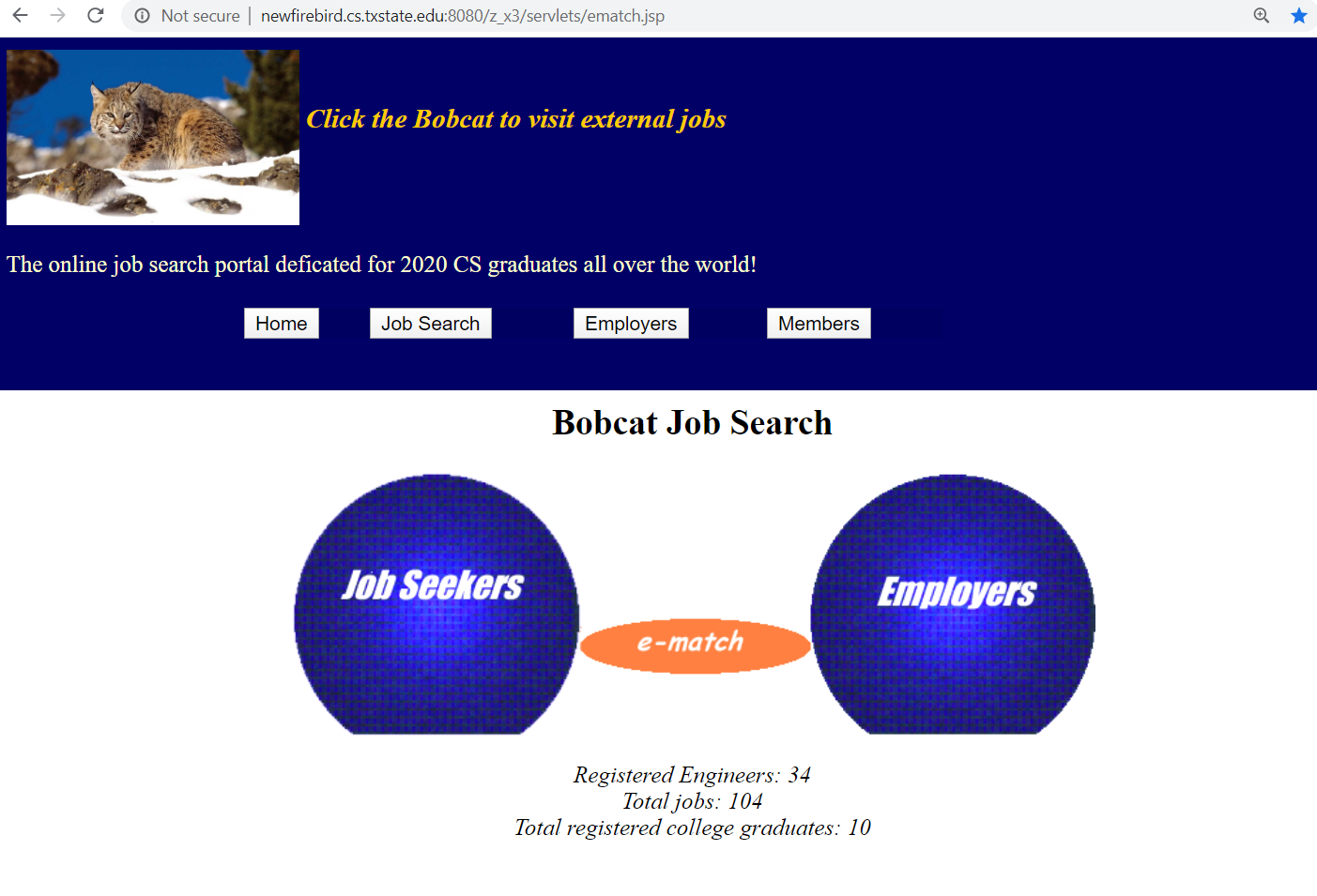
# **ReadMe file for Final project**

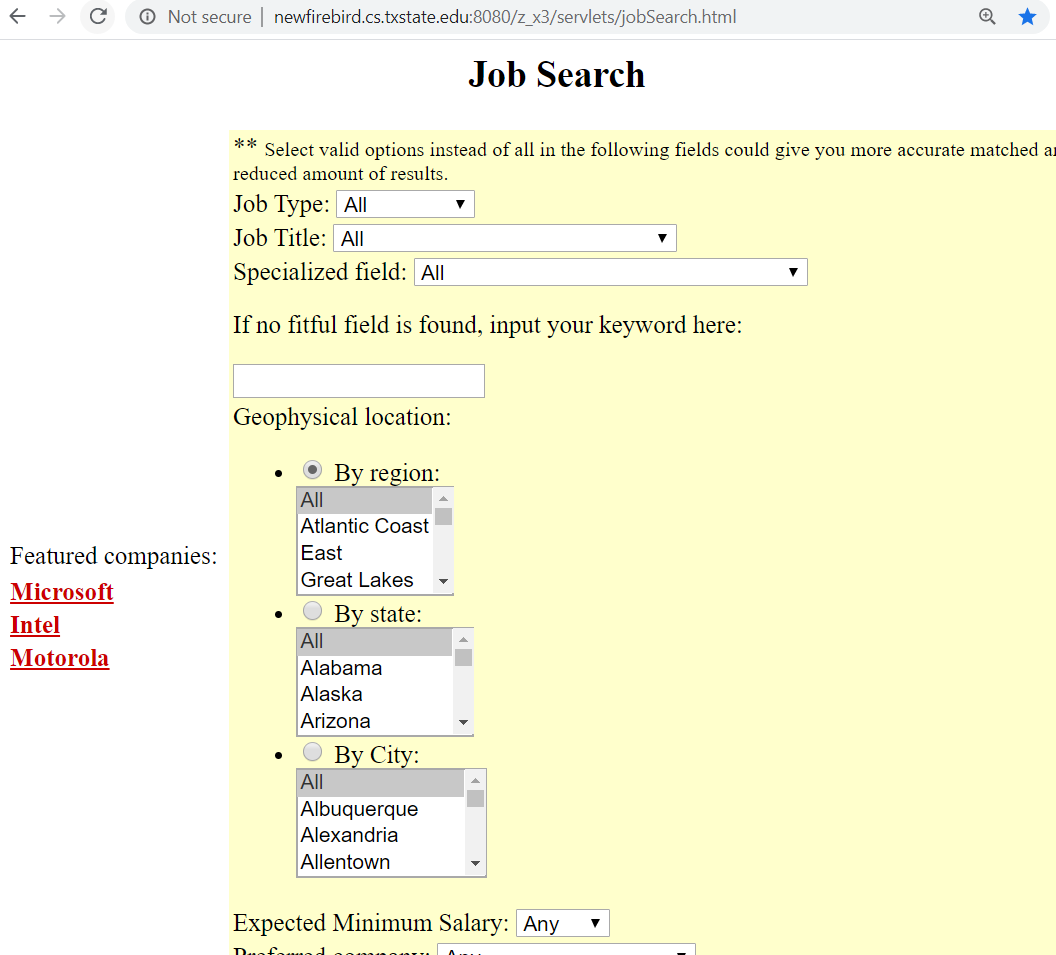
## **Zebo Xiong**

1. I am the only one in this team.

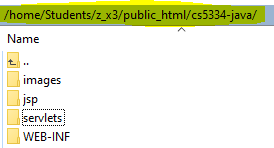
1. Portal: <http://newfirebird.cs.txstate.edu:8080/z_x3/servlets/ematch.jsp>



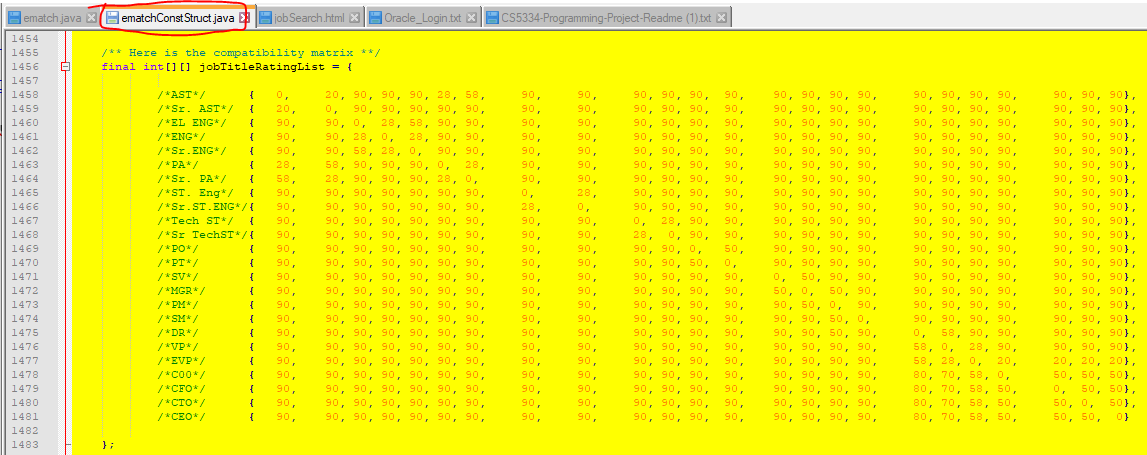
1. Testing URL: <http://newfirebird.cs.txstate.edu:8080/z_x3/servlets/jobSearch.html>



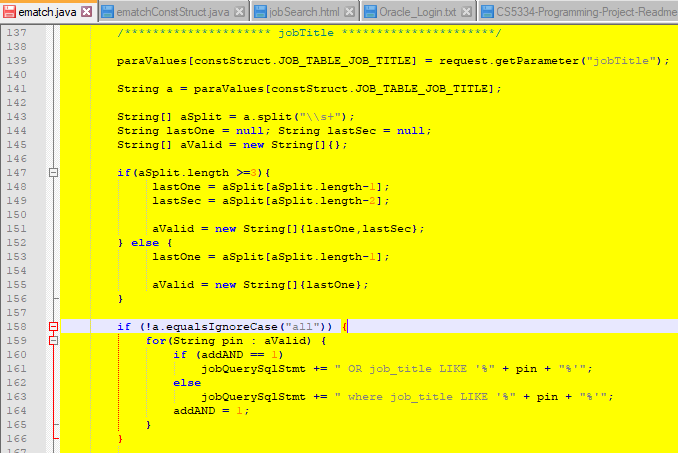
1. Source code location: **/home/Students/z\_x3/public\_html/cs5334-java**



1. Extended function one: rating with “Job Title”
2. In ematchConstStruct.java 🡪 Setup the job title compatibility matrix

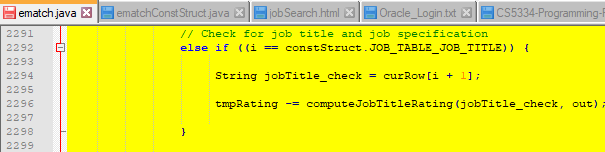


1. In ematch.java 🡪 buildJobQuery method 🡪 Updated the SQL (Select the records related to “Job Title”)

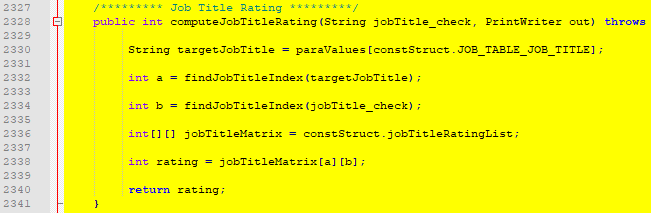


1. In ematch.java 🡪 computeAJobRating method 🡪 incorporate the “Job Title” rating

* Further calculate the rate by calling another function “computeJobTitleRating”



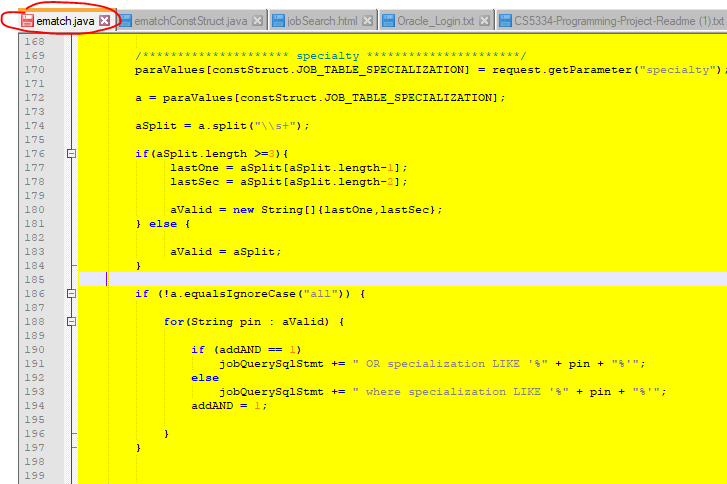
* This will calculate the rate according to the job title relationship matrix. For example, if 100% match, then the rate will comes to 0 or negative. If they are related, the rate will become smaller which means better match.



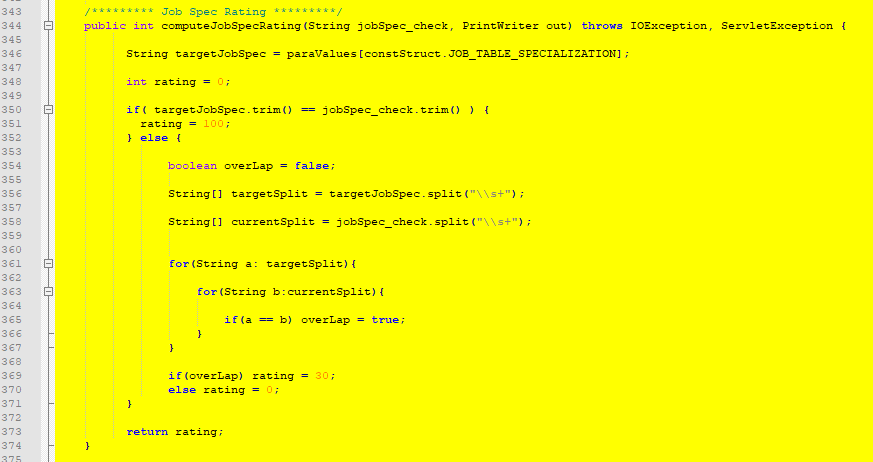
1. Extended function two: rating with “Job specification”
2. In ematchConstStruct.java I Originally I plan to implement the specialityAreaNeighbor matrix but it is too big



1. Later I directly updated the ematch.java

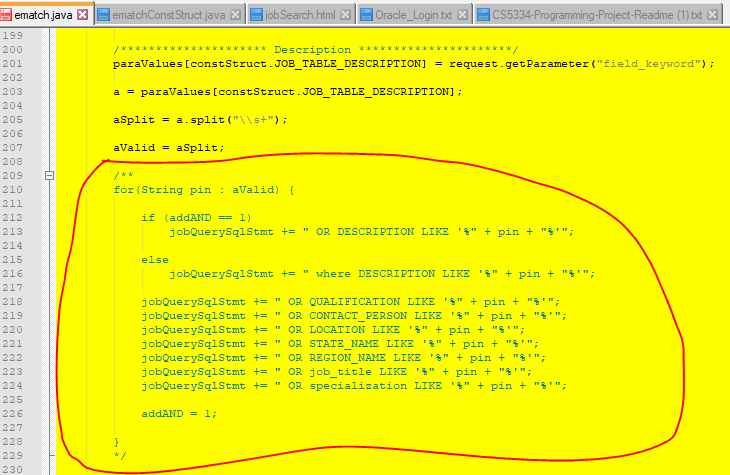


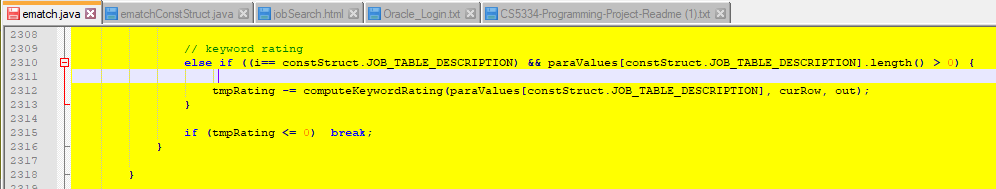
1. Added the rating logic as below:



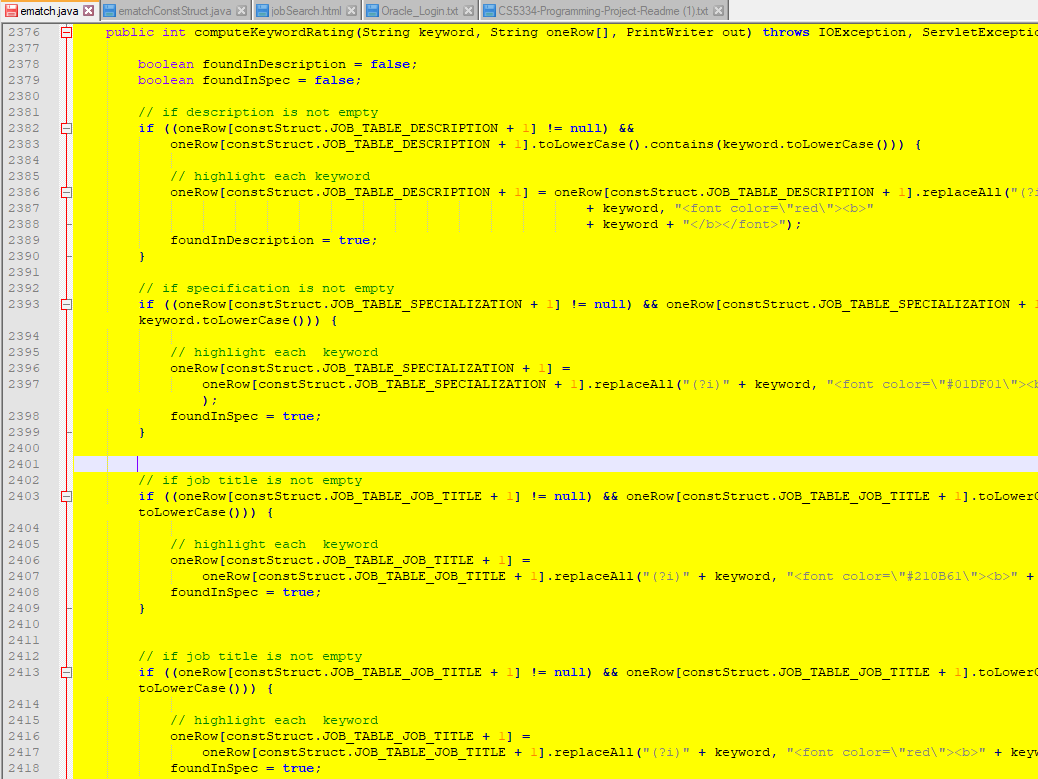
If the areas are the same, 100 score earned. If there are words overlapping, 30 score earned. Otherwise 0 score earned.

1. Extended function three: search by “Key Word”
2. This one bear more logics. I originally decided to filter on SQL level but I found the result is not good.



1. Then I released the SQL conditions and attempt to work on ematch.java 

1. Added the calculation function:



Firstly I check the description. Earn 20 score if found a good match.

Secondly, I will check specification. Earn 55 score if found a good match.

Thirdly, I will scan most of the other fields in the job table. Earn 40 score if found a good match.

1. To compile and test:

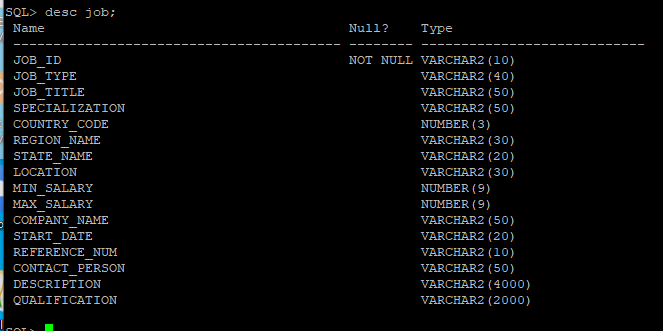
rm -f \*.class

javac \*.java

chmod 744 \*.class

touch web.xml

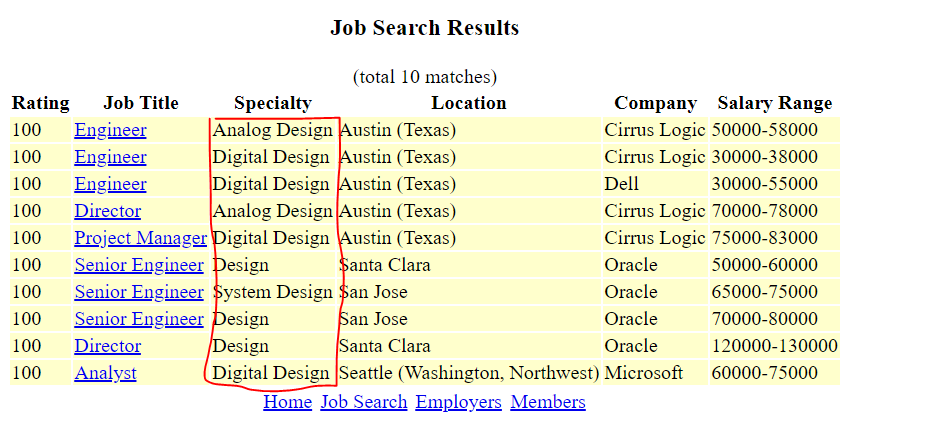
1. The job table:



1. Same test case:
2. Search “Programmer Analyst” on Job Title field



1. Search “Analog Design” on Job Specification filed



1. Search “Development” on Key Word fields:



1. Search “Austin” on Key Words field：

