

Automated Notification System for Document Expiration Management

CSCI 409 Senior Project II – Final Project Report

Spring 2025

Group Members: Kuanysh Murat, Islam Koshim, Nurtore Kelessov, Azamat Ormanov

Adviser(s): Askar Boranbayev, Kuanysh Yersakhanov

Nazarbayev University

Submitted: April 25, 2025

Contents

1	Executive Summary	2
2	Introduction	2
2.1	Problem Statement.....	2
2.2	Motivation and Significance.....	2
2.3	Proposed Solution.....	2
2.4	Report Organization.....	3
3	Background and Related Work	3
3.1	Prior Research and Existing Solutions.....	3
3.2	Comparison of Approaches.....	3
3.3	Justification of Methodology.....	3
4	Project Approach	3
4.1	Solution Description.....	3
4.2	Architecture.....	4
4.3	Third-Party Components.....	4
4.4	Tools and Technologies.....	4
4.5	Use Case Diagram.....	5
4.6	Team Functioning.....	5
5	Project Execution	5
5.1	Fall 2024.....	5
5.2	Spring 2025.....	6
5.3	What Went Right.....	6
5.4	What Went Wrong.....	7
5.5	Problem-Solving.....	7
5.6	Teamwork.....	7
6	Evaluation	7
6.1	Evaluation Methodology.....	7
6.2	Data Collected.....	7
6.3	Analysis.....	8
6.4	Validation of Computing-Based Solution.....	8
7	Conclusion and Possible Future Work	8
7.1	Key Findings and Contributions.....	8
7.2	Future Enhancements.....	9
7.3	Conclusion.....	9
8	References	9

1. Executive Summary

The *Automated Notification System for Document Expiration Management* is a web-based software solution designed to address the inefficiencies and risks associated with manual document expiration tracking at Nazarbayev University. The system automates the process of managing document expiration dates, sending timely notifications to relevant stakeholders, and ensuring compliance with legal and institutional requirements. Key objectives included reducing human errors, enhancing operational efficiency, and securing sensitive data.

The project was executed over two semesters. In Fall 2024, the team focused on system design, requirements analysis, and initial implementation of the notification engine and user interface. In Spring 2025, we completed integration with external systems (e.g., Google Calendar), implemented escalation protocols, and deployed the system locally. The solution was evaluated through user acceptance testing, performance metrics, and compliance checks, confirming its effectiveness in mitigating missed deadlines and improving document management.

This computing-based solution leverages a 3-tier architecture (React frontend, Java Spring Boot backend, PostgreSQL database) and integrates with multiple notification channels (email, Telegram, in-app alerts). The project successfully aligns with the university's needs, delivering a scalable, secure, and user-friendly system.

2. Introduction

2.1 Problem Statement

The manual process of tracking document expiration dates at Nazarbayev University is error-prone and susceptible to oversight. Critical documents, such as legal contracts, certifications, and licenses, may expire without adequate notice, leading to non-compliance, operational disruptions, and resource wastage. The lack of automation increases the administrative burden and risks legal or financial penalties.

2.2 Motivation and Significance

The motivation for this project stems from the need to streamline document management, reduce human errors, and ensure compliance with regulatory and institutional deadlines. By automating notifications and integrating with existing systems, the solution enhances efficiency, saves time, and mitigates risks, benefiting university administrators, department heads, and compliance officers.

2.3 Proposed Solution

The Automated Notification System provides a comprehensive platform for managing document expirations. It features automated notifications, role-based access control (RBAC), integration with calendar systems, and compliance tracking. The system supports multiple notification channels and offers a user-friendly interface.

2.4 Report Organization

This report is structured as follows: Section 3 reviews related work, Section 4 details the project approach, Section 5 describes project execution, Section 6 presents the evaluation, Section 7 concludes with future work, and Section 8 lists references.

3. Background and Related Work

3.1 Prior Research and Existing Solutions

Manual document tracking is a common challenge in large organizations, often leading to missed deadlines and non-compliance. Existing solutions include enterprise document management systems (e.g., DocuWare, SharePoint) and calendar-based reminder tools. However, these systems often lack automated, customizable notifications tailored to expiration dates or integration with university-specific workflows.

Dalal et al. (2012, 2014) explored automated email notification systems for medical test results, demonstrating improved compliance and reduced oversight through timely reminders. Their work highlights the efficacy of automated notifications in time-sensitive scenarios, a principle applied in our system. Holmes et al. (2024) analyzed automated notification systems for hospital wards, emphasizing cost-utility and scalability, which informed our focus on a cost-effective, scalable solution.

3.2 Comparison of Approaches

Unlike generic document management systems, our solution is tailored to university needs, offering advanced tagging and escalation protocols. Calendar-based tools lack the robust notification engine and compliance tracking features of our system. The use of open source frameworks like Spring Boot and React, as demonstrated in example projects by Priom7 [2023], kantega [2018], CyberButton [2025a], and CyberButton [2025b], provides a robust foundation for building scalable web applications with real-time notification capabilities, justifying our technology stack.

3.3 Justification of Methodology

Our methodology leverages a 3-tier architecture to ensure scalability, maintainability, and user accessibility. The use of PostgreSQL supports complex data management, while REST APIs enable seamless integration with external systems. The focus on security (RBAC) aligns with data protection laws, addressing stakeholder concerns about sensitive data.

4. Project Approach

4.1 Solution Description

The Automated Notification System is a web-based application that automates the tracking and notification of document expirations. It consists of the following components:

1. **Document Management Module:** Allows users to upload, categorize, tag, and manage documents with expiration dates. Supports bulk uploads and advanced search.

2. **Notification Engine:** Calculates expiration timelines and sends reminders via email, Telegram, and in-app alerts. Includes escalation protocols for critical documents.
3. **User Interface:** A React-based frontend with an intuitive dashboard for tracking expirations.
4. **Integration Module:** Synchronizes with Google Calendar and existing document management systems using REST APIs.
5. **Analytics and Reporting:** Generates reports on expiration trends and compliance metrics.
6. **Security Layer:** Implements RBAC and compliance with Kazakhstan's data protection laws.

4.2 Architecture

The system follows a 3-tier architecture:

- **Presentation Layer:** React frontend, accessible via Chrome, Firefox, Edge, and Safari (any browser).
- **Application Layer:** Java Spring Boot backend handling business logic, notifications, and API requests.
- **Data Layer:** PostgreSQL database storing documents, metadata, and user configurations.

4.3 Third-Party Components

- **Email Services:** Integrated with SMTP servers for email notifications.
- **Telegram API:** Used for real-time notifications via Telegram bots.
- **Google Calendar API:** Enables calendar synchronization for reminders.

4.4 Tools and Technologies

- **Backend:** Java, Spring Boot, RESTful APIs with DTO pattern.
- **Frontend:** React with TypeScript, featuring protected routes, responsive layouts.
- **Database:** PostgreSQL for entity relationship management, data validation, and optimized queries.
- **Security:** RBAC, JWT-based authentication, CORS configuration, XSS/CSRF protection, and secure password handling.

4.5 Use Case Diagram

- **Actors:** Document Manager, Department Administrator, End User
- **Use Cases:**
 - Add/Update/Delete Document
 - Receive Notifications

4.6 Team Functioning

The team was structured with clear roles:

- **Nurtore Kelessov** (Project Manager, Backend Developer): Managed timelines, implemented backend logic, and handled integrations.
- **Kuanysh Murat** (Frontend Developer, UX/UI Designer): Designed and developed the user interface with multi-language support.
- **Azamat Ormanov** (DevOps Engineer, Security Specialist): Architected the system, implemented security measures, and managed deployments.
- **Islam Koshim** (Database Administrator, QA Engineer): Designed the database schema, ensured data integrity, and led testing.

Collaboration was facilitated through GitHub for code sharing, and Telegram for communication.

5. Project Execution

5.1 Fall 2024

The Fall semester focused on foundational work:

- **System Design:** Finalized the 3-tier architecture, use cases, and database schema.
- **Requirements Analysis:** Gathered functional (e.g., document management, notifications) and non-functional (e.g., scalability, security) requirements.
- **Initial Implementation:**
 - Built a preliminary React-based user interface with document upload/download

functionality.

- Experimented with scheduling libraries and notification channels.

- **Challenges:**

- WhatsApp and SMS integrations were paused due to paid service limitations.
- Initial database performance issues were resolved by optimizing queries.

- **Decisions:**

- Prioritized free notification channels (email, Telegram) to align with budget constraints.

5.2 Spring 2025

The Spring semester focused on completion and refinement:

- **Escalation Protocols:** Implemented tiered reminders for critical documents, increasing frequency as deadlines approached.
- **System Integration:** Completed integration with Google Calendar and existing document management systems via REST APIs.
- Developed the notification engine to send email and Telegram
- Implemented Google Calendar integration for event creation.
- **Testing and Debugging:**
 - Conducted unit, integration, and system tests to ensure notification accuracy and system reliability.
 - Performed security testing, including penetration tests, to address vulnerabilities.
- **User Feedback:** Conducted usability testing with 20 university students, leading to UI refinements (e.g., simplified navigation, enhanced localization).
- **Documentation:** Prepared user guides, technical documentation, and training materials.
- **Deployment:** Deployed the system on a local server, however in future it can be deployed on a cloud based server ensuring scalability for 1,000 simultaneous users.

5.3 What Went Right

- Effective team collaboration by GitHub streamlined development.
- Early prototyping of the notification engine ensured core functionality was robust.

5.4 What Went Wrong

- Initial underestimation of integration complexity with Outlook required additional research.
- Localization of UI elements in Kazakh and Russian was time-consuming due to various challenges.
- The process of setting up SSL/TLS certificates and configuring HTTPS haven't been fully understood and explored.
- Limited budget restricted SMS integration, requiring reliance on alternative channels.
- The authorization system had a bug, and the token got messed up.
- Time management - some of the needed features weren't deployed.

5.5 Problem-Solving

- Integration issues were resolved by leveraging community-driven API documentation and alternative endpoints.
- Budget constraints were mitigated by prioritizing free, reliable notification channels.

5.6 Teamwork

Responsibilities were divided based on expertise, with regular sprint reviews to ensure alignment. Nurtore's leadership ensured timely milestone completion, while Azamat's security focus strengthened the system's compliance. Kuanysh and Islam collaborated closely on UI-database integration, resolving cross-layer issues efficiently.

6. Evaluation

6.1 Evaluation Methodology

The system was evaluated based on four criteria:

1. **Notification Accuracy:** Verified that notifications were sent at specified intervals (e.g., 30, 15, 7 days before expiration).
2. **System Reliability:** Measured uptime and performance.
3. **User Satisfaction:** Collected feedback via surveys and usability testing with 20 university students.
4. **Compliance and Security:** Ensured adherence to Kazakhstan's data protection laws and university policies.

6.2 Data Collected

- **Notification Accuracy:**
 - Tested with 100 sample documents: 98% of notifications were sent on schedule,

with minor delays (<1 minute) due to server load

- **User Satisfaction:**

- Survey results (n=20): 85% rated the system as “very user-friendly,” 90% found notifications timely.
- Common feedback: Request for mobile app integration (planned for future work).

- **Compliance and Security:**

- Audit logs captured all user actions, meeting compliance requirements.
- Penetration testing identified and resolved two minor vulnerabilities (e.g., session timeout configuration).

6.3 Analysis

The evaluation confirms that the system effectively addresses the problem of manual document tracking. Notification accuracy and reliability ensure timely reminders, reducing the risk of missed deadlines. High user satisfaction validates the intuitive design and customization options. Compliance with data protection laws and robust security measures protect sensitive information, aligning with stakeholder requirements.

6.4 Validation of Computing-Based Solution

The system’s use of a scalable 3-tier architecture, REST APIs, and automated notification logic demonstrates a robust computing-based solution. Integration with external systems and real-time analytics enhance its practical utility, while the focus on security and accessibility ensures broad applicability.

7. Conclusion and Possible Future Work

7.1 Key Findings and Contributions

The Automated Notification System successfully automates document expiration management, reducing manual workload, enhancing compliance, and mitigating risks. Key contributions include:

- A scalable, secure platform.
- Automated notifications via multiple channels, with escalation protocols for critical documents.
- Integration with Google Calendar and existing systems, streamlining workflows.

- Comprehensive compliance tracking and analytics, providing actionable insights.

7.2 Future Enhancements

- **Mobile App:** Develop iOS and Android apps to enhance accessibility.
- **SMS Integration:** Explore cost-effective SMS services for broader notification coverage.
- **AI-Powered Predictions:** Implement machine learning to predict renewal timelines based on historical data.
- **Extended Integrations:** Support additional calendar systems (e.g., Apple Calendar) and document management platforms.
- **Accessibility Improvements:** Enhance WCAG compliance for users with disabilities.

7.3 Conclusion

The project delivers a robust, user-friendly solution that addresses the university's document management challenges. By leveraging modern technologies and iterative development, the system sets a foundation for future enhancements, ensuring long-term value.

8. References

CyberButton. React notification cursor, 2025a. CyberButton/react-notification-cursor.

CyberButton. [https://github.com/ Notification spring backend](https://github.com/Notification-spring-backend), 2025b.

[https://github.com/ CyberButton/notification-spring-backend](https://github.com/CyberButton/notification-spring-backend).

Dalal, A. K., Roy, C. L., Poon, E. G., Williams, D. H., Nolido, N., Yoon, C., Budris, J., Gandhi, T., Bates, D. W., & Schnipper, J. L. (2014). Impact of an automated email notification system for results of tests pending at discharge: a cluster-randomized controlled trial. *Journal of the American Medical Informatics Association : JAMIA*, 21(3), 473–480.

<https://doi.org/10.1136/amiainl-2013-002030>

Dalal, A. K., Roy, C. L., Poon, E. G., Williams, D. H., Nolido, N., Yoon, C., Budris, J., Gandhi, T., Bates, D. W., & Schnipper, J. L. (2014). Impact of an automated email notification system for results of tests pending at discharge: A cluster-randomized controlled trial. *Journal of the American Medical Informatics Association*, 21(3), 473–480.

<https://doi.org/10.1136/amiainl-2013-002030>

Holmes E, Lloyd Williams H, Hughes D, Naujokat E, Duller B, Subbe CP (2024) A model-based cost-utility analysis of an automated notification system for deteriorating patients on general wards. *PLoS ONE* 19(5): e0301643.

<https://doi.org/10.1371/journal.pone.0301643>

kantega. Example of a spring boot project with a react frontend, 2018. URL <https://github.com/kantega/react-and-spring>