



**Unit:
Analysis, Design and Implementation**

**Assignment title:
Job Hound**

December 2015 – Sample Assignment

Important notes

- Please refer to the Assignment Presentation Requirements for advice on how to set out your assignment. These can be found on the NCC Education *Campus*. Click on Policies and Advice in the left-hand menu and look under the Advice section.
- You must read the NCC Education documents 'What is Academic Misconduct? Guidance for Candidates' and 'Avoiding Plagiarism and Collusion: Guidance for Candidates' and ensure that you acknowledge all the sources that you use in your work. These documents are available on *Campus*. Click on Policies and Advice in the left-hand menu and look under the Policies section.
- You **must** complete the '**Statement and Confirmation of Own Work**'. The form is available on *Campus*. Click on Policies and Advice in the left-hand menu and look under the Policies section.
- Please make a note of the recommended word count. You could lose marks if you write 10% more or less than this.
- You must submit a paper copy and digital copy (on disk or similarly acceptable medium). Media containing viruses, or media that cannot be run directly, will result in a fail grade being awarded for this assessment.
- All electronic media will be checked for plagiarism.

Scenario

The Job Hound Employment Agency (JHEA) requires software to manage its ever growing roster of clients and employers. JHEA performs a simple task – it maintains a catalogue of job opportunities with the employers who have registered with them. The catalogue is organised by a set format of requirements.

Clients who register with JHEA provide their employment details. Client details are organised and catalogued in a similar format to employers. As a consequence, employers can then be provided with details of a client who meet their requirements, and a client can be informed the employers who are offering jobs for which they are qualified. Previously, JHEA have completed its work by hand with pen and paper. However, they want to capitalise on upcoming opportunities and have asked you to design and build this software.

The whole of the JHEA hinges on the Standardised Employment Profile, which is a set of data elements that are compared for each search.

Available jobs require the following information to be set:

1. Title
2. Salary per Year
3. Minimum Education (1 for school, 2 for college, 3 for degree, 4 for PhD)
4. Skills required (as a list)
5. Skills desired (as a list)
6. Desired years of employment

All jobs belong to an employer. Employers are required to set the following information:

1. Name
2. Location

Clients are asked to provide the following information:

1. Name
2. Address
3. Telephone Number
4. Desired Location (as a list)
5. Education (1 for school, 2 for college, 3 for degree, 4 for PhD)
6. Desired minimum salary
7. Skills possessed (as a list)
8. Years of Employment
9. Whether currently seeking a new job.

Scenario continues on next page

After opening the system, the user should be able to generate a list 'matches'. These 'matches' are not directly available to either the employer or the client. Instead, the matches are relayed to clients and employers as part of the personal touch of the JHEA. When pressing a button, the system should match clients to jobs and jobs to clients in accordance with the following criteria:

- A client must meet or exceed the education requirement for a job;
- A client must be within 2 years of the desired years of employment of a job;
- A job must meet or exceed the desired minimum salary for a client;
- The employer's location must match one of the client's desired locations;
- A client must have **all** of the required skills of a job;
- A should have at least one of the desired skills of a job;

When calculating matches, the software will generate a text output that match specific jobs to clients, with accompanying names and contact details. From that point on, it is the job of the personnel department of JHEA to discuss the match with clients and employers. As a result, the software has no further work to complete.

If the JHEA agent has successfully negotiated a placement, the job is set as 'filled' and the client is set as no longer seeking a job. Jobs that have been filled and clients that are not currently seeking a job should be omitted from all future searches.

This is a long-term system, and subsequently, a robust system of loading and saving user data **must** implemented.

Your application **must** also provide the following functionality:

- Allows for new clients to be registered.
- Allows for all existing clients to be listed.
- Allows for new employers to be registered;
- Allows for new jobs to be allocated to employers;
- Allows for jobs to be marked as 'filled' and removed from active searches;
- Allows for clients to be marked as 'no longer seeking' a job. And omitted from future searches;
- Calculates the applicability of each job to each client;
- Outputs a list of all matching jobs and clients;
- Saves and loads user data.

Your solution will consist of: (1) a class diagram; (2) a use-case diagram; (3) an activity diagram for the process of the user adding a client, employer and job and then matching up clients to jobs; and (5) the completed code in Java.

Tasks are on next page

Task 1 – Candidate Class List and Diagrams (26 Marks)

The candidate class list should incorporate justifications and discussion as to why each class was selected for inclusion, and how its relationship to other classes was derived. The class diagram should show attributes, operations, scope and relationship of classes to each other.

Task 2 – Activity Diagram (25 Marks)

The activity diagram should incorporate the classes involved in a user specifying a computer system.

Task 3 – Use Case Diagrams (8 Marks)

The use case diagram should incorporate each of the user activities indicated in the brief.

Task 4 – Code Architecture (15 Marks)

This involves creating a code architecture that shows an appropriate level of coupling and cohesion, along with the necessary amount of inheritance and encapsulation to express the system.

Task 5 – System Implementations (26 Marks)

This is for implementing the system as described and providing the completed Java code.

Submission requirements

- Your program must be submitted as a zip file of the full project.
 - Whatever IDE you use, it should be possible to open and run the project directly from the extracted archive.
- Diagrams and materials associated with the tasks above should be presented in a word-processed document.
- All references and citations must use the Harvard Style.

Candidate checklist

Please use the following checklist to ensure that your work is ready for submission.

Have you read the NCC Education documents 'What is Academic Misconduct? Guidance for Candidates' and 'Avoiding Plagiarism and Collusion: Guidance for Candidates' and ensured that you have acknowledged all the sources that you have used in your work?

Have you completed the 'Statement and Confirmation of Own Work' form and attached it to your assignment? **You must do this.**

Have you ensured that your work has not gone over or under the recommended word count by more than 10%?

Have you ensured that your work does not contain viruses and can be run directly?