MEF University
Department of Computer Engineering

COMP 302 Software Engineering

Report #1

Group Members

Zeynep Yılmaz 042101088

Onat Sarıbıyık 042101097

Ömer Faruk Tüccar 042001032

Sude Göktaş 042101127

Zehra Mert 042201058

Contents

Project Description

This project aims to develop a platform inspired by LinkedIn, focusing on core features such as user profiles, networking, job listings, and a feed of user-generated posts. The platform enables users to create accounts, build professional profiles, connect with others, share content, and apply for jobs. Additionally, an admin panel will allow system administrators to manage users and content.

The backend of the application will be developed using **Python** with **FastAPI** connection, and data will be stored in a **SQLite** database. The frontend will be built with **HTML** and **CSS** for simplicity and accessibility.

The system will consist of the following core modules:

- **User Account & Authentication**
  Users can register using their email and password and log into the platform.

- **Profile Management**
  Users can update their profiles, including experience, education, and a short bio.

- **Connections (Networking)**
  Users can send and accept connection requests to build their professional network.

- **News Feed & Posts**
  Users can share posts, like and comment on others' posts. The home page shows updates from connections.

- **Job Listings & Applications**
  Employers can post jobs, and users can apply to listings that match their interests and skills.

- **Search & Filtering**
  Users can search for people or posts and apply filters to job listings (e.g., by location, experience level, or job type).

- **Admin Panel**
  Admins can manage users and posts, delete, and perform moderation tasks.

1) Natural Language Scenario

Scenario 1 – Job Seeker (Ahmet)

Ahmet is a recent graduate looking for job opportunities. He registers on the platform and creates a user profile. He fills in his education background, experience, and bio. Then he searches for other users and sends them connection requests.

On the dashboard, he sees posts shared by his connections and engages by liking and commenting on them. He navigates to the job listings page, filters results to find "Software Developer Intern" roles in Istanbul, and submits a few applications.

Ahmet uses:

- User registration and login
- Profile editing
- Sending and accepting connection requests
- Viewing, liking, and commenting on posts
- Filtering and applying for job listings

Scenario 2 – Employer (Elif)

Elif is an HR manager at a tech company. After logging into the platform, she visits the "Post a Job" section and creates a job listing for a "Frontend Developer" position. She specifies details such as job type (full-time), remote work availability, and experience requirements. After publishing the job, she regularly checks applications and views applicants' profiles to select suitable candidates.

Elif uses:

- Job posting
- Application management
- Reviewing applicant profiles

Scenario 3 – Admin (Mert)

Mert is an administrator of the platform. He logs into the **Admin Panel** using his credentials. He notices a post that has been reported by multiple users. He accesses the post and the user who created it. After reviewing the content, he decides to remove the post and temporarily delete the user comment.

Mert uses:

- Admin login
- Moderation actions (delete post, user ,and ,delete comment)

Scenario 4 – Active Networking User (Ayşe)

Ayşe is a senior software engineer who actively uses the platform. She regularly updates her profile, shares posts, and interacts with others by liking comments on her posts or sends her a connection request. She also uses advanced search and filters to discover new professionals in her field.

Ayşe uses:

- Profile management
- Posting and engaging with content
- Receiving and responding to notifications
- Searching and filtering for people and content

2-) USE CASE DIAGRAM

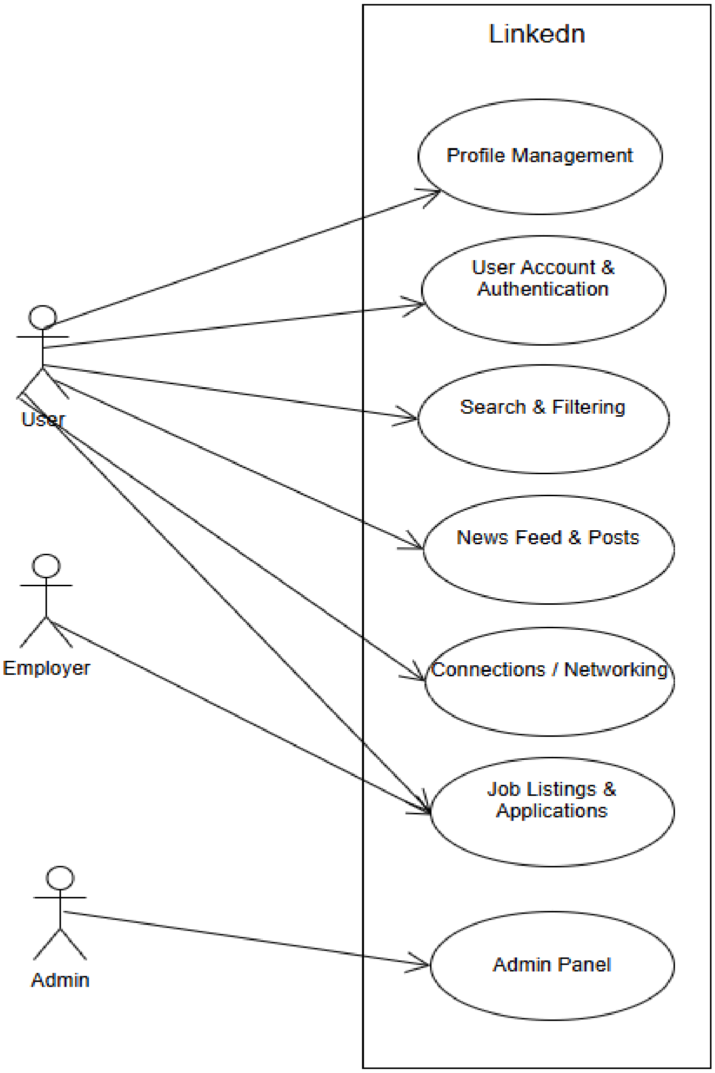


Figure 2.1 Linkedn Use Case Diagram

This use case diagram shows the main functions of the system.

3-) USE CASES

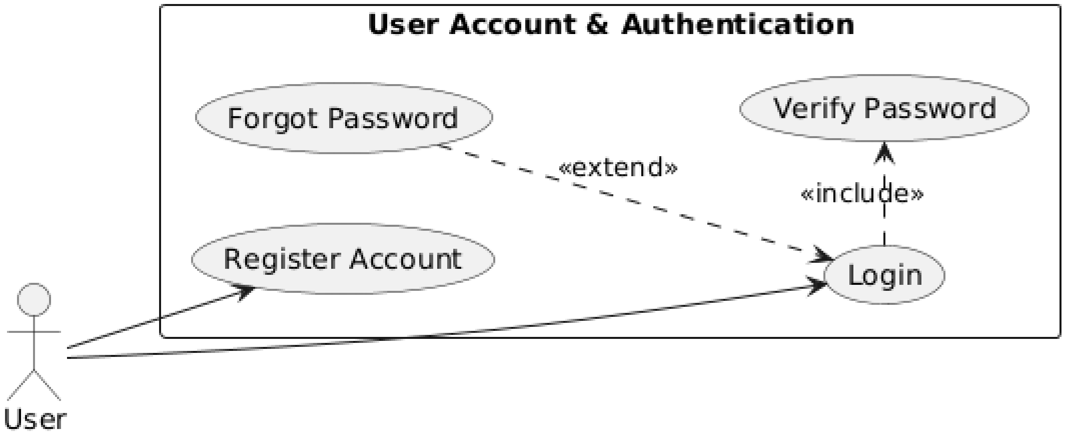


Figure 3.1 User Account & Authentication Use Case

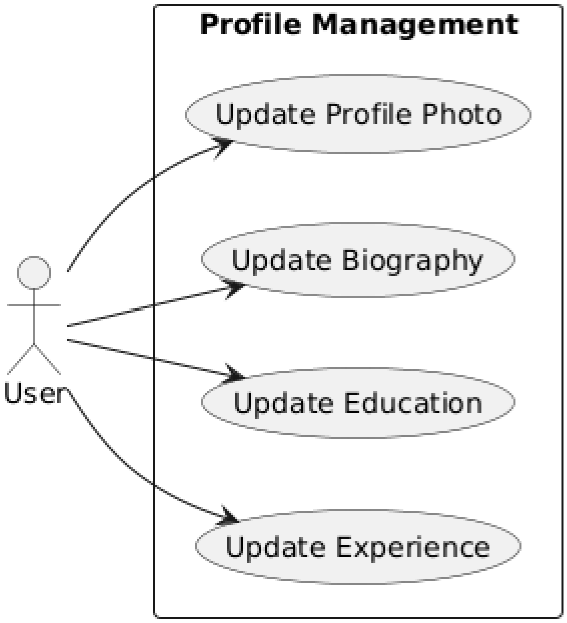


Figure 3.2 Profile Management Use Case

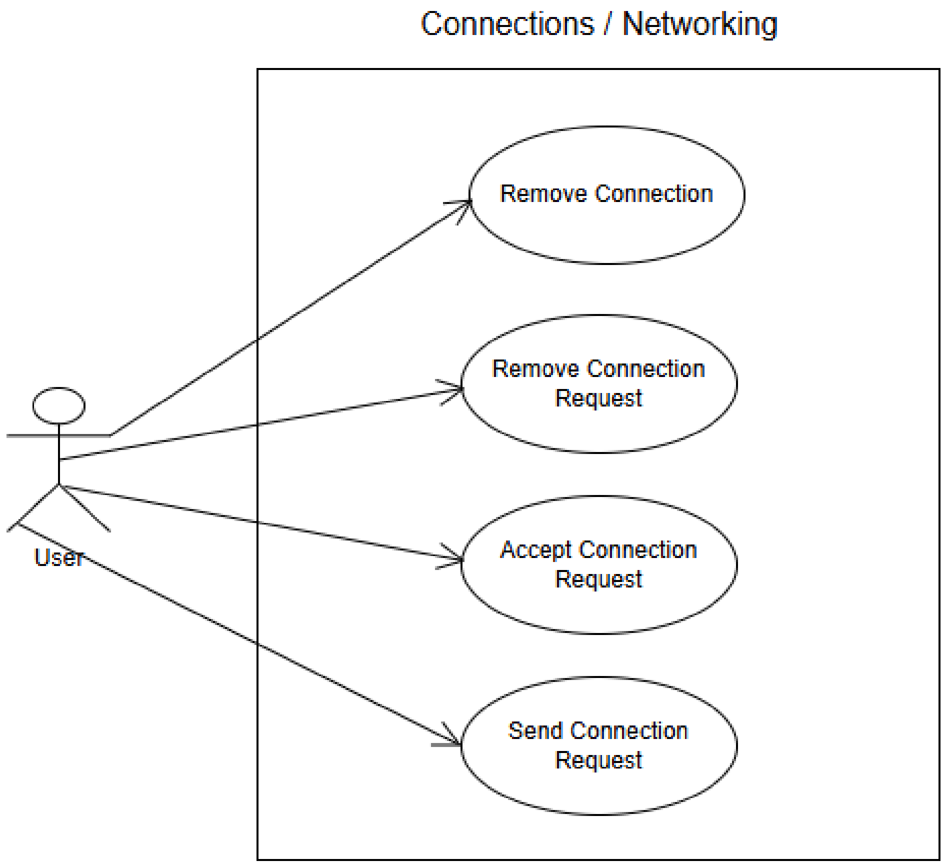


Figure 3.3 Connections / Networking Use Case

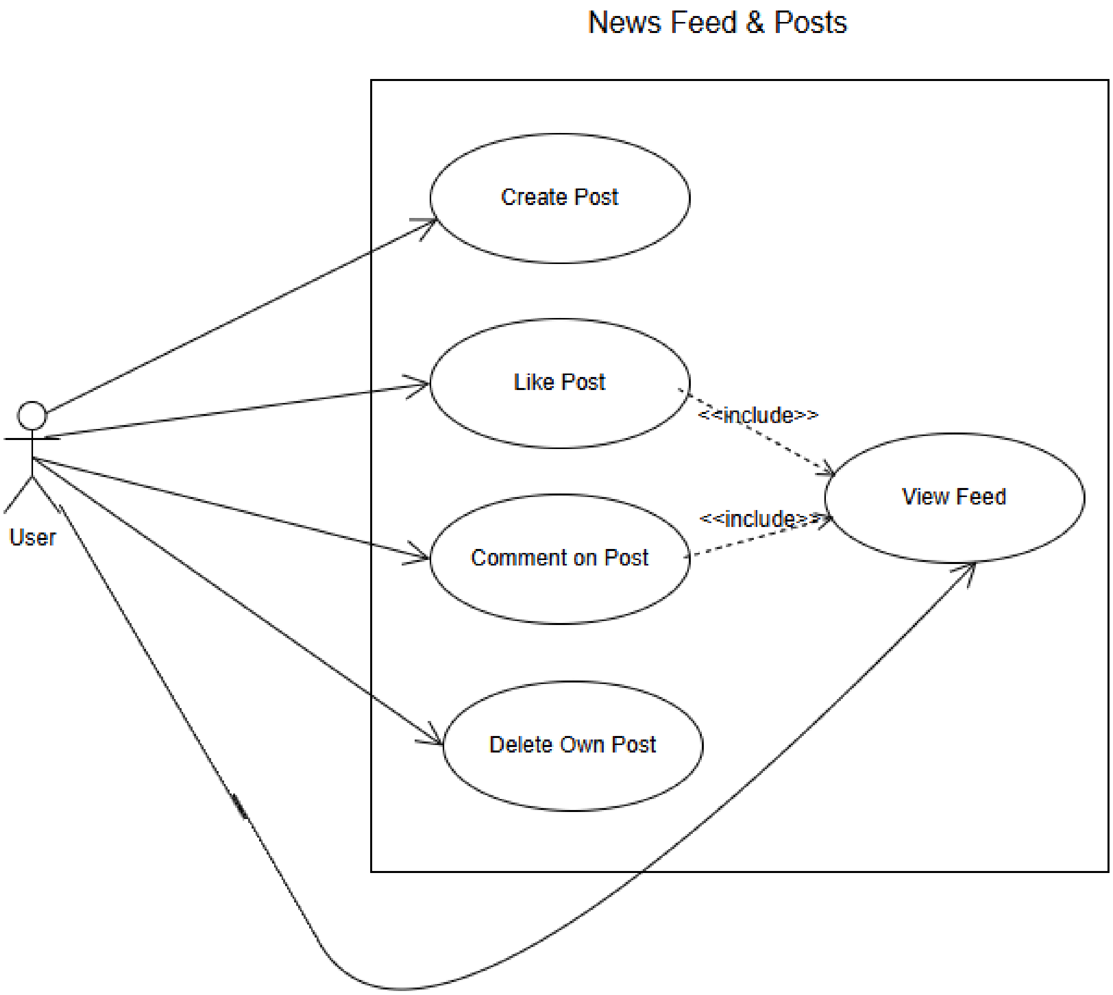


Figure 3.4 News Feed & Posts Use Case

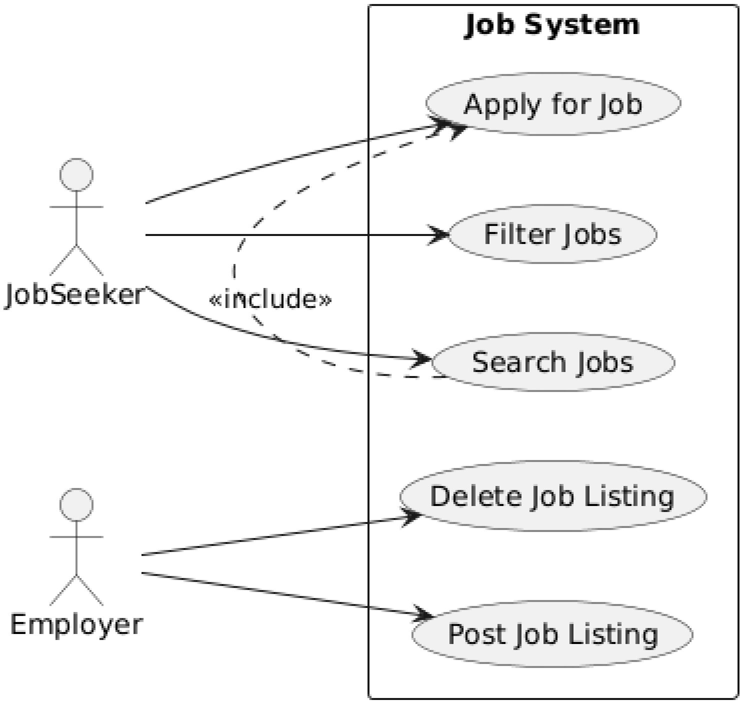


Figure 3.5 Job System Use Case

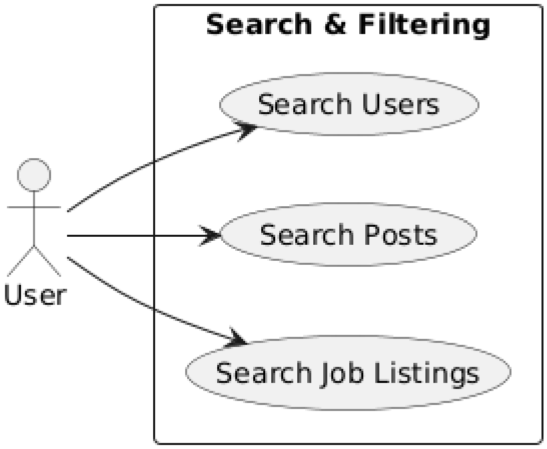


Figure 3.6 Search & Filtering Use Case

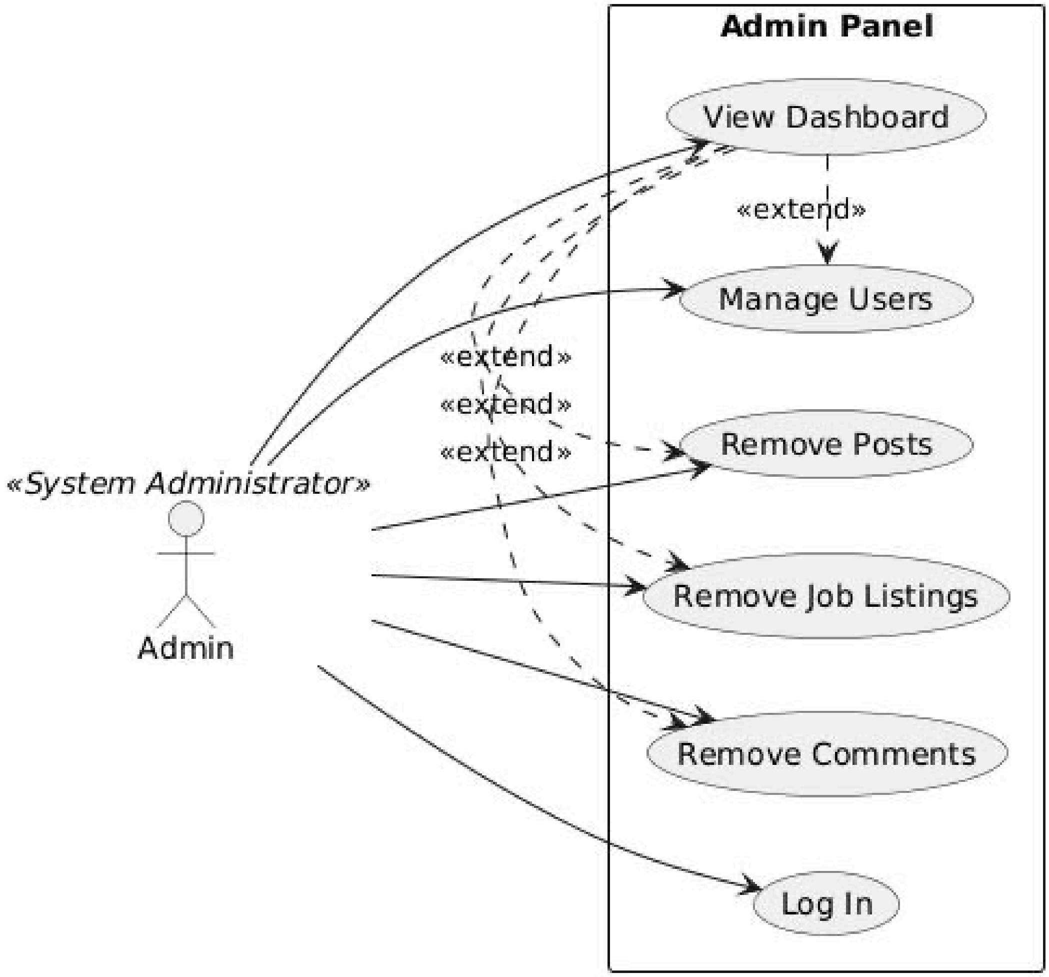


Figure 3.7 Admin Panel Use Case

User Account & Authentication : For users to register and login to the system with some required credentials.

Profile Management : For users to update their information and profile photo

Connections / Networking : For users to be able to control who they can interact with or don't interact with.

News Feed & Posts : Its design purpose is to provide users to comment / like on posts or create their own posts and delete them if they want.

Job System : This part of the system provides some options for employers and users. Users can look for a job and employers can post a job to be shown in the job listing.

Search & Filtering : Users can search jobs , posts and users.

Admin Panel : Some features to control the system's reliability. This feature is only to be logged in by administrators. Admins check suspicious people , inappropriate posts , jobs, comments or images for the system's safety.

4) Class Diagrams

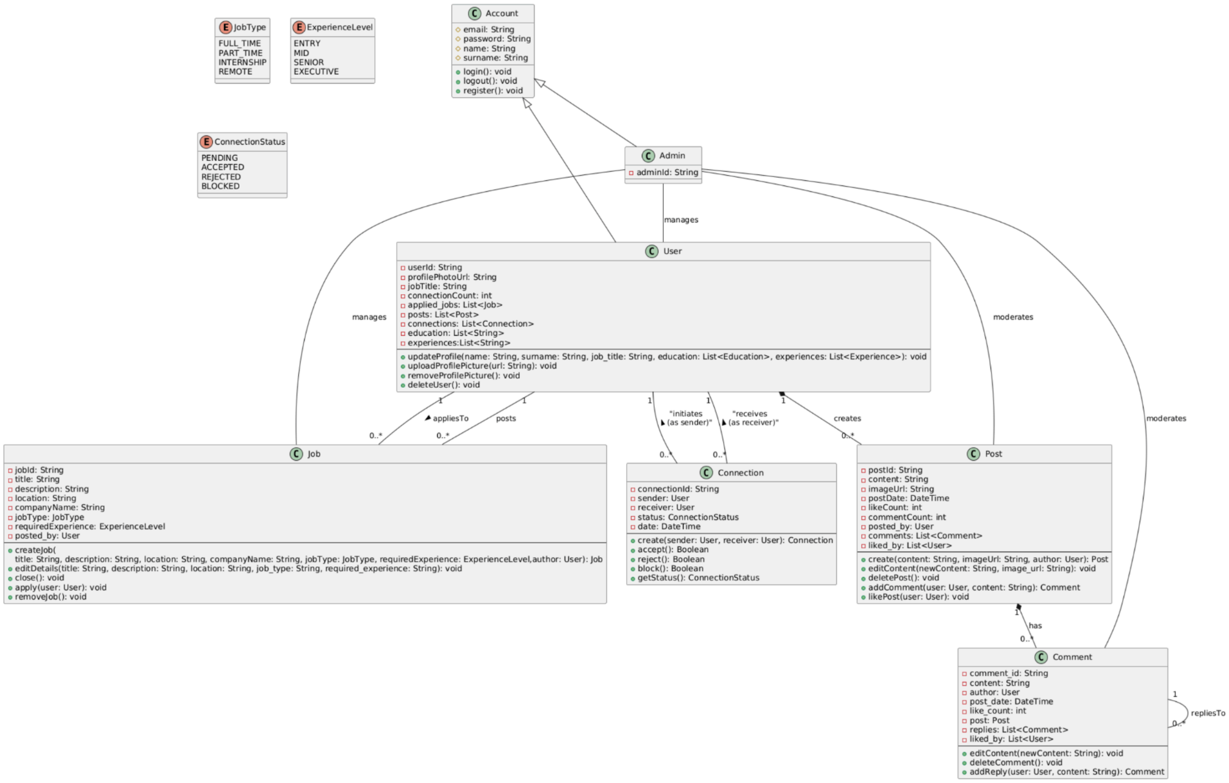


Figure 4.1 UML Class Diagram

This system simulates a professional networking platform, including users, admins, posts, comments, connections, and job listings. The system promotes modularity and logical separation of responsibilities across classes, which ensures maintainability

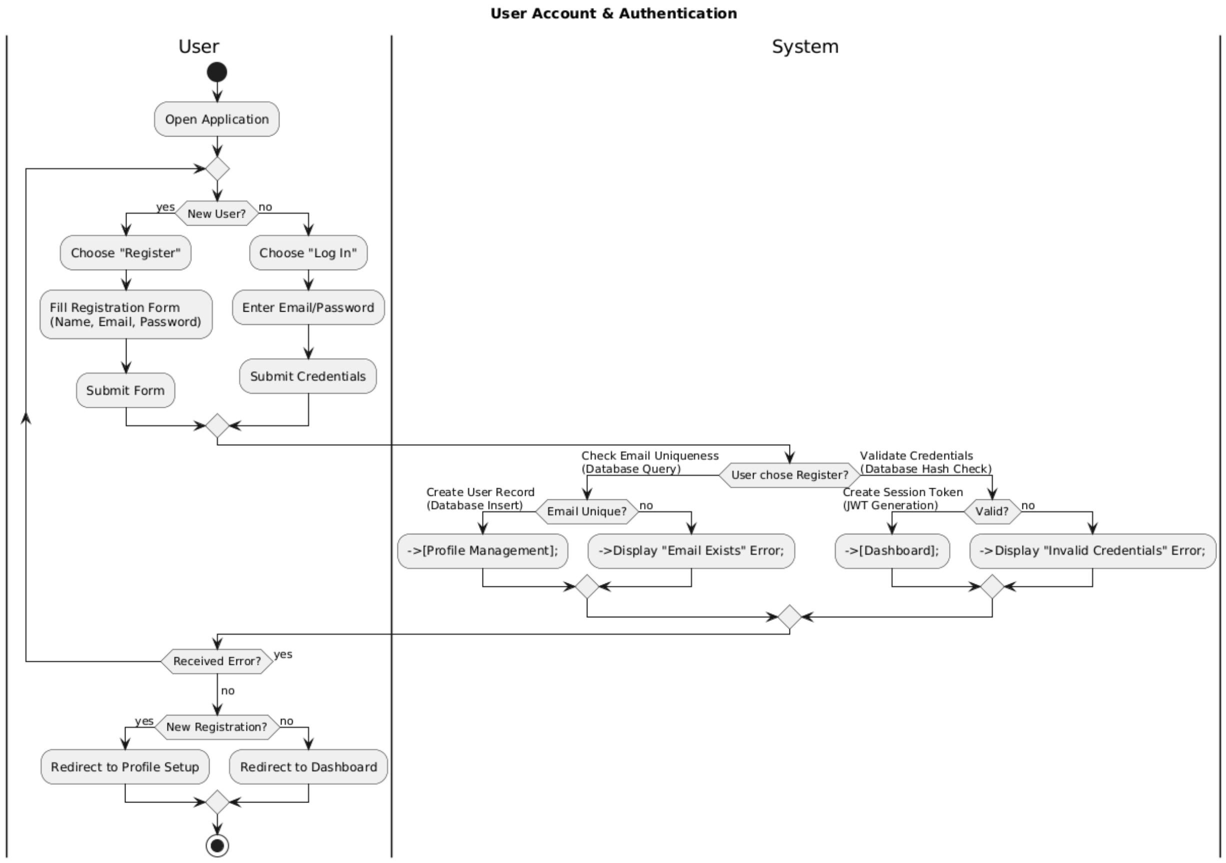
- **The Account class** is a base class that contains common entities such as email, password etc and methods such as login, logout and registration. Both **User** and **Admin** Classes are inherited from this class.
- **The User class** represents individuals using this platform. Each user has a profile containing personal information such as education history, job title, work experience. Moreover, users are allowed to share posts as well as interact with other users' posts by liking or leaving comments.
- **The Admin class** has elevated privileges to moderate the platform. It is responsible for managing users, posts, comments and jobs.

- **The Post class** represents the user generated content and each of them links to an user who created it and contains a list of comments and users who liked the post.
- **The Job class** is a listing of jobs posted by users and includes some attributes about jobs' details as well as a method to update information about those jobs.
- **The Comment class** represents user interactions under posts. Comments can be liked or edited and can be nested through replies.
- **The Connection class** models connection between two users. Each connection has status, pending - accepted - rejected, and links two users.

5) Activity Diagrams

5.1) User Account & Authentication (Login & Signin)

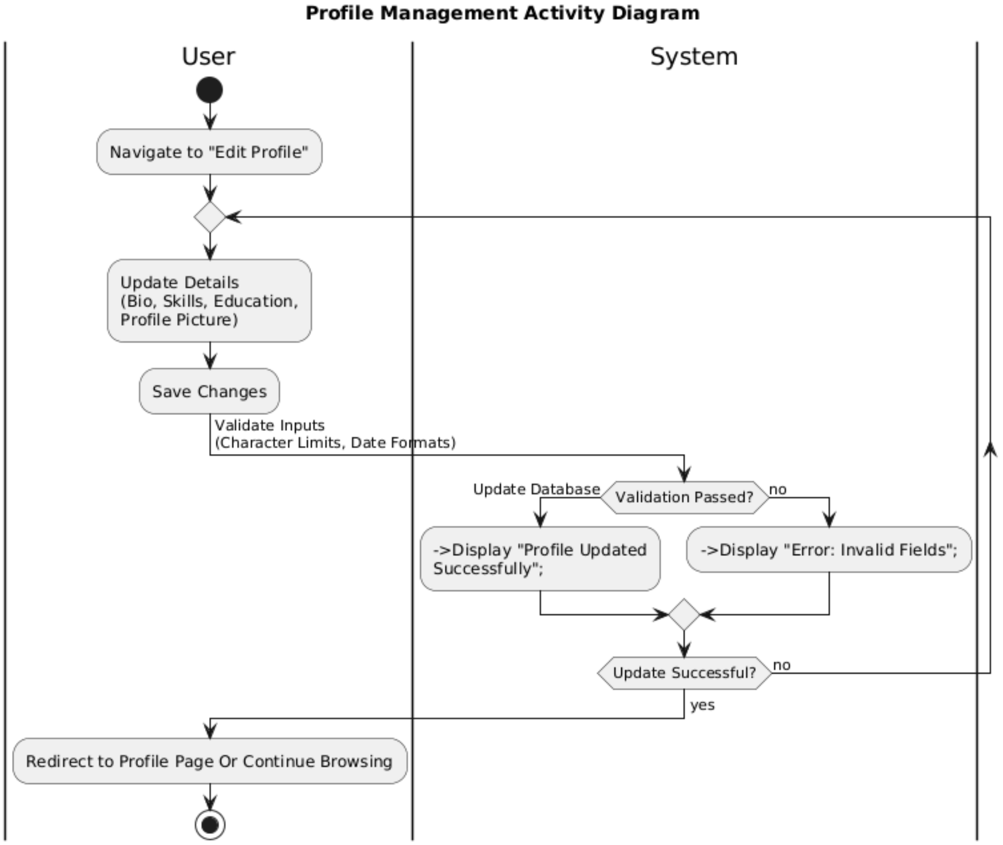
The user begins by opening the application, where they are presented with the option to log in or register. If they choose to register as a new user, they fill out a form with their name, email, and password before submitting it. The system then checks whether the email is already in use—if it's unique, the account is created in the database, and the user is redirected to complete their profile setup. However, if the email exists, an error message appears, prompting the user to try again. For returning users, the login process requires entering their email and password, which the system verifies against stored credentials. Successful authentication grants access to the dashboard, while failed attempts trigger an error, allowing the user to retry or reset their password. Once logged in, users can either continue browsing or log out. The entire process is designed to be straightforward, with clear feedback at each step to guide the user smoothly through registration or login.



5.1: User Account & Authentication Activity Diagram

5.2) Profile Management

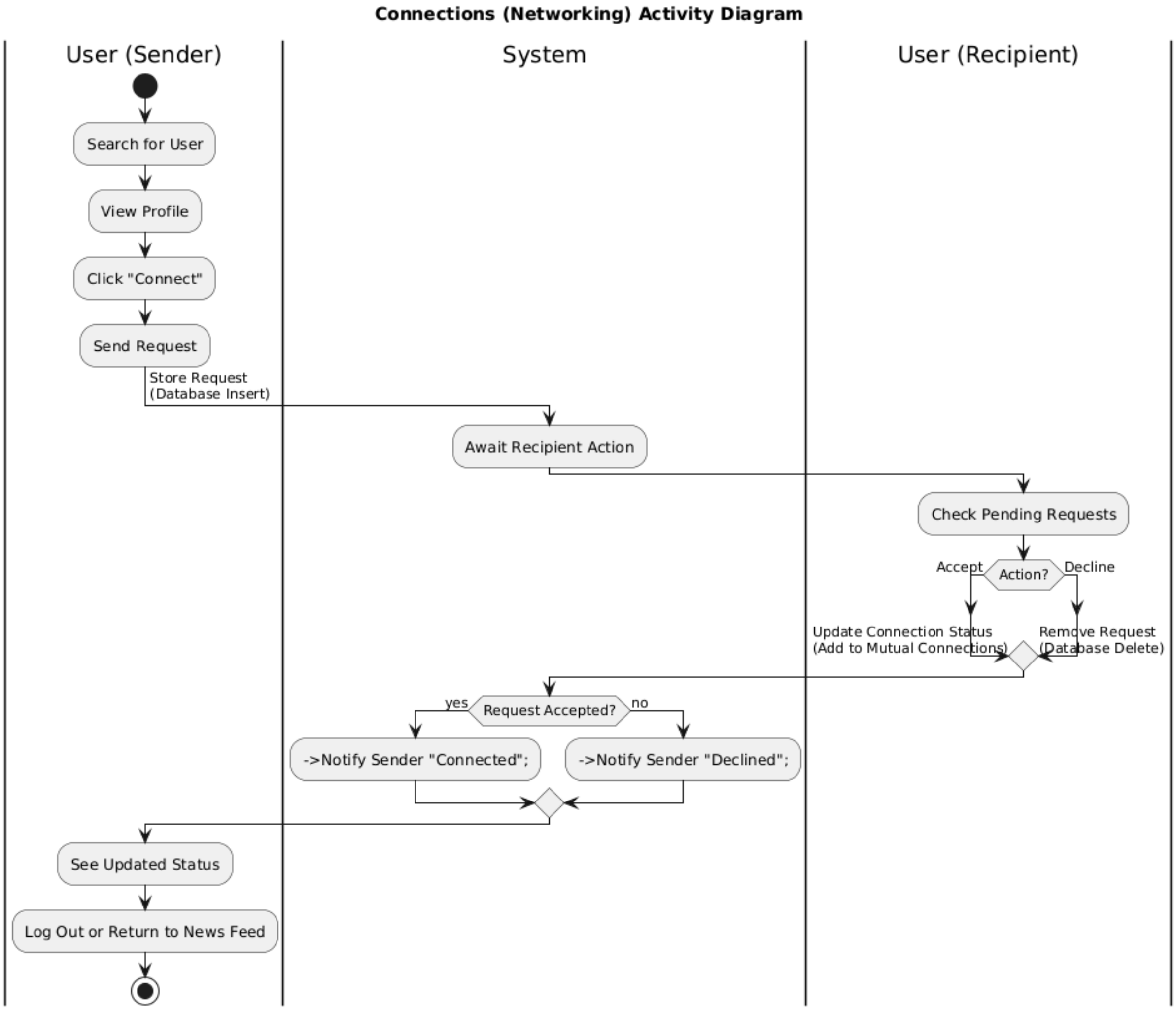
The profile management process begins when the user navigates to their profile page and selects the option to edit their information. They can update various details such as their bio, skills, education history, and profile picture before saving the changes. Upon submission, the system validates the inputs—checking for proper formatting, character limits, and file types—then processes the updates. If all validations pass, the new data is saved to the database, and a confirmation message ("Profile Updated Successfully") is displayed. If errors occur—like an invalid date format or an oversized image—the system highlights the problematic fields with specific feedback, allowing the user to correct and resubmit. Once updated, the user is redirected back to their profile page, where they can either log out or continue browsing. This flow ensures a seamless and intuitive way to maintain accurate and up-to-date profile information.


Profile Management Activity Diagram
User
System
Navigate to "Edit Profile"
Update Details
(Bio, Skills, Education,
Profile Picture)
Save Changes
Validate Inputs
(Character Limits, Date Formats)
Update Database
Validation Passed?
no
->Display "Profile Updated
Successfully";
->Display "Error: Invalid Fields";
Update Successful?
no
yes
Redirect to Profile Page Or Continue Browsing

5.2: Profile Management Activity Diagram

5.3) Connections (Networking)

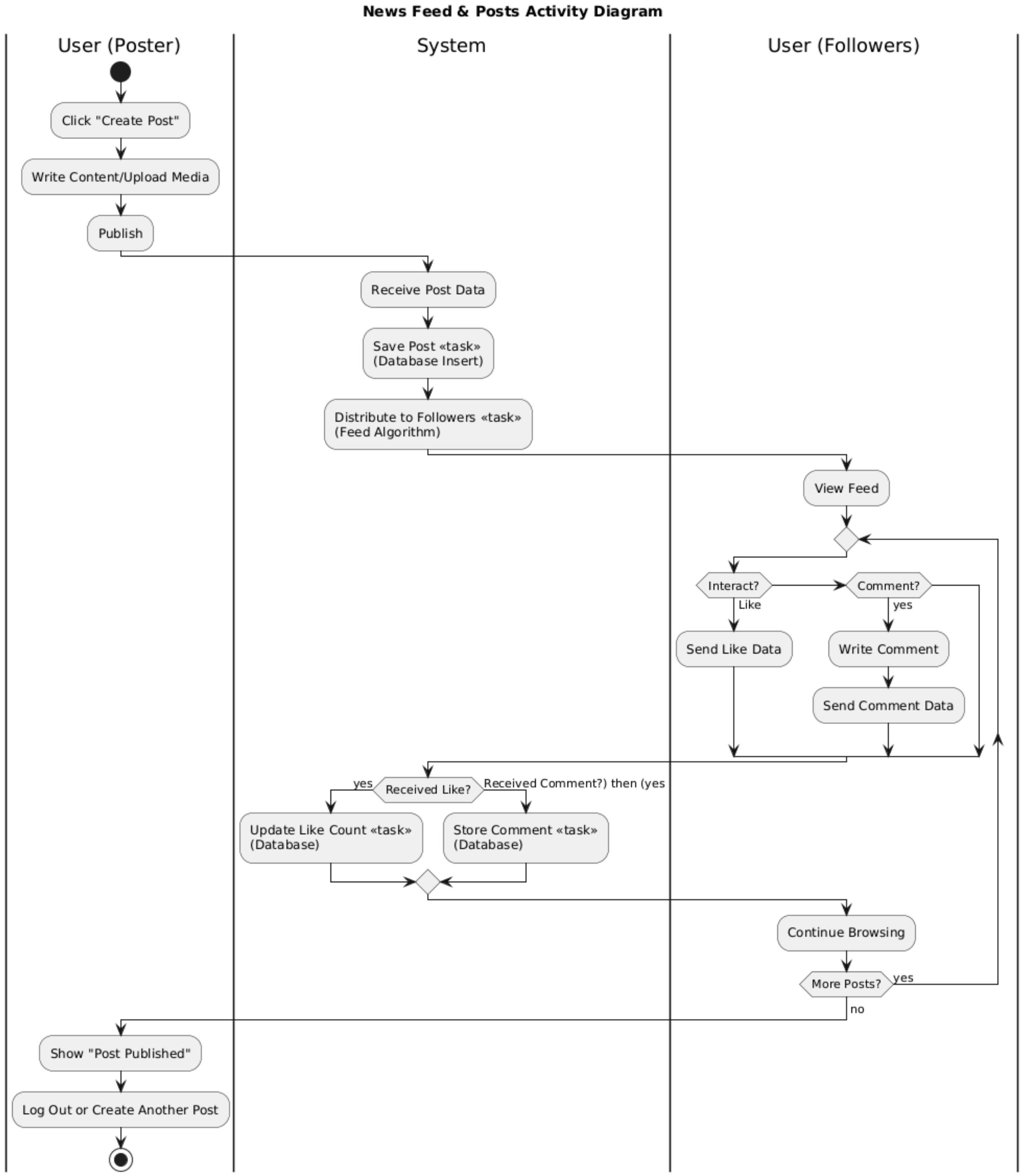
The process begins when a user searches for another profile and clicks the "Connect" button, optionally adding a personalized message. The system then stores this request in the database and notifies the recipient. On the recipient’s side, they can review pending requests and choose to either accept or decline. If accepted, the system updates both users' connection lists, adds them to each other’s networks, and may display a confirmation notification. If declined, the sender receives a polite decline notification. Once connected, users can view each other’s profiles, share content, or message directly.



5.3: Connections (Networking) Activity Diagram

5.4) News Feed & Posts

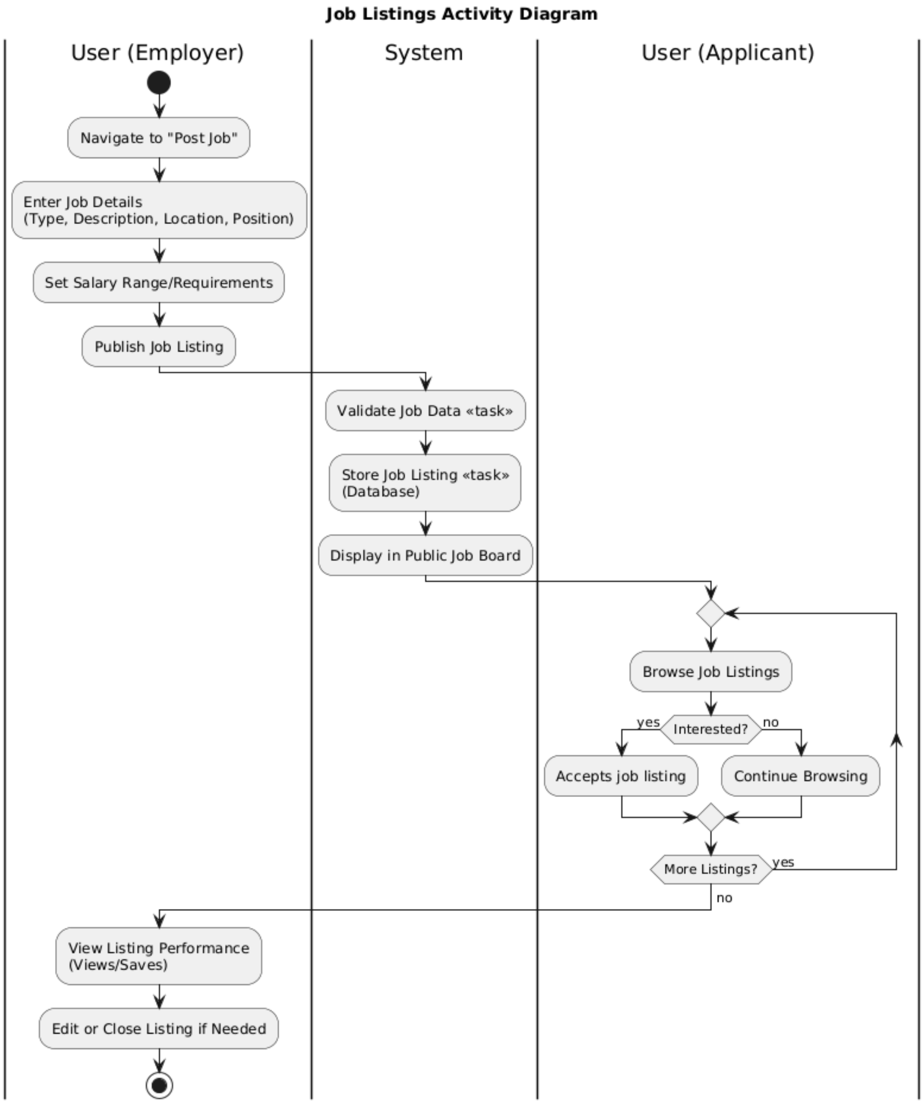
User creates a post by clicking the compose button, entering text, and optionally attaching media files. Upon publishing, the system validates the content, saves it to the database, and distributes it to followers' feeds according to algorithmic sorting. Followers viewing their feed can then engage with posts through reactions (like), comments, or shares. Each interaction triggers real-time updates - likes increment counters, comments and shares propagate content to additional networks.



5.4: News Feed & Posts Activity Diagram

5.5) Job Listings & Applications

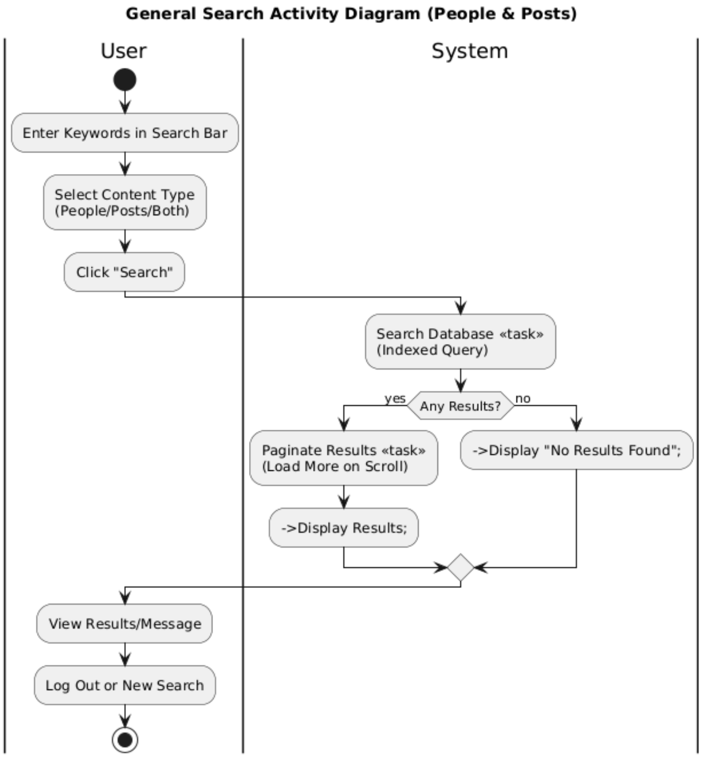
Employers begin by creating a listing, entering key details like job title, description, location, and requirements. The system validates this information before publishing it to the job board. Candidates can then browse listings using the search functionality, where they apply filters for location, job type, experience level, and salary range to refine results. When an interesting job is found, candidates view the full details and submit their application by uploading a resume and cover letter. The system verifies the file formats, stores the application, and notifies the employer.

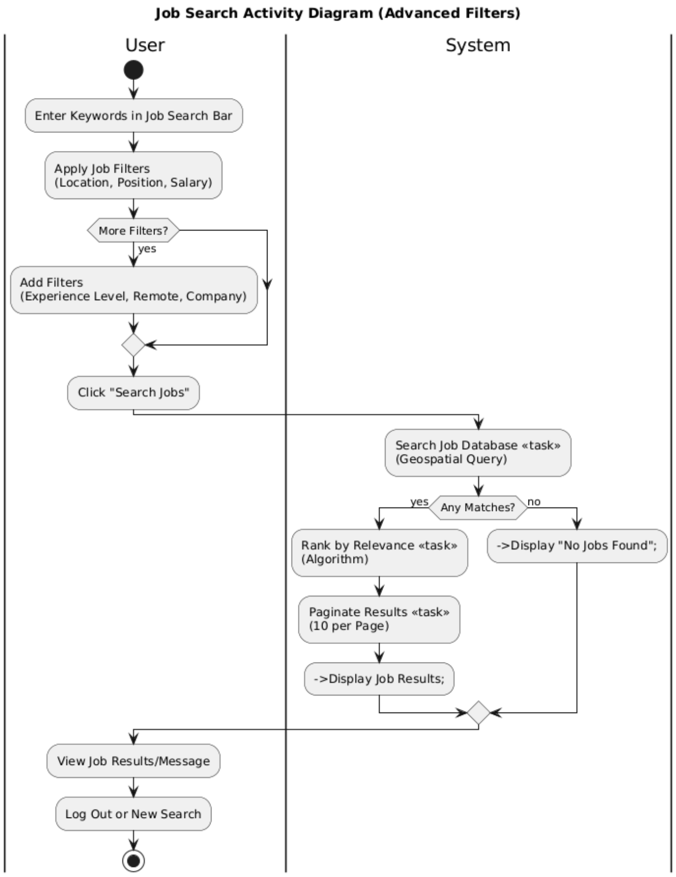

Job Listings Activity Diagram
User (Employer)
System
User (Applicant)
Navigate to "Post Job"
Enter Job Details (Type, Description, Location, Position)
Set Salary Range/Requirements
Publish Job Listing
Validate Job Data «task»
Store Job Listing «task» (Database)
Display in Public Job Board
Browse Job Listings
Interested?
yes
no
Accepts job listing
Continue Browsing
More Listings?
yes
no
View Listing Performance (Views/Saves)
Edit or Close Listing if Needed

5.5: Job Listing and Applications Activity Diagram

5.6) Search & Filtering

The platform features two distinct search systems. The general search scans people and posts, allowing keyword searches with basic filters like content type and date range. The dedicated job search offers advanced filtering by location (with radius selection), position level, company, and remote availability. Both systems process queries through optimized database searches, ranking results by relevance before displaying paginated outcomes. If no matches are found, users receive tailored suggestions to adjust their filters.


General Search Activity Diagram (People & Posts)
User
System
Enter Keywords in Search Bar
Select Content Type (People/Posts/Both)
Click "Search"
Search Database «task» (Indexed Query)
Any Results?
yes
no
Paginate Results «task» (Load More on Scroll)
->Display "No Results Found";
->Display Results;
View Results/Message
Log Out or New Search

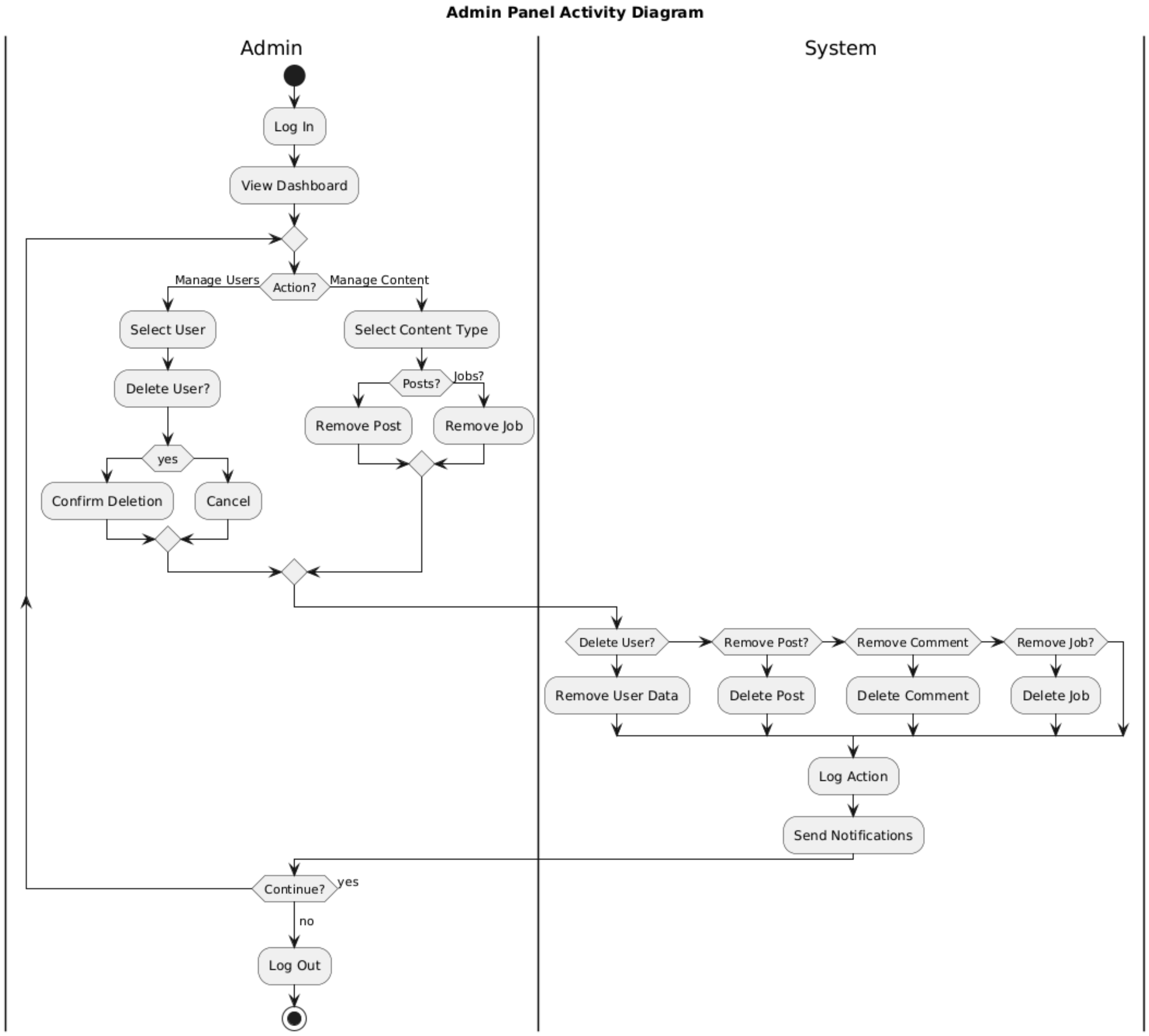

Job Search Activity Diagram (Advanced Filters)
User
System
Enter Keywords in Job Search Bar
Apply Job Filters (Location, Position, Salary)
More Filters?
yes
Add Filters (Experience Level, Remote, Company)
Click "Search Jobs"
Search Job Database «task» (Geospatial Query)
Any Matches?
yes
no
Rank by Relevance «task» (Algorithm)
->Display "No Jobs Found";
Paginate Results «task» (10 per Page)
->Display Job Results;
View Job Results/Message
Log Out or New Search

5.6: Search and Filtering for General Search

5.7: Search and Filtering for Job Search

5.7) Admin Panel

The admin panel provides centralized control for monitoring and managing user accounts. Administrators begin by logging in through a secure authentication system that verifies their elevated privileges. Once authenticated, they access a dashboard displaying all registered users with key details (registration date, activity status). The primary function enables admins to remove problematic accounts—when selecting a user for deletion, the system prompts for confirmation before permanently erasing the account and all associated data (posts, connections) from the database.



5.8: Admin Panel Activity Diagram

6) TEST CASES

· **REGISTRATION**
Users successfully register to the system with valid information.
Input:Name , Surname, email address , and password
Process: The user fills in the information and presses the registration button, and the information is saved to the database. Then the user is directed to the login screen.
Expected Output: User account is created and the user is directed to the login page.

· **REGISTRATION FOR EXISTING EMAIL**
The user tries to register with an email already registered in the system.
Input:Name , surname , existing email address, and password
Process: Whether the information provided by the user is unique is checked via a database query.
Expected Output: 'Email Exists' error message by the system.

· **REGISTRATION WITH INVALID CREDENTIALS**
The user tries to register with information in invalid formats.
Input:Name ,surname , invalid email address or password
Process: The system checks whether the email and password formats are valid. If not, the registration is rejected.
Expected Output: 'Invalid credentials' error message is displayed.

· **LOGIN SUCCESSFULLY**
The user logs into the system by entering the correct inputs.
Input:Email,password
Process: The inputs filled by the user are compared with the data in the database. In the case of match, the user is logged into the system.
Expected Output: The newly registered user is assigned directly to the profile setup screen, while the users who logged in before are directed to the dashboard.

· **UPDATE PROFILE WITH VALID DATA**
The user updates profile information with valid data.
Input:Biography, Skills,Education, Profile Picture
Process: Validates data types and formats (e.g. date format or character limit). If validation is successful, the database is updated.
Expected Output: The message 'Profile Updated Successfully' is displayed.And the user redirected to profile page.

· **UPDATE PROFILE WITH INVALID DATA**
The user tries to update profile information with invalid data.

Input: Input:Biography, Skills,Education, Profile Picture
Process: The system checks the information types and formats. If it is invalid, it does not update.
Expected Output:The message ‘Invalid Fields’ is displayed.

- **SEND CONNECTION REQUESTS**

The user sends a connection request to another user.
Input: The ID of the user who sent the request and ID of the user to whom the request was sent
Process: The system records the connection request in the Connections table .The status field is set to "pending" by default. Then, an informational message is added to the Notifications table.
Expected Output: ‘Request Sent’ message is displayed.

- **ACCEPT REQUEST**

The user whom the request was sent accepts.
Input: Request ID
Process: When the user accepts the request, the status is updated to ‘accepted’. The connections of both users are updated. The user who sent the request is notified that it has been accepted.
Expected Output: Databases updated correctly and connection count updated.The sender receives a notification that the request has been accepted.

- **DECLINE REQUEST**

The user who received the request declines it.
Input: Request ID
Process:The request status is updated to 'declined'.
Expected Output:The sender will receive a notification that the request was not accepted.

- **CREATE NEW POST**

The user publishes a post that contains text or media.
Input:Content of text or media file.
Process: When a post is shared, the system saves the post to the Posts table by 'insert into Posts' in the database. Then, the feed algorithm for user connections is run.
Expected Output: The post becomes visible in the feed of the accounts the user is connected to. The 'Post Published' message is displayed to the user who shared the post.

- **LIKE A POST**

The user likes a post of other users.
Input:Post ID, User ID
Process: The like count in the Posts table is increased.
Expected Output: The number of likes on the post increases.

· **COMMENT ON A POST**
The user comments on a post of other users.
Input: Post ID, User ID,Content of the comment
Process: The comment count in the Posts table is increased.
Expected Output: The comment is added below the post.

· **POST JOB LISTING**
The employer posts a new job.
Input:Type,Description,Location,Position and Salary Range
Process: The system checks whether the information is valid or not. It is saved in the Job List table with Insert Into the Database.
Expected Output: The job listing is published and becomes available for users.

· **ACCEPT JOB LISTING**
Users accept the job listing. They can go to the website and submit a real application.
Input:Job ID,User ID
Process:The user marks the job as "interested" and optionally saves the job listing.

· **SEARCH FOR PEOPLE AND POSTS**
Users enter keywords for the person or post they want to find. Relevant ones are listed.
Input:Keyword
Process: After the user clicks the search button, the keyword is queried in the database.
Expected Output: If there are any matching posts or people, they are listed. Otherwise, the message 'No results found' is displayed.

· **FILTER JOBS**
Users can search and filter jobs according to location,experience level,type or salary.
Input: Keyword according to the filter to be applied. (e.g. Istanbul , Entry Level…)
Process: User enters the keyword as filters e.g. experience or salary. Then the user clicks on the search button. The entered information is checked to see if there is any matching data in the Geospatial query.
Expected Output: If there are, they are listed. Otherwise, the message 'No jobs found' is displayed.

· **USER DELETED BY ADMIN**
Admin can permanently delete a user.
Input:User ID
Process: The admin selects the user they want to delete from among the registered users. When the admin confirms the deletion, all posts, likes and comments belonging to the user are deleted from the database.

Expected Output: User account is deleted.

· **DELETE POST BY ADMIN**
Admin can permanently delete posts.
Input:Post ID
Process: Admin selects a post and confirms deletion.Post is removed from the database.
Expected Output: Post is removed from the user profile.

· **DELETE COMMENT BY ADMIN**
Admin can permanently delete comments.
Input:Comment ID
Process:Admin selects a comment and confirms deletion.The selected comment will also be deleted from the comments in the database.
Expected Output:Comment is removed.

· **DELETE JOB BY ADMIN**
Admin can permanently delete jobs.
Input:Job ID
Process:Admin selects a job and confirms deletion.The selected job is also deleted from the jobs in the database.
Expected Output:Job is removed.