Software Requirement Specifications

1. Scope:

The system will include modules for students, companies, and administrators, allowing. Students to create profiles, browse job postings, apply for jobs, and track application status. Companies to post job listings, view applicant profiles, and shortlist candidates. Admins to manage users, oversee job listings, and generate reports. Notifications for job alerts, interview updates, and application status changes. Placement analytics to monitor trends and track success rate.

1.1 Identification

Identification No	CUTM-SRS-PD-VER-0.1
Title	Placement Dashboard
Abbreviation	CUTM-SRS-PD
Version No	0.1
Release No	0.1

1.2 System Overview

The Placement Dashboard consists of two primary components: the student interface and the administrator interface, both connected to a centralized server. The student interface allows users to register, update their profiles, view job postings, and track applications. The administrator interface enables recruiters and placement officers to manage job listings, review student applications, and generate reports. The server handles authentication, database management, and data processing to ensure smooth communication between both interfaces.

1.3 Document Overview

This document provides a comprehensive overview of the Placement Dashboard system. It details system connectivity, software and hardware requirements, functional descriptions, and key features. The document is structured as follows:

- Section 1: Introduction to system connectivity, requirements, and functionalities
- Section 2: Technical specifications, including software and hardware dependencies
- Section 3: Functional modules, detailing user interactions and workflows
- Section 4: Security considerations and performance metrics
- Section 5: Future enhancements and scalability considerations

1.4 Requirements of Software/Hardware

Software Requirements:

- Front-end: Kotlin
- Back-end: Firebase
- Development Environment - IDE

Hardware Requirements:

- Processor: Intel Core i5 and above , Ryzen 5.
- Equivalent RAM: 4 GB
- Higher Storage: 128 GB SSD

1.5 Brief Software Functional Description

The Placement Dashboard is a web-based application designed to facilitate and streamline the campus placement process. The system provides students with access to job postings, application tracking, and interview scheduling, while enabling administrators to manage job listings, evaluate applicants, and generate insightful reports. The software integrates machine learning models to recommend suitable job opportunities based on student profiles and past trends, enhancing placement success rates.

1.6 Functional Requirements

1.6.1 Major Functionalities:

i. Handling the Data:

- Store, retrieve, and manage student profiles, job listings, company details, and placement records.
- Ensure data consistency and integrity with a structured database (Spring Boot).
- Generate analytical reports on placement statistics, student performance, and hiring trends.

ii. Handling Administration:

- Allow administrators to manage job postings, eligibility criteria, and application deadlines.
- Enable recruiters to review student applications, schedule interviews, and provide feedback.
- Provide role-based access control for different user categories (students, admins, recruiters).
- Generate insights and reports to track placement trends, employer engagement, and hiring success.
- Implement notification and communication systems for event updates and job postings.

iii. Handling User Functions:

- Enable students to register, update profiles, and upload resumes.
- Allow students to browse job opportunities, apply for jobs, and track their application status.
- Ensure a user-friendly and intuitive interface for seamless navigation and interaction.

1.7 Brief Description of the System:

The Placement Dashboard is a web-based platform designed to facilitate campus placements by connecting students, administrators, and recruiters. It enables students to apply for jobs, track applications, and receive recommendations, while administrators manage job listings, monitor student progress, and reports. Recruiters can review applications and schedule interviews. The system ensures a smooth and efficient placement process with secure authentication, real-time data updates, and a user-friendly interface.

1.7.1 Input Interfaces:

Student Interface:

- Registration Form: Allows students to create an account with details like name, email, username and password.
- Profile Management: Students can enter educational qualifications, skills, and resume.
- Job Application Form: Students can apply for job postings by selecting a job and submitting necessary documents.

1.8 Development Details:

s.no	Role	Details
1	Developer-Admin	Centurion University
2	User-Client	Placement Dashboard

2.Referenced Documents:

1. Garcia, R., & Puig, J. (2010). Student Internship Placements: Improving the quality of engineering internship programmes. *IEEE EDUCON 2010 Conference*, 91–98. https://doi.org/10.1109/EDUCON.2010.5492592.
2. Ibrahim, N., Hanum, H. M., Abu Bakar, Z., & Abdullah, N. A. S. (2021). Student-Industry Matching for Internship Placement. *2021 Fifth International Conference on Information Retrieval and Knowledge Management (CAMP)*, 122–126. https://doi.org/10.1109/CAMP51653.2021.9498088.
3. Jewani, G., Sahare, S., Kamble, T., Kathalkar, R., & Unhale, A. (2023). Online Training and Placement System. *2023 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS)*, 1–5. https://doi.org/10.1109/SCEECS57921.2023.10063051
4. Kousik, R. K., & Nagappan, G. (2024). Computer Human Interface for Placement Management System. *2024 IEEE International Conference on Computing, Power and CommunicationTechnologies(IC2PCT)*,1245–1248. https://doi.org/10.1109/IC2PCT60090.2024.10486671
5. Pradipta, P. P., Edward, I. J. M., Iskandar, & Favitri Hariyanto, D. (2021). Design and Implementation of Software and Web Dashboard on Long Range Communication Systems for Rural Area. *2021 7th International Conference on Wireless and Telematics (ICWT)*, 1–6. https://doi.org/10.1109/ICWT52862.2021.9678448
6. Shimpi, P., Balinge, B., Golait, T., Parthasarathi, S., Arunima, C. J., & Mali, Y. (2024). Job Crafter - The One-Stop Placement Portal. *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, 1–8. https://doi.org/10.1109/ICCCNT61001.2024.10725010
7. Shivani, Srivastava, R., & Tiwari, N. (2022). Developing an E-learning and Job Portal for IT Aspirants. *2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*, 160–165. https://doi.org/10.1109/ICAAIC53929.2022.9792727.
8. Srivastava, N., Tripathi, M., & Rai, V. (2023). The Development of a Job Portal to Facilitate Incampus Placement. *2023 5th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*, 1549–1556. https://doi.org/10.1109/ICAC3N60023.2023.10541560

9. Tolle, H., Fanani, L., & Pasaribu, S. A. (2023). Design of Job Placement Center Dashboard for Monitoring Alumni Performance in Job Fair Application. *2023 Eighth International Conference on Informatics and Computing (ICIC)*, 1–6. https://doi.org/10.1109/ICIC60109.2023.10381912
10. Tyagi, D., Kazim, D., Bhadra, S., Gupta, A., Kumar, P., Sharma, A., & Chaudhary, H. (2023). Job and Internship Assistance Application. *2023 International Conference on DisruptiveTechnologies(ICDT)*,571–575. https://doi.org/10.1109/ICDT57929.2023.10150490.

3. Requirements:

3.1 Required Mode of Operation:

The Placement Dashboard operates in multiple modes to streamline the placement process. Student Mode allows students to register, update profiles, apply for jobs, and track applications. Administrator Mode enables job management, application monitoring, and report generation, while Recruiter Mode lets recruiters post jobs, review candidates, and schedule interviews. The System Mode handles authentication, data processing, and job recommendations, while Real-time Notification Mode sends automated updates. Lastly, Reporting & Analytics Mode provides insights into placement trends, ensuring an efficient and transparent process.

3.2 System Capability Requirements

3.2.1 System Software Requirements

The placement dashboard system runs on windows with a FireBase as backend,a kotlin -based Androidapp for frontend ,and firestore for database.

3.2.2 Application Software Requirements

The Placement Dashboard system allows users to manage profiles, track placement records, and facilitate recruitment processes. It ensures secure authentication, real-time data access, and seamless interaction between students, companies, and administrators. The system supports multiple users, maintains data integrity, and provides efficient communication through REST APIs, ensuring a high response rate.

3.2.2.1 Requirements Table

Table 1: Software Requirements

ID No	Software Requirements
PDS-SRS-01	User Authentication (Login/Signup)
PDS-SRS-02	Student Profile Management
PDS-SRS-03	Admin Dashboard
PDS-SRS-04	Placement Record Tracking
PDS-SRS-05	Company and Job Management
PDS-SRS-06	Logout Functionality

3.2.2.2 Brief Description of Requirements

Table 2: Brief Functional Requirements

ID No	Functional Requirement
PDS-SRS-01	User Authentication (Login/Signup)
PDS-SRS-01-01	Username
PDS-SRS-01-02	Password
PDS-SRS-01-03	Login
PDS-SRS-02	Student Dashboard
PDS-SRS-02-01	Profile Management
PDS-SRS-02-02	Applied Jobs
PDS-SRS-02-03	Placement Status
PDS-SRS-03	Admin Dashboard
PDS-SRS-03-01	Placement Records
PDS-SRS-03-01-01	Navigation Bar
PDS-SRS-03-01-01-01	Overall Dashboard
PDS-SRS-03-01-01-02	Student Dashboard
PDS-SRS-03-01-01-03	Company Dashboard

PDS-SRS-03-01-01-04	Job Dashboard
PDS-SRS-04	Company and Job Management
PDS-SRS-04-01	Job ID
PDS-SRS-04-02	Job Type
PDS-SRS-04-03	Eligibility Criteria
PDS-SRS-04-04	Job Openings
PDS-SRS-05	Placement Tracking
PDS-SRS-05-01	Student Placement Records
PDS-SRS-05-02	Company Hiring History
PDS-SRS-06	Logout Functionality

3.2.3 Detailed Description of Requirements

Table 3: Detailed Functional Requirements

ID	Description
PDS-SRS-01	**Feature:** User Authentication (Login/Signup) **Input:** User enters username and password. **Process:**•The system displays username and password fields. •User enters credentials and taps Login. • The system encrypts credentials and sends a request to the backend. •The server validates credentials against the database. •If correct, it returns user data and a session token. •If incorrect, an error message is displayed.

	•On successful login, the user is redirected to the dashboard. **Output:** User logs in successfully or sees an error message.
PDS-SRS-01-01	**Feature:** Username Field **Input:** User enters a username. **Process:** • The system checks if the username exists in the database. • If found, allows the user to proceed to the password field. • If not found, prompts the user with "Invalid username." **Output:** Accepts username or prompts for re-entry.
PDS-SRS-01-02	**Feature:** Password Field **Input:** User enters a password. **Process:** • System checks if the password matches the stored data. • If correct, moves to authentication; otherwise, an error is prompted. • User can toggle password visibility for verification. **Output:** If correct, allows login; otherwise, an error appears.
PDS-SRS-02	**Feature:** Admin Dashboard **Input:** Admin logs in. **Process:** • Fetches user role and permissions from the backend. • Loads relevant admin options like student management, job postings, and placement tracking. • Displays a summary of system operations:- Number of registered

	students - Number of job postings.- Placement statistics. **Output:** Admin can access system functions.
PDS-SRS-02-01	**Feature:** Student Management **Input:** Admin selects "Student Management." **Process:** • Retrieves all student records from the database. •Displays student details including: - Student Name- Enrollment Number - Placement Status • Allows filtering by placement status (Placed/Unplaced). **Output:** Student details are displayed.
PDS-SRS-02-02	**Feature:** Job Postings **Input:** Admin selects "Job Postings." **Process:** • Retrieves all job postings from the database. • Displays job details including:- Company Name - Job Role - Eligibility Criteria • Allows adding new job postings. **Output:** Job postings are displayed and managed.
PDS-SRS-02-03	**Feature:** Placement Statistics **Input:** Admin selects "Placement Statistics." **Process:** • Retrieves placement data from the database. • Displays statistics including: - Number of students placed- Companies visited- Average package offered **Output:** Placement statistics are displayed.

Placement Dashnoard-PD

PDS-SRS-03	**Feature:** Student Dashboard **Input:** Student logs in. **Process:** • Fetches student details from the database. • Displays job opportunities based on student profile. • Allows students to apply for jobs. **Output:** Students can view and apply for job opportunities.
PDS-SRS-03-01	**Feature:** Apply for Jobs **Input:** Student selects a job posting. **Process:** • The system fetches job details •Verifies if the student meets the eligibility criteria. •Allows students to submit applications. **Output:** Student application is submitted.
PDS-SRS-04	**Feature:** Notifications System **Input:** System generates notifications for job postings, interview schedules, and placement updates. **Process:** • Retrieves new job postings and updates from the database. • Sends notifications to students based on their profile. **Output:**Students receive relevant notifications.
PDS-SRS-05	**Feature:** Logout **Input:** User selects "Logout." **Process:** •System clears session data. • Redirects the user to the login screen.

	Output: User is logged out successfully.

3.3 CSCI (Computer Software Configuration Item):

The system consists of an **Android frontend** built with Kotlin for user interaction, a **FireBase backend** for processing API requests and managing business logic, and a **MySQL database** for storing student records, job postings, applications, and placement statistics. This architecture ensures a scalable, efficient, and well-integrated system.

3.3.1 CSCI External Interface Requirements

The system does not require external interfaces, as all operations are managed within the internal network of the institution. Users (students, administrators, and recruiters) interact with the system through the Android application, which communicates with the backend via secure APIs.

3.4 CSCI Internal Interface Requirements

All internal interfaces will be designed to ensure smooth data exchange between the **Android application, Fire Base backend, and MySQL database**. These interfaces will support functionalities such as:

- User authentication and authorization (login/signup).
- Job postings retrieval and application submissions.
- Placement statistics management and reporting.
- Notifications for job updates and interview schedules.

3.5 CSCI Internal Data Requirements

Internal data, including **student profiles, job postings, applications, interview schedules, and placement records**, will be securely structured and stored in the MySQL database. The database schema will be designed to efficiently manage data access and retrieval for seamless system performance.

3.6 Adaptation Requirements

The system is designed to be adaptable across different institutions and placement processes with minimal customization. Configurations such as **job eligibility criteria, user roles, and placement policies** can be modified at the implementation level to fit specific requirements.

3.7 Safety Requirements

Since the system primarily handles digital data, no critical safety measures are required. However, proper **data validation, error handling, and backup mechanisms** will be in place to prevent data corruption and ensure system stability.

3.8 Security and Privacy Requirements

The system will enforce strict **role-based access control (RBAC)** to ensure that sensitive student and placement data remain protected. Key security features include:

- **Encryption** of stored passwords and sensitive information.
- **Access control** to restrict database operations based on user roles (Student, Admin, Recruiter).
- **Secure API communication** using authentication tokens.
- **Data privacy measures** to prevent unauthorized access to student and company records.

4 Database backend

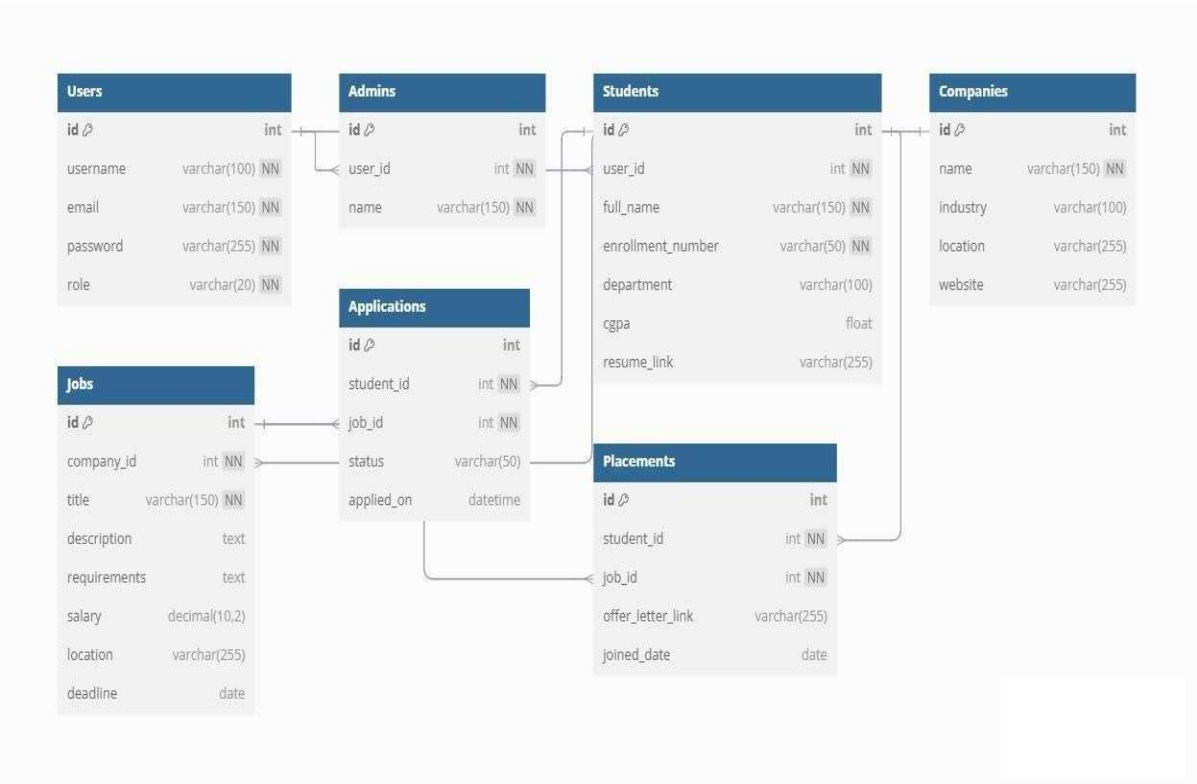

Fig 4.1 Database Backend

The **ERD** represents **Placement Dashboard System** with structured relationships between **Users**, **Jobs**, **Companies**, **Applications**, and **Placements**. **Students** apply for jobs, and their applications are tracked in the **Applications** table. **Admins** manage job postings from **Companies**, while **Placements** store selected students' details, including offer letters and joining dates. The system ensures efficient data flow and role-based access control.