TurBo Grader:
Automatic Grading Tool for Programming Projects

Supervisor: *Prof. Dr. Süleyman TOSUN*

Fatih Ay, Furkan Aydın*, Zübeyde Civelek**

**Department of Computer Science, Hacettepe University, Ankara, Türkiye*

MOTIVATION

The motivation behind TurBoGrader's development stems from the prevalent challenges encountered in manual code assignment evaluations. Traditional assessment methods are fraught with issues such as human error, subjective judgments, and time constraints. These methods often fall short in objectively reflecting students' true performance. TurBoGrader aims to address these shortcomings by enhancing the automated code assignment evaluation process.

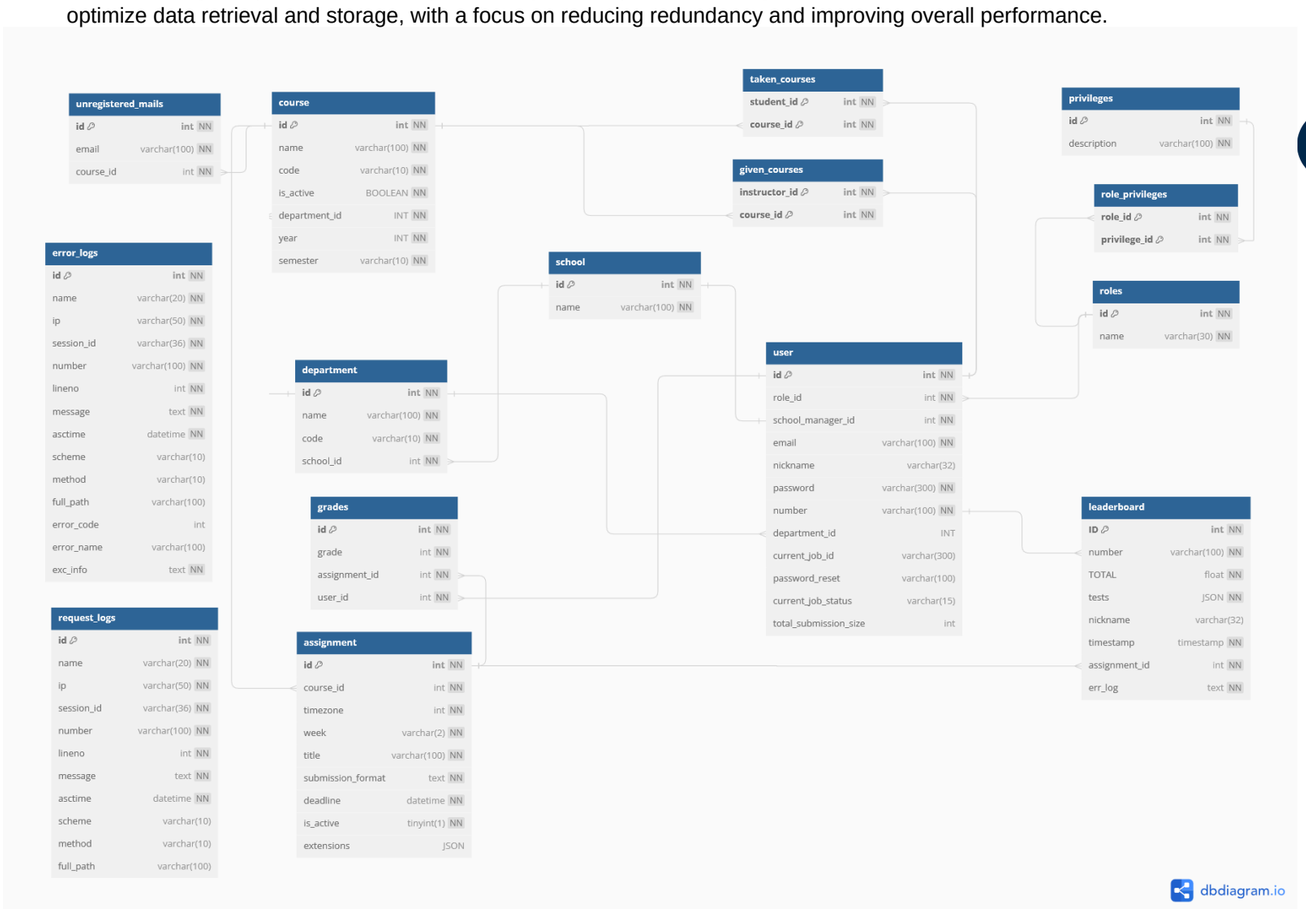
In addition to these, TurBoGrader ensures that data remains within the country, enhancing data security and privacy, thus catering to the specific needs of the country. This emphasis on local data storage and processing not only ensures compliance with local regulations but also mitigates risks associated with data transfer across borders. By prioritizing data security and privacy, TurBoGrader provides users with peace of mind and fosters trust in the system. Moreover, a local solution often offers faster and more reliable technical support, further bolstering its appeal and usability within the country

AIM

The primary aim of the TurBoGrader project is to transform our existing system into a versatile and robust educational tool capable of grading multiple assignments across diverse courses simultaneously. This involves restructuring the database and codebase to support concurrent auto-grading, developing an interactive platform for student feedback and grade reviews, and optimizing the user interface for seamless assignment management. Ultimately, we aspire to not only match but surpass the capabilities of leading solutions, offering advanced features such as in-system quizzes, comprehensive statistical analyses, and integrated notifications, thereby setting a new standard in automated academic evaluation.

DESIGN CHANGES

Significant changes have been made to the database structure to accommodate the evolving needs of TurBoGrader. The new design emphasizes scalability, efficiency, and improved data organization. Tables have been restructured to optimize data retrieval and storage, with a focus on reducing redundancy and improving overall performance.

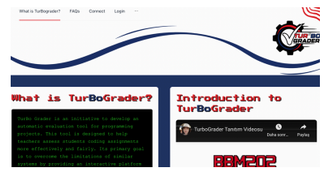


Database changes introduced features like assigning multiple assignments simultaneously and incorporating department structures, enhancing TurBoGrader's ability to handle complex workflows.

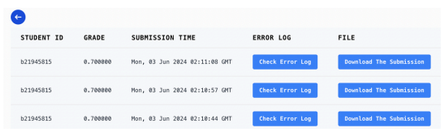
The web application has been refactored to follow the Model-View-Controller-Service (MVCS) architecture, improving code organization and maintainability. The front-end has been rebuilt with React, enhancing performance and user experience.

The back-end was redesigned as an API, improving integration with the front-end and data handling flexibility. These changes result in a more robust and scalable web application.

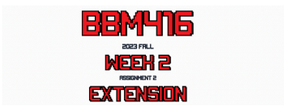
NEW FEATURES



- **React Integration:** The system has been updated to integrate React, a modern JavaScript library, into the front-end, enhancing user interface responsiveness and interactivity.
- **Improved Logging System:** A new logging system has been implemented, accompanied by a user-friendly interface, making it easier for administrators and users to track system activities and errors efficiently.

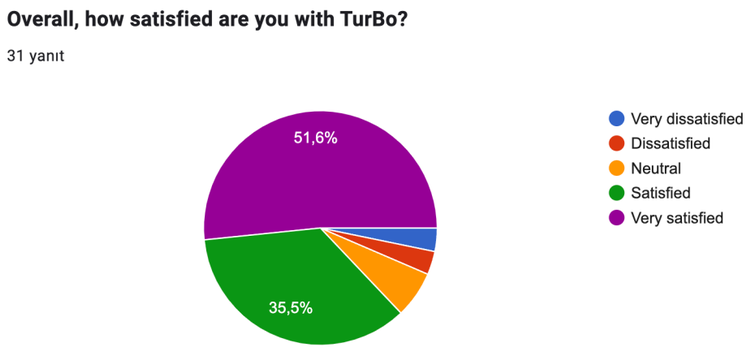


- **Access to Old Submissions:** Users now have the ability to download and review past submissions, allowing for better analysis and comparison of previous work.
- **Enhanced Submission Analysis:** Instructors can now easily access and review submission results, error logs, and submission histories, providing valuable insights into student performance and progress.
- **Expanded Download Options for Instructors**: Additional download options have been added for instructors, including the ability to download student grades and submissions in Excel format, streamlining assessment and grading processes.
- **Database Changes for University Structures:** The database has been updated to accommodate new structures such as schools, departments, facilitating better organization and management of educational data.
- **Late Submission:** A late submission feature has been added, allowing students to submit their assignments after the deadline



- **Transition to Nearly Atomic-Level Privilege-Based Permission System:** The system has transitioned from role-based to nearly atomic-level privilege-based permission system. As a result, new roles can be easily accommodated without the need for code changes, providing greater flexibility and scalability in managing user access and permissions. This enhances the overall security and adaptability of the system, ensuring that users have precisely the permissions they need without unnecessary restrictions.

Survey Results



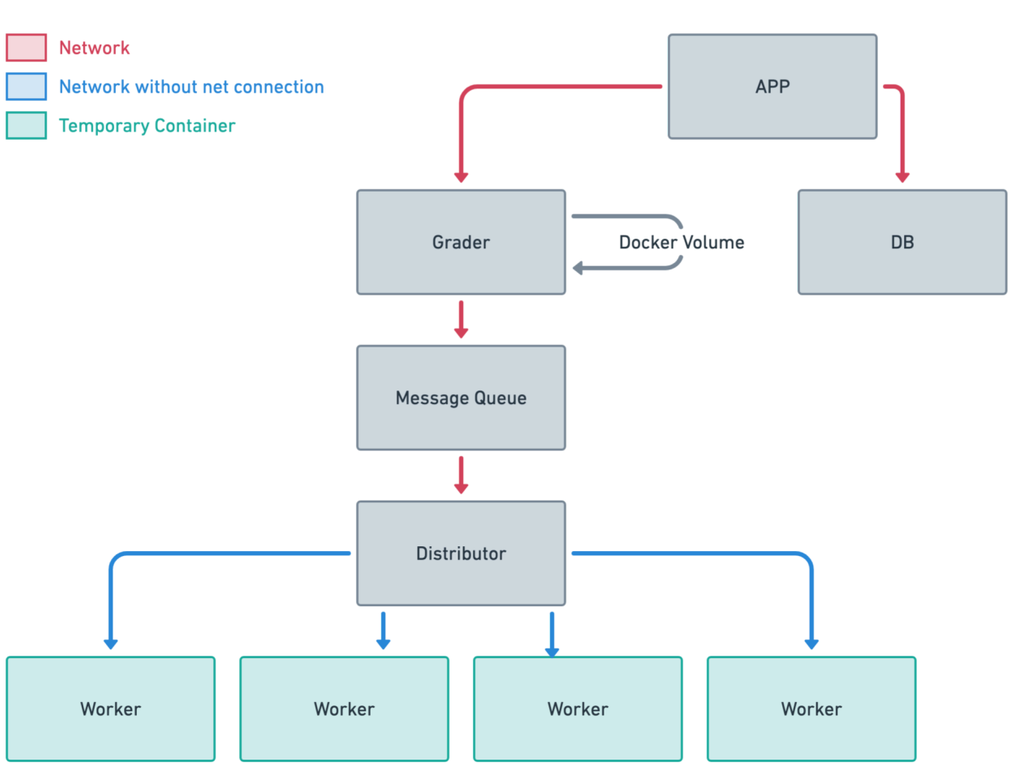
The survey was done at the end of the project and the results show %87.1 of Users were satisfied with TurBo Grader overall.

GRADER CHANGES

We completely revamped the Grader part of TurBo, making it a robust and scalable automatic grading system. By addressing known threats and implementing a Distributor, we scaled the number of Workers and prevented them from interacting with the message queue.

TurBo's sandbox environment uses Docker to made sandbox environments with specific resource limitations (CPU, memory, processes). Using a premade Docker image as a base, which includes the OS and settings, we reduced environment set up time to under a second.

Testing and user surveys show a remarkable reduction in queue times from 60-30 minutes during peak times to almost no wait at all.



Student Feedbacks

"It is much faster. In the previous semester, we were waiting in the queue for like 30 mins sometimes. But this semester I never waited."

"I think it is faster and I can see my previous grades (which I believe is a huge improvement). I don't think there is anything worse than the previous version."

"The wait time for the grader has really slowed down, which is also a great thing, it used to keep us waiting last semester."

"Thanks to interactive grading, I had the chance to see my mistakes instantly. This was hardly possible without TurBo."

"Seeing my history is great since it helps with calculating my overall grade. Everything else is fine, no downs in the latest version."

"We can see our past submissions and did not face any long queues while testing."

CONCLUSIONS

TurBo Grader has improved in terms of features, scalability, and security. It has been used in the Algorithms Laboratory for a semester and the feedback was quite positive towards to new changes. As a future work, the implementation of LLMs to check malicious code and allow users to interact with their code is aimed.

Acknowledges

We would like to express our sincere gratitude to **Res. Asst. PhD SELMA DİLEK and Res. Asst. Alperen Çakın** for their assistance and support throughout the course of the project.