Case Study: Notification Management in the Neptun Academic Portal

Author: Ganzorig Boldsukh (KU)

Supervisor: Prof. László Pitlik

Date: 2025.02.18

Email: pitlik.laszlo@kodolanyi.hu, boldoo.ganzoo08@gmail.com

(Please, never communicate in working phases based on PDF-files! We do need docx files e.g. for change tracking. A pdf-file is a sign for closed processes! We still have to do – even a lot…)

This document does finally offer reproducible facts/processes. The previous ones were quasi just empty bubbles without any reproducible parts.

Position in user: <https://miau.my-x.hu/miau/320/moodle_neptun_tests/Case%20Study%20Addendum%20-%20Neptun_pl.docx>

Remark about the winner: we only need one single frame, each other task should be part of this frame, creating new frames are forbidden. All change tracking signs try to explain, what we really need!

Concurrent, but late-sent files for the position in use:

* <https://miau.my-x.hu/miau/320/moodle_neptun_tests/the_single_position_for_empty_bubbles_is_already_in_use_1.docx>
* <https://miau.my-x.hu/miau/320/moodle_neptun_tests/the_single_position_for_empty_bubbles_is_already_in_use_2.docx>
* <https://miau.my-x.hu/miau/320/moodle_neptun_tests/the_single_position_for_empty_bubbles_is_already_in_use_3.docx>

Remarks about the competitive, but late-sent files:

1. \*in\_use\_1: Please, send me such a version, where the original text is black&white AND the text written by you is a kind of "change-tracking-like-red"... It is important: each text in a final thesis, what is not your own performance MAY NEVER be written without any quotation signs... or the own text should be highlighted with other unmisinterpratable techniques (like change tracking)....
2. \*in\_use\_2: A képen szöveg, képernyőkép, Betűtípus, algebra látható

   Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.
3. \*in\_use\_3: copy&send the previous 2 remarks

Motto: requirement analysis <> software testing

It is a totally other point of view, whether somebody is agreed with the realized functionalities of a software or (s)he is responsible for testing each functionality being ordered!

# Contents

Introduction: ................................................................................................................................................. 1

Functionality Analysis.................................................................................................................................... 2

Impact: .......................................................................................................................................................... 3

Focus & Objectives ........................................................................................................................................ 3

Hypotheses & Suspicions .............................................................................................................................. 3

Proposed Solutions & Improvements ........................................................................................................... 4

* UI/UX Enhancements: ...................................................................................................................... 4
* Backend Upgrades: .......................................................................................................................... 4
* Documentation & Training: .............................................................................................................. 4

Conclusion ..................................................................................................................................................... 4

Figures ........................................................................................................................................................... 5

Thesis Contribution: .................................................................................................................................... 12

Introduction: The Neptun system is a widely used educational management platform, providing students and faculty with access to academic records, communication tools, and administrative functionalities. However, students often face challenges in managing notifications efficiently within the system. This case study examines the limitations of Neptun's notification management, compares them with standard email services, and proposes improvements to enhance user experience. 🡨if we are working in a predefined frame, it is never needed to write about the frame as such… We only need appropriate (reproducible) examples for the predefined positions!

Problem statement:Students using Neptun encounter difficulties in organizing and

managing notifications on their dashboard.<--typical point of view of an personally unsatisfied user! See MOTTO above! The default sections—Upcoming Events, Results, Messages, Fulfilled Credits, Exams, News, To Do, Averages, and Debts (Figure#1, Figure#2, Figure#3.1-3.9)—serve essential functions, but critical usability issues arise in the following sections:

* **Messages**: No “Select All” or “Mark as Read” functionality, making it cumbersome to track new messages. (Figure#5, Figure#7, Figure#8) 🡨New expectations <> detected errors! Please, never forget: your final thesis and/or your software could also be evaluated based on the unsatisfied-principle. Result: each final thesis would have a grade = 1! Each evaluator has good fantasy to dream new expectations instead of the existing ones chosen/covered by the author!
* **To Do:** Contains persistent notifications that cannot be marked as read, causing unnecessary clutter. (Figure#5, Figure#6, Figure#10) **🡨this part is not even a command for realization of clear-communicated parameters**
* **Results:** Students receive grade notifications without an option to organize or dismiss old entries. (Figure#5, Figure#11) **Dream-status! 😊**

Unlike modern webmail services such as Gmail, Outlook, and ProtonMail, which provide efficient message organization tools, Neptun lacks these basic features, forcing (maybe wanted?:-) students to manually open each notification to track new updates. (Figure#4.1, Figure#4.2, Figure#4.3)

## Functionality Analysis

Key functional shortcomings in the *Messages, Results*, and *To Do* sections (Figure#3.1, Figure#3.3, Figure#3.4) include:

1. **Manual Workflow Dependency**:
   * Users must open each message individually to mark it as read. o Persistent *To Do* items (e.g., outdated reminders) cannot be dismissed or archived.
   * *Results* notifications remain static, even after grades are reviewed.
2. **Lack of Bulk Actions**:
   * No options to **select multiple notifications** for batch operations (e.g., mark as read, delete).
   * No **filtering** or **sorting** tools (e.g., by date, urgency, course).
3. **Ambiguous Status Indicators**:
   * No visual distinction between "read" and "unread" items in *Messages* or *Results*. o *To Do* tasks lack progress tracking (e.g., "pending" vs. "completed").

Impact: Simple notification issues could be cause of increasing cognitive load and risking oversight of critical academic updates.

* **Time Consumption:** Students must open each message individually to check for updates, making notification handling tedious.
* **Inefficiency in Organization:** Without categorization or read/unread status management, important messages can get lost among older notifications.
* **Mental Load:** Persistent To Do notifications create unnecessary cognitive stress by constantly displaying unresolved tasks.
* **Limited Control:** The lack of user-configurable filters and batch actions reduces efficiency, leading to frustration and missed updates.

## Focus & Objectives

**Core Focus**: Redesign Neptun’s notification system to align with user expectations and industry standards.

**Objectives**:

1. **User-Centric Assessment**:
   * Identify pain points through student surveys and usability testing. o Benchmark against platforms like Gmail, Canvas, and Moodle.
2. **Technical Analysis**:
   * Evaluate backend logic for notification persistence (e.g., why *To Do* items cannot be marked complete).
   * Audit frontend code for scalability in implementing bulk actions.
3. **Solution Design**:
   * Propose UI/UX improvements to enhance clarity and efficiency.
   * Develop a roadmap for phased implementation.

## Hypotheses & Suspicions

1. **Hypothesis 1**: The absence of bulk actions stems from a database architecture limitation.
   * *Suspicion*: Notifications may lack unique identifiers or batch-processing APIs.
   * *Evidence*: Manual workflows suggest server-side restrictions on bulk updates.
2. **Hypothesis 2**: Persistent *To Do* items result from incomplete backend triggers.
   * *Suspicion*: Tasks are not auto-archived upon completion (e.g., after grade submission).
   * *Evidence*: Similar issues in legacy academic systems often involve unlinked task-status modules.

## Proposed Solutions & Improvements

* UI/UX Enhancements:
  + **Bulk Action Toolbar**: Add "Select All," "Mark as Read," and "Archive" buttons above notification lists (*Messages, Results and To Do*).
  + **Status Indicators**:

▪ Use **color-coded badges** (e.g., red for unread, gray for read).

* + **Smart Filters**: Let users sort notifications by date, course, or type (e.g., "Unread Messages").
* Backend Upgrades:
  + **Batch Processing API**: Develop an endpoint to handle bulk requests (e.g., /notifications/mark-read).
  + **Auto-Archiving Logic**: Link *To Do* tasks to backend events (e.g., auto-archive after grade submission). o **Real-Time Updates**: Implement WebSocket connections to refresh notifications dynamically.
* Documentation & Training:
  + Publish a **student-facing guide** with screenshots/videos explaining notification management.
  + Add **tooltips** and **interactive tutorials** within the dashboard.

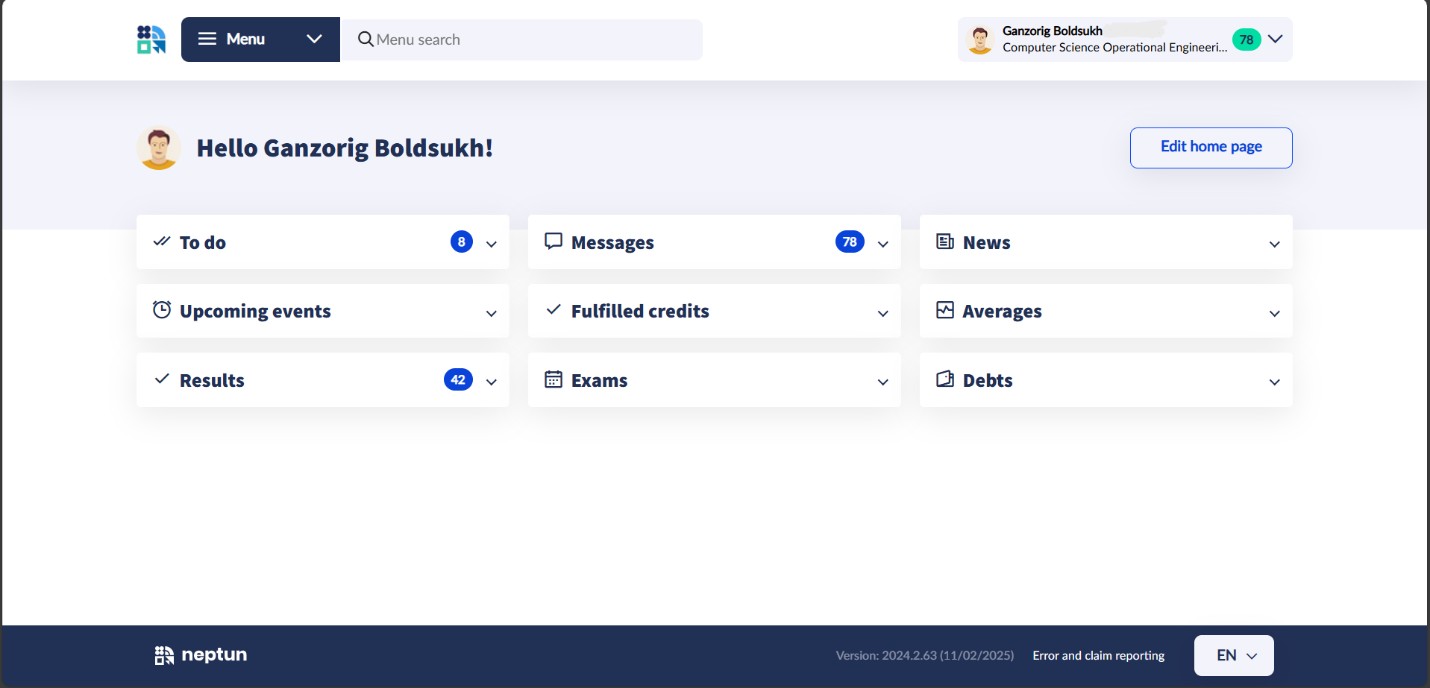
## Conclusion

The Neptun dashboard’s notification system fails to meet modern usability standards, creating inefficiencies that hinder academic productivity. By addressing these gaps through:

1. **Frontend redesign** (bulk actions, visual cues),
2. **Backend optimization** (batch APIs, auto-archiving),
3. **Comprehensive documentation**,

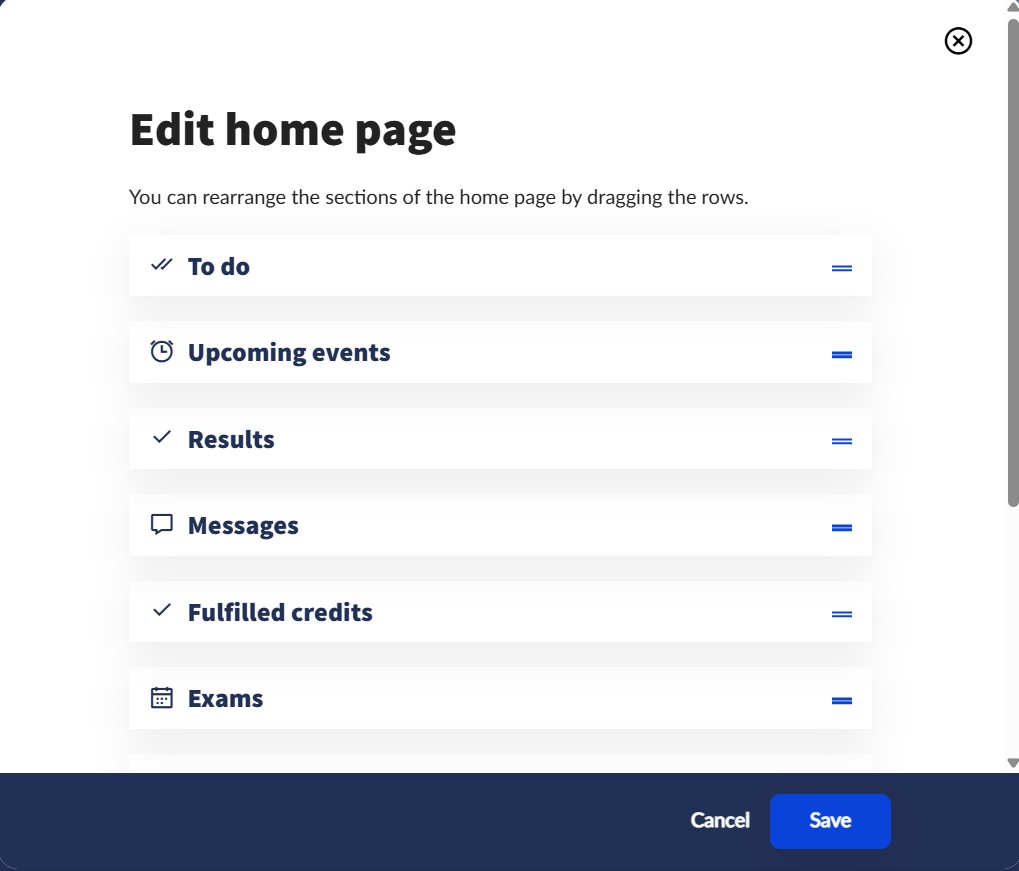
Neptun can transform its notification module into a streamlined, user-friendly interface. Prioritizing these changes will reduce cognitive load, minimize errors, and align the platform with institutional goals of supporting student success.

## Figures



Figure#1: Home page of User in Neptun Dashboard

(URL=https://neptun.kodolanyi.hu/hallgato\_NG/dashboard)



Figure#2: Edit section of Default home page in Neptun Dashboard

(URL=https://neptun.kodolanyi.hu/hallgato\_NG/dashboard)



Figure#3.1: To do section in edit home page section

(URL=https://neptun.kodolanyi.hu/hallgato\_NG/dashboard)



Figure#3.2: Upcoming events section in edit home page section

(URL=https://neptun.kodolanyi.hu/hallgato\_NG/dashboard)



Figure#3.3: Results section in edit home page section

(URL=https://neptun.kodolanyi.hu/hallgato\_NG/dashboard)



Figure#3.4: Messages section in edit home page section

(URL=https://neptun.kodolanyi.hu/hallgato\_NG/dashboard)



Figure#3.5: Fulfilled credits section in edit home page section

(URL=https://neptun.kodolanyi.hu/hallgato\_NG/dashboard)



Figure#3.6: Exams section in edit home page section

(URL=https://neptun.kodolanyi.hu/hallgato\_NG/dashboard)



Figure#3.7: News section in edit home page section

(URL=https://neptun.kodolanyi.hu/hallgato\_NG/dashboard)



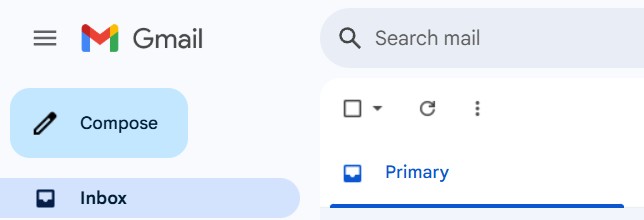
Figure#3.8: Averages section in edit home page section

(URL=https://neptun.kodolanyi.hu/hallgato\_NG/dashboard)



Figure#3.9: Debts section in edit home page section

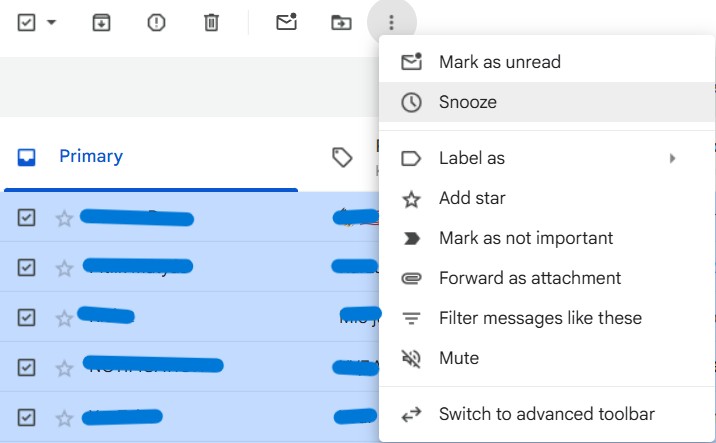
(URL=https://neptun.kodolanyi.hu/hallgato\_NG/dashboard)



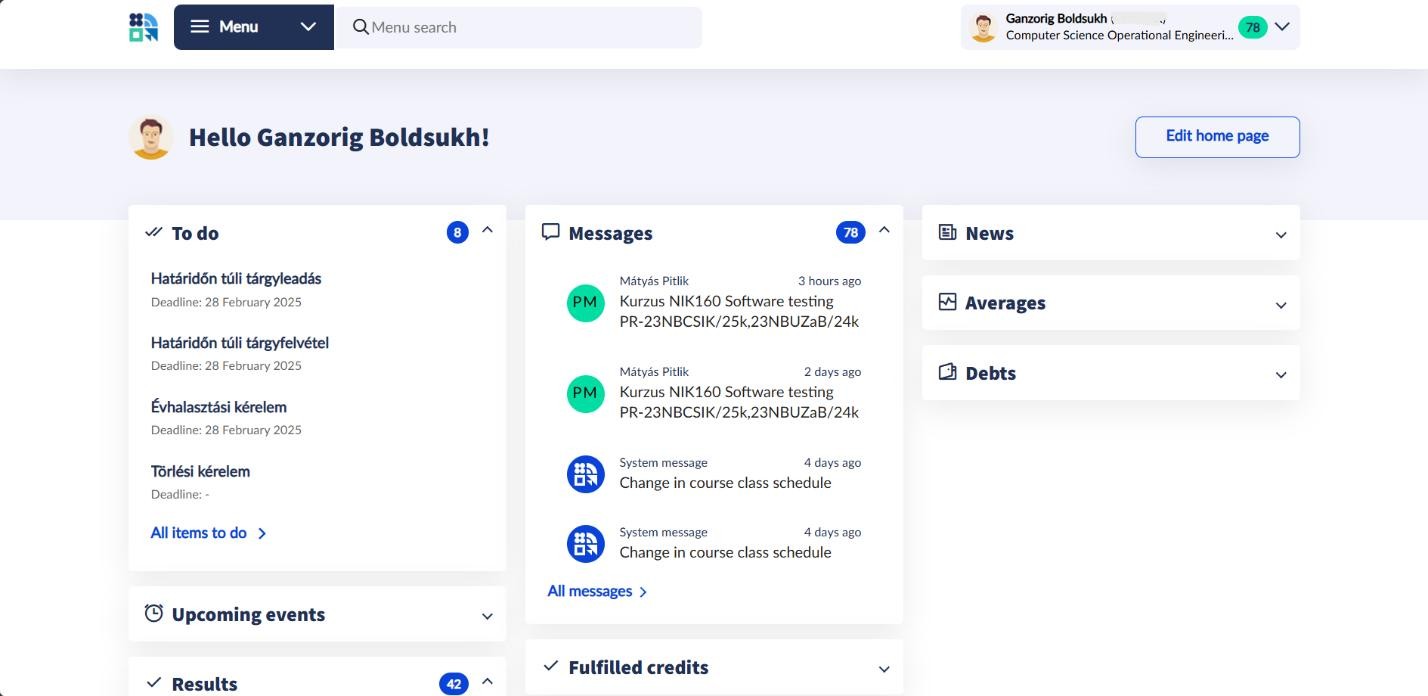
Figure#4.1: Gmail Select button (URL=https://mail.google.com/mail/u/0/#inbox)



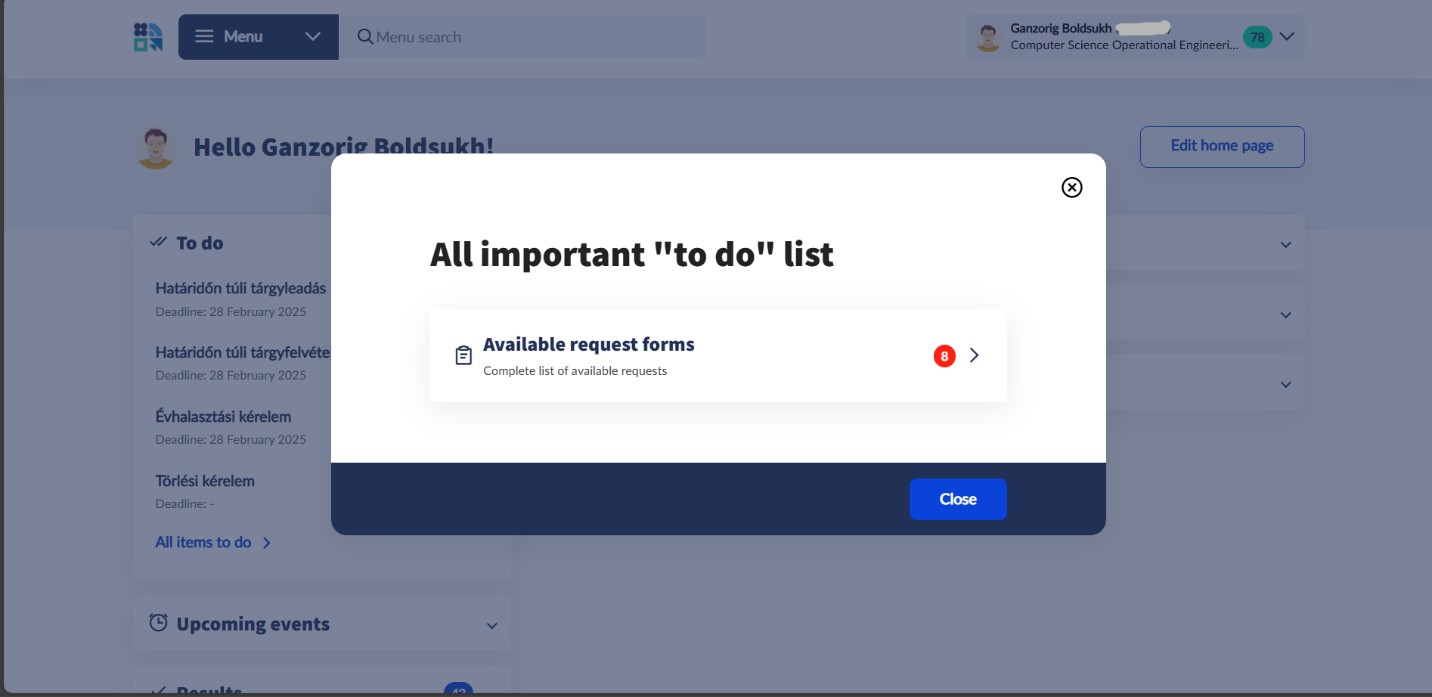
Figure#4.2: Gmail mark as read button (URL=https://mail.google.com/mail/u/0/#inbox)



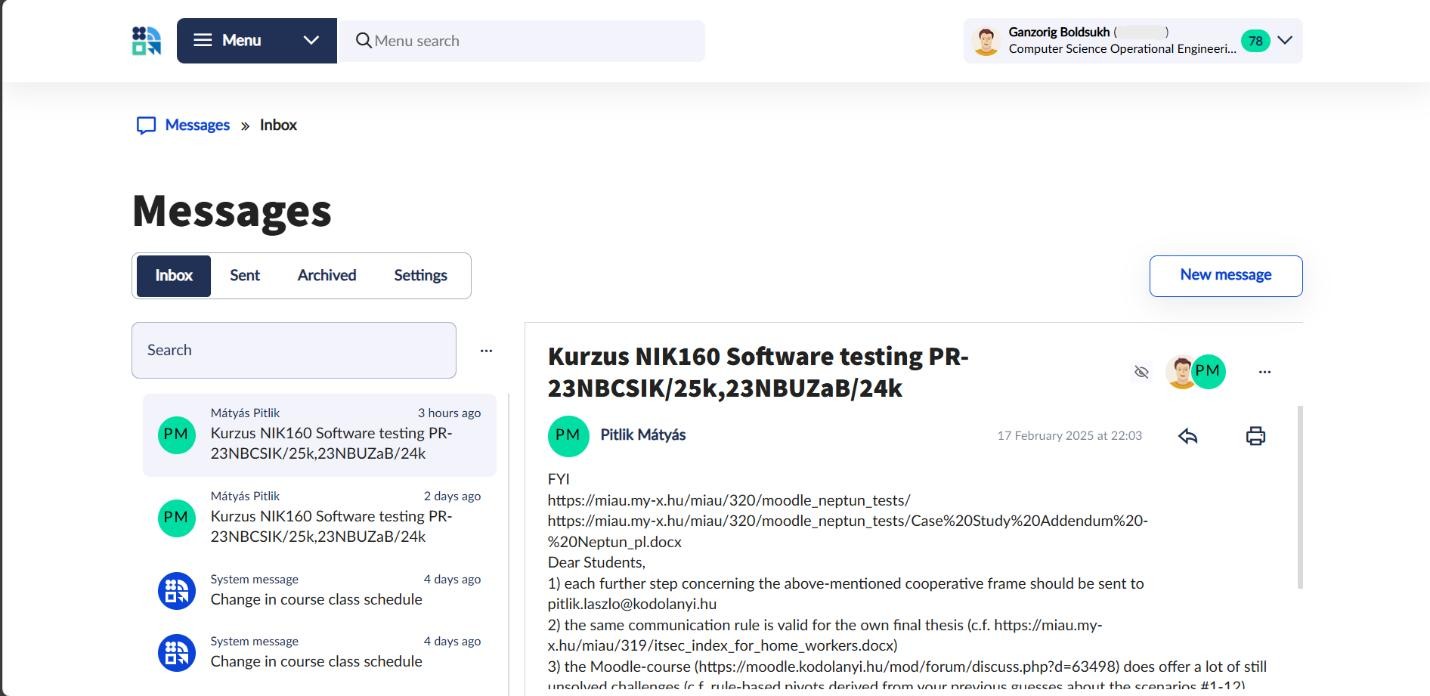
Figure#4.3: Gmail Mark as unread button (URL=https://mail.google.com/mail/u/0/#inbox)



Figure#5: Home page notification (URL=https://neptun.kodolanyi.hu/hallgato\_NG/dashboard)

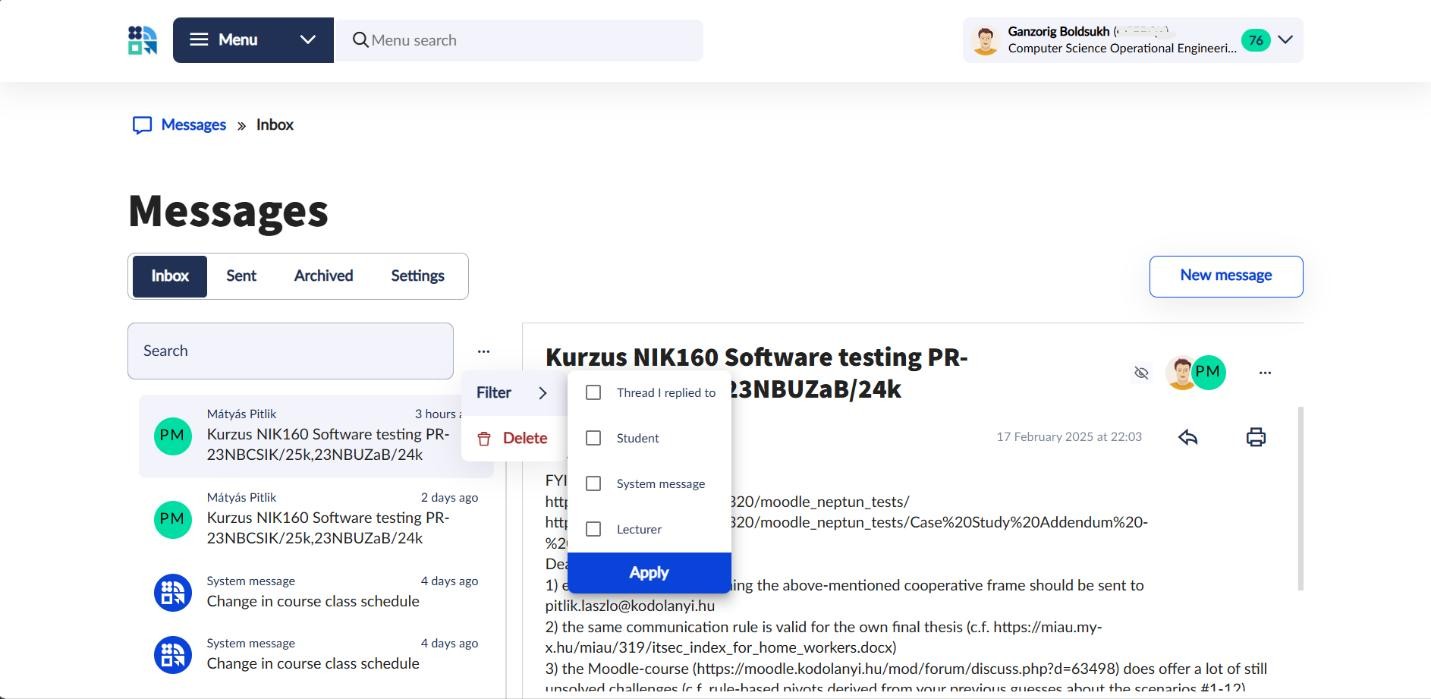


Figure#6: To Do section (URL=https://neptun.kodolanyi.hu/hallgato\_NG/dashboard)



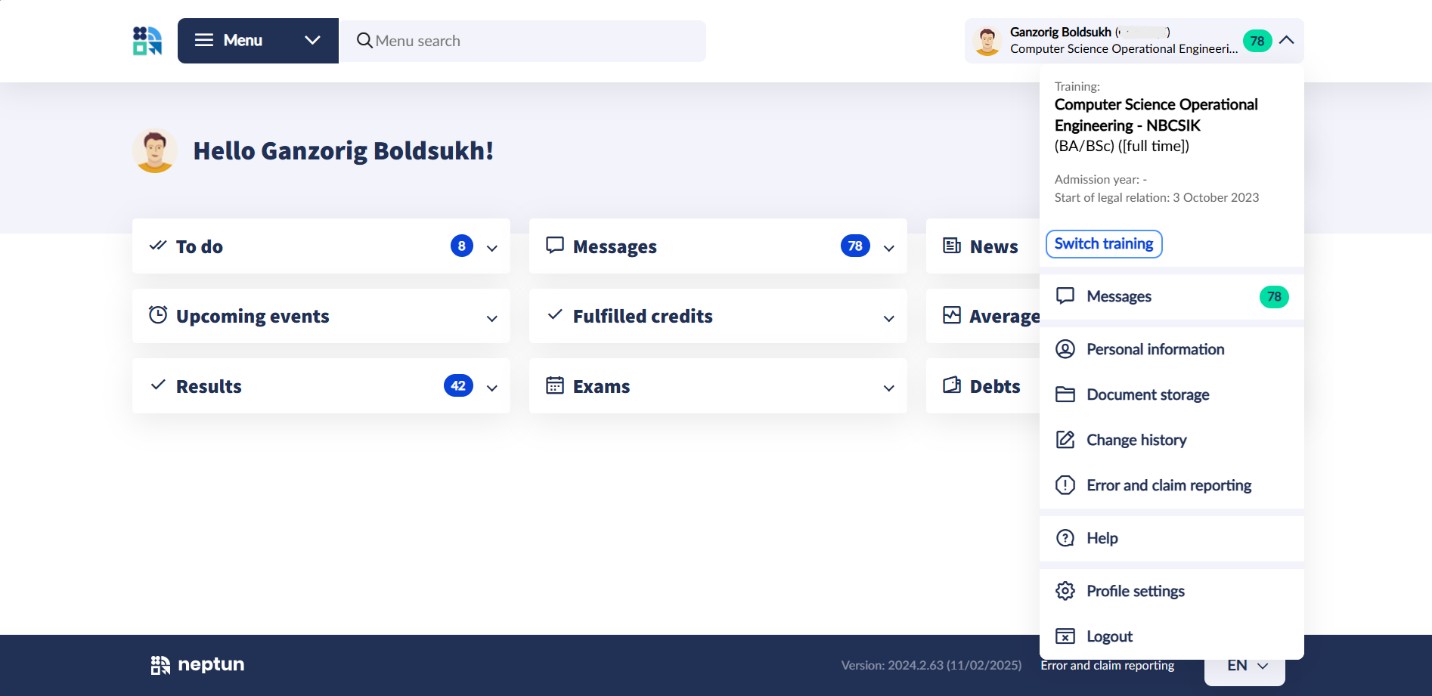
Figure#7: Content of Messages section

(URL=https://neptun.kodolanyi.hu/hallgato\_NG/messages?message-tab=0&tab=Inbox)

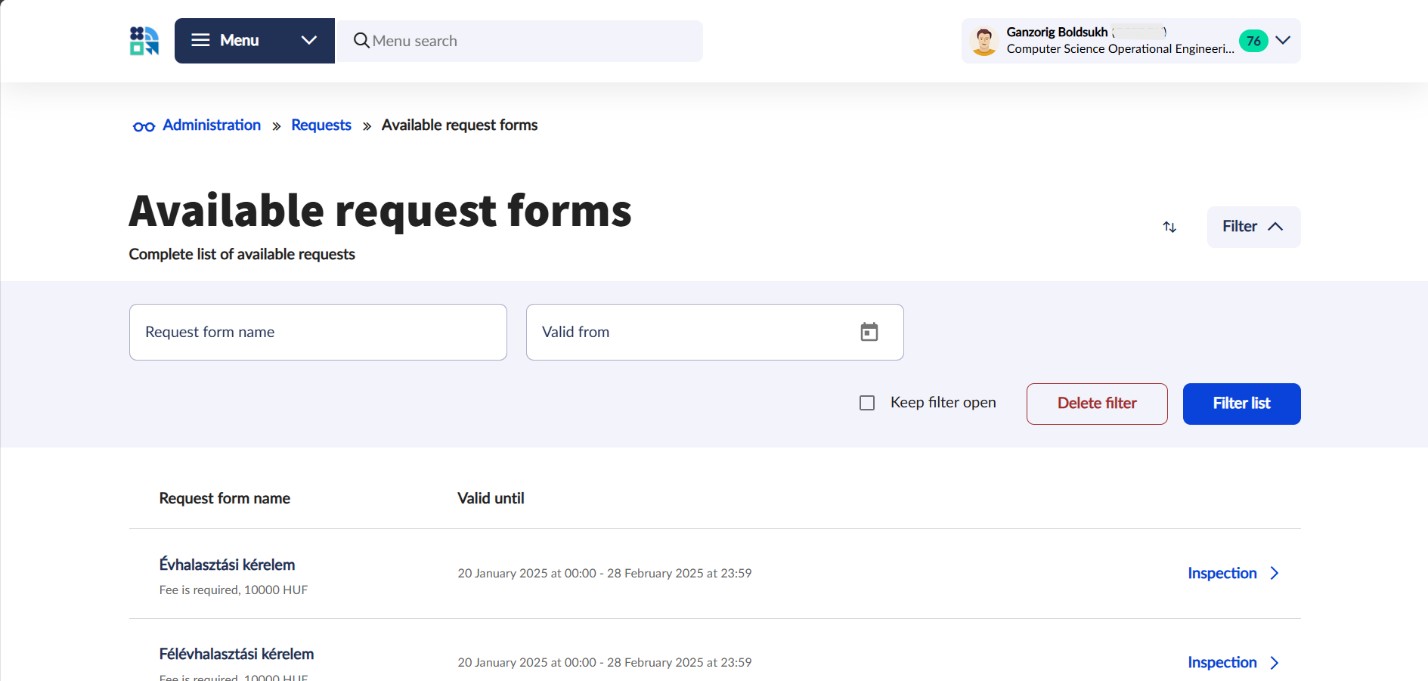


Figure#8: Options of Messages section

(URL=https://neptun.kodolanyi.hu/hallgato\_NG/messages?message-tab=0&tab=Inbox)

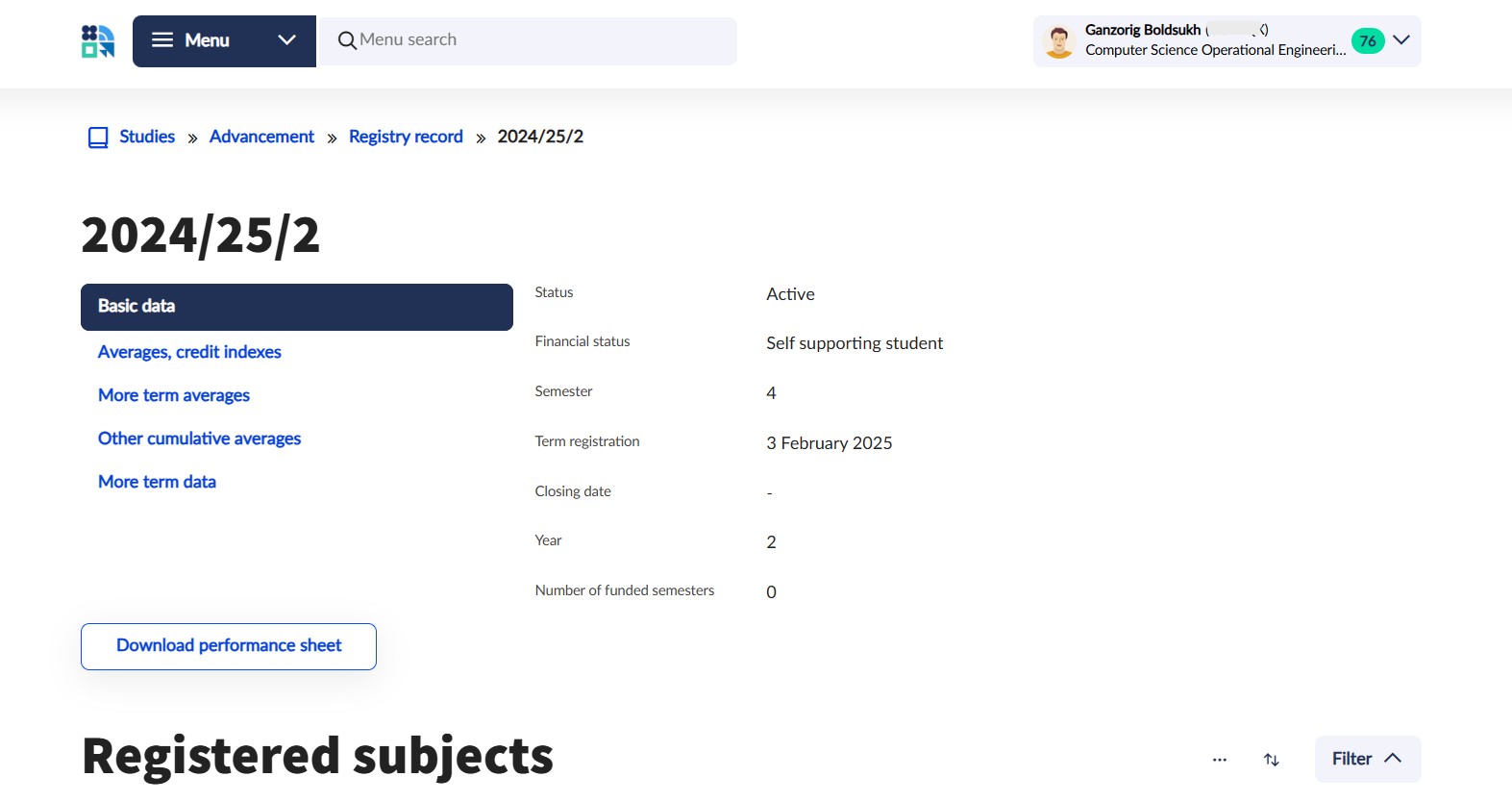


Figure#9: User advanced optimization (URL=https://neptun.kodolanyi.hu/hallgato\_NG/dashboard)



Figure#10: Content of To Do section

(URL=https://neptun.kodolanyi.hu/hallgato\_NG/administrations/request-forms/to-be-completed)



Figure#11: Content of Results section

(URL=https://neptun.kodolanyi.hu/hallgato\_NG/studies/advancement/registrysheet/204518091?term=2024%2F25%2F2&tab-selector-semester-details=0)

Thesis Contribution: This study provides an actionable blueprint for Neptun’s developers

to bridge the gap between administrative functionality and student-centric design, fostering a more intuitive academic ecosystem.