OOP JAVA PROJECT

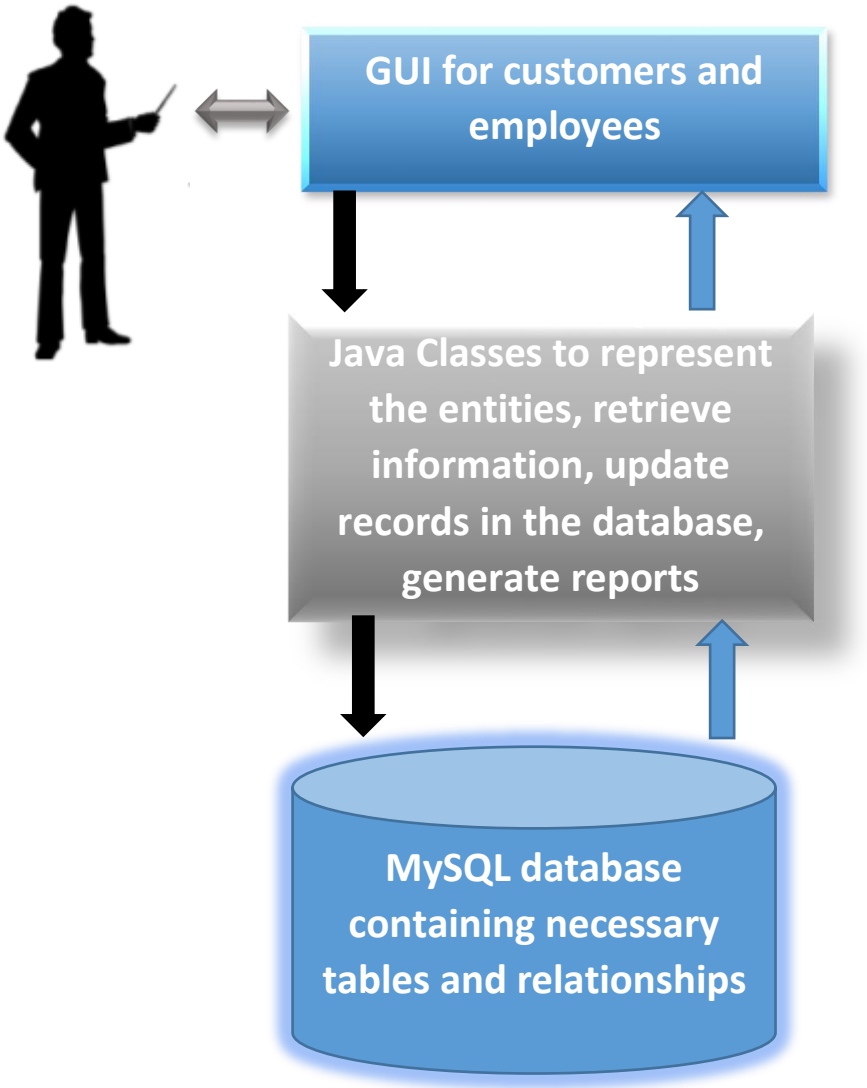
Contents

Group 9

Recruitment agency project

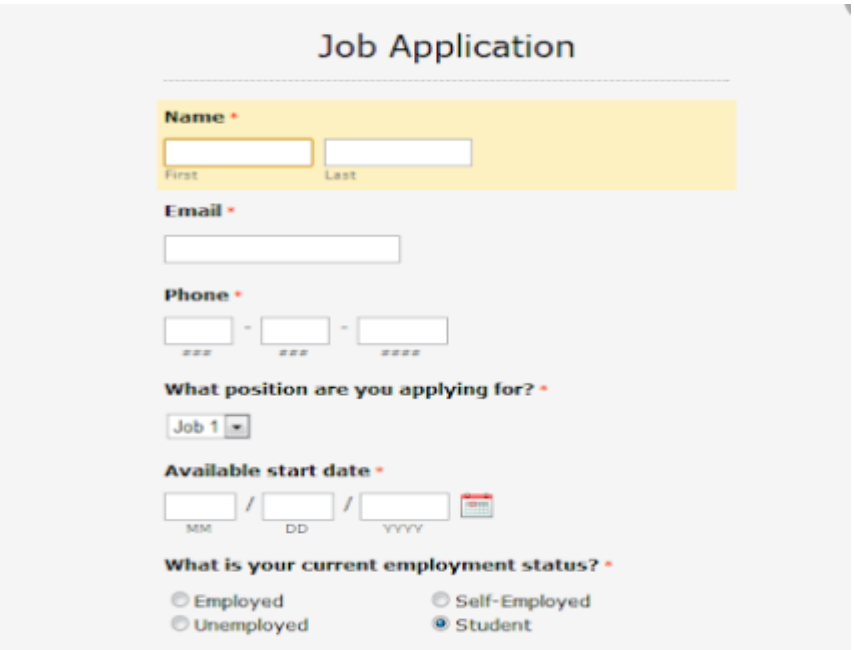
Goal:

The goal of the project is to enable the job seeks to apply for the job and support the companies to reach the right candidate. The application will also support officials of recruitment agency to manage the records of job seekers and the employers.



Program Description:

In this program you will write a set of supporting classes for recruitment agency. Here is an example of GUI.



The application should allow the jobseekers to browse through all the available jobs of different category. The job seekers will be able to apply for the job that suits their profile.

The employer will be able to put the new jobs on the application when required and will also be able to review the candidates which have registered as well those who have applied for the jobs. There will be two types of employers: New employers and member employers. The new employers will have to post the jobs with no discounts offered. The member employer could of type micro or non-micro. The member employers will be able to post the jobs with discount offered on the fees based on the type of membership.

The officials of the recruitment agency will be able to add new jobs, delete old jobs and will also be able to analyse most popular jobs.

The application primarily involves details of the available jobs, their features and as well as records generated and maintained for job seekers and employers.

The application should be developed for three types of users:

1. Job Seekers – Browse the jobs, apply for the jobs, review the update on application, Bill calculation with/without discounts, browsing the availability etc.
2. Employers– Update the currently available jobs, add new jobs, review candidates

3. Recruitment agency: add/delete jobs, review popularity of the jobs.

You are expected to design and develop the database for this application along with the Java classes necessary to implement the application.

Implementation requirements

- Necessary classes, methods and attributes should be designed using UML diagram notation. All the classes, methods and attributes should be explained in your documentation. Please discuss the design with me before you start implementation.
- You should be able to identify and introduce inheritance and aggregation relationship wherever applicable
- Necessary GUI screens should be added for successful execution.
- Records should be maintained in the database. Your Java code is expected to read and write to multiple tables as required.
- Every table must be populated with at least 6 records.

General System Architecture (MVC Pattern)

In this section, the general architecture for managing this schedule has been described. This system counts mainly 5 modules:

- The Information Search module: all possible requirements in the database, according to several search criteria
- The Data Update module : any modification, addition or deletion operation in the database
- The Reporting module: statistics in the form of graphs (pie charts, histograms etc.)
- The Data Access (DAO) module queries or updates the data in the database and communicates with the 3 modules previous
- The graphical interface communicates with the first 3 modules to visualize the schedule

According to the MVC pattern, your graphical interface constitutes the View (only the display) dependent on the actions of the user (event management) at Controller level (Research, Update and reporting modules).
This will ask the Model to retrieve or update- via the data access module (DAO) - the information of the database, organize and assemble them (for example, by storing them in collections). Then the Controller will ask the Model for the data, analyse it, make decisions and send the text back to the View.
You are advised to adopt the model for the development of a coherent project.
You can find more about the MVC Pattern at

https://openclassrooms.com/fr/courses/4670706-adoptez-une-architecture-mvc-en-php/4678736-comment-fonctionne-une-architecture-mvc

https://accu.org/journals/overload/16/88/grenyer_1524/

https://www.oracle.com/technical-resources/articles/javase/mvc.html

Guidance for structured development of project

Step 1: Relationship model

Review all possible requirements of the database and the search criteria. Identify the possible entities, attributes in the database. It is important to carefully recognize the role of each attribute and then decide the datatype of the attribute. It is will also play a key contribution towards determining the primary and foreign key attributes. Document the relationships between the entities.

Step 2: Creating database

Based on the relationship model above, create tables and relationships using MySQL. Insert the records into the tables.

Step 3: Information Finder

Review the user requirements to identify the possible range of information you need to retrieve from the database. Specifically in the case of business organizations, it is important for the employees to analyse the sales. It is equally essential for the customers to analyse their past purchases. For example:

- The number of jobs advertised by a particular employer
- The number of applications made by job seeker in last three months

Develop the necessary classes to represent entities which will enable the user to query the database.

Step 4: Data access

This module represents the data access layer (DAO) in the DB. Via a JDBC access to the database, this module executes the queries responsible for retrieving or updating the data in the database. This is a type of object that loads to retrieve the data in the database and that another type of object be used to handle this data (business layer).

Step 5: GUI & Reporting

A welcome window will allow the user to connect to the database by entering their EMAIL and PASSWORD. These information, if stored in the USER table, will give him access or / and update rights to certain data of the schedule.

Your graphical interface will display in an ergonomic, clear and fluid way all the relevant information. It will allow you to navigate intuitively from one page to

another. For example, a page of your interface graph can contain menus with menu items, or tabs if you prefer.

This module is used to generate statistics (pie charts, histograms, etc.) using JFreeChart. You can find the details in the resources section.

Deliverables

The deliverable should be a zipped file per team containing the following:

1. A PowerPoint presentation including
    a. Title
    b. Name of the team members
    c. Summary
    d. Class diagram(draw.io) or any other similar tool
    e. Database design
    f. GUI screen samples
    g. Your own evaluation of the project
    h. Bibliography
2. Java code: All the folders and files of the project developed on Eclipse or Netbeans with the sources, the executable .jar in mode graphics and Javadoc documentation with comments on classes and methods.

Resources

JDBC: https://www.jmdoudoux.fr/java/dej/chap-jdbc.htm, (Author: Jean-Michel Doudoux)

My SQL: http://dev.mysql.com/doc/refman/5.7/en/

JFreeChart:

The JFreeChart Class Library (Author: David Gilbert)

http://www.jfree.org/jfreechart/api/javadoc/index.html

http://www.java2s.com/Code/Java/Chart/CatalogChart.htm

http://www.jfree.org/forum/

Wireframe: https://webdesign.tutsplus.com/articles/a-beginners-guide-to-wireframing--webdesign-7399