Job Markets Trend Application Technical Design

(Up to date version: https://docs.google.com/document/d/1dvv37WRrlAl0sXfrrPW4SbblWHOJwqhbNW39IIrV6O4/edit#)

Penn Engineering

Overview

Team Members & Responsibility

1. Jie Bao: Data sourcing + ETL (including web crawler)
2. Zhong (Kevin) Liu: Architecture design, Infrastructure (AWS etc) and API Service
3. Terry Zhang: front-end development
4. Project management: all.

Core Problems

Core problems this application aims to address including:

1. One stop to access information about trending jobs in the markets and more specifically, top demanding skills.
2. Matching users' skills to relevant opportunities, and possibly with notifications.

Technical Design

Architecture

We are trying to use the JAVA knowledge we've learned from the class and beyond to build this application. The application includes multiple layers.

1. **Data Fetche**r: a crawler to fetch jobs data from external data sources. Written by JAVA.
2. **ETL**: a component to Extract, Transform, and Load clean data into storatte. Written by JAVA.
3. **Jobs Data**: use files as storage to store clean job data prepared by the ETL component.
4. **API Service**: a set of restful APIs to provide client application access to jobs data. Written by JAVA.
5. **User Interface**: an interface user can interact to see top demanding skills and jobs recommendation.

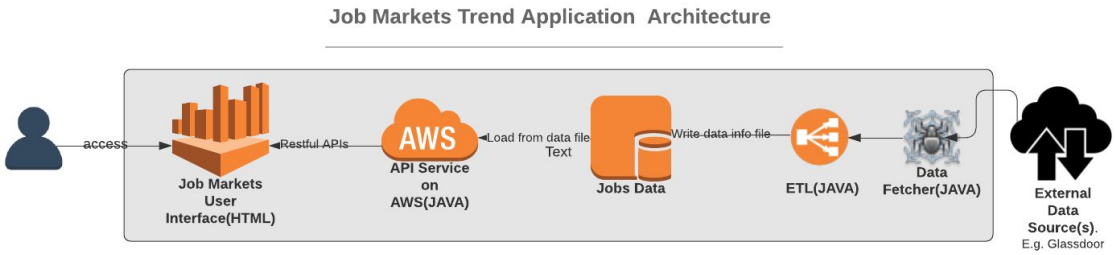


Diagram 1 - Jobs Markets Trend Application Architecture

Detailed Technical Design

Data Fetcher

[@Bao Jie to complete]

1. Java Class 1 - SpiderHelperFunctions.java
    - Make HTTP request
    - Parse the page
    - Search for words in pre-specified dictionary
    - Return all web links in this web page
2. Java Class 2 - Spider.java
    - Read webpage lines related to words in pre-specified dictionary
    - Convert inputStream object to file

ETL

[@Bao Jie to complete] cont'd from Data Fetcher

3. Java Class 3 - JobDataAnalysis.java
    - Group information by job
    - Compare salary with average salary per location
    - Compare skills with job positions close to average salary per location
4. Java Class 4 - JobDataWriter.java
    - Write fields directly fetched from web as well as derived fields into a csv file

Jobs Data File Format

[@Bao Jie to complete]
A csv file with the following fields:

- Job title
- Job description
- Company
- Primary Location
- Required Skills
- Salary Range

- (Derived) Same Location Avg Salary
- (Derived) Required Skills associated with Avg Salary Job Positions

Data file example:
https://github.com/UPenn-CIT599/final-project-team-17-jobs-markets/blob/master/data/jobs%20data.csv

API Service

CRCs for Core Classes

Following tables list some core classes used by API service. Please note that the following talbe doesn't mean to include all classes.

Class	Responsibilities	Collaborators	Notes
Config (Application configuration class)	Jobs Data File Location		Class: com.upenn.cit591.jobmarkets.Config
StreamLambdaHandler (handles HTTP/HTTPs request for RESTFUL APIs)	HTTP(s) requests handling		Class: com.upenn.cit591.jobmarkets.StreamLambdaHandler
JobQuery (a facade class to provide interface to fetch jobs by various query terms)	Find jobs by terms	Jobs	
CSVReader (a generic class to help read data from CSV file)	Read current data row		
	Read next row		
	Read cell value		
Jobs (Represents a collection of jobs)	Filter Jobs by Conditions	Job	Class: com.upenn.cit591.jobmarkets.domain.Jobs
	List Hiring Companies	Hiring Company	
Job (Represents a Job)	Title	Hiring Company	Class: com.upenn.cit591.jobmark

			ets.domain.Job
	Required Skills		
	Salary Range & Check		
	Location		
Company	Name	Job	Class: com.upenn.cit591.jobmarkets.domain.Company
	List jobs		
WordPair (relation of a pair of words)	Calculate similarity of a pair of words		Class: com.upenn.cit591.jobmarkets.libs.WordPair
	Calculate commonality of a pair of words		

Table 1 - API Service CRC design

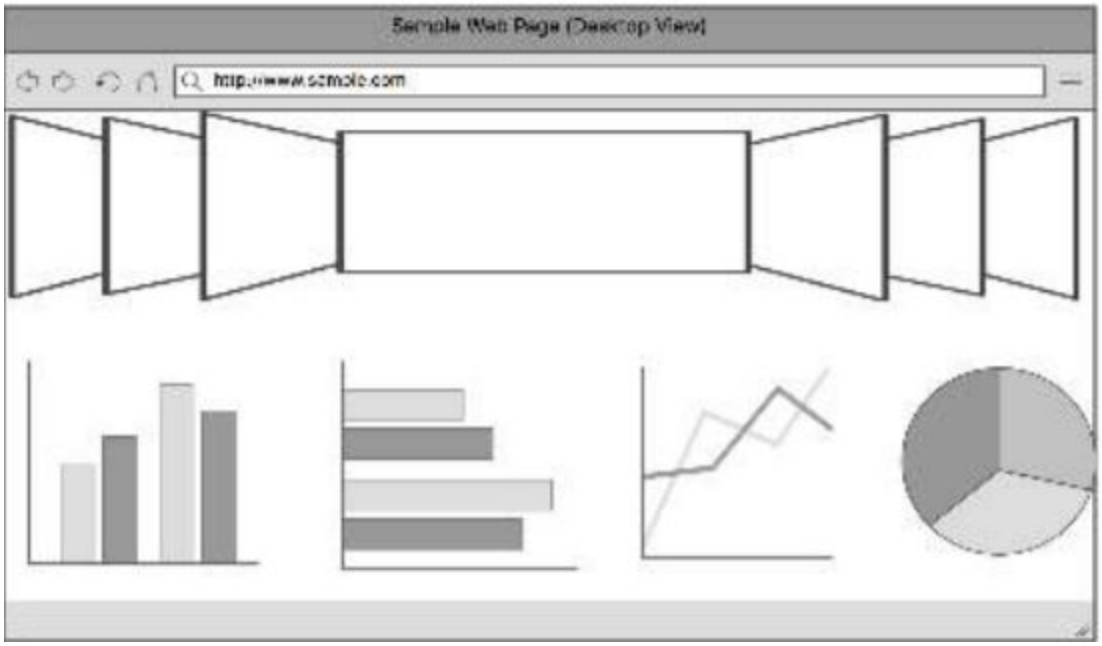
Restful APIs by API Service

[**@Kevin** to complete]

1. **API - Jobs (example, to be updated)**
    a. **Method: http GET**
    b. **URL: https://yrdltjhgh7.execute-api.us-east-1.amazonaws.com/Prod/jobs**
    c. Data in Response (Json)
    Example:
    [{"title":"First Job","postDate":1585697364953,"description":"this is job description","company":{"companyName":"hiring company 1","jobs":[]},"location":"New York","requiredSkills":["java","c++","communication"],"optionalSkills":null,"salaryMin":110000.0,"salaryMax":1500000.0,"benchmarkJob":null},{"title":"Second Job","postDate":1585697364953,"description":"this is job description","company":{"companyName":"hiring company 2","jobs":[]},"location":"New Jersey","requiredSkills":["everything","machine learning","you name it"],"optionalSkills":null,"salaryMin":130900.0,"salaryMax":2000000.0,"benchmarkJob":null}]

User Interface

[**@Terry** to complete]
Identify



User interaction design, class, function design etc.

Reference

1. Project Proposal: https://docs.google.com/document/d/1EmIof7MMFyxsASwXP0RPINLKWGnf9sVVePpdcviRGYk/edit
2. Technical Design: Which is this document. A live version is here: https://docs.google.com/document/d/1dvv37WRrlAl0sXfrrPW4SbblWHOJwqhbNW39IIrV6O4/edit#
3. Github: https://github.com/UPenn-CIT599/final-project-team-17-jobs-markets