Software Requirements Specification

for

ALUMNEXUS

(ALUMINI ASSOCIATION PLATFORM)

Prepared by

GroupName: TEAM 3

KANIKA R	23Z334	23z334@psgtech.ac.in
KEERTHANAA P	23Z335	23z335@psgtech.ac.in
MEENALOSHINI V	23Z343	23z343@psgtech.ac.in

INSTRUCTOR	: Mr.Ramesh.A.C
COURSE	: Application Development Laboratory
LAB SECTION	: Friday.
DATE	: 06/02/2025

Contents

1 Introduction

This section provides an introduction to the **Alumnexus** platform, outlining its purpose, scope, and intended audience. The document specifies the software requirements, functional and non-functional expectations, and system interactions for the platform. Readers will find a structured breakdown of the system's features, goals, and interfaces to ensure a comprehensive understanding of the project.

1.1 Document Purpose

This document defines the software requirements specification (SRS) for **Alumnexus**— a digital alumni engagement platform. The purpose of this document is to describe the functional and non-functional requirements of the system, ensuring clarity for stakeholders, developers, and testers.

Alumnexus aims to streamline alumni engagement, mentorship, career opportunities, and donations through a centralized web-based platform. This SRS document details the core functionalities, user roles, and key system interfaces. It covers all major system components, including alumni networking, job portal, event management, and donation handling.

1.2 Product Scope

The **Alumnexus** platform is a **web-based alumni association system** that connects **graduates, faculty, and students** of an institution. It facilitates professional networking, job postings, mentorship programs, event management, and donation processing.

Key Benefits and Objectives:

- **Strengthen Alumni Engagement** – Provides an interactive platform for alumni to stay connected.
- **Networking & Mentorship** – Enables alumni to connect with peers and guide students.
- **Career Advancement** – Offers a job portal for alumni and students to find career opportunities.
- **Event Management** – Simplifies alumni reunions and professional event coordination.
- **Fundraising & Donations** – Provides a secure and transparent system for financial contributions.

By integrating these features**,** Alumnexus fosters lifelong relationships between alumni and their alma mater, ensuring continued professional growth and institutional support**.**

1.3 Intended Audience and Document Overview

This document is intended for the following stakeholders:

1. **Developers** – Responsible for implementing the platform based on the specified requirements. They will use this document to understand system functionalities, external interfaces, and performance constraints.

2. **Project Managers** – Oversee the development process, ensuring the system is built according to the requirements and delivered on time.
3. **Clients & Professors** – Review the project's progress and ensure the final product meets institutional expectations.
4. **Testers & QA Engineers –** Validate system performance, security, and usability against the specified requirements before deployment.
5. **System Administrators –** Manage and maintain the platform after deployment, ensuring security, data integrity, and system availability.

Document Overview

This Software Requirements Specification (SRS) document provides a detailed description of the **Alumnexus** platform, outlining its objectives, features, constraints, and system interactions.

The document is structured as follows:

- **Section 1: Introduction –** Provides an overview of the document, including the purpose, scope, and audience.
- **Section 2: Overall Description –** Explains system context, dependencies, assumptions, and high-level functional descriptions.
- **Section 3: External Interface Requirements –** Details user, hardware, and software interfaces necessary for system operation.
- **Section 4: System Requirements –** Specifies performance, security, and software quality attributes required for system functionality.

Reading Guide for Different Stakeholders

- **Clients & Professors**: Start with **Section 1 (Introduction)** and **Section 2 (Overall Description)** to understand system goals and scope.
- **Developers**: Focus on **Sections 3 & 4** for detailed technical specifications.
- **Testers & QA Engineers**: Refer to **Section 4** to validate functional and performance requirements.
- **System Administrators**: Review **Sections 3.1 (Hardware & Software Interfaces)** and **4.2 (Security & Safety Requirements)** for infrastructure and security guidelines.

1.4 Definitions, Acronyms and Abbreviations

Term	Definition
Alumnexus	The Alumni Association Networking Platform being developed.
API	Application Programming Interface – A set of functions that allow external software systems to communicate with Alumnexus.
Authentication	The process of verifying a user's identity using credentials such as email/password or social login.
Backend	The server-side component of the application responsible for processing data and managing logic.
CDN	Content Delivery Network – A system of distributed servers that improves web content delivery speed and availability.
CMS	Content Management System – A software application used to create, manage, and modify digital content.
DBMS	Database Management System – Software used to store, manage, and retrieve structured data (e.g., MySQL, PostgreSQL).
Frontend	The user interface (UI) that allows users to interact with the system through a web browser or mobile app.
GDPR	General Data Protection Regulation – A European regulation that governs the collection and processing of personal data.
HTTPS	Hypertext Transfer Protocol Secure – A secure version of HTTP that encrypts communications between the client and the server.
MFA	Multi-Factor Authentication – A security feature that requires users to verify their identity using multiple authentication methods.
OAuth	Open Authorization – A standard for secure user authentication via third-party services (e.g., Google, LinkedIn).
RBAC	Role-Based Access Control – A security model that restricts user access based on their assigned roles (e.g., alumni, admin).
REST API	Representational State Transfer API – A web service that allows communication between the frontend and backend using HTTP requests.
RSVP	Répondez s'il vous plaît – A feature allowing users to confirm attendance for events.
SRS	Software Requirements Specification – A document outlining the functional and non-functional requirements of the software.
TLS	Transport Layer Security – A cryptographic protocol that secures online communications.
UI/UX	User Interface/User Experience – The design and usability aspects of the software to ensure a smooth user experience.

1.5 Document Conventions

This document follows the **IEEE 830-1998 Standard for SRS** formatting guidelines.

Formatting Conventions:

- **Font:** Arial, size **11** or **12**.
- **Spacing:** Single-spaced with **1" margins**.
- **Section Titles:** Numbered (e.g., 1.1, 1.2) and bold.
- **Italics:** Used for notes or comments.

Naming Conventions:

- All **database tables** use **Pascal Case** (e.g., User Profile, Event Registration).
- Variables and functions in **code** follow **camelCase** (e.g., getUserData()).
- API endpoints follow **RESTful naming** (/api/alumni/profile).

1.6 References and Acknowledgements

The following sources were used as references for research, design, and development of the **Alumnexus** platform:

1. **International Journal of Research in Engineering and Science (IJRES)**
   - A Study on Alumni Management Systems
   - This paper analyzes different alumni engagement models, highlighting best practices in system design and functionality.
2. **Reconnectify: An Alumni Association Platform**
   - ResearchGate Publication
   - A study on how digital platforms facilitate alumni networking, focusing on community-building features.
3. **Implementation of a Multi-Environment Alumni Association with Location Identification**
   - Academia Research Paper
   - This research explores how geolocation-based services can enhance alumni networking opportunities.
4. **Development of an Intelligent Alumni Management System for Universities**
   - Academia.edu Paper
   - This source provides insights into AI-driven alumni engagement, focusing on automation and smart recommendations.
5. **International Journal for Multidisciplinary Research (IJFMR)**
   - Alumni Association and Career Services Integration
   - A comprehensive study on how alumni platforms integrate with job placement and career mentoring services.
6. **LinkedIn – Professional Networking & Alumni Engagement**
   - LinkedIn Alumni Tool
   - This platform provides a structured way for alumni to connect based on industries, locations, and institutions.

- Used as a reference for designing networking and career-building features in Alumnexus.

7. **College Alumni Websites**
    - Official alumni pages from various universities were analyzed for **event management, career services, and donation functionalities**.
    - Used to understand **how institutions manage alumni engagement** through structured portals.

Acknowledgments

We extend our gratitude to:

- **Our professors and mentors** for their guidance and valuable feedback throughout the development of this platform.
- **Our peers and testers** who provided insights to improve the usability and functionality of the system.
- **LinkedIn and various university alumni portals** for serving as inspiration in designing effective networking and engagement strategies.
- **Open-source communities and research contributors** whose studies and frameworks helped shape the design and implementation of the Alumnexus platform.

2 Overall Description

2.1 Product Overview

The **Alumnexus** platform is a **new, self-contained web-based system** designed to enhance **alumni engagement, professional networking, mentorship, career opportunities, and event coordination** within educational institutions. Traditional alumni management methods, such as spreadsheets, social media groups, and manual event coordination, lack structure and scalability. **Alumnexus replaces these fragmented approaches** with a centralized **digital ecosystem** that seamlessly integrates alumni, students, faculty, and administrators.

Product Context

The system is designed as a **standalone web application** but will integrate with **external APIs and third-party services** to enhance its functionality.

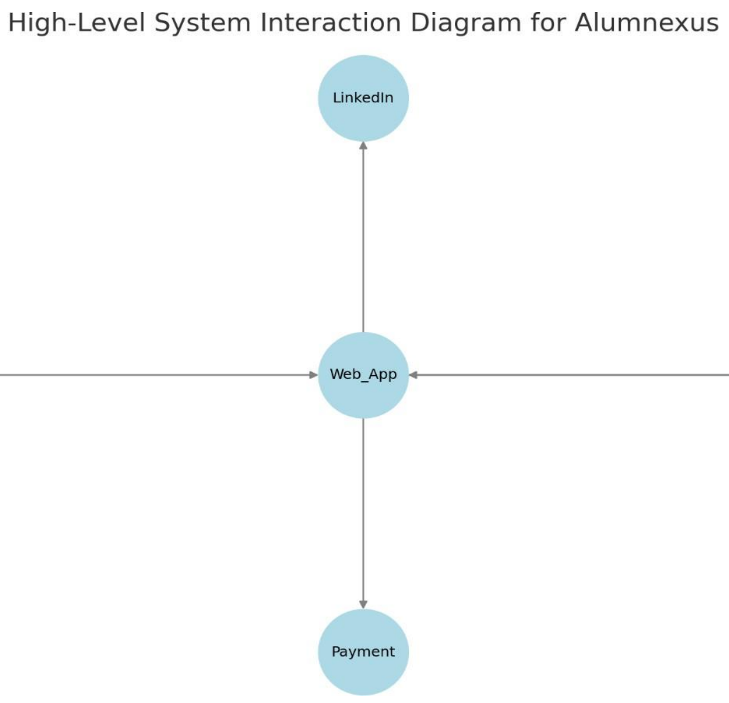
1. Users interact via the Alumnexus Web Platform through an intuitive graphical interface.
2. Data is stored and processed in a **secure cloud-hosted database**, supporting alumni profiles, job postings, events, and donations.
3. **External services such as LinkedIn and third-party payment gateways** are integrated to extend networking and financial functionalities.

Key Features & Differentiation

- **Seamless Networking & Mentorship:** Advanced filtering and AI-based recommendations for alumni connections.
- **Career Services & Job Portal:** Enables alumni to post job opportunities while students can apply for career guidance.
- **Event Management & Reunions:** Simplifies event coordination with automated notifications and RSVP tracking.
- **Fundraising & Donations:** Secure payment processing for institutional support and scholarship contributions.
- **Role-Based Access Control (RBAC):** Ensures user security with role-specific permissions for alumni, students, faculty, and administrators.

System Context & Interactions

The following **high-level diagram** illustrates the **major system components** and their interactions.



Here is the **High-Level System Interaction Diagram** for **Alumnexus**:

- **Users (Alumni, Students, Faculty, Admins)** interact with the **Alumnexus Web Platform**.
- The platform retrieves and updates data from the **Alumnexus Database**.
- **LinkedIn API** allows users to integrate their professional profiles.
- **Payment Gateway** processes donations and financial transactions.
- **Admin Dashboard** enables administrators to manage system activities.

2.2 Product Functionality

The **Alumnexus** platform provides the following core functionalities: User

Management & Authentication

- Secure user registration and login with **multi-factor authentication (MFA)**.
- Role-Based Access Control (RBAC) for alumni, students, faculty, and admins.
- **Profile Management** with editable education, career, and contact details.

Alumni Networking & Mentorship

- Advanced search and filter options to find alumni based on graduation year, industry, location, and skills.
- Direct messaging and group discussions for mentorship and networking.
- Connection requests with approval-based networking.

Job Portal & Career Services

- Alumni and recruiters can post job opportunities **with detailed descriptions.**
- Students and alumni can apply for jobs, upload resumes, and track applications.
- **Career guidance and mentorship** features for job seekers.

Event Management & Reunions

- Event organizers can create and manage events with **RSVP tracking**.
- Automated email reminders for upcoming reunions, webinars, and networking sessions.
- Event calendar integration with user dashboards.

Donations & Fundraising

- Secure **online payment gateway integration** (Stripe, PayPal) for alumni contributions.
- Real-time donation tracking for transparency.
- Custom fundraising campaigns for scholarships and research initiatives.

Admin Dashboard & Content Management

- User role and permission management.
- Event moderation and approval system.
- Analytics dashboard for tracking user engagement and financial contributions.

2.3 Design and Implementation Constraints

The development of **Alumnexus** is subject to the following constraints: Technology

2.3.1.Stack Constraints

- Must use COMET methodology for software design. (Reference: COMET - Component-based Software Engineering)
- UML (Unified Modeling Language) must be used for designing system models. (Reference: UML Standards by OMG - Object Management Group)
- Developed as a web-based platform with support for mobile-responsive design.

2.3.2.Hardware & Software Limitations

- Cloud-hosted infrastructure (AWS, Azure, or Google Cloud) required **to handle scalability.**
- Database must **support at least 1 million alumni profiles while maintaining fast query execution times.**
- Optimized for modern web browsers (Chrome, Firefox, Safari, Edge)**.**

2.3.3.Security & Compliance Constraints

- Encryption (AES-256 for data, TLS 1.2+ for communication) is mandatory**.**
- Compliance with GDPR & CCPA **for data privacy.**
- Rate limiting & CAPTCHA integration **to prevent spam and bot activity.**

2.3.4.Integration Constraints

- The system must integrate with **LinkedIn's API** for profile synchronization**.**
- Must support third-party payment gateways (Stripe, PayPal, Razorpay) **for donations.**
- **Email & Notification System** must support SMTP or API-based services (e.g., SendGrid, Mailgun).

2.4 Assumptions and Dependencies

The **Alumnexus** platform development is based on the following assumptions and dependencies:

Assumptions

- Users will have stable internet access to interact with the web-based platform.
- Educational institutions will provide initial alumni data for population in the system.
- Alumni and students will actively engage with the platform for networking and mentorship.
- Third-party APIs (LinkedIn, payment gateways) will remain stable and functional for seamless integration.

Dependencies

- **Cloud Hosting Services –** The platform depends on **AWS/Azure/GCP** for deployment and storage.
- **Database Management System (DBMS)** – **MySQL/PostgreSQL** is required for structured data handling.
- **Authentication Services** – OAuth2.0 for secure third-party login options (Google, LinkedIn).
- **Payment Processing –** The donation system relies on third-party payment gateways (Stripe, PayPal).
- **SMTP/Email API Services –** Required for email notifications and alerts (SendGrid, Mailgun, AWS SES).

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The **Alumnexus** platform provides an intuitive and interactive **Graphical User Interface (GUI)** accessible via **web browsers**. The interface is designed to support **alumni, students, faculty, and administrators**, ensuring seamless access to networking, job opportunities, and event management.

User Interaction Methods:

- **Navigation Menu**: A sidebar or top navigation bar containing links to key features.
- **Dashboard Interface**: A personalized homepage displaying recent alumni activities, event notifications, and suggested connections.
- **Profile Management**: Users can update their profile details, manage privacy settings, and interact with alumni.
- **Event & Job Portal**: A structured interface for browsing job postings and registering for events.
- **Messaging & Notifications**: Users receive real-time updates via pop-up notifications and inbox messages.

3.1.2 Hardware Interfaces

The **Alumnexus** platform interacts with several hardware components, primarily through web-based operations. Below are the key hardware interfaces:

Supported Device Types:

- **User Devices:**
  - Desktops, Laptops (Windows, macOS, Linux)
  - Smartphones, Tablets (iOS, Android)
  - Web Browsers (Chrome, Firefox, Safari, Edge)
- **Servers:**
  - Web Servers (Handles HTTP requests and dynamic content)
  - Database Servers (Stores alumni data, events, job listings)
  - Authentication Servers (Manages user authentication and security)
- **Networking Equipment:**
  - Routers, Firewalls (Ensures secure and reliable communication)
  - CDN (Optimizes platform performance and load balancing)
- **Payment Gateways:**
  - Online payment processing systems for handling alumni donations.
- **Storage Systems:**
  - Cloud Storage (Stores multimedia content, user documents)
  - Database (MySQL, PostgreSQL, MongoDB for structured data storage)

Each component exchanges data securely through **encrypted connections**, ensuring efficient and safe operations.

3.1.3 Software Interfaces

The **Alumnexus** platform interacts with multiple software components to ensure full functionality:

- **Database Management System (DBMS):**
    - Stores alumni profiles, messages, job postings, and event details.
    - Provides structured queries for efficient data retrieval.
- **Authentication and Security Modules:**
    - Manages user authentication, password encryption, and session handling.
- **Payment Gateway API:**
    - Processes online donations through third-party payment providers.
- **Messaging & Notification System:**
    - Sends real-time notifications via email, in-app alerts, and push notifications.
- **RESTful APIs for Integration:**
    - Facilitates interactions between the **mobile app and web platform**.

3.2 Functional Requirement

Functional requirements define the core behaviours and features of the **Alumnexus** system.

3.2.1 F1: User Registration & Authentication

The system shall allow users to register and log in securely.

- Users can register using email, social media accounts, or institutional credentials.
- The system shall enforce **strong password policies** and multi-factor authentication (MFA).

3.2.2 F2: Profile Management

Users shall be able to manage their personal and professional profiles.

- Users can update their name, job details, education history, and contact information.
- Privacy settings allow customization of profile visibility.

3.2.3 F3: Alumni Networking & Messaging

The platform shall allow users to search, connect, and communicate with other alumni.

- Alumni can send connection requests and engage in private or group messaging.
- Users can filter connections based on industry, graduation year, and location.

3.2.4 F4: Job Portal

Alumni and companies can post and apply for job opportunities.

- The system provides job recommendations based on user profiles.
- Applicants can submit resumes and track application statuses.

3.2.5 F5: Events & Reunions

Users shall be able to view, register, and participate in alumni events.

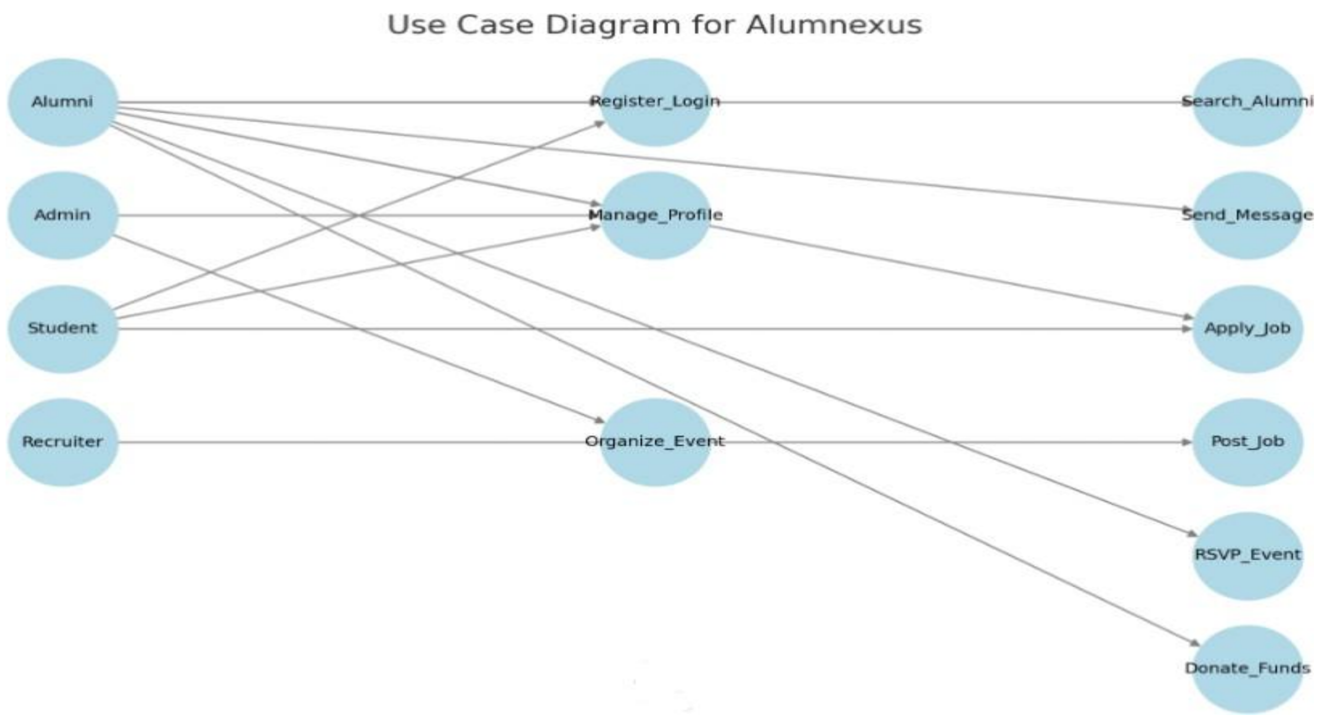
- Event organizers can create and manage event listings.
- Users receive event reminders and notifications.

3.2.6 F6: Donation & Fundraising

Alumni shall be able to contribute financially to their institution.

- Secure payment gateway for processing transactions.
- Donation tracking system displaying contribution history.

3.3 Use Case Model



Here is the **Use Case Diagram** for **Alumnexus**, illustrating system interactions between actors (**Alumni, Students, Recruiters, and Admin**) and core platform functionalities:

- **Alumni**: Can register, search for other alumni, send messages, apply for jobs, RSVP for events, and donate.
- **Students**: Can register, manage profiles, and apply for jobs.
- **Recruiters**: Can post jobs.
- **Admins**: Can manage profiles and organize events.

Use Case 1: User Registration & Authentication (U1)

- **Author:** Team 3
- **Purpose:** Enable users to register and log in securely.
- **Requirements Traceability:**
    - Secure registration/login with email, password, or social login.
    - Enforce multi-factor authentication (MFA).
    - Implement role-based access control (RBAC).
- **Priority: High –** Critical for accessing the platform.
- **Preconditions:** User must have a valid email and accept terms.
- **Postconditions:** Account is created, user is authenticated, and session is active.
- **Actors:** User (Alumni, Student, Faculty, Admin), Authentication Server.
- **Flow of Events:**
    1. User enters credentials and submits.
    2. System validates and sends a verification email.
    3. User verifies email, and account is activated.
    4. User logs in and is redirected to the dashboard.
- **Exceptions:**
    - Invalid email/password → Error message.
    - Unverified email → Resend verification option.
    - Too many failed attempts → Account lockout.

Use Case 2: Alumni Networking & Messaging (U2)

- **Author:** Team 3
- **Purpose:** Allow users to search, connect, and message other alumni.
- **Requirements Traceability:**
    - Search alumni by filters (year, industry, location).
    - Send/accept connection requests.
    - Direct messaging between connected users.
- **Priority: High –** Core feature for alumni engagement.
- **Preconditions:** User must be logged in and have a profile.
- **Postconditions:** New connection added; messages exchanged.
- **Actors:** User (Alumni, Student, Faculty), Database.
- **Flow of Events:**
    1. User searches for alumni and sends a connection request.
    2. Recipient accepts/rejects the request.
    3. If accepted, messaging is enabled.
- **Exceptions:**
    - Connection request denied → No further action.
    - User blocks messages → Messaging disabled.

Use Case 3: Job Posting & Application (U3)

- **Author:** Team 3
- **Purpose:** Allow recruiters and alumni to post jobs, and students to apply.
- **Requirements Traceability:**
  - Alumni/recruiters can post job listings.
  - Users can apply with resumes.
  - Recruiters can review and shortlist candidates.
- **Priority: High –** Essential for career networking.
- **Preconditions:** User must have an active profile.
- **Postconditions:** Job post is live, and applications are stored.
- **Actors:** User (Alumni, Student, Recruiter), Job Portal.
- **Flow of Events:**
  1. Recruiter creates a job post.
  2. Job is published and visible to users.
  3. Users apply with resumes.
  4. Recruiter reviews applications and selects candidates.
- **Exceptions:**
  - Job posting expired → Archived automatically.
  - Duplicate application → Error message.

4 Other Non-functional Requirements

4.1 Performance Requirements

The **Alumnexus** platform is designed to handle a large number of users while ensuring high- speed responsiveness. Below are the key performance requirements:

Performance	Requirement Category	Specification
P1	**System Response Time**	The system shall load the dashboard within 2 seconds for standard user requests under normal server load.
		Search queries (e.g., alumni directory, job postings) shall return results within 3 seconds**.**
P2	**Scalability & Concurrent Users**	The system shall support at least 10,000 concurrent users without significant performance degradation.
		The database shall handle up to 1 million user profiles efficiently.
P3	**Data Processing**	Profile updates, job postings, and event registrations shall be processed within 5 seconds.
		Batch processes (e.g., notifications, event reminders) shall complete within 1 minute for 10,000 users**.**
P4	**Availability & Downtime**	The system shall maintain 99.9% uptime, allowing for a maximum downtime of 8.76 hours per year.
		Regular maintenance windows will be scheduled outside peak usage hours to minimize disruption.
P5	**Database Performance & Query Optimization**	The system shall optimize database queries using indexing and caching techniques to ensure faster data retrieval.
		Heavy queries (e.g., searching alumni across multiple filters) shall execute within 5 seconds under peak load**.**
P6	**Job & Event Processing Speed**	A job posting shall be published within 2 seconds after submission.
		Event registrations shall reflect instantly in the system without delays.
P7	**Notification System Latency**	In-app and email notifications shall be delivered within 10 seconds of an event trigger (e.g., new message, event reminder).
P8	**Load Handling & Auto-Scaling**	The system shall use cloud based autoscaling mechanisms to handle traffic spikes (e.g., during major alumni reunions or job fairs).
		Server response time shall not exceed 3 seconds under peak load conditions.

4.2 Safety and Security Requirements

The **Alumnexus** platform deals with **sensitive alumni data**, requiring strong security measures.

S1: User Authentication & Access Control

- The system **shall enforce multi-factor authentication (MFA)** for all user logins.
- User roles (alumni, students, faculty, admin) **shall have role-based access controls (RBAC)** to limit unauthorized actions.

S2: Data Encryption

- All personal user data **shall be encrypted using AES-256 encryption** in the database.
- All communication between the user and the server **shall be secured using TLS 1.2+ encryption**.

S3: Privacy Compliance

- The system **shall comply with data protection regulations** (e.g., **GDPR, CCPA**) to safeguard user information.
- Users **shall have the ability to manage their privacy settings**, including visibility of their profiles.

S4: Security Against Threats

- The system **shall implement rate limiting** to prevent **DDoS (Distributed Denial-of-Service) attacks**.
- **Regular penetration testing** shall be conducted to identify and fix security vulnerabilities.
- The system **shall automatically log out inactive users after 15 minutes** for security reasons.

S5: Backup & Disaster Recovery

- The system **shall perform daily backups** of all critical data.
- In case of failure, **data restoration shall occur within 1 hour** to minimize service disruption.

S6: Data Retention and Deletion Policies

- The system **shall allow users to request account deletion**, removing all personal data within **30 days** of approval.
- Backup retention shall follow a **30-day rolling window** before permanent deletion.

S7: Secure Payment Transactions

- All financial transactions (donations, event ticketing) **shall be processed through PCI DSS-compliant payment gateways**.
- **Fraud detection mechanisms** (e.g., anomaly detection on payment attempts) shall be in place.

S8: Audit Logging & Monitoring

- The system **shall log all administrative and sensitive actions (e.g., data updates, donations) for audit purposes**.
- **Security monitoring tools** shall detect and report suspicious activities in real time.

4.3 Software Quality Attributes

4.3.1 Reliability

The system shall operate continuously with minimal failures, ensuring 99.9% uptime.

Implementation Approach:

- Load balancers and failover mechanisms will ensure redundancy.
- Automated monitoring tools will detect server crashes and downtime, triggering instant alerts.

4.3.2 Adaptability

The system shall be adaptable to future enhancements, such as integration with new alumni engagement tools or career services.

Implementation Approach:

- The modular architecture will allow easy updates to job portals, donation systems, and event management modules.
- API-based design will enable seamless integration with third-party services (e.g., LinkedIn, HR portals).

4.3.3 Maintainability

The system shall be easy to maintain and update with minimal downtime**.**

Implementation Approach:

- Microservices-based architecture will allow independent updates without affecting the entire system.
- Comprehensive documentation and automated testing will ensure smooth maintenance.

4.3.4 Usability

The system shall be user-friendly, allowing new users to navigate the platform with minimal training.

Implementation Approach:

- A clean, intuitive UI design with clear menus and search functions will enhance usability.
- Tooltips and FAQs will provide instant user assistance.

4.3.5 Scalability

The system shall scale dynamically based on user load to maintain optimal performance.

Implementation approach:

- Horizontal scaling (adding more servers) will be used to accommodate high traffic.
- CDN (Content Delivery Network) will distribute static content to reduce server load.

4.3.6 Testability

The system shall be thoroughly tested using automated and manual testing techniques.

Implementation approach:

- Unit, integration, and UI tests will cover at least 90% of the codebase.
- Continuous Integration/Continuous Deployment (CI/CD pipelines) will enable automated testing and deployment.