SYNOPSIS

Corporate Recruitment System (CRS) is web-based tool to reduce communication gap between Job providers and Job seekers.

The Main objective of this solution is to make easy the recruitment process of any organization. This CRS is designed by keeping in mind both parties Job providers and Job seekers. CRS allows Job seekers to register their details like skills and experience with the system, and then on the other hand even it allows job providers to post their requirements with the system.

Corporate Recruitment Management system is helpful for the job providers i.e. companies which are in need of employees, job seekers who are in need of job, (for both Exp and freshers). This portals main aim is to provide the vacancies available for the job seekers without taking any charge from them in IT technologies.CRS will automatically send mails to all job seekers whose skills are matched with the requirement.

Features :

- This project can be used very easily in the process of decision making in new recruitments.
- Effective way of providing communication between job providers and job seekers.
- Reliable and consistent way of searching jobs.
- Conducting secured and restricted online exam for screened employees.
- Sending Email notification to all job seekers.

PROJECT DESCRIPTION

The Project titled “Corporate Recruitment System” is one which helps out the HR Personal in the recruitment of new candidates to the company and helps the recruitment process as a whole.

Corporate recruitment System (CRS) is a part of the Human Resource Management System that structures and manages the entire recruitment process. This corporate recruitment service system will primarily focus on the posting and management of job vacancies. However, this will be the initial step towards achieving the long term goal of delivering broader services to support recruitment.

This will provide service to the potential job applicants to search for working opportunities and if they choose they may be able to make an application online. It is planned that ultimately all vacancies will be posted online and that this site will offer employers the facilities both to post their vacancies online and to review and manage the resulting applications efficiently through the web with the help of the CRS. CRS will allow job provider to establish one to one relationships with candidates, by keeping in close communications with them through out the application, interview, and hiring process. It even allows the candidates to track the progress of their application.

In other words, enables the employer to treat candidates like customers.

Features:-

Corporate Recruitment System (CRS) has all the features and functions required for executing a successful recruitment task, providing exceptional case of use for recruitment.

The Following are the overview of the features and benefits of CRS.

1. Database software installed and pre-configures for the immediate use of the system effectively and efficiently.
2. Pre-configured and ready to run Jobs database with management module for adding and deleting efficiently.
3. Database to store the candidate's details securely.
4. Customizable authentication to control access to database files using assigned user login and password control.
5. Provides information to the managers so that they can make judgment about particular situations.
6. Reductions in the cost of hiring – there will be between 50-60 percent decrease in the cost of hiring.
7. Reduces the time required to complete the recruitment process of any organization.

Objective:-

1. This system provides service to the potential job applicants to search for working opportunities.

2. This system helps the HR Personal in the recruitment of new candidates to the company.

3. Corporate Recruitment System will allow job provider to establish one to one relationships with candidates.

4. This corporate recruitment service system will primarily focus on the posting and management of job vacancies.

5. This system is designed such that ultimately all vacancies will be posted online and would offer employers the facilities to post their vacancies online.

6. It helps to review and manage the resulting applications efficiently through the web.

7. It even allows the candidates to track the progress of their application.

User Requirements

1. To create a database to store the candidate's details securely.

2. To reduce the cost of hiring of new candidates to the company.

3. To reduce the time required to recruit the new employees.

4. Helps to provide control access to database files using assigned user login and password control.

MODULES DESCRIPTION

Recruiter - Creating the searching jobs for web sites in users.

Job Search – The employees are many types of searching the jobs.

Log in- The user can login our details to use the security password.

Job Provider- Publishing the number of vacancies our companies in HR personal and placing the details are privacy.

Job Seeker- The seeker can applying our platform job and registers now.

RECRUITER HOME PAGE:

The home is located in the root directory of an web sites , this page is the main page of out in application.

LOGIN:

This page allows the authorized user to login our application only authorized can through login and proceed further options in our application.

JOB SEARCH:

Job search are used to manage the technologies on various searching criteria.

JOB PROVIDER:

We can create the jobs for create , read , update , and delete
are available the application.

SYSTEM STUDY & FEASIBILITY ANALYSIS

Having an established system, one has to determine whether an alternative system is feasible compared to existing system. An analysis of the ability to complete a project successfully, taking into account legal, economic, technological, scheduling and other factors. Feasibility study is conducted in two-step; first, a project team is formed. The team develop system flowchart that identify the characteristics of the existing system, evaluate the performance of the system, cost data, and select the best system for the job.

It is the most frequently used method for evaluating the effectiveness of a candidate system. If benefits out weight cost, then the decision is made to design and implement the system, thus this ongoing that improves in accuracy. Thus to pay a less cost of effort will be need for this project.

It centers on the existing computer system and to what extent it can support the proposed addition. This involves financial consideration to accommodation technical feasibility. The configuration is very enough to allow tIhis project.

The purpose of Operational Feasibility study is to determine whether then process the new system will be used if it is developed and implemented or will there be resistance from the users that will take the possible application benefits.

The execution of managed code, thereby creating a software environment that can exploit both managed and unmanaged features.

EXISTING SYSTEM

The existing system has some drawbacks, that we are gone to develop a new system in visual basic 6.0. in the existing system, the problem is maintained computerized using COBOL, so there are many problems in a maintaining the system . The manpower is considerably wasted in finding the records and the maintaining the records. Major problem is wastages of money for even a simple task.In maintaining the inventory many problem is there. Among it major problem maintaining the record. And keep in track of the records, which are needs of tenthly for everything is waste .

PROPOSED SYSTEM

To overcome the limitation in the existing system, it is decided to develop a new system in visual Basic 6.0 as front end and oracle 8.0 as back end.

Manipulation of lager volume of data, retrieval of old data, identification of specified data is easily processed through the proposed system. Since the hardware requirement for implementation of the system is available; the proposed system can be implementation easily, there by providing a way for less cost, better utilization and flexibility

HARDWARE & SOFTWARE SPECIFICATION

Hardware Requirements:

CPU type	: Intel core i3
Ram size	: 6 GB
Hard disk capacity	: 256 GB

Software Requirements:

Operating System	: Windows 10
Language	: ASP.NET
IDE	: Visual Studio 10.0 (2010)
Back End	: MY SQL 2010

SOFTWARE DESCRIPTION

ASP.NET

Server Application Development

Server-side applications in the managed world are implemented through runtime hosts. Unmanaged applications host the common language runtime, which allows your custom managed code to control the behavior of the server. This model provides you with all the features of the common language runtime and class library while gaining the performance and scalability of the host server.

The following illustration shows a basic network schema with managed code running in different server environments. Servers such as IIS and SQL Server can perform standard operations while your application logic executes through the managed code.

SERVER-SIDE MANAGED CODE

ASP.NET is the hosting environment that enables developers to use the .NET Framework to target Web-based applications. However, ASP.NET is more than just a runtime host; it is a complete architecture for developing Web sites and Internet-distributed objects using managed code. Both Web Forms and XML Web services use IIS and ASP.NET as the publishing mechanism for applications, and both have a collection of supporting classes in the .NET Framework. Navigator. Instead, XML Web services consist of reusable software components designed to be consumed by other applications, such as traditional client

DEVELOPMAENT TOOLS & TECHNOLOGIES:

Applications, Web-based applications, or even other XML Web services. As a result, XML Web services technology is rapidly moving application development and deployment into the highly distributed environment of the Internet.

Power and Flexibility Because ASP.NET is based on the common language runtime, the power and flexibility of that entire platform are available to Web application developers. The .NET Framework class library, Messaging, and Data Access solutions are all seamlessly accessible from the Web. ASP.NET is also language-independent, so you can choose the language that best applies to your application or partition your application across many languages. Further, common language runtime interoperability guarantees that your existing investment in COM-based development is preserved when migrating to ASP.NET.

the ASP.NET page framework allows you to build user interfaces that cleanly separate application logic from presentation code and to handle events in a simple, Visual Basic - like forms processing model. Additionally, the common language runtime simplifies development, with managed code services such as automatic reference counting and garbage collection.

WHAT IS ASP.NET WEB FORMS:

Manage ability ASP.NET employs a text-based, hierarchical configuration system, which simplifies applying settings to your server environment and Web applications. Because configuration information is stored as plain text, new settings may be applied without the aid of local administration tools.

Scalability and Availability. ASP.NET has been designed with scalability in mind, with features specifically tailored to improve performance in clustered and multiprocessor environments. Further, processes are closely monitored and managed by the ASP.NET runtime, so that if one misbehaves (leaks, deadlocks), a

INTRODUCTION TO ASP.NET SERVER CONTROL:

New process can be created in its place, which helps keep your application constantly available to handle request.

Customizability and Extensibility ASP.NET delivers a well-factored architecture that allows developers to "plug-in" their code at the appropriate level. In fact, it is possible to extend or replace any subcomponent of the ASP.NET runtime with your own custom-written component. Implementing custom authentication or state services has never been easier.

Security. With built-in Windows authentication and per-application configuration, you can be assured that your applications are secure.

ASP.NET MANAGED CODE:

Database management, or DBMS, gives the user access to their data and helps them transform the data into information. Such database management systems include dBase, paradox, IMS, SQL Server and SQL Server. These systems allow users to create, update and extract information from their database.

A database is a structured collection of data. Data refers to the characteristics of people, things and events. SQL Server stores each data item in its own fields. In SQL Server, the fields relating to a particular person, thing or event are bundled together to form a single complete unit of data, called a record..

DATA ADAPTERS IN DATA:

Web Services Description Language tool included with SDK can query an XML Web service published on the Web, parse its WSDL description, and produce C# or Visual Basic source code that your application can use to become a client of the XML Web service. The source code can create classes derived from classes in the class library that handle all the underlying communication using SOAP and XML parsing. Although you can use the class library to consume XML Web services directly, the Web Services Description Language tool and the other tools contained in the SDK facilitate your development efforts .

If you develop and publish your own XML Web service, the provides a set of classes that conform to all the underlying communication standards, such as SOAP, WSDL, and XML. Using those classes enables you to focus on the logic of your service, without concerning yourself with the communications infrastructure required by distributed software development.

ACTIVE SERVER PAGES.NET

ASP.NET is a programming framework built on the common language runtime that can be used on a server to build powerful Web applications. ASP.NET offers several important advantages over previous Web development models:Enhanced Performance. ASP.NET is compiled common language runtime code running on the server. Unlike its interpreted predecessors, ASP.NET can take advantage of early binding, just-in-time compilation, native optimization, and caching services right out of the box. This amounts to dramatically better performance before you ever write a line of code.

World-Class Tool Support:. The ASP.NET framework is complemented by a rich toolbox and designer in the Visual Studio integrated development environment. WYSIWYG editing, drag-and-drop server controls, and automatic deployment are just a few of the features this powerful tool provides

Power and Flexibility: Because ASP.NET is based on the common language runtime, the power and flexibility of that entire platform are available to Web application developers. The .NET Framework class library, Messaging, and Data Access solutions are all seamlessly accessible from the Web. ASP.NET is also language-independent, so you can choose the language that best applies to your application or partition your application across many languages. Further, common language runtime interoperability guarantees that your existing investment in COM-based development is preserved when migrating to ASP.NET.

Simplicity: ASP.NET makes it easy to perform common tasks, from a simple form of submission and client authentication to deployment and site configuration. For example, the ASP.NET page framework allows you to build user interfaces that cleanly separate application logic from presentation code and to handle events in a simple, Visual Basic - like forms processing model. Additionally, the common language runtime simplifies development, with managed code services such as automatic reference counting and garbage collection.

Manageability. ASP.NET employs a text-based, hierarchical configuration system, which simplifies applying settings to your server environment and Web.

local administration philosophy extends to deploying ASP.NET Framework applications as well. An ASP.NET Framework application is deployed to a server simply by copying the necessary files to the server. No server restart is required, even to deploy or replace running compiled code.

Scalability and Availability. ASP.NET has been designed with scalability in mind, with features specifically tailored to improve performance in clustered and multiprocessor environments. Further, processes are closely monitored and managed by the ASP.NET runtime, so that if one misbehaves (leaks, deadlocks), a new process can be created in its place, which helps keep your application constantly available to handle requests.

Customizability and Extensibility. ASP.NET delivers a well-factored architecture that allows developers to "plug-in" their code at the appropriate level. In fact, it is possible to extend or replace any subcomponent of the ASP.NET runtime with your own custom-written component. Implementing custom authentication or state services has never been easier.

Security. With built-in Windows authentication and per-application configuration, you can be assured that your applications are secure.

Enhanced Performance. ASP.NET is compiled common language runtime code running on the server. Unlike its interpreted predecessors, ASP.NET .

The ASP.NET framework is complemented by a rich toolbox and designer in the Visual Studio integrated development environment. WYSIWYG editing, drag-and-drop server controls, and automatic deployment.

OVER VIEW OF SQL SERVER

Database management, or DBMS, gives the user access to their data and helps them transform the data into information. Such database management systems include dBase, paradox, IMS, SQL Server and SQL Server. These systems allow users to create, update and extract information from their database.

A database is a structured collection of data. Data refers to the characteristics of people, things and events. SQL Server stores each data item in its own fields. In SQL Server, the fields relating to a particular person, thing or event are bundled together to form a single complete unit of data, called a record (it can also be referred to as raw or an occurrence). Each record is made up of a number of fields. No two fields in a record can have the same field name

During an SQL Server Database design project, the analysis of your business needs identifies all the fields or attributes of interest. If your business needs change over time, you define any additional fields or change the definition of existing fields.

SERVER TABLES

SQL Server stores record relating to each other in a table. Different tables are created for the various groups of information. Related tables are grouped together to form a database.

PRIMARY KEY

Every table in SQL Server has a field or a combination of fields that uniquely identifies each record in the table. The Unique identifier is called the Primary Key, or simply the Key. The primary key provides the means to distinguish one record.

RELATIONAL DATABASE

Sometimes all the information of interest to a business operation can be stored in one table. SQL Server makes it very easy to link the data in multiple tables. Matching an employee to the department in which they work is one example. This is what makes SQL Server a relational database management system, or RDBMS. It stores data in two or more tables and enables you to define relationships between the table and enables you to define relationships between the tables.

SERVER TABLES

SQL Server stores record relating to each other in a table. Different tables are created for the various groups of information. Related tables are grouped together to form a database.

PRIMARY KEY

Every table in SQL Server has a field or a combination of fields that uniquely identifies each record in the table. The Unique identifier is called the Primary Key, or simply the Key. The primary key provides the means to distinguish one record from all other in a table. It allows the user and the database system to identify, locate and refer to one particular record in the database.

RELATIONAL DATABASE

Sometimes all the information of interest to a business operation can be stored in one table. SQL Server makes it very easy to link the data in multiple tables. Matching an employee to the department in which they work is one example. This is what makes SQL Server a relational database management system, or RDBMS. It stores data in two or more tables and enables you to define relationships.

DISADVANTAGES OF DBMS

A significant disadvantage of the DBMS system is cost. In addition to the cost of purchasing of developing the software, the hardware has to be upgraded to allow for the extensive programs and the workspace required for their execution and storage. While centralization reduces duplication, the lack of duplication requires that the database be adequately backed up so that in case of failure the data can be recovered.

FEATURES OF SQL SERVER (RDBMS)

SQL SERVER is one of the leading database management systems (DBMS) because it is the only Database that meets the uncompromising requirements of today's most demanding information systems. From complex decision support systems (DSS) to the most rigorous online transaction processing (OLTP) application, even application that requires simultaneous DSS and OLTP access to the same critical data, SQL Server leads the industry in both performance and capabilitySQL SERVER is a truly portable, distributed, and open DBMS that delivers unmatched performance, continuous operation and support for every database.SQL Server RDBMS is high-performance fault-tolerant DBMS which is specially designed for online transactions processing and for handling large database application.

FOREIGN KEY

When a field is one table matches the primary key of another field is referred to as a foreign key. A foreign key is a field or a group of fields in one table whose values match those of the primary key of another table.

DATA ABSTRACTION

A major purpose of a database system is to provide users with an abstract view of the data. This system hides certain details of how the data is stored and maintained. Data abstraction is divided into three levels. Physical level: This is the lowest level of abstraction at which one describes how the data are actually stored. Conceptual Level: At this level of database abstraction all the attributed and what data are actually stored is described and entries and relationship among them.

View level: This is the abstraction at which one describes only part of the database.

ADVANTAGES OF RDBMS

• Redundancy can be avoided

• Inconsistency can be eliminated

• Data can be Shared

• Standards can be enforced

• Integrity can be maintained

• Conflicting requirements can be balanced

DISADVANTAGES OF DBMS

A significant disadvantage of the DBMS system is cost. In addition to the cost of purchasing of developing the software, the hardware has to be upgraded to allow for the extensive programs and the workspace required for their execution and storage. While centralization reduces duplication, the lack of duplication requires that the database be adequately backed up so that in case of failure the data can be recovered.

OBJECTIVES:

This system provides service to the potential job applicants to search for working opportunities.
This system helps the HR Personal in the recruitment of new candidates to the company.
Corporate Recruitment System will allow job provider to establish one to one relationships with candidates.
This corporate recruitment service system will primarily focus on the posting and management of job vacancies.
This system is designed such that ultimately all vacancies will be posted online and would offer employers the facilities to post their vacancies online.
It helps to review and manage the resulting applications efficiently through the web.

It control the behavior of the server. This model provides you with all the features of the common language runtime and class library while gaining the performance and scalability of the host server.

\

SYSTEM DESIGN

CLASS DIAGRAM

A class diagram is an illustration of the relationships and source code dependencies among classes in the Unified Modeling Language (UML). In this context, a class defines the methods and variables in an object, which is a specific entity in a program or the unit of code representing that entity. Class diagrams are useful in all forms of object-oriented programming (OOP). The concept is several years old but has been refined as OOP modeling paradigms have evolved.

SEQUENCE :

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

COLLABORATION: Like sequence diagrams, collaboration diagrams are also interaction diagrams. Collaboration diagrams convey the same information as sequence diagrams but focus on object roles instead of the times that messages are sent. In a sequence diagram, object roles are the vertices and messages are the connecting links. In a collaboration diagram, as follows, the object-role rectangles are labeled with either class or object names (or both). Colons precede the class names (:). B) DFD: A data flow diagram (dfd) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. A dfd is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated. dfds can also be used for the visualization of data processing (structured design).

DATA FLOW DIAGRAMS:

There are two types of DFD's they are

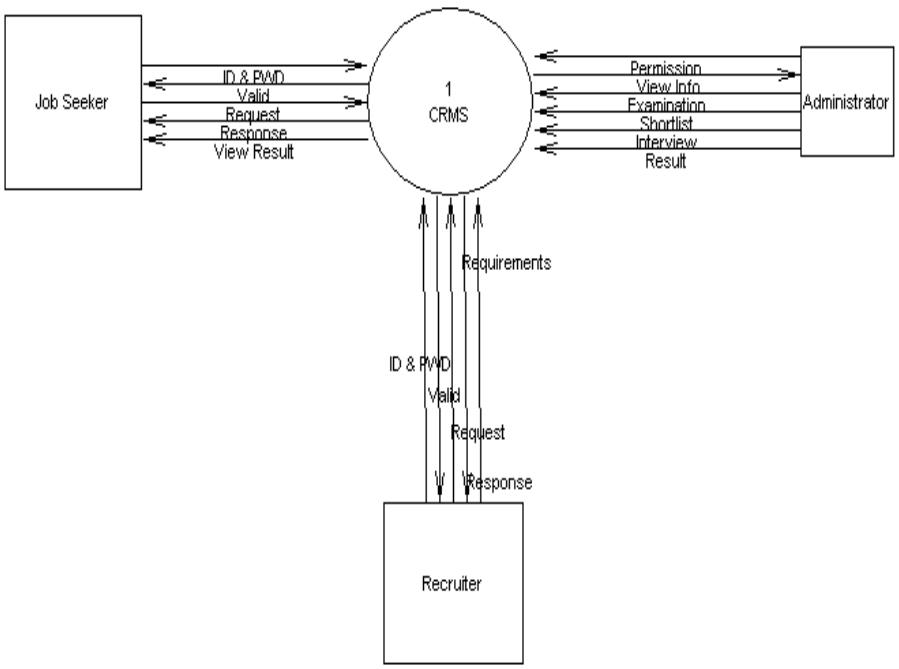
Context Level DFD

Top Level DFD

Context Level DFD:

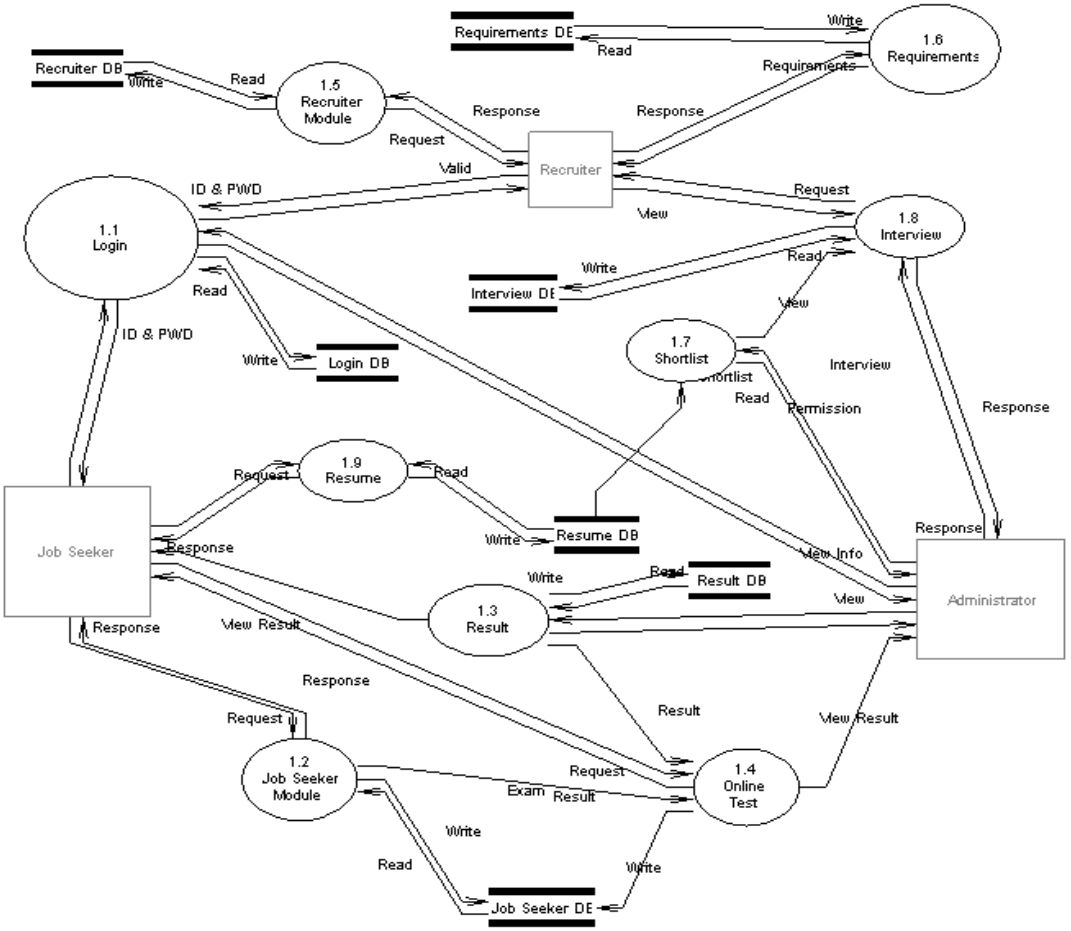
In the Context Level the whole system is shown as a single process.

- No data stores are shown.
- Inputs to the overall system are shown together with data sources (as External entities).
- Outputs from the overall system are shown together with their destinations (as External entities).



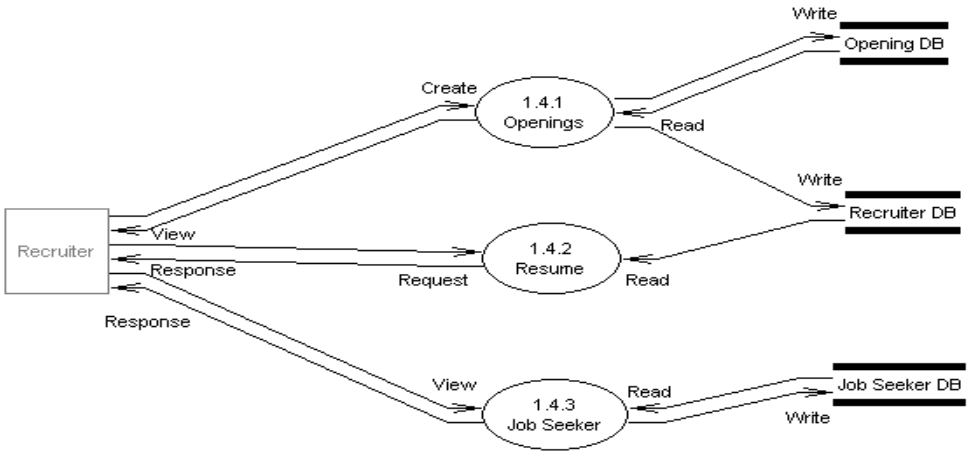
Top Level DFD:

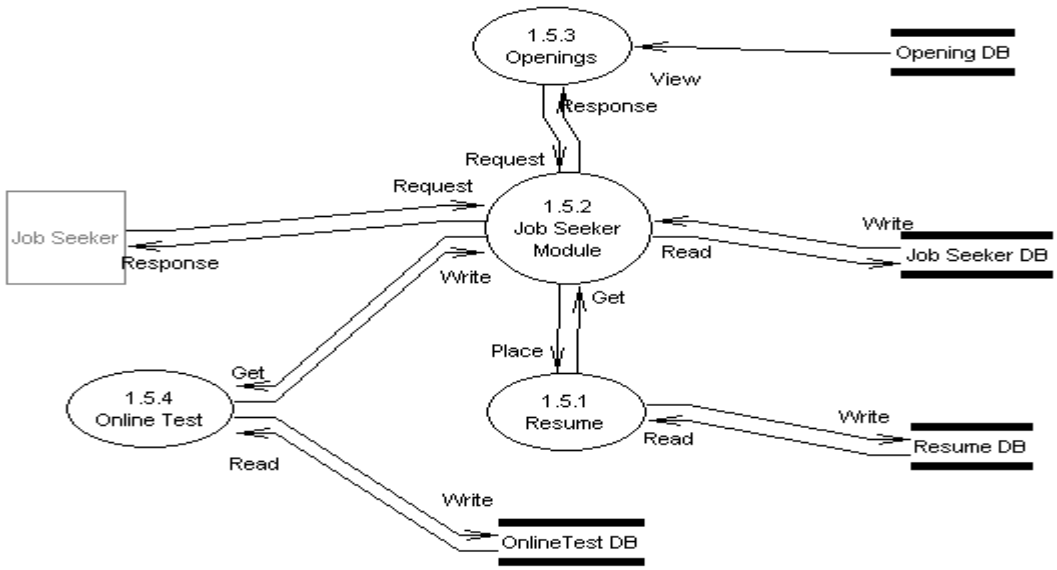
The Top Level DFD gives the overview of the whole system identifying the major system processes and data flow. This level focuses on the single process that is drawn in the context diagram by 'Zooming in' on its contents and illustrates what it does in more detail.



Detailed Level DFD:

In Detailed D.F.Ds the main process is divided into sub processes and we try to find out the flow from one process to another process. We find the interaction among External entities, processes, sub processes and database.





UML Diagrams

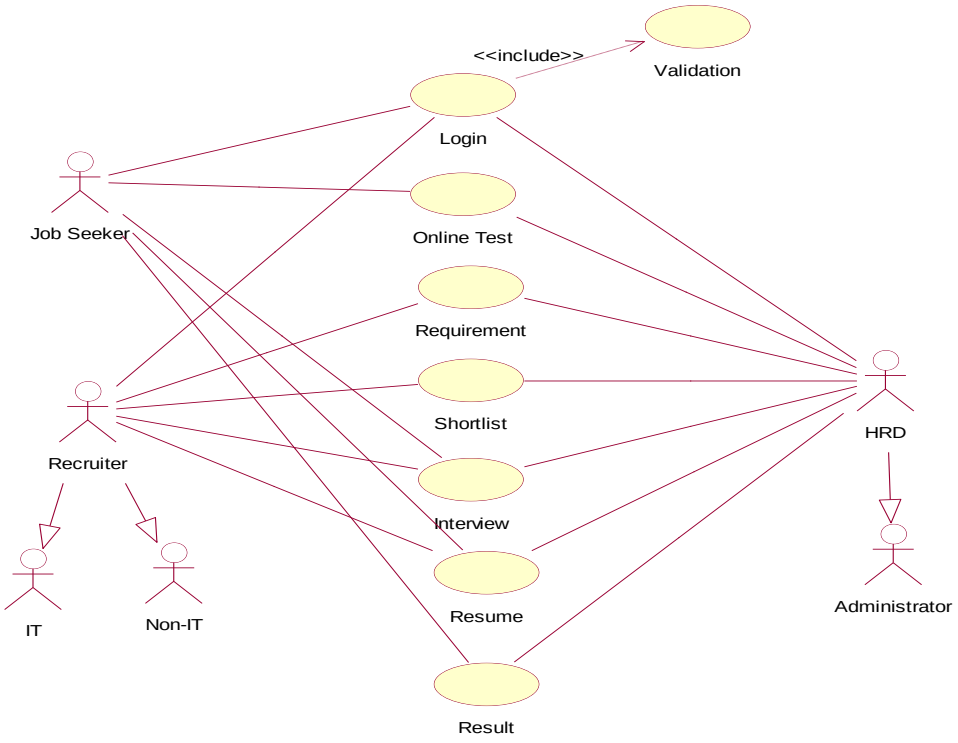
USE CASE DIAGRAM

- A use case diagram is a diagram that shows a set of use cases and actors and relationships.

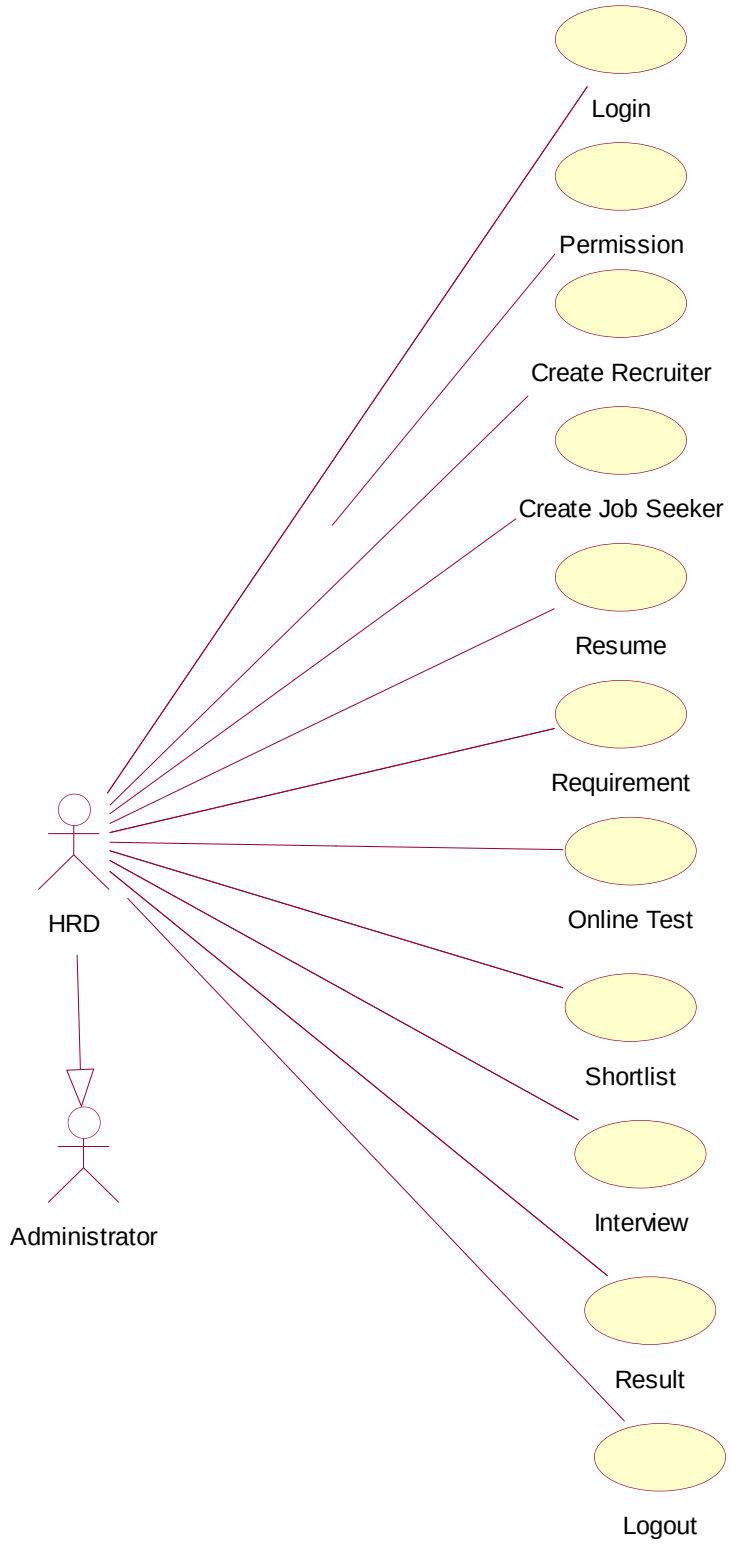
Contents

- Use case commonly contain
  - Use cases
  - Actors
  - Dependency, generalization and association relationships

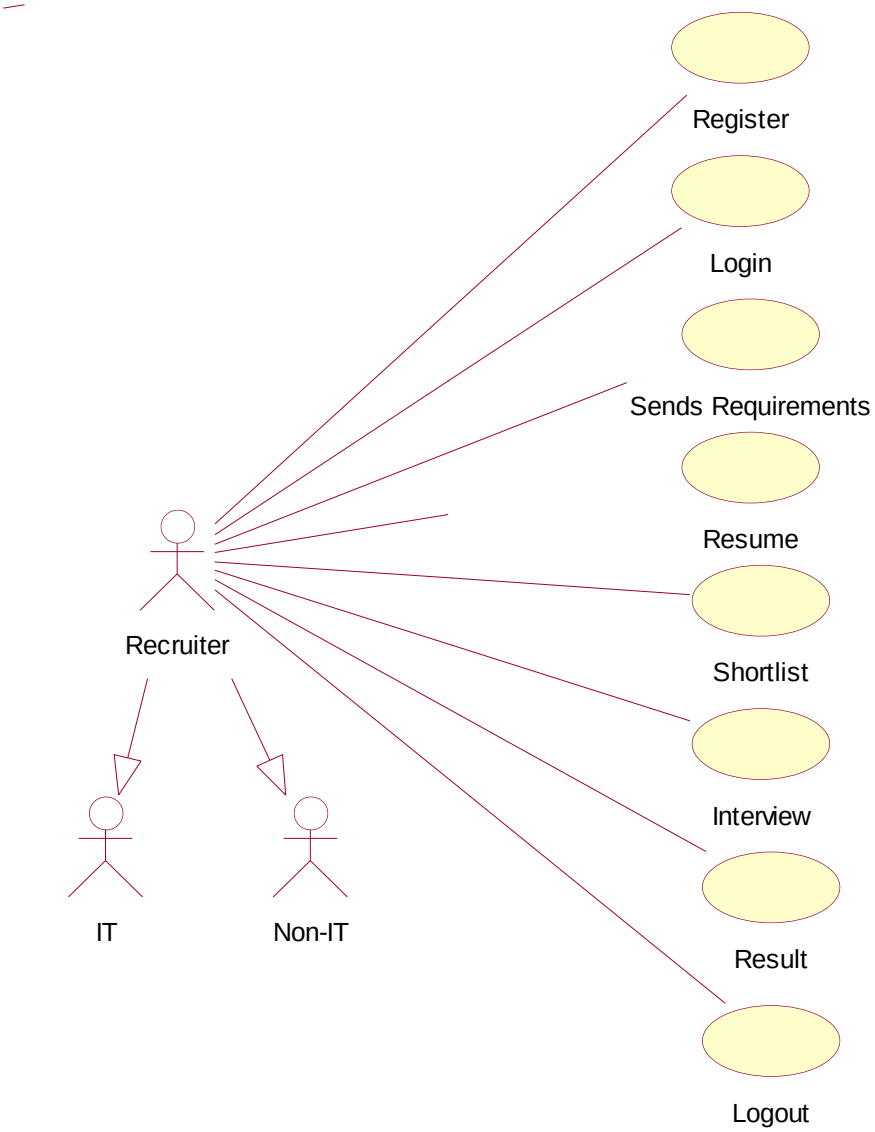
Over all Use Case



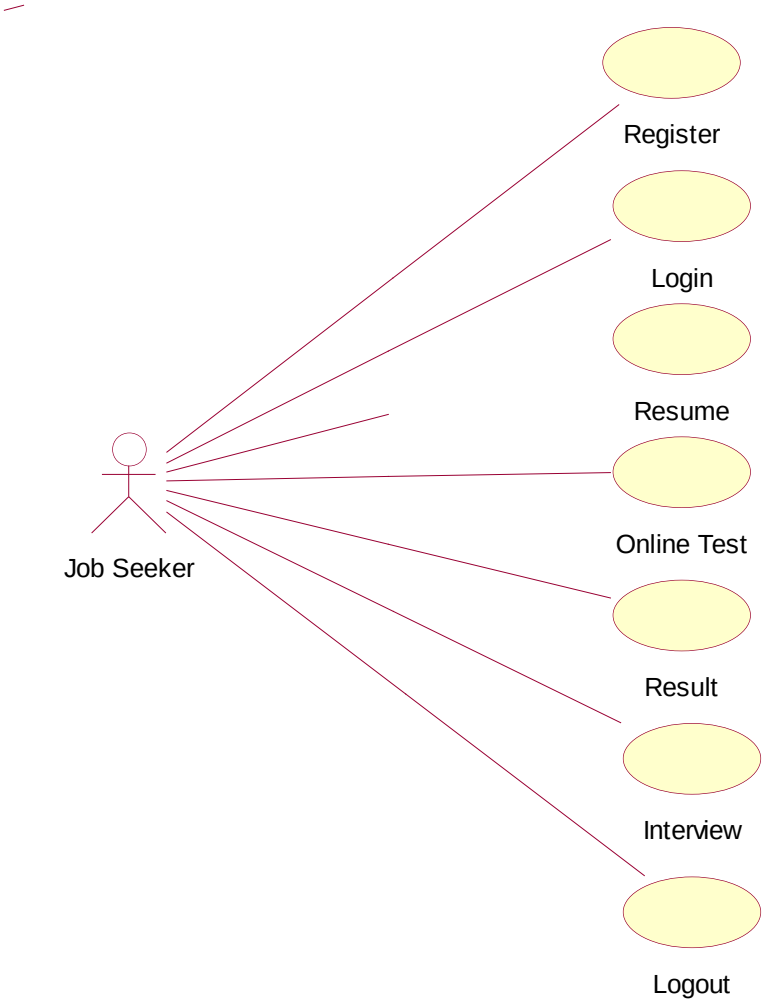
Administrator Use Case



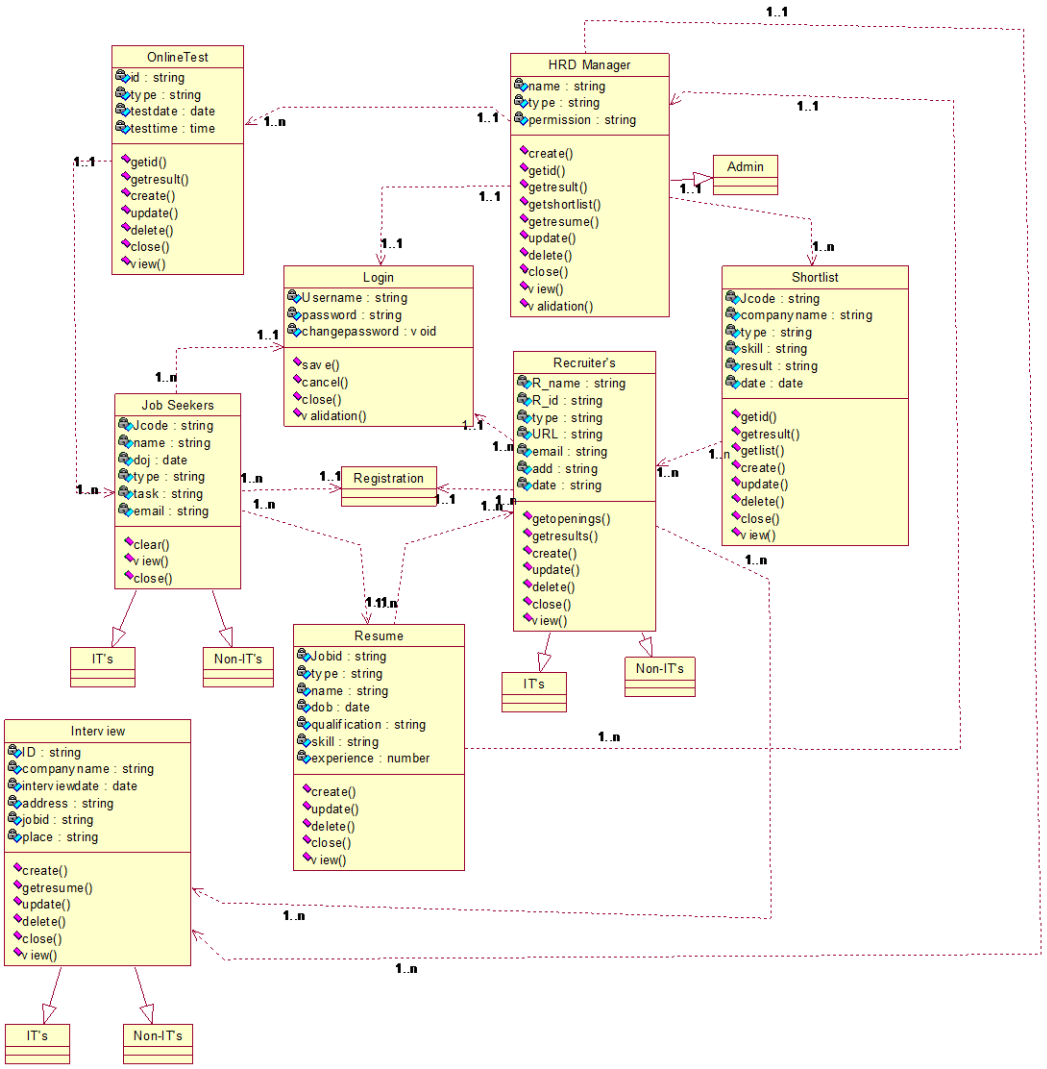
Recruiter Use Case



Job Seeker Use Case



Class Diagram



ACTIVITY DIAGRAM:

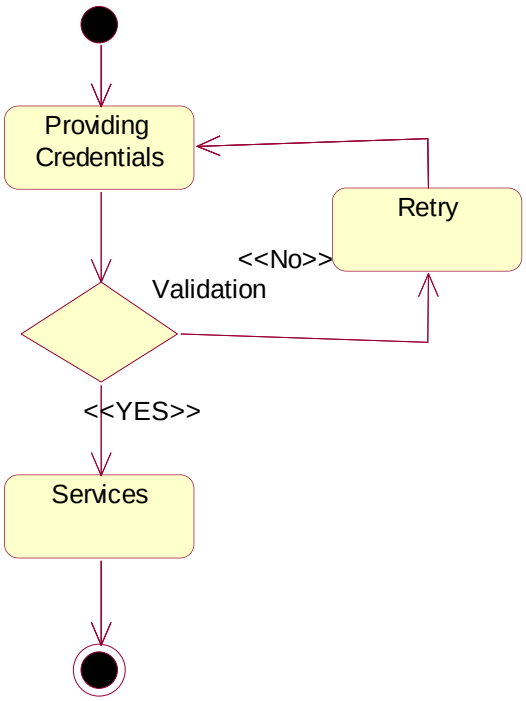
- An activity diagram shows the flow from activity to activity. An activity is an ongoing non-atomic execution within a state machine.
- Activities ultimately result in some action, which is made up of executable atomic computations that result in a change in state of the system or the return of a value.

Activity diagrams commonly contain

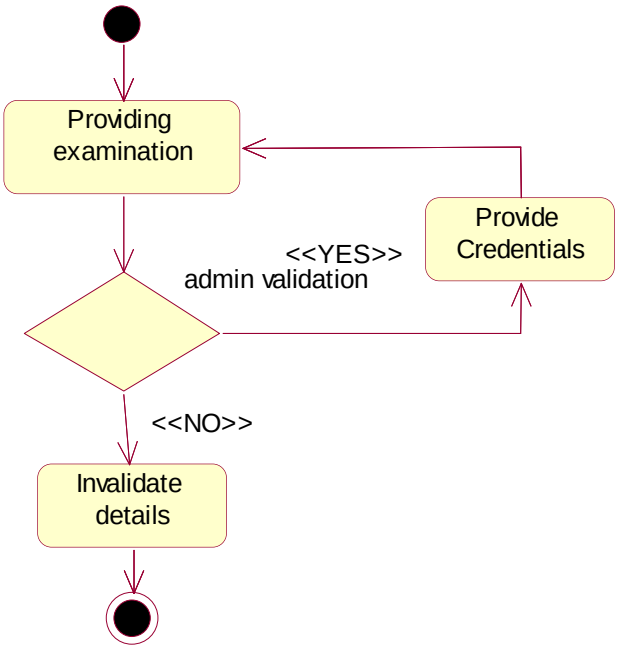
- Activity states and action states
- Transitions
- Objects

Like all other diagrams, activity diagrams may contain notes and constrains.

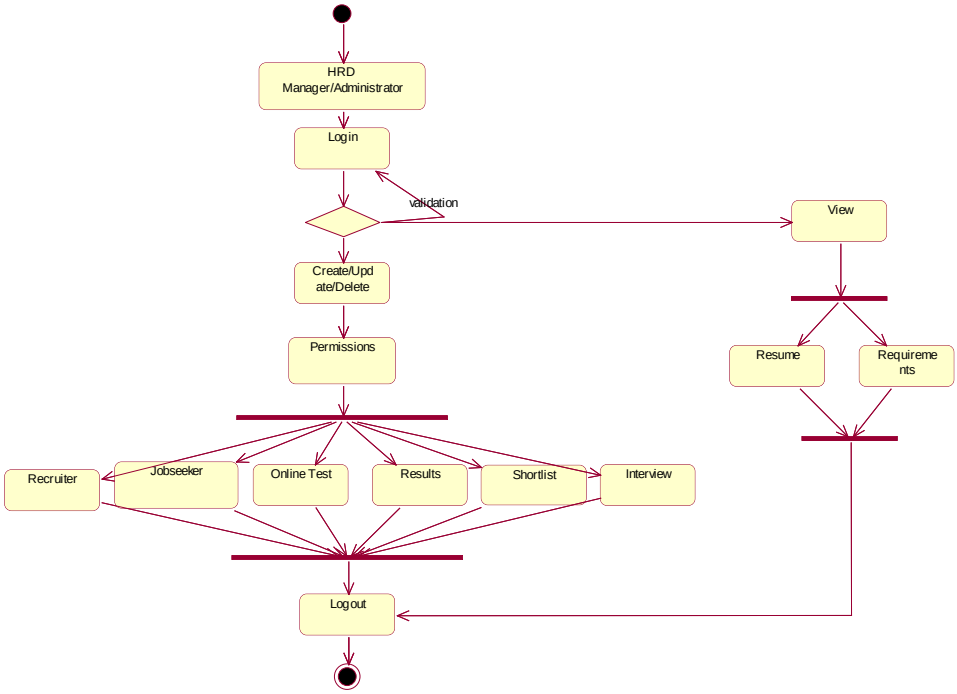
Login Process



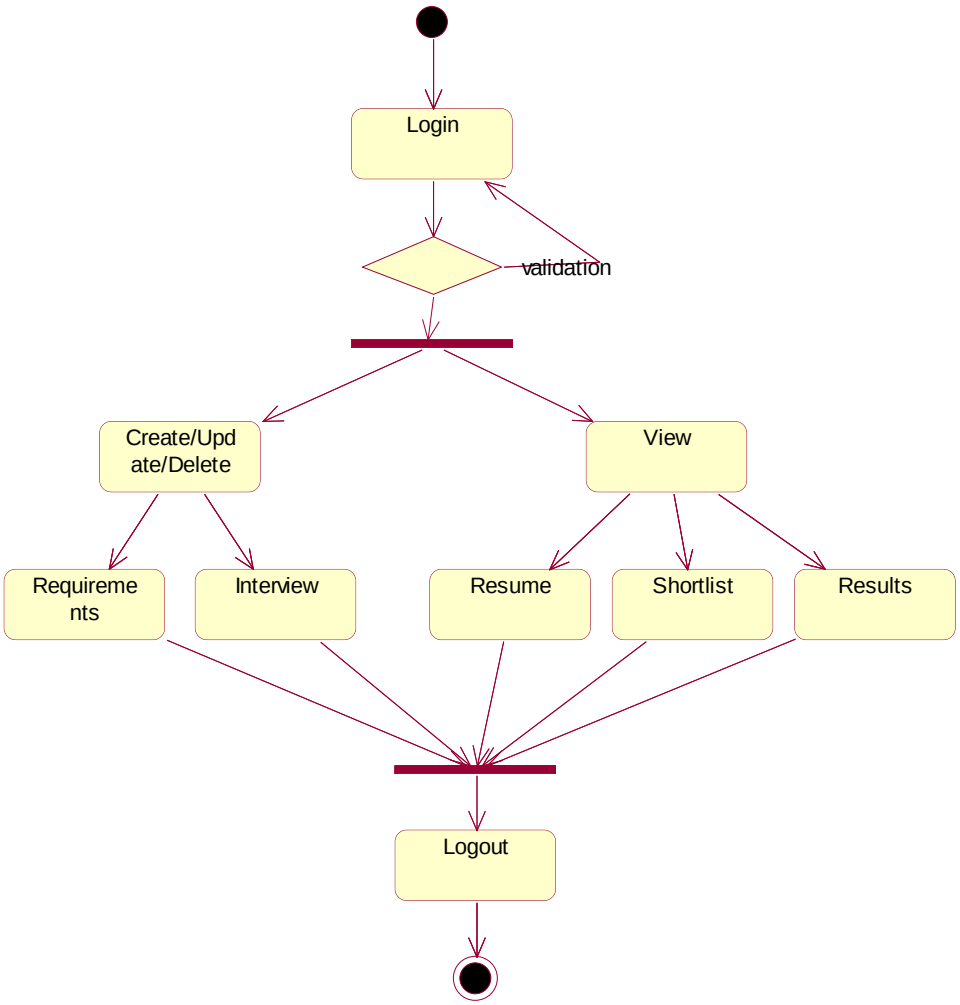
Registration Process



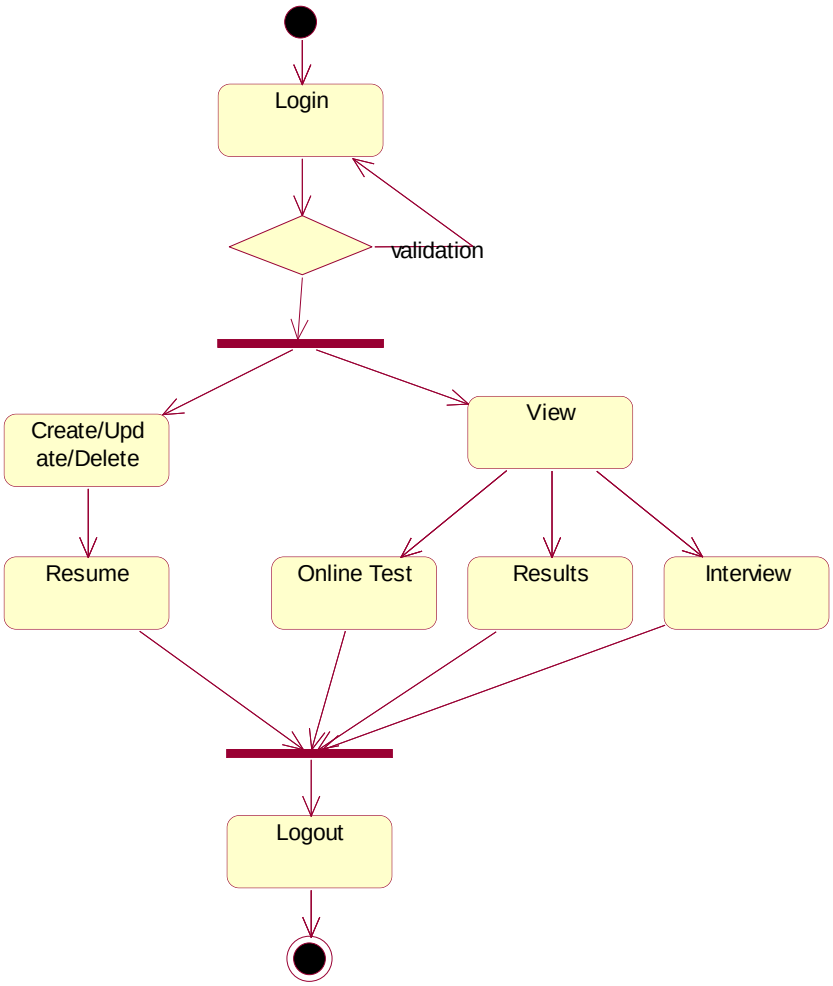
Administrator Process



Recruiter Process

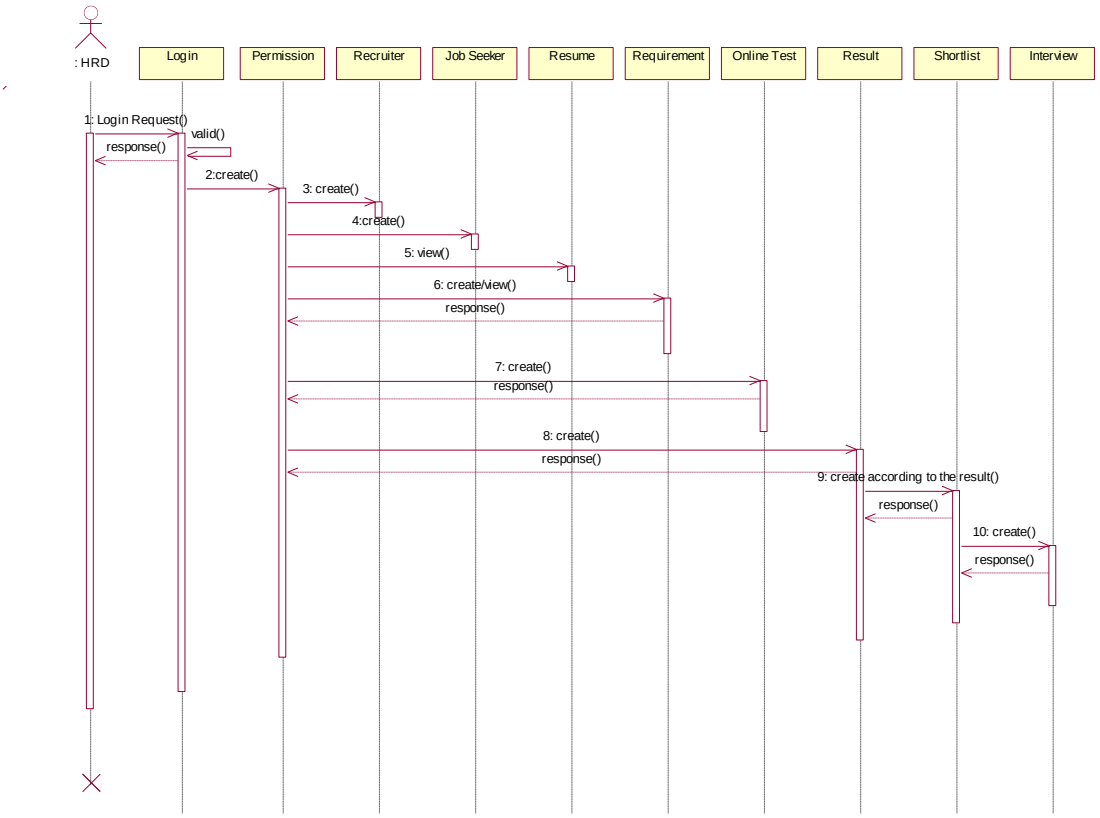


Job Seeker Process

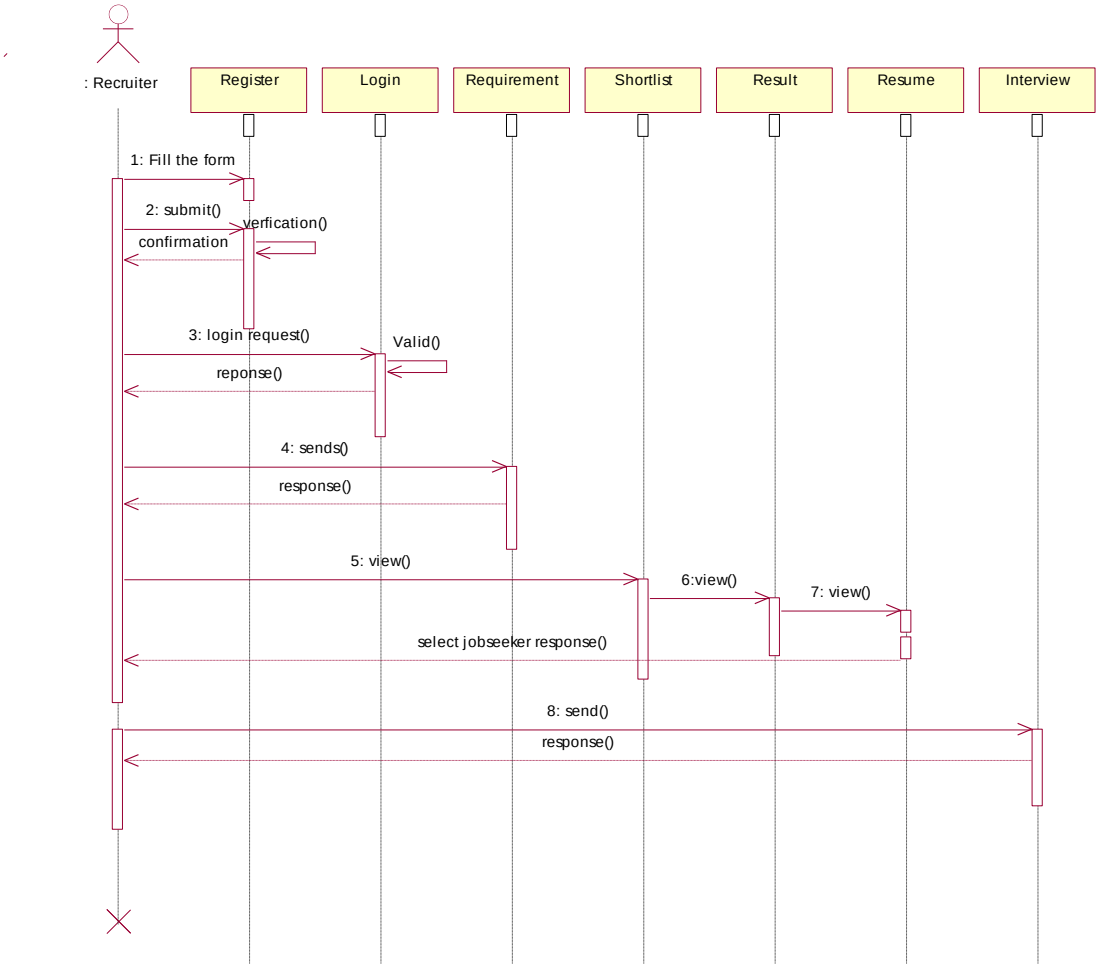


SEQUENCE DIAGRAM

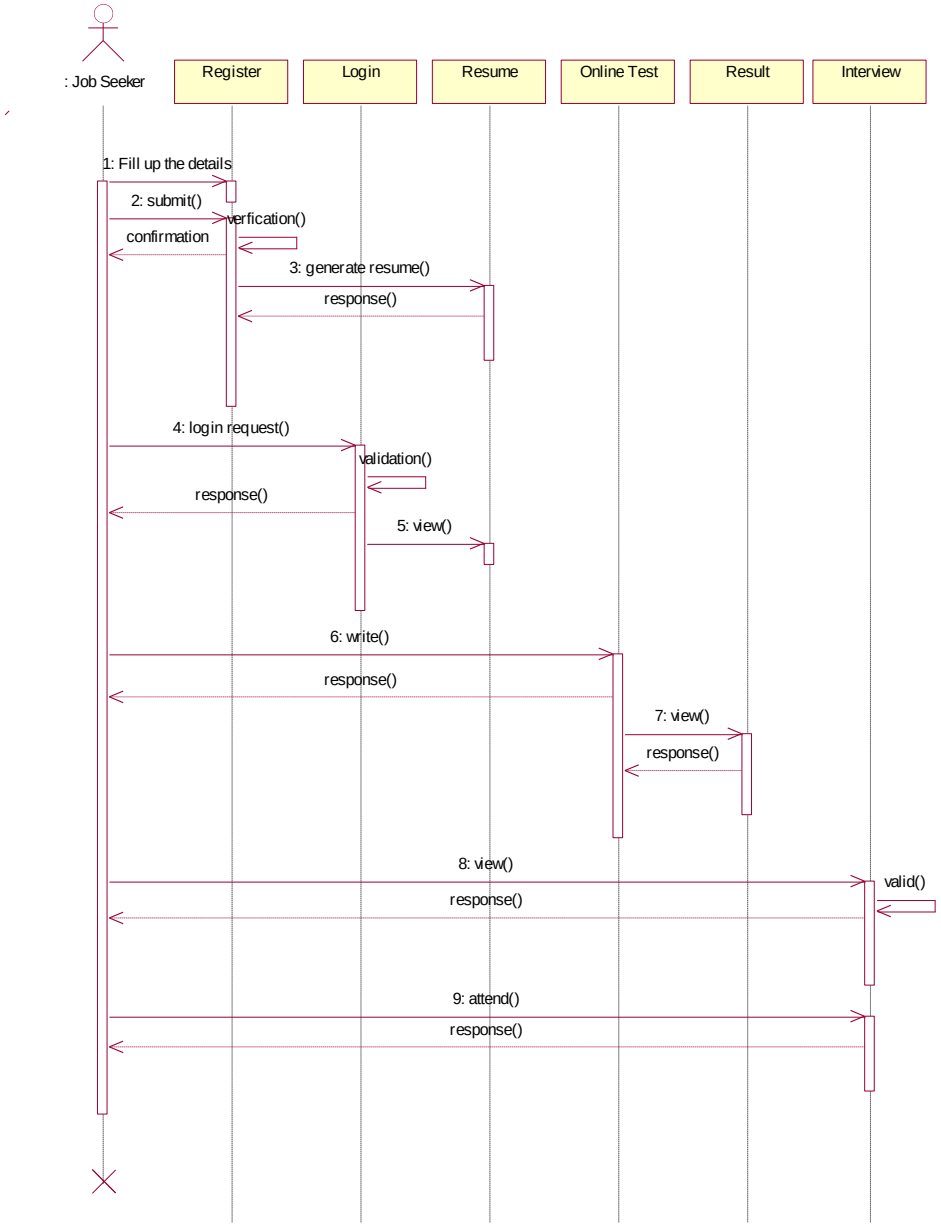
Administrator Sequence



Recruiter Sequence

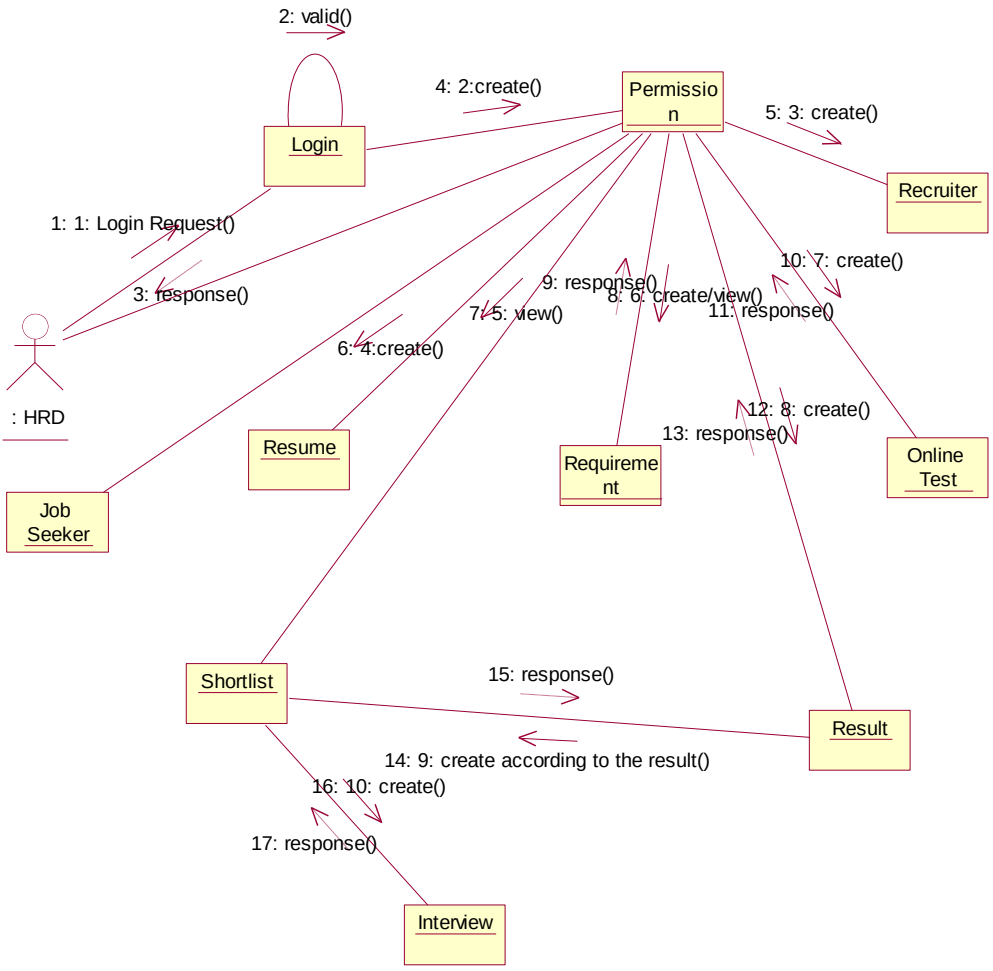


Job Seeker Sequence

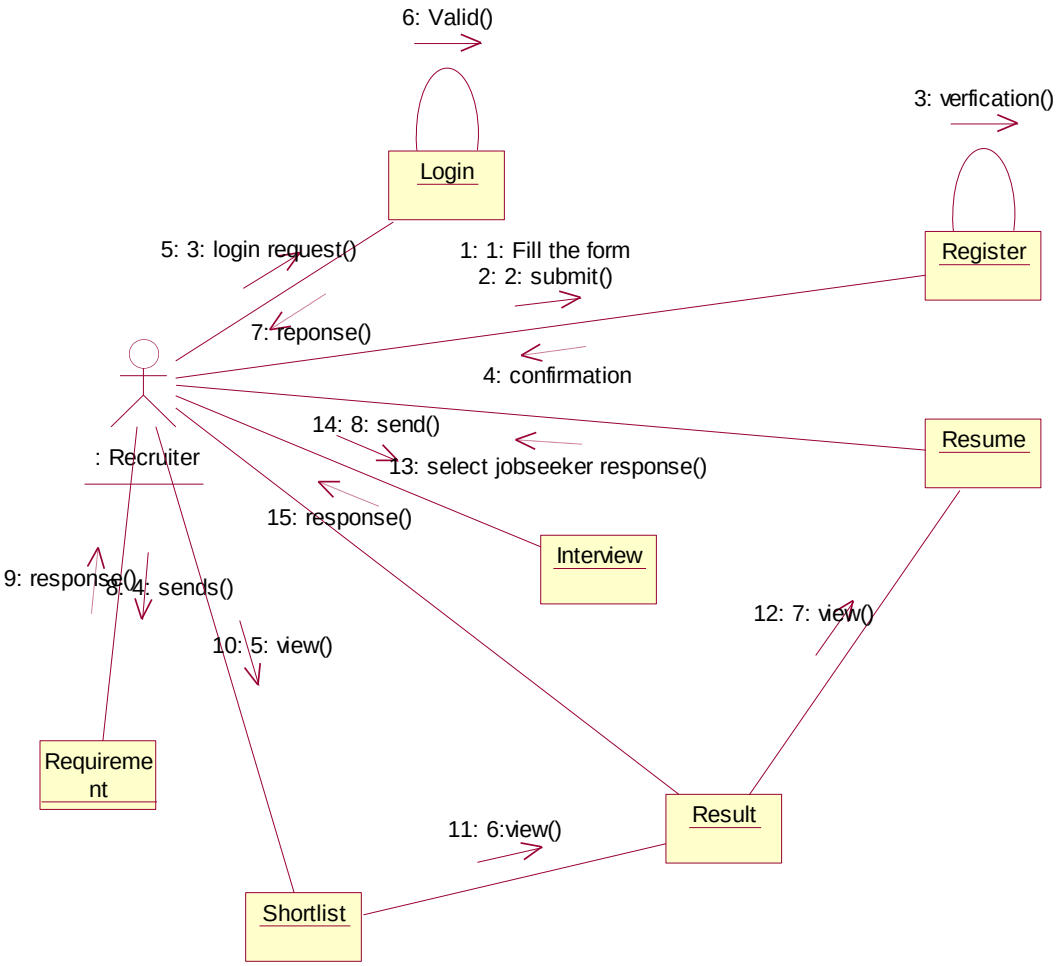


COLLABORATION DIAGRAM

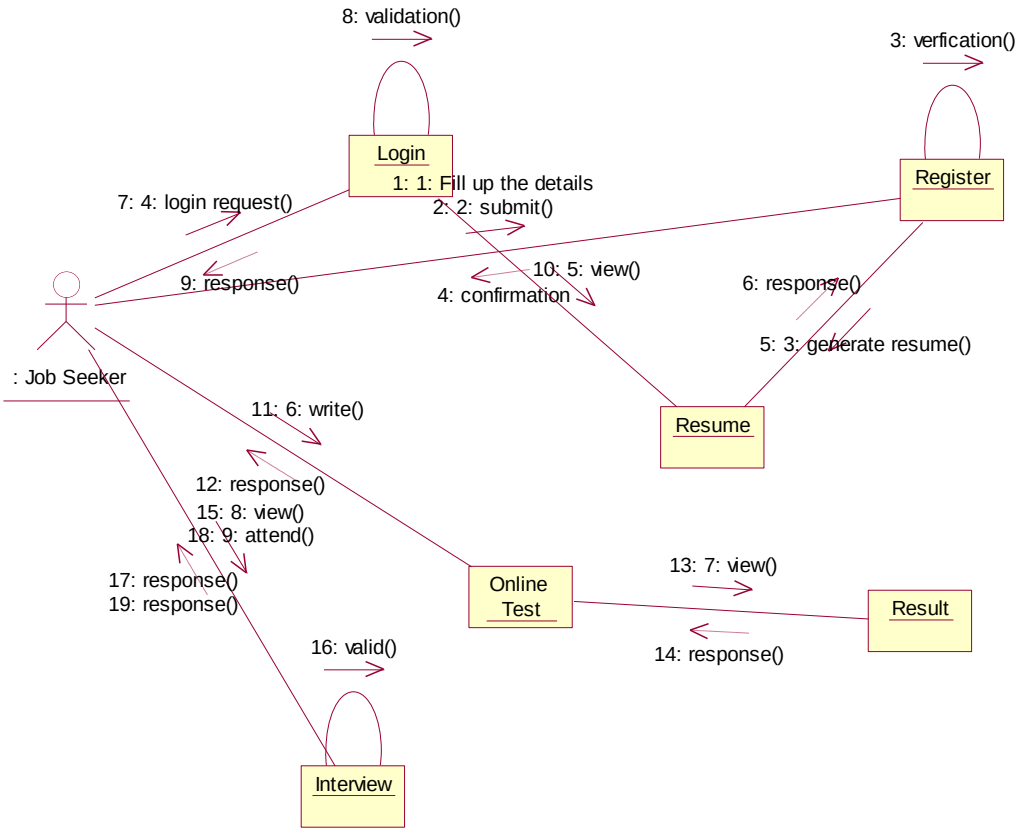
Administrator Collaboration



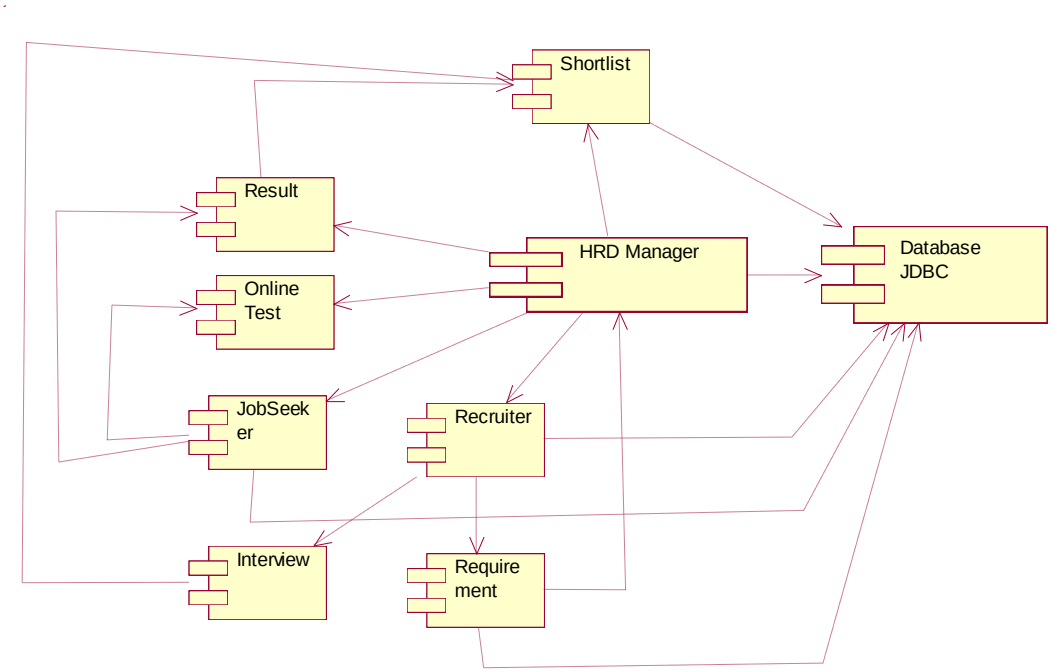
Recruiter Collaboration



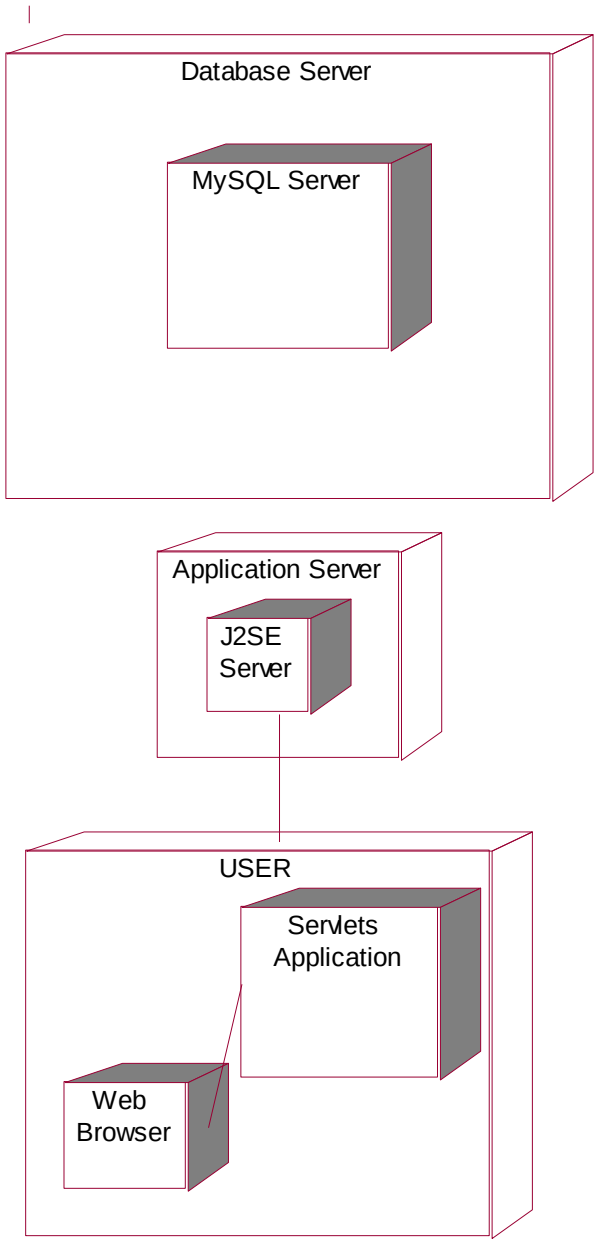
Job Seeker Collaboration



COMPONENT DIAGRAM:



DEPLOYMENT DIAGRAM



TESTING & IMPLEMENTATION SYSTEM TESTING

Testing is the stage of implementation of which aimed at ensuring that the system works accurately and efficiently before live operation commences. Testing is vital to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct the goal will be achieved. The candidates system subject to a variety of tests. Online response, volume, stress, recovery, security and usability tests. A series of testing are performed for the proposed system before the system is ready for user acceptance testing. UNIT TESTING Unit testing involves the design of test cases that validate that the internal program logic is functioning properly and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at the component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event-driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components. College Transport Management System

FUNCTIONAL TEST

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals. Functional testing is centered on the following items: Valid Input : identified classes of valid input must be accepted Invalid Input : identified classes of invalid input must be rejected Functions : identified functions must be exercised Output : identified classes of application outputs must be exercised Systems/Procedures : interfacing systems or procedures must be invoked Organization and preparation of functional tests are focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

SYSTEM TEST

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

WHITE BOX TESTING

White Box Testing is a testing in which the software tester has knowledge of the inner workings, structure, and language of the software, or at least its purpose. It is a purpose. It is used to test areas that cannot be reached from a black box level.

BLACK BOX TESTING

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, like most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. College Transport Management System .

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases TEST STRATEGY AND APPROACH Field testing will be performed manually and functional tests will be written in detail.

TEST OBJECTIVES

• All field entries must work properly. • Pages must be activated from the identified link. • The entry screen, messages, and responses must not be delayed. FEATURES TO BE TESTED • Verify that the entries are of the correct format • No duplicate entries should be allowed • All links should take the user to the correct page.

INTEGRATION TESTING

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error. Test Results: All the test cases mentioned above passed successfully. No defects encountered.

IMPLEMENTATION

The term implementation has different meaning, ranging from the conversion of a basic application to a complete replacement of a computer system. The procedure however is virtually the same. Implementation is used here to mean the process of converting new system design into an operational one. Conversion is one aspect of implementation. The other aspects are the post implementation overview and maintenance.

Types of Implementation Implementation of a computer system to replace a manual system. The problems encountered are converting files, training users, creating accurate files and verification printouts for integrity. Implementation of a new computer system to replace an existing one. This is usually a difficult conversion. If not properly planned, there can be many problems. Some large computer systems taken years to convert. Implementation of a modified application to replace an existing one uses the same computer. This type of conversion is relatively easy to handle provided there are no major changes in the files.

SOFTWARE MAINTENANCE

Maintenance is the enigma of system development. It holds the software industry captive. Typing up programming resources. Analysis and programmers spend far more time maintaining programs than they do within them. Maintaining accounts for 50-80 percent in total system development.

INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on

controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining privacy.

Input Design considered the following things: 1. What data should be given as input? 2. How the data should be arranged or coded? 3. The dialog to guide the operating personnel in providing input. Methods for preparing input validations and steps to follow when an error occurs.

OBJECTIVES

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2. It is achieved by creating user-friendly screens for the data entry to handle a large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow. College Transport Management System

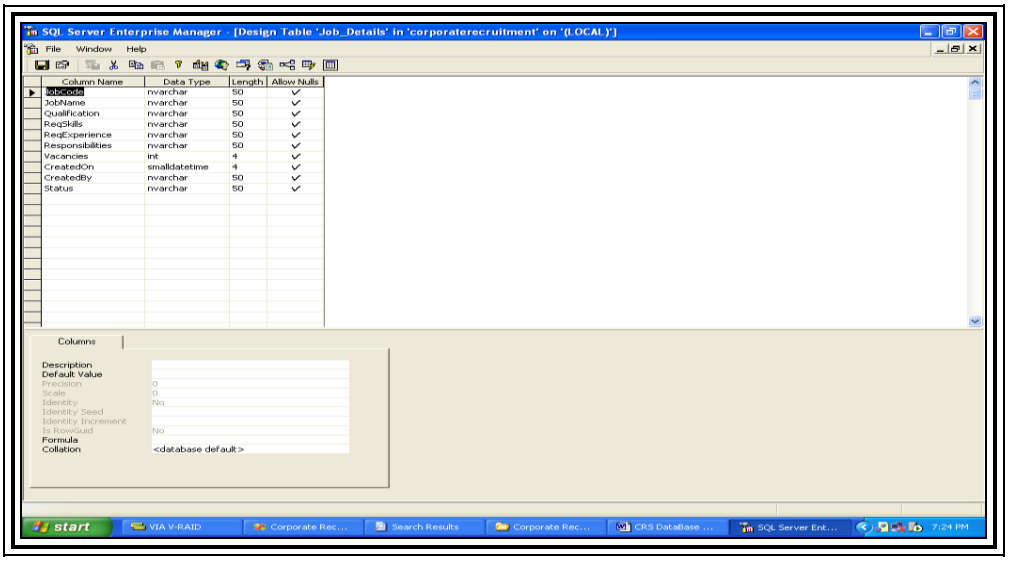
OUTPUT DESIGN

Quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to another system through outputs. In output design, it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the

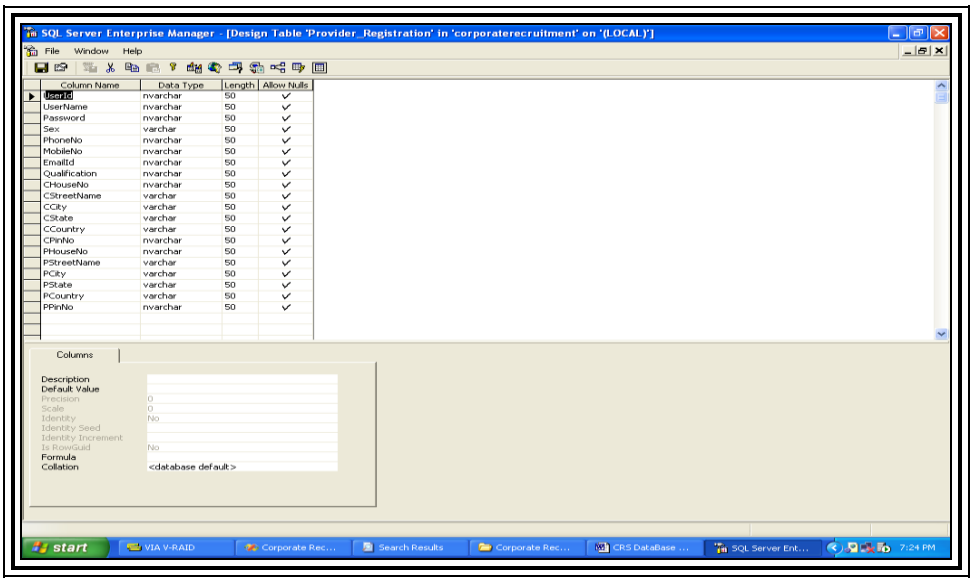
most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making. 1.Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements. 2. Select methods for presenting the information.. Create a document, report, or other formats that contain information produced by the system. The output form of an information system should accomplish one or more of the following objectives. o Convey information about past activities, current status or projections of the Future. o Signal important events, opportunities, problems, or warnings. o Trigger an action. o Confirm an action. The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

DATABASE DESIGN

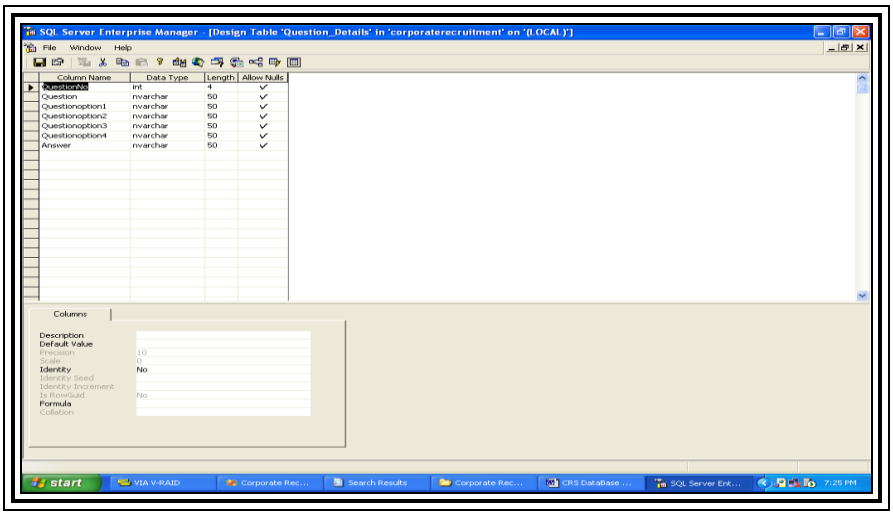
Job Details



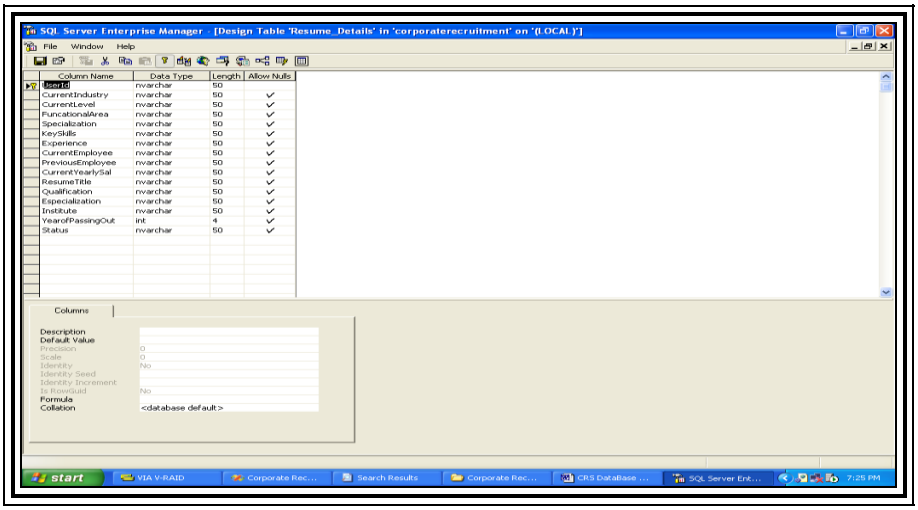
Provider Details



Question Details



Resume Details



SAMPLE SOURCE CODINGS

REGISTER FORM:

```
using System;

usingSystem.Collections.Generic;
usingSystem.ComponentModel;
usingSystem.Data;
usingSystem.Drawing;
usingSystem.Linq;
usingSystem.Text;
usingSystem.Windows.Forms;
usingSystem.Data.SqlClient;
namespace WindowsFormsApplication3
{
publicpartialclassForm1 : Form
{
public Form1()
InitializeComponent();
}
privatevoid panel1_Paint(object sender, PaintEventArgs e)
{
}
SqlCommandcmd;
SqlConnection con;
SqlDataReaderrd;
string a, b,c;
publicstaticstringcs = @"Data
Source=.\SQLEXPRESS;AttachDbFilename=C:\Users\hp\Desktop\pattern
search\WindowsFormsApplication3\Database1.mdf;Integrated Security=True;User
Instance=True";
privatevoid Form1_Load(object sender, EventArgs e)
{
panel1.BackColor = Color.FromArgb(100, 0, 0, 0);
con = newSqlConnection(cs);
con.Open();
}
privatevoid button2_Click(object sender, EventArgs e)
{
this.Hide();
Form2 f2 = newForm2();
f2.Show();
}
privatevoid label4_Click(object sender, EventArgs e)
{
this.Hide();
Form5 f5 = newForm5();
```

```
f5.Show();
}
privatevoid label5_Click(object sender, EventArgs e)
{
this.Hide();
Form3 f3 = newForm3();
f3.Show();
}
privatevoid button3_Click(object sender, EventArgs e)
{
cmd = newSqlCommand("select uname,pwd from admin where uname='" +
textBox1.Text + "'and pwd='" + textBox2.Text + "'", con);
rd = cmd.ExecuteReader();
while (rd.Read())
{
b = rd.GetString(0);
c = rd.GetString(1);
}
rd.Close();
if (textBox1.Text == b && textBox2.Text == c)
{
MessageBox.Show("LOGIN SUCCESS");
this.Hide();
Form6 F6 = newForm6();
F6.Show();
}
else
{
MessageBox.Show("LOGIN FAILURE");
textBox1.Text = "";
textBox2.Text = "";
}
}
privatevoid button1_Click(object sender, EventArgs e)
{
cmd = newSqlCommand("select uname,pwd from login where uname='" +
textBox1.Text + "'and pwd='" + textBox2.Text + "'", con);
rd = cmd.ExecuteReader();
while (rd.Read())
{
a = rd.GetString(0);
b = rd.GetString(1);
}
rd.Close();

if ((textBox1.Text == a) && (textBox2.Text == b))
```

```
{
MessageBox.Show("LOGIN SUCCESS");
this.Hide();
Form7 F7 = newForm7();
F7.Show();
}
else
{
MessageBox.Show("LOGIN FAILURE");
textBox1.Text = "";
textBox2.Text = "";
}
using System;
usingSystem.Collections.Generic;
usingSystem.ComponentModel;
usingSystem.Data;
usingSystem.Drawing;
usingSystem.Linq;
usingSystem.Text;
usingSystem.Windows.Forms;
usingSystem.Data.SqlClient;
namespace WindowsFormsApplication3
{
publicpartialclassForm6 : Form
{
public Form6()
{
InitializeComponent();
}
SqlCommandcmd;
SqlConnection con;
SqlDataReaderrd;
string a, filename;
publicstaticstring text;
privatevoid Form6_Load(object sender, EventArgs e)
{
// TODO: This line of code loads data into the 'database1DataSet18.file1' table. You can
move, or remove it, as needed.
this.file1TableAdapter.Fill(this.database1DataSet18.file1);
// TODO: This line of code loads data into the 'database1DataSet1.product' table. You
can move, or remove it, as needed.
///this.productTableAdapter.Fill(this.database1DataSet1.product);
panel1.BackColor = Color.FromArgb(100, 0, 0, 0);
con = newSqlConnection(Form1.cs);
con.Open();
```

```
}
privatevoid textBox2_TextChanged(object sender, EventArgs e)
{
}
privatevoid tabPage1_Click(object sender, EventArgs e)
{
}
privatevoid button2_Click(object sender, EventArgs e)
{
if ((textBox1.Text != "") && (textBox2.Text != "") && (textBox3.Text != "") &&
(textBox4.Text != "") )
{
cmd = newSqlCommand("insert into file1 values('" + textBox1.Text + "','" +
textBox2.Text + "','" + textBox3.Text + "','" + textBox4.Text + "','" + text + "')", con);
cmd.ExecuteNonQuery();
MessageBox.Show("File Detail is uploaded");
}
else
{
MessageBox.Show("All Feilds are required");
textBox3.Text = "";
textBox2.Text = "";
textBox1.Text = "";
textBox4.Text = "";
text = "";
}
}
privatevoid label5_Click(object sender, EventArgs e)
{
}
privatevoid textBox1_TextChanged(object sender, EventArgs e)
{
if (filename == null)
{
MessageBox.Show("Select the File");
textBox1.Text = "";
}
}

privatevoid textBox3_TextChanged(object sender, EventArgs e)
{
}

privatevoid textBox4_TextChanged(object sender, EventArgs e)
{
```

```
}

privatevoid textBox5_TextChanged(object sender, EventArgs e)
{
}
privatevoid button1_Click(object sender, EventArgs e)
{
this.Hide();
Form1 f1 = newForm1();
f1.Show();
}
privatevoid button3_Click(object sender, EventArgs e)
{
this.Hide();
Form6 f6 = newForm6();
f6.Show();
}
privatevoid label7_Click(object sender, EventArgs e)
{
}
privatevoid label1_Click(object sender, EventArgs e)
{
}
privatevoid button4_Click(object sender, EventArgs e)
{
openFileDialog1.Filter = "text files(*.txt; )|*.*; ";
if (openFileDialog1.ShowDialog() == DialogResult.OK)
{
text = System.IO.File.ReadAllText(openFileDialog1.FileName);
filename = openFileDialog1.FileName;

}
//  MessageBox.Show(text);
MessageBox.Show("FILE IS SELECTED ");
}
}
}
using System;
usingSystem.Collections.Generic;
usingSystem.ComponentModel;
usingSystem.Data;
usingSystem.Drawing;
usingSystem.Linq;
usingSystem.Text;
usingSystem.Windows.Forms;
usingSystem.Data.SqlClient;
```

```
using System.IO;
namespace WindowsFormsApplication3
{
publicpartialclassForm7 : Form
{
public Form7()
{
InitializeComponent();
}
SqlCommandcmd ;
SqlConnection con;
SqlDataReaderrd;
DataTabledt = newDataTable();
string[] x = newstring[4]{"File_name","Author_name","Year","Volume"};
int z=0;
publicstaticstring a1,c;

privatevoid Form7_Load(object sender, EventArgs e)
{
con = newSqlConnection(Form1.cs);
con.Open();
// TODO: This line of code loads data into the 'database1DataSet17.tmp' table. You can
move, or remove it, as needed.
this.tmpTableAdapter2.Fill(this.database1DataSet17.tmp)

}

privatevoidfilldata()
{
SqlDataAdapter ad = newSqlDataAdapter();
cmd = newSqlCommand("select * from file1", con);
ad.SelectCommand = cmd;
ad.Fill(dt);
//dataGridView1.DataSource = dt;

}
privatevoid button1_Click_1(object sender, EventArgs e)
{
cmd =  newSqlCommand("insert into tmp select * from file1",con);
cmd.ExecuteNonQuery();

// dataGridView1.Rows.Clear();
cmd = newSqlCommand("select * from tmp ", con);
rd = cmd.ExecuteReader();
while (rd.Read())
{
```

```
dataGridView1.Rows.Add(rd.GetString(0), rd.GetString(1), rd.GetString(2),
rd.GetString(3), rd.GetString(4));
}
rd.Close();
string a = textBox1.Text;
string[] col = a.Split(newchar[] {'+','-'});
char[] c = newchar[100];
string[] cha = newstring[100];
int i = 0,n=0;
foreach(charchin a)
{

if (ch == '+')
{
c[i] = '+';
i++;
}
if (ch == '-')
{
c[i] = '-';
i++;
}
}

foreach(stringchin col)
{
cha[n] = ch;
n++;
}
cmd = newSqlCommand("delete  from tmp where  File_Name NOT LIKE '" + cha[0] +
"'", con);
cmd.ExecuteNonQuery();

if (c[0] == '+')
{
cmd = newSqlCommand("insert into t2 select * from tmp", con);
cmd.ExecuteNonQuery();
int x = 0;
string[] b = cha[1].Split(',');
foreach (string l in b)
{
x++;
}
foreach (string l in b)
{
```

```
cmd = newSqlCommand("delete  from tmp where  Author_Name NOT LIKE '" + l + "'",
con);
cmd.ExecuteNonQuery();
if (x != 1)
{
cmd = newSqlCommand("insert into tmp select * from t2 where Author_Name NOT
LIKE '" + l + "'", con);
cmd.ExecuteNonQuery();
}
}
}
elseif (c[0] == '-')
{
string[] b = cha[1].Split(',');
foreach (string l in b)
{
cmd = newSqlCommand("delete  from tmp where  Author_Name  LIKE '" + l + "'",
con);
cmd.ExecuteNonQuery();
}
}
if (c[1] == '+')
{
cmd = newSqlCommand("delete from t2", con);
cmd.ExecuteNonQuery();
cmd = newSqlCommand("insert into t2 select * from tmp", con);
cmd.ExecuteNonQuery();
int x = 0;
string[] b = cha[2].Split(',');
foreach (string l in b)
{
x++;
}
foreach (string l in b)
{
cmd = newSqlCommand("delete  from tmp where  Year NOT LIKE '" + l + "'", con);
cmd.ExecuteNonQuery();
if (x != 1)
{
cmd = newSqlCommand("insert into tmp select * from t2 where Year NOT LIKE '" + l +
"'", con);
cmd.ExecuteNonQuery();
}
}
}
elseif (c[1] == '-')
```

```
{
string[] b = cha[2].Split(',');
foreach (string l in b)
{
cmd = newSqlCommand("delete  from tmp where  Year  LIKE '" + l + "'", con);
cmd.ExecuteNonQuery();
}
}
if (c[2] == '+')
{
cmd = newSqlCommand("delete from t2", con);
cmd.ExecuteNonQuery();
cmd = newSqlCommand("insert into t2 select * from tmp", con);
cmd.ExecuteNonQuery();
int x = 0;
string[] b = cha[3].Split(',');
foreach (string l in b)
{
x++;
}
foreach (string l in b)
{
cmd = newSqlCommand("delete  from tmp where  Volume NOT LIKE '" + l + "'", con);
cmd.ExecuteNonQuery();
if (x != 1)
{cmd = newSqlCommand("insert into tmp select * from t2 where Year NOT LIKE '" + l
+ "'", con);
cmd.ExecuteNonQuery();
}
elseif (c[2] == '-')
{
string[] b = cha[3].Split(',');
foreach (string l in b)
{
cmd = newSqlCommand("delete  from tmp where  Volume  LIKE '" + l + "'", con);
cmd.ExecuteNonQuery();
}
}
dataGridView1.Rows.Clear();
cmd = newSqlCommand("select * from tmp ", con);
rd = cmd.ExecuteReader();
{
cmd = newSqlCommand("select * from file1 where  text='" + a1 + "'", con);
rd = cmd.ExecuteReader();
while (rd.Read())
{
```

```
a1 = rd.GetString(0);
}
rd.Close();
MessageBox.Show("FILE IS  SELECTED");
TextWriter txt = newStreamWriter(@"E:\" + c + ".txt");
txt.Write(Form6.text);
txt.Close();
MessageBox.Show("downloading");
MessageBox.Show(" FILE DOWNLOADED @ E: IN THE NAME OF",c);
}
else
{
MessageBox.Show("FILE IS NOT SELECTED");
}

}
privatevoid dataGridView1_CellClick(object sender, DataGridViewCellEventArgs e)
{
privatevoid dataGridView1_CellClick_1(object sender, DataGridViewCellEventArgs e)
}
<%@ Page Language="C#" AutoEventWireup="true"  CodeFile="Default.aspx.cs"
Inherits="_Default" %>
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
<title>Untitled Page</title>
</head>
<body>
<form id="form1" runat="server">
<div>
<strong>                 
 
              college transport management
system<br />
<br />
<asp:Label ID="Label1" runat="server" Text="Student name"
Width="108px"></asp:Label>
                 
<asp:TextBox ID="TextBox1" runat="server"></asp:TextBox><br />
```

```
                     
<br />
<asp:Label ID="Label2" runat="server" Text="Department"></asp:Label>
                     
  


<
<%@ Page Language="C#" MasterPageFile="~/MasterPage.master"
AutoEventWireup="true" CodeFile="surgeries.aspx.cs" Inherits="surgeries"
Title="Untitled Page" %>
<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
<asp:Label ID="Label1" runat="server" Font-Bold="False" Font-Names="Modern"
Font-Size="24pt"
ForeColor="White" Style="z-index: 100; left: 146px; position: absolute; top: 184px;
color: red;"
Text="SurgeryInformation                    "></asp:Label>
<table style="width: 306px; z-index: 101; left: 239px; position: absolute; top: 289px;
color: white; font-family: 'Lucida Console'; height: 113px;">
<tr>
<td style="width: 682px; height: 20px"> Patient Id
</td>
<td style="width: 115px; height: 20px">
<asp:DropDownList ID="pidddl" runat="server" Width="155px" AutoPostBack="True"
OnSelectedIndexChanged="pidddl_SelectedIndexChanged" >
</asp:DropDownList>
</td>
</tr>
<tr>
<td style="width: 682px; height: 20px"> Patient Name</td>
<td style="width: 115px; height: 20px">
<asp:TextBox ID="pntxt" runat="server"></asp:TextBox>
</td>
```

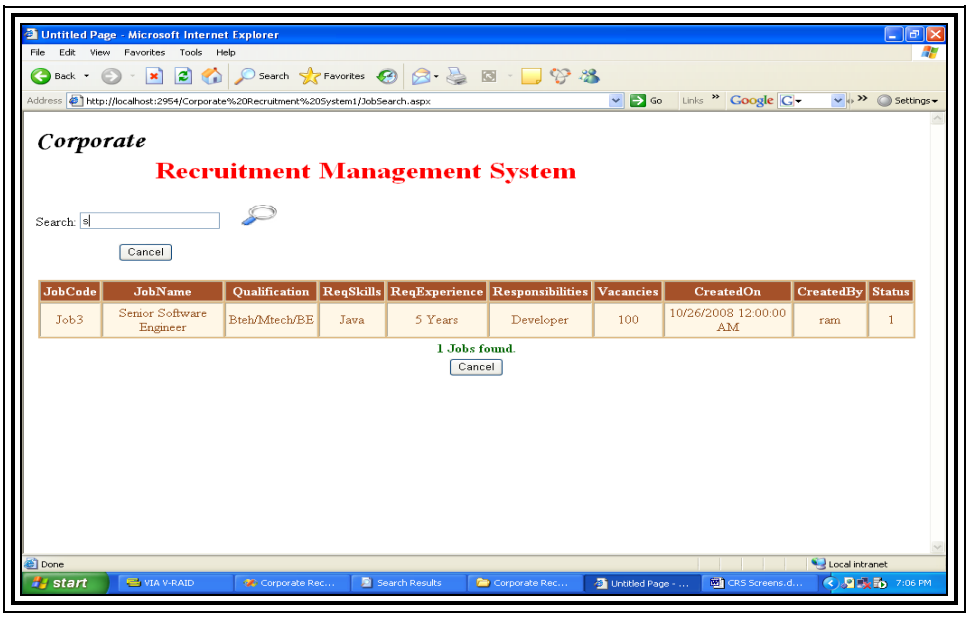
```
</tr>
<tr>
<td style="width: 682px; height: 20px"> Age
</td>
<td style="width: 115px; height: 20px">
<asp:TextBox ID="agtxt" runat="server"></asp:TextBox>
</td>
</tr>
<tr>
<td style="width: 682px; height: 6px"> Surgery Date
</td>
<td style="width: 115px; height: 6px">
<asp:TextBox ID="sdtxt" runat="server"></asp:TextBox>
</td>
</tr>
<tr>
<td style="width: 682px; height: 20px"> Department
</td>
<td style="width: 115px; height: 20px">
<asp:TextBox ID="deptxt" runat="server"></asp:TextBox>
</td>
</tr>
<tr>
<td style="width: 682px; height: 20px"> Doctor
</td>
<td style="width: 115px; height: 20px">
<asp:TextBox ID="doctxt" runat="server"></asp:TextBox>
</td>
</tr>
</table>
<table style="z-index: 102; left: 289px; position: absolute; top: 485px">
```

SAMPLE SCREENS & OUTPUTS

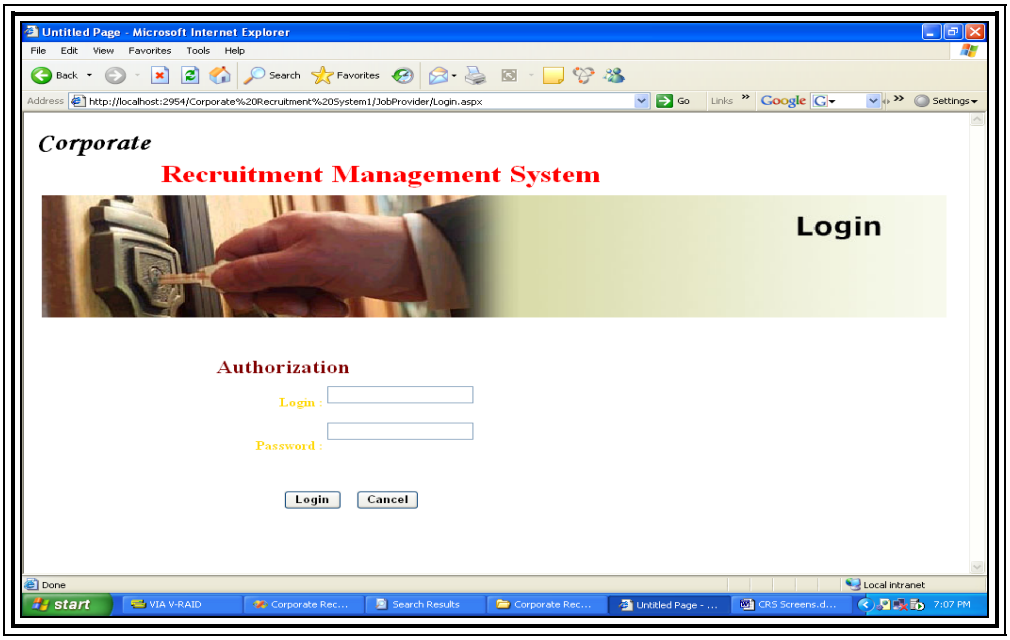
SEEKER REGISTRATION

Home Page

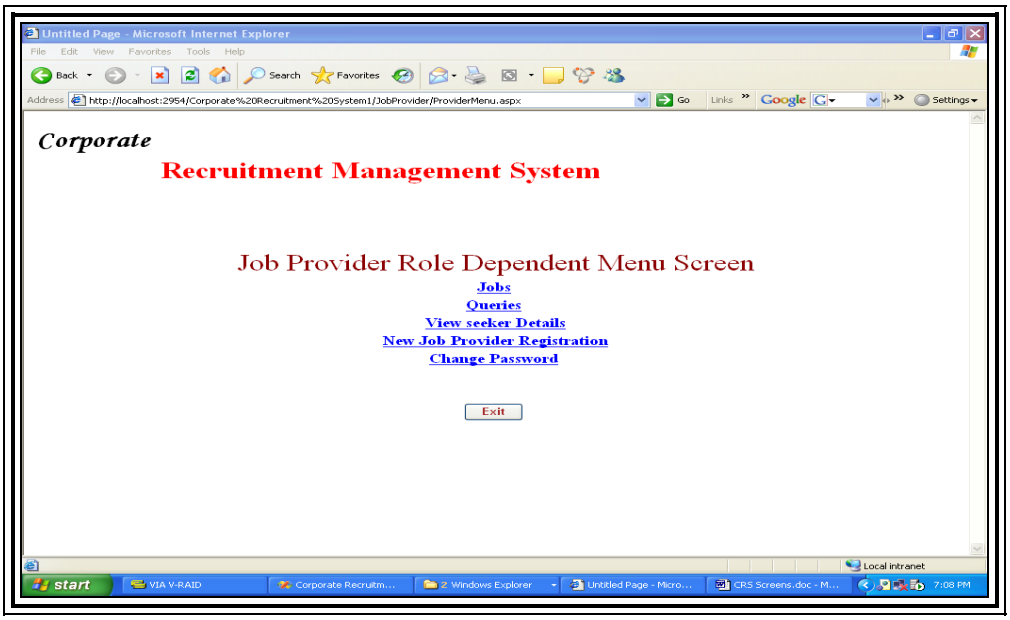
Search Jobs



Job Provider Login

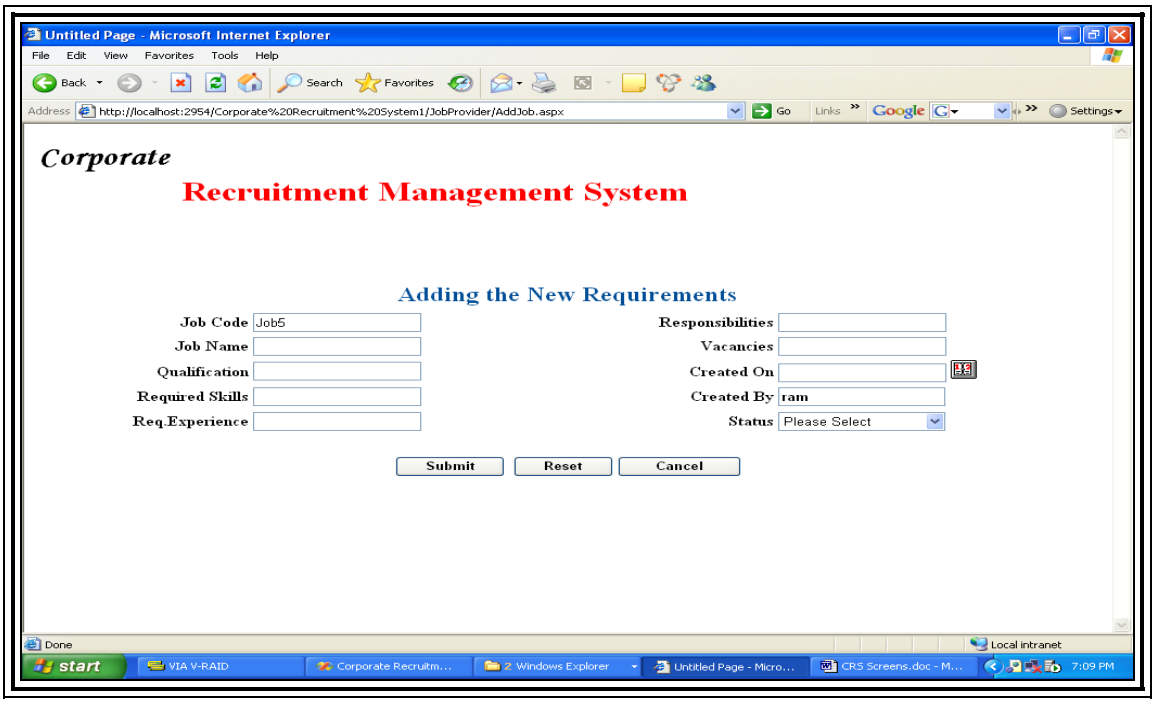


Job Provider Role Dependent Screen

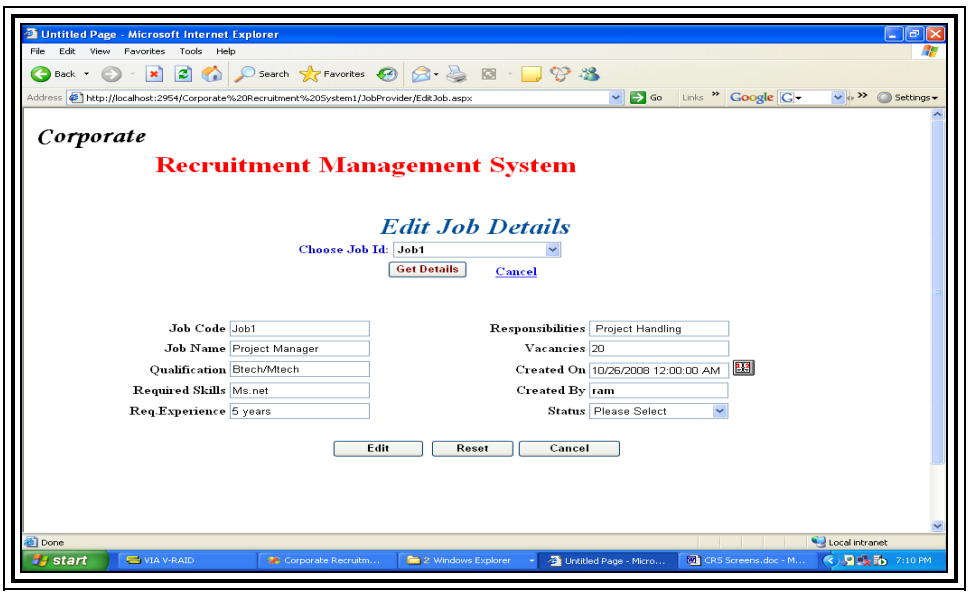


Jobs Home Page

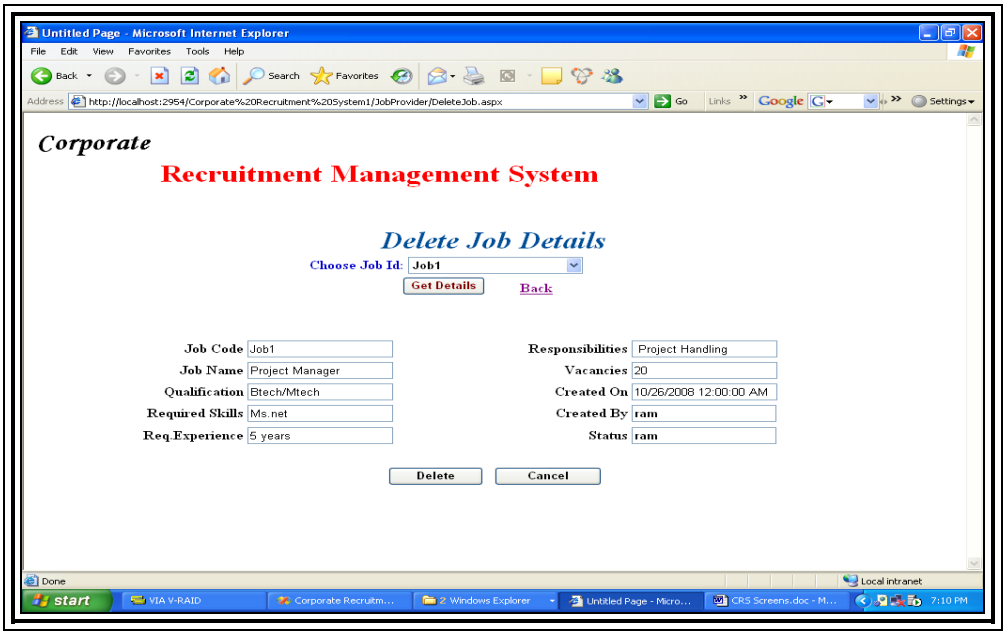
Add New Job



Edit Job Details



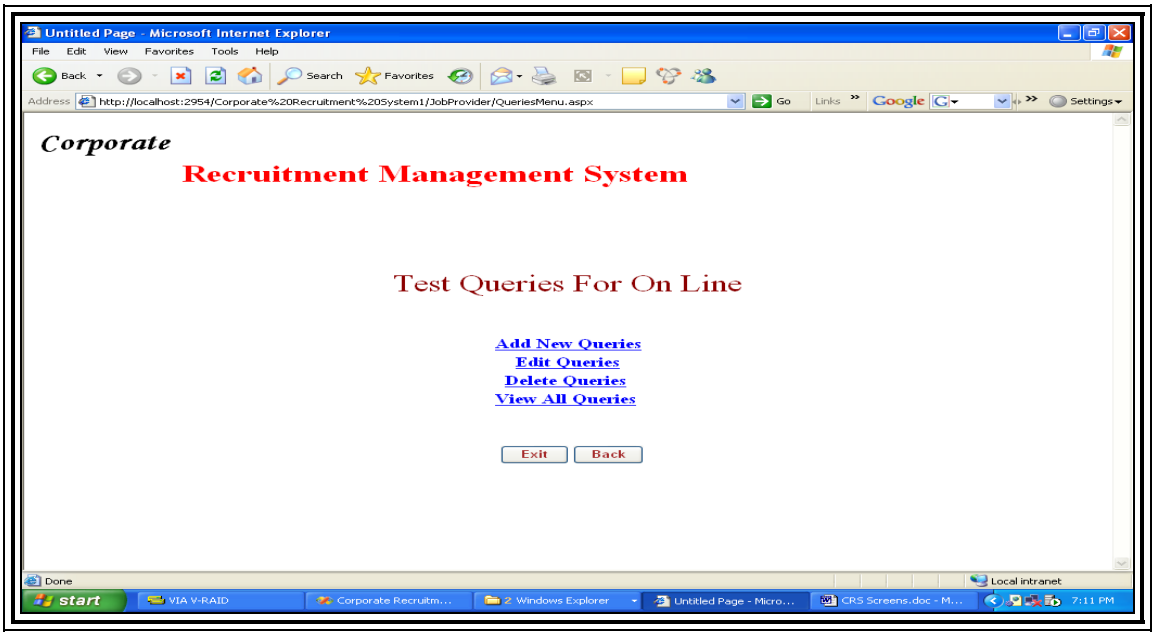
Delete Job Details



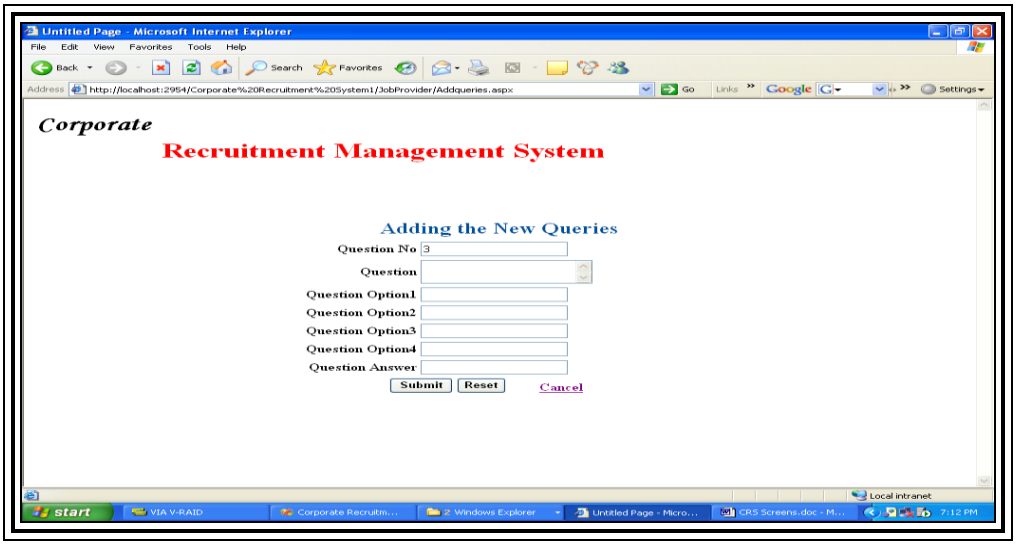
View Jobs

JobCode	JobName	Qualification	ReqSkills	ReqExperience	Responsibilities	Vacancies	CreatedOn	CreatedBy	Status
Job1	Project Manager	Btech/Mtech	Ms.net	5 years	Project Handling	20	10/26/2008 12:00:00 AM	ram	1
Job2	Team Leader	Btech/Mtech/BE	Ms.net,SqlServer	4 Years	Team Lead	50	10/26/2008 12:00:00 AM	ram	1
Job3	Senior Software Engineer	Bteh/Mtech/BE	Java	5 Years	Developer	100	10/26/2008 12:00:00 AM	ram	1
Job4	Programmer	Btech	Advanced Java	5 Years	Design	30	10/27/2008 12:00:00 AM	ram	1

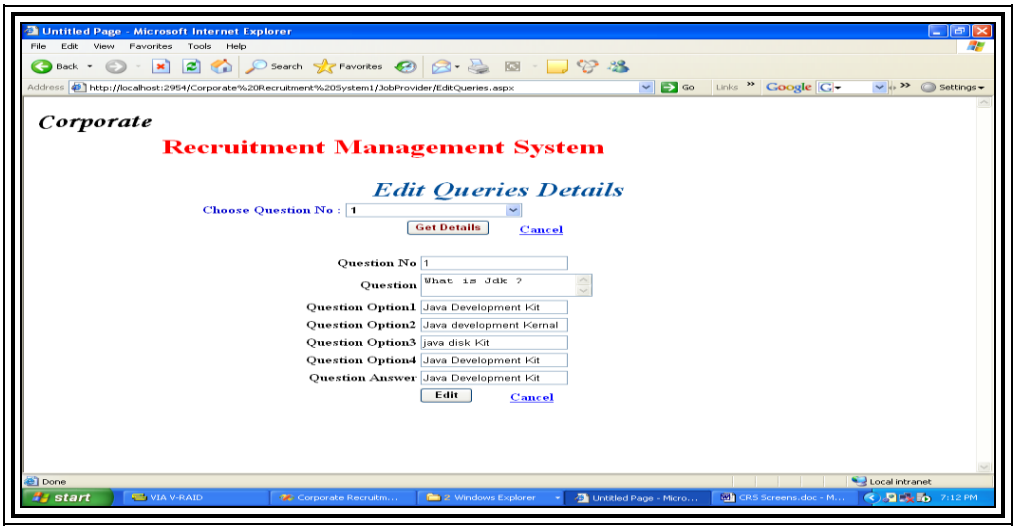
Queries Menu Page



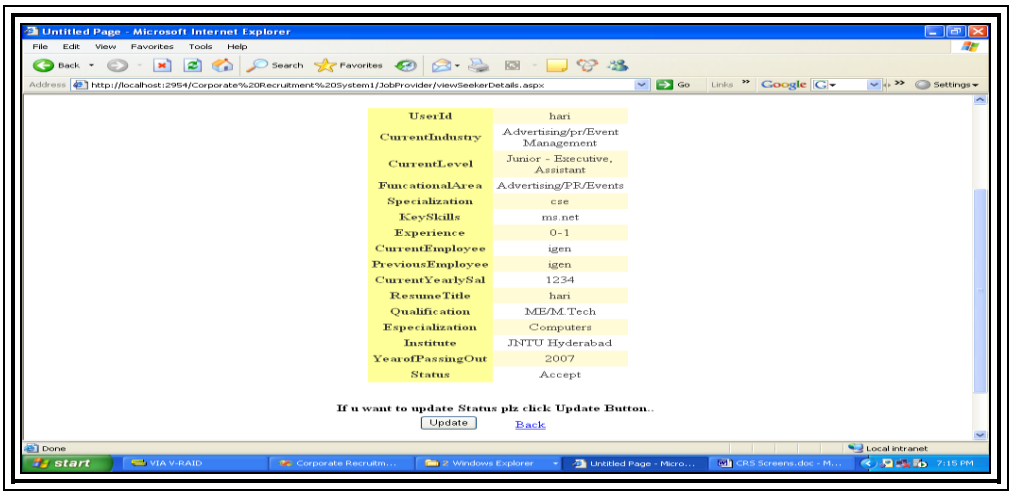
Add New Query



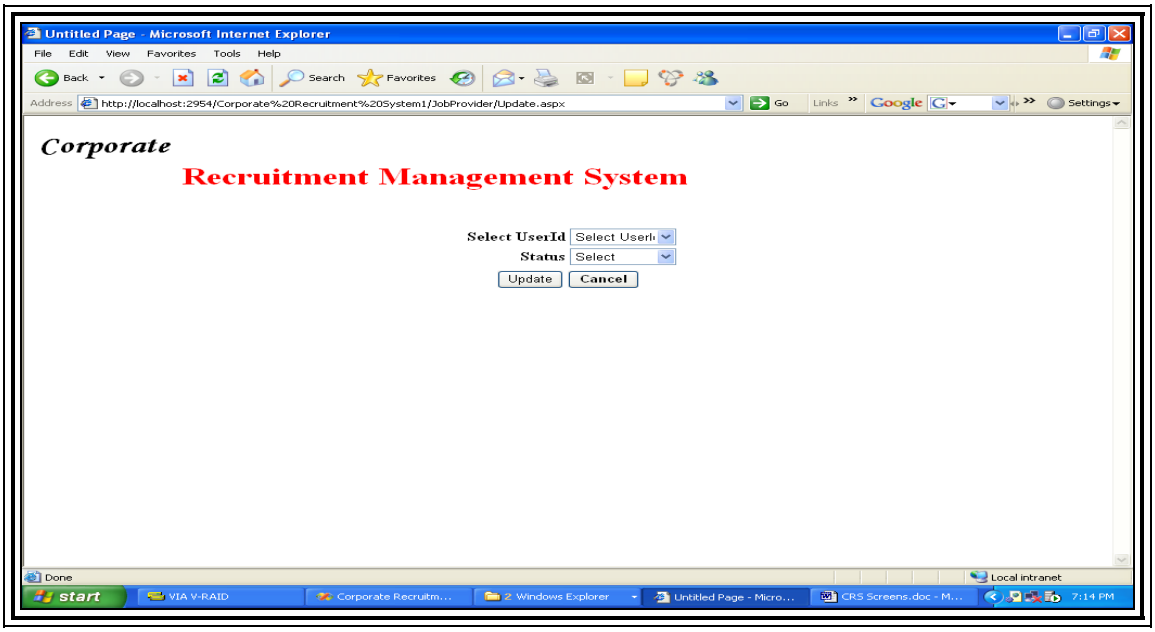
Edit Query Details



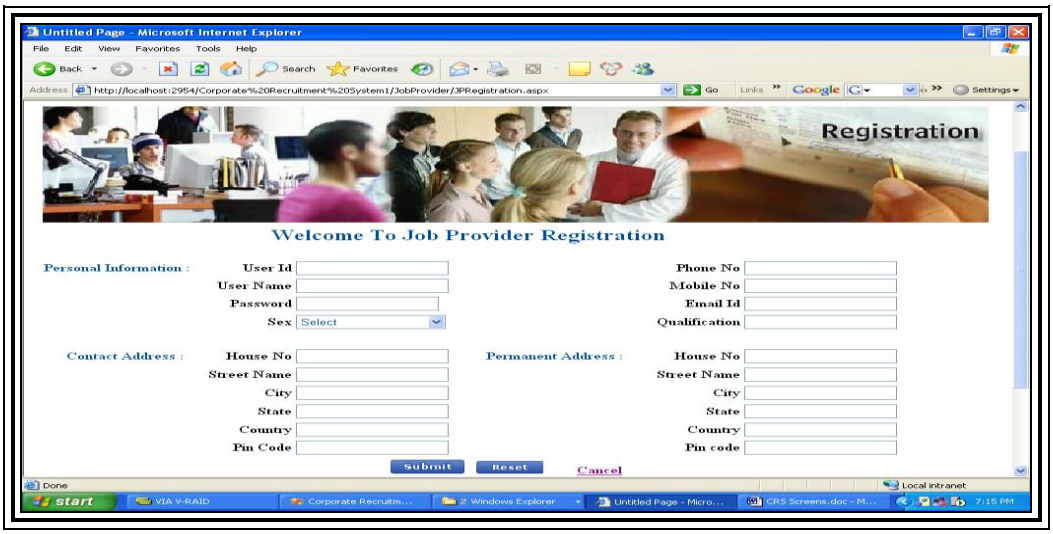
View Seeker Details



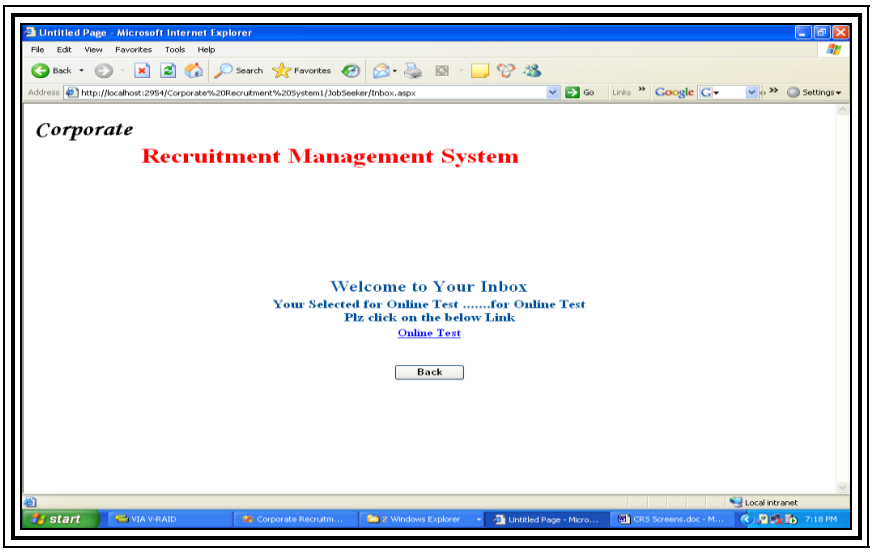
Update Details



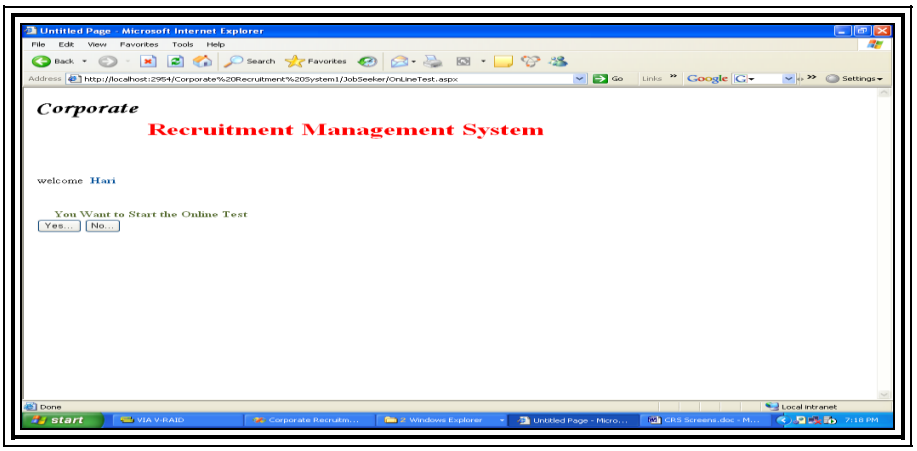
New Job Provider Registration



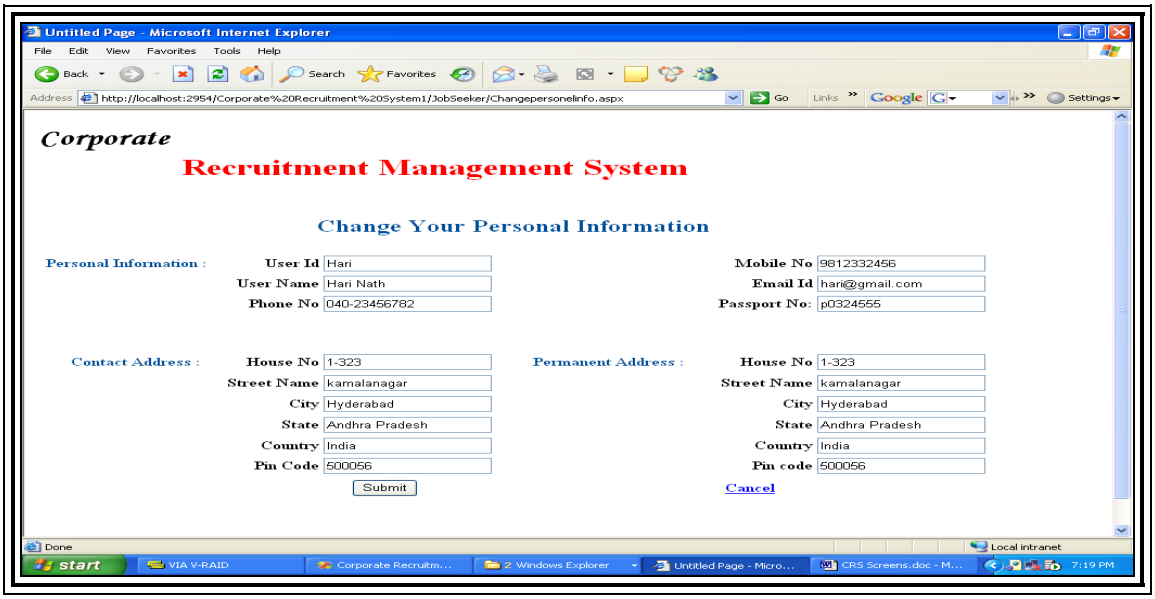
Inbox



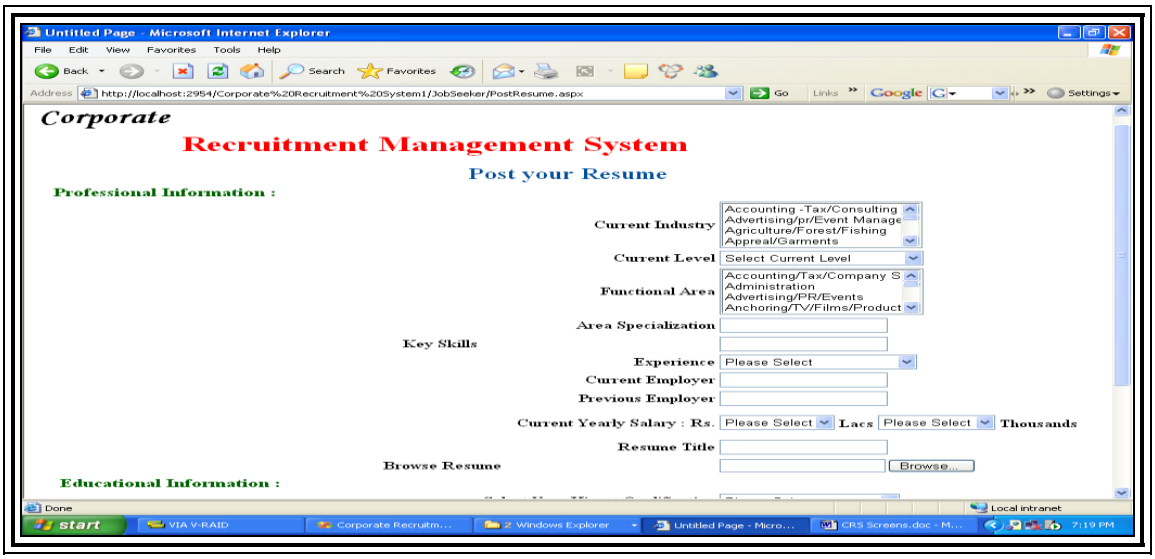
Online Test



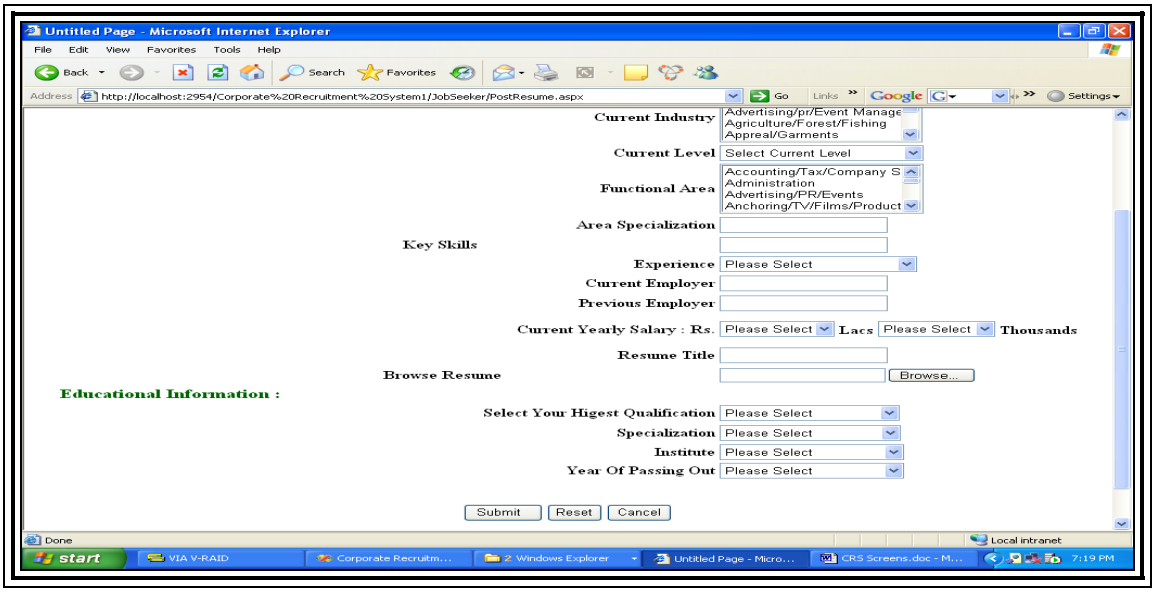
Change Personal Profile



Post Resume



Educational Information



View Jobs

JobCode	JobName	Qualification	ReqSkills	ReqExperience	Responsibilities	Vacancies	CreatedOn	CreatedBy	Status
Job1	Project Manager	Btech/Mtech	Ms.net	5 years	Project Handling	20	10/26/2008 12:00:00 AM	ram	1
Job2	Team Leader	Btech/Mtech/BE	Ms.net,SqlServer	4 Years	Team Lead	50	10/26/2008 12:00:00 AM	ram	1
Job3	Senior Software Engineer	Bteh/Mtech/BE	Java	5 Years	Developer	100	10/26/2008 12:00:00 AM	ram	1
Job4	Programmer	Btech	Advanced Java	5 Years	Design	30	10/27/2008 12:00:00 AM	ram	1

Register New User

CONCLUSION:

From a proper analysis of positive points and constraints on the component, it can be safely concluded that the product is a highly efficient GUI based component. This application is working properly and meeting to all user requirements. This component can be easily plugged in many other systems.

BIBLIOGRAPHY

The following books were referred during the analysis and execution phase of the project

BOOK REFERENCES

1.PROGRAMMING MICROSOFT ASP .NET 2.0 APPLICATION

-WorkS Professional Guide

2.BEGINNING ASP .NET 2.0 E-COMMERCE IN C# 2005

3.Practical .Net2 and C#2: Harness the Platform, the Language, and the Framework by Patrick Smacchia.

4.SOFTWARE ENGINEERING

By Roger s. Pressman

5.UNIFIED MODELING LANGUAGE

By Grady Booch, Ranbaugh, Jacobson

6. SQL FOR PROFESSIONALS

By Jain

7. C#.NET Black Book

By Evangeleous Petersons

MSDN 2005

FOR DEPLOYMENT AND PACKING ON SERVER

www.developer.com

www.15seconds.com

FOR SQL

www.msdn.microsoft.com

FOR ASP.NET

www.msdn.microsoft.com/net/quickstart/aspplus/default.com

www.asp.net

www.fmexpense.com/quickstart/aspplus/default.com

www.asptoday.com

www.aspfree.com

www.4guysfromrolla.com/index.aspx