**NAME: MSHAHZEB**

**ROLL: SP22-BSE-073**

**Fully Dressed UseCase:**

**Use Case: Post Job Listings**

**Use Case ID**: UC-001  
**Use Case Name**: Post Job Listings  
**Primary Actor**: Employer (or Admin)  
**Stakeholders and Interests**:

* **Employer**: Wants to post job listings to attract qualified candidates.
* **Job Seekers**: Interested in viewing job listings for potential employment opportunities.
* **System Administrator**: Ensures the job listing conforms to platform guidelines and policies.

**Preconditions**

1. Employer must be registered and logged into the system.
2. Employer has a verified company profile on the platform.

**Postconditions**

1. Job listing is successfully created and visible to job seekers.
2. Employer can manage (edit, deactivate, or delete) the job listing if needed.

**Main Success Scenario (Basic Flow)**

1. **Employer navigates to the "Post a Job" page**:  
   The system displays a job posting form.
2. **Employer enters job details**:
   * Job Title
   * Job Description
   * Required Skills and Qualifications
   * Job Type (e.g., Full-time, Part-time, Contract)
   * Salary Range
   * Location
   * Application Deadline
3. **Employer reviews job listing details**:  
   The system displays a preview of the job listing.
4. **Employer submits the job listing**:
   * The system validates the information (e.g., checks for mandatory fields, correct format).
5. **System confirms job listing creation**:
   * The job listing is saved in the database and marked as "Active."
   * The system sends a confirmation notification to the employer.
6. **System makes job listing available**:
   * The job listing is now visible to job seekers browsing available positions.

**Extensions (Alternate Flows)**

* **3a. Employer provides incomplete or invalid details**:
  + The system prompts the employer to correct the missing or invalid fields.
  + Employer corrects and resubmits the details.
* **5a. Job listing fails system validation**:
  + The system provides error messages detailing what needs to be corrected.
  + Employer makes the required changes and resubmits.

**Special Requirements**

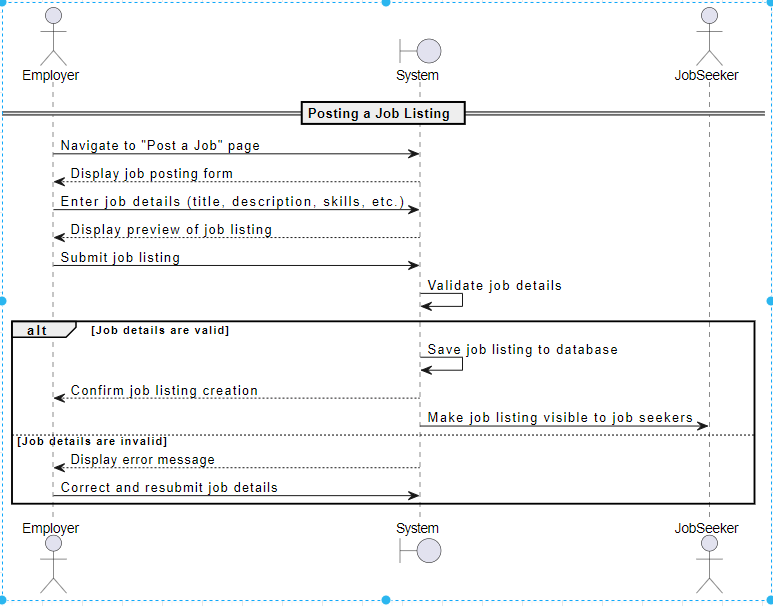
* The job listing should have the option to auto-deactivate after the application deadline.
* Employers can select whether they want applications submitted directly on the platform or via an external link.

**Assumptions**

* Employers have the necessary permissions to post job listings.
* Job listings are only accessible to authenticated job seekers if the platform is private.

**Open Issues**

* Should employers be able to preview how the listing appears to job seekers before finalizing?
* Are there specific guidelines or moderation for content within job postings?

**SSD**

SSD diagram code:

// --- Presentation Layer ---

class Employer {

private String name;

private JobListingService jobListingService;

public Employer(String name, JobListingService jobListingService) {

this.name = name;

this.jobListingService = jobListingService;

}

public void postJobListing(String title, String description, String skills, String jobType, String location, double salary) {

System.out.println(name + " is navigating to 'Post a Job' page.");

JobListing jobListing = JobListingFactory.createJobListing(title, description, skills, jobType, location, salary);

jobListingService.postJob(jobListing, this);

}

}

// --- Business Logic Layer ---

class JobListing {

private String title;

private String description;

private String skills;

private String jobType;

private String location;

private double salary;

public JobListing(String title, String description, String skills, String jobType, String location, double salary) {

this.title = title;

this.description = description;

this.skills = skills;

this.jobType = jobType;

this.location = location;

this.salary = salary;

}

public String getTitle() { return title; }

public String getDescription() { return description; }

public String getSkills() { return skills; }

public String getJobType() { return jobType; }

public String getLocation() { return location; }

public double getSalary() { return salary; }

}

class JobListingService {

private SystemService systemService;

public JobListingService(SystemService systemService) {

this.systemService = systemService;

}

public void postJob(JobListing jobListing, Employer employer) {

systemService.displayJobPostingForm();

systemService.previewJobListing(jobListing);

if (systemService.validateJobListing(jobListing)) {

systemService.saveJobListing(jobListing);

systemService.confirmJobListingCreation(employer);

systemService.makeJobListingVisibleToJobSeekers(jobListing);

} else {

systemService.displayErrorMessage();

}

}

}

// --- Data Access Layer ---

interface JobListingDAO {

void save(JobListing jobListing);

}

class JobListingDAOImpl implements JobListingDAO {

@Override

public void save(JobListing jobListing) {

System.out.println("Saving job listing to database: " + jobListing.getTitle());

}

}

// --- Utility Factory ---

class JobListingFactory {

public static JobListing createJobListing(String title, String description, String skills, String jobType, String location, double salary) {

return new JobListing(title, description, skills, jobType, location, salary);

}

}

// --- Singleton System Service ---

class SystemService {

private static SystemService instance;

private JobListingDAO jobListingDAO;

private SystemService() {

this.jobListingDAO = new JobListingDAOImpl();

}

public static synchronized SystemService getInstance() {

if (instance == null) {

instance = new SystemService();

}

return instance;

}

public void displayJobPostingForm() {

System.out.println("Displaying job posting form.");

}

public void previewJobListing(JobListing jobListing) {

System.out.println("Displaying job preview for '" + jobListing.getTitle() + "'.");

}

public boolean validateJobListing(JobListing jobListing) {

return jobListing.getTitle() != null && !jobListing.getTitle().isEmpty()

&& jobListing.getDescription() != null && !jobListing.getDescription().isEmpty();

}

public void saveJobListing(JobListing jobListing) {

jobListingDAO.save(jobListing);

}

public void confirmJobListingCreation(Employer employer) {

System.out.println("Job listing creation confirmed for " + employer + ".");

}

public void makeJobListingVisibleToJobSeekers(JobListing jobListing) {

System.out.println("Making job listing visible to job seekers.");

}

public void displayErrorMessage() {

System.out.println("Job details are invalid. Displaying error message.");

}

}

// --- Main Program ---

public class Main {

public static void main(String[] args) {

SystemService systemService = SystemService.getInstance();

JobListingService jobListingService = new JobListingService(systemService);

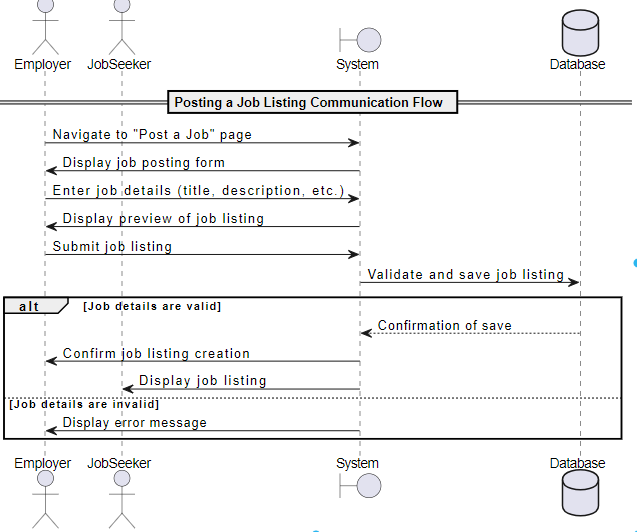
Employer employer = new Employer("John Doe", jobListingService);

employer.postJobListing("Software Engineer", "Responsible for developing applications.", "Java, Spring Boot", "Full-Time", "Remote", 85000.00);

}

}

Communication Diagram

s

Package diagram

