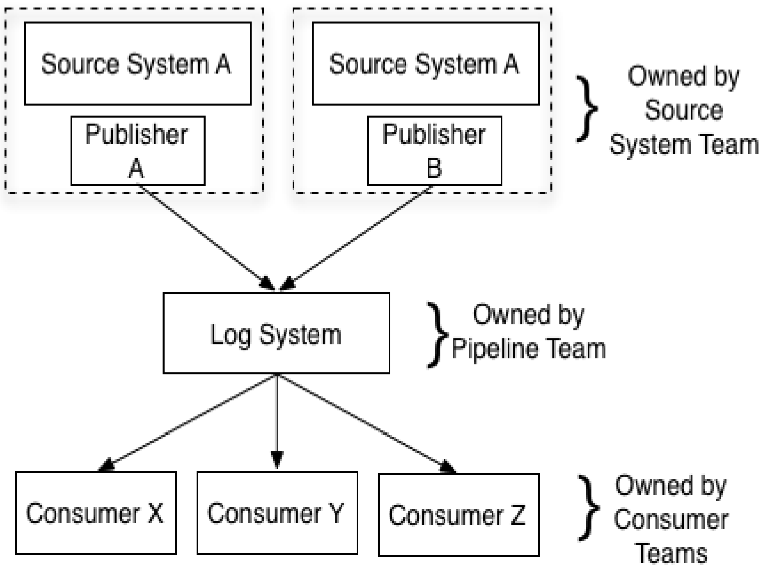


Fig 1.1 Overview of log System

There are various applications (known as log file analysers or log files visualization tools) that can digest a log file of specific vendor or structure and produce easily human readable summary reports. Such tools are undoubtedly useful, but their usage is limited only to log files of certain structure. Although such products have configuration options, they can answer only built-in questions and create built-in reports.[i]

There’s little bit about cloud computing below:-

Cloud computing is an online platform which full fills on-demand availability of computer system, especially data storage. It is an online database which stores user’s data. Nowadays it is in demand as it is easier for small scale companies to rent out the servers of large scale companies like Amazon, Google etc. As servers required bulky amount transaction. Also, lot of services. The goal of cloud computing is to allow users to take benefit from all of these technologies, without the need for deep knowledge about or expertise with each one of them. The cloud aims to cut costs, and helps the users focus on their core business instead of being impeded by IT obstacles.

Apart from this regex played an important role in this project of mine. Now using this software it would be easy to take records of the clients which our performing certain tasks like login-in, changing passwords and many more. I have made this software to work only on few of them. As per my mentor there are around hundreds of functions which a user can perform. I have made it work for few and for rest of them this application would act as a basic building block.

1.3 Existing System:

In the existing system all the work is done manually. The chances of commitment of errors are more and it will take more time to perform any procedure successfully. There are so many limitations in the existing system. So, the existing system should be atomized. If the system is carried over manually, for every procedure it takes more time. So, it is difficult to take immediate decisions. It is difficult to find out where the problem is occurring.

Prior to a more formal definition, let us simply describe log files and their usage. Typically, log files are used by programs in the following way:

- The log file is an auxiliary output file, distinct from other outputs of the program. Almost all log files are plain text files.
- On start-up of the program, the log file is either empty, or contains whatever was left from previous runs of the program.
- During program operation, lines (or groups of lines) are gradually appended to the log file, never deleting or changing any previously stored information.
- Each record (i.e. a line or a group of lines) in a log file is caused by a given event in the program, like user interaction, function call, input or output procedure etc.
- Records in log files are often parameterized, i.e. they show current values of variables, return values of function calls or any other state information
- The information reported in log files is the information that programmers consider important or useful for program monitoring and/or locating faults.

Syntax and format of log files can vary a lot. There are very brief log files that contain just sets of numbers and there are log files containing whole essays. Nevertheless, an average log file is a compromise of these two approaches; it contains minimum unnecessary information while it is still easily human-readable. Such files for example contain both variable names both variable values, comprehensive delimiters, smart text formatting, hint keywords, comments etc. Records in log files are often provided by time stamps. Although it is not a strict rule, log files without timestamps are rarely seen, because time-less events are of less valuable information.

Disadvantages of the Existing System

- Existing system is not user friendly
- System is not well organized and precise
- It is time consuming
- Information is redundant and inconsistent.
- •It didn't integrate all the modules hence, decision making is difficult

1.4 Proposed System:

Need for Automation

The existing system consists of many faults in which one the major fault was the human made errors. Writing down every single log is a very hectic task. To overcome this the new system that is automated system exists (log setter).

Features of Proposed system

• Flexibility:-

The tool must be usable for various analytical tasks on any log file. Flexibility concerns log file format, syntax, semantics and types of queries (questions, statistics and other analytical tasks) that can be performed (see page 10). The main point here is that exact types of queries are not known during the design time of a tool, so the tool must accept new queries at run time. Therefore the tool should offer a way (a programmable interface) to allow users to specify their own assessments of analysis and write corresponding programs or scripts. The tool must cope well with either small or huge log files, either simple or very complex log files, either statistical or analytical queries[ii]

. •Open Design, Extensibility:-

The tool must be open for later third-parties extensions. There must exist a way (may be an API) how to access internal data structures. This would probably results in a multi–layer design of the tool, with a core layer and many plug-in modules in higher layers. 27 Open designs also means proper selection of involved technologies, i.e. open, standardized formats like XML, ULM, HTML etc. are preferred instead of any proprietary or ad-hoc solutions.[iii]

• Modularity:-

- The tool should be modular. A modular design allows easy extension of tool capabilities via additional packages for specific tasks. For example a package for log analysis of concrete piece of software can contain dozens of modules for various log cleaning and filtration, log transformation, queries processing and visualization. The user shall activate or deactivate appropriate modules in the analysis process according his or her wishes. The plug-in modules can be divided into disjunctive classes according their function:
- Input filters that perform data cleaning, filtering and transformation into internal data structures (or DBMS)
- (ii)Data analysers that process queries, one type/kind of query per one module
- Reporters that present results of log file analysis in a form of text reports, lists, tables or in some visual way

• Easy Application:-

The tool must be easy to use. It means a well-designed user interface and a good language for data and query description. For example, spread sheets or AWK are powerful enough even for non-trivial analytical tasks but they are not easy to use mainly because of lack of structured data types, abstraction, variable handling and overall approach. The tool must work at high level of abstraction (to be universal) but on the other hand, users must be able to express their ideas in a simple and short way. For instance, the tool can utilize regular expressions for easy log cleaning, filtration and transformation. The user-written modules should access log data via a structured way. An object model (like in dynamic HTML) seems to satisfy well this requirement. The object model allows both native and synthetic (computed) attributes and easy data manipulation. The tool should support automatic recognition of data types (number, string, date, time, IP address) and allow multiple formats where applicable (i.e. the 28 time). The tool should also offer handy operators for DNS translation, time calculation, statistical tasks, precedence, causality and similar.[iii]

• Speed:-

The speed issue is very important in case of working with large log files. The duration of log analysis depends on log size, CPU speed, free memory; number and complexity of queries and possibly on network/hard drive speed. Some log files can grow at amazing speed. If the tool must store hundreds of new log entries per minute, the consumption of system resources at “idle” must be also considered. If the tool should be used for continuous, near real-time monitoring/analysis, the speed becomes the most critical criterion.

• Robustness:-

The tool is expected to work with log files of hundreds of megabytes. Some log entries can be damaged of missing due to errors in software, hardware errors, network down time and so on. Metadata information may not match log file and there can be also errors in user-written modules. All these aspects can cause fatal miss-function or crash of the tool. Therefore the following issues should be incorporated into the design.

- all log entries that do not conform supplied metadata information must be filtered out
- the metadata information must be validated for correct syntax
- the user-written programs must be “executed” in a safe environment. The tool must handle or monitor allocation of memory, I/O operations and perhaps somehow limit the time of module execution (in case of infinite loops etc.)
- all user-supplied information must be checked for correctness
- tool design should avoid any inefficiency and unnecessary complexity

These are only the basic ideas; robustness is the art of proper software design and careful implementation.[iii]

• Language of Analysis:-

A language for description of the analytical task (i.e. the language of user-written modules) is a subject of this research. It should be definitely a programming language to be enough powerful and to allow all necessary flexibility. The language should work at high level of abstraction; it must certainly offer some descriptive and declarative features. Concerning the programming style, a data–driven or imperative approach seems to best choices. If possible, the language should allow usage in connection with both compilers and interpreters. Further, the language should be simple (to allow easy implementation) but enough flexible and powerful, probably with object oriented data model with weak type checking. Easy mapping (translation) into lower–level commands of the corresponding engine (for example OLAP techniques or SQL commands) is also an important feature.

• The Analytical Engine:-

This paragraph concerns the key operation of the tool, but because the engine is more or less hidden from the user, there are no special requirements or user expectations. The only user-visible part of the analytical engine is the task specification, i.e. the language of analysis. There are three possible operational modes of an engine:

- data–driven, single–pass approach (like AWK[24]); only a portion (possibly one record) of the analysed log must be in the memory at the same time
- multiple–pass (or single–pass with partial rewinds allowed); a portion of the analysed log must be in the memory at the same time but not the whole log
- Database–oriented approach, whole log is stored in a database; memory usage depends on the DBMS used From a different point of view, the tool can process batch–jobs or perform continuous operation. In the second case, the tool accepts a stream of incoming log entries and updates the database. The tool can also regularly expire oldest log entries; this way of operation is suitable for system monitoring, when only recent data (one day, week, month) are relevant for analysis.

• Visualization and Data Exports:-

Presentation of results of analysis in the form of tables, histograms, charts or other easily comprehensive way is the goal of all effort. On the other way, the requirements put on visualization can considerably vary with different analytical task. In addition, including visualization engine directly into the engine would make it too complicated. Thus the suggestion here is to separate visualization and results reports into separate, user-written plug-in modules and provide necessary API. It could contain primitive functions to print desired data in a list, a table, a histogram, a pie chart etc. that would be processed by some generic visualization module.

• User Interface User interface should be possibly platform independent and should allow the following tasks:

- selection of log source (local files or network socket)
- selection of metadata information if applicable (from a text file)

- selection/activation of all modules that should participate on the analysis
- display results of analysis or handle output files

Opposite to the analytical engine, the user interface requires only few system resources. A Java application or web–based interface seems to be a good choice.

Chapter 2

System Requirements and Specification

2.1 Introduction

In existing system , companies do the work manually which is time consuming and also takes a lot of efforts. This system may cause many errors which are human made errors.

2.1.1 *Purpose*

Using this software we can easily convert the logs into the excel sheet. Using this we can save time, money and physical strength .We are also able to get quick excel sheet and for choosing the important part we need to put in the excel.

2.1.2 *Scope*

To overcome the limitation of an existing system of writing the logs manually thus this new system has been made. This software also includes the upload option which would enable the user to select the text file which he/she wants to convert into the excel sheet. Also, besides this an user can convert the file into any format in which he/she is comfortable with.

2.2 Software System Attributes

To achieve good performance the following requirements must be satisfied:

- Scalability: The ease with which a system or component can be modified to fit the problem area.
- Portability: The ease with which a system or component can be transferred from one hardware or software environment to another.
- Security: It is the ideal state where all information can be communicated across the internet / company secure from unauthorized persons being able to read it and/or manipulate it. It is also the process of preventing and detecting unauthorized use of one’s computer.

- Maintainability: The ease with which a software system or component can be modified to correct faults, improve performance, or other attributes, or adapt to a changed environment.

- Reliability: The ability of a system or component to perform its required functions under stated conditions for a specified period of time.

- Reusability: The degree to which a software module or other work product can be used in more than one computing program or software system.
- Flexibility: The tool must be usable for various analytical tasks on any log file. Flexibility concerns log file format, syntax, semantics and types of queries (questions, statistics and other analytical tasks) that can be performed. The main point here is that exact types of queries are not known during the design time of a tool, so the tool must accept new queries at run time. Therefore the tool should offer a way (a programmable interface) to allow users to specify their own assessments of analysis and write corresponding programs or scripts.
- Easy Application: The tool must be easy to use. It means a well-designed user interface and a good language for data and query description. For example, spread sheets or AWK are powerful enough even for non-trivial analytical tasks but they are not easy to use mainly because of lack of structured data types, abstraction, variable handling and overall approach. The tool must work at high level of abstraction (to be universal) but on the other hand, users must be able to express their ideas in a simple and short way. For instance, the tool can utilize regular expressions for easy log cleaning, filtration and transformation.
- Speed: The speed issue is very important in case of working with large log files. The duration of log analysis depends on log size, CPU speed, free memory, number and complexity of queries and possibly on network/hard drive speed. Some log files can grow at amazing speed. If the tool must store hundreds of new log entries per minute, the consumption of system resources at “idle” must be also considered. If the tool should be used for continuous, near real-time monitoring/analysis, the speed becomes the most critical criterion.

Chapter 3
System Design (Diagrams)

3.1 Use Case Diagram

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.

Why Make Use Case Diagrams?

Use case diagrams are valuable for visualizing the functional requirements of a system that will translate into design choices and development priorities.

They also help identify any internal or external factors that may influence the system and should be taken into consideration.

They provide a good high-level analysis from outside the system. Use case diagrams specify how the system interacts with actors without worrying about the details of how that functionality is implemented.

A use case diagram contains four components.

- The boundary, which defines the system of interest in relation to the world around it.
- The actors, usually individuals involved with the system defined according to their roles.
- The use cases, which the specific roles are played by the actors within and around the system.
- The relationships between and among the actors and the use cases.

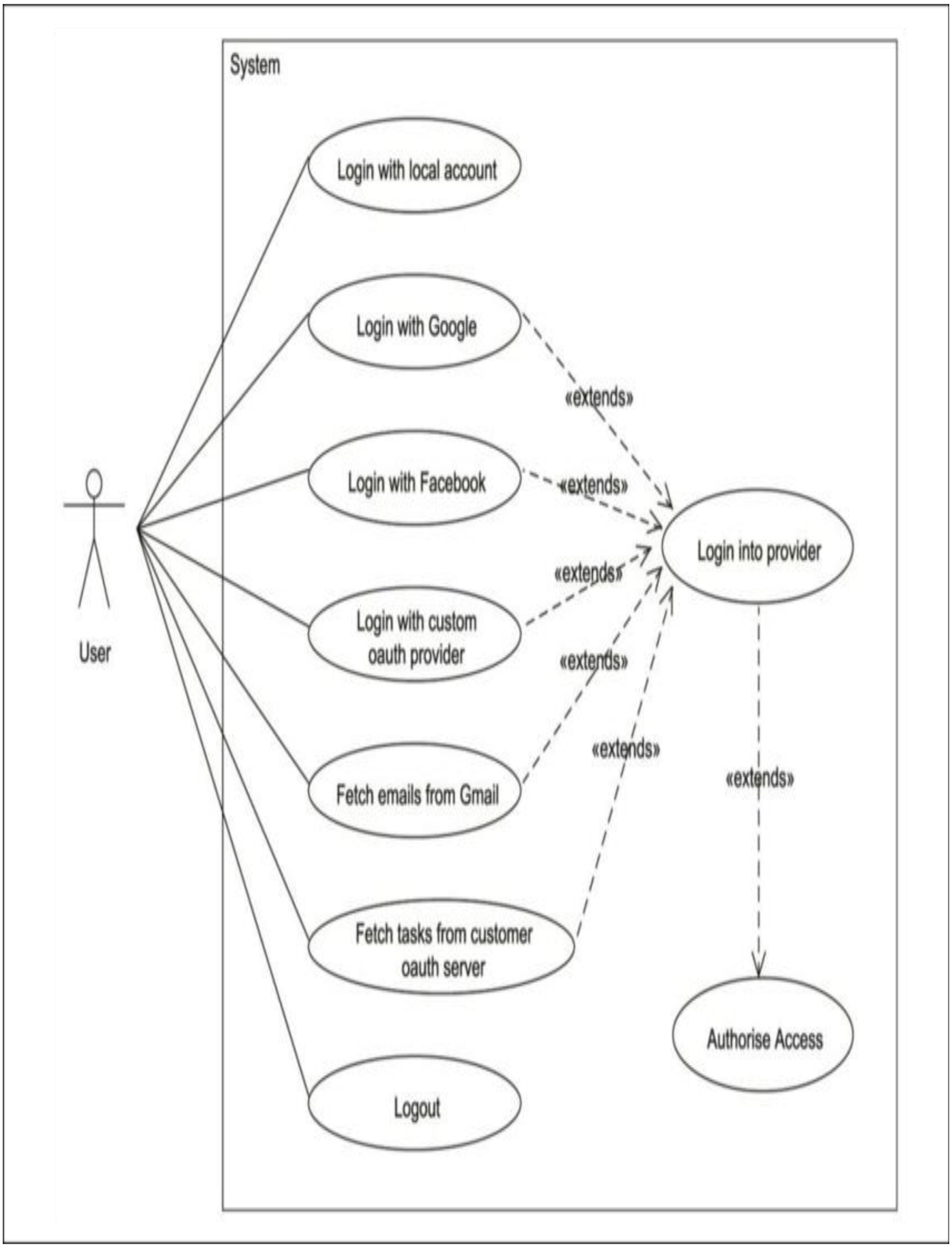


Fig 3.1 Use case diagram of log(Seen on online application)

3.2 Data Flow Diagram (DFD)

Data Flow Diagrams are used to represent the flow of data from one part of the system to another part of the system. Mainly these DFDs are used to represent the existing system. We can divide the study of DFDs into 3 parts.

- Notations
- Rules
- Levels

Notations

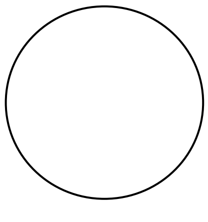
We use four notations in Data Flow Diagrams. They are as follows.

- External Entity

It is a thing which is external to the system and uses the system, these are graphically represented using rectangle.

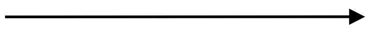
- Process

Process is an action taking place in the system. Process is represented using a circle.



- Data Flow

Data Flow indicates the direction of the flow of data from one part of the system to another part of the system. Data Flow is represented using a one directional arrow.



- Data Store

Data Stores are used to represent the data bases in the system. Data Stores are graphically represented using double lines.

Rules

While drawing Data Flow Diagrams we have follow some rules. The following should not be violated.

- There should not be any data flow between two external entities directly.
- There should not be any data flow between two data stores directly.
- There should not be any data flow between data store and external entity.

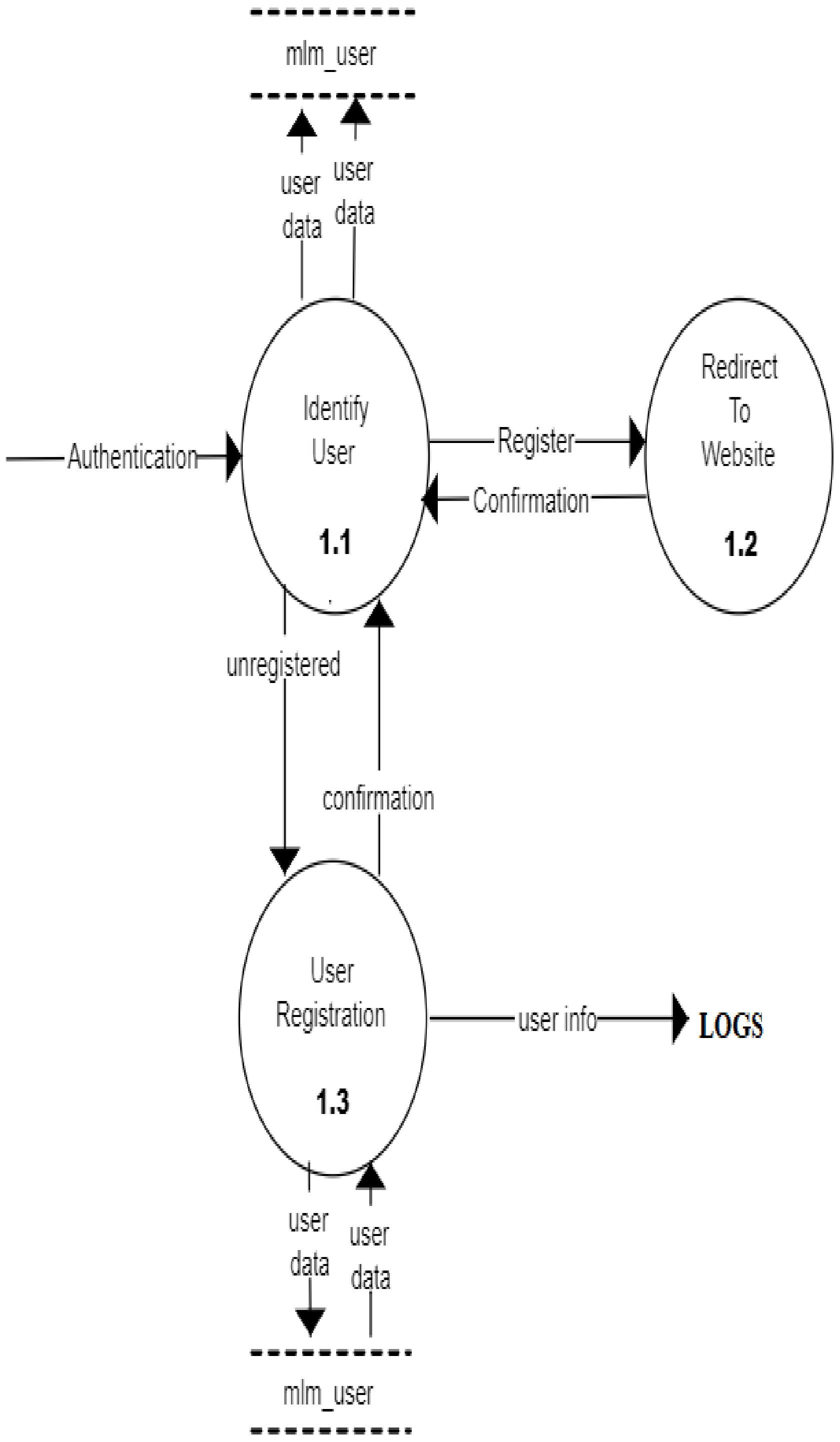


Fig 3.2 Data Flow Diagram of log system

3.3 Entity-Relationship Diagram (E-R)

ER-modelling is a data modelling method used in software engineering to produce a conceptual data model of an information system. Diagrams created using this ER-modelling method are called Entity-Relationship Diagrams or ER diagrams or ERDs.

Purpose of ERD

- The database analyst gains a better understanding of the data to be contained in the database through the step of constructing the ERD.
- The ERD serves as a documentation tool.
- Finally, the ERD is used to connect the logical structure of the database to users. In particular, the ERD effectively communicates the logic of the database to users

Components of an ER Diagrams

- Entity

An entity can be a real-world object, either animate or inanimate, that can be merely identifiable. An entity is denoted as a rectangle in an ER diagram. For example, in a school database, students, teachers, classes, and courses offered can be treated as entities. All these entities have some attributes or properties that give them their identity.

- Attributes

Entities are denoted utilizing their properties, known as attributes. All attributes have values. For example, a student entity may have name, class, and age as attributes.

There exists a domain or range of values that can be assigned to attributes. For example, a student's name cannot be a numeric value. It has to be alphabetic. A student's age cannot be negative, etc.

- Relationships

The association among entities is known as relationship. Relationships are represented by the diamond-shaped box. For example, an employee works at a department, a student enroll in a course. Here, Works at and Enroll are called relationships.

The number of participating entities in a relationship describes the degree of the relationship. The three most common relationships in E-R models are:

Unary (degree1)

Binary (degree2)

Ternary (degree3)

Unary relationship**:** This is also called recursive relationships. It is a relationship between the instances of one entity type. For example, one person is married to only one person.

Binary relationship: It is a relationship between the instances of two entity types. For example, the Teacher teaches the subject.

Ternary relationship: It is a relationship amongst instances of three entity types. In fig, the relationships "may have" provide the association of three entities, i.e., TEACHER, STUDENT, and SUBJECT. All three entities are many-to-many participants. There may be one or many participants in a ternary relationship.

In general, "n" entities can be related by the same relationship and is known as n-ary relationship.

- Cardinality

Cardinality describes the number of entities in one entity set, which can be associated with the number of entities of other sets via relationship set.

Types of Cardinalities

One to One: One entity from entity set A can be contained with at most one entity of entity set B and vice versa. Let us assume that each student has only one student ID, and each student ID is assigned to only one person. So, the relationship will be one to one.

One to many: When a single instance of an entity is associated with more than one instances of another entity then it is called one to many relationships. For example, a client can place many orders; order cannot be placed by many customers.

Many to One: More than one entity from entity set A can be associated with at most one entity of entity set B, however an entity from entity set B can be associated with more than one entity from entity set A. For example - many students can study in a single college, but a student cannot study in many colleges at the same time.

Many to Many: One entity from A can be associated with more than one entity from B and vice-versa. For example, the student can be assigned to many projects, and a project can be assigned to many students.

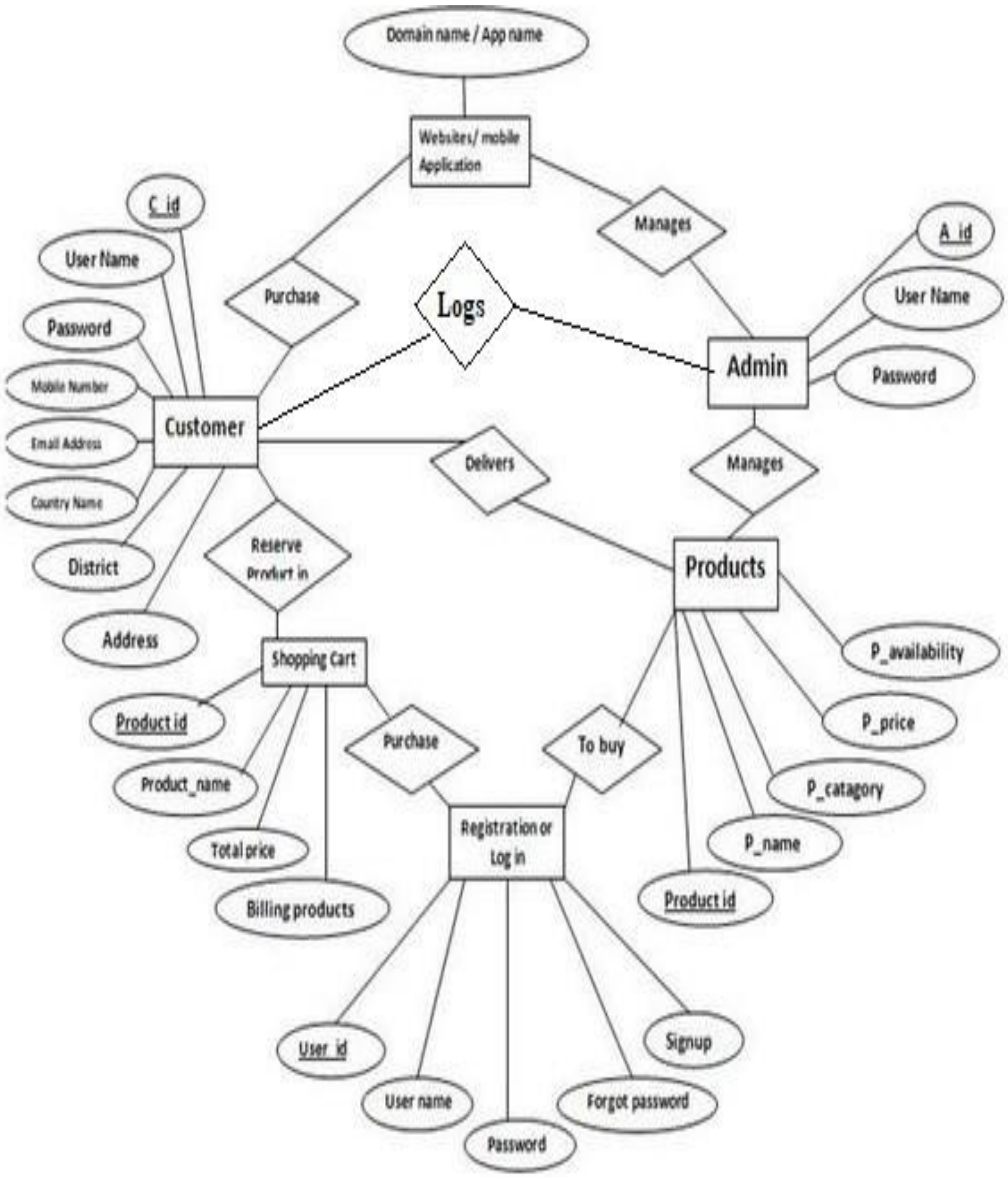


Fig 3.3 ER-Diagram of log

Chapter 4

Process Selection

4.1 <u>INTRODUCTION TO JAVA:</u>

WHAT IS JAVA?

Java is a general-purpose computer **programming language** that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers **"write once, run anywhere" (WORA)**, meaning that compiled Java code can run on all platforms that support Java without the need for recompilation.

For example, you can write and compile a Java program on UNIX and run it on Microsoft Windows, Macintosh, or UNIX machine without any modifications to the source code. Is achieved by compiling a Java program into an intermediate language called byte **code**. The format of byte code is *platform-independent*. A virtual machine, called the Java Virtual Machine (JVM), is used to run the byte code on each platform.

Fig 4.1 Java Logo

The Oracle implementation is packaged into two different distributions:

1. **Java Runtime Environment (JRE)** which contains the parts of the Java SE platform required to run Java programs and is intended for end users.
2. **Java Development Kit (JDK)** which is intended for software developers and includes development tools such as the Java compiler, Javadoc, Jar, and a debugger.

Why to Learn java Programming?

Java is a MUST for students and working professionals to become a great Software Engineer especially when they are working in Software Development Domain. I will list down some of the key advantages of learning Java Programming:

- **Object Oriented** − In Java, everything is an Object. Java can be easily extended since it is based on the Object model.
- **Platform Independent** − unlike many other programming languages including C and C++, when Java is compiled, it is not compiled into platform specific machine, rather into platform independent byte code. This byte code is distributed over the web and interpreted by the Virtual Machine (JVM) on whichever platform it is being run on.
- **Simple** − Java is designed to be easy to learn. If you understand the basic concept of OOP Java, it would be easy to master.
- **Secure** − With Java's secure feature it enables to develop virus-free, tamper-free systems. Authentication techniques are based on public-key encryption.
- **Architecture-neutral** − Java compiler generates an architecture-neutral object file format, which makes the compiled code executable on many processors, with the presence of Java runtime system.
- **Portable** − Being architecture-neutral and having no implementation dependent aspects of the specification makes Java portable. Compiler in Java is written in ANSI C with a clean portability boundary, which is a POSIX subset.

- **Robust** − Java makes an effort to eliminate error prone situations by emphasizing mainly on compile time error checking and runtime checking.

Applications of Java Programming

The latest release of the Java Standard Edition is Java SE 8. With the advancement of Java and its widespread popularity, multiple configurations were built to suit various types of platforms. For example: J2EE for Enterprise Applications, J2ME for Mobile Applications.

The new J2 versions were renamed as Java SE, Java EE, and Java ME respectively. Java is guaranteed to be **Write Once, Run Anywhere.**

- **Multithreaded** − With Java's multithreaded feature it is possible to write programs that can perform many tasks simultaneously. This design feature allows the developers to construct interactive applications that can run smoothly.
- **Interpreted** − Java byte code is translated on the fly to native machine instructions and is not stored anywhere. The development process is more rapid and analytical since the linking is an incremental and light-weight process.
- **High Performance** − with the use of Just-In-Time compilers, Java enables high performance.
- **Distributed** − Java is designed for the distributed environment of the internet.
- **Dynamic** − Java is considered to be more dynamic than C or C++ since it is designed to adapt to an evolving environment. Java programs can carry extensive amount of run-time information that can be used to verify and resolve accesses to objects on run-time.

The JFrame Class

Whenever you create a graphical user interface with Java Swing functionality, you will need a container for your application. In the case of Swing, this container is called a **JFrame**. All GUI applications require a JFrame. In fact some Applets even use a JFrame. Why?

You can't build a house without a foundation. The same is true in Java: Without a container in which to put all other elements, you won't have a GUI application. In other words, the JFrame is required as the foundation or base container for all other graphical components.

Java Swing applications can be run on any system that supports Java. These applications are lightweight. This means that doesn’t take up much space or use many system resources.

JFrame is a class in Java and has its own methods and constructors. **Methods** are functions that impact the JFrame, such as setting the size or visibility. **Constructors** are run when the instance is created: One constructor can create a blank JFrame, while another can create it with a default title.

4.2 Logstash Introduction:-

Logstash is an open source data collection engine with real-time pipelining capabilities. Logstash can dynamically unify data from disparate sources and normalize the data into destinations of your choice. Cleanse and democratize all your data for diverse advanced downstream analytics and visualization use cases.

While Logstash originally drove innovation in log collection, its capabilities extend well beyond that use case. Any type of event can be enriched and transformed with a broad array of input, filter, and output plugins, with many native codecs further simplifying the ingestion process. Logstash accelerates your insights by harnessing a greater volume and variety of data.

The Power of Logstash

The ingestion workhorse for Elasticsearch and more

Horizontally scalable data processing pipeline with strong Elasticsearch and Kibana synergy

Pluggable pipeline architecture

Mix, match, and orchestrate different inputs, filters, and outputs to play in pipeline harmony

Community-extensible and developer-friendly plugin ecosystem

Over 200 plugins available, plus the flexibility of creating and contributing your own

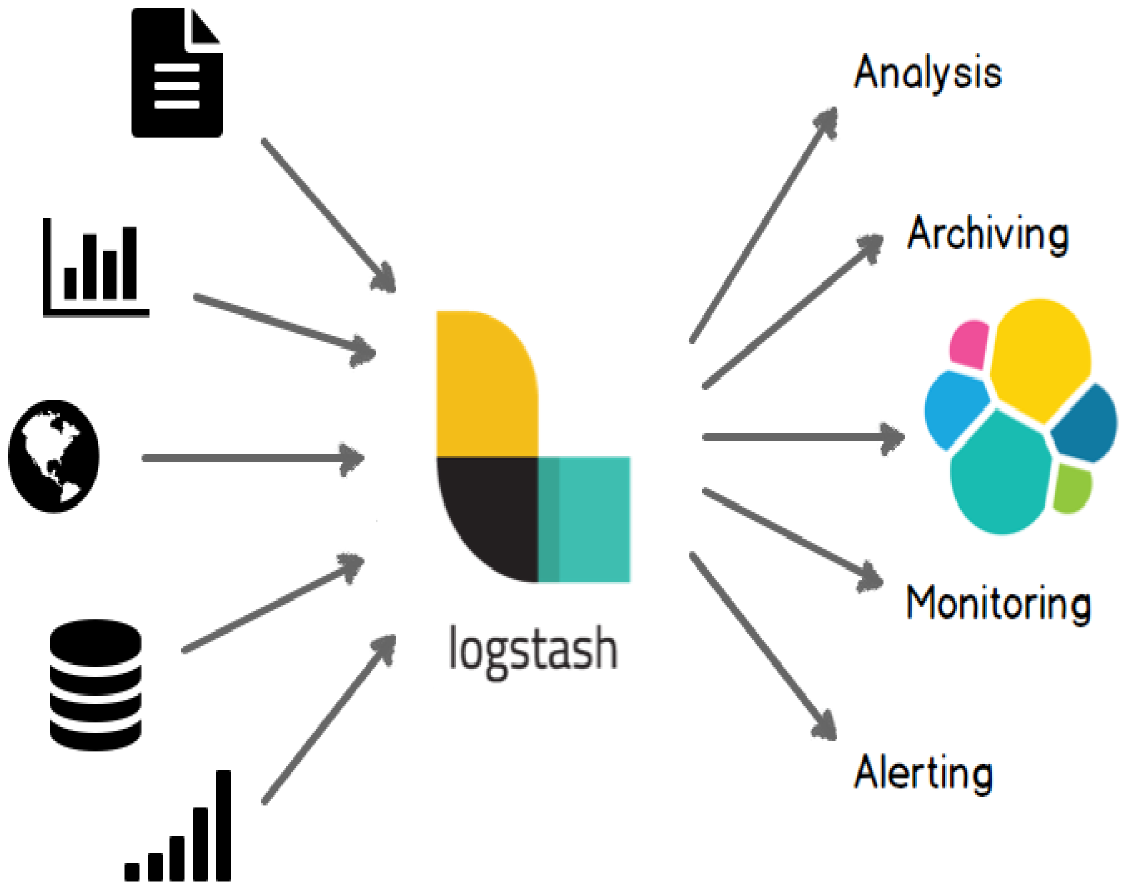


Fig 4.2 Overview of Logstash

4.3Platform (Eclipse):-

In the context of computing, Eclipse is an integrated development environment (IDE) for developing applications using the Java programming language and other programming languages such as C/C++, Python, PERL, Ruby etc.

The Eclipse platform which provides the foundation for the Eclipse IDE is composed of plug-ins and is designed to be extensible using additional plug-ins. Developed using Java, the Eclipse platform can be used to develop rich client applications, integrated development environments and other tools. Eclipse can be used as an IDE for any programming language for which a plug-in is available.

The Java Development Tools (JDT) project provides a plug-in that allows Eclipse to be used as a Java IDE, PyDev is a plugin that allows Eclipse to be used as a Python IDE, C/C++ Development Tools (CDT) is a plug-in that allows Eclipse to be used for developing application using C/C++, the Eclipse Scala plug-in allows Eclipse to be used an IDE to develop Scala applications and PHPeclipse is a plug-in to eclipse that provides complete development tool for PHP.

eclipse

Fig 4.3 Eclipse Logo

Parts of an Eclipse Window

The major visible parts of an eclipse window are −

- Views
- Editors (all appear in one editor area)

- Menu Bar
- Toolbar

An eclipse perspective is the name given to an initial collection and arrangement of views and an editor area. The default perspective is called java. An eclipse window can have multiple perspectives open in it but only one perspective can be active at any point of time. A user can switch between open perspectives or open a new perspective. A perspective controls what appears in some menus and tool bars.

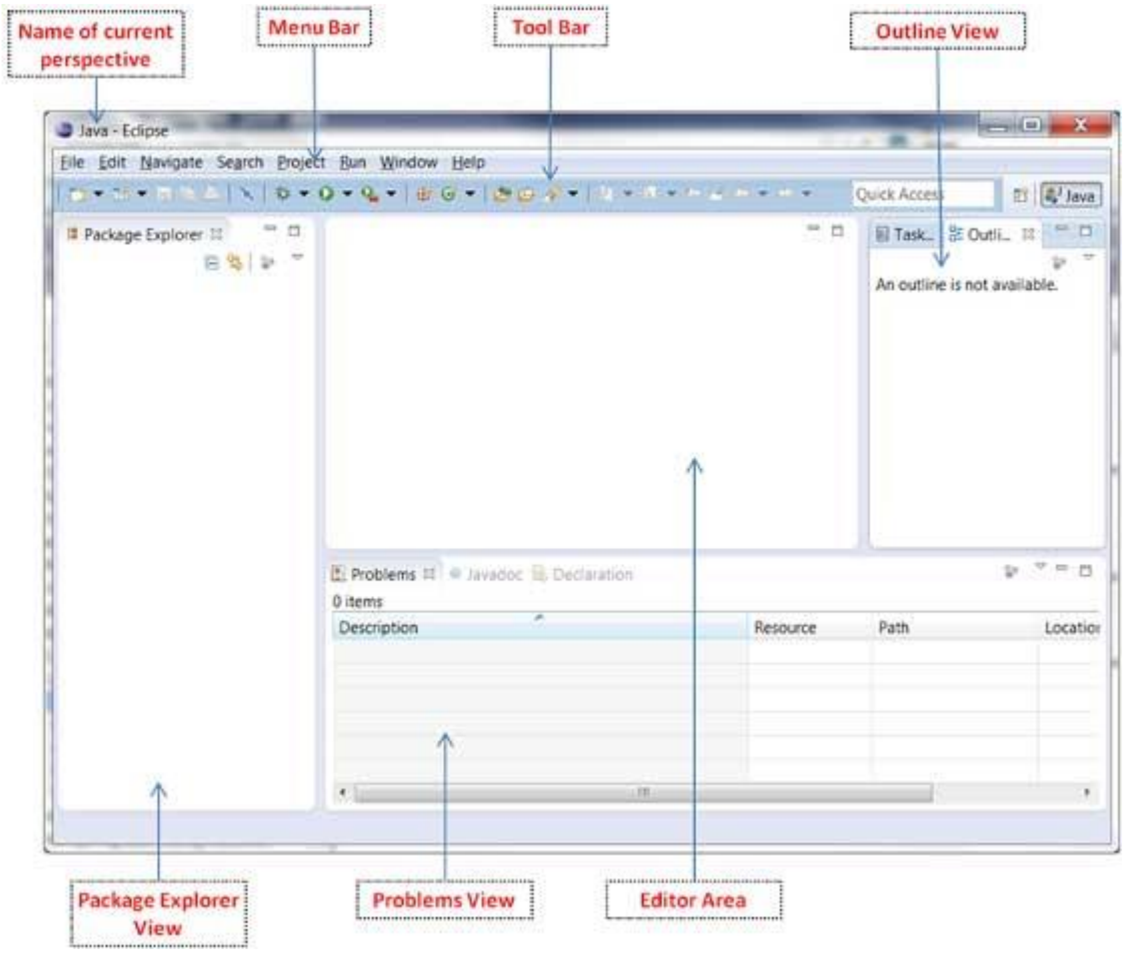


Fig 4.4 Eclipse Interface

A perspective has only one editor area in which multiple editors can be open. The editor area is usually surrounded by multiple views. In general, editors are used to edit the project data and views are used to view the project metadata. For example the package explorer shows the java files in the project and the java editor is used to edit a java file.

The eclipse window can contain multiple editors and views but only one of them is active at any given point of time. The title bar of the active editor or view looks different from all the others.

Basic working:-

In today’s world millions of internet users login and performs several activities daily. The companies rent out the cloud servers of the large companies one of the major provider is AWS (Amazon Web Service). To keep track of user’s activity for security purposes this purposed system will be used.

It is written on java programming language. Consist Four main classes which are:-

- Design
- Excel
- New Pattern
- Process

Design class mainly consists of the GUI part of this project. Java Swing is used to provide the graphics user interface and a simple one. The start button has a function which is used to call convertor function which converts the text file into the excel file. In excel class it first converts the file log file into the useful file and then extracts the main important information and puts them into the cells of the excel sheet which would result in the required excel sheet. There’s a pattern which a user can define.

New Pattern class has a pattern define which can be further changed as per the user’s requirements. That is which part of the log file has to stay in the excel sheet for the future references.

Process class basically is a test file in which we can test the new pattern class as the new pattern file creates a text file. So whenever we define we can test it using the process file by creating its object and calling its function.

Excel is the main file which converts the text file into the required excel sheet.

GRAPHICS USER INTERFACE USING JAVA:-

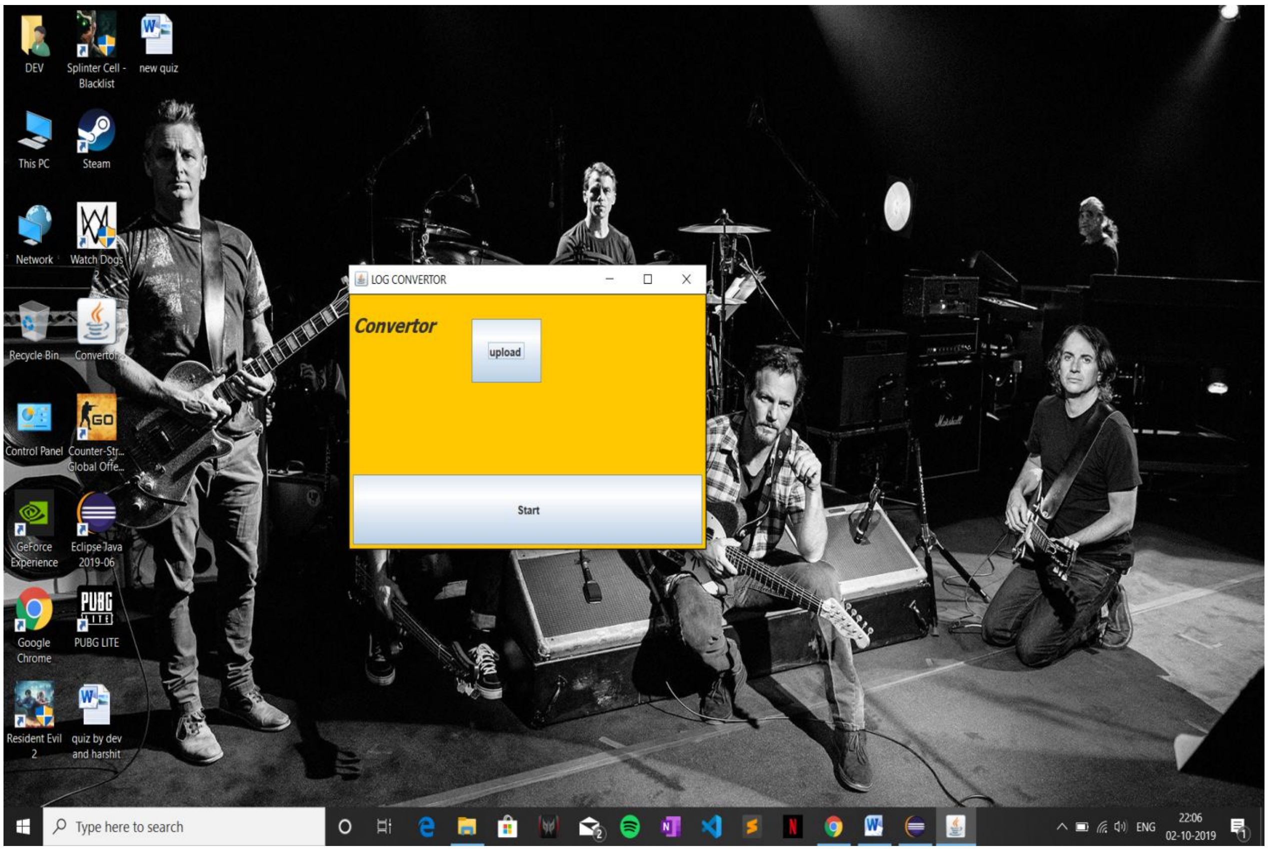
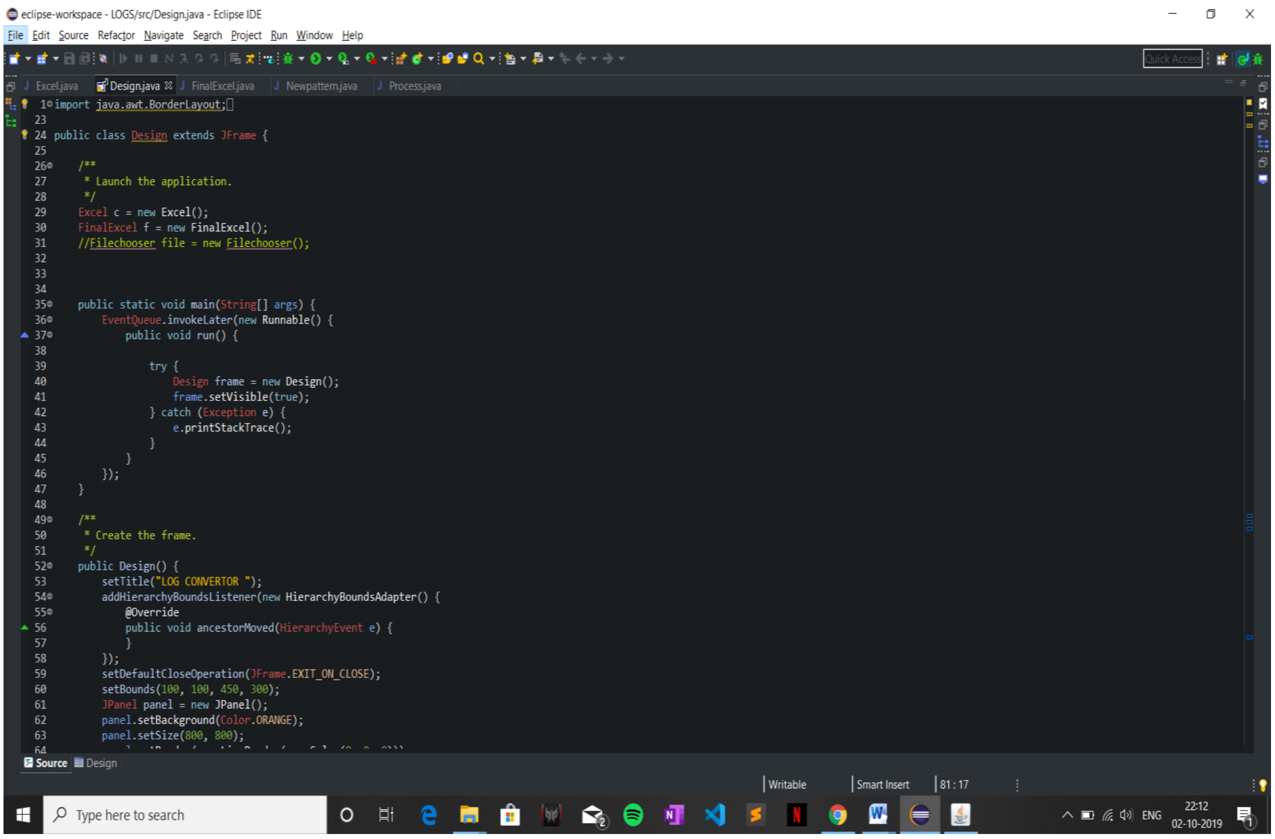


Fig 4.5 Log Setter’s Interface

A graphical user interface (GUI) is an interface through which a user interacts with electronic devices such as computers, hand-held devices and other appliances. This interface uses icons, menus and other visual indicator (graphics) representations to display information and related user controls, unlike text-based interfaces, where data and commands are in text. GUIl representations are manipulated by a pointing device such as a mouse, trackball, stylus, or a finger on a touch screen.

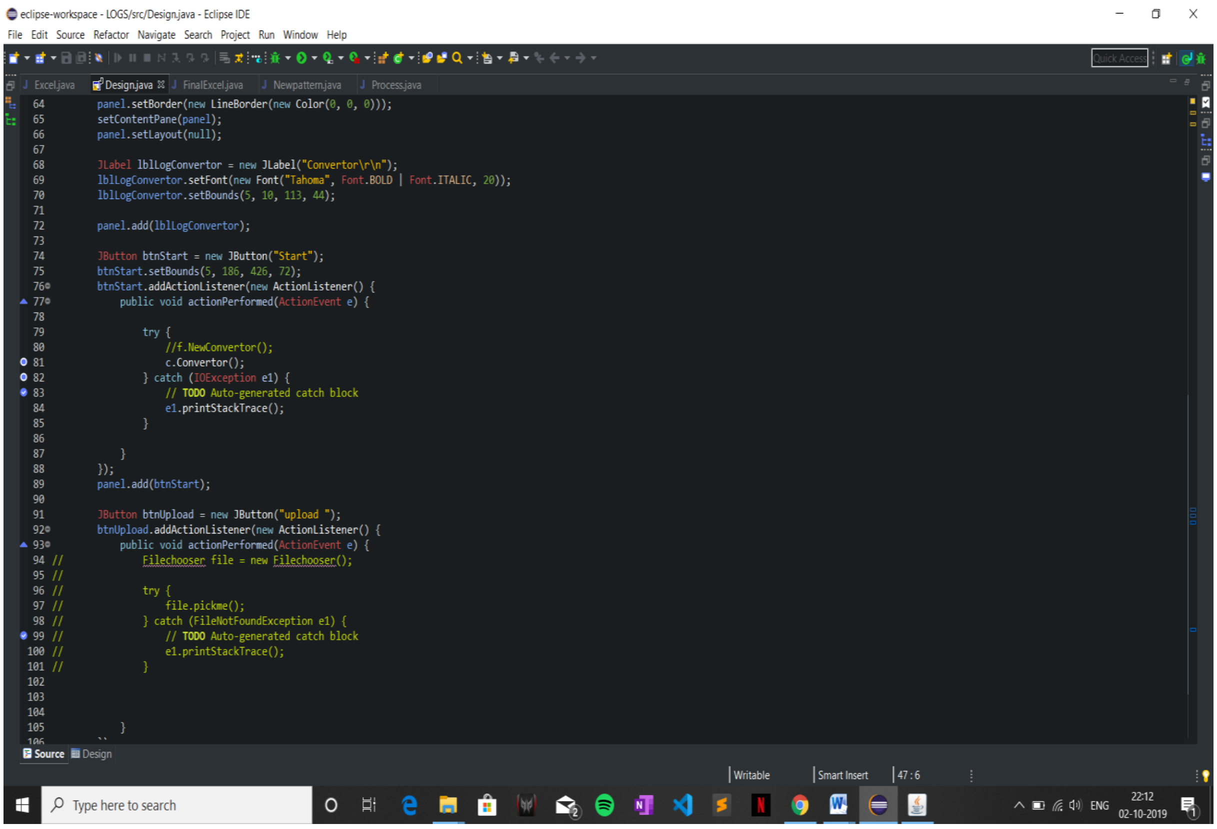
The need for GUI became apparent because the first human/computer text interface was through keyboard text creation by what is called a prompt (or DOS prompt). Commands were typed on a keyboard at the DOS prompt to initiate responses from a computer. The use of these commands and the need for exact spelling created a cumbersome and inefficient interface.

<u>CODE OF GUI:-</u>


eclipse-workspace - LOGS/src/Design.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Excel.java Design.java FinalExcel.java Newpattern.java Process.java
Source Design
Writable Smart Insert 81 : 17
Type here to search
ENG 22:12 02-10-2019

```java
import java.awt.BorderLayout;

public class Design extends JFrame {

    /**
     * Launch the application.
     */
    Excel c = new Excel();
    FinalExcel f = new FinalExcel();
    //Filechooser file = new Filechooser();



    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {

                try {
                    Design frame = new Design();
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    /**
     * Create the frame.
     */
    public Design() {
        setTitle("LOG CONVERTOR ");
        addHierarchyBoundsListener(new HierarchyBoundsAdapter() {
            @Override
            public void ancestorMoved(HierarchyEvent e) {
            }
        });
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 300);
        JPanel panel = new JPanel();
        panel.setBackground(Color.ORANGE);
        panel.setSize(800, 800);
```

Fig 4.6 Code of GUI of Log Setter



```java
        panel.setBorder(new LineBorder(new Color(0, 0, 0)));
        setContentPane(panel);
        panel.setLayout(null);

        JLabel lblLogConvertor = new JLabel("Convertor\r\n");
        lblLogConvertor.setFont(new Font("Tahoma", Font.BOLD | Font.ITALIC, 20));
        lblLogConvertor.setBounds(5, 10, 113, 44);

        panel.add(lblLogConvertor);

        JButton btnStart = new JButton("Start");
        btnStart.setBounds(5, 186, 426, 72);
        btnStart.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {

                try {
                    //f.NewConvertor();
                    c.Convertor();
                } catch (IOException e1) {
                    // TODO Auto-generated catch block
                    e1.printStackTrace();
                }

            }
        });
        panel.add(btnStart);

        JButton btnUpload = new JButton("upload ");
        btnUpload.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
//              Filechooser file = new Filechooser();
//
//              try {
//                  file.pickme();
//              } catch (FileNotFoundException e1) {
//                  // TODO Auto-generated catch block
//                  e1.printStackTrace();
//              }



            }
```

Fig 4.7 Code of GUI of Log Setter Continued

Java Swing is a part of Java Foundation Classes (JFC) which was designed for enabling large-scale enterprise development of Java applications. Java Swing is a set of APIs that provides graphical user interface (GUI) for Java programs. Java Swing is also known as Java GUI widget toolkit.

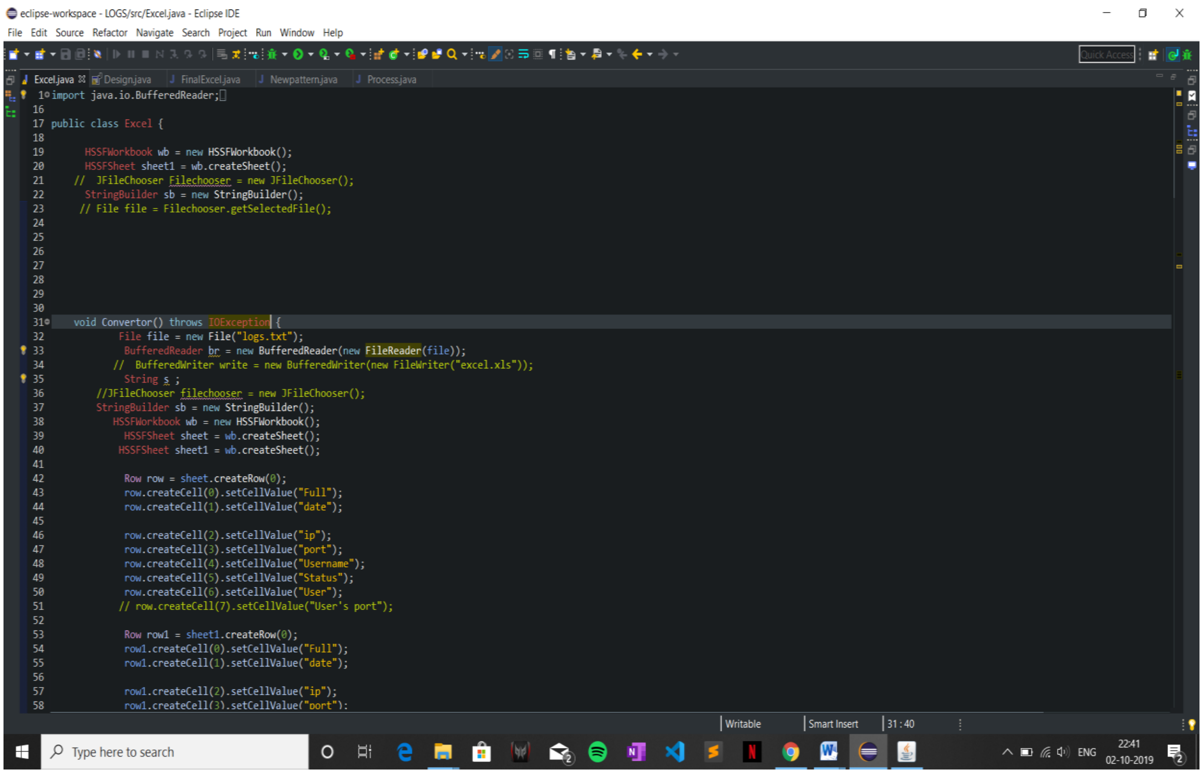
Java Swing or Swing was developed based on earlier APIs called Abstract Windows Toolkit (AWT). Swing provides richer and more sophisticated GUI components than AWT. The GUI components are ranging from a simple label to complex tree and table. Besides emulating look and feel of various platforms, Swing also provides *the pluggable look and feel* to allow look and feel of Java programs independent from the underlying platform.

Swing Architecture

Swing is platform independent and enhanced MVC (Model –View – Controller) framework for Java application. Here are the most important features in Swing architecture.

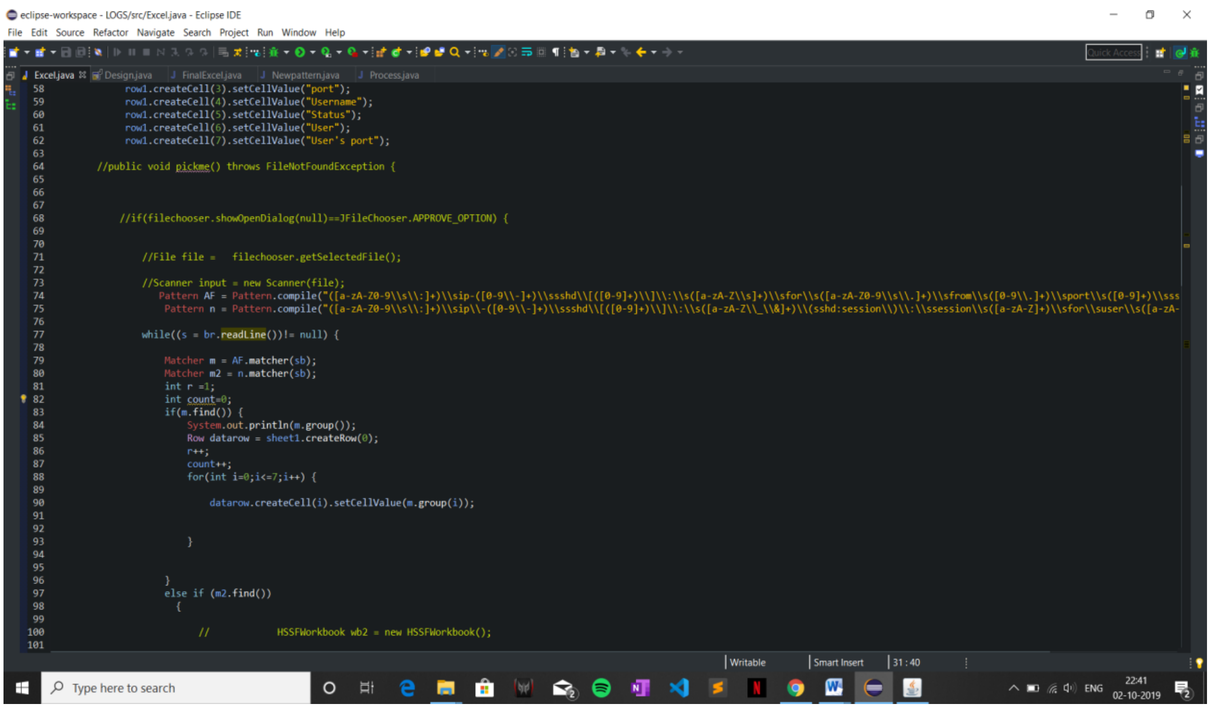
- **Pluggable look and feel**: Swing supports several looks and feels and currently supports Windows, UNIX, Motif, and native Java metal look and feel. Swing allows users to switch look and feel at runtime without restarting the application. By doing this, users can make their own choice to choose which look and feel is the best for them instantly.
- **Lightweight components**: All swing components are lightweight except some top-level containers. Lightweight means component renders or paints itself using drawing primitives of the *Graphics* object instead of relying on the host operating system (OS). As the result, the application presentation is rendered faster and consumed less memory than previous Java GUI applications.
- **Simplified MVC**: Swing uses simplified model-view-architecture (MVC) as the core design behind each of its components called model-delegate. Based on this architecture, each swing component contains a model and a UI delegate. A UI delegate wraps a view and a controller in MVC architecture as the picture below. UI delegate is responsible for painting screen and handling GUI events. Model is in charge of maintaining information or states of the component.

CODE FOR EXCEL SHEET CONVERTOR:-



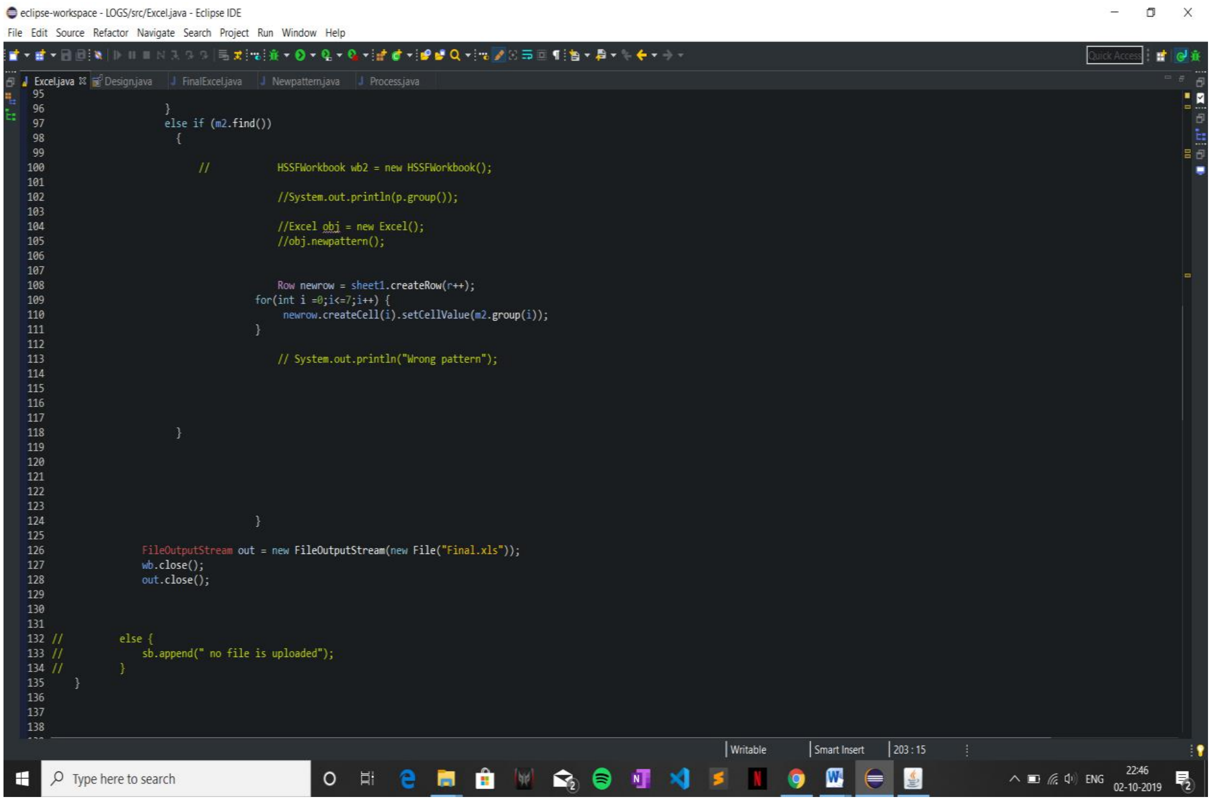
```
import java.io.BufferedReader;

public class Excel {

      HSSFWorkbook wb = new HSSFWorkbook();
      HSSFSheet sheet1 = wb.createSheet();
    //  JFileChooser Filechooser = new JFileChooser();
      StringBuilder sb = new StringBuilder();
     // File file = Filechooser.getSelectedFile();







    void Convertor() throws IOException {
            File file = new File("logs.txt");
             BufferedReader br = new BufferedReader(new FileReader(file));
           //  BufferedWriter write = new BufferedWriter(new FileWriter("excel.xls"));
             String s ;
        //JFileChooser filechooser = new JFileChooser();
        StringBuilder sb = new StringBuilder();
           HSSFWorkbook wb = new HSSFWorkbook();
             HSSFSheet sheet = wb.createSheet();
            HSSFSheet sheet1 = wb.createSheet();

             Row row = sheet.createRow(0);
             row.createCell(0).setCellValue("Full");
             row.createCell(1).setCellValue("date");

             row.createCell(2).setCellValue("ip");
             row.createCell(3).setCellValue("port");
             row.createCell(4).setCellValue("Username");
             row.createCell(5).setCellValue("Status");
             row.createCell(6).setCellValue("User");
            // row.createCell(7).setCellValue("User's port");

             Row row1 = sheet1.createRow(0);
             row1.createCell(0).setCellValue("Full");
             row1.createCell(1).setCellValue("date");

             row1.createCell(2).setCellValue("ip");
             row1.createCell(3).setCellValue("port");
```

Fig 4.8 Code of Excel Sheet Convertor



```
             row1.createCell(3).setCellValue("port");
             row1.createCell(4).setCellValue("Username");
             row1.createCell(5).setCellValue("Status");
             row1.createCell(6).setCellValue("User");
             row1.createCell(7).setCellValue("User's port");

        //public void pickme() throws FileNotFoundException {



            //if(filechooser.showOpenDialog(null)==JFileChooser.APPROVE_OPTION) {


                //File file =   filechooser.getSelectedFile();

                //Scanner input = new Scanner(file);
                   Pattern AF = Pattern.compile("([a-zA-Z0-9\\s\\:]+)\\sip-([0-9\\-]+)\\ssshd\\[([0-9]+)\\]\\:\\s([a-zA-Z\\s]+)\\sfor\\s([a-zA-Z0-9\\s\\.]+)\\sfrom\\s([0-9\\.]+)\\sport\\s([0-9]+)\\sss
                    Pattern n = Pattern.compile("([a-zA-Z0-9\\s\\:]+)\\sip\\-([0-9\\-]+)\\ssshd\\[([0-9]+)\\]\\:\\s([a-zA-Z\\_\\&]+)\\(sshd:session\\)\\:\\ssession\\s([a-zA-Z]+)\\sfor\\suser\\s([a-zA-

                while((s = br.readLine())!= null) {

                    Matcher m = AF.matcher(sb);
                    Matcher m2 = n.matcher(sb);
                    int r =1;
                    int count=0;
                    if(m.find()) {
                        System.out.println(m.group());
                        Row datarow = sheet1.createRow(0);
                        r++;
                        count++;
                        for(int i=0;i<=7;i++) {

                            datarow.createCell(i).setCellValue(m.group(i));


                        }



                    }
                    else if (m2.find())
                      {

                          //            HSSFWorkbook wb2 = new HSSFWorkbook();

```

Fig 4.9 Code of Excel Sheet Continued



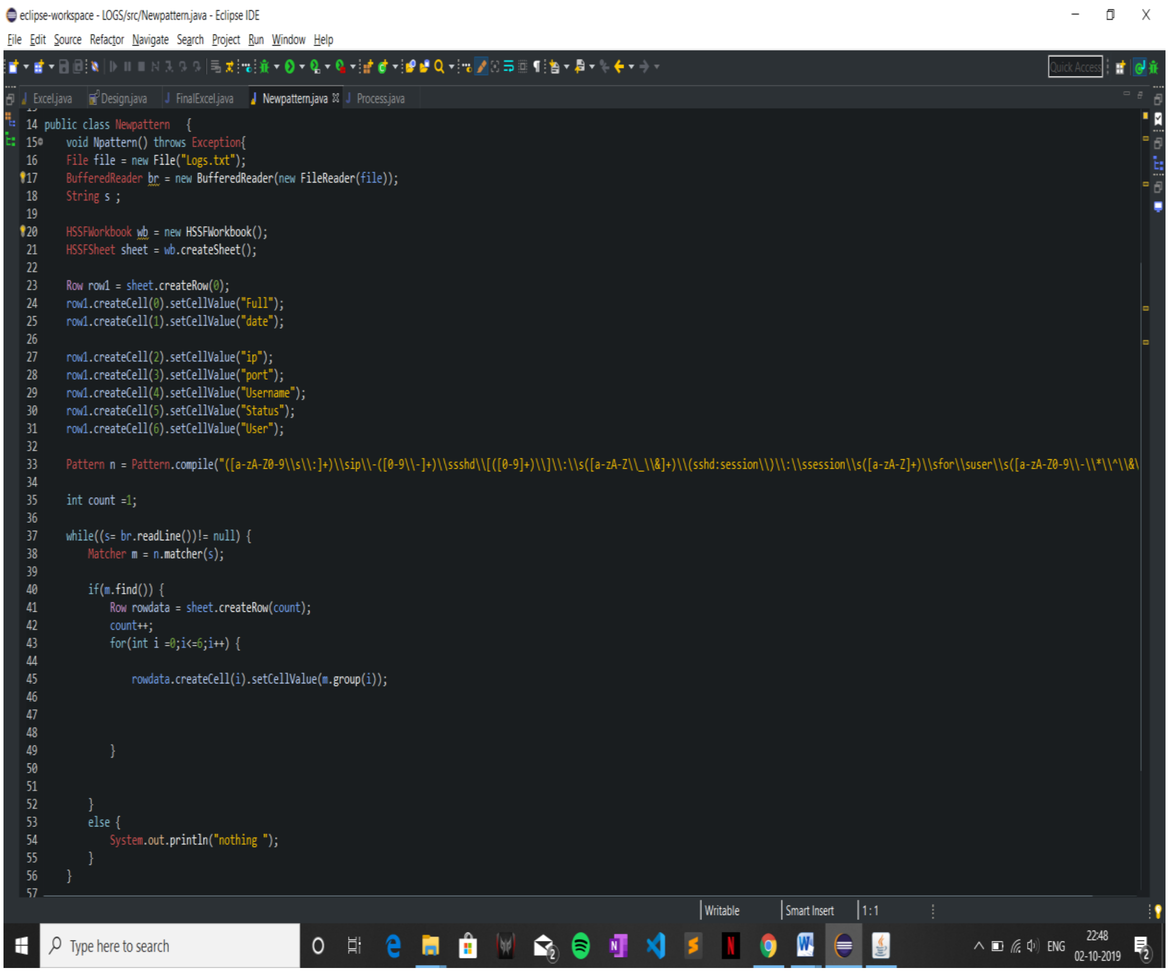
```
                   }
                   else if (m2.find())
                     {

                         //           HSSFWorkbook wb2 = new HSSFWorkbook();

                                      //System.out.println(p.group());

                                      //Excel obj = new Excel();
                                      //obj.newpattern();


                                      Row newrow = sheet1.createRow(r++);
                                  for(int i =0;i<=7;i++) {
                                       newrow.createCell(i).setCellValue(m2.group(i));
                                  }

                                      // System.out.println("Wrong pattern");




                     }





                                  }

               FileOutputStream out = new FileOutputStream(new File("Final.xls"));
               wb.close();
               out.close();



//          else {
               sb.append(" no file is uploaded");
//          }
    }
```

Fig 4.10 Code of Excel Sheet continued

Apache POI:-

Apache POI is an open source java library to create and manipulate various file formats based on Microsoft Office. Using POI, one should be able to perform create, modify and display/read operations on following file formats. For Example, Java doesn't provide built-in support for working with excel files, so we need to look for open source APIs for the job. Apache POI provides Java API for manipulating various file formats based on the Office Open XML (OOXML) standard and OLE2 standard from Microsoft. Apache POI releases are available under the Apache License (V2.0).[iv]

New Pattern File:-



```
public class Newpattern   {
    void Npattern() throws Exception{
    File file = new File("Logs.txt");
    BufferedReader br = new BufferedReader(new FileReader(file));
    String s ;

    HSSFWorkbook wb = new HSSFWorkbook();
    HSSFSheet sheet = wb.createSheet();

    Row row1 = sheet.createRow(0);
    row1.createCell(0).setCellValue("Full");
    row1.createCell(1).setCellValue("date");

    row1.createCell(2).setCellValue("ip");
    row1.createCell(3).setCellValue("port");
    row1.createCell(4).setCellValue("Username");
    row1.createCell(5).setCellValue("Status");
    row1.createCell(6).setCellValue("User");

    Pattern n = Pattern.compile("([a-zA-Z0-9\\s\\:]+)\\sip\\-([0-9\\-]+)\\ssshd\\[([0-9]+)\\]\\:\\s([a-zA-Z\\_\\&]+)\\(sshd:session\\)\\:\\ssession\\s([a-zA-Z]+)\\sfor\\suser\\s([a-zA-Z0-9\\-\\*\\^\\&\

    int count =1;

    while((s= br.readLine())!= null) {
        Matcher m = n.matcher(s);

        if(m.find()) {
            Row rowdata = sheet.createRow(count);
            count++;
            for(int i =0;i<=6;i++) {

                rowdata.createCell(i).setCellValue(m.group(i));



            }


        }
        else {
            System.out.println("nothing ");
        }
    }
```

eclipse-workspace - LOGS/src/Newpattern.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Excel.java
Design.java
FinalExcel.java
Newpattern.java
Process.java
Writable
Smart Insert
1:1
Type here to search
ENG
22:48
02-10-2019

Fig 4.11 Code of Regex Pattern used in Log setter

Regular Expressions:-What Regular Expressions Are Exactly - Terminology basically, a regular expression is a pattern describing a certain amount of text. Their name comes from the mathematical theory on which they are based. But we will not dig into that. Since most people including myself are lazy to type, you will usually find the name abbreviated to regex or regexp. I prefer regex, because it is easy to pronounce the plural “regexes”. In this book, regular expressions are printed between guillemots: «regex». They clearly separate the pattern from the surrounding text and punctuation. This first example is actually a perfectly valid regex. It is the most basic pattern, simply matching the literal text „regex”. A "match" is the piece of text, or sequence of bytes or characters that pattern was found to correspond to by the regex processing software. Matches are indicated by double quotation marks, with the left one at the base of the line. «\b[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,4}\b» is a more complex pattern. It describes a series of letters, digits, dots, underscores, percentage signs and hyphens, followed by an at sign, followed by another series of letters, digits and hyphens, finally followed by a single dot and between two and four letters. In other words: this pattern describes an email address. With the above regular expression pattern, you can search through a text file to find email addresses, or verify if a given string looks like an email address. In this tutorial, I will use the term "string" to indicate the text that I am applying the regular expression to. I will indicate strings using regular double quotes. The term "string" or "character string" is used by programmers to indicate a sequence of characters. In practice, you can use regular expressions with whatever data you can access using the application or programming language you are working with.[v]

^	Matches beginning of line.
$	Matches end of line.
.	Matches any single character except newline. Using m option allows it to match newline as well.
[...]	Matches any single character in brackets.
[^...]	Matches any single character not in brackets
\A	Beginning of entire string
\z	End of entire string
\Z	End of entire string except allowable final line terminator.
re*	Matches 0 or more occurrences of preceding expression.
re+	Matches 1 or more of the previous thing
re?	Matches 0 or 1 occurrence of preceding expression.
re{ n}	Matches exactly n number of occurrences of preceding expression.
re{ n,}	Matches n or more occurrences of preceding expression.
re{ n, m}	Matches at least n and at most m occurrences of preceding expression.
a| b	Matches either a or b.
(re)	Groups regular expressions and remembers matched text.
(?: re)	Groups regular expressions without remembering matched text.
(?> re)	Matches independent pattern without backtracking.
\w	Matches word characters.
\W	Matches nonword characters.
\s	Matches whitespace. Equivalent to [\t\n\r\f].

Regular Expression in java:-

Java provides the java.util.regex package for pattern matching with regular expressions. Java regular expressions are very similar to the Perl programming language and very easy to learn. A regular expression is a special sequence of characters that helps you match or find other strings or sets of strings, using a specialized syntax held in a pattern. They can be used to search, edit, or manipulate text and data. The java.util.regex package primarily consists of the following three classes: Pattern Class: A Pattern object is a compiled representation of a regular expression. The Pattern class provides no public constructors. To create a pattern, you must first invoke one of its public static compile methods, which will then return a Pattern object. These methods accept a regular expression as the first argument. Matcher Class: A Matcher object is the engine that interprets the pattern and performs match operations against an input string. Like the Pattern class, Matcher defines no public constructors. You obtain a Matcher object by invoking the matcher method on a Pattern object. PatternSyntaxException: A PatternSyntaxException object is an unchecked exception that indicates a syntax error in a regular expression pattern.[v]

CHAPTER 5

RESULT

The paper provides an overview of the current log setter system and the new system built using java software.

Log plays an important role in today's world. Especially, in the cloud computing monitoring. Millions of people access to the cloud centre daily and company has to keep records for many purposes such as security, crashes and many more things.

Using log setter has its own benefits such as it consumes less time, also saves the hard work of the employers. Keeping in mind the security purposes logs are compulsory in the companies.

Chapter 6

Comparison and Analysis

In the past decades there was surprisingly low attention paid to problem of getting useful information from log files. It seems there are two main streams of research. The first one concentrates on validating program runs by checking conformity of log files to a state machine. Records in log file are interpreted as transitions of given state machine. If some illegal transitions occur, then there is certainly a problem, either in the software under test or in the state machine specification or in the testing software itself. The second branch of research is represented by articles that just describe various ways of production statistical output. The following items summarize current possible usage of log files:

- generic program debugging and profiling
- tests whether program conforms to a given state machine
- various usage statistics, top ten, etc.
- security monitoring

According to available scientific papers it seems that the most evolving and developed area of log file analysis is the WWW industry. Log files of HTTP servers are nowadays used not only for system load statistic but they offer a very valuable and cheap source of feedback. Providers of web content were the first one who lack more detailed and sophisticated reports based on server logs. They require detecting behavioural patterns, paths, trends etc. Simple statistical methods do not satisfy these needs so an advanced approach must be used.

Some log files, especially small and simple, can be also analysed using common spread sheet or database programs. In such case, the logs are imported into a worksheet or database and then analysed using available functions and tools. The remaining but probably the most promising way of log file processing represents data driven tools like AWK. In connection with regular expressions are such tools very efficient and flexible. On the other hand, they are too low–level, i.e. their usability is limited to text files, one-way, single-pass operation. For higher–level tasks they lack mainly advanced data structures.

To look after every single activity done by client is a very difficult job. Just like in any other programming language software requires logs so that if that software crashes at some point we can track by the help of logs. Similarly, in cloud computing the companies have to keep tracks on the logs. They usually write it in an excel sheet manually. This software will solve this problem and reduce the working time.

This system is user friendly and also in case of different pattern a user can create its own pattern that is user has its own choice to select certain important information that would really help in time consuming.

Chapter 7
CONCLUSION AND FUTURE SCOPE

7.1 Conclusion

WORK DONE:

- The project was successfully designed and is tested for accuracy and quality.
- During this project I have accomplished all the objectives and this project meets the needs of the organization.
- The project will be used for collecting and converting logs into excel.
- The excel file will be formed.
- The sheet would be filled row wise.

GOALS:

- Reduced paper-based work.
- Easy retrieval of information.
- Reduced errors due to human intervention.
- Portable and flexible for further enhancement.

7.2 Future Scope

Logs are very basic ingredient for the online applications and services. It can be further used for security purposes and also to keep track on the old records. It also has scope in data sciences. As internet usage has been rapidly increased in past few years, the bulky amount of data needs to be stored for the future calculations. This system will help us in collecting important data and storing in the form of excel sheet. Also, cloud computing is becoming the major lead of this era.

References

[1] J. H. Andrews: “Theory and practice of log file analysis.” Technical Report 524, Department of Computer Science, University of Western Ontario, May 1998.[i]

[2] Tec-Ed, Inc.: “Assessing Web Site Usability from Server Log Files White Paper.” http://citeseer.nj.nec.com/290488.html[ii]

[3] https://en.wikipedia.org/wiki/Log_file[iii]

[4] https://poi.apache.org/[iv]

[5] https://www.regular-expressions.info/[v]