

Nazarbayev University Housing Management System

Dilnaz Aitmaganbetova, Guldana Zhubandyk, Alma Serikkyzy, Ainur Taskynova, Bakdaulet Dauletov

BSc Computer Science

Nazarbayev University

Astana, Kazakhstan

I. EXECUTIVE SUMMARY

A. Project Overview

Nazarbayev University Housing Management System is a web-based application intended for automation the current dormitory allocation procedure. The system enhances application processing using Large Language Model (LLM). It connects students, housing staff, and admins in one platform.

B. Objectives and goals

- Automation document processing through LLM;
- Reducing housing members workload;
- Allow students to submit their applications;
- Decrease document processing time;
- Enable housing staff to review applications;
- Allow students to pay for accomodation through the system;
- Allocate students efficiently;
- Replace manual checking with hybrid method (manual checking+AI based)

C. Methodology

The project uses full-stack client-server architecture where frontend and backend was connected using Rest API. Next.js and React were used for developing frontend, while the Gin framework was used to implement backend in Go.

The team used an Agile and Iterative development method, where the project was developed incrementally over two semesters. First of all, the project was already included important components such as authentication, application submission, and dashboards for each roles (Admin, Student, and Housing members). After that, the project was extended by adding automated document verification, decision logic, and room allocation.

D. Results

Housing Management System was developed successfully and resulted in several key outcomes:

- Built a system that connects students, housing staff, and admins on a single platform.
- Students can submit their application, upload necessary files and check their application status.
- Housing members can review applications and access to student documents.
- Included LLM that checks the document contonent

- System decides approve or reject documents
- Integrated automated room allocation

II. INTRODUCTION

A. Problem Description

The main problem that was addressed is ineffectiveness of current management system of dormitory application. The current housing system collects applications through Google forms and managers review each application. This process usually takes a long time due to confirming of eligibility of submitted documents, allocating rooms and sending emails to dormitory residents. So, the method is effective, but time consuming and raises the possibility of human errors.

B. Motivation

Student Housing managers spend a big amount of time processing the applications manually. Moreover, manual review may result in disparity due to the reason that housing staff retrieve identical information from a large amount of comparable documents. Also, it is crucial to make the system transparent and efficient because the decision regarding getting dormitory affects the life of the students. So, the new system should resolve the issue and automate the process of application and document review.

C. Proposed Solution

As a solution to the problems the system offers web application that will facilitate the process of application to the dormitory for everyone. For students, the application allows them to submit documents and instantly get the result of the application whether it is approved or not. Moreover, the students can see the reasoning of rejection or manual review and edit their application in the future. The system provides instant analyzes using an LLM. For housing managers the application gives an opportunity to save time and manually review only special cases.

III. CONSTRAINTS

A. Economic and Technical Constraints

Our project introduces cost and rate limitations because of the usage of OpenAPI key for AI-based document analysis. The costs should have been considered while testing the documents submission for the relevant work of the LLM since available credits could eventually run out.

B. Ethical, Social, and Political Constraints

Fairness has to be maintained during decision making. It was also important to make the results understandable and allow for manual review as necessary. Transparency was essential since poor decisions could have a serious social impact on students. From a social point of view, transparency was necessary as poor decision-making can lead to students distrusting the work of the Housing.

C. Health, Safety, and Sustainability

The project does not include dangerous equipment or immediate physical safety concerns therefore health and safety risks are reduced. However, the reliability of the system remains important. Failure of the system in the housing application phase may lead to students delaying decisions or causing uncertainty regarding deadlines and outcomes.

D. Risks, Trade-offs, and Limitations

A balance between automation and manual review was required to be considered. AI decision making does not guarantee complete dependability, although it can lead to efficiency. Therefore, a certain amount of simple and unclear cases were automated for manual examination which also facilitates the efficiency and fairness of the system.

IV. BACKGROUND AND RELATED WORK

A. Related Research

Nowadays, most student housing and hostel staffs use the traditional method of document processing. Recent research shows that this is less effective due to slow processing time and increased housing members' workload. In order to solve this issue, several web-based housing management systems were introduced offering document processing and room allocation. For example, [1] proposed hostel web-application that uses Artificial Intelligence (AI) to allocate rooms automatically.

In a more recent investigation, [2] created a system for student room allocation using modern software engineering methods. The project offers application submission, document verification and automatic room allocations. The authors used Django, React, PostgreSQL, and Docker to develop web-application. They found that this method can reduce manual checking and the possibilities of human error.

Another important component of related research is automated document verification. The system should not only receive applications but also check that uploaded documents satisfy requirements. According to [3], manual document verification in higher education has issues, while AI-based systems improve fake documents detection and review efficiency.

B. Comparative Analysis

Compared to existing housing systems, our method is more focused on document verification. Many systems offer digital submission and room allocation but it does not replace manual document checking. The project implements LLM based document evaluation. Students upload their applications and upload required files to the system, which extracts required

information. As a result of AI checking, the system decides to give approve, reject or send it for manual checking for housing members. This method addresses the main problem of manual checking in uncertain situations.

V. PROJECT APPROACH

A. Development Process

To build this system we used an incremental development approach. In the beginning we created a simple system that have user authentication, admin features, basic student housing application and housing dashboard tables. This helped us establish the foundation of this system. The initial version of the system interface is shown in Figure 1.

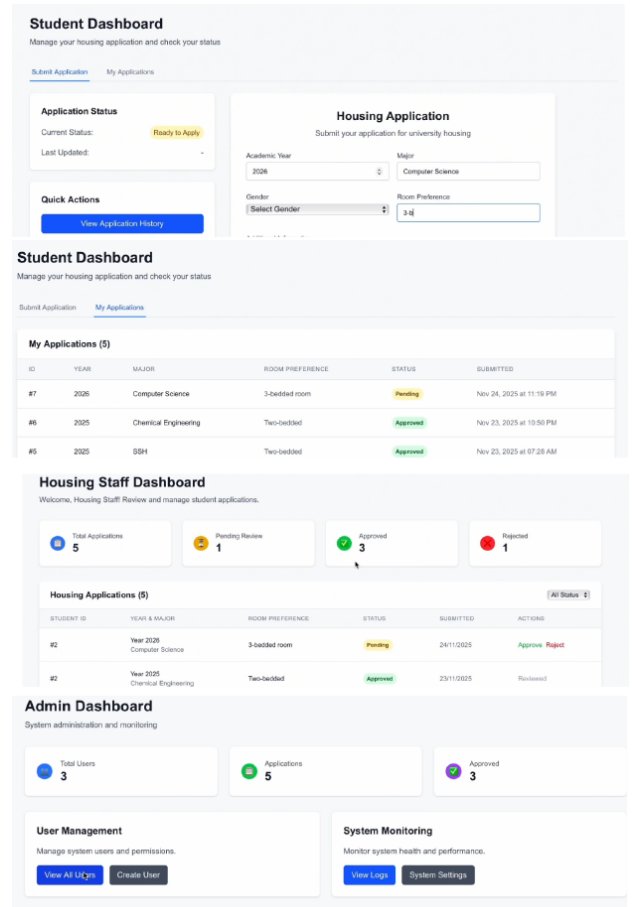


Fig. 1. Initial version of the housing management system interface

After that, we divided tasks among the team members and improved the system's automation and usability. We introduced pdf document upload and more fields to get full information in student application form. In role based dashboards we made interface more user friendly. In subsequent phases, we focused on how to automate housing application review system, we integrated LLM-based document verification, email notifications and payment system. The updated version of the system is shown in the Figure 2.

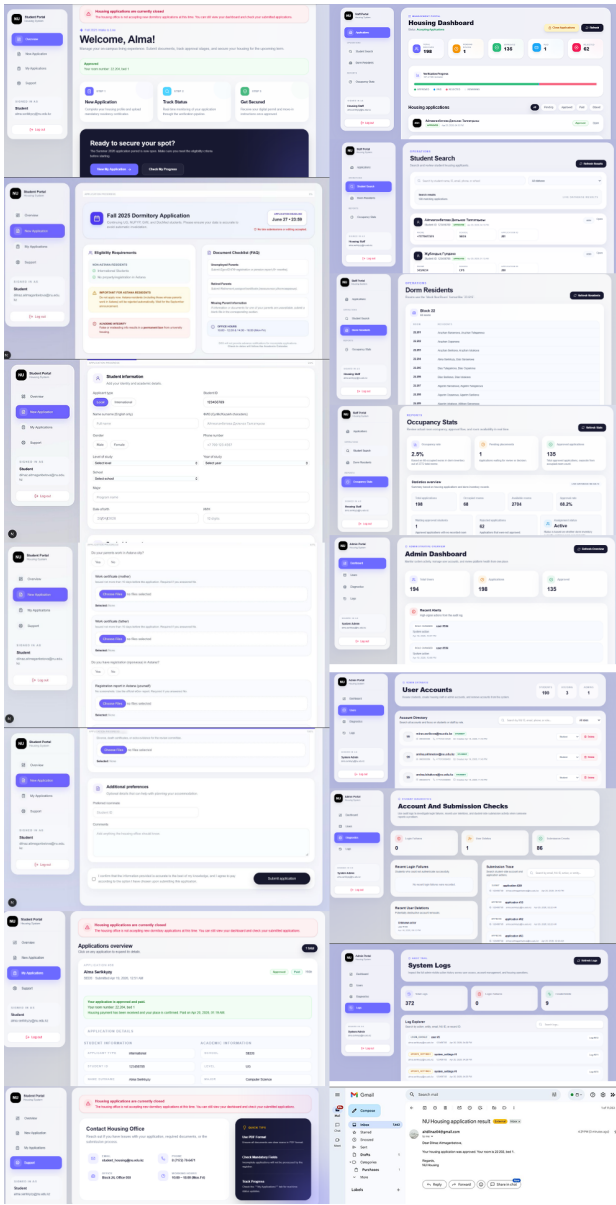


Fig. 2. Updated version of the System

We considered all edge cases that may appear in the early-semester housing application process, like duplicate applications and incorrect document submissions, incorrect room allocation. At each stage, we tested and verified that the system works without bugs. Using an iterative approach allowed us to develop stable and scalable solution.

B. Architecture Design

For this system, we used client-server architecture, where frontend and backend code are developed separately and communicates using RESTful APIs. This design allowed us to divide the tasks to exact team members and work efficiently. The client side of the system was built using Next.js and React, and handles user interaction and displaying data. All three roles in our system (admin, housing, student) had their own

dashboard interface. The backend side of the system was developed in GO using Gin framework. It handles business logic, request processing, communication with external services, and authentication. We used backend API endpoints to connect the frontend with system functionalities like users, applications, documents, payments and system settings. Supabase was used as a PostgreSQL-based database service to store user records including all their documents, payments, system records. The PDF files that are uploaded to the system are stored in MiniO and their references and related data are saved in the database. To extend system's functionality, we integrated external services like LLM for automated document analysis, Stripe for payment processing and SMTP services for email notifications. They are shown in the Figure 3 and 4.

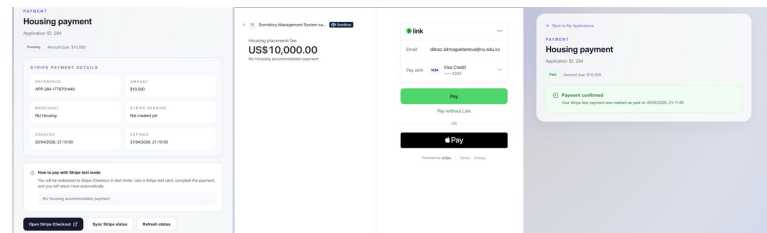


Fig. 3. Payment Processing

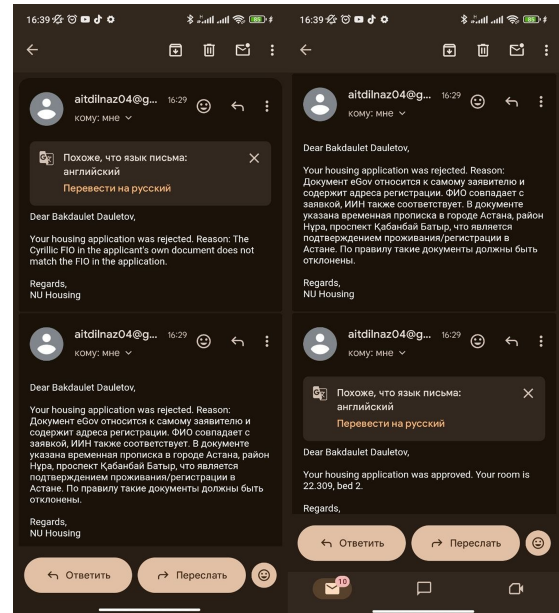


Fig. 4. Automated Email notifications

For authentication, we used JWT tokens, and role-based authorization ensures users can only access relevant features. The Architectural Design Model diagram is shown in Figure 3 to understand this section better.

C. Technologies and Tools

The system was built using these technologies:

Frontend

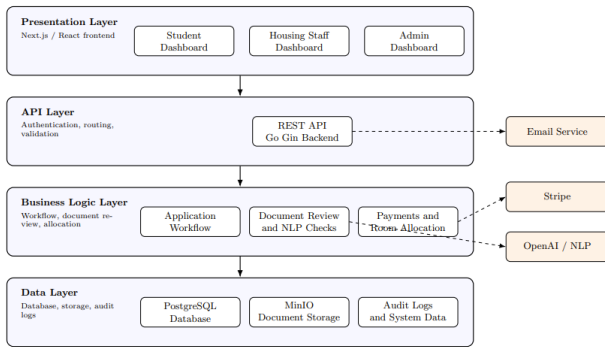


Fig. 5. The Architecture of the System

- Next.js 16 - used for building the web application
- React 19 - for interactive user interfaces
- TypeScript - improves code reliability and maintainability
- Tailwind CSS - for the UI design
- lucide-react - for icons used in the website

Backend

- Go - used for backend development, chosen for its performance and efficiency
- Gin - built REST APIs in this framework
- Supabase - (PostgreSQL-based) - storing and managing the data
- MinIO - storing uploaded documents
- Docker - for containerization to run the application consistently across environments
- JWT & OAuth – for secure authentication and Google ID token verification
- Stripe - payment processing
- Role-based authorization middleware - to control access for different user roles
- SMTP (Gmail) - for sending email notifications

D. Team Roles

For a smooth teamwork process we assigned roles to each team member, so they could be responsible for their respective parts of the project.

- Dilnaz Aitmaganbetova – responsible for frontend development, payment integration, and email notification features.
- Guldana Zhubandyk – responsible for backend development and LLM integration for automated document verification.
- Ainur Taskynova – responsible for backend development, documentation, database setup using Supabase, and system testing.
- Bakdaulet Dauletov – responsible for backend development, including API implementation and business logic.
- Alma Serikkyzy – responsible for frontend development, role-based dashboard implementation, documentation, and testing.

Throughout two semesters, we had regular team meetings (online/offline) to discuss the overall progress. Each member

took turns leading weekly task planning, fostering shared ownership. This approach helped us successfully complete all project goals.

VI. PROJECT EXECUTION

A. Development Process

The application was executed during two semesters. In the first semester only the fundamental platform was executed. The main work was done in the second semester where the main features were added to the system. During the second semester the project was enhanced by adding advanced functionalities like implementation of LLM for document verification, email notification system via Gmail SMTP, automatic room allocation, edit application buttons, and online payment system. Also, the database was changed from local PostgreSQL setup to Supabase for cloud deployment.

B. Key Changes

The main design choice was to replace a fully manual review system with a hybrid system that includes AI-based document analysis, rule-based validation, and manual document review. It was initially intended to rely more on automation, but as the project progressed, it was discovered that AI was not always dependable, particularly when dealing with ambiguous or scanned papers. Based on this, a manual review was developed as a backup.

Additionally, Stripe payment was integrated, which was not included in the original design. This made it possible for authorized students to finish the housing application process using the same system.

Another important addition was the automatic room assignment based on approved applications and specified limits. At first, the system just managed applicants' submissions and evaluations.

C. Challenges and Solutions

One of the most significant challenges was LLM integration for document verification. Initially, this feature occasionally misinterpreted document content or were unable to extract necessary information from ambiguous files. To address these issues, prompts were enhanced, the logic for handling documents was refined, and a manual review option was incorporated, allowing housing staff to verify uncertain cases.

Another problem involved linking the backend system with the online Supabase database. The team initially dealt with configuration and connection issues. This problem was solved through extensive research and viewing technical tutorials.

Another difficulty was that in order to make an effective system there were different external services—like payment processing, email communication, and file storage. Since every service had a different API interface and configuration demand, debugging was difficult. To resolve this aspect of this, the team tested each integration individually before integrating it into the final workflow, merging them into the final workflow.

D. Team Collaboration

Team members divided responsibilities based on their strength and abilities. While some members were responsible for frontend, others focused on backend development, database management and testing. In order to track progress and discuss problems, we held regular offline meetings. Team members helped each other to solve the problems. Leadership roles were distributed according to the task.

VII. EVALUATION

A. Evaluation Methodology

To assess the system's effectiveness, we compared it with the current manual housing application process. The review concentrated on important factors such as application processing time, automation level, and user experience.

We reproduced real-world circumstances by producing several types of student applications, such as valid, invalid, and confusing document cases. This enabled us to examine how the system handles automated decisions and when manual review is required.

Moreover, we provided questionnaire between twelve students from Nazarbayev University. We asked them to evaluate the system by submitting example applications and interacting with the site. We received feedbacks using a Google Form questionnaire.

B. Results

Twelve Nazarbayev University students were participated in the evaluation. As a result of applications:

- 9 out of 12 applications (75%) were processed automatically
- 3 out of 12 applications (25%) required manual review
- Application processing takes approximately 15-30 seconds after submission

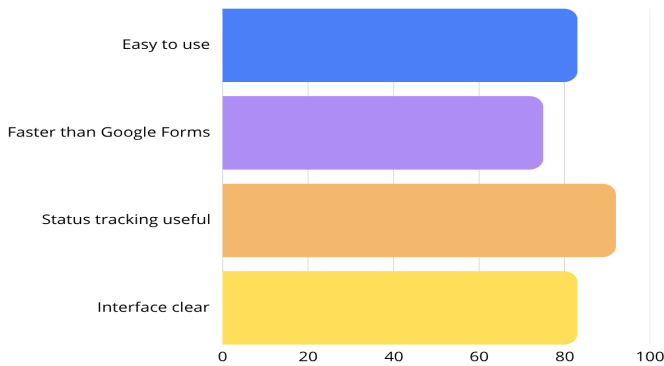


Fig. 6. User Feedback

The findings of the user feedback survey are displayed in Figure 4. Over 80% of students thought the system was straightforward and easy to use, and 92% preferred real-time application tracking. This suggests that the developed solution is widely accepted by users.

C. Analysis

The result shows how the system effectively handles the processing of applications. The majority of applications were processed automatically which decreased the workload for housing management. This allowed significantly decrease the application processment time. The system gives instant notification to the students and the process of application is now more organized by allowing for students to edit their submission. As it is shown in the survey students have a positive attitude towards the new system. The most appreciated feature was the ability to track the status of applications in real time, which is not achievable with the existing Google Forms application.

VIII. CONCLUSION AND FUTURE WORK

A. Summary

We developed web application that can replace current Nazarbayev University dormitory application process. The product effectively connects students and housing members in one single platform that includes staff review, automated verification, document upload, room assignment, and student application submission.

Overall, our project shows that modern approaches using web development and AI integrations can solve a real-world administrative problem.

B. Contributions

The main contribution of the project is AI integration enabling the system to automatically evaluate important dormitory eligibility requirements. This simplifies the procedure in general and minimizes manual checking for housing staff.

Another major contribution is the role-based system architecture, which gives administrators, housing staff, and students their own interfaces and features. This makes the system more organized and user-friendly for a different type of use.

The system also improves transparency by allowing students to track their application status and receive detailed justifications for the result of applications. Furthermore, the implementation of automatic room allocation demonstrates how the system can manage the entire workflow.

C. Future Work

Although web-application is working properly, there are many areas that could be improved in the future:

- Implement stronger OCR fallback for scanned documents without extractable text
- Improve admin tools to manually re-run document verification if results need rechecking
- Advance room allocation logic, including reallocation options
- Provide access to the system for Housing members and Nazarbayev University students to collect their feedbacks to improve usability.

REFERENCES

- [1] A. Azeeta, S. Misra, M. Odusami, O. U. Peter, and R. Ahuja, "An intelligent student hostel allocation system based on web applications," in *The International Conference on Recent Innovations in Computing*, pp. 779–791, Springer, 2020.
- [2] A. Bidakhmetov, "Automation of the student settlement process in the dormitory," *Data Science*, vol. 3, no. 3, pp. 66–71, 2025.
- [3] J. Tam, A. H. Lam, and D. K. Chiu, "Exploring the aids of machine learning and artificial intelligence for the detection of fraudulent documents in university admissions databases: A perspective from university officers and it professionals," *Library, Information & Services*, vol. 1, no. 1, p. 2, 2025.