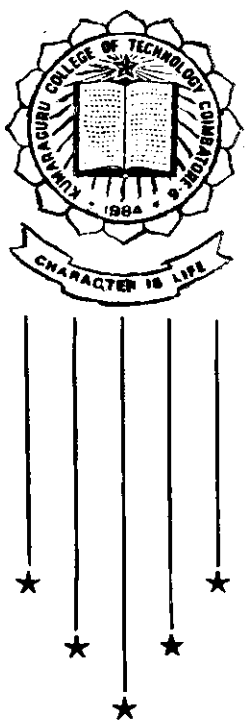
Lab Maintenance And Job Scheduling



2000 - 2001

Project Report

P-549

Submitted by

Miss. A. SARAVANA PRIYA
(Reg. No. 9827Q0158)

Mr. R. KARTHICK
(Reg. No. 9827Q0127)

Guided by

Miss. N. RAJATHI, B.E.,
Lecturer / CSE Dept.

In partial fulfilment of the requirements for the award of the Degree of B.Sc (Engg)
IN APPLIED SCIENCE & TECHNOLOGY
of the Bharathiar University, Coimbatore.

Department Of Computer Science And Engineering
Kumaraguru College Of Technology
Coimbatore - 641 006

Department Of Computer Science And Engineering
Kumaraguru College Of Technology
Coimbatore- 641006

Project Work 2000 – 2001

Certificate that this is the Bonafide Record of the Project Work Done By

Miss. A.SARAVANA PRIYA
(Reg.No9827Q0158)

Mr. R.KARTHICK
(Reg.No9827Q0127)

In partial fulfillment of the requirements for the award of the Degree of B.Sc., (Engg) in Applied Science & Computer Technology of the Bharathiar University

14/3/2001

Guide

S. Jayaraj 14/3/01

Head of the Department

Submitted for the University Examination held on 15/03/2001

J. M 15/3

Internal Examiner

15/3/2001

External Examiner

Department of Computer Science and Engineering

B.Sc (CT) Stream

CERTIFICATE

This is to certify that (1) Ms. A. Saravana Priya, Reg No : 9827Q0158 and (2) Mr. R. Karthick, Reg No : 9827Q0127 has been developed the project title " Lab Maintenance and Scheduling " for the B.Sc Computer Technology stream of our college is well tested and implemented

13/3/2009

Class Advisor

III B.Sc (CT)

Course Co-Ordinator

14/3/09

HOD / CSE

DEDICATED...

To our beloved parents

For the unconditional love and affection

To all our honourable teachers

For the Knowledge acquire and support

To our respected guides

For the intuition, advice and inspiration

To all our friends and loved ones

For the unfailing support and love.

ACKNOWLEDGEMENT

With profound gratitude, we praise the Lord for his abundant blessings.

It gives us great felicity to convey our deep and sincere thanks to our Principal **Dr. K. K. Padmanabhan, B.Sc. (Engg), M.Tech., Ph.D.,** and our **HOD Dr. S. Thangasamy**.

We express our sincere thanks to our guide **Miss.N.Rajathi B.E.,** Lecturer/CSE, Kumaraguru College Of Technology, for his valuable and regular supervision in the course of completion of this project.

We express our gratitude to our project co-ordinator **Mr. M. Raju MCA, B.Ed.,** Lecturer/CSE, Kumaraguru College of Technology, for his valuable and regular supervision in the course of completion of this project. We take this opportunity to to thank each and every staff member of Department of Computer Science and Engineering, Kumaraguru College of Technology, Coimbatore, our parents and friends for their excellent guidance and support during the project work.

ORGANISATION PROFILE

Kumaraguru College of Technology,Coimbatore a private co-educational Engineering College was started in 1984 under the auspices of Ramananda Adiglar Foundation ,a charitable and educational trust and a unit of Sakthi group.Located in a congenial environment on the outskirts of the city,just 3 kms from city limits and 2 kms from Coimbatore -Mysore highway..the college campus spreads over a sprawling 150 acres and is well developed with black-topped roads,sodium vapour lamps,drainage adequate power and water supply.The management has invested more than Rs.35 Crores for the creation of infrastructural facilities.

Under the able guidance and patronage of Arutselvar Dr.N.Mahalingam,Chairman,Sakthi Group and prof.K.Arumugam,Correspondent ,the college has developed excellent infrastructural facilities such as building .well equipped labaratories ,24 hours internet center ,a earth station ,workshops and well qualified faculty.The college offers under-graduate and post-graduate courses under affiliation to the Bharathiar University ,Ciombatore and with the approval of All India Council for Technical Education (AICTE).

The calibre of our students and the well deserved reputation of the college for quality attracts leading companies to hold campus recruitment.In the

last academic year more than 250 students have been placed through our Placement Center .

The alumini have done us proud by proving their worth wherever they have happened to be employed .All our 2700 alumini are happily settled in life after college.

SYNOPSIS

Analysing program is virtually important when it comes to writing software. This Can be achieved by writing efficiency programs & judging the relationship to performance.

This Project entitled " **Lab Maintenance & Job Scheduling** " has been done at Kumaraguru College of Technology for the maintenance of the system & service allocation of those system to the service engineers by the system. This project is developed in order to reduce the manual work & to have a report about the functioning of the system.

The project is programmed to get report of fault occurrence in the system & job alloted to service engineers by the system allocating the jobs to service engineers according to there ability, priority & no. of. jobs given . It's role is also Found in maintaining the records of jobs attended & registering the status. gathering information about lab attender, Lab-Incharge & Service Engineers.

Contents

CONTENTS

Introduction

INTRODUCTION

"LAB MAINTENANCE & JOB SCHEDULING", is developed to perform the maintenance of Inter connected systems and also to schedule the jobs to Service Engineers in priority wise .

Purpose:

To maintain all the systems through a network. And also to allocate Jobs to the service engineers according to their skills.

Scope:

The main objective of this project is job allocation to Service Engineers by the system used mainly for reducing manual allotment.

DEFINITIONS:

Web-Server:

The Java Web-Server is an implementation of the Java-Server architecture, which defines a generic server & service frame work. This framework defines services, the server process & the Servlet API.

Servlet Framework:

A Servlet is a Java object that conforms to a specific interface as defined by Java-Server architecture. Servlets are defined by Java server architecture.

Servlets are loaded & invoked by services and a service can utilize multiple Servlets.

Similar to services, Servlets can be added, removed (or) configured while the server is running.

WEB-BROWSER:

A Browser also known as web client is a Software invention used to the most important Internet Software in use. The major browser used to surf the Internet is Microsoft IE.

HTTP[HYPER TEXT TRANSFER PROTOCOL]:

HTTP, is the acronym for Hyper Text Transfer Protocol. It is the internet protocol used to communicate between clients & servers. Web clients are

Browsers. HTTP'S primary task is to move request from clients to servers, and response .

URL/ *Universal Resource Locator]*

URL is the primary naming scheme which is used to identify web resources that can either be HTML does or other services present in the web. These web resources are identified with special names called UNIFORM RESOURCE IDENTIFIERS(URI). URL is the standard method to identify any resource. Ex: DOCS,GRAPHICS etc.

Client Side Scripting:

To enhance the functionality of client side(i.e) browser we use Java Script.

Acronyms:

- ✓ **JSDK**- JAVA SERVLET DEVELOPMENT KIT.
- ✓ **HTML**-HYPER TEXT MARKUP LANGUAGE.
- ✓ **HTTP**-HYPER TEXT TRANSFER PROTOCOL.
- ✓ **URL**-UNIVERSAL RESOURCE LOCATOR.
- ✓ **API**-APPLICATION PROTOCOL INTERFACE.
- ✓ **JVM**-JAVA VIRTUAL MECHINE.

- ✓ **JRE**- JAVA RUNTIME ENVIRONMENT.
- ✓ **GIF**-GRAFICAL INTERCHANGABLE FORMAT.
- ✓ **AMC**-ANNUAL MANTENANCE CONTRACT.

ABOUT THE PROJECT

The Project is basically an On-Line process it could be treated as Batch Processing.

- Maintaining the Labs & the systems in the lab of every department.
- Indication of fault occurrence in the system by the lab attender.
- Indication of job allotment to service department by the system, according to the priority.
    - Service Engineer skill or ability
    - Number of jobs allotted.
- Reallocation of job done by the supervisor incase,
    - When particular Service - Engineer is absent.
    - Pending jobs.
- Maintaining the service Engineer status record.

General Description

GENERAL DESCRIPTION

Product Perspective:

Servlets :-

Servlets are server side Applets. Servlets are loaded and executed by a WebServer in the same manner that applets are loaded and executed by a web Browser. Servlets are platform independent, pasistant (servlets are loaded only once by web server and can maintain services between requests) fast, extensible (all the benefits of java can be brought into our servlets). Servlets are secure [Only way to gain access to servlet is protected as well, servlets can be used by various clients.

JDBC in servlets :-

JDBC is an SQ-Level API that allows to execute SQL and Access statements and retrieve the results. By using this API for db access we can change the under going db driver without having to modify our applets.

About MS - Access 7.0:-

Microsoft Access 7.0 provides user friendly ways to design and modify databases. It hosts a wide range of data types as well. It is a RDBMS and has built is SQL, as most RDBMS and has built is SQL, as most RDBMS have.

Few of the terms used in MS Access 7.0 are

- Dynast : A table that is updatable.
- Snapshot : A table that cannot be updated
- Fields : columns of a table
- Records : Row of a table

- Standard databases formats for many applications are in built.
- Database creation is very simple and is done in a GUI environment
- Modification can be easily done to create database.
- Inputs to do can be formatted using the validation test.

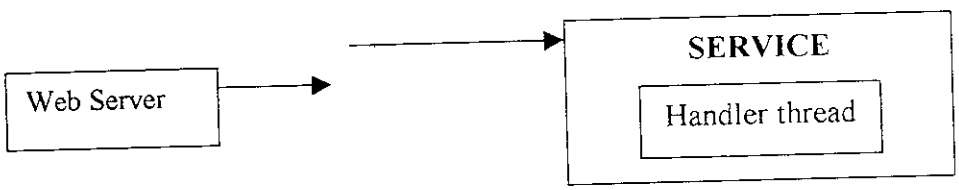
Java Web Server :

The Java Web server is an an implementation of the JavaServer architecture,which defines a generic server and service framework.This framework defines services,the server process and the servlet API.

The Service Framework:

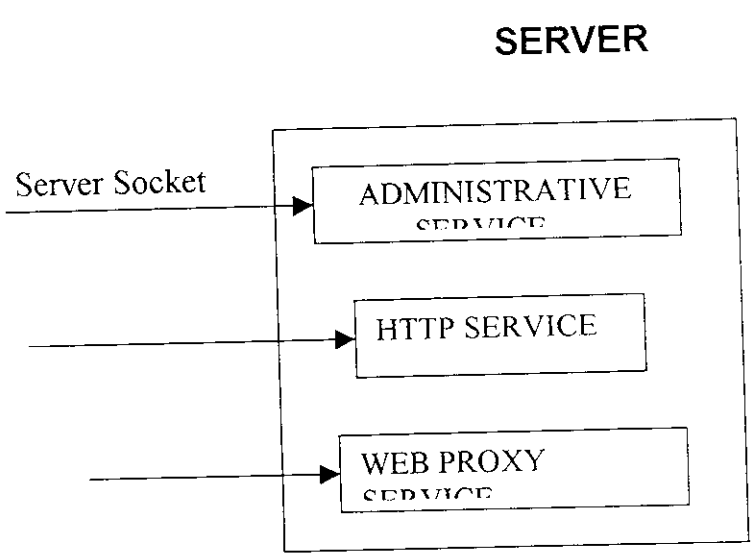
The service framework defines a set of interface for implementing services that interact with clients using multiple handler threads. A service is defined as an individual protocol such as HTTP or FTP .when service is initiated

,it will acquires a specific server socket. After acquiring a server socket, the service will create a pool of handler threads. Each thread is idle while it waits for a connection request. Once a connection request has been received ,a handler thread will perform all protocol interaction on that connection.



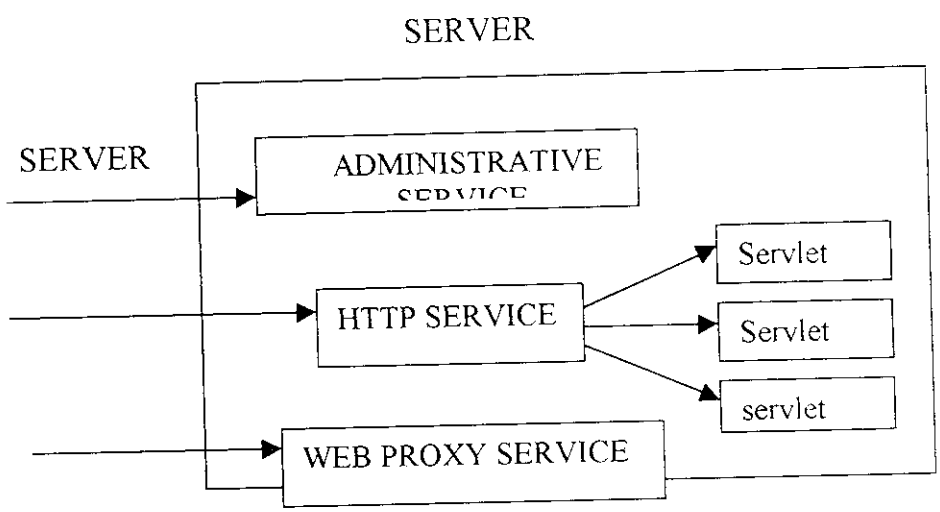
The Server Frame Work:

A server is a instance of a java virtual machine .One server can support multiple services, which are configured to start when the server process is initiated. For example, a java web server can start an administrative service, an HTTP service or a web proxy service. Services can be added, removed or configured while the server is running.



The Servlet Framework:

A servlet is a java object that conforms to a specific interface as defined by the Java Web Server architecture. Servlet are loaded and invoked by services and a service can utilize multiple servlets. Similar to services, servlets can be added, removed or configured while the server is running .



Javaweb Server Administration:

By default, the java web server administration tool is installed on port 9090.To invoke the administration tool, we must access the administration tool, we must access the administration port of the server using a java enabled browser.

The default user name and password are both"admin".manage server and services shows current state of the Java Web Server installed and currently running.The services are shown :

- ✓ **Web Service:**A standard HTTP protocol service,which,by default,is installed on port 8080.
- ✓ **Secure Web Service** : A secure HTTP protocol service(known as SHTTP),which is by default installed on port 7070.
- ✓ **Proxy service :**A web proxy which by default is installed on port 6060.

To manage a service, either click on the service and press the manage button or double-click. to start or stop a service, use restart and stop button respectively. The administrative controls available for the HTTP Web service are setup, monitor, security and servlets.

JDBC

Establish A Connection To Database:

The first step in using a database product via JDBC is to establish a connection .JDBC connection are specified by a URL which has the format:

Jdbc:<subprotocol>:<subname>

Where subprotocol is the kind of database connectivity being requested and subname provides additional information required to establish a connection .

When a connection URL is requested from the JDBC Driver Manager each of the JDBC Driver is asked if it can service the given URL.

There are two mechanism by which Driver Manager can be notified that a JDBC driver is available the SQL driver property and JDBC registration .The SQL drivers system priority is referenced by the DriverManager to get a list of JDBC driver available on the system. It contains a colon separated list of JDBC driver class names.

These names can be used by Driver Manager to satisfy a connection request .Driver registration gives us greater control over which JDBC driver. All JDBC drivers are required to register themselves with the DriverManager when they are instantiated.

Executing An SQL Statement:

After establishing a connection to the database we are ready to execute SQL statements that perform some kind of work. Before executing an SQL statement ,we need to create a statement object ,which provides an interface to an underlying database SQL engine .

JDBC Driver Types:

Sun has defined four JDBC driver types:

- JDBC-ODBC Bridge,plus ODBC driver
- Native-API,partly-Java driver
- JDBC-net ,pure Java driver
- Native-protocol,pure Java driver

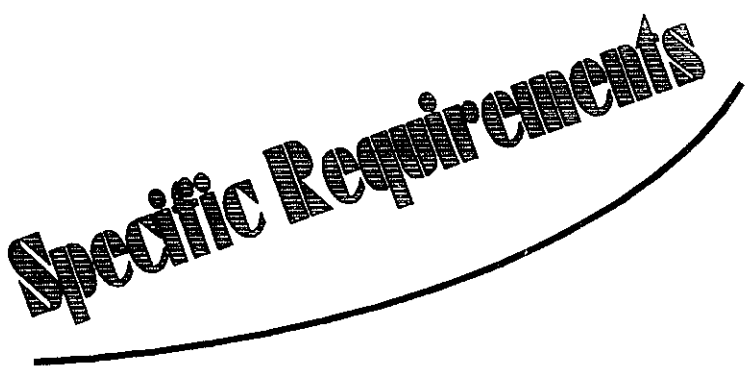
Each of these types meets a different application need JDBC-ODBC Bridge, plus ODBC Driver

This driver type is provided by sun with the jdk1.1 and later. JDBC access to databases through ODBC drivers. The ODBC driver must be configured on the client for the bride to work. Those driver type is commonly used for prototyping or when there is no JDBC driver available for a particular DatabaseManagementSystem.

Statement:

This is the base statement objects that provides methods to execute. SQL statement that perform some kind of work. Before executing an SQL statements directly against the database. The statement objects that provides the

methods to execute SQL statements directly against the database .The statement objects is useful in creating one time queries and DDL statements such as CREATE TABLE ,DROP TABLE AND SO ON.


Specific Requirements

SPECIFIC REQUIREMENTS

3.1 Functional Requirement:

3.1.1 Introduction:

Inputs are normally entered into the forms that are available in various modules.

3.1.2 List of Inputs:

- Registering the new User.
- Registering the new System.
- Registering the new Lab.
- Sending messages to the Users.
- Registering the Complaints.

3.1.3 Information Processing Required:

The Information that are entered in the form of field has to be validated and stored in the database.

PERFORMANCE REQUIREMENTS

Security:

Security is very important for any software .A software project without security is useless. Security is a specially necessary in a network .In a multi-user system ,many users will be using the software .Hence it should be protected for intentional and accidental damage. To implement security each user is given a userid and password. After validating the user's id and password only he is allowed to login inside the system

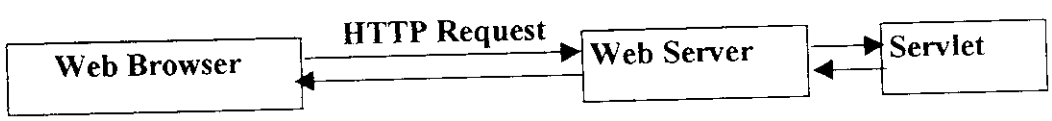
3.2.1 Availability:

Servlets are server side applets.Servlets are loaded & executed by a Web server in the same manner that applets are loaded and executed by a Web browser.

Servlets are a replacement for Common Gateway Interface(CGI) scripts. Since servlets are written in Java, they are platform independent .

The advantages of using Servlets are given below:

- ✓ Servlets are persistent. Servlets are loaded once by the Web server and can maintain services between the Web server each time a request is made to it.
- ✓ Servlets are fast. They offer better performance since they need to be loaded only once.
- ✓ Servlets are platform independent as they are written in Java.
- ✓ Servlets are extensible. All the benefits of Java can be brought into our servlet.
- ✓ Servlets are secure. The only way to gain access to a servlet is through a server. If the server is protected, the servlet is protected as well.
- ✓ Servlets can be used with a variety of clients.
- ✓ A servlet accepts a request from a client the Web server, performs some task and returns the results.



HTTPResponse:

The following steps describe the basic flow while using servlets.

- ❖ The client (Web browser) makes a request via an HTTP.

- The Web server receives the request and sends it to the servlet. If the servlet has not yet been loaded, the Web server will load it into the Java machine and execute it.
- The servlet will receive the HTTP request and perform tasks.
- The servlet will return a response to the Web server.
- The web server will forward the response to the client.

Since the servlet executes on the server, the security issues usually associated with applets do not apply. The Web browser also not communicate directly with a servlet, the servlet is loaded & executed by the Web server. This implies that if the web server is secure , then the servlet is also secure.

The Java Servlet Development Kit (JSDK) is used to write servlets. The **Java Web server** from JavaSoft is used to run servlets.

HTML:

HTML stands for Hyper Text Markup Language, which is an application of Standard Generalized Markup Language(SGML).It is a simple language used to defined and describe the layout of a Web page.HTML also supports Multimedia and document links.

HTML consists of special which when embedded in text, adds formatting. The special characters, which separate HTML from ordinary text, are

the left and right brackets. (< >) .These brackets contain instructions known as TAGS that are not case sensitive.

JAVA SCRIPT:

We the advent of Internet and WWW, interactive communication At an exigency has become a necessity for mankind. The WWW is a cluster of pages of information, combining text, pictures and sounds. Each page has a hyperlink that refers to system of interlinked documents is called as **hypertext.**

Browsers help in manipulating pages of hypertext information To enhance interactivity between pages. One such browser is called Netscape Navigator.

This Browser was developed by Netscape Communications, which incorporates Java, a Programming language from Sun Micro Systems & Java Script, a scripting language used to enhance the functionality of the browser. Netscape offers a host tools to add value to their applications. These tools aid Web pages developers and authors in adding dynamic interaction to the information provided on the Internet.

JavaScript is integrated with HTML and Navigator .JavaScript facilitates the developer with properties related to documents windows, frames, forms, loaded documents & links. This scripting language also tarps user events so programs can be developed for such events. This is an interpreter based language and source code files are directly executed at runtime. JavaScript includes built-in objects related to the current windows & documents as well as objects such as Math, String and Date that contain mathematical functions, string functions and date functions respectively. Since JavaScript in an Object based language, it supports instances, methods and properties.

DESIGN CONSTRAINT

System Design is a modelling process .It can be defined as a transition from a user's view to view of programmers (developers) and the database personnel. It concentrates on translating requirements specification to design specification .This system design methodology and developed ,input design phase acts as a bridge between the requirements and the implementation phase.

The major steps in the design phase are design methodology and development input design ,output design, database design. First step is design methodology and development ,which specifies various steps in systems development life cycle .Next step is to define input and output screens and database on choosing the database that suits most to the application environment.

Input Design:

In input design the user oriented inputs are converted into computer recognizable forms. The collection of inputs data is the most expensive part of the system. In the input design data is accepted and it can be readily used for data processing or can be stored in database for further use. Input design is that part of design phase which requires the most attention .Data should be accurate because in accurate data is the most common cause for errors in data processing

.The input screens are very user friendly .Different names are screens and data item that makes data entry an easy job.

Each data entry screen contain separate button "SUBMIT" and "RESET" for submitting the form. While entering the data, proper validation checking are carried out and the necessary message will alerted by software if incorrect data has been entered.

Database Design:

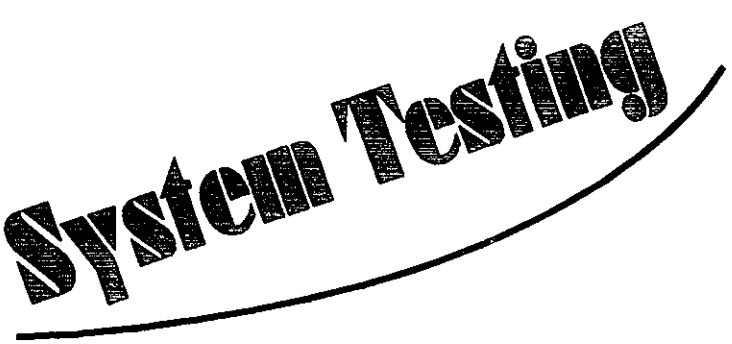
Database design is a important part of the system design phase .In a database environment ,the available data is used by several users .Instead of each program managing its own data, authorized users share data across application with database software managing the data as an entity. The primary objective of the database design include fast response time to inquiries ,more information at low cost, control of redundancy, clarity and integrity of the system, fast recovery and availability of a powerful end-user language. The theme behind the database is to handle information on the whole integrated thus making access to information easy, quick in expensive flexible for the users.

Data directory specifies the major elements in the system and cause should be taken while designing in order to avoid unnecessary duplication of data. The entire package depends on how the tables required for processing of various

data as well as storing intermediate modules to meet their needs access the stored data.

Output Design:

In the output design the emphasis is on producing the hardcopy as softcopy of information requested for output are the most important and direct source of informtion to clients.Intelligent and formatted outputs will make it easier to understand .outputs Are also provide a permanent hardcopy of results for later consideration.


System Testing

SYSTEM TESTING

This forms another major part of any system development process care should be given during the whole process of testing. The performance of the system is measured in this phase.

Source Code Testing:

Testing is a process of executing a program with the interest of finding an error. A good test is one that has a high probability of finding the yet undiscovered error. Testing should systematically uncover different classes of errors in a minimum amount of time with minimum effort. Two classes of inputs are provided to the test process. They are

a) A software configuration that includes a software requirement specification, a design specification, and a source code.

b) A test configuration that includes a test plan and procedure, any testing tools that are to be used and test cases and their expected results.

Testing is divided into three distinct operations namely modular testing, integration testing and system testing. In the series of testing the following tests are implemented.

Integration Testing:

Though each program works individually, they should work after linking them together. This is also referred to as interfacing. Data may be last across interface and one module can have an adverse effect on another. Subroutines, after linking, may not do the desired function expected by the main routine. Integration testing is a systematic technique for constructing a program structure while at same time, conducting test to uncover errors associated with the interface. In the testing, the programs are constructed and tested in small segments.

Data Validation Testing:

Data validation is done to see whether the corresponding entries made in the tables are correct. Proper validation checks are done in case of insertion, deletion and updation of tables. Duplication of data must be avoided to the maximum extent. If any such cases arise, then proper error message or warnings, if any, has to be displayed. A double confirmation is made before deleting any specific entries. Whitebox testing is a test case design to divide the test cases. The different test cases are

1) Guarantee that all independent parts within a module have been exercised at least once.
2) Exercise all logical decisions on their true/false side.

3) Execute all loops at their boundaries and within their operational bounds.

4) Exercise internal data structure to ensure their validity.

Each module was tested and the tested modules were linked and integration was carried out.

Test Data:

The system analyst will provide the test data, specially designed to show that the system will operate successfully in all its aspects and produce expected results under expected conditions. The test should take place at the same environment preparation of test data and the checking of results should be carried out in conjunction with the appropriate users and operational departments. The test objectives should be clear. Also the extend to which the system should be tested must be planned.

Debugging:

The potential ability of JAVA to handle exceptions was used extensively during the debugging process. All types of exceptions were caught and explicitly handled. Java language exceptions were caught generally whereas other exceptions like SQL exceptions were caught separately and various modes of their validation were found. Errors in case of back-end tables were used to display to

their user in a number of ways. Exceptions occur at many stages like what happens during start of a program to any abnormal operations done or any missing threads. Various errors occurred in the cases of variables, which carried same name and caused a lot of problems when all modules were linked.

Conclusion

CONCLUSION

The project deals with user-interactive on-Line system working under any platform. It provides facilities to maintain and manipulate data, retrieve information and to create report quickly and effectively.

References

REFERENCES

JAVA:

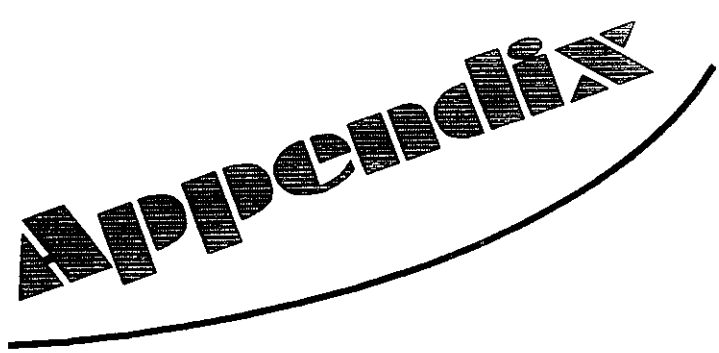
#1. James Goodwill ,"**Servlets**"Sam's-2nd Edition.

#2. Weber, "**Using Java-2 Platform**" Prentice-hall Edition.

#3. Wyke , Gilliam & Ting, "**Pure Java Script**" Sam's -2nd Edition.

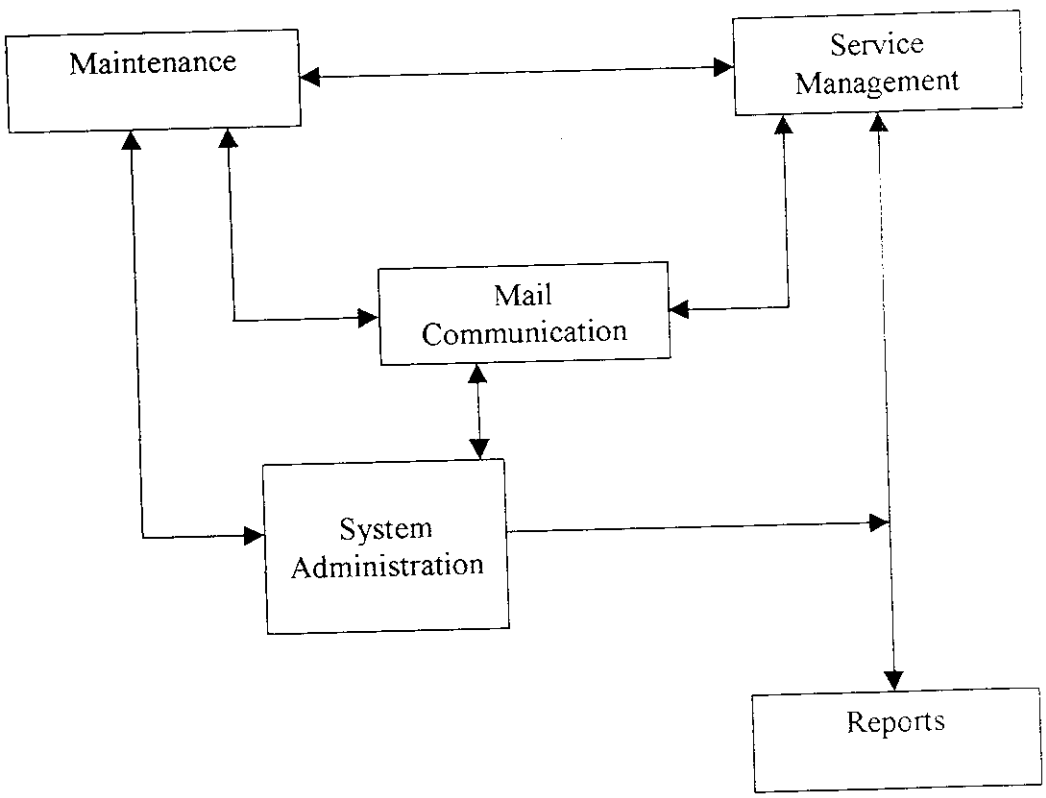
RDBMS:

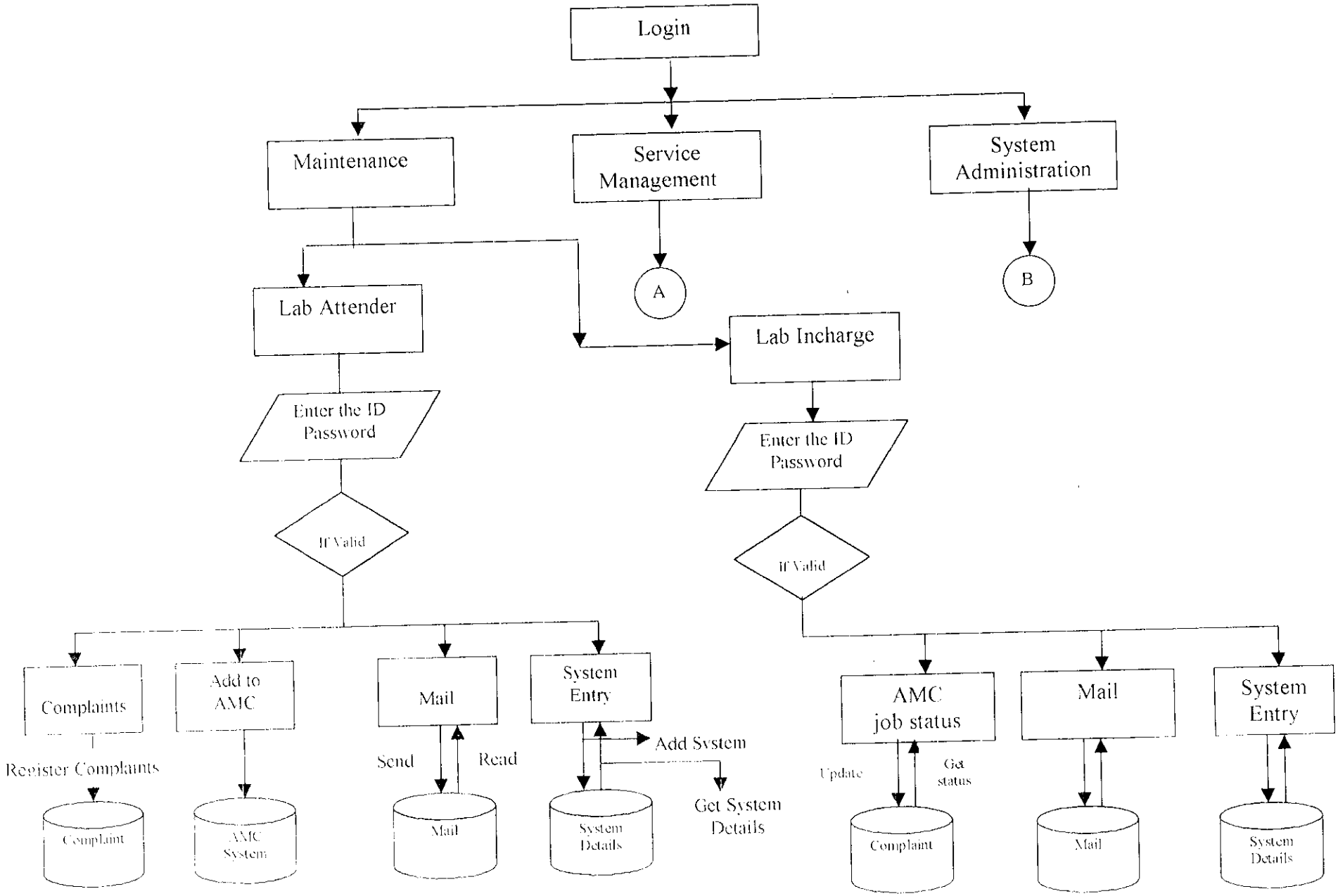
#1. ***Oracle 8i*** The Complete Reference, Kevin Loney & George Koch

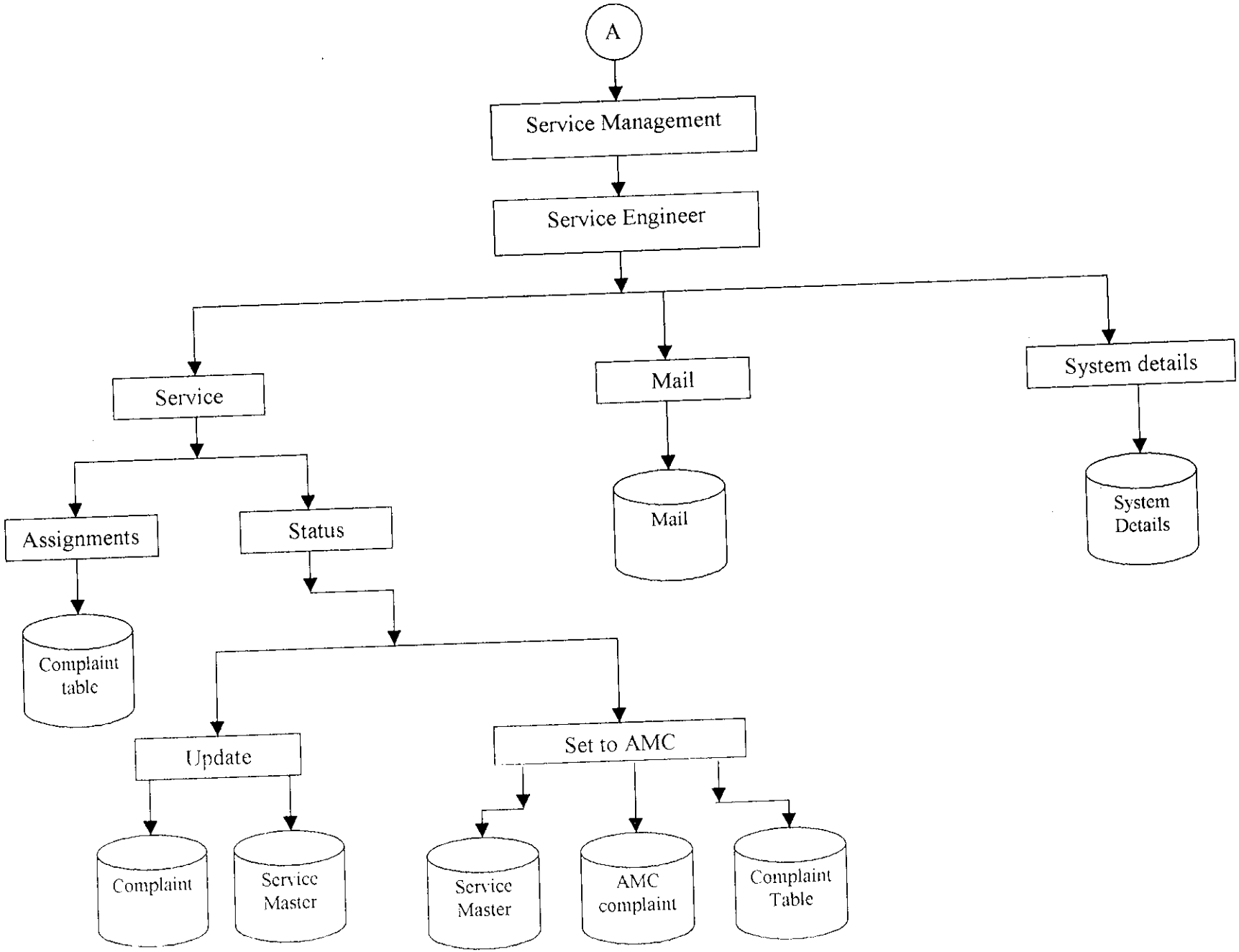

Appendix

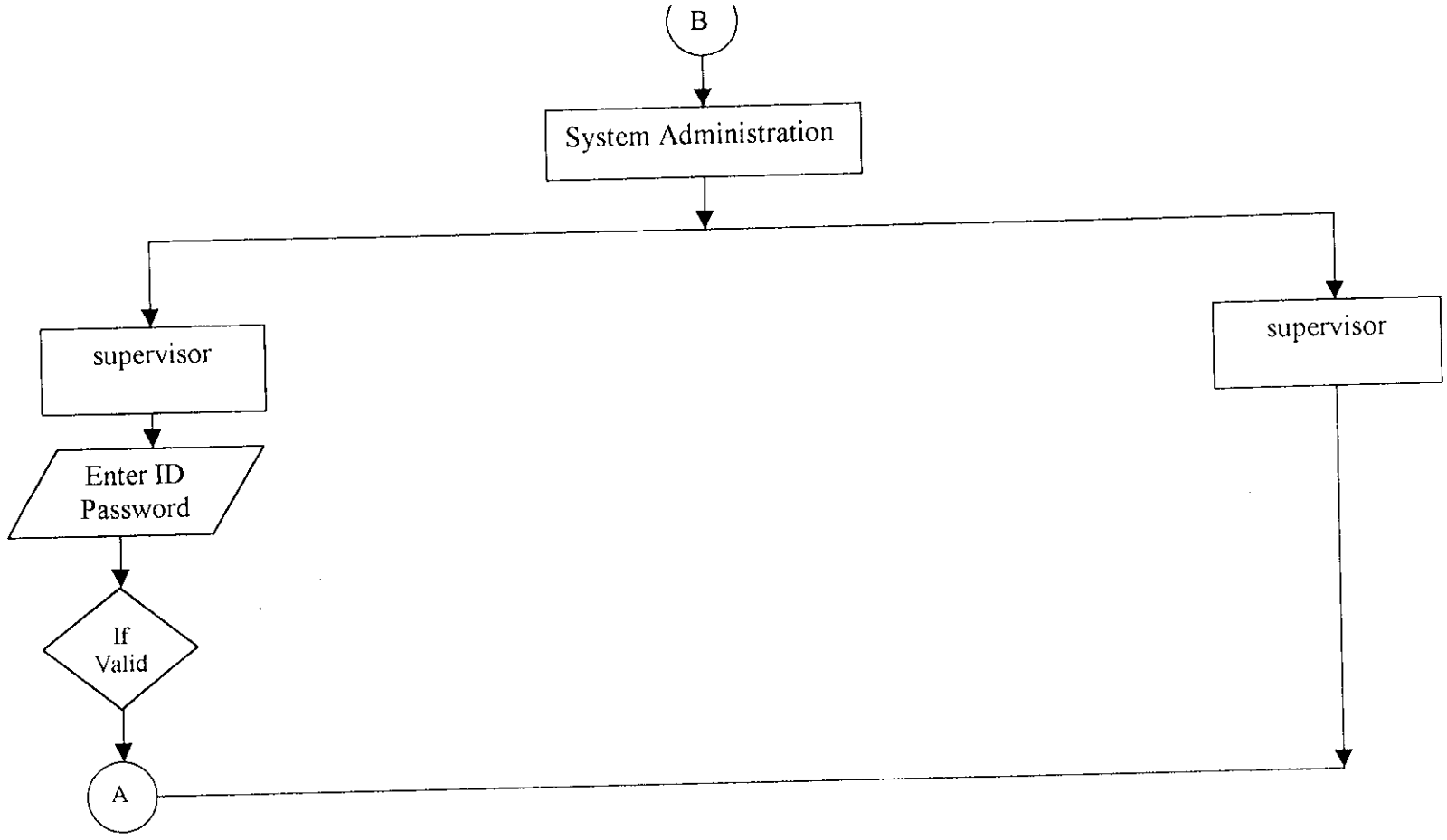
Flow Charts

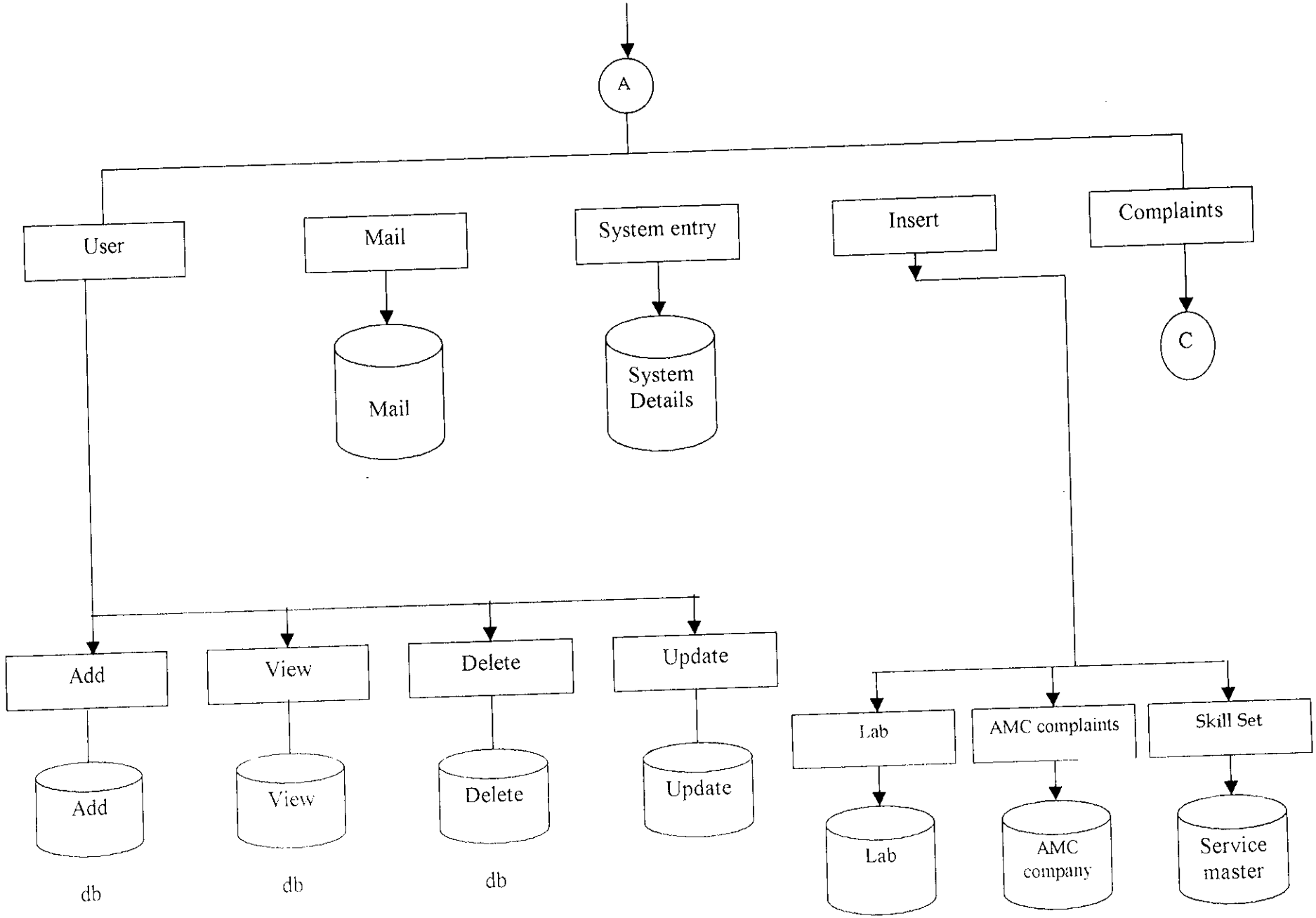
APPENDIX

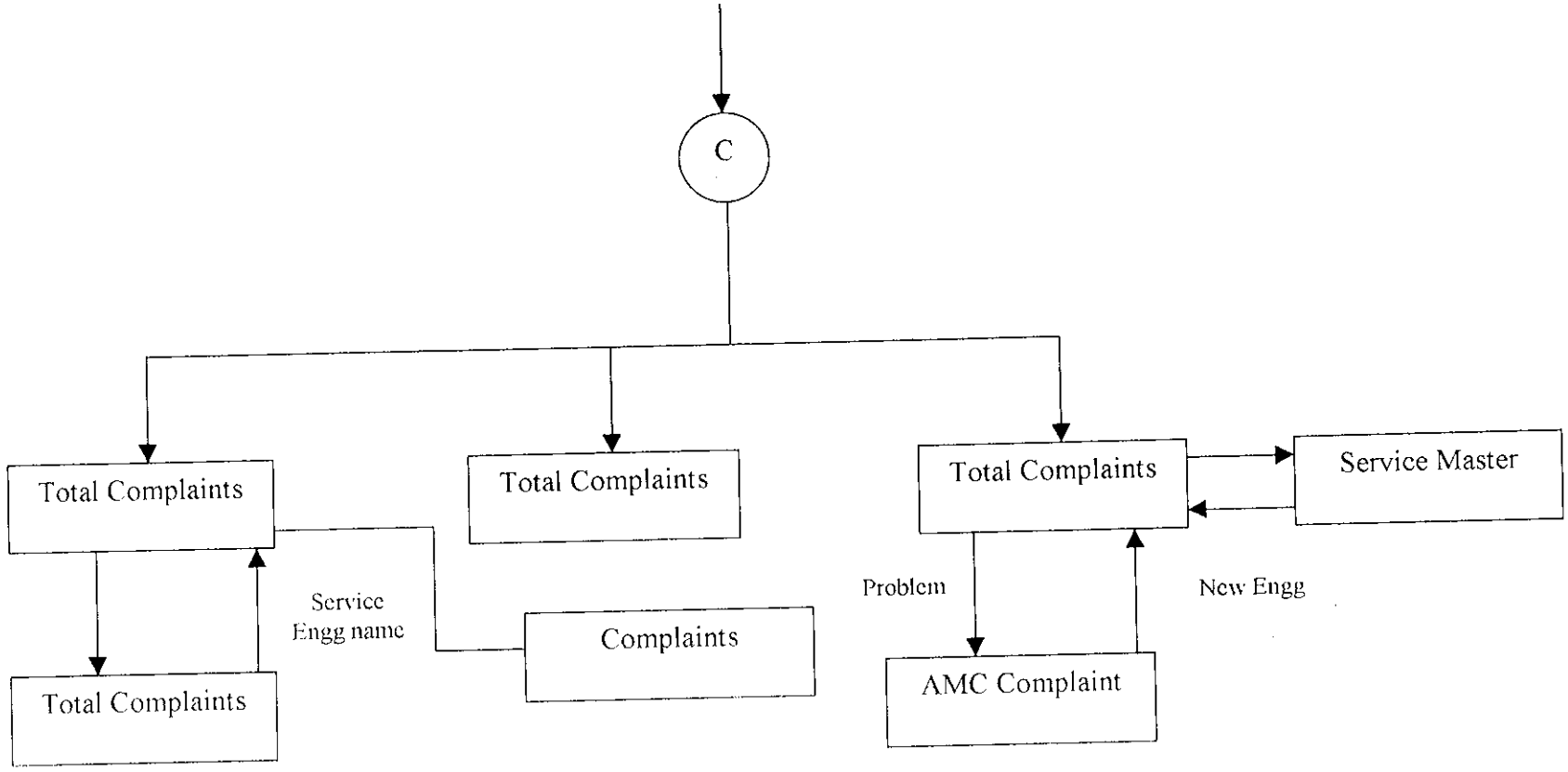

Maintenance
Service Management
Mail Communication
System Administration
Reports

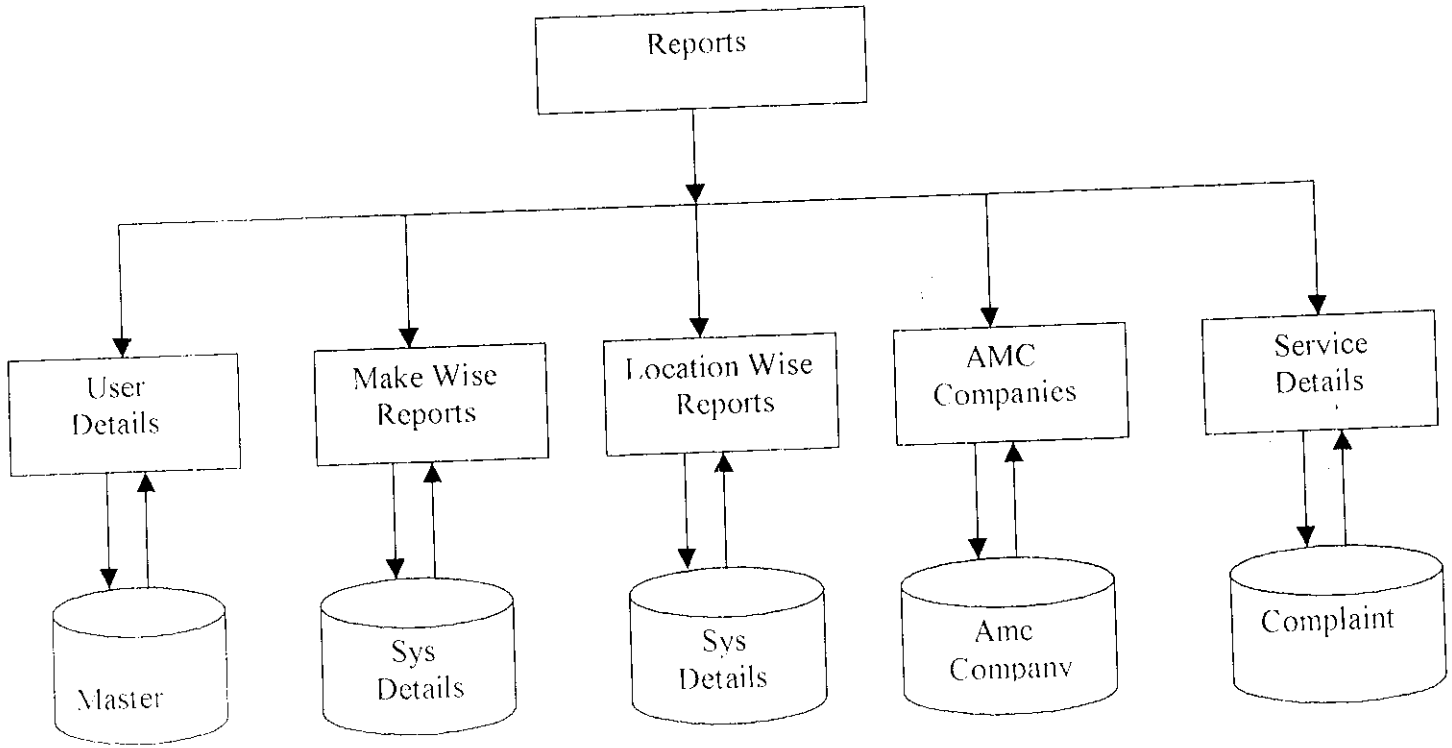

Login
Maintenance
Service Management
System Administration
A
B
Lab Attender
Lab Incharge
Enter the ID Password
Enter the ID Password
If Valid
If Valid
Complaints
Add to AMC
Mail
System Entry
AMC job status
Mail
System Entry
Register Complaints
Send
Read
Add System
Get System Details
Update
Get status
Complaint
AMC System
Mail
System Details
Complaint
Mail
System Details


A
Service Management
Service Engineer
Service
Mail
System details
Assignments
Status
Mail
System Details
Complaint table
Update
Set to AMC
Complaint
Service Master
Service Master
AMC complaint
Complaint Table


B
System Administration
supervisor
Enter ID
Password
If
Valid
A
supervisor


A
User
Mail
System entry
Insert
Complaints
Mail
System Details
C
Add
View
Delete
Update
Lab
AMC complaints
Skill Set
Add
View
Delete
Update
Lab
AMC company
Service master
db
db
db


C
Total Complaints
Total Complaints
Total Complaints
Service Master
Service
Engg name
Complaints
Problem
New Engg
Total Complaints
AMC Complaint


Reports
User Details
Make Wise Reports
Location Wise Reports
AMC Companies
Service Details
Master
Sys Details
Sys Details
Amc Company
Complaint

TABLES

TABLE-NAME: AMC

FIELD-NAME	DATA-TYPE
COMPANY	TEXT(30)
CON-DATE	DATE
DUE-DATE	DATE
ADDRESS	TEXT(50)
EMAIL	TEXT(50)
PHONE	NUMBER(15)
FAX	NUMBER(15)

TABLE-NAME: AMC COMPLAINTS

FIELD-NAME	DATA-TYPE
DEPARTMENT	TEXT(20)
LAB	TEXT(20)
SYSNO	TEXT(20)
PROBLEM	TEXT(20)
COMPANY	TEXT(20)
ATTENDDATE	DATE
STATUS	TEXT(15)

TABLE-NAME: AMC SYSTEMS

FIELD-NAME	DATA-TYPE
DEPARTMENT	TEXT(20)
LAB	TEXT(20)
SYSNO	TEXT(10)
COMPANY	TEXT(30)

TABLE NAME: COMPLAINT

FIELD-NAME	DATA-TYPE
LABATTENDERNAME	TEXT(20)
DEPARTMENT	TEXT(20)
LAB	TEXT(20)
SYSNO	TEXT(20)
CALLDATE	DATE
PROBLEM	TEXT(20)
REMARKS	MEMO
ENGNAME	TEXT(20)
ATTENDDATE	DATE
STATUS	TEXT(20)

TABLE NAME:LOGIN

FIELD-NAME	DATA-TYPE
CATEGORY	TEXT(20)
PID	TEXT(20)
PASSWORD	TEXT(20)

TABLE NAME: LAB

FIELD-NAME	DATA-TYPE
DEPARTMENT	TEXT(20)
LAB	TEXT(20)

TABLE NAME: MAIL

FIELD-NAME	DATA-TYPE
MAIL-NO	AUTO NUMBER
SENDER-CATEGORY	TEXT(20)
SENDER-NAME	TEXT(20)
TO	TEXT(20)
READER-CATEGORY	TEXT(20)
SUBJECT	TEXT(20)
CONTENT	MEMO

TABLE-NAME: MASTER

FIELD-NAME	DATA-TYPE
FIRST-NAME	TEXT(20)
LAST-NAME	TEXT(20)
ADDRESS1	TEXT(50)
ADDRESS2	TEXT(50)
PHONE	NUMBER(20)
CITY	TEXT(20)
ZIP	NUMBER(20)
STATE	TEXT(20)
CATEGORY	TEXT(20)
PID	TEXT(20)
PASSWORD	TEXT(20)
REGISTER-DATE	DATE

TABLE-NAME: SERVICE MASTER

FIELD-NAME	DATA-TYPE
NAME	TEXT(20)
ABLITY	TEXT(20)
JOB	NUMBER(3)

TABLE-NAME: SYSDETAILS

FIELD-NAME	DATA-TYPE
DEPARTMENT	TEXT(20)
LAB	TEXT(20)
ADDED-BY	TEXT(20)
SYSNO	TEXT(20)
PROCESSOR	TEXT(20)
RAM	TEXT(20)
FDD	TEXT(20)
HDD	TEXT(20)
MONITOR	TEXT(20)
KEYBOARD	TEXT(20)
MOUSE	TEXT(20)
SOUNDCARD	TEXT(20)
CDDRIVE	TEXT(20)
MACHINENO	TEXT(20)
MONITORNO	TEXT(20)
PROVIDER	TEXT(20)

SOURCE CODE

LOGIN:

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import sun.jdbc.odbc.*;
import java.sql.*;


public class login extends HttpServlet
{
Connection con=null;
Statement st=null;
ResultSet rs=null;
String cata;


public void init(ServletConfig config)throws ServletException
{
super.init(config);
}
public void doPost(HttpServletRequest req,HttpServletResponse res) throws
ServletException,IOException
```

```
{
res.setContentType("text/html");
PrintWriter pw=res.getWriter();
try
{
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
}
catch(Exception e)
{
System.out.println("eror in driver");
}
try
{
con=DriverManager.getConnection("jdbc:odbc:pro"," "," ");
}
catch(Exception e)
{
System.out.println("error in odbc driver:");
}
try
{
```

```
String name=req.getParameter("user");
String pass=req.getParameter("pass");
cata=req.getParameter("cat");


String a=new String("hello");
String b=new String("hello1");

Cookie u=new Cookie(a.name);
res.addCookie(u);

Cookie c=new Cookie(b,cata);
res.addCookie(c);

st=con.createStatement();

rs=st.executeQuery("select pid  from login  where pid='"+name+"' and
pass='"+pass+"' and category='"+cata+"' ");
 if(rs.next())
 {
 if(cata.equals("Supervisor"))
 {
```

```
pw.println("<html>");
pw.println("<frameset rows='21%,*' bordercolor='#152c76' border=5>");
pw.println("<frame src='D:/supervisor.html' noresize >");
pw.println("<frameset cols='19%,*'>");
pw.println("<frame src='D:/supervisormenu.html' target='display' noresize>");
pw.println("<frame name='display' src='D:/display.html' noresize>");
pw.println("</frameset>");
pw.println("</frameset>");
pw.println("</html>");
}
else if(cata.equals("Management"))
{
pw.println("<html>");
pw.println("<frameset rows='21%,*' bordercolor='#152c76' border=5>");
pw.println("<frame src='D:/manager.html' noresize>");
pw.println("<frameset cols='19%,*'>");
pw.println("<frame src='D:/managermenu.html' target='display' noresize>");
pw.println("<frame name='display' src='D:/display.html' noresize>");
pw.println("</frameset>");
pw.println("</frameset>");
pw.println("</html>");
}
```

```
else if(cata.equals("Lab-Attender"))
{
pw.println("<html>");
pw.println("<frameset rows='21%,*' bordercolor='#152c76' border=5>");
pw.println("<frame src='D:/lab.html' noresize>");
pw.println("<frameset cols='19%,*'>");
pw.println("<frame src='D:/attendermenu.html' target='display' noresize>");
pw.println("<frame name='display' src='D:/display.html' noresize>");
pw.println("</frameset>");
pw.println("</frameset>");
pw.println("</html>");
}

else if(cata.equals("Lab-Incharge"))
{
pw.println("<html>");
pw.println("<frameset rows='24%,*' bordercolor='#152c76' border=5>");
pw.println("<frame src='D:/labincharge.html' noresize>");
pw.println("<frameset cols='19%,*'>");
pw.println("<frame src='D:/inchargemenu.html' target='display' noresize>");
pw.println("<frame name='display' src='D:/display.html' noresize>");
pw.println("</frameset>");
```

```
pw.println("</frameset>");
pw.println("</html>");
}
 else
 {
   pw.println("<html>");
pw.println("<frameset rows='24%,*' bordercolor='#152c76' border=5>");
pw.println("<frame src='D:/service.html' noresize>");
pw.println("<frameset cols='19%,*'>");
pw.println("<frame src='D:/servicemenu.html' target='display' noresize>");
pw.println("<frame name='display' src='D:/display.html' noresize>");
pw.println("</frameset>");
pw.println("</frameset>");
pw.println("</html>");
  }
 }
 else
 {
  pw.println("<html><script>alert('INVALID USER,PLEASE CHECK THE
  USER NAME AND PASSWORD')</script></html>");
  pw.println("<html><script
  language=javaScript>history.back()</script></html>");
```

```
}
pw.close();
}
catch(Exception e)
{
System.out.println("error in checking");
}
}
}
```

REPORTS:

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import java.sql.*;
import sun.jdbc.odbc.*;

public class  companywise  extends HttpServlet
{
Connection con=null;
Statement st=null;
Statement st1=null;
ResultSet rs=null;
ResultSet rs1=null;

public void init(ServletConfig config)throws ServletException
{
super.init(config );
}

public void doPost(HttpServletRequest req,HttpServletResponse res)throws
ServletException,IOException
```

```
{
res.setContentType("text/html");
PrintWriter pw=res.getWriter();
try
{
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
}
catch(Exception e)
{
System.out.println("error in driver:");
}
try
{
con=DriverManager.getConnection("jdbc:odbc:pro","","");
}
catch(Exception e)
{
System.out.println("error in odbc:");
}
try
{
st=con.createStatement();
```

```
st1=con.createStatement();
rs=st.executeQuery("select distinct company from amc");
pw.println("<html><body text=blue><h2 align='center'>AMC COMPANY
DETAILS</h2></body></html>");
while(rs.next())
{
String s=rs.getString(1);
s=s.toUpperCase();
System.out.println(s);
pw.println("<html><body text=blue><br><hr  size=1 color='red'><h4
align='center'> <font size=4><b>COMPANY NAME: "+s+"</b></font></h4>");
rs1=st1.executeQuery("select condate,duedate,address,email,phone,fax from amc
where company='"+s+"' ");
pw.println("<table border='2'width='100%'>");
pw.println("<col align=center><colgroup span=1 align=center valign=bottom>");
pw.println(" <th>CONTRACT-DATE</th> <th>DUE-DATE</th>");
pw.println(" <th>ADDRESS</th> <th>EMAIL</th>");
pw.println(" <th>PHONE</th> <th>FAX</th>");
while(rs1.next())
{
pw.println("<tr><td>"+rs1.getString(1)+"</td>");
pw.println("<td>"+rs1.getString(2)+"</td>");
```

```
pw.println("<td>"+rs1.getString(3)+"</td>");
pw.println("<td>"+rs1.getString(4)+"</td>");
pw.println("<td>"+rs1.getString(5)+"</td>");
pw.println("<td>"+rs1.getString(6)+"</td></tr>");
}
pw.println("</table>");
}
pw.println("</body></html>");
}
catch(Exception e)
{
System.out.println("error in sql");
}
}
}
```

ASSIGNMENTS:

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import sun.jdbc.odbc.*;
import java.awt.*;
import java.sql.*;

public class postcomp extends HttpServlet
{
Connection con=null;
Statement st=null;
Statement st1=null;
Statement st2=null;
Statement st3=null;
Statement st4=null;
ResultSet rs=null;
ResultSet rs1=null;
ResultSet rs2=null;
String cata;
```

```
public void init(ServletConfig config)throws ServletException
{
super.init(config);
}

public void doPost(HttpServletRequest req,HttpServletResponse res) throws
ServletException,IOException
{
res.setContentType("text/html");
PrintWriter pw=res.getWriter();
try
{
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
}
catch(Exception e)
{
System.out.println("eror in driver");
}
try
{
con=DriverManager.getConnection("jdbc:odbc:pro"," "," ");
}
```

```
catch(Exception e)
{
System.out.println("error in odbc driver:");
}
try
{
st=con.createStatement();
st1=con.createStatement();
st2=con.createStatement();
st3=con.createStatement();
st4=con.createStatement();
String name=req.getParameter("name");
String dept=req.getParameter("Dept");
String lab=req.getParameter("lab");
String no=req.getParameter("no");
String prob=req.getParameter("prob");
String remarks=req.getParameter("remarks");
String date=req.getParameter("date");
rs2=st4.executeQuery("select Dept,Lab,sysno,problem from complaint where
Dept='"+dept+"' and Lab='"+lab+"' and sysno='"+no+"' and problem='"+prob+"'
and status='InComplete' ");
System.out.println("hello");
```

```
if(rs2.next())
{
pw.println("<html><body text='blue'><br><br><br><br><br><br><h2
align='center'>THANKS,THIS COMPLAINT IS ALREADY
REGISTERED</h2></body></html>");
}
else
{
rs1=st3.executeQuery("select department from sysdetails  where
department='"+dept+"' and labno='"+lab+"'and sysno='"+no+"' ");
if(rs1.next())
{
rs=st1.executeQuery("select name from servicemaster where ablity='"+prob+"'
and job=(select min(job) from servicemaster where ablity='"+prob+"')");
String engname;
if(rs.next())
{
engname=rs.getString(1).toUpperCase();
System.out.println(engname);
int i=st.executeUpdate("insert into
complaint(LabAtname,dept,lab,sysno,calldate,problem,remarks,engname)
```

```
values('"+name+"','"+dept+"','"+lab+"','"+no+"','"+date+"','"+prob+"','"+remarks+"
','"+engname+"')");
int j=st2.executeUpdate("update servicemaster set job=job+1 where
name='"+engname+"' and ablity='"+prob+"' ");
pw.println("<html><body text='blue'><br><br><br><br><br><h2
align='center'>YOUR COMPLAINT HAVE BEEN
REGISTERED.<br><br><font color='red'>"+engname+" </font>WILL BE
ATTENDING THIS JOB</H2></body></html>");
}
}
else
{
 pw.println("<html><body text='blue'><br><br><br><br><br><h3
 align='center'>PLEASE MAKE SURE THAT YOU HAVE SPECIFIED THE
 CORRECT SYSTEM-NO</H3></body></html>");
 }
 }
 pw.println("<html><body><br><br><br><a href
 ='http://localhost:8080/servlet/register'
 target='display'>back</a></body></html>");
 pw.close();
 }
```

```
catch(Exception e)
{
 pw.println("<html><script>alert('INCOMPLETE DATA.PLEASE FILL IN
EVERY DATA')</script></html>");
 pw.println("<html><script
language=javaScript>history.back()</script></html>");
}
}
}
```

AMC DETAILS:

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import java.sql.*;
import sun.jdbc.odbc.*;


public class  comdetails  extends HttpServlet
{
Connection con=null;
Statement st=null;
ResultSet rs=null;

public void init(ServletConfig config)throws ServletException
{
super.init(config );
}

public void doPost(HttpServletRequest req,HttpServletResponse res)throws
ServletException,IOException
```

```
{
res.setContentType("text/html");
PrintWriter pw=res.getWriter();
try
{
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
}
catch(Exception e)
{
System.out.println("error in driver:");
}
try
{
con=DriverManager.getConnection("jdbc:odbc:pro","","");
}
catch(Exception e)
{
System.out.println("error in odbc:");
}
try
{
st=con.createStatement();
```

```
String dept=req.getParameter("dept");
String lab=req.getParameter("lab");
String sysno=req.getParameter("sysno");
rs=st.executeQuery("select problem,Remarks from complaint where
Dept='"+dept+"' and Lab='"+lab+"' and sysno='"+sysno+"' ");
String problem,remark;
if(rs.next())
{
pw.println("<html><script language='javascript'>function fdate(){");
pw.println("d=new Date();da=d.getDate();m=d.getMonth()+1;y=d.getYear();");
pw.println("document.comdetails.date.value=" + da +"/"+ m +"/"+ y;}");
pw.println("</script>");
pw.println("<body text=blue onLoad='fdate()'><h1 align='center'>UPDATE THE
STATUS</h1><br><br><br><br>");
pw.println("<form name='comdetails'
action='http://localhost:8080/servlet/statusupdate' method='post'>");
problem=rs.getString(1);
remark=rs.getString(2);
pw.println("<table border=0  width=70% align='center'>");
pw.println("<tr><td>PROBLEM</td><td><input type='text' name='problem'
value="+problem+" size=20 ></td></tr>");
```

```
pw.println("<tr><td>REMARKS</td><td><input type='textarea' cols='40'
rows='10' name='remark' value="+remark+" size=20 ></td></tr>");
pw.println("<tr><td>ATTENDED ON</td><td><input name='date'
size=20></td></tr>");
pw.println("<tr><td>STATUS</td><td><select name='status'
style={width:154px}>");
pw.println("<option value='InComplete'>InComplete</option>");
pw.println("<option value='Completed'>Completed</option>");
pw.println("</select></td></tr>");
pw.println("<tr></tr><tr></tr><tr></tr><tr></tr><tr></tr><tr></tr><tr></tr><tr></tr><tr>
</tr>");
pw.println("<tr><td> </td><td><input type='hidden'name 'dept'
value='"+dept+"'></td></tr>");
pw.println("<tr><td> </td><td><input type='hidden'name='lab'
value='"+lab+"'></td></tr>");
pw.println("<tr><td> </td><td><input type='hidden'name='sysno'
value='"+sysno+"'></td></tr>");
pw.println("<tr><td><input type=submit value=' UPDATE
'></td></form><form name='comdetails1'
action='http://localhost:8080/servlet/settoamc' method='post'>");
pw.println("<input type='hidden'name='dept' value='"+dept+"'><input
type='hidden'name='problem' value='"+problem+"'><input
```

```
type='hidden'name='lab' value='"+lab+"'><input type='hidden'name='sysno'
value='"+sysno+"'>");
pw.println("<td><input type=submit value='SET TO AMC'></td></tr>");
pw.println("</table></form></body></html>");
}
}
catch(Exception e)
{
System.out.println("error in inserting:");
}
}
}
```

SERVICE ROOM

OPTIONS

ASSIGNMENTS
READ MESSAGE
SEND MESSAGE
UPDATE STATUS
SYSTEM DETAILS

LOG OUT

YOUR ASSIGNMNETS TILL DATE ARE

LAB-ATTENDER	DEPARTMENT	LAB-NO	SYSTEM-NO	PROBLEM	REMARKS
SACHIN	EEE	LAB-1	22	POWER	LOOSECONTACT
PRAKASH	CIVIL	LAB-2	13	HARDDISK	CANNOT SAVE
NISHANTH	MSc.C.T	LAB-1	34343	SMPS	NO POWER
ROBIN	CIVIL	LAB-2	56	HARDDISK	CRASH
SAM	EEE	LAB-1	23	POWER	NO POWER
SHIVA	MCA	LAB-3	7	SMPS	NO POWER TO CPU

- ANALYSIS
- ADD A LAB
- ADD A USER
- VIEW USERS
- REALLOCATE
- COMPLAINTS
- DELETE USERS
- UPDATE A USER
- SEND MESSAGE
- READ MESSAGE
- ADD SKILL SETS
- AMC CONTRACTS
- SYSTEM DETAILS

LOG OUT

K.C.T

THE CENTER OF EXCELLENCE

Our bag is finding you great stuff

LAB ROOM

OPTIONS

COMPLAINTS

ADD TO AMC

SEND MESSAGE

READ MESSAGE

ADD A SYSTEM

SYSTEM DETAILS

LOG OUT

COMPLAINTS REGISTER

NAME A

DEPARTMENT EEE

LAB-NAME L1

SYSTEM-NO

DATE 11/3/2001

PROBLEM HARD-DISK

REMARKS

submit

OPTIONS

ASSIGNMENTS

READ MESSAGE

[illegible]

UPDATE STATUS

SYSTEM DETAILS

LOG OUT

SEND YOUR MESSAGES

To:

Category: Supervisor

Subject:

Contents:

SEND MAIL

Our bag is finding you great stuff

STAFF ROOM

OPTIONS

AMC STATUS

READ MESSAGE

SEND MESSAGE

ADD A SYSTEM

SYSTEM DETAILS

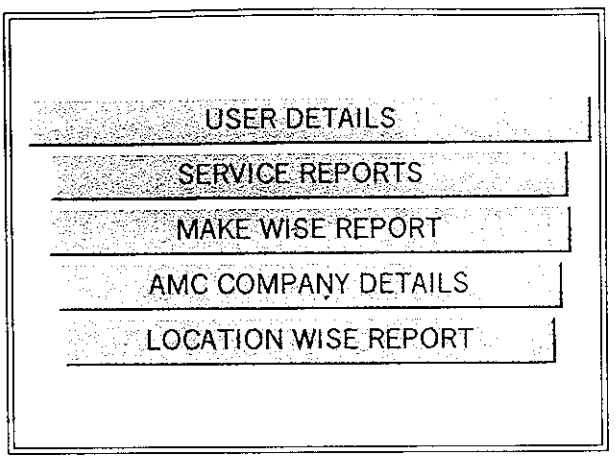
LOG OUT

SELECT THE DEPARTMENT

DEPARTMENT EEE

VIEW

Reports


USER DETAILS
SERVICE REPORTS
MAKE WISE REPORT
AMC COMPANY DETAILS
LOCATION WISE REPORT


LIBRARY