**“DETECTING AND CHARACTERIZING EXTREMIST REVIEWER GROUPS IN ONLINE PRODUCT REVIEWS”**

A Project Report submitted in partial fulfillment of the degree of the   
Bachelor of Technology in Computer Science and Engineering

**By**

**POTHIGANTI SRILATHA 19C41A05F4(srilathachinnu23@gmail.com)**

**KATANGUR SANDESH REDDY 18C41A0538(sandeshkatangur100@gmail.com**

**GADDAM BARGHAV 18C41A0515(buntybhargav79@gmail.com)**

**GODUGU MEENAKSHI 19C41A05C6(meenakshigodugu@gmail.com)**

Under the Guidance of

**Mrs.N.SINDHUJA**

**(Assistant Professor)**



**Department of Computer Science and Engineering**

**JAYAMUKHI INSTITUTE OF TECHNOLOGICAL SCIENCES**

**NARSAMPET, WARANGAL-506 332**

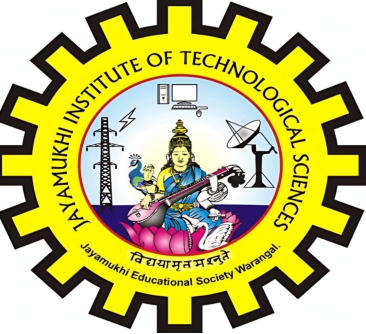
**(Autonomous, Affiliated to JNTUH, Accredited by NAAC with “A” GRADE)**

**JAYAMUKHI INSTITUTE OF TECHNOLOGICAL SCIENCES**

**NARSAMPET, WARANGAL-506 332**

**(Autonomous, Affiliated to JNTUH, Accredited by NAAC with “A” GRADE)**

**CERTIFICATE**



This is to certify that the Major project Report entitled “**DETECTING AND CHARACTERIZING EXTREMIST REVIEWER GROUPS IN ONLINE PRODUCT REVIEWS**” is a bona fide work of the students **POTHIGANTI SRILATHA (19C41A05F4), KATANGUR SANDESH REDDY(18C41A0538), GADDAM BARGHAV (18C41A0515), GODUGU MEENAKSHI (19C41A05C6)** submitted in partial fulfilment of the requirements for the award of the degree **of Bachelor of Technology** in **CSE** during the academic year 2022-2023.

**Guide Head of the Department**

**Principal**

**ACKNOWLEDGEMENT**

We express our immense gratitude and sincere thanks to **Prof.V.JANAKI,**

Principal of Jayamukhi Institute of Technological Sciences, Narsampet, Warangal for his

constant motivation and support.

We express our immense gratitude and sincere thanks to **Dr.P.SRINIVAS RAO**, Head

of the Department of CSE, of Jayamukhi Institute of Technological Sciences for his constant

motivation and support.

We express our immense gratitude and sincere thanks to **Mrs.N.SINDHUJA**,

Assistant Professor for her valuable guidance and giving us the opportunity to do our work in

a better manner.

We express our immense gratitude and sincere thanks to our project coordinator **Mr.SANDEEP**, for his guidance.

**THANK YOU**

**ABSTRACT**

Online marketplaces often witness opinion spam in the form of reviews. People are often hired to target specific brands for promoting or impeding them by writing highly positive or negative reviews. This often is done collectively in groups. Although some previous studies attempted to identify and analyze such opinion spam groups, little has been explored to spot those groups who target a brand as a whole, instead of just products. In this article, we collected the reviews from the Amazon product review site and manually labeled a set of 923 candidate reviewer groups. The groups are extracted using frequent itemset mining over brand similarities such that users are clustered together if they have mutually reviewed a lot of brands. We hypothesize that the nature of the reviewer groups is dependent on eight features specific to a pair. We develop a feature-based supervised model to classify candidate groups as extremist entities. We run multiple classifiers for the task of classifying a group based on the reviews written by the users of that group to determine whether the group shows signs of extremity. A three-layer perceptron-based classifier turns out to be the best classifier. We further study behaviors of such groups in detail to understand the dynamics of brand-level opinion fraud better. These behaviors include consistency in ratings, review sentiment, verified purchase, review dates, and helpful votes received on reviews. Surprisingly, we observe that there are a lot of verified reviewers showing extreme sentiment, which, on further investigation, leads to ways to circumvent the existing mechanisms in place to prevent unofficial incentives on Amazon.

**CONTENTS**

**S.NO TOPIC NAME PAGE NO**

1. INTRODUCTION 6-8

2. J2EE SOFTWARE ENVIRONMENT 9-17

3. SYSTEM DESIGN 18-25

4. IMPLEMENTATION 26

5. SYSTEM STUDY 27-30

6. SYSTEM TESTING 31-38

7. CONCLUSION 39

8. REFERENCE 40-41

**1.INTRODUCTION**

In today’s world dominated by online marketplaces, review portals and websites play a crucial role in the buyer’s decision for their next purchase. “It is a virtuous cycle–the more reviews, the more buys. The more buys, the more reviews. The more buys, the higher your rank in search and the more sales you get,” says Alice [1], the owner of online cosmetic brand Elizabeth Mott. Undoubtedly, it is highly likely that some people write reviews that are less than truthful to manipulate widespread decision of buyers in their favor. These people act either individually or in groups. While individual reviewers write such reviews in a matter of frustration or joy, they do not influence the overall opinion on a product to a large extent but help other buyers by stating their experiences. However, a more compelling case is when multiple individuals form an intricate web, and due to sheer higher number of people reviewing , they end up being a major influence on the overall sentiment of the product. The extent of such influence is not just limited to the reviews by opinion spam. Previous work [2] has shown that 10%–15% reviews are essentially echoing the earliest reviews, and thus, a misleading early review has an even higher influential potential. This is widespread opinion spam, and every review website must be aware of this activity and take appropriate measures for the identification and/or prevention of this phenomenon. This is a classic example of collective fraud behavior, where several users are part of a business network and work together to target and influence a particular product. This is a lesser known identify such groups that try to influence a product [4]–[6]; however, groups exhibiting a brand-based opinion spamming is a phenomenon that remains widely unexplored. A detailed discussion is required for these brand-related activities because these practices are against the code of conduct of these review websites since they negatively skew the brand-based competition, giving innate (dis)advantages to certain brands. Since only the nonverified reviews are limited by the policies,1 reviewers from these groups can often purchase the product via Amazon in exchange for unofficial discounts (e.g., cash backs) and post verified reviews since they did not receive a discount via amazon’s mechanisms (e.g., coupons) (see further discussed in Section VIII). Fig. 1 shows an example of such extremist groups (taken from our annotated data set as mentioned in Section III). Four rows correspond to the products belonging to four different brands. Four columns represent four different reviewers who, according to our annotation, are part of the same group. Each box represents the review information. This is an example of reviewers showing extreme likeliness toward these products/brands as can be seen from the extreme ratings, similar comments, and almost the same date [7]. It is clear that this group of reviewers had extreme sentiments toward the brands reviewed, both in terms of the ratings and the review content. While this has a large overlap with the classic collision in which reviewers target a product to bring up its ratings, this kind of extremism in reviews is not in order to promote/demote the ranking of a product, but rather to influence the perception of people for a brand. It is worth noting that such a kind of characterization is different from just combining the groups of people who provide extreme reviews on a product, because while the groups focusing on a product may be extreme in their opinion, they do not necessarily intend to influence the brand image. This, coupled with the fact that the same product may be sold by different sellers, reduces the chance that a product-level opinion spamming reviewer group would target products of the same brand. Sellers may not have any inclination toward promoting any particular brand’s products; rather, they would prefer to gain a better revenue on all products (may belong to different brands) by their promotional campaigns. Hence, such a group is more likely to have a very narrow and specialized source, e.g., the competing brands or the manufacturing brand itself. In this article, we identify and study the behavioral characteristics of extremist reviewer groups. We also build a feature-based classifier based on the brand-specific activities of reviewer groups to identify the extremist groups on the Amazon India marketplace. We then further analyze our methodology to unfold behaviors that best signify such activities and compare and analyze the overall trend of these groups viz-a-viz their behaviors. The major contributions of this article are fourfold: 1) a manually labeled data set of 923 reviewer groups that are classified into “extremist” and “moderate” categories; 2) the first-ever characterization and study of the novel problem of identifying brand-level extremism; 3) detailed characterization of extremist reviewer groups; 4) design supervised approach to detect extremist reviewer groups.

**EXISTING SYSTEM:**

* Sentiment analysis can be conducted on three different levels: review-level, sentence-level, and phrase-level.
* Review-level analysis and sentence-level analysis attempt to classify the sentiment of a whole review to one of the predefined sentiment polarities, including positive, negative and sometimes neutral.
* While phrase-level analysis attempt to extract the sentiment polarity of each feature that a user expresses his/her attitude to the specific feature of a specific product.
* Zhang et al.propose a self-supervised and lexicon-based sentiment classification approach to determine sentiment polarity of a review that contains both textual words and emoticons. And they use sentiment for recommendation.
* Lee et al.propose a recommender system using the concept of Experts to find both novel and relevant recommendations. By analyzing the user ratings, they can recommend special experts to a target user based on the user population.

**DISADVANTAGES OF EXISTING SYSTEM:**

* The existing work mainly focuses on classifying users into binary sentiment, and they do not go further in mining user’s sentiment.
* The existing approaches mainly leverage product category information or tag information to study the interpersonal influence.
* These methods are all restricted on the structured data, which is not always available on some websites. However, user reviews can provide us ideas in mining interpersonal inference and user preferences.

**PROPOSED SYSTEM:**

* We propose a sentiment-based rating prediction method in the framework of matrix factorization. In our work, we make use of social users’ sentiment to infer ratings.
* First, we extract product features from user reviews. Then, we find out the sentiment words, which are used to describe the product features. Besides, we leverage sentiment dictionaries to calculate sentiment of a specific user on an item/product.
* The main contributions of our approach are as follows:
* We propose a user sentimental measurement approach, which is based on the mined sentiment words and sentiment degree words from user reviews.
* We make use of sentiment for rating prediction. User sentiment similarity focuses on the user interest preferences. User sentiment influence reflects how the sentiment spreads among the trusted users. Item reputation similarity shows the potential relevance of items.
* We fuse the three factors: user sentiment similarity, interpersonal sentimental influence, and item reputation similarity into a probabilistic matrix factorization framework to carry out an accurate recommendation. The experimental results and discussions show that user's social sentiment that we mined is a key factor in improving rating prediction performances.

**ADVANTAGES OF PROPOSED SYSTEM:**

* In our paper, we not only mine social user’s sentiment, but also explore interpersonal sentimental influence and item’s reputation. Finally, we take all of them into the recommender system.
* The purpose of our approach is to find effective clues from reviews and predict social users’ ratings.
* We fuse user sentiment similarity, inter personal sentiment influence, and item reputation similarity into a unified matrix factorization frame work to achieve the rating prediction task.

**2.J2EE SOFTWARE ENVIRONMENT**

CLIENT SERVER

With the varied topic in existence in the fields of computers, Client Server is one, which has generated more heat than light, and also more hype than reality. This technology has acquired a certain critical mass attention with its dedication conferences and magazines. Major computer vendors such as IBM and DEC, have declared that Client Servers is their main future market. A survey of DBMS magazine reveled that 76% of its readers were actively looking at the client server solution.

Client server implementations are complex but the underlying concept is simple and powerful. A client is an application running with local resources but able to request the database and relate the services from separate remote server. The software mediating this client server interaction is often referred to as MIDDLEWARE.

#### **What is a Client-Server?**

Two prominent systems in existence are client server and file server systems. It is essential to distinguish between client servers and file server systems. Both provide shared network access to data but the comparison dens there! The file server simply provides a remote disk drive that can be accessed by LAN applications on a file by basis. The client server offers full relational database services such as SQL-Access, Record modifying, Insert, delete with full relational integrity backup/ restore performance for high volume of transactions, etc. the client server middleware provides a flexible interface between client and server, who does what, when and to whom.

Front End or User Interface Design

#### The entire user interface is planned to be developed in browser specific environment with a touch of Intranet-Based Architecture for achieving the Distributed Concept. The browser specific components are designed by using the HTML standards, and the dynamism of the designed by concentrating on the constructs of the Java Server Pages.

#### Communication or Database Connectivity Tier

The Communication architecture is designed by concentrating on the Standards of Servlets and Enterprise Java Beans. The database connectivity is established by using the Java Data Base Connectivity. The standards of three-tire architecture are given major concentration to keep the standards of higher cohesion and limited coupling for effectiveness of the operations.

#### About Java

Initially the language was called as “oak” but it was renamed as “Java” in 1995. The primary motivation of this language was the need for a platform-independent (i.e., architecture neutral) language that could be used to create software to be embedded in various consumer electronic devices.

* Java is a programmer’s language.
* Java is cohesive and consistent.
* Except for those constraints imposed by the Internet environment, Java gives the programmer, full control.

Finally, Java is to Internet programming where C was to system programming.

Importance of Java to the Internet

Java has had a profound effect on the Internet. This is because; Java expands the Universe of objects that can move about freely in Cyberspace. In a network, two categories of objects are transmitted between the Server and the Personal computer. They are: Passive information and Dynamic active programs. The Dynamic, Self-executing programs cause serious problems in the areas of Security and probability. But,Java addresses those concerns and by doing so, has opened the door to an exciting new form of program called the Applet.

# Features of Java

**1.Security**

Every time you that you download a “normal” program, you are risking a viral infection. Prior to Java, most users did not download executable programs frequently, and those who did scanned them for viruses prior to execution. Most users still worried about the possibility of infecting their systems with a virus. Java answers both these concerns by providing a “firewall” between a network application and your computer. When you use a Java-compatible Web browser, you can safely download Java applets without fear of virus infection or malicious intent.

**2.Portability**

For programs to be dynamically downloaded to all the various types of platforms connected to the Internet, some means of generating portable executable code is needed. As you will see, the same mechanism that helps ensure security also helps create portability. Indeed, Java’s solution to these two problems is both elegant and efficient.

**3.The Byte Code**

The key that allows the Java to solve the security and portability problems is that the output of Java compiler is Byte code. When the JIT compiler is a part of JVM, it compiles byte code into executable code in real time, on a piece-by-piece, demand basis. It is not possible to compile an entire Java program into executable code all at once, because Java performs various run-time checks that can be done only at run time. The JIT compiles code, as it is needed, during execution.

**Java Virtual Machine (JVM)**

Beyond the language, there is the Java virtual machine. The Java virtual machine is an important element of the Java technology. The virtual machine can be embedded within a web browser or an operating system. Byte code verification takes place at the end of the compilation process to make sure that is all accurate and correct. So byte code verification is integral to the compiling and executing of Java code.

**Compilation of code**

**Source**

**Code**

**………..**

**………..**

**………..**

**…………**

# PC Compiler

**Macintosh**

**Compiler**

**SPARC**

###### Compiler

**Java**

**Byte code**

**(Platform**

**indepen**

**dent)**

**Java**

**Interpreter**

**(PC)**

**Java**

**Interpreter**

**(Macintosh)**

**Java**

**Interpreter**

**(Sparc)**

**JAVASCRIPT**

JavaScript is a script-based programming language that was developed by Netscape Communication Corporation. JavaScript was originally called Live Script and renamed as JavaScript to indicate its relationship with Java. Even though JavaScript supports both client and server Web programming, we prefer JavaScript at client-side programming since most of the browsers supports it. JavaScript is almost as easy to learn as HTML, and JavaScript statements can be included in HTML documents by enclosing the statements between a pair of scripting tag.

<SCRIPTS></SCRIPT>.

<SCRIPT LANGUAGE = “JavaScript”>

JavaScript statements

</SCRIPT>

* **Hyper Text Markup Language**
* Hypertext Markup Language (HTML), the languages of the World Wide Web (WWW), allows users to produces Web pages that include text, graphics and pointer to other Web pages (Hyperlinks(The idea behind Hypertext is that instead of reading text in rigid linear structure, we can easily jump from one point to another point.HTML provides tags (special codes) to make the document look attractive. HTML tags are not case-sensitive. Using graphics, fonts, different sizes, color, etc.,

**Basic HTML Tags:**

**<!-- -->** Specifies comments

**<A>……….</A>** Creates hypertext links

**<B>……….</B>** Formats text as bold

**<BIG>……….</BIG>** Formats text in large font.

**<BODY>…</BODY>** Contains all tags and text in the HTML document

**<DD>…</DD>** Definition of a term

**<DL>...</DL>**  Creates definition list

**<FONT>…</FONT>** Formats text with a particular font

**<FORM>...</FORM>** Encloses a fill-out form

**<FRAME>...</FRAME>** Defines a particular frame in a set of frames

**<HEAD>...</HEAD>** Contains tags that specify information about a document

**<HR>...</HR>** Creates a horizontal rule

**<HTML>…</HTML>** Contains all other HTML tags

**<META>...</META>** Provides meta-information about a document

**<SCRIPT>…</SCRIPT>** Contains client-side or server-side script

**<TD>…</TD>** Indicates table data in a table

**<TR>…</TR>** Designates a table row

**<TH>…</TH>** Creates a heading in a table

**Java Database Connectivity**

**What is JDBC?**

JDBC is a Java API for executing SQL statements. (As a point of interest, JDBC is a trademarked name and is not an acronym; nevertheless, JDBC is often thought of as standing for Java Database Connectivity. It consists of a set of classes and interfaces written in the Java programming language. JDBC provides a standard API for tool/database developers and makes it possible to write database applications using a pure Java API. Using JDBC, it is easy to send SQL statements to virtually any relational database.

**What Does JDBC Do?**

Simply put, JDBC makes it possible to do three things:

* Establish a connection with a database
* Send SQL statements
* Process the results.

**JDBC versus ODBC and other APIs**

At this point, Microsoft's ODBC (Open Database Connectivity) API is that probably the most widely used programming interface for accessing relational databases. It offers the ability to connect to almost all databases on almost all platforms. So why not just use ODBC from Java? The answer is that you can use ODBC from Java, but this is best done with the help of JDBC in the form of the JDBC-ODBC Bridge, which we will cover shortly. The question now becomes "Why do you need JDBC?" There are several answers to this question:

1. ODBC is not appropriate for direct use from Java because it uses a C interface. Calls from Java to native C code have a number of drawbacks in the security, implementation, robustness, and automatic portability of applications.
2. A literal translation of the ODBC C API into a Java API would not be desirable. For example, Java has no pointers, and ODBC makes copious use of them, including the notoriously error-prone generic pointer "void \*".

**What Is the JDBC- ODBC Bridge?**

The JDBC-ODBC Bridge is a JDBC driver, which implements JDBC operations by translating them into ODBC operations. To ODBC it appears as a normal application program. The Bridge implements JDBC for any database for which an ODBC driver is available. Java package and contains a native library used to access ODBC. The Bridge is a joint development of Java Soft.

**Java Server Pages (JSP)**

Java server Pages is a simple, yet powerful technology for creating and maintaining dynamic-content web pages. Based on the Java programming language, Java Server Pages offers proven portability, open standards, and a mature re-usable component model. The Java Server Pages architecture enables the separation of content generation from content presentation. This separation not eases maintenance headaches, it also allows web team members to focus on their areas of expertise. Now, web page designer can concentrate on layout, and web application designers on programming, with minimal concern about impacting each other’s work.

**Features of JSP**

**1. Portability**

Java Server Pages files can be run on any web server or web-enabled application server that provides support for them. Dubbed the JSP engine, this support involves recognition, translation, and management of the Java Server Page lifecycle and its interaction components.

**2.Components**

It was mentioned earlier that the Java Server Pages architecture can include reusable Java components. The architecture also allows for the embedding of a scripting language directly into the Java Server Pages file. The components current supported include Java Beans, and Servlets.

**3.Processing**

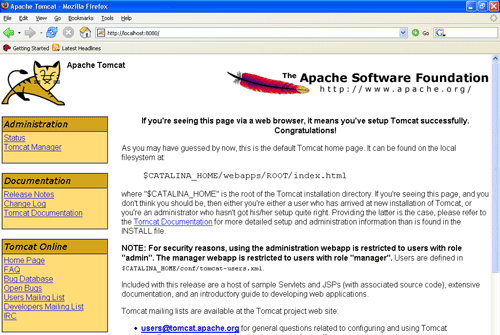
A Java Server Pages file is essentially an HTML document with JSP scripting or tags. The Java Server Pages file has a JSP extension to the server as a Java Server Pages file. Before the page is served, the Java Server Pages syntax is parsed and processed into a Servlet on the server side. The Servlet that is generated outputs real content in straight HTML for responding to the client.

**4.Access Models**

A Java Server Pages file may be accessed in at least two different ways. A client’s request comes directly into a Java Server Page. In this scenario, suppose the page accesses reusable Java Bean components that perform particular well-defined computations like accessing a database. The result of the Beans computations, called result sets is stored within the Bean as properties. The page uses such Beans to generate dynamic content and present it back to the client.

# **Tomcat 6.0 web server**

Tomcat is an open source web server developed by Apache Group. Apache Tomcat is the servlet container that is used in the official Reference Implementation for the Java Servlet and Java Server Pages technologies. The Java Servlet and Java Server Pages specifications are developed by Sun under the Java Community Process. Web Servers like Apache Tomcat support only web components while an application server supports web components as well as business components.



**Bibliography:**

* **Oracle**

PL/SQL Programming by Scott Urman

SQL complete reference by Livion

* **JAVA Technologies**

JAVA Complete Reference

Java Script Programming by Yehuda Shiran

Mastering JAVA Security

JAVA2 Networking by Pistoria

JAVA Security by Scotl oaks

Head First EJB Sierra Bates

J2EE Professional by Shadab siddiqui

JAVA server pages by Larne Pekowsley

JAVA Server pages by Nick Todd

* **HTML**

HTML Black Book by Holzner

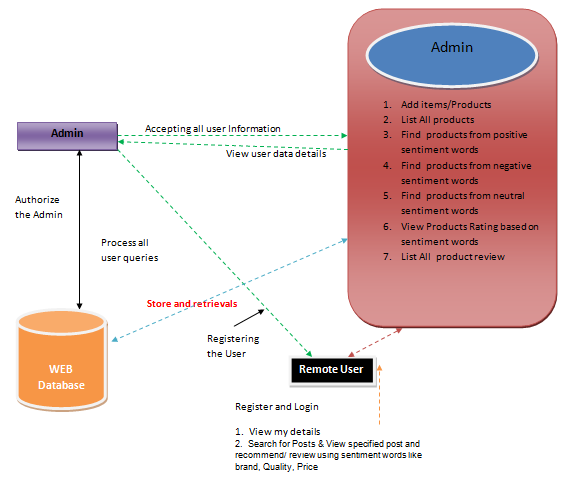
* **JDBC**

Java Database Programming with JDBC by Patel moss.

Software Engineering by Roger Pressman

**3.SYSTEM DESIGN**

**Architecture Diagram**



**CLASS DIAGRAM**

Administrator

Add items/Products ,List All Products ,Find products from positive sentiment words ,Find products from negative sentiment words ,Find products from neutral sentiment words ,View Products Rating based on sentiment words ,List All product review

Product name, price, description and corresponding product image

Methods



Login

Login, Register, Reset

User Name, Password



Members

Register, login, Reset

Name, password, e-mail, mob no, location, DOB, address, gender



Methods

Members

End User

View Profile, Search for Products, View specified post and recommend, review using sentiment words like brand, Quality, Price, and Logout

Product name, price, description and corresponding product image

View Profile Details, Search for Products, View specified post and recommend, review using sentiment words like brand, Quality, Price, and Logout

Product name, price, description and corresponding product image

Methods



Members

**Data Flow Diagram**

End User

Database

Admin

**Flow Chart1: User**

User Register

Start

Login

Yes No

View your profile

Username & Password Wrong

Search for product

Logout

Review using sentiment words based on brand, Quality, Price

**Flow Chart 2: Admin**

**Admin Login**

**Start**

**Login**

Yes No

**Add items/Products**

**Username & Password Wrong**

No

**List All Products**

**Log Out**

**List of all users and authorize**

**Find products from positive sentiment words**

**Find products from negative sentiment words**

**Find products from neutral sentiment words, View Products rating based on sentiment words**

**,List All product review**

**SEQUENCE DIAGRAM**

**Database**

**End User**

**Admin**

Provides login authorization

Add items/Products

Added post responce

Provides login authorization

Registration request

Register confirmation

List all users and authorize

Search for Posts

Authorize the admin

List All Posts with rating

Review using sentiment words like brand, Quality, Price

Registering the user

Find products from positive sentiment words

View my search History

View recommends Products from your friends

Find products from negative sentiment words

**Use Case:**

User

**IMPLEMENTATION**

* **Admin**

In this module, the Admin has to login by using valid user name and password. After login successful he can do some operations such as add categories, add posts, list of all posts, list of all recommended posts, view good reviews, view bad reviews, list of all reviewed posts, list of users, list of all search history, update posts, lists of bad reviews by date wise, list of good reviews by date wise.

**Add Products Posts**

In this module, the admin can add the post by including product name, price, description and corresponding product image.

**View all products**

In this module, the admin can view the post by searching keyword and can get all the information about the product like product name, price, description and corresponding product image.

* **User**

In this module, there are n numbers of users are present. User should register before doing some operations. After registration successful he has to login by using authorized user name and password. Login successful he will do some operations like view user details, search for products posts, view my search history, view recommended, search for top N posts and logout.

**Searches for good reviews, bad review and neutral reviews**

In this module, user searches for reviews for the post and can get the following information like product name, price, description and corresponding product image. The user can recommend the product and can give review using sentiment words (such as good or bad product like that) based on brand, Quality, Price.

**SYSTEM STUDY**

**FEASIBILITY STUDY**

Three key considerations involved in the feasibility analysis are

* ECONOMICAL FEASIBILITY
* TECHNICAL FEASIBILITY
* SOCIAL FEASIBILITY

**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.

### 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.

**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.

**FEASIBILITY STUDY**

**PRELIMINARY INVESTIGATION**

The first and foremost strategy for development of a project starts from the thought of designing a mail enabled platform for a small firm in which it is easy and convenient of sending and receiving messages, there is a search engine ,address book and also including some entertaining games. When it is approved by the organization and our project guide the first activity, ie. preliminary investigation begins. The activity has three parts:

* **Request Clarification**
* **Feasibility Study**
* **Request Approval**

**REQUEST CLARIFICATION**

After the approval of the request to the organization and project guide, with an investigation being considered, the project request must be examined to determine precisely what the system requires.

Here our project is basically meant for users within the company whose systems can be interconnected by the Local Area Network(LAN). In today’s busy schedule man need everything should be provided in a readymade manner. So taking into consideration of the vastly use of the net in day to day life, the corresponding development of the portal came into existence.

**FEASIBILITY ANALYSIS**

An important outcome of preliminary investigation is the determination that the system request is feasible. This is possible only if it is feasible within limited resource and time. The different feasibilities that have to be analyzed are

* **Operational Feasibility**
* **Economic Feasibility**
* **Technical Feasibility**

###### **Operational Feasibility**

Operational Feasibility deals with the study of prospects of the system to be developed. This system operationally eliminates all the tensions of the Admin and helps him in effectively tracking the project progress. This kind of automation will surely reduce the time and energy, which previously consumed in manual work. Based on the study, the system is proved to be operationally feasible.

**Economic Feasibility**

Economic Feasibility or Cost-benefit is an assessment of the economic justification for a computer based project. As hardware was installed from the beginning & for lots of purposes thus the cost on project of hardware is low. Since the system is a network based, any number of employees connected to the LAN within that organization can use this tool from at anytime. The Virtual Private Network is to be developed using the existing resources of the organization. So the project is economically feasible.

###### **Technical Feasibility**

According to Roger S. Pressman, Technical Feasibility is the assessment of the technical resources of the organization. The organization needs IBM compatible machines with a graphical web browser connected to the Internet and Intranet. The system is developed for platform Independent environment. Java Server Pages, JavaScript, HTML, SQL server and WebLogic Server are used to develop the system. The technical feasibility has been carried out. The system is technically feasible for development and can be developed with the existing facility.

**4.3.3 REQUEST APPROVAL**

Not all request projects are desirable or feasible. Some organization receives so many project requests from client users that only few of them are pursued. However, those projects that are both feasible and desirable should be put into schedule. After a project request is approved, it cost, priority, completion time and personnel requirement is estimated and used to determine where to add it to any project list. Truly speaking, the approval of those above factors, development works can be launched.

**SYSTEM DESIGN AND DEVELOPMENT**

**INPUT DESIGN**

Input Design plays a vital role in the life cycle of software development, it requires very careful attention of developers. The input design is to feed data to the application as accurate as possible. So inputs are supposed to be designed effectively so that the errors occurring while feeding are minimized. According to Software Engineering Concepts, the input forms or screens are designed to provide to have a validation control over the input limit, range and other related validations.

This system has input screens in almost all the modules. Error messages are developed to alert the user whenever he commits some mistakes and guides him in the right way so that invalid entries are not made. Let us see deeply about this under module design.Validations are required for each data entered. Whenever a user enters an erroneous data, error message is displayed and the user can move on to the subsequent pages after completing all the entries in the current page.

OUTPUT DESIGN

The Output from the computer is required to mainly create an efficient method of communication within the company primarily among the project leader and his team members, in other words, the administrator and the clients. The output of VPN is the system which allows the project leader to manage his clients in terms of creating new clients and assigning new projects to them, maintaining a record of the project validity and providing folder level access to each client on the user side depending on the projects allotted to him. After completion of a project, a new project may be assigned to the client. User authentication procedures are maintained at the initial stages itself. A new user may be created by the administrator himself or a user can himself register as a new user but the task of assigning projects and validating a new user rests with the administrator only.

The application starts running when it is executed for the first time. The server has to be started and then the internet explorer in used as the browser. The project will run on the local area network so the server machine will serve as the administrator while the other connected systems can act as the clients. The developed system is highly user friendly and can be easily understood by anyone using it even for the first time.

**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**

**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.

**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.

**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.

**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.

**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.

**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.

**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.

# **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.

**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.

**SYSTEM TESTING**

### TESTING METHODOLOGIES

The following are the Testing Methodologies:

* **Unit Testing.**
* **Integration Testing.**
* **User Acceptance Testing.**
* **Output Testing.**
* **Validation Testing.**

**Unit Testing**

Unit testing focuses verification effort on the smallest unit of Software design that is the module. Unit testing exercises specific paths in a module’s control structure to ensure complete coverage and maximum error detection. This test focuses on each module individually, ensuring that it functions properly as a unit. Hence, the naming is Unit Testing.

During this testing, each module is tested individually and the module interfaces are verified for the consistency with design specification. All important processing path are tested for the expected results. All error handling paths are also tested.

**Integration Testing**

Integration testing addresses the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of high order tests are conducted. The main objective in this testing process is to take unit tested modules and builds a program structure that has been dictated by design.

**The following are the types of Integration Testing:**

**1)Top-Down Integration**

This method is an incremental approach to the construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main program module. The module subordinates to the main program module are incorporated into the structure in either a depth first or breadth first manner.

In this method, the software is tested from main module and individual stubs are replaced when the test proceeds downwards.

**2. Bottom-up Integration**

This method begins the construction and testing with the modules at the lowest level in the program structure. Since the modules are integrated from the bottom up, processing required for modules subordinate to a given level is always available and the need for stubs is eliminated. The bottom up integration strategy may be implemented with the following steps:

* The low-level modules are combined into clusters into clusters that perform a specific Software sub-function.
* A driver (i.e.) the control program for testing is written to coordinate test case input and output.
* The cluster is tested.
* Drivers are removed and clusters are combined moving upward in the program structure

The bottom-up approaches tests each module individually and then each module is module is integrated with a main module and tested for functionality.

**OTHER TESTING METHODOLOGIES**

**User Acceptance Testing**

User Acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes wherever required. The system developed provides a friendly user interface that can easily be understood even by a person who is new to the system.

**Output Testing**

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in the specified format. Asking the users about the format required by them tests the outputs generated or displayed by the system under consideration. Hence the output format is considered in 2 ways – one is on screen and another in printed format.

**Validation Checking**

Validation checks are performed on the following fields.

**Text Field:**

The text field can contain only the number of characters lesser than or equal to its size. The text fields are alphanumeric in some tables and alphabetic in other tables. Incorrect entry always flashes and error message.

**Numeric Field:**

The numeric field can contain only numbers from 0 to 9. An entry of any character flashes an error messages. The individual modules are checked for accuracy and what it has to perform. Each module is subjected to test run along with sample data. The individually tested modules are integrated into a single system. Testing involves executing the real data information is used in the program the existence of any program defect is inferred from the output. The testing should be planned so that all the requirements are individually tested.

A successful test is one that gives out the defects for the inappropriate data and produces and output revealing the errors in the system.

**Preparation of Test Data**

Taking various kinds of test data does the above testing. Preparation of test data plays a vital role in the system testing. After preparing the test data the system under study is tested using that test data. While testing the system by using test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

**Using Live Test Data:**

Live test data are those that are actually extracted from organization files. After a system is partially constructed, programmers or analysts often ask users to key in a set of data from their normal activities. Then, the systems person uses this data as a way to partially test the system. In other instances, programmers or analysts extract a set of live data from the files and have them entered themselves.

It is difficult to obtain live data in sufficient amounts to conduct extensive testing. And, although it is realistic data that will show how the system will perform for the typical processing requirement, assuming that the live data entered are in fact typical, such data generally will not test all combinations or formats that can enter the system. This bias toward typical values then does not provide a true systems test and in fact ignores the cases most likely to cause system failure.

**Using Artificial Test Data:**

Artificial test data are created solely for test purposes, since they can be generated to test all combinations of formats and values. In other words, the artificial data, which can quickly be prepared by a data generating utility program in the information systems department, make possible the testing of all login and control paths through the program.

The most effective test programs use artificial test data generated by persons other than those who wrote the programs. Often, an independent team of testers formulates a testing plan, using the systems specifications.

The package “Virtual Private Network” has satisfied all the requirements specified as per software requirement specification and was accepted.

**USER TRAINING**

Whenever a new system is developed, user training is required to educate them about the working of the system so that it can be put to efficient use by those for whom the system has been primarily designed. For this purpose the normal working of the project was demonstrated to the prospective users. Its working is easily understandable and since the expected users are people who have good knowledge of computers, the use of this system is very easy.

**MAINTAINENCE**

This covers a wide range of activities including correcting code and design errors. To reduce the need for maintenance in the long run, we have more accurately defined the user’s requirements during the process of system development. Depending on the requirements, this system has been developed to satisfy the needs to the largest possible extent. With development in technology, it may be possible to add many more features based on the requirements in future. The coding and designing is simple and easy to understand which will make maintenance easier.

**TESTING STRATEGY :**

A strategy for system testing integrates system test cases and design techniques into a well planned series of steps that results in the successful construction of software. The testing strategy must co-operate test planning, test case design, test execution, and the resultant data collection and evaluation .A strategy for software testing must accommodate low-level tests that are necessary to verify that a small source code segment has been correctly implemented as well as high level tests that validate major system functions against user requirements.

Software testing is a critical element of software quality assurance and represents the ultimate review of specification design and coding. Testing represents an interesting anomaly for the software. Thus, a series of testing are performed for the proposed system before the system is ready for user acceptance testing.

**SYSTEM TESTING:**

Software once validated must be combined with other system elements (e.g. Hardware, people, database). System testing verifies that all the elements are proper and that overall system function performance is achieved. It also tests to find discrepancies between the system and its original objective, current specifications and system documentation.

**UNIT TESTING:**

In unit testing different are modules are tested against the specifications produced during the design for the modules. Unit testing is essential for verification of the code produced during the coding phase, and hence the goals to test the internal logic of the modules. Using the detailed design description as a guide, important Conrail paths are tested to uncover errors within the boundary of the modules. This testing is carried out during the programming stage itself. In this type of testing step, each module was found to be working satisfactorily as regards to the expected output from the module.

In Due Course, latest technology advancements will be taken into consideration. As part of technical build-up many components of the networking system will be generic in nature so that future projects can either use or interact with this.The future holds a lot to offer to the development and refinement of this project.

**CONCLUSION**

In this article, we discussed an unexplored form of opinion spam, where spammers target brands as a whole, posting extreme reviews, to change the overall sentiment about the brand. These groups are often part of a complex business Web that is capable of influencing the overall popularity and reputation of several brands on review websites. This article is the first step toward linking brand-level group activities and extremism in reviews, which uncovers important insights about marketplace activities. These insights would help in developing a better recommendation that makes use of online reviews. A set of candidate spam groups was retrieved using FIM, and extremist groups were identified by observing their actions as a group based on various features, using a supervised learning technique based on a ground truth of manually annotated labels. We then classified extremist and moderate groups and compared the accuracy across multiple classification methods. After classifying these groups, we observed the behaviors for extremist groups in detail to gain further insights about the phenomenon and the overall trends of how these groups target these brands. We have also released the codes and annotated data set for further studies.

**REFERENCES**

[1] A. Kim. (2017). That review you wrote on Amazon? Priceless. [Online]. Available: https://www.usatoday.com/story/tech/news/ 2017/03/20/review-you-wrote-amazon-pricess/99332602/

[2] E. Gilbert and K. Karahalios, “Understanding deja reviewers,” in Proc. ACM Conf. Comput. supported Cooperat. Work (CSCW), 2010, pp. 225–228, doi: 10.1145/1718918.1718961.

[3] Amazon.in. (2018). Review Community Guidelines. [Online]. Available: https://www.amazon.in/gp/help/customer/display.html?nodeId= 201929730

[4] A. Mukherjee, B. Liu, and N. Glance, “Spotting fake reviewer groups in consumer reviews,” in Proc. 21st Int. Conf. World Wide Web (WWW), 2012, pp. 191–200.

[5] Y. Lu, L. Zhang, Y. Xiao, and Y. Li, “Simultaneously detecting fake reviews and review spammers using factor graph model,” in Proc. 5th Annu. ACM Web Sci. Conf. (WebSci), 2013, pp. 225–233.

[6] S. Rayana and L. Akoglu, “Collective opinion spam detection: Bridging review networks and metadata,” in Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD), 2015, pp. 985–994.

[7] S. Dhawan, S. C. R. Gangireddy, S. Kumar, and T. Chakraborty, “Spotting collective behaviour of online frauds in customer reviews,” 2019, arXiv:1905.13649. [Online]. Available: [http://arxiv.org/abs/1905. 13649](http://arxiv.org/abs/1905.%2013649)

[8] K. Dave, S. Lawrence, and D. M. Pennock, “Mining the peanut gallery: Opinion extraction and semantic classification of product reviews,” in Proc. 12th Int. Conf. World Wide Web (WWW), 2003, pp. 519–528.

[9] B. Pang, L. Lee, and S. Vaithyanathan, “Thumbs up? Sentiment classification using machine learning techniques,” in Proc. Conf. Empirical Methods Natural Lang. Process., 2002, pp. 79–86.

[10] K. Mouthami, K. N. Devi, and V. M. Bhaskaran, “Sentiment analysis and classification based on textual reviews,” in Proc. Int. Conf. Inf. Commun. Embedded Syst. (ICICES), Feb. 2013, pp. 271–276.

[11] Q. Ye, Z. Zhang, and R. Law, “Sentiment classification of online reviews to travel destinations by supervised machine learning approaches,” Expert Syst. Appl., vol. 36, no. 3, pp. 6527–6535, Apr. 2009.

[12] M. Chelliah and S. Sarkar, “Product recommendations enhanced with reviews,” in Proc. 11th ACM Conf. Recommender Syst., Aug. 2017, pp. 398–399.

[13] L. Chen and F. Wang, “Preference-based clustering reviews for augmenting e-commerce recommendation,” Knowl.-Based Syst., vol. 50, pp. 44–59, Sep. 2013.

[14] J. Feuerbach, B. Loepp, C.-M. Barbu, and J. Ziegler, “Enhancing an interactive recommendation system with review-based information filtering,” in Proc. IntRS@RecSys, 2017, pp. 10–55.

[15] A. Almahairi, K. Kastner, K. Cho, and A. Courville, “Learning distributed representations from reviews for collaborative filtering,” 2018, arXiv:1806.06875. [Online]. Available: <http://arxiv.org/abs/1806.06875>

[16] A.-M. Popescu and O. Etzioni, “Extracting product features and opinions from reviews,” in Proc. HLT, 2005, pp. 339–346.

[17] B. Liu, M. Hu, and J. Cheng, “Opinion observer: Analyzing and comparing opinions on the Web,” in Proc. 14th Int. Conf. World Wide Web (WWW), 2005, pp. 342–351.

[18] M. Hu and B. Liu, “Mining opinion features in customer reviews,” in Proc. AAAI, 2004, pp. 755–760.

[19] T. Donkers, B. Loepp, and J. Ziegler, “Explaining recommendations by means of user reviews,” in Proc. 1st Workshop Explainable Smart Syst. (ExSS), 2018. [Online]. Available: http://ceur-ws.org/Vol-2068/exss8.pdf [20] B. Pang et al., “Opinion mining and sentiment analysis,” Found. Trends Inf. Retr., vol. 2, nos. 1–2, pp. 1–135, 2008.

[21] M. Hu and B. Liu, “Mining and summarizing customer reviews,” in Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD), 2004, pp. 168–177.

[22] G. Somprasertsri and P. Lalitrojwong, “Mining feature-opinion in online customer reviews for opinion summarization,” J. UCS, vol. 16, no. 6, pp. 938–955, 2010.

[23] L. Zhuang, F. Jing, and X.-Y. Zhu, “Movie review mining and summarization,” in Proc. 15th ACM Int. Conf. Inf. Knowl. Manage. (CIKM), 2006, pp. 43–50.