# 

# B.Sc. 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. Business Computing**

**Dublin Institute of Technology**

**Year 4**

**Supervised by – Mary Regan**

**May 2014**

**Acknowledgements**

Throughout 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: 1st May**

Table of Contents

[B.Sc. Business Computing 1](#_Toc386453688)

[Chapter 1 Introduction 7](#_Toc386453689)

[Project Overview 7](#_Toc386453690)

[Project Plan 7](#_Toc386453691)

[Introduction 8](#_Toc386453692)

[Objectives of the recruitment solution 9](#_Toc386453693)

[Business Case 9](#_Toc386453694)

[Chapter 2: Requirements capture and analysis 11](#_Toc386453695)

[2.1 Actors 11](#_Toc386453696)

[2.2 Requirement analysis: 11](#_Toc386453697)

[Chapter 3: Design 14](#_Toc386453698)

[System Architecture 14](#_Toc386453699)

[3.1 SSH2 framework 15](#_Toc386453700)

[3.1.1 Struts2 15](#_Toc386453701)

[3.1.2 Spring 16](#_Toc386453702)

[3.1.3 Hibernate 17](#_Toc386453703)

[3.1.4 SSH2 integration framework 18](#_Toc386453704)

[3.2 Front End 19](#_Toc386453705)

[3.2.1 JQuery 19](#_Toc386453706)

[3.2.2 Ajax 19](#_Toc386453707)

[3.2.3 HTML 20](#_Toc386453708)

[3.2.4 CSS 20](#_Toc386453709)

[3.2.7 JSP 20](#_Toc386453710)

[3.2.7 Bootstrap 21](#_Toc386453711)

[3.2.8 Google Map plugin 21](#_Toc386453712)

[3.3 Choice of Technologies for Recruitment Solution 21](#_Toc386453713)

[3.3.1 Database Options 22](#_Toc386453714)

[3.3.2 Database Connectivity Options 22](#_Toc386453715)

[3.3.3 Web Server Options 23](#_Toc386453716)

[3.3.4 SMTP Plugin 24](#_Toc386453717)

[3.3.5 RMI 24](#_Toc386453718)

[3.3.6 GIT 25](#_Toc386453719)

[3.3.7 JFreeChart 25](#_Toc386453720)

[Chapter 4 Implementation 26](#_Toc386453721)

[4.1 Environment Set Up 26](#_Toc386453722)

[4.1.1 AOP(Aspect-oriented programming) 26](#_Toc386453723)

[4.1.2 Spring control all action 26](#_Toc386453724)

[Struts.xml 26](#_Toc386453725)

[4.2 Project Structure 27](#_Toc386453726)

[4.2.1 DAO Layer 27](#_Toc386453727)

[4.2.2 Service Layer 28](#_Toc386453728)

[4.2.3 Inversion of Control 29](#_Toc386453729)

[4.2.4 JSP 29](#_Toc386453730)

[4.2.4.1 OGNL(Object Graph Navigation Library) 29](#_Toc386453731)

[4.2.4.2 JSTL(JavaServer Pages Standard Tag Library) 30](#_Toc386453732)

[4.3 Registration 30](#_Toc386453733)

[4.3.1 JavaScript, JQuery and AJAX 30](#_Toc386453734)

[Jobseeker and Employer registration validation 30](#_Toc386453735)

[Update password validation 31](#_Toc386453736)

[Update Validation 32](#_Toc386453737)

[4.4 Authentication 32](#_Toc386453738)

[4.4.1 RMI and Session 32](#_Toc386453739)

[4.5 List and Search 33](#_Toc386453740)

[4.5.1 Select and Select2 34](#_Toc386453741)

[4.6 Post Job 35](#_Toc386453742)

[4.7 Delete 35](#_Toc386453743)

[4.8 Update 35](#_Toc386453744)

[4.9 Upload CV and Upload profile Image 35](#_Toc386453745)

[4.10 Download CV 36](#_Toc386453746)

[4.11 Send Interview 36](#_Toc386453747)

[4.12 Update Interview result 37](#_Toc386453748)

[4. 13 Administrator 39](#_Toc386453749)

[4.13.1 Area 39](#_Toc386453750)

[4.14 JFreeChart 39](#_Toc386453751)

[4.14.1 Jobseeker Employment Statistic 39](#_Toc386453752)

[4.14.2 Job Create Statistic 40](#_Toc386453753)

[4. 14.3 Job Category Statistic 41](#_Toc386453754)

[4.15 GitHub 42](#_Toc386453755)

[4.16 Google Map 42](#_Toc386453756)

[4.17 Bootstrap and UI design 44](#_Toc386453757)

[Chapter 5 User Guidelines 47](#_Toc386453758)

[5.1 Main Page 47](#_Toc386453759)

[5.2 Registration 49](#_Toc386453760)

[5.2.1 JobSeeker Registration 49](#_Toc386453761)

[5.2.2 Employer Registration 50](#_Toc386453762)

[5.3 Jobseeker 50](#_Toc386453763)

[5.3.1 Authentication 51](#_Toc386453764)

[5.3.2 Upload CV 56](#_Toc386453765)

[5.3.3 View Interview History 56](#_Toc386453766)

[5.3.4 List Skill 57](#_Toc386453767)

[5.3.5 Update Password 57](#_Toc386453768)

[5.3.6 Manage CV 57](#_Toc386453769)

[5. 4 Employer 58](#_Toc386453770)

[5.4.1 Authentication 58](#_Toc386453771)

[5.4.2 Profile 59](#_Toc386453772)

[5.4.3 Add Job 61](#_Toc386453773)

[5.4.4 View Posted Job 62](#_Toc386453774)

[5.4.5 Send Interview Email 64](#_Toc386453775)

[5.4.6 View Send Interview Letter 65](#_Toc386453776)

[5.4.7 Update password 66](#_Toc386453777)

[5.5. 1 Administrator 66](#_Toc386453778)

[Chapter 6 Installation Manual 73](#_Toc386453779)

[Chapter 7 Test Plan 76](#_Toc386453780)

[Conclusion 83](#_Toc386453781)

[References 84](#_Toc386453782)

# Chapter 1 Introduction

**Project Title**

Online Recruitment Solution

## Project Overview

The system uses a modularized program design method, and to the full Struts2, Spring, Hibernate, though MYSQL Database server, it exploits MVC design pattern, therefore it provides data query, modify, add, delete, maintenance function. It implements member information management. This system is reliable, scalable, reusable, and it provides a convenient searching platform for the users. The integrated development environment of the systems uses 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. Then build more technical functionalities after December.

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

**Preparation**

I will be working 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 backend 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 backend 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 jobseeker 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.

## Introduction

My idea for my final year project is Recruitment website, as I am currently looking for a graduate job who experiencing view a large amount of the jobs that I am not interested in and take time for search jobs. Also, the recruitment site usually crowded with load of advertisement and unrelated information about other news, event. Especially that I am a final year student with load of college works and part-time job. With my experiences of using recruitment website that I would like to develop a site that will help others make quick search and find right jobs with neat layout and better searching environment. With vast majority using recruitment site for searching job and hiring people, it seemed like the best useful site to develop. J2EE is fundamental of building website. I would like to build this knowledge J2EE from my Final Year Project to again hand experiences on how Struts2, Spring and Hibernate combined together to build a powerful website.

## 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 registers with this product by filling in the registration form, then receiving notification email confirming registered. Login to search applicant’s skills matched with Jobseekers, to update company information, post jobs and view jobs advertised and view who applied, manage/edit job advertised.

For job seeker who registers with this site, will receive notification email confirming registered. Jobseeker login to system, only view the jobs that matches with jobseeker’s skills at registration. Job seeker use this website to find jobs and able to search and view jobs advertised by employer, edit/upload CV, cover letter, update user profile, contact with administrator.

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, and manage all the data that stored in database.

## 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 incorporate 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 website is fulfilled with advertisement, 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 assists 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 four different users: administrators, visitor, 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. Administrator can view the reports of statistic, percentage of employed.

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

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

*Visitor:* view all jobs in system, search jobs.

## 2.2 Requirement analysis:

In order to develop a comprehensive project and understand the crucial request for making the best of web site and technology used around, I conducted an extensive requirement analysis. This analysis involved deep research on various functionalities, such as knowing relationship of each objects, to help with all these functionalities and requirements that I investigated other recruitment site to see their features and what I should add to make it a better than these site. I also ask friend of mine who have solid experiences on developing J2EE website and give me more insight at what way I could possibly perform my process on the application. This helped me a great deal in what approach I should work my web site. They all each gave me their own personal insight at what might happen in certain situations and what they would want to happen in these predicaments relating to create this web site, so that I document the requirements as follow.

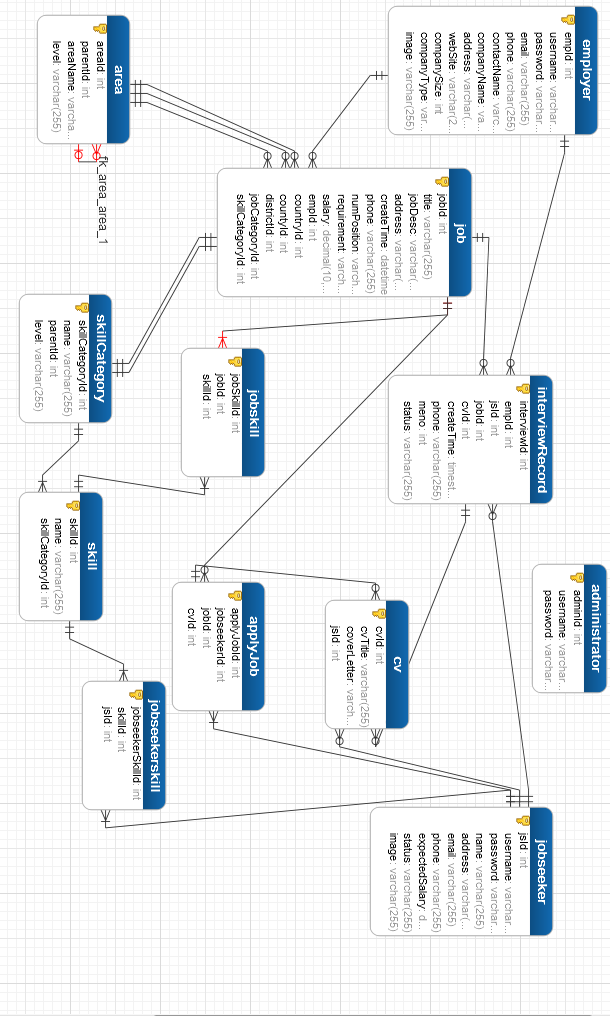
**Requirement:**

*Visitor:* is someone who is not logged in to the system, who is able to view all jobs that are posted in system and search for jobs by country, city, district, job title, job category, job, job skills and date. Also visitor can sort jobs by category, salary, create time, Visitor is not able to apply for a job.

*Employer:* someone who is registered with this product by filling in the registration form, then receiving notification email confirming registered. Employer can login into this website and post a job with Job Category, Job, and skills, employer can post more than one job. Once this job is posted, the system will match this job with skill or skills with Jobseekers and show in main page. If there is no Jobseeker with matched skill or skills exists, there will be no Jobseeker shown when Employer login. When a jobseeker applies for a job, the Employer will see the Jobseeker detail with CV and cover letter under this job, system will check if this Jobseeker’s status is unemployed, so that Employer is able to send interview invitation email to this jobseeker specifying time and message to this Jobseeker. This is designed as, during a period of time the Jobseeker may get a job with another employer. After interviewing this Jobseeker, the employer can login to the system, change the status of interview and the system will send an email to inform them of result. If interview success, change the status of this Jobseeker to be employed, also number of position minus one. If the number of position for this job equals 0 then this job no longer appears in the page. Employer can view, update, delete and insert jobs in profile page and update its password. Employer also can make search of Jobseekers.

*Jobseeker* registers with system by filling in the registration form with their skills and uploading a profile image, when they login they will be shown the jobs that match the Jobseeker’s skills in order of best match and possible match, best match order is the job’s skills are exactly matches with Jobseeker’s skills. Possible match order is job’s skill is not exactly matches with Jobseeker’s skills, but at least one skill matches. After successful registration, notification of welcome email will be sent to this Jobseeker. Then Jobseeker can view and apply for jobs, interview record, also update, delete CV, skills, password and profile. When Jobseeker is applying for a job the system will ask the Jobseeker to select an uploaded CV, if none exists will ask Jobseeker to upload one.

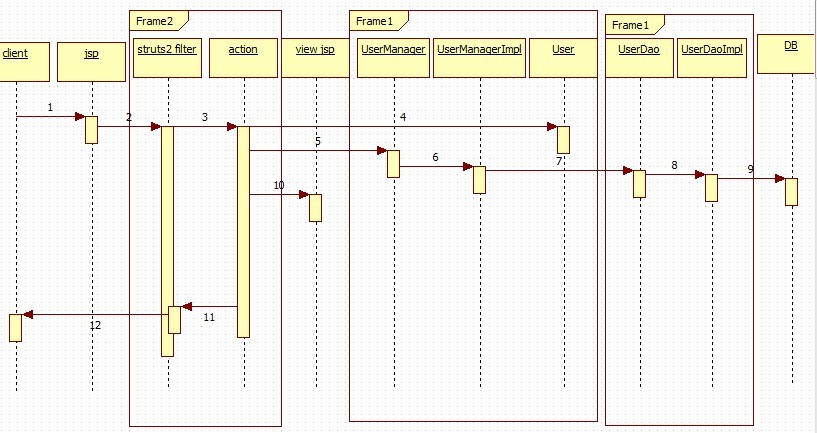
I used Navicat for MySQL to draw this ER diagram in next page.

******

# Chapter 3: Design

## System Architecture

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



Client accesses the JSP page that is managed by Struts2 filter controller specified by the action that directs to relevant 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. This level deals 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 frameworks 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 separate 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)

## _wk_80289377480beac24100f51bc45ebc60_0

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, Spring is not limited to the use of server-side development. Java applications benefit from Spring with Simplicity, testability, and loosely-coupled. 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. With Spring the processing overhead is very cheap. In addition, In addition, Spring is non-intrusive: typically, Spring Application object does not reply on the 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 simplify 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)

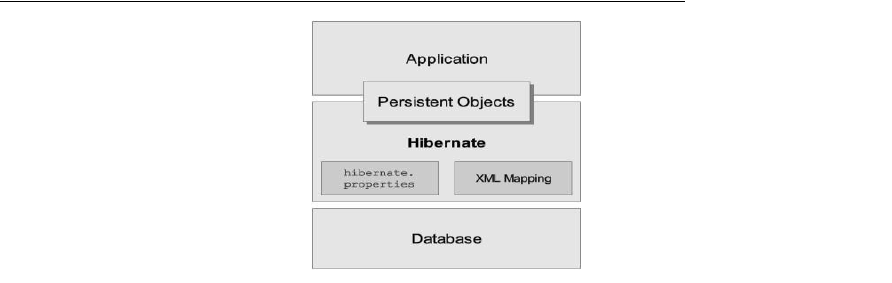
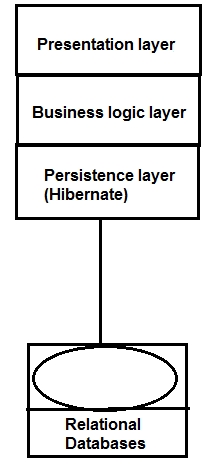
****

Figure 1 - 2 Hibernate architecture overview diagram

Hibernate has 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, you do not need to insert a lot 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 does not have a separate 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 database file is cross-platform. SQLite is lacking in some features due to small size of SQLite and is not very scalable. Nevertheless, performance is quite good even in low memory environment.

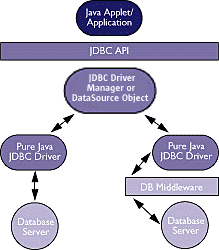
**Microsoft Access:** is known as Microsoft office access, database management system from Microsoft that combines relational Microsoft Jet Database Engine with a graphical user interface and software-development tools. It storing data in own format based on the Access Jet Database Engine. It import 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) 

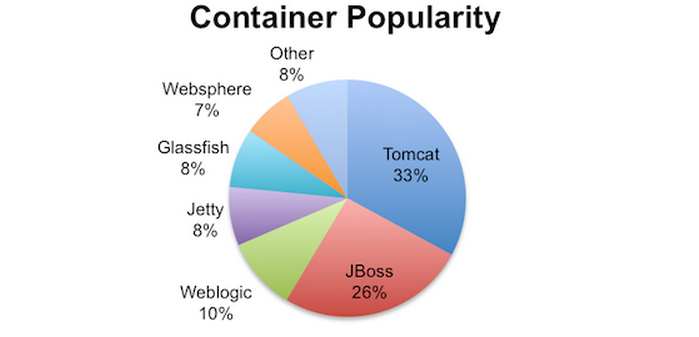
**Hibernate:** object-oriented mapping library for programs written in java based language it provides a framework for mapping java classes to database table. It also provides data query and retrieval facilities. It create SQL call and tries to relieve the developer from manual result set handling and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

**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 the Simplicity, testability, and loosely-coupled,



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 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 practical experiences on version control. (git-scm.com/)

### 3.3.7 JFreeChart

JFreeChart is based on Java language, for applications, Applets, Servlets, and JSP, etc. JFreeChart can generate a variety of charts, ie charts (pie charts), histogram (bar charts), Scatter (scatter plots), the timing diagram (time series), Gantt charts (Gantt charts). JFreeChart is a free 100% Java chart library that makes it easy for developers to display professional quality charts in their applications. JFreeChart's extensive feature set includes:

* a consistent and well-documented API, supporting a wide range of chart types;
* a flexible design that is easy to extend, and targets both server-side and client-side applications;
* support for many output types, including Swing components, image files (including PNG and JPEG), and vector graphics file formats (including PDF, EPS and SVG);
* JFreeChart is "open source" or, more specifically, free software. It is distributed under the terms of the GNU Lesser General Public Licence (LGPL), which permits use in proprietary applications. (.jfree.org/jfreechart/)

# 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 response 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"*>

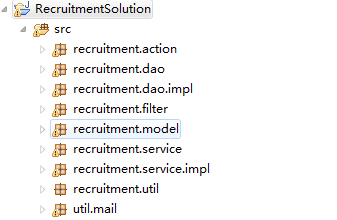
<result name=*"success"*>/success.jsp</result>

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

</action>

## 4.2 Project Structure

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



### 4.2.1 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.

### 4.2.2 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.

### 4.2.3 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.

## 4.2.4 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.

### 4.2.4.1 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

### 4.2.4.2 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.

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

### 4.3.1 JavaScript, JQuery and AJAX

### Jobseeker and Employer registration validation

All validation JavaScript files is under WebRoot/js.

I coded every forms page that required fill in with client side validations, when user move away mouse from textbox with \* marks beside it, the message will appears beside the textbox to indicate user if data entered is validate. It gives quick response to user to let them know if data entered is correct. I implemented this by using blur() in JQuery to check if data entered is valid.

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.

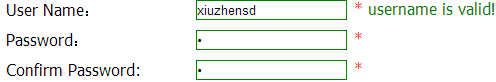
Also for registration form, I also have client side validations, when user’s mouse move away from input box, will call JQuery of blur() method to check to see if the user is entering valid data, if it is, red dot(\*) beside relevant input boxes will change to green font of texts indicates validate data,

Upload profile that user only allow to upload image in jpg|png|gif format, if not error message will appears to ask user to enter correct format.

E.g

C:\Users\MARY~1.REG\AppData\Local\AppData\Roaming\Tencent\Users\27248466\QQ\WinTemp\RichOle\R~J}ASLHMT}K860PSFZ9XWN.jpg

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.

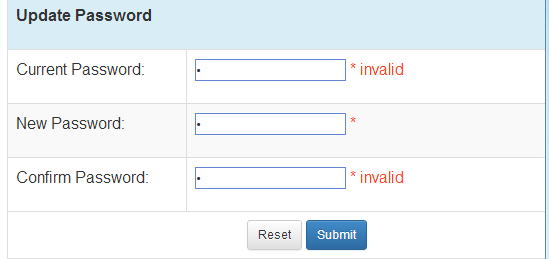
jsCheckUsername.action?js.username=" + username;

"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

### Update password validation



For Jobseeker update password is coded in jsPasswordValidation.js, for Employer password validation is in empPasswordValidation.jsp.

To check the current password using AJAX to check with database if matches, if not the mouse move away from it, the message will appear to indicates the error. I use JavaScript to compare if two passwords entered are same to each other.

### Update Validation

WebRoot/js/upEmpValidation.jsp, WebRoot/js/updateJspValidation.jsp and

WebRoot/js/editJobValidation.jsp to make sure all textboxes beside the \* marks are filled by the Employer and the file uploaded is in required format of jpg, png and gif.

## 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’.

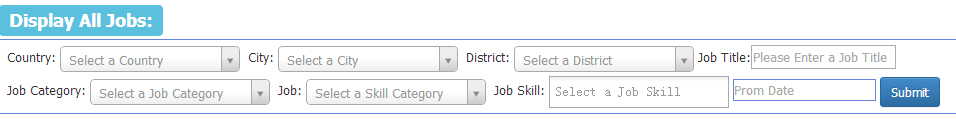
ServletActionContext.*getRequest*().getSession().setAttribute("admin", admin);

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



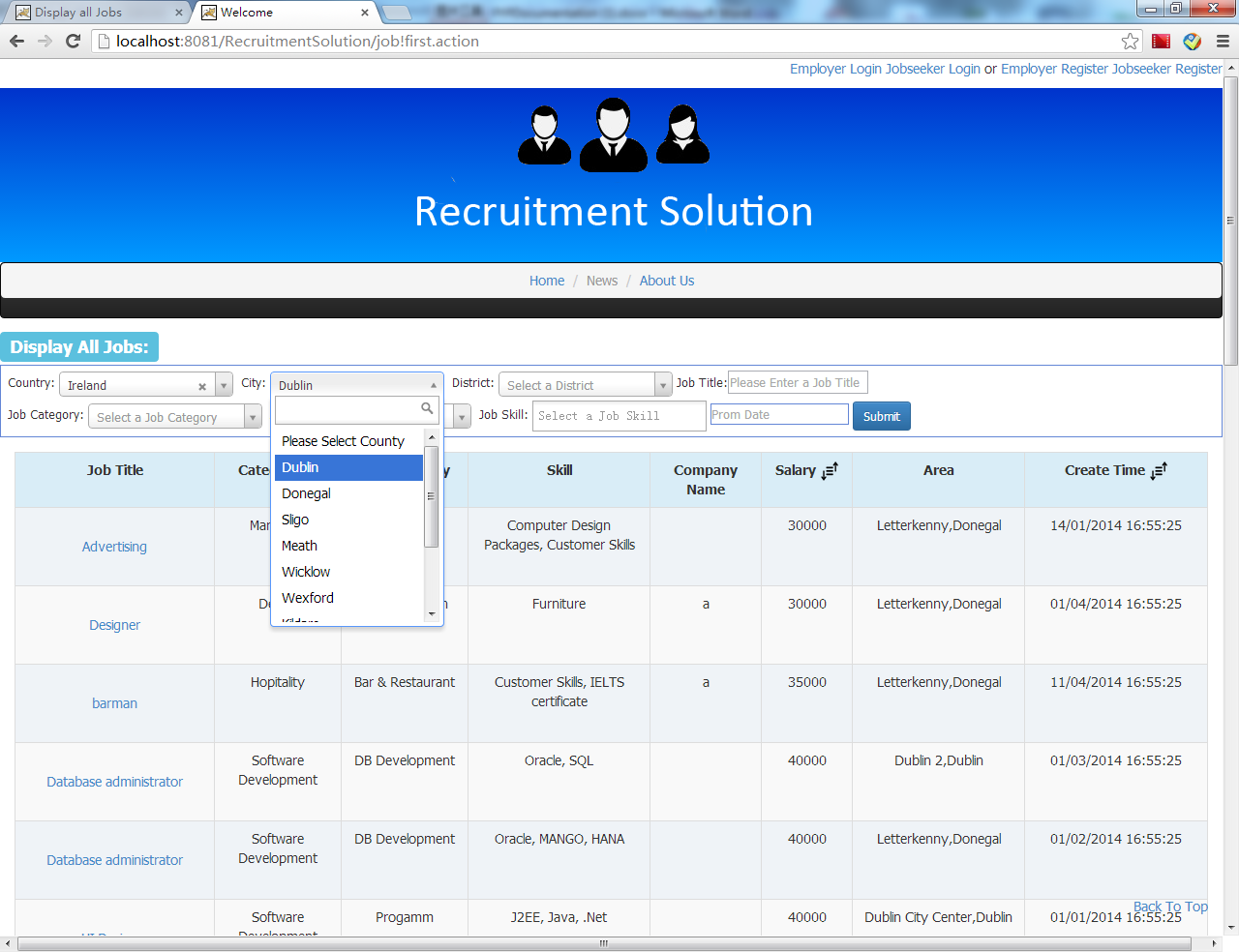
For list and search Jobs that I implemented method getJobs() in JobDaoImpl class to show all jobs by HQL. One method with if statement and append into HQL make search.

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 HQL 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.

Note that all the searches if it is entered by textbox are partial search.

### 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 and Upload profile Image

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.

Upload image is implemented same way as upload CV, except that when upload image is at registration, user id is auto-increment, image need to append user id after save user into database. I saved user into database first then get user id and append it with profile image name as id\_image.png to save into database.

I set restriction for upload CV format as only doc, docx, pdf format can be upload to system. For upload image, I set restriction for upload image file format as jpg, png, gif in jsValidation.js, blur() method will check if user select correct format, if not error message will appears immediately to user.

## 4.10 Download CV

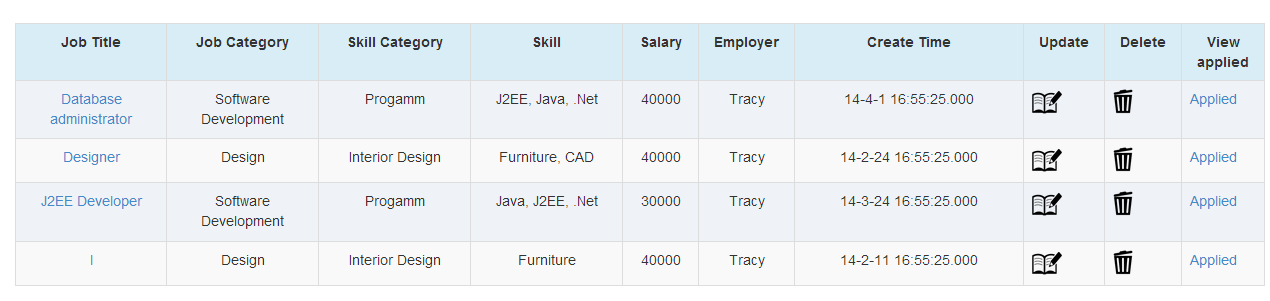
Every CV name in JSP pages allow user to download CV, when Jobseeker, Employer, Administrator, click a href link of CV names that will allow user to down load CV.

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

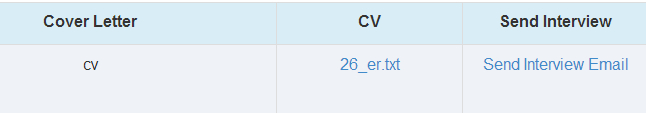
This is calling downLoadCV in CVAction with url getting file name from url, then download it from Tomcat webapps\RecruitmentSolution\upload.

## 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.

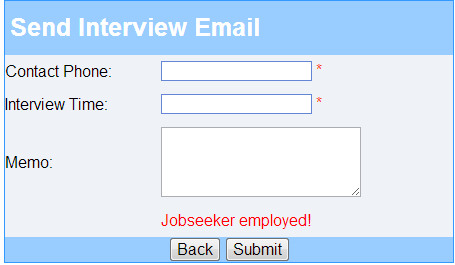


Then show all the jobseekers who applied this job, Employer clicks View a href link url to job!listCVByJsId.action?job.jobId=1&js.jsId=1, Get CV from this jobseeker with job id and Jobseekers id. Employer enable to click a href link to download CV,



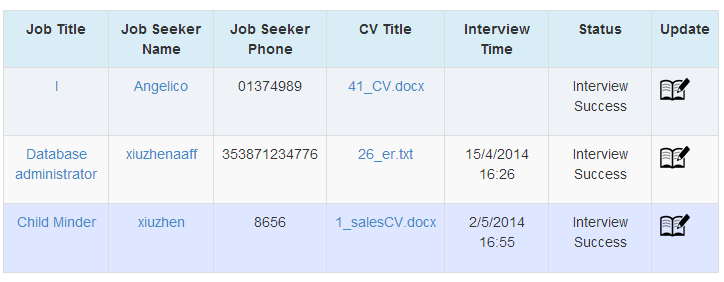
When Employer clicks Send Interview Letter links ir!preSend?ir.job.jobId=1&ir.js.jsId=1&ir.cv.cvId=8, brings page of Send Interview Letter, this is to generate Interview Record table and send email to this Jobseeker’s email address for interviewing, Jobseeker also able to view the history of interviews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

As today busy environment Jobseeker may find a job before that Employer views this Jobseeker’s CV. Validation will check in here, as interview email only can sent to Jobseeker who’s status as unemployed, status = 0. If Jobseeker status = 1, then message will show in this form indicates Employer ‘Jobseeker is Employed’. Also, Employer not allows sent two interviews of one job to one Jobseeker, if interview is already sent, the error message will shows you had already sent! This is implemented in InterviewRecordAction with isAlreadySend() isEmployed() to check and append the message in the table.



## 4.12 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.





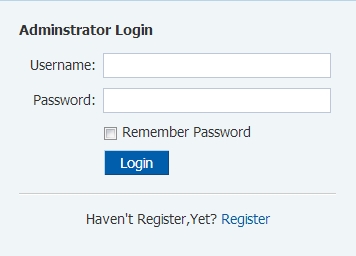
update url of ir!preUpdate?ir.interviewId=5, get this interviewed of interview record, then update interview result with default ‘No Interview’ status is 1, ‘Interview Success’ status is 2, ‘Interview fail ’ status is 0. Once Employer updates the interview status, it will also update Jobseeker’s table, 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.

Validation in this page, when Employer update the status of interview record that system will check if Jobseeker is employed, if it is employed that error message will appear This Jobseeker is employed! The logic behind is that one Jobseeker can only have one job.

## 4. 13 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.

## 4.13.1 Area

I was going to reorganize my project into one Dao instead of each model. In area, update, insert and delete, I have the DBDao and DBImpl to do it and inject Dao in action in order to achieve this, but I ran out of time to restructure it all.

## 4.14 JFreeChart

### 4.14.1 Jobseeker Employment Statistic

I implemented JFreeChart in recruitment.action package with ReportAction class to display the statistics of Jobseeker employed and unemployed. I map it in struts.xml to set size and action of the chart. I coded the HQL statement in JobseekerDaoImpl to count employment and unemployment of jobseeker and return rate of Action and display it in page.

<package name=*"report"* extends=*"struts-default,jfreechart-default"*>

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

<result name=*"success"* type=*"chart"*>

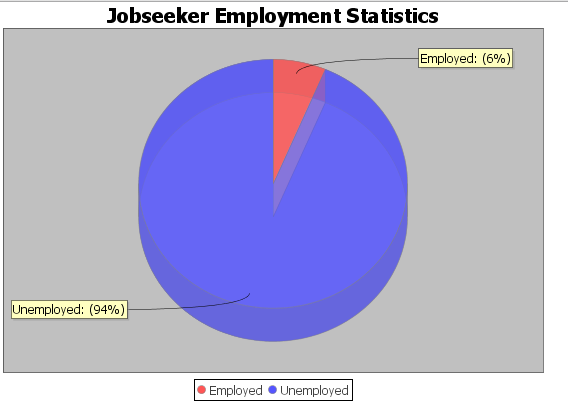
<param name=*"height"*>400</param>

<param name=*"width"*>550</param>

</result>

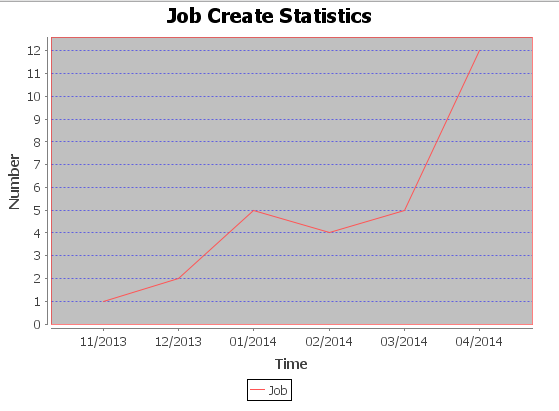
</action>

</package>



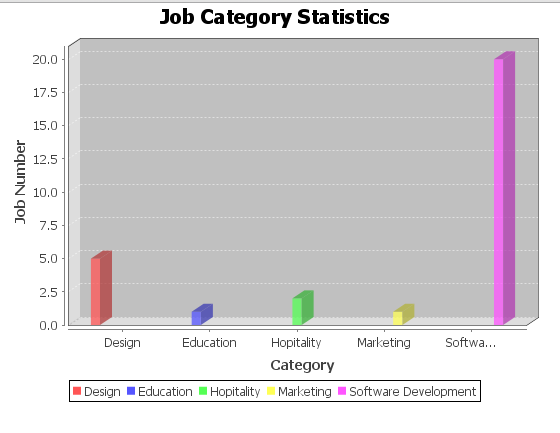
### 4.14.2 Job Create Statistic

In Report2.Action, I invoke the method of getJobForAdmin() to get all the jobs that created in system and use Map to show jobs in chart, and set it in struts.xml



### 4. 14.3 Job Category Statistic

In report3Action, to list different jobs in different category, I have done this by list all jobs that in system by using for loop and put in Map and set the in struts.xml



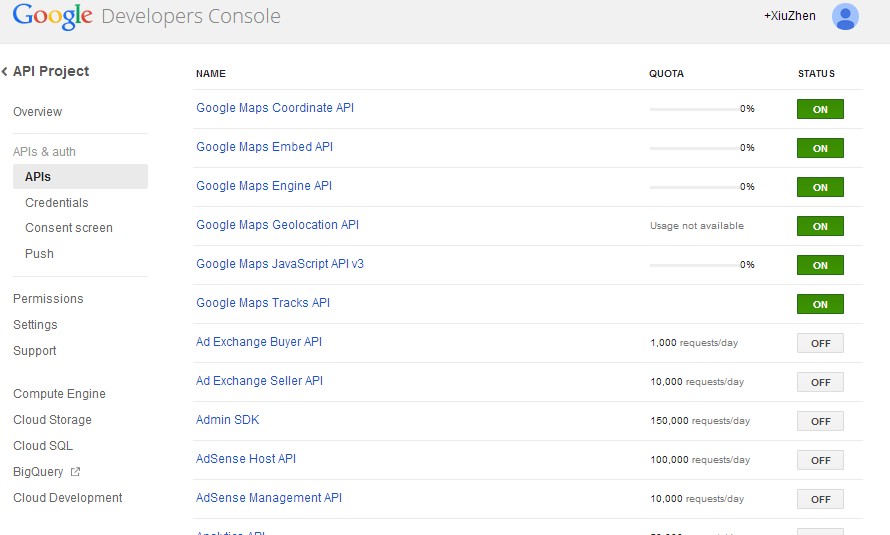
## 4.15 GitHub

Repository: <https://github.com/xiuzhen1103/RecruitmentSolution>

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 GitHub for Desktop from <https://windows.github.com/> that is easier to synchronize 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.

## 4.16 Google Map

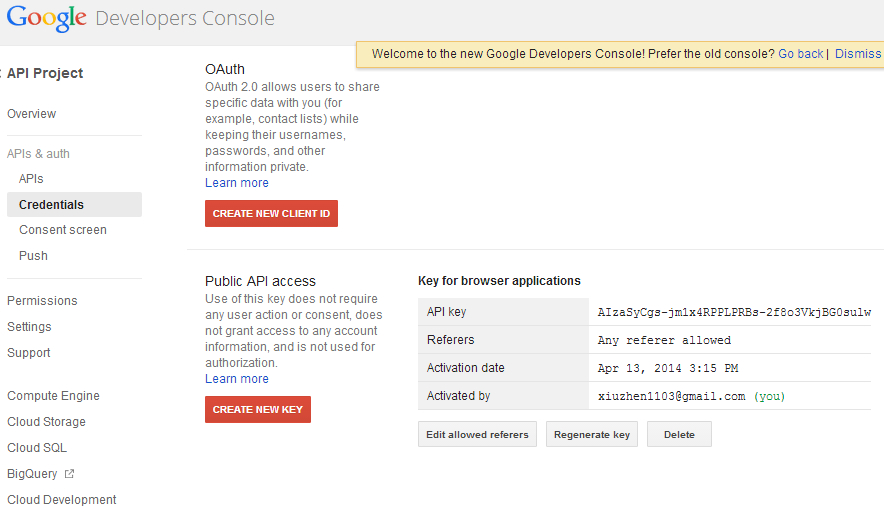
I embedded Google Map into About Page with JavaScript, firstly go to https://code.google.com/apis/console/, and log in with your Google Account. Click API on left hand side and turn on Google Map status.



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>

Add “We are here” to indicate the location of office location, I set zoom size to be 18 so that will show detail of location.



## 4.17 Bootstrap and UI design

I used bootstrap and my own CSS Style sheet page for designing UI,

I include these links in order to use Bootstrap’s function.

<link href=*"*<%=basePath%>*style/bootstrap-theme.min.css"* type=*"text/css"* rel=*"StyleSheet"* />

<link href=*"*<%=basePath%>*style/bootstrap.min.css"* type=*"text/css"* rel=*"StyleSheet"* />

<script type=*"text/javascript"* src=*"*<%=basePath%>*js/bootstrap.min.js"*></script>

My own style sheet is following that specifies some design that I aim for the page.

<link href=*"*<%=basePath%>*style/style.css"* type=*"text/css"* rel=*"StyleSheet"* />

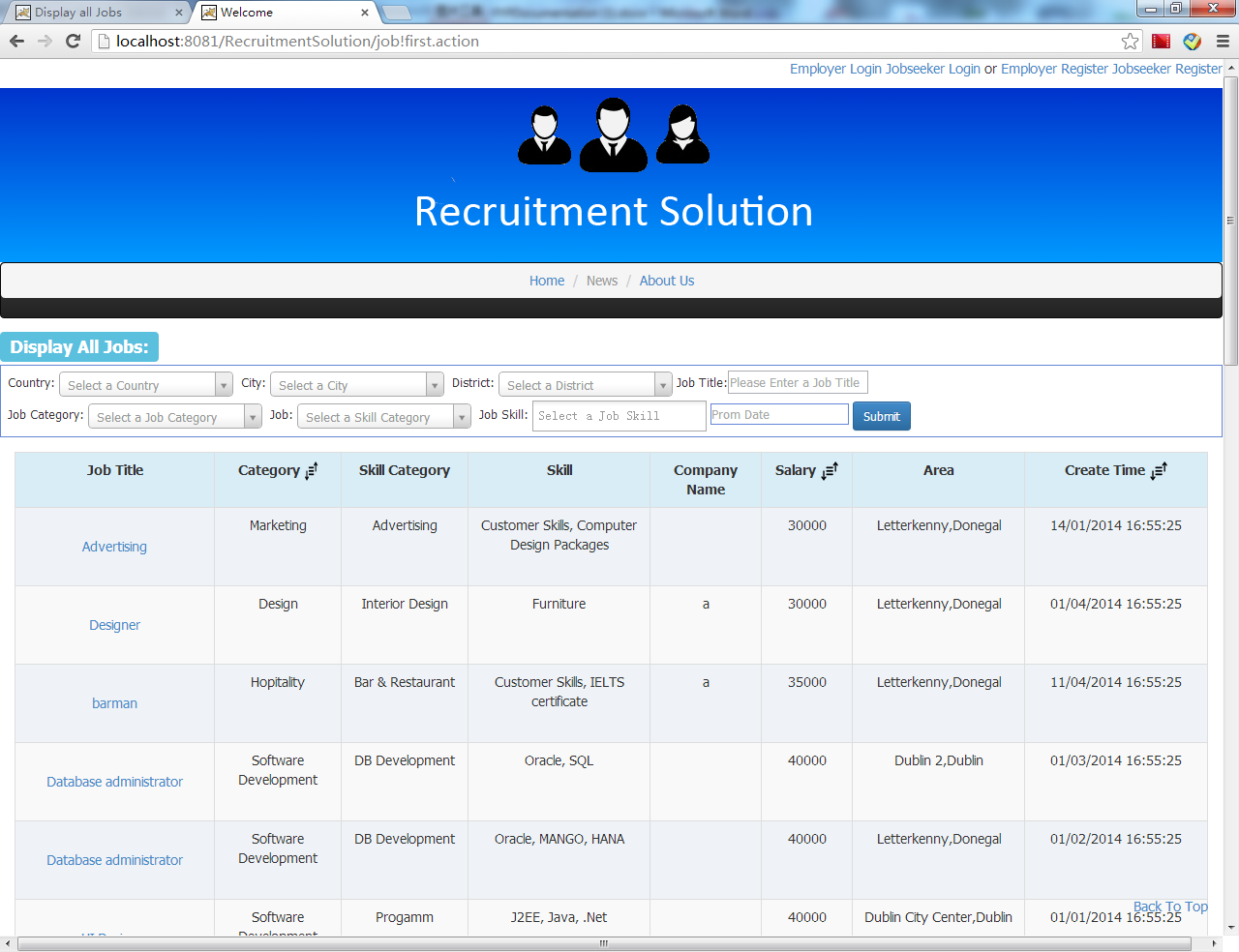
For main page, the banner I used Photoshop to have Recruitment Solution as transparent then cut out the icons without white space, use .banner to set size and background as that image to compatible with different browser such as Old browsers, FF3.6+, Chrome, Safari4, Chrome10+, Safari5.1+, Opera 11.10+, IE10, W3C, IE6-9.

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 | Collection framwork |
| 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 |  |  |

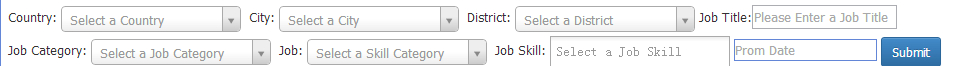
# Chapter 5 User Guidelines

## 5.1 Main Page



Any user that is not logged in to this Recruitment Solution site is able to view all the jobs that are posted in this site. The search bar in the top allows the user to search jobs by country, city, district, job category, job and skills. User only can choose country first, then choose city under this country, after choose county able to choose district. Job title can be entered by any character for search purpose. For date search, search jobs between the date entered and current date.

Note: for search title, is partial search, user not required to enter the full job title.

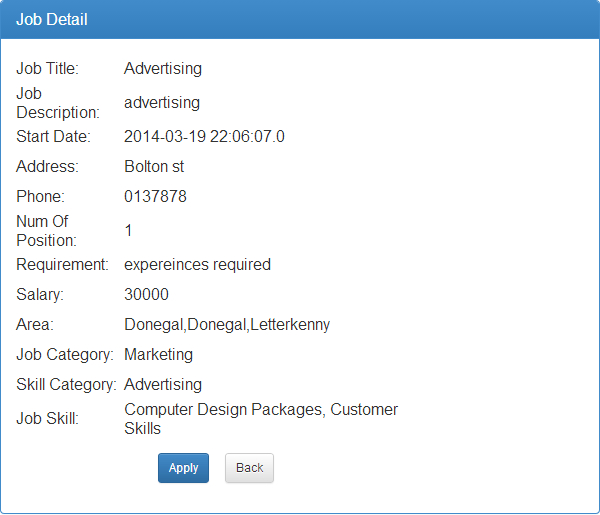


C:\Users\MARY~1.REG\AppData\Local\AppData\Roaming\Tencent\Users\27248466\QQ\WinTemp\RichOle\7{E0PBPRU{~EU{MULO(}695.jpg beside option can be close if user change choice. User can search more than one skills.

C:\Users\User\AppData\Roaming\Tencent\Users\27248466\QQ\WinTemp\RichOle\$`@%GD474W]X0_LV82}7L~T.jpgC:\Users\User\AppData\Roaming\Tencent\Users\27248466\QQ\WinTemp\RichOle\5BR9E4}4J5BNSB8Z{SHLT31.jpgC:\Users\User\AppData\Roaming\Tencent\Users\27248466\QQ\WinTemp\RichOle\]U9ZCK1AW15E6)PG)L{GQ0Y.jpg Click sort icons beside it, can sort jobs with Asc or Desc orders.

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)

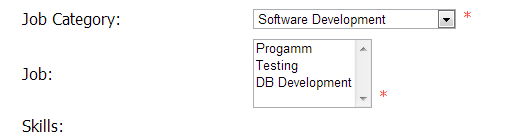
## 5.2 Registration

### 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. User name should be unique, two passwords entered need to match each other, name, job category, job and skills should not be empty. Profile image can be empty as default image is given, but if user decide to upload image, should be in format of jpg, png and gif.



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.

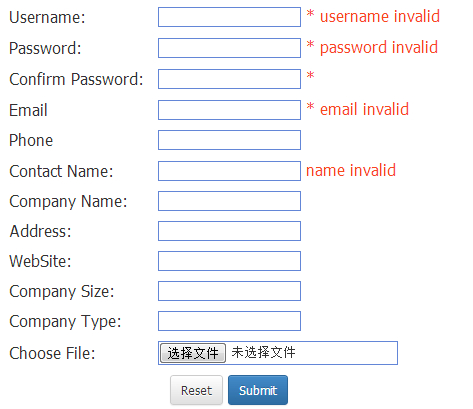


Multiple choice of skills allow Jobseeker to select.

C:\Users\MARY~1.REG\AppData\Local\AppData\Roaming\Tencent\Users\27248466\QQ\WinTemp\RichOle\_RGO2_7KTNGY9O%JN[R5SKC.jpg

### 5.2.2 Employer Registration

Employer registration is straight forward without any entering valid user name and two match passwords, contact name and website and profile image can be empty as default image is given, but if user decide to upload image, should be in format of jpg, png and gif.



5.3 Jobseeker

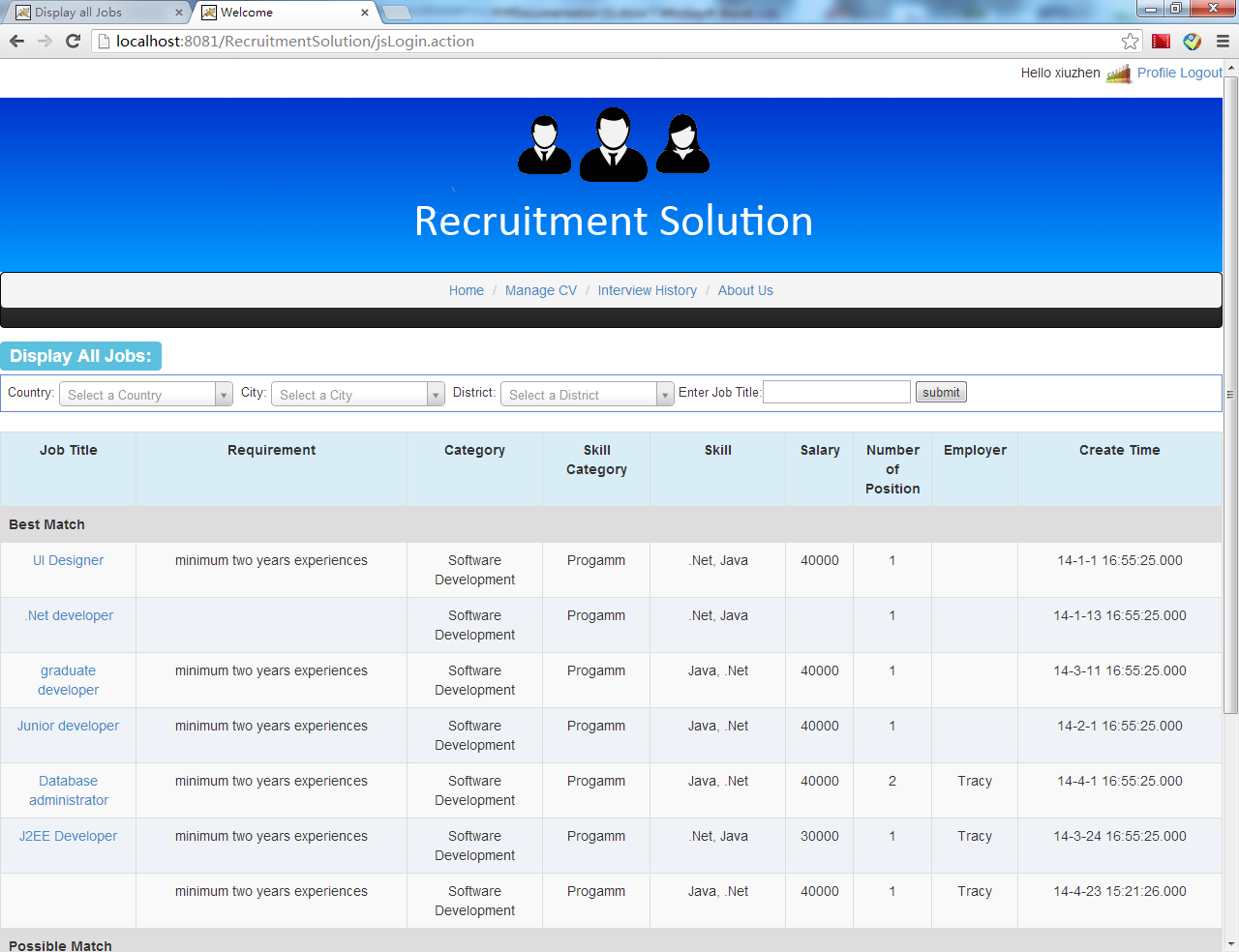
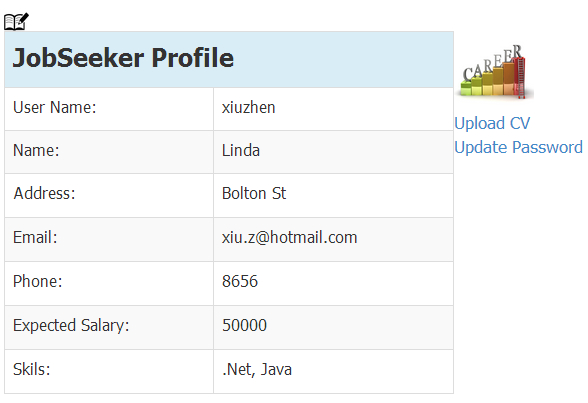
### 5.3.1 Authentication

![C:\Users\User\AppData\Roaming\Tencent\Users\27248466\QQ\WinTemp\RichOle\H}[7`HE~]1O7941D~J6](LM.jpg](data:image/jpeg;base64,)

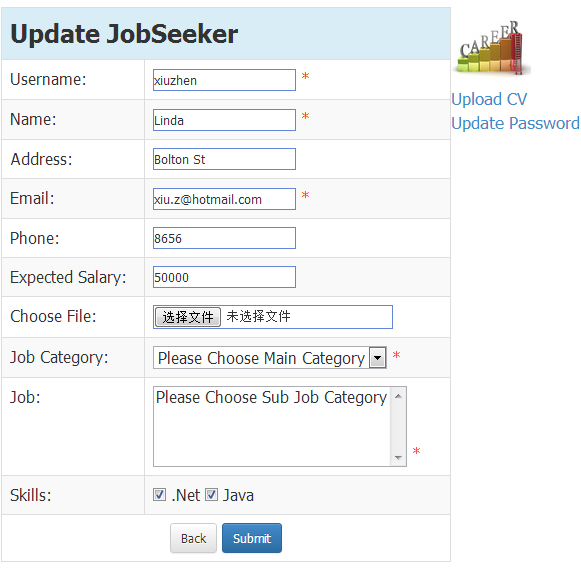
After completed jobseeker registration the jobseeker is directed to Jobseeker login, if authenticated, the system loads jobs that match with Jobseeker previously selected skills in registration. With order of best match and possible match, best match means all the skills exactly matches with jobs. Possible 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. Jobseeker can view a job, then click Apply button, 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 directs the Jobseeker to view and update their details by directly modifying the data and pressing the submit button, then success, will bring back to this page again. More href links on the right hand side of Jobseeker profile that link to [Upload CV](http://localhost:8081/RecruitmentSolution/addCV.jsp)  [Update Password](http://localhost:8081/RecruitmentSolution/updateJsPassword.jsp)  [List Skill](http://localhost:8081/RecruitmentSolution/jsSkill!list.action)

Click edit icon that bring to update the profile page



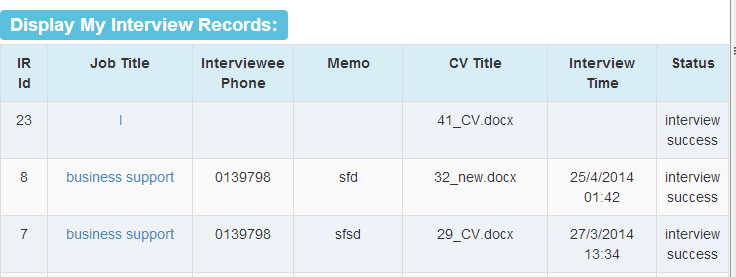
### 5.3.2 Upload CV



Upload CV links to Add CV page, Jobseeker upload CV and enter cover letter, in here upload CV the format of CV should be doc, docx and pdf, if not the correct format, an error message will be given.

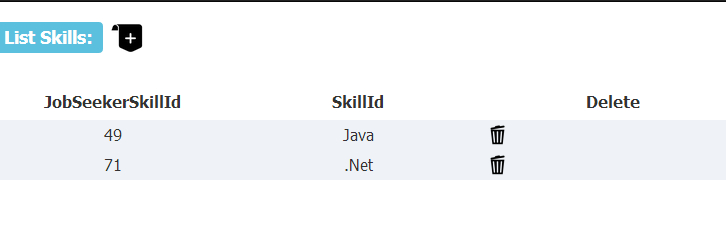
### 5.3.3 View Interview History

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



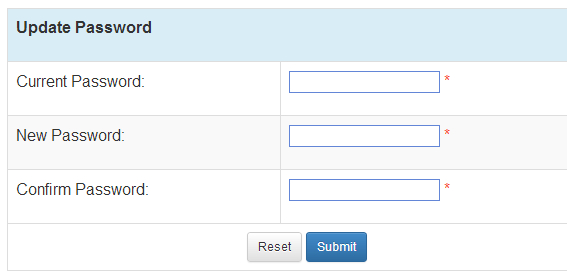
### 5.3.4 List Skill

List skill link shows all skills that the Jobseeker selected at registration, Jobseeker can delete skills or click update skill to go to update skill page.



### 5.3.5 Update Password

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



The current password entered should be the same as the one as stored in the database. Two passwords entered should match each other, if not a message will immediately appear to user as invalid.

### 5.3.6 Manage CV

Jobseeker is able to manage the CV that is uploaded to system, also able to view CV and edit it by click the icon for updating and bin icon for delete the CV. Click a href link of CV name will download the CV. 

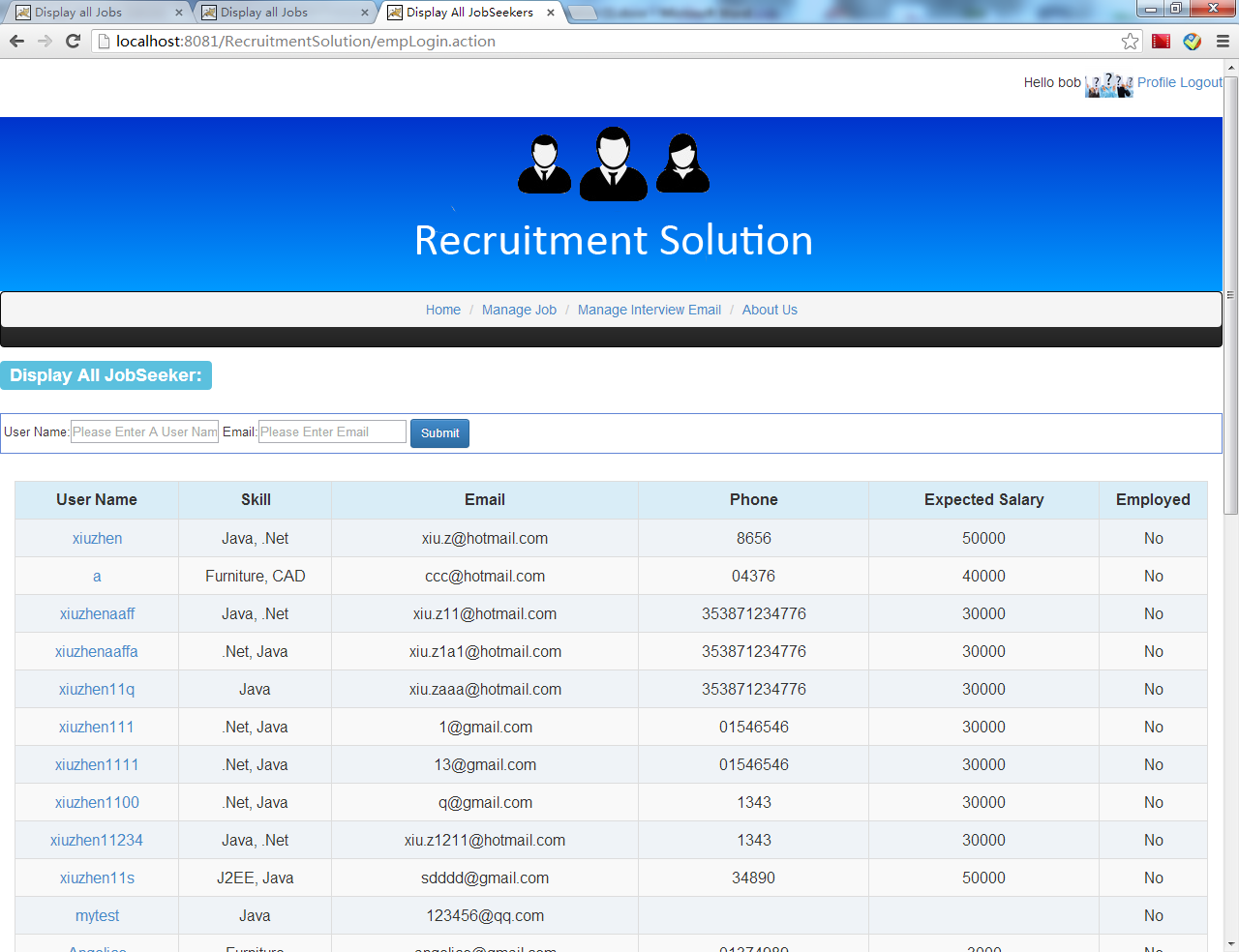
When Jobseeker click update icon will brings to following page, also the upload CV format only can be doc, docx and pdf.



## 5. 4 Employer

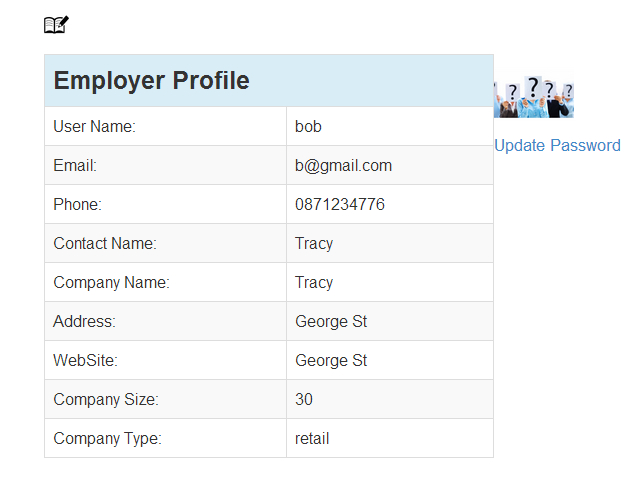
### 5.4.1 Authentication

After Employer is authenticated, they are directed 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.



### 5.4.2 Profile

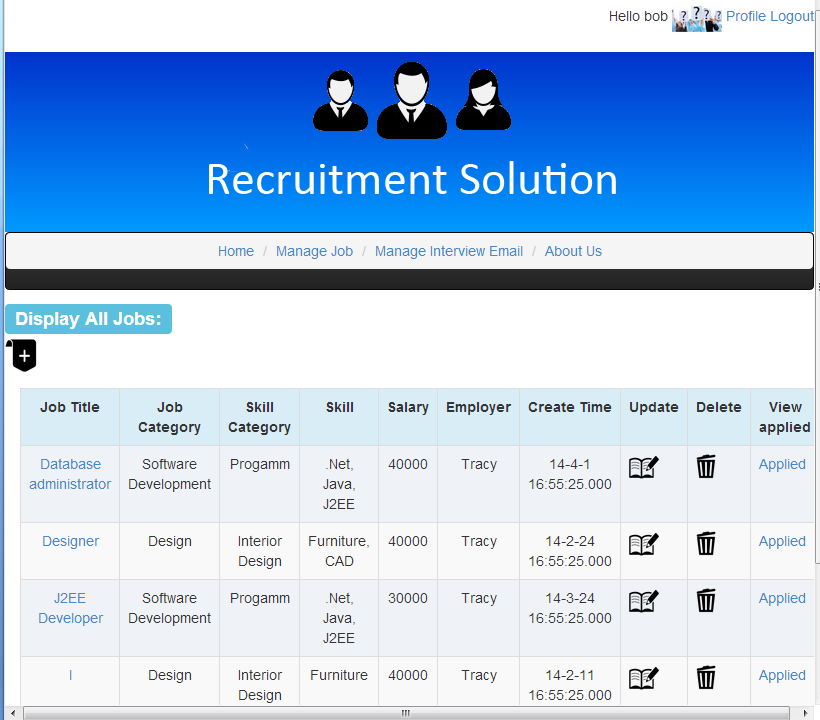
Employer’s profile is updated by clicking the edit button, for update this detail, the company name should not be empty and format of upload image should only be jpg, png and gif, otherwise the error message will immediately appear beside the textbox when user move away mouse form textbox, as Employer modifies the page and click submit button, success of modification will return back this profile page.



Click edit icon to update the Employer profile



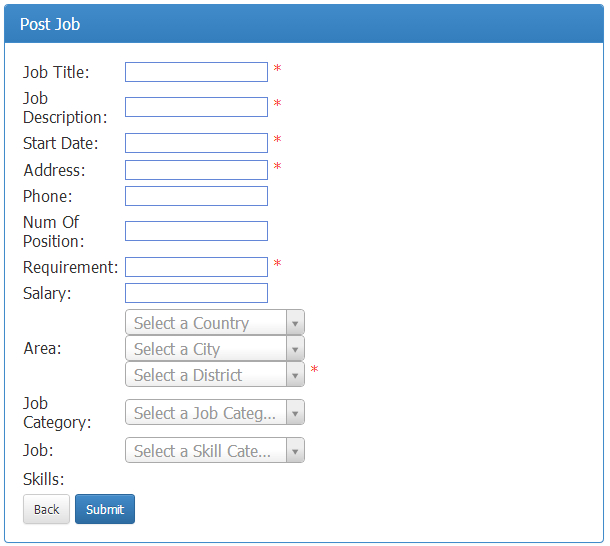
In right hand side of Employer a href link direct Employer to [Update password](http://localhost:8081/RecruitmentSolution/updateEmpPassword.jsp)  pages.



### 5.4.3 Add Job

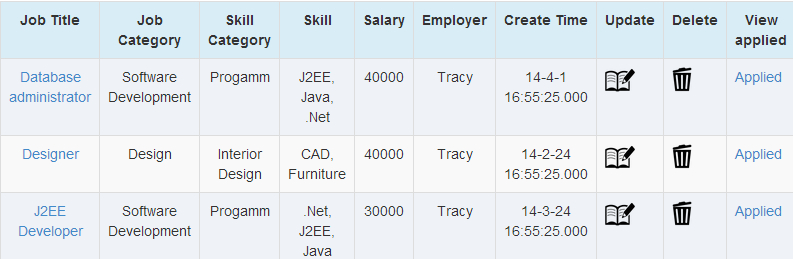
Click manage job in navigation panel will list all jobs, click add icon to add a new job.

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. The \* beside the textboxes means they need to be filled in, otherwise the error message will appear.

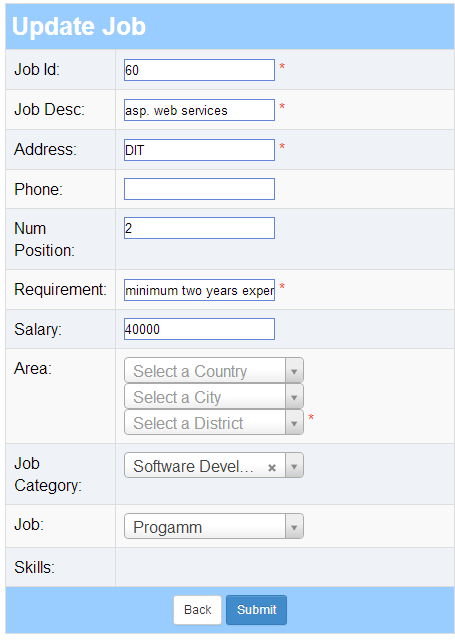


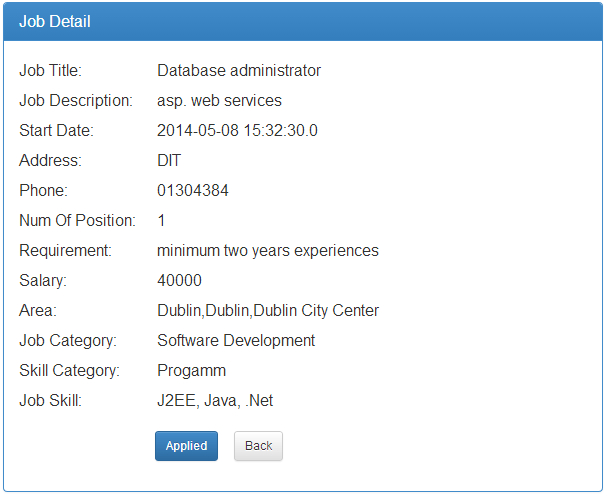
### 5.4.4 View Posted Job

This page lists all posted jobs by Employer, when Employer clicks a href link under Job Title, it links to the page of detail job. Delete icon enable Employer to delete this job and update icon will bring to the page for update.

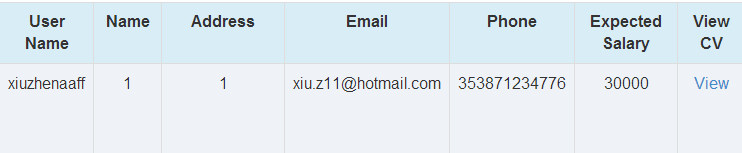


For update job same as add Job all the textbox with \* beside it should be filled, otherwise error message will immediately appear.

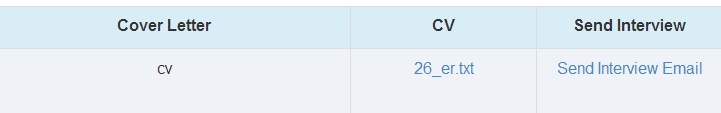




This job detail has an Applied Button, if there is a Jobseeker who has applied for this job, it will list the jobseeker’s details.



The View link directs to the page showing the CV and cover letter for this Jobseeker. Clicking 26\_er.txt (the CV filename) 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) directs to the form of sending Interview to this Jobseeker .



### 5.4.5 Send Interview Email

Also this interview email can’t sent to Jobseeker that who’s status as employed, otherwise error message will appears will Jobseeker clicks the submit button, as one Jobseeker can only have on Job. Also in fact that in today’s busy environment, Jobseeker may got a job before Employer views Jobseeker’s CV. Another business rule is that one interview email cannot be sent twice to same Jobseeker for the same job.

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

Format of Interview letter is:

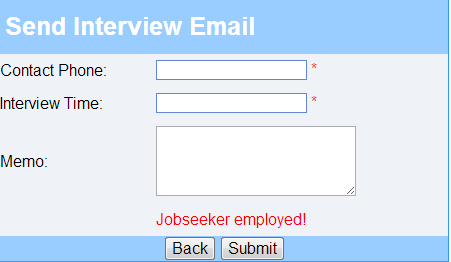
Dear XiuZhen,

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

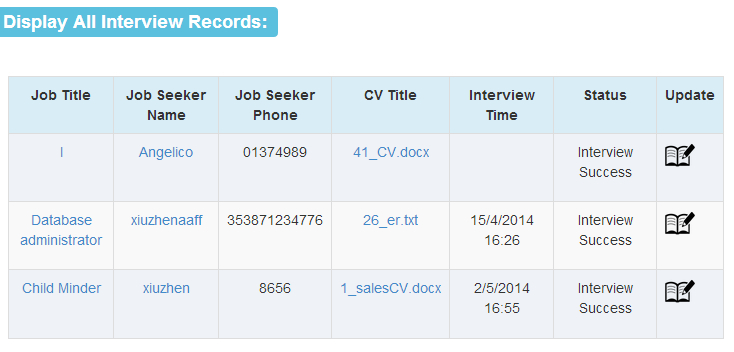
Best Regards

Tracy

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



### 5.4.6 View Send Interview Letter



This page shows all the sent interviews, Update a href link, directs to the page for update interview record, Interview record that cannot be sent to Jobseeker whose status is Employed, otherwise error message will appear as Jobseeker may have got job already. Employer selects Interview Result of ‘No Interview’, ‘Interview Success’ or ‘Interview Fail’ to indicate status of interview. If success, system changes the Jobseeker’s Employed status to ‘Yes’, and sends Interview Result Letter to this Jobseeker and updates Jobseeker’s status to Employed. If result failed, the Interview Result Email will be sent to this Jobseeker’s email to inform them of the result. It also updates the Jobseeker’s History of Interview, when Jobseeker logins to the site, clicks profile, under History of Interview they will also able to find interview result.



### 5.4.7 Update password

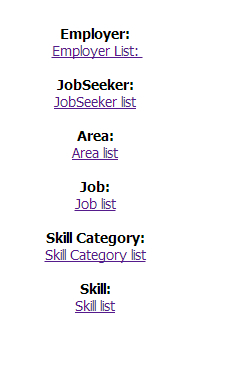


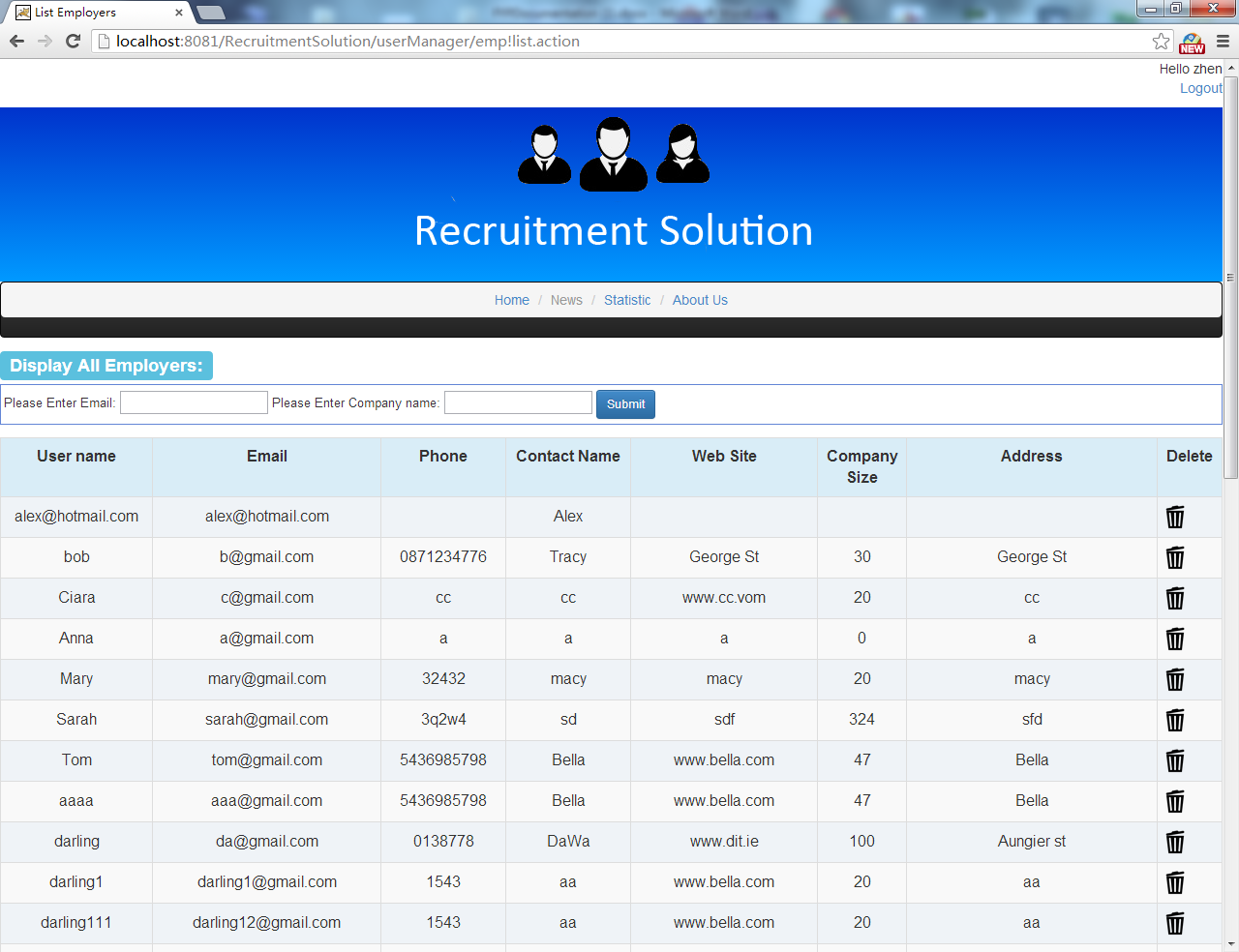
The password entered in current password textbox should match the one that exists in the database, new passwords should exactly match each other, otherwise error messages will appear immediately.

## 5.5. 1 Administrator

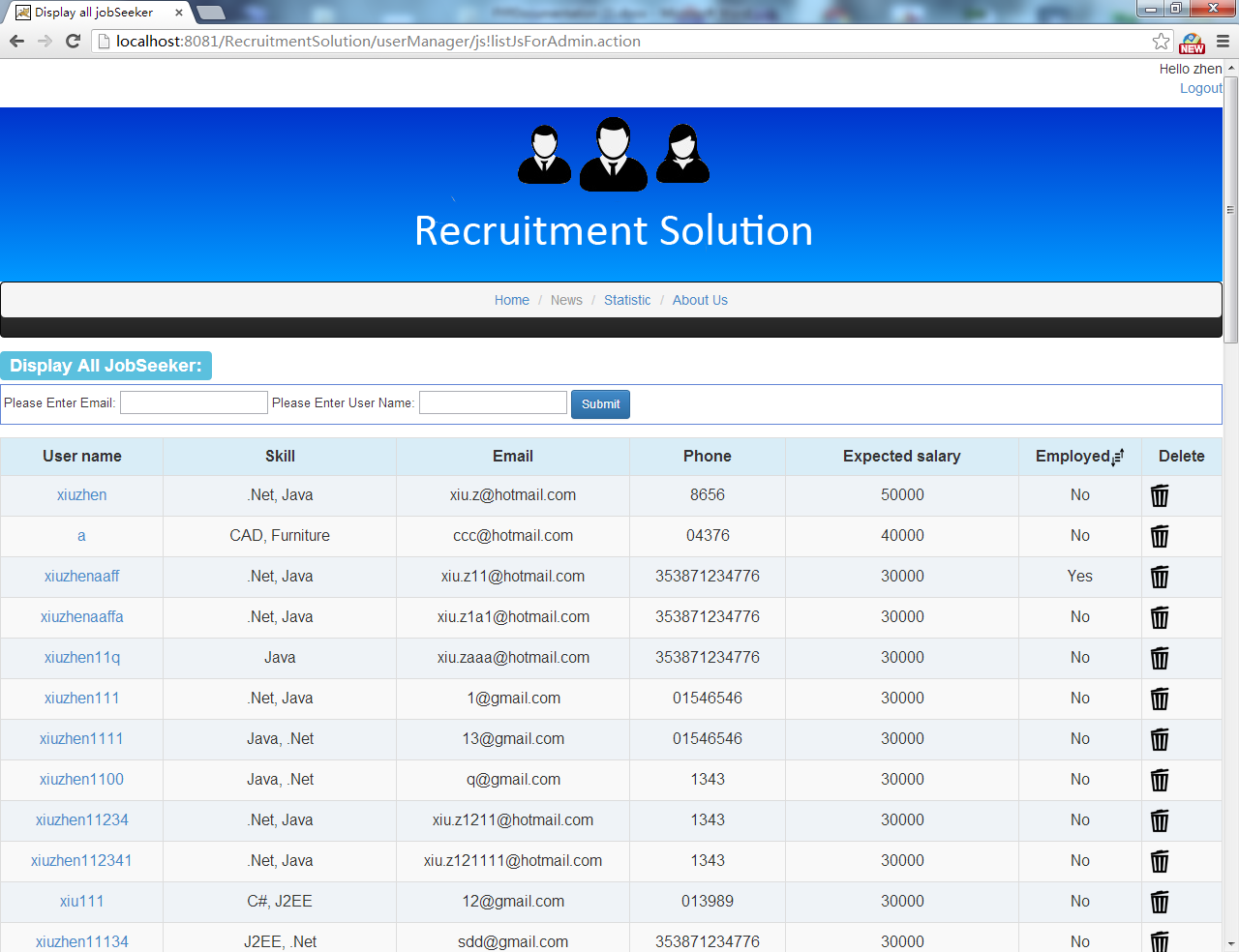
Administrator login page url is <http://localhost:8081/RecruitmentSolution/userManager/>

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

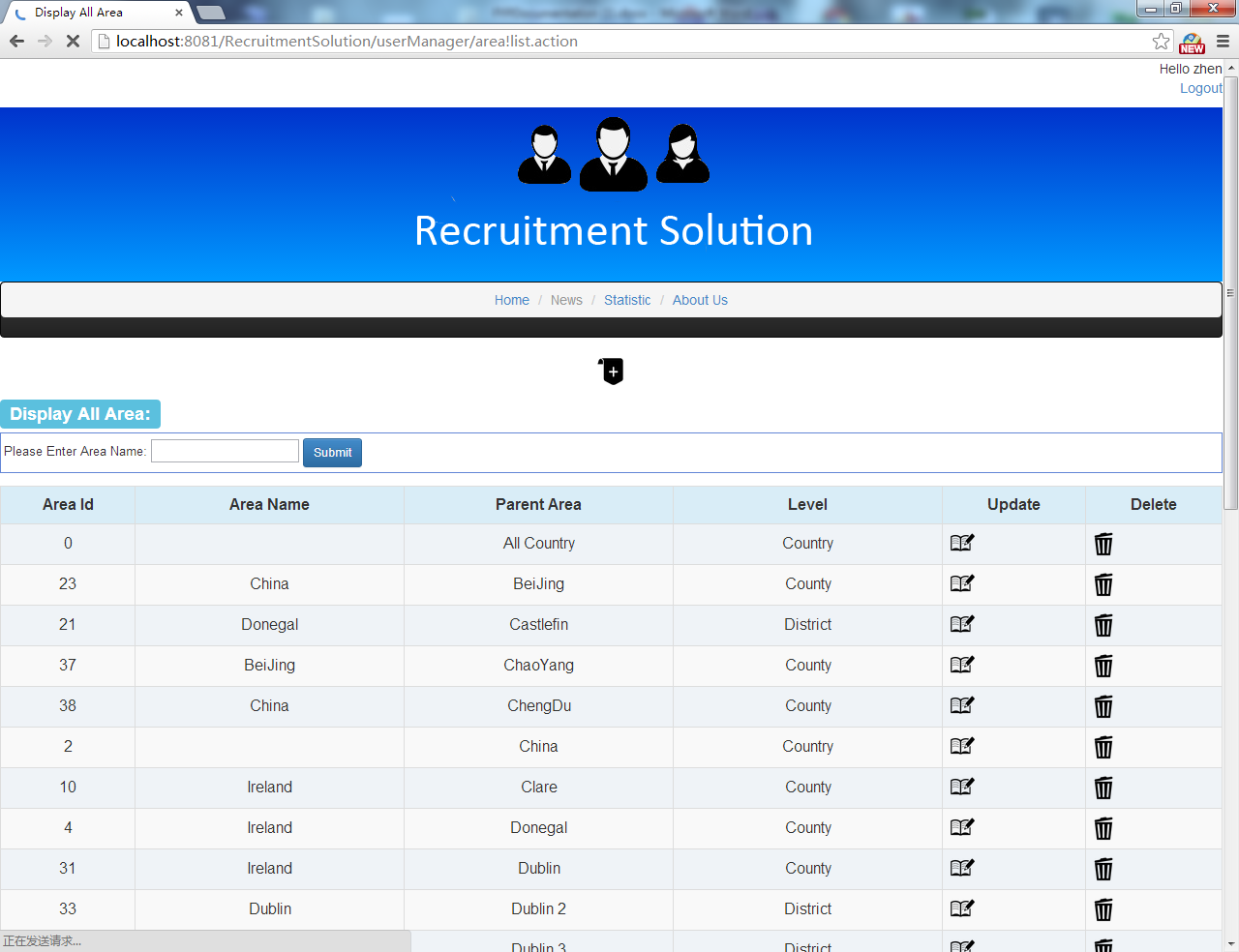




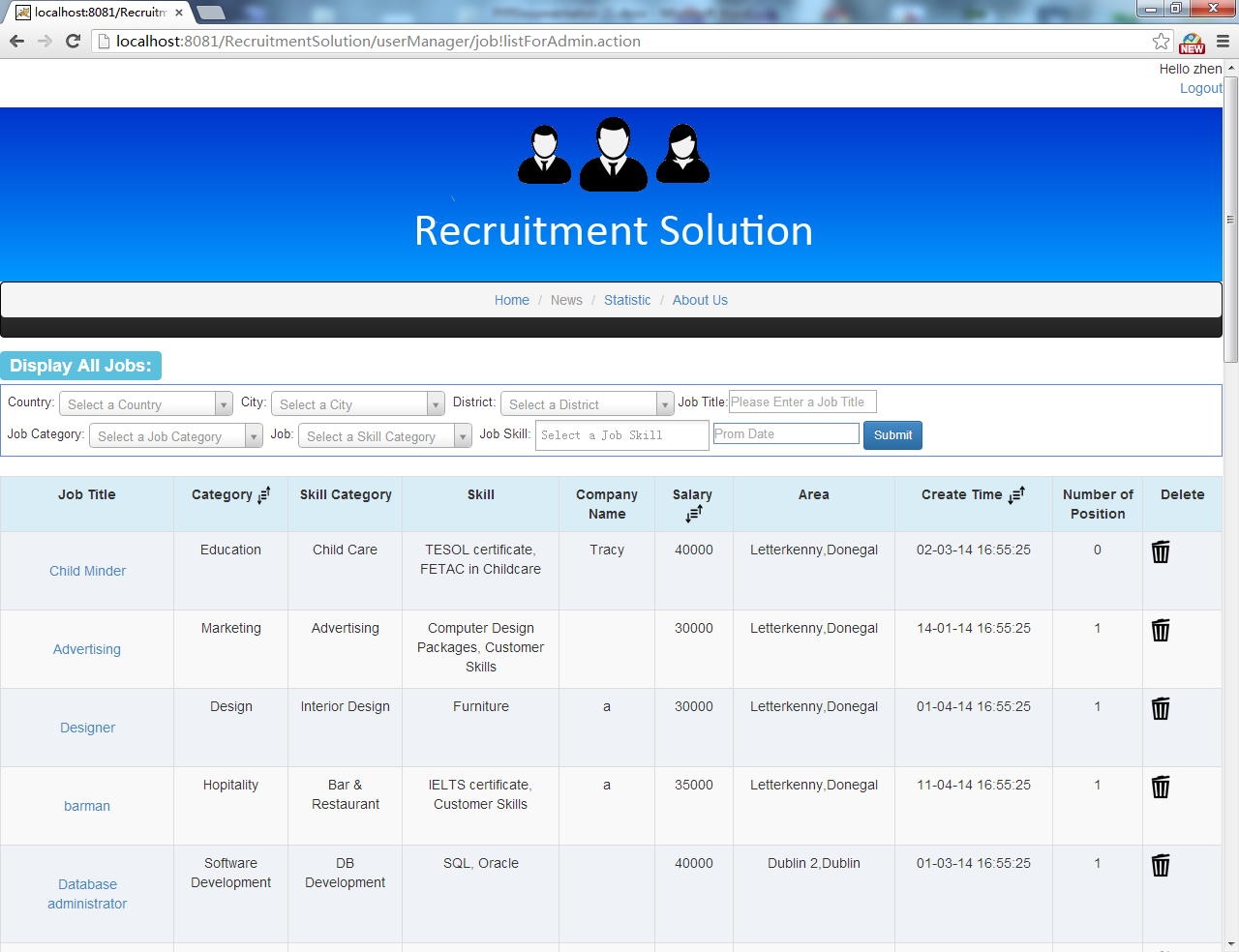
This page displays all the employers that exist in the database. The administrator has the right to delete the employer if data is out of date or employer has not logged in for a certain period of time, also administrator can search Employer by email or by company name, these searches are a partial search.



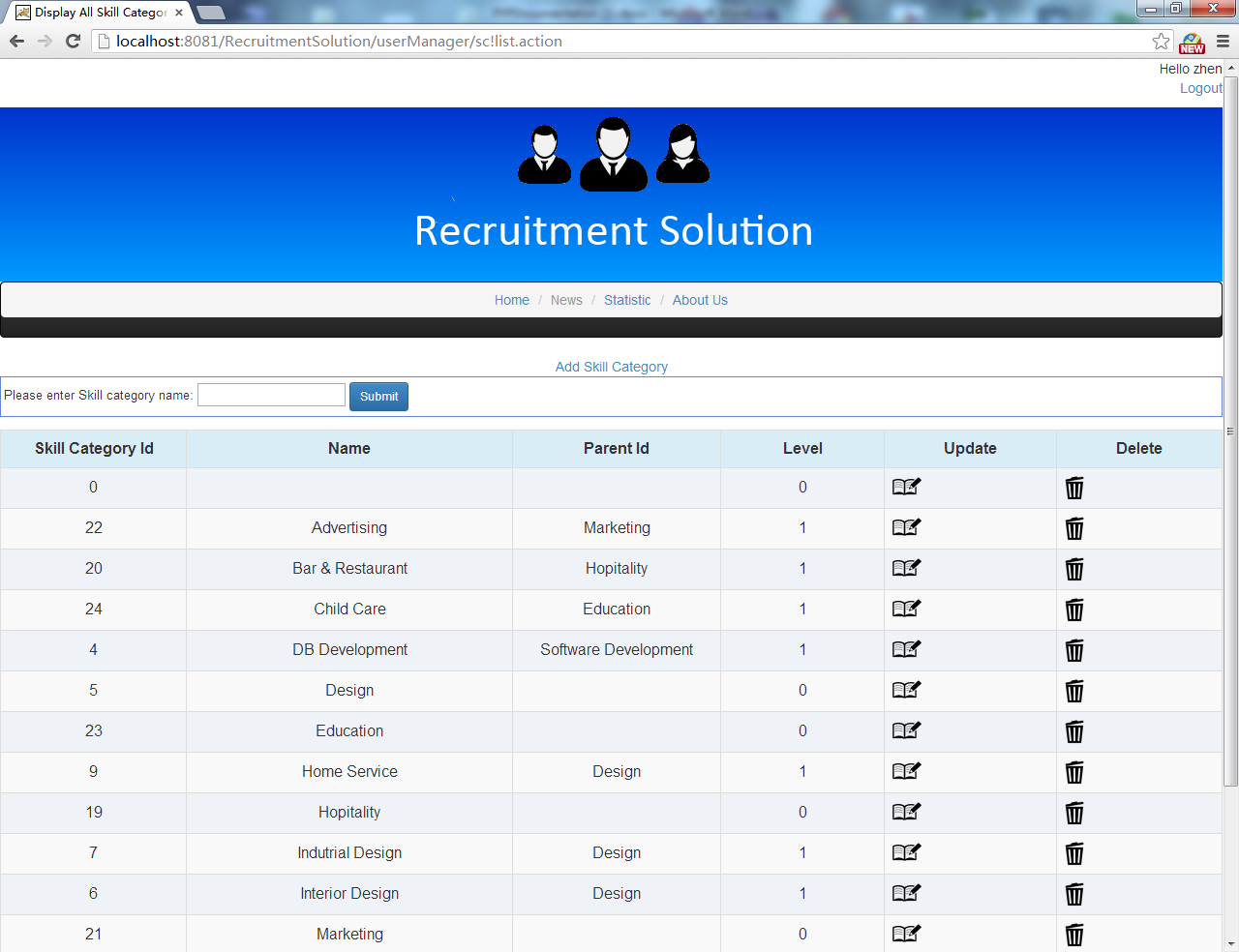
Administrator has the right to search all the Jobseekers in the system with partial search by entering email and username, also sort Jobseekers by employed or unemployed, bin icon for delete the Jobseeker.



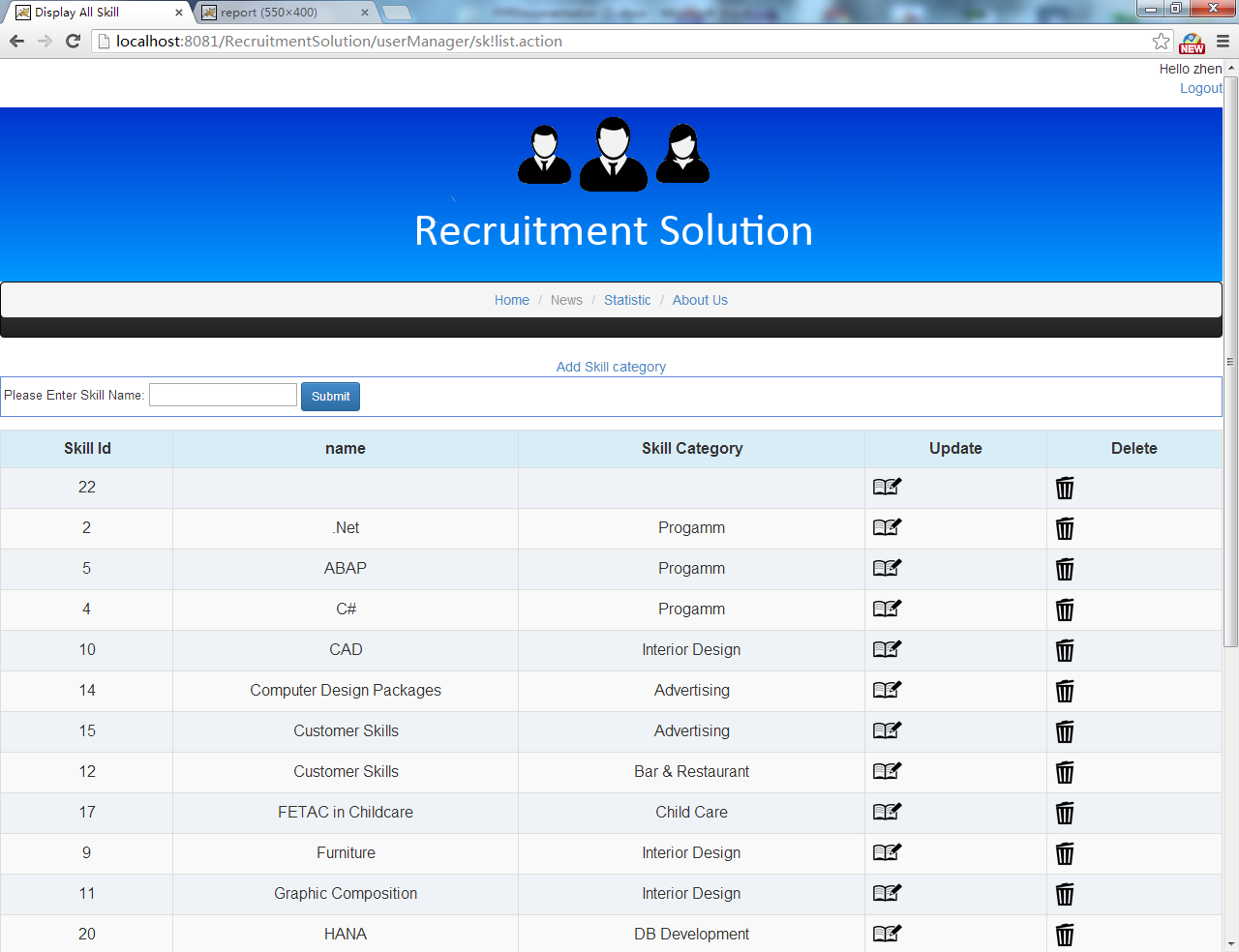
The Area page displays all the area data that is stored in database, with partial search option of search by area name, and update and delete icon for editing the area. Update icon will bring to the page for update this area, add icon in center will allow administrator to add new area into database.



Administrator is able to view all the jobs that are posted in the system with same search way as visitors, sort category, salary, create time and number of position. Administrator has the right to delete the jobs by clicking the delete icon.



Administrator can view the skill categories in the system with the option of search, update, delete and insert skill categories.



Skill is child table of Skill Category, administrator can search, delete, update the skills.

# 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,

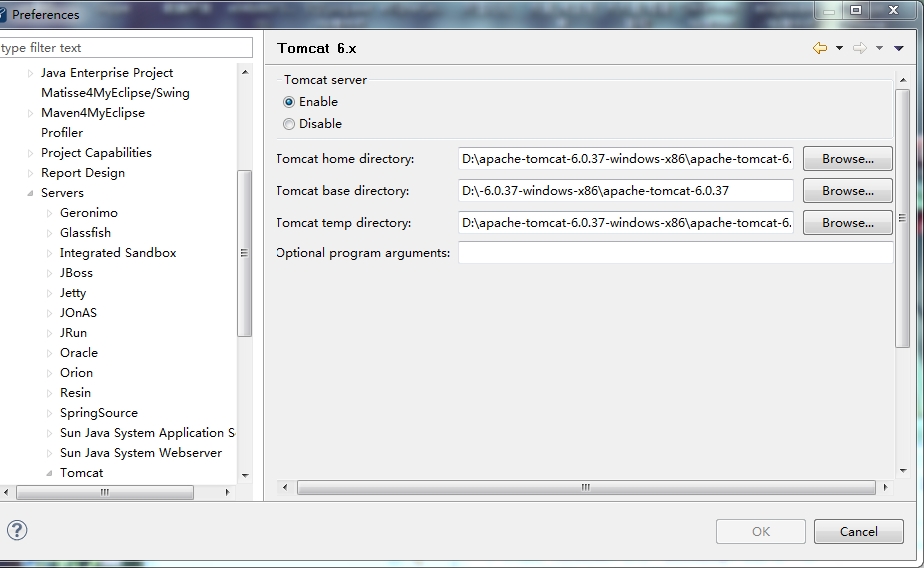
MyEclipse

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

Subscriber：huahua

Subscription Code：uLR8ZO-655444-69678657696224504

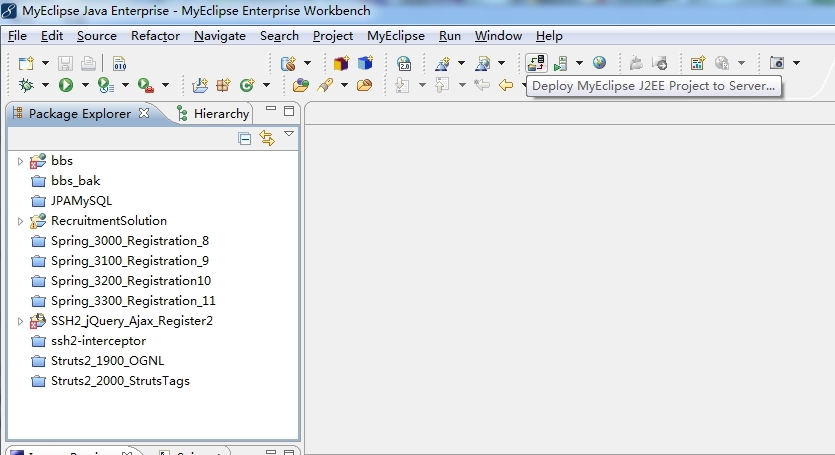
Tell MyEclipse about Tomcat. Select Window, Preferences, MyEclipse, Servers, Tomcat, Tomcat 6. Click "Enable". Select "Tomcat Home Directory" and hit "Browse". Navigate to the Tomcat installation directory Click Apply & OK.



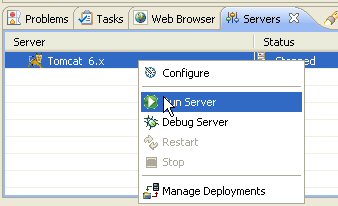
Also click on the JDK sub-entry under Tomcat and make sure this matches the main Java version you are using. In particular, if you are using Java 7 but the Tomcat JDK refers to Java 6, you will get illegal class file errors when executing

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**

# Chapter 7 Test Plan

**Application to be Tested: Recruitment Solution**

**Tester Name: Xiuzhen Chen**

**Test Environment Details: Java**

| Scenario # | Scenario Description | Pre  Condition | Test Data | Test Conditions/Steps | Results | Pass/Fail (Y/N) |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Check Jobseeker Login  Functionality | Web server must be running | **email: xiu.z@hotmail.com**  **Password:xiuzhen** | Start web server enter email and password, click login button | Login success, jobs that matches with jobseeker’s skills appears, if any. | Pass |
| 2 | Upload CV | Web server must be running, , Login success, | **Cover Letter: Dear Sir Madam, I am looking for a job that in Java role.**  **Choose File: cv.doc** | Click manage CV in navigation then click add icon, enter test data, then submit. | Upload success, CV that uploaded, return to list CV page | Pass |
| 3 | Update CV | Web server must be running, , Login success,  Clicked manage CV in navigation |  | Click update icon to update CV,  Cover letter: Dear Hiring Manager, I am very interested in this position  Choose File: CV2.doc | Update success, CV that updated, return to list CV page, | Pass |
| 4 | Deleted CV | Web server must be running, , Login success, |  | Click manage CV in navigation then click the bin Button to delete 1\_CV.doc | Delete success, return to list CV page | Pass |
| 5 | Apply Job | Web server must be running, Login success,  Click Home Button, click a job that interested in with Apply button as able to click | **Select CV that uploaded in system or upload new one** | In main page, click a job title with a href link, select a CV  Click submit to apply a job | Apply success, return to main page | Pass |
| 6 | View Interview History | Web server must be running, Login success, | **Click Interview History in navigation bar** | Click Interview History in navigation bar | List all the interview s that Jobseeker has. | Pass |
| 7 | View Jobseeker profile | Web server must be running, Login success, | **Click profile href link in the top right corner** | Click profile href link in the top right corner | Jobseeker’s profile appears | Pass |
| 8 | Update Jobseeker profile | Web server must be running, Login success,  Data entered must not have red text as invalid beside it | **Change Phone number**: 0871234776  **Choose file**: cat.png | Click submit button, return to main page | Update success, return to main page | P ass |
| 9 | Update password | Web server must be running, Login success,  Old password, matches with the one stored in database. Two password entered should match | **Current Password:** xiuzhen  **New Password:**  zhen  **Confirm Password:**  zhen | Click submit button | Update success, close window, return to Jobseeker profile | Pass |
| 10 | Search Job | Web server must be running, Login success,  In main page |  | Click main page, select Ireland, click submit | Search success, show all the jobs that in Ireland with matched skills in best match and possible match order | Pass |
| 11 | Logout | Web server must be running, Login success, |  | Click logout href in upper right corner to logout | Logout success, return to the pages shows all the jobs that in system | Pass |
| 12 | Check Employer Login  Functionality | Web server must be running, | **Email:** b@gmail.com  **Password:** bing | Start web server enter email and password, click login button | Login success, jobs that matches with jobseeker’s skills appears, if any. | Pass |
| 13 | Update password | Web server must be running, Login success, Old password, matches with the one stored in database. Two password entered should match | **Current Password:** bing  New Password:  bing1  **Confirm Password:**  Bing1 | Click submit button | Update success, close window, return to Employer profile | Pass |
| 14 | Post job | Web server must be running, Login success | **Job Title**: Java Developer  **Job Description**: a fantastic jobs that in Dublin with great experiences in multinational environment.  **Address:** UpperO'Connell St, Dublin  **Phone**: 013547897  **Number Of Position**: 3  **Requirement**: minimum two years experiences  **Salary**: 40000  **Area**: Ireland  Dublin  Dublin City Center  **Job Category:** Software Development  **Skills**: Java, J2EE | Click manage job in navigation, click add icon | Add success, close window, return to main page. | Pass |
| 15 | Update Job | Web server must be running, Login success | **Number of position**: 4  **Phone**: 0871234776 | Click manage job in navigation, click update icon, for Java Developer the one that I posted in last test | Update success, return to main page | Pass |
| 16 | Delete Job | Web server must be running, Login success |  | Click manage job in navigation, click delete button to delete the job that posted as listed in last one | Delete success, return to the main page | Pass |
| 17 | Download CV | Web server must be running, Login success |  | Click manage job, then click a href link of Applied of Data administrator, click View of first jobseeker, download the a href link of CV | Download success | Pass |
| 18 | Send Interview Email | Web server must be running, Login success | **Contact phone:** 0134978  **Interview Time:** 05/05/2014 10:30  **Memo:** Dear Candidates | Click Sent | Send success, close window and return to main page | Pass |
| 19 | Update interview result | Web server must be running, Login success | Click update as fail | Click interview email in navigation, click update for update | Update success, email sent | Pass |
| 20 | Delete interview reslut | Web server must be running, Login success |  | Click bin icon for delete | Delete success, stay in same page with remove of that record | Pass |
| 20 | Update Employer profile | Web server must be running, Login success, | **Change Phone number**: 0871234776  **Choose file**: cat.png | Click submit button, return to main page | Update success, return to main page | Pass |
| 21 | Search Jobseeker | Web server must be running, Login success, | **Email**: xiu | Click submit | Search success, list all the jobseekers who’s mail contain ‘xiu’ | Pass |
| 22 | Check Admin Login  Functionality | Web server must be running, | **Username**: zhen  **Password**: zhen | Start web server enter email and password, click login button | Login success, lists of a href links that appears in center | Pass |
| 23 | View Employer data | Web server must be running, Login success, |  | Click a href link of Employer List | All employers record appears in screen | Pass |
| 24 | Delete Employer | Web server must be running, Login success, |  | Click bin button beside each record will delete the data of Employer | Success page indicates that success deleted, then gp back to the list Employer page. | Pass |
| 25 | Search | Web server must be running, Login success, | **Company Name**: DIT | Click submit button | Shows record of company name is DIT | Pass |
| 26 | View Jobseeker data | Web server must be running, Login success, |  | Click a href link of Jobseeker List | All jobseekers record appears in screen | Pass |
| 27 | Delete Jobseeker | Web server must be running, Login success, |  | Click bin button beside each record will delete the data of Jobseeker | Success page indicates that success deleted, then gp back to the list Jobseeker page. | Pass |
| 28 | Search | Web server must be running, Login success, | **Email**: [xiu.z@hotmail.com](mailto:xiu.z@hotmail.com) | Click submit button | List a jobseeker who is email is [xiu.z@hotmail.com](mailto:xiu.z@hotmail.com) | Pass |
| 29 | View Area Date | Web server must be running, Login success, |  | Click a href link of Area List | All Areas record appears in screen | Pass |
| 30 | Add Area | Web server must be running, Login success, | **Country**: Ireland  **District**: Dublin  **Area Name**: City West | Click submit button | Add success, list areas page | Pass |
| 31 | Update Area | Web server must be running, Login success, | **County**: Ireland  **District**: Dublin  **Area Name**: Dublin 24 | Click submit button | Update success, return to list areas page | Pass |
| 32 | Delete Area | Web server must be running, Login success |  | Click bin icon beside the area that just created | Delete the record stay in same page | Pass |
| 33 | Search Area | Web server must be running, Login success | **Area Name**: Ireland | Click submit button | List areas country is Ireland | Pass |
| 34 | View Jobs data | Web server must be running, Login success |  | Click a href link of Job List | All Jobs record appears in screen | Pass |
| 35 | Delete Job | Web server must be running, Login success |  | Click bin icon beside the job | Success page appeared, then return to the list jobs page | Pass |
| 36 | Search Job | Web server must be running, Login success | **Country**: Ireland  **City**: Dublin | Click submit button | Jobs that in Dublin, Ireland appears | Pass |
| 37 | View Skill Category | Web server must be running, Login success |  | Click List Skill Category List | Show all the skill categories in system | Pass |
| 38 | Delete Skill Category | Web server must be running, Login success |  | Click bin icon beside the Skill Category | Return success page | Pass |
| 39 | Update Skill Category | Web server must be running, Login success | Name: Advertisings | Click update icon beside skill category of 22, change the name, click submit button | Return success page | Pass |
| 40 | Add Skill Category |  | **Name**: Test  **Level**: Job  **Parent** Id: 2 |  | Return success page | Pass |
| 41 | Search Skill Category | Web server must be running, Login success | **Please enter skill category name**: Design | Click submit button in search panel | Return all the skill category with design | Pass |
| 42 | View all skills | Web server must be running, Login success |  | Click List Skills in main page | Return all the Skills in system | Pass |
| 43 | Add Skill | Web server must be running, Login success | **Add Skill**: Test  **Skill Category Id**: 2 | Click add icon in list skill page then, click submit button | Return success page. | Pass |
| 43 | Update Skill | Web server must be running, Login success | **Add Skill**: Testing | Click update icon beside the skill just added, change the data, click submit | Return success page | Pass |
| 44 | Delete Skill | Web server must be running, Login success |  | Click bin icon beside the skill just added. | Deleted success, return success page | Pass |
| 45 | Search Skill | Web server must be running, Login success | **Please enter skill name:** skill | Click submit button | Show all the skills with ‘skill’ | Pass |

# 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 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 functions. 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 a week on compile match job and jobseeker with job category was wasted. Then in area table, firstly, I designed only have one level for area, then I changed that to 3 levels of self joins 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.

I also want to achieve the functionalities of different level of administrators have different authorities to manage this system, but at last, I realized that I have all the methods in the same with Jobseeker and Employer so that make it impossible to achieve it. From all above challenges that I realized that the good plan is very important.

In this project, I gained experience of learning of new technologies and I built knowledge of developing software on J2EE environment. I gained skills of being patient and independent of learning and overcome challenges, more importantly, I realized my capacity to learn and gained confidence in myself, I believe all of these are important in starting my career.

# References

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

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

<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/>

<http://www.jfree.org/jfreechart/>

<http://www.layoutit.com/build>