**EX:NO:3 CLASS DIAGRAM FOR WOMEN EMPLOYMENT SYSTEM**

**DATE:01.08.25**

**AIM:**

The aim of this exercise is to identify the relevant classes and their relationships within the Expo Management System using UML. It also focuses on applying notations such as class, association, aggregation, and composition to model the structure of the system. Additionally, the goal is to recognize the conceptual classes and understand how they interact within the system to support object-oriented design.

**IDENTIFICATION OF CLASS:**

- Admin: Contains attributes such as adminID, name, and userID. It includes the operation update() to modify admin details.
- User: Represents general users of the system with attributes such as userID, usertype, username, and password. It provides operations like register(), login(), and delete().
- JobSeeker: Represents women seeking employment. Attributes include seekerID, name, skills, qualification, resume, and contact. Operations include createProfile() and updateProfile().
- Employer: Represents companies or individuals posting job opportunities. Attributes are employerID, companyName, contact, address, and userID. Operations include create() and update().
- Job: Represents job listings posted on the platform. Includes jobID, title, description, location, salary, and employerID. Supports postJob(), updateJob(), and deleteJob() operations.
- Application: Represents job applications submitted by seekers. Attributes include applicationID, seekerID, jobID, status, and dateApplied. Operations are submit() and updateStatus().
- TrainingProgram: Represents training or upskilling programs offered to job seekers. Attributes include programID, title, description, duration, and provider. Supports create() and enroll() operations.
- Feedback: Collects feedback or reviews from users. Attributes include feedbackID, userID, message, and rating. Operations are submit() and view().
- Reports: Used to generate various reports (e.g., job stats, user activity). Attributes include reportID, type, and description. Operations supported are generate() and download().

**NOTATION:**

**2.1) Class**

Each class is represented as a rectangle divided into three compartments:

**Top**: Class Name (e.g., Admin)

**Middle**: Attributes (e.g., adminID: int)

**Bottom**: Operations/Methods (e.g., update())

**2.2) Association**

Represents a basic relationship between two classes (usually shown by a straight line).

Example from the diagram:

User is associated with Admin, JobSeeker, and Employer.

Employer is associated with Job.

**2.3) Aggregation**

Aggregation (denoted by a hollow diamond) represents a "has-a" relationship where one class contains another but can exist independently.

From the diagram:

- Employer aggregates Job (An Employer has Jobs, but a Job can exist independently).
- TrainingProgram aggregates JobSeeker (JobSeekers can enroll, but programs can exist independently).

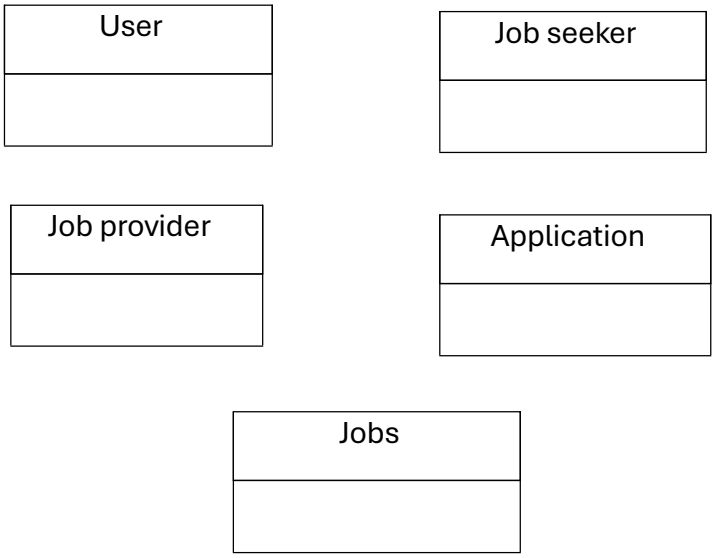
**2.4) Composition**

Composition (denoted by a filled diamond) represents a stronger "part-of" relationship, where the part cannot exist without the whole.

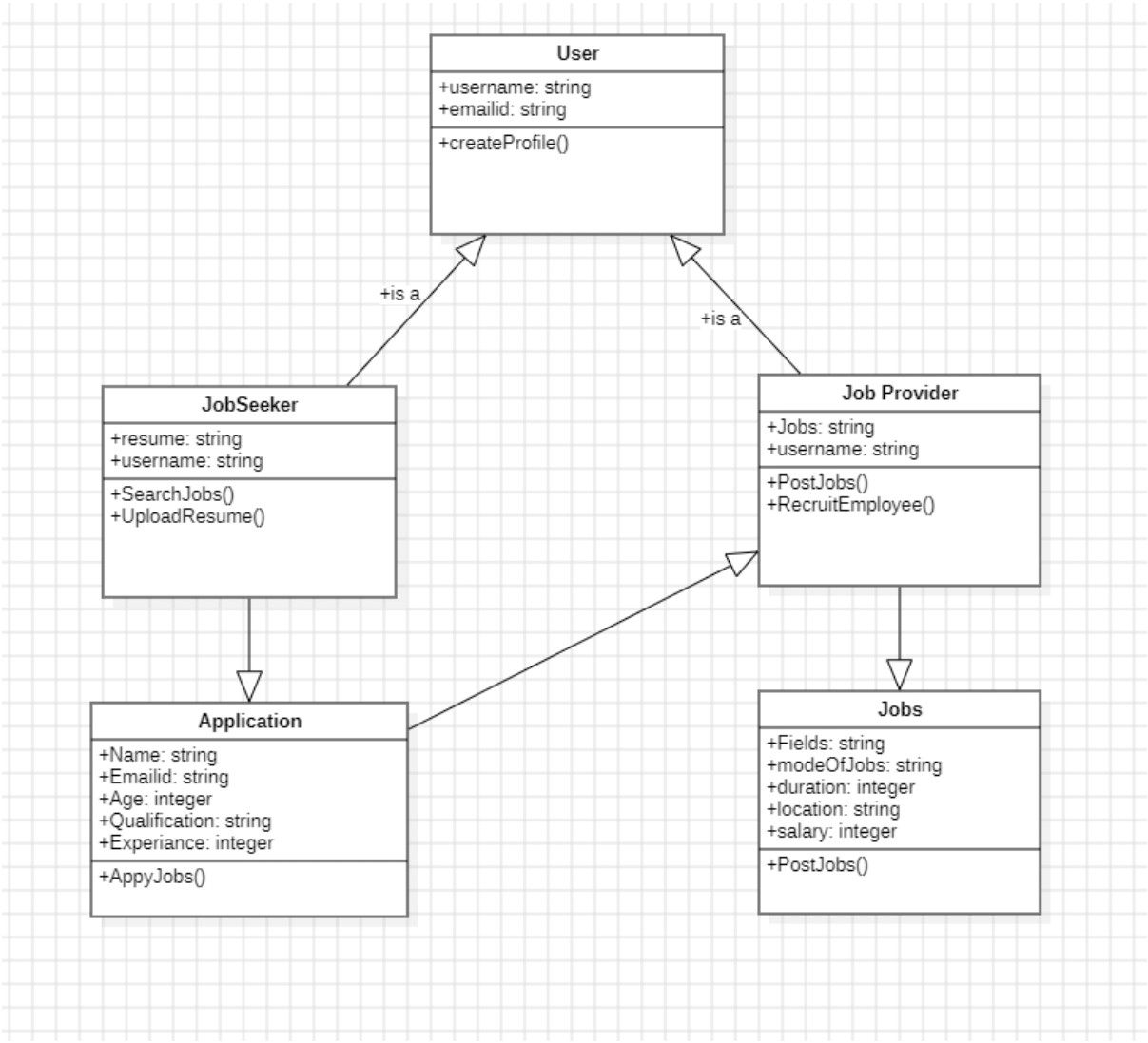
From the diagram:

- Admin is composed with User (Admin cannot exist without being a User).
- Application is composed with JobSeeker (Application cannot exist without a JobSeeker).

## IDENTFICATION OF CONCEPTUAL CLASS:



**CLASS DIAGRAM:**




BAVAKARNI G
23CSEA06

| | |
|---|---|
| **OBSERVATION (20)** | |
| **RECORD (5)** | |
| **TOTAL (25)** | |

**RESULT:**

The Women Employemnt  System's class structure was successfully modeled using UML. Classes, their attributes, methods, and interrelationships were clearly identified. UML notations such as class, association, and aggregation helped visualize the system’s design. This detailed analysis supports better understanding and implementation of an object-oriented solution for expo event management

BAVAKARNI G
23CSEA06