MIDDLE EAST TECHNICAL UNIVERSITY

ENGINEERING FACULTY

DEPARTMENT OF COMPUTER ENGINEERING

FINAL PACKAGE REPORT

SUBJECT: ONLINECV

ASSISTANT: ALI ORKAN BAYER

"I$KOLIK" *BY AIVA*

Contents:

1. User's Guide

When an internet user navigates our website's home page, he/she will be prompted to enter the login information or register to our system.

If user registration has not been done before, he/she will be registered by filling information in registration pages. There are separated pages for employers and job-seekers registration.

After registration, user needs to wait for administrator approval which is essential for authentication. After login the user will be automatically directed to Employer/Job- seeker/ Administrator profile page.

1.1 Job Seeker Manual

When a job-seeker is on his/her profile page, there will be user profile info (name, surname, e-mail etc.) and links for CV-creation, announcement search, visualization of notifications. User is able to see the links of the announcements. All of the information regarding to the announcement may be reached with these links which are external web-site links.

There are lots of criteria for announcement search (sector, position, department, country, city, military status, driving license, the start and final dates of the announcement). User is able to search appropriate jobs with all of these criteria.

User is able to visualize links of the announcements which are selected by our system. Job-seeker is prompted to enter sector, position, country, city information which are used by our system for notifications.

CV-creation pages are separated into seven pages. CV-form1 contains the general information of the CV. CV-form2 contains the sector, position, city, country preference information.

Experience, Foreign Language, School, Reference, Computer Skill pages contains a button for adding new experience, foreign language, school, reference, computer skill information. User also can see the added information on these pages.

1.2 Employer's Manual

When an employer is on his/her profile page, there will be user profile info (name, surname, e-mail, company name, sector, position, city etc.) and links for announcement-creation, CV search, visualization of notifications. User is able to see the links of the CVs. All of the information regarding to the CVs may be reached with these links which are external web-site links.

There are lots of criteria for CV search (sector, position, department, country, city, military status, driving license). User is able to search appropriate jobs with all of these criteria.

User is able to visualize links of the CVs which are selected by our system. Employer is prompted to enter sector, position, country, city information which are used by our system for notifications.

CV-creation pages are separated into three pages. Announcement-form contains general and detailed information of the announcement.

Foreign Language, Computer Skill pages contains a button for adding new foreign language, computer skill information. User also can see the added information on these pages.

2. Developer's Manual

2.1 System Requirements

Our project being a web-application is built using Java technology, so it is platform independent.

The server machine requires that Java Enterprise Edition is installed. An a server application certified by Java Enterprise Edition 5. Here is the list:

- Sun Java System Application Server Platform Edition 9.0, based on the open-source server GlassFish
- WebLogic Application Server 10.0 from BEA Systems
- SAP NetWeaver Application Server, Java EE 5 Edition from SAP
- JEUS 6, an Application Server from TmaxSoft
- Apache Geronimo 2.0
- IBM WebSphere Application Server Community Edition 2.0, based on Apache Geronimo
- Oracle Containers for Java EE 11
- GlassFish
- Apache OpenEJB via Apache Geronimo

As database management system "Oracle 10g Express Edition" has been used, so Oracle 10g compatible database management system is required.

2.2 Deployment

Deployment of Iskolik is standard deployment of a J2EE application. Iskolik is packed into an EAR package.

2.2.1 WAR archive

WAR file (which stands for "web application archive" [1]) is a JAR file used to distribute a collection of JavaServer Pages, servlets, Java classes, XML files, tag libraries and static Web pages (HTML and related files) that together constitute a Web application.

A WAR file may be digitally signed in the same way as a JAR file in order to assert that the code is trusted. There are special files and directories within a WAR file. The /WEB-INF directory in the WAR file contains a file named web.xml which defines the structure of the web application. If the web application is only serving JSP files, the web.xml file is not strictly necessary. If the web application uses servlets, then the servlet container uses web.xml to ascertain which servlet a URL request should be routed to. web.xml is also used to define context variables which can be referenced within the servlets and it is used to define environmental dependencies which the deployer is expected to set up. An example of this is a dependency on a mail session used to send email. The servlet container is responsible for providing this service.

Unfortunately, one disadvantage of web deployment using WAR files in very dynamic environments is that minor changes cannot be made during runtime. Any change whatsoever requires regenerating and redeploying the entire WAR file.

2.2.2 EAR archive

An Enterprise ARchive, or EAR, is a file format used by Java EE for packaging one or more modules into a single archive so that the deployment of the various modules onto an application server happens simultaneously and coherently. It also contains XML files called deployment descriptors which describe how to deploy the modules.

An EAR file is a standard JAR file with an .ear extension, with one or more entries representing the modules of the application, and a metadata directory called META-INF which contains one or more deployment descriptors.

Different artifacts can be embedded within an EAR file, artifacts which are deployed by the application server:
A Web module is a WAR file.The web module is contained in a hierarchy of directories and files in a standard web application format.
POJO Java classes may be deployed in .jar files.
An Enterprise Java Bean module has a .jar extension, and contains in its own META-INF directory descriptors describing the persistent classes deployed. When deployed, entity beans are visible to other components and (if remotely exported), remote clients. Message Beans and Session Beans are available for remote access.
A Resource Adapter module has a .rar extension

Most application servers load classes from a deployed EAR file as an isolated tree of java classloaders, isolating the application from other applications, but sharing classes between deployed modules. For example, a deployed WAR file would be able to create instances of classes defined in a JAR file that was also included in the containing EAR file, but not necessarily those in JAR files in other EAR files. One key reason for this behaviour is to allow complete separation between applications which use static singletons (e.g. Log4J), which would otherwise mess-up the configuration between separate applications. This also enables different versions of applications and libraries to be deployed side-by-side.

The JBoss application server is notable in that it does not isolate deployed components. A web application deployed in one EAR file would have access to classes in other EAR and WAR files. This is a somewhat controversial policy. The Unified Classloader design reduces communications overhead between running applications, as class data can be shared by reference or simple copies. It also allows developers to avoid having to understand the problems that a tree of classloaders can create. However, it prevents different versions of dependent libraries from being deployed in separate applications. JBoss 4.0.2 switched to a hierarchical classloader, but as of version 4.0.3, it has reverted to a Unified Classloader for backwards compatibility reasons. There is now a configuration option to change this behavior.

The META-INF directory contains at least the application.xml deployment descriptor, known as the J2EE Deployment Descriptor. It contains the following XML entities:
icon, which specifies the locations for the images that represent the application. A subdivision is made for small-icon and large-icon.
display-name, which identifies the application
description
A module element for each module in the archive

Zero or more security-role elements for the global security roles in the application

Each module element contains an ejb, web or java element which describes the individual modules within the application. Web modules also provide a context-root which identifies the web module by its URL.

Next to the J2EE deployment descriptor one can have zero or more runtime deployment descriptors. These are used to configure implementation-specific J2EE parameters.

2.2.3 EAR deployment

EAR file deployment is made by server/IDE specific utilities. Developer should consult relevant server/IDE reference.

2.2.4 Database Deployment

Proper database management system application must be installed, and connection to it must be installed. Into it sql-scripts, provided with package, should be run. Within Iskolik database connection configuration must be made.

2.3 Development

"Iskolik" is a Java Enterprise Edition Project. Java Platform, Enterprise Edition or Java EE is a widely used platform for server programming in the Java programming language Our project consist of JSP pages and java classes that run behind the pages and handle the business logic and java classes that invoke the web services. Java beans have been used for carrying information during session.

2.3.1 Modules

There are five modules namely Job-seeker module, Employer module, Administration module, Registration/Authentication module, Web-services modules in this project.

2.3.1.1 Job-Seeker

In order to achieve the capabilities the Job seeker profile has, there are JSP pages and Java classes running behind them. js_profile.jsp, Jobseekers.jsp, insertcv.jsp, ExperiencesMain.jsp, experiences.jsp, cv_form1.jsp, cv_form2.jsp, ComputerSkillsMain.jsp, comp_skill.jsp, addNotification.jsp, js_search.jsp, js_searching.jsp, languages.jsp, LanguagesMain.jsp, NotificationMain.jsp, references.jsp, ReferencesMain.jsp, school.jsp, SchoolMain.jsp are the pages belonging to the Job-Seeker module. Schools.java, Reference.java, Mainclass.java, ForeignLanguages.java, DB.java, CVdraft.java, ComputerSkill.java are the classes that act as JavaBeans behind the pages and perform the cv insertion. displayNotificationJS.java, Notification.java are the classes that present the notifications for the job-seeker profile. StringOperations.java is the class that acts as helper class for the string operations for example it is used for the verification of the strings entered from the user interface.

2.3.1.2 Employer

As mentioned for the Job-Seeker profile, this module also consists of jsp pages and the classes that run behind them and take care of the business logic. AddNotification.jsp, ann_saving.jsp,

anouncement_form.jsp, comp_skill_emp.jsp, displayNotificationEmp.jsp, emp_language.jsp, emp_register.jsp, emp_search.jsp, emp_searching.jsp, employer_profile.jsp, employernotification.jsp are the pages reserved for Employer profiled users' service. The classes that run the logic behind are DB.java, Employer_profile.java, Mainclass.java, StringOperations.java, Notification.java.

2.3.1.3 Administrator

Administrative jobs are handled by the jsp pages and the "sdo"s within the pages.

The pages are namely adminLogin.jsp, adminLogging.jsp, adminMain.jsp.

2.3.1.4 Registration/Authentication

Registration is handled by empregistration.jsp, emp_register.jsp, js_register.jsp, jsregistration.jsp, register.jsp pages with Mainclass.java, EmpRegister.java, register.java. Login and logout functionalities are handled by Login.jsp, logging.jsp, loggingOut.jsp pages with Mainclass.java.

2.3.1.5 Web-Services

Web services that handle the jobs on the external websites are invoked by bean structured classes under src/org/tempuri folder .These web services have the sample client codes in folders with names ending with "SoapProxy" under Webcontent.

2.3.2 External Systems

External systems can be interated into I$kolik with little effort. All interaction with external systems being done with web-services all that needs to be done is to generate client proxies and insert function calls to those.
There is one implementation of external system which is essentially a feature-rich web-site written using Microsoft ASP.NET technology in C£. It consists of ASP.NET pages and C# classes that run behind the pages and handle the business logic and database operations. Session arrays have been used for carrying information during session. External Website has 3 user types, namely, Job-Seeker, Employer and Administrator.

- Mainpage.aspx: Start page of external website which includes login box for all users. Database connection and login information check is done in cs class of this page.

Job-Seeker Related Pages;

- Registration.aspx: Registration page for Job-Seeker, in this page job-seeker registration

information filled from user is inserted to database by insert functions defined in Registration.cs

- Profile_js.aspx: Personal information of user that registered as job-seeker is displayed in this page.

- CV_form.aspx & CV_form2.aspx & References.aspx & Schools.aspx & Experience.aspx & Comp_Skill.aspx & For_Lang.aspx & CV_Title.aspx : All these pages are designed to insert CV information of registered and online user into database system.

- CV_list.aspx: Basic information of all submitted CV's of related user are listed in this page by datagrid.

- Cv_show.aspx: Detailed information of a selected CV is displayed in this page by the help of datagrid and sql query functions.

Employer Related Pages

- Emp_Registration.aspx: Registration page for Employer, in this page employer registration information filled from user is inserted to database by insert functions defined in Emp_Registration.cs

- Profile_emp.aspx: Personal information of user that registered as employer is displayed in his page.

- Ann_form.aspx & Comp_Skill_Emp.aspx & For_Lang_Emp.aspx & Ann_Title.aspx: All these pages are designed to insert Job Announcement information of registered and online user into database system.

- Ann_list.aspx: Basic information of all submitted Job Announcements of related user are listed in this page by datagrid.

- ann_show.aspx: Detailed information of a selected Job Announcement is displayed in this age by the help of datagrid and sql query functions.

Administrator Related Pages

- admin/listing_ia.aspx: this page is where administrator can list information of all job-seekers and decativate or activate registered job-seekers.

- admin/listing_iv.aspx: this page is where administrator can list information of all employers and decativate or activate registered employers.

2.3.3 Web-Services

Web-services are the only way of interacting with external system. It provides great flexibiliy and decoupling. Web-services client proxies are created using WSDL2Java utilities. Special package org.tempuri contains web-service related Java code.