**Design Document (DD) for Students & Companies Platform**

**1. Introduction**

**1.1 Purpose**

This document describes the architectural and design aspects of the "Students & Companies (S&C)" platform. It includes the rationale behind the architectural choices, detailed design components, and the mapping of requirements to the design. The purpose of the S&C platform is to connect students seeking internships with companies offering opportunities, providing a streamlined application and matching system enhanced by machine learning.

**1.2 Scope**

The platform serves two primary user groups:

1. **Students**: Create profiles, upload CVs, search for internships, apply, and track application status.
2. **Companies**: Post internships, review applications, shortlist candidates, and communicate with applicants.

The system leverages recommendation algorithms to match internships and candidates efficiently and incorporates feedback mechanisms to improve its services continuously.

**1.3 Definitions, Acronyms, Abbreviations**

* **RASD**: Requirements Analysis and Specification Document
* **DD**: Design Document
* **S&C**: Students & Companies
* **UI**: User Interface
* **ML**: Machine Learning

**1.4 Revision History**

* Version 1.0 (January 2025): Initial draft.

**1.5 Reference Documents**

* Requirements Analysis and Specification Document (RASD) for S&C platform.
* DD Codekata sample.

**2. Architectural Design**

**2.1 Overview**

The architecture follows a **three-tier design** to ensure scalability, modularity, and separation of concerns:

* **Presentation Layer**: Web application interface for students and companies.
* **Application Layer**: Business logic, including recommendation algorithms and application tracking.
* **Data Layer**: Relational database to store user data, internship postings, and application records.

**2.2 Component View**

The system consists of the following components:

1. **Web Application**: Interface for users to interact with the platform.
2. **Authentication Manager**: Handles user authentication and session management.
3. **Profile Manager**: Manages user profiles and CV uploads.
4. **Recommendation Engine**: Suggests internships based on student profiles.
5. **Internship Manager**: Allows companies to post internships and manage applications.
6. **Application Tracker**: Tracks applications, statuses, and communication.
7. **Notification System**: Sends alerts and updates to users.
8. **Database**: Stores user data, postings, and application records.

**2.3 Deployment View**

The platform will be deployed on cloud infrastructure using:

* **Web Server**: Hosts the web application.
* **Application Server**: Runs business logic and API endpoints.
* **Database Server**: Manages relational data storage.

**2.4 Runtime View**

During runtime, the system processes the following:

1. User login and authentication.
2. Profile creation or editing.
3. Internship posting and application.
4. Application status updates and notifications.
5. Recommendation generation based on new postings or profile changes.

**2.5 Component Interfaces**

* **WebApp**: Exposes the user interface.
* **API**: Facilitates communication between the frontend and backend components.
* **Database**: Provides persistent storage.
* **Notification Service**: Sends notifications to users about updates.

**3. User Interface Design**

**3.1 Screens**

1. **Login and Registration**: Allows users to create accounts or log in.
2. **Dashboard**:
   * Students: View recommended internships and application statuses.
   * Companies: View internship postings and applicant details.
3. **Profile Management**:
   * Students: Upload CVs and update personal information.
   * Companies: Update company profiles.
4. **Internship Management**:
   * Students: Search and apply for internships.
   * Companies: Post and manage internships.

**4. Requirements Traceability**

**Functional Requirements Mapping**

1. **[FR1]** User registration and login:
   * Components: Authentication Manager, Database.
2. **[FR2]** Profile creation and management:
   * Components: Profile Manager, Database.
3. **[FR3]** Internship posting and management:
   * Components: Internship Manager, Database.
4. **[FR4]** Application tracking:
   * Components: Application Tracker, Notification System, Database.
5. **[FR5]** Recommendation engine:
   * Components: Recommendation Engine, Database.

**5. Implementation, Integration, and Test Plan**

**5.1 Overview**

Implementation will follow an iterative and incremental approach. Each component will be implemented, tested, and integrated into the overall system.

**5.2 Implementation Plan**

1. Develop the web application interface.
2. Implement backend components:
   * Authentication Manager
   * Profile Manager
   * Internship Manager
   * Recommendation Engine
3. Set up the database schema and integrate with backend APIs.
4. Deploy the system to a cloud environment.

**5.3 Testing Plan**

1. **Unit Testing**: Test individual components for functionality.
2. **Integration Testing**: Ensure seamless communication between components.
3. **System Testing**: Verify the entire system against requirements.
4. **Performance Testing**: Test the system under high loads.
5. **Usability Testing**: Gather user feedback for interface improvements.

**6. Effort Spent**

Each team member contributed equally, spending approximately 20 hours each on research, drafting, and revising this document.

**7. Bibliography**

1. RASD-V1 Document for S&C Platform.
2. DD Codekata sample document.
3. General software engineering resources on architectural and design principles.