

CSCI 409 Senior Project II: Final Report

Web Application for Alumni

Adviser: Askar Boranbayev

Co-adviser: Marat Isteleyev

Alisher Abylkassimov

Anuar Seitkan

Dias Utekin

Yerkebulan Stal

Makhambet Kozhakhmet

Spring 2025

Table of Contents

1. Executive Summary	3
2. Introduction	3
3. Background and Related Work	4
4. Project Approach	5
5. Project Execution	9
6. Evaluation	11
7. Conclusion and Possible Future Work	12
8. References	13

1. Executive Summary

Today, the alumni management system relies on Excel spreadsheets in Nazarbayev University. It presents significant drawbacks like data inconsistency, manual processing inefficiencies (human error), lack of scalability, and difficulties in maintaining data integrity. Our main goal is to develop a scalable web-based solution that replaces the existing Excel-based system. So, we started to implement the basic CRUD (Create, Read, Update, Delete) operations. We aimed to add the bulk data upload functionality. It speeds up the adding of new alumni to the system since we can receive files from registrar.nu.edu.kz.

Our solution involves modern, scalable technologies including React for frontend development, Java Spring Boot for the backend, PostgreSQL for efficient data storage, and Docker for streamlined and reliable deployment. The project successfully achieved its objectives, resulting in a fully functional web application featuring comprehensive CRUD operations, intuitive role-based user interfaces, bulk data upload/download capabilities, and robust JWT-based security authentication. Overall, it enhances current system management in different ways such as data handling efficiency, reduces manual errors, and ensures secure and convenient access for administrators. Through careful requirement gathering, structured development methodologies, and testing we developed a robust alumni management system for CAC.

2. Introduction

Currently, Career and Advising Center (CAC) manages alumni data by using Excel spreadsheets. This manual approach has proven to be inefficient, time-consuming, and prone to human errors. Therefore, it was suggested by the IT department of the university to work on a scalable and user-friendly solution. During our interviews with CAC members and IT department representatives, we found out about frequent system crashes and excessively long loading times when working with large datasets. Additionally, it is challenging to maintain the security and preservation of the alumni data with current Excel sheets since they could get potentially deleted, corrupted or stolen at any moment.

Therefore, it is important for the university to address these issues to efficiently and safely keep track of the data and foster long-term relationships between alumni and the university. Replacing the outdated Excel-based system by a website and online database will lead to better resource management, increased efficiency and safety of the data. To

tackle the current problems, we have developed a centralized "Web Application for Alumni". The application allows CAC to easily check the status of all students, manually add or delete alumni records, approve their updates, and efficiently communicate with them via email(рассылка). It is also adapted to receive files in CSV and XLSX formats to convert them into the database, helping administrators to seamlessly transfer data from sources like Office of the Registrar. In addition, the system effectively separates access, ensuring that only administrators have access to sensitive student information. Compared to Excel spreadsheets, our solution is significantly more efficient and less time consuming, especially when dealing with large volumes of data.

3. Background and Related Work

In recent years, numerous alumni management systems have been developed across both academic and commercial contexts. These systems often share common features such as alumni profile management (CRUD operations), secure authentication, bulk data import/export (CSV/XLSX), role-based access control, and communication tools to facilitate alumni engagement. The following section summarizes several relevant platforms and case studies, illustrating how other institutions have addressed similar challenges.

Enihe and Omopariola conducted a case study identifying deficiencies in existing alumni relations at Nigerian universities and proposed a web alumni portal. Their system includes features for tracking alumni contact information, communication management, organizing events/reunions, and handling donations/fundraising. The goal was to enhance alumni engagement and provide networking opportunities between current students and alumni [1]. Moreover, Khan *et al.* designed a web-based alumni information system to strengthen ties between graduates and their college. The platform facilitates communication between the university and its alumni, provides a space for alumni to interact (e.g. forums for exchanging ideas), and offers career support – alumni can discover job opportunities and post announcements (with administrator approval). The system also records alumni employment data and emphasizes security (including access controls) and usability [2]. A modern open-source project that integrates secure user authentication (including 2FA), QR-based event check-ins, private messaging, and a newsfeed. It handles bulk data and event operations efficiently, showing how custom solutions can rival commercial ones [3]. On the other hand, Raklet is an example of a modern commercial alumni management software offering comprehensive functionality.

It includes a powerful alumni CRM that supports bulk data upload – institutions can import alumni spreadsheets (Excel/CSV) to quickly populate the database and create detailed alumni profiles. The platform provides robust communication tools as well, essentially a private social network for alumni: administrators and members can share announcements, and alumni engage via discussion forums and one-to-one private messaging on the portal. Raklet (and similar products like Vaave and Almabase) also support event management, email newsletters, and fundraising modules, reflecting the common feature set expected in alumni management systems [4].

Each of the above solutions addresses core requirements of alumni management in similar ways. They all implement CRUD operations for alumni profiles (with options for alumni to update their information), enforce secure authentication and role-based privileges (e.g. admin vs. alumni user rights), allow bulk handling of alumni data, and include communication channels to nurture an active alumni community. The convergence of features across academic prototypes and real-world platforms underlines their importance for any “Web Application for Alumni” project.

4. Project Approach

The Alumni Management System is structured into a clear, maintainable three-layer architecture:

Frontend Layer:

- Login/Register pages
- Alumni data management views (CRUD forms, alumni list with search/filter/pagination)
- Newsletter composition form
- Dedicated dashboards based on user roles

Backend Layer:

- Secure JWT-based authentication and role-based access control
- CRUD operations and data handling via RESTful APIs
- Automated notifications and email services
- Audit log management for tracking administrative actions
- Work obligation status tracking and updates

Database Layer:

- Structured schemas to manage alumni data, user accounts, notifications, and audit logs
- Strong dependency relationships between tables to ensure data integrity and consistency

The entire application is containerized using Docker, ensuring easy deployment and scalability, while Nginx acts as a reverse proxy for handling incoming requests efficiently.

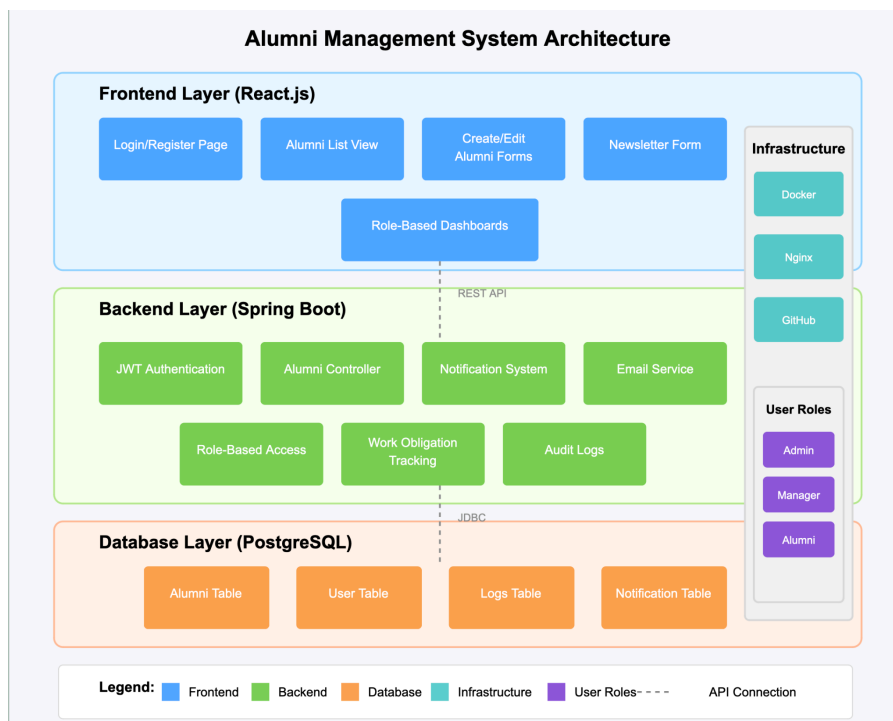


Figure 1. System Architecture

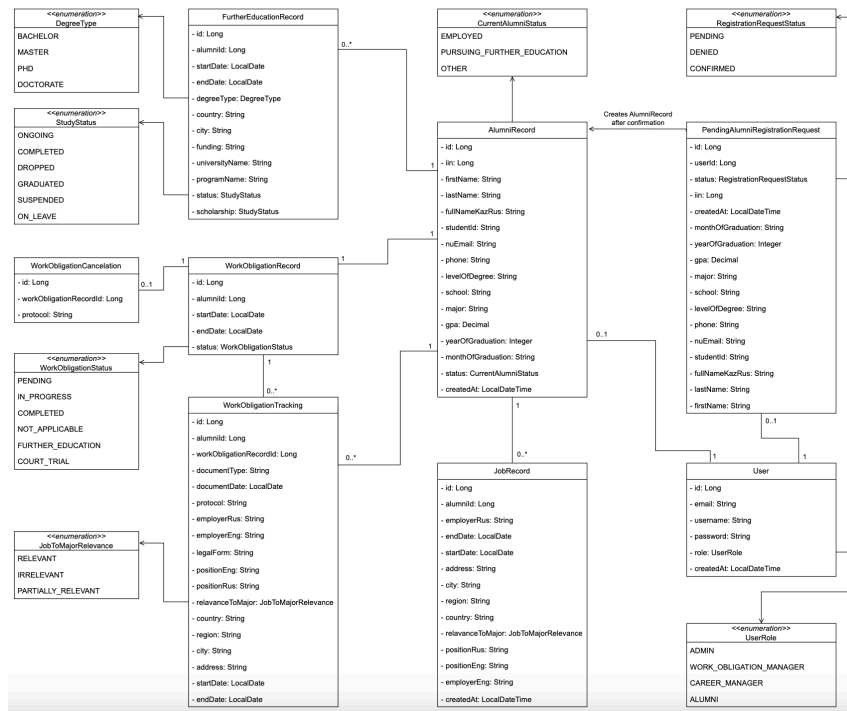


Figure 2. Database schema

CRUD functionalities utilize Spring Boot controllers and services, providing efficient data validation, transaction management, and interaction with the PostgreSQL database. Frontend forms dynamically interact with backend endpoints via RESTful APIs for creating, updating, or deleting records.

- Bulk Upload/Download Handling:
 - Excel/CSV file uploads are managed by the backend, which processes and validates data, mapping columns to the database schema to ensure accurate bulk imports.
 - Optimized batch processing methods handle large volumes efficiently, significantly improving performance compared to manual Excel handling.
- JWT-Based Authentication:
 - Secure JWT tokens are generated upon successful login, ensuring stateless session management.
 - Middleware intercepts and validates tokens for each request, maintaining secure role-based access.

The application distinctly separates permissions based on clearly defined roles:

- Admins: Full access to alumni data, including creation, deletion, modification, approval of data updates, bulk operations, audit logs, and email communications.
- Managers: Limited permissions, managing specific alumni groups and tracking work obligations without full administrative access.
- Alumni: Limited self-service capabilities, allowing them to update personal information pending administrative approval.

Tools Used:

- Frontend: React.js
- Backend: Java Spring Boot
- Database: PostgreSQL
- Deployment: Docker (containerization), Nginx (reverse proxy)
- Version Control: GitHub
- API Testing: Postman
- Task & Project Management: Notion (used to define tasks, document decisions, track progress)

The core functionalities are illustrated through clear use-case diagrams covering:

- Authentication and authorization processes
- CRUD operations and management of alumni records
- Bulk data handling operations (Excel/CSV uploads)
- Role-specific interactions and actions (Admins, Managers, Alumni)

The system incorporates several reliable third-party libraries and tools, including:

- Spring Security: For implementing JWT authentication and authorization.
- Apache POI: For parsing Excel files for bulk upload/download functionality.
- Redux toolkit RTK query: For seamless frontend-backend HTTP communication in React.
- Docker and Docker Compose: For containerizing application components, ensuring portability and ease of deployment.

The team adopted agile methodologies (Scrum), breaking the project into smaller achievable tasks organized within fixed-length iterations. Regular in-person meetings every Friday and daily Telegram communications ensured continuous collaboration and

rapid issue resolution. Responsibilities were clearly divided based on team members' strengths, as illustrated below:

- Frontend Developers: Anuar Seitkan, Yerkebulan Stal
- Backend Developers: Dias Utekin, Makhambet Kozhakhmet
- Full-stack & DevOps: Alisher Abylkassimov

The team effectively utilized Notion for task management, detailed documentation, and tracking progress. GitHub facilitated collaborative coding, version control, and CI/CD strategies. This structured approach ensured the team remained aligned, efficient, and responsive to stakeholder feedback and evolving project requirements.

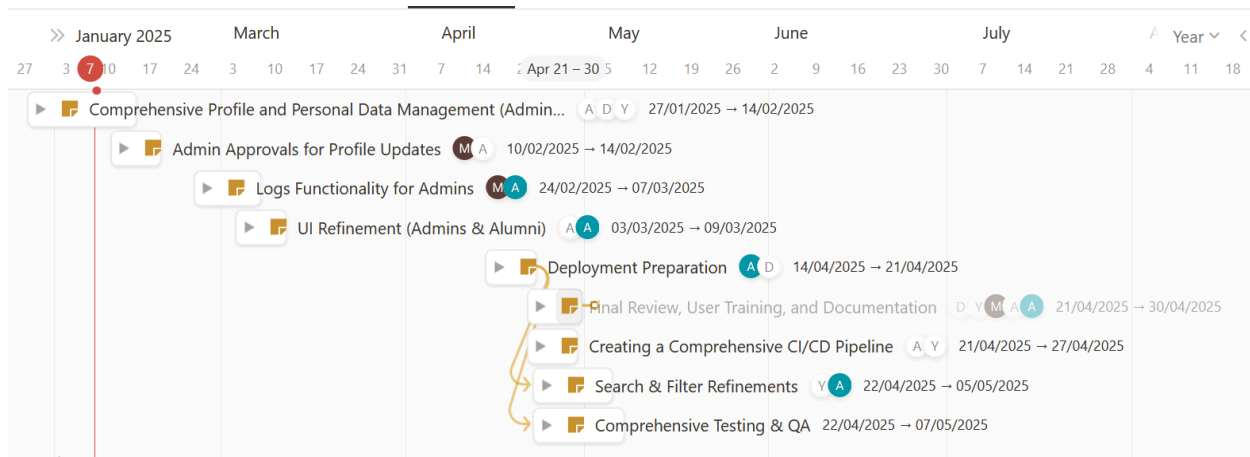


Figure 3. Gantt chart

5. Project Execution

Throughout the semester, the project team focused on expanding and refining the core functionalities of the "Web Application for Alumni" based on feedback received during previous development cycles. We have implemented the following key improvements to achieve a scalable, user-friendly, and time efficient system for both administrators and alumni.

We used Scrum methodology to manage the development process. We set ourselves small, achievable goals, and scheduled to work on them at fixed-length iterations. We held team meetings every Friday to keep track of each others' progress, adjust goals and tasks, and share feedback. Our team was mainly divided into two groups: frontend and

backend, each consisting of two members. A fifth person, Alisher Abylkassimov, oversaw the general process, contributing to both teams and facilitating integration between frontend and backend work. The development process was split into two major areas:

1. Intuitive UI. One of the main goals of our project was to provide a user-friendly interface for administrators and alumni. Our UI presents the data in an easily readable form, and allows users to manipulate and customize the tables. This area was handled by the frontend team utilizing React.
2. Efficient backend and enhanced database architecture. Scalability and performance were another focal points of our project. We significantly improved the database schema to store more comprehensive alumni data. Dependency constraints were added to ensure data integrity and effectively manage table relationships. Backend team used Java Spring Boot to build a robust and modular API.

Challenges and Solutions:

- Data synchronization. Handling large datasets with numerous fields and role-specific constraints presented synchronization challenges. The team optimized backend queries and ensured consistent API responses for all frontend operations.
- Performance with large datasets. Considering that NU has over 10000 alumni, the system was expected to have serious performance problems while facing such a large amount of data. Therefore, the system initially experienced delays when we tested out bulk data uploads. We have addressed this issue by introducing pagination and optimizing database indices.
- Team coordination. It was challenging to synchronize team members' schedules due to everyone having different coursework at first. However, we have settled on meeting every Friday to keep track of each others' progress. We also used Telegram for daily communication, Zoom for online meetings, and Notion for sharing ideas.

Collaboration and Tools:

- Meetings: Weekly in-person meetings (1 hour each Friday).
- Communication: Daily updates and troubleshooting via Telegram.

- Task management: We utilized Notion for roadmap planning, tracking/sharing tasks, and documenting stakeholder requirements.
- Version control: GitHub used for collaborative coding.

The project was divided among members in the following way:

- Comprehensive profile and data management: Alisher Abylkassimov and Anuar Seitkan
- Admin approvals: Dias Utekin and Makhambet Kozhakhmet
- Logs functionality: Yerkebulan Stal and Alisher Abylkassimov
- Bulk upload/download: Dias Utekin and Makhambet Kozhakhmet
- UI refinement: Anuar Seitkan and Yerkebulan Stal
- Search & Filter improvements: Alisher Abylkassimov and Dias Utekin
- Deployment preparation: Anuar Seitkan and Makhambet Kozhakhmet
- Testing & QA: Everyone

6. Evaluation

To validate the effectiveness of our solution, we conducted interviews with key stakeholders throughout the project. This included meetings with members of the IT department—Yernar Temirgali and Kuanysh Yersakhanov—on September 13th to discuss the goals and required technical features of the project, and a follow-up session on September 25th with members of the Career and Advising Center (CAC) to gather detailed feedback on current pain points with Excel-based data management.

Additionally, we remained in close communication with our project advisor and co-adviser, incorporating their technical feedback into our design choices. Feedback was continuously collected via in-person reviews and sprint presentations, helping us assess usability, feature completeness, and system performance.

Manual testing by ourselves emphasized the drastic improvement in usability, especially in terms of search speed, data structure clarity, and role-based access. The new UI made it easier for admins to check alumni status, perform bulk actions, and securely manage sensitive data. The platform's ability to upload `.csv` and `.xlsx` files—formats aligned with registrar.nu.edu.kz—was identified as one of the most helpful features for real-world use. Compared to Excel, our system processed large datasets more efficiently, without crashes or slowdowns.

7. Conclusion and Possible Future Work

In conclusion, the development of the Alumni Web Application system has successfully addressed the issues that were faced by Nazarbayev University's Career and Advising Center. By replacing the outdated and error-prone Excel-based system with a centralized, web-based platform, the project addresses critical challenges in usability, scalability and effectiveness. The solution not only streamlines CRUD operations and improves data accuracy, but also introduces safe authentication system, role-specific interfaces, and bulk upload/download features compatible with existing university data formats.

Looking forward, there are several promising ideas that could further enhance the system's value to Nazarbayev University. The existing system provides a solid foundation that can be extended in multiple directions:

- **Enhanced analytics:** Adding advanced data visualization tools, such as power BI, could provide more insights on our alumni. Currently, CAC downloads the data and sends it to the IT department for data visualization. Implementing a data visualization system into our project could potentially increase efficiency of such processes.
- **Alumni networking hub:** Currently, our web application is more focused on tackling the problems faced by CAC employees. However, the platform could further evolve to facilitate direct connections between alumni. This would create a professional networking ecosystem where NU graduates could connect based on many factors including industry, graduation year, or academic interests.
- **Mobile application development:** A companion mobile application would be a good addition if alumni networking hub was properly implemented.
- **AI-Driven features:** Current trend on AI shows that we can utilize it in our system too. For example, develop a recommendation system about the events. It builds a strong network among the students and alumni. Additionally, it could be helpful for alumni to find the correct career path after graduation.

However, in order to enhance our system with mentioned points we need to consider the effectiveness of these features. We require careful planning and prioritization to receive high output for alumni and staff. By continuing to upgrade its current data management focus, Nazarbayev University has the opportunity to create a comprehensive digital ecosystem.

8. References

1. “Alumni Management System using PHP/MySQL with Source Code,” SourceCodester. [Online]. Available: <https://www.sourcecodester.com/php/14524/alumni-management-system-using-phpmysql-source-code.html>.
2. N. A. Khan, A. M. U. Siddiqi, and M. Ahmad, “Development of intelligent alumni management system for universities,” *Asian Journal of Basic Science & Research*, vol. 03, no. 02, pp. 51–60, Jan. 2021, doi: 10.38177/ajbsr.2021.3206.
3. R. Enihe, V. Omopariola, “Alumni Portal System for Nigerian Universities (A Case Study),” *International Journal of Science and Research (IJSR)*, vol. 12, no. 2, pp. 1047–1055, Feb. 2023, doi: 10.21275/sr23217134355.
4. Raklet Inc, “The Ultimate Guide to Alumni Association Management,” Raklet Inc, Dec. 13, 2016. [Online]. Available: <https://www.raklet.com/blog/alumni-association-management/>.