Advanced Challenges in Web Technologies

Job Application Board

Report

Andrew Altwasser, Daniel Searle and Steve Skidmore

Table of Contents

Stephen Skidmore:

Introduction

The aim of this coursework was in groups to develop a Web application of the group’s choice that demonstrates our understanding of the core module learning objectives. The application includes client and server side development, reflecting on topics we have been taught. This report describes the system specification and details the project development.

Group 1 is made up of three students Daniel Searle, Andrew Altwasser and Stephen Skidmore. The functionality chosen for the system was Rich client experience, usage of Spring MVC framework, Security and RESTful services.

Aims & Objectives

The aim of the project was to produce a web application based on a job application board system. The stakeholders of the system would be Companies and employers who are offering the jobs and potential employees which are the main users of the system who are looking for a job.

The following are the requirements for the system.

- The system has to be secure and requires users to login
- To have pre programmed job examples.
- The user is able to search for a job.
- The system provides accurate information about the selected job to the user.
- The system allows the user to apply for a selected job.
- Companies can add themselves to the system.
- Companies can add jobs to the existing job list.

System Architecture

The systems architecture is split into several layers, these layers are:

- Presentation Layer
- Business Logic Layer
- Database Layer

Presentation Layer

The presentation layer handles all the JSP views and any other data that relates to the presentation including CSS files, JavaScript and images. In figure 1 is a Use case diagram of the job board system.

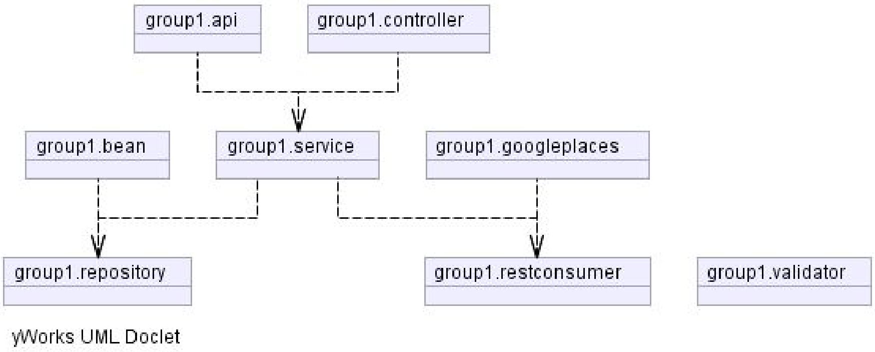
Business Logic Layer

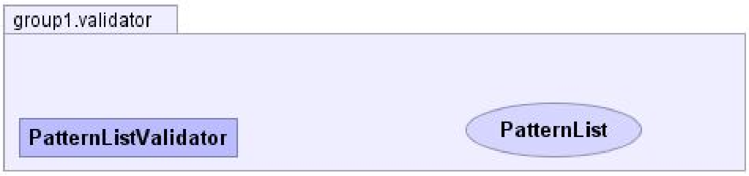
The business logic layer handles all the functions between the database layer and the presentation layer. In the web application these are handled in multiple services such as the JobListingService, LocationService and SearchService.

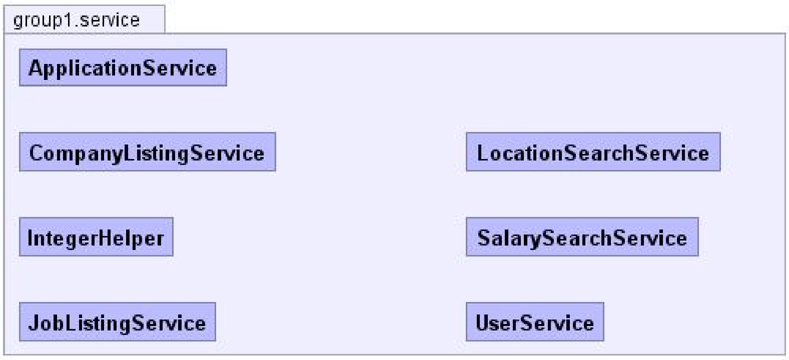
Database Layer

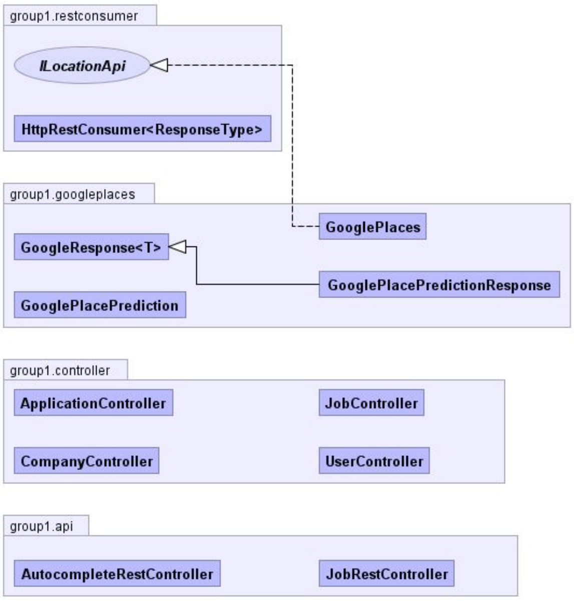
The database layer stores all the data objects used in the web application such as the Job and Company beans.

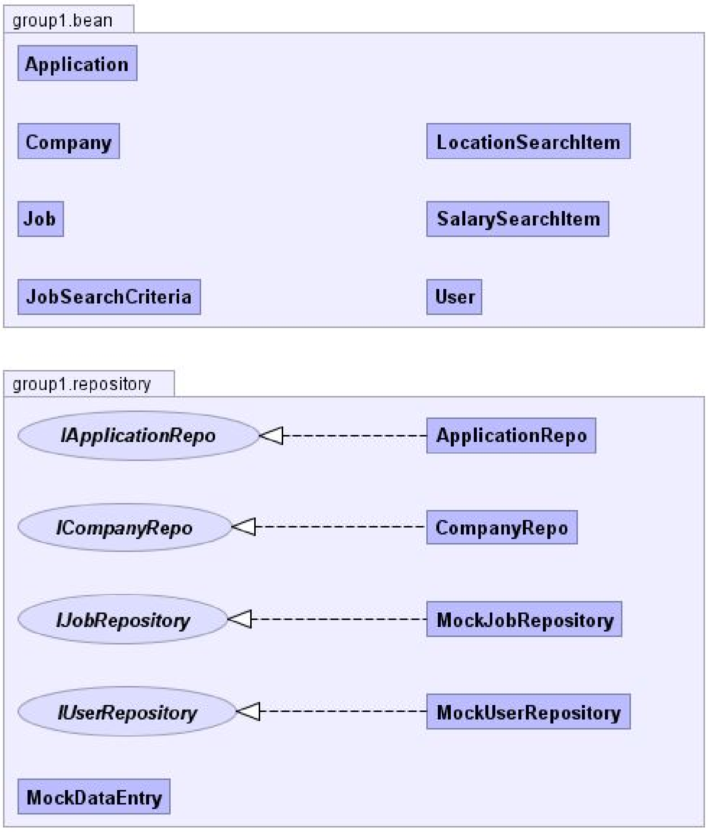
Class Diagram











The diagram above shows the class diagram with each package. The project was split into separate packages for example the controller package which stores all the different controllers that are used. We had many different types of data which is stored in the repository package, for example we have jobs, users, companies and applications. We also use the google api for the map feature which is stored in the googleplaces package.

Use Case Diagram

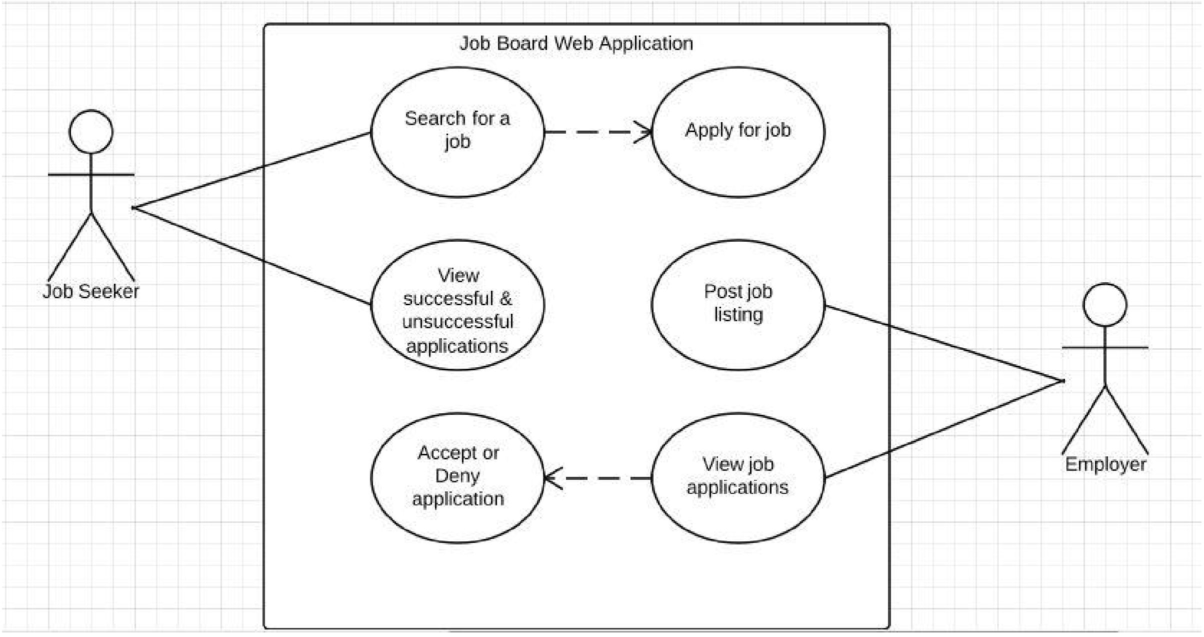

Job Board Web Application
Search for a job
Apply for job
View successful & unsuccessful applications
Post job listing
Accept or Deny application
View job applications
Job Seeker
Employer

Figure 1: Use Case Diagram

Figure 1 shows the use case diagram for the job board web application. As you can see there are two actors the Job Seeker and the Employer. The top level use cases for an employer are that they can post job listings and also view job applications from job seekers. Jobseekers can search for a job and also apply for the selected job, as well as applying for jobs they can also view successful and unsuccessful applications.

Behavioural View Diagrams

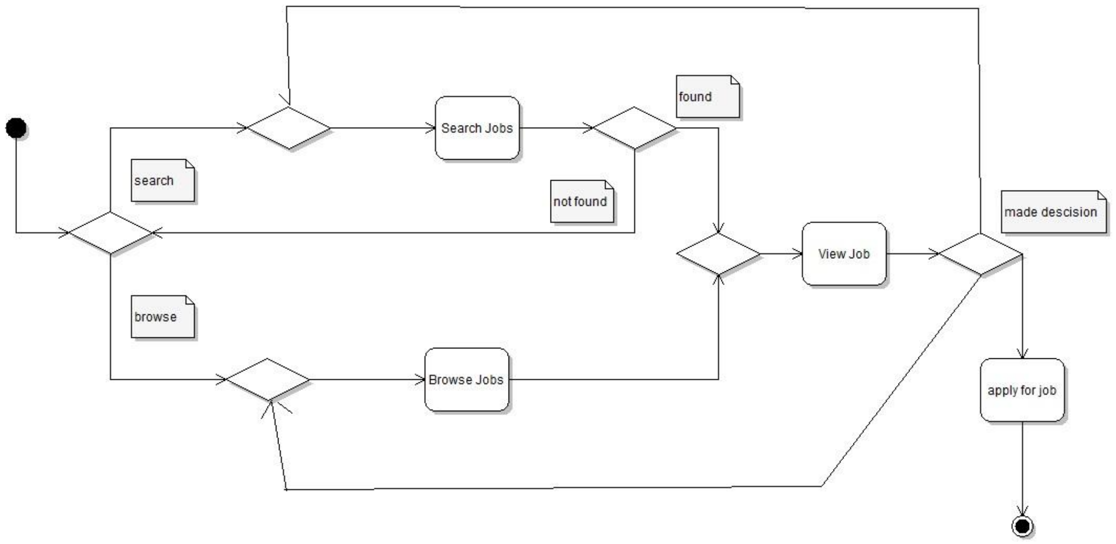


Figure 2: Activity Diagram Apply for Job

Figure 2 shows an activity diagram of one of the most important function of the system which is applying for a job. The scenario starts with the users deciding whether to browse the list of jobs or to search for a specific job. If the user searches for a specific job they may search a company or a certain role. If there are any jobs meeting the search criteria then the search will view the job otherwise the user will have to search again or browse the list. If the user chooses to browse the list then they can view any of the jobs listed. They then can decide whether to apply for the selected job are start all over again looking.

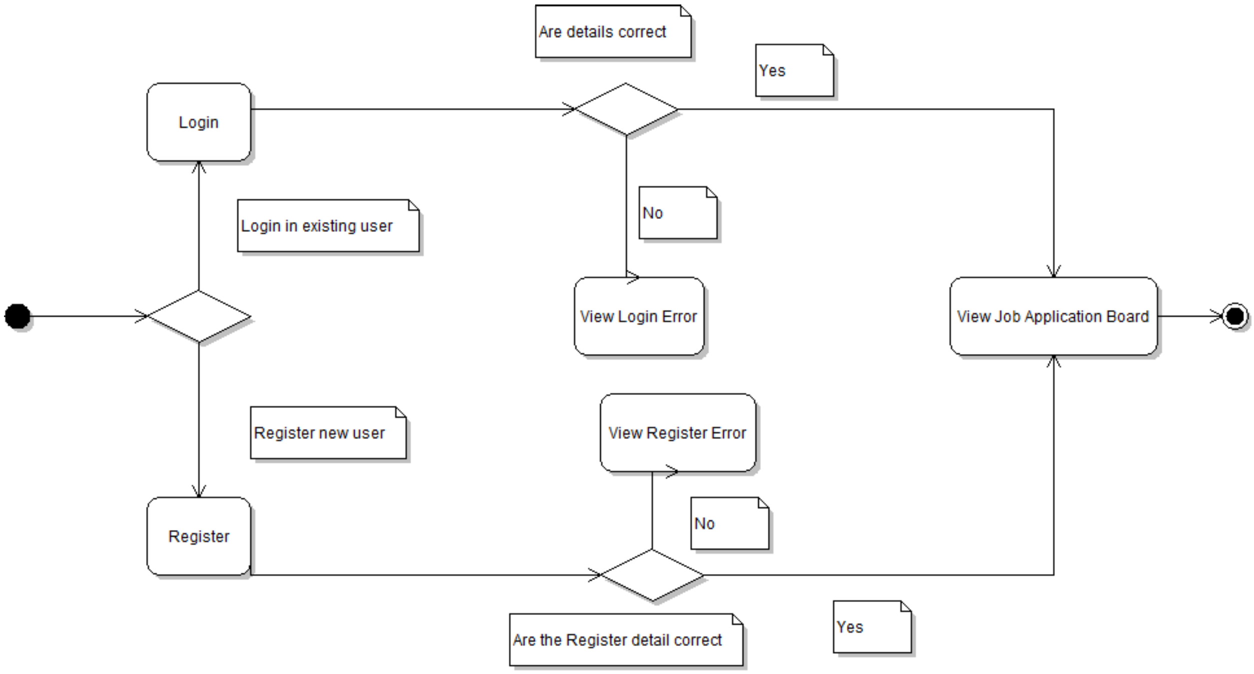


Figure 3: Activity Diagram Login

Figure 3 shows an activity diagram for login into the application. The scenario starts with the user Either Login or Registering based on whether to user has an account. If the user decides to login then they are required to supply their username and password. If any of the details are incorrect the user will be sent to the login failed page and have to login again. If the login is a success the user will be allowed access to the job board. This is the same procedure for registering a user.

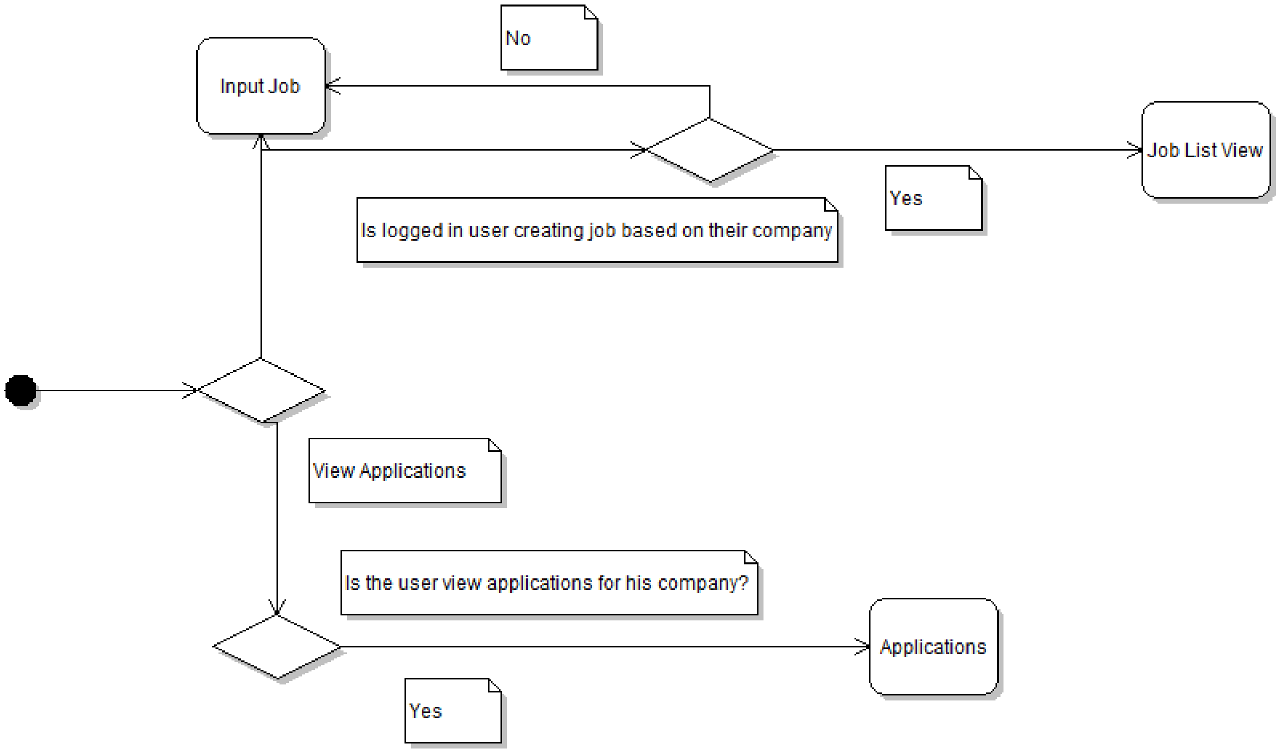

No
Input Job
Job List View
Yes
Is logged in user creating job based on their company
View Applications
Is the user view applications for his company?
Applications
Yes

Figure 4: Activity Diagram Job Input

Figure 4 shows an activity diagram for the input of a job. The scenario starts with the user either view existing job applications or inputting a job. If the user decides to input a job they can only add jobs for companies that they are associated with. If they try to create a job for another company the form will not be submitted. This is the same for view job applications, the user may only view job applications for their company.

Deployment Manual

No deployment specific deployment requirements the project is a self contained Netbeans project with no external dependencies.

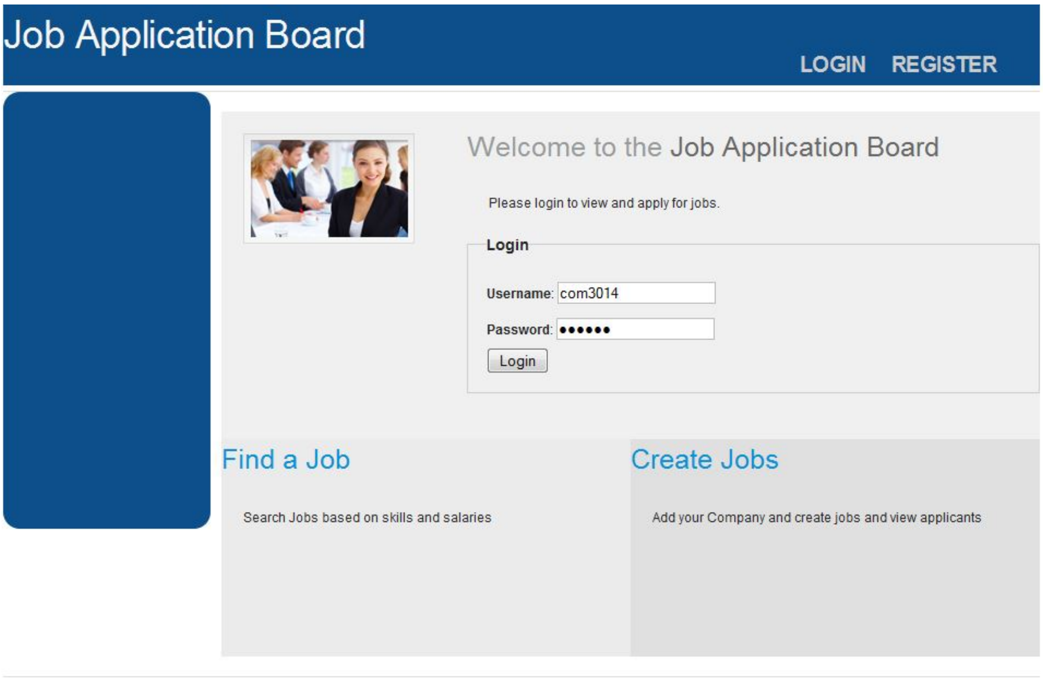
User Manual

This manual has been written keeping in mind the end- user, who uses the web application and which will act as a point of reference. This guide presents brief instructions for use by Users of the system.

Logging into the Job Application Board

You must have a valid Account to access the Job Application. You will not be able to access the application unless you have a valid account.

On the login screen your should see the image below.



Welcome to the Job Application Board

Please login to view and apply for jobs.

Username: com3014

Password: ••••••

Search Jobs based on skills and salaries

Add your Company and create jobs and view applicants

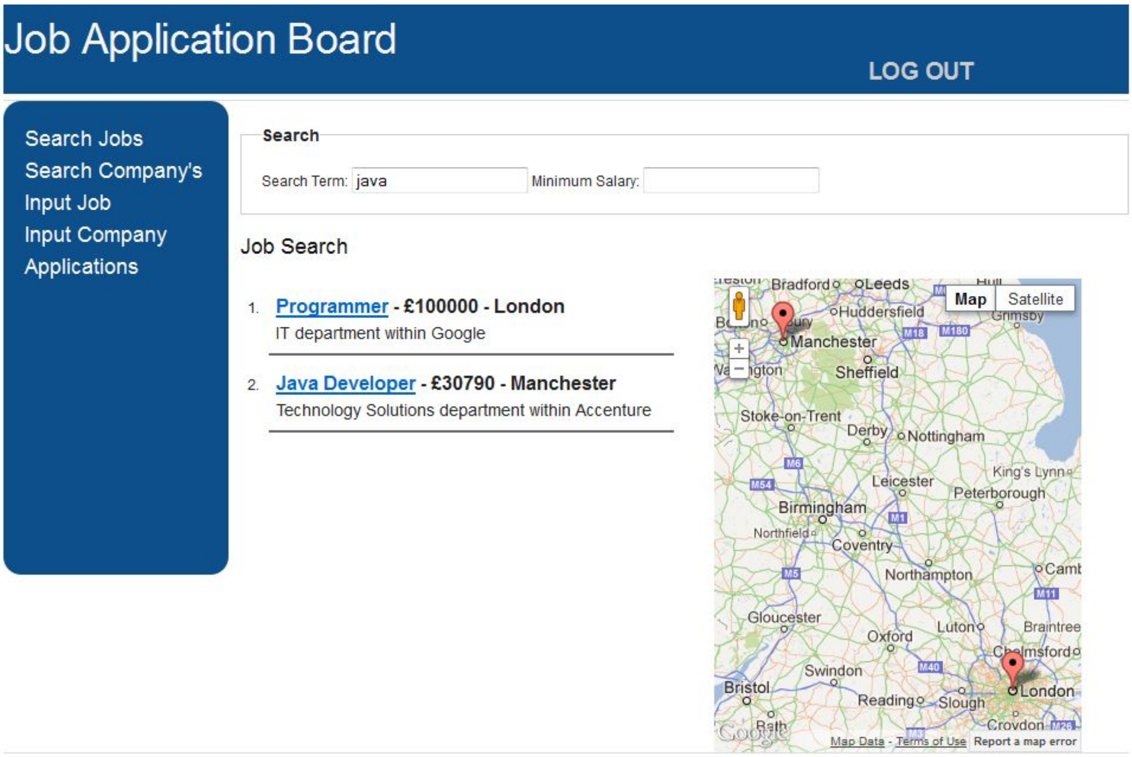
Enter you username and password and click submit to enter the application. If you have entered a valid username and password you will be taken to the job board. If the information entered is invalid you will be shown the following page. If this happens please double check you have typed your details in correctly being aware of caps lock and attempt to type your information again.

The default username is com3014 and password is 123456.

If you do not have an account then please create one by selecting register, when registering an account you are required to provide a username and password. In addition to this you are also required to select whether you are a company or a user. If you select company than once you have registered you will be required to input your company details.

The Job List

The job list is the initial page shown immediately after logging into the application.



Job Application Board

Search

Search Term: java Minimum Salary:

Job Search

1. Programmer - £100000 - London
   IT department within Google
2. Java Developer - £30790 - Manchester
   Technology Solutions department within Accenture

The job list screen lists the current jobs that the logged-in user can apply for. If the list is empty then there are no current jobs available at that time. The user can search for a job then scroll through all the possible jobs and click on ones that interest them by clicking on the jobs name, this will take them to a page specifically about the chosen job. As well as the list of jobs there is also a map showing the location of the jobs.

The Job Screen

The job screen details all the information of the selected job and allows the user to apply for the job. The screen shows the job title, company, location, brief description of the role, salary and also required and desired skills.

Applying for a Job

To apply for a job click apply on the job screen then fill out the required detail and submit the form. This application will then be sent to the applications list.

Navigating through the Job Application

Accessing the various areas of the application is accomplished via the navigation frame of the left hand portion of the web browser. Below details the different navigation options.

Adding a Job in the Job Application

Once a company has registered and added themselves to the company list they can begin to create jobs that users can apply to. The job form is very simple to fill in and is validated to ensure that mobile number and emails are correct. The user is notified if there are any problems with any of the inputs.

Rest API

The API provided uses JSON as the text transport to transfer entities and all api calls are made under the /api/ prefix of the web application. Note that all these functions need to be run under a registered user by using the username and password and basic HTTP authentication. Furthermore the URL arguments must be URL encoded.

- /api/job/search/?term={SearchTerm}&salary={MinimumSalary}
  - SearchTerm: String term which should be matched to any part of the returned job. This term is required to be non-null otherwise a blank list is returned.
  - MinimumSalary: Integer defining the minimum salary of the jobs to return. Can be null to return all jobs which match the search criteria.
  - Note that both GET arguments need to be provided otherwise the call will fail.
  - Returns: JSON object with a single element named “iterable” which contains a list of jobs which match the criteria each job contains the associated company information.
- /api/autocomplete/company/{CompanyName}
  - CompanyName: Start of the company name to search for.
  - Returns: JSON object with a single element named “iterable” with a list of companies where their name starts with CompanyName.
- /api/autocomplete/location/{Location}
  - Location: Text string of a entry to auto-complete.
  - Returns: JSON object with a single element named “iterable” with a list of suggested auto complete matches for the entered location, using the Google Places API.

- /api/autocomplete/salary/{Salary}
    - Salary: Text to try and extrapolate to a salary.
    - Returns JSON object with a single element named “iterable” with a list of suggested salaries for the given input. Mainly used internally for the search auto complete functionality.

Group Members Roles

Group 1 contained three members and the project was split into task for each member to complete. This section describes the role of each group member and what they contributed to the project.

Daniel Searle:

Main project overview and reviewer as well as creating the REST API and dynamic functionality using javascript throughout the forms. Important dynamic features include the job search which uses Google maps integration to display the geographical location of jobs which matched the search criteria. Further dynamic elements included adding location entry suggestions using the auto-complete API provided by Google Places and auto-complete functionality on the company search and job entry forms. I was also responsible for creating the initial start project and layout using the Apache Tiles library.

Andrew Altwasser:

I was responsible for the job seeker view, this meant designing and creating the pages for the list of jobs and the detailed jobs screen. This involved creating the jsp pages and accessing the job information from the Job bean and JobController. I was also responsible for the overall look of the web application using CSS styling and I was responsible for the web applications security using the Spring Security Framework. In addition to this i was responsible for creating the application form and application view.

Finally I was responsible for writing the report.

Stephen Skidmore:

I was responsible for the development of the company view and form inputs for both the job and company. This involved create forms, that feed data into the system and allow user to create companies and jobs on the system (this included uploading images for logos, creating multiple selection boxes for the current list of companies when creating a job and doing all the background processing and storing of that data). I was also responsible for the company view and screen which shows information about the companies then shows more detailed versions including logos, addresses, telephone and email contacts. As well I also assisted Andrew with the implementation of the user registration form and storing the user data.