

Mini Project Report on
“SECURED NET AUCTION”

Submitted to the Department of Information Technology, SNIST in the partial fulfillment of the academic requirements for the award of

B.Tech in Information Technology

Under JNTUH

By

NIKHIL BHARGAVA E (15311A1258)

B.SHIVANI(15311A1260)

Under the guidance of

Mrs.G.Sumalatha,Assistant Professor



Department of Information Technology
School of Computer Science and Informatics
SREENIDHI INSTITUTE OF SCIENCE AND
TECHNOLOGY

(An Autonomous Institution)

Yamnampet, GhatkesarMandal, R.R Dist, Hyderabad - 501 301.

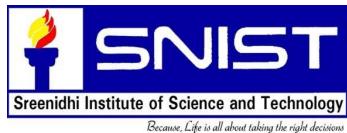
Affiliated to

Jawaharlal Nehru Technological University Hyderabad
Hyderabad- 500 085.

2017-18

Department of Information Technology

**School of Computer Science and Informatics
Sreenidhi Institute of Science and Technology**



Certificate

This is certify that the Group Project report on “SECURED NET AUCTION” is a bonafide work carried out by

Nikhil Bhargava E(15311A1258), B.Shivani(15311A1260) in the partial fulfillment for the award of B.Tech degree in Information Technology, Sreenidhi Institute of Science and Technology, Hyderabad affiliated to Jawaharlal Nehru Technology University Hyderabad, under our guidance and supervision.

Internal Guide
G.SUMA LATHA,
Assistant Professor

Acknowledgements

We would like to express our sincere thanks to Dr. P. Narasimha Reddy, Executive Director, Dr. K.Sumanth, Principal, Dr. V. V. S. S. Balaram, Professor & Head of the Department of Information Technology, Ms.G.Sumalatha, Assistant Professor & Group-Project Work Coordinator of the Department of Information Technology, SreeNidhi Institute of Science and Technology (An Autonomous Institution), Hyderabad for permitting us to do our Group-Project work.

Finally, we would also like to thank the people who have directly and indirectly helped us and parents and friends for their cooperation in completing the

Group-Project work.
Nikhil Bhargava E(15311A1258)
B.Shivani(15311A1260)

CONTENTS

Sl. No.	Details	Page No.
0	Abstract	5
1	Introduction	6
2	Survey	7
3	Software requirements	8
4	Design and Analysis	13
5	Data Design	16
6	Modelling	22
7	Implementation of outputs	27
8	Testing	43
9	Conclusion	45
10	Reference	46

ABSTRACT

Secured Net Auction main idea is to implement a web application for bidding different category of goods from home. This application front end is implemented in java platform and oracle is used for backend data storage purpose. Auctions are one of the oldest procedures for bidding products but which need manual presence on the site for participating in bidding but with this application users can bid from home using a simple online application. In this application, users should register with the application by providing accurate contact information and get unique id and password. Users who are interested in selling their products through bidding can contact administrator and submit their product with basic bidding price.

Admin will check the product with its details and if it satisfies all rules and regulations he will make the product online. In regular bidding procedure bidders will not share different bidder's information which is not a efficient method so in this online application we provide detailed information of bidders and consider all security issues.

As soon as the product goes online registered users will start bidding on the product and process continues until the product is sold out.

1. INTRODUCTION

1.1 Introduction

Since the emergence of the World Wide Web (WWW), electronic commerce, commonly known as e-commerce, has become more and more popular. Websites such as eBay and Amazon allow Internet users to buy and sell products and services online, which benefits everyone in terms of convenience and profitability. The traditional online shopping business model allows sellers to sell a product or service at a preset price, where buyers can choose to purchase if they find it to be a good deal. Copyright is held by the International World Wide Web Conference Committee (IW3C2). Distribution of these papers is limited to classroom use, and personal use by others. WWW 2012, April 16–20, 2012, Lyon, France. ACM 978-1-4503-1229-5/12/04. Online auction however is a different business model by which items are sold through price bidding. There is often a starting price and expiration time specified by the sellers. Once the auction starts, potential buyers bid against each other, and the winner gets the item with their highest winning bid. Similar to any platform supporting financial transactions, online auction attracts criminals to commit fraud. The varying types of auction fraud are as follows. Products purchased by the buyer are not delivered by the seller. The delivered products do not match the descriptions that were posted by sellers. Malicious sellers may even post non-existing items with false descriptions to deceive buyers, and request payments to be wired directly to them via bank-to-bank wire transfer. Furthermore, some criminals apply phishing techniques to steal high-rated seller's accounts so that potential buyers can be easily deceived due to their good rating. Victims of fraud transactions usually lose their money and in most cases are not recoverable. As a result, the reputation of the online auction services is hurt significantly due to fraud crimes.

We consider the problem of building online machine-learned models for detecting auction frauds in e-commerce web sites. Since the emergence of the world wide web, online shopping and online auction have gained more and more popularity. While people are enjoying the benefits from online trading, criminals are also taking advantages to conduct fraudulent activities against honest parties to obtain illegal profit.

Survey on:

1. Spatio-Temporal Models for Estimating Click-through Rate:

We propose novel spatio-temporal models to estimate click throughrates in the context of content recommendation. We track article CTR at a fixed location over time through a dynamicGamma-Poisson model and combine information fromcorrelated locations through dynamic linear regressions, significantlyimproving on per-location model. Our models adjustfor user fatigue through an exponential tilt to the first view CTR (probability of click on first article exposure) thatis based only on user-specific repeat-exposure features. We illustrate our approach on data obtained from a module (TodayModule) published regularly on Yahoo! Front Page anddemonstrate significant improvement over commonly usedbaseline methods. Large scale simulation experiments tostudy the performance of our models under different scenariosprovide encouraging results. Throughout, all modelingassumptions are validated via rigorous exploratory dataanalysis.

2. Detecting Frauds In Online Auction System:

We know that these days online shopping increased drastically. E-commerce is growing faster thanpredicted as it is up over 400% in the past 3 years. As customers have the ease to buy things without spending muchtime there are also criminals who try to fraud and get profit in illegal ways. Many pro-active fraud detectiontechniques came to reduce the frauds and illegal activities. Machine-learned models especially those are learned online, are able to catch frauds more efficiently and quickly than human tuned rule-based systems. In this paper, we study about online fraud detection.

3. Online Auction Fraud And Ebay:

EBay claims that users face only a 1 in 10,000 risk of fraud. Evidence from the FBI shows the rate ismore like 1 in 100, or one hundred times greater than the company alleges. Despite the usefulness ofeBay's feedback mechanism buyers still face substantial risk of being defrauded. This paper detailsthe level and dimension of online fraud, examines the history of fraud on eBay, followed by a discussion of feedback and its attendant limitations. EBAY's new feedback policies and other fraud reduction efforts are then evaluated. Finally, fraud reduction recommendations are provided for allthree auction participant groups: buyers, sellers, and eBay itself.

3. SOFTWARE REQUIREMENTS SPECIFICATION

3.1 Requirement Analysis

The requirements gathering process takes as its input the goals identified in the high-level requirements section of the project plan. Each goal will be refined into a set of one or more requirements. These requirements define the major functions of the intended application, define operational data areas and reference data areas, and define the initial data entities. Major functions include critical processes to be managed, as well as mission critical inputs, outputs and reports.

1. Problem statement
2. Data flow diagrams
3. Use case diagram
4. Other UML diagrams.

The above mentioned documents gives us diagrammatical view of the system what we are going to develop.

3.2 Problem Statement

The problem statement system only requires a semi-trusted third party, responsible for carrying out simple matching operations correctly.

3.3 Purpose

The purpose of this document is to present a detailed description of “**Online Modeling of Proactive Moderation System for Auction Fraud Detection**” application. It will explain the purpose and features of the system that it will provide, constraints under which it must operate and how the system will react. The document also describes the non functional requirements of the system.

3.3.1 Technologies Used

JAVA PLATFORM

The programmer writes Java source code in a text editor which supports plain text. Normally the programmer uses an *Integrated Development Environment* (IDE) for programming. An

IDE supports the programmer in the task of writing code, e.g. it provides auto-formatting of the source code, highlighting of the important keywords, etc.

J2SE

J2SE is a collection of Java Programming Language API (Application programming interface) that is very useful to many Java platform programs. It is derived from one of the most dynamic programming language known as "JAVA"

J2SE is a collection of Java Programming Language API (Application programming interface) that is very useful to many Java platform programs. It is derived from one of the most dynamic programming language known as "JAVA" and one of its three basic editions of Java known as Java standard edition being used for writing Applets and other web based applications.

J2EE

Java 2 Platform Enterprise Edition. J2EE is a platform-independent, Java-centric environment from Sun for developing, building and deploying Web-based enterprise applications online. The J2EE platform consists of a set of services, APIs, and protocols that provide the functionality for developing multilayer, Web-based applications.

3.3.2 JAVA DATA BASE CONNECTION

JDBC stands for **Java Database Connectivity**, which is a standard Java API for database-independent connectivity between the Java programming language and a wide range of databases.

The JDBC library includes APIs for each of the tasks commonly associated with database usage:

- Making a connection to a database
- Creating SQL or MySQL statements
- Executing that SQL or MySQL queries in the database
- Viewing & Modifying the resulting records

All of these different executables are able to use a JDBC driver to access a database and take advantage of the stored data.

JDBC provides the same capabilities as ODBC, allowing Java programs to contain database-independent code.

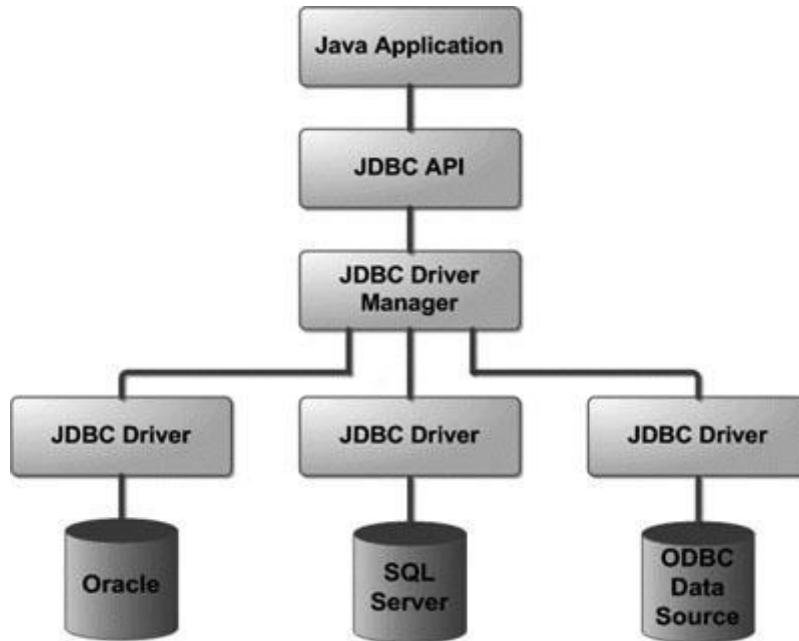
JDBC Architecture:

The JDBC API supports both two-tier and three-tier processing models for database access but in general JDBC Architecture consists of two layers:

- **JDBC API:** This provides the application-to-JDBC Manager connection.
- **JDBC Driver API:** This supports the JDBC Manager-to-Driver Connection.

The JDBC API uses a driver manager and database-specific drivers to provide transparent connectivity to heterogeneous databases. The JDBC driver manager ensures that the correct driver is used to access each data source. The driver manager is capable of supporting multiple concurrent drivers connected to multiple heterogeneous databases.

Following is the architectural diagram, which shows the location of the driver manager with respect to the JDBC drivers and the Java application.



3.5 Software Requirements

The software interface is the operating system, and application programming interface used for the development of the software.

Operating System: Windows XP or higher / Linux

Platform: JDK

Application Server: Apache Tomcat 7 or higher

Database: MySQL

Technologies used: Java, J2EE.

3.6 Hardware Requirements

CLIENT				
OPERATING SYSTEM	SOFTWARE	PROCESSOR	RAM	Hard disk
Windows/Linux	Any Advanced Browser. Chrome/Opera	Intel/AMD processor	256 Mb	160 GB

Table 2:1 Client Requirements

SERVER				
OPERATING SYSTEM	SOFTWARE	PROCESSOR	RAM	HARD DISK
Windows/Linux	Any Browser. JDK 1.7 or above Apache 7 or above MySQL 5.0 or above	Intel/AMD processor	256Mb	160 GB

Table 2:2 Server Requirements

3.7 Functional Requirements (Modules)

The project having its own respective functionalities.

Our application is to detect online auction frauds for a major Asian site where hundreds of thousands of new auction cases are posted every day. Every new case is sent to the proactive anti-fraud moderation system for pre-screening to assess the risk of being fraud. The current system is featured by:

- **Rule-based features:**

Human experts with years of experience created many rules to detect whether a user is fraud or not. An example of such rules is “blacklist”, i.e. whether the user has been detected or complained as a fraud before. Each rule can be regarded as a binary feature that indicates the fraud likeliness.

- **Linear scoring function:**

The existing system only supports linear models. Given a set of coefficients (weights) on features, the fraud score is computed as the weighted sum of the feature values.

3.8 Non-Functional Requirements

3.8.1 Flexibility & Scalability

Oracle itself has given a set of applications with JDK but the whole developer community can develop their own applications and they have access to same resources and public API which are accessible to core applications.

3.8.2 Fragmentation

Java gave the same environment which is open; the entire API's which is open to all the devices which reduces fragmentation. If you develop a Java application, it will run on all the devices.

3.8.3 Open Source:

Java open source is free and easy to download. Java is a platform-independent programming language and The Java virtual machine (JVM) is a software implementation of a computer that executes programs like a real machine.

3.8.4 Reliability:

Since the application is being developed through Java, the most famous, efficient and reliable language, so it is reliable in every aspect until and unless there is an error in the programming side. Thus the application can be a compatible and reliable one.

3.8.5 Feasibility study

A key part of the preliminary investigation that reviews anticipated costs and benefits and recommends a course of action based on operational, technical, economic, and time

factors. The purpose of the study is to determine if the systems request should proceed further.

4.1 ANALYSIS AND DESIGN

4.1.1 Purpose

In this section the purpose of the document and the project is described.

4.1.1.1 Document Purpose

An SDD (Software design description) is a representation of a software system that is used as a medium for communicating software design information.

4.1. Project Purpose

The prime purpose of this “**Online Modeling of Proactive Moderation System for Auction Fraud Detection**” is to create a fully fledged web application which would communicate with the remote server to send and retrieve data as per requirement. This application works when there is internet connectivity

The application retrieves user details from the server and syncs it to the user machines. These details can be stored in the Mysql database. We consider the problem of building online machine-learned models for detecting auction frauds in e-commerce web sites. In this paper, we propose an online probit model framework which takes online feature selection, coefficient bounds from human knowledge and multiple instance learning into account simultaneously. By empirical experiments on a real-world online auction fraud detection data we show that this model can potentially detect more frauds and significantly reduce customer complaints compared to several baseline models and the humantuned rule-based system.

4.1.2 Scope

In this section the scope of the document and the project is explained in brief.

4.1.2.1 Document Scope

This document contains a thorough description of the high level architecture that will be used in developing the system. Communicating at a purposefully high level, it will only form the basis for the Software Detailed Design and implementation. However, the SDD itself will not be in sufficient detail to implement the code. It will convey the overall system design of the system, the user interface design and higher level module design and the

architecture working of the Java Virtual Machine. Design details that will not be included in the SDD are:

- Low level classes that will be used in the implementation. The full description of the implementation of each module is not needed, but the public modules that will be interfaced will be described.
- Exact detailed description of interactions within each module

4.2 System Overview

4.2.1 Development Tools

Java framework uses certain development tools which are as follows:

4.2.1.1 JDK

The Java Development Kit (JDK) is provided by Sun Microsystems as a basic development environment for Java. The JDK provides similar facilities to the cc compiler for C programs, plus a JVM simulator and some additional facilities such as a debugger. To use the JDK, programs are constructed as ascii text files (by using an editor, for example). The program files are compiled, which translates the Java code to JVM bytecode in .class files.

Each public class must be in a file having the class name (case sensitive on Unix) followed by a .java suffix. There may be any number of classes defined in a .java file, but the compiler produces a separate .class file for each class. A file is compiled with the **javac** command, which is similar to the cc (or gcc) command. A class is executed (or more precisely, the method **main** in a class is executed) by the command **java** with the class name (not the .class file) as the parameter. Thus, for example, to compile the program in file Hi.java, we would use the command

javac Hi.java

and then to execute the program we would use the command

java Hi

Both compile-time and execution-time (exceptions) error messages include the file name and line where the error occurred. No .class file is produced if there is a compile-time error.

4.2.1.2 TOMCAT 7.0 WEB SERVERS

Apache Tomcat is a web container developed at the Apache Software Foundation (ASF). Tomcat implements the servlet and the Java Server Pages (JSP) specifications from Sun Microsystems, providing an environment for Java code to run in cooperation with a web server. It adds tools for configuration and management but can also be configured by editing configuration files that are normally XML-formatted. Tomcat includes its own HTTP server internally.

- The deployer package includes a ready to use Ant script, with the following targets:
- compile (default): Compile and validate the web application. This can be used standalone, and does not need a running Tomcat server. The compiled application will only run on the associated Tomcat 5.0.x server release, and is not guaranteed to work on another Tomcat release, as the code generated by Jasper depends on its runtime component. It should also be noted that this target will also compile automatically any Java source file located in the /WEB-INF/classes folder of the web application.
- deploy: Deploy a web application (compiled or not) to a Tomcat server
- undeploy: Undeploy a web application
- Start: Start web application
- reload: Reload web application
- Stop: Stop web application

4.2.1.3 MySQL

The MySQL (TM) software delivers a very fast, multi-threaded, multi-user, and robust SQL (Structured Query Language) database server. MySQL Server is intended for mission critical, heavy-load production systems as well as for embedding into mass-deployed software. MySQL is a trademark of MySQL AB. The MySQL software has Dual Licensing, which means you can use the

MySQL software free of charge under the GNU General Public License (<http://www.gnu.org/licenses/>). You can also purchase commercial MySQL licenses from MySQL AB if you do not wish to be bound by the terms of the GPL. The MySQL web site (<http://www.mysql.com/>) provides the latest information about the MySQL software.

What Is MySQL?

MySQL, the most popular Open Source SQL database, is developed and provided by MySQL AB. MySQL AB is a commercial company that builds its business providing services around the MySQL database. The MySQL web site (<http://www.mysql.com/>) provides the latest information about MySQL software and MySQL AB.

Why use the MySQL Database Server?

The MySQL Database Server is very fast, reliable, and easy to use. If that is what you are looking for, you should give it a try. MySQL Server also has a practical set of features developed in close cooperation with our users. You can find a performance comparison of MySQL Server to some other database managers on our benchmark page.

4.4 Data Design

4.4.1 Databases

MySQL RDBMS

Name
Auction

Table 4:1 MySQL Database

4.4.2.1 Table: FLOGIN

Columns

Alter Table 'cregister' in 'auction'			
	Field Name	Datatype	Len
*	cid	varchar	250
	cname	varchar	250
	caddr	varchar	250
	cdist	varchar	250
	cstate	varchar	250
	ccountry	varchar	250
	czip	varchar	250
	cphn	varchar	250
	cemail	varchar	250
	cpwd	varchar	250
	cstatus	varchar	250

Table 4:2 Structure of CREGISTER table of MySQL Database

Definition

```
CREATE TABLE `cregister` (
  `cid` varchar(250) DEFAULT NULL,
  `cname` varchar(250) DEFAULT NULL,
  `caddr` varchar(250) DEFAULT NULL,
  `cdist` varchar(250) DEFAULT NULL,
  `cstate` varchar(250) DEFAULT NULL,
  `ccountry` varchar(250) DEFAULT NULL,
  `czip` varchar(250) DEFAULT NULL,
  `cphn` varchar(250) DEFAULT NULL,
  `cemail` varchar(250) DEFAULT NULL,
  `cpwd` varchar(250) DEFAULT NULL,
  `cstatus` varchar(250) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

4.4.2.2 Table: sregister

Columns:

Alter Table 'sregister' in 'auction'			
	Field Name	Datatype	Len
*	sid	varchar	250
	sname	varchar	250
	saddr	varchar	250
	sdist	varchar	250
	sstate	varchar	250
	szip	varchar	250
	sphn	varchar	250
	semail	varchar	250
	spwd	varchar	250
	sstatus	varchar	250

Table 4:4 Structure of SREGISTERTableMySQL Database

Definition:

```
CREATE TABLE `sregister` (
  `sid` varchar(250) DEFAULT NULL,
  `sname` varchar(250) DEFAULT NULL,
  `saddr` varchar(250) DEFAULT NULL,
  `sdist` varchar(250) DEFAULT NULL,
  `sstate` varchar(250) DEFAULT NULL,
  `szip` varchar(250) DEFAULT NULL,
  `sphn` varchar(250) DEFAULT NULL,
  `semail` varchar(250) DEFAULT NULL,
  `spwd` varchar(250) DEFAULT NULL,
  `sstatus` varchar(250) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

4.4.2.3Table: SDETAILS

Columns:

	Field Name	Datatype	Len
*	sid	varchar	250
	sname	varchar	250
	semail	varchar	250
	pid	varchar	250
	cate	varchar	250
	subcate	varchar	250
	pname	varchar	250
	pcost	varchar	250
	desp	varchar	2000
	pimage	varchar	200
	pstatus	varchar	200
	pdate	varchar	200
	pvalidate	varchar	200

Table 4:3 Structure of SDETAILS table of MySQL Database

Definition:

```
CREATE TABLE `sdetails` (
  `sid` varchar(250) DEFAULT NULL,
  `sname` varchar(250) DEFAULT NULL,
  `semail` varchar(250) DEFAULT NULL,
  `pid` varchar(250) DEFAULT NULL,
  `cate` varchar(250) DEFAULT NULL,
  `subcate` varchar(250) DEFAULT NULL,
  `pname` varchar(250) DEFAULT NULL,
  `pcost` varchar(250) DEFAULT NULL,
  `desp` varchar(2000) DEFAULT NULL,
  `pimage` varchar(200) DEFAULT NULL,
  `pstatus` varchar(200) DEFAULT NULL,
  `pdate` varchar(200) DEFAULT NULL,
  `pvalidate` varchar(200) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

4.4.2.4 Table: UCOST

Columns:

	Field Name	Datatype	Len
*	cid	varchar	250
	cemail	varchar	250
	pid	varchar	250
	cate	varchar	250
	subcate	varchar	250
	pname	varchar	250
	pcost	varchar	250
	pimage	varchar	200
	ucost	varchar	200
	pvalidate	varchar	200
	upst	varchar	200
	sid	varchar	200
	sname	varchar	200
	semail	varchar	200

Table 4:4 Structure of UCOST table of MySQL Database

Definition:

```
CREATE TABLE `ucost` (
  `cid` varchar(250) DEFAULT NULL,
  `cemail` varchar(250) DEFAULT NULL,
  `pid` varchar(250) DEFAULT NULL,
  `cate` varchar(250) DEFAULT NULL,
  `subcate` varchar(250) DEFAULT NULL,
  `pname` varchar(250) DEFAULT NULL,
  `pcost` varchar(250) DEFAULT NULL,
  `pimage` varchar(200) DEFAULT NULL,
  `ucost` varchar(200) DEFAULT NULL,
  `pvalidate` varchar(200) DEFAULT NULL,
  `upst` varchar(200) DEFAULT NULL,
  `sid` varchar(200) DEFAULT NULL,
  `sname` varchar(200) DEFAULT NULL,
  `semail` varchar(200) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

4.4.2.5 Table: GRAPH

Columns:

Alter Table 'graph' in 'auction'			
	Field Name	Datatype	Len
*	sno	int	11
	Avg	int	11
	Tvalue	int	11
	Total	int	11
	email	varchar	100

Table 4:5 Structure of GRAPHtable of MySQL Database

Definition:

```
CREATE TABLE `graph` (
    `sno` int(11) DEFAULT NULL,
    `Avg` int(11) DEFAULT NULL,
    `Tvalue` int(11) DEFAULT NULL,
    `Total` int(11) DEFAULT NULL,
    `email` varchar(100) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

4.4.2.6Table: UCOST

Columns:FEEDBACK

Alter Table 'feedback' in 'auction'				
	Field Name	Datatype	Len	Def
*	fbid	varchar	250	
	cid	varchar	250	
	pid	varchar	250	
	cfbpoints	varchar	250	
	cdesp	varchar	2500	
	sid	varchar	200	
	sname	varchar	200	
	semail	varchar	200	

Table 4:6 Structure of FEEDBACK table of MySQL Database

Definition:

```
CREATE TABLE `feedback` (
  `fbid` varchar(250) DEFAULT NULL,
  `cid` varchar(250) DEFAULT NULL,
  `pid` varchar(250) DEFAULT NULL,
  `cfbpoints` varchar(250) DEFAULT NULL,
  `cdesp` varchar(2500) DEFAULT NULL,
  `sid` varchar(200) DEFAULT NULL,
  `sname` varchar(200) DEFAULT NULL,
  `semail` varchar(200) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

5. MODELING

5.1 Design

Requirements gathering followed by careful analysis leads to a systematic Object Oriented Design (OOAD). Various activities have been identified and are represented using Unified Modeling Language (UML) diagrams. UML is used to specify, visualize, modify, construct and document the artifacts of an object-oriented software-intensive system under development.

5.1.1. Use Case Diagram

In the Unified Modeling Language (UML), the use case diagram is a type of behavioral diagram defined by and created from a use-case analysis. It represents a graphical overview of the functionality of the system in terms of actors, which are persons, organizations or external systems that play a role in one or more interactions with the system. These are drawn as stick figures. The goals of these actors are represented as use cases, which describe a sequence of actions that provide something of measurable value to an actor and any dependencies between those use cases.

In this application there is only one actor – soldier and below is the use case diagram of this application.

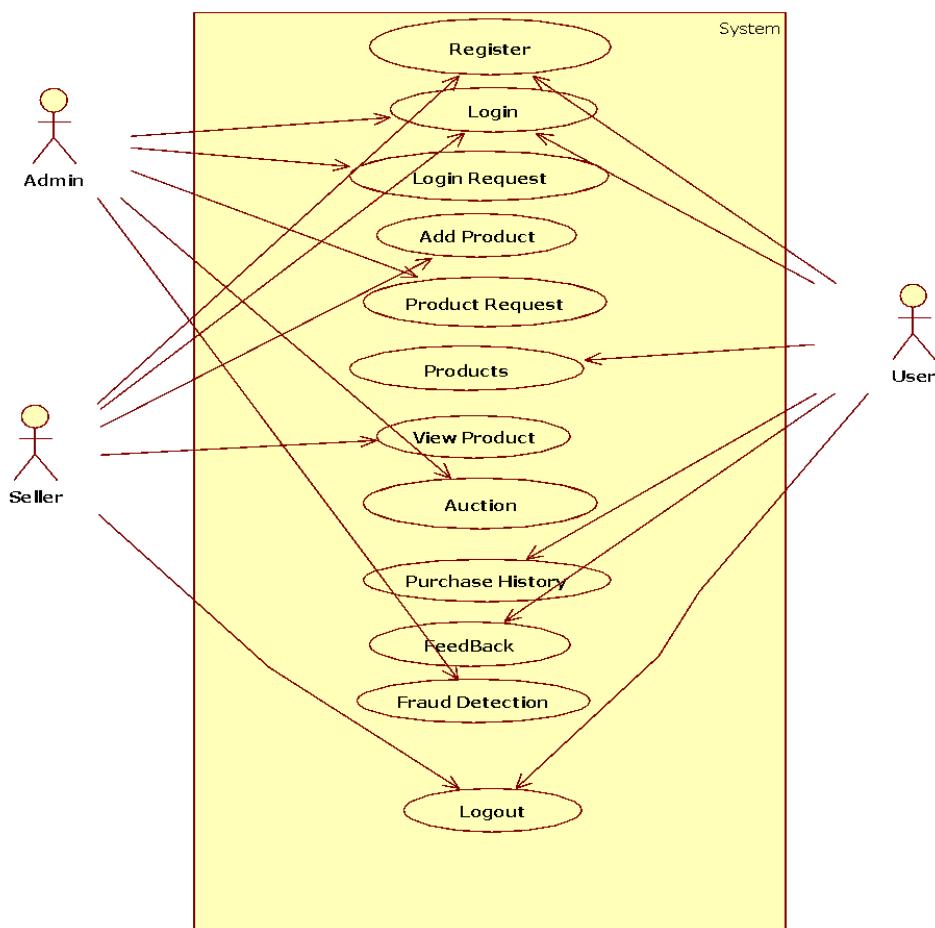


Figure: Use Case Diagram for System

5.1.2 Sequence Diagram

UML sequence diagrams are used to show how objects interact in a given situation. An important characteristic of a sequence diagram is that time passes from top to bottom: the interaction starts near the top of the diagram and ends at the bottom (i.e. Lower equals later).

A popular use for them is to document the dynamics in an object-oriented system. For each key, collaboration diagrams are created that show how objects interact in various representative scenarios for that collaboration.

Sequence diagram is the most common kind of interaction diagram, which focuses on the message interchange between a numbers of lifelines.

The following nodes and edges are typically drawn in a UML sequence diagram: lifeline, execution specification, message, combined fragment, interaction use, state invariant, continuation, destruction occurrence.

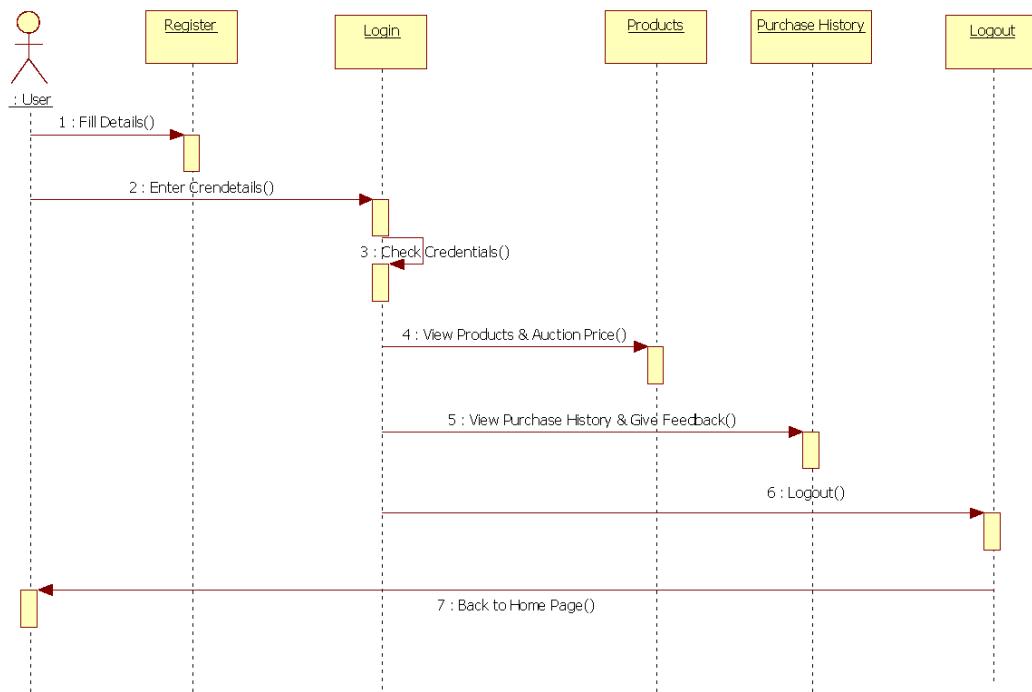


Figure : Sequence Diagram for User

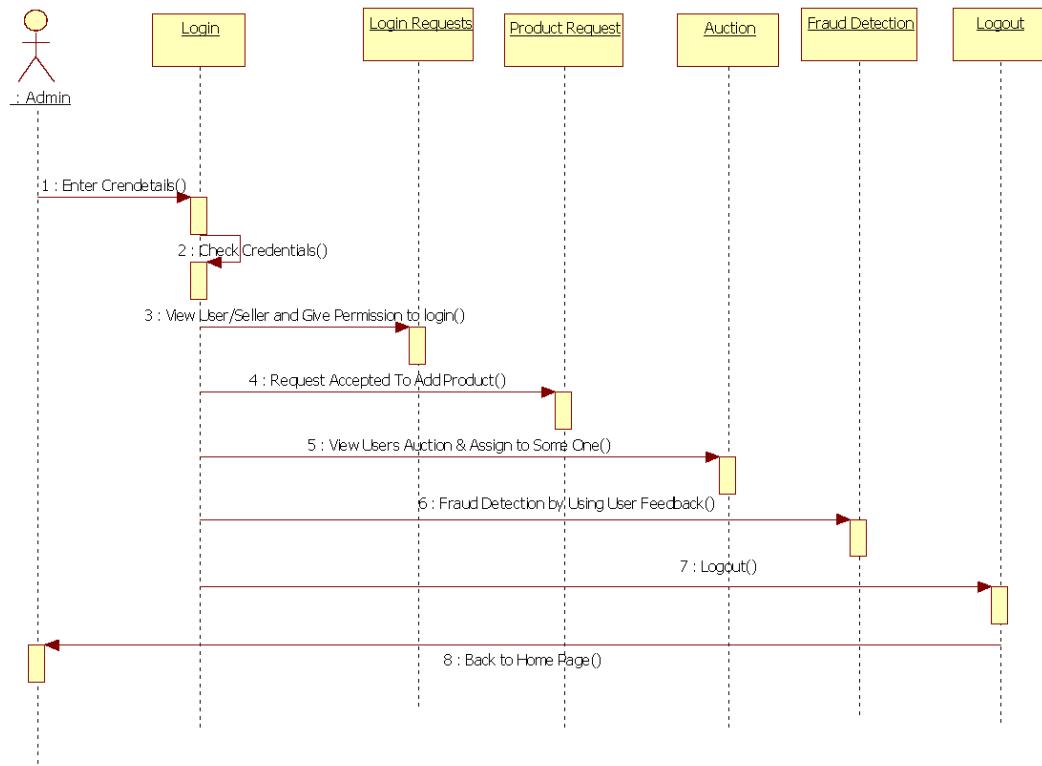


Figure : Sequence Diagram for Admin

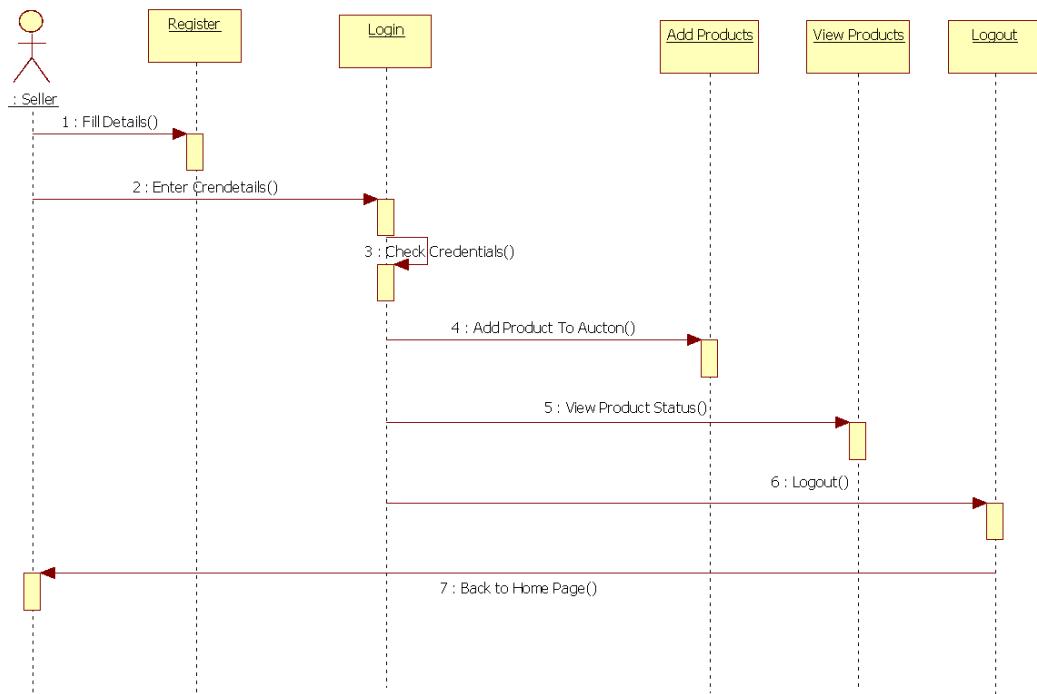


Figure : Sequence Diagram for Seller

5.1.9 Deployment Diagram

Deployment diagram shows execution architecture of systems that represent the assignment (deployment) of software artifacts to deployment targets (usually nodes).

Nodes represent either hardware devices or software execution environments. They could be connected through communication paths to create network systems of arbitrary complexity. Artifacts represent concrete elements in the physical world that are the result of a development process and are deployed on nodes.

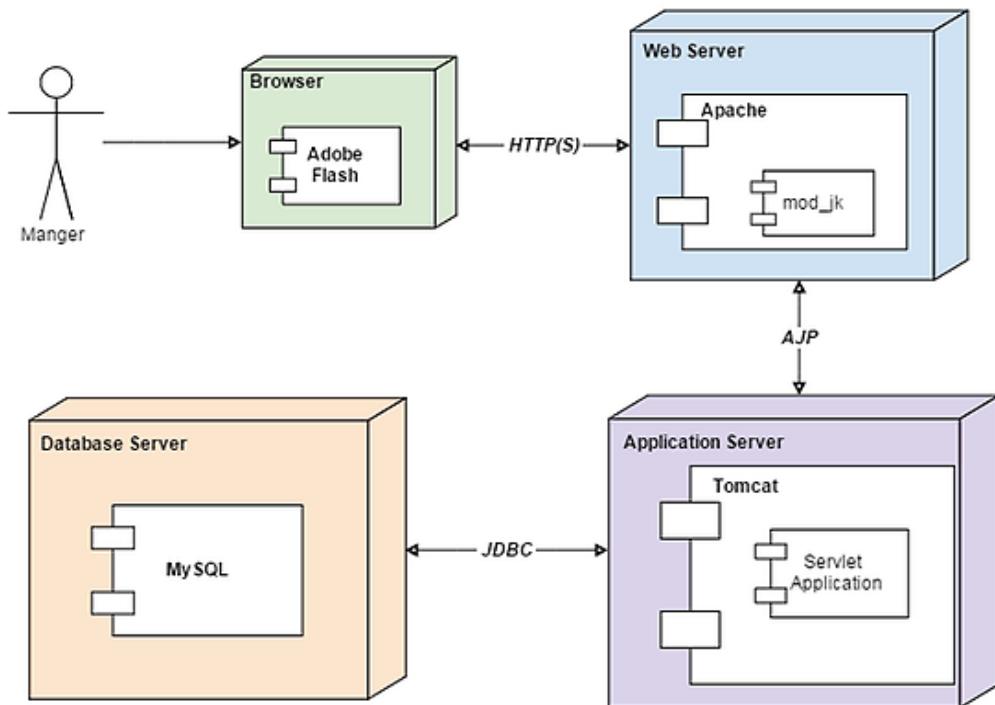


Figure : Deployment Diagram of the system

6. IMPLEMENTATION

6.1 Sample Code

6.1.1 Code for databasecon.java

```
package databaseconnection;  
import java.sql.*;  
  
public class databasecon  
{  
    static Connection co;  
    public static Connection getconnection()  
    {  
  
        try  
        {  
            Class.forName("com.mysql.jdbc.Driver");  
            co =  
                DriverManager.getConnection("jdbc:mysql://localhost:3306/auction","root","root");  
        }  
        catch(Exception e)  
        {  
            System.out.println("Database Error"+e);  
        }  
        return co;  
    }  
}
```

6.1.2 Code for arequest4.jsp

```
<%@page import="java.sql.* ,java.util.* ,java.text.* ,Mail.*"%>
<%@page import="databaseconnection.*"%>
<html>
<head>
<title></title>

</head>

<body>
<%
String tdname=null,tdmno=null;

String aid= request.getParameter("aid");
String email=request.getParameter("email");

try
{
ResultSet rs2=null;
ResultSet rs3=null;
Connection con=databasecon.getconnection();
Statement st=con.createStatement();

intrs=st.executeUpdate("update cregister set cstatus='approve' where cid='"+aid+"' and
cemail='"+email+"'");
if(rs!=0)
{
response.sendRedirect("arequests2.jsp?req=success");
}
}
```

```
}

catch(Exception e1)
{
out.println(e1.getMessage());
}
```

```
%>
</body>
</html>
```

```
<%@ include file="footer.jsp"%>
```

6.1.3 Code for fraud.jsp

```
<%@include file="theader.jsp"%>
<%@page import="java.sql.* ,java.util.* ,java.text.* "%>
<%@page import="databaseconnection.* "%>
<html>
<body>
<br><br><br>
<div>

<center><h1><font size="7" color="orange">Auction Page</font></h1></center>
```

```
<table align="center" width="900" cellpadding="5">
<tr align="center">
<th align="center"><h3><font color="orange">Email</font></h3></th>
<th align="center"><h3><font color="orange">Avg score</font></h3></th>
<th align="center"><h3><font color="orange">Total Records</font></h3></th>
```

```

<th align="center"><h3><font color="orange">Status</h3></th>
</tr>

<%
    if(request.getParameter("date")!=null)
    {
        out.println("<script>alert('The Product Date Is Not
Exceed ')</script>");
    }

    if(request.getParameter("fail")!=null)
    {
        out.println("<script>alert('Request
Accepted')</script>");
    }

    if(request.getParameter("succ")!=null)
    {
        out.println("<script>alert('Product
User')</script>");
    }
%>

<br>

<%
String pcost, cid, pid,semail,status,subcate,pname,pimage;
try
{
    ResultSets=null;
    ResultSet rs2=null;

```

```

ResultSet rs3=null;

Connection con = databasecon.getconnection();

Statement st=con.createStatement();
Statement st2=con.createStatement();
st2.executeUpdate("delete from graph");
Statement st3=con.createStatement();
rs=st.executeQuery("select distinct semail from ucost where upst='winner' ");

while(rs.next())
{
    semail=rs.getString("semail");
    rs2=st2.executeQuery("select avg(cfbpoints), count(*) from feedback where
semail='"+semail+"' ");
    if(rs2.next()){
        st3.executeUpdate("insert into graph values('1',
"+rs2.getInt(2)+",'5','"+rs2.getInt(1)+"','"+semail+"')");
        if(rs2.getInt(1)>=5){
            status="normal";
        }else{
            status="fraud";
        }
    }
%>

<!--tr><td><%=semail%><td><%=rs2.getInt(2)%><td><%=rs2.getInt(1)%><td><%=status
%><td><a href="graph.jsp?e=<%=semail%>">Graph</a-->
<tr align="center">
<td align="center"><p style="(255,0,255); "><font size="4"><%=semail%></font></p></td>
<td align="center"><p style="(255,0,255); "><font size="4"><%=rs2.getInt(2)%></font></p></td>
<td><p style="(255,0,255); "><font size="4"><%=rs2.getInt(1)%></font></p></td>
<td><p style="(255,0,255); "><font size="4"><%=status%></font></p></td>

```

```
<td><p style="(255,0,255);"><font size="4"><a href="graph.jsp?e=<%=semail%>">Graph</a></font></p></td></tr>
<%
}
%>

<%
}>%>

</table>
</form>

<%
}

catch(Exception e1)
{
out.println(e1);
}

%>
<div>
</body>
</html>

<br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br>
```

```
<%@ include file="footer.jsp"%>
```

6.1.4 Code for graph.jsp

```
<%@ page import="databaseconnection.*" %>

<%@ page import="java.sql.*" %>
<%@ page import="java.io.*" %>
<%@ page import="java.awt.*" %>
<%@ page import="org.jfree.chart.ChartFactory" %>
<%@ page import="org.jfree.chart.ChartUtilities" %>
<%@ page import="org.jfree.chart.JFreeChart" %>
<%@ page import="org.jfree.chart.plot.PlotOrientation" %>
<%@ page import="org.jfree.data.*" %>
<%@ page import="org.jfree.data.jdbc.JDBCCategoryDataset" %>

<%
String topic=request.getParameter("id");

int c1=0, c2=0, c3=0,c4=0;
try
{
Connection con = databasecon.getconnection();
Statement st=con.createStatement();

String query="SELECT * from graph where email='"+request.getParameter("e")+"'";
JDBCCategoryDataset dataset=new JDBCCategoryDataset("jdbc:mysql://localhost:3306/auction","com.mysql.jdbc.Driver","root","root");

dataset.executeQuery( query);
```

```

JFreeChart chart = ChartFactory.createBarChart3D("EVALUATION GRAPH", "", "", dataset,
PlotOrientation.VERTICAL, true, true, true);
//
ChartUtilities.saveChartAsJPEG(new File("D:/Apache
8.0/webapps/Auction/images/logotype.jpg"), chart, 700, 300);
response.sendRedirect("graph2.jsp");
}
catch (Exception e)
{
System.out.println("Problem in creating chart."+e);
}
%>

```

6.1.4 Code for aproview.jsp

```

<%@page import="java.sql.* ,java.util.* ,java.text.* ,Mail.*"%>
<%@page import="databaseconnection.*"%>
<html>
<head>
<title></title>

</head>

<body>
<%
String tdname=null,tdmno=null;

String pid= request.getParameter("aid");

try
{

```

```

ResultSet rs2=null;
ResultSet rs3=null;
Connection con=databasecon.getconnection();
Statement st=con.createStatement();

intrs=st.executeUpdate("update sdetails set pstatus='not approve' where pid='"+pid+"'");
if(rs!=0)
{
    response.sendRedirect("aproview.jsp?fail=Approved");
}

}

catch(Exception e1)
{
out.println(e1.getMessage());
}

%>
</body>
</html>

<%@ include file="footer.jsp"%>

```

6.2 Screen Captures

6.2.1 Welcome Screen



Figure 6-1 Welcome Activity

Description: After launching the application a Welcome message will come into view.

6.2.2 Login Screen for Admin

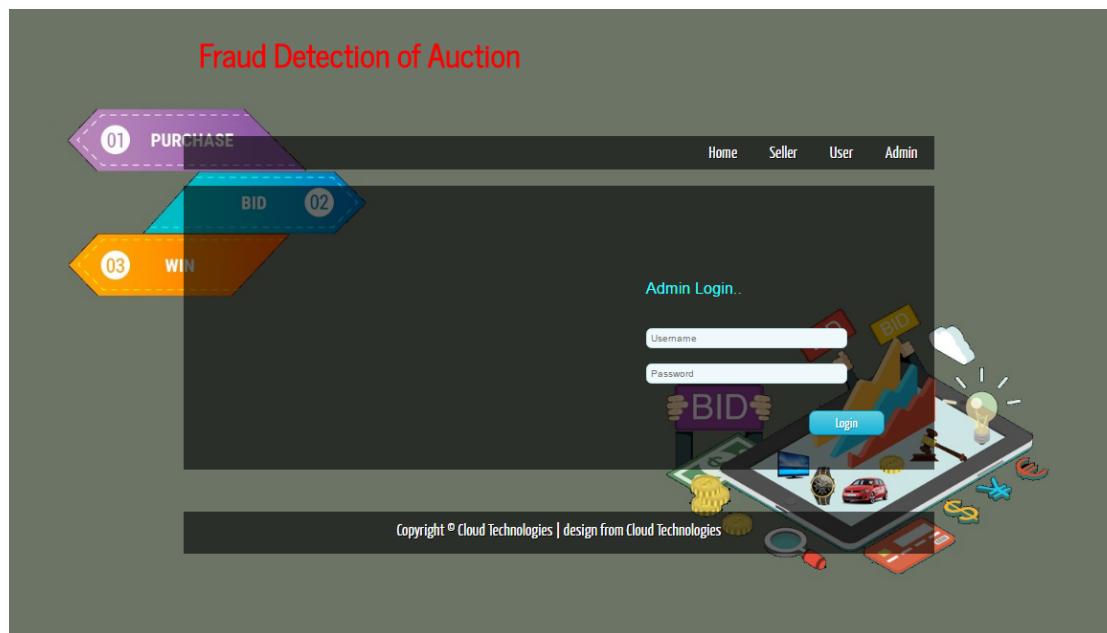


Figure 6-2 Login Activity

Description: The application has an admin login which is used to start the application to prevent authenticity infringement.

6.2.3 Login Screen for User

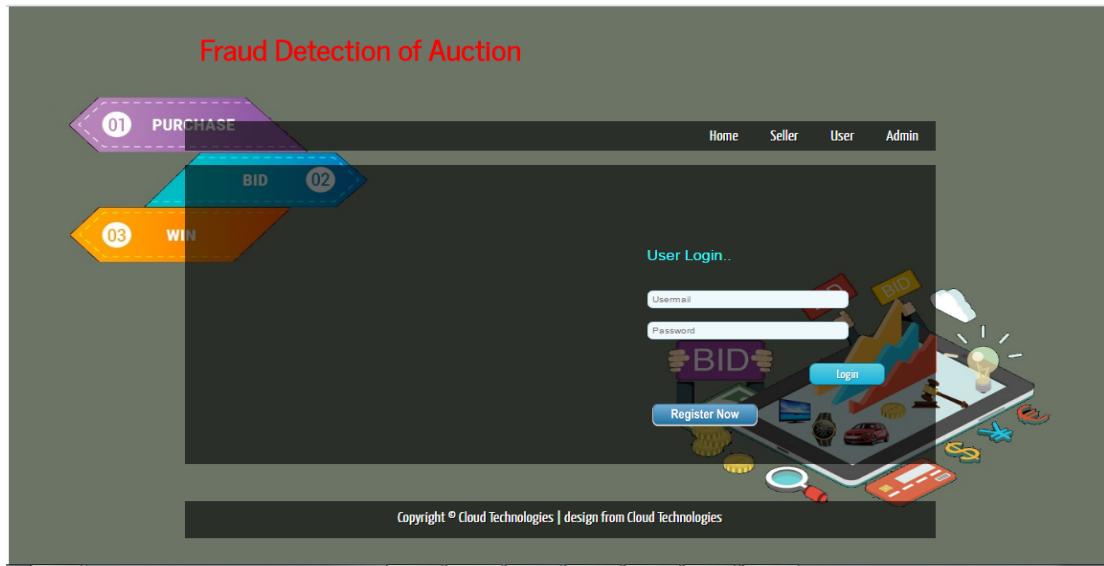


Figure 6-3 Login Activity

Description: The application has an User login which is used to start the application to prevent authenticity infringement.

6.2.4 User & Seller Registration

The image shows the registration screen for the 'Fraud Detection of Auction' application. On the left, there is a vertical sidebar with three steps: '01 PURCHASE' (purple arrow), 'BID' (teal arrow), and '03 WIN' (orange arrow). The main area features a 'Register Here' form with fields for Name, Address, District, State, ZIP (with a validation message 'Please fill out this field.'), Contact No, Email, and Password. A 'Register' button is located at the bottom of the form. To the right of the form is a decorative graphic of a smartphone displaying a bidding interface with various items like a watch, car, and TV. The background of the page is dark grey, and at the bottom, there is a copyright notice: 'Copyright © Cloud Technologies | design from Cloud Technologies'.

Figure 6-4User Registration

Description: In this activity is useful for the new user to register themselves by giving their valid details such as email id, user name, Phone number, etc. The user has to fill all the details else message is displayed to the user.

6.2.5 Login Request



Figure 6-5 Login Request

Description: This activity used to permission to login .

6.2.6 Login Screen for Seller

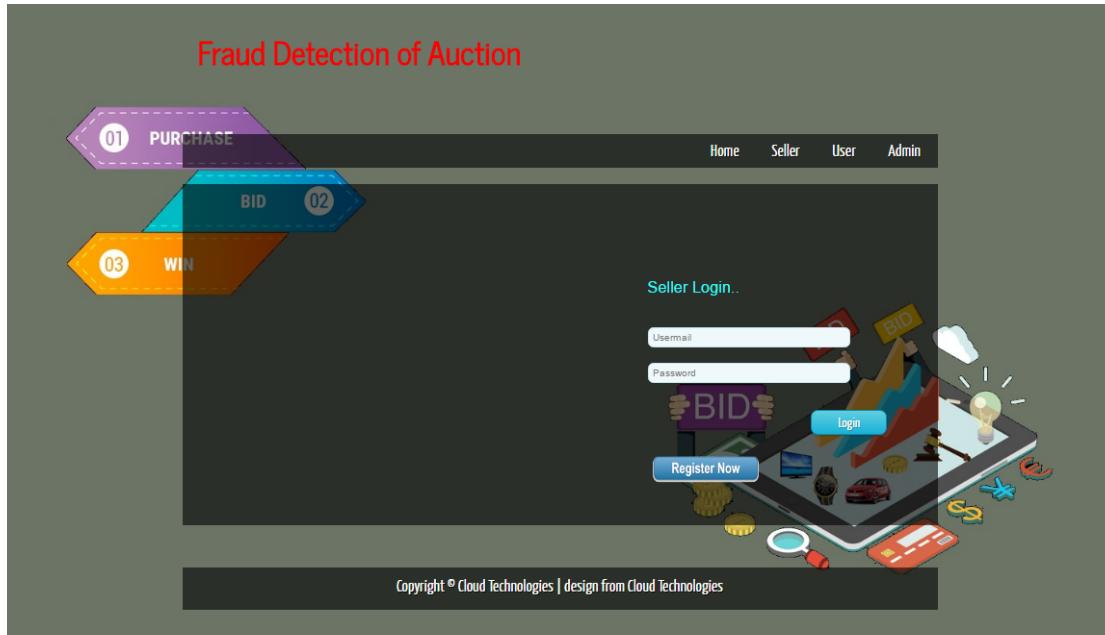


Figure 6-6Login Screen for Seller

Description: The application has an User login which is used to start the application to prevent authenticity infringement.

6.2.7 Add Product

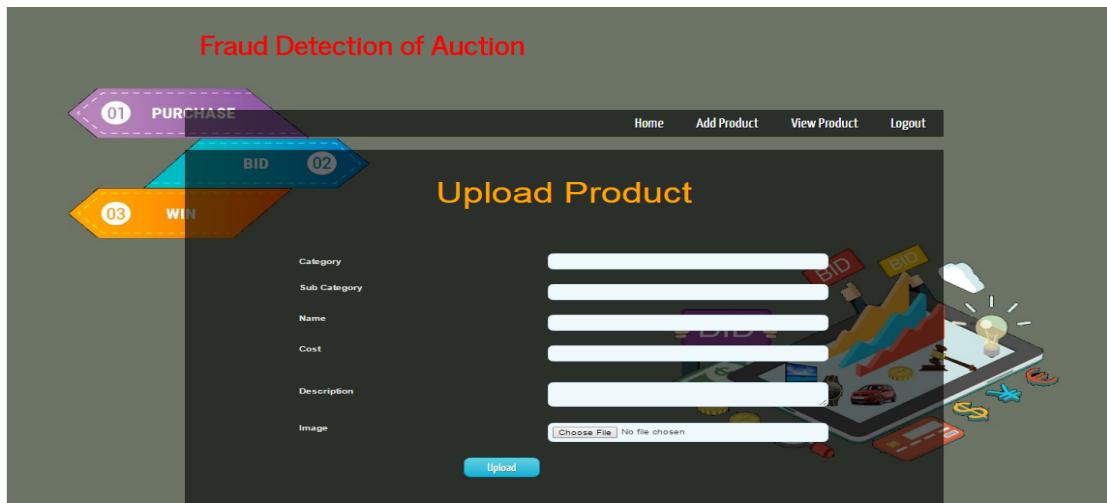


Figure 6-7Add Product

Description: The Seller can add their product to auction.

6.2.8 Admin Product Request

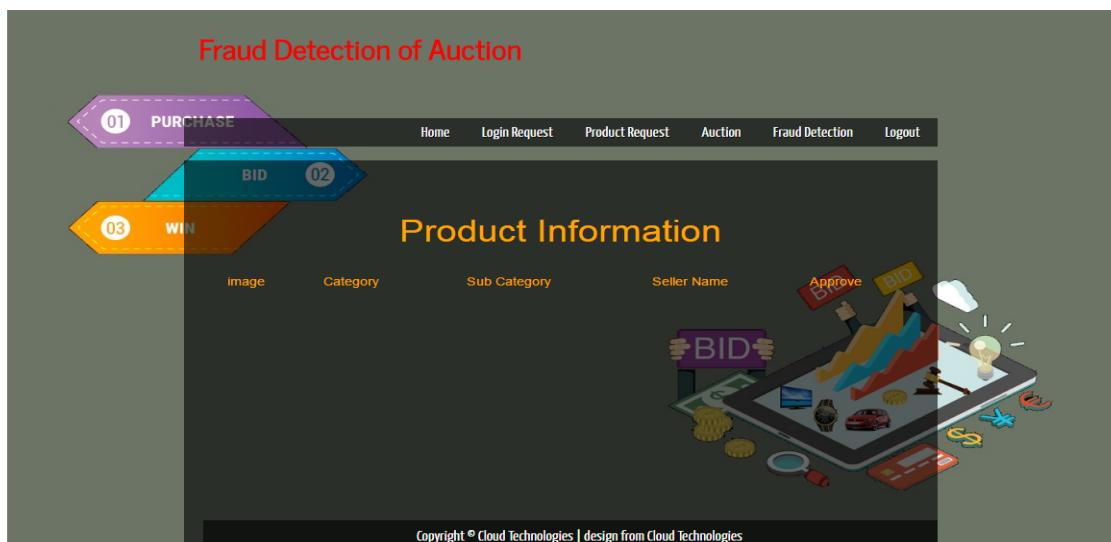


Figure 6-8Admin Product Request

Description: The admin can check the product details and take action as either accept or reject.

6.2.9 View Added Product



Figure 6-9 View Added Product

Description: Seller can view their products and status.

6.2.10 User Product View

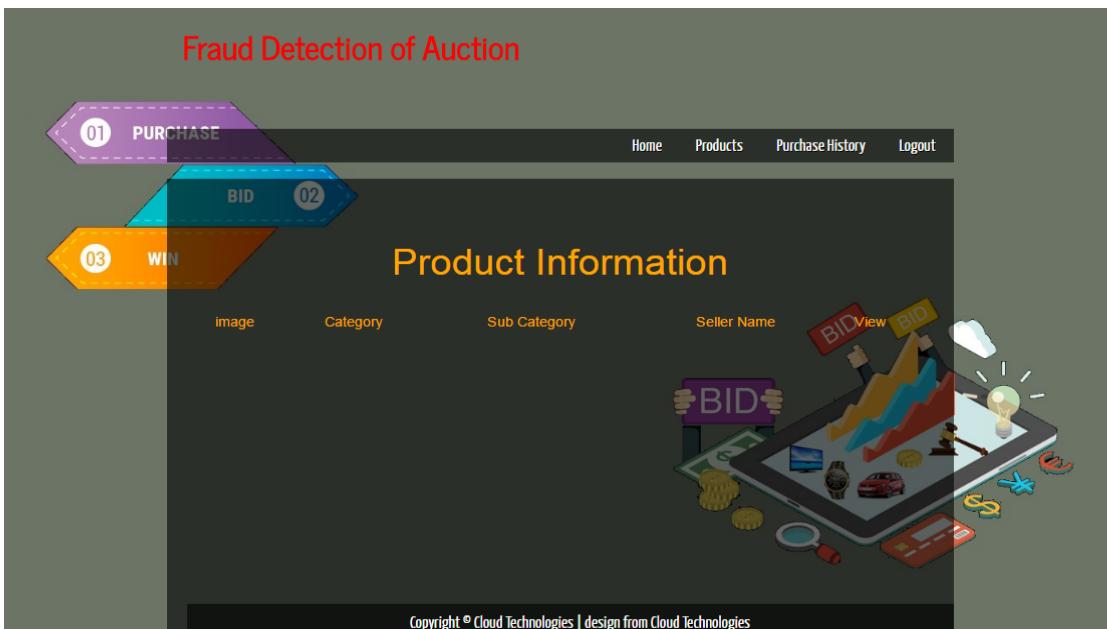


Figure 6-10 User Product View

Description: The Login User can see the all products and add his price to the product.

6.2.11 Admin Auction page

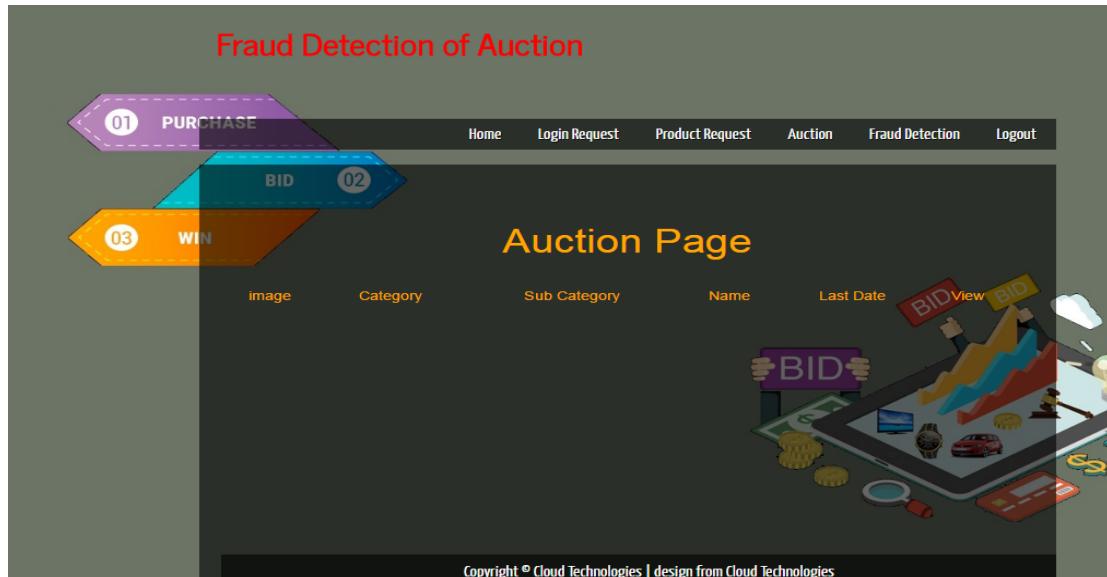


Figure 6-11 Admin Auction page

Description: Admin can see the all auction price and product send to that highest auction price buyer(user) .

6.2.12 User Purchase History

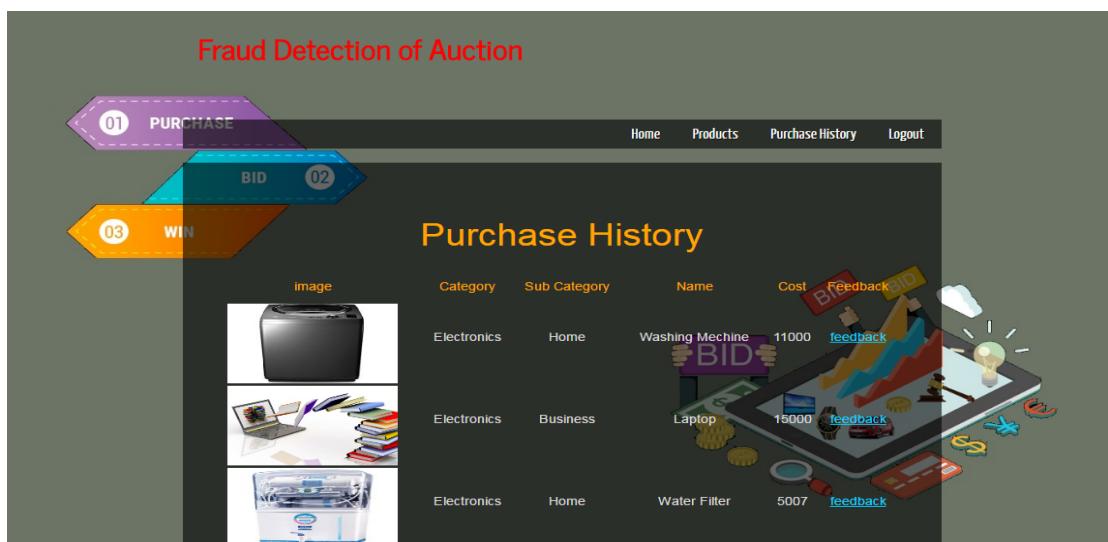


Figure 6-12 User Purchase History

Description: User can See the Purchase history and give the feedback for product about that quality and etc.. .

6.2.13 Fraud Detection

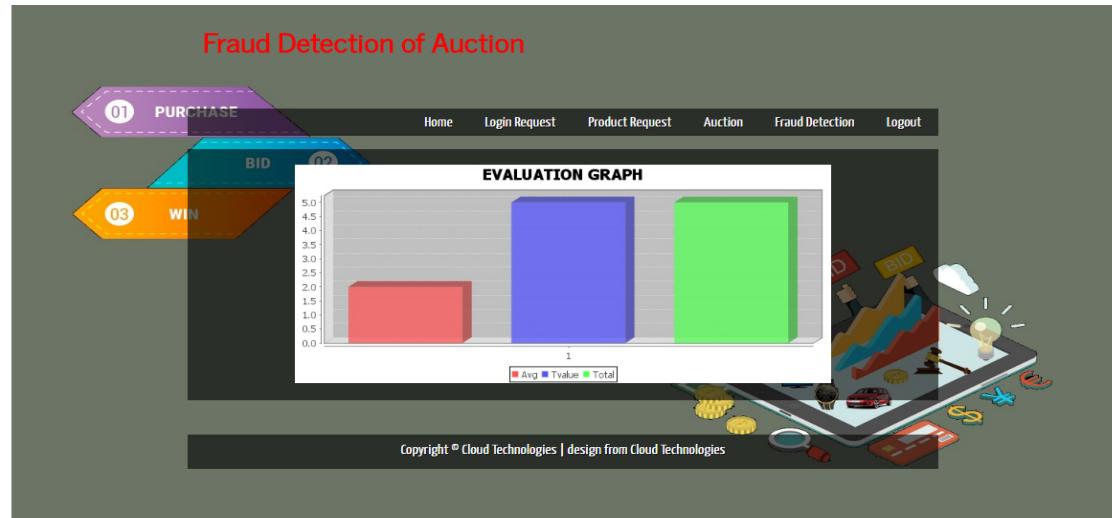


Figure 6-13 Fraud Detection

Description: Admin can find the fraud seller by using users feedback.

7. TESTING

7.1 Software Testing

Software testing is the process of validating and verifying that a software application meets the technical requirements which are involved in its design and development. It is also used to uncover any defects/bugs that exist in the application. It assures the quality of the software. There are many types of testing software viz., manual testing, unit testing, black box testing, performance testing, stress testing, regression testing, white box testing etc. Among these performance testing and load testing are the most important one for an android application and next sections deal with some of these types.

7.2 Black box Testing

Black box testing treats the software as a "black box"—without any knowledge of internal implementation. Black box testing methods include: equivalence partitioning, boundary value analysis, all-pairs testing, fuzz testing, model-based testing, traceability matrix, exploratory testing and specification-based testing.

7.3 White box Testing

White box testing is when the tester has access to the internal data structures and algorithms including the code that implement these.

7.4 Performance Testing

Performance testing is executed to determine how fast a system or sub-system performs under a particular workload. It can also serve to validate and verify other quality attributes of the system such as scalability, reliability and resource usage.

7.5 Load Testing

Load testing is primarily concerned with testing that can continue to operate under specific load, whether that is large quantities of data or a large number of users.

7.6 Manual Testing

Manual Testing is the process of manually testing software for defects. Functionality of this application is manually tested to ensure the correctness. Few examples of test case for Manual Testing are discussed later in this chapter.

Test Case 1	
Test Case Name	Empty login fieldstesting
Description	In the login screen if the username and password fields are empty
Output	Login fails showing an alert box asking to enter username and password.

Figure 7-1 Test Case for Empty Login Fields

Test Case 2	
Test Case Name	Wrong login fields testing
Description	A unique username and password are set by administrator. On entering wrong username or password gives.
Output	Login fails showing an alert box username or password incorrect.

Test Case 3	
Test Case Name	Add to MySql testing
Description	Entity code and Name are mandatory fields. If left empty.
Output	Insert Fails and a toast message appears asking to enter entity code and name.

Test Case 4	
Test Case Name	Data Storing Testing
Description	Testing at database server side weather data is stored or not .
Output	n rows effeced

8. CONCLUSION

8.1 Conclusions

In this paper we build online models for the auction fraudmoderation and detection system designed for a major Asianonline auction website. By empirical experiments on a realword online auction fraud detection data, we show that ourproposed online probit model framework, which combinesonline feature selection, bounding coefficients from expertknowledge and multiple instance learning, can significantly improve over baselines and the human-tuned model. Notethat this online modeling framework can be easily extendedto many other applications, such as web spam detection,content optimization and so forth.Regarding to future work, one direction is to include theadjustment of the selection bias in the online model trainingprocess. It has been proven to be very effective for offlinemodels . The main idea there is to assume all theunlabeled samples have response equal to 0 with a very smallweight. Since the unlabeled samples are obtained from aneffective moderation system, it is reasonable to assume thatwith high probabilities they are non-fraud. Another futurework is to deploy the online models described in this paperto the real production system, and also other applications.

8.2BIBLIOGRAPHY

- [1] D. Agarwal, B. Chen, and P. Elango. Spatio-temporalmodels for estimating click-through rate. InProceedings of the 18th international conference onWorld wide web, pages 21–30. ACM, 2009.
- [2] S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instancelearning. Advances in neural information processingsystems, pages 577–584, 2003.
- [3] C. Bliss. The calculation of the dosage-mortalitycurve. Annals of Applied Biology, 22(1):134–167, 1935.
- [4] A. Borodin and R. El-Yaniv. Online computation andcompetitive analysis, volume 53. Cambridge UniversityPress New York, 1998.
- [5] L. Breiman. Random forests. Machine learning,45(1):5–32, 2001.
- [6] R. Brent. Algorithms for minimization withoutderivatives.Dover Pubns, 2002.[7] D. Chau and C. Faloutsos. Fraud detection inelectronic auction. In European Web Mining Forum(EWMF 2005), page 87.
- [8] H. Chipman, E. George, and R. McCulloch. Bart:Bayesian additive regression trees. The Annals of Applied Statistics, 4(1):266–298, 2010.
- [9] W. Chu, M. Zinkevich, L. Li, A. Thomas, and B. Tseng. Unbiased online active learning in datastreams. In Proceedings of the 17th ACM SIGKDDinternational conference on Knowledge discovery anddata mining, pages 195–203. ACM, 2011.[10] C. Chua and J. Wareham. Fighting internet auctionfraud: An assessment and proposal. Computer,37(10):31–37, 2004.
- [11] R. Collins, Y. Liu, and M. Leordeanu. Online selectionof discriminative tracking features. IEEE Transactionson Pattern Analysis and Machine Intelligence, pages 1631–1643, 2005.
- [12] N. Cristianini and J. Shawe-Taylor. An introduction to support Vector Machines: and other kernel-basedlearning methods. Cambridge university press, 2006.
- [13] T. Dietterich, R. Lathrop, and T. Lozano-P’erez. Solving the multiple instance problem withaxis-parallel rectangles. Artificial Intelligence,89(1-2):31–71, 1997.
- [14] Federal Trade Commission. Internet auctions: A guidefor buyers and sellers. <http://www.ftc.gov/bcp/conline/pubs/online/auctions.htm>, 2004.

WWW 2012 – Session: Security 2 April 16–20, 2012, Lyon, France677

- [15] J. Friedman. Stochastic gradient boosting. Computational Statistics & Data Analysis, 38(4):367–378, 2002.