



ELECTRONIC DECLARATION SYSTEM (EDS)

CSCI 409 SENIOR PROJECT II - FINAL PRESENTATION

Azat Akash, Nurlan Serikov, Sultan Turan, Nurtore Onggarbay

Spring 2025





Table of Contents

1 Introduction

► Introduction

► Project Development

► Evaluation

► Demonstration

► Conclusion and Future Work



Problem Description

The Challenge with Manual CoI Management

- Nazarbayev University's Conflict of Interest (CoI) statement management was historically *manual*.
- **Drawbacks:**
 - Significant administrative workload.
 - Potential delays in evaluation and resolution.
 - Increased risks of non-compliance with NU policy.
 - Lack of transparency and efficient tracking mechanisms.
- Maintaining compliance with the NU Policy and Procedure on Conflict of Interest was challenging.



Motivation and Significance

Why Develop the EDS?

- Need to modernize and improve the CoI management process at NU.
- Align the process with the official NU CoI Policy.
- **Benefits of Digitalization:**
 - Increased efficiency via automated workflows.
 - Enhanced transparency for declarants and managers.
 - Improved compliance tracking with better audit trails.
 - Minimized administrative workload associated with manual processing.



Proposed Solution: EDS Overview

The Electronic Declaration System

- A dedicated web-based application: **Electronic Declaration System (EDS)**.
- Manages the entire lifecycle of CoI statements.
- **Core Purpose:** Automate submission, evaluation, and resolution of potential conflicts.
- **Key Declaration Types:**
 - *Initial Statements* (hiring/system startup).
 - *Ad hoc Statements* (updates/new conflicts).
 - *Management Plans* (prepared by Managers to handle known conflicts).
- **Role-Based Access:** Personalized dashboards for:
 - **Declarants** (Employees)
 - **Managers** (Compliance Office)
 - **Administrators / Super-Administrator** (IT Department)



Table of Contents

2 Project Development

- ▶ Introduction
- ▶ Project Development
- ▶ Evaluation
- ▶ Demonstration
- ▶ Conclusion and Future Work



System Architecture

How EDS is Built

- Multi-tier architecture for scalability and separation of concerns.
- **Frontend:** React.js (Next.js) for user interface.
- **Backend API:** Java Spring Boot handling business logic.
- **Database:** PostgreSQL for structured data (users, declarations).
- **File Storage:** MinIO (S3-compatible) for document attachments.
- **Load Balancer:** Nginx for distributing traffic.
- Communication primarily via HTTP/HTTPS.



Team Contributions

Division of Responsibilities

Responsibilities were divided based on expertise and interest:

- **Azat Akash:**
 - PostgreSQL Database Design
 - Backend Development (Primary Contributor)
 - Shared Project Management
- **Sultan Turan:**
 - Frontend/UI Development (Primary Contributor)
- **Nurlan Serikov:**
 - Frontend Development
 - UI Design & Testing
- **Nurtore Onggarbay:**
 - DevOps (Hosting: Dev & Prod)
 - Documentation (Report)
 - Project Management & Stakeholder Communication



EDS System Architecture Diagram

2 Project Development

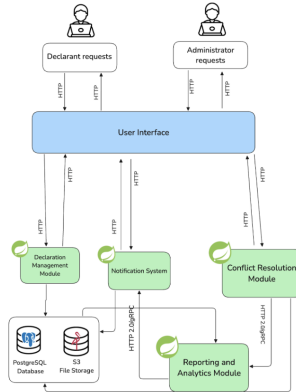


Figure: EDS System Architecture Diagram



Core Workflow

Declaration Lifecycle

1. **Submission:** Declarant submits Initial or Ad hoc declaration via web UI.



Core Workflow

Declaration Lifecycle

1. **Submission:** Declarant submits Initial or Ad hoc declaration via web UI.
2. **Notification:** System saves data & notifies Manager(s).



Core Workflow

Declaration Lifecycle

1. **Submission:** Declarant submits Initial or Ad hoc declaration via web UI.
2. **Notification:** System saves data & notifies Manager(s).
3. **Review:** Manager examines the submission.



Core Workflow

Declaration Lifecycle

1. **Submission:** Declarant submits Initial or Ad hoc declaration via web UI.
2. **Notification:** System saves data & notifies Manager(s).
3. **Review:** Manager examines the submission.
4. **Resolution (No Conflict):** Manager marks declaration as resolved.



Core Workflow

Declaration Lifecycle

1. **Submission:** Declarant submits Initial or Ad hoc declaration via web UI.
2. **Notification:** System saves data & notifies Manager(s).
3. **Review:** Manager examines the submission.
4. **Resolution (No Conflict):** Manager marks declaration as resolved.
5. **Resolution (Conflict):** Manager drafts a Management Plan outlining mitigation steps.



Core Workflow

Declaration Lifecycle

1. **Submission:** Declarant submits Initial or Ad hoc declaration via web UI.
2. **Notification:** System saves data & notifies Manager(s).
3. **Review:** Manager examines the submission.
4. **Resolution (No Conflict):** Manager marks declaration as resolved.
5. **Resolution (Conflict):** Manager drafts a Management Plan outlining mitigation steps.
6. **Acknowledgement:** System notifies Declarant, who reviews and approves the plan.



Core Workflow

Declaration Lifecycle

1. **Submission:** Declarant submits Initial or Ad hoc declaration via web UI.
2. **Notification:** System saves data & notifies Manager(s).
3. **Review:** Manager examines the submission.
4. **Resolution (No Conflict):** Manager marks declaration as resolved.
5. **Resolution (Conflict):** Manager drafts a Management Plan outlining mitigation steps.
6. **Acknowledgement:** System notifies Declarant, who reviews and approves the plan.
7. **Tracking:** System tracks status throughout and maintains an audit trail.



EDS Sequence Diagram

2 Project Development

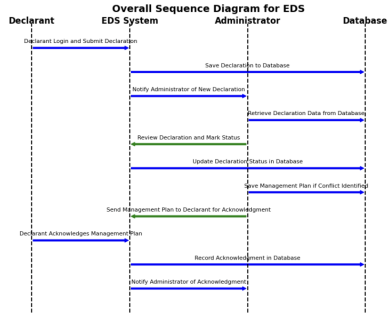


Figure: EDS Sequence Diagram



User Roles

Who Uses the System?

- **Declarant:**
 - NU employees submitting CoI declarations.
 - Submit Initial/Ad hoc forms, view status, review and accept Management Plans.
- **Manager:**
 - Staff from the Conflict of Interest, Ethics, and Compliance Office.
 - Review submitted declarations, determine conflicts, create/oversee Management Plans, monitor dashboards, generate CoI compliance reports.
- **Administrator:**
 - NU IT department employees.
 - Overall system management (user accounts, esp. Managers), monitoring health/performance/stats, technical maintenance, backups.
- **Super-Administrator:**
 - NU IT department employee.
 - Same as Administrator, but can create other Administrators.



Key Features Developed

What Can EDS Do?

- **User Management:** Role-Based Access Control, login/registration (Declarants), Admin tools for Managers.
- **Declaration Management:** Submission (*Initial, Ad hoc*), status monitoring for declarants.
- **Conflict Review & Management Plans:** Manager tools for review, conflict spotting, plan creation/oversight, tracking declarant acknowledgements.
- **Notification System:** Automated email/in-app alerts for submissions, status updates, necessary actions, and reminders.



Key Features Developed

What Can EDS Do?

- **Dashboards:** Separate dashboards for Declarants, Managers, and Administrators.
- **Data Analytics/Reporting:** Managers can export data/generate basic reports; Admins access system-level analytics.
- **Audit Trail:** Comprehensive logging of significant actions for compliance/accountability checks.



Use Cases

System Functionality Examples

Main use cases of the system include:

- Declarant can submit an Initial Declaration.
- Declarant can submit an Ad hoc Declaration.
- Manager can review a Declaration.
- Manager can create a Management Plan.
- Declarant can acknowledge a Management Plan.
- Administrator can create/manage user accounts.
- Administrator can view system statistics.



Table of Contents

3 Evaluation

- ▶ Introduction
- ▶ Project Development
- ▶ **Evaluation**
- ▶ Demonstration
- ▶ Conclusion and Future Work



Evaluation Method

How We Assessed EDS

- **Primary Method:** User Beta Testing.
- **Participants:** Key stakeholders:
 - Potential *Declarants* (NU employees).
 - *Managers* (Compliance Office representatives).
 - Potentially *Administrators* (IT personnel).
- **Focus:** Verify if the system successfully addresses ineffective manual CoI management, beyond basic bug testing.
- **Process:**
 - Participants given specific, real-world tasks/scenarios (e.g., submit declaration, review conflict, create plan, acknowledge plan, manage users).
 - Feedback gathered via observation, task completion analysis (success rates, time), and structured questionnaires (usability, clarity, satisfaction).



Validation Goals

What We Validated During Testing

- **Usability:** How easily and intuitively could users (Declarants, Managers, Administrators/Super-Administrator) complete tasks compared to manual methods?
- **Requirement Fulfillment:** Did implemented features directly address functional requirements for each role?
- **Improvement Over Manual Process:** Was the system perceived as more dependable, transparent, and efficient (mainly for Managers/Declarants)?
- **Effectiveness:** Did the system fulfill its intended role supporting the end-to-end CoI lifecycle and admin tasks?

Positive feedback and successful task completion confirm EDS offers a better, practical solution.



Table of Contents

4 Demonstration

- ▶ Introduction
- ▶ Project Development
- ▶ Evaluation
- ▶ **Demonstration**
- ▶ Conclusion and Future Work



Table of Contents

5 Conclusion and Future Work

- ▶ Introduction
- ▶ Project Development
- ▶ Evaluation
- ▶ Demonstration
- ▶ Conclusion and Future Work



Conclusion

Key Achievements

- Successfully designed and implemented a working **Electronic Declaration System (EDS)** web platform.
- Addressed inefficiencies and lack of transparency in NU's manual CoI process.
- Delivered a customized computing solution automating the CoI declaration lifecycle (submission, review, management plan creation/acknowledgement).
- Supported by distinct Manager, Administrator, and Super-Administrator roles.
- Enhanced compliance tracking, administrative efficiency, and provided a more transparent, user-friendly interface compared to previous methods.



Future Work

Potential Enhancements and Next Steps

- **Enhanced Analytics & Reporting:** Advanced reporting for Managers (trends, risks); improved monitoring dashboards for Admins.
- **Integration with NU Systems:** Explore integration with EDMS, 1C, or Bitrix.
- **Mobile Interface:** Develop native mobile apps (Swift for iOS, Jetpack Compose for Android).
- **Workflow Customization:** Allow Admins/Managers to modify review processes or notification templates via Admin interface.
- **Technical Refinements:**
 - Implement more comprehensive logging/error handling, integrate with NU systems.
 - Expand unit and integration test coverage for greater reliability.
 - Optimize DB queries/backend processes based on load testing.
 - Fine-tune UI elements based on user feedback.
 - Add more granular controls/customizable reminders to notifications.



Q&A

Thank you for listening!
Your feedback will be highly appreciated!