

## **JobSeeker App - Preliminary Project Documentation**

### **1. Title for the Project: JobSeeker App**

A web-based job application tracking system for managing job applications, interviews, and offers throughout the job search process.

### **2. Problem and Domain Description:**

#### **Problem Statement:**

Job seekers often apply to multiple positions across various companies, making it challenging to:

- Track application statuses (Applied, Interviewing, Offer, Rejected)
- Remember important details such as application dates, sources, and resume versions used
- Manage interview rounds and their outcomes
- Keep track of offers and their deadlines
- Organize information by company and job posting

Currently, many job seekers rely on spreadsheets, notes, or memory, which leads to:

- Lost opportunities due to missed deadlines
- Difficulty tracking progress across multiple applications
- Lack of insights into application patterns and success rates
- Inefficient organization of job search data

#### **Domain**

The domain involves:

- **USERS:** Job seekers who need to track their applications
- **COMPANIES:** Organizations offering job positions
- **JOB POSTINGS:** Specific positions advertised by companies
- **APPLICATIONS:** User submissions for specific job postings
- **INTERVIEW ROUNDS:** Stages in the interview process (Phone Screen, Technical, Onsite, etc.)
- **OFFERS:** Job offers received from companies

This domain requires tracking temporal data (application dates, interview schedules, offer deadlines), status management (application status, interview status, offer status), and relationships between entities (users apply to jobs, jobs belong to companies, applications have multiple interview rounds).

### 3. Solution Description

#### Overview

- JobSeeker App is a Spring Boot web application that provides a centralized platform for job seekers to manage their entire job search process. The application allows users to track applications, monitor interview progress, manage offers, and gain insights into their job search activities.

#### Key Features

1. User Authentication: Secure signup and login system with password hashing
2. Application Management: Create, view, update, and delete job applications
3. Company Tracking: View and manage companies with application counts
4. Interview Management: Track multiple interview rounds per application with status and feedback
5. Offer Management: Record and manage job offers with compensation details and deadlines
6. Dashboard Analytics: View summary statistics including total applications, open applications, and company counts

#### User Interfaces

The solution includes the following five distinct user interfaces:

##### 1. Login Page (`/login`)

- User authentication interface
- Allows existing users to log in with username and password
- Redirects to dashboard upon successful login

##### 2. Signup Page (`/signup`)

- New user registration interface
- Collects username, email, and password
- Validates password requirements and confirms password match
- Redirects to the login page upon successful registration

##### 3. Dashboard (`/dashboard`)

- Main landing page after login
- Displays key metrics: total applications, open applications, company count
- Shows recent application updates
- Provides a quick overview of the job search status
  - Primary content generated from database: Application counts, company summaries, recent applications

##### 4. Applications Page (`/applications`)

- List view of all users' job applications
- Displays application details: company name, job title, location, status, and applied date
- Allows creating new applications with company and job posting information - Supports updating application status and notes
- Supports deleting applications

- Primary content generated from database: Application list with joined company and job posting data

#### 5. Application Detail Page (`/applications/{id}`)

- Detailed view of a single application
- Shows complete application information
- Displays all interview rounds associated with the application
- Shows all offers associated with the application
- Allows adding new interview rounds
- Allows adding new offers
- Primary content generated from database: Application details, interview rounds, offers

#### 6. Companies Page (`/companies`)

- List view of all companies
- Shows company information: name, industry, headquarters location, website
- Displays application count per company
- Sorted by application count and company name
- Primary content generated from database: Company list with aggregated application counts

All user interfaces are dynamically generated from data stored in the MySQL database, ensuring real-time updates and data consistency.

#### 4. Preliminary ER Diagram

The Entity-Relationship model consists of 6 entity sets with the following relationships:

Entity Sets:

##### 1. User:

- Attributes: user\_id (PK), username, email, password\_hash, created\_at - Represents job seekers using the application

##### 2. Company

- Attributes: company\_id (PK), name, industry, hq\_location, website
- Represents companies offering job positions

##### 3. Job Posting

- Attributes: job\_id (PK), company\_id (FK), title, location, employment\_type, job\_level, posting\_date, application\_deadline, season, is\_open
- Represents specific job positions
- Related to: Company (many-to-one)

##### 4. Application

- Attributes: application\_id (PK), user\_id (FK), job\_id (FK), applied\_date, status, source, resume\_version, notes, last\_updated

- Represents a user's application to a job posting
- Related to: User (many-to-one), JobPosting (many-to-one)

## 5. Interview Round

- Attributes: interview\_round\_id (PK), application\_id (FK), round\_number, status, feedback, created\_at
- Represents interview stages for an application
- Related to: Application (many-to-one)
- Note: Each application can have multiple interview rounds, identified by round\_number

## 6. Offer

- Attributes: offer\_id (PK), application\_id (UNIQUE FK), compensation, start\_date, decision\_deadline, status, notes, created\_at
- Represents job offers received
- Related to: Application (one-to-zero-or-one)
- Note: Each application can have at most one offer (application\_id is UNIQUE)

## Relationships and Cardinalities:

User → Application: One-to-Many (1:N)

- One User can submit N Applications
- Cardinality: 1 on User side, N on Application side

JobPosting → Application: One-to-Many (1:N)

- One JobPosting can receive N Applications from different users
- Cardinality: 1 on JobPosting side, N on Application side

Company → JobPosting: One-to-Many (1:N)

- One Company can have N JobPostings
- Cardinality: 1 on Company side, N on JobPosting side

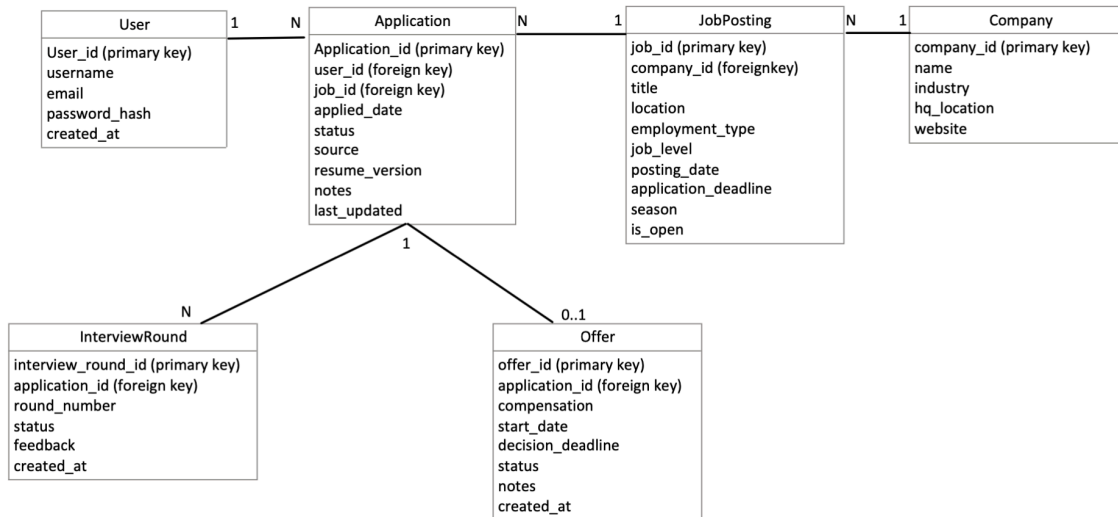
Application → InterviewRound: One-to-Many (1:N)

- One Application can have N InterviewRounds
- Cardinality: 1 on Application side, N on InterviewRound side

Application → Offer: One-to-Zero-or-One (1:0..1)

- One Application can have 0 or 1 Offer
- Cardinality: 1 on Application side, 0..1 on Offer side
- This is enforced by the UNIQUE constraint on application\_id in the Offer table

## ER Diagram Representation:



### Key Design Decisions:

- InterviewRound uses `interview\_round\_id` as the primary key and `round\_number` to identify the sequence of interview rounds for an application. The composite key (application\_id, round\_number) ensures uniqueness of rounds per application.

- Offer has a UNIQUE constraint on `application\_id`, meaning each application can have at most one offer. This models the business rule that typically one application results in one final offer decision, though the offer status can change (Pending, Accepted, Declined).

### 5. Technologies

**Backend:** - Java: Programming language for server-side development

- Spring Boot 3.1.4: Framework for building the web application

- Spring Boot Starter Web: Enables RESTful web services and MVC architecture

- Spring Boot Starter JDBC: Provides JDBC support for database connectivity

- Spring Security Crypto: For secure password hashing using BCrypt

### Database: -

MySQL 8.0: Relational database management system

- JDBC (Java Database Connectivity): API for connecting to MySQL database

- MySQL Connector/J 8.0.33: JDBC driver for MySQL

**Frontend: -**

Mustache Templates: Server-side templating engine for HTML generation

HTML/CSS/JavaScript: Standard web technologies for user interface

Font Awesome: Icon library for UI elements

**Build Tool:**

- Maven: Project management and build automation tool
- Spring Boot Maven Plugin: For packaging and running the Spring Boot application

**Development Environment:**

- Docker: For running MySQL database instance (port 33306)

**Security:**

- BCrypt: Password hashing algorithm for secure password storage
- Prepared Statements: For SQL injection prevention

**Third-Party Libraries:**

All libraries used are either built-in Java libraries, Spring Boot dependencies, or libraries included in the project 2 starter code. No external libraries requiring instructor permission are used.

**Summary**

The JobSeeker App addresses the real-world problem of job application management by providing a comprehensive web-based solution. The application uses a well-designed database schema with six entity sets capturing all necessary relationships:

- User and Company as core entities
- JobPosting - linking companies to available positions
- Application - connecting users to job postings
- InterviewRound - tracking the interview process for each application
- Offer - managing job offers with a one-to-zero-or-one relationship to applications

The solution provides six distinct user interfaces (Login, Signup, Dashboard, Applications List, Application Detail, and Companies) that generate content directly from the database. The technology stack leverages Spring Boot for rapid development, MySQL for reliable data storage, and modern web technologies for an intuitive user experience.