A

Project Report

On

DETECTION OF FAKE ONLINE REVIEWS USING SEMI-SUPERVISED AND SUPERVISED LEARNING

Submitted in partial fulfillment of the Requirements

For the award of the Degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINNERING

Submitted by

T.RAMA KRISHNA REDDY	**(19BG1A0553)**
B.NAGENDRA REDDY	**(19BG1A0512)**
S.NITHIN REDDY	**(19BG1A0546)**
P.NARASIMHA REDDY	**(19BG1A0543)**
B.KALYAN YADAV	**(19BG1A0507)**

Under The Guidance Of

Dr.M.M PRASADA REDDY M.Tech,p.h.D

Associate Professor & HOD, DEPT of CSE.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SRI VENKATESWARA INSTITUTE OF SCIENCE AND TECHNOLOGY

TADIGOTALA(v), YSR KADAPA DISTRICT, AP-516003

Jawaharlal Nehru Technological University, Ananthapuramu, AP.

(2019-2023)

SRI VENKATESWARA INSTITUTE OF SCIENCE AND TECHNOLOGY

BONAFIDE CERTIFICATE

Certified that this project report "DETECTION OF FAKE ONLINE REVIEWS USING SEMI-SUPERVISED AND SUPERVISED" is the bonafide work done by " T.RAMAKRISHNA REDDY (19BG1A0553),B. NAGENDRA REDDY (19BG1A0512), S.NITHIN REDDY (19BG1A0546), P.NARASIMHA REDDY (19BG1A0543) ,B.KALYAN YADAV (19BG1A0507)" who carried out the project under my guidance duringyear 2022-2023, towards partial fulfillment of the requirements of the degree of Bachelor of Technology in computer Science & Engineering from Jawaharlal Nehru Technical University, Ananthapuramu. The results embodied in this reporthave not been submitted to any other University for the award of any Degree.

PROJECT GUIDE	HEAD OF THE DEPARTMENT
Dr M. MARUTHY PRASAD REDDY	
	Dr M. MARUTHY PRASAD REDDY
Professor & HOD	**HEAD OF THE DEPARTMENT**
Department of CSE.	**Department of CSE.**

Submitted for viva voice examination held on _____

Internal Examiner **External Examiner**

ACKNOWLEDGMENTS

At every outset I express my gratitude to almighty lord for showering his grace and blessings upon me to complete this project.

Although our name appears on the cover of this book, many people had contributed in some form or the other form to this project development. We cloud not done this project without the assistance or support of the following we thank you all.

We wish to place on my record my deep sense of gratitude to my project guide,

Dr M. MARUTHY PRASAD REDDY **M. Tech,ph.D** for his constant motivation and valuable helpthrough the project work.

Express our gratitude to **Dr M. MARUTHY PRASAD REDDY** **Ph.D. Head of the Department**, Sri Venkateswara Institute of Science and Technology for his valuable suggestions and advices throughout the course.

Express our thankfulness to **Dr. G. V. SURESH BABU** **Ph.D**. Principle of Sri Venkateswara Institute of Science and Technology for his valuable suggestions and advices throughout the course.

Express our thanksgiving to **Er. R. SAI SHARAN REDDY**, **Vice President** of Sri Venkateswara Institute of Science and Technology.

Finally, we Would like to thanks my friends for their cooperation to complete this project.

PLACE: KADAPA

DATE:

T.RAMAKRISHNA REDDY (19BG1A0553)

B. NAGENDRA REDDY (19BG1A0512)

S. NITHIN REDDY (19BG1A0546)

P.NARASIMHA REDDY (19BG1A0543)

B.KALYAN YADAV (19BG1A0507)

ABSTRACT

Online reviews have great impact on today's business and commerce. Decision making for purchase of online products mostly depends on reviews given by the users. Hence, opportunistic individuals or groups try to manipulate product reviews for their own interests. This paper introduces some semi-supervised and supervised text mining models to detect fake online reviews as well as compares the efficiency of both techniques on dataset containing hotel reviews.

TABLE OF CONTENTS

1. INTRODUCTION

Technologies are changing rapidly. Old technologies are continuously being replaced by new and sophisticated ones. These new technologies are enabling people to have their work done efficiently. Such an evolution of technology is online marketplace. We can shop and make reservation using online websites. Almost, every one of us checks out reviews before purchasing some products or services. Hence, online reviews have become a great source of reputation for the companies. Also, they have large impact on advertisement and promotion of products and services. With the spread of online marketplace, fake online reviews are becoming great matter of concern. People can make false reviews for promotion of their own products that harms the actual users. Also, competitive companies can try to damage each other's reputation by providing fake negative reviews.

Researchers have been studying about many approaches for detection of these fake online reviews. Some approaches are review content based and some are based on behavior of the user who is posting reviews. Content based study focuses on what is written on the review that is the text of the review where user behavior based method focuses on country, ip-address, number of posts of the reviewer etc. Most of the proposed approaches are supervised classification models. Few researchers, also have worked with semi-supervised models. Semi-supervised methods are being introduced for lack of reliable labeling of the reviews.

In this paper, we make some classification approaches for detecting fake online reviews, some of which are semi supervised semi supervised and others are supervised. For semi-supervised learning, we use Expectation-maximization algorithm. Statistical Naive Bayes classifier and Support Vector Machines (SVM) are used as classifiers in our research work to improve the performance of classification. We have mainly focused on the content of

the review based approaches. As feature we have used word frequency count, sentiment polarity and length of review.

2. LITERATURE SURVEY

[1] J. K. Rout, A. Dalmia, and K.-K. R. Choo, "Revisiting semi-supervised learning for online deceptive review detection," IEEE Access, Vol. 5, pp. 1319–1327, 2017.

With more consumers using online opinion reviews to inform their service decision making, opinion reviews have an economical impact on the bottom line of businesses. Unsurprisingly, opportunistic individuals or groups have attempted to abuse or manipulate online opinion reviews (e.g., spam reviews) to make profits and so on, and that detecting deceptive and fake opinion reviews is a topic of ongoing research interest. In this paper, we explain how semi-supervised learning methods can be used to detect spam reviews, prior to demonstrating its utility using a data set of hotel reviews.

[2] A. Heydari, M. A. Tavakoli, N. Salim, and Z. Heydari, "Detection of Review spam: a survey", Expert Systems with Applications, vol. 42, no. 7, pp. 3634–3642, 2015.

Major Society of people using internet trust the contents of internet. The possibility that anyone can take survey about anything provides a great opportunity to spammers to make fake reviews about product, its services. So to identify the spammers and their spams is really a debatable issue for research and despite that there are many studies in this context yet none of them has a great significance. In this application, we use a structure, stated as Net Spam, which proposes spam features for giving a practical hotel review datasets to design a spam review detection method into a classification issue. Utilizing the role of spam features helps us to have good outcome in context of different metrics on review datasets. The results shows that Net Spam outcome with the previous methods and encompassed by features of the four categories; involving behavior and language of review feature, behavior and language of user feature, better outcomes can be obtained from first

type of feature rather than other ones. The contribution work is when user will search query it will display all top products as well as there is recommendation of the product

[3] Chengai Sun, Qiaolin Du and Gang Tian, "Exploiting Product Related Review Features for Fake Review Detection," Mathematical Problems in Engineering, 2016

Authenticity and reliability of the information spread over the cyberspace is becoming increasingly important. This is especially important in e-commerce since potential customers check reviews and customer feedbacks online before making a purchasing decision. Although this information is easily accessible through related websites, lack of verification of the authenticity of these reviews raises concerns about their reliability. Besides, fraudulent users disseminate misinformation to deceive people into acting against their interest. So, detection of fake and unreliable reviews is a crucial problem that must be addressed by the security researchers. Here we propose a spam review detection framework that incorporates knowledge extracted from the textual content of the reviews with information obtained by exploiting the underlying reviewer-product network structure. In the proposed framework, first, feature vectors are learned for each review, reviewer and product by utilizing state-of-the-art algorithms developed for learning document and node embeddings, and then these are fed into a classifier to identify opinion spam. The effectiveness of our framework over existing techniques on detecting spam reviews is demonstrated in three different data sets containing online reviews. The experimental results obtained confirm that combining representations learned from reviewer-product network and textual review data significantly improves the detection of spam reviews.

[4] J. W. Pennebaker, M. E. Francis, and R. J. Booth, "Linguistic Inquiry and Word Count: Liwc," vol. 71, 2001.

Social media has become a prominent platform for opinions and thoughts. This stated that the characteristics of a person can be assessed through social media status updates. The purpose of this research article is to provide a web application in order to detect one's

personality using linguistic feature analysis. The personality of a person has classified according to Eysenck's Three Factor personality model. The proposed technique is based on ontology based text classification, linguistic feature-vector matrix using LIWC (Linguistic Inquiry and Word Count) features including semantic analysis using supervised machine learning algorithms and questionnaire based personality detection. This is vital for HR management system when recruiting and promoting employees, R&D Psychologists can use the dynamic ontology for storage purposes and all the other API users including universities and sports clubs. According to the test results the proposed system is in an accuracy level of 91%, when tested with a real world personality detection

questionnaire based application, and results demonstrate that the proposed technique can detect the personality of a person with considerable accuracy and a speed.

[5] E. P. Lim, V.-A. Nguyen, N. Jindal, B. Liu, and H. W. Lauw, "Detecting product review spammers using rating behaviors," in Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM), 2010.

Nowadays, many people depending on online reviews for the purchasing decision of a product/ service. One of the characteristics of an online review system is that anyone can post a review that allows spammers to compose fake reviews. Recently, these spammers work as groups to intensify their activities and for maximum profit gains. Few works are concentrated on the detection of group spammers compared to individual review/reviewer spamming. This work proposes a framework to detect spammer groups using graph- based algorithms with five group spamming features and also proposes a new group spamming feature, Group Rating Similarity (GRS) based on the review rating score. The results show that the proposed framework performs well with five features when comparing with the existing work having seven features. Also, the proposed feature (GRS) shows better performance in discriminating spam and non-spam when experimented on realworldreview datasets from the Yelp website.

3. SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

- Content based methods focus on what is the content of the review. That is the text of the review or what is told in it. Heydari et al. [2] have attempted to detect spam review by analyzing the linguistic features of the review. Ott et al. [3] used three techniques to perform classification. These three techniques are- genre identification, detection of psycholinguistic deception and text categorization.
- Behavior feature based study focuses on the reviewer that includes characteristics of the person who is giving the review. Lim et al. [7] addressed the problem of review spammer detection, or finding users who are the source of spam reviews. People who post intentional fake reviews have significantly different behavior than the normal user. They have identified the following deceptive rating and review behaviors.
- Deceptive online review detection is generally considered as a classification problem and one popular approach is to use supervised text classification techniques [5]. These techniques are robust if the training is performed using large datasets of labeled instances from both classes, deceptive opinions (positive instances) and truthful opinions (negative examples) [8]. Some researchers also used semi-supervised classification techniques.

Disadvantages:

- In the existing work, the system uses only to semi-supervised learning.
- Only Text Classification as sentiment text and it never finds fake review

3.2 PROPOSED SYSTEM

- In the proposed system, each review goes through tokenization process first. Then, unnecessary words are removed and candidate feature words are generated.

- Each candidate feature words are checked against the dictionary and if its entry is available in the dictionary then its frequency is counted and added to the column in the feature vector that corresponds the numeric map of the word.
  Alongside with counting frequency, the length of the review is measured and added to the feature vector.
- Finally, sentiment score which is available in the data set is added in the feature vector. We have assigned negative sentiment as zero valued and positive sentiment as some positive valued in the feature vector.

Advantages:

- The system is very fast and effective due to semi-supervised and supervised learning.
- Focused on the content of the review based approaches. As feature we have used word frequency count, sentiment polarity and length of review.

3.3 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure

that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

ECONOMICAL FEASIBILIT

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.

3.4 MODULES OF THE PROJECT

- **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 Movies,View Uploaded Movies,View Positive Sentiment Reviews,View Negative Sentiment Reviews,View Neutral Sentiment Reviews,View Ratings Results,View Dislikes

  Results,View Likes Results,View Remote Users,View Moview Reviews,View Trending Movies,View Moviews recommended,View Fake Reviews,View Fake ratings

  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 all movies and make reviews, arating,like and dislike,View all movie reviews,View Trending movies, View Your profile,View all movies recommended

4. HARDWARE AND SOFTWARE ARE REQUIREMENT SPECIFICATION

- **H/W System Configuration**

- Processor - Pentium –IV
- RAM - 4 GB (min)
- Hard Disk - 500 GB

Software Requirements:

- Operating System - Windows XP
- Coding Language - Java/J2EE(JSP,Servlet)
- Front End - HTML/CSS/JAVASCRIPT
- Web Server - Apache Tomcat7.x
- Back End - MySQL 5.5.4.1

5. SYSTEM DESIGN

5.1 INTRODUCTION

A graphical tool used to describe and analyze the moment of data through a system manual or automated including the process, stores of data, and delays in the system. Data Flow Diagrams are the central tool and the basis from which other components are developed. The transformation of data from input to output, through processes, may be described logically and independently of the physical components associated with the system. The DFD is also known as a data flow graph or a bubble chart. DFDs are the model of the proposed system. They clearly should show the requirements on which the new system should be built. Later during design activity this is taken as the basis for drawing the system's structure charts. The Basic Notation used to create a DFD's are as follows:

1. **Dataflwo:** Data move in a specific direction from an origin to a destination.

2. **Process:** People, procedures, or devices that use or produce (Transform) Data. The physical component is not identified.

3. **Source**: External sources or destination of data, which may be People, programs, organizations or other entities.

Systems design is the process or art of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. One could see it as the application of systems theory to product development. There is some overlap and synergy with the disciplines of systems analysis, systems architecture and systems engineering. Database Designing is a part of the

development process. In the linear development cycle, it is used during the system requirements phase to construct the data

symbols from a modeling method known as entity relationship analysis. components of the analysis model. This model represents the major data objects and the relationship between them. It should not be confused with data analysis, which takes place in the system design phase. As in a DFD, a model of data consists of a number of symbols joined up according to certain conventions. System designers describe these conceptual modeling using

Entity Relationship Diagram

Entity relationship analysis uses three major abstractions to describe data. These are

1. Entities, which are distinct things in the enterprise.
2. Relationships, which are meaningful interactions between the objects.
3. Attributes, which are the properties of the entities and relationship.

The relative simplicity and pictorial clarity of this diagramming technique may well account in large part for the widespread use of ER model. Such a diagram consists of the following major components.

E-R Diagram Components

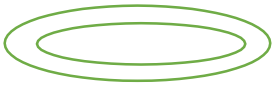
Rectangles, which represent the entity set.

Ellipse, which represent attributes.

Diamonds, which represent relationship sets.

Lines,

which link attributes to entity sets and entity sets to relationships.



Double Ellipse, which represents multi valued attributes.

Double lines, which indicates total participation of an entity in a relationship set.

Entity

• An entity is an object that exists and is distinguishable from other objects.

• An entity may be concrete or abstract.

• An entity is a set of entities of the same type.

• Entity sets need not be disjoint.

• An entity is represented by a set of attributes.

Mapping Constraints

An E-R diagram may define certain constraints which the contents of a database must conform.

Mapping Cardinalities

It expresses the number of entities to which another entity can be associated via a relationship. For binary relationship sets between entity sets A and B, the mapping cardinality must be one of the following:

One-to-One – An entity in A is associated with at most one entity in B, and an entity in B is associated with at most one entity in A.

One-to-many -An entity in A is associated with any number in B. An entity in B is associated with any number in A.

Many-to-many – Entities in A and B are associated with any number from each other.

Cardinality: It indicates that which type relationship the business rule follows is called cardinality.

Connectivity: It specifies that which type of notation the entities are connected in both sides that one side or many side.

5.2 DATA DICTIONARY

The logical characteristics of current systems data stores, including name, description, aliases, contents, and organization, identifies processes where the data are used and where immediate access to information required, Serves as the basis for identifying database requirements during system design.

Uses of Data Dictionary

• To manage the details in large systems.

• To communicate a common meaning for all system elements.

• To Document the features of the system.

- To facilitate analysis of the details in order to evaluate characteristics and determine where system changes should be made.
- To locate errors and omissions in the system.

5.3 UML DIAGRAMS

It is a language to specifying, visualizing and constructing the artifacts of softwaresystem as well as for business models. UML was originally motivated by the desire to standardize the disparate notational system and approaches to software design developed by Grady Booch, Ivar Jacobson and James Rumbaugh at Rational Software in 1994-95. The UML notation is useful for graphically depicting Object

Oriented Analysis and Object Oriented Design modules. The unified modeling language is a standard language for specifying, Visualizing, Constructing and documenting the software system and its components. It is a graphical language that provides a vocabulary and set of semantics andrules. The UML focuses on the conceptual and physical representation of the system. It captures the decisions and understandings about systems that must be constructed. It is used to understand, design, configure, maintain and control information about the systems.

An Overview of UML

The Unified Modeling Language is a language for

- Visualizing.
- Specifying.
- Constructing.
- Documenting.

UML Models

User model view

- This view represents the system from the user's perspective.
- The analysis representation describes a usage scenario from the end-usersperspective.

Structural model view

- In this model the data and functionality are arrived from inside the system.
- This model view models the static structures.

Behavioral model view

It represents the dynamic of Behavioral as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

Implementation model view

In this the structural and behavioral as parts of the system are represented as they are to be built.

Environmental model view

In this the structural and Behavioral aspects of the environment in which the system is to be implemented are represented. UML is specifically constructed through two different domains they are

- UML Analysis modeling, this focuses on the user model and structural modelviews of the system.
- UML design modeling, which focuses on the Behavioral modeling,implementation modeling and environmental model views.

A Conceptual model of UML

- The three major elements of UML are
- The UML's basic building blocks.
- The rules that dictate how those building blocks may be put together.
- Some common mechanisms that apply throughout the UML.

Basic building blocks of the UML

The vocabulary of UML encompasses three kinds of building blocks

- Things.
- Relationships.
- Diagrams.

Things

Things are the abstractions that are first-class citizens in a model. Relationships tie these things together. Diagrams group the interesting collection of things. There arefour kinds of things in the UML

Structural things

Behavioral things

Grouping thing

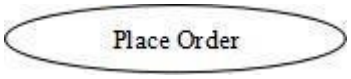
Annotational things

Structural Things

Structural things are the nouns of the UML models. These are mostly static parts ofthe model, representing elements that are either conceptual or physical. In all, there are seven kinds of Structural things.

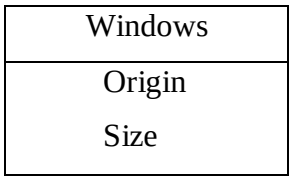
Use Case

Use case is a description of a set of sequence of actions that a system performs thatyields an observable result of value to a particular things in a model. Graphically, Use Caseis rendered as an ellipse with dashed lines, usually including only its name as shown below.



class

A class is a description of set of object that share the same attribute,operation,relationships, and semantics. A class implements one or more interfaces. Graphically a class is rendered as a rectangle, usually including its name, attributes and operations, as shown below.



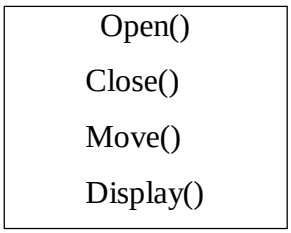


Fig -5.3.2:Sample Class Diagram

Interface

An interface is a collection of operations that specify a service of a class or component. An interface describes the externally visible behaviour of element. Graphicallythe interface is rendered as a circle together with its name.

Fig -5.3.3 : Interface

Collaboration

Collaboration defines an interaction and is a society of roles and other elements thatwork together to provide some cooperative behaviour that's bigger than the sum of all theelements. Graphically, collaboration is rendered as an ellipse with dashed lines, usually including only its name as shown below.

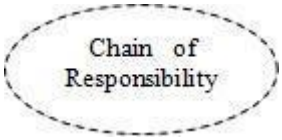


Fig-5.3.4 : Sample Collaboration Diagram

Component

Component is a physical and replaceable part of a system that conforms to and provides the realization of a set of interfaces. Graphically, a component is rendered as a rectangle with tabs, usually including only its name, as shown below.

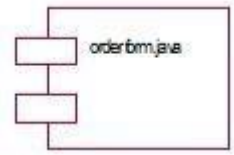


Fig-5.3.5 : Sample Component Diagram

Node

A Node is a physical element that exists at run time and represents a computational

resource, generally having at least some memory and often, processing capability. Graphically, a node is rendered as a cube, usually including only its name, as shown below.

Fig-5.3.6 : Sample Node Diagram

Behavioral Things

Behavioral things are the dynamic parts of UML models. These are the verbs of amodel, representing behavior over time and space.

Interaction

An interaction is a behavior that comprises a set of messages exchanged among aset of objects within a particular context to accomplish a specific purpose.

State Machine Display

Fig-5.3.7 : Sample Interaction Diagram

A state machine is a behavior that specifies the sequence of states an object or an interaction goes through during its lifetime on response to events, together with its responses to those events. Graphically, a state is rendered as rounded rectangle usually including its name and its sub-states, if any, as shown below.

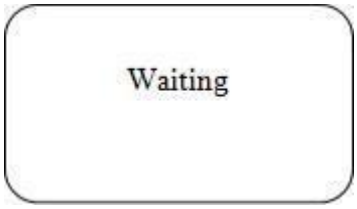


Fig-5.3.8 : Sample State Machine

3. Grouping Things

Grouping things are the organizational parts of the UML models. These are theboxes into which a model can be decomposed.

Package

A package is a general-purpose mechanism for organizing elements into groups.

Fig-5.3.9 : Sample Package Diagram

Annotational Things

Notes

Annotat ory parts of the UML models.

ional

things

are the A note is simply a symbol for rendering constraints and comments attached to an

explanat

element or a collection of elements. Graphically a note is rendered as a rectangle with dog-eared corner together, with a textual or graphical comment, as shown below.

Fig-5.3.10 : Sample Note

DiagramRelationships in the UML

There are four kinds of relationships in the UML

- Dependency.
- Association.
- Generalization.
- Realization.

These relationships are the basic relational building blocks of the UML. You use them to write well-formed models. Graphically dependency is rendered as a dashed line, possibly directed, and occasionally including a label.

---------->

Fig-5.3.11 : Dependency

Second, an association is a structural relationship that describes a set of links, a link beinga connection among objects. Aggregation is a special kind of association, representing a structural relationship between a whole and its parts. Graphically, an association isrendered as a solid line, possibly directed, occasionally including a label, and often containing other adornments, such as multiplicity and role names.

Employer ——————— Employee

Fig-5.3.12 : Association

Third, a generalization is a specialization/generalization relationship in which objects of the specialized element are substitutable for objects of the generalized element (the parent).In this way, the child shares the structure and the behavior of the parent. Graphically a generalization relationship is rendered as a solid line with a hollow arrowhead pointing tothe parent.

Fig-5.3.13 : Generalization

Fourth, a realization is a semantic relationship between classifiers, wherein one classifier specifies a contract that another classifier guarantees to carry out. You'll encounter

realization relationships in two places between interfaces and the classes or components that realize them and between use cases and the collaborations that realize them.

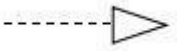


Fig-5.3.14 : Realization

Each UML diagram is designed to let developers and customers view a software system from a different perspective and in varying degrees of abstraction. Use Case Diagram displays the relationship among actors and use cases.

Class Diagram models class structure and contents using design elements such as classes,packages and objects. It also displays relationships such as containment, inheritance, associations and others.

Interaction Diagrams

Sequence Diagram displays the time sequence of the objects participating in the interaction. This consists of the vertical dimension (time) and horizontal dimension (different objects).

- Collaboration Diagram displays an interaction organized around the objects andtheir links to one another. Numbers are used to show the sequence of messages.
- State Diagram displays the sequences of states that an object of an interaction goesthrough during its life in response to received stimuli, together with its responses and actions.

Activity Diagram displays a special state diagram where most of the states are action states and most of the transitions are triggered by completion of the actions in the source states. This diagram focuses on flows driven by internal processing.

Physical Diagrams

- Component Diagram displays the high level packaged structure of the code itself. Dependencies among components are shown, including source code components, binary code components, and executable components. Some components exist at compile time, at link time, at run times well as at more than one time.
- Deployment Diagram displays the configuration of run-time processing elements and the software components, processes, and objects that live on them. Software component instances represent run-time manifestations of code.

Diagrams in the UML

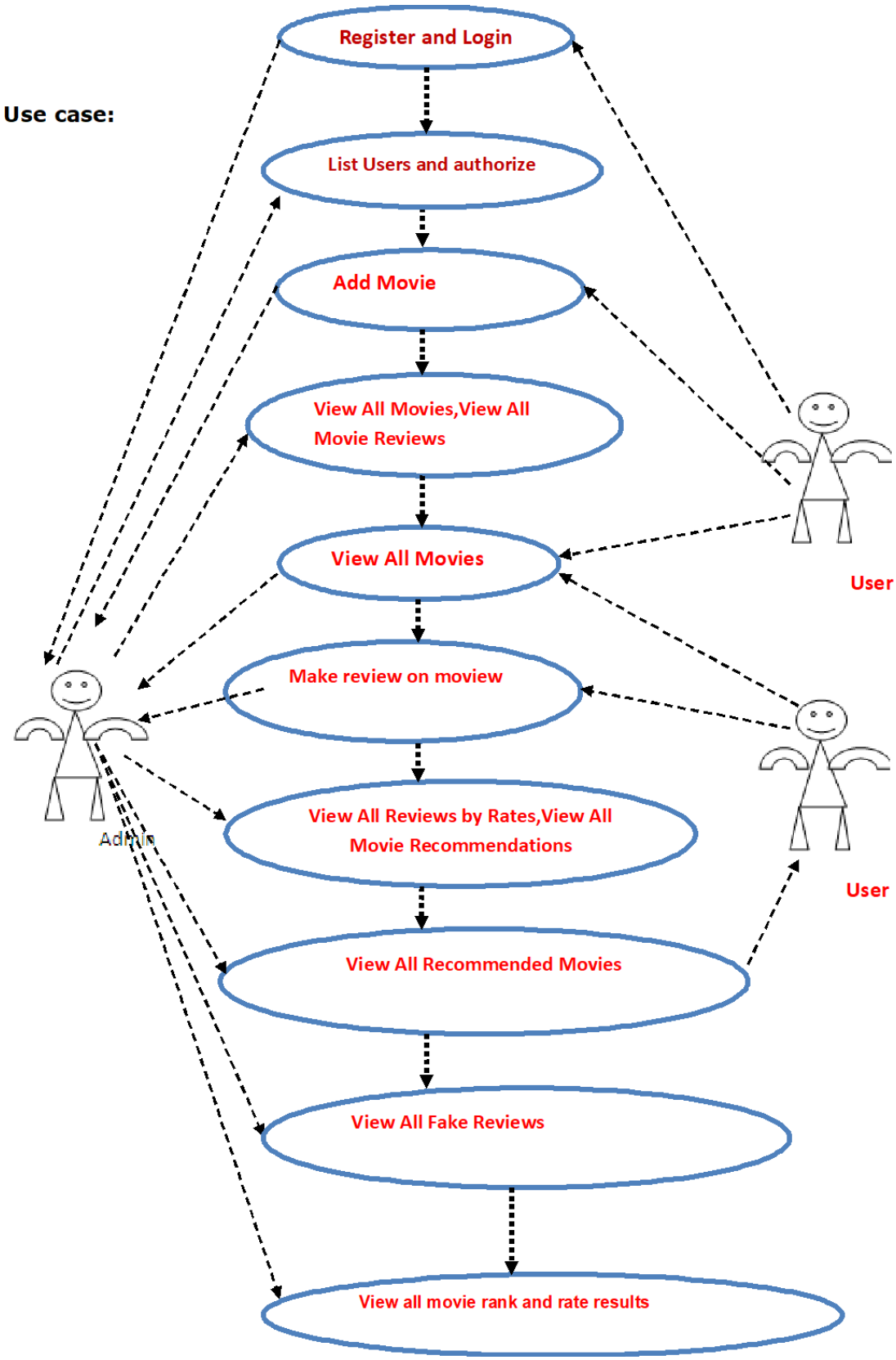
5.3.1 Use Case Diagram

Use case diagram graphically depict system behavior. These diagrams present a high level view of how the system is used as viewed from an outsider's (actor's) perspective. A use-case diagram may depict all or some of the use cases of a system.

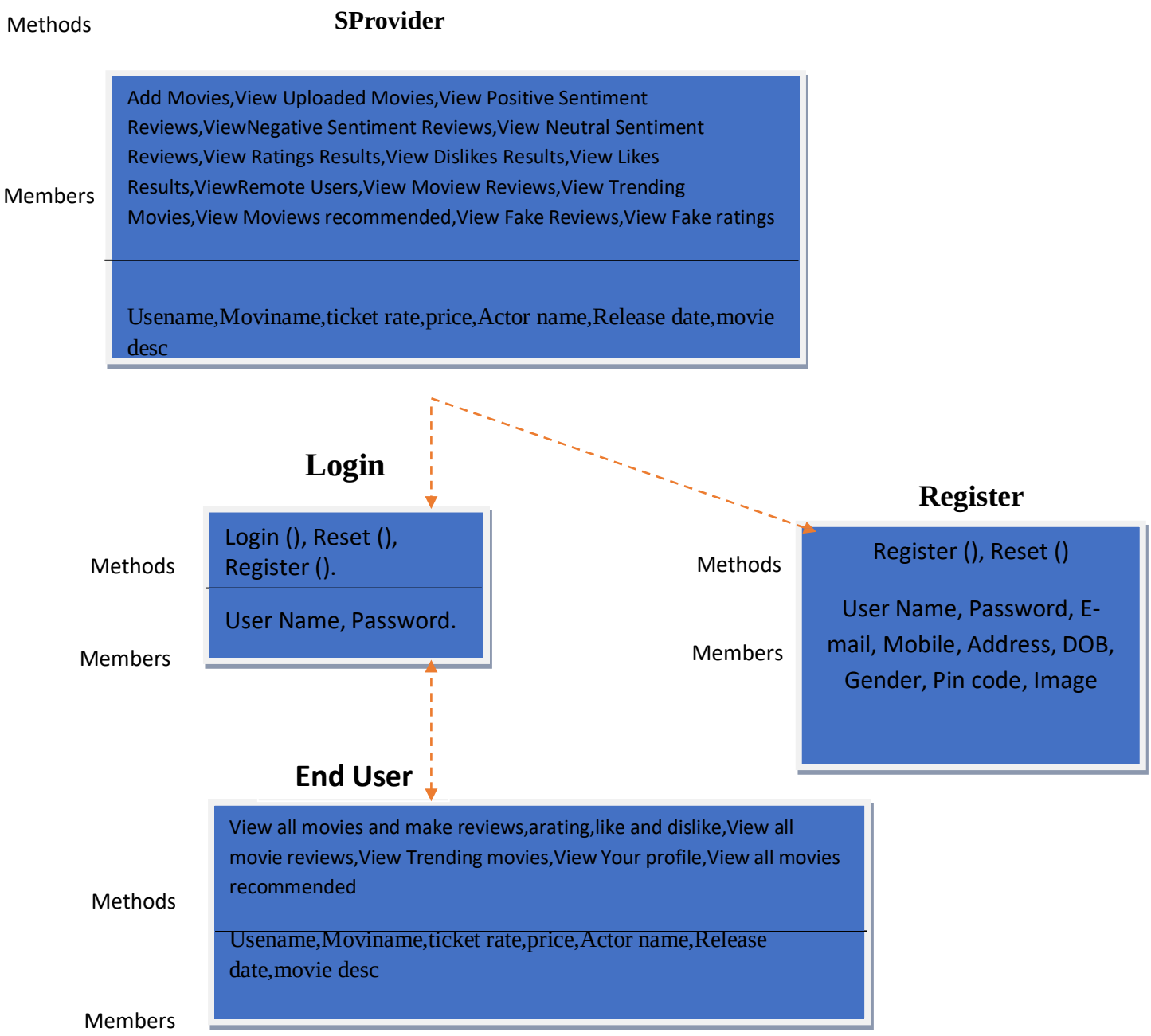
A use-case diagram can contain:

- Actors
- Use cases

Interaction or relationship between actor and use cases in the system including the associations, dependencies, and generalizations. Use-case diagram can be used during analysis to capture the system requirements and to understand how the system should work.During the design phase, you can use use-case diagrams to specify the behavior of the systems implemented.

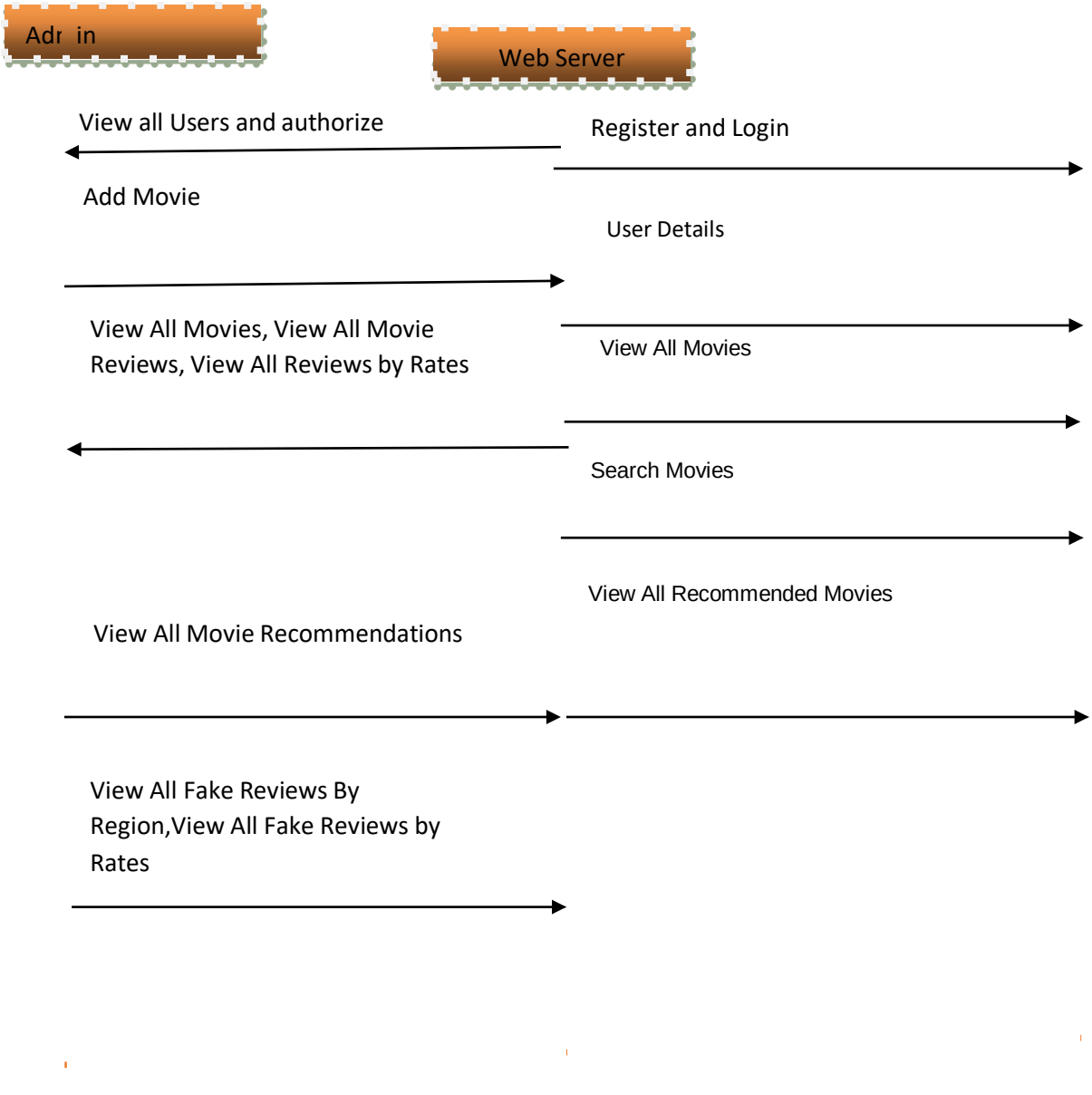

Use case:
Register and Login
List Users and authorize
Add Movie
View All Movies,View All Movie Reviews
View All Movies
Make review on moview
View All Reviews by Rates,View All Movie Recommendations
View All Recommended Movies
View All Fake Reviews
View all movie rank and rate results
Admin
User
User

5.3.2. Class Diagram



5.3.2 Sequence Diagram

A sequence diagram in Unified Modeling Language (UML) is a kind of interactiondiagram 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.



6. IMPLEMENTATION AND RESULTS

6.1 TECHNOLOGY USED

Java technology is both a programming language and a platform.

The Java Programming Language

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

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

With most programming languages, you either compile or interpret a program so that you can run it on your computer. The Java programming language is unusual in that aprogram is both compiled and interpreted. With the compiler, first you translate a programinto 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.

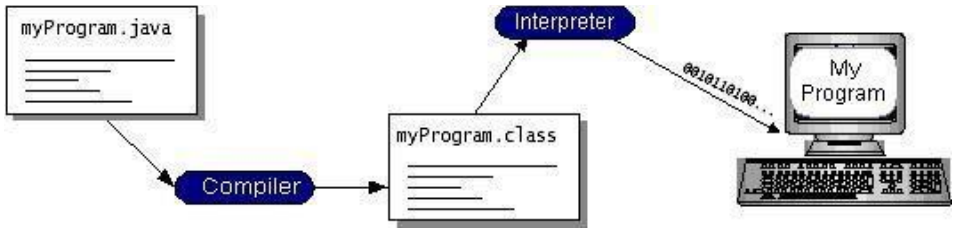
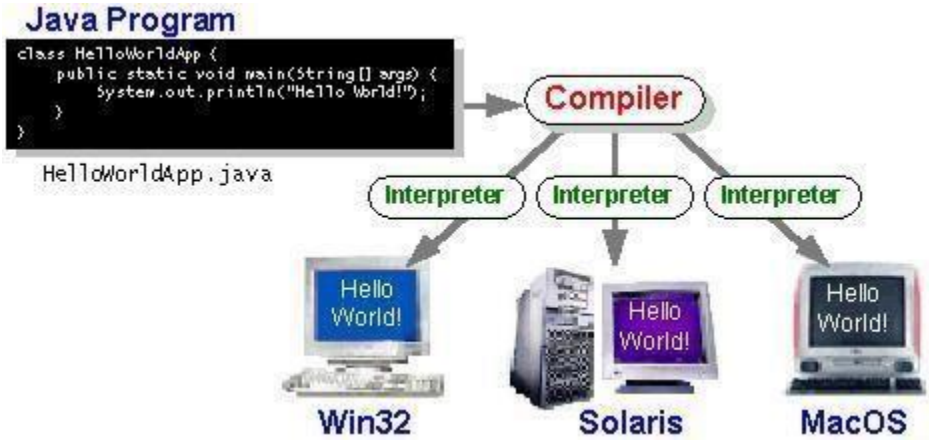


Fig- 6.1.1 : Java Compiler Architecture

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 bytecodes 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.

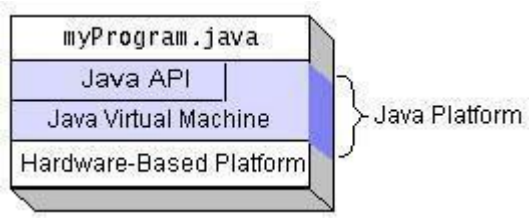
The Java platform has two components:

- 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 platformand is ported onto various hardware-based platforms.

The Java API is a large collection of ready-made software components that providemany 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 functionalitysome of the packages in the Java API provide.

The following figure depicts a program that's running on the Java platform. As thefigure 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 bitslower 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 runwithin a Java-enabled browser.

However, the Java programming language is not just for writing cute, entertainingapplets for the Web. The general-purpose, high-level Java programming

language is also apowerful software platform. Using the generous API, you can write many types ofprograms.

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 webapplications, 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 ofsoftware 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, datastructures, system properties, date and time, and so on.
- **Applets**: The set of conventions used by applets.
- **Networking**: URLs, TCP (Transmission Control Protocol), UDP (User Data gram Protocol) sockets, and IP (Internet Protocol) addresses.
- **Internationalization**: Help for writing programs that can be localized for users worldwide. Programs can automatically adapt to specific locales andbe displayed in the appropriate language.
- **Security**: Both low level and high level, including electronic signatures,public and private key management, access control, and certificates.
- **Software components**: Known as JavaBeans^{TM}, can plug into existingcomponent architectures.

- **Object serialization**: Allows lightweight persistence and communicationvia Remote Method Invocation (RMI).
- **Java Database Connectivity (JDBC^{TM})**: Provides uniform access to awide range of relational databases.

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

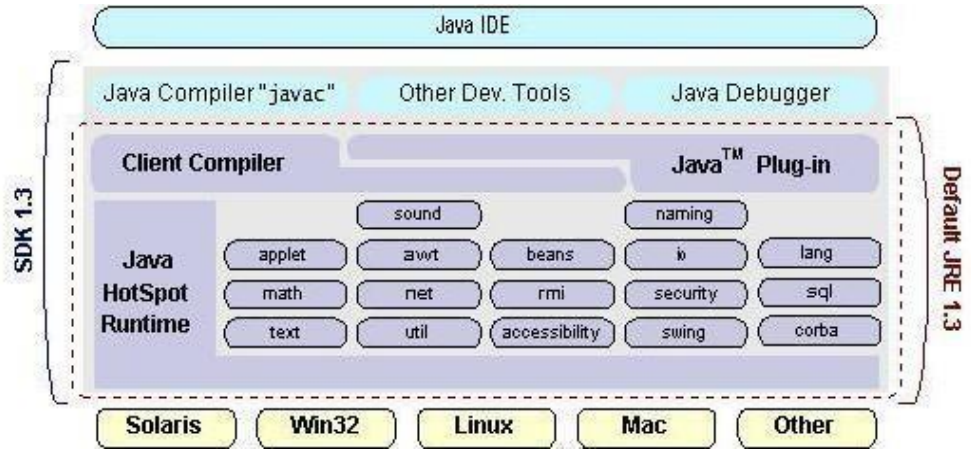


Fig- 6.1.3 :

JRE

JDBC

In an effort to set an independent database standard API for Java; Sun Microsystemsdeveloped 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. Asyou 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 knowwhat it is about and how to use it effectively. This is by no means a complete overview ofJDBC. 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 frameworkfor building database applications in Java.

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

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.

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 addressingand class D uses all 32.

Subnet address

Internally, the UNIX network is divided into sub networks. Building 11 is currentlyon 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 256machines 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. Tosend 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 WriteFile 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 overa network create a socket each. These are similar to two ends of a pipe - but the actual pipedoes 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 setincludes:

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-sideapplications;

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 proprietaryapplications.

1. Map

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 projectinclude:

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, andintegrating this with the existing XYPlot class in JFreeChart;

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

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 themain chart.

1.Dashboards

There is currently a lot of interest in dashboard displays. Create a flexible dashboardmechanism 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.

2.Property Editors

The property editor mechanism in JFreeChart only handles a small subset of the properties that can be set for charts. Extend (or reemployment) this mechanism to providegreater 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 theJavaOne 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.

6.3 DAT

ABASE

SQL

Level

API

The designers felt that their main goal was to define a SQL interface for Java. Althoughnot the lowest database interface level possible, it is at a low enough level for higher-leveltools 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.

1. **SQL Conformance**

SQL syntax varies as you move from database vendor to database vendor. In an effortto 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 handlenon-standard functionality in a manner that is suitable for its users.

1. 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 allowsJDBC to use existing ODBC level drivers by the use of a software interface. This interfacewould translate JDBC calls to ODBC and vice versa.

2. 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 feelthat they should not stray from the current design of the core Java system.

3. 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 servesto confuse the users of the API.

4. Use strong, static typing wherever possible

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

5. 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 simpleto perform with JDBC. However, more complex SQL statements should also be possible.

They are,

1. TABLE
2. QUERY
3. FORM
4. REPORT
5. MACRO

1. Design View
2. Datasheet View

Design View: To build or modify the structure of a table we work in the table design view.We can specify what kind of data will be hold.

Datasheet View: To add, edit or analyses the data itself we work in tables datasheet viewmode.

Query: A query is a question that has to be asked the data. Access gathers data that answersthe question from one or more table. The data that make up the answer is either dynaset (ifyou edit it) or a snapshot (it cannot be edited).Each time we run query, we get latest information in the dynaset. Access either displays the dynaset or snapshot for us to view or perform an action on it, such as deleting or updating.

7.SAMPLE CODE

Admin.jsp

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<title>Admin Login</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

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

<link rel="stylesheet" type="text/css" href="css/coin-slider.css" />

<script type="text/javascript" src="js/cufon-yui.js"></script>

<script type="text/javascript" src="js/cufon-titillium-250.js"></script>

<script type="text/javascript" src="js/jquery-1.4.2.min.js"></script>

<script type="text/javascript" src="js/script.js"></script>

<script type="text/javascript" src="js/coin-slider.min.js"></script>

<script language="javascript" type="text/javascript">

function valid()

{

var na3=document.s.userid.value;

if(na3=="")
```

```
{
alert("Please Enter Name");
document.s.userid.focus();
return false;
}
else
{

}
var na4=document.s.pass.value;
if(na4=="")

{
alert("Please Enter Password");
document.s.pass.focus();
return false;
}


}
```

```
</script>



<style type="text/css">
<!--
.style2 {font-size: 40px}
.style3 {font-size: 35px; color: #fc6400; }
.style4 {color: #FFFFFF}
.style5 {font-family: "Times New Roman", Times, serif}
.style7 {font-size: 25px}
-->
</style>
</head>
<body>
<div class="main">
  <div class="header">
    <div class="header_resize">
      <div class="logo">
        <h1><a href="index.html" class="style2">Detection of fake online reviews using semi supervised and supervised learning</a></h1>
      </div>
```

```
    <div class="clr"></div>
    <div class="menu_nav">

  <ul>
      <li><a href="index.html"><span>Home Page</span></a></li>
      <li class="active"><a href="a_login.jsp"><span>admin</span></a></li>
      <li><a href="u_login.jsp"><span>User</span></a></li>
    </ul>
    </div>
    <div class="clr"></div>
    <div class="slider">
     <div id="coin-slider"> <a href="#"><img src="images/slide1.jpg" width="960" height="320" alt="" /></a> <a href="#"><img src="images/slide2.jpg" width="960" height="320" alt="" /></a> <a href="#"><img src="images/slide3.jpg" width="960" height="320" alt="" /></a> </div>
     <div class="clr"></div>
    </div>
    <div class="clr"></div>
   </div>
  </div>
```

```
 <div class="content">
  <div class="content_resize">
   <div class="mainbar">
    <div class="article">
     <p align="center" class="style3">Admin LogIn</p>
```

```
<p align="center"><img src="images/alogin.jpg" width="210" height="109" /></p>
<form name="s" action="a_authentication.jsp" method="post" onSubmit="return valid()"  ons target="_top">
 <table align="center" border="1" width="51%" height="14
<tr>
  <td width="48%" height="25" bgcolor="#fc6400" class="style4 style2"><span class="style9 style4 style5 style7"> Name </span></td>
    <td width="55%" height="25" bgcolor="#fc6400"><input type="text" name="userid" size="15" /></td>
  </tr>
  <tr>
    <td width="48%" height="25" bgcolor="#fc6400" class="style5 style2"> <span class="style9 style4 style5 style7">Password</span></td>
    <td width="55%" height="25" bgcolor="#fc6400"><input type="password" name="pass" size="15" /></td>
```

```
</tr>
<tr>
  <td height="78" colspan="2" bgcolor="#999999"><p align="center">
    <input type="submit"  value="Login" name="B1" />
    <input type="reset" value="Reset" name="B2" />
  </td>
</tr>

</table>
</form>
```

Admin login

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Admin Login</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<link href="css/style.css" rel="stylesheet" type="text/css" />
<link rel="stylesheet" type="text/css" href="css/coin-slider.css" />
<script type="text/javascript" src="js/cufon-yui.js"></script>
<script type="text/javascript" src="js/cufon-titillium-250.js"></script>
```

```
<script type="text/javascript" src="js/jquery-1.4.2.min.js"></script>

<script type="text/javascript" src="js/script.js"></script>

<script type="text/javascript" src="js/coin-slider.min.js"></script>

<script language="javascript" type="text/javascript">

function valid()

{

var na3=document.s.userid.value;

if(na3=="")




{

alert("Please Enter Name");

document.s.userid.focus();

return false;

}

else

{



}

var na4=document.s.pass.value;

if(na4=="")
```

```
{
alert("Please Enter Password");
document.s.pass.focus();
return false;
}
```

8. TESTING

8.1 SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

Unit testing

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

Integration testing

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

Functional test

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

Functional testing is centered on the following items:

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

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

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

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

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

System Test

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

White Box Testing

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

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software

under test is treated, as a black box .you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works

1 Unit Testing:

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

Test strategy and approach

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

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

6.2 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 byinterface 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.

6.3 Acceptance Testing

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

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

SYSTEM TESTING

TESTING METHODOLOGIES

The following are the Testing Methodologies:

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

Unit Testing

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

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

Integration Testing

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

The following are the types of Integration Testing:

1)Top Down Integration

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

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

2. Bottom-up Integration

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

- The low-level modules are combined into clusters into clusters that perform a specific Software sub-function.

- A driver (i.e.) the control program for testing is written to coordinate test case
  input and output.
- The cluster is tested.
- Drivers are removed and clusters are combined moving upward in the program structure

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

OTHER TESTING METHODOLOGIES

User Acceptance Testing

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

Output Testing

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

Validation Checking

Validation checks are performed on the following fields.

Text Field:

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

Numeric Field:

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

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

Preparation of Test Data

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

Using Live Test Data:

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

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

Using Artificial Test Data:

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

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

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

USER TRAINING

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

expected users are people who have good knowledge of computers, the use of this system is very easy.

MAINTAINENCE

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

designing is simple and easy to understand which will make maintenance easier.

TESTING STRATEGY :

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

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

SYSTEM TESTING:

Software once validated must be combined with other system elements (e.g. Hardware, people, database). System testing verifies that all the elements are proper and that overall

system function performance is achieved. It also tests to find discrepancies between the system and its original objective, current specifications and system documentation.

UNIT TESTING:

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

In Due Course, latest technology advancements will be taken into consideration. As part of technical build-up many components of the networking system.

9.SCREEN

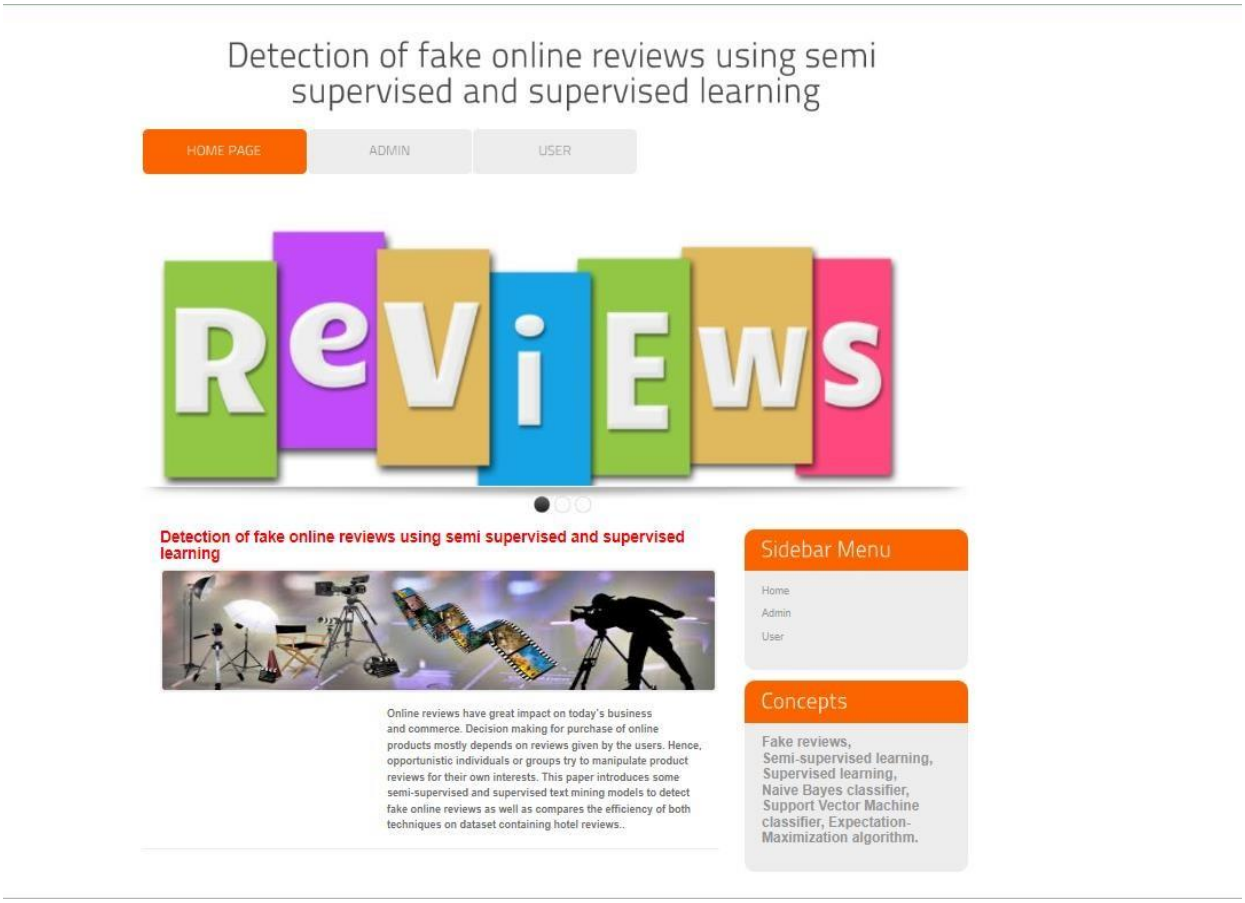
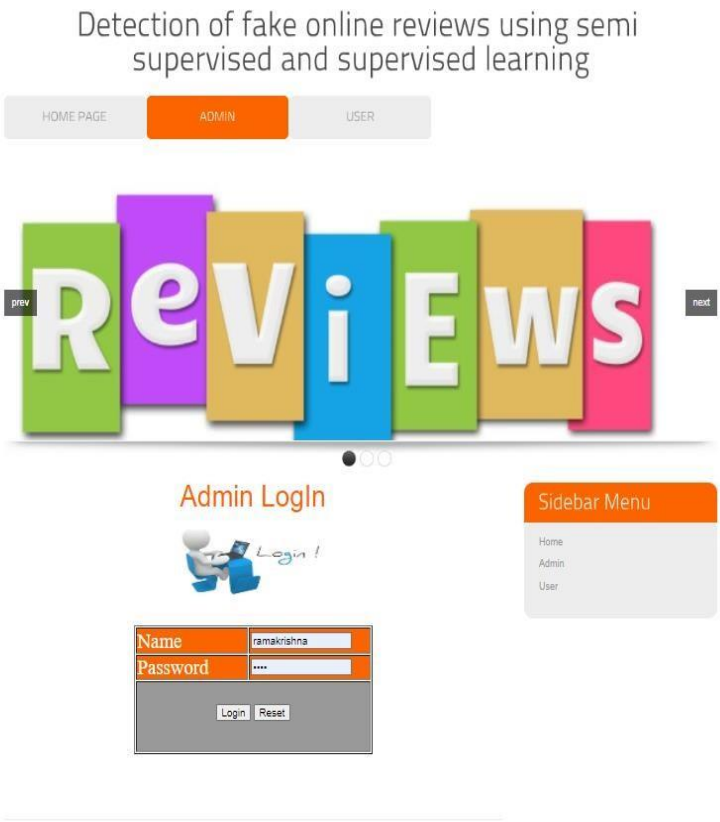


Fig-9.1:Home Page

9.2 Admin login



9.3 Home page of admin

9.4Users and Authorise

View All Users and Authorise

Admin Menu

Admin Main

Log Out

Id	User Image	User Name	User Full Details	Status
1	Submit	omkar	Click here	Authorized
2	Submit	rakesh	Click here	Authorized
3	Submit	ganesh	Click here	Authorized
4	Submit	mahesh	Click here	Authorized
5	Submit	umesh	Click here	Authorized
6	Submit	Manjunath	Click here	Authorized
7	Submit	Ramesh	Click here	Authorized
8	Submit	ramakrishna	Click here	Authorized
9	Submit	Nagendra Reddy	Click here	waiting

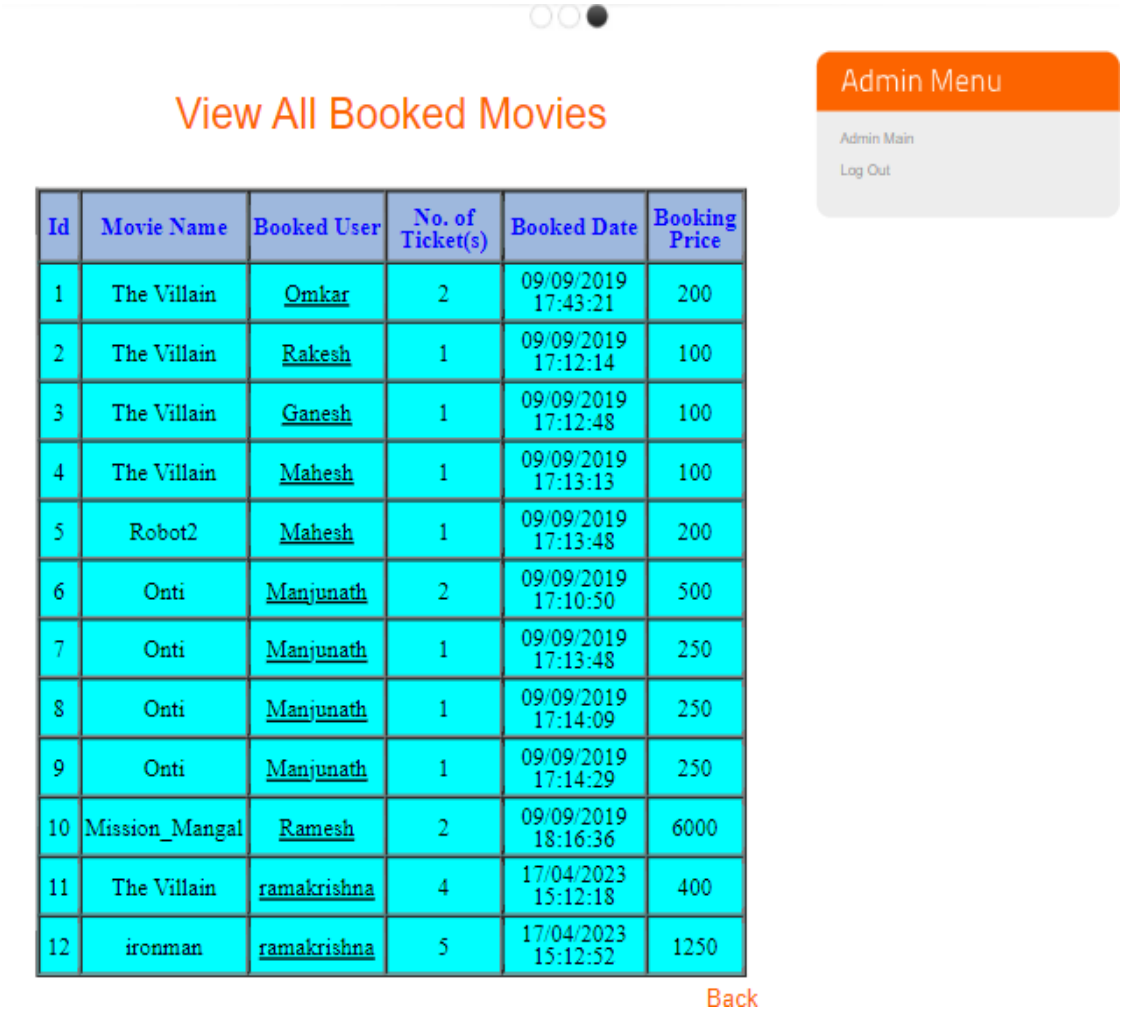
Back

9.5Addmovie

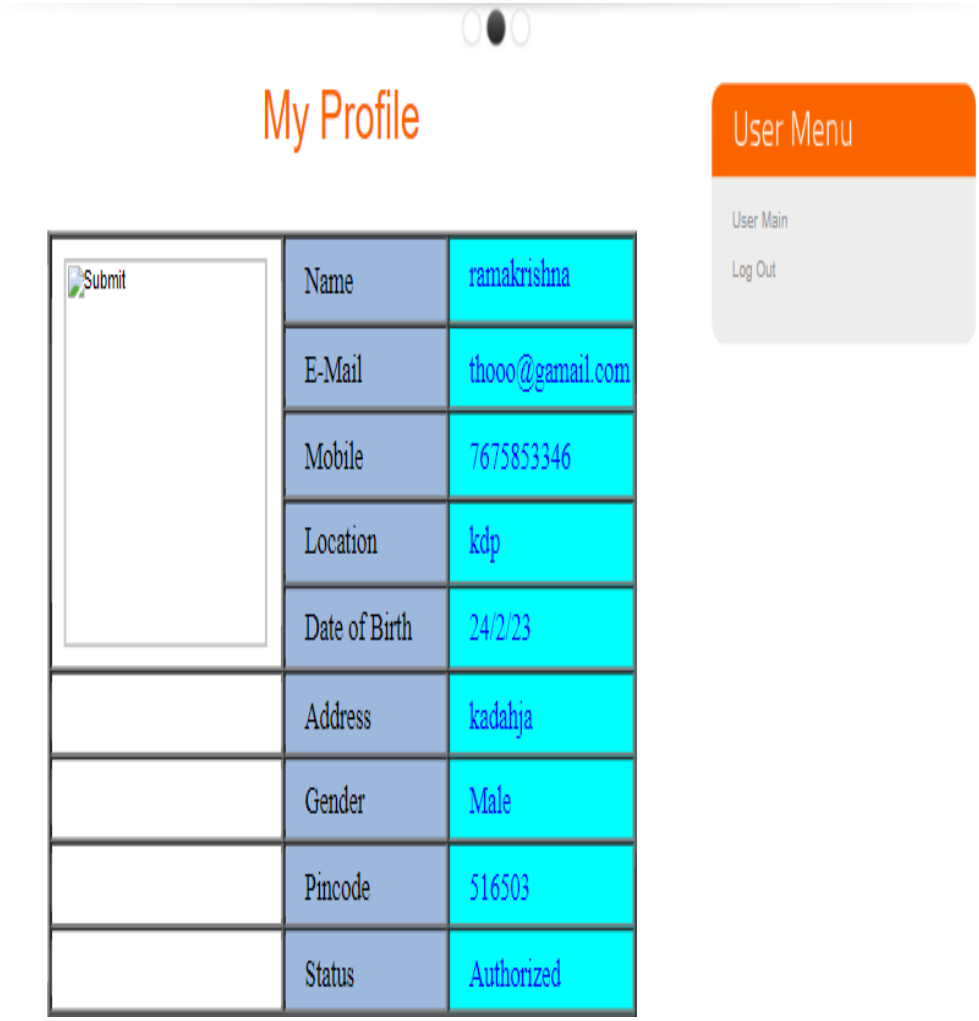
View all reviews

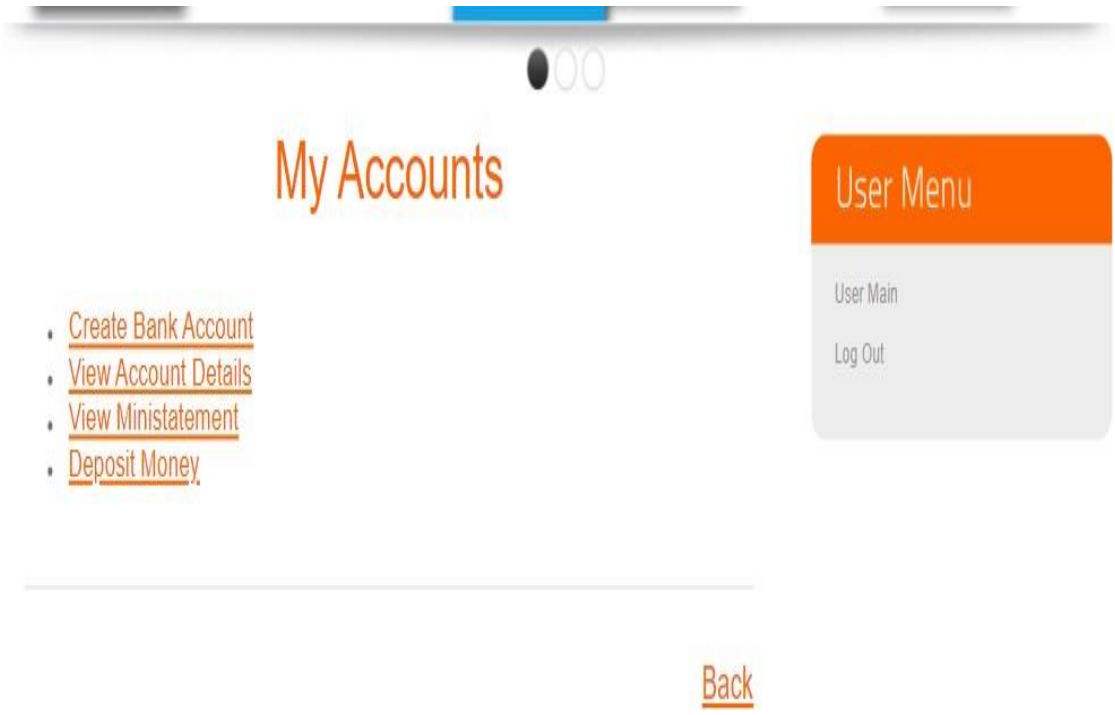
Id	User Name	Movie Name	Review Details	Date and Time	Region
3	Omkar	The Villain	nice	09/09/2019 17:35:25	vijayanagar
4	Omkar	The Villain	villain is good	09/09/2019 13:03:00	vijayanagar
5	Omkar	The Villain	villain is bad	09/09/2019 13:03:30	vijayanagar
6	Omkar	The Villain	villain is bad	09/09/2019 13:04:02	vijayanagar
7	Manjunath	Onti	It is bad movie	09/09/2019 17:16:29	Malleshwaram
8	Manjunath	Onti	It is good movie and beautiful	09/09/2019 17:16:48	Malleshwaram
9	Manjunath	Robot2	It is good movie	09/09/2019 16:12:50	Malleshwaram
10	Manjunath	Robot2	It is totally animation based project.	09/09/2019 16:36:14	Malleshwaram
11	Ramesh	Mission_Mangal	It is best movie	09/09/2019 18:17:16	Malleshwaram
12	Ramesh	Mission_Mangal	It is bad movie	09/09/2019 18:17:27	Malleshwaram
13	Manjunath	Mission_Mangal	It is good movie	09/09/2019 18:18:46	Malleshwaram
14	Manjunath	Mission_Mangal	It is beautiful film	09/09/2019 18:19:53	Malleshwaram
15	Manjunath	Mission_Mangal	It is worst movie	09/09/2019 18:37:21	Malleshwaram
16	Manjunath	Kaala	It is best movie	09/09/2019 18:45:45	Malleshwaram
17	Manjunath	Kaala	It is worst movie	09/09/2019 18:46:13	Malleshwaram
18	omkar	Rjakumara	It is good	09/09/2019 18:47:16	vijaynagar
19	omkar	Rjakumara	It is bad	09/09/2019 18:47:44	vijaynagar
20	ramakrishna	The Villain	goood	17/04/2023 12:31:27	kdp
21	ramakrishna	ironman	goood movie	17/04/2023 15:13:18	kdp

Back


Admin Menu
Admin Main
Log Out
View All Booked Movies
Id
Movie Name
Booked User
No. of Ticket(s)
Booked Date
Booking Price
1 The Villain Omkar 2 09/09/2019 17:43:21 200
2 The Villain Rakesh 1 09/09/2019 17:12:14 100
3 The Villain Ganesh 1 09/09/2019 17:12:48 100
4 The Villain Mahesh 1 09/09/2019 17:13:13 100
5 Robot2 Mahesh 1 09/09/2019 17:13:48 200
6 Onti Manjunath 2 09/09/2019 17:10:50 500
7 Onti Manjunath 1 09/09/2019 17:13:48 250
8 Onti Manjunath 1 09/09/2019 17:14:09 250
9 Onti Manjunath 1 09/09/2019 17:14:29 250
10 Mission_Mangal Ramesh 2 09/09/2019 18:16:36 6000
11 The Villain ramakrishna 4 17/04/2023 15:12:18 400
12 ironman ramakrishna 5 17/04/2023 15:12:52 1250
Back

9.5 User login


My Profile
User Menu
User Main
Log Out
Submit
Name
ramakrishna
E-Mail
thooo@gamail.com
Mobile
7675853346
Location
kdp
Date of Birth
24/2/23
Address
kadahja
Gender
Male
Pincode
516503
Status
Authorized


My Accounts
User Menu
User Main
Log Out
Create Bank Account
View Account Details
View Ministatement
Deposit Money
Back

View All Released Movies

Id	Movie Name	Ticket Price(Rs.)	Released Date	View Details
1	The Villain	100	09/09/2019	Click here
2	Robot2	200	09/09/2019	Click here
3	Rjakumara	100	09/09/2019	Click here
4	Kaala	150	04/04/2018	Click here
5	Onti	250	09/09/2019	Click here
6	Mission_Mangal	3000	08/09/2019	Click here
7	ironman	250	12/2/2023	Click here
8	kgf	300	14/2/2023	Click here
9	aakashame nee hadduraa	200	12/2/2023	Click here

Back

User Menu

User Main

Log Out

10 CONCLUSION

CONCLUSION

We have shown several semi-supervised and supervised text mining techniques for detecting fake online reviews in this research. We have combined features from several research works to create a better feature set. Also we have tried some other classifer that were not used on the previous work. Thus, we have been able to increase the accuracy of previous semisupervised techniques done by Jiten et al. We have also found out that supervised Naive Bayes classifier gives the highest accuracy. This ensures that our dataset is labeled well as we know semi-supervised model works well when reliable labeling is not available.

11. FUTURE ENHANCEMENTS

It is not possible to develop a system that makes all the requirements of the user. User requirements keep changing as the system is being used. Some of the future enhancements that canbe done to this system are

In our research work we have worked on just user reviews. In future, user behaviors can be combined with texts to construct a better model for classification. Advanced preprocessing tools for tokenization can be used to make the dataset more precise. Evaluation of the effectiveness of the proposed methodology can be done for a larger data set. This research work is being done only for English reviews. It can be done for Bangla and several other languages.

12. BIBLIOGRAPHY

[1] Chengai Sun, Qiaolin Du and Gang Tian, "Exploiting Product Related Review Features for Fake Review Detection," Mathematical Problems in Engineering, 2016.

[2] A. Heydari, M. A. Tavakoli, N. Salim, and Z. Heydari, "Detection of review spam: a survey", Expert Systems with Applications, vol. 42, no. 7, pp. 3634–3642, 2015.

[3] M. Ott, Y. Choi, C. Cardie, and J. T. Hancock, "Finding deceptive opinion spam by any stretch of the imagination," in Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT), vol. 1, pp. 309–319,

Association for Computational Linguistics, Portland, Ore, USA, June 2011.

[4] J. W. Pennebaker, M. E. Francis, and R. J. Booth, "Linguistic Inquiry and Word Count: Liwc," vol. 71, 2001.

[5] S. Feng, R. Banerjee, and Y. Choi, "Syntactic stylometry for deception detection," in Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers, Vol. 2, 2012.

[6] J. Li, M. Ott, C. Cardie, and E. Hovy, "Towards a general rule for identifying deceptive opinion spam," in Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL), 2014.

[7] E. P. Lim, V.-A. Nguyen, N. Jindal, B. Liu, and H. W. Lauw, "Detecting product review spammers using rating behaviors," in Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM), 2010.

[8] J. K. Rout, A. Dalmia, and K.-K. R. Choo, "Revisiting semi-supervised learning for online deceptive review detection," IEEE Access, Vol. 5, pp. 1319–1327, 2017.

[9] J. Karimpour, A. A. Noroozi, and S. Alizadeh, "Web spam detection by learning from small labeled samples," International Journal of Computer Applications, vol. 50, no. 21, pp. 1–5, July 2012