DEPARTMENTAL RESOURCE MANAGER

Enrollment No	**071293**	**071312**
Name of student	**Ashish kapoor**	**Ankur vashisth**
Name of supervisor	**Mr. Yashwant singh**	

Submitted in partial fulfillment of the Degree of Bachelor of Technology

DEPARTMENT OF COMPUTER SCIENCE

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY,

WAKNAGHAT

Table of contents

CERTIFICATE

This is to certify that the work titled "**DEPARTMENTAL RESOURCE MANAGER**" submitted by "**ASHISH KAPOOR and ANKUR VASHISTH**" in partial fulfillment for the award of degree of **B-TECH** of Jaypee University of Information Technology, Waknaghat has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.

Signature of Supervisor
Name of Supervisor MR. YASHWANT SINGH
Designation
Date 16th May 2011

ACKNOWLEDGEMENT

We take this opportunity to express our profound sense of gratitude and respect to all those who have helped us throughout our training period on the project.

This project would not have been possible without the valuable guidance of our Project Guide His enthusiasm, valuable suggestions & encouragement enabled us to handle a project of this enormity with confidence.

We are indebted to our Project Guide for his timely advice and valuable help throughout our project. We have learned a lot during the course of the project and definitely looking forward to continue the excellent rapport I shared.

We are thankful to the members of the staff at our university for their constant encouragement and the valuable inputs from time to time throughout the course of our project.

Signature of the student   Ashish   Ankur

Name of Students   ASHISH KAPOOR   ANKUR VASHISTH

Date   16th May 2011

SUMMARY

Departmental resource management is the active control program which helps a particular department to make good use of their available resources departmental resource management software helps a particular department to know about the available resources(like in our project if any student wants to use a computer which has redhat (linux) operating system then the student will get the information from our project).

This project is developed to automate the existing manual system into a computerized system so that the complexity in the working of manual system can be reduced and the time can be saved. The overall motive of this project is to convert the existing manual system into a computerized system so that the complexity in the working of manual system can be reduced and the time can be saved. It also leads to the minimization of the errors prone to the manual system.

The present system is totally manual system. Every type of data processing/recording/manipulation is done manually. So the present system is not accurate and very time consuming. All type of work is done on papers/files. In the present system record of item detail, supplier detail ,billing detail ,purchase detail is maintained on the files. So any type of manipulation is very difficult to be done in the existing system. And maintenance of present system is very costly. End user interface is not good. Lack of communication between the different departments is there. It does not provide integrity between different departments. The present system needs a lot of man power. The record (in the form of files) is not kept for long duration because it becomes very bulky in size and is very difficult to maintain and handled it.

Ashish .. Ankur

Signature of Student
Name ASHISH ANKUR
Date 24th May 2011.

Signature of Supervisor
Name

CHAPTER -1

INTRODUCTION

DEPARTMENTAL RESOURCE MANAGEMENT

1.1 What is departmental resource management?

Departmental resource management is the active control program which helps a particular department to make good use of their available resources.

Departmental resource management software helps a particular department to know about the available resources(like in our project if any student wants to use a computer which has redhat(linux) operating system then the student will get the information from our project).

The present system is totally manual system. Every type of data processing/recording/manipulation is done manually. So the present system is not accurate and very time consuming. All type of work is done on papers/files. In the present system record of item detail, supplier detail ,billing detail ,purchase detail is maintained on the files. So any type of manipulation is very difficult to be done in the existing system. And maintenance of present system is very costly. End user interface is not good. Lack of communication between the different departments is there. It does not provide integrity between different departments. The present system needs a lot of man power. The record (in the form of files) is not kept for long duration because it becomes very bulky in size and is very difficult to maintain and handled it.

1.2 Purpose of project

□ **Data Redundancy and Inconsistency:**

In Manual System, as all the data is entered on paper by hand so there can be some human errors and also some data can be missed which can lead to inconsistent data. Also there can be duplicate copies of same data.

□ **Data Isolation:**

Since data is scattered in different files that may be in different format, it is difficult to write new application program to retrieve data.

□ **Security Problem:**

Anybody can access system or data files and can change data resulting into inconsistent and incorrect data, so there is no security provision.

□ **Updating Problem:**

There is a different way to update records and files in the manual system. If there is any updating of data at one place, the person who is maintaining records has to go through all the records corresponding to that updating records and maintain the consistent data.

□ **Time Consuming:**

A lot of time is required to access or retrieve data and to generate reports.

1.2.1 OBJECTIVE OF PROJECT

The overall motive of this project is to convert the existing manual system into a computerized system so that the complexity in the working of manual system can be reduced and the time can be saved. It also leads to the minimization of the errors prone to the manual system. In this respect, the project aims at accomplished of the following:

To design a system in such a way that even a non-technical person can use it.

To make software completely user friendly.

To prevent the physical loss of data which is possible in the manual System.

To reduce working hours.

To provide up-to-date information.

It should be able to store and retrieve data efficiently.

It should be flexible enough so that there is ample scope for modification in the future.

This project is developed to automate the existing manual system into a computerized system so that the complexity in the working of manual system can be reduced and the time can be saved.

1.2.2 OTHER OBJECTIVES

To identify and track all data processing assets in an Department resource Repository.

To define the process by which assets are identified and maintained in the department resource.

To provide department resource access to all necessary personnel (data entry, update and deletion).

To provide a full range of reports that will satisfy informational requirements.

To document the department Management System within the Standards and Procedures Manual.

To provide training to personnel responsible for supporting the department Management System.

Up-to-date information about data processing resources through the creation and archiving of records in a centralized repository.

Financial records specific to a single component, or groups of components.

Component Status Indicators to identify a component as Active (A), Redeployed (R), Donated (D), or Terminated (T).

Service records for all components in the department.

OBJECTIVE OF PROPOSED SYSTEM

- **Database:**

  Create a database to store all the records and data or to reduce the storage space for maintaining and storing data.

- **Less Time:**

  Through computerization wastage of time would be avoided as less time is required to access or retrieve data and to generate reports.

- **Less Error:**

  It would provide error free maintenance of data as the existing manual system was more prone to errors.

- **Networking:**

  Through the networking of computers there would be easy communication between the various sub systems or the departments of the company.

- **Updating:**

  Computers provide the facility for fast and easy updating of records and data.

- **Increase Efficiency:**

  Being online it provide the easy way to perform task and the efficiency of the various departments will increase.

- **Friendly Interface:**

  By the use of front end software the user of the system will be provided with a very friendly and easy interface so that the user knows the system better.

- **Reports Generation:**

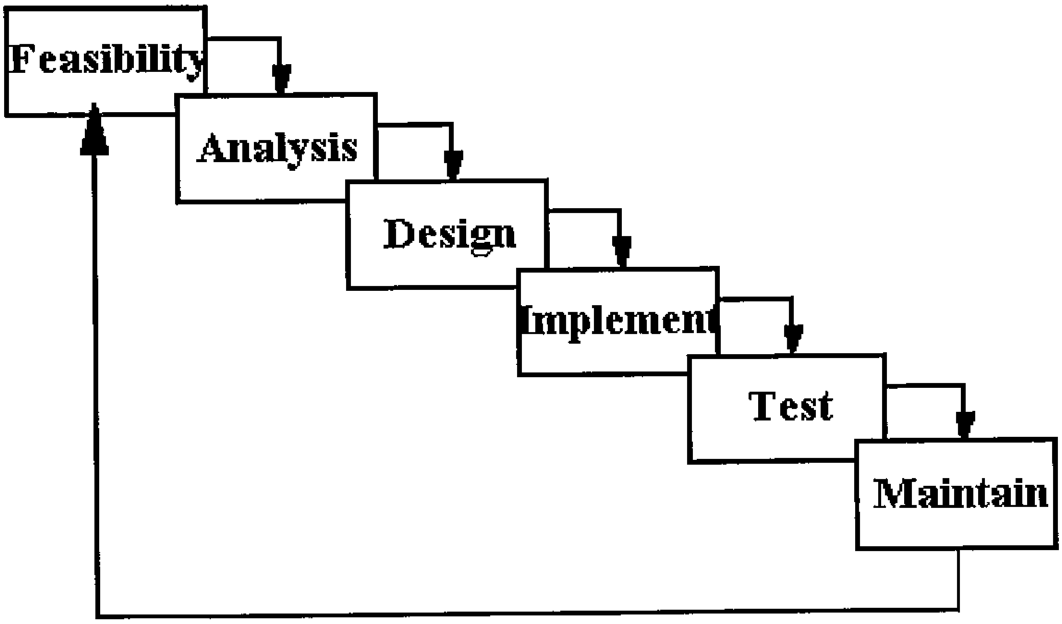
  The system will be provided the facility for generating printed reports, which were previously written by hand.

- **Security:**

  Security must be provided so that the unauthorized person can not access the system.

1.3 SOFTWARE DEVELOPMENT LIFE CYCLE

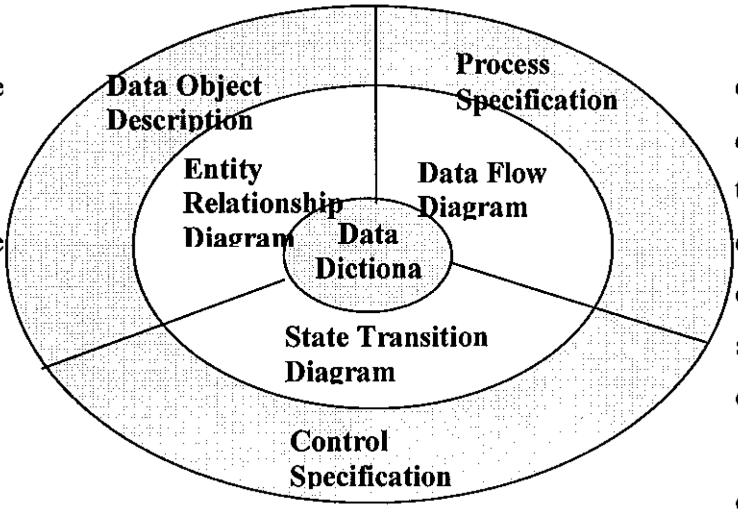
The Systems Development Life Cycle (SDLC) is a conceptual model used in project management that describes the stages involved in an information system development project from an initial feasibility study through maintenance of the completed application. Various SDLC methodologies have been developed to guide the processes involved including the waterfall model (the original SDLC method). Documentation is crucial regardless of the type of model chosen or devised for any application, and is usually done in parallel with the development process. Some methods work better for specific types of projects, but in the final analysis, the most important factor for the success of a project may be how closely particular plan was followed.



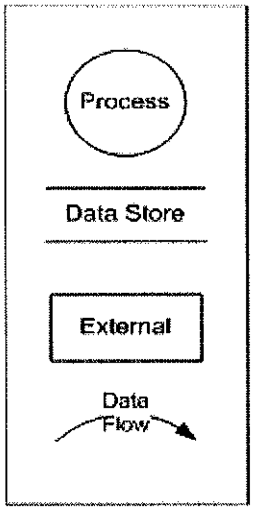
The Analysis model :

The analysis model must achieve three primary objectives:

- To describe what the user requires.
- To establish the basis for the enhancement of a software design.
- To define a set of requirements that can be validated once the software is completely enhanced. The main elements of the analysis model are briefly described below.



- At the core of the model lies the *data* *dictionary*, which is a repository that contains descriptions of all the data objects consumed or produced by the software .Three different diagrams surround the core.
- The **entity relation diagram** depicts relationships between data objects.



A **data flow diagram (DFD)** is a graphical representation of the "flow" of data through an information system. A data flow diagram can also be used for the visualization of data processing (structured design). It is common practice for a designer to draw a context-level DFD first which shows the interaction between the system and outside entities. This context-level DFD is then "exploded" to show more detail of the system being modeled.

DATA FLOW DIAGRAM NOTATION

External Entities/Terminators are outside of the system being modeled. Terminators (also referred to as sources or sinks, depending on whether data flows from or into them) represent where information comes from or where it goes. In designing a system, we have no idea about what these terminators do or how they do it.

Processes modify the inputs in the process of generating the outputs

Data Stores represent a place in the process where data comes to rest. A DFD does not say anything about the relative timing of the processes, so a data store might be a place to accumulate data over a year for the annual accounting process.

Data Flows are how data moves between terminators, processes, and data stores (those that cross the system boundary are known as IO or Input Output Descriptions).

Every page in a DFD should contain fewer than 10 components. If a process has more than 10 components, then one or more components (typically a process) should be combined into one and another DFD be generated that describes that component in more detail. Each component should be numbered, as should each subcomponent, and so on. So for example, a top level DFD would have components 1 2 3 4 5, the subcomponent DFD of component 3 would have components 3.1, 3.2, 3.3, and 3.4; and the subsubcomponent DFD of component 3.2 would have components 3.2.1, 3.2.2, and 3.2.3. DFD can used to present the project.

CHAPTER-2
FEASIBILITY

2.1 FEASIBILITY STUDY

The feasibility study is used to determine if the project should get the go-ahead. If the project is to proceed, the feasibility study will produce a project plan and budget estimates for the future stages of development. Feasibility is the determination of whether or not a project is worth doing the process followed in making this determination is called a feasibility study. It is an analysis of possible alternative solutions to a_problem and a recommendation on the best alternative.Feasibility study is carried out to select the best system that meets system performance requirements.

Different types of feasibility study:

- Technical feasibility
- Operational feasibility
- Economic feasibility

1. TECHNICAL FEASIBILITY:

This is concerned with the specifying equipment and software that will successfully satisfy the requirements. The proposed system is technically feasible as it can be developed easily with the help of available technology. The proposed system requires SQL which is used as back-end and Core Java with Swings and JDBC as front end.

In the technical needs of the system these points are considered.

- The facility to produce in given time.
- Response time under conditions.

2. ECONOMICAL FEASIBILITY:

Economic analysis is the most frequently used technique for evaluating the effectiveness of the proposed system. The producer is to determine the benefits and the saving that are expected from a proposed system and compare them with the proposed system.

The only tangible benefits proposed that the manual work and burden is reduced maximum as possible, resulting the reduction in manpower requirement and cost incurred . The system provides many benefits that can't be measured in terms of money for e.g. user friendliness, more efficient user response, maintenance of database etc.

3. OPERATIONAL FEASIBILITY:

The proposed system is highly user friendly and it is much easily to interact with the user. Therefore, the user will easily accept the system as data entry system and queries can be easily solved. Initial stages of the system might face some resistance but once complete automation is achieved and operators are trained. The system will provide maximum easiness.

2.2 FRONT-END/BACK-END

Front-End: **Core Java using Swings and JDB**

Back-End: **Microsoft Access**

2.2.1 Java

Java is a blend of the best elements of its rich heritage languages combined with the innovative concepts required by its unique mission. Although Java has become inseparably linked with the online environment of the Internet, it is important to remember that Java is first and foremost a programming language. Much of the character of Java is inherited from C and C++. From C, Java derives its syntax. Many of Java's object oriented features were influenced by C++.

James Gosling, Patrick Naughton, Chris Warth, Ed Frank and Mike Sheridan developed Java at Sun Microsystems, Inc. in 1991. This Language was initially called "Oak" but was renamed "Java" in 1995. The original impetus for Java was the need for a platform independent language that could be used to create software to be embedded in various consumer electronics devices such as remote controls. But, with the emergence of the World Wide Web, Java was propelled to the forefront of computer language design, because the web too, demanded portable programs.

The Java Buzzwords

The key feature responsible for the development of Java are

Simple

Java was design to be easy for the professional programmers to learn and use efficiently. If one already understand the basic concepts of OOP, learning Java will be ever easier.

Robust

The multiplatformed environment of the Web places extraordinary demands on a program, because the program must execute reliably in a verity of systems.

Multithreaded

Java was design to meet the real world requirements of creating interactive, networked programs. To accomplish this, Java supports multithreaded programming, which allow us to write programs that do many things simultaneously.

Distributed

Java is design for the distributed environment of the Internet, because it handles the TCP/IP protocols.

The brief discussion of APIs

AWT

It stands for Abstract Window Toolkit. Since all applets run in a window, it is necessary to include AWT, which supports window-based applet programs. Applets are event-driven. An applet waits until an event occurs. The AWT notifies the applet about an event by calling an event handler that had been provided by the applet. Once this happens, the applet must take appropriate actions and then quickly return control to the AWT. The architecture of an applet is not as easy to understand as that of a console-based program, Java's AWT makes it as simple as possible.

awt.event applets are event-driven programs. Thus, event handle is at the core of successful applet programming. Most events to which user's applet will respond are generated by the user. These events are passed to user's applet in variety of ways, with the specific method depending upon the actual event. There are several types of events. The most commonly handled events are those generated by the mouse, the keyboard, and various controls, such as push button. Events are supported by the **java.awt.event** package.

Exception Handling-

An **exception** is an abnormal condition that arises in a code sequence at run time. In other words, an exception is a run-time error. In computer languages that do not support exception handling, errors must be checked and handled manually—typically through the use of error codes, and so on. This approach is as cumbersome as it is troublesome. Java's exception handling avoids these problems and, in the process, brings run-time error management into the object-oriented world.

A Java exception is an object that describes an exceptional condition that has occurred in a piece of code. When an exceptional condition arises, an object representing that exception is created and **thrown** in the method that caused the error. That method may choose to handle the exception itself, or pass it on. Either way, at some point, the exception is **caught** and processed. Exceptions can be generated by the Java run-time system, or they can be manually generated by the user's code. Exceptions thrown by Java relate to fundamental errors that violate the rules of the Java language or the constraints of the Java execution environment. Manually generated exceptions are typically used to report some error condition to the caller of a method.

Swing

Swing is a set of classes that provides more powerful and flexible components than are possible with the AWT. In addition to the familiar components, such as buttons, check boxes, and labels, Swing supplies several exciting additions, including tabbed panes, scroll panes, trees, and tables. Even familiar components such as buttons have more capabilities in Swing. For example, a button may have both an image and a text string associated with it. Also the image can be changed as the state of the button changes. Unlike AWT components, Swing components are not implemented by platform-specific code. Instead, they are written entirely in Java and, therefore, are platform independent. The Swing-related classes are contained in **javax.swing**. Some of the classes used are JApplet, JLabel, JButton, JScrollPane, JTextArea, JTextField, JMenuBar, JMenu, JFrame. JApplet and JFrame provides a method getContentPane() which returns a reference to their associated content pane. So to get a reference to the JApplet's container where user will add their GUI objects.

```
        Container appletContainer = getContentPane();
```

This statement can be in the instance variable initialization list, where user declare and instantiate the GUI objects. Then, subsequent adds or sets of layout managers in the constructor or in init() would be on this appletContainer. For example,

```
        AppletContainer.add(nameLabel);
```

JApplet

Fundamental to Swing is the JApplet class, which extends Applet. Applets that use Swing must be subclasses of **JApplet**. JApplet is rich with functionality that is not found in Applet.

JLabel Swing labels are instances of **JLabel** class, which extends

JComponent. It can display text and / or an icon. This class is used for putting a label on an applet, like a label used is "Enter your name".

JTextField

It allows user to edit one line of the text. Like a text field of 20 characters long is created with the label "Enter your name" to enter the name of the user.

JButton

JButton class provides the functionality of a push button. JButton allows an icon, a string, or both to be associated with push button.

JTextArea

It allows user to edit in an area specified in terms of rows and columns.

JScrollPane

Scroll panes are implemented in Swing by the JScrollPane, which extends JComponent. A scroll pane is a component that presents a rectangular area in which a component may be viewed. Horizontal and / or vertical scroll bars may be provided if necessary.

JFrame

The JFrame implements frames in Swings. Frame encapsulates a "window". It is the subclass of a window and has a titlebar, menubar, borders and resizing corners.

JMenuBar and JMenu A menu bar displays a list of top-level menu choices. JMenuBar and JMenu implement it.

CHAPTER-3

3.1 SYSTEM DESIGN

The design phase involves converting the informational, functional, and network requirements identified during the initiation and planning phases into unified design specifications that developers use to script programs during the development phase. Program designs are constructed in various ways. Using a top-down approach, designers first identify and link major program components and interfaces, then expand design layouts as they identify and link smaller subsystems and connections. Using a bottom-up approach, designers first identify and link minor program components and interfaces, then expand design layouts as they identify and link larger systems and connections. Contemporary design techniques often use prototyping tools that build mock-up designs of items such as application screens, database layouts, and system architectures. End users, designers, developers, database managers, and network administrators should review and refine the prototyped designs in an iterative process until they agree on an acceptable design.

Designers should carefully document completed designs. Detailed documentation enhances a programmer's ability to develop programs and modify them after they are placed in production. The documentation also helps management ensure final programs are consistent with original goals and specifications. Organizations should create initial testing, conversion, implementation, and training plans during the design phase. Additionally, they should draft user, operator, and maintenance manuals.

The main objective of the system design is to make the system user friendly. System design involves various stages as:

- Data Entry
- Data Correction
- Data Deletion
- Processing
- Sorting and Indexing
- Report Generation

System design is the creative act of invention, developing new inputs, a database, offline files, procedures and output for processing business to meet an organization objective. System design builds information gathered during the system analysis.

The designing phase of any software development is carried out in the following stages:

- Architectural design(high level design)
- Detailed design(low level design or physical design)

HIGH LEVEL DESIGN:

The top level design aims to identify the modules that should be in the system, the specification of these modules, and how they interact with each other to priduce the desired results, at the end of the system design all the major data structures, file formats, output formats, and the major modules in the system and their specifications are decided.

It involves:

Identifying the entities

All the entities related to the model were identified, checked and consolidated.

Identifying the relationships

The relationships between the entities within and outside the system were identified.

LOW LEVEL DESIGN:

During detailed design, the internal logic of each of the modules specified design is decided. During the phase further details of the data structure and algorithmic design of each of the modules is specified. The logic of the module is usually specified in a high level design description language, which is independent of the target language in which the software will eventually be implemented.

In the system design the focus is on identifying the modules, where as during detailed design focus is on designing the logic for each of the modules. The three main tools of system design are:

- Data flow diagram
- Flow charts
- Data dictionary.

3.2 TESTING

The testing phase requires organizations to complete various tests to ensure the accuracy of programmed code, the inclusion of expected functionality, and the interoperability of applications and other network components. Thorough testing is critical to ensuring systems meet organizational and end-user requirements. Test plans created during initial project phases enhance an organization's ability to create detailed tests.

A bottom-up approach tests smaller components first and progressively adds and tests additional components and systems. A top-down approach first tests major components and connections and progressively tests smaller components and connections.

Bottom-up tests often begin with functional (requirements based) testing. Functional tests should ensure that expected functional, security, and internal control features are present and operating properly. Testers then complete integration and end-to-end testing to ensure application and system components interact properly. Users then conduct acceptance tests to ensure systems meet defined acceptance criteria. Organizations should review and complete user, operator, and maintenance manuals during the testing phase. Additionally, they should finalize conversion, implementation, and training plans.

Testing the system

- Testing can be done with two types of data. Live data and test data.
- Live data is the data actually to be used in the proposed system.
- Test data is previously designed sample input to achieve predictable results.

Testing Objective

- Testing is a process of execution a program with the intent of finding an error.
- A good test case is one that has a high portability of finding an undiscovered error.
- A successful test is one that uncovers an as-yet-discovered error.

Testing Principles

- ATI tests should be traceable to customer requirements.
- Test should be planned long before testing phase.
- Testing should begin “in small “ and progress toward testing “in the large”.
- Exhaustive testing is completely possible.
- To be most effective, an independent third party should conduct testing.

Testing Methods

WHITE BOX TESTING:

White box testing of software is predicated on close examination of procedural detail. Providing test cases that exercise specific sets of conditions and or loops tests logical paths through the software. White Box Testing, sometime called glass box testing, is a test case design method that uses the control structure of the procedural design to derive test cases.

Using white box testing methods, following test cases can be derived.

- Guarantee that all independent paths within module have been exercised at least once.
- Exercise all logical decisions on their true and false sides. Execute all loops at their boundaries and within their operational bounds.
- Exercise internal data structures to assure their validity.
- The errors that can be encountered while conducting white box testing are-
    - Logical errors and incorrect assumptions.
    - Typographical errors

BLACK BOX TESTING:

Black box testing is carried out to check the functionality of the various modules. Although they are designed to uncover errors, black box tests are used to demonstrate that software functions are optional; that input is properly accepted and output is correctly produced, and the integrity of external information is maintained, a black box test examines some fundamental aspects of the system with little regard for the internal logical structure of the software. Black box testing focuses on the functional requirements of the software.

Black box testing attempts to find errors in the following categories:

- Incorrect or missing functions.
- Interface errors.
- Errors In data structure or external database access.
- Performance errors
- Initialization and termination errors.

Unlike white box testing, which is performed early in the testing process, black box tends to apply during later stages of testing. Because black box testing purposely disregards control structure, attention is focused on the information domain.

SOFTWARE TESTING STRATEGIES:

A strategies for software testing integrates software test case design methods into a well planned series of steps that result in the successful construction of software. An important, software testing strategy provides a road map. Testing is set of activities that can be planned in advanced and conducted systematically. Various strategies are given below:

UNIT TESTING: Unit testing focuses verification effort on the smallest unit of software design i.e module. Using procedural design as a guide, important control paths are tested to uncover errors within the boundary of the module.

INTEGEARTION TESTING: Integration testing is systematic technique for constructing the program structure while conducting tests to uncover errors associated with interfacing.

ACCEPTANCE TESTING: To ensure that the final system, as it will be delivered, complies with all the client's requirements of the system, as detailed in the SRS. If monitor alterations are required, then some form of regression testing may also have to be used.

VALIDATION TESTING: At the culmination of Integration testing, software is completely assembled as a package, Interfacing errors have been uncovered and corrected and a final series of software test-validation testing may begin.

SYSTEM TESTING: Software is incorporated with other system elements and a series of system integration and validation test are conducted. The various types of system testing are:

- **Recovery Testing:** many computer based systems must recover from faults and resume processing within a pre-specified time.
- **Security Testing:** Security testing attempts to verify that protection mechanisms built into a system will in fact protect it from improper penetration.
- **Stress Testing:** Stress tests are designed to confront programs with abnormal situations.
- **Performance Testing:** Performance testing is designed to test run time performance of software within the context to an integrated system.

All above mentioned testing principles have been applied to all the modules and the modules have passed the tests successfully.

TESTING OF FORMS

Forms are the interface between the user and database system. Testing of forms was extensive task. All data entry and query formats are designed using forms. Forms were tested to ensure that they are performing the tasks and well they are designed fork and correction and modifications were made found to be necessary.

System testing is designed to uncover weakness that were not found in the earlier tests. This includes forced system failure and its users in the operational environment will implement validation of the total system as it. The total system is also tested for recovery and fallback after various major failures to ensures that data are lost during the emergency. All this is done with the old system still in operation.

After a successful testing of the individual programs and forms, the whole system was through a series of test to ensure the proper working of the system as a whole. The system as a whole unit put to all-possible inputs. The consistency and validity is then checked by the outputs. The activities involved in the system testing are:

- Integration Testing
- Acceptance Testing

In integration testing the entire system is tested and Acceptance Testing involves planning and execution of functional tests, performance tests and stress testing order to demonstrate that the implemented system satisfied its requirements. The functional tests, performance tests and stress tests are performed to determine the limitation of the system.

FUNCTIONAL TESTS:

The system went through functional tests all along the development stage. Whenever a new function was added it went through a thorough testing for all possible input values and its interoperability with other functions by the coder. On delivery of the intermediate system to the customer, the customer studied the system functionalities and provided the feedback, which often mentioned the changed desired. So the system, went through functional test both at the developer and customer ends.

PERFORMANCE TESTS:

Before the delivery of the system to the customers the new system went through various tests to benchmark its performance.

- These included the GUI testing.
- The functional testing.
- And, the overall system integrity testing.

STRESS TESTS:

The system was pushed to its limits and beyond during the stress testing, to measure its performance and stability versus the load.

SECURITY TESTS:

The functional environment of the system posed no real security threst but the system was developed considering the entire data to be highly critical and thus denying free access to anyone without proper access rights. The entire system was password protected while the passwords themselves were protected through foolproof encryption algorithms.

3.3 SYSTEM IMPLEMENTATION

Implementation is the stage in the project where the theoretical design is turned into the working system and is giving confidence to the new system for the users i.e. will work efficiently and effectively. It involves careful planning, investigation of the current system and its constraints on implementation, design of method to achieve the change over, an evaluation, of change over methods. A part from planning major task of preparing the implementation is education of users. The more complex system is implemented, the more involved will be the system analysis and design effort required just for implementation. An implementation coordinating committee based on policies of individual organization has been appointed. The implementation process begins with preparing a plan for the implementation for the system. According to this plan, the activities are to be carried out, discussions may regarding the equipment has to be acquired to implement the new system.

Implementation is the final and important phase. The most critical stage is in achieving a successful new system and in giving the users confidence that the new system will work and be effective. The system can be implemented only after thorough testing is done and if it found to working according to the specification. This method also offers the greatest security since the old system can take over if the errors are found or inability to handle certain types of transaction while using the new system.

At the beginning of the development phase a preliminary implementation plan is created to schedule and manage the many different activities that must be integrated into plan. The implementation plan is updated throughout the development phase, culminating in a change over plan for the operation phase. The major elements of implementation plan are test plan, training plan, equipment installation plan, and a conversion plan.

There are three types of implementation:

- Implementation of a computer system to replace a manual system. (This is the type of implementation we are using in our project)
- Implementation of a new computer system to replace an existing system.
- Implementation of a modified application to replace an existing one, using the same computer.

4.CODING

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
class global
{
Connection con;
ResultSet rs;
Statement stmt;
public Connection getconnection()
{
return(con);
}
global()
{
 try {
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
con = DriverManager.getConnection("jdbc:odbc:plan");
stmt = con.createStatement();
}catch(Exception e)
 {
  System.out.println(e.getMessage());
 }
}
ResultSet  execute(String s) throws Exception
{
rs = stmt.executeQuery(s);
return(rs);
}
void update(String s) throws Exception
{
```

```
stmt.executeUpdate(s);
}
}
class mytest
{
public static void main(String s[ ])
{
myframe f1 = new myframe("   USER AUTHENTICATION");
f1.setVisible(true);
int i;
int x=1;
for(i=2;i<=300;i+=4,x+=6)
{
  f1.setLocation((400-((i+x)/2) ),300-(i/2));
  f1.setSize(i+x,i);
}
}
}
class myframe extends JFrame implements ActionListener
{
JPasswordField t1;
JTextField t2;
JButton b1,b2;
myframe(String s)
{
super(s);
Container con=getContentPane();
con.setLayout(new FlowLayout());
t1 = new JPasswordField(10);
t2 = new JTextField(10);
b1= new JButton(" OK ");
b2= new JButton(" CANCEL");
ImageIcon c1 = new ImageIcon("login.png");
```

```
JLabel m1 = new JLabel(c1);
con.add(m1);
con.add(new JLabel("User Name  "));
con.add(t2);
con.add(new JLabel("Password "));
con.add(t1);
con.add(b1);
con.add(b2);
b1.addActionListener(this);
b2.addActionListener(this);
}
public void actionPerformed(ActionEvent ae)
 {
 if(ae.getSource() == b1)
 {
   global g1 = new global();
   ResultSet rs=null;
try {
String ss="select * from login where username='"+ t2.getText()+"'    and password  ='"+t1.getText()+"'
";
System.out.println(ss);
rs = g1.execute(ss);
rs.next();
int n= rs.getRow();
if(n>0)
 {
 master b1 = new master();
 b1.setSize(1400,800);
 b1.setVisible(true);
 this.setVisible(false);
  }
 else
 JOptionPane.showMessageDialog(null,  "I N V A L I D username or password");
}catch(Exception e)
{ t2.setText(e.getMessage());}
}
}
}




class master extends JFrame implements ActionListener
{
 master()
  {
   super("Sitting Arrangement System");
   JMenuBar mb = new JMenuBar();
   setJMenuBar(mb);
   JMenu m1 = new JMenu("Masters");
   mb.add(m1);
  JMenuItem t1 = new JMenuItem("New Users");

   JMenuItem t4 = new JMenuItem("List Users");

  JMenuItem t5 = new JMenuItem("Computers");
  JMenuItem t6 = new JMenuItem("List Computers");

  m1.add(t1);

   m1.add(t4);
   m1.add(t5);
  m1.add(t6);
```

```
   t1.addActionListener(this);
    t4.addActionListener(this);
   t5.addActionListener(this);
    t6.addActionListener(this);

ImageIcon c1 = new ImageIcon("real.jpg");
JLabel mm2 = new JLabel(c1);
 getContentPane().setLayout(new BorderLayout());
getContentPane().add(mm2,BorderLayout.CENTER);
    }
 public void actionPerformed(ActionEvent ae)
  {
   String msg = ae.getActionCommand();


   if(msg.equals("New Users"))
     {
        usernew b1 = new usernew();
        b1.setSize(250,250);
        b1.setVisible(true);
       b1.setLocation(200,200);
     }

 if(msg.equals("List Users"))
     {

    userlist b1 = new userlist();
        b1.setSize(450,150);
        b1.setVisible(true);
       b1.setLocation(200,200);


      }
```

```
if(msg.equals("Computers"))
   {

    com b1 = new com("Add Computer Information");
        b1.setSize(650,250);
        b1.setVisible(true);
       b1.setLocation(200,300);

    }
if(msg.equals("List Computers"))
   {

    lablist b1 = new lablist();
        b1.setSize(650,250);
        b1.setVisible(true);
       b1.setLocation(200,300);

     }

}
}
class usernew extends JFrame implements ActionListener
{
JTextField t1,t2,t3,t4;
JLabel l1,l2,l3;
JButton b1,b2;
usernew()
{
super(" New User Account ");
t1 = new JTextField(20);
t2 = new JTextField(20);
t3 = new JTextField(20);
t4=new JTextField(20);
```

```
l1 = new JLabel(" User Name");
l2 = new JLabel(" Password");
l3 = new JLabel(" Type");
b1 = new JButton("Add");
b2= new JButton("Save");
getContentPane().setLayout(new FlowLayout());
getContentPane().add(l1);
getContentPane().add(t1);
getContentPane().add(l2);
getContentPane().add(t2);
getContentPane().add(l3);
getContentPane().add(t3);
getContentPane().add(new JLabel("Email Id"));
getContentPane().add(t4);

getContentPane().add(b1);
getContentPane().add(b2);
b1.addActionListener(this);
b2.addActionListener(this);
}
public void actionPerformed(ActionEvent ae)
{
 if(ae.getSource() == b1)
 {
  t1.setText("");
  t2.setText("");
 t3.setText("");
}
else
{
try
  {
    global g1 = new global();
    String qry = "insert into login values('"+t1.getText()+"',' "+ t2.getText()+" ' , '"+ t3.getText()
+"','"+t4.getText()+"') ";
   g1.update(qry);
   JOptionPane.showMessageDialog(null," Record is saved");
}catch(Exception e) { t1.setText(""+e);}


}
}
}

class userlist extends JFrame
{
userlist()
{
try
{
global g1 = new global();
ResultSet rs = g1.execute("select * from login");
String c [ ] = { "username","pasword ","Type"};
String[ ][ ] data = new String[30][3];
 int i,j;
 i=0;
 j=0;
  while(rs.next())
 {
  data[i][ j++]=rs.getString("username");
 data[i][ j++]=rs.getString("password");
 data[i][ j]=rs.getString("type");
   j=0;   i++;
  }
 JTable jtab = new JTable(data,c);
 getContentPane().setLayout(new BorderLayout());
 int v = ScrollPaneConstants.VERTICAL_SCROLLBAR_AS_NEEDED;
int h = ScrollPaneConstants.HORIZONTAL_SCROLLBAR_AS_NEEDED;
JScrollPane jsp = new JScrollPane(jtab, v, h);

getContentPane().add(jsp,BorderLayout.CENTER);
 }
 catch(Exception e) {System.out.println(e); }
}
}



class lablist extends JFrame
{
lablist()
{
try
{
global g1 = new global();
ResultSet rs = g1.execute("select * from computer");
String c [ ] = { "Lab Name","No Of Computers "," Processor","Hard disk","Ram" ,"Operating
System"};
String[ ][ ] data = new String[30][6];
 int i,j;
 i=0;
 j=0;
  while(rs.next())
 {
  data[i][ j++]=rs.getString("labname");
 data[i][ j++]=rs.getString("noofcomputer");
 data[i][ j++]=rs.getString("processor");
 data[i][ j++]=rs.getString("harddisk");

 data[i][ j++]=rs.getString("ram");
data[i][ j++]=rs.getString("os");

 j=0;  i++;
 }
JTable jtab = new JTable(data,c);
getContentPane().setLayout(new BorderLayout());
int v = ScrollPaneConstants.VERTICAL_SCROLLBAR_AS_NEEDED;
int h = ScrollPaneConstants.HORIZONTAL_SCROLLBAR_AS_NEEDED;
JScrollPane jsp = new JScrollPane(jtab, v, h);

getContentPane().add(jsp,BorderLayout.CENTER);
 }
 catch(Exception e) {System.out.println(e); }
}
}


  class com extends JFrame implements ActionListener
   {
    JLabel l1,l2,l3,l4,l5,l6;
    JTextField t1,t2,t3,t4,t5,t6;
    JPanel p1,p2,p3,p4,pp;
    JButton bf,bl,bn,bp;

    Connection con1;
    Statement stmt;
    ResultSet rs;
    String str;

    com(String s1)
     {
      super(s1);
      str="";
```

```
Container con=getContentPane();
try
 {
  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
  con1 = DriverManager.getConnection("jdbc:odbc:plan");
  stmt = con1.createStatement();
  rs= stmt.executeQuery("select * from computer");
 }
catch(Exception e) { str=e.getMessage();}

p1 = new JPanel();
p2 = new JPanel();
p3 = new JPanel();
p4 = new JPanel();
pp = new JPanel();

l1=new JLabel("Lab Name");
l2=new JLabel("No Of Computers");
l3=new JLabel("Processor ");
l4=new JLabel("Hard Disk");
l5=new JLabel("RAM");
l6 = new JLabel("Operating System");

t1=new JTextField(10);
t2=new JTextField(10);
t3=new JTextField(10);
t4=new JTextField(10);
t5=new JTextField(10);
t6 = new JTextField(10);

p1.add(l1);
p1.add(t1);
p1.add(l2);
```

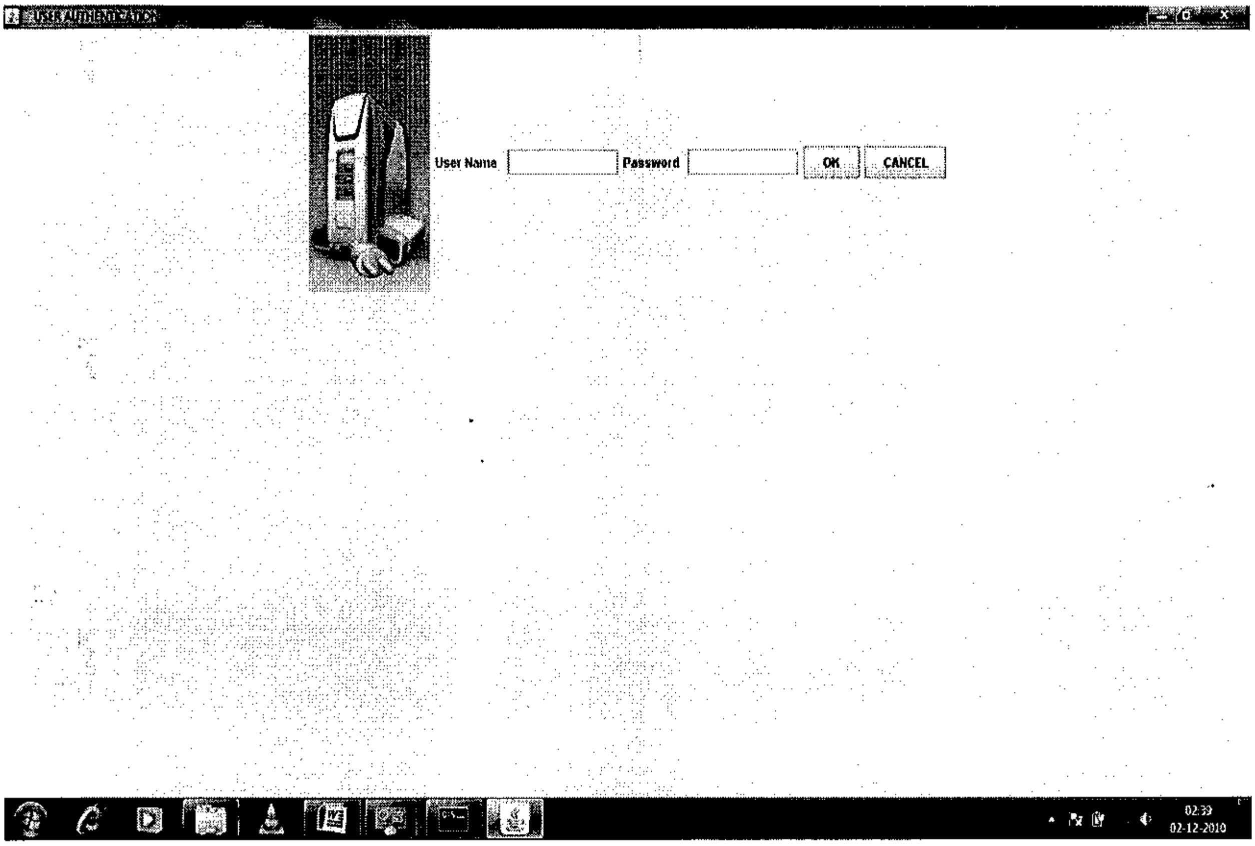
```
    p1.add(t2);

    p2.add(l3);
    p2.add(t3);
    p2.add(l4);
    p2.add(t4);

    p3.add(l5);
    p3.add(t5);
    p3.add(l6);
    p3.add(t6);

t1.setEnabled(false);
t2.setEnabled(false);
t3.setEnabled(false);
t4.setEnabled(false);
t5.setEnabled(false);
t6.setEnabled(false);
    con.setLayout(new BorderLayout());

    bf= new JButton("Add Record");
    bl= new JButton("Update Record");
    bn = new JButton("Move Next");
    bp = new JButton("Delete Record");

    p4.add(bf);
    p4.add(bl);
    p4.add(bn);
    p4.add(bp);

    bl.setEnabled(false);

    pp.setLayout(new BorderLayout());
    pp.add(p2,BorderLayout.NORTH);
    pp.add(p3,BorderLayout.CENTER);

    con.add(p1,BorderLayout.NORTH);
    con.add(pp,BorderLayout.CENTER);
    con.add(p4,BorderLayout.SOUTH);

    bf.addActionListener(this);
    bl.addActionListener(this);
    bn.addActionListener(this);
    bp.addActionListener(this);

    t1.setText(str);
   }

    public void actionPerformed(ActionEvent ae)
      {
       String s1,s2="";
       String a,b,c,d,e,f,g;
       s1=ae.getActionCommand();

       try
        {
         if(s1.equals("Add Record"))
          {
t1.setEnabled(true);
t2.setEnabled(true);
t3.setEnabled(true);
t4.setEnabled(true);
t5.setEnabled(true);
t6.setEnabled(true);

           t1.setText("");
```

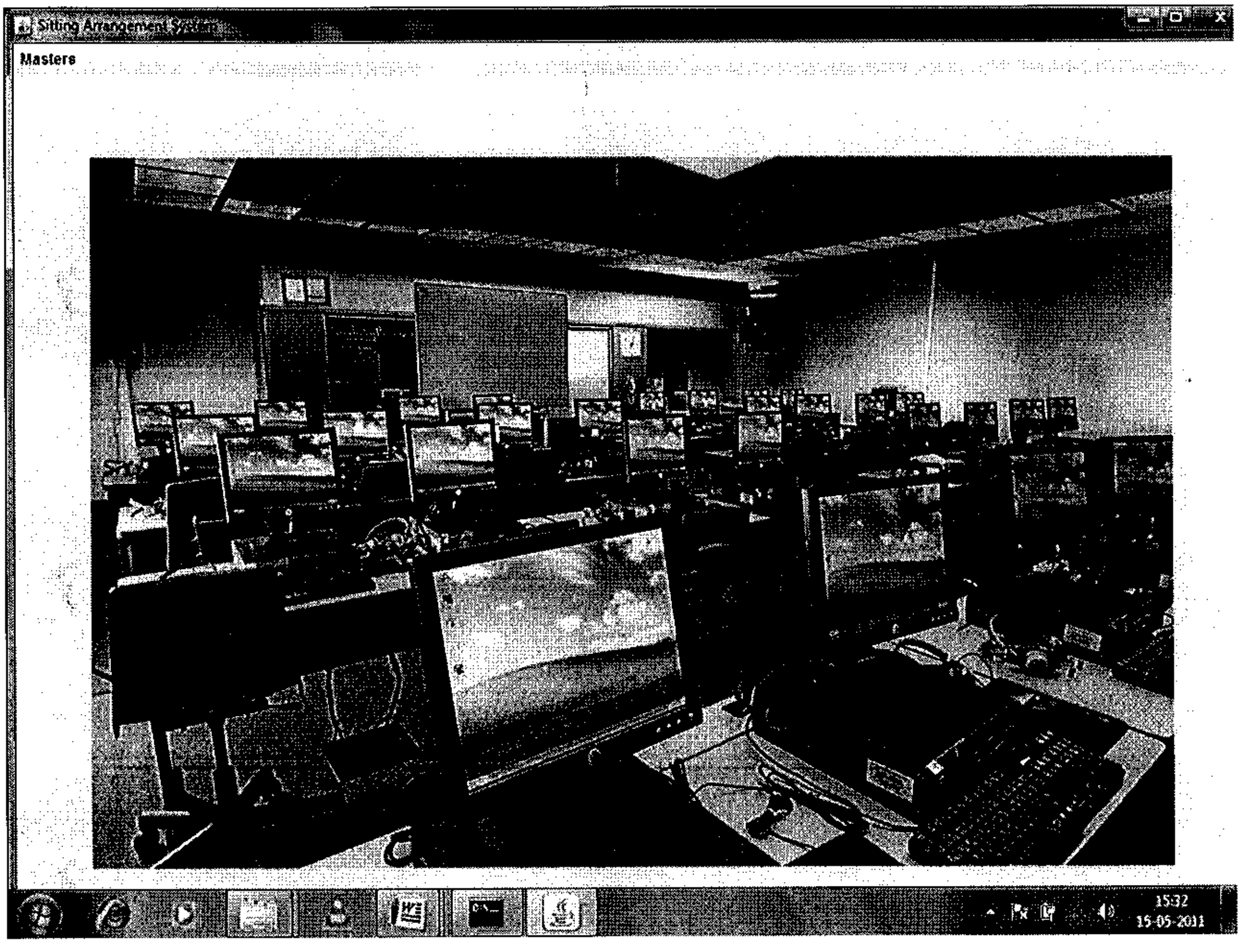
```
            t2.setText("");
            t3.setText("");
            t4.setText("");
            t5.setText("");
            t6.setText("");


            bl.setEnabled(true);
            bf.setEnabled(false);
          return;
            }
          if(s1.equals("Update Record"))
            {
            a = t1.getText();
            b = t2.getText();
            c = t3.getText();
            d = t4.getText();
            e = t5.getText();
            f = t6.getText();

 s2="insert into computer values('"+a+"','"+b+"','"+c+ "','"+d+ "','"+e+ "','"+f+ "')" ;

            stmt.executeUpdate(s2);
           t1.setText("");
            t2.setText("");
            t3.setText("");
            t4.setText("");
            t5.setText("");
            t6.setText("");
        t1.setEnabled(false);
t2.setEnabled(false);
t3.setEnabled(false);
t4.setEnabled(false);
```

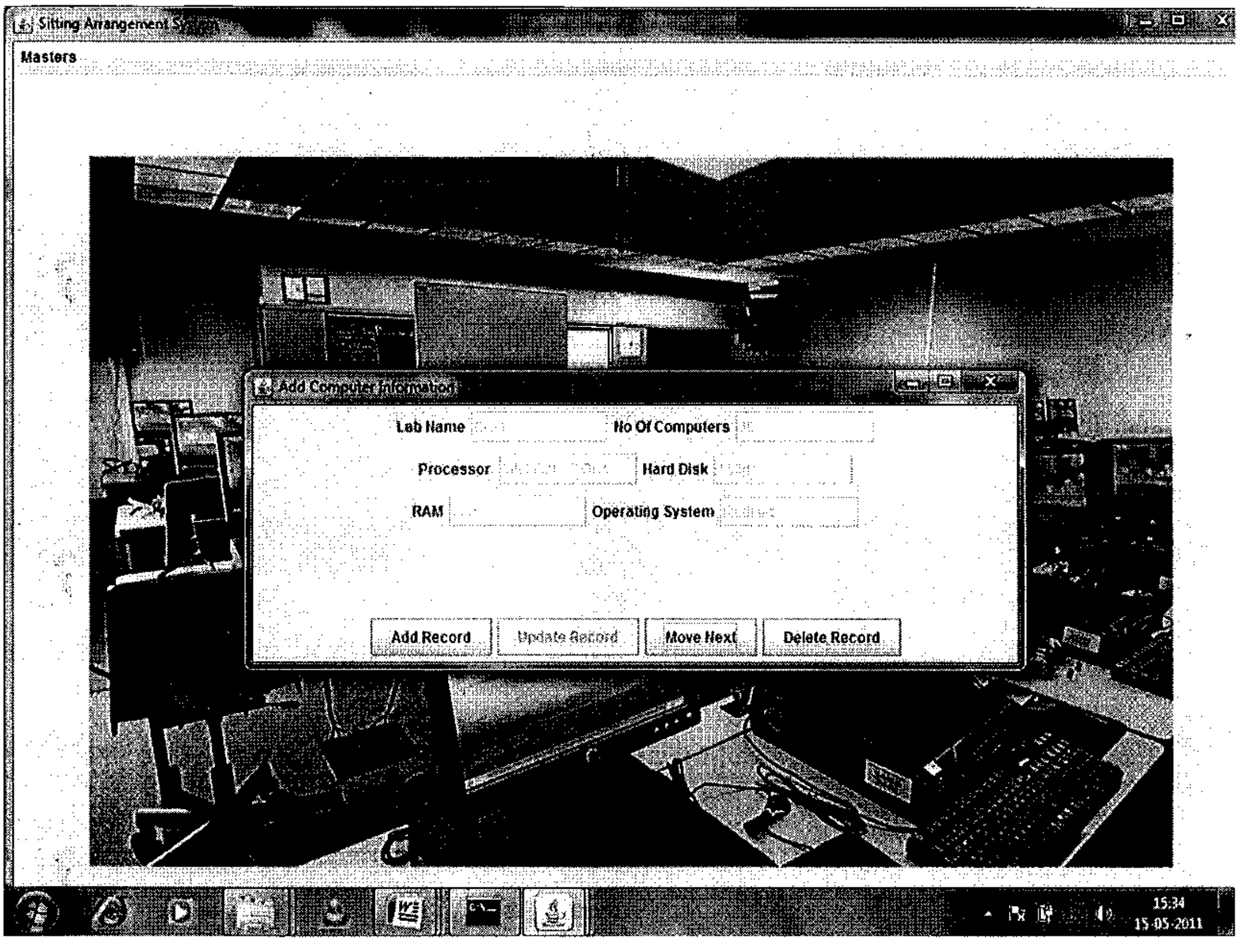
```
t5.setEnabled(false);
t6.setEnabled(false);
        bl.setEnabled(false);
        bf.setEnabled(true);
 return;
        }
       if(s1.equals("Delete Record"))
        {
        a = t1.getText();
        s2 = "delete from computer where labname ="+ "\'" + a + "\'";
        stmt.executeUpdate(s2);
        t1.setText("");
        t2.setText("");
        t3.setText("");
        t4.setText("");
        t5.setText("");
        t6.setText("");
 return;
        }
       else
        {
        if(rs==null)
        rs= stmt.executeQuery("select * from computer");
        rs.next();
        t1.setText(rs.getString("labname"));
        t2.setText(rs.getString("noofcomputer"));
        t3.setText(rs.getString("processor"));
        t4.setText(rs.getString("harddisk"));
        t5.setText(rs.getString("ram"));
        t6.setText(rs.getString("os"));
 return;
        }
```

```
    }
    catch(Exception ee)
     {
      str=ee.getMessage();
      t1.setText(str);
      }
    }
}
```

4.2Frames layouts



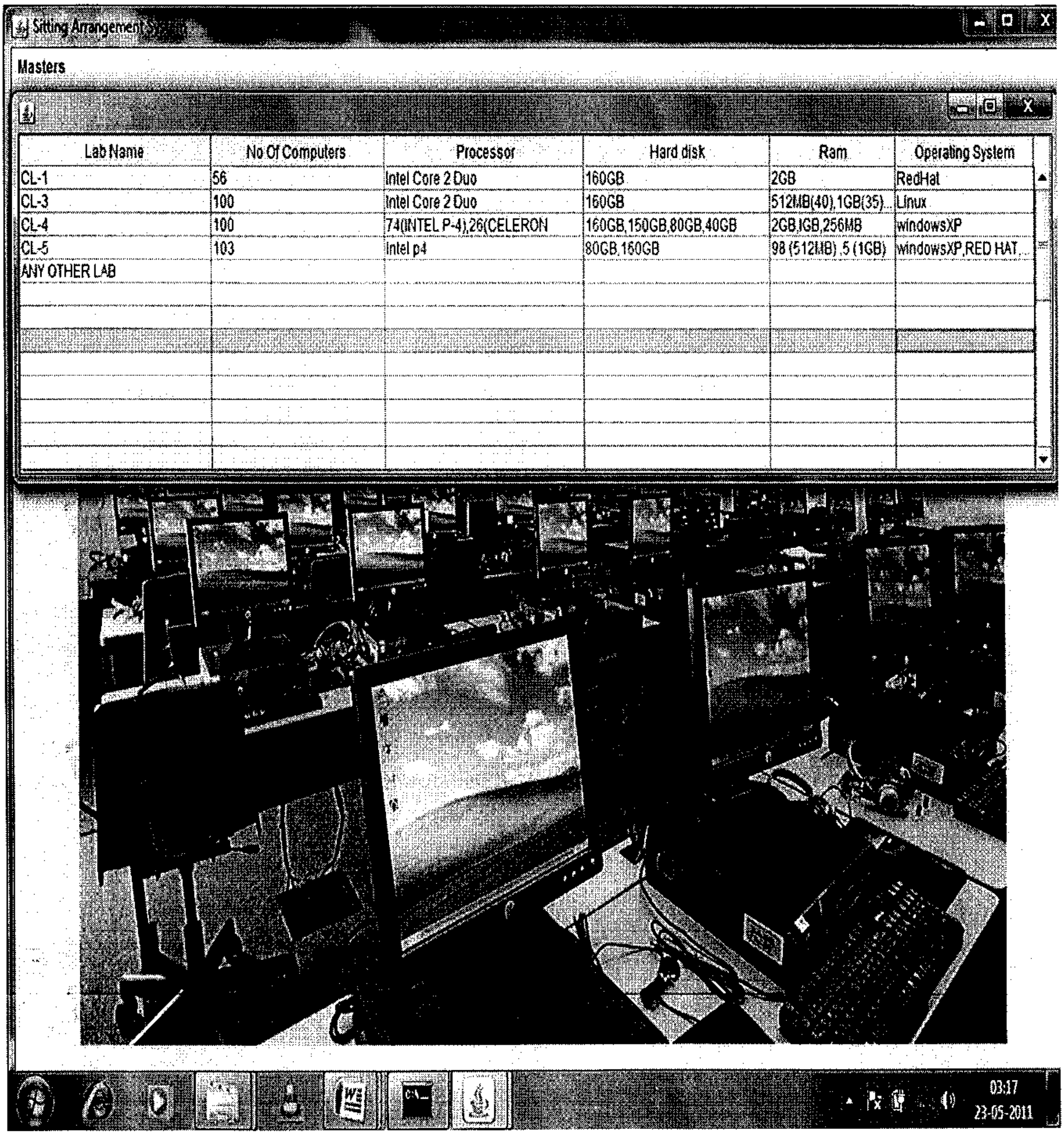

Sitting Arrangement System
Masters
15:32
15-05-2011



Lab Name [illegible] No Of Computers [illegible]

Processor [illegible] Hard Disk [illegible]

RAM [illegible] Operating System [illegible]



Sitting Arrangement System

Masters

Lab Name	No Of Computers	Processor	Hard disk	Ram	Operating System
CL-1	56	Intel Core 2 Duo	160GB	2GB	RedHat
CL-3	100	Intel Core 2 Duo	160GB	512MB(40),1GB(35)...	Linux
CL-4	100	74(INTEL P-4),26(CELERON	160GB,150GB,80GB,40GB	2GB,IGB,256MB	windowsXP
CL-5	103	Intel p4	80GB,160GB	98 (512MB) ,5 (1GB)	windowsXP,RED HAT,...
ANY OTHER LAB					

4.3 MAINTENANCE

The maintenance phase involves making changes to hardware, software, and documentation to support its operational effectiveness. It includes making changes to improve a system's performance, correct problems, enhance security, or address user requirements. To ensure modifications do not disrupt operations or degrade a system's performance or security, organizations should establish appropriate change management standard and procedures.

Routine changes are not as complex as major modifications and can usually be implemented in the normal course of business. Routine change controls should include procedures for requesting, evaluating, approving, testing, installing, and documenting software modifications.

Maintaining accurate, up-to-date hardware and software inventories is a critical part of all change management processes. Management should carefully document all modifications to ensure accurate system inventories. Management should coordinate all technology related changes through an oversight committee and assign an appropriate party responsibility for administering software patch management programs. Quality assurance, security, audit, regulatory compliance, network, and end-user personnel should be appropriately included in change management processes. Risk and security review should be done whenever a system modification is implemented to ensure controls remain in place.

The system has been developed for the given condition and is found working . The developed system is flexible and changes whenever can be made easy. Using the facilities and functionalities of Core Java, the software has been developed in a neat and simple manner, thereby reducing the operators work.

The speed and accuracy are maintained in proper way. The user friendly nature of this software developed in Core Java is very easy to work with both for the higher management as well as other users with little knowledge of computer. The results obtained were fully satisfactory from the user point of view.

The system was verified with valid as well as invalid data in each manner. The system is run with an insight into the necessary modifications that may require in the future. Hence the system can be maintained successfully without much network.

ADVANTAGES:

- It really takes less of time and pain to plan and design the program of such vast dimension.
- It reduces manual work and hence make output more convenient and consistent.
- System is menu driven and therefore it is more user-friendly.
- Authentication checks are built into the system at all levels.
- Existing data validation checks are built in a system.
- Database design is such that it ensure data integrity.

LIMITATIONS:

The software check for the validity of users at start up of application after that all user have all the permissions available. This allows the use of software by external user as this will invite for trouble. It is developed to single user environment.

Project Legacy: The project's development with the aim of satisfactory user requirement on reaching the fringe of completion of the project, it becomes necessary to give back on the development of this project and future possibility.

CURRENT STATUS OF THE PROJECT

The project has been developed keeping in mind the following stages of the System Development Life Cycle (SDLC)

Recognition of need:

At this stage we have checked the existing system, its drawbacks and the basic requirement of software.

Analysis Phase:

We have drawn different flow chart and checked the flow of data.

Design Phase:

At this stage we designed the Tables, Forms, DFD's and flow of data.

Acceptance Criteria: The prime importance is given to involvement of user during whole development the system is very successful to be compatible with requirement specification document.

The system is menu driven window based, so hence it is user friendly. The form for entering data and for selecting the parameter for queries and reports or made very user-friendly and easy to use.

SCOPE OF THE PROJECT

Despite the presence of system in the department's infrastructure all the tasks were being manually maintained & managed. The elegant presence of company system was being underlined by the ubiquitous presence of huge Stacks of files & other stationary.

To count on a few of many drawbacks that plagued the existing system were-

The existing system presented a disorganized look to the department resource.

The existing system was time consuming.

Resources were not properly utilized.

Existing system was decentralized and error-prone.

5. Conclusion

The Software development is a complete process of designing a software from the analysis part of software from user requirements to designing, coding and finally testing the complete functioning of the software in order to certify the accuracy of the software in every aspect. It's a complete process starting from scratch to completion of the whole system. and lastly I would like to thank the project panel and my project guide Mr. Yashwant singh for their kind assistance in my learning process and help in this project.

Thank you

Ashish kapoor – 071293
Ankur vashisth -071312

6.References:

Complete Reference By Herbert Schildt

www.google.com

www.w3schools.com