

Computer Science Department
Senior Project Final report – Fall 2025

Title of the project:	"NU Registrar System"
Team Members:	Azamat Kaliyev , Abzal Ziyabek , Medgat Auken , Sultan Khabibulin, Yeldos Nurpeissov
Project Advisor/Co-Advisors	Askar Boranbayev
Executive Summary	
The given project will address one of the most widespread problems in academia the antiqueness of registrar systems, the lack of user-friendliness and appropriateness in the times of high enrollment rates. The current workflow is not intuitive to the students and lacks feedback and visibility on why some actions were not possible to be done, and teachers tend to use some extra software to manage registrations. The aim of this project is to develop a contemporary registrar system, which would address the mentioned problems. The proposed registrar system will be a web-based application that will include a number of modules, which are inter-linked: a student portal, a registration engine and administration module. All the required academic processes such as the browsing process, registration, dropping of courses, swapping of courses, waiting list processes and administration would be integrated into our registrar system. The scholastic regulations like prerequisites, time clashes, section capacities, holds and restrictions would be applied and also easy feedback on all the actions of failure. This project was undertaken in a systematic manner to software engineering in which, requirements gathering, system architecture, user interface design, implementation plan and evaluation plan were involved. This brought about a fully operational prototype which demonstrates the fundamental operations in a registrar.	
Introduction	
The registrar systems are very important in the administration of the university since they are in charge of class enrollments, scheduling, student record management and administration. The inability of some of these registrar systems to be easily navigated and their slow responses will create a lot of confusion amongst students and add up to the workloads of the administrative staff members.	

The NU Registrar System was developed to address these issues by developing a more friendly interface, which will allow students to accomplish the course registration tasks. This project aims at developing a web portal to provide students and administrators registrar functions in a single web page.

The proposed system includes the following major components:

- a course catalog for browsing and searching available courses,
- section pages for viewing enrollment options,
- a schedule view for checking conflicts,
- a registration engine for enroll, drop, and swap actions,
- a waitlist mechanism for full sections,
- an administrative interface for course and capacity management.

The most important consideration is clarity. The functionality is key for this project, but we must not forget about user experience. As an example, when the system fails to register, then it must inform the user of why, not just issue a generic error message. This report summarizes the limitations of the project, literature background, design technique, implementation, assessment and future research.

Identification of constraints

The NU registrar system development process has been guided by a number of limitations in both its technological aspects and design considerations.

Technological limitations

During the busiest registration period, many students will be trying to sign up using the platform concurrently. It is crucial that the system be able to handle this without slowing down to keep pace with demand. Therefore, efficiency is one of the critical criteria of its development, as it needs to provide good performance even in such busy periods while being easy to develop.

Legality and ethical considerations

As the platform deals with sensitive student data, certain expectations regarding legal compliance and privacy need to be met. Namely, there should not be any unnecessary information collected about the users or exposed through the application, as the data only needs to relate to the functions of a university registrar. Role-based access control would be needed here as the system allows interaction by two different groups of people.

Constraints on usability and accessibility

For a registrar system to be usable, its usage needs to cater to people with various skills in terms of technical background and accessibility issues. This is why the project is designed in such a way that high importance is given to the navigability, legibility, stable elements, and consideration of the WCAG 2.1 AA guidelines. The project has a new plan that gives a clear outline of how the accessibility and usability will be prioritized during the design of the user interface.

Constraints imposed by the organization

The scope of the project needed to be established since this is a senior level project with established deadlines and expectations by the academic environment. A registrar solution on an enterprise level cannot be constructed as a full-scale solution; therefore, there is a necessity to create only an MVP with a focus on the key workflows of user registration.

Constraints on sustainability and maintainability

The software should be simple to comprehend and to the developer in the future. By extension, the implementation of modular architectures, web technology standards and separation of concerns were important in the project implementation stage.

These constraints shaped the final design by encouraging the team to prioritize:

- clarity over complexity,
- modularity over monolithic structure,
- privacy over unnecessary data exposure,
- and practical MVP delivery over feature overload.

Background and Related Work

Typically, universities use either student information systems or registrar modules that are based on enterprise resource planning software. While those systems can offer a fully functional solution, they usually suffer from aging architecture and were developed several years ago. Thus, they might be inconvenient for extension and integration with other systems, as well as have an outdated interface.

In this regard, one of the major drawbacks of traditional registrar systems is a poor real-time interaction experience. Students cannot receive a prompt answer regarding the reason for the failure of some request, the status of enrollment into specific sections, or whether there are other options available. Additionally, the integration process with such solutions as LMS systems, login providers, or billing systems may require writing additional code.

The suggested by our team concept is much more modern in terms of architecture and can offer such advantages as:

- it improves maintainability,
- it supports cleaner integration,
- it makes the system easier to scale,
- and it allows the UI to present information in a more transparent way.

These issues have already been mentioned in the previous project report, and there was a clear rationale behind the development of a modular approach with an emphasis on an API-driven registrar. The revised requirements and the project plan only emphasize this strategy even more.

Project Approach

The NU Registrar System was created in the form of a web application, wherein there is a distinct division between its front-end and back-end components. The application follows the principle that the former is only to be used in the presentation, navigation and interaction process, whereas the latter is the official one.

System architecture

The fundamental domains of the registrar are arranged in the system architecture:

- authentication and roles,
- course list and time-table,
- registration engine,
- records and transcripts,
- administrative control.

Such design fits in perfectly with the service approach mentioned in the previous report and enables a modular development of the software. In addition, it can be expanded in the future without having to redesign the entire system.

Frontend structure

The revised project plan outlines a page hierarchy of a registrar style with role-based navigation. The most important pages are:

- course browsing/filtering catalog,
- course/id to see course and sections,
- section/{id} to enroll/drop/swap,
- weekly schedule visualization schedule,
- admin/* to have staff-facing administration tools.

This enables a natural progression of navigation where a student can look up a course, go to the course detail page, look at available sections, can he have a scheduling conflict and whatever action he wants to take.

The system is constructed on the basic registrar processes:

- course search and filtering,
- enrollment,
- dropping a course,
- swapping sections,
- waitlist support,
- schedule conflict visualization,
- management of sections and capacity administratively.

Immediate and clear feedback on frequent registration issues like: is also highlighted in the project plan.

- conflicts,
- full sections,
- missing prerequisites,
- and holds.

Handling rules

One of the most important steps in the design of the software, perhaps, would be to make sure that the registration rules are addressed adequately and that they are clearly explained. Denying specific actions of users is not enough since the action is illegal, the justification of the disapproval must be explained to the user.

Tools and technologies

The standard set of development tools assisted the team in their work:

React on the frontend,

API-based backend communication,

Version control: GitHub,

Jira, task tracking,

Figma to plan and prototype interfaces.

Team organization

The tasks were divided into front end, back end and concept/ proto- type tasks. This made it possible for the team members to do their tasks concurrently without affecting the consistency of the overall design.

The plan demonstrates clearly the roles and responsibilities of who does what in regards to the project activities.

Project Execution

The project implementation was carried out in two semesters with a starting phase of requirements elicitation to an implementation-oriented design.

Stage one: Requirements and high-level design.

The focus of the group during the first semester was specification of the scope of the project, identification of the core problem, and design direction. The process involved:

- identifying user needs,
- describing the purpose of the system,
- describing key registrar processes,
- and developing a modular architecture.

This is what the above report describes this stage as a combination of lightweight agility, document management, and services orientation.

Second stage: developing a more comprehensive implementation plan

During the second semester the group changed the design work into a more tangible plan of implementation of the project. The polished set of requirements consisted of defined frontend routes, defined interactions, and administrative elements. It was a considerable step towards making the project implementation prepared as it detailed the interactions between the UI and the backend components.

Key decisions during implementation

Some of the major decisions that were arrived at in the execution of this project were:

Scope control:

- The registrar side is vast, but the team deliberately aimed to address fundamental registration functionality initially.

Clear error messaging:

- Messages were made in meaningful way instead of confusing fail messages

Role-based behavior:

- Access and view of same system were different for the students and administration respectively

Modular development:

- This type of development made it easier to divide work across the team, coordinate it quickly

Challenges encountered**The main challenges were:**

- controlling scope without blowing the project out,
- dealing with the intricacies of academic rules,
- and integrating work between the team members in various elements.

The manner in which the team managed them.**The team replied by:**

- ready MVP,
- programmable logic,
- components have owner,
- making design choices based on real registrar processes.

All in all, the implementation of the project was successful, since the team was focused, flexible in adjusting the design where needed, and the end product was user-focused.

Evaluation

This project was examined in terms of usability, functionality, and performance. The aim of this evaluation was not to confirm that the system works but to prove whether or not the system enhances the experience of the registrar.

Functional evaluation

The system must effectively facilitate the main registrar functions:

- enroll,
- drop,
- swap,
- waitlist handling,
- and rule-based validation.

The fact that the features exist should be not only ensured but also their proper operation in such circumstances should be guaranteed. E.g. the system must allow no invalid action to be performed using such constraints as conflicting action, capacity, prerequisite or hold. This aspect is crucial to the very essence of what a registrar system is about.

Usability evaluation

The project plan particularly focuses on usability testing, particularly whether or not the user can:

- find courses easily,
- understand conflicts,
- and learn to lose mistakes.

This is an excellent assessment standard because one of the greatest shortcomings linked with registrar systems is not just technical- dependability but also ambiguity. The user-friendly system would minimize confusion, make decision making easier and enable students to get their work done faster.

Performance Evaluation

As stated earlier the system is busy during times of registration. In this way the system must be tested in terms of its performance under load. Also a mere multi-user test will be useful where this would indicate whether the architecture can withstand the pressure of having many students doing similar operations simultaneously. With one of the key weaknesses of the system as it has been made clear in the earlier report peak load performance, performance becomes a key evaluation criteria in this instance.

The contents of the final submission.

- the amount of test cases they run,
- the percentage or number of passes in the tests,
- examples of typical cases of failure that have been dealt with,
- an abbreviated report of usability feedback,
- and any performance testing observations.

This is to say that it cannot just show that there is testing, but it must also support the project by showing that these results confirm that the system under consideration is effective and reliable.

Conclusion and possible future work

An instance of a smart and a modern solution to a university registration system construction is the NU Registrar System Project. The project does not recreate all the issues with the past versions of such programs, instead it is based on transparency, modularity and clear rules. Through this, the solution would be more appropriate to the users who need quick and transparent registration processes.

The main finalised item in the project is a working prototype that will contain all of the features needed in a well-organised system. This will be capable of registering, managing wait-lists, looking at schedules and managing administration with a realistic MVP scope of the academic project.

Improvements that can be made current system:

- degree requirement audit,
- powerful reporting and analytics,
- extensive load testing,
- increasing participation with the university systems,
- and increased audit/logging facilities.

Additions such as these would imply a more complex system, one that is not taken off-track of the course that the current system sets. The project, therefore, serves as a prototype and a good foundation to make subsequent developments.

References

- [1] Nazarbayev University, NU Registrar official website and online course catalog (accessed Spring 2026).
- [2] Django Software Foundation, "Django Documentation," <https://docs.djangoproject.com/>.

[3] Meta, "React Documentation," <https://react.dev/>.

[4] World Wide Web Consortium (W3C), "Web Content Accessibility Guidelines (WCAG) 2.1".

[5] Group 25, "Updated requirements and design – NU Registrar System," Spring 2026.