Feature-rich, Practical Online Application for the Training and Placement Dept.
Software Requirements Specification

Version 1.0

Submitted in Partial Fulfillment for the Award of Degree of Bachelor of Technology in InformationTechnology from Rajasthan Technical University, Kota

MENTOR:

Mrs. Shalini Singhal

(Dept. of Information Technology)

COORDINATOR:

Dr. Richa Rawal

(Dept. of Information Technology)

SUBMITTED BY:

Vishwas Vijay Vargiya (21ESKIT120)

Prateek Sharma (21ESKIT311)

Vivek Garg (21ESKIT121)

DEPARTMENT OF INFORMATION TECHNOLOGY

SWAMI KESHWANAND INSTITUTE OF TECHNOLOGY, MANAGEMENT & GRAMOTHAN

Ramnagaria (Jagatpura), Jaipur – 302017

SESSION 2024-25

Revision History

Date	Version	Description	Author
30/11/24	1.0	Contains basic requirements for this project.	Vishwas Vijay Vargiya

Table of Contents

1	**Introduction**	1
	1.1 Purpose	1
	1.2 Scope	1
	1.3 Definitions, Acronyms and Abbreviations	1
	1.4 References	1
	1.5 Technologies to be used	2
	1.6 Overview	2
2	**Literature survey**	2
	2.1 Review of Related Work	2
	2.2 Knowledge gaps	3
	2.3 Comparative Analysis	3
	2.4 Summary	3
3	**Specific Requirements**	4
	3.1 Functional Requirements	4
	3.2 Non- Functional Requirements	5
	3.3 Hardware Requirements	5
	3.4 Software Requirements	5
	3.5 Agile Methodology	5
	3.6 Business Process Model	6
	3.7 Supplementary Requirements	6
4	**System Architecture**	6
	4.1 Client-Server Architecture	6
	4.2 Communications Interfaces	8
5	**Overall Description**	9
	5.1 Product Features	9
	5.2 Data flow diagram (DFD-0, DFD-1, DFD-2)	10
	5.3 E-R Diagram	12
	5.4 Class Diagram	13
	5.5 Use-Case Model Survey	14
	5.6 Behaviors Diagram	15
	5.7 Assumptions and Dependencies	17
6	**Supporting Information**	18
7	**Conclusion & Future Scope**	19
8	**Concerns / Queries / Doubts if any:**	21

1 Introduction

This **Software Requirements Specification (SRS)** document outlines the detailed requirements for the development of the "Online Centralized Platform for the Training and Placement (TP) Department." This platform is designed to address the inefficiencies in the current placement management system by providing a feature-rich and user-friendly digital solution. It aims to streamline the placement process by offering a centralized platform for students, Training and Placement Officers (TPOs), and recruiters to interact effectively.

1.1 Purpose

The primary focus is to simplify the management of student data, facilitate seamless communication, and improve the overall experience of the placement process for students and recruiters, while reducing administrative overhead for TPOs.

1.2 Scope

The primary purpose of this Software Requirement Specification (SRS) document is to comprehensively outline the requirements, external behavior, and operational constraints of the "Online Centralized Platform for the Training and Placement (TP) Department." This document ensures that the system aligns with the objectives of streamlining placement activities, enhancing communication, and providing a secure and efficient platform for students, Training and Placement Officers (TPOs), and recruiters.

The document describes functional and non-functional requirements, ensuring clarity on features such as student profile management, CV uploads, real-time notifications, placement tracking, and recruiter access. It also specifies usability factors, design constraints, and the underlying technologies necessary for the development of a robust and scalable solution.

1.3 Definitions, Acronyms, and Abbreviations

- SRS**:** Software Requirements Specification
- TP**:** Training and Placement
- TPO**:** Training and Placement Officer
- UI**:** User Interface
- UX**:** User Experience
- CV**:** Curriculum Vitae
- Firebase**:** Google’s Backend-as-a-Service platform used for authentication & database.

1.4 References

- Agile Software Development Practices Guide, 2020, Scrum Alliance

- Android Developer Documentation - Official Guide (https://developer.android.com/)
- Firebase Documentation - Google Cloud Platform (https://firebase.google.com/docs/)
- User Experience Design Fundamentals - Interaction Design Foundation, 2021
- IEEE Standards for Software Requirements Specifications, IEEE Std 830-1998
- Research papers on Centralized Placement Systems and Cloud-based Solutions for Educational Institutions

1.5 Technologies to be Used

- **Frontend:** Android (Java) for mobile app development
- **Backend:** Firebase for authentication, real-time database management, and file storage.
- **Security:** Firebase Authentication for secure login processes,
- **Security**: OAuth 2.0, SSL/TLS Encryption
- **APIs**: RESTful APIs for smooth communication

1.6 Overview

This SRS document is organized into sections that progressively detail the platform's requirements:

- **Section 2:** Overall Description – System objectives, interactions, and dependencies.
- **Section 3:** Specific Requirements – Functional and non-functional requirements.
- **Section 4:** System Models – Use case diagrams, sequence diagrams, and data flow representations.
- **Section 5:** Appendices – Additional details, glossaries, and references for better understanding.

2 Literature Survey

The **Literature Survey** section examines existing systems, research studies, and methodologies relevant to the management of placement activities in educational institutions. It reviews current online platforms designed to facilitate the interaction between students, Training and Placement Officers (TPOs), and recruiters, highlighting their strengths and limitations.

2.1 Review of Related Work

1) **Placement Management Systems:**

- Allow students to apply for jobs but lack real-time tracking and communication features.
- No personalized job recommendations or candidate filtering based on specific skills.

2) **Corporate Recruiting Platforms (e.g., LinkedIn):**

- Focus on job postings and candidate searches but lack integration with educational institutions.
- Do not provide tools for managing recruitment drives or interview scheduling.

3) **Job Portals (e.g., Naukri, Glassdoor):**

- Primarily designed for general job searches and not tailored for academic institutions.
- Do not include features like placement tracking, CV management, or real-time updates.

4) **ERP Systems for Educational Institutions:**

- Manage student data but lack specialized tools for collaboration between students, TPOs, and recruiters.
- User interfaces are often not optimized for efficient placement management.

2.2 Knowledge Gaps

1) **Lack of Comprehensive Platforms:**
   - Existing platforms focus on individual aspects like job applications or resume uploads but fail to integrate functionalities such as placement tracking, real-time communication, and personalized job recommendations.
2) **Limited Matchmaking Mechanisms:**
   - Current systems do not provide advanced matchmaking mechanisms for aligning students with the right recruiters based on specific skills, qualifications, and interests.
3) **Absence of Corporate-Centric Features:**
   - There is a lack of tools for corporates to evaluate student compatibility based on specific criteria like job roles, skills, and qualifications.
4) **Minimal Focus on Startups' Needs:**
   - Many platforms do not fully cater to students' needs by offering features such as job alerts, personalized guidance, or streamlined interview scheduling.

2.3 Comparative Analysis

Feature	AngelList	LinkedIn	CSR Platforms	Proposed Platform
Startup-Corporate Matching	✗	✗	✗	✓
CSR Funding Opportunities	✗	✗	✓	✓
Mentorship Opportunities	✓	✗	✗	✓
B2B Sales and Partnerships	✗	✗	✗	✓
Region and Domain Filtering	✗	✗	✗	✓
Ease of Use and Scalability	✓	✓	✗	✓

2.4 Summary

- Existing systems offer limited solutions for managing placement-related activities, with

separate tools for job applications, student profiles, and recruiter interactions, but lack integration and real-time updates.

- The proposed **Online Centralized Platform for the TP Department** addresses these gaps by providing a unified system for student profile management, CV uploads, placement tracking, and real-time communication with recruiters.
- By offering streamlined processes, real-time updates, and a user-friendly interface, the platform enhances placement efficiency and establishes itself as a comprehensive solution for managing the placement process effectively.

3 Specific Requirements

This section outlines the functional, non-functional, hardware, and software requirements necessary for the Online Centralized Platform for the TP Department. It also includes the methodology, business process model, and supplementary requirements to guide the design, development, and testing of the system.

3.1 Functional Requirements

The core functional requirements include:

1) **User Registration and Login:**
   - Students, TPOs must register and log in securely using email, phone, or social media credentials.
   - Admin panel for managing user registrations and approvals.
2) **Profile Management:**
   - Students: Add and update personal details, academic qualifications, and upload CVs.
   - TPOs: Manage student data, track placement progress, and schedule interviews.
3) **Placement Matching Algorithm:**
   - Suggest relevant job postings to students based on qualifications, interests, and skills.
   - Suggest suitable candidates to recruiters based on their job requirements and student profiles.
4) **Search and Filter Options:**
   - Search for students and recruiters and filter results by skills, location, and job preferences.
5) **Messaging and Collaboration:**
   - Secure messaging feature to allow students, TPOs, and recruiters to communicate and share updates or job details.
6) **Job Posting and Application Management:**
   - Recruiters can create job listings and manage applications, while students can apply for positions and track their application status.
7) **Notifications and Updates:**
   - Notify users about job openings, application status updates, interview schedules, and placement drive reminders.
8) **Admin Dashboard:**
   - Admins can manage user profiles, monitor system activity, and generate reports to analyze placement trends and system performance.

3.2 Non-Functional Requirements

1) **Performance Requirements:**
   - The system must handle 1,000 concurrent users with response times under 2 seconds.
2) **Scalability:**
   - Platform should support future scaling for an increasing user base.
3) **Security:**
   - Implement OAuth 2.0 authentication and SSL/TLS encryption for secure transactions.
4) **Usability:**
   - Provide an intuitive UI/UX for all user types.
5) **Availability:**
   - Ensure 99.9% uptime with minimal maintenance downtime.
6) **Data Integrity:**
   - Ensure accurate and consistent data storage and retrieval.

3.3 Hardware Requirements

1) **Server-Side Hardware:**
   - Processor: Quad-Core Intel Xeon or equivalent
   - RAM: Minimum 16 GB
   - Storage: Minimum 500 GB SSD
   - Network: High-speed internet connection (1 Gbps)
2) **Client-Side Hardware:**
   - Device: Desktop, Laptop, Tablet, or Smartphone
   - Browser: Latest version of Chrome, Firefox, or Safari
   - OS: Windows 10/11, Android

3.4 Software Requirements

1) **Server-Side Software:**
   - OS: Linux (Ubuntu 22.04 or CentOS)
   - Database: Firebase
   - Cloud Platform: AWS or Azure
2) **Client-Side Software:**
   - Frontend Framework: Android (Java)
   - Browser Compatibility: Support for modern web browsers
3) **Development Tools:**
   - Version Control: Git and GitHub
   - IDE: Android Studio

3.5 Agile Methodology

1) **Scrum Framework:**
   - **Sprints:** 2-week development cycles.
   - **Roles:** Product Owner, Scrum Master, Development Team.
   - **Artifacts:** Product Backlog, Sprint Backlog, Increment.

2) **User Stories:**
   - Example: “As a student, I want to receive notifications for upcoming placement drives."
3) **Daily Standups:**
   - Regular meetings to ensure progress, address challenges, and align goals.
4) **Continuous Integration/Continuous Deployment (CI/CD):**
   - Automated testing and deployment pipeline for rapid iteration.

3.6 Business Process Model

1) **High-Level Workflow:**
   - **Step 1** Students, TPOs, and recruiters register and set up profiles.
   - **Step 2:** Matchmaking algorithm connects relevant students with recruiters based on skills, qualifications, and preferences.
   - **Step 3:** Recruiters review student profiles, shortlist candidates, and schedule interviews.
   - **Step 4:** Students apply for jobs, attend interviews, and track their application progress.
   - **Step 5:** Admin monitors platform activity, manages user access, and ensures compliance with platform policies.
2) **Key Processes:**
   - Job application and interview scheduling process.
   - Placement tracking from application to final placement.
   - Real-time updates and notifications for students and recruiters.

3.7 Supplementary Requirements

1) **Regulatory Compliance:**
   - Adherence to data protection laws such as GDPR and CCPA to ensure user privacy and data security.
2) **Audit Logs:**
   - Maintain logs of user activity for security, performance analysis, and troubleshooting.
3) **Localization:**
   - Support multiple languages to accommodate users from diverse regions and backgrounds.
4) **Help and Support:**
   - Provide a help center and customer support options for users requiring technical assistance.

4 System Architecture

The system architecture of the **Online Centralized Platform for the TP Department** is designed to be scalable, secure, and modular. It aims to ensure high performance and ease of maintenance. This section outlines the interaction between key components, including clients, servers, databases, and external APIs.

4.1 Client-Server Architecture

The system follows a **three-tier client-server architecture**:

1) Client Tier (Frontend):

- Users (students, TPOs) interact with the system through a mobile app interface, developed using **Android (Java**).
- Key responsibilities:
  - Displaying data retrieved from the server.
  - Sending user requests to the server through RESTful APIs.
  - Validating user inputs before sending them to the backend.

2) Server Tier (Backend):

- The backend built using **Firebase** services (Authentication, Realtime Database, etc.), processes requests from clients, performs business logic, and manages communication with the database.
- Key responsibilities:
  - Handling client requests via RESTful APIs.
  - Implementing the placement matching algorithm and processing placement requests.
  - Managing secure communication and authentication using **OAuth 2.0**.

3) Database Tier:

- **Firebase Realtime Database** stores student profiles, job applications, placement data, and messaging content.
- Key responsibilities:
  - Storing structured and unstructured data.
  - Ensuring data integrity and fast retrieval.

Interaction Flow:

- User Action (Frontend): A student or recruiter submits a request (e.g., applying for a job or reviewing student profiles).
- API Request (Backend): The client sends the request to the server via RESTful APIs.
- Processing (Server): The server validates inputs, processes business logic (e.g., placement matching), and retrieves data from the database.
- Response (Frontend): The server sends a response back to the client, displaying the results (e.g., job applications, matched students).

Diagram:

The architecture can be represented with the following components:

- **Frontend (Client):** Mobile app interface for students, TPOs, and recruiters.
- **Backend (Server):** Application server managing business logic and user requests.
- **Database:** Firebase Realtime Database for storing and managing data.

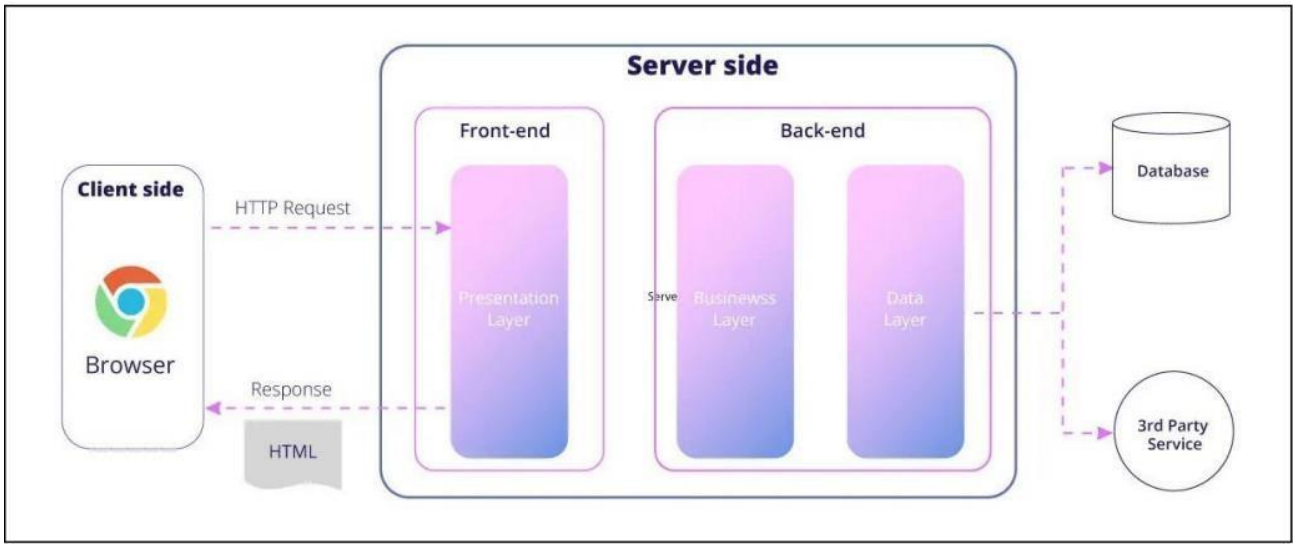

Fig-4.1 Client-server Architecture

4.2 Communication Interfaces

1) **RESTful APIs:**
   - **Purpose:** Enable seamless communication between the frontend (Android app) and backend (Firebase).
   - **Endpoints:** Examples include /login, /register, /search, /applyJob, /getMatches.
   - **Data Format:** JSON for lightweight and fast data exchange.
2) **Database Communication:**
   - **Backend to Database:**
     - Communication occurs using Firebase Realtime Database SDK, which allows the backend to retrieve and store data in real time.
     - Queries include user authentication, student profile data retrieval, job application tracking, and message storage.
   - **Database Protocols:**
     - Firebase SDK handles real-time data synchronization and updates automatically.
     - For any future changes, if a relational database like PostgreSQL is chosen, SQL-based queries will be used.

3) **Security Protocols:**
   - **Authentication:** OAuth 2.0 for secure user login, ensuring that only authorized users can access the platform.
   - **Encryption:** SSL/TLS encryption ensures the security of data during transmission between the frontend and backend.
   - **Tokens:** JSON Web Tokens (JWT) are used for session management and maintaining secure access for users.

5 Overall Description

This section provides a comprehensive background for the requirements outlined in the SRS.

It includes the product perspective, features, user characteristics, constraints, assumptions, and dependencies to offer a clear understanding of the platform and its context.

5.1 Product Features

The **Online Centralized Platform for the TP Department** is designed to streamline the placement process by providing key features that enhance the interaction between students, TPOs, and recruiters. Key features include:

1) **User Registration and Authentication:**
   - Secure registration for students, TPOs, and recruiters using email, phone, or social media credentials.
   - Role-based authentication and access control to ensure appropriate permissions for different users.
2) **Comprehensive User Profiles:**
   - **Students:** Create detailed profiles with personal information, academic qualifications, skills, and uploaded CVs.
   - **TPOs**: Manage student data, track placement activities, and schedule interviews.
3) **Advanced Matchmaking Algorithm:**
   - Matches students with suitable job openings based on skills, location, and preferences.
   - Personalized job recommendations for students and candidate suggestions for recruiters.
4) **Job Posting and Application Management:**
   - **TPOs**: Post job openings, managing applications, and tracking candidates through the hiring process.
   - **Students**: Apply for jobs, track application status, and receive updates on interviews or placements.
5) **Admin Dashboard:**
   - Manage user accounts, monitor platform activity, and generate detailed reports on placement trends, student progress, and recruiter interactions.
6) **Communication:**
   - Built-in secure messaging system for students, TPOs, and recruiters to facilitate communication and sharing of updates.
   - Notifications for new job postings, interview schedules, placement drives, and application status updates.

5.2 Data flow diagram

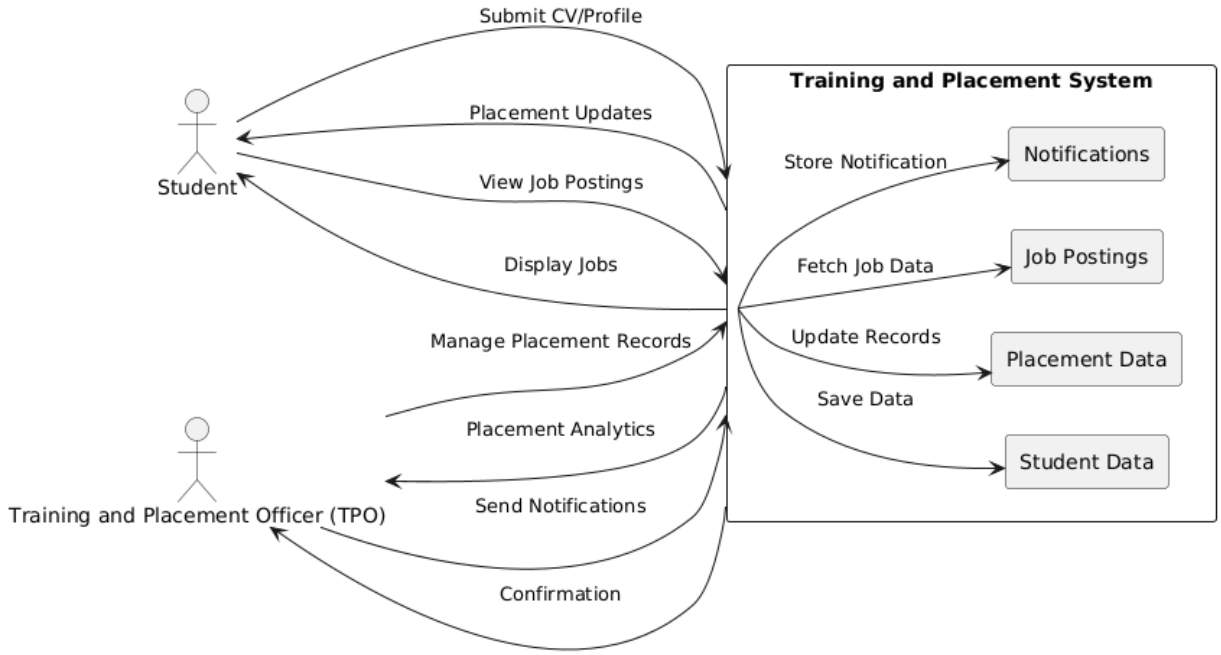

Fig-5.2.1 Data Flow Diagram level 0

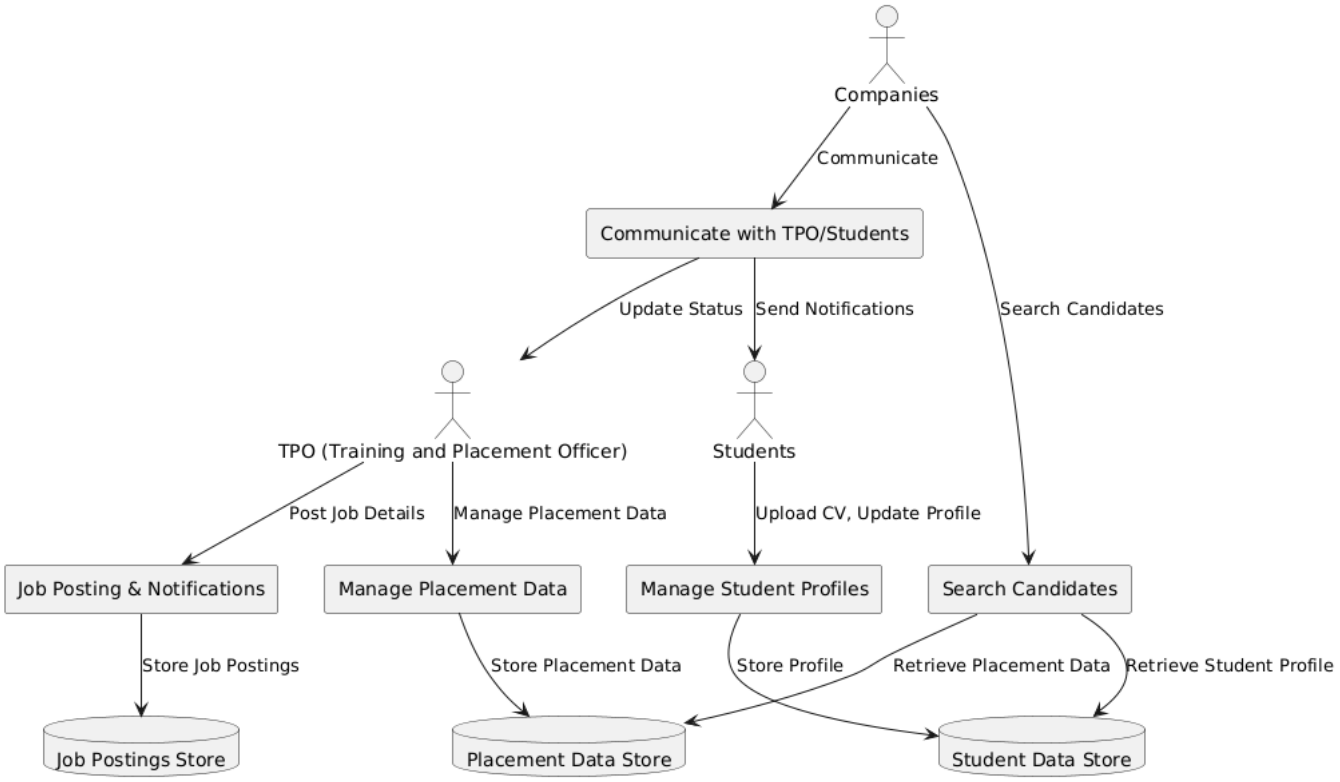

Fig 5.2.2 Data Flow Diagram level 1

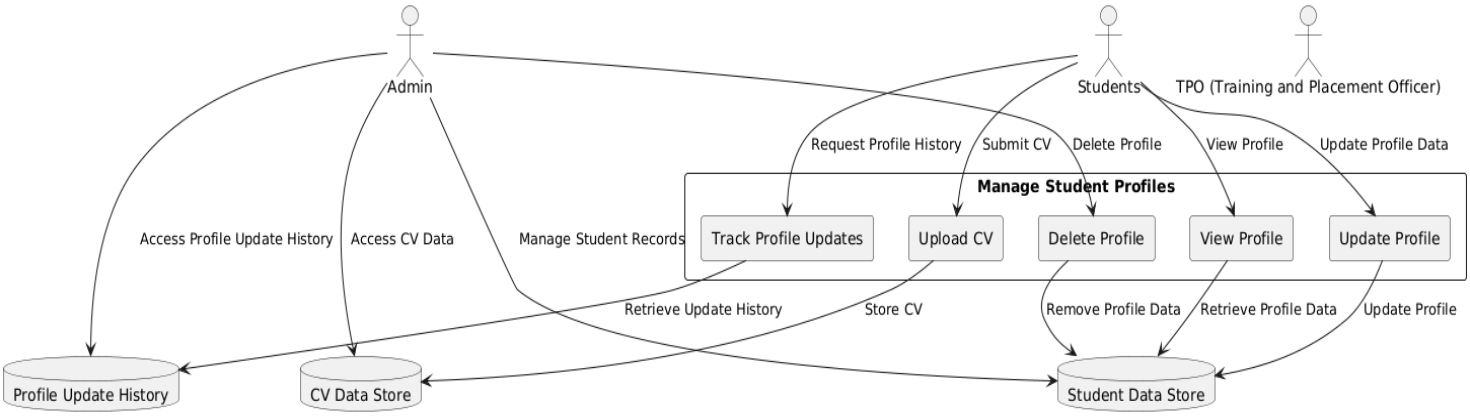

Fig 5.2.3 Data Flow Diagram level 2

5.3 E-R diagram

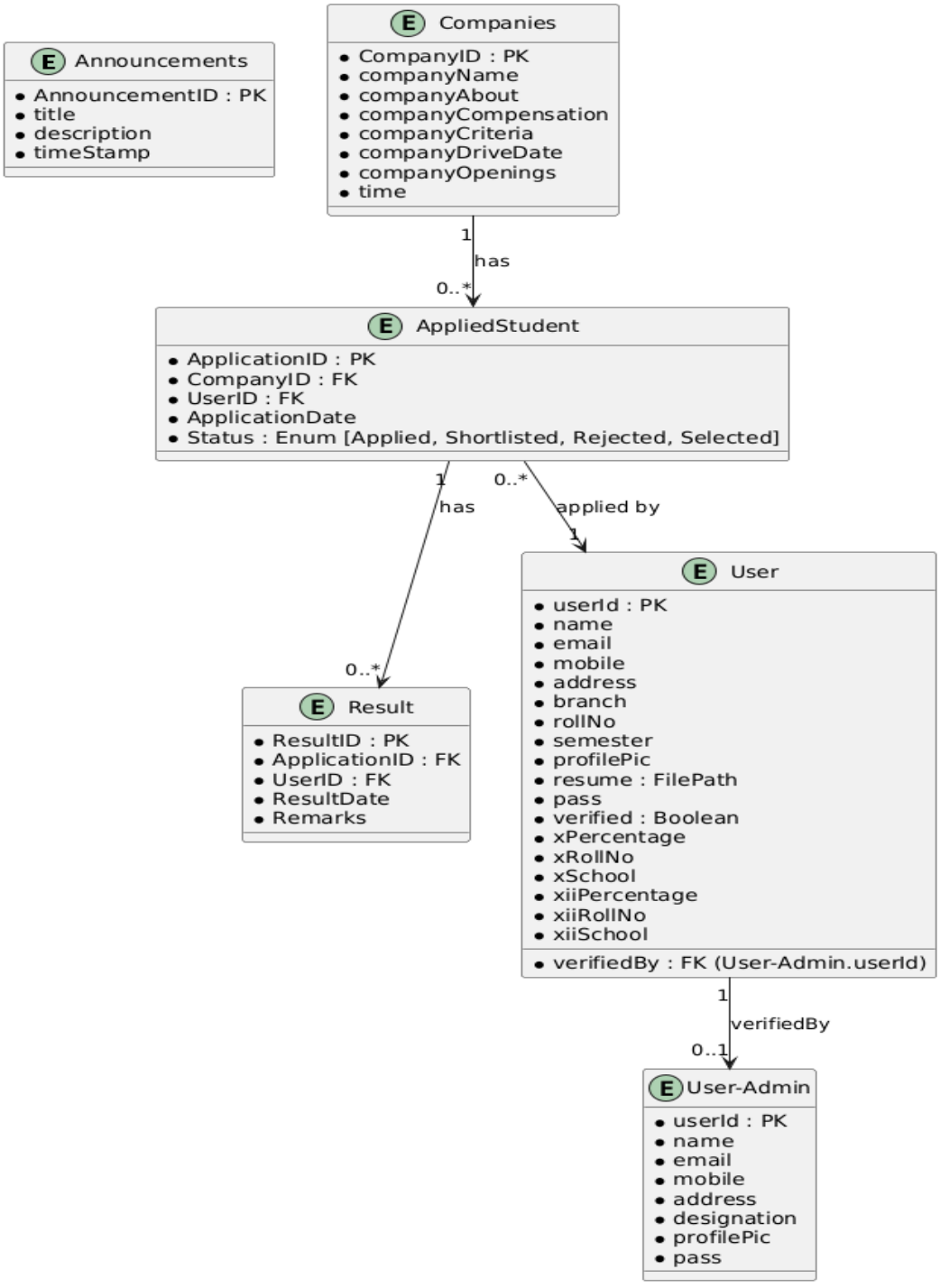

Fig 5.3 Entity Relationship Diagram

5.4 Class diagram

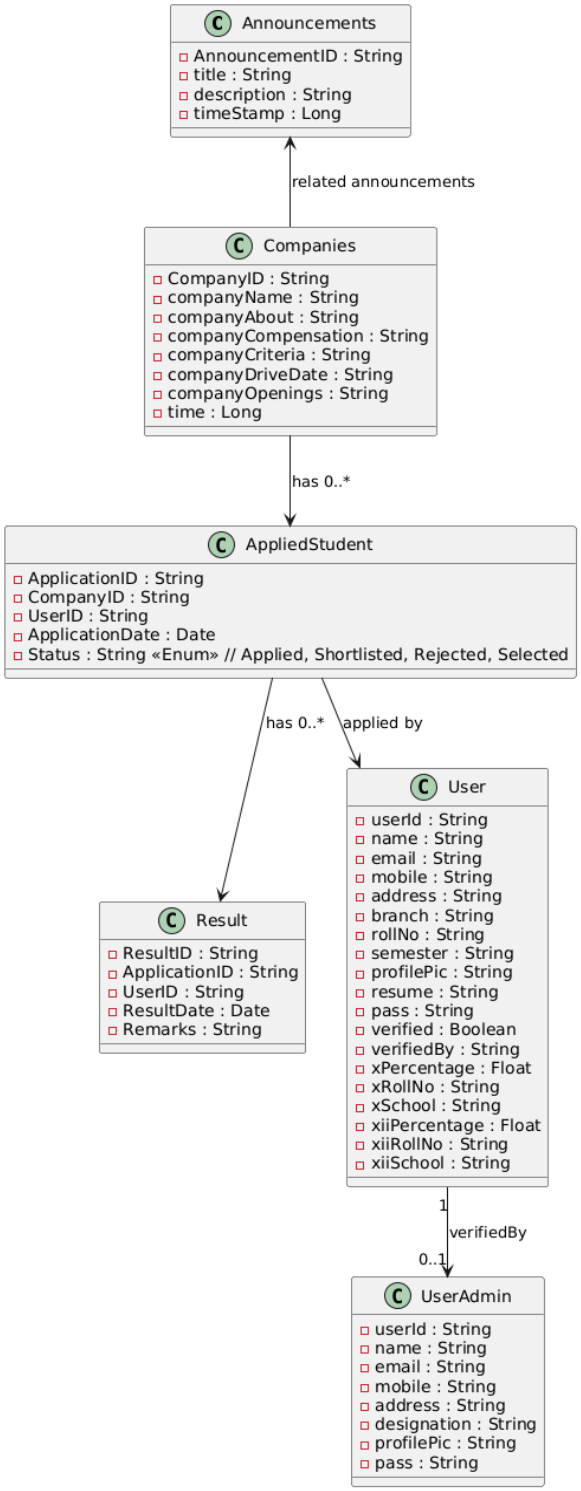

Fig 5.4 Class Diagram

5.5 Use Case Model Survey

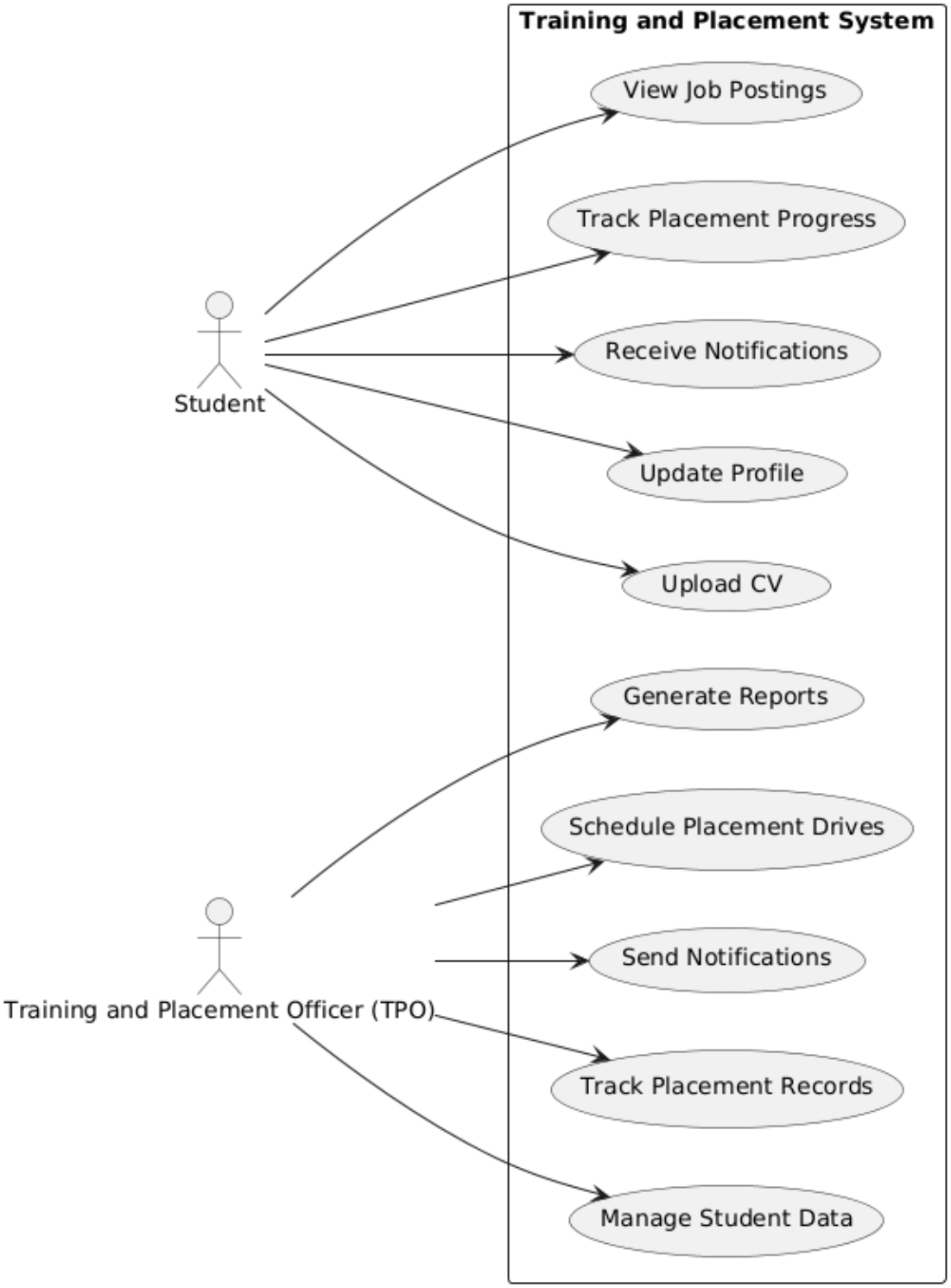

Fig 5.5 Use Case Diagram

5.6 Behaviors Diagram

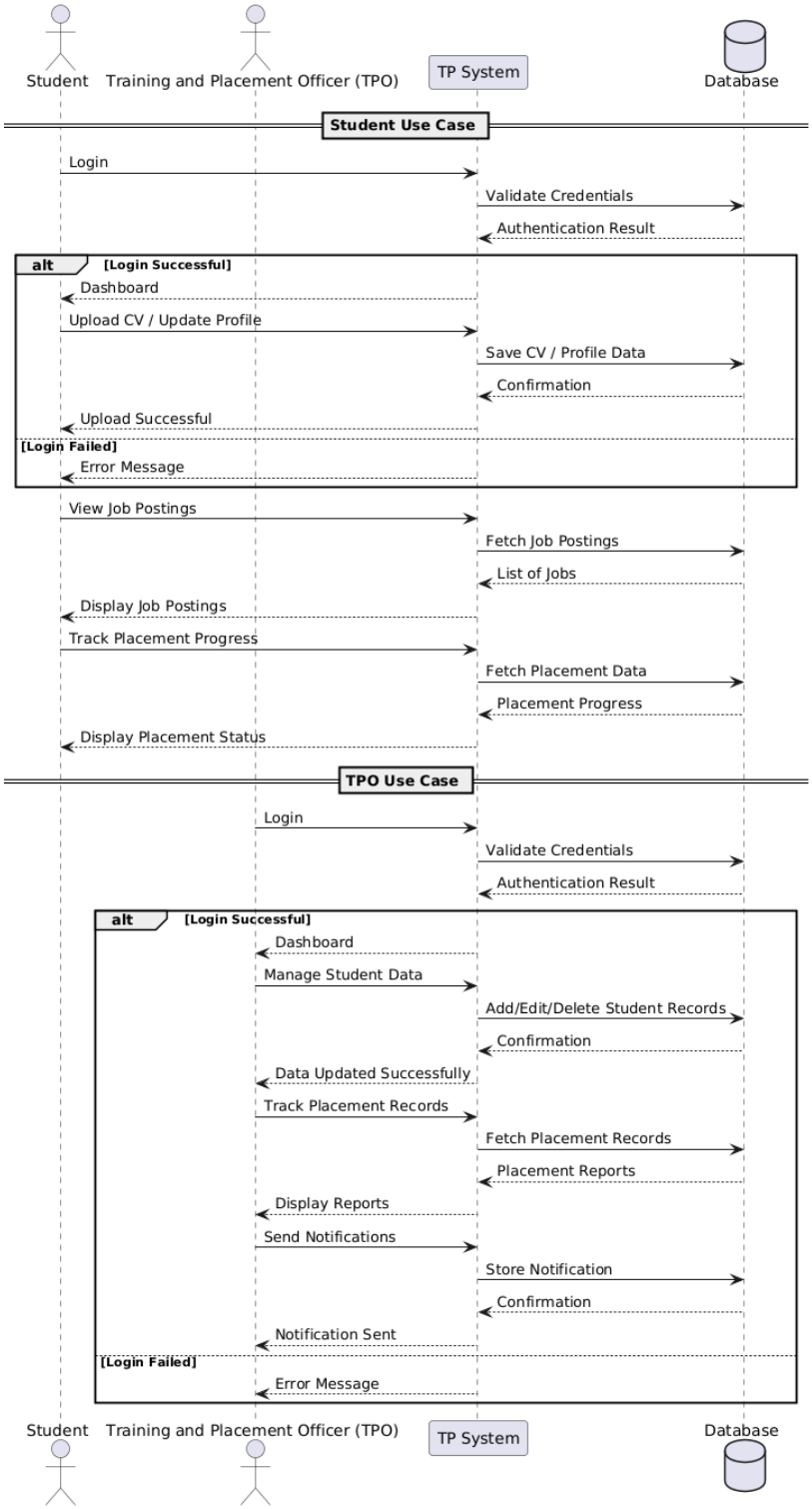

Fig 5.6.1 Sequence Diagram

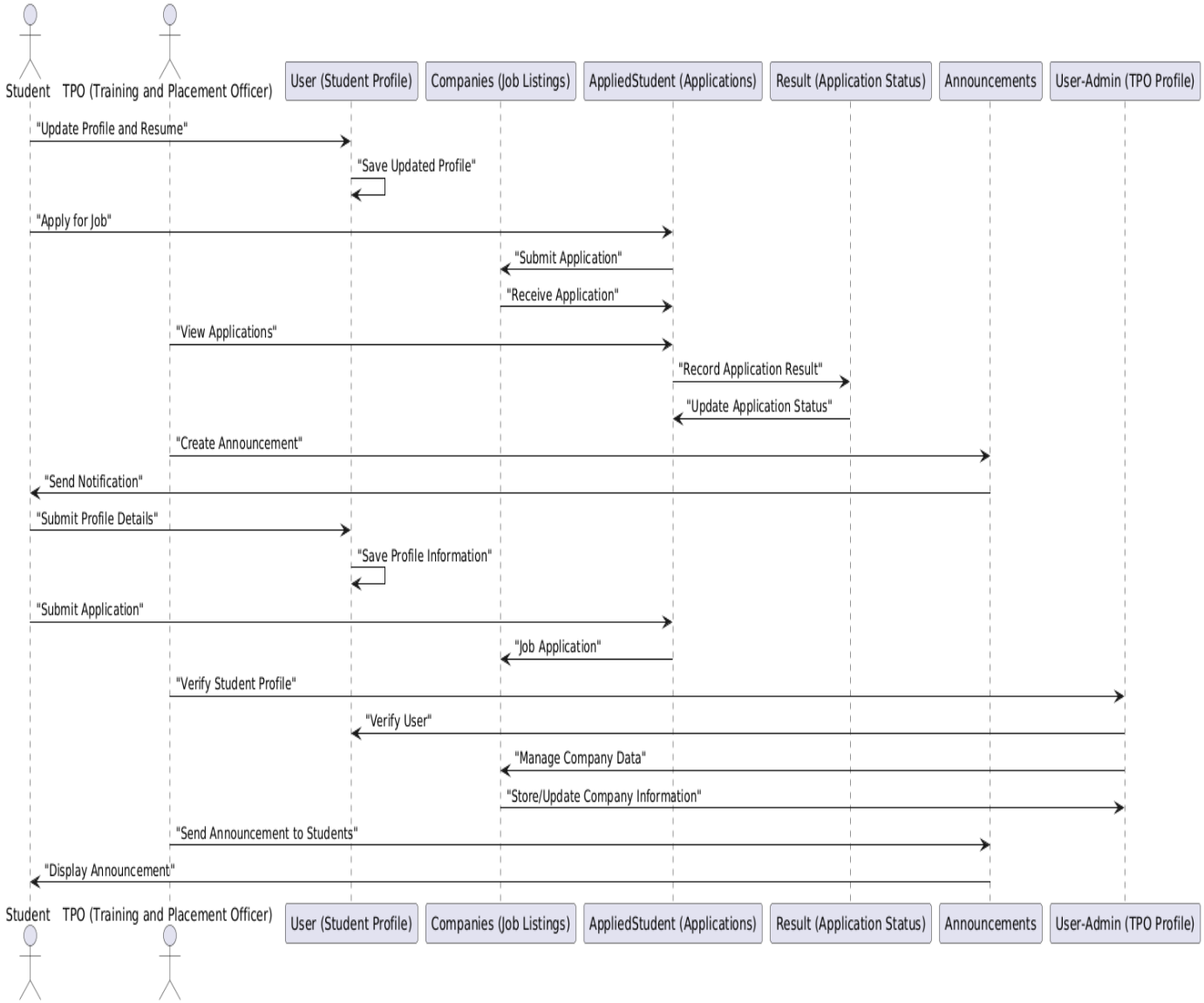

Fig 5.6.2 Communication Diagram

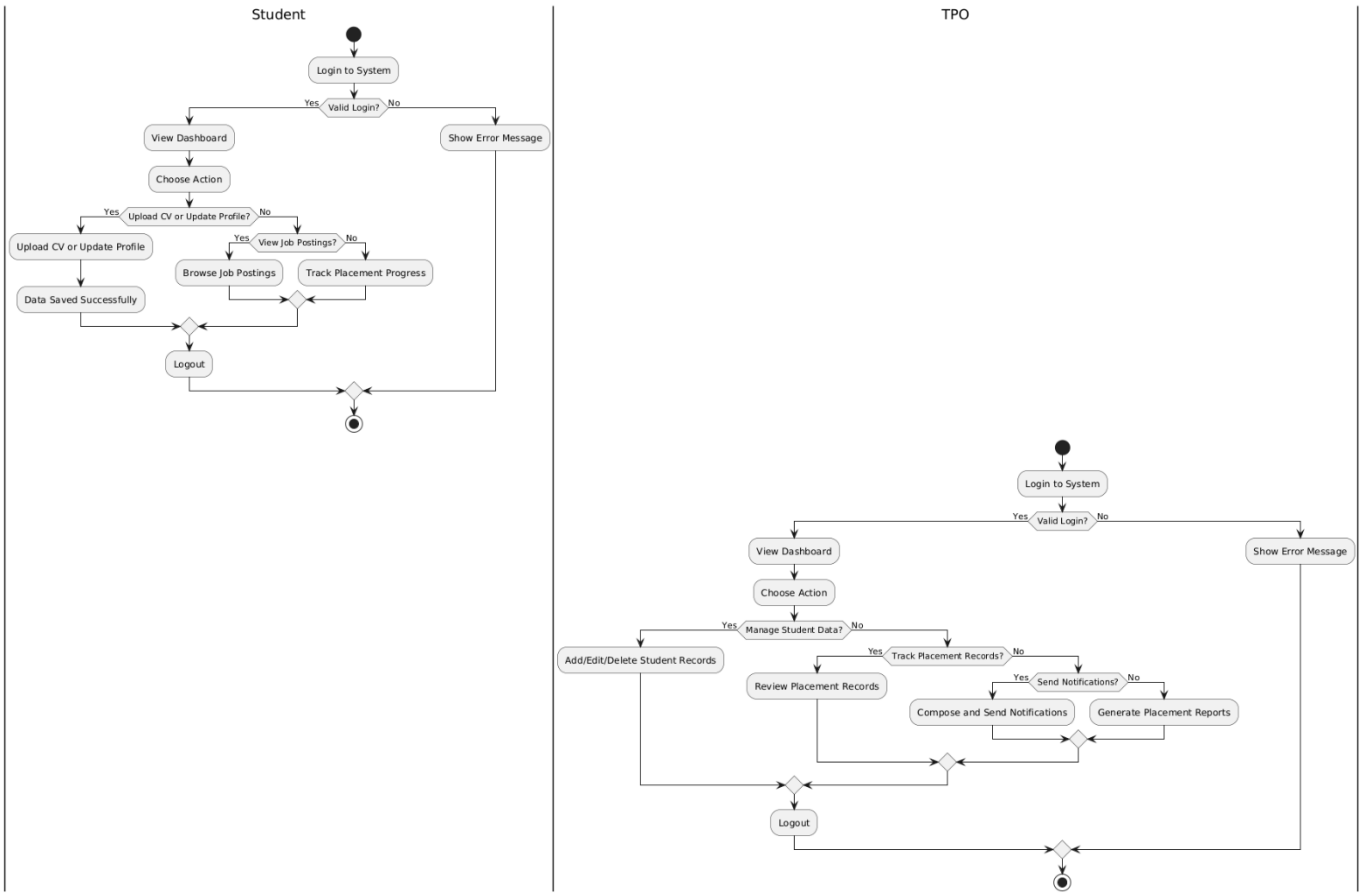

Fig 5.6.3 Activity Diagram

5.7 Assumptions and Dependencies

Assumptions

1) **User Availability:**
   - Students and TPOs will be available for feedback during development phases to ensure the system meets user requirements and expectations.
2) **Internet Access:**
   - The platform requires stable internet connectivity for users to interact with the system and for real-time data syncing.
3) **Data Accuracy:**
   - Users are expected to provide accurate, up-to-date information during registration and profile management for effective placement matching and tracking.
4) **Scalability Support:**
   - The system will support a limited user base initially, with plans to scale as the platform grows in terms of users and features.
5) **Compliance:**
   - The platform will comply with data protection regulations, such as GDPR, and other applicable regional laws regarding user data and privacy.

Dependencies

1) **Third-Party Services:**
   - The system relies on external APIs for notifications (email, SMS) and other integrations such as real-time chat or third-party tools if required for advanced features).
2) **Database Management System:**
   - The platform will use **Firebase Realtime Database** to store user and transaction data, providing fast access and synchronization for all users.
3) **Development Frameworks:**
   - The platform's development will depend on **Android (Java)** for the mobile front-end and **Firebase** for the backend to ensure scalability and real-time updates.
4) **Team Collaboration Tools:**
   - Project development and management will utilize tools such as **GitHub** for version control, **Jira** for task tracking, and **Slack** for team communication.
5) **Browser Compatibility:**
   - The platform assumes that users will access it on Android devices, with compatibility for devices running Android OS version 5.0 or higher.

6 Supporting Information

This section contains additional resources and tools to support the SRS document.

a) Table of Contents:

A detailed Table of Contents is included at the beginning of the SRS document to facilitate easy navigation.

b) Index:

An index is provided for quick lookup of key terms and topics within the document:

- SRS: Software Requirements Specification
- Matchmaking: Finding suitable connections between startups and corporates
- UI: User Interface
- UX: User Experience
- API: Application Programming Interface

c) Appendices:

1) **Use Case Storyboards:**
   - High-level illustrations or flowcharts depicting user interactions with the system.
   - Example: How a student applies for a job or updates their profile.
2) **User Interface Prototypes:**
   - Mockups or wireframes for major functionalities such as:
     - Registration/Login
     - Job search results
     - Profile management

d) Glossary of Terms:

Here is the glossary of technical and project-related terms for the **Training and Placement Department Online Platform**:

1) **SRS (Software Requirements Specification):** A document outlining the functional, non-functional, and operational requirements for the software system.
2) **Matchmaking:** The process of matching students with relevant job opportunities based on skills and qualifications.
3) **UI (User Interface):** The visual elements with which users interact, such as forms, buttons, and menus.
4) **UX (User Experience):** The overall experience users have while interacting with the platform, focusing on ease of use and satisfaction.
5) **API (Application Programming Interface):** A set of protocols for interacting with external systems and enabling smooth communication between the frontend and backend.
6) **OAuth 2.0:** A secure authorization protocol used for user login and data protection.
7) **SSL/TLS Encryption:** Protocols for ensuring secure data transmission over the internet.
8) **CI/CD (Continuous Integration/Continuous Deployment):** Practices for automating code integration, testing, and deployment to streamline development cycles.

e) References:

- Research papers, technical guides, and API documentation utilized during the project:
  - Global Startup Ecosystem Report (GSER) 2023
  - Firebase Documentation

7 Conclusion & Future Scope

Conclusion:

The **Online Centralized Platform for the TP Department** is designed to address the challenges faced by students, TPOs, and recruiters in the placement process, creating a seamless interface for interaction and collaboration. By providing a centralized platform, it facilitates efficient placement management, real-time communication, and streamlined recruitment processes. Features such as profile management, placement matching, interview scheduling, and notifications establish the platform as a comprehensive ecosystem for

improving placement success rates and the overall experience for students, TPOs, and recruiters.

The platform ensures that students have access to relevant job opportunities and updates, while recruiters can efficiently manage applications and interact with candidates. With its scalable architecture and user-friendly interface, the platform is adaptable for future feature enhancements and expansion, laying the foundation for a robust placement management system.

Future Scope:

The platform has significant potential for growth and enhancement. Below are additional opportunities for its future development:

1) **AI-Driven Placement Matching:**
   - Incorporate advanced AI algorithms to provide smarter, more accurate job matches for students based on their behavior and preferences.
2) **Mobile Application:**
   - Extend platform accessibility with native mobile apps for both Android and iOS devices, ensuring wider reach and user convenience.
3) **Global Expansion:**
   - Support multiple languages and currencies to cater to international students and recruiters.
4) **Enhanced Analytics:**
   - Provide deeper insights to students, TPOs, and recruiters through advanced data visualization and predictive analytics for placement trends and success rates.
5) **Gamification:**
   - Introduce gamified features such as badges, achievements, and leaderboards to increase user engagement and motivation.
6) **Community and Networking:**
   - Build an interactive community for students, TPOs, and recruiters to share knowledge, host webinars, and discuss placement-related challenges.
7) **Customer Support Enhancements:**
   - Implement AI-powered chatbots for real-time support and offer multi-channel assistance through email, chat, and voice to cater to diverse user needs.
8) **Premium Membership Tiers:**
   - Introduce subscription-based tiers offering advanced features like personalized mentorship programs, exclusive job opportunities, and priority placement matching.
9) **Partnership with Educational Institutions:**
   - Collaborate with universities and research centers to provide students with internships and student project opportunities, and connect corporates with academic collaborators for R&D.
10) **Integration with External Platforms:**
   - Integrate with platforms like LinkedIn and Glassdoor to import/export professional data, expand user reach, and enhance the overall user experience.

8 Concerns / Queries / Doubts

1) Technological Questions:

- What is the best framework to build the mobile frontend of the platform? Should Android (Java) or cross-platform frameworks like Flutter be used?
- Should the platform use a rule-based system or machine learning for personalized job recommendations and matchmaking between students and recruiters?

2) Scalability Queries:

- How should the platform scale handle a growing number of students, recruiters, and TPOs? Can Firebase scale effectively, or should we consider an alternative cloud solution like AWS or Google Cloud?
- How can the platform be designed to support future growth in terms of features and user capacity?

3) Security Concerns:

- s OAuth 2.0 sufficient for securing user sessions, or should multi-factor authentication (MFA) be implemented for added security?
- How can sensitive student and recruiter data be securely stored and transmitted, especially considering the potential for high-value information like resumes and job offers?

4) User Feedback Integration:

- How can the platform gather real-time feedback from students, recruiters, and TPOs to continuously improve the matching algorithm and overall user experience?

5) Regulatory Compliance:

- What steps are necessary to comply with GDPR and CCPA regulations for handling student and recruiter data, especially across different regions?
- How can we ensure that all user data is collected and managed in accordance with privacy laws, allowing users to access, update, and delete their data.

.