

A Project report on
**NetSpam: A Network Based Spam Detection Framework For Reviews in
Online Social Media**

A Dissertation submitted to JNTU Hyderabad in partial fulfillment of the
academic requirements for the award of the degree.

Bachelor of Technology
in
Computer Science and Engineering

Submitted by

S. Sai Kiran Rao
(17H51A05B0)
G. Sai Shashank Goud
(19H51A05N2)
V. Laxmi Shivani
(20H55A0524)

Under the esteemed guidance of

Mr. Ch. Raja Kishore Babu
(Associate Professor)



Department of Computer Science and Engineering

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

(UGC Autonomous)

*Approved by AICTE *Affiliated to JNTUH *NAAC Accredited with A⁺ Grade

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD - 501401.

2019- 2023

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD – 501401

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



CERTIFICATE

This is to certify that the Major Project report entitled "**NetSpam: A Network Based Spam Detection Framework for Reviews in Online Social Media**" being submitted by S. Sai Kiran Rao (17H51A05B0), G. Sai Shashank Goud (19H51A05N2), V. Laxmi Shivani (20H55A0524) in partial fulfillment for the award of **Bachelor of Technology in Computer Science and Engineering** is a record of bonafide work carried out his/her under my guidance and supervision.

The results embodied in this project report have not been submitted to any other University or Institute for the award of any Degree.

Mr. Ch. Raja Kishore Babu
Associate Professor
Dept. of CSE

Dr. Siva Skandha Sanagala
Associate Professor & HOD
Dept. of CSE

ACKNOWLEDGEMENT

With great pleasure we want to take this opportunity to express my heartfelt gratitude to all the people who helped in making this project work a grand success.

We are grateful to **Mr. Ch. Raja Kishore Babu, Associate Professor, Project Guide,** Department of Computer Science and Engineering for his valuable technical suggestions and guidance during the execution of this project work.

We would like to thank **Dr. Siva Skandha Sanagala,** Head of the Department of Computer Science and Engineering, CMR College of Engineering and Technology, who is the major driving forces to complete my project work successfully.

We are very grateful to **Dr. Vijaya Kumar Koppula,** Dean-Academics, CMR College of Engineering and Technology, for his constant support and motivation in carrying out the project work successfully.

We are highly indebted to **Major Dr. V A Narayana,** Principal, CMR College of Engineering and Technology, for giving permission to carry out this project in a successful and fruitful way.

We would like to thank the **Teaching & Non- teaching** staff of Department of Computer Science and Engineering for their co-operation

We express our sincere thanks to **Shri. Ch. Gopal Reddy,** Secretary, CMR Group of Institutions, for his continuous care.

Finally, We extend thanks to our parents who stood behind us at different stages of this Project. We sincerely acknowledge and thank all those who gave support directly and indirectly in completion of this project work.

S. Sai Kiran Rao	(17H51A05B0)
G. Sai Shashank Goud	(19H51A05N2)
V. Laxmi Shivani	(20H55A0524)

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	LIST OF FIGURES	ii
	LIST OF TABLES	iii
	ABSTRACT	iv
1	INTRODUCTION	1
	1.1 Problem Statement	4
	1.2 Research Objective	4
	1.3 Project Scope and Limitations	4
2	BACKGROUND WORK	5
	2.1. Linguistic-Based methods	6
	2.2. Behavior-Based methods	6
	2.3. Heterogeneous-Based Methods	7
	2.3.1. Introduction	7
	2.3.2. Merits, Demerits and Challenges	7
	2.3.3. Implementation of Heterogeneous Information Network	8
	2.4. System analysis	11
	2.4.1. System specifications	12
	2.4.2. System study	13
	2.4.3. System testing	14
	2.4.4. Software environment	17
3	PROPOSED SYSTEM	35
	3.1. Objective of Proposed Model	36
	3.2. Algorithms Used for Proposed Model	36
	3.3. Designing	36
	3.3.1. Input design and output design	36
	3.3.2.UML Diagram	38
	3.3.2.1 Data flow diagram	38
	3.3.2.2. Flow chat	39
	3.3.2.3. Class diagram	41
	3.3.2.4. Sequence diagram	42

	3.3.2.5. Use case diagram	43
	3.4. Stepwise Implementation	45
4	RESULTS AND DISCUSSION	47
	4.1. Data collection	48
	4.2. Performance metrics	48
	4.3. Screen-shots	50
5	CONCLUSIONS	62
	5.1. Conclusion	63
	5.2. Future Enhancement	63
	REFERENCES	65
	GitHub Link	67

List of Figures

FIGURE

NO.	TITLE	PAGE NO.
1.2	Architecture	20

List of Tables

FIGURE

NO.	TITLE	PAGE NO.
2.1.2	TABLE I: Features for users and reviews in four defined categories	09
3	TABLE II: REVIEW DATASETS USED IN THIS WORK	12

ABSTRACT

Nowadays, a big part of people relies on available content in social media in their decision, for example, reviews and feedback on a topic or product. The possibility that anybody can leave a review provides a golden opportunity for spammers to write spam reviews about products and services for different interests. Identifying these spammers and the spam content is a hot topic of research and although a considerable number of studies have been done recently toward this end, but so far, the methodologies put forth still barely detect spam reviews, and none of them show the importance of each extracted feature type. In this study, we propose a novel framework, named Net Spam, which utilizes spam features for modeling review datasets as heterogeneous information networks to map spam detection procedure into a classification problem in such networks. Using the importance of spam features help us to obtain better results in terms of different metrics experimented on real-world review datasets from Yelp and Amazon websites. The results show that NetSpam outperforms the existing methods and among four categories of features; including review-behavioral, user-behavioral, review linguistic, user-linguistic, the first type of features performs better than the other categories.

CHAPTER 1

INTRODUCTION

CHAPTER 1

INTRODUCTION

Online Social Media portals play an influential role in information propagation which is considered as an important source for producers in their advertising campaigns as well as for customers in selecting products and services. In the past years, people relied a lot on the written reviews in their decision-making processes, and positive/negative reviews encouraged/discouraged them in their selection of products and services. In addition, written reviews also help service providers to enhance the quality of their products and services. These reviews thus have become an important factor in the success of a business. While positive reviews can bring benefits for a company, negative reviews can potentially impact credibility and cause economic losses. The fact that anyone with any identity can leave comments as review, provides a tempting opportunity for spammers to write fake reviews designed to mislead users' opinion. These misleading reviews are then multiplied by the sharing function of social media and propagation over the web. The reviews written to change users' perception of how good a product or a service are considered as spam, and are often written in exchange for money. The 20% of the reviews in the Yelp website are actually spam reviews. On the other hand, a considerable amount of literature has been published on the techniques used to identify spam and spammers as well as different types of analysis on this topic. These techniques can be classified into different categories; some using linguistic patterns in text, which are mostly based on bigram, and unigram, others are based on behavioral patterns that rely on features extracted from patterns in users' behavior which are mostly metadata based, and even some techniques using graphs and graph-based algorithms and classifiers.

Despite this great deal of efforts, many aspects have been missed or remained unsolved. One of them is a classifier that can calculate feature weights that show each feature's level of importance in determining spam reviews. The general concept of our proposed framework is to model a given review dataset as a Heterogeneous Information Network (HIN) and to map the problem of spam detection into a HIN classification problem. In particular, we model the review dataset as a HIN in which reviews are connected through different node types (such as

features and users). A weighting algorithm is then employed to calculate each feature's importance (or weight). These weights are utilized to calculate the final labels for reviews using both unsupervised and supervised approaches.

To evaluate the proposed solution, we used two sample review datasets from Yelp and Amazon websites. Based on our observations, defining two views for features (review-user and behavioral-linguistic), the classified features as review behavioral have more weights and yield better performance on spotting spam reviews in both semi-supervised and unsupervised approaches. In addition, we demonstrate that using different supervisions such as 1%, 2.5% and 5% or using an unsupervised approach, make no noticeable variation on the performance of our approach. We observed that feature weights can be added or removed for labeling and hence time complexity can be scaled for a specific level of accuracy. As a result of this weighting step, we can use fewer features with more weights to obtain better accuracy with less time complexity.

In addition, categorizing features in four major categories (review-behavioral, user-behavioral, review linguistic, user-linguistic), helps us to understand how much each category of features contributes to spam detection. In summary, our main contributions are as follows:

- We propose a NetSpam framework that is a novel networkbased approach which models review networks as heterogeneous information networks. The classification step uses different metapath types which are innovative in the spam detection domain.
- A new weighting method for spam features is proposed to determine the relative importance of each feature and shows how effective each of features are in identifying spam from normal reviews. Previous works [12], [20] also aimed to address the importance of features mainly in terms of obtained accuracy, but not as a built-in function in their framework (i.e., their approach is dependent on ground truth for determining each feature importance). As we explain in our unsupervised approach, NetSpam is able to find features of importance even without ground truth, and only by relying on metapath definition and based on values calculated for each review.
- NetSpam improves the accuracy compared to the state-of-the-art in terms of time complexity, which highly depends on the number of features used to identify a spam

review; hence, using features with more weights will result in detecting fake reviews easier with less time complexity.

1.1. Problem Statement

NetSpam is able to find features of importance even without ground truth, and only by relying on metapath definition and based on values calculated for each review. NetSpam improves the accuracy compared to the state-of-the-art in terms of time complexity, which highly depends on the number of features used to identify a spam review; hence, using features with more weights will result in detecting fake reviews easier with less time complexity. A new Content Based Algorithm for spam features is proposed to determine the relative importance of each feature and shows how effective each of features are in identifying spam from normal reviews.

1.2 Research Objective

We propose a NetSpam framework that is a novel network-based approach which models review networks as heterogeneous information networks. The classification step uses different metapath types which are innovative in the spam detection domain.

1.3 Project Scope and Limitations

- Social Media
- Data Recognition
- Classification of Reviews (Positive, Negative, Fake)

CHAPTER-2

BACKGROUND

WORK

CHAPTER-2

BACKGROUND WORK

2.1. Linguistic-Based Methods

This approach extracts linguistic-based features to find spam reviews. Feng et al. use unigram, bigram and their composition. Other studies use other features like pairwise features (features between two reviews; e.g. content similarity), percentage of CAPITAL words in a review for finding spam reviews. Lai et al. use a probabilistic language modeling to spot spam. This study demonstrates that 2% of reviews written on business websites are actually spam.

2.2. Behavior-Based Methods

Approaches in this group almost use reviews metadata to extract features; those which are normal patterns of a reviewer's behaviors. Feng et al. in focus on distribution of spammers rating on different products and trace them. In Jindal et. all extract 36 behavioral features and use a supervised method to find spammers on Amazon and indicates behavioral features show spammers' identity better than linguistic ones. Xue et al. in use rate deviation of a specific user and use a trust-aware model to find the relationship between users for calculating final spamicity score. Minnich et al. use temporal and location features of users to find unusual behavior of spammers. Li et al. use some basic features (e.g polarity of reviews) and then run a HNC (Heterogeneous Network Classifier) to find final labels on Dianpings dataset. Mukherjee et al. almost engage behavioral features like rate deviation, extremity etc. Xie et al. also use a temporal pattern (time window) to find singleton reviews (reviews written just once) on Amazon. Luca and Zervas use behavioral features to show increasing competition between companies leads to very large expansion of spam reviews on products. Crawford et al. indicates using different classification approaches need different numbers of features to attain desired performance and propose approaches which use fewer features to attain that performance and hence recommend to improve their performance while they use fewer features which leads them to have better complexity. With this perspective our framework is arguable. This study shows using different approaches in classification yield different performance in terms of different metrics.

2.3. Heterogeneous Information Network (HIN)

2.3.1. Introduction

Despite this great deal of efforts, many aspects have been missed or remained unsolved. One of them is a classifier that can calculate feature weights that show each feature's level of importance in determining spam reviews. A heterogeneous information network is a special type of information network with the underneath data structure as a directed graph, which either contains multiple types of objects or multiple types of links.

Heterogeneous Information Network: Suppose we have $r(> 1)$ types of nodes and $s(> 1)$ types of relation links between the nodes, then a heterogeneous information network is defined as a graph $G = (V, E)$ where each node $v \in V$ and each link $e \in E$ belongs to one particular node type and link type respectively. If two links belong to the same type, the types of starting node and ending node of those links are the same.

2.3.2. Merits and Demerits Advantages of Existing systems:

Advantages of Existing systems:

1. To identify spam and spammers as well as different type of analysis on this topic.
2. Written reviews also help service providers to enhance the quality of their products and services.
3. To identify the spam user using positive and negative reviews in online social media.
4. To display only trusted reviews to the users

Disadvantages of Existing system:

1. This utilizes spam features for modeling review datasets as heterogeneous information networks to map spam detection procedure into a classification problem in such networks.
2. Time Complexity.

2.3.3. Implementation of Heterogeneous Information Network (HIN)

The general concept of our proposed framework is to model a given review dataset as a Heterogeneous Information Network (HIN) and to map the problem of spam detection into a HIN classification problem. In particular, we model the review dataset as a HIN in which reviews are connected through different node types. The general concept of our proposed framework is to model a given review dataset as a Heterogeneous Information Network and to map the problem of spam detection into a HIN classification problem. In particular, we model

review dataset as in which reviews are connected through different node types. A weighting algorithm is then employed to calculate each feature's importance. These weights are utilized to calculate the final labels for reviews using both unsupervised and supervised approaches. Classification problem in heterogeneous information networks:

Given a heterogeneous information network $G = (V, E)$, suppose V' is a subset of V that contains nodes of the target type (i.e., the type of nodes to be classified). k denotes the number of the class, and for each class, say $C_1 \dots C_k$, we have some pre-labeled nodes in V' associated with a single user. The classification task is to predict the labels for all the unlabeled nodes in V' . A metapath is defined as a path between two nodes, which indicates the connection of two nodes through their shared features. When we talk about metadata, we refer to its general definition, which is data about data. In our case, the data is the written review, and by metadata we mean data about the reviews, including user who wrote the review, the business that the review is written for, rating value of the review, date of written review and finally its label as spam or genuine review.

In particular, in this work features for users and reviews fall into the categories as follows (shown in Table I):

1) Review-Behavioral (RB) Based Features: This feature type is based on metadata and not the review text itself. The RB category contains two features; Early time frame (ETF) and Threshold rating deviation of review.

2) Review-Linguistic (RL) Based Features: Features in this category are based on the review itself and extracted directly from the text of the review. In this work we use two main features in the RL category; the Ratio of 1st Personal Pronouns (PP1) and the Ratio of exclamation sentences containing '!'.

3) User-Behavioral (UB) Based Features: These features are specific to each individual user and they are calculated per user, so we can use these features to generalize all of the reviews written by that specific user. This category has two main features; the Burstiness of reviews written by a single user, and the average of a users' negative ratio given to different

businesses.

4) User-Linguistic (UL) Based Features: These features are extracted from the users' language and shows how users are describing their feeling or opinion about what they've experienced as a customer of a certain business. We use this type of feature to understand how a spammer communicates in terms of wording. There are two features engaged for our framework in this category; Average Content Similarity (ACS) and Maximum Content Similarity (MCS). These two features show how much two reviews written by two different users are similar to each other, as spammers tend to write very similar reviews by using template pre-written text.

TABLE I: Features for users and reviews in four defined categories

Spam Feature	User-based	Review-based
Behavioral based Features	<p>Burstiness: Spammers usually write their spam reviews in a short period of time for two reasons: first, because they want to impact readers and other users, and second because they are temporal users, they have to write as many reviews as they can in a short time.</p> $x_{BST}(i) = \begin{cases} 0 & (L_i - F_i) \notin (0, \tau) \\ 1 - \frac{L_i - F_i}{\tau} & (L_i - F_i) \in (0, \tau) \end{cases} \quad (1)$ <p>where $L_i - F_i$ describes days between last and first review for $\tau = 28$. Users with a calculated value greater than 0.5 take value 1 and others take 0.</p> <p>Negative Ratio: Spammers tend to write reviews which defame</p>	<p>Early Time Frame: Spammers try to write their reviews asap, in order to keep their review in the top reviews which other users visit them sooner.</p> $x_{ETF}(i) = \begin{cases} 0 & (T_i - F_i) \notin (0, \delta) \\ 1 - \frac{T_i - F_i}{\delta} & (T_i - F_i) \in (0, \delta) \end{cases} \quad (2)$ <p>where $L_i - F_i$ denotes days specified written review and first written review for a specific business. We also have $\delta = 7$. Users with a calculated value greater than 0.5 take value 1 and others take 0.</p> <p>Rate Deviation using threshold: Spammers also tend to promote businesses they have contracts with, so they rate these businesses</p>

	<p>businesses which are competitors with the ones they have contract with, this can be done with destructive reviews, or with rating those businesses with low scores. Hence, the ratio of their scores tends to be low. Users with an average rate equal to 2 or 1 take 1 and others take 0.</p>	<p>with high scores. In result, there is high diversity in their given scores to different businesses which is the reason they have high variance and deviation.</p> $xDEV(i) = \begin{cases} 0 & \text{otherwise} \\ \frac{1 - r_{ij} - \text{avge} \in E * j}{r(e)} & 4 > \beta_1 \end{cases} \quad (3)$ <p>where β_1 is some threshold determined by recursive minimal entropy partitioning. Reviews are close to each other based on their calculated value, take the same values (in $[0, 1)$).</p>
Linguistic based Features	<p>Average Content Similarity, Maximum Content Similarity: Spammers often write their reviews with the same template and they prefer not to waste their time to write an original review. As a result, they have similar reviews. Users who have close calculated values take the same values (in $[0, 1)$).</p>	<p>Number of first-Person Pronouns, Ratio of Exclamation Sentences containing '!': First, studies show that spammers use second personal pronouns much more than first personal pronouns. In addition, spammers put '!' in their sentences as much as they can to increase impressions on users and highlight their reviews among other ones. Reviews are close to each other based on their calculated value, take the same values (in $[0, 1)$).</p>

2.4. System analysis

EXISTING SYSTEM

In today's world everything has become very fast due to internet. As there are too many social networking sites hence people are interacting with each other across the world. They can share their ideas on internet. Also internet provides the facility of online shopping, so related to this on company's website or some review web sites such as Amazon, dhgate, Ebay, flipcart, Yelp

and many more provides lots of reviews about products. Before purchasing any product, it is a normal human behavior to do a survey on that. Hence these websites are helpful to the people to check quality of product. Based on available reviews customer can compare different brands of product and can buy a product. Hence these reviews will change the mind set of customer. If these reviews are true then it can help customer to select proper product satisfying their requirements. Similarly, if reviews are false or not true then it can yield wrong information to customers.

DISADVANTAGES

Generally, we define review manipulation as publishers, writers, authors or company people or any third-party those who writing bad comments or feedback on behalf of customer when needed, to maximize their sales of productivity. A customer review contains two parts, first one is rating with stars and second is with textual comments. If unauthorized user posts comments, then he may either give maximum rating to the product or can manipulate textual comment. So by analyzing and concluding writing behavior of customer we can identify fake reviews.

PROPOSED SYSTEM

This system proposes the method to recognize the untruthful reviews that are given by the consumers which is having different semantic content based on sentiment analysis as the reviews of particular product. This system proposes a behavioral approach to identify review spammers those who are trying to manipulate the ratings on some products. Here we derive an aggregated behavior methods for rank reviewers based on the degree that they have demonstrated the spamming behaviors. So as to verify our proposed methods, which conducts user evaluation on an Amazon dataset which contains reviews of different company's products? It is found the proposed method generally outperform the baseline method based votes. Also we learnt a regression model from the consumer ground truth spammers. The feedback and viewpoints are used by web users and companies for the manufacturing of new products.

ADVANTAGES

This paper mainly focuses on review centric spam identification which provides greater focus on feedback content. As part of future work, we can enhance review spammer identification into the review identification and vice versa. Exploring different ways to learn behavior patterns which are related to the spamming so as to improve the accuracy of the current

regression model is an interesting research direction in current era.

2.4.1. System specifications

HARDWARE REQUIREMENTS:

- System : Pentium IV 2.4 GHz.
- Hard Disk : 40 GB.
- Floppy Drive : 1.44 Mb.
- Monitor : 15 VGA Colour.
- Mouse : Logitech.
- Ram : 512 Mb.

SOFTWARE REQUIREMENTS:

- Operating system : Windows XP/7/8/10.
- Coding Language : JAVA 1.8/J2EE (Web Application)
- Data Base : MYSQL
- Web server : Apache Tomcat 8.1
- IDE Tool : NETBEANS 8.1
- Client-Side Technologies : HTML, CSS and JAVASCRIPT
- Server-Side Technologies : JSP (Java Server Pages)
- Database Connectivity : JDBC
- Other tolls : sqlyog607

2.4.2. System study

FEASIBILITY STUDY

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

Three key considerations involved in the feasibility analysis are

ECONOMICAL FEASIBILITY

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.

2.4.3. 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 the software tester has knowledge of the inner workings, structure and language of the software, or at least its 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 life-cycle, 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.

2.4.4. Software environment

Java Technology

Java technology is both a programming language and a platforms.

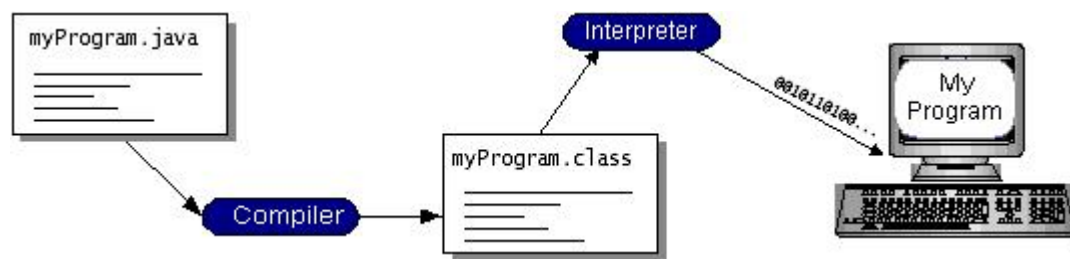
The Java Programming Language

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

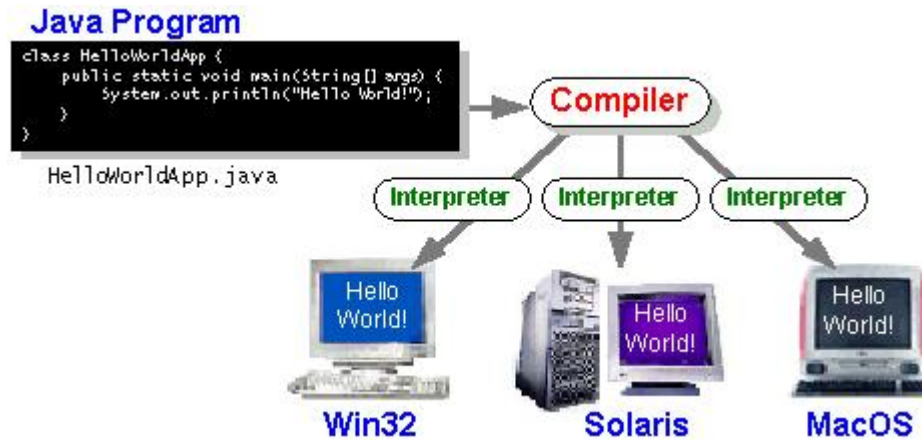
- Simple
- Architecture neutral
- Object oriented
- Portable
- Distributed
- High performance
- Interpreted

- Multi-threaded
- Robust
- Dynamic
- Secure

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



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



The Java Platform

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

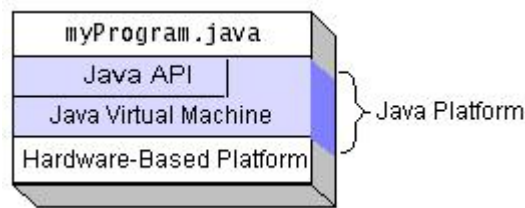
The Java platform has two components:

- The *Java Virtual Machine* (Java VM)
- The *Java Application Programming Interface* (Java API)

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

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

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



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

What Can Java Technology Do?

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

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

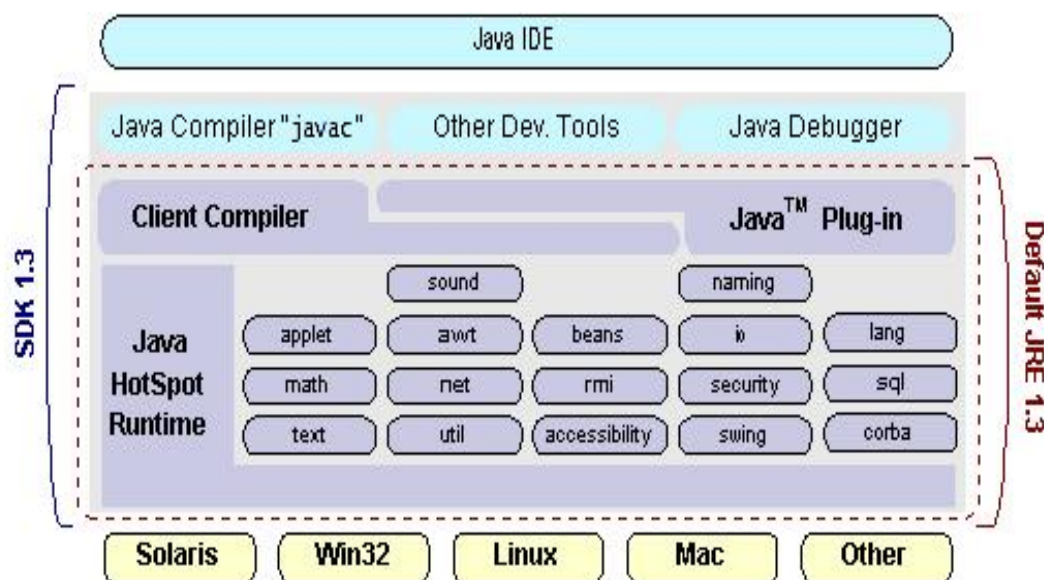
An application is a standalone program that runs directly on the Java platform. A special kind of application known as a *server* serves and supports clients on a network. Examples of servers are Web servers, proxy servers, mail servers, and print servers. Another specialized program is a *servlet*.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

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

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

- **Applets:** The set of conventions used by applets.
- **Networking:** URLs, TCP (Transmission Control Protocol), UDP (User Datagram 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 wide-ranging, easily extendible API let you reuse other people's tested code and introduce fewer bugs.
- **Develop programs more quickly:** Your development time may be as much as twice as fast versus writing the same program in C++. Why? You write fewer lines of code and it is a simpler programming language than C++.
- **Avoid platform dependencies with 100% Pure Java:** You can keep your program portable by avoiding the use of libraries written in other languages. The 100% Pure Java™ 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. This goal allows JDBC to use existing ODBC level drivers by the use of a software interface. This interface would translate JDBC calls to ODBC and vice versa.

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.

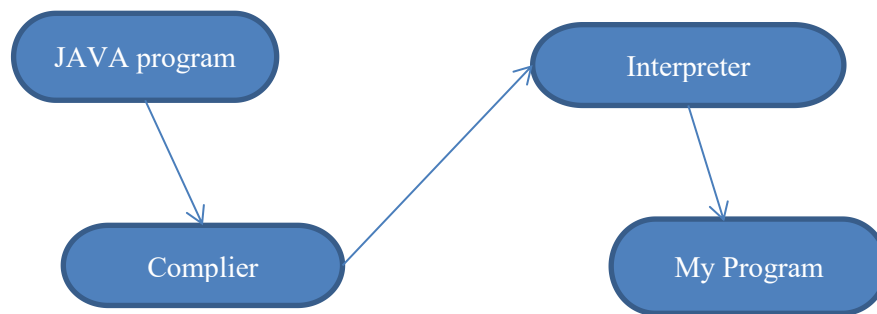
Finally we decided to proceed the implementation using Java [Networking](#). And for dynamically updating the cache table we go for MS [Access](#) database.

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

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

Simple	Architecture-neutral
Object-oriented	Portable
Distributed	High-performance
Interpreted	multithreaded
Robust	Dynamic
Secure	

Java is also unusual in that each Java program is both compiled and interpreted. With a compile you translate a Java program into an intermediate language called Java byte codes the platform-independent code instruction is passed and run on the computer. Compilation happens just once; interpretation occurs each time the program is executed. The figure illustrates how this works.



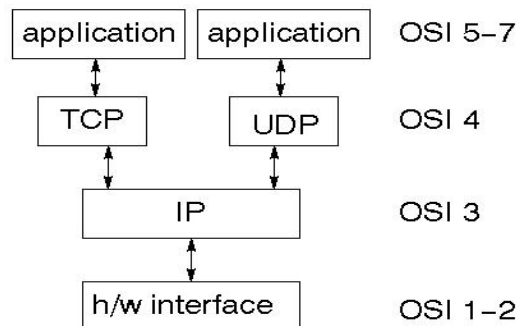
You can think of Java byte codes as the machine code instructions for the Java Virtual Machine (Java VM). Every Java interpreter, whether it's a Java development tool or a Web browser that can run Java applets, is an implementation of the Java VM. The Java VM can also be implemented in hardware.

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

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 and class D uses all 32.

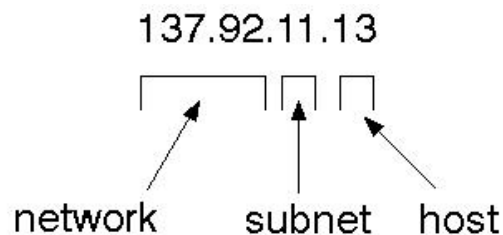
Subnet address

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. The tasks in this project include:

Sourcing freely redistributable vector outlines for the countries of the world, states/provinces in particular countries (USA in particular, but also other areas);

Creating an appropriate dataset interface (plus default implementation), a rendered, and integrating this with the existing XYPlot class in JFreeChart;

Testing, documenting, testing some more, documenting some more.

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

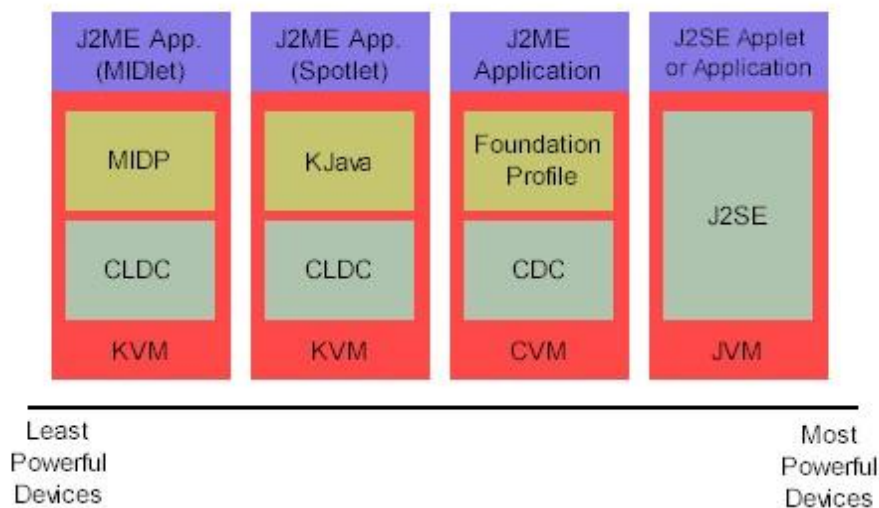
The property editor mechanism in JFreeChart only handles a small subset of the properties that can be set for charts. Extend (or reimplement) this mechanism to provide greater end-user control over the appearance of the charts.

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, screen-phones, 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 reverification 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:

- Keep it simple. Remove unnecessary features, possibly making those features a separate, secondary application.
- Smaller is better. This consideration should be a "no brainer" for all developers. Smaller applications use less memory on the device and require shorter installation times. Consider packaging your Java applications as compressed Java Archive (jar) files.
- Minimize run-time memory use. To minimize the amount of memory used at run time, use scalar types in place of object types. Also, do not depend on the garbage collector. You should manage the memory efficiently yourself by setting object references to null when you are finished with them. Another way to reduce run-time memory is to use lazy instantiation, only allocating objects on an as-needed basis. Other ways of reducing overall and peak memory use on small devices are to release resources quickly, reuse objects, and avoid exceptions.

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 Sun-specific 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

CHAPTER 3

PROPOSED SYSTEM

CHAPTER 3

PROPOSED SYSTEM

3.1. Objective of Proposed Model

We propose NetSpam framework that is a novel network-based approach which models review networks as heterogeneous information networks. The classification step uses different metapath types which are innovative in the spam detection domain.

3.2. Algorithms Used for Proposed Model

CONTEXT BASED ALGORITHM:

Context based algorithm making some prediction which is based on the history that history uses the some context which is out of order. Distribution is based on the history of sequence. We use that history to determine a sequence in a predictive manner, such scheme is known as context based algorithm.

PPM (Predictive with partial match) models is a technique based on context modelling and prediction. We only need to store those contexts that have occurred in the sequence which are being encoded. At the beginning of encoding, we will need to code letters that have not occurred previously in the context. We use an “escape” symbol is used to signal that the letters to be encoded as not been seen in the context.

Basic Algorithm: If the symbol has not occurred in the context, an escape symbol is encoded. This algorithm attempts to next smaller contexts. Each time a symbol is encountered, the count corresponding to that symbol is updated in all the tables.

3.3. Designing

3.3.1. Input design and output design

INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people

keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.
2. Select methods for presenting information.
3. Create document, report, or other formats that contain information produced by the system.

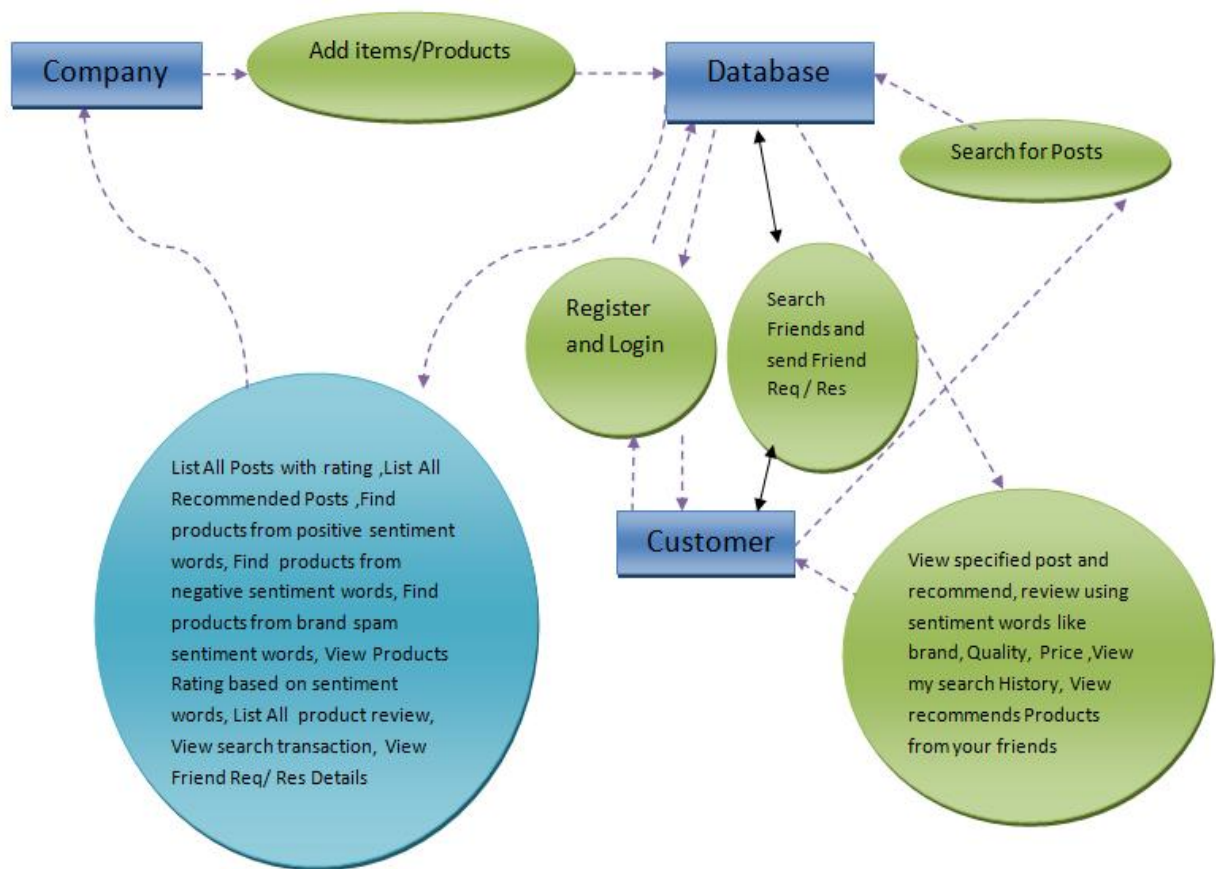
The output form of an information system should accomplish one or more of the following objectives.

- ❖ Convey information about past activities, current status or projections of the future.
- ❖ Signal important events, opportunities, problems, or warnings.
- ❖ Trigger an action.
- ❖ Confirm an action.

3.3.2.UML Diagram

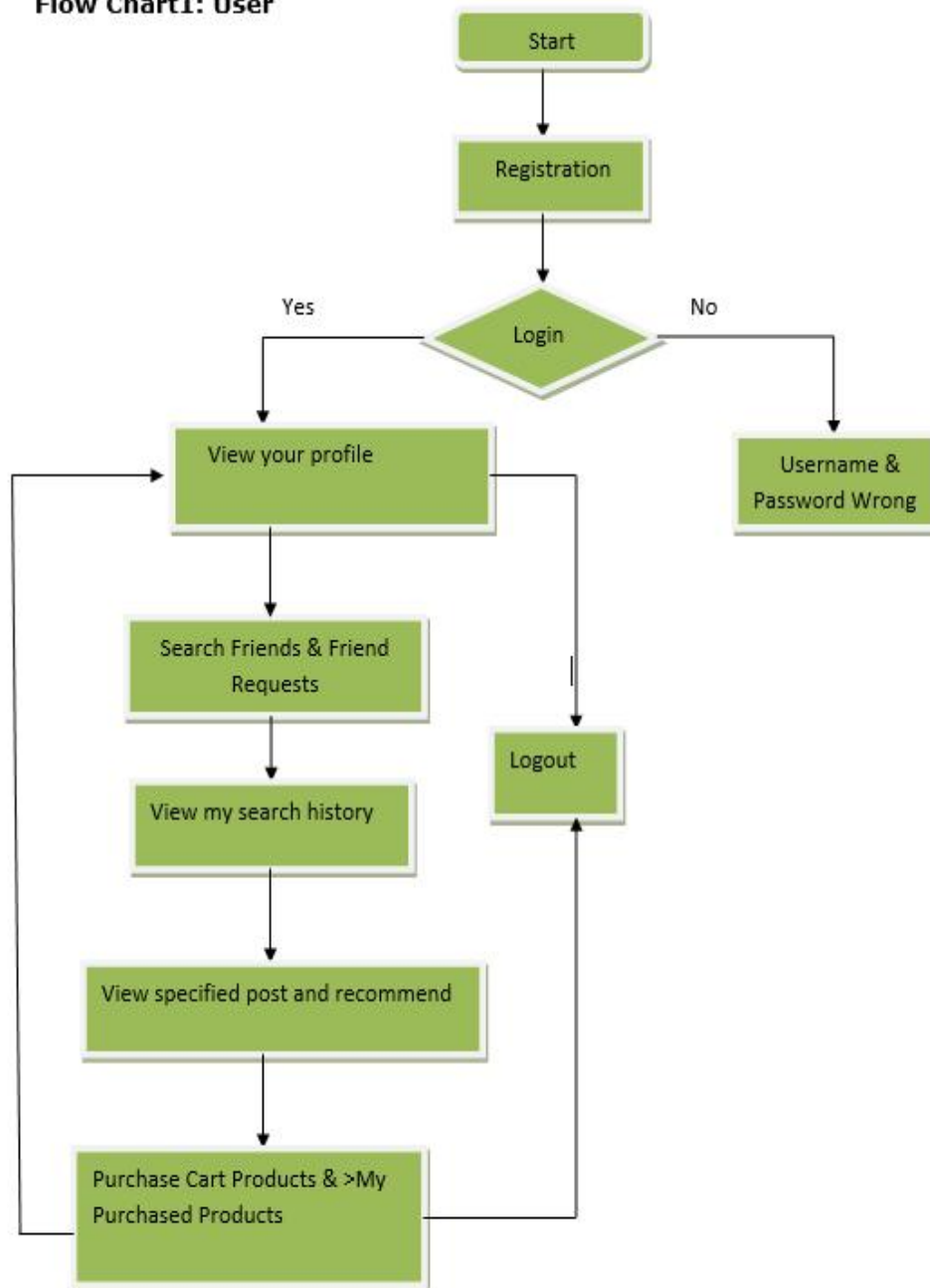
3.3.2.1 Data flow diagram

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

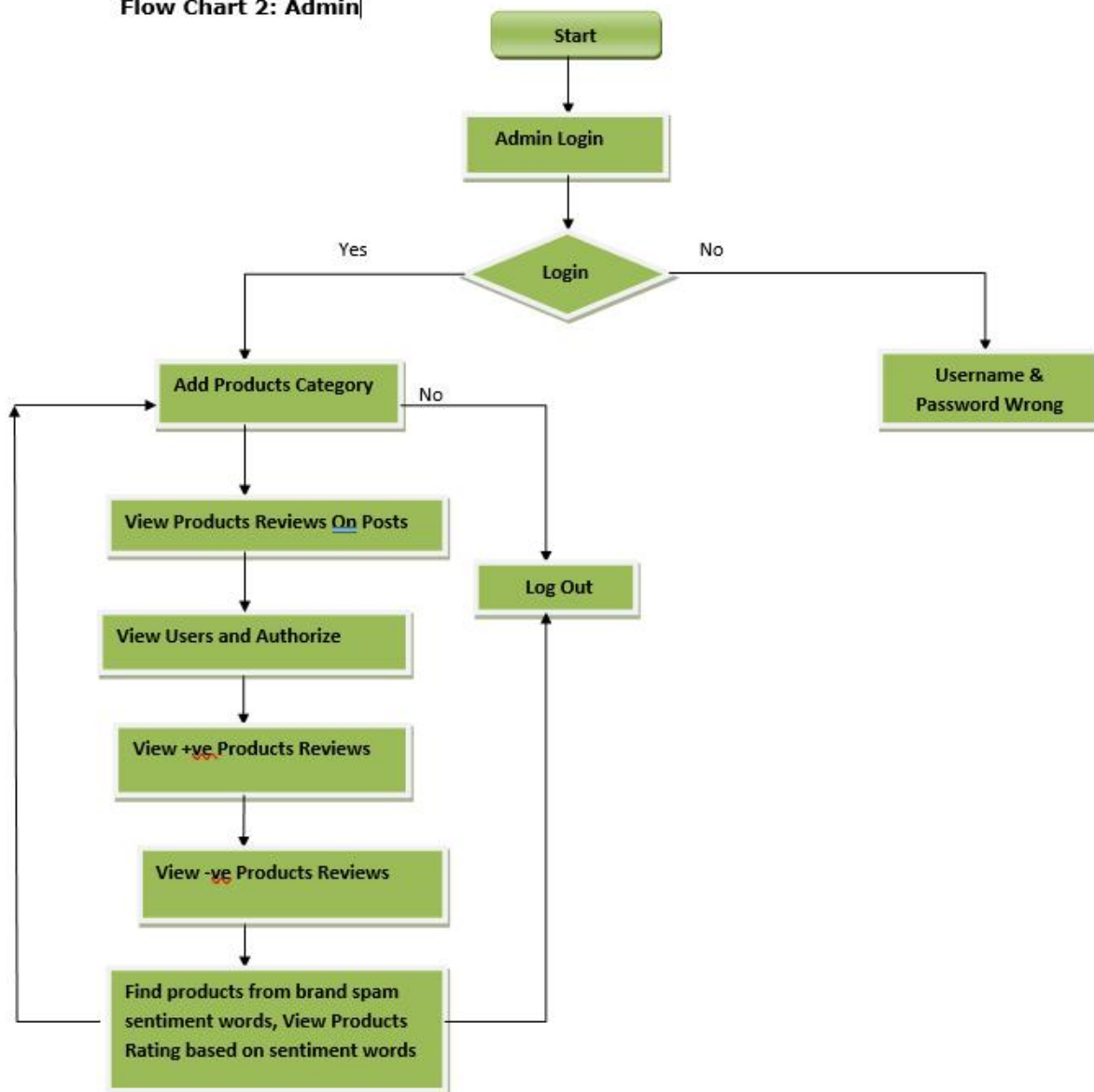


3.3.2.2. Flow chat

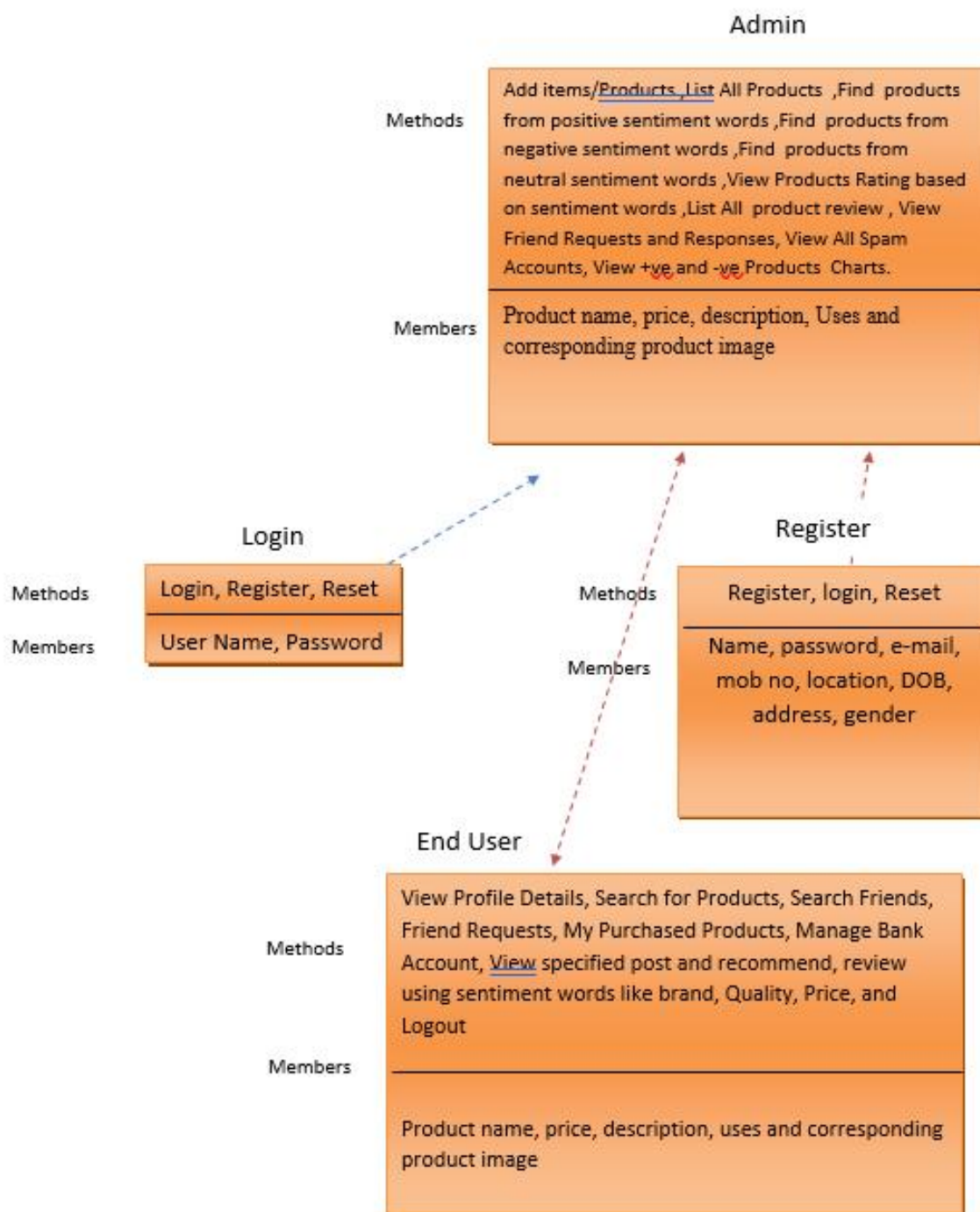
Flow Chart1: User



Flow Chart 2: Admin

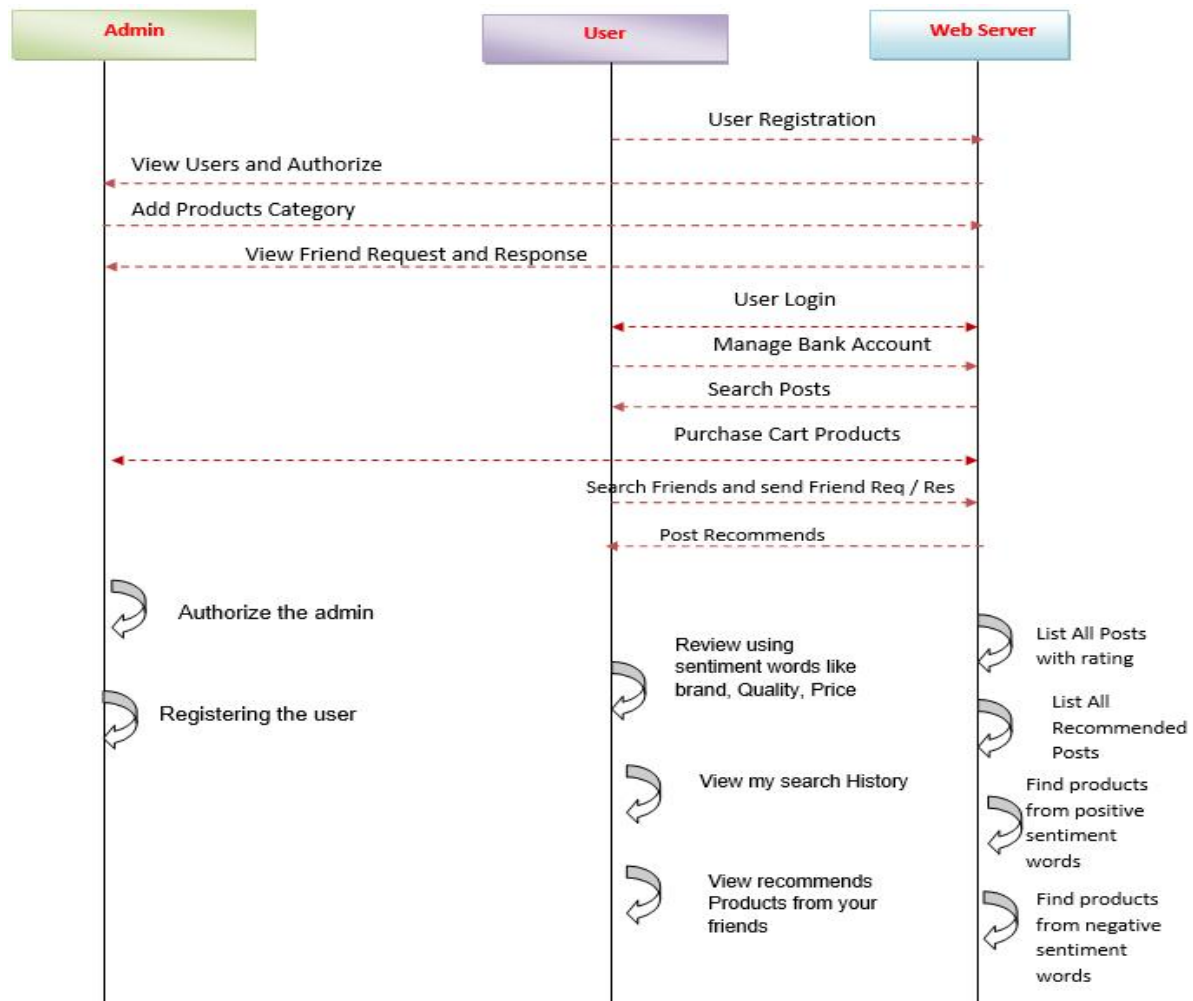


3.3.2.3. Class diagram



3.3.2.4. Sequence diagram

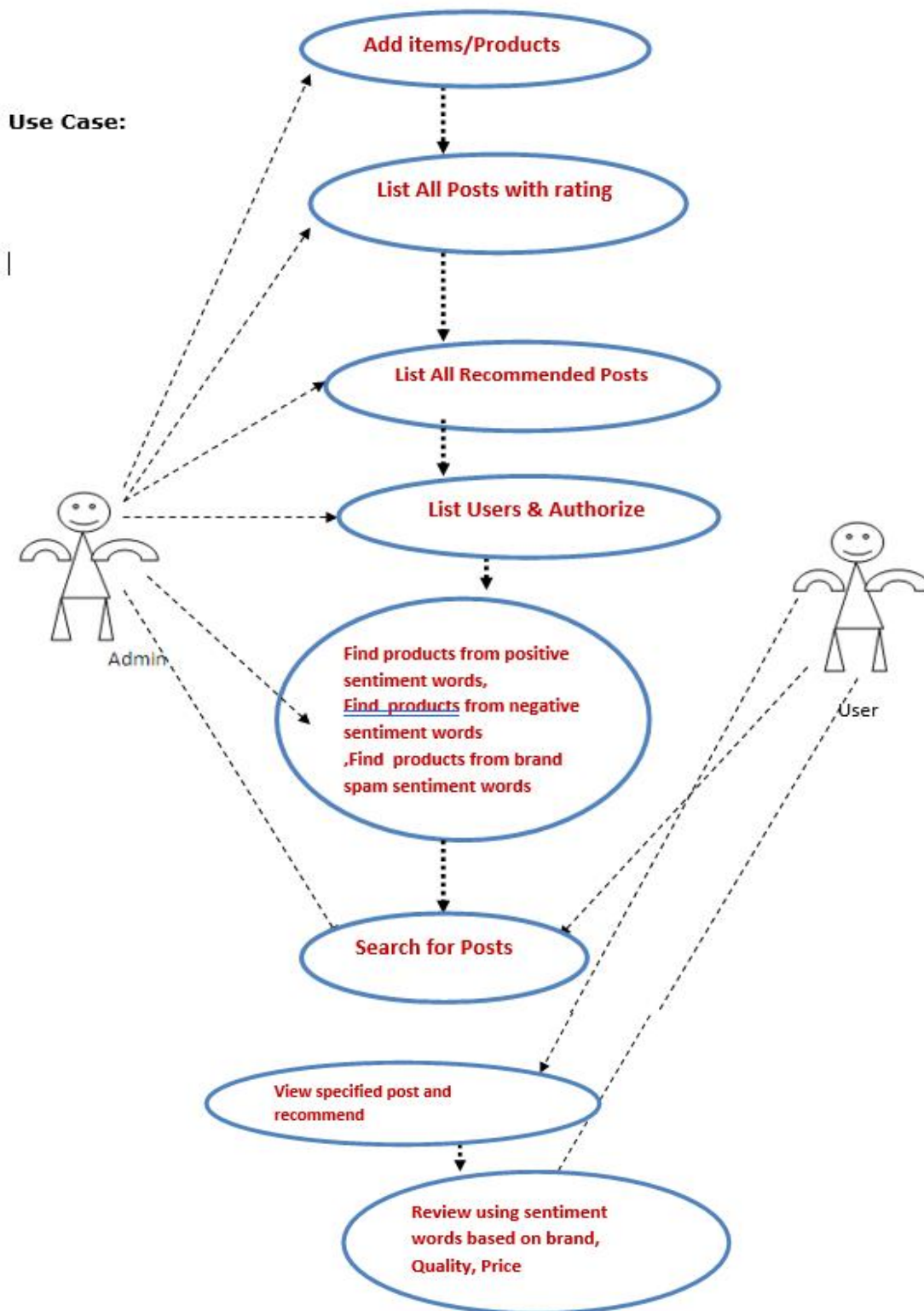
A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



3.3.2.5. Use case diagram

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

the system can be depicted.



3.4. Stepwise Implementation

MODULES DESCRIPTION

- **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 +ve and -ve Products Reviews, list of all reviewed posts, View Users and Authorize, list of all search history, update posts, View All Spam Accounts, View All Spam Reviews, View Recommended Posts, View +ve and -ve Products Charts, etc.

- **Classification of review (positive or negative) :**

Review contains more than one opinion sentence. For extracting opinion words from opinion sentence we have used Stanford NLP parser. Semantic orientation of opinion words is calculated using algorithm. Based on semantic orientation (SO) value classification of review is done, If SO value is positive then opinion is classified as positive and if value is negative opinion is classified as negative.

- **Classification of review (spam or non-spam) :**

Review has two main parts 1.Content and 2. Rating. Rating is very important in opinion. For this classification the data mining tool, naïve bayes algorithm is used. Further it generates ARFF (Attribute-Relation File Format) from the distinct features to detect the untruthful reviews using Support Count in Association Rules and detects Brands in Fake Reviews.

- **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 Friends, Friend Requests, Purchase Cart Products and logout.

- **Searches for good reviews and bad review**

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.

CHAPTER 4

RESULTS AND DISCUSSION

CHAPTER 4

RESULTS AND DISCUSSION

4.1. DATA COLLECTION

Data Collection We created three datasets as follow: - Review-based dataset, includes 10% of the reviews from the Main dataset, randomly selected using uniform distribution. - Item-based dataset, composed of 10% of the randomly selected reviews of each item, also based on uniform distribution (as with Review-based dataset). - User-based dataset, includes randomly selected reviews using uniform distribution in which one review is selected from every 10 reviews of a single user and if the number of reviews was less than 10, uniform distribution has been changed in order to at least one review from every user get selected. In addition to the presented dataset, we also used another real-world set of data from Amazon to evaluate our work on unsupervised mode. There is no credible label in the Amazon dataset, but we used this dataset to show how much our idea is viable on other datasets beyond Yelp and results for this dataset are presented on Sec. IV-C3.

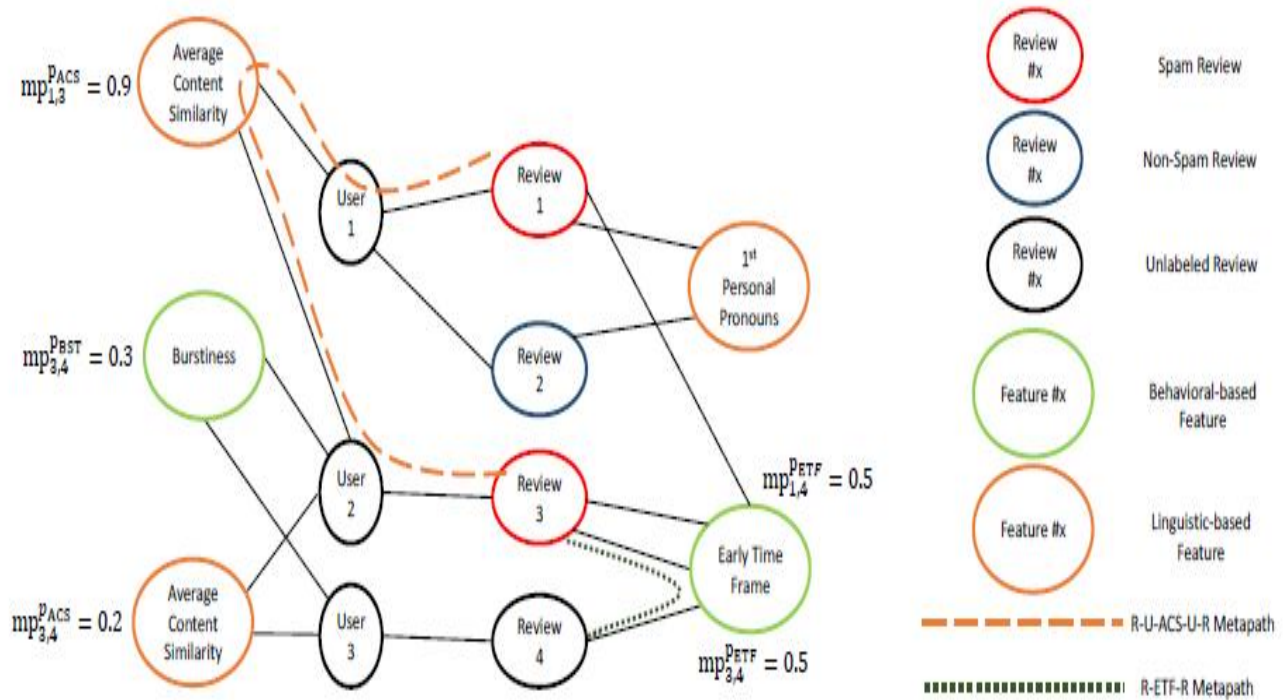
TABLE II : REVIEW DATASETS USED IN THIS WORK

Data Set	Reviews (Spam%)	Users	Business(Resto. & hotels)
Main	608,598 (13%)	260,277	5,044
Review-based	62,990 (13%)	48,121	3,278
Item-based	66,841 (34%)	52,453	4,588
User-based	183,963(19%)	150,278 4,568 Amazon 8,160(-) 7,685 243	4,568
Amazon	8,160(-)	7,685	243

4.2. Performance metrics

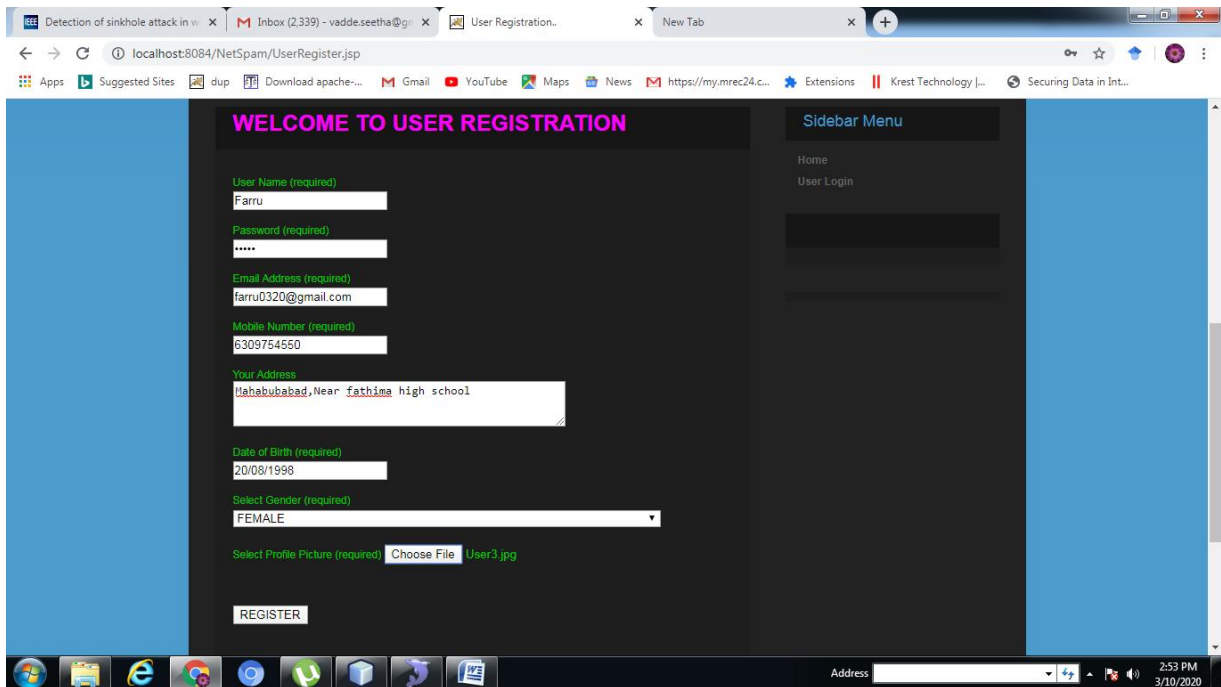
We have used Average Precision (AP) and Area Under the Curve (AUC) as two metrics in our evaluation. AUC measures the accuracy of our ranking based on False Positive Ratio (FPR as y-axis) against True Positive Ratio (TPR as x-axis) and integrates values based on

these two measured values. The value of this metric increases as the proposed method performs well in ranking, and vise-versa. Let A be the list of sorted spam reviews so that $A(i)$ denotes a review sorted on the i th index in A . If the number of spam (non-spam) reviews before review in the j th index is equal to n_j and the total number of spam (non-spam) reviews is equal to f , then TPR (FPR) for the j th is computed as n_j/f . To calculate the AUC , we set TPR values as the x -axis and FPR values on the y -axis and then integrate the area under the curve for the curve that uses their values. We obtain a value for the AUC using: $AUC = \sum_{i=2}^n (FPR(i) - FPR(i-1)) * (TPR(i))$ -- (7) where n denotes number of reviews. For AP we first need to calculate the index of top sorted reviews with spam labels. Let indexes of sorted spam reviews in list A with spam labels in ground truth be like list I , then for AP we have: $AP = \sum_{i=1}^n i / I(i)$ --(8)



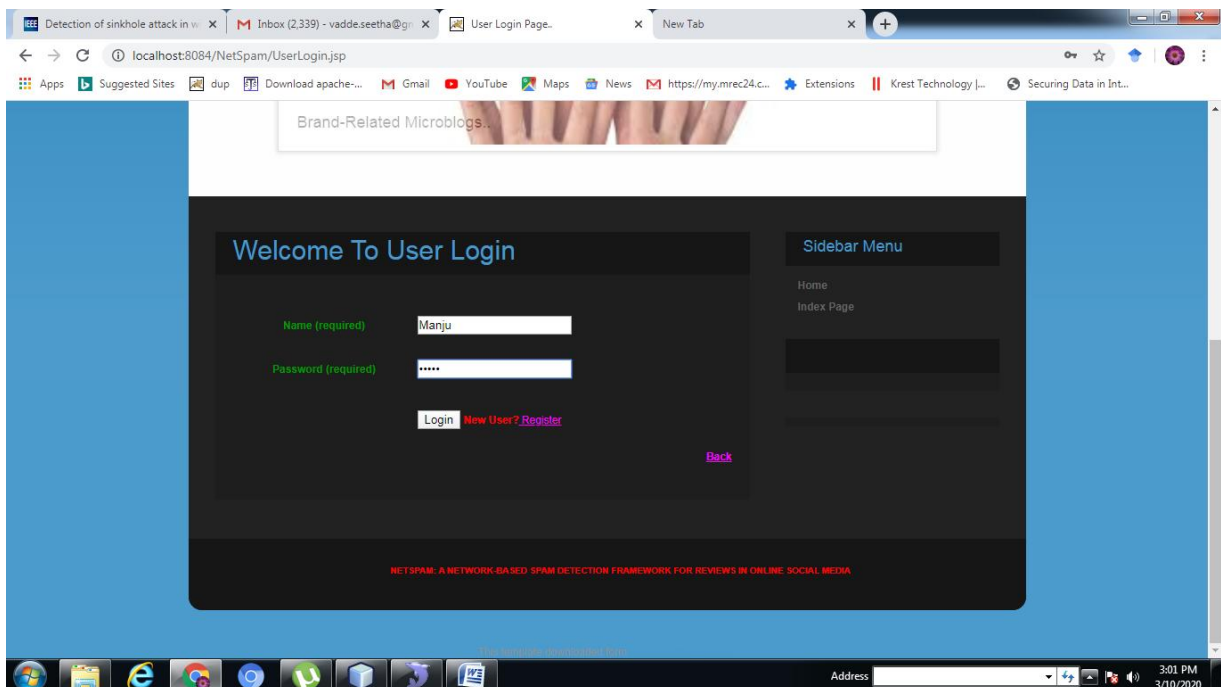
4.3. SCREENSHOTS:

User registration form:



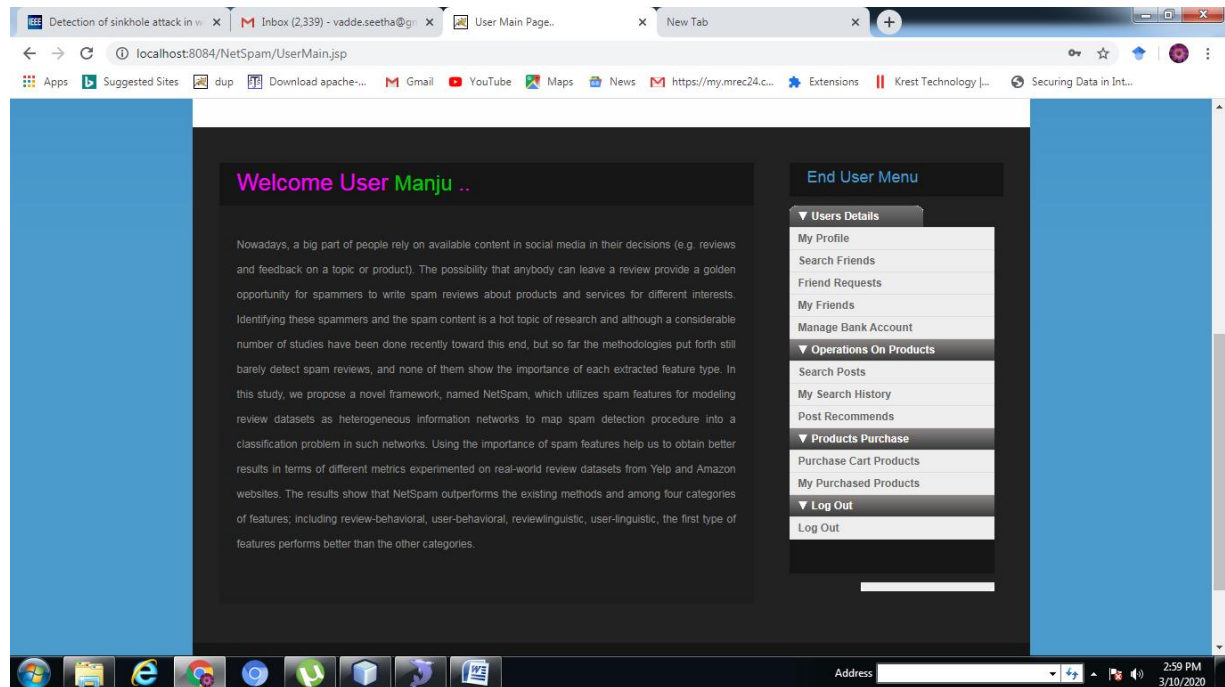
The screenshot shows a web browser window with the URL `localhost:8084/NetSpam/UserRegister.jsp`. The page has a dark blue header with the text "WELCOME TO USER REGISTRATION" in pink. On the right, there is a "Sidebar Menu" with links for "Home" and "User Login". The main content area contains a registration form with the following fields: "User Name (required)" with the value "Faru", "Password (required)" with masked characters "*****", "Email Address (required)" with the value "faru0320@gmail.com", "Mobile Number (required)" with the value "6309754550", "Your Address" with the value "Mahabubabad, Near fathima high school", "Date of Birth (required)" with the value "20/08/1998", "Select Gender (required)" with a dropdown menu showing "FEMALE", and "Select Profile Picture (required)" with a "Choose File" button and the filename "User3.jpg". A "REGISTER" button is at the bottom of the form. The browser's address bar shows the URL, and the taskbar at the bottom displays various application icons and the system clock showing 2:53 PM on 3/10/2020.

User Login

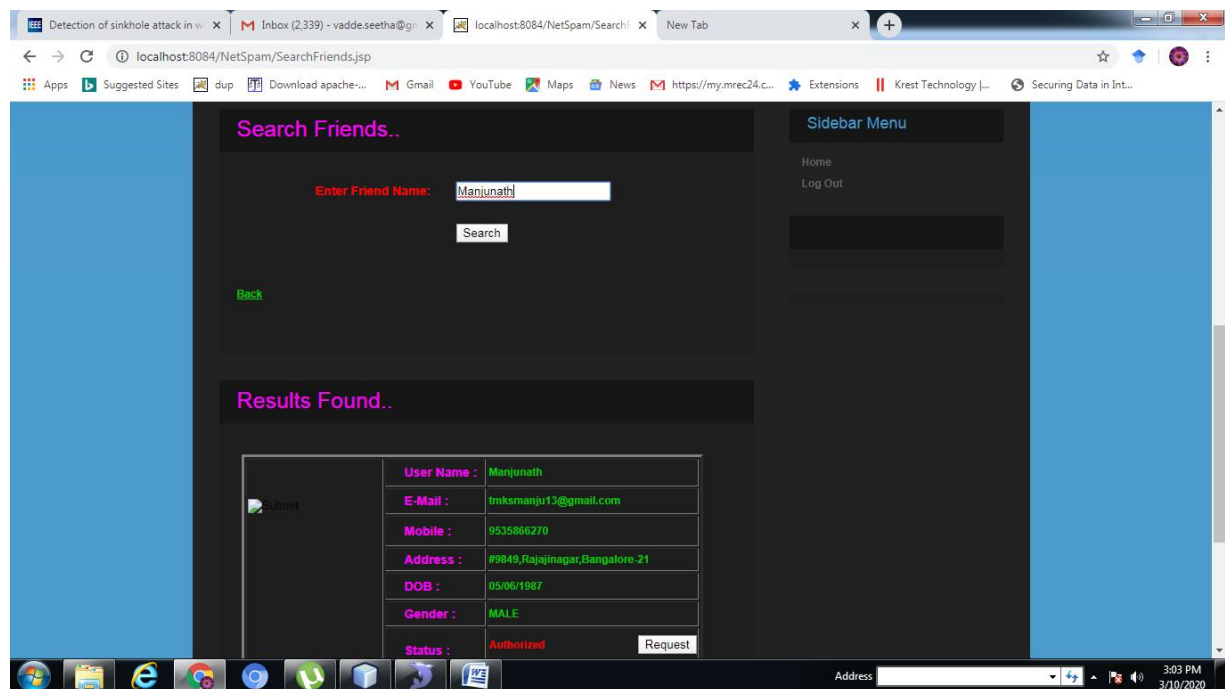


The screenshot shows a web browser window with the URL `localhost:8084/NetSpam/UserLogin.jsp`. The page has a dark blue header with the text "Welcome To User Login" in white. On the right, there is a "Sidebar Menu" with links for "Home" and "Index Page". The main content area contains a login form with the following fields: "Name (required)" with the value "Manju", "Password (required)" with masked characters "*****", and a "Login" button. Below the login button, there are links for "New User? Register" and "Back". At the bottom of the page, there is a red text line that reads "NETSPAM: A NETWORK-BASED SPAM DETECTION FRAMEWORK FOR REVIEWS IN ONLINE SOCIAL MEDIA". The browser's address bar shows the URL, and the taskbar at the bottom displays various application icons and the system clock showing 3:01 PM on 3/10/2020.

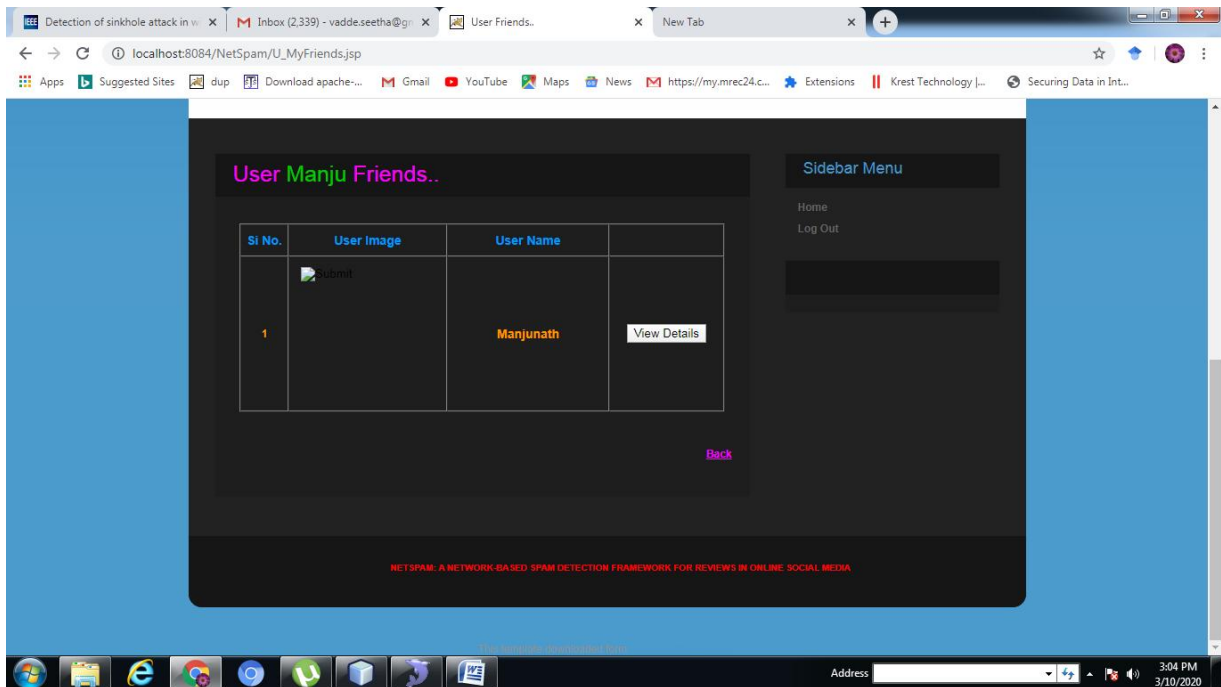
Welcome user Manju



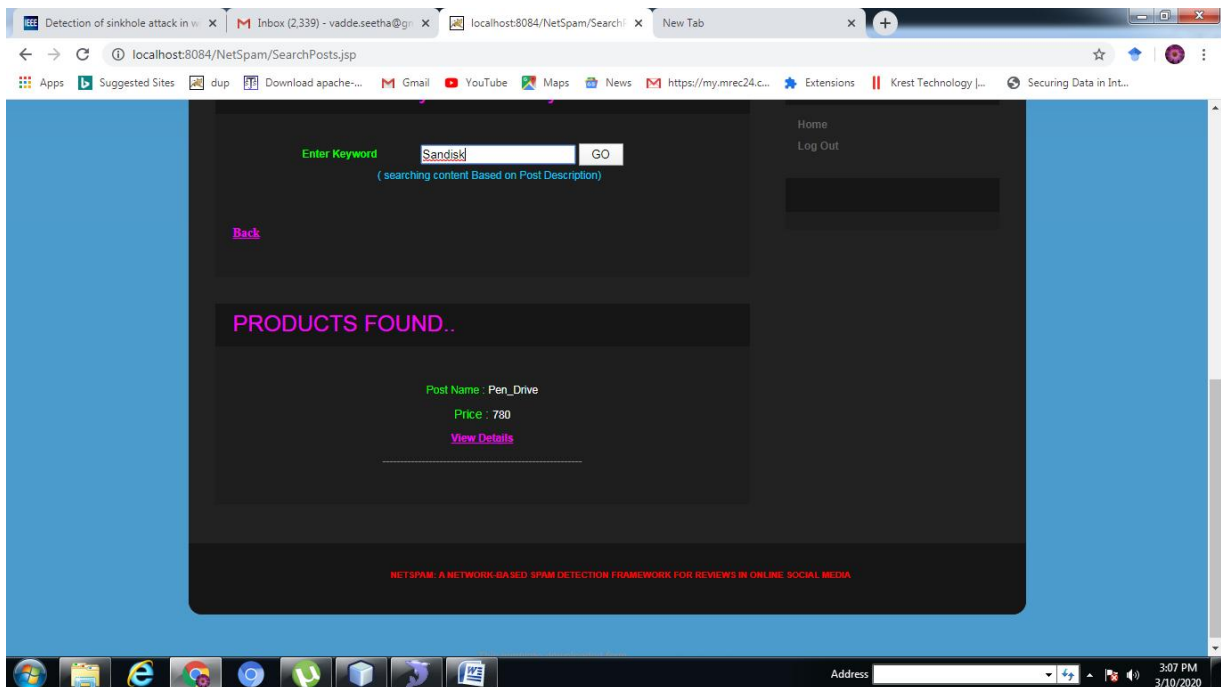
User Friend requests



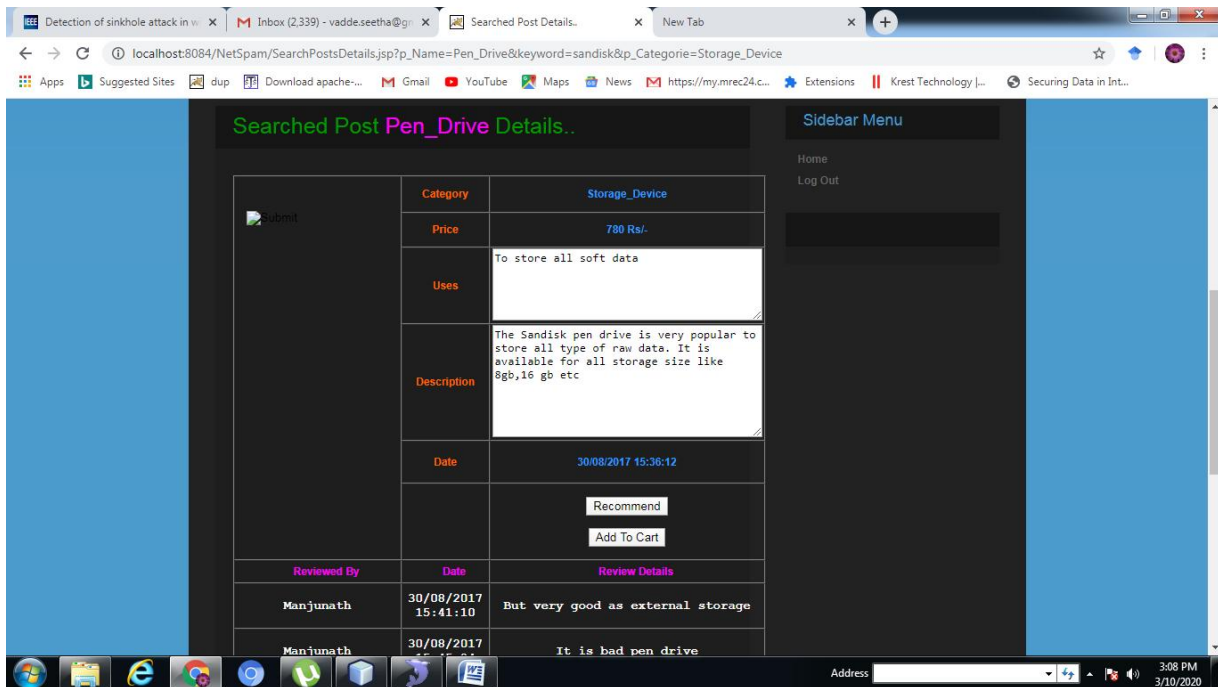
View My Friends



User Search posts



View post details



Searched Post Pen_Drive Details..

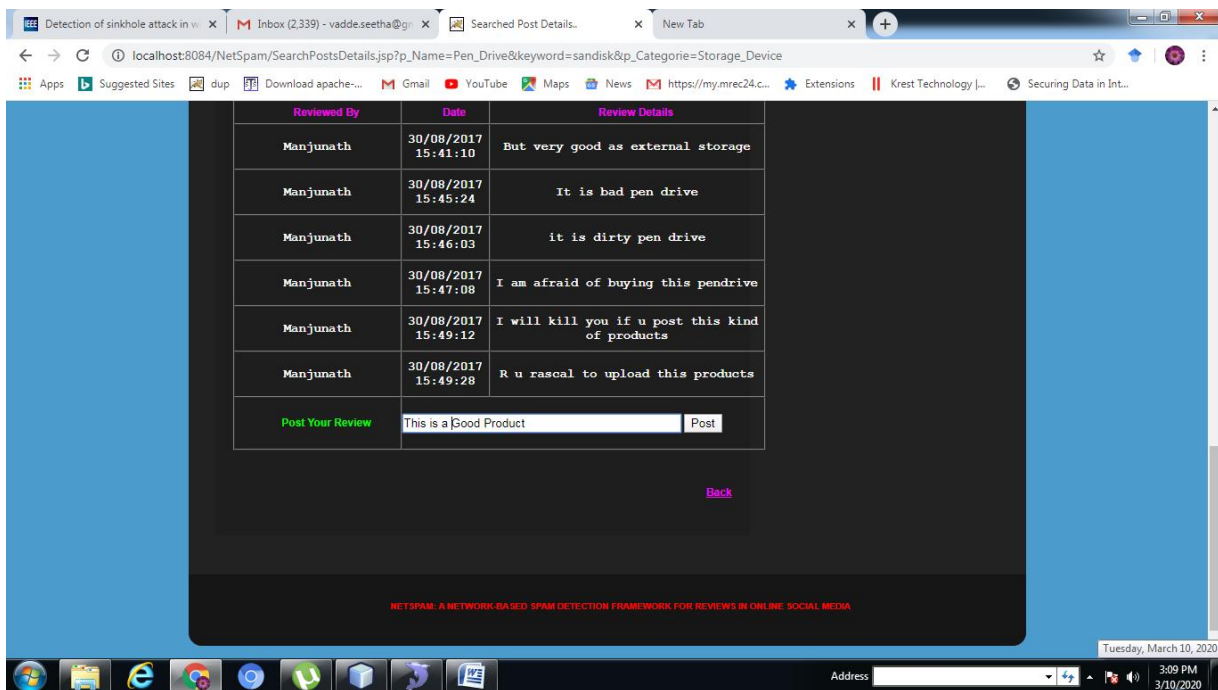
	Category	Storage_Device
	Price	780 Rs/-
	Uses	To store all soft data
	Description	The Sandisk pen drive is very popular to store all type of raw data. It is available for all storage size like 8gb,16 gb etc
	Date	30/08/2017 15:36:12
		Recommend
		Add To Cart
Reviewed By	Date	Review Details
Manjunath	30/08/2017 15:41:10	But very good as external storage
Manjunath	30/08/2017 15:45:24	It is bad pen drive
Manjunath	30/08/2017 15:46:03	it is dirty pen drive
Manjunath	30/08/2017 15:47:08	I am afraid of buying this pendrive
Manjunath	30/08/2017 15:49:12	I will kill you if u post this kind of products
Manjunath	30/08/2017 15:49:28	R u rascal to upload this products

[Post Your Review](#) This is a Good Product [Post](#)

[Back](#)

NETSPAM: A NETWORK-BASED SPAM DETECTION FRAMEWORK FOR REVIEWS IN ONLINE SOCIAL MEDIA

User Positive Review on Product



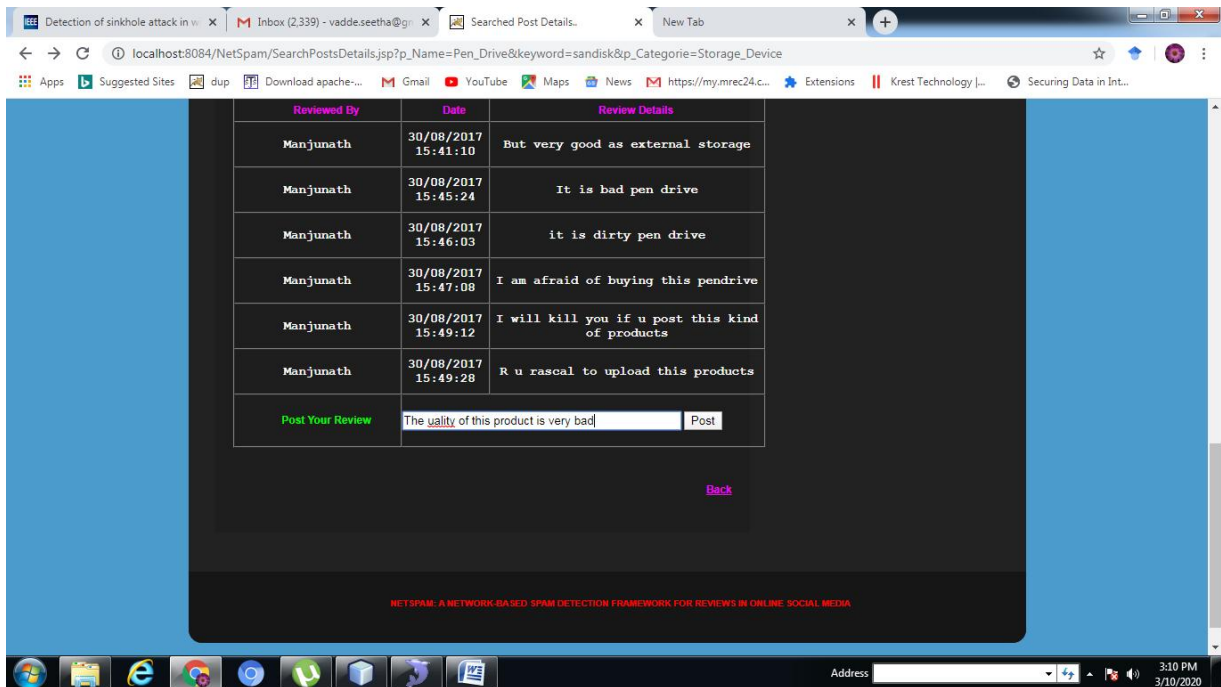
Reviewed By	Date	Review Details
Manjunath	30/08/2017 15:41:10	But very good as external storage
Manjunath	30/08/2017 15:45:24	It is bad pen drive
Manjunath	30/08/2017 15:46:03	it is dirty pen drive
Manjunath	30/08/2017 15:47:08	I am afraid of buying this pendrive
Manjunath	30/08/2017 15:49:12	I will kill you if u post this kind of products
Manjunath	30/08/2017 15:49:28	R u rascal to upload this products

[Post Your Review](#) This is a Good Product [Post](#)

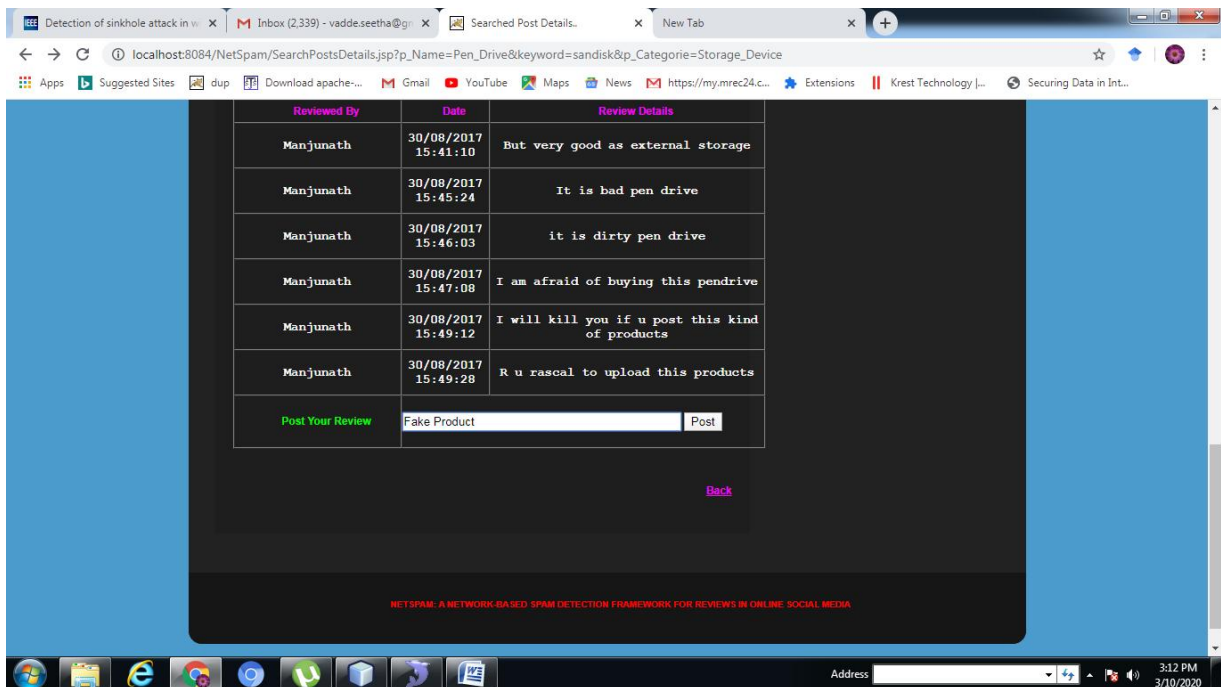
[Back](#)

NETSPAM: A NETWORK-BASED SPAM DETECTION FRAMEWORK FOR REVIEWS IN ONLINE SOCIAL MEDIA

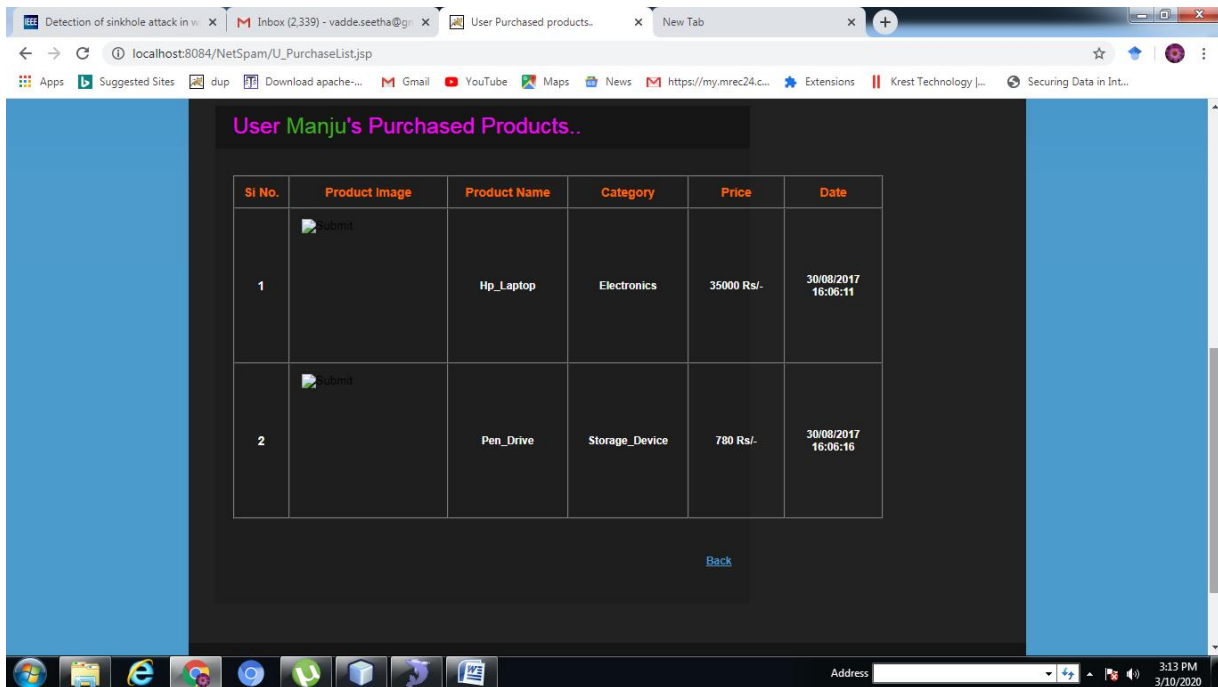
User Negative Review on product





User Spam review on given product



User Purchased Products:

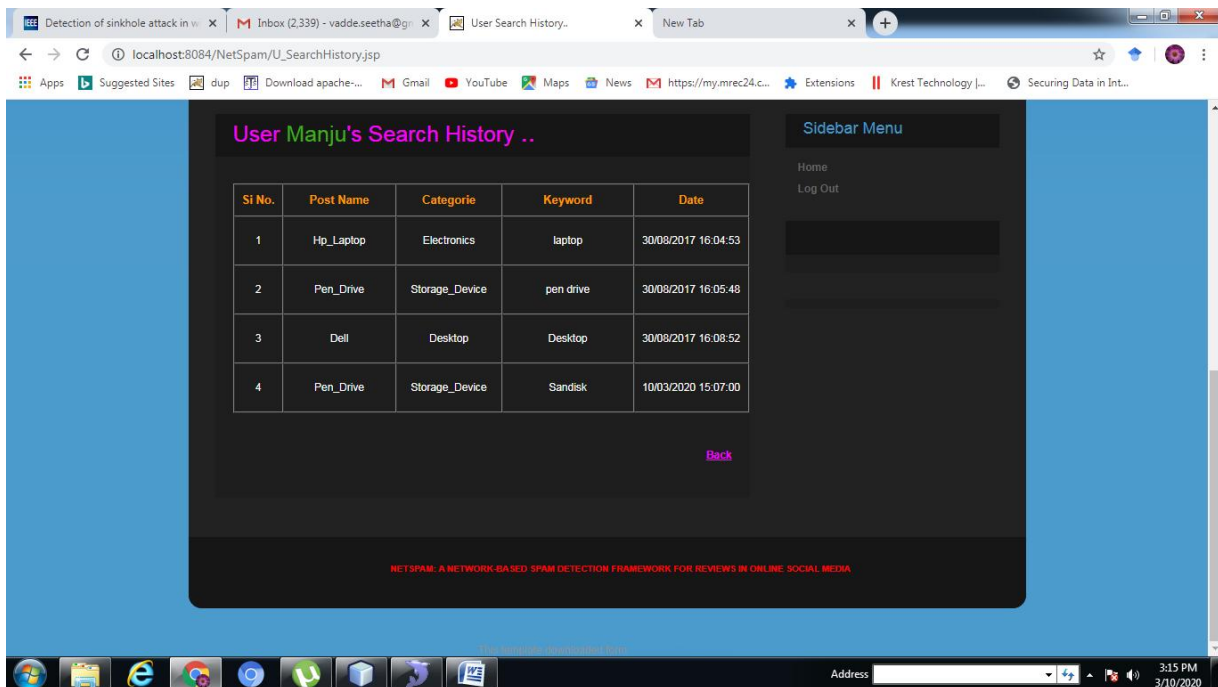


User Manju's Purchased Products..

Si No.	Product Image	Product Name	Category	Price	Date
1		Hp_Laptop	Electronics	35000 Rs/-	30/08/2017 16:06:11
2		Pen_Drive	Storage_Device	780 Rs/-	30/08/2017 16:06:16

[Back](#)

My search history



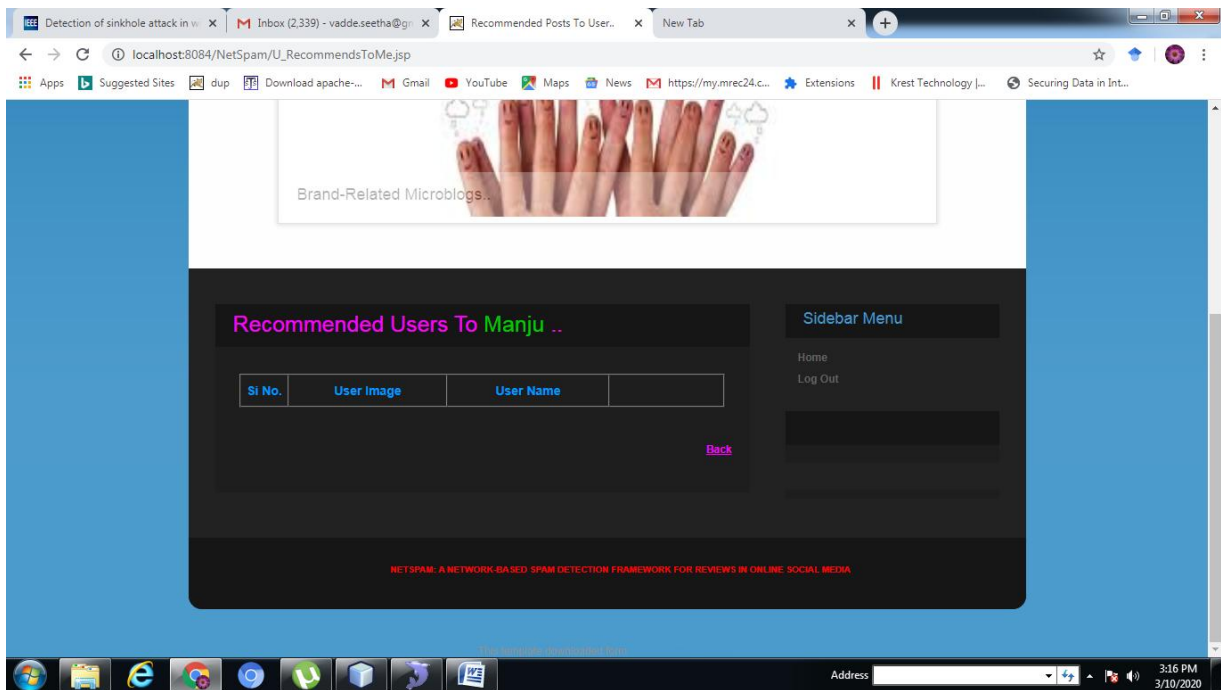
User Manju's Search History ..

Si No.	Post Name	Categorise	Keyword	Date
1	Hp_Laptop	Electronics	laptop	30/08/2017 16:04:53
2	Pen_Drive	Storage_Device	pen drive	30/08/2017 16:05:48
3	Dell	Desktop	Desktop	30/08/2017 16:08:52
4	Pen_Drive	Storage_Device	Sandisk	10/03/2020 15:07:00

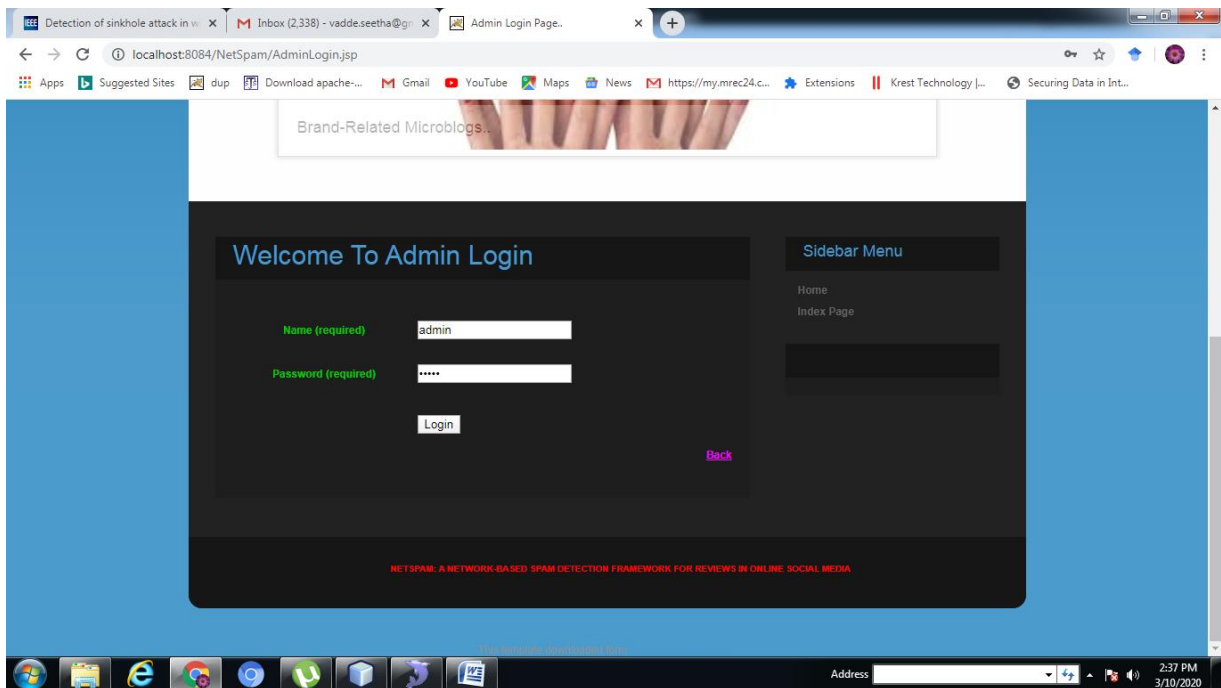
[Back](#)

NETSPAM: A NETWORK-BASED SPAM DETECTION FRAMEWORK FOR REVIEWS IN ONLINE SOCIAL MEDIA

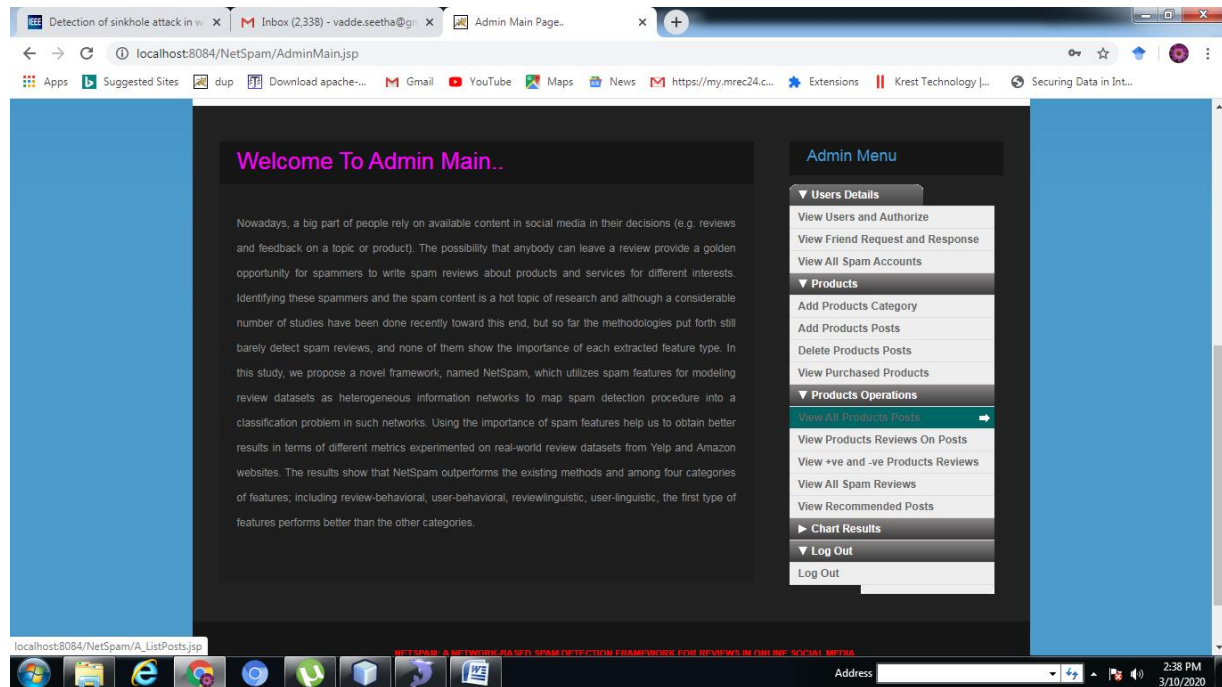
Post recommends



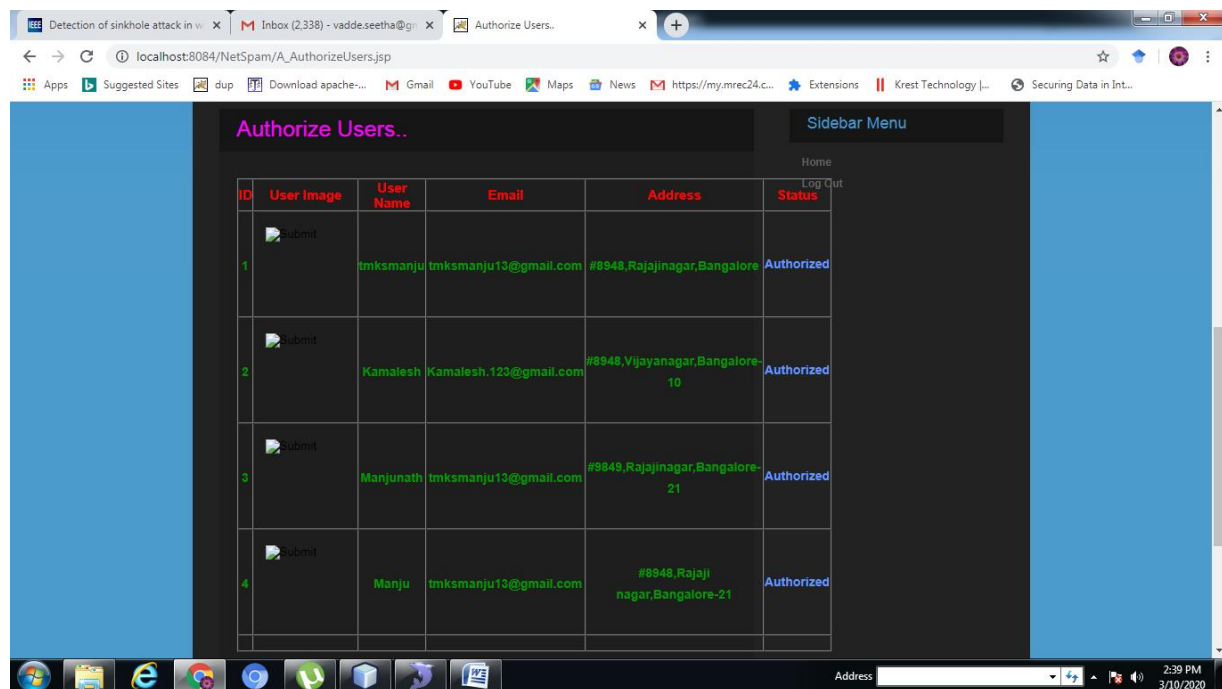
Admin login page



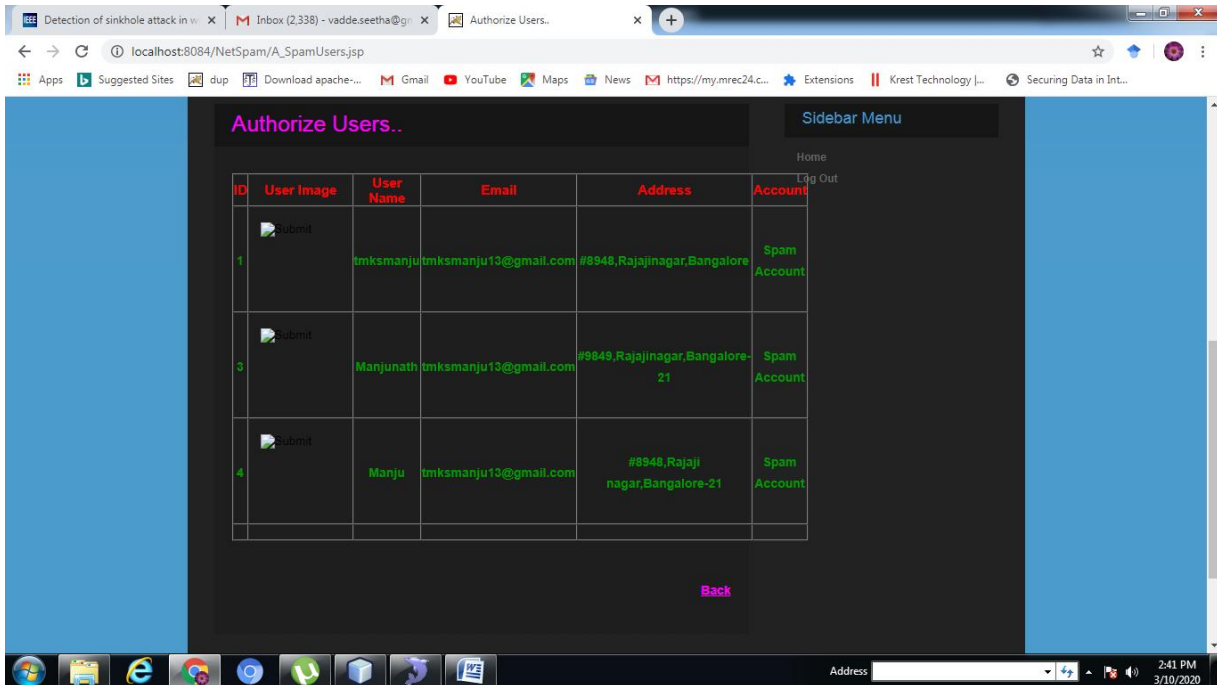
Admin Operation






Users & Authorize



View All spam Users

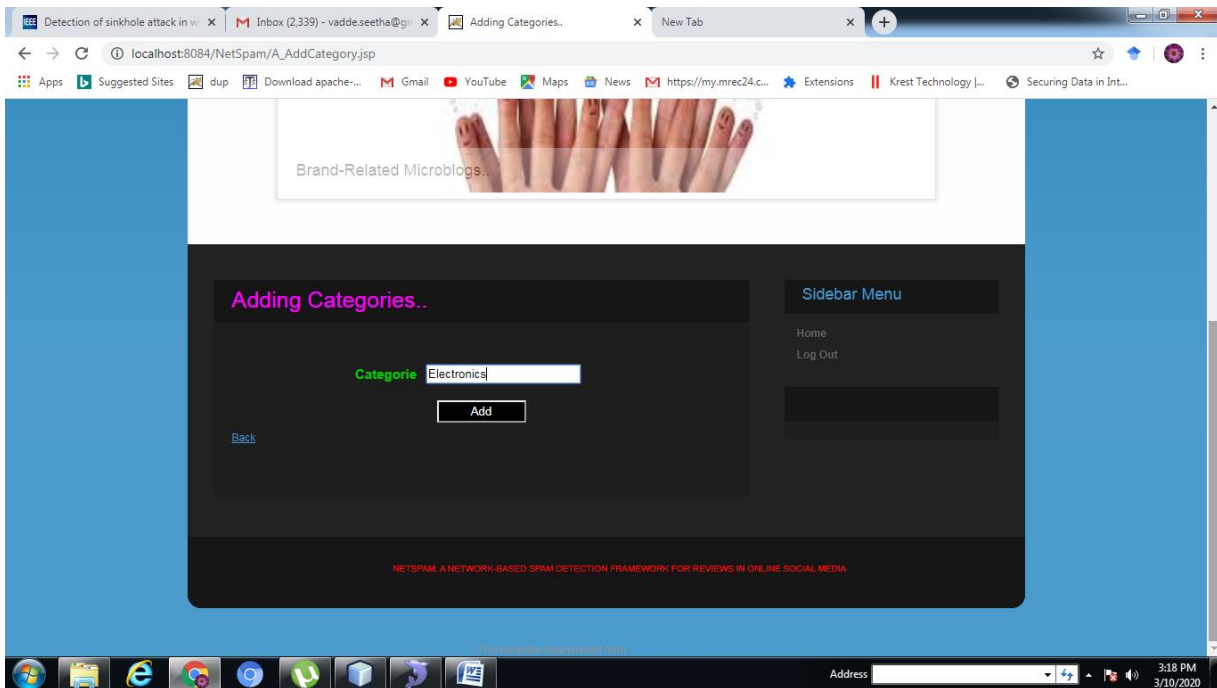


The screenshot shows a web browser window with the URL `localhost:8084/NetSpam/A_SpamUsers.jsp`. The page title is "Authorize Users..". It features a sidebar menu with "Home" and "Log Out" options. The main content area displays a table of spam users with the following data:

ID	User Image	User Name	Email	Address	Account
1		tmksmanju	tmksmanju13@gmail.com	#8948,Rajajinagar,Bangalore	Spam Account
3		Manjunath	tmksmanju13@gmail.com	#9649,Rajajinagar,Bangalore-21	Spam Account
4		Manju	tmksmanju13@gmail.com	#8948,Rajajinagar,Bangalore-21	Spam Account

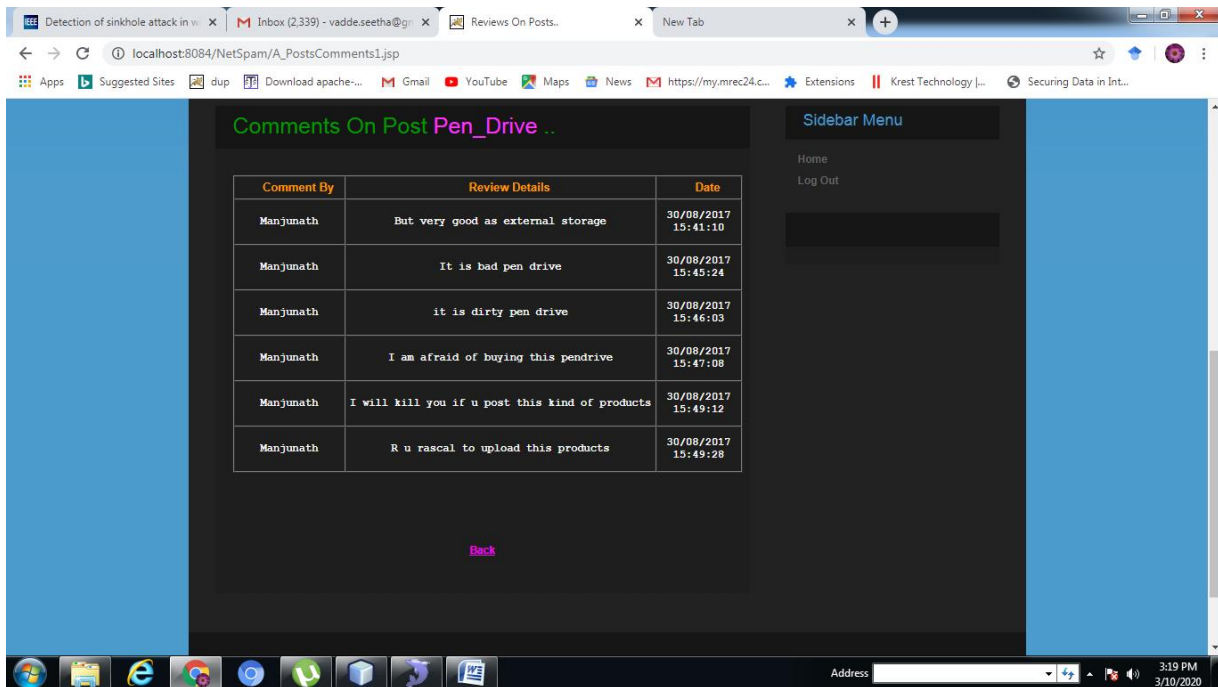
Below the table is a "Back" button. The browser's taskbar at the bottom shows the date and time as 2:41 PM on 3/10/2020.

Add Product category

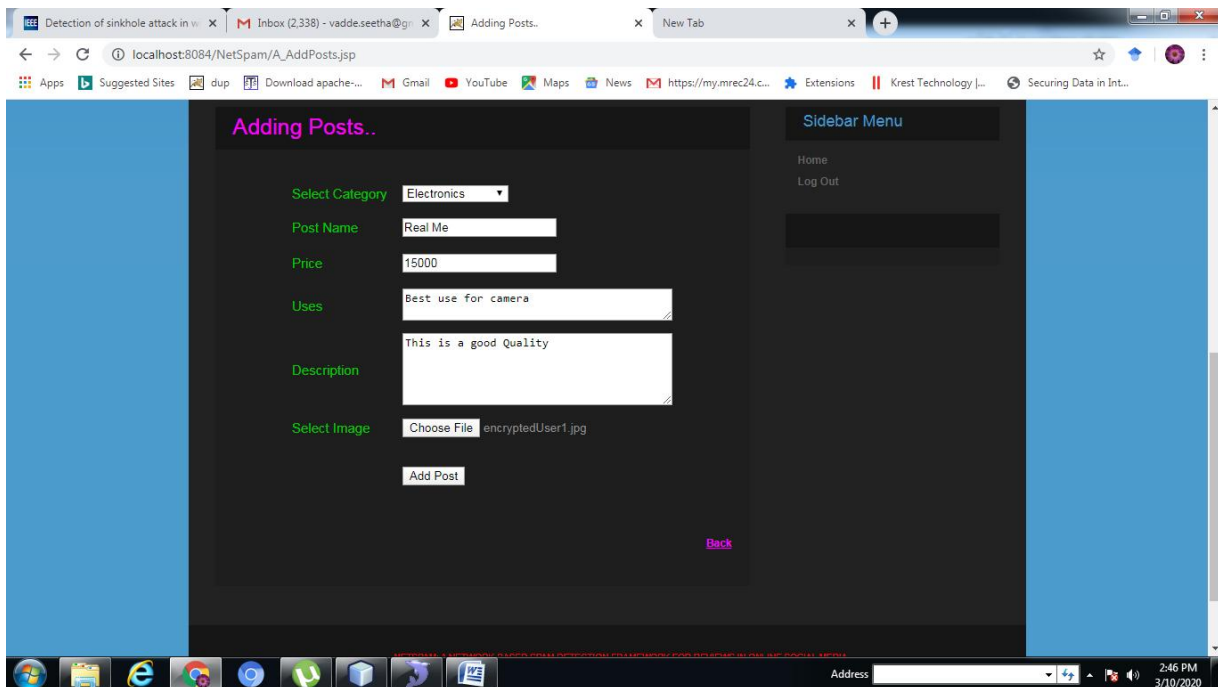


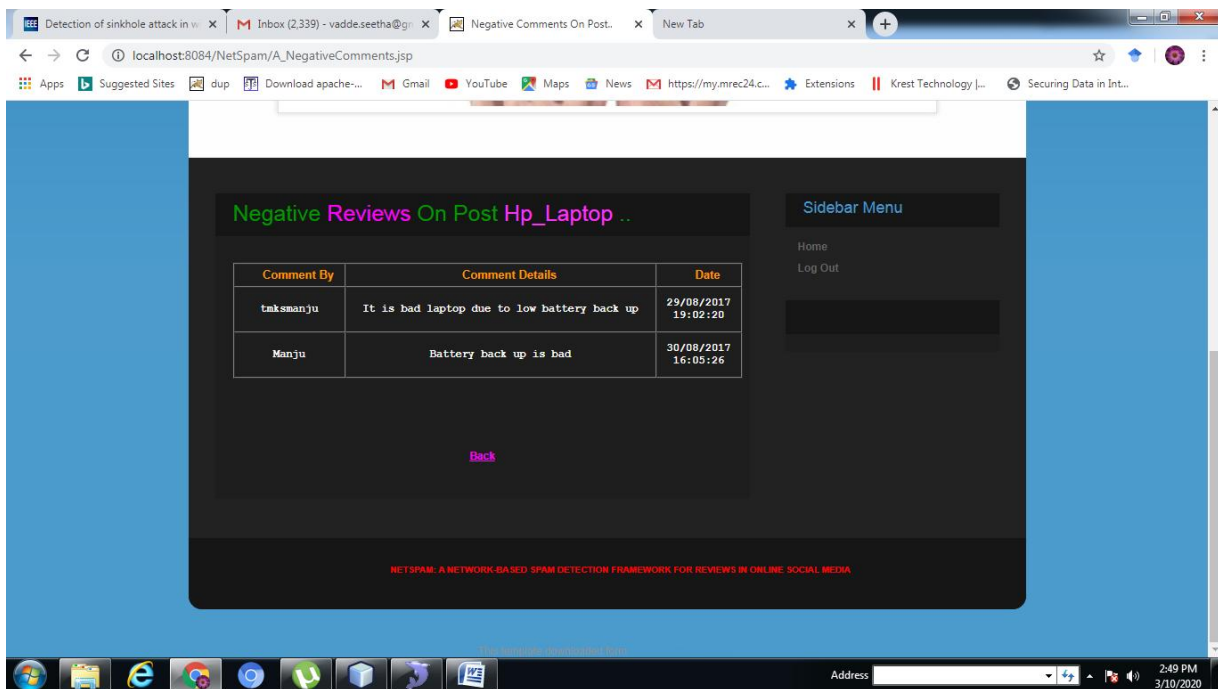
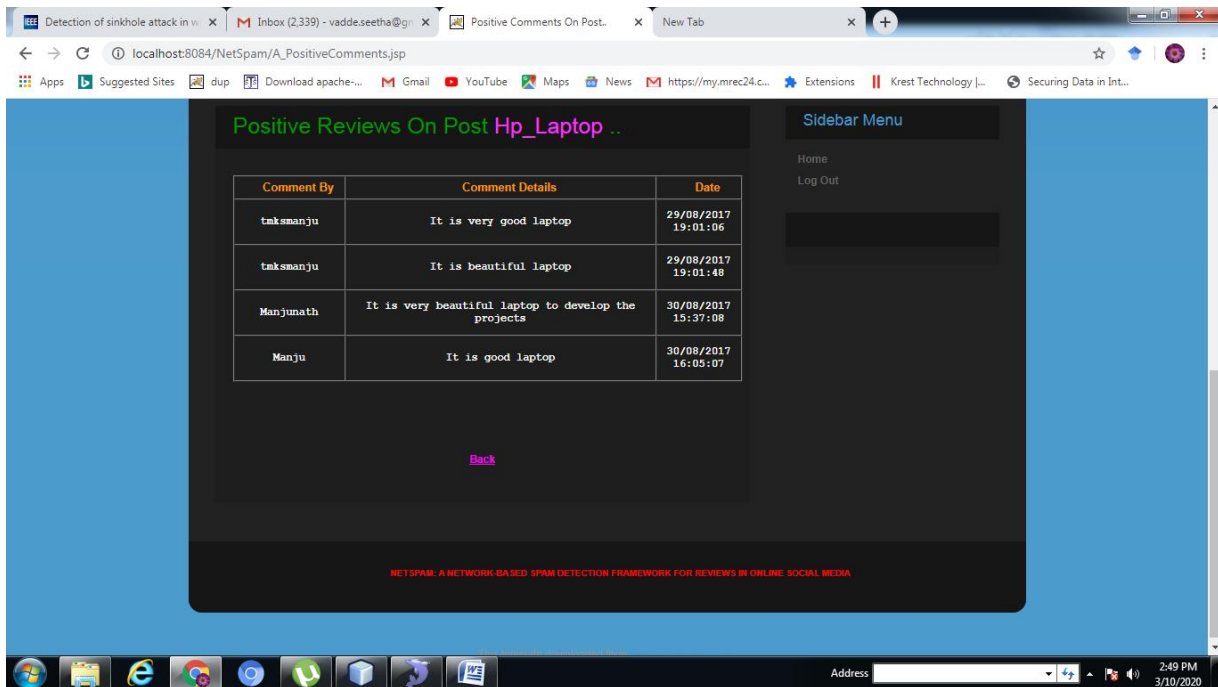
The screenshot shows a web browser window with the URL `localhost:8084/NetSpam/A_AddCategory.jsp`. The page title is "Adding Categories..". It features a sidebar menu with "Home" and "Log Out" options. The main content area displays a form with a "Category" input field containing the text "Electronics" and an "Add" button. Below the form is a "Back" button. At the top of the page, there is a banner image of hands and the text "Brand-Related Microblogs..". At the bottom of the page, there is a footer that reads "NETSPAM: A NETWORK-BASED SPAM DETECTION FRAMEWORK FOR REVIEWS IN ONLINE SOCIAL MEDIA". The browser's taskbar at the bottom shows the date and time as 3:18 PM on 3/10/2020.

Admin Can view Comments For each products



Adding posts





CHAPTER-5

CONCLUSION

CHAPTER-5

CONCLUSION

5.1. CONCLUSION

This study introduces a novel spam detection framework namely NetSpam based on a metapath concept as well as a new graph-based method to label reviews relying on a rank-based labeling approach. The performance of the proposed framework is evaluated by using two real-world labeled datasets of Yelp and Amazon websites. Our observations show that calculated weights by using this metapath concept can be very effective in identifying spam reviews and leads to a better performance. In addition, we found that even without a train set, NetSpam can calculate the importance of each feature and it yields better performance in the features' addition process, and performs better than previous works, with only a small number of features. Moreover, after defining four main categories for features our observations show that the reviews behavioral category performs better than other categories, in terms of AP, AUC as well as in the calculated weights. The results also confirm that using different supervisions, similar to the semi-supervised method, have no noticeable effect on determining most of the weighted features, just as in different datasets.

5.2. FUTURE ENHANCEMENT:

For future work, metapath concept can be applied to other problems in this field. For example, similar framework can be used to find spammer communities. For finding community, reviews can be connected through group spammer features and reviews with highest similarity based on metapath concept are known as communities. In addition, utilizing the product features is an interesting future work on this study as we used features more related to spotting spammers and spam reviews. Moreover, while single networks has received considerable attention from various disciplines for over a decade, information diffusion and content sharing in multilayer networks is still a young research. Addressing the problem of spam detection in such networks can be considered as a new research line in this field.

REFERENCES

REFERENCES

- [1] J. Donfro, A whopping 20 % of yelp reviews are fake. <http://www.businessinsider.com/20-percent-of-yelp-reviews-fake-2013-9>. Accessed: 2015-07-30.
- [2] M. Ott, C. Cardie, and J. T. Hancock. Estimating the prevalence of deception in online review communities. In ACM WWW, 2012.
- [3] M. Ott, Y. Choi, C. Cardie, and J. T. Hancock. Finding deceptive opinion spam by any stretch of the imagination. In ACL, 2011.
- [4] Ch. Xu and J. Zhang. Combating product review spam campaigns via multiple heterogeneous pairwise features. In SIAM International Conference on Data Mining, 2014.
- [5] N. Jindal and B. Liu. Opinion spam and analysis. In WSDM, 2008.
- [6] F. Li, M. Huang, Y. Yang, and X. Zhu. Learning to identify review spam. Proceedings of the 22nd International Joint Conference on Artificial Intelligence; IJCAI, 2011.
- [7] G. Fei, A. Mukherjee, B. Liu, M. Hsu, M. Castellanos, and R. Ghosh. Exploiting burstiness in reviews for review spammer detection. In ICWSM, 2013.
- [8] A. j. Minnich, N. Chavoshi, A. Mueen, S. Luan, and M. Faloutsos. Trueview: Harnessing the power of multiple review sites. In ACM WWW, 2015.
- [9] B. Viswanath, M. Ahmad Bashir, M. Crovella, S. Guah, K. P. Gummadi, B. Krishnamurthy, and A. Mislove. Towards detecting anomalous user behavior in online social networks. In USENIX, 2014.
- [10] H. Li, Z. Chen, B. Liu, X. Wei, and J. Shao. Spotting fake reviews via collective PU learning. In ICDM, 2014.
- [11] L. Akoglu, R. Chandy, and C. Faloutsos. Opinion fraud detection in online reviews by network effects. In ICWSM, 2013.
- [12] R. Shebuti and L. Akoglu. Collective opinion spam detection: bridging review networks and metadata. In ACM KDD, 2015.
- [13] S. Feng, R. Banerjee and Y. Choi. Syntactic stylometry for deception detection. Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers; ACL, 2012.
- [14] N. Jindal, B. Liu, and E.-P. Lim. Finding unusual review patterns using unexpected rules. In ACM CIKM, 2012.

- [15] E.-P. Lim, V.-A. Nguyen, N. Jindal, B. Liu, and H. W. Lauw. Detecting product review spammers using rating behaviors. In ACM CIKM, 2010.
- [16] A. Mukherjee, A. Kumar, B. Liu, J. Wang, M. Hsu, M. Castellanos, and R. Ghosh. Spotting opinion spammers using behavioral footprints. In ACM KDD, 2013.
- [17] S. Xie, G. Wang, S. Lin, and P. S. Yu. Review spam detection via temporal pattern discovery. In ACM KDD, 2012.
- [18] G. Wang, S. Xie, B. Liu, and P. S. Yu. Review graph based online store review spammer detection. IEEE ICDM, 2011.
- [19] Y. Sun and J. Han. Mining Heterogeneous Information Networks; Principles and Methodologies, In ICCCE, 2012.
- [20] A. Mukerjee, V. Venkataraman, B. Liu, and N. Glance. What Yelp Fake Review Filter Might Be Doing?, In ICWSM, 2013.
- [21] S. Feng, L. Xing, A. Gogar, and Y. Choi. Distributional footprints of deceptive product reviews. In ICWSM, 2012.
- [22] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. In VLDB, 2011.
- [23] Y. Sun and J. Han. Rankclus: integrating clustering with ranking for heterogeneous information network analysis. In Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology, 2009.
- [24] C. Luo, R. Guan, Z. Wang, and C. Lin. HetPathMine: A Novel Transductive Classification Algorithm on Heterogeneous Information Networks. In ECIR, 2014

GITHUB link: