Software Requirements Specification

for

ICS ARTMS

Version 0.2

Prepared by CMSC128 - A1-2L

University of the Philippines - Los Baños

March 04, 2025

Table of Contents

Revision History

Name	Date	Reason for Changes	Version
Audije, Balede, Borja, Fernandez, Gregorio, Mende, Ramos, Reyes, Panem, Villena	2025-03-02	Initial Product Features	0.1
Panem	2025-03-11	Updated Stimulus and Response, added routing diagram	0.2

1. Introduction

1.1 Purpose

The goal of this document is to introduce the ICS Alumni Tracker and Relations Management System is designed to provide a centralized platform for managing and engaging with the alumni of the Institute of Computer Science (ICS) at the University of the Philippines - Los Baños (UPLB). The system will enable ICS to efficiently maintain alumni records, track their achievements, organize events, and facilitate career networking and fundraising efforts.

Currently, ICS lacks a structured system for alumni relations, making it difficult to organize reunions, fundraising initiatives, and career-related activities. This system aims to address these challenges by offering features such as profile management, job postings, event management, alumni search, donations, and a communication module.

The system will also generate analytical reports to monitor alumni engagement, employment distribution, and donation histories, allowing ICS to foster stronger relationships with its graduates.

1.2 Document Conventions

- The title page of this document is written in **Trebuchet** font, size **32**, with subtitles in size **14**.
- Section headers are written in **bold Trebuchet**, with main headers at size **18** and subheaders at size **14**.
- The appendix is formatted in **Trebuchet, size 10**.
- All other content follows **Trebuchet, size 12**, for readability and consistency.

1.3 Intended Audience and Reading Suggestions

This document is intended for the following stakeholders:

- **ICS Administrators**: To oversee alumni engagement and data management.
- **System Developers**: To implement the features outlined in this document.
- **ICS Alumni**: To understand how they can interact with the system.

The document provides an overview of the system's scope (Section 1.4), key functionalities (Section 2), and detailed technical specifications (Section 3). Developers should focus on Section 3 for implementation details, while administrators and alumni can refer to Sections 2 and 4 for user functionalities and privileges.

1.4 Project Scope

The ICS Alumni Tracker and Relations Management System is designed to:

- **Allow Alumni to Register and Manage Their Profiles**: Alumni can create, update, and maintain their personal, educational, and career records.
- **Facilitate Job Postings and Career Opportunities**: Alumni and companies can share job openings for students and fellow graduates.
- **Enable Event Organization and RSVP Management**: ICS can organize reunions, webinars, and networking events, allowing alumni to RSVP.
- **Provide an Alumni Search Feature**: Users can filter alumni based on graduation year, field of work, location, or skills.
- **Support Donations and Sponsorships**: The system will allow alumni to contribute to scholarships or events.
- **Include a Communications Module**: ICS can send announcements, newsletters, and invitations.
- **Generate Reports and Analytics**: Summarized engagement statistics, alumni employment trends, and donation histories.

The system is **web-based** and will be accessible via a modern web browser. It is intended for ICS alumni and ICS staff, with role-based access privileges. The project does not include third-party integrations for external social media or job platforms at this stage.

1.5 References

This document follows the Software Requirements Specification (SRS) guidelines outlined by Karl E. Wiegers. It also references previous CMSC 128 software projects to maintain compliance with academic and software engineering best practices.

2. Overall Description

2.1 Product Perspective

The ICS Alumni Tracker and Relations Management System is an independent web-based system developed for the ICS community. It is a self-contained application designed to enhance alumni engagement and streamline data management for ICS staff.

2.2 Product Features

The system will provide the following features:

- User Registration and Profile Management
- Job Posting and Career Opportunities
- Event Management and RSVP Tracking
- Alumni Search and Filtering
- Donations and Sponsorship Handling
- Communications Module for Announcements and Newsletters
- Reports and Analytics Dashboard

2.3 User Classes and Characteristics

1. **Alumni**: Can create and manage profiles, search for fellow alumni, post job opportunities, and RSVP to events.
2. **Admin (ICS Staff)**: Can manage alumni records, organize events, send communications, and monitor alumni engagement metrics.

2.4 Operating Environment

- **Back-End**: NextJS
- **Front-End**: Tailwind with DaisyUI
- **Supported Browsers**: Google Chrome, Mozilla Firefox, Microsoft Edge

2.5 Design and Implementation Constraints

The design and implementation of the ICS Alumni Tracker and Relations Management System will adhere to the ICS academic calendar to ensure timely project completion. As the system is web-based, continuous internet connectivity is required for access and proper functionality. Additionally, the system must comply with institutional data privacy policies, limiting access to sensitive alumni information only to authorized users. These constraints guide the system's development, ensuring security, reliability, and usability within the given timeframe.

2.6 User Documentation

A user manual and system documentation will be provided upon project completion.

2.7 Assumptions and Dependencies

The development and functionality of the ICS Alumni Tracker and Relations Management System rely on several key assumptions and dependencies. These factors ensure the system operates effectively and meets the needs of both ICS administrators and alumni users.

- Users must provide accurate and up-to-date information.
- The system will rely on a secure database for storing alumni records.
- The application is optimized for desktop usage but may be accessed on mobile devices.

3. System Features

3.1 Administrator Creation

3.1.1 Description and Priority

This feature allows the creation of an administrator account to manage users and system functionalities.
Priority: High

3.1.2 Stimulus/Response Sequence

- The assigned administrator clicks the "Sign Up" hyperlink and is redirected to the "Create Admin Account" page.
- The administrator enters their first and last name, email, password, and a reconfirmation of their password in the text fields provided.
- The administrator clicks the "Sign Up" button.
- Upon successful sign-up, a message appears below the text fields stating that the user was saved and the user is redirected to the "Login" page.
- Upon failure to sign up, a series of warning messages appear below the text field of interest.

- The administrator may choose to click the cancel button. This redirects them to the “Landing” page.

3.1.3 Functional Requirements

- **AC1** - Form Validation to check for the following:
    - All input fields must be filled.
    - Email must be unique.
    - Password must meet formatting and security requirements.
- **AC2** - The System must display a confirmation message upon successful sign-up.

3.2 User Registration

3.2.1 Description and Priority

Allows an administrator to create new user accounts.
Priority: High

3.2.2 Stimulus/Response Sequence

- User navigates to "Sign Up".
- User enters personal details (name, email, password, etc.).
- System validates input and creates the account.
- System sends a confirmation email.

3.2.3 Functional Requirements

- **UR1:** Form Validation Checks for the following:
    - All input fields must be filled.
    - Email must be unique.
    - Password must meet formatting and security requirements.
- **UR2:** The system must display a confirmation message upon successful user creation.

3.3 View Users

3.2.1 Description and Priority

Allows an administrator to view all accounts with options.
Priority: High

3.2.2 Stimulus/Response Sequence

- The administrator clicks the "Sidebar" icon in the top-left corner of the screen.
- The administrator clicks the “Manage Users” button.
- The administrator is navigated to the “Manage Users” page.
- Searches and filters.
- Admin can delete or promote users.

3.2.3 Functional Requirements

- **VU1:** The system must return a list of users.
- **VU2:** The system must allow filtration.
- **VU3:** Admins can delete or promote users.
- **VU4:** Only admins can view all users.

3.4 Delete Users

3.4.1 Description and Priority

Allows an administrator to remove a user from the system.
Priority: High

3.4.2 Stimulus/Response Sequence

- The administrator clicks the "Sidebar" icon in the top-left corner of the screen.
- The administrator clicks the “Manage Users” button.
- The administrator is navigated to the “Manage Users” page.
- In the “Manage Users” page, the administrator clicks the delete icon on the row of the existing regular user to be deleted.
- Upon deletion, the “Manage Users” page will automatically be refreshed.

3.4.3 Functional Requirements

- **DU1:** The system must verify the user’s role (administrator or regular user) before accessing the “Manage Users” page.
- **DU2:** The system must refresh the page to reflect the updated user list after a deletion.
- **DU3:** The system must not allow an administrator to delete themselves.

- **DU4:** The system must remove all associated data when a user is deleted.

3.5 Promote User

3.5.1 Description and Priority

Allows an administrator to promote a regular user to admin.
Priority: Medium

3.5.2 Stimulus/Response Sequence

- The administrator clicks the "Sidebar" icon in the top-left corner of the screen.
- The administrator clicks the “Manage Users” button.
- The administrator is navigated to the “Manage Users” page.
- In the “Manage Users” page, the administrator clicks the promote icon on the row of the existing regular user to be promoted.
- The administrator selects "Promote User".
- The system updates the user role.
- A notification is sent to the user.

3.5.3 Functional Requirements

- **PU1:** The system must allow only administrators to promote users.
- **PU2:** The system must notify the promoted user of their new role.

3.6 User Authentication

3.6.1 Description and Priority

Ensures secure login for users.
Priority: High

3.6.2 Stimulus/Response Sequence

- User enters an email and password.
- System verifies credentials.
- User gains access upon successful authentication.

3.6.3 Functional Requirements

- **UA1:** The system must store encrypted passwords.

- **UA2:** The system must verify email and password credentials during login.
- **UA3:** The system must provide an option for password recovery in case of lockout.

3.7 Edit User Profile

3.7.1 Description and Priority

Allows users to update their profile information.
Priority: Low

3.7.2 Stimulus/Response Sequence

- The user clicks the "Sidebar" icon in the top-left corner of the screen.
- The user clicks the “Profile” button.
- The user is navigated to the “Profile” page.
- In the “Profile” page, the user can change their information
- User saves changes
- Page refreshes to reflect change

3.7.3 Functional Requirements

- **EP1:** The system must prevent users from changing their email address.
- **EP2:** The system must validate changes before saving them.
- **EP3:** The system notifies the user of a successful edit.

3.8 Change Account Password

3.8.1 Description and Priority

Enables users to update their passwords.
Priority: High

3.8.2 Stimulus/Response Sequence

- The user clicks the "Sidebar" icon in the top-left corner of the screen.
- The user clicks the “Profile” button.
- The user is navigated to the “Profile” page.
- In the “Profile” page, the user can change their password
- User saves changes
- Page refreshes to reflect change

3.8.3 Functional Requirements

- **CP1:** The system must enforce password complexity requirements.
- **CP2:** The system must prevent reuse of previous passwords.
- **CP3:** The system notifies the user of a successful password change.

3.9 Create Job Post

3.9.1 Description and Priority

Alumni can create job listings.
Priority: High

3.9.2 Stimulus/Response Sequence

- The user clicks the "Sidebar" icon in the top-left corner of the screen.
- The user clicks the “Job Listings” button.
- The user is navigated to the “Job Listings” page.
- In the “Job Listings” page, the user can click the “Post a Job” button
- The user fills out the information in the modal
- The user presses the submit button
- The page is refreshed to reflect changes

3.9.3 Functional Requirements

- **JP1:** The system must require a job title, description, and contact details before posting a job.
- **JP2:** The system must notify user of a successful job posting.

3.10 View Job Posts

3.10.1 Description and Priority

Users can browse job listings.
Priority: High

3.10.2 Stimulus/Response Sequence

- The user clicks the "Sidebar" icon in the top-left corner of the screen.
- The user clicks the “Job Listings” button.
- The user is navigated to the “Job Listings” page.
- The user can click on job listings to view more information about the job

3.10.3 Functional Requirements

- **VJ1:** The system must return a list of available jobs.
- **VJ2:** The system must allow users to search and filter job listings.
- **VJ3:** The user must be able to view the details of a job.

3.11 Edit Job Post

3.11.1 Description and Priority

Alumni can update their job posts.
Priority: Medium

3.11.2 Stimulus/Response Sequence

- The user clicks the "Sidebar" icon in the top-left corner of the screen.
- The user clicks the “Job Listings” button.
- The user is navigated to the “Job Listings” page.
- The user clicks the “View my job listings” button
- The page will show the job listings created by the user
- The user can click the edit button
- The user can change the information in the modal
- The user presses the submit button
- The page is refreshed to reflect changes

3.11.3 Functional Requirements

- **EJ1:** The system must restrict editing permissions to the job post creator.
- **EJ2:** The system must notify the user of successful edit.

3.12 Delete Job Post

3.12.1 Description and Priority

Alumni can remove job listings.
Priority: Medium

3.12.2 Stimulus/Response Sequence

- The user clicks the "Sidebar" icon in the top-left corner of the screen.
- The user clicks the “Job Listings” button.

- The user is navigated to the “Job Listings” page.
- The user clicks the “View my job listings” button
- The page will show the job listings created by the user
- The user can click the delete button
- The page is refreshed to reflect the changes

3.12.3 Functional Requirements

- **DJ1**: Only the job creator can delete their own job posts.
- **DJ2**: The system must confirm the deletion action before removing the job post.

3.13 Create Event

3.13.1 Description and Priority

Admins can create events for alumni engagement.
Priority: High

3.13.2 Stimulus/Response Sequence

- The administrator clicks the "Sidebar" icon in the top-left corner of the screen.
- The administrator clicks the “Events” button.
- The administrator is navigated to the “Events” page.
- The administrator clicks the “New Event” button
- The administrator fills out the modal with the information of the event
- The administrator clicks the submit button
- The page is refreshed to reflect changes

3.13.3 Functional Requirements

- **CE1**: Events must have a required name, date, and location fields.
- **CE2**: Admins must be able to set RSVP options for events.
- **CE3** - Admins can enable or disable sponsorship requests when creating events.
- **CE4** - The system should display sponsorship requests in the event details.
- **CE5**: System should allow optional details like event description and contact information.

3.14 View Events

3.14.1 Description and Priority

Alumni can RSVP to events.
Priority: High

3.14.2 Stimulus/Response Sequence

- The user clicks the "Sidebar" icon in the top-left corner of the screen.
- The user clicks the “Events” button.
- The user is navigated to the “Events” page.
- The user can filter and search events
- The user can click on event to respond or to learn more about the event

3.14.3 Functional Requirements

- **VE1** - RSVP status must be updatable.
- **VE2** - Admins can view attendee lists.
- **VE3** - Alumni can pledge financial or material sponsorships.
- **VE4** - Sponsorship offers should be recorded and viewable by admins.
- **VE5** - Admins must be able to approve or reject sponsorship offers.

3.15 Edit Event

3.15.1 Description and Priority

Admins can update event details.
Priority: High

3.15.2 Stimulus/Response Sequence

- The administrator clicks the "Sidebar" icon in the top-left corner of the screen.
- The administrator clicks the “Events” button.
- The administrator is navigated to the “Events” page.
- The administrator can click the “Edit” button of an event to open an edit modal
- The administrator edits information and presses the save button
- The page is refreshed to reflect the changes

3.15.3 Functional Requirements

- **EE1:** System should prevent editing of past events.
- **EE2**: System notifies the user of a successful edit.

- **EE3:** Users that have responded to the event must be notified.

3.16 Delete Event

3.16.1 Description and Priority

Admins can remove events.
Priority: Medium

3.16.2 Stimulus/Response Sequence

- The administrator clicks the "Sidebar" icon in the top-left corner of the screen.
- The administrator clicks the “Events” button.
- The administrator is navigated to the “Events” page.
- The administrator can click the “Delete” button of an event to delete the event
- After confirmation, the page refreshes to reflect the changes

3.16.3 Functional Requirements

- **DE1:** System should notify RSVP’d users when an event is deleted

3.17 View Alumnus and Filter

3.17.1 Description and Priority

Users can search for alumni.
Priority: High

3.17.2 Stimulus/Response Sequence

- The user clicks the "Sidebar" icon in the top-left corner of the screen.
- The user clicks the “Alumni” button.
- The user is navigated to the “Alumni” page.
- The user can search for alumni or use filters
- Hovering over a profile shows more information about the alumnus

3.17.3 Functional Requirements

- **VA1**: Alumni profiles must be searchable by multiple criteria (e.g., name, graduation year, field, location).
- **VA2**: Filters must allow combination searches for better precision.

- **VA3**: Search results should be paginated for performance optimization.

3.18 Send Announcements

3.18.1 Description and Priority

Admins can send messages to alumni.
Priority: High

3.18.2 Stimulus/Response Sequence

- The administrator clicks the "Sidebar" icon in the top-left corner of the screen.
- The administrator clicks the “Communications” button.
- The administrator is navigated to the “Communications” page.
- The administrator can compose an announcement that will be sent to all mailing lists

3.18.3 Functional Requirements

3.19.3 Functional Requirements

- **SA1**: System must allow scheduling of announcements for future delivery.
- **SA2**: System must track delivery status of announcements.

3.19 Logs

3.19.1 Description and Priority

Tracks system activities.
Priority: Low

3.19.2 Stimulus/Response Sequence

- The administrator clicks the "Sidebar" icon in the top-left corner of the screen.
- The administrator clicks the “Logs” button.
- The administrator is navigated to the “Logs” page.
- The administrator can view system logs

3.19.3 Functional Requirements

- **LG1:** The system must record timestamp, user, and action for each log entry.
- **LG2:** The system must allow only administrators to delete logs.

3.20 About

3.20.1 Description and Priority

Displays project and developer details.
Priority: Low

3.20.2 Stimulus/Response Sequence

- User navigates to "About".
- System displays project information.

3.20.3 Functional Requirements

- **AB1:** The system must include team members and contact details on the "About" page.

3.21 Reports Summary

3.21.1 Description and Priority

Admins can generate reports for alumni engagement.
Priority: High

3.21.2 Stimulus/Response Sequence

- The administrator clicks the "Sidebar" icon in the top-left corner of the screen.
- The administrator clicks the “Communications” button.
- The administrator is navigated to the “Communications” page.
- The administrator can view different statistics of the system
- The administrator can generate reports

3.20.3 Functional Requirements

- **RS1:** Filters could be used to limit the report based on certain parameters (year, event).

- **RS2:** Reports must be downloadable in multiple formats (PDF, CSV, Excel).
- **RS3:** Only authorized admins can generate reports.
- **RS4:** Reports should be stored for future reference and analysis.
- **RS5:** Provide a dashboard that contains interactive visualizations (charts, and graphs) on the trends in the number of active alumni, their current industries, and donation history.

4. External Interface Requirements

4.1 User Interfaces

The user interface of the system was built using NextJS on the frontend with TypeScript for development, along with the usage of Tailwind CSS and DaisyUI plugins for creating an interactive and responsive user interface for the web.

For best compatibility, the user of the system can use any modern web browser from 2025, such as Google Chrome, Microsoft Edge, Mozilla Firefox, and Apple Safari, to access the user interface of the system. Although the website is responsive and reactive, it is also recommended that the user views the web application on a monitor with a resolution of at least 1440x1024 pixels

Component	Description
Alumni Registration Page	Contains the digital form to fill out and submit in order to create an account.
Admin Registration Page	Contains the digital form to fill out in order to create an admin account.
Admin Dashboard Page	The page an admin is directed to after successful login containing recent announcements.
Admin User Management Page	A page accessed by the admin that contains a list of users. Clicking on the user gives a modal of their details. Action buttons are provided to allow users to be deleted and promoted.
Alumni Login Page	Contains the input fields for the alumni to input login details.
Admin Login Page	Contains the input fields for an admin to input login details.

Alumni Dashboard Page	The page an alumni is directed to upon successful login.
Alumni Profile Page	The alumni’s view of their own public info with a button to edit their profile information and password.
Job Listings Page	A page containing a list of available job opportunities, where users can filter and search for specific jobs. Clicking on a specific job opens a modal with its details. There is also an option to add a job listing which will open a modal that asks for job listing details.
My Job Listings	A page similar to the job listings page, but only contains listings by the user. Has options to edit and delete listings.
Event Listings Page	A page that shows a list of upcoming events with the event name, date, and description, where users can RSVP. Clicking on an event gives users more details. Successful response to an event notifies users. Event details include options for donations and sponsorships. Admins have options to edit and delete events they have posted. Admins have the extra functionality of adding, editing, and deleting events.
Alumni Search Page	A page designed to find fellow graduates of an alumni using their name with filtering options for graduation year, field of work, location, and skills. Clicking on an entry gives a modal giving more details about the alumni
Donation and Sponsorships Page	A page where information about donating and sponsoring is displayed. Details on different organizations or causes would also be listed here.
Email Newsletter Page (Admin)	A page where admins can compose and send email newsletters to alumni.

Notification Center	A centralized hub displaying system alerts, announcements, and updates for users.
Reports Dashboard (Admin)	A dashboard where administrators can generate and view reports on alumni engagement, job postings, event participation, donations, and other statistics.
Home/Landing Page	A page that users may first land on as they open the website.
About Page	A page that provides contact information about the admins and information about the developers.
Privacy Policy Page	A page that provides an explanation on what information the system will collect and how it will be managed.
Error Pages	Display the error in case a user has a failed attempt when accessing a page.

4.2 Hardware Interfaces

Client Devices:

1. **General Requirements:**

Devices must support modern web browsers (Chrome, Firefox, Edge, Brave, Safari) compatible with HTML5, CSS3, and JavaScript.

2. **Recommended Client Specifications:**

Desktop/Laptop: 2 GHz dual-core processor, 4 GB RAM
Tablet/Smartphone: Devices with comparable performance to ensure a responsive user experience.

Server Specifications:

1. **Cloud Deployment:**

The entire application is deployed on Vercel, which natively supports Next.js for a unified full-stack development.

2. **Development and Testing:**

For local development, standard developer workstations (e.g., laptops/desktops with a quad-core processor, 8 GB RAM, and SSD storage) are sufficient.

3. **Production:**

Vercel automatically manages resource allocation and scaling in the cloud environment, removing the need for specifying detailed hardware specifications for a small-scale system.

4.3 Software Interfaces

Application Architecture:

Technology Stack

The project will be developed using TypeScript across all layers, ensuring type safety, improved maintainability, and scalability.

1. **Frontend**

The frontend is built using Next.js with Tailwind CSS and DaisyUI to create a responsive and consistent user interface. Figma is used for UI/UX design, ensuring an intuitive and well-structured user experience.

2. **Backend**

The backend is built using Next.js API routes, eliminating the need for a separate backend framework. These API routes will handle business logic and facilitate communication between the frontend and database. Google Apps Script will be used for handling email-based announcements. NextAuth.js will be used for authentication, supporting both OAuth providers and credentials-based login.

3. **Database**

The database is built using MongoDB for storing user and object information. Cloudinary will be used for hosting image and video assets of the application interface as well as user-generated images such as profile pictures.

4. **Deployment**

The entire application (frontend and backend) will be deployed on Vercel.

Third-Party Services

1. **Version Control:** All source code is managed in a GitHub private repository under a GitHub organization.
2. **Task Tracking:** GitHub Projects is used for tracking development progress, assigning tasks, and managing workflows.

4.4 Communications Interfaces

Communication between the client and the server will be done using REST APIs via HTTPS. Request and response data will be transmitted using JSON data format and UTF-8 encoding. User data encryption will be done using the various service providers and well-established libraries used by the application to ensure industry-standard encryption.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The ICS Alumni Tracker and Relations Management System is designed to operate efficiently even under heavy loads. The system must support concurrent users, ensuring that operations such as user authentication, job searches, and event RSVPs are processed promptly—typically within two seconds. In addition, database queries and report generation are optimized to be completed at a reasonable time during normal usage. The overall system architecture is scalable and incorporates appropriate caching strategies to handle increased loads without sacrificing performance.

5.2 Safety Requirements

Although the system does not involve direct physical safety risks, it must maintain the integrity and availability of sensitive data, more specifically, user account details. The system must include robust data backup and recovery procedures to prevent data loss in the event of unexpected failures. Moreover, critical operations—particularly those involving the deletion of records—must require explicit user confirmation, thereby minimizing the risk of accidental data loss and ensuring that data integrity is maintained at all times.

5.3 Security Requirements

Security is a core requirement for the ICS Alumni Tracker and Relations Management System. The system employs industry-standard encryption techniques, such as password hashing, to protect user credentials. All communications between the client and server are secured using SSL/TLS encryption, ensuring data confidentiality during transmission. In addition, a robust role-based access control (RBAC) mechanism is in place, which guarantees that only authorized users can access or modify sensitive information. The system also maintains comprehensive security logs for audit purposes and complies with applicable data privacy regulations and university IT policies.

The system's security is implemented using the Google Identity Services JavaScript library. This library supports both authentication for user sign-in and authorization to obtain an access token for secure interactions with Google APIs.

Moreover, the authentication and authorization flow are as follows:

1. **User Authentication and Authorization**
    - Authentication establishes user identity (user sign-up or sign-in).
    - On the other hand, authorization grants or denies access to data or resources.
    - The system utilizes the OAuth 2.0 implicit flow for in-browser authentication and the OAuth 2.0 authorization code flow for securely exchanging an authorization code with the backend.
2. **Obtaining OAuth 2.0 Credentials**
    - OAuth 2.0 client credentials, including a client ID and client secret, are obtained from the Google API Console.
    - These credentials are securely stored in a *.env* file to prevent unauthorized access.
3. **User Authentication and Token Handling**
    - When a user attempts to log in, the system redirects them to Google's OAuth endpoint for authentication.
    - Upon successful authentication, an authorization code is received and exchanged for an access token.
    - The access token is validated and used to authenticate API requests.

- If necessary, a refresh token is securely stored to maintain user sessions without requiring repeated logins.

4. **Scope and Permission Management**
    - The system requests only the necessary permissions (scopes) required for its functionalities.
    - The granted scopes are examined to ensure proper access control before enabling any features that require Google API access.

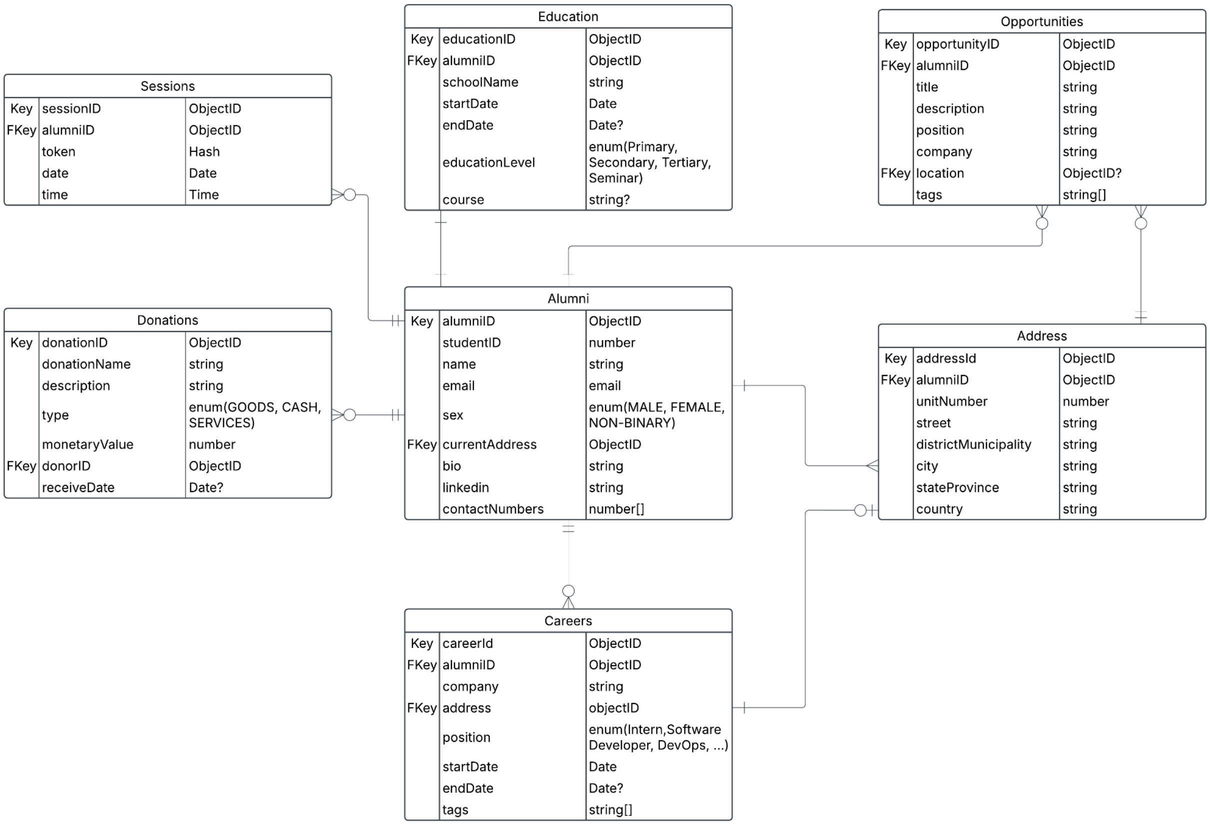
All in all, using the Google Identity Services JavaScript library ensures that users authenticate through their Google accounts – reducing the security risks associated with handling passwords directly.

5.4 Software Quality Attributes

The overall quality of the system is reflected in its high usability, reliability, maintainability, and scalability. The user interface adheres to Material Design principles, providing an intuitive and consistent user experience that facilitates easy navigation and operation. The system is engineered to achieve a high level of reliability, targeting 99.9% uptime, and includes failover mechanisms to handle potential service interruptions. Furthermore, the codebase is modular and well-documented, enabling straightforward maintenance and future enhancements. Scalability is also a key focus, with the architecture designed to accommodate growth in both the user base and data volume without compromising performance. Automated testing frameworks are integrated to ensure that the quality and stability of the system are maintained over time.

6. Appendix

6.1 Entity Relationship Diagram (ERD)



Education

Key	educationID	ObjectID
FKey	alumniID	ObjectID
	schoolName	string
	startDate	Date
	endDate	Date?
	educationLevel	enum(Primary, Secondary, Tertiary, Seminar)
	course	string?

Opportunities

Key	opportunityID	ObjectID
FKey	alumniID	ObjectID
	title	string
	description	string
	position	string
	company	string
FKey	location	ObjectID?
	tags	string[]

Sessions

Key	sessionID	ObjectID
FKey	alumniID	ObjectID
	token	Hash
	date	Date
	time	Time

Donations

Key	donationID	ObjectID
	donationName	string
	description	string
	type	enum(GOODS, CASH, SERVICES)
	monetaryValue	number
FKey	donorID	ObjectID
	receiveDate	Date?

Alumni

Key	alumniID	ObjectID
	studentID	number
	name	string
	email	email
	sex	enum(MALE, FEMALE, NON-BINARY)
FKey	currentAddress	ObjectID
	bio	string
	linkedin	string
	contactNumbers	number[]

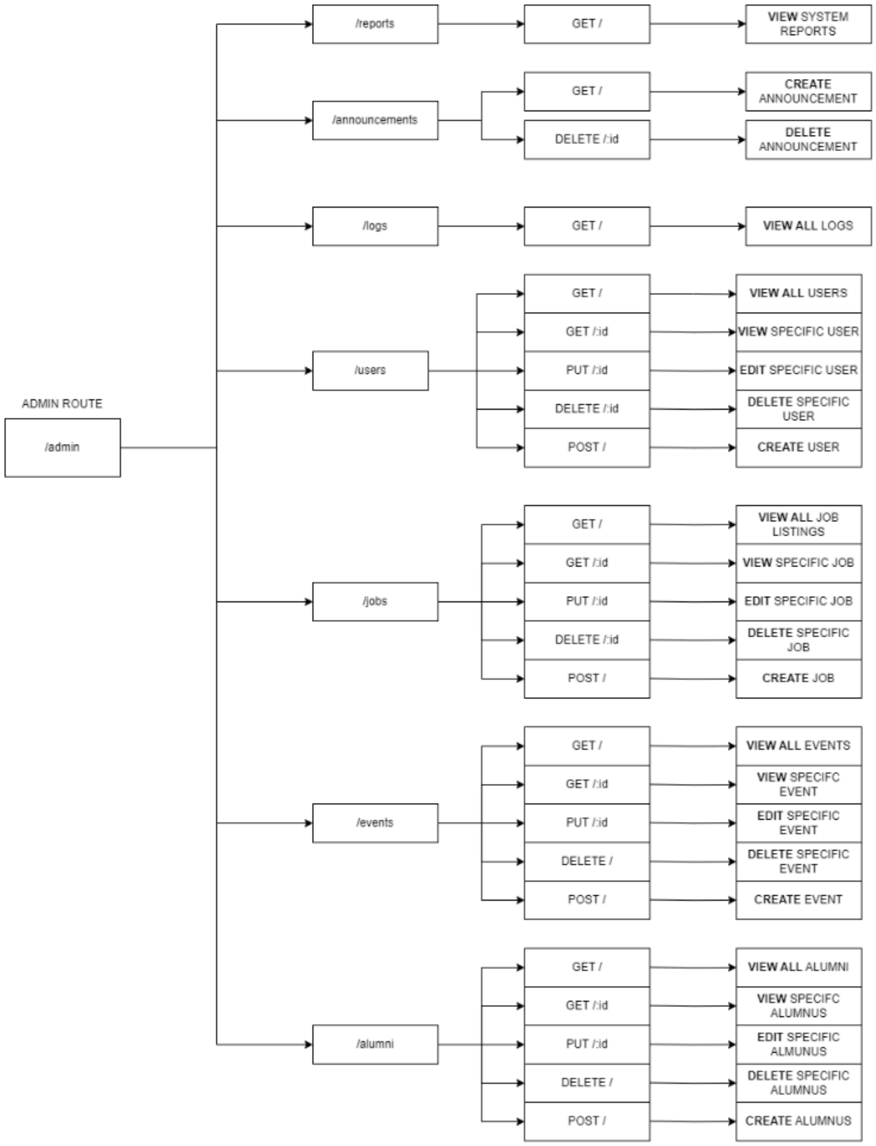
Address

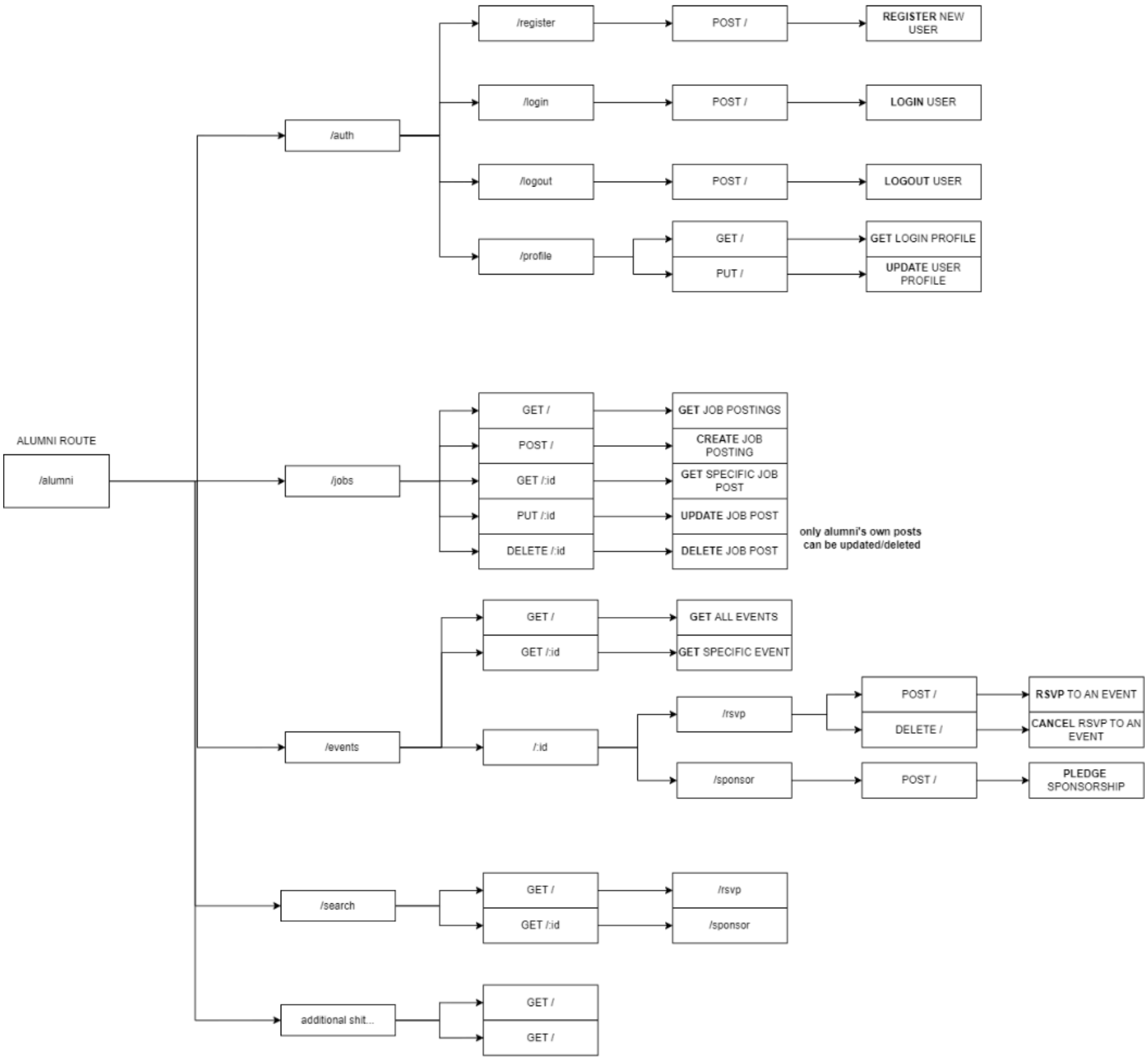
Key	addressId	ObjectID
FKey	alumniID	ObjectID
	unitNumber	number
	street	string
	districtMunicipality	string
	city	string
	stateProvince	string
	country	string

Careers

Key	careerId	ObjectID
FKey	alumniID	ObjectID
	company	string
FKey	address	objectID
	position	enum(Intern,Software Developer, DevOps, ...)
	startDate	Date
	endDate	Date?
	tags	string[]

6.2 Routing Diagram




ALUMNI ROUTE
/alumni
/auth
/register
POST /
REGISTER NEW USER
/login
POST /
LOGIN USER
/logout
POST /
LOGOUT USER
/profile
GET /
GET LOGIN PROFILE
PUT /
UPDATE USER PROFILE
/jobs
GET /
GET JOB POSTINGS
POST /
CREATE JOB POSTING
GET /:id
GET SPECIFIC JOB POST
PUT /:id
UPDATE JOB POST
DELETE /:id
DELETE JOB POST
only alumni's own posts can be updated/deleted
/events
GET /
GET ALL EVENTS
GET /:id
GET SPECIFIC EVENT
/:id
/rsvp
POST /
RSVP TO AN EVENT
DELETE /
CANCEL RSVP TO AN EVENT
/sponsor
POST /
PLEDGE SPONSORSHIP
/search
GET /
/rsvp
GET /:id
/sponsor
additional shit...
GET /
GET /