Project Detail Designing & Analysis

Title: A holistic approach to campus recruitment and student profile analysis for placement.

❖ Algorithm

▪ Student Module

Step 1 - Start
Step 2 - If valid credentials are provided by student, proceed to Step 3; otherwise, repeat Step- 2.
Step 3 - Allow students to create, update, and delete their profiles.by securely storing data.
Step 4 - Participates in skill assessments in various formats.
Step 5 - Allow students to apply for jobs by completing skill assessments.
Step 6 - Notify students of relevant job opportunities based on their preferences.
Step 7 - Provide students with real-time updates/track on the progress/status of their job applications.
Step 8 - Enable easy access for students to view alumni success stories.
Step 9 - Stop

▪ HR Module

Step 1 - Start
Step 2 - If valid credentials are provided by HR , proceed to Step 3; otherwise, repeat Step-2.
Step 3 - Allow HR professionals to manage and store job vacancies.
Step 4 - Create, manage, and score quizzes for candidates as part of the assessment process.
Step 5 - Consider candidate profiles and job vacancy requirements to make suitable recommendations.
Step 6 - Create a hiring workflow for candidate selection, interviews, and offers.
Step 7 - Allow rating and feedback on candidate profiles evaluation and rating.
Step 8 - Set up secure communication for HR professionals.
Step 9 - Enable securely release assessment scores to candidates.
Step 10 - Notify candidates about their selection status, interviews, and offers.
Step 11 - Stop

▪ Admin Module

Step 1 - Start
Step 2 - If valid credentials are provided by Admin, proceed to Step 3; otherwise, repeat Step- 2.
Step 3 - Allow administrators to add, update, or remove student profiles and job listings.
Step 4 - Allow administrators to add, edit, or remove alumni success stories.
Step 5 - Integrate reporting and analytics for data-driven decisions such as user activity, job statistics, and success stories.
Step 6 - Stop

1) System Architecture:

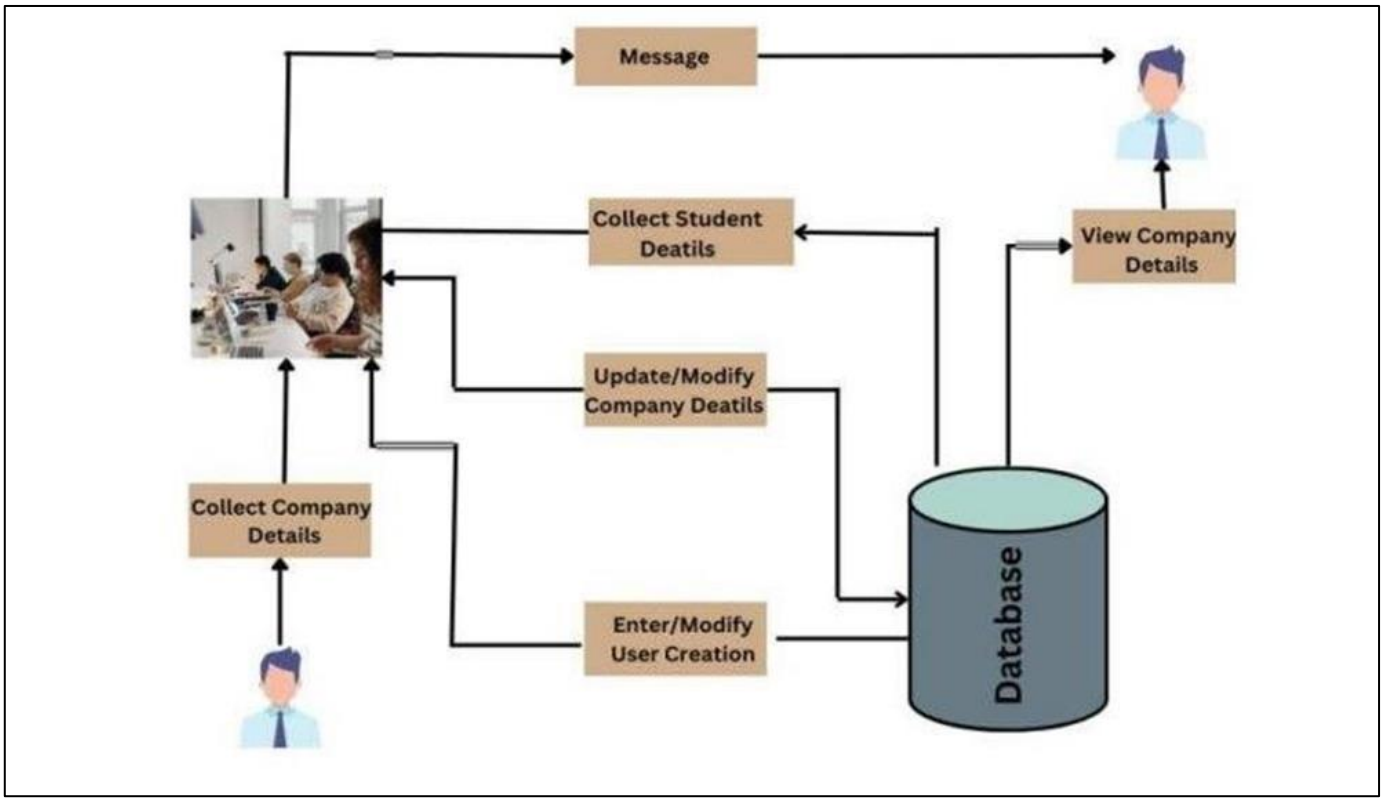

Fig 1.1 System Architecture

Above figure explains the working of the proposed system. Next, The ICRPT Application represents the core software system. Users interact with the application through different modules, including the "Student Module," Admin Module, and HR Module.

The Core System Features encompass essential functionalities of ICRPT, including user profiles, job posting, assessments, comparison tools, notifications, data management, security, reporting, and analytics.

Teacher(TPO) : Teacher can create and manage students profiles. They can also post job related information. Teacher can communicate with students to provide guidance, feedback, or additional information regarding job placements. Access to analytics and reports related to student performance, job placement and the effectiveness of their recommendations.

Student: Students can create detailed profiles, including their education history, skills, achievements and career aspirations. Access to job listings posted by teachers or HR personnel. Students can apply for jobs, track application statuses. They can also receive updates on application statuses, interview invitations and other relevant information. Optional assessments or quizzes to evaluate and show their skills.

HR: HR personnel can create and manage company profiles including company information job listing and contact details & post job opening for students to apply. view and manage application from students including reviewing resume and profile. They can also communicate with students and teachers regarding job listing and selection processes. Provide feedback on the platform usability and the quality of candidates

The application design should ensure a seamless and efficient interaction between these modules to facilitate successful job placements.

2) Entity-Relation Diagram (ER):

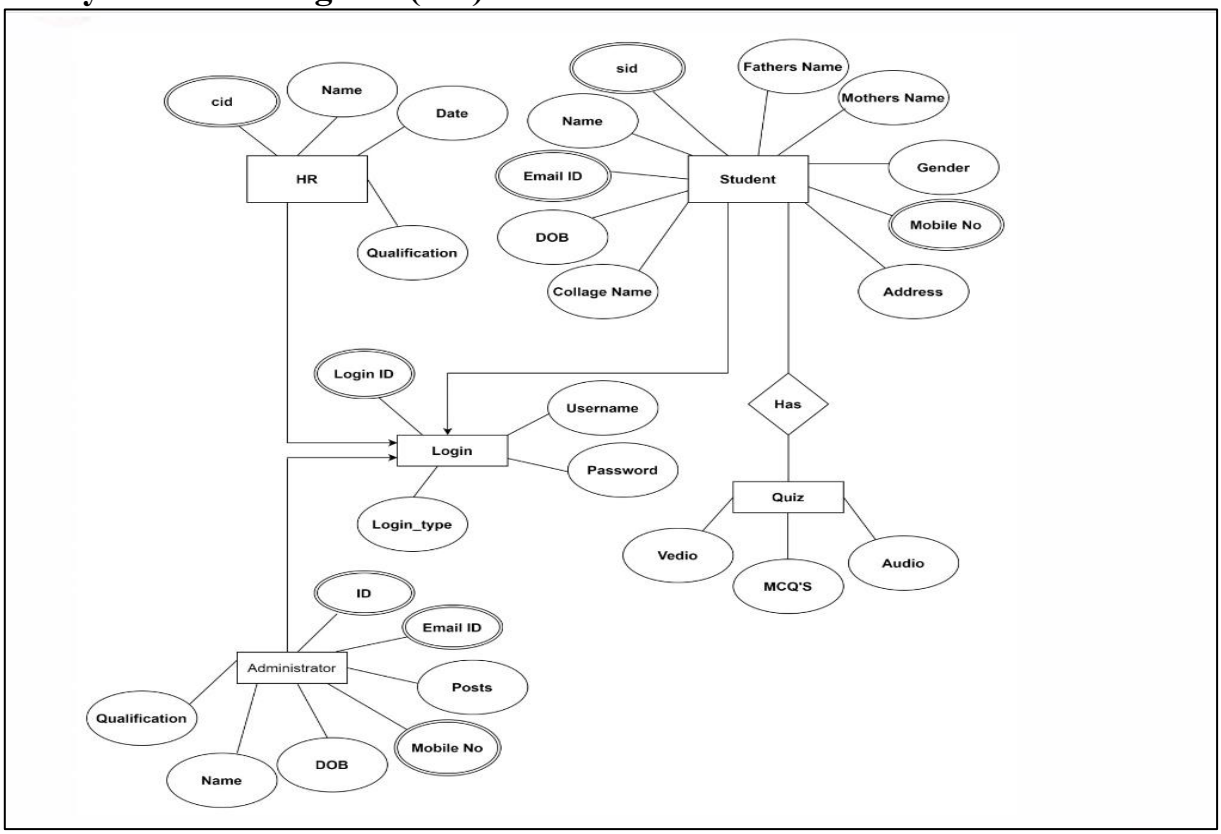

Fig 1.2 Entity-Relation Diagram (ER)

The application can be displayed by means of graphical notation known as Entity Relationship diagram. The ER model describes data as entities, relationship and attributes. For a Training and Placement System, an ER diagram would represent the structure of the database or data model that stores information about students, companies, job placements, and related entities. Here's a detailed explanation of the components you might find in an ER diagram for such a system

1. Entities:

- **Student:**
  Attributes: StudentID (Primary Key), Name, Contact Info, Gender, DOB, Email Id, College name, Address, Mobile no etc.
- **Admin:**
  Attributes: AdminID (Primary Key), Name, Email, Mobile no, Qualification Posts etc.
- **HR:**
  Attributes: HRID (Primary Key), Name, Email, Qualification, etc.
- **Profile:**
  This entity represents the profiles created by students.
  Attributes: ProfileID (Primary Key), StudentID (Foreign Key), Bio, Resume, Father_name, Mother_name, Sem_score, etc.
- **Assessment:**
  Represents assessments or quizzes created by HR.
  Attributes: AssessmentID (Primary Key), HRID (Foreign Key), Quiz Title, Date, etc.
- **Assessment Score:**
  Represents the scores of students in assessments.
  Attributes: ScoreID (Primary Key), AssessmentID (Foreign Key), StudentID (Foreign Key), Score, Date, etc.

2. Relationships:

- **Applies:**
  Connects Student to Placement to show that a student has applied for a specific placement.
  Attributes: Application Date, Status (e.g., Pending, Accepted, Rejected).
- **Creates:**
  Links Student to Profile to indicate that students create and manage their profiles.
  No additional attributes required.
- **Manages Profile:**
  Connects Admin to Profile to allow administrators to view and manage student profiles.
  No additional attributes required.
- **Adds Company:**
  Connects Admin to Company to show that administrators can add company information.
  No additional attributes required.
- **Posts Job:**
  Connects Admin to Placement to allow administrators to post job-related posts.
  Attributes: Job Description, Posting Date, etc.
- **Lists Jobs:**
  Connects HR to Placement to show HR can view job listings.
  No additional attributes required.
- **Creates Quiz:**
  Connects HR to Assessment to allow HR to create quizzes.
  Attributes: Quiz Title, Date, etc.
- **Manages Scores:**
  Connects HR to Assessment Score to enable HR to manage assessment scores.
  Attributes: Score, Date, etc.
- **Views Profile:**
  Connects HR to Student to allow HR to view student profiles.
  No additional attributes required.

3) Data Flow Diagram

i. DFD (Level-0)

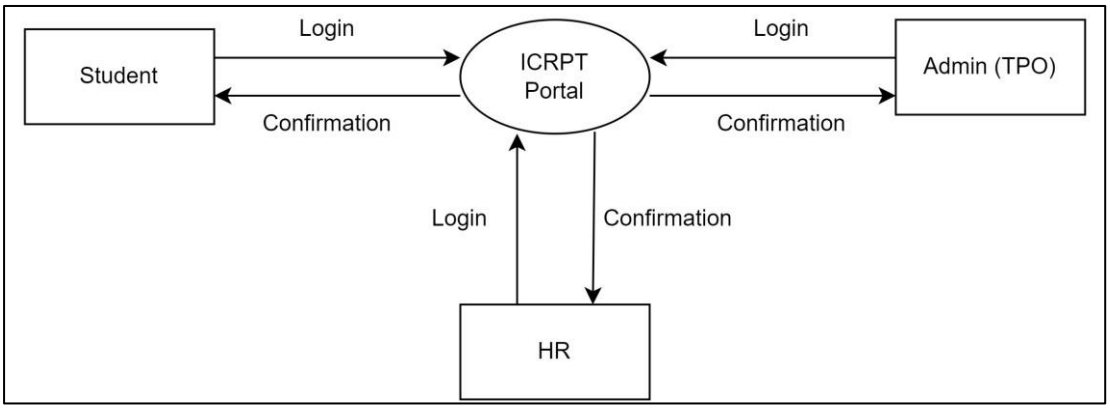

Fig.1.3 DFD Level 0

ii. DFD (Level-1)

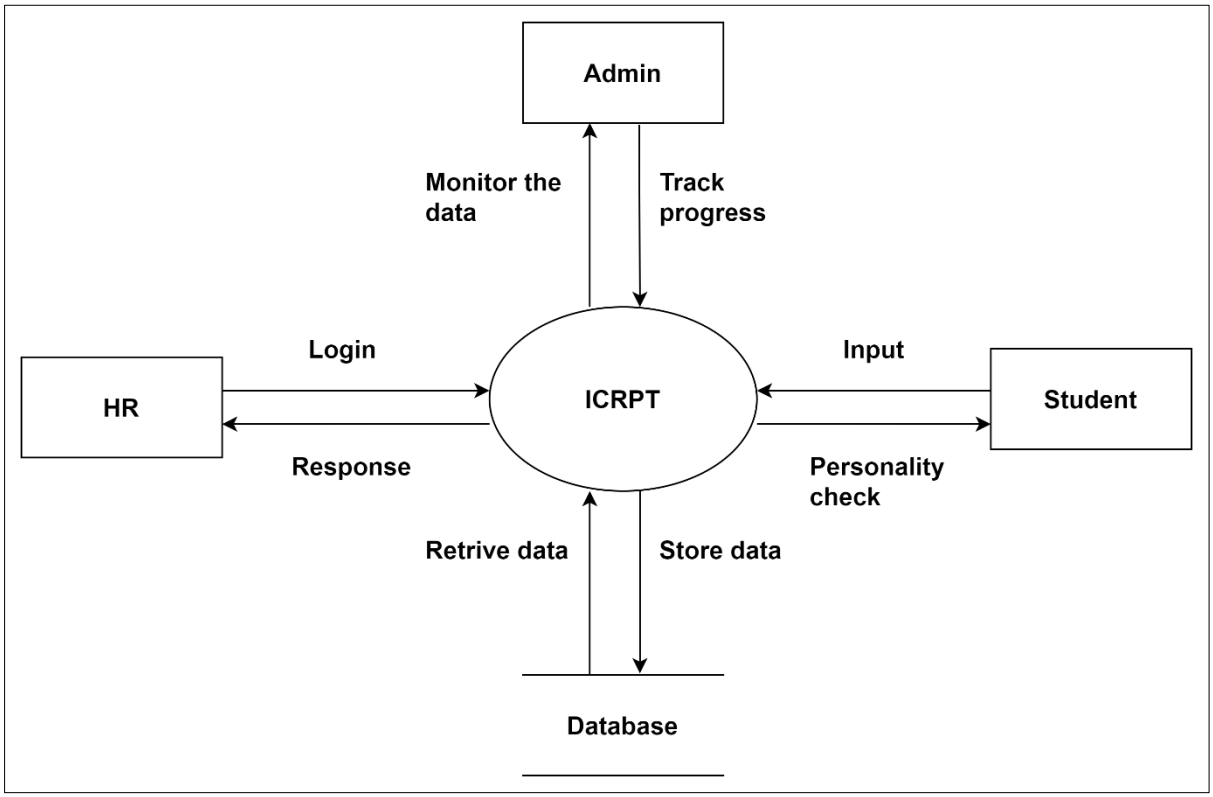

Fig.1.4 DFD Level-1

iii. DFD (Level-2)

- **Student Module**

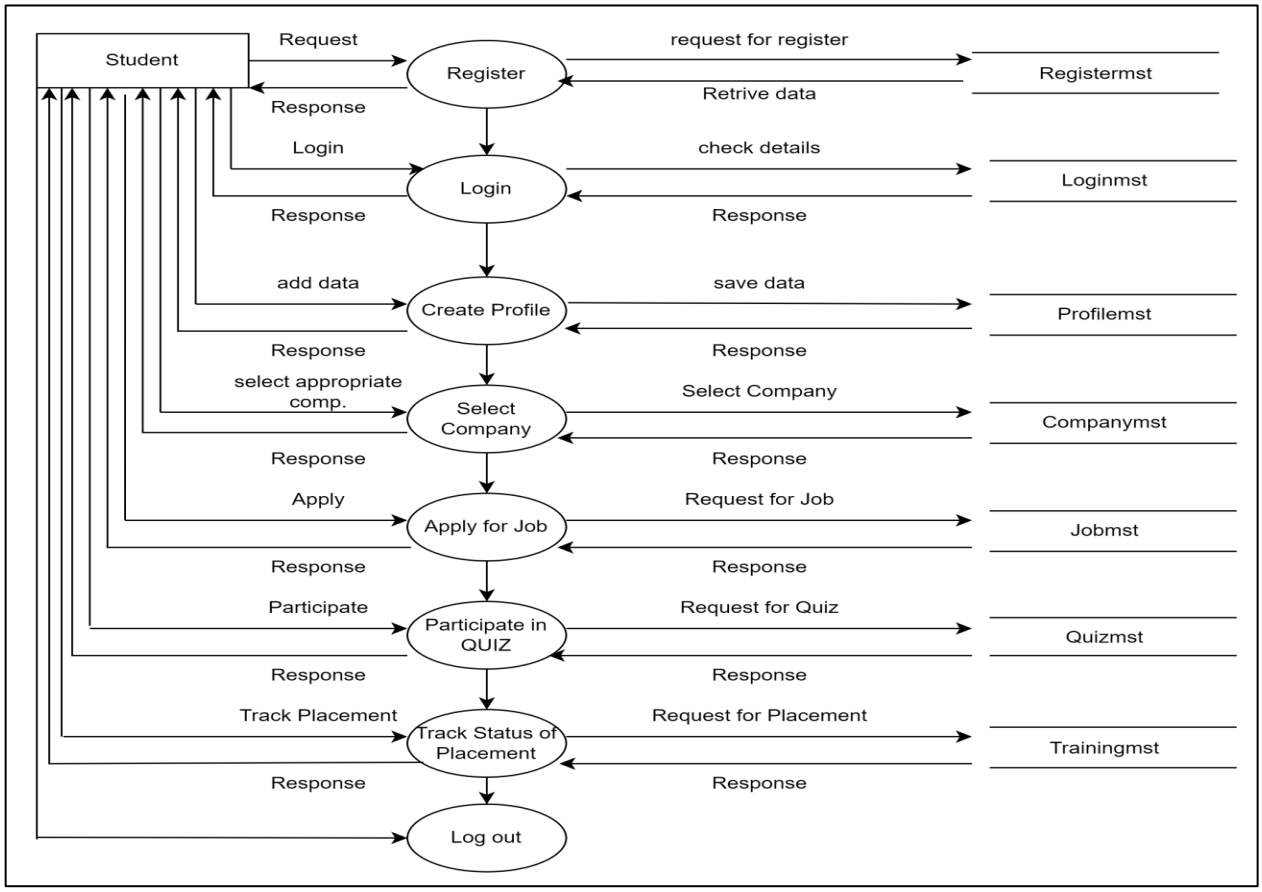

Fig 1.5 Student

- **Admin (TPO) Module:**

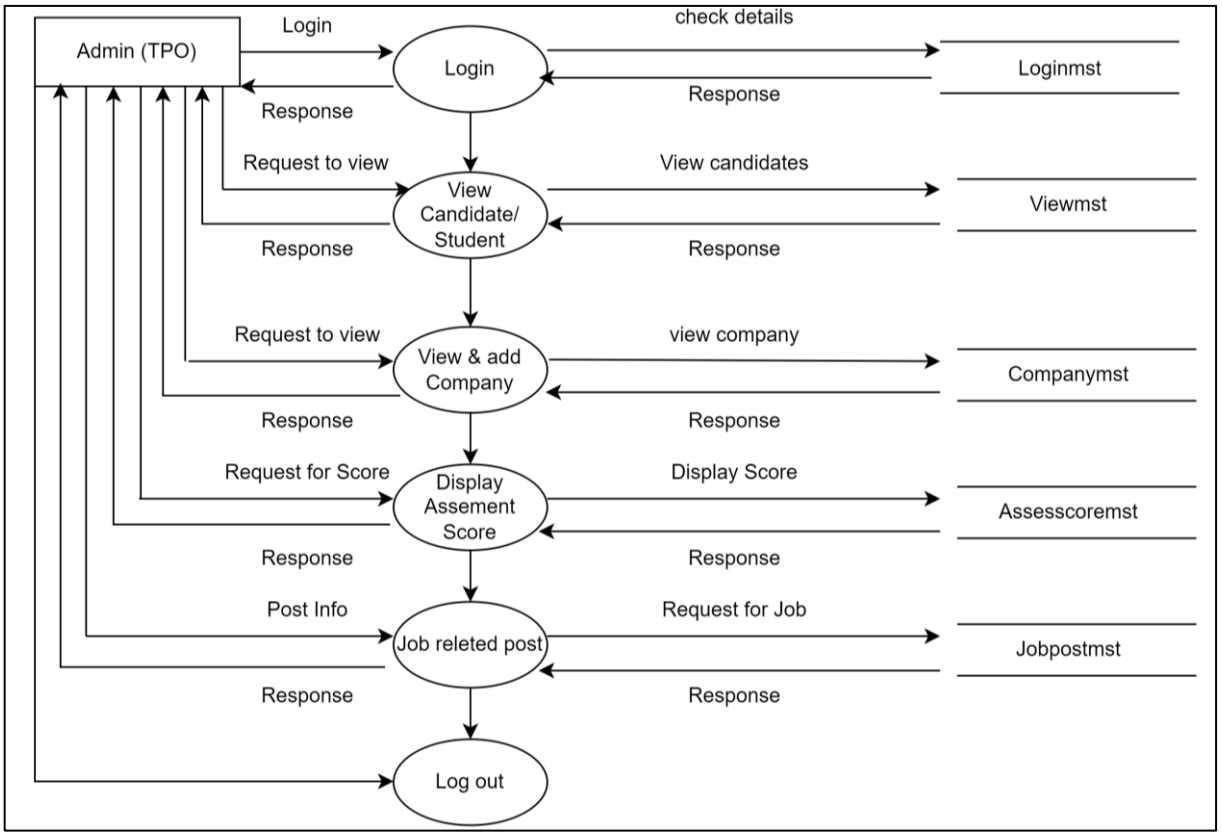

Fig 1.6 Admin

- **HR Module:**

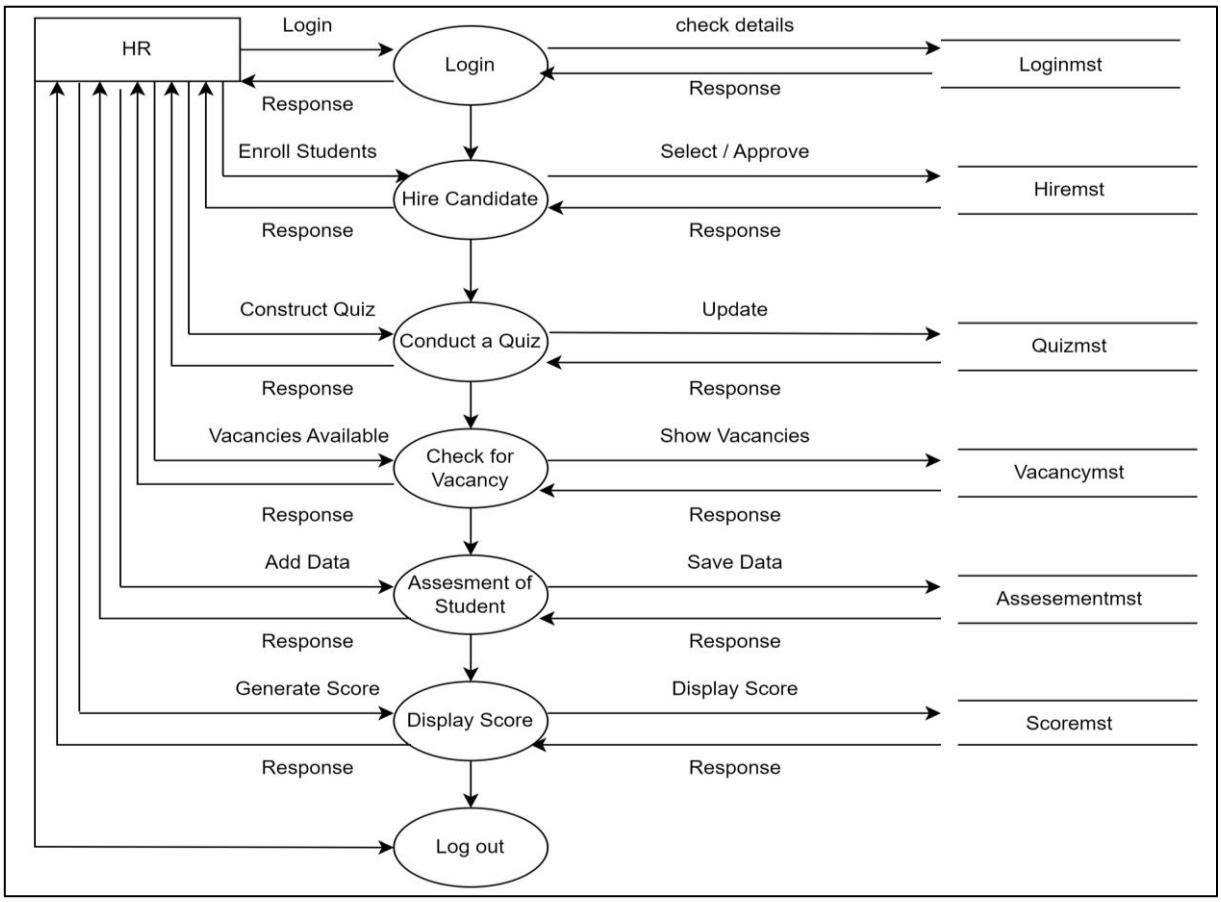

Fig. 1.7 HR

A Data Flow Diagram (DFD) is a graphical representation of how data flows within a system. It's commonly used in software engineering and business analysis to model processes and data flow within an organization or a specific system. In the context of a Training and Placement System, the DFD diagram can illustrate how data moves within the system to manage the training and placement processes.

4) Use Case Diagram:

Student:

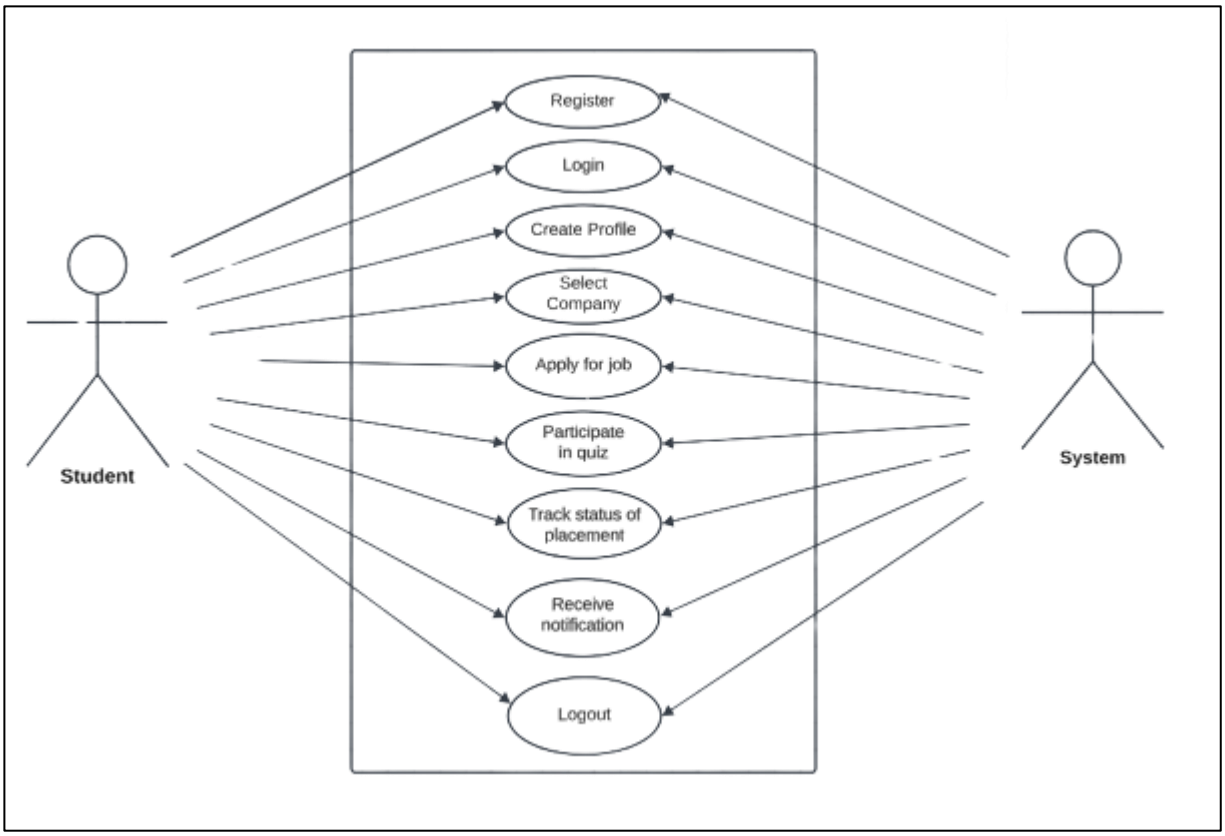

Fig. 1.8 Student

A use case diagram is a visual representation of the interactions and relationships between different actors and the system or application they use. In this case, we are creating a use case diagram for a student module that encompasses various functionalities such as registration, login, creating a profile, selecting a company, applying for a job, participating in a quiz, tracking placement status, receiving notifications, and logging out. Let's break down the elements of this use case diagram:

Student: The primary actor who interacts with the student module.
System: Represents the student module itself.

Use Cases:

1. **Register:** This use case allows a student to register for the system. It involves providing personal information and creating an account.
2. **Login:** Students can log in to the system using their registered credentials.
3. **Create Profile:** After logging in, students can create their profiles by providing additional details such as their educational background, skills, and interests.
4. **Select Company:** This use case enables students to browse and select companies they are interested in. It might involve viewing company profiles and job listings.
5. **Apply for Job:** After selecting a company, students can apply for job positions that match their qualifications and interests.
6. **Participate in Quiz:** Some companies may require students to take quizzes or assessments as part of the application process. This use case represents the student's participation in such quizzes.
7. **Track Placement Status:** Students can check the status of their job applications and track the progress of their placement. This may include seeing whether their application is pending, approved, or rejected.

8. **Receive Notification:** The system can send notifications to students regarding application status changes, new job opportunities, or other relevant information.
9. **Logout:** When students want to end their session, they can log out of the system to protect their account and information.

Admin (TPO):

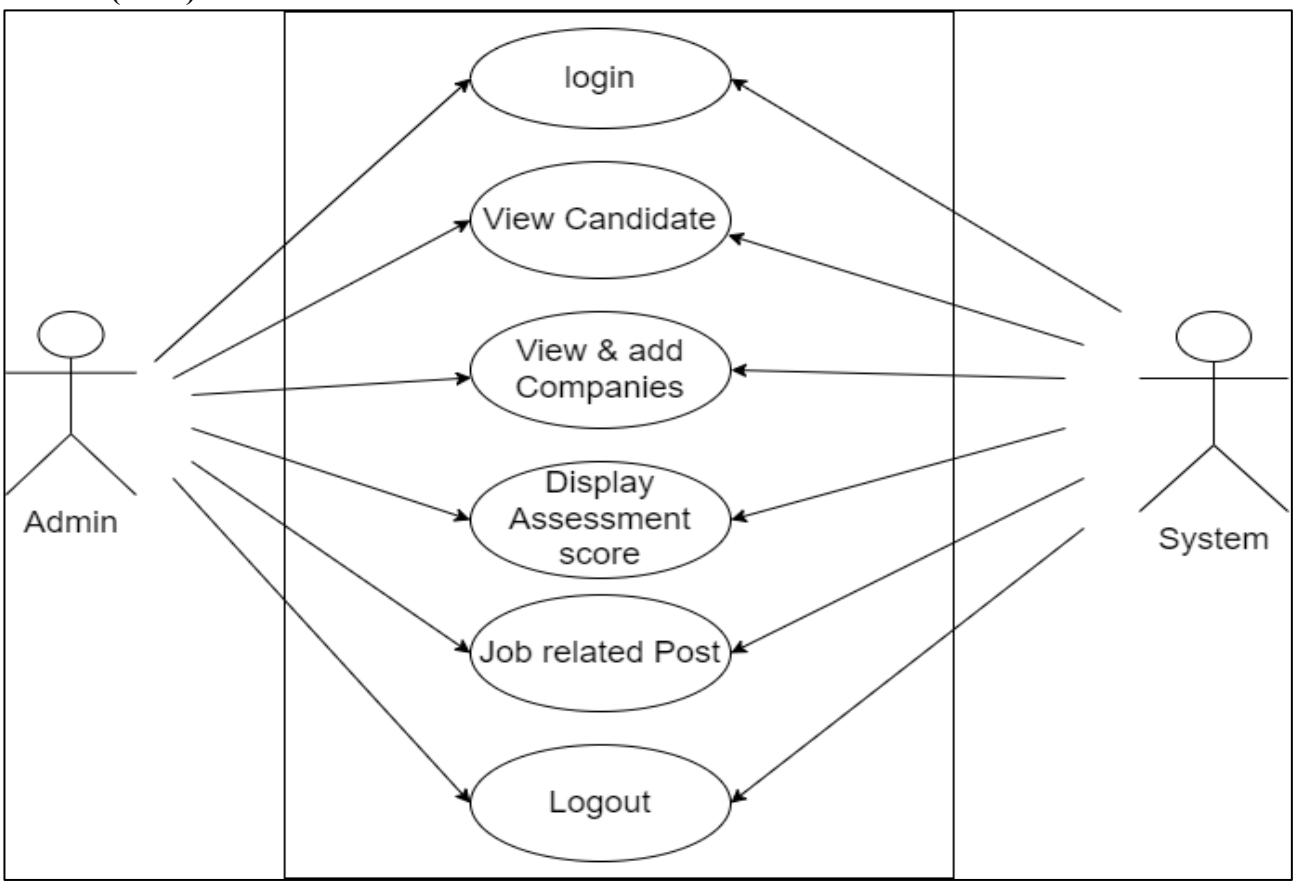

Fig. 1.9 Admin(TPO)

In this case, we are creating a use case diagram for an admin module that encompasses various functionalities such as registration, login, viewing candidates, viewing and adding companies, displaying assessment scores, making job-related posts, and logging out. Let's break down the elements of this use case diagram:

Admin: The primary actor who interacts with the admin module.
System: Represents the admin module itself.

Use Cases:

1. **Register:** This use case allows an admin to register for the system. It involves providing personal information and creating an admin account.
2. **Login:** Admins can log in to the system using their registered credentials.
3. **View Candidate:** Admins can view candidate profiles, which may include details such as educational background, skills, and application status.
4. **View and Add Companies:** Admins have the ability to view existing company profiles and add new companies to the system.
5. **Display Assessment Score:** Admins can view assessment scores for candidates who have taken quizzes or assessments.
6. **Job-related Post:** Admins can make job-related posts, such as job openings or announcements, which can be viewed by candidates.

7. **Logout:** When admins want to end their session, they can log out of the system to protect their account and information.

HR:

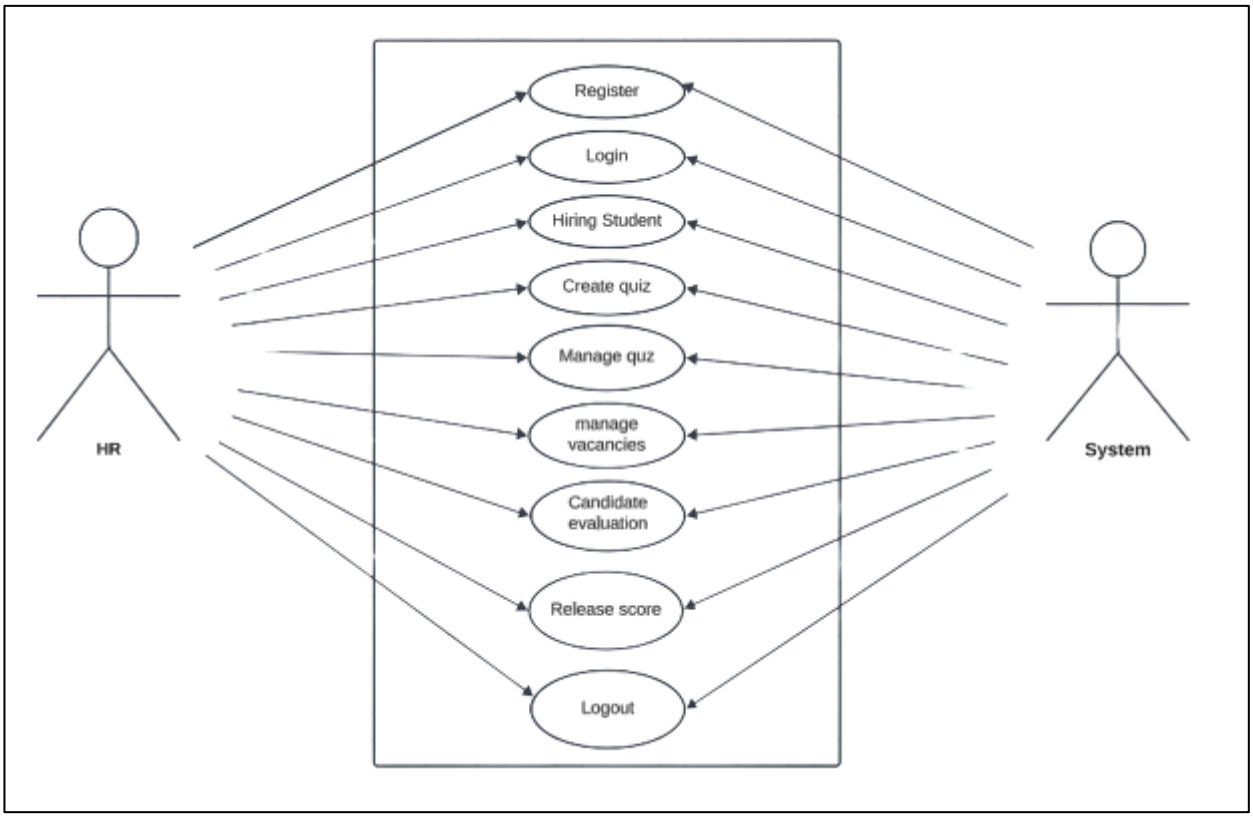

Fig. 1.10 HR

In this case, we are creating a use case diagram for an HR module that encompasses various functionalities such as registration, login, hiring students, creating quizzes, managing quizzes, managing vacancies, evaluating candidates, releasing scores, and logging out. Let's break down the elements of this use case diagram:

HR: The primary actor who interacts with the HR module.
System: Represents the HR module itself.

Use Cases:

1. **Register:** This use case allows an HR representative to register for the system. It involves providing personal information and creating an HR account.
2. **Login:** HR representatives can log in to the system using their registered credentials.
3. **Hiring Students:** HR representatives can initiate the process of hiring students by reviewing applications, conducting interviews, and making job offers.
4. **Create Quiz:** HR representatives have the ability to create quizzes or assessments for candidates to take as part of the hiring process.
5. **Manage Quiz**: HR representatives can manage and update existing quizzes, including adding, editing, or removing questions and answers.
6. **Manage Vacancies:** HR representatives can manage job vacancies by posting new job openings, updating existing ones, or closing positions when they are filled.
7. **Candidates Evaluation:** HR representatives can evaluate candidates based on their applications, interviews, assessment scores, and other relevant criteria.
8. **Release Score:** This use case allows HR representatives to release the assessment scores of candidates to inform them about their performance.

9. **Logout:** When HR representatives want to end their session, they can log out of the system to protect their account and information.

5) Class Diagram:

The class diagram represents the structure of an Integrated Campus Recruitment, Placement, and Training (ICRPT) system. This system appears to be a comprehensive platform for managing various aspects of campus recruitment, including student and employer interactions, job postings, assessments, and administrative functions.

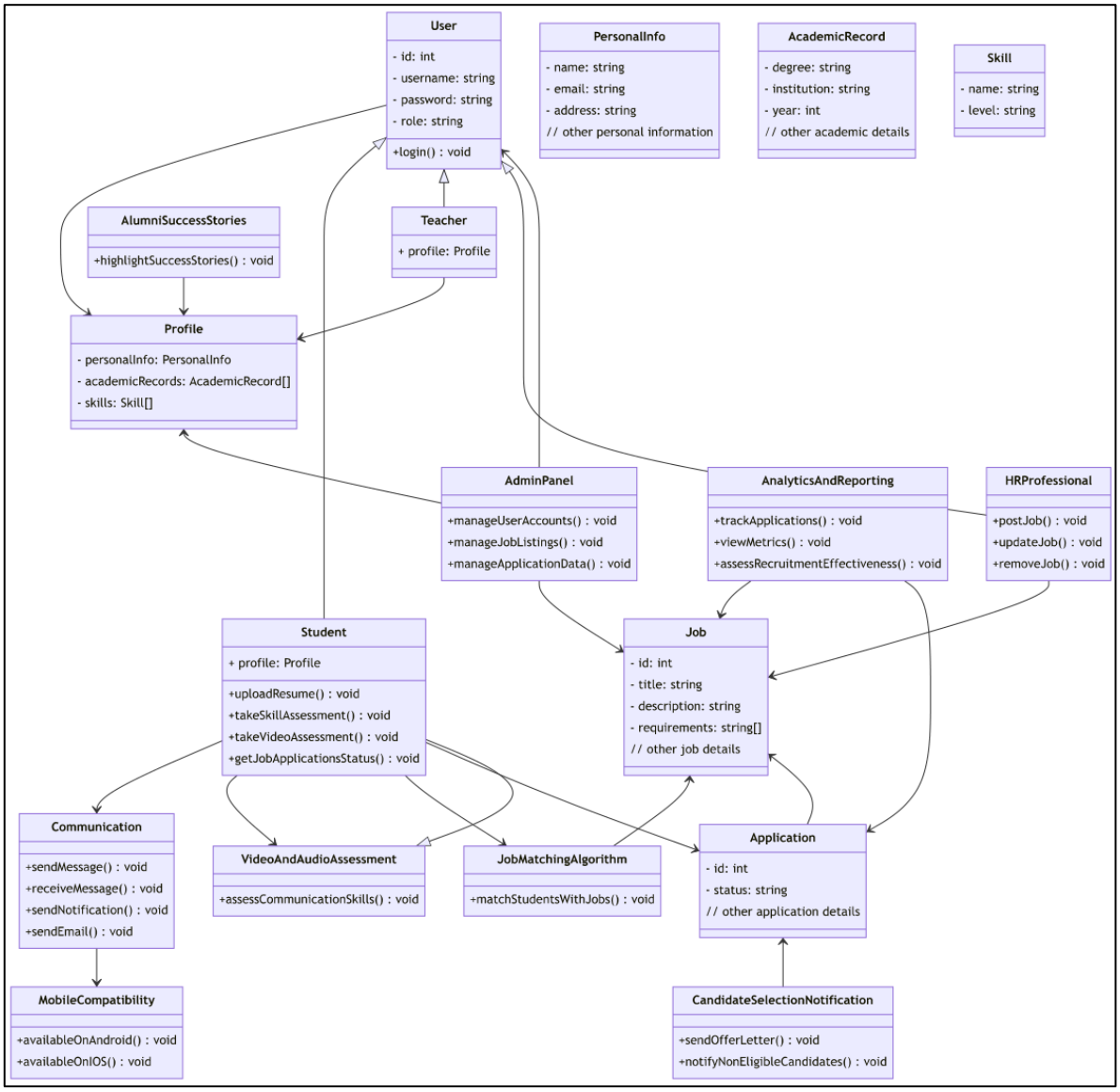

Fig.1.11 Class Diagram

A detailed explanation of each class and its associated methods and attributes:

1. User:

Attributes:

id (int): Unique identifier for each user.

username (string): User's login username.

password (string): User's login password.

role (string): Defines the role of the user (Student, Teacher, HRProfessional, etc.).

Methods:

login(): A method that allows users to log into the system.

2. Student:

Inherits from User.

Attributes:

profile: An instance of the `Profile` class, which contains personal, academic, and skill-

related information.
Methods:
uploadResume(): Allows students to upload their resumes.
takeSkillAssessment(): Enables students to take skill assessment tests.
takeVideoAssessment(): Allows students to take video assessments.
getJobApplicationsStatus(): Retrieves the status of job applications.

3. Teacher:
Inherits from User.
Attributes:
profile: An instance of the `Profile` class, which contains personal, academic, and skill-related information.

4. HRProfessional:
Inherits from User.
Methods:
postJob(): Allows HR professionals to post job listings.
updateJob(): Enables HR professionals to update job listings.
removeJob(): Allows HR professionals to remove job listings.

5. Profile:
Attributes:
personalInfo: An instance of the `PersonalInfo` class, containing personal information.
academicRecords: An array of `AcademicRecord` instances representing academic achievements.
skills: An array of `Skill` instances representing the user's skills.

6. PersonalInfo:
Attributes:
name (string): User's name.
email (string): User's email address.
address (string): User's physical address and contact information.

7. AcademicRecord:
Attributes:
degree (string): Academic degree achieved.
institution` (string): Educational institution where the degree was earned.
Year (int): Year of graduation.

8. Skill:
Attributes:
name (string): The name of the skill.
level (string): The user's proficiency level in the skill.

9. Communication:
Methods:
sendMessage(): Allows users to send messages.
receiveMessage(): Allows users to receive messages.
sendNotification(): Enables users to send notifications.
sendEmail(): Allows users to send emails.

10. AdminPanel:
Methods:
manageUserAccounts(): Manages user accounts, including creation, modification, and deletion.

manageJobListings(): Manages job listings, including posting, updating, and removal.
manageApplicationData(): Manages application data, including processing and updating applications.

11. MobileCompatibility:

Methods:
availableOnAndroid(): Indicates compatibility with Android devices.
availableOnIOS(): Indicates compatibility with iOS devices.

12. VideoAndAudioAssessment:

Methods:
assessCommunicationSkills(): Conducts assessments of communication skills using video and audio.

13. Job:

Attributes:
id (int): Unique identifier for each job.
title (string): Job title.
description (string): Job description.
requirements (string[]): An array of requirements for the job posting.

14. JobMatchingAlgorithm:

Methods:
matchStudentsWithJobs(): Implements an algorithm to match students with suitable job listings.

15. AnalyticsAndReporting:

Methods:
trackApplications(): Tracks and manages job applications.
viewMetrics(): Provides metrics related to job placements.
assessRecruitmentEffectiveness(): Assesses the effectiveness of the recruitment process.

16. Application:

Attributes:
Id (int): Unique identifier for each application.
status (string): Represents the status of a job application.

17. AlumniSuccessStories:

Methods:
highlightSuccessStories(): Showcases alumni success stories.

18. CandidateSelectionNotification:

Methods:
sendOfferLetter(): Sends offer letters to selected candidates.
notifyNonEligibleCandidates(): Notifies candidates who were not selected.

The class diagram provides a comprehensive view of the classes, their attributes, and methods, showcasing how the various components of the ICRPT system interact and facilitate the recruitment and placement process on campus. Each class plays a specific role in the system's functionality, from user management to job postings, assessments, and reporting.

6) Activity Diagram:

In the "Integrated Campus Recruitment, Placement, and Training Analysis" (ICRPT) system, three distinct modules — HR, Student, and Teacher—are responsible for managing various aspects of campus recruitment and placement. The activity diagrams for each module provide a comprehensive view of their operations.

i. Student Module

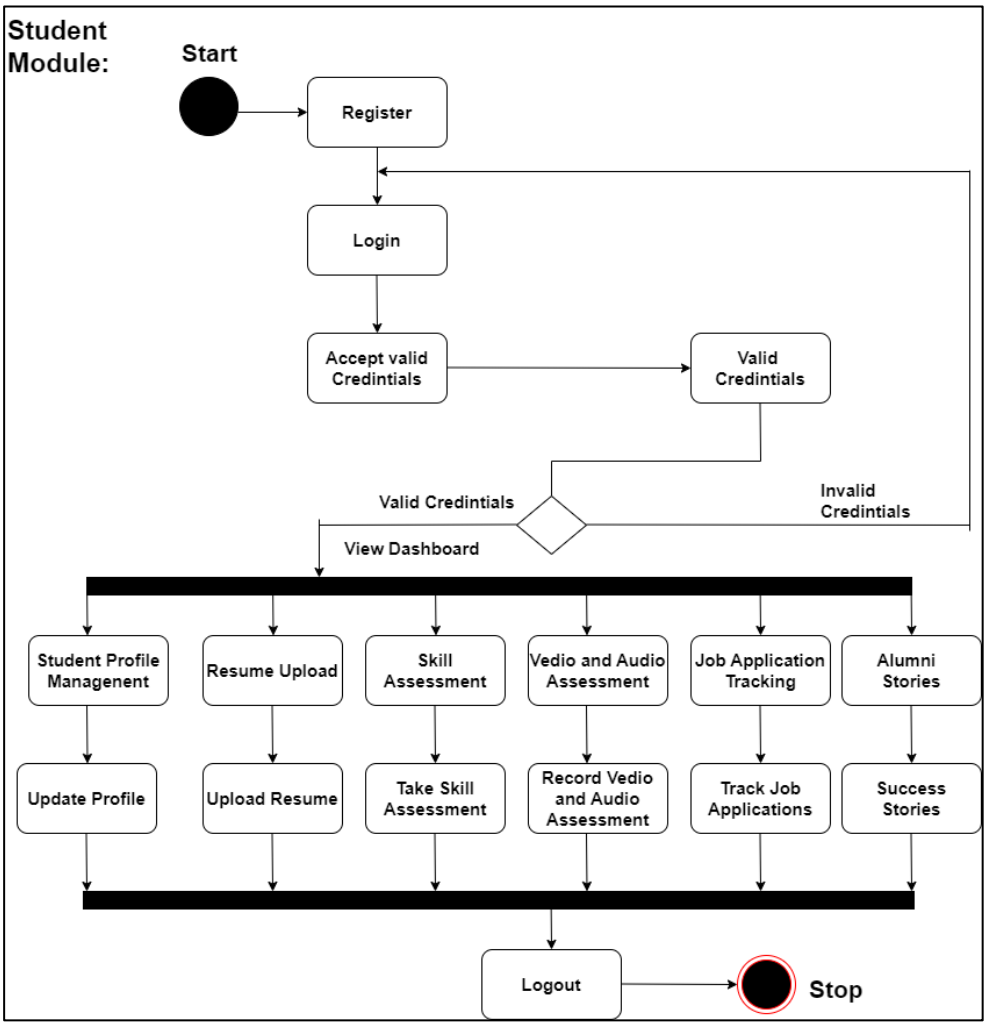

Fig.1.12 Activity- Student

The Student Module focuses on students who are seeking placement opportunities. After a secure login and validation process, students can access their dashboard, update their profiles, upload resumes, communicate with HR professionals, and participate in various assessments and tracking of job applications. If the login is valid, students can access functionalities such as profile management, communication, and skill assessment. In the case of an invalid login, an error message is displayed. This module empowers students to actively engage with job opportunities, showcase their skills, and track their applications efficiently.

ii. HR Module

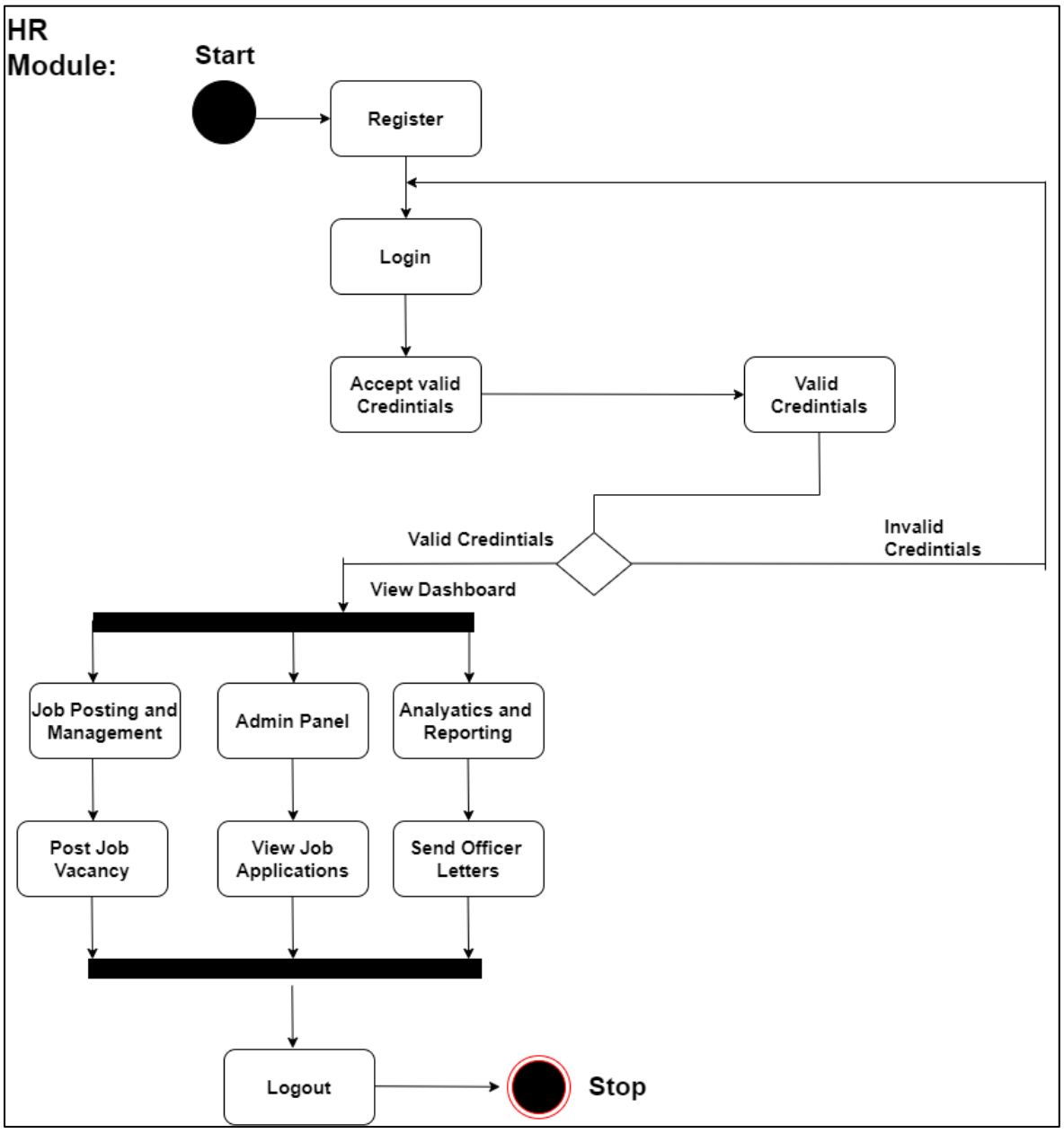

Fig.1.13 Activity- HR

The HR Module is designed to cater to the needs of human resources professionals. It begins with a secure login process, where user authentication and authorization are essential to ensure data integrity. Once logged in, HR professionals can access the dashboard, post job vacancies, view job applications, and send offer letters. If the user login is valid, they can also access functionalities such as job posting and management, admin panel, and analytics and reporting. However, in the case of an invalid login, the system displays an error message. The HR Module plays a central role in managing job postings and interactions with potential candidates.

iii. TPO Module:

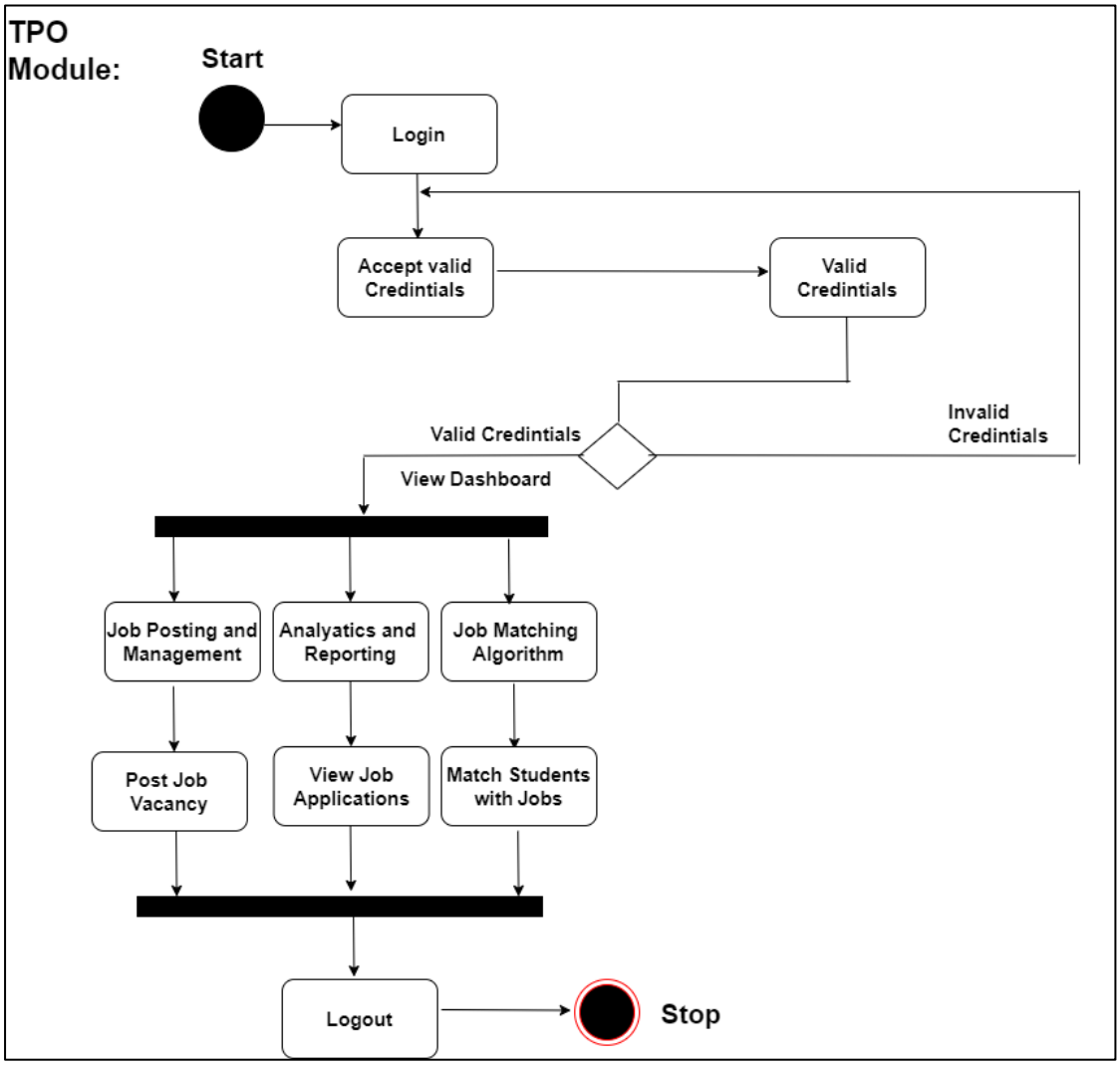

Fig.1.14 Activity- Admin

The Teacher Module is tailored for the training and placement officers (TPOs) who facilitate the job matching process. Similar to the other modules, it begins with a secure login and validation. Upon successful login, TPOs can access the dashboard, post job vacancies, view job applications, match students with job opportunities, and review analytics and reports. The decision node for valid login leads to functionalities like job posting and management, analytics and reporting, and the job matching algorithm. For invalid logins, an error message is displayed. This module is instrumental in connecting students with suitable job openings and analyzing the effectiveness of recruitment processes.

7) Deployment Diagram:

The deployment diagram shows how the ICRPT system's components work together to provide a comprehensive solution for integrated campus recruitment, placement, and training analysis. It starts with users accessing the system through client devices, progresses through load balancing, web servers, and data processing on the application server. The system interacts with database servers, cloud services, and various other components to deliver a secure, reliable, and data-driven platform for both Students and Employers. The email server, security infrastructure, and integration with university systems add layers of functionality to meet the needs of all system stakeholders.

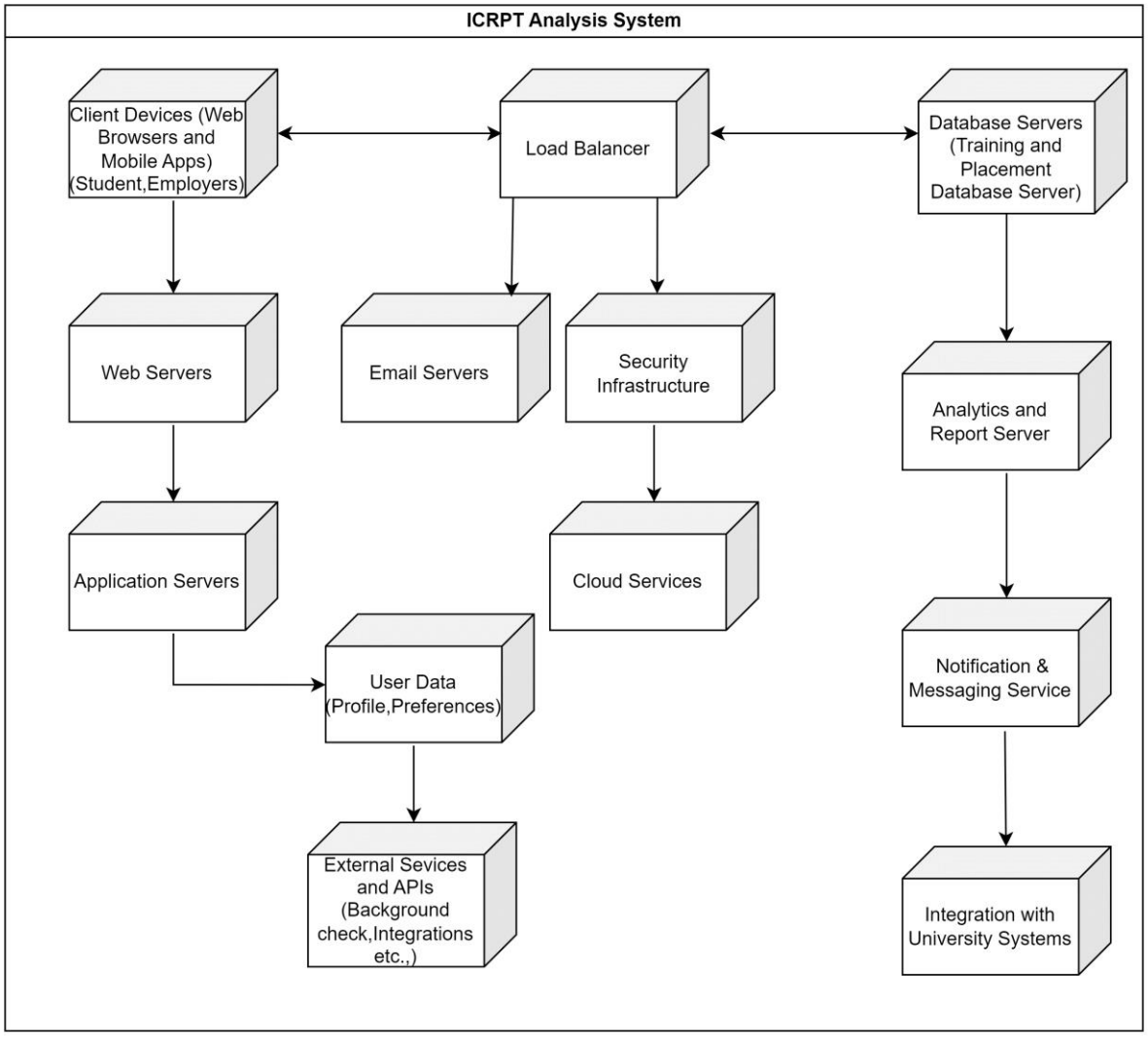

Fig.1.15 Deployment of ICRPT

i. Client Devices (Students, Employers):

Users, including Students and Employers, initiate the process by accessing the ICRPT system through web browsers on their computers or mobile apps on their mobile devices. These clients interact with the system to search for jobs, apply to positions, and review training and placement opportunities.

ii. Load Balancer:

The Load Balancer ensures that incoming user requests are efficiently distributed across multiple Web Servers.

Balances the traffic to prevent overload on any specific server.

Enhances scalability and provides high availability for the system.

iii. Web Servers:

Web Servers serve as the user interface and initial point of contact for clients.
Handle HTTP requests and responses for clients.
Render and display web pages with job listings, training details, and user profiles.
Facilitate user interactions such as submitting job applications and accessing training content.

iv. Application Server:

The Application Server manages the core business logic and functionality of the ICRPT system..
Handles user requests, processes data, and manages application behavior.
Communicates with the Database Servers to retrieve and store data.
Supports integration with external services and cloud resources.

v. User Data (Profiles, Preferences):

User data, including user profiles and preferences, is stored on the Application Server.
Stores user profiles, contact information, and preferences.
Personalizes the user experience by displaying relevant job listings and training opportunities.
Enables users to review their application history and preferences.

vi. Database Servers:

Database Servers store and manage the system's data.
Main Database Server:
Stores student profiles, employer profiles, job listings, and training records.
Manages data retrieval and storage.
Training and Placement Database Server:
Houses data related to training and placement outcomes.
Stores detailed analysis and reports for decision-making.

vii. Email Server:

The Email Server handles email notifications and communication between the ICRPT system and its users.
Sends email notifications, including interview invitations, application updates, and training reminders.
Facilitates communication between Students, Employers, and the system.

viii. Security Infrastructure:

The Security Infrastructure provides network security, intrusion detection, and access control.
Ensures data security, privacy, and compliance with data protection regulations.
Safeguards user data and system integrity from potential threats.

ix. Cloud Services:

Cloud Services may be utilized for data storage, backups, and additional processing capacity.
Enhance system reliability and scalability by leveraging cloud resources.
Support data redundancy and disaster recovery measures.

x. Analytics & Report Server:

The Analytics & Report Server processes data to generate reports and analytics.
Stores historical data for trend analysis and performance evaluation.
Provides valuable insights to both Students and Employers for informed decision-making.

xi. Notification & Messaging Service:

The Notification & Messaging Service manages real-time communication and notifications.
Sends alerts, reminders, and updates to users, enhancing user engagement.
Facilitates timely communication within the system.

xii. Integration with University Systems:

The ICRPT system interfaces with university systems to access academic data.
Integrates with educational institution systems to retrieve student profiles, academic records, and course enrollment data.
Ensures seamless interaction between the ICRPT system and the university's infrastructure for comprehensive analysis and decision-making.

The ICRPT deployment diagram illustrates the step-by-step process of how the software components work together to deliver a comprehensive solution for integrated campus recruitment, placement, and training analysis. It showcases the interactions between users, load balancing, web and application servers, databases, security, cloud services, analytics, and university systems to meet the needs of Students and Employers effectively.