

# CSRS - Custom Feedback Form for Assignments

Engineering Design Review

**Author:** Michael Gale

**Date:** 20 October 2024

## Introduction

Providing effective feedback is crucial for promoting student learning and improving academic performance in educational assessments. However, the current feedback mechanism in the Student Submission Review System (CSRS) exhibits significant limitations. At present, markers are restricted to providing a numeric score and a single free-text feedback box for each student submission. This approach restricts the depth and specificity of feedback, preventing markers from comprehensively evaluating different aspects of an assignment.

To address this issue, we propose the development of a new feature that allows module organizers to specify custom feedback forms for each assessment. These forms will include input fields for component marks corresponding to various aspects of the assignment, multiple free-text fields for distinct feedback areas, and the ability to label submissions with tags (e.g., "1st software engineering project") for easier filtering by markers. This solution will enable module organizers to design and implement structured feedback forms that facilitate more effective and systematic feedback from markers, while ensuring that students receive targeted and meaningful evaluation comments.

This new feature not only enhances the quality and efficiency of feedback but also aligns with our goal of improving the student learning experience and support, ultimately contributing to the institution's mission of fostering academic excellence.

## Goals and non-goals

- **Goal:** Ensure that each custom feedback form includes at least three input fields per component (e.g., score fields and text boxes) to facilitate comprehensive evaluations.
- **Goal:** All feedback provided by markers must be recorded in the database, ensuring that each feedback entry is associated with the corresponding assignment and marker.
- **Goal:** Enable students to access their feedback within 24 hours of submission to promote timely review and learning.
- **Non-goal:** Implement an automated grading feature for assessments.
- **Non-goal:** Include course content management functionalities as part of the overall system.

## Design Overview

The custom feedback form feature we are designing will allow module organisers to create tailored feedback forms for each assignment. These forms will include input fields for grades and text feedback, enabling markers to provide structured assessments of student performance.

## System Architecture

This feature will be integrated into the existing CSRS system, leveraging its role control and data storage mechanisms. Module organisers will use a new user interface element to create feedback forms, while markers will fill out the forms through the existing assignment grading interface.

The feature will be implemented using the following key components:

1. **Form Creation Interface:** A page for module organisers to create custom feedback forms. Each form will contain at least three components, including grade fields (e.g., numeric inputs) and multiple free-text fields for detailed feedback. Each component can also include instructions to guide markers on how to complete it.
2. **Form Filling and Data Storage:** Markers will see the custom feedback form linked to each assignment during the grading process. The feedback they provide, including both grades and comments, will be saved instantly in the database, ensuring each piece of feedback is associated with the correct assignment and marker.
3. **Feedback Viewing Interface:** Students will be able to access their feedback within 24 hours of submitting their assignments via the CSRS interface. The feedback will be displayed according to the structure of the custom form, ensuring that students can clearly see how their work was assessed across different components.

## Data Model and Storage

The CSRS system already uses several tables related to assignments, and this new feature will reuse existing data structures while introducing new tables for form definitions and feedback storage:

- **assignment:** Stores basic information about each assignment, such as the deadline and the associated module.
- **assignment\_submission:** Stores each student's assignment submission details, including submission status and grades.
- **feedback\_form (new table):** Stores the structure of each custom feedback form, including component names, grading items, and feedback fields.
- **feedback\_entry (new table):** Stores the feedback provided by markers, including individual grades and free-text feedback, ensuring all data is linked to the assignment, marker, and submission.

## Security and Access Control

CSRS's existing role control mechanism will remain in place. Module organisers will have the ability to create and edit feedback forms for their assignments, while

markers can only view and fill out feedback forms for the submissions assigned to them. Students will only be able to access feedback related to their own submissions.

To ensure security and privacy, we will follow CSRS's current access control patterns, adding authorisation checks and logging functionality to new controller routes, preventing unauthorised access.

### **Scalability and Future Improvements**

This design supports scalability by employing a flexible form creation mechanism and data storage model. Future enhancements could include features like automated grading, additional feedback form templates, or detailed analysis reports.

Another possible extension involves introducing a mechanism to monitor and compare markers' grading patterns. By analyzing how strict or lenient each marker is, the system can flag markers who consistently grade more harshly or leniently than their peers. This would enable module organisers or administrators to take action, ensuring grading consistency across the board and improving fairness.

### **Alternatives**

#### **Alternative Solution for Custom Feedback Forms**

Provide several classic templates for module organizers to use, reducing the complexity and time required to create feedback forms. Module organizers can select appropriate templates based on different types of assignments, allowing them to quickly generate feedback forms.

- Pro: Module organizers can complete feedback forms more quickly, reducing repetitive work.
- Pro: By providing standardized templates, the consistency and comprehensiveness of feedback content are ensured.
- Pro: Module organizers can make slight adjustments to the classic templates to fit specific assignment needs.
- Con: While providing templates speeds up the process, they may not fully meet the personalized needs of some assignments.
- Con: Regular updates and maintenance of these templates are required to ensure their effectiveness and relevance.

#### **Alternative Solution for Student Access to Assessor Feedback**

Introduce a tiered notification system that provides assessors with different levels of feedback notifications based on students' scores. The specific implementation is as follows:

#### **System Mechanism:**

## Notification Levels:

- **Normal Notification:** When a student's score is above the passing grade, the system provides basic feedback information, labeled as "normal."
- **Warning Notification:** If a student's score falls below the passing grade, the system triggers a "warning" notification, emphasizing the feedback that needs attention.
- **Strong Warning Notification:** For students who consistently fail multiple times, the system issues a "strong warning," reminding assessors that additional attention is needed.
- **Pro:** Helps assessors quickly identify students who require attention, allowing for more targeted support.
- **Pro:** The tiered notification mechanism enables assessors to effectively adjust their focus on different students, ensuring resources are allocated appropriately.
- **Con:** If notifications are too frequent, assessors may overlook some important information.
- **Con:** Implementing a tiered notification mechanism in the system may add complexity to development and maintenance.

## Milestone

*Milestone 1:* Develop backend logic that allows module organizers to create and configure custom feedback forms. The forms should include scoring fields, multiple free-text boxes, and be associated with specific assignments. Once completed, you can begin testing these configuration features to ensure they correctly store data in the CSRS database.

*Milestone 2:* Build and refine the user interface, allowing module organizers to easily create and edit custom forms through a simple, intuitive page. The interface should ensure that graders can easily fill out and submit feedback forms. After this, initial user testing can be conducted to gather feedback from various departments.

*Milestone 3:* Implement functionality for graders to use custom feedback forms for each submitted assignment and ensure the form data is stored in the CSRS database. The goal of this step is to guarantee that each feedback submission is correctly linked to the corresponding student and assignment, and can be displayed to students after grading is complete.

*Milestone 4:* Allowing students to view their feedback within 24 hours after grading is completed, through CSRS. Ensure that feedback is stored in the database and displayed correctly on the frontend.

*Milestone 5:* Introduce a notification system that sends tiered notifications to students based on grading outcomes. For example, if a student receives a failing grade, the system sends a warning notification, with the intensity of notifications adjusted based on the student's performance. This will be managed through a backend task.

*Milestone 6 : (Optional):* Implement a mechanism to compare the grading strictness or leniency of graders, allowing for monitoring and ensuring fairness and consistency in grading. This milestone is dependent on departmental priorities and can be introduced as a future enhancement if the system performs well initially.

## **Dependencies**

- *UI Team:* Responsible for designing and refining the interface where module organizers create and configure custom feedback forms, as well as the interface for graders to submit feedback and students to view it.
- *Database Team:* This team will be responsible for ensuring that the feedback data is properly stored in the CSRS database. They will also manage table configurations to link feedback submissions to specific assignments, graders, and students.
- *Frontend Development Team:* Responsible for implementing the frontend pages where module organizers, graders, and students interact with the system. This includes creating and editing feedback forms, filling them out, and displaying the feedback to students.
- *Notification Team:* In charge of implementing the notification system for students based on grading results. This team will ensure that the tiered notifications function properly depending on the performance level, and that notifications are triggered automatically when necessary.
- *Testing Team:* This team will test the overall system, ensuring that all functionalities—from form creation to student notifications—are working as intended. They will perform both user acceptance testing (UAT) and performance tests on the feedback forms and notification system.
- *Legal Team:* Ensures that all new features comply with institutional and data privacy regulations, especially regarding student feedback storage and notification processes.
- *Localization Team (Optional):* If the system will be used in multiple languages, this team will localize the content of the feedback forms, notifications, and user interface to meet the language needs of various regions or universities.

## **Cost**

The custom form project is built on the foundation of the existing system, which is already quite comprehensive. Since this project only adds some custom forms to the current architecture, there will be no need for a new database or additional server hardware. Consequently, maintenance costs will remain unchanged, and we do not anticipate any extra operational expenses. All new features will be implemented within the existing resources, ensuring the sustainability of the project.

## **Privacy and security concerns**

The custom form project does not involve collecting any new personal data, as it is built on the existing system that does not require introducing new roles or

permissions. This system has already processed student data in accordance with privacy regulations. While the existing system's access controls are in place, it is essential to ensure that the new forms and related features adhere to the same permission management principles to prevent unauthorized access. Additionally, comprehensive security testing and vulnerability assessments should be conducted before releasing updates to identify and fix any potential security vulnerabilities.

## Risks

Risk	Potential Impact	Mitigation Measures
Poor User Experience	Poorly designed interface or inconvenient operations lead to decreased efficiency for assessors, preventing them from using the system proficiently.	Aim to design a simple and user-friendly interface, and provide instructional videos on the homepage to help assessors learn how to use the system.
Data Leakage	Unauthorized access or leakage of personal information may lead to decreased user trust and legal liability.	Conduct regular audits and monitoring of data access to ensure data security and promptly address any potential vulnerabilities.
Inability to Send Notifications Promptly	Failure to send notifications to underperforming students or those who have failed multiple times may result in a lack of timely attention to problem students.	Implement an email system to send additional reminder emails to students with issues to ensure they receive timely notifications.

## Supporting material

- Cybersecurity, Critical Infrastructure. "Framework for improving critical infrastructure cybersecurity." URL: <https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.4162018> (2018): 7.