

Spring 4-1-2009

Resource Outsourcing

Sowjanya Guntupalli
Dakota State University

Follow this and additional works at: <https://scholar.dsu.edu/theses>

Recommended Citation

Guntupalli, Sowjanya, "Resource Outsourcing" (2009). *Masters Theses*. 161.
<https://scholar.dsu.edu/theses/161>

This Thesis is brought to you for free and open access by Beadle Scholar. It has been accepted for inclusion in Masters Theses by an authorized administrator of Beadle Scholar. For more information, please contact repository@dsu.edu.

RESOURCE OUTSOURCING

A graduate project submitted to Dakota State University in partial fulfillment of the requirements for the degree of

Master of Science

In

Information Systems

April, 2009

By

Sowjanya Guntupalli

Project Committee:

Dr. Ronghua Shan

Dr. Stephan Krebsbach

Dr. Mark Moran





PROJECT APPROVAL FORM

We certify that we have read this project and that, in our opinion, it is satisfactory in scope and quality as a project for the degree of Master of Science in Information Systems.

Student Name: Sowjanya Guntupalli

Master's Project Title: Resource Outsourcing

Faculty supervisor: Ronghua Shem Date: 12/17/08

Committee member: Mark Moran Date: 12/17/08

Committee member: Steph Kulus Date: 12/17/08

ABSTRACT

Resource outsourcing is developed for creating an interactive job vacancy for candidates. In examining one of the fastest growing trends in the business world today, namely outsourcing to service providers, we find evidence of positive market reaction to HR service provider contract announcements, and we find some evidence of differential effects on subsequent firm performance.

This web application is to be conceived in its current form as a dynamic site- requiring constant updates both from the seekers as well as the companies. On the whole the objective of the project is to enable jobseekers to place their resumes and companies to publish their vacancies.

DECLARATION

I hereby certify that this project constitutes my own product, that where the language of others is set forth, quotation marks so indicate, and that appropriate credit is given where I have used the language, ideas, expressions or writings of another.

I declare that the project describes original work that has not previously been presented for the award of any other degree of any institution.

Signed,

Sowjanya Guntupalli

Table of Contents

PROJECT APPROVAL FORM.....	II
ABSTRACT	III
LIST OF TABLES.....	VII
LIST OF FIGURES.....	VIII
INTRODUCTION	ERROR! BOOKMARK NOT DEFINED.
Organization Profile.....	1
About the project	1
SYSTEM ANALYSIS	ERROR! BOOKMARK NOT DEFINED.
Definition of the problem	3
Existing system	3
Proposed System.....	3
Feasibility study.....	4
SYSTEM DESIGN	ERROR! BOOKMARK NOT DEFINED.
Design Description	6
Project Modules	7
Hardware and Software Requirements	9
Data tables.....	11
TESTING AND IMPLEMENTATION	ERROR! BOOKMARK NOT DEFINED.

Testing	20
Implementation	24
CONCLUSION.....	ERROR! BOOKMARK NOT DEFINED.
REFERENCES.....	26
APPENDIX A: USERS' MANUAL.....	27
UML Diagrams	27
Sequence Diagrams	29
E-R Diagrams	31
FIGURE 5 : ATTRIBUTES OF EXPERIENCE.....	33
Screenshots	38
APPENDIX B: SYSTEM TECHNICAL.....	59
DOCUMENTATION	59
Overview of Java Technology	59
About HTML	63
Java Script.....	63
Overview of Oracle and SQL/PLSQL	85
PL/SQL.....	91
JDBC (Java Database Connectivity).....	95
JDBC Driver Types	96

LIST OF TABLES

Table 1 : Jobseekers	11
Table 2: Client	13
Table 3: Job Provider	14
Table 4: Education.....	16
Table 5: Experience.....	17
Table 6 : Job Order.....	18
Table 7: Projects.....	19
Table 8: Resume.....	19

LIST OF FIGURES

Figure 1: Use Case Diagram	28
Figure 2:Sequence Diagram of Administrator	29
Figure 3:Sequence Diagram of Job Seeker	30
Figure 4: E-R diagram of a Job Seeker	32
Figure 5 : Attributes of Experience	33
Figure 6: Attributes of Project.....	34
Figure 7: Attributes of Resume	35
Figure 8 : E-R Diagram of Job Order.....	37

CHAPTER 1

INTRODUCTION

Organization Profile

Business intelligence Solutions Inc. (BIS) is the leading solution provider firm focused solely on helping clients successfully execute their business strategies. Our complementary combination of expertise and principles of Strategy Focused Organization and Corporate Performance Management methodologies and technologies is changing the way the firms around the world manage their business.

BIS provides strategic consulting, application services, technology solutions and managed services to Global 1000 companies and government organizations. We help customers achieve results by identifying mission critical issues and implementing innovative and customized solutions designed to generate revenue, reduce costs and access the right information at the right time.

About the project

The actual problem is to create a website for the Job Provider. It is developed for creating an interactive job vacancy for candidates. This web application is to be conceived in its current form as a dynamic site-requiring constant updates both from the seekers as well as the companies. On the whole the objective of the project is to enable jobseekers to place their resumes and companies to publish their vacancies. It enables jobseekers to post their resume, search for jobs, view personal job listings. It will

provide various companies to place their vacancy profile on the site and also have an option to search candidate resumes. Apart from this there will be an administration module for the customer to make changes to the database content. It consists of 5 modules:

The users of this system are searching for job, registration their personal, educational, skills, project and resume details. This system is designed in such a way that the users can easily interact with the system with minimum knowledge to browser the net and company rules.

CHAPTER 2

SYSTEM ANALYSIS

Definition of the problem

To create or develop a new system first we have to study the prior system, Analysis difficult problems faced by the operator of that system. System Analysis therefore understands such problems and proposes a new system in which the above problems are rectified.

Existing system

Before creating this website, all jobseekers send their resumes or information through mails or they use person to person contacts with each other. It will take long time to send their requirements through this type of communications. This kind of process is prone to errors. The administration faces the problems to collect all the information from clients and Job Providers to analyze the requirement in the corresponding Clients. Administration has to send requirements information to different Job Providers and jobseekers.

Proposed System

Our Proposed system provides an online solution for the traditional method of hiring a suitable employee. The proposed website consists of five main modules that deal with the Job Seeker, Job Provider, Client, Administrator and the Search and Help options. It simply acts as a bridge between all the user classes.

Feasibility study

It is necessary and prudent to evaluate the feasibility of a project at the earliest possible time. There may be different ways of checking whether a system is feasible or not. The following feasibility studies were performed to gauge the system feasibility.

Operational Feasibility

In this test, the operational scope of the system is checked. The system under consideration should have enough operational reach. It is observed that the proposed system is very user friendly and since the system is built with enough help, even persons with little knowledge of windows can find the system very easy.

Technical Feasibility

This test includes a study of function, performance and constraints that may affect the ability to achieve an acceptable system. This test begins with an assessment of the technical viability of the proposed system. One of the main things to be assessed is the need of various kinds of resources for the successful implementation of the proposed system.

Economical Feasibility

An evaluation of development cost weighed against the ultimate income or benefit derived from the development of the proposed system is made. Care must be taken that incurred in the development of the proposed of the system should not exceed from the system. The income can be in terms of money or goodwill, since the software brings in both, the system is highly viable.

CHAPTER 3

SYSTEM DESIGN

Design Description

Design is essentially a blue print or it acts as a bridge between the requirement specification and the final solution for satisfying the requirements. Based on the work-flow described in Appendix-A we can draw the following conclusions for the Software System that has to be developed:

The System needs to be a web-based system so that it allows the Job Providers, clients & jobseekers to access the company database over the Internet.

Being a web-based system also enables the Company staff to send e-mails immediately to Suppliers, whenever a requirement for Parts arises.

An added advantage is since the e-mail is delivered instantly, there could be instant responses from the Jobseekers.

The whole process depends on communications between jobseekers & the Administrators, different clients & the consultants. If all these communications are done through a web-based system, then the time period for the whole process can be considerably brought down.

The System needs to store the details of all the information (personal, education, skills, experience, projects etc) held by all the Jobseekers.

The System needs to store the details of all the requirements held by different clients.

The System needs to store the details of all the jobs held by Job Providers.

Since it is a web-based system, a Login authorization should be provided so that Job Providers, jobseekers, and clients will be able to lookup and use options that are specific to them.

The System should allow the Clients to enter their Requirements.

The System should allow the Job Provider to provide for jobs for jobseekers.

The System should provide an option to generate a client Report.

The System should provide an option to generate a Job Providers Report.

The System should provide an option to short list applicants Report.

The System should provide an option to generate selected applicants Report.

Project Modules

The proposed system is developed by using five modules:

1. Job Seeker.
2. Job Provider.
3. Client.
4. Administrator.
5. Job Search. And Help.

1. Job Seeker.

This module contains details about Job Seeker, i.e. employee or un-employee details. Like employee name, email, experience etc.. Here employee can do update, modify and delete, He can update experience and skills details also.

2. Job Provider.

This module having information about job provider and requirement details, which client recruiting the employees? And what based them recruiting the employees? Here client releasing the primary skills, experience, no. of vacancies, opening date and closing date.

3. Client.

This module contains details about the Clients, Opening date and Client profile etc...

4. Administrator.

The administrator module having all privileges about this entire project, he can update, delete, and modify the details about job seeker, job provider, client and Job Search details. Administrator maintains the client and job seeker database, where ever client is releasing their requirements (vacancies) with particular primary skills and experience, on that time administrator search for job seekers, who are having that primary skills and experience. Administrator sends the message for selected candidates.

5. Job Search and Help:

This module is having all current vacant jobs and details like which client is offering that job etc... So that one can search appropriate jobs.

Module connectivity:

In the administrator module the administrator will be responsible for the registering the Job Providers and clients at the site. This module is also responsible for search for skilled applicants, shortlist the applicants and send the call letters to the applicants. In the jobseekers module the new user can registration their information, or existing jobseeker can update their information, search for job based on skills or experience.

The client module, different clients are fetching new job lists, and no. of vacates, opening date and closing date.

Hardware and Software Requirements

The development of this project deals with the following environment

- Hardware requirements
- Software requirements

Hardware Requirements

The selection of hardware is very important in the existence and proper working of any software. In the selection of hardware, the size and the capacity requirements are also important. The Resource outsourcing can be efficiently run on Pentium system with at least 128 MB RAM and Hard disk drive having 20 GB. Floppy disk drive of 1.44 MB and 14 inch Samsung color monitor suits the information system operation.(A Printer is required for hard copy output).

Intel processor ----- 233 MHZ or above

RAM Capacity ----- 256MB

Hard Disk ----- 20GB

Software Requirements

One of the most difficult tasks is that, the selection of the software, once system requirement is known then we have to select a particular software package that fits the requirements. After initial selection, further security is needed to determine the desirability of particular software compared with other candidates. This section first summarizes the application requirement question and then suggests more detailed comparisons.

- Operating System ----- Windows 95/98/NT/2000
- Browser ----- IE, Mozilla.
- Web/Application Server ----- BEA Web Logic 7/Tomcat
- Database Server ----- Oracle
- Database Connectivity ----- JDBC
- Other Tools & Technologies ----- Java (JDK), Servlets (JSDK), JSP, HTML

Data Tables

Table 1 : Jobseekers

SL.NO	FIELD NAME	DATA TYPE	DESCRIPTION
1	USERID	Varchar2(10)	This is the Primary Key of the jobseekers table to identify them uniquely.
2	PASS	Varchar2(20)	This the password of the Jobseeker
3	FNAME	Varchar2(40)	This is the First name of the jobseeker
4	MNAME	Varchar2(30)	This is the Middle name of the jobseeker
5	LNAME	Varchar2(20)	This is the Last name of the jobseeker
6	PRSTADD	VARCHAR2(30)	This is the present address of the jobseeker
7	PARTADD	VARCHAR2(30)	This is the Permanent address of the jobseeker
8	ZIPCODE	VARCHAR2(30)	This is the Zip code number of the jobseeker
9	CITY	VARCHAR2(30)	This is the city of the jobseeker
10	STATE	VARCHAR2(30)	This is the state of the jobseeker

11	COUNTRY	VARCHAR2(30)	This is the Country of the jobseekers
12	CTNTNOOFICE	VARCHAR2(30)	This is the off-contact address of the jobseekers
13	EXTNO	VARCHAR2(30)	This is the ext number of the jobseekers
14	CTNTNORESNO	VARCHAR2(30)	This is the res ph no of the jobseekers
15	PASSPORTNO	VARCHAR2(30)	This is the passport number of the jobseekers
16	CARROBJ	VARCHAR2(200)	This is the career object of the jobseekers
17	MOBILENO	VARCHAR2(30)	This is the contact mobile no of the jobseekers
18	GENDER	VARCHAR2(30)	This is the gender of the jobseeker

This table is used to enter new jobseeker's information.

Table 2: Client

SL.NO	FIELD NAME	DATA TYPE	DESCRIPTION
1	CLIENTID	Varchar2(10)	This is the Primary Key used for the Client's table.
2	CLNAME	Varchar2(20)	This the Name of the client
3	ADDRESS	Varchar2(40)	This is the Address of the client
4	CITY	Varchar2(40)	This is the city of the client
5	STATE	Varchar2(40)	This is the state of the client
6	NATION	Varchar2(40)	This is the nation of the client
7	NOOFBRANCH	Varchar2(40)	This is the no of branches of the client
8	CLTYPE	Varchar2(40)	This is the client type.
9	CONTPHNO	Varchar2(40)	This is the contact ph number of the client
10	EXT	Varchar2(10)	This is the ext-number of the client
11	COMDESC	Varchar2(200)	This is the client description of the client
12	CTPERSON	Varchar2(40)	This is the contact person of the client
13	WEBSITE	Varchar2(40)	This is the website of the client
14	PASS	Varchar2(40)	This is the password of Client.

This table is used to enter client's information.

Table 3: Job Provider

SL.NO	FIELD NAME	DATA TYPE	DESCRIPTION
1	CONID	Varchar2(20)	This is unique identifier given to a Job Provider to identify it uniquely. This is the Primary Key of the table.
2	CONNAME	Varchar2(20)	This the Name of the Job Provider
3	CONPERSION1	Varchar2(20)	This the Contact person Name of the Job Provider
4	CONPERSION2	Varchar2(20)	This the Contact person Name of the Job Provider
5	CONPERSION3	Varchar2(10)	This the Contact person Name of the Job Provider
6	PHNO	Varchar2(20)	This the Contact phone no of the Job Provider
7	ADDRESS	Varchar2(20)	This the Contact address of the Job Provider
8	CITY	Varchar2(20)	This the City of the Job Provider
9	STATE	Varchar2(20)	This the state of the Job Provider
10	NATIONAL	Varchar2(20)	This the nation of the Job Provider
11	FIELD1	Varchar2(20)	This the field type in the Job Provider
12	FIELD2	Varchar2(20)	This the field type in the Job Provider

13	FIELD3	Varchar2(20)	This the field type in the Job Provider
14	STANDARD	Varchar2(20)	This the standard of the Job Provider
15	DESC1	Varchar2(200)	This the description of the Job Provider
16	PASS	Varchar2(20)	This the password of the Job Provider

This table is used to enter Job Provider's information.

Table 4: Education

SL.NO	FIELD NAME	DATA TYPE	DESCRIPTION
1	QUL	Varchar2(10)	This the height Qualification of the specific jobseeker
2	GRAD	Varchar2(20)	This tells about the Graduation of the jobseeker
3	GUN	Varchar2(20)	This is the Graduation university of the jobseeker
4	GGPA	Number	This tells about the GPA of the jobseeker
5	UNDERGRAD	Number	This is the Under-Graduation of the jobseeker
6	UGUN	Varchar2(20)	This the Under- Graduation university of the jobseeker
7	UGGPA	Varchar2(20)	This the Under-Graduation GPA of the jobseeker
8	HIGHS	Varchar2(20)	This is the High school qualification of the jobseeker
9	HIGHCERTNO	Varchar2(20)	This the High school certificate number of the jobseeker
12	OTHERQUL	Varchar2(20)	This the other qualification of the jobseeker
13	OTHERCERT	Varchar2(20)	This the other certification of the jobseeker
14	USERID	Varchar2(20)	This the user id of the jobseeker, This is the Primary Key of the table

Table 5: Experience

SL.NO	FIELD NAME	DATA TYPE	DESCRIPTION
1	USERID	Varchar2(20)	This is the primary key of the experience table to identify them uniquely
2	EXPE	Varchar2(20)	This is the experience of the jobseeker.
3	FIELD	Varchar2(15)	This is the field of the jobseeker
4	PRESENTWORKING	Varchar2(15)	This is the present working company of the jobseeker
5	OFFORONSITE	Varchar2(20)	This is the offsite/onsite of the jobseeker doing projects.
6	EXPFROM	Varchar2(20)	This is the experience starting date of the jobseeker
7	EXPTO	Varchar2(20)	This is the experience till end date of the jobseeker
8	PRESENTSAL	Varchar2(20)	This is the present salary of the jobseeker
9	EXPTSAL	Varchar2(20)	This is the expected salary of the jobseeker
10	WWW	Varchar2(20)	This is the website of the jobseeker working company

This table is used for store the experience details of the jobseeker.

Table 6 : Job Order

SL.NO	FIELD NAME	DATA TYPE	DESCRIPTION
1	JOBTITLE	Varchar2(20)	This is the job title.
2	JOBCODE	Varchar2(10)	This is Primary key along with the ClientID & the Tdate fields
3	SKILL1	Varchar2(20)	This is the skill1 of required job.
4	SKILL2	Varchar2(20)	This is the skill2 of required job.
5	SKILL3	Varchar2(20)	This is the skill3 of required job.
6	SKILL4	Varchar2(20)	This is the skill4 of required job.
7	SKILL5	Varchar2(20)	This is the skill5 of required job.
8	SKILL6	Varchar2(20)	This is the skill6 of required job.
9	NOYEXP	Varchar2(20)	This is the number of experience of required job.
10	RELDATE	Varchar2(20)	This is the releasing date of the required job.
11	CLODATE	Varchar2(20)	This is the closing date of the required job.
12	CLIENTID	Varchar2(20)	This is the client id.

This table is used for the maintenance of the Job order's information.

Table 7: Projects

SL.NO	FIELD NAME	DATA TYPE	DESCRIPTION
1	USERID	Varchar2(20)	This is the Unique identifier given to User.
2	PROJTITLE	Varchar2(20)	This is the project title.
3	PROJDESC	Varchar2(20)	This is the project description
4	USED SKILLS	Varchar2(20)	This is the used skills of project
5	TEAMSIZ	Varchar2(20)	This is the team size
6	ENVRN	Varchar2(20)	This is the project environment
7	ROLLS	Varchar2(20)	This is the project rolls

This table is used to maintain project information

Table 8: Resume

SL.NO	FIELD NAME	DATA TYPE	DESCRIPTION
1	USERID	Varchar2(20)	This is unique identifier given to a user.
2	Resume	BLOB	This is to store resume.

This table is used to store resume information.

CHAPTER 4

TESTING AND IMPLEMENTATION

Testing

Testing plays a critical role for quality assurance and for ensuring the reliability of the software. Its basic function is to detect the errors. After the coding phase, testing is done to test the proper working of the new system. Testing is the process of executing a program with the intention of finding errors. It is a complete verification to determine whether the objectives are met and the user requirements are satisfied. The testing phase involves testing of a system using various test data. Preparation of the test data plays a vital role in the system testing. After preparing the test data, the system under study is testing using those test data. Errors were found and corrected by using the following testing steps and corrections are recorded for future references. Thus, a series of testing is performed on the system before it is ready for coding. Since code is the only product that can be executed frequently whose actual behavior can be observed, this phase is so important for the successful implementation of the software product. Thus, the goal of testing is to uncover the requirements, design and coding errors in the program.

Unit Testing

The first step in the testing is the unit testing. Unit test is normally considered as an adjunct to the coding step. After the coding has been developed, received and verified for correct syntax, unit testing begins. The standalone modules were tested individually for their correct functionality, with the corresponding data. This ensures the reliability of the modules when integrated. Each and every module is tested independently with sample data and it was found that all modules are properly functioning. Using the unit test plans, prepared in the design phase of the system as a guide, important control paths are tested to uncover errors within the boundary of the modules. Boundary conditions were checked, all independent paths were exercised to ensure that all statements in the module are checked at least once and all error handling paths were tested. Each unit was thoroughly tested to check if it might fall in any possible situation. This testing was carried out during the programming itself. At the end of this testing phase, each unit was found to be working satisfactory, as regard to the expected output from the module.

Integration Testing

The second step in the testing process is the Integration testing. Integration testing is the systematic technique for constructing the program structure while conducting tests to uncover errors associated with interfacing. All the modules when unit testing will work properly but after interfacing the data can be lost across an interface, one module can have an inadvertent, adverse effect on other, sub functions when combined may not produce the desired major function, global data structures can cause problems, etc.

Integration testing was performed by integrating all the individual modules and the activities of the user such as loading layers, retrieving information from any functions applying themes based on the records present in the database etc. and is found that it works good to the examination of the end users. Hence, the objective of integration testing is to take unit tested modules and build a final program structure.

All the modules developed are independent. Even the whole process of approval for all. Each module is integrated well with other modules. And all the interfaces are tested successfully.

Functional Testing

This test involves testing the system under typical operating conditions with sample input values. Functional testing was performed on the system by giving existing industry id or plot number and a null or string as the input for any field in which case the user should be redirected to the same state with the appropriate message, rather than proceeding and crashing in the system.

Functional testing was performed on the system by raising the demand with an eye to check all the validations. The total processing of the system is satisfactory with the following results.

All the validations are clearly notified to the user regarding jobseekers registration, new client registration, job order, job providers, and job search preparation etc.

Almost all the functional errors, data storage errors and all types of logical errors are tested successfully.

Acceptance Testing

User acceptance test of a system is the factor for the success of the system. The system under consideration was listed for user acceptance by keeping constant touch with the perspective user of the system at the time of design, development and making changes whenever required for unit testing.

The requirements of the customer are gathered at regular intervals at the developing site itself. The problems that are to be visualized through this tool are been gathered by the customer and are reported. The user at the user's site carried this test. Live data entered and the system's output was compared with what was manually prepared. Here the system has met the user's requirement in the following fields:

Data Entry

Error Handling

Reporting and corrections

Data Access Protections

System Output

Implementation

Implementation includes all those activities that take place to convert the old system to the new system. The new system will replace the existing system. The aspects of implementation are as follows. Conversion, Post Implementation Review.

Conversion

Conversion means changing from one system to another. The objective is to put the tested system into operation. It involves proper installation of the software package developed and training the operating staff. The software has been installed and found to be functioning properly. The users are trained to handle the system effectively. Sample data provide to the operating staff and were asked to operate on the system. The operating staffs now have a clear out look of the software and are ready for practical implementation of the package.

Post Implementation Review

A post implantation review is an evaluation of system in terms of the extent to which the system accomplishes the stated objectives. This starts after the system is implemented and conversion is complete.

CHAPTER 5

CONCLUSION

This system has been developed successfully incorporate all the requirements. Appropriate care has taken during database design maintain database integrity and to avoid redundancy of data. This site was developed in such a way that any further modifications needed can be easily done. User feels freely while using this site. In this all technical complexities are hidden. This site is a more user friendly.

I was involved in the design, development and implementation of the project like gathered, analyzed and documented business requirements to identify the core functionality of the project, User creation of tables, Database links, Writing and Implementation of HTML code.

The quality features like correctness, efficiency, usability, maintainability, portability, accuracy, errors free, tolerance, expandability and communicatively all are successfully done.

REFERENCES

1. Kathy Sierra, Bert Bates(2005). *Head First Java: Your Brain on Java-A learner's Guide*. 2nd. California: O'Reilly & Associates Inc.
2. Scott Urman, Ron Hardman, Michael McLaughlin(2004). *Oracle Database 10g PL/SQL Programming*. 1st. McGraw-Hill.
3. Pressman, R. S.(1996). *Software Engineering: A Practitioner's Approach*. 4th. McGraw-Hill Higher Education.
4. *Java Overview*. Retrieved from Sun Microsystems Web site:
<http://java.sun.com/j2ee/overview.html>

APPENDIX A: USERS' MANUAL

UML Diagrams

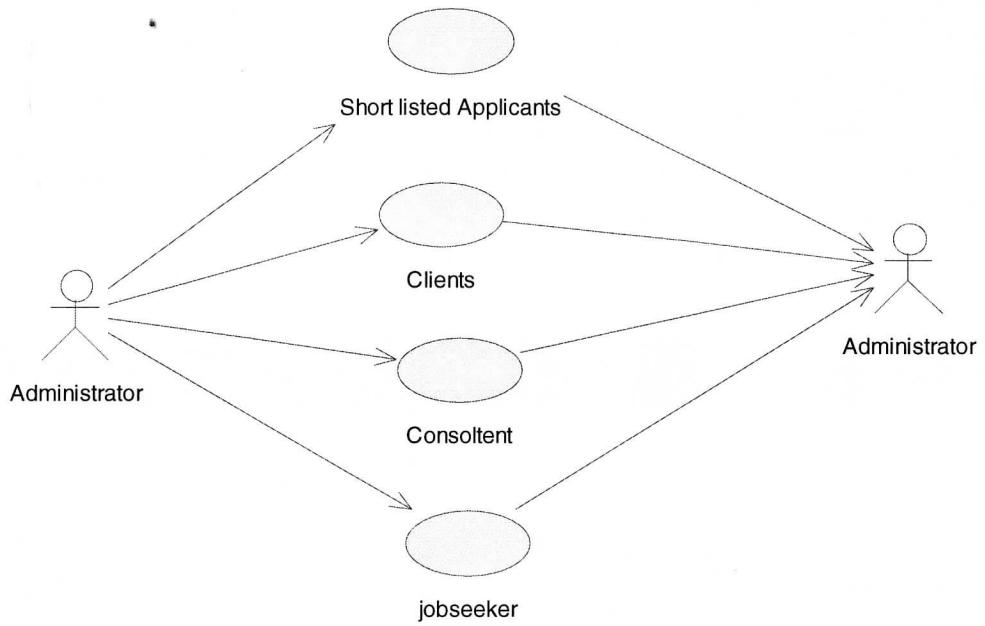


Figure 1: Use Case Diagram

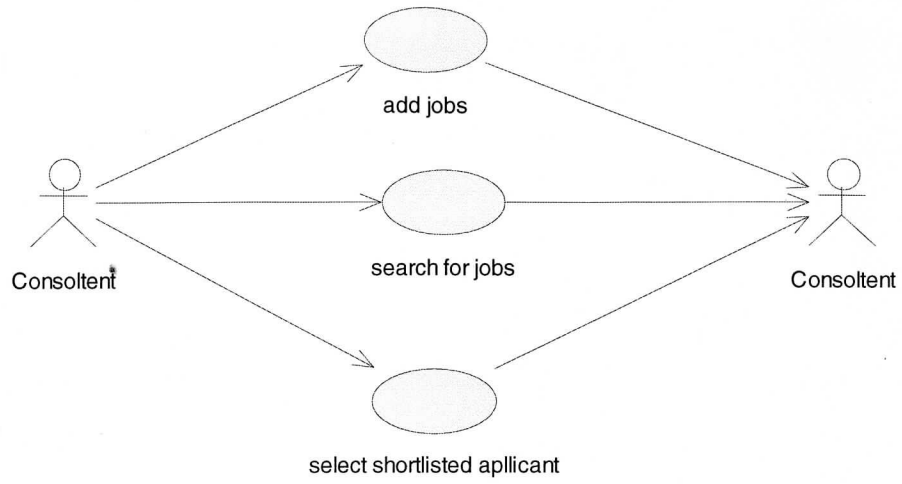


Figure 1: Use Case Diagram

Sequence Diagrams

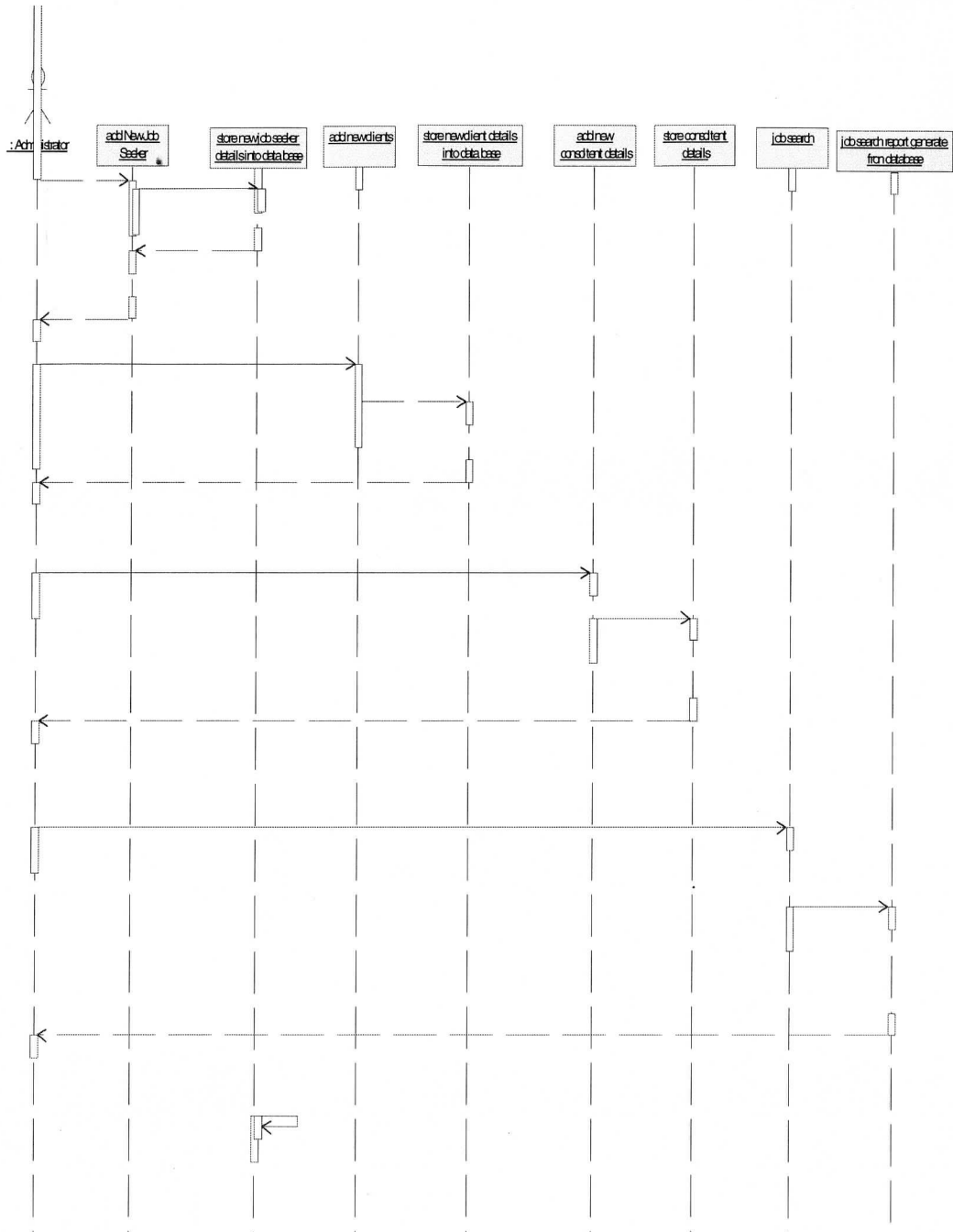


Figure 2: Sequence diagram of Administrator

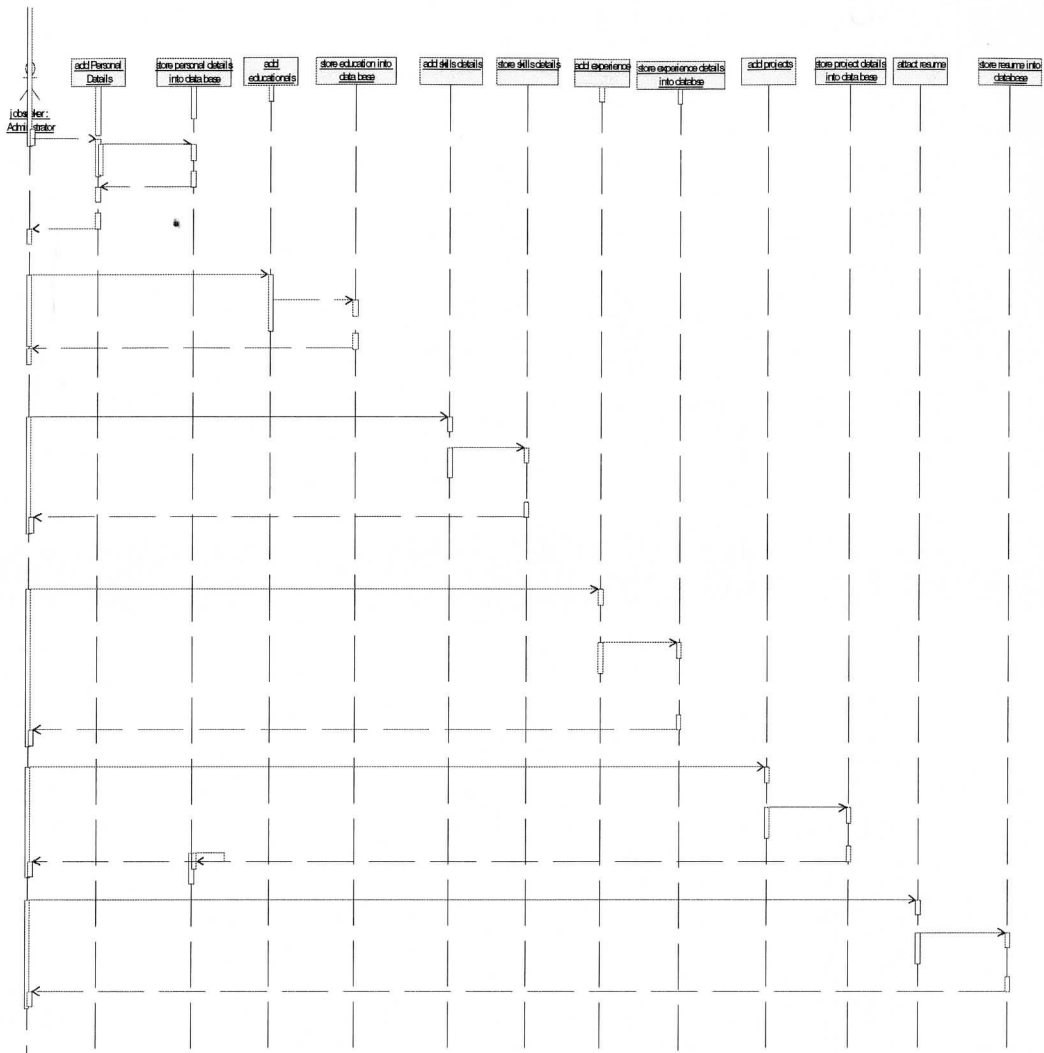
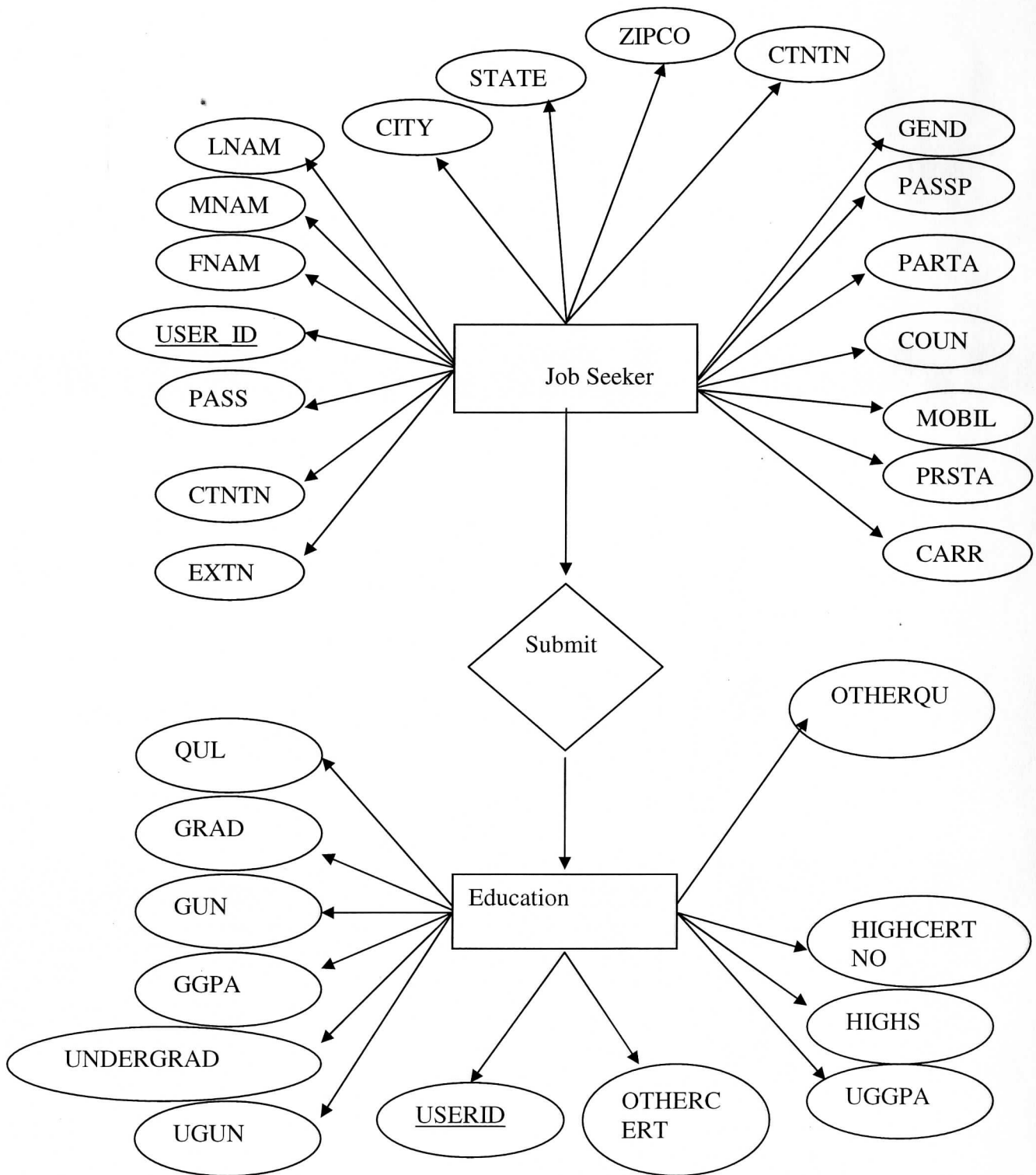


Figure 3: Sequence diagram of Job Seeker

E-R Diagrams



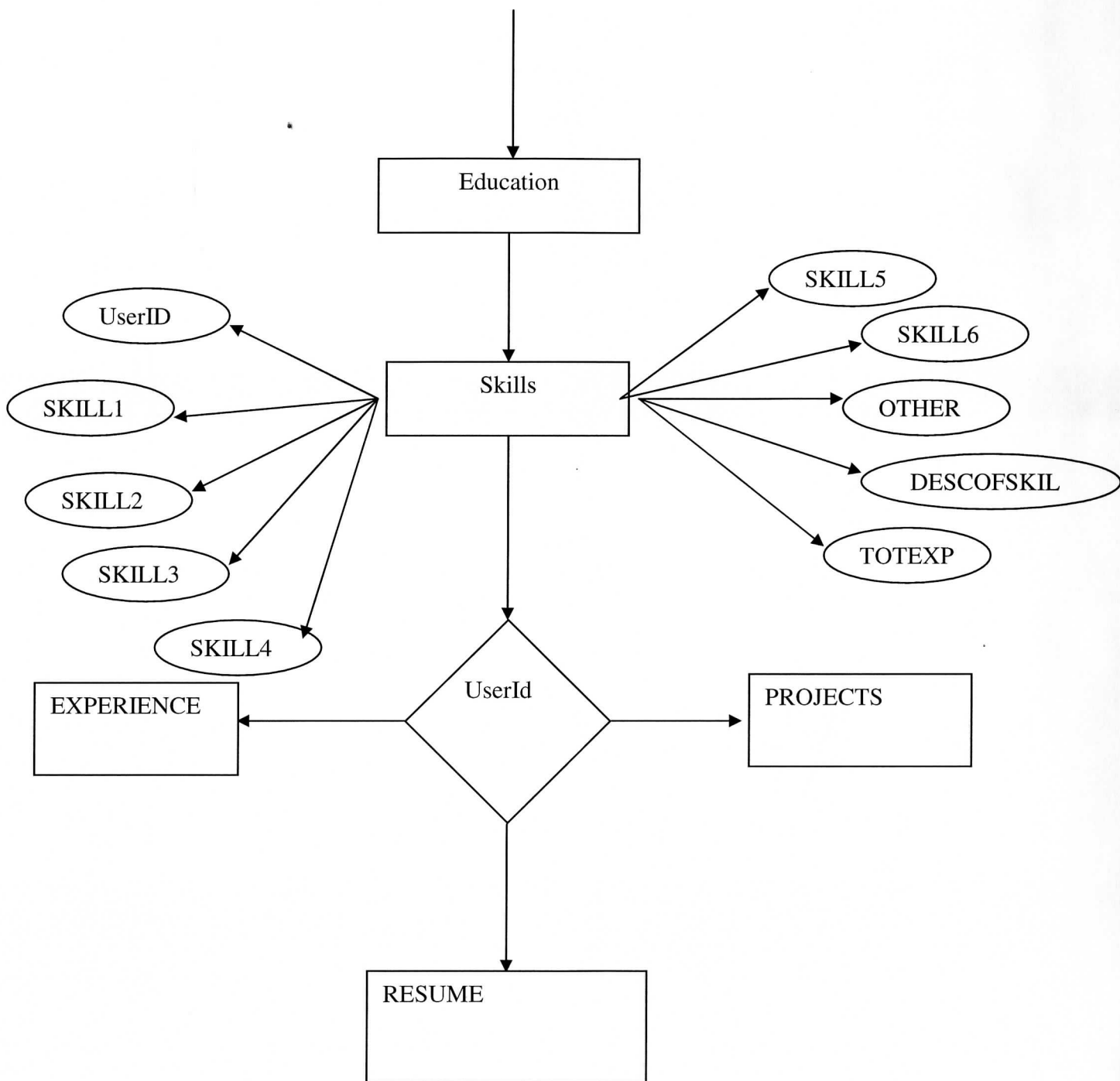


Figure 4: E-R diagram of a Job Seeker

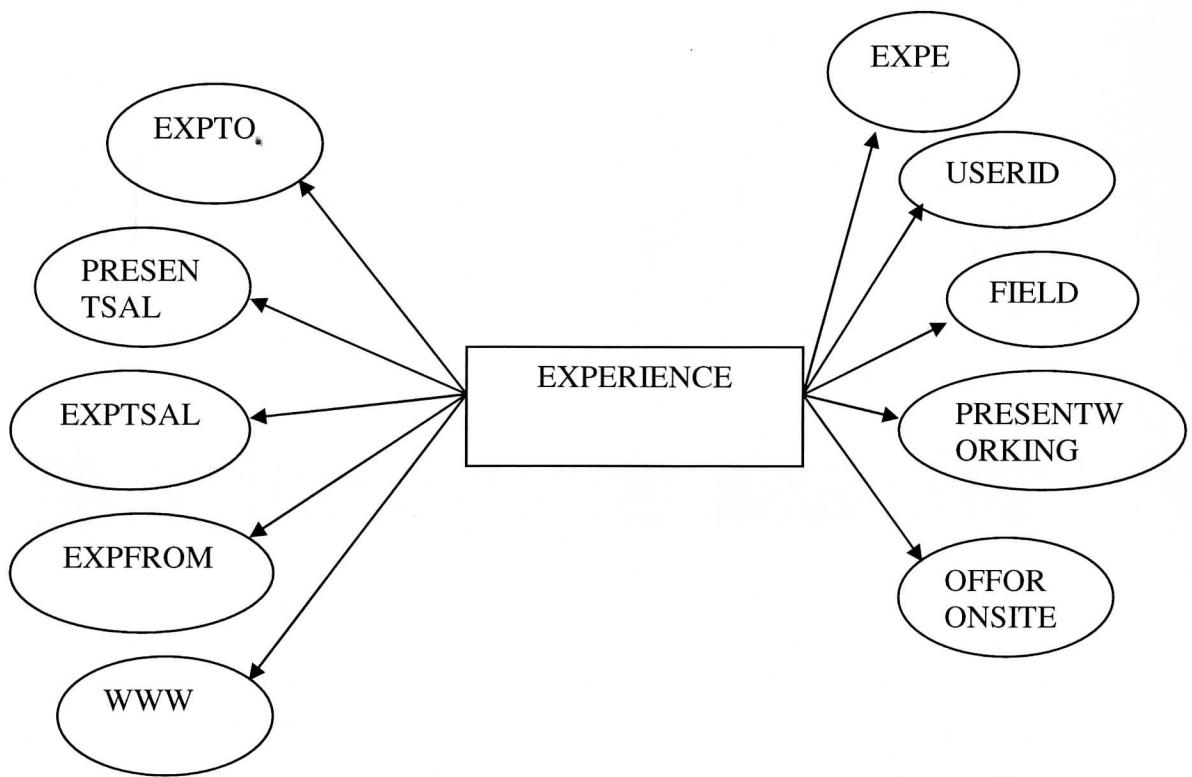


Figure 5 : Attributes of Experience

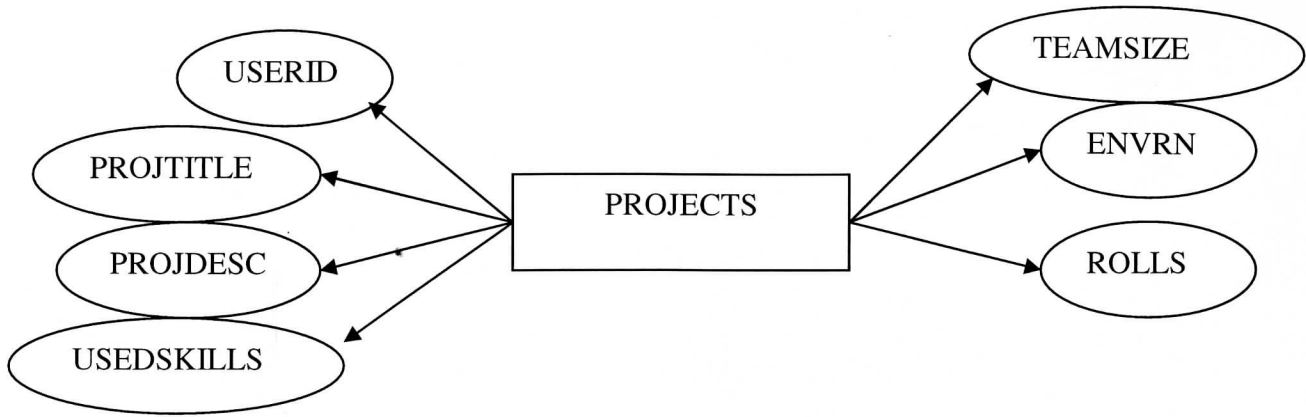


Figure 6: Attributes of Project

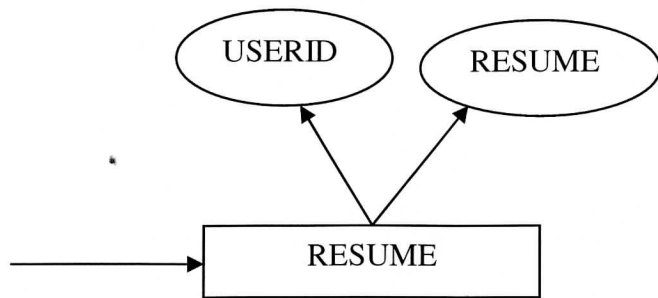
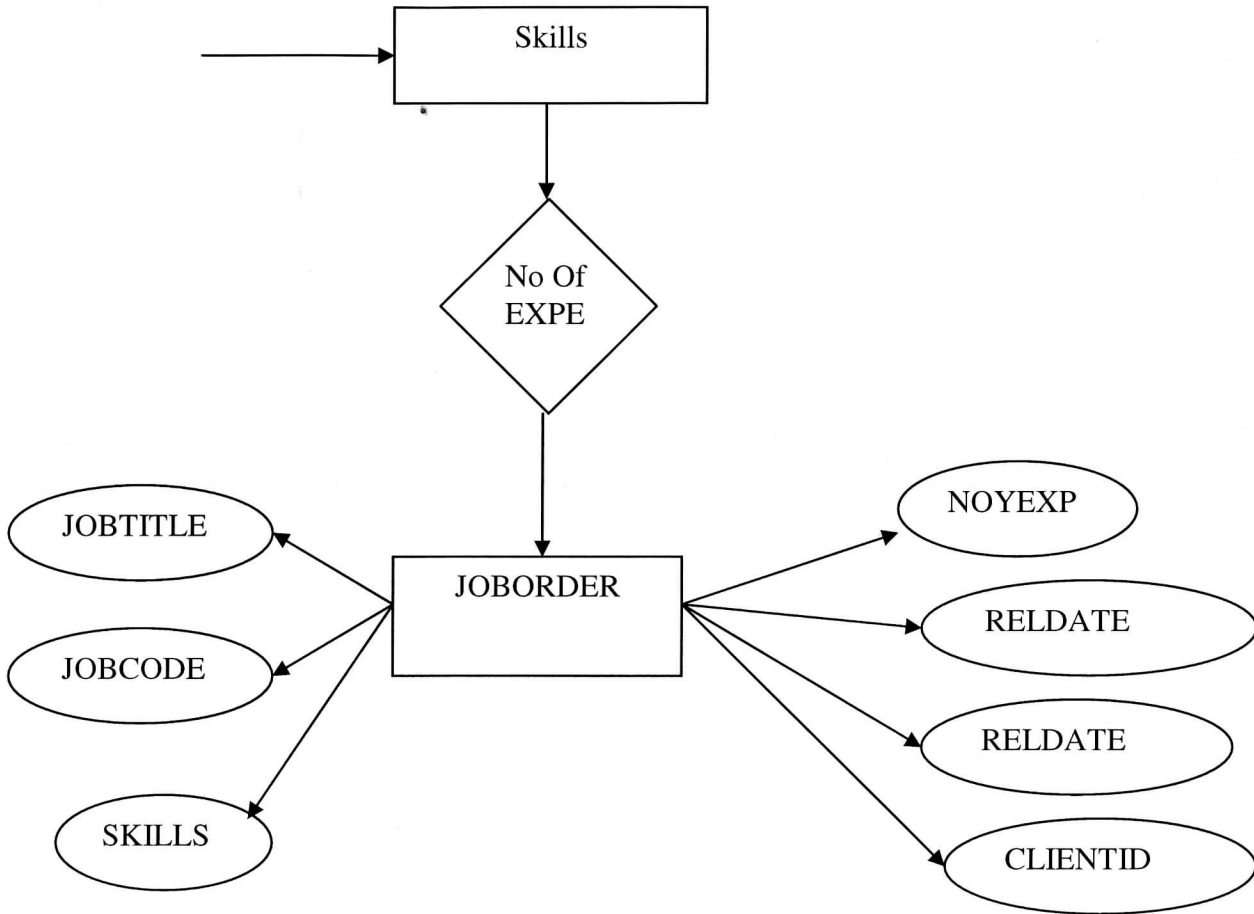


Figure 7: Attributes of Resume



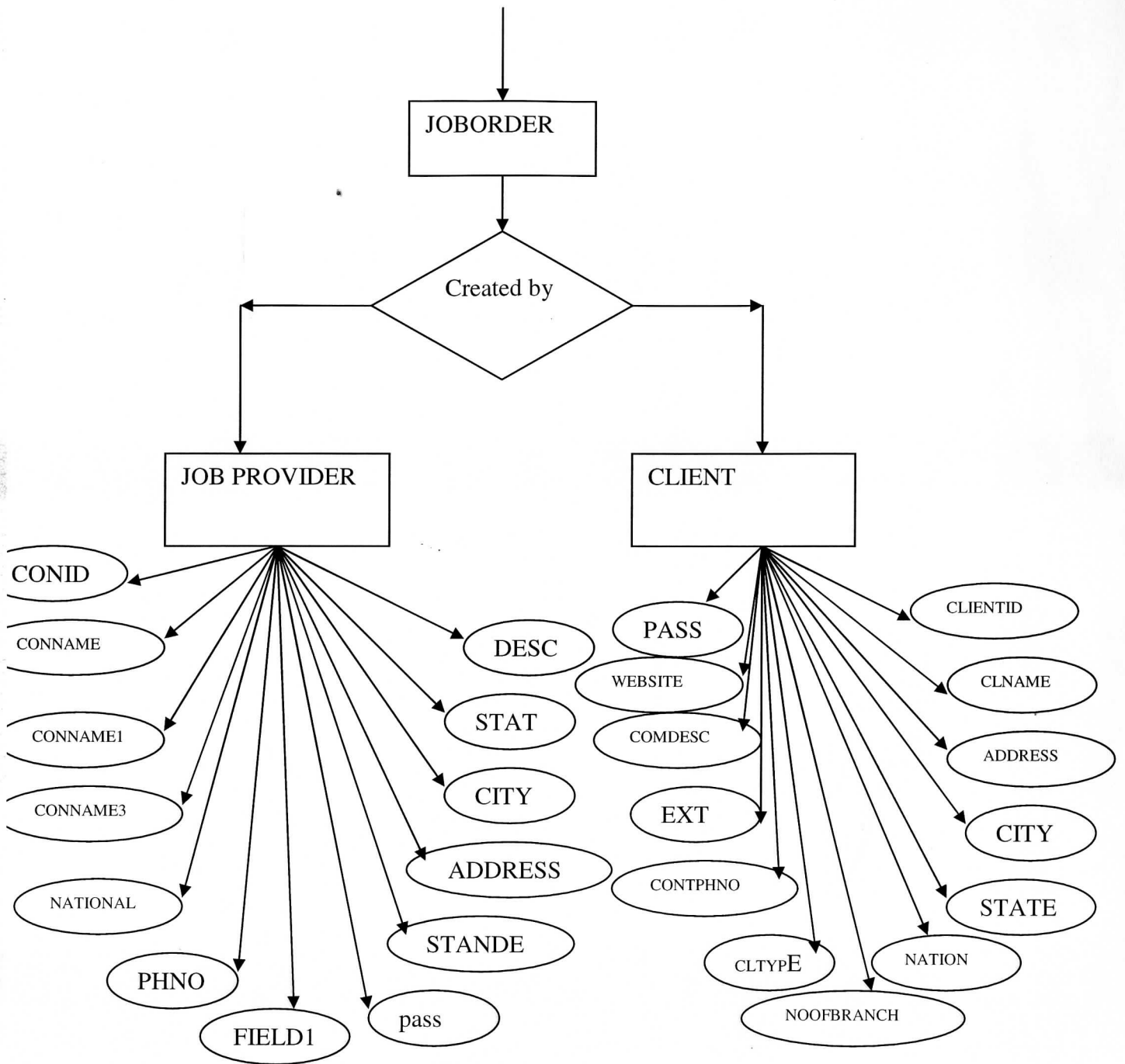
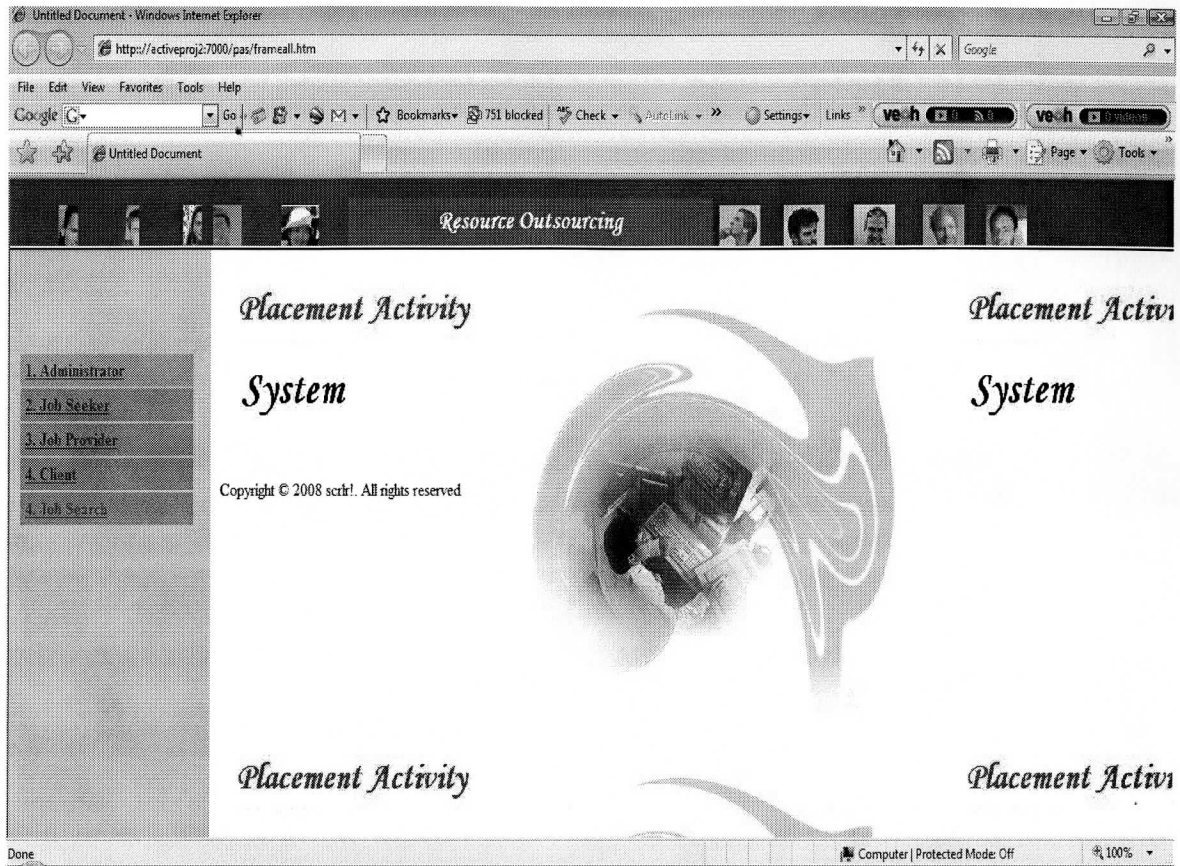
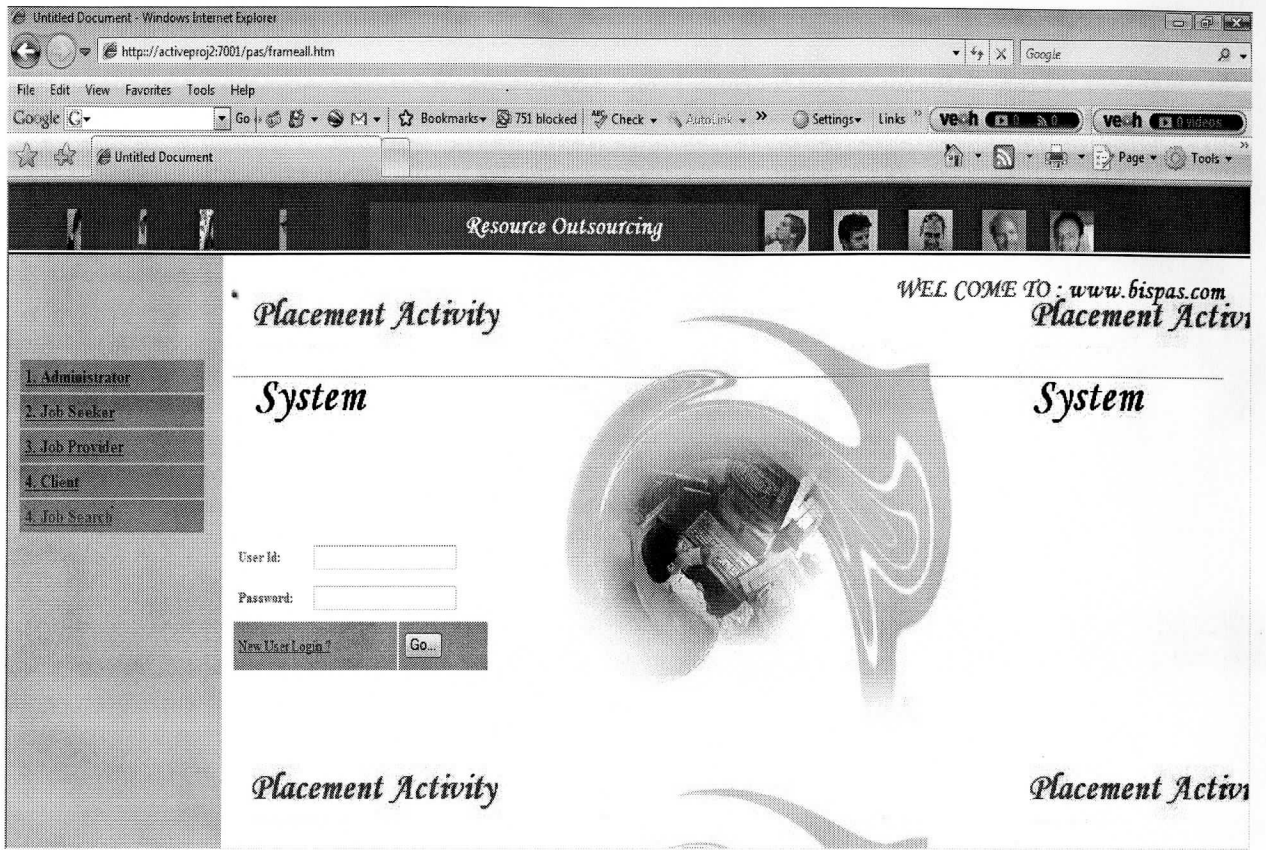


Figure 8 : E-R Diagram of Job Order

Screenshots



Screenshot1. Home Page



Screenshot2. Job Seeker Login page

Untitled Document - Windows Internet Explorer

http://activeproj27001/pas/frameall.htm

File Edit View Favorites Tools Help

Google G- Go Bookmarks 751 blocked Check AutoLink Settings Links ve-h ve-h

Untitled Document

Resource Outsourcing

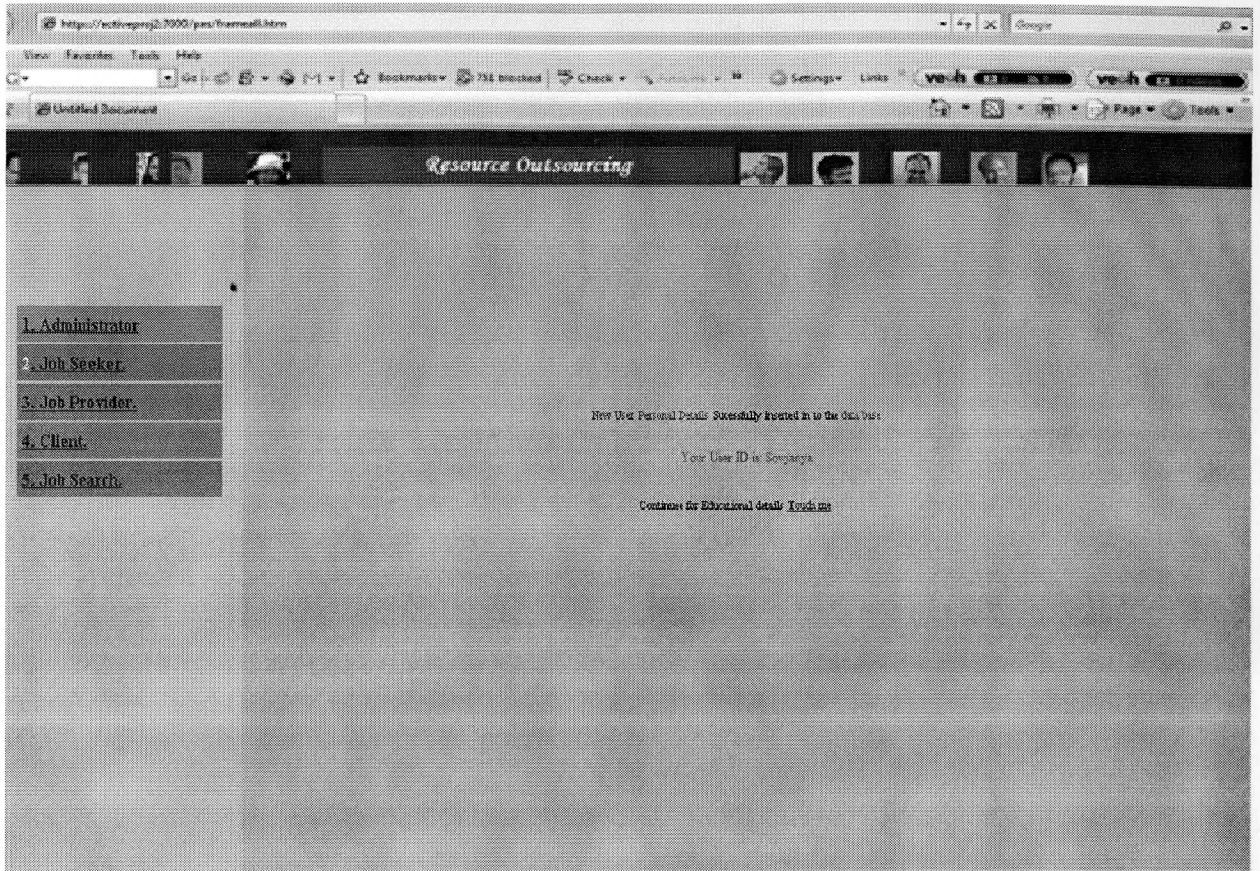
Enter the new user personal details

1. Administrator
2. Job Seeker
3. Job Provider
4. Client
5. Job Search

User Name: Sojanya
 Password: squntap@6
 First Name: Sojanya
 Middle Name:
 Last Name: guntapalli
 Gender: Female
 Present Address: 1709 Deer creek Dr
 Permanent Address: 1709 Deer creek Dr
 City: plainsboro
 State: Newjersey
 Country: United states
 Phone: 08536
 Contact Number: 8082283301
 Contact Number: 8082283301
 Mobile No: 8082283301
 Zip/Post:
 Career Specialty: SQL/PLSQL DEVELOPER
 Microsoft Certification:

Save Cancel

Screenshot3. Jobseeker's Registration Form



Screenshot4. Jobseeker's Success Message page

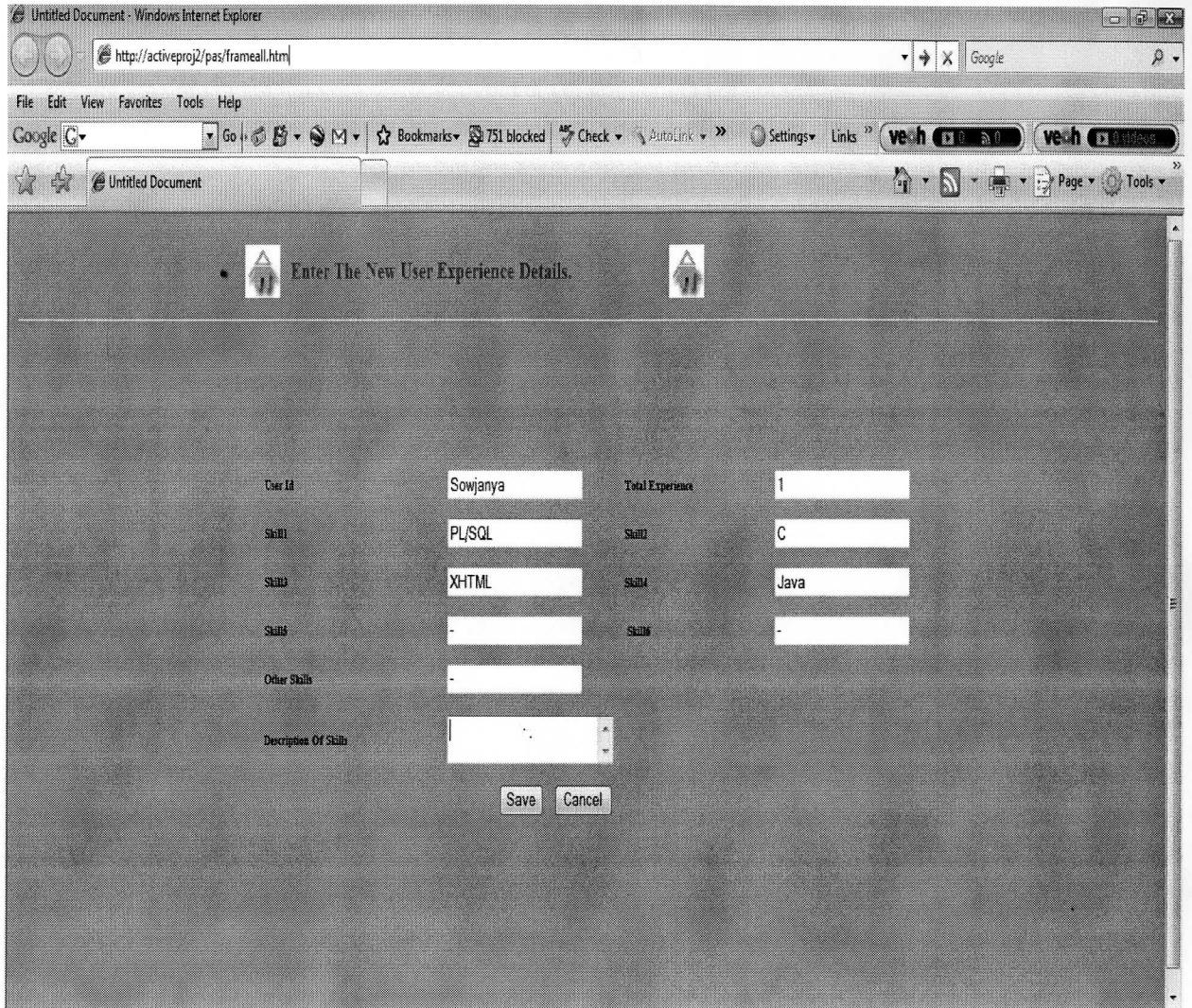
C

Untitled Document - Windows Internet Explorer
http://activeproj27001/pas/frameall.htm
File Edit View Favorites Tools Help
Google G
Go Bookmarks 751 blocked Check AutoLink Settings Links ve-h ve-h
Untitled Document
Enter the new user Education details.

High Qualification	MS	User Id	Sowjanya		
Graduation	MS	University	DSU	GPA	3.8
Under Graduation	BS	University	JNTU	GPA	3.55
High School	SSC	Certificate NO	U123456		
Other Qualifications	NA				
Other Certifications	NA				

Save Cancel

Screenshot5. Jobseeker's Registration Form2



Screenshot6. Jobseeker's Registration Form3

Untitled Document - Windows Internet Explorer

http://activeproj2:7001/pas/frameall.htm

File Edit View Favorites Tools Help

Google

Go Bookmarks 751 blocked Check AutoLink Settings Links ve-h ve-h

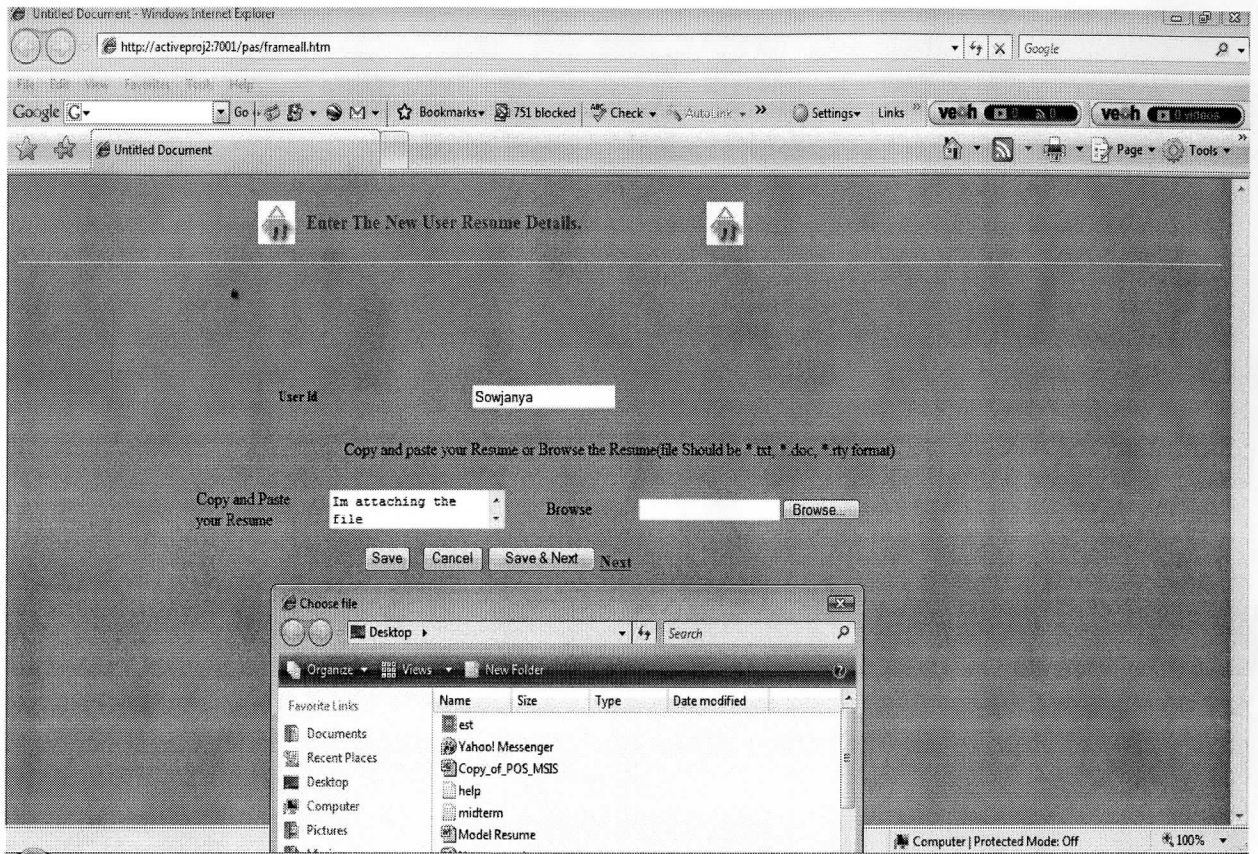
Untitled Document

Enter The New User Experience Details.

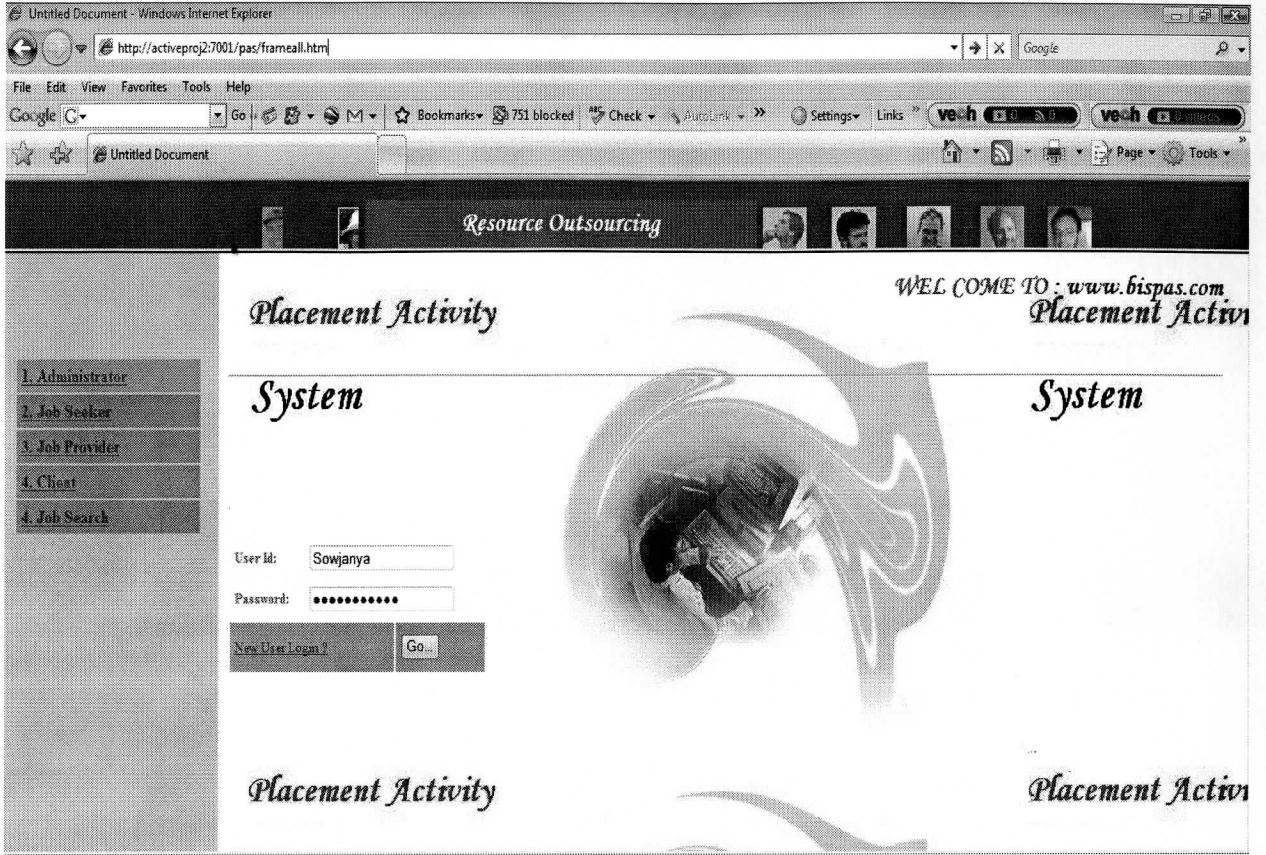
User ID	Sowjanya	Total Experience	1
Experience in the Field	PL/SQL Developer	Present Working Company	Business Intelligence sc
Off-site / On-site	off-site	If On-site Submit URL(EX. www.abc.com)	
From Exp(dd-mm-yyyy)	15-06-08	To Exp(dd-mm-yyyy)	15-11-08
Present Salary	-	Expected Salary	-

Save Cancel Save & Next Next

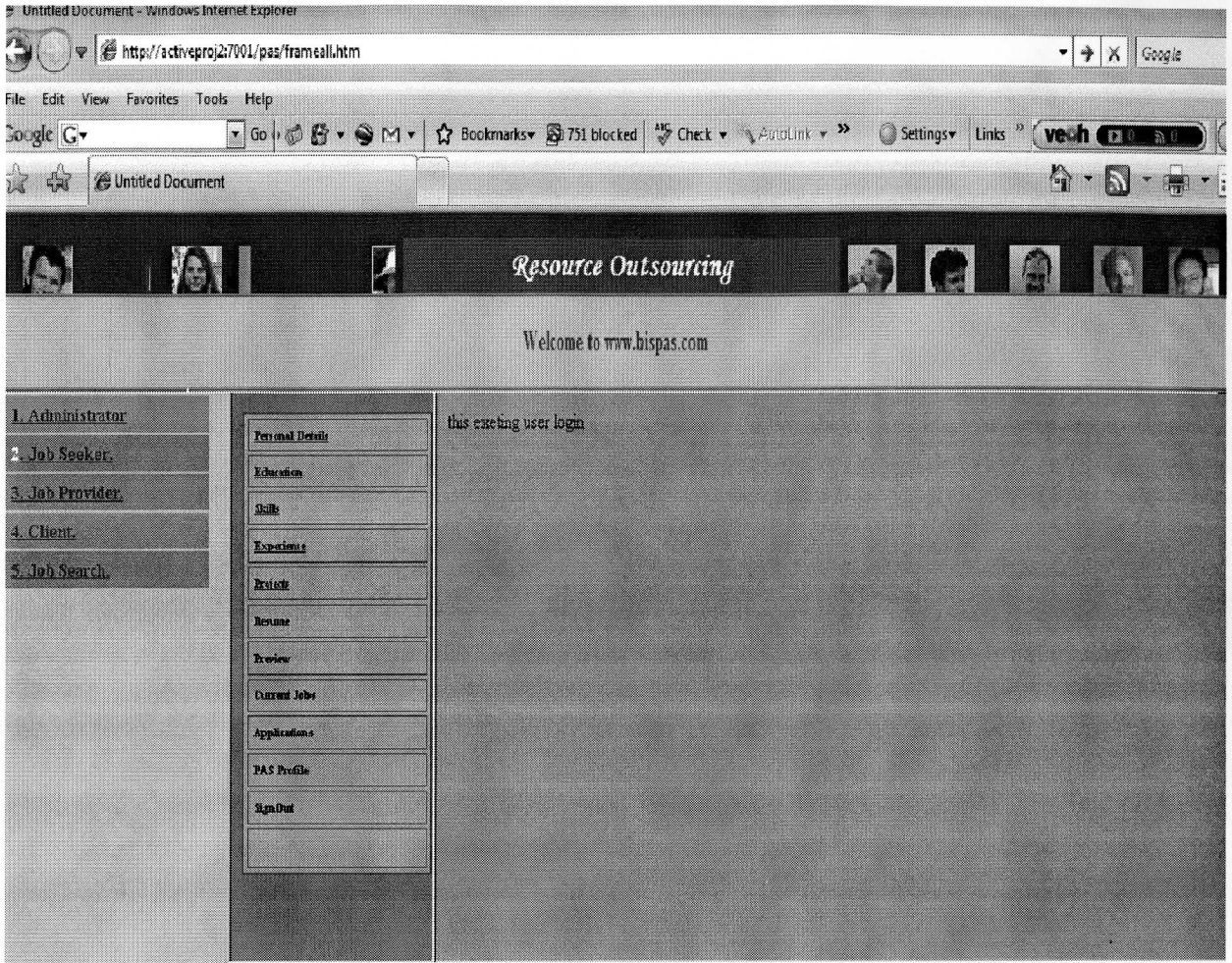
Screenshot7. Jobseeker's Registration Form4



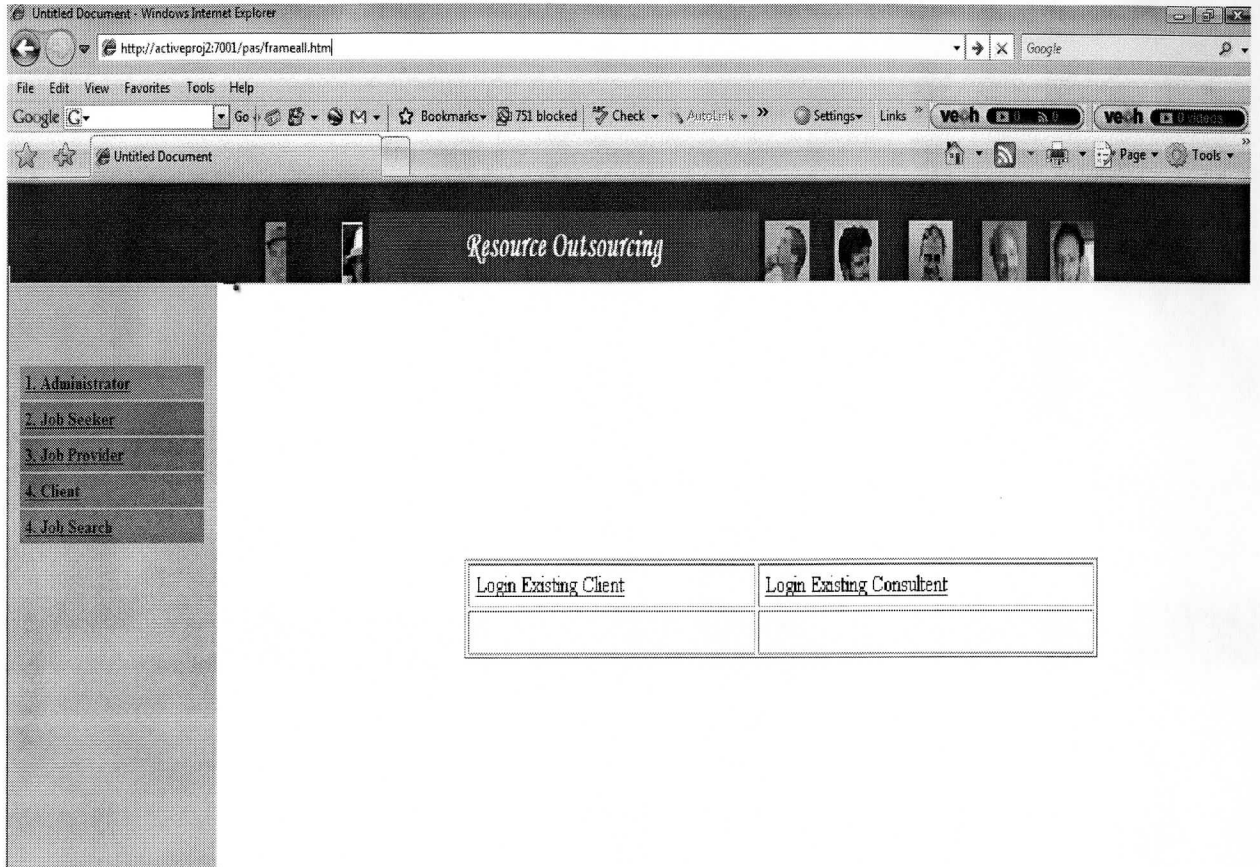
Screenshot8. Jobseeker's final Registration Form



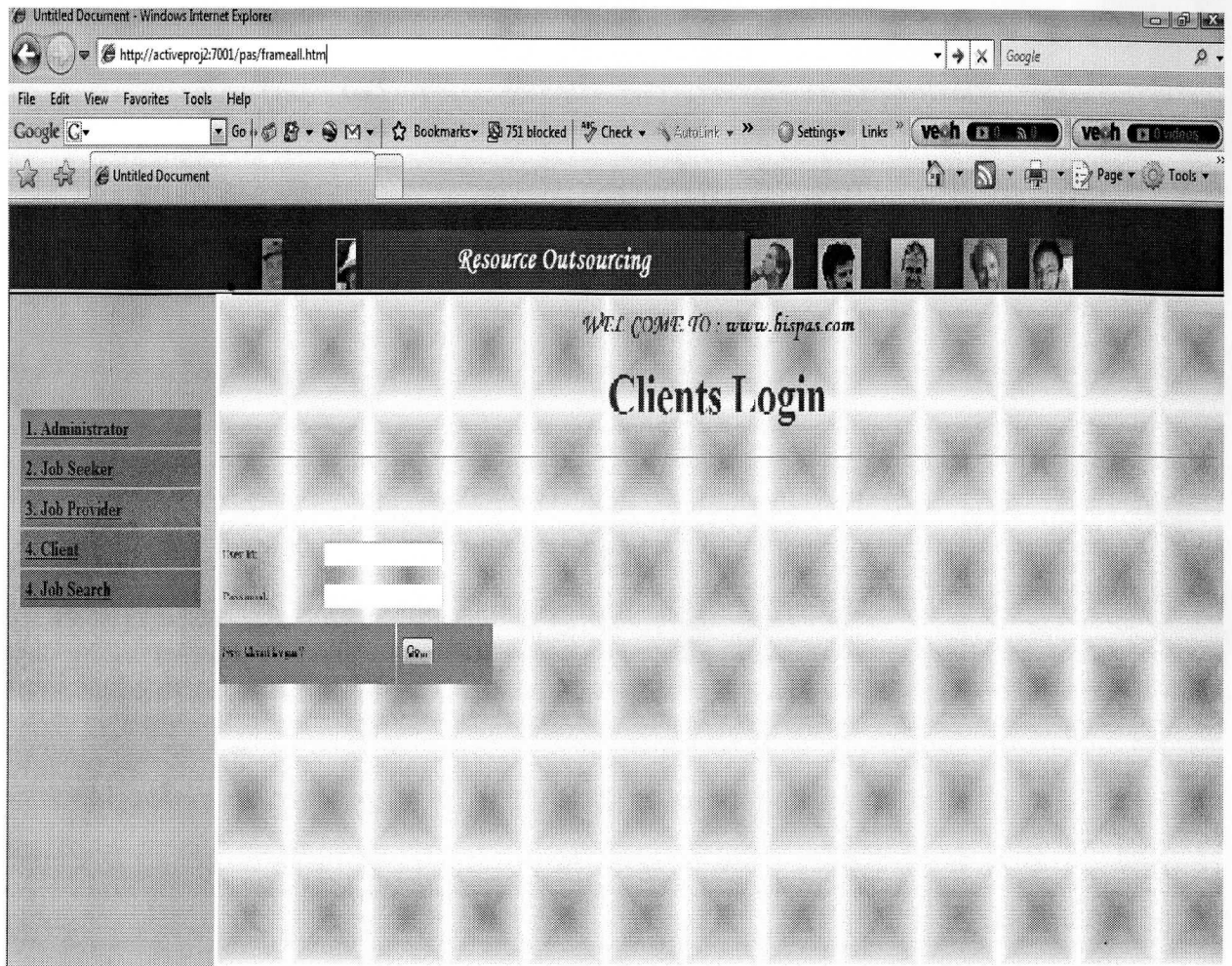
Screenshot9. Jobseekers Login Form



Screenshot10. Administrator Page



Screenshot11. Job Provider's Homepage



Screenshot12. Client Login Page

Untitled Document - Windows Internet Explorer
 http://activeproj27001/pas/frameall.htm

File Edit View Favorites Tools Help
 Google Go Bookmarks 751 blocked Check Autolink Settings Links veoh veoh

Untitled Document Page Tools

Resource Outsourcing

Enter the new Clients details.

1. Administrator

2. Job Seeker

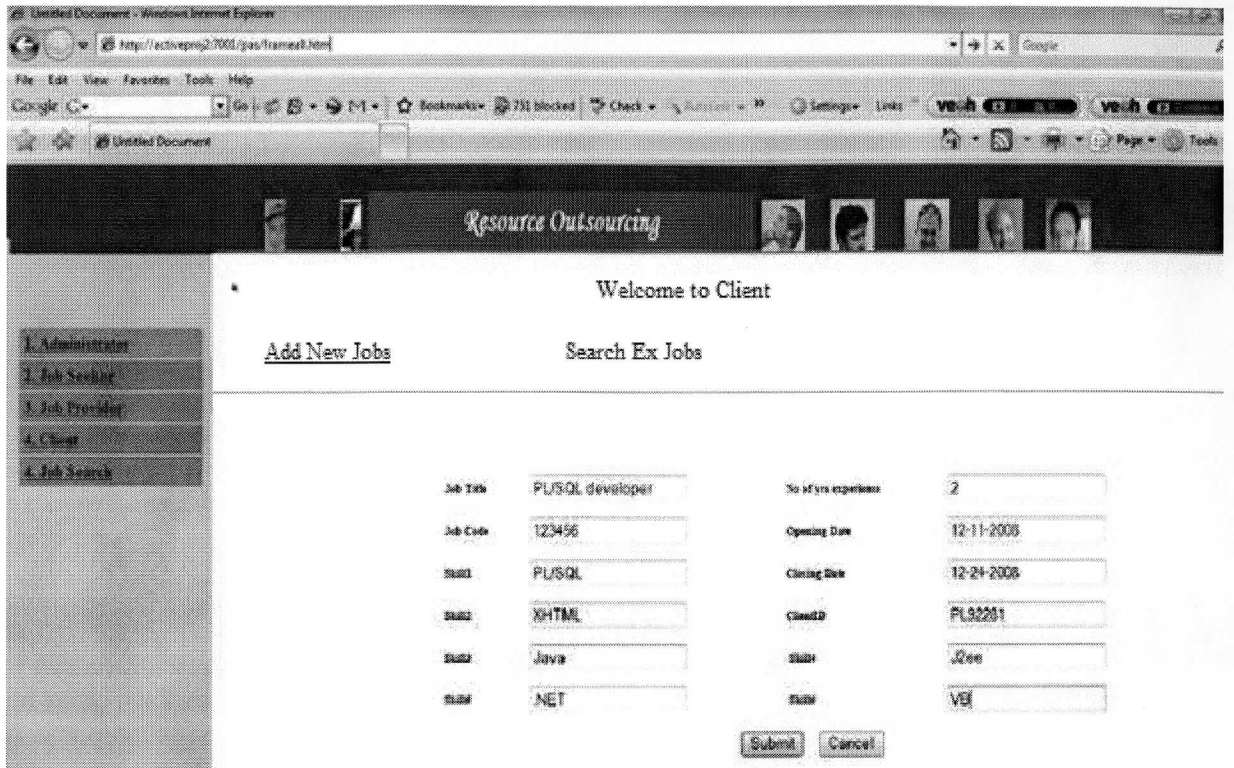
3. Job Provider

4. Client

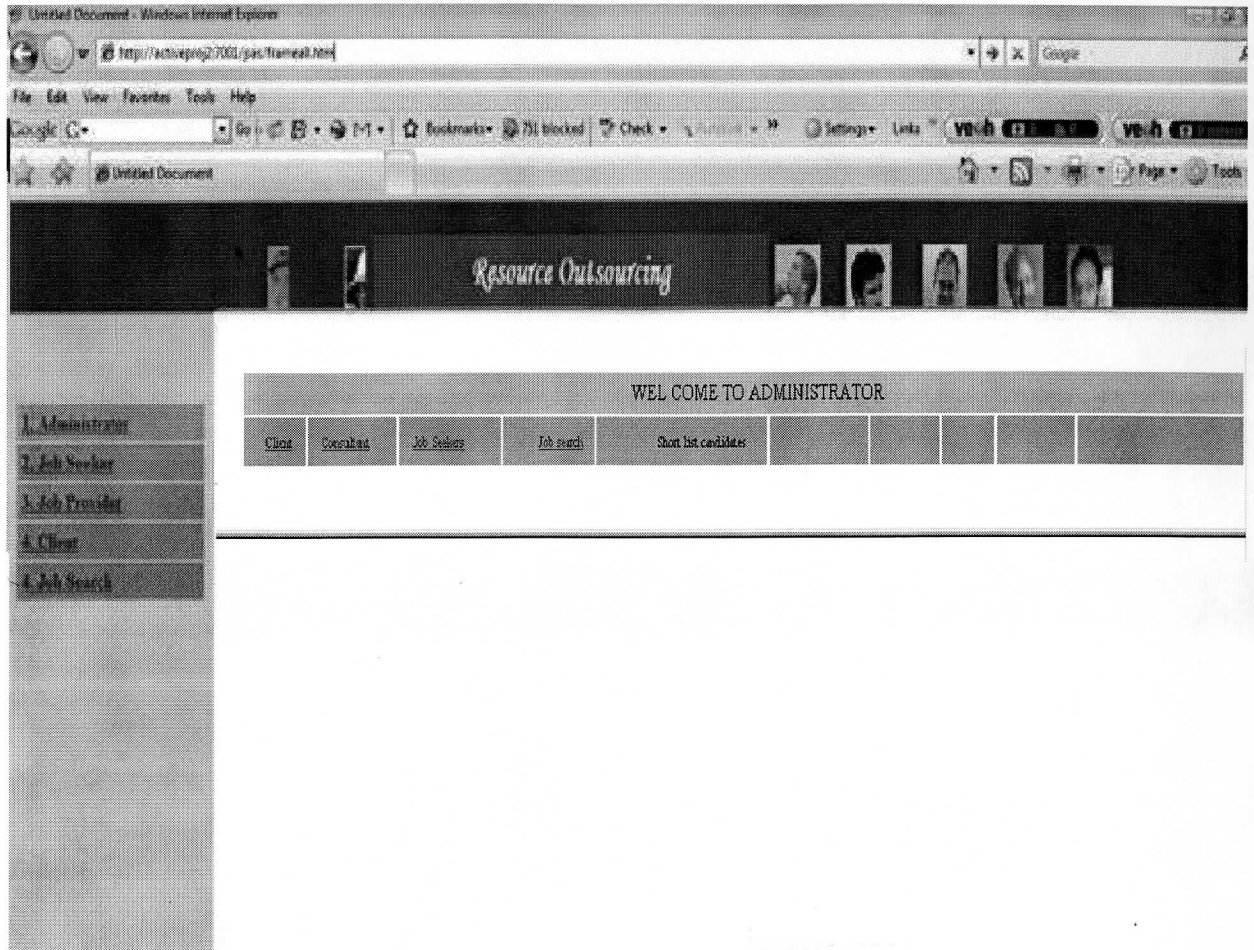
4. Job Search

Client Id	<input type="text" value="PL92281"/>	Contact Ph No	<input type="text" value="7326564060"/>
Client Name	<input type="text" value="Business Intelligence St"/>	Location	<input type="text"/>
Address	<input type="text" value="100 Overlook center"/>	Company Desc	<input type="text" value="Leading solution provider"/> <small>704 804 8044 308 Characters</small>
City	<input type="text" value="princeton"/>	Contact Person	<input type="text" value="Rajesh Dugga'sapudi"/>
State	<input type="text" value="New Jersey"/>	Website	<input type="text" value="www.bintelgroup.com"/>
Country	<input type="text" value="United States"/>	Password	<input type="text" value="bin1el"/>
No. Of Branches	<input type="text" value="4"/>		
Client Type	<input type="text" value="IT industry"/>		

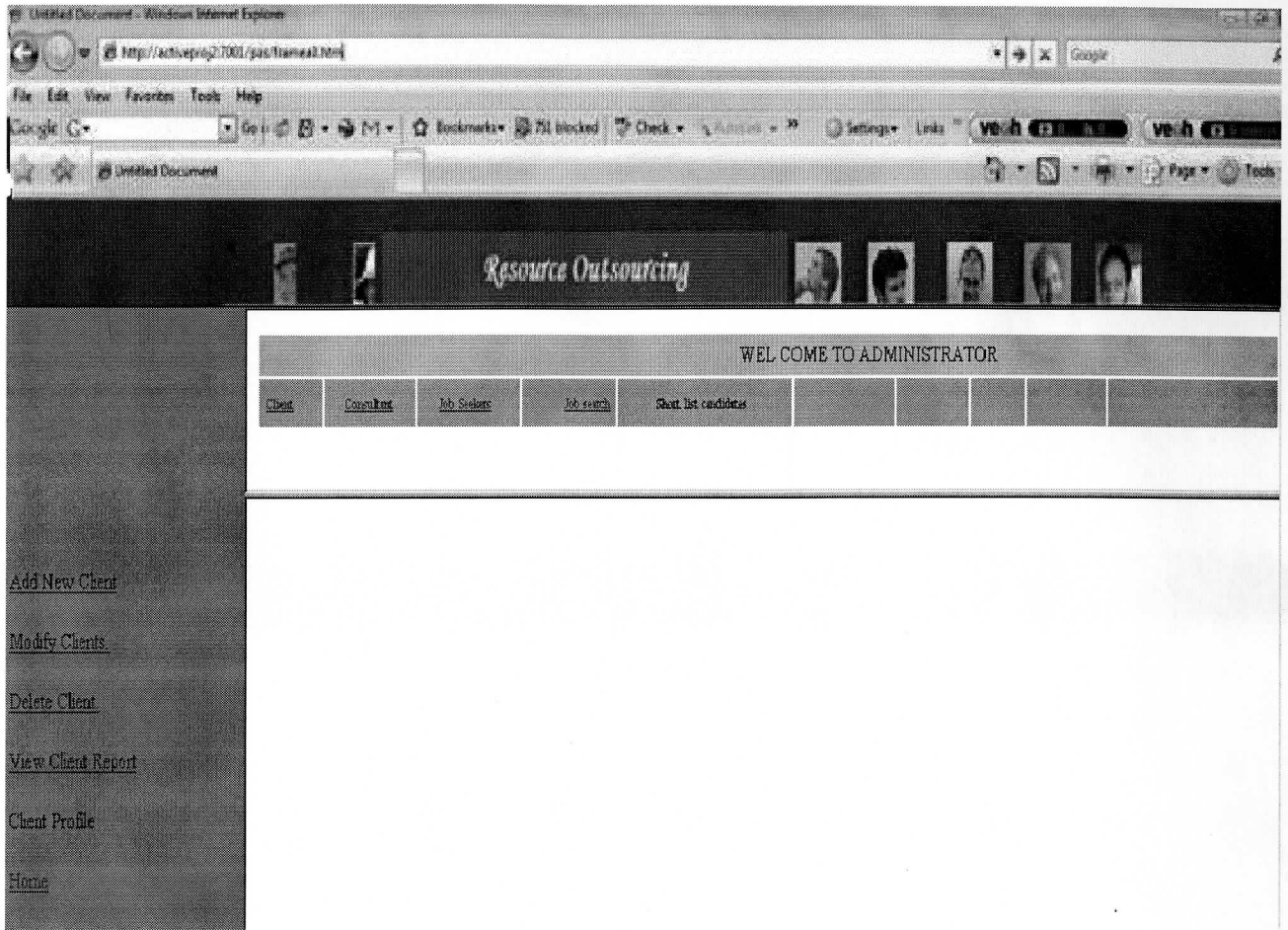
Screenshot13. Client Registration Form



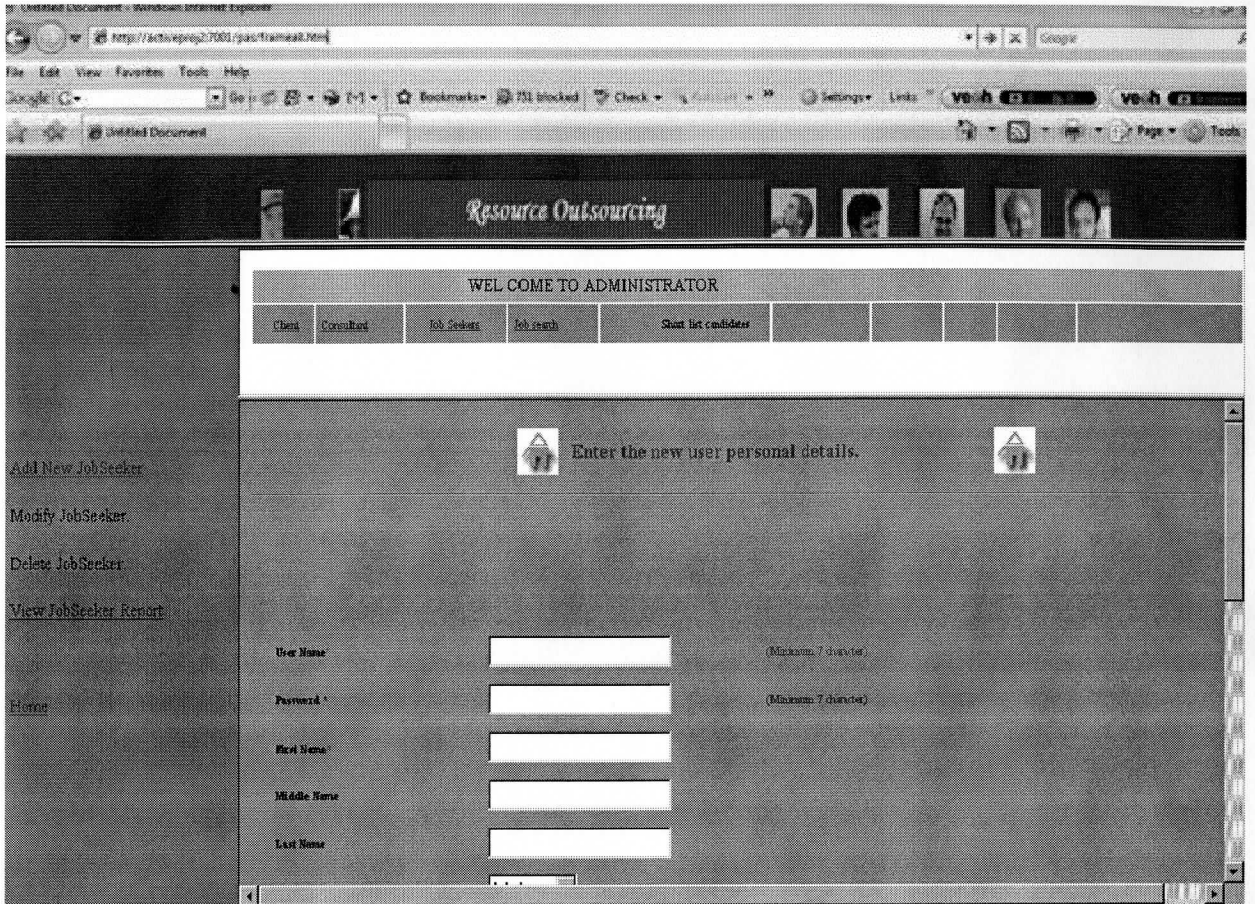
Screenshot14. Client's Job Adding Page



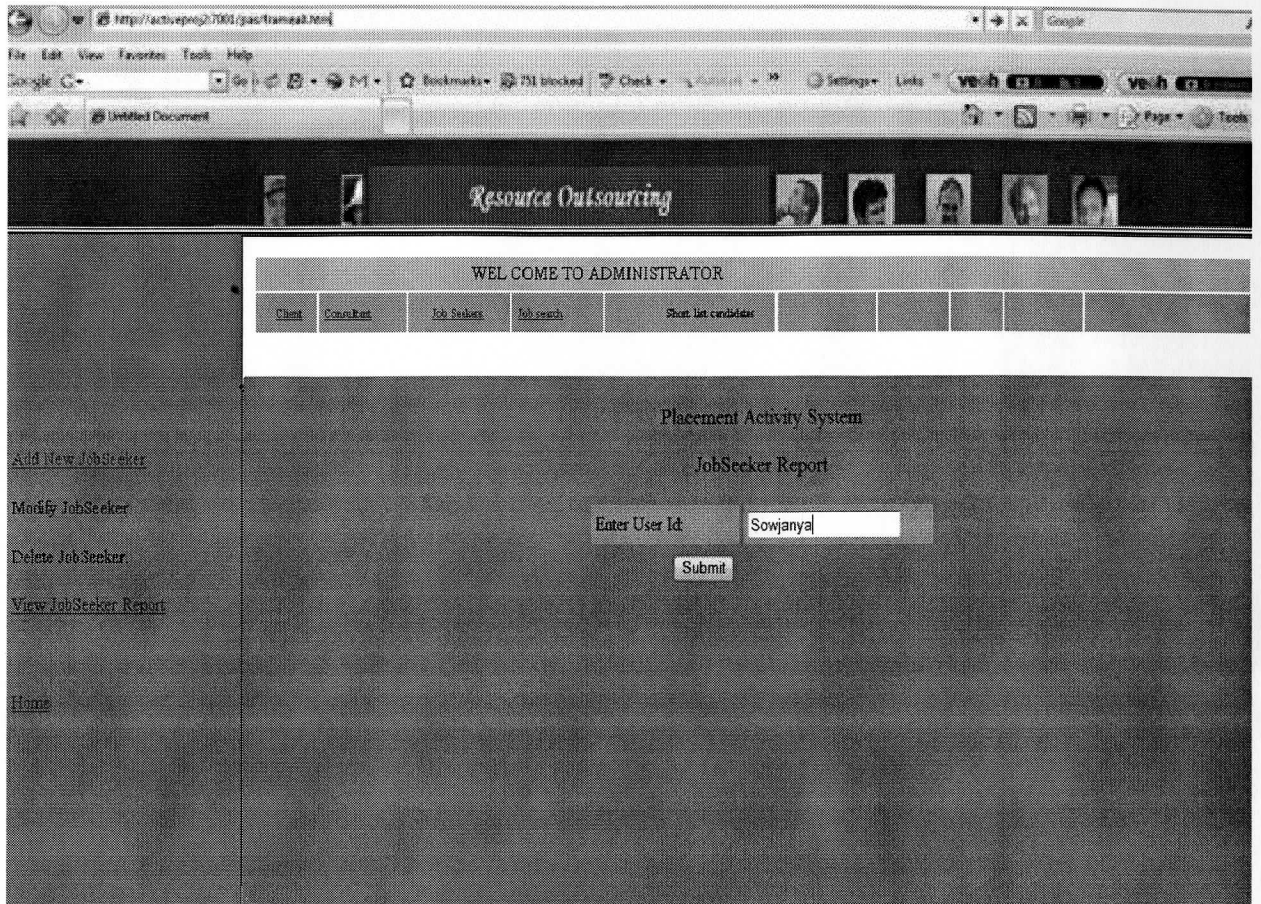
Screenshot15. Administrator Modules Page



Screenshot16. Administrator's Client Page



Screenshot17. Administrator's Jobseekers page



Screenshot18. Administrators Jobseeker search Page

http://activeproj27000/pas/frameat.htm

File Edit View Favorites Tools Help

Google

751 blocked

Settings Links

vb-h

vb-h

Untitled Document

Page Tools

Resource Outsourcing

WELCOME TO ADMINISTRATOR

Client Consultants Job Seekers Job Search Short list candidates

Placement Activity System

Job Seekers Reports

Job Seeker User Id: Sowjanya

[Add New Job Seeker](#)

[Modify Job Seeker](#)

[Delete Job Seeker](#)

[View Job Seeker Report](#)

[Home](#)

User Id	Sowjanya	City	Plainsboro
Password	sguntupalli	State	Newjersey
First Name	Sowjanya	Country	United States
Middle Name		Contact Off No	
Last Name	Guntupalli	Off-Ext.No	
Present Address	1708 deer creek dr	Cont-Personal No	8082263301
Permanent Address	1708 deer creek dr	Passport No	
Zipcode	08536	Gender	Female
Mobile No	8082263301		
Career Object	PL SQL DEVELOPER		

Screenshot19. Administrator's Job Seeker Report

Untitled Document - Windows Internet Explorer
 http://activeproj2:7001/pas/frameall.htm

File Edit View Favorites Tools Help
 Google G- Go 751 blocked Check AutoLink Settings Links ve-h ve-h

Untitled Document

Resource Outsourcing

WEL COME TO ADMINISTRATOR

Client Controlling Job Seekers Job Alerts Short list candidates

[Skill Based](#)
[Experience Based](#)
[Client Based](#)
[View Job Seeker Report](#)
[Home](#)

Client Id	Job Title	Jobcode	Requied Skills	Experience	Closing Date
PL92281	PL/SQL developer	123456	PL/SQL,XHTML,Java,J2ee,NET,VB	2 Years	12-24-2008

Screenshot20. Administrator's Jobseeker Information page

Untitled Document - Windows Internet Explorer

http://actv.epsg27001/jas/frame2.htm

File Edit View Favorites Tools Help

Google G- Go | [Icons] | Bookmarks | [Icons] | Check | Settings | Links | [Icons]

Untitled Document

Resource Outsourcing

WEL COME TO ADMINISTRATOR

[Client](#) | [Consultant](#) | [Job Seekers](#) | [Job Search](#) | [Recruit Candidates](#)

Placement Activity System

Client Reports

[Add New Client](#)

[Modify Clients](#)

[Delete Client](#)

[View Client Report](#)

[Client Profile](#)

[Home](#)

Client Id : PL92281

Client Id	PL92281	Client Pa NO	7326664060
Client Name	Business Intelligence sol	Ext No	
Address	100 Overlook Center	Client Desc	good
City	Princeton	Contact Person	Rajesh Doppalapudi
State	New Jersey	Website	www.bintelgroup.com
Nation	United States	Password	bintel
No Of Branches	4		
Client Type	IT Industry		

Screenshot21. Administrator's Client information Page

APPENDIX B: SYSTEM TECHNICAL DOCUMENTATION

Overview of Java Technology

History of Java:

Java language was developed by James Gosling and his team at sun micro systems and released formally in 1995. Its former name is oak. Java Development Kit 1.0 was released in 1996. To popularize java and is freely available on Internet.

How Java works?

Java is loosely based on C++ syntax, and is meant to be Object-Oriented Structure; java is midway between an interpreted and a compiled language. Java programs are compiled by the java compiler into Byte Codes which are secure and portable across different platforms. These byte codes are essentially instructions encapsulated in single type, to what is known as java virtual machine (JVM) which resides in standard browser.

JVM verifies these byte codes when downloaded by the browser for integrity. JVM is available for almost all Operating systems. JVM converts these byte codes into machine specific instructions at runtime.

Features of Java:

Java is object-oriented language and supports encapsulation, inheritance, polymorphism and dynamic binding, but does not support multiple inheritances. Everything in Java is an object except some primitive data types.

Java is a portable architecture which means programs once compiled can be executed on any machine that is enabled.

Java is distributed in its approach and used for internet programming.

Java is robust, secured, high performing and dynamic in nature.

Java supports multithreading. There for different parts of the program can be executed at the same time

Java and Internet:

Java is strongly associated with internet and known as internet programming language. Internet users can use java to create applet programs and run them locally using java enabled browser search as hot java. Applets can be downloaded from remote machine via internet and run it on local machine.

Java and World Wide Web

World Wide Web is an open ended information retrieval system designed to be used in the distributed environment. This system contains web pages that provide both information and controls. We can navigate to a new web page in any direction. This is made possible with HTML java was meant to be used in distributed environment such as internet. So java could be easily incorporated into the web system and is capable of supporting animation graphics, games and other special effect. The web has become more dynamic and interactive with support of java. We can run a java program on remote machine over internet with the support of web.

Java Environment

Java environment includes a large number of tools which are part of the system known as java development kit (JDK) and hundreds of classes, methods, and interfaces grouped into packages forms part of java standard library (JSL).

Java Architecture

Java architecture provides a portable, robust, high performing environment for development. Java provides portability by compiling the byte codes for the java virtual machine which are then interpreted on each platform by the runtime environment. Java also provides stringent compile and runtime checking and automatic memory management in order to ensure solid code.

Java Virtual Machine

When we compile the code, java compiler creates machine code (byte code) for a hypothetical machine called java virtual machine (JVM). JVM will execute the byte code and overcomes the issue of portability. The code is written and compile for one machine and interpreted all other machines. This machine is called java virtual machine.

Paradigm of Java

Dynamic down loading applets (small application programs);

Elimination of flatware phenomenon that is providing those features of a product that user needs at a time. The remaining features of a product can remain in the server.

Changing economic model of the software

Up-to-date software availability

Supports network entire computing

Supports CORBA & DCOM

About HTML

HTML (hyper text markup language) is a language used to create hyper text documents that have hyper links embedded in them. It consists of tags embedded in the text of a document with HTML. We can build web pages or web documents. It is basically a formatting language and not a programming language. The browser reading the document interprets mark up tags to help format the document for subsequent display to a reader. HTML is a language for describing structured documents. HTML is a platform independent. WWW (world wide web) pages are written using HTML. HTML tags control in part the representation of the WWW page when view with web browser. The browser interprets HTML tags in the web document and displays it. Different browsers show data differently. Examples of browsers used to be web pages include:

Netscape and Internet Explorer

Java Script

Java script is a general purpose, prototype based, object oriented scripting language developed jointly by sun and Netscape and is meant for the WWW. It is designed to be embedded in diverse applications and systems, without consuming much memory. Java script borrows most of its syntax from java but also inherits from awk and Perl, with some indirect influence from self in its object prototype system.

Java scripts dynamically typed that is programs do not declare variable types, and the type of variable is unrestricted and can change at runtime. Source can be generated at run time and evaluated against an arbitrary scope. Typical implementations compile by translating source into a specified byte code format, to check syntax and source consistency. Note that the

availability to generate and interpret programs at runtime implies the presence of a compiler at runtime.

Java script is a high level scripting language that does not depend on or expose particular machine representations or operating system services. It provides automatic storage management, typically using a garbage collector.

Features of Java Script

- Java script is embedded into HTML documents and is executed with in them.
- Java script is browser dependent
- Java script is an interpreted language that can be interpreted by the browser at run time
- Java script is loosely typed language
- Java script is an object based language.
- Java script is an Event-Driven language and supports event handlers to specify the functionality of a button.

Advantages of Java Script

Java script can be used for client side application

Java script provides means to contain multiframe windows for presentation of the web.

Java script provides basic data validation before it is sent to the server. Ex: login and password checking or whether the values entered are correct or whether all fields in a form are filled and reduced network traffic

It creates interactive forms and client side lookup tables.

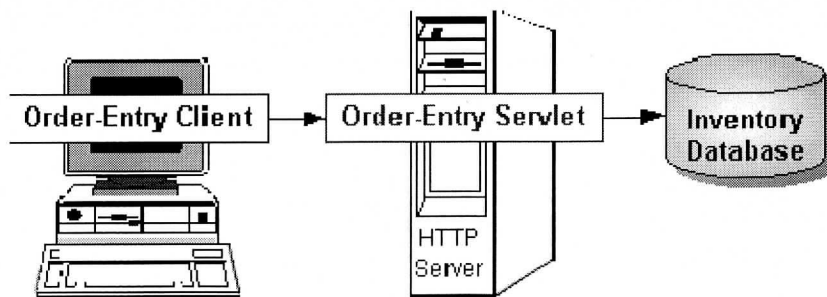
Introduction to Servlets

Servlets provide a Java^(TM)-based solution used to address the problems currently associated with doing server-side programming, including inextensible scripting solutions, platform-specific APIs, and incomplete interfaces.

Servlets are objects that conform to a specific interface that can be plugged into a Java-based server. Servlets are to the server-side what applets are to the client-side -- object byte codes that can be dynamically loaded off the net. They differ from applets in that they are faceless objects (without graphics or a GUI component). They serve as platform-independent, dynamically-loadable, pluggable helper byte code objects on the server side that can be used to dynamically extend server-side functionality.

What is a Servlet?

Servlets are modules that extend request/response-oriented servers, such as Java-enabled web servers. For example, a servlet might be responsible for taking data in an HTML order-entry form and applying the business logic used to update a company's order database.



Servlets are to servers what applets are to browsers. Unlike applets, however, Servlets have no graphical user interface.

Servlets can be embedded in many different servers because the servlet API, which you use to write Servlets, assumes nothing about the server's environment or protocol. Servlets have become most widely used within HTTP servers; many web servers support the Servlet API.

Use Servlets instead of CGI Scripts!

Servlets are an effective replacement for CGI scripts. They provide a way to generate dynamic documents that is both easier to write and faster to run. Servlets also address the problem of doing server-side programming with platform-specific APIs: they are developed with the Java Servlet API, a standard Java extension.

So Servlets are used to handle HTTP client requests. For example, have Servlets process data posted over HTTPS using an HTML form, including purchase order or credit card data. A servlet like this could be part of an order-entry and processing system, working with product and inventory databases, and perhaps an on-line payment system.

Other Uses for Servlets:

Here are a few more of the many applications for Servlets:

Allowing collaboration between people. A servlet can handle multiple requests concurrently, and can synchronize requests. This allows Servlets to support systems such as on-line conferencing.

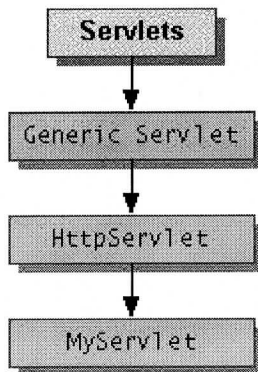
Forwarding requests. Servlets can forward requests to other servers and Servlets. Thus Servlets can be used to balance load among several servers that mirror the same content, and to partition a single logical service over several servers, according to task type or organizational boundaries.

Architecture of the Servlet Package

The `javax.servlet` package provides interfaces and classes for writing Servlets. The architecture of the package is described below.

The Servlet Interface:

The central abstraction in the Servlet API is the Servlet interface. All Servlets implement this interface, either directly or, more commonly, by extending a class that implements it such as `HttpServlet`.



The Servlet interface declares, but does not implement, methods that manage the servlet and its communications with clients. Servlet writers provide some or all of these methods when developing a servlet.

Client Interaction:

When a Servlet accepts a call from a client, it receives two objects:

A `ServletRequest`, which encapsulates the communication from the client to the server.

A `ServletResponse`, which encapsulates the communication from the Servlet back to the client.

`ServletRequest` and `ServletResponse` are interfaces defined by the `javax.servlet` package.

The ServletRequest Interface:

The `ServletRequest` interface allows the Servlet access to:

Information such as the names of the parameters passed in by the client, the protocol (scheme) being used by the client, and the names of the remote host that made the request and the server that received it.

The input stream, `ServletInputStream`. Servlets use the input stream to get data from clients that use application protocols such as the HTTP POST and PUT methods.

Interfaces that extend `ServletRequest` interface allow the Servlet to retrieve more protocol-specific data. For example, the `HttpServletRequest` interface contains methods for accessing HTTP-specific header information.

The ServletResponse Interface:

The `ServletResponse` interface gives the Servlet methods for replying to the client. It

Allows the Servlet to set the content length and MIME type of the reply.

Provides an output stream, `ServletOutputStream`, and a `Writer` through which the Servlet can send the reply data.

Interfaces that extend the `ServletResponse` interface give the Servlet more protocol-specific capabilities. For example, the `HttpServletResponse` interface contains methods that allow the servlet to manipulate HTTP-specific header information.

Additional Capabilities of HTTP Servlets:

The classes and interfaces described above make up a basic Servlet. HTTP Servlets have some additional objects that provide session-tracking capabilities. The servlet writer can use these APIs to maintain state between the servlet and the client that persists across multiple connections during some time period. HTTP Servlets also have objects that provide cookies. The servlet writer uses the cookie API to save data with the client and to retrieve this data.

The classes mentioned in the Architecture of the Servlet Package section are shown in the example.

`SimpleServlet` extends the `HttpServlet` class, which implements the `Servlet` interface.

`SimpleServlet` overrides the `doGet` method in the `HttpServlet` class. The `doGet` method is called when a client makes a GET request (the default HTTP request method), and results in the simple HTML page being returned to the client within the `doGet` method.

The user's request is represented by an HttpServletRequest object.

The response to the user is represented by an HttpServletResponse object.

Because text data is returned to the client, the reply is sent using the Writer object obtained from the HttpServletResponse object.

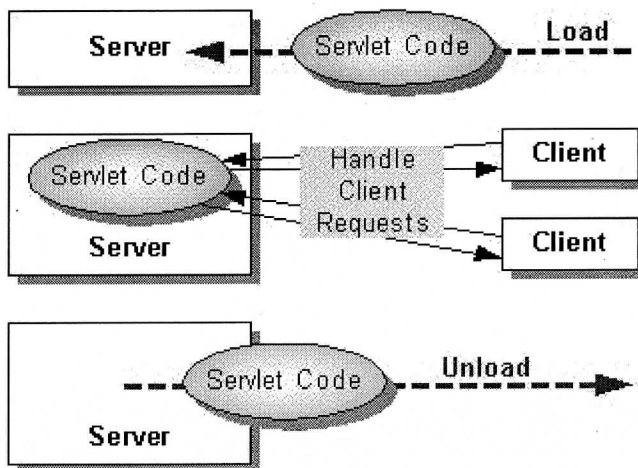
Servlet Lifecycle

Each Servlet has the same life cycle.

A server loads and initializes the Servlet.

The Servlet handles zero or more client requests.

The server removes the Servlet



Initializing a Servlet:

When a server loads a Servlet, the server runs the servlet's init method. Initialization completes before client requests are handled and before the servlet is destroyed. Even though

most Servlets are run in multi-threaded servers, Servlets have no concurrency issues during Servlet initialization. The server calls the init method once, when the server loads the servlet, and will not call the init method again unless the server is reloading the servlet. The server cannot reload a servlet until after the server has destroyed the servlet by calling the destroy method.

The init Method:

The init method provided by the HttpServlet class initializes the servlet and logs the initialization. To do initialization specific to your servlet, override the init () method following these rules:

If an initialization error occurs that renders the servlet incapable of handling client requests, throw an Unavailable Exception.

An example of this type of error is the inability to establish a required network connection.

Do not call the System. Exit method.

Initialization Parameters:

The second version of the `init` method calls the `getInitParameter` method. This method takes the parameter name as an argument and returns a `String` representation of the parameter's value.

The specification of initialization parameters is server-specific. In the Java Web Server, the parameters are specified with a servlet is added then configured in the Administration Tool. For an explanation of the Administration screen where this setup is performed, see the Administration Tool: Adding Servlets online help document. If, for some reason, you need to get the parameter names, use the `getParameterNames` method.

Destroying a Servlet:

Servlets run until the server destroys them, for example at the request of a system administrator. When a server destroys a servlet, the server runs the servlet's `destroy` method. The method is run once; the server will not run that servlet again until after the server reloads and reinitializes the servlet.

When the `destroy` method runs, another thread might be running a service request. The Handling Service Threads at Servlet Termination section shows you how to provide a clean shutdown when there could be long-running threads still running service requests.

Using the Destroy Method:

The destroy method provided by the HttpServlet class destroys the servlet and logs the destruction. To destroy any resources specific to your servlet, override the destroy method. The destroy method should undo any initialization work and synchronize persistent state with the current in-memory state.

The following example shows the destroy method that accompanies the init method shown previously:

```
public class BookDBServlet extends GenericServlet {  
  
    private Bookstore books;  
  
    ... // the init method  
  
    public void destroy () {  
  
        // Allow the database to be garbage collected  
  
        Books = null;  
  
    }  
  
}
```

A server calls the destroy method after all service calls have been completed, or a server-specific number of seconds have passed, whichever comes first. If your servlet handles any long-running operations, service methods might still be running when the server calls the

destroy method. You are responsible for making sure those threads complete. The next section shows you how.

The destroy method shown above expects all client interactions to be completed when the destroy method is called, because the servlet has no long-running operations.

Handling Service Threads at Servlet Termination

All of a servlet's service methods should be complete when a servlet is removed. The server tries to ensure this by calling the destroy method only after all service requests have returned, or after a server-specific grace period, whichever comes first. If your servlet has operations that take a long time to run (that is, operations that may run longer than the server's grace period), the operations could still be running when destroy is called. You must make sure that any threads still handling client requests complete; the remainder of this section describes a technique for doing this.

If your servlet has potentially long-running service requests, use the following techniques to:

- Keep track of how many threads are currently running the service method.

- Provide a clean shutdown by having the destroy method notify long-running threads of the shutdown and wait for them to complete.

- Have the long-running methods poll periodically to check for shutdown and, if necessary, stop working, clean up and return.

Tracking Service Requests

To track service requests, include a field in your servlet class that counts the number of service methods that are running. The field should have access methods to increment, decrement, and return its value.

The service method should increment the service counter each time the method is entered and decrement the counter each time the method returns. This is one of the few times that your HttpServlet subclass should override the service method. The new method should call super. Service to preserve all the original HttpServlet. Service method's functionality.

Providing a Clean Shutdown

To provide a clean shutdown, your destroy method should not destroy any shared resources until all the service requests have completed. One part of doing this is to check the service counter. Another part is to notify the long-running methods that it is time to shut down. For this, another field is required along with the usual access methods.

Creating Polite Long-running Methods

The final step in providing a clean shutdown is to make any long-running methods behave politely. Methods that might run for a long time should check the value of the field that notifies them of shut downs, and interrupt their work if necessary.

Servlet-client Interaction

Handling HTTP Clients

An HTTP Servlet handles client requests through its service method. The service method supports standard HTTP client requests by dispatching each request to a method designed to

handle that request. For example, the service method calls the `doGet` method shown earlier in the simple example servlet.

Requests and Responses

Methods in the `HttpServlet` class that handle client requests take two arguments:

An `HttpServletRequest` object, which encapsulates the data *from* the client.

An `HttpServletResponse` object, which encapsulates the response *to* the client

HttpServletRequest Objects

An `HttpServletRequest` object provides access to HTTP header data, such as any cookies found in the request and the HTTP method with which the request was made. The `HttpServletRequest` object also allows you to obtain the arguments that the client sent as part of the request.

To access client data:

The `getParameter` method returns the value of a named parameter. If your parameter could have more than one value, use `getParameterValues` instead. The `getParameterValues` method returns an array of values for the named parameter. (The method `getParameterNames` provides the names of the parameters.)

For HTTP GET requests, the `getQueryString` method returns a `String` of raw data from the client. You must parse this data yourself to obtain the parameters and values.

For HTTP POST, PUT, and DELETE requests, if you expect text data, the `getReader` method returns a `BufferedReader` for you to use to read the raw data. If you expect binary data, the `getInputStream` method returns a `ServletInputStream` for you to use to read the raw data

Note: Use *either* a `getParameter[Values]` method *or* one of the methods that allow you to parse the data yourself. They cannot be used together in a single request.

HttpServletResponse Objects

An `HttpServletResponse` object provides two ways of returning data to the user:

The `getWriter` method returns a `Writer`.

The `getOutputStream` method returns a `ServletOutputStream`.

Use the `getWriter` method to return text data to the user, and the `getOutputStream` method for binary data. Closing the `Writer` or `ServletOutputStream` after you send the response allows the server to know when the response is complete.

HTTP Header Data

You must set HTTP header data *before* you access the `Writer` or `Output Stream`. The `HttpServletResponse` class provides methods to access the header data. For example, the `setContentType` method sets the content type. (This header is often the only one manually set.)

Handling GET and POST Requests

The methods to which the service method delegates HTTP requests include,

`doGet`, for handling GET, conditional GET, and HEAD requests .

`doPost`, for handling POST requests.

`doPut`, for handling PUT requests.

`doDelete`, for handling DELETE requests .

By default, these methods return a `BAD_REQUEST` (400) error. Your servlet should override the method or methods designed to handle the HTTP interactions that it supports. This section shows you how to implement methods that handle the most common HTTP requests: GET and POST.

The `HttpServlet`'s service method also calls the `doOptions` method when the servlet receives an `OPTIONS` request and `doTrace` when the servlet receives a `TRACE` request. The default implementation of `doOptions` automatically determines what HTTP options are supported and returns that information. The default implementation of `doTrace` causes a response with a message containing all of the headers sent in the trace request. These methods are not typically overridden.

Servlet Descriptions

In addition to handling HTTP client requests, some applications, such as the Java Web Server's Administration Tool, get descriptive information from the servlet and display it. The servlet description is a string that can describe the purpose of the servlet, its author, its version number, or whatever the servlet author deems important.

The method that returns this information is `getServletInfo`, which returns null by default. You are not required to override this method, but applications are unable to supply a description of your servlet unless you do.

Writing Your First Servlet

Servlets are also easy to develop. This document discusses the following minimum steps needed to create any servlet:

1. Write the servlet
 - a. Import the necessary Java packages
 - b. Inherit from `GenericServlet` or the HTTP convenience class `HttpServlet`
 - c. Override the service method (this is where the actual work is done by the servlet)
 - d. Save the file with a `.java` filename extension
2. Compile the servlet
 - a. Make sure `jws.jar` is included in your class path
 - b. Invoke `javac`
3. Install the servlet
 - a. Use the Java Web Server's Administration Tool to install it, and optionally configure it.

4. Test the servlet
 - a. Invoke the servlet from a JDK1.1-compatible browser.

About Session Tracking

Session T tracking is a flexible, lightweight mechanism that enables stateful programming on the web. Its general implementation serves as a basis for more sophisticated state models, such as persistent user profiles or multi-user sessions.

A *session* is a series of requests from the same user that occur during a time period. This transaction model for sessions has many benefits over the single-hit model. It can maintain state and user identity across multiple page requests. It can also construct a complex overview of user behavior that goes beyond reporting of user hits.

Server-Side Session Objects and Users

Session tracking gives Servlets and other server-side applications the ability to keep state information about a user as the user moves through the site. Server-side applications can use this facility to create more stateful user experiences and to track who's doing what on the site.

Java Web Server maintains user state by creating a Session object for each user on the site. These Session objects are stored and maintained on the server. When a user first makes a request to a site, the user is assigned a new Session object and a unique session ID. The session ID matches the user with the Session object in subsequent requests. The Session object is then passed as part of the request to the Servlets that handle the request. Servlets can add information to Session objects or read information from them.

Session Endurance

After the user has been idle for more than a certain period of time (30 minutes by default), the user's session becomes invalid, and the corresponding Session object is destroyed. A session is a set of requests originating from the same browser, going to the same server, bounded by a period of time. Loosely speaking, a session corresponds to a single *sitting* of a single anonymous user (anonymous because no explicit login or authentication is required to participate in session tracking).

The first part of the doGet method associates the Session object with the user making the request. The second part of the method gets an integer data value from the Session object and increments it. The third part outputs the page, including the current value of the counter.

When run, this servlet should output the value of the counter that increments every time you reload the page. You must obtain the Session object before you actually write any data to the servlet's output stream. This guarantees that the session tracking headers are sent with the response.

The Session object has methods similar to java.util.Dictionary for adding, retrieving, and removing arbitrary Java objects. In this example, an Integer object is read from the Session object, incremented, and then written back to the Session object.

Any name, such as sessiontest.counter, may be used to identify values in the Session object. When choosing names, remember that the Session object is shared among any Servlets that the user might access. Servlets may access or overwrite each

other's values from the Session. Thus, it is good practice to adopt a convention for organizing the namespace to avoid collisions between Servlets, such as:

servletname.name

Session Invalidation

Sessions can be invalidated automatically or manually. Session objects that have no page requests for a period of time (30 minutes by default) are automatically invalidated by the Session Tracker `sessionInvalidationTime` parameter. When a session is invalidated, the Session object and its contained data values are removed from the system.

After invalidation, if the user attempts another request, the Session Tracker detects that the user's session was invalidated and creates a new Session object. However, data from the user's previous session will be lost.

Session objects can be invalidated manually by calling `Session.invalidate()`. This will cause the session to be invalidated immediately, removing it and its data values from the system.

Handling Non-Cookie Browsers (URL Rewriting)

The Session Tracker uses a session ID to match users with Session objects on the server side. The session ID is a string that is sent as a cookie to the browser when the user first accesses the server. On subsequent requests, the browser sends the session ID back as a cookie and the server uses this cookie to find the session associated with that request.

There are situations, however, where cookies will not work. Some browsers, for example, do not support cookies. Other browsers allow the user to disable cookie support. In such cases, the Session Tracker must resort to a second method, URL rewriting, to track the user's session.

URL rewriting involves finding all links that will be written back to the browser, and rewriting them to include the session ID. For example, a link that looks like this:

```
<a href="/store/catalog">
```

might be rewritten to look like this:

```
<a href="/store/catalog;$sessionid$DA32242SSGE2">
```

If the user clicks on the link, the rewritten form of the URL will be sent to the server. The server's Session Tracker will be able to recognize the `;$sessionid$DA32242SSGE2` and extract it as the session ID. This is then used to obtain the proper Session object.

Implementing this requires some reworking by the servlet developer. Instead of writing URLs straight to the output stream, the servlet should run the URLs through a special method before sending them to the output stream.

The `encodeUrl` method performs two functions:

1. **Determine URL Rewriting:** The `encodeUrl` method determines if the URL needs to be rewritten. Rules for URL rewriting are somewhat complex, but in general if the server detects that the browser supports cookies, then the URL is not rewritten. The server tracks information indicating whether a particular user's browser supports cookies.
2. **Return URL (modified or the same):** If the `encodeUrl` method determined that the URL needs to be rewritten, then the session ID is inserted into the URL and returned. Otherwise, the URL is returned unmodified.

In addition to URLs sent to the browser, the servlet must also encode URLs that would be used in `sendRedirect()` calls. For example, a servlet that used to do this:

```
response.sendRedirect ("http://myhost/store/catalog");
```

should now do this:

```
response.sendRedirect  
  
(response.encodeRedirectUrl ("http://myhost/store/catalog"));
```

The methods `encodeUrl` and `encodeRedirectUrl` are distinct because they follow different rules for determining if a URL should be rewritten.

Multiple Servlets URL conversions are required only if the servlet supports session tracking for browsers that do not support cookies or browsers that reject cookies. The consequences of not doing these conversions is that the user's session will be lost if the user's browser does not support cookies and the user clicks on an un-rewritten URL. Note that this can have consequences for other servlets. If one servlet does not follow these conventions, then a user's session could potentially be lost for all Servlets.

Using Session Tracking with the Page Compiler

Page compilation is a feature of the Java Web Server that allows HTML pages containing Java code to be compiled and run as Servlets. Page compilation also simplifies the task of supporting session tracking. To that end, if URL rewriting is enabled, page compilation automatically adds the `encodeUrl` call to links in the HTML page.

Overview of Oracle and SQL/PLSQL

Introduction to Oracle 9i

Oracle is a Comprehensive Operating Environment that packs the power of mainframe Relational database Management system into users microcomputers. It provides a set of functional programs that user can use as tools to build structures and perform tasks, because *applications* developed in oracle are completely portable to other versions of the programmer who can create a complex application in a single user environment and then move it to a multi user platform. Users do not have to be an expert to appreciate Oracle but, the better user understands the programs, the productively, creatively he can use the tools it provides.

Why Oracle?

I selected Oracle for developing the project work because it supports RDBMS features. Also it provides tools like SQL * PLUS, SQL*FORMS, SQL*REPORT WRITER, SQL*MENUS. Also it supports high security to the Data and faster Accessing capability.

It can be run on a variety of platforms and Operating systems. It provides Host language procedures like PRO*C, PRO*COBOL. An application that requires many lines of Host language code can be developed very easily. One can develop an Application easily by providing User-friendly Environment Support for Codd's Rules:

Oracle supports the following rules of Dr.E.F.CODD:

Rule1: Information Rule (Representation of information)

Rule 2: Guaranteed Access

Rule3: Systematic Representation of missing Information

Rule 4: Comprehensive on Line Catalogue

Rule 5: Comprehensive Data Sub-Language

Rule 6: View Updating

Rule 7: High Level Insert, Update, Delete

Rule 8: Physical Data Independence

Rule 9: Logical Data Independence

Rule 10: Integrity Independence

Rule 11: Distribution Dependence

Rule 12: Non-Subversion

Features of Oracle

Oracle is portable: The Oracle RDBMS is available on wide range of platforms, ranging from PCs to super computers and as a multi-user network loadable module (NLM) for Novell Netware. If you develop an application on one system you can run the same application on other systems without any modifications.

Oracle is Compatible: The Oracle command can be used for communicating with IBM, DB/2, Mainframe RDBMS, which is different from Oracle, i.e., Oracle is compatible with DB/2. Oracle is a high performance fault tolerant DBMS which is specially designed for on-line transaction processing and for handling the large database applications.

Oracle RDBMS is available with 2 options

Oracle RDBMS version 9i with transaction processing option and Oracle RDBMS version 9i without transaction processing options. Oracle is very high level of transaction processing, throughout, which is as follows:

The Row Level Lock Manager

PL/SQL a procedural language extension of SQL

Forms 5.0

Oracle Tools

Oracle is RDBMS, which stores and displays the Data in the form of tables. A table consists of rows and columns. A single row is called Record. Oracle is a modular system that contains Oracle Database (DB Manager) and several Tools (Functional Programs).

Oracle Tools do 4 major kinds of work

Database management

Data access and manipulation

Programming

Connectivity.

Data Access and Manipulation Tools

These are the tools used for communication with database manager for data access and manipulation. These tools can be used for not only access and manipulation but you can use design or use an application. Each tool Provides separate entry point and a unique approach to the Oracle system. The tools are firmly based on ANSI standard SQL.

SQL*PLUS

SQL* Plus is direct access to the Oracle RDBMS. You can see SQL commands to define, control and manipulate and query data. All users like DBA's, high-level system developers and others can talk straight in Oracle RDBMS.

Connectivity Tools

The connectivity tools help in connecting the Oracle databases through network and to other database systems. SQL* Plus allows for accessing the IBM, DB/2 (an IBM Mainframe RDBMS) and SQL/DS (Structured query language for data system) databases directly using the normal Oracle commands without doing any modifications.

SQL

The name SQL stands for structure query language. SQL is data access language, like any other language, it is used for communication. SQL communicates with database manager. The database manager could be Oracle, DB2, and SQL base, in grace or any RDBMS that supports SQL language. These database systems understand SQL.

SQL is easy to learn. Despite the fact that the SQL is a computer programming language, it is much simpler than traditional programming language like COBOL, BASIC, FORTRAN or APL. This is due to the fact that SQL is non-procedural language.

Features of SQL

SQL uses a free form (A non mathematical syntax), English like structure for its commands.

Ex: You select some data from your table, where certain conditions are met, you insert your values into some table; you delete data from some table where conditions are met. It is very logical and easy to follow.

SQL decides how it gets your data to and from database. All you have to specify is what and SQL does the rest. This is being called non navigational and it promises large productivity gains for the data processing identity. Sometimes programs in traditional processing system can be replaced with a single SQL query.

1. Most traditional RDBMS support both interactive and static SQL processing i.e. SQL statements can be executed in an alternative fashion where you talk directly to the database manager or SQL statement can be embedded in traditional computer programming language like COBOL. This is necessary because SQL is originally intended to use with other programming language. By itself it has no commands for screen dialogue or for more than crude report formatting. So this dual mode feature is very important in any kind of formal data processing application. The embedded SQL statements themselves are very simple to their interactive counter parts.
2. Finally, SQL process data at the set level, meaning your updates will change a set of records (rows) and query output will comeback in a set of records (a result table)

SQL Processing Capabilities

SQL is composed of a Definition language, a Data manipulation language and a Data control language. These three languages support the complete spectrum of Relational Data processing activity. In fact most SQL based products all access to the data through SQL.

1. Data definition language: DDL allows creation, deletion and modification of data structures for bar system. These structures include tables, databases, and indexes.

Ex: Creation, Drop, Alter.

2. Data Manipulation Language: These commands are used to manipulate the data in tables directly or through views. There are four standard DML statements. They are Delete, Insert, and Update.

3. Data control language: These commands are used to control usage and access of data. The most commonly found one's are Grant and Revoke.

PL/SQL

PL/SQL is an extension to SQL. It allows us to use all the SQL data Manipulation statements including insert, delete, update and select as well as the transaction processing statements Commit, Rollback and save point.

PL/SQL blocks can contain any number of SQL combined with the following:

Flow of control statements such as IF...THEN...ELSE, EXIT and GOTO.

Repetition statement such as for loop and while loop.

Assignment statements such as $x = y + z$ unlike SQL, PL/SQL allows logically group a set of statements and send them to the RDBMS as a single block

Advantages of PL/SQL

PL/SQL is completely portable, high performance Transaction processing language (TPL) that gives us more and better ways to express problems and design database application. Specifically PL/SQL provides the following advantages.

Procedural Capabilities.

Improved Performance

Enhanced Productivity

Portability

Integration with the RDBMS.

Procedural capabilities:

PL/SQL is a TPL that offers procedural solutions. It supports variable and constant declarations, error handling and a wide variety of useful functions within the same PL/SQL block; we can use SQL and all the PL/SQL extensions.

Improved performance:

Without PL/SQL the Oracle RDBMS must Process SQL statements one at a time. Each SQL statement results in another call to be RDBMS and higher performance overhead can become significant when we are issuing many SQL statements in a network environment.

Enhanced Productivity:

PL/SQL also brings added functionality to non-procedural tools Such as SQL forms. With PL/SQL in these tools, software developers can use familiar procedural language construct to develop applications.

Portability:

Applications written in PL/SQL are portable to any computer hardware and operating system environment running the oracle version 8i RDBMS.

Integration with the RDBMS:

Both PL/SQL and Oracle have their foundation in SQL also. Most PL/SQL variable has Data types native to the RDBMS Data Dictionary

Main Features of SQL

SQL Data manipulation statements are built into PL/SQL. This allows inserting new data into a database, retrieving, modifying and deleting data.

Support for SQL

By extending SQL, PL/SQL offers a unique combination of power and ease of use. We can access our Oracle database and manipulate its data flexibly and safely because PL/SQL supports SQL DML statements, SQL TPL statements, SQL functions and SQL predicates. SQL data definition statements such as Alter, Create and Rename and data control statements connect; Grant and Revoke statements are supported.

SQL Data Manipulation Statements:

A transaction is a sequence of SQL statements that Oracle treats as a unit, so that all changes brought about by the statements are made permanent or undone for the same time. The consistency of the database PL/SQL lets you use the Commit, Rollback and Save point statements. The Commit statement makes permanent any changes made during the current transaction until you commit your changes, other users cannot see them. The Rollback statement ends the current transaction and undoes any changes made since the transaction began. The Save point statement marks the current point in the processing of a transaction.

JDBC (Java Database Connectivity)

The JDBC interface is a pure Java API used to execute SQL statements. The JDBC provides a set of classes and interfaces that can be used by developers to write database applications. It can be broken down into four steps.

Open a Connection to the Database.

Execute a SQL statement.

Process the results.

Close the connection to the database.

The JDBC provides support for two and three-tier database access models.

If you use the two-tier database access model, your Java application talks directly to the database. The results of these commands are then sent back from the database directly to the application.

When you use three-tier model, your JDBC sends commands to a middle-tier, which in turns sends commands to the database. The results of these commands are then sent back to the middle-tier, which communicates them to the application.

JDBC Driver Types

JDBC-ODBC Bridge, plus ODBC Driver

Native-API, Partly- Java Driver

JDBC-Net, Pure Java Driver

Native-Protocol, pure Java Driver

JDBC-ODBC Bridge

It provides JDBC access to databases through ODBC drivers. The ODBC driver must be configured on the client for the bridge to work.

Native-API

The native-API driver converts JDBC commands into DBMS-specific native calls. The client must have some binary code loaded on its machine.

JDBC-Net, Pure Java Driver

The JDBC-Net drivers are a three-tier solution. This type of driver translates JDBC calls into a database-independent network protocol that is sent to a middleware server. This server then translates this DBMS-independent protocol into a DBMS-specific protocol, which is sent to a particular database.

Native-Protocol, pure Java Driver

These drivers are pure Java drivers that communicate directly with the vendor's database. They do this by converting JDBC commands directly into the database engine's native protocol.