Overview

The Integrated Placement Management System follows a modern microservices architecture designed to handle multiple stakeholder interactions, real-time data processing, and AI-driven recommendations. The system is built on cloud-native principles ensuring scalability, reliability, and maintainability while providing seamless user experiences across web and mobile platforms.

(Client-Side)

- Mobile Applications (Flutter - iOS/Android)
- Database (firestore)

Detailed Module Breakdown

Module 1: User Management Service

- Responsibility: user profiles, role-based access control
- Technology: dart
- Database: firestore
- Interfaces: gmail API

Module 2: Company Management Service

- Responsibility: Company profiles, job postings, recruitment drives, company requirements

Module 3: Student Profile Service

- Responsibility: emails, placement progress tracking

Module 5: Mentorship Platform Service

- Responsibility: Alumni-student connections, mentorship sessions, feedback systems
- Features: chat

Module 6: Communication Service

- Responsibility: Notifications, emails, SMS, in-app messaging
- Features: Multi-channel notifications, delivery tracking, template management

Module 7: Analytics and Reporting Service

- Responsibility: Placement statistics, performance metrics, predictive analytics
- Features: Real-time dashboards, trend analysis, predictive modeling

Module 8: Assessment and Quiz Service

- Responsibility: Skill assessments, company-specific quizzes, progress tracking

- Features: Adaptive questioning, performance analytics

9.2.4 Data Flow Architecture

Primary Data Flows:

1. User Registration & Authentication Flow
2. Client → API Gateway → User Management Service → firestore
3. ↓
4. Student → Company details, ask alumni, placement history, announcements, quizzes', resume, alumni career
5. HOD → Emails,Companies,Placed Students,Announcements,Quizzes,Student Scores,Alumni Careers,Ask Alumni
6. Faculty → Emails,Companies,Placed Students,Announcements,Quizzes,Student Scores,Alumni Careers,Ask Alumni
7. placement corrdinator → Emails,Companies,Placed Students,Announcements,Quizzes,Student Scores,Alumni Careers,Ask Alumni
8. alumni→

9.2.5 Modularity Benefits

Maintainability

- Each service can be updated independently without affecting others
- Clear separation of concerns reduces debugging complexity

Reusability

- Authentication service can be reused across all modules
- Communication service provides notifications for any business event

Extensibility

- New features can be added as separate services
- Existing services can be enhanced without system-wide changes
- Third-party integrations can be easily incorporated through API adapters

9.3 Technology Stack Specification

Mobile Application: Flutter

- Justification: Single codebase for iOS and Android reduces development and maintenance costs by 40%. Flutter's native performance ensures smooth animations and quick response times crucial for student engagement. Hot reload functionality accelerates development cycles.