SOFTWARE REQUIREMENT SPECIFICATIONS

****

**ALUMNI NETWORK PORTAL**

**Pranav Mudigonda AP22110010765**

**Yuva Sri Sai Thota AP22110010750**

**Bhavya Sree Kasa AP22110010770**

**Mohan Satya Venkat Kommana AP22110010745**

## TABLE OF CONTENTS

### Introduction

* 1. Purpose
  2. Scope
  3. Definitions, Acronyms, and Abbreviations
  4. References
  5. Overview

### Overall description

* 1. Product perspective
  2. Product Functions
  3. User Characteristics
  4. Constraints
  5. Assumptions and Dependencies

### Specific Requirements

### External Interfaces

### Use Case Diagram

### Functional Requirements

### Non-Functional Requirements

### Logical Database Requirements

### Design Constraints

### Software System Quality Attributes

1. **Appendices**

**1. Introduction**

**1.1 Purpose**

The **Alumni Portal** is a comprehensive web-based platform designed to enhance the connection between alumni and the college while fostering professional networking, communication, and collaboration. This system provides a centralized hub where alumni and students can interact through features such as secure user authentication, an alumni directory, event management, discussion forums, a messaging system, an alumni dashboard, and profile customization.

**Use-**The primary use of this project is to simplify alumni engagement, enabling former students to stay connected with their peers, participate in discussions, share job opportunities, and attend events organized by the college.

**Importance of the Project**

* Helps alumni and students stay connected.
* Makes event and job opportunities easily accessible.
* Encourages networking and mentorship.
* Ensures secure and organized communication.

**1.2 Scope**

The **Alumni Portal** aims to streamline alumni management and engagement through the following key features:

* + 1. **User Authentication**
  + Secure login via OAuth Authentication (Google).
  + JWT-based authentication for session management.
  + Role-based access control for Admin, Alumni, and Students.

**1.2.2 Alumni Directory**

* + A searchable database of alumni with professional details.
  + Editable profiles with admin verification for authenticity.
  + LinkedIn & Email Integration for direct communication.

**1.2.3 Event Management**

* + Alumni and admins can post and manage events.
  + RSVP system for tracking participation.
  + Notifications & reminders for upcoming events.
  + Scheduled event announcements for better engagement.

**1.2.4 Discussion Forums**

* + Topic-based discussion threads for alumni and students.
  + Users can create, comment, and like posts to encourage interaction.
  + Real-time updates & moderation tools to maintain content quality.

**1.2.5 Messaging System**

* + Real-time chat between alumni and students using Socket.IO.
  + Group chats for event discussions and mentorship.
  + Admin moderation to ensure secure and meaningful conversations.

**1.2.6 Alumni Dashboard**

* + Admin dashboard to monitor user engagement, job postings, and event participation.
  + Visual analytics using charts & graphs for better insights.
  + Tracking active users, top discussions, and mentorship participation.

**1.2.7 Profile Customization**

* + Alumni can update their profiles with achievements and social media links.
  + Profile verification system for credibility and authenticity.
  + Featured alumni stories & testimonials to highlight successful alumni.

**1.3 Definitions, Acronyms, and Abbreviations**

* **Admin** – Users responsible for data maintenance and event management.
* **Students** – Users who can connect with alumni and view event updates.
* **Alumni** – Graduates who engage with the network and mentor students.
* **Firebase** – Backend service for authentication and database storage.
* **MongoDB** – NoSQL database used for storing alumni-related data.
* **Vite** – Frontend build tool used for the React application.
* **UI/UX** – User Interface/User Experience design aspects of the application.
* **Scalability** – The ability of the application to support a growing user base without performance degradation.
* **Mentorship Program** – A structured initiative connecting students with alumni mentors.
  1. **References**
* Gate Smashers. (2022, April 30). *Software Requirements Specification (SRS) | Software Engineering* [Video]. YouTube. <https://www.youtube.com/watch?v=83-S5Qu6VP8>
* GeeksforGeeks. (2023, September 20). *Software Requirement Specification (SRS) Format*. GeeksforGeeks.

<https://www.geeksforgeeks.org/software-requirement-specification-srs-format/>

* GeeksforGeeks. (2024, September 13). Hospital Management System project in Software development. GeeksforGeeks.

[https://www.geeksforgeeks.org/hospital-management-system-project-in-software-deve](https://www.geeksforgeeks.org/hospital-management-system-project-in-software-development/) [lopment/](https://www.geeksforgeeks.org/hospital-management-system-project-in-software-development/)

* Lane, G. K. (2023, January 17). *How to write an SRS Document (Software Requirements Specification Document)*. Perforce Software.

[https://www.perforce.com/blog/alm/how-write-software-requirements-specification-sr](https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document) [s-document](https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document)

* Existing College Website <https://alumni.srmap.edu.in/>
* Firebase Documentation, MongoDB Documentation, React and Vite Documentation, Web Development Best Practices

**1.5 Overview**

The Alumni Portal is a web-based platform designed to connect alumni with the college and students. It helps alumni stay engaged, share opportunities, and participate in events, discussions, and mentorship programs. The portal includes secure user authentication, an alumni directory, event management, discussion forums, a messaging system, an alumni dashboard, and profile customization. This system makes communication easy, strengthens alumni relationships, and provides a space for networking, career opportunities, and knowledge sharing.

**2. Overall Description**

**2.1 Product Perspective**

This project was created to solve the problem of weak alumni connections in colleges. Many colleges struggle to keep track of alumni, organize events, or provide a proper space for discussions and mentorship. Without a dedicated system, students miss out on valuable career advice, networking, and job opportunities.

The Alumni Portal fixes this by offering a simple and efficient platform where alumni and students can connect easily. It helps manage alumni data, send notifications, organize events, and enable real-time communication, making alumni relations stronger and more effective.

**2.2 Product Functions**

The Alumni Portal is designed to strengthen the connection between alumni, students, and the college. It provides a platform for networking, communication, and event management. The key functions include:

1. **User Authentication** – Secure login with Google and role-based access for Admin, Alumni, and Students.
2. **Alumni Directory** – A searchable database of alumni with contact details and LinkedIn integration.
3. **Event Management** – Allows alumni and admins to create, manage, and send reminders for events.
4. **Discussion Forums** – Provides a space for alumni and students to engage in topic-based discussions.
5. **Messaging System** – Enables real-time chat between alumni and students, with group chat support.
6. **Alumni Dashboard** – Admins can manage alumni records, events, and discussions.
7. **Profile Customization** – Alumni can update their profiles, add achievements, and connect social media accounts.

**2.3 User Characteristics**

* **Admin:** Requires basic data management skills, responsible for maintaining alumni records and events.
* **Students:** Users with moderate technical skills, able to navigate the platform to connect with alumni and explore opportunities.
* **Alumni:** Graduates who register to provide mentorship, job opportunities, and engage with students.

**2.4 Constraints**

* Must support moderate response times under expected user loads.
* Must be scalable for future expansions.
* Should operate within the existing college infrastructure.
* UI/UX should be optimized for accessibility and ease of navigation.
* Data privacy regulations must be considered for future compliance.

**2.5 Assumptions and Dependencies**

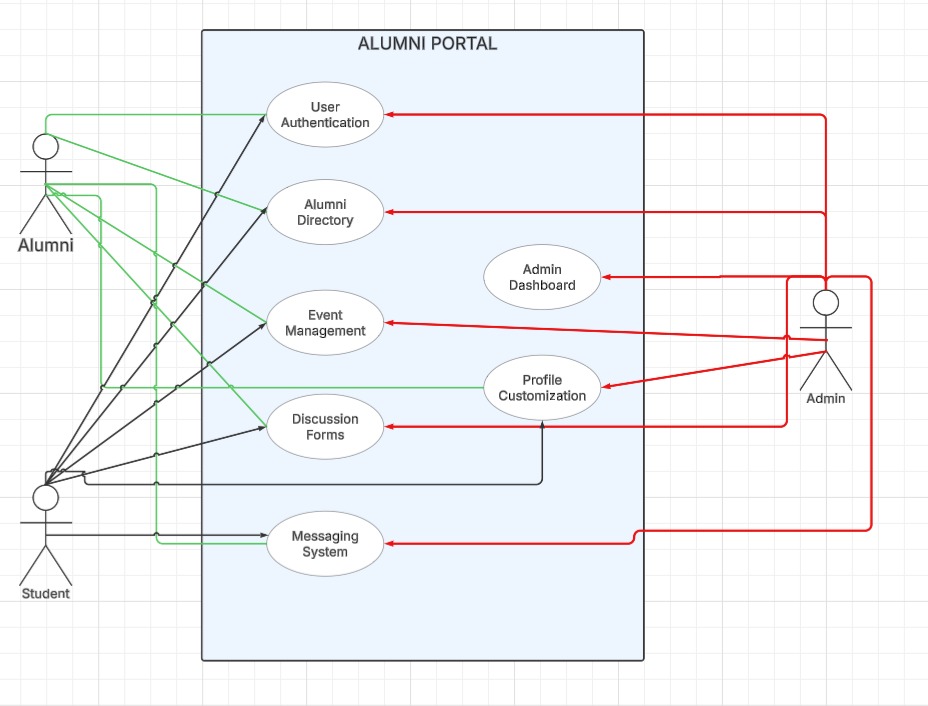
* No legal compliance requirements at present.
* Future integrations may include third-party APIs such as LinkedIn for professional networking.
* The system should be cloud-hosted to ensure high availability and reliability.
* Regular updates and feature expansions will be planned post-deployment.

**3. Specific Requirements**

**3.1 External Interfaces**

* **User Interface:** React-based frontend with intuitive navigation and responsive design.
* **Database:** MongoDB for storing user profiles, events, and job listings.
* **Authentication:** Firebase Google login integration for secure access.
* **Communication:** Built-in messaging system for interactions between students and alumni.

**3.2 USE CASE DIAGRAM**



**3.3 Functional requirements**

**F-001: User Authentication**

* Users can log in securely using Google OAuth or a traditional email/password system.
* Role-based authentication ensures that Admins, Alumni, and Students have different access levels.
* Protects user data with JWT-based session management for security.

**F-002: Alumni Directory**

* A searchable database where students and alumni can find and connect with each other.
* Displays professional details, such as job roles, industries, and locations.
* Allows direct contact via LinkedIn or email (if enabled by the alumni).
* Admin verification ensures data accuracy and prevents unauthorized changes.

**F-003: Event Management**

* Alumni and admins can create, manage, and track events like reunions, networking sessions, or webinars.
* Users can RSVP for events and receive automatic reminders via email or notifications.
* Admins can schedule event announcements and track attendee participation.

**F-004: Discussion Forums**

* Users can create topic-based discussions to share knowledge, experiences, and career advice.
* Features include posting, commenting, and liking discussions.
* Alumni and students can form interest groups (e.g., AI, Cybersecurity, Startups) for industry-specific discussions.
* Admins have moderation tools to ensure healthy and professional discussions.

**F-005: Messaging System**

* Real-time chat allows direct communication between alumni and students using Socket.IO.
* Group chats can be created for event discussions, mentorship, or interest-based conversations.
* Admins can monitor messages to prevent misuse or spam.
* Chat history is stored securely for better user experience.

**F-006: Alumni Dashboard**

* A personalized dashboard where users can see:
  + Recent events and discussions they are involved in.
  + Pending mentorship requests (if applicable).
  + Job postings relevant to their industry.
  + Notifications for important updates.
* Helps alumni and students stay engaged and updated.

**F-007: Profile Customization**

* Users can edit their profiles, add achievements, work experience, and connect social media links (LinkedIn, GitHub, etc.).
* A profile verification system enhances credibility by ensuring only real alumni can update their details.
* Featured Alumni Spotlights showcase success stories and career growth.
* Alumni Testimonials allow users to share their experiences with the institution.
  1. **Non-Functional Requirements**

**N-001: Performance**

* The system should work fast and respond quickly when users search for alumni, send messages, or check events.
* It should handle many users at the same time without slowing down.
* Data should load quickly by using smart techniques like caching.

**N-002: Scalability**

* The system should support more users as the alumni network grows.
* It should allow adding more servers if needed to keep it running smoothly.
* It should be designed for future expansion without big changes.

**N-003: Security**

* Users should log in securely with Google OAuth and encrypted passwords.
* Only authorized users should be able to access or edit certain features.
* Important data like emails and messages should be protected from hackers.
* The system should get regular security updates to stay safe.

**N-004: Usability**

* The platform should be easy to use, even for new users.
* It should work well on both computers and mobile phones.
* Users should easily find what they need using search bars and filters.
* It should be accessible for everyone, including differently-abled users.

**N-005: Availability**

* The system should be available 24/7, so users can log in anytime.
* It should have backup servers to avoid downtime.
* If something goes wrong, data backups should help restore the system quickly.

**N-006: Maintainability**

* The code should be organized and easy to update when needed.
* Bug fixes and new features should be added without breaking the system.
* Developers should use version control (like Git) to track changes.
* The system should be flexible to add new features in the future.

**3.5 Logical Database Requirements**

* Alumni profiles with professional details and contact information.
* Event records with descriptions, dates, and locations.
* User authentication records with secure credential storage.
* Job postings and applications data.
* Mentorship program logs tracking student-alumni interactions.
* Engagement analytics tracking user activity and interactions.
  1. **Design Constraints**
* C-001 Developed using React, Firebase, and MongoDB.
* C-002 UI optimized for scalability, usability, and accessibility.
* C-003 Light, professional design with an intuitive user experience.
* C-004 Mobile-friendly layout to ensure compatibility with various devices.
* C-005 Modular architecture to facilitate future feature additions.

**3.7 Software System Quality Attributes**

* **Scalability:** Designed to support a growing user base and additional features.
* **Usability:** Ensuring easy navigation with an intuitive design.
* **Security:** Secure authentication and data protection measures.
* **Performance:** Low response times and efficient handling of concurrent users.
* **Maintainability:** Clean code architecture for easy future updates.
* **Accessibility:** Compliance with web accessibility standards for inclusivity.
* **Reliability:** Ensuring data consistency and backup mechanisms.
* **Interoperability:** Ability to integrate with external job portals and professional networking platforms.

**4. Appendices**

* UI Wireframes (To be added later)
* API Documentation (To be added later)
* Sample Database Schema
* Performance Testing Reports