

Fall 12-1-2009

Peripheral Trouble Shooting and Management

Naveen Malladi
Dakota State University

Follow this and additional works at: <https://scholar.dsu.edu/theses>

Recommended Citation

Malladi, Naveen, "Peripheral Trouble Shooting and Management" (2009). *Masters Theses*. 167.
<https://scholar.dsu.edu/theses/167>

This Thesis is brought to you for free and open access by Beadle Scholar. It has been accepted for inclusion in Masters Theses by an authorized administrator of Beadle Scholar. For more information, please contact repository@dsu.edu.

PERIPHERAL TROUBLE SHOOTING AND MANAGEMENT

A graduate project submitted to Dakota State University in partial fulfillment

Of the requirements for the degree of

Master of Science

In

Information Systems

December, 2009

By

Naveen Malladi

Project Committee:

Dr. Ronghua Shan

Dr. Wayne Pauli

Dr. Steven Graham



PROJECT APPROVAL FORM

We certify that we have read this project and that, in our opinion, it is satisfactory in scope and quality as a project for the degree of Master of Science in Information Systems.

Student Name: Naveen Malladi

Master's Project Title: Peripheral Troubleshooting And Management

Faculty supervisor: Ronghua Shan Date: 12/17/2009

Committee member: Wayne Paul Date: 12/17/09

Committee member: A.K. Gul Date: 12/17/09

ABSTRACT

Advances in technology and the growth and use of internet in different domains has opened new ways of addressing and finding a far more better Solution to problems in different domains.

One such domain is addressing to Peripheral Trouble Shooting and Management System of computers and other sharable peripherals connected to a given network

The use of interconnected shared peripherals in a network has become a norm in all huge, small, private, government corporate, non corporate organizations ,especially in huge organizations where the number of employees who use these shared peripherals is quite large allocation of these peripherals and subsequent tracking and reporting & management becomes a cumbersome ,complicated and inefficient .And in organization where timelines play a very vital role immediate handling of all trouble shooting becomes essential, A physical communication of this problems rather becomes too time consuming and is prone to be inefficient. Thereby having a management system which allows centralized control and management becomes rather essential.

The need of the hour is to implement an online control and reporting system is for maintaining the peripherals in the network of a huge corporate organization. In such a system the entire peripheral allotment and subsequent reporting should be centralized by implementing a complete control from a central controlling authority say an administrator who is responsible for allocating different shared peripheral including computers to the organization employees, the employees in turn can lodge their complaints online, in turn the administrator can assign the trouble shooting rectifications to the concerned engineers

DECLARATION

I hereby certify that this project constitutes my own product, that where the language of others is set forth, quotation marks so indicate, and that appropriate credit is given where I have used the language, ideas, expressions or writings of another.

I declare that the project describes original work that has not previously been presented for the award of any other degree of any institution.

Signed,

Naveen Malladi

TABLE OF CONTENTS

PROJECT APPROVAL FORM	ii
ABSTRACT	iii
DECLARATION	iv
TABLE OF CONTENTS	v
LIST OF TABLES	viii
LIST OF FIGURES	ix
INTRODUCTION	1
1. Goals of the System	3
2. Need for the Computerized System	4
3. Automation	5
3.1 Advantages of Speed and Accuracy	5
3.2 Requirement Specification	5
3.3 System allocations	5
3.4 System complaints	5
3.5 Stock details	6
4. MODULES	7
5. MODULE DESCRIPTION	8
6. TECHNOLOGY	9
7. OVERVIEW OF JAVA TECHNOLOGY	10
7.1 JAVA Environment	11
7.2 JAVA Architecture	11
7.3 JAVA Virtual Machine	11
7.4 Paradigm of JAVA	11
8. ABOUT HTML	12
9. Java Data Base Connectivity (JDBC)	13
9.1 Result set enhancements	13
9.2 Batch updates	13
9.3 Advanced data types	13
9.4 Row sets	14
9.5 JNDI for naming databases	14
9.6 Connection Pooling	14

9.7	Distributed transaction support	14
9.8	Other new features	15
9.9	Result Set Enhancements	15
9.10	Scrolling	15
9.11	Performance	16
10.	Introduction to Servlets	17
11.	What is a Servlet?	18
12.	Use Servlets instead of CGI Scripts	19
12.1	The Servlets Interface	19
12.2	Client Interaction	20
12.3	The ServletRequest Interface	20
12.4	The ServletResponse Interface	20
12.5	Additional Capabilities of HTTP Servlets	21
12.6	A Simple Servlet	21
13.	Servlet Lifecycle	24
14.	Initializing a Servlet	25
14.1	Initialization Parameters	25
14.2	Destroying a Servlet	26
15.	Servlet-client Interaction	27
15.1	Handling HTTP Clients	27
15.2	Requests and Responses	27
15.3	HttpServletRequest Objects	27
15.4	To access client data:	27
15.5	HttpServletResponse Objects	28
15.6	HTTP Header Data	28
15.7	Handling GET and POST Requests	29
16.	Writing Your First Servlet	30
16.1	About Session Tracking	30
16.2	Server-Side Session Objects and Users	31
16.3	Session Invalidation	31
16.4	Multiple Servlets	32
	CONCLUSION	69

BIBLIOGRAPHY	70
WEB SITE REFERENCES	71

LIST OF TABLES

Table 1: Login.....	45
Table 2: Employee	45
Table 3: Division.....	46
Table 4: Department	46
Table 5: Pc details.....	47
Table 6: Pr details	47
Table 7: User master	48
Table 8: Supplier.....	49
Table 9: Contract.....	49
Table 10: Engineer.....	50
Table 11: Complaint	50

LIST OF FIGURES

Figure 1: Servlet Architecture.....	18
Figure 2: Servlet Interface	19
Figure 3: Servlet Lifecycle.....	24
Figure 4: CONTEXT LEVEL DIAGRAMS	33
Figure 5: E-R DIAGRAMS	35
Figure 6: USE CASES	40
Figure 7: Screen shots	Error! Bookmark not defined.

INTRODUCTION

In the project we are implementing centralized controlling system on resources and peripherals of computers, which are connected in a LAN.

In the proposed system we are implementing online complaint posting and checking the status of the complaint, which was posted by user (i.e. any of the staff in an organization). In this system each user is provided with unique login ID. Each user makes a login using his ID if any complaints are there to be made regarding computer peripherals.

When logged as an administrator using administrator login. Administrator has complete control on allocating the resources, complaint clearance, Enquiries and Reports.

The user has no permissions to access any of the administration part he is restricted only to post and check status of complaint he made.

Administrator when logged takes view of the complaint which is made earlier which are pending, new complaints which are made by users and clear complaints which are to be cleared.

In complaint clearance he put the engineers to work, Administrator assigns work for engineer. It is also the duty of administrator to take a glance at adding the new user information, computers information and engineer information. Administrator should also have to maintain the database of the supplier who supplies machines to organization and contract information.

Administrator also maintains report of activities which are performed by him and team of engineers.

On the whole by providing the proposed system to client we make sure that there will be a centralized control for administrator, more user friendly, efficient usage of the technical skill and usage of new technology available.

The new system developed is a GUI (Graphical User Interface). HTML is used as front-end and ORACLE or any other RDBMS as data storage. This system is developed with an eye towards the future and can easily be updated with new modules, as it may need expansion at a later stage i.e., data transparency has been maintained to an extent.

The importance of new system is that it is user friendly and has a better interface with users working on it. It can overcome the problems of manual system as many users can access simultaneously and the security problem is solved as it prevents the unauthorized users to access the system.

1. Goals of the System

- To have a centralized control and management of all the shared peripherals in a given network
- Efficiently addressing to all the troubleshooting of the shared resources in a given network
- The need to have a personnel interaction for trouble shoot reporting and subsequent follow up must be eradicated and all issues must be handled by the system, thereby helping those organizations where the network is spread across a geographical domain
Contracts and subcontracts management with various concerned agencies must be handled by the System.
- Design a more secure system where unauthorized access to resources is totally eradicated by implementing an appropriate authentication mechanism.
- Generating reports about various managerial aspects such as allocation of resources, enquiry reports etc.

2. Need for the Computerized System

- The need for computerized system felt because of following inefficiencies in the existing system:
- Existing system is subjected to all types of manual errors.
- It is inefficient in speedy retrieval of data from the database. For example just to find out how many PC's a particular person is main the administrator has to scan through the entire database.
- It is cumbersome for example suppose the location of the system has been changed thus the new IP address has to be assigned to that particular PC. Cutting down that entry and writing does it over that. These kinds of changes, which lead to changes in, more than one table, may create confusion if there is, some mistake.
- User identity is not checked while a user is registering a complaint on
- User is not sure of the status of his complaint
- After the complaint has been corrected there is no provision so that the user can give his feedback about the nature of services rendered by the Engineer.

Thus for the above reason it is clear that automation of the existing manual system would be of immense help to the user.

3. Automation

3.1 Advantages of Speed and Accuracy

Machines are supposed to be faster and accurate than human beings. Machines can work continuously hours together where as human beings are bound to make mistakes more hours and after hour. A computer system provides up to date information to management with the help of managerial decisions can be made. The speed and memory of human beings are limited. The mechanized data processing helps in these directions without much difficulty because millions of operations are done within few seconds with the help of the computer. Therefore ultimately it is feasible to go for the process of computerization.

3.2 Requirement Specification

The objective of the project is to provide the administrator and the user an environment where the administrator can allocate systems and know the current status of the database. The user can register the complaint. The major sections of the applications can be classified as follows

3.3 System allocations

- Allocation
- Enquiries
- Reports

3.4 System complaints

- New complaints
- Complaint status

3.5 Stock details

- PC entry
- Printer entry

It is decided that this project is to be programmed using Java Servlets at the business logic end and HTML as a Client Tier, hence the design is architecturally neutral and helps the programmer design with ease.

Since Java is Object Oriented, the project adheres to OOAD design paradigm.

4. MODULES

➤ Administration

- Allocation of resources
- Complaint handling through assigning work to engineers
- Creation of reports for managerial use

➤ Employee interactions

5. MODULE DESCRIPTION

Administration:

- Complaint Handling

- Allocation
 - Work for Engineers
 - Sharable peripherals to employees

- Adding of details
 - Sharable peripheral details
 - Employee details
 - Contract management details

- Enquires
 - Resource Enquires
 - Dept Enquires

- Reports
 - Resource Details
 - Issue Details

Employee:

- Complaint Posting

- Status Acquiring

6. TECHNOLOGY

Technology Used:

- Java
- Servlets
- Oracle

Operating System:

- Any Operating System

7. OVERVIEW OF JAVA TECHNOLOGY

Java, whether you love it, or hate it, it's here to stay. Like everyone's favorite language C, Java has had a major impact on the computing scene. When the history of computers is written, its name will be up there with the stars.

If you were to choose just one language to learn today, it should be Java. It's being pushed aggressively by Sun and is growing by leaps and bounds. There are lots of Java programmers out there and more join the party every day.

Java started out as a bit of an accident. A team under Bill Joy was working at Sun on a new programming language for embedded applications. Java was originally expected to work in toasters and fridges, not on modern computers! The initial prognosis for Java was not good and it was only the rise of the Internet which saved Java from oblivion. Since then, neither the Net nor Sun nor Java has looked back and all have grown from strength to strength.

World Wide Web is an open ended information retrieval system designed to be used in the distributed environment. This system contains web pages that provide both information and controls. We can navigate to a new web page in any direction. This is made possible with HTML java was meant to be used in distributed environment such as internet. So java could be easily incorporated into the web system and is capable of supporting animation graphics, games and other special effect. The web has become more dynamic and interactive with support of java. We can run a java program on remote machine over internet with the support of web.

7.1 JAVA Environment

Java environment includes a large number of tools which are part of the system known as java development kit (JDK) and hundreds of classes, methods, and interfaces grouped into packages forms part of java standard library (JSL).

7.2 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 are then interpreted on each platform by the runtime environment. Java also provides stringent compile and runtime checking and automatic memory management in order to ensure solid code.

7.3 JAVA Virtual Machine

When we compile the code, java compiler creates machine code (byte code) for a hypothetical machine called java virtual machine (jvm). The jvm will execute the byte code and overcomes the issue of portability. The code is written and compile for one machine and interpreted all other machines. This machine is called java virtual machine.

7.4 Paradigm of JAVA

- Dynamic down loading applets (small application programs);
- Elimination of flatware phenomenon that is providing those features of a product that user needs at a time. The remaining features of a product can remain in the server.
- Changing economic model of the software
- Up-to-date software availability
- Supports network entire computing

8. ABOUT HTML

HTML (hyper text markup language) is a language used to create hyper text documents that have hyper links embedded in them. It consists of tags embedded in the text of a document with HTML. We can build web pages or web documents. It is basically a formatting language and not a programming language. The browser reading the document interprets mark up tags to help format the document for subsequent display to a reader. HTML is a language for describing structured documents. HTML is a platform independent. WWW (World Wide Web) pages are written using HTML. HTML tags control in part the representation of the WWW page when view with web browser. The browser interprets HTML tags in the web document and displays it. Different browsers show data differently. Examples of browsers used to be web pages include:

- Netscape
- Internet Explorer

9. Java Data Base Connectivity (JDBC)

Overview of New Features

9.1 Result set enhancements

The JDBC 1.0 API provided result sets that had the ability to scroll in a forward direction. Scrollable result sets allow for more flexibility in the processing of results by providing both forward and backward movement through their contents. In addition, Scrollable result sets allow for relative and absolute positioning. For example, it's possible to move to the fourth row in a scrollable result set directly, or to move directly to the third row following the current row provided the row exists. The JDBC API allows result sets to be directly updatable, as well.

9.2 Batch updates

The batch update feature allows an application to submit multiple update statements (insert/update/delete) in a single request to the database. This can provide a dramatic increase in performance when a large number of update statements need to be executed.

9.3 Advanced data types

Increased support for storing persistent Java programming language objects (Java objects). And a mapping for SQL99 data types such as binary large objects, and structured Types, has been added to the JDBC API. An application may also customize the mapping of SQL99 structured types into Java programming language classes.

9.4 Row sets

As its name implies, a row set encapsulates a set of rows. A row set may or may not maintain an open database connection. When a row set is 'disconnected' from its data source, updates performed on the row set are propagated to the underlying database using an optimistic concurrency control algorithm. Row sets add support to the JDBC API for the JavaBeans component model. A row set object is a bean. A row set implementation may be serializable. Row sets can be created at design time and used in conjunction with other JavaBeans components in a visual builder tool to construct an application.

9.5 JNDI for naming databases

The Java Naming and Directory Interface (JNDI) API can be used in addition to a JDBC technology-based driver manager (JDBC driver manager) to obtain a connection to a database. When an application uses the JNDI API, it specifies a logical name that identifies a particular database instance and JDBC driver for accessing that database.

9.6 Connection Pooling

The JDBC API contains 'hooks' that allow connection pooling to be implemented on top of the JDBC driver layer. This allows for a single connection cache that spans the different JDBC drivers that may be in use. Since creating and destroying database connections is expensive, connection pooling is important for achieving good performance, especially for server applications.

9.7 Distributed transaction support

Support for distributed transactions has been added as an extension to the JDBC API.

This feature allows a JDBC driver to support the standard 2-phase commit protocol used by the Java Transaction Service (JTS) API.

9.8 Other new features

Support for character streams has been added. This means that character data can be retrieved and sent to the database as a stream of internationalized Unicode characters.

Methods to allow `java.math.BigDecimal` values to be returned with full precision have also been added. Support for time zones has been added.

9.9 Result Set Enhancements

This chapter discusses the new functionality that has been added to result sets. The goal of the enhancements is to add two new basic capabilities to result sets: scrolling and updatability. Several methods have also been added to enable a JDBC driver to deliver improved performance when processing results. A variety of examples are included to illustrate the new features.

9.10 Scrolling

A result set created by executing a statement may support the ability to move backward (last-to-first) through its contents, as well as forward (first-to-last). Result sets that support this capability are called scrollable result sets. Result sets that are scrollable also support relative and absolute positioning. Absolute positioning is the ability to move directly to a row by specifying its absolute position in the result set, while relative positioning gives the ability to move to a row by specifying a position that is relative to the current row. The definition of absolute and relative positioning in the JDBC API is modeled on the X/Open SQL CLI specification.

9.11 Performance

Two performance hints may be given to a JDBC 2.1 technology-enabled driver to make access to result set data more efficient. Specifically, the number of rows to be fetched from the database each time more rows are needed can be specified, and a direction for processing the rows forward, reverse, or unknown can be given as well. These values can be changed for an individual result set at any time. A JDBC driver may ignore a performance hint if it chooses.

10. Introduction to Servlets

Servlets provide a Java(TM)-based solution used to address the problems currently associated with doing server-side programming, including inextensible scripting solutions, platform-specific APIs, and incomplete interfaces.

Servlets are objects that conform to a specific interface that can be plugged into a Java-based server. Servlets are to the server-side what applets are to the client-side - object byte codes that can be dynamically loaded off the net. They differ from applets in that they are faceless objects (without graphics or a GUI component). They serve as platform-independent, dynamically-loadable, pluggable helper byte code objects on the server side that can be used to dynamically extend server side functionality.

11. What is a Servlet?

Servlets are modules that extend request/response-oriented servers, such as Java enabled web servers. For example, a Servlets might be responsible for taking data in an HTML order-entry form and applying the business logic used to update a company's order database.

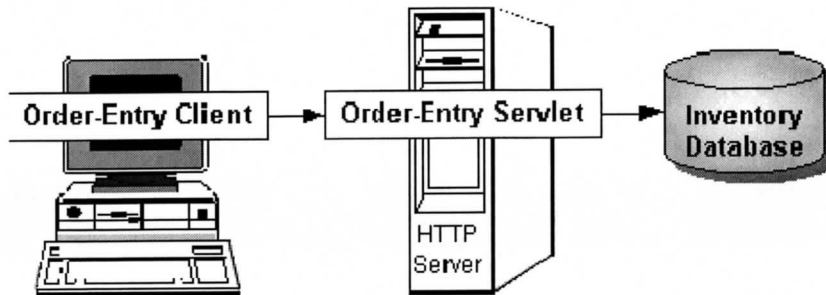


Figure 1: Servlet Architecture

Servlets are to servers what applets are to browsers. Unlike applets, however, Servlets have no graphical user interface.

Servlets can be embedded in many different servers because the Servlets API, which you use to write Servlets, assumes nothing about the server's environment or protocol. Servlets have become most widely used within HTTP servers; many web servers support the Servlets API.

12. Use Servlets instead of CGI Scripts

Servlets are an effective replacement for CGI scripts. They provide a way to generate dynamic documents that is both easier to write and faster to run. Servlets also address the problem of doing server-side programming with platform specific APIs: they are developed with the Java Servlets API, a standard Java extension.

So use Servlets to handle HTTP client requests. For example, have Servlets process data posted over HTTPS using an HTML form, including purchase order or credit card data. A Servlets like this could be part of an order-entry and processing system, working with product and inventory databases, and perhaps an on-line payment system.

12.1 The Servlets Interface

The central abstraction in the Servlets API is the Servlets interface. All Servlets implement this interface, either directly or, more commonly, by extending a class that implements it such as `HttpServlet` or `GenericServlet`.

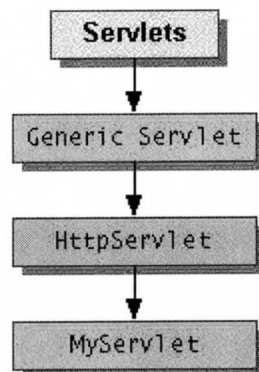


Figure 2: Servlet Interface

The Servlets interface declares, but does not implement, methods that manage the Servlets and its communications with clients. Servlets writers provide some or all of these methods when developing a Servlets.

12.2 Client Interaction

When a Servlets accepts a call from a client, it receives two objects:

- A ServletRequest, which encapsulates the communication from the client to the server
- A ServletResponse, which encapsulates the communication from the Servlets back to the client
- ServletRequest and ServletResponse are interfaces defined by the javax.Servlet package.

12.3 The ServletRequest Interface

The ServletRequest interface allows the Servlet access to:

- Information such as the names of the parameters passed in by the client, the protocol (scheme) being used by the client, and the names of the remote host that made the request and the server that received it.
- The input stream, ServletInputStream. Servlets use the input stream to get data from clients that use application protocols such as the HTTP POST and PUT methods.
- Interfaces that extend ServletRequest interface allow the Servlet to retrieve more protocol-specific data. For example, the HttpServletRequest interface contains methods for accessing HTTP specific header information.

12.4 The ServletResponse Interface

The ServletResponse interface gives the Servlet methods for replying to the client. It:

- Allows the Servlet to set the content length and MIME type of the reply.
- Provides an output stream, `ServletOutputStream`, and a `Writer` through which the Servlet can send the reply data.
- Interfaces that extend the `ServletResponse` interface give the Servlet more protocol-specific capabilities. For example, the `HttpServletResponse` interface contains methods that allow the Servlet to manipulate HTTP-specific header information.

12.5 Additional Capabilities of HTTP Servlets

The classes and interfaces described above make up a basic Servlet. HTTP Servlets have some additional objects that provide session-tracking capabilities. The Servlet writer can use these APIs to maintain state between the Servlet and the client that persists across multiple connections during some time period. HTTP Servlets also have objects that provide cookies. The Servlet writer uses the cookie API to save data with the client and to retrieve this data.

12.6 A Simple Servlet

The following class completely defines Servlet:

```
Public class SimpleServlet extends HttpServlet's
```

```
{
```

```
/**
```

```
* Handle the HTTP GET method by building a simple web page.
```

```
*/
```

```
Public void doGet (HttpServletRequest request,
```

HttpServletResponse response)

Throws ServletException, IOException

```
{  
  
    Print Writer out;  
  
    String title = "Simple Servlet Output";  
  
    // set content type and other response header fields first  
  
    response.setContentType ("text/html");  
  
    // then write the data of the response  
  
    Out = response.getWriter ();  
  
    out.println ("<HTML><HEAD><TITLE>");  
  
    out.println (title);  
  
    out.println ("</TITLE></HEAD><BODY>");  
  
    out.println ("<H1>" + title + "</H1>");  
  
    out.println ("<P>this is output from SimpleServlet.");
```



```
out.println ("</BODY></HTML>");

out.close ();

}

}
```

The classes mentioned in the Architecture of the Servlet Package section are shown in the example in bold:

- SimpleServlet extends the **HttpServlet**'s class, which implements the Servlet interface.
- Within the doGet method,
 - The user's request is represented by an **HttpServletRequest** object.
 - The response to the user is represented by an **HttpServletResponse** object.
 - Because text data is returned to the client, the reply is sent using the **Writer** object obtained from the **HttpServletResponse** object.

13. Servlet Lifecycle

Each Servlet has the same life cycle:

- A server loads and initializes the Servlet
- The Servlet handles zero or more client requests
- The server removes the Servlet

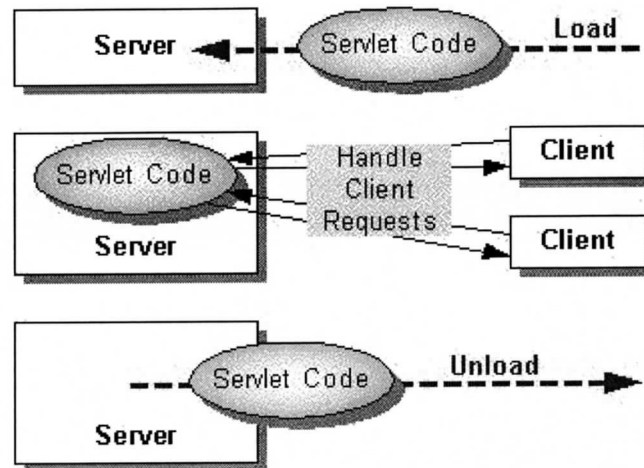


Figure 3: Servlet Lifecycle

14. Initializing a Servlet

When a server loads a Servlet, the server runs the Servlets init method. Initialization completes before client requests are handled and before the Servlet is destroyed.

Even though most Servlets are run in multi-threaded servers, Servlets have no concurrency issues during Servlet initialization. The server calls the init method once, when the server loads the Servlet, and will not call the init method again unless the server is reloading the Servlet. The server cannot reload a Servlet until after the server has destroyed the Servlet by calling the destroy method.

The init method provided by the HttpServlet's class initializes the Servlet and logs the initialization. To do initialization specific to your Servlet, override the init () method following these rules:

If an initialization error occurs that renders the Servlet incapable of handling client requests, throw an Unavailable Exception.

14.1 Initialization Parameters

The second version of the init method calls the getInit Parameter method. This method takes the parameter name as an argument and returns a String representation of the parameter's value.

The specification of initialization parameters is server-specific. In the Java Web Server, the parameters are specified with a Servlet is added then configured in the Administration Tool. For an explanation of the Administration screen where this setup is performed, see the Administration Tool: Adding Servlets online help document.

If, for some reason, you need to get the parameter names, use the `getParameterNames` method.

14.2 Destroying a Servlet

Servlets run until the server destroys them, for example at the request of a system administrator. When a server destroys a Servlet, the server runs the Servlets `destroy` method. The method is run once; the server will not run that Servlet again until after the server reloads and reinitializes the Servlet.

When the `destroy` method runs, another thread might be running a service request. The [Handling Service Threads at Servlet Termination](#) section shows you how to provide a clean shutdown when there could be long-running threads still running service requests.

15. Servlet-client Interaction

15.1 Handling HTTP Clients

An HTTP Servlet handles client requests through its service method. The service method supports standard HTTP client requests by dispatching each request to a method designed to handle that request. For example, the service method calls the `doGet` method shown earlier in the simple example Servlet.

15.2 Requests and Responses

Methods in the `HttpServlet`'s class that handle client requests take two arguments:

- An `HttpServletRequest` object, which encapsulates the data from the client
- An `HttpServletResponse` object, which encapsulates the response to the client

15.3 HttpServletRequest Objects

An `HttpServletRequest` object provides access to HTTP header data, such as any cookies found in the request and the HTTP method with which the request was made. The `HttpServletRequest` object also allows you to obtain the arguments that the client sent as part of the request.

15.4 To access client data:

- The `getParameter` method returns the value of a named parameter. If your parameter could have more than one value, use `getParameterValues` instead. The `getParameterValues` method returns an array of values for the named parameter. (The method `getParameterNames` provides the names of the parameters.)

- For HTTP GET requests, the `getQueryString` method returns a String of raw data from the client. You must parse this data yourself to obtain the parameters and values.
- For HTTP POST, PUT, and DELETE requests,
 - If you expect text data, the `getReader` method returns a Buffered Reader for you to use to read the raw data.
 - If you expect binary data, the `getInputStream` method returns a ServletInputStream for you to use to read the raw data

Note: Use either a `getParameter [Values]` method or one of the methods that allow you to parse the data yourself. They cannot be used together in a single request.

15.5 HttpServletResponse Objects

An HttpServletResponse object provides two ways of returning data to the user:

- The `getWriter` method returns a Writer
- The `getOutputStream` method returns a ServletOutputStream

Use the `getWriter` method to return text data to the user, and the `getOutputStream` method for binary data.

Closing the Writer or ServletOutputStream after you send the response allows the server to know when the response is complete.

15.6 HTTP Header Data

You must set HTTP header data before you access the Writer or Output Stream. The HttpServletResponse class provides methods to access the header data. For example, the `setContentType` method sets the content type. (This header is often the only one manually set.)

15.7 Handling GET and POST Requests

The methods to which the service method delegates HTTP requests include,

- doGet, for handling GET, conditional GET, and HEAD requests
- doPost, for handling POST requests
- doPut, for handling PUT requests
- doDelete, for handling DELETE requests

By default, these methods return a BAD_REQUEST (400) error. Your Servlet should override the method or methods designed to handle the HTTP interactions that it supports. This section shows you how to implement methods that handle the most common HTTP requests: GET and POST.

The HttpServlet's service method also calls the doOptions method when the Servlet receives an OPTIONS request and doTrace when the Servlet receives a TRACE request. The default implementation of doOptions automatically determines what HTTP options are supported and returns that information. The default implementation of doTrace causes a response with a message containing all of the headers sent in the trace request. These methods are not typically overridden.

16. Writing Your First Servlet

Servlets are also easy to develop. This document discusses the following minimum steps needed to create any Servlet:

- Write the Servlet
 - Import the necessary Java packages
 - Inherit from Generic Servlet or the HTTP convenience class `HttpServlet`
 - Override the service method (this is where the actual work is done by the Servlet)
 - Save the file with a `.java` filename extension
- Compile the Servlet
 - Make sure `jws.jar` is included in your class path
 - Invoke `javac`
- Install the Servlet
 - Use the Java Web Server's Administration Tool to install it, and optionally configure it.
- Test the Servlet
 - Invoke the Servlet from a JDK1.1-compatible browser.

16.1 About Session Tracking

Session Tracking is a flexible, lightweight mechanism that enables stateful programming on the web. Its general implementation serves as a basis for more sophisticated state models, such as persistent user profiles or multi-user sessions.

A session is a series of requests from the same user that occur during a time period. This transaction model for sessions has many benefits over the single-hit model. It can maintain state

and user identity across multiple page requests. It can also construct a complex overview of user behavior that goes beyond reporting of user hits.

16.2 Server-Side Session Objects and Users

Session tracking gives Servlets and other server-side applications the ability to keep state information about a user as the user moves through the site. Server-side applications can use this facility to create more stateful user experiences and to track who's doing what on the site.

Java Web Server maintains user state by creating a Session object for each user on the site. These Session objects are stored and maintained on the server. When a user first makes a request to a site, the user is assigned a new Session object and a unique session ID. The session ID matches the user with the Session object in subsequent requests. The Session object is then passed as part of the request to the Servlets that handle the request. Servlets can add information to Session objects or read information from them.

16.3 Session Invalidation

Sessions can be invalidated automatically or manually. Session objects that have no page requests for a period of time (30 minutes by default) are automatically invalidated by the Session Tracker session Invalidation Time parameter. When a session is invalidated, the Session object and its contained data values are removed from the system.

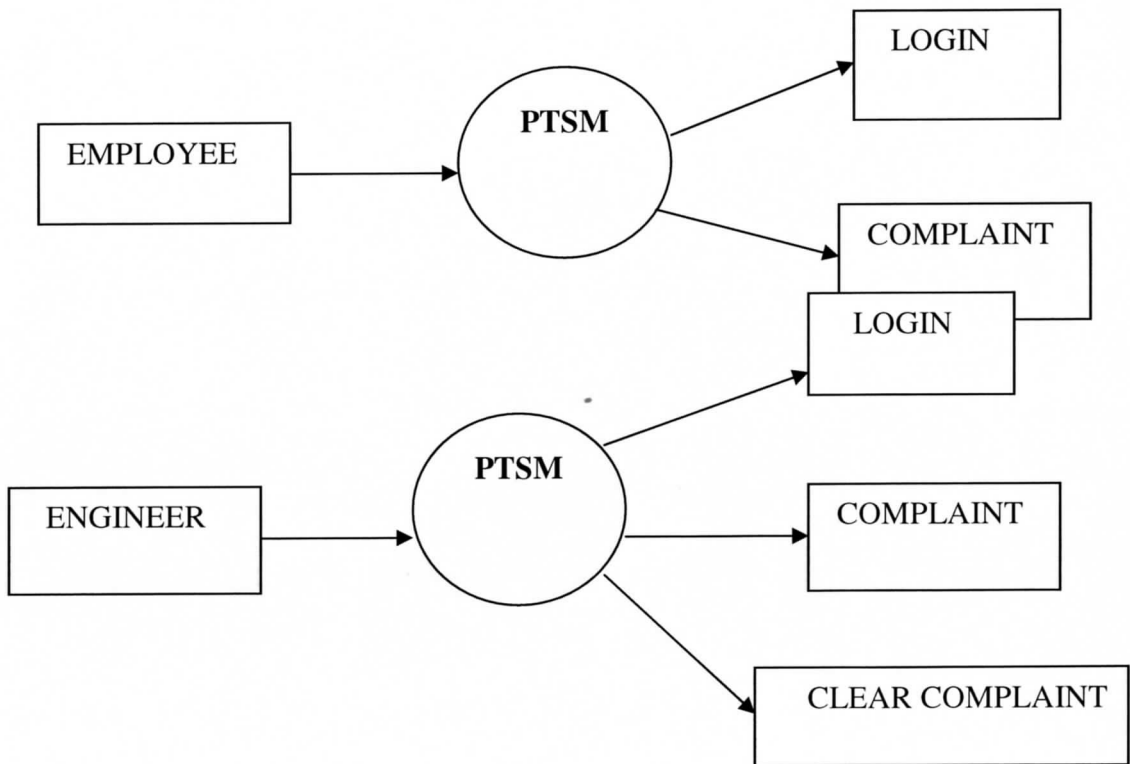
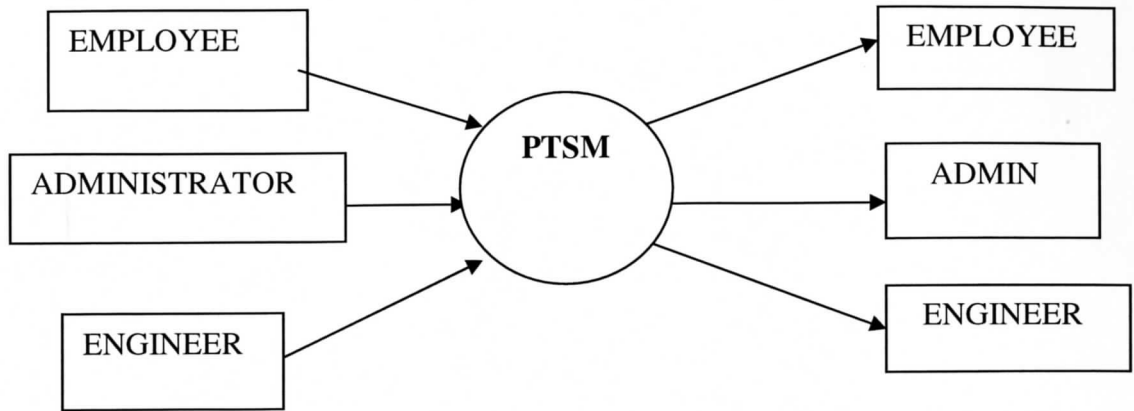
After invalidation, if the user attempts another request, the Session Tracker detects that the user's session was invalidated and creates a new Session object. However, data from the user's previous session will be lost.

Session objects can be invalidated manually by calling `Session.invalidate()`. This will cause the session to be invalidated immediately, removing it and its data values from the system.

16.4 Multiple Servlets

URL conversions are required only if the Servlet supports session tracking for browsers that do not support cookies or browsers that reject cookies. The consequences of not doing these conversions is that the user's session will be lost if the user's browser does not support cookies and the user clicks on an un-rewritten URL. Note that this can have consequences for other Servlets. If one Servlet does not follow these conventions, then a user's session could potentially be lost for all Servlets.

Figure 4: CONTEXT LEVEL DIAGRAMS



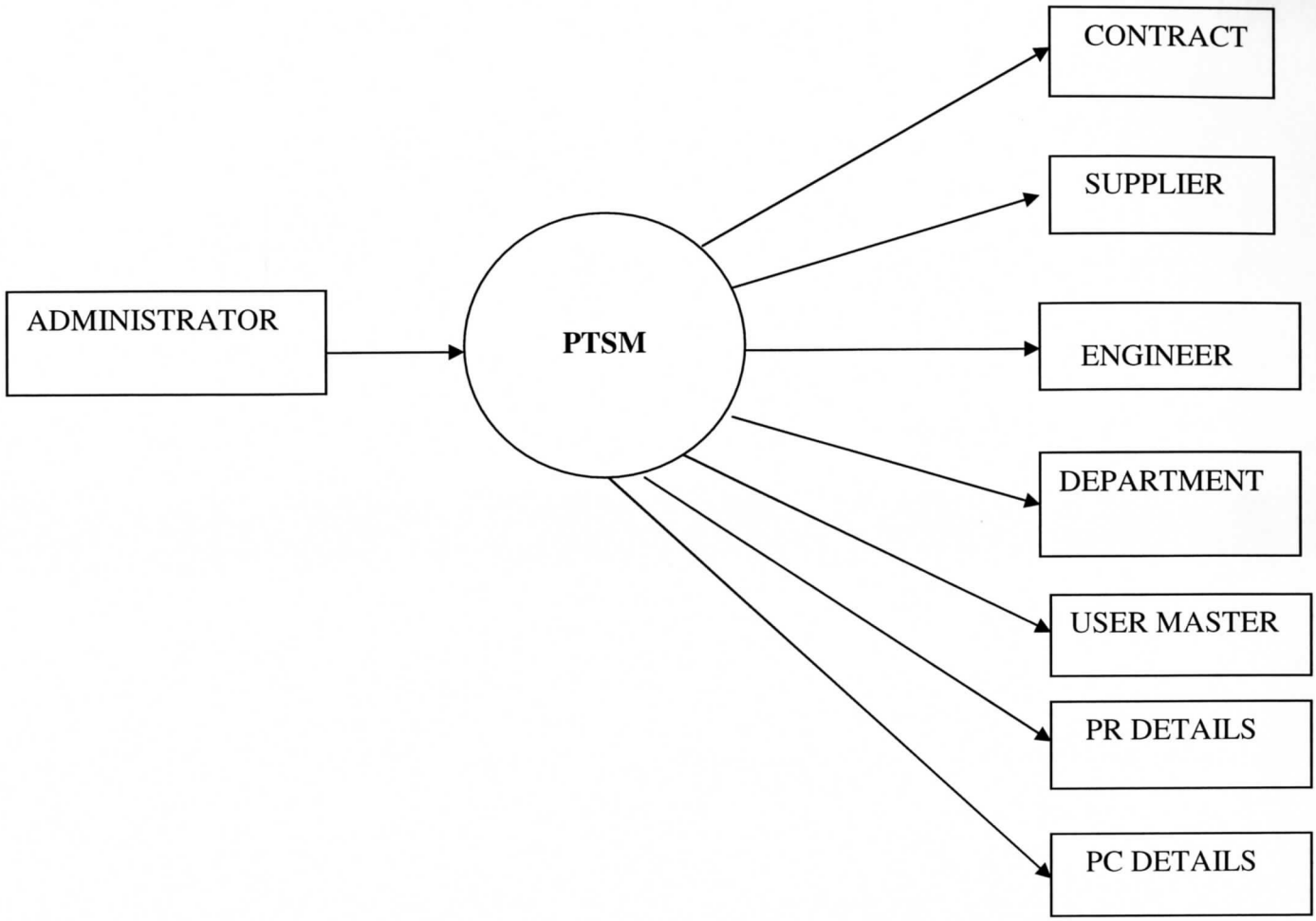
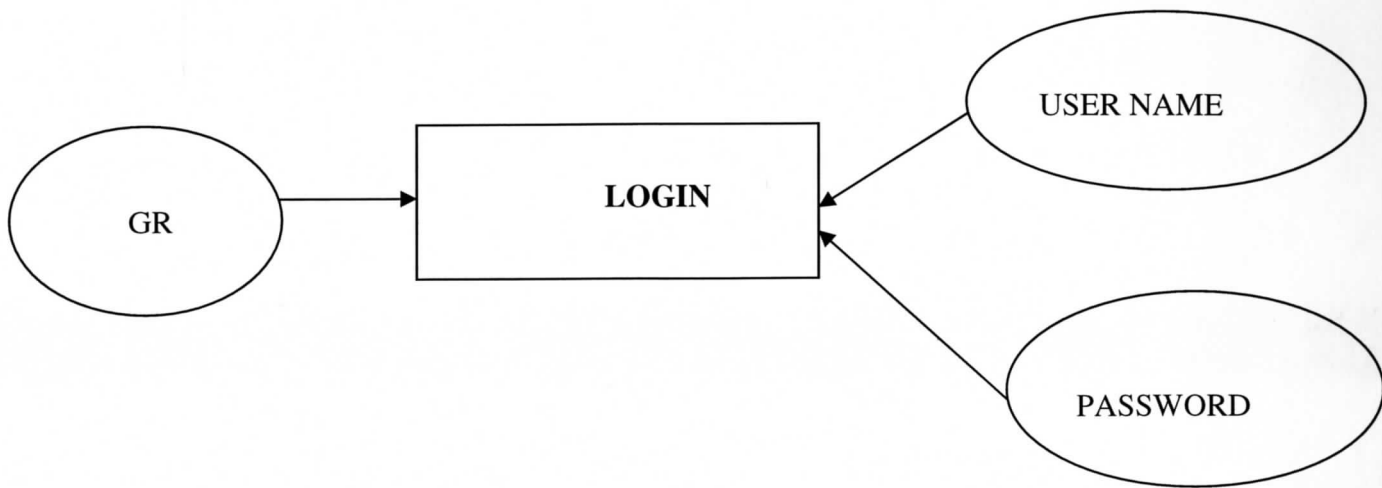
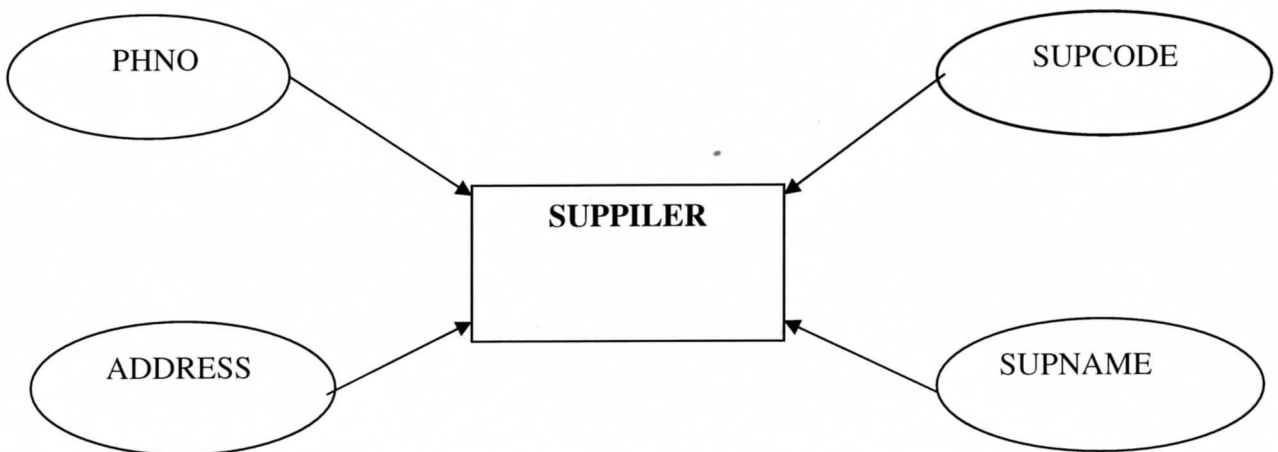


Figure 5: E-R DIAGRAMS

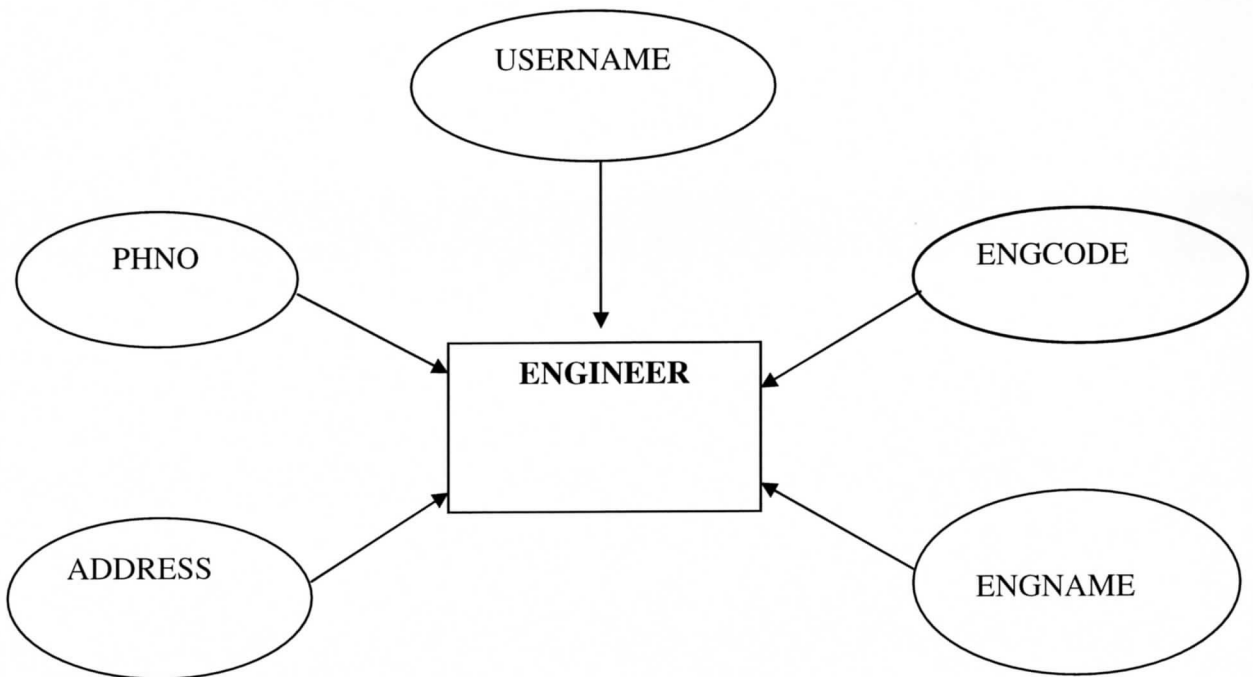
1) LOGIN:



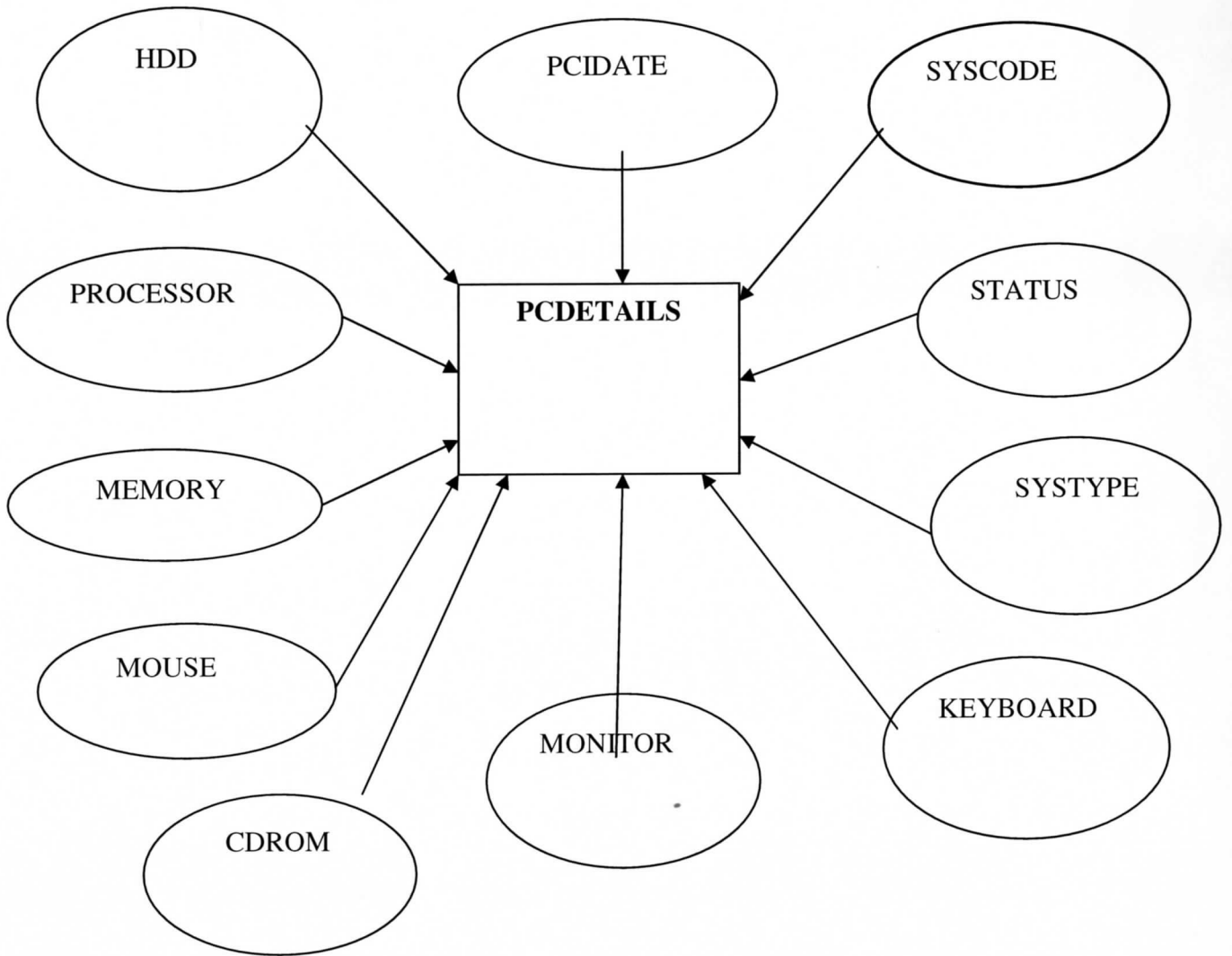
2) SUPPLIER:



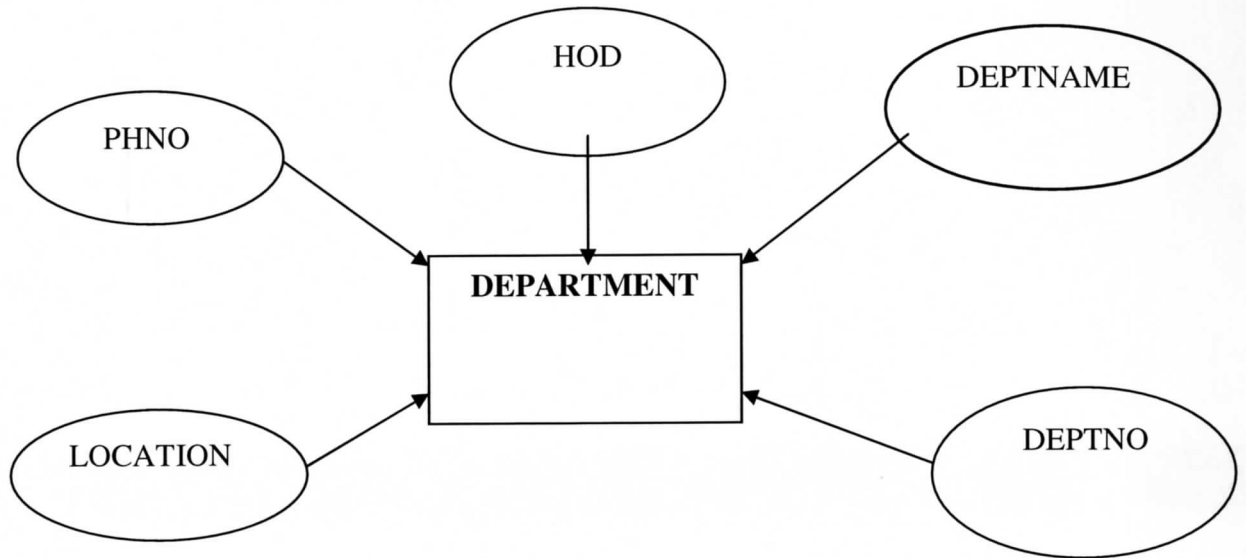
3) **ENGINEER:**



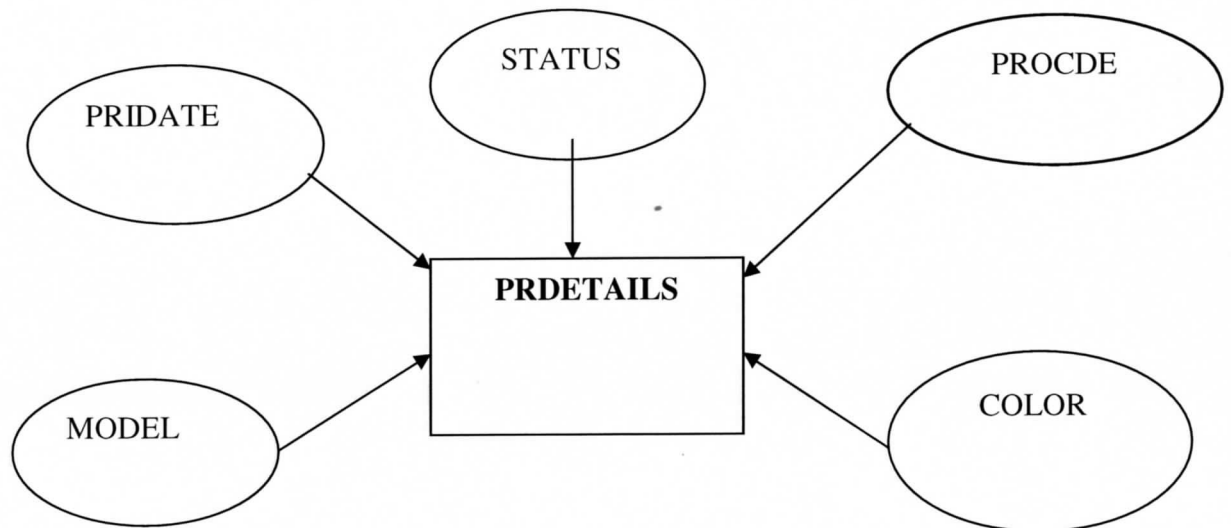
4) PCDETAILS:



5) **DEPARTMENT:**



6) **PRDETAILS:**



7) COMPLAINT:

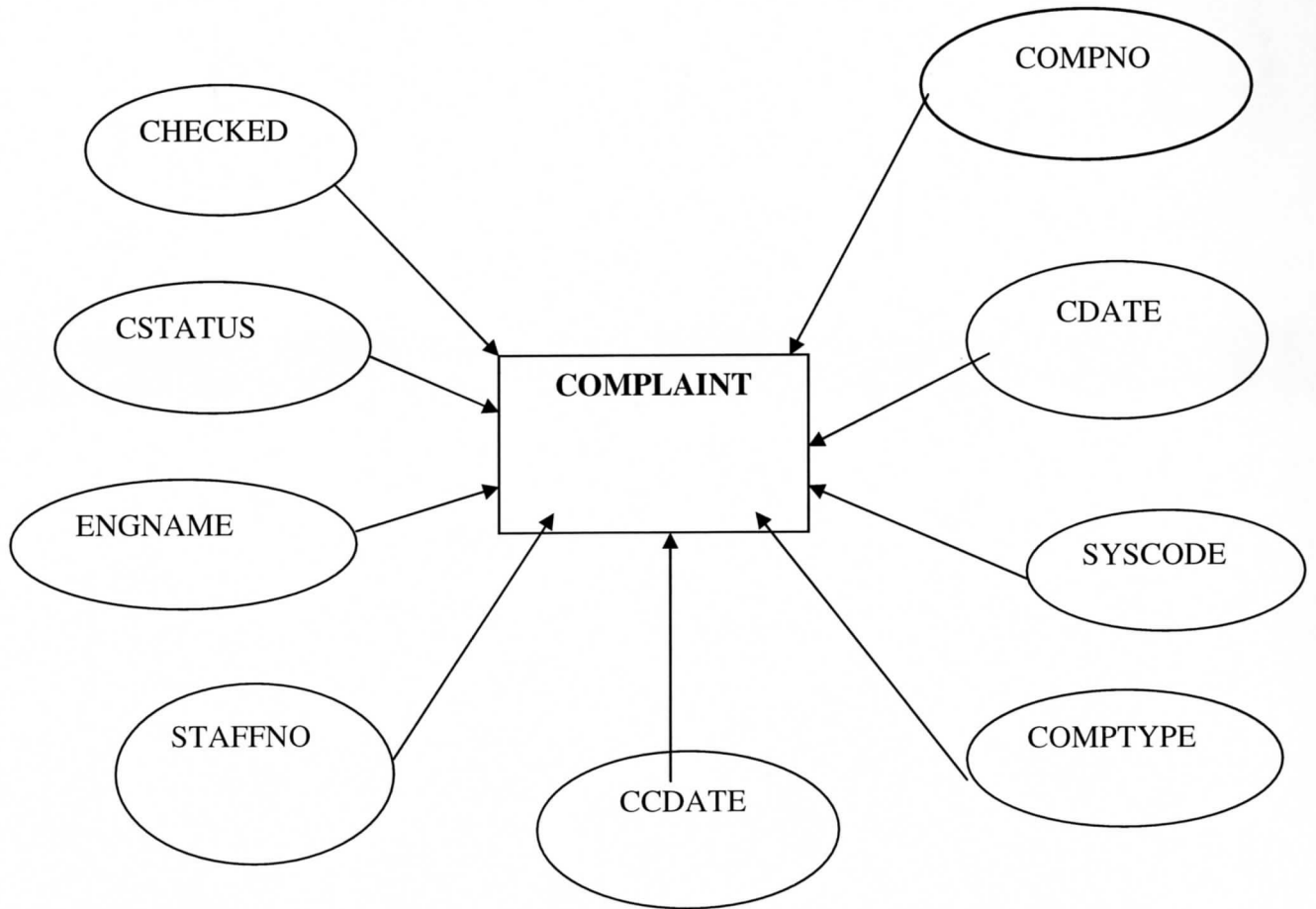


Figure 6: USE CASES

A.

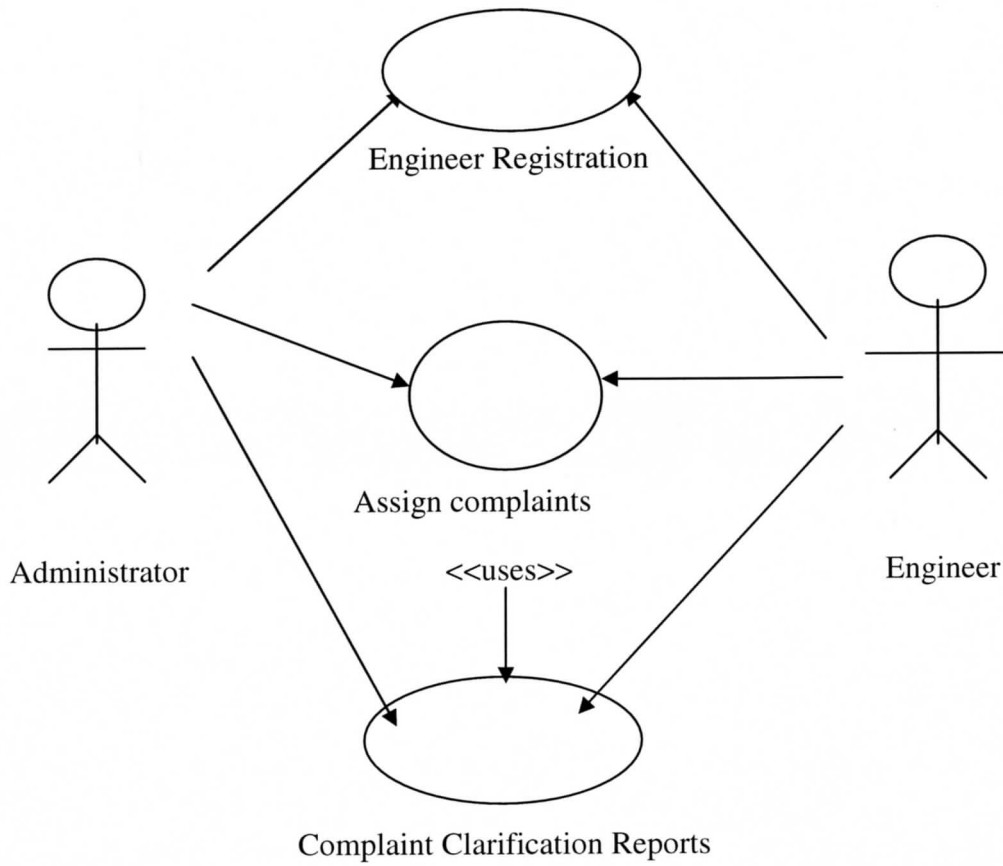


Fig: Administrator and Engineer User cases.

B.

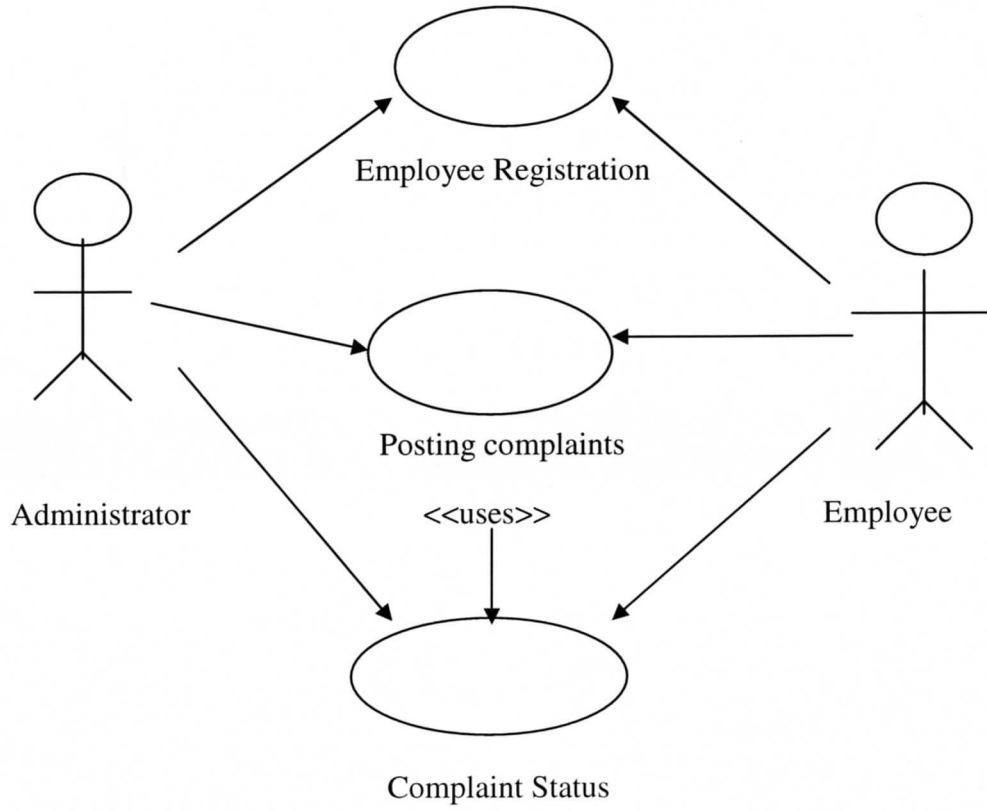


Fig: Administrator and Employee User cases.

C.

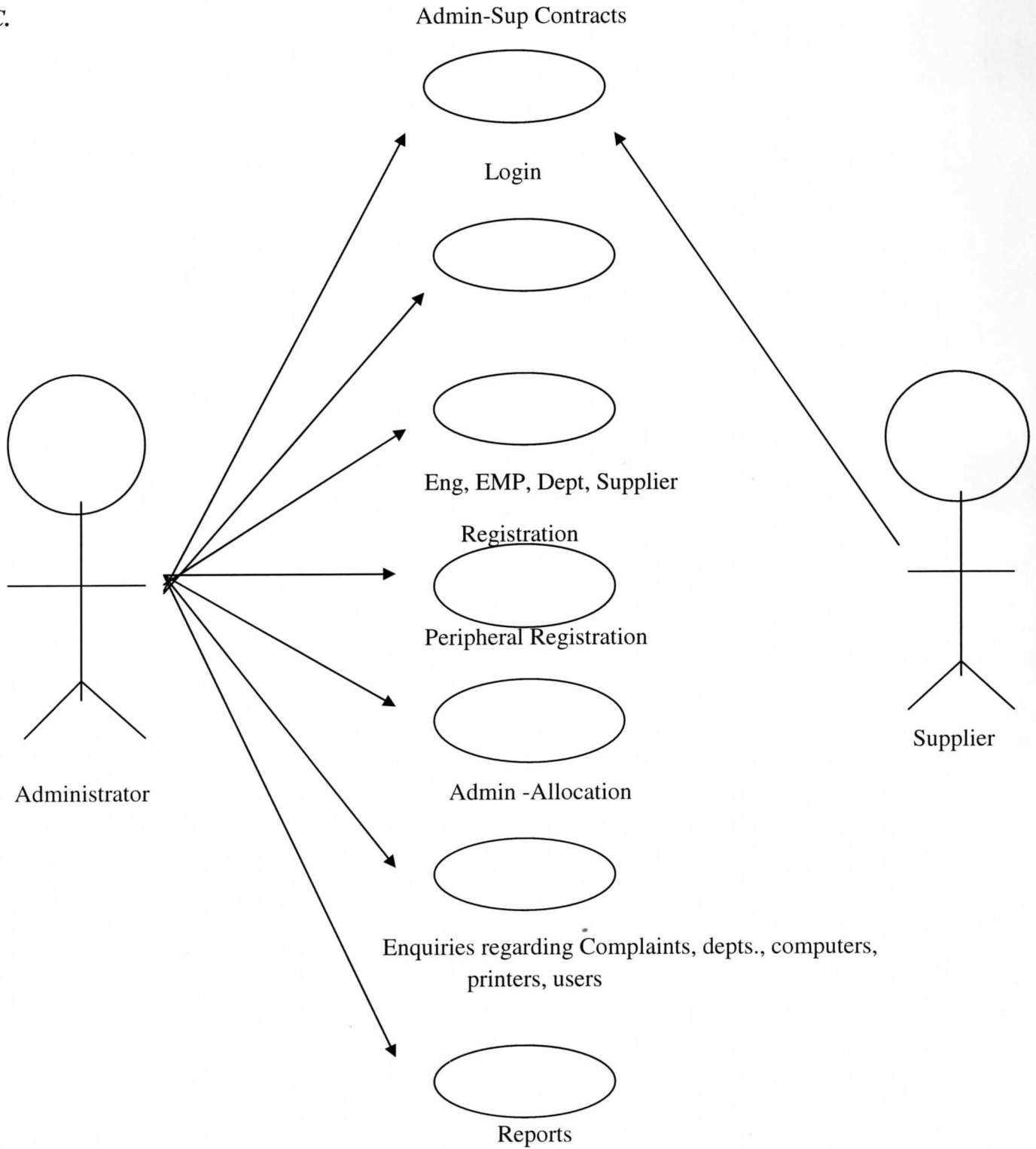


Fig: Use case Diagram for Admin-Supplier

D.

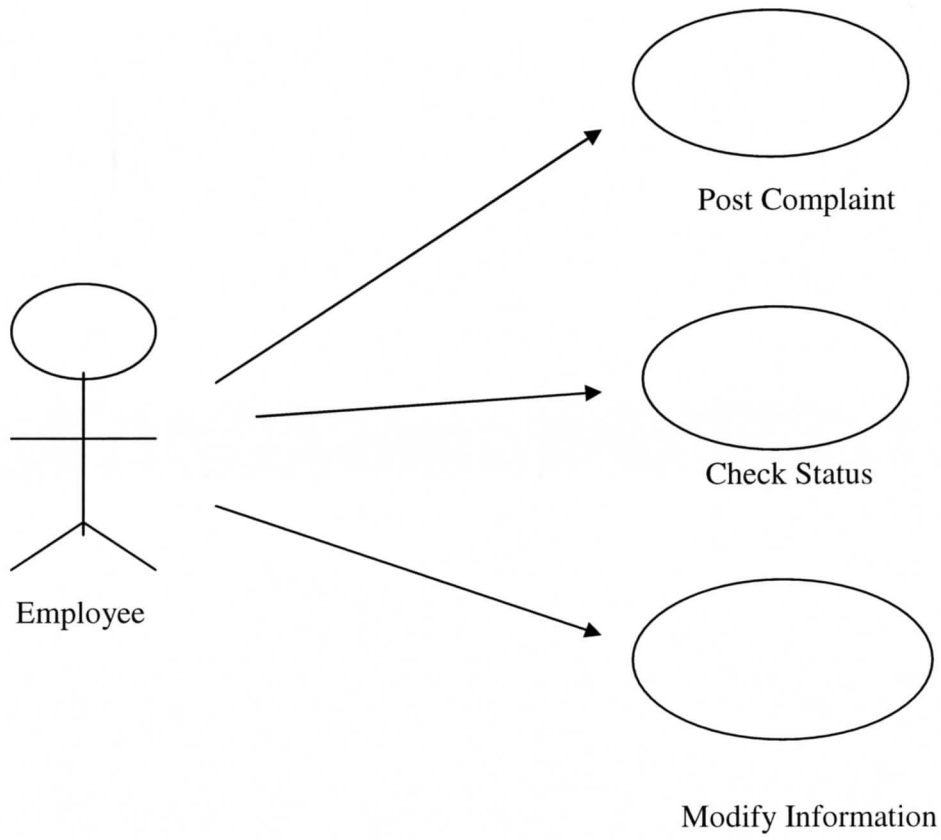


Fig: Employee User case Diagram.

E.

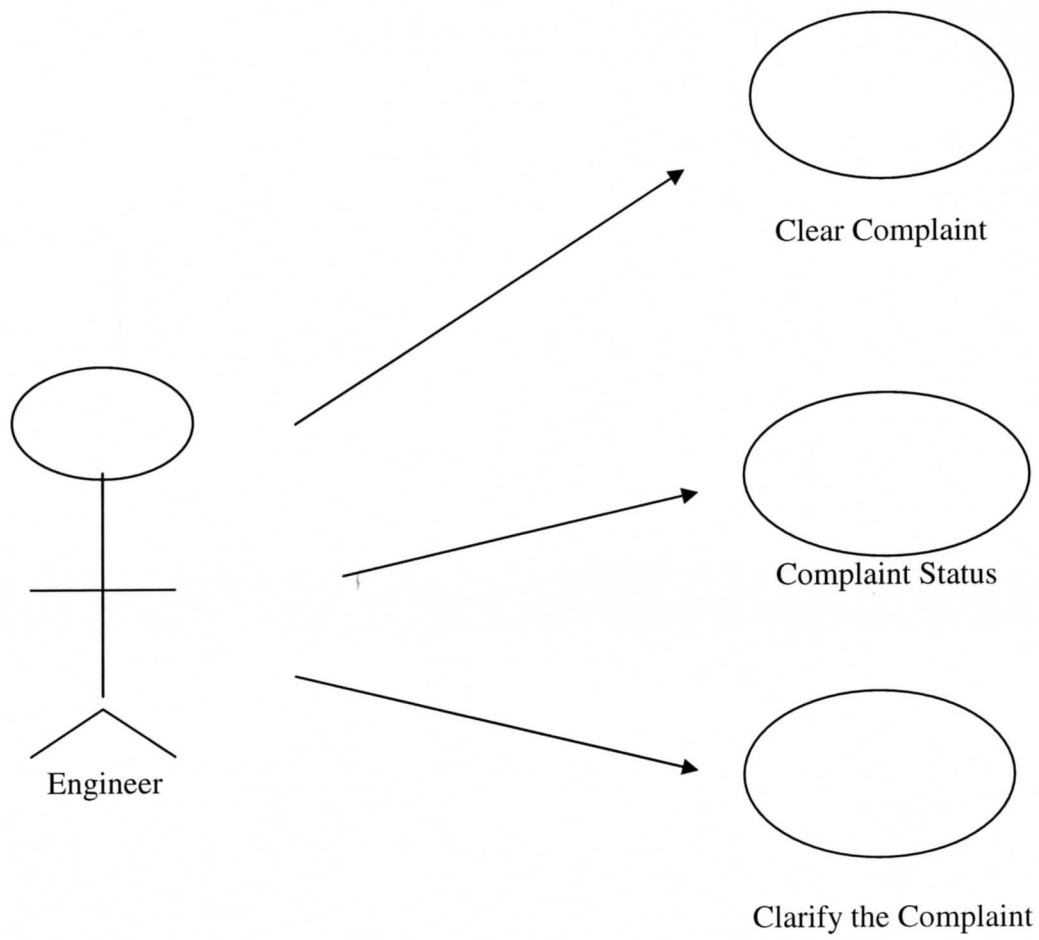


Fig: Engineer User case Diagram

DATA DICTIONARY

Database Design:

Table 1: Login

Column Name	Type
staff no	varchar2
Password	Varchar2

Table 2: Employee

Column name	Type
Staff no	Varchar2
U name	Varchar2
Dept	Varchar2
Designation	Varchar2
Loc	Varchar2

Table 3: Division

Column name	Type
Div name	Varchar2
Loc	Varchar2
Ph no	Varchar2

Table 4: Department

Column name	Type
Dept no	Varchar2
Div name	Varchar2
H O D	Varchar2
Ph no	Varchar2

Table 5: Pc details

Column name	Type	Constraint
Sys code	Varchar2	Primary key
Sys type	Varchar2	not null
Kb type	Varchar2	not null
Monitor	Varchar2	not null
Cd rom	Varchar2	not null
Memory	Varchar2	not null
Processor	Varchar2	not null
status	Varchar2	
Pci date		

Table 6: Pr details

Column name	Type(size)	Constraint
Pr code	Varchar2(10)	prd1 primary key
Color	Varchar2(10)	not null
Model	Varchar2(10)	not null

Pi date	Date	
Status	Varchar2(1)	

Table 7: User master

Column name	Type	Constraint
Staff no	varchar2(10)	use1 primary key
Div name	varchar2(10)	references division(div name)
Dept no	varchar2(10)	references department(dept no)
Designation	varchar2(10)	not null
Loc	varchar2(10)	not null
Ph no	varchar2(10)	not null
Sys code	varchar2(10)	
Sys I date		
Pr code	varchar2(10)	
Pri date		

Table 8: Supplier

Column name	Type	Constraint
Sup code	varchar2	Primary key
Sup name	varchar2	Not null
address	varchar2	
Ph no	varchar2	

Table 9: Contract

Column name	Type	Constraint
Con id	varchar2	Primary key
Sup code	varchar2	References supplier(Sup code)
Con date	Date	
N pc	varchar2	
N pr	varchar2	
Sys type	varchar2	
P type	varchar2	

Table 10: Engineer

Column name	Type	Constraint
Eng code	varchar2	
Eng name	varchar2	not null
Sup name	varchar2	constraint eng1 references supplier(supcode)
Ph no	varchar2	not null

Table 11: Complaint

Column Name	Type	constraints
Emp no	number	primary key
C date	Date	
Sys code	varchar2	
Com type	varchar2	
Cc date		
Staff no	varchar2	constraint com2 references user master(staff no)
Eng name	varchar2	

C status	varchar2	not null
----------	----------	----------

SOFTWARE REQUIREMENTS

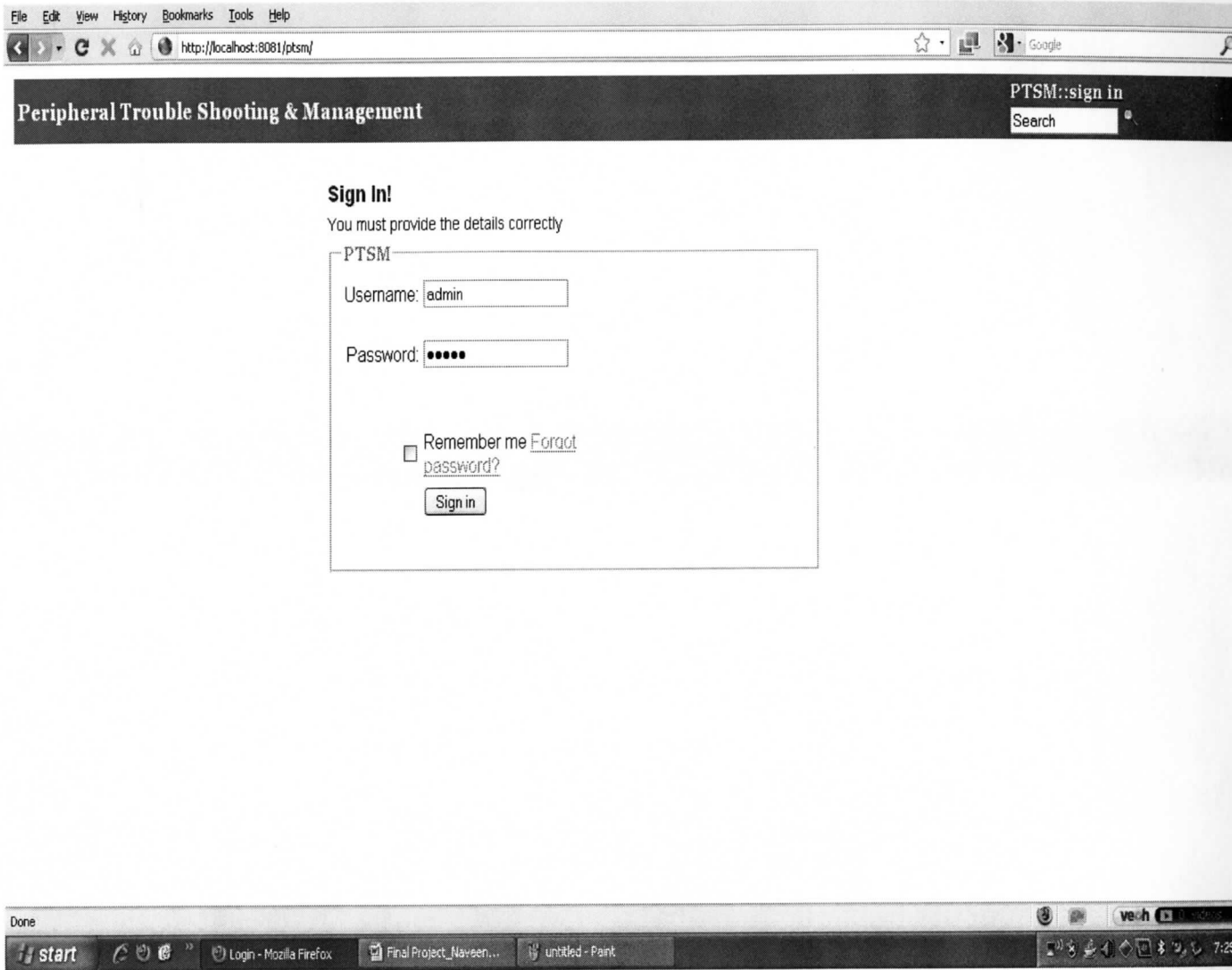
- Jdk 1.2, Jsdk1.2 (Servlets)
- ODBC Drivers installed
- JDBC Drivers installed
- Fully Functional Browser
- J2EE Complaint web server
- Database (Oracle or SQL server)

HARDWARE REQUIREMENTS

- Personal computer with 80486 or higher processor
- 5 GB hard disk space
- 128 MB RAM

SCREEN SHOTS

Login Page:



PC Information:

File Edit View History Bookmarks Tools Help

http://localhost:8081/ptsm/PCDetails.jsp

Add New Details Allocations Complaints Enquiries System Reports Logout

1. PC Information
2. Printer Information
3. Department Information
4. Employee Information
5. Engineer Information
6. Supplier Information
7. Contact Information
8. Modify Employee Details
9. Modify Engineer Details

NEW PC DETAILS

System Code:

System Type:

Keyboard Type:

Monitor:

Mouse:

Cdrom:

Hard Disk:

Memory:

Processor:

Done

start Add PC Details - Mozl... untitled - Paint INFG 788_Naveen Ma... 3:39 PM

Printer Details

File Edit View History Bookmarks Tools Help

http://localhost:8081/ptsm/PrinterDetails.jsp

Add New Details Allocations Complaints Enquiries System Reports Logout

1. PC Information
2. Printer Information
3. Department Information
4. Employee Information
5. Engineer Information
6. Supplier Information
7. Contact Information
8. Modify Employee Details
9. Modify Engineer Details

NEW PRINTER DETAILS

Printer Details (* fields are compulsory)

Printer Code: *

Color: *

Model: *

Done

start PrinterDetails - Mozill... untitled - Paint INF5 736_Naveen Ma... 3:40 PM

Department Details:

NEW DEPARTMENT DETAILS

Department Details (* fields are compulsory)

Department Code: *

Department Name: *

Location: *

Phno:

1. PC Information
2. Printer Information
3. Department Information
4. Employee Information
5. Engineer Information
6. Supplier Information
7. Contact Information
8. Modify Employee Details
9. Modify Engineer Details

Done

start Department Details - ... untitled - Paint INPS 789_Naveen Me... 3:41 PM

Employee Details

File Edit View History Bookmarks Tools Help

http://localhost:8081/ptsm/EmployeeDetails.jsp

Add New Details Allocations Complaints Enquiries System Reports Logout

1. PC Information
2. Printer Information
3. Department Information
4. Employee Information
5. Engineer Information
6. Supplier Information
7. Contact Information
8. Modify Employee Details
9. Modify Engineer Details

NEW EMPLOYEE DETAILS

User Name:

Employee Name:

Department Number:

Designation:

Location:

Ph No:

Done

start Employee Details - M... untitled - Paint INFS 788_Naveen Ma... 3:42 PM

Engineer Details

File Edit View History Bookmarks Tools Help

http://localhost:8081/ptsm/EngineerDetails.jsp

Add New Details Allocations Complaints Enquiries System Reports Logout

1. PC Information
2. Printer Information
3. Department Information
4. Employee Information
5. Engineer Information
6. Supplier Information
7. Contact Information
8. Modify Employee Details
9. Modify Engineer Details

ENGINEER DETAILS

Engineer Details (* fields are compulsory)

User Name: *

Engineer Name: *

Address: *

Phone:

Done

start [Taskbar icons] 3:43 PM

Supplier Details

File Edit View History Bookmarks Tools Help

http://localhost:8081/ptsm/Supplier.jsp

Add New Details Allocations Complaints Enquiries System Reports Logout

- 1. PC Information
- 2. Printer Information
- 3. Department Information
- 4. Employee Information
- 5. Engineer Information
- 6. Supplier Information
- 7. Contact Information
- 8. Modify Employee Details
- 9. Modify Engineer Details

SUPPLIER DETAILS

Supplier Details (* fields are compulsory)

Supplier Code: *

Supplier Name: *

Address: *

Phone:

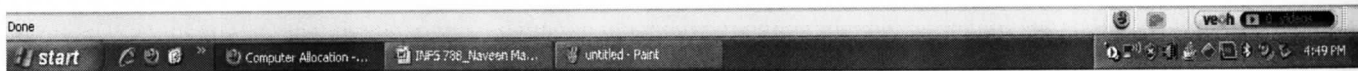
Done

start Supplier Details - Moo... untitled - Paint IMFS 788_Naveen Ms... 3:44 PM

PC Allocation to Employees

The screenshot shows a web browser window with the URL `http://localhost:8081/ptsm/ComputerAllocation.jsp`. The browser's menu bar includes File, Edit, View, History, Bookmarks, Tools, and Help. Below the address bar is a navigation menu with buttons for Add New Details, Allocations, Complaints, Enquiries, System Reports, and Logout. A dropdown menu is open under 'Allocations', listing: 1. Computer Allocation, 2. Printer Allocation, 3. HDD Allocation, and 4. Engineer Allocation. The main content area is titled 'PC ALLOCATION TO EMPLOYEES' and contains the following form elements:

- Staff No:** A dropdown menu with the value '12) Tom' selected.
- System No:** A dropdown menu with the value '7' selected.
- Three buttons: 'Allot', 'Reset', and 'Back'.



Printer Allocation to Employees

File Edit View History Bookmarks Tools Help

http://localhost:8081/ptsm/PrinterAllocation.jsp

Add New Details Allocations Complaints Enquiries System Reports Logout

1. Computer Allocation
2. Printer Allocation
3. HOD Allocation
4. Engineer Allocation

PRINTER ALLOCATION TO EMPLOYEES

Select Staffno & Printer no to proceed

Staff No: 10| Stallon

Printer No: Printer 06

Allot Reset Back

Done

start Printer Allocation - M... INF5 788_Naveen Ma... untitled - Paint

4:50 PM

Complaint Status

File Edit View History Bookmarks Tools Help

http://localhost:8081/jpsm/ComplaintStatus.jsp

Add New Details Allocations **Complaints** Enquiries System Reports Logout

1. New Complaints
2. Pending Complaints
3. Complaint Status

COMPLAINT STATUS

Complaint No	System Code
<u>1</u>	1
<u>2</u>	2
<u>3</u>	1
<u>4</u>	3
<u>5</u>	3
<u>6</u>	112
<u>7</u>	1111

Done

start Complaint Status - M... JNFS 788_Naveen Ma... untitled - Paint 4:54 P

Details of PC:

File Edit View History Bookmarks Tools Help

http://localhost:8081/ptsm/PcEnquiryDetails.jsp

Add New Details Allocations Complaints Enquiries System Reports Logout

1. PC Enquiries
2. Printer Enquiries
3. Department Enquiries
4. User Enquiries
5. Contract Enquiries

DETAILS OF PC NO: 1

PC No	1
System Type	client
Keyboard Type	LG
Monitor	LG
Cdrom	LG
Mouse	Scrole Mouse
Memory(Ram)	1gb
Processor	p4
Hard Disk	LG
Allotted	y
PcIssuse Date	2009-02-28 16:30:43

Done

start PC Enquiry Details - ... INF5 769_Naveen Ma... untitled - Paint 4:55 PM

Details of Printer

The screenshot shows a web browser window with the URL `http://localhost:8081/ptsm/PrinterEnquiryDetails.jsp`. The browser's menu bar includes File, Edit, View, History, Bookmarks, Tools, and Help. Below the address bar is a navigation menu with buttons for Add New Details, Allocations, Complaints, Enquiries, System Reports, and Logout. The Enquiries menu is open, showing a list of options: 1. PC Enquiries, 2. Printer Enquiries, 3. Department Enquiries, 4. User Enquiries, and 5. Contract Enquiries. The main content area displays the title "DETAILS OF PRINTER NO: Printer 01" and a table with the following data:

Printer No	Printer 01
Color	Silver
Model	HP
Allotted	y
Issue Date	2009-12-03 14:10:54

The Windows taskbar at the bottom shows the Start button, several open applications (Printer Enquiry Detail..., INF5 788_Naveen Ma..., untitled - Paint), and the system tray with the time 4:56 PM.

Department Enquires

File Edit View History Bookmarks Tools Help

http://localhost:8081/ptsm/DepartmentEnquiry.jsp

Add New Details Allocations Compliants Enquiries System Reports Logout

1. PC Enquiries
2. Printer Enquiries
3. Department Enquiries
4. User Enquiries
5. Contract Enquiries

DEPARTMENT ENQUIRIES

Please Select Department Code to proceed

Department Code:

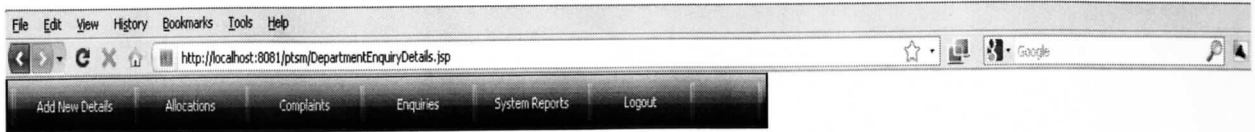
Display

Done

start Printer Enquiry - Mozl... INF3 788_Naveen Ma... untitled - Paint

vech 4:58 PM

Details of Department

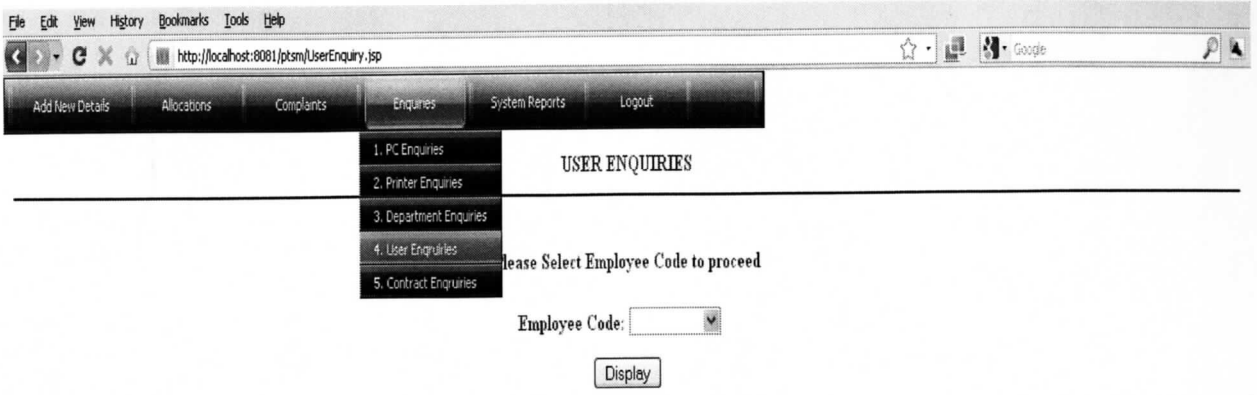


DETAILS OF DEPARTMENT : 003

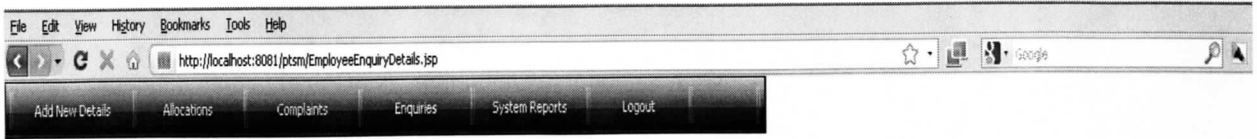
Department Number	003
Department Name	Accounting
HOD	null
Location	California
Phno	null



User Enquires:



Details of Employee



DETAILS OF EMPLOYEE : 6

Employee Id	6
User Name	Malladi
Employee Name	Naveen
Department no	xyz
Department name	Engineering
Hod	null
Designation	Dev
Location	Sioux Falls
Phno	1234567890
System code	001
System Issue date	2009-12-03 14:06:44
Printer code	Printer 01
Printer Issue date	2009-12-03 14:10:54



CONCLUSION

The efficiency of any system designed to suit an organization depends cooperation during the implementation stage and also flexibility of the system to adopt itself to the organization.

“Peripheral Trouble Shooting and Management System” has been developed to overcome the problems with traditional peripheral management system. Advantages over traditional peripheral management system are centralized control, efficient handling and reporting mechanism, prevention of unauthorized access.

BIBLIOGRAPHY

- H.M.DIETEL.P.J.DIETEL, Java How to Program, Prentice Hall Ltd, Second Edition
- Gray Cornett, Horstmann, Corel Java, Sunsoft Press, 1996.
- Patrick Naughton & Herbert Schildt, Java: The Complete Reference, Tata McGraw-Hill, March 1997.
- Grady Bouch, Object Oriented Analysis and Design with Applications, the Benjamin/Cummings, 1994.
- Distilled UML by Martin Fowler, Low Price Edition, 2005.

WEB SITE REFERENCES

<http://java.sun.com>

<http://sun.com>

<http://coreservlets.com>

<http://jsp-in.com>

<http://javaworld.com>