## FRAUD DETECTION IN ONLINE PRODUCT REVIEW SYSTEMS VIA HETEROGENEOUS GRAPH TRANSFORMER

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

By

A.RAKESH 20C45A0514

R.PRAVEEN 20C45A0516

B.AKHIL 20C45A0517

Under the Guidance of

Mr.G.PanduKrishna Assistant Professor



**Department of Computer Science and Engineering**

## JAYAMUKHI INSTITUTE OF TECHNOLOGICAL SCIENCES

#### NARSAMPET, WARANGAL-506 332

**(Affiliated to JNTUH, Accredited by NAAC ‘A’ Grade) 2023**

## JAYAMUKHI INSTITUTE OF TECHNOLOGICAL SCIENCES

**NARSAMPET, WARANGAL – 506332**

### (Autonomous, Affiliated to JNTUH, Accredited by NBA, NAAC-A)

**CERTIFICATE**



This is to certify that Major project Report entitled **“FRAUD DETECTION IN ONLINE PRODUCT REVIEW SYSTEMS VIA HETEROGENEOUS GRAPH**

**TRANSFORMER**” is a Bonafide work of the students **A.RAKESH (20C45A0514) ,R.PRAVEEN (20C45A0516), B.AKHIL (20C45A0517)** submitted requirements in partial fulfilment of the for the award of the degree of Bachelor of Technology in COMPUTER SCIENCE AND ENGINEERING during the academic year 2022-2023.

**Guide Head of the department**

**Principal**

## ACKNOWLEDGMENT

We express our immense gratitude and sincere thanks to **Dr. V. JANAKI**, principal, Jayamukhi Institute of Technological Sciences for her constant motivation and support.

We express our gratitude and sincere thanks to **Dr. P. SRINIVAS RAO**, Head of the Department of Computer science and engineering, Jayamukhi Institute of Technological Sciences for his constant Motivation and support.

We are grateful to our Project Guide **Mr.G.PanduKrishna,** Assistant Professor for his Valuable guidance and giving us the opportunity to do our work in a better manner.

We are also thankful to our Management for providing all the facilities in completing the project.

INDEX

|  |  |
| --- | --- |
| **CONTENTS** | **PAGE NUMBERS** |
| 1. **INTRODUCTION** 2. **LITERATURE REVIEW** 3. **SYSTEMREQUIREMENTS**    1. Hardware    2. Software 4. **EXISTING SYSTEM**    1. Disadvantages 5. **PROPOSED SYSTEM**    1. Advantages 6. **INPUT AND OUTPUT DESIGN**    1. Input    2. Output 7. **MODULE**    1. Admin    2. User 8. **SYSTEM STUDY**    1. Feasibility Study       1. Economic Feasibility       2. Technical Feasibility       3. Social Feasibility    2. Preliminary Investigation       1. Request Clarification       2. Feasibility Study       3. Request Approval 9. **SYSTEM DESIGN**    1. Architecture Diagram    2. Data Flow Diagram       1. Admin       2. User | 1-3  4-8  9  10  11  12-13  14  15-18  19-22 |

|  |  |
| --- | --- |
| 1. **UML DIAGRAM**    1. Class Diagram    2. Sequence Diagram   10.3.Use Case Diagram   1. **SOFTWARE ENVIRONMENT**    1. Front end or User Interface Design    2. Features of JAVA    3. Java Script    4. HTML    5. Java Data base Connecting       1. ODBC       2. JDBC    6. JSP    7. TOM CAT   **11. SYSTEM TESTING**   * 1. Types of Testing   2. Testing Methodoligies   3. Other Testing Methodologies  1. **RESULTS** 2. **CONCLUSION** 3. **REFERENCES** | 23-25  26-57  58-65  66-75  76  77-78 |

# ABSTRACT

In online product review systems, users are allowed to submit reviews about their purchased items or services. However, fake reviews posted by fraudulent users often mislead consumers and bring losses to enterprises. Traditional fraud detection algorithm mainly utilizes rule-based methods, which is insufficient for the rich user interactions and graph-structured data. In recent years, graph-based methods have been proposed to handle this situation, but few prior works have noticed the camouflage fraudster’s behavior and inconsistency heterogeneous nature. Existing methods have either not addressed these two problems or only partially, which results in poor performance. Alternatively, we propose a new model named Fraud Aware Heterogeneous Graph Transformer (FAHGT), to address camouflages and inconsistency problems in a unified manner. FAHGT adopts a type-aware feature mapping mechanism to handle heterogeneous graph data, then implementing various relation scoring methods to alleviate inconsistency and discover camouflage. Finally, the neighbors’ features are aggregated together to build an informative representation. Experimental results on different types of real-world datasets demonstrate that FAHGT outperforms the state-of-the-art baselines.

# INTRODUCTION

Internet services have brought human beings with e-commerce, social networking, and entertainment platforms, which not only facilitate information exchange but also provide chances to fraudsters. Fraudsters disguise themselves as ordinary users to publish spam information [1] or collect user privacy, compromising the interest of both platforms and users. In addition, multiple entities on the Internet are connected with multiple relationships. Traditional machine learning algorithms cannot handle this complicated heterogeneous graph data well. The current approach is to model the data as a heterogeneous information network so that similarities in characteristics and structure of fraudsters can be discovered. Due to its effectiveness in learning node representations and discovering structure pattern, graph neural networks have been drawn attention in fraud detection domain including product review, mobile application distribution cybercrime identification and financial services. However, most existing GNN based solutions just directly apply homogeneous GNNs, ignoring the underlying heterogeneous graph nature and camouflage node behaviors. This problem has drawn great attention with many solutions proposed. GraphConsis found that there are three inconsistency problems in fraud detection and CAREGNN further proposed two camouflage behaviors. These problems could be summarized as follows:

**Camouflage:** Previous work showed that crowd workers could adjust their behavior to alleviate their suspicion via connecting to benign entities like connecting to highly reputable users, disguise fraudulent URLs with special characters or generate domain-independent fake reviews via generative language model to conceal their suspicious activities.

**Inconsistency:** Two users with distinct interests could be connected via reviewing a common product such as food or movies. Direct aggregation makes GNNs hardly distinguish the unique semantic user pattern. Also, if a likely to be distrustful if they are connected by common activity relation since fraudulent users tend to generate spam content in the same short period.

To address the above two problems, many methods have been proposed. GraphConsis addresses the inconsistency problem by computing the similarity score between node embeddings, which cannot distinguish nodes with different types. CAREGNN enhances GNN-based fraud detectors against camouflaged fraudsters by reinforcement learning based neighbor selector and relation aware aggregator. Its performance still suffers from the heterogeneous graph. In this paper, we introduce the Fraud Aware

Heterogeneous Graph Transformer(FAHGT), where we propose heterogeneous mutual attention to address the inconsistency problem and design a label-aware neighbor selector to solve the camouflage problem. Both

are implemented in a unified manner called the ‘‘score head mechanism’’. We demonstrate the fraud detection performance of FAHGT on many real-world datasets. It is verified that FAHGT can considerably improve F1 score, KS and AUC over several baselines.

# LITERATURE REVIEW

L1: Fraudetector: A graph-mining-based framework for fraudulent phone call detection. Authors:

S. Tseng, J. Ying, C. Huang, Y. Kao, and K. Chen

Description:

In recent years, fraud is increasing rapidly with the development of modern technology and global communication. Although many literatures have addressed the fraud detection problem, these existing works focus only on formulating the fraud detection problem as a binary classification problem. Due to limitation of information provided by telecommunication records, such classifier-based approaches for fraudulent phone call detection normally do not work well. In this paper, we develop a graph-mining-based fraudulent phone call detection framework for a mobile application to automatically annotate fraudulent phone numbers with a "fraud" tag, which is a crucial prerequisite for distinguishing fraudulent phone calls from normal phone calls. Our detection approach performs a weighted HITS algorithm to learn the trust value of a remote phone number. Based on telecommunication records, we build two kinds of directed bipartite graph: i) CPG and ii) UPG to represent telecommunication behavior of users. To weight the edges of CPG and UPG, we extract features for each pair of user and remote phone number in two different yet complementary aspects:

1) duration relatedness (DR) between user and phone number; and 2) frequency relatedness (FR) between user and phone number. Upon weighted CPG and UPG, we determine a trust value for each remote phone number. Finally, we conduct a comprehensive experimental study based on a dataset collected through an anti-fraud mobile application, Whoscall. The results demonstrate the effectiveness of our weighted HITS- based approach and show the strength of taking both DR and FR into account in feature extraction.

L2 :Fraudster detection via graph convolutional networks in online app review system. Authors:

J.Wang, R. Wen, C. Wu, Y. Huang, and J. Xion

Description:

Online review system enables users to submit reviews about the products. However, the openness of Internet and monetary rewards for crowdsourcing tasks stimulate a large number of fraudulent users to write fake reviews and post advertisements to interfere the rank of apps. Existing methods for detecting spam reviews have been successful but they usually aims at e-commerce (e.g. Amazon, eBay) and recommendation (e.g.

Yelp, Dianping) systems. Since the behaviors of fraudulent users are complexity and varying across different review platforms, existing methods are not suitable for fraudster detection in online app review

system.

To shed light on this question, we are among the first to analyze the intentions of fraudulent users from different review platforms and categorize them by utilizing characteristics of contents (similarity, special symbols) and behaviors (timestamps, device, login status). With a comprehensive analysis of spamming activities and relationships between normal and malicious users, we design and present FdGars, the first graph convolutional network approach for fraudster detection in online app review system. Then we evaluate FdGars on real-world large-scale dataset (with 82,542 nodes and 42,433,134 edges) from Tencent App Store. The result demonstrates that *F*1-score of FdGars can achieve 0.938+, which outperforms several baselines and state-of-art fraudsters detecting methods. Moreover, we deploy FdGars on Tencent Beacon Anti-Fraud Platform to show its effectiveness and scalability. To the best of our knowledge, this is the first work to use graph convolutional networks for fraudster detection in the large-scale online app review system. It is worth to mention that FdGars can uncover malicious accounts even the data lack of labels in anti-spam tasks.

**L3: Spam review detection with graph convolutional networks**. **Authors:** Li, Z. Qin, R. Liu, Y. Yang, and D. Li

Description:

Reviews on online shopping websites affect the buying decisions of customers, meanwhile, attract lots of spammers aiming at misleading buyers. Xianyu, the largest second-hand goods app in China, suffering from spam reviews. The anti-spam system of Xianyu faces two major challenges: scalability of the data and adversarial actions taken by spammers. In this paper, we present our technical solutions to address these challenges. We propose a large-scale anti-spam method based on graph convolutional networks (GCN) for detecting spam advertisements at Xianyu, named GCN-based Anti-Spam (GAS) model. In this model, a heterogeneous graph and a homogeneous graph are integrated to capture the local context and global context of a comment. Offline experiments show that the proposed method is superior to our baseline model in which the information of reviews, features of users and items being reviewed are utilized. Furthermore, we deploy our system to process million-scale data daily at Xianyu. The online performance also demonstrates the effectiveness of the proposed method.

L4:Alleviating the inconsistency problem of applying graph neural network to fraud detection.

**Authors**: Z. Liu, Y. Dou, P. S. Yu, Y. Deng, and H. Peng

Description:

Graph-based models have been widely used to fraud detection tasks. Owing to the development of Graph Neural Networks~(GNNs), recent works have proposed many GNN-based fraud detectors based on either

homogeneous or heterogeneous graphs. These works leverage existing GNNs and aggregate the neighborhood information to learn the node embeddings, which relies on the assumption that the neighbors share similar context, features, and relations. However, the inconsistency problem incurred by fraudsters is hardly investigated, i.e., the context inconsistency, feature inconsistency, and relation inconsistency. In this paper, we introduce these inconsistencies and design a new GNN framework, GraphConsis, to tackle the inconsistency problem: (1) for the context inconsistency, we propose to combine the context embeddings with node features; (2) for the feature inconsistency, we design a consistency score to filter the inconsistent neighbors and generate corresponding sampling probability; (3) for the relation inconsistency, we learn the relation attention weights associated with the sampled nodes. Empirical analysis on four datasets demonstrates that the inconsistency problem is critical in fraud detection tasks. Extensive experiments show the effectiveness of GraphConsis. We also released a GNN-based fraud detection toolbox with implementations of SOTA models.

**L5: Enhancing graph neural network-based fraud detectors against camouflaged fraudsters. Authors:** Y. Dou, Z. Liu, L. Sun, Y. Deng, H. Peng, and P. S. Yu

Description:

Graph Neural Networks (GNNs) have been widely applied to fraud detection problems in recent years, revealing the suspiciousness of nodes by aggregating their neighborhood information via different relations. However, few prior works have noticed the camouflage behavior of fraudsters, which could hamper the performance of GNN-based fraud detectors during the aggregation process. In this paper, we introduce two types of camouflages based on recent empirical studies, i.e., the feature camouflage and the relation camouflage. Existing GNNs have not addressed these two camouflages, which results in their poor performance in fraud detection problems. Alternatively, we propose a new model named CAmouflage- REsistant GNN (CARE-GNN), to enhance the GNN aggregation process with three unique modules against camouflages. Concretely, we first devise a label-aware similarity measure to find informative neighboring nodes. Then, we leverage reinforcement learning (RL) to find the optimal amounts of neighbors to be selected. Finally, the selected neighbors across different relations are aggregated together. Comprehensive experiments on two real-world fraud datasets demonstrate the effectiveness of the RL algorithm. The proposed CAREGNN also outperforms state-of-the-art GNNs and GNN-based fraud detectors.

L6: Adversary situation awareness via heterogeneous graph convolutional networks. Authors:

R. Wen, J. Wang, C. Wu, and J. Xiong

Description:

Given a large graph with millions of vertices and different types, how can we spot anomalies and find potential adversaries in time? Most graph-based fraud detection algorithms focus on finding dense blocks, discovering local subgraphs, designing belief propagation, and latent factor models. However, as fraudsters in online social networks gradually become adversarial, distributed, and invisible, it is easy for them to evade traditional detection methods. Even worse, existing detection systems are fragile to the new attacks. Once attackers update their tragedies, they will be inefficient. Our focus is to detect potential adversaries as soon as possible. We design and propose ASA (Adversary Situation Awareness), a system that (a) uncovers potential adversaries in time, (b) utilizes heterogeneous information in the graph, and (c)is effective in real- world data. We do experiments on a heterogeneous graph including 198,541 nodes with five types and 224,384 edges. The result shows that ASA can spot potential adversaries and successfully recovers 60 of potential abnormal users within a few minutes.

L7: Key player identification in underground forums over attributed heterogeneous information network embedding framework.

**Authors:** Y. Zhang, Y. Fan, Y. Ye, L. Zhao, and C. Shi

Description:

Online underground forums have been widely used by cybercriminals to exchange knowledge and trade in illicit products or services, which have played a central role in the cybercriminal ecosystem. In order to combat the evolving cybercrimes, in this paper, we propose and develop an intelligent system named iDetective to automate the analysis of underground forums for the identification of key players (i.e., users who play the vital role in the value chain). In iDetective, we first introduce an attributed heterogeneous information network (AHIN) for user representation and use a meta-path based approach to incorporate higher-level semantics to build up relatedness over users in underground forums; then we propose Player2Vec to efficiently learn node (i.e., user) representations in AHIN for key player identification. In Player2Vec, we first map the constructed AHIN to a multi-view network which consists of multiple single- view attributed graphs encoding the relatedness over users depicted by different designed meta-paths; then we employ graph convolutional network (GCN) to learn embeddings of each single-view attributed graph; later, an attention mechanism is designed to fuse different embeddings learned based on different single- view attributed graphs for final representations. Comprehensive experiments on the data collections from different underground forums (i.e., Hack Forums, Nulled) are conducted to validate the effectiveness of iDetective in key player identification by comparisons with alternative approaches

L8: Deep learning on graphs.

**Authors :** Z. Zhang, P. Cui, and W. Zhu

Description:

Deep learning has been shown to be successful in a number of domains, ranging from acoustics, images, to natural language processing. However, applying deep learning to the ubiquitous graph data is non-trivial because of the unique characteristics of graphs. Recently, substantial research efforts have been devoted to applying deep learning methods to graphs, resulting in beneficial advances in graph analysis techniques. In this survey, we comprehensively review the different types of deep learning methods on graphs. We divide the existing methods into five categories based on their model architectures and training strategies: graph recurrent neural networks, graph convolutional networks, graph autoencoders, graph reinforcement learning, and graph adversarial methods. We then provide a comprehensive overview of these methods in a systematic manner mainly by following their development history. We also analyze the differences and compositions of different methods. Finally, we briefly outline the applications in which they have been used and discuss potential future research directions.

L9:Heterogeneous graph attention network

**Authors :** X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. S. Yu **Description:**

Graph neural network, as a powerful graph representation technique based on deep learning, has shown superior performance and attracted considerable research interest. However, it has not been fully considered in graph neural network for heterogeneous graph which contains different types of nodes and links. The heterogeneity and rich semantic information bring great challenges for designing a graph neural network for heterogeneous graph. Recently, one of the most exciting advancements in deep learning is the attention mechanism, whose great potential has been well demonstrated in various areas. In this paper, we first propose a novel heterogeneous graph neural network based on the hierarchical attention, including node- level and semantic-level attentions. Specifically, the node-level attention aims to learn the importance between a node and its metapath based neighbors, while the semantic-level attention is able to learn the importance of different meta-paths. With the learned importance from both node-level and semantic-level attention, the importance of node and meta-path can be fully considered. Then the proposed model can generate node embedding by aggregating features from meta-path based neighbors in a hierarchical manner. Extensive experimental results on three real-world heterogeneous graphs not only show the superior performance of our proposed model over the state-of-the-arts, but also demonstrate its potentially good interpretability for graph analysis.

## SYSTEM REQUIREMENTS:

* H/W System Configuration **:-**

|  |  |
| --- | --- |
| * Processor | - Pentium –IV |

|  |  |
| --- | --- |
| * RAM | - 4 GB (min) |
| * Hard Disk | - 20 GB |
| * Key Board | - Standard Windows Keyboard |
| * Mouse | - Two or Three Button Mouse |
| * Monitor | - SVGA |

## SOFTWARE REQUIREMENTS:

|  |  |
| --- | --- |
|  Operating system | **:** Windows 7 Ultimate. |
|  Coding Language | **:** Java**/**J2EE(JSP servlet) |
|  Front-End | **:** J2EE |
|  Back-End | **:** Django-ORM |
|  Designing | **:** Html, Css, Javascript. |
|  Data Base | **:** MySQL (WAMP Server). |

###### EXISTING SYSTEM

ChebNet and GCN are proposed to improve efficiency by using approximation. For GNNs on spatial domain, GraphSAGE samples a tree rooted at each node and computes the root’s hidden representation by hierarchically aggregating hidden node representations from the bottom to top. GAT further proposes to learn in the spatial domain by computing different importance of neighbor nodes via the masked self attention mechanism. All these methods are designed for homogeneous graphs. They cannot be directly applied to a heterogeneous graph with multiple types of entities and relations.

In recent years, lots of heterogeneous GNN based methods have been developed. HAN , HAHE , and Deep- HGNN transforms a heterogeneous graph into several homogeneous graphs based on handcrafted meta-paths, applies GNN separately on each graph, and aggregates the output representations by attention mechanism. GraphInception constructs meta-paths between nodes with the same object type. HetGNN first samples a fixed number of neighbors via random walk strategy. Then it applies a hierarchical aggregation mechanism for intratype and intertype aggregation. HGT extends transformer architecture to heterogeneous graphs. They directly calculate attention scores for all the neighbors of a target node and perform aggregation accordingly without considering domain knowledge.

For relation-aware graph fraud detectors, their main solution is to build multiple homogeneous graphs based on edge type information of the original graph then perform type independent node level aggregation and graph level concatenation. GEM learns weighting parameters for different homogeneous subgraph. Player2Vec and Semi GNN both adopt attention mechanism in feature aggregation and Semi GNN further leverages a structure loss to guarantee the node embeddings homophily. Some works directly aggregate heterogeneous information in the graph. For instance, under a user-review-item heterogeneous graph, GAS learns a unique set of aggregators for different node types and updates the embeddings of each node type iteratively.

Disadvantages :

* + In the existing work, the system did not implement Fraud Aware Heterogeneous Graph Transformer(FAHGT) to measure frauds exactly.
  + This system is less performance due to lack of META RELATION SCORING.

#### PROPOSED SYSTEM:

 GraphConsis addresses the inconsistency problem by computing the similarity score between node embeddings, which cannot distinguish nodes with different types. CAREGNN enhances GNN-based fraud detectors against camouflaged fraudsters by reinforcement learning based neighbor selector and relation aware aggregator. Its performance still suffers from the heterogeneous graph.

 In this paper, the system introduces the Fraud Aware Heterogeneous Graph Transformer(FAHGT), where we propose heterogeneous mutual attention to address the inconsistency problem and design a label-aware neighbor selector to solve the camouflage problem. Both are implemented in a unified manner called the “score head mechanism”. We demonstrate the effectiveness and efficiency of FAHGT on many real world datasets. Experimental results suggest that FAHGT can significantly improve KS and AUC over state-of-the-art GNNs as well as GNN-based fraud detectors.

##### Advantages :

The advantages of FAHGT can be summarized as follows.

* Heterogeneity: FAHGT is able to handle heterogeneous graphs with multi-relation and multi-node type without designing meta-path manually.
* Adaptability: FAHGT attentively selects neighbors given a noise graph from real-world data. The selected neighbors are either informative for feature aggregation or risky for fraud detection.
* Efficiency: FAHGT admits a low computational complexity via a parallelizable multi-head mechanism in relation scoring and feature aggregation.
* Flexibility: FAHGT injects domain knowledge by introducing a flexible relation scoring mechanism. The score of a relation connecting two nodes not only comes from direct feature interaction but is also constrained by domain knowledge.

# INPUT AND OUTPUT DESIGN:

#### INPUT DESIGN:

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

This system has input screens in almost all the modules. Error messages are developed to alert the user whenever he commits some mistakes and guides him in the right way so that invalid entries are not made. Let us see deeply about this under module design.

Input design is the process of converting the user created input into a computer-based format. The goal of the input design is to make the data entry logical and free from errors. The error is in the input are controlled by the input design. The application has been developed in user-friendly manner. The forms have been designed in such a way during the processing the cursor is placed in the position where must be entered. The user is also provided with in an option to select an appropriate input from various alternatives related to the field in certain cases.

Validations are required for each data entered. Whenever a user enters an erroneous data, error message is displayed and the user can move on to the subsequent pages after completing all the entries in the current page.

#### OUTPUT DESIGN:

The Output from the computer is required to mainly create an efficient method of communication within the company primarily among the project leader and his team members, in other words, the administrator and the clients. The output of VPN is the system which allows the project leader to manage his

clients in terms of creating new clients and assigning new projects to them, maintaining a record of the project validity and providing folder level access to each client on the user side depending on the projects allotted to him. After completion of a project, a new project may be assigned to the client. User authentication procedures are maintained at the initial stages itself. A new user may be created by the administrator himself or a user can himself register as a new user but the task of assigning projects and validating a new user rests with the administrator only.

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

# Modules

##### Admin:

In this module, admin has to login with valid username and password. After login successful he can do some operations such as View and Authorize Users,Add & View Categories, Add & View Sub-Categories, Add Product Items, Top Products Results,View Users Search History,View Fraud Detection On Product Reviews, View Products with Ranks and Comments, View Products Ranks Results. **Viewing and Authorizing Users :**

In this module, the admin views all users details and authorize them for login permission. User Details such as User Name, Address, Email Id and Mobile Number.

##### User :

In this module, there are n numbers of users are present. User should register before doing some. After registration successful he can login by using valid user name and password. Login successful he will do some operations Register and Login,My Profile, Search Products & Give Review,View Top Products,My Search History.

**Viewing Profile Details :**

In this module, the user can see their own profile details, such as their address, email, mobile number, profile Image

# 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  TECHNICAL FEASIBILITY

SOCIAL FEASIBILITY

#### ECONOMICAL FEASIBILITY

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

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

#### SOCIAL FEASIBILITY

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

must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

## PRELIMINARY INVESTIGATION

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

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

#### REQUEST CLARIFICATION

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

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

#### FEASIBILITY ANALYSIS

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

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

##### Operational Feasibility

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

##### Economic Feasibility

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

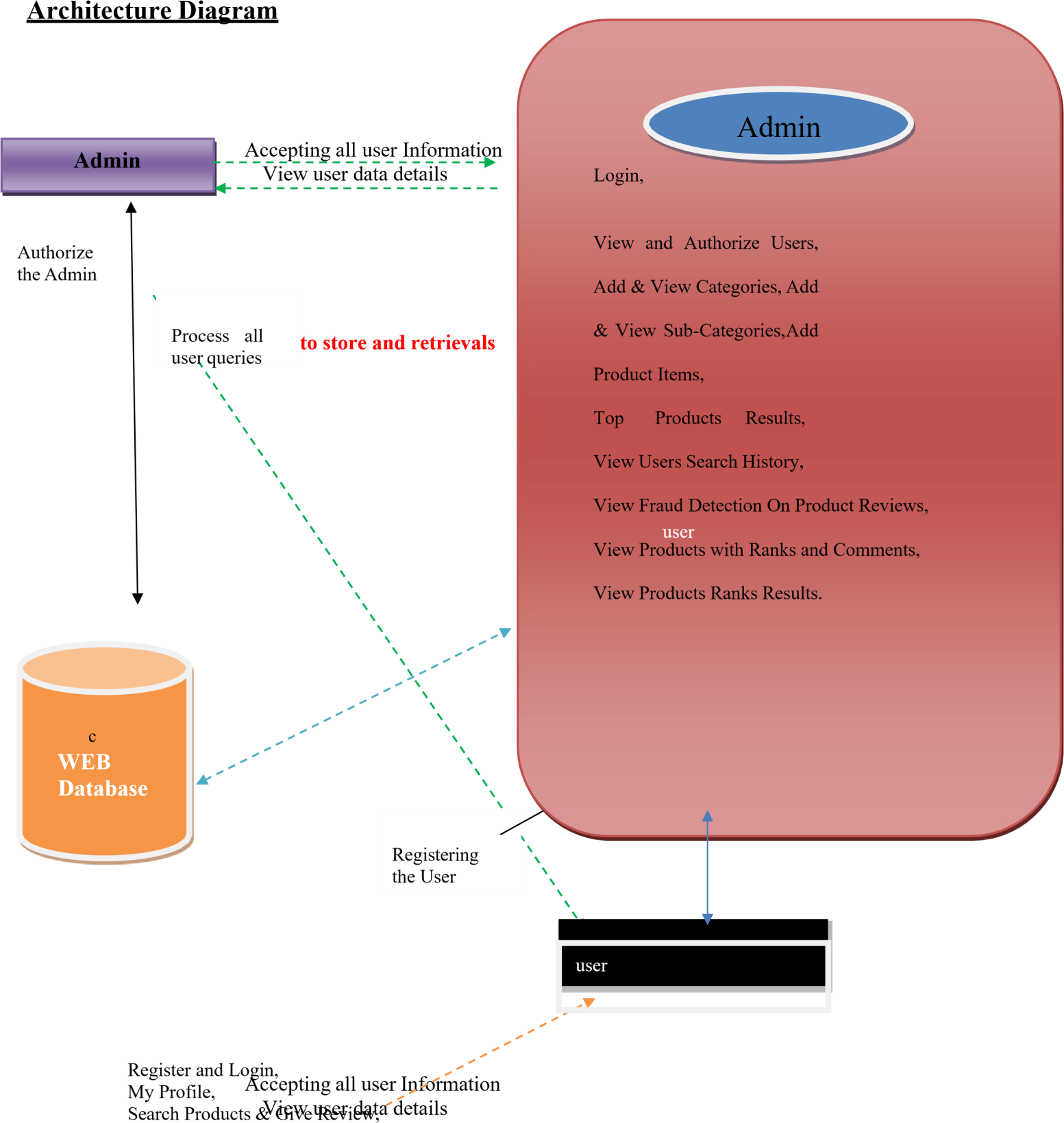
##### Technical Feasibility

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

#### REQUEST APPROVAL

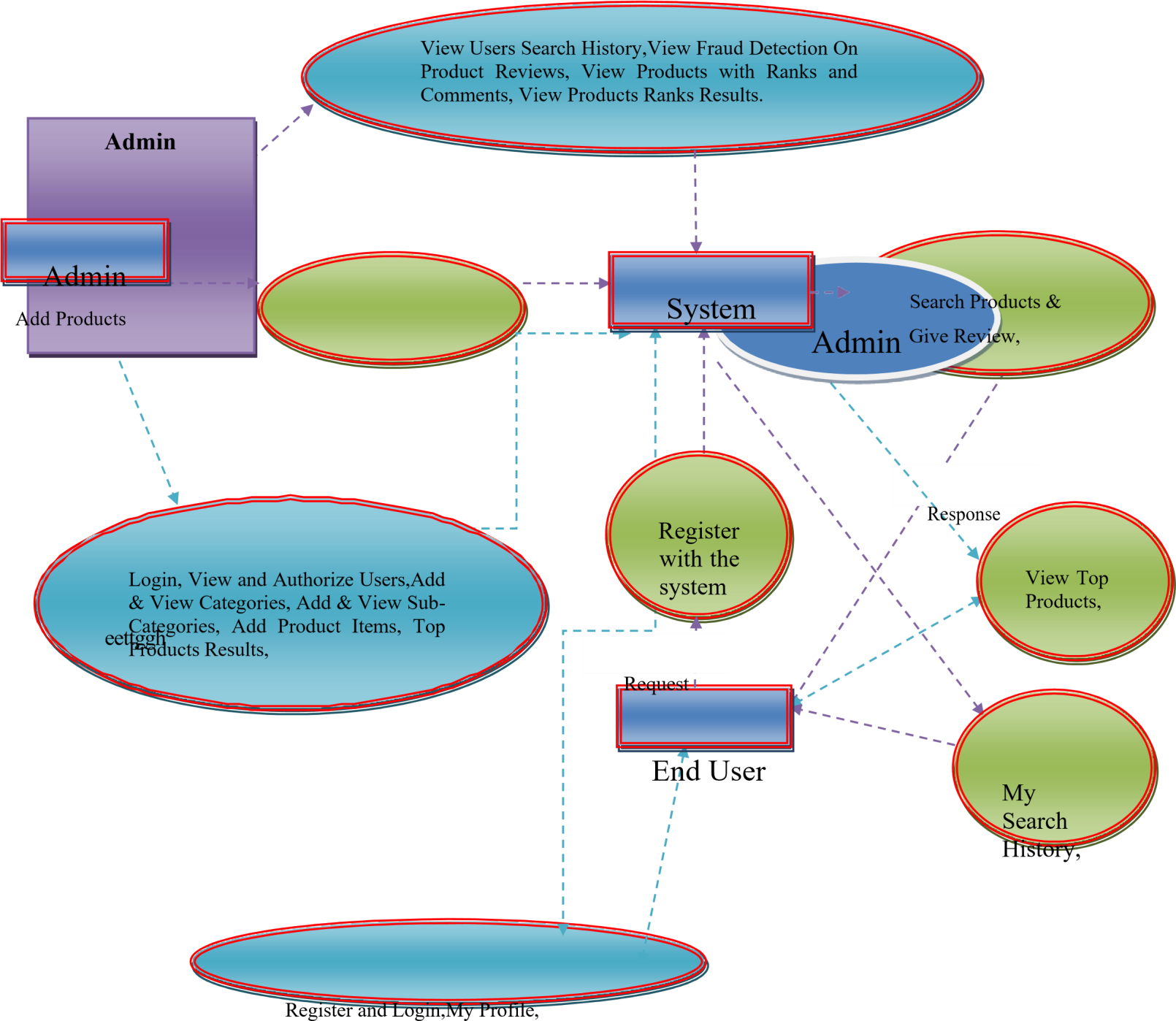
Not all request projects are desirable or feasible. Some organization receives so many project requests from client users that only few of them are pursued. However, those projects that are both feasible and desirable should be put into schedule. After a project request is approved, it cost, priority,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

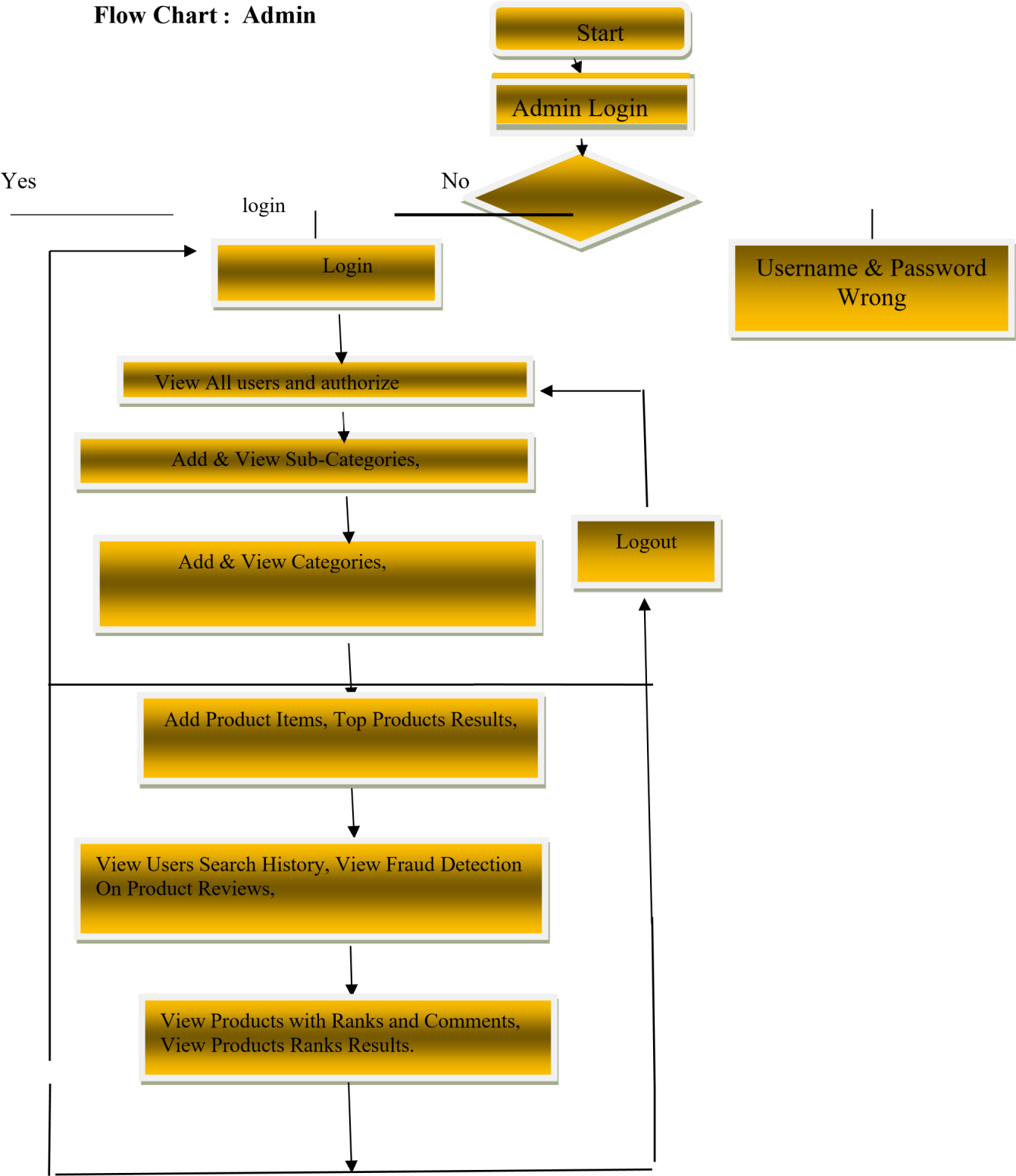
# SYSTEM DESIGN AND DEVELOPMENT



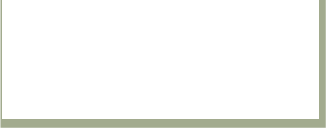
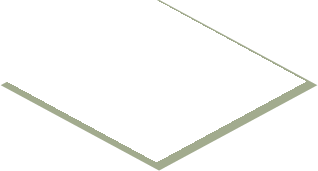
View Top Products, My search history

#### DATA FLOW DIAGRAM:





##### Flow Chart : User

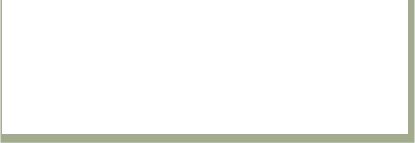
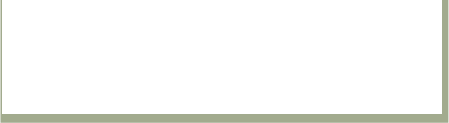


**Username & Password Wrong**

**User Register**

**Start**

**Login**



My Search History,

View Profile

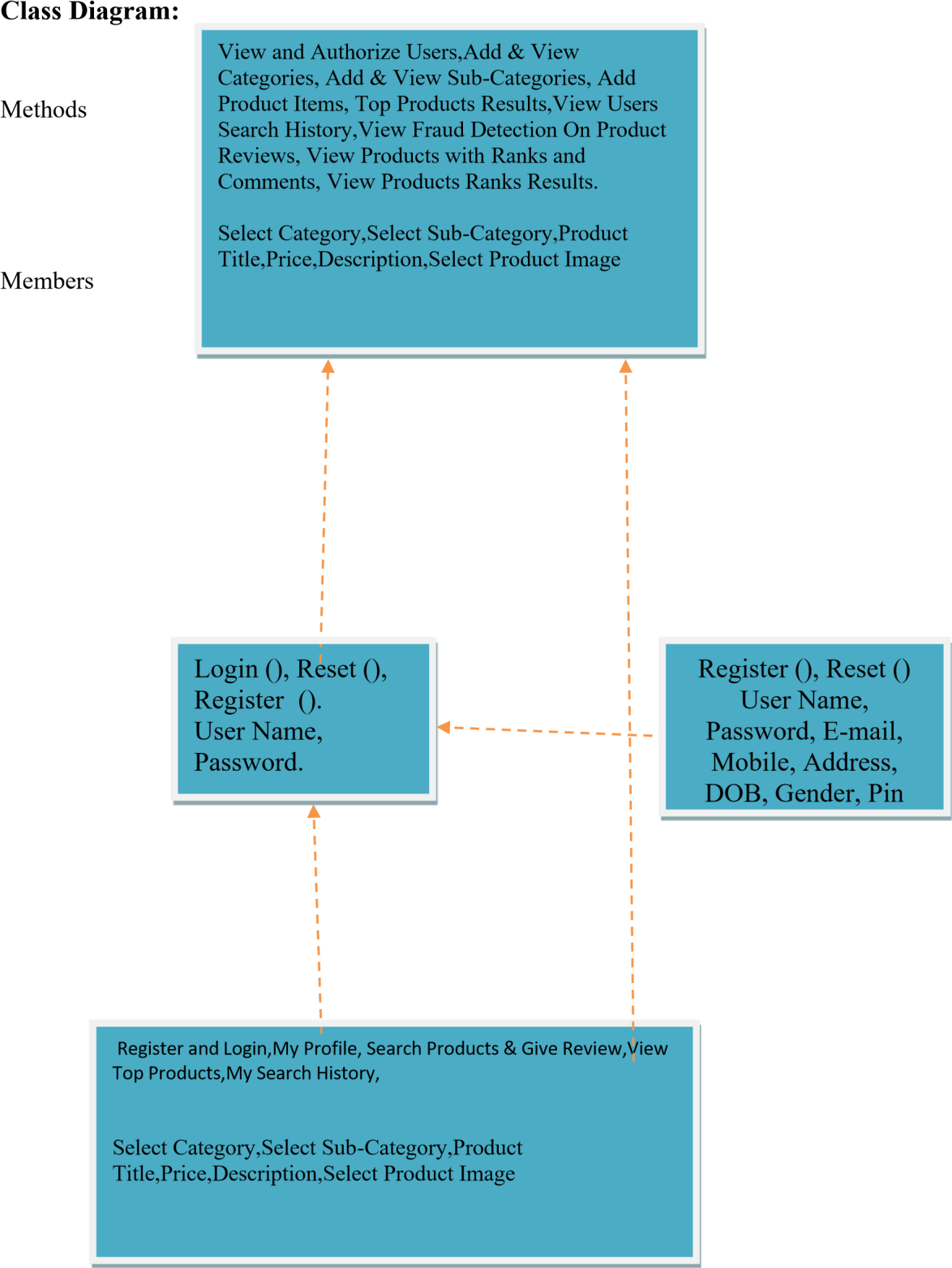
View Top Products,

**Log Out**

Search Products & Give Review,

Yes No

## UML DIAGRAM





Admin

Register and Login

View and Authorize Users,

Add & View Categories,

Search Products & Give Review,

Add & View Sub-Categories,

Add Product Items,

. My Search History,

Top Products Results,

View Users Search History,

View Fraud Detection On Product Reviews,

View Top Products

View Products with Ranks and Comments,

View Products Ranks

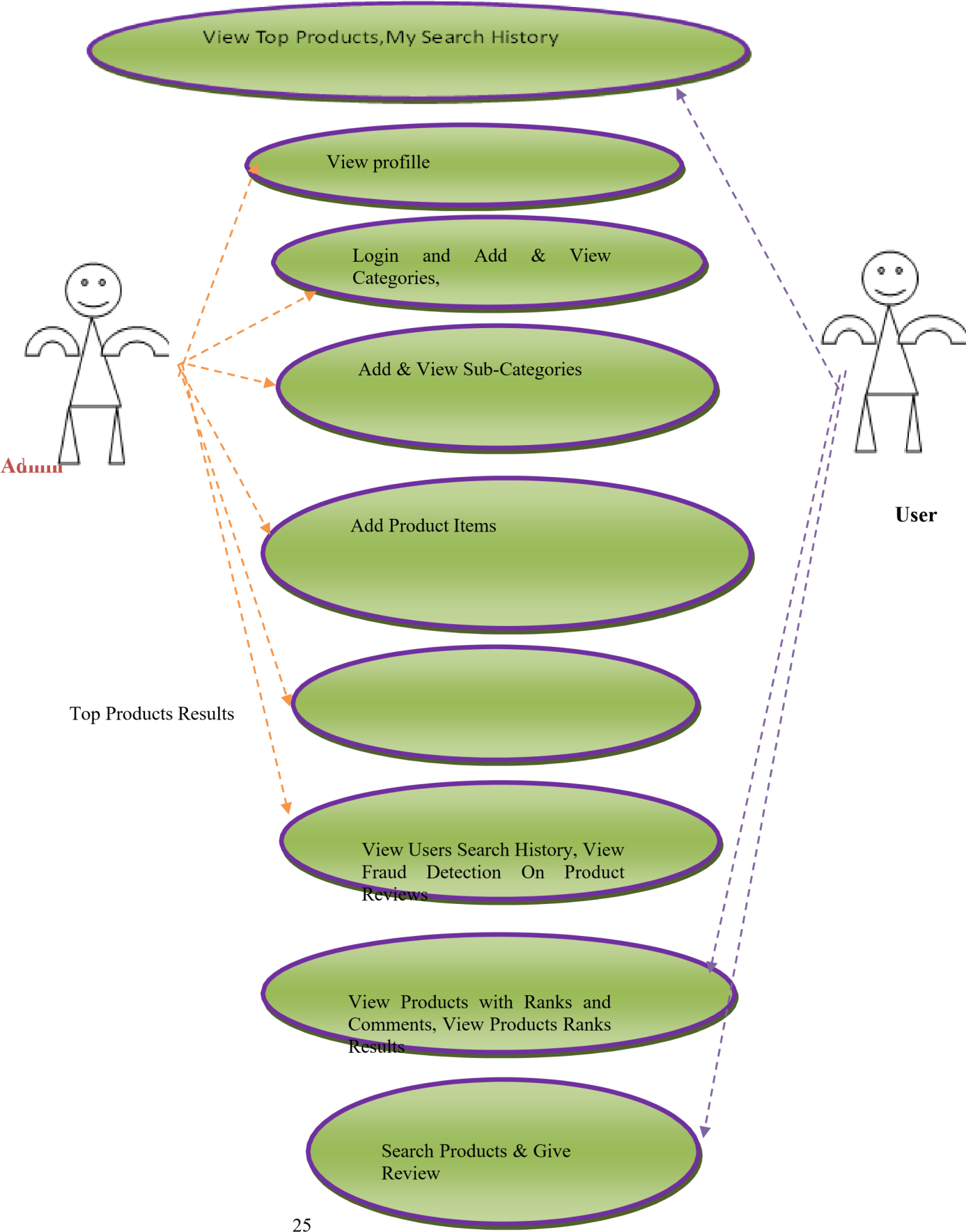
Results.

User

Web Server

**Sequence diagram**

**Use case diagram**



# Software Environment

##### Java Technology

Java technology is both a programming language and a platform.

The Java Programming Language

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

 Simple

 Architecture neutral

 Object oriented

 Portable

 Distributed

 High performance

 Interpreted

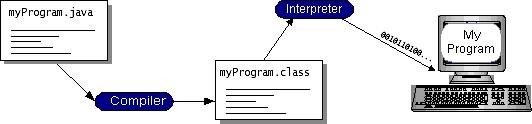
 Multithreaded

 Robust

 Dynamic

 Secure

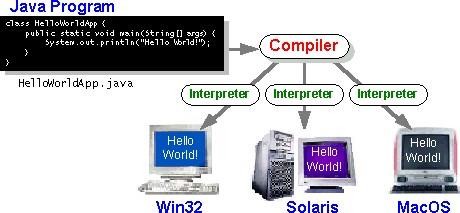
With most programming languages, you either compile or interpret a program so that you can run it on your computer. The Java programming language is unusual in that a program is both compiled and interpreted. With the compiler, first you translate a program into an intermediate language called *Java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



You can think of Java byte codes as the machine code instructions for the *Java Virtual Machine* (Java VM). Every Java interpreter, whether it’s a development tool or a Web browser that can run applets, is an implementation of the Java VM. Java byte codes help make “write once, run anywhere” possible. You can

compile your program into byte codes on any platform that has a Java compiler. The byte codes can then be

run on any implementation of the Java VM. That means that as long as a computer has a Java VM, the same program written in the Java programming language can run on Windows 2000, a Solaris workstation, or on an iMac.



The Java Platform

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

The Java platform has two components:

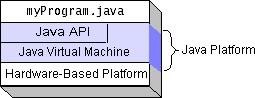
* The *Java Virtual Machine* (Java VM)
* The *Java Application Programming Interface* (Java API)

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

The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets. The Java API is grouped into libraries of related

classes and interfaces; these libraries are known as *packages*. The next section, What Can Java Technology Do? Highlights what functionality some of the packages in the Java API provide.

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



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

***What Can Java Technology Do?***

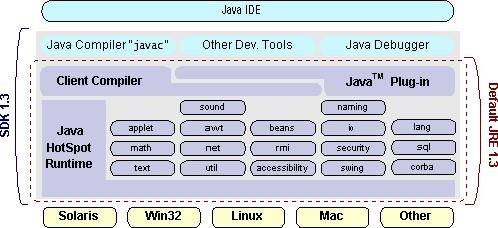
The most common types of programs written in the Java programming language are *applets* and *applications*. If you’ve surfed the Web, you’re probably already familiar with applets. An applet is a program that adheres to certain conventions that allow it to run within a Java-enabled browser.

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

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

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

* **The essentials**: Objects, strings, threads, numbers, input and output, data structures, system properties, date and time, and so on.
* **Applets**: The set of conventions used by applets.
* **Networking**: URLs, TCP (Transmission Control Protocol), UDP (User Data gram Protocol) sockets, and IP (Internet Protocol) addresses.
* **Internationalization**: Help for writing programs that can be localized for users worldwide. Programs can automatically adapt to specific locales and be displayed in the appropriate language.
* **Security**: Both low level and high level, including electronic signatures, public and private key management, access control, and certificates.
* **Software components**: Known as JavaBeansTM, can plug into existing component architectures.
* **Object serialization**: Allows lightweight persistence and communication via Remote Method Invocation (RMI).
* **Java Database Connectivity (JDBCTM)**: 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 JavaTM 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:

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.

1. 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.

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 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.

1. 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.

1. 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.

1. 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.

1. 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.

**Java Program**

**Compilers**

**My Program**

**Interpreter**

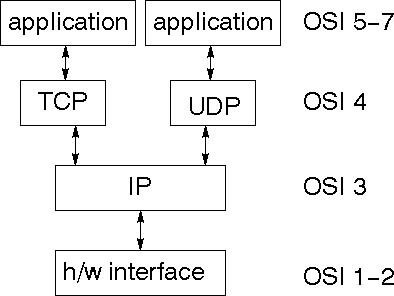
You can think of Java byte codes as the machine code instructions for the Java Virtual Machine (Java VM). Every Java interpreter, whether it’s a Java development tool or a Web browser that can run Java applets, is an implementation of the Java VM. The Java VM can also be implemented in hardware.

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.

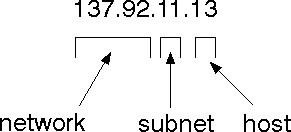
Subnet address

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.](http://www.gnu.org/philosophy/free-sw.html) [I](http://www.gnu.org/philosophy/free-sw.html)t is distributed under the terms of the [GNU](http://www.gnu.org/licenses/lgpl.html) [Lesser](http://www.gnu.org/licenses/lgpl.html) [General](http://www.gnu.org/licenses/lgpl.html) [Public](http://www.gnu.org/licenses/lgpl.html) [Licence](http://www.gnu.org/licenses/lgpl.html) (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.

1. 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.

1. 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.

1. Property Editors

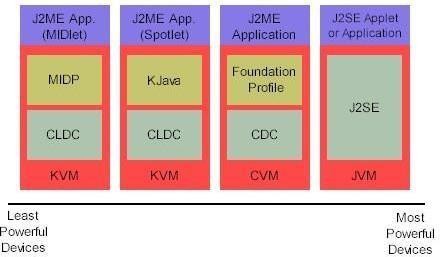
The property editor mechanism in JFreeChart only handles a small subset of the properties that can be set for charts. Extend (or reimplement) this mechanism to provide greater end-user control over the appearance of the charts.

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.

1. Developing J2ME applications

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

1. 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.

1. 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.

1. 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

### Client Server

##### Over view:

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

$200 million in 1992 to more than $1.2 billion in 1996.

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

The typical client either a PC or a Work Station connected through a network to a more powerful PC, Workstation, Midrange or Main Frames server usually capable of handling request from more than one client. However, with some configuration server may also act as client. A server may need to access other server in order to process the original client request.

The key client server idea is that client as user is essentially insulated from the physical location and formats of the data needs for their application. With the proper middleware, a client input from or report can transparently access and manipulate both local database on the client machine and remote databases on one or more servers. An added bonus is the client server opens the door to multi-vendor database access indulging heterogeneous table joins.

What is a Client Server

Two prominent systems in existence are client server and file server systems. It is essential to distinguish between client servers and file server systems. Both provide shared network access to data but the comparison dens there! The file server simply provides a remote disk drive that can be accessed by LAN applications on

a file by file basis. The client server offers full relational database services such as SQL-Access, Record modifying, Insert, Delete with full relational integrity backup/ restore performance for high volume of transactions, etc. the client server middleware provides a flexible interface between client and server, who does what, when and to whom.

Why Client Server

Client server has evolved to solve a problem that has been around since the earliest days of computing: how best to distribute your computing, data generation and data storage resources in order to obtain efficient, cost effective departmental an enterprise wide data processing. During mainframe era choices were quite limited. A central machine housed both the CPU and DATA (cards, tapes, drums and later disks). Access to these resources was initially confined to batched runs that produced departmental reports at the appropriate intervals. A strong central information service department ruled the corporation. The role of the rest of the corporation limited to requesting new or more frequent reports and to provide hand written forms from which the central data banks were created and updated. The earliest client server solutions therefore could best be characterized as “SLAVE-MASTER”.

##### Front end or User Interface Design

The entire user interface is planned to be developed in browser specific environment with a touch of Intranet- Based Architecture for achieving the Distributed Concept.

The browser specific components are designed by using the HTML standards, and the dynamism of the designed by concentrating on the constructs of the Java Server Pages.

Communication or Database Connectivity Tier

The Communication architecture is designed by concentrating on the Standards of Servlets and Enterprise Java Beans. The database connectivity is established by using the Java Data Base Connectivity.

The standards of three-tire architecture are given major concentration to keep the standards of higher cohesion and limited coupling for effectiveness of the operations.

Features of The Language Used

In my project, I have chosen *Java* language for developing the code.

About Java

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

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

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

Importance of Java to the Internet

Java has had a profound effect on the Internet. This is because; Java expands the Universe of objects that can move about freely in Cyberspace. In a network, two categories of objects are transmitted between the Server

and the Personal computer. They are: Passive information and Dynamic active programs. The Dynamic, Self- executing programs cause serious problems in the areas of Security and probability. But, Java addresses those concerns and by doing so, has opened the door to an exciting new form of program called the Applet.

Java can be used to create two types of programs

**Applications and Applets:** An application is a program that runs on our Computer under the operating system of that computer. It is more or less like one creating using C or C++. Java’s ability to create Applets makes it important. An Applet is an application designed to be transmitted over the Internet and executed by a Java – compatible web browser. An applet is actually a tiny Java program, dynamically downloaded across the network, just like an image. But the difference is, it is an intelligent program, not just a media file. It can react to the user input and dynamically change.

Features Of Java Security

Every time you that you download a “normal” program, you are risking a viral infection. Prior to Java, most users did not download executable programs frequently, and those who did scanned them for viruses prior to execution. Most users still worried about the possibility of infecting their systems with a virus. In addition, another type of malicious program exists that must be guarded against. This type of program can gather private information, such as credit card numbers, bank account balances, and passwords. Java answers both these concerns by providing a “firewall” between a network application and your computer.

When you use a Java-compatible Web browser, you can safely download Java applets without fear of virus infection or malicious intent.

Portability

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

The key that allows the Java to solve the security and portability problems is that the output of Java compiler is Byte code. Byte code is a highly optimized set of instructions designed to be executed by the Java run-time system, which is called the Java Virtual Machine (JVM). That is, in its standard form, the JVM is an interpreter for byte code.

Translating a Java program into byte code helps makes it much easier to run a program in a wide variety of environments. The reason is, once the run-time package exists for a given system, any Java program can run on it.

Although Java was designed for interpretation, there is technically nothing about Java that prevents on-the- fly compilation of byte code into native code. Sun has just completed its Just In Time (JIT) compiler for byte code. When the JIT compiler is a part of JVM, it compiles byte code into executable code in real time, on a piece-by-piece, demand basis. It is not possible to compile an entire Java program into executable code all at once, because Java performs various run-time checks that can be done only at run time. The JIT compiles code, as it is needed, during execution.

Java, Virtual Machine (JVM)

Beyond the language, there is the Java virtual machine. The Java virtual machine is an important element of the Java technology. The virtual machine can be embedded within a web browser or an operating system. Once a piece of Java code is loaded onto a machine, it is verified. As part of the loading process, a class loader is invoked and does byte code verification makes sure that the code that’s has been generated by the compiler will not corrupt the machine that it’s loaded on. Byte code verification takes place at the end of the compilation process to make sure that is all accurate and correct. So byte code verification is integral to the compiling and executing of Java code.

Overall **Description**

Java .Class

Picture showing the development process of JAVA Program

Java programming uses to produce byte codes and executes them. The first box indicates that the Java source code is located in a. Java file that is processed with a Java compiler called javac. The Java compiler produces a file called a. class file, which contains the byte code. The. Class file is then loaded across the network or loaded locally on your machine into the execution environment is the Java virtual machine, which interprets and executes the byte code.

Java Architecture

Java architecture provides a portable, robust, high performing environment for development. Java provides portability by compiling the byte codes for the Java Virtual Machine, which is then interpreted on each

platform by the run-time environment. Java is a dynamic system, able to load code when needed from a machine in the same room or across the planet.

Compilation of code

When you compile the code, the Java compiler creates machine code (called byte code) for a hypothetical machine called Java Virtual Machine (JVM). The JVM is supposed to execute the byte code. The JVM is created for overcoming the issue of portability. The code is written and compiled for one machine and interpreted on all machines. This machine is called Java Virtual Machine.

**PC Compiler**

**Java**

**Source Code**

**………..**

**………..**

**Byte code**

**………..**

**(Platform indepen dent)**

**Java**

**Interpreter (Sparc )**

**SPARC**

**Java**

**Interpreter (Macintosh )**

**Java**

**Interpreter (PC)**

**Macintosh Compiler**

During run-time the Java interpreter tricks the byte code file into thinking that it is running on a Java Virtual Machine. In reality this could be a Intel Pentium Windows 95 or Sun SARC station running Solaris or Apple Macintosh running system and all could receive code from any computer through Internet and run the Applets. **Simple**

Java was designed to be easy for the Professional programmer to learn and to use effectively. If you are an experienced C++ programmer, learning Java will be even easier. Because Java inherits the C/C++ syntax and many of the object oriented features of C++. Most of the confusing concepts from C++ are either left out of Java or implemented in a cleaner, more approachable manner. In Java there are a small number of clearly defined ways to accomplish a given task.

Object-Oriented

Java was not designed to be source-code compatible with any other language. This allowed the Java team the freedom to design with a blank slate. One outcome of this was a clean usable, pragmatic approach to objects. The object model in Java is simple and easy to extend, while simple types, such as integers, are kept as highperformance non-objects.

Robust

The multi-platform environment of the Web places extraordinary demands on a program, because the program must execute reliably in a variety of systems. The ability to create robust programs was given a high priority in the design of Java. Java is strictly typed language; it checks your code at compile time and run time.

Java virtually eliminates the problems of memory management and deallocation, which is completely automatic. In a well-written Java program, all run time errors can –and should –be managed by your program. **JAVASCRIPT**

JavaScript is a script-based programming language that was developed by Netscape Communication Corporation. JavaScript was originally called Live Script and renamed as JavaScript to indicate its relationship with Java. JavaScript supports the development of both client and server components of Web-based applications. On the client side, it can be used to write programs that are executed by a Web browser within the context of a Web page. On the server side, it can be used to write Web server programs that can process information submitted by a Web browser and then updates the browser’s display accordingly

Even though JavaScript supports both client and server Web programming, we prefer JavaScript at Client side programming since most of the browsers supports it. JavaScript is almost as easy to learn as HTML, and JavaScript statements can be included in HTML documents by enclosing the statements between a pair of scripting tags

<SCRIPTS>..</SCRIPT>.

<SCRIPT LANGUAGE = “JavaScript”>

JavaScript statements

</SCRIPT>

Here are a few things we can do with JavaScript :

* Validate the contents of a form and make calculations.
* Add scrolling or changing messages to the Browser’s status line.
* Animate images or rotate images that change when we move the mouse over them.
* Detect the browser in use and display different content for different browsers.
* Detect installed plug-ins and notify the user if a plug-in is required. We can do much more with JavaScript, including creating entire application.

JavaScript and Java are entirely different languages. A few of the most glaring differences are:

* Java applets are generally displayed in a box within the web document; JavaScript can affect any part of the Web document itself.
* While JavaScript is best suited to simple applications and adding interactive features to Web pages; Java can be used for incredibly complex applications.

There are many other differences but the important thing to remember is that JavaScript and Java are separate languages. They are both useful for different things; in fact they can be used together to combine their advantages.

* JavaScript can be used for Sever-side and Client-side scripting.
* It is more flexible than VBScript.
* JavaScript is the default scripting languages at Client-side since all the browsers supports it.

Hyper Text Markup Language

Hypertext Markup Language (HTML), the languages of the World Wide Web (WWW), allows users to produces Web pages that include text, graphics and pointer to other Web pages (Hyperlinks).

HTML is not a programming language but it is an application of ISO Standard 8879, SGML (Standard Generalized Markup Language), but specialized to hypertext and adapted to the Web. The idea behind Hypertext is that instead of reading text in rigid linear structure, we can easily jump from one point to another point. We can navigate through the information based on our interest and preference. A markup language is simply a series of elements, each delimited with special characters that define how text or other items enclosed within the elements should be displayed. Hyperlinks are underlined or emphasized works that load to other documents or some portions of the same document.

HTML can be used to display any type of document on the host computer, which can be geographically at a different location. It is a versatile language and can be used on any platform or desktop.

HTML provides tags (special codes) to make the document look attractive. HTML tags are not case-sensitive. Using graphics, fonts, different sizes, color, etc., can enhance the presentation of the document. Anything that is not a tag is part of the document itself.

Basic HTML Tags :

|  |  |
| --- | --- |
| **<!-- -->** | Specifies comments |
| **<A>** **</A>** | Creates hypertext links |
| **<B>** **</B>** | Formats text as bold |
| **<BIG>** **</BIG>** | Formats text in large font. |
| **<BODY>…</BODY>** | Contains all tags and text in the HTML document |
| **<CENTER>...</CENTER>** | Creates text |
| **<DD>…</DD>** | Definition of a term |
| **<DL>...</DL>** | Creates definition list |
| **<FONT>…</FONT>** | Formats text with a particular font |
| **<FORM>...</FORM>** | Encloses a fill-out form |
| **<FRAME>...</FRAME>** | Defines a particular frame in a set of frames |
| **<H#>…</H#>** | Creates headings of different levels |
| **<HEAD>...</HEAD>** | Contains tags that specify information about a document |
| **<HR>...</HR>** | Creates a horizontal rule |
| **<HTML>…</HTML>** | Contains all other HTML tags |
| **<META>...</META>** | Provides meta-information about a document |
| **<SCRIPT>…</SCRIPT>** | Contains client-side or server-side script |
| **<TABLE>…</TABLE>** | Creates a table |
| **<TD>…</TD>** | Indicates table data in a table |
| **<TR>…</TR>** | Designates a table row |
| **<TH>…</TH>** | Creates a heading in a table |

ADVANTAGES

* A HTML document is small and hence easy to send over the net. It is small because it does not include formatted information.
* HTML is platform independent.
* HTML tags are not case-sensitive.

Java Database Connectivity

What Is JDBC?

JDBC is a Java API for executing SQL statements. (As a point of interest, JDBC is a trademarked name and is not an acronym; nevertheless, JDBC is often thought of as standing for Java Database Connectivity. It

consists of a set of classes and interfaces written in the Java programming language. JDBC provides a standard API for tool/database developers and makes it possible to write database applications using a pure Java API. Using JDBC, it is easy to send SQL statements to virtually any relational database. One can write a single program using the JDBC API, and the program will be able to send SQL statements to the appropriate database. The combinations of Java and JDBC lets a programmer write it once and run it anywhere.

What Does JDBC Do?

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

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

JDBC versus ODBC and other APIs

At this point, Microsoft's ODBC (Open Database Connectivity) API is that probably the most widely used programming interface for accessing relational databases. It offers the ability to connect to almost all databases on almost all platforms.

So why not just use ODBC from Java? The answer is that you can use ODBC from Java, but this is best done with the help of JDBC in the form of the JDBC-ODBC Bridge, which we will cover shortly. The question now becomes "Why do you need JDBC?" There are several answers to this question:

1. ODBC is not appropriate for direct use from Java because it uses a C interface. Calls from Java to native C code have a number of drawbacks in the security, implementation, robustness, and automatic portability of applications.
2. A literal translation of the ODBC C API into a Java API would not be desirable. For example, Java has no pointers, and ODBC makes copious use of them, including the notoriously error-prone generic pointer "void \*". You can think of JDBC as ODBC translated into an object-oriented interface that is natural for Java programmers.
3. ODBC is hard to learn. It mixes simple and advanced features together, and it has complex options even for simple queries. JDBC, on the other hand, was designed to keep simple things simple while allowing more advanced capabilities where required.
4. A Java API like JDBC is needed in order to enable a "pure Java" solution. When ODBC is used, the ODBC driver manager and drivers must be manually installed on every client machine. When the JDBC driver is written completely in Java, however, JDBC code is automatically installable, portable, and secure on all Java platforms from network computers to mainframes.

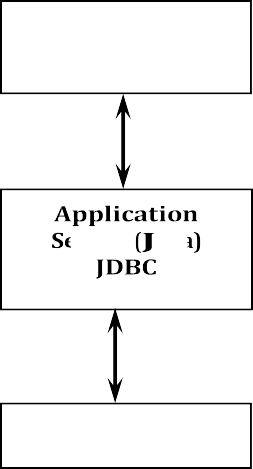
Two-tier and Three-tier Models

The JDBC API supports both two-tier and three-tier models for database access.

In the two-tier model, a Java applet or application talks directly to the database. This requires a JDBC driver that can communicate with the particular database management system being accessed. A user's SQL statements are delivered to the database, and the results of those statements are sent back to the user. The database may be located on another machine to which the user is connected via a network. This is referred to as a client/server configuration, with the user's machine as the client, and the machine housing the database as the server. The network can be an Intranet, which, for example, connects employees within a corporation, or it can be the Internet.

In the three-tier model, commands are sent to a "middle tier" of services, which then send SQL statements to the database. The database processes the SQL statements and sends the results back to the middle tier, which then sends them to the user. MIS directors find the three-tier model very attractive because the middle tier makes it possible to maintain control over access and the kinds of updates that can be made to corporate data.

**Client machine (GUI)**



**Java applet or Html browser**

**HTTP, RMI, or CORBA calls**

**Server machine (busine~~ss L~~og~~ic)~~**

**Database server**

Another advantage is that when there is a middle tier, the user can employ an easy-to-use higher-level API which is translated by the middle tier into the appropriate low-level calls. Finally, in many cases the three-tier architecture can provide performance advantages.

Until now the middle tier has typically been written in languages such as C or C++, which offer fast performance. However, with the introduction of optimizing compilers that translate Java byte code into efficient machine-specific code, it is becoming practical to implement the middle tier in Java. This is a big plus, making it possible to take advantage of Java's robustness, multithreading, and security features.

JDBC is important to allow database access from a Java middle tier.

JDBC Driver Types

The JDBC drivers that we are aware of at this time fit into one of four categories:

* JDBC-ODBC bridge plus ODBC driver
* Native-API partly-Java driver
* JDBC-Net pure Java driver  Native-protocol pure Java driver

JDBC-ODBC Bridge

If possible, use a Pure Java JDBC driver instead of the Bridge and an ODBC driver. This completely eliminates the client configuration required by ODBC. It also eliminates the potential that the Java VM could be corrupted by an error in the native code brought in by the Bridge (that is, the Bridge native library, the ODBC driver manager library, the ODBC driver library, and the database client library).

What Is the JDBC- ODBC Bridge?

The JDBC-ODBC Bridge is a JDBC driver, which implements JDBC operations by translating them into ODBC operations. To ODBC it appears as a normal application program. The Bridge implements JDBC for any database for which an ODBC driver is available. The Bridge is implemented as the

sun.jdbc.odbc Java package and contains a native library used to access ODBC. The Bridge is a joint development of Intersolv and JavaSoft.

Java Server Pages (JSP*)*

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

Features of JSP Portability:

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

Components

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

Processing

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

Access Models*:*

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

In both of the above cases, the page could also contain any valid Java code. Java Server Pages architecture encourages separation of content from presentation.

Steps in the execution of a JSP Application:

1. The client sends a request to the web server for a JSP file by giving the name of the JSP file within the form tag of a HTML page.
2. This request is transferred to the JavaWebServer. At the server side JavaWebServer receives the request and if it is a request for a jsp file server gives this request to the JSP engine.
3. JSP engine is program which can understands the tags of the jsp and then it converts those tags into a Servlet program and it is stored at the server side. This Servlet is loaded in the memory and then it is executed and the result is given back to the JavaWebServer and then it is transferred back to the result is given back to the JavaWebServer and then it is transferred back to the client.

JDBC connectivity

The JDBC provides database-independent connectivity between the J2EE platform and a wide range of tabular data sources. JDBC technology allows an Application Component Provider to:

* + Perform connection and authentication to a database server
  + Manager transactions
  + Move SQL statements to a database engine for preprocessing and execution
  + Execute stored procedures
  + Inspect and modify the results from Select statements.

### Tomcat 6.0 web server

Tomcat is an open source web server developed by Apache Group. Apache Tomcat is the servlet container that is used in the official Reference Implementation for the Java Servlet and Java Server Pages technologies. The Java Servlet and Java Server Pages specifications are developed by Sun under the Java Community Process. Web Servers like Apache Tomcat support only web components while an application server supports web components as well as business components (BEAs Weblogic, is one of the popular application server).To develop a web application with jsp/servlet install any web server like JRun, Tomcat etc to run your application.



# SYSTEM TESTING

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

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

#### TYPES OF TESTS

Unit testing

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

Integration testing

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

Functional test

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

Functional testing is centered on the following items:

|  |  |
| --- | --- |
| Valid Input | : identified classes of valid input must be accepted. |
| Invalid Input | : identified classes of invalid input must be rejected. |
| Functions | : identified functions must be exercised. |

Output : identified classes of application outputs must be exercised. Systems/Procedures: interfacing systems or procedures must be invoked.

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

##### System Test

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

##### White Box Testing

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

##### Black Box Testing

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

##### Unit Testing

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

##### Test strategy and approach

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

##### Test objectives

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

##### Features to be tested

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

##### Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

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

##### Acceptance Testing

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

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

# SYSTEM TESTING

#### TESTING METHODOLOGIES

The following are the Testing Methodologies:

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

##### 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. o Drivers are removed and clusters are combined moving upward in the program structure

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

## OTHER TESTING METHODOLOGIES

##### User Acceptance Testing

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

##### Output Testing

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

##### Validation Checking

Validation checks are performed on the following fields.

##### Text Field:

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

**Numeric Field:**

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

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

**Preparation of Test Data**

Taking various kinds of test data does the above testing. Preparation of test data plays a vital role in the system testing. After preparing the test data the system under study is tested using that test data. While

testing the system by using test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

**Using Live Test Data:**

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

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

**Using Artificial Test Data:**

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

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

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

###### USER TRAINING

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

###### MAINTAINENCE

This covers a wide range of activities including correcting code and design errors. To reduce the need for maintenance in the long run, we have more accurately defined the user’s requirements during the process of system development. Depending on the requirements, this system has been developed to satisfy the needs to the largest possible

extent. With development in technology, it may be possible to add many more features based on the requirements in future. The coding and designing is simple and easy to understand which will make maintenance easier.

###### TESTING STRATEGY :

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

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

###### SYSTEM TESTING:

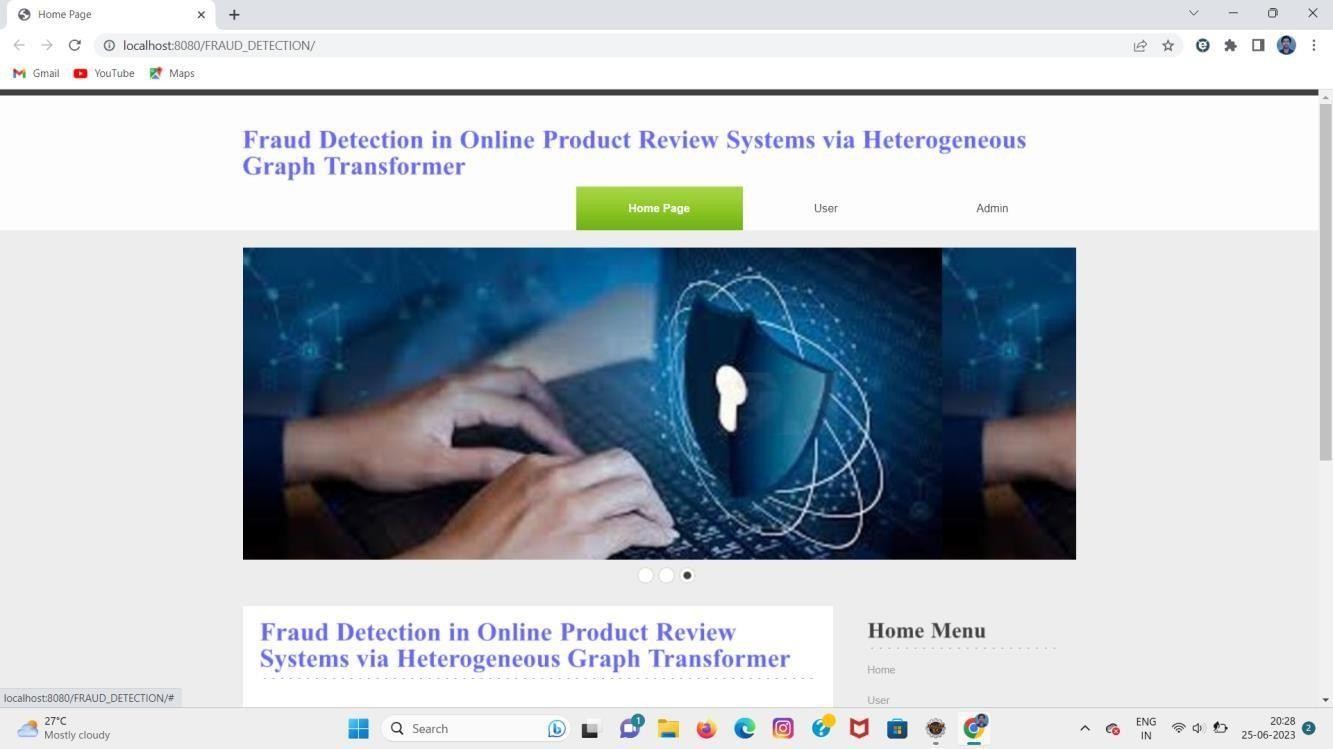
Software once validated must be combined with other system elements (e.g. Hardware, people, database). System testing verifies that all the elements are proper and that overall system function performance is achieved. It also tests to find discrepancies between the system and its original objective,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

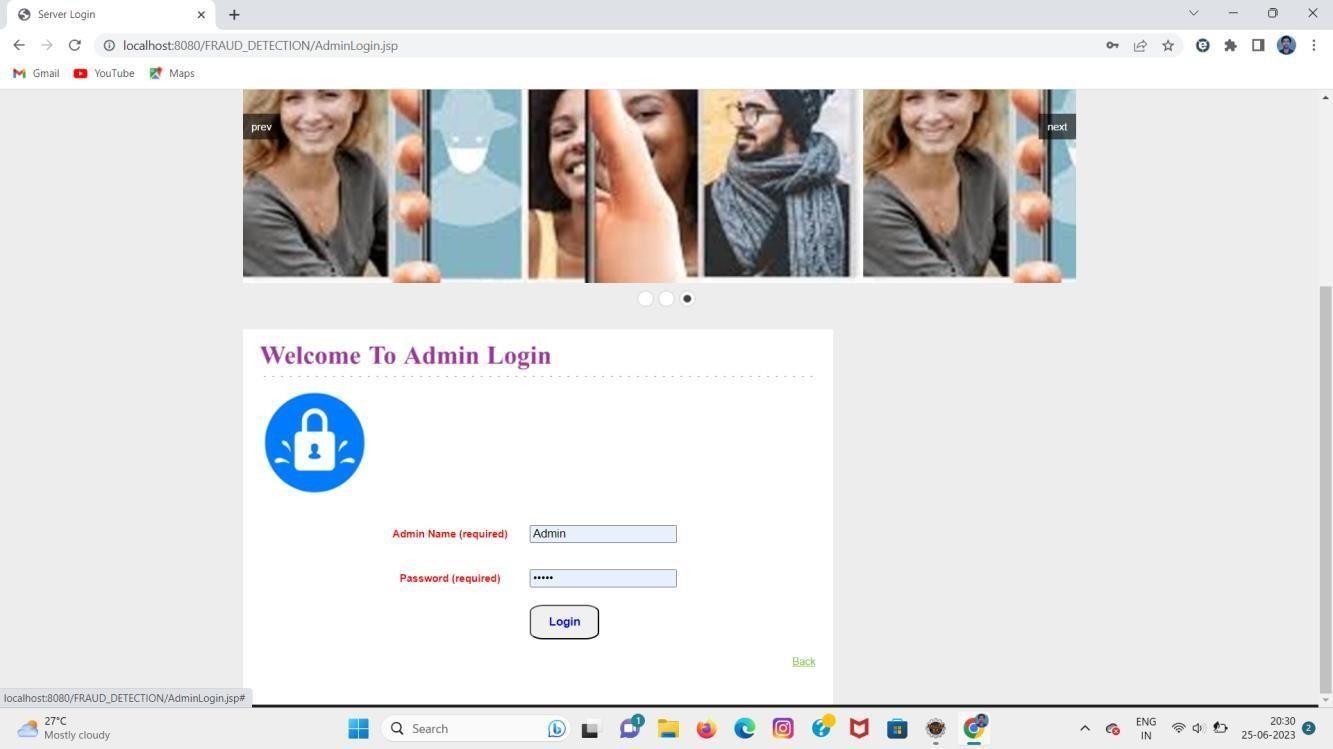
###### UNIT TESTING:

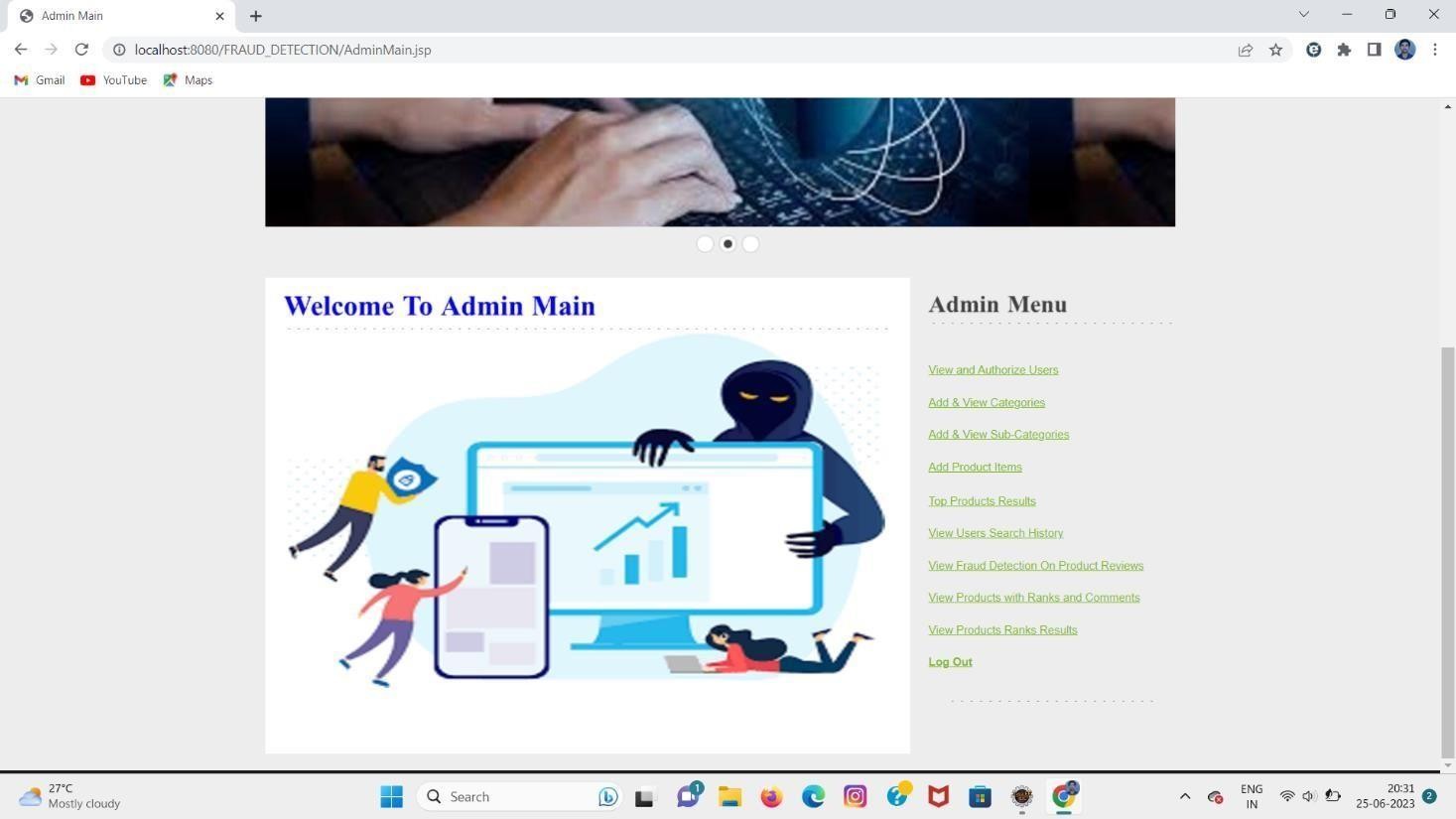
In unit testing different are modules are tested against the specifications produced during the design for the modules. Unit testing is essential for verification of the code produced during the coding phase, and hence the goals to test the internal logic of the modules. Using the detailed design description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

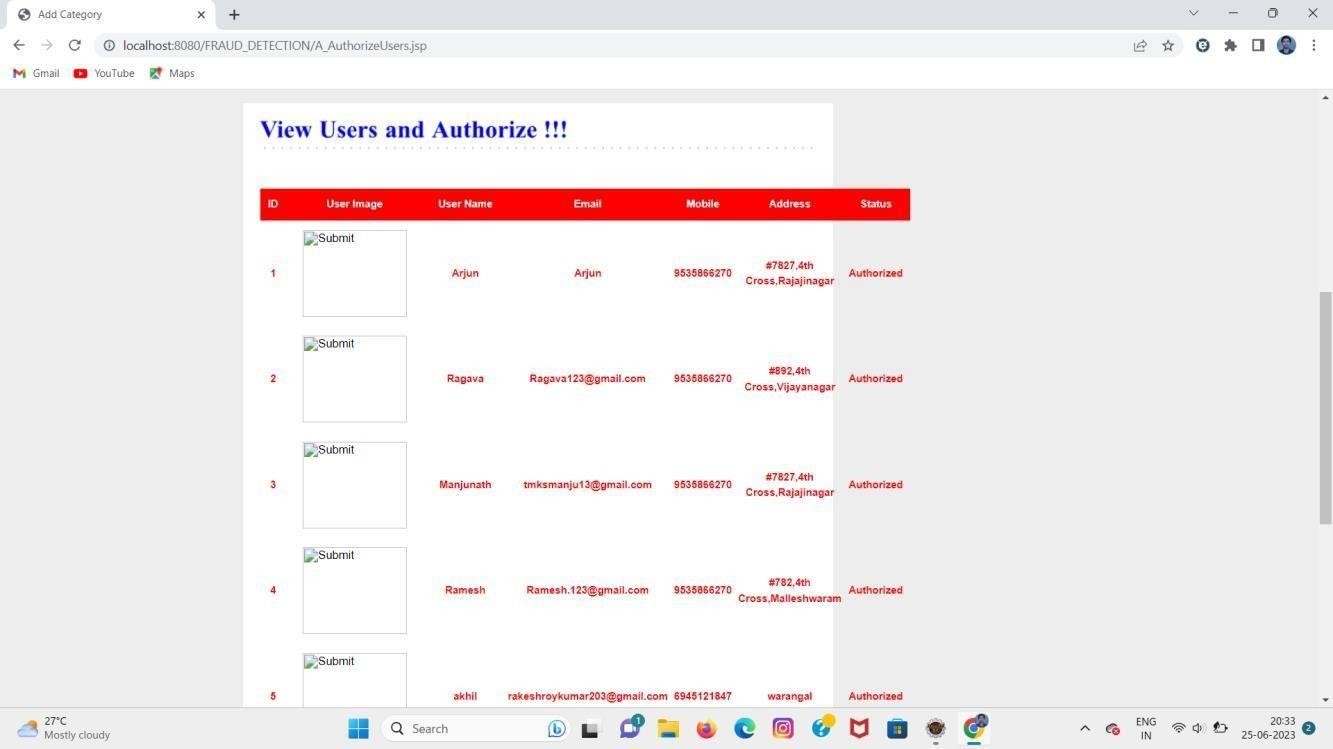
In Due Course, latest technology advancements will be taken into consideration. As part of technical build-up many components of the networking system will be generic in nature so that future projects can either use or interact with this. The future holds a lot to offer to the development and refinement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

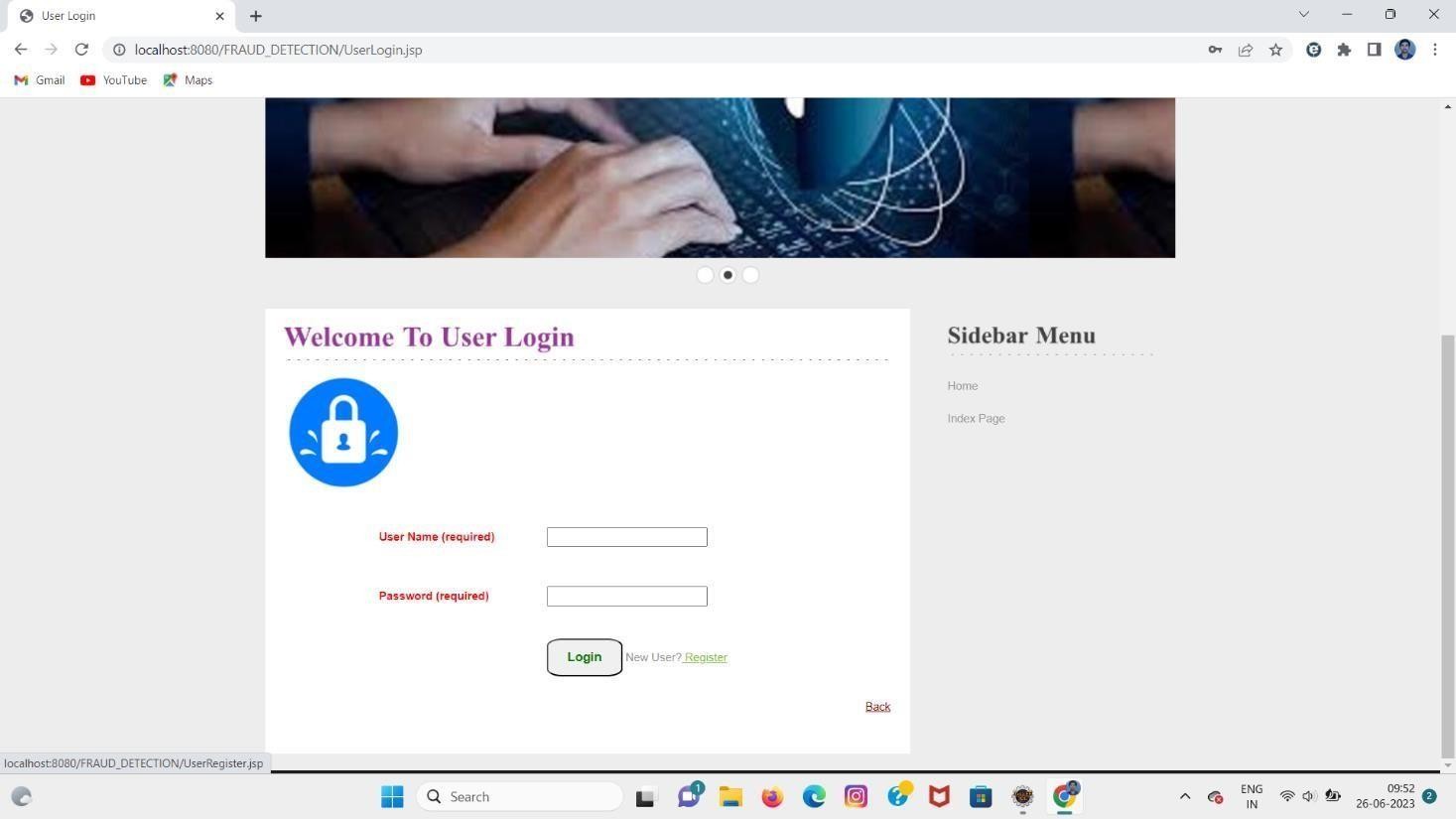
# RESULTS

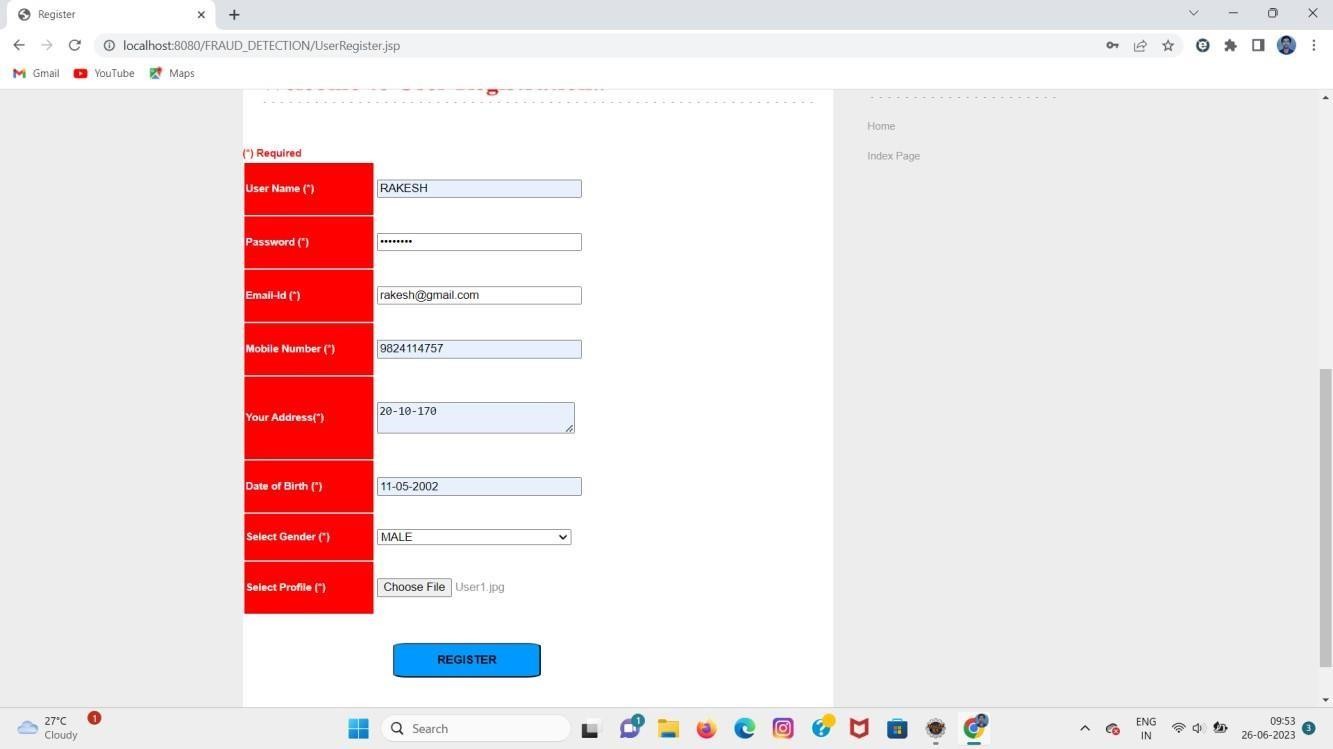


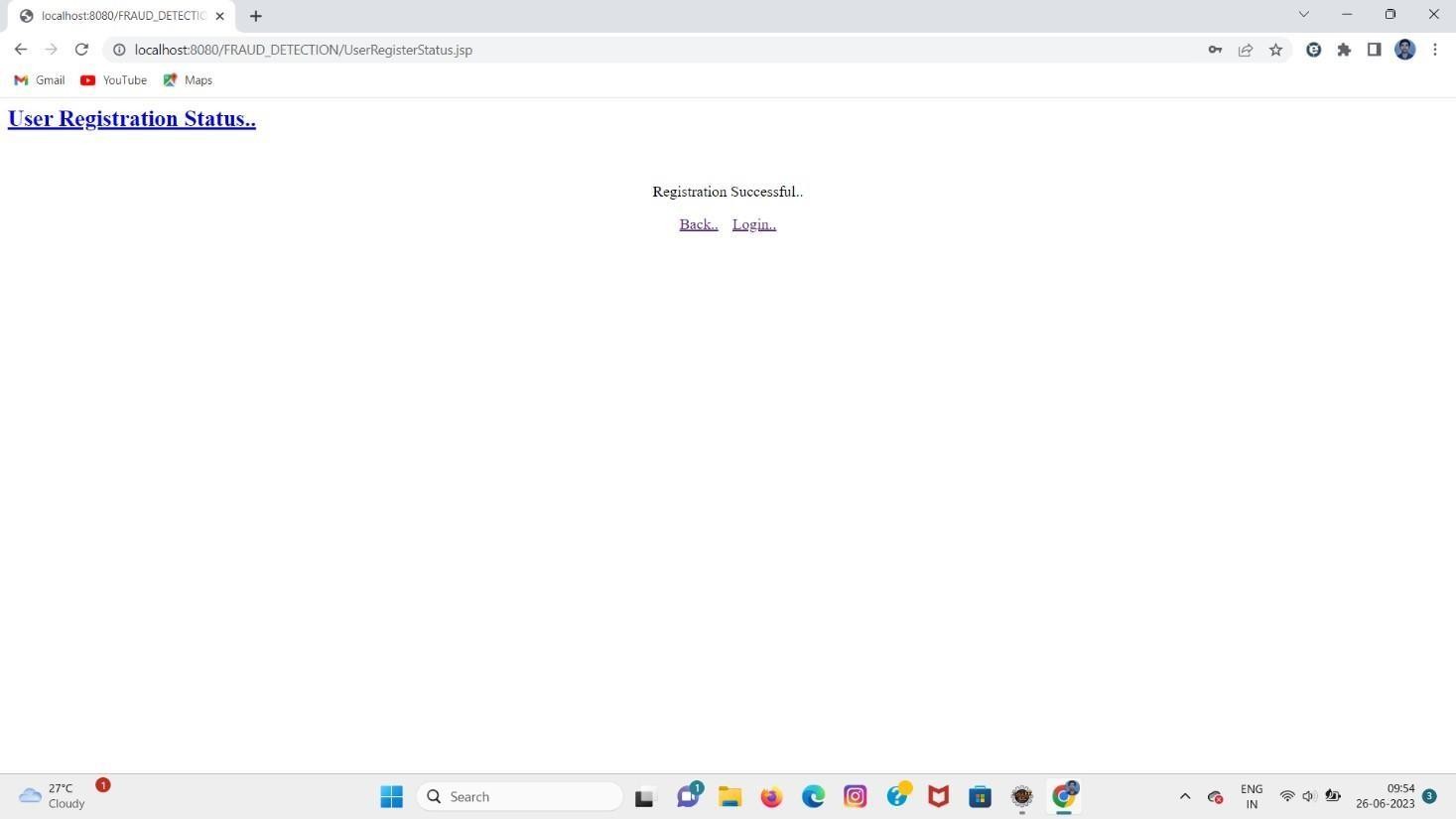


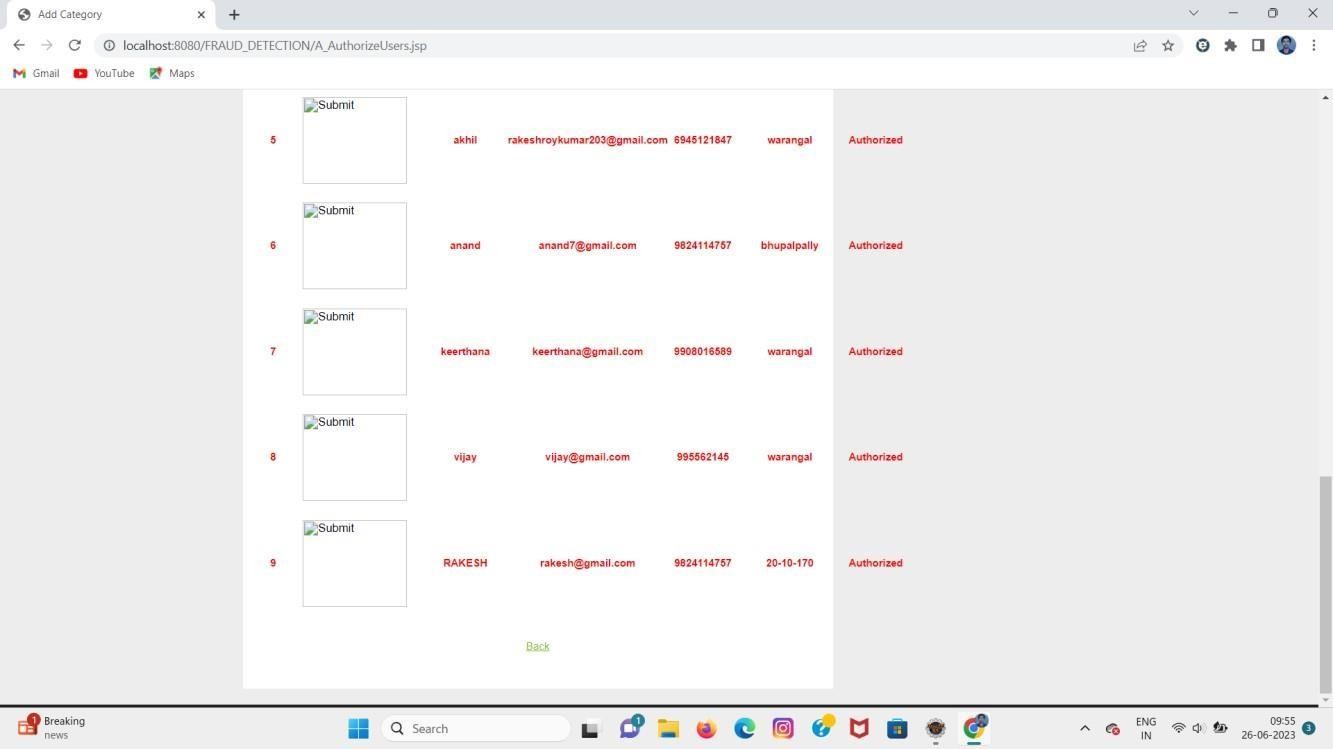


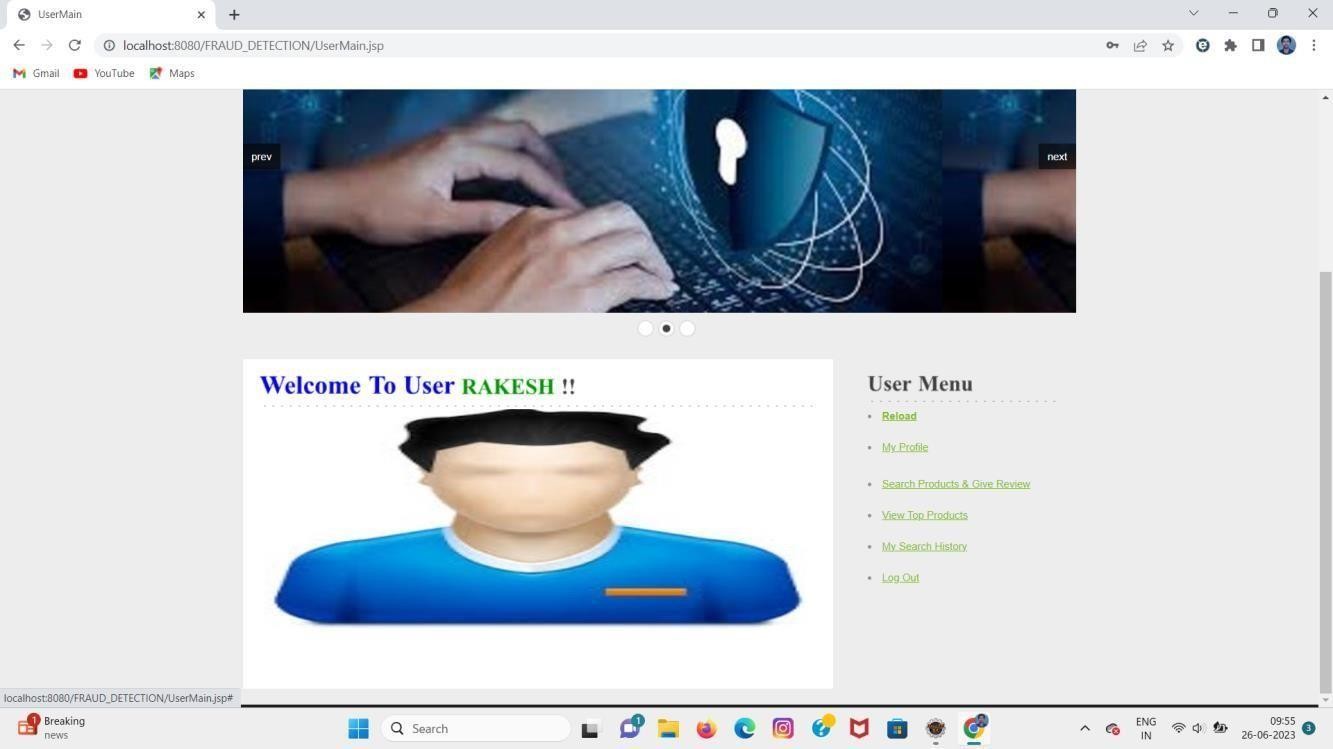


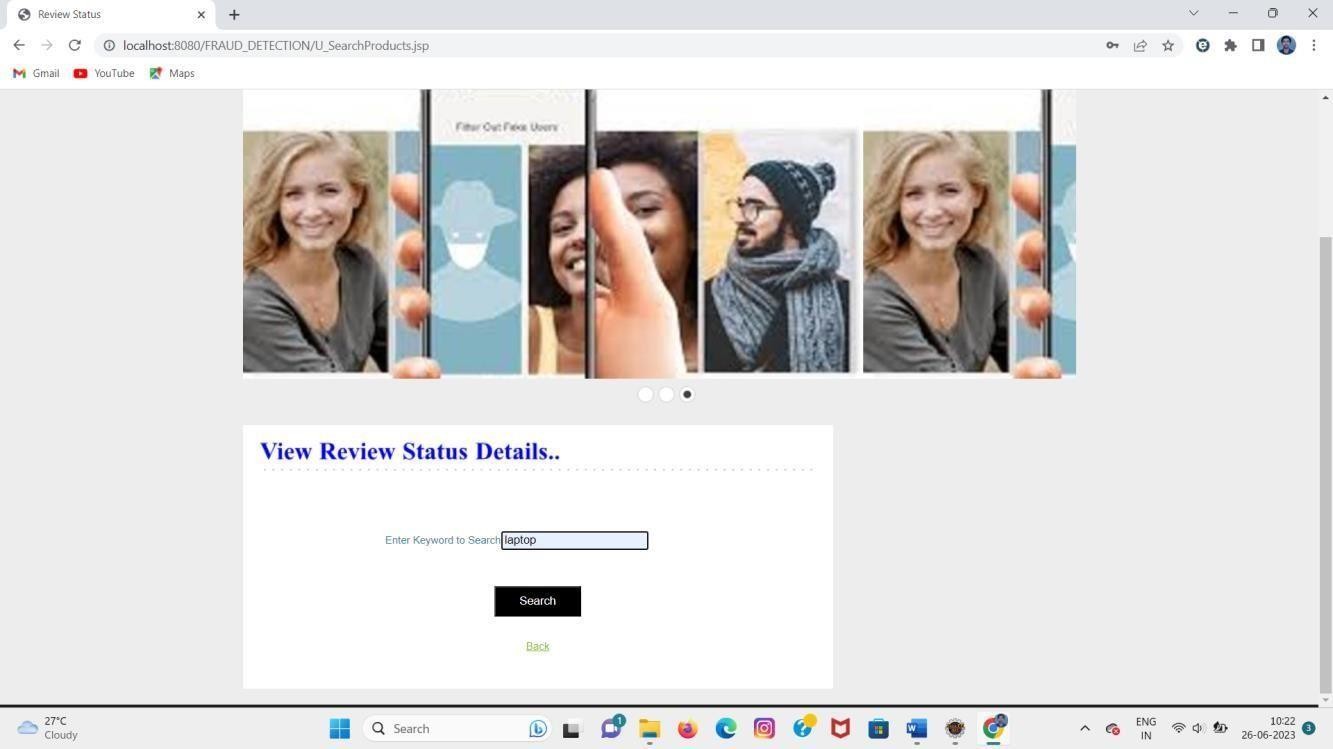


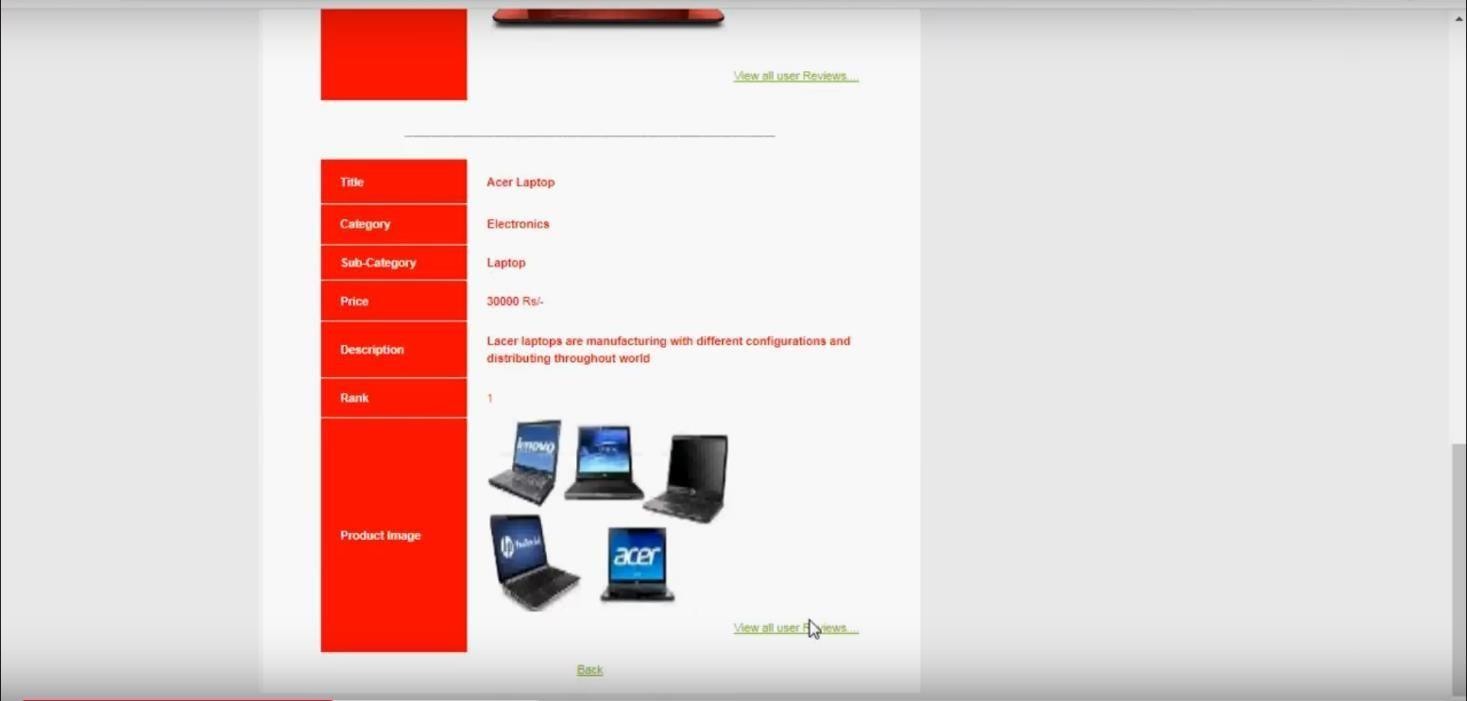


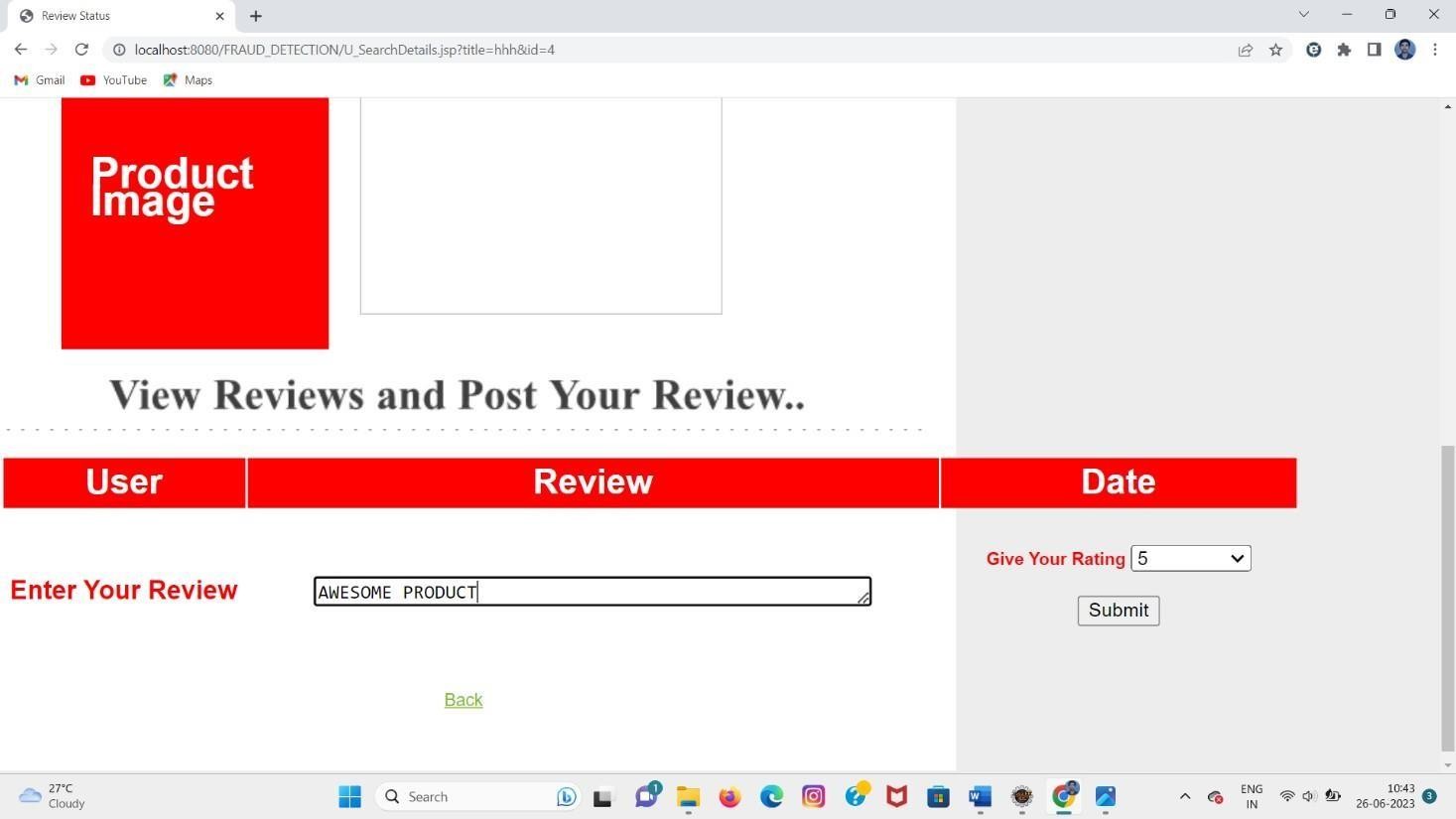


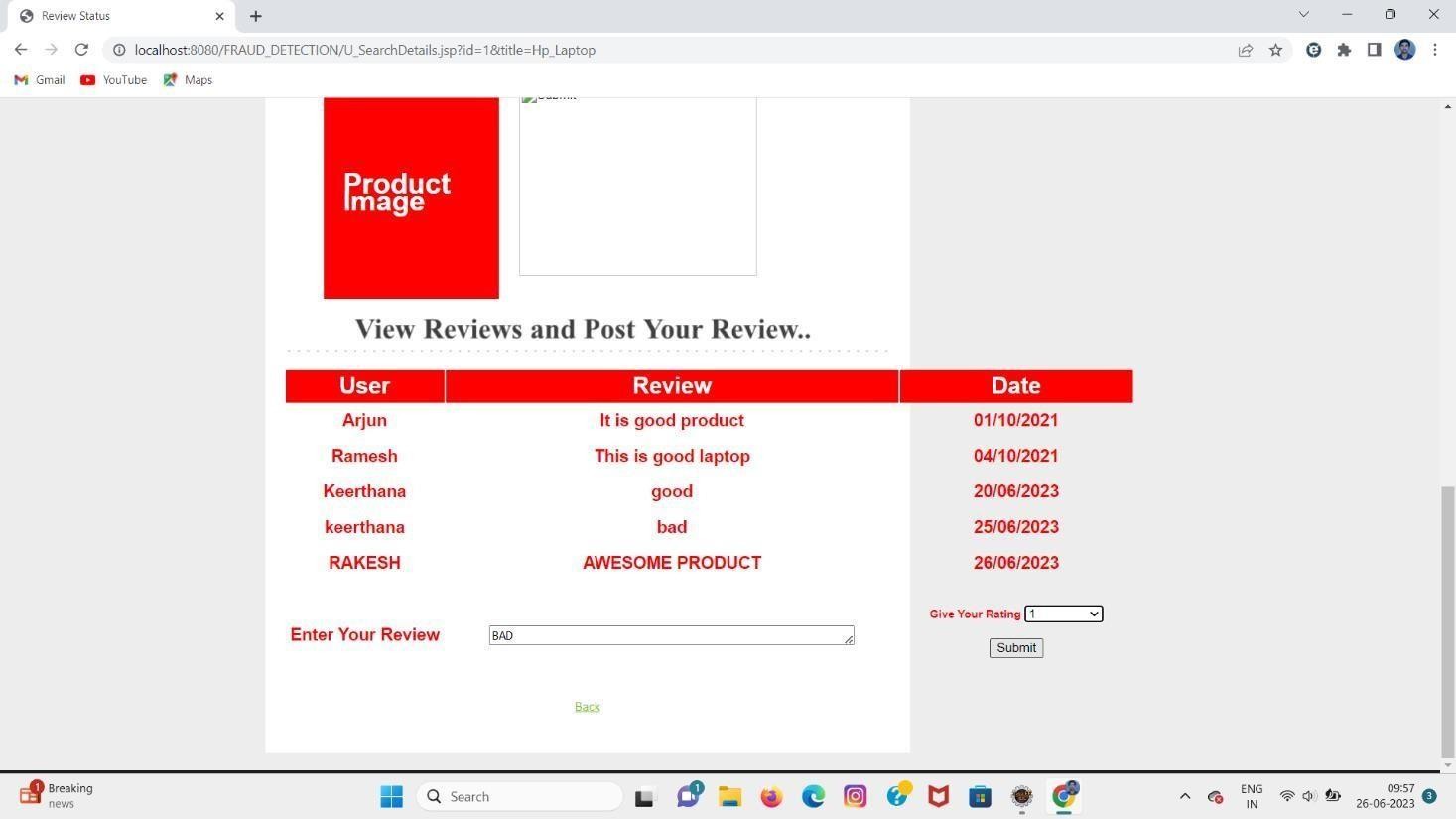
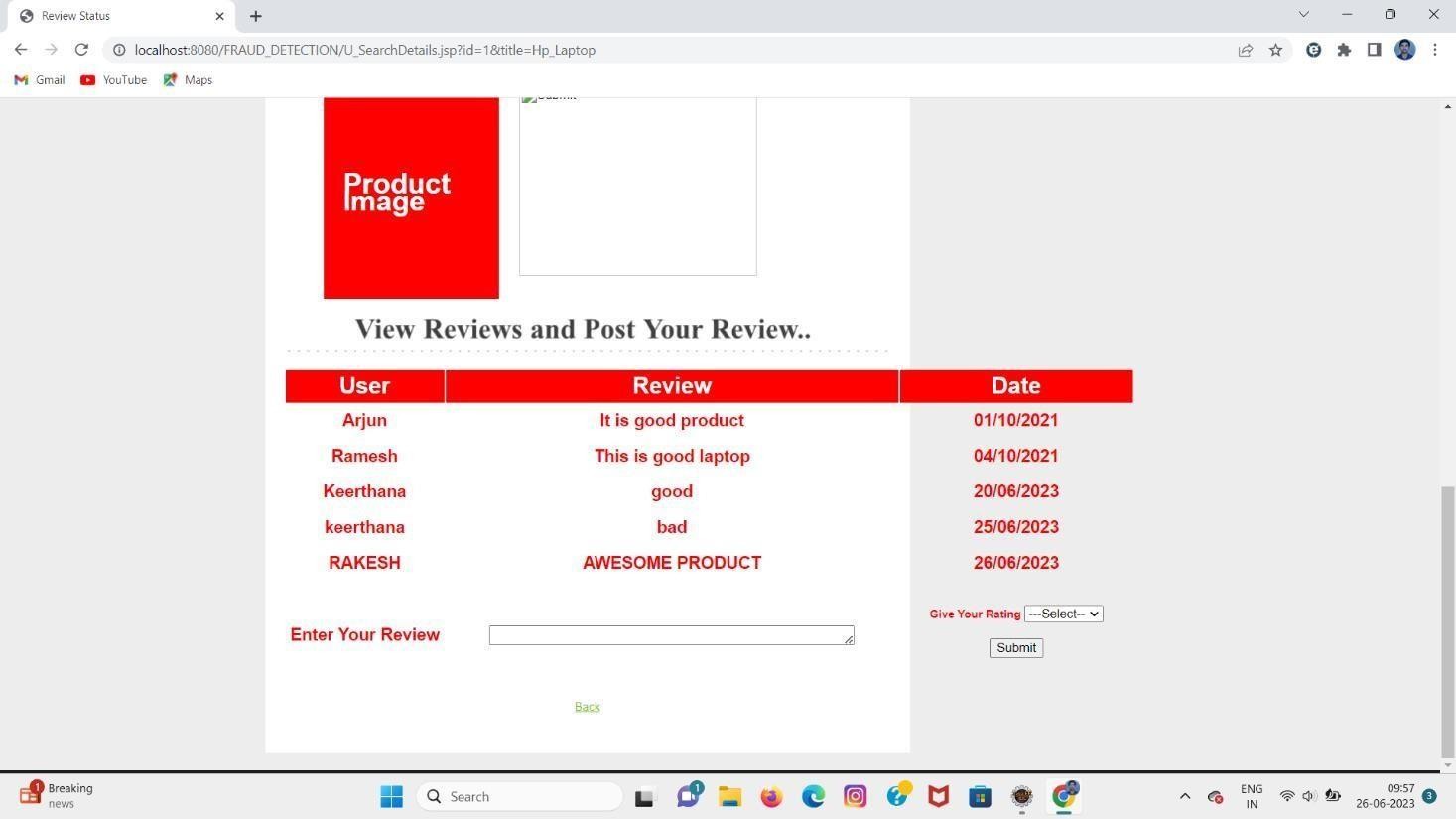


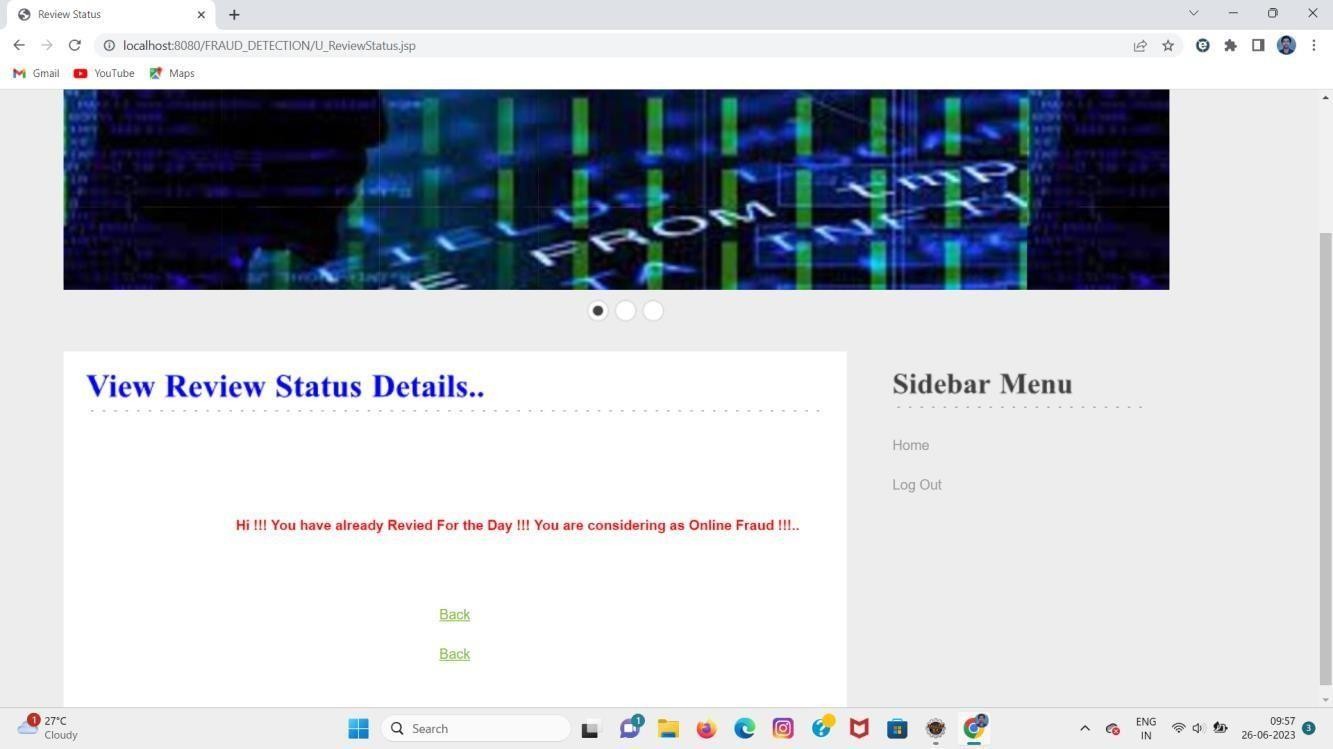












**CONCLUSION**

In this paper, we propose FAHGT, a novel heterogeneous graph neural network for fraudulent user detection in online review systems. To handle inconsistent features, we adopt heterogeneous mutual attention for automatic meta path construction. To detect camouflage behaviors, we design the label aware scoring to filter noisy neighbors. Two neural modules are combined in a unified manner called “score head mechanism” and both contribute to edge weight computation in final feature aggregation. Experiment results on real-world business datasets validate the excellent effect on fraud detection of FAHGT. The hyper-parameter sensitivity and visual analysis further show the stability and efficiency of our model. In summary, FAHGT is capable of alleviating inconsistency and discover camouflage and thus achieves state-of-art performance in most scenarios. In the future, we plan to extend our model in handing dynamic graphs data and incorporate fraud detection into other areas, such as robust item recommendation in E-commerce or loan default prediction in financial services.

# REFERENCES

1. V. S. Tseng, J. Ying, C. Huang, Y. Kao, and K. Chen, “Fraudetector: A graph-mining-based framework for fraudulent phone call detection,” in Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015, L. Cao, C. Zhang, T. Joachims, G. I. Webb,

D. D. Margineantu, and G. Williams, Eds. ACM, 2015, pp. 2157–2166.

[Online]. Available: https://doi.org/10.1145/2783258.2788623

1. J. Wang, R. Wen, and C. Wu, “Fdgars: Fraudster detection via graph convolutional networks in online app review system,” in WWW Workshops, 2019.
2. A. Li, Z. Qin, R. Liu, Y. Yang, and D. Li, “Spam review detection with graph convolutional networks,” in CIKM, 2019.
3. Z. Liu, Y. Dou, P. S. Yu, Y. Deng, and H. Peng, “Alleviating the inconsistency problem of applying graph neural network to fraud detection,” in SIGIR, 2020.
4. Y. Dou, Z. Liu, L. Sun, Y. Deng, H. Peng, and P. S. Yu, “Enhancing graph neural network-based fraud detectors against camouflaged fraudsters,” in CIKM, 2020.
5. R. Wen, J. Wang, C. Wu, and J. Xiong, “Asa: Adversary situation awareness via heterogeneous graph convolutional networks,” in WWW Workshops, 2020.
6. Y. Zhang, Y. Fan, Y. Ye, L. Zhao, and C. Shi, “Key player identification in underground forums over attributed heterogeneous information network embedding framework,” in CIKM, 2019.
7. D. Wang, J. Lin, P. Cui, Q. Jia, Z. Wang, Y. Fang, Q. Yu, and J. Zhou, “A semi-supervised graph attentive network for fraud detection,” in ICDM, 2019.
8. Z. Liu, C. Chen, X. Yang, J. Zhou, X. Li, and L. Song, “Heterogeneous graph neural networks for malicious account detection,” in CIKM, 2018. [10] Y. Dou, G. Ma, P. S. Yu, and S. Xie, “Robust spammer detection by nash reinforcement learning,” in KDD, 2020.
9. P. Kaghazgaran, M. Alfifi, and J. Caverlee, “Wide-ranging review manipulation attacks: Model, empirical study, and countermeasures,” in CIKM, 2019.
10. Z. Zhang, P. Cui, andW. Zhu, “Deep learning on graphs: A survey,” TKDE, 2020.
11. J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, “Spectral networks and locally connected networks on graphs,” arXiv preprint arXiv:1312.6203, 2013.
12. M. Defferrard, X. Bresson, and P. Vandergheynst, “Convolutional neural networks on graphs with fast localized spectral filtering,” in NeurIPS, 2016, pp. 3844–3852.
13. T. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in ICLR, 2017.
14. W. Hamilton, Z. Ying, and J. Leskovec, “Inductive representation learning on large graphs,” in NeurIPS, 2017.
15. P. Veliˇckovi´c, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, “Graph attention networks,” in ICLR, 2017.
16. X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. S. Yu, “Heterogeneous graph attention network,” in WWW, 2019, pp. 2022–2032.
17. S. Zhou, J. Bu, X. Wang, J. Chen, and C. Wang, “Hahe: Hierarchical attentive heterogeneous information network embedding,” arXiv preprint arXiv:1902.01475, 2019.
18. S. Wang, Z. Chen, D. Li, Z. Li, L.-A. Tang, J. Ni, J. Rhee, H. Chen, and

P. S. Yu, “Attentional heterogeneous graph neural network: Application to program reidentification,” in Proceedings of the 2019 SIAM International Conference on Data Mining. SIAM, 2019, pp. 693–701.