Project Report

On

**TITLE**

Submitted in fulfillment of the

requirement for the award of the degree of

**Master of Computer Applications**

(Batch 2010 - 2013)



**By**

**Name of the candidate**

**(Roll No.)**

**Under the Supervision**

**of**

(Name of the Internal Supervisor with Designation)

**M.M. Institute of Computer Technology & Business Management**

**Maharishi Markandeshwar University, Mullana (Ambala)**

**May 2013**

**CANDIDATE’S DECLARATION**

I hereby declare that the work which is being presented in this project report entitled **“XYZ”,** in partial fulfillment of the requirement for the award of the degree of **MASTER OF COMPUTER APPLICATIONS,** submitted at M.M. Institute of Computer Technology & Business Management, **Maharishi Markandeshwar University, Mullana, Ambala** is an authentic work done by me during a period from …..…………………… to ………………..………. under the Guidance of **(Internal Supervisor).**

The work presented in this project report has not been submitted by me for the award of any other degree of this or any other Institute/University.

**Signature**

**Name of the Candidate**

**Roll no.**

**This is to certify that the above statement made by the candidate is correct to best of my knowledge and belief.**

**Date :**  **Signature**

**Place : Name of the Supervisor**

**Designation**

**COMPANY CERTIFICATE ON COMPANY’S LETTER HEAD**

**ACKNOWLEDGEMENT**

At the very outset, I am highly indebted to my esteemed HOD for giving me an opportunity to carry out my project on **“OFFICE MANAGEMENT SYSTEM”** at their esteemed organization . I would specially thank **MR. RAJINDER ARORA**, Project Guide, MAHARISHI MARKENDSHWAR UNIVERISTY, MULLANA for giving time and guidance throughout my training without whom it would have been impossible to attain success.

My heartiest thanks to Teachers and employees at MAHARISHI MARKENDSHWAR UNIVERISTY, MULLANA for helping me throughout the project tenure.

I own my regards to the entire faculty of the department of computer science and information technology at **Kurukshetra University** from where I have learnt the basics of Computer Science and whose informal discussions and able guidance was a become light for me in the entire duration of this work.

**Abhishek Somra**

**(2509227)**

# Preface

This project was undertaken at during industrial training Engineering to automate the system .The project is named as **Student award and Degree Generations**

The purpose of this report is to assemble under one cover a sufficient body of knowledge about management and development a successful software engineering project. The following quotes outlines the basic idea behind this technical report. This report assembles various functions like *planning, organizing, designing, testing,* and *maintenance* of the software developed during the Last Semester.

This report is about the adaptation of the techniques of project development and reflects the practice and methods of software engineering project This report is intended for:

• *Project managers*—The report delivers the necessary information of the process a software development project

• *Project coordinators* —The tutorial presents the state of the practice in software development and management techniques.

• *Software engineers, programmers, analysts,and other computer personnel* —The report contains a general description of—and problems in—software engineering project development, plus a number of methodologies and techniques for managing a software development project.

**Table of Contents**

**Title**

1. Acknowledgement
2. Abstract & Preface
3. General Introduction
4. Introduction to Project
5. Planning Phase Analysis Phase
6. Feasibility study and proposed objectives
7. Design & Dataflow Diagram
8. Introduction to JAVA
9. Introduction to AWT
10. Introduction to EVENT HANDLING in JAVA
11. Introduction to APPLET
12. Introduction to OFFICE MANAGEMENT SYSTEM
13. Design & Implementation
14. Screen Shots & their Description
15. Testing
16. Bibliography

### INTROduction

### A computer is a fast electronic machine which accepts the data, process the data at a very fast speed and stores the data for a number of years. The invention of computer had completely revolutionized the human life in mid twentieth century. Early human beings do the calculation manually because these are small in nature. As the time pass the calculation become long and he man have a machine to read calculate and write as per his desire. There is hardly any field left without the use of computer. In almost every field such as science, education, engineering, industries and economy with some contract with computer. In commercial organization, the handling of all the aspects of accounting, billing, sales, recording, inventory of stocks and work scheduling in industries lead to electronic data process.

### Public utility industries like railways, air line reservations, banks, govt. Offices are Using computer in a great extent. The utility of computer is recognized in every field and it is accepted in every developing country.

Now the computers are effecting every commercial organization. Taking the above concept in view, I decided to go through the function of **Employee Health Index System**to develop a system to provide better services to the user and the owner. I got the idea after surveying with my Classmates and my project guide who allotted me the topic.

I decided to go to the function to study the system to provide details of the system of Employee Health Index System

**INTRODCUTION TO PROJECT**

At the time of selection of the topic of the project, though several other topics are available but we choose the particular topic, as we were most well acquainted with this topic. We have the first hand information on this topic. Thus, all the facts we have covered had been very well with. The topic should be selected keeping some important points in mind. Some of them are as follows:-

* The proper information must be available regarding the project.
* The cost factor must be surely kept in mind.
* The time factor employed is the most important for all the facts.

Thus we have selected this topic as all the factors mentioned have been fulfilled.

The aim of this project work is to study the health status of the employees in the company. It will provide information regarding the employee details and tests conducted on the employees, their details etc. The admin can enter the details, delete, modify the details and employee can view the respective details.

The project has been developed as a query based user friendly software which measures the management of employee health. So, it can support the decision making system of the company

The project provides a user-friendly menu driven interface for making various reports and updating of databases. The various salient features of the system are:-

1. This provides an up-to-date database, which supports all the basic operations.
2. Queries can be made to obtain the reports:
   * Updated information of employee and health status
3. Complete security checking of data
4. There is the login facility to prevent unauthorized access.
5. Database entries for the registered Employee are updated and stored.

The system is computerized system in the **java** Language. The proposed system will provide efficient and timely information for the decision making. The proposed system offers a number of options related to the Employee Health Index System The system is very easy to handle and provide easy way for maintaining and Office records on the java.

**REQUIREMENT ANALYSIS**

Requirement Analysis is the first step towards solving the problem statement. It begins with problem statement generated by clients and possibly the developers. The goal of the analysis is to fully specify the problem and application domain without introducing a bias to any particular implementation. The main purpose of analysis model is to understand the requirements what needs to be done, regardless how it will be done.

The problem of Existing System was that the far being for office marks details abd degree generation. So there was a need to develop a system which can be used by any person at any place and at any occurrence of time. So in this system help the user to access this facility for storing the data and quering the data as as when required.

**PROBLEM ANALYSIS:**

The problem analysis deals with the various specifications dealing with the problem being specified. The features are being collected on the basis of various requirement of the client. These requirements are studied and discussed with all the management people who are supervisors and thus a problem specification is developed. These features are integrated and thus made feasible for the development process held in the company.

Problem analysis clearly gives the picture of overall requirement of the system under study and gives the enhanced feature of the new proposed system. Thus it covers all the analysis part for the general system being developed and applied. The project is able to give the enhancement to the manual or computerized system. It provides the user with the facility to work online and save time and energy to work with. Beside it, the module also helps user to take a quick access to the data and can access the query any time and give the preview of degree to student

The application which is being develop at this time provides and easy and consistent way to implement the management software in the office. This application provides the user with the benefit to search for the required data and storing the data . To choose from, the user has the facility to view the different types of options from menus

The requirement was to build the system so as to facilitate the user with the search and allow them to access the data. So it covers the problems dealing with the various issues and services which are to be provided to the application.

The problem analysis phase deals with all such problems and it covers the following steps:

**PRELIMINARY SURVEY:**

This was done to get the first hand knowledge about the working environment. The idea was to know the requirement of the information at different levels. More information as should be possible gathered and analyzed concerning the existing system. This was important so as to know the specific areas that need improvement, to determine specific areas that need to be.

**FEASIBILITY ANALYSIS:**

An initial survey has been made on the proposal; feasibility of the system has been carrying out to check the workability of the system. The feasibility study that describe and evaluate candidate system and provide a solution of the best system that meets system performance requirements with the effective use of resources.

**TECHNICAL** **FEASIBILITY:**

During technical analysis, the technical merits of the system are studied and at the same time collecting additional information about performance, reliability, maintainability and predictability.

Technical analysis begins with an assessment of the technical viability of the proposed system.

* What technologies are required to accomplished system function and performance?
* What new materials, methods, algorithms or processes are required and what is their development risk?
* How will these obtained from technical analysis form the basis for another go/no-go decision on the test system? If the technical risk is severe, if models indicate that the desired function can not be achieved, if the pieces just won’t fit together smoothly-it’s back to the drawing board.

As the software is vary much economically feasible, then it is really important for it to be technically sound. The software will be build among:

Front-End: JAVA

Back-End: MS ACCESS

**ECONOMICAL FEASIBILITY:**

Economical feasibility is the most frequent used method for evaluating the effectiveness of the candidate system. Also commonly known as the cost benefit analysis, i.e. the benefit or output expected from candidate system is compared with the cost of making it. If benefit gives more weight then the cost, then the decision is made to design and implement

the candidate system. Otherwise alternate system has to be considered The proposed system has many cost advantages over the existing system. The online management provides the various profits for the company and thus it is accepted as proposed system.

**Cost Benefit Analysis:**

The various costs evaluated include:

* **Equipment Costs.**

This includes capital and leasing costs of

* + Computer and peripherals.
  + The initial systems supplies. (E.g. disks, tapes, etc.)
  + Additional hardware.
  + UPS and Voltage Regulators.
* **Development Costs**.
  + The purchasing of software. (Programming, DB and OS’s)
  + Consulting work.
  + System analysis and programming.
* **Installation Costs.**

- Preparation of the computer room.

(E.g. wiring, air-conditioning, etc.)

Accommodation (e.g. new building or building alteration,

Etc.)

* **Personal Costs.**
  + Staff training. (Assumption: we will be incurring the cost to train the employees in the Proposed Hospital System)
  + Staff recruitment or relocations.
  + Staff salaries.
  + Redundancy payments.
* **Operating Costs.**
  + Consumable (e.g.- tapes, disks, stationery)
  + Accommodation costs.
  + Power.
  + Insurance.
  + Telecommunications.
  + Standby arrangements.

**LEGAL FEASIBILITY**

The proposed system is made legally feasible as possible as with the enforcement of IT ACT2000 and IT Rules and Regulations2001. It is an online system so it has to be made authenticated and legal.

**OPERATIONAL FEASIBILITY**

Now a days computer based help desk systems have been extensively used by the organizations to handle queries and requests. Also the interest and support shown by the department during the system study don't seem to reflect any probable in this regard. Hence the proposed system is said to be behaviorally feasible.

From the observation made in the feasibility study described above, it **is** recommended that the proposed system is feasible and justified that the feasibility study could be followed by system analysis and the system design.

**SCHEDULE FEASIBILITY**

Time chart for each activity is carefully made in detail by considering all possible aspects. So the system will be available for common public within due time.

**2.2 SYSTEM REQUIREMENT SPECIFICATION DOCUMENT**

Requirement specification appears to be relatively simple task, but appearance is often

deceiving. There is always a chance of wrong specification because of communication gap between user and developer, ambiguity in requirement or a wrong specification problem. Requirement specification begins with a clear statement of the problem and the task to be performed. Then requirements are described in a technical manner in precise statements. After the initial specification reports are received, they are analyzed and refined through user developer interaction.

**INTRODUCTION TO SRS:**

An SRS is basically an organization's understanding (in written) of a customer or potential client's system requirements and dependencies at a particular point in time (usually) prior to any actual design **or** development work. It's a two way insurance policy that assures that both the client and the organization understand the other's requirements from that perspective at a given point in time.

The SRS document itself states in precise and explicit language those functions and capabilities a software system (i.e. a software application, a booking web site and so on) must provide, as well as states any required constraints by which the system must abide. The SRS also function as a blueprint for completing a project as little cost growth as possible. The SRS often refer to as a “parent” document because all subsequent project management documents such as design specifications, testing and validation plans and documentation plans are related to it.

It's important to note that an SRS contains functional and non functional requirements only; it doesn't offer design suggestions, possible solution to technology or business issues, or any other information other than what the development team understands the customer's system requirement to be.

**A well-designed, well-written SRS accomplishes four major goals:**

* It provides feedback to the customer. An SRS is the customer's assurance  
  that the development organization understands the issue or problems to be

solved and the software behavior necessary to address those problems.  
Therefore, the SRS should be written in natural language, in an unambiguous  
manner that may also include charts, tables, data flow diagrams, and decision  
tables and so on.

* It decomposes the problem into component parts. The simple act of writing  
  down software requirements in a well-designed format organizes  
  information, place borders around the problem, solidifies ideas, and helps  
  break down the problem into its component parts into an orderly fashion.
* It serves as an input to the design specification. As mentioned previously,  
  the SRS serves as the parent document to subsequent documents, such as the  
  software design specification and statement of work. Therefore the SRS must contain sufficient detail in the functional system requirement so that a design solution can be devised.
* It serves as a product validation check. The SRS also serves as a parent document for testing and validation strategies that will be applied for the requirements for verification.

**WHAT KIND OF INFORMATION SHOULD AN SRS INCLUDE?**

You probably a member of the SRS team (if not, ask to be), which means SRS development will be a collaborative effort for a particular project. In these cases your company will have develop SRSs before, so you should have examples to use. But let’s assume you will be starting from scratch. Several standard organizations (including the IEEE) have identified nine topics that must be addressed when designing and writing an SRS:

* Interfaces
* Functional capabilities

(9)

* Performance Levels
* Data Structure Elements
* Safety
* Reliability
* Security/Privacy
* Quality
* Constraints and limitation

**General Description**

**Product Perspective:**

This application is an independent product and doesn’t depend on any other application for its normal working. This software interfaces with the data-entry operator’s keyboard and mouse. It makes usage of the standard QWERTY keyboard with special key combinations that are made available to the user to reduce the number of keystrokes that are required to affect common steps in data updation transactions. The mouse under usage is also the normal standardized three-button mouse or optical mouse. Also, this system requires a printer—dotmatrix/inkjet/laser, as per the requirements for the generation of reports to be saved as hard copy or to be provided to the user.

**Product Functions:**

This project requires automating the office marking details For this the system has been analyzed and clearly understood. The customer record till date is maintained on the system,. We are required to make this computerized and also provide access of the order student data

Providing user friendly interface

To store all the information about the student/marks

* Easy access of data
* Easy maintenance
* Maintaining data consistency
* Providing better performance
* Increasing the efficiency through automation
* Proper authorization/Proper security

**Characteristics**

The systems have following characteristics :

* To reduce the workload of the staff
* To have the details of the employees and their health status
* To develop a flexible system according to frequent changes.
* To maintain and update the data.
* User Friendly
* Secured

**Constraints/Assumptions**

The following are the assumptions/constraints assumed in the project:

* The accuracy of output depends of the input data.
* The admin can entered/edit/delete the details. Employee can only view.
* The password to the employee will be allotted by the admin only.
* Only two semesters are considered.

**Specific Requirements**

**Functional Requirements**

The function and performance specification for the system is divided into two major roles: Admin, Operator. These functions can be accessed after going through the Login Screen on loading the menu menu

* **Functional Requirement One**

**Introduction:**

It deals with the administrator’s login that has complete access to all files and records there in.

**Inputs:** The main data that is inputting the data.

**Processing:** The input is processed with the help of a required procedure

**Outputs:** Maximum output is generated through this section. Employee can see his health status.

**Performance Requirements:** Well handled design time errors and minimal generation of run-time errors and if any generated be handled carefully and effectively through centralized error handling.

**Design Constraints:**

**Software Limitations:**

* Microsoft Windows NT/XP 192 MB RAM
* JDK1.6 or JDK 1.7
* Microsoft Acess 2003/2007
* **Hardware Limitations:**

1) PC with Pentium II Processor, 450MHz (Recommended Pentium III Processor, 800MHz)

2) 192 MB RAM

3) Minimum 1.2 GB hard disk space.

4) CD-ROM (48 X or higher recommended).

5) Mouse or Similar Pointing device.

6) A Printer to take out Reports.

**SYSTEM DESIGN:**

The purpose of Design phase is to plan a solution for problem specified by the requirement document. In other words, starting with what is needed; design takes us towards how to satisfy the needs.

The design of a system is most critical factor affecting the quality of the software. It is an iterative process through which the requirements are translated into a blue-print for constructing the software or the system.

**PRELIMINARY INVESTIGATION :**

Preliminary Investigation is generally conducted to determine what type of the Computer Information System is needed. Computer information systems serve many different purposes. All the purposes must be thoroughly investigated before any further development to take place. Preliminary investigation is generally carried out by the System Analyst. But in this project I have analysis the entire system and finds alternative solution for the existing system. The collection of facts is also called preliminary investigation. I took the appointments with office officer and the staff to collect the information regarding their requirements.

**METHODS OF INVESTIGATION :**

For investigating the existing system of the organization I have used the following methods ;-

* Onsite observation/Review of the previous literature of the organization
* Questioning
* Interviewing

**Onsite Observation :**

It means to visit the office premises and to observe the working. For this I have visited the office a number of times. During my visit in the premises, I get the information about details of employees and their health status. Also I got the knowledge regarding the working procedure.

I also underwent various records stored by office personally in the manual registers of the previous year from him, all the procedure reports of the office gives me the true picture of maintaining the data.

I inspected a number of details there. Then I have seen how they are they are prepared.

**Questioning :**

For the further investigations of the system I have made the following questions?

I have asked the following questions to get the details about the system

* What is your existing system ?
* Why you require the computerization of your system ?
* What is the use of this management system?
* What is the health index chart
* What are the details of the employees are required?

**Interviewing:**

I have interviewed various staff and officer of the office and put the above set questions from them. Another they were not responding to my questions because they were thinking that there job will be in danger because of the computerization. But I told them the computerization will help in your working and your job will not be affected and you will be given a proper training to work in the software. Then they respond in a very good way on my interview.

I have also gone through the details of Employee and collect the necessary information regarding organization. I studied all that facts and developed the project in the phased manner. These phases are the part of my project work.

I have asked the aforesaid questions which are explained in onsite observations. In addition to this following information was collected by me:-

* Information about the companies
* Information about the Offices

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

1. ARCHITECTURAL DESIGN OR HIGH LEVEL DESIGN . DETAILED

2. DESIGN, LOW LEVEL DESIGN OR PHYSICAL DESIGN

**ARCHITECTURAL DESIGN (HIGH LEVEL DESIGN):**

The Top Level Design aims to aims to identify the modules that should be in the system, the specification of these modules, and how they interact with each other to produce the desired results. A: the end of the system designs all the major data structures, file formats, output formats, and the major modules in the system and their specification are decided.

It includes the following steps:

1. **Identifying the Entities:**

All the entities related to the model were identified, checked and consolidated. The following are the main Entities included: Student Teacher and marks details

**2. Identifying the Relationships:**

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

**3. Attribute Definition:**

Attributes for each entity were identified and their field characteristics were identified. At the end of the logical design, a system modeled on DBMS should be fully organized. Further, though a fully normalized design may not be the design adopted in the final physical model ( for the purpose of optimality), it is still desirable to fully normalize the design and then demoralize it in a controlled manner to take care of optimality. Since normalization is body of rules addressing analysis and conversion of data structure into relation that exhibit more desirable properties of internal consistency, minimum redundancy and maximum stability.

**4.** **Data Normalization:**

The focus of normalization is on reducing storage overheads.

Especially the aims are:

* To identify and isolate the entities involved.
* To remove redundant information.
* To determine the identifying attributes of records.

An entity is something exits. It may be something tangible or it may even be just a relationship. Every occurrence of an entity will have some discernible attributes. Removing redundancy the second aim of normalization is the other way of eliminating waste space. The last aim or having identifying attribute can be accessed. It also facilitates linking of tables.

**5. Building of database:**

The general theme behind a database is to handle information as an integrated whole. The database used in this project consists of tables, which are defined in MS-Access.

Analysis means the study of existing system, in other words we can say that how

**OFFICE REGISTER**

The purpose of this register is to enter each and every student details. The following is the format for student register.

**LAYOUT FOR STUDENT REGISTER**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NAME** | **ROLL NO.** | **EMAIL ID** | **BRANCH** | **SEMESTER** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**TEACHER REGISTER**

The purpose of this register is to store the details of teacher

**LAYOUT FOR TEACHER REGISTER**

|  |  |  |  |
| --- | --- | --- | --- |
| **NAME** | **ID** | **BRANCH** | **PHONENO** |
|  |  |  |  |
|  |  |  |  |

**MARKS REGISTER**

The purpose of this register is to store the marks details of the student.

**LAYOUT FOR MARKS REGISTER**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STUDENT** | **ROLL NO.** | **SUB1** | **SUB2** | **SUB3** | **SUB4** | **SUB5** |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

**DRAWBACKS OF THE EXISTING SYSTEM**

The existing system have the following drawbacks:-

1. Wastage of time.

2. Difficulty in management of student and company data.

3. Daily progress cannot judge properly due to a large amount of of data.

4. Lots of manpower and paper work is required as a result of which the existing system is too costly as well as slow process.

5. There is chance of error .

**ALTERNATIVE SOLUTIONS**

There are a number of alternative solutions. Those are as follows.

To maintain the data of Employee and companies properly the first solution is to increase the Office officers subordinates. It is efficient way to store the data but the subordinates can’t be increased. The experienced Office officer as well as his subordinates is required and he will be paid a high salary. But the office does not want to pay such a salary monthly.

The second way is to computerize the process of office . It is the easiest and simple way because the office had already the computer for this work. Only the software is required.

The Third alternative is to have some external system and provide them data for future storage but that is not a efficient way because the office does not want to gives it details to the outsiders due to the secrecy of the data.

**DESIGN AND DATAFLOW DIAGRAM**

**SYSTEM DESIGN**

**INTRODUCTION :**

Software development is a team work. It requires a lot of work between system analysts, programmer and end user. But here in this project I am working as the system analyst and programmers both. For designing of the present system I visited the office . There I met the Office officer and other staff of the office and I got the details about the system. From the Office officer I come to know about the various reports he wants. After studying all the above data and material provided by the Office officer and staff I designed the files various input screens & reports which are the included in my project.

The system design is a important because it reflects the basic understanding of the problem. For designing a good system what we have to do is to get correct definition of the problem and data. The main features and user requirement must be kept in mind.

**OUTLINES OF INPUT,OUTPUT &FILE DESIGN**

**INPUT DESIGN :**

Input Design means providing the interface for data entry. The main objective of input design is to make data entry easy, logical and free from errors. Correct input data is necessary for the efficiency and effectiveness of the system because the user of the system may not be expert person and may not know how to enter the data. Messages have been provide to make data entry easy

**OUTPUT DESIGN :**

The objective of the output design is to represent the data in a well defined order. The output is the main source of providing the information for further action.

**DATABASE DESIGN**

The required following database

* Employee
* Health Record

Structure for Database File : Employee

No. Field Name Type

1. name text

2. emailed text

3. dept text

4. sex text

5. Age-group text

6. Cdate date

7. Ndate Date

8. Phone No Number

Structure for Database File: Health Record

No. Field Name Type

1. name text

2. emailed text

3. test1 text

4. test2 text

5. test3 text

6. test4 text

7. test5 text

8. Date date

8. IndexNo Number

**CODING &TESTING CODING**

**SELECTION OF LANGUAGE**

Different programmer use different programming language for designing the software. Before writing the software, One must select the language in which the software is to be developed. I have selected the **java** my project work . While selecting the language I have kept the following features of language selection in my mind.

**1. Purpose :** There are different languages for every application. The language must be selected according the requirement of the software to be developed. The language must provide all the features through which the user needs can to meet.

**2. Simplicity :** Programming language that are simple and easy to learn and used for software development. java is a simple and easy language for software development.

**3.** **Nature less :** A Programming language should be natural for application area it has been designed. Programming language java can be used for application program.

**4.** **Efficiency :** Programming language **java** is efficient and fast in processing. It can handle a large volume of data any time

5. Structure ness : Programming language **java** is a structural language. A program can be divided in modules and functions.

**6. Extendable :** Programming language **java** is extendable. Any new feature in the software can be added or removed very easily.

**7. Suitable to environment :** Programming language **java** can be implemented on a different of operating system like Dos, Windows, Unix etc.

**8.Efficiency of Programmer :** That language should be used in which programmers can easily be programmed the project. I have chosen **java** because I have complete knowledge of the language.

Programming Practice

The goal of the coding & programming phase is to translate the design of the system produced during the design phase into code in a given programming language, which can be executed by a computer and that performs the computation specified by the design.

the coding phase affects both testing & maintenance profoundly. It should be clear that the goal during coding should not be reduce the implementation cost, but the goal should be to reduce the cost of later phases, even if it means that the cost of this phase has to increase. The goal during this phase is not to simply the job of the programmer. Rather, the goal should be to simplify the job of the tester and the maintainer.

good programming is a skill that can only be acquired by practice. However, much can be learned from the experience of other, and some general rules and guidelines can be laid for the programmer. Good programming (producing correct and simple programs) is a practice independent of the target programming language. Following are the concepts related to coding in a language-independent manner.

* Top-down approach
* Bottom-up approach
* Structured programming
* Information hiding

Programming style

It is impossible to provide an exhaustive list of what to do and what to do to produce simple readable code. Some general rules are listed here.

1. Names: selecting module and variable names is considered very important. They have kept very closely related to the entity they represent, and module names should reflect their activity.
2. Control constructs: the care has been taken for using entry & single exit constructs as much as possible.
3. Use of goto: go to statement is used rarely & sparingly and in disciplined manner. In any case, alternatives were tried before finally using a go to statement. When go to be must, at that time only forward transfer of control is used instead of jumping backward.
4. Nesting: the differ control constructs, particularly the if-then-else, can be nested. If the nesting become too deep, the programs become harder to understand. In case of deeply nested if-then-else, so too deep nesting is avoided as much as possible.
5. Module size: large module often will not be functionally cohesive, and too small module might incur unnecessary overhead. There can be no hard & fast rule about module sizes.
6. Module interface: if any module whose interface has more than five parameters is carefully examined and brooked into multiple modules with a simpler interface.

Code efficiency & optimization of code

Following steps were taken for improving the efficiency of code & to optimize the code.

Verification of output: it is primarily intended for detecting errors in introduced during this phase. That is, the goal of the code produced is to show that the code is consistent with the design it is supposed to implement.

Program verification method falls in 2 categories. These are *static & dynamic*. Both the methods were used for verification of output.

Code reading: this process is reverse of design. In design we start from an abstraction & move towards more details. In code reading we starts from the details of a program and move toward an abstract description.

Symbolic execution: another approach where the program is not executed with actual data. Instead the program is “symbolically executed” with symbolic data. The outputs are symbolic formulas of input values.

**INTRODUCTION TO JAVA**

**Introduction:**

Java is an object-oriented programming language with a built-in application programming interface (API) that can handle graphics and user interfaces and that can be used to create applications or applets. Because of its rich set of API's, similar to Macintosh and Windows, and its platform independence, Java can also be thought of as a platform in itself. Java also has standard libraries for doing mathematics.

Much of the syntax of Java is the same as C and C++. One major difference is that Java does not have pointers. However, the biggest difference is that you must write object oriented code in Java. Procedural pieces of code can only be embedded in objects. In the following we assume that the reader has some familiarity with a programming language. In particular, some familiarity with the syntax of C/C++ is useful.

In Java we distinguish between applications, which are programs that perform the same functions as those written in other programming languages, and applets, which are programs that can be embedded in a Web page and accessed over the Internet. Our initial focus will be on writing applications. When a program is compiled, a byte code is produced that can be read and executed by any platform that can run Java.

**Characteristics:**

**1) Platform Independent**

Java is a platform for application development. A platform is a loosely defined computer industry buzzword that typically means some combination of hardware and system software that will mostly run all the same software.

Java byte code is exactly the same on every platform.. Java programs that have been compiled into byte code still need an interpreter to execute them on any given platform. The interpreter reads the byte code and translates it into the native language of the host machine on the fly. Since the byte code is completely platform independent, only the interpreter and a few native libraries need to be ported to get Java to run on a new computer or operating system.

All these pieces, the javac compiler, the java interpreter, the Java programming language, and more are collectively referred to as Java.

**2) Object oriented**

In object-oriented programs data is represented by objects. Objects have two sections, fields (instance variables) and methods. Fields tell you what an object is. Methods tell you what an object does. These fields and methods are closely tied to the object's real world characteristics and behavior. When a program is run messages are passed back and forth between objects. When an object receives a message it responds accordingly as defined by its methods.

**3) Robust**

Java implements a robust exception handling mechanism to deal with both expected and unexpected errors. The worst that an applet can do to a host system is bringing down the runtime environment. It cannot bring down the entire system.

Most importantly Java applets can be executed in an environment that prohibits them from introducing viruses, deleting or modifying files, or otherwise destroying data and crashing the host computer. A Java enabled web browser checks the byte codes of an applet to verify that it doesn't do anything nasty before it will run the applet.

**4) Multithreaded**

Java is inherently multi-threaded. A single Java program can have many different threads executing independently and continuously. Three Java applets on the same page can run together with each getting equal time from the CPU with very little extra effort on the part of the programmer.

**5) High performance**

Java byte codes can be compiled on the fly to code that rivals C++ in speed using a "just-in-time compiler." Several companies are also working on native-machine-architecture compilers for Java. These will produce executable code that does not require a separate interpreter, and that is indistinguishable in speed from C++.

**6) Dynamic**

Java programs carry with them substantial amounts of run-time type information that is used to verify and resolve accesses to objects at run-time. This makes it possible to dynamically link code in a safe and expedient manner. The java programming language is unusual than other programming languages it first compiles and then interprets the program. Compile first translate the program into intermediate language called intermediate language called java byte code. Java byte code is platform independent code, which is further interpreted by the interpreter on the java platform. Interpreter parses and run each java byte code instruction on the computer. Compilation occurs only once, interpretation occurs each time when the program is executed.



Java byte code helps in making the program “write once, run anywhere”. The program can be compiled into byte code by any platform that is having the java compiler; the compiled java byte code program is ready to run on any machine having the java virtual machine (JVM). JVM is an interpreter for byte code.

**Java is too effective:**

**Java's Cool**

Java has caused more excitement than any development on the Internet since Mosaic. Java was the first way to include inline sound and animation in a web page. Java also lets users interact with a web page. Instead of just reading it and perhaps filling out a form, users can now play games, calculate spreadsheets, chat in real-time, get continuously updated data and much, much more.

Here are just a few of the many things Java can do for a web page:

* Inline sounds that play in real-time whenever a user loads a page
* Music that plays in the background on a page
* Cartoon style animations
* Real-time video

**What makes Java special?**

Java is a programming language for distributed applications. It doesn't just allow you to add new types of content to your pages like Netscape and Internet Explorer do. Rather it lets you add both the content and the code

necessary to interact with that content .

e.g. Let’s say you want to use EPS files on your Web site. Previously you had to wait until at least one web browser implemented EPS support. Now with the help of java you can write your own code to view EPS files and send it to any student that requests your page at the same time they request the EPS file.

e.g. Suppose you want people to be able to search your electronic card catalog. However the card catalog database exists on a mainframe system that doesn't speak HTTP. Before Java you could hope that some browser implemented your proprietary card catalog protocol or you could try to program some intermediate cgi-bin on a UNIX box that can speak HTTP and talk to the card catalog, not an easy task. With Java when a student wants to talk to your card catalog you can send them the code they need to do so. You don't have to try to force things through an httpd server on port 80 that were never meant to go through it.

If that were all Java was, it would still be more interesting than a <marquee> or <frame> tag in some new browser beta.

Java is platform independent. A Java program can run equally well on any architecture that has a Java enabled browser. With the release of Netscape Navigator 2.0 that includes Windows 95, Windows NT, the MacOS, Sun Solaris, Sun OS 4.1.3, SGI IRIX, OSF/1, HP-UX with more to come.

Java isn't just for web sites. Java is a programming language that lets you do almost anything you can do with a traditional programming language like Fortran.

**Advantages of Java:**

**1) Simple**

Java has the bare bones functionality needed to implement its rich feature set. It does not add lots of syntactic sugar or unnecessary features.

**2) Object-Oriented**

Almost everything in Java is either a class, a method or an object. Only the most basic primitive operations and data types (int, for, while, etc.) are at a sub-object level.

**3) Platform Independent**

Java programs are compiled to a byte code format that can be read and run by interpreters on many platforms including Windows 95, Windows NT, and Solaris 2.3 and later.

**4) Safe**

Java code can be executed in an environment that prohibits it from introducing viruses, deleting or modifying files, or otherwise performing data destroying and computer crashing operations.

**5) High Performance**

Java can be compiled on the fly with a Just-In-Time compiler (JIT) to code that rivals C++ in speed.

**6) Multi-Threaded**

Java is inherently multi-threaded. A single Java program can have many different things processing independently and continuously.

**Object Oriented Principles**

The object oriented programming languages provide mechanisms that help you implement the object-oriented model.

**1) Data Hiding and Encapsulation**

One of the important object-oriented techniques is hiding the data within the class and making it available only through the methods. This technique is known as encapsulation because it seals the data (and internal methods) safely inside the "capsule" of the class, where it can be accessed only by trusted users (i.e., by the methods of the class). The most important reason is to hide the internal implementation details of your class. If you prevent programmers from relying on those details, you can safely modify the implementation without worrying that you will break existing code that uses the class. Another reason for encapsulation is to protect your class against accidental or willful stupidity. A class often contains a number of interdependent fields that must be in a consistent state. If you allow a programmer (including yourself) to manipulate those fields directly, he may change one field without changing important related fields, thus leaving the class in an inconsistent state. If, instead, he has to call a method to change the field, that method can be sure to do everything necessary to keep the state consistent. Similarly, if a class defines certain methods for internal use only, hiding these methods prevents users of the class from calling them. When all the data for a class is hidden, the methods define the only possible operations that can be performed on objects of that class. Once you have carefully tested and debugged your methods, you can be confident that the class will work as expected. On the other hand, if all the fields of the class can be directly manipulated, the number of possibilities you have to test becomes unmanageable. There are other reasons to hide fields and methods of a class, as well:

* Internal fields and methods that are visible outside the class just clutter up the API. Keeping visible fields to a minimum keeps your class tidy and therefore easier to use and understand.
* If a field or method is visible to the users of your class, you have to document it. Save yourself time and effort by hiding it instead.

**2) Inheritance**

Inheritance is the process by which one object acquires the properties of another object. This is important because it supports the concept of hierarchical classification. Most knowledge is made by hierarchical classification. Inheritance is a compile-time mechanism in Java that allows you to extend a class (called the base class or super class) with another class (called the derived class or subclass).

In Java, inheritance is used for two purposes:

* Class inheritance - create a new class as an extension of another class, primarily for the purpose of code reuse. That is, the derived class inherits the methods of the base class.
* Interface inheritance - create a new class to implement the methods defined as part of an interface for the purpose of sub typing. That is a class that implements an interface “conforms” to the interface.

In Java, unlike C++, these two types of inheritance are made distinct by using different language syntax. For class inheritance, Java uses the keyword extends and for interface inheritance Java uses the keyword implements.In Java, unlike C++, only single class inheritance is supported. I.e., for a given class, there is only one super class.

**3) Polymorphism**

Polymorphism is the capability of an action or method to do different things based on the object that it is acting upon. This is the third basic principle of object-oriented programming.

Two types of polymorphism:

* Overloading and Overriding.
* Dynamic method binding.

**Java platform:**

A java platform is the hardware or software environment in which a program runs. The java platform has two components: Java Virtual Machine (JVM)

1.Java Application Programming Interface (Java API)

2.JVM is a standardized hypothetical computer, which is emulated inside your computer by a program.

Java Object Code

Java Source Code

Java Compiler

Java Virtual Machine

The program By the interpreter

Java interpreter

Computer Operating System

The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI). The java API is grouped into libraries of related classes and interfaces these libraries are known as packages.



**Programming language**

**1) Java language fundamentals**

* Java is purely object-oriented programming language.
* Java is neither a superset nor a subset of C / C++.

**2) Java and C**

* Java does not include keywords like goto , sizeof and typedef.
* Java does not contain data types like struct, union and enum.
* Java does not define the type modifiers keywords like auto, register, signed.
* Java does not support an explicit pointer type.
* Java does not have preprocessor.
* Java does not support variables arguments to functions.
* Java adds many features required for Object-Oriented Programming.

**3) Java and C++**

* Java does not support operator overloading.
* Java does not support multiple inheritances.
* Java does not support global variables.
* Java has replaced the destructor function with a finalize() function.
* Java is a true object-oriented language while C++ is basically C with object-oriented extension.
* Java does not provide the access to the pointers.
* **General Style:**

In order to write a program in Java, one has to follow the well-defined structure of the Java program and should have the basic idea of all the programming constructs like statements, tokens, etc.

Following points are to be noted:

**Java is case-sensitive language.**

In Java, uppercase and lowercase letters are not the same.Java is freeform language. Freeform means that one need not have to indent any lines to make the program work properly, but it is good practice to use indent to make the program readable.

**INTRODUCTION TO AWT**

**Introduction:-**

The Java programming language class library provides a user interface toolkit called the Abstract Windowing Toolkit, or the AWT. The AWT is both powerful and flexible. Newcomers, however, often find that its power is veiled. The class and method descriptions found in the distributed documentation provide little guidance for the new programmer. Furthermore, the available examples often leave many important questions unanswered. Of course, newcomers should expect some difficulty. Effective graphical user interfaces are inherently challenging to design and implement, and the sometimes complicated interactions between classes in the AWT only make this task more complex. However, with proper guidance, the creation of a graphical user interface using the AWT is not only possible, but relatively straightforward.This article covers some of the philosophy behind the AWT and addresses the practical concern of how to create a simple user interface for an applet or application.

**What is a User Interface?**

The user interface is that part of a program that interacts with the user of the program. User interfaces take many forms. These forms range in complexity from simple command-line interfaces to the point-and-click graphical user interfaces provided by many modern applications. At the lowest level, the operating system transmits information from the mouse and keyboard to the program as input, and provides pixels for program output. The AWT was designed so that programmers don't have worry about the details of tracking the mouse or reading the keyboard, nor attend to the details of writing to the screen. The AWT provides a well-designed object-oriented interface to these low-level services and resources. Because the Java programming language is platform-independent, the AWT

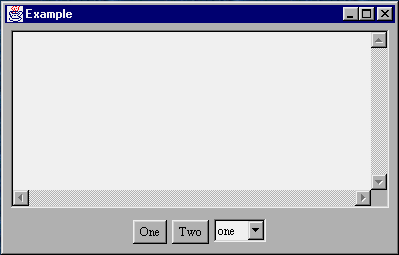
must also be platform-independent. The AWT was designed to provide a common set of tools for graphical user interface design that work on a variety of platforms. The user interface elements provided by the AWT are implemented using each platform's native GUI toolkit, thereby preserving the look and feel of each platform. This is one of the AWT's strongest points. The disadvantage of such an approach is the fact that a graphical user interface designed on one platform may look different when displayed on another platform.

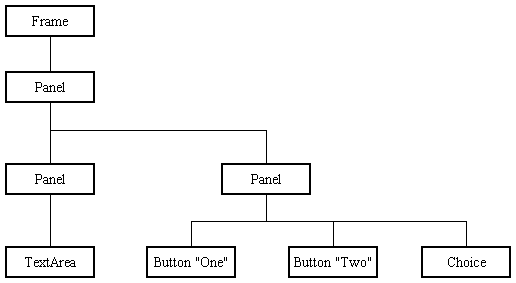
**Components and Containers**

A graphical user interface is built of graphical elements called components. Typical components include such items as buttons, scrollbars, and text fields. Components allow the user to interact with the program and provide the user with visual feedback about the state of the program. In the AWT, all user interface components are instances of class Component or one of its subtypes.  Top of Form

[Click here to find out more!](http://ad.doubleclick.net/click;h=v8/359e/0/0/*/p;44306;0-0;0;14363614;31-1/1;0/0/0;;~sscs=?)[http://ad.doubleclick.net/ad/idg.us.nwf.jw_core/;tile=99;sz=1x1;ord='%20+%20ord%20+%20'?](http://ad.doubleclick.net/jump/idg.us.nwf.jw_core/;tile=99;sz=1x1;ord='%20+%20ord%20+%20'?)Spatially, components must fit completely within the container that contains them. This nesting of components (including containers) into containers creates a tree of elements, starting with the container at the root of the tree and expanding out to the leaves, which are components such as buttons.

The illustration in Figure 1 depicts a simple graphical user interface as it would look when displayed under Windows 95. Figure 2 shows the interface components from Figure 1 arranged as a tree.

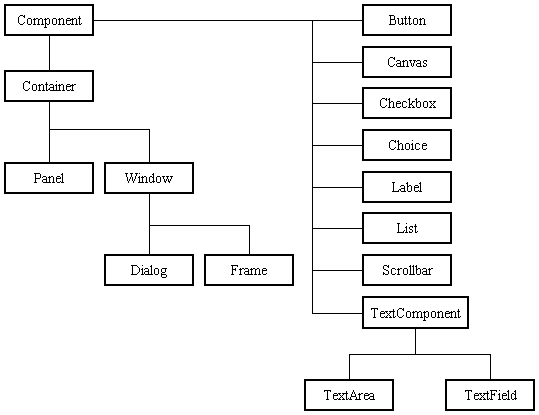
  
**Figure 1. A simple graphical user interface**



**Figure 2. User interface component tree**

**Types of Components:**

Figure 3 shows the inheritance relationship between the user interface component classes provided by the AWT. Class Component defines the interface to which all components must adhere.

  
**Figure 3. The inheritance rela**tionship

The AWT provides nine basic non-container component classes from which a user interface may be constructed. (Of course, new component classes may be derived from any of these or from class Component itself.) These nine classes are class Button, Canvas, Checkbox, Choice, Label, List, Scrollbar, TextArea, and TextField.

**Types of Containers**

The AWT provides four container classes. They are class Window and its two subtypes -- class Frame and class Dialog -- as well as the Panel class. In addition to the containers provided by the AWT, the Applet class is a container -- it is a subtype of the Panel class and can therefore hold components. Brief descriptions of each container class provided by the AWT are provided below.

|  |  |
| --- | --- |
| **Window** | A top-level display surface (a window). An instance of the Window class is not attached to nor embedded within another container. An instance of the Window class has no border and no title. |
| **Frame** | A top-level display surface (a window) with a border and title. An instance of the Frame class may have a menu bar. It is otherwise very much like an instance of the Window class. |
| **Dialog** | A top-level display surface (a window) with a border and title. An instance of the Dialog class cannot exist without an associated instance of the Frame class. |
| **Panel** | A generic container for holding components. An instance of the Panel class provides a container to which to add components. |

**Creating a Container**

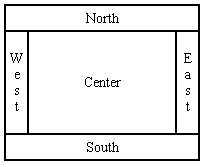
Before adding the components that make up a user interface, the programmer must create a container. When building an application, the programmer must first create an instance of class Window or class Frame. When building an applet, a frame (the browser window) already exists. Since the Applet class is a subtype of the Panel class, the programmer can add the components to the instance of the Applet class itself.

**Adding Components to a Container**

To be useful, a user interface must consist of more than just a container -- it must contain components. Components are added to containers via a container's add() method. There are three basic forms of the add() method. The method to use depends on the container's layout manager (see the section titled Component layout).

**Component Layout**

Up to this point, nothing has been said about how the components that have been added to a container are laid out. Layout is controlled not by the container, but by a layout manager associated with the container. The layout manager makes all of the component Office decisions. In the AWT, all layout manager classes implement the LayoutManager interface. The AWT provides five layout managers. They range from very simple to very complex. This article covers only the two layout manager classes used by the examples herein: the FlowLayout class and the BorderLayout class. The FlowLayout class places components in a container from left to right. When the space in one row is exhausted, another row is started. The single-argument version of a container's add() method is used to add components. The BorderLayout class has five zones as depicted in Figure 7. The zones are named "North", "South", "East", "West", and "Center". A single component can be placed in each of these five zones. When the enclosing container is resized, each border zone is resized just enough to hold the component placed within. Any excess space is given to the center zone. The two-argument version of a container's add() method is used to add components. The first argument is a String object that names the zone in which to place the component.

  
**Figure 7. The border layout zones**

Each container class has a default layout manager. The default layout manager for the Frame class and Dialog class is the BorderLayout manager. The default layout manager for the Panel class (and the Applet class) is the FlowLayout manager.

**EVENT HANDLING IN JAVA**

**INTRODUCTION:**

You are leaving for work in the morning and someone rings the doorbell….  
That is an event!

In life, you encounter events that force you to suspend other activities and respond to them immediately. In Java, events represent all activity that goes on between the user and the application. Java’s Abstract Windowing Toolkit (AWT) communicates these actions to the programs using events. When the user interacts with a program let us say by clicking a command button, the system creates an event representing the action and delegates it to the event-handling code within the program. This code determines how to handle the event so the user gets the appropriate response.

In today’s tutorial we are going to learn event-driven programming, the event model of Java, and the different ways in which you can handle events.

**Components of an Event:**

Can be put under the following categories.  
**1. Event Object:** When the user interacts with the application by clicking a mouse button or pressing a key an event is generated. The Operating System traps this event and the data associated with it. For example, info about time at which the event occurred, the event types (like keypress or mouse click). This data is then passed on to the application to which the event belongs. In Java, events are represented by objects, which describe the events themselves. And Java has a number of classes that describe and handle different categories of events.  
**2. Event Source:** An event source is the object that generated the event. Example if you click a button an ActionEvent Object is generated. The object of the ActionEvent class contains information about the event.  
**3. Event-Handler**: Is a method that understands the event and processes it. The event-handler method takes the Event object as a parameter. Java uses Event-Delegation Model :with JDK1.1 onwards; you can specify the objects that are to be notified when a specific event occurs. If the event is irrelevant, it is discarded. The four main components based on this model are Event classes, Event Listeners, Explicit event handling and Adapters. Let us take a closer look at them one by one.  
**Event Classes:**

The EventObject class is at the top of the event class hierarchy. It belongs to the java.util package. While most of the other event classes are present in java.awt.event package. The getSource() method of the EventObject class returns the object that initiated the event. The getId () method returns the nature of the event. For example, if a mouse event occurs, you can find out whether the event was click, a press, a move or release from the event object. AWT provides two conceptual types of events: Semantic and low-level events.

**Semantic events:**

These are defined at a higher-level to encapsulate the semantics of user interface component’s model. Now let us see what are the various semantic event classes and what they generate:   
  
• An Action Event object is generated when a component is activated  
• An Adjustment Event Object is generated when scrollbars and other adjustment elements are used.  
• A Text Event object is generated when text of a component is modified.  
• An Item Event is generated when an item from a list, a choice or checkbox is selected.  
Low-Level Events is one that represents a low-level input or windows-system occurrence on a visual component on the screen. The various low-level event classes and what they generate are as follows:   
• A Container Event Object is generated when component are added or removed from container.  
• A Component Event object is generated when a component is resized, moved etc.  
• A Focus Event object is generated when component receives focus for input.

• A Key Event object is generated when key on keyboard is pressed, released etc.  
• A Window Event object is generated when a window activity, like maximizing or close occurs.  
• A Mouse Event object is generated when a mouse is used.  
• A Pain Event object is generated when component is painted.  
  
**Event Listeners:**

An object delegates the task of handling an event to an event listener. when an event occurs, an event object of the appropriate type (as illustrated below) is created. This object is passed to a Listener. A listener must implement the interface that has the method for event handling. A component can have multiple listeners, and a listener can be removed using removeActionListener () method. Next question in your mind must be what is an interface?. An Interface contains constant values and method declaration. The difference between classes and interface is that the methods in an interface are only declared and not implemented, that is, the methods do not have a body. What is the Need for interface? Are interfaces are used to define behavior protocols (standard behavior) that can be implemented by any class anywhere in the class hierarchy. The java.awt.event package contains definitions of all event classes and listener interface. The semantic listener interfaces define by AWT for the above mentioned semantic events are:  
• Action Listener   
• Adjustment Listener   
• Item Listener   
• Text Listener   
The low-level event listeners are as follows:   
• Component Listener   
• Container Listener   
• Focus Listener   
• Key Listener   
• Mouse Listener   
• Mouse Motion Listener   
• Windows Listener.

**TESTING**

**OVERVIEW**

The philosophy behind testing is to find errors. The common view of testing is that it is performed to prove that there are no errors in the program. However it is virtually impossible since no program will be clear and free from errors. Therefore the useful and practical approach for tracing the errors should be employed. Testing is the process of executing the program with the explicit intention of finding errors, which makes the program, fails. Testing is an activity, however, is not restricted to being performed after the development phase is complete, but is carried parallel with all the stages of system development, starting with the requirement specification.

Test cases were devised with this purpose in mind. A test case is set of data that a system will process as normal input. The software units developed in the system are modules and routines that are assembled and integrated to performed the required function of the system. Test results, once gathered and evaluated, provide a qualitative indication of software quality and reliability and serve as a basis for design modification if required. In this phase, testing is done at different levels. Actually testing is the state of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operations commence.

Testing is the major quality control measure employed during software development. Testing is the process of executing a program with the intent of finding an error. No piece of code is completely ready unless it has been fully tested. This stage is very important as at this stage it is verified whether the code developed meet the requirement specifications or not. Moreover, all validations are also checked in the testing stage.

**NEED FOR TESTING**

Testing is vital to the success of the system. Testing makes a logical assumption that if all parts of the system are correct, the goal will be successfully achieved. Inadequate testing or non-testing leads to errors that may not appear until months later. A small system error can conceivably explode into a much larger problem.

Effective testing early in the process translates directly into long term cost saving from a reduced number of errors.

Second reason for testing is its utility as a user-oriented vehicle before implementation.

Finally, testing leads to software reliability, identifying and removing faults, during testing can make the software more reliable. This increases user’s confidence in the system.

**TESTING OBJECTIVES**

1. Testing is the process of executing a program with the intent of finding an error.
2. A good test case is one that has high profitability of finding an as yet discovered error.
3. A successful test is one that uncovers a yet undiscovered error.

**Selection of test cases:**

For testing to be successful, proper selection of test cases is essential. There are two different approaches to selecting cases – functional testing and structural testing.

**In functional testing** the software or the module to be tested is treated as a black ox, and the test cases are decided based on the specifications of the system or the module. For this reason this type of testing is also called “black box testing”. The focus here is on testing the external behavior of the system.

**In structural testing** the test cases are decided based on the logic of the module to be tested. A common approach here is to achieve some type of coverage of the statements in the code. One common criterion is statement coverage, which requires that test cases be selected so that together they execute each statement exactly once.

**Test Plan**

Testing is an extremely critical and time consuming activity. It requires proper planning of the overall testing process. Testing process starts with a test plan. This plan identifies all the testing related activities that must be preformed and specifies the schedule, allocates the resources, and specifies guidelines for testing. The test plan specifies conditions that should be tested, different units to be tested, and the manner in which the modules will be integrated together.

The basic levels of testing are:

1. Code testing
2. Unit testing
3. Integration testing
4. Validation testing
5. System testing
6. Output testing
7. User Acceptance testing

These different levels of testing attempt to detect different faults. The relation of the faults introduced in different phases and the different phases, and the different levels of testing are shown:

Client needs Acceptance testing

Requirement System testing

Design Integration testing

Code Unit testing

**Code testing**

This was done side by side with coding. This examines the logic of the program. Every path of the program was tested.

**Validation testing:**

At the culmination of the integration testing, software is completely assembled as a package; interfacing errors have been uncovered and corrected. Then as a final series of software test: validation tests were carried out. Validation testing can be defined in a several ways, but a simple definition is that validation succeeds when the software functions in a manner reasonably as expected by the user. Both client side as well as se4rver side validation test are performed.

**System testing:**

The next level of testing is system testing. Here the entire software system is testing. The reference document for this process is requirement document, and the goal is to see if the software meets its requirements. This is essentially a validation exercise.

**Unit testing:**

These different modules are tested against the specifications produced during design for the modules. Unit testing is essential for verification of the code produced during the coding phase and hence the goal is to test the internal logic of the modules.

**Integration testing:**

The next level of testing is often called integration testing. In this many tested modules are combined into sub-systems, which are then tested. The goal here is to see if the modules can be integrated properly, the emphasis being on testing interfaces between modules. This activity can be considered as testing the design, and hence the emphasis on testing module interactions.

**Output testing:**

The next step of performing testing is output testing of the proposed system since no system could be useful if it does not produce the output in the required format. Comparing the output with the format required by the user is output testing.

**User Acceptance testing:**

The last level testing is acceptance testing. Acceptance testing is performed with realistic data of the client to demonstrate that the software is working satisfactorily. Testing here focuses on the external behavior of the system; the internal logic of the program is not emphasized.

**SYSTEM IMPEMENTATION**

* Training the operating staff
* Installing hardware
* Installing terminals
* Installing telecommunication networks before system is up and running

In the implementation phase, the project reached its fruition. After the development phase of the software development life cycle (SDLC) is complete, the system is implemented. The software which was designed in design and programmed in development phase of the SDLC was installed on all the PCs that require it. The persons using the program were trained during this phase of SDLC. Moreover, both the hardware and software are tested. Although we found and fixed many programs, almost in variably, the users helped us to uncover problems that we were unable to simulate. These were the main activities performed by us in the course which leads to its proper completion.

**POST IMPLEMENTATION REVIEWS**

When computer based systems are build therefore we must develop mechanism for evaluating controlling and making modifications, maintenance issued to improve the case with which the changes can be accumulated and reduce the amount expended on its maintenance activity occurs because it is unreasonable to assume that software testing will uncover all latest errors in a large software system. The final event in the post implementation flow is review that revalidates all elements the system and ensures correctness. After the software maintenance, software reviews is being conducted for future maintenance, efforts and provides feedback, which is important to effectively management of software organization.

**CONCLUSION**

At the end of the project, I find that Office management system play a very important role in any office . A office must keep its data on the Computer. All the daily reports generated by the system are to be checked by the concerned official so as to ensure that all the data have been processed accurately. When I developed the project at last I come to know about the following fact that a computerized system over shines over the manual system. All the workings of the office are computerized and the computerized system helps in the work of office and makes it more easy, error free and understandable.

**Benefits Of The Office Management System**

The following are the benefits from the Office management system analysis :-

* All the reports are timely generated with accuracy.
* No computational error in the results.
* High Morale of the Office officer and subordinates while managing Office details of Employee.
* Fast retrieval and searching of data.
* No Paper work and calculations are required .
* Neat and Clean results.

**LIMITATION OF THE PROJECT**

In developing the software, every precaution has been taken but still the following are limitation :-

* There is no provision for backup of data from system failures.
* The output depends upon the accuracy of data input .

**FURTHER SCOPE OF THE PROJECT**

The following points will be included in the further scope of the project :-

* All the limitation discussed above can be removed.
* The system can be attached with Network

1. The information can be available on internet or office web site.

**BIBLIOGRAPHY**

**Books: -**

1. The Complete Reference Java 2

Herbert Schildt

Tata McGraw-Hill

1. Oracle

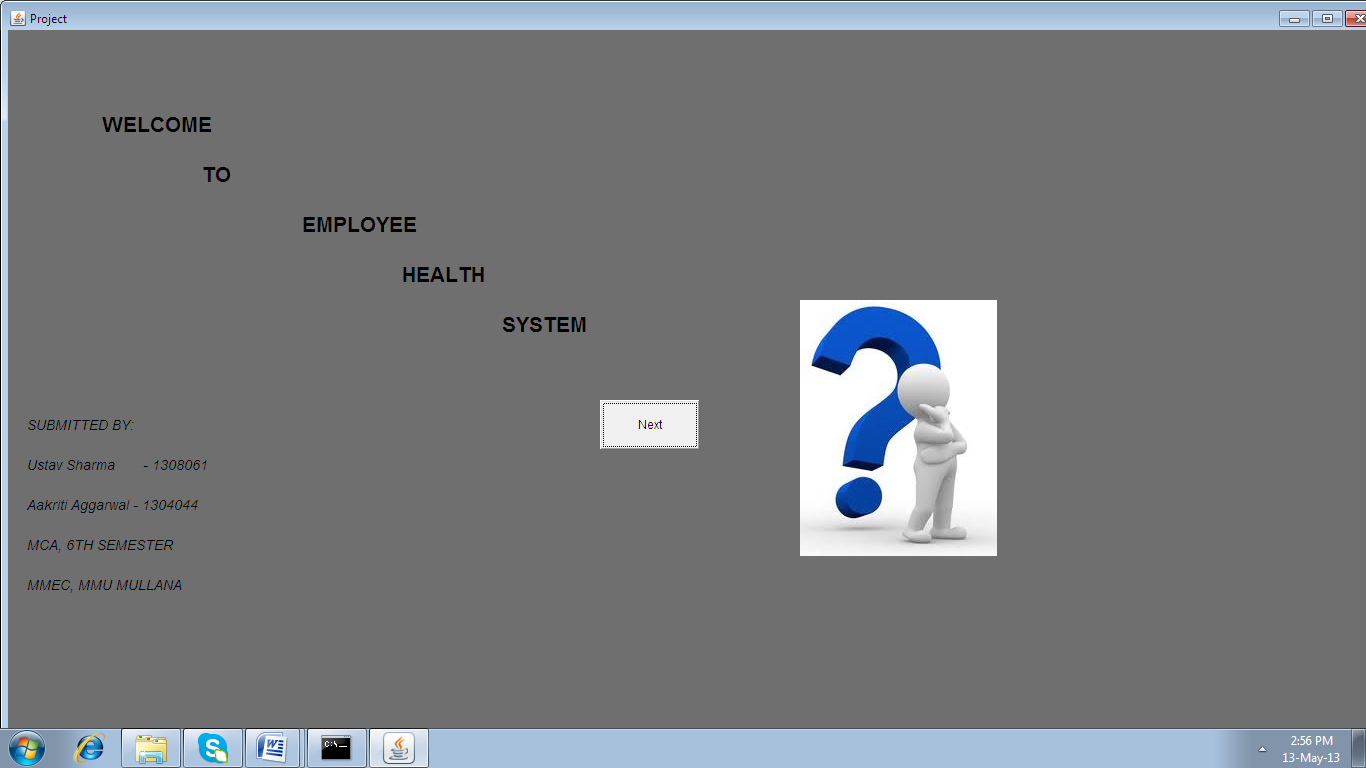
Evan Bayross

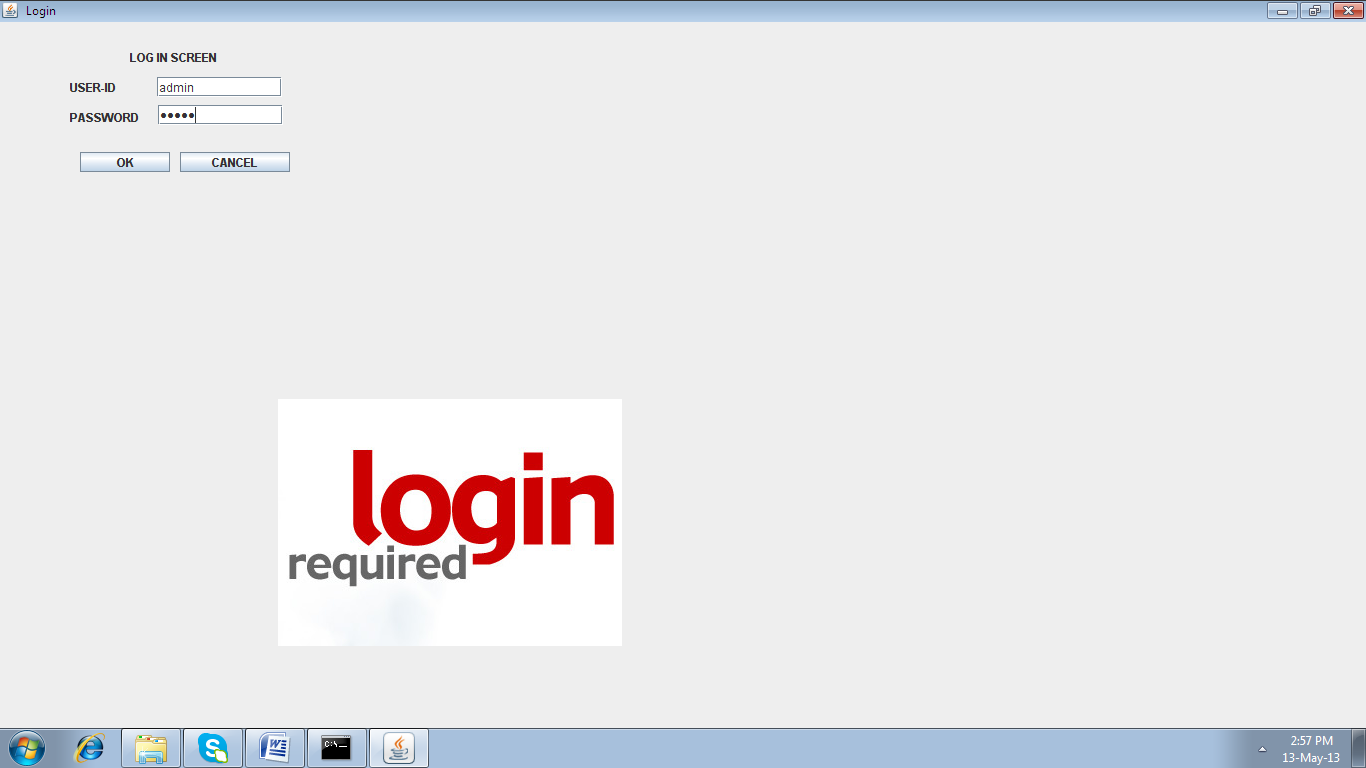
**Websites: -**

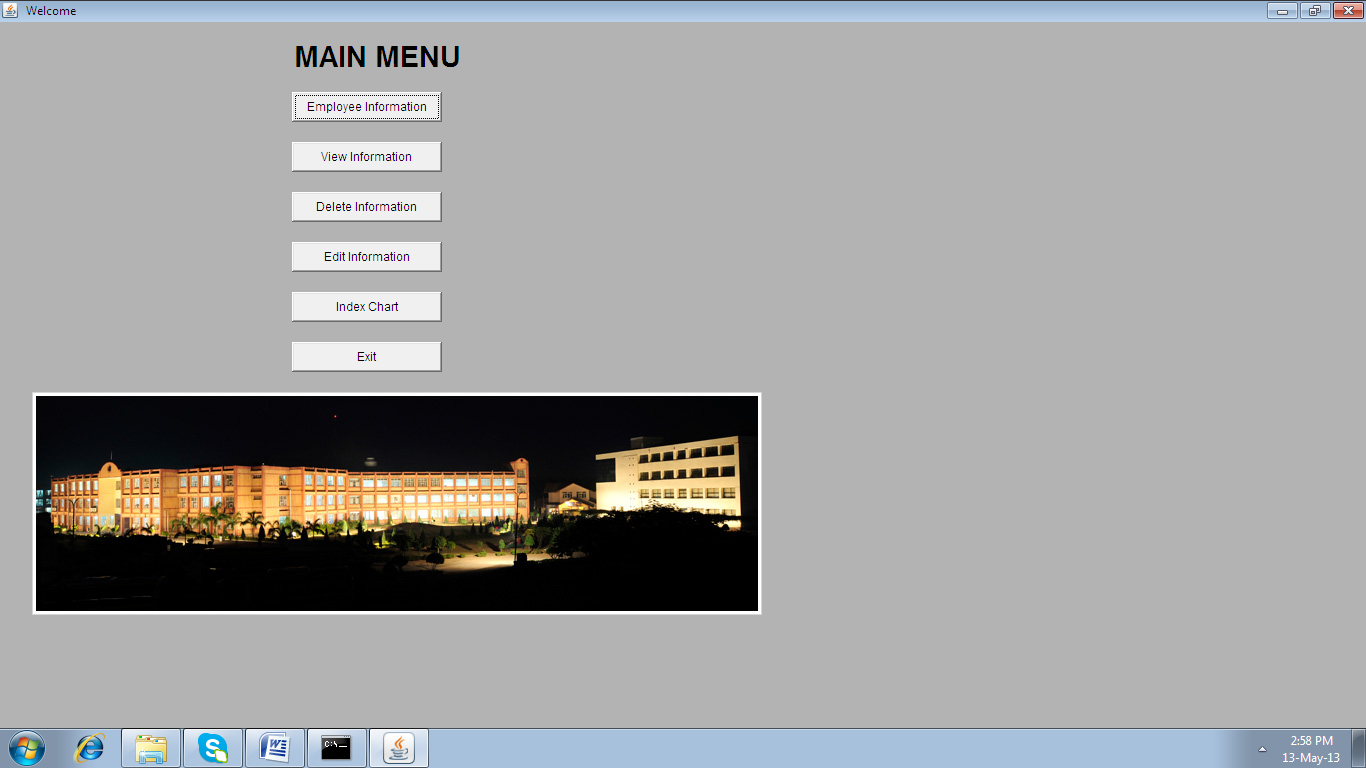
<http://www.java.net/sun>

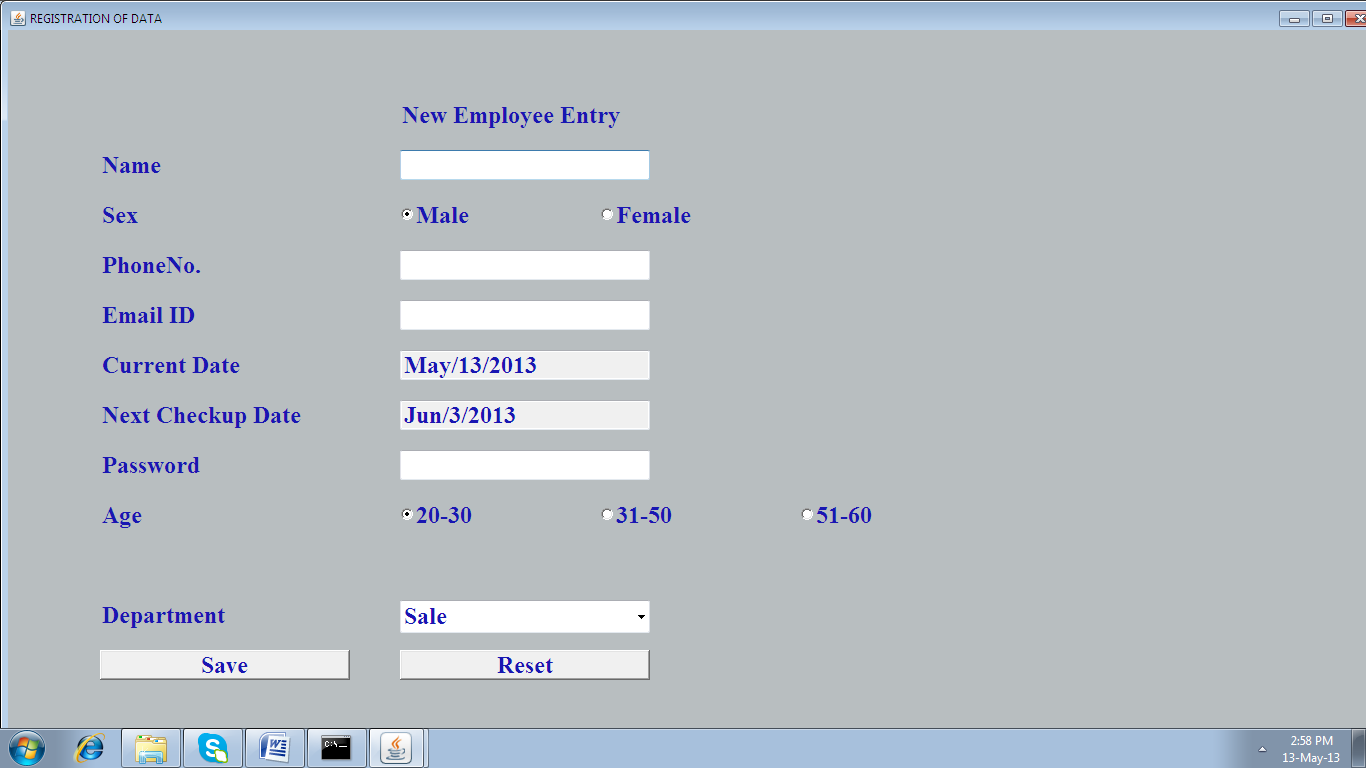
<http://www.google.co.in/>

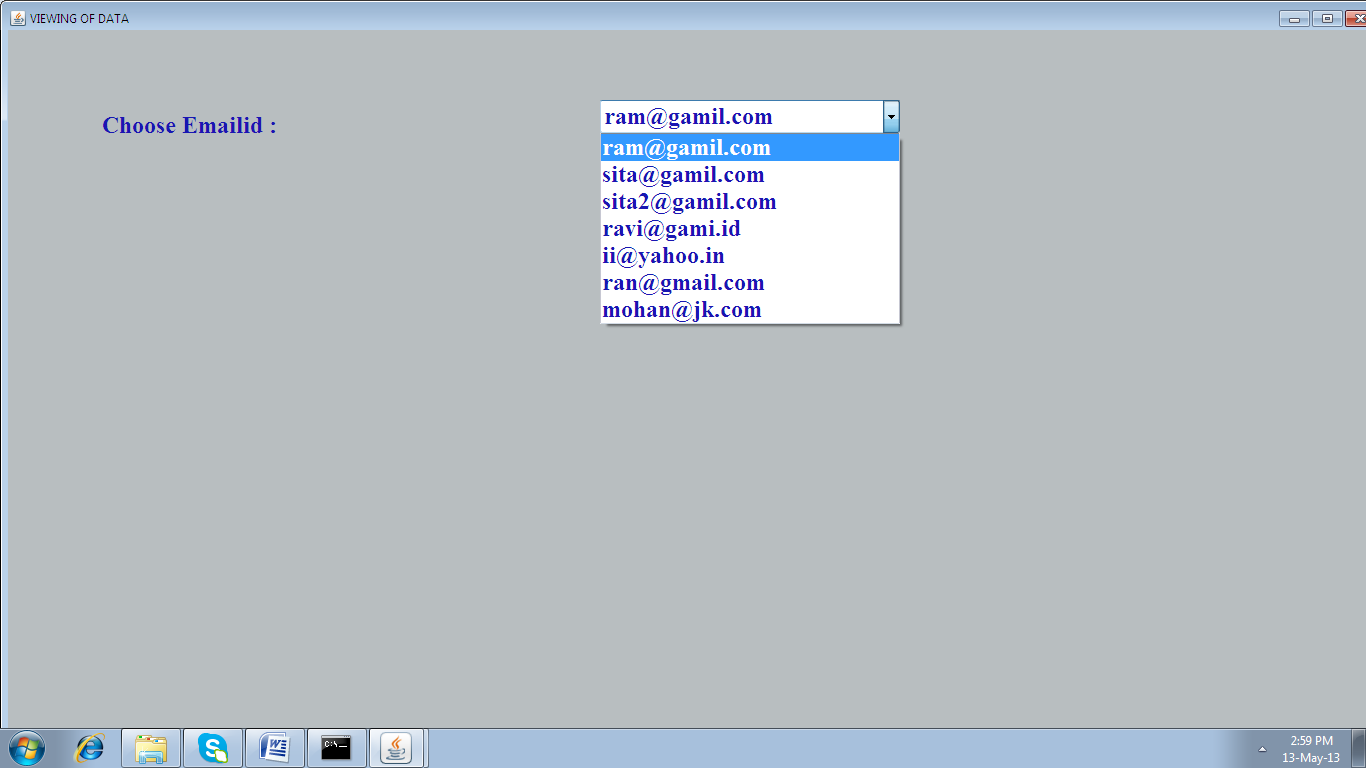
**SCREEN SHOTS**

****

****

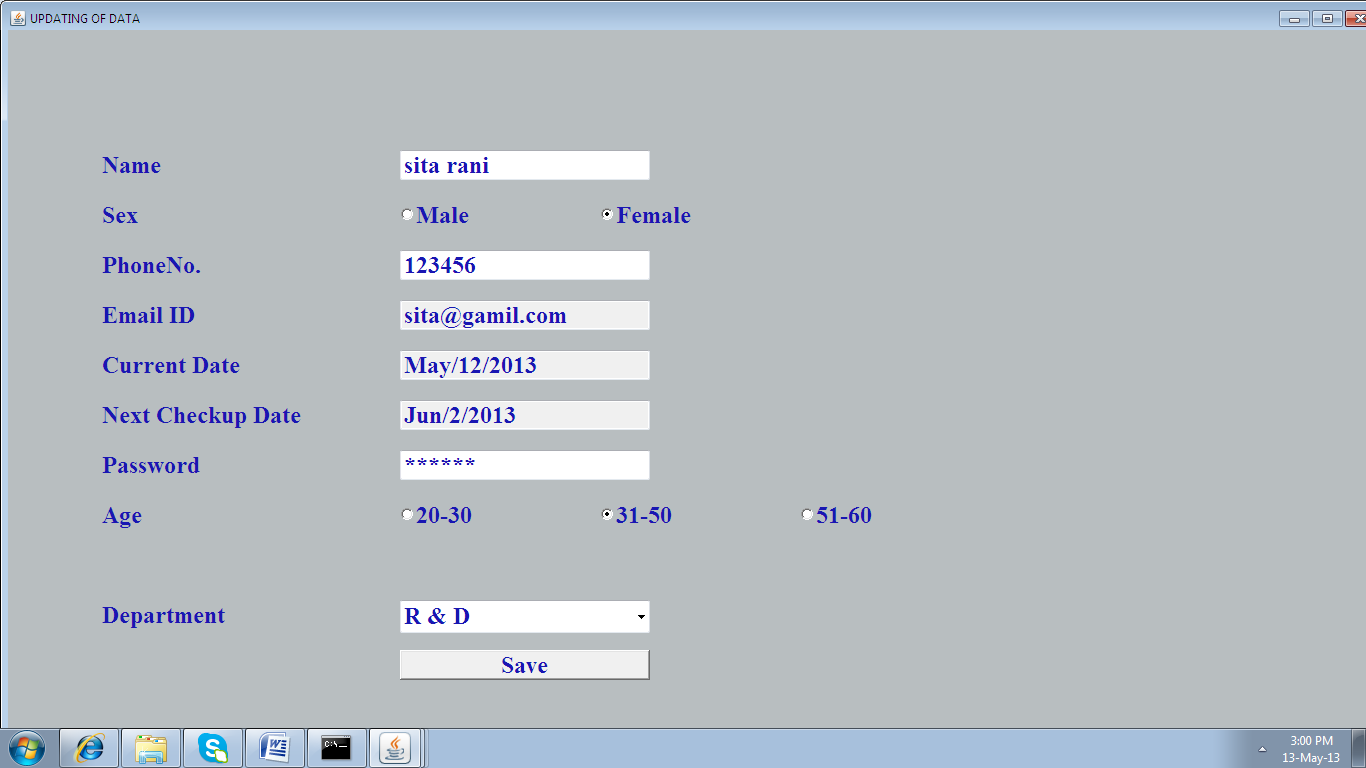
****

****

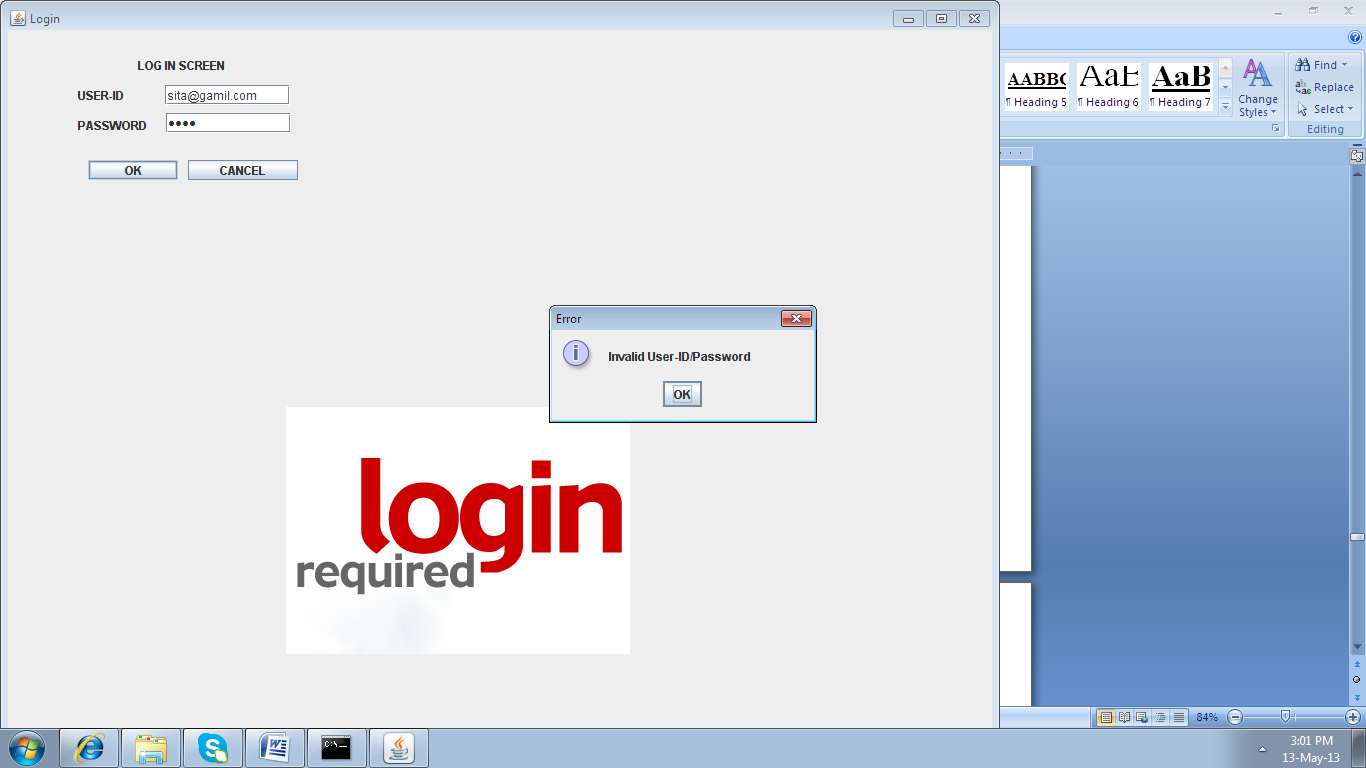
****

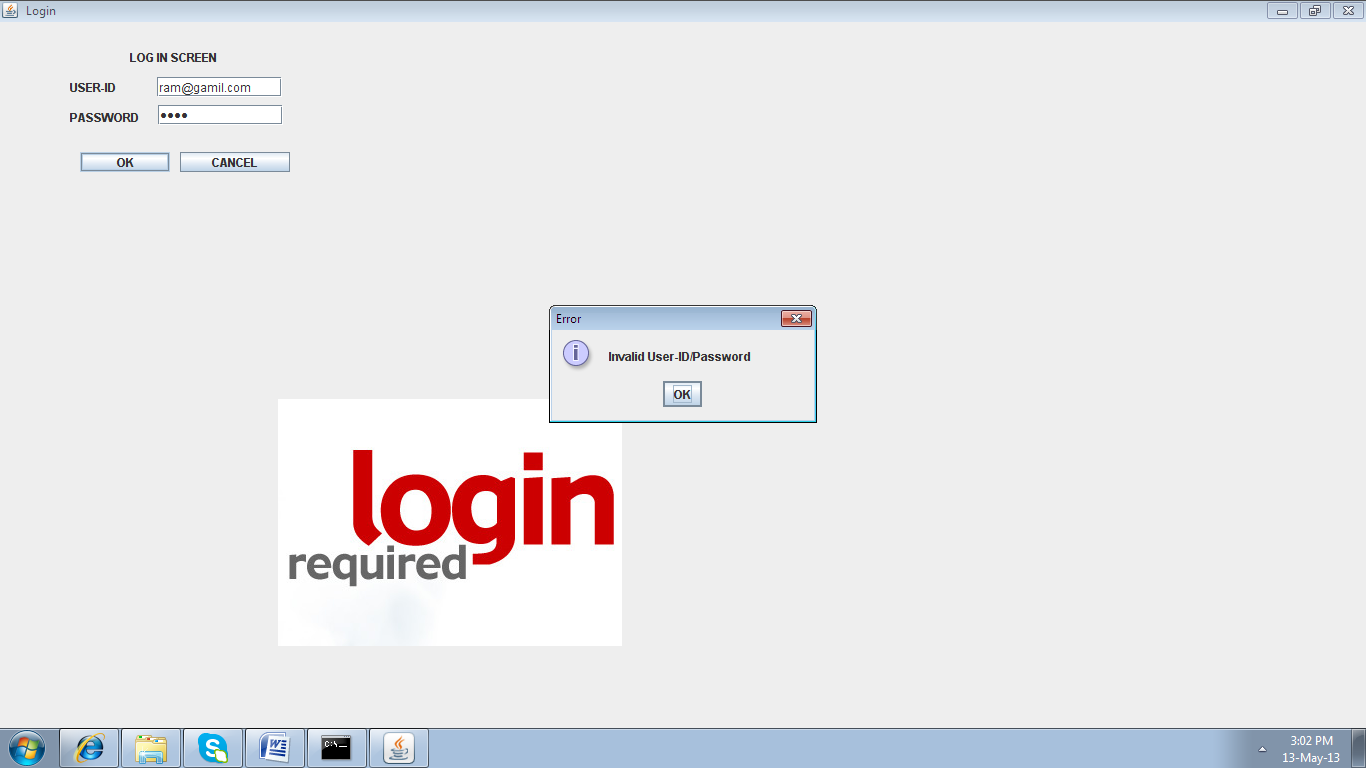
****

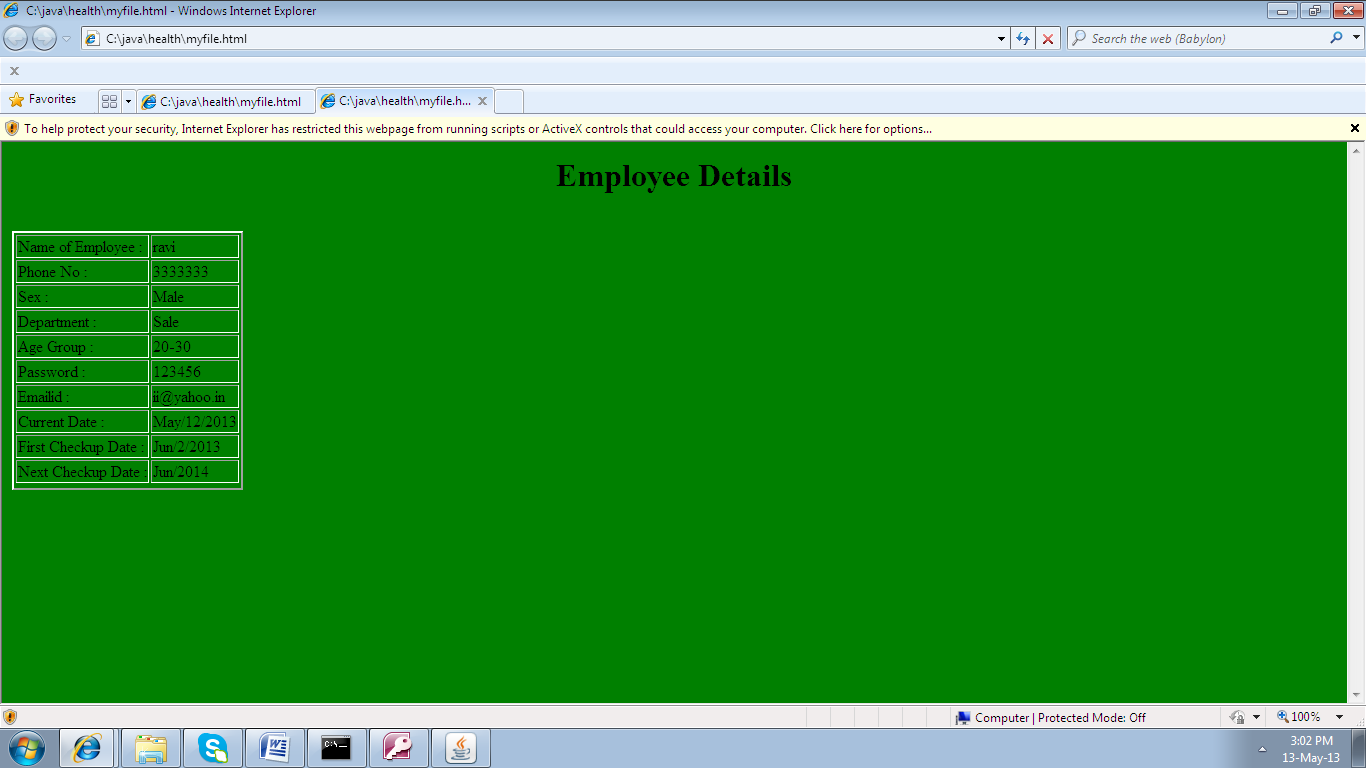
****

****

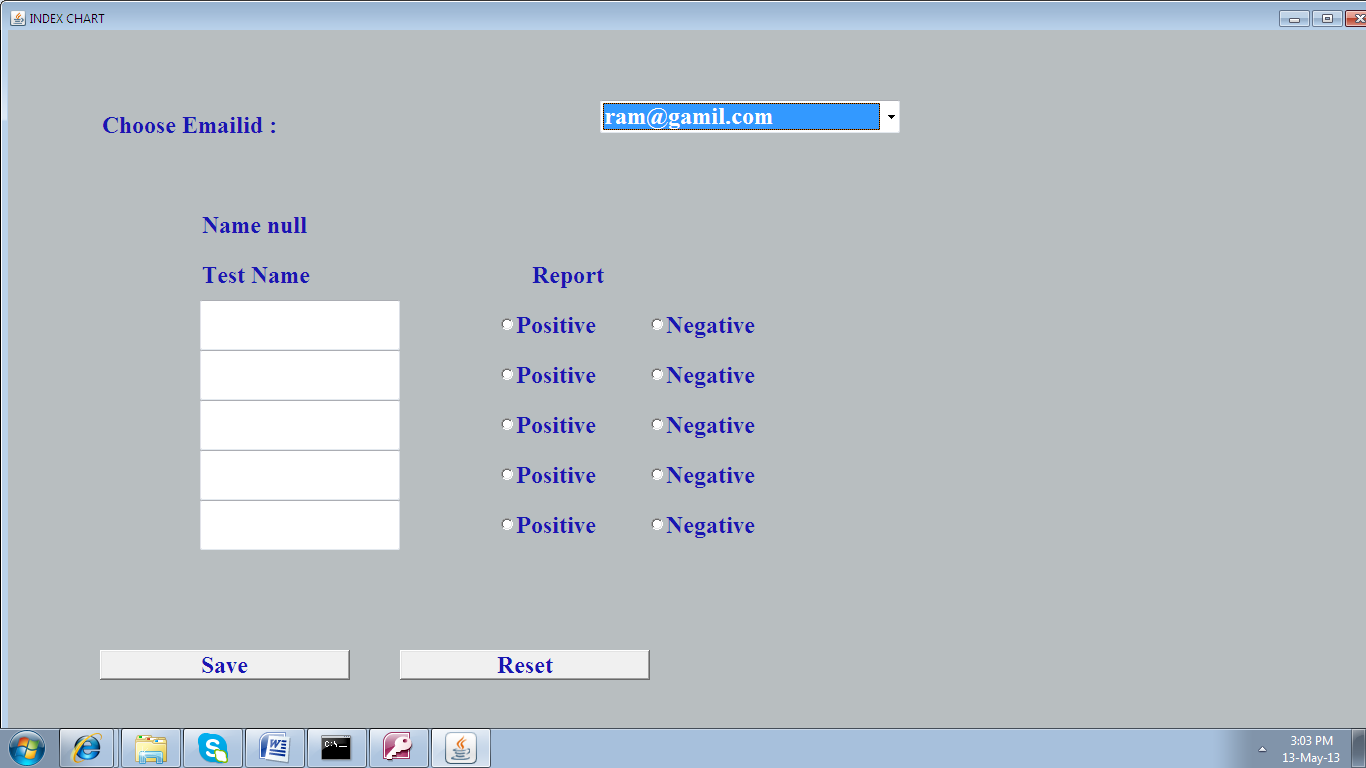
**USER LOGIN**

****

****

****

**INDEX CHART**

****

**SAMPLE SOURCE CODE**

**MAIN PROGRAM**

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

class mainmenu extends Frame implements ActionListener

{

private Closer Handler;

Label l1,l2,l3,l4,l5,l6,l7,l8,l9,l10;

Font f;

Button next;

public mainmenu()

{

super("Project");

}

public void init()

{

}

public void setup()

{

Handler=new Closer();

addWindowListener(Handler);

setLayout(null);

resize(1700,2000);

next= new Button("Next");

add(next);

next.addActionListener(this);

l1=new Label ("WELCOME");

l2=new Label ("TO");

l3=new Label ("EMPLOYEE");

l4=new Label ("HEALTH");

l5=new Label ("SYSTEM");

l6=new Label ("SUBMITTED BY:");

l7=new Label ("Ustav Sharma - 1308061");

l8=new Label ("Aakriti Aggarwal - 1304044");

l9=new Label ("MCA, 6TH SEMESTER");

l10=new Label ("MMEC, MMU MULLANA");

l1.setBounds(100,100,150,50);

l2.setBounds(200,150,100,50);

l3.setBounds(300,200,250,50);

l4.setBounds(400,250,300,50);

l5.setBounds(500,300,300,50);

l6.setBounds(25,400,400,50);

l7.setBounds(25,440,400,50);

l8.setBounds(25,480,400,50);

l9.setBounds(25,520,400,50);

l10.setBounds(25,560,500,50);

next.setBounds(600,400,100,50);

f=new Font("Comic Sans MS",Font.BOLD,21);

l1.setFont(f);

f=new Font("Comic Sans MS",Font.BOLD,21);

l2.setFont(f);

f=new Font("Comic Sans MS",Font.BOLD,21);

l3.setFont(f);

f=new Font("Comic Sans MS",Font.BOLD,21);

l4.setFont(f);

f=new Font("Comic Sans MS",Font.BOLD,21);

l5.setFont(f);

f=new Font("Comic Sans MS",Font.ITALIC,14);

l6.setFont(f);

l7.setFont(f);

l8.setFont(f);

l9.setFont(f);

l10.setFont(f);

l1.setForeground(Color.black);

l2.setForeground(Color.black);

l3.setForeground(Color.black);

l4.setForeground(Color.black);

l5.setForeground(Color.black);

l6.setForeground(Color.black);

l7.setForeground(Color.black);

l8.setForeground(Color.black);

l9.setForeground(Color.black);

l10.setForeground(Color.black);

add(l1);

add(l2);

add(l3);

add(l4);

add(l5);

add(l6);

add(l7);

add(l8);

add(l9);

add(l10);

show();

}

public void actionPerformed(ActionEvent ae)

{

String str=ae.getActionCommand();

if(str.equals("Next"))

{

setVisible(false);

Password x=new Password();

}

repaint();

}

public void paint(Graphics g)

{

Color c1=new Color(111,111,111);

g.setColor(c1);

setBackground(c1);

l1.setBackground(c1);

l2.setBackground(c1);

l3.setBackground(c1);

l4.setBackground(c1);

l5.setBackground(c1);

l6.setBackground(c1);

l7.setBackground(c1);

l8.setBackground(c1);

l9.setBackground(c1);

l10.setBackground(c1);

Image pic=Toolkit.getDefaultToolkit().getImage("1.jpg");

if(pic!=null)

g.drawImage(pic,800,300,this);

}

public static void main(String args[])

{

mainmenu A=new mainmenu();

A.setup();

}

}

class Closer extends WindowAdapter

{

public void windowClosing(WindowEvent we)

{

System.exit(0);

}

}

import java.awt.Desktop;

import java.io.IOException;

import java.net.URI;

import java.net.URISyntaxException;

import java.net.URL;

import java.net.URLConnection;

import java.net.URLEncoder;

import java.net.HttpURLConnection;

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

import java.sql.\*;

import java.io.\*;

import java.util.Calendar;

import javax.swing.\*;

public class registration extends Frame implements ActionListener,ItemListener

{

private Closer Handler;

String k[] = {"Jan", "Feb", "Mar", "Apr","May","Jun","July","Aug","Sep","Oct","Nov","Dec"};

Calendar kp=Calendar.getInstance();

Button Submit,Reset;

TextField name,phoneno,emailid,cdate,ndate,pass;

Label lname,lphone,lemail,ldept,lcdate,lndate,lpass,age,n1,lsex,lage;

Checkbox male,female,age1,age2,age3;

CheckboxGroup cbg,cbg2;

Choice dept;

public registration()

{

super("REGISTRATION OF DATA");

}

public void setup()

{

Handler=new Closer();

addWindowListener(Handler);

Font f1=new Font ("TimesRoman",Font.BOLD,24);

setFont (f1);

resize(1500,800);

setLayout(null);

Color c1=new Color(184,190,192);

setBackground(c1);

Color c2=new Color(25,22,178);

setForeground(c2);

cbg=new CheckboxGroup();

cbg2=new CheckboxGroup();

male=new Checkbox("Male",cbg,true);

female=new Checkbox("Female",cbg,false);

age1=new Checkbox("20-30",cbg2,true);

age2=new Checkbox("31-50",cbg2,false);

age3=new Checkbox("51-60",cbg2,false);

n1=new Label("New Employee Entry");

dept=new Choice();

lname=new Label("Name");

lphone=new Label("PhoneNo.");

lemail=new Label("Email ID");

lpass=new Label("Password");

lcdate=new Label("Current Date");

lsex=new Label("Sex");

lage=new Label("Age");

lndate=new Label("Next Checkup Date");

ldept=new Label("Department");

dept.add("Sale");

dept.add("Marketing");

dept.add("Finance");

dept.add("Production");

dept.add("Personnel");

dept.add("R & D");

name=new TextField(20);

phoneno=new TextField(20);

emailid=new TextField(20);

cdate=new TextField(20);

ndate=new TextField(20);

pass=new TextField(20);

pass.setEchoChar('\*');

add(name);

add(phoneno);

add(emailid);

add(cdate);

add(ndate);

add(pass);

add(n1);

add(lphone);

add(lemail);

add(lcdate);

add(lndate);

add(lpass);

add(lname);

add(lsex);

add(lage);

add(dept);

add(ldept);

n1.setBounds(400,100,250,30);

lname.setBounds(100,150,250,30);

lsex.setBounds(100,200,50,30);

lphone.setBounds(100,250,250,30);

lemail.setBounds(100,300,250,30);

lcdate.setBounds(100,350,250,30);

lndate.setBounds(100,400,250,30);

lpass.setBounds(100,450,250,30);

lage.setBounds(100,500,250,30);

ldept.setBounds(100,600,250,30);

dept.setBounds(400,600,250,50);

name.setBounds(400,150,250,30);

male.setBounds(400,200,150,30);

female.setBounds(600,200,250,30);

phoneno.setBounds(400,250,250,30);

emailid.setBounds(400,300,250,30);

cdate.setBounds(400,350,250,30);

ndate.setBounds(400,400,250,30);

pass.setBounds(400,450,250,30);

age1.setBounds(400,500,200,30);

age2.setBounds(600,500,200,30);

age3.setBounds(800,500,200,30);

add(male);

add(female);

add(age1);

add(age2);

add(age3);

Submit=new Button("Save");

Reset=new Button("Reset");

add(Submit);

add(Reset);

Submit.setBounds(100,650,250,30);

Reset.setBounds(400,650,250,30);

Submit.addActionListener(this);

Reset.addActionListener(this);

String p0,p3;

int p2,p11;

p0=k[kp.get(Calendar.MONTH)];

p11= kp.get(Calendar.DATE);

p2=kp.get(Calendar.YEAR);

p3=p0+"/"+p11+"/"+p2;

cdate.setText(p3);

cdate.setEditable(false);

p11= kp.get(Calendar.DATE)+20;

if(p11>30)

{

p11=p11-30;

p0=k[kp.get(Calendar.MONTH)+1];

}

p3=p0+"/"+p11+"/"+p2;

ndate.setText(p3);

ndate.setEditable(false);

show();

}

public void actionPerformed(ActionEvent ae)

{

if(ae.getSource()==Reset)

{

name.setText("");

phoneno.setText("");

pass.setText("");

emailid.setText("");

//cdate.setText("");

//ndate.setText("");

Submit.setVisible(true);

}

if(ae.getSource()==Submit)

{

String s1,s2,s3,s4,s5,s6,s9,s7,s8;

s5=cdate.getText();

s6=ndate.getText();

s9=dept.getSelectedItem();

s7=cbg.getSelectedCheckbox().getLabel();

s8=cbg2.getSelectedCheckbox().getLabel();

long s11;

s2=name.getText();

if(s2.length()<3)

{

JOptionPane.showMessageDialog(null,"Enter the Employee Name");

return;

}

s3=pass.getText();

if(s3.length()<3)

{

JOptionPane.showMessageDialog(null,"Enter the Password");

return;

}

s4=emailid.getText();

if(s4.length()<3)

{

JOptionPane.showMessageDialog(null,"Enter the Employee Emailid");

return;

}

int p1,p2,p3,p4;

p1=s4.indexOf("@");

p2=s4.indexOf(".");

if((p1==-1)||(p2==-1))

{

JOptionPane.showMessageDialog(null,"Enter the Correct Employee Emailid");

return;

}

p3=s4.lastIndexOf("@");

p4=s4.lastIndexOf(".");

if((p1!=p3)||(p2!=p4))

{

JOptionPane.showMessageDialog(null,"Enter the Correct Employee Emailid");

return;

}

s1=phoneno.getText();

if(s1.length()<6)

{

JOptionPane.showMessageDialog(null,"Enter the Phone No");

return;

}

try

{

s11=Integer.parseInt(s1);

}

catch(Exception e1)

{

JOptionPane.showMessageDialog(null,"Enter the Phone No in digits");

phoneno.setText("");

return;

}

///////////

final String fileName = "C://java//health//master.mdb";

//final String fileName = "it.accdb";

Connection con = null;

try {

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

String url = "jdbc:odbc:Driver={Microsoft Access Driver (\*.mdb, \*.accdb)};DBQ="+fileName;

con = DriverManager.getConnection(url,"","");

Statement st=con.createStatement();

ResultSet rs=st.executeQuery("select \*from table1 where emailid='"+s4+"'");

int i=0;

while(rs.next())

{

i=1;

}

if(i==1)

{

JOptionPane.showMessageDialog(null,"Emailid Already Exsits");

return;

}

rs.close();

st.executeUpdate("insert into table1(ename,phoneno,pass,emailid,cdate,ndate,sex,agegroup,dept) values ('"+s2+"','"+s1+"','"+s3+"','"+s4+"','"+s5+"','"+s6+"','"+s7+"','"+s8+"','"+s9+"')");

JOptionPane.showMessageDialog(null,"Date Submited Successfully");

Submit.setVisible(false);

st.close();

}

catch(Exception e)

{

System.out.println(e+"1");

}

//////

FileOutputStream out; // declare a file output object

PrintStream ps1; // declare a print stream object

try

{

out = new FileOutputStream("myfile.html");

ps1 = new PrintStream( out );

ps1.println ("<body bgcolor='GREEN' text-color='red'>");

ps1.println ("<h1><center>Employee Details</h1></center>");

ps1.println("<br>");

ps1.println("<table border='2'>");

ps1.println("<b><tr><h3>");

ps1.print("<td>Name of Employee : </td><td>"+s2+"</td>");

ps1.println("</tr><tr><h3>");

ps1.print("<td>Phone No :</td><td> "+s1+"</td>");

ps1.println("</tr><tr><h3>");

ps1.print("<td>Sex :</td> <td>"+s7+"</td>");

ps1.println("</tr><tr><h3>");

ps1.print("<td>Department : </td><td>"+s9+"</td>");

ps1.println("</tr><tr><h3>");

ps1.print("<td>Age Group :</td><td> "+s8+"</td>");

ps1.println("</tr><tr><h3>");

ps1.print("<td>Password :</td><td> "+s3+"</td>");

ps1.println("</tr><tr><h3>");

ps1.print("<td>Emailid : </td><td>"+s4+"</td>");

ps1.println("</tr><tr><h3>");

ps1.print("<td>Current Date : </td><td>"+s5+"</td>");

ps1.println("</tr><tr><h3>");

ps1.print("<td>First Checkup Date : </td><td>"+s6+"</td>");

ps1.println("</tr><tr><h3>");

}

catch(Exception e)

{

}

try

{

URI uri = new URI("file:///C:/java/health/myfile.html");

uri.normalize();

Desktop.getDesktop().browse(uri);

}

catch (Exception e ) {

}

}

}

public void itemStateChanged(ItemEvent ie)

{

}

public void paint(Graphics g)

{

}

public static void main(String args[])

{

registration r1=new registration();

r1.setup();

}

}

import java.awt.Desktop;

import java.io.IOException;

import java.net.URI;

import java.net.URISyntaxException;

import java.net.URL;

import java.net.URLConnection;

import java.net.URLEncoder;

import java.net.HttpURLConnection;

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

import java.sql.\*;

import java.io.\*;

import java.util.Calendar;

import javax.swing.\*;

public class dele extends Frame implements ActionListener,ItemListener

{

String s1,s2,s3,s4,s5,s6,s9,s7,s8;

int f=0;

private Closer Handler;

Button Submit,Reset;

Label l1;

Choice dept;

public dele()

{

super("DELETING OF DATA");

}

public void setup()

{

Handler=new Closer();

addWindowListener(Handler);

Font f1=new Font ("TimesRoman",Font.BOLD,24);

setFont (f1);

l1=new Label("Choose Emailid :");

resize(1500,800);

setLayout(null);

Color c1=new Color(184,190,192);

setBackground(c1);

Color c2=new Color(25,22,178);

setForeground(c2);

final String fileName = "C://java//health//master.mdb";

//final String fileName = "it.accdb";

Connection con = null;

dept=new Choice();

try {

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

String url = "jdbc:odbc:Driver={Microsoft Access Driver (\*.mdb, \*.accdb)};DBQ="+fileName;

con = DriverManager.getConnection(url,"","");

Statement st=con.createStatement();

ResultSet rs=st.executeQuery("select emailid from table1");

int i=0;

while(rs.next())

{

f=1;

dept.add(rs.getString(1));

i=1;

}

if(i==0)

{

JOptionPane.showMessageDialog(null,"No Data Exsits");

return;

}

rs.close();

}

catch(Exception e)

{

System.out.println(e+"1");

}

add(dept);

add(l1);

l1.setBounds(100,100,400,50);

dept.setBounds(600,100,300,50);

Submit=new Button("DELETE");

Reset=new Button("Reset");

add(Submit);

//add(Reset);

Submit.setBounds(100,650,250,30);

Reset.setBounds(400,650,250,30);

Submit.addActionListener(this);

Reset.addActionListener(this);

dept.addItemListener(this);

show();

}

public void actionPerformed(ActionEvent ae)

{

if(ae.getSource()==Submit)

{

s9=dept.getSelectedItem();

////////

final String fileName = "C://java//health//master.mdb";

//final String fileName = "it.accdb";

Connection con = null;

try {

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

String url = "jdbc:odbc:Driver={Microsoft Access Driver (\*.mdb, \*.accdb)};DBQ="+fileName;

con = DriverManager.getConnection(url,"","");

Statement st=con.createStatement();

st.executeUpdate("Delete from table1 where emailid='"+s9+"'");

s1="";

s2="";

s3="";

s4="";

s5="";

s6="";

s7="";

s8="";

s9="";

repaint();

}

catch(Exception e)

{

System.out.println(e+"1");

}

}

}

public void itemStateChanged(ItemEvent ie)

{

s1="";

s2="";

s3="";

s4="";

s5="";

s6="";

s7="";

s8="";

s9="";

f=1;

s9=dept.getSelectedItem();

////////

final String fileName = "C://java//health//master.mdb";

//final String fileName = "it.accdb";

Connection con = null;

try {

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

String url = "jdbc:odbc:Driver={Microsoft Access Driver (\*.mdb, \*.accdb)};DBQ="+fileName;

con = DriverManager.getConnection(url,"","");

Statement st=con.createStatement();

ResultSet rs=st.executeQuery("select \* from table1 where emailid='"+s9+"'");

int i=0;

while(rs.next())

{

s2=rs.getString(1);

s7=rs.getString(2);

s1=rs.getString(3);

s4=rs.getString(4);

s5=rs.getString(5);

s6=rs.getString(6);

s3=rs.getString(7);

s8=rs.getString(8);

s9=rs.getString(9);

i=1;

}

if(i==0)

{

JOptionPane.showMessageDialog(null,"No Data Exsits");

return;

}

rs.close();

}

catch(Exception e)

{

System.out.println(e+"1");

}

/////

repaint();

}

public void paint(Graphics g)

{

if((f==1)&&(s1.length()>0))

{

g.drawString("Name of Employee : "+s1,200,200);

g.drawString("Phone No :"+s2,200,250);

g.drawString("Sex :"+s7,200,300);

g.drawString("Department : "+s9,200,350);

g.drawString("Age Group :"+s8,200,400);

g.drawString("Password :"+s3,200,450);

g.drawString("Emailid : "+s4,200,500);

g.drawString("Current Date : "+s5,200,550);

g.drawString("First Checkup Date : "+s6,200,600);

}

}

public static void main(String args[])

{

dele r1=new dele();

r1.setup();

}

}