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.

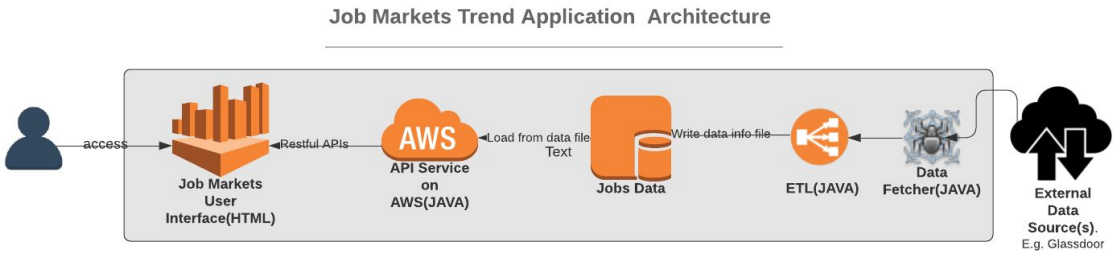


Diagram 1 - Jobs Markets Trend Application Architecture

Detailed Technical Design

Data Fetcher

[@Bao Jie to complete]
Local HTML file that consists of job posting data

ETL

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

1. Java Class 1 - HTMLParser.java
    - Parse HTML file to get table element as well as its rows and columns elements
    - Transform table element to a 2D array
    - Transform each row of the 2D array as string and convert into ArrayList
2. Java Class 2 - JobPostingDataWriter.java
    - Write fields stored in the ArrayList into a csv file
3. Java Class 3 - JobPostingDataSourcing.java
    - Run all the steps to generate the JobPostingData.csv

Jobs Data File Format

A csv file with the following fields:

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

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

Penn Engineering

API Service

CRCs for Core Classes

Following tables list some core classes used by API service. Please note that the following table 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.jobmarkets.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

Unit Test Cases

A list of unit test cases can be found here.

- https://github.com/UPenn-CIT599/final-project-team-17-jobs-markets/tree/master/src/test/java/com/upenn/cit591/jobmarkets

Restful APIs by API Service

The Frontend Web Application should use following APIs to fetch jobs data. For this project, we make data public available without authentication.

Domain: https://yrdltjhgh7.execute-api.us-east-1.amazonaws.com/Prod (please make this configurable in the frontend web application)

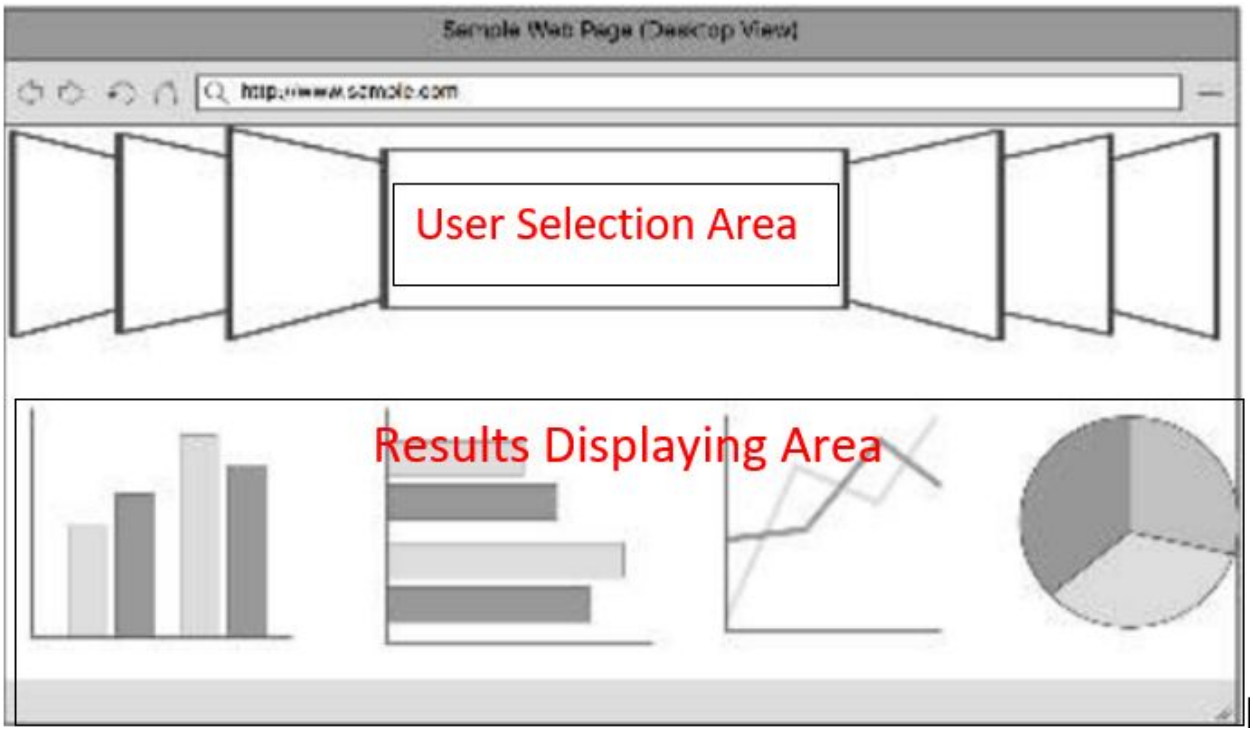
API	Purpose	Notes	Example
/jobs/all	List all job postings	See real data on: https://yrdltjhgh7.execute-api.us-east-1.amazonaws.com/Prod/jobs/all	Too long to paste here
/jobs/titles	List of job titles with count of each	https://yrdltjhgh7.execute-api.us-east-1.amazonaws.com/Prod/jobs/titles	{"Data Service Engineer and Systems Engineer":1,"Director ~ Data Science (Datalab)":1,"Copernicus Data Processing Operations Engineer":1,"Senior Data Scientist":3,"Hosted Processing Services Coordinator":1,"Data

			Scientist ~ Senior":1,"Sr Data Engineer":1,"Principal Computer Vision Engineer":1,"Freelance Data Scientist for an Award-Winning Music Software Company":1,"Data Scientist":1,"Machine Learning Engineers":1,"Data Science Manager (Sr.Data Scientist)":1,"Data Ambassador (f/m)":1}
/jobs/states	List of states with count of open positions in each	https://yrdltjhgh7.execute-api.us-east-1.amazonaws.com/Prod/jobs/states	{"NC":1,"FL":2,"MD":1,"ME":1,"OH":1,"WA":1,"NY":4,"CA":2}
jobs/companies	List of companies with count of open positions in each	https://yrdltjhgh7.execute-api.us-east-1.amazonaws.com/Prod/jobs/companies	{" VOYAGER":1," MIXED IN KEY":1," EQUINOX CORPORATION":1," EUMETSAT":3," UNIVERSAL MUSIC GMBH":1," AMPERSAND":2," D. E. SHAW RESEARCH":1," HEALTHCARE.COM":1," VISA":2," DSS ~ INC":1," GLOBAL FISHING WATCH":1}

User Interface

Initial Design

The results will show on one page, consisting of two parts. The top half contains forms to allow the user to select locations, years, skills, salary range, etc. The bottom half will be graphs, tables, and charts for visualization and results presentation.



User interaction design, class, function design etc.

Libraries and Languages:

The Web Application will be accomplished using the following languages or libraries:

- HTML allows us to specify the structure of Web content
- CSS is a formatting language used to describe the appearance of content in an HTML file
- Bootstrap: open source front-end development framework produced and maintained by Twitter that aids in producing clean, responsive web pages and applications.
- JavaScripts: Dynamically change Web Content

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/1dvv37WRrlAI0sXfrrPW4SbblWHOJwqhbNW39IIrV6O4/edit#
3. Github: https://github.com/UPenn-CIT599/final-project-team-17-jobs-markets