Recruitment System

1.	Tanish Dhariwal	–	22BD1A661U
2.	Kushal Bang	–	22BD1A660W
3.	Gaurav Jain	–	22BD1A660G
4.	Tanay Bansal	–	22BD1A661R
5.	Akanksha Parkala	–	22BD1A661B

Problem Statement:

A Recruitment System plays a critical role in matching the right talent with organizational needs. It should allow employers to post jobs, track applications, and filter candidates based on skills and qualifications, while enabling job seekers to find relevant opportunities that align with their experience and preferences. Effective job categorization—by role, skills, experience, location, and other criteria—is essential for targeted filtering, ensuring that recruiters can easily narrow down candidates and job seekers can refine their searches.

Real-time updates are vital; job listings must reflect availability accurately, with filled positions promptly removed and new postings quickly added. The system should integrate seamlessly with databases (SQL and NoSQL) for storing job postings and candidate profiles, ensuring data privacy and security to protect sensitive information.

An intuitive user interface helps recruiters and candidates navigate efficiently. Additionally, features like communication tools, interview tracking, and job bookmarking enhance the user experience. A responsive and secure recruitment platform benefits both employers and job seekers by simplifying the hiring process.

Software Requirement Specification

for

Recruitment System

Version 1.0 approved

Prepared by:

Tanish Dhariwal – 22BD1A661U
Kushal Bang – 22BD1A660W
Gaurav Jain – 22BD1A660G
Tanay Bansal – 22BD1A661R
Akanksha Parkala – 22BD1A661B

Keshav Memorial Institute of Technology

18-10-2024

Table of Contents

Revision History

Name	**Date**	**Reasons for Changes**	**Version**
Week-1	25-09-2024	SRS Creation (Introduction)	1.0
Week -2	19-10-2024	SRS Documentation – Use case & Class Diagram	2.0

1.Introduction

1.1 Purpose:

The primary purpose of developing this system is to optimize the college recruitment process. The system will enable the sorting of qualified applicants based on their qualifications and the specific requirements of companies. Additionally, it can predict suitable companies for applicants or suggest companies where they are likely to be placed, based on their skills and areas of interest. Another objective of the software is to facilitate registration and communication between students, companies, and the placement office.

1.2 Document Convention:

Heading:

Font-Size:16

Font-Style: Bold

Font: Times New Roman

Subheading:

Font-Size:14

Font-Style: Bold

Font: Times New Roman

Content:

Font-Size:12

Font: Times New Roman

1.3 Intended Audience and Reading Suggestions:

This document serves as a prototype for the recruitment system and will be useful for different audiences in various ways. Developers can use it to design and implement the project, while managers can refer to it for managing costs and timelines. It will assist HR professionals in understanding the features of the system and how it stands out from traditional recruitment methods.

Candidates can assess whether the platform meets their expectations and provide feedback for improvements if necessary. Testers can use the document to ensure thorough testing of the system's features. Additionally, the outlined functional and non-functional requirements offer valuable insights for developers during the development process.

1.4 Product Scope:

Our recruitment system is similar to existing solutions but introduces new advancements. The platform will be available 24/7, with scheduled maintenance once a month. The primary objective is to streamline the recruitment process and ensure a smooth experience for both applicants and companies. Through this system, candidates will save significant time by applying online, and companies can efficiently sort and evaluate applicants based on their qualifications and requirements.

The platform will guide companies in selecting suitable candidates through an intelligent matching system that considers skills, qualifications, and areas of interest. Applicants will be asked to rate their experience with the recruitment process, helping us identify areas for improvement. Feedback about specific companies can also be collected to ensure quality placements and address any concerns.

User data will be safeguarded with robust security measures to prevent data breaches. We will conduct regular surveys to gather insights on improving the recruitment process and implementing necessary changes. Additionally, a comprehensive database will be maintained to manage applicant and company profiles effectively.

1.5 References:

We referenced various platforms and resources for this recruitment system, including job portals like LinkedIn, Indeed, Glassdoor, and Monster. Additionally, we consulted academic resources on software requirements specifications to ensure a comprehensive approach.

2. Overall Description

2.1 Product Perspective

Our project builds upon existing recruitment systems and incorporates key functionalities from these platforms. It will allow recruiters to set up job postings, candidates to browse available positions, and administrators to approve or reject new job listings while maintaining a comprehensive list of job categories.

The system will be mobile-friendly, featuring highly customizable themes. Additionally, it includes advancements such as an advanced search engine for easy job searching, robust data security measures to protect user information from data breaches, and an engaging, interactive user interface designed for ease of use for all stakeholders.

2.2 Product functions

2.2.1 Administrator

a) Administrators will have the ability to insert, modify, and delete job listings posted by recruiters.

b) Can approve or reject recruiter accounts and job postings based on platform policies and compliance with standards.

c) Can manage student accounts, including activating or deactivating user access.

d) Will be able to notify students and recruiters about relevant updates, deadlines, and job fairs.

e) Administrators can generate system usage reports and monitor the status of job applications and interview outcomes.

2.2.2 Recruiter

a) Recruiters will be able to create, modify, and delete job postings.

b) They can review student applications and filter candidates based on qualifications, skills, and other criteria.

c) Recruiters will have the ability to schedule interviews with shortlisted candidates and send out interview invites.

d) Can provide interview feedback and decide to make job offers to selected candidates.

e) Recruiters can update the status of job applications (e.g., pending, accepted, rejected).

2.2.3 Student

a) Students will receive notifications about new job postings, relevant application deadlines, and interview schedules.

b) Students can upload and update their resumes and profiles with qualifications, skills, and experience.

c) They can search and filter job postings based on location, industry, role, or other preferences.

d) Students will have the ability to apply for jobs, view the status of their applications, and respond to interview invitations.

e) Students can track their interview results and any offers extended by recruiters.

2.3 Operating Environment

This recruitment system shall operate in all major web browsers, including Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge. For reference, we are considering compatibility with the latest versions of these browsers.

a) The system will be accessible on Windows operating systems, including Windows 7, 8, 10, and later versions, as well as on macOS.

b) A minimum processor requirement of Intel Pentium 3 or equivalent is recommended.

c) The system will require at least 1 GB of RAM for optimal performance.

2.4 User Characteristics

The users of this recruitment system include candidates, recruiters, and administrators responsible for maintaining the platform.

a) **Candidates**: Individuals seeking job opportunities who will upload their resumes and apply for positions. They are expected to have basic knowledge of computers and internet browsing.

b) **Recruiters**: Companies or hiring managers who will post job listings, review applications, and manage candidate interactions. They should also have a fundamental understanding of online tools and processes.

c) **Administrators**: Users responsible for maintaining the overall system, including approving new job listings and managing user accounts. Administrators should have knowledge of the internal modules and be able to troubleshoot any issues that arise.

2.5 Design and Implementation Constraints

Users may access this recruitment system using any device with a stable internet connection and internet browsing capabilities.

a) **Data Storage**: User information, job listings, and application details will be stored in a database accessible by the system.

b) **Availability**: The system will be operational 24/7 to accommodate users' needs.

c) **Platform Compatibility**: The software does not require a specific operating system to run; it can function on any platform, including mobile devices.

d) **User Authentication**: Users must enter their correct username and password to access their accounts and perform actions within the system.

2.6 User Documentation

The user documentation for the Campus Recruitment System will include comprehensive guides tailored to the needs of different user groups. There will be detailed user manuals for students, recruiters, and administrators, explaining how to navigate the system and use its various features. For students, the documentation will guide them through registering, browsing job listings, applying for jobs, tracking application statuses, and managing their profiles. Recruiters will have a manual to help them post job opportunities, review applications, and communicate with potential candidates. The administrator manual will cover tasks such as managing users, overseeing the job postings, and maintaining system integrity. Additionally, training materials will be provided to help all user groups quickly learn how to efficiently use the system. For technical support, a troubleshooting section will be included to address common issues and provide solutions.

2.7 Assumptions and Dependencies

The recruitment system requires the following third-party products:

a) **Database Management**: Microsoft SQL Server will be utilized to store the database.

b) **Development Framework**: ASP.NET will be employed for developing the application.

c) **Data Extraction**: JSON will be used for retrieving data from the database.

The success of this recruitment system depends on:

a) **Internet Access**: Reliable internet service must be available to all users.

b) **User Proficiency**: All candidates and recruiters should be comfortable using computers, laptops, or mobile devices and possess sufficient knowledge to navigate the platform.

c) **User-Friendly Design**: The recruitment system must be intuitive and easy to use.

d) **Efficient Search Mechanism**: The search functionality should be simple, fast, and secure to facilitate quick access to job listings and applications.

3. Specific Requirements

3.1 User Interfaces

The Campus Recruitment System will provide multiple user interfaces tailored to different user roles, such as Students, Admins, and Company Representatives. These interfaces will be accessible via web browsers and mobile devices, ensuring usability and responsiveness.

3.1.1 Student Dashboard:

a) Provides features for students to create and update their profiles, including personal details, resume uploads, and academic information.
b) Students can search and filter job postings based on job type, location, and company.
c) Allows students to apply for jobs directly from the dashboard, view the status of their applications, and accept or reject job offers.
d) Displays notifications for interview schedules and application updates.

3.1.2 Company Dashboard:

a) Allows company representatives to create and update job postings with details like job title, description, location, requirements, and compensation.
b) Provides a view of applicants for each job and allows companies to shortlist candidates.
c) Companies can schedule interviews with selected candidates and send interview invites.
d) Enables companies to make job offers and update job statuses (e.g., open, closed).

3.1.3 Admin Dashboard:

a) Admins can manage user accounts (students and companies), including activation, deactivation, and profile updates.
b) Allows admins to oversee job postings and applications, and remove inappropriate content.
c) Provides reporting tools to generate analytics on job applications, student activity, and placement rates.
d) Enables bulk actions like importing/exporting user data and sending announcements to users.

3.2 Hardware Interfaces

Server Requirements:

a) Minimum specifications: 16 GB RAM, 8-core CPU, 500 GB SSD for the database server.
b) Web and application servers should have at least 8 GB RAM, 4-core CPU, and 250 GB SSD.

c) Servers should support virtualization for better resource management and scalability.

Client Devices:

a) Compatible with standard PCs, laptops, and mobile devices with 4 GB RAM and 1 GHz processor.
b) Devices must support a modern web browser (Chrome, Firefox, Safari, or Edge) with JavaScript enabled.
c) Requires stable internet connection for real-time updates, with a minimum speed of 2 Mbps.

3.3 Software Interfaces

Database:

a) Uses MySQL or PostgreSQL for relational data storage, handling user accounts, job postings, applications, and interview schedules.
b) Database must support ACID (Atomicity, Consistency, Isolation, Durability) properties to ensure data reliability.
c) Regular synchronization with backup servers for data redundancy.

Backend:

a) Developed using Java Spring Boot or Python Django to provide a RESTful API for interactions between the client and server.
b) Includes business logic for processing user requests, managing sessions, and applying access control.
c) Integrates with third-party services for email, SMS, and external job boards.

Frontend:

a) Built with React.js or Angular for a dynamic user experience and single-page application behaviour.
b) Uses HTML5 and CSS3 for responsive design to ensure compatibility with different screen sizes.
c) Supports localization to allow customization of content based on user preferences.

Authentication:

a) Integrated with OAuth 2.0 to enable secure login using email accounts or social media platforms (e.g., Google, LinkedIn).
b) Stores user session data securely and supports token-based authentication for API access.

3.4 Communication Interfaces

Email Notifications:

a) Automatically sends emails for critical actions like account verification, job application confirmations, interview schedules, and offer letters.
b) Supports customizable email templates for different types of notifications.

API Integrations:

a) Integrates with third-party job boards to share job listings and expand reach.
b) SMS Gateway integration for sending interview reminders and updates to students.
c) Provides a RESTful API for external systems to query job listings and retrieve analytics.

Network Protocols:

a) Operates over HTTPS using SSL/TLS encryption to secure data transmission between clients and servers.
b) Uses WebSockets for real-time updates, such as notifying students of new job postings or interview schedule changes.
c) Firewall and VPN support for secure access by admin users and company representatives.

4. System Features

4.1 Global Database

4.1.1 Description and Priority:

The global database feature stores all relevant data, including student profiles, company information, job postings, and application records. It acts as the primary repository, facilitating data management across the recruitment process. This feature is of High Priority, as the system relies heavily on efficient and secure data storage and retrieval.

4.1.2 Stimulus/Response Sequences:

a) Stimulus: Input data is entered by students, companies, or administrators.
b) Response: The system securely stores the data in the database and makes it accessible for job searches, applications, and administrative tasks.

4.1.3 Functional Requirements:

a) REQ-1: The system must securely store student, recruiter, job, and application data.
b) REQ-2: Administrators must be able to manage the database (add, update, delete entries).
c) REQ-3: The database must ensure fast data retrieval for an optimal user experience.

4.2 Login and User Authentication

4.2.1 Description and Priority:

The login feature provides secure access to the system based on user roles (students, recruiters, and admins). High Priority, as it is crucial for maintaining security and data privacy.

4.2.2 Stimulus/Response Sequences

Stimulus: A user enters their login credentials.

Response: The system verifies the credentials and grants or denies access based on the user's role.

4.2.3 Functional Requirements:

a) REQ-4: The system must authenticate user credentials and control access.
b) REQ-5: Different interfaces must be provided for each user role (student, admin, recruiter).
c) REQ-6: After multiple failed login attempts, the system must lock the account and allow password recovery.

4.3 Job Posting and Application Management

4.3.1 Description and Priority:

This feature enables companies to post job listings and students to apply for them. High Priority, as it directly supports the core function of the recruitment system.

4.3.2 Stimulus/Response Sequences:

Stimulus: A recruiter posts a job or a student applies.

Response: The system stores the job listing or the application and provides a confirmation to the respective user.

4.3.3 Functional Requirements:

a) REQ-7: Recruiters must be able to post, edit, and remove job listings.
b) REQ-8: Students should be able to search and filter jobs by different criteria (location, role, etc.).
c) REQ-9: The system must track all student applications for recruiters to review, shortlist, or reject.

4.4 Notifications and Alerts

4.4.1 Description and Priority:

This feature notifies users (students and recruiters) about important events, such as new job postings, application status updates, and interview schedules. Medium Priority, as it enhances user engagement but isn't critical to core system functionality.

4.4.2 Stimulus/Response Sequences:

Stimulus: An event occurs, such as a job posting or an application status update.

Response: The system sends an email or SMS notification to the user.

4.4.3 Functional Requirements:

REQ-10: The system must send notifications for key events like new jobs, application statuses, and interviews.

REQ-11: Users should be able to manage their notification preferences.

REQ-12: All sent notifications must be logged for future reference.

4.5 Report Generation

4.5.1 Description and Priority:

The report generation feature provides insights on the recruitment process, such as job postings, applications, and placements. High Priority, as it aids decision-making for recruiters and administrators.

4.5.2 Stimulus/Response Sequences:

Stimulus: An admin or recruiter requests a report on the recruitment data.

Response: The system retrieves the relevant data and generates the requested report.

4.5.3 Functional Requirements:

a) REQ-13: The system must generate reports on student applications, job postings, and recruitment success rates.
b) REQ-14: Admins and recruiters should be able to customize the report parameters.
c) REQ-15: Reports must be exportable in various formats (PDF, CSV).

5. Nonfunctional Requirements

5.1 Performance Requirements

a) Response Time: System should respond within 3 seconds for basic operations and 5 seconds for complex queries like report generation.
b) Scalability: Supports up to 5,000 concurrent users.
c) Throughput: Handles 100 requests per second during peak times.
d) Availability: Ensures 99.5% uptime with scheduled maintenance during off-peak hours.

5.2 Safety Requirements

a) Data Backup: Daily backups with 24-hour recovery capability.
b) Error Handling: User-friendly error messages and secure error logging.
c) User Session Timeout: Sessions time out after 15 minutes of inactivity.

5.3 Security Requirements

a) Authentication: Multi-factor authentication (MFA) for admin accounts; role-based access control.
b) Encryption: Uses HTTPS for data transfer; passwords are hashed with bcrypt.
c) Vulnerability Protection: Safeguards against SQL Injection, XSS, and CSRF.
d) Data Privacy: Complies with regulations like GDPR and CCPA.

5.4 Software Quality Attributes

a) Usability: Simple, intuitive interfaces with user guides.
b) Maintainability: Modular codebase with thorough documentation.
c) Reliability: Data integrity checks; recovery from unexpected shutdowns.
d) Portability: Compatible with major web browsers and mobile devices.

5.5 Business Rules

a) Student Applications: Students can apply to up to 5 jobs per day.
b) Job Posting: Jobs remain active for 30 days unless extended by companies.
c) Interview Scheduling: 3 days' notice required for scheduling; status updated within 48 hours.
d) Offer Acceptance: Students can accept only one job offer at a time.
e) Data Retention: Application records retained for 1 year; inactive accounts reviewed after 2 years.

6. Other Requirements

6.1 User Registration and Authentication

The **Campus Recruitment System** must allow users (students, recruiters, and administrators) to register by providing their details and secure credentials. Authentication will be performed through a username and password combination or via third-party authentication providers (e.g., Google, LinkedIn). All user data should be encrypted and securely stored to prevent unauthorized access.

6.2 Job Search and Filtering

The system should allow students to search for job opportunities using various filters such as job title, company name, location, salary range, job type (e.g., full-time, internship), and industry. Recruiters should be able to search for potential candidates based on qualifications, skills, and other relevant filters.

6.3 Application Tracking and Management

The system must enable students to apply for job openings and track the status of their applications. Recruiters should have a dashboard to review, shortlist, and manage applications efficiently. Students should receive updates when their application status changes, such as when they are shortlisted or rejected.

6.4 Interview Scheduling

The system should facilitate interview scheduling between recruiters and students. Recruiters must be able to propose available time slots, and students should have the ability to confirm the scheduled interviews. Automatic reminders should be sent to both parties before the scheduled interview.

6.5 Resume Upload and Management

Students should have the ability to upload and manage their resumes and other relevant documents such as certificates or portfolios. They should also be able to update these documents whenever necessary.

6.6 Feedback and Ratings

Recruiters should have the option to provide feedback and rate students based on their interview performance. Similarly, students should be able to provide feedback on their recruitment experience, allowing continuous system improvement.

6.7 Notifications and Alerts

The system should send notifications and alerts to users regarding key events such as job postings, application deadlines, interview schedules, and application status updates. These notifications should be sent via email or SMS to ensure timely communication.

6.8 Data Privacy and Security

The system should adhere to strict data privacy policies to protect sensitive information such as personal details, job application history, and feedback. Only authorized users should have access to sensitive data, and role-based access control must be implemented to ensure that data is not exposed to unauthorized individuals.

6.9 Mobile-Friendly Interface

The **Campus Recruitment System** must be optimized for mobile devices to allow students and recruiters to access the platform easily from smartphones or tablets. This includes job searching, applying, and receiving notifications on the go.

6.10 Integration with Third-Party Services

The system should support integration with third-party services such as LinkedIn for profile import, Google Calendar for interview scheduling, and other external recruitment platforms to widen job and candidate pools.

6.11 Reports and Analytics

The system should generate reports for administrators and recruiters, including statistics on student applications, job postings, company participation, and placement success rates. These reports should help in decision-making and strategy formulation for the campus recruitment process.

6.12 Technical Support

A dedicated technical support system should be available to assist users in case of technical issues or questions. This can include live chat, email support, and a knowledge base of frequently asked questions (FAQs).

Appendix A: Campus Recruitment Glossary

a) **Job Application:** This is the process where students submit their resumes and other relevant documents to apply for job openings posted by recruiters on the platform. It allows students to showcase their skills and experiences for available positions.

b) **Resume Upload:** Students can upload their resumes to the system for potential employers to review. This feature ensures that their profiles are up-to-date and accessible to recruiters during the hiring process.

c) **Job Posting:** Recruiters use the system to post available job openings, providing details such as job description, required qualifications, and application deadlines. These postings are visible to students for applying.

d) **Shortlisting:** After reviewing the applications, recruiters select suitable candidates for the next stage of the recruitment process, such as interviews. Shortlisted students are notified via the system.

e) **Interview Scheduling:** The system allows recruiters and students to schedule interview dates. Notifications are sent to both parties, and the platform ensures smooth coordination for interview times.

f) **Offer Letter:** Once the selection process is complete, recruiters can issue digital offer letters to successful candidates through the system. This document formally invites the candidate to join the company.

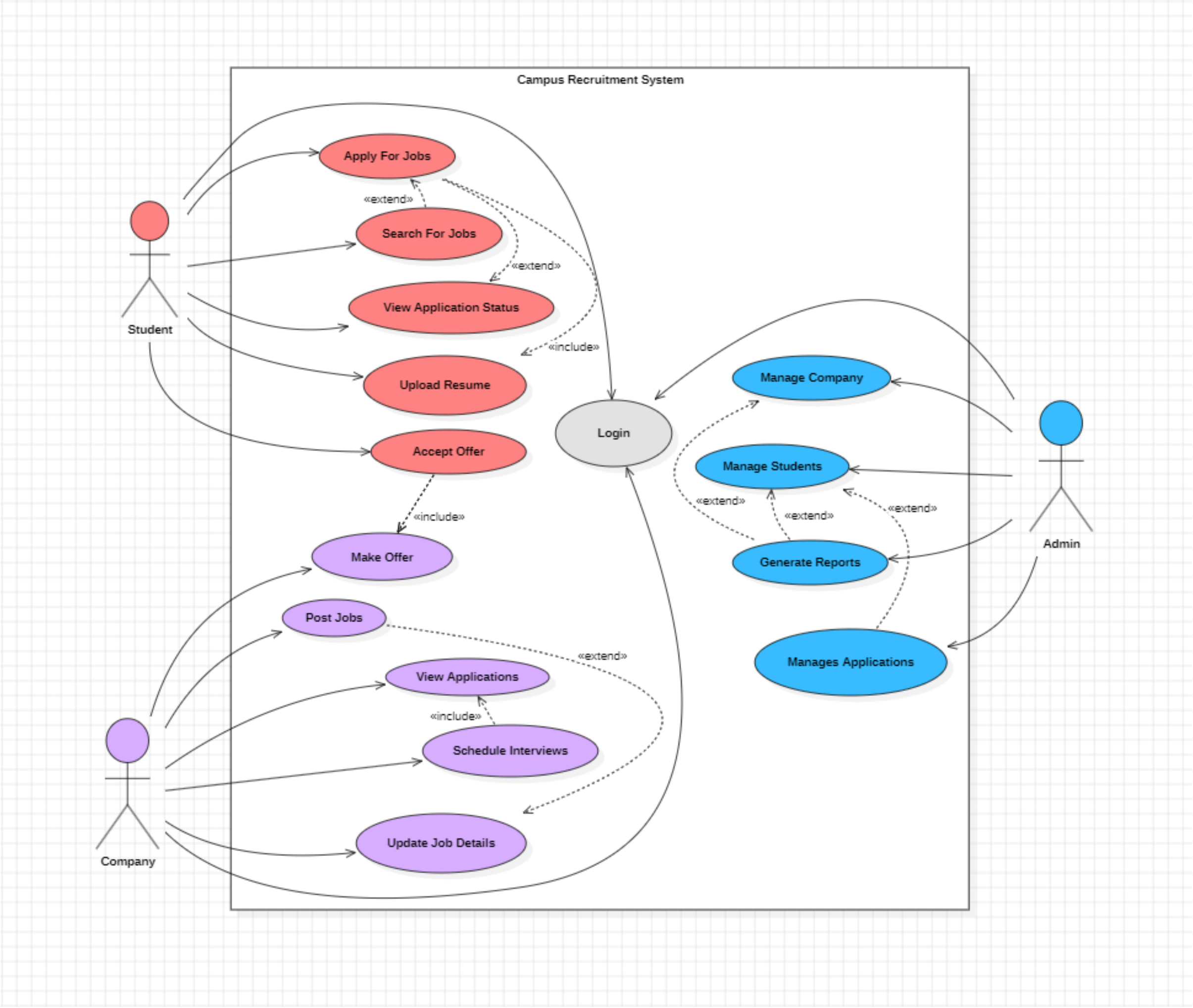
Appendix B: Analysis Models

Use Case template:

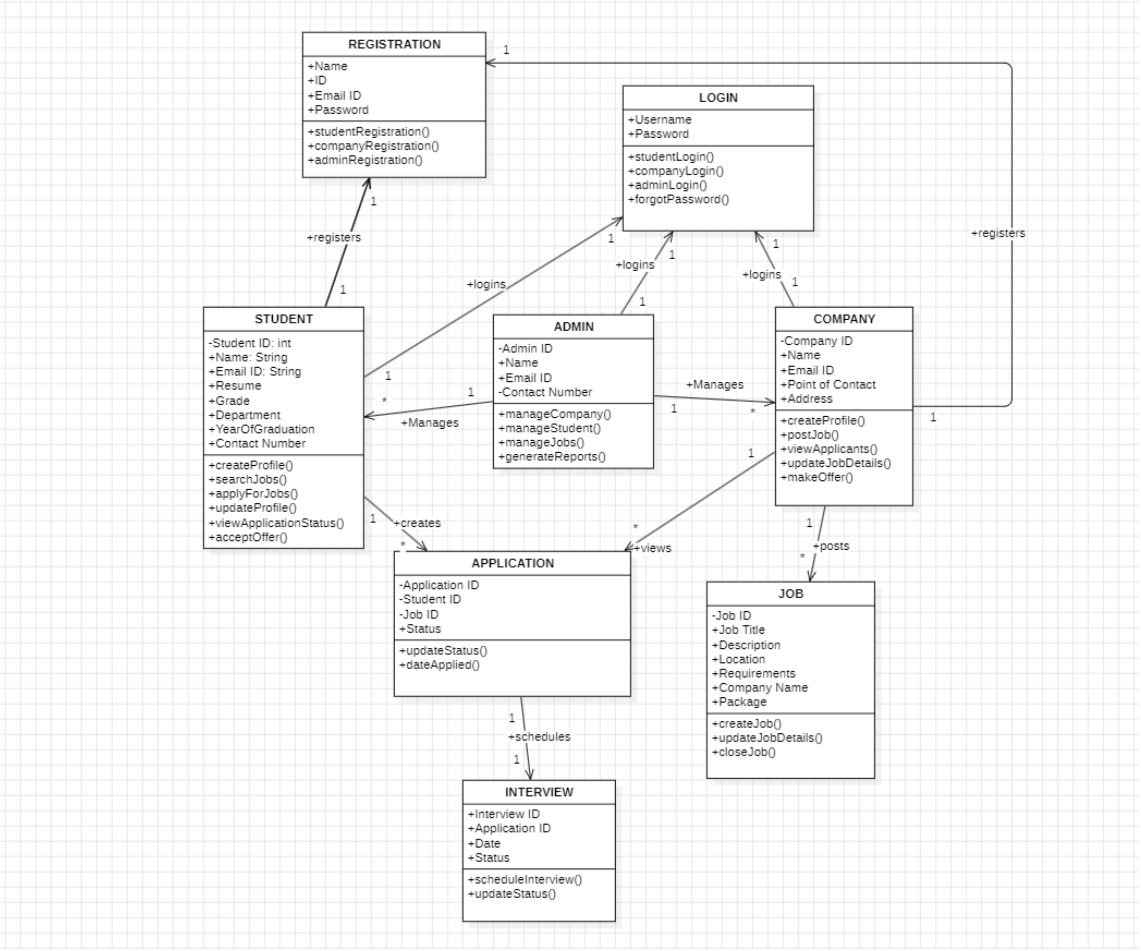
Use Case ID	084318919716		
Use Case Name	Recruitment System		
End Objective	To enable students, recruiters, and administrators to efficiently manage job applications, interviews, and recruitment processes in the campus recruitment system.		
Created by	1. Tanish Dhariwal 2. Gaurav Jain 3. Kushal Bang 4. Tanay Bansal 5. Akanksha Parkala	On (date):	October 18, 2024
User/Actor	Student (User), Recruiter, Administrator, System		
Trigger	A user initiates an action, such as registering for the system, applying for a job, or scheduling an interview.		
Basic/Normal Flows			
User Actions		System Actions	
The user logs into the system.		The login page prompts the user to enter a valid username and password.	

The student uploads their resume. The recruiter posts a job opening. The student applies for a job. The recruiter shortlists applicants. The administrator schedules an interview. The student attends the interview and receives feedback. The student checks their application status.	The system allows the student to upload a resume, which is stored in their profile for future job applications. The system provides a form for the recruiter to input job details such as title, description, requirements, and deadlines. The system records the application, sends a confirmation notification, and updates the student's application status. The system allows the recruiter to filter and shortlist candidates based on criteria such as qualifications and experience. The system sends interview details to the selected students and updates their application status. The system stores the feedback from the recruiter and updates the student's profile with their interview status. The system displays the current status of the application, whether it's under review, shortlisted, or rejected.
Exception Flows	
User Actions	System Actions
The user enters incorrect login credentials. The student attempts to apply for a closed job. The recruiter encounters a scheduling conflict. The student tries to upload a resume of invalid format.	The system displays a message indicating invalid username or password and prompts the user to try again. The system prevents the application and displays a message indicating that the application deadline has passed. The system sends a notification to both the recruiter and the student and prompts the recruiter to reschedule the interview. The system displays an error message indicating that the uploaded file is in an unsupported format and suggests valid formats like PDF, DOCX, etc.

Use Case Diagram:



Class Diagram:



Appendix C: To Be Determined List

- **Industries to Prioritize for Campus Recruitment:** Specific sectors or industries that the system should focus on attracting for campus placements.
- **Student Qualification Criteria:** Detailed criteria for students' eligibility to apply for certain job roles, based on academic performance or skills.
- **Company Engagement Levels:** Defining levels of company participation in recruitment drives (e.g., premium partnerships, exclusive access to top talent).
- **Event Scheduling and Management:** Finalizing schedules for recruitment events, workshops, and webinars to maximize student engagement.
- **Platform User Roles and Permissions:** Determining different access levels for students, recruiters, and administrators to streamline operations.