SOFTWARE REQUIREMENTS SPECIFICATION
JOBNODE: SEARCH & GET RECOMMENDED

Submitted By:
Sheikh Nahian | BSSE-1403
Shahid-E-Kaiser Md. Tashrif | BSSE-1448

Supervised By:
Dr. Naushin Nower
Professor
Institute of Information Technology
University of Dhaka

Submitted On: December 09, 2024

Table of Contents

1. QFD

1.1 Normal Requirements:

1. User Registration and Login:
Users and companies will be able to create accounts by providing necessary details such as name, email, and password. Users will be able to login with their username and password.

2. User Profile :
Job seekers can create personalized profiles providing skills, experience, location, and salary expectations. Companies will be able to provide email, industry, location, used skills, job-positions.
3. Job Posting:
Companies can post jobs with descriptions, required skills, location, salary, and experience.
4. Recommendation:
Users will get job recommendations according to their skills, salary expectation and experience.
5. Job Searching:
Job seekers can use filters like location, skills, and salary.
6. Job Application:
Job seekers apply for a job and provide resumes which are sent to company email.
7. Hiring Management:
Companies can keep a record of which job seeker got hired to which job post.

1.2 Expected Requirements:

1. Authentication:
User accounts will be authorized and authenticated.
2. Application Tracking:
Users can track the status of their application.
3. Role-Based Access Control:
Job seekers and companies will have different features and permissions based on their account type.

4. Chat System:

- Company: Companies will be able to initiate chat with applicants to transfer text or files.
- Job Seekers: Job seekers will be able to start a chat after the company initiates it and transfer text or files.

1.3 Exciting Requirements:

1. Mock Interview/Question:
Mock interviews or questions will be automatically generated by prompting the chatbot.
2. Graphical Insights:
Users will have an analytics view of their overall skills and gaps.

2. User Story

2.1 User Management

A user will sign up to access the platform by providing their name,email and password. The password will be at least 6 characters long. If the email exists, the user is informed that the email is already registered and they are prompted to log in instead. If the email does not exist, the user is registered successfully. The user can log in into the system after registration providing their email and password. After the credentials are validated the user is redirected to their dashboard. The user can choose to reset the password by providing their email. An OTP is sent to the validated email. The user can reset their password ensuring the new password meets the required criteria.

2.2 Job Post Management

A company user can post a job with : Job title, Job description, Required skills, Experience level, Salary range and Location. If the inputs are valid (e.g., all required fields are filled) the job post becomes visible to job seekers in the search results.A company can view a list of all their job posts in the dashboard. They can edit job details (e.g., update the salary or description), delete outdated job posts.

Job seekers can use filters like location, skills, and salary to narrow down the results and apply if interested.

2.3 Job Application Management

A job seeker can apply for a job by uploading their resume (PDF or Word format). Once submitted, the application details (resume and applicant info) are sent to the company’s email. The job seeker can view a list of jobs they’ve applied to, check the status of each application (e.g., Pending, Reviewed, Shortlisted, Rejected) and withdraw an application if needed.

Companies can view all applicants for a specific job post in their dashboard. For each applicant, the system displays the applicant's profile (skills, experience, location) and uploaded resume. Companies can filter applicants based on skills or experience and update application statuses (e.g., Shortlisted, Rejected).

2.4 Recommendation

Job seekers are recommended job posts that match with their skills. The user can click on a recommended job post to view details or apply directly. If a user updates their profile (e.g., adds new skills or changes salary expectations), the recommendations are refreshed to show the updated preferences.

2.5 Chat Management

A Company can initiate a chat or conversation with an applicant to discuss job roles, share details about openings, or clarify any queries related to the hiring process.

Users can exchange real-time messages . They can also manage their chat history and share files with each other.

2.6 Hiring Management

A company can view and manage job applications, update their status to "Hired" or "Rejected," and input offer letter details for hired candidates. After confirmation, the system sends the offer letter to the selected applicant through email.

A job seeker can track the status of their applications (e.g., Pending, Rejected, Hired). If hired, they can view the offer letter and accept or reject it, also they can download the offer letter sent by the company.

3. USE CASE DIAGRAM

Level: 0
USE CASE ID: 0

Name: JobNode: Search and Get Recommended
Primary Actor: Job Seeker, Company
Secondary Actor: Mail Service

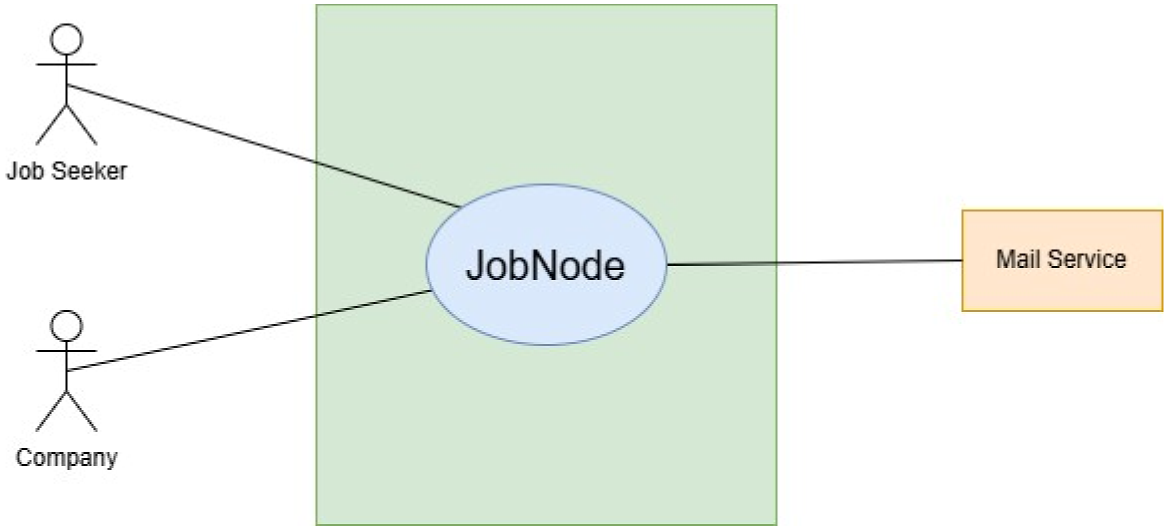


Fig1: JobNode: Search and Get Recommended

Level: 1
USE CASE ID: 1

Name: JobNode: Search and Get Recommended
Primary Actor: Job Seeker, Company

Secondary Actor: Mail Service

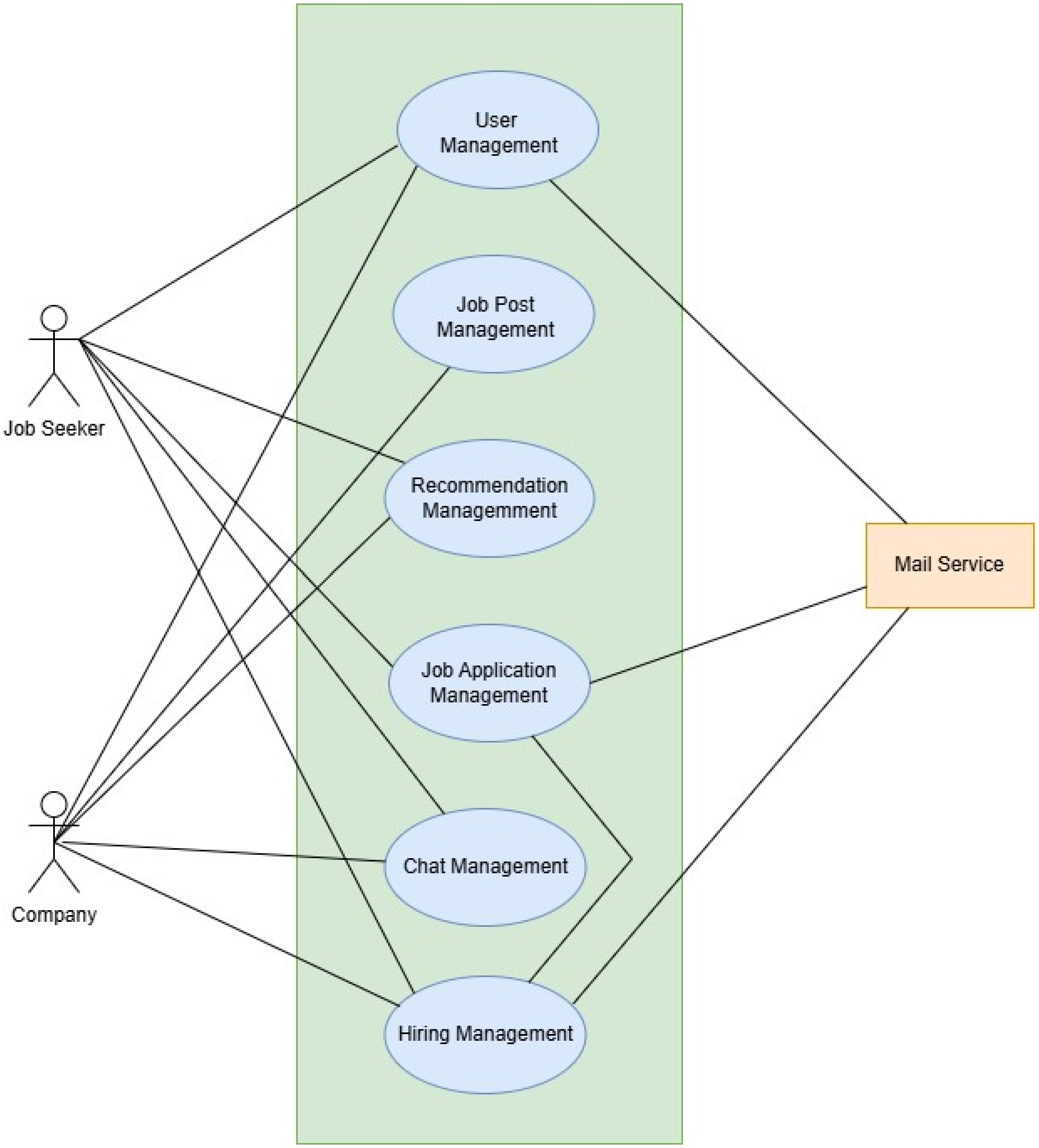


Fig2: JobNode: Search and Get Recommended(Detailed)

Level: 1.1
USE CASE ID: 1.1

Name: User Management

Primary Actor: Job Seeker, Company

Secondary Actor: None

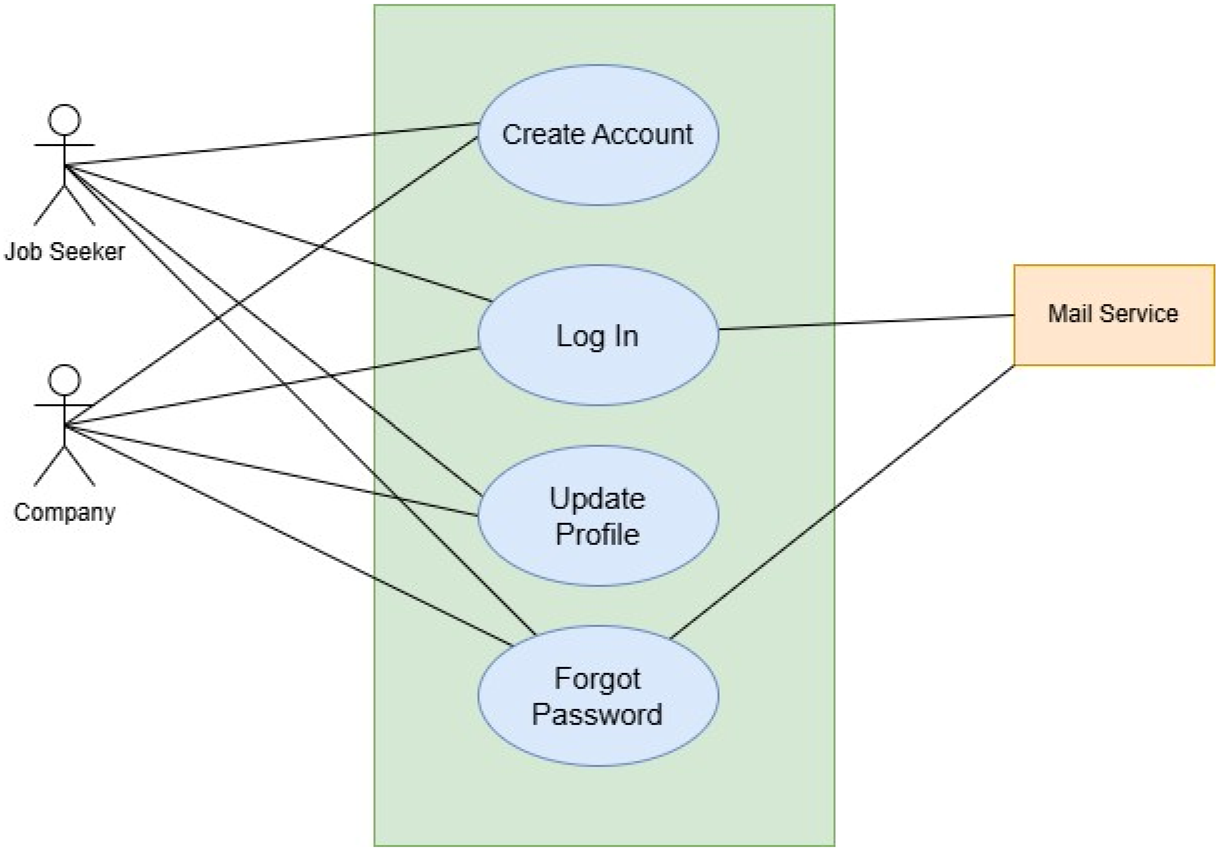


Fig3: User Management

Level: 1.2

USE CASE ID: 1.2

Name: Job Post Management

Primary Actor: Company

Secondary Actor: None

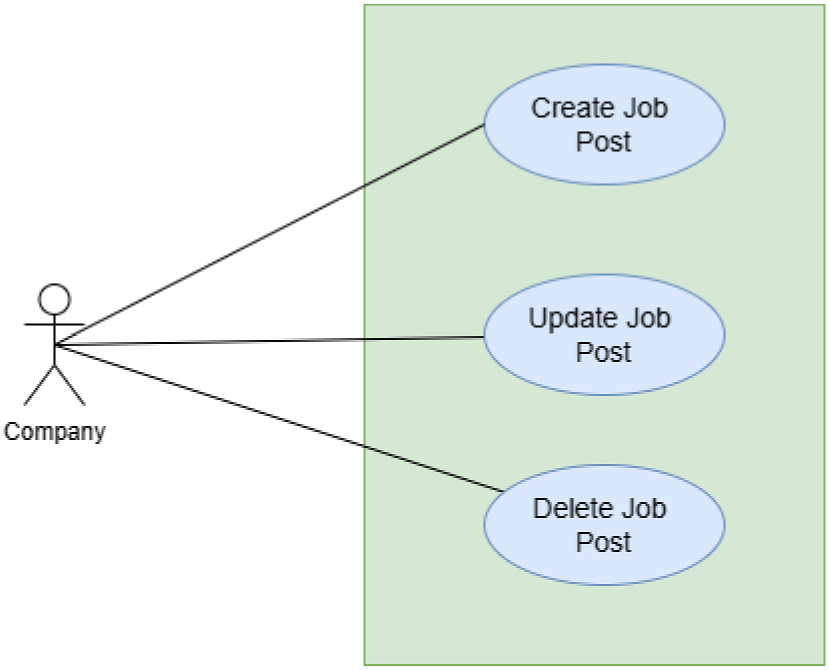


Fig4: Job Post Management

Level: 1.3
USE CASE ID: 1.3

Name: Recommendation Management
Primary Actor: Job Seeker
Secondary Actor: Company

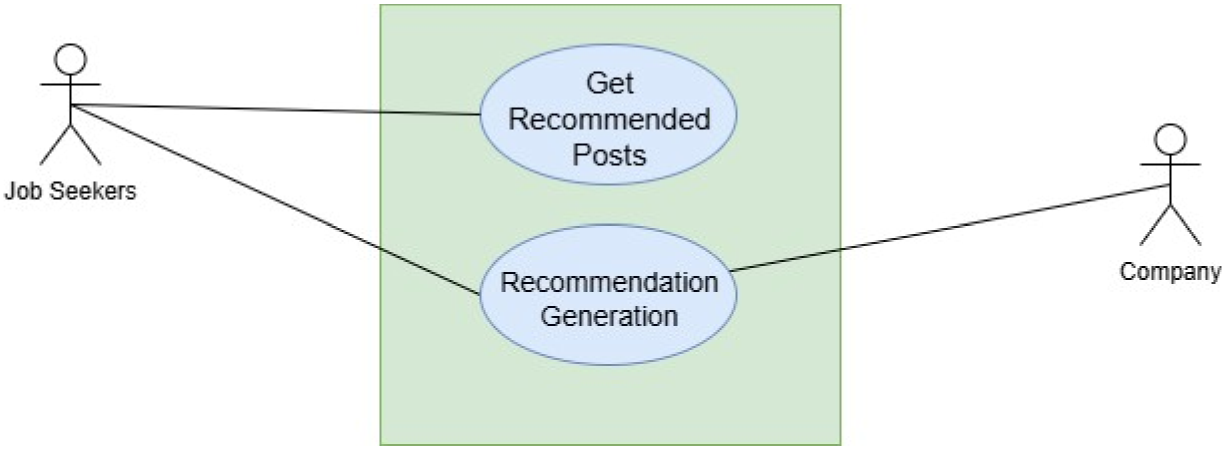


Fig5: Recommendation Management

Level: 1.4
USE CASE ID: 1.4

Name: Job Application Management

Primary Actor: Job Seeker

Secondary Actor: Mail Service

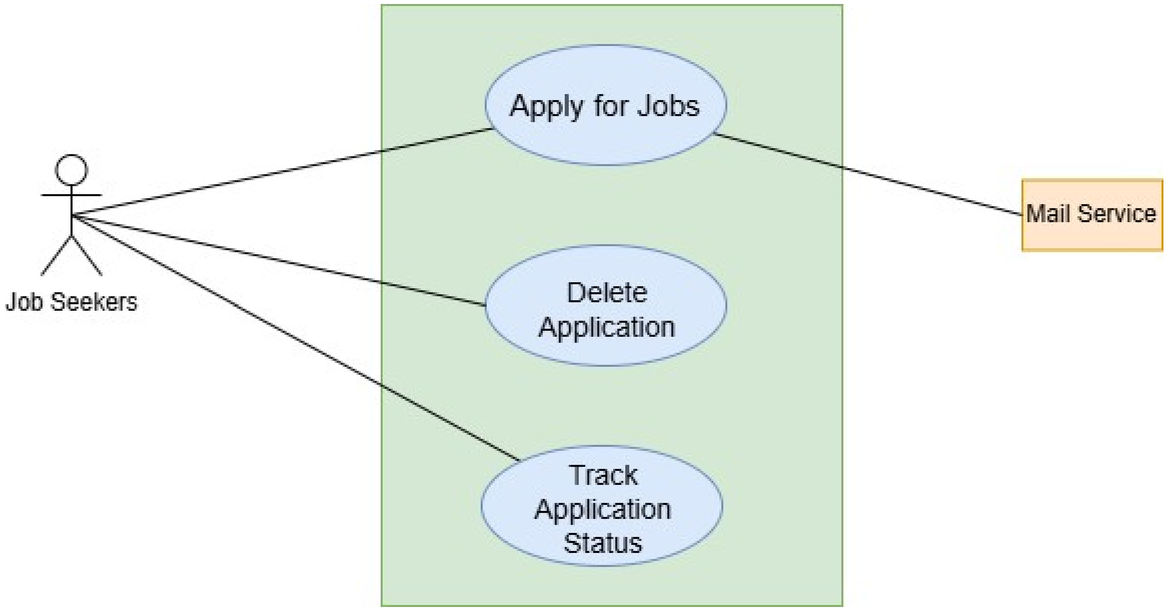


Fig6: Job Application Management

Level: 1.5
USE CASE ID: 1.5

Name: Chat Management

Primary Actor: Company, Job seeker

Secondary Actor: None

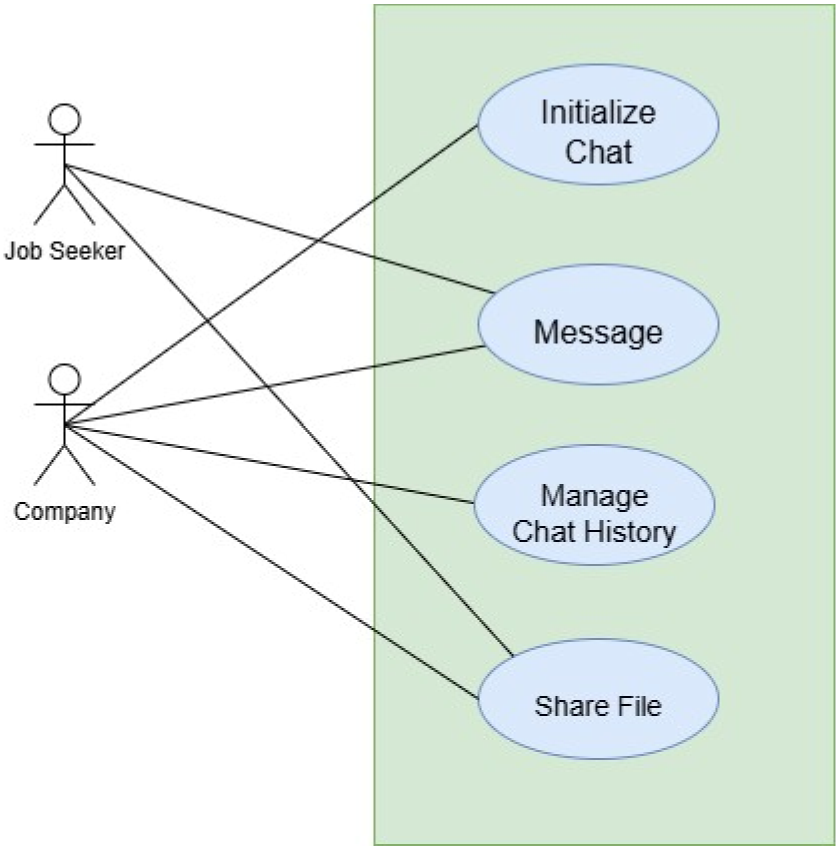


Fig7: Chat Management

Level: 1.6
USE CASE ID: 1.6

Name: Hiring Management

Primary Actor: Company

Secondary Actor: Mail Service

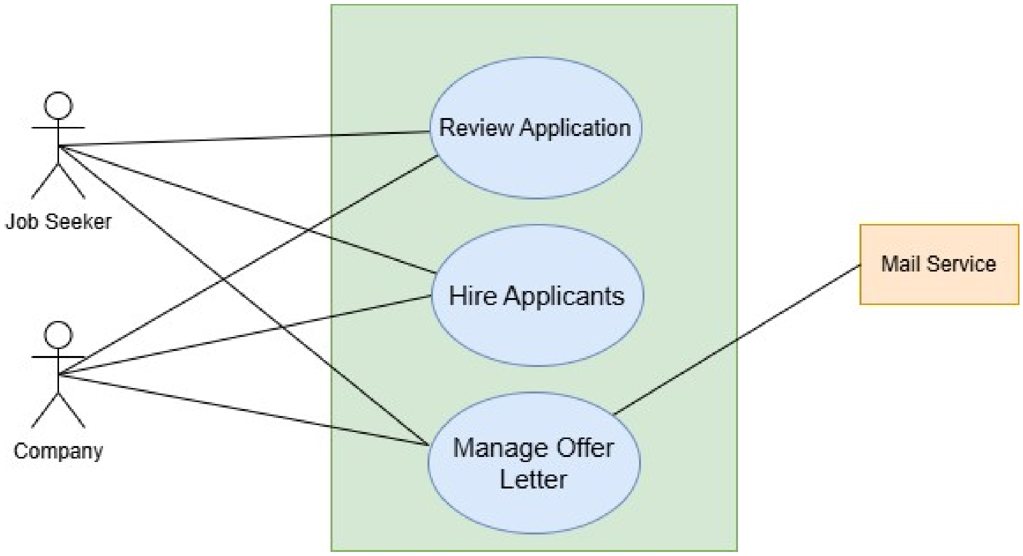


Fig8: Hiring Management

4. Activity Diagram

Level: 1

Name: Jobnode: Search and Get Recommended
Reference: Use Case Diagram Level-1

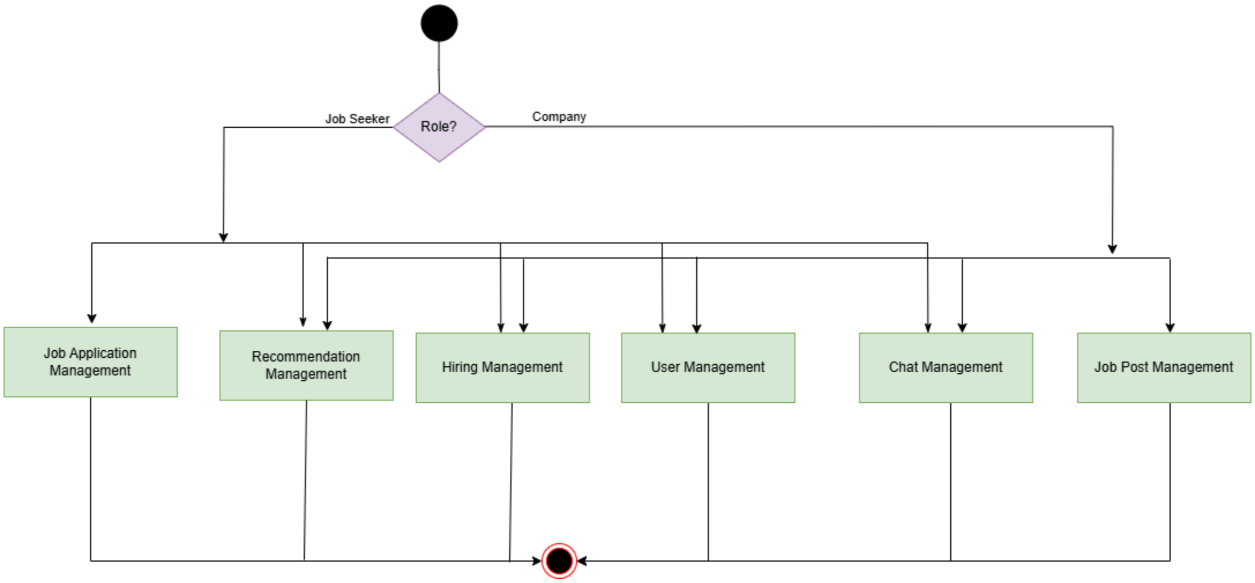


Fig9: JobNode: Search and Get Recommended(Activity Level-1)

Level: 1.1

Name: User Management

Reference: Use Case Diagram Level-1.1

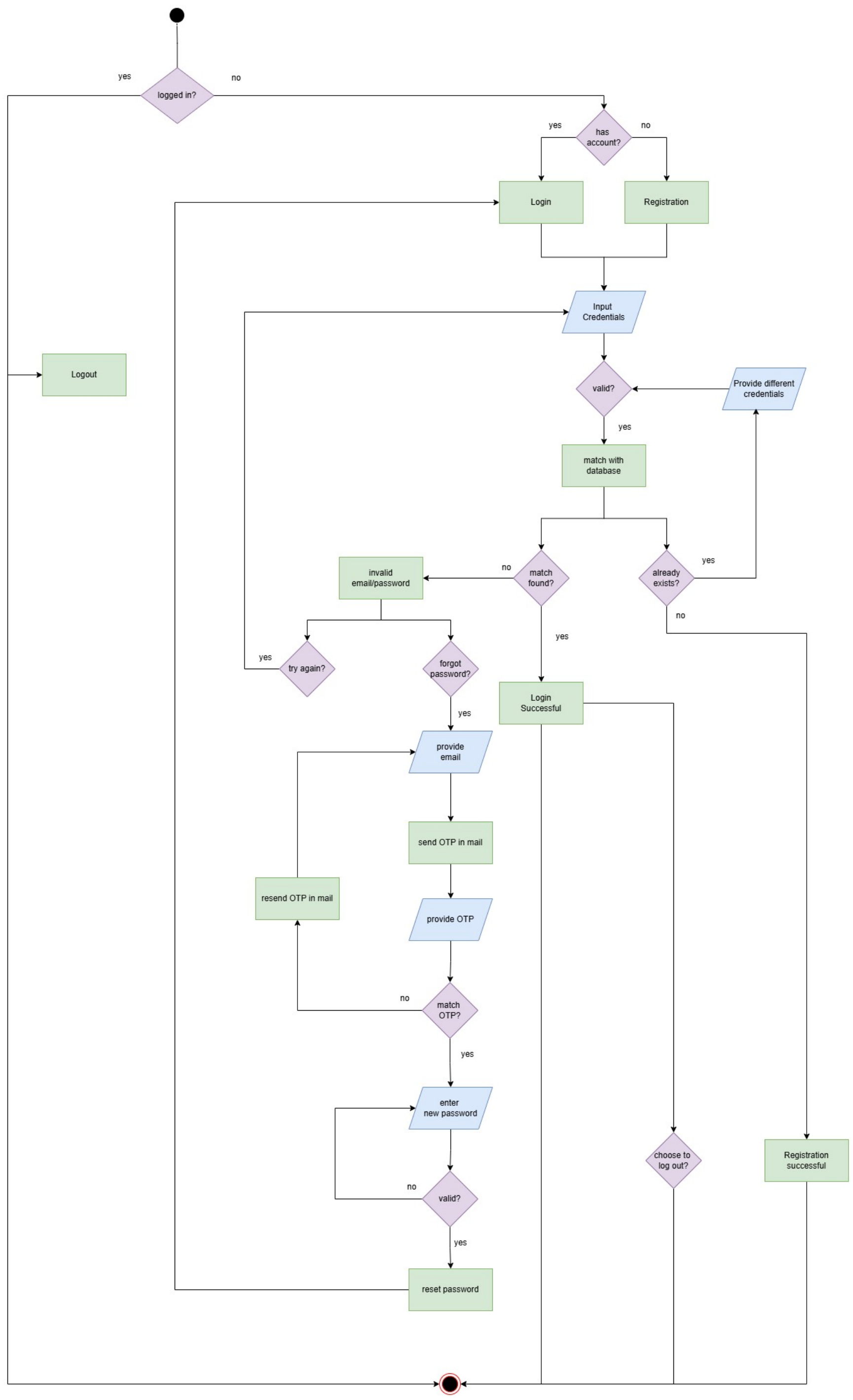


Fig10: User Management(Activity Level-1.1)

Level: 1.2

Name: Job Post Management

Reference: Use Case Diagram Level-1.2

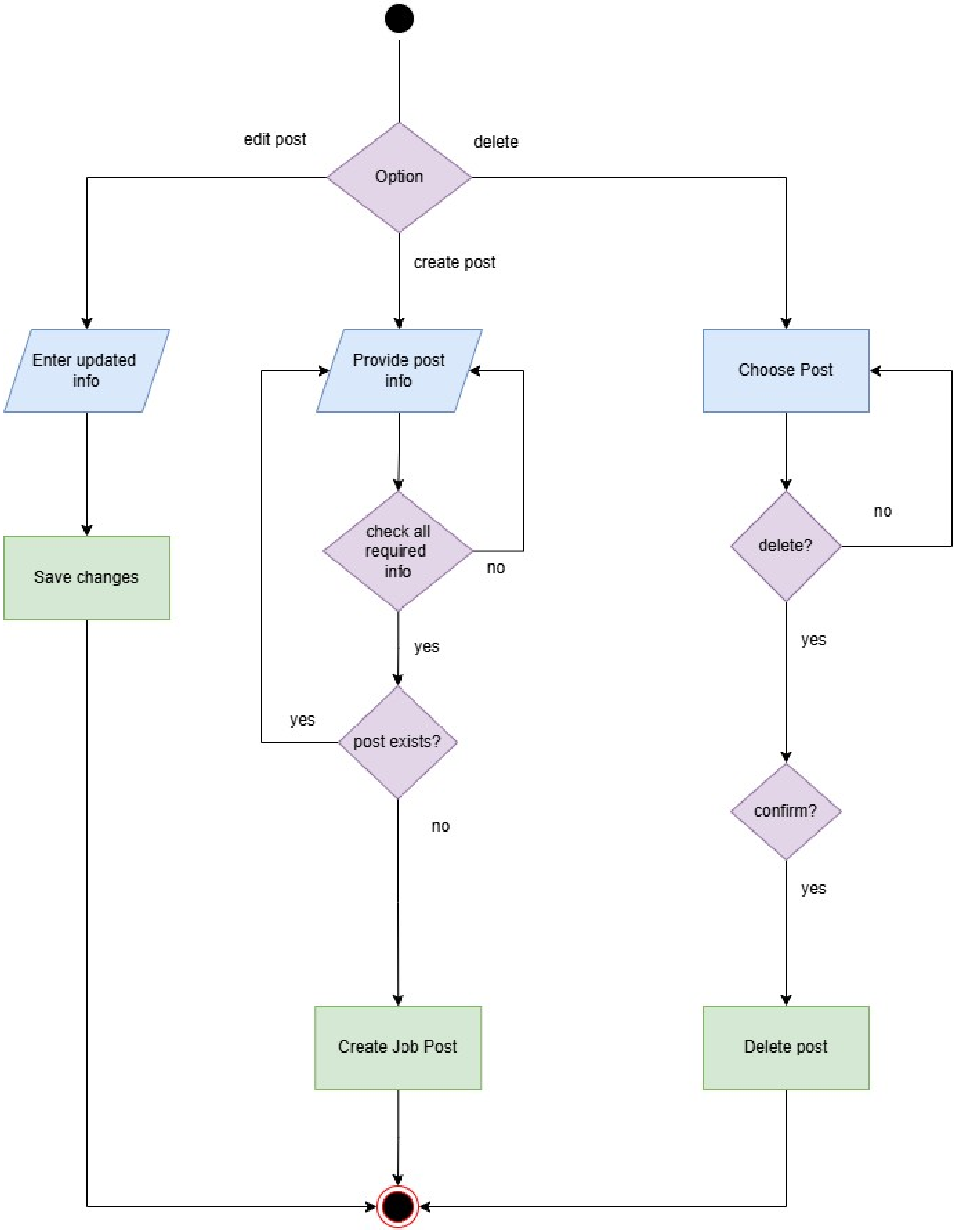


Fig11: Job Post Management(Activity Level-1.2)

Level: 1.3

Name: Recommendation Management

Reference: Use Case Diagram Level-1.3

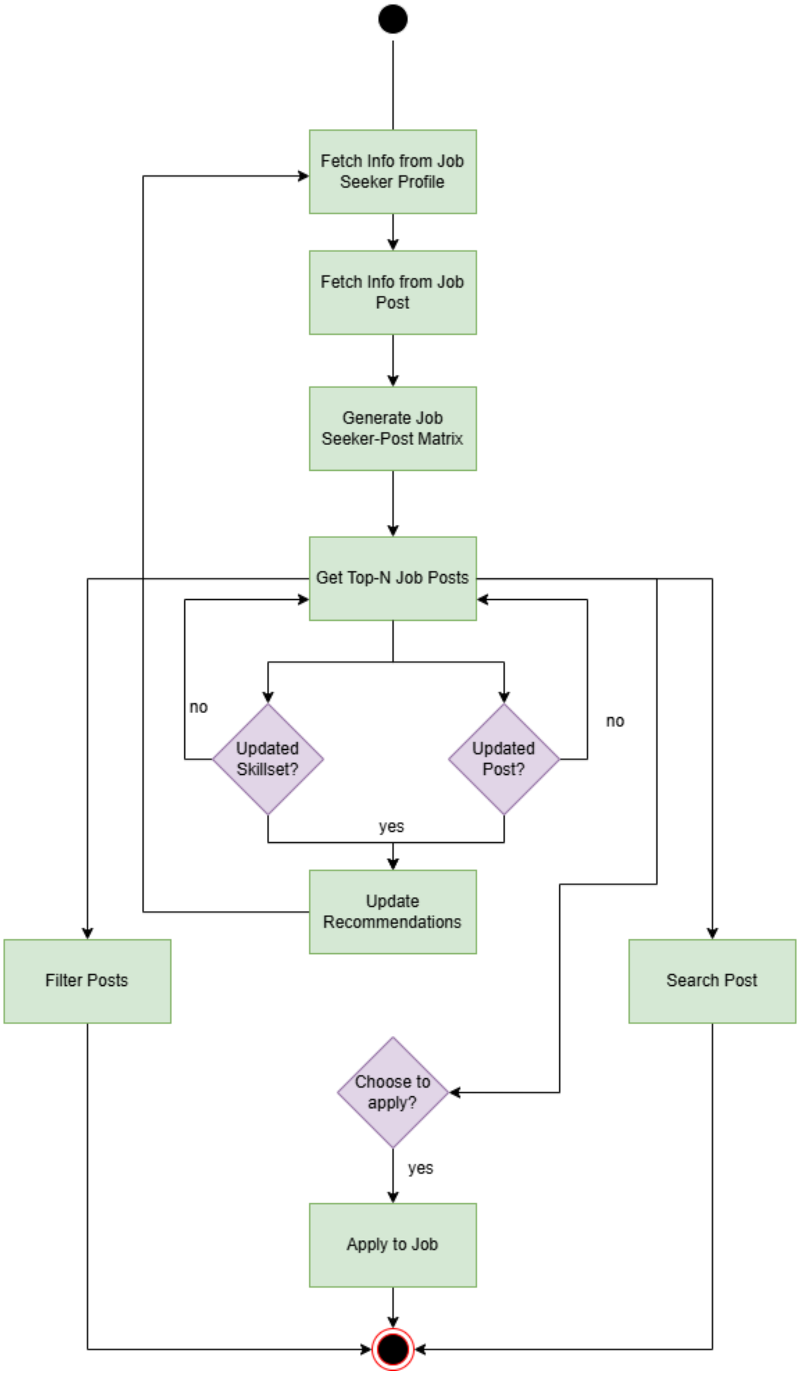


Fig12: Recommendation Management(Activity Level-1.3)

Level: 1.4
Name: Job Application Management
Reference: Use Case Diagram Level-1.4

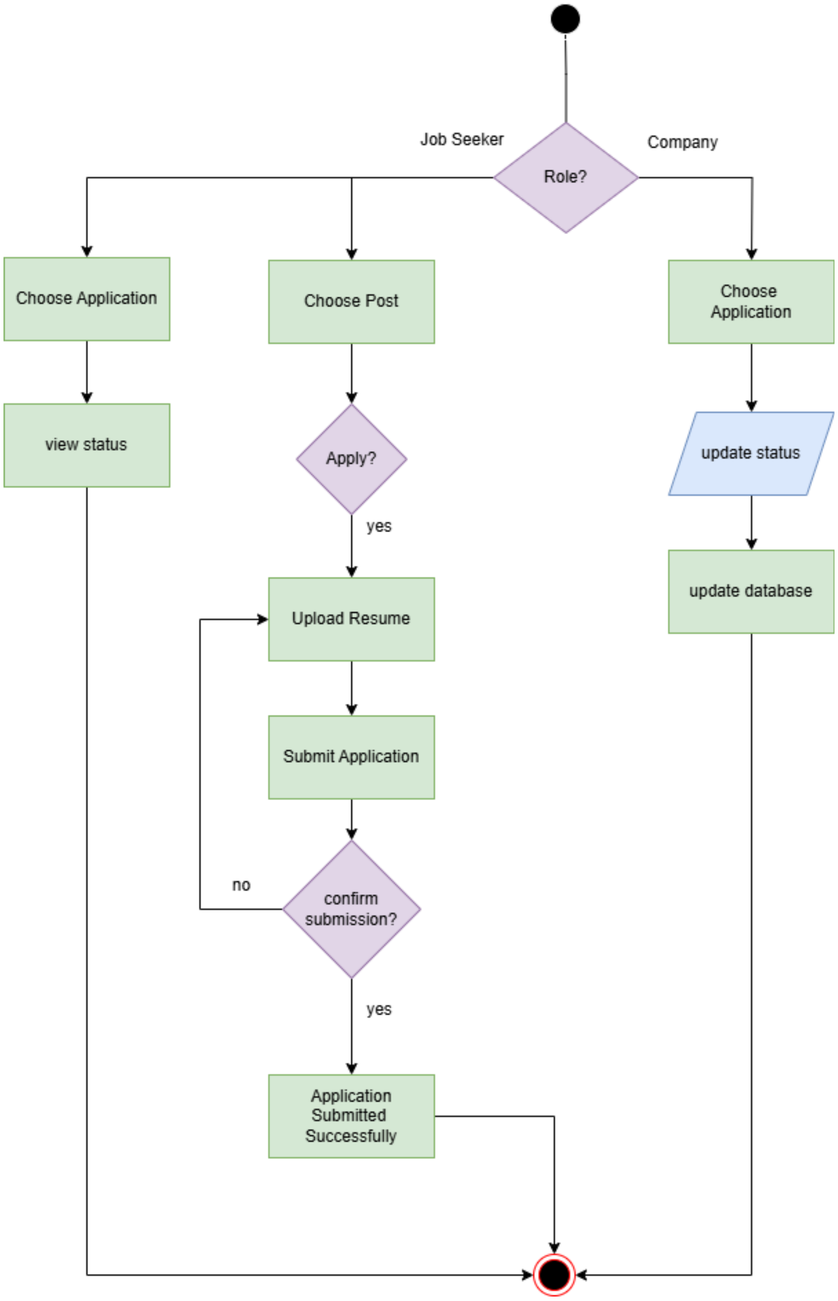


Fig13: Job Application Management(Activity Level-1.4)

Level: 1.5
Name: Chat Management
Reference: Use Case Diagram Level-1.5

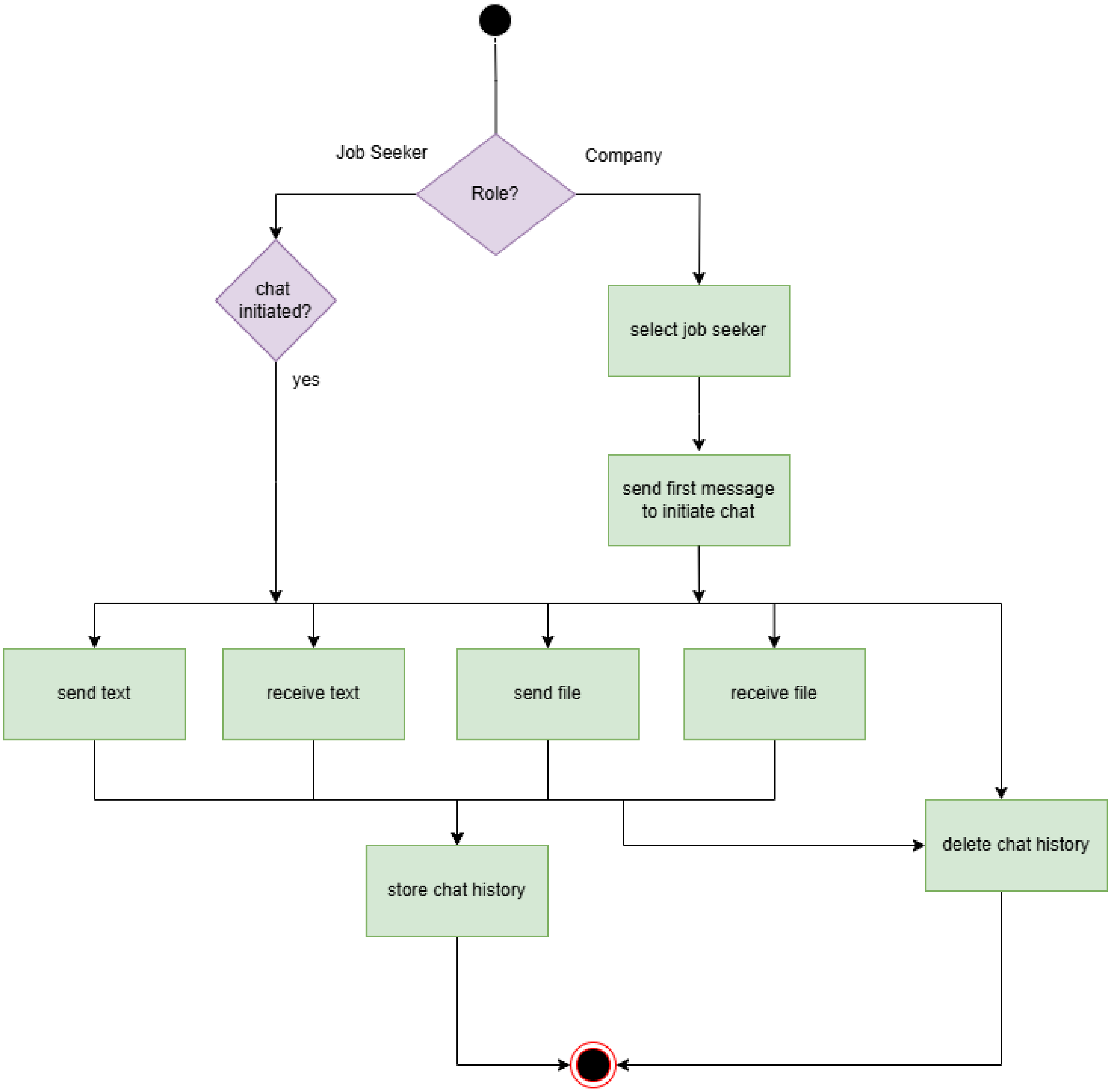


Fig14: Chat Management(Activity Level-1.5)

Level: 1.6

Name: Hiring Management

Reference: Use Case Diagram Level-1.6

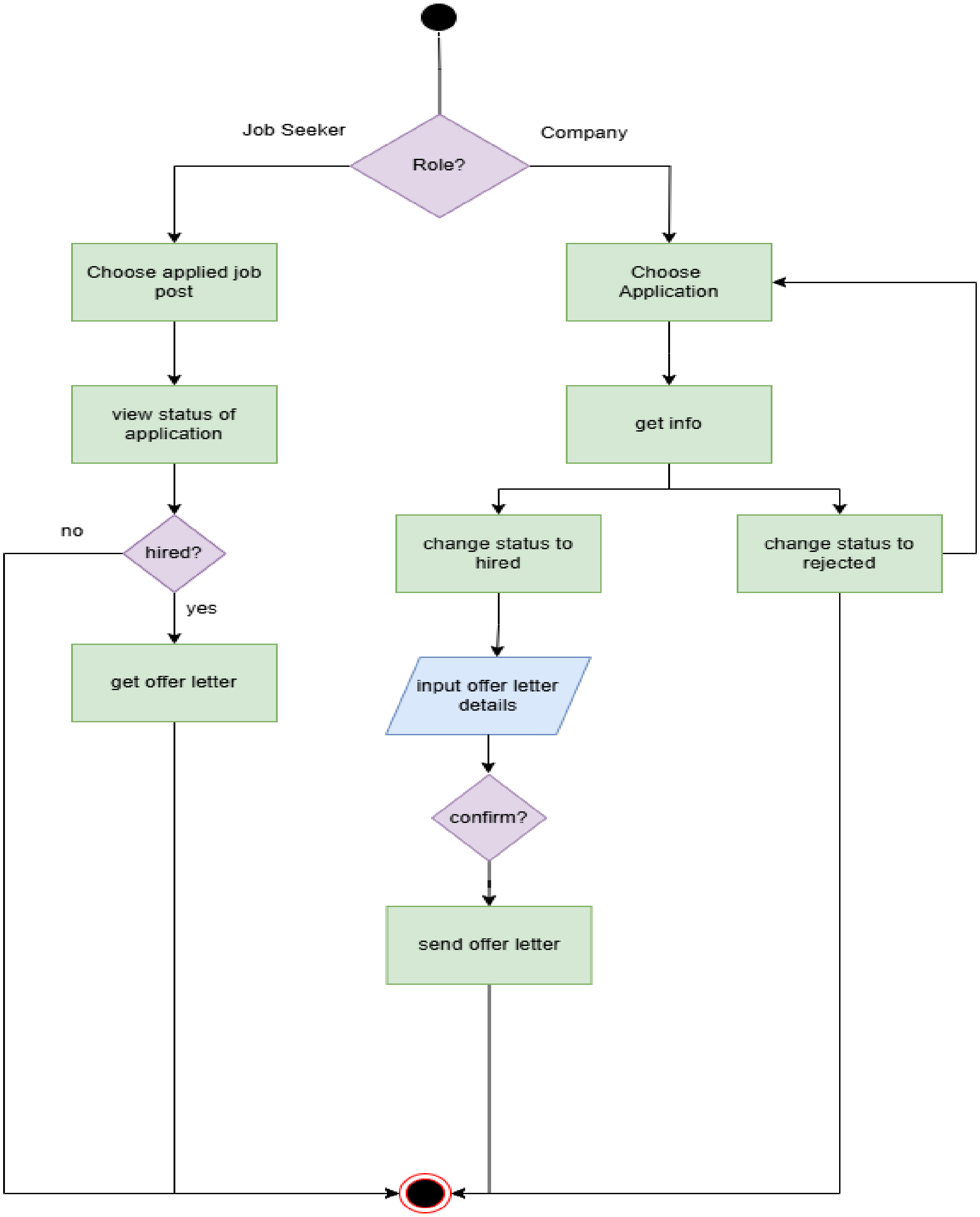


Fig15: Hiring Management(Activity Level-1.6)

5. Swimlane Diagram

Level: 1

Name: Jobnode: Search and Get Recommended
Reference: Use Case & Activity Diagram Level-1

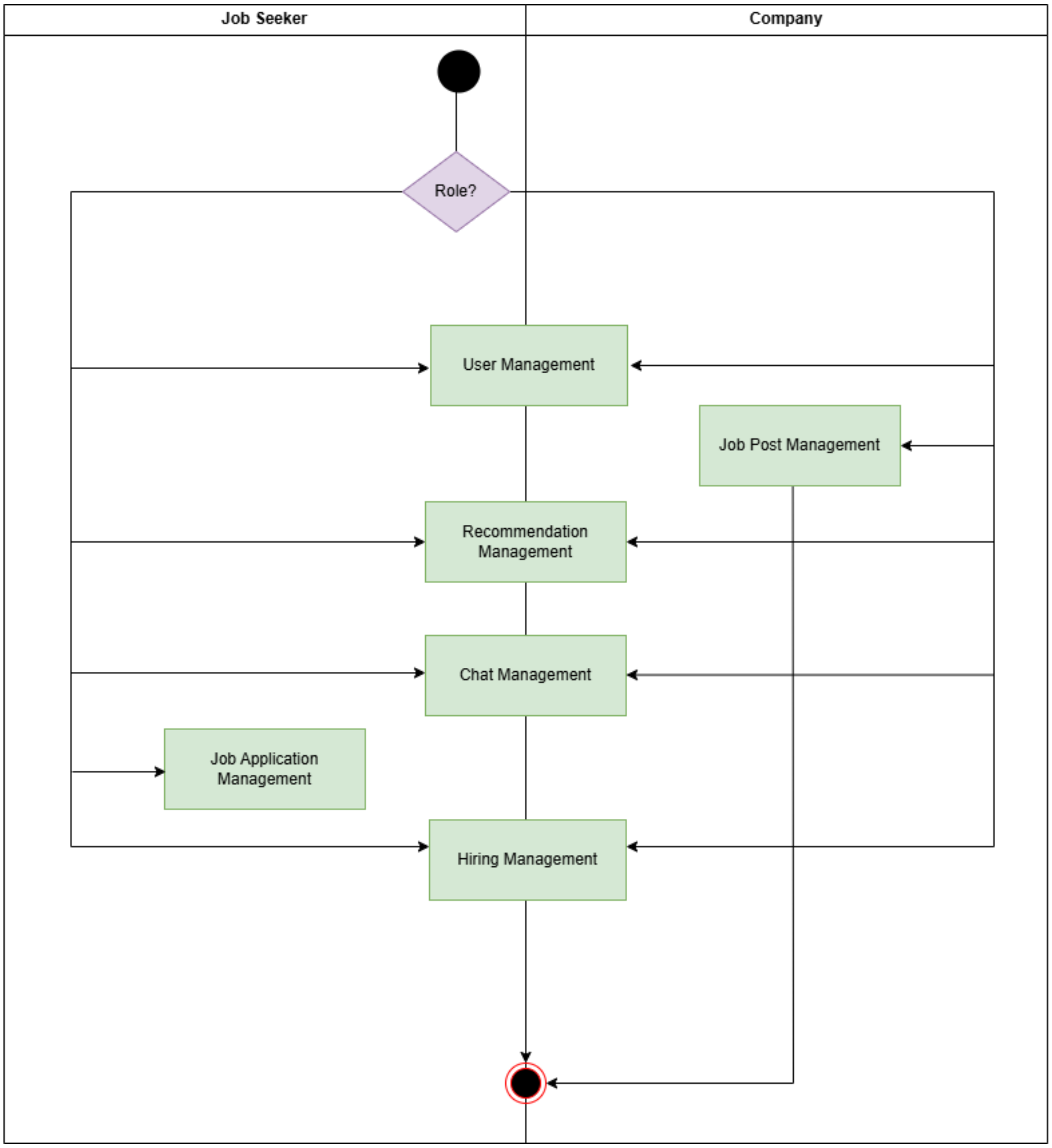


Fig16: JobNode: Search & Get Recommended(Swimlane level-1)

Level: 1.1

Name: User Management

Reference: Use Case & Activity Diagram Level-1.1

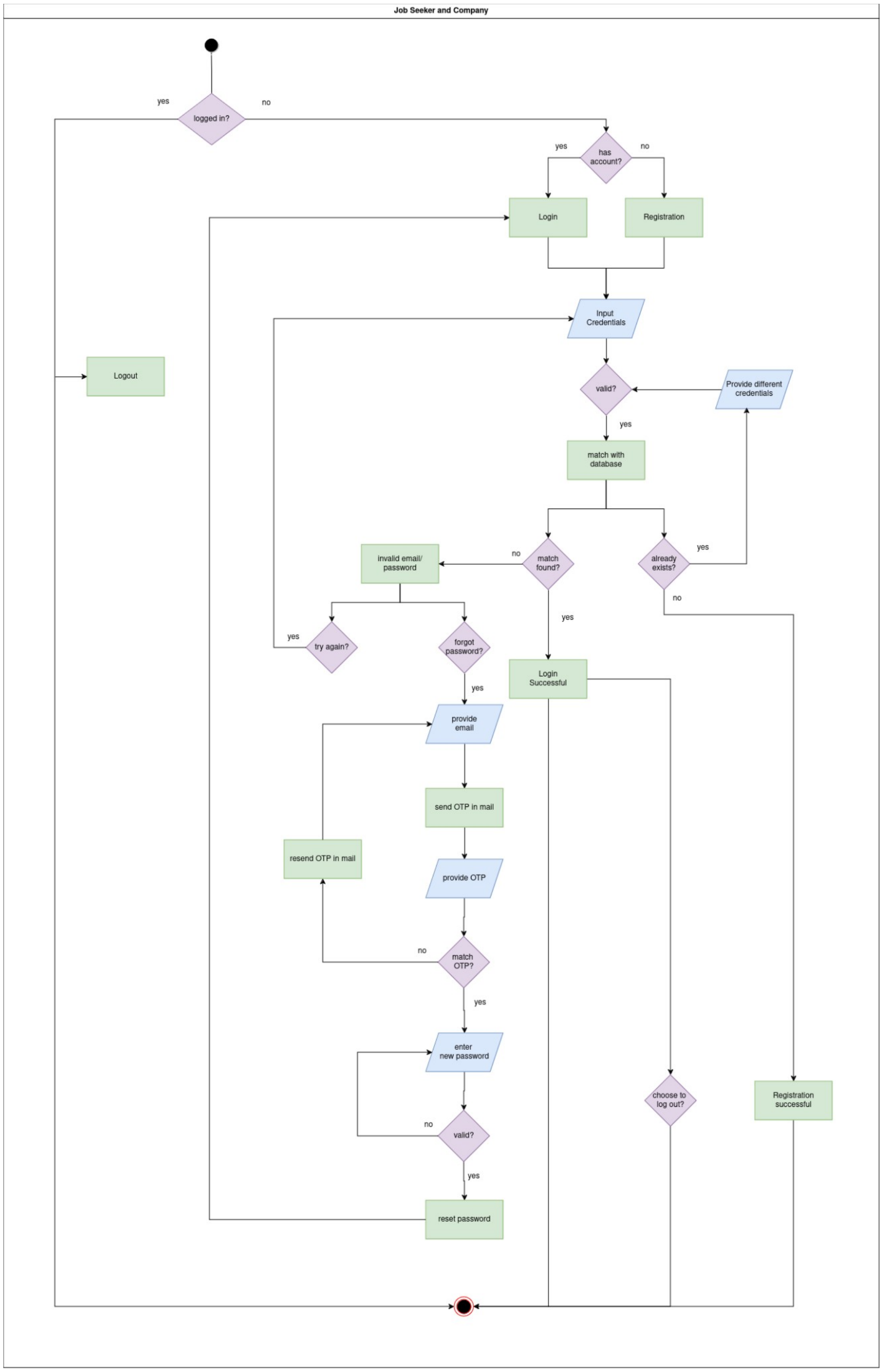


Fig17: User Management(Swimlane level-1.1)

Level: 1.2

Name: Job Post Management

Reference: Use Case & Activity Diagram Level-1.2

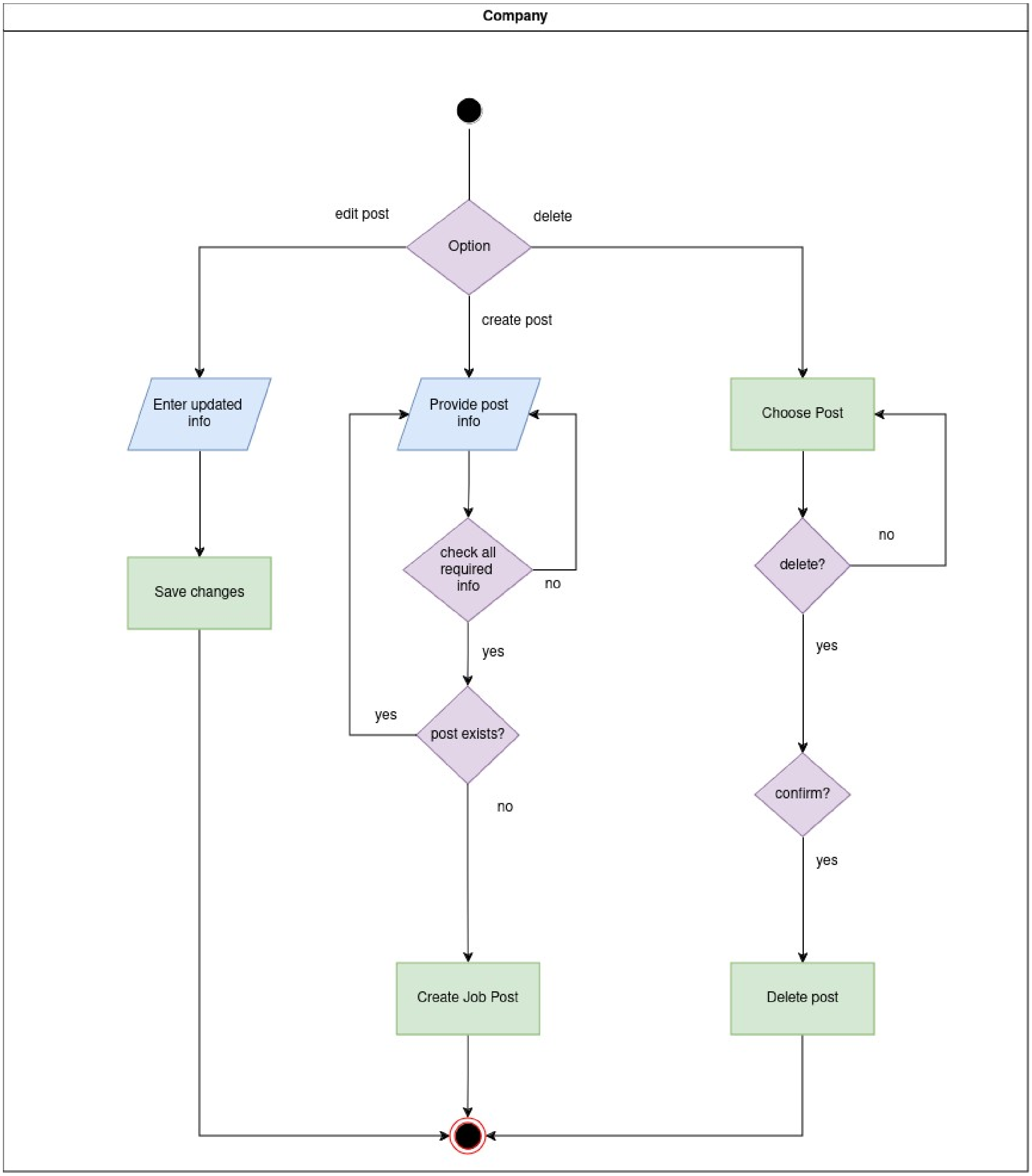


Fig18: Job Post Management(Swimlane level-1.2)

Level: 1.3

Name: Recommendation Management

Reference: Use Case & Activity Diagram Level-1.3

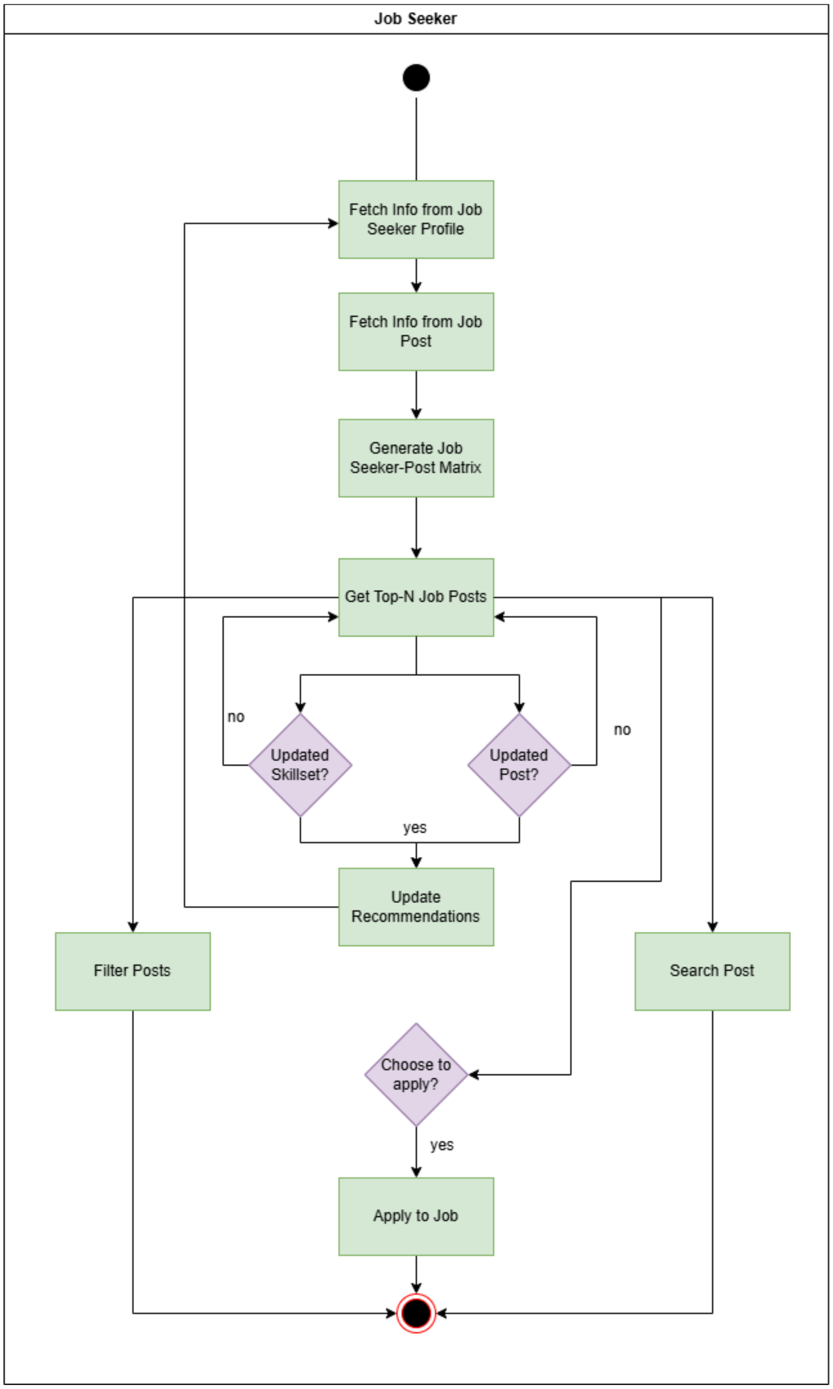


Fig19: Recommendation Management(Swimlane level-1.3)

Level: 1.4

Name: Job Application Management

Reference: Use Case & Activity Diagram Level-1.4

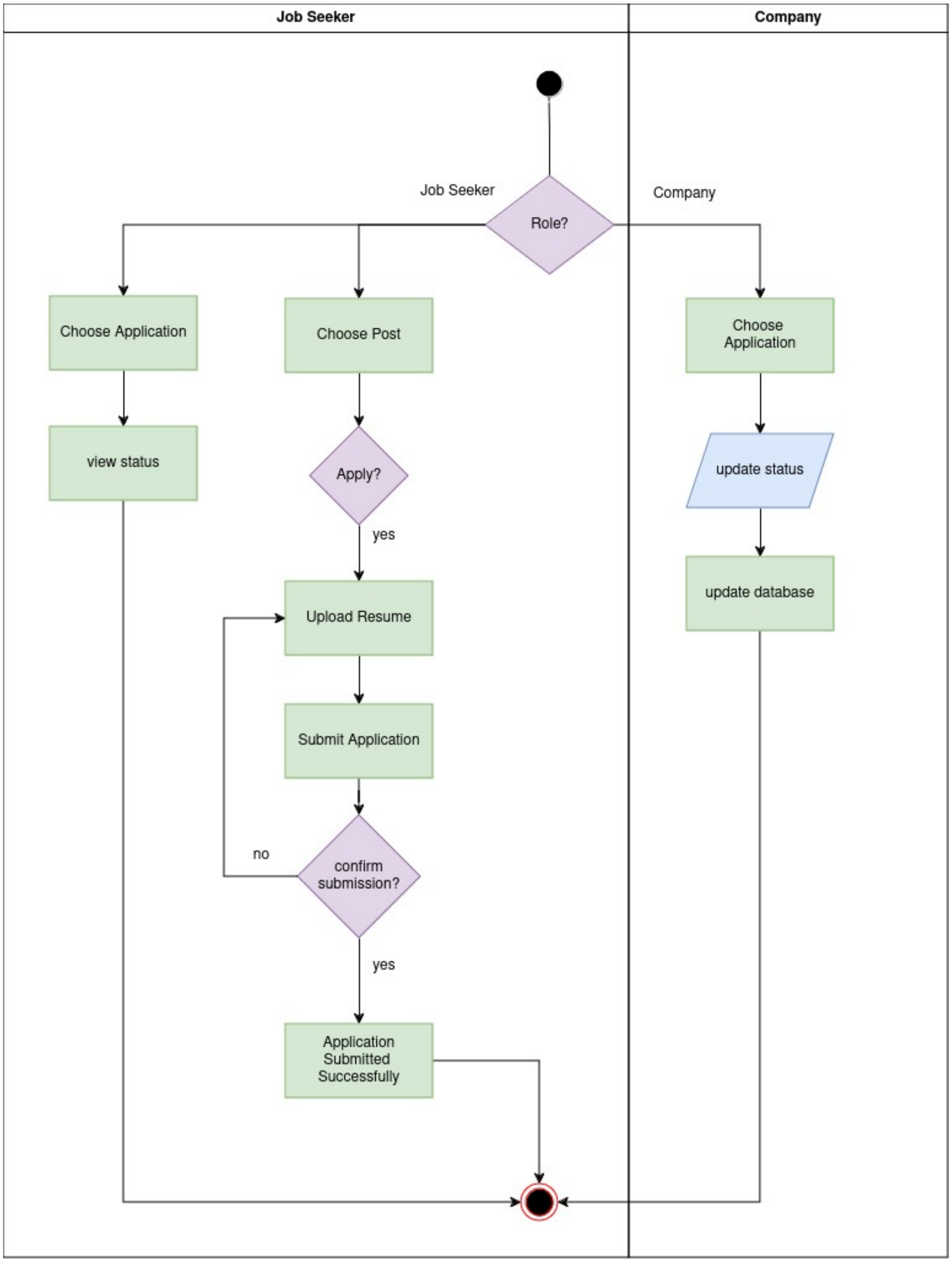


Fig20: Job Application Management(Swimlane level-1.4)

Level: 1.5

Name: Chat Management

Reference: Use Case & Activity Diagram Level-1.5

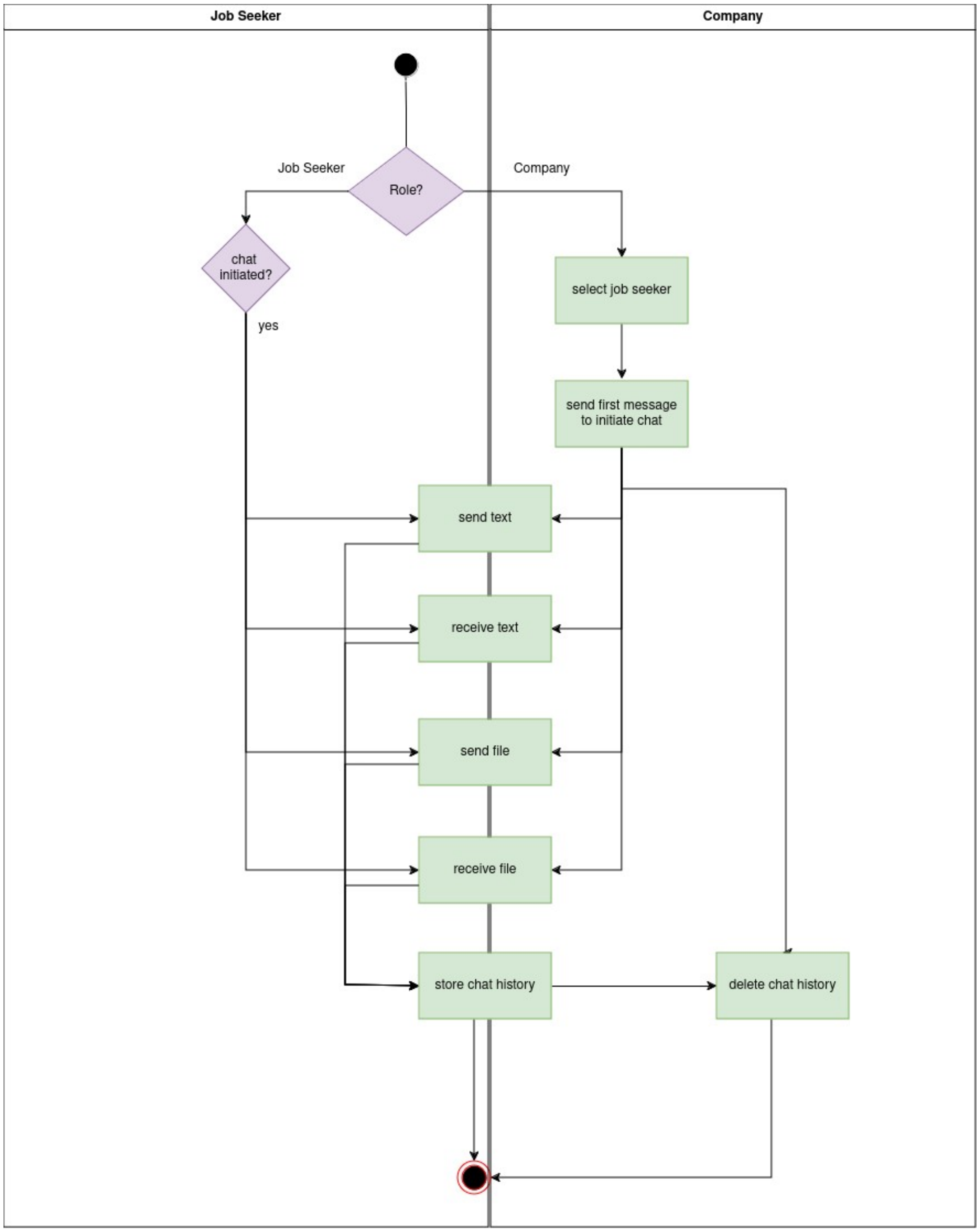


Fig21: Chat Management(Swimlane level-1.5)

Level: 1.6

Name: Hiring Management

Reference: Use Case & Activity Diagram Level-1.6

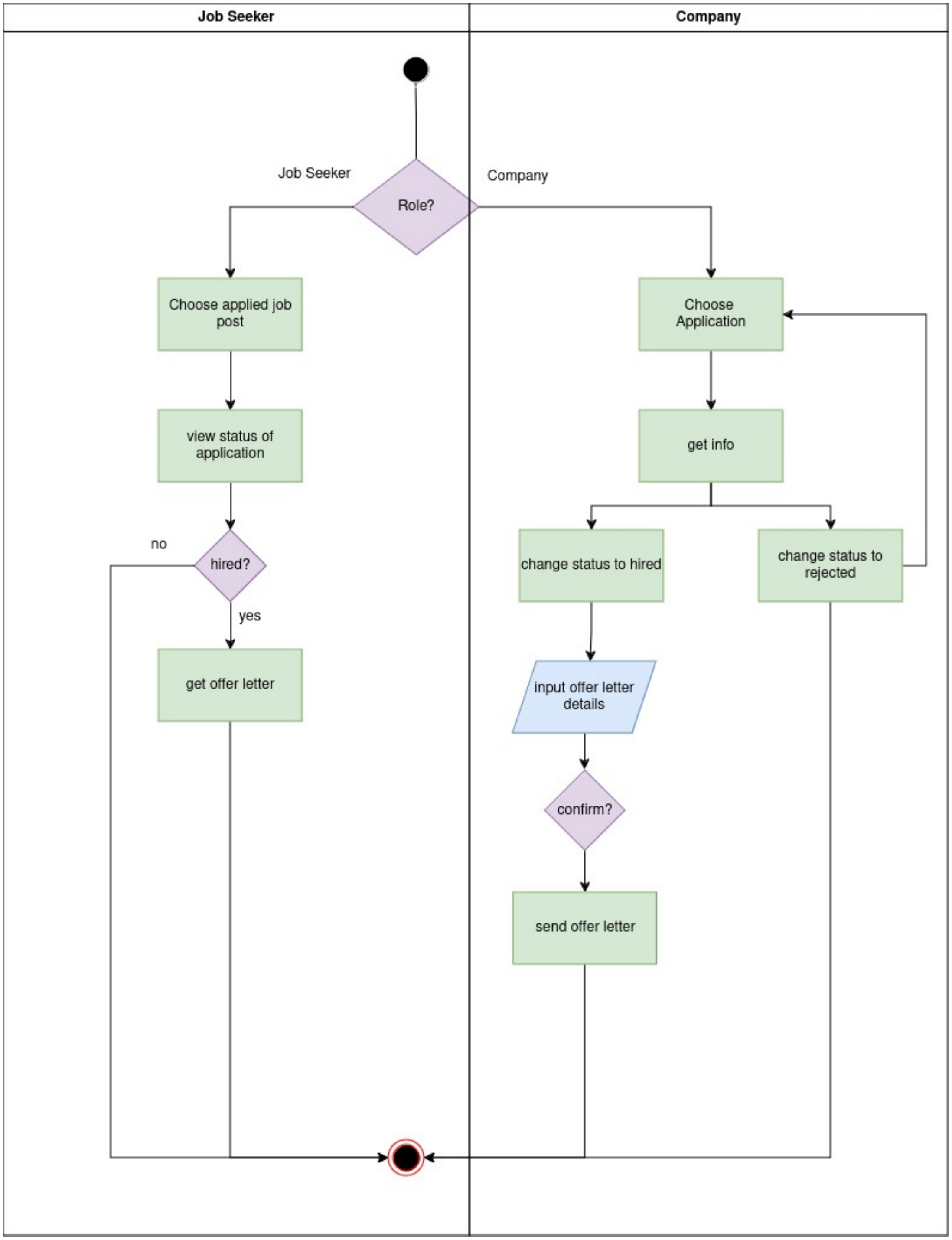


Fig22: Hiring Management(Swimlane level-1.6)

6. Data Based Modeling

6.1 Data Object Identification

SL	Noun	Problem/Solution Space	Attribute
1	User	s	3,4,5,14,15
2	Platform	p	
3	Name	s	1,14,15
4	Email	s	1,14,15
5	Password	s	1,14,15
6	Character	p	
7	Login	s	11
8	System	p	
9	Credential	s	1
10	Dashboard	p	
11	OTP	s	7
12	Criteria	p	
13	Job Post	s	14,15,16,17,18,19,20,21,25
14	Company	s	3,4,5,13,15,35,44
15	Job Seeker	s	3,4,5,13,14,19,20,47,34,35
16	Job title	s	13
17	Job description	s	13
18	Job Required skills	s	13
19	Experience	s	13
20	Salary	s	13
21	Location	s	13

22	Input	p	
23	Field	p	
24	List	p	
25	Job details	s	13
26	Filter	p	
27	Skills	s	15,18,47
28	Job Application	s	14,15,29,30,31,32
29	Resume	s	28
30	Application details	s	28
31	Applicant info	p	
32	Status	s	28
33	Applicant	p	
34	Recommendation	s	15
35	Profile	s	14,15
36	Preference	p	
37	Chat	s	41,42,43,
38	Job Role	p	
39	Process	p	
40	Opening	p	
41	Message	s	14,15, 42,46
42	Chat History	s	41,46
43	File	s	41,46
44	Offer letter	s	41,46
45	Details	p	
46	Conversation	s	14,15,37,41,42,43
47	Job seeker’s skills	s	15

6.2 Final Data Objects

- User
- Job Seeker
- Company
- Job Post
- Skills
- JobSeekerSkills
- JobRequiredSkills

6.3 Relationship Between Data Objects

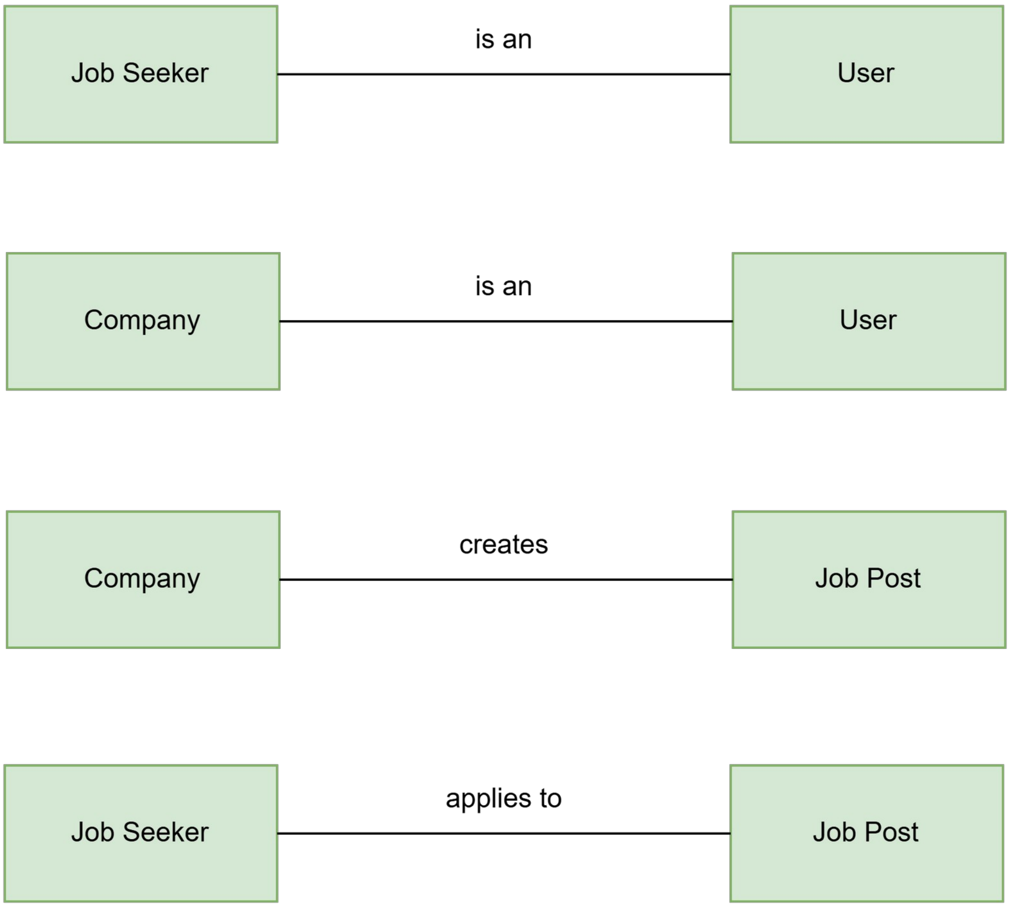
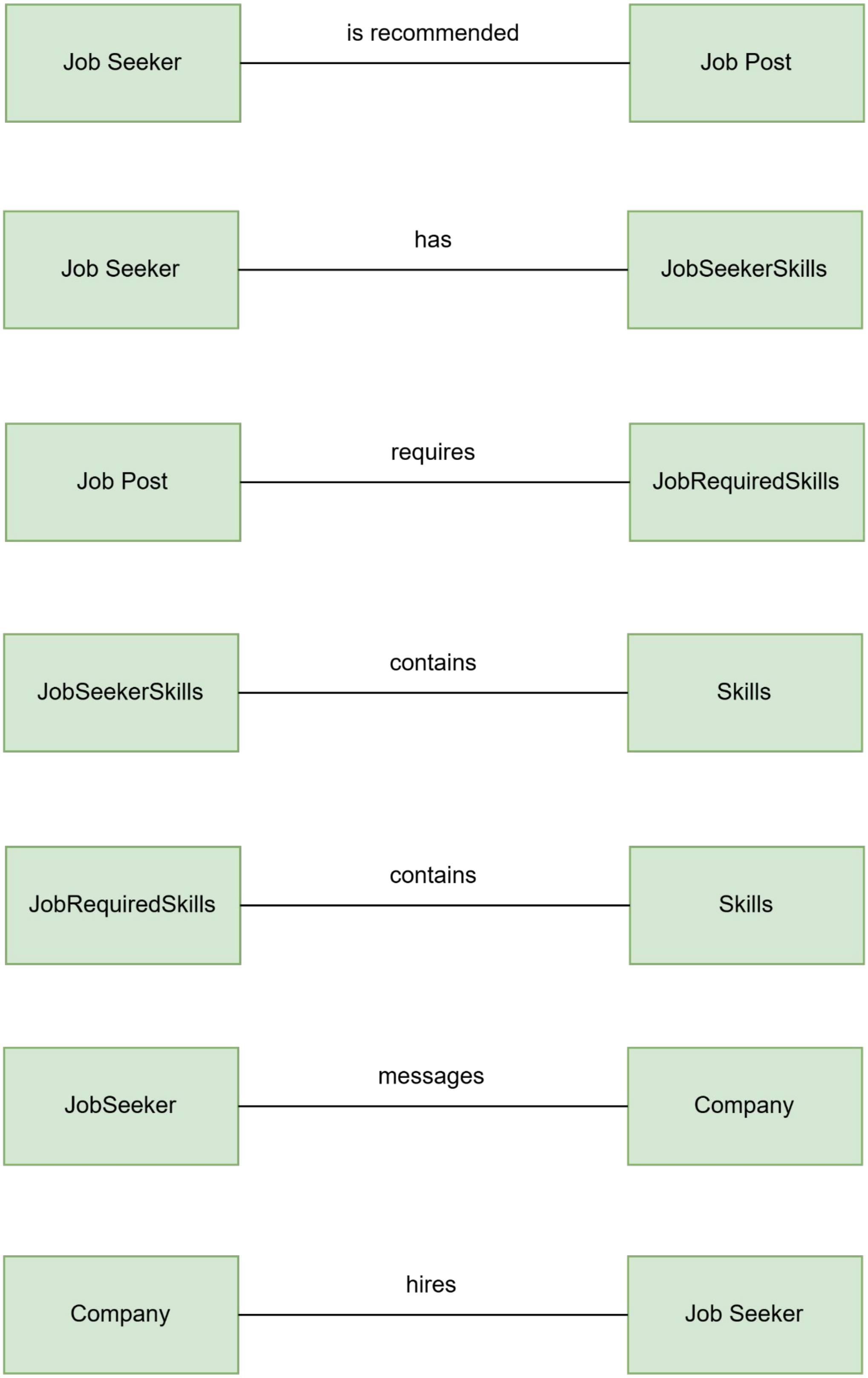


Fig23: Relationships between Data Objects


Job Seeker
is recommended
Job Post
Job Seeker
has
JobSeekerSkills
Job Post
requires
JobRequiredSkills
JobSeekerSkills
contains
Skills
JobRequiredSkills
contains
Skills
JobSeeker
messages
Company
Company
hires
Job Seeker

6.4 ER Diagram

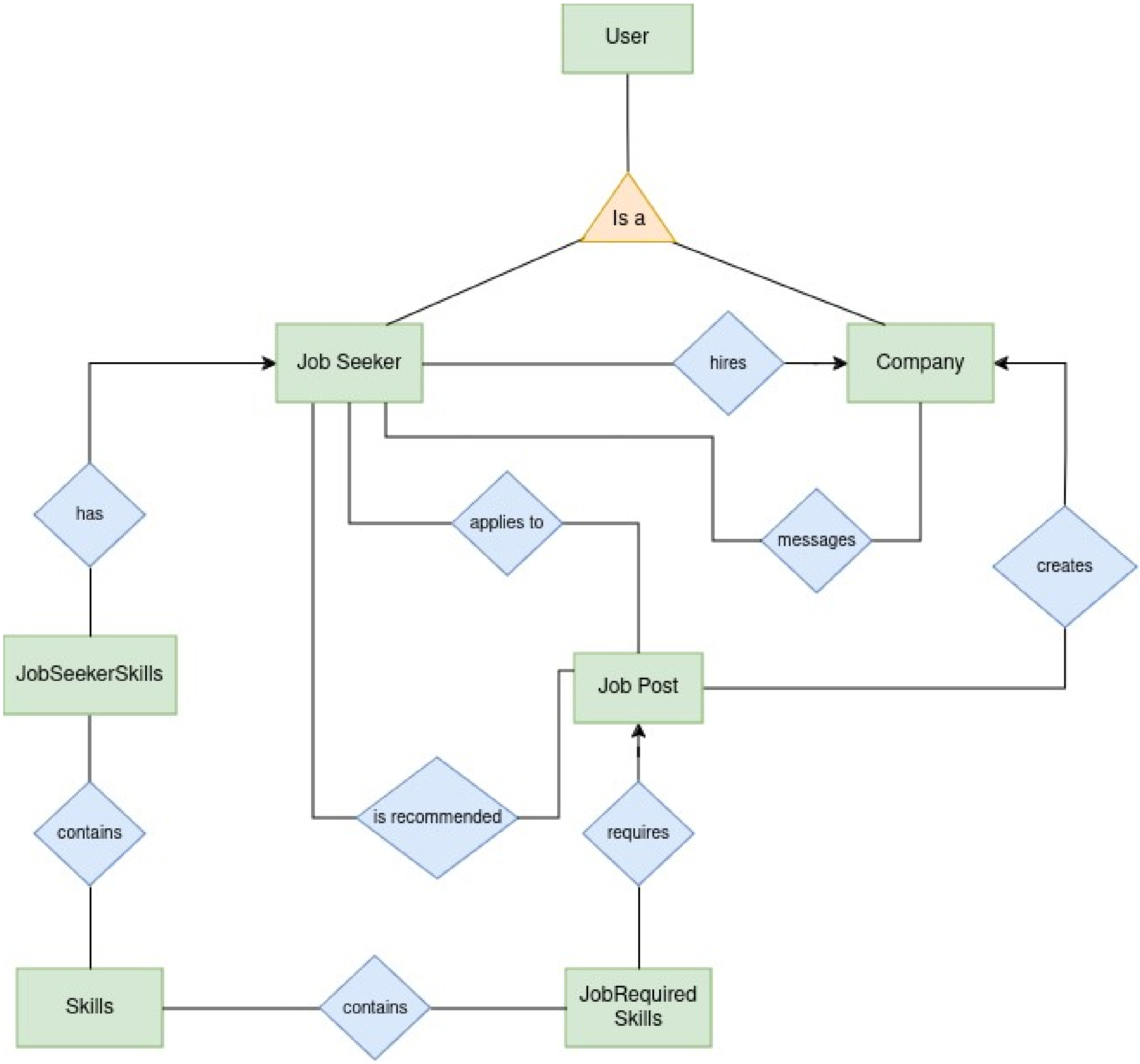


Fig24: ER Diagram of JobNode

6.5 Schema Diagram

Data Object	Attribute	Type	Size
User	UserID Username Email PhoneNumber Password AccountType	Int Varchar Varchar Varchar Varchar Iarchar	4 30 50 15 60 20
Job Seeker	JobSeekerID Skills Location SalaryExpectation Experience	Int Varchar Varchar Int Int	4 50 30 4 4
Company	CompanyID Industry Location JobPositions	Int Varchar Varchar varchar	4 30 30 30
Job Post	JobPostID CompanyID Title Description Location Salary RequiredSkills RequiredExperience DatePosted	Int Int Varchar Varchar Varchar Int Varchar Int DATE	4 4 30 30 30 4 30 4 3
Apply	ApplicationID JobSeekerID JobPostID IsApproved ApplicationDate	Int Int Int Bit DATE	4 4 4 1 3
Recommendation	RecommendationID JobPostID JobSeekerID Score	Int Int Int Int	4 4 4 4

Skills	SkillId SkillName	Int Varchar	4 20
JobSeekerSkills	JobSeekerSkillsId UserId SkillId	Int Int	4 4
JobRequiredSkills	JobRequiredSkillsId JobPostId SkillId	Int Int	4 4
Messages	MessagesId ConversationId JobSeekerId CompanyId Message CreatedAt	Int Int Int Int Varchar Date	4 4 4 4 100 20
Hires	HiresId JobSeekerId CompanyId JobPostId HireDate OfferId	Int Int Int Int Varchar Int	4 4 4 4 30 4

7. Class Based Modeling

7.1 CLASS BASED MODELING CONCEPT

Class-based modeling represents the objects that the system will manipulate, the operations that will be applied to the objects, the relationships between the objects and the collaborations that occur between the classes that are defined.

7.2 Noun List

SL	Noun
1	User
2	Platform
3	Name
4	Email
5	Password
6	Character
7	Login
8	System
9	Credential
10	Dashboard
11	OTP
12	Criteria
13	Job Post
14	Company
15	Job Seeker
16	Job title
17	Job description
18	Job Required skills
19	Experience
20	Salary
21	Location
22	Input
23	Field

24	List
25	Job details
26	Filter
27	Skills
28	Job Application
29	Resume
30	Application details
31	Applicant info
32	Status
33	Applicant
34	Recommendation
35	Profile
36	Preference
37	Chat
38	Job Role
39	Process
40	Opening
41	Message
42	Chat History
43	File
44	Offer letter
45	Details
46	Conversation
47	Job seeker’s skills

7.3 List of Verbs

SL	Verb
1	Sign up
2	Access
3	Provide
4	Exist
5	Inform
6	Register
7	Prompt
8	Login
9	Validate
10	Redirect
11	Choose
12	Reset
13	Send
14	Ensure
15	Meet
16	Require
17	Post
18	Fill
19	View
20	Edit
21	Use

22	Apply
23	Upload
24	Submit
25	Check
26	Withdraw
27	Need
28	Display
29	Filter
30	Update
31	Status
32	Recommend
33	Match
34	Change
35	Refresh
36	Show
37	Initiate
38	Hire
39	Exchange
40	Manage
41	Share
42	Track
43	Download

7.4 General Classification

Candidate classes are categorized based on the seven general classifications. The analysis classes manifest themselves in one of the following ways:

1. External entities
2. Things
3. Events
4. Roles
5. Organizational units
6. Places
7. Structures

A candidate class is selected for special classification if it fulfills three or more Characteristics.

SL	Solution Space Noun	General Classification
1	User	4
2	Name	2
3	Email	2
4	Password	2
5	Login	3
6	Credential	2
7	OTP	2
8	Job Post	2
9	Company	5,7
10	Job Seeker	4
11	Job title	2
12	Job description	2
13	Job Required skills	2

14	Experience	2
15	Salary	2
16	Location	6
17	Job details	2,7
18	Skills	2
19	Job Application	2
20	Resume	2
21	Application details	7
22	Status	2
23	Recommendation	3
24	Profile	2
25	Chat	3
26	Message	2,3
27	Chat History	7
28	File	2
29	Offer letter	2
30	Conversation	3
31	Job seeker’s skills	2

7.5 Selection Criteria

The candidate classes are then selected as classes by six Selection Criteria. A candidate class generally becomes a class when it fulfills around three characteristics.

1. Retain information
2. Needed services
3. Multiple attributes
4. Common attributes
5. Common operations

6. Essential requirements

7.6 Potential general classified nouns to become a class

SL	Solution Space Noun	Selection Criteria
1	User	1,3,4
2	Job Seeker	1-5
3	Company	1-5
4	Job Post	1-5
5	Job Application	1-5
6	Recommendation	1,2,3,6
7	Message	1,2,3
8	Conversation	1,4,5
9	OfferLetter	1,3,6

7.7 Analysis

The following classes are deemed essential for the system due to their fundamental roles in ensuring the overall functionality and efficiency of the platform:
1. Database
2. Models
3. Authentication
4. Job Post Service
5. Job Application Service

6. OfferLetterHandler
7. HiringRecord

7.8 Attribute and Method identification

Class	Attribute	Methods
Database	-connection	+connectDB()

Models	-Database	+create()
		+update()
		+delete()
		+getProfile()

User	-userId
	-userName
	-email
	-phoneNumber
	-password(hashed)
	-accountType
		+changePassword()
		+verifyCredentials()

Authentication	-Database
	-User
		+ login()
		+ logout()
		+ register()
		+sendResetMail()
		+ generateToken()
		+ resetPassword()
		+sendOTP()

Job Seeker	-skills[]
	-location
	-salaryExpectation
	-experience
	-expectedPosition
	-Applications[]
	-Recommendations[]
		+saveAppliedJobs()
		+saveRecommendation()

Company	-industry
	-location
	-requiredSkills[]
		+addPost()
		+removePost()

	-jobPositions[] -JobPosts[]	

Job Post	-jobPostId
	-companyId
	-title
	-description
	-requiredSkills[]
	-location
	-salary
	-experienceRequired
	-postDate
		+viewJobPost()
		+updatePost()

Job Post Service	-JobPost[]
	-companyId
		+searchPost()
		+filterPost()
		+getPost()
		+createPost()
		+getPostById()

Job Application	-applicationId
	-jobPost
	-jobSeeker
	-resume
	-applicationStatus
	-submissionDate
		+updateStatus(status)
		+viewStatus()

JobApplicationService	-JobApplications[]	+addApplication()
		+removeApplication()

Recommendation	-recommendationId
	-JobseekerId
	-recommendedJobs[]
		+generateRecommendations(JobSeeker)
		+filterRecommendations(criteria)
		+getRecommendation

Message	-messageID	+sendMessage()

	-senderID -receiverID -Content -createdAt -ConversationId	+getMessage()

Conversation	-conversationID
	-Participants
	-createdAt
	-OwnedBy
	-Messages[]
		+addMessages()
		+getParticipants()
		+getConversation()
		+deleteConversation()

OfferLetter	-offerLetterId
	-companyId
	-jobSeekerId
	-JobPostId
	-OfferDate
	-OfferStatus
	-letterPath
		+acceptOffer()
		+rejectOffer()

OfferLetterHandler		+generateOfferLetter()
		+sendOfferLetter()
		+getOfferLetter()

HiringRecord	-hiringId
	-companyId
	-JobSeekerId
	-JobPostId
	-hireDate
	-salaryOffered
	-OfferLetterId
		+addHiredApplicant()
		+getHiredApplicants()
		+removeApplicant()

7.9 Class Card

Database	
Attribute	Method

Database	
-connection	+connectDB()
Responsibilities	Collaborator
Connect to Database Maintain Database access and retrieval	User Job Seeker Company Authentication Job Post Application Recommendation Message HiringRecord

Models	
Attribute	Method

-Database	+create()
	+update()
	+delete()
	+receiveOTP()
	+getProfile()

Responsibilities	Collaborator
Manages database interactions for creating, updating, and deleting records.	Database User JobSeeker Company

User	
Attribute	Method

User	

-userId
-userName
-email
-phoneNumber
-password(hashed)
-accountType
	+changePassword()
	+verifyCredentials()

Responsibilities	Collaborator
Store and manage user’s personal details Maintain user profile updates and credential verification	Authentication Database Models

Authentication	
Attribute	Method

-User
-Database
	+ login()
	+ logout()
	+ register()
	+sendResetMail()
	+ generateToken()
	+ resetPassword()
	+sendOTP()

Responsibilities	Collaborator
Handles user authentication and password management.	Database User

Job Seeker	
Attribute	Method

-skills[]
-location
-salaryExpectation
	+saveApplication()
	+saveRecommendation()

Job Seeker	
-experience -expectedPosition -Applications[] -Recommendations[]	
Responsibilities	Collaborator
Represents job seekers, allows updating profiles, saving applied jobs, and recommendations.	User Recommendation JobPost Models

Company	
Attribute	Method

-industry
-location
-requiredSkills[]
-jobPositions[]
-JobPosts[]
	+addPost()
	+removePost()

Responsibilities	Collaborator
Represents companies, allows updating company profiles and creating job Posts.	User Job Post Models

Job Post	
Attribute	Method
-jobPostId	+viewJobPost()

Job Post	
-companyId -title -description -requiredSkills[] -location -salary -experienceRequired -postDate	+updatePost()
Responsibilities	Collaborator
Represents individual job posts with required details and allows CRUD operations on posts.	Company JobPostService Models

Job Post Service	
Attribute	Method

-JobPosts[]	+ searchPost()
	+ filterPost()
	+ getPost()
	+ createPost()
	+ getPostById()

Responsibilities	Collaborator
Manages job post search, filter, and CRUD operations on Job post for companies	Company JobPost

Job Application	
Attribute	Method
-applicationId	+updateStatus(status)

Job Application	
-jobPost -jobSeeker -resume -applicationStatus -submissionDate	+viewStatus()
Responsibilities	Collaborator
Represents job applications and handles their submission, status updates, and view functionalities.	Job Seeker ApplicationService Job Post

Job Application Service	
Attribute	Method

-JobApplications[]	+addApplication()
	+removeApplication()

Responsibilities	Collaborator
Manages job applications received by companies and handles updating and retrieving applications	Job Application Company

Recommendation	
Attribute	Method

-recommendationId
-recommendedJobs[]
	+generateRecommendations(JobSeeker)
	+filterRecommendations(criteria)
	+getRecommendation()

Responsibilities	Collaborator
Generates and manages job recommendations for job seekers based on criteria.	Job Post Job Seeker

Message	
Attribute	Method

-messageId
-senderId
-receiverId
-content
-createdAt
-conversationId
	+sendMessage()
	+getMessage()

Responsibilities	Collaborator
Represents messages sent between users. Handles message sending and retrieval functionalities.	Conversation Job Seeker Company

Conversation	
Attribute	Method

- conversationId
- Participants[]
- createdAt
-ownedBy
-messages[]
	+ addMessage()
	+ getParticipants()
	+getConversation()
	+ deleteConversation()

Responsibilities	Collaborator
Represents a conversation between multiple users and manages new conversations, retrieving participants, and deleting the conversation.	Message Job Seeker Company

OfferLetter	
Attribute	Method

- offerLetterId
- companyId
-jobSeekerId
- jobPostId
-offerDate
-offerStatus
- letterPath
	+acceptOffer()
	+rejectOffer()

Responsibilities	Collaborator
Represents an offer letter and handles acceptance and rejection.	Job Post Job Seeker Company

OfferLetterHandler	
Attribute	Method

-OfferLetter	+generateOfferLetter()
	+sendOfferLetter()
	+getOfferLetter()

Responsibilities	Collaborator
Manages sending and receiving of Offer Letter and generates the Offer letter	OfferLetter

HiringRecord	
Attribute	Method

- hiringId
-company
-jobSeeker
	+ addHiredApplicant()
	+addCompany()
	+removeCompany()

HiringRecord	

-jobPost
-hireDate
-salaryOffered
-offerLetter
	+ getHiredApplicants()
	+ removeApplicant()

Responsibilities	Collaborator
Records hiring details and allows for adding, retrieving, and removing hiring records.	OfferLetter Company JobSeeker JobPost

7.10 CRC Diagram

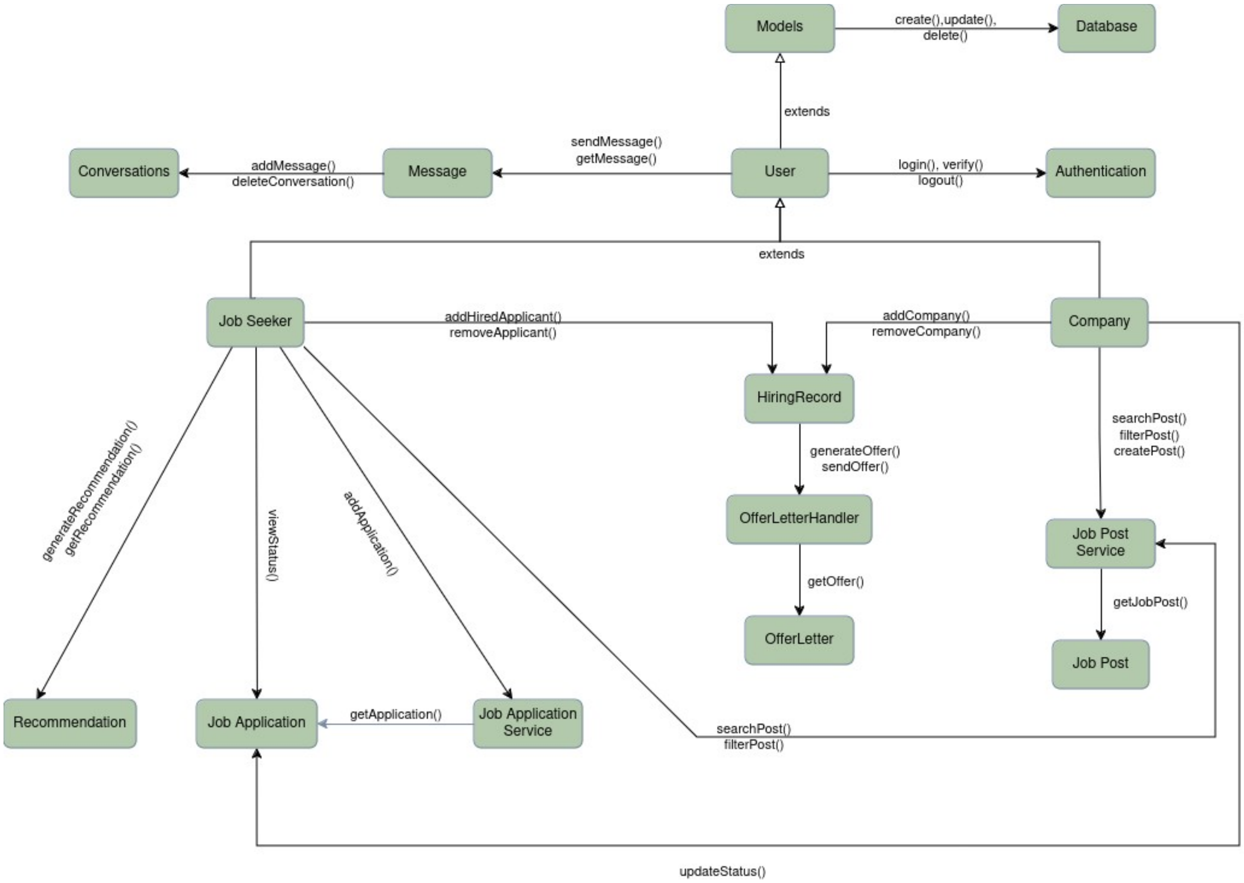


Fig25: Class-Responsibility-Collaborator Diagram

8. Behavioral Modeling

The behavioral model indicates how software will respond to external events or stimuli. In the context of behavioral modeling, two different characterizations of states must be considered:

(1) the state of each class as the system performs its function and

(2) the state of the system as observed from the outside as the system performs its function.

8.1 State Transition Diagram

One component of a behavioral model is a UML state diagram that represents active states for each class and the events (triggers) that cause changes between these active states.

8.1.1 Event Table

SL	Event	Event name	Initiator	Collaborator	Associated methods

1	User Registration	registerUser	User	Authentication, Database	register()
					hashPassword()

2	User Login	userLogin	User	Authentication, Database	login()
					generateToken()

3	Authenticate User	authenticate User	User	Authentication, Database	verifyCredentials()
4	Password Reset Request	reqPassword Reset	User	Authentication	sendResetMail()
5	Password Reset	resetPassword	User	Authentication, Database	resetPassword()
6	SendOTP	send_otp	Authentication	User	sendOTP()
7	ReceiveOTP	receive_otp	User	Authentication, Models	receiveOTP()
8	Update Password	updatePassword	User	Authentication,	update()

				Database,Models	
9	Logout	userLogout	User	Authentication	logout()
10	Create account	createAccount	User	Models	create()
11	View Profile	viewProfile	User	Models, Database	getProfile()

12	Profile Update	updateUserProfile	User	Database, Models	update()
					getProfile()

13	Delete Account	deleteAccount	User	Database, Models	delete()
14	Save Jobs	saveApplied Jobs	Job Seeker	Job Post	saveApplied Jobs()
15	Job Post Creation	createJobPost	Job Post service	Company, Job Post	createPost()
16	Add Job Post	addJobPost	Company	Database	addPost()
17	Job Post Update	updateJobPost	Company	Job Post, Database	updatePost()
18	Job Post Deletion	deleteJobPost	Company	Job Post, Job Post Service	removePost()

19	Search Job Posts	searchJobs	Job Seeker	Database, Job Post, Job Post Service	searchPost()
					getPost()

20	View Job Posts	viewJobPost	Job Post	User	viewJobPost()
21	GetJobPosts	getPosts	Job Post Service	Job Post, User	getPost()
22	Application	reviewAppli	Company	Job	updateStatus

	Status Change	cation		Application	()
23	Application Status Check	checkApplicationStatus	Job Seeker	Job Application	viewStatus()
24	Add Application	addApplication	Job Seeker	Job Application Service	addApplication()
25	Remove Application	removeApplication	Job Seeker	Job Application Service	removeApplication()

26	Job Recommendation	getRecommendations	Recommendation	Database	generateRecommendations()
					filterRecommendations()

27	Get Recommendation	getRecommendation	Job Seeker	Recommendation	getRecommendation()
28	View Job Application Status	viewApplicationStatus	Job Seeker	Job Application	viewStatus()
29	Conversation initiating	startConversation	Company	Message, Database	sendMessage()
30	Message Sending	sendMessage	User	Message, Conversation	sendMessage(), addMessage()
31	Get Conversation	getConversation	User	Conversation	getConversation()
32	View conversation list	viewParcipants()	Company	Conversation	getParticipants()

33	Delete Conversation	deleteConversation	Company	Conversation, Database	deleteConversation()
34	Offer letter Generation	generateOfferLetter	Company	Offer Letter Handler, Database	generateOfferLetter()
35	Sending Offer Letter	sendOfferLetter	Company	Offer Letter, Offer Letter Handler	sendOfferLetter()
36	Offer acceptance	acceptOffer	Job Seeker	Offer Letter, Company	acceptOffer()
37	Offer Rejection	rejectOffer	Job Seeker	Offer Letter, Company	rejectOffer()
38	Hiring Completion	completeHiring	Company	Hiring Record, Database	addHiredApplicant()
39	View Hired Applicant List	getHiredApplicants()	Company	Hiring Record, Database	getHiredApplicants()
40	Remove Applicant	removeApplicant	Company	Hiring Record, Database	removeApplicant()

8.1.2 State Transition Diagram

Diagram ID: 1

Name: User

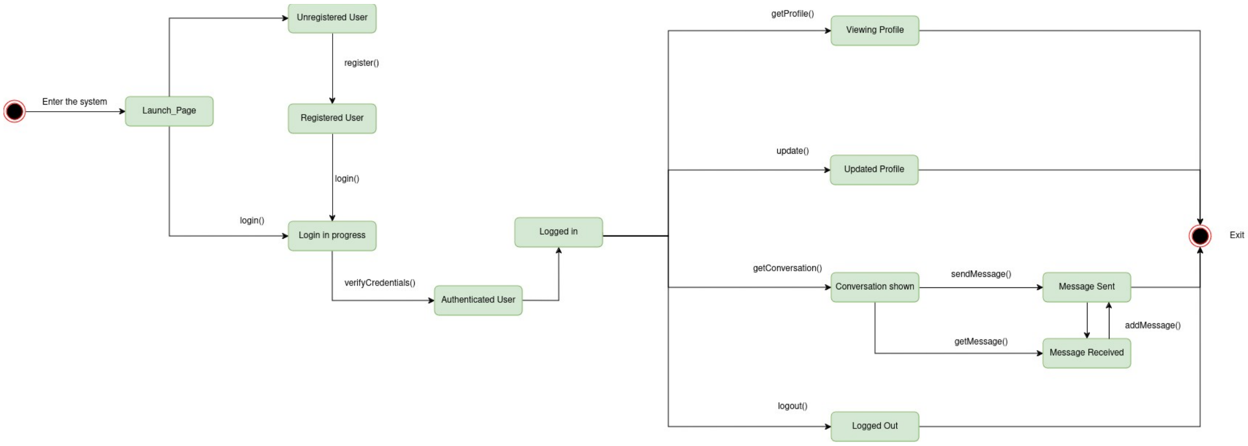


Fig26: State Transition Diagram for User Class

Diagram ID: 2

Name: Authentication

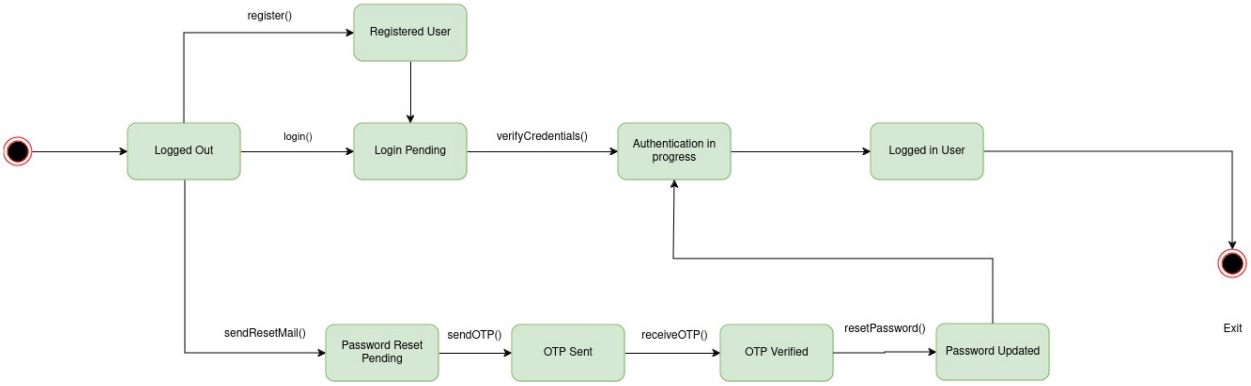


Fig27: State Transition Diagram for Authentication Class

Diagram ID: 3

Name: Models

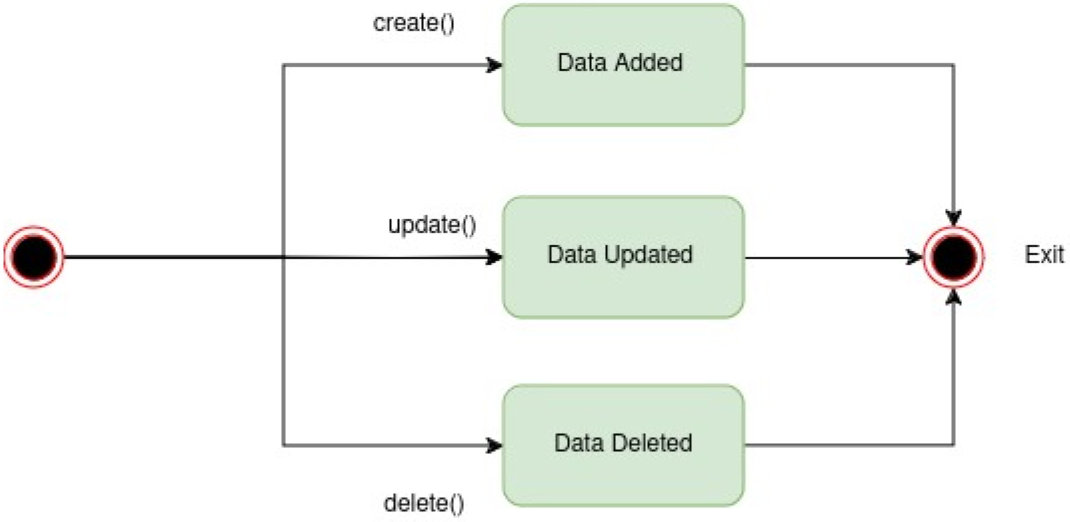


Fig28: State Transition Diagram for Models Class

Diagram ID: 4

Name: Database

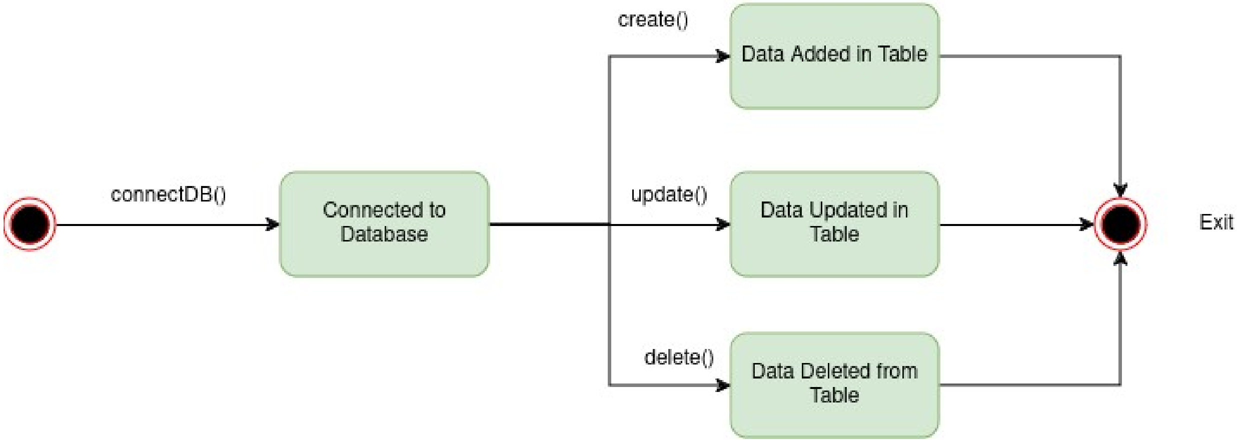


Fig29: State Transition Diagram for Database Class

Diagram ID: 5

Name: Job Seeker

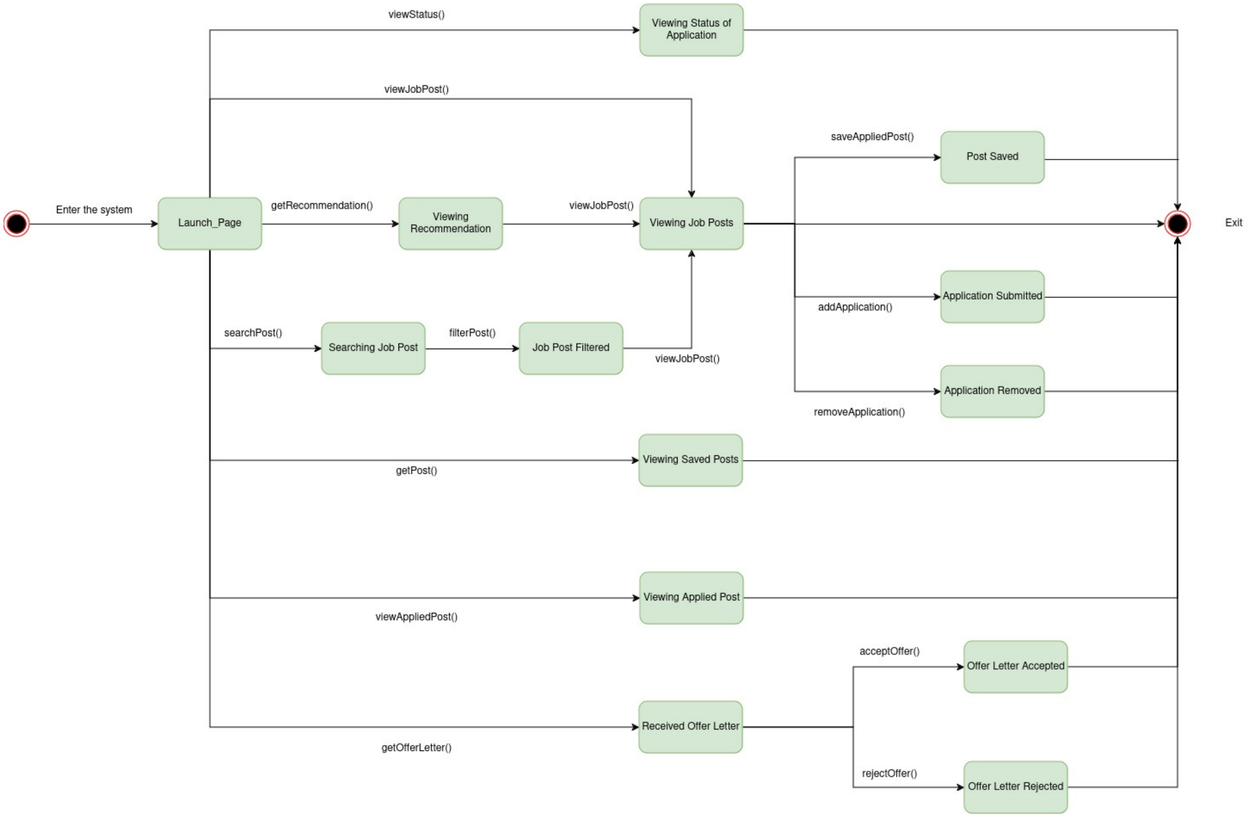


Fig30: State Transition Diagram for Job Seeker Class

Diagram ID: 6

Name: Company

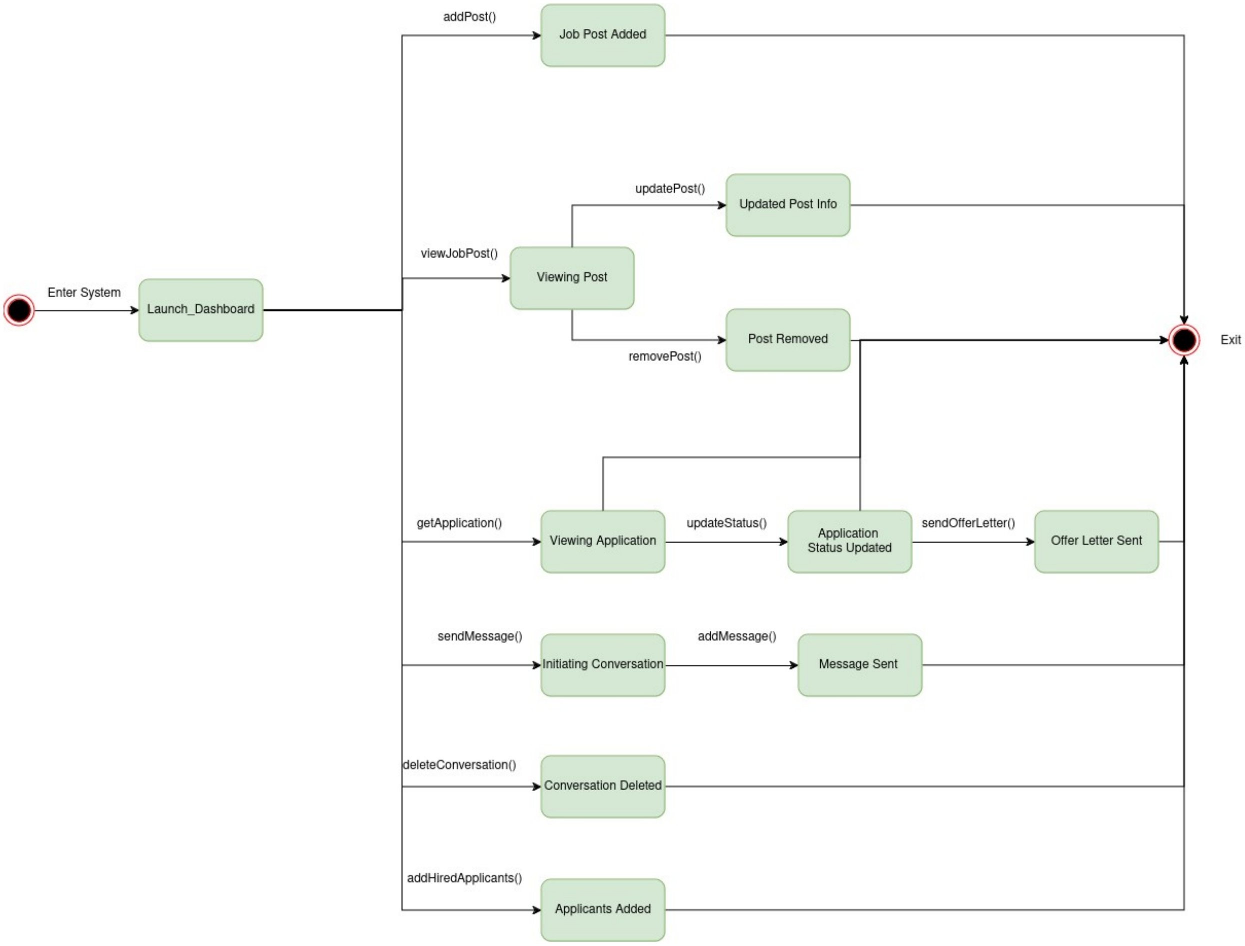


Fig31: State Transition Diagram for Company Class

Diagram ID: 7

Name: Job Application

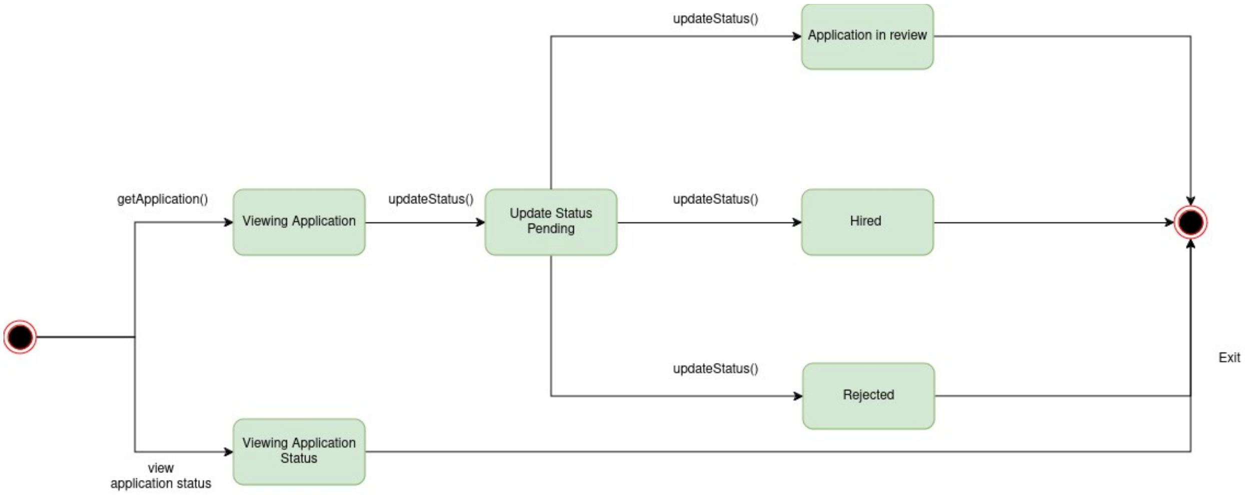


Fig32: State Transition Diagram for Job Application Class

Diagram ID: 8

Name: Job Post

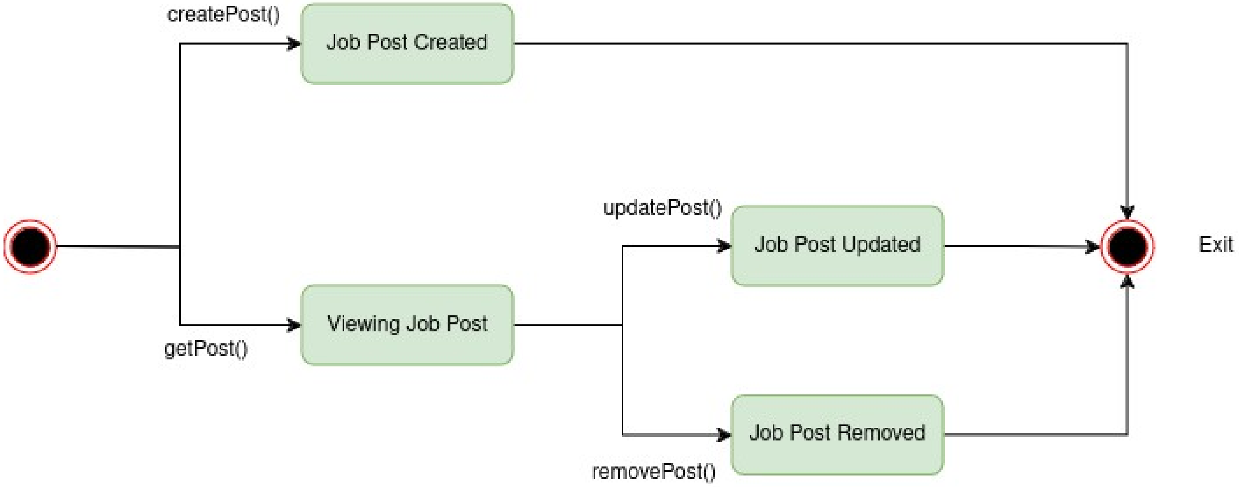


Fig33: State Transition Diagram for Job Post Class

Diagram ID: 9

Name: Recommendation

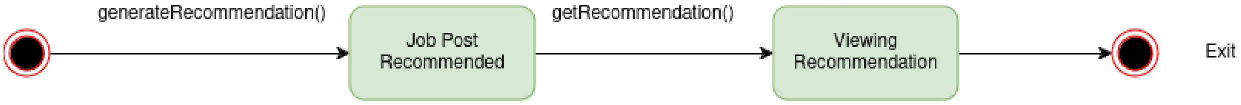


Fig34: State Transition Diagram for Recommendation Class

Diagram ID: 10

Name: Offer

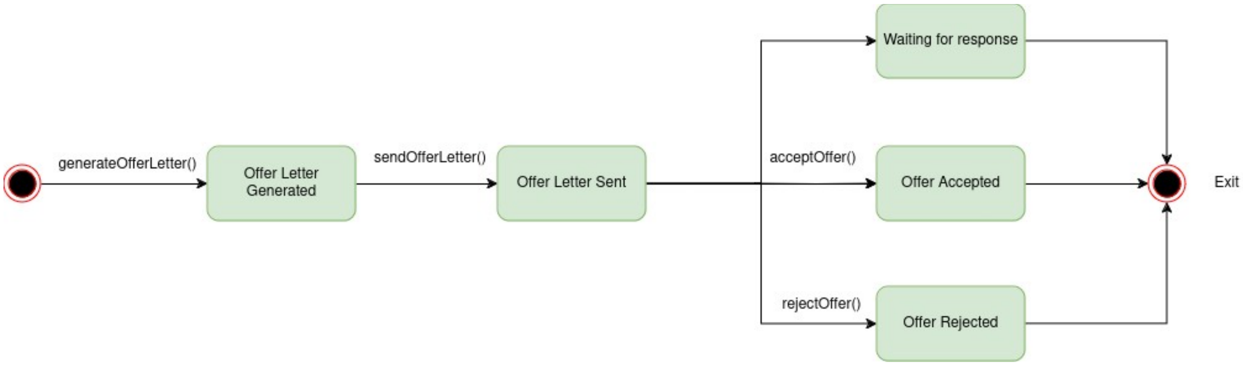


Fig35: State Transition Diagram for Offer Class

Diagram ID: 11

Name: Message

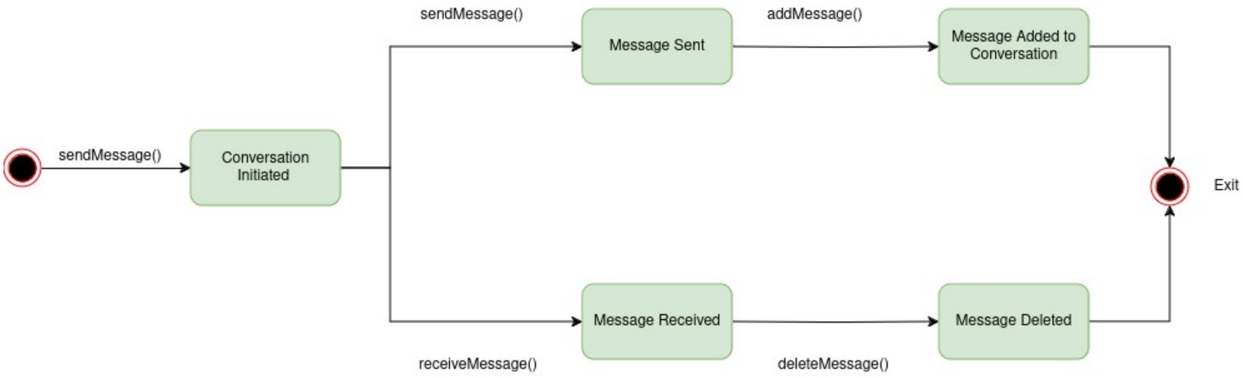


Fig36: State Transition Diagram for Message Class

Diagram ID: 12

Name: Conversation

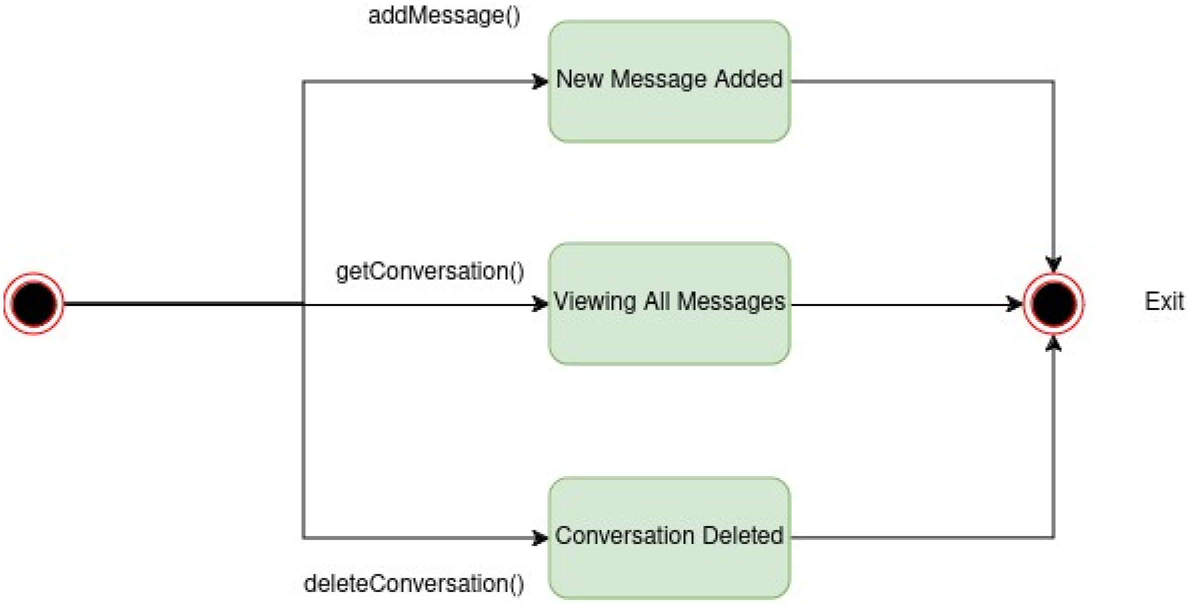


Fig37: State Transition Diagram for Conversation Class

Diagram ID: 13

Name: HiringRecord

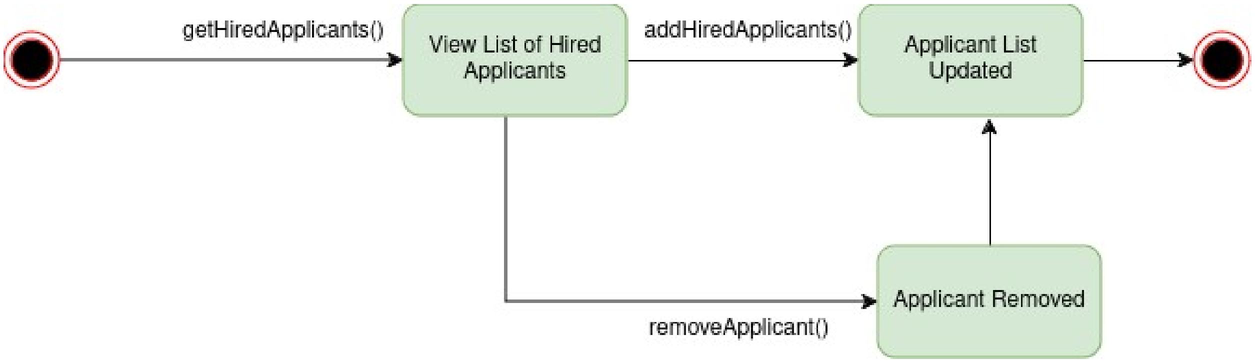


Fig38: State Transition Diagram for HiringRecord Class

Diagram ID: 14

Name: Job Post Service

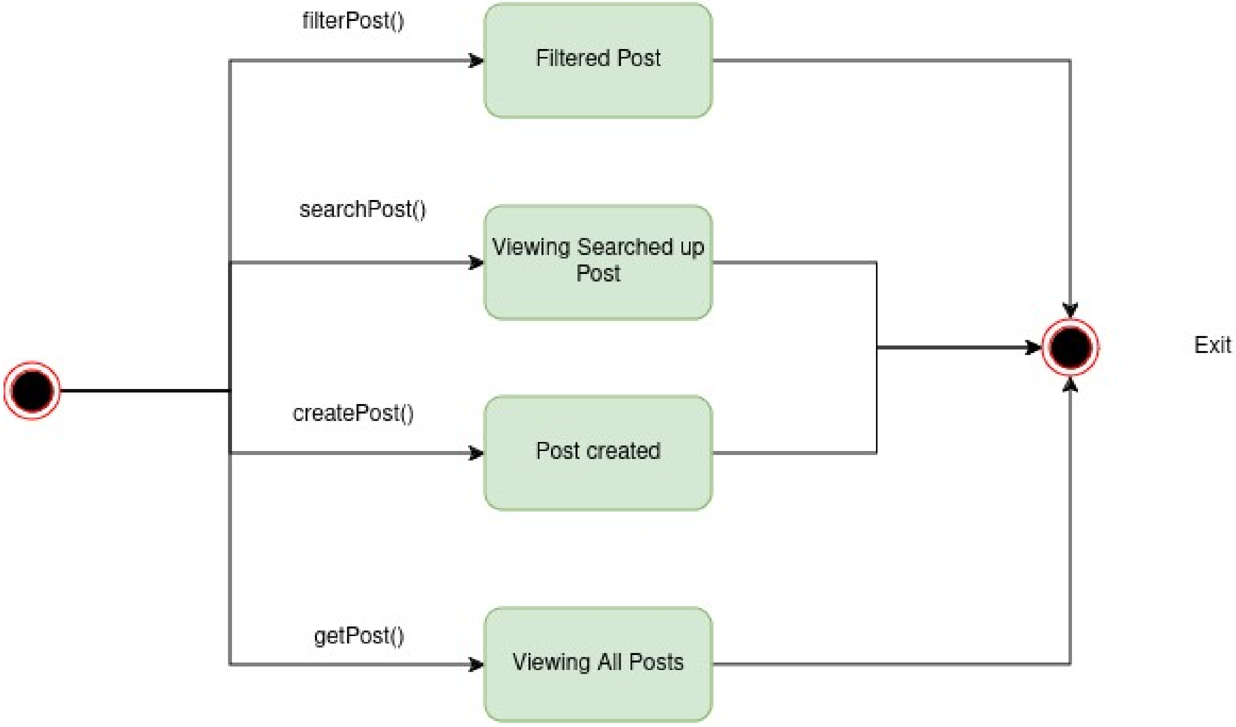


Fig39: State Transition Diagram for Job Post Service Class

Diagram ID: 15

Name: Job Application Service

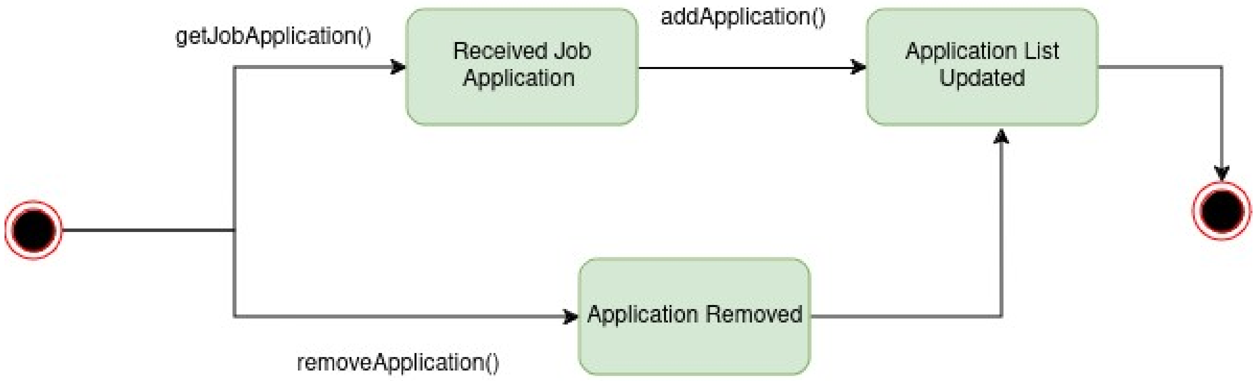
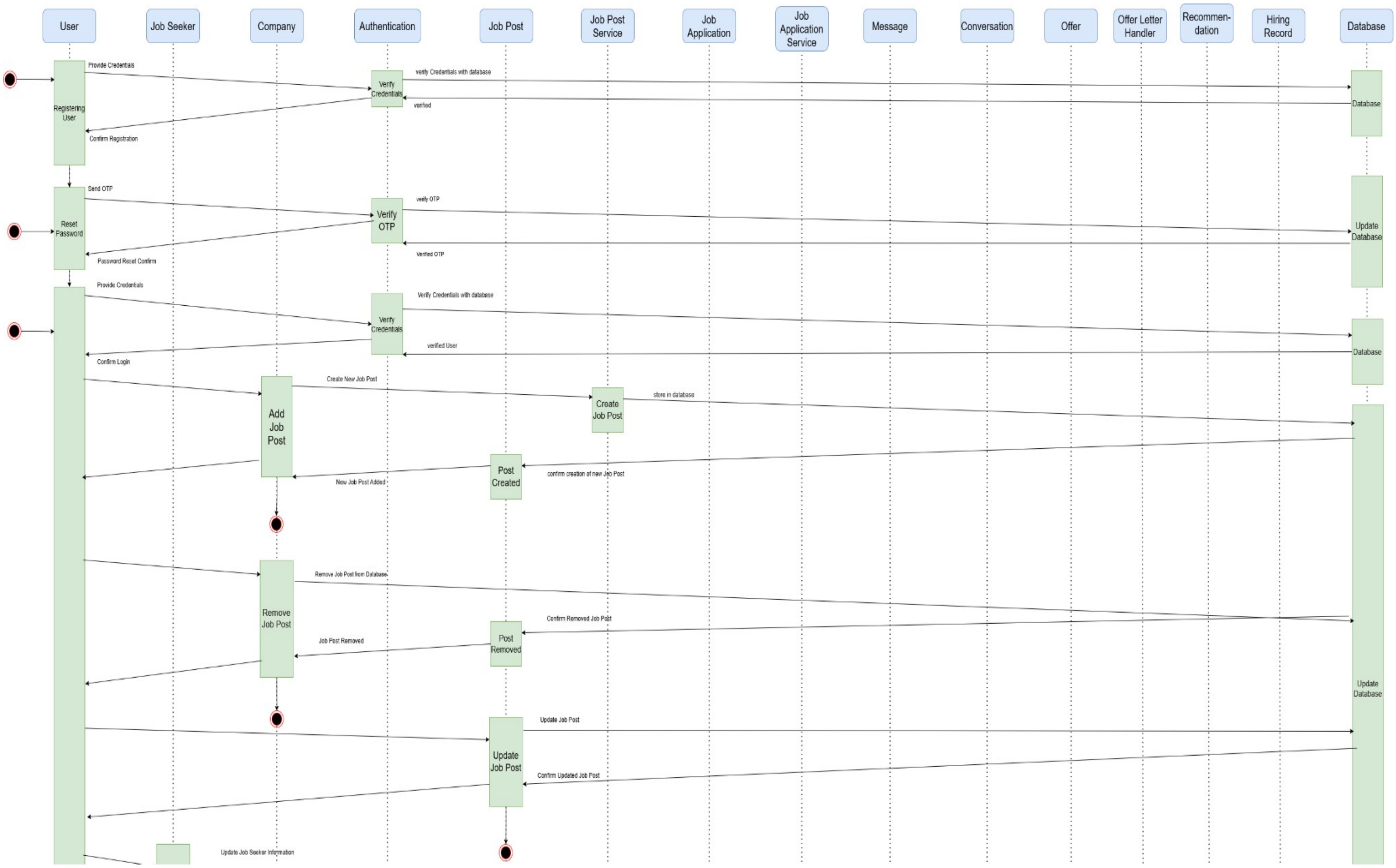
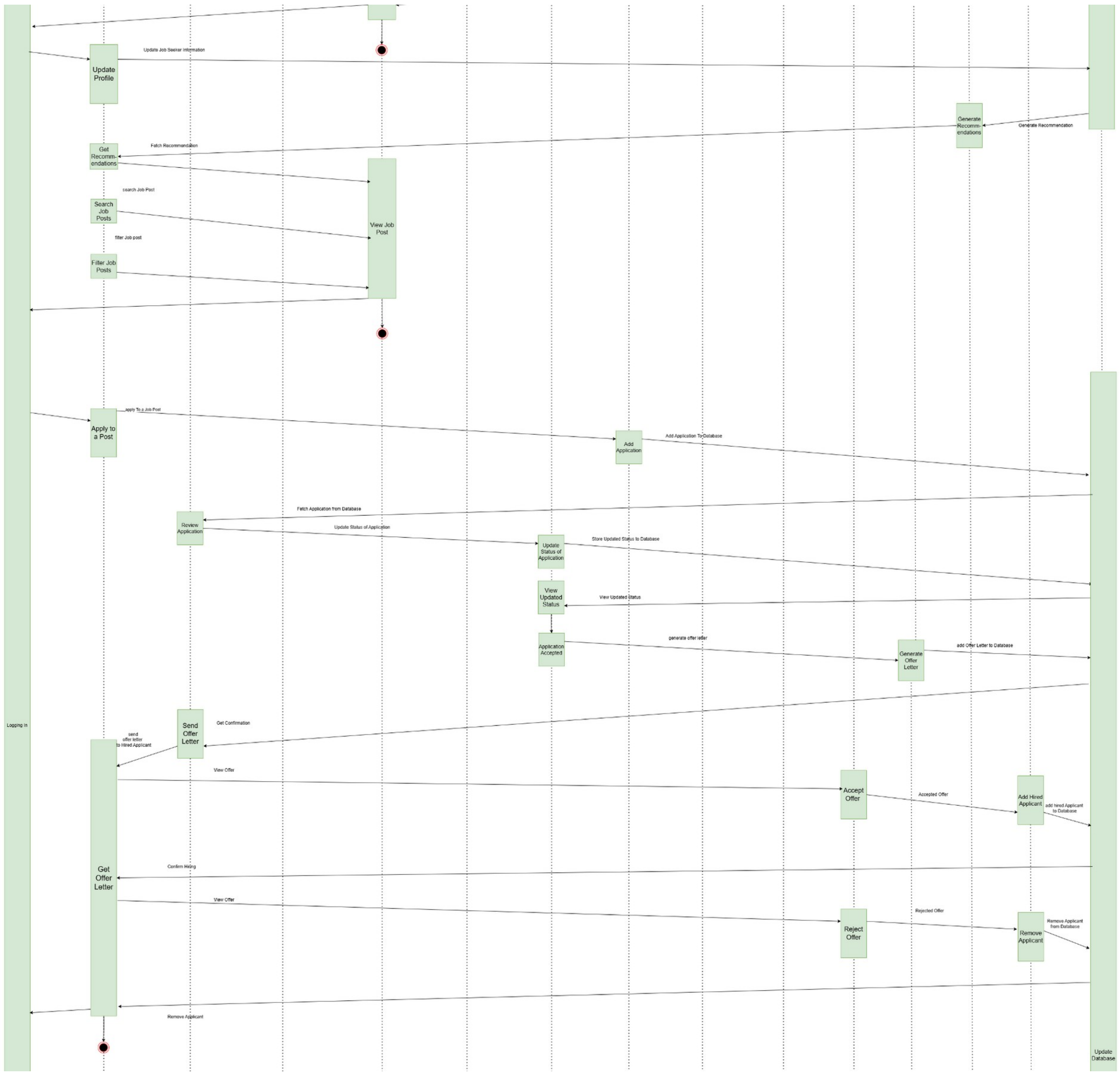


Fig40: State Transition Diagram for Job Application Service Class

8.2 Sequence Diagram


User
Job Seeker
Company
Authentication
Job Post
Job Post Service
Job Application
Job Application Service
Message
Conversation
Offer
Offer Letter Handler
Recommen-dation
Hiring Record
Database
Provide Credentials
Registering User
Verify Credentials
verify Credentials with database
Database
verified
Confirm Registration
Send OTP
Reset Password
Verify OTP
verify OTP
Update Database
Verified OTP
Password Reset Confirm
Provide Credentials
Verify Credentials
Verify Credentials with database
Database
verified User
Confirm Login
Create New Job Post
Add Job Post
Create Job Post
store in database
Post Created
confirm creation of new Job Post
New Job Post Added
Remove Job Post from Database
Remove Job Post
Confirm Removed Job Post
Post Removed
Job Post Removed
Update Database
Update Job Post
Update Job Post
Confirm Updated Job Post
Update Job Seeker Information


Update Job Seeker Information
Update Profile
Generate Recommendation
Generate Recomm-endations
Fetch Recommendation
Get Recomm-endations
search Job Post
Search Job Posts
View Job Post
filter Job post
Filter Job Posts
apply To a Job Post
Apply to a Post
Add Application To Database
Add Application
Fetch Application from Database
Review Application
Update Status of Application
Update Status of Application
Store Updated Status to Database
View Updated Status
View Updated Status
Application Accepted
generate offer letter
Generate Offer Letter
add Offer Letter to Database
Logging In
Get Confirmation
Send Offer Letter
send offer letter to Hired Applicant
View Offer
Accept Offer
Accepted Offer
Add Hired Applicant
add hired Applicant to Database
Confirm Hiring
Get Offer Letter
View Offer
Reject Offer
Rejected Offer
Remove Applicant
Remove Applicant from Database
Remove Applicant
Update Database

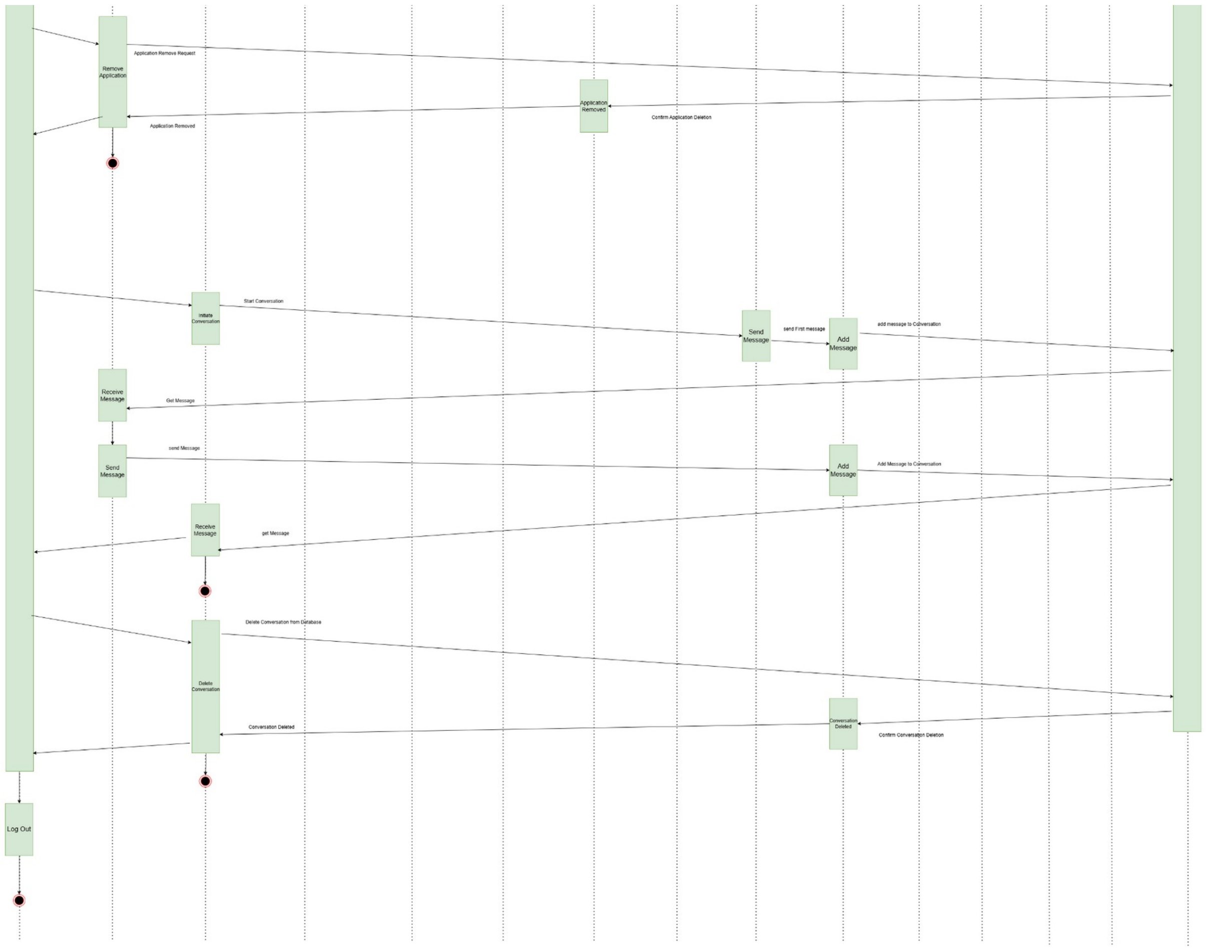


Fig41: Sequence Diagram for JobNode

9. Data Flow Modeling

9.1 Data Flow Diagram (DFD)

A data-flow diagram is a visual representation of how data moves through a system or a process. A data flow diagram (DFD) shows how information moves through any system or process. It displays data inputs, outputs, storage locations, and routes between each destination using predefined symbols such as rectangles, circles, and arrows as well as brief text labels.

Level 1: JobNode

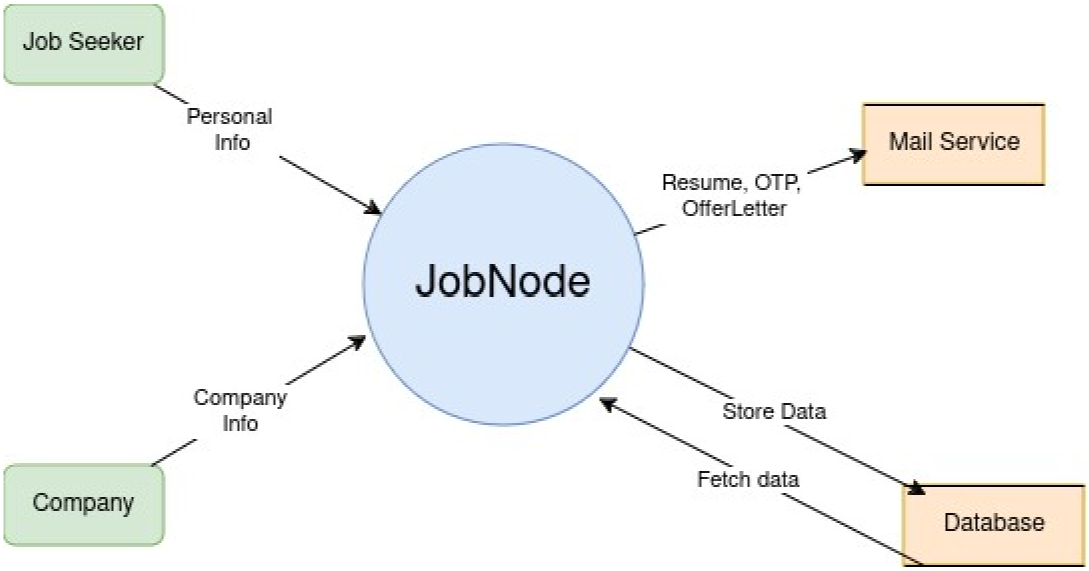


Fig42: Job Node: Search and Get Recommended(DataFlow Level-1)

Level 1: User Management

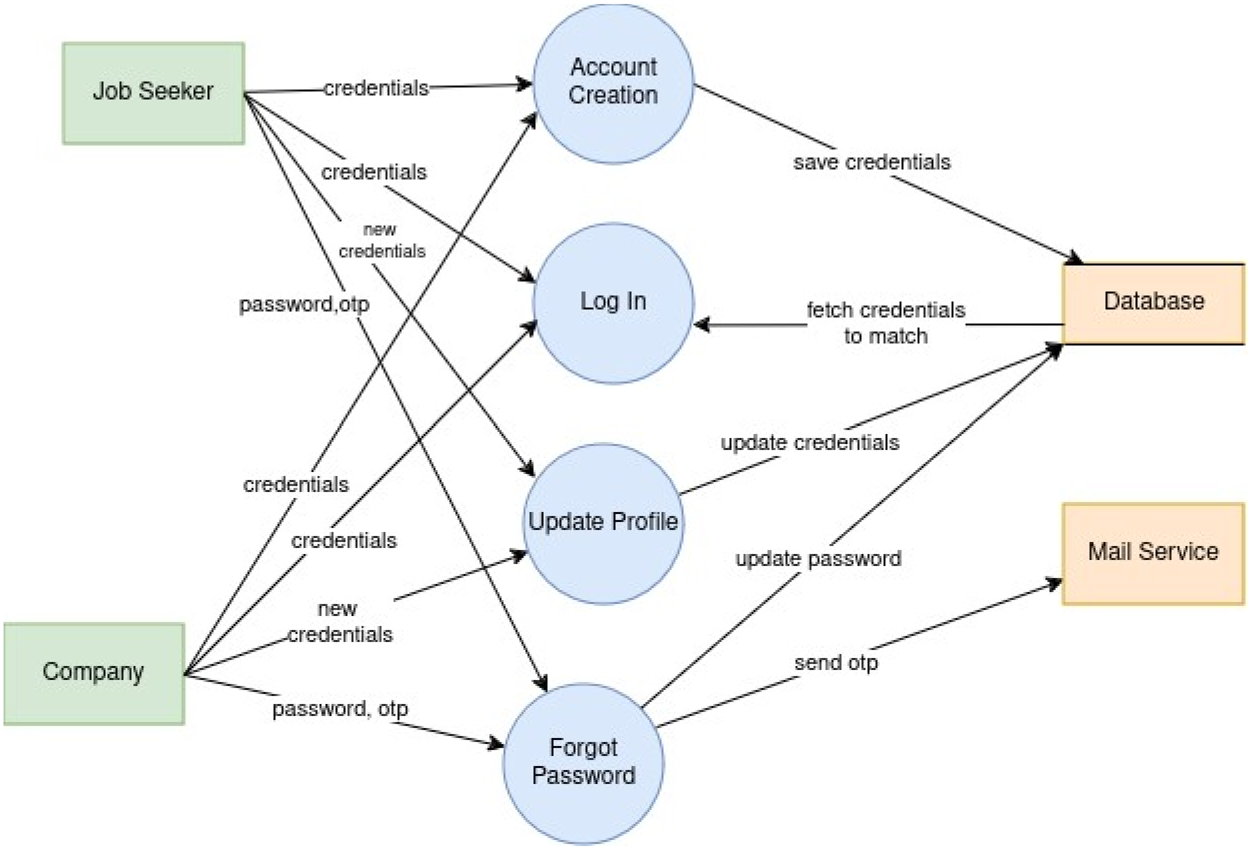


Fig43: User Management(DataFlow Level-1.1)

Level 1.2: Job Post Management

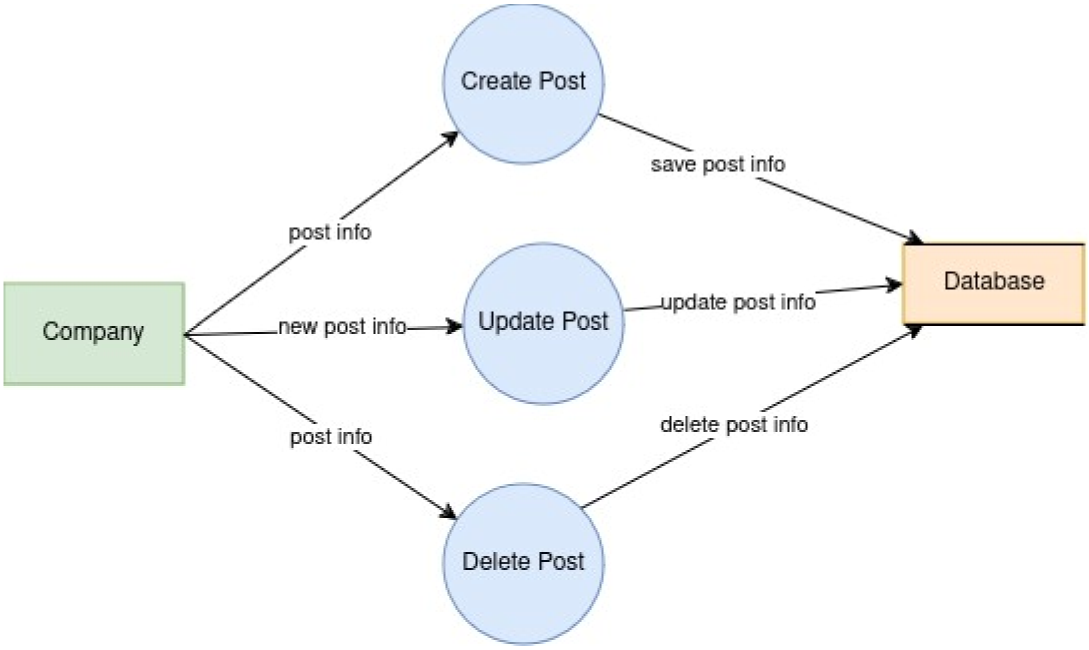


Fig44: Job Post Management(DataFlow Level-1.2)

Level 1.3: Recommendation Management

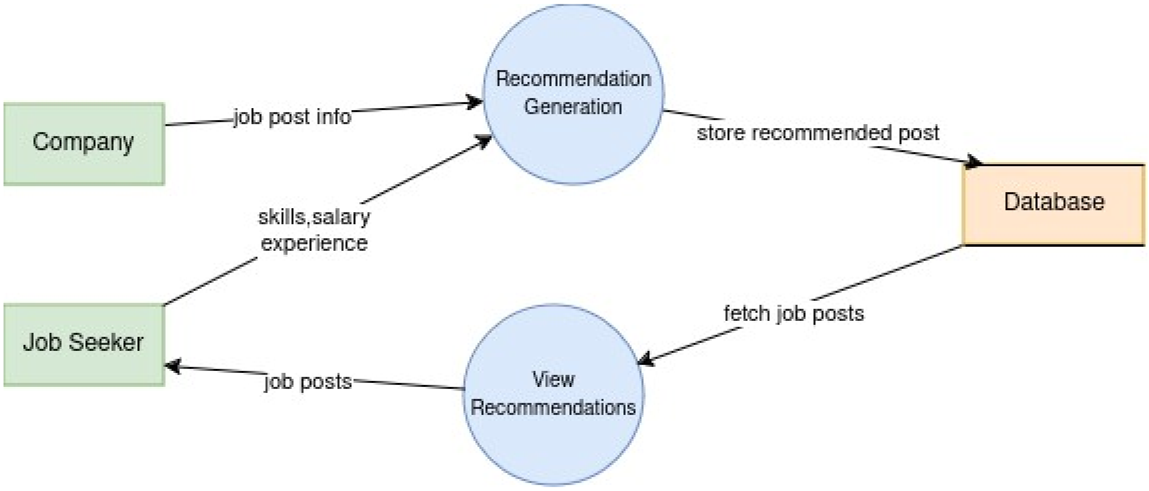


Fig45: Recommendation Management(DataFlow Level-1.3)

Level 1.4: Job Application Management

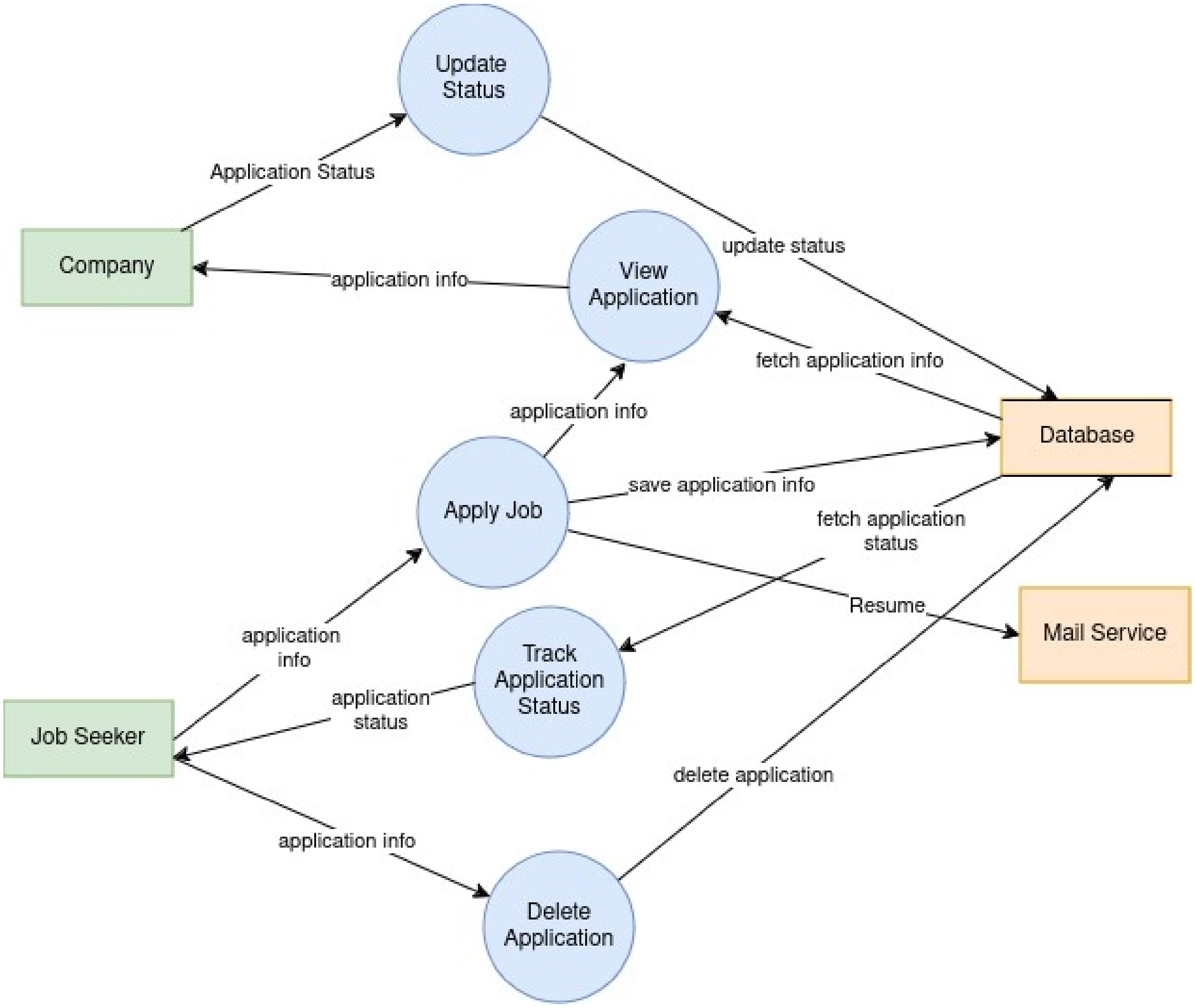


Fig46: Job Application Management(DataFlow Level-1.4)

Level 1. 5: Chat Management

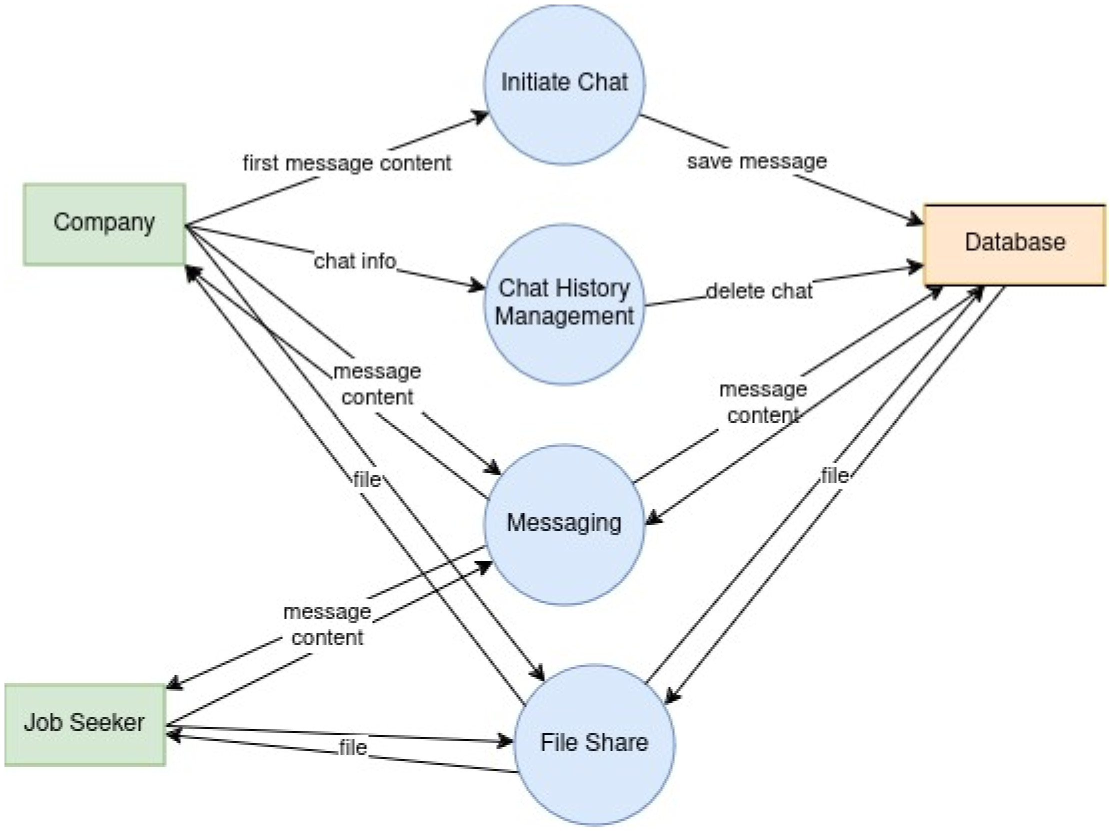


Fig47: Chat Management(DataFlow Level-1.5)

Level 1.6: Hiring Management

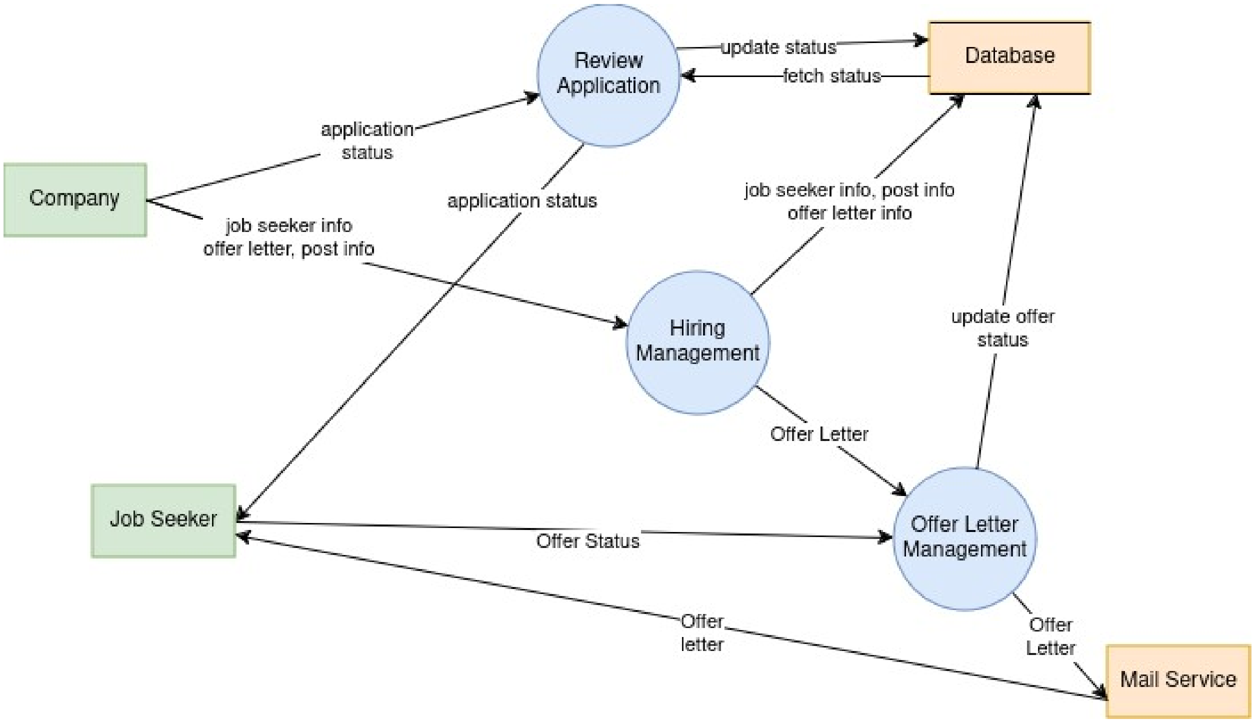


Fig48: Hiring Management(DataFlow Level-1.6)