

# B.Sc. in Business Computing

**DT354/4**

**Recruitment Solution**

**By**

**Author – Xiuzhen Chen**

**Student Number**

**C10712147**

**Submitted in partial fulfilment of the requirements for the degree of**

**B.Sc. in Business Computing**

**Dublin Institute of Technology**

**Year 4**

**Supervised by – Mary Regan**

**June 2013**

**Acknowledgements**

Through out my project I would to acknowledge several people in which I received help with in certain aspects of the project. I would like to thank firstly my supervisor Mary Regan for helping me with certain aspects of the project. She took time out of her busy schedule to meet up and talk about my project.

The next I would like to thank my family and friends for supporting me throughout the project. They helped me come up with new ideas for the project.

**Declaration**

**This is an original work. All References and assistance are acknowledged.**

**Signed: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Date: 9th May**

Table of Contents

[B.Sc. in Business Computing 1](#_Toc384612037)

[Chapter 1 Introduction 6](#_Toc384612038)

[Project Overview 6](#_Toc384612039)

[Project Plan 6](#_Toc384612040)

[Objectives of the recruitment solution 7](#_Toc384612041)

[Business Case 8](#_Toc384612042)

[Chapter 2: Requirements capture and analysis 9](#_Toc384612043)

[2.1 Actors 9](#_Toc384612044)

[Chapter 3: Design 12](#_Toc384612045)

[System Architecture 12](#_Toc384612046)

[3.1 SSH2 framework 13](#_Toc384612047)

[3.1.1 Struts2 14](#_Toc384612048)

[3.1.2 Spring 15](#_Toc384612049)

[3.1.3 Hibernate 16](#_Toc384612050)

[3.1.4 SSH2 integration framework 17](#_Toc384612051)

[3.2 Front End 18](#_Toc384612052)

[3.2.1 JQuery 18](#_Toc384612053)

[3.2.2 Ajax 18](#_Toc384612054)

[3.2.3 HTML 19](#_Toc384612055)

[3.2.4 CSS 19](#_Toc384612056)

[3.2.5 JSP 19](#_Toc384612057)

[3.3 Choice of Technologies for Recruitment Solution 20](#_Toc384612058)

[3.3.1 Database Options 20](#_Toc384612059)

[3.3.2 Database Connectivity Options 21](#_Toc384612060)

[3.3.3 Web Server Optons 22](#_Toc384612061)

[3.3.4 SMTP Plugin 23](#_Toc384612062)

[3.3.5 RMI 23](#_Toc384612063)

[3.3.6 GIT 23](#_Toc384612064)

[Chapter 4 Implementation 23](#_Toc384612065)

[4.1 Environment Set Up 24](#_Toc384612066)

[4.1.1 AOP(Aspect-oriented programming) 24](#_Toc384612067)

[4.1.2 Spring control all action 24](#_Toc384612068)

[4.2 Project Structure 24](#_Toc384612069)

[JSP 26](#_Toc384612070)

[OGNL(Object Graph Navigation Library) 26](#_Toc384612071)

[JSTL(JavaServer Pages Standard Tag Library) 26](#_Toc384612072)

[4.3 Registration 26](#_Toc384612073)

[4.3.1 JavaScript, JQuery and AJAX 26](#_Toc384612074)

[4.4 Authentication 27](#_Toc384612075)

[4.4.1 RMI and Session 27](#_Toc384612076)

[4.5 List and Search 28](#_Toc384612077)

[4.5.1 Select and Select2 28](#_Toc384612078)

[4.6 Post Job 29](#_Toc384612079)

[4.7 Delete 29](#_Toc384612080)

[4.8 Update 29](#_Toc384612081)

[4.9 Upload CV 29](#_Toc384612082)

[4.10 Download CV 30](#_Toc384612083)

[4.11 Send Interview 30](#_Toc384612084)

[Update Interview result 31](#_Toc384612085)

[Administrator 32](#_Toc384612086)

[JFreeChart 33](#_Toc384612087)

[Chapter 5 User Guidelines 35](#_Toc384612088)

[5.1 Main Page 35](#_Toc384612089)

[5.2 Registration 37](#_Toc384612090)

[5.2.1 JobSeeker Registration 37](#_Toc384612091)

[5.2.2 Employer Registration 38](#_Toc384612092)

[5.3 Jobseeker 39](#_Toc384612093)

[5.3.1 Authentication 39](#_Toc384612094)

[Upload CV 41](#_Toc384612095)

[View Interview History 41](#_Toc384612096)

[List Skill 42](#_Toc384612097)

[Update Password 42](#_Toc384612098)

[List CV(还没完成) 43](#_Toc384612099)

[Employer 43](#_Toc384612100)

[Authentication 43](#_Toc384612101)

[Profile 44](#_Toc384612102)

[Post Job 45](#_Toc384612103)

[Sent Interview 45](#_Toc384612104)

[Update Interview 45](#_Toc384612105)

[Administrator 46](#_Toc384612106)

[Chapter 6 Installation Manual 46](#_Toc384612107)

[Conclusion 49](#_Toc384612108)

[Reference 50](#_Toc384612109)

# Chapter 1 Introduction

**Project Title**

Online recruitment solution

## Project Overview

The system uses the modularized program design method, and to the full of Struts2, Spring, Hibernate , though MYSQL Database server, it exploits MVC design pattern, therefore it provides data query, modify, add, delete, maintenance function. It can realize the member information management. This system is reliable, scalability, reusability, and it provides a convenient searching platform for the users. The integrated development environment of the systems use MyEclipse 8.5, JDK6.0 and Tomcat 6.0, and the database use MYSQL 5.5 database server.

## Project Plan

My plan is to complete this product by getting basic function work first, therefore I will ensure I got basic mark before December. Then build more technical functionalities after December.

Preparation for 1nd checkpoint(30th September to 6th Octobtor)

**Preparation**

I will be work on revision on J2EE, doing exercise on what we learned from last year and gathering more ideas and getting more technique for FYP. I will start from bottom, set up tomcat environment, build JSP to display web page. Connect to database with JDBC, using framework of struts 2.

First checkpoint(7th October to 28th October)

I will start build database of product, to complete following functionalities:

Build backside management interface

* Creation of database
* connect with database
* Registration for employer, job seeker
* Delete users (employer, job seeker)
* Update users (employer, job seeker)
* Display users(employer, job seeker, advertised jobs)
* Build interface of backside management

Preparation for 2nd checkpoint (29th October to 29th November)

I will build employer interface and implement with following functionalities:

* Send registered notification to employer/job seeker
* Search job seekers
* Display jobs seekers
* Save job seekers
* Send interview invitation to job seekers, job seeker receive email
* Edit jobs, company profile

Preparation for 2nd checkpoint (30th November to 30th December)

I will build job seeker interface and implement with following functionalities:

* Edit/upload CV, cover letter
* Edit/update job seeker profile
* Search jobs
* View jobs
* Apply jobs
* Integrated product with google search engine

2nd Checkpoint(1 January to 26th January)

* Testing all functionalities
* Add feature of high chart to display volume of job seeker, employers, jobs
* Integrate framework with Spring, Hibernate, Struts2

**Constraints and Assumptions**

I presume that integrate framework of Spring, Hibernate and Struts could be complicated as I built whole framework with struts then integrated others.

**Risks and Contingencies**

Risk of doing this project is low, as this project is expand on the technologies that I have done before.

## Objectives of the recruitment solution

The main purpose of this product helps employers find suitable jobseekers, as well job seekers will search right jobs. This product is not only design for job seeker and employer, also more importantly for administrators who manage the data that stored in database at back end.

For employer who registered with this product by registration form, then receiving notification email confirming registered. Employer can login into this website to search applicant, to update company information, view jobs advertised and who applied, manage/edit job advertised.

For job seeker who registered with this site, will receive notification email confirming registered. Job seeker use this website to find jobs and able to search and view jobs advertised by employer, save the jobs they interested and share it into social networking site, edit/upload CV, cover letter, update user profile, contact with administrator.

Administrator is categorized into two types, one is head of administrator has right of view all users and delete users who do not login into this website for certain period of time or user is posting unrelated information. Administrator who has right of view of users details, report any activities to head of administrator.

## Business Case

When I was at the decision making of what sort of project I was going to follow, this project stood out to me the most. The idea behind this project was not only to learn from developing J2EE and using various technologies around it, but also to develop an idea that could incorporated in people’s jobs to ensure data safety, because I am passionate about web development, I believe that this would have being the best project that I would take pleasure in the most.

My inspiration and motivation behind the project was not just the learning of different technologies, but also change the way people searching for jobs or jobseekers. My intension is to try and make recruitment site more convenience to use when they use this application readily available for help. As today’s environment online market are all over places, especially in web site. These largely unrelated data are time consuming and distraction for user to search required data. I developed a way to eliminate irrelevant data and faster research time in busy environment. Jobseeker only view the jobs with preset skills when login. Employer only views jobseekers with relevant skills by posting jobs. It makes research information easier, faster, and more convenient.

With technology being integrated in everyday life, it is my belief that people could make more use of technologies to assits them in life and hopefully my project will demonstrate the use of technology in everyday life. I feel it is important that people should jump on board and use these technologies and not to be set back in learning them.

# Chapter 2: Requirements capture and analysis

## 2.1 Actors

This application is designed for three different users: administrators, jobseekers and employers.

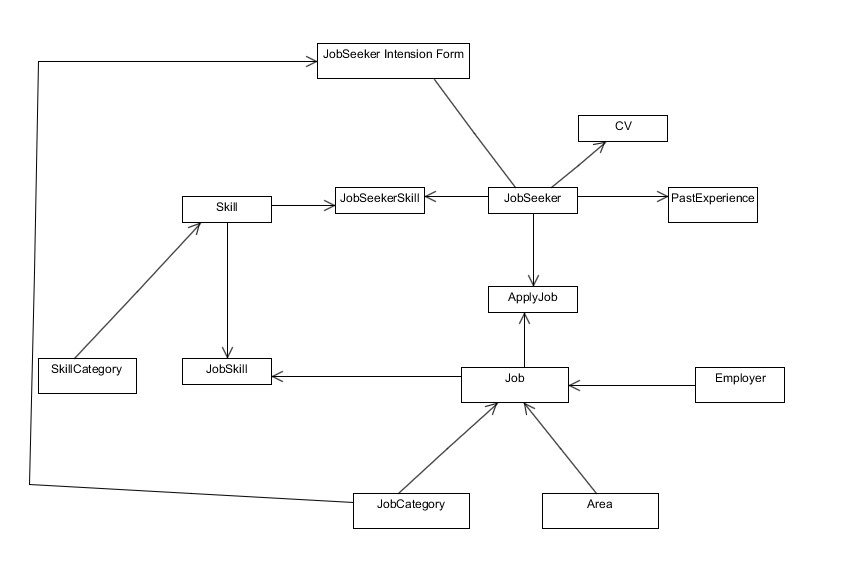
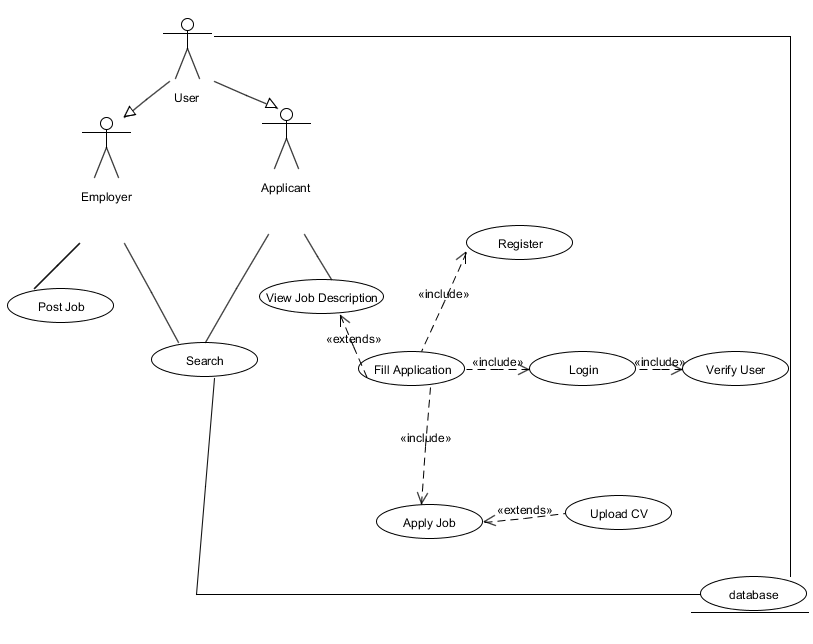
New jobseeker can choose a specific job category at registration and will only see jobs in that category that they can apply to. Similarly, all employers can only view jobseekers in their specific category by posting jobs. This function greatly reduces the amount of searching for both jobseekers and employers.

*Administrator*: the main user of the application is administrator of the web site. They will be responsible for managing data. Has ability to manage the entire database and can add, update, search and delete the existing data.

*Jobseeker*: able to register, login to the site, update password and profile, view interview histories and upload their CV and apply for jobs,

*Employer*: can register, login, update password and profile, post jobs and set up interviews.

**Business Process overview**



Employer(empId, userName, password, email, phone, contactName, companyName, address, webSite, companySize, type)

JobSeeker(jsId, userName, password, name, address, phone, email, expectedSalary)

CV(cvId, cvTitle, jsId, coverLetter)

PastExperience(pastExId, jobDescription, duty, empName, email, phone, address, startDate, endDate, reasonForLeave, jsId,)

Job(jobId, jobDesc, startDate, empId, address, areaId, phone, numPosition, requirement, salary, jobCategoryId)

JobCategory(CategoryId, name, parentId)

Area(areaId, county, area, latitude, longitude)

JobApplication(applyJobId, jobId, jobSeekerId, dateApplied)

Skill(skillId, skillName, skillCategoryId)

SkillCategory(skillCategoryId, name, parentId)

jobSeekerSkill(jsSkillId, skillId, jsId)

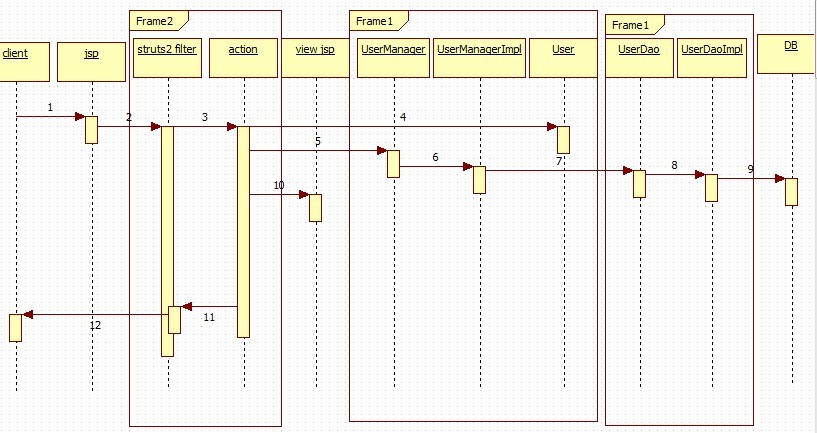
jobSkill(jskillId, skillId, jobId)

JobSeekerIntensionForm(intensionFormId, expectedSalary, startDate, expectedJob, jobSeekerId)

# Chapter 3: Design

## System Architecture

The recruitment solution project is based on web development by used technologies of integrating Spring, Struts2 and Hibernate framework to build application.



Client who access to jsp page that managed by struts2 filter controller specified by action that direct to relative view jsp page.

In Manager level consist of UserManager is interface, userManagerImpl is implementation of User Interface. User is object. In this level returns request view back to view jsp send to client.

In Dao level consist of UserDao that is interface, and UserDaoImpl is implementation of UserDao. In this level deal with DB(database).

This section introduces the relevant technologies used in this system, include Struts2, Spring, Hibernate, JQuery, Ajax.

## 3.1 SSH2 framework

The system is mainly to achieve some of the basic functions needed for recruitment solution system. I used Java language for development. Front end using JSP, CSS, JavaScript, Jquery, back end exploit MVC layer 3 architecture development models. I use Struts2, Spring and Hibernate three major framework for development of the system. Integrate development environment using MyEclipse8.5, database choose MYSQL5.5, Web server used by Tomcat6.0.

### 3.1.1 Struts2

It is open-source of powerful application framework, which is the next generation of Struts that is based Struts1 and WebWork technology to carry out merger with brand new Struts 2 framework. There are huge differences in architecture between Struts and Struts2. Struts 2 take WebWork as the core, the interceptor mechanism  to process with the user's request, such design enable the business logic controller to separated completely from with Servlet API, Struts 2 is the WebWork renewal product. Although Struts 1 and 2 have a big change, but for WebWork, there is small change in Sturt2.

Struts1 and Struts2 system is very different because the Struts2 uses WebWork core, rather than Struts1 by design core. Struts2 extensively use of interceptors to process the user's request, allowing users to separate business logic controllers and Servlet API.

Struts2 framework processing as following:

1) Load Class (FilterDispatcher)

2) Reads the configuration (Action in Struts configuration file)

3) Dispatch request (client sends a request)

4) Call Action (FilterDispatcher read corresponding Action from the Struts configuration file)

5) Enable interceptor (WebWork interceptor link automatically apply to requests functions such as validation)

6) Processing operations (returns Action's execute () method)

7) Returns a response (by the execute method return the information to the FilterDispatcher)

8) Search response (FilterDispatcher find what response is based on configuration information, such as: SUCCESS, ERROR, which will return the JSP page)

9) Responding to user (Figure 1-1 shows the flowchart)

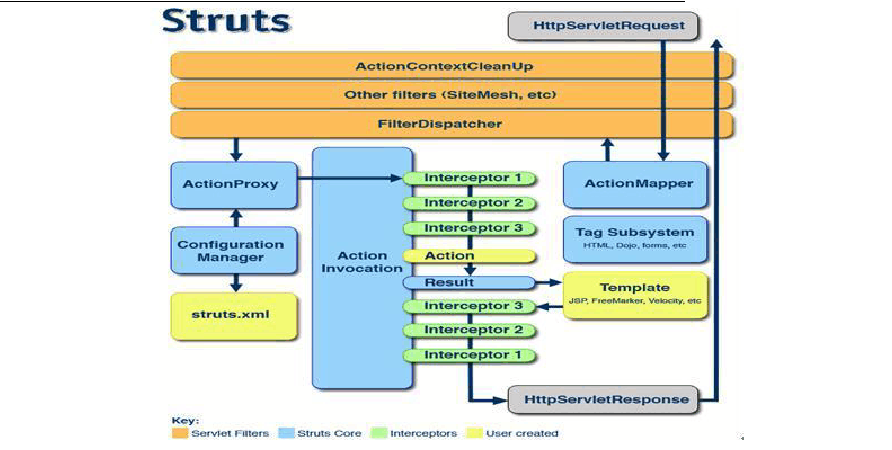


Figure 1 - 1 Strut2 access flow chart

### 3.1.2 Spring

It is an open source framework that created to address the complexity of enterprise application development. However, the Spring is not limited to the use of server-side development. From the Simplicity, testability, and loosely-coupled, any Java applications are available from the Spring benefit. Although the framework does not impose any specific programming model, it has become popular in the Java community as an alternative to, replacement for, or even addition to the Enterprise JavaBean (EJB) model.

1) Objective: To solve the complexity of enterprise application development

2) Features: uses basic JavaBean instead of EJB, and provides additional enterprise features

3) Range: Spring is a lightweight inversion of control (IOC) and aspect oriented (AOP) of the container framework

4) Lightweight: Complete the Spring framework can be released in JAR file with one size only about 1 MB. And Spring the processing overhead is very cheap. In addition, In addition, Spring non-intrusive: typically, Spring Application object does not reply on the a particular class of Spring.

5) Inversion of control: Spring through a known as inversion of control (IOC) technology for loose coupling. When has applied IOC, object depends on other objects can transmit through the passive way, rather than the objects you create or find dependent objects. This is considered as the IOC and JNDI.

6) Aspect oriented--Spring provides rich support for aspect oriented programming, to allow the separation of business logic from the application of system-level services (such as audit and transaction management), carries on development of cohesion. Application object is only to complete business logic.

7) Container, Spring Contains and manages Application object's configuration and the life cycle, and in that sense it is a container, you can configure how each bean to be created based on a configurable prototype.

8) Framework--Spring can configure simple components, combine them into complex applications. Application object is declared in combination, typically in a XML files. Spring also provides a lot of infrastructure functions, transaction management, persistence, framework integration, etc. Developer can decide logical development, all Spring features enables programmers to write more clean, more manageable, and easier to test the code.

### 3.1.3 Hibernate

Hibernate, is known as the best and dominated object/relational persistence (ORM) tool for Java language, providing a framework for mapping an object-oriented domain model to a traditional database. It provides many elegant and innovative ways to simplifies the relational database handling task in Java.it had a very lightweight JDBC encapsulation so that Java programmers can use its own programming ideas to manipulate the database. Hibernate can be applied to any use JDBC applications,  use either a Java client program, also in the Servlet/JSP Web application.(Diagram as shown in Figure 1 -2)

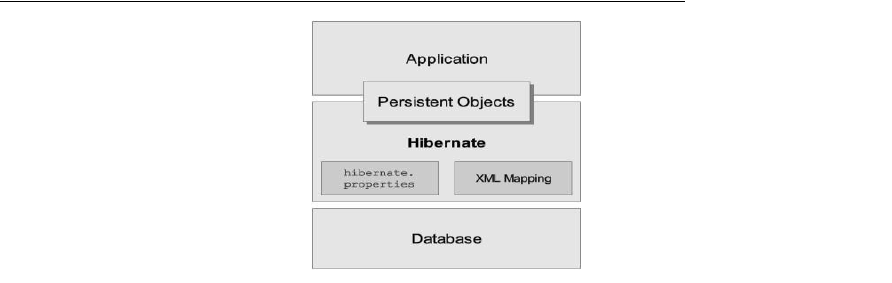
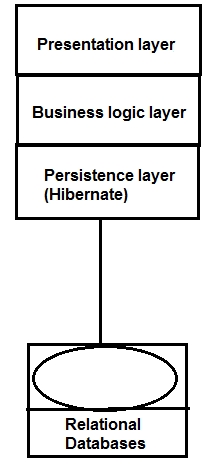


Figure 1 - 2 Hibernate architecture overview diagram

Hibernate have 6 Hibernate Core interfaces, they are: Session, SessionFactory, Transaction, Query, Criteria and Configuration. These six core interfaces will be used in any development. Through these interfaces, you can not only access to persistent objects, can also be used for transaction control. Following the 6 core interfaces are described

### 3.1.4 SSH2 integration framework

Hibernate provides solution for a persistence layer, Struts2 supplies the solution to presentation layer, Spring is an integrated framework. Using Spring of IOC containerto managing data access objects, the business logic object and action object dependenciesto combine with the interface programming**.** Itallows objects that are completely decoupled. Spring DAO support can simplify Hibernate data access, use Spring transaction management without application server, you can use declarative transaction.



## 3.2 Front End

### 3.2.1 JQuery

Following the prototype then JQuery is an excellent JavaScript framework. It is lightweight JavaScript library (only compressed into 21k), compatible with CSS3, and various browsers (IE 6.0 + FF 1.5 + Safari 2.0 + Opera 9.0 +). JQuery enable users easily to deal with HTML documents, events, to achieve animation effects, and provides AJAX interaction. Advantage it's documentation is very complete, and various applications are very detailed, but there are various plugins to choose from. JQuery HTML page enables users to maintain separate code and html content. In other words, do not need to insert bunch of js inside html to invoke a command just defines id.

### 3.2.2 Ajax

AJAX refers to asynchronous of JavaScript and XML.The core of AJAX is XmlHttpRequest of JavaScript object. The object was first introduced in Internet Explorer 5, in brief, XmlHttpRequest use JavaScript to request and process the response to the server, but does not block the user. It provides a way to create better and faster, and more interactive Web applications technology.

Traditional web applications allow users to fill out a form , when the form is submitted it sends a request to a web server . The server receives and processes the form, and then returns a new page. This approach waste both times and usage, because majority of HTML codes often is same in pages. Because each interaction application needs to send requests to servers, application response time depends on the server's response time. This leads to respond much slower than local applications of the user interface. In contrast, AJAX applications can only sent to server and retrieves the required data, it uses SOAP or some other XML based Web service interface, and client-side JavaScript handling the response from the server. A significant reduction in data exchanged between the browser and the server, and we can see a more responsive application. While a lot of work can be done on the client machine that made the request, it also reduces Web server processing time. The biggest advantage to using AJAX is able to maintain data without updating the whole page. This makes Web applications more quickly respond to user actions, and avoids sending information that has not changed over the network. AJAX does not require any browser plugins, but requires the user to allow JavaScript executed on the browser.

### 3.2.3 HTML

HTML (Hypertext Markup Language) is the set of markup symbols or codes inserted in a file intended for display on a World Wide Web browser page. The markup indicates the Web browser how to display a Web page's words and images for the user. Each individual markup code is referred to as an element. Some elements come in pairs that indicate when some display effect is to begin and when it is to end. In other words, the text on a web page is “marked up” with these codes to tell the browser how to display the text.

### 3.2.4 CSS

CSS is the language for describing the presentation of Web pages, including colors, layout, and fonts. It allows one to adapt the presentation to different types of devices, such as large screens, small screens, or printers. CSS is independent of HTML and can be used with any XML-based markup language. The separation of HTML from CSS makes it easier to maintain sites, share style sheets across pages, and tailor pages to different environments. This is referred to as the separation of structure (or: content) from presentation. (w3.org)

### 3.2.7 JSP

JavaServer Pages (JSP) technology provides a simplified, fast way to create dynamic web content. JSP technology enables rapid development of web-based applications that are server- and platform-independent. The JSP Standard Tag Library (JSTL) is a collection of tag libraries that implement general-purpose functionality common to many Web applications. (oracle.com/technetwork/java/javaee/jsp/index.html)

**Technologies used in JSP are OGNL and JSTL:**

OGNL stands for Object-Graph Navigation Language; it is an expression language for getting and setting properties of Java objects, plus other extras such as list projection and selection and lambda expressions. You use the same expression for both getting and setting the value of a property. ([commons.apache.org/proper/commons-ognl/](http://commons.apache.org/proper/commons-ognl/))

The JavaServer Pages Standard Tag Library (JSTL) is a collection of useful JSP tags which encapsulates core functionality common to many JSP applications.

JSTL has support for common, structural tasks such as iteration and conditionals, tags for manipulating XML documents, internationalization tags, and SQL tags. It also provides a framework for integrating existing custom tags with JSTL tags.( [tutorialspoint.com/jsp/jsp\_standard\_tag\_library.htm](http://www.tutorialspoint.com/jsp/jsp_standard_tag_library.htm))

I included both JSP and OGNL in my project to express my knowledge of these two technologies, as I understand both SJP and OGNL is for getter and setter variable for action, OGNL is more power widely used, but I found syntax of JSTL is easier to read and code.

### 3.2.7 Bootstrap

Bootstrap is a most popular UI framework for developing responsive, mobile projects on web. Bootstrap’s CSS files are provided in LESS which makes it very easy to customize if you already use LESS for CSS pre-processing. Bootstrap is easy to started, simply download the bootstrap zip from its site, then include it in heading of HTML document. Bootstrap is based styling for most HTML element that can be styled and enhanced with extensible classes. Bootstrap bundle with JavaScript plugin, such as component with drop down menu may interactive that bundled with bootstrap package. I used Photoshop to cut out the images by shape then linked it to JSP page. I used Bootstrap for my front end designs that provides pages with neat and clean layout.

### 3.2.8 Google Map plugin

The maps displayed through the Google Maps API contain UI elements to allow user interaction with the map. These elements are known as controls and you can include variations of these controls in your Google Maps API application. Alternatively, you can do nothing and let the Google Maps API handle all control behavior. Most of Google Map API is free of charge expect that if you generate high amount of traffic. (developers.google.com/maps/documentation/javascript/tutorial)

## 3.3 Choice of Technologies for Recruitment Solution

I spent a lot of time researching different technologies before deciding which would best suit my web application. There were so many options that available for the different components of my projects so I had to analyze pros and cons of each technology available to me. I will outline the list of technologies I had short-list for my project and explain the choice of my final decisions.

The initial choice I faced was which programming language to apply for project. Through four years Business Computing course I have learnt many programming languages as C#, C, Ruby and Java. As Java is our main focus over 4 years. I have always favoured the Java programming language, however I have took other programming language such as Ruby and C# into consideration, but in the end I decided to take java as main language to use as it is my strongest language with deep understanding and there is wide arrange of materials available which I felt that I would benefit from. I also want to use my Java as my main language for my career development in future.

### 3.3.1 Database Options

MYSQL: is world second most used open-source relational database management system. It derived from structured query language (SQL), which is used for inserting, deleting, and updating data in the database. MYSQL is suitable with many different applications. It is especially useful for websites that must access information from a database. It is a fully integrated transaction-safe, ACID compliant database with full commit, rollback, crash recovery and row level locking capabilities. MySQL provides scalability, ease of use. Some well know website such as Facebook, Google, Tickemaster, and eBay uses MYSQL for their business critical applications.

**SQLite:** is an embedded SQL database engine, different with other SQL databases SQLite dose not have separent server process. SQLite reads and writes directly to ordinary disk files. A complete SQL database with multiple tables, triggers, indices, and views is contained in a single disk file. The format for databse file is cross-platform. SQLite is lacking in some features due to small size of SQLite and is not very scalale. Nevertheless, performance is quite good even in low memory environment.

**Microsoft Access:** is known as Microsoft office access, database management system from Microsft that combines relational Microsoft Jet Database Engine with a graphical user interface and software-development tools. It storing data in own format based on the Access Jet Database Engine. It import or link directly to data stored in applications and databases. It is perfect match for relatively small database and a limited number of users.

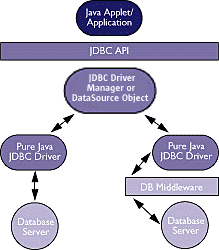
For my choice I decided to use with MYSQL I understand it is best for my project. My research indicates that it’s secure, MYSQL contains solid data security layers that protect sensitive data from intruders. Rights can be set to allow some or all privileges to individuals. Passwords are encrypted. It supports several development interfaces, such as JDBC, ODBC, and scripting (PHP and Perl), allow me to create database solutions that run not only in your NetWare 6.5 environment, but across all major platforms, such as Linux, UNIX, and Windows. It's fast, In the interest of speed, MySQL designers made the decision to offer fewer features than other major database competitors, such as Sybase\* and Oracle\*. MySQL provides all of the features required by most database developers. It manages memory well, MySQL server has been thoroughly tested to prevent memory leaks to enhance security.(http://www.novell.com)

### 3.3.2 Database Connectivity Options

**JDBC(Java Database Connectivity):**

**JDBC:** The Java Database Connectivity (JDBC) API is the industry standard for database-independent connectivity between the Java programming language and a wide range of databases SQL databases and other tabular data sources, such as spreadsheets or flat files. The JDBC API provides a call-level API for SQL-based database access.

JDBC technology allows to using the Java programming language is flexible and efficient have capabilities for applications that require access to enterprise data. With a JDBC technology-enabled driver, you can connect all corporate data even in a heterogeneous environment. (oracle.com)



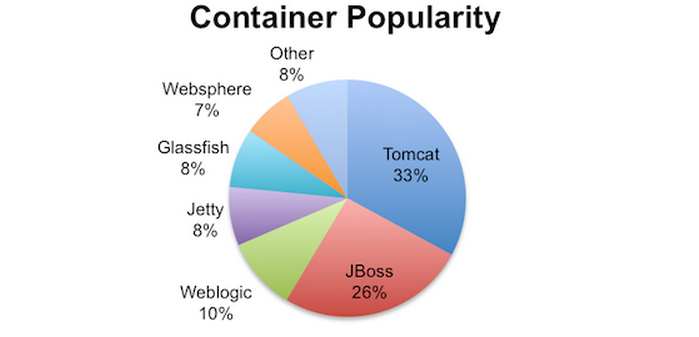
**Hibernate:** object-oriented mapping library for programs written in java based language it provides a framework for mapping java classes to database table. It also provides data query and retrieval facilities. It create SQL call and trys to relieve the developer from manual result set handling and object conversion to keeps the application portable to all supported SQL database.

My choice is decide to use JDBC as database connection. A pure JDBC technology-based driver does not require special installation. The JDBC API simplified a way to identify and connect to a data source by a DataSource object. This makes code even more portable and easier to maintain. Hibernate took up more space than JDBC as require xml files it generates to connect to java class.

### 3.3.3 Web Server Optons

**Apache Tomcat:** Apache Tomcat is the most popular java application server in the world. It is an application server by the Apache Software Foundation used to execute Java servlets and renders Web pages that include Java Server Page coding. It describe "reference implementation" of the Java Servlet and the Java Server Page specifications, the result of Tomcat is an open collaboration of developers that available in both binary and source versions from the Apache Web site. Tomcat product can be used as its own with its own internal Web server or together with other Web servers, including Enterprise Server, Apache, Netscape, Microsoft Internet Information Server (IIS), and Microsoft Personal Web Server. It requires a Java Runtime Enterprise Environment that conforms to JRE 1.1 or later.

**Jetty Server:** Jetty is an open source servlet container, it provides Java-based web content such as servlets and JSPs plus support for SPDY, WebSocket, OSGi, JMX, JNDI, JAAS and many other integrations. These components are open source and available for commercial use and distribution. Jetty is based on Java and its API is available as a set of JARs. Developers can instantiate a Jetty container as an object, instantly adding network and web connectivity to a stand-alone Java app. Jetty is used in a wide variety of projects and products, both in development and production.



From above two technologies identified pros and cons with using two servers. From my research tomcat is easier to embed than jetty, implements the Servlet 3.0 and JSP-EL 2.2 support. It is easy to integrate with other application as I will use it with Spring. It is faster with JSP parsing. Tomcat is flexible and extensible, enterprise scalable. I am will use MyEclipse as my developer tool that self installed with Tomcat. I see lots of posted problems with Jetty with jsp pages. My final decision is to choose Tomcat as my web server.

### 3.3.4 SMTP Plugin

SMTP (Simple Mail Transfer Protocol), It's a set of communication guidelines enable software to transmit email over the Internet. Majority of email software is designed to use SMTP for communication purposes when sending email, and it only works for outgoing messages. Once, set up their email programs, the address of their Internet service provider's SMTP server required for outgoing mail. There are two other protocols - POP3 and IMAP - that are used for retrieving and storing email. The most advantage of SMTP are reliability and simplicity. It's easy to set up software that uses the SMTP communication rules. Messages either get to a recipient, if failed, there is an error message sent back explains the errors.

### 3.3.5 RMI

Java Remote Method Invocation (Java RMI) enables the programmer to create distributed Java technology-based to Java technology-based applications, in which the methods of remote Java objects can be invoked from other Java virtual machines, possibly on different hosts. RMI uses object serialization to marshal and unmarshal parameters and does not truncate types, supporting true object-oriented polymorphism. (oracle.com)

### 3.3.6 GIT

Git is a free and open source distributed version control system used to handle everything from variety size of projects with speed and efficiency. Git is easy to learn and has a tiny footprint with lightning fast performance. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows. I used GIT in my project to track progress and backup of my project. I also want to gain more pratical experiences on version control. (git-scm.com/)

# Chapter 4 Implementation

Welcome page is http://localhost:8080/RecruitmentSolution that will redirect user to <http://localhost:8080/rs/job!first.action>. This is specified in RecruitmentSolution/WebRoot/index.jsp sent reponse to job!first.action by Servlet.

## 4.1 Environment Set Up

### 4.1.1 AOP(Aspect-oriented programming)

It declared in beans.xml, transaction

<aop:config>

<aop:pointcut id=*"bussinessService"*

expression=*"execution(public \* recruitment.service.\*.\*(..))"* />

<aop:advisor pointcut-ref=*"bussinessService"*

advice-ref=*"txAdvice"* />

</aop:config>

## 4.1.2 Spring control all action

The function of Action is when the user clicks on some connection or the button, needs to transfer the Action corresponding method which I establish, but in these methods transfers Action in the level service object handle by business logic.

Under directory of WebRoot/WEB-INF, defines filter that all actions should not be filtered by Struts2, but handled by Spring.

<filter>

<filter-name>struts2</filter-name>

<filter-class>org.apache.struts2.dispatcher.ng.filter.StrutsPrepareAndExecuteFilter</filter-class>

</filter>

### Struts.xml

In struts.xml, I specified operation in action level, e.g <action name=*"js"* this is a variable name as like a tag that called by JSP page, result name is for invoking the appropriate method for the return value, then struts see the return value jump into corresponding page.

<action name=*"js"* class=*"js"*>

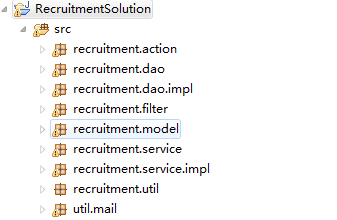
<result name=*"success"*>/success.jsp</result>

<result name=*"fail"*>/fail.jsp</result>

</action>

## 4.2 Project Structure

Recruitment solution project is based on integrated Spring, Struts2, Hibernate (SSH2).



### DAO Layer

Entity/model has been generated automatically by Hibernate, Dao layer, Action layer, Service layer we need to write code myself. Dao layer is made up of two parts, that is, interfaces and implementations, all implementation written in the Dao.impl package. All the interfaces written in Dao Dao layer package, its structure is shown in the following figure:



Dao layer is SSH2 more underlying elements in the framework, all database queries is implemented in Dao layer, entity can be considered to be returned from the query results in Mysql entity layer, which is for each item and correspond to a table in the database, Dao layer returns the result to entity level.

### Service Layer

Service Layer and Dao level code is similar, it is also required to write Interface and implementation. Service Layer is mainly responsible for dealing with business logic. This is a proxy that is responsible for connecting Action layer and Dao layer, if directly provides the Dao level to the Action level the content, this way of doing it is not secure, and SSH2 framework also does not allow you to do so.

**4.2.2 Inversion of Control**

All business logics coded in recruitment.dao implemented by recruitment.dao.impl package in this level, I injected HibernateTemplate into DAO class used by Hibernate annotation @Resource.

@Resource

**public** **void** setHibernateTemplate(HibernateTemplate hibernateTemplate) {

**this**.hibernateTemplate = hibernateTemplate;

}

For service level, I defined interfaces in recruitment.service implemented by recruitment.service.impl , I injected DAO level into service class by declare variable, then getter and setter. I injected DAO into service level by Hibernate annotation @Resource.

@Resource

**public** **void** setCvDao(CVDao cvDao) {

**this**.cvDao = cvDao;

}

Therefore all the methods that I have in DAO can be used in service level.

## JSP

<%  
String path = request.getContextPath();  
String basePath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/";  
%>

The above code included in JSP page is to simplify directory, so that I only need to link <%basPath%> with absolute directory instead of relative path.

### OGNL(Object Graph Navigation Library)

<%@ taglib prefix="s" uri="/struts-tags" %>

This is Struts2 tag included in JSP page for <s:property> get value from action

### JSTL(JavaServer Pages Standard Tag Library)

The format of JSTL is ${j.job.jobId} to get value from Action class. I found this format is easier to read without import struts tag each time in JSP page and easier when I track syntax error caused by miss type single or double quotes for OGNL. That’s the reason I have both JSTL and OGNL included in my project.

## 4.3 Registration

For Jobseeker’s registration URL is <http://localhost:8081/RecruitmentSolution/js!registerJs.action>

For Employer’s registration URL is <http://localhost:8081/RecruitmentSolution/addEmp.jsp>

Registration form with valid data submitted by user, hibernateTemplate(HQL) will save user into database. Send welcome notification email to Employer and Jobseeker’s email address by SMTP email. This is coded in SMTP mail with MailSenderInfo.java implemented in both EmployerAction and JobSeekerAction.

### 4.3.1 JavaScript, JQuery and AJAX

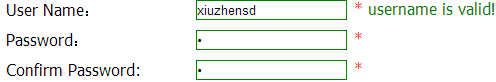
This is implemented by AJAX, registerJs is a method in JobseekerAction class, this is for loading JobCategory when open Jobseeker registration page. When jobseeker chose a job category that will load jobs in that job category, then chose a job and load skills under that job. Use of AJAX will help user have better experience as only reload required column of form.

Also for registration form, I also have client side validations, when user’s mouse move away from input box, will call JQuery of blur() method to check to see if the user is entering valid data, if it is, red dot(\*) beside relevant input boxes will change to green font of texts indicates validate data,

E.g



also has green color highlight line of input box. If data entered is invalid that will highlight line of input box as red, and change red dot(\*) beside that input box with invalid message appears in red font.



For Employer and Jobseeker name and email has to be unique, this is implemented by AJAX to post following URL to check if that exist in database.

jsCheckUsername.action?js.username=" + username;

"jsCheckEmail.action?js.email=" + email;

I did this is validation of form allow user to check if data entered is correct before fill next one that provide better user experience.

All of these required data should be filled before enable submit button. Form validation for Jobseeker and Employer is under directory WebRoot/js/jsValidation.js

Note: There is no registration for Administrator, because in business sense, administrator can not open to public.

## 4.4 Authentication

In this project, currently are three users, administrator, jobseeker and employer. For Jobseeker and Employer login by enter email and password, Administrator authentications by entering username and password. As for administrator, cannot be registered by user, username and password has given by developer. Authentication is implemented by HQL to check to see if email/username or password is match with database, it is return this user object. If authenticated, return success, failed, direct to fail page.

If Jobseeker authentication success, will direct to the page where shows all jobs that relevant to this Jobseeker’s skills that specified in registration. I implemented this in Jobseeker action, load Jobseeker’s object from session then get all the Jobseeker’s skills from Jobseeker skills, get all jobs matches with Jobseeker’s skills.

If Employer authentication success, return to the page list all Jobseeker’s that skills match jobs skills that posted by Employer. If there is no job or posted job’s skills not match with Jobseeker’s skills, returned page show empty.

If Administrator login success, return to administrator page.

4.4.1 RMI and Session

When Jobseeker, Administrator and Employer login in to this site stores object instance in session.

I used RMI and session to bind Jobseeker object instance to the name ‘user’,

ServletActionContext.*getRequest*().getSession().setAttribute("user", js);

Bind Administrator object with name of ‘admin’ and Employer object instance with name in ‘employer’.

ServletActionContext.*getRequest*().getSession().setAttribute("admin", admin);

ServletActionContext.*getRequest*().getSession().setAttribute("emp\_user", emp);

I implemented binding processes when Jobseeker, Employer or Administrator login.

Session is removed when user close browser.

## 4.5 List and Search

For search Job with one getJobs()n JobDaoImpl to show all jobs by HQL.

check and append condition for skills, job category, category, job, employer, title, country, county, area, skills in order to list all jobs. Use if condition to append SQL statement for search purpose. If user want to search job category, then select job category in page, then enter submit, will go to if statement to append job category id, if user enter another condition, e.g. country, then will append job with this job category within the country. It used in action, declare a variable type of List, then assign the jobs returned in getJobs(), then display in in JSP page.

All searches and lists data are done in this way with one method. This is convenient way of searching and listing data in one method. In future, it simplified modifying code purpose.

### 4.5.1 Select and Select2

I used select tag for search and select 2 for decorations. JavaScript file is under directory of WebRoot/js/area.js and WebRoot/js/jobskill.js. I used select tag to load the skills and areas from AJAX, show data in option tag for user to select.

For search Area, I injected AreaManager in JobAction. When user opens main page load all countries in from Action, when user select country, AJAX post this request "listCountry.action?area.areaId=" + countryId; then load countries by this county id from database. Once I loaded counties, I append in JobAction with county Id and county name as id\_name,id\_name. Such as 0\_Please Select County, 3\_Monaghan, 4\_Donegal, 5\_Sligo, 6\_Meath, 7\_Wicklow, 8\_Wexford, 9\_Kildare, 10\_Clare, 11\_Galway, 15\_Waterford, 31\_Dublin, then I split id and “\_” in function getCountry(c) in area.js. 0\_Please Select County is for once, counties are loaded will show 0\_Please Select County.

When user select county, then load all the areas under this county id, AJAX post request "listDistrict.action?area.areaId=" + districtId; then loads all the areas. Such as

0\_Please Select District, 32\_Dublin City Center, 33\_Dublin 2, 35\_Dublin 4, 36\_Dublin 3, then I split id and “\_” in function getDistrict(d) in area.js

In Action, I appended loaded

## 4.6 Post Job

Employer only can view jobseekers that skills matches with posted job’s skills. If there is no job posted, the Employer’s page will show empty. If posted jobs matches with Jobseeker’s skills will list the Jobseekers in main page.

This is implemented by HQL save(), firstly, in post job page load all job categories, when Employer choose one Job Category will load skills under this Job Category . Then saves skills into JobSkill table, this is middle table. When Employer fill in this form should fill required fields, otherwise form validation will stop Employer to post a job.

## 4.7 Delete

All delete method achieved by HQL, at first, pass this object id into parameter and deleted to check to see if selected Item is existed in database, then. If it is, then will allow user to delete, Here I used hibernateTemplate to do delete operation.

## 4.8 Update

Update is implemented by HQL to get required id in the JSP page when user click Update a href link get id, load object by this id. Use If statement check to if this object exist. If exist, then set data by parameter. I used HibernateTemplate for save purpose.

## 4.9 Upload CV

In order to implement upload CV, in CVDaoImpl class I used HibernateTemplate to save CV into databse. In CV Action class I get JobSeeker’s object from session, then load its Id, upload file into directory of Tomcat webapps\RecruitmentSolution\upload. When I upload, I store CV title into database with append Jobseeker id and “\_” file name. E.g file name is Linda.doc when it uploaded into Database, file name will be append with my Jobseeker’s id 1, so that stored in database is 1\_Linda.doc. This way it can be identified by which Jobseeker’s CV as unique Jobseeker’s id.

## 4.10 Download CV

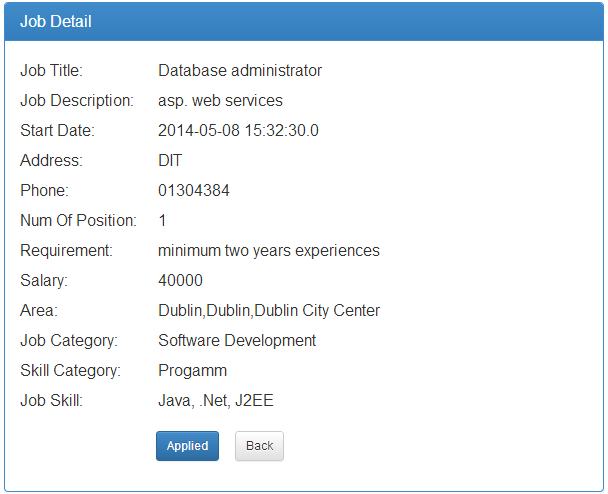
In download CV, when Jobseeker, Employer, Administrator, click a href of CV name, that will allow user to down load CV.

<a href="downLoadCV.action?filename=<s:property value='#c.cvTitle' />">

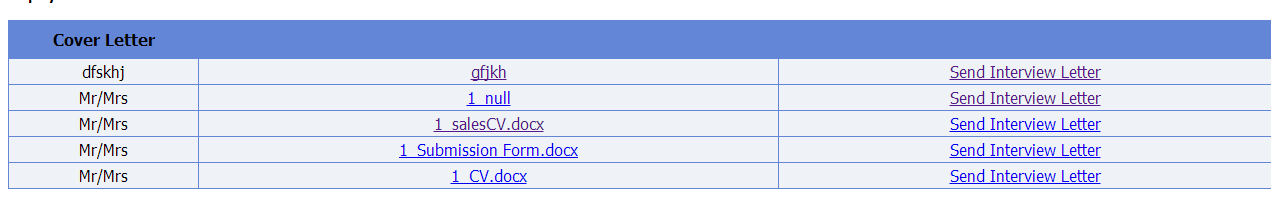
This is calling downLoadCV in CVAction with url getting file name from url, then download it from Tomcat webapps\RecruitmentSolution\upload.

## 4.11 Send Interview

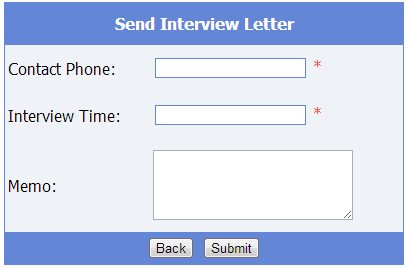
When Employer login and view the jobs that posted, click job detail href is job!detail?job.jobId=60, it load this job detail by get this job id in this page, then click Applied button link to a url address job!listAppliedJs?job.jobId=60, it load who applied from ApplyJob table by job id.



Then show all the jobseekers who applied this job, Employer clicks View a href link url to job!listCVByJsId.action?job.jobId=1&js.jsId=1, Get CV from this jobseeker with job id and Jobseekers id. Employer enable to click a href link to download CV,

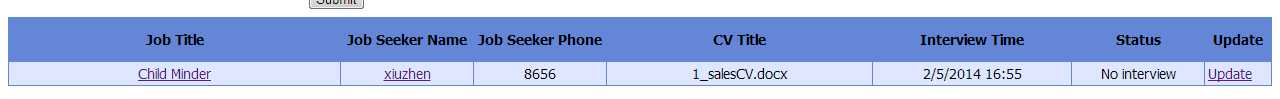


When Employer clicks Send Interview Letter links ir!preSend?ir.job.jobId=1&ir.js.jsId=1&ir.cv.cvId=8, brings page of Send Interview Letter, this is to generate Interview Record table and send email to this Jobseeker’s email address for interviewing, Jobseeker also able to view the history of interviews in profile.



## Update Interview result

After interview login into Recruitment Solution, in Profile [View send Interview Letter](http://localhost:8081/RecruitmentSolution/ir!listSend?employer.empId=6), a url link of ir!listSend?employer.empId=6, shows all send interview records.

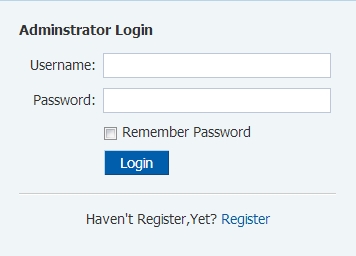


update url of ir!preUpdate?ir.interviewId=5, get this interviewed of interview record, then update interview result with default ‘No Interview’ status is 1, ‘Interview Success’ status is 2, ‘Interview fail ’ status is 0. Once Employer update the interview status, it will also update Jobseeker’s table, change status to 1 as employed, 0 means unemployed. As one job can looking for a number of jobseekers, to determine this by if number of position in job equals to 0, means this job is gone. In order to achieve this Employer updates the interview result to success, number of position in job minus one, change job status to be 0. //改改

## Administrator

Administrator authentication, by enter username and password.

url : <http://localhost:8080/rss/userManager> that will redirect user to <http://localhost:8080/rss/userManager/adminLogin.jsp>. This is specified in RecruitmentSolution/WebRoot/userManager/index/jsp sent response to adminLogin.jsp by Servlet



Administrator has right to access to all database in Recruitment Solution to update, insert, delete and search maintain database.

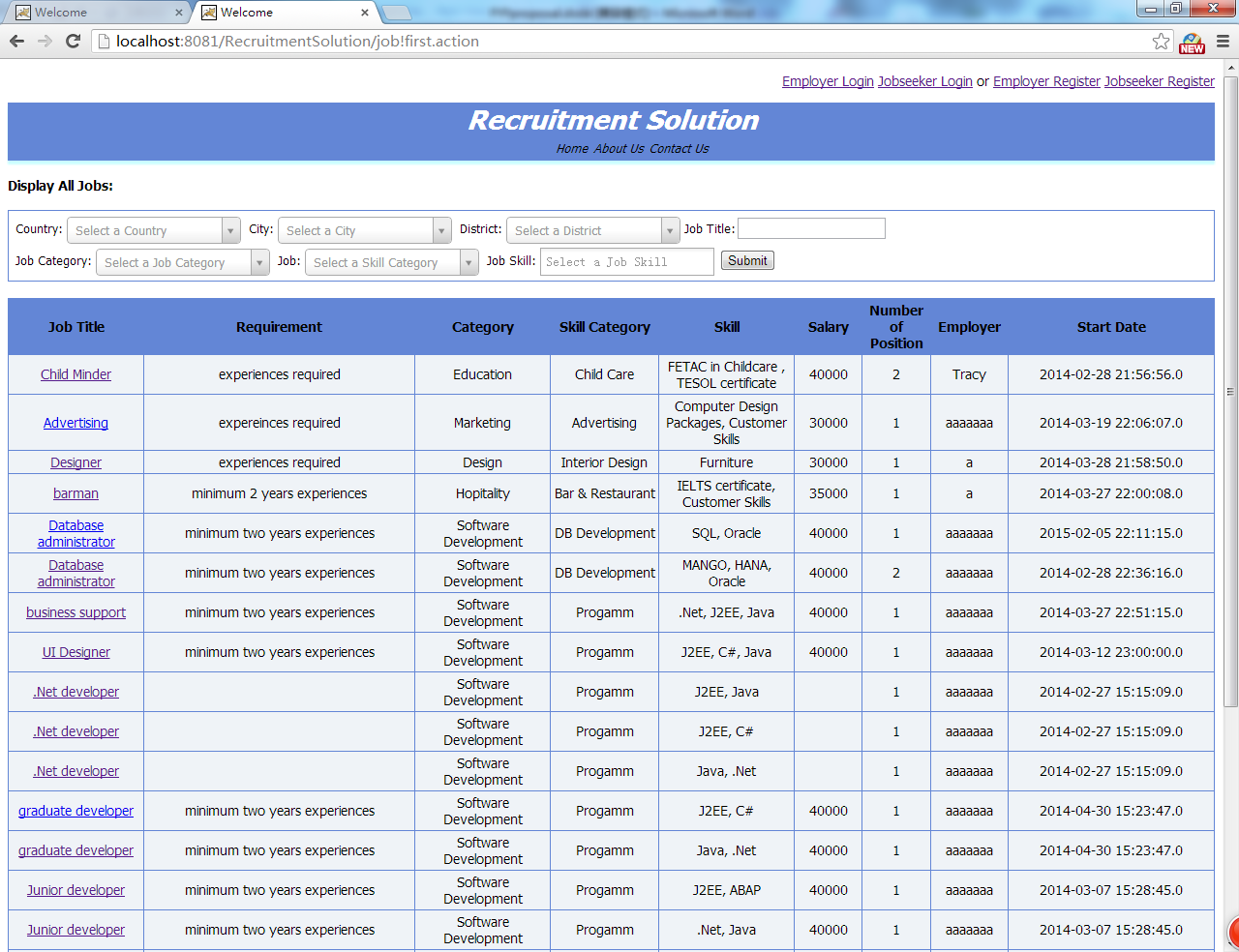
## JFreeChart

Jars required for integration

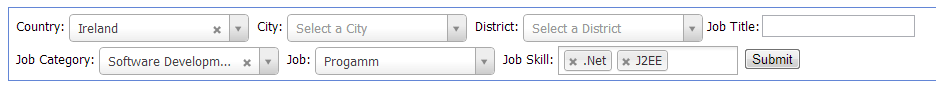
|  |  |  |
| --- | --- | --- |
| Jar name | location | Description |
| antlr-2.7.6.jar | hibernate/lib/required | Analyze HQL |
| aspectjrt | spring/lib/aspectj | AOP |
| aspectjweaver | .. | AOP |
| cglib-nodep-2.1\_3.jar | spring/lib/cglib | Proxy,Binary enchancement |
| common-annotations.jar | spring/lib/j2ee | @Resource |
| commons-collections-3.1.jar | hibernate/lib/required | 集合框架 |
| commons-fileupload-1.2.1.jar | struts/lib | struts |
| commons-io-1.3.2 | struts/lib | struts |
| commons-logging-1.1.1 | Separate download delete1.0.4(struts/lib) | struts  spring |
| dom4j-1.6.1.jar | hibernate/required | Analyze xml |
| ejb3-persistence | hibernate-annotation/lib | @Entity |
| freemarker-2.3.13 | struts/lib | struts |
| hibernate3.jar | hibernate |  |
| hibernate-annotations | hibernate-annotation/ |  |
| hibernate-common-annotations | hibernate-annotation/lib |  |
| javassist-3.9.0.GA.jar | hiberante/lib/required | hibernate |
| jta-1.1.jar | .. | hibernate transaction |
| junit4.5 |  |  |
| mysql- |  |  |
| ognl-2.6.11.jar | struts/lib |  |
| slf4j-api-1.5.8.jar | hibernate/lib/required | hibernate-log |
| slf4j-nop-1.5.8.jar | hibernate/lib/required |  |
| spring.jar | spring/dist |  |
| struts2-core-2.1.6.jar | struts/lib |  |
| xwork-2.1.2.jar | struts/lib | struts2 |
| commons-dbcp | spring/lib/jarkata-commons |  |
| commons-pool.jar | .. |  |
| struts2-spring-plugin-2.1.6.jar | struts/lib |  |
| Servlet.jar |  |  |

# Chapter 5 User Guidelines

## 5.1 Main Page



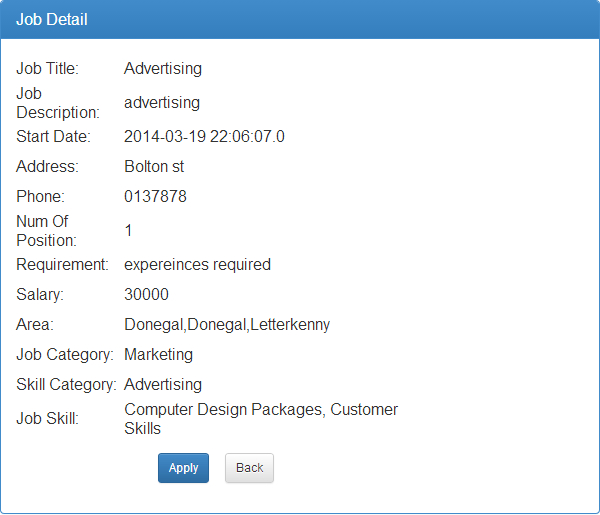
Any user is unlogged to this Recruitment Solution site and able to view all the jobs that posted in this site. With search bar in the top allow user to search jobs by country, city, district, job category, job and skills. User only can choose country first, then able to choose city under this country, after choose county enable to choose district. Job title can be entered by any character for search purpose.



beside option can be close if user change choice. User can search more than one skills.



When User click a href link under Job Title will link to detail of job, but only can be applied by authenticated Jobseeker.



In upper left hand side are href coded with action or JSP page that linked to login page for Employer or Jobseeker, also registration page for employer or jobseeker.

[Employer Login](http://localhost:8081/RecruitmentSolution/loginEmp.jsp) [Jobseeker Login](http://localhost:8081/RecruitmentSolution/loginJs.jsp) or [Employer Register](http://localhost:8081/RecruitmentSolution/addEmp.jsp) [Jobseeker Register](http://localhost:8081/RecruitmentSolution/js!registerJs.action)

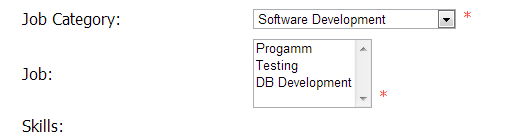
## 5.2 Registration

### 5.2.1 JobSeeker Registration

When user clicks a href link of [Jobseeker Register](http://localhost:8081/RecruitmentSolution/js!registerJs.action) direct to Jobseeker Registration form, in order to complete registration, user required to fill the required form with \* beside the text box. If required form is not entered before move to next text box or not valid data, red error message with indicate beside the box ask user to fill it with valid data. If data entered is valid, green message beside the box indicates valid.



Jobseeker should select Job Category first before select Job, so that the data will load Jobs under this Category. Note, this is important, Jobseeker login only view the jobs matches with skills that you selected in here, but you can modify it in Jobseeker profile.

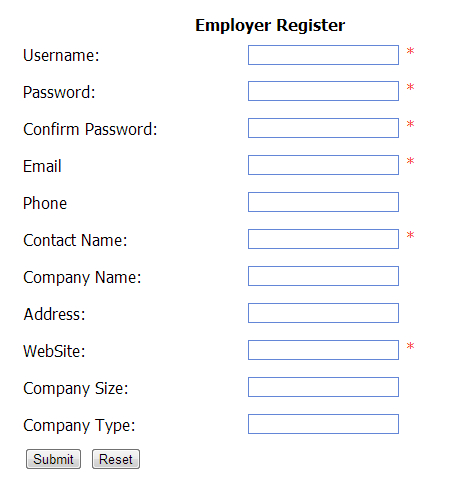


Multiple choice of skills allow Jobseeker to select.



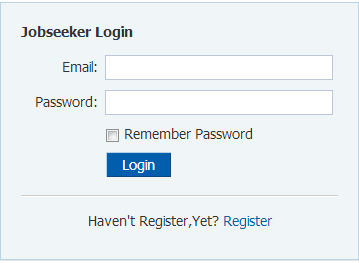
### 5.2.2 Employer Registration

Employer registration is straight forward without any entering valid user name and two match passwords, contact name and website.

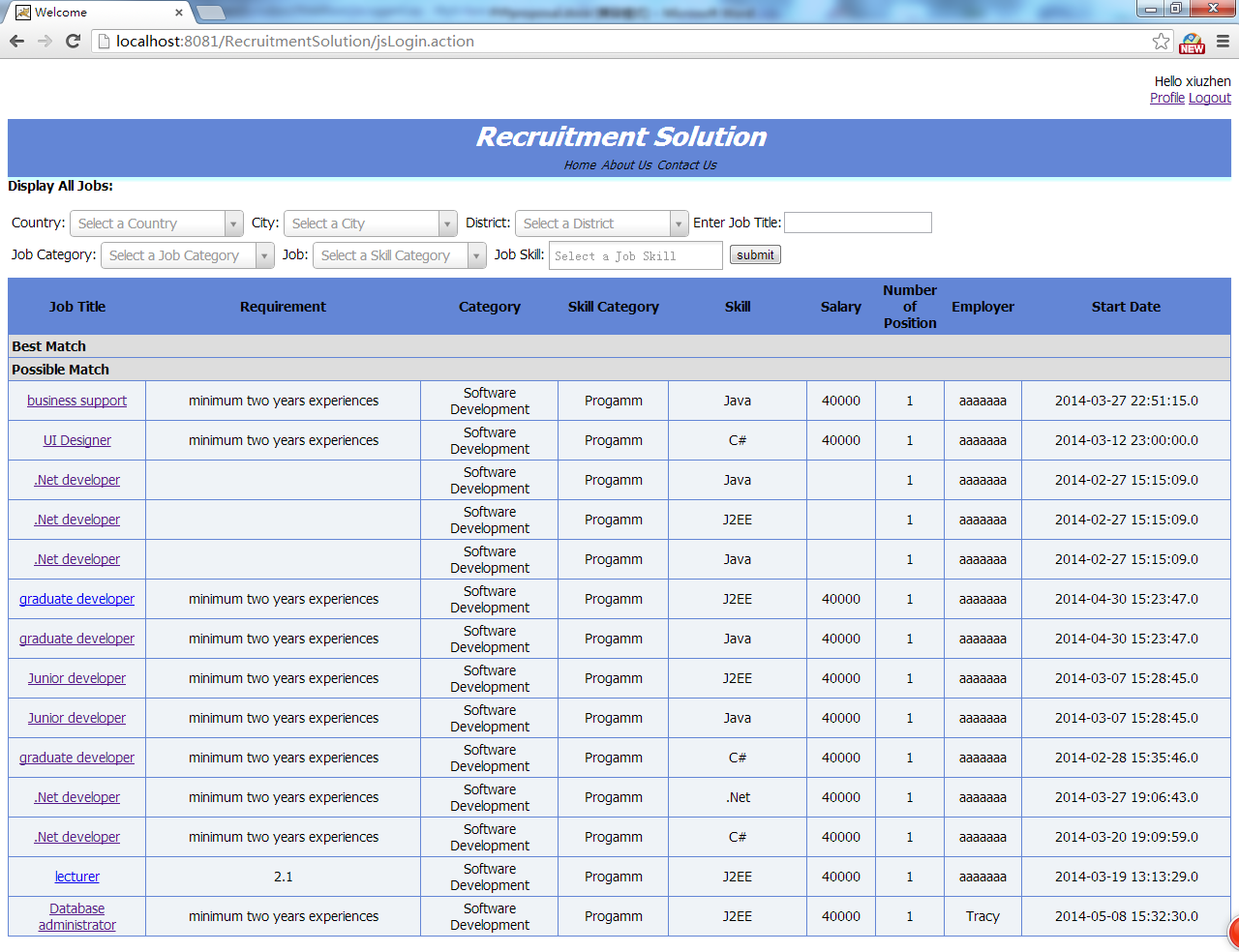


5.3 Jobseeker

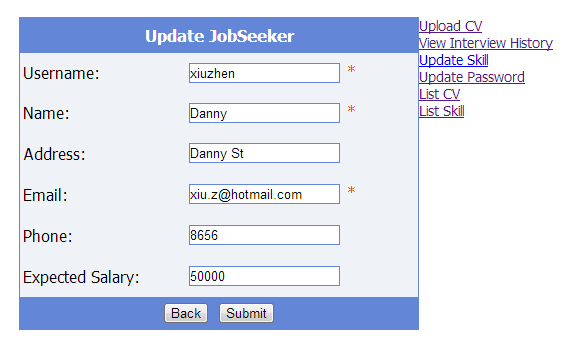
### 5.3.1 Authentication



After completed jobseeker registration direct to Jobseeker login, if authenticated, load jobs matches with Jobseeker previously selected skills in registration. With order of best match and possible match, best match means all the skills exactly matches with jobs. Possible match is only one or less than a number of Job’s skills with Jobseeker’s skills. In same way as unlogged user, Jobseeker can search jobs by click search bar or click a href link under Job Title to view more detail of jobs, but in this case, Jobseeker able to apply Job. If there is no CV uploaded, it will direct user to upload CV, then user can click Job detail again to apply this job.



In upper right hand side, there is a [Profile](http://localhost:8081/RecruitmentSolution/js!get?jsId=1) link that direct Jobseeker to view and update their details by direct modify data and press submit button, then success, will bring back to this page again. More a href links on the right handside of Jobseeker profile that link to [Upload CV](http://localhost:8081/RecruitmentSolution/addCV.jsp) [View Interview History](http://localhost:8081/RecruitmentSolution/ir!listMySend) [Update Skill](javascript:void(0)) [Update Password](http://localhost:8081/RecruitmentSolution/updateJsPassword.jsp) [List CV](http://localhost:8081/RecruitmentSolution/cv!list.action) [List Skill](http://localhost:8081/RecruitmentSolution/jsSkill!list.action)



### Upload CV



Upload CV link to Add CV page, Jobseeker upload CV and enter cover letter.

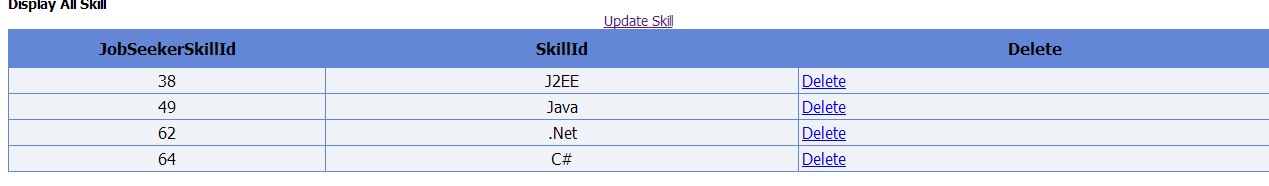
### View Interview History

The status in last column indicates interview status, job title a href link will direct to job detail. If Employer update the status of interview result, the status will also update in interview history page.



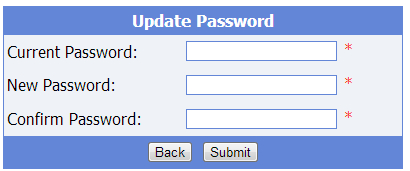
### List Skill

List skill a href shows all skills that Jobseeker selected in registration, Jobseeker can delete skills or click update skill a href direct to update skill page for updating.



### Update Password

In order to update password, current password should match with new password, otherwise, error message will indicates update failure.

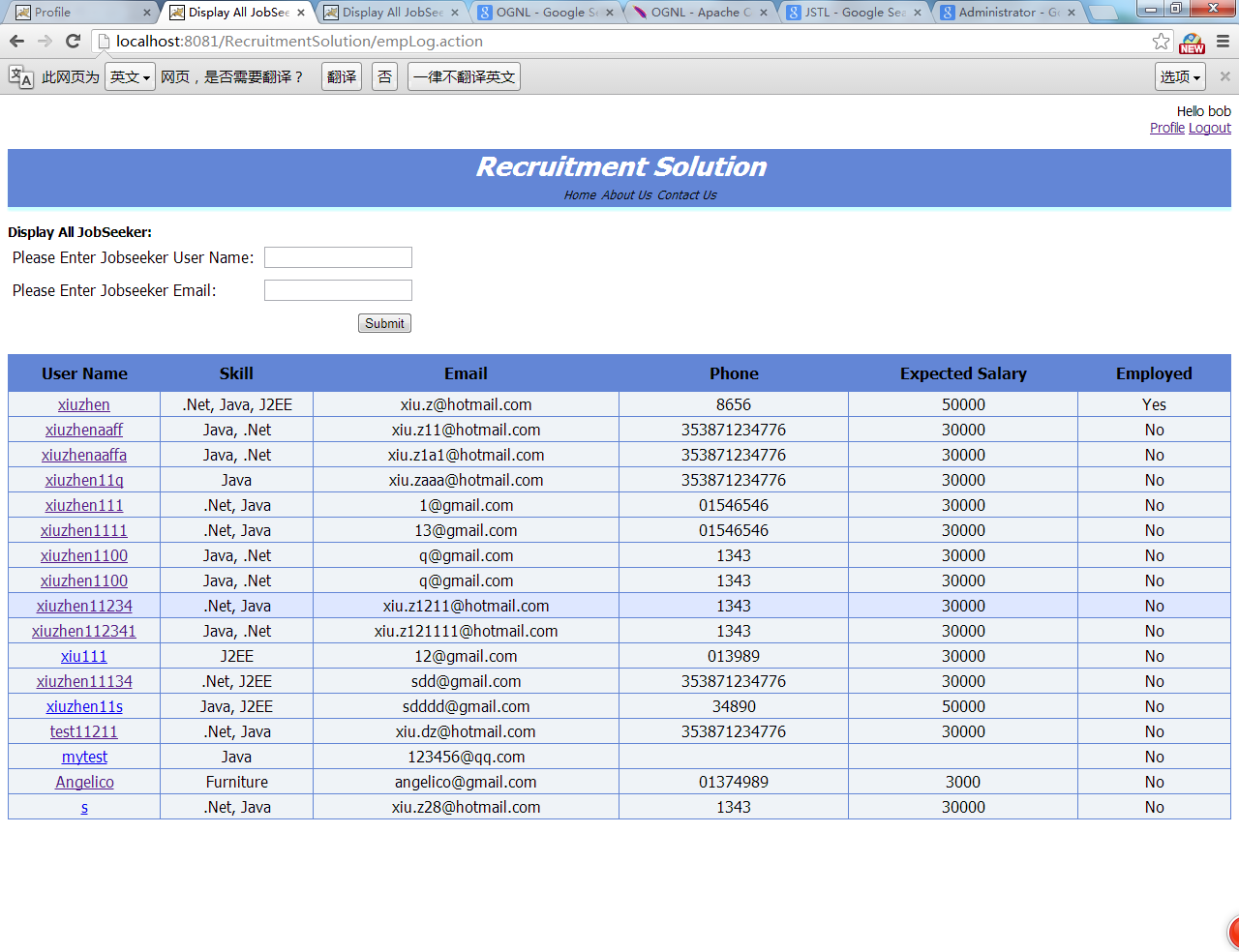


### List CV(还没完成)

## 5. 4 Employer

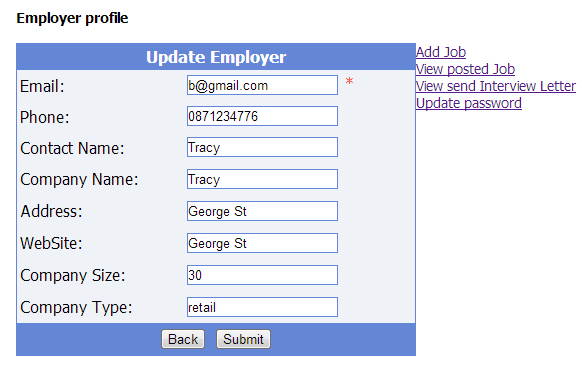
### Authentication

After Employer authenticated, direct to the page list Jobseeker’s skills that matches with posted job’s skills. If there is no posted job by employer or posted job not match with Jobseeker’s skills, then Employer won’t see any Jobseekers.



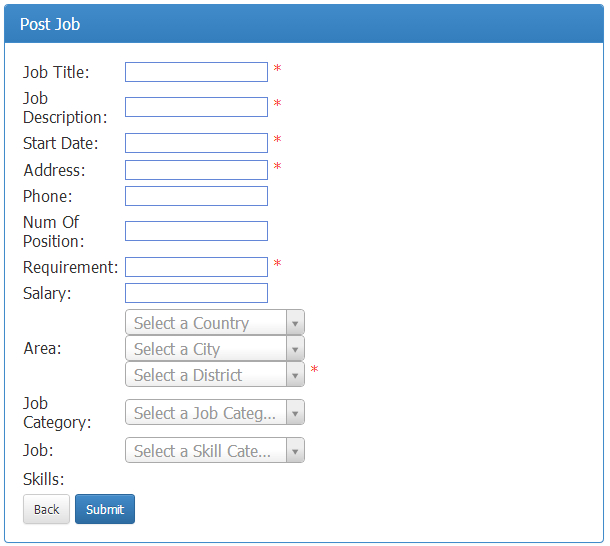
### Profile

Employer’s profile is for update Employer’s profile, as Employer directly modifies the textbox and click submit button, success of modification will return back this profile page. In right hand side of Employer a href link direct Employer to [Add Job](http://localhost:8081/RecruitmentSolution/job!registerJob.action)  [View posted Job](http://localhost:8081/RecruitmentSolution/job!listEmpJob?job.employer.empId=6) [View send Interview Letter](http://localhost:8081/RecruitmentSolution/ir!listSend?employer.empId=6) [Update password](http://localhost:8081/RecruitmentSolution/updateEmpPassword.jsp)  pages.



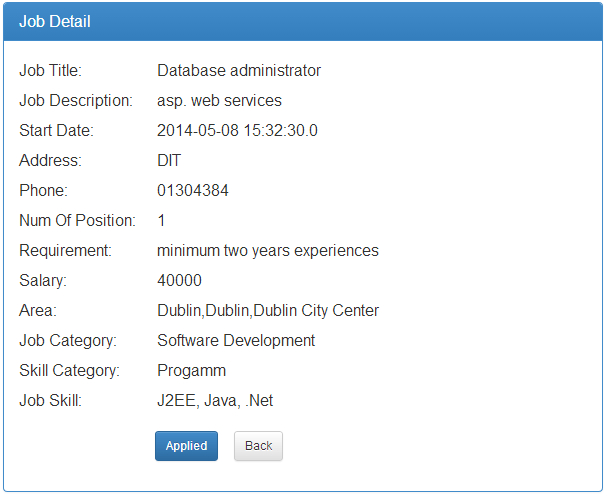
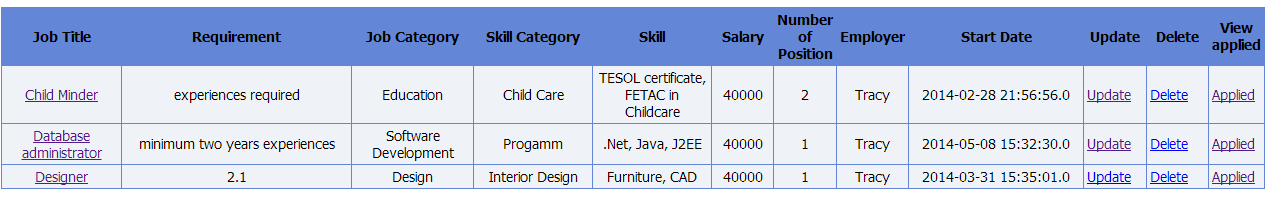
### Post Job,

Employer post job, once this job with skills return to main page, Recruitment Solution will show suitable Jobseekers with matched skills. This post job is designed in same way as Jobseeker registration with client side validation to give quick check of whether data entered is valid. If data entered is invalid, red error message surround text box with red error message beside it. When user select Country load all the Cities under this Country, Finally select Area. Job Category, Job and Skills designed in same way as well.



### View Posted Job

This page list all posted job by Employer, when Employer click a href link under Job Title link to the page of detail job.



This job detail has an Applied Button, if there is a Jobseeker who applied this job will list jobseeker’s detail s,



View a href link r direct to the page shows CV and cover letter of this Jobseeker. Click [1\_CV.docx](http://localhost:8081/RecruitmentSolution/downLoadCV.action?filename=1_CV.docx) will download CV from database. [Send Interview Letter](http://localhost:8081/RecruitmentSolution/ir!preSend?ir.job.jobId=1&ir.js.jsId=1&ir.cv.cvId=8) direct to the form of sending Interview to this Jobseeker .



### Send Interview Letter

Employer fill in contact phone and Interview time to this Jobseeker and click submit button. Email will send to this Jobseeker’s email address with Interview Time and messages by [27248466@qq.com](mailto:27248466@qq.com)

Format of Interview letter is:

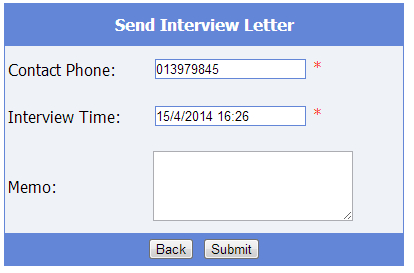
Dear XiuZhen,

I'm very glad to invite you attend our company interview at 23/4/2014 12:46, my phone is 0139798

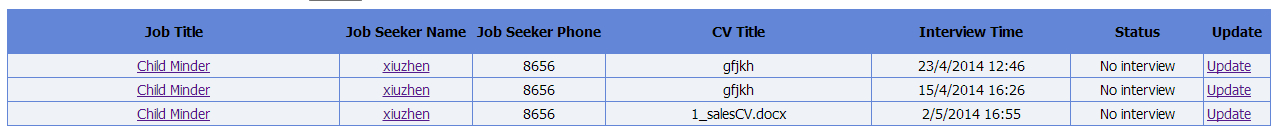
Best Regards

Tracy

Note: This interview record will also generated in Jobseeker’s profile who can find the link of History of Interview.



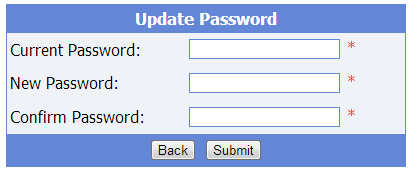
### View Send Interview Letter(改图, 补发email格式)



This page show all the send interview, Update a href link, direct to the page for update interview record, Employer select Interview Result of ‘No Interview’ ‘Interview Success’ and ’Interview Fail’ to indicate status of interview. If success, system change the Jobseeker’s Employed status to ‘Yes’, and send Interview Result Letter to this Jobseeker. If result failed, the Interview Result Letter will send to this Jobseeker’s email to inform the result. It also update the Jobseeker’s History of Interview, when Jobseeker login to site, click profile, under History of Interview will also able to find interview result.



### Update password

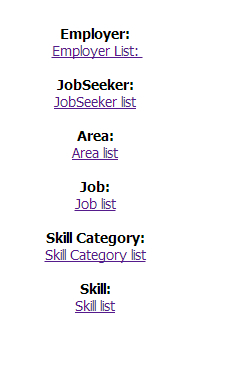


Employer enter current password and two matched new password will direct to update success page. If incorrect data filled in, return to failed page.

## Administrator

Administrator login page url is <http://localhost:8081/RecruitmentSolution/userManager/>

After authentication of this administrator that list a href link of information that stored in database. Administrator has right to update, delete, insert all data stored in database.

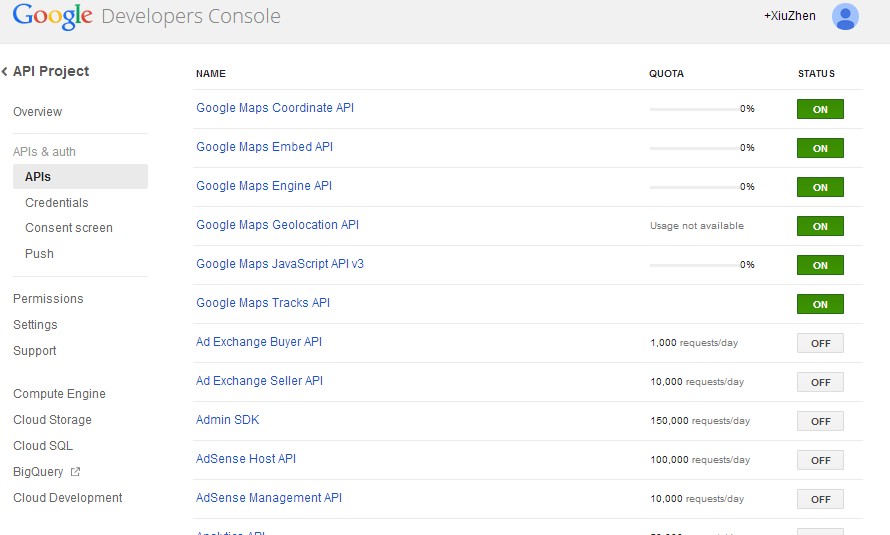


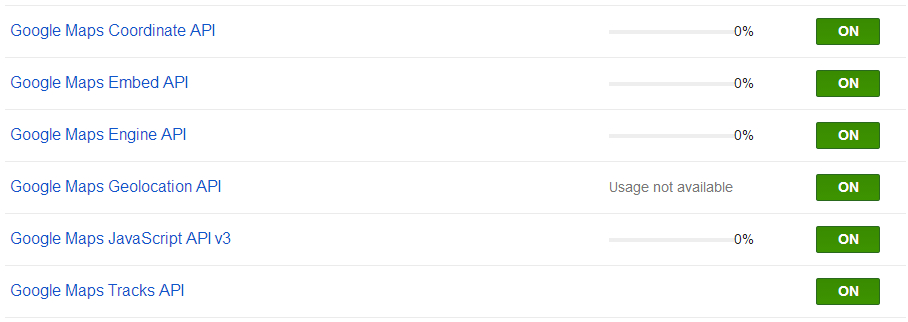
## 5.5 GitHub

I used GitHub to record my progress of my project, also that help me to back up projects in case of accident of lose projects. I download Gitub for Desktop from <https://windows.github.com/> that is easier to synchronized my projects with GitHub server, especially that when I work my project at home with PC and I bring my laptop to college, so that I can simply download my project from GitHub everywhere. I registered my GitHub account with micro plan that allow me to have my project in private repository so that no one else can see it to keep it safe.

## 5.6 Google Map

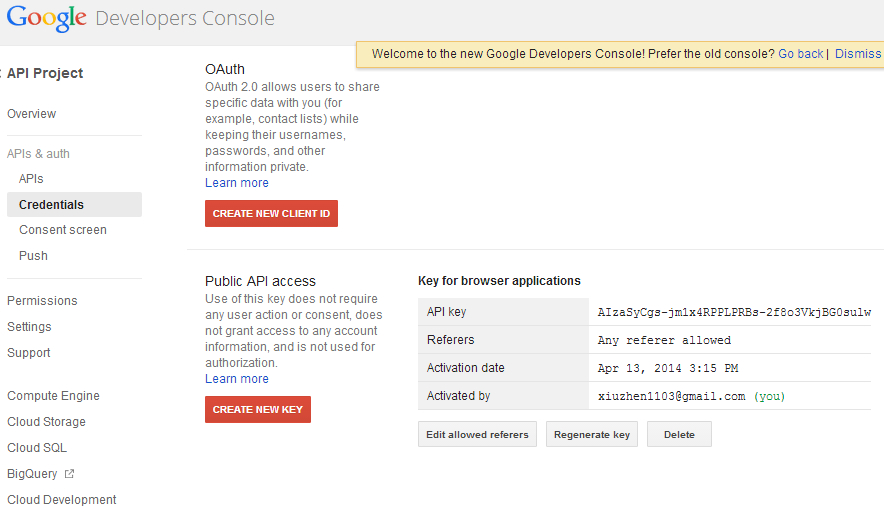
I embedded Google Map into About Page with JavaScript, firstly go to https://code.google.com/apis/console/, and log in with your Google Account. Click API on left hand side and turn on Google Map status.





Click Credentials and save API key AIzaSyCgs-jm1x4RPPLPRBs-2f8o3VkjBG0sulw include it into my page

<script src="http://maps.googleapis.com/maps/api/js?key= AIzaSyCgs-jm1x4RPPLPRBs-2f8o3VkjBG0sulw&sensor=false">  
</script>





# Chapter 6 Installation Manual

MySQLServer and SQLYog Enterperise

Download and install mysql-5.1.exe by run as administrator from <http://dev.mysql.com/downloads/mysql/5.1.html> into top level of the C or D drive.

Download SQLYog10.zip from <https://www.webyog.com/product/sqlyog>

Subscriber：3ddown

Subscription code：7c799fe606582b12

JDK

Download and install Java from http://java.sun.com/javase/downloads/. I use JDK 1.6.0\_13, but any Java 6 or 7 version will work.

Tomcat

Download Tomcat. Unzip 6.0.37 32-bit Windows into the top level of the C or D drive.

* Right click "Computer" or "My Computer" in the Start Menu and choose "Properties" from the context menu. Under the "Advanced" tab, you can define new variables.
* Set up environment with set the JAVA\_HOME variable to point to your JDK's main directory. Additionally, add %JAVA\_HOME%\jre\bin to the PATH variable,

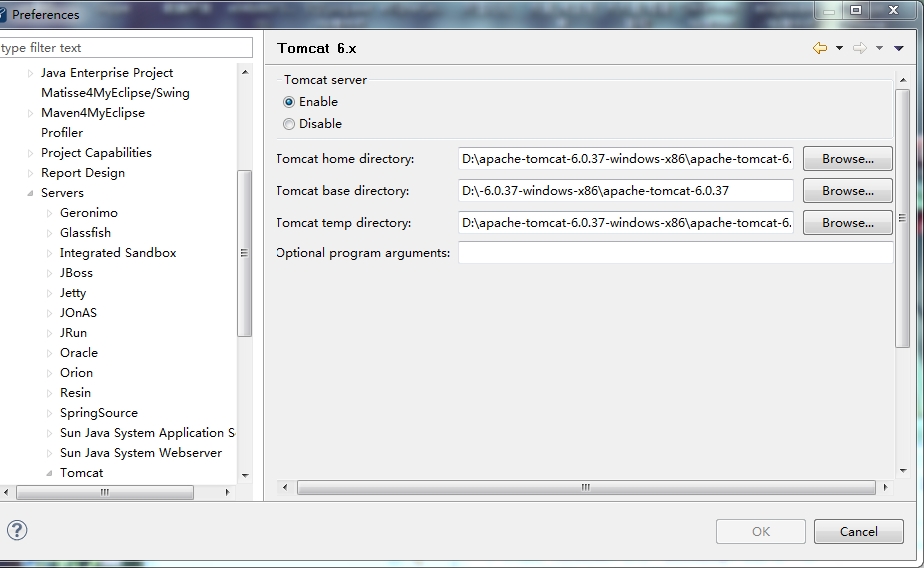
MyEclipse

Install MyEclipse 8.5 Download from http://myeclipseide.com/ and run installer.

Subscriber：huahua

Subscription Code：uLR8ZO-655444-69678657696224504

Tell MyEclipse about Tomcat. Select Window, Preferences, MyEclipse, Servers, Tomcat, Tomcat 6. Click "Enable". Select "Tomcat Home Directory" and hit "Browse". Navigate to the Tomcat installation directory Click Apply & OK.

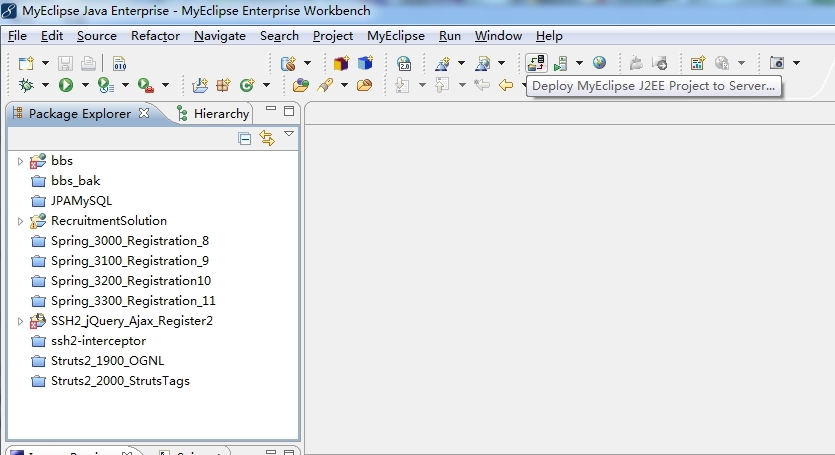


Also click on the JDK sub-entry under Tomcat and make sure this matches the main Java version you are using. In particular, if you are using Java 7 but the Tomcat JDK refers to Java 6, you will get illegal class file errors when executing

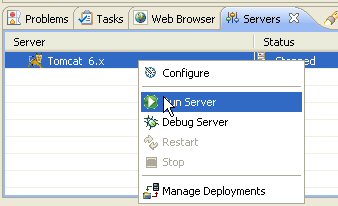
servlets.



Import my project RecuitmentSolution into MyEclipse, then deploy MyEclipse J2EE project to server.



Run Tomcat. Click on Servers tab at bottom. R-click on Tomcat 6, choose "Run Server". Open http://localhost:8080/RecruitmentSolution/job!first.action in a browser.



Directory showing site address, be aware of case sensitive

Welcome page

* **http://localhost:8080/RecruitmentSolution**

Administrator login page 

* **http://localhost:8080/ RecruitmentSolution/userManager**

# Conclusion

Overall the project was very challenging experience throughout. Having to learn how to develop new features or functionalities was a constant demand for the web application.

The first challenge of the project was that I had to learn Spring frame work which was new to me, this first major stepping stone of project that I had to overcome. For these combinations of framework of Struts2, Spring and Hibernate. In first week of giving the final year project, I revised struts that we studied last year built a simple project with user and administrator registration, update, delete, insert. Then I added Hibernate into my web application, to map relationship between each entity. Lastly, I watched tutorial and search resources on Spring, I added Spring into my project which is difficult part, I needed to remove all the transaction with connectivity previously used by Hibernate, changed it to Inversion of Control manage the transaction. Then figure out how to do with configuration.

Second challenge was this project needed to continually update all the time, as I achieved one function may have affected on previously function. Firstly, I decided to when jobseeker login into system only see jobs in job category that interested to this jobseeker. Then when I added skill table for jobseeker and job, this function need to be changed, I needed to match multiple jobseeker’s skills to match job’s skills. Thus, the times that I spent two days on compile match job and jobseeker with job category was wasted. Then in area entity, firstly, I designed only have one filed for area, then I changed that to 3 levels with county, district, area. More modification required, as I have to change the jsp pages that I created with all attributes. Especially in the job, I have done function on search Job by area, then I have to change the code for job. After second check point, I realize there was better way to do the search by loading data from database allows user to search job, I used Ajax.

Third biggest challenge would have to be a large number of entities, in this project I have13 tables. I needed to manage relationship between each tables and huge database needed to be take care of.

# Reference

[http://www.oracle.com](http://www.oracle.com/)

<http://www.w3schools.com/>

Tutorial: Using Tomcat 6 with MyEclipse

<http://www.coreservlets.com/Apache-Tomcat-Tutorial/myeclipse.html>

myEclipse subscription code

<http://www.csdn123.com/html/blogs/20130813/52211.htm>

<http://thenounproject.com/>

http://www.cssbuttongenerator.com/

<https://developers.google.com/maps/documentation/javascript/tutorial>

https://windows.github.com/