

Online Resume Mart

(Final Project Report)



Submitted to Prof. Ausif Mahmood

Submitted By:

Name: *Prithviraj Banerjee*

ID # 666753

Course: *CPEG 597*

INDEX

	Page
1. Abstract	3
2. Introduction	4
3. System Analysis	4
4. Identification of Need	5
5. Drawbacks of existing system	5
6. Feasibility Study.....	6
7. Software and Hardware Specification	8
8. Tools/Platforms/Language used	9
9. Server Application Development	14
10.Active Server Pages.NET	16
11.Language Support	18
12.Introduction to ASP.Net Server Control	20
13.Active X Data Objects.Net	21
14.System Design	25
15.Normalization	29
16.Relationship Diagram	32
17.Dataflow Diagram	38
18.Output Screen Shots.....	44
19.Conclusion	58
20.Bibliography	59

Abstract

The main objective of the online resume is to design and develop a web service which is a place for Job Seekers and Job Providers to meet. The data base is used to collect all the details of the job seeker and provider. Online resume collects multiple resumes from the Job seeker and the job provider is provided with the data required as per need of the company.

Introduction

Designing and Implementing Job Seekers & Job Providers

The aim of this module is to collect data from the user; he may a job seeker or a job provider. Both of them are potential clients to Resume Art. A user should be registered regardless of whether he is a job seeker or a provider. In this module we register the user and collect as many details as possible about the user.

Designing and implementing Resume Database, job database

The aim of the module is to create a resume for all job seekers asking the details about the experience, education, skills, affiliations and references for the project

Designing a resume search engine

The aim of the module is to design a dynamic search engine for the Resume art data base which can provide data for the job seekers and job providers.

SYSTEM ANALYSIS

Definition and reason for Condition Analysis

System analysis will be performed to determine if it is feasible to design an information based on policies and plans of the organization and on user requirements and to eliminate the weaknesses of the present system.

General requirements are: -

1. The new system should be cost effective.
2. To augment management, improve productivity and services.
3. To enhance User/System interface.
4. To improve information quality and usability.
5. To upgrade system's reliability, availability, flexibility and growth potential.

IDENTIFICATION OF NEED

Online Web hunt maintains information about the different job providers as well as the job seekers. It notifies every job seeker with the availability of the job as per the category in which the job seeker has registered user's resume. The system also notifies the job provider with the information about the persons registered under the category required by the job provider. It also maintains a specialized search engine which provides instant availability of the jobs as the user's category. The system maintains information of the users who have registered with the site and every user can post multiple resumes in every category. The system helps the user in formulating the resume in proper manner.

After searching the required job on the site the seekers can directly forward their resume to the corresponding email address listed in the search. This kind of functionality is again provided to the job provider who can instantly mail the candidates if one falls under their category.

Drawbacks of the Existing System:

The following are the drawbacks of the existing manual System.

Time Delay: In the existing system, information related to all transactions is stored in different registers. Since all the transactions are stored in different registers it takes lot of time to prepare different reports.

Redundancy: As the information passes through different registers, each register is consolidated and sent to next register. So the same information is being tabulated at each register, which involves lot of complication and duplication in work, thus it causes redundancy.

Accuracy: Since the same data is compiled at different sections, the possibility of tabulating data wrongly increases. Also if the data is more, validations become difficult. This may result in loss of accuracy of data.

Information Retrieval: As the information is stored in the particular Format, it can only be retrieved in the same format. But if it is to be retrieve in different format, it is not possible.

Storage Media: In the existing system, data transaction being stored are too long registers it is very difficult to refer after some time.

Reports: various reports are tabulated manually. They are not such Attractive and require more time. They do not provide adequate help in maintaining the accounts.

Enquiry: Enquiry for different level of information is much more difficult. Online enquiry of data is not possible.

FEASIBILITY STUDY

TECHINICAL FEASIBILITY:

Evaluating the technical feasibility is the trickiest part of a feasibility study. This is because, at this point in time, not too many detailed design of the system, making it difficult to access issues like performance, costs on (on account of the kind of technology to be deployed) etc. A number of issues have to be considered while doing a technical analysis.

i) Understand the different technologies involved in the proposed system:

Before commencing the project, we have to be very clear about what are the Technologies that are to be required for the development of the new system.

ii) Find out whether the organization currently possesses the required technologies:

Is the required technology available with the organization?

If so is the capacity sufficient?

For instance –

“Will the current printer be able to handle the new reports and forms required for the new system?”

OPERATIONAL FEASIBILITY:

Proposed projects are beneficial only if they can be turned into information systems that will meet the organizations operating requirements. Simply stated, this test of feasibility asks if the system will work when it is developed and installed. Are there major barriers to Implementation? Here are questions that will help test the operational feasibility of a project:

- Is there sufficient support for the project from management from users? If the current system is well liked and used to the extent that persons will not be able to see reasons for change, there may be resistance.
- Are the current business methods acceptable to the user? If they are not, Users may welcome a change that will bring about a more operational and useful systems.
- Have the user been involved in the planning and development of the project?

Early involvement reduces the chances of resistance to the system and in General increases the likelihood of successful project.

SOFTWARE AND HARDWARE SPECIFICATIONS

Hardware:

Processor	:	Intel Pentium or more
Ram	:	256 MB or more
Cache	:	512 KB
Hard disk	:	16 GB hard disk recommended for primary partition.

Software:

Operating system	:	Windows 2000 or later
Front End Software	:	ASP.NET (Visual Basic .NET)
Back End Software	:	Microsoft SQL Server

TOOLS, PLATFORM/LANGUAGES USED

SELECTED SOFTWARE

Microsoft.NET Framework:

The .NET Framework is a new computing platform that simplifies application development in the highly distributed environment of the Internet. The .NET Framework is designed to fulfill the following objectives:

- To provide a consistent object-oriented programming environment whether object code is stored and executed locally, executed locally but Internet-distributed, or executed remotely.
- To provide a code-execution environment that minimizes software deployment and versioning conflicts.
- To provide a code-execution environment that guarantees safe execution of code, including code created by an unknown or semi-trusted third party.

- To provide a code-execution environment that eliminates the performance problems of scripted or interpreted environments.
- To make the developer experience consistent across widely varying types of applications, such as Windows-based applications and Web-based applications.
- To build all communication on industry standards to ensure that code based on the .NET Framework can integrate with any other code.

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 is 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 our application or partition our application across many languages. Further, common language runtime interoperability guarantees that our existing investment in COM-based development is preserved when migrating to ASP.NET.

- ***Simplicity.*** ASP.NET makes it easy to perform common tasks, from simple form 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 our 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. This "zero 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 our 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 our 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 our applications are secure.

Language Support

The Microsoft .NET Platform currently offers built-in support for three languages: C#, Visual Basic, and JScript.

SYSTEM DESIGN

SOFTWARE ENGINEERING PARADIGM APPLIED- (RAD-MODEL)

The two design objectives continuously sought by developers are reliability and maintenance.

Reliable System

There are two levels of reliability. The first is meeting the right requirements. A careful and thorough systems study is needed to satisfy this aspect of reliability. The second level of systems reliability involves the actual working delivered to the user. At this level, the systems reliability is interwoven with software engineering and development. There are three approaches to reliability.

1. **Error avoidance:** Prevents errors from occurring in software.
2. **Error detection and correction:** In this approach errors are recognized whenever they are encountered and correcting the error by effect of error, of the system does not fail.

3. **Error tolerance:** In this approach errors are recognized whenever they occur, but enables the system to keep running through degraded perform or by applying values that instruct the system to continue process.

Maintenance:

The key to reducing need for maintenance, while working, if possible to do essential tasks.

1. More accurately defining user requirement during system development.
2. Assembling better systems documentation.
3. Using more effective methods for designing, processing, login and communicating information with project team members.
4. Making better use of existing tools and techniques.
5. Managing system engineering process effectively.

Output Design:

One of the most important factors of an information system for the user is the output the system produces. Without the quality of the output, the entire system may appear unnecessary that will make us avoid using it possibly causing it to fail. Designing the output should process the in an organized well throughout the manner. The right output must be developed while ensuring that each output element is designed so that people will find the system easy to use effectively.

The term output applying to information produced by an information system whether printed or displayed while designing the output we should identify the specific output that is needed to information requirements select a method to present the formation and create a document report or other formats that contains produced by the system.

Types of output:

Whether the output is formatted report or a simple listing of the contents of a file, a computer process will produce the output.

- A Report
- A Document
- A Message
- Retrieval from a data store
- Transmission from a process or system activity
- Directly from an output sources

Layout Design:

It is an arrangement of items on the output medium. The layouts are building a mock up of the actual reports or document, as it will appear after the system is in operation. The output layout has been designated to cover information. The outputs are presented in the appendix.

Input design and control:

Input specifications describe the manner in which data enter the system for processing. Input design features will ensure the reliability of the systems and produce results from accurate data, or thus can be result in the production of erroneous information. The input design also determines whenever the user can interact efficiently with this system.

Objectives of input design:

Input design consists of developing specifications and procedures for data preparation, the steps necessary to put transaction data into a usable form for processing and data entry, the activity of data into the computer processing. The five objectives of input design are:

- Controlling the amount of input
- Avoiding delay
- Avoiding error in data
- Avoiding extra steps
- Keeping the process simple

Controlling the amount of input:

Data preparation and data entry operation depend on people, Because labour costs are high, the cost of preparing and entering data is also high. Reducing data requirement expense. By reducing input requirement the speed of entire process from data capturing to processing to provide results to users.

Avoiding delay:

The processing delay resulting from data preparation or data entry operations is called bottlenecks. Avoiding bottlenecks should be one objective of input.

Avoiding errors:

Through input validation we control the errors in the input data.

Avoiding extra steps:

The designer should avoid the input design that cause extra steps in processing saving or adding a single step in large number of transactions saves a lot of processing time or takes more time to process.

Keeping process simple:

If controls are more people may feel difficult in using the systems. The best-designed system fits the people who use it in a way that is comfortable for them.

NORMALIZATION

It is a process of converting a relation to a standard form. The process is used to handle the problems that can arise due to data redundancy i.e. repetition of data in the database, maintain data integrity as well as handling problems that can arise due to insertion, updation, deletion anomalies.

Decomposing is the process of splitting relations into multiple relations to eliminate anomalies and maintain anomalies and maintain data integrity. To do this we use normal forms or rules for structuring relation.

Insertion anomaly: Inability to add data to the database due to absence of other data.

Deletion anomaly: Unintended loss of data due to deletion of other data.

Update anomaly: Data inconsistency resulting from data redundancy and partial update

Normal Forms: These are the rules for structuring relations that eliminate anomalies.

First Normal Form:

A relation is said to be in first normal form if the values in the relation are atomic for every attribute in the relation. By this we mean simply that no attribute value can be a set of values or, as it is sometimes expressed, a repeating group.

Second Normal Form:

A relation is said to be in second Normal form is it is in first normal form and it should satisfy any one of the following rules.

- 1) Primary key is a not a composite primary key
- 2) No non key attributes are present
- 3) Every non key attribute is fully functionally dependent on full set of primary key.

Third Normal Form:

A relation is said to be in third normal form if their exists no transitive dependencies.

Transitive Dependency: If two non key attributes depend on each other as well as on the primary key then they are said to be transitively dependent.

The above normalization principles were applied to decompose the data in multiple tables thereby making the data to be maintained in a consistent state.

Data Dictionary

After carefully understanding the requirements of the client the the entire data storage requirements are divided into tables. The below tables are normalized to avoid any anomalies during the course of data entry.

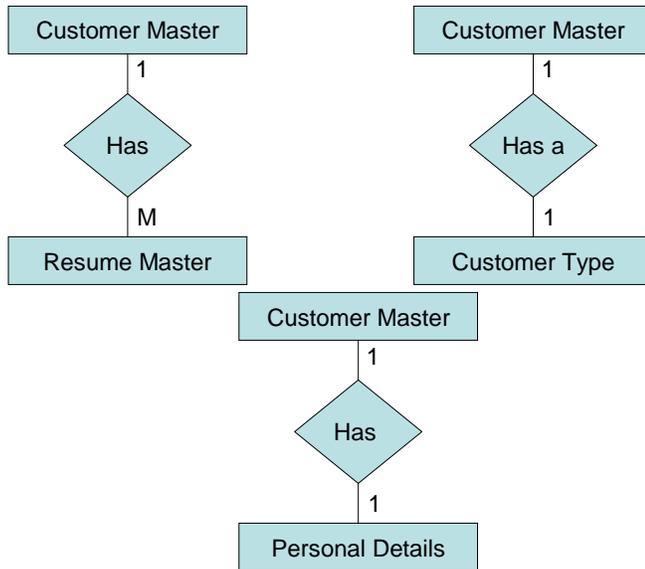
Personal Details

Field Name	Data Type	Comments
User Id	Varchar(30)	Primary Key
First Name	Varchar(30)	
Last Name	Varchar(30)	
Address 1	Varchar(25)	Not Null
Address 2	Varchar(25)	
City	Varchar(50)	Not null
State	Varchar(2)	Not null
Postal code	Number(6)	Not null
Primary phone	Varchar(14)	Not null
Secondary phone	Varchar(14)	
URL	Varchar(100)	
Career level	Varchar(3)	Not null
Marital status	Char(1)	Not null
Father Name	Varchar(30)	Not null
Passport No	Varchar(10)	
Gender	Char(1)	Not null
Date of birth	Date	Not null
Active Resume Id	Number(11)	FK

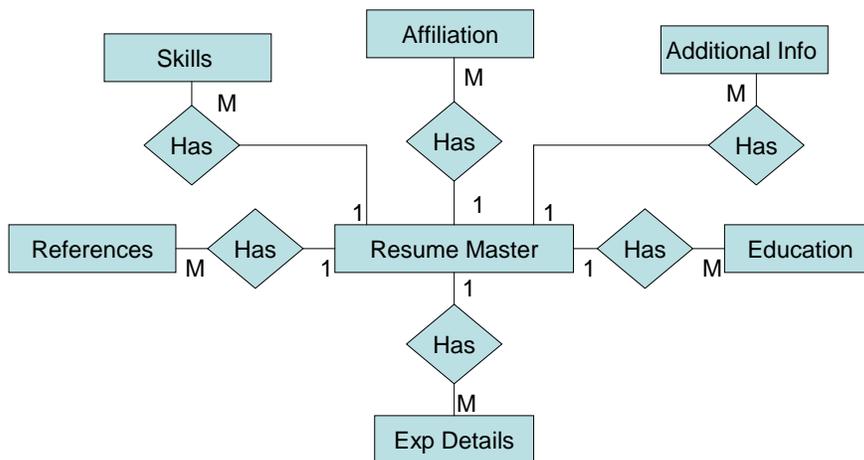
RELATIONSHIP DIAGRAM

E-R DIAGRAMS

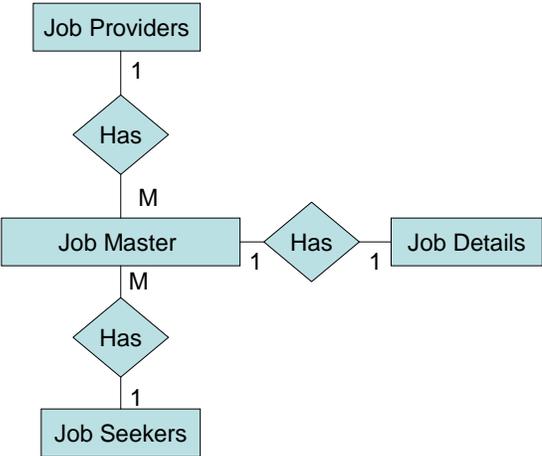
Resume Mart – E-R Diagrams



Resume Mart – E-R Diagrams

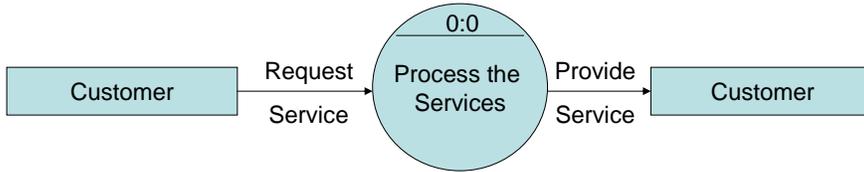


Resume Mart – E-R Diagrams

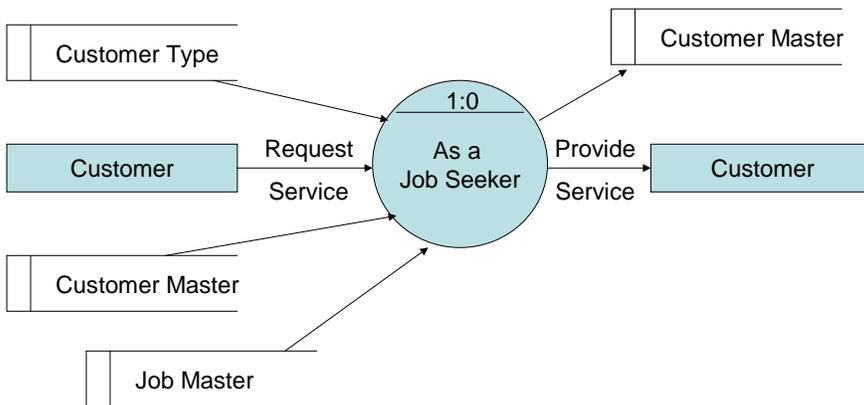


DATA FLOW DIAGRAMS

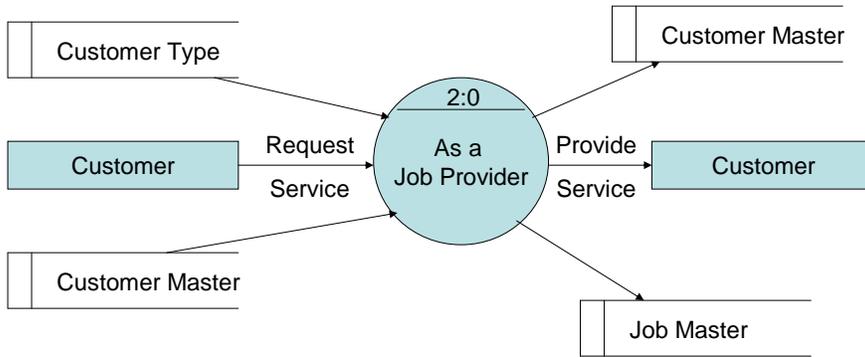
Resume Mart – Data Flow Diagrams



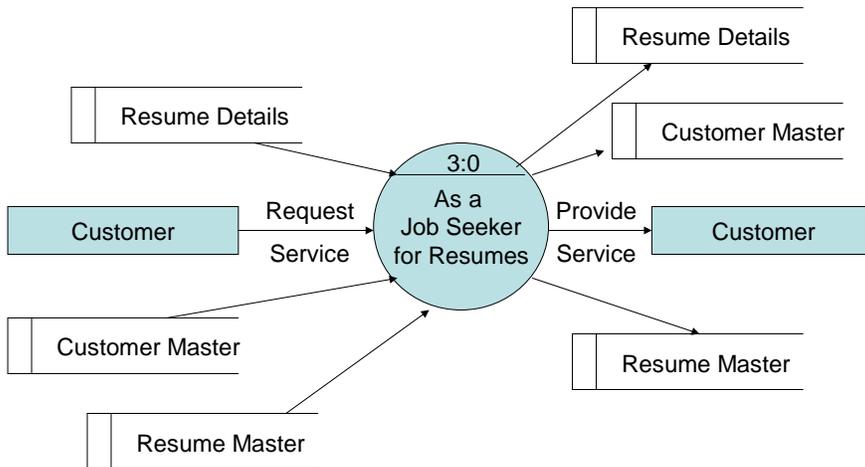
Resume Mart – Data Flow Diagrams



Resume Mart – Data Flow Diagrams



Resume Mart – Data Flow Diagrams



DATA FLOW DIAGRAM:

A data flow diagram is graphical tool used to describe and analyze movement of data through a system. These are the central tool and the basis from which the other components are developed. The transformation of data from input to output, through processed, may be described logically and independently of physical components associated with the system. These are known as the logical data flow diagrams. The physical data flow diagrams show the actual implements and movement of data between people, departments and workstations. A full description of a system actually consists of a set of data flow diagrams. Using two familiar notations Yourdon, Gane and Sarson notation develops the data flow diagrams. Each component in a DFD is labeled with a descriptive name. Process is further identified with a number that will be used for identification purpose. The development of DFD's is done in several levels. Each process in lower level diagrams can be broken down into a more detailed DFD in the next level. The top-level diagram is often called context diagram. It consists a single process bit, which plays vital role in studying the current system. The process in the context level diagram is exploded into other process at the first level DFD.

The idea behind the explosion of a process into more process is that understanding at one level of detail is exploded into greater detail at the next level. This is done until further explosion is necessary and an adequate amount of detail is described for analyst to understand the process.

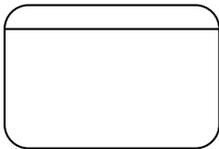
Larry Constantine first developed the DFD as a way of expressing system requirements in a graphical form, this lead to the modular design.

A DFD is also known as a "bubble Chart" has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. So it is the starting point of the design to the lowest level of detail. A DFD consists of a series of bubbles joined by data flows in the system.

DFD SYMBOLS:

In the DFD, there are four symbols

1. A square defines a source(originator) or destination of system data
2. An arrow identifies data flow. It is the pipeline through which the information flows
3. A circle or a bubble represents a process that transforms incoming data flow into outgoing data flows.
4. An open rectangle is a data store, data at rest or a temporary repository of data



Process that transforms data flow.



Source or Destination of data



Data flow



Data Store

CONSTRUCTING DFD:

Several rules of thumb are used in drawing DFD's:

1. Process should be named and numbered for an easy reference. Each name should be representative of the process.
2. The direction of flow is from top to bottom and from left to right. Data Traditionally flow from source to the destination although they may flow back to the source. One way to indicate this is to draw long flow line back to a source. An alternative way is to repeat the source symbol as a destination. Since it is used more than once in the DFD it is marked with a short diagonal.
3. When a process is exploded into lower level details, they are numbered.
4. The names of data stores and destinations are written in capital letters. Process and dataflow names have the first letter of each word capitalized

A DFD typically shows the minimum contents of data store. Each data store should contain all the data elements that flow in and out.

Questionnaires should contain all the data elements that flow in and out. Missing interfaces redundancies and like is then accounted for often through interviews.

SILENT FEATURES OF DFD's

1. The DFD shows flow of data, not of control loops and decision are controlled considerations do not appear on a DFD.
2. The DFD does not indicate the time factor involved in any process whether the dataflow take place daily, weekly, monthly or yearly.
3. The sequence of events is not brought out on the DFD.

TYPES OF DATA FLOW DIAGRAMS

1. Current Physical
2. Current Logical
3. New Logical
4. New Physical

CURRENT PHYSICAL:

In Current Physical DFD process label include the name of people or their positions or the names of computer systems that might provide some of the overall system-processing label includes an identification of the technology used to process the data. Similarly data flows and data stores are often labels with the names of the actual physical media on which data are stored such as file folders, computer files, business forms or computer tapes.

CURRENT LOGICAL:

The physical aspects at the system are removed as much as possible so that the current system is reduced to its essence to the data and the processors that transform them regardless of actual physical form.

NEW LOGICAL:

This is exactly like a current logical model if the user were completely happy with the user were completely happy with the functionality of the current system but had problems with how it was implemented typically through the new logical model will differ from current logical model while having additional functions, absolute function removal and inefficient flows recognized.

NEW PHYSICAL:

The new physical represents only the physical implementation of the new system.

RULES GOVERNING THE DFD'S PROCESS

- 1) No process can have only outputs.
- 2) No process can have only inputs. If an object has only inputs than it must be a sink.
- 3) A process has a verb phrase label.

DATA STORE

- 1) Data cannot move directly from one data store to another data store, a process must move data.
- 2) Data cannot move directly from an outside source to a data store, a process, which receives, must move data from the source and place the data into data store
- 3) A data store has a noun phrase label.

SOURCE OR SINK

The origin and /or destination of data.

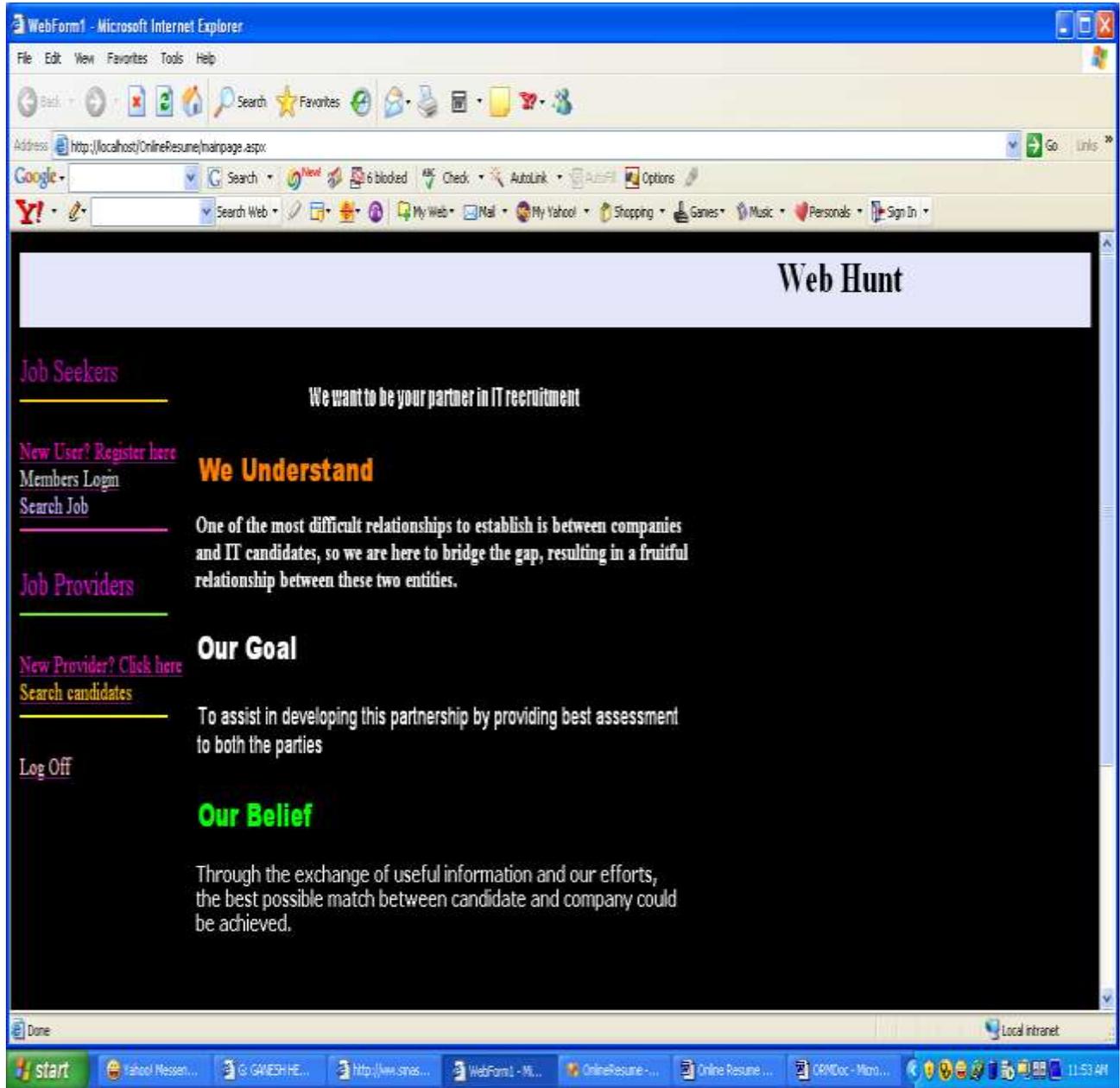
- 1) Data cannot move directly from a source to sink it must be moved by a process
- 2) A source and /or sink has a noun phrase label

DATA FLOW

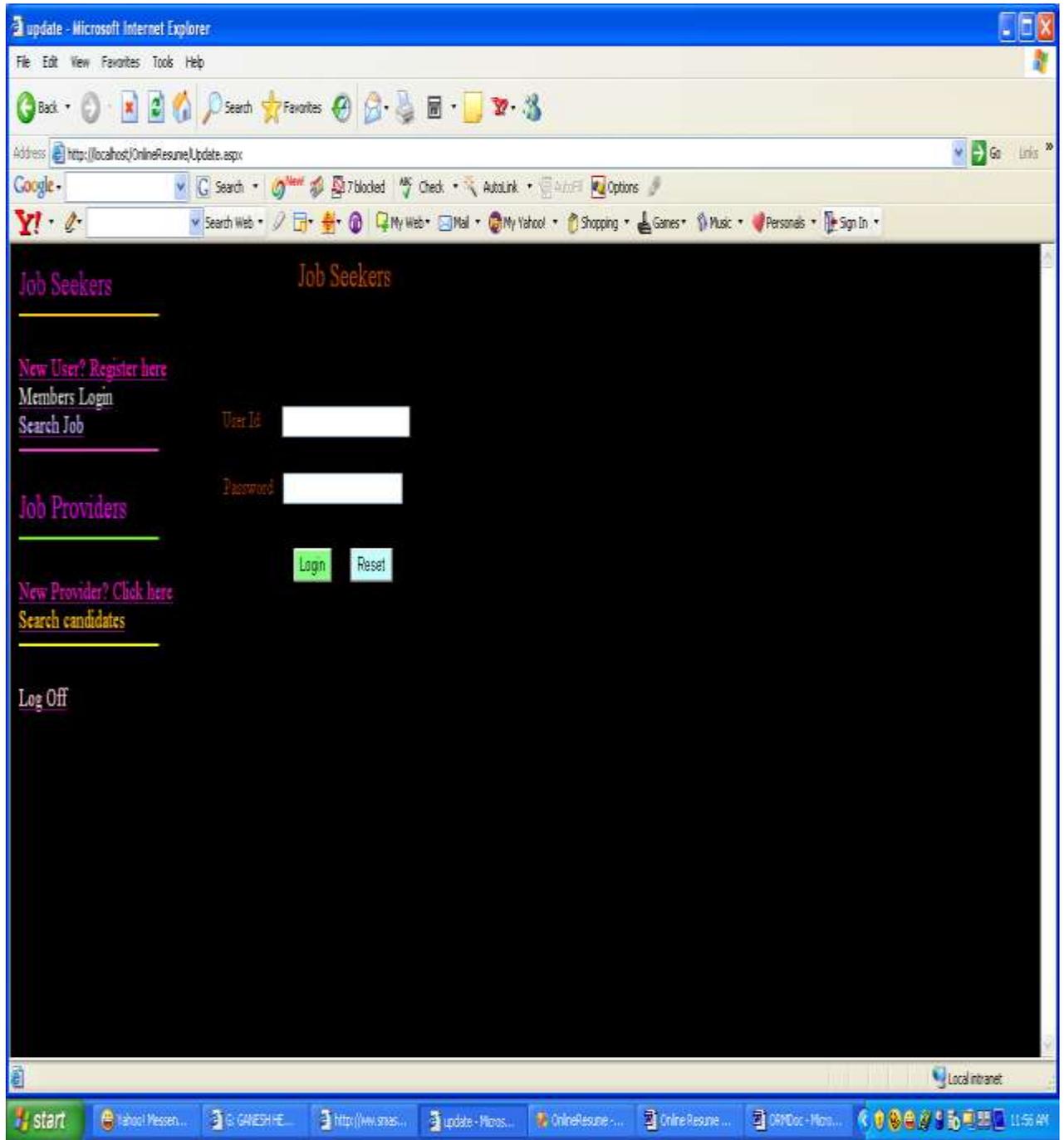
- 1) A Data Flow has only one direction of flow between symbols. It may flow in both directions between a process and a data store to show a read before an update. The latter is usually indicated however by two separate arrows since these happen at different times.
- 2) A join in DFD means that exactly the same data comes from any of two or more different processes, data store or sink to a common location.
- 3) A data flow cannot go directly back to the same process it leads. There must be at least one other process that handles the data flow produce some other data flow returns the original data into the beginning process.
- 4) A Data flow to a data store means update (delete or change).
- 5) A data Flow from a data store means retrieve or use.
- 6) A data flow has a noun phrase label more than one data flow noun phrase can appear on a single arrow as long as all of the flows on the same arrow move together as one package.

Output Screens:

Main Page



The Member Login Screen



The New User Registration Screen for Job Seeker

The screenshot shows a Microsoft Internet Explorer browser window displaying a registration page for job seekers. The page has a dark blue background with orange and white text. The main heading is "Job Seekers" in a large, stylized font. Below the heading, there are several navigation links: "New User? Register here", "Members Login", "Search Job", "Job Providers", "New Provider? Click here", "Search candidates", and "Log Off". The registration form itself consists of several input fields: "Userid", "Password", "First Name", "Last Name", "Father's Name", "DOB" (with dropdown menus for day, month, and year), "Address" (a large text area), "Pin", "City", "State", and "Phone". The browser's address bar shows the URL "http://localhost/OnlineResume/post.aspx". The taskbar at the bottom shows several open applications, including Yahoo! Messenger, a file explorer, and the browser itself.

post - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address <http://localhost/OnlineResume/post.aspx>

Google - Search

Y! Search Web

Job Seekers

Page 1-5

[New User? Register here](#)

[Members Login](#)

[Search Job](#)

Job Providers

[New Provider? Click here](#)

[Search candidates](#)

[Log Off](#)

Userid

Password

First Name

Last Name

Father's Name

DOB

Address

Pin

City

State

Phone

Local intranet

start Yahoo! Messen... G. GANEHHE... http://www.onas... post - Microsoft... OnlineResume... Online Resume... CAPDoc - Micp...

11:59 AM

The Screen of a Member who is a job seeker were he can update his data

The screenshot shows a web browser window with the address bar displaying `http://localhost/OnlineResume/save.aspx`. The page content is as follows:

Job Seekers

[New User? Register here](#)
[Search Job](#)

Job Providers

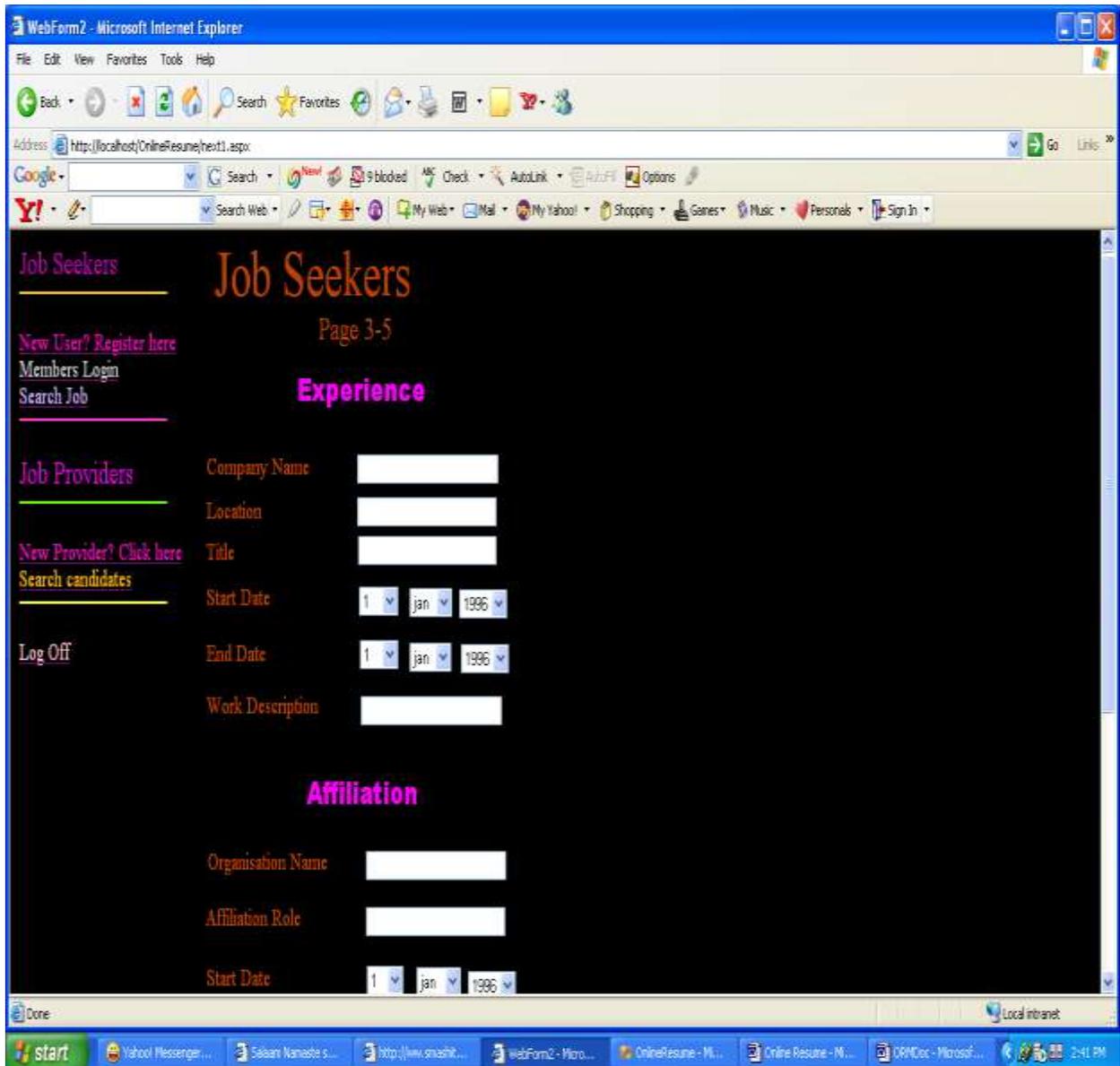
[New Provider? Click here](#)
[Search candidate:](#)

Choices	userid	password	fname	lname	fatname	dob	address	zip	city	state	phone	status	gender	passport	email	carlev
Edit	peihviraj	prtriv	peihviraj	banerjee	Amal Banerjee	2/7/1982 12:00:00 AM	Lafayette Street HYD	6604	Bridgeport	CT	2033319217	Single	M	1234	peihviraj81@yahoo.com	Master

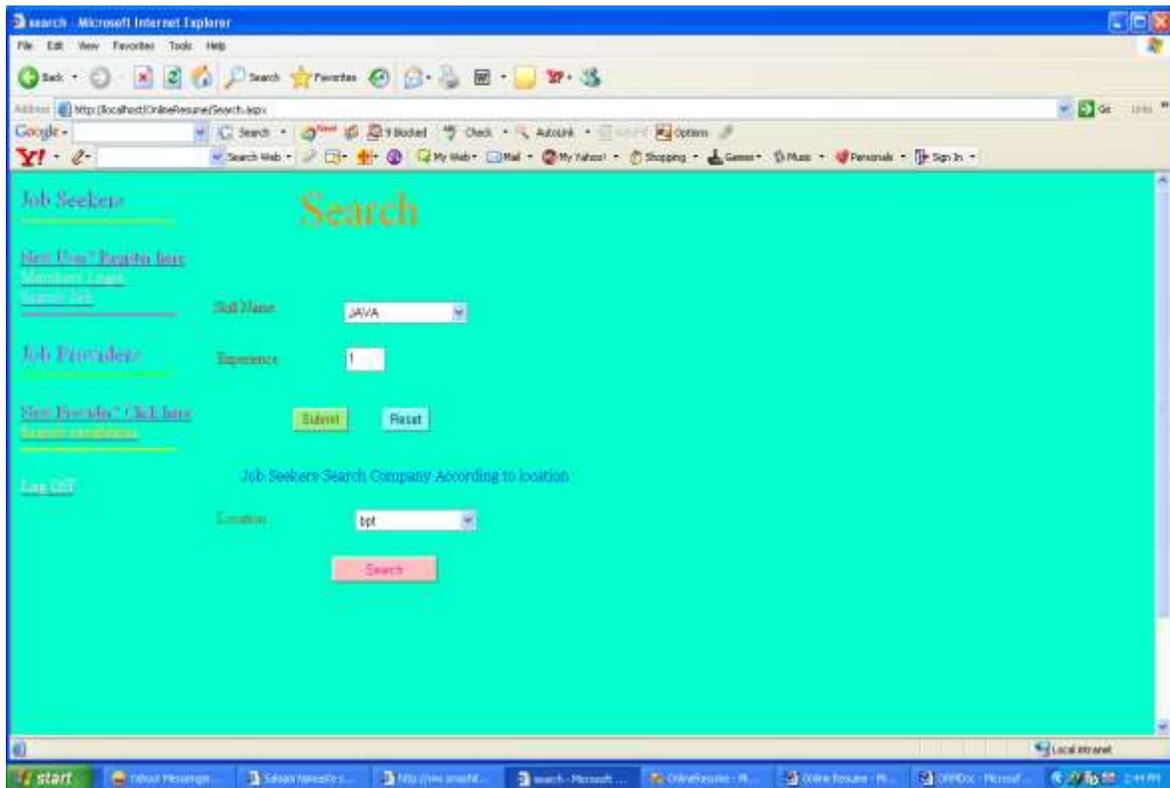
Options	RESUMEID	SCPRNAME	CITY	STATE	COUNTRY	DEGLEVEL	STARTDATE	ENDDATE	DES
Edit	1	CYGNUS	Bridgeport	CT	USA	Maters	7/9/2003 12:00:00 AM	7/12/2005 12:00:00 AM	usa

Select	RESUMEID	SKLNAME	LASTUSED	SKLLEVEL	VRSEXP	CERTIFY
Edit	1	ASP.NET	JAVA	Inter	2	JAVA

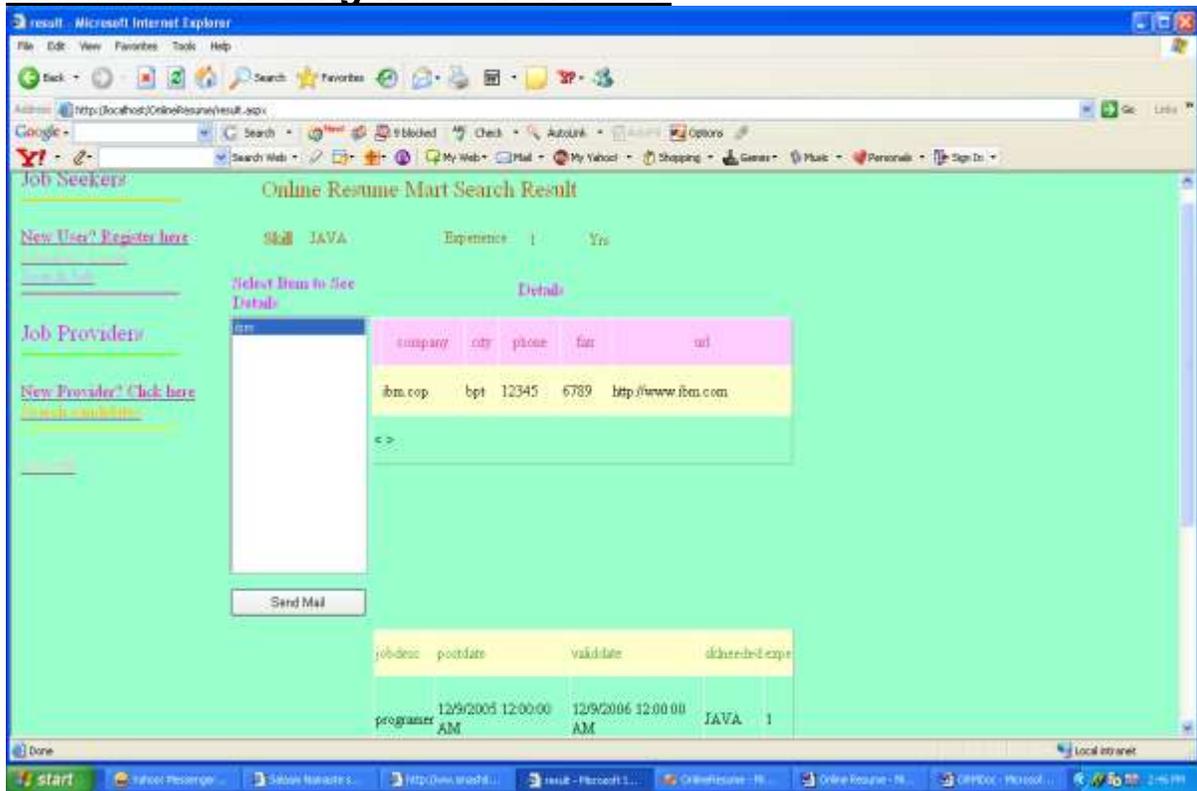
The screen of a job seeker to add up his experience and affiliation



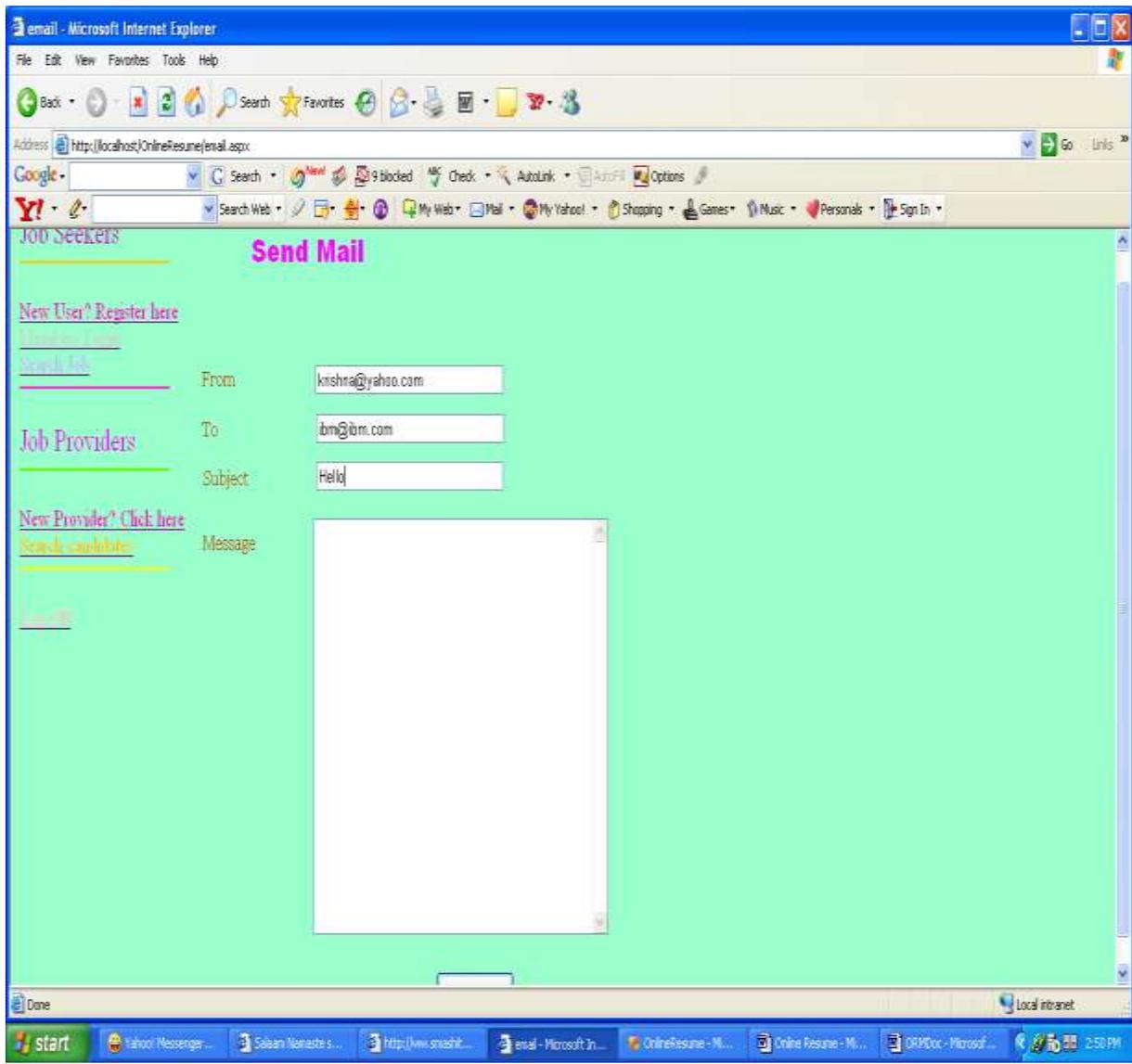
The Screen for searching job for job seekers



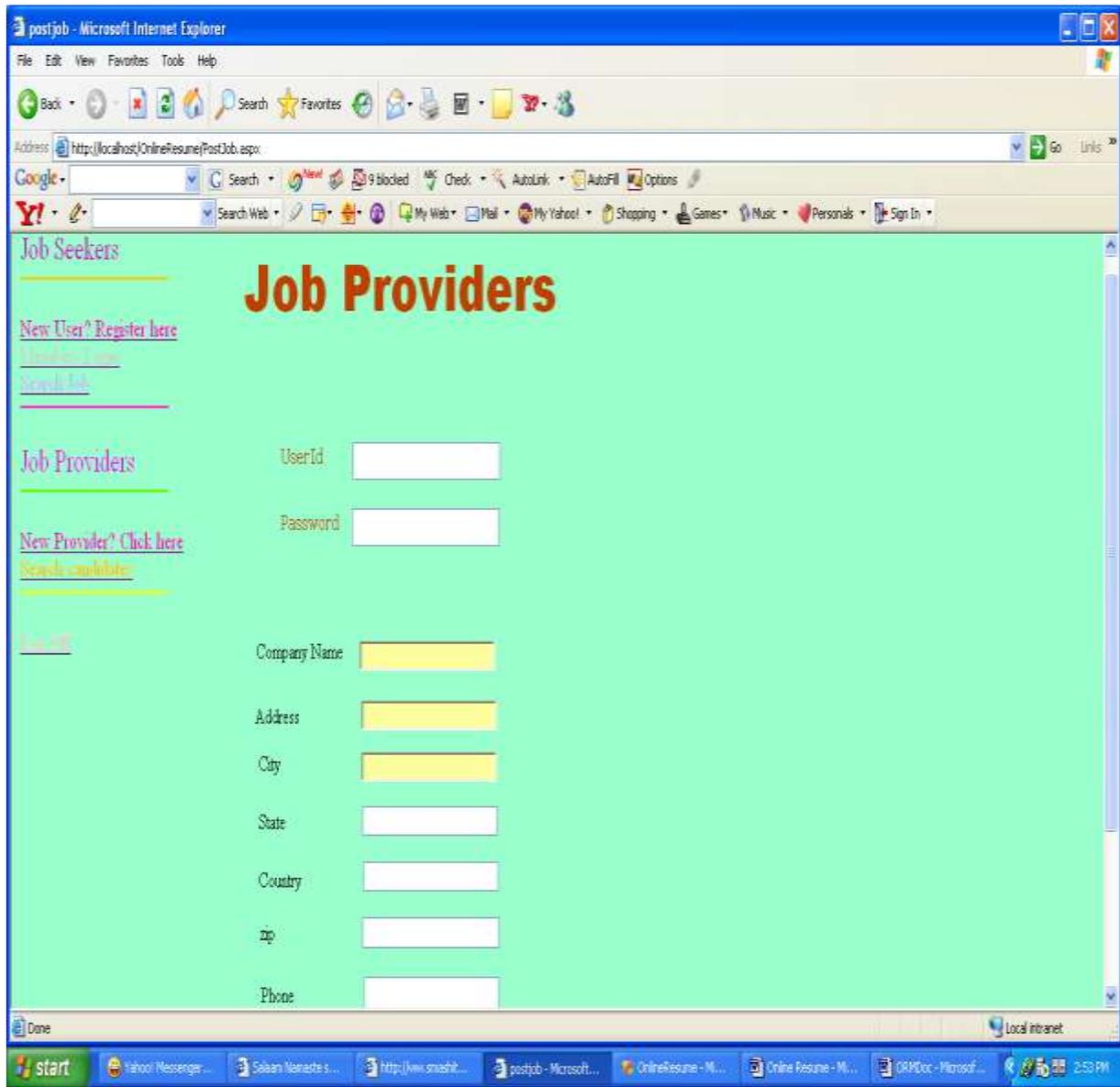
The screen showing the search result



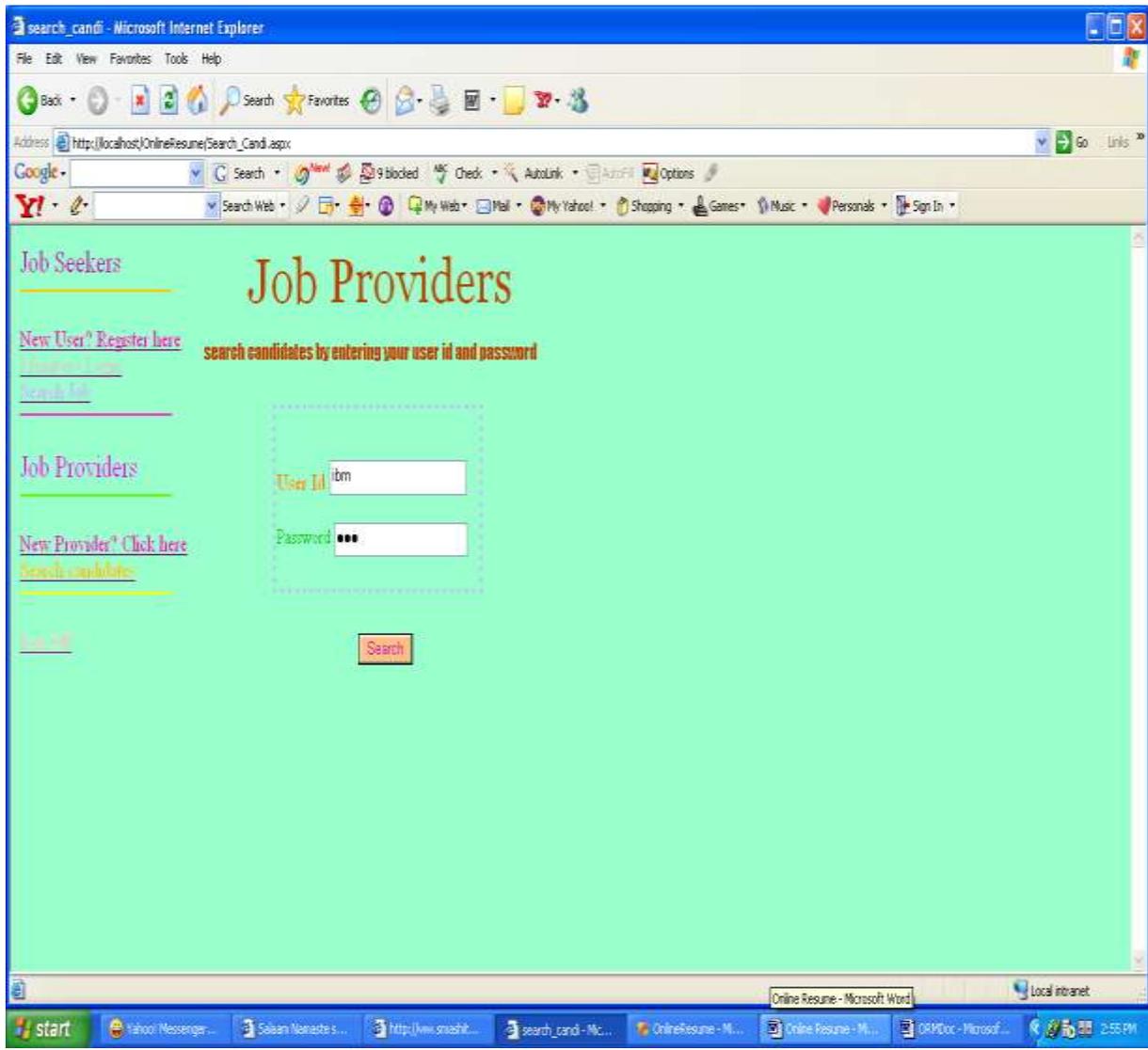
The screen showing the page from where a seeker can email the provider



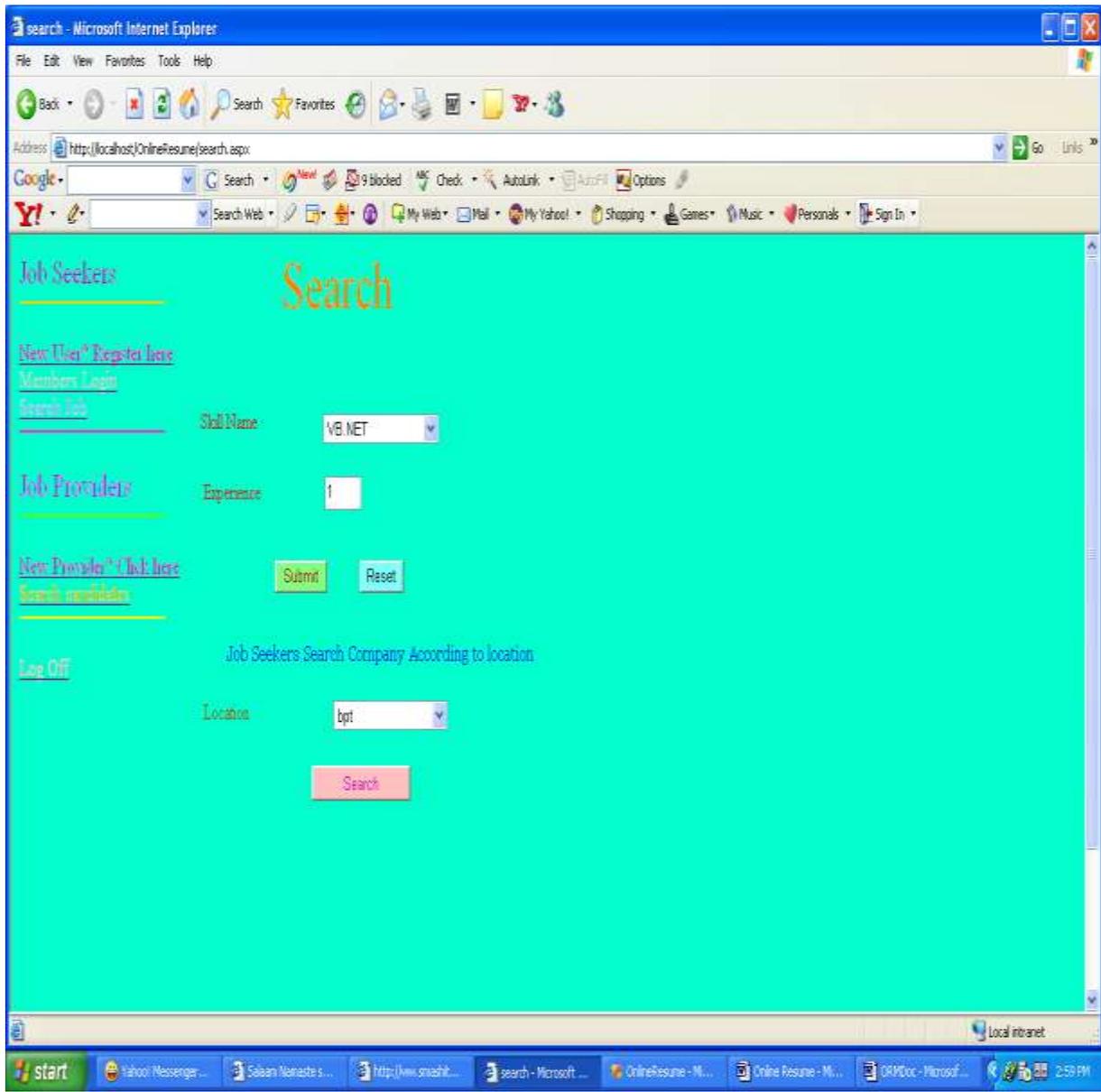
The screen showing the registration form of the job provider.



The Member logon screen for job providers



Search screen for Job providers



CONCLUSION

The “ONLINE WEB HUNT” has been successfully completed. The goal of the system is achieved and problems are solved. The package is developed in a manner that it is user friendly and required help is provided at different levels.

The project can be easily used in the process of decision making. Different types of reports can be generated which help the management to take correct decision and reduce the time delay which automatically increases the company’s work standards as well as the economical state of the company.

This system never decreases the manpower but helps the development of available manpower and optimizes the manpower by which company’s standards and capabilities can be scaled to higher dimensions.

BIBLIOGRAPHY

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

Software Engineering: By Roger.S.Pressman

Professional ASP.NET: Wrox Publication (By Alex Homer, Dave Sussman, Rob Howard and Ricahrd Anderson.

Visual Basic .NET Programming: Black Book (By Steven Holzner)

MSDN 2002: By Microsoft

Referred Online.

SQL School: <http://www.w3schools.com/sql>

MSDN: <http://msdn.microsoft.com>