Job portal web application

<u>Contents</u>

1. Introduction:

Job Portal is a social networking platform for professionals where they can create their profile, The Job Portal platform also provides various tools help users search for job opportunities .

2. Project Objective:

The objective of this software engineering project is to build a LinkedIn-like platform for professionals, where users can create their profile and search for job opportunities.

3. Functional Requirements:

1- User Signup and Login: Users should be able to create their profiles by providing their personal and professional details such as name, email address, work experience, Users should also be able to login to their accounts using their email address and password.

2- Profile Creation: Users should be able to create their profiles by providing their personal and professional details such as name, email address, work experience, education .

3- Job Search: Users should be able to search for job opportunities on the platform by entering keywords, location, and job title. Users should also be able to filter job search results by various criteria such as job type, experience level, and salary range.

4. Non-Functional Requirements:

1- Security: The platform should ensure the security of user data by implementing appropriate security measures such as encryption and access controls.
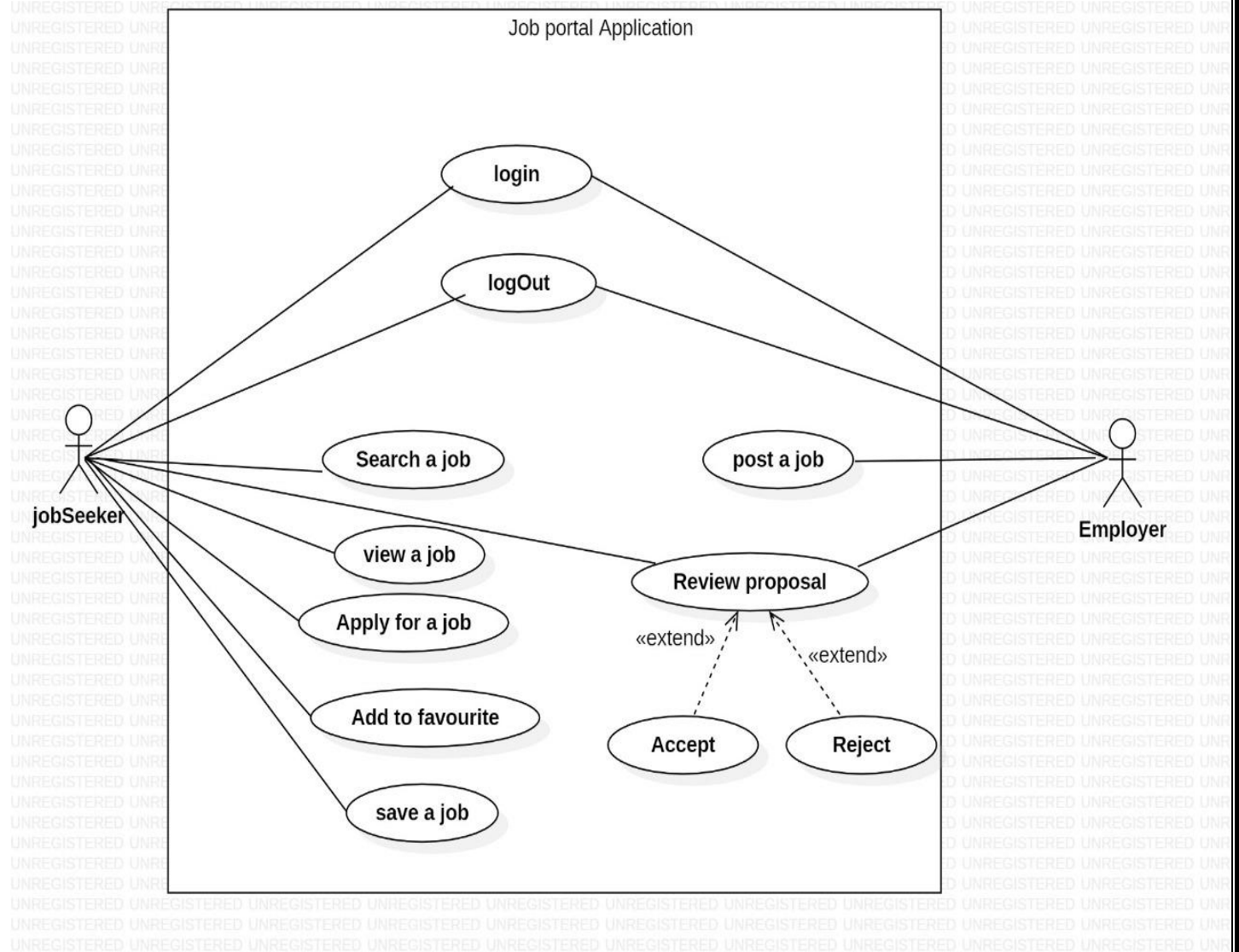
2- Scalability: The platform should be designed to handle a large number of users and data.

3- Availability: The platform should be available 24/7 with minimal downtime for maintenance and upgrades.

4- Performance: The platform should be designed to provide a fast and responsive user experience.

5. <u>Use-Cases Diagram</u>

6.   Use-Case Description

Use case name	Log in
Actor	Employer , Employee , Admin
Goal	Logging in Successfully
Pre-Condition	User has an account so he/she can log in
Post-Condition	The user logged in and can interact with the system
Main Success scenario	1- the user will write his information. 2- the system checks the database for this info. 3- a3- the user will be logged in successfully

Use case name	Register
Actor	Employee
Goal	User Create an account
Pre-Condition	User must be connected to internet
Post-Condition	The user has created an account and can interact with the system
Main Success scenario	1- The user should open the program. 2- The user should put some info like username, email and password. 3- the user will create an account and be logged in successfully

Use case name	Log out
Actor	Employer , Employee , Admin
Goal	Logged out from user's account
Pre-Condition	User has an account, and he/she is logged in
Post-Condition	The user logged out successfully
Main Success scenario	1- the user logged in. 2- the user chooses to log out 3- the user is logged out successfully

8

Use case name	Search
Actor	Employee
Goal	Employer can find Jobs
Pre-Condition	User has an account and he/she is logged in as Employee
Post-Condition	User find what he was searching for successfully
Main Success scenario	1- the Employee logged in 2- the Employee open search icon 3- the user write what he is looking for(job) 4- the user's found what he/she wants

Use case name	post job
Actor	Employer
Goal	To offer job
Pre-Condition	User has an account and he/she is logged in s employer
Post-Condition	The user offer the job successfully
Main Success scenario	1- the user logged in 2- the user click on offer job option 3- the user wrote job's information 4- the user offered the job successfully

Use case name	Apply Job
Actor	Employee
Goal	applying for a job posted on job board
Pre-Condition	1- The user must have an account. 2- The user must be logged in to their account. 3- The user must have identified a job posting they want to apply for
Post-Condition	1- If the user's application is successful, they may be invited for an interview or further consideration for the job. 2- If the user's application is unsuccessful, they may continue to search and apply for other jobs on LinkedIn's job board
Main Success scenario	1- the user navigates to the job posting on LinkedIn's job board. 2- The user clicks on the "Apply" button on the job posting. 3- The user fills out the application form with their personal information, work experience, education, skills 4- The user reviews their application and makes any necessary edits or updates. 5- The user clicks on the "Submit" button to send their application to the employer.

Use case name	Accept a Job
Actor	Admin
Goal	To accept Job post
Pre-Condition	User has an account and he/she is logged in as Admin
Post-Condition	The Employer offer the job successfully
Main Success scenario	1- the user logged in as Admin 3- the Admin click on view job 4- the Admin Click on accept job

Use case name	Refuse job
Actor	Admin
Goal	To Refuse a job
Pre-Condition	User has an account and he/she is logged in as Admin
Post-Condition	The Employer offer is not posted
Main Success scenario	1- the user logged in 2- the user click on offer job option 3- the user wrote job's information 4- the user offered the job successfully

Use case name	View job
Actor	Employee
Goal	To apply for a job
Pre-Condition	User has an account and he/she is logged in s employee
Post-Condition	The employee apply for the job successfully
Main Success scenario	1- the user logged in as emplyee 2- the user click on apply for a job option 3- the employee write his information 4- the user applied for the job successfully

Use case name	Add to favorites
Actor	Employee
Goal	To add a job for his favorites
Pre-Condition	User has an account and he/she is logged in as employee
Post-Condition	The employee apply for the job successfully
Main Success scenario	1- the user logged in as employee 2- the user click on add to favorite option 3- the employee add the job to his favorites successfully

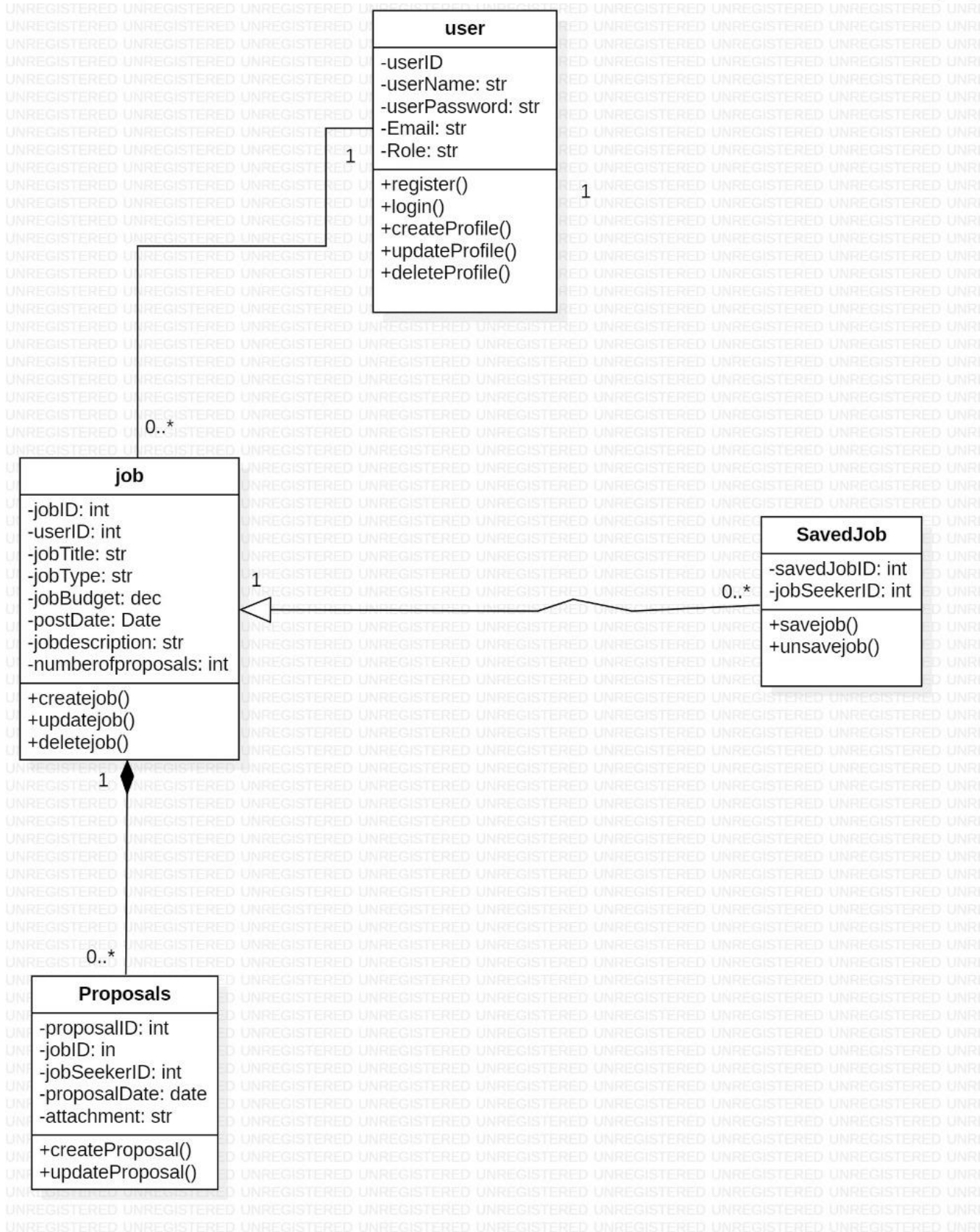| Use case name	Review proposals
Actor	Employer
Goal	To view the applied forms for a job
Pre-Condition	User has an account and he/she is logged in sss employer
Post-Condition	The employer view applies for the job successfully
Main Success scenario	1- the user logged in as employer 2- the user click on view proposals option 3- the employer view the applies for job successfully

Use case name	Reject proposal
Actor	Employer
Goal	To reject a proposal for his posted jobs
Pre-Condition	User has an account and he/she is logged in as employer
Post-Condition	The employer reject proposal for the job successfully
Main Success scenario	1- the user logged in as employer 2- the user click on view proposals option 3- the employer reject the proposal successfully

Use case name	Accept proposal
Actor	Employer
Goal	To accept a proposal for his posted jobs
Pre-Condition	User has an account and he/she is logged in as employer
Post-Condition	The employer accept proposal for the job successfully
Main Success scenario	1- the user logged in as employer 2- the user click on view proposals option 3- the employer accept the proposal successfully

Use case name	Communicate with employee
Actor	Employer
Goal	To communicate with the accepted employee
Pre-Condition	User has an account and he/she is logged in as employer
Post-Condition	The employer communicate with the accepted employee for the job successfully
Main Success scenario	1- the user logged in as employer 2- the user click on view proposals option 3- the employer communicate with the accepted employee successfully

Use case name	Manage employees
Actor	Admin
Goal	To Update , create ,Delete and Read employees
Pre-Condition	User has an account and he/she is logged in as Admin
Post-Condition	The employer manage employees
Main Success scenario	1- the user logged in as employer 2- the user click on view proposals option 3- the employer reject the proposal successfully

user

-userID
-userName: str
-userPassword: str
-Email: str
-Role: str

+register()
+login()
+createProfile()
+updateProfile()
+deleteProfile()

1

1

0..*

job

-jobID: int
-userID: int
-jobTitle: str
-jobType: str
-jobBudget: dec
-postDate: Date
-jobdescription: str
-numberofproposals: int

+createjob()
+updatejob()
+deletejob()

1

SavedJob

-savedJobID: int
-jobSeekerID: int

+savejob()
+unsavejob()

0..*

1

0..*

Proposals

-proposalID: int
-jobID: in
-jobSeekerID: int
-proposalDate: date
-attachment: str

+createProposal()
+updateProposal()

OCL

Context User
Invariants:
  - uniqueUserId: self.allInstances()->forAll(u1, u2 | u1 <> u2 implies u1.user_id <> u2.user_id)

  - nonEmptyFields: not self.username.isEmpty() and not self.password.isEmpty() and not self.email.isEmpty() and not self.role.isEmpty()

  - validEmail: self.email.matches('^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$')

- Unique_Username: User.allInstances()->forAll(u1, u2 | u1 <> u2 implies u1.username <> u2.username)

- Unique_Email: User.allInstances()->forAll(u1, u2 | u1 <> u2 implies u1.email <> u2.email)

Context job
Invariants:

Invariants:
  - uniqueJobId: self.allInstances()->forAll(j1, j2 | j1 <> j2 implies j1.job_id <> j2.job_id)

  - nonNegativeBudget: self.job_budget >= 0

  - validPostDate: self.post_date <= Date::today()

  - nonEmptyTitle: not self.job_title.isEmpty()

  - Positive_Number_of_Proposals: self.number_of_proposals >= 0

Context Proposals
Invariants:

- uniqueProposalId: self.allInstances()->forAll(p1, p2 | p1 <> p2 implies p1.proposal_id <> p2.proposal_id)

   - validProposalDate: self.proposal_date <= Date::today()

   - validStatus: self.status = 'pending' or self.status = 'accepted' or self.status = 'rejected'

   - nonEmptyAttachment: not self.attachment.isEmpty()

   -Unique_Proposal_For_Job_Seeker: Proposal.allInstances()->forAll(p1, p2 | p1 <> p2 implies p1.job_seeker_id <> p2.job_seeker_id or p1.job_id <> p2.job_id)

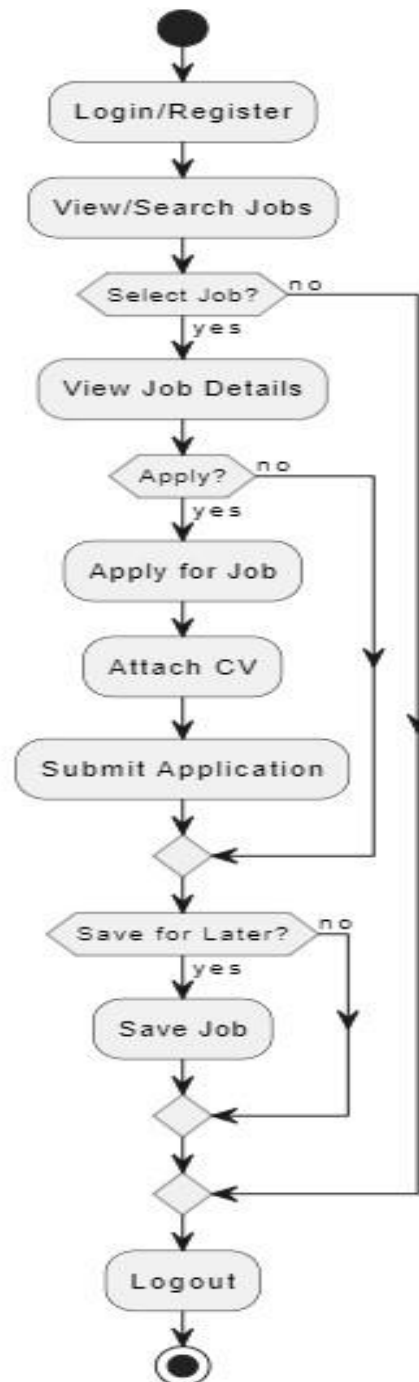**Context Saved Jobs**
Invariants:

- uniqueSavedJobId: self.allInstances()->forAll(s1, s2 | s1 <> s2 implies s1.saved_job_id <> s2.saved_job_id)
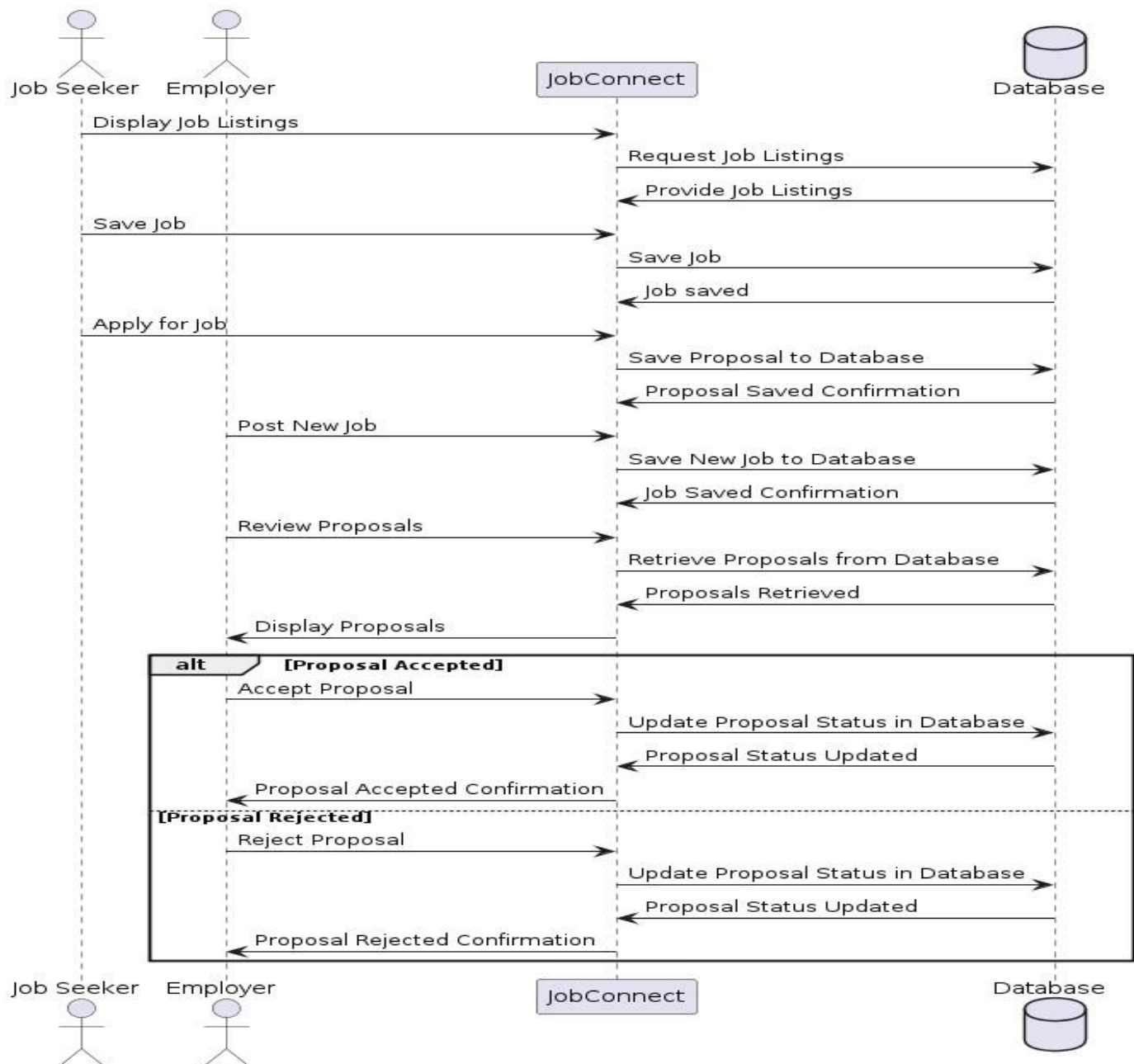
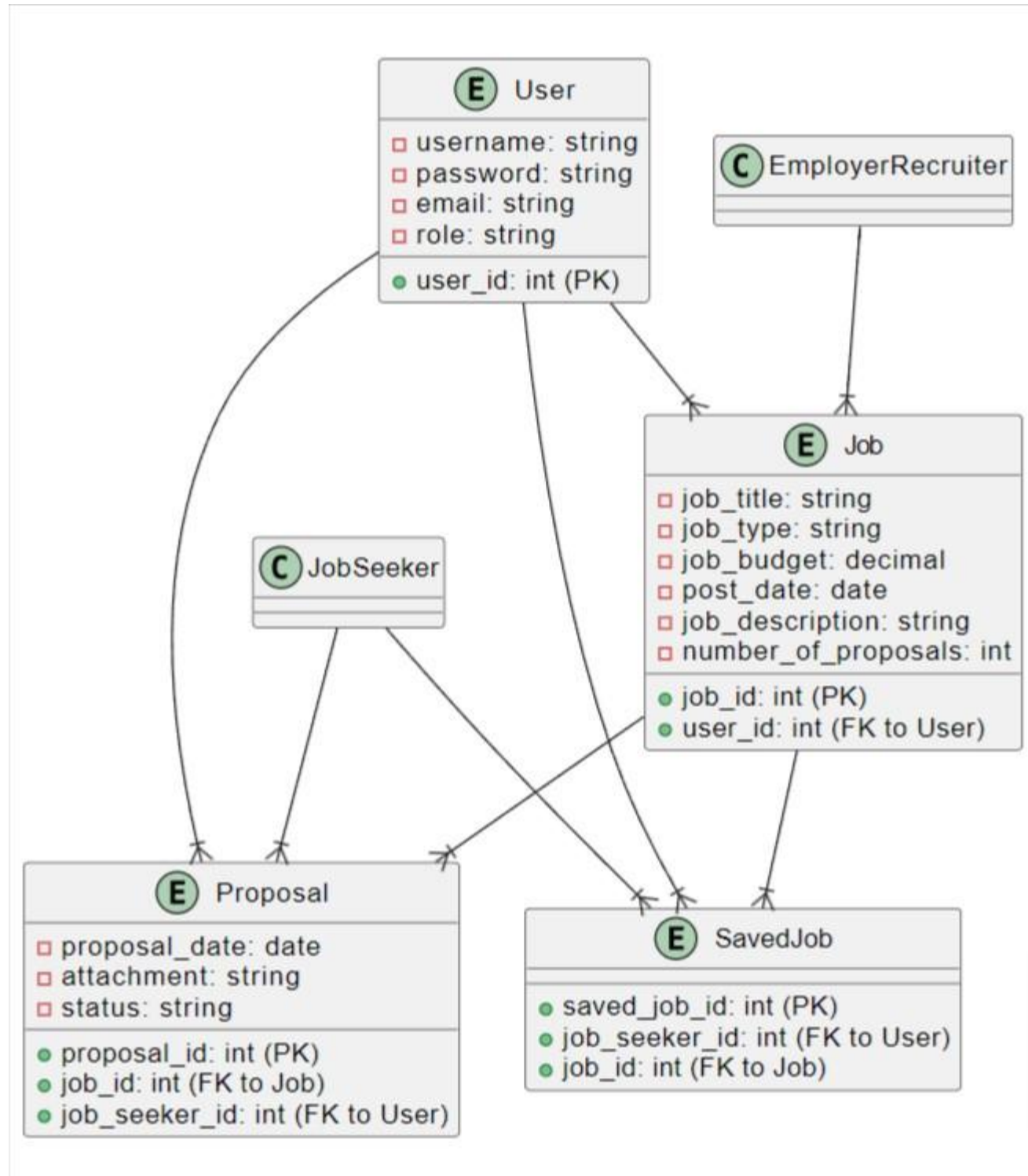# Activity Diagram

Employer Activity Flow on JobConnect
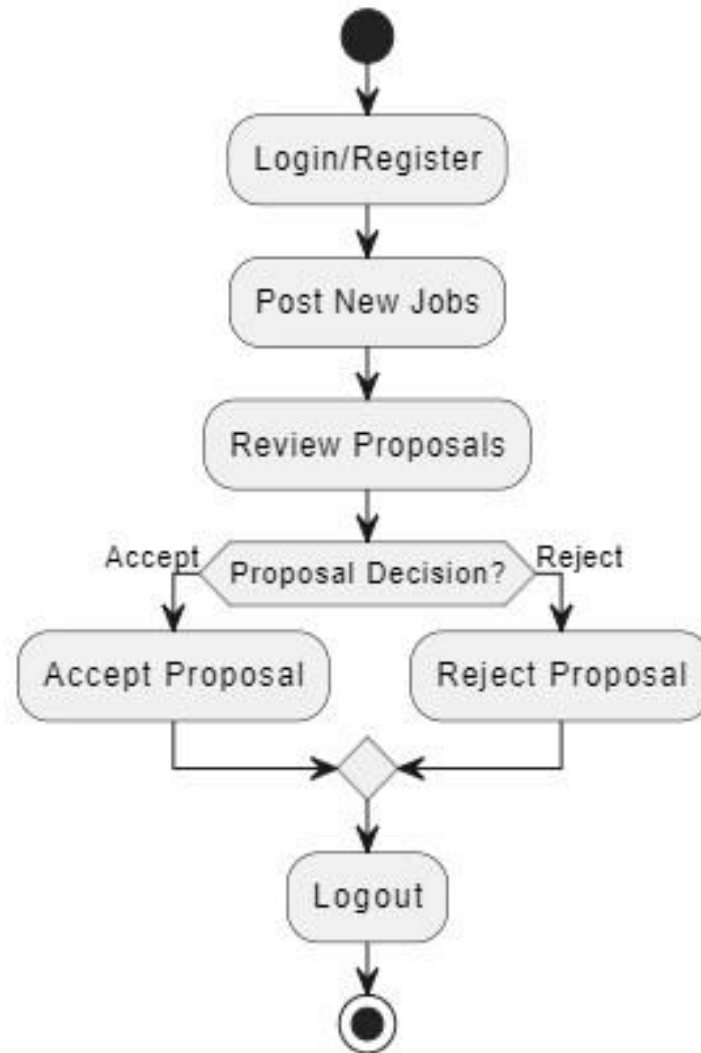
Job Seeker Activity Flow on JobConnect

Sequence Diagram

Database Specification

System architecture diagram(client server)