

Documentation :

Overview of the assignment :

The recruiter will come on the web portal and search for their Company using a search box and select any one company from a list of companies that appear in a dropdown.

The recruiter will be then displayed a form where he/she has to enter the Job Details.

Following data points are to be captured in the form:

- Job Title
- Seniority Level
- Location
- Job Description
- Skills Required
- A link to the original job posting (in the company's website)
- Salary
- Date of posting

Once the recruiter submits this information. They will be taken to a page where they can see all the Job Openings, they can choose to filter out Job Openings only posted by their company, filter by Job Title, and Location on this page.

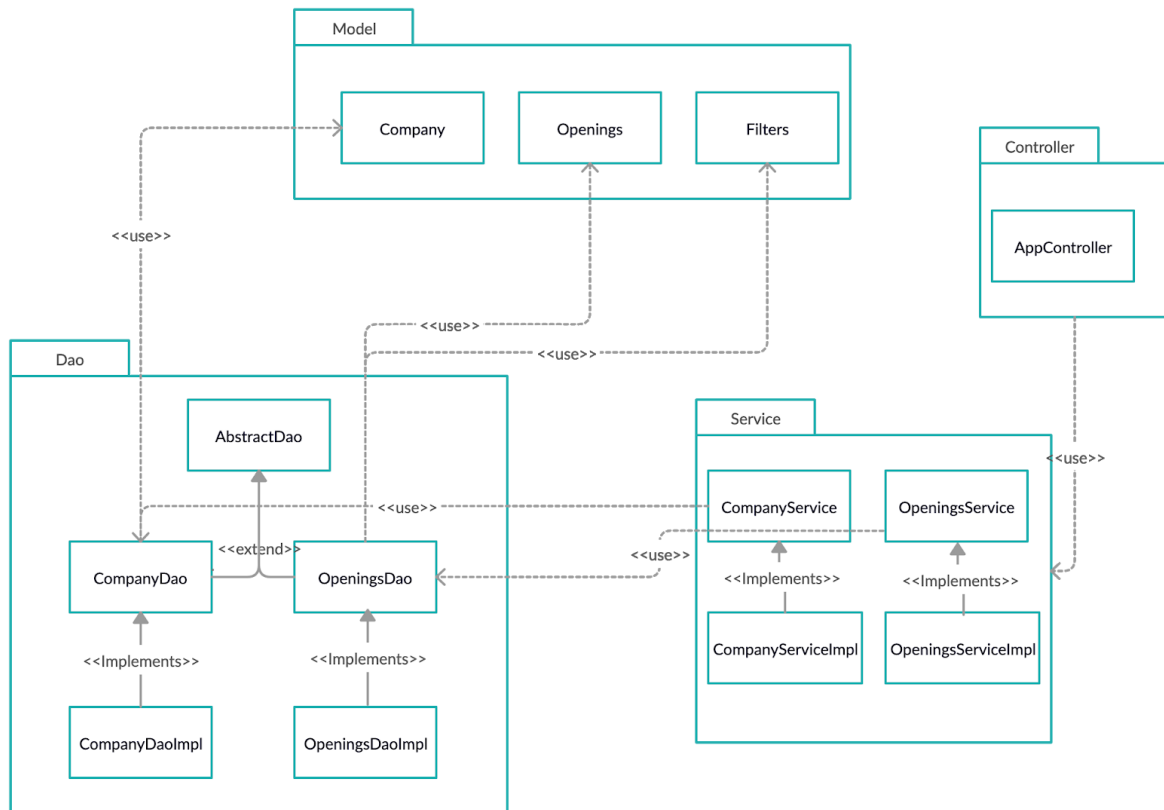
Functionalities:

- Creating a new Job Opening
- Searching for Companies
- Listing Existing Job Opening
- Filtering Job Openings
- Data validations wherever required

Assumptions:

- There is no need for authentication for the web portal, you can assume it to be open-for-all.
- You can assume that information for the company that the recruiter works for already exists in our database, and will be shown in the dropdown.

Abstracted Class Diagram :



Choice of Platforms:

1.) Spring

- a.) Supports Inversion of Control using beans.
- b.) MVC Framework.
- c.) Transaction Management.
- d.) JDBC Exception Handling.
- e.) Easy integration with other frameworks like Hibernate.

2.) Hibernate

- a.) Simplifies interaction with the database.
- b.) Has data persistence.
- c.) Lightweight.
- d.) Automatic table creation.

3.) Validation-api and hibernate-validator

- a.) Used for validating the inputs.

4.) MYSQL

- a.) Relational database management system.
- b.) Easy to use.
- c.) Free to download.

5.) Maven

- a.) Easy to build projects with
- b.) Simplifies project object model.

6.) Tomcat

- a.) For server.

7.) TestNG and mockito

- a.) For testing.

Design Advantages :

- 1.) Different independent entities for company and openings where company name acts like a foreign key. This gives the advantage of viewing the openings independently of the company giving the applicants more freedom for choosing options easily. This however, required authentication at the openings level along with the Company level.
- 2.) The API for calling company related information and openings related information is kept separate. This makes webportal and mobile calling the same API for openings but mobile calls company api to fetch additional details. This helps in avoiding duplication.
- 3.) Data Access Object Pattern or DAO pattern is used to separate low level data accessing API or operations from high level business services. This provides abstraction and encapsulation.
- 4.) Interfaces are used to add more abstraction.
- 5.) Filters are made as another entity in order to decouple user-inputs with the business logic.

How to Run :

Please install the requirements mentioned above. If not, the target folder is attached in the repository so you just need to copy the Upraised.war file to the tomcat webapps folder.

- 1.) Compile using “mvn clean install”
- 2.) Copy the Upraised.war from target folder to the apache-tomcat's webapps folder.
- 3.) Start the server.
- 4.) Hit the URL : <https://localhost:{port}/Upraised/companies>.