Contents

1.Introduction

KMMIPS is an educational society established in 2000 by a group of retired civil servant. This is an educational society which serves for the betterment of the students. Initially this college offers three major programs MCA, MBA and B.Ed.

The main aim of the project “Incremental Algorithm For Association Rule Mining Under Dynamic Threshold” is providing an easy to handle the frequent pattern sets into Uncertain graphs. This project also provides features and interface for maintaining the frequent pattern sets in to the particular structure of the graph.

KMM Institute of Post Graduate Studies:

Provide good higher-level educational services.

Develop good communicational skills to students.

Provide good managerial skills to MBA, MCA &B. ED Students.

This approach rests on:

A strategy where we architect, integrate and manage technology services and solutions.A robust off shore development methodology and reduced Manual works of students.

They understand the importance of timing, getting details of necessary information as well as storing the information. A rich portfolio of reusable, modular frameworks helps jump start projects.

A full-service portfolio:

This offers Student and Administrator maintain, architect, integrate and manage through the system.

Services:

This system provides its services to Educational institutions, which are in the field of providing good helpful college Student information system.

1.1 Organization Profile:

“Sathyameva Jayathi” – Truth always Triumph is the motto of our KMM Educational Society, Tirupati.

Nothingis permanent except change. The ongoing advances in computer communications technology continue to have profound effect on the way people work and play. Both the technology itself and the expectations of the people who use it are altering the features of the information system that analysis, design and the widespread deployment of information systems in changing the very nature of the society in which the systems are used. The development of the information economy.

KMM Institute of postgraduate studies very popularly known as KMMIPS has emerged as a major Technological Institute managed by KMM SOCIAL AND EDUCATIONAL DEVELOPMENT SOCIETY, Tirupati. The KMM Society has taken the lead role to establish the institute in academic year 2001-2002. Sprawled over an area of 25 acres, permanent infrastructural facilities are being developed near Tirupati-Madanapalli state Highway at Ramireddypalle. KMMIPS has already secured the approval from the All India Council of Technical Education (AICTE). New Delhi, and Government of Andhra Pradesh and is affiliated to Sri Venkateswara University, Tirupati.

KMMIPS offers admission into the professional courses MCA and MBA with annual intake of 180 and 180 seats each. The institution is governed by the chairman Sri S. Srinivasulu garu, retired IRS (Indian Revenue Service) officer with the support of an advisory body consisting of the eminent personalities form different fields.

Company: KMM Educational Institutions, Tirupathi.

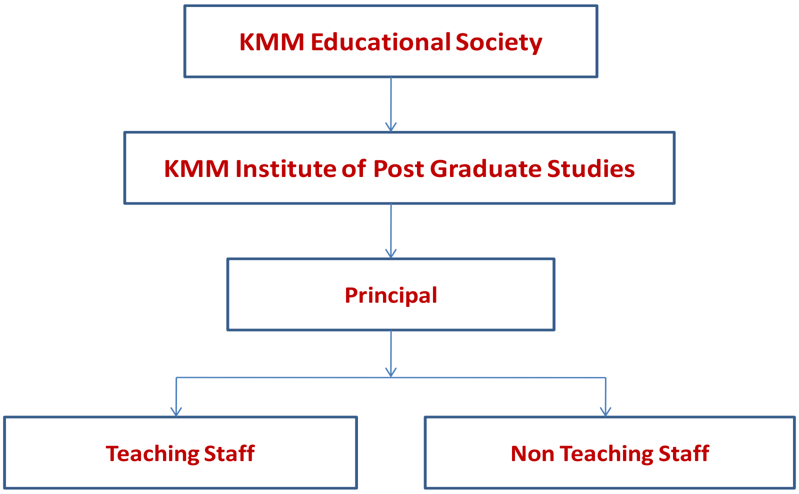
Contact: Madanapalli Highway Road, Ramireddipalle, Tirupathi.

Operating Since: 2000

Managing Director: Sri S. Srinivasulu Garu

Legal Status: Educational Society

1.2 Organization Chart:



1.3 Introduction

Expertise has come to be probably the most probably utilized term this present day. The phrase 'knowledge' has been begun from the Latin phrase 'Datum' throughout the center of the seventeenth century. 'Knowledge' can be viewed as the plural type of 'Datum' which implies the realities and measurement data assembled for further examination and references. At this time life in the world fully relies upon this 'expertise'. From the investigation of existing know-how new hypotheses and articulations will also be drawn. This priceless information drawn from the expertise may also be made favorable for some purposes. For On the off chance that we don't forget exhibit bushel investigation, for the clients, regularly there might be an instance of buying matters. On the off risk that we will to find that example from current information, on the way to be important for the further trade. This is the position the importance of knowledge lies. To put it evidently, we can say that information mining is investigating expertise about expertise. The titanic knowledge which is getting accrued step-by-step must be overseen and the system information mining fills this ought to a measure. Since understanding mining is a big method to be executed, there are numerous sub spaces like factor set mining, consult with and infrequent mining, affiliation rule mining and so on. Consult with thing set mining is the way in which toward investigating things from a dataset established on the bottom restrict. Within the event that the recurrence of these thing units are extra prominent than an aspect, at that point they're viewed as talk over with factor units. In the event that their recurrence isn't exactly the given restrict those things can be remembered for the amazing or rare factor sets. When the incessant or exclusive things are observed, the principle errand of understanding mining is association Rule Mining (ARM). Rule Mining alludes discovering useful requirements with the goal that we can join into an finish. A normal will also be characterized as A→B, which means if A occurs B is likewise in charge to be arise with A. A general has a common structure. Rule includes of predecessor (An) and subsequent (B). Forerunner is the reality and the consequent is the aftereffect of that truth. Take into account the mannequin dataset which is regarded within the table I. Six certain exchanges are given which can be acquired via 6 quite a lot of consumers. The exchanges comprise matters like bread, margarine, sugar, milk, tea, rice, wheat and so on. On this mannequin on the off hazard that we recall the base support as four we can locate the incessant matters as Milk, Bread, Butter and Sugar. From these incessant thing sets it's conceivable to extricate rules which is useful to observe the purchasing sample of a client. For illustration the average Milk → Bread painting the examples the place milk is bought alongside bread and the commonplace (Milk, Bread) → Butter depicts the occasions the place milk and bread bought alongside margarine. To find the value of rules we make use of the measure simple task. On the off hazard that the certainty of a general is extra noteworthy than the limit, at that point that regular can be regarded as a important principle. The staying of the paper is prepared as follows. Phase 2 depicts the writing survey accomplished for the proposed technique. The earlier techniques for affiliation rule mining is laid out here. In section three the proposed credibility situated method for gradual datasets is depicted. Area 4 assessments various investigation of the proposed procedure ultimately section 5 gives a finish for the paper.

1.4 Project Report Layout:

This project consists of eight chapters with an overview of computerization of Staff Information System.

Chapter 1 Introduction: It deals with the overview of the organization and its administrative hierarchy.

Chapter 2 Genesis of the Study: It outlines the Existing problem, the solution proposed, methodology used and looks at how the preliminary investigation carried out, what are all its scope and objectives and limitations of existing manual system.

Chapter 3 Feasibility Study: It is used to test the feasibility of the project i.e., Operational, Technical and Economical feasibilities of the system.

Chapter 4 System Analysis: It deals with software requirement analysis. The various entities and their requirements and their relationships are discussed using E-R Diagrams.

Chapter 5 System requirements: It explains about the various hardware and software requirements and their features.

Chapter 6 System Design: It contains the description of all the tables and its attributes, design principles, HIPO charts and user interface design.

Chapter 7 System Testing: It presents software-testing strategies and techniques like white box testing, back box testing.

Chapter 8 Implementation: It gives the software and hardware implementation details of the system.

Chapter 9 Conclusion:

2. Genesis of Study

This chapter deals with the study of the existing system and describes the need for the proposed system to overcome the drawbacks in the existing system. It also specifies the objectives, scope of the system and also the methodology for the system development.

Hence the genesis of the study clearly depicts the factors regarding the beginning of the existing system and its extension to the proposed system and it includes the following.

2.1 Aim:

“Incremental Algorithm For Association Rule Mining Under Dynamic Threshold” is used to draw the uncertain graphs for the frequent pattern data sets. Every sort of task is performed by the system, such as take a csv file and then process that file after that we can show it in a particular table etc. Also, the latest information is right available for the officials and executives wherever they require.

2.2 Problem Description:

2. 2. 1. Description Of Existing System:

From the analysis of existing data new theories and statements can be drawn. These useful information drawn from the data can be made advantageous for many applications. For eg. if we consider market basket analysis, for the customers , always there will be a pattern of buying things. If we are able to find that pattern from existing data, that will be useful for the further business. This is where the relevance of data lies.

Disadvantages:

- It difficult process
- It requires more time

2.3 Scope:

The system extends its scope from manual to computerize. The system provides scope for the Efficient Mining of Frequent Patterns on Uncertain Graphs.

2.4 Methodology:

The source of information for developing the proposed system is gathered directly from clients of end user who is going to use the package becomes the primary source to give information.

A detailed study and understanding of the existing system is done either by questionnaires or by conducting interviews before developing the proposed one. Different inputs, process and output are well understood before designing the system.

Application software likeJavaScriptwith HTML. So as to give the results as per the requirements of the system. The concepts of software engineering have been implemented successfully and uniformly throughout the system. The performance of the integrated system will be uniform.

For building this project, we followed Rapid Application Development Model as the requirements of this project are completed analyzed at the beginning of the project itself.

"Rapid Application Development (RAD) is an incremental software development process model that emphasizes a very short development cycle [typically 60-90 days]," The RAD model, shown in the above figure is a high-speed adaptation of the waterfall model, where the result of each cycles a fully functional system.

2.5 Proposed System:

We advocate a method based on the believability measure. Credibility is the measure for being clear. Like the confabulation enlivened method, this credibility situated method additionally exams the dataset simply as soon as and this improves the memory effectiveness. So a clearness based ordinary mining calculation is proposed which makes use of a fascinating measure referred to as credibility. Believability is the likelihood of the anticipated actualities to be valid if the end is valid. This proposed calculation can mine affiliation administers through a solitary go by means of the record. The precept bit of leeway of this proposed calculation is that it will be enormously worthy for gradual datasets. Proposed a further calculation which oversees each continuous and infrequent item sets more proficiently than Apriori and FP growth Algorithms. In this proposed technique we makes use of the inspiration of cogency or credibility on steady datasets.

2.5.1 Advantages:

- It easy process
- It require less time

Greater Processing Speed:

Using computers inherently able to calculate, sort, retrieve data with greater speed than that of the manual doing. We can get results in less time.

Better Accuracy and Improved Consistency:

The computer carries our computing steps including arithmetic's accurately and consistently from which really human is escaped which yields more fatigue and boredom.

Cost Reduction:

Using computerization, we can do the required operations with lower cost than any other methods. Hence by computerization, we can reduce the cost drastically.

2.5.2 Limitations of Proposed system:

This system doesn't work on the Linux Operating System. It works only on windows environment.

If software is corrupted, whole data will be collapsed.

Complicated to operate.

Online system require high-speed internet connectivity.

Risk of computer virus.

3. Feasibility Study

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- Economical feasibility
- Technical feasibility
- Social feasibility

3.1 Economical Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

3.2 Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

3.3 Social Feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

4. System Analysis

4.1 UML diagrams:

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS:

The Primary goals in the design of the UML are as follows:

Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.

Provide extendibility and specialization mechanisms to extend the core concepts.

Be independent of particular programming languages and development process.

Provide a formal basis for understanding the modeling language.

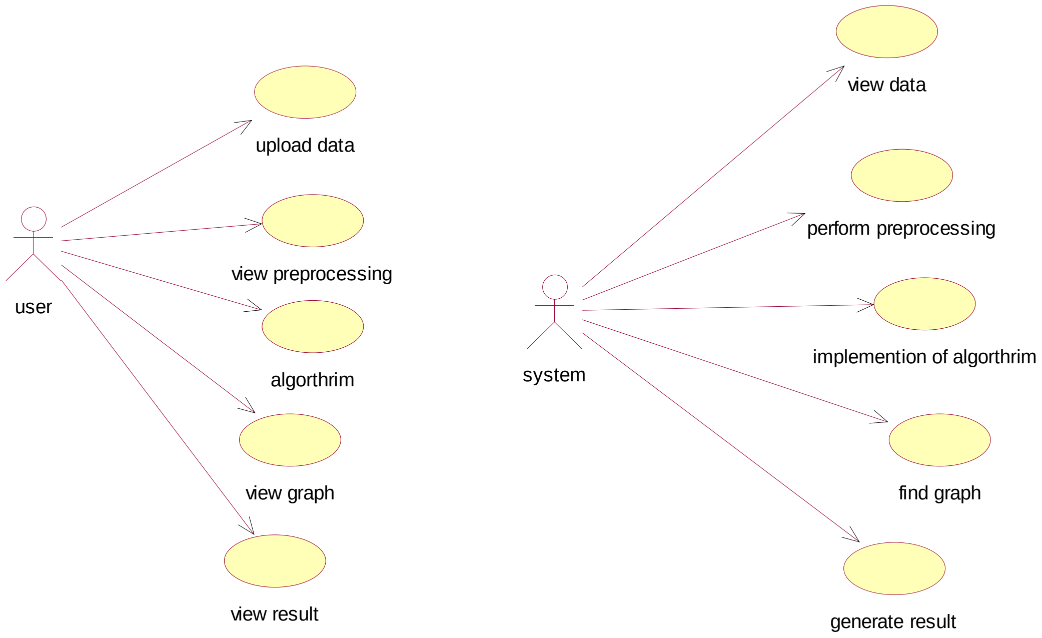
Encourage the growth of OO tools market.

Support higher level development concepts such as collaborations, frameworks, patterns and components.

Integrate best practices.

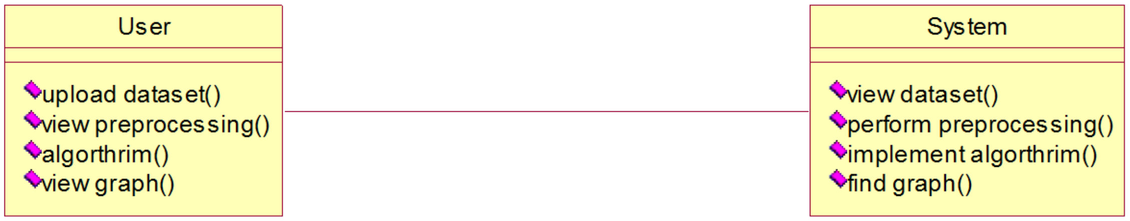
4.2 Use case diagram:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



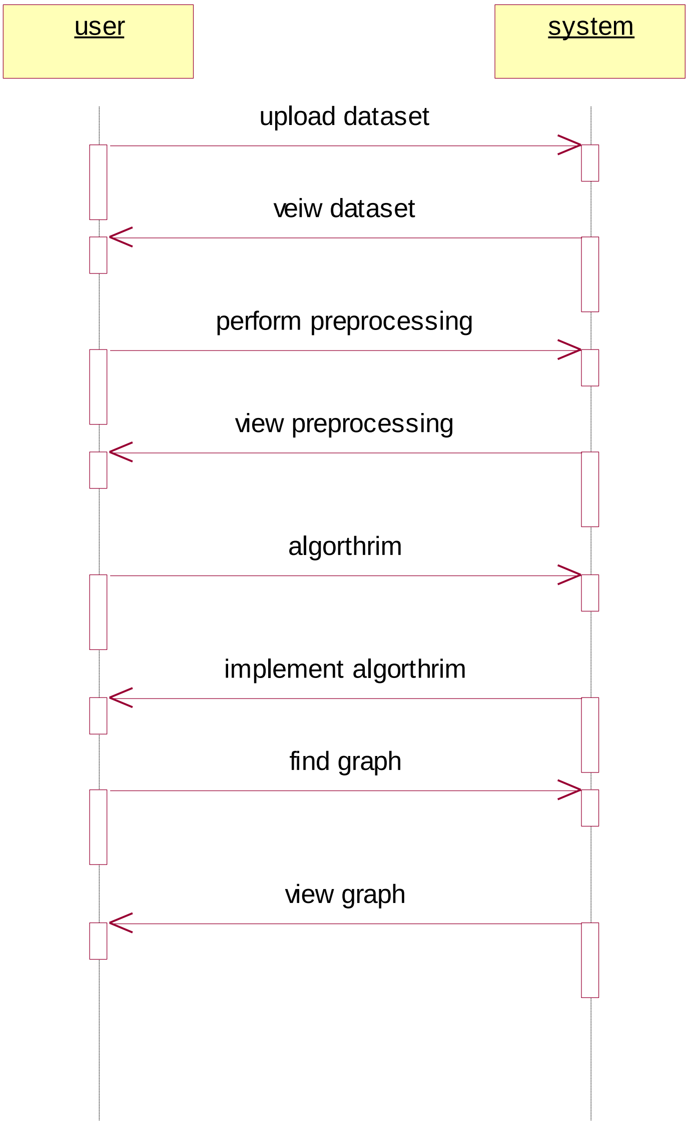
4.3 Class diagram:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



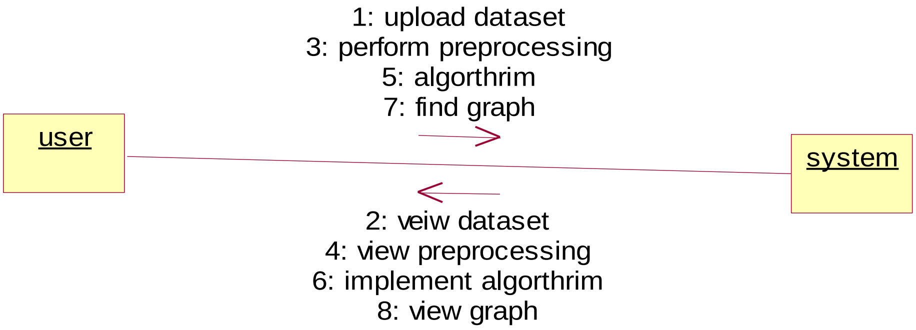
4.4 Sequence Diagram:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios



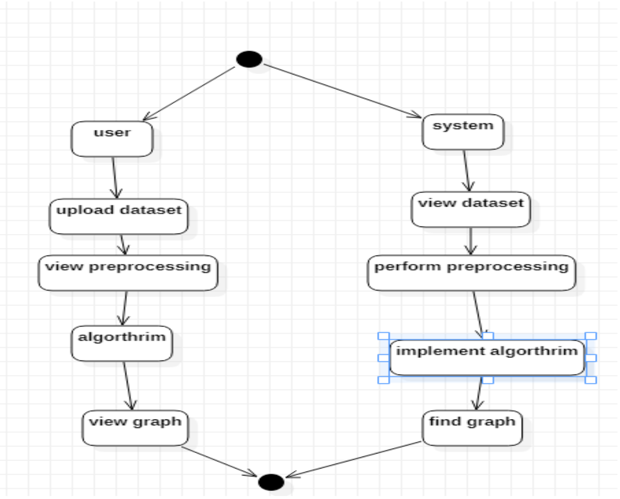
4.5 Collaboration Diagram:

In collaboration diagram the method call sequence is indicated by some numbering technique as shown below. The number indicates how the methods are called one after another. We have taken the same order management system to describe the collaboration diagram. The method calls are similar to that of a sequence diagram. But the difference is that the sequence diagram does not describe the object organization whereas the collaboration diagram shows the object organization.



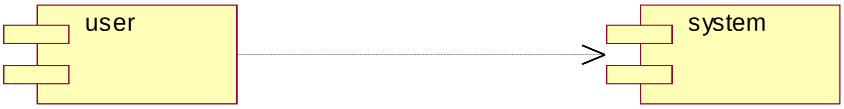
4.6 Activity diagram:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



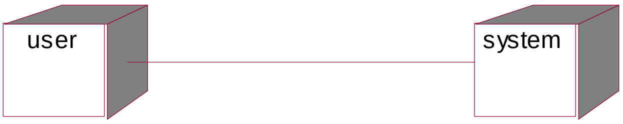
4.7 Component Diagram

Component diagrams are used to describe the physical artifacts of a system. This artifact includes files, executable, libraries etc. So the purpose of this diagram is different, Component diagrams are used during the implementation phase of an application. But it is prepared well in advance to visualize the implementation details. Initially the system is designed using different UML diagrams and then when the artifacts are ready component diagrams are used to get an idea of the implementation.



4.8 Deployment Diagram

Deployment diagram represents the deployment view of a system. It is related to the component diagram. Because the components are deployed using the deployment diagrams. A deployment diagram consists of nodes. Nodes are nothing but physical hard ware's used to deploy the application.



5. System Requirements Analysis

System requirements gives the idea about what are the necessary things that are needed for proposed system, which plays very important role in development of any system. This chapter deals with what are hardware components that are needed for the system, application software that are required for the development of the system.

The environment deals with the features of software. HTML, CSS. Front end tools help to visualize the system through naked eyes while back end helps in activities which are unseen to the end user.

5.1 Hardware Requirements:

Processor	-	I3/Intel Processor
RAM	-	4GB (min)
Hard Disk	-	160GB
Key Board	-	Standard Windows Keyboard
Mouse	-	Two or Three Button Mouse
Monitor	-	SVGA

5.2 Software Requirements:

Operating System	:	Windows 7/8/10
Application Server	:	Tomcat 7.0
Front End	:	HTML, JSP
Scripts	:	JavaScript.
Server side Script	:	Java Server Pages.
Database	:	My SQL 6.0
Database Connectivity	:	JDBC

5.2.1 Features of Software

The Java Programming Language

The Java programming language is a high-level language that can be characterized by all of the following buzzwords:

Simple

Architecture neutral

Object oriented

Portable

Distributed

High performance

Interpreted

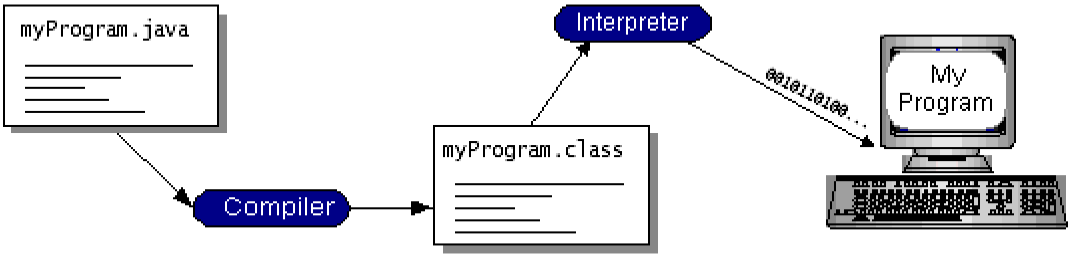
Multithreaded

Robust

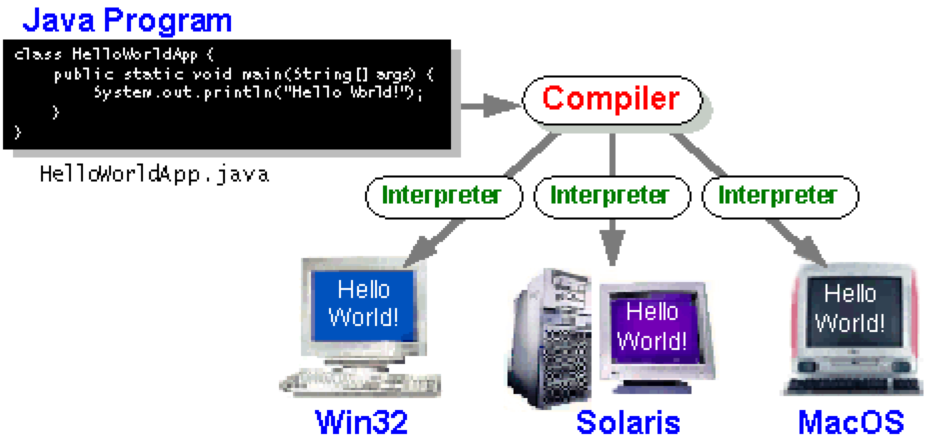
Dynamic

Secure

With most programming languages, you either compile or interpret a program so that you can run it on your computer. The Java programming language is unusual in that a program is both compiled and interpreted. With the compiler, first you translate a program into an intermediate language called *Java byte codes* —the platform-independent codes interpreted by the interpreter on the Java platform. The interpreter parses and runs each Java byte code instruction on the computer. Compilation happens just once; interpretation occurs each time the program is executed. The following figure illustrates how this works.



You can think of Java byte codes as the machine code instructions for the *Java Virtual Machine* (Java VM). Every Java interpreter, whether it’s a development tool or a Web browser that can run applets, is an implementation of the Java VM. Java byte codes help make “write once, run anywhere” possible. You can compile your program into byte codes on any platform that has a Java compiler. The byte codes can then be run on any implementation of the Java VM. That means that as long as a computer has a Java VM, the same program written in the Java programming language can run on Windows 2000, a Solaris workstation, or on an iMac.



```
class HelloWorldApp {
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

The Java Platform

A *platform* is the hardware or software environment in which a program runs. We've already mentioned some of the most popular platforms like Windows 2000, Linux, Solaris, and MacOS. Most platforms can be described as a combination of the operating system and hardware. The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other hardware-based platforms.

The Java platform has two components:

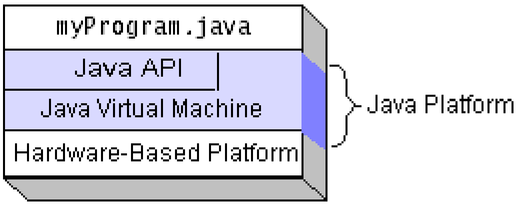
The *Java Virtual Machine* (Java VM)

The *Java Application Programming Interface* (Java API)

You've already been introduced to the Java VM. It's the base for the Java platform and is ported onto various hardware-based platforms.

The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets. The Java API is grouped into libraries of related classes and interfaces; these libraries are known as *packages*. The next section, What Can Java Technology Do? Highlights what functionality some of the packages in the Java API provide.

The following figure depicts a program that's running on the Java platform. As the figure shows, the Java API and the virtual machine insulate the program from the hardware.



Native code is code that after you compile it, the compiled code runs on a specific hardware platform. As a platform-independent environment, the Java platform can be a bit slower than native code. However, smart compilers, well-tuned interpreters, and just-in-time byte code compilers can bring performance close to that of native code without threatening portability.

What Can Java Technology Do?

The most common types of programs written in the Java programming language are *applets* and *applications*. If you've surfed the Web, you're probably already familiar with applets. An applet is a program that adheres to certain conventions that allow it to run within a Java-enabled browser.

However, the Java programming language is not just for writing cute, entertaining applets for the Web. The general-purpose, high-level Java programming language is also a powerful software platform. Using the generous API, you can write many types of programs.

An application is a standalone program that runs directly on the Java platform. A special kind of application known as a *server* serves and supports clients on a network. Examples of servers are Web servers, proxy servers, mail servers, and print servers. Another specialized program is a *servlet*. A servlet can almost be thought of as an applet that runs on the server side. Java Servlets are a popular choice for building interactive web applications, replacing the use of CGI scripts. Servlets are similar to applets in that they are runtime extensions of applications. Instead of working in browsers, though, servlets run within Java Web servers, configuring or tailoring the server.

How does the API support all these kinds of programs? It does so with packages of software components that provides a wide range of functionality. Every full implementation of the Java platform gives you the following features:

The essentials: Objects, strings, threads, numbers, input and output, data structures, system properties, date and time, and so on.

Applets: The set of conventions used by applets.

Networking: URLs, TCP (Transmission Control Protocol), UDP (User Data gram Protocol) sockets, and IP (Internet Protocol) addresses.

Internationalization: Help for writing programs that can be localized for users worldwide. Programs can automatically adapt to specific locales and be displayed in the appropriate language.

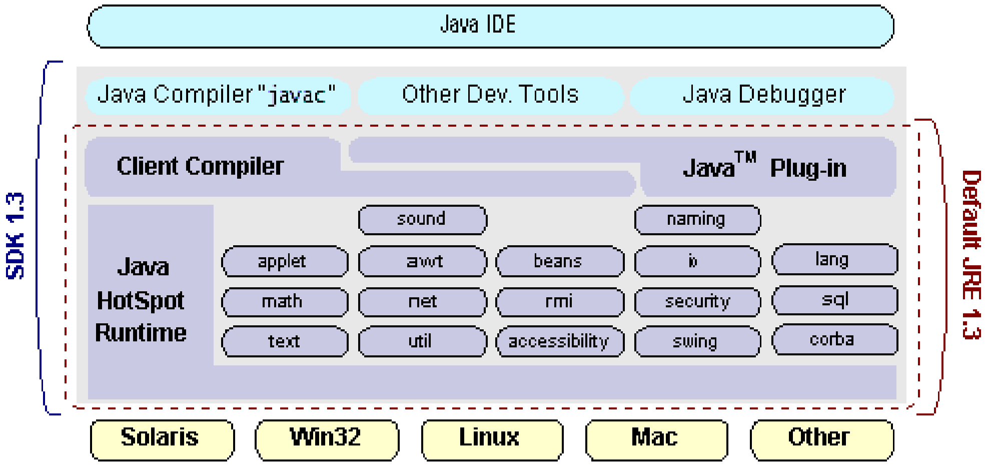
Security: Both low level and high level, including electronic signatures, public and private key management, access control, and certificates.

Software components: Known as JavaBeans^{TM}, can plug into existing component architectures.

Object serialization: Allows lightweight persistence and communication via Remote Method Invocation (RMI).

Java Database Connectivity (JDBC^{TM}): Provides uniform access to a wide range of relational databases.

The Java platform also has APIs for 2D and 3D graphics, accessibility, servers, collaboration, telephony, speech, animation, and more. The following figure depicts what is included in the Java 2 SDK.



How Will Java Technology Change My Life?

We can’t promise you fame, fortune, or even a job if you learn the Java programming language. Still, it is likely to make your programs better and requires less effort than other languages. We believe that Java technology will help you do the following:

Get started quickly: Although the Java programming language is a powerful object-oriented language, it’s easy to learn, especially for programmers already familiar with C or C++.

Write less code: Comparisons of program metrics (class counts, method counts, and so on) suggest that a program written in the Java programming language can be four times smaller than the same program in C++.

Write better code: The Java programming language encourages good coding practices, and its garbage collection helps you avoid memory leaks. Its object orientation, its JavaBeans component architecture, and its wide-ranging, easily extendible API let you reuse other people’s tested code and introduce fewer bugs.

Develop programs more quickly: Your development time may be as much as twice as fast versus writing the same program in C++. Why? You write fewer lines of code and it is a simpler programming language than C++.

Avoid platform dependencies with 100% Pure Java: You can keep your program portable by avoiding the use of libraries written in other languages. The 100% Pure Java^{TM}Product Certification Program has a repository of historical process manuals, white papers, brochures, and similar materials online.

Write once, run anywhere: Because 100% Pure Java programs are compiled into machine-independent byte codes, they run consistently on any Java platform.

Distribute software more easily: You can upgrade applets easily from a central server. Applets take advantage of the feature of allowing new classes to be loaded “on the fly,” without recompiling the entire program.

ODBC

Microsoft Open Database Connectivity (ODBC) is a standard programming interface for application developers and database systems providers. Before ODBC became a *de facto* standard for Windows programs to interface with database systems, programmers had to use proprietary languages for each database they wanted to connect to. Now, ODBC has made the choice of the database system almost irrelevant from a coding perspective, which is as it should be. Application developers have much more important things to worry about than the syntax that is needed to port their program from one database to another when business needs suddenly change.

Through the ODBC Administrator in Control Panel, you can specify the particular database that is associated with a data source that an ODBC application program is written to use. Think of an ODBC data source as a door with a name on it. Each door will lead you to a particular database. For example, the data source named Sales Figures might be a SQL Server database, whereas the Accounts Payable data source could refer to an Access database. The physical database referred to by a data source can reside anywhere on the LAN.

The ODBC system files are not installed on your system by Windows 95. Rather, they are installed when you setup a separate database application, such as SQL Server Client or Visual Basic 4.0. When the ODBC icon is installed in Control Panel, it uses a file called ODBCINST.DLL. It is also possible to administer your ODBC data sources through a stand-alone program called ODBCADM.EXE.

From a programming perspective, the beauty of ODBC is that the application can be written to use the same set of function calls to interface with any data source, regardless of the database vendor. The source code of the application doesn’t change whether it talks to Oracle or SQL Server. We only mention these two as an example. There are ODBC drivers available for several dozen popular database systems. Even Excel spreadsheets and plain text files can be turned into data sources. The operating system uses the Registry information written by ODBC Administrator to determine which low-level ODBC drivers are needed to talk to the data source (such as the interface to Oracle or SQL Server). The loading of the ODBC drivers is transparent to the ODBC application program. In a client/server environment, the ODBC API even handles many of the network issues for the application programmer.

The advantages of this scheme are so numerous that you are probably thinking there must be some catch. The only disadvantage of ODBC is that it isn't as efficient as talking directly to the native database interface. ODBC has had many detractors make the charge that it is too slow. Microsoft has always claimed that the critical factor in performance is the quality of the driver software that is used. In our humble opinion, this is true. The availability of good ODBC drivers has improved a great deal recently. And anyway, the criticism about performance is somewhat analogous to those who said that compilers would never match the speed of pure assembly language. Maybe not, but the compiler (or ODBC) gives you the opportunity to write cleaner programs, which means you finish sooner. Meanwhile, computers get faster every year.

JDBC

In an effort to set an independent database standard API for Java; Sun Microsystems developed Java Database Connectivity, or JDBC. JDBC offers a generic SQL database access mechanism that provides a consistent interface to a variety of RDBMSs. This consistent interface is achieved through the use of "plug-in" database connectivity modules, or *drivers*. If a database vendor wishes to have JDBC support, he or she must provide the driver for each platform that the database and Java run on.

To gain a wider acceptance of JDBC, Sun based JDBC's framework on ODBC. As you discovered earlier in this chapter, ODBC has widespread support on a variety of platforms. Basing JDBC on ODBC will allow vendors to bring JDBC drivers to market much faster than developing a completely new connectivity solution.

JDBC was announced in March of 1996. It was released for a 90 day public review that ended June 8, 1996. Because of user input, the final JDBC v1.0 specification was released soon after.

The remainder of this section will cover enough information about JDBC for you to know what it is about and how to use it effectively. This is by no means a complete overview of JDBC. That would fill an entire book.

JDBC Goals

Few software packages are designed without goals in mind. JDBC is one that, because of its many goals, drove the development of the API. These goals, in conjunction with early reviewer feedback, have finalized the JDBC class library into a solid framework for building database applications in Java.

The goals that were set for JDBC are important. They will give you some insight as to why certain classes and functionalities behave the way they do. The eight design goals for JDBC are as follows:

SQL Level API

The designers felt that their main goal was to define a SQL interface for Java. Although not the lowest database interface level possible, it is at a low enough level for higher-level tools and APIs to be created. Conversely, it is at a high enough level for application programmers to use it confidently. Attaining this goal allows for future tool vendors to “generate” JDBC code and to hide many of JDBC’s complexities from the end user.

SQL Conformance

SQL syntax varies as you move from database vendor to database vendor. In an effort to support a wide variety of vendors, JDBC will allow any query statement to be passed through it to the underlying database driver. This allows the connectivity module to handle non-standard functionality in a manner that is suitable for its users.

JDBC must be implemental on top of common database interfaces

The JDBC SQL API must “sit” on top of other common SQL level APIs. This goal allows JDBC to use existing ODBC level drivers by the use of a software interface. This interface would translate JDBC calls to ODBC and vice versa.

Provide a Java interface that is consistent with the rest of the Java system

Because of Java’s acceptance in the user community thus far, the designers feel that they should not stray from the current design of the core Java system.

Keep it simple

This goal probably appears in all software design goal listings. JDBC is no exception. Sun felt that the design of JDBC should be very simple, allowing for only one method of completing a task per mechanism. Allowing duplicate functionality only serves to confuse the users of the API.

Use strong, static typing wherever possible

Strong typing allows for more error checking to be done at compile time; also, less error appear at runtime.

Keep the common cases simple

Because more often than not, the usual SQL calls used by the programmer are simple SELECT's, INSERT's, DELETE's and UPDATE's, these queries should be simple to perform with JDBC. However, more complex SQL statements should also be possible.

Finally we decided to proceed the implementation using JavaNetworking.

And for dynamically updating the cache table we go for MSAccessdatabase.

Java ha two things: a programming language and a platform.

Java is a high-level programming language that is all of the following

Simple

Object-oriented

Distributed

Interpreted

Robust

Secure

Architecture-neutral

Portable

High-performance

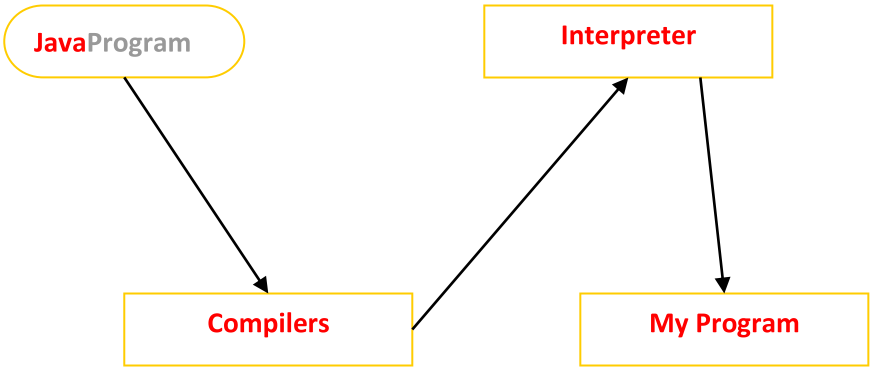
multithreaded

Dynamic

Java is also unusual in that each Java program is both compiled and interpreted. With a compile you translate a Java program into an intermediate language called

Java byte codes the platform-independent code instruction is passed and run on the computer.

Compilation happens just once; interpretation occurs each time the program is executed. The figure illustrates how this works.



You can think of Java byte codes as the machine code instructions for the Java Virtual Machine (Java VM). Every Java interpreter, whether it's a Java development tool or a Web browser that can run Java applets, is an implementation of the Java VM. The Java VM can also be implemented in hardware.

Java byte codes help make "write once, run anywhere" possible. You can compile your Java program into byte codes on my platform that has a Java compiler. The byte codes can then be run any implementation of the Java VM. For example, the same Java program can run Windows NT, Solaris, and Macintosh.

HTML

HTML stands for Hyper Text Markup Language, which is the most widely used language on Web to develop web pages. HTML was created by Berners-Lee in late 1991 but "HTML 2.0" was the first standard HTML specification which was published in 1995. HTML 4.01 was a major version of HTML and it was published in late 1999. Though HTML 4.01 version is widely used but currently we are having HTML-5 version which is an extension to HTML 4.01, and this version was published in 2012.Hypertext refers to the way in which Web pages (HTML documents) are linked together. Thus the link available on a webpage is called Hypertext.As its name suggests, HTML is a Markup Language which means you use HTML to simply mark up a text document with tags that tell a Web browser how to structure it to display.

Originally, HTML was developed with the intent of defining the structure of documents like headings, paragraphs, lists, and so forth to facilitate the sharing of scientific information between researchers.

These are some basic tags of html

Tag	Description
<!DOCTYPE...>	This tag defines the document type and HTML version.
<html>	This tag encloses the complete HTML document and mainly comprises of document header which is represented by <head>...</head> and document body which is represented by <body>...</body> tags.
<head>	This tag represents the document's header which can keep other HTML tags like <title>, <link> etc.
<title>	The <title> tag is used inside the <head> tag to mention the document title.
<body>	This tag represents the document's body which keeps other HTML tags like <h1>, <div>, <p> etc.
<h1>	This tag represents the heading.
<p>	This tag represents a paragraph.

6. System Design

6.1 Introduction:

Design is the first step in the development phase for any system. It may be defined as the "Process of applying various techniques and principles for the purpose of designing a device, a process, or a system".

Software design is an iterative process through which requirements are translated into a "Blue Print" for constructing the software. Preliminary design is concerned with the transformation of requirements in to data and software architecture.

The design is a solution, a "how to" approach to the creation of a new system. This is composed of several steps. It provides the understanding and procedural details necessary for implementing the system recommended.

The database design transforms the information domain model created during analysis into the data structures that will be required to implement software.The architectural design defines the relationship among major structural elements of the program.

The interface design describes how the software communicates within itself, to systems that interoperate with it, and with humans who use it. An interface implements flow of information.

6.2 Design Principles:

Basic design principles that enable the software engineer to navigate the design process are:

The design should be traceable to the analysis model

The design should minimize the intellectual distance between the software and the problem, as it exists in the real world.

6.3 Database Design:

The goal of database design is to generate a set of relation schemes that allow us to store information without necessary redundancy and allows us to retrieve information easily. We can achieve optimization, ease of use in which data is stored in the form of tables and there exists a relation between or among tables.

The design objectives must be

To reduce redundancy

To arrive at loss-less join

To reduce the time as compared to the present system.

To reduce the number of errors.

6.3.1 Normalization

Normalization of relation schema is done to eliminate insertion and deletion anomalies that exist in databases. Normalization is step-by-step reversible process of converting given collection of relations have a progressively simpler and regular structure.

To make it feasible to represent any relation in the database.

To obtain powerful retrieval algorithms based on a simpler collection of relational operations.

To free relations from undesirable insertions, update and deletion dependencies.

A relation R is said to be in 1 NF if all underlying domains contain atomic values only.

A relation R is said to be in 2NF if and only if it is in 1 NF and every non-key attribute is non-transitively dependent on the primary key.

A relation R is said to be in 3 NF if it is in 2 NF and its non-key attribute is non-transitively dependent on its primary key.

6.4 Data Base Tables:

Table 1

Transaction	List of Items

T1	I1,I2,I3
T2	I2,I3,I4
T3	I4,I5
T4	I1,I2,I4
T5	I1,I2,I3,I5
T6	I1,I2,I3,I4

Table 2

Item	Count
I1	4
I2	5
I3	4
I4	4
I5	2

Table 3

Item	Count
I2	5
I1	4
I3	4
I4	4

Table 4

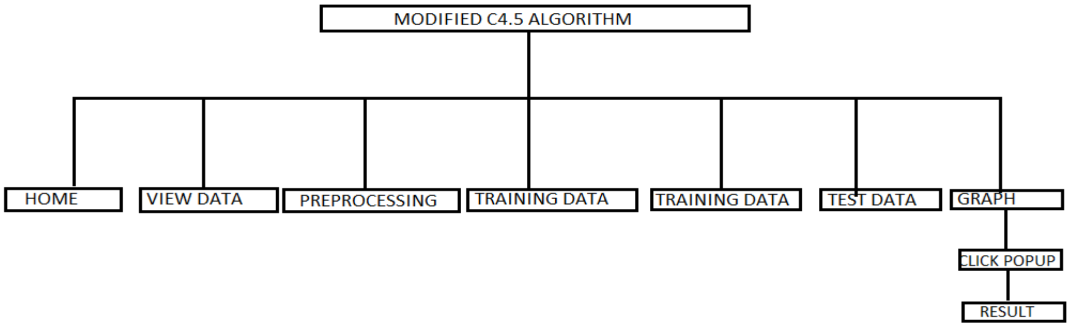
Item	Conditional pattern base	Conditional FP Tree	Frequents Patterns
I4	{I2,I1,I3:1},{I2,I3:1}	{I2:2, I3:2}	{I2,I4:2},{I3,I4:2},{I2,I3,I4:2}
I3	{I2,I1:3},{I2:1}	{I2:4, I1:3}	{I2,I3:4}, {I1:I3:3}, {I2,I1,I3:3}
I1	{I2:4}	{I2:4}	{I2,I1:4}

6.5 HIPO Chart

HIPO (Hierarchical Input Process Output) diagram is a combination of two organized method to analyze the system and provide the means of documentation.HIPO diagram represents the hierarchy of modules in the software system. Analyst uses HIPO diagram in order to obtain high-level view of system functions. It decomposes functions into sub-functions in a hierarchical manner.

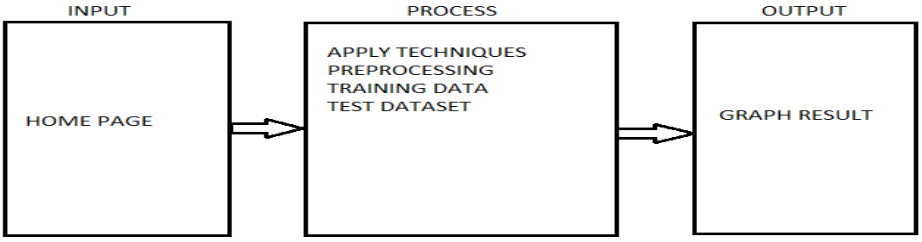
HIPO diagrams are good for documentation purpose. Their graphical representation makes it easier for designers and managers to get the pictorial idea of the system structure.

6.5.1 Hierarchy Chart



6.5.2 IPO Chart

In contrast to IPO (Input Process Output) diagram, which depicts the flow of control and data in a module, HIPO does not provide any information about data flow or control flow.



6.6 Proposed Algorithm:

Algorithm: Frequent Pattern Growth Algorithm

This algorithm is an improvement to the Apriori method. A frequent pattern is generated without the need for candidate generation. FP growth algorithm represents the database in the form of a tree called a frequent pattern tree or FP tree.

This tree structure will maintain the association between the itemsets. The database is fragmented using one frequent item. This fragmented part is called “pattern fragment”. The itemsets of these fragmented patterns are analyzed. Thus with this method, the search for frequent itemsets is reduced comparatively.

FP Tree

Frequent Pattern Tree is a tree-like structure that is made with the initial itemsets of the database. The purpose of the FP tree is to mine the most frequent pattern. Each node of the FP tree represents an item of the itemset.

The root node represents null while the lower nodes represent the itemsets. The association of the nodes with the lower nodes that is the itemsets with the other itemsets are maintained while forming the tree.

Frequent Pattern Algorithm Steps

The frequent pattern growth method lets us find the frequent pattern without candidate generation.

Let us see the steps followed to mine the frequent pattern using frequent pattern growth algorithm:

1) The first step is to scan the database to find the occurrences of the itemsets in the database. This step is the same as the first step of Apriori. The count of 1-itemsets in the database is called support count or frequency of 1-itemset.

2) The second step is to construct the FP tree. For this, create the root of the tree. The root is represented by null.

3) The next step is to scan the database again and examine the transactions. Examine the first transaction and find out the itemset in it. The itemset with the max count is taken at the top, the next itemset with lower count and so on. It means that the branch of the tree is constructed with transaction itemsets in descending order of count.

4) The next transaction in the database is examined. The itemsets are ordered in descending order of count. If any itemset of this transaction is already present in another branch (for example in the 1st transaction), then this transaction branch would share a common prefix to the root.

This means that the common itemset is linked to the new node of another itemset in this transaction.

5) Also, the count of the itemset is incremented as it occurs in the transactions. Both the common node and new node count is increased by 1 as they are created and linked according to transactions.

6) The next step is to mine the created FP Tree. For this, the lowest node is examined first along with the links of the lowest nodes. The lowest node represents the frequency pattern length 1. From this, traverse the path in the FP Tree. This path or paths are called a conditional pattern base.

Conditional pattern base is a sub-database consisting of prefix paths in the FP tree occurring with the lowest node (suffix).

7) Construct a Conditional FP Tree, which is formed by a count of item sets in the path. The item sets meeting the threshold support are considered in the Conditional FP Tree.

8) Frequent Patterns are generated from the Conditional FP Tree.

7. Implementation

Implementation is the process of converting a new or revised system design into an operational one. Apart from planning, the major tasks of preparing for implementation or education and training of users. Implementation includes following activities:

Obtaining and installing the system hardware

Providing user access to the system

Creating and updating the database

Training the users on the new system

Documenting the system for its users

Evaluating the operation and use of the system

7.1 Implementation Methods:

There are four basic methods of implementation:

Direct conversion

Parallel conversion

Pilot conversion

Phasing conversion

7.1.1Direct Conversion:

Description:

In this method the new one replaces the old system. This makes organization to fully rely on the new system.

Advantages:

This method forces users to make the new system work. There are immediate benefits from new methods and controls.

Disadvantages:

There is no other system to fall back on if difficulties arise with new system. This method also requires most careful planning.

7.1.2 Parallel Conversion:

Description:

In this method, the old system is operated along with the new system.

Advantages:

This method offers greatest security. The old system can take over if errors are found in the new system or if some usage problems occur.

Disadvantages:

This method doubles operating costs also the new system may not get fair trial.

The present Library Management System has been done by using Parallel Conversion in which the old one is replaced with the newly developed system.

7.1.3 Pilot Conversion:

Description:

In this method the system is implemented to a small part of the organization. Based on the feedback, changes are made and the system is installed in the rest of the organization by one of the other methods.

Advantages:

This method provides experience and live test before implementation.

Disadvantages:

This method may give the impression that the old system is unreliable and not error free.

7.1.4 Phased Conversion:

Description:

In this method, the system is implemented gradually across all users.

Advantages:

It allows some users to take advantages of the system early. It allows training and installing without unnecessary use of resources.

Disadvantages:

A long phase-in causes user problems whether the project goes well (over enthusiasm)

7.2 Source Code:

Index.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

    pageEncoding="ISO-8859-1"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<title>Road Accidents</title>

<!-- Bootstrap CSS -->

<link rel="stylesheet" href="css/bootstrap.css">

<link rel="stylesheet" href="css/font-awesome.min.css">

<link rel="stylesheet" href="css/style.css">

</head>
```

```
<body>
        <header class="header_area">
           <%@include file="title.jsp"%>
           <%@include file="menu.jsp"%>
        </header>
        <section class="choice_area p_120">
                <div class="container">
                        <div class="bgarea">
                        </div></div>
        </section>
        <footer class="footer-area">
           <%@include file="footer.jsp"%>
        </footer>
        <script src="js/bootstrap.min.js"></script>
</body>
</html>
```

Menu.jsp

```
<div class="main_menu">
        <nav class="navbar navbar-expand-lg navbar-light">
        <div class="container">
        <div class="container_inner">
        <div class="collapse navbar-collapse offset" id="navbarSupportedContent">
        <ul class="nav navbar-nav menu_nav">
```

```
<li class="nav-item active"><a class="nav-link
href="${pageContext.request.contextPath}/eclat?query=Home">Home</a></li>

<li class="nav-item"><a class="nav-link”
href="${pageContext.request.contextPath}/eclat?query=View">View</a></li>

<li class="nav-item"><a class="nav-link"

        href="${pageContext.request.contextPath}/eclat?query=Transaction">Freque
nt Items</a></li>

<li class="nav-item"><a class="nav-link"

                href="${pageContext.request.contextPath}/eclat?query=Count">Count

                                                Transactions</a></li>

<li class="nav-item"><a class="nav-link"


        href="${pageContext.request.contextPath}/eclat?query=Support">Minimum
                                                Support</a></li>

                                        </ul>

                                </div>

                        </div>

                </div>

        </nav>

</div>
```

Title.jsp

```
<div class="logo_part">

        <div class="container">

                <div class="title #17a2b8">
```

```
                              <h1>A Coherent Rule Mining Method for Incremental Datasets based on Plausibility</h1>

                    </div>

          </div>

</div>
```

Footer.jsp

```
<div class="container">

          <div class="row footer-bottom d-flex justify-content-between align-items-center">

<p class="col-lg-8 col-md-8 footer-text m-0">Copyright &copy;

<script>document.write(new Date().getFullYear());</script>

All rights reserved | This template is made with

<I class="fa fa-heart-o" aria-hidden="true"></i> by <a href="#"

                                        target="_blank">Colorlib</a>

          </div>

</div>
```

View .jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

          pageEncoding="ISO-8859-1"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html lang="en">

<head>
```

```
<meta charset="utf-8">

<meta name="viewport"

        content="width=device-width, initial-scale=1, shrink-to-fit=no">

<title> FP-growth  Algorithm</title>



<link rel="stylesheet" href="css/bootstrap.css">

<link rel="stylesheet" href="css/font-awesome.min.css">

<link rel="stylesheet" href="css/style.css">

<script src="js/angular.min.js"></script>

<script src="js/Script.js"></script>

</head>

<body>

        <header class="header_area"> <%@include file="title.jsp"%>

        <%@include file="menu.jsp"%> </header>

        <section class="choice_area">

        <div class="container" ng-app="myApp" ng-controller="myController">

                <div class="bgarea1">

                        <div class="main-body">

                                <div class="panel panel-default"

                                        style="margin-left: -61px !important; margin-right:
148px !important;">

                                        <div class="panel-heading" id="heading"
```

```
                                                  style="color: #fff !important; background-
color: #ff1857 !important; width: 119% !important; padding-left: 10px !important;">
                                                  Products DataSet Details</div><div
class="panel-body">
<div class="table-responsive" style="width: 119% !important;">
<table class="table table-striped table-bordered table-hover" id="dataTables-
example">
<thead>
<tr>
<th>Transaction Id</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr class="odd gradeX" ng-repeat="product in products">
<td>{{product.tid}}</td>
<td>{{product.item}}</td>
</tr>
</tbody>
</table>
</div>
</div>
</div>
</div>
```

```
        </div>

        </div>

        </section>

        <footer class="footer-area"> <%@include file="footer.jsp"%>

        </footer>

        <script src="js/bootstrap.min.js"></script>

</body>

</html>
```

Count.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

        pageEncoding="ISO-8859-1"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="viewport"

        content="width=device-width, initial-scale=1, shrink-to-fit=no">

<title> FP-growth Algorithm</title>

<link rel="stylesheet" href="css/bootstrap.css">

<link rel="stylesheet" href="css/font-awesome.min.css">

<link rel="stylesheet" href="css/style.css">

<script src="js/angular.min.js"></script>
```

```
<script src="js/Script.js"></script>

</head>

<body>

<header class="header_area"> <%@include file="title.jsp"%>

<%@include file="menu.jsp"%> </header>

<section class="choice_area">

<div class="container" ng-app="myApp1" ng-controller="myController1">

<div class="bgarea1">

<div class="main-body">

<div class="panel panel-default" style="margin-left: -61px !important; margin-right:
148px !important;">

<div class="panel-heading" id="heading"style="color: #fff !important; background-
color: clr.jpeg !important; width: 119% !important; padding-left: 10px
!important;">Set Support Of Each Items DataSet Details</div><div class="panel-
body">

<div class="table-responsive" style="width: 119% !important;">

<table class="table table-striped table-bordered table-hover" id="dataTables-
example">

<thead>

<tr>

<th>Sl.No</th>

<th>Items</th>

<th>Tid-List</th>

</tr>
```

```
</thead>
<tbody>
<tr class="odd gradeX" ng-repeat="transaction in transactions"><td>{{$index +
1}}</td>
<td>{{transaction.item}}</td>
<td>{{transaction.count}}</td>
</tr>
</tbody>
</table>
</div>
</div>
</div>
</div>
</div>
</div>
</section>
<footer class="footer-area"> <%@include file="footer.jsp"%>
</footer>
<script src="js/bootstrap.min.js"></script>
</body>
</html>
```

Support.jsp

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
          pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html lang="en">
<head>
<!-- Required meta tags -->
<meta charset="utf-8">
<meta name="viewport"
          content="width=device-width, initial-scale=1, shrink-to-fit=no">
<title> FP-growth Algorithm</title>
<!-- Bootstrap CSS -->
<link rel="stylesheet" href="css/bootstrap.css">
<link rel="stylesheet" href="css/font-awesome.min.css">
<link rel="stylesheet" href="css/style.css">
</head>
<body>
          <header class="header_area"> <%@include file="title.jsp"%>
          <%@include file="menu.jsp"%> </header>
          <section class="choice_area">
          <div class="container">
                    <div class="bgarea1">
                              <div class="main-body">
```

```
                <div class="panel panel-default"
                    style="margin-left: -61px !important; margin-right 148px !important;">
                    <div class="panel-heading" id="heading"
                        style="color: #fff !important; background-color: clr.jpeg !important; width: 119% !important; padding-left: 10px !important;">
Minimum Support DataSet Details(Minimum Support Is :
<c:out value="${highNumber}" />)
</div>
<div class="panel-body">
<div class="table-responsive" style="width: 119% !important;">
<table class="table table-striped table-bordered table-hover"id="dataTables-example"><thead>
<tr>
<th>Sl.No</th>
<th>Items</th>
<th>Tid-List</th>
</tr>
</thead>
<tbody>
<c:forEach items="${details}" var="details">
<c:set var="count" value="${count + 1}" scope="page" />
<tr class="odd gradeX">
<td><c:out value="${count}" /></td><td><c:out value="${details.item}" /></td>
```

```
<td><c:out value="${details.count}" /></td>
</tr>
</c:forEach>
</tbody>
</table>
</div>
</div>
</div>
</div>
</div>
</div>
</section>
<footer class="footer-area"> <%@include file="footer.jsp"%></footer>
<script src="js/bootstrap.min.js"></script>
</body>
</html>
```

Transaction.jsp:

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
        pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html lang="en">
<head>
```

```
<meta charset="utf-8">

<meta name="viewport"

        content="width=device-width, initial-scale=1, shrink-to-fit=no">

<title> FP-growth Algorithm</title>



<link rel="stylesheet" href="css/bootstrap.css">

<link rel="stylesheet" href="css/font-awesome.min.css">

<link rel="stylesheet" href="css/style.css">

<script src="js/angular.min.js"></script>

<script src="js/Script.js"></script>

</head>

<body>

        <header class="header_area"> <%@include file="title.jsp"%>

        <%@include file="menu.jsp"%> </header>

        <section class="choice_area">

        <div class="container" ng-app="myApp1" ng-controller="myController1">

                <div class="bgarea1">

                        <div class="main-body">

                                <div class="panel panel-default"

                                        style="margin-left: -61px !important; margin-right:
148px !important;">

                                        <div class="panel-heading" id="heading"
                                        style="color: #fff !important; background-
color: #ff1857 !important; width: 119% !important; padding-left: 10px !important;">

Items  Market Transaction DataSet Details</div>

<div class="panel-body">

<div class="table-responsive" style="width: 119% !important;">

<table class="table table-striped table-bordered table-hover" id="dataTables-
example">

<thead>

<tr><th>Sl.No</th>

<th>Items</th>

<th>Tid-List</th>

</tr>

</thead>

<tbody>

<tr class="odd gradeX" ng-repeat="transaction in transactions">

<td>{{$index + 1}}</td>

<td>{{transaction.item}}</td>

<td>{{transaction.tids}}</td>

</tr>

</tbody>

</table>

</div>

</div>
```

```
                                        </div>
                              </div>
                    </div>
          </div>
          </section>
          <footer class="footer-area"> <%@include file="footer.jsp"%>
          </footer>
          <script src="js/bootstrap.min.js"></script>
</body>
</html>
```

Ecletservlet.java

```
package org.project;
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Date;
import java.util.Iterator;
import java.util.List;
import java.util.Map;
import java.util.Map.Entry;

import java.util.Set;

import java.util.TreeMap;

import java.util.TreeSet;


import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;


public class EclatServlet extends HttpServlet {

        private static final long serialVersionUID = 1L;

        public static List<Products> products = new ArrayList<>();

        public static List<Products> transactions = new ArrayList<>();

        public static List<Products> minimumSupport = new ArrayList<>();

        public static Set<String> items = new TreeSet<>();

        public static Set<Integer> minCount = new TreeSet<>();

        public static String datasetPaths = FilePath.datasetPaths;

        public static String productsFile = FilePath.products;

        public static String transactionsFile = FilePath.transactions;


        protected void doGet(HttpServletRequest request,
                        HttpServletResponse response) throws ServletException,
IOException {
                String folderPath = getServletContext().getRealPath("/dataset");
                String datasetPath = folderPath + datasetPaths;
                String query = request.getParameter("query");
                if ("Home".equals(query)) {
                        try {
                                RequestDispatcher view = request
                                                .getRequestDispatcher("jsp/index.jsp");
                                view.forward(request, response);
                        } catch (Exception ae) {
                                System.out.println(ae);
                        }
                } else if ("Support".equals(query)) {
                        try {
                                minimumSupport.clear();
                                TreeMap<Integer, Integer> m = new TreeMap<>();
                                Iterator itr = minCount.iterator();
                                while (itr.hasNext()) {
                                        int count = 0;
                                        int s = (Integer) itr.next();
                                        for (Products p : transactions) {
                                                if (p.getCount() == s) {
```

```
            count = count + 1;
        }
    }
    m.put(count, s);
}
int highNumber = 0;
Set<Entry<Integer, Integer>> entires = m.entrySet();
for (Entry<Integer, Integer> ent : entires) {
    // System.out.println(ent.getKey() + " ==> " +
    // ent.getValue());
    highNumber = ent.getValue();
}
for (Products p : transactions) {
    if (p.getCount() >= highNumber) {
        Products minsup = new Products();
        minsup.setItem(p.getItem());
        minsup.setCount(p.getCount());
        minimumSupport.add(minsup);
    }
}
request.setAttribute("details", minimumSupport);
request.setAttribute("highNumber", highNumber);
RequestDispatcher view = request
```

```
                .getRequestDispatcher("jsp/Support.jsp");
        view.forward(request, response);
    } catch (Exception ae) {
        System.out.println(ae);
    }
} else if ("Count".equals(query)) {
    try {
        JsonObject.jsonObject(transactions, folderPath,
                transactionsFile);
        RequestDispatcher view = request
                .getRequestDispatcher("jsp/Count.jsp");
        view.forward(request, response);
    } catch (Exception ae) {
        System.out.println(ae);
    }
} else if ("Transaction".equals(query)) {
    try {
        transactions.clear();
        Iterator itr = items.iterator();
        while (itr.hasNext()) {
            String tids = "";
            int count = 0;
            Products transaction = new Products();
```

```
                String s = (String) itr.next();

                for (Products p : products) {

                    if (p.getItem().contains(s)) {

                        int tid = p.getTid();

                        String si = String.valueOf(tid);

                        tids = si + "," + tids;

                        count = count + 1;

                    }

                }

                if (tids.endsWith(",")) {

                    tids = tids.substring(0, tids.length() - 1);

                }

                minCount.add(count);

                transaction.setItem(s);

                transaction.setTids(tids);

                transaction.setCount(count);

                transactions.add(transaction);

            }

            JsonObject.jsonObject(transactions, folderPath,

                    transactionsFile);

            RequestDispatcher view = request


.getRequestDispatcher("jsp/Transaction.jsp");
```

```
                view.forward(request, response);

            } catch (Exception ae) {

                System.out.println(ae);

            }

        } else if ("View".equals(query)) {

            try {

                products.clear();

                String line = "";

                String cvsSplitBy = ",";

                BufferedReader br = new BufferedReader(new
FileReader(

                        datasetPath));

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

                    String[] data = line.split(cvsSplitBy);

                    Products pro = new Products();

                    pro.setTid(Integer.parseInt(data[0]));

                    pro.setItem(data[1] + "," + data[2] + "," + data[3]);

                    products.add(pro);

                    items.add(data[1]);

                    items.add(data[2]);

                    items.add(data[3]);

                }
                JsonObject.jsonObject(products, folderPath,
productsFile);

                RequestDispatcher view = request

                        .getRequestDispatcher("jsp/View.jsp");

                view.forward(request, response);

            } catch (Exception ae) {

                System.out.println(ae);

            }

        }

    }


    protected void doPost(HttpServletRequest request,

            HttpServletResponse response) throws ServletException,
IOException {



    }

}
```

Filepath.java

```
package org.project;

public class FilePath {

    public static String datasetPath="C:\\Users\\Saiprasanna\\Desktop\\";

    public static String datasetPaths="\\Products.csv";

    public static String products="\\Products.json";
```

```
        public static String transactions="\\Transactions.json";

}
```

JsonObject.java

```
package org.project;

import com.fasterxml.jackson.databind.ObjectMapper;

import java.io.File;

import java.io.IOException;

import java.util.List;

import java.util.Set;

public class JsonObject {

        public static void jsonObject(List<Products> accidents,String s,String fname)
{

                ObjectMapper mapper = new ObjectMapper();

                String jsonPath = s+fname;

                File file = new File(jsonPath);

                try {

                        mapper.writeValue(file, accidents);

                } catch (IOException e) {

                        e.printStackTrace();

                }

        }

        public static void jsonObjects(Set<String> states,String s,String fname) {

                ObjectMapper mapper = new ObjectMapper();
                String jsonPath = s+fname;
                File file = new File(jsonPath);
                try {
                        mapper.writeValue(file, states);
                } catch (IOException e) {
                        e.printStackTrace();
                }
        }
        public static void jsonObjectList(List<String> states,String s,String fname) {
                ObjectMapper mapper = new ObjectMapper();
                String jsonPath = s+fname;
                File file = new File(jsonPath);
                try {
                        mapper.writeValue(file, states);
                } catch (IOException e) {
                        e.printStackTrace();
                }
        }
}
```

Products.java

```
package org.project;
public class Products {
        int tid;
        int count;

        String item;

        String tids;

        public final int getTid() {

                return tid;

        }

        public final void setTid(int tid) {

                this.tid = tid;

        }

        public final String getItem() {

                return item;

        }

        public final void setItem(String item) {

                this.item = item;

        }

        public final String getTids() {

                return tids;

        }

        public final void setTids(String tids) {

                this.tids = tids;

        }

        public int getCount() {

                return count;
        }
        public void setCount(int count) {
                this.count = count;
        }
}
```

TimeList.java

```
package org.project;
import java.util.ArrayList;
import java.util.List;
public class TimeList {
        public static List<String> timeList() {
                List<String> timesList = new ArrayList<>();
                double time = 0.00;
                double time1 = 2.00;
                String t1 = String.valueOf(time).replace(".0", ":00");
                String t2 = String.valueOf(time1).replace(".0", ":00");
                timesList.add(t1 + "-" + t2);
                for (int i = 0; i < 11; i++) {
                        time = time + 2;
                        time1 = time1 + 2;
                        t1 = String.valueOf(time).replace(".0", ":00");
                        t2 = String.valueOf(time1).replace(".0", ":00");
                        timesList.add(t1 + "-" + t2);
                }

                return timesList;

        }

}
```

8. System Testing

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies.Assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

8.1 Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

8.2 Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

8.3 Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

8.4 System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

8.5 White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

8.6 Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

8.7 Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

All field entries must work properly.

Pages must be activated from the identified link.

The entry screen, messages and responses must not be delayed.

Features to be tested

Verify that the entries are of the correct format

No duplicate entries should be allowed

All links should take the user to the correct page.

8.8 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

8.9 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

9. Conclusions

We endorse a believability centered manner for affiliation rule mining in regular datasets. This approach utilizes the credibility measure which is a lucidity measure for the mining approach. The two errands related to this procedure are knowledge community development and rule mining by means of the credibility measure. Scan results exhibit the method is educated in the quantity of rules mined.

Appendices

Appendix-A: User Manual

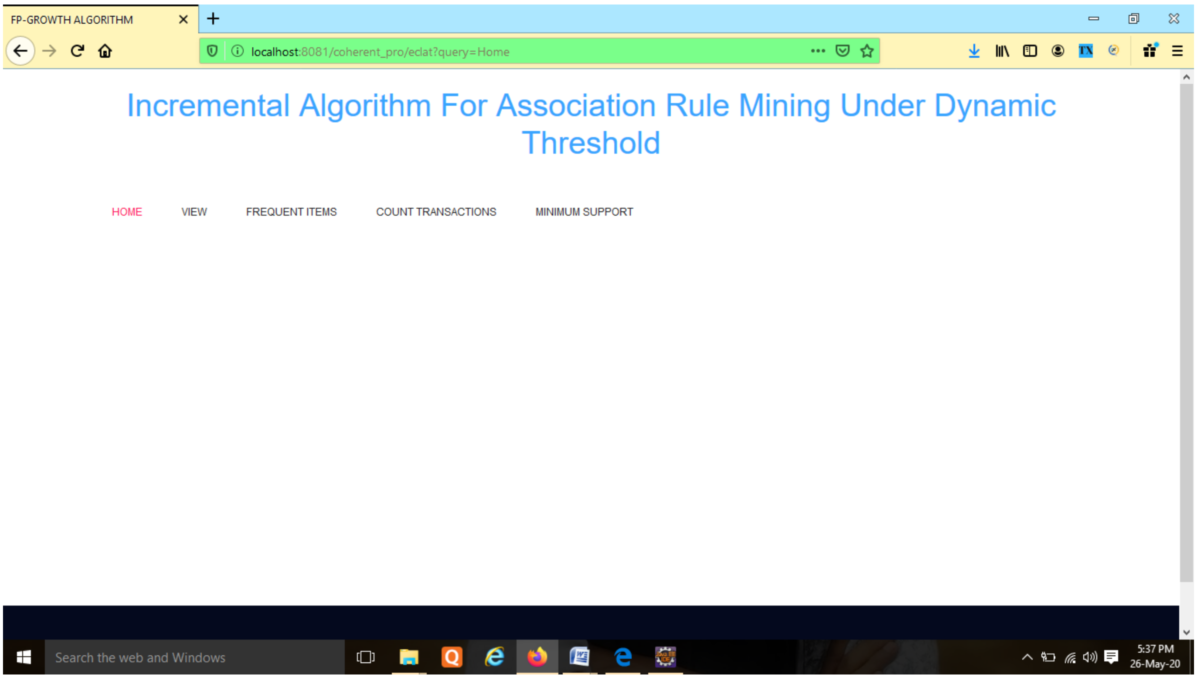
User Manual is the guide to the Users of the system. It paves a path to the Corresponding user to help him how to proceed further in the proper understanding of the system. The interface of the system gets familiar to the user, based on this manual only.

The first form is the user has to upload the CSV file by clicking the choose file then submit. After the submission if it shows successful, then user have to go through pre-processing module, then support and finally go throw the graph.

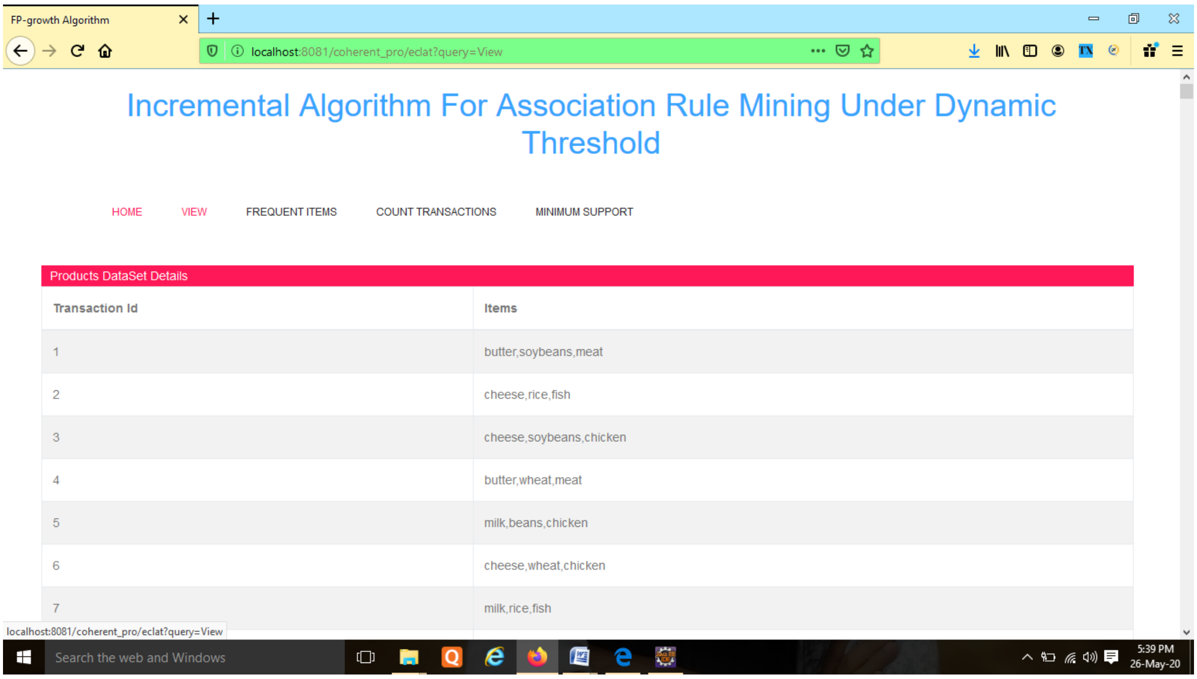
In pre-processing form the user can get the structured output data, after this by clicking support module then user may get the average results of the CSV file.

APPENDIX-B: Test Screens

Home page:


FP-GROWTH ALGORITHM
localhost:8081/coherent_pro/eclat?query=Home
Incremental Algorithm For Association Rule Mining Under Dynamic Threshold
HOME
VIEW
FREQUENT ITEMS
COUNT TRANSACTIONS
MINIMUM SUPPORT
Search the web and Windows
5:37 PM
26-May-20

View page:

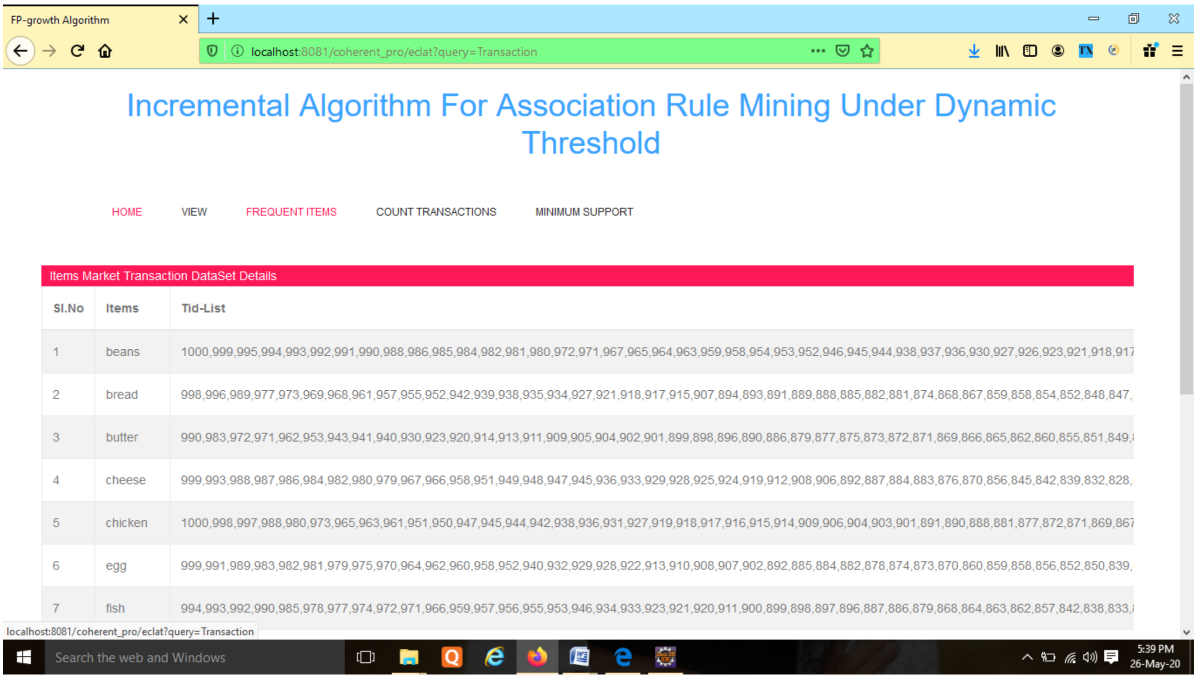

FP-growth Algorithm
localhost:8081/coherent_pro/eclat?query=View
HOME
VIEW
FREQUENT ITEMS
COUNT TRANSACTIONS
MINIMUM SUPPORT
localhost:8081/coherent_pro/eclat?query=View
Search the web and Windows
5:39 PM
26-May-20

Incremental Algorithm For Association Rule Mining Under Dynamic Threshold

Products DataSet Details

Transaction Id	Items
1	butter,soybeans,meat
2	cheese,rice,fish
3	cheese,soybeans,chicken
4	butter,wheat,meat
5	milk,beans,chicken
6	cheese,wheat,chicken
7	milk,rice,fish

Frequent item sets page:

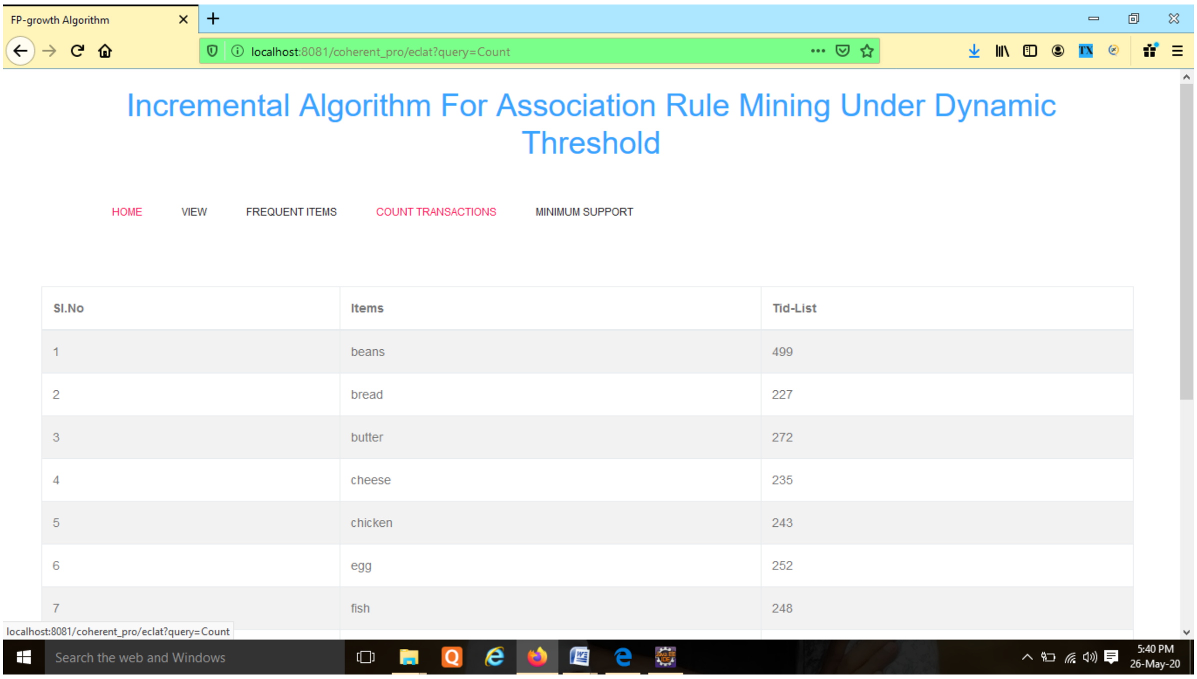

FP-growth Algorithm
localhost:8081/coherent_pro/eclat?query=Transaction
HOME
VIEW
FREQUENT ITEMS
COUNT TRANSACTIONS
MINIMUM SUPPORT
localhost:8081/coherent_pro/eclat?query=Transaction
Search the web and Windows
5:39 PM
26-May-20

Incremental Algorithm For Association Rule Mining Under Dynamic Threshold

Items Market Transaction DataSet Details

Sl.No	Items	Tid-List
1	beans	1000,999,995,994,993,992,991,990,988,986,985,984,982,981,980,972,971,967,965,964,963,959,958,954,953,952,946,945,944,938,937,936,930,927,926,923,921,918,917
2	bread	998,996,989,977,973,969,968,961,957,955,952,942,939,938,935,934,927,921,918,917,915,907,894,893,891,889,888,885,882,881,874,868,867,859,858,854,852,848,847,
3	butter	990,983,972,971,962,953,943,941,940,930,923,920,914,913,911,909,905,904,902,901,899,898,896,890,886,879,877,875,873,872,871,869,866,865,862,860,855,851,849,
4	cheese	999,993,988,987,986,984,982,980,979,967,966,958,951,949,948,947,945,936,933,929,928,925,924,919,912,908,906,892,887,884,883,876,870,856,845,842,839,832,828,
5	chicken	1000,998,997,988,980,973,965,963,961,951,950,947,945,944,942,938,936,931,927,919,918,917,916,915,914,909,906,904,903,901,891,890,888,881,877,872,871,869,867
6	egg	999,991,989,983,982,981,979,975,970,964,962,960,958,952,940,932,929,928,922,913,910,908,907,902,892,885,884,882,878,874,873,870,860,859,858,856,852,850,839,
7	fish	994,993,992,990,985,978,977,974,972,971,966,959,957,956,955,953,946,934,933,923,921,920,911,900,899,898,897,896,887,886,879,868,864,863,862,857,842,838,833,

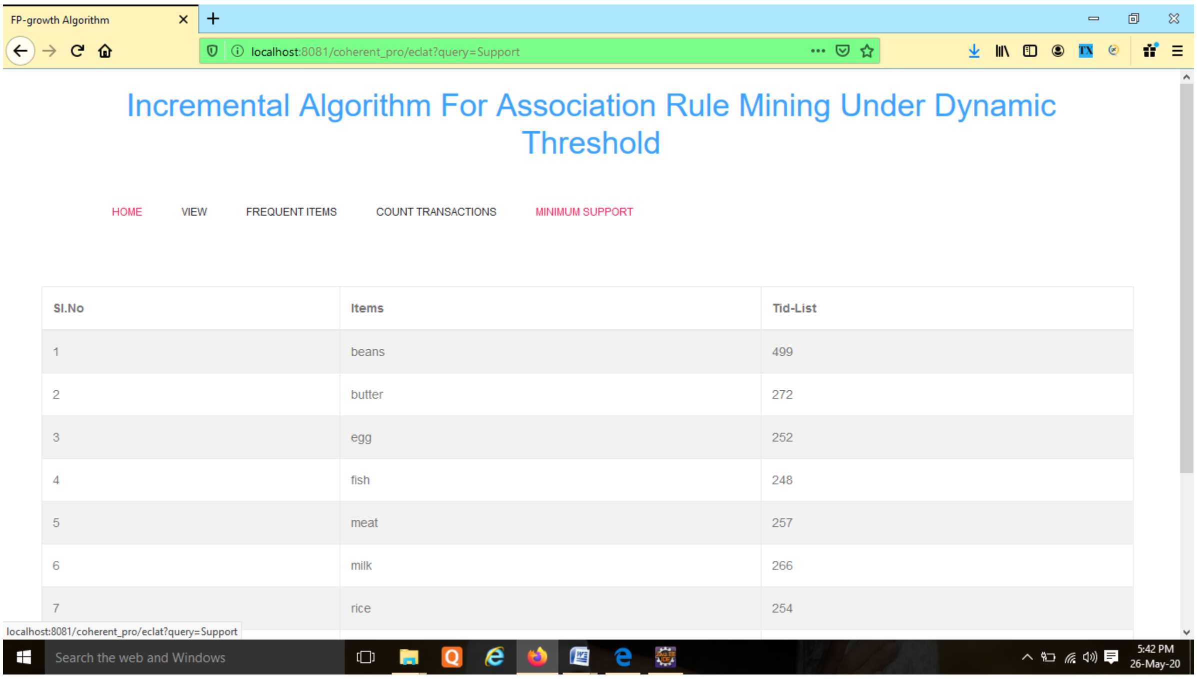
Count transactions page:


FP-growth Algorithm
localhost:8081/coherent_pro/eclat?query=Count
HOME
VIEW
FREQUENT ITEMS
COUNT TRANSACTIONS
MINIMUM SUPPORT
localhost:8081/coherent_pro/eclat?query=Count
Search the web and Windows
5:40 PM
26-May-20

Incremental Algorithm For Association Rule Mining Under Dynamic Threshold

Sl.No	Items	Tid-List
1	beans	499
2	bread	227
3	butter	272
4	cheese	235
5	chicken	243
6	egg	252
7	fish	248

Minimum support page:


FP-growth Algorithm
localhost:8081/coherent_pro/eclat?query=Support
HOME
VIEW
FREQUENT ITEMS
COUNT TRANSACTIONS
MINIMUM SUPPORT
localhost:8081/coherent_pro/eclat?query=Support
Search the web and Windows
5:42 PM
26-May-20

Incremental Algorithm For Association Rule Mining Under Dynamic Threshold

Sl.No	Items	Tid-List
1	beans	499
2	butter	272
3	egg	252
4	fish	248
5	meat	257
6	milk	266
7	rice	254

Bibliography

References

[1] Azadeh Soltani and M.-R. Akbarzadeh-T., “Confabulation-Inspired Association Rule Mining for Rare and Frequent Item sets” IEEE Transactions on Neural Networks And Learning Systems, VOL. 25, NO. 11, November 2014

[2] R. Agrawal and R. Srikant, “Fast algorithms for mining association rules in large databases”, in Proc. 20th Int. Conf. Very Large Data Bases, San Francisco, CA, USA, 1994, pp. 487499.

[3] B. Liu, W. Hsu, and Y. Ma, “Mining association rules with multiple minimum supports”, in Proc. 5th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 1999, pp. 337341.

[4] A. Ghosh and B. Nath, “Multi-objective rule mining using genetic algorithms”, Inf. Sci., vol. 163, nos. 13, pp. 123133 [5] B. Nath, D. Bhattacharyya, and A. Ghosh,“Discovering association rules from incremental datasets”, IJCSC, vol. 1, no. 2, pp. 433441, 2010

[6] J. Han, J. Pei, Y. Yin, and R. Mao, “Mining frequent patterns without candidate generation: A frequent-pattern tree approach, Data Mining” Knowl. Discovery, vol. 8, no. 1, pp. 5387, 2004. [7] M. J. Zaki and K. Gouda, “Fast vertical mining using diffsets”, in Proc. 9th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2003, pp. 326335.

[8] M. J. Zaki, “Scalable algorithms for association mining”, IEEE Trans. Knowl. Data Eng., vol. 12, no. 3, pp. 372390, Jun.2000.

[9] J. Pei, J. Han, H. Lu, S. Nishio, S. Tang, and D.Yang, “H-mine: Hyperstructure mining of frequent patterns in large databases”, in Proc. IEEE Int. Conf. Data Mining, Feb. 2001, pp. 441448.

[10] L. Gu, J. Li, H. He, G. Williams, S. Hawkins, and C. Kelman, “Association rule discovery with unbalanced class Distributions, in Proc. 16th Austral. Joint Conf. Artif. Intell., 2003, pp. 221232.