1. INTRODUCTION

Reviews from consumers are very important information in E-commerce systems. Many online shops have developed reviews system for users to post their reviews. With the rapid development of social networking media, more and more people are willing to share their feelings, opinions and suggestions on their bought items with their friends or even strangers in social network applications or E-commerce systems. These reviews can be very useful for people's decision making in many different scenarios such as users' preference mining and personalized recommendation.

At present, more and more review mining based applications are being applied to make our decision process easier than before. These applications have greatly changed people's behavior patterns, especially in E-commerce activities. For example, WHEN people want to buy Product, book a hotel or restaurant, they usually not only ask for advice from their friends but also refer to reviews available online. To adapt to this change, many famous Ecommerce companies, such as Amazon, eBay and Taobao (China), have built up wellfunction consumer review systems.

Online experience from various people can help one make decisions. In this case, people and their experience are required to be trusted by others. It makes sense that we usually ask for advice from our friends or family members before we make a decision. But the question is, why individuals are inclined to rely on strangers in cyber space to make decision? Scholars find a primary reason for that is their lack of trust in companies that they only experience through the web medium.

The virtual nature of the web medium challenges traditional understanding of customer trust. In E-commerce scenario, customers have no chance to have a face-to-face interaction with a sales man or a direct physical experience with the store and the products they want to buy. On one hand, their experience is mediated through the web which is a twodimensional graphical display. They usually feel somewhat lost and need someone to give them advices. On the other hand, reviews from consumers who purchase an item have direct physical experiences with it, are seem to be more reliable than vendor's promotions or advertising words.

However, E-commerce websites usually accumulate large scale text based reviews which records historical commentary about one subject or item. Usually, consumers are unable to distinguish which reviews can be trusted under so large information. Different consumers can hold different aspects and standpoints in viewing things. And their attitudes, interests, preferences, etc. will vary greatly towards the products or services. Some users give a positive rating because they like certain attributes of the product, while others give a negative rating because they don't like these attributes. Therefore, it is impossible for a consumer to judge whose reviews are suitable and which users can be trusted. The consumers

urgently need to be established a trust between other users, which give their view she can trusts, provide him with an opinion reference, and shield the untrusted comments to prevent misleading to the user when he wants to purchase an item. Many scholars have spent much effort on the phenomenon of trust relationship between strangers in E-commerce environment and found an interesting result: people are more willing to trust the individuals who are similar with them in as many respects as possible.

The similarity factors include the brought items, the sentiment style of reviews, the words used, etc. There are many studies trying to explore the relationship between people's mutual trust and their similarity quantitatively, and find that there is a strong correlation between both trust and interest similarity. Although there is a certain relationship between the trust of users and the similarity of users, this relationship is not an obvious linear relationship between trust and similarity, and it also includes many other influential factors. How to correctly find the relationship between trust and similarity still faces great challenges.

Due to its human-related properties, trust is difficult to be uniformly defined or even precisely described. The vast majority of existing studies focused on trust construction and maintenance between customers and companies over time and after repeated experiences. While limited effort is spent on trust between consumers and potential consumers in Ecommerce systems. Obviously, in the field of E-commerce reviews, people are more concerned about the credibility of reviews and the trust of user who post the reviews. In our work, we aim to investigate trust between users in E-commerce systems quantitatively by exploring their reviews and evaluations regarding to various

2. LITERATURAL SURVEY

TITLE: TruCom: Exploiting domain-specific trust networks for multicategory item recommendation.

Author: H. Liu, F. Xia, Z. Chen, N. Y. Asabere, J. Ma, and R. Huang.

Recommender systems (RSs) have become important tools for solving the problem of information overload. With the advent and popularity of online social networks, some studies on network-based recommendation have emerged, raising the concern of many researchers. Trust is one kind of important information available in social networks and is often used for performance improvement in social-network-based RSs. However, most trust-aware RSs ignore the fact that people trust different subsets of friends pertaining to different domains, such as music and movies, because people behave differently in diverse domains according to different interests. This paper proposes a novel recommendation method called TruCom. In a multicategory item recommendation domain, TruCom first generates a domain-specific trust network pertaining to each domain and then builds a unified objective function for improving recommendation accuracy by incorporating the hybrid information of direct and indirect trust into a matrix factorization recommendation model. Through relevant benchmark experiments on two real-world data sets, we show that TruCom achieves better performance than other existing recommendation methods, which demonstrates the effectiveness and reliability of TruCom.

TITLE: Effects of sentiment on recommendations in social network.

Author: P.-Y. Hsu, H.-T. Lei, S.-H. Huang, T. H. Liao, Y.-C. Lo, and C.-C. Lo.

This study adopted a sentiment word database to extract sentiment-related data from microblog posts. These data were then used to investigate the effect of different types of sentiment-related words on product recommendations. The results indicate that posts containing strong sentiments received more clicks than posts containing neutral sentiments. Posts containing more than one positive sentiment word generate more effective recommendations than posts containing only one positive sentiment word. This study also demonstrated that posts with a negative polarity classification received more clicks than those with a positive polarity classification. Additionally, the microblog posts containing implicit sentiment words received more clicks than those containing explicit sentiment words. The findings presented here could assist product or service marketers who use Plurk or similar microblogging platforms better focus their limited financial resources on potential online customers to achieve maximum sale revenue.

Title: The joint beta distribution with refund rate in online C2C trust building: A theoretical study on Taobao.

Author: C. Qin, W. Siyi, and A. Lin.

Lack of trust is one of the fundamental reasons for losing customers from “faceless” ecommerce websites in the consumer-to-consumer market. In order to combat problems with dishonest market participants, a reputation system based on the trustworthiness of sellers has been widely established by the service provider. Traditionally this trust model is mainly based on feedback mechanism while neglecting the other important factors, such as, refund rate, which may lead to transaction risks in the new transaction. This paper proposes a global reputation rating method by using joint beta probability density functions to combine positive feedback rating and refund rate. The new trust model has the advantage of tractability and scalability as well as its theoretical sound basis on statistics.

3. SYSTEM ANALYSIS

3.1 Existing System:

All this information may involve consumers' views on things that can express interest, sentiments, and opinions. Many kinds of research have shown that people are more likely to trust each other with the same attitude toward similar things. In this paper, we consider seeking and accepting sentiments and suggestions in E-commerce systems somewhat implies a form of trust between consumers during shopping. Following this view of point, an E-commerce system reviews mining oriented sentiment similarity analysis approach is put forward to exploring users' similarity and their trust. We divide the trust into two categories, namely direct trust, and propagation of trust, which represents a trust relationship between two individuals. The direct trust degree is obtained from sentiment similarity

3.2 Proposed System:

We present an entity-sentiment word pair mining method for similarity feature extraction. The propagation of trust is calculated according to the transitivity feature. Using the proposed trust representation model, we use the shortest path to describe the tightness of trust and put forward an improved shortest path algorithm to figure out the propagation trust relationship between users. A large-scale E-commerce website reviews dataset is collected to examine the accuracy of the algorithms and feasibility of the models. The experimental results indicate that the sentiment similarity analysis can be an efficient method to find trust between users in E-commerce systems.

4. SYSTEM REQUIREMENTS

4.1 HARDWARE REQUIREMENTS:

- Processor - minimum i3.
- RAM - minimum 4 GB.
- Hard Disk - 20 GB

4.2 SOFTWARE REQUIREMENTS:

- Operating System : Windows 8
- Application Server : Tomcat8
- Front End : HTML, Java, Jsp
- Database : Mysql
- Database Connectivity : JDBC.

5. SYSTEM STUDY

FRAMEWORKSTUDY

PRACTICABLENESS STUDY

The utility of the task is stone-broke down during this stage ANd business proposition is advanced with an exceptionally broad arrangement for the endeavor and a few price gauges. Amid framework examination the believability investigation of the projected framework is to be done. this is often to ensure that the projected framework is not a weight to the organization. For possibility examination, some comprehension of the many conditions for the framework is basic.

Three key contemplations related to the likelihood examination area unit

1. ECONOMICAL PRACTICABLENESS
2. TECHNICAL PRACTICABLENESS
3. SOCIAL PRACTICABLENESS
4. PRACTICAL PRACTICABLENESS

This examination is completed to examine the money impact that the framework can wear the association. The live of reserve that the organization will fill the innovative work of the framework is restricted. The consumptions should be advocated. Consequently the created framework too within the financial statement and this was accomplished in lightweight of the actual fact that the overwhelming majority of the advances used area unit uninhibitedly accessible. simply the altered things should be bought.

SPECIALIZED PRACTICABLENESS

This examination is completed to examine the specialised possibility, that is, the specialised requirements of the framework. Any framework created should not have AN attractiveness on the accessible specialized assets. this may prompt levels of recognition on the accessible specialised assets. this may prompt levels of recognition being placed on the client. The created framework should have AN retiring requirement, as simply negligible or invalid changes area unit needed for actualizing this framework.

SOCIAL PRACTICABLENESS

The part of study is to examine the amount of acknowledgment of the framework by the shopper. This incorporates the method toward making ready the shopper to utilize the framework effectively. The shopper should not feel debilitated by the framework, rather ought to acknowledge it as a necessity. the amount of acknowledgment by the purchasers solely depends upon the ways that area unit used to show the shopper regarding the framework and to form him conversant in it. His level of certainty should be raised therefore he's likewise able to build some useful feedback, that is invited, as he's the last shopper of the framework.

6. SYSTEM DESIGN

SYSTEM ARCHITECTURE

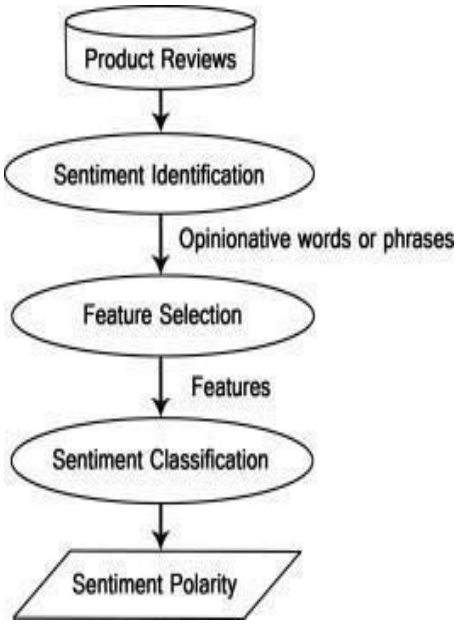


FIG :1 System architecture

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; or associated with, 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:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process. 4.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

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.

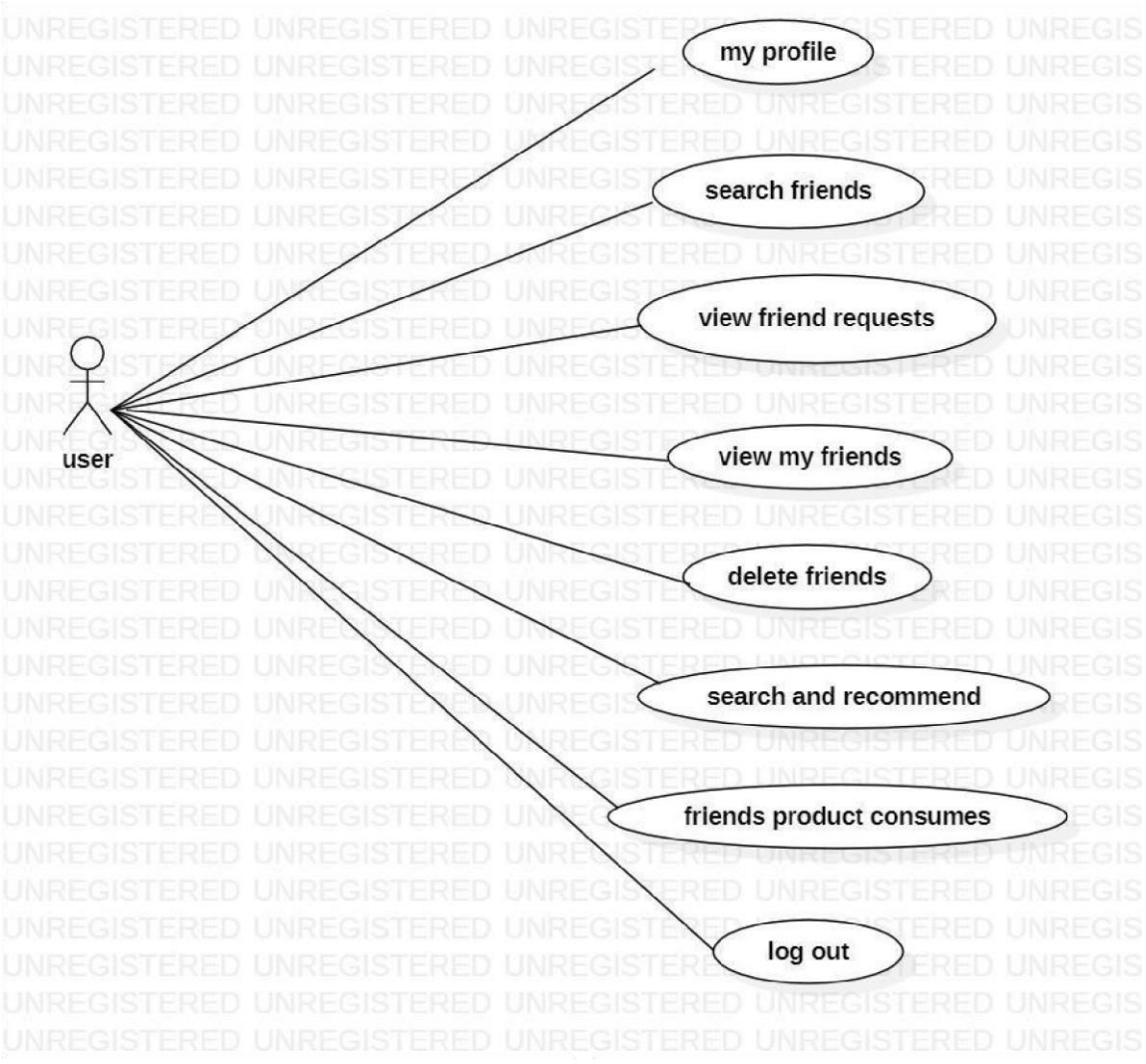


Fig2. Use Case Diagram(1)

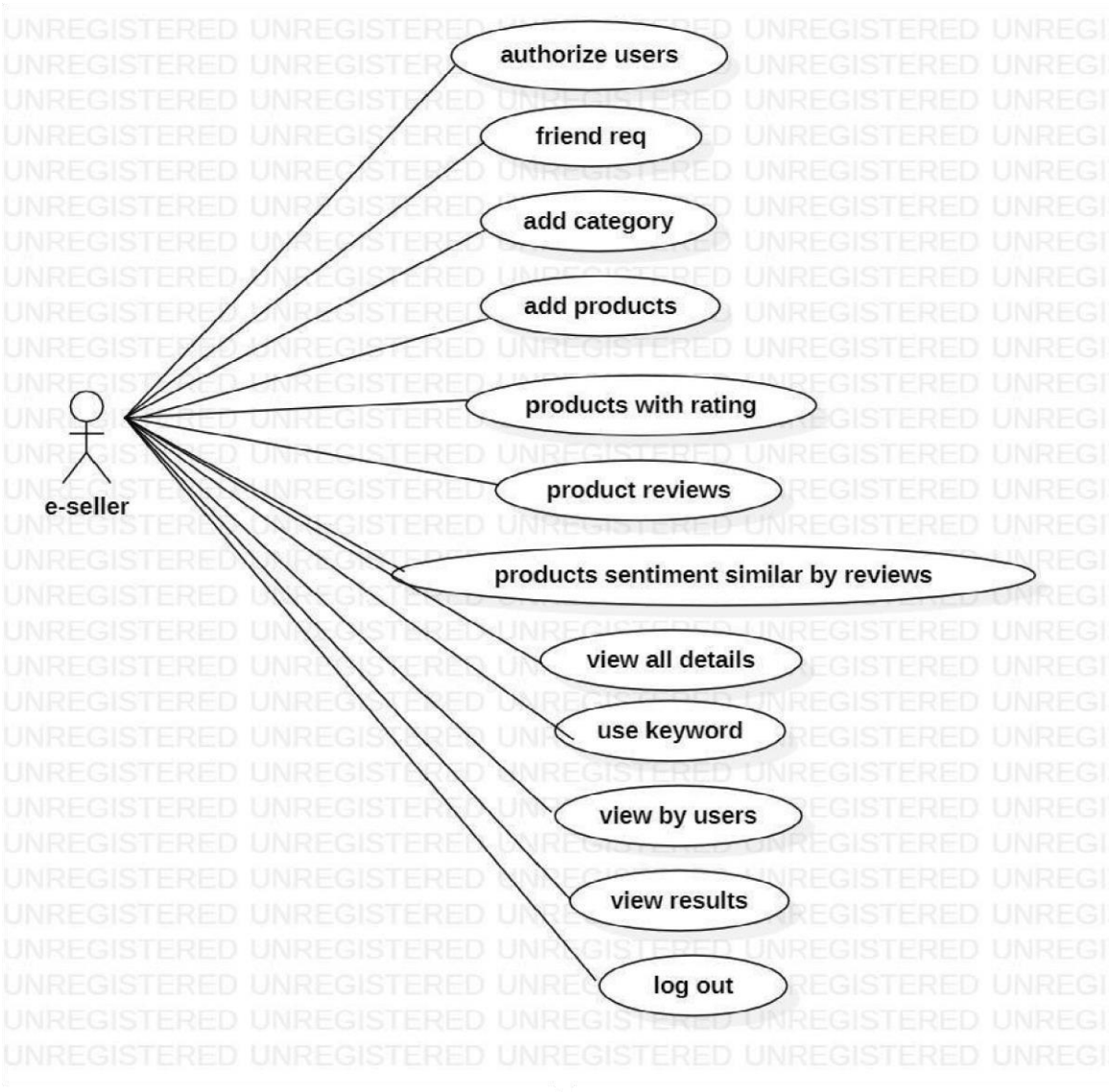


Fig3. Use Case Diagram(2)

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.

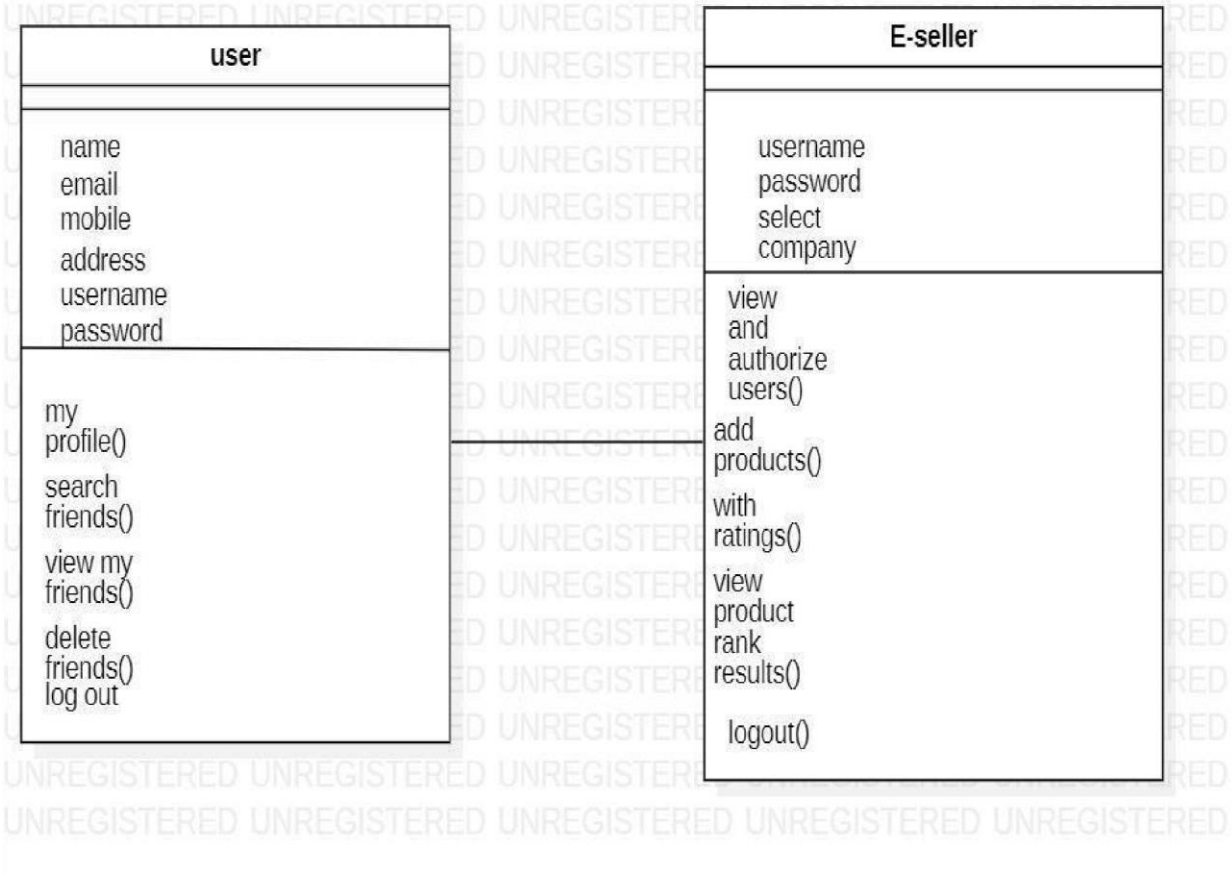


Fig4. Class Diagram

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, and timing diagrams.

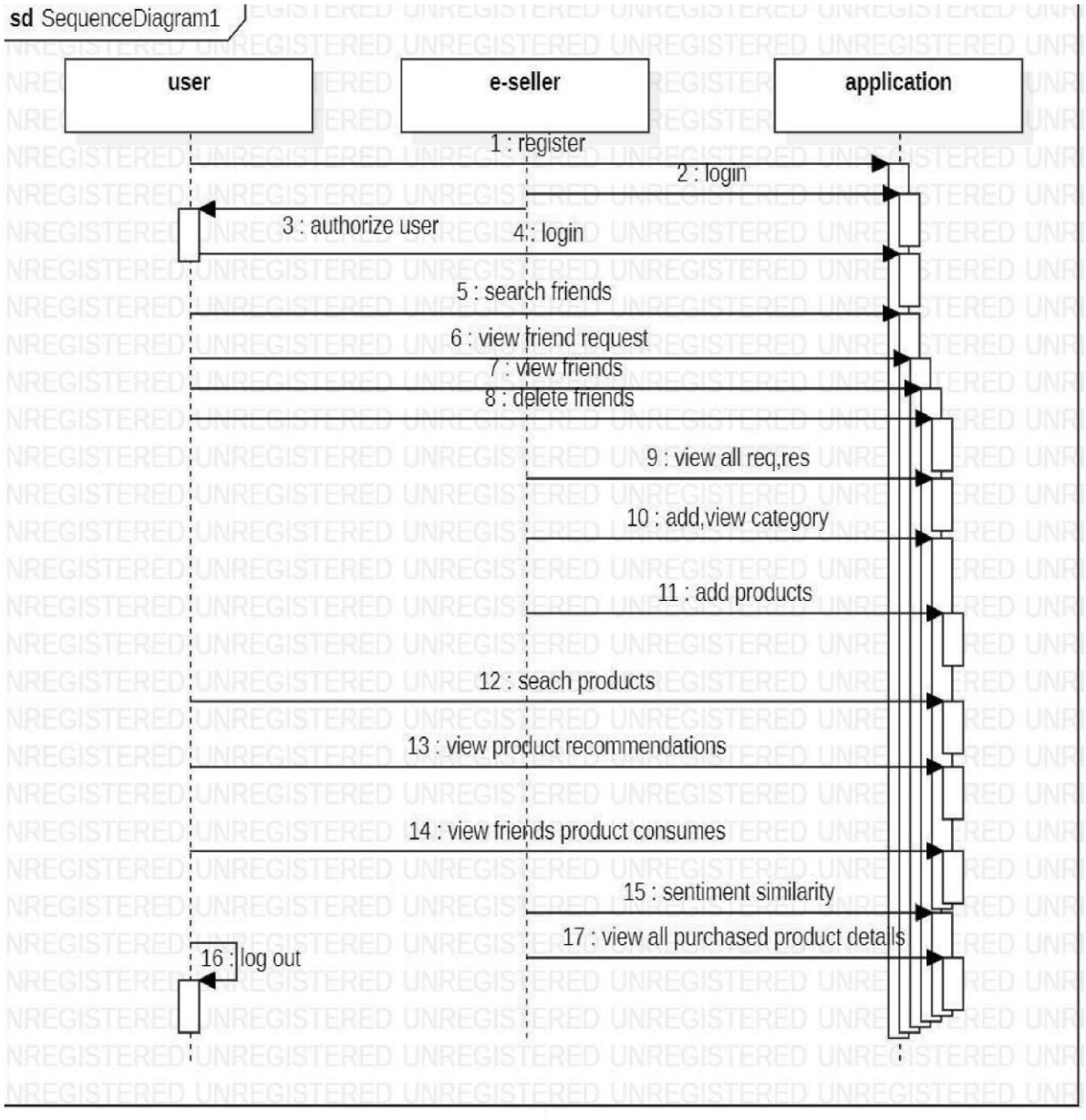


Fig5. Sequence Diagram

7. IMPLEMENTATION

MODULES

1. User
2. E-seller Module Description

User:

Here the user one of the module and the user should register with the application after the registration the user must be authorized by the e-seller admin. Then only the user can able login with the application.

After the user successful login, the user can able to perform the following operations such as

- My Profile ,Search Friends
- View Friend Requests
- View My Friends
- Delete My Friends
- Search Products And Recommend
- View Post Recommend
- Friends Products Consumes

E-seller:

Here the e-seller is the main module and e-seller can login directly with the application and after the successful login the e-seller can able perform the some operations such as

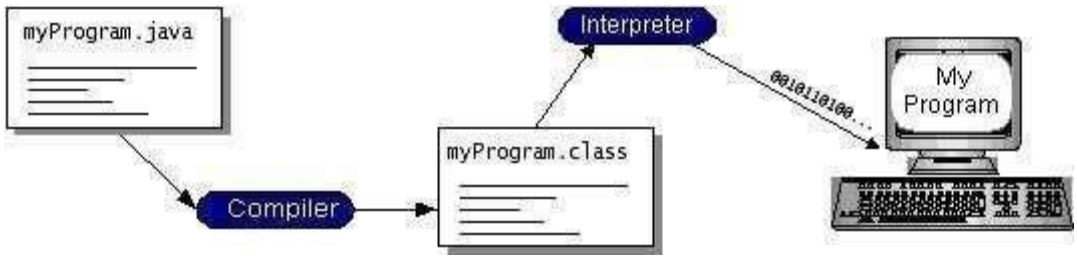
- View And Authorized Users
- All Friend Request/Response
- Add and View Categories
- Add Products
- View All Products With Ratings
- View All Products Reviews
- View All Products Sentiment Similarly By Reviews
- View All Purchased Product Details
- View User Query Keyword
- View All Products Consumes By Users
- View All Recommended Products
- View Product Rank Results

8. SOFTWARE ENVIRONMENT

JAVA TECHNOLOGY:

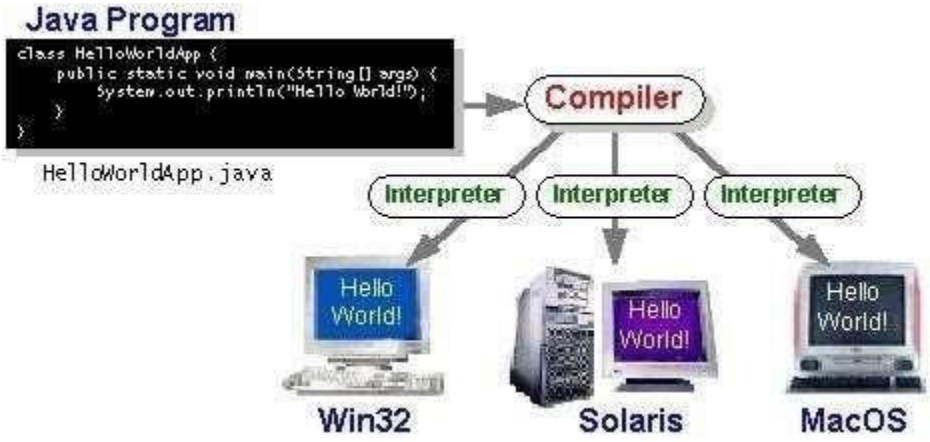
- Java Technology Is Both A Programming Language And A Platform.
- The Java Programming Language
- The Java Programming Language Is A High-Level Language
- That Can Be Characterized By All Of The Following Buzzwords:
    1. Simple
    2. Architecture neutral
    3. Object oriented
    4. Portable
    5. Distributed
    6. High performance
    7. Interpreted
    8. Multithreaded
    9. Robust
    10. Dynamic
    11. 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.



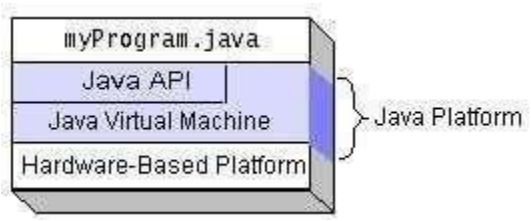
THE JAVA PLATFORM:

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:

1. The Java Virtual Machine (Java VM)
2. 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, welltuned 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 Javaenabled 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.

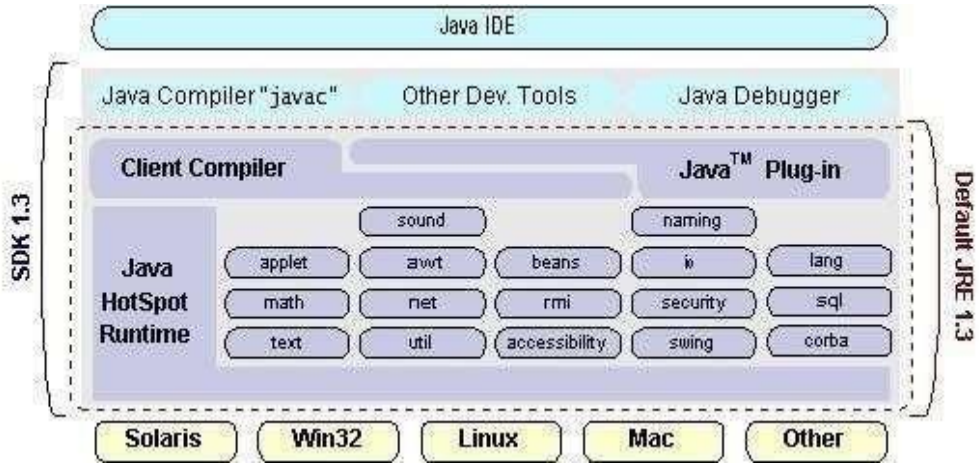
An application is a standalone program that runs directly on the Java platform. A special kind of application known as a server serves and supports clients on a network. Examples of servers are Web servers, proxy servers, mail servers, and print servers. Another specialized program is a servlet. A servlet 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™, can plug into existing component architectures.
- **Object serialization:** Allows lightweight persistence and communication via Remote Method Invocation (RMI).
- **Java Database Connectivity (JDBC™):** 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

wideranging, 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™ 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. There is a 16-bit and a 32-bit version of this program and each maintains a separate list of ODBC data sources.

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:

1. 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.

2. 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.

3. JDBC

Must be implemental on top of common database interfaces. The JDBC SQL API must "sit" on top of other common SQL level APIs.

4. 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.

5. 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.

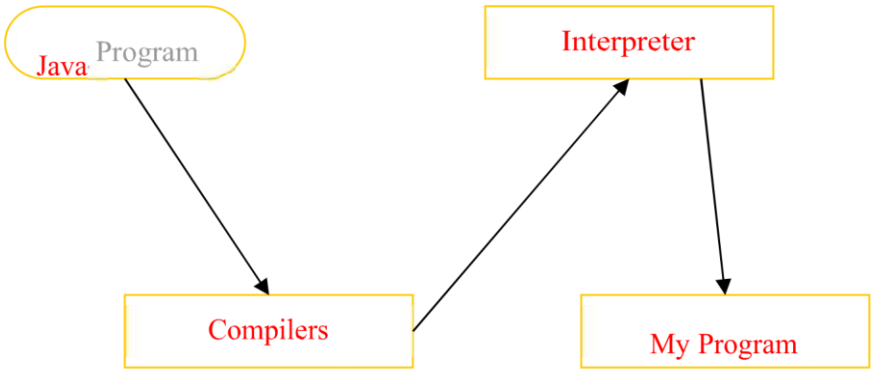
6. **Use strong, static typing wherever possible**

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

7. **Keep the common cases simple**

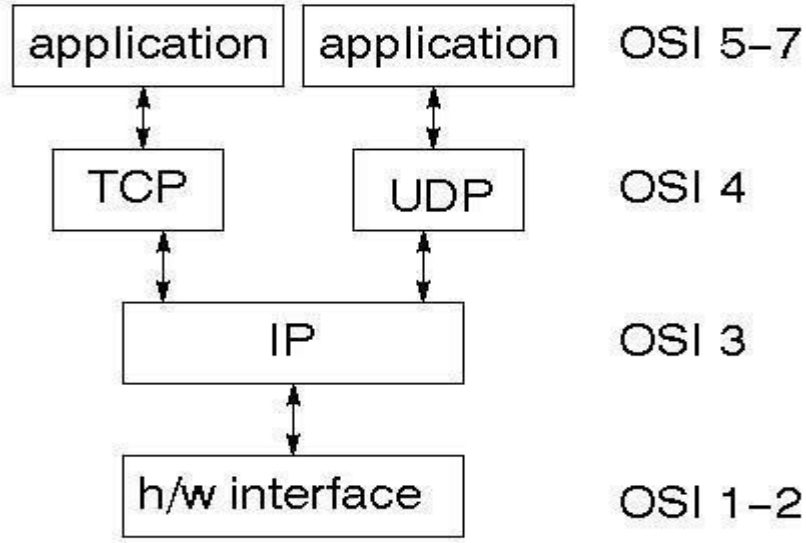
Because more often than not, the usual SQL calls used by the programmer are simple SELECT's, INSERT's, DELETE's and UPDATE's, these queries should be simple to perform with JDBC. However, more complex SQL statements should also be possible.

Finally we decided to proceed the implementation using Java Networking.



NETWORKING:

TCP/IP STACK: The TCP/IP stack is shorter than the OSI one:



TCP is a connection-oriented protocol; UDP (User Datagram Protocol) is a connectionless protocol.

IP DATAGRAM'S

The IP layer provides a connectionless and unreliable delivery system. It considers each datagram independently of the others. Any association between datagram must be supplied by the higher layers. The IP layer supplies a checksum that includes its own header. The header includes the source and destination addresses. The IP layer handles routing through an Internet. It is also responsible for breaking up large datagram into smaller ones for transmission and reassembling them at the other end.

UDP

UDP is also connectionless and unreliable. What it adds to IP is a checksum for the contents of the datagram and port numbers. These are used to give a client/server model - see later.

TCP

TCP supplies logic to give a reliable connection-oriented protocol above IP. It provides a virtual circuit that two processes can use to communicate.

INTERNET ADDRESSES

In order to use a service, you must be able to find it. The Internet uses an address scheme for machines so that they can be located. The address is a 32 bit integer which gives the IP address. This encodes a network ID and more addressing. The network ID falls into various classes according to the size of the network address.

NETWORK ADDRESS

Class A uses 8 bits for the network address with 24 bits left over for other addressing.
Class B uses 16 bit network addressing.
Class C uses 24 bit network addressing
Class D uses all 32.

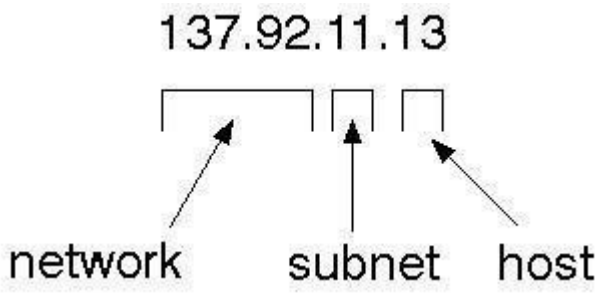
SUBNET ADDRESS

Internally, the UNIX network is divided into sub networks. Building 11 is currently on one sub network and uses 10-bit addressing, allowing 1024 different hosts.

HOST ADDRESS

8 bits are finally used for host addresses within our subnet. This places a limit of 256 machines that can be on the subnet.

TOTAL ADDRESS



The 32 bit address is usually written as 4 integers separated by dots.

PORT ADDRESSES

A service exists on a host, and is identified by its port. This is a 16 bit number. To send a message to a server, you send it to the port for that service of the host that it is running on. This is not location transparency! Certain of these ports are "well known".

SOCKETS

A socket is a data structure maintained by the system to handle network connections. A socket is created using the call socket. It returns an integer that is like a file descriptor. In fact, under Windows, this handle can be used with Read File and Write File functions.

```
#include <sys/types.h>

#include <sys/socket.h>

int socket(int family, int type, int protocol);
```

Here "family" will be AF_INET for IP communications, protocol will be zero, and type will depend on whether TCP or UDP is used. Two processes wishing to communicate over a network create a socket each. These are similar to two ends of a pipe - but the actual pipe does not yet exist.

JFree Chart :-

JFreeChart is a free 100% Java chart library that makes it easy for developers to display professional quality charts in their applications. JFreeChart's extensive feature set includes:

- A consistent and well-documented API, supporting a wide range of chart types;
- A flexible design that is easy to extend, and targets both server-side and client-side applications;
- Support for many output types, including Swing components, image files (including PNG and JPEG), and vector graphics file formats (including PDF, EPS and SVG);
- JFreeChart is "open source" or, more specifically, free software. It is distributed under the terms of the GNU Lesser General Public Licence (LGPL), which permits use in proprietary applications.

1. MAP VISUALIZATIONS

Charts showing values that relate to geographical areas. Some examples include: (a) population density in each state of the United States, (b) income per capita for each country in Europe, (c) life expectancy in each country of the world.

2. TIME SERIES CHART INTERACTIVITY

Implement a new (to JFreeChart) feature for interactive time series charts --- to display a separate control that shows a small version of ALL the time series data, with a sliding "view" rectangle that allows you to select the subset of the time series data to display in the main chart.

3. DASHBOARDS

There is currently a lot of interest in dashboard displays. Create a flexible dashboard mechanism that supports a subset of JFreeChart chart types (dials, pies, thermometers, bars, and lines/time series) that can be delivered easily via both Java Web Start and an applet.

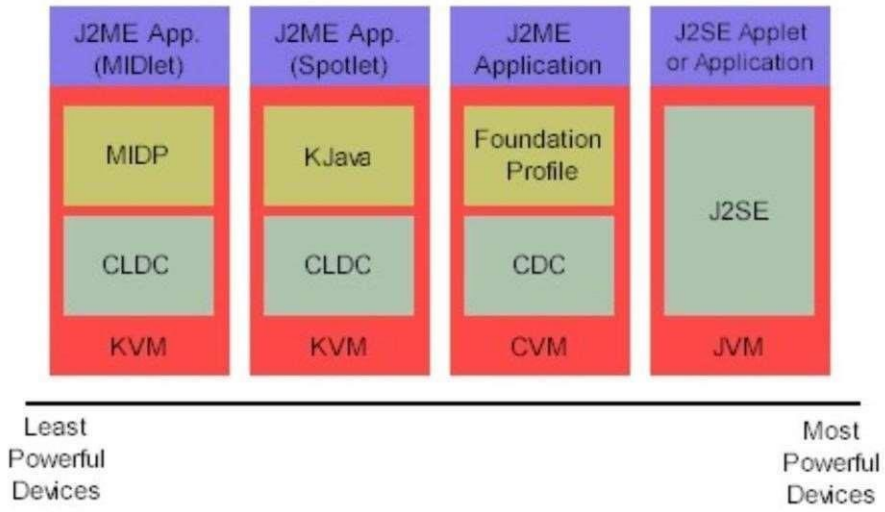
4. PROPERTY EDITORS

The property editor mechanism in JFreeChart only handles a small subset of the properties that can be set for charts. Extend (or reimplement) this mechanism to provide greater end-user control over the appearance of the charts.

J2ME (Java 2 Micro edition):-

Sun Microsystems defines J2ME as "a highly optimized Java run-time environment targeting a wide range of consumer products, including pagers, cellular phones, screenphones, digital set-top boxes and car navigation systems." Announced in June 1999 at the JavaOne Developer Conference, J2ME brings the cross-platform functionality of the Java language to smaller devices, allowing mobile wireless devices to share applications. With J2ME, Sun has adapted the Java platform for consumer products that incorporate or are based on small computing devices.

1. General J2ME Architecture



J2ME uses configurations and profiles to customize the Java Runtime Environment (JRE). As a complete JRE, J2ME is comprised of a configuration, which determines the JVM used, and a profile, which defines the application by adding domain-specific classes. The configuration defines the basic run-time environment as a set of core classes and a specific JVM that run on specific types of devices. We'll discuss configurations in detail in the The profile defines the application; specifically, it adds domain-specific classes to the J2ME configuration to define certain uses for devices. We'll cover profiles in depth in the The following graphic depicts the relationship between the different virtual machines, configurations, and profiles. It also draws a parallel with the J2SE API and its Java virtual machine. While the J2SE virtual machine is generally referred to as a JVM, the J2ME virtual machines, KVM and CVM, are subsets of JVM. Both KVM and CVM can be thought of as a kind of Java virtual machine -- it's just that they are shrunken versions of the J2SE JVM and are specific to J2ME.

2. Developing J2ME applications

Introduction In this section, we will go over some considerations you need to keep in mind when developing applications for smaller devices. We'll take a look at the way the compiler is invoked when using J2SE to compile J2ME applications. Finally, we'll explore packaging and deployment and the role preverification plays in this process.

3. Design considerations for small devices

Developing applications for small devices requires you to keep certain strategies in mind during the design phase. It is best to strategically design an application for a small device before you begin coding. Correcting the code because you failed to consider all of the "gotchas" before developing the application can be a painful process. Here are some design strategies to consider:

4. Configurations overview

The configuration defines the basic run-time environment as a set of core classes and a specific JVM that run on specific types of devices. Currently, two configurations exist for J2ME, though others may be defined in the future:

- Connected Limited Device Configuration (CLDC) is used specifically with the KVM for 16-bit or 32-bit devices with limited amounts of memory. This is the configuration (and the virtual machine) used for developing small J2ME applications. Its size limitations make CLDC more interesting and challenging (from a development point of view) than CDC. CLDC is also the configuration that we will use for developing our drawing tool application. An example of a small wireless device running small applications is a Palm hand-held computer.

- Connected Device Configuration (CDC) is used with the C virtual machine (CVM) and is used for 32-bit architectures requiring more than 2 MB of memory. An example of such a device is a Net TV box.

5. J2ME profiles

What is a J2ME profile?

As we mentioned earlier in this tutorial, a profile defines the type of device supported. The Mobile Information Device Profile (MIDP), for example, defines classes for cellular phones. It adds domain-specific classes to the J2ME configuration to define uses for similar devices. Two profiles have been defined for J2ME and are built upon CLDC: KJava and MIDP. Both KJava and MIDP are associated with CLDC and smaller devices. Profiles are

built on top of configurations. Because profiles are specific to the size of the device (amount of memory) on which an application runs, certain profiles are associated with certain configurations.

A skeleton profile upon which you can create your own profile, the Foundation Profile, is available for CDC.

Profile 1: KJava

KJava is Sun's proprietary profile and contains the KJava API. The KJava profile is built on top of the CLDC configuration. The KJava virtual machine, KVM, accepts the same byte codes and class file format as the classic J2SE virtual machine. KJava contains a Sunspecific API that runs on the Palm OS. The KJava API has a great deal in common with the J2SE Abstract Windowing Toolkit (AWT). However, because it is not a standard J2ME package, its main package is com.sun.kjava. We'll learn more about the KJava API later in this tutorial when we develop some sample applications.

Profile 2: MIDP

MIDP is geared toward mobile devices such as cellular phones and pagers. The MIDP, like KJava, is built upon CLDC and provides a standard run-time environment that allows new applications and services to be deployed dynamically on end user devices. MIDP is a common, industry-standard profile for mobile devices that is not dependent on a specific vendor. It is a complete and supported foundation for mobile application development. MIDP contains the following packages, the first three of which are core CLDC packages, plus three MIDP-specific packages.

* java.lang
* java.io
* java.util
* javax.microedition.io
* javax.microedition.lcdui
* javax.microedition.midlet
* javax.microedition.rms

9. SYSTEM TEST

FRAMEWORK TESTING

The motivation behind testing is to seek out blunders. Testing is that the approach toward making an attempt to seek out every attainable blame or disadvantage in an exceedingly work item. It offers associate degree approach to examine the utility of segments, sub congregations, gatherings or doubtless a completed item it's the approach toward active programming with the expectation of guaranteeing that the Programming framework lives up to its stipulations associate degreed consumer wishes and doesn't bomb in an disappointing approach. There area unit differing types of take a look at every take a look at compose addresses a selected testing necessity.

SORTS OF TESTS:-

UNIT TESTING:-

Unit testing includes the arrange of experiments that approve that the inner program principle is functioning fitly, which program inputs deliver substantial yields. All alternative branches and within code stream got to be approved. It's the attempting of individual programming units of the applying .it is done once the fulfillment of a personal unit before connexion. This can be a basic testing, that depends on data of its development and is obtrusive. Unit take a look at perform elementary tests at half level and test a selected business method, application, furthermore as framework setup. Unit tests guarantee that each extraordinary approach of a business procedure performs exactly to the according details and contains clearly characterised inputs and expected outcomes.

COMBINATION TESTING:-

Combination tests area unit meant to check coordinated programming segments to make a decision whether or not they very keep running united program. Testing is occasion driven and is additional troubled regarding the essential results of screens or fields. Reconciliation tests exhibit that in spite of the very fact that the components were solely fulfillment, as appeared by effectively unit testing, the combination of segments is correct and reliable. Incorporation testing is especially gone for uncovering the problems that emerge from the combination of segments.

UTILITARIAN:-

Utilitarian tests provide economical showings that capacities tried area unit accessible as determined by the business and specialised wants, framework documentation, and consumer manuals.

Utilitarian testing is fixated on the related to things:

Legitimate Input : distinguished categories considerable data should be acknowledged.

Invalid Input : distinguished categories of invalid data should be rejected.

Capacities : distinguished capacities should be figured out.

Yield : distinguished categories of utilization yields should be figured out.

Association and readiness of helpful tests is focused around wants, key capacities, or distinctive experiments. What is additional, economical inclusion regarding distinguish Business method streams; data fields, predefined forms, and progressive procedures should be thought-about for testing. Before sensible testing is finished, additional tests area unit recognized and therefore the powerful estimation of current tests is resolved.

FRAMEWORK:-

Framework testing guarantees that the entire incorporated programming framework meets stipulations. It tests a style to ensure better-known and expected outcomes. A case of framework take a look at ing is that the style organized framework incorporation test. Framework testing depends on method portrayals and streams, underlining predriven method connections and incorporation focuses.

WHITEBOX TESTING:-

White Box Testing may be attempting within which within which the merchandise instrument is aware of regarding the inner workings, structure and nonstandard speech of the merchandise, or presumably its motivation. It's reason. It’s used to check regions that cannot be come back to from a discovery level.

DISCOVERY TESTING:-

Discovery Testing are going to be attempting the merchandise with no data of the inner workings, structure or non-standard speech of the module being tried. Discovery tests, as most differing types of tests, should be composed from a conclusive supply report,

for instance, detail or wants record, for instance, determination or stipulations archive. It's a attempting within which the merchandise below take a look at is restrained, as a discovery. You can't "see" into it. The take a look at offers knowledge sources and reacts to yields while not considering however the merchandise functions.

UNIT TESTING:

Unit testing is often directed as a significant side of a joined code and unit trial period of the merchandise lifecycle, despite the very fact that it's not extraordinary for writing and unit testing to be junction rectifier as 2 clear stages.

•Verify that the sections area unit of the proper configuration

•No copy passages got to be permissible

•All connections ought to take the consumer to the proper page.

INTEGRATION TESTING:-

Programming reconciliation testing is that the progressive connexion testing of a minimum of 2 coordinated programming segments on a solitary stage to deliver disappointments caused by interface abandons.

The trip of the incorporation take a look at is to look at that components or programming applications, e.g. segments in an exceedingly product framework or – one stage up – programming applications at the organization level – connect while not mistake. Test outcomes: All the experiments aforementioned higher than passed effectively. No deformities veteran.

ACCEPTANCE TESTING:-

Client Acceptance Testing may be a basic amount of any task and needs Brobdingnagian cooperation by the top consumer. It likewise guarantees that the framework meets the utilitarian stipulations.

Test outcomes: All the experiments specified higher than passed effectively. No deformities veteran.

10. SAMPLE CODE

```
<%--
    Document   : AdminMain
    Author     : user
--%>


<%@page import="java.sql.ResultSet"%>
<%@page import="java.sql.Statement"%>
<%@page import="java.sql.DriverManager"%>
<%@page import="java.sql.Connection"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>   <%
String name=request.getParameter("name"); session.setAttribute("name",name);  String
pwd=request.getParameter("password"); String select=request.getParameter("role");
session.setAttribute("role",select); try{       Class.forName("com.mysql.jdbc.Driver");
Connection  con=DriverManager.getConnection("jdbc:mysql://localhost:3306/mute",
"root", "root");
    Statement st=con.createStatement();
    ResultSet rs=st.executeQuery("select * from admin where name='"+name+"' and
password='"+pwd+"'");     if(rs.next()){                                %>
        <script type="text/javascript">            window.alert("login success");
window.location="AdminHome.jsp";
        </script>
        <%            }  else{
%>
        <script type="text/javascript">            window.alert("login failed");
window.location="AdminLogin.jsp";
          </script>
        <%
}
}
```

```
catch(Exception   e){
out.println(e); }   %>
```

UserMain.jsp

```
<%--
   Document   : UserMain
   Author     : user
--%>


<%@page import="java.sql.ResultSet"%>
<%@page import="java.sql.Statement"%>
<%@page import="java.sql.DriverManager"%>
<%@page import="java.sql.Connection"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>   <html>
   <head>
      <title>Users Login/Register</title>
      <style>                        body{                        background-image:
url("images/backgrondimages.jpg");                  background-repeat: no-repeat;
background-size: cover;                        }          h2{              color: blue;
}                    #home ul li {              float: right;              width: 150px;
list-style: none;              margin-top:
20px;                              }          a:hover{
color: whitesmoke;             font-size: 30px;
backgroundcolor:
activecaption;
            }             a{
text-decoration: none;
            }          img{              margintop:
40px;
         }
```

```
    </style>
  </head>
  <body>
  <center><h2>Mining Users Trust From E Commerce Reviews Based On
Sentiment Similarity Analysis</h2></center>
  <div id="home">
    <ul>
      <li><a href="AdminLogin.jsp">ESeller</a></li>
      <li><a href="UserLogin.jsp">Users</a></li>
      <li><a href="index.html">Home</a></li>
    </ul>
  </div>
   <div id="img">
    <img src="images/e commerce.png" alt="" width="1340" height="500">
  </div>
  <%
  String name=request.getParameter("username");      session.setAttribute("username",
name);                String    pwd=request.getParameter("password");                try{
Class.forName("com.mysql.jdbc.Driver");                                        Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/mute",  "root", "root");
   Statement st=con.createStatement();
   ResultSet rs=st.executeQuery("select * from user where username='"+name+"' and
password='"+pwd+"'");     if(rs.next()){
       String status=rs.getString("status");         if(status.equals("waiting")){     %>
           <script type="text/javascript">              window.alert("Your not
a authorized person");              window.location="UserLogin.jsp";
           </script>
      <%
         }else{
             String mail=rs.getString("email");              session.setAttribute("email",mail);
String   id=rs.getString("id");                    session.setAttribute("id",id);
%>
          <script type="text/javascript">              window.alert("Login Successful.....!!");
```

```
            window.location="User_Home.jsp";
           </script>
        <% }
      }else{
        response.sendRedirect("UserLogin.jsp?msg=Login Failed Check  Username/password");
      }    }    catch(Exception e){        out.println(e);
}
  %>
</body>
</html>
```

viewProfile.jsp

```
<%--
    Document   : User_Profile
    Created on : Jan 20, 2020, 6:25:05 PM
    Author     : user
--%>

<%@page import="java.sql.ResultSet"%>
<%@page import="java.sql.Statement"%>
<%@page import="java.sql.Connection"%>
<%@page import="java.sql.DriverManager"%>
<%@page contentType="text/html" pageEncoding="UTF-
8"%>   <html>
    <head>
        <title>User Home</title>
        <style>             body{               background-image:
url("images/backgrondimages.jpg");                  background-repeat: no-repeat;
background-size: cover;
            }           h2{
                color: blue;
            }
            #home ul li a:hover{                color: whitesmoke;
```

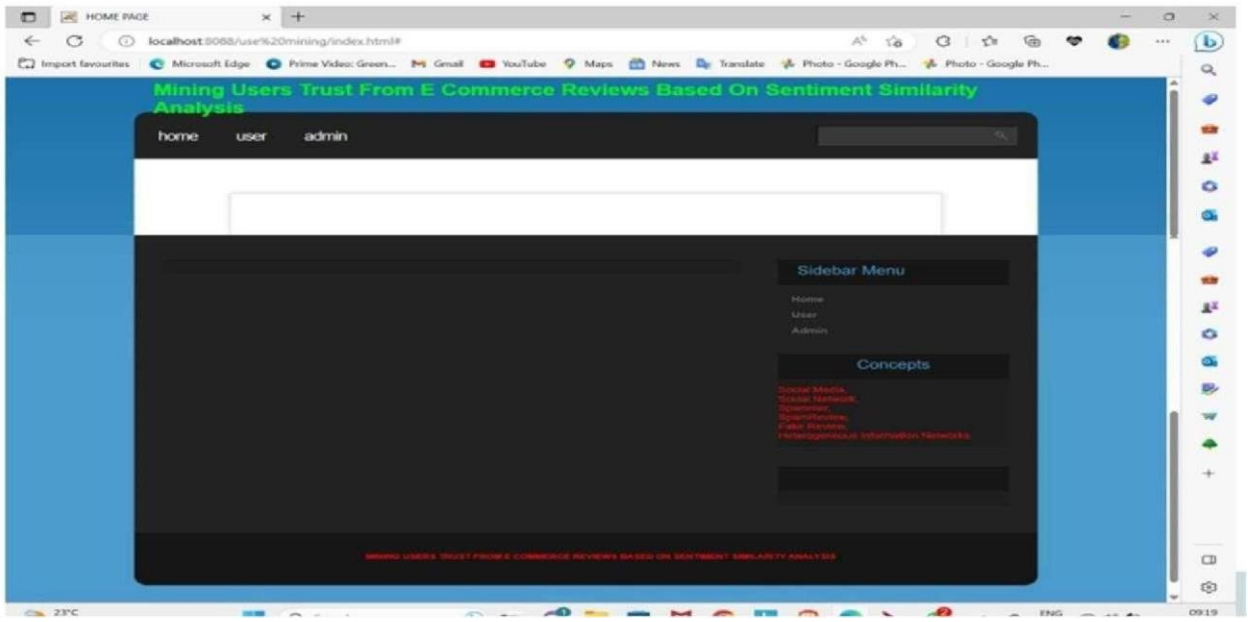
```
            font-size: 30px;                background-color:
activecaption;          }
        #home ul li {              float: right;              width:
150px;                        list-style:  none;
margin-top: 20px;
}            img{              margintop: 40px;                        }
         #sidebar ul li{                        list-style: none;
         }             a{              text-decoration: none;
         }           table{               border-color: white;
         }           tr td{               color:blue;
         }           tr th{              color: #66ffff;              background-color: deeppink;


         }
      </style>
   </head>
   <body>
   <center><h2>Mining Users Trust From E Commerce Reviews Based On
Sentiment Similarity Analysis</h2></center>
   <div id="home">
      <ul>
        <li><a href="">ESeller</a></li>
        <li><a href="">Users</a></li>
        <li><a href="index.html">Home</a></li>
      </ul>        </div>
   <div id="img">
      <img src="images/e commerce.png" alt="" width="1340" height="500">    </div>
        <% String name=(String)session.getAttribute("username");%>
```

```
        <h2><b><center> User <%=name%>'s Profile</center></b></h2>
        <h3>Sidebar Menu</h3>
        <div id="sidebar">
        <ul>
            <li><a href="User_Profile.jsp">Home</a></li>
            <li><a href="User_Home.jsp">LogOut</a></li>
             </ul>
        </div>
        <center>
        <table border="5">
        <tr>
        <th>Image</th>
        <th>E-mail</th>
        <th>mobile</th>
        <th>Address</th>
        <th>Date Of Birth</th>
        <th>status</th>
</tr>           <%        try{
        Class.forName("com.mysql.jdbc.Driver");     Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/mute",  "root", "root");
   Statement st=con.createStatement();
   ResultSet rs=st.executeQuery("select * from user where username='"+name+"'");
while(rs.next()){
   %>
   <tr>
<td><image src="view.jsp?name=<%=rs.getString("username")%>" width="50" height="50"></td>
<td><%=rs.getString("email")%></td>
<td><%=rs.getString("mobile")%></td>
<td><%=rs.getString("address")%></td>
<td><%=rs.getString("dateofbirth")%></td>
<td><%=rs.getString("status")%></td>
</tr>
<%
      }  }         catch(Exception e){            out.println(e);
}
      %>
```

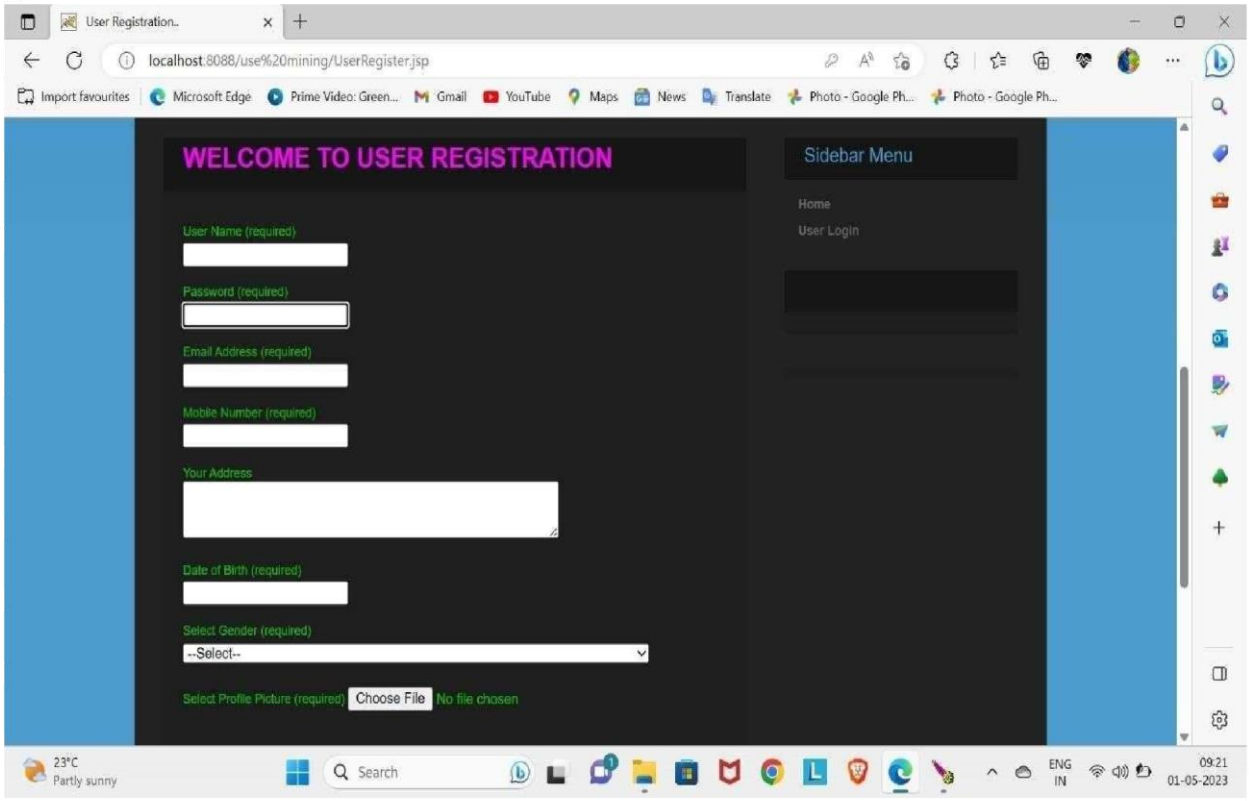
```
        </table>
  </center>
        <a href="User_Home.jsp"><center>Back</center></a>
    </body>
</html>
```

11. SCREEN SHOTS

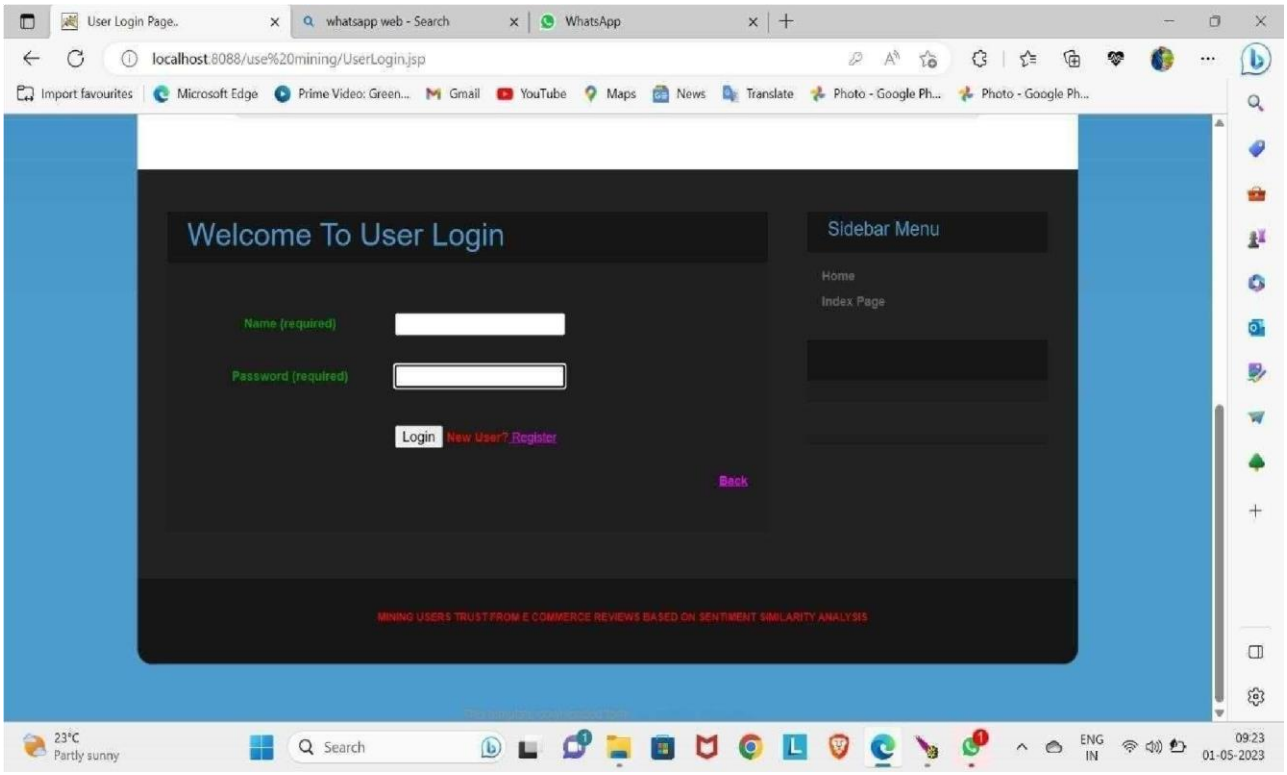
1.HOME SCREEN:



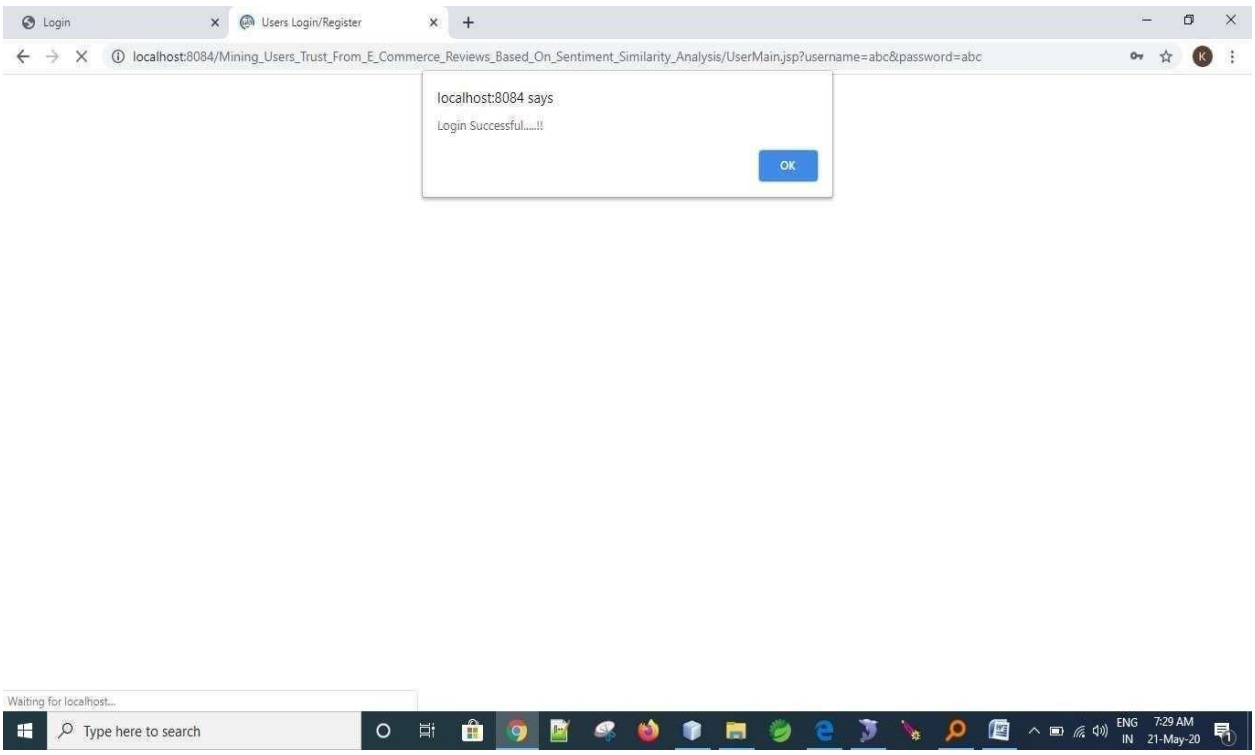
2.USER REGISTRATION



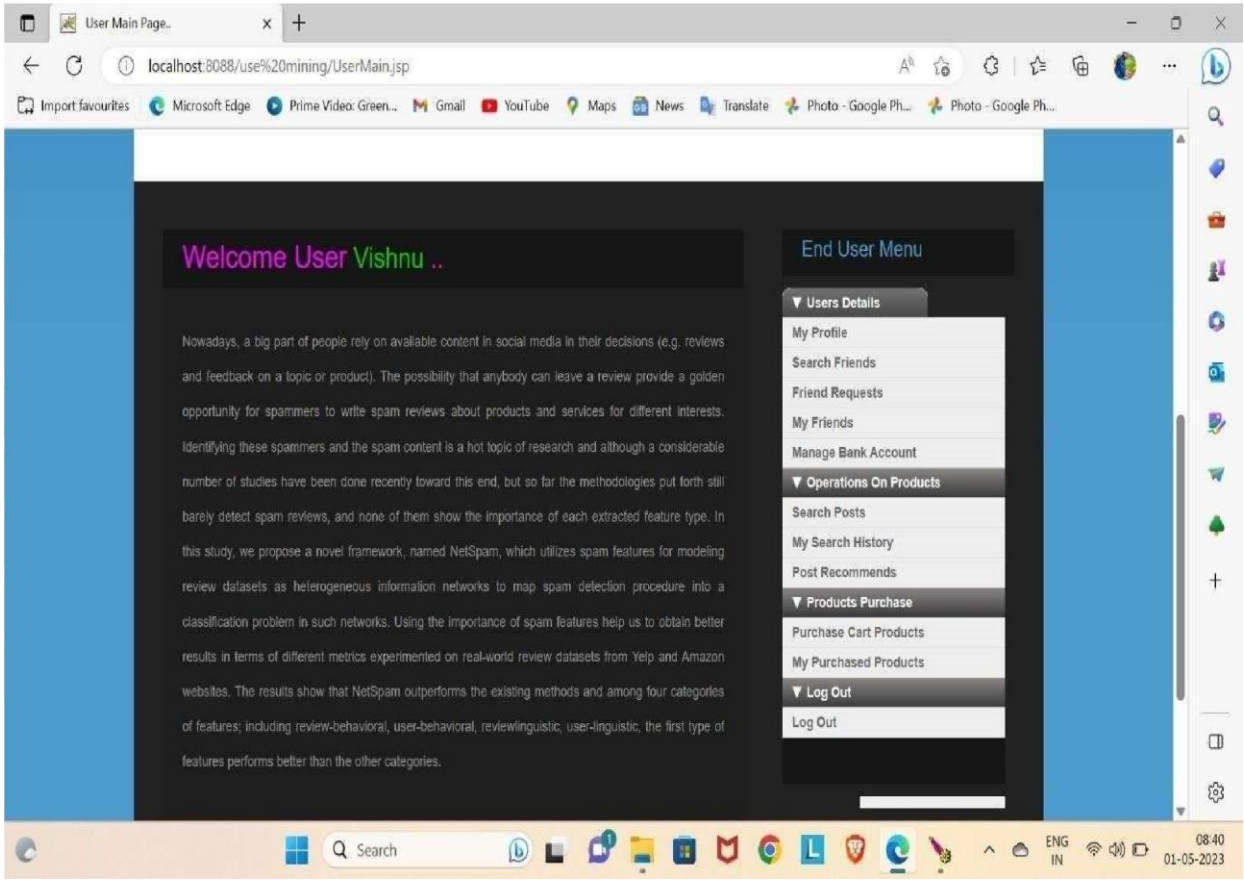
3. USER LOGIN



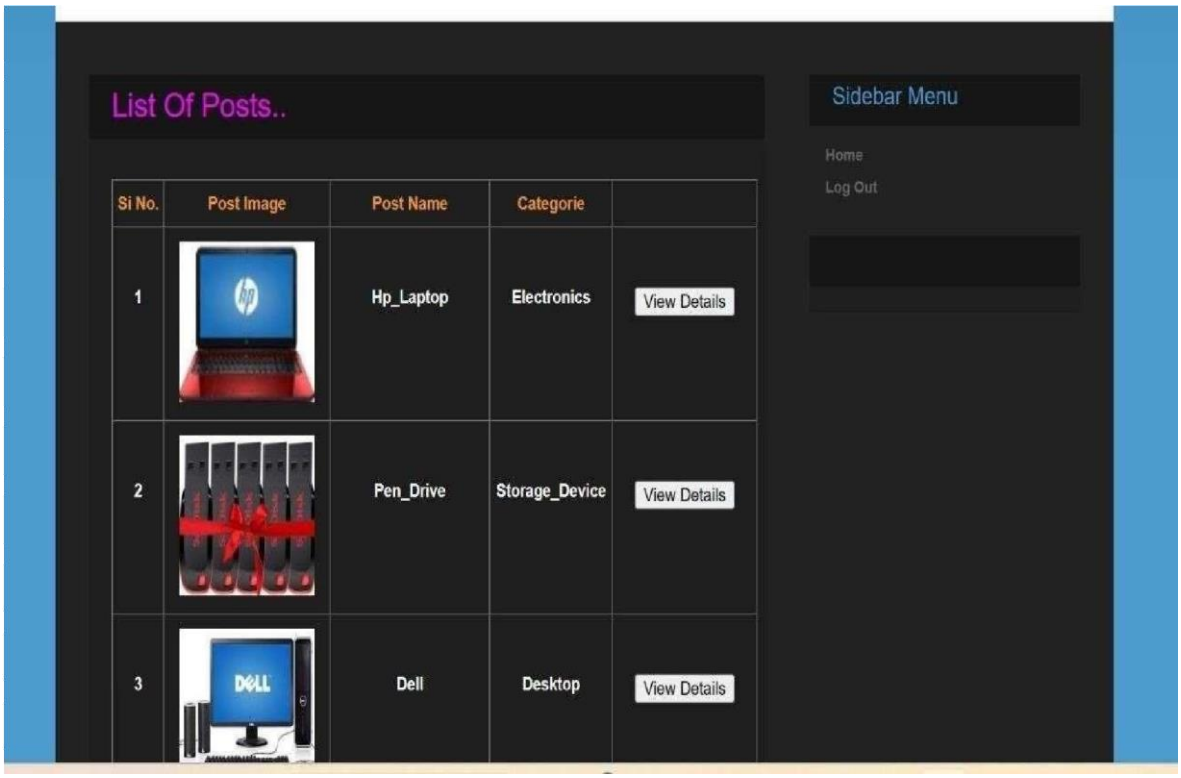
4. LOGIN STATUS



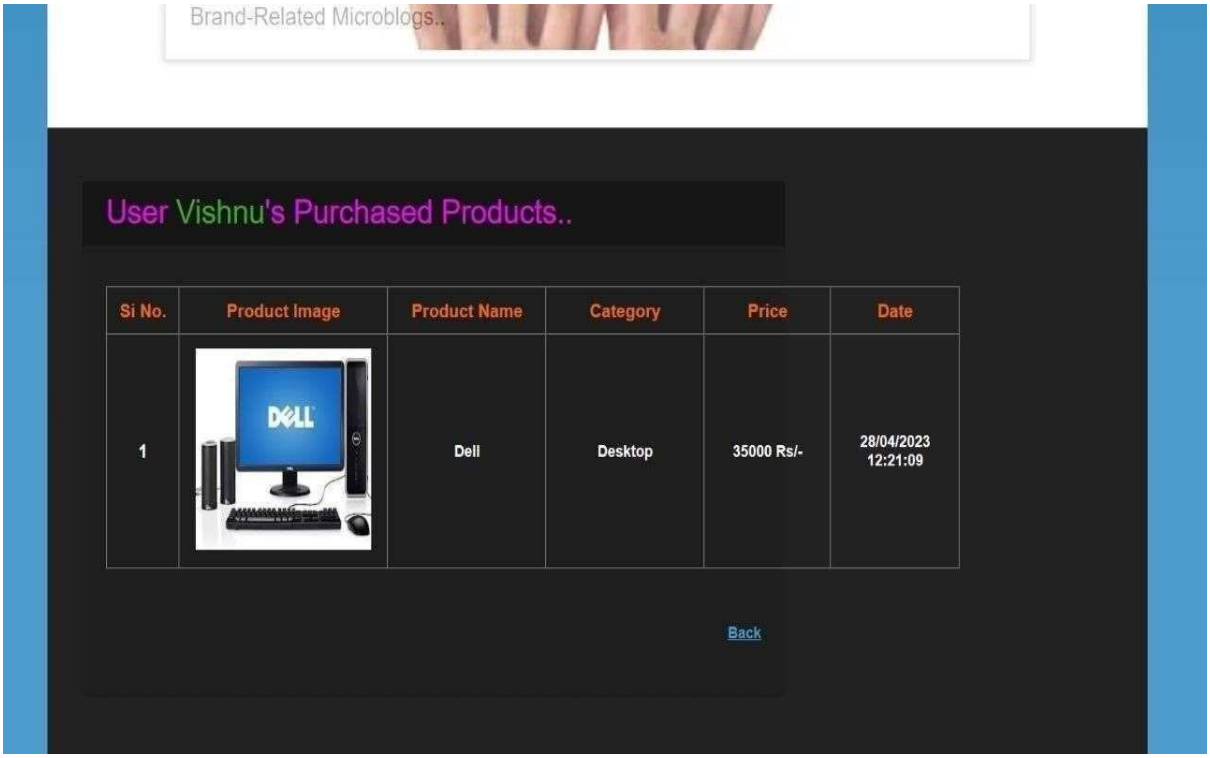
5. USER'S HOME PAGE



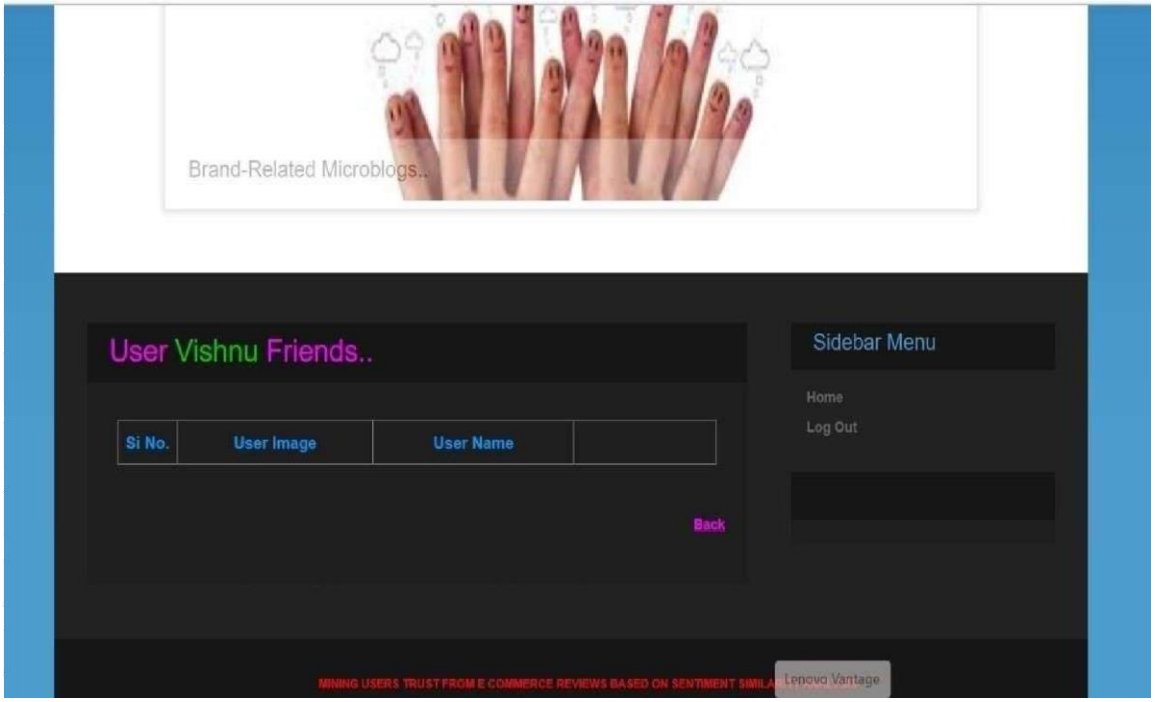
6. LIST OF PRODUCTS:



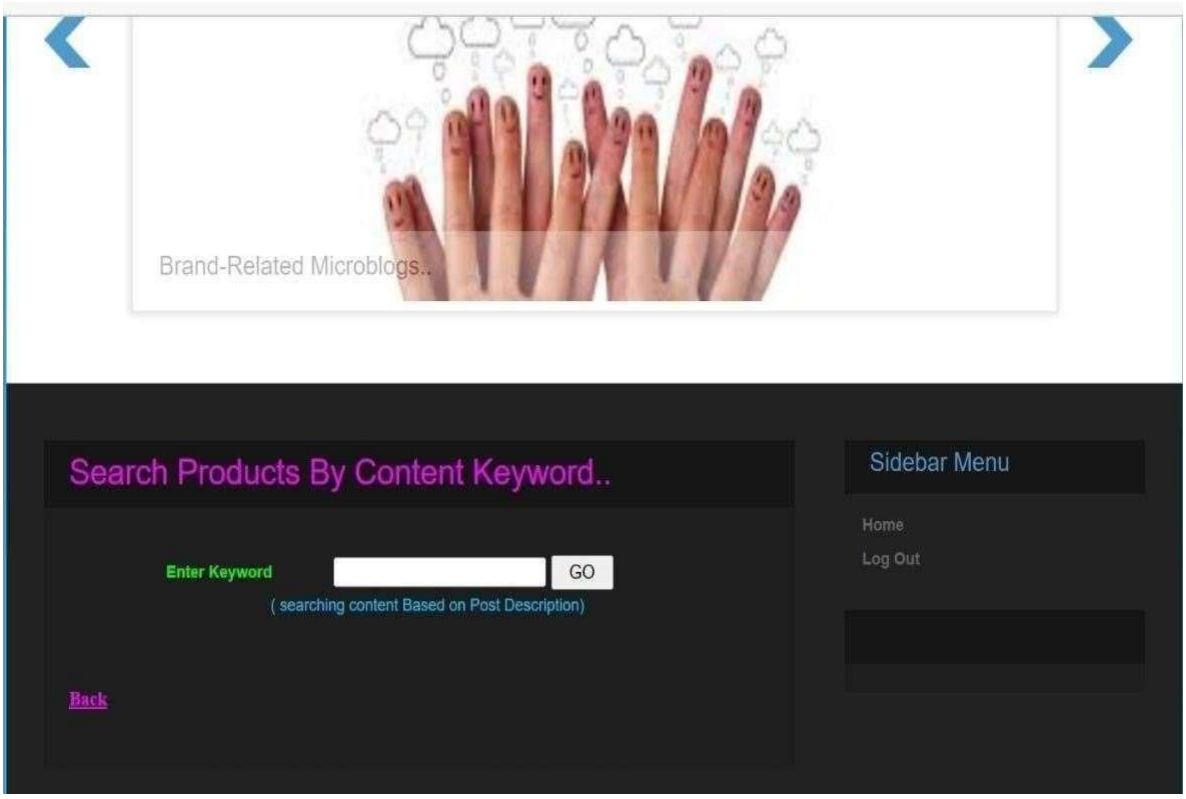
7. USERS PURCHASED PRODUCTS :



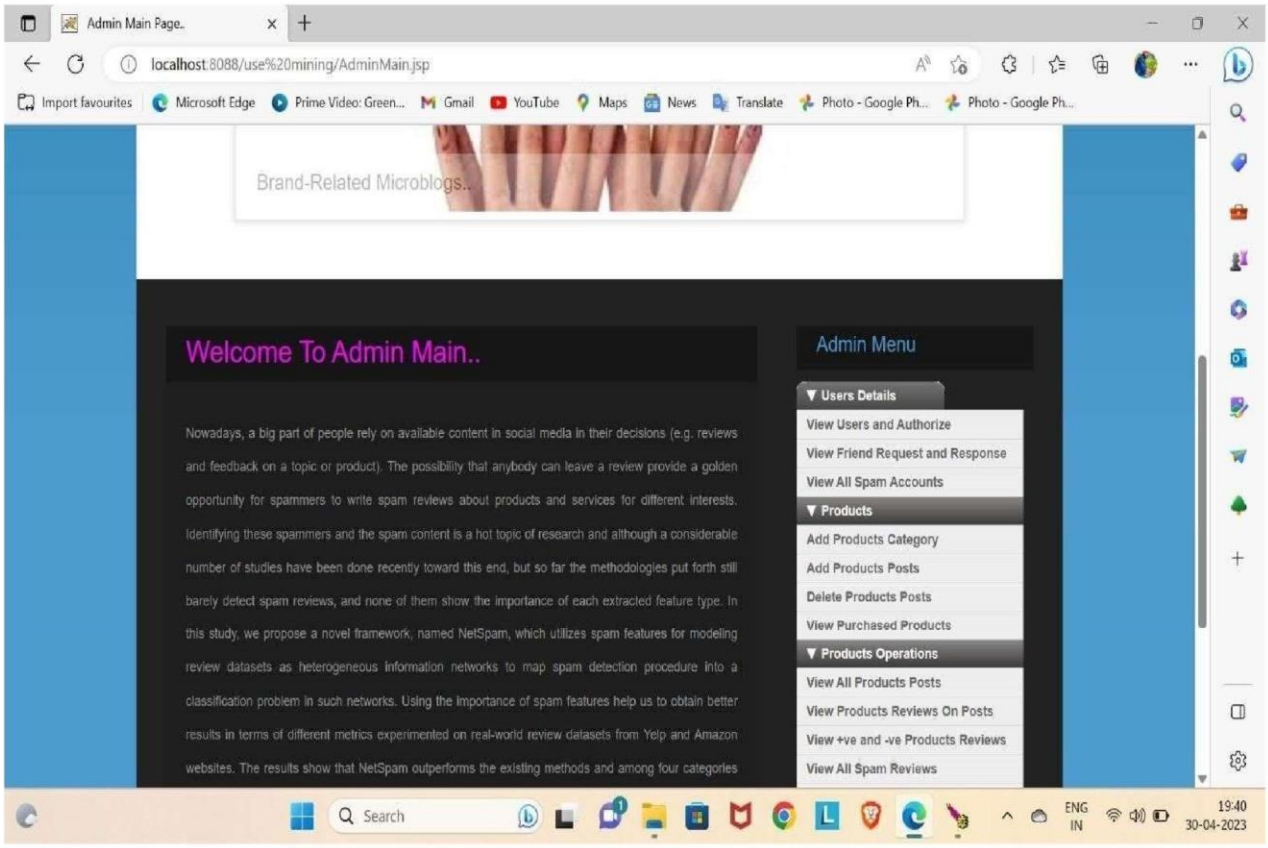
8. USERS FRIENDS PAGE :



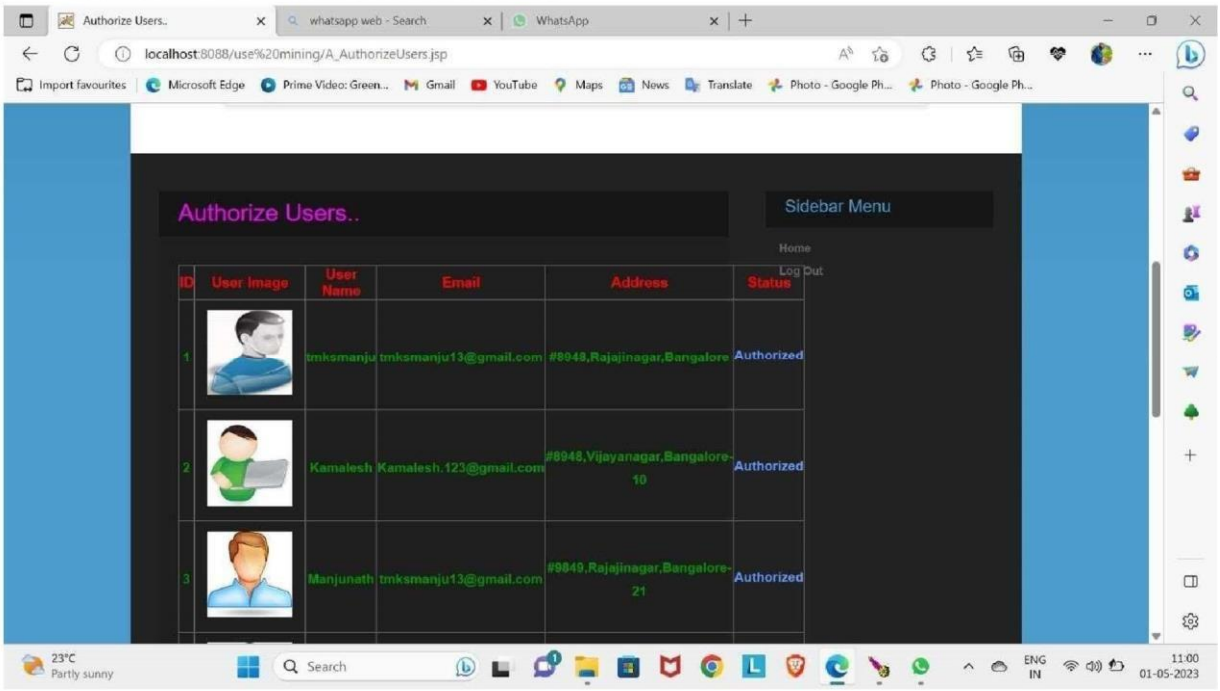
9. SEARCH PRODUCT PAGE:



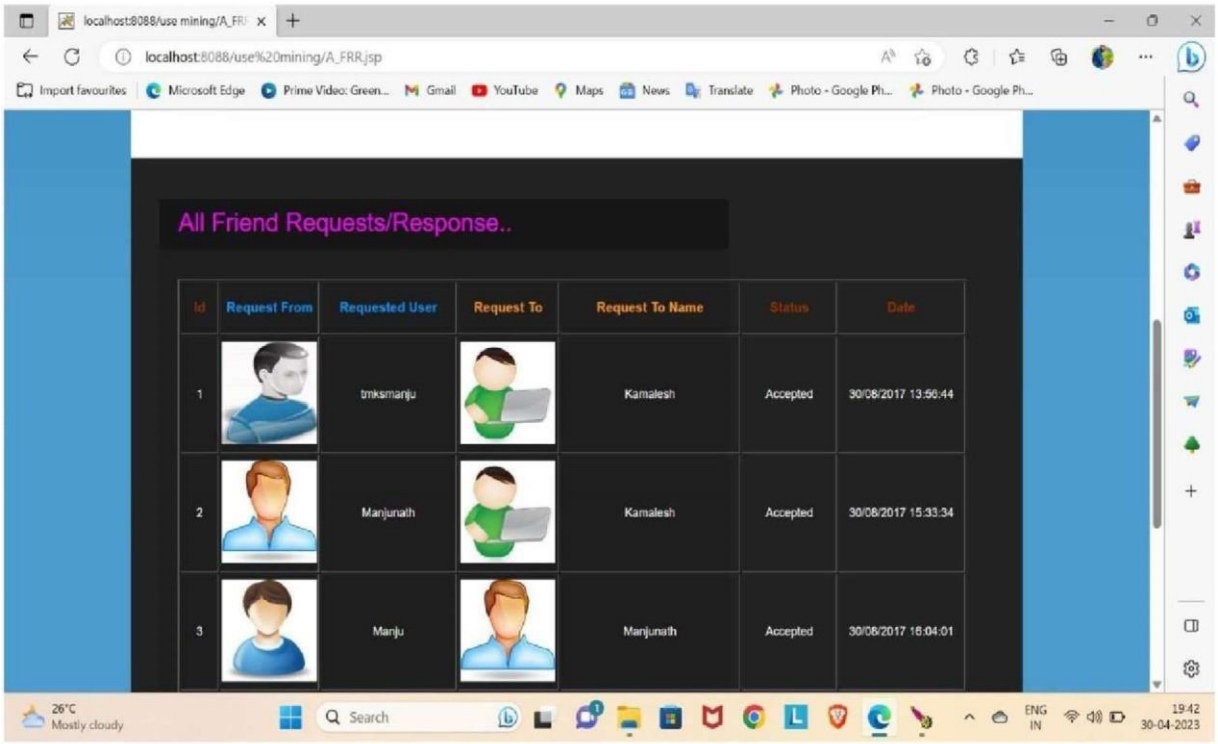
10. ADMIN HOME PAGE:



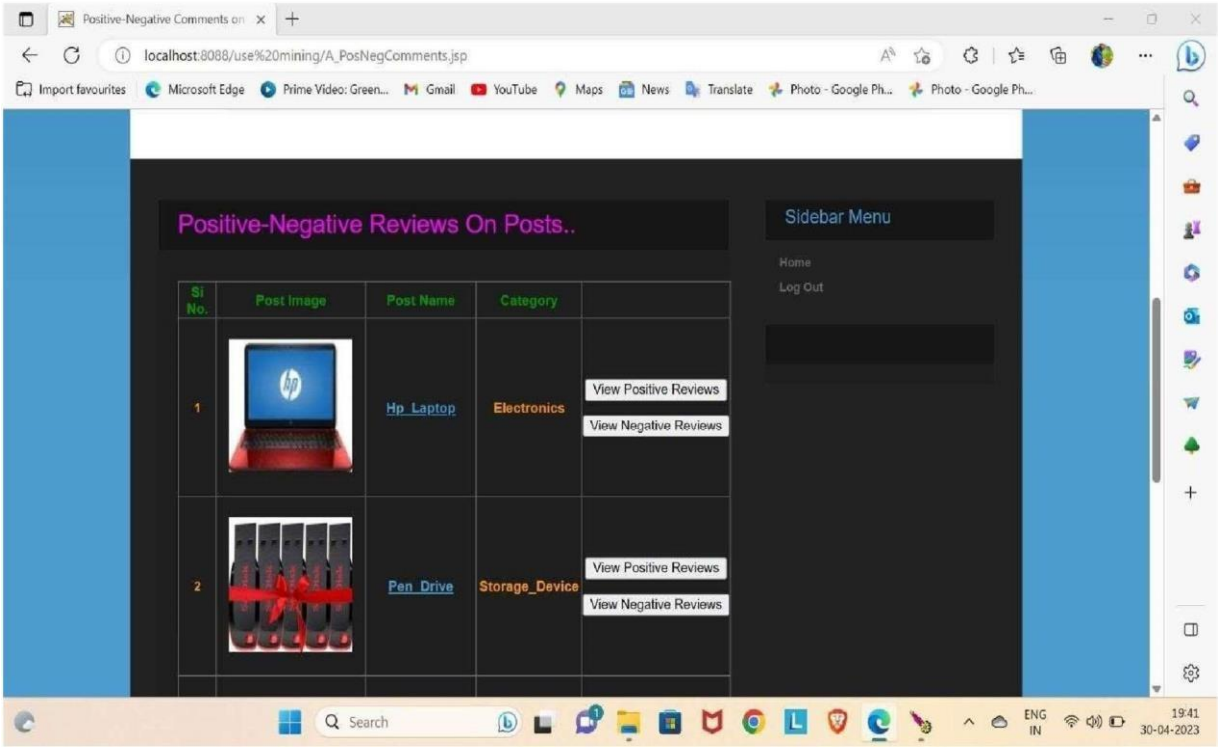
11. ADMIN AUTHORIZE USERS PAGE :



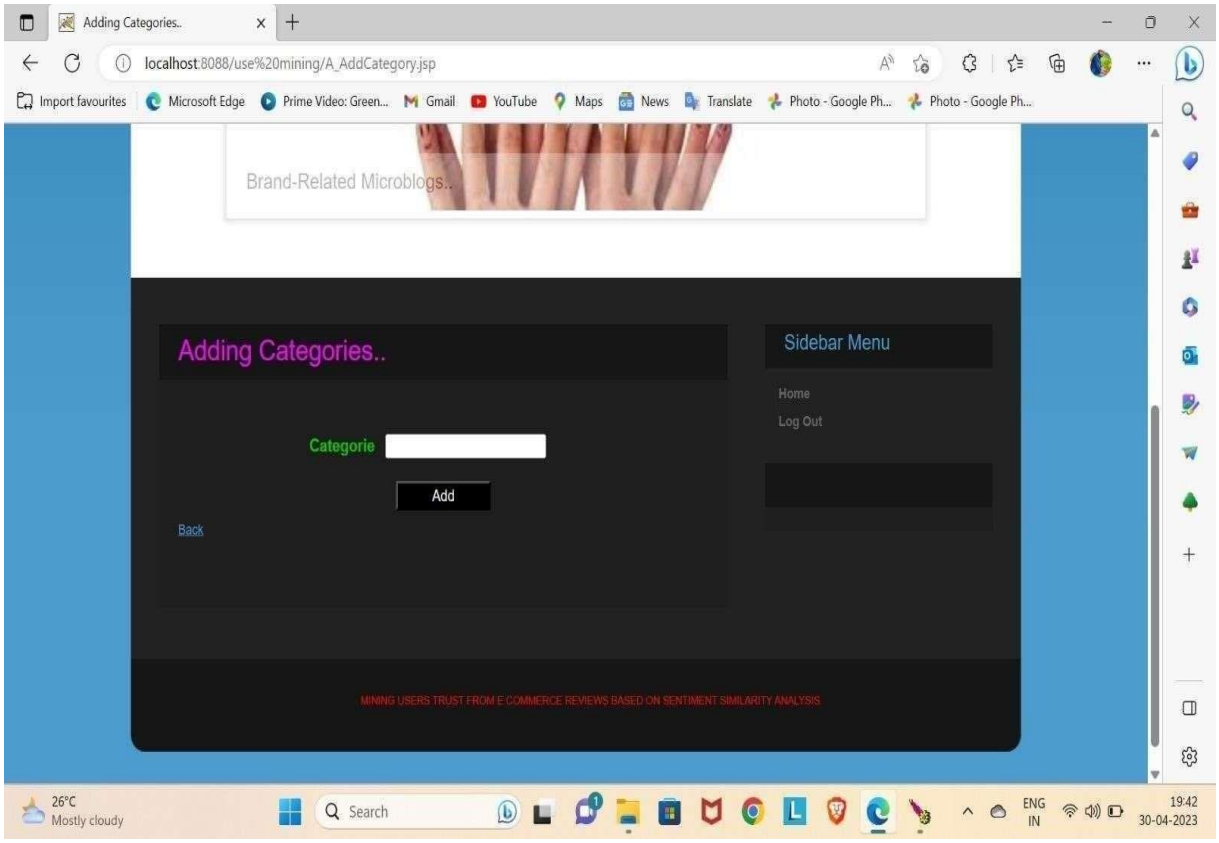
12. ALL FRIEND REQUESTS / RESPONSES



13. POSITIVE – NEGATIVE REVIEWS ON PRODUCTS



14. ADDING CATEGORIES



15. CHART RESULTS:

12. CONCLUSION

In our work, we address the problem of mining users trust in E-commerce system. By defining two kinds of trust relationship, namely, direct trust and propagation trust, we transfer the point of exploring trust between users into calculation of sentiment similarity of their reviews. With the help of entity-sentiment word pairs mining, sentiment similarity of reviews can be calculated and direct trust relationships can be obtained through sentiment similarity analysis, which contains of sentiments and ratings aspect. These two aspects can be used jointly to analyse the sentiment direct trust relationship. We establish a weighed trust graph model for propagation trust computing. Propagation trust is the use of the propagation characteristics of trust. It is an indirect trust between two users without direct trust and is obtained through intermediate users who have direct trust between these two users. The propagation trust calculation approaches based on the improved shortest path algorithm, and the time complexity is O(V2), where V is the number of node in the graph. Ways to improve the computational complexity of the algorithm is a new problem that needs further study because the relatively large number of users in modern e-commerce system.

13. REFERENCES

[1] H. Liu, F. Xia, Z. Chen, N. Y. Asabere, J. Ma, and R. Huang, ‘‘TruCom: Exploiting domain-specific trust networks for multicategory item recommendation,’’ IEEE Syst. J., vol. 11, no. 1, pp. 295–304, Mar. 2017.

[2] P.-Y. Hsu, H.-T. Lei, S.-H. Huang, T. H. Liao, Y.-C. Lo, and C.-C. Lo,‘‘Effectsofsentimentonrecommendationsinsocialnetwork,’’inElectron Markets.Berlin,Germany:Springer,2018,pp.110,doi:10.1007/s12525018-0314-5.

[3] C. Qin, W. Siyi, and A. Lin, ‘‘The joint beta distribution with refund rate in online C2C trust building: A theoretical study on Taobao,’’ in Proc. Int. Conf. E-Learn. ETechnol. Educ. (ICEEE), Lodz, Poland, Sep. 2012, pp. 191–196.

[4] S. Kraounakis, I. N. Demetropoulos, A. Michalas, M. S. Obaidat, P. G. Sarigiannidis, and M. D. Louta, ‘‘A robust reputation-based computationalmodelfortrustestablishmentinpervasivesystems,’’IEEESyst.J., vol. 9, no. 3, pp. 878–891, Sep. 2015.

[5] P. De Meo, E. Ferrara, D. Rosaci, and G. M. L. Sarné, ‘‘Trust and compactness in social network groups,’’ IEEE Trans. Cybern., vol. 45, no. 2, pp. 205–216, Feb. 2015.

[6] M. G. Ozsoy and F. Polat, ‘‘Trust based recommendation systems,’’ in Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining (ASONAM), Niagara Falls, ON, Canada, Aug. 2013, pp. 1267–1274.

[7] L. Sheugh and S. H. Alizadeh, ‘‘A fuzzy aproach for determination trust threshold in recommender systems based on social network,’’ in Proc. 9th Int. Conf. E-Commerce Developing Countries, Focus E-Bus. (ECDC), Isfahan, Iran, Apr. 2015, pp. 1–5.

[8] Y. Ruan, L. Alfantoukh, and A. Durresi, ‘‘Exploring stock market using twitter trust network,’’ in Proc. IEEE 29th Int. Conf. Adv. Inf. Netw. Appl., Gwangiu, South Korea, Mar. 2015, pp. 428–433. [9] C.-N. Ziegler, Social Web Artifacts for Boosting Recommenders, vol.487. Berlin, Germany: Springer, 2013.

[10] C.-N. Ziegler and J. Golbeck, ‘‘Investigating interactions of trust and interest similarity,’’ Decis. Support Syst., vol. 43, no. 2, pp. 460–475, 2007.