JOB PORTAL SYSTEM

A PROJECT REPORT

Submitted in Partial Fulfillment for the degree of Bachelor of Technology in

SWAMI VIVEKANADA INSTITUTE OF SCIENCE AND TECHNOLOGY

From WEST BENGAL UNIVERSITY OF TECHNOLOGY

Submitted by

BISWAJIT SAMANTA

PRITAM BERA

SOURAV SIKARI

PRANTIK MITRA

RAJARSHI HAZRA

Under the guidance of

Prof. Mrs. Sulekha Kundu

Mr. Subhendu Sekhar Patra

Assistant Professor of CSE Dept.

National Highway 6, Banitabla, Uluberia, Howrah,

West Bengal-711316

July, 2021

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

SWAMI VIVEKANADA INSTITUTE OF SCIENCE AND TECHNOLOGY

Affiliated to

MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY (MAKAUT)

MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY (MAKAUT)

BF Block, Sector 1, Bidhannagar, Kolkata, West Bengal 700064

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE OF APPROVAL

THIS IS TO CERTIFY THAT **Biswajit Samanta (24107018045)**, **Pritam Bera (24107018032)**, **Sourav Sikari (24107018017)**, **Prantik Mitra (24107018033), Rajarshi Hazra (24107018031)** HAS COMPLETED HIS FINAL YEAR PROJECT OF BACHELOR OF TECHNOLOGY (ELECTRONICS & COMMUNICATION ENGINEERING) ENTITLED " **JOB PORTAL SYSTEM**".THAY HAVE DONE THE PROJECT WORK UNDER THE SUPERVISION OF Mrs. SULEKHA KUNDU, Mr. SUBHENDU SEKHAR PATRA Assistant Professor OF CALCUTTA INSTITUTE OF TECHNOLOGY, ULUBERIA.
WE ARE SATISFIED WITH HIS WORK, WHICH IS BEING PRESENTED FOR THE PARTIAL FULFILLMENT FOR THE DEGREE OF BACHELOR OF TECHNOLOGY (ELECTRONICS & COMMUNICATION ENGINEERING), MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY (FORMARLY KNOWN AS WEST BENGAL UNIVERSITY OF TECHNOLOGY), SALT LAKE, KOLKATA-700064.

Prof. (Dr.) Priyabrota Sarkar
Principal-In-Charge,
Calcutta Institute of Technology
Uluberia, Howrah-711316

(Mrs. Sulekha Kundu)
Assistant Professor,
Department of Computer Science and Engineering
Calcutta Institute of Technology
Uluberia, Howrah-711316

Krishna Kumar Jha
Assistant Professor, Teacher-In-Charge
Department of Computer Science and Engineering
Calcutta Institute of Technology
Uluberia, Howrah-711316

(Mr. Subhendu Sekhar Patra)
Assistant Professor,
Department of Computer Science and Engineering
Calcutta Institute of Technology
Uluberia, Howrah-711316

ACKNOWLEDGEMENT

Success of any project depends largely on the encouragement and guidelines of many others. We take this sincere opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project work.

I am highly indebted to for his guidance and constant supervision as well as for providing necessary information regarding the project & also for his support in completing the project.

I would like to express my gratitude towards my parents & our University Teachers for their kind co-operation and encouragement which help me in completion of this project.

My thanks and appreciations also go to my friends in developing the project and people who have willingly helped me out with their abilities.

Words are inadequate in offering our thanks to the other trainees and other members at Calcutta Institute of Technology for their encouragement and cooperation in carrying out this project work. The guidance and support received from all the members and who are contributing to this project, was vital for the success of this project.

TABLE OF CONTENTS

JOB PORTAL SYSTEM

1. INTRODUCTION

This project is aimed at developing a search Portal for job seeker. The system is a application that can be accessed throughout the organization and outside as well with proper login provided. This system can be used as a Job Portal for job seekers. Job Seekers logging should be able to upload their information in the form of a CV. Visitors/Company representatives logging in may also access/search any information put up by Job aspirants.

Today there is no place for errors, so as to make a system more effective and efficient we need such technology where error prone chances must be least. In the scenario of this project, we are developed a web-based application on Job Portal. In this time of recession where everyone, either experienced or fresher, is in search for a job. This job portal can prove to be very helpful since it allows users of a different profile to upload their CVs, search job on the basis of their qualification.

2.OBJECTIVE

This system can be used as a Job Portal System for the Placements providing to the unemployed who are seeking for a job placement. Job Seeker logging into the system and he can should be able to upload their information in the form of a CV. Visitors/Company representatives logging in may also access/search any information put up by Job Seeker. The system provides job catalogue and information to the members to helps them decide on the jobs to apply. The admin and employers can keep the jobs catalogue updated all the time so that the job seekers get the updated information all the time.

3. SCOPE

The future of the job portals looks bright. Job portals are like the meeting points for the recruiters as well as job seekers. Most of the jobseekers who're actively seeking new employment opportunities are believed to be registered on multiple job portals. Finding a job opportunity per your choice and qualification through a job portal is relatively easier. But the sourcing and operation process often takes a long time. Many base an employment agency to expedite this process.

SYSTEM ANALYSIS

4. IDENTIFICATION OF NEED

Once the project is initiated it is important to understand the functioning and structure of the currently operating manually driven order processing system. In order to identify the problems that plagued the performance of trading firms the following things have to be briefly scanned

1. The performance of the system.

2. The kind of information being processed.

3. The cost incurred in maintaining the operation of the Job Portal System,

4. Central Administrator in managing the operation in totality. Level of confidentiality of information in the present set up. Since a particular decision is made in the context of certain given objectives, identification of the problem is the real beginning of decision-making process. A problem is a felt need a question thrown forward for solution. The category of information that this project required are noted as follows: -

a) Information about the Jobseeker's as well as Employer's' feedback in order to maintain the objectives and organization structure.

b) Information about the staff that is to ascertain the skill of the people, who are working in the present system their job functions and information requirement, the relationship of their jobs to the existing system and the interpersonal network that holds the user group together.

Analysis:

While the feedback of the problem gives the understanding of what should be done analysis of problem takes it a step further. The whole approach of analysis of problem should however be based around critical factors like the availability of information for making the decision, the time available for servicing complain

1. Planned approach towards working:

The working in the organization will be well planned and organized. The data will be stored properly in data stores, which will help in retrieval of information as well as its storage

2.Accuracy:

The level of accuracy in the proposed system will be higher. All operation would be done correctly and it ensures that whatever information is coming from the center is accurate.

3.Reliability:

The reliability of the proposed system will be high due to the above stated reasons. The reason for the increased reliability of the system is that now there would be proper storage of information.

4.No Redundancy:

In the proposed system utmost care would be that no information is repeated anywhere, in storage or otherwise. This would assure economic use of storage space and consistency in the data stored.

5.Immediate retrieval of information:

The main objective of proposed system is to provide for a quick and efficient retrieval of information. Any type of information would be available whenever the user requires.

6.Immediate storage of information:

In manual system there are many problems to store the largest amount of information.

7.Easy to Operate:

The system should be easy to operate and should be such that it can be developed within a short period of time and fit in the limited budget of the user.

8. Economy:

In the manual operation the cost of manpower, paperwork associated with each customer is quite high when compared to computer based inventory management System. Moreover, tracking and monitoring the cost of inputs can be determined efficiently and whether the inventory management is performing inconsonance with the objective of the Annual Maintenance Company.

9. Timeliness:

Timeliness relates more to the transmission of information than to the processing and storing of it. Most manual systems suffer from the problem of timeliness as information cannot be retrieved when it is needed.

10. Throughput:

Presently the manually operated recruiting System are very much slow in producing relevant output and hence there is a need for development of an Automated Job portal System which can increase the rate of throughput manifold.

11. Control of information processing:

The manually driven Skilled people Hiring System prevalent until now has some redundant control mechanism like duplication of the same task, unnecessary documentation for internal check, etc.

12. Security:

The manual Skilled people Hiring System takes precious little effort in maintaining the security of the information. Vital information can be leaked with ease and consequent fixing of accountability to the erring staff sometimes become difficult.

13. Integration of business areas:

Coordination of business activities through capture and distribution of information is of prime importance and the present setup finds it extremely difficult to operate with the growing complexities and therefore the need is to evolve a computerized system.

5. FUNCTIONAL REQUIREMENTS

Modules:

The modules used in this software are as follows:

- **Registration:** This page contains the customer name, email id, contact number, address, password.

- **Login:** This module is for registered customer to login. The **ADMIN** has the authority to Add, Delete, and Update etc. The **jobseeker** can search job,Apply For the job etc.

- **Home:** This page contains the site details and information about the site i.e. the Online Job Portal.
- **Jobseeker Index:** This page contains the details about the details of Jobseeker, details of jobs, details of requirement of skill.

- **Admin Index:** This section is for admin. Admin can **INSERT, UPDATE, DELETE** and **SEARCH** in any section like, in **Company, Job, Jobseeker's information** etc.

6. NON-FUNCTIONAL REQUIREMENTS

- **Usability Requirement**: The system shall allow the users to access the system from any browsers, no special training is required. The system user friendly and the system is written in simple English.

- **Availability Requirement**: The system is available 100% for the user and is used by 24 hours a day and 365 days a year. The system shall be operational 24 hours a day and 7 days a week.

- **Accuracy**: The system should accurately provide real time information taking into consideration various issues. The system shall provide 100% access reliability.

- **Performance Requirement**: The information is refreshed at regular intervals depending upon whether some updates have occurred or not. The system shall respond the member in less than 2 seconds.

- **Security Requirement**: System will use a secured database and the system will have different users and each user has different types of constraints. Only admin have the rights to update database information of other users.

7. HARDWARE & SOFTWARE REQUIREMENTS

HARDWARE REQUIREMENTS

- Computer that has a 1GHz or faster processor
- 4GB or more RAM
- HD 20 GB Hard Disk Space

SOFTWARE REQUIREMENTS

- WINDOWS 10
- JAVA (JDK 1.8)
- MYSQL
- APACHE NETBEANS IDE 12.0

SYSTEM DESIGN

1. DATA FLOW DIAGRAM

A Data Flow Diagram (DFD) is a diagram that describes the flow of data and the processes that change data throughout a system. A structured analysis and design tool that can be used for flowcharting in place of or in association with information. Oriented and process-oriented system flowcharts. When analysts prepare the Data Flow Diagram, they specify the user needs at a level of detail that virtually determines the information flow into and out of the system and the required data resources. This network is constructed by using a set of symbols that do not imply physical implementations. The Data Flow Diagram reviews the current physical system, prepares input and output specification, specifies the implementation plan etc.

Four basic symbols are used to construct data flow diagrams. They are symbols that represent data source, data flows, and data transformations and data storage. The points at which data are transformed are represented by enclosed figures, usually circles, which are called nodes.

Steps to Construct Data Flow Diagram

Four Steps are generally used to construct a DFD.

- Process should be named and referred for easy reference. Each name should be representative of the reference.
- The destination of flow is from top to bottom and from left to right.
- When a process is distributed into lower-level details they are numbered.
- The names of data stores, sources and destinations are written in capital letters.

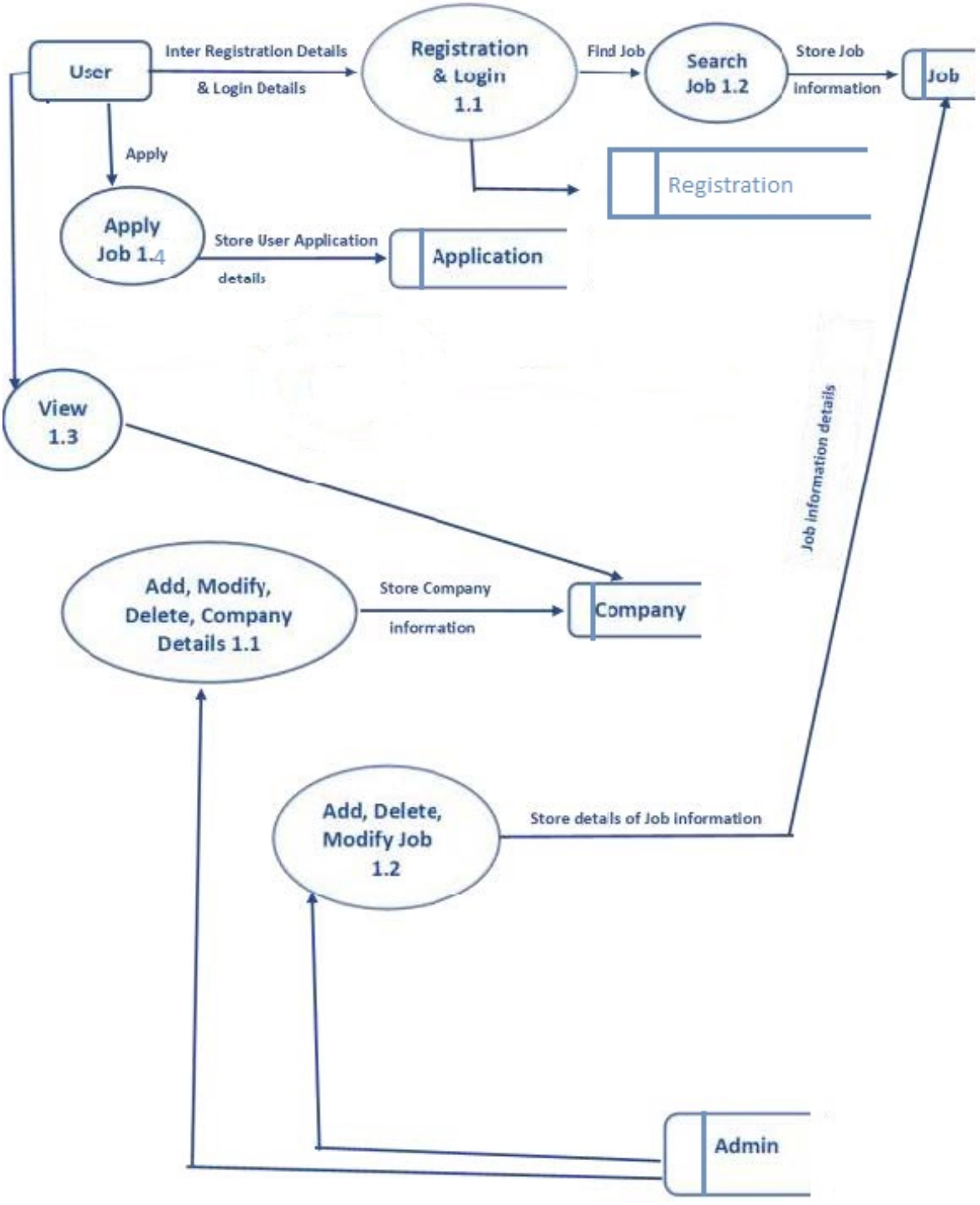
Rules for constructing a Data Flow Diagram-

- Arrows should not cross each other.
- Squares, Circles, Files must bear a name.
- Decomposed data flow squares and circles can have same names.
- Draw all data flow around the outside of the diagram.

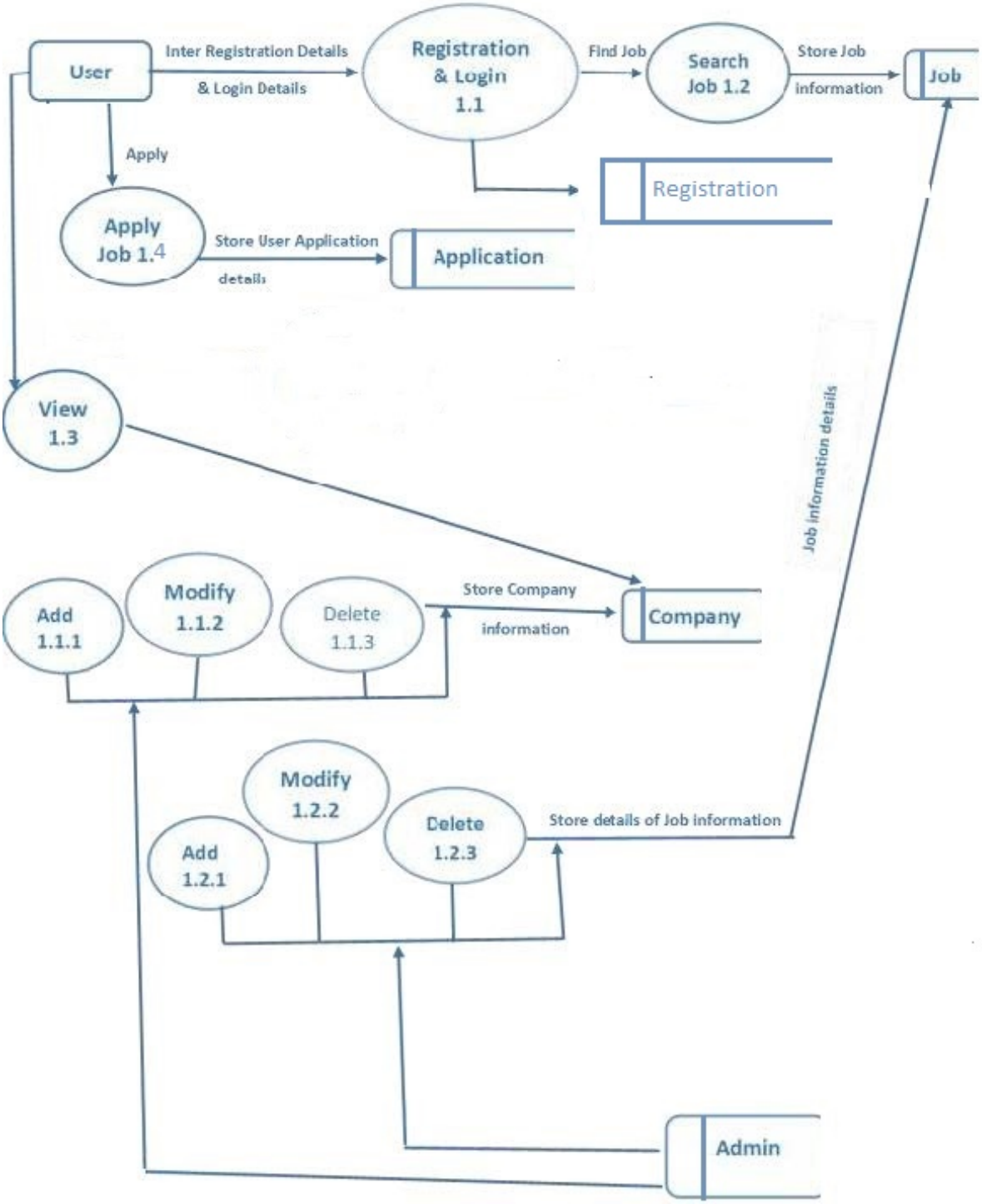
- **DATA FLOW DIAGRAM**

Level-0

Level 1


User
Inter Registration Details
& Login Details
Registration
& Login
1.1
Find Job
Search
Job 1.2
Store Job
information
Job
Apply
Registration
Apply
Job 1.4
Store User Application
details
Application
View
1.3
Job information details
Add, Modify,
Delete, Company
Details 1.1
Store Company
information
Company
Add, Delete,
Modify Job
1.2
Store details of Job information
Admin

Level 2


User
Inter Registration Details
& Login Details
Registration
& Login
1.1
Find Job
Search
Job 1.2
Store Job
information
Job
Apply
Registration
Apply
Job 1.4
Store User Application
details
Application
View
1.3
Job information details
Add
1.1.1
Modify
1.1.2
Delete
1.1.3
Store Company
information
Company
Modify
1.2.2
Add
1.2.1
Delete
1.2.3
Store details of Job information
Admin

2. ENTITY RELATIONSHIP DIAGRAM

In software engineering, an **entity–relationship model** (**ER model**) is a data model for describing the data or information aspects of a business domain or its process requirements, in an abstract way that lends itself to ultimately being implemented in a database such as a relational database. The main components of ER models are entities (things) and the relationships that can exist among them.

However, variants of the idea existed previously, and have been devised subsequently such as super type and subtype data entities and commonality relationships.

An entity–relationship model is a systematic way of describing and defining a business process. The process is modeled as components (*entities*) that are linked with each other by *relationships* that express the dependencies and requirements between them, such as: *one building may be divided into zero or more apartments, but one apartment can only be located in one building.* Entities may have various properties (*attributes*) that characterize them. Diagrams created to represent these entities, attributes, and relationships graphically are called entity–relationship diagrams.

An ER model is typically implemented as a database. In the case of a relational database, which stores data in tables, every row of each table represents one instance of an entity. Some data fields in these tables point to indexes in other tables; such pointers represent the relationships.

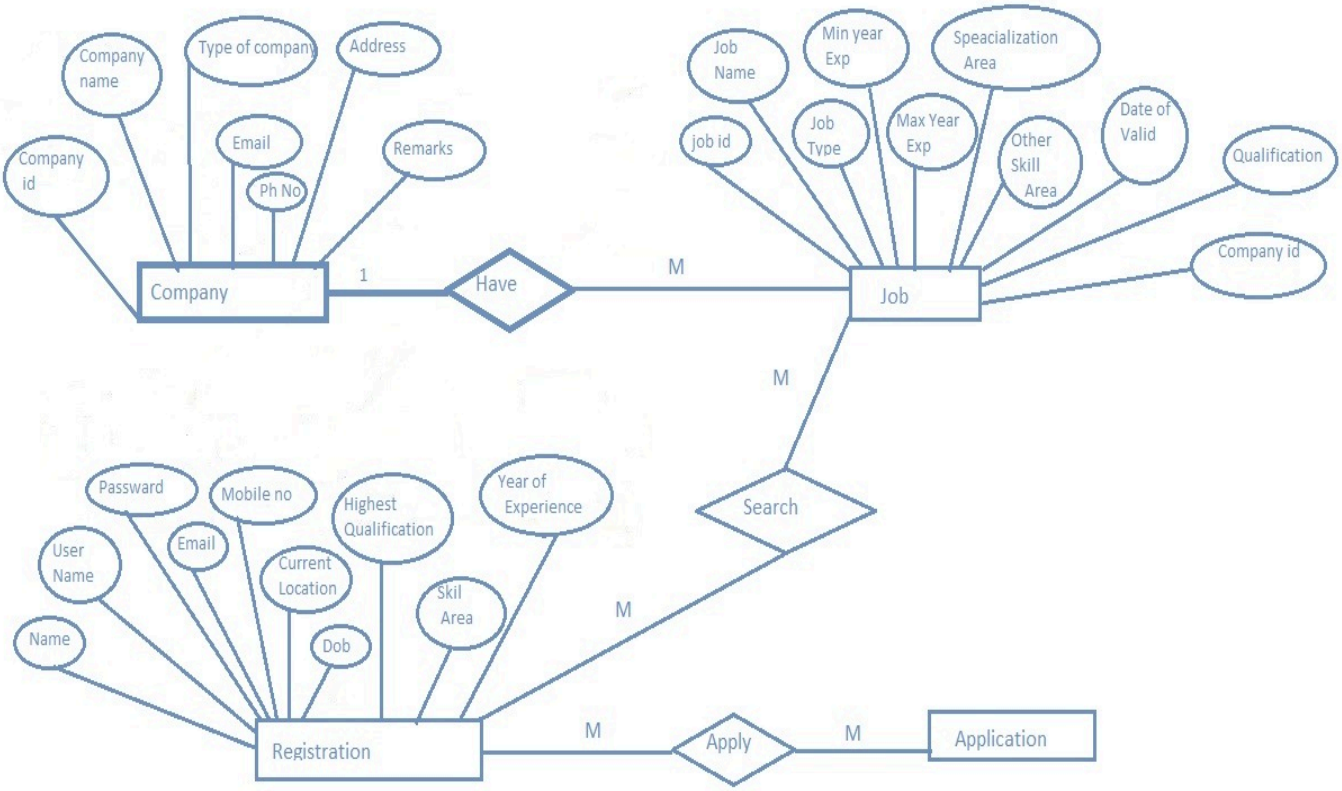
The three schema approach to software engineering uses three levels of ER models that may be developed.

An entity may be defined as a thing capable of an independent existence that can be uniquely identified. An entity is an abstraction from the complexities of a domain. When we speak of an entity, we normally speak of some aspect of the real world that can be distinguished from other aspects of the real world.

A relationship captures how entities are related to one another. Relationships can be thought of as verbs, linking two or more nouns.

Cardinality constraints are expressed as follows:

- a double line indicates a *participation constraint*, totality or subjectivity : all entities in the entity set must participate in *at least one* relationship in the relationship set;
- an arrow from entity set to relationship set indicates a key constraint, i.e. injectivity: each entity of the entity set can participate in *at most one* relationship in the relationship set;
- A thick line indicates both, i.e. bijectivity: each entity in the entity set is involved in *exactly one* relationship.
- An underlined name of an attribute indicates that it is a key: two different entities or relationships with this attribute always have different values for this attribute.


Company name
Type of company
Address
Company id
Email
Ph No
Remarks
Company
1
Have
M
Job
Job Name
Min year Exp
Speacialization Area
job id
Job Type
Max Year Exp
Other Skill Area
Date of Valid
Qualification
Company id
M
Search
Password
Mobile no
Highest Qualification
Year of Experience
User Name
Email
Current Location
Skil Area
Name
Dob
M
Registration
M
Apply
M
Application

ER DIAGRAM

3. USE CASE DIAGRAM

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well.

So only static behavior is not sufficient to model a system rather dynamic behavior is more important than static behavior. In UML there are five diagrams available to model dynamic nature and use case diagram is one of them. Now as we have to discuss that the use case diagram is dynamic in nature there should be some internal or external factors for making the interaction.

These internal and external agents are known as actors. So, use case diagrams are consists of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. A single use case diagram captures a particular functionality of a system. So, to model the entire system numbers of use case diagrams are used. The purpose of use case diagram is to capture the dynamic aspect of a system. But this definition is too generic to describe the purpose. Because other four diagrams (activity, sequence, collaboration and State chart) are also having the same purpose. So we will look into some specific purpose which will distinguish it from other four diagrams.

Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. So, when a system is analyzed to gather its functionalities use cases are prepared and actors are identified.

Now when the initial task is complete use case diagrams are modeled to present the outside view. So, in brief, the purposes of use case diagrams can be as follows:

- Used to gather requirements of a system.
- Used to get an outside view of a system.
- Identify external and internal factors influencing the system.
- Show the interacting among the requirements are actors.

How to draw Use Case Diagram?

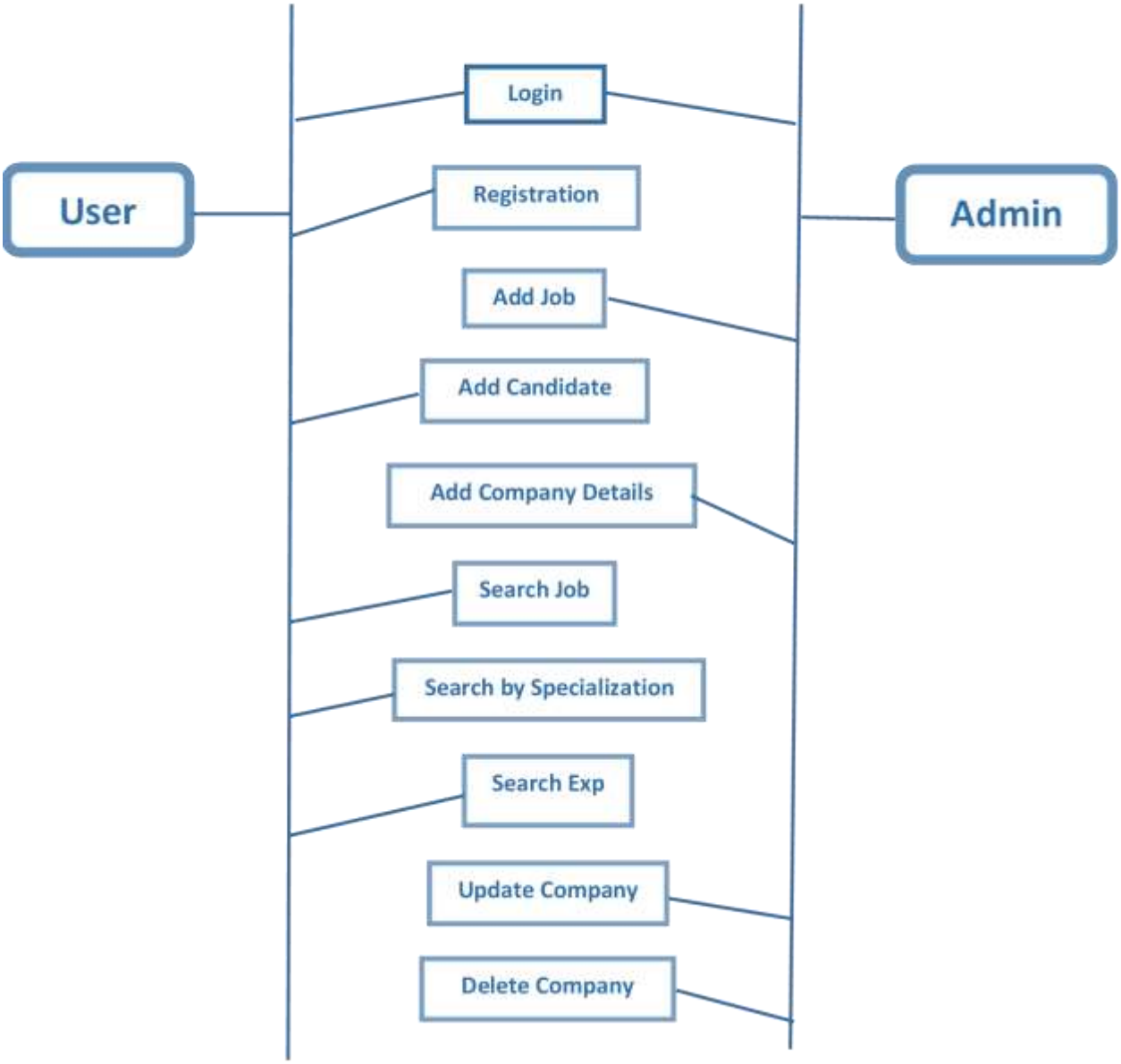
Use case diagrams are considered for high level requirement analysis of a system. So, when the requirements of a system are analyzed the functionalities are captured in use cases. So, we can say that uses cases are nothing but the system functionalities written in an organized manner. Now the second things which are relevant to the use cases are the actors. Actors can be defined as something that interacts with the system. The actors can be human user, some internal applications or may be some external applications. So, in a brief when we are planning to draw a use case diagram, we should have the following items identified.

- Functionalities to be represented as an use case
- Actors
- Relationships among the use cases and actors.

Use case diagrams are drawn to capture the functional requirements of a system. So, after identifying the above items we have to follow the following guidelines to draw an efficient use case diagram.

- The name of a use case is very important. So, the name should be chosen in such a way so that it can identify the functionalities performed.
- Give a suitable name for actors.
- Show relationships and dependencies clearly in the diagram.
- Do not try to include all types of relationships. Because the main purpose of the diagram is to identify requirements.
- Use note whenever required to clarify some important points.

USE CASE DIAGRAM



4. MODULARIZATION DETAILS

As Modularization has gained increasing focus from companies outside its traditional industries of aircraft and automotive, more and more companies turn to it as strategy and product development tool. I intend to explain the importance aspects of modularization and how it should be initiated within a company. After determining the theoretical steps of modularization success described in literature, I intend to conduct a multiple case study of companies who have implemented modularization in order to find how real-world modularization was initiated and used to improve the company's competitiveness. By combining theory and practical approach to modularization I will derive at convergence and divergence between theoretical implementation to modularization and real-world implementation to modularization. This gives a valuable input for both implantations in companies as well as new aspects to be further.

DATA INTEGRITY AND CONSTRAINTS

Data integrity is normally enforced in a database system by a series of integrity constraints or rules. Three types of integrity constraints are an inherent part of the relational data model: entity integrity, referential integrity and domain integrity:

- *Entity integrity* concerns the concept of a primary key. Entity integrity is an integrity rule which states that every table must have a primary key and that the column or columns chosen to be the primary key should be unique and not null.

- Concerns the concept of a foreign key. The referential integrity rule states that any foreign-key value can only be in one of two states. The usual state of affairs is that the foreign-key value refers to a primary key value of some table in the database. Occasionally, and this will depend on the rules of the data owner, a foreign-key value can be null. In this case we are explicitly saying that either there is no relationship between the objects represented in the database or that this relationship is unknown.

- *Domain integrity* specifies that all columns in a relational database must be declared upon a defined domain. The primary unit of data in the relational data model is the data item. Such data items are said to be non-decomposable or atomic. A domain is a set of values of the same type.

5. DATABASE DESIGN

A database is an organized mechanism that has capability of storing information through which a user can retrieve stored information in an effective and efficient manner. The data is the purpose of any database and must be protected.

The database design is two level processes. In the first step, user requirements are gathered together and a database is designed which will meet these requirements as clearly as possible. This step is called information Level design and it is taken independent of any individual DBMS.

In the following snapshots we display the way we have used SQL Server as the back-end RDBMS for our project and the various entities that have been used along with their table definition and table data.

6.DATA DICTIONARY

TABLE: JOB

Field Name	*Data type*	*Size*	*constraints*
Job_Id	Int	11	
Job_Name	Varchar	30	
Job_Type	Varchar	30	
Minimum_Year_Experience	Varchar	20	
Maximum_Year_Experience	Varchar	20	
Specialization_Area	Varchar	30	
Other_Skill_Area	Varchar	30	
Date_of_Valid	Varchar	20	
Qualification	Varchar	20	
Company_Id	Int	11	

Table: Resume

Field Name	*Datatype*	*Size*	*constraints*
RESUMEID	Varchar2	25	Primary key
IMAGEBLOB	Varchar2	10	
IMAGEBLOB	Varchar2	10	

Table: Registration

Field Name	*Datatype*	*Size*	*Constraints*
Name	*Varchar*	*25*	
Username	Varchar	15	Primary key
Password	Varchar	15	
Email	varchar	30	
Mobile_No	Varchar	10	
Current Location	Varchar	25	
Highest Qualification	Varchar	25	
Year_of_Experience	Varchar	10	
Skill_Area	Varchar	10	
DOB	Varchar	20	

Table: Application

Field name	Datatype	Size	Constraints
Job_Id	Int	15	Primary key
Job_Name	Varchar	30	
Job_Type	Varchar	30	
Minimum_Year_Experience	Varchar	30	
Maximum_Year_Experience	Varchar	30	
Specialization_Area	Varchar	30	
Date_of_Valid	Varchar	20	
Qualification	Varchar	20	
Company_Id	Varchar	15	
Company_Name	Varchar	30	
Username	Varchar	30	

Table: Company

Field Name	*Datatype*	*Size*	*Constraints*
Company_Id	Int	11	
Company_Name	Varchar	20	
Types_of_Company	Varchar	30	
Email	varchar	30	
Ph_No	Varchar	10	
Address	Varchar	50	
Remarks	Varchar	30	

Table: Company_Job_Report

Field name	**Datatype**	**Size**	**Constraints**
Company_Id	Int	11	Primary key
Company_Name	Varchar	20	
Job_Id	Int	15	
Job_Name	Varchar	30	
Job_Type	Varchar	30	
Minimum_Year_Experience	Varchar	20	
Minimum_Year_Experience	Varchar	20	
Specialization_Area	Varchar	30	
Date_of_Valid	Varchar	20	
Qualification	Varchar	20	

7. USER INTERFACE DESIGN

User interface design (UID) or **user interface engineering** is the design of user interfaces for machines and software, such as computers, home appliances, mobile devices, and other electronic devices, with the focus on maximizing the user experience. The goal of user interface design is to make the user's interaction as simple and efficient as possible, in terms of accomplishing user goals (user-centered design).

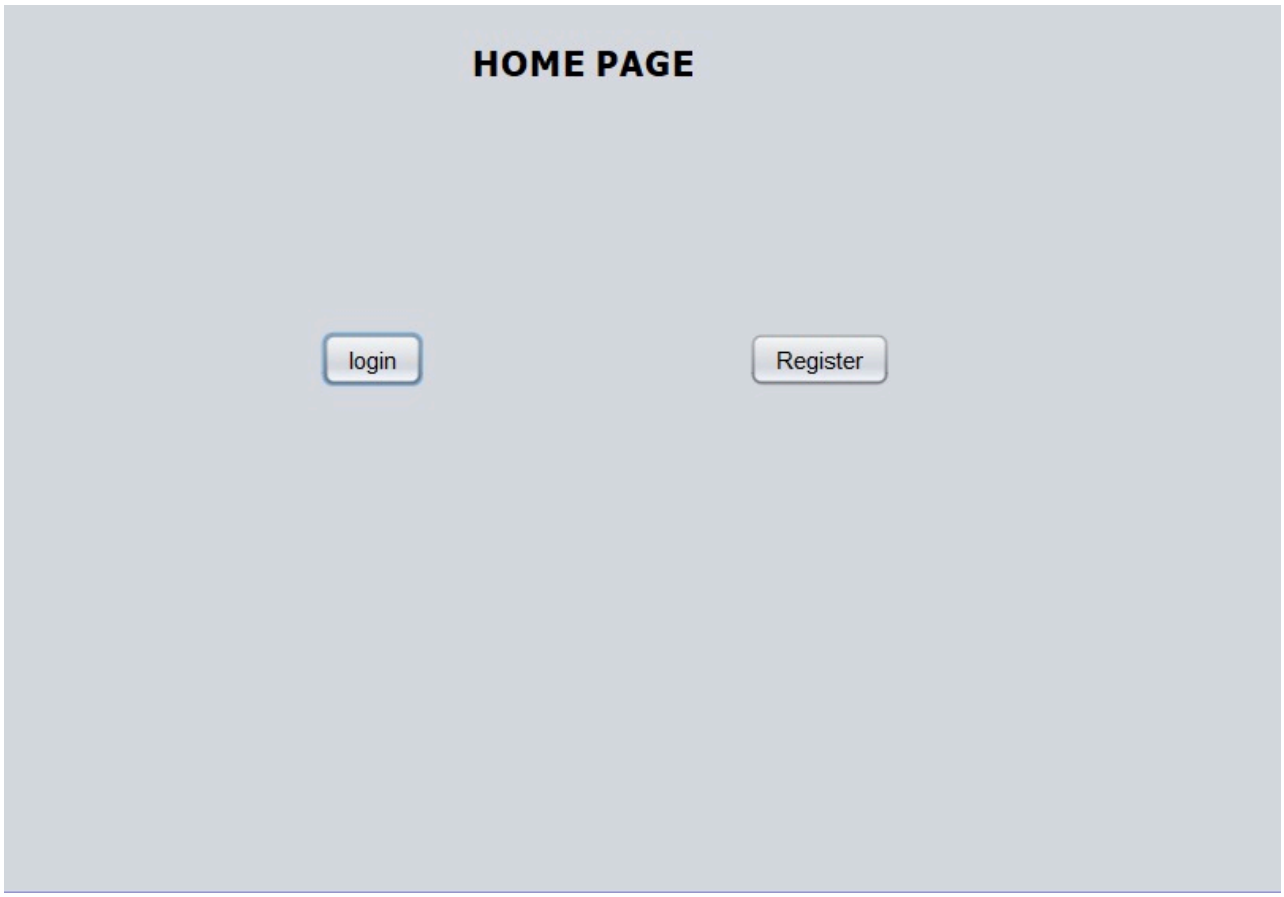
Good user interface design facilitates finishing the task at hand without drawing unnecessary attention to it. Graphic design and typography are utilized to support its usability, influencing how the user performs certain interactions and improving the aesthetic appeal of the design; design aesthetics may enhance or detract from the ability of users to use the functions of the interface. The design process must balance technical functionality and visual elements (e.g., mental model) to create a system that is not only operational but also usable and adaptable to changing user needs.

Interface design is involved in a wide range of projects from computer systems, to cars, to commercial planes; all of these projects involve much of the same basic human interactions yet also require some unique skills and knowledge. As a result, designers tend to specialize in certain types of projects and have skills centered on their expertise, whether that be software design, user research, web design, or industrial design.

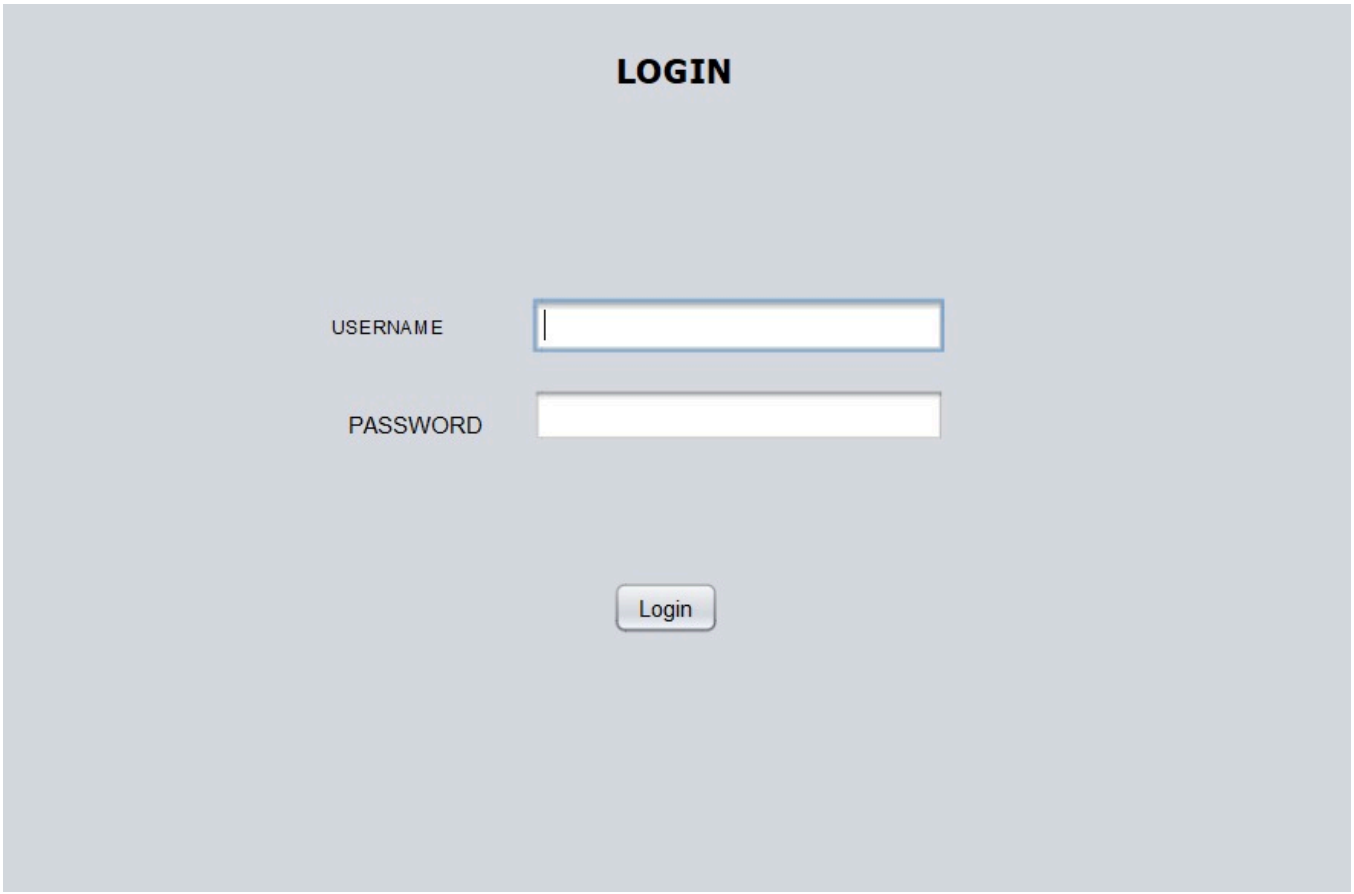
SNAPSHOTS

ADMIN PART:

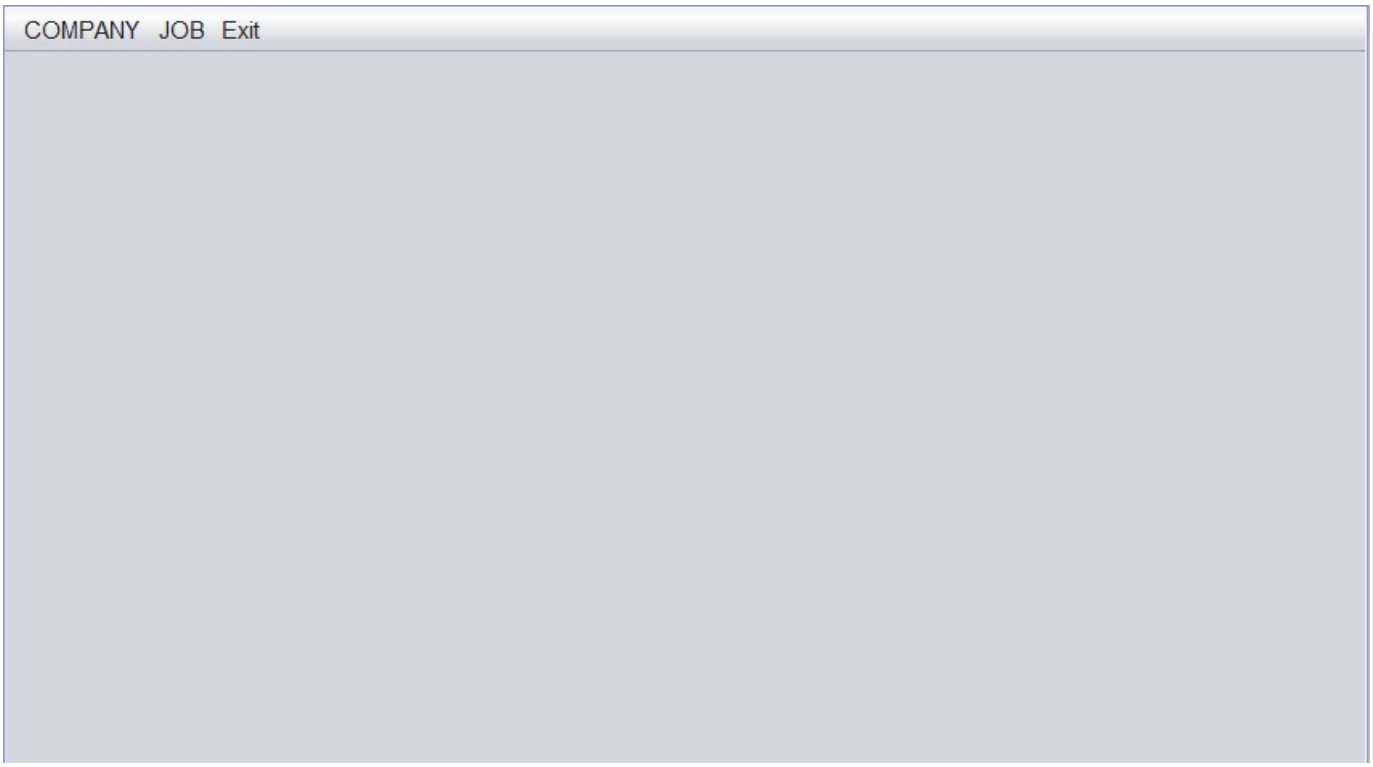
HOME:



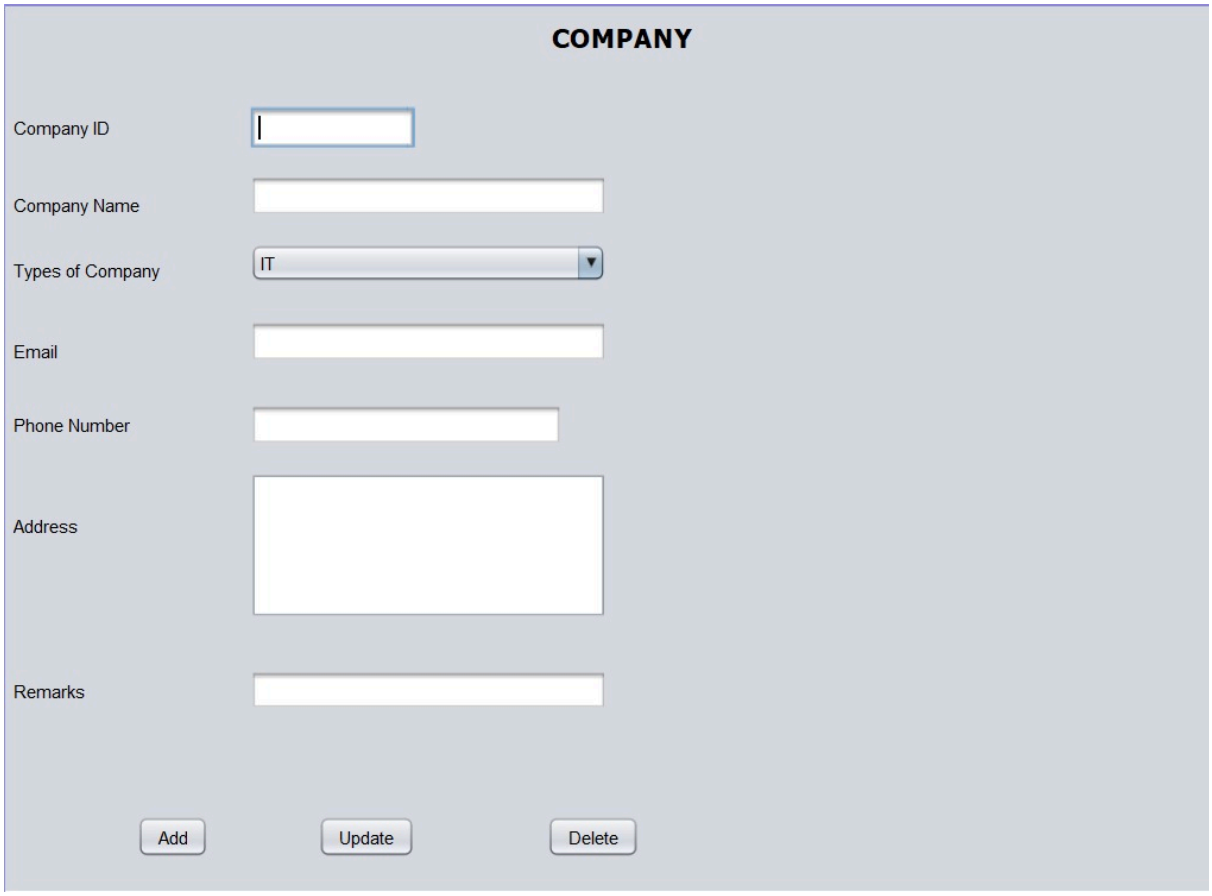
ADMIN LOGIN:



ADMIN HOME PAGE:

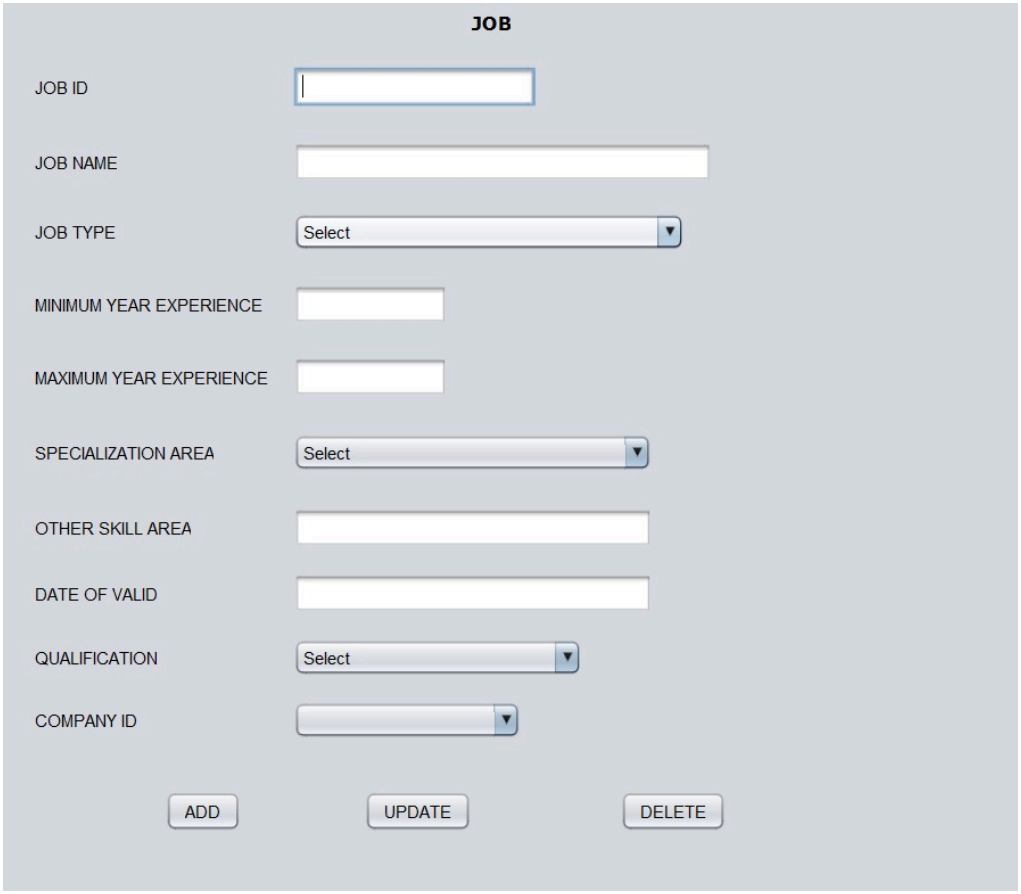


ADD COMPANY:


COMPANY
Company ID
Company Name
Types of Company
IT
Email
Phone Number
Address
Remarks
Add
Update
Delete

COMPANY REPORT:

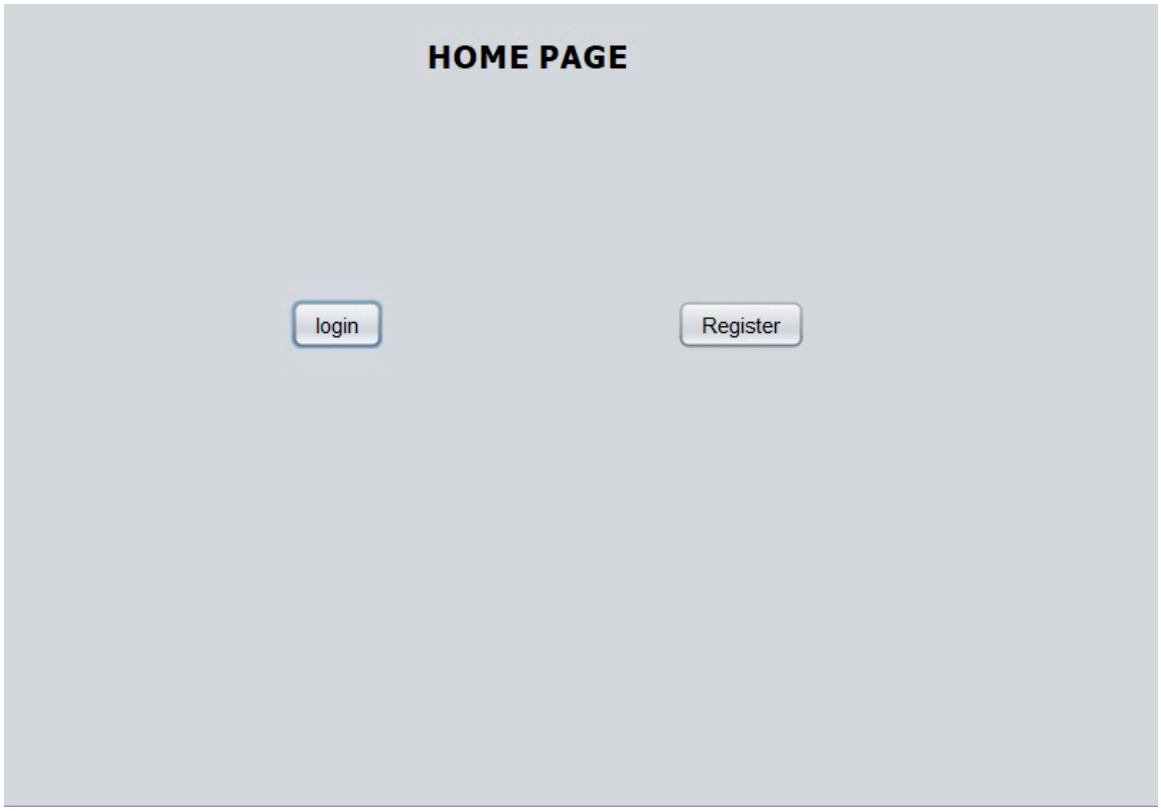
ADD JOB:


JOB
JOB ID
JOB NAME
JOB TYPE
Select
MINIMUM YEAR EXPERIENCE
MAXIMUM YEAR EXPERIENCE
SPECIALIZATION AREA
Select
OTHER SKILL AREA
DATE OF VALID
QUALIFICATION
Select
COMPANY ID
ADD
UPDATE
DELETE

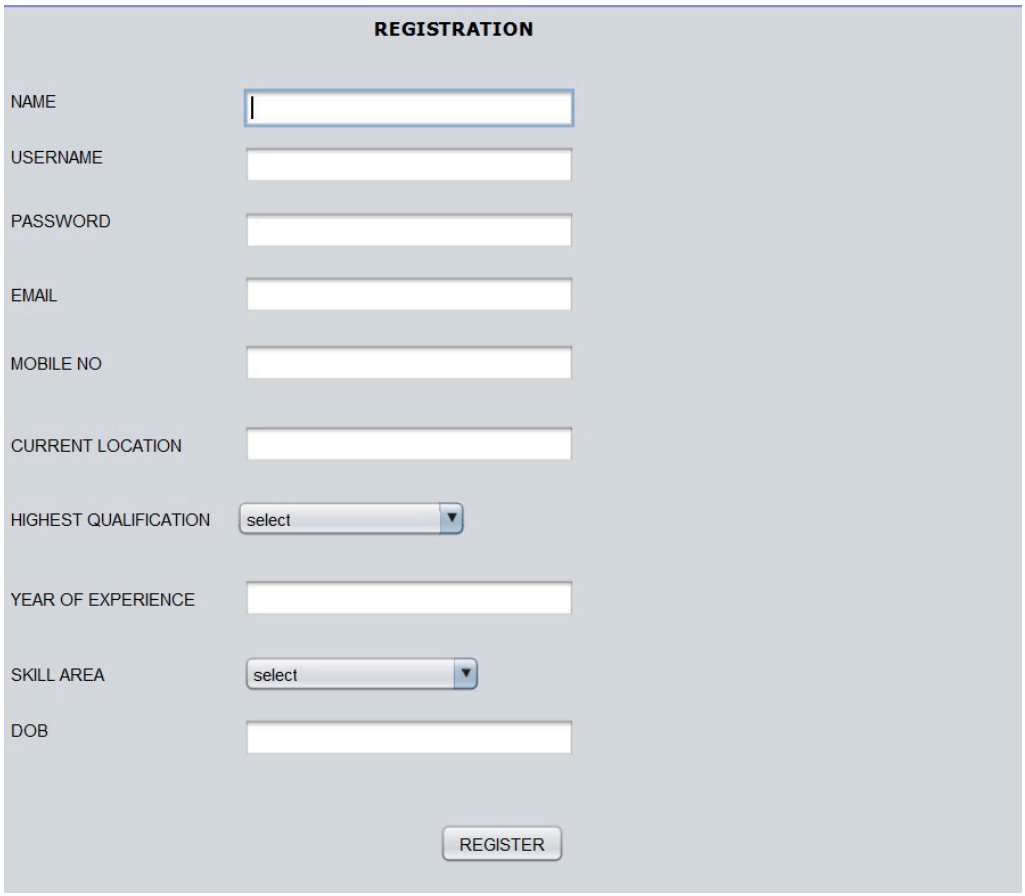
JOB REPORT:

USER PAGE:

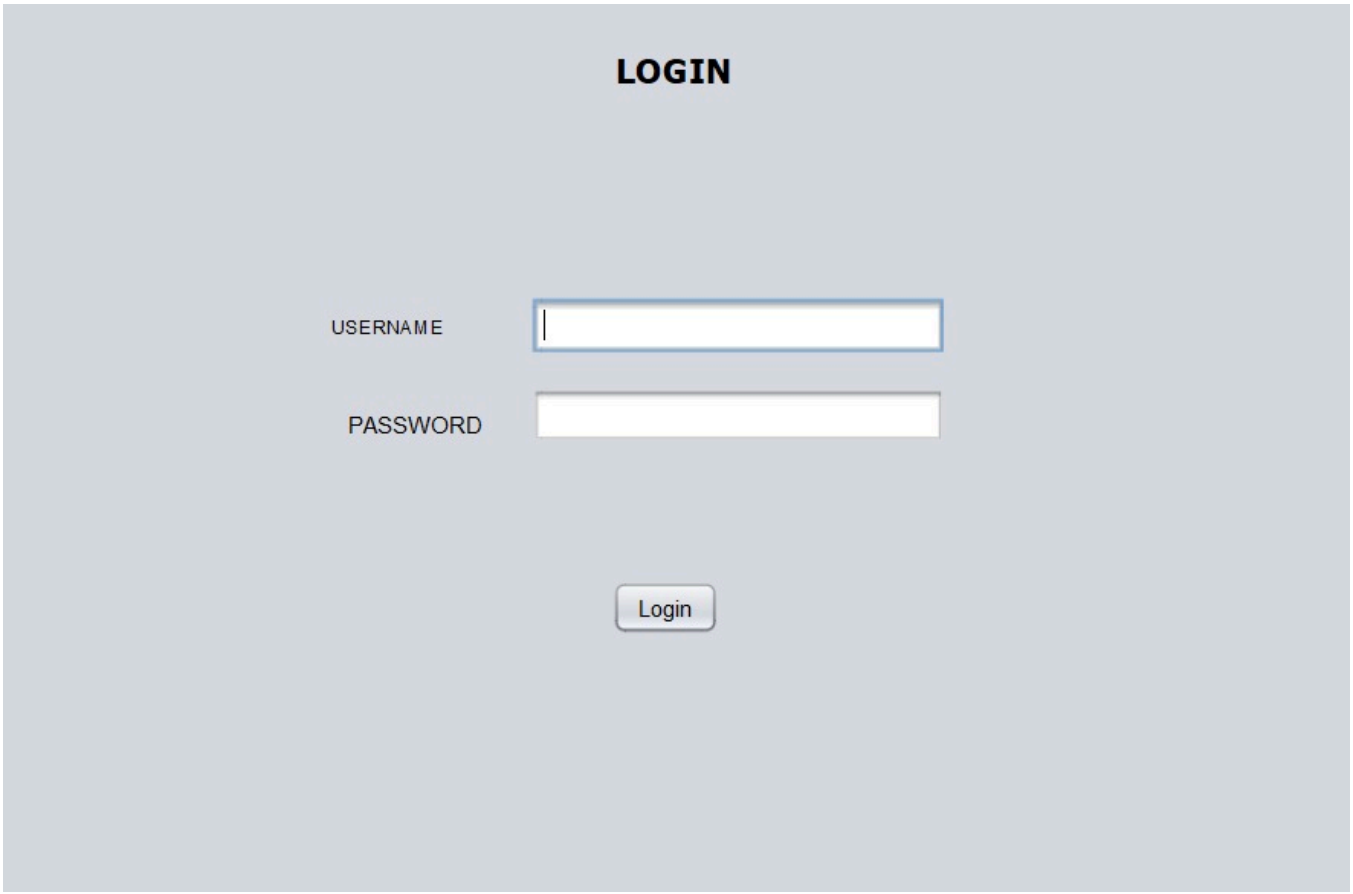
HOME:



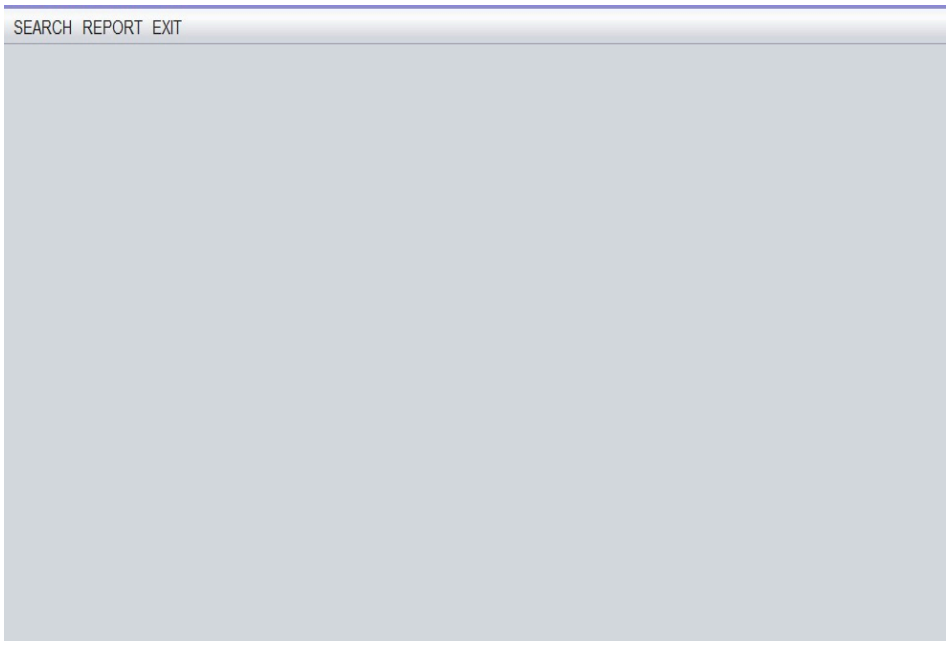
REGISTRATION PAGE:



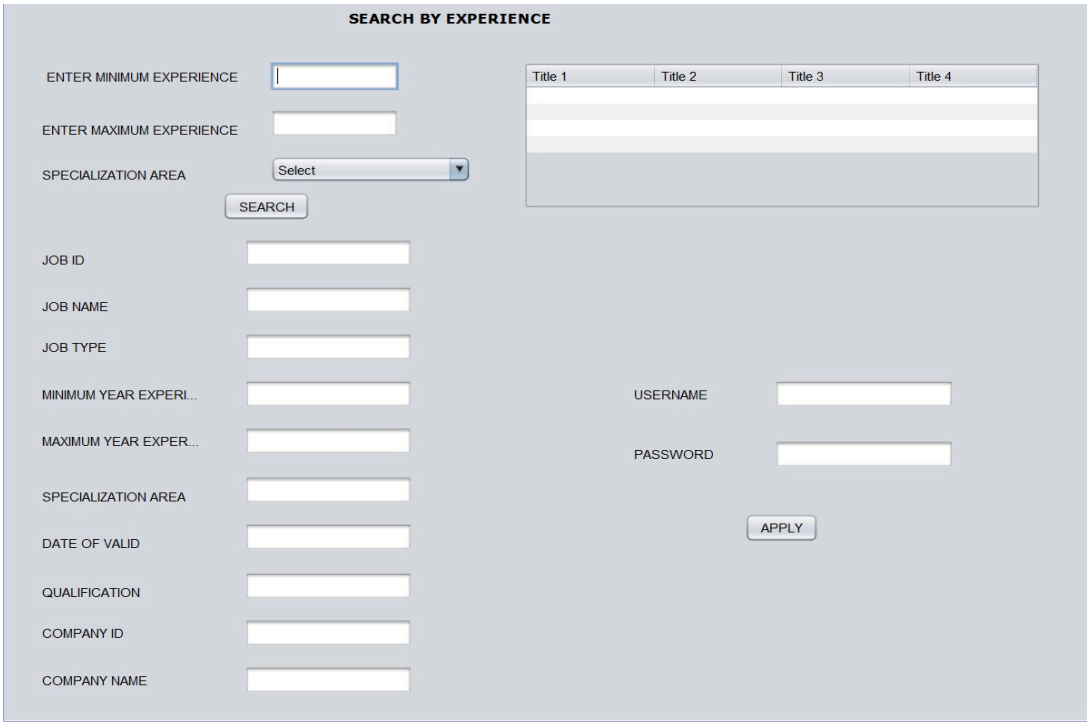
LOGIN:



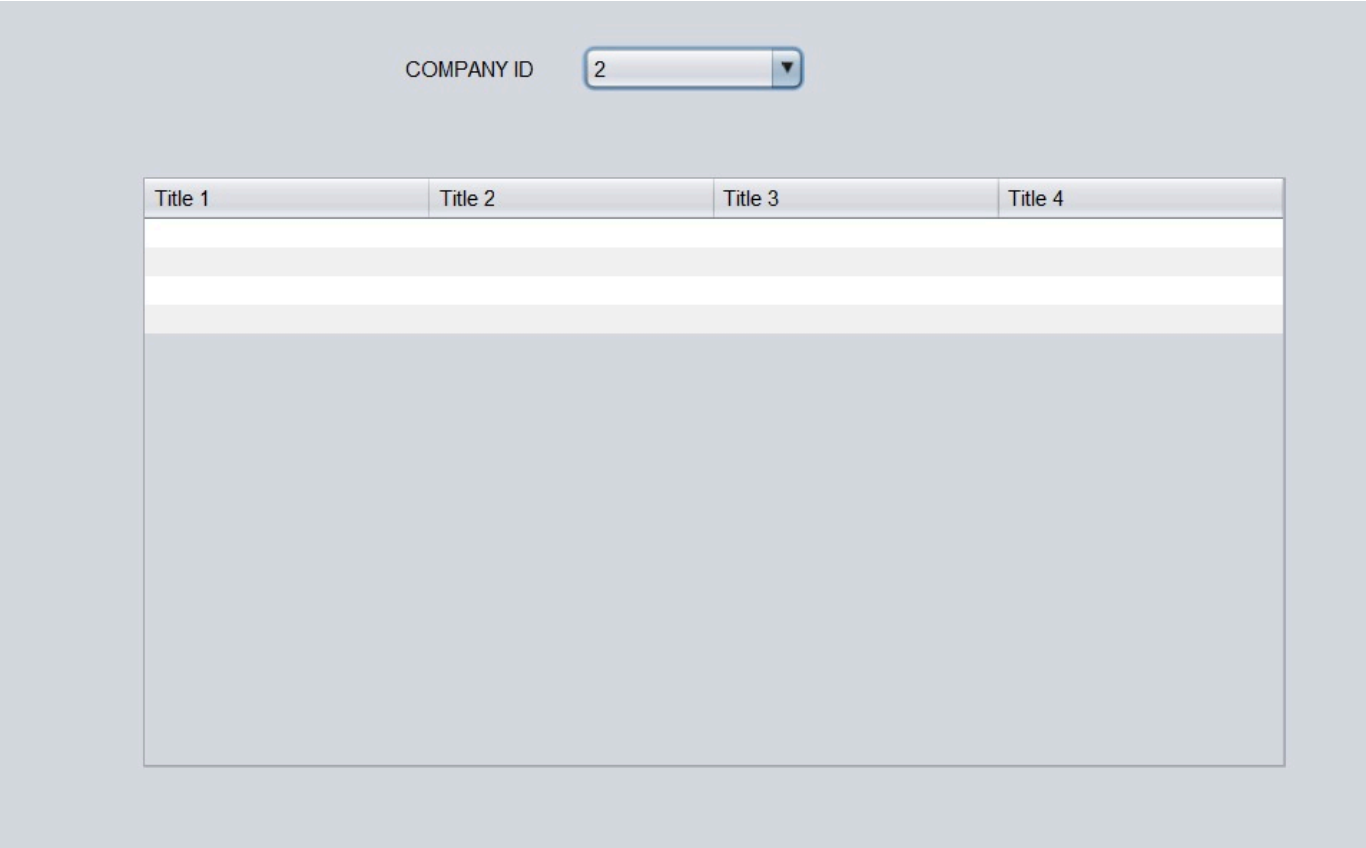
HOME PAGE:



SEARCH BY EXPERIENCE:



SEARCH BY COMPANY:



REPORT:

JOB REPORT:

COMPANY REPORT:

CODING

ADMIN LOGIN :

```
package Job_Portal_System;
import java.sql.*;
import javax.swing.JOptionPane;
public class Admin_login extends javax.swing.JFrame
{ Connection cn=null;
PreparedStatement ps=null;
public Admin_login() {
      initComponents();
}
public void refreshData()
 {
     uname_txt.setText("");
    password_txt.setText("");
 }
   @SuppressWarnings("unchecked")
  private void
login_buttonActionPerformed(java.awt.event.ActionEvent evt) {

       String ext = new String(password_txt.getPassword());
      String user= uname_txt.getText();
```

```
if(user.equalsIgnoreCase("admin") &&
ext.equalsIgnoreCase("pass") ){
            JOptionPane.showMessageDialog(this,"login
successfully");
            new Company().setVisible(true);
        }
        else{
            JOptionPane.showMessageDialog(this,"Wrong
combination");
        }
        refreshData();
    }
  public static void main(String args[]) {

        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Admin_login().setVisible(true);}
);}
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JButton login_button;
    private javax.swing.JPasswordField password_txt;
    private javax.swing.JTextField uname_txt;
    }
```

COMPANY :

```
public class Company extends javax.swing.JFrame {
 Connection cn=null;
    PreparedStatement ps=null;
    ResultSet rs=null;
    Statement st=null;
    String insert_sql="insert into company values(?,?,?,?,?,?,?)";
    String update_sql="update company set  Email=?,PH_No=?,Address=?  where
Comapany_Id=?";
    String delete_sql="delete from company where Comapany_Id=?";
 String sql="select Types_Of_Company from company ";
public void addData() {
       try{
         ConnectionFactory con=new ConnectionFactory();
         cn=con.getConn();
           ps=cn.prepareStatement(insert_sql;
           ps.setString(1,cid_txt.getText(;
           ps.setString(2,cname_txt.getText(;
          ps.setString(3,company_combo.getSelectedItem(.toString(;
           ps.setString(4,email_txt.getText(;
           ps.setString(5,phno_txt.getText(;
           ps.setString(6,address_txt.getText(;
           ps.setString(7,remarks_txt.getText(;
           ps.executeUpdate();
           JOptionPane.showMessageDialog(this,"DATA
         INSERTED");
```

```
}catch(SQLException se){
      se.printStackTrace();
     }
  }
   public void updateData() {
     try{
       ConnectionFactory con=new ConnectionFactory();
       cn=con.getConn();
         ps=cn.prepareStatement(update_sql);
         ps.setString(1,email_txt.getText());
         ps.setString(2,phno_txt.getText());
         ps.setString(3,address_txt.getText());
          ps.setString(4,cid_txt.getText());
         ps.executeUpdate();
        JOptionPane.showMessageDialog(this,"DATA UPDATED");
     }catch(SQLException se){
      se.printStackTrace();
     }
  }
public void deleteData() {
     try{
       ConnectionFactory con=new ConnectionFactory();
       cn=con.getConn();
         ps=cn.prepareStatement(delete_sql);
        ps.setString(1,cid_txt.getText());
```

```
int i= ps.executeUpdate();
        JOptionPane.showMessageDialog(this,"DATA DELETED");
 }catch(SQLException se){
        se.printStackTrace();
       }
```

COMPANY REPORT:

```
public class Company_report extends javax.swing.JFrame {
 Connection cn=null;
    PreparedStatement ps=null;
    ResultSet rs=null;
    Statement st=null;
     String select_sql="select * from company order by Company_Id";
public void getData() {
       try{
         ConnectionFactory con=new ConnectionFactory();
         cn=con.getConn();
         st=cn.createStatement();
         rs=st.executeQuery(select_sql);
          company_table.setModel(DbUtils.resultSetToTableModel(rs));
   }catch(SQLException se){
        se.printStackTrace();
       }
   }
 private void show_buttonActionPerformed(java.awt.event.ActionEvent evt)
{       getData();    }
```

COMPANY WISE JOB:

```
public class Company_wise_job extends javax.swing.JFrame {
Connection cn =null;
   Statement st= null;
   ResultSet rs=null;
   PreparedStatement ps=null;
   String sql="select Company_Id from job";
 public void getId()
   {
      try
      {
         ConnectionFactory con= new ConnectionFactory();
         cn=con.getConn();
         st=cn.createStatement();
         rs=st.executeQuery(sql);
         while(rs.next()){
            cid_combo.addItem(rs.getString(1));
         }
}
      catch(SQLException se)
            {
               se.printStackTrace();
            }
   }
```

```
public void getIdDetails(String Company_Id){
        String id="select * from job where Company_Id='"+Company_Id+"'";
         try
       {
         ConnectionFactory con= new ConnectionFactory();
         cn=con.getConn();
         st=cn.createStatement();
         rs=st.executeQuery(id);
        cwj_table.setModel(DbUtils.resultSetToTableModel(rs));

       }
       catch(SQLException se)
             {
                se.printStackTrace();
             }
     }
 private void formWindowOpened(java.awt.event.WindowEvent evt) {
    getId();
   }
 private void search_buttonActionPerformed(java.awt.event.ActionEvent evt) {
   String Company_Id=cid_combo.getSelectedItem().toString();
      getIdDetails(Company_Id);
   }
```

APPLY LOGIN:

```
public class Apply_login extends javax.swing.JFrame {
 private void login_buttonActionPerformed(java.awt.event.ActionEvent evt) {
       String pass = pass_txt.getText();
       String uname= uname_txt.getText();
       LoginDao ldao=new LoginDao();
boolean flag=ldao.checkLogin(uname, pass);
          if(flag==true)
          {
            JOptionPane.showMessageDialog(this,"LOGIN SUCCESSFUL");
            new Application_Frm().setVisible(true);
          }
else {
            JOptionPane.showMessageDialog(this,"WRONG USERNAME &
PASSWORD");
            new Apply_login().setVisible(true); }}
```

LOGIN DAO:

```
import java.sql.*;
public class LoginDao {
    Connection cn=null;
    PreparedStatement ps=null;
    ResultSet rs=null;
    Statement st=null;
   public boolean checkLogin(String uname,String pass)
   {
        boolean flag=false;
         try{
String select_sql="select Username, Password from registration where Username='"+uname+"'
and Password='"+pass+"'";
         ConnectionFactory con=new ConnectionFactory();
         cn=con.getConn();
         st=cn.createStatement();
        rs=st.executeQuery(select_sql);
        if(rs.next()){
            flag=true;
        }
         }
catch(SQLException se){
        se.printStackTrace();
       }
         return flag;
}}
```

JOB:

```
public class job_Frm extends javax.swing.JFrame {
    Connection cn =null;
    Statement st= null;
    ResultSet rs=null;
    PreparedStatement ps=null;
    String insert_sql="insert into job values(?,?,?,?,?,?,?,?,?,?)";
    String update_sql="update job set job_name=? where job_id=?";
    String delete_sql="delete from job where job_id=?";
    String sql="select Company_Id from company";
 public void insertData()
    {
        try
        {
            ConnectionFactory con= new ConnectionFactory();
            cn=con.getConn();
            ps=cn.prepareStatement(insert_sql);
            ps.setString(1,jobid_txt.getText());
            ps.setString(2,jobname_txt.getText());
            ps.setString(3,jobtype_combo.getSelectedItem().toString());
            ps.setString(4,min_txt.getText());
            ps.setString(5,max_txt.getText());
            ps.setString(6,specializationarea_combo.getSelectedItem().toString());
            ps.setString(7,otherskill_txt.getText());
            ps.setString(8,dateofvalid_txt.getText());
```

```
          ps.setString(9,qualification_combo.getSelectedItem().toString());
            ps.setString(10,cid_combo.getSelectedItem().toString());
          ps.executeUpdate();
           JOptionPane.showMessageDialog(this,"INSERT SUCCESSFUL");
        }
        catch(SQLException se)
                {
                   se.printStackTrace();
                }
    }
public void updateData()
    {
        try
        {
           ConnectionFactory con= new ConnectionFactory();
           cn=con.getConn();
           ps=cn.prepareStatement(update_sql);
           ps.setString(1,jobname_txt.getText());
           ps.setString(2,jobid_txt.getText());
 ps.executeUpdate();
        }
        catch(SQLException se)
                {
                   se.printStackTrace();
                }}
 public void deleteData()
    {
        try
        {
           ConnectionFactory con= new ConnectionFactory();
           cn=con.getConn();
           ps=cn.prepareStatement(delete_sql);
           ps.setString(1,jobid_txt.getText());
           ps.executeUpdate();
        }
        catch(SQLException se)
                {
                   se.printStackTrace();
                }
public void getId()
    {
        try
        {
           ConnectionFactory con= new ConnectionFactory();
           cn=con.getConn();
           st=cn.createStatement();
           rs=st.executeQuery(sql);
           while(rs.next()){
              cid_combo.addItem(rs.getString(1));
           }

        }
```

```
catch(SQLException se)
            {
               se.printStackTrace();
            }
   }
     public void getIdDetails(String Company_Id){
        String id="select * from company where Company_Id='"+Company_Id+"'";
         try
      {
         ConnectionFactory con= new ConnectionFactory();
         cn=con.getConn();
         st=cn.createStatement();
         rs=st.executeQuery(id);
}
      catch(SQLException se)
            {
               se.printStackTrace();
            }
     }
 public void refreshData()
  {
     jobid_txt.setText("");
    jobname_txt.setText("");
    min_txt.setText("");
    max_txt.setText("");

   otherskill_txt.setText("");
   dateofvalid_txt.setText("");

  }
   private void add_buttonActionPerformed(java.awt.event.ActionEvent evt) {
       this.insertData();
       this.refreshData();
   }
 private void delete_buttonActionPerformed(java.awt.event.ActionEvent evt) {
      this.deleteData();
       this.refreshData();
   }
 private void update_buttonActionPerformed(java.awt.event.ActionEvent evt) {
      this.updateData();
       this.refreshData();
   }
private void formWindowOpened(java.awt.event.WindowEvent evt) {
      this.getId();
   }
private void cid_comboPopupMenuWillBecomeInvisible(javax.swing.event.PopupMenuEvent
evt) {
      String Company_Id=cid_combo.getSelectedItem().toString();
       getIdDetails(Company_Id);
   }
```

JOB REPORT:

```
public class Job_report extends javax.swing.JFrame {
Connection cn =null;
    Statement st= null;
    ResultSet rs=null;
    PreparedStatement ps=null;
    String select_sql="select * from job order by job_id";
public void getData()
    {
        try
        {
            ConnectionFactory con= new ConnectionFactory();
            cn=con.getConn();
            st=cn.createStatement();
            rs=st.executeQuery(select_sql);
            job_table.setModel(DbUtils.resultSetToTableModel(rs));
        }
        catch(SQLException se)
                {
                   se.printStackTrace();
                }
    }
private void show_buttonActionPerformed(java.awt.event.ActionEvent evt) {
         getData();
    }
```

REGISTRATION:

```
public class Registration extends javax.swing.JFrame {
Connection cn=null;
    PreparedStatement ps=null;
    ResultSet rs=null;
    Statement st=null;
    String insert_sql="insert into registration values(?,?,?,?,?,?,?,?,?,?)";
public void addData(){
        try{
         ConnectionFactory con=new ConnectionFactory();
         cn=con.getConn();
           ps=cn.prepareStatement(insert_sql);
           ps.setString(1,name_txt.getText());
           ps.setString(2,uname_txt.getText());
           ps.setString(3,pass_txt.getText());
           ps.setString(4,email_txt.getText());
           ps.setString(5,mno_txt.getText());
           ps.setString(6,cloc_txt.getText());
           ps.setString(7,hq_combo.getSelectedItem().toString());
           ps.setString(8,yoe_txt.getText());
           ps.setString(9,sa_combo.getSelectedItem().toString());
           ps.setString(10,dob_txt.getText());
           ps.executeUpdate();
```

```
 JOptionPane.showMessageDialog(this,"DATA INSERTED");
 }catch(SQLException se){
       se.printStackTrace();
      }
   }
   public void resetData(){
     name_txt.setText("");
     uname_txt.setText("");
     pass_txt.setText("");
     email_txt.setText("");
     mno_txt.setText("");
     cloc_txt.setText("");
     yoe_txt.setText("");
     dob_txt.setText("");
  }
private void register_buttonActionPerformed(java.awt.event.ActionEvent evt) {
      addData();
       resetData();
       }
```

SEARCH BY EXPERIENCE:

```
public class SearchByExperience extends javax.swing.JFrame {
 Connection cn =null;
   Statement st= null;
   ResultSet rs=null;
   PreparedStatement ps=null;
 String insert_sql="insert into application values(?,?,?,?,?,?,?,?,?,?,?,?)";
public void insertData()
   {
      try
       {
          ConnectionFactory con= new ConnectionFactory();
          cn=con.getConn();
          ps=cn.prepareStatement(insert_sql);
          ps.setString(1,jobid_txt.getText());
          ps.setString(2,jobname_txt.getText());
          ps.setString(3,jobtype_txt.getText());
          ps.setString(4,min_txt1.getText());
          ps.setString(5,max_txt1.getText());
          ps.setString(6,specializationarea_txt1.getText());
          ps.setString(7,dateofvalid_txt.getText());
          ps.setString(8,qualification_txt.getText());
          ps.setString(9,cid_txt.getText());
          ps.setString(10,cname_txt.getText());
          ps.setString(11,uname_txt.getText());
        ps.executeUpdate();
 JOptionPane.showMessageDialog(this,"APPLY SUCCESSFUL");
       }
```

```
 catch(SQLException se)
              {
                 se.printStackTrace();
              }
    }
public void getIdDetails(String Minimum_Year_of_Experience,String
Maximum_Year_of_Experience,String Specialization_Area ){
          String id="select  * from company_job_report where
Minimum_Year_of_Experience='"+Minimum_Year_of_Experience+"'and
Maximum_Year_of_Experience='"+Maximum_Year_of_Experience+"'and
Specialization_Area='"+Specialization_Area+"'";
           try
        {
           ConnectionFactory con= new ConnectionFactory();
           cn=con.getConn();
           st=cn.createStatement();
           rs=st.executeQuery(id);
          search_table.setModel(DbUtils.resultSetToTableModel(rs));
           }
catch(SQLException se)
              {
                 se.printStackTrace();
              }
      }
 private void search_buttonActionPerformed(java.awt.event.ActionEvent evt) {
       String Minimum_Year_of_Experience =min_txt.getText();
        String Maximum_Year_of_Experience =max_txt.getText();
         String Specialization_Area=specializationarea_combo.getSelectedItem().toString();
 getIdDetails(Minimum_Year_of_Experience,Maximum_Year_of_Experience ,Specialization_
Area);
    }

    private void apply_buttonActionPerformed(java.awt.event.ActionEvent evt) {
      String uname= uname_txt.getText();
       String pass = pass_txt.getText();
          LoginDao ldao=new LoginDao();
           boolean flag=ldao.checkLogin(uname, pass);
           if(flag==true)
           {
              this.insertData();
              new Application_Frm().setVisible(true);
           }
           else
           {
              JOptionPane.showMessageDialog(this,"WRONG USERNAME & PASSWORD");
              new Index_frm().setVisible(true);
           }
    }
 private void search_tableMouseClicked(java.awt.event.MouseEvent evt) {
int row=search_table.getSelectedRow();
```

```
jobid_txt.setText(search_table.getValueAt(row,2).toString());
        jobname_txt.setText(search_table.getValueAt(row,3).toString());
          jobtype_txt.setText(search_table.getValueAt(row, 4).toString());
          min_txt1.setText(search_table.getValueAt(row, 5).toString());
          max_txt1.setText(search_table.getValueAt(row, 6).toString());
          specializationarea_txt1.setText(search_table.getValueAt(row, 7).toString());
          dateofvalid_txt.setText(search_table.getValueAt(row, 8).toString());
            qualification_txt.setText(search_table.getValueAt(row, 9).toString());
            cid_txt.setText(search_table.getValueAt(row, 0).toString());
            cname_txt.setText(search_table.getValueAt(row, 1).toString());
}
```

APPLICATION:

```
public class Application_Frm extends javax.swing.JFrame {
    Connection cn =null;
    Statement st= null;
    ResultSet rs=null;
    PreparedStatement ps=null;
    String select_sql="select * from application";
public void getData()
    {
        try
        {
            ConnectionFactory con= new ConnectionFactory();
            cn=con.getConn();
            st=cn.createStatement();
            rs=st.executeQuery(select_sql);
            application_table.setModel(DbUtils.resultSetToTableModel(rs));
        }
        catch(SQLException se)
                {
                    se.printStackTrace();
                }
    }
 private void formWindowOpened(java.awt.event.WindowEvent evt) {
        this.getData();
    }
```

IMPLEMENTATION

AND

TESTING

IMPLEMENTATION AND TESTING

A software system test plan is a document that describes the objectives, scope, approach and focus of software testing effort. The process of preparing a test plan is a usual way to think the efforts needed to validate the acceptability of a software product. The complete document will help people outside the test group understand the "WHY" and "HOW" product validation. It should be through enough to be useful but not so through that no one outside the test group will read it.

1. INTRODUCTION

Testing is the process of running a system with the intention of finding errors. Testing enhances the integrity of a system by detecting deviations in design and errors in the system. Testing aims at detecting error-prone areas. This helps in the prevention of errors in a system. Testing also adds value to the product by conforming to the user requirements.

The main purpose of testing is to detect errors and error-prone areas in a system. Testing must be thorough and well-planned. A partially tested system is as bad as an untested system. And the price of an untested and under-tested system is high.

The implementation is the final and important phase. It involves user-training, system testing in order to ensure successful running of the proposed system. The user tests the system and changes are made according to their needs. The testing involves the testing of the developed system using various kinds of data. While testing, errors are noted and correctness is the mode.

2. OBJECTIVES OF TESTING

The objective our test plan is to find and report as many bugs as possible to improve the integrity of our program. Although exhaustive testing is not possible, we will exercise a broad range of tests to achieve our goal. Our user interface to utilize these functions is designed to be user-friendly and provide easy manipulation of the tree. The application will only be used as a demonstration tool, but we would like to ensure that it could be run from a variety of platforms with little impact on performance or usability.

Process Overview

The following represents the overall flow of the testing process:

1. Identify the requirements to be tested. All test cases shall be derived using the current Program Specification.

2. Identify which particular test(s) will be used to test each module.

3. Review the test data and test cases to ensure that the unit has been thoroughly verified and that the test data and test cases are adequate to verify proper operation of the unit.

4. Identify the expected results for each test.

5. Document the test case configuration, test data, and expected results.

6. Perform the test(s).

7. Document the test data, test cases, and test configuration used during the testing process. This information shall be submitted via the Unit/System Test Report (STR).

8. Successful unit testing is required before the unit is eligible for component integration/system testing.

9. Unsuccessful testing requires a Bug Report Form to be generated. This document shall describe the test case, the problem encountered, it's possible cause, and the sequence of events that led to the problem. It shall be used as a basis for later technical analysis.

10. Test documents and reports shall be submitted. Any specifications to be reviewed, revised, or updated shall be handled immediately.

3. TEST CASES

A test case is a document that describe an input, action, or event and expected response, to determine if a feature of an application is working correctly. A test case should contain particular such as test case identifier, test condition, input data Requirement expected results. The process of developing test cases can help find problems in the requirement or design of an application, since it requires completely thinking through the operation of the application.

TESTING STEPS

UNIT TESTING

Unit testing focuses efforts on the smallest unit of software design. This is known as module testing. The modules are tested separately. The test is carried out during programming stage itself. In this step, each module is found to be working satisfactory as regards to the expected output from the module.

INTEGRATION TESTING

Data can be lost across an interface. One module can have an adverse effect on another, sub functions, when combined, may not be linked in desired manner in major functions. Integration testing is a systematic approach for constructing the program structure, while at the same time conducting test to uncover errors associated within the interface. The objective is to take unit tested modules and builds program structure. All the modules are combined and tested as a whole.

VALIDATION

At the culmination of the integration testing, Software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of software test begin in validation testing. Validation testing can be defined in many ways, but a simple definition is that the validation succeeds when the software functions in a manner that is expected by the customer. After validation test has been conducted, one of the three possible conditions exists.

a) The function or performance characteristics confirm to specification and are accepted.

b) A deviation from specification is uncovered and a deficiency lists is created.

c) Proposed system under consideration has been tested by using validation test and found to be working satisfactory.

Tested By:	**Sathi Nag, Ramit Das**
Test Type	Unit Testing
Test Case Number	1
Test Case Name	User Identification
Test Case Description	The user should enter his/ her accurate user id and password so that he/she can able to go for the further options. The test case will check the application for the same since a user can only login with the correct user id and password.

Item(s) to be tested	
1	Verification of the user id and password with the record in the database.

Specifications	
Input	**Expected Output/Result**
1) Correct User id and password **2) Incorrect Id or Password**	1) Successful login 2) Failure Message

Tested By:	**Sathi Nag, Ramit Das**
Test Type	Unit Testing
Test Case Number	2
Test Case Name	Job Searching
Test Case Description	A jobseeker when wants to job searching and comes in the searching page, he/she can see a login form. There first login id field is given. If jobseeker register their details in job portal then automatically, they can get there id by drop down. By selecting the id jobseeker can see their name generated in the name section. If jobseeker not registered in the portal then he/she can't see any job details.
Item(s) to be tested	
1	Required fields in the form are not empty, validation of proper credentials
Specifications	
Input	**Expected Output/Result**
1) User id, name, password, phone, address, email. **2) Not Registered in customer register**	1) Successful registration 2) Booking is not completed.

4. WHITE BOX TESTING

In white box testing, the UI is bypassed. Inputs and outputs are tested directly at the code level and the results are compared against specifications. This form of testing ignores the function of the program under test and will focus only on its code and the structure of that code. Test case designers shall generate cases that not only cause each condition to take on all possible values at least once, but that cause each such condition to be executed at least once. To ensure this happens, we will be applying Branch Testing. Because the functionality of the program is relatively simple, this method will be feasible to apply.

Each function of the binary tree repository is executed independently; therefore, a program flow for each function has been derived from the code.

5. BLACK BOX TESTING

Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end-user would. We have decided to perform Equivalence Partitioning and Boundary Value Analysis testing on our application.

System Testing

The goals of system testing are to detect faults that can only be exposed by testing the entire integrated system or some major part of it. Generally, system testing is mainly concerned with areas such as performance, security, validation, load/stress, and configuration sensitivity. But in our case well focus only on function validation and performance. And in both cases, we will use the black-box method of testing.

6. OUTPUT TESTING

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in a specific format. The output format on the screen is found to be correct. The format was designed in the system design time according to the user needs. For the hardcopy also; the output comes as per the specified requirements by the user. Hence output testing did not result in any correction for the system.

7.USER ACCEPTANCE TESTING

User acceptance of a system is the key factor for the success of any system. The system under consideration is tested for the user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes whenever required.

This is done in regard to the following point:

a) Input Screen Design

b) Output Screen Design

c) Format of reports and other outputs.

8. INTEGRATION TESTING

Software testing is always used in association with verification and validation. In the testing phase of this project our aim is to find the answer to following two questions.

- Whether the software matches with the specification (i.e. process base) to verify the product.

- Whether this software in one client what wants (i.e. product base) to validate the product.

- Unit testing and integration testing has been carried out to find the answer to above questions. In unit testing each individual module was test to find any unexpected behaviour if exists. Later all the module was integrated and flat file was generated.

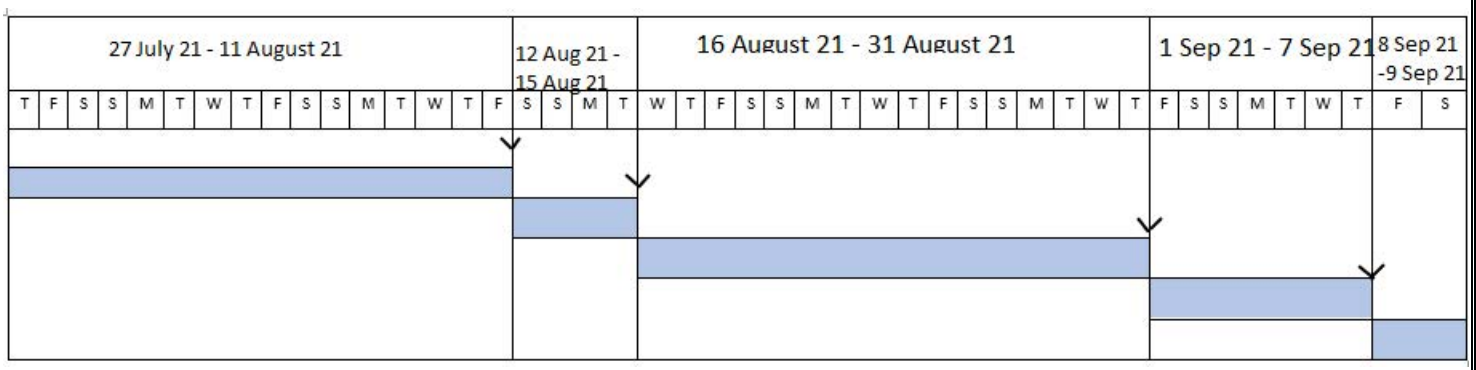
FUNCTIONAL TESTING

These are the points concerned during the stress test:

- Nominal input: character is in putted in the place of digits and the system has to flash the message "Data error"
- Boundary value analysis: exhaustive test cases have designed to create an output report that produces the maximum (and minimum) allowable number of table entries.

7. GANTT CHART

TASK NAME	DURATION	START	FINISH
1. Requirement Gathering	16 days	27-JULY-21	11-AUGUST-21
2. Project Planning	4 days	12-AUGUST-21	15-AUGUST-21
3. Coding	16 days	16-AUGUST-21	31-AUGUST-21
4. Design	7 days	1-SEPTEMBER-21	7-SEPTEMBER-21
5. Testing	2 days	8-SEPTEMBER-21	9-SEPTEMBER-21
TOTAL	45 days	27-JULY-21	9-SEPTEMBER-21



SOFTWARE SYSTEM ATTRIBUTES

1. DATABASE SECURITY

System security measure is meant to be provided to make your system reliable and secured from unauthorized user may create threats to the system. So you should follow some security measures. We have used security levels in database level at system level.

2. SYSTEM SECURITY

If we talk about the system security in our proposed system we have implemented with the help of maintain the session throughout the system's use. Once a user has logged out than he/she will not be able to perform any task before signing back again. A high level of authentic login is given to the system so this is a very tedious task to enter without authorization and authentication.

3. LIMITATIONS

- Since it is an online project, customers need internet connection to use it.
- People who are not familiar with computers can't use this software.
- Customer must have debit card or credit card to book car.

4. CONCLUSION

This project has been appreciated by all the users in the organization. It is easy to use, since it uses the GUI provided in the user dialog. User friendly screens are provided. The usage of software increases the efficiency, decreases the effort. It has been efficiently employed as a Site management mechanism. It has been thoroughly tested and implemented.

The project **"JOB PORTAL SYSTEM"** *is the use of control systems and information technologies to reduce the need for human work in the production of goods and services*. The wider area of job searching facilitate the quick and easy access to opportunities the increasing job opportunities and changing scenario of the business environment today has made more people to search for better career and employers to search for better potential.

5. FUTURE SCOPE AND FURTHER ENHANCEMENTS

The future of the job portals looks bright. Job portals are like the meeting points for the recruiters as well as the job seekers. Most of the jobseekers who're actively seeking new employment opportunities are believed to be registered on multiple job portals. Finding a job opportunity per your choice and qualification through a job portal is relatively easier. But the sourcing and operations process often takes a long time. Many use an employment agency to expedite this process. Recruitment firms usually do not charge a single penny from the candidates. But due to the Coronavirus Situation, most of the placement agencies are currently out of business. Many placement agencies are bringing their services online. In the short term, we will likely see a wave of new job portals, but just as in most markets, this initial influx will see various winners and losers. This will anyway benefit the job seekers.

6. REFERENCES

1. https://www.w3schools.com
2. https://www.slideshare.com
3. https://www.scribd.com
4. https://
   www.tutorialspoint.com
5. https://www.Javatpoint.com

THANK YOU